

# QGIS Desktop 3.44 User Guide

QGIS Project

2026 년 03 월 14 일

# Contents


**1 시작하기** **3**
- 1.1 QGIS 설치 . . . 3
- 1.2 QGIS 실행 및 종료 . . . 4
- 1.3 예시 세션: 래스터 및 벡터 레이어 불러오기 . . . 5

**2 프로젝트 파일 작업** **13**
- 2.1 QGIS 프로젝트 소개 . . . 13
- 2.2 깨진 파일 경로 처리 . . . 16
- 2.3 산출물 생성 . . . 16

**3 QGIS GUI** **17**
- 3.1 메뉴 바 . . . 18
- 3.2 패널 및 툴바 . . . 33
- 3.3 상태 바 . . . 35

**4 탐색기 패널** **39**
- 4.1 탐색기에서 열/실행할 수 있는 리소스들 . . . 41
- 4.2 탐색기 패널 최상위 수준 항목들 . . . 41

**5 QGIS 환경 설정** **51**
- 5.1 옵션 . . . 51
- 5.2 사용자 프로파일 작업 . . . 92
- 5.3 프로젝트 속성 . . . 94
- 5.4 사용자 지정 작업 . . . 111
- 5.5 키보드 단축키 . . . 113
- 5.6 고급 설정으로 QGIS 실행 . . . 115

**6 투영 작업** **123**
- 6.1 투영 지원 개요 . . . 123
- 6.2 레이어 좌표계 . . . 123
- 6.3 프로젝트 좌표계 . . . 125
- 6.4 좌표계 선택기 . . . 127
- 6.5 사용자 정의 좌표계 . . . 128
- 6.6 기준점 (datum) 변환 . . . 130

**7 맵 가시화** **133**
- 7.1 2 차원 맵 뷰 . . . 133

7.2 3 차원 맵 뷰 . . . 160
7.3 표고 단면 뷰 . . . 171

**8 일반 도구 179**
8.1 컨텍스트 도움말 . . . 179
8.2 패널 . . . 179
8.3 외부 프로젝트로부터 레이어를 내포시키기 . . . 197
8.4 대화형 피처 작업 . . . 198
8.5 레이어 속성 저장 및 공유 . . . 206
8.6 사용자 데이터를 문서로 작성 . . . 209
8.7 값을 변수로 저장 . . . 212
8.8 일반 위젯 . . . 212

**9 표현식으로 레벨업 231**
9.1 표현식 . . . 231
9.2 함수 목록 . . . 239

**10 스타일 라이브러리 439**
10.1 스타일 관리자 . . . 439
10.2 심볼 선택기 . . . 449
10.3 라벨 설정 . . . 471
10.4 3 차원 심볼 생성 . . . 493

**11 데이터소스 관리 499**
11.1 데이터 열기 . . . 499
11.2 레이어 생성 . . . 540
11.3 지리참조자 . . . 563
11.4 데이터 포맷 및 필드 탐구 . . . 571

**12 벡터 데이터 작업 581**
12.1 벡터 속성 대화창 . . . 581
12.2 속성 테이블 작업 . . . 671
12.3 편집 작업 . . . 688
12.4 크로스 레이어 데이터 연결 및 편집 작업 . . . 730

**13 래스터 데이터 작업 751**
13.1 래스터 속성 대화창 . . . 751
13.2 래스터 분석 . . . 780

**14 메시 데이터 작업 785**
14.1 메시란 무엇일까요? . . . 785
14.2 지원 포맷 . . . 787
14.3 메시 데이터셋 속성 . . . 788
14.4 Exploring mesh datasets . . . 803
14.5 메시 레이어 편집 . . . 803
14.6 메시 계산기 . . . 813

**15 벡터 타일 작업 817**
15.1 벡터 타일이란? . . . 817
15.2 지원 포맷 . . . 817
15.3 벡터 타일 데이터셋 속성 . . . 819
15.4 QGIS Server Properties . . . 822

**16 점구름 작업 823**
16.1 점구름 소개 . . . 823

16.2 점구름 속성 . . . 824
16.3 가상 점구름 . . . 842

**17 3D 타일 작업** **845**
17.1 3D 타일이란? . . . 845
17.2 3D 타일 속성 . . . 847

**18 맵 조판 작업** **851**
18.1 인쇄 조판기 개요 . . . 851
18.2 조판기 항목 . . . 868
18.3 출력물 생성 작업 . . . 925
18.4 보고서 생성 작업 . . . 937

**19 OGC / ISO 프로토콜 작업** **955**
19.1 WMS/WMTS 클라이언트 . . . 956
19.2 WCS 클라이언트 . . . 967
19.3 WFS 및 WFS-T 클라이언트 . . . 969
19.4 SensorThings . . . 971
19.5 STAC (SpatioTemporal Asset Catalogs) . . . 975
19.6 ArcGIS REST Server . . . 980

**20 GPS 데이터 작업** **981**
20.1 GNSS/GPS 데이터 소개 . . . 981
20.2 실시간 GPS 추적 . . . 984

**21 인증 시스템** **991**
21.1 인증 시스템 개요 . . . 991
21.2 사용자 인증 작업 흐름 . . . 1004

**22 GRASS GIS 통합** **1019**
22.1 예제 데이터셋 . . . 1019
22.2 GRASS 래스터 및 벡터 레이어 불러오기 . . . 1020
22.3 드래그 & 드롭으로 GRASS 로케이션에 데이터 가져오기 . . . 1020
22.4 QGIS 탐색기에서 GRASS 데이터 관리하기 . . . 1020
22.5 GRASS 옵션 . . . 1021
22.6 GRASS 플러그인 시작하기 . . . 1021
22.7 GRASS 맵셋 열기 . . . 1021
22.8 GRASS 로케이션 및 맵셋 . . . 1021
22.9 GRASS 로케이션에 데이터 가져오기 . . . 1022
22.10 GRASS 벡터 데이터 모델 . . . 1025
22.11 새 GRASS 벡터 레이어 생성하기 . . . 1025
22.12 GRASS 벡터 레이어 디지타이즈 및 편집하기 . . . 1026
22.13 GRASS 영역 도구 . . . 1029
22.14 GRASS 툴박스 . . . 1029

**23 QGIS 공간 처리 프레임워크** **1037**
23.1 소개 . . . 1037
23.2 공간 처리 프레임워크 환경 설정 . . . 1039
23.3 툴박스 . . . 1043
23.4 이력 관리자 . . . 1054
23.5 모델 설계자 . . . 1056
23.6 배치 공간 처리 인터페이스 . . . 1070
23.7 콘솔에서 공간 처리 알고리즘 사용 . . . 1074
23.8 명령줄에서 공간 처리 알고리즘 사용 . . . 1082
23.9 새 공간 처리 알고리즘을 파이썬 스크립트로 작성하기 . . . 1084

23.10 공간 처리 알고리즘의 입력물 및 산출물 유형 . . . . . 1092
23.11 외부 응용 프로그램 환경 설정 . . . . . 1096

**24 공간 처리 제공자 및 알고리즘 1107**
24.1 QGIS 알고리즘 제공자 . . . . . 1107
24.2 GDAL 알고리즘 제공자 . . . . . 1712

**25 플러그인 1815**
25.1 QGIS 플러그인 . . . . . 1815
25.2 QGIS 핵심 플러그인 사용하기 . . . . . 1820
25.3 QGIS 파이썬 콘솔 . . . . . 1841

**26 부록 1847**
26.1 부록 C: QGIS 파일 포맷 . . . . . 1847
26.2 부록 D: QGIS R 스크립트 문법 . . . . . 1851
26.3 부록 E: QGIS 응용 프로그램 네트워크 연결 . . . . . 1857

**27 참고 문헌 및 웹사이트 1859**

CHAPTER 1

# 시작하기

이 장에서 QGIS 를 설치하는 법, QGIS 웹페이지에서 예시 데이터를 다운로드하는 법, 그리고 래스터 및 벡터 레이어를 가시화하는 간단한 첫 번째 세션을 수행하는 법을 간단히 살펴보겠습니다.

## 1.1 QGIS 설치

QGIS 프로젝트는 사용자 플랫폼에 따라 QGIS 를 설치할 수 있는 여러 가지 방법을 제공합니다.

### 1.1.1 바이너리로부터 설치

MS 윈도우와 맥 OS 에서 표준 인스톨러를 사용할 수 있습니다. 수많은 GNU/리눅스 배포판을 위한 바이너리 패키지 (rpm 및 deb) 나 설치 관리자에 추가할 수 있는 소프트웨어 저장소도 제공합니다.

사용자의 OS 를 위한 추가 정보 및 지침을 살펴보려면 https://qgis.org/download/ 를 방문하십시오.

### 1.1.2 소스로부터 설치

소스로부터 QGIS 를 빌드해야 하는 경우, 설치 지침을 따라주십시오. QGIS 소스 코드와 함께 배포되는 `INSTALL` 이라는 파일에 있습니다. 온라인 https://github.com/qgis/QGIS/blob/release-3_44/INSTALL.md 에서도 찾아볼 수 있습니다.

사용자가 개발 중인 버전이 아니라 특정 배포판을 빌드하려는 경우, `master` 브랜치를 앞에서 언급한 링크에 있는 (주로 `release-X_Y` 라는 명칭인) 원하는 배포판의 브랜치로 전환해야 합니다. (배포판에 따라 설치 지침이 다를 수 있기 때문입니다.)

### 1.1.3 외부 저장장치에 설치

QGIS 를 (모든 플러그인과 설정들을 포함해서) 플래시 드라이브에 설치할 수 있습니다. 기본 경로사용자 프로파일 을 무시하고 **QSettings** 가 사용자 설정 디렉터리를 사용하도록 강제하는*–profiles-path* 옵션을 정의해주면 됩니다. 추가 정보를 원한다면시스템 설정 을 참조하십시오.

### 1.1.4 예시 데이터 다운로드

이 사용자 지침서는 (`Alaska dataset` 이라고도 불리는) QGIS 샘플 데이터셋을 기반으로 하는 예시를 담고 있습니다. https://github.com/qgis/QGIS-Sample-Data/archive/master.zip 에서 예시 데이터를 다운로드해서 사용자 시스템 상의 편리한 위치에 압축을 푸십시오.

이 알래스카 데이터셋은 사용자 지침서에 있는 예시 및 스크린샷에 쓰이는 모든 GIS 데이터를 포함하고 있습니다. 또 작은 GRASS 데이터베이스를 담고 있기도 합니다. 이 QGIS 샘플 데이터셋에 적용된 투영법은 피트 단위의 알래스카 알베르스 정적원추도법 (Alaska Albers Equal Area) 이며, EPSG 코드는 2964 입니다.

```
PROJCS["Albers Equal Area",
GEOGCS["NAD27",
DATUM["North_American_Datum_1927",
SPHEROID["Clarke 1866",6378206.4,294.978698213898,
AUTHORITY["EPSG","7008"]],
TOWGS84[-3,142,183,0,0,0,0],
AUTHORITY["EPSG","6267"]],
PRIMEM["Greenwich",0,
AUTHORITY["EPSG","8901"]],
UNIT["degree",0.0174532925199433,
AUTHORITY["EPSG","9108"]],
AUTHORITY["EPSG","4267"]],
PROJECTION["Albers_Conic_Equal_Area"],
PARAMETER["standard_parallel_1",55],
PARAMETER["standard_parallel_2",65],
PARAMETER["latitude_of_center",50],
PARAMETER["longitude_of_center",-154],
PARAMETER["false_easting",0],
PARAMETER["false_northing",0],
UNIT["us_survey_feet",0.3048006096012192]]
```

QGIS 를 GRASS 용 그래픽 프론트엔드로 이용하고자 할 경우, 공식 GRASS GIS 웹사이트 https://grass.osgeo.org/download/data/ 에서 (사우스다코타 주의 스피어피시 또는 노스캐롤라이나 주 같은) 샘플 위치들을 모아 놓은 것을 찾아보십시오.

## 1.2 QGIS 실행 및 종료

사용자의 운영체제에서 다른 응용 프로그램을 실행하는 것과 동일하게 QGIS 를 실행할 수 있습니다. 즉 다음과 같은 방법으로 QGIS 를 실행할 수 있다는 뜻입니다:

- 응용 프로그램 (Applications) 메뉴, 시작 (Start) 메뉴, 또는 독 (Dock) 사용
- 사용자의 응용 프로그램 폴더나 바탕화면에 있는 아이콘 더블 클릭
- 기존 QGIS 프로젝트 (`.qgz` 또는 `.qgs`) 파일을 더블 클릭 (이때 해당 프로젝트가 함께 열립니다)
- 명령 프롬프트에서 `qgis` 를 입력 (QGIS 가 사용자의 PATH 에 추가돼 있거나, QGIS 설치 폴더에서 입력한다고 가정)

QGIS 를 종료하려면:

- *Project* → *Exit QGIS* 메뉴 또는 `Ctrl+Q` 단축키 입력
- *QGIS* → *Quit QGIS* 메뉴 또는 `Cmd+Q` 단축키 입력
- 또는 응용 프로그램의 주 인터페이스 우상단에 있는 빨간 가위표를 클릭

## 1.3 예시 세션: 래스터 및 벡터 레이어 불러오기

이제*QGIS 를 설치* 하고*예시 데이터도 다운로드* 했으니, 래스터 및 벡터 레이어를 가시화 하는 첫 예시 세션을 보여드리고자 합니다. 다음 파일들을 사용할 것입니다:

- `landcover` 래스터 레이어 (`qgis_sample_data/raster/landcover.img`)
- `lakes` 벡터 레이어 (`qgis_sample_data/gml/lakes.gml`)

예시 경로의 `qgis_sample_data` 는 데이터셋의 압축을 해제한 경로를 가리킵니다.

1. *QGIS 실행 및 종료* 에서 배운대로 QGIS 를 실행하십시오.
2. 여기에서 작업할 데이터는 알베르스 정적원추도법 (`Albers Equal Area`) 으로 투영되었으므로, 이에 맞춰 프로젝트 좌표계를 설정해줍시다:
   1. QGIS 인터페이스의 우하단에 있는 Select projection 버튼을 클릭하십시오. 프로젝트 속성 대화창의 *CRS* 탭이 열립니다.
   2. Type `2964` in the *Filter* text area.
   3. `NAD27 / Alaska Albers` 좌표계 이름을 가진 행을 선택하십시오.
   4. *OK* 를 클릭하십시오.

   > **참고**
   >
   > "근사치 변환 (ballpark transform)"이라는 메시지가 표시될 수도 있는데, 지금은 그냥 무시하거나 메시지 창을 닫으면 됩니다.

3. QGIS 에 파일들을 불러오십시오:
   1. Open Data Source Manager 아이콘을 클릭하십시오. 데이터소스 관리자 (Data Source Manager) 가 탐색기 (Browser) 모드로 열릴 것입니다.
   2. `qgis_sample_data/raster/` 폴더를 찾으십시오.
   3. ERDAS IMG 파일 *landcover.img* 를 선택한 다음 더블클릭하십시오. 배경에'landcover'레이어가 추가될 것입니다. 데이터소스 관리자 창은 닫히지 않습니다.
   4. 'lakes'데이터를 불러오려면, `qgis_sample_data/gml/` 폴더를 찾은 다음 *lakes.gml* 파일을 QGIS 주 대화창으로 드래그 & 드롭하십시오. (또는 앞에서 설명한 대로 더블클릭하기만 해도 됩니다.)
   5. *Select Items to Add* 대화창이 열리고 파일을 스캔합니다. `.gml` 파일 포맷이 한 번에 하나 이상의 레이어를 저장할 수 있기 때문입니다.
   6. 이 경우 *lakes* 레이어 한 개가 있습니다. 이 레이어를 선택한 다음 *Add Layers* 버튼을 누르십시오.
   7. 레이어가 *Layers* 패널에 추가됩니다.

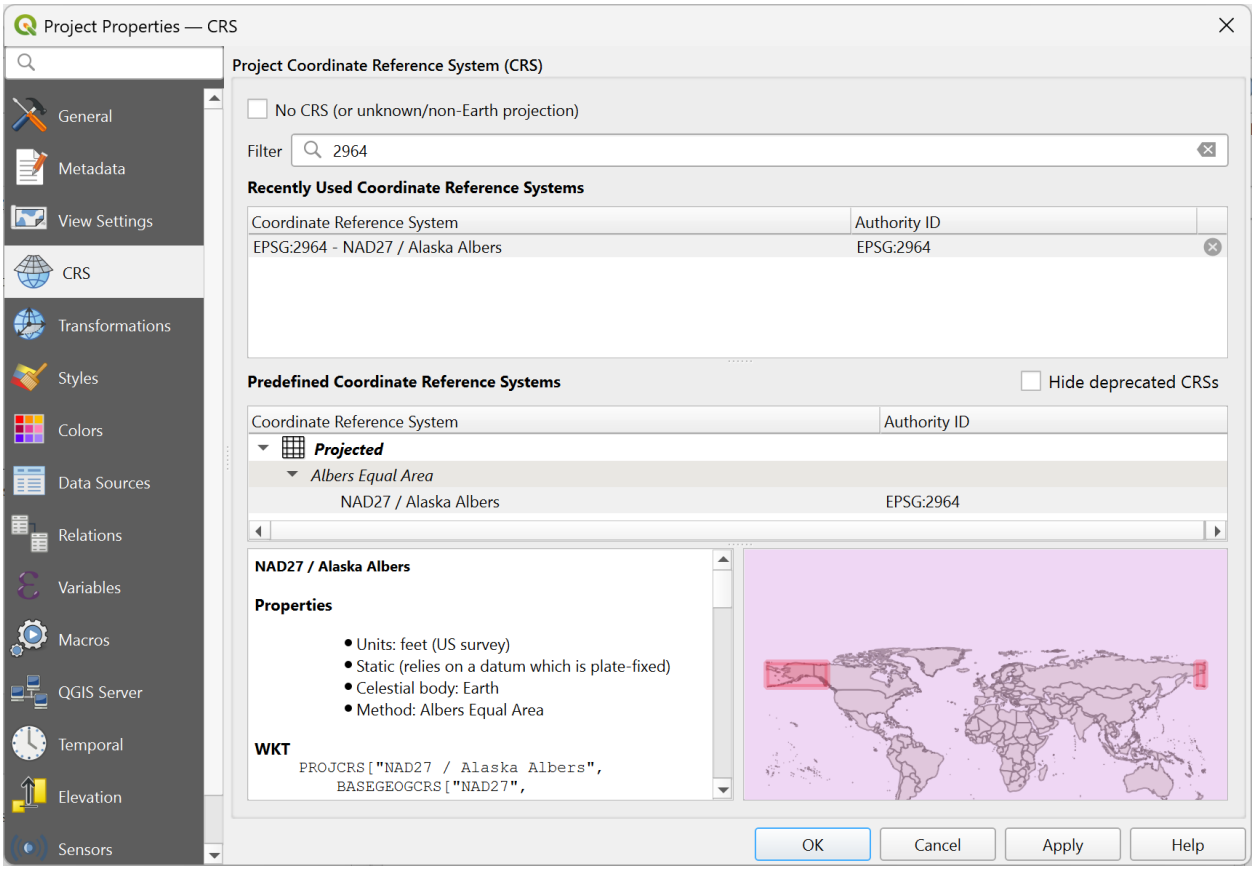



그림 1.1: 해당 데이터의 좌표계를 선택

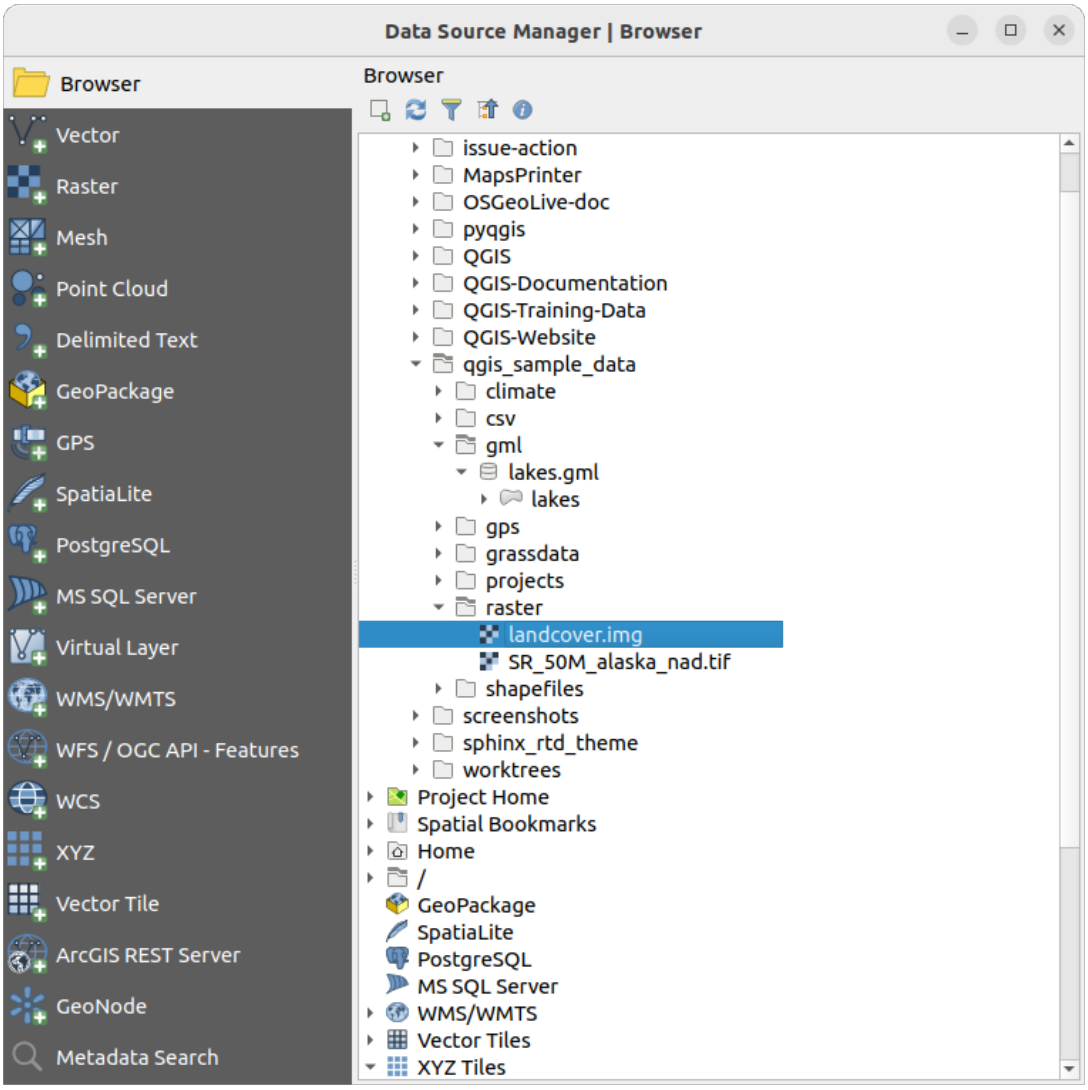



그림 1.2: QGIS 의 새 프로젝트에 데이터 추가하기

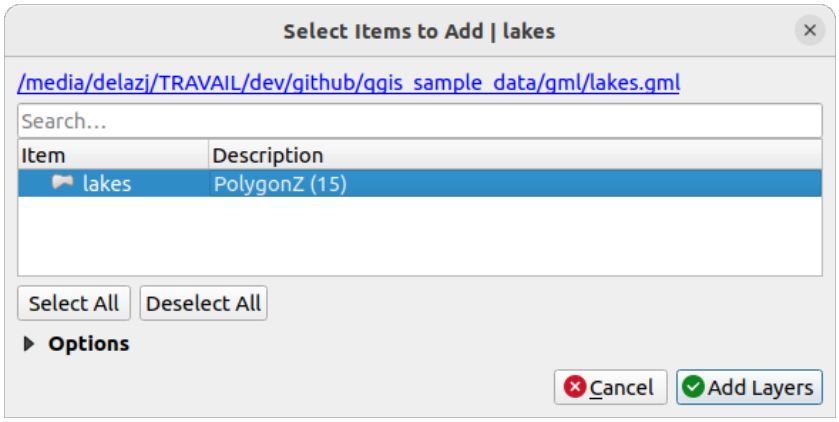



그림 1.3: 파일 안에 있는 레이어를 선택하기

4. 데이터소스 관리자 창을 닫으십시오.

*Layers* 패널을 보면, *lakes* 레이어 옆에 Layer has no coordinate reference system set 아이콘이 표시되어 있을 것입니다. 이를 바로잡아 봅시다.

1. 아이콘을 클릭하십시오. *Coordinate Reference System Selector* 대화창이 열립니다.
2. 앞에서와 마찬가지로, *NAD27 / Alaska Albers* 좌표계 항목을 찾아 선택하십시오.
3. *OK* 를 클릭하십시오.

이제 사용자의 프로젝트 안에서 몇몇 랜덤한 색상으로 표시된 레이어 2 개를 사용할 수 있습니다. 'lakes'레이어에 대해 간단한 사용자 지정 작업을 해보겠습니다.

1. *Navigation* 툴바에 있는 Zoom In 도구를 선택하십시오.
2. 호수가 몇 개 보이도록 마음에 드는 영역으로 조금 확대해보세요.
3. 맵 범례에 있는 `lakes` 레이어를 더블 클릭해서 *Properties* 대화창을 엽니다.
4. 호수의 색상을 변경하려면:
    1. *Symbology* 탭을 클릭합니다.
    2. 채우기 (fill) 색상으로 파란색을 선택하십시오.

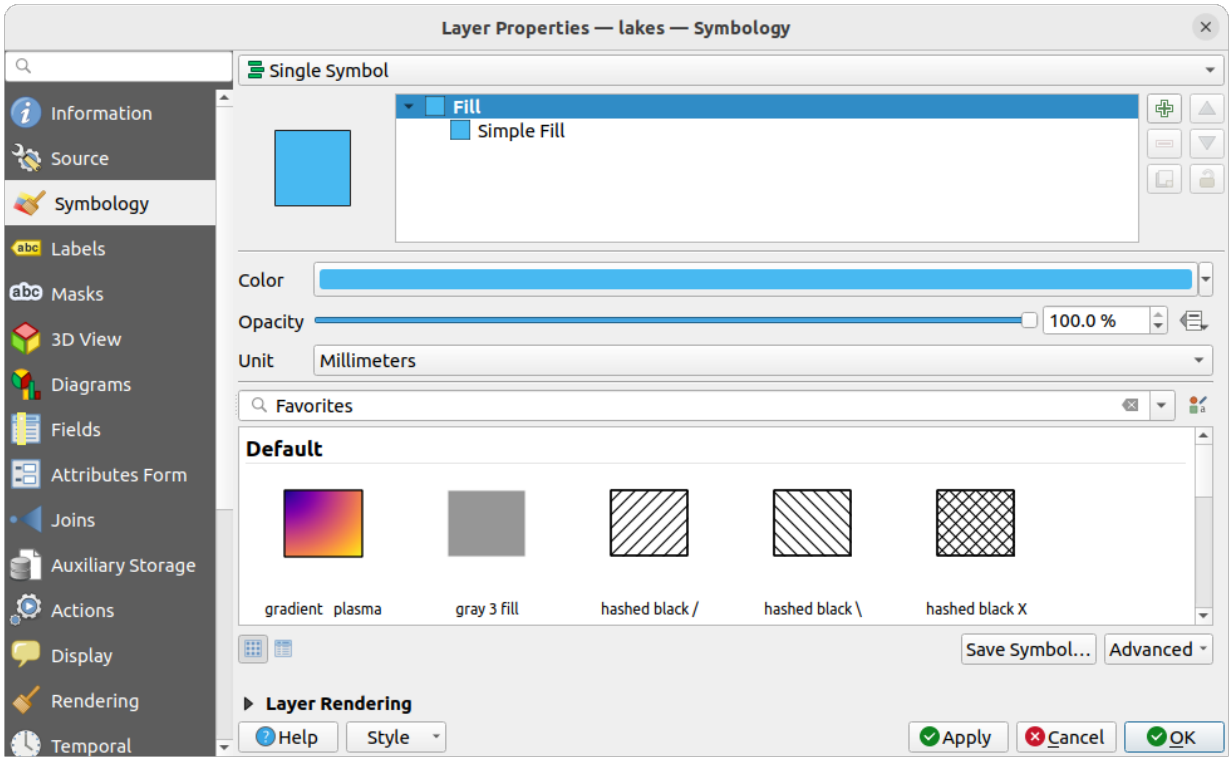



그림 1.4: 호수 색상을 선택하기

    3. *OK* 를 클릭하십시오. 이제 호수들이 맵 캔버스 상에서 파란색으로 보일 것입니다.
5. 호수의 이름을 보이게 하려면:
    1. `lakes` 레이어의 *Properties* 대화창을 다시 여십시오.
    2. *Labels* 탭을 클릭합니다.

3. 드롭다운 메뉴에서 *Single labels* 를 선택해서 라벨 작업을 활성화합니다.
4. *Label with* 목록에서 NAMES 항목을 선택하십시오.

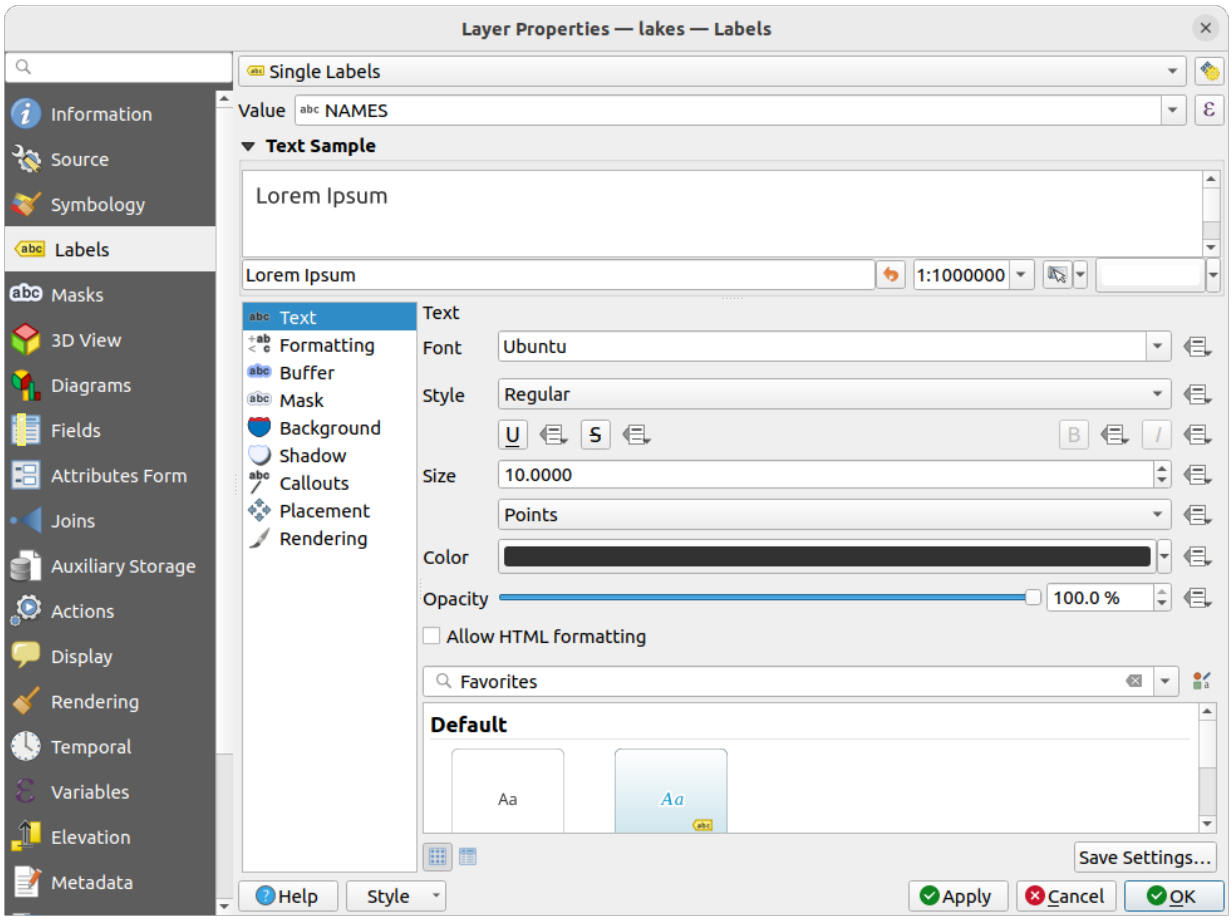



그림 1.5: 호수의 이름을 표시

5. *Apply* 를 클릭하십시오. 호수의 경계선 위로 이름을 불러올 것입니다.

6. 라벨 텍스트 주위에 하얀색의 버퍼를 추가해서 라벨의 가독성을 향상시킬 수 있습니다.
   1. 왼쪽에 있는 목록에서 *Buffer* 탭을 클릭하십시오.
   2. *Draw text buffer* 를 체크하십시오.
   3. 버퍼 크기로 3 을 선택하십시오.
   4. *Apply* 를 클릭하십시오.
   5. 결과가 보기에 좋은지 확인하고, 필요할 경우 버퍼 크기 값을 조정하십시오.
   6. 마지막으로 *OK* 를 클릭해서 *Layer Properties* 대화창을 닫고 변경 사항을 적용합니다.

이제 맵을 조판하기 위한 몇몇 장식을 추가한 다음 QGIS 에서 내보내보겠습니다:

1. *View* → *Decorations* → *Scale Bar* 메뉴를 선택하십시오.
2. 대화창이 열리면, *Enable Scale Bar* 옵션을 체크하십시오.
3. 사용자가 바라는대로 대화창의 옵션을 지정해보십시오.

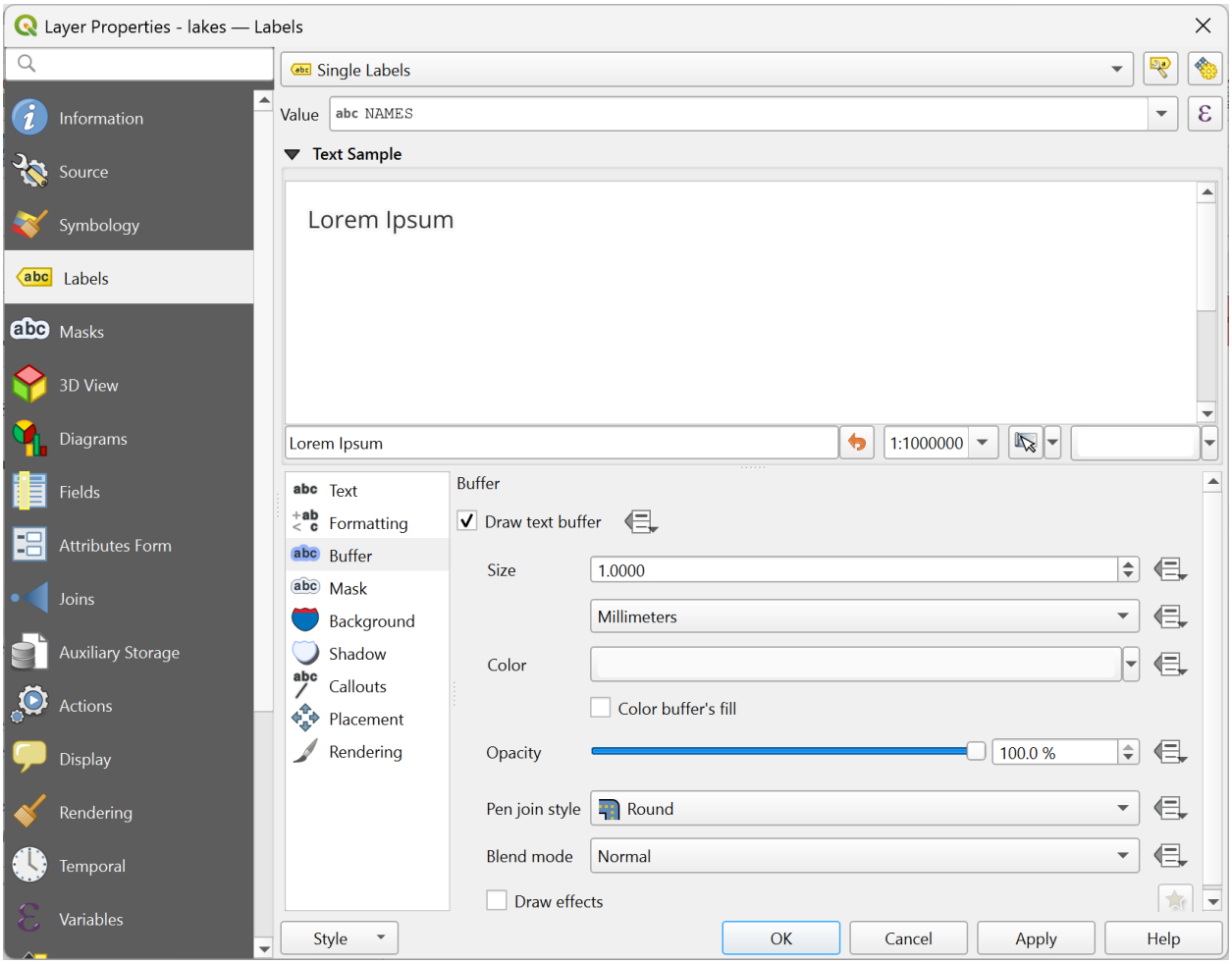



그림 1.6: 라벨 주위에 버퍼 표시하기

4. *Apply* 를 클릭합니다.
5. 같은 방식으로, 장식 (Decorations) 메뉴에서 더 많은 (방위표, 저작권 등등의) 항목들을 사용자 지정 속성으로 맵 캔버스에 추가해보십시오.
6. *Project* → *Import/Export* → *Export Map to Image*…를 클릭하십시오.
7. 열린 대화창에 있는 *Save* 버튼을 클릭하십시오.
8. 파일 위치와 포맷을 선택한 다음 *Save* 를 다시 클릭해서 확정하십시오.
9. 사용자의 변경 사항을 `.qgz` 프로젝트 파일로 저장하려면 *Project* → *Save*…메뉴를 클릭하십시오.

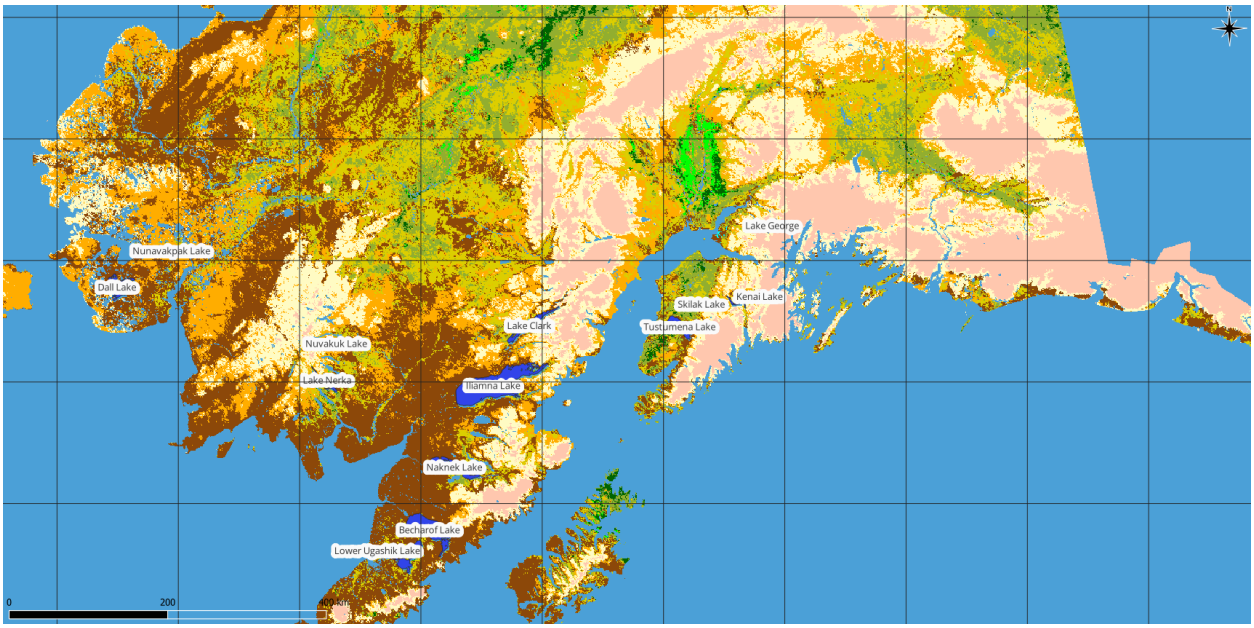



그림 1.7: 장식을 추가해서 내보낸 맵 표시하기

끝났습니다! QGIS 에서 얼마나 쉽게 래스터 및 벡터 레이어를 가시화하고 옵션을 설정한 다음, 사용자의 맵을 다른 소프트웨어에서 사용할 수 있는 이미지 포맷으로 생성할 수 있는지 아시겠지요. 다음 장에서는 사용할 수 있는 기능, 특징 및 설정들과 그 사용법을 배워보겠습니다.

> **참고**
>
> 계속해서 QGIS 를 단계적인 예제들을 통해 배워보고 싶다면 QGIS 교육 교재 를 공부하십시오.

CHAPTER 2

# 프로젝트 파일 작업

## 2.1 QGIS 프로젝트 소개

프로젝트란 사용자의 QGIS 세션 상태를 말합니다. QGIS 는 한 번에 프로젝트 하나에서만 작업 가능합니다. 설정은 프로젝트 별로 관리되며 새 프로젝트의 경우 기본 설정 (옵션 참조) 을 따릅니다. QGIS 는 메뉴 옵션 *Project* → *Save* 또는 *Project* → *Save As*…를 통해 사용자의 작업 공간 (workspace) 을*QGIS* 프로젝트 파일 로 저장할 수 있습니다.

> **참고**
>
> 프로젝트가 작업 중 변경됐다면 창의 제목 막대에 * 심볼이 나타나고, QGIS 가 기본적으로 프로젝트 파일에 변경 사항을 덮어 쓸지 물어볼 것입니다. *Settings* → *Options* → *General* 메뉴 아래에 있는 *Prompt to save project and data source changes when required* 를 체크하면 이 옵션을 활성화할 수 있습니다.

You can load existing projects through *Project* → *Open*…, *Project* → *New from template* or *Project* → *Open Recent* →.

From the *Browser panel* you can *Open Project*, *Extract Symbols*… –opens the style manager to export symbols to XML, add them to the default style, or export as PNG/SVG. You can also access the *File Properties*… dialog. The project file can be expanded to see its layers. The context menu of a layer offers the same actions as elsewhere in the browser.

QGIS 구동 시, *Project Templates* 및 *Recent Projects* 목록이 표시됩니다. 이 목록은 최근 프로젝트 10 개까지의 스크린샷, 프로젝트 이름 및 파일 경로를 포함합니다. *Recent Projects* 목록을 통해 최근 사용한 프로젝트에 쉽게 접근할 수 있습니다. 프로젝트를 열려면 이 목록의 항목 하나를 더블클릭하거나 프로젝트 템플릿을 더블클릭하십시오. 목록의 항목을 오른쪽 클릭하면 *Pin to List*, *Open Directory*…또는 *Remove from List* 메뉴가 나타납니다. 사용자 프로젝트를 *Recent Projects* 목록에서는 볼 수 있지만 파일 관리자에서는 찾을 수 없다면 *Open Directory*…옵션을 사용해서 없어졌거나, 이동되었거나, 재명명되었을 수도 있는 프로젝트를 찾아볼 수 있을 겁니다. *Recent Projects* 목록에서 프로젝트들을 모두 제거하고 싶다면 *Clear List* 메뉴를 사용하면 됩니다. 이전에 프로젝트를 고정 (pin) 시킨 적이 있다면, *Clear List* 메뉴를 클릭했을 때 고정시킨 프로젝트도 제거해야 할지를 물어보는 메시지 창이 뜰 것입니다. 새 프로젝트를 자동으로 생성하려면 레이어를 추가하면 됩니다. 그러면 목록이 사라지고 맵 캔버스가 보일 것입니다.

사용자 세션을 정리하고 새롭게 시작하고 싶다면, 메뉴에서 *Project* ▯ *New* 를 선택하십시오. 기존 프로젝트를 불러온 후 또는 마지막으로 저장한 후에 변경 사항이 발생했을 경우, 이 메뉴는 사용자에게 기존 프로젝트를 저장할 것인지 물어볼 것입니다.

새 프로젝트를 열었다면, 사용자가 프로젝트를 저장하기 전까지 제목 막대에 제목  없는  프로젝트 라고 표시될 것입니다.

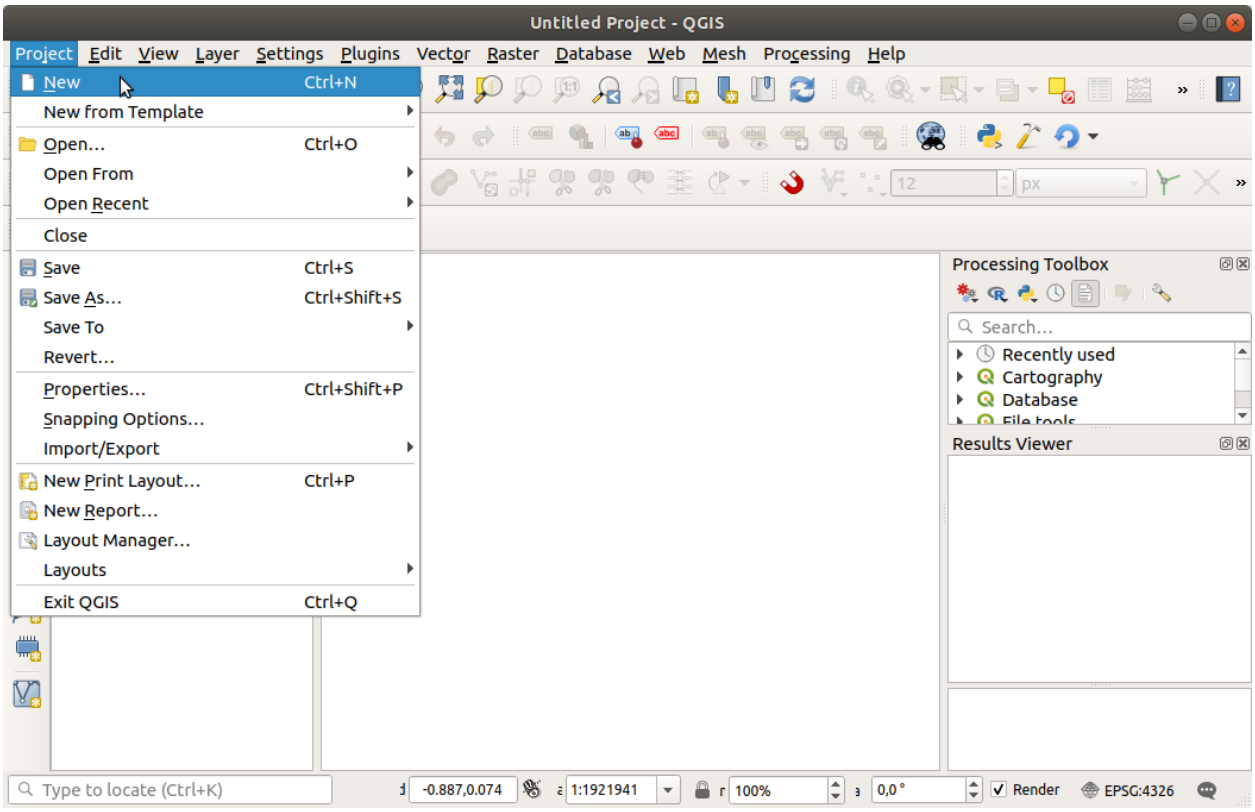



그림 2.1: QGIS 에서 새 프로젝트 시작

프로젝트 파일이 저장하는 정보는 다음을 포함합니다:

- 추가된 레이어
- 레이어 별 쿼리 가능 여부
- 심볼 및 스타일을 포함한 레이어 속성
- 레이어 노트
- 2 차원 및 3 차원 맵 뷰
- 각 맵 뷰의 투영법
- 각 맵의 가장 최근 보았던 범위
- 인쇄 조판
- 설정을 포함한 인쇄 조판의 구성요소들
- 인쇄 조판의 지도 설정
- 디지타이즈 작업 설정
- 테이블 관계

- 프로젝트 매크로
- 프로젝트 기본 스타일
- 플러그인 설정
- 프로젝트 속성의 OWS 설정 탭에서 지정한 QGIS 서버 설정
- DB 관리자에 저장된 쿼리들

프로젝트 파일은 XML 유형 (*QGS/QGZ — QGIS 프로젝트 파일 포맷* 참조) 으로 저장되기 때문에, 사용자가 방법만 안다면 QGIS 외부에서 파일을 편집할 수 있습니다. 프로젝트 파일 포맷은 여러 번 업데이트됐기 때문에, QGIS 과거 버전에서 저장된 프로젝트 파일이 더 이상 제대로 동작하지 않을 수도 있습니다.

> **참고**
>
> QGIS 는 기본적으로 사용자에게 버전이 다르다는 사실을 경고할 것입니다. *Settings — Options* 메뉴의 *General* 탭에서 해당 옵션을 끄거나 켤 수 있습니다. (*Warn when opening a project file saved with an older version of QGIS*)

QGIS 에서 `.qgs` 프로젝트 파일을 저장할 때마다 동일 디렉터리에 프로젝트 파일의 백업을 `.qgs~` 확장자로 생성합니다.

QGIS 프로젝트 파일의 확장자는 `.qgs` 이지만, 기본적으로 `.qgz` 확장자를 가진 압축 포맷으로 저장됩니다. `.qgz` 파일 (ZIP 압축 파일) 은 `.qgs` 파일과 함께보조 데이터 용 연관 SQLite 데이터베이스 (`.qgd`) 를 담고 있습니다. `.qgz` 파일의 압축을 해제하면 이 파일들이 나옵니다.

> **참고**
>
> 보조 저장소 속성 메커니즘은 프로젝트 파일을 압축하는 효용성을 크게 높여줍니다. 압축 파일이 보조 데이터를 담고 있기 때문입니다.

다음 프로젝트 메뉴 항목들을 사용하면 PostgreSQL, GeoPackage 또는 오라클 데이터베이스에 프로젝트를 저장하거나 불러올 수 있습니다:

- *Project — Open from*
- *Project — Save to*

이 메뉴 항목들에는 추가적인 프로젝트 저장 구현 (PostgreSQL, GeoPackage 및 오라클) 목록을 가진 하위 메뉴가 있습니다. 각각을 클릭하면 GeoPackage 연결과 프로젝트를, 또는 PostgreSQL 연결과 스키마와 프로젝트를, 또는 오라클 연결과 소유자와 프로젝트를 선택할 수 있는 대화창이 열릴 것입니다.

QGIS 탐색기 패널에서 PostgreSQL, GeoPackage 또는 오라클에 저장된 프로젝트를 불러올 수도 있습니다. 탐색기 패널에서 해당 항목을 더블클릭하거나 맵 캔버스로 드래그하면 됩니다.

## 2.2 깨진 파일 경로 처리

프로젝트를 열 때, 더 이상 사용할 수 없는 서비스/데이터베이스 또는 재명명되었거나 이동된 파일 때문에 QGIS 가 일부 데이터소스에 접근할 수 없을 수도 있습니다. 이런 경우 QGIS 는 찾을 수 없는 레이어의 참조 정보를 표시하는 *Handle Unavailable Layers* 대화창을 엽니다. 이 창에서:

- *Datasource* 필드를 더블클릭하고, 각 레이어의 경로를 수정한 다음 *Apply changes* 버튼을 클릭하면 됩니다.
- 한 행을 선택하고, *Browse* 버튼을 눌러 정확한 위치를 찾아낸 다음 *Apply changes* 버튼을 클릭하면 됩니다.
- *Auto-Find* 버튼을 눌러 폴더들을 탐색해서 모든 또는 선택한 엉터리 경로 (들) 를 자동으로 수정하도록 할 수 있습니다. 이 경우 탐색하는 데 시간이 걸릴 수도 있습니다. 그 다음 *Apply changes* 버튼을 클릭하십시오.
- *Keep Unavailable Layers* 버튼을 누르면 대화창을 무시하고 엉터리 경로 (들) 를 가진 채로 사용자 프로젝트를 엽니다. 이 경우 사용자 레이어가 *Layers* 패널에 표시는 되지만, *Layers* 패널에서 사용자 레이어 옆에 있는 Unavailable layer! 아이콘 또는 해당 레이어의 컨텍스트 메뉴 가운데 *Repair Data Source*…항목을 사용해서 경로를 고치기 전에는 어떤 데이터도 없습니다.

  *Repair Data Source*…도구로 레이어 경로를 수정하고 나면, QGIS 가 다른 모든 엉터리 경로를 스캔해서 동일한 엉터리 파일 경로를 가진 레이어를 자동으로 고치려 시도합니다.
- *Remove Unavailable Layers* 아이콘을 클릭하면 사용할 수 없는 레이어를 프로젝트에서 삭제할 수 있습니다.

명령줄에서*--skipbadlayers* 옵션을 이용해서 QGIS 를 실행하면 QGIS 실행 시 *Handle Unavailable Layers* 대화창을 건너뛸 수 있습니다.

## 2.3 산출물 생성

사용자의 QGIS 세션에서 산출물을 생성하는 방법이 몇 가지 있습니다. 이미*QGIS 프로젝트 소개* 에서 프로젝트 파일로 저장한다는 한 가지 방법을 설명했습니다. 다음은 산출물 파일을 생성하는 다른 방법들의 예시입니다:

- 이미지 생성하기: *Project* → *Import/Export* → *Export Map to Image*…메뉴는 맵 캔버스를 사용자가 지정한 축척, 해상도, 크기 등으로 렌더링한 이미지 포맷 (PNG, JPG, TIFF…) 으로 생성합니다. *Append georeference information (embedded or via world file)* 옵션을 체크하기만 하면 내보낸 이미지에 지리참조 정보를 포함시킬 수도 있습니다. 더 자세한 내용은맵 뷰 내보내기 를 참조하세요.
- PDF 파일로 내보내기: *Project* → *Import/Export* → *Export Map to PDF*…메뉴는 맵 캔버스를 사용자 지정 축척, 해상도 그리고 여러 고급 설정 (단순화, 지리참조 등) 으로 렌더링한 PDF 파일을 생성합니다. 더 자세한 내용은맵 뷰 내보내기 를 참조하세요.
- DXF 파일로 내보내기: *Project* → *Import/Export* → *Export Project to DXF*…메뉴는 사용자가 DXF 파일로 내보내고자 하는'심볼 모드', '심볼 축척'그리고 벡터 레이어를 정의할 수 있는 대화창을 엽니다. '심볼 모드'를 통해 QGIS 심볼을 원본에 충실하게 내보낼 수 있습니다. (새 *DXF* 파일 생성하기 참조)
- 맵 디자인: *Project* → *New Print Layout*…메뉴는 사용자가 현재 맵 캔버스를 조판하고 인쇄할 수 있는 대화창을 엽니다. (맵 조판 작업 참조)

CHAPTER 3

# QGIS GUI

QGIS GUI(Graphical User Interface) 는 다음 그림에서 보이는 대로입니다. (노란색 원 안의 1 부터 5 까지의 숫자는 QGIS GUI 의 주요 구성 요소를 나타냅니다. 차례대로 설명하겠습니다.)

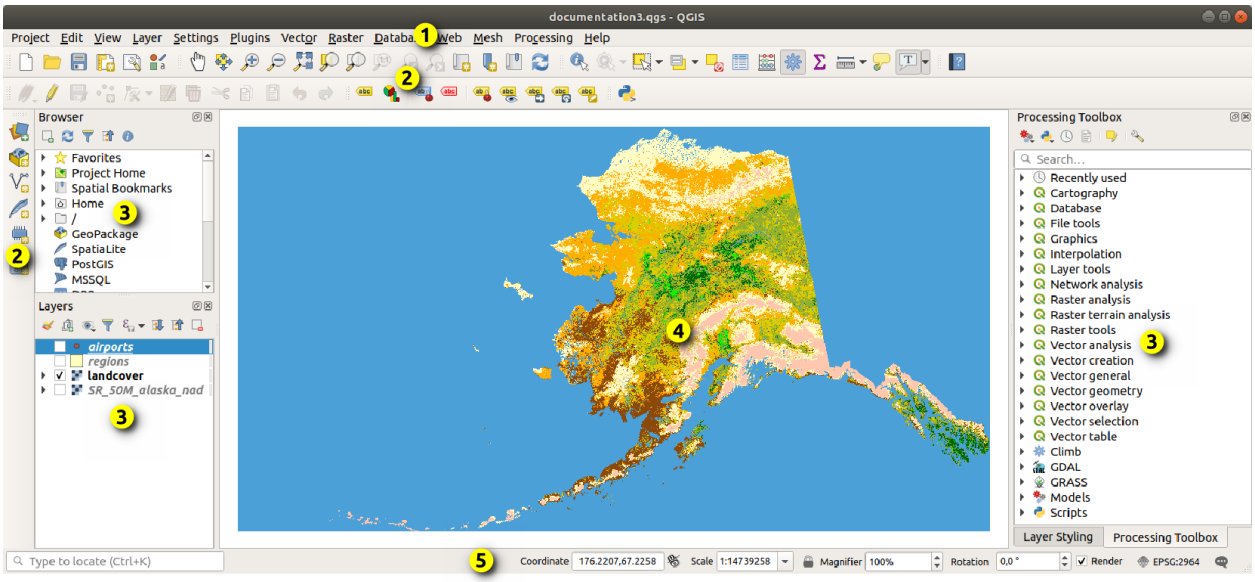



그림 3.1: 알래스카 예시 데이터를 불러온 QGIS GUI

> **참고**
>
> 사용자의 운영체제 및 창 관리자에 따라 창 양식 (제목 바 등등) 이 다르게 나타날 수도 있습니다.

주요 QGIS GUI(그림 3.1) 를 다음 5 개의 구성 요소/구성 요소 유형으로 구분할 수 있습니다:

1. 메뉴 바

2. 툴바
3. 패널
4. 맵 뷰
5. 상태 바

이들에 대한 자세한 설명을 보려면 아래로 스크롤하세요.

## 3.1 메뉴 바

메뉴 바는 표준적인 위계 메뉴를 이용해서 다양한 QGIS 기능에 접근할 수 있도록 해줍니다. 메뉴, 메뉴 옵션, 관련 아이콘과 키보드 단축키를 다음에 정리해 놓았습니다. 키보드 단축키는 재설정할 수 있습니다. (*Settings → Keyboard Shortcuts*)

대부분의 메뉴에 그에 대응하는 툴바 아이콘이 존재하고 대부분의 툴바 아이콘에도 대응하는 메뉴가 있지만, 메뉴는 툴바와 동일하게 구성되어 있지 않습니다. 툴바에 있는 각 메뉴 옵션의 위치를 표로 정리해놓았습니다. 일부 메뉴 옵션은 대응하는 플러그인을 불러왔을 경우에만 나타납니다. 도구 및 툴바에 대해 더 많은 정보를 알고 싶다면, 툴바 를 참조하십시오.

> **참고**
>
> QGIS 는 크로스 플랫폼 응용 프로그램으로, 운영체제 별로 동일한 도구를 제공하긴 하지만 운영체제에 따라 서로 다른 메뉴에 위치할 수도 있습니다. 다음 목록은 이런 차이가 있을 경우 가장 일반적인 위치를 표시합니다.

### 3.1.1 프로젝트

*Project* 메뉴는프로젝트 파일 에 대한 액세스/엑시트 포인트 (access and exit points) 를 제공합니다. 다음과 같은 도구들을 사용할 수 있습니다:

- *New* : 프로젝트 파일을 처음부터, 또는 다른 프로젝트 파일을 템플릿 삼아 생성합니다. (템플릿 설정은프로젝트 파일 옵션 을 참조하세요.)
- *Open…* : 파일, GeoPackage, PostgreSQL 또는 오라클 데이터베이스로부터 프로젝트를 불러옵니다.
- *Close* : 프로젝트를 닫거나 가장 최근 저장한 상태로 되돌립니다.
- *Save* : 프로젝트를 `.qgs` 또는 `.qgz` 파일 포맷으로 저장하거나, GeoPackage, PostgreSQL 또는 오라클 데이터베이스 안에 저장합니다.
- 맵 캔버스를 여러 포맷으로 내보내거나, 인쇄 조판기 를 이용해서 좀 더 복잡한 산출물로 내보냅니다.
- 프로젝트 속성 및 도형 편집 작업에 필요한 스냅 옵션을 설정합니다.

표 3.1: 프로젝트 메뉴 항목

| 메뉴 옵션 | 단축키 | 툴바 | 참조 |
|---|---|---|---|
| *New* | `Ctrl+N` | *Project* | *QGIS* 프로젝트 소개 |
| *New from template* → | | | *QGIS* 프로젝트 소개 |
| *Open…* | `Ctrl+O` | *Project* | *QGIS* 프로젝트 소개 |
| *Open from* → *GeoPackage…* | | | *QGIS* 프로젝트 소개 |

다음 페이지에 계속

표 3.1 – 이전 페이지에서 계속

| 메뉴 옵션 | 단축키 | 툴바 | 참조 |
|---|---|---|---|
| *PostgreSQL…* | | | *QGIS* 프로젝트 소개 |
| *Oracle…* | | | *QGIS* 프로젝트 소개 |
| *Open Recent* | Alt+J + R | | *QGIS* 프로젝트 소개 |
| *Clear List* | | | *QGIS* 프로젝트 소개 |
| *Close* | | | *QGIS* 프로젝트 소개 |
| *Save* | Ctrl+S | *Project* | *QGIS* 프로젝트 소개 |
| *Save As…* | Ctrl+Shift+S | *Project* | *QGIS* 프로젝트 소개 |
| *Save to* | | | *QGIS* 프로젝트 소개 |
| *Templates…* | | | *QGIS* 프로젝트 소개 |
| *GeoPackage…* | | | *QGIS* 프로젝트 소개 |
| *PostgreSQL…* | | | *QGIS* 프로젝트 소개 |
| *Revert…* | | | |
| *Properties…* | Ctrl+Shift+P | | 프로젝트 속성 |
| *Snapping Options…* | | | 스냅 작업 및 디지타이즈 작업 옵션 |
| *Import/Export* | | | |
| *Export Map to Image…* | | | 맵 뷰 내보내기 |
| *Export Map to PDF…* | | | 맵 뷰 내보내기 |
| *Export Project to DXF…* | | | 새 *DXF* 파일 생성하기 |
| *Import Layers from DWG/DXF…* | | | *DXF* 또는 *DWG* 파일 가져오기 |
| *New Print Layout…* | Ctrl+P | *Project* | 맵 조판 작업 |
| *New Report…* | | | 보고서 생성 작업 |
| *Layout Manager…* | | *Project* | 맵 조판 작업 |
| *Layouts* | | | 맵 조판 작업 |
| *Models* | | | 모델 설계자 |
| *Exit QGIS* | Ctrl+Q | | |

맥 OS 에서는 *Exit QGIS* 명령어가 *QGIS* *Quit QGIS* (Cmd+Q) 로 동작합니다.

### 3.1.2 편집

*Edit* 메뉴는 레이어 속성이나 도형을 편집하는 데 필요한 내장 도구 대부분을 제공합니다. *Edit* 메뉴 옵션들을 활성화시키려면 Toggle editing 아이콘을 클릭해서 편집 작업 모드로 전환해야 합니다. (자세한 내용은편집 작업 을 참조하세요.)

표 3.2: 편집 메뉴 항목

| 메뉴 옵션 | 단축키 | 툴바 | 참조 |
|---|---|---|---|
| *Undo* | Ctrl+Z | *Digitizing* | 실행 취소 및 재실행 |
| *Redo* | Ctrl+Shift+Z | *Digitizing* | 실행 취소 및 재실행 |
| *Cut Features* | Ctrl+X | *Digitizing* | 피처 잘라내기, 복사 및 붙여넣기 |

다음 페이지에 계속

표 3.2 – 이전 페이지에서 계속

| 메뉴 옵션 | 단축키 | 툴바 | 참조 |
|---|---|---|---|
| *Copy Features* | Ctrl+C | *Digitizing* | 피처 잘라내기, 복사 및 붙여넣기 |
| *Paste Features* | Ctrl+V | *Digitizing* | 피처 잘라내기, 복사 및 붙여넣기 |
| *Paste Features as* | | | 속성 테이블 작업 |
| *New Vector Layer…* | | | 속성 테이블 작업 |
| *Temporary Scratch Layer…* | Ctrl+Alt+V | | 속성 테이블 작업 |
| *Delete Selected* | | *Digitizing* | 선택한 피처 삭제하기 |
| *Select* | | | 피처 선택 |
| *Select Feature(s)* | | *Selection* | 피처 선택 |
| *Select Features by Polygon* | | *Selection* | 피처 선택 |
| *Select Features by Freehand* | | *Selection* | 피처 선택 |
| *Select Features by Radius* | | *Selection* | 피처 선택 |
| *Select Features by Value…* | F3 | *Selection* | 피처 선택 |
| *Select Features by Expression…* | Ctrl+F3 | *Selection* | 피처 선택 |
| *Deselect Features from All Layers* | Ctrl+Alt+A | *Selection* | 피처 선택 |
| *Deselect Features from the Current Active Layer* | Ctrl+Shift+A | *Selection* | 피처 선택 |
| *Reselect Features* | | | 피처 선택 |
| *Select All Features* | Ctrl+A | *Selection* | 피처 선택 |
| *Invert Feature Selection* | | *Selection* | 피처 선택 |
| *Add Record* | Ctrl+. | *Digitizing* | |
| *Add Point Feature* | Ctrl+. | *Digitizing* | 피처 추가하기 |
| *Add Line Feature* | Ctrl+. | *Digitizing* | 피처 추가하기 |
| *Add Polygon Feature* | Ctrl+. | *Digitizing* | 피처 추가하기 |
| *Add Annotation* | | | 주석 도구 |
| *Form Annotation* | | *Annotations* | 주석 도구 |
| *HTML Annotation* | | *Annotations* | 주석 도구 |
| *Edit Attributes* | | | |
| *Modify Attributes of Selected Features* | | *Digitizing* | 속성값 편집 |
| *Merge Attributes of Selected Features* | | *Advanced Digitizing* | 선택한 피처의 속성 병합 |
| *Edit Geometry* | | | |
| *Move Feature(s)* | | *Advanced Digitizing* | 피처 (들) 이동 |
| *Copy and Move Feature(s)* | | *Advanced Digitizing* | 피처 (들) 이동 |

다음 페이지에 계속

표 3.2 – 이전 페이지에서 계속

| 메뉴 옵션 | 단축키 | 툴바 | 참조 |
|---|---|---|---|
| *Rotate Feature(s)* | | *Advanced Digitizing* | 피처 (들) 회전 |
| *Scale Feature(s)* | | *Advanced Digitizing* | 피처 크기 조정 |
| *Simplify Feature* | | *Advanced Digitizing* | 피처 단순화 |
| *Add Ring* | | *Advanced Digitizing* | 고리 추가 |
| *Add Part* | | *Advanced Digitizing* | 부분 추가 |
| *Fill Ring* | | *Advanced Digitizing* | 고리 채우기 |
| *Delete Ring* | | *Advanced Digitizing* | 고리 삭제 |
| *Delete Part* | | *Advanced Digitizing* | 부분 삭제 |
| *Reshape Features* | | *Advanced Digitizing* | 피처 재형성 |
| *Offset Curve* | | *Advanced Digitizing* | 곡선 오프셋 |
| *Split Features* | | *Advanced Digitizing* | 피처 분할 |
| *Split Parts* | | *Advanced Digitizing* | 부분 분할 |
| *Merge Selected Features* | | *Advanced Digitizing* | 선택한 피처 병합 |
| *Vertex Tool (All Layers)* | | *Digitizing* | 꼭짓점 도구 |
| *Vertex Tool (Current Layer)* | | *Digitizing* | 꼭짓점 도구 |
| *Reverse Line* | | *Advanced Digitizing* | 라인 반전 |
| *Trim/extend Feature* | | *Advanced Digitizing* | 피처 다듬기/연장하기 |
| *Rotate Point Symbols* | | *Advanced Digitizing* | 포인트 심볼 회전 |
| *Offset Point Symbols* | | *Advanced Digitizing* | 포인트 심볼 오프셋 |

선택한 레이어 도형 유형 — 포인트, 폴리라인, 폴리곤 — 을 따르는 도구들은 선택한 유형에 따라 활성화됩니다:

표 3.3: “피처 이동”도형 기반 아이콘

| 메뉴 옵션 | 포인트 | 폴리라인 | 폴리곤 |
|---|---|---|---|
| *Move Feature(s)* |  |  |  |
| *Copy and Move Feature(s)* |  |  |  |

### 3.1.3 뷰

맵은 맵 뷰로 렌더링됩니다. *View* 도구를 이용해서 이런 뷰들과 상호작용할 수 있습니다. 예를 들면 다음과 같은 작업을 할 수 있습니다:

- 주 맵 캔버스 옆에 새로운 2D 또는 3D 맵 뷰를 생성할 수 있습니다.
- 어느 곳으로든확대/축소 및 이동 할 수 있습니다.
- 화면에 표시된 피처의 속성 또는 도형을 쿼리할 수 있습니다.
- 미리보기 모드, 주석 또는 장식으로 맵 뷰를 향상시킬 수 있습니다.
- 모든 패널 또는 툴바에 접근할 수 있습니다.

또 이 메뉴의 다음과 같은 옵션을 통해 QGIS 인터페이스 자체를 재구성할 수도 있습니다:

- *Toggle Full Screen Mode*: 제목 바를 숨기면서 전체 화면으로 확대합니다.
- *Toggle Panel Visibility*: 활성화된패널 을 켜고 끕니다 — 피처 디지타이즈 작업 시 (캔버스 최대 가시화에) 유용합니다. 또 QGIS 의 주 캔버스를 사용한 프리젠테이션 (프로젝터 용/녹화 용) 용도로도 유용합니다.
- *Toggle Map Only*: hides panels, toolbars, menus and status bar and only shows the map canvas. Combined with the full screen option, it makes your screen display only the map.

표 3.4: 뷰 메뉴 항목

| 메뉴 옵션 | 단축키 | 툴바 | 참조 |
|---|---|---|---|
|  *New Map View* | `Ctrl+M` | *Map Navigation* | *2* 차원 맵 뷰 |
| *3D Map Views* | | | *3* 차원 맵 뷰 |
|  *New 3D Map View* | `Ctrl+Alt+M` | *Map Navigation* | *3* 차원 맵 뷰 |
| *Manage 3D Map Views* | | | *3* 차원 맵 뷰 |
|  *Pan Map* | | *Map Navigation* | 맵 뷰 탐구하기 |
|  *Pan Map to Selection* | | *Map Navigation* | 맵 뷰 탐구하기 |
|  *Zoom In* | `Ctrl+Alt++` | *Map Navigation* | 맵 뷰 탐구하기 |
|  *Zoom Out* | `Ctrl+Alt+-` | *Map Navigation* | 맵 뷰 탐구하기 |
| *Data Filtering* | | | *2* 차원 맵 뷰 |

다음 페이지에 계속

표 3.4 – 이전 페이지에서 계속

| 메뉴 옵션 | 단축키 | 툴바 | 참조 |
|---|---|---|---|
| *Temporal Controller* | | *Map Navigation* | 시계열 제어기 패널 |
| *Elevation Controller* | | | *The Elevation Controller panel* |
| *Identify Features* | `Ctrl+Shift+I` | *Attributes* | 피처 식별 |
| *Measure* | | *Attributes* | 측정 |
| *Measure Line* | `Ctrl+Shift+M` | *Attributes* | 측정 |
| *Measure Area* | `Ctrl+Shift+J` | *Attributes* | 측정 |
| *Measure Bearing* | | *Attributes* | 측정 |
| *Measure Angle* | | *Attributes* | 측정 |
| *Statistical Summary* | | *Attributes* | 통계 요약 패널 |
| *Elevation Profile* | | | 표고 단면 뷰 |
| *Zoom Full* | `Ctrl+Shift+F` | *Map Navigation* | 맵 뷰 탐구하기 |
| *Zoom To Selection* | `Ctrl+J` | *Map Navigation* | 맵 뷰 탐구하기 |
| *Zoom To Layer(s)* | | *Map Navigation* | 맵 뷰 탐구하기 |
| *Zoom To Native Resolution (100%)* | | *Map Navigation* | 맵 뷰 탐구하기 |
| *Zoom Last* | | *Map Navigation* | 맵 뷰 탐구하기 |
| *Zoom Next* | | *Map Navigation* | 맵 뷰 탐구하기 |
| *Decorations* | `Alt+V` + `D` | | 맵 장식하기 |
| *Grid…* | | | 그리드 |
| *Scale Bar…* | | | 축척 막대 |
| *Image…* | | | 이미지 장식 |
| *North Arrow…* | | | 방위표 |
| *Title Label…* | | | 제목 라벨 |
| *Copyright Label…* | | | 저작권 라벨 |
| *Layout Extents…* | | | 조판 범위 |
| *Preview mode* | | | |
| *Normal* | | | |
| *Simulate Monochrome* | | | |
| *Simulate Achromatopsia Color Blindness (Grayscale)* | | | |
| *Simulate Protanopia Color Blindness (No Red)* | | | |

다음 페이지에 계속

표 3.4 – 이전 페이지에서 계속

| 메뉴 옵션 | 단축키 | 툴바 | 참조 |
|---|---|---|---|
| *Simulate Deuteranopia Color Blindness (No Green)* | | | |
| *Simulate Tritanopia Color Blindness (No Blue)* | | | |
| *Show Map Tips* | | *Attributes* | 표시 속성 |
| *New Spatial Bookmark…* | Ctrl+B | *Map Navigation* | 맵 상에서의 범위 북마크 작업 |
| *Show Spatial Bookmarks* | Ctrl+Shift+B | *Map Navigation* | 맵 상에서의 범위 북마크 작업 |
| *Show Spatial Bookmark Manager* | | | 맵 상에서의 범위 북마크 작업 |
| *Refresh* | F5 | *Map Navigation* | |
| *Layer Visibility* | | | 레이어 패널 |
| *Show All Layers* | Ctrl+Shift+U | | 레이어 패널 |
| *Hide All Layers* | Ctrl+Shift+H | | 레이어 패널 |
| *Show Selected Layers* | | | 레이어 패널 |
| *Hide Selected Layers* | | | 레이어 패널 |
| *Toggle Selected Layers* | | | 레이어 패널 |
| *Toggle Selected Layers Independently* | | | 레이어 패널 |
| *Hide Deselected Layers* | | | 레이어 패널 |
| *Panels* | | | 패널 및 툴바 |
| *Toolbars* | | | 패널 및 툴바 |
| *Toggle Full Screen Mode* | F11 | | |
| *Toggle Panel Visibility* | Ctrl+Tab | | |
| *Toggle Map Only* | Ctrl+Shift+Tab | | |

리눅스 KDE 에서는 *Panels*, *Toolbars* and *Toggle Full Screen Mode* 메뉴가 *Settings* 메뉴 아래 있습니다.

### 3.1.4 레이어

*Layer* 메뉴는 새 데이터소스를생성 하고, 생성한 데이터소스를 프로젝트에추가 하거나, 생성한 데이터소스에변경사항을 저장 할 수 있는 다양한 도구 모음을 제공합니다. 이 데이터소스를 이용해서 다음과 같은 작업을 할 수 있습니다:

- *Duplicate*: 레이어를 복제해서 사용자가 명칭, 스타일 (심볼, 라벨 등등), 결합 등을 수정할 수 있는 복사본을 생성합니다. 복사본은 원본과 동일한 데이터소스를 이용합니다.
- *Copy* 및 *Paste*: 한 프로젝트에서 레이어 또는 그룹을 복사해서 또다른 프로젝트의 새 인스턴스로 붙여넣습니다. 새 인스턴스의 피처 및 속성을 독립적으로 수정할 수 있습니다. *Duplicate* 과 마찬가지로, 이 두 레이어는 계속 동일한 데이터소스 기반입니다.
- *Embed Layers and Groups…* : 다른 프로젝트에서 레이어 또는 그룹을 읽기 전용으로 복제해옵니다. 복제한 인스턴스는 수정할 수 없습니다. (외부 프로젝트로부터 레이어를 내포시키기 참조)

*Layer* 메뉴는 (스타일, 축척, 좌표계 등의) 레이어 속성을 설정, 복사, 붙여넣기할 수 있는 도구도 가지고 있습니다.

표 3.5: 레이어 메뉴 항목

| 메뉴 옵션 | 단축키 | 툴바 | 참조 |
|---|---|---|---|
| *Data Source Manager* | Ctrl+L | *Data Source Manager* | 데이터 열기 |
| *Create Layer* | | | 새 벡터 레이어 생성 |
| *New GeoPackage Layer…* | Ctrl+Shift+N | *Data Source Manager* | 새 *GeoPackage* 레이어 생성하기 |
| *New Shapefile Layer…* | | *Data Source Manager* | 새 셰이프파일 레이어 생성하기 |
| *New SpatiaLite Layer…* | | *Data Source Manager* | 새 *SpatiaLite* 레이어 생성하기 |
| *New Temporary Scratch Layer…* | | *Data Source Manager* | 새 임시 스크래치 레이어 생성하기 |
| *New Mesh Layer…* | | *Data Source Manager* | 새 메시 레이어 생성하기 |
| *New GPX Layer…* | | *Data Source Manager* | 새 *GPX* 레이어 생성하기 |
| *New Virtual Layer…* | | *Data Source Manager* | 가상 레이어 생성 |
| *Add Layer* | | | 데이터 열기 |
| *Add Vector Layer……* | Ctrl+Shift+V | *Manage Layers* | 파일에서 레이어 불러오기 |
| *Add Raster Layer…* | Ctrl+Shift+R | *Manage Layers* | 파일에서 레이어 불러오기 |
| *Add Mesh Layer…* | | *Manage Layers* | 메시 레이어 불러오기 |
| *Add Delimited Text Layer…* | Ctrl+Shift+T | *Manage Layers* | 구분 텍스트 파일 가져오기 |
| *Add PostgreSQL Layer…* | Ctrl+Shift+D | *Manage Layers* | 데이터베이스 관련 도구 |
| *Add SpatiaLite Layer…* | Ctrl+Shift+L | *Manage Layers* | *SpatiaLite* 레이어 |
| *Add MS SQL Server Layer…* | | *Manage Layers* | 데이터베이스 관련 도구 |
| *Add Oracle Spatial Layer…* | | *Manage Layers* | 데이터베이스 관련 도구 |
| *Add SAP HANA Spatial Layer…* | | *Manage Layers* | 데이터베이스 관련 도구 |
| *Add/Edit Virtual Layer…* | | *Manage Layers* | 가상 레이어 생성 |
| *Add WMS/WMTS Layer…* | Ctrl+Shift+W | *Manage Layers* | *WMS/WMTS* 레이어 불러오기 |
| *Add XYZ Layer…* | | | *XYZ* 타일 서비스 사용하기 |
| *Add WCS Layer…* | | *Manage Layers* | *WCS* 클라이언트 |
| *Add WFS / OGC API - Features Layer…* | | *Manage Layers* | *WFS* 및 *WFS-T* 클라이언트 |
| *Add ArcGIS REST Server Layer…* | | *Manage Layers* | *ArcGIS REST* 서버 이용하기 |
| *Add Vector Tile Layer…* | | | 벡터 타일 서비스 사용하기 |

다음 페이지에 계속

표 3.5 – 이전 페이지에서 계속

| 메뉴 옵션 | 단축키 | 툴바 | 참조 |
|---|---|---|---|
| *Add Point Cloud Layer…* | | | 점구름 작업 |
| *Add GPX Layer…* | | | *GNSS/GPS* 데이터 소개 |
| *Embed Layers and Groups…* | | | 외부 프로젝트로부터 레이어를 내포시키기 |
| *Add from Layer Definition File…* | | | 레이어 정의 파일 |
| *Georeferencer…* | | | 지리참조자 |
| *Copy Style* | | | 레이어 속성 저장 및 공유 |
| *Paste Style* | | | 레이어 속성 저장 및 공유 |
| *Copy Layer* | | | |
| *Paste Layer/Group* | | | |
| *Open Attribute Table* | F6 | *Attributes* | 속성 테이블 작업 |
| *Filter Attribute Table* | | | 속성 테이블 작업 |
| *Open Attribute Table (Selected Features)* | Shift+F6 | *Attributes* | 속성 테이블 작업 |
| *Open Attribute Table (Visible Features)* | Ctrl+F6 | *Attributes* | 속성 테이블 작업 |
| *Open Attribute Table (Edited and New Features)* | | *Attributes* | 속성 테이블 작업 |
| *Toggle Editing* | | *Digitizing* | 기존 레이어 디지타이즈 작업 |
| *Save Layer Edits* | | *Digitizing* | 편집한 레이어 저장하기 |
| *Current Edits* | | *Digitizing* | 편집한 레이어 저장하기 |
| *Save for Selected Layer(s)* | | *Digitizing* | 편집한 레이어 저장하기 |
| *Rollback for Selected Layer(s)* | | *Digitizing* | 편집한 레이어 저장하기 |
| *Cancel for Selected Layer(s)* | | *Digitizing* | 편집한 레이어 저장하기 |
| *Save for all Layers* | | *Digitizing* | 편집한 레이어 저장하기 |
| *Rollback for all Layers* | | *Digitizing* | 편집한 레이어 저장하기 |
| *Cancel for all Layers* | | *Digitizing* | 편집한 레이어 저장하기 |
| *Save As…* | | | 기존 레이어로부터 레이어 생성 |
| *Save As Layer Definition File…* | | | 레이어 정의 파일 |
| *Remove Layer/Group* | Ctrl+D | | |
| *Duplicate Layer(s)* | | | |
| *Set Scale Visibility of Layer(s)* | | | 가시성 축척 선택기 |
| *Set CRS of Layer(s)* | Ctrl+Shift+C | | 레이어 좌표계 |
| *Set Project CRS from Layer* | | | 프로젝트 좌표계 |

다음 페이지에 계속

표 3.5 – 이전 페이지에서 계속

| 메뉴 옵션 | 단축키 | 툴바 | 참조 |
|---|---|---|---|
| *Layer Properties…* | | | • *vector*<br>• *raster*<br>• 메시<br>• 점구름<br>• 벡터 타일 |
| *Filter…* | Ctrl+F | | 쿼리 작성기 |
| *Labeling* | | | 라벨 속성 |
| *Show in Overview* | | | 오버뷰 패널 |
| *Show All in Overview* | | | 오버뷰 패널 |
| *Hide All from Overview* | | | 오버뷰 패널 |

### 3.1.5 설정

표 3.6: 설정 메뉴 항목

| 메뉴 옵션 | 참조 |
|---|---|
| *User Profiles* | 사용자 프로파일 작업 |
| *default* | 사용자 프로파일 작업 |
| *Open Active Profile Folder* | 사용자 프로파일 작업 |
| *New Profile…* | 사용자 프로파일 작업 |
| *Style Manager…* | 스타일 관리자 |
| *Custom Projections…* | 사용자 정의 좌표계 |
| *Keyboard Shortcuts…* | 키보드 단축키 |
| *Interface Customization…* | 사용자 지정 작업 |
| *Options…* | 옵션 |

리눅스 KDE 에서는 *Settings* 메뉴 아래에서 *Panels*, *Toolbars* and *Toggle Full Screen Mode* 같은 더 많은 도구들을 볼 수 있습니다.

### 3.1.6 플러그인

표 3.7: 플러그인 메뉴 항목

| 메뉴 옵션 | 단축키 | 툴바 | 참조 |
|---|---|---|---|
| *Manage and Install Plugins…* | | | 플러그인 대화창 |
| *Python Console* | Ctrl+Alt+P | *Plugins* | *QGIS* 파이썬 콘솔 |

처음으로 QGIS 를 실행했을 경우 핵심 플러그인을 모두 불러오지는 않습니다.

### 3.1.7 벡터

The *Vector* menu provides tools for analyzing, managing and manipulating vector datasets. By default, it is filled with a set of predefined *Processing* algorithms, grouped by sub-menus. This provides shortcuts for many common vector-based GIS tasks from core providers. If not all these sub-menus are available, enable the *Processing* plugin in *Plugins* → *Manage and Install Plugins*….

> **참고**
>
> The list of tools and their menu can be modified/extended with any Processing algorithms (read 공간 처리 프레임워크 환경 설정) or by some third-party *plugins*.

This is what the *Vector* menu looks like if all the core plugins are enabled.

표 3.8: 벡터 메뉴 항목

| 메뉴 옵션 | 단축키 | 툴바 | 참조 |
|---|---|---|---|
| *Check Geometries*… | | | 도형 점검기 플러그인 |
| *Topology Checker* | | *Vector* | 위상 점검기 플러그인 |
| *Geoprocessing Tools* | `Alt+O` + `G` | | |
| *Buffer*… | | | 버퍼 |
| *Clip*… | | | 잘라내기 *(Clip)* |
| *Convex Hull*… | | | 볼록 껍질 |
| *Difference*… | | | 차감하기 *(Difference)* |
| *Dissolve*… | | | 융해 |
| *Intersection*… | | | 교차 *(Intersection)* |
| *Symmetrical Difference*… | | | 대칭 차감 |
| *Union*… | | | 통합 *(Union)* |
| *Eliminate Selected Polygons*… | | | 선택한 폴리곤 소거하기 |
| *Geometry Tools* | `Alt+O` + `E` | | |
| *Centroids*… | | | 중심 |
| *Collect Geometries*… | | | 도형 모으기 |
| *Extract Vertices*… | | | 꼭짓점 추출하기 |
| *Multipart to Singleparts*… | | | 다중 부분에서 단일 부분으로 |
| *Polygons to Lines*… | | | 폴리곤을 라인으로 |
| *Simplify*… | | | 단순화 |
| *Check Validity*… | | | 무결성 점검하기 |
| *Delaunay Triangulation*… | | | 델로네 삼각분할 |
| *Densify by Count*… | | | 개수로 치밀화하기 |
| *Add Geometry Attributes*… | | | 도형 속성 추가하기 |

다음 페이지에 계속

표 3.8 – 이전 페이지에서 계속

| 메뉴 옵션 | 단축키 | 툴바 | 참조 |
|---|---|---|---|
| *Lines to Polygons…* | | | 라인을 폴리곤으로 |
| *Voronoi Polygons…* | | | 보로노이 폴리곤 |
| *Analysis Tools* | Alt+O + A | | |
| *Line Intersection…* | | | 라인 교차 |
| *Mean Coordinate(s)…* | | | 평균 좌표 (들) |
| *Basic Statistics for Fields…* | | | 필드에 대한 기본 통계 |
| *Count Points in Polygon…* | | | 폴리곤에서 포인트 개수 세기 |
| *Distance Matrix…* | | | 거리 매트릭스 |
| *List Unique Values…* | | | 유일 값 목록 |
| *Nearest Neighbour Analysis…* | | | 최근접 이웃 분석 |
| *Sum Line Lengths…* | | | 라인 길이 합계 |
| *Data Management Tools* | Alt+O + D | | |
| *Merge Vector Layers…* | | | 벡터 레이어 병합하기 |
| *Reproject Layer…* | | | 레이어 재투영하기 |
| *Create Spatial Index…* | | | 공간 인덱스 생성하기 |
| *Join Attributes by Location…* | | | 위치로 속성 결합하기 |
| *Split Vector Layer…* | | | 벡터 레이어 분할하기 |
| *Research Tools* | Alt+O + R | | |
| *Create Grid…* | | | 그리드 생성하기 |
| *Extract Layer Extent…* | | | 레이어 범위 추출하기 |
| *Random Points in Extent…* | | | 범위 안에 랜덤 포인트 생성하기 |
| *Random Points in Polygons…* | | | 폴리곤에 랜덤 포인트 생성하기 |
| *Random Points on Lines…* | | | 라인에 랜덤 포인트 생성하기 |
| *Select by Location…* | | | 위치로 선택하기 |
| *Select Within Distance…* | | | 거리 안에서 선택하기 |
| *Random Points in Layer Bounds…* | | | 레이어 경계 안에 랜덤 포인트 생성하기 |
| *Random Points Inside Polygons…* | | | 폴리곤 내부에 랜덤 포인트 생성하기 |
| *Random Selection…* | | | 랜덤하게 선택하기 |
| *Random Selection Within Subsets…* | | | 하위 집합 내에서 랜덤하게 선택하기 |
| *Regular Points…* | | | 정규 포인트 생성하기 |

### 3.1.8 래스터

The *Raster* menu provides tools for analyzing, managing and manipulating raster datasets. By default, it is filled with a set of predefined *Processing* algorithms, grouped by sub-menus. This provides shortcuts for many common raster-based GIS tasks from core providers. If not all these sub-menus are available, enable the *Processing* plugin in *Plugins* ▸ *Manage and Install Plugins*….

> **참고**
>
> The list of tools and their menu can be modified/extended with any Processing algorithms (read 공간 처리 프레임워크 환경 설정) or by some third-party *plugins*.

This is what the *Raster* menu looks like if all the core plugins are enabled.

표 3.9: 래스터 메뉴 항목

| 메뉴 옵션 | 단축키 | 툴바 | 참조 |
|---|---|---|---|
| *Raster calculator*… | | | 래스터 계산기 |
| *Align Rasters*… | | | 래스터들 정렬 |
| *Analysis* | | | |
| *Aspect*… | | | 향 |
| *Fill nodata*… | | | *NODATA* 채우기 |
| *Grid (Moving Average)*… | | | 그리드 *(*이동 평균*)* |
| *Grid (Data Metrics)*… | | | 그리드 *(*데이터 메트릭*)* |
| *Grid (Inverse Distance to a Power)*… | | | 그리드 *(*역거리 제곱*)* |
| *Grid (Nearest Neighbor)*… | | | 그리드 *(*최근접 이웃을 검색하는 역거리 가중치법*)* |
| *Near Black*… | | | 근사 검은색 |
| *Hillshade*… | | | 음영기복 |
| *Proximity (Raster Distance)*… | | | 근접성 *(*래스터 거리*)* |
| *Roughness*… | | | 거칠기 |
| *Sieve*… | | | 거르기 |
| *Slope*… | | | 경사 |
| *Topographic Position Index (TPI)*… | | | 위상 위치 지수 *(TPI)* |
| *Terrain Ruggedness Index (TRI)*… | | | 지형 험상 지수 *(TRI)* |
| *Projections* | | | |
| *Assign Projection*… | | | 투영체 할당하기 |
| *Extract Projection*… | | | 투영체 추출하기 |
| *Warp (Reproject)*… | | | 왜곡하기 *(*재투영*)* |
| *Miscellaneous* | | | |
| *Build Virtual Raster*… | | | 가상 래스터 작성하기 |

다음 페이지에 계속

표 3.9 – 이전 페이지에서 계속

| 메뉴 옵션 | 단축키 | 툴바 | 참조 |
|---|---|---|---|
| *Raster Information…* | | | 래스터 정보 |
| *Merge…* | | | 병합하기 |
| *Build Overviews (Pyramids)…* | | | 오버뷰 작성하기 (피라미드) |
| *Tile Index…* | | | 타일 인덱스 |
| *Extraction* | | | |
| *Clip Raster by Extent…* | | | 래스터를 범위로 잘라내기 |
| *Clip Raster by Mask Layer…* | | | 래스터를 마스크 레이어로 잘라내기 |
| *Contour…* | | | 등고선 |
| *Conversion* | | | |
| *PCT to RGB…* | | | *PCT* 에서 *RGB* 로 |
| *Polygonize (Raster to Vector)…* | | | 폴리곤화 (래스터를 벡터로) |
| *Rasterize (Vector to Raster)…* | | | 래스터화 (벡터를 래스터로) |
| *RGB to PCT…* | | | *RGB* 를 *PCT* 로 |
| *Translate (Convert Format)…* | | | 번역하기 (포맷 변환) |

### 3.1.9 데이터베이스

The *Database* menu provides tools for Database interaction. If no database-related plugins are enabled, there will be no *Database* menu. This is what the *Database* menu looks like if all the core plugins are enabled.

표 3.10: 데이터베이스 메뉴 항목

| 메뉴 옵션 | 단축키 | 툴바 | 참조 |
|---|---|---|---|
| *Offline editing…* | Alt+D + O | | 오프라인 편집 작업 플러그인 |
| *Convert to Offline Project…* | | *Database* | 오프라인 편집 작업 플러그인 |
| *Synchronize* | | *Database* | 오프라인 편집 작업 플러그인 |
| *Query History…* | | *Database* | *Creating SQL Query Layers* |
| *DB Manager…* | | *Database* | 데이터베이스 관리자 플러그인 |

처음으로 QGIS 를 실행했을 경우 핵심 플러그인을 모두 불러오지는 않습니다.

### 3.1.10 웹

다음은 모든 핵심 플러그인을 활성화했을 때 볼 수 있는 *Web* 메뉴의 모습입니다. 활성화된 웹 플러그인이 없을 경우, *Web* 메뉴도 표시되지 않을 것입니다.

표 3.11: 웹 메뉴 항목

| 메뉴 옵션 | 단축키 | 툴바 | 참조 |
|---|---|---|---|
| *MetaSearch* | Alt+W + M | | 메타검색 카탈로그 클라이언트 *(MetaSearch Catalog Client)* |
| *Metasearch* | | *Web* | 메타검색 카탈로그 클라이언트 *(MetaSearch Catalog Client)* |
| *Help* | | | 메타검색 카탈로그 클라이언트 *(MetaSearch Catalog Client)* |

처음으로 QGIS 를 실행했을 경우 핵심 플러그인을 모두 불러오지는 않습니다.

### 3.1.11 메시 (mesh)

*Mesh* 메뉴는메시 레이어 를 취급하는 데 필요한 도구를 제공합니다. 제 3 자 플러그인으로 이 메뉴에 다른 메뉴 항목을 추가할 수 있습니다.

표 3.12: 메시 메뉴 항목

| 메뉴 옵션 | 단축키 | 툴바 | 참조 |
|---|---|---|---|
| *Mesh Calculator…* | | | 메시 계산기 |
| *Reindex Faces and Vertices* | | | 메시 인덱스 재생성하기 |

### 3.1.12 공간 처리

표 3.13: 공간 처리 메뉴 항목

| 메뉴 옵션 | 단축키 | 툴바 | 참조 |
|---|---|---|---|
| *Toolbox* | Ctrl+Alt+T | | 툴박스 |
| *Model Designer…* | Ctrl+Alt+G | | 모델 설계자 |
| *History…* | Ctrl+Alt+H | | 이력 관리자 |
| *Results Viewer* | Ctrl+Alt+R | | 외부 응용 프로그램 환경 설정 |
| *Edit Features In-Place* | | | 제자리 *(in-place)* 공간 처리 레이어 수정자 |

처음으로 QGIS 를 실행했을 경우 핵심 플러그인을 모두 불러오지는 않습니다.

### 3.1.13 도움말

표 3.14: 도움말 메뉴 항목

| 메뉴 옵션 | 단축키 | 툴바 | 참조 |
|---|---|---|---|
| *Help Contents* | F1 | *Help* | |
| *API Documentation* | | | |
| *Plugins* | | | |
| *Report an Issue* | | | |
| *Need commercial support?* | | | |
| *QGIS Home Page* | Ctrl+H | | |
| *Check QGIS Version* | | | |
| *About* | | | |
| *QGIS Sustaining Members* | | | |

### 3.1.14 QGIS

맥 OS 에서만 이 메뉴를 사용할 수 있으며, OS 관련 명령어가 일부 포함되어 있습니다.

표 3.15: QGIS 메뉴 항목

| 메뉴 옵션 | 단축키 | 참조 |
|---|---|---|
| *About QGIS* | | |
| *Preferences* | Cmd+, | 옵션 |
| *Hide QGIS* | Cmd+H | |
| *Show All* | | |
| *Hide Others* | Shift+Cmd+H | |
| *Quit QGIS* | Cmd+Q | |

*Preferences* 는 다른 플랫폼의 *Settings* → *Options* 와 대응하고, *About QGIS* 는 *Help* → *About* 과 대응하며, *Quit QGIS* 는 *Project* → *Exit QGIS* 와 대응합니다.

## 3.2 패널 및 툴바

*View* 메뉴에서 (또는 *Settings* 에서) QGIS 위젯 (*Panels* ) 과 툴바 (*Toolbars* ) 를 켜고 끌 수 있습니다. 사용자가 메뉴 바 또는 툴바를 오른쪽 클릭해서 어떤 항목이라도 선택해서 활성화/비활성화시킬 수 있습니다. QGIS 인터페이스 안이라면, 각 패널 또는 툴바를 사용자가 편리하게 사용할 수 있는 어떤 위치로든 이동시킬 수 있습니다. 또한핵심 또는 외부 플러그인 을 활성화하면 목록이 늘어날 수도 있습니다.

### 3.2.1 툴바

메뉴와 동일한 기능들을 대부분 툴바를 통해서 접근할 수 있으며, 또한 맵과 상호 작용할 수 있는 추가적인 도구들도 제공합니다. 툴바의 각 항목은 팝업 도움말을 표시할 수 있습니다. 항목 위에 마우스를 가져 가면 해당 도구의 목적을 설명하는 짧은 텍스트를 표시합니다.

다음과 같은 툴바를 사용할 수 있습니다:

표 3.16: QGIS 툴바

| 툴바 이름 | 도구에 대한 기본 참조 |
|---|---|
| *Advanced Digitizing* | 고급 디지타이즈 작업 |
| *Annotations* | 주석 도구 |
| *Attributes* | 속성 테이블 작업, 일반 도구 |
| *Data Source Manager* | 데이터소스 관리 |
| *Database* | 데이터베이스 관리자 플러그인 |
| *Digitizing* | 기존 레이어 디지타이즈 작업 |
| *GPS* | 실시간 *GPS* 추적 |
| *GRASS* | *GRASS GIS* 통합 |
| *Help* | |
| *Label* | 라벨 툴바 |
| *Manage Layers* | 데이터 열기 |
| *Map Navigation* | 맵 뷰 탐구하기 |
| *Mesh Digitizing* | 메시 레이어 편집 |
| *Plugins* | 플러그인 |
| *Project* | 프로젝트 파일 작업, 맵 조판 작업, 스타일 라이브러리 |
| *Processing Algorithms* | 공간 처리 프레임워크 환경 설정 |
| *Raster* | 플러그인 |
| *Selection* | 피처 선택 |
| *Shape digitizing* | 형태 디지타이즈 작업 |
| *Snapping* | 스냅 허용 오차 및 검색 반경 설정 |
| *Vector* | 플러그인 |
| *Web* | 플러그인, 메타검색 카탈로그 클라이언트 *(MetaSearch Catalog Client)* |

**참고**

제 3 자 플러그인이 기본 툴바를 자체 도구로 확장시킬 수도 있고, 자체 툴바를 제공할 수도 있습니다.

**팁**

**툴바를 복구하기**

사용자가 실수로 툴바를 꺼버렸을 경우, *View* → *Toolbars* → 메뉴 옵션을 (또는  *Settings* → *Toolbars* → 를) 선택해서 다시 켤 수 있습니다. 어떤 이유로 인터페이스에서 툴바가 (또는 다른 위젯이) 완전히 사라졌을 경우, 초기 *GUI* 복구 를 보면 복구에 필요한 팁을 얻을 수 있습니다.

### 3.2.2 패널

QGIS 는 수많은 패널을 제공하고 있습니다. 패널이란 좀 더 복잡한 작업을 수행하기 위해 사용자와 상호작용 (옵션 선택, 체크박스 체크, 값 입력 등등) 할 수 있는 특별한 위젯입니다.

다음은 QGIS 가 제공하는 기본 패널 목록입니다:

표 3.17: QGIS 패널

| 패널 이름 | 단축키 | 참조 |
|---|---|---|
| *Advanced Digitizing* | `Ctrl+4` | 고급 디지타이즈 작업 패널 |
| *Browser* | `Ctrl+2` | 탐색기 패널 |
| *Browser (2)* | | 탐색기 패널 |
| *Debugging/Development Tools* | `F12` | 디버그/개발 도구 패널 |
| *Elevation Profile* | | 표고 단면 뷰 |
| *Geometry Validation* | | 디지타이즈 작업 속성 |
| *GPS Information* | `Ctrl+0` | 실시간 *GPS* 추적 |
| *GRASS Tools* | | *GRASS GIS* 통합 |
| *Layer Order* | `Ctrl+9` | 레이어 순서 패널 |
| *Layer Styling* | `Ctrl+3` | 레이어 스타일 작업 패널 |
| *Layers* | `Ctrl+1` | 레이어 패널 |
| *Log Messages* | | 로그 메시지 패널 |
| *Overview* | `Ctrl+8` | 오버뷰 패널 |
| *Processing Toolbox* | | 툴박스 |
| *Results Viewer* | | 툴박스 |
| *Snapping and Digitizing Options* | | 스냅 허용 오차 및 검색 반경 설정 |
| *Spatial Bookmark Manager* | `Ctrl+7` | 맵 상에서의 범위 북마크 작업 |
| *Statistics* | `Ctrl+6` | 통계 요약 패널 |
| *Temporal Controller* | | 시계열 제어기 패널 |
| *Tile Scale* | | 타일셋 |
| *Undo/Redo* | `Ctrl+5` | 실행 취소/재실행 패널 |
| *Vertex Editor* | | 꼭짓점 편집기 패널 |

## 3.3 상태 바

상태 바는 맵 뷰 및 처리 중이거나 처리할 수 있는 동작에 대한 일반적인 정보를 제공하며, 맵 뷰를 관리할 수 있는 도구들도 들어 있습니다.

### 3.3.1 위치 탐색란

상태 바 왼쪽에 있는 빠른 검색 위젯인 위치 탐색란 (locator bar) 을 통해 QGIS 에 있는 모든 기능 또는 옵션들을 찾아서 실행할 수 있습니다:

1. 위치 탐색란을 활성화하려면 텍스트 위젯을 클릭하거나 `Ctrl+K` 조합키를 누르십시오.
2. 사용자가 찾고 있는 항목과 관련된 텍스트를 (이름, 태그, 키워드 등등) 입력하십시오. 기본적으로 활성화된 모든 위치 탐색 필터에 대해 검색 결과를 반환하지만, 사용자의 텍스트 앞에위치 탐색 필터 접두어를 입력하면 특정 필터로 검색을 제한할 수 있습니다. 예를 들어 `l cad` 라고 입력하면 레이어명에 `cad` 가 들어간 레이어만 반환할 것입니다.

   위치 탐색 위젯을 활성화하면 나타나는 메뉴에서 특정 필터를 더블 클릭해서 필터를 선택할 수도 있습니다.
3. 검색 결과를 클릭하면 항목의 유형에 따라 대응하는 액션을 실행합니다.

> **팁**
>
> **검색을 활성 레이어의 특정 필드 (들) 로 제한하기**
>
> "활성 레이어 피처 (active layer features)"필터 (`f`) 로 검색하면 기본적으로 해당 레이어의 속성 테이블 전체를 대상으로 검색을 실행합니다. `@` 접두어를 사용하면 특정 필드만 검색하도록 제한할 수 있습니다. 예를 들어, `f @name sal` 또는 `@name sal` 이라고 입력하면"name"속성에'sal'을 포함하고 있는 피처만 반환합니다. 입력 도중 자동완성이 활성화되기 때문에 `Tab` 키를 사용하면 자동완성 제안을 적용할 수 있습니다.
>
> 레이어의 *Fields* 탭에서 쿼리 대상 필드에 대해 더 고급의 조정을 할 수 있습니다. 자세한 내용은필드 속성 을 읽어보세요.

검색은 스레드를 통해 처리되기 때문에 느린 검색 필터를 설치했어도 가장 빠른 속도로 검색 결과를 반환합니다. 또 필터에 걸리자마자 결과값을 반환하기 때문에, 예를 들면 파일 검색 필터는 파일 트리를 스캔하는 과정에서 검색 결과를 하나씩 표시하게 됩니다. 즉 아주 느린 (예를 들어 온라인 서비스를 사용하는) 검색 필터인 경우에도 사용자 인터페이스가 멈추는 일은 없을 것입니다.

> **참고**
>
> 노미나팀 위치 탐색 도구는 오픈스트리트맵의 노미나팀 활용 정책 에 따라 다른 습성을 (자동 완성 검색 비활성화, 검색 결과 가져오기 지연 등등을) 보일 수도 있습니다.

> **팁**
>
> **위치 탐색 환경 설정에 빠르게 접근하기**
>
> 상태 바에 있는 위치 탐색 위젯의 아이콘을 클릭하면 사용할 수 있는 필터 목록을 표시하는데, 이 목록에서 *Configure* 항목을 클릭하면 *Settings* ▸ *Options*⋯메뉴의 *Locator* 탭이 열립니다.

### 3.3.2 액션 보고하기

필요한 경우 사용자가 수행해온 (레이어에 있는 피처를 선택, 레이어를 제거, 이동한 거리 및 방향 같은) 액션들의 요약이, 또는 사용자가 마우스를 가져다 댄 도구에 대한 자세한 설명이 (모든 도구가 이를 지원하지는 않습니다) 위치 검색란 옆에 표시될 것입니다.

래스터 레이어에서 통계를 수집하거나, 공간 처리 알고리즘을 실행하거나, 맵 뷰에서 레이어 몇 개를 렌더링하는 등 시간이 걸리는 작업의 경우, 상태 바에 진행 막대를 표시해서 해당 작업의 현재 진행 상태를 보여줍니다.

### 3.3.3 맵 캔버스 제어하기

*Coordinate* 옵션은 맵 뷰 상에서 마우스의 움직임을 따라가며 커서의 현재 위치를 표시합니다. *Project* ▸ *Properties*⋯▸ *General* 탭에서 이 기능에 사용할 단위 (와 정확도) 를 설정할 수 있습니다. 텍스트 란 왼쪽에 있는 작은 버튼을 클릭하면 이 *Coordinate* 옵션과, 사용자가 맵을 이동하거나 확대/축소할 때 맵 뷰의 현재 좌하단 및 우상단 포인트의 좌표를 맵 단위로 표시하는 *Extents* 옵션을 서로 바꿔가며 켜고 끌 수 있습니다.

좌표 표시 옆에 *Scale* 표시가 보입니다. 이 표시는 맵 뷰의 축척을 보여줍니다. 사용자가 맵 뷰에 적용할사전 정의된/사용자 지정 축척 을 선택할 수 있는 축척 선택기 (scale selector) 가 있습니다.

축척 표시 오른쪽에 있는  버튼을 누르면 돋보기로 확대 또는 축소할 때 사용할 축척을 고정시킬 수 있습니다. 돋보기를 사용하면 맵 축척을 변경하지 않은 채 맵을 확대할 수 있어서 라벨 및 심볼의 위치를 쉽고 정확하게 조정할 수 있습니다. 돋보기의 수준은 백분율로 표시됩니다. *Magnifier* 가 100% 수준이라면 현재 맵은 확대되지 않은 상태, 다시 말해 모니터의 해상도 (DPI) 에 정확히 맞는 축척으로 렌더링된 상태입니다. *Settings* ▸ *Options* ▸ *Rendering* ▸ *Rendering Behavior* 에서 돋보기의 확대 기본값을 정의할 수 있습니다. 이는 고해상도 스크린에서 작은 심볼을 확대해서 봐야 하는 경우 매우 유용합니다. 또한, *Settings* ▸ *Options* ▸ *Canvas & Legend* ▸ *DPI* 설정을 통해 QGIS 가 각 모니터의 물리적 DPI 를 존중할지 아니면 전체 시스템의 논리적 DPI 를 존중할지를 정할 수 있습니다.

돋보기 (Magnifier) 오른쪽에서 맵 뷰의 시계 방향 회전량 (rotation) 을 도 (°) 단위로 정의할 수 있습니다.

상태 바 오른쪽에 있는  *Render* 체크박스를 이용하면 맵 뷰 렌더링을 일시적으로 막을 수 있습니다. (맵 렌더링 제어하기 을 참조하십시오.)

현재 프로젝트 좌표계를 보여주는  *EPSG:code* 버튼이  *Render* 기능 오른쪽에 있습니다. 이 버튼을 클릭하면 맵 뷰를 재투영하거나, 다른 프로젝트 속성을 조정할 수 있는 *Project Properties* 대화창이 열립니다.

> **팁**
>
> **사용자 맵 캔버스의 정확한 축척을 계산하기**
>
> QGIS 실행 시, 기본 좌표계는 `WGS 84 (EPSG 4326)` 이고 단위는 도 (°) 입니다. 즉 QGIS 는 사용자 레이어의 모든 좌표를 도 단위로 해석할 것이란 뜻입니다. 정확한 축척값을 얻으려면 이 설정을, 예를 들어 *Project* ▸ *Properties*…메뉴의 *General* 탭에서 미터 단위로 직접 변경하거나, 또는 앞에서 설명한  EPSG:code 아이콘을 통해 변경할 수도 있습니다. 후자의 경우, 프로젝트의 투영체가 지정하는 대로 (예를 들어 `+units=us-ft` 처럼) 단위를 설정하게 됩니다.
>
> *Settings* ▸ *Options* ▸ *CRS Handling* 메뉴 옵션에서 QGIS 시작 시 기본적으로 적용할 좌표계를 설정할 수 있다는 사실을 알아두십시오.

### 3.3.4 메시지 작업

그 옆의  Messages 버튼을 클릭하면 기저 프로세스 (QGIS 실행, 플러그인 불러오기, 공간 처리 도구 등등) 에 대한 정보를 알려주는 *Log Messages Panel* 을 엽니다.

CHAPTER 4

# 탐색기 패널


- 탐색기에서 열/실행할 수 있는 리소스들
- 탐색기 패널 최상위 수준 항목들
  - 즐겨찾기
  - 공간 북마크
  - 프로젝트 홈
  - 드라이브 및 파일 시스템
  - *Data type*
  - 데이터베이스 항목
  - 타일 및 웹 서비스
  - *Project and supported file types*


QGIS 탐색기 패널은 QGIS 리소스를 탐색하고, 검색하고, 점검하고, 복사하고, 불러올 수 있는 훌륭한 도구입니다. QGIS 가 어떻게 처리하는지 알고 있는 리소스만 탐색기에 표시됩니다.

탐색기 패널을 사용하면탐색기 패널 에서 설명한대로 데이터를 찾고, 점검하고, 추가할 수 있습니다. 또한 탐색기 패널은 프로젝트 파일, 파이썬 스크립트, 공간 처리 스크립트, 공간 처리 모델 등과 같은 여러 QGIS 리소스의 드래그 & 드롭을 지원합니다.

파이썬 스크립트, 공간 처리 스크립트 및 모델은 각각 외부 편집기 및 모델 설계자 (model designer) 에서도 열고 편집할 수 있습니다.

You can drag and drop layers from the *Layers* panel to the *Browser* panel, for instance into a GeoPackage or a PostgreSQL database.

탐색기 패널 (그림 4.1) 은 탐색기가 처리하는 리소스들을 체계화시키는 몇몇 고정된 최상위 수준 항목들을 가진, 확장 가능한 위계로 이루어져 있습니다. 노드 항목명 왼쪽에 있는 ▶ 아이콘을 클릭하면 노드 항목을 펼칩니다.

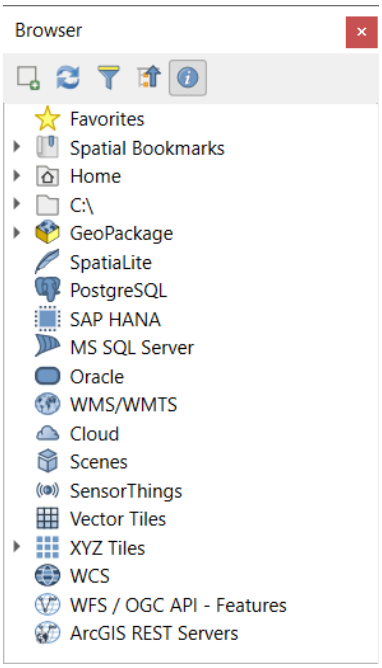



그림 4.1: 탐색기 패널

아이콘을 클릭하면 해당 분기 (branch) 를 접습니다. Collapse All 버튼을 누르면 모든 최상위 항목을 접습니다.

*Settings → Interface Customization* 메뉴에서 리소스들을 비활성화시킬 수 있습니다. 예를 들어 사용자가 탐색기에서 파이썬 스크립트를 보지 않고자 하는 경우 *Browser → py* 메뉴 항목을 비활성화하면 되고, 탐색기에서 사용자 홈 폴더를 제거하고 싶다면 *Browser → special:Home* 메뉴 항목을 비활성화하면 됩니다.

항목명 (위계에 있는 지엽 (leaf) 항목과 노드 항목 둘 다) 기반 검색에 필터 ( Filter Browser) 를 쓸 수 있습니다. 필터 텍스트란 옆에 있는 Options 풀다운 메뉴를 사용하면:

- *Case Sensitive* 대소문자를 구분하는 검색 기능을 켜고 끌 수 있습니다.
- *Filter pattern syntax* 필터 패턴 문법을 다음 가운데 하나로 설정할 수 있습니다.
  - *Normal*
  - *Wildcard(s)*
  - *Regular Expressions*

Enable/disable properties widget 버튼을 사용해서 일부 항목/리소스에 대한 유용한 정보를 표시하는 속성 위젯 *(Properties widget)* 을 활성화/비활성화시킬 수 있습니다. 이 위젯을 활성화하면 탐색기 패널 하단에 그림 4.2 처럼 표시됩니다.

*View → Panels* 메뉴에서 *Browser (2)* 패널을 활성화하면 두번째 탐색기 패널을 열 수 있습니다. 탐색기 위계에서 서로 다른 분기들 깊숙히 위치한 리소스들 사이에 레이어를 복사하는 경우 두번째 탐색기 패널이 있으면 편리합니다.

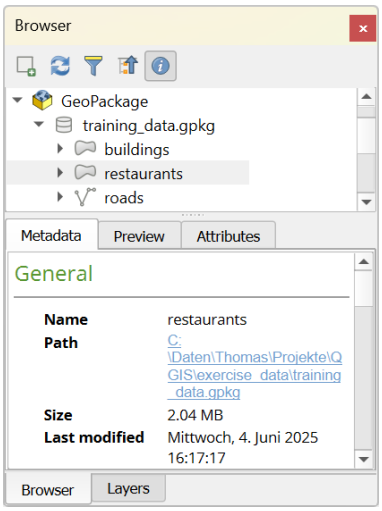



그림 4.2: 속성 위젯

## 4.1 탐색기에서 열/실행할 수 있는 리소스들

탐색기 패널에서 많은 작업을 완수할 수 있습니다:

- 더블클릭, 맵 캔버스로 드래그, 또는 (레이어를 선택한 다음) Add Selected Layers 버튼을 클릭해서 사용자 맵에 벡터, 래스터 및 메시 레이어를 추가할 수 있습니다.
- 더블클릭 또는 맵 캔버스로 드래그해서 (공간 처리 알고리즘 포함) 파이썬 스크립트를 실행할 수 있습니다.
- 더블클릭 또는 맵 캔버스로 드래그해서 모델을 실행할 수 있습니다.
- *Extract Symbols*…컨텍스트 메뉴를 통해 QGIS 프로젝트 파일에서 심볼을 추출할 수 있습니다.
- 각 파일 유형의 (컨텍스트 메뉴의 *Open <file type> Externally*…에서 설정하는) 기본 응용 프로그램으로 파일을 열 수 있습니다. 예를 들면 HTML 파일, 스프레드시트, 이미지, PDF, 텍스트 등등의 파일들 말이죠.
- 항목을 복사할 수 있습니다.
- (여러 개의) 레이어들을 재명명 및 삭제할 수 있습니다. (컨텍스트 메뉴의 *Manage* )
- *Show in Files* 메뉴를 통해 파일 탐색창을 열어서 파일을 직접 선택할 수 있습니다.

리소스 전용 액션은 다음에 정리한 최상위 수준 항목 목록 아래 정렬된 서로 다른 리소스 그룹별로 목록화돼 있습니다.

## 4.2 탐색기 패널 최상위 수준 항목들

### 4.2.1 즐겨찾기

자주 사용하는 파일 시스템 위치를 즐겨찾기로 태그할 수 있습니다. 사용자가 태그한 위치가 여기 나타날 것입니다.

홈 *(Home)* 에서 설명하는 작업들 이외에도, *Rename Favorite*…및 *Remove Favorite* 컨텍스트 메뉴를 통해 즐겨찾기를 재명명하거나 제거할 수 있습니다.

### 4.2.2 공간 북마크

This is where you will find your *spatial bookmarks*, divided into *Project Bookmarks* and *User Bookmarks*.

표 4.1: available actions for spatial bookmarks

| 수준 | Contextual menu |
|---|---|
| *Spatial Bookmarks* | *New Spatial Bookmark…* |
| | *Show the Spatial Bookmark Manager* |
| | *Import Spatial Bookmarks…* |
| | *Export Spatial Bookmarks…* |
| Bookmark group (*Project Bookmarks* and *User Bookmarks*) | *New Spatial Bookmark…* |
| | *Import Spatial Bookmarks…* |
| | *Export Spatial Bookmarks…* |
| Individual bookmark | *Zoom to Bookmark* |
| | *Edit Spatial Bookmark…* |
| | *Delete Spatial Bookmark* |

### 4.2.3 프로젝트 홈

프로젝트 파일을 저장하면 사용할 수 있는 *Project home* 항목은 현재 프로젝트 안에서 쓰일 수도 있는 데이터 및 기타 내용 (스크립트, 모델, 텍스트 등등) 을 담고 있는 폴더입니다. 이 항목은 *Browser* 패널에 표시되어 사용자가 프로젝트의 데이터 및 기타 파일에 빠르게 접근할 수 있게 해줍니다.

프로젝트 홈 폴더의 기본 경로는 프로젝트 파일 폴더이지만, *Project → Properties… → General → Project home* 옵션을 통해 또는 탐색기 패널의 *Project Home* 항목을 오른쪽 클릭해서 *Set project home…*을 선택하면 변경할 수 있습니다. QGIS 프로젝트가 데이터셋과 함께 조직화된'프로젝트'의 루트 폴더에 저장되어 있지 않은 경우 이 폴더를 사용자 지정하는 것이 특히 유용합니다.

### 4.2.4 드라이브 및 파일 시스템

*Browser* 패널의 다음 항목은 사용 중인 OS 에 따라 다르며, 해당 OS 의 파일 시스템의 최상위 수준 항목과 관련되어 있습니다.

이 최상위 수준 항목이란 주로:

- 현재 사용자의 홈 폴더를 가리키는 *Home* 폴더
- 또는 유닉스 기반 기기의 경우 루트 `/` 폴더
- 그리고 로컬 또는 네트워크 연결 드라이브입니다. 이런 드라이브들은 OS 에 따라 직접 목록화되거나 (예: `C:\`, `D:\`) 또는 `/Volumes` 항목을 통해 목록화됩니다.

이런 폴더 또는 드라이브 각각의 컨텍스트 메뉴로부터 다음과 같은 작업을 할 수 있습니다:

- 폴더 및 드라이브의 내용을 새로고침할 수 있습니다.
- *New*: 새로운 하위 항목을 *Directory*, *GeoPackage* 또는 ESRI *Shapefile* 포맷 데이터셋으로 생성할 수 있습니다.
- *Hide from Browser*: 홈 디렉터리를 숨길 수 있습니다.
- *Set color*: 복잡한 폴더 구조를 빠르게 탐색할 수 있도록 폴더 아이콘 색상을 사용자 지정할 수 있습니다.
- *Scanning*: 스캔 작업을 활성화할 수 있습니다.
  - *Monitor for changes*: 특정 디렉터리를 모니터링하면서 자동 업데이트할지를 직접 조정할 수 있게 해줍니다. 이 설정은 선택한 디렉터리와 모든 하위 디렉터리에 적용됩니다. 즉 사용자가 아무 문제도

없다는 사실을 알고 있는 경우 네트워크 드라이브 모니터링을 직접 동의할 수도 있고, 또는 다른 이유로 모니터링을 원하지 않는 대용량 디렉터리의 모니터링을 동의하지 않을 수도 있습니다. 기본적으로 원격 또는 네트워크 드라이브를 자동 모니터링하지는 않습니다.

  - *Fast scan this directory*: 선택한 디렉터리를 빠르게 스캔합니다.

- *Open Directory*…: 사용자의 파일 관리자에서 디렉터리를 열 수 있습니다.
- *Open in Terminal*…: 터미널 창에서 디렉터리를 열 수 있습니다.
- 속성을 검사하거나 (*Properties*…), 디렉터리 속성을 관리할 수 있습니다 (*Directory Properties*…).

### 4.2.5 Data type

표 4.2: Available actions for plain layer files

| Level and Context menu | Types of layers<br>Raster (*Reference*) | Vector (*Reference*) | Point clouds (*Reference*) | Mesh (*Reference*) |
|---|---|---|---|---|
| **File** | | | | |
| *Delete File <dataset name>* | ☑ | ☑ | | |
| *Show in Files* | ☑ | ☑ | ☑ | ☑ |
| *Manage* | ☑ | ☑ | ☑ | ☑ |
| *File Properties*… | ☑ | ☑ | ☑ | ☑ |
| *Open <file type> Externally*… | ☑ | | | |
| **Layer** | | | | |
| *Add Layer to Project* | ☑ | ☑ | ☑ | ☑ |
| *Export Layer* | ☑ | ☑ | | |
| *Layer Properties*… | ☑ | ☑ | ☑ | ☑ |
| *Open with Data Source Manager*… | ☑ | ☑ | | |
| *Execute SQL*… | | ☑ | | |

### 4.2.6 데이터베이스 항목

사용자의 OS 와 설치된 드라이버에 따라, QGIS 에서 사용할 수 있는 서로 다른 데이터베이스 유형에 접근할 수 있을지도 모릅니다. 다음은 데이터셋 트리의 각 수준에서 사용할 수 있는 서로 다른 컨텍스트 메뉴 목록입니다.

표 4.3: Available actions for database entries

| Level and Context menu | 데이터베이스 유형 | | | | | |
|---|---|---|---|---|---|---|
| | GeoPackage ([1]) | SpatiaLite | PostgreSQL | SAP HANA | MS SQL Server | Oracle |
| **Top menu** | | | | | | |

다음 페이지에 계속

표 4.3 – 이전 페이지에서 계속

| | | | | | | |
|---|---|---|---|---|---|---|
| *New Connection…*: 기존 데이터베이스에 새 연결 생성 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| *Create Database…*: 새 데이터베이스 생성 | ✓ | ✓ | | | | |
| *Create Database and Layer…* | ✓ | | | | | |
| *Save Connections…*: 연결의 상세 내용을 파일로 저장 | | | ✓ | ✓ | ✓ | |
| *Load Connections…*: 저장한 연결 파일을 불러오기 | | | ✓ | ✓ | ✓ | |
| **Connection / Database** | | | | | | |
| *Refresh*: 연결을 새로고침 | | | ✓ | ✓ | ✓ | ✓ |
| *Edit Connection…*: 연결 설정을 편집 | | | ✓ | ✓ | ✓ | ✓ |
| *Duplicate Connection* | | | ✓ | ✓ | ✓ | ✓ |
| *Remove Connection…* | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| *Import Vector Layer…* | ✓ | ✓ | ✓ | ✓ | ✓ | |
| *New Field Domain* → *New Range Domain*: 새 필드 도메인을 범위로 설정 | ✓ | | | | | |
| *New Field Domain* → *New Coded Values Domain*: 새 필드 도메인을 코딩된 값으로 설정 | ✓ | | | | | |
| *New Field Domain* → *New Glob Domain*: 새 필드 도메인을 glob 함수로 설정 | ✓ | | | | | |
| *Delete* <*database_name*>: 데이터베이스를 삭제 | ✓ | ✓ | | | | |
| *Compact Database (VACUUM)*: 데이터베이스를 정리 | ✓ | | | | | |
| *New Schema…*: 새 스키마를 생성 | | | ✓ | ✓ | ✓ | |
| *New Table…*: 새 테이블을 생성 | ✓ | ✓ | ✓ | ✓ | | |
| *Execute SQL…*: SQL 쿼리를 실행 | ✓ | ✓ | ✓ | ✓ | | |
| **Schema** | | | | | | |
| *Refresh*: 스키마를 새로고침 | | | ✓ | ✓ | ✓ | |
| *Schema Operations* → *Rename Schema…*: 스키마 이름을 재명명 | | | ✓ | ✓ | ✓ | |

다음 페이지에 계속

표 4.3 – 이전 페이지에서 계속

| | | | | | |
|---|---|---|---|---|---|
| *Schema Operations → Delete Schema…*: 스키마를 삭제 | | | ☑ | ☑ | ☑ |
| *New Table…*: 새 테이블을 생성 | | | ☑ | ☑ | |
| *Execute SQL…*: SQL 쿼리를 실행 | | | ☑ | ☑ | |
| **Table / Layer** | | | | | |
| *Table Operations → Rename Table…*: 테이블 이름을 재명명 | | | ☑ | ☑ | ☑ |
| *Table Operations → Truncate Table…*: 테이블의 모든 행을 제거 | | | ☑ | | ☑ |
| *Execute SQL…*: SQL 쿼리를 실행 | ☑ | ☑ | ☑ | | |
| *Export Layer → To file…*: 레이어를 파일로 내보내기 | ☑ | ☑ | ☑ | ☑ | ☑ |
| *Manage → Rename Layer <layer_name>…*: 레이어 이름을 변경 | ☑ | ☑ | | | ☑ |
| *Manage → Delete Layer <layer_name>…*: 레이어를 삭제 | ☑ | ☑ | ☑ | ☑ | ☑ |
| *Manage → Delete Selected Layers*: 선택한 레이어들을 삭제 | ☑ | ☑ | ☑ | ☑ | ☑ |
| *Manage → Add Layer to Project*: 프로젝트에 레이어를 추가 | ☑ | ☑ | ☑ | ☑ | ☑ |
| *Manage → Add Selected Layers to Project*: 프로젝트에 선택한 레이어들을 추가 | ☑ | ☑ | ☑ | ☑ | ☑ |
| *Manage → Move to Another Schema* | | | ☑ | | ☑ |
| *Layer Properties…*: 레이어 속성 대화창 열기 | ☑ | ☑ | ☑ | ☑ | ☑ |
| *File Properties…*: 파일 속성 대화창 열기 | ☑ | | | | |
| *Open with Data Source Manager* | ☑ | ☑ | | | |
| **Fields** | | | | | |
| *Add New Field…*: 새로운 필드를 추가 | ☑ | ☑ | ☑ | ☑ | |
| **Field** | | | | | |

다음 페이지에 계속

표 4.3 – 이전 페이지에서 계속

| | | | | | |
|---|---|---|---|---|---|
| *Set Field Domain*: 필드 도메인을 설정 | ☑ | | | | |
| *Rename Field* ([2]) | ☑ | | ☑ | | ☑ |
| *Set Alias*…: 별명을 설정 | ☑ | | | | |
| *Set Comment*…: 주석을 설정 | ☑ | | ☑ | | |
| *Delete Field*…: 필드를 삭제 | ☑ | ☑ | ☑ | ☑ | |

**참고**

QGIS 에 벡터 레이어를 불러올 때, (GeoPackage 또는 ESRI 파일 지오데이터베이스에서 정의하는 필드들처럼) *Field Domains* 를 가진 필드들을 자동으로 감지합니다. 이런 도메인은 데이터베이스 수준의 제약조건으로, 다시 말해 데이터베이스 자체가 강제하는 도메인이기 때문에 QGIS 만이 아니라 서로 다른 응용 프로그램들 전반에 적용됩니다.

### Importing vector layers into databases using Browser panel

QGIS allows you to import vector layers directly into a database from the *Browser* panel. This provides a simple way to transfer datasets into supported database formats without using DB Manager (see 데이터베이스 관리자 플러그인).

You can start the import in two ways:

- Drag and drop a single vector layer from the Browser panel or from the Project into a database connection or schema item.
- Right-click on a database connection or schema item and choose *Import Vector Layer*… from the context menu.

**참고**

When multiple layers are dragged into a database, they are imported immediately without further options. All imports run in the background, so the QGIS interface remains responsive during the process.

To add a layer using *Import Vector Layer*… follow these steps:

1. Add *Source layer*: the layer to import, chosen from the current QGIS project. To import only selected features of the source layer use the ☑ *Selected features only* option.
2. Set the *Table name*: the name of the destination table in the database. By default, it is taken from the source layer name but can be modified. You can also use the ☑ *Replace destination table (if exists)* to overwrite an existing table.
3. Define the *Output CRS*: the coordinate reference system of the destination table. By default, it is the CRS of the source layer, but it can be changed using the *CRS selector*.
4. Configure the *Attributes*:

[1] 호환될 수 있을 경우, ESRI File Geodatabase, FlatGeobuf, GeoParquet, NetCDF 등등 같은 몇몇 GDAL 지원 벡터 파일 포맷들 에 대해 다른 항목들을 사용할 수 있을 수도 있습니다.

[2] For PostgreSQL connections: only supported with ALTER TABLE privilege.

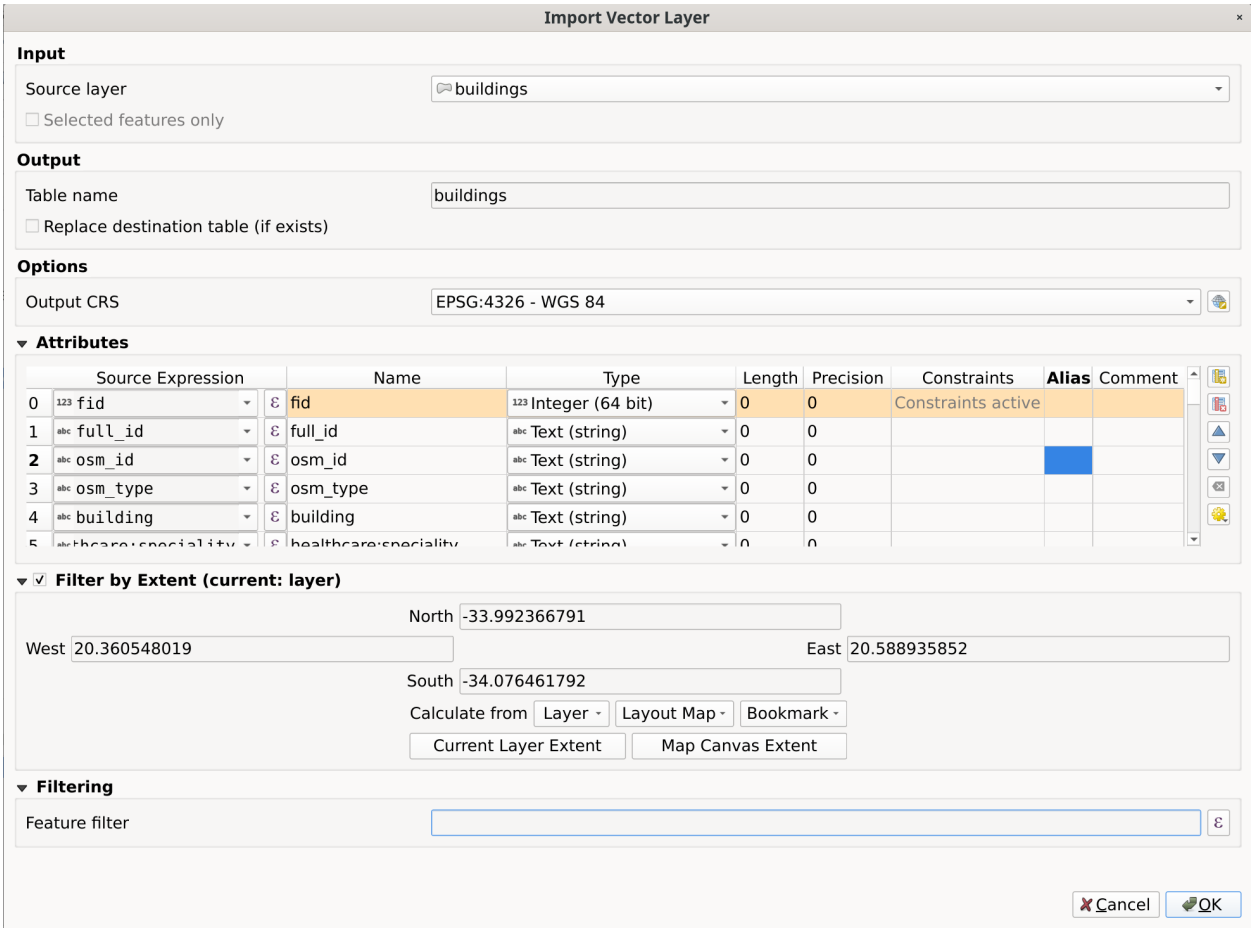



그림 4.3: Import Vector Layer dialog

- *Source expression*: expression used to populate the field. By default, this corresponds to the original source field.
- *Name*: name of the field in the destination table.
- *Type*: field data type (integer, text, real, etc.).
- *Length* and *Precision*: length and precision of the field, where applicable.
- *Constraints*: rules to enforce on the field values (e.g. NOT NULL, UNIQUE, DEFAULT). Support depends on the destination database.
- *Alias*: an alternate display name for the field.
- *Comment*: a description or note attached to the field.

You can use buttons on the right side to:

-  Add new field to add a new field to the destination table.
-  Delete selected field to remove a field from the destination table.
-  Move selected field up or  Move selected field down to reorder fields.
-  Reset all fields to clear all fields and start from scratch.
-  Convert All Fields To Lowercase or  Convert All Fields To Uppercase, to change the case of all field names.

5. Optionally, apply a spatial filter with *Filter by Extent*. This restricts the import to features within a defined bounding box.
6. Optionally, apply an attribute filter with *Feature filter*. This allows you to use an expression to select which features are imported.

### 4.2.7 타일 및 웹 서비스

표 4.4: Available actions for web services (1)

| Level and Context menu | 서비스 유형 | | | | |
|---|---|---|---|---|---|
| | WMS / WMTS (*Reference*) | XYZ Tiles (*Reference*) | WCS (*Reference*) | WFS / OGC API - Features (*Reference*) | ArcGIS REST Servers (*Reference*) |
| **Top menu** | | | | | |
| *New Connection*…: 새 연결을 생성 | ☑ | ☑ | ☑ | ☑ | ☑ |
| *Save Connections*…: 연결의 상세 내용을 파일로 저장 | ☑ | ☑ | ☑ | ☑ | ☑ |
| *Load Connections*…: 저장한 연결 파일을 불러오기 | ☑ | ☑ | ☑ | ☑ | ☑ |
| **Connection** | | | | | |
| *Refresh*: 연결을 새로고침 | ☑ | ☑ | ☑ | ☑ | ☑ |

다음 페이지에 계속

표 4.4 – 이전 페이지에서 계속

| Level and Context menu | 서비스 유형<br>WMS / WMTS (*Reference*) | XYZ Tiles (*Reference*) | WCS (*Reference*) | WFS / OGC API - Features (*Reference*) | ArcGIS REST Servers (*Reference*) |
|---|---|---|---|---|---|
| *Edit Connection…*: 연결 설정을 편집 | ☑ | ☑ | ☑ | ☑ | ☑ |
| *Duplicate Connection* | ☑ | ☑ | ☑ | ☑ | ☑ |
| *Remove Connection…* | ☑ | ☑ | ☑ | ☑ | ☑ |
| *View Service Info*: 웹 브라우저에서 서비스 정보를 보기 | | | | | ☑ |
| **Table / Layer** | | | | | |
| *Export Layer ⍰ To File…* | ☑ | ☑ | ☑ | ☑ | ☑ |
| *Add Layer to Project* | ☑ | ☑ | ☑ | ☑ | ☑ |
| *Add Filtered Layer to Project…* | | | | | ☑ |
| *Layer Properties…*: 레이어 속성 대화창 열기 | ☑ | ☑ | ☑ | ☑ | ☑ |
| *View Service Info*: 웹 브라우저에서 서비스 정보를 보기 | | | | | ☑ |

표 4.5: Available actions for web services (2)

| Level and Context menu | 서비스 유형<br>Vector Tiles (*Reference*) | Scene (*Reference*) | Cloud (*Reference*) | SensorThings (*Reference*) | STAC (*Reference*) |
|---|---|---|---|---|---|
| **Top menu** | | | | | |
| *New Connection…*: 새 연결을 생성 | | ☑ | ☑ | ☑ | ☑ |
| *New Generic Connection…*: 새 일반 연결을 생성 | ☑ | | | | |
| *New ArcGIS Vector Tile Service Connection…*: 새 ArcGIS 벡터 타일 서비스 연결을 생성 | ☑ | | | | |
| *Save Connections…* details to a file | ☑ | ☑ | ☑ | ☑ | ☑ |
| *Load Connections…*: 저장한 연결 파일을 불러오기 | ☑ | ☑ | ☑ | ☑ | ☑ |
| **Connection** | | | | | |

다음 페이지에 계속

표 4.5 – 이전 페이지에서 계속

| Level and Context menu | 서비스 유형 Vector Tiles (*Reference*) | Scene (*Reference*) | Cloud (*Reference*) | Sensor-Things (*Reference*) | STAC (*Reference*) |
|---|---|---|---|---|---|
| *Edit Connection…*: 연결 설정을 편집 | ☑ | ☑ | ☑ | ☑ | ☑ |
| *Duplicate Connection* | ☑ | ☑ | ☑ | ☑ | |
| *Delete Connection…* | ☑ | ☑ | ☑ | ☑ | ☑ |
| *Refresh* | | | | | ☑ |
| *Details…* | | | | | ☑ |
| **Table / Layer** | | | | | |
| *Export Layer → To File…* | | | | ☑ | |
| *Add Layer to Project* | ☑ | ☑ | | ☑ | |
| *Layer Properties…*: 레이어 속성 대화창 열기 | ☑ | ☑ | | ☑ | |
| *Refresh* | | | | | ☑ |
| *Download assets…* | | | | | ☑ |
| *Details…* | | | | | ☑ |

### 4.2.8 Project and supported file types

표 4.6: Available actions for various supported files :class: longtable

| File type | 컨텍스트 메뉴 |
|---|---|
| QGIS Project file (.qgz, .qgs) | *Open Project* |
| | *Extract Symbols…* (opens Style Manager to export symbols to XML, add to default style, or export as PNG/SVG) |
| | *File Properties…* |
| QGIS Layer Definition (.qlr) | *Export Layer → To File…* |
| | *Add Layer to Project* |
| | *Layer Properties…* |
| Processing Model (.model3) | *Run Model…* |
| | *Edit Model…* |
| Print Composer Template (.qpt) | *New Layout from Template* |
| Python script (.py) | *Run Script…* |
| | *Open in External Editor* |

CHAPTER 5

# QGIS 환경 설정

*Settings* 메뉴를 통해 QGIS 환경을 다음과 같이 매우 자세하게 설정할 수 있습니다:

- *Style Manager*···: 심볼, 스타일 및 색상표 를 생성하고 관리합니다.
- *Custom Projections*···: 사용자 자신만의좌표계 를 생성합니다.
- *Keyboard Shortcuts*···: 사용자 자신만의키보드 단축키 모음을 정의합니다. 그리고 각 QGIS 세션마다 (*Project* 메뉴 아래 있는) 프로젝트 속성 을 이용해서 이 단축키 정의를 무시할 수도 있습니다.
- *Interface Customization*···: 사용자가 필요로 하지 않는 대화창이나 도구를 숨기는 등, 응용 프로그램 인터페이스 를 설정합니다.
- *Options*···: 소프트웨어의 서로 다른 영역들에 적용할 전체 수준옵션 을 설정합니다. 설정한 옵션들은 활성화된사용자 프로파일 설정에 저장되어, 해당 프로파일과 함께 새로 프로젝트를 열 때마다 기본적으로 적용됩니다.

## 5.1 옵션

*Options* 대화창을 통해 QGIS 에 대한 몇몇 기본 옵션을 선택할 수 있습니다. *Settings* → *Options* 메뉴 옵션을 선택하십시오. 사용자의 필요에 따라 옵션을 수정할 수 있습니다. 일부 변경 사항은 적용되기 전에 QGIS 재시작이 필요할 수도 있습니다.

사용자 고유 옵션을 사용자 지정할 수 있는 탭들에 대해 다음 항에서 설명하고 있습니다.

> **참고**
>
> **플러그인 설정을 옵션 대화창에 내장시킬 수 있습니다.**
>
> 여기에서는 핵심 설정만 설명하지만, 설치된 플러그인 이 표준 옵션 대화창에 고유 옵션을 삽입하여 해당 목록을 확장시킬 수 있다는 사실을 기억하십시오. 이렇게 해서 각 플러그인이 각각 자신만의 추가 메뉴 항목을 가진 설정 대화창을 여는 것을 막을 수 있습니다.

### 5.1.1 일반 설정

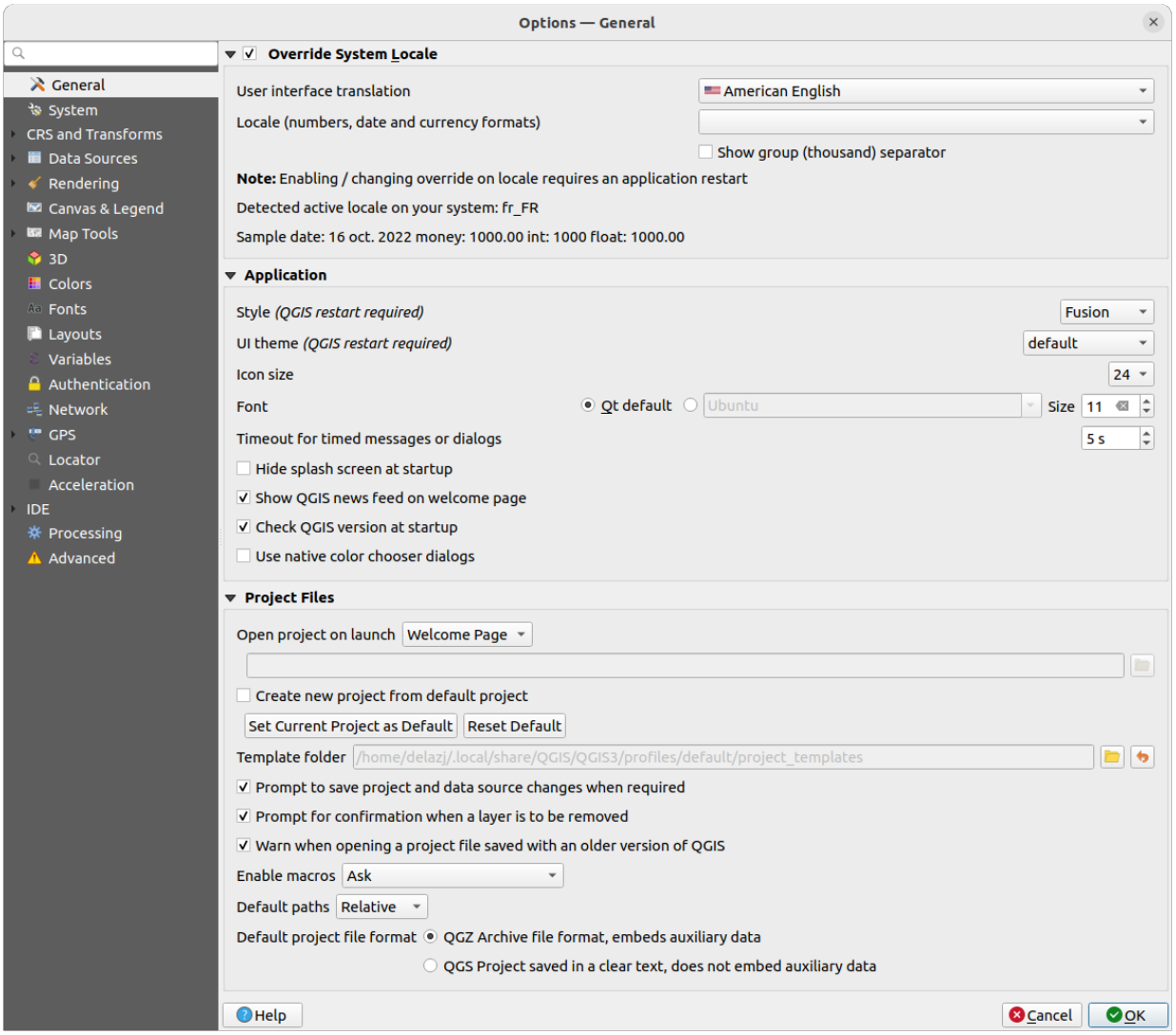



그림 5.1: 일반 설정

#### 시스템 로케일 무시

QGIS 는 기본적으로 사용자의 운영체제 설정에 의존해서 언어를 설정하고 숫자값 (numerical value) 을 처리합니다. 이 옵션 그룹을 활성화하면 사용자가 이런 로케일 습성을 사용자 지정할 수 있습니다.

- *User interface translation* 에서 GUI 에 적용할 언어를 선택합니다.

- *Locale (number, date and currency formats)* 에서 날짜 및 수치를 입력하고 렌더링하는 체계를 선택합니다.
- *Show group (thousand) separator* : 천단위 구분자를 활성화합니다.

사용자가 선택한 설정 및 해당 설정이 어떻게 해석되는지에 대한 요약 설명이 대화창 프레임의 하단에 표시됩니다.

**응용 프로그램**

- 대화창에 표시되는 위젯의 모양 및 위치 등을 설정하려면 *Style (QGIS restart required)* 을 선택하십시오. 사용자의 운영체제에 따라 선택할 수 있는 옵션값이 달라집니다.
- *UI theme (QGIS restart required)* 에서 UI 테마를 정의하십시오. 'default', 'Night Mapping', 또는'Blend of Gray'가운데 하나를 선택할 수 있습니다.
- *Icon size* 에서 아이콘 크기를 정의하십시오.
- *Font* 와 *Size* 에서 글꼴 및 그 크기를 정의하십시오. 글꼴은 *Qt default* 일 수도 있고, 사용자가 글꼴을 선택할 수도 있습니다.
- *Timeout for timed messages or dialogs* 에서 시한이 설정되는 메시지 또는 대화창의 시한을 정의하십시오.
- *Hide splash screen at startup* 에서 QGIS 실행 시 로딩 이미지 표시 여부를 설정하십시오.
- *Show QGIS news feed on welcome page*: 환영 페이지에 사용자가 직접 프로젝트 소식 (사용자/개발자 모임 날짜 및 요약, 커뮤니티 설문조사, 배포판 발표, 다양한 팁들 등등) 을 볼 수 있는 기획된 QGIS 뉴스피드를 표시할지 여부를 설정하십시오.
- *Check QGIS version at startup* 을 체크하면 새 버전이 출시됐는지에 대한 소식을 놓치지 않을 수 있습니다.
- *Use native color chooser dialogs* : 내장된 색상 선택 대화창 사용 여부를 설정합니다. (색상 선택기 참조)

**프로젝트 파일**

- *Open project on launch*
  - '환영 페이지'(기본값): "뉴스"피드, 프로젝트 템플릿 (들), 그리고사용자 프로파일 의 (썸네일로 표시된) 최근 프로젝트들을 표시할 수 있습니다. 기본적으로 어떤 프로젝트도 열지 않습니다.
  - '새 프로젝트': 기본 템플릿에 기반한 새 프로젝트를 엽니다.
  - '최근 프로젝트': 마지막으로 저장했던 프로젝트를 다시 엽니다.
  - '특정 프로젝트': 특정 프로젝트를 엽니다. …버튼을 통해 기본값으로 사용할 프로젝트를 정의합니다.
- *Create new project from default project* 를 선택하면 *Set current project as default* 또는 *Reset default* 가운데 하나를 클릭할 수 있습니다. 사용자 파일을 탐색하고 사용자가 직접 정의한 프로젝트 템플릿이 있는 디렉터리를 정의할 수 있습니다. 해당 템플릿은 *Project → New From Template* 메뉴에 추가됩니다. 먼저 *Create new project from default project* 를 활성화한 경우, 프로젝트 템플릿 폴더에 프로젝트를 저장하십시오.
- *Prompt to save project and data source changes when required*: 사용자의 변경 사항을 잃지 않으려면 이 옵션을 체크하십시오.
- *Prompt for confirmation when a layer is to be removed*: 실수로 레이어를 삭제하는 일이 없으려면 이 옵션을 체크하십시오.
- *Warn when opening a project file saved with an older version of QGIS*: QGIS 구 버전의 프로젝트 파일을 열려고 할 때 경고창을 엽니다. QGIS 신 버전은 언제나 구 버전에서 생성된 프로젝트를 열 수 있지만, 신 버전의 프로젝트를 구 버전으로 열려고 할 경우 해당 버전에서 지원하지 않는 피처가 있을 수 있기 때문에 실패할 수도 있습니다.

- *Enable project's embedded Python code*  옵션은 프로젝트 이벤트에 대한 액션을 수행하기 위해 작성된 매크로는 물론 표현식으로 사용될 사용자 지정 파이썬 함수의 실행을 처리합니다. 'Never', 'Ask', 'For this session only'그리고'Always (not recommended)'가운데 하나로 선택할 수 있습니다.
- *Default paths*: 새 프로젝트에서 쓰이는 파일 및 레이어 경로가 프로젝트 파일 기준'절대 경로'로 저장되는지 또는'상대 경로'로 저장되는지 여부를 정의합니다. 이 설정은 프로젝트 수준에서 무시될 수 있습니다.
- *Default project file format*
  -  *QGZ Archive file format, embeds auxiliary data*: QGZ 압축 파일 포맷으로 보조 데이터를 내장합니다. (보조 데이터 참조)
  -  *QGS Project saved in a clear text, does not embed auxiliary data*: 프로젝트 파일과 별개로, 개별 `.qgd` 파일에 보조 데이터를 저장합니다.

### 5.1.2 시스템 설정

#### SVG 경로

*Path(s) to search for Scalable Vector Graphic (SVG) symbols* 를 추가하거나 삭제합니다. 피처 심볼 또는 라벨을 적용하거나, 사용자 맵 조판을 장식하는 데 이 SVG 파일을 쓸 수 있습니다.

QGIS 경로에서 SVG 파일을 참조할 수 있는 다른 방법에 대해서는원격 또는 내장 파일 선택기 도 읽어보세요.

#### 플러그인 경로

*Path(s) to search for additional C++ plugin libraries* 를 추가하거나 삭제합니다.

#### 문서 경로

QGIS 도움말에 사용할 *Documentation Path(s)* 를 추가하거나 삭제합니다. 사용 중인 QGIS 버전에 대응하는 공식 온라인 사용자 지침서를 가리키는 링크 (예: `https://docs.qgis.org/$qgis_short_version/$qgis_locale/docs/user_manual/`) 가 기본값으로 추가됩니다. 하지만 다른 링크들을 추가하고 위에서 아래로 우선순위를 정할 수 있습니다. 사용자가 대화창에서 *Help* 버튼을 클릭할 때마다, 가장 위에 있는 링크를 검색하고 만약 해당 페이지를 찾을 수 없다면 그 다음 링크, 다시 그 다음 링크를 검색할 것입니다.

> **참고**
>
> QGIS LTR(Long Term Release) 버전에 대해서만 문서의 버전을 붙이고 번역합니다. 즉 사용자가 (QGIS 3.0 같은) 일반 배포판을 실행하는 경우, 도움말 버튼을 클릭하면 기본적으로 다음 (QGIS 3.4 LTR 같은) LTR 버전 사용자 지침서 페이지를 열 것입니다. 이 사용자 지침서는 (3.2 및 3.4 같은) 신 배포판의 기능 설명을 포함하고 있을 수도 있습니다. LTR 버전 문서를 사용할 수 없을 경우, 최신 개발 중인 버전의 기능을 설명하는 실험용 문서로 연결합니다.

#### 설정

사용자가 어떤사용자 지정 사항이라도 만들었을 경우 *Reset user interface to default settings (restart required)* 옵션을 통해 초기화할 수 있습니다.

#### 환경

System environment variables can be viewed, and many configured, in the **Environment** group. This is useful for platforms, such as Mac, where a GUI application does not necessarily inherit the user's shell environment. It's also useful for setting and viewing environment variables for the external tool sets controlled by the Processing toolbox (e.g., GRASS), and for turning on debugging output for specific sections of the source code.

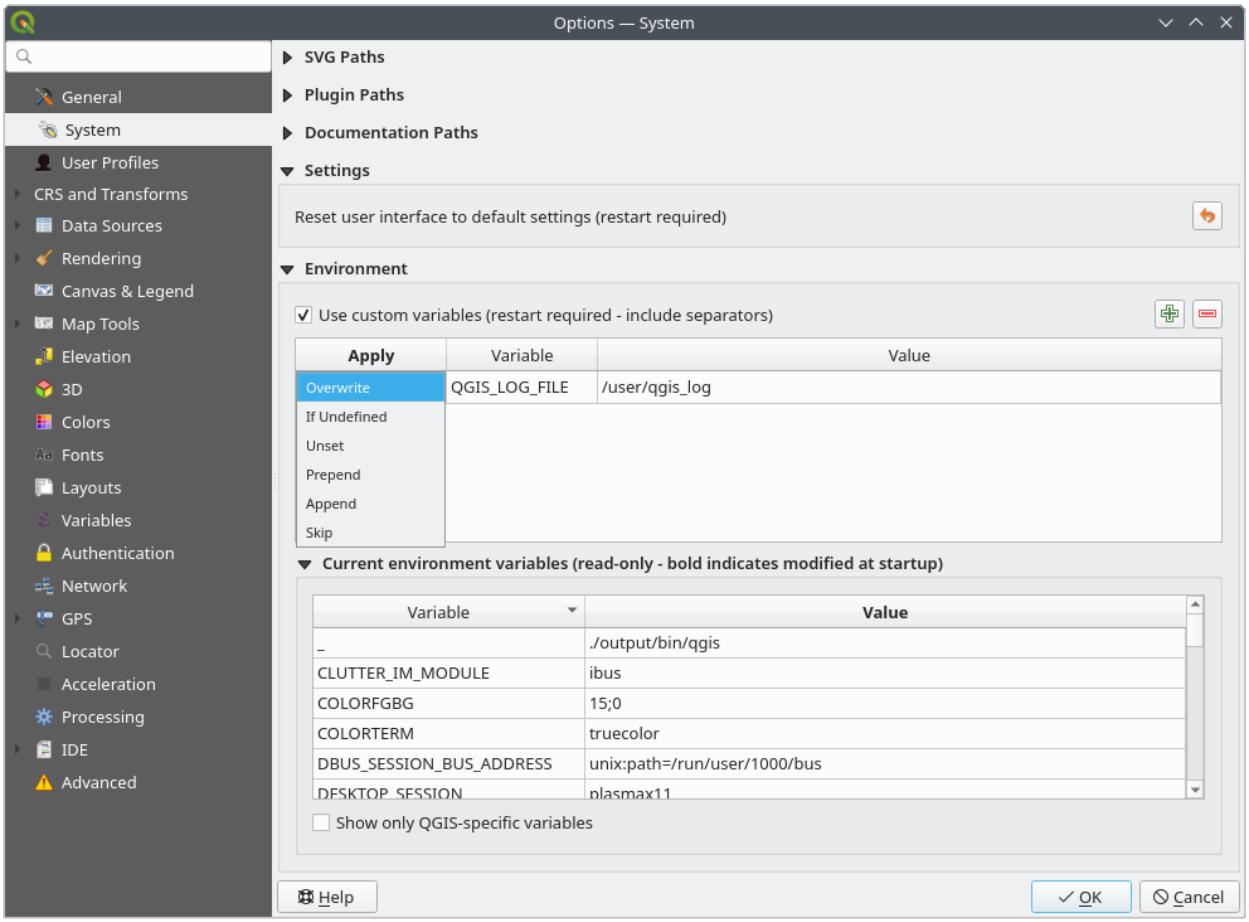



그림 5.2: 시스템 환경 변수

*Use custom variables (restart required - include separators)* 옵션을 활성화하면 환경 변수를 추가 Add 하고 제거 Remove 할 수 있습니다. 새로운 항목 각각에 대해 이름을 명명 *Variable* 하고 값을 설정 *Value* 하는 데 사용할 메소드를 환경 설정 *Apply* 할 수 있습니다. 이런 메소드들 가운데 일부를 소개하자면:

- *Overwrite*: 변수의 모든 기존 값을 대체합니다.
- *If undefined*: 해당 변수가 더 상위 수준에서 (예를 들면 OS 또는 응용 프로그램 수준에서) 이미 정의되어 있지 않은 경우, 변수에 이 값을 사용합니다.
- *Unset*: 환경으로부터 이 변수를 제거합니다. (*Value* 파라미터는 쓰이지 않습니다.)
- *Prepend*: 변수의 기존 값 앞에 이 값을 붙입니다.
- *Append*: 변수의 기존 값 뒤에 이 값을 붙입니다.
- *Skip*: 향후 참조를 위해 목록에는 이 항목을 유지하지만, 사용하지는 않습니다.

기존에 정의된 환경 변수는 *Current environment variables* 에 표시되며, *Show only QGIS-specific variables* 옵션을 활성화하면 이 현재 환경 변수들을 필터링할 수 있습니다.

### 5.1.3 사용자 프로파일 설정

> **참고**
>
> 사용자 프로파일을 어떻게 관리하는지 자세히 알고 싶다면, 사용자 프로파일 작업 에서 전문적으로 설명하고 있습니다.

### 5.1.4 좌표계 및 변형 설정

> **참고**
>
> QGIS 가 레이어 투영체를 어떻게 처리하는지 자세히 알고 싶다면, 투영 작업 에서 전문적으로 설명하고 있습니다.

#### 좌표계 처리

*CRS Handling* 탭에서 새 프로젝트 또는 레이어에 사용될 좌표계를 환경 설정할 수 있습니다.

**프로젝트 용 좌표계**

새 프로젝트의 좌표계를 자동적으로 설정하는 옵션이 있습니다:

- *Use CRS from first layer added*: 프로젝트의 좌표계를 처음 불러온 레이어의 좌표계로 설정할 것입니다.
- *Use a default CRS*: 모든 새 프로젝트에 미리 설정해둔 좌표계를 적용하며, 프로젝트에 어느 레이어를 추가해도 변경하지 않습니다.

사용자가 어떤 옵션을 선택하든, 선택한 옵션을 차후 QGIS 세션에도 사용하도록 저장할 것입니다. 하지만 언제든지 *Project* → *Properties…* → *CRS* 탭에서 프로젝트의 기존 좌표계를 무시하고 새로 적용할 수 있습니다.

**레이어 용 좌표계**

*Default CRS for layers*: 새 레이어를 생성할 때 사용할 좌표계를 선택합니다.

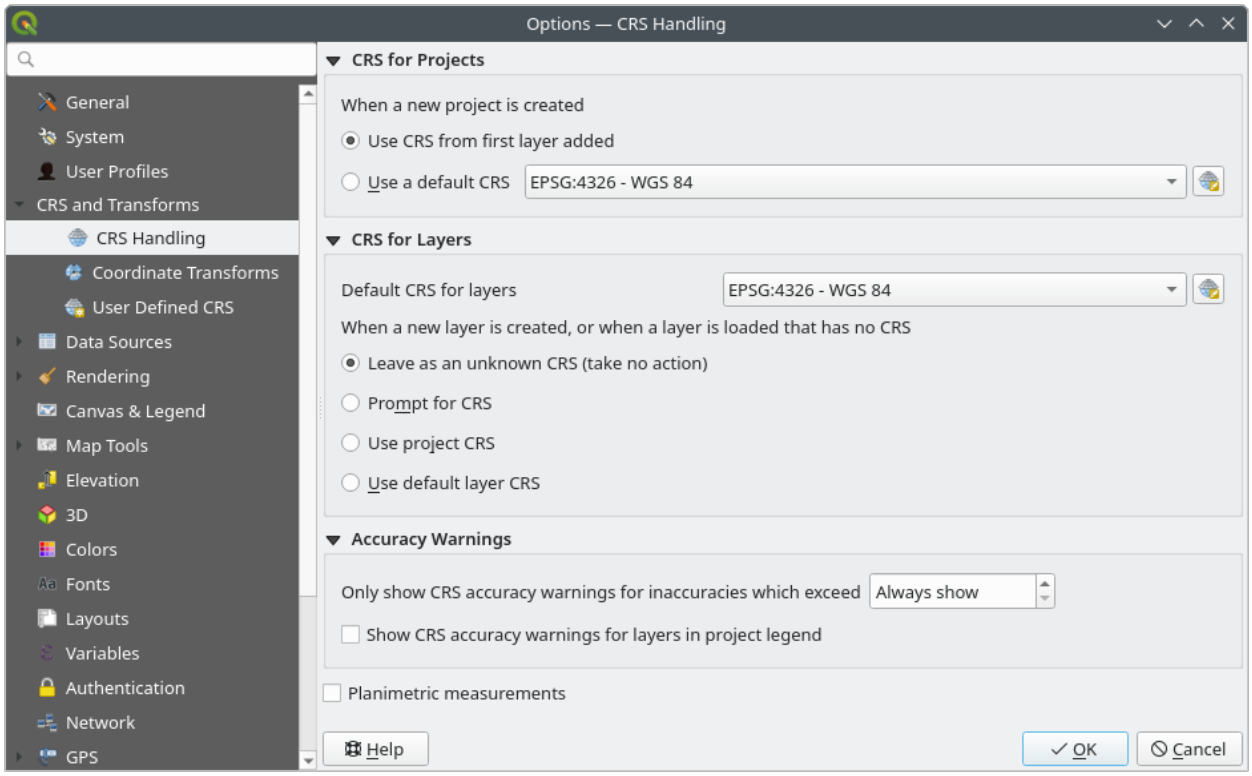



그림 5.3: 좌표계 설정

새 레이어를 생성하거나 좌표계가 정의되지 않은 레이어를 불러올 때 수행할 액션을 정의할 수도 있습니다.

- *Leave as unknown CRS (take no action)*: 좌표계를 정의하지 않은 상태로 유지합니다.
- *Prompt for CRS*: 좌표계 선택기를 엽니다.
- *Use project CRS*: 프로젝트 좌표계를 적용합니다.
- *Use default layer CRS*: 레이어 기본 좌표계를 적용합니다.

#### 정밀도 경고

*Only show CRS accuracy warnings for inaccuracies which exceed* a given distance: occurs when you are explicitly creating or modifying a dataset and select a CRS based on a datum ensemble with lower accuracy. The default is to `Always show` the warning if any inaccuracy.

*Show CRS accuracy warning for layers in project legend*: 이 옵션을 활성화하면 *Layers* 패널에서 정밀도 문제가 있는 (예를 들어 사용 가능한 좌표 시대 (coordinate epoch) 가 없는 동적 좌표계, 또는 사용자가 설정한 제한을 초과하는 비정밀도를 내재한 기준점 총체를 기반으로 하는 좌표계를 가진) 모든 레이어에 낮은 정밀도를 나타내는 경고 아이콘을 표시할 것입니다.

이 옵션은 미터/미터 미만 수준의 정밀도 문제가 아주 위험할 수 있거나 비용이 크게 소모될 수도 있는 엔지니어링, BIM(Building Information Modeling), 자산 관리 및 기타 영역에서 활용할 수 있도록 개발되었습니다!

*Planimetric measurements*: 새로 생성한 프로젝트 용평면 측정 *(planimetric measurements)* 속성 기본값을 설정합니다.

### 좌표 변환

*Coordinate Transforms* 탭에서, 프로젝트에 레이어를 불러오거나 레이어를 재투영할 때 적용할 좌표 변환 방법을 설정할 수 있습니다.

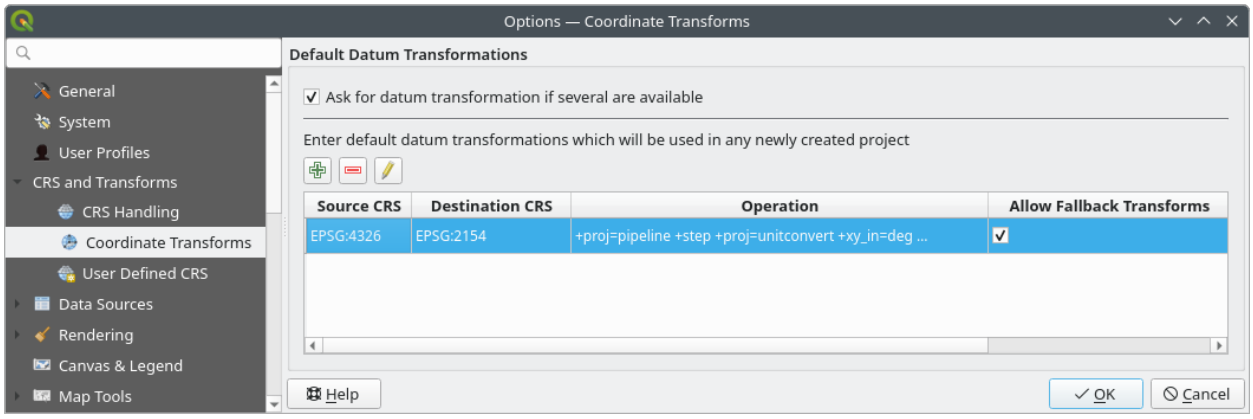



그림 5.4: 변환 설정

#### 기본 기준 (datum) 변환

이 부분에서는 레이어 좌표계를 또다른 좌표계로 재투영하는 작업을 다음과 같이 제어할 수 있습니다:

- QGIS 기본 변환 설정을 사용해서 자동적으로 공간 처리할 수 있습니다.
- 그리고/또는 사용자 지정 선호도를 통해 다음과 같이 더 자세히 조정할 수 있습니다:
  - *Ask for datum transformation if several are available*: 여러 변환 방법을 사용할 수 있을 때 기준 변환 방법을 사용자가 설정할 수 있게 할지 여부를 설정합니다.
  - 기본으로 적용할 사전 정의 데이터 변환 목록을 선택할 수 있습니다. 자세한 내용은기준점 *(datum)* 변환을 참조하세요.

변환을 추가 Add, 제거 Remove 또는 편집 Edit 할 수 있습니다. 새로 생성된 모든 프로젝트에서 이 작업들을 할 수 있습니다.

### 사용자 정의 좌표계

*User Defined CRS* 탭에서 사용자 지정 좌표계를 정의할 수 있습니다. 이때 사용자 정의 좌표계는 WKT 또는 PROJ 문자열 서식을 준수해야만 합니다.

*Name* 을 설정한 다음 Add new CRS 아이콘을 클릭하십시오. 기존 사용자 정의 좌표계를 삭제하려면 Remove CRS 아이콘을 클릭하면 됩니다.

정의

- ***Format***
  - WKT (권장)
  - PROJ 문자열 (레거시 - 권장하지 않음)
- ***Parameters***

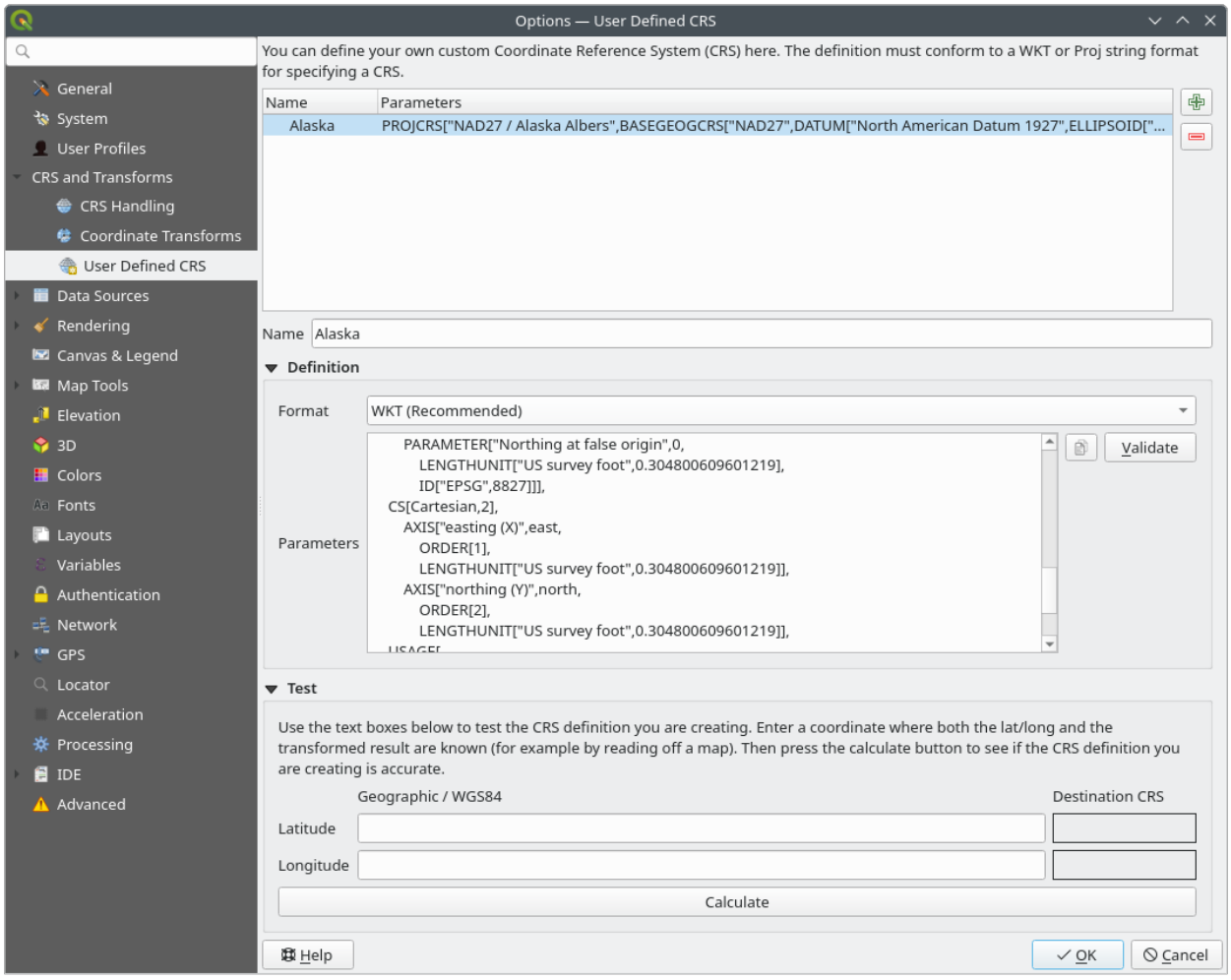



그림 5.5: 사용자 정의 좌표계

- : 기존 좌표계로부터 파라미터를 복사
- *Validate*: 사용자의 표현식에 오류가 없는지 검사

**테스트**

이 부분에서 사용자가 생성한 좌표계 정의를 위도 및 경도로 테스트할 수 있습니다. 이미 알고 있는 좌표를 사용하면 사용자 정의 좌표계가 정확한지를 알 수 있습니다.

### 5.1.5 데이터소스 설정

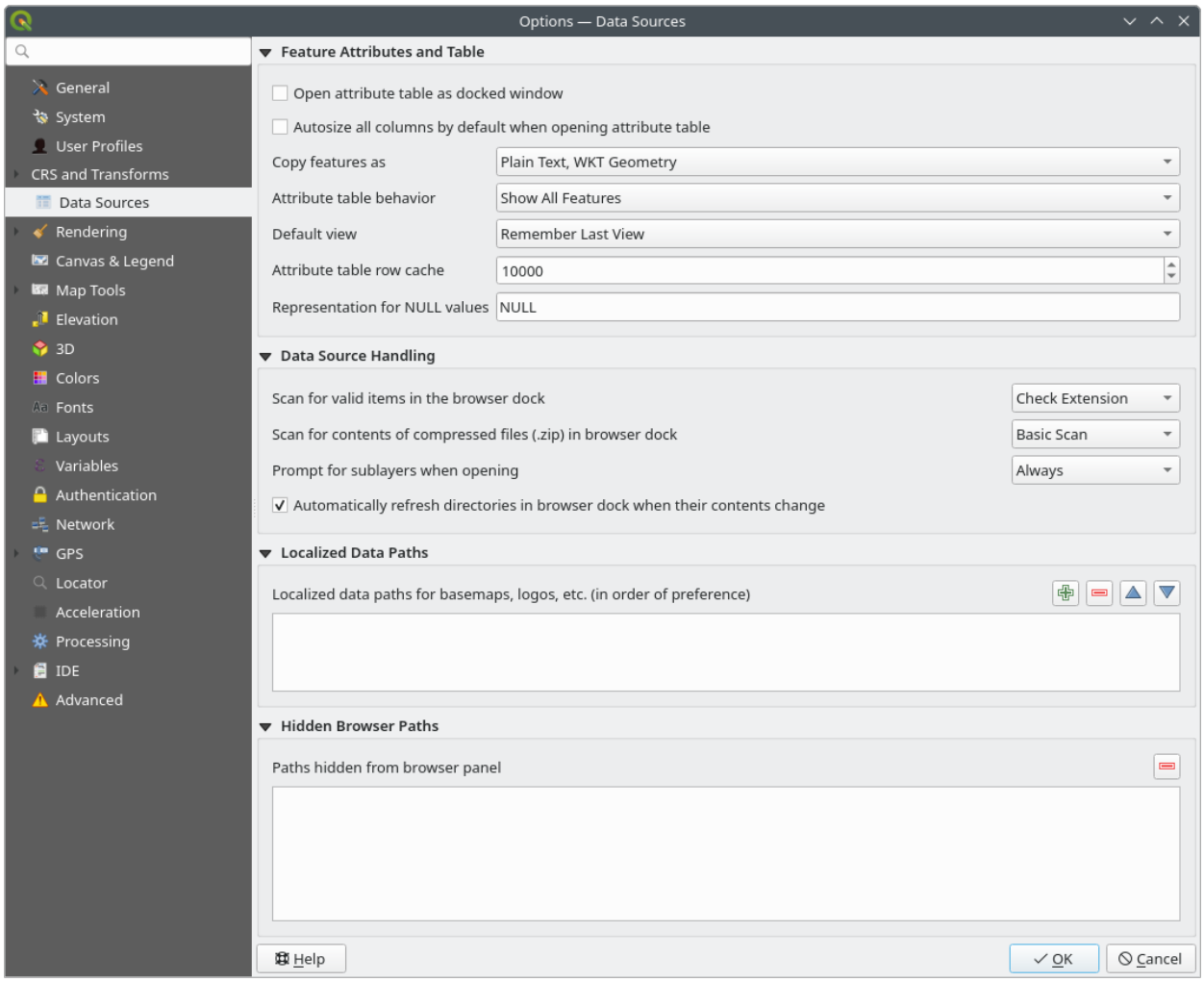



그림 5.6: 데이터소스 설정

**피처 속성 및 테이블**

- *Open attribute table as docked window*: 속성 테이블을 도킹된 창으로 엽니다.
- *Autosize all columns by default when opening attribute table*: 속성 테이블을 열 때 모든 열의 크기를 기본값으로 조정합니다.
- *Copy features as*: 다른 응용 프로그램에 피처를 붙여넣을 때 데이터에 사용할 포맷을 정의합니다.

  - *Plain text, no geometry*: 속성을 텍스트로 복사하지만 도형은 건너뜁니다.
  - *Plain text, WKT geometry*: 속성을 텍스트로 복사하고 도형을 WKT 로 반환합니다.
  - *Plain text, WKB geometry*: 속성을 텍스트로 복사하고 도형을 WKB(HEX 포맷) 로 반환합니다.
  - *GeoJSON*: 속성 및 도형을 GeoJSON 데이터 포맷으로 만듭니다.
- *Attribute table behavior*  : 속성 테이블을 열 때의 필터를 설정합니다. 'Show all features', 'Show selected features'그리고'Show features visible on map'가운데 하나를 선택할 수 있습니다.
- *Default view*: 속성 테이블을 열 때의 뷰 모드를 정의합니다. 'Remember last view', 'Table view'또는'Form view' 가운데 하나를 선택할 수 있습니다.
- *Attribute table row cache*  에서 속성 테이블 관련 작업을 빠르게 처리할 수 있도록 최근 속성 행을 몇 개나 행 캐시에 저장할지 설정할 수 있습니다. 속성 테이블을 닫으면 캐시를 삭제합니다.
- *Representation for NULL values* 을 통해 NULL 값을 포함하는 데이터 항목을 위한 값을 정의할 수 있습니다.

> **팁**
>
> **대용량 속성 테이블 열기를 향상시키기**
>
> 대용량 레코드를 보유한 레이어를 작업하는 경우, 대화창이 레이어에 있는 모든 행을 요구하기 때문에 속성 테이블을 여는 데 시간이 많이 걸릴 수도 있습니다. *Attribute table behavior* 를 **Show Features Visible On Map** 으로 설정하면, QGIS 가 테이블을 열 때 현재 맵 캔버스에 있는 피처만 요구하므로 데이터를 빨리 불러올 수 있습니다.
>
> 이 속성 테이블 인스턴스에 있는 데이터가 언제나 테이블을 열 당시의 캔버스 범위에 구속된다는 사실을 기억하십시오. 즉 해당 테이블에서 **Show All Features** 를 선택해도 새 피처들을 표시하지 않을 것이라는 뜻입니다. 하지만 속성 테이블에서 **Show Features Visible On Map** 옵션을 선택하고 캔버스 범위를 변경하면 화면에 표시된 피처 집합을 업데이트할 수 있습니다.

**데이터소스 처리**

- *Scan for valid items in the browser dock*  을 통해'Check extension'과'Check file contents'가운데 하나를 선택할 수 있습니다.
- *Scan for contents of compressed files (.zip) in browser dock*  을 통해 `.zip` 압축 파일을 쿼리할 경우 탐색기 패널 하단의 위젯 정보를 얼마나 자세히 표시할지 정의합니다. 'No', 'Basic scan'그리고'Full scan'가운데 하나를 선택할 수 있습니다.
- *Prompt for sublayers when opening*: 일부 래스터는 GDAL 에서 하위 데이터셋이라고 부르는 하위 레이어를 지원합니다. netCDF 파일이 한 예입니다. netCDF 변수가 여러 개 있을 경우, GDAL 은 모든 변수를 하위 데이터셋으로 간주합니다. 이 옵션은 하위 레이어를 보유한 파일을 열었을 때 하위 레이어를 어떻게 처리할지를 제어할 수 있게 해줍니다. 다음 가운데 하나를 선택할 수 있습니다:
  - 'Always': (하위 레이어가 존재할 경우) 항상 물어봅니다.
  - 'If needed': 레이어에 밴드는 없지만 하위 레이어는 있을 경우 물어봅니다.
  - 'Never': 물어보지 않고, 아무것도 불러오지 않습니다.
  - 'Load all': 물어보지 않지만, 모든 하위 레이어를 불러옵니다.
-  *Automatically refresh directories in browser dock when their contents change*: 이 옵션을 비활성화하면 기본적으로 (예를 들면 네트워크 지연으로 인한 속도 저하 가능성을 피하기 위해) *Browser* 패널의 디렉터리들을 모니터링하지 않습니다.

**단축화된 데이터 경로**

모든 파일 기반 데이터소스에 대해 단축화된 (localized) 경로를 사용할 수 있습니다. 단축 경로란 데이터소스의 위치를 추출하는 데 사용되는 경로들의 목록을 말합니다. 예를 들어 단축 경로 목록에 `C:\my_maps` 가 존재하는 경우, 데이터소스가 `C:\my_maps\my_country\ortho.tif` 인 레이어는 프로젝트에 `localized:my_country\ortho.tif` 경로로 저장될 것입니다.

이 경로들은 선호도 순서대로 목록화됩니다. 다시 말해 QGIS 가 먼저 첫 번째 경로에서 파일을 찾아본 다음 두 번째 경로로, 다시 세 번째…경로로 넘어갈 것입니다.

#### 숨은 탐색기 경로

이 위젯은 사용자가탐색기 패널 에서 숨기기로 선택한 모든 폴더 목록을 표시합니다. 이 목록에서 폴더를 제거하면 *Browser* 패널에 표시될 것입니다.

### GDAL 설정

GDAL 은 수많은 벡터 및 래스터 포맷을 지원하는 지리공간 데이터 교환 라이브러리입니다. GDAL 라이브러리는 데이터를 지원 포맷들로 읽기 그리고 (대부분) 쓰기 할 수 있는 드라이버들을 제공합니다. *GDAL* 탭은 래스터 및 벡터 포맷 용 드라이버를 그 기능과 함께 보여줍니다.

#### GDAL 래스터 및 벡터 드라이버

*Raster Drivers* 및 *Vector Drivers* 탭에서 파일을 읽고 그리고/또는 쓸 수 있도록 어떤 GDAL 드라이버를 활성화할지를 정의할 수 있습니다. 사용할 수 있는 GDAL 드라이버가 하나 이상일 수도 있는 경우가 있기 때문입니다.

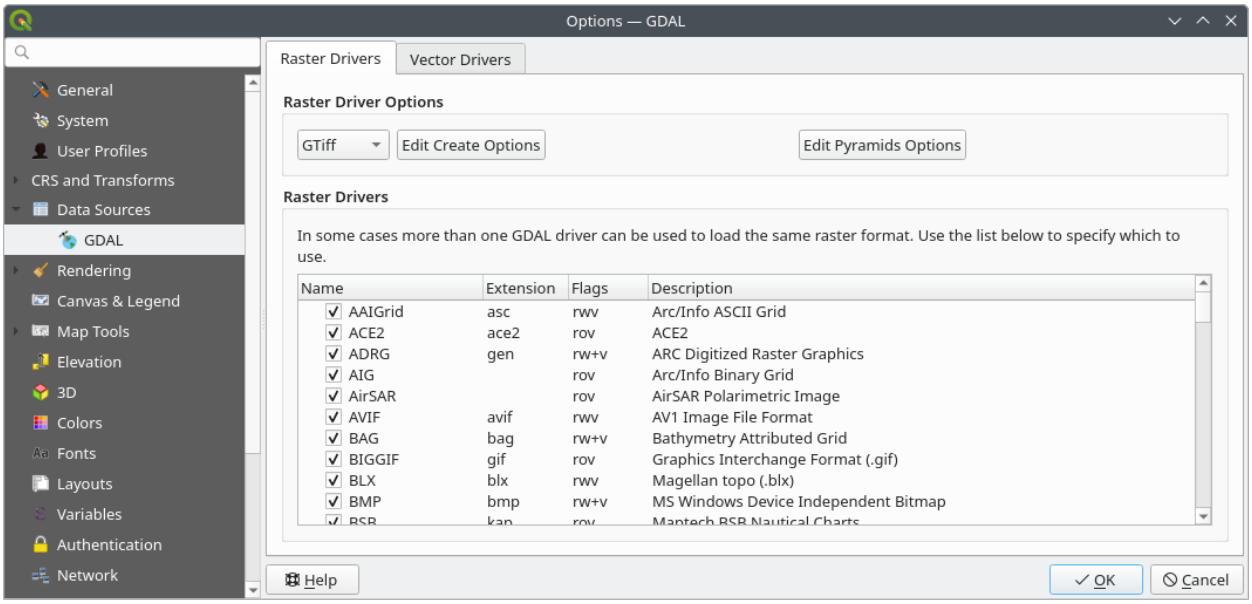



그림 5.7: GDAL 설정하기 - 래스터 드라이버

> **팁**
>
> 읽기와 쓰기 권한 (`rw+(v)`) 에 접근할 수 있는 래스터 드라이버를 더블클릭하면 드라이버를 사용자 지정할 수 있는생성 옵션 편집 대화창이 열립니다.

#### 래스터 드라이버 옵션

이 프레임은 읽기 및 쓰기 권한 접근을 지원하는 래스터 드라이버들의 습성을 사용자 지정할 수 있는 방법을 제공합니다:

- *Edit create options*: allows you to edit or add different profiles of file transformation, i.e., a set of predefined combinations of parameters (type and level of compression, BIGTIFF support, blocks size, overview, colorimetry, alpha ⋯) to use when outputting raster files. The parameters depend on the driver.

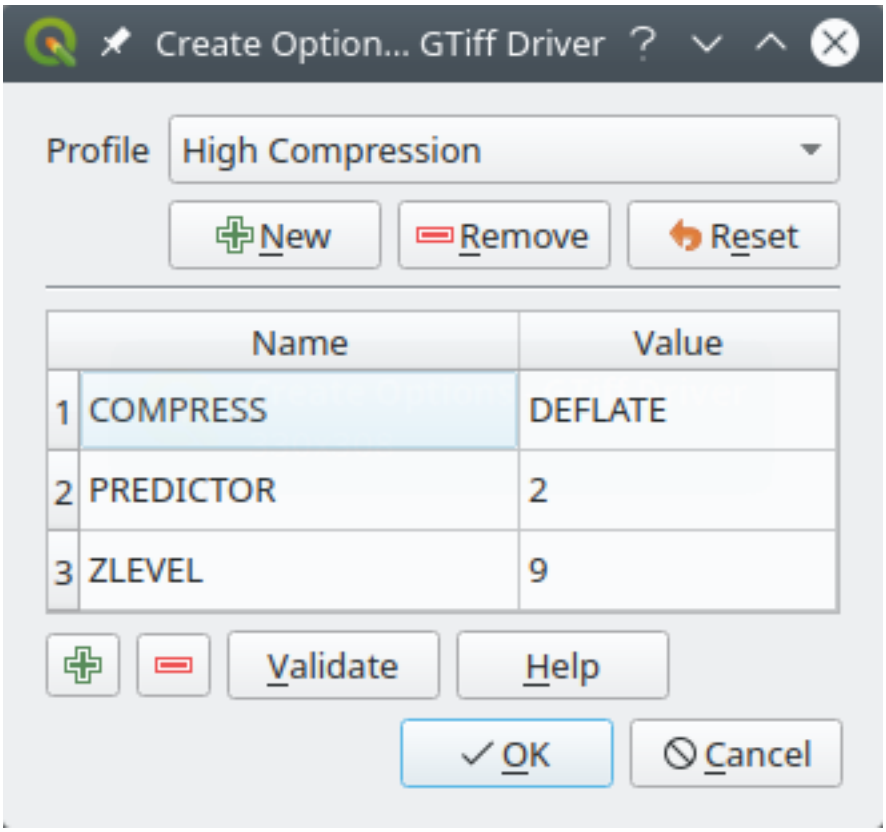



그림 5.8: 옵션 프로파일 생성 (GeoTiff 용) 예시

대화창의 상단 부분은 현재 프로파일 목록을 표시하며, 사용자가 프로파일을 새로 추가하거나 목록에서 제거할 수 있습니다. 사용자가 프로파일의 파라미터를 변경했다면 파라미터의 기본값으로 프로파일을 초기화할 수도 있습니다. GeoTiff 같은 몇몇 드라이버는 사용자가 작업할 수 있는 예시 프로파일을 제공합니다.

대화창 하단 부분에서는:

- 버튼으로 파라미터명과 그 값을 입력해서 행을 추가할 수 있습니다.
- 버튼으로 선택한 파라미터를 삭제할 수 있습니다.
- *Validate* 버튼을 클릭하면 해당 포맷에 대해 입력한 생성 옵션이 무결한지를 확인합니다.
- 사용할 파라미터를 찾으려면 *Help* 버튼을 클릭하거나, GDAL 래스터 드라이버 문서 를 참조하십시오.

- *Edit Pyramids Options*

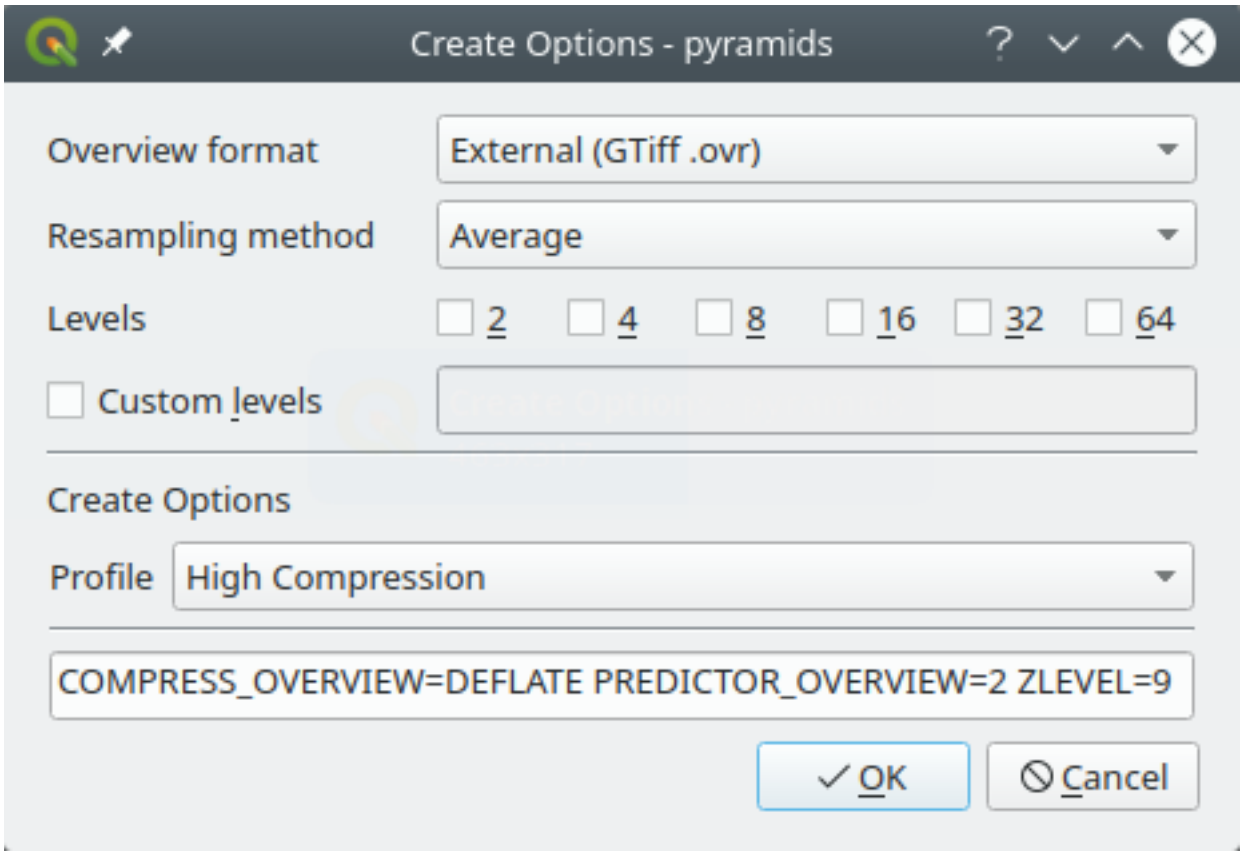



그림 5.9: 피라미드 프로파일 예시

### 5.1.6 렌더링 설정

*Rendering* 탭에서, 맵 캔버스에서의 레이어 렌더링을 조정하기 위한 설정을 정의할 수 있습니다.

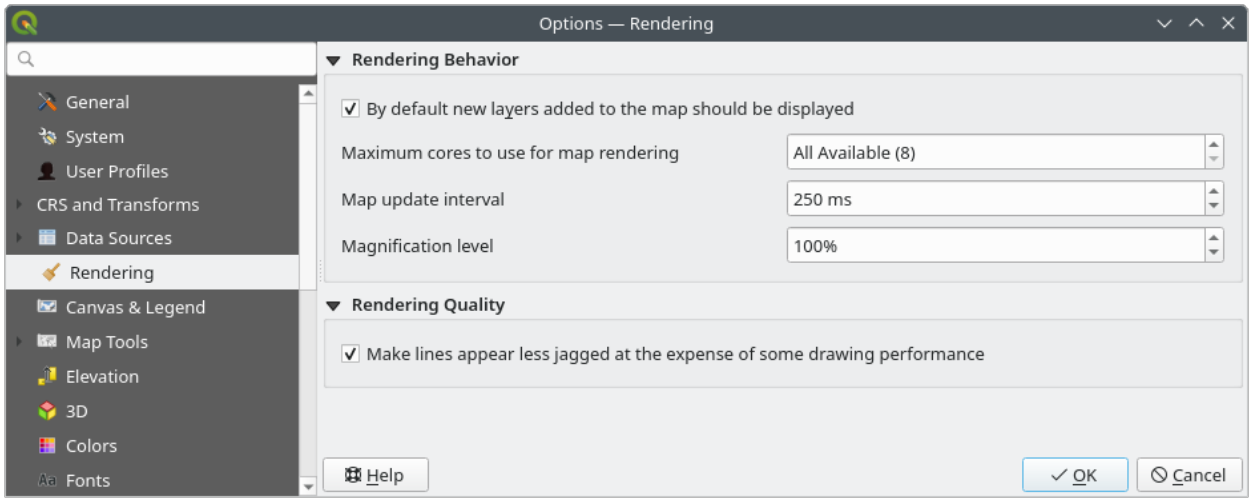



그림 5.10: 렌더링 설정

#### 렌더링 습성

- *By default new layers added to the map should be displayed*: 이 옵션을 체크 해제하는 것은 여러 레이어를 동시에 불러오는 경우 캔버스에 새 레이어들 각각을 동시에 렌더링해서 처리 속도가 느려지는 상황을 피하고자 할 때 유용합니다.
- *Maximum cores to use for map rendering*: 맵 렌더링에 사용할 최대 코어 개수를 설정하십시오.
- 백그라운드에서 맵 캔버스를 별개의 이미지로 렌더링해서 각 *Map update interval* 시간마다 (기본값: 250 밀리초) 이 (오프스크린) 이미지의 내용을 가져다가 맵 캔버스를 업데이트하게 됩니다. 하지만 렌더링 작업이 이 간격보다 빨리 끝나는 경우, 그 즉시 화면에 업데이트됩니다.
- *Magnification level* (돋보기 참조)

#### 렌더링 품질

- *Make lines appear less jagged at the expense of some drawing performance*: 이 옵션은 그리기 성능을 일부 희생해서 라인을 조금 더 부드럽게 표시할 수 있게 해줍니다.

#### 벡터 렌더링 설정

*Vector* 탭에서는 벡터 레이어를 렌더링하기 위한 특정 설정들을 정의할 수 있습니다.

- *Enable Feature Simplification by Default for Newly Added Layers* 를 체크하면, 피처의 도형을 단순화 (노드 감소) 해서 그 결과 피처를 더 빨리 표시합니다. 이때 렌더링 일관성을 해치게 될 수도 있다는 점을 주의하십시오. 다음과 같은 설정을 사용할 수 있습니다:
  - *Simplification threshold (higher values result in more simplification)*: 단순화 한계값을 정의합니다. (값이 클수록 단순화가 심해집니다.)
  - *Simplification algorithm*: 이 옵션은 로컬에서 피처를"실시간 (on-the-fly)"으로 단순화시켜 도형 렌더링 속도를 향상시킵니다. 데이터 제공자에서 불러온 도형을 변경하지는 않습니다. 사용자가 피처 도형을 사용하는 표현식을 사용할 경우 (예를 들어 면적을 계산할 경우) 이 옵션은 중요합니다. 단순화된 도형이

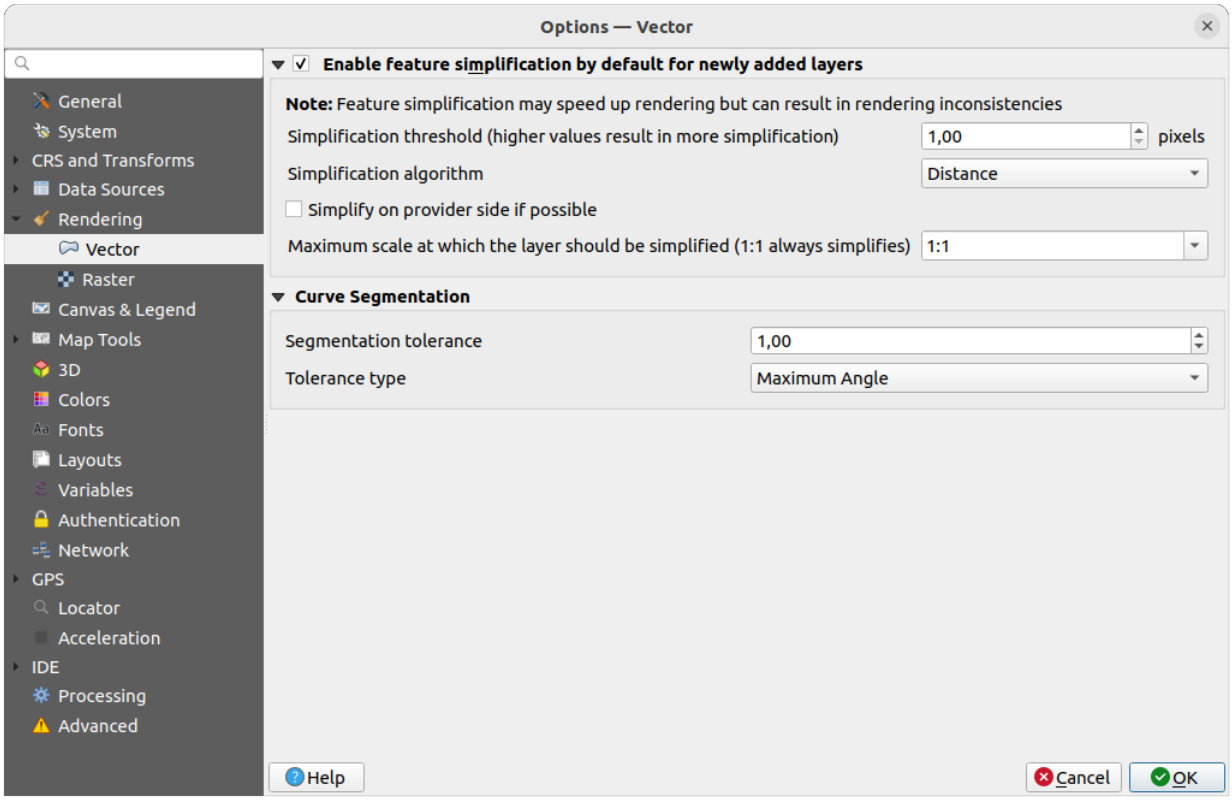



그림 5.11: 벡터 렌더링 설정

아니라 원본 도형에 대해 계산이 이루어진다는 것을 보장하기 때문입니다. 이를 위해 QGIS 는'Distance' (기본값), 'SnapToGrid'그리고'Visvalingam'세 가지 알고리즘을 제공합니다.

- *Simplify on provider side if possible*: the geometries are simplified by the provider (PostgreSQL, Oracle …) and unlike the local-side simplification, geometry-based calculations may be affected
- *Maximum scale at which the layer should be simplified (1:1 always simplifies)*: 레이어를 단순화해야 할 최대 축척을 정의합니다. (1:1 축척의 경우 항상 단순화할 것입니다.)

> **참고**
>
> 전체 수준 설정 이외에도, 특정 레이어의 *Layer properties* → *Rendering* 메뉴를 통해서도 피처 단순화를 설정할 수 있습니다.

- *Curve Segmentation*
  - *Segmentation tolerance*: 이 설정은 원호 (circular arc) 를 렌더링하는 방식을 제어합니다. (연속된 꼭짓점 2 개와 곡선 중점 사이의, 도 단위의) 최대 각도 또는 (두 꼭짓점의 선분과 곡선 사이의, 맵 단위의) 최대 거리 가운데 **더 작은** 쪽, 그리고 **더 직선에 가까운** 선분을 이용해서 렌더링할 것입니다.
  - *Tolerance type*: 근사치와 곡선 사이의 최대 각도 *(Maximum angle)* 또는 최장 거리 *(Maximum difference)* 를 선택할 수 있습니다.

### 래스터 렌더링 설정

*Raster* 탭에서는 벡터 레이어를 렌더링하기 위한 특정 설정들을 정의할 수 있습니다.

*Bands and Resampling* 부분에서:

- *RGB band selection* 을 이용해서 적색, 녹색, 청색 밴드의 번호를 정의할 수 있습니다.
- *Zoomed in resampling* 과 *Zoomed out resampling* 을 정의할 수 있습니다. *Zoomed in resampling* 의 경우 다음 'Nearest neighbour', 'Bilinear (2x2 kernel)'와'Cubic (4x4 kernel)'세 가지 리샘플링 메소드 가운데 하나를 선택할 수 있습니다. *Zoomed out resampling* 의 경우'Nearest Neighbour'와'Average'가운데 하나를 선택할 수 있습니다. *Oversampling* 값도 설정할 수 있습니다. (0.0 ~ 99.99 - 값이 커질수록 QGIS 작업량이 많아집니다. 기본값은 2.0 입니다.)
- *Early resampling*: 이 옵션을 체크하면 소스의 해상도가 밝혀져 있는 제공자 수준에서 래스터 렌더링을 계산하기 때문에, 확대 렌더링 시 QGIS 사용자 지정 스타일 작업을 더 잘 반영하게 됩니다. 타일 래스터를해석 *(interpretation)* 메소드 를 통해 불러온 경우 이 옵션이 매우 유용합니다. 이 옵션은 레이어 수준에서도 (*Symbology* 속성) 설정할 수 있습니다.

*Single band gray*, *Multi band color (byte/band)* 또는 *Multi band color (>byte/band)* 에 *Contrast Enhancement* 옵션을 적용할 수 있습니다. 각 항목에 대해 다음 옵션을 설정할 수 있습니다:

- *Algorithm*: 사용할 알고리즘을 설정합니다. 'No stretch', 'Stretch to MinMax', 'Stretch and Clip to MinMax'또는 'Clip to MinMax'가운데 하나를 선택할 수 있습니다.
- *Limits (minimum/maximum)*: 적용할 제한을 설정합니다. 'Cumulative pixel count cut', 'Minimum/Maximum', 'Mean +/- standard deviation'가운데 하나를 선택할 수 있습니다.

*Contrast Enhancement* 옵션은 다음도 포함합니다:

- *Cumulative pixel count cut limits*: 누적 픽셀 개수 절단 제한값을 정의합니다.
- *Standard deviation multiplier*: 표준 편차 배수를 정의합니다.

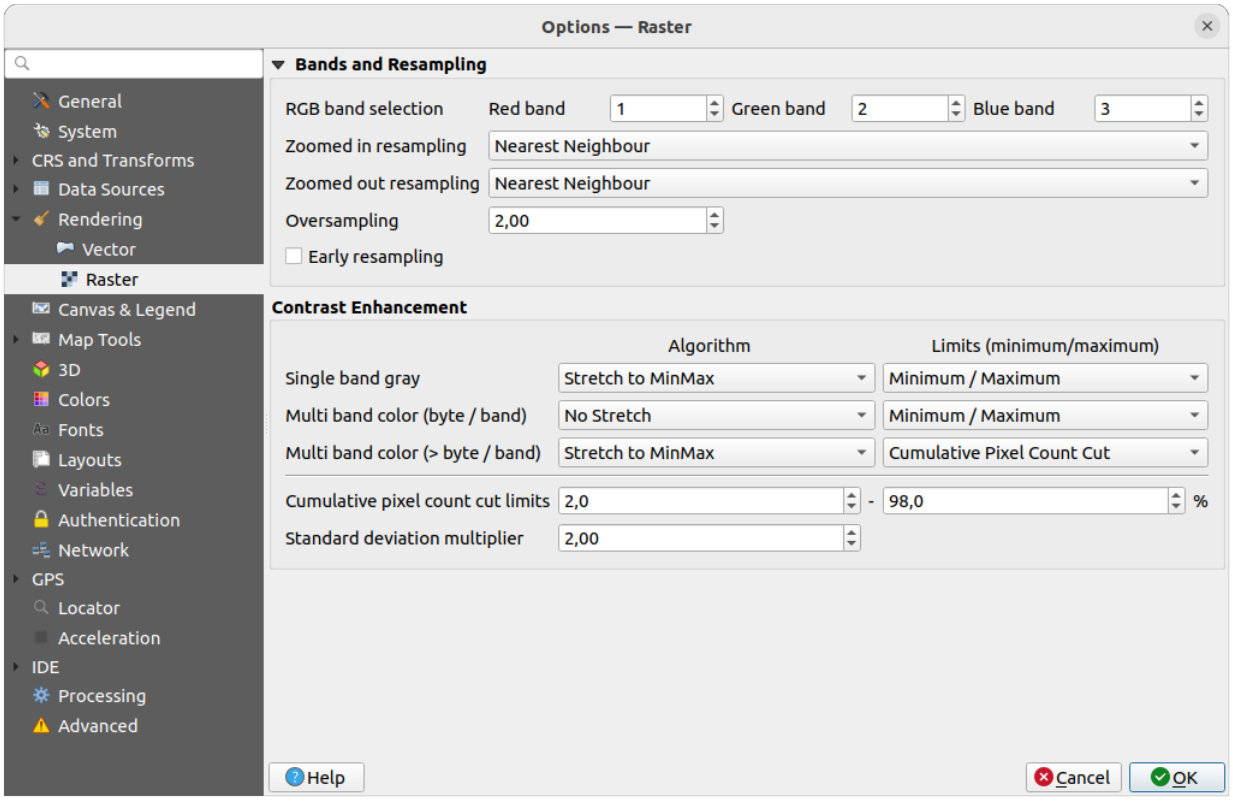



그림 5.12: 래스터 렌더링 설정

### 5.1.7 캔버스 및 범례 설정

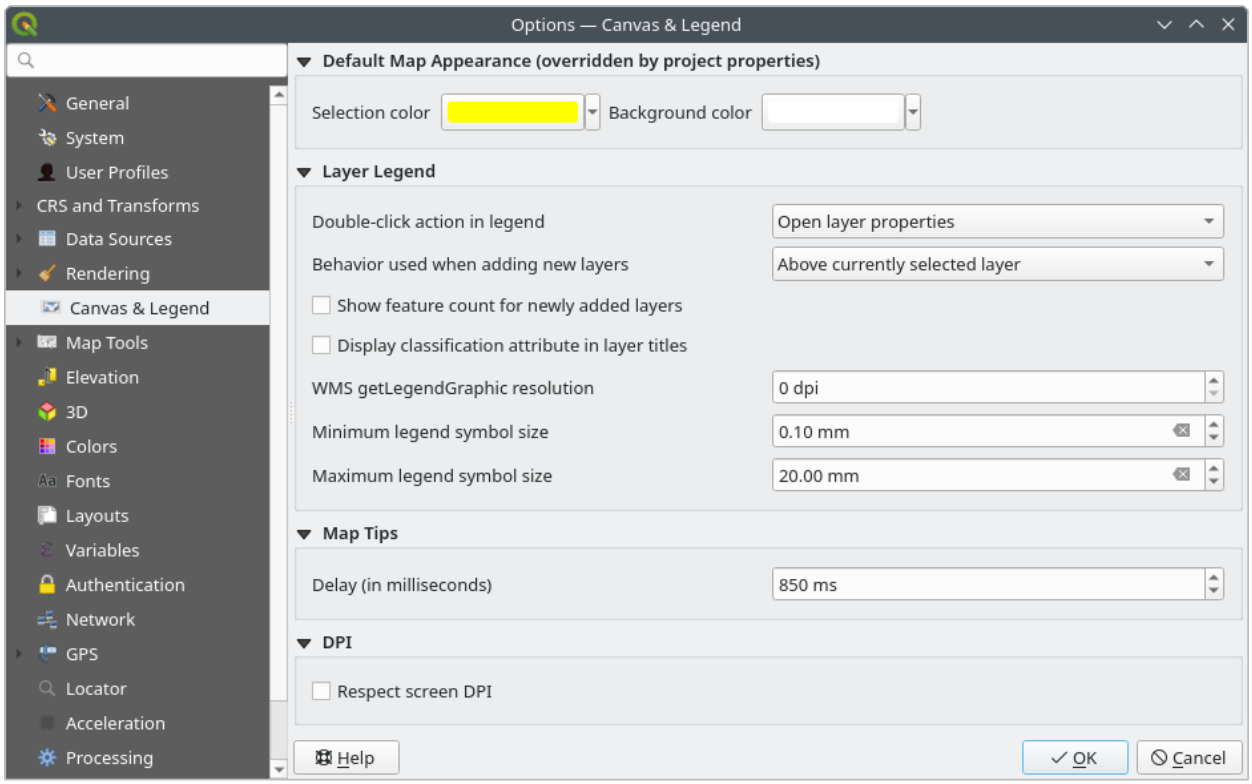



그림 5.13: 캔버스 및 범례 설정

다음 속성을 설정할 수 있습니다:

- **(프로젝트 속성으로 무시되는) 기본 맵 모양**: *Selection color* 및 *Background color* 를 정의합니다.
- **레이어 범례** 상호작용:
  - *Double click action in legend*: 레이어를 더블클릭하는 경우 레이어 설정을 열지 *Open layer properties* (기본값), 속성 테이블을 열지 *Open attribute table*, 또는 레이어 스타일 작업 도크를 열지 *Open layer styling dock* 선택할 수 있습니다.
  - *Behavior used when adding new layers*: QGIS 에 레이어를 불러왔을 때 *Layers* 패널 어디에 레이어를 배치할지 결정할 수 있습니다:
    - *Above currently selected layer*: 현재 선택된 레이어 위
    - *Always on top of the layer tree*: 항상 레이어 트리 최상단에
    - *Optimal index within current layer tree group*: 새 레이어들 간에 새 레이어들을 정렬하고 원하는 위치에 스택 (stack) 으로 배치하는 다른 옵션들과는 달리, 이 옵션은 정렬 작업 논리를 전체 레이어 트리로 (또는 활성화된 그룹으로) 확장시켜 포인트 레이어는 포인트 레이어들의 최상단에, 그 다음 라인 레이어는 라인 데리어들의 최상단에, 그 다음 폴리곤 레이어는 폴리곤 레이어들의 최상단에 자리하도록 하는"최적"의 방식으로 새 레이어들을 삽입시킵니다.
  - ☐ *Show feature count for newly added layers*: 이 옵션을 체크하면 *Layers* 패널의 레이어 이름 옆에 피처 개수를 표시합니다. 범주 (class) 가 존재하는 경우 범주의 피처 개수도 표시할 것입니다. 레이어를 오른쪽 클릭하면 이 피처 개수를 켜고 끌 수 있습니다.

  - *Display classification attribute names*: 예를 들어 범주화 또는 규칙 기반 렌더링 작업자를 적용할 경우 레이어 패널에 범주 속성 명칭을 표시할지 여부를 정의합니다. (자세한 내용은심볼 속성 을 참조하세요.)
  - *WMS getLegendGraphic Resolution* 을 설정할 수 있습니다.
  - *Minimum* 및 *Maximum legend symbol size*: *Layers* 패널에 표시되는 심볼의 크기를 조정할 수 있습니다.
- *Delay*: 레이어의맵 팁 표시를 1,000 분의 1 초 단위로 지연시킬 수 있습니다.
- *Respect screen DPI*: QGIS 가 모니터의 DPI 를 존중할지 여부를 정의합니다. 이 옵션을 활성화하는 경우, QGIS 가 스크린 상에 맵 캔버스를 모니터의 물리적 DPI 에 따라 물리적으로 정확한 축척으로 표시하려 시도할 것입니다. 지정 디스플레이 크기를 가진 심볼도 정확하게 렌더링될 것입니다. 예를 들어 10 밀리미터 크기의 심볼은 모니터 상에 10 밀리미터로 표시될 것입니다. 하지만 맵 캔버스 상의 라벨 글꼴 크기는 QGIS 의 UI 또는 다른 응용 프로그램의 UI 에서의 글꼴 크기와 달라질 수도 있습니다. 이 옵션을 체크 해제하는 경우, QGIS 는 OS 의 논리적 DPI 를 따를 것입니다. 이 경우 시스템 상의 다른 응용 프로그램들의 DPI 와 일치할 것입니다. 하지만 스크린 상에서 맵 캔버스 축척과 심볼 크기가 물리적으로 정확하지 않을 수도 있습니다. 특히 고해상도 모니터의 경우 심볼이 아주 작게 표시될 가능성이 큽니다.

  최적의 경험을 원한다면, 특히 모니터를 여러 개 또는 서로 다른 모니터들을 사용하면서 시각적으로 고품질의 맵을 준비하는 경우, *Respect screen DPI* 옵션을 활성화하는 것을 권장합니다. *Respect screen DPI* 옵션을 비활성화하면, 스크린 상에서만 사용할 목적으로 맵을 생성하는 경우–특히 글꼴 크기가 다른 응용 프로그램들과 일치해야 하는 경우에 더 적합한 산출물을 생성할 것입니다.

> **참고**
>
> 조판 (layout) 으로 렌더링하는 작업은 *Respect screen DPI* 설정의 영향을 받지 않습니다. 이 작업은 언제나 대상 산출 기기의 지정된 DPI 를 따릅니다. 이 설정은 OS 가 알려주는 물리적 스크린 DPI 를 사용한다는 사실 또한 기억해두십시오. 이 DPI 는 모든 디스플레이에 대해 정확하지 않을 수도 있습니다.

### 5.1.8 맵 도구 설정

**Measurements, Coordinate and Bearing Display**

This section provides ways to configure default units parameters:

- *Decimal places* 를 정의합니다.
- *Keep base unit* 을 체크하면 자동적으로 큰 숫자로 (예를 들어 미터를 킬로미터로) 변환하는 걸 막아줍니다.
- *Preferred angle units*: '도', '라디안', '곤/그라디안 (Gon/gradians)', 'Minutes of arc', 'Seconds of arc', 'Turns/revolutions', '밀리라디안 (milliradian, SI 정의)'또는'밀 (mil, NATO/군사 정의)'가운데 하나를 선택합니다.
- *Default distance units for new projects*: options are 'Meters', 'Kilometers', 'Feet', 'Yards', 'Miles', 'Nautical Miles', 'Centimeters', 'Millimeters', 'Inches', 'Degrees'or 'Map Units'(automatically matches the project's CRS units)
- *Preferred area units for new projects*: options are 'Square meters', 'Square kilometers', 'Square feet', 'Square yards', 'Square miles', 'Hectares', 'Acres', 'Square nautical miles', 'Square centimeters', 'Square millimeters', 'Square inches', 'Square degrees'or 'Map Units'(automatically matches the project's CRS units)
- *Default coordinate format for new projects*: 새 프로젝트의 기본 좌표 서식을 정의합니다. 이 서식은 QGIS 상태 바의 *Coordinates* 란 그리고 Identify features 도구의 결과물 가운데 *Derived* 부분에 표시됩니다.
- *Default bearing format for new projects*: 새 프로젝트의 기본 방향 서식을 정의합니다. 이 서식은 상태 바에 맵 캔버스 이동 방향으로 그리고 Measure bearing 도구의 결과물로 표시됩니다.

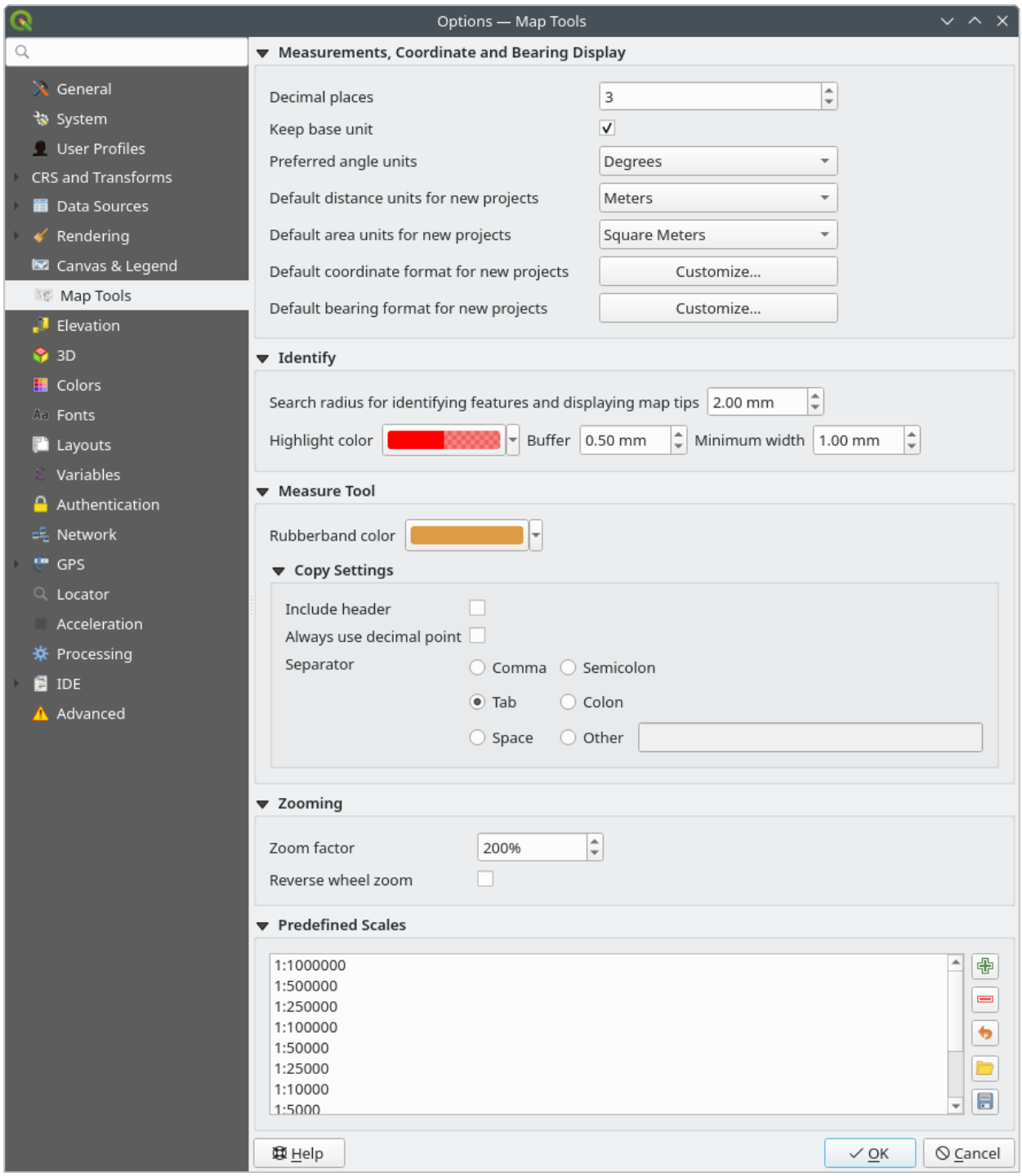

Options — Map Tools
General
System
User Profiles
CRS and Transforms
Data Sources
Rendering
Canvas & Legend
Map Tools
Elevation
3D
Colors
Fonts
Layouts
Variables
Authentication
Network
GPS
Locator
Acceleration
Processing
IDE
Advanced
Measurements, Coordinate and Bearing Display
Decimal places
3
Keep base unit
Preferred angle units
Degrees
Default distance units for new projects
Meters
Default area units for new projects
Square Meters
Default coordinate format for new projects
Customize...
Default bearing format for new projects
Customize...
Identify
Search radius for identifying features and displaying map tips
2.00 mm
Highlight color
Buffer
0.50 mm
Minimum width
1.00 mm
Measure Tool
Rubberband color
Copy Settings
Include header
Always use decimal point
Separator
Comma
Semicolon
Tab
Colon
Space
Other
Zooming
Zoom factor
200%
Reverse wheel zoom
Predefined Scales
1:1000000
1:500000
1:250000
1:100000
1:50000
1:25000
1:10000
1:5000
Help
OK
Cancel

그림 5.14: 맵 도구 설정

The global options can be overridden at the project level, *here* for measurement units and *here* for coordinate and bearing formatting.

**Identify tool**

This group offers some options regarding the behavior of the *Identify tool*.

- *Search radius for identifying features and displaying map tips* 는 사용자가 해당 허용 오차 내에서 클릭하는 한 식별 도구가 결과를 표시하게 될 허용 오차 거리입니다.
- *Highlight color* 에서 식별된 피처를 어떤 색상으로 강조할지 선택할 수 있습니다.
- *Buffer* 에서 식별 강조 윤곽선을 어느 정도의 버퍼 거리로 렌더링할지 설정합니다.
- *Minimum width* 에서 강조된 객체의 윤곽선을 얼마나 굵게 할지 설정합니다.

**측정 도구**

- 측정 도구 용 *Rubberband color* 를 정의합니다.

**측정 도구 복사 설정**

이 설정을 통해 Measure Line 창에 있는 *Copy* 버튼의 습성을 제어할 수 있습니다. 열 이름을 유지시키려면 *Include header* 를 선택하면 되고, 구분자를 사용자 지정하려면 *Separator* 도 선택할 수 있습니다. 사용자의 좌표를 클립보드로 복사할 때, QGIS 옵션에 있는 사용자 언어 설정에서 쉼표를 소수점 구분자로 사용하도록 설정되어 있더라도, 마침표를 소수점 구분자로 사용하려면 *Always use decimal point* 를 선택하면 됩니다. 이 경우 *Always use decimal point* 옵션을 체크하지 않으면 쉼표를 필드 구분자로 선택할 수 없게 될 것입니다.

**이동 및 확대/축소**

- 확대/축소 도구 또는 마우스 휠 용 *Zoom factor* 를 정의합니다.
- *Reverse wheel zoom* 옵션으로 마우스 휠의 스크롤 방향을 조정할 수 있습니다.

**사전 정의 축척**

이 부분에서는 상태 바의 *Scale* 란, 가시성 축척 선택기, 또는 보조 2 차원 맵 뷰 설정 등과 같은 축척 관련 드롭다운 위젯에 기본적으로 표시되는 사전 정의 축척 목록을 볼 수 있습니다. 사용자는 및 버튼을 통해 사용자 지정 축척을 추가 및 제거할 수 있습니다. 또한 이 축척들을 `.XML` 파일로 내보내거나, 외부 `.XML` 파일을 가져올 수도 있습니다. 어떤 경우에도 사용자가 변경한 내용을 제거하고 사전 정의 목록으로 리셋할 수 있다는 사실을 기억하십시오.

사용자는 프로젝트 속성 대화창에서 위젯의 전체 수준 축척 목록을 무시하고 사용자 자신만의 축척 목록을 설정할 수도 있습니다.

### 디지타이즈 작업 설정

이 탭에서벡터 레이어 편집 시 (속성 및 도형) 이용할 수 있는 일반 설정의 환경을 설정할 수 있습니다.

**피처 생성**

- *Suppress attribute form pop-up after feature creation*: 이 옵션을 활성화하더라도, 각 레이어의 속성 대화창에서 무시할 수 있습니다.
- *Reuse last entered attribute values*: 마지막으로 입력한 속성값들을 재사용할지 여부를 정의합니다. 이 옵션을 체크하면 마지막으로 사용한 모든 속성값들을 기억해서 다음으로 디지타이즈하는 피처의 기본값으로 사용합니다. 이 옵션은 레이어 별로 작동합니다. 이 속성을 필드 별 기반으로 (필드 습성 환경 설정 참조) 조정할 수도 있습니다.

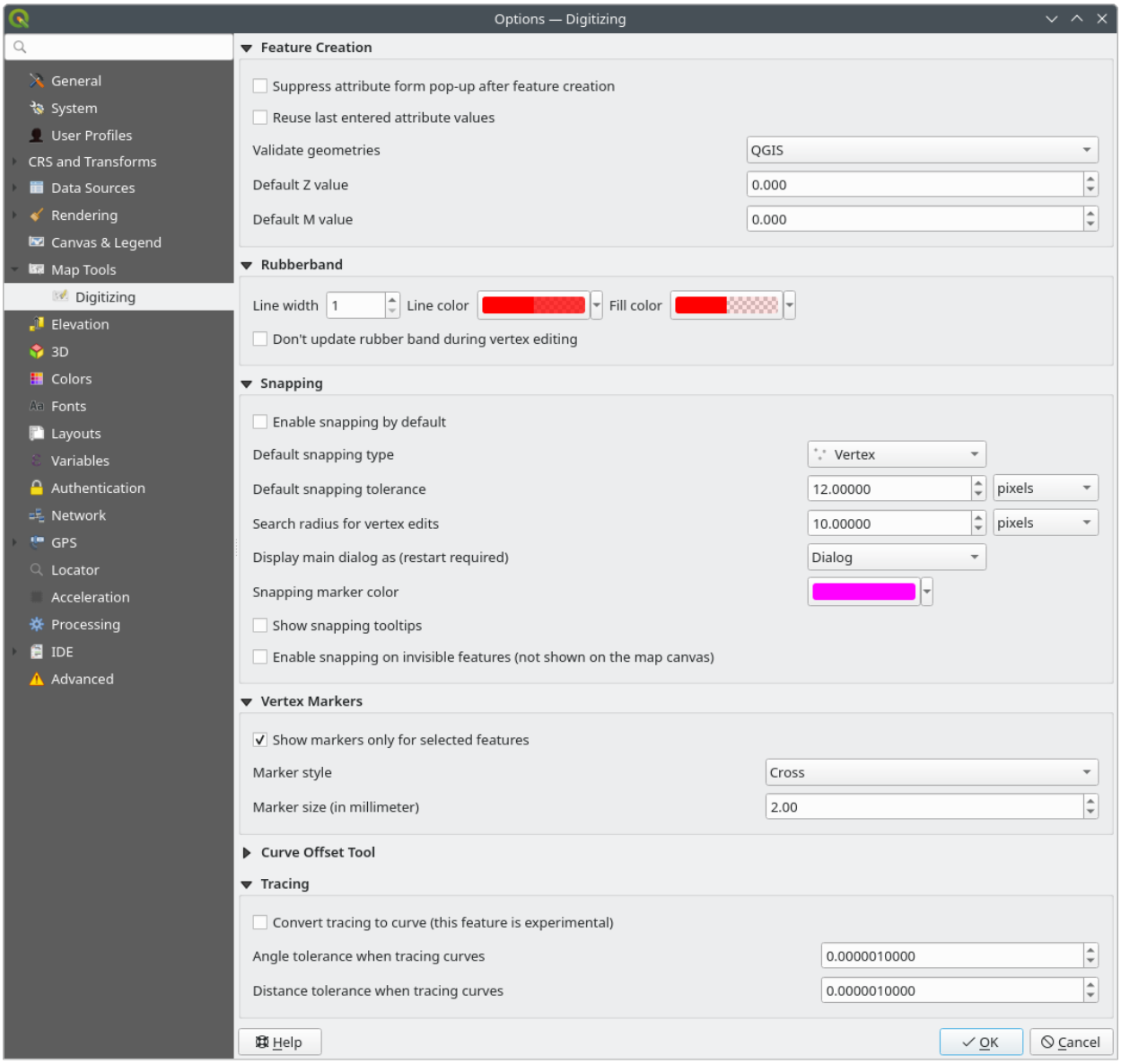



그림 5.15: 디지타이즈 작업 설정

- *Validate geometries*. Editing complex lines and polygons with many nodes can result in very slow rendering. This is because the default validation procedures in QGIS can take a lot of time. To speed up rendering, it is possible to select GEOS geometry validation or to switch it off. GEOS geometry validation is much faster, but the disadvantage is that only the first geometry problem will be reported.

  선택 집합에 따라 도형 오류에 대한 보고도 달라질 수 있다는 점을 기억하십시오. (오류 메시지의 유형 및 그 의미 참조)

- *Default Z value*: 새 3 차원 피처를 생성할 때 사용할 기본 Z 값을 정의합니다.

고무줄

- 고무줄의 *Line width*, *Line color* 와 *Fill color* 를 정의합니다.
- *Don't update rubberband during vertex editing*: 꼭짓점 편집 시 고무줄을 표시하지 않습니다.

스냅

- *Enable snapping by default*: 프로젝트를 열 때부터 스냅 작업을 활성화합니다.
- *Default snap mode* : '꼭짓점', '선분', '중심', '선분 중앙점', '라인 종단점', '면'가운데에서 기본 스냅 모드를 정의합니다.
- *Default snapping tolerance*: 스냅 작업 기본 허용 오차를 정의합니다. 맵 단위 또는 픽셀 개수로 설정할 수 있습니다.
- *Search radius for vertex edits*: 꼭짓점 편집 시 검색 반경을 정의합니다. 맵 단위 또는 픽셀 개수로 설정할 수 있습니다.
- *Display main dialog as (restart required)*: 고급 스냅 작업 대화창을'대화창'또는'독'가운데 어느쪽으로 표시할지 설정합니다.
- *Snapping marker color*: 스냅 작업 마커의 색상을 설정합니다.
- *Show snapping tooltips*: 예를 들어 사용자가 스냅하고자 하는 피처의 레이어명을 보여주는 툴팁의 표시 여부를 설정합니다. 여러 피처가 중첩돼 있을 경우 유용합니다.
- *Enable snapping on invisible features (not shown on the map canvas)*: 이 설정을 활성화하면 맵 캔버스 상에 표시되지 않는 가시화되지 않은 피처에도 스냅 작업을 할 수 있게 해줍니다.

꼭짓점 마커

- *Show markers only for selected features*: 선택한 피처에만 마커를 표시합니다.
- 꼭짓점 *Marker style* 을 정의합니다. 'Cross'(기본값), 'Semi transparent circle'또는'None'가운데 하나를 선택할 수 있습니다.
- 꼭짓점 *Marker size (in millimeter)* 를 밀리미터 단위로 정의합니다.

곡선 오프셋 도구

다음 옵션들은고급 디지타이즈 작업 의 Offset Curve 에서 쓰입니다. 다양한 설정을 통해, 라인 오프셋의 형태를 변경할 수 있습니다.

- *Join style*: 라인에 있는 모서리를 오프셋시킬 때'Round', 'Miter'또는'Bevel'결합을 사용할지 여부를 지정합니다.
- *Quadrant segments*: 둥글린 (rounded) 오프셋을 생성하는 경우 사분원에 가깝게 만드는 데 사용할 라인 선분의 개수를 제어합니다.
- *Miter limit*: 마이터 결합을 생성하는 경우 사용할 오프셋 도형으로부터의 최장 거리를 오프셋 거리의 인자로 설정합니다. (마이터 결합 스타일에만 적용할 수 있습니다)

- *End cap style*: 라인 끝부분을 어떻게 처리할지 제어합니다. 'Round', 'Flat'또는'Square'가운데 하나로 선택할 수 있습니다.

**트레이스 작업**

*Convert tracing to curve* 옵션을 활성화하면, 사용자가 디지타이즈 작업을 하는 도중 곡선 선분을 생성할 수 있습니다. 사용자의 데이터 제공자가 이 기능을 지원해야만 한다는 사실을 기억하십시오.

### 5.1.9 표고 설정

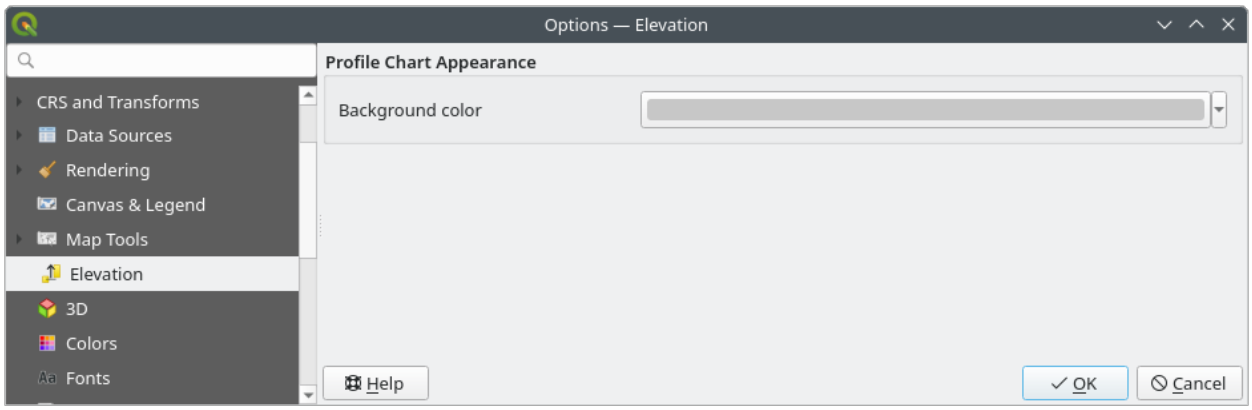



그림 5.16: 표고 설정

*Elevation* 메뉴에서, 표고 단면 용 *Background color* 로 사용할 특정 색상을 설정할 수 있습니다. 이렇게 배경색을 지정하면 예를 들어 RGB 색상이 적용된 점구름처럼 기본 배경색이 포인트 색상과 너무 비슷해서 쉽게 분간이 안 되는 특정 데이터셋의 도표를 좀 더 가독성 있게 만들어줄 수 있습니다. 배경색을 변경하지 않는 경우, 표고 단면은 예전과 마찬가지로 표준 시스템 배경색을 사용해서 표시될 것입니다.

### 5.1.10 3 차원 설정

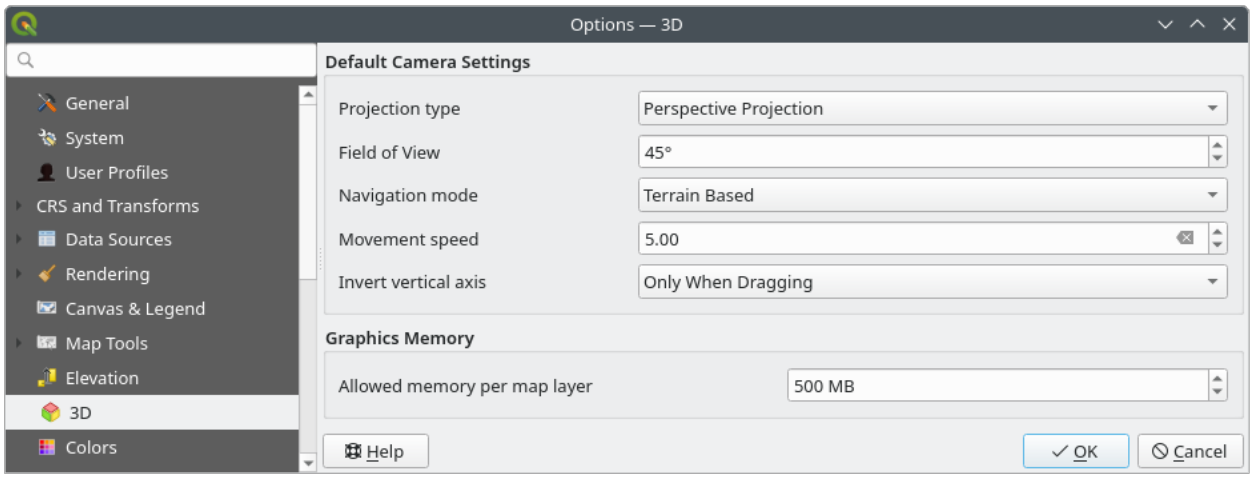



그림 5.17: 3 차원 설정

 *3D* 메뉴를 통해 모든 *3D Map view* 에 사용할 몇몇 기본 설정을 환경 설정할 수 있습니다. 이 설정들은 다음을 참조할 수도 있습니다:

- *Default Camera Settings*:
  - *Projection type*: 3 차원 신 (scene) 을 다음 유형으로 볼 수 있습니다:
    - *Perspective projection* (기본값): 평행선들이 먼 거리에서 만나는 것처럼 보입니다. 객체들이 카메라에서 멀어질수록 점점 작아지는 것으로 보입니다.
    - *Orthogonal projection*: 평행선들이 평행하는 것으로 보입니다. 객체들은 거리에 상관없이 동일한 크기로 보입니다.
  - 카메라의 *Field of view*: 투시 투영 (perspective projection) 을 선택한 경우에만 적용됩니다. 현재 수직 시야 (field of view) 를 도 단위로 지정해서 신이 카메라에 어느 정도 비칠지를 결정합니다. 기본값은 45° 입니다.
  - *Navigation mode*: 3 차원 신과 대화형 작업을 할 수 있는 서로 다른 수단들을 제공합니다. 다음과 같은 모드들을 사용할 수 있습니다:
    - *Terrain based*: 신을 탐색하면, 카메라가 지표 상에 있는 고정 위치 주변을 따라갑니다.
    - *Walk mode (first person)*: 카메라의 1 인칭 시점을 따라갑니다.

    어떤 모드를 선택했는지에 따라탐색 명령어 가 달라집니다.
  - *Movement speed*: 이동 속도를 정의합니다.
  - *Invert vertical axis*: 수직축 이동이 일반 습성과는 반대로 이루어져야 하는지 여부를 정의합니다. *Walk mode* 인 경우에만 이동에 영향을 미칩니다. 다음 가운데 하나로 설정할 수 있습니다:
    - *Never*: 수직축 이동 습성을 변경하지 않습니다.
    - *Only when dragging*: 카메라를 클릭 & 드래그로 회전시키는 경우에만 수직축 이동 습성을 역전시킵니다.
    - *Always*: 카메라를 클릭 & 드래그하는 경우 그리고 카메라 이동을 (~ 키를 눌러서) 커서로 고정시키는 경우 둘 다 수직축 이동 습성을 역전시킵니다.
- *Graphics memory* 부분에 있는 *Allowed memory per layer* 옵션은 사용자가 각 레이어에 대한 GPU 메모리 제한을 설정할 수 있게 해줍니다. 이 옵션은 사용 가능한 GPU 메모리 자원을 소모시키는 대용량 3 차원 신을 활용하는 사용자에게 유용합니다. 메모리 제한을 넘어설 때 경고 메시지도 표시해주기 때문에, 대용량 신에서 발생한 문제를 해결하는 데에도 도움이 될 것입니다.

### 5.1.11 색상 설정

이 메뉴는색상 선택기 위젯 을 통해 응용 프로그램 전반에 걸쳐 사용되는 색상표를 생성하거나 업데이트할 수 있게 해줍니다. 다음과 같은 색상표들을 선택할 수 있습니다:

- 최근 사용한 색상들을 보여주는 *Recent colors*
- 기본 색상표인 *Standard colors*
- 현재 프로젝트에 특정해 사용되는 색상표인 *Project colors* (자세한 내용은스타일 속성 참조)
- QGIS 에 새 레이어를 추가할 경우 기본적으로 사용되는 색상표인 *New layer colors*
- 또는 색상표 콤보박스 옆의 …버튼으로 생성하거나 가져올 수 있는 사용자 지정 색상표 (들)

기본적으로, *Recent colors*, *Standard colors* 및 *Project colors* 색상표는 색상 버튼 드롭다운 목록에 표시되도록 설정돼 있으며 이들을 삭제할 수는 없습니다. *Show in Color Buttons* 옵션을 통해 사용자 지정 색상표도 이 위젯에 추가할 수 있습니다.

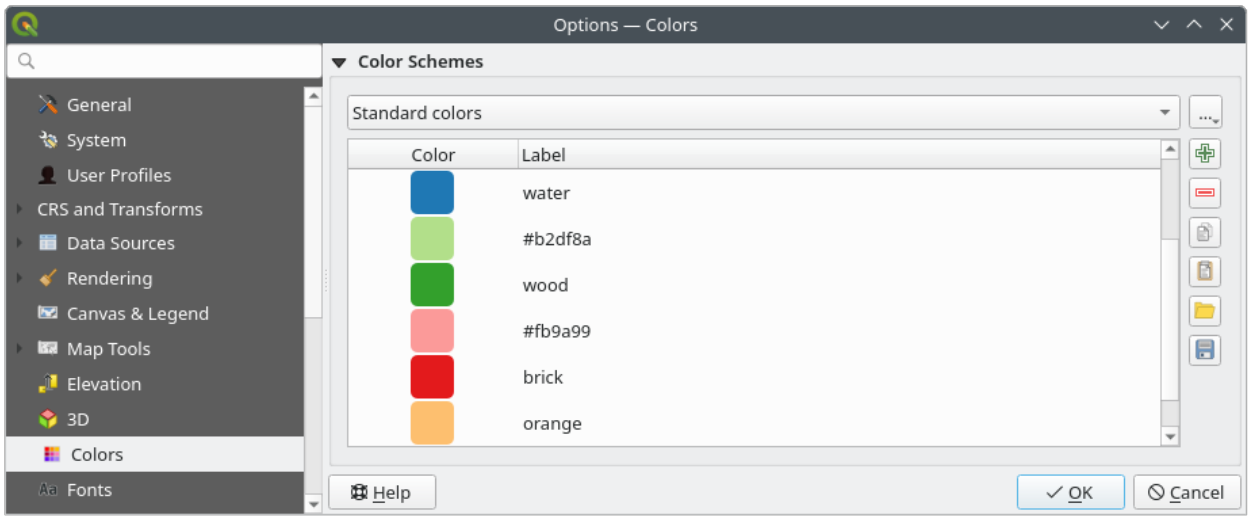



그림 5.18: 색상 설정

프레임 옆의 다음과 같은 도구 모음을 사용해서, 모든 색상표의 색상 목록을 관리할 수 있습니다:

- *Add* 또는 *Remove*: 색상을 추가하거나 삭제합니다.
- *Copy* 또는 *Paste*: 색상을 복사하거나 붙여넣습니다.
- *Import* 또는 *Export*: 색상 모음을 `.gpl` 파일로부터 가져오거나, `.gpl` 파일로 내보냅니다.

목록에 있는 색상을 더블클릭하면색상 선택기 대화창의 해당 색상을 조정하거나 대체할 수 있습니다. *Label* 열을 더블클릭하면 해당 색상의 명칭을 변경할 수도 있습니다.

### 5.1.12 글꼴 설정

*Fonts* 탭에서 프로젝트 전반에 걸쳐 쓰이는 글꼴을 관리할 수 있습니다:

- *Font Replacements*: 프로젝트 또는 스타일을 불러올 때 자동으로 대체할 글꼴 목록을 채울 수 있습니다. 이 설정은 프로젝트 및 스타일이 서로 다른 OS 를 교차하며 사용되는 경우 (예를 들어"Arial"글꼴을"Helvetica" 글꼴로 대체해서) 더 나은 지원을 할 수 있게 해줍니다.
- *User Fonts*: 사용자 프로파일 의 `fonts` 하위 폴더 안에 TTF 또는 OTF 글꼴을 넣을 수 있습니다. QGIS 구동 시 이런 사용자 글꼴을 자동적으로 불러오도록 할 수 있습니다. 기업 환경에서는 OS 수준에서 글꼴을 설치할 수 없는 경우가 많은데, 사용자 글꼴로 등록하면 OS 수준에서 글꼴을 설치하지 않아도 사용할 수 있습니다. 이 패널은 설치된 사용자 글꼴을 모두 보여주며, 기존에 설치된 사용자 글꼴을 관리할 수 (예: 제거할 수) 있습니다.

  *Automatically download missing, freely-licensed fonts*: 누락되었거나 또는 자유 라이선스인 글꼴을 자동으로 다운로드할 수도 있습니다. 예를 들어 현재 사용할 수 없는 글꼴을 참조하는 프로젝트 또는 스타일 또는 벡터 타일 레이어 등을 열거나 불러오는 경우, URL 을 통해 다운로드할 수 있는 자유 라이선스 글꼴들의 하드 코딩된 목록을 참고해서 글꼴을 (글꼴 라이선스 알림을 포함해서) 사용자 프로파일의 글꼴 디렉터리로 자동 다운로드할 수 있는지 여부를 결정합니다.

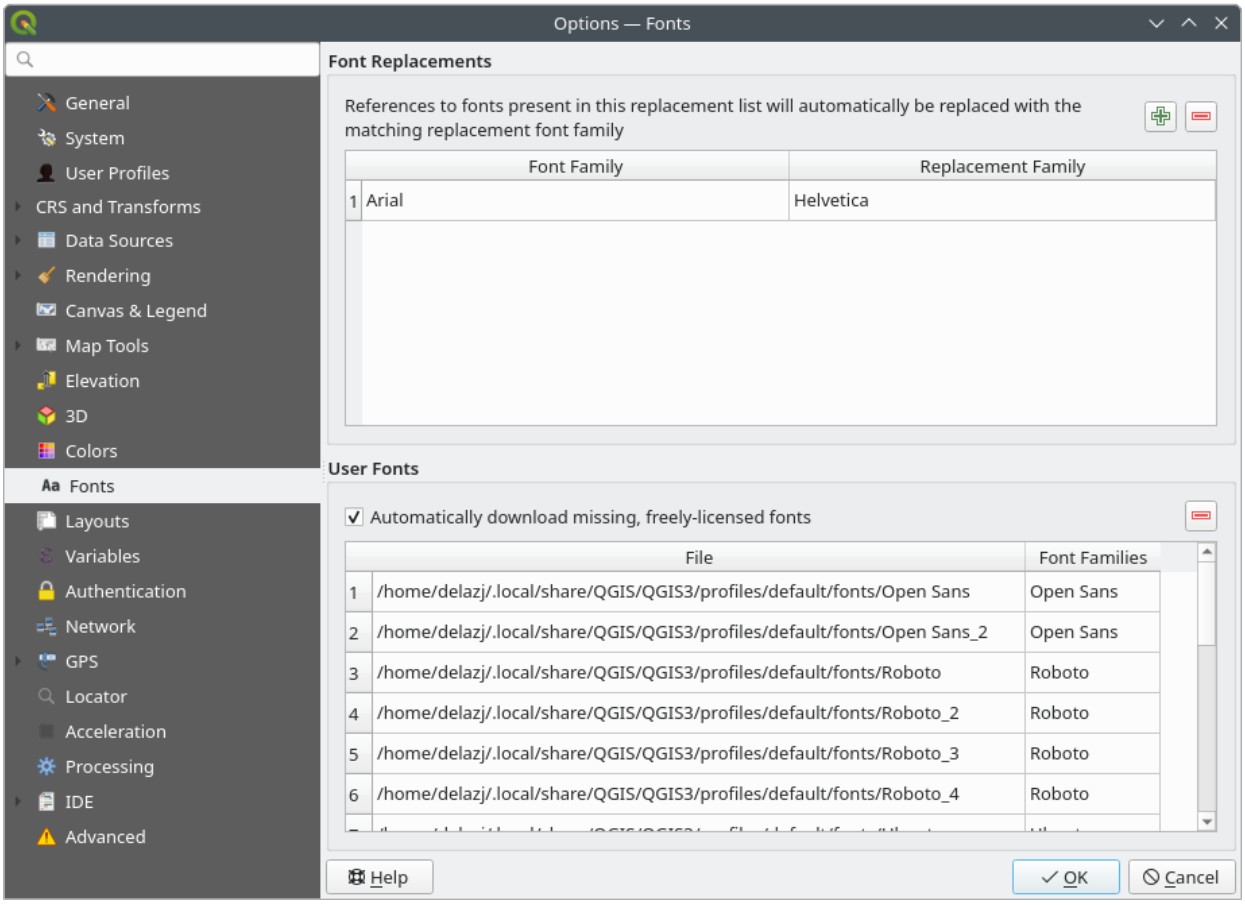

Options — Fonts
General
System
User Profiles
CRS and Transforms
Data Sources
Rendering
Canvas & Legend
Map Tools
Elevation
3D
Colors
Fonts
Layouts
Variables
Authentication
Network
GPS
Locator
Acceleration
Processing
IDE
Advanced
Font Replacements
References to fonts present in this replacement list will automatically be replaced with the matching replacement font family
Font Family
Replacement Family
1 Arial
Helvetica
User Fonts
Automatically download missing, freely-licensed fonts
File
Font Families
1 /home/delazj/.local/share/QGIS/QGIS3/profiles/default/fonts/Open Sans
Open Sans
2 /home/delazj/.local/share/QGIS/QGIS3/profiles/default/fonts/Open Sans_2
Open Sans
3 /home/delazj/.local/share/QGIS/QGIS3/profiles/default/fonts/Roboto
Roboto
4 /home/delazj/.local/share/QGIS/QGIS3/profiles/default/fonts/Roboto_2
Roboto
5 /home/delazj/.local/share/QGIS/QGIS3/profiles/default/fonts/Roboto_3
Roboto
6 /home/delazj/.local/share/QGIS/QGIS3/profiles/default/fonts/Roboto_4
Roboto
Help
OK
Cancel

그림 5.19: 글꼴 설정

### 5.1.13 조판 설정

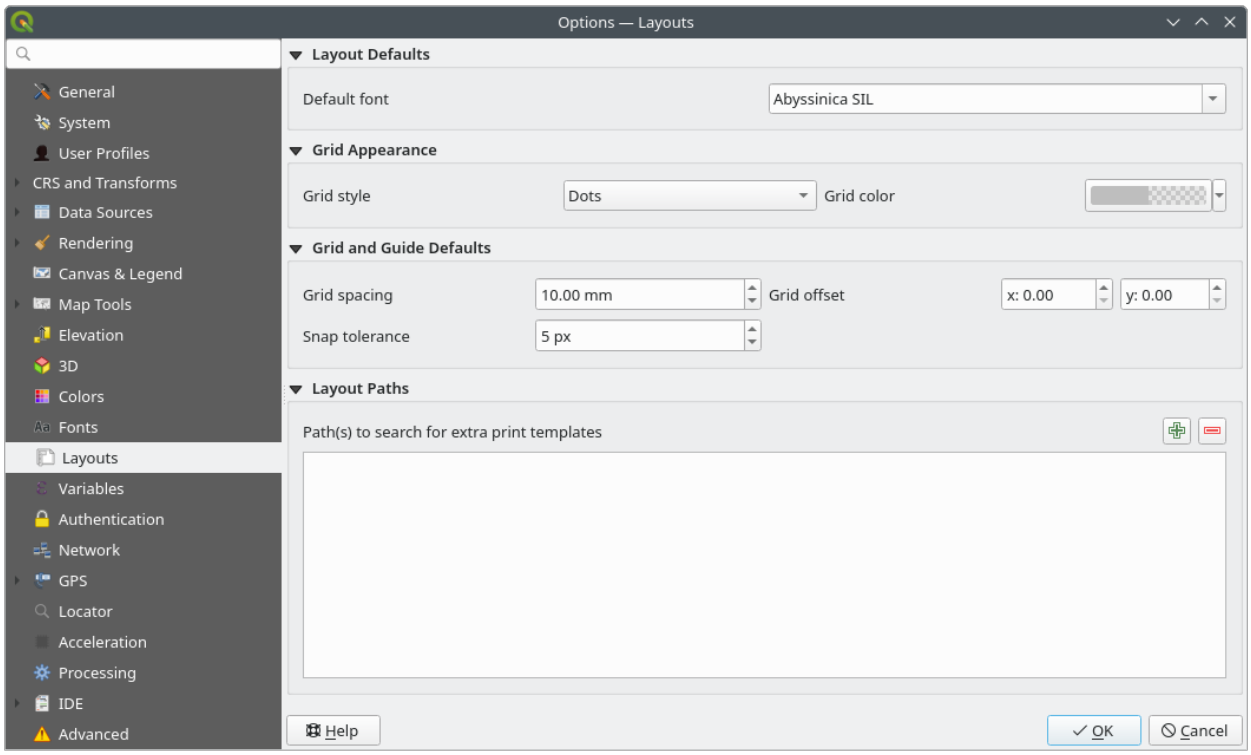



그림 5.20: 조판 설정

**조판 기본값**

인쇄 조판기 내부에서 사용되는 *Default font* 를 정의할 수 있습니다.

**그리드 모양**

- *Grid style*: 그리드 스타일을'Solid', 'Dots', 'Crosses'가운데 하나로 정의합니다.
- *Grid color*: 그리드 색상을 정의합니다.

**그리드 및 가이드 기본값**

- *Grid spacing*: 그리드 간격을 정의합니다.
- *Grid offset*: X 축 및 Y 축에 대한 그리드 오프셋을 정의합니다.
- *Snap tolerance*: 스냅 허용 오차를 정의합니다.

**조판기 경로**

- *Path(s) to search for extra print templates*: 추가 인쇄 템플릿을 검색할 경로를 정의합니다. 새 인쇄 템플릿 생성 시 사용자 정의 조판 템플릿을 담은 폴더 목록을 사용할 수 있습니다.

### 5.1.14 변수 설정

*Variables* 탭은 전체 수준에서 쓸 수 있는 모든 변수들의 목록을 표시합니다.

또한 사용자가 전체 수준 변수를 관리할 수도 있습니다. 버튼을 클릭해서 새 사용자 지정 전체 수준 변수를 추가합니다. 마찬가지로, 목록에서 사용자 지정 전체 수준 변수를 선택한 다음 버튼을 클릭해서 제거합니다. 값을 변수로 저장 에서 변수에 대한 더 자세한 정보를 살펴볼 수 있습니다.

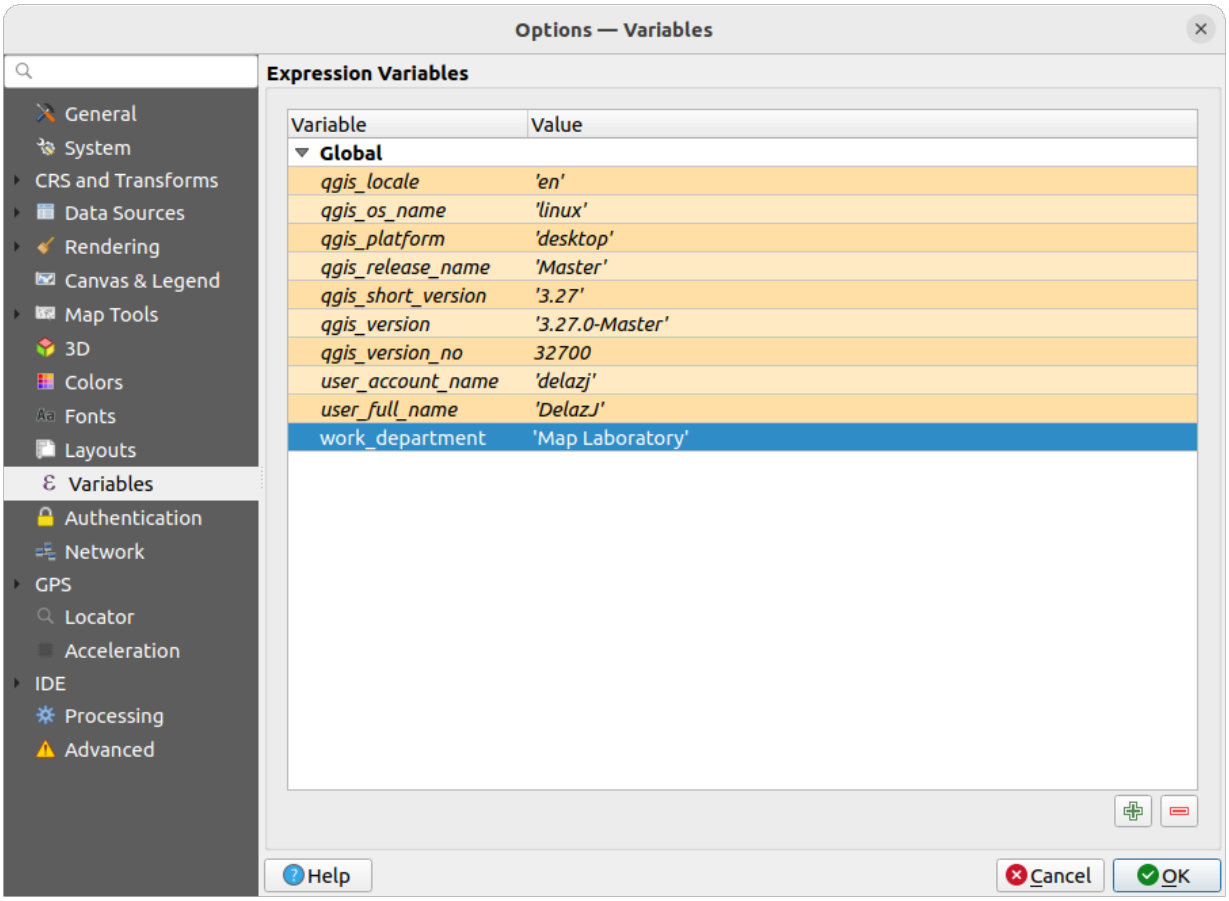



그림 5.21: 변수 설정

### 5.1.15 인증 설정

*Authentication* 탭에서 인증 환경을 설정하고 PKI 인증서를 관리할 수 있습니다. 더 자세한 내용은인증 시스템 을 참조하세요.

인증을 관리하기 위해 프레임 옆에 있는 다음과 같은 도구들을 사용할 수 있습니다:

- Add new authentication configuration: 새로운 인증 환경 설정을 추가합니다.
- Remove selected authentication configuration: 선택한 인증 환경 설정을 제거합니다.
- Edit selected authentication configuration: 선택한 인증 환경 설정을 편집합니다.

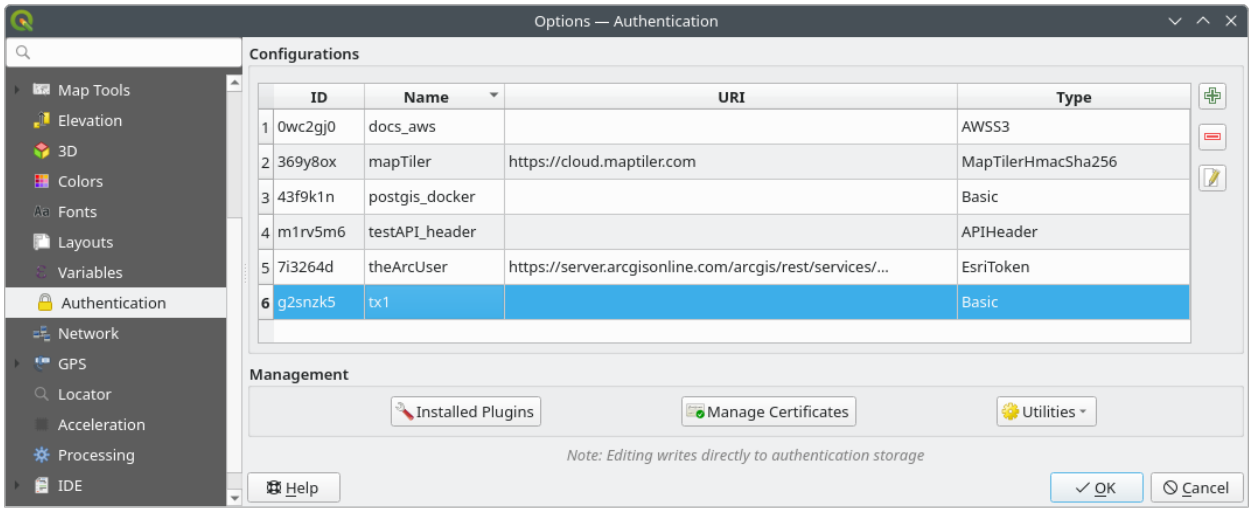



그림 5.22: 인증 설정

### 5.1.16 네트워크 설정

#### 일반

- *Timeout for network requests (ms)*: 네트워크 요청에 대한 타임아웃 시간을 정의합니다. 기본값은 60,000 밀리초입니다.
- *Default expiration period for WMS Capabilities (hours)*: WMS 케이버필리티에 대한 기본 만료 기간을 정의합니다. 기본값은 24 시간입니다.
- *Default expiration period for WMS-C/WMTS tiles (hours)*: WMS-C/WMTS 타일에 대한 기본 만료 기간을 정의합니다. 기본값은 24 시간입니다.
- *Max retry in case of tile or feature request errors*: 타일 또는 피처 요청 오류가 발생하는 경우의 재시도할 최대 횟수를 정의합니다.
- 기본값이 `Mozilla/5.0` 인 *User-Agent prefix* 를 정의합니다. 이 값은 사용자 에이전트를 표현하기 위해, 예를 들면 `Mozilla/5.0 QGIS/33801/Ubuntu 22.04.4 LTS` 처럼 QGIS 버전과 OS 버전 둘 모두의 앞에 붙게 됩니다.

#### 캐시 설정

Defines the *Directory* and a *Size* for the cache specified in megabytes. You can clear the value to enable the *Smart cache size*, which sets the maximum cache size based on available disk space. Also offers tools to *automatically clear the connection authentication cache on SSL errors (recommended)*.

#### 웹 접근 용 프록시

- *Use proxy for web access*
- 사용자의 필요에 따라 *Proxy type* 을 설정하고'Host'와'Port'를 정의하십시오. 다음과 같은 프록시 유형을 선택할 수 있습니다:
  - *Default Proxy*: 시스템의 프록시를 기반으로 프록시를 결정합니다.
  - *Socks5Proxy*: 모든 종류의 연결에 쓸 수 있는 일반 프록시입니다. TCP, UDP, 지정 포트 (들어오는 연결) 그리고 인증을 지원합니다.
  - *HttpProxy*: "CONNECT"명령어로 시행되며, 나가는 TCP 연결만 지원하고, 인증을 지원합니다.

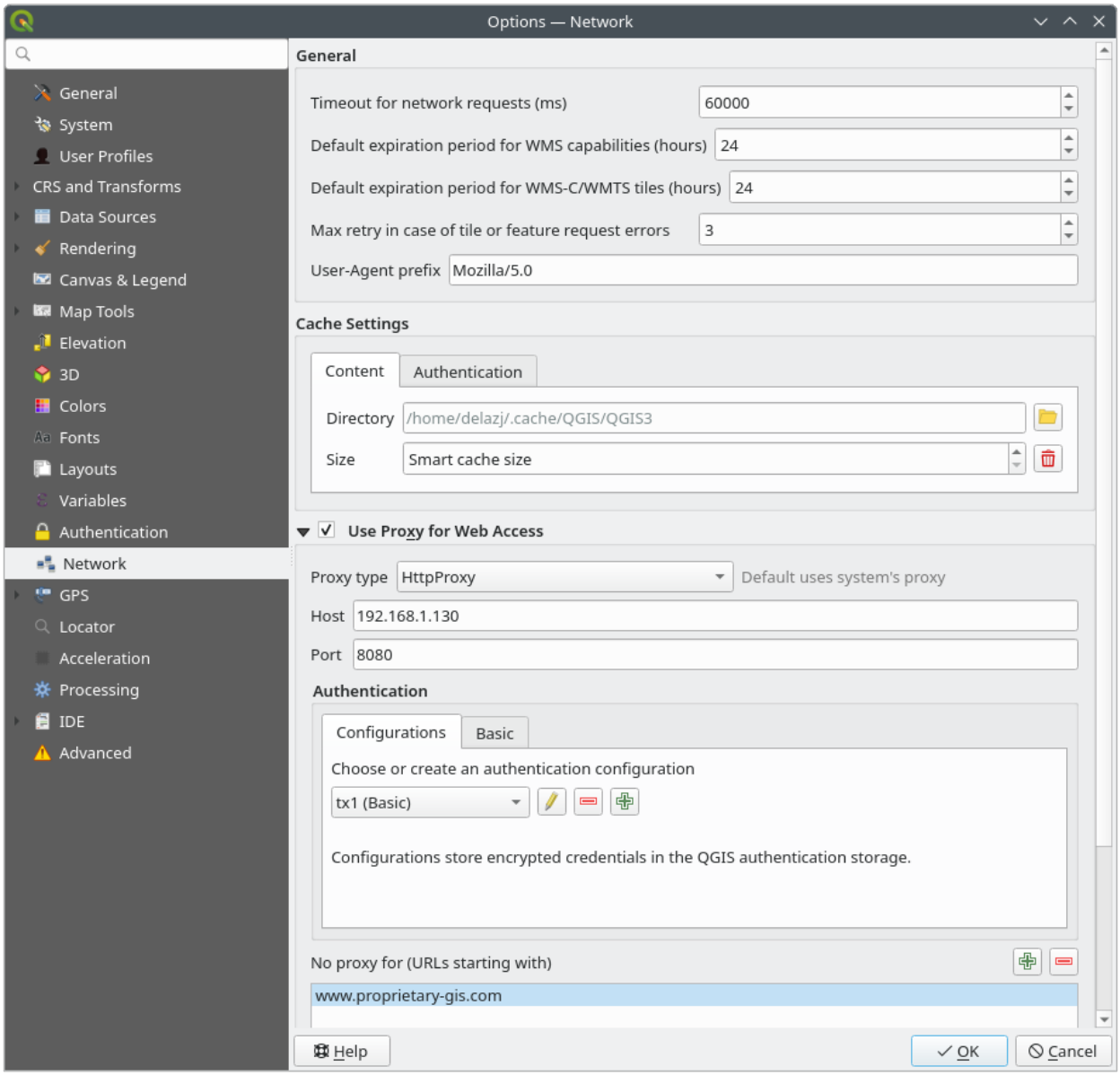



그림 5.23: 네트워크 및 프록시 설정

- *HttpCachingProxy*: 통상적인 HTTP 명령어로 시행되며, HTTP 요청이라는 맥락 안에서만 유용합니다.
- *FtpCachingProxy*: FTP 프록시를 이용해서 시행되며, FTP 요청이라는 맥락 안에서만 유용합니다.

프록시의 인증 정보 (credential) 는인증 위젯 을 사용해서 설정합니다.

프록시 설정 아래에 있는 텍스트 상자에 제외시킬 URL 을 추가할 수 있습니다. (그림 5.23 참조) 목표 URL 이 텍스트 상자 목록에 있는 문자열 가운데 하나로 시작된다면 어떤 프록시도 사용하지 않을 것입니다.

서로 다른 프록시 설정들에 대한 자세한 정보가 필요한 경우, https://doc.qt.io/archives/qt-5.9/qnetworkproxy.html#ProxyType-enum 에 있는 기저 Qt 라이브러리 문서를 참조하십시오.

> **팁**
>
> **프록시 사용**
>
> 프록시 사용은 때로 까다로울 수 있습니다. 앞에서 설명한 프록시 유형들을'시행착오'를 거쳐 가며 사용자의 환경에서 어떤 유형이 작동하는지 확인하는 편이 좋습니다.

### 5.1.17 GPS 설정

*GPS* 대화창은 QGIS 내에서 GPS 기기의 연결과 속성을 환경설정할 수 있게 해줍니다. 또한 GPS 추적 및 데이터 디지타이즈 작업에 대한 설정도 할 수 있습니다.

사용자는 다음을 지정할 수 있습니다:

- QGIS 에 GPS 기기를 연결할 수 있는 몇 가지 서로 다른 수단을 제공합니다.
  - *Autodetect*
  - *Serial device* (새 GPS 기기를 연결한 경우 다시 불러오기를 해야 합니다)
  - *gpsd* (사용자의 GPS 기기가 연결된 *Host* 및 *Port* 를 선택한 다음 기기 이름 *Device* 을 입력해야 합니다.)

  연결에 문제가 생겼을 경우, 이 수단들 가운데 하나로 전환시켜볼 수 있습니다.
- *Filtering*: 수신기가 정적 상태일 때도 커서를 활성화 상태로 유지하려면, *Acquisition interval (seconds)* 그리고/또는 *Distance threshold (meters)* 파라미터도 설정하면 됩니다.
- *Geometry Options* 부분에 있는 *Store in M values* 옵션은 내장 GPS 로그 작업 도구들로부터 M 값을 가진 도형을 생성할 수 있게 해줍니다. 이 옵션은 GPS 로그로부터 그리고로그를 *Geopackage/SpatiaLite* 로 기능으로부터 디지타이즈된 피처 모두에 적용됩니다. 타임스탬프 (에포크에서 시작하는 밀리초 단위), 지상 속도, 고도, 방향, 그리고 정확도 구성요소들을 M 값으로 저장하는 옵션들을 포함하고 있습니다.
- *GPS Location Marker*:
  - 현재 GPS 위치를 나타내는 *GPS location marker* 용 심볼을 환경설정합니다.
  - *Rotate location marker to match GPS bearing*: 마커 심볼을 GPS 방향과 일치하도록 회전시켜야 할지 여부를 제어합니다.
- *GPS Bearing*:
  - QGIS 라인 심볼 속성을 사용하는 *Bearing line style* 을 환경설정합니다.
  - *Calculate Bearings from travel direction*: 진행 방향으로부터 방향을 계산할지 말지를 설정합니다. 이 옵션을 체크하면 GPS 기기가 보고하는 방향을 무시하고 기존의 두 GPS 위치 사이의 각도로 방향을 계산할 것입니다.

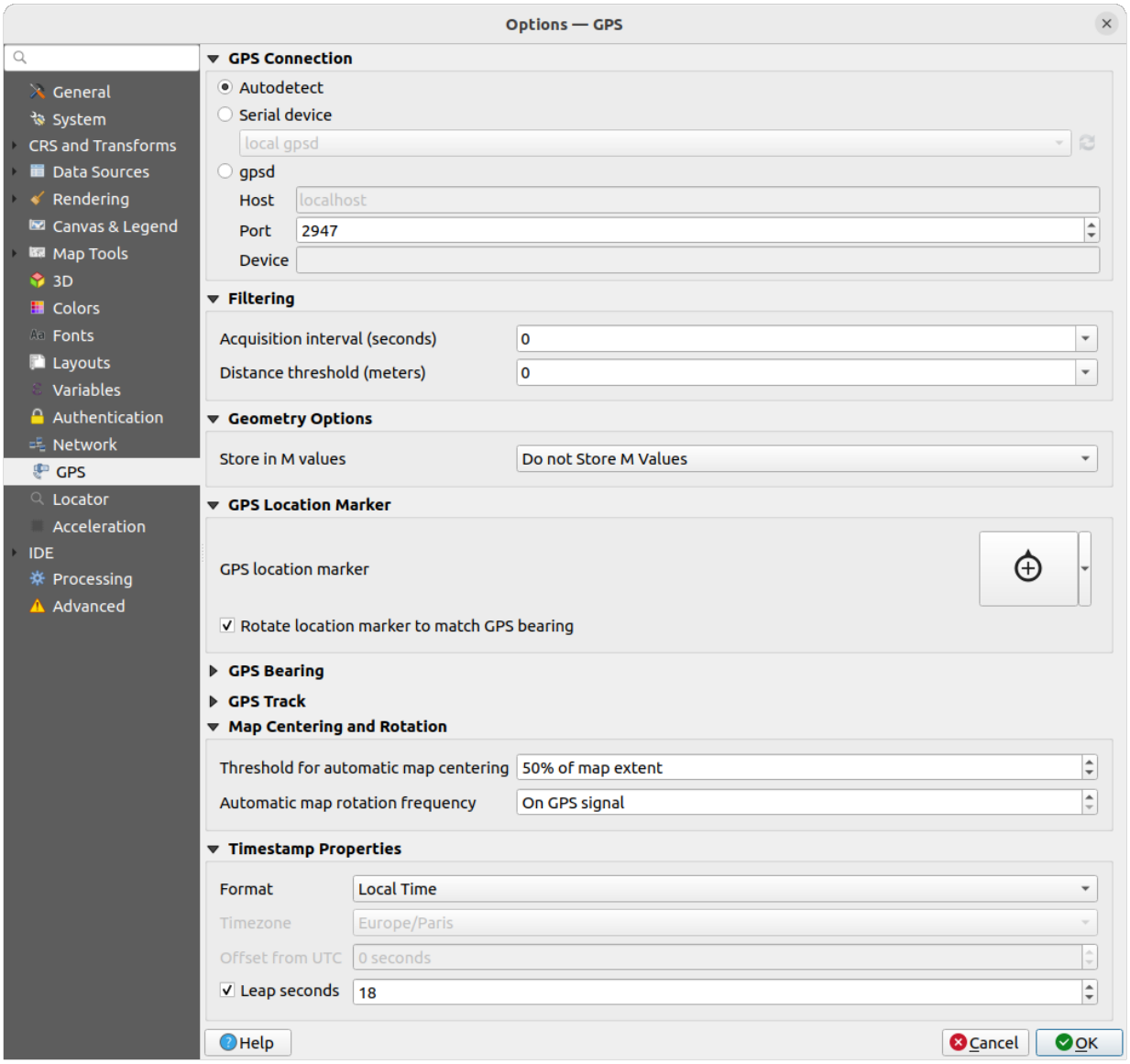

Options — GPS
General
System
CRS and Transforms
Data Sources
Rendering
Canvas & Legend
Map Tools
3D
Colors
Fonts
Layouts
Variables
Authentication
Network
GPS
Locator
Acceleration
IDE
Processing
Advanced
GPS Connection
Autodetect
Serial device
local gpsd
gpsd
Host
localhost
Port
2947
Device
Filtering
Acquisition interval (seconds)
0
Distance threshold (meters)
0
Geometry Options
Store in M values
Do not Store M Values
GPS Location Marker
GPS location marker
Rotate location marker to match GPS bearing
GPS Bearing
GPS Track
Map Centering and Rotation
Threshold for automatic map centering
50% of map extent
Automatic map rotation frequency
On GPS signal
Timestamp Properties
Format
Local Time
Timezone
Europe/Paris
Offset from UTC
0 seconds
Leap seconds
18
Help
Cancel
OK

그림 5.24: GPS 설정

- *GPS Track*: *Track line style* 에 사용할 심볼을 설정합니다.
- *Map Centering and Rotation*: 맵 캔버스를 언제 현장에서 사용자의 위치 변화에 따라 업데이트할지를 정의합니다:
  - *Threshold for automatic map centering*: 이 옵션을 활성화하면, 맵 중심좌표 재계산 을 촉발하려면 GPS 위치가 맵 캔버스 중심으로부터 최소 어느 정도 벗어나야 하는지를 정의해야 합니다.
  - *Automatic map rotation frequency*: 맵을 GPS 방향과 일치시키기 위해 어느 정도의 빈도로 회전시킬지를 정의합니다. 사용자가 그 빈도를 설정할 수도 있고, 또는 GPS 신호에 맞춰 *On GPS signal* 회전하도록 할 수도 있습니다.
- *Timestamp Properties*: 데이터 내에 시간값을 어떻게 표시하고 저장할지를 환경설정합니다. 이 옵션에는 다음을 설정할 수 있는 *Format* 파라미터가 존재합니다:
  - *Local time*: 현지 시간을 설정합니다.
  - *UTC*: 협정 세계시를 설정합니다.
  - *UTC with offset*: 일광 절약 시간제 (서머타임) 또는 다른 복잡한 시간대 (time zone) 문제를 위해 시간 오프셋을 설정할 수 있습니다.
  - *Time zone*: 특정 시간대를 설정합니다.

  *Leap seconds*: GPS 타임스탬프에 초 단위 시간을 추가하는 방법으로 윤초 수정도 할 수 있습니다.

#### GPSBabel

GPSBabel 소프트웨어는 가민 (Garmin) 또는 마젤란 (Magellan) 같은 유명한 GPS 수신기와 구글 어스 또는 베이스캠프 (Basecamp) 같은 매핑 프로그램 사이의 웨이포인트 (waypoint), 트랙 (track), 그리고 루트 (route) 를 변환합니다. 말 그대로 수백 개의 GPS 수신기와 프로그램을 지원합니다. QGIS 는 이런 기기들과 대화형 작업을 하고*GPS* 데이터를 처리 하기 위해 GPSBabel 에 의존하고 있습니다.

사용 방법에 대해 자세히 알고 싶다면기기로 또는 기기에서 불러오기 를 참조하세요.

### 5.1.18 위치 탐색란 설정

 The *Locator* 탭에서위치 탐색란 의 환경을 설정할 수 있습니다. 위치 탐색란이란 상태 바에서 사용할 수 있는 빠른 검색 위젯으로, 응용 프로그램에서 검색 작업을 수행할 수 있도록 해줍니다. 다음과 같은 (접두어를 가진) 기본 필터를 제공합니다:

- *Project Layers* (`l`): *Layers* 패널에서 레이어를 찾아서 선택합니다.
- *Project Layouts* (`pl`): 인쇄 조판을 찾아서 엽니다.
- *Actions* (`.`): QGIS 액션을 찾아서 실행합니다. 액션이란 QGIS 에 있는 어떤 도구나 메뉴도 될 수 있고, 패널을 열 수도 있고, …
- *Active Layer Features* (`f`): 현재 활성화된 레이어의 어느 항목에서건 일치하는 속성을 검색해서 선택된 피처로 확대/축소합니다. 검색 결과의 최대 개수를 환경 설정하려면  아이콘을 클릭하십시오.
- *Features in All Layers* (`af`): 각검색 가능한 레이어 의표시 이름 에서 일치하는 속성을 검색해서 선택된 피처로 확대/축소합니다. 검색 결과의 최대 개수 및 레이어별 검색 결과의 최대 개수를 환경설정하려면  아이콘을 클릭하십시오.
- *Calculator* (`=`): 모든 QGIS 표현식을 검증할 수 있고, 표현식이 무결한 경우 결과물을 클립보드로 복사할 수 있습니다.
- *Spatial Bookmarks* (`b`): 북마크를 찾아서 북마크의 범위로 확대/축소합니다.

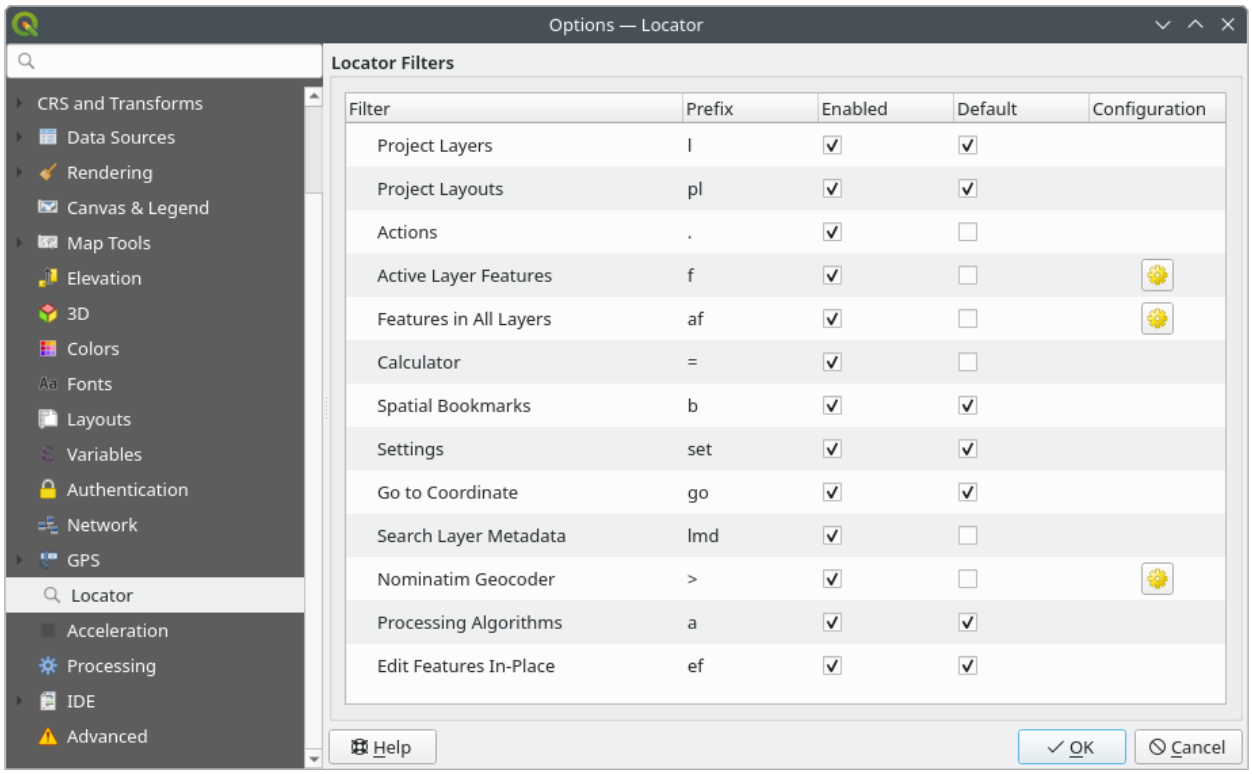



그림 5.25: 위치 탐색란 설정

- *Settings* (`set`): 프로젝트를 탐색해서 연 다음, 응용 프로그램 수준의 속성 대화창을 엽니다.
- *Go to Coordinate* (`go`): 쉼표 또는 공백으로 구분된 x 및 y 좌표 쌍 또는 서식화된 URL(예: 오픈스트리트맵, 리플릿 (Leaflet), 오픈레이어스, 구글 지도 등등) 이 정의하는 위치로 맵 캔버스를 이동시킵니다. 좌표는 WGS 84 (`epsg:4326`) 그리고/또는 맵 캔버스 좌표계여야 합니다.
- *Nominatim Geocoder* (`>`): geocodes using the Nominatim geocoding service of the OpenStreetMap Foundation. Press  to limit results to one or more countries using list of country codes. This helps narrow down the search to specific countries, avoiding ambiguous results from other locations. For example, entering `DE` or `DE,FR` restricts results to Germany or Germany and France, respectively.
- *Processing Algorithms* (`a`): 공간 처리 알고리즘을 검색해서 공간 처리 알고리즘 대화창을 엽니다.
- *Edit Selected Features* (`ef`): 빠른 접근을 허용해서 활성 레이어에 호환되는제자리 수정 공간 처리 알고리즘을 실행합니다.

이 대화창에서 다음과 같은 작업을 할 수 있습니다:

- customize the filter *Prefix*, i.e., the keyword to use to trigger the filter
- 필터를 *Enabled* 할지 여부를 설정할 수 있습니다: 활성화된 필터는 검색에 사용할 수 있으며, 위치 탐색란 메뉴에서 단축키도 사용할 수 있습니다.
- 필터가 *Default* 인지 여부를 설정할 수 있습니다: 필터를 사용하지 않은 검색은 기본 필터 범주에서만 결과를 반환합니다.
- 일부 필터는 검색 결과의 개수를 환경설정할 수 있는 방법을 제공합니다.

플러그인을 통해 기본 위치 탐색 필터 모음을 확장시킬 수 있습니다. 예를 들어 OSM 노미나팀 검색, 데이터베이스 직접 검색, 레이어 카탈로그 검색 플러그인 등등을 사용할 수 있습니다.

### 5.1.19 가속 설정

OpenCL 가속을 설정합니다.

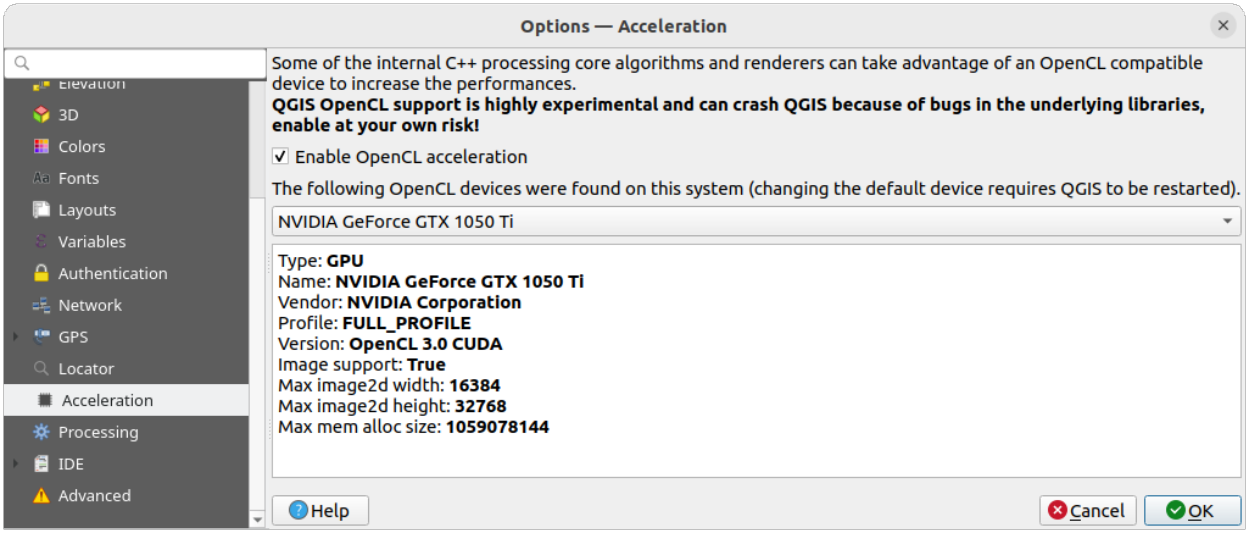



그림 5.26: 가속 설정

사용자의 하드웨어 및 소프트웨어에 따라, OpenCL 가속을 활성화하려면 추가 라이브러리를 설치해야 할 수도 있습니다.

### 5.1.20 IDE 설정

Under *GitHub access token*, you can generate a personal token allowing you to share code snippets from within the Python code editor. More details on GitHub authentication

#### 코드 편집기 설정

*Code Editor* 탭에서 코드 편집기 위젯 (파이썬 대화형 콘솔 및 편집기, 표현식 위젯, 함수 편집기 등등) 의 모양 및 습성을 제어할 수 있습니다.

대화창 상단에 있는 위젯은 현재 설정 상태를 여러 코딩 언어 (파이썬, QGIS 표현식, HTML, SQL, 자바스크립트) 의 실시간 미리보기로 제공합니다. 매우 편리하게 설정을 조정할 수 있죠.

- *Override Code Editor Font* 옵션을 체크하면 기본 글꼴 집합 *Font* 과 글꼴 크기 *Size* 를 수정할 수 있습니다.
- *Colors* 부분에서 다음 작업을 할 수 있습니다:
  - *Color scheme* 를 선택할 수 있습니다: 미리 정의된 설정으로 `Default`, `Solarized Dark` 그리고 `Solarized Light` 이 있습니다. 사용자가 색상을 수정하는 순간 `Custom` 스키마가 촉발되는데, 사전 정의 스키마를 선택하면 되돌릴 수 있습니다.
  - 주석, 인용, 함수, 배경 등등에 사용할 색상과 같은 코드 작성의 각 요소의색상 을 변경할 수 있습니다.

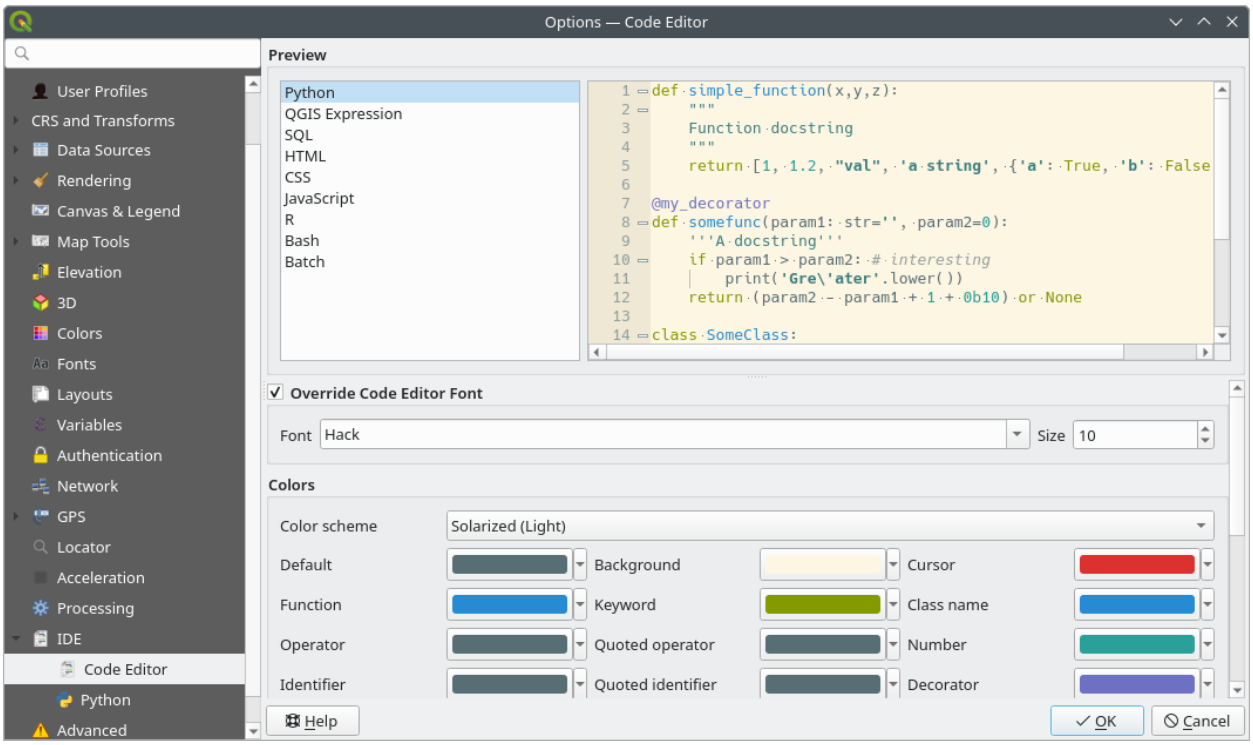



그림 5.27: 코드 편집기 설정

## Python settings

The *Python* settings help you manage and control the behavior of the Python editors (*interactive console*, *code editor*, *project macros*, *custom expressions*, ⋯). It can also be accessed using the Options⋯ button from:

- *Python console* 툴바
- *Python console* 위젯의 컨텍스트 메뉴
- 코드 편집기의 컨텍스트 메뉴

사용자는 다음을 지정할 수 있습니다:

- *Autocompletion*: Enables code completion. You can get autocompletion from the current document, the *installed API* files or both.
  - *Autocompletion threshold*: Sets the threshold for displaying the autocompletion list (in number of characters)
- *Typing* 아래 있는
  - *Automatic parentheses insertion*: 어떤 텍스트도 선택하지 않았을 때 여는 문자 (괄호, 따옴표, 꺾쇠표 등등) 를 입력하면, 커서 바로 뒤에 해당 문자와 일치하는 닫는 문자를 삽입합니다. 현재 커서가 문자열 또는 주석 안에 위치하는 경우에는 이 습성이 비활성화된다는 점을 기억하십시오.
  - *Automatically surround selection when typing quotes or brackets*: 여는 문자를 입력했을 때, 선택한 텍스트를 여는 문자/닫는 문자 쌍으로 에워쌉니다. 선택 텍스트는 동일하게 유지되기 때문에, " 그 다음 ( 를 입력하기만 하면 선택한 텍스트를 괄호로 에워싼 상태로 인용할 수 있습니다.

    작은따옴표와 큰따옴표를 가진 여러 줄을 선택한 특수한 경우: 선택한 텍스트를 삼중 작은/큰 따옴표로 에워쌉니다.

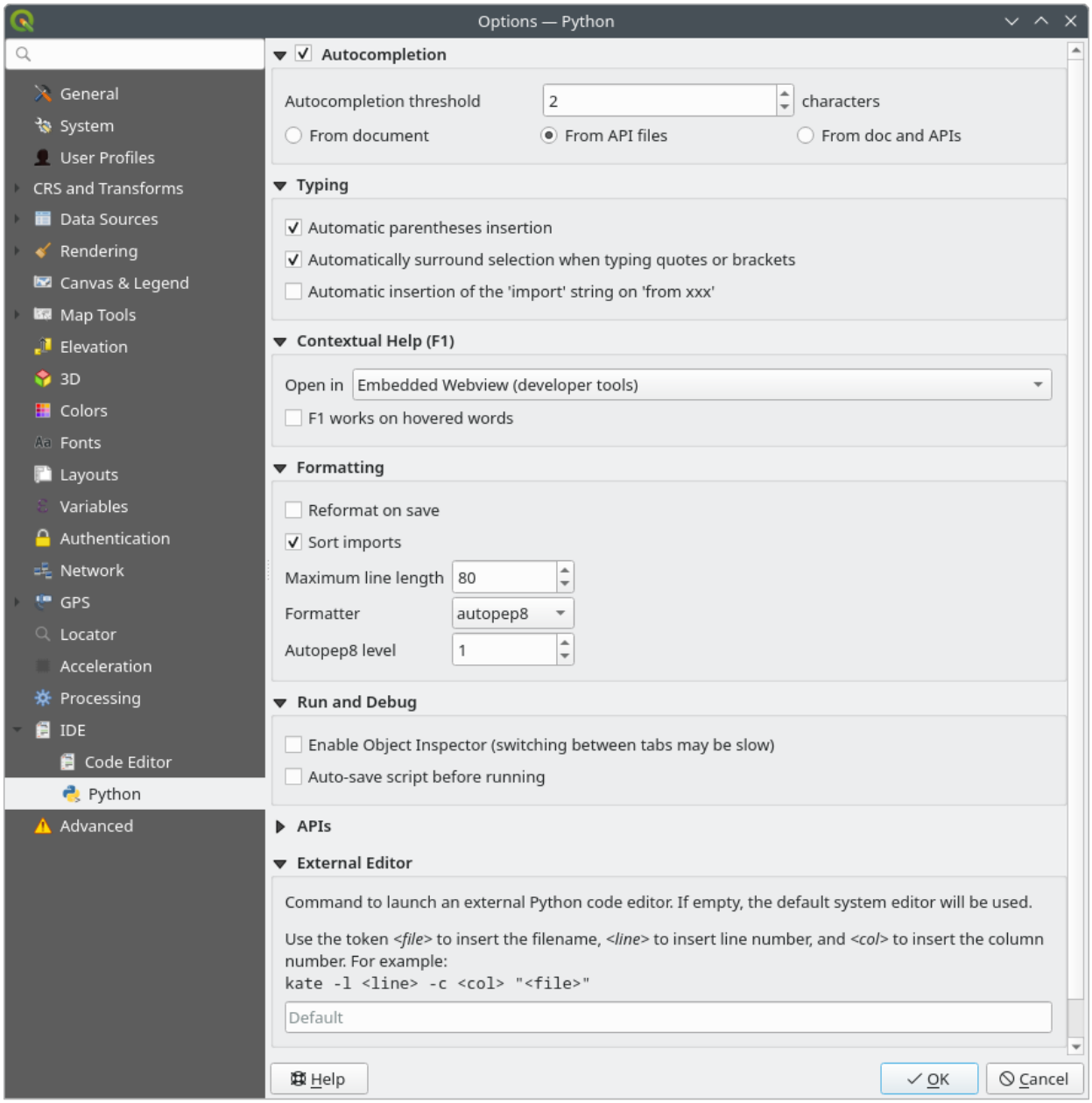



그림 5.28: Python settings

  - *Automatic insertion of the 'import' string on 'from xxx'*: 가져오기를 지정할 때 'import' 문자열의 삽입을 활성화합니다.

- *Contextual Help (F1)* provides options to control QGIS behavior when requesting help on a PyQGIS or Qt object in a Python code sample. When pressing `F1`, QGIS will open the API page (or a search results page) matching:

  1. the selected text if any OR
  2. the word at the mouse position (if *F1 works on hovered words* is enabled) OR
  3. the word at the text cursor position

  This works for both Qt and PyQGIS API documentation. The page is opened either in QGIS, as an *Embedded webview (developer tools)*, or in the *Default system web browser*.

- *Formatting* 아래에서 사용자가 작성 중인 코드를 재서식화할 수 있는 자동화 도구들을 추가할 수 있습니다:
  - *Reformat on save*: 스크립트를 저장하기 직전에 서식을 적용합니다.
  - *Sort imports*: isort 라이브러리 를 이용해서 'import' 문을 정렬합니다.
  - *Maximum line length*: 편집기 자 (ruler) 를 제어해서 서식 결정자 (*Formatter*) 가 라인을 어떻게 줄바꿈시킬지를 제어합니다.
  - *Formatter*: 다음 전용 옵션을 가진 *autopep8* 및 *black* 도구를 지원합니다:
    * *Autopep8 level* - autopep8 고급 활용 페이지에서 더 자세한 내용을 볼 수 있습니다.
    * *Normalize quotes* - 가능한 경우, 작은 따옴표를 모두 큰 따옴표로 대체합니다.
- *Run and Debug* 아래 있는
  - *Enable Object Inspector (switching between tabs may be slow)*: 이 옵션을 체크하면 객체 점검기를 활성화합니다. (탭 변경이 조금 느려질 수도 있습니다.)
  - *Auto-save script before running*: 실행 시 스크립트를 자동으로 저장하는 옵션입니다. 시스템의 임시 디렉터리에 임시 파일을 저장하고, 실행 후에 자동적으로 삭제할 것입니다.

Under *APIs*, you can specify which APIs your code should rely on:

- *Using preloaded APIs file*: You can choose if you would like to use the preloaded API files. By default, supported APIs are:
  - Python
  - PyQGIS
  - PyQt
  - QScintilla2
  - osgeo-gdal-ogr
  - osgeo-geos

  You can also choose to add specific API files.

- *Using prepared APIs file*: 이 옵션을 활성화하면 코드 자동 완성을 위해 `*.pap` 파일을 사용할 것입니다. 준비된 API 파일을 생성하려면, 적어도 하나의 `*.api` 파일을 불러온 다음 *Compile APIs*… 버튼을 클릭해서 컴파일해야 합니다.

The *External Editor* group allows you to provide command line instructions to launch an external Python code editor, given a file name, a line and column number. If empty, the default system editor will be used.

### 5.1.21 공간 처리 설정

The *Processing* tab provides you with general settings of tools and data providers that are used in the QGIS Processing framework. More information at 공간 처리 프레임워크 환경 설정.

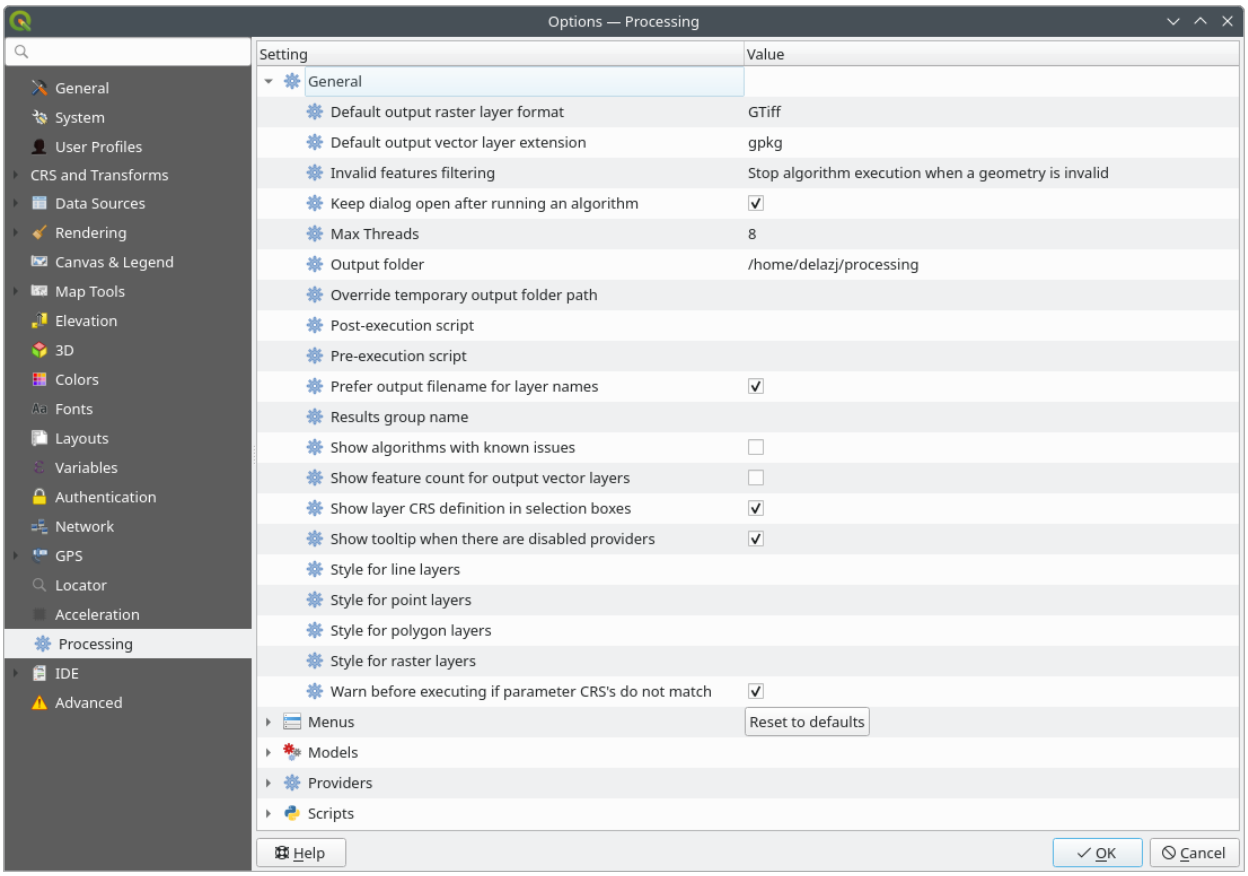



그림 5.29: 공간 처리 설정

### 5.1.22 고급 설정

QGIS 와 관련된 모든 설정은 (UI, 도구, 데이터 제공자, 공간 처리 환경 설정, 기본값 및 기본 경로, 플러그인 옵션, 표현식, 도형 검사 등등) 활성화된사용자 프로파일 디렉터리 아래 `QGIS/QGIS3.ini` 파일로 저장됩니다. 이 파일을 다른 설치본으로 복사하면 환경 설정을 공유할 수 있습니다.

QGIS 내에서라면, *Advanced* 탭은 *Advanced Settings Editor* 를 통해 이런 설정들을 관리할 수 있는 방법을 제공합니다. 사용자가 조심하겠다는 확인을 하고 나면, 해당 위젯은 존재하는 모든 설정들의 트리로 채워지고 그 값을 편집할 수 있게 됩니다. 어떤 설정이나 그룹을 오른쪽 클릭하면 해당 항목을 삭제할 수 있습니다. (설정 또는 그룹을 추가하려면, `QGIS3.ini` 파일을 편집해야 합니다.) 변경 사항은 `QGIS3.ini` 파일에 자동적으로 저장됩니다.

**경고**

**고급 설정 탭을 무턱대고 이용하지 말 것**

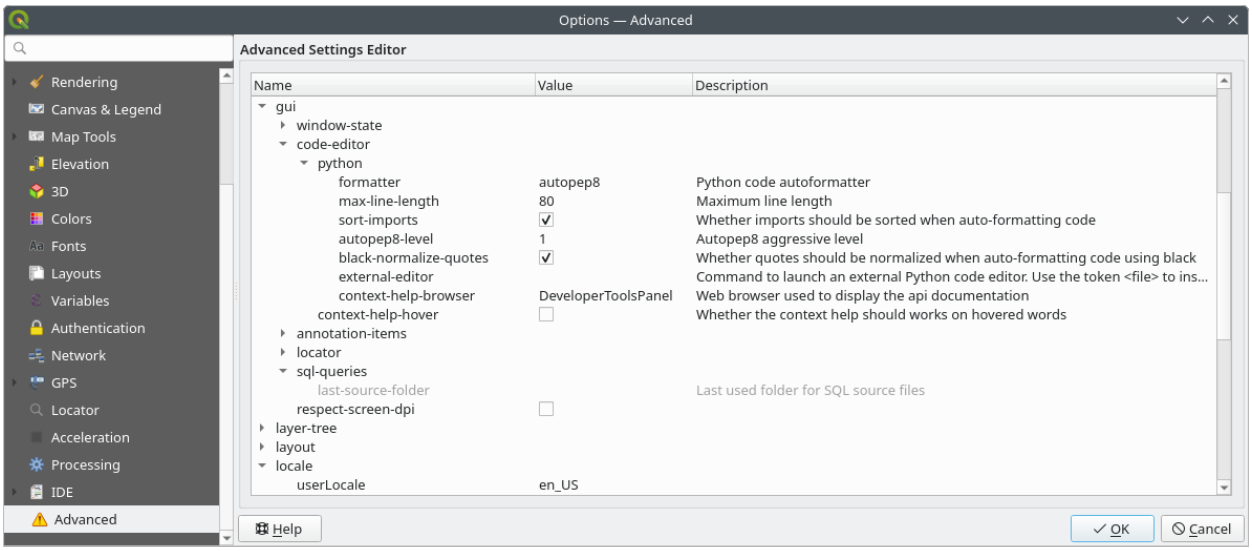



그림 5.30: 고급 설정

이 대화창에서 항목을 수정할 때 조심해야 합니다. 변경 사항이 자동적으로 적용되기 때문입니다. 자세한 내용을 모른 채 설정을 수정하면 사용자의 QGIS 설치본이 여러 방식으로 고장날 수 있습니다.

## 5.2 사용자 프로파일 작업

### 5.2.1 개념

*Settings* ▸ *User Profiles* 메뉴는 사용자 프로파일을 설정하고 프로파일에 접근하는 기능을 제공합니다. 사용자 프로파일이란 통합 응용 프로그램 환경 설정으로, 단일 폴더 안에 저장할 수 있습니다:

- 로케일, 투영체, 인증 설정, 색상표, 단축키 등등을 포함하는 모든전체 수준 설정들
- GUI 환경 설정 및사용자 지정 설정
- 데이터 변형 (transformation) 을 위해 설치된 그리드 파일 및 기타 PROJ 도우미 파일
- 설치된플러그인 과 그 환경 설정
- 투영 템플릿 그리고 그 이미지 미리보기를 포함한 저장된 프로젝트의 이력
- 공간 처리 설정, 로그, 스크립트, 모델

QGIS 설치본은 기본적으로 `default` 라는 단일 사용자 프로파일을 담고 있습니다. 하지만 사용자가 원하는만큼 많은 사용자 프로파일을 생성할 수 있습니다:

1. *Settings* ▸ *User Profiles* ▸ *New profile*…메뉴 항목을 클릭하십시오.
2. 프로파일 명칭을 입력해야 하는데, 명칭을 입력하면 `~/<UserProfiles>/` 아래 동일 명칭의 폴더를 생성합니다. 이때:
   - `~` 문자가 **홈** 디렉터리를 나타냅니다. 윈도우 상에서의 홈 디렉터리란 보통 `C:\Users\<username>` 같은 경로입니다.
   - `<UserProfiles>` 는 주 프로파일 폴더를 나타냅니다. 예를 들면:

  - .local/share/QGIS/QGIS3/profiles/
  - %AppData%\Roaming\QGIS\QGIS3\profiles\
  - Library/Application Support/QGIS/QGIS3/profiles/

  *Open Active Profile Folder* 메뉴를 클릭하면 QGIS 에서 사용자 프로파일 폴더를 열 수 있습니다.

3. 새 QGIS 인스턴스를 시작하면, 초기화된 환경 설정을 사용합니다. 그러면 환경을 사용자 지정 설정할 수 있습니다.

사용자의 QGIS 설치본이 프로파일을 하나 이상 가지고 있을 경우, 활성화된 프로파일의 이름이 응용 프로그램 제목 막대에 대괄호로 감싸여 표시됩니다.

각 사용자 프로파일이 독립된 설정, 플러그인, 이력을 담고 있으므로, 서로 다른 워크플로나 데모, 동일 머신의 서로 다른 사용자들, 또는 테스트용 설정 등에 매우 유용할 수 있습니다. *Settings* → *User Profiles* 메뉴에서 프로파일을 선택하면 한 프로파일에서 다른 프로파일로 전환할 수 있습니다. 또명령 줄 에서 특정 사용자 프로파일로 QGIS 를 실행할 수도 있습니다.

**팁**

**버그 지속성 검사를 위해 새 사용자 프로파일로 QGIS 실행하기**

Bugs you may encounter with functions in QGIS can be related to leftovers in the current user profile. Running QGIS under another user profile can help you solve them or check the origin of the issue. The general advice is to launch QGIS under a new user profile, thus a cleaner configuration, and run the commands again.

If the bug prevents you to create a new user profile from within the *Settings* → *User Profiles* menu, you can either:

- Rename in the file explorer, the "broken"user profile folder in the `QGIS3/profiles` folder and restart QGIS. A new `default` user profile will be created and executed.
- Start QGIS from the command line, using the new *profile name* argument:

```
qgis-ltr --profile newprofilename
```

### 5.2.2 사용자 프로파일 설정하기

QGIS 는 기본적으로 마지막으로 닫았던 세션의 프로파일로 새 세션을 엽니다. *Settings* → *Options* → *User Profiles* 탭에 있는 다른 설정들 사이에서 이 습성을 사용자 지정할 수 있습니다:

- *Startup profile*: QGIS 세션을 시작할 때 사용할 사용자 프로파일을 나타냅니다. 다음 가운데 하나로 설정할 수 있습니다:
  - *Use last closed profile*: 마지막으로 닫았던 프로파일을 사용합니다.
  - *Always use profile*: 이 드롭다운 메뉴에서 선택한 특정 사용자 프로파일을 사용합니다.
  - *Choose profile at start up*: 사용할 수 있는 사용자 프로파일 목록을 가진 *User Profile Selector* 대화창을 엽니다. 어떤 항목을 더블클릭하거나, 프로파일 하나를 선택한 다음 *OK* 버튼을 누르면 해당 프로파일을 사용해서 세션을 시작합니다. 이 목록에 새 프로파일을 추가 *Add new profile* 할 수도 있습니다. 이 버튼을 누르면 프로파일 하나를 처음부터 생성한 다음 생성한 프로파일을 사용해서 자동으로 세션을 엽니다.
- *Profile display* 부분에서 다음을 설정할 수 있습니다:
  - *User Profile Selector* 대화창에서 프로파일을 선택할 때 사용할 아이콘의 크기를 설정할 수 있습니다.

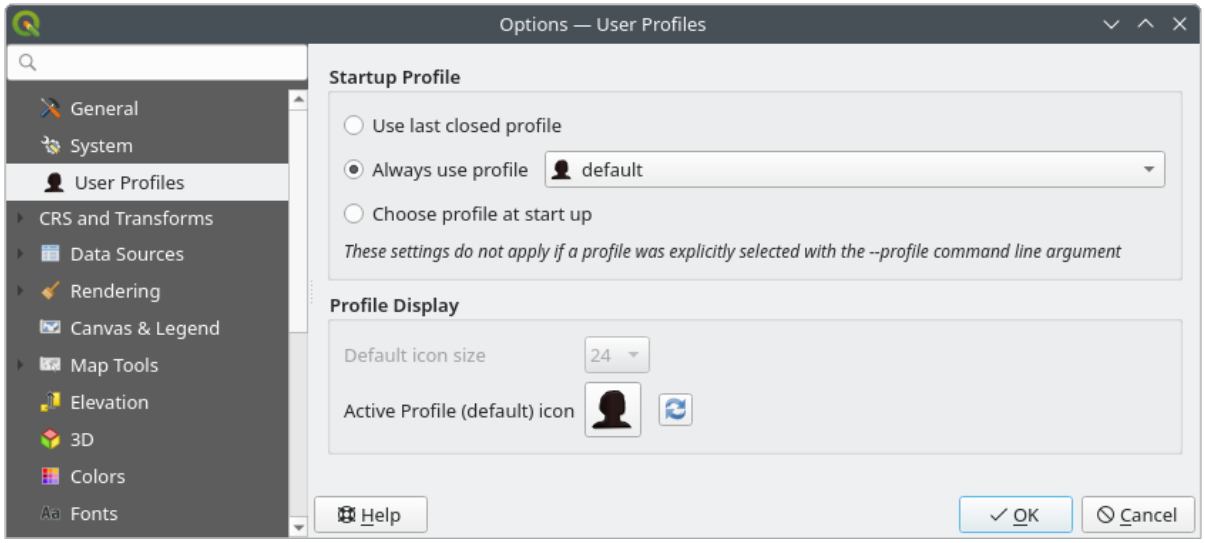



그림 5.31: 사용자 프로파일 설정

– *Settings* ▸ *User profiles* 메뉴 또는 *User Profile Selector* 대화창에 있는 현재 프로파일 옆에 표시할 특정 아이콘을 선택할 수 있습니다. 현재 사용자 지정 수정 사항을 제거하려면 Reset profile icon 버튼을 누르십시오.

## 5.3 프로젝트 속성

*Project* ▸ *Project Properties* 메뉴를 클릭하면 열리는 프로젝트 용 속성 창에서 프로젝트에 특화된 옵션들을 설정할 수 있습니다. 이 프로젝트 특화 옵션은 위에서 설명하는 *Options* 대화창의 대응하는 옵션을 덮어씁니다 (무시하고 재설정합니다).

### 5.3.1 일반 속성

*General* 탭에 있는 *General settings* 부분에서:

- 프로젝트 파일의 위치를 볼 수 있고
- (*Browser* 패널의 *Project home* 항목에서) 프로젝트 홈을 위한 폴더를 설정할 수 있습니다. 이 경로는 (입력한) 프로젝트 파일 폴더에 대한 상대 경로일 수도 있고 절대 경로일 수도 있습니다. 데이터 및 프로젝트에 유용한 다른 콘텐츠 저장을 위해 프로젝트 홈을 사용할 수 있습니다. 데이터셋과 프로젝트 파일이 동일한 위치에 저장되어 있지 않은 경우 유용합니다. 프로젝트 홈을 설정하지 않는 경우, *Project home* 의 기본값은 프로젝트 파일 폴더입니다.
- 프로젝트 파일 경로 옆에 프로젝트 제목을 부여할 수 있고
- 피처 선택 시 피처에 적용할 색상을 선택할 수 있고
- 맵 캔버스에 적용할 배경색을 선택할 수 있으며
- 프로젝트 내부의 레이어를 가리키는 경로를 절대 경로 (전체 경로) 로 저장할지 프로젝트 파일 위치를 기준으로 하는 상대 경로로 저장할지 설정할 수 있습니다. 레이어 및 프로젝트 파일이 이동 또는 공유될 가능성이 있거나, 다른 플랫폼을 이용하는 컴퓨터에서 프로젝트에 접근하는 경우, 상대 경로가 더 적합할 수도 있습니다.

- 프로젝트를 맵 타일로 렌더링할 때 오류 (artifact) 를 회피하도록 선택할 수 있습니다. 이 옵션을 체크하면 성능이 저하될 수도 있다는 점을 주의하십시오.
- *Remember attribute tables windows and docks between sessions*: 어떤 프로젝트에 대해 이 옵션을 체크하면, 열려 있는 모든 속성 테이블을 프로젝트로 저장한 다음 해당 프로젝트를 불러올 때 즉시 복원할 것입니다. 프로젝트를 사용자의 필요에 맞춘 특정 속성 테이블 집합으로 구성한 경우, 이 속성 테이블들을 다시 설정할 필요가 없도록 워크플로를 향상시키는 데 도움이 될 수 있습니다.

면적 및 거리 계산은 GIS 에서 흔히 필요로 하는 작업입니다. 하지만, 이 값들은 기저 투영체 설정과 떼려야 뗄 수 없는 관계입니다. *Measurements* 프레임에서 다음과 같은 파라미터들을 선택하고 조정할 수 있습니다:

- the *Ellipsoid*, on which distance, area and angle calculations are entirely based; it can be:
  - **None/Planimetric**: 이 경우 반환되는 값이 데카르트 평면 상에서 측정한 값이 됩니다. *Settings* → *Options* → *CRS Handling* 메뉴에서 이 옵션을 새 프로젝트의 기본 옵션으로 설정할 수 있습니다.
  - **Custom**: 이 타원체를 선택하면 긴반지름 (semi-major) 과 짧은반지름 (semi-minor) 의 값을 설정해야 합니다.
  - 또는 미리 정의된 목록에서 기존 타원체 가운데 (Clarke 1866, Clarke 1880 IGN, New International 1967, WGS 84, ⋯) 하나를 선택할 수도 있습니다.
- the *Units for distance measurements* for length and perimeter, and the *Units for area measurements*. These settings default to their corresponding *global options* and can be manually changed for the current project. They are used in:
  - 속성 테이블 필드 업데이트 바
  - 필드 계산기 계산
  - 식별 도구 에서 파생되는 길이, 둘레 및 면적 값
  - 측정 대화창
  - 축척 막대 장식

  > **참고**
  >
  > To ensure measurement units (for distance and area) always reflect the units of the current project's CRS, select *Map Units* for these options.

- the *Scale calculation method*: when working on large map extents, especially with geographic CRS, the scale bar may not reflect actual distance measurement depending on the latitude at which the calculation is done. This setting helps you define the area you want the calculation to be taken from:
  - *Average top, middle and bottom scales*
  - *Calculate along top of map*
  - *Calculate along middle of map* (the default)
  - *Calculate along bottom of map*
  - *Always calculate at equator*, regardless of the actual visible map extent. It can be used to provide a consistent, static scale for maps in geographic reference systems, regardless of the latitudes actually visible in the map (permitting consistent appearance of these maps when rendering relies on scale based visibility or calculations). Otherwise a project in e.g. `EPSG:4326` which uses scale based visibility of layers and symbols will see layers and features "randomly" disappear as the map is panned, even though you have not zoomed in or out of the map.

> **참고**
>
> This method is only applicable when calculating scales with a degree based reference system, and while it ensures that the scale remains constant and does not change as the map is panned, it will calculate misleading scales when the map extent is not close to the equator.

Changing the scale calculation method has the following impacts:

- New layout scale bars will default to the project's scale calculation method
- The scale calculations for the status bar widget and map renders will be changed (including flow on impacts like the value of `@map_scale` variable, scale based visibility of layers and symbols)
- This also affects QGIS server map rendering
- Processing algorithms which render maps will respect the project's scale calculation method

> **참고**
>
> Symbology sizes in map units are NOT affected by this setting.

*Coordinate and Bearing display* 프레임에서는 다음을 사용자 지정할 수 있습니다:

- QGIS 상태 바의 *Coordinates* 란 그리고 Identify features 도구의 결과물 가운데 *Derived* 부분에 표시되는 좌표
- 상태 바에 맵 캔버스 이동 방향으로 그리고 Measure bearing 도구의 결과물로 표시되는 방향값

다음과 같은 파라미터를 사용할 수 있습니다:

- *Display coordinates using*: 좌표를 다음 단위 가운데 하나로 표시하도록 설정합니다:
  - `Map Units`: 프로젝트 좌표계 기반 맵 단위로 좌표를 표시합니다.
  - `Map Geographic (degrees)`: 프로젝트 좌표계가 지리 좌표계 유형인 경우 프로젝트 좌표계 기반 단위로, 그렇지 않을 경우 관련 지리 좌표계 기반 단위로 좌표를 표시합니다. 프로젝트 좌표계가 예를 들어 지구 기반이 아닌 천구좌표계인 경우 이 옵션이 유용합니다.
  - 또는 `Custom Projection Units`: 사용자가 좌표를 표시하는 데 사용하길 원하는 어떤 좌표계라도 지정할 수 있습니다.
- 사용자는 *Coordinate CRS* 옵션에서 사용자의 디스플레이 모드에 따라 사용할 좌표계를 살펴보거나 정의할 수 있습니다.
- *Coordinate format*: 좌표 서식을 십진 도, 도, 분 또는 도, 분, 초 가운데 하나로 정의하고, 다음과 같이 표시되도록 환경 설정할 수 있습니다:
  - ☐ *Show directional suffix*: 좌표 뒤에 방향을 표시합니다.
  - ☐ *Show leading zeros for minutes and seconds*: 분과 초 앞에 선행 0 을 표시합니다.
  - ☐ *Show leading zeros for degrees*: 도 앞에 선행 0 을 표시합니다.
  - ☐ *Show trailing zeros*: 소수점 오른쪽의 후행 0 을 표시합니다.
- *Coordinate precision*: 소수점 이하 자릿수를 (좌표계 유형으로부터 파생된) 자릿수로 자동으로 설정되도록 하거나 또는 직접 설정할 수 있습니다.
- *Coordinate order*: 좌표를 좌표계의 원래 순서로 (기본값) 표시하거나, 편동, 편북 (경도, 위도) 또는 편북, 편동 (위도, 경도) 가운데 하나로 표시하도록 변경할 수 있습니다.

- *Bearing format*: 방향 서식을 `E/W` 접두어를 가진 `0 ~ 180°`, `-180 ~ +180°` 또는 `0 ~ 360°` 가운데 하나로 설정할 수 있습니다. *Decimal places* 의 소수점 이하 자릿수는 물론 *Show trailing zeros* 의 후행 0 표시 여부까지 설정할 수 있습니다.

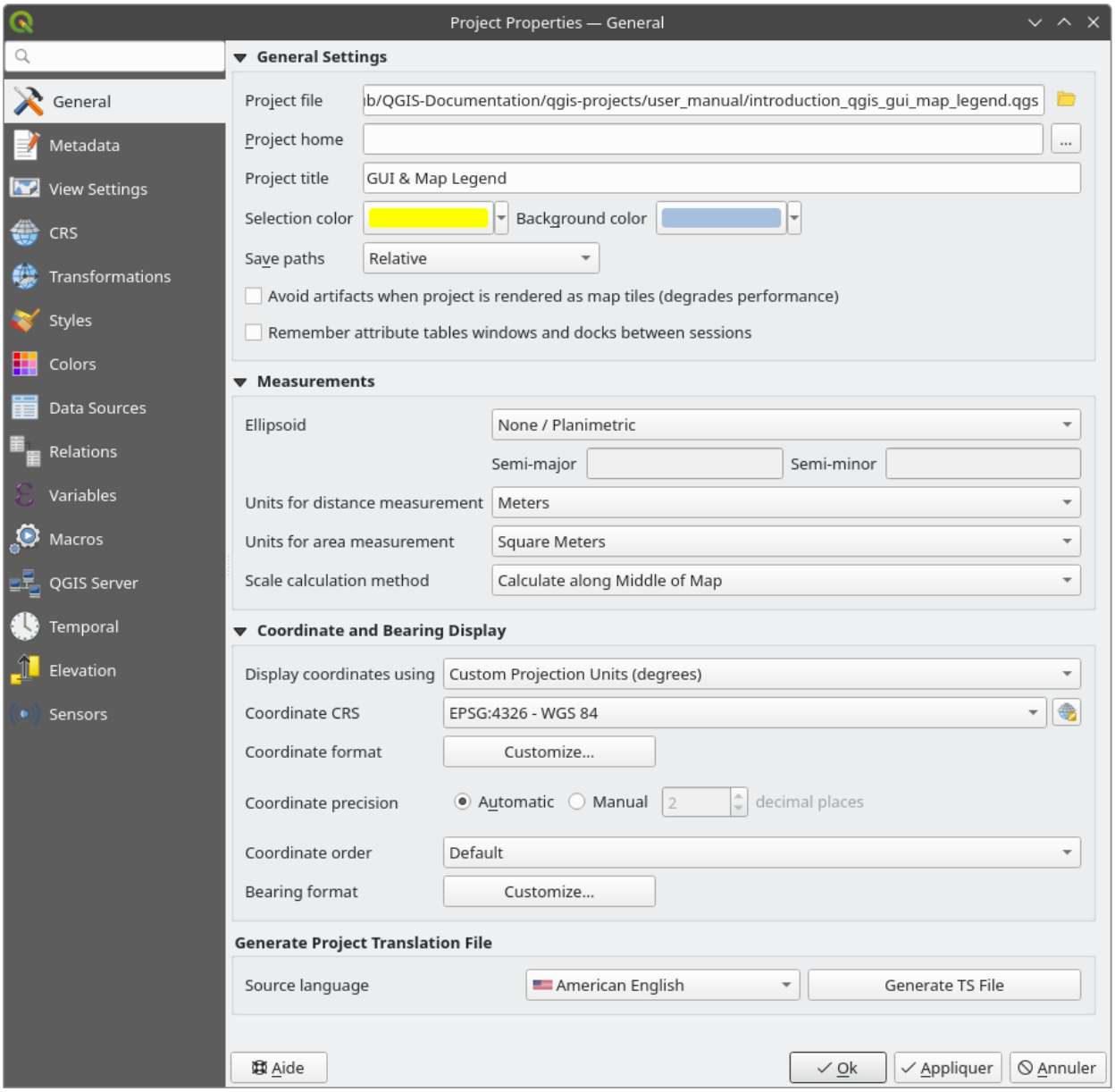



그림 5.32: 프로젝트 속성 대화창의 일반 탭

### 5.3.2 메타데이터 속성

The *Metadata* tab provides you with options to create and edit a metadata report on your project. See 메타데이터 for more information.

### 5.3.3 뷰 설정

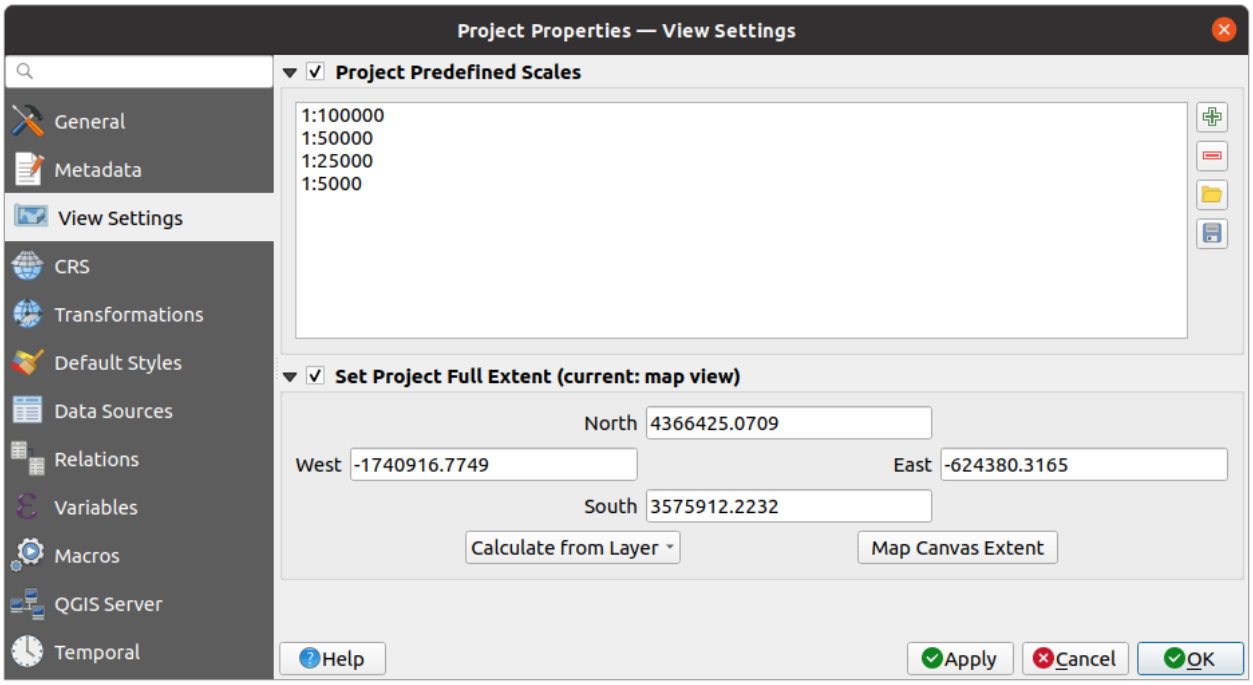



그림 5.33: 프로젝트 속성 대화창의 뷰 설정 탭

*View Settings* 탭은 프로젝트 맵 캔버스를 조정할 수 있는 수단을 제공합니다. 다음을 설정할 수 있습니다:

- *Project predefined scales*: 상태 바의 *Scale* 란, 가시성 축척 선택기, 또는 보조 2 차원 맵 뷰 설정 등과 같은 축척 관련 드롭다운 위젯에 표시되는 프로젝트 수준 사전 정의 축척 목록을 설정할 수 있습니다. 이 목록은 전체 수준사전 정의 축척 을 대체합니다.
- *Set Project full Extent*: 프로젝트의 전체 범위를 설정할 수 있습니다. 맵의 전체 범위로 확대/축소하는 경우 () 모든 레이어의 범위 대신 설정한 범위를 이용할 것입니다. 프로젝트가 웹 레이어/국가 레이어/전 세계 레이어를 담고 있지만 프로젝트의 실제 관심 지역은 좁은 지리 영역인 경우 이 범위를 설정하는 편이 유용합니다. 프로젝트 전체 범위 좌표는범위 선택기 위젯을 통해 설정할 수 있습니다.

### 5.3.4 좌표계 속성

> **참고**
>
> QGIS 가 어떻게 프로젝트 투영체를 처리하는지 더 자세히 알고 싶다면, 투영 작업 에서 심도 있게 다루고 있습니다.

*CRS* 탭에서 현재 프로젝트에 사용할 좌표계를 설정할 수 있습니다. 다음 가운데 하나를 선택할 수 있습니다:

- *No CRS (or unknown/non-Earth projection)*: 레이어의 가공되지 않은 (raw) 좌표를 기반으로 레이어를 렌더링합니다.
- 또는 기존 좌표계에서 지리 *(geographic)*, 투영 *(projected)* 이나 사용자 정의 *(user-defined)* 좌표계 가운데 하나를 선택할 수도 있습니다. 프로젝트에 추가한 레이어를, 원래 좌표계와 상관없이 중첩시키기 위해, 선택한 좌표계로 실시간 (on-the-fly) 재투영합니다.

### 5.3.5 변환 속성

*Transformations* 탭에서는 현재 프로젝트에 적용할 데이터 변환 선호도의 환경을 설정해서 레이어 재투영 설정을 조정할 수 있습니다. 이런 설정들은 마찬가지로 대응하는 전체 수준 설정을 무시합니다. 자세한 내용은기준점 *(datum)* 변환 을 참조하세요.

### 5.3.6 스타일 속성

*Styles* 탭에서, 서로 다른 컴퓨터 사이에서 프로젝트를 안전하게 공유할 수 있도록 프로젝트에 내재된 심볼 및 색상을 환경 설정할 수 있습니다.

#### 기본 심볼

*Default Symbols* 프레임에서는 새 레이어에 기존 `.qml` 스타일 파일이 정의되어 있지 않은 경우 프로젝트에 새 레이어를 어떻게 그릴 것인지를 조정할 수 있습니다. 레이어 도형 유형에 따라 적용할 *Marker*, *Line*, *Fill* 을 설정하는 것은 물론 기본 *Color Ramp* 및 (예를 들어 라벨 작업을 활성화한 경우) *Text Format* 도 설정할 수 있습니다. 대응하는 드롭다운 위젯에서 *Clear* 를 선택하면 이 모든 항목들을 리셋할 수 있습니다.

#### Symbol options

In the *Symbol options* group, you can:

- *Opacity*: 새 레이어에 적용할 기본 투명도를 설정할 수 있습니다.
- *Assign random colors to symbols*: 심볼에 랜덤한 채우기 색상을 할당해서, 모든 레이어가 동일하게 렌더링되는 결과를 피할 수 있습니다.

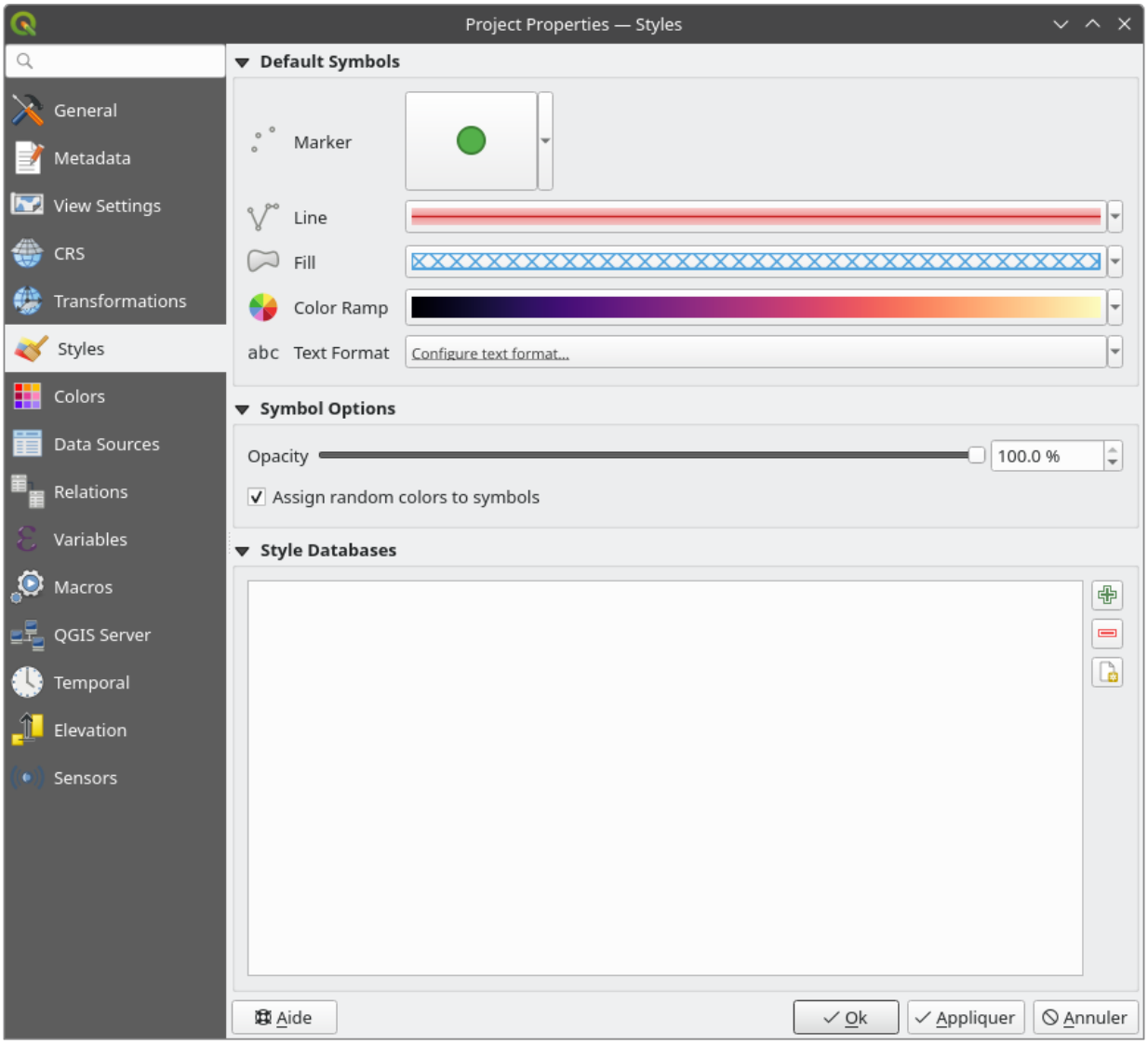

Project Properties — Styles
General
Metadata
View Settings
CRS
Transformations
Styles
Colors
Data Sources
Relations
Variables
Macros
QGIS Server
Temporal
Elevation
Sensors
Default Symbols
Marker
Line
Fill
Color Ramp
abc Text Format
Configure text format...
Symbol Options
Opacity
100.0 %
Assign random colors to symbols
Style Databases
Aide
Ok
Appliquer
Annuler

그림 5.34: 스타일 탭

### 스타일 데이터베이스

QGIS 에서의 스타일 데이터베이스란 심볼, 텍스트 서식, 그리고 다른 스타일 작업 요소들을 저장하기 위해 설계된, 구조화된 저장소를 말합니다. 스타일 데이터베이스는 사용자의 심볼 자원을 효율적으로 구성하고 관리할 수 있는 중심적 위치 역할을 합니다. 사용자는 특정 클라이언트의 필요에 맞춘 심볼을 보관하는, 해당 클라이언트 전용 스타일 데이터베이스를 생성할 수 있습니다. 이렇게 하면 기본 스타일 데이터베이스를 복잡하게 만들지 않고서도 깔끔하고 체계적인 접근 방식을 보장합니다. 다중 사용자 환경에서는 공유 위치에 스타일을 프로젝트별로 저장할 수 있습니다. 이런 스타일에 프로젝트를 연결시키면 해당 프로젝트의 모든 사용자들이 공통 심볼에 접근할 수 있게 되기 때문에 개별적인 로컬 스타일 데이터베이스에 직접 스타일을 가져오거나 업데이트할 필요가 없어집니다. 프로젝트 파일 안에 심볼을 저장하는 방식은 사용자들이 전체 수준 스타일 데이터베이스를 프로젝트별 심볼로 과부하시키는 일을 방지하는 실용적인 솔루션입니다. 이 접근 방식은 동일한 프로젝트를 작업하는 다른 사용자들이 프로젝트를 불러오자마자 필요한 모든 심볼에 접근할 수 있도록 보장합니다.

*Style Database* 부분에서 스타일 데이터베이스를 추가 *Add* 또는 제거 *Remove* 하도록 선택할 수 있으며, 새로운 스타일 데이터베이스를 생성 *Create new style database* 할 수도 있습니다. 여기에서 스타일 데이터베이스를 추가하거나 제거하는 경우, 변경 사항이 *Symbology Properties* 에 자동적으로 반영될 것입니다.

## 5.3.7 Colors properties

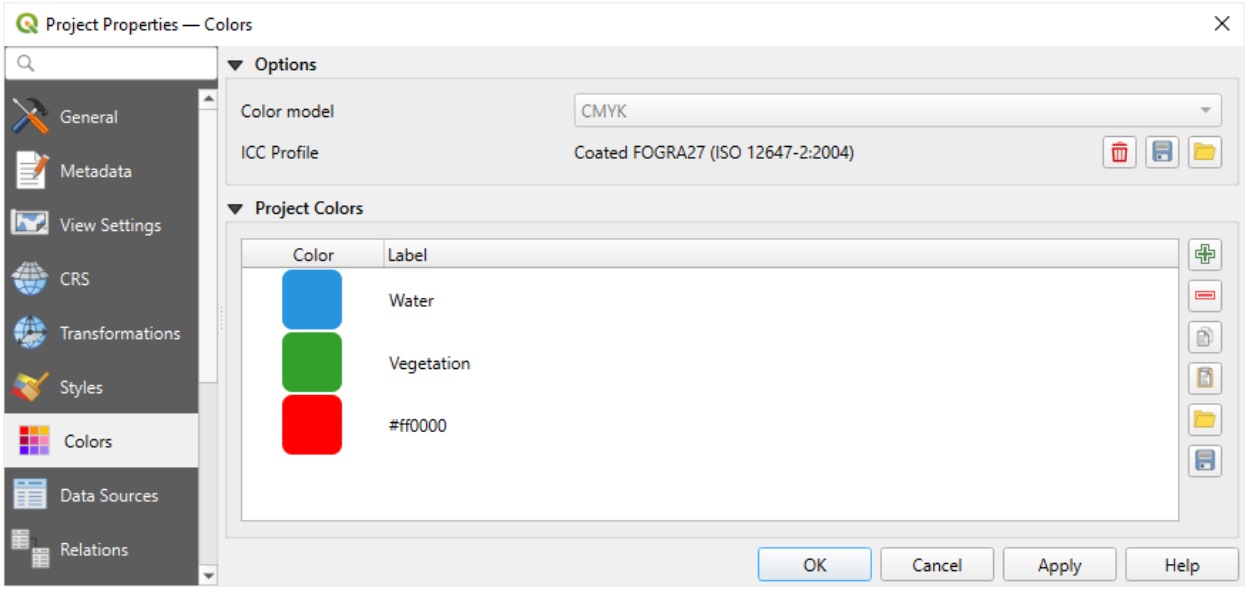



그림 5.35: Colors tab

In the *Colors* tab, you can select the *Color model*, i.e., whether selecting a color should by default be defined as `RGB` or `CMYK`. Any color defined in a different color model than the one specified here will be converted to this color model when exporting a layout.

Additionally, an option is available for selecting an *ICC Profile* which allows you to *Load* a specific ICC profile that defines the color space for your project. After you load the ICC profile, you can choose to *Save ICC Profile* to a file on disk or to *Remove selected ICC Profile*. Once applied, QGIS will enforce the selected color model to ensure consistent color representation.

> **참고**
>
> Note that the *ICC Profile* option will only be visible if QGIS is built with Qt 6.8.0 or higher.

실행 중인 프로젝트에 대해 특정 색상들을 정의할 수 있는 추가 옵션도 있습니다. 예를 들어전체 수준 색상 이라면, 다음과 같은 작업을 할 수 있습니다:

- *Add* 또는 *Remove*: 색상을 추가하거나 삭제합니다.
- *Copy* 또는 *Paste*: 색상을 복사하거나 붙여넣습니다.
- *Import* 또는 *Export*: 색상 모음을 `.gpl` 파일로부터 가져오거나, `.gpl` 파일로 내보냅니다.

목록에 있는 색상을 더블클릭하면색상 선택기 대화창의 해당 색상을 조정하거나 대체할 수 있습니다. *Label* 열을 더블클릭하면 해당 색상의 명칭을 변경할 수도 있습니다.

이 색상들은 *Project colors* 로 식별되어, 색상 위젯 의 한 부분으로 목록화됩니다.

> **팁**
>
> **색상 위젯에 바로 적용하고 업데이트할 수 있는 프로젝트 색상 이용하기**
>
> 프로젝트 색상의 라벨을 이용해서 프로젝트 색상을 참조할 수 있는데, 프로젝트 색상을 사용하는 색상 위젯은 프로젝트 색상과 묶이게 됩니다. 즉 수많은 속성에 대해 동일한 색상을 반복 설정하면서 느리고 긴 업데이트 과정을 겪는 대신, 다음과 같은 작업을 할 수 있다는 뜻입니다:
>
> 1. 어떤 색상을 프로젝트 색상으로 정의하려면
> 2. 사용자가 설정하고자 하는 색상 속성 옆에 있는데이터 정의 무시 위젯 을 클릭한 다음
> 3. *Color* 메뉴에 마우스를 가져가서 프로젝트 색상을 선택합니다. 그러면 속성이 표현식 `project_color('color_label')` 에 할당되어 색상 위젯이 해당 색상을 반영합니다.
> 4. 필요한만큼 두 번째와 세 번째 단계를 반복합니다.
> 5. 프로젝트 색상을 한 번 업데이트해주면, 프로젝트 색상이 사용되는 모든 곳에 변경 사항을 반영할 것입니다.

### 5.3.8 데이터소스 속성

*Data Sources* 탭에서 다음을 설정할 수 있습니다:

- *Transaction mode*: 데이터 제공자에게 편집 내용을 어떻게 전송할 것인지를 정의합니다:
  - *Local Edit Buffer*: 편집 내용을 로컬 버퍼에 저장한 다음 레이어 편집 모드를 켜고 끌 때 또는 *Save edits* 버튼을 클릭할 때 데이터 제공자에게 전송합니다.
  - *Automatic Transaction Groups*: 지원하는 데이터소스의 경우 (PostgreSQL 및 GeoPackage 데이터베이스) 동일 데이터베이스로부터 발생한 모든 테이블의 편집 상태를 서버 향 트랜잭션에서 동기화하고 실행합니다. 또한 편집 변경 내용을 로컬 버퍼에 저장하는 대신, 레이어 편집 모드를 켜고 끌 때 또는 *Save edits* 버튼을 클릭할 때 데이터베이스에 커밋되는 트랜잭션으로 직접 전송합니다.
  - *Buffered Transaction Groups*: 데이터 제공자에 상관없이, 편집 가능한 모든 레이어를 동시에 켜고 끄며 모든 편집 내용을 로컬 편집 버퍼에 저장합니다. 모든 레이어에 대한 (제공자 별) 단일 트랜잭션을 통해 변경 내용을 저장합니다.

  프로젝트 내에서 편집 중인 레이어가 하나도 없을 경우에만 이 옵션을 변경할 수 있다는 사실을 기억하십시오.

- *Evaluate default values on provider side*: PostgreSQL 테이블에 새 피처를 추가할 때, 기본값 제약조건을 가진 필드를 평가한 다음 커밋할 때가 아니라 양식을 열 때 표시합니다. 즉 *Add Feature* 양식에 있는 필드가 `nextval('serial')` 같은 표현식 대신 기대한 값을 (예를 들어 `25` 를) 표시할 것이라는 뜻입니다.
- *Remember editable layer status between sessions*: 프로젝트 저장 시 프로젝트에서 편집 가능한 모든 레이어를 그대로 기억하게 하고, 프로젝트를 다시 열 때마다 해당 레이어들을 즉시 편집 가능한 상태로 복원합니다.
- *Layers Capabilities* 를 다음과 같이 환경 설정할 수 있습니다:
  - Set (or disable) which layers are `identifiable`, i.e., will respond to the *identify tool*. By default, layers are set queryable.
  - 어떤 레이어가 `read-only` 로 나타나야 하는지, 즉 데이터소스 제공자의 성능에 상관없이 사용자가 편집할 수 없는지를 설정합니다. 낮은 수준의 보호 정책이기는 해도, 파일 기반 레이어를 작업하는 경우 최종 사용자가 데이터를 수정하는 일을 막을 수 있는 간편한 환경 설정인 것은 틀림없습니다.
  - Define which layers are `searchable`, i.e., could be queried using the *locator widget*. By default, layers are set searchable.
  - 어떤 레이어가 `required` 상태로 정의돼 있는지 설정합니다. 목록에서 체크된 레이어들은 프로젝트에서 우연히 제거되는 일이 없도록 보호됩니다.
  - Define which layers are `private`, i.e., hidden from the *Layers* panel. This is meant for accessory layers (basemap, join, lookups for value-relations, most probably aspatial layers, …) that you still need in a project but you don't want them to pollute the legend tree and other layer selection tools. If set visible, they are still displayed in the map canvas and rendered in the print layout legend. Use the *Filter legend* → *Show private layers* option in the *Layers* panel top toolbar to temporarily turned them on for any interaction.

  *Layers Capabilities* 테이블은 다음과 같은 작업을 할 수 있는 편리한 도구를 제공합니다:
  - 셀을 여러 개 선택한 다음 *Toggle Selection* 을 클릭하면 한꺼번에 체크박스의 체크 상태를 바꿀 수 있습니다.
  - *Show spatial layers only*: 레이어 목록에서 비공간 레이어를 필터링해서 빼버립니다.
  - *Filter layers*…: 환경 설정할 특정 레이어를 빨리 찾을 수 있습니다.
- *Advanced Settings* 프레임의 *Trust project when data source has no metadata*: 데이터 검사를 하지 않고 프로젝트를 불러오는 속도를 높이려면 이 옵션을 활성화하십시오. QGIS 서버 환경 또는 대용량 데이터베이스 뷰/구체화 뷰 (materialized view) 를 가진 프로젝트에서 유용한 옵션입니다. 레이어 범위를 (데이터소스 대신) QGIS 프로젝트 파일에서 읽어올 것입니다. 또 PostgreSQL 제공자를 사용하는 경우, 뷰 및 구체화 뷰에 대해 기본 키 (primary key) 의 단일성 (unicity) 을 검사하지 않을 것입니다.

### 5.3.9 관계 속성

*Relations* 탭에서 레이어들 간의 관계를 정의합니다. 이 관계는 일대일, 다대다 또는 다형성 관계 (polymorphic relation) 유형이 될 수 있습니다. 관계는 프로젝트 속성 대화창에서 정의됩니다. 어떤 레이어에 대해 관계가 존재하면, (예를 들어 피처 하나를 식별하고 그 양식을 여는 경우) 양식 뷰의 새로운 사용자 인터페이스 요소가 관련 항목들의 목록을 표시할 것입니다. 관계는, 예를 들자면 배관 또는 도로 선분의 길이에 대한 검사 이력을 표현할 수 있는 강력한 수단을 제공합니다. 여러 레이어 간 관계 설정하기 절에서 일대다 관계 지원에 관한 자세한 내용을 찾아볼 수 있습니다.

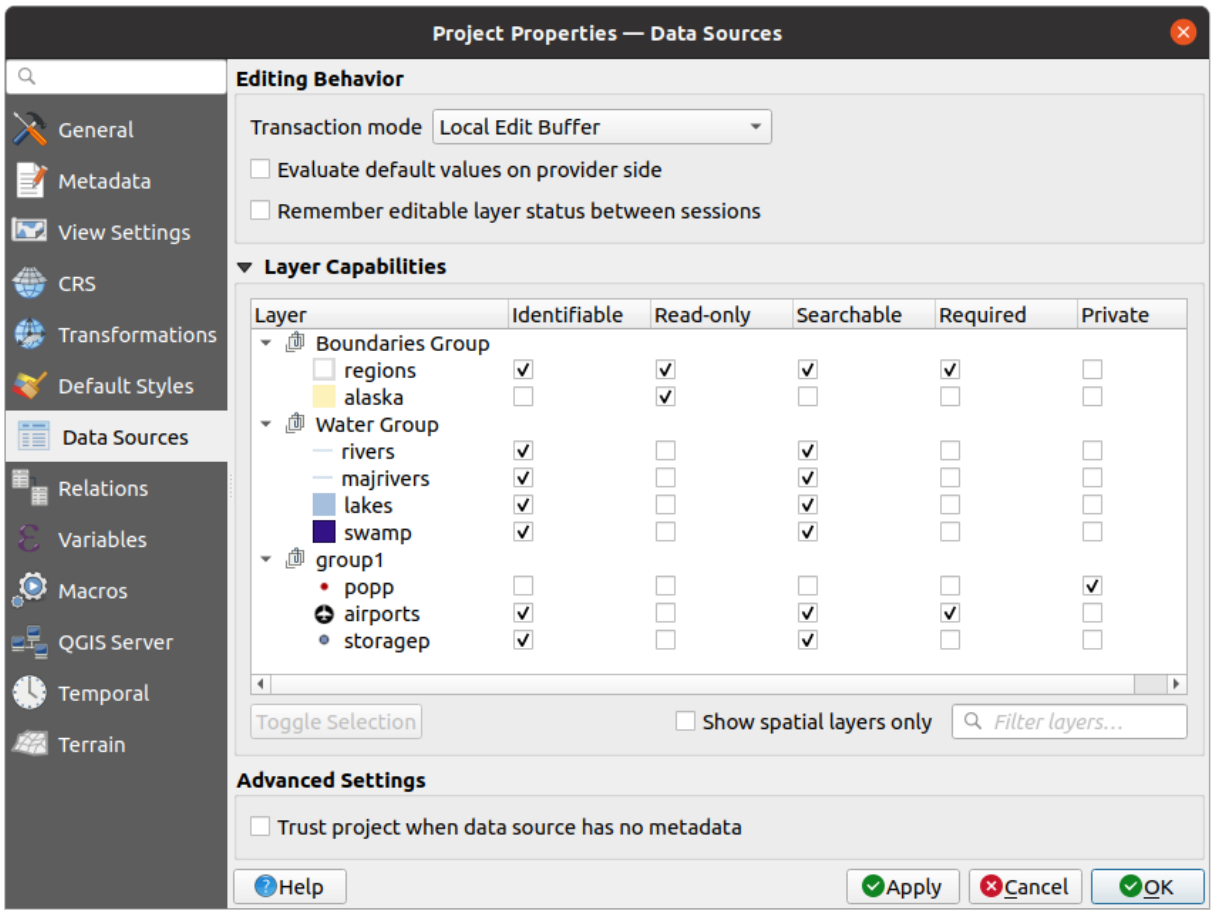



그림 5.36: 데이터소스 탭

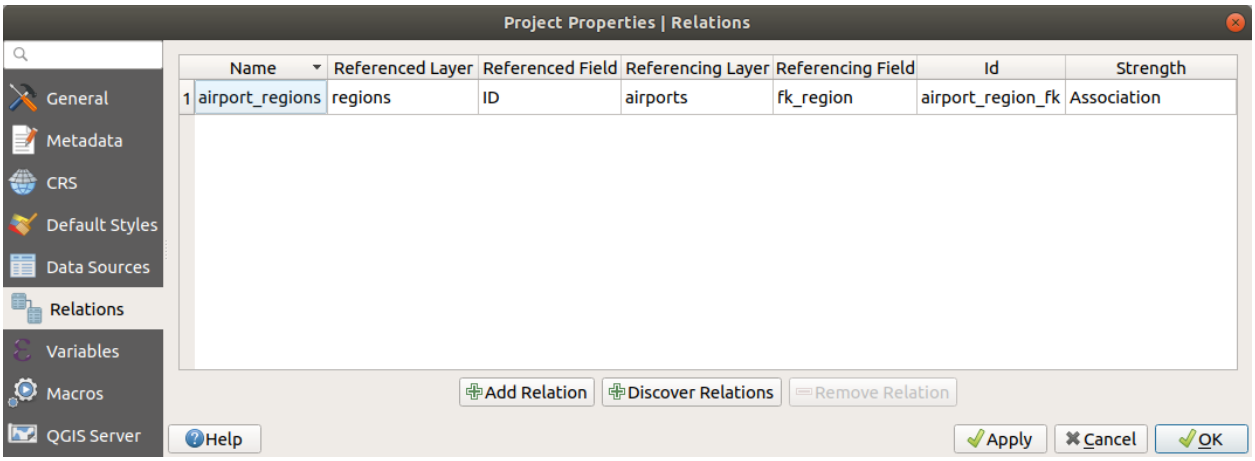



그림 5.37: 관계 탭

### 5.3.10 변수 속성

*Variables* 탭은 (모든 전체 수준 변수를 포함하는) 프로젝트 수준에서 사용할 수 있는 모든 변수 목록을 표시합니다. 게다가, 사용자가 프로젝트 수준 변수를 관리할 수도 있습니다. 버튼을 클릭해서 새 사용자 지정 프로젝트 수준 변수를 추가합니다. 마찬가지로, 목록에서 사용자 지정 프로젝트 수준 변수를 선택한 다음 버튼을 클릭해서 제거합니다. 일반 도구 장의값을 변수로 저장 절에서 변수 사용법에 대한 자세한 정보를 살펴볼 수 있습니다.

### 5.3.11 매크로 속성

*Macros* 탭은 프로젝트를 위한 파이썬 매크로를 편집하는 데 쓰입니다. 현재, `openProject()`, `saveProject()` 그리고 `closeProject()` 세 가지 매크로만 사용할 수 있습니다.

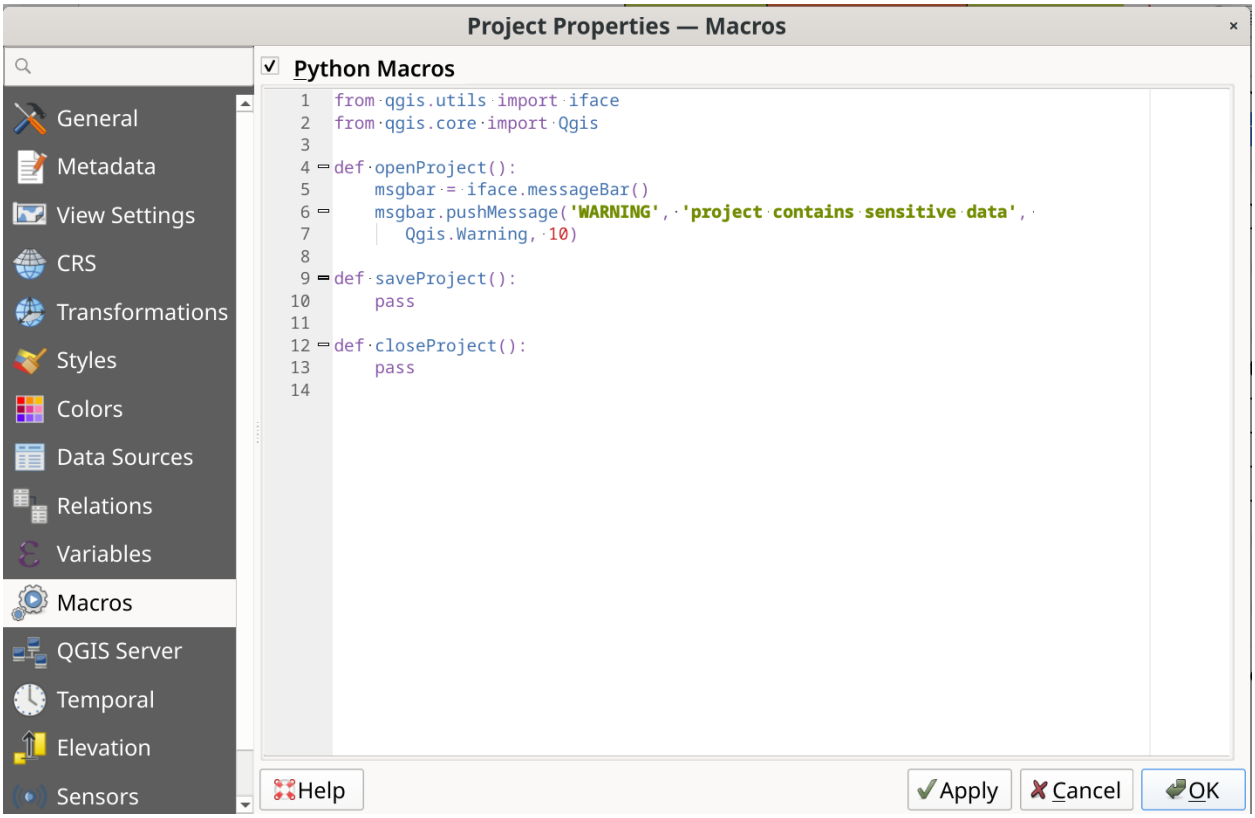



그림 5.38: 매크로 설정

### 5.3.12 QGIS 서버 속성

*QGIS Server* 탭에서는 사용자 프로젝트를 온라인 상에 게시하기 위한 환경을 설정할 수 있습니다. 이 탭에서 QGIS 서버의 WMS 및 WFS 성능, 범위, 그리고 좌표계 제약 조건에 관한 정보를 정의할 수 있습니다. Creatingwms-fromproject 및 이 절과 이어지는 부분에서 더 자세한 내용을 다루고 있습니다.

### 5.3.13 시계열 속성

*Temporal* 탭은 사용자가 직접 *Start date* 와 *End date* 를 입력해서든 또는 현재 프로젝트의 시계열 레이어로부터 계산해서든 사용자 프로젝트의 시계열 범위를 설정하는 데 쓰입니다. 시계열 범위를 설정하고 나면, 맵 캔버스시계열 탐색 을 관리하는 *Temporal controller panel* 에서 프로젝트의 시간 범위를 사용할 수 있게 됩니다.

### 5.3.14 표고 속성

The *Elevation* tab helps you configure default settings for the terrain and elevation. When any new *3d map* is created in the project, the map will default to using the same terrain settings as are defined for the project. The project elevation settings will also be respected by the *elevation profile* tool.

- *Vertical Reference System*: If the CRS of your project is compound one (including a Z dimension), then the vertical CRS used for the project will be the vertical component of the project CRS. If your project CRS is horizontal, then you can select a specific vertical CRS by clicking on the Select CRS.
- *Elevation Range*: helps you define the *Lower* and *Upper* limits for the *elevation controller* in the project.
- 다음 설정에서 *Terrain* 및 표고 옵션을 사용할 수 있습니다:
  - *Flat terrain*: *Terrain height* 설정을 사용합니다.
  - *DEM (Raster Layer)*: *Raster layer* 를 정의하기 위한 설정 및 밴드 값 및 수직 *Offset* 에 적용할 *Vertical scale* 인자를 사용합니다.
  - *Mesh*: *Mesh layer* 를 정의하기 위한 설정 및 꼭짓점 Z 값 및 수직 *Offset* 에 적용할 *Vertical scale* 인자를 사용합니다.

  3 차원 맵환경 설정 대화창 에서 이 설정들을 덮어쓸 수 있습니다.
- *Global map shading* 설정을 체크하면, 표고 속성이 활성화된 모든 활성 레이어의 표고를 바탕으로, 맵에 전체 수준 음영 효과를 적용합니다. 이때 지원하는 레이어 유형은 다음과 같습니다:
  - 래스터 레이어
  - 메시 레이어
  - 점구름 레이어

  *Method to combine*: 모든 관련 레이어의 모든 특정 위치의 표고값을 결합하고, 다음 메소드 가운데 하나에 따라 고려된 표고를 선택합니다:
  - *Highest elevation*: 최고 표고값을 고려합니다.
  - *Based on layer's order*: 레이어 트리에서 최상위에 있는 레이어의 표고를 고려합니다.

  다음 음영 옵션을 지원합니다:
  - *Eye-dome lighting*: 보다 나은 심도 (depth) 렌더링을 위해 맵 캔버스에 음영 효과를 적용합니다. 다음 파라미터들을 제어할 수 있습니다:
    * *Strength*: 심도를 보다 더 인식할 수 있도록 대조를 증가시킵니다.

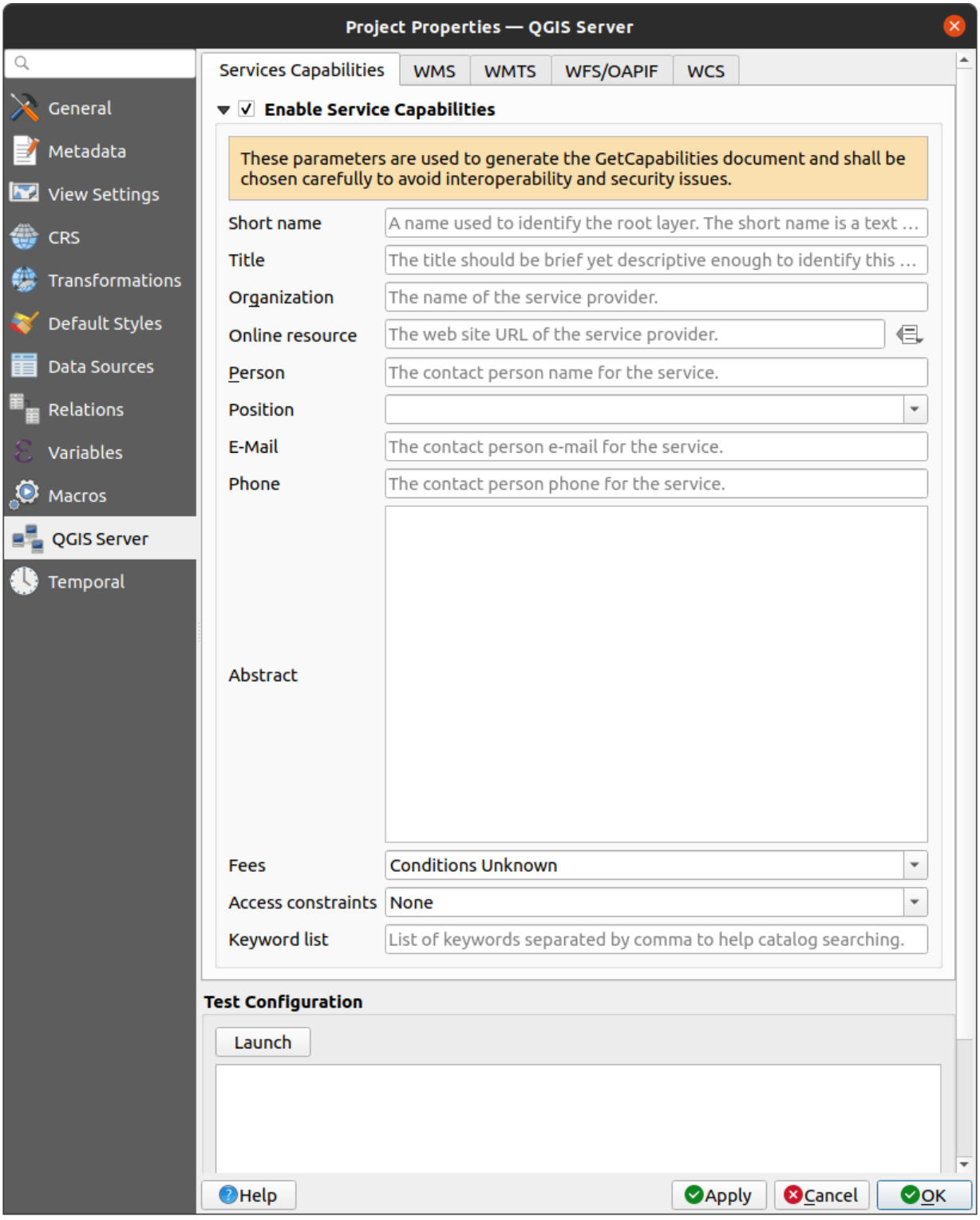



그림 5.39: QGIS 서버 설정

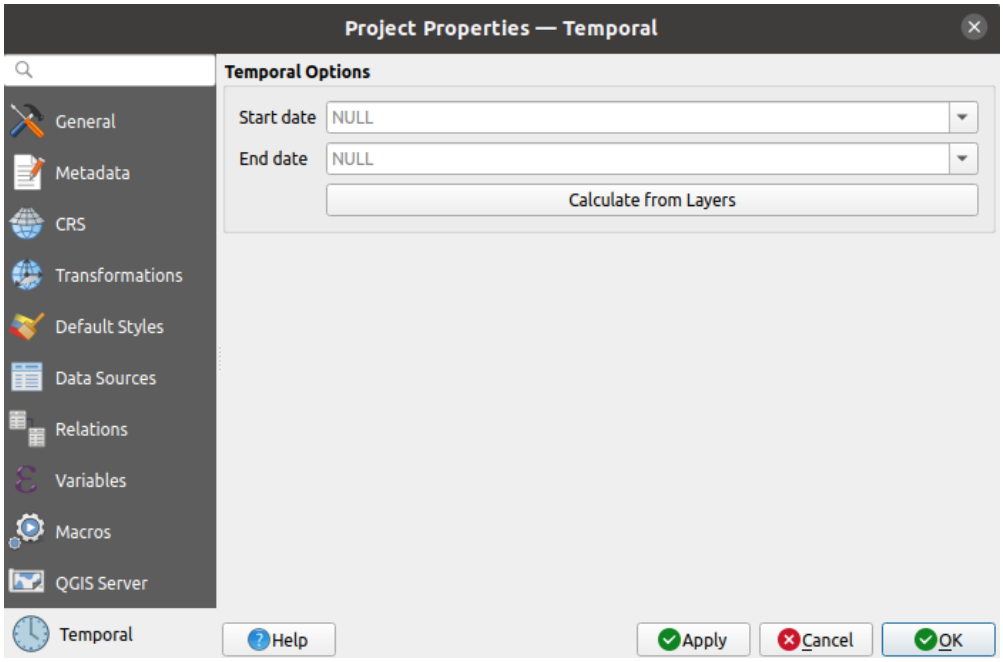



그림 5.40: 프로젝트 시계열 탭

  - * *Distance*: 사용된 픽셀이 중심 픽셀로부터 벗어난 거리를 나타내어 경계를 더 진하게 보이도록 합니다.
- – *Hillshading*: 음영 (회색조 수준) 을 이용해서 맵 위에 어느 정도의 기복 (relief) 을 형성합니다:
  - * *Z Factor*: 표고값에 대한 크기 조정 인자입니다.
  - * *Multidirectional*: 다방향 음영기복을 사용할지 여부를 결정합니다.
  - * *Azimuth*: 조명 소스의 방위각입니다.
  - * *Altitude*: 조명 소스의 앙각 (elevation angle) 입니다.

> **참고**
>
> *Layer Styling* 패널에서 *Global map shading* 속성을 가리키는 단축키를 사용할 수 있습니다.

### 5.3.15 센서 속성

*Sensors* 탭에서 센서를 환경설정하고 센서의 연결 상태를 켜고끌 수 있습니다. 센서를 활성화시키면, 센서가 백그라운드에서 수동적으로 데이터를 수집해서 표현식 및 파이썬 스크립트가 센서의 최신 데이터를 사용할 수 있도록 해줄 것입니다.

새로운 센서를 추가하려면, 아이콘을 클릭하십시오. 센서를 설정하는 하위 패널이 열리고 다음을 환경설정할 수 있을 것입니다:

- *Sensor name*: 표현식 및 파이썬 스크립트가 센서 값을 검색하기 위해 사용할 센서의 이름입니다.
- *Sensor type*: 센서의 유형을 설정합니다. TCP, UDP, 시리얼 포트 등이 있으며,

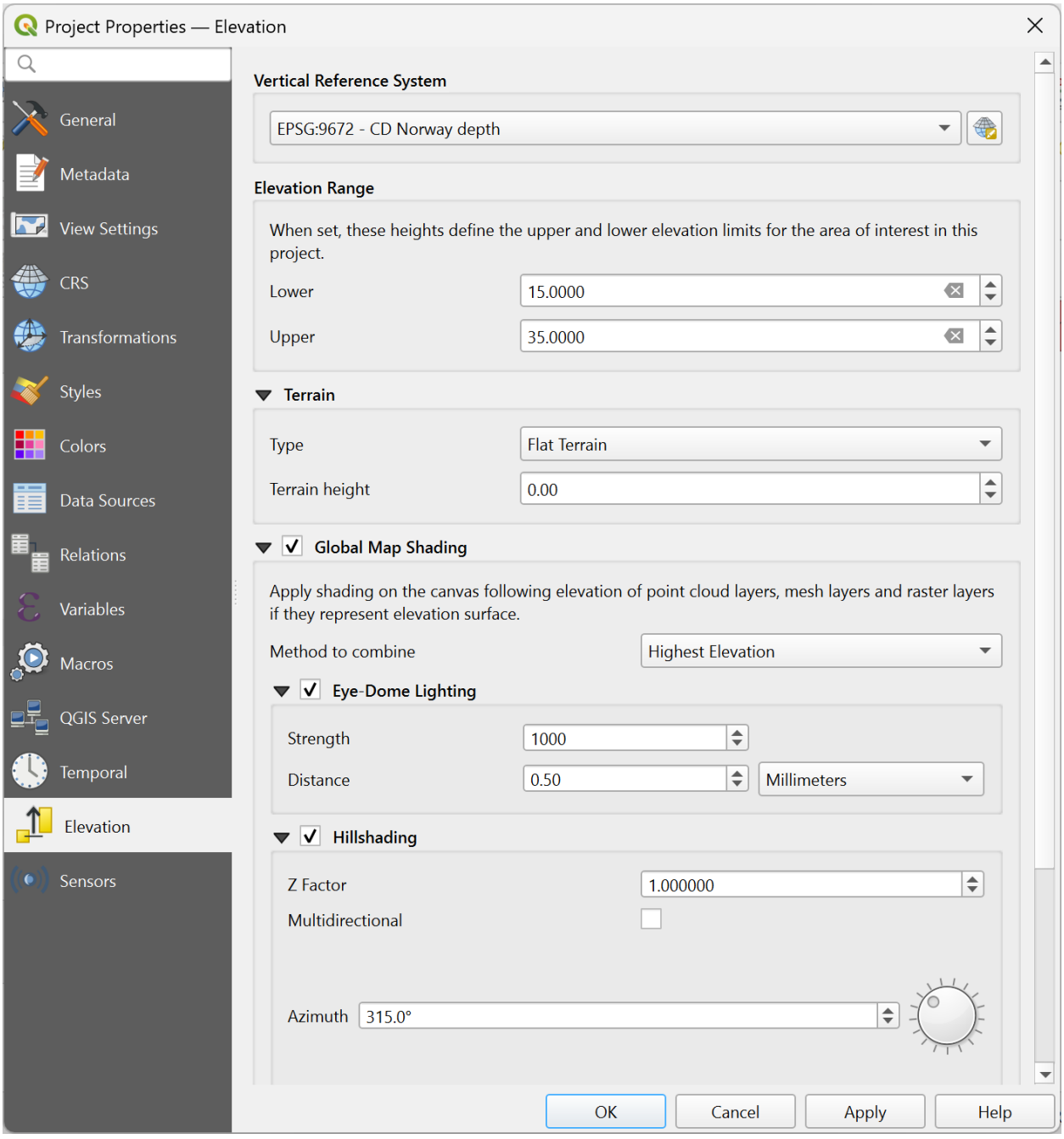



그림 5.41: Project Elevation tab

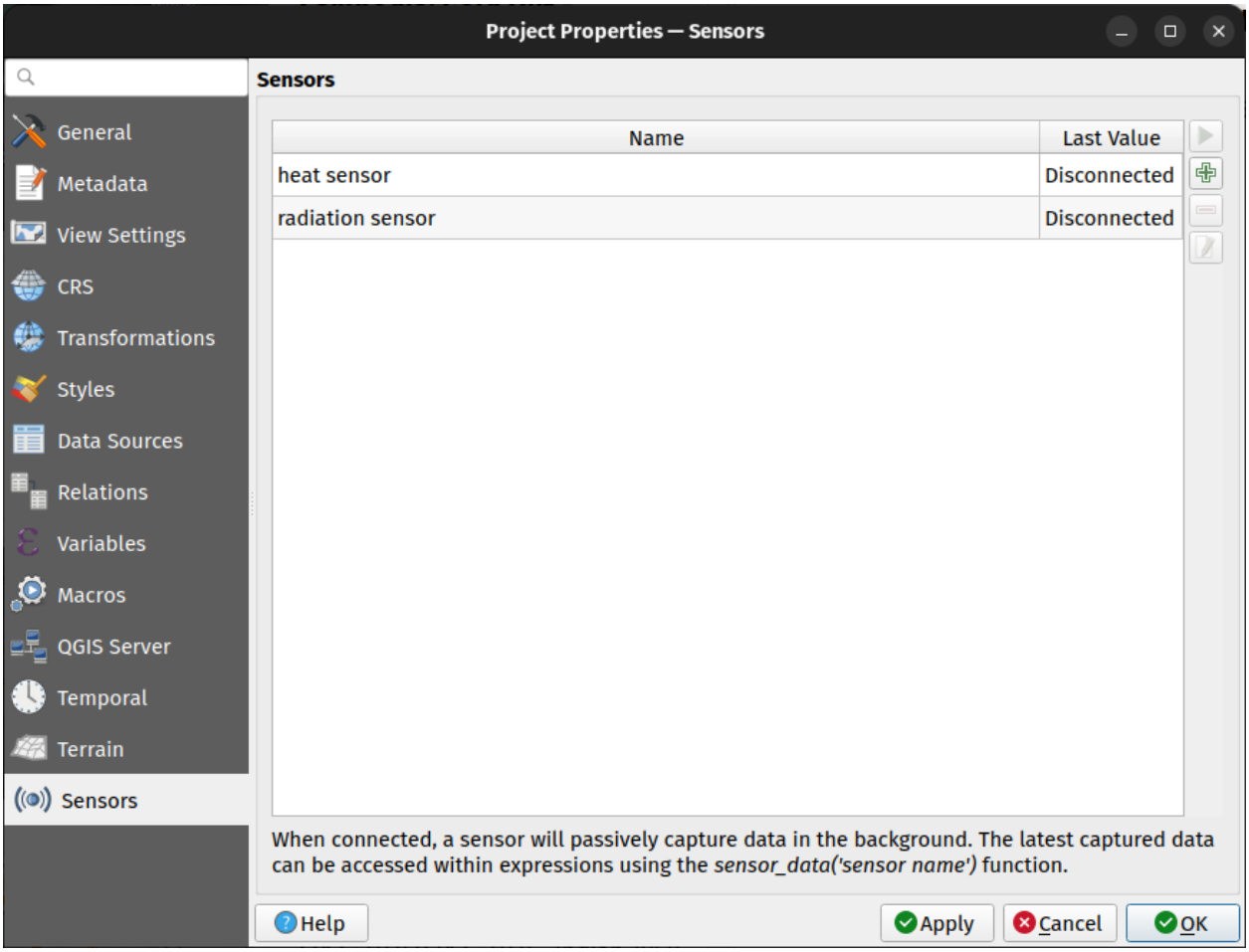



그림 5.42: 프로젝트 센서 탭

- 각 유형별 추가적인 상세 사항들을 (예를 들면 호스트명과 포트를) 설정합니다.

  For serial port sensors, comboboxes with the system's available serial ports and connection baud rate are available for their selection. A *Data frame delimiter* can be used to parse returned data. In case multiple data frames are found in the last chunk of data pushed by the serial port, all the data frames are returned. You can then use an expression to further split the returned sensor last data value into individual frames (e.g., using *string_to_array* function).

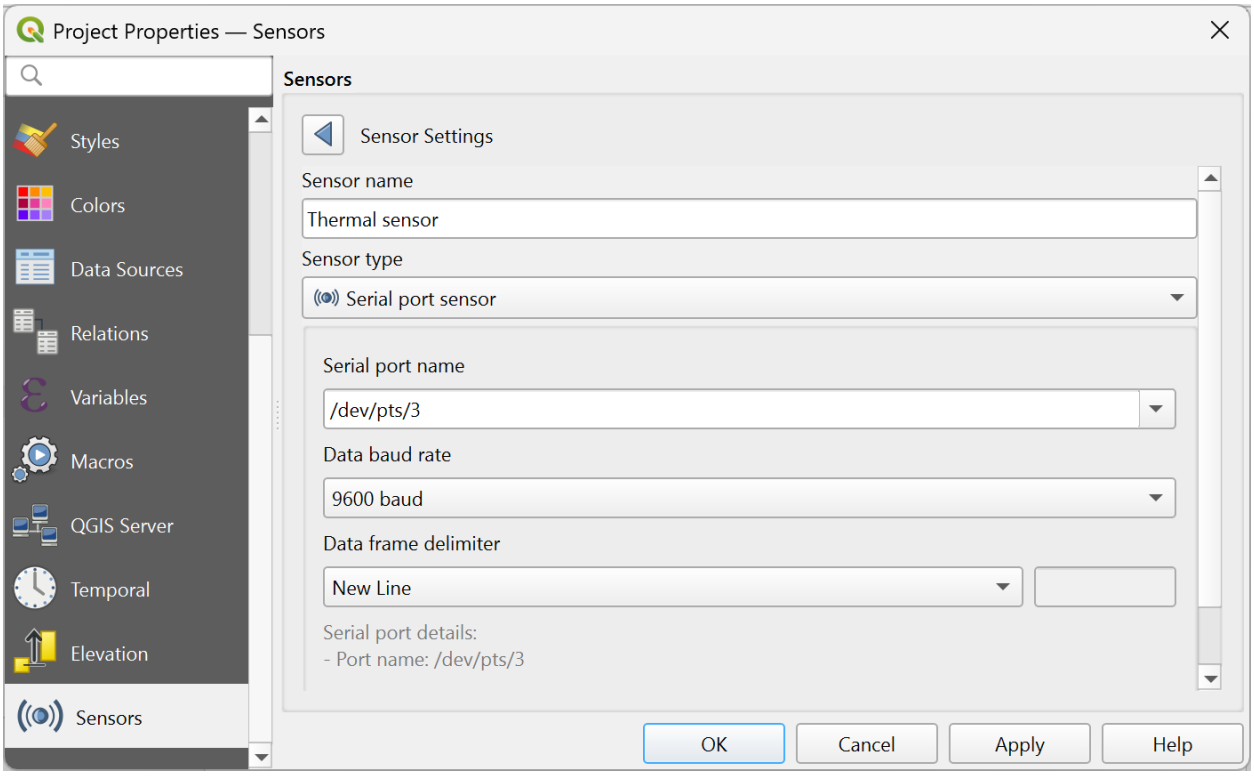



그림 5.43: 센서 설정 하위 패널

센서를 환경설정하고 나면, Start 버튼을 눌러 센서를 연결시킬 수 있습니다. 센서를 활성화하면 마지막으로 수집된 데이터가 센서 테이블의 *Last value* 열에 표시될 것입니다.

## 5.4 사용자 지정 작업

*Customization* 대화창을 통해 QGIS 사용자 인터페이스의 거의 모든 요소를 (비) 활성화할 수 있습니다. 최종 사용자가 필요로 하는 아이콘, 메뉴 또는 패널만 있는'가벼운'QGIS 버전을 제공하고자 할 때 매우 유용합니다.

> **참고**
>
> 사용자 변경 사항을 적용하려면 QGIS 를 재시작해야 합니다.

*Enable customization* 체크박스를 체크하는 일이 QGIS 사용자 지정 작업의 첫 번째 단계입니다. 툴바 및 위젯 패널을 활성화하는데, 이 패널에서 몇몇 GUI 항목을 체크 해제해서 쓸 수 없게 만들 수 있습니다.

환경을 설정할 수 있는 항목은 다음과 같습니다:

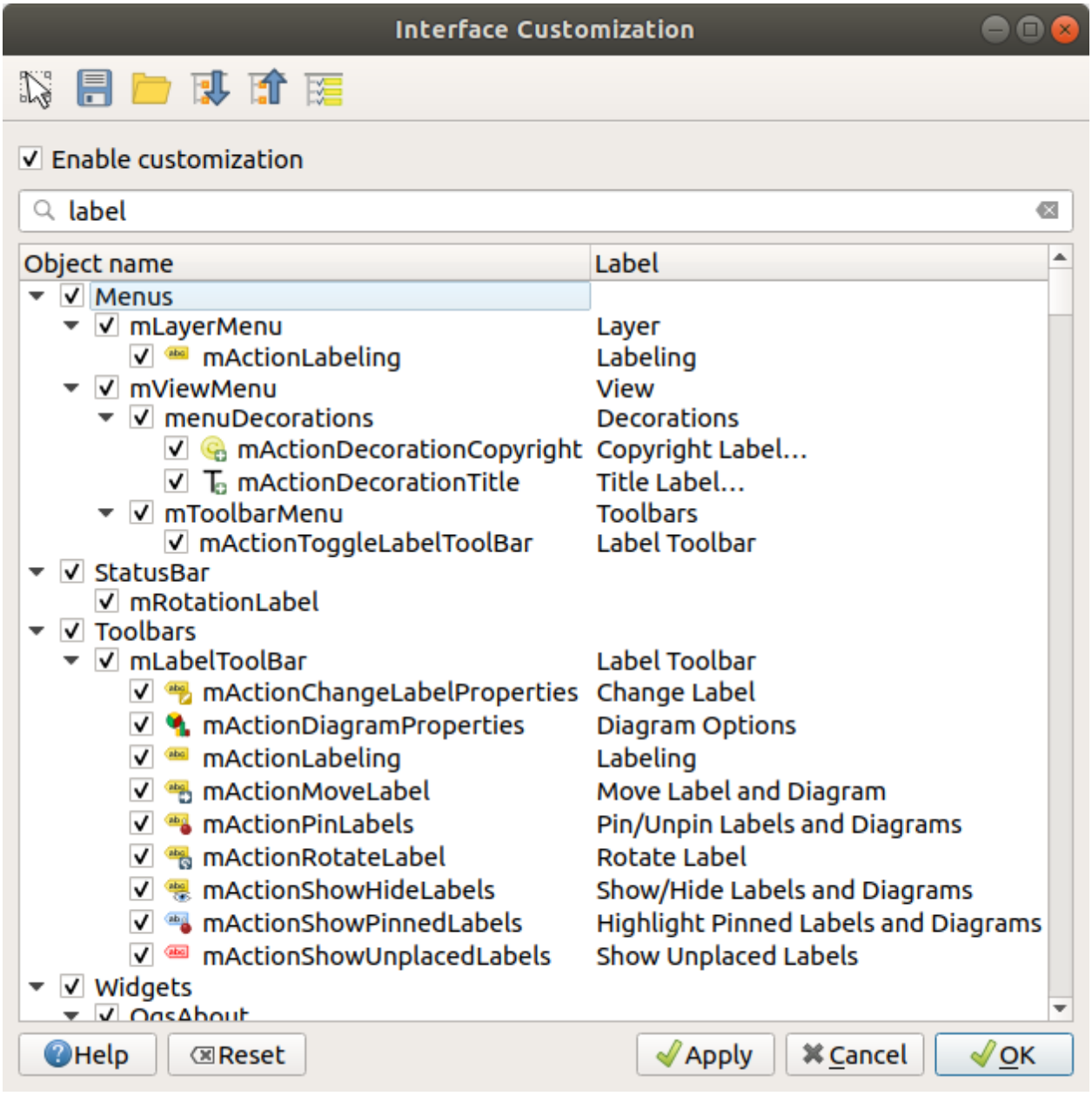



그림 5.44: 사용자 지정 작업 대화창

- **메뉴** 또는메뉴 바 에 있는 몇몇 하위 메뉴
- **패널** 전체 (패널 및 툴바 참조)
- 상태 바 에서 설명한 **상태 바** 또는 상태 바의 몇몇 항목들
- **툴바** 전체 또는 툴바의 일부 아이콘들
- 라벨, 버튼, 콤보박스 등 QGIS 의 모든 대화창의 모든 **위젯**

 Switch to catching widgets in main application 을 사용해서 QGIS 인터페이스에서 사용자가 숨기고 싶은 항목을 클릭할 수 있습니다. 그러면 QGIS 가 사용자 지정 작업 대화창에서 대응 항목을 자동적으로 체크 해제할 것입니다. *Search* 란에 항목의 명칭이나 라벨을 입력해서 항목을 찾을 수도 있습니다.

사용자 환경 설정을 마친 다음, 사용자가 변경한 사항을 확인하려면 *Apply* 또는 *OK* 를 클릭하십시오. 다음번에 QGIS 를 시작할 때 이 환경 설정을 기본값으로 이용할 것입니다.

 Save To File 버튼을 클릭하면 변경 사항들을 `.ini` 파일로도 저장할 수 있습니다. 사용자 여러 명이 QGIS 인터페이스를 공유할 수 있는 간편한 방법입니다. 다른 컴퓨터에서 이 `.ini` 파일을 불러오려면  Load from File 만 클릭하면 됩니다. 또명령어 도구 를 실행해서 서로 다른 사용 목적을 위한 다양한 설정을 저장할 수도 있습니다.

**팁**

**사전 정의된 QGIS 를 손쉽게 복구**

다음 방법들 가운데 하나를 통해 초기 QGIS GUI 환경 설정을 복구할 수 있습니다:

- 사용자 지정 작업 대화창에서  *Enable customization* 을 체크 해제하거나  Check All 버튼을 클릭합니다.
- *Settings* → *Options* 메뉴의 *System* 탭의 **Settings** 프레임에 있는 *Reset* 버튼을 클릭합니다.
- 명령 프롬프트에서 QGIS 를 다음 명령어 `qgis --nocustomization` 로 실행합니다.
- *Settings* → *Options* 메뉴의 *Advanced* 탭에 있는 *UI* → *Customization* → *Enabled* 변수의 값을 `false` 로 설정합니다. (경고 참조)

대부분의 경우, 이런 변경 사항을 적용하려면 QGIS 를 재시작해야 합니다.

## 5.5 키보드 단축키

QGIS 는 많은 기능에 대해 기본 키보드 단축키를 제공합니다. 메뉴 바 에서 기본 단축키를 찾아볼 수 있습니다. 또한 *Settings* → *Keyboard Shortcuts*…메뉴 옵션에서 기본 키보드 단축키를 변경하고 QGIS 기능에 새 키보드 단축키를 추가할 수도 있습니다.

설정 방법은 아주 간단합니다. 대화창 상단에 있는 검색란을 통해 특정 기능을 찾은 다음 목록에서 기능을 선택하고 다음을 클릭하십시오:

- *Change* 를 클릭하고 새 단축키로 할당할 키 조합을 입력하십시오.
- 할당된 단축키를 삭제하려면 *Set None* 을 클릭하십시오.
- 또는 단축키를 원래 기본값으로 되돌리려면 *Set default* 를 클릭하십시오.

사용자 지정하고자 하는 다른 모든 도구에 대해서도 이와 같이 진행하십시오. 환경 설정을 완료한 다음 변경 사항을 적용하려면, *Close* 버튼으로 대화창을 그냥 닫기만 하면 됩니다. *Save* 버튼을 누르면 변경 사항들을 사용자 단축키만 담은 또는 모든 단축키를 담은 `.XML` 파일로, 또는 모든 단축키를 담은 `.PDF` 파일로 저장할 수 있습니다. 이렇게 저장한 파일은 또다른 QGIS 설치본에서 *Load* 버튼으로 불러올 수도 있습니다.




그림 5.45: 단축키 옵션 정의

## 5.6 고급 설정으로 QGIS 실행

### 5.6.1 명령 줄과 환경 변수

We've seen that *launching QGIS* is done as for any application on your OS. QGIS provides command line options for more advanced use cases (in some cases you can use an environment variable instead of the command line option). To get a list of the options, enter `qgis --help` on the command line, which returns:

```
QGIS is a user friendly Open Source Geographic Information System.
Usage: /usr/bin/qgis.bin [OPTION] [FILE]
  OPTION:
        [-v, --version]     display version information and exit
        [-s, --snapshot filename]   emit snapshot of loaded datasets to given file
        [-w, --width width] width of snapshot to emit
        [-h, --height height]       height of snapshot to emit
        [-l, --lang language]       use language for interface text (changes existing␣
↪override)
        [-p, --project projectfile] load the given QGIS project
        [-e, --extent xmin,ymin,xmax,ymax]  set initial map extent
        [-n, --nologo]      hide splash screen
        [-V, --noversioncheck]      don't check for new version of QGIS at startup
        [-P, --noplugins]   don't restore plugins on startup
        [-B, --skipbadlayers]     don't prompt for missing layers
        [-C, --nocustomization]     don't apply GUI customization
        [-z, --customizationfile path]      use the given ini file as GUI␣
↪customization
        [-g, --globalsettingsfile path]     use the given ini file as Global Settings␣
↪(defaults)
        [-a, --authdbdirectory path] use the given directory for authentication␣
↪database
        [-f, --code path]   run the given python file on load
        [-F, --py-args arguments]   arguments for python. These arguments will be␣
↪available for each python execution via 'sys.argv' including the file specified by
↪'--code'. All arguments till '--' are passed to python and ignored by QGIS.
        [-d, --defaultui]   start by resetting user ui settings to default
        [--hide-browser]        hide the browser widget
        [--dxf-export filename.dxf]     emit dxf output of loaded datasets to given␣
↪file
        [--dxf-extent xmin,ymin,xmax,ymax]      set extent to export to dxf
        [--dxf-symbology-mode none|symbollayer|feature] symbology mode for dxf output
        [--dxf-scale-denom scale]       scale for dxf output
        [--dxf-encoding encoding]       encoding to use for dxf output
        [--dxf-map-theme maptheme]      map theme to use for dxf output
        [--take-screenshots output_path]        take screen shots for the user␣
↪documentation
        [--screenshots-categories categories]   specify the categories of screenshot␣
↪to be used (see QgsAppScreenShots::Categories).
        [--profile name]        load a named profile from the user's profiles folder.
        [-S, --profiles-path path]  path to store user profile folders. Will create␣
↪profiles inside a {path}\profiles folder
        [--version-migration]   force the settings migration from older version if␣
↪found
        [--openclprogramfolder]         path to the folder containing the sources for␣
↪OpenCL programs.
        [--help]                this text
        [--]            treat all following arguments as FILEs
```

(다음 페이지에 계속)

(이전 페이지에서 계속)

```
FILE:
  Files specified on the command line can include rasters,
  vectors, and QGIS project files (.qgs and .qgz):
   1. Rasters - supported formats include GeoTiff, DEM
      and others supported by GDAL
   2. Vectors - supported formats include ESRI Shapefiles
      and others supported by OGR and PostgreSQL layers using
      the PostGIS extension
```

**팁**

**명령 줄 인자 사용 예시**

명령 줄에서 하나 이상의 데이터 파일을 지정해서 QGIS 를 실행할 수 있습니다. 예를 들어 지금 `qgis_sample_data` 디렉터리에 있다면, 다음과 같은 명령어를 사용해서 QGIS 실행 시 벡터 레이어와 래스터 파일을 불러오도록 할 수 있습니다: `qgis ./raster/landcover.img ./gml/lakes.gml`

#### --version

이 옵션은 QGIS 버전 정보를 반환합니다.

#### --snapshot

이 옵션은 현재 뷰의 스냅샷을 PNG 포맷으로 생성할 수 있게 해줍니다. 사용자가 수많은 프로젝트를 가지고 있어서 사용자 데이터로부터 스냅샷을 생성하고자 하거나, 동일 프로젝트에서 데이터가 업데이트되어 스냅샷을 생성해야 할 때 유용합니다.

현재 이 옵션은 800x600 픽셀 해상도의 PNG 파일을 생성합니다. 이 크기는 `--width` 및 `--height` 인자를 사용해 조정할 수 있습니다. 또 `--snapshot` 인자 뒤에 파일명을 추가할 수도 있습니다. 다음은 그 예시입니다:

```
qgis --snapshot my_image.png --width 1000 --height 600 --project my_project.qgs
```

#### --width

이 옵션은 생성될 스냅샷의 너비를 반환합니다. (`--snapshot` 과 함께 사용)

#### --height

이 옵션은 생성될 스냅샷의 높이를 반환합니다. (`--snapshot` 과 함께 사용)

**--lang**

QGIS 는 사용자 로케일을 기반으로 정확한 현지화 언어를 선택합니다. 만약 사용자 언어를 변경하고자 한다면, 언어 코드를 지정할 수 있습니다. 예를 들어, `qgis --lang it` 명령어는 QGIS 를 이탈리아 현지화 설정으로 실행할 것입니다.

**--project**

QGIS 를 기존 프로젝트 파일과 함께 실행할 수도 있습니다. 명령 줄 옵션 `--project` 뒤에 사용자의 프로젝트 명칭만 추가하면 불러온 프로젝트에 있는 모든 레이어와 함께 QGIS 를 실행할 것입니다.

**--extent**

QGIS 를 특정한 맵 범위로 실행하려면 이 옵션을 사용하십시오. 사용자가 지정한 범위 상자를 쉼표로 구분된 다음 순서로 추가해야 합니다:

```
--extent xmin,ymin,xmax,ymax
```

이 옵션은 특정 프로젝트를 원하는 범위로 열기 위해 `--project` 옵션과 함께 사용해야 의미가 있을 수도 있습니다.

**--nologo**

QGIS 실행 시 로고 이미지를 숨깁니다.

**--noversioncheck**

QGIS 실행 시 새 버전을 검색하지 않습니다.

**--noplugins**

실행 시 플러그인 관련 오류가 생기는 경우, 이 옵션을 사용하면 플러그인을 불러오지 않습니다. 실행 후 플러그인 관리자에서 플러그인을 불러올 수 있습니다.

**--nocustomization**

이 옵션을 사용하면, 실행 시 모든 기존사용자 지정 *GUI* 를 무시할 것입니다. 즉 사용자가 숨겼던 모든 버튼, 메뉴 항목, 툴바 등등이 QGIS 실행 시 나타날 것입니다. 이 변경 사항은 자동적으로 저장되지 않습니다. QGIS 를 이 옵션 없이 다시 시작하면, 사용자 지정 GUI 가 다시 적용될 것입니다.

이 옵션은 사용자가 제거했던 도구들에 일시적으로 접근하고자 할 때 유용합니다.

#### --skipbadlayers

이 옵션을 사용하면, QGIS 실행 시 *Handle Unavailable Layers* 대화창이 뜨는 것을 막을 수 있습니다. 누락된 레이어를 계속 사용 불가로 유지한 채 프로젝트 파일을 불러옵니다. 이 문제에 대한 자세한 내용은깨진 파일 경로 처리 절을 참조하세요.

#### --customizationfile

이 옵션을 사용하면, 사용자가 사용자 지정 UI 파일을 지정해서 QGIS 실행 시 사용할 수 있습니다.

#### --globalsettingsfile

이에 대응하는 환경 변수는 `QGIS_GLOBAL_SETTINGS_FILE` 입니다.

이 옵션을 사용하면, 기본 설정으로도 알려진 전체 수준 설정 파일 (`.ini`) 의 경로를 지정할 수 있습니다. 지정한 파일 안에 있는 설정들은 그때그때 즉시 처리되는 (inline) 기본 설정들은 대체하지만, 사용자 프로파일의 설정들을 우선할 것입니다.

QGIS 는 기본 전체 수준 설정 파일들을 다음 순서대로 검색하고 가장 먼저 검색된 파일만 사용할 것입니다:

- 명령 줄 파라미터가 지정하는 경로
- 환경 변수가 정의하는 경로
- 영구 응용 프로그램 데이터를 저장할 수 있는 AppDataLocation 폴더. 이 폴더는 사용자 또는 시스템 어드민이 관리하기 때문에 설치 파일이 건드릴 수 없으며, 명령 줄 파라미터 전송 또는 환경 변수 설정과 같은 추가적인 설정을 해야 할 필요가 없습니다. OS 에 따라, 이 폴더는:
  - `$HOME/.local/share/QGIS/QGIS3/`
  - `C:\Users\<username>\%AppData%\Roaming\QGIS\QGIS3\`
  - `$HOME/Library/Application Support/QGIS/QGIS3/`
- the installation directory, i.e., `your_QGIS_package_path/resources/qgis_global_settings.ini`.

현재로서는 설정을 기록할 파일을 지정할 방법이 없습니다. 따라서 원래 설정 파일을 복사해서, 재명명한 다음, 수정하는 방법을 쓸 수 있습니다.

`qgis_global_setting.ini` 파일의 경로를 네트워크 공유 폴더로 설정하면, 시스템 관리자가 파일 하나를 편집해서 여러 시스템의 전체 수준 설정 및 기본값을 변경할 수 있습니다.

#### --authdbdirectory

이 옵션은 `--globalsettingsfile` 과 비슷하지만, 인증 데이터베이스를 저장하고 불러올 디렉터리를 가리키는 경로를 정의합니다.

#### --code

이 옵션을 사용하면 QGIS 를 실행한 직후 지정한 파이썬 파일을 실행할 수 있습니다.

예를 들어, 다음과 같은 내용의 `load_alaska.py` 라는 파이썬 파일을 가지고 있을 때:

```
from qgis.utils import iface
raster_file = "/home/gisadmin/Documents/qgis_sample_data/raster/landcover.img"
layer_name = "Alaska"
iface.addRasterLayer(raster_file, layer_name)
```

Assuming you are in the directory where the file `load_alaska.py` is located, you can start QGIS, load the raster file `landcover.img` and give the layer the name 'Alaska'using the following command:

```
qgis --code load_alaska.py
```

#### --py-args

This option allows to pass arguments to the script provided via the *–code* parameter and more generally to each python execution. After `--py-args`, it is not possible to pass any other parameter than `–` to QGIS. All arguments after `--py-args` till `--` are passed over to the python interpreter and ignored by QGIS.

```
qgis --code /home/user/test.py --py-args --specialScriptArguments "a text arg" 'and␣
↪another arg' -- layer1 layer2
```

In the above code, `test.py` will have this content in `sys.argv`: ['/home/user/test.py', '–specialScriptArguments', 'a text arg', 'and another arg']. `layer1` and `layer2` will be normally handled by QGIS as layers to load.

#### --defaultui

QGIS 를 이 옵션으로 실행하면, UI 를 기본 설정으로 **항구적으로 재설정** 합니다. 이 옵션은 패널 및 툴바의 가시성, 위치 및 크기를 복구할 것입니다. UI 설정을 다시 변경하지 않는 이상, 다음 세션에서도 기본 UI 설정을 사용할 것입니다.

이 옵션이사용자 지정 *GUI* 에는 어떤 영향도 끼치지 않는다는 사실을 기억하십시오. `--defaultui` 옵션을 사용하더라도 GUI 환경을 사용자 지정해서 숨겨진 (상태 바 같은) 항목들은 숨겨진 채로 남을 것입니다. `--nocustomization` 옵션도 참조하세요.

#### --hide-browser

실행 시 사용자 인터페이스에서 *Browser* 패널을 숨깁니다. 툴바의 공백 부분에 오른쪽 클릭을 하거나 *View* → *Panels* 메뉴를 클릭해서 탐색기 패널을 활성화시킬 수 있습니다. (  Linux KDE 설치본에서는 *Settings* → *Panels* 메뉴로 접근합니다.)

다시 활성화시키지 않는 이상, 탐색기 패널은 다음 세션에서도 숨겨진 채일 것입니다.

#### --dxf-*

QGIS 프로젝트를 DXF 파일로 내보내고자 할 때 이 옵션을 사용할 수 있습니다. 몇 가지 옵션을 사용할 수 있습니다:

- *–dxf-export*: 레이어를 내보낼 DXF 파일명
- *–dxf-extent*: 최종 DXF 파일의 범위
- *–dxf-symbology-mode*: 몇 가지 값을 사용할 수 있습니다 — `none` (심볼 없음), `symbollayer` (심볼 레이어의 심볼), `feature` (피처 심볼)
- *–dxf-scale-denom*: 심볼의 축척 분모
- *–dxf-encoding*: 파일 인코딩
- *–dxf-map-theme*: 레이어 트리 환경 설정에서맵 테마 를 선택

#### --take-screenshots

사용자 문서 용 스크린샷을 찍습니다. 문서 스크린샷을 어느 카테고리/단락으로 생성해야 할지를 필터링하기 위한 `--screenshots-categories` 옵션과 함께 사용할 수 있습니다. (QgsAppScreenShots::Categories 참조)

#### --profile

Loads QGIS using a specific profile from the user's profile folder. If the named profile doesn't exist, it will be created. This option takes precedence over *user profile startup setting*.

#### --profiles-path

이 옵션은 프로파일 (사용자 설정) 을 저장하고 불러올 경로를 선택할 수 있게 해줍니다. `{path}\profiles` 폴더에 프로파일을 생성하는데, 이 프로파일은 설정, 설치한 플러그인, 공간 처리 모델 및 스크립트 등을 담고 있습니다.

예를 들자면 이 옵션을 사용해서 USB 저장 장치에 사용자의 모든 플러그인과 설정을 담아 가지고 다니 거나, 파일 공유 서비스를 통해 서로 다른 컴퓨터들끼리 설정을 공유할 수 있습니다.

이에 대응하는 환경 변수는 `QGIS_CUSTOM_CONFIG_PATH` 입니다.

#### --version-migration

구 버전의 설정을 찾았을 경우 (예를 들어 QGIS 2.18 버전의 `.qgis2` 폴더) 이 옵션은 해당 설정을 기본 QGIS 프로파일로 가져올 것입니다.

#### --openclprogramfolder

이 옵션을 사용하면 사용자의 OpenCL 프로그램을 가리키는 대안 경로를 지정할 수 있습니다. 개발자가 기존 프로그램을 대체할 필요없이 새 버전을 테스트하는 데 유용합니다.

이에 대응하는 환경 변수는 `QGIS_OPENCL_PROGRAM_FOLDER` 입니다.

--

This option allows to pass following arguments as files to open in QGIS. They can be supported rasters or vectors files, QGIS layer definition files or project files.

```
qgis -- /home/user/project.qgs /home/user/path_to_data/layer1.gpkg
```

The above code will start QGIS, open the `project` file and load `layer1` in it.

### 5.6.2 조직 내에서 QGIS 활용하기

어떤 조직 내에서 사용자 지정 환경 설정 파일을 이용해서 QGIS 를 구동해야 하는 경우, 먼저 사용자 지정 환경 설정 파일에 `your_QGIS_package_path/resources/qgis_global_settings.ini` 에 있는 기본 설정 파일의 내용을 복사/붙여넣기해야 합니다. 이 기본 설정 파일은 이미 `[]` 로 시작하는 블록으로 식별되는 몇몇 기본 절들을 담고 있습니다. 이 기본값들을 그대로 유지한 채 사용자 고유의 절들을 사용자 지정 환경 설정 파일 맨 아래에 추가할 것을 권장합니다. 파일 안에서 어떤 절이 중복되는 경우, QGIS 는 위에서 아래 방향으로 가장 마지막에 있는 절을 사용할 것입니다.

QGIS 버전 확인을 비활성화하려면 `allowVersionCheck=false` 로 변경하면 됩니다.

QGIS 를 새로 설치한 다음 마이그레이션 창 (migration window) 이 나타나지 않기를 원한다면, 다음 단락이 필요할 것입니다:

```
[migration]
fileVersion=2
settings=true
```

전체 수준 범위 (global scope) 에서 사용자 지정 변수를 추가하려면:

```
[variables]
organisation="Your organization"
```

A lot of settings can be set using the `.INI` file such as WMS/WMTS, PostgreSQL connections, proxy settings, maptips, default values for algorithms parameters···For example, if you want to configure organization wide default values for some algorithms it would be handy if you could configure it in the prepared ini-file for all users. If the users then open the toolbox to execute an algorithm, the default values will be there. To set the default value, e.g., for `GRID_SIZE` parameter for the `native:intersection` algorithm, you could write the following into the `.INI` file:

```
[Processing]
DefaultGuiParam\native%3Aintersection\GRID_SIZE=0.01
```

QGIS GUI 로부터 모든 설정을 준비할 수는 없습니다. 하지만 `.INI` 파일 설정의 가능성을 발견하기 위해, 사용자가 QGIS 데스크톱에서 바라는 환경을 설정한 다음 텍스트 편집기를 통해 사용자 프로파일에 위치한 `.INI` 파일에서 그 내용을 검색해볼 것을 추천하는 바입니다.

마지막으로, 사용자 지정 파일의 경로를 가리키는 환경 변수 `QGIS_GLOBAL_SETTINGS_FILE` 을 설정해야 합니다.

또한 QGIS 시스템 디렉터리나 QGIS 사용자 프로파일 폴더에 있는 파이썬 매크로, 색상표, 조판기 템플릿, 프로젝트 템플릿 등과 같은 파일들도 활용할 수 있습니다.

- 조판기 템플릿은 `composer_templates` 디렉터리에서 활용해야만 합니다.
- 프로젝트 템플릿은 `project_templates` 디렉터리에서 활용해야만 합니다.
- 사용자 지정 파이썬 매크로는 `python` 디렉터리에서 활용해야만 합니다.

CHAPTER 6

# 투영 작업

좌표계 또는 CRS(Coordinate Reference System) 는 지표면 상의 위치에 수치 좌표를 결합시키는 체계를 말합니다. QGIS 는 약 7,000 개의 표준 좌표계를 지원하고 있는데, 각 좌표계는 서로 다른 사용 사례 (use case) 와 장점 및 단점을 가지고 있습니다. 사용자의 QGIS 프로젝트와 데이터를 위한 적절한 좌표계를 선택하는 일은 쉽지 않은 업무일 수도 있지만, 다행히도 QGIS 는 이 선택 과정 전반에 걸쳐 사용자를 안내하며, 서로 다른 좌표계를 작업하는 일을 가능한 한 투명하고 정확하게 만들어줍니다.

## 6.1 투영 지원 개요

QGIS 는 알려진 좌표계 약 7,000 개를 지원합니다. 이 표준 좌표계들은 유럽 석유 탐사 연합 (EPSG; European Petroleum Search Group) 과 프랑스 국립지리원 (IGNF; Institut Geographique National de France) 이 정의한 좌표계에 기반하며, 기저"Proj"투영체 라이브러리를 통해 QGIS 에서 사용할 수 있습니다. 일반적으로 이 표준 좌표계들은"EPSG"또는"IGNF"같은 조직명이 기관 (authority) 이고 특정 좌표계에 할당된 유일한 숫자가 코드인 기관: 코드 (authority:code) 조합을 사용해서 식별됩니다. 예를 들어, 흔히 쓰이는 WGS 84 위도/경도 좌표계는 식별자 `EPSG:4326` 으로 알려져 있으며, 웹 맵핑 표준 좌표계는 `EPSG:3857` 입니다.

사용자 지정, 사용자 생성 좌표계는 사용자 좌표계 데이터베이스에 저장됩니다. 사용자 지정 좌표계 관리에 대한 자세한 내용은사용자 정의 좌표계 를 참조하십시오.

## 6.2 레이어 좌표계

In order to correctly project data into a specific target CRS, either your data must contain information about its coordinate reference system or you will need to manually assign the correct CRS to the layer. For PostgreSQL layers, QGIS uses the spatial reference identifier that was specified when that PostgreSQL layer was created. For data supported by GDAL, QGIS relies on the presence of a recognized means of specifying the CRS. For instance, for the Shapefile format this is a file containing an ESRI Well-Known Text (WKT) representation of the layer's CRS. This projection file has the same base name as the `.shp` file and a `.prj` extension. For example, `alaska.shp` would have a corresponding projection file named `alaska.prj`.

QGIS 는 레이어를 불러올 때마다 자동적으로 레이어의 정확한 좌표계를 판별하려 합니다. 어떤 경우 — 예를 들어 불러온 레이어가 해당 정보를 담고 있지 않은 경우 — 이는 불가능합니다. 사용자는 QGIS 가 레이어의 정확한 좌표계를 자동적으로 판별할 수 없을 때마다 어떻게 반응할지를 환경 설정할 수 있습니다:

1. *Settings* → *Options…* → *CRS* 메뉴를 클릭합니다.

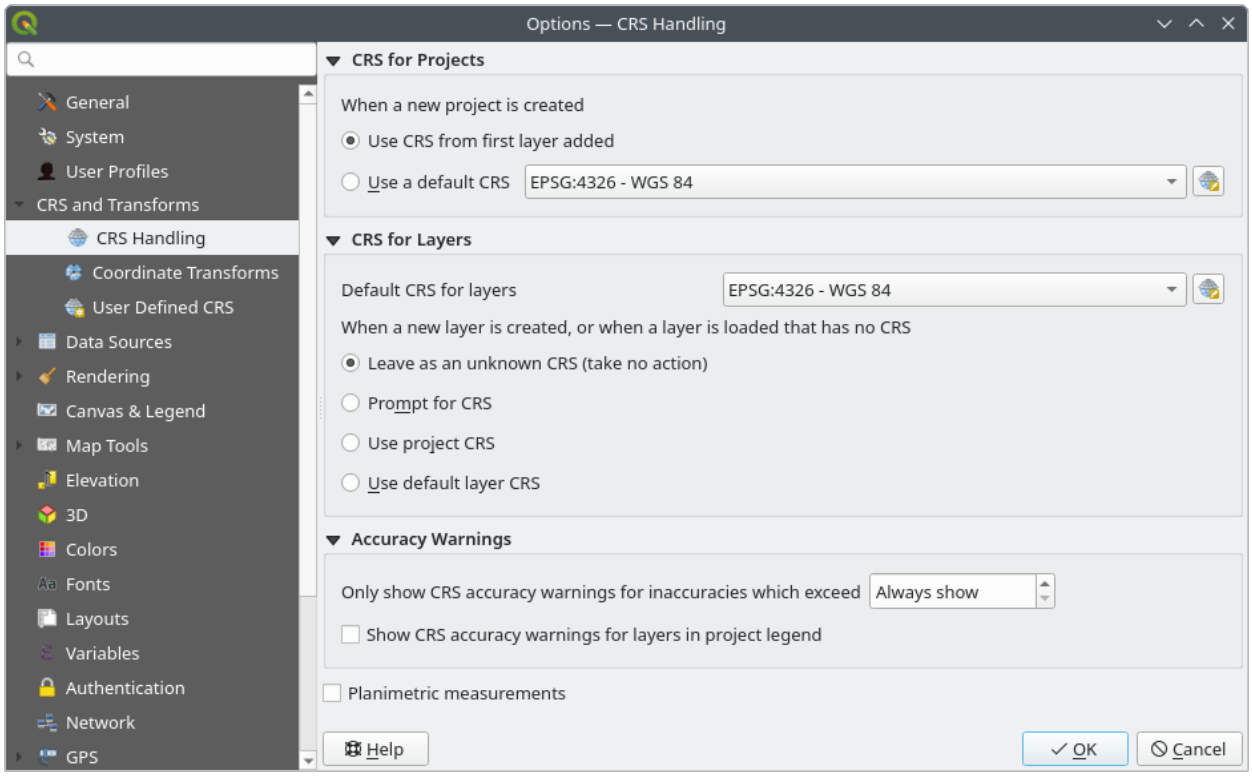



그림 6.1: QGIS 옵션 대화창의 CRS 탭

2. *CRS for layers* 그룹에 있는 *when a new layer is created, or when a layer is loaded that has no CRS* 에서 불러온 레이어가 좌표계 정보를 가지고 있지 않을 경우 어떻게 반응할지 설정하십시오. 다음 옵션 가운데 하나를 선택할 수 있습니다:

   - *Leave as unknown CRS (take no action)*: 좌표계 정보가 없는 레이어를 불러왔을 때 좌표계를 선택하라고 물어보지 않고, 다음에 선택하도록 미룰 것입니다. 한번에 많은 레이어들을 불러올 때 편리합니다. *Layers* 패널에서 좌표계가 없는 레이어는 옆의 아이콘으로 식별할 수 있습니다. 이들은 비참조 (un-referenced) 되어, 레이어의 좌표를 지구와는 관계없는 순수한 수치로 취급할 것입니다. 즉프로젝트가 좌표계를 갖지 않도록 설정 했을 때의 모든 레이어처럼 취급될 것입니다.
   - *Prompt for CRS*: 사용자에게 좌표계를 직접 선택하도록 요청할 것입니다. 정확하게 선택하는 것이 중요합니다. 잘못 선택하면 지표면 상의 잘못된 위치에 사용자 레이어를 배치할 것이기 때문이죠! 첨부된 메타데이터가 레이어의 정확한 좌표계를 설명할 때도 있고, 정확한 좌표계를 결정하기 위해 데이터의 원저자와 연락해야 할 경우도 있습니다.
   - *Use project CRS*: 프로젝트 좌표계를 적용합니다.
   - *Use default layer CRS*: 위에 있는 콤보박스에 설정된 *Default CRS for layers* 대로 기본 레이어 좌표계를 사용합니다.

> **팁**
>
> 좌표계를 가지고 있지 않거나 잘못된 좌표계를 가지고 있는 레이어 여러 개에 동일한 좌표계를 한번에 할당하려면:
>
> 1. *Layers* 패널에서 레이어를 선택합니다.
> 2. `Ctrl+Shift+C` 키를 동시에 입력합니다. 선택된 레이어 가운데 하나를 오른쪽 클릭하거나 *Layer* → *Set CRS of layer(s)* 메뉴를 선택해도 됩니다.
> 3. 사용할 올바른 좌표계를 찾아 선택합니다.
> 4. *OK* 를 클릭하십시오. 레이어 속성 대화창의 *Source* 탭에서 정확히 설정됐는지 확인할 수 있습니다.
>
> 이 설정에서 좌표계를 변경하는 것은 기저 데이터소스를 전혀 변경시키지 않는다는 사실을 기억하십시오. 그보다는 QGIS 가 현재 QGIS 프로젝트에서 레이어가 가지고 있는 원시 (raw) 좌표를 해석하는 방법을 변경할 뿐입니다.

## 6.3 프로젝트 좌표계

QGIS 의 모든 프로젝트도 좌표계와 관련돼 있습니다. 프로젝트 좌표계는 사용자의 QGIS 맵 캔버스 내에서 데이터의 기저 원시 좌표를 렌더링된 평면 맵으로 어떻게 투영할지 결정합니다.

QGIS 는 래스터와 벡터 데이터의"실시간 (on-the-fly)"좌표계 변환을 지원합니다. 즉 사용자 프로젝트에 있는 어떤 맵 레이어의 기저 좌표계와도 상관 없이, 사용자 프로젝트 용으로 정의된 일반 좌표계로 자동적으로 변환될 것이라는 뜻입니다. 겉으로 드러나지는 않지만, QGIS 는 사용자 프로젝트가 담고 있는 모든 레이어를 프로젝트의 좌표계로 그대로 재투영해서, 레이어들이 각각 서로 정확한 위치로 렌더링될 것입니다!

사용자의 QGIS 프로젝트 용 좌표계를 적절히 선택하는 일은 중요합니다. 적절하지 않은 좌표계를 선택하면 사용자 맵이 일그러져 보일 수도 있고, 실재 세계의 상대적 크기와 피처의 위치를 제대로 반영하지 못할 수도 있습니다. 상대적으로 작은 지리적 면적을 작업한다면, 특정 국가 또는 행정구역 내에서 사용되는 몇몇 표준 좌표계가 있는 것이 보통입니다. 사용자가 맵핑하고 있는 지역에서 어떤 좌표계가 적절한지 또는 표준인지 조사하는 일이 중요합니다. 그 다음 사용자의 QGIS 프로젝트가 그런 표준을 따르도록 하십시오.

QGIS 는 기본적으로 새 프로젝트를 시작할 때 전체 수준 기본 투영체를 사용합니다. 전체 수준 기본 좌표계는 ("WGS 84"로도 알려진) `EPSG:4326` 으로, 지구 위도/경도 기반 좌표계입니다. 이 기본 좌표계는 *Settings* → *Options*…메뉴를 클릭하면 열리는 대화창의 *CRS* 탭에 있는 *CRS for New Projects* 설정을 통해 변경할 수 있습니다. (그림 6.1 참조) 프로젝트 좌표계를 새 프로젝트에 처음으로 불러온 레이어의 좌표계에 자동적으로 일치시키도록 설정할 수도 있고, 다른 방법으로는 새로 생성할 프로젝트에 사용할 다른 기본 좌표계를 선택할 수도 있습니다. 이렇게 선택한 좌표계는 이후 새로 시작되는 QGIS 세션이 사용하도록 저장될 것입니다.

*Project* → *Properties*… *CRS* 탭을 통해서도 대화창의 프로젝트 좌표계를 설정할 수 있습니다. 프로젝트 좌표계는 QGIS 상태 바의 우하단에도 표시될 것입니다.

다음과 같은 옵션을 사용할 수 있습니다:

- ☐ *No CRS (or unknown/non-Earth projection)*: 이 설정을 체크하면 QGIS 내에서 모든 투영체 처리를 비활성화할 것입니다. 즉 모든 레이어 및 맵 좌표를 지표면 상의 위치와는 아무 상관 없는 단순 2 차원 데카르트 평면 좌표로 취급할 것입니다. 레이어 좌표계를 (원시 좌표를 기반으로) 추측하거나, 롤 플레잉 게임, 건물 맵핑, 또는 미소 (microscopic) 사물 등과 같이 지구와 관련이 없는 목적으로 QGIS 를 사용하는 데 이 옵션을 사용할 수 있습니다. 이런 경우에는:
  - 레이어를 렌더링하는 동안 어떤 재투영도 이루어지지 않습니다: 피처를 고유의 원시 좌표를 이용해서 그릴 뿐입니다.
  - 타원체를 무시하고 `None/Planimetric` 으로 강제합니다.

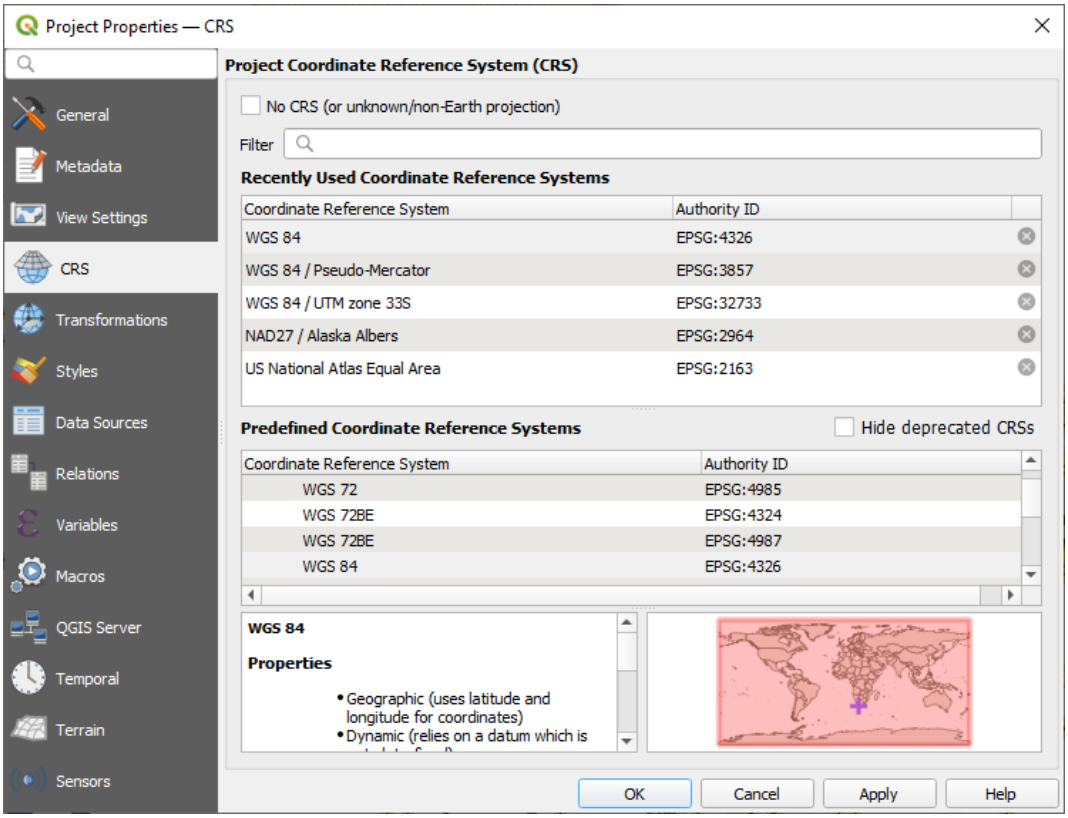



그림 6.2: 프로젝트 속성 대화창

  - 거리 및 면적 단위와 좌표 표시를 무시하고"unknown units"로 강제합니다. 모든 측정은 알 수 없는 맵 단위로 이루어지며, 어떤 변환도 할 수 없습니다.

- 또는 기존 좌표계에서 지리 *(geographic)*, 투영 *(projected)* 또는 사용자 정의 *(user-defined)* 좌표계 가운데 하나를 선택할 수도 있습니다. 사용자가 적절한 좌표계를 선택하는 것을 돕기 위해 지표면 상의 좌표계 범위의 미리보기를 표시합니다. 프로젝트에 추가한 레이어를, 원래 좌표계와 상관없이 중첩시키기 위해, 선택한 좌표계로 실시간 (on-the-fly) 재투영합니다. 단위 및 타원체 설정을 사용할 수 있고, 의미가 통하며, 사용자가 설정에 따라 계산을 수행할 수 있습니다.

If you change your project's CRS and want the units (for distance and area) to match the units defined by the CRS, make sure you have selected *Map units* in the relevant settings, in the *General* tab of the *Project* → *Properties*··· dialog.

**팁**

**레이어로부터 프로젝트 좌표계를 설정하기**

레이어 좌표계를 사용해서 프로젝트에 좌표계를 할당할 수 있습니다:

1. *Layers* 패널에서 선택하고자 하는 좌표계를 오른쪽 클릭하십시오.
2. *Set project CRS from Layer* 를 선택하십시오.

프로젝트 좌표계를 레이어 좌표계로 재정의합니다. 그에 맞춰 맵 캔버스 범위, 좌표 표시를 업데이트하고 프로젝트의 모든 레이어를 새 프로젝트 좌표계로 실시간 변환합니다.

## 6.4 좌표계 선택기

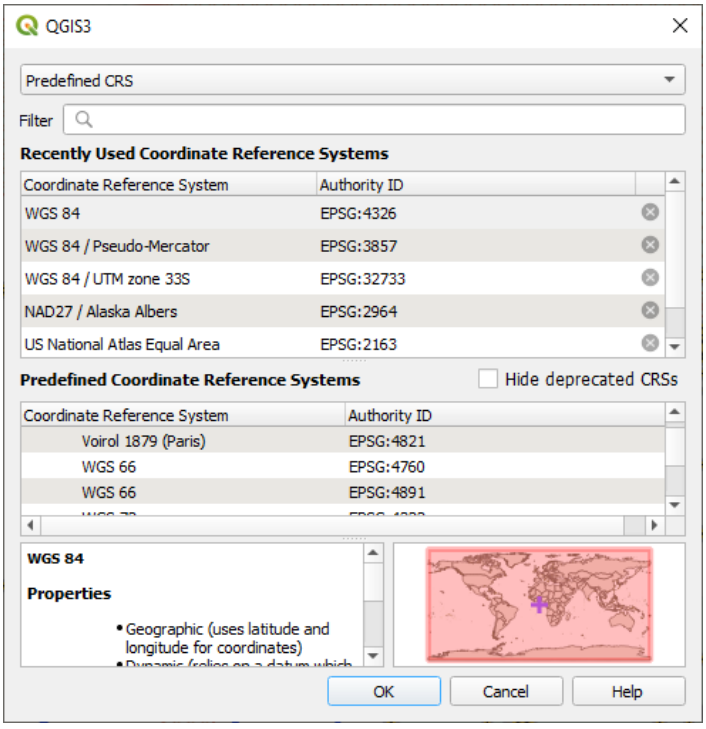



그림 6.3: 좌표계 선택기

이 대화창을 이용해 투영체 데이터베이스에서 지원되는 좌표계를 프로젝트 또는 레이어에 할당할 수 있습니다. 대화창에 있는 항목은 다음과 같습니다:

- **Filter**: EPSG 코드, 식별자, 또는 좌표계 이름을 알고 있을 경우, 검색 기능을 이용해서 좌표계를 찾을 수 있습니다. EPSG 코드, 식별자, 또는 이름을 입력해보십시오.
- **Recently used coordinate reference systems**: GIS 작업 시 사용자가 흔히 이용하는 특정 좌표계가 있을 경우, 이 목록에 해당 좌표계들이 표시될 겁니다. 항목들 가운데 하나를 클릭해서 해당 좌표계를 선택해보십시오.
- **Coordinate reference systems of the world**: QGIS 가 지원하는 모든 좌표계들의 목록을 보여줍니다. 지리 좌표계, 투영 좌표계 및 사용자 지정 좌표계가 들어 있습니다. 좌표계를 지정하려면, 적절한 노드를 확장해서 좌표계를 선택하십시오. 현재 사용중인 좌표계가 미리 선택돼 있습니다.
- **PROJ text**: PROJ 투영 엔진이 사용하는 좌표계 문자열입니다. 이 텍스트는 읽기 전용으로, 정보 제공 용도로 표시됩니다.

목록에서 최근 사용한 좌표계를 제거할 수 있는 방법은 여럿 있습니다:

- 오른쪽에 있는 Remove from recently used CRS 버튼을 누르십시오.
- 제거하려는 좌표계를 선택한 다음 `Del` 키를 누르십시오.
- 사용한 좌표계 하나를 오른쪽 클릭한 다음 *Remove selected CRS from recently used CRS* 메뉴를 선택하거나, 또는 사용한 좌표계들을 모두 제거하려면 *Clear all recently used CRS* 메뉴를 선택하십시오.

좌표계 선택기는 선택한 좌표계를 무결하게 사용할 수 있는 지리적 지역을 개략적으로 보여주는 미리보기도 표시합니다. 많은 좌표계는 작은 지리적 지역에서만 사용하도록 설계되어 있어서, 해당 좌표계가 사용되도록 설계된 지역에서만 사용해야 합니다. 미리보기 맵은 목록에서 좌표계를 선택할 때마다 사용할 수 있는 지역을 추정해서 표시합니다. 또한, 이 미리보기 맵은 현재 주 맵 캔버스의 범위도 보여줍니다.

## 6.5 사용자 정의 좌표계

QGIS 에서 사용자가 필요로 하는 좌표계를 찾을 수 없는 경우, 사용자 지정 좌표계를 정의할 수 있습니다. 좌표계를 정의하려면, *Settings* 메뉴에서 *Custom CRS*…를 선택하십시오. 사용자 지정 좌표계는 QGIS 사용자 데이터베이스에 저장됩니다. 이 데이터베이스는 사용자 지정 좌표계 외에도 사용자 공간 북마크 및 다른 사용자 지정 데이터도 저장하고 있습니다.

QGIS 에서 사용자 지정 좌표계를 정의하려면 PROJ 투영 라이브러리를 잘 이해하고 있어야 합니다. 먼저, 제럴드 I. 이븐던 (Gerald I. Evenden) 이 1990 년 작성한 미국지리조사국 오픈 파일 보고서 90-284 "UNIX 환경의 지도 제작 투영 과정 - 사용자 설명서"를 참조하십시오. (https://pubs.usgs.gov/of/1990/of90-284/ofr90-284.pdf 에서 다운로드할 수 있습니다.)

이 문서는 `proj` 및 관련 명령 줄 유틸리티의 사용법을 설명하고 있습니다. 이 문서가 설명하는 `proj` 에 쓰이는 지도 제작 관련 파라미터는 QGIS 에서 쓰이는 파라미터와 동일합니다.

*Custom Coordinate Reference System Definition* 대화창에서 사용자 좌표계를 정의하려면 파라미터 2 개만 입력하면 됩니다.

1. 좌표계를 설명하는 이름
2. PROJ 또는 WKT 문법으로 된 지도 제작 파라미터

새 좌표계를 생성하려면:

1. Add new CRS 아이콘을 클릭합니다.
2. 좌표계를 설명하는 이름을 입력합니다.
3. 형식을 선택합니다: *Proj String* 또는 *WKT* 가운데 선택하십시오.

4. 좌표계의 *Parameters* 를 추가합니다.

> **참고**
>
> **좌표계 정의를 WKT 형식으로 저장할 것을 추천합니다**
>
> `Proj String` 및 `WKT` 형식 둘 다 지원하긴 하지만, 투영체 정의를 WKT 형식으로 저장할 것을 강력하게 추천하는 바입니다. 따라서 사용할 수 있는 정의가 PROJ 형식으로 되어 있다면, 해당 형식을 선택하고 파라미터를 입력한 다음 WKT 형식으로 바꾸십시오. QGIS 가 해당 정의를 WKT 형식으로 변환할 것입니다. 이후 사용자가 저장하면 됩니다.

5. 사용자가 정의한 좌표계가 무결한 투영체 정의인지 테스트해보려면 *Validate* 를 클릭하십시오.

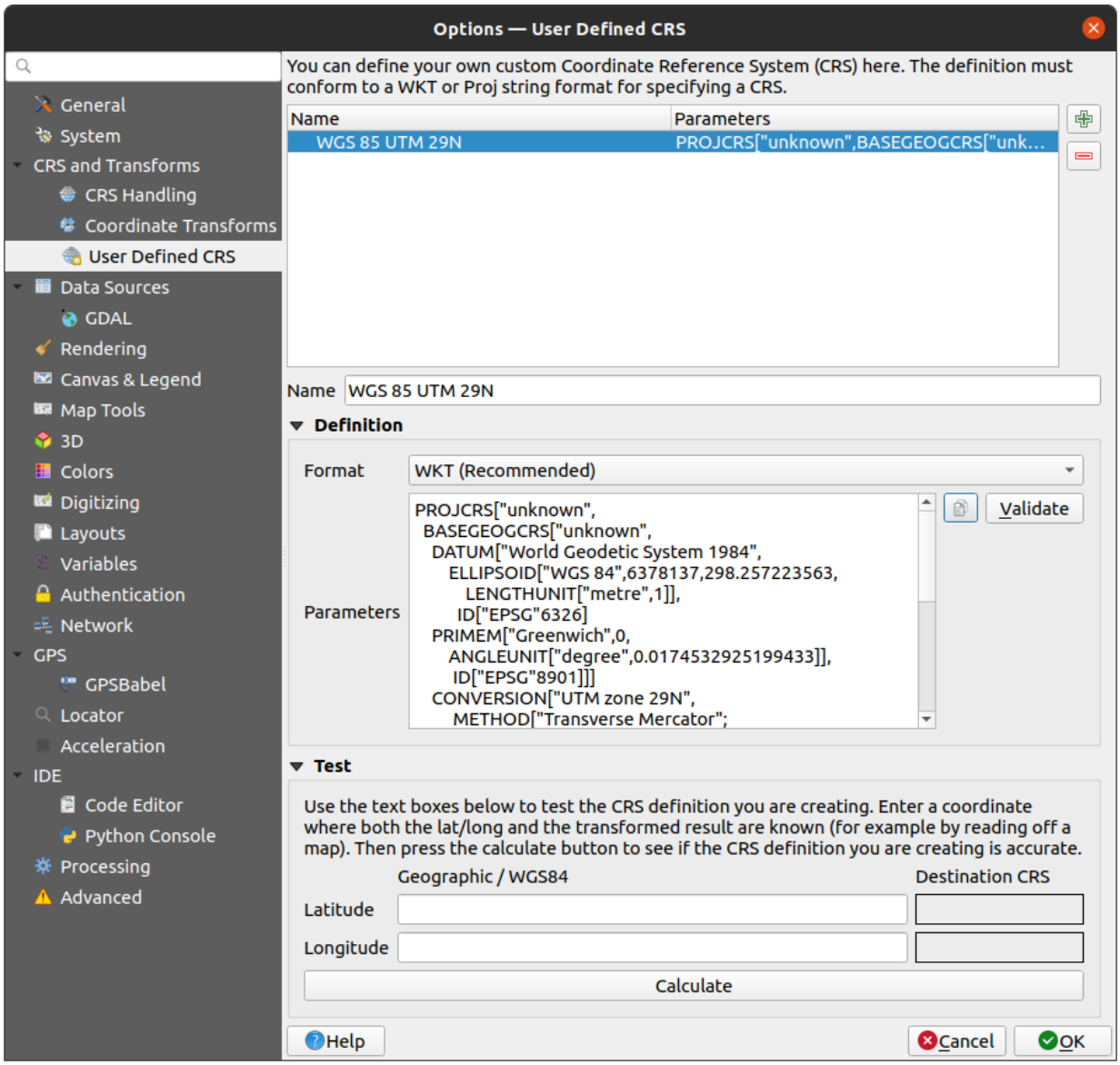



그림 6.4: 사용자 지정 좌표계 대화창

사용자 좌표계가 제대로 된 결과를 내놓는지 확인하기 위해 사용자 좌표계 파라미터를 테스트해볼 수 있습니다.

*North* 와 *East* 란에 사용자가 위치를 알고 있는 WGS 84 위도와 경도 값을 각각 입력하십시오. *Calculate* 를 클릭한 다음, 그 결과를 사용자 좌표계에서 위치를 알고 있는 값과 비교해보십시오.

### 6.5.1 QGIS 에서 NTv2 변환 통합

QGIS 에서 NTv2 변환 파일을 통합하려면 한 가지 단계를 더 거쳐야 합니다:

1. QGIS 가 사용하는 `CRS/Proj` 폴더에 (예를 들어 윈도우 사용자의 경우 `C:\OSGeo4W64\share\proj` 폴더에) NTv2 파일 (`.gsb`) 을 넣으십시오.
2. *Settings* → *Custom Projections*…대화창의 *Custom Coordinate Reference System Definition* 부분의 *Parameters* 란에 있는 PROJ 정의에 **nadgrids** (`+nadgrids=nameofthefile.gsb`) 를 추가하십시오.



그림 6.5: NTv2 변환 설정하기

## 6.6 기준점 (datum) 변환

QGIS 는 기본적으로'실시간 (on-the-fly)'좌표계 변환을 활성화시켜 놓습니다. 즉 다른 좌표계를 가진 레이어를 사용할 때마다 QGIS 는 해당 레이어를 그대로 프로젝트 좌표계로 재투영한다는 뜻입니다. 일부 좌표계의 경우, 여러 가지 방법으로 프로젝트 좌표계로 재투영할 수 있습니다!

QGIS 는 기본적으로 사용할 수 있는 변환 방법 가운데 가장 정확한 방법을 사용하려 할 것입니다. 하지만, 그럴 수 없는 경우도 있습니다. 예를 들어 변환할 때마다 추가적인 지원 파일이 필요한 경우가 있겠습니다. 더 정확한 변환 방법이 있지만 현재 사용할 수는 없을 때마다, QGIS 는 사용자에게 더 정확한 변환 방법과 사용자 시스템에서 해당 변환을 활성화시키는 방법을 알려주는 경고 메시지를 표시할 것입니다. 일반적으로 이렇게 변환 방법을 활성화시키려면 외부 변환 지원 파일 패키지를 다운로드해서 QGIS 의사용자 프로파일 폴더 아래에 있는 `proj` 폴더에 압축 해제해야 합니다.

사용자가 원한다면 QGIS 가 두 좌표계 사이에서 변환할 수 있는 변환 방법이 두 개 이상일 때마다 알려주고, 사용자 데이터를 사용하는 데 어떤 변환이 가장 적절한지에 대한 정보를 제공해서 올바르게 선택할 수 있도록 도와주게 할 수도 있습니다.

*Default datum transformations* 그룹 아래에 있는 *Settings* → *Options* → *Transformations* 탭 메뉴에서 이렇게 사용자 지정할 수 있습니다.

- *Ask for datum transformation if several are available* 사용: 입력/산출 (source/destination) 좌표계 조합에 적절한 기준점 변환 방법이 하나 이상 있을 경우, 사용자에게 프로젝트에서 어떤 기준점 변환을 사용할지 물어보는 대화창이 자동적으로 열릴 것입니다. 이 대화창에서 변환 방법을 선택했을 때 *Make default* 체크박스를 체크했다면, 선택 사항을 기억해서 새로 생성하는 모든 QGIS 프로젝트에 자동적으로 적용할 것입니다.
- 또는 프로젝트에 레이어를 불러오거나 레이어를 재투영할 때 기본값으로 사용할 적절한 기준점 변환 방법 목록을 정의할 수도 있습니다.

  버튼을 클릭해서 *Select Datum Transformations* 대화창을 연 다음:

  1. 드롭다운 메뉴 또는 Select CRS 위젯을 사용해서 레이어의 *Source CRS* 를 선택합니다.
  2. 동일한 방법으로 *Destination CRS* 도 선택합니다.
  3. 입력/산출할 수 있는 변환 방법 목록이 테이블에 표시될 것입니다. 행을 클릭하면 적용된 설정에 대한 상세 정보 및 그에 대응하는 변환의 정확도 및 사용 범위 (area of use) 를 보여줍니다.

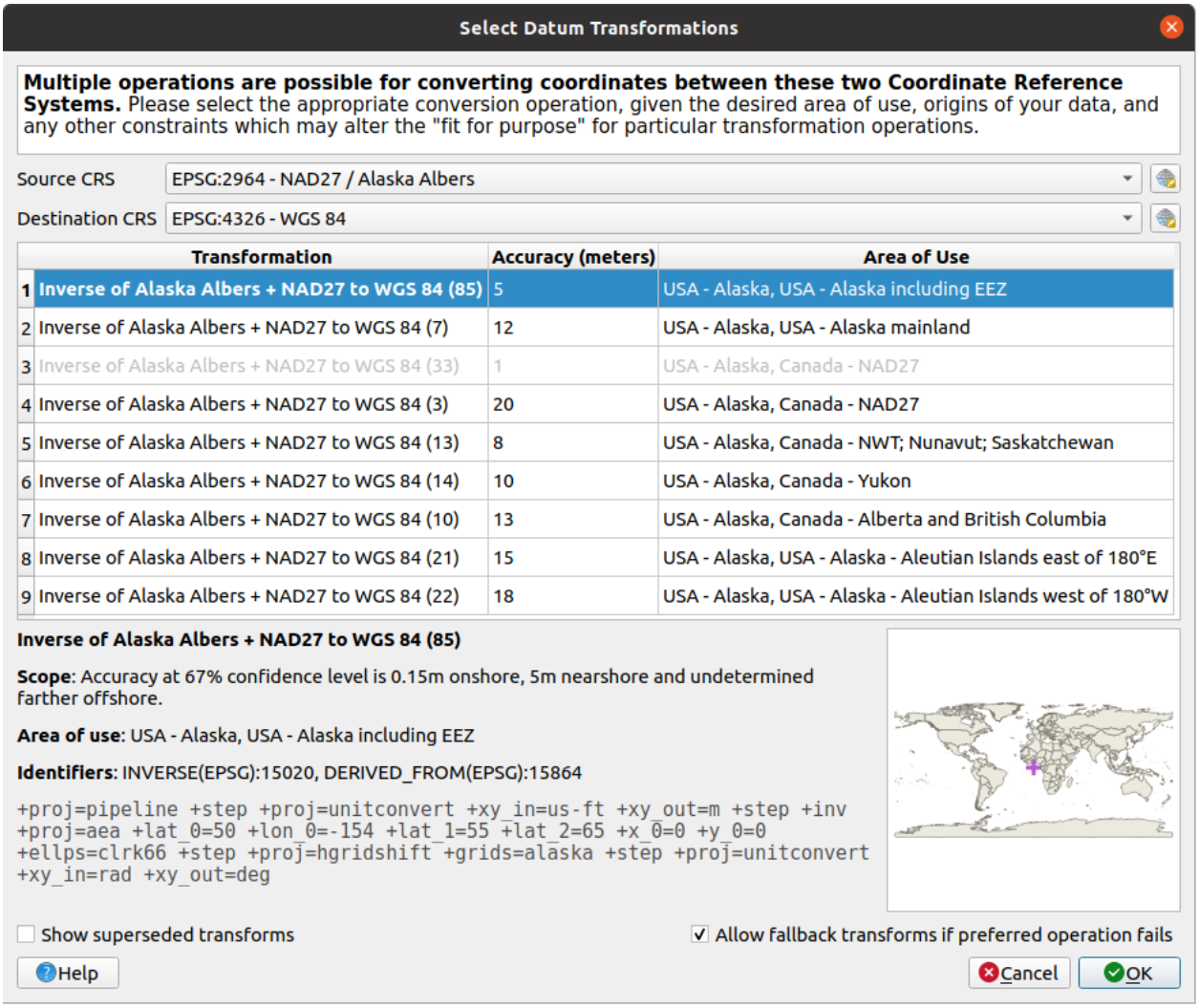



그림 6.6: 선호하는 기본 기준점 변환을 선택하기

사용자 시스템 상에서 변환을 사용할 수 없을 수도 있습니다. 이럴 경우, 해당 변환이 목록에서 사라지지는 않지만 (회색으로 표시되어) 사용자가 필수 변환 지원 패키지를 설치하기 전에는 선택할 수도 없을 것입니다. 일반적으로, 대응하는 그리드를 다운로드해서 설치할 수 있는 버튼이 표시됩니다. 다운로드된 파일은 활성화된사용자 프로파일 디렉터리의 `proj` 폴더에 저장됩니다.

4. 사용자가 선호하는 변환을 찾아 선택하십시오.
5. *Allow fallback transforms if preferred operation fails*: 사용자가 선호하는 변환 방법이 실패할 경우 변환을 되돌릴지 여부를 설정하십시오.
6. :guilabel:`OK`를 클릭하십시오.

   *Default Datum Transformations* 에 있는 테이블에 *Source CRS* 및 *Destination CRS* 는 물론 변환에 적용된 *Operation*, 그리고 *Allow fallback Transforms* 옵션 활성화 여부와 함께 행이 추가됩니다.

QGIS 는 이제부터 자동적으로 선택한 기준점 변환을 이 두 좌표계 사이를 변환하는 데 사용합니다. 사용자가 목록에서 해당 변환을 제거 ( ) 하거나 목록에 있는 항목을 변경 ( ) 하기 전까지는 말이죠.

이 시스템 상에서 생성된 모든 새 QGIS 프로젝트가 *Settings* → *Options* → *Transformations* 탭에서 설정한 기준점 변환을 상속할 것입니다. 또한 특정 프로젝트는 *Project properties* 대화창의 (*Project* → *Properties*…) *CRS* 탭을 통해 지정한 자신만의 특정 변환 방법 모음을 가질 수도 있습니다. 이 설정은 현재 프로젝트에만 적용됩니다.

CHAPTER 7

# 맵 가시화

맵을 생성하는 일은 QGIS 의"상업적 목표"가운데 하나입니다. 맵 뷰는 프로젝트에 불러온 공간 레이어들에 다양한 렌더링 설정들을 적용해서 가시화하는 캔버스입니다. 맵 뷰는*2* 차원 또는*3* 차원 유형 또는단면 선 *(profile line)* 을 따라 그려진 라인 가운데 하나일 수 있고, 서로 다른 축척 또는 범위를 보여줄 수 있으며, 불러온 레이어들의 서로 다른 조합을맵 테마 를 통해 표출할 수도 있습니다.

## 7.1 2 차원 맵 뷰

(**맵 캔버스** 라고도 불리는) 2 차원 맵 뷰는 맵이 표시되는 중심적인 위치입니다. QGIS 는 기본적으로 레이어를 2 차원적으로 표시하며 *Layers* 패널과 강력하게 연동되는 (**메인 맵** 이라고 불리는) 단일 맵 뷰와 함께 열립니다. 이 창은 불러온 레이어들에 사용자가 적용한 렌더링 (심볼, 라벨, 가시성 등등) 을 반영합니다.

### 7.1.1 맵 뷰 탐구하기

사용자가 아무것도 없는 QGIS 프로젝트에 레이어를 추가하면 (데이터 열기 참조) QGIS 는 자동적으로 레이어의 좌표계를 찾습니다. 만약 프로젝트에 기본적으로 설정된 좌표계가 (프로젝트 좌표계 참조) 레이어 좌표계와 다를 경우, 레이어 범위를 프로젝트 좌표계로"실시간 (on-the-fly)"변환한 다음 레이어 범위로 맵 뷰를 확대하거나 축소합니다. 레이어가 이미 프로젝트에 존재하는 경우라면 맵 캔버스 크기 조정은 일어나지 않으므로 현재 맵 캔버스의 범위 안에 들어오는 피처들만 표시될 것입니다.

맵 뷰를 클릭하면, 맵의 서로 다른 영역들로 이동하거나 확대/축소하는 등 맵 뷰와 대화형 작업을 할 수 있을 것입니다. *Navigation Toolbar* 및 *View* 메뉴에서 키보드 또는 마우스 버튼을 통해 사용할 수 있는 여러 단축키를 가진 전용 도구들을 제공합니다.

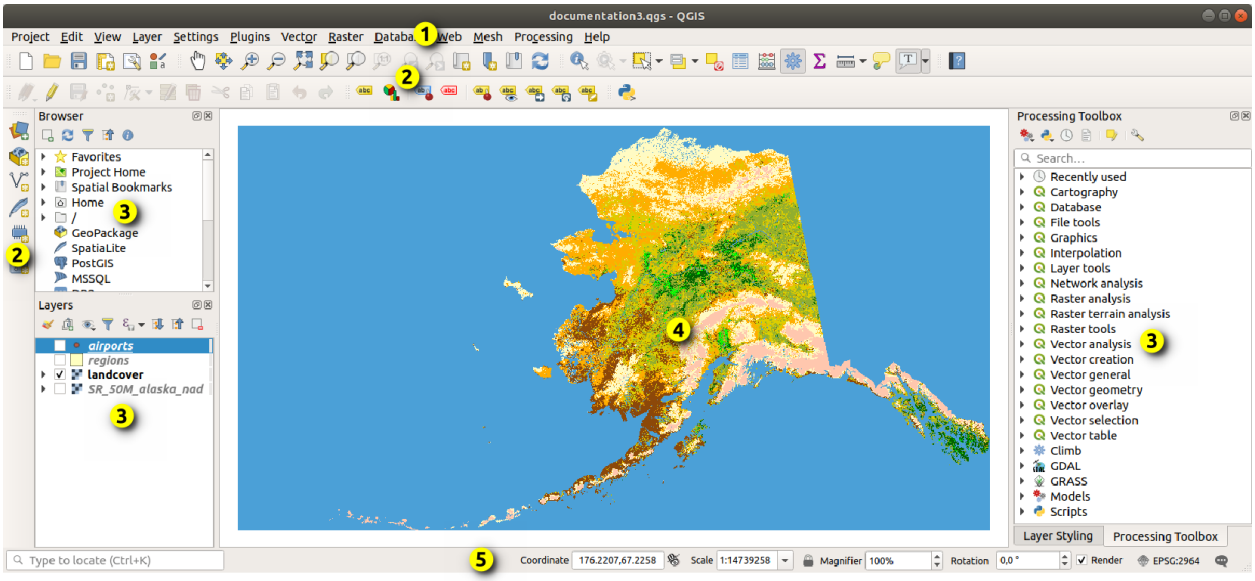



그림 7.1: QGIS GUI

표 7.1: 맵 캔버스 탐색 도구

| 도구 | 사용법 |
|---|---|
| Pan Map | • 왼쪽 클릭 한 번: 맵의 중심이 클릭한 위치로 이동합니다. 축척은 바뀌지 않습니다.<br>• 마우스 왼쪽 버튼을 누른 상태로 맵 캔버스를 드래그할 수 있습니다. |
| Zoom In | • 왼쪽 클릭 한 번: 맵의 중심이 클릭한 위치로 이동합니다. 축척은 2 배가 됩니다.<br>• 마우스 왼쪽 버튼을 누른 채 맵 캔버스 상에 사각형을 드래그하면 해당 영역으로 확대됩니다.<br>• `Alt` 키를 누르고 있으면 Zoom Out 축소 도구로 변환됩니다. |
| Zoom Out | • 왼쪽 클릭 한 번: 맵의 중심이 클릭한 위치로 이동합니다. 축척은 1/2 배가 됩니다.<br>• 마우스 왼쪽 버튼을 누른 채 맵 캔버스 상에 사각형을 드래그하면 해당 영역으로부터 축소됩니다.<br>• `Alt` 키를 누르고 있으면 Zoom In 확대 도구로 변환됩니다. |
| Pan Map to Selection | 맵을 *Layers* 패널에서 선택한 모든 선택 레이어의 선택 피처들로 이동시킵니다. |
| Zoom To Selection | 맵을 *Layers* 패널에서 선택한 모든 선택 레이어의 선택 피처들로 확대/축소시킵니다. 레이어의 컨텍스트 메뉴에서도 사용할 수 있습니다. |
| Zoom To Layer(s) | 맵을 *Layers* 패널에서 선택한 모든 선택 레이어의 범위로 확대/축소시킵니다. 레이어의 컨텍스트 메뉴에서도 사용할 수 있습니다. |
| Zoom Full | 맵을 프로젝트의 모든 레이어들의 범위 또는프로젝트 전체 범위 로 확대/축소시킵니다. |
| Zoom Last | 맵을 액션 이력에 있는 이전 범위로 확대/축소시킵니다. |

다음 페이지에 계속

표 7.1 – 이전 페이지에서 계속

| 도구 | 사용법 |
| --- | --- |
|  Zoom Next | 맵을 액션 이력에 있는 다음 범위로 확대/축소시킵니다. |
|  Zoom to Native Resolution | 맵을 활성화된 래스터 레이어의 픽셀 1 개가 모니터 스크린의 픽셀 1 개와 동일한 크기가 되는 수준으로 확대/축소시킵니다.<br>레이어의 컨텍스트 메뉴에서도 사용할 수 있습니다. |
| 마우스 휠 | • 맵 이동: 마우스 휠을 누른 채 드래그합니다.<br>• 확대/축소: 마우스 휠을 돌리면 맵을 확대하거나 축소할 수 있습니다. `Ctrl` 키를 누른 상태로 마우스 휠을 돌리면 더 세밀하게 확대/축소할 수 있습니다.<br>• 마우스의 이전 또는 다음 버튼을 누르면 맵 캔버스 확대/축소 이력을 오갈 수 있습니다. |
| 키보드 | • 맵 이동: `Space` 키를 누른 채 마우스를 움직입니다. 화살표 키를 통해 맵을 상하좌우로 이동시킬 수 있습니다.<br>• 확대: `PgUp` 키 또는 `Ctrl++` 단축키를 누릅니다.<br>• 축소: `PgDown` 키 또는 `Ctrl+-` 단축키를 누릅니다.<br>• 영역으로 확대/축소: 특정 (식별, 측정 등등의) 맵 도구가 활성화되어 있는 경우 `Shift` 키를 누른 채 맵 상에 사각형을 드래그하면 맵을 해당 영역으로 확대/축소시킵니다. 선택 또는 편집 도구가 활성화되어 있는 상태에서는 사용할 수 없습니다. |

맵을 오른쪽 클릭하면 클릭한 포인트의 좌표를 맵 좌표계, WGS84 좌표계, 또는 사용자 지정 좌표계 단위로 복사  *Copy coordinates* 할 수 있을 것입니다. 그렇게 복사한 정보를 표현식, 스크립트, 텍스트 편집기, 또는 스프레드시트 등에 붙여넣을 수 있습니다.

### 7.1.2 맵 렌더링 제어하기

기본적으로 QGIS 는 맵 캔버스를 업데이트할 때마다 모든 가시 레이어를 렌더링합니다. 다음과 같은 이벤트가 맵 캔버스 업데이트를 촉발합니다:

- 레이어의 가시성 변경
- 가시 레이어의 심볼 수정
- 레이어 추가
- 맵 이동 또는 확대/축소
- QGIS 창 크기 변경

사용자는 QGIS 상에서 다양한 방법으로 렌더링 과정을 제어할 수 있습니다.

- 전체 수준 에서
- 예를 들어축척에 따른 렌더링 을 이용해서 레이어 별로
- 또는 GUI 에 있는 전용 도구로

맵 렌더링을 중단하려면 `Esc` 키를 누르십시오. 맵 캔버스 업데이트를 중단하고 맵 일부만 렌더링된 상태로 남겨둘 겁니다. 하지만 `Esc` 키를 누른 다음 실제로 맵 렌더링이 중단될 때까지 시간이 조금 걸릴 수도 있습니다.

렌더링 작업을 보류하려면, 상태 바 우하단에 있는 *Render* 체크박스를 클릭하십시오. *Render* 체크박스를 해제하면, QGIS 는 앞의맵 렌더링 제어하기 에서 설명한 어떤 일반적인 트리거에도 대응하지 않으며 캔버스를 다시 그리지 않습니다. 사용자가 렌더링을 보류하길 원할 수도 있는 경우는 다음과 같습니다:

- 많은 레이어를 추가하고 렌더링 전에 심볼 작업을 하고자 하는 경우
- 하나 이상의 대용량 레이어를 추가하고 렌더링 전에 축척 의존성을 설정하고자 하는 경우
- 하나 이상의 대용량 레이어를 추가하고 렌더링 전에 특정한 뷰로 확대/축소하고자 하는 경우
- 이런 경우들의 조합

*Render* 체크박스를 체크하면 렌더링을 활성화해서 즉시 맵 캔버스를 업데이트하게 됩니다.

### 7.1.3 맵 캔버스에 대한 시간 기반 제어

QGIS 는 불러온 레이어에 대한 시계열 제어를 처리할 수 있습니다. 예를 들어 시간 변화를 기반으로 맵 캔버스 렌더링을 수정할 수 있습니다. 시계열 제어에 필요한 사항은 다음과 같습니다:

1. 동적 시계열 속성이 설정된 레이어가 필요합니다. QGIS 는 서로 다른 데이터 제공자에 대해 각각의 맞춤 설정을 통해 시계열 제어를 지원합니다. 중요한 것은 레이어에 시간 범위를 설정하는 것으로, 이 경우 해당 레이어에 대해 각각 다음 작업을 할 수 있습니다:
   - 래스터 레이어: 레이어를 표시할지 표시하지 않을지 여부를 제어할 수 있습니다.
   - *WMTS* 레이어: 데이터를 정적 시간 범위를 기반으로 렌더링해야 할지 또는 동적 시계열 범위를 따라 렌더링해야 할지를 제어할 수 있습니다.
   - 벡터 레이어: 피처 속성과 관련된 시간값을 기반으로 피처를 필터링할 수 있습니다.
   - 메시 레이어: 활성화된 데이터셋 그룹의 값들을 동적으로 표시할 수 있습니다.

   어떤 레이어에 대해 동적 시계열 옵션들이 활성화된 경우, *Layers* 패널의 레이어 옆에 아이콘이 표시되어 해당 레이어를 시계열 제어할 수 있다는 사실을 나타냅니다. 이 아이콘을 클릭하면 시계열 설정을 업데이트할 수 있습니다.
2. 시계열 제어기 패널 을 이용해서 맵 캔버스의 시계열 탐색을 활성화시킬 수 있습니다. 다음 방법으로 이 패널을 활성화시켜보십시오:
   - *Map Navigation* 툴바에 있는 Temporal controller panel 아이콘을 클릭
   - *View* → *Panels* → *Temporal controller panel* 메뉴 항목을 선택
   - or from the *View* → *Data Filtering* → *Temporal controller panel* menu

#### 시계열 제어기 패널

*Temporal controller* 패널에서 다음 모드들을 선택할 수 있습니다:

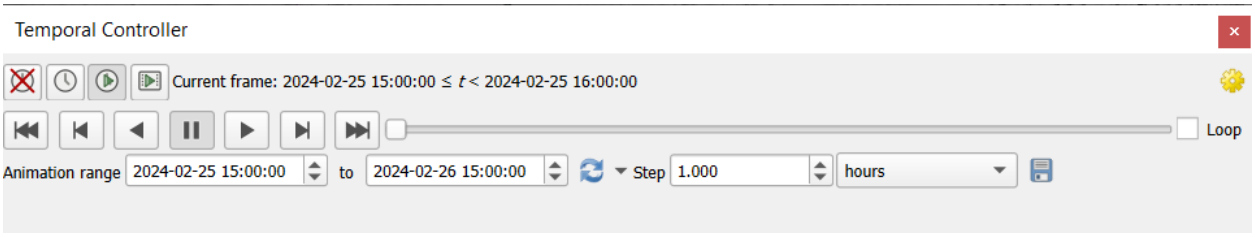



그림 7.2: 탐색 모드의 시계열 제어기 패널

-  Turn off temporal navigation: 모든 시계열 설정을 비활성화하고 가시 레이어들을 보통 때와 같이 렌더링합니다.
-  Fixed range temporal navigation: 시간 범위를 정하고, 레이어 시계열 범위가 이 범위와 겹치는 레이어 (또는 피처) 만 맵 상에 표시합니다.
-  Animated temporal navigation: 단계별로 분할된 시간 범위를 정하고, 레이어 시계열 범위가 분할된 각 시계열 프레임과 겹치는 레이어 (또는 피처) 만 맵 상에 표시합니다.
-  Animated movie: a total number of frames is set for the animation, with the map settings advancing at each step, displaying the progression frame by frame without any time-based filtering of data.
-  Settings: 시계열 애니메이션의 일반 설정들을 제어합니다.
  - *Frames rate*: 초 당 표시되는 단계들의 개수
  -  *Cumulative range*: 모든 애니메이션 프레임의 시작 날짜 & 시간은 동일하지만 끝나는 날짜 및 시간은 다를 것입니다. 사용자 데이터의 전반에 걸쳐'움직이는 시간 창'을 표시하는 대신 사용자의 시계열 가시화에 데이터를 누적시키고자 하는 경우 유용한 옵션입니다.

### 시계열 탐색 애니메이션 작업

애니메이션은 어떤 시간 범위 안의 특정 시간들에서 연속적으로 변화하는 가시 레이어들의 집합을 기반으로 합니다. 시계열 애니메이션을 생성하는 방법은 다음과 같습니다:

1.  Animated temporal navigation 도구를 켜면 애니메이션 플레이어 위젯이 나타납니다.
2. 사용할 *Time range* 시간 범위를 입력하십시오.  버튼을 이용하면 시간 범위를 다음과 같이 정의할 수 있습니다:
   - *Set to full range*: 활성화된 모든 레이어들의 전체 시간 범위로 정의합니다.
   - *Set to preset project range*: 프로젝트 속성 에 정의된 시간 범위로 정의합니다.
   - *Set to single layer's range*: 시간이 활성화된 레이어의 시계열 범위로부터 추출합니다.
3. *Step* 에 시간 범위를 분할할 시간 단계를 입력하십시오. 초 에서 세기 에 걸친 다양한 단위를 지원합니다. 소스 타임스탬프 옵션도 시간 단계로 사용할 수 있습니다. 이 옵션을 선택하면 프로젝트에 있는 레이어들로부터 사용할 수 있는 모든 시간 범위들 사이의 단계를 시계열 탐색하게 됩니다. 이 옵션은 프로젝트가 담고 있는 레이어에서 사용 가능한 시간들이 불연속적인 경우, 예를 들어 불규칙적인 날짜에 생성된 사용 가능 이미지들을 제공하는 WMS-T 서비스 등에 유용합니다. 이 옵션을 선택하면 다음으로 사용할 수 있는 이미지가 표시되는 시간 범위 사이의 단계만 사용할 수 있습니다.
4. 애니메이션 미리보기를 하려면  버튼을 클릭하십시오. QGIS 가 설정 시간에서 렌더링된 레이어를 사용한 신 (scene) 을 생성할 것입니다. 레이어들이 겹치는 개별 시간 프레임의 존재 여부에 따라 레이어들을 표시합니다.

   시간 슬라이드 바를 움직여서 애니메이션을 미리보기할 수도 있습니다.  *Loop* 체크박스를 체크하면 애니메이션을 반복 실행할 것입니다. 이때  버튼을 누르면 애니메이션을 정지합니다. 완전한 비디오 플레이어 버튼 집합을 사용할 수 있습니다.

   맵 캔버스 위에 커서를 둔 채 마우스 휠을 (지원하는 경우) 수평으로 돌리면 시계열 탐색 슬라이드 바를 앞뒤로 탐색 또는"문지를 (scrub)"수 있습니다.
5. 애니메이션 신을 표현하는 일련의 이미지들을 생성하려면  Export animation 버튼을 클릭하십시오. 향후 영상 편집 소프트웨어에서 이 이미지들을 결합할 수 있습니다:
   - *Template*: 파일명 템플릿입니다. `####` 은 프레임 시퀀스 번호로 대체됩니다.

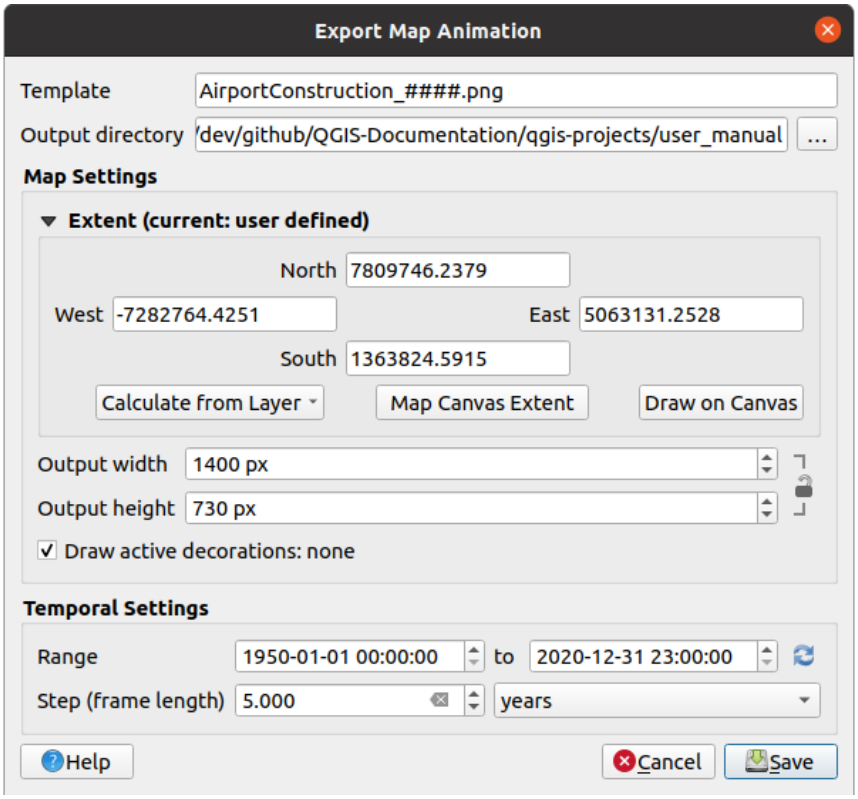



그림 7.3: 맵 캔버스 애니메이션 신을 이미지로 내보내기

- *Output directory*: 산출 이미지를 저장할 디렉터리입니다.
- *Map settings* 부분에서 다음 작업을 할 수 있습니다:
  - 사용할공간 범위 를 재정의
  - 이미지 *Resolution* 해상도를 제어 (*Output width* 산출물 너비와 *Output height* 산출물 높이를 설정)
  - *Draw active decorations*: 활성화되어 있는장식 을 산출 이미지에 유지해야 할지 여부를 선택
- *Temporal settings* 부분에서 다음을 재정의할 수 있습니다:
  - *Range*: 애니메이션의 시간 범위
  - *Step (frame length)*: 시간 단계를 사용자가 선택한 단위로 설정

### 7.1.4 The Elevation Controller panel

*Elevation Controller* allows you to handle elevation z-range data in 2D maps. This option currently supports point cloud layers and raster DEMs. Use the panel to visualize and interact with elevation data in the map canvas.

Activate the *Elevation Controller* panel in the map canvas by clicking on the *Data Filtering* in the *View* menu. The *Elevation Controller* appears as a range slider on the left side of the map canvas. At the top of the slider, there is a Settings menu where you can:

- Set the *Fixed Range Size* value to lock the elevation range to a specific size. This setting is saved within the project and retained across sessions, ensuring consistency for your elevation filter. Use the *Invert Direction* option to

reverse the elevation filter slider, so that it moves from high values at the bottom to low values at the top.

- *Set Elevation Range* to open the project elevation tab, where you can set dedicated lower and upper bounds for the elevation range
- *Disable Elevation Filter* to remove the elevation filter from the map canvas

### 7.1.5 맵 상에서의 범위 북마크 작업

공간 북마크를 통해 지리적 위치에"책갈피"를 끼워 나중에 다시 해당 위치로 돌아올 수 있습니다. 기본적으로 북마크는 사용자 프로파일에 (*User Bookmarks* 로) 저장되는데, 사용자가 연 어떤 프로젝트에서도 사용할 수 있다는 뜻입니다. 북마크를 단일 프로젝트 용으로 (*Project Bookmarks* 라는 이름으로) 저장할 수도 있습니다. 프로젝트 북마크는 프로젝트 파일 안에 저장되는데, 다른 사용자와 공유해야 하는 프로젝트인 경우 유용할 것입니다.

#### 북마크 생성하기

북마크를 생성하려면:

1. 관심 지역으로 확대/축소 또는 이동하십시오.
2. *View* → *New Spatial Bookmark*…메뉴 옵션을 선택하고, `Ctrl+B` 키 조합을 누르거나 *Browser* 패널에 있는 *Spatial Bookmarks* 항목을 오른쪽 클릭한 다음 *New Spatial Bookmark* 를 선택하십시오. *Bookmark Editor* 대화창이 열립니다.

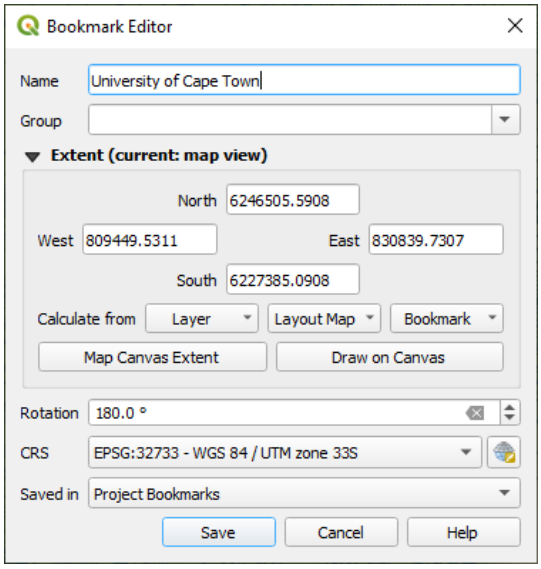



그림 7.4: 북마크 편집기 대화창

3. 북마크를 잘 설명하는 이름을 입력하십시오.
4. 관련 북마크를 저장할 그룹명을 입력하거나 선택하십시오.
5. 범위 선택기 위젯을 이용해서 저장하고자 하는 영역의 범위를 선택하십시오.
6. 맵의 *Rotation* 회전 각도를 변경하십시오.
7. 선택한 범위에 사용할 *CRS* 좌표계를 지정하십시오.
8. *Saved in* 옵션에서 북마크를 *User Bookmarks* 로 저장할지 또는 *Project Bookmarks* 로 저장할지 선택하십시오. (기본적으로 이 드롭다운 목록은 *User Bookmarks* 로 설정되어 있습니다.)

9. *Save* 를 클릭하면 북마크가 목록에 추가됩니다.

북마크 여러 개를 동일한 이름으로 만들 수 있다는 사실을 기억하십시오.

### 북마크 관련 작업

To use and manage bookmarks, you can either use the *Spatial Bookmarks* panel or *Browser*:

- Select *View* → *Show Spatial Bookmark Manager* or press `Ctrl+7` to open the *Spatial Bookmark Manager* panel.
- Select *View* → *Show Spatial Bookmarks* or press `Ctrl+Shift+B` to show the *Spatial Bookmarks* entry in the *Browser* panel.

다음과 같은 작업들을 수행할 수 있습니다:

표 7.2: 북마크 작업 관리하기

| 작업 | 공간 북마크 관리자 | 탐색기 |
|---|---|---|
| **북마크로 확대/축소** | 북마크를 더블 클릭하거나, 북마크를 선택한 다음 *Zoom to bookmark* 버튼을 누르십시오. | 북마크를 더블 클릭하거나, 북마크를 맵 캔버스로 드래그 & 드롭하거나, 북마크를 오른쪽 클릭한 다음 *Zoom to Bookmark* 를 선택하십시오. |
| **북마크 삭제** | 북마크를 선택하고 *Delete bookmark* 버튼을 클릭한 다음, 확인을 누르십시오. | 북마크를 오른쪽 클릭하고 *Delete Spatial Bookmark* 를 선택한 다음, 확인을 누르십시오. |
| **북마크를 XML 로 내보내기** | *Import/Export Bookmarks* 버튼을 클릭한 다음 *Export* 를 선택하십시오. 모든 (사용자 또는 프로젝트) 북마크를 `xml` 파일로 저장합니다. | 하나 이상의 폴더 (사용자 또는 프로젝트) 또는 하위 폴더 (그룹) 를 선택한 다음, 오른쪽 클릭하고 *Export Spatial Bookmarks …*를 선택하십시오. 선택한 북마크 부분 집합 (subset) 을 저장합니다. |
| **XML 에서 북마크 가져오기** | *Import/Export Bookmarks* 버튼을 클릭한 다음 *Import* 를 선택하십시오. `xml` 파일에 있는 모든 북마크를 사용자 북마크로 가져옵니다. | *Spatial Bookmarks* 항목이나, 북마크를 가져올 하나 이상의 북마크 폴더 (사용자 또는 프로젝트) 또는 하위 폴더 (그룹) 를 오른쪽 클릭한 다음, *Import Spatial Bookmarks* 를 선택하십시오. *Spatial Bookmarks* 항목을 오른쪽 클릭했을 경우, *User Bookmarks* 로 북마크를 추가합니다. |
| **북마크 편집** | 테이블에 있는 값을 변경해서 북마크를 수정할 수 있습니다. 사용자가 이름, 그룹, 범위 및 프로젝트에 저장됐는지 여부를 편집할 수 있습니다. | 원하는 북마크를 오른쪽 클릭하고 *Edit Spatial Bookmark…*를 선택합니다. *Bookmark Editor* 가 열려, 마치 사용자가 북마크를 처음으로 생성하는 것처럼 해당 북마크의 모든 속성을 재정의할 수 있습니다.<br>폴더 (사용자 또는 프로젝트) 또는 하위 폴더 (그룹) 로 북마크를 드래그 & 드롭할 수도 있습니다. |

*Spatial Bookmarks Manager* 에서 원하는 북마크를 오른쪽 클릭하면 북마크 작업을 관리할 수 있습니다. 또위치 탐색자 에 북마크 이름을 입력하면 북마크로 확대/축소할 수도 있습니다.

### 7.1.6 맵 장식하기

지도 장식은 그리드, 제목 라벨, 저작권 라벨, 이미지, 방위표, 축척 막대 및 조판 범위를 포함하며, 지도 제작 요소를 추가해 맵을'장식'하는 데 쓰입니다.

#### 그리드

*Grid* 를 통해 맵 캔버스에 좌표 격자 및 좌표 주석을 추가할 수 있습니다.

1. *View* → *Decorations* → *Grid*…메뉴 옵션을 선택해서 대화창을 여십시오.

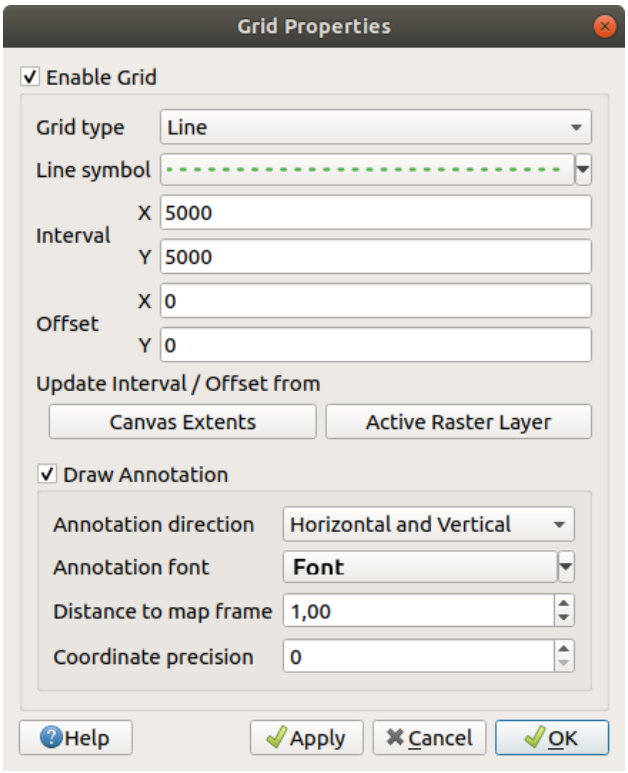



그림 7.5: 그리드 대화창

2. *Enable grid* 체크박스를 체크해서 활성화한 다음 맵 캔버스에 불러온 레이어에 맞춰 그리드 정의를 설정하십시오:
   - *Grid type*: *Line* 또는 *Marker* 가운데 하나로 설정할 수 있습니다.
   - 그리드를 표시하는 데 관련라인 심볼 또는마커 심볼 을 사용합니다.
   - *Interval X* 와 *Interval Y*: 그리드 표시 사이의 X 방향 간격과 Y 방향 간격을 맵 단위로 설정합니다.
   - *Offset X* 와 *Offset Y*: 맵 캔버스의 좌하단에서 그리드 표시까지의 X 방향, Y 방향 거리를 맵 단위로 설정합니다.
   - 다음에 기반해서 간격 및 오프셋 파라미터를 설정할 수 있습니다:
     - *Canvas Extents*: 캔버스 범위의 약 1/5 간격으로 그리드를 생성합니다.
     - *Active Raster Layer*: 가시 래스터 레이어의 해상도를 기반으로 그리드를 생성합니다.

3. *Draw annotations* 옵션을 체크하면 그리드 표시의 좌표를 보이게 하고 다음을 설정할 수 있습니다:
   - *Annotation direction*: 주석의 방향, 즉 그리드 선을 기준으로 라벨을 어떻게 배치할지 다음 가운데 하나를 선택합니다.
     - *Horizontal* 또는 *Vertical*: 모든 라벨을 그리드 선에 대해 수평 또는 수직으로 배치합니다.
     - *Horizontal and Vertical*: 각 라벨을 각 라벨이 참조하는 그리드 표시에 평행하게 배치합니다.
     - *Boundary direction*: 각 라벨을 캔버스 경계 (boundary) 를 따라, 각 라벨이 참조하는 그리드 표시에 수직으로 배치합니다.
   - *Annotation font*: 글꼴 선택기 위젯 을 통해 주석의 글꼴 (텍스트 서식, 버퍼, 음영 등) 을 선택합니다.
   - *Distance to map frame*: 주석과 맵 캔버스 경계 사이의 여백을 설정합니다. 예를 들어 이미지 포맷이나 PDF 로맵 캔버스를 내보내면서 주석이"용지"한계에 걸리지 않도록 하는 데 유용합니다.
   - *Coordinate precision*: 좌표 수치의 정밀도를 설정합니다.
4. 기대한대로 모양이 나오는지 확인하려면 *Apply* 를, 또는 사용자가 만족한다면 *OK* 를 클릭하십시오.

### 제목 라벨

*Title Label* 도구는 사용자 맵에 **제목** 을 장식할 수 있습니다.

제목 라벨 장식을 추가하려면:

1. *View* → *Decorations* → *Title Label*…메뉴 옵션을 선택해서 대화창을 여십시오.

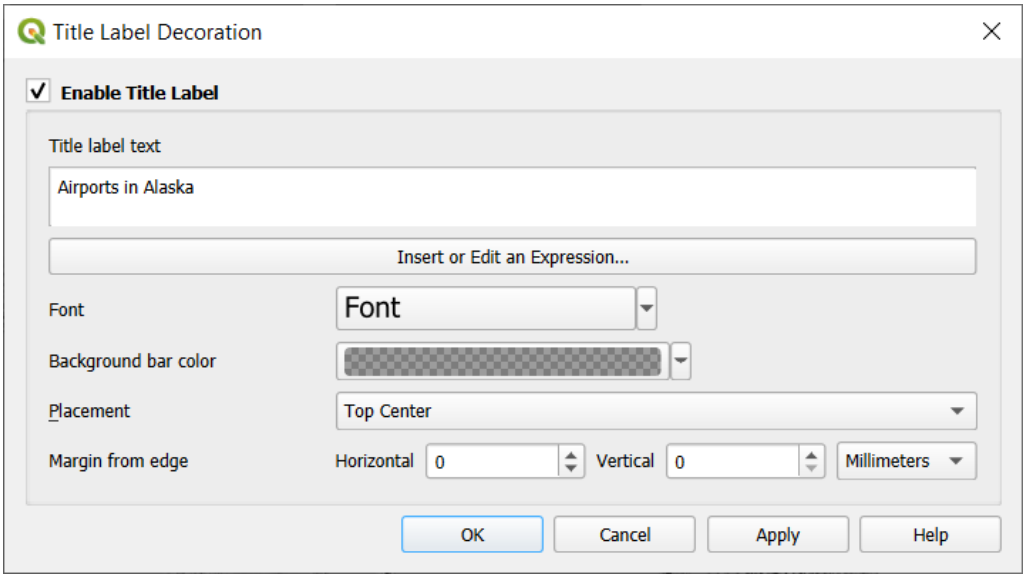



그림 7.6: 제목 장식 대화창

2. *Enable Title Label* 체크박스가 체크돼 있는지 확인하십시오.
3. 맵 상에 배치하려 하는 제목 텍스트를 입력하십시오. *Insert or Edit an Expression*…버튼을 사용하면 동적 텍스트를 만들 수 있습니다.
4. *Font* 에서 QGIS 의텍스트 서식 설정 전체 옵션을 사용할 수 있는글꼴 선택기 위젯 을 통해 라벨 용 글꼴을 선택하십시오. 글꼴 콤보박스 오른쪽에 있는 검은 화살표를 클릭하면 글꼴 색상 및 투명도를 빠르게 설정할 수 있습니다.

5. *Background bar color* 에서 제목의 배경 막대에 적용할색상 을 선택하십시오.
6. *Placement* 에서 캔버스에 라벨을 어떻게 배치할지 선택하십시오. *Top left*, *Top Center* (기본값), *Top Right*, *Bottom left*, *Bottom Center* 및 *Bottom Right* 가운데 하나를 선택할 수 있습니다.
7. *Margin from Edge* 에서 수평 그리고/또는 수직 경계 여백을 설정해서 항목의 배치를 미세 조정하십시오. **밀리미터** 또는 **픽셀** 값으로 설정할 수도 있고, 맵 캔버스의 너비 또는 높이의 **백분율** 값으로 설정할 수도 있습니다.
8. 기대한대로 모양이 나오는지 확인하려면 *Apply* 를, 또는 사용자가 만족한다면 *OK* 를 클릭하십시오.

### 저작권 라벨

*Copyright Label* 을 통해 사용자 맵에 **저작권** 라벨을 장식할 수 있습니다.

이 장식을 추가하려면:

1. *View* → *Decorations* → *Copyright Label*…메뉴 옵션을 선택해서 대화창을 여십시오.

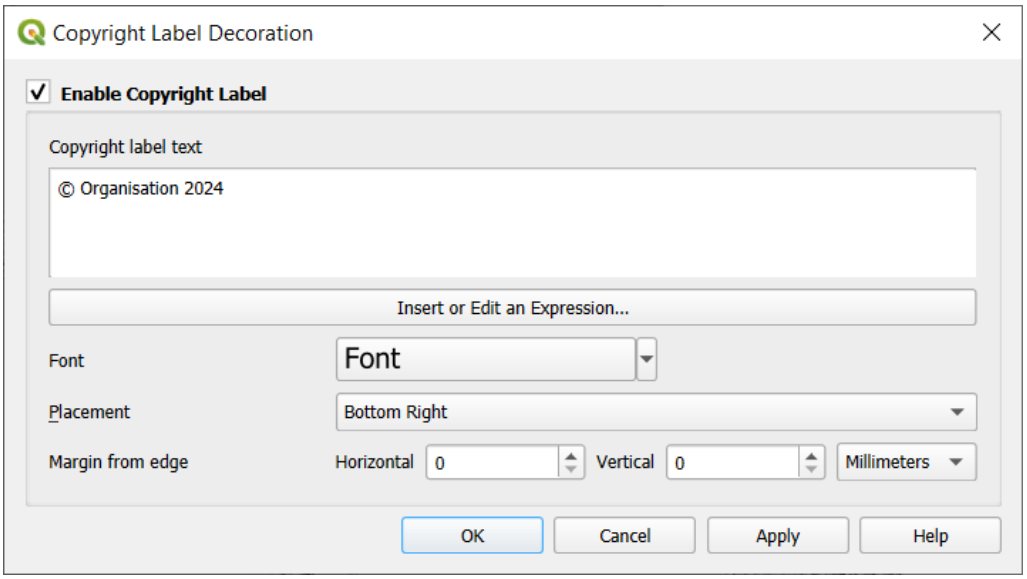



그림 7.7: 저작권 장식 대화창

2. *Enable Copyright Label* 체크박스가 체크돼 있는지 확인하십시오.
3. 맵 상에 배치하려 하는 저작권 텍스트를 입력하십시오. *Insert or Edit an Expression*…버튼을 사용하면 동적 텍스트를 만들 수 있습니다.
4. *Font* 에서 QGIS 의텍스트 서식 설정 전체 옵션을 사용할 수 있는글꼴 선택기 위젯 을 통해 라벨 용 글꼴을 선택하십시오. 글꼴 콤보박스 오른쪽에 있는 검은 화살표를 클릭하면 글꼴 색상 및 투명도를 빠르게 설정할 수 있습니다.
5. *Placement* 에서 캔버스에 라벨을 어떻게 배치할지 선택하십시오. *Top left*, *Top Center*, *Top Right*, *Bottom left*, *Bottom Center* 및 *Bottom Right* (저작권 장식의 기본값) 가운데 하나를 선택할 수 있습니다.
6. *Margin from Edge* 에서 수평 그리고/또는 수직 경계 여백을 설정해서 항목의 배치를 미세 조정하십시오. **밀리미터** 또는 **픽셀** 값으로 설정할 수도 있고, 맵 캔버스의 너비 또는 높이의 **백분율** 값으로 설정할 수도 있습니다.
7. 기대한대로 모양이 나오는지 확인하려면 *Apply* 를, 또는 사용자가 만족한다면 *OK* 를 클릭하십시오.

### 이미지 장식

*Image* 를 통해 맵 캔버스 위에 이미지 (로고, 범례 등등) 를 추가할 수 있습니다.

이미지를 추가하려면:

1. *View* → *Decorations* → *Image*…메뉴 옵션을 선택해서 대화창을 여십시오.

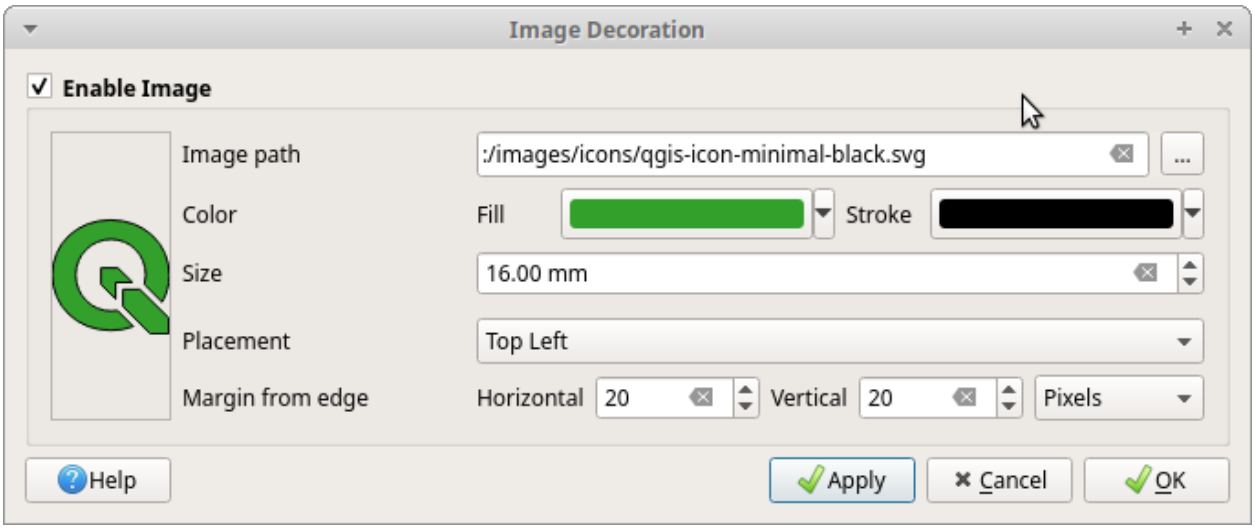



그림 7.8: 이미지 장식 대화창

2. *Enable Image* 체크박스가 체크돼 있는지 확인하십시오.
3. … Browse 버튼을 사용해서 (png 또는 jpg 같은) 비트맵 이미지 또는 SVG 이미지를 선택하십시오.
4. SVG 를 활성화시키는 파라미터를 선택했다면 *Fill* 또는 *Stroke* (외곽선) 색상도 설정할 수 있습니다. 비트맵 이미지의 경우 색상 설정은 비활성화됩니다.
5. *Size* 에서 이미지의 크기를 밀리미터 단위로 설정하십시오. 선택한 이미지의 너비를 설정한 *Size* 로 맞춰 크기를 재조정합니다.
6. *Placement* 콤보박스를 통해 맵 상에 이미지를 어떻게 배치할지 선택하십시오. 기본 위치는 *Top Left* 입니다.
7. *Margin from (Canvas) Edge* 에서 *Horizontal* 및 *Vertical* 경계 여백을 설정하십시오. **밀리미터** 또는 **픽셀** 값으로 설정할 수도 있고, 맵 캔버스의 너비 또는 높이의 **백분율** 값으로 설정할 수도 있습니다.
8. 기대한대로 모양이 나오는지 확인하려면 *Apply* 를, 사용자가 만족한다면 *OK* 를 클릭하십시오.

### 방위표

*North Arrow* 를 통해 맵 캔버스 위에 방위표를 추가할 수 있습니다.

방위표를 추가하려면:

1. *View* → *Decorations* → *North Arrow*…메뉴 옵션을 선택해서 대화창을 여십시오.
2. *Enable North Arrow* 체크박스가 체크돼 있는지 확인하십시오.
3. 색상 및 크기 옵션을 변경하거나, 사용자 지정 SVG 를 선택하십시오.
4. 각도 옵션을 변경하거나, **자동 선택 (Automatic)** 을 선택해서 QGIS 가 방향을 결정하도록 하십시오.

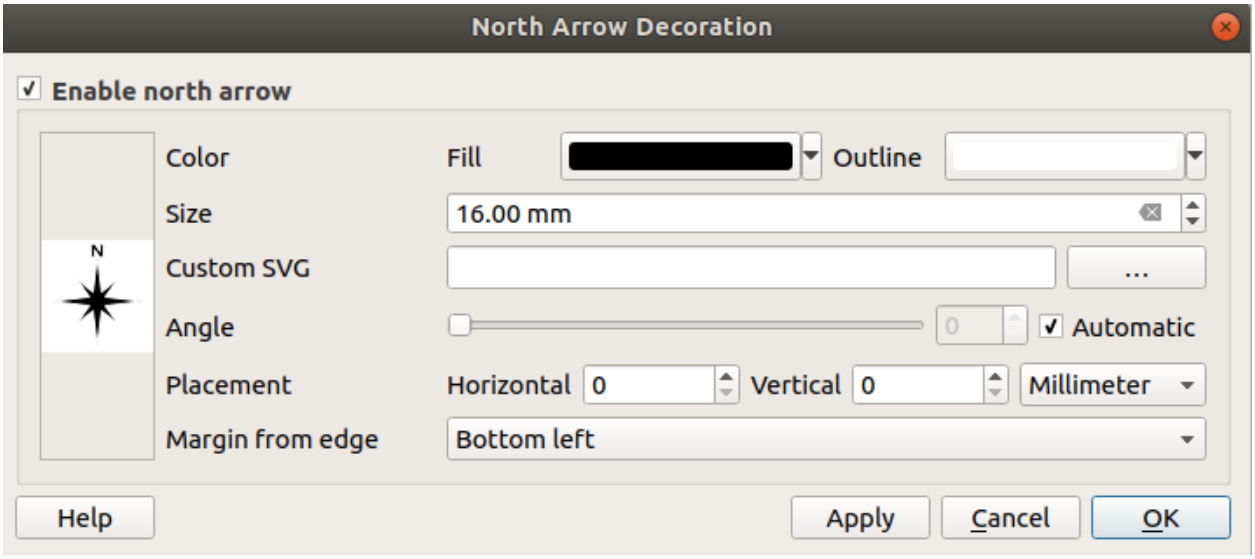



그림 7.9: 방위표 대화창

5. *Placement* 콤보박스에서 방위표의 배치를 선택할 수도 있습니다.
6. *Margin from (Canvas) Edge* 에서 수평 그리고/또는 수직 경계 여백을 설정해서 항목의 배치를 미세 조정하십시오. **밀리미터** 또는 **픽셀** 값으로 설정할 수도 있고, 맵 캔버스의 너비 또는 높이의 **백분율** 값으로 설정할 수도 있습니다.
7. 기대한대로 모양이 나오는지 확인하려면 *Apply* 를, 사용자가 만족한다면 *OK* 를 클릭하십시오.

### 축척 막대

*Scale Bar* 는 맵 캔버스에 단순한 축척 막대를 추가합니다. 사용자가 축척 막대의 스타일과 배치는 물론 라벨 작업도 제어할 수 있습니다. 축척 막대는 *Project properties* ▸ *General* ▸ *Units for distance measurement* 에서 정의된 활성프로젝트의 거리 단위 를 따릅니다.

축척 막대를 추가하려면:

1. *View* ▸ *Decorations* ▸ *Scale Bar*…메뉴 옵션을 선택해서 대화창을 여십시오.
2. *Enable Scale Bar* 체크박스가 체크돼 있는지 확인하십시오.
3. *Scale bar style* 콤보박스를 이용해서 축척 막대의 스타일을 선택하십시오.
4. *Color of bar* 에서 채우기 색상 (기본값: 검정색) 및 외곽선 색상 (기본값: 흰색) 을 선택하십시오. 색상 입력 오른쪽에 있는 아래쪽 화살표를 클릭하면 축척 막대의 채우기 및 외곽선을 불투명하게 만들 수 있습니다.
5. *Font of bar* 콤보박스에서 축척 막대 용 글꼴을 선택하십시오.
6. *Size of bar* 에서 축척 막대의 크기를 활성 단위로 설정하십시오.
7. *Automatically snap to round number on resize* 옵션을 체크하면 읽기 쉬운 값을 표시하도록 할 수 있습니다.
8. *Placement* 콤보박스에서 축척 막대의 배치를 선택하십시오.

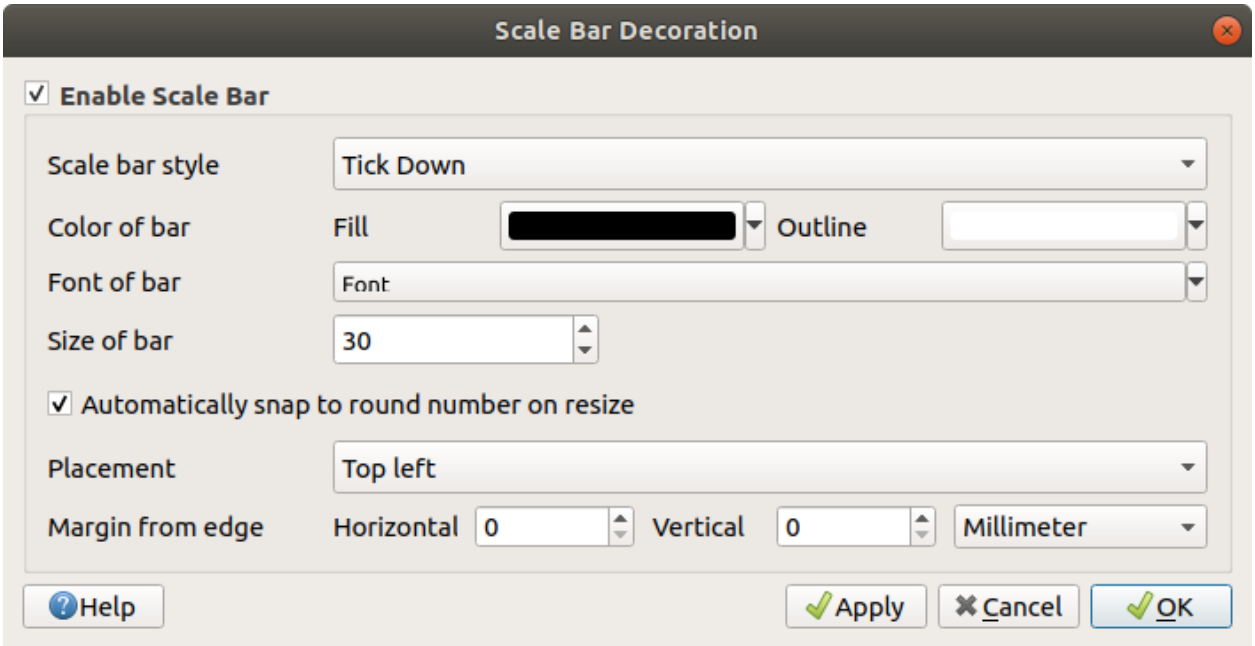



그림 7.10: 축척 막대 대화창

9. *Margin from (Canvas) Edge* 에서 수평 그리고/또는 수직 경계 여백을 설정해서 항목의 배치를 미세 조정하십시오. **밀리미터** 또는 **픽셀** 값으로 설정할 수도 있고, 맵 캔버스의 너비 또는 높이의 **백분율** 값으로 설정할 수도 있습니다.
10. 기대한대로 모양이 나오는지 확인하려면 *Apply* 를, 또는 사용자가 만족한다면 *OK* 를 클릭하십시오.

### 조판 범위

*Layout Extents* 도구는 캔버스에 인쇄 조판기 (들) 에 있는맵 항목 *(들)* 의 범위를 추가합니다. 이 도구를 활성화하면 모든 인쇄 조판기에 있는 모든 맵 항목의 범위를 인쇄 조판기 및 맵 항목의 이름 라벨이 붙은 점선으로 된 경계선으로 표시합니다. 표시된 조판 범위의 스타일과 라벨을 제어할 수 있습니다. 사용자가 라벨 같은 맵 요소의 위치를 조정하는 도중 인쇄 조판기의 실제 가시화 영역을 알아야 하는 경우 이 장식이 유용합니다.

조판 범위 (들) 를 추가하려면:

1. *View* → *Decorations* → *Layout Extents* 메뉴 옵션을 선택해서 대화창을 여십시오.
2. *Show layout extents* 체크박스가 체크돼 있는지 확인하십시오.
3. 범위의 심볼 및 라벨을 변경할 수 있는 옵션이 있습니다.
4. 기대한대로 모양이 나오는지 확인하려면 *Apply* 를, 사용자가 만족한다면 *OK* 를 클릭하십시오.

> **팁**
>
> **장식 설정**
>
> 사용자가 QGIS 프로젝트 파일을 저장할 때, 프로젝트에 그리드, 방위표, 축척 막대, 저작권 및 조판 범위의 모든 변경 사항이 저장되어 다음에 프로젝트를 불러올 때 복원될 것입니다.




그림 7.11: QGIS 프로젝트에 인쇄 조판기 2 개의 조판 범위가 표시돼 있는 예시. 'Sights'라는 인쇄 조판기는 맵 항목 2 개를 담고 있는 반면, 다른 인쇄 조판기는 맵 항목 1 개를 담고 있다.

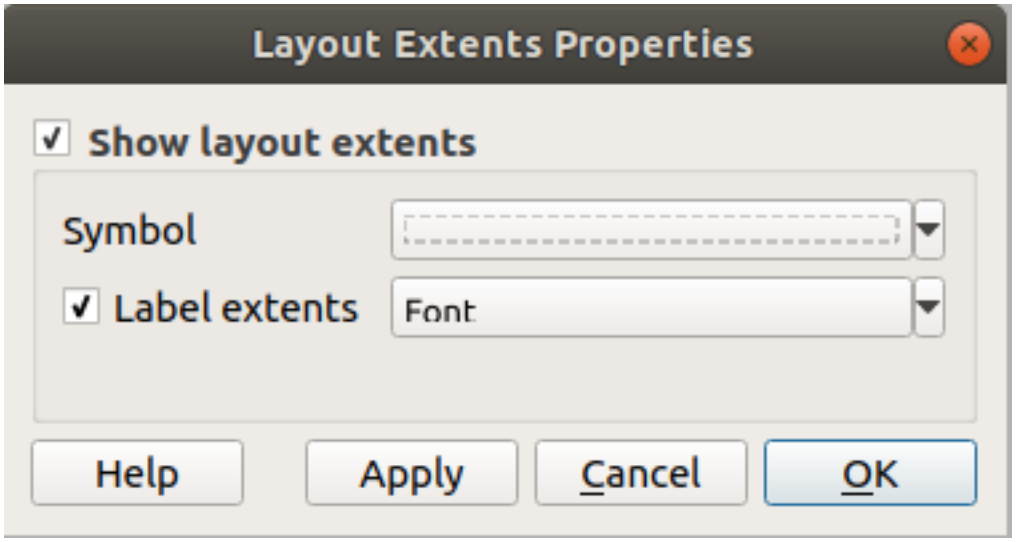



그림 7.12: 조판 범위 대화창

### 7.1.7 주석 도구

주석이란 맵 캔버스 상에 렌더링된 레이어로서는 표현할 수 없는 추가 정보를 제공하기 위해 추가되는 또다른 유형의 요소입니다. 벡터 레이어에 저장된 속성값에 의존하는라벨 과는 달리, 주석은 프로젝트 파일 자체에 저장된 독립적인 세부 정보입니다.

QGIS 에서는 두 가지 계열의 주석을 사용할 수 있습니다:

- **피처 주석**: 이 주석 유형은"주석 레이어"라는 특수 레이어 안에 저장된 텍스트, 마커, 라인, 또는 폴리곤 유형의 지리참조된 실제 피처입니다. 이 주석은 특정 지리 위칭에 고정되어 있습니다. 다시 말해 사용자가 맵을 이동하거나, 축척을 변경하거나, 또는 투영법을 변경해도 이 주석의 위치는 변하지 않고 사용자가 주석을 그린 위치에 그대로 있을 겁니다.
- **Balloon annotations**: these are individual HTML or form annotations inside a bubble. They can be associated to any layer for their visibility and drawn on top of the map canvas. The size is dependent from the map canvas scale, and its position can be anchored.

> **팁**
>
> **주석을 가진 맵을 조판하기**
>
> 다음 도구를 사용해서 사용자 맵을 주석과 함께 인쇄하거나 여러 포맷으로 내보낼 수 있습니다:
>
> - *Project* 메뉴에서 사용할 수 있는 맵 캔버스 내보내기 도구들
> - 인쇄 조판기 를 이용하는 경우, 내보내고자 하는 맵 항목 속성에서 *Draw map canvas items* 옵션을 활성화해야 합니다.

*Annotations Toolbar* 가 두 가지 주석 계열을 생성하고 대화형 작업을 할 수 있는 도구 집합을 제공합니다.

표 7.3: 주석 툴바 작업

| 도구 | 사용법 | 계열 |
|---|---|---|
| New Annotation Layer | 주석을 저장할 새 레이어를 생성합니다. | *Feature annotations* |
| Main Annotation Layer Properties | 주 주석 레이어의 설정을 제어합니다. | |
| Modify Annotations | 주석의 심볼 속성을 선택, 이동, 크기 조정 및 수정합니다. | |
| Create Polygon Annotation | 폴리곤 피처 주석을 생성합니다. | |
| Create Line Annotation | 폴리라인 피처 주석을 생성합니다. | |
| Create Marker Annotation | 포인트 피처 주석을 생성합니다. | |
| Create Text Annotation at Point | Create an annotation as a text label anchored at a point | |
| Create Text Annotation along Line | Create an annotation as a curved text along a line feature | |
| Create Text Annotation in Rectangle | Create an annotation as a text label in a rectangular text box | |
| Create Picture Annotation | Create an annotation representing an `.SVG` or raster image | |
| HTML Annotation | Select and create annotation with a `HTML` formatted text | *Balloon annotations* |
| Form Annotation | 사용자 지정 양식 파일에 있는 벡터 레이어 및 속성을 선택해서 그 속성을 보여주는 말풍선 주석을 생성합니다. | |

> **참고**
>
> Starting with QGIS 3.40, it is no longer possible to create balloon annotations of SVG or text type. When loading old projects, any of these annotations will automatically be converted to the newer picture and rectangular text annotation item types.

### 피처 주석

Feature annotations are stored in **annotation layers**. Unlike conventional layers, an annotation layer is available only in the current project and can contain features of different types (text, marker, line, polygon, picture). The layer has no attributes and no symbology associated, but instead each feature can be symbolized on an item-by-item basis, through *Layer Styling* panel.

QGIS 에서는 두 가지 계열의 주석 레이어를 사용할 수 있습니다:

- *Annotation Layer*: 일반 주석 레이어.  New Annotation Layer 도구를 사용해서 생성할 수 있습니다. 일반 주석 레이어는 *Layers* 패널의 레이어 목록에 포함되기 때문에, 다른 레이어와 마찬가지로 피처의 가시성을 제어하고 사용자 맵에서 특정 레이어의 위 또는 아래에 표시되도록 이동시킬 수 있습니다. 이 레이어를 더블클릭하면 레이어 속성에 접근할 수 있습니다.
- *Main Annotation Layer*: 주 주석 레이어. 기본적으로 프로젝트에서 사용 가능한 주석 레이어가 없거나, 생성 시 선택한 주석 레이어가 없는 경우 이 레이어에 주석을 저장합니다. 이 레이어는 언제나 사용자 맵 가장 위에 렌더링되며, *Layers* 패널의 레이어 목록에 사용자 프로젝트의 다른 레이어와 함께 표시되지 않습니다. 즉 주 주석 레이어의 피처는 항상 가시화되어 있는 상태입니다. *Annotations* 툴바의 *Main Annotation Layer Properties* 항목을 클릭하면 주 주석 레이어의 속성 대화창을 열 수 있습니다.

### 레이어 속성

주석 레이어의 속성 대화창은 다음 탭들을 제공합니다:

- *Information*: 현재 레이어에 대한 요약 정보 및 메타데이터를 빠르게 확인할 수 있는 읽기 전용 대화창입니다. 레이어 범위, 주석 유형 별 항목 개수 및 총 개수, 좌표계 상세 정보 등등을 포함할 수도 있습니다.
- *Source*: 소스 탭에서는 주석 레이어의 다음과 같은 일반 설정을 정의할 수 있습니다:
  - *Layer name*: (*Layers Panel* 에서 표현식 등을 이용해서) 프로젝트 내의 레이어를 식별하기 위해 쓰일 레이어명을 설정할 수 있습니다.
  - *Assigned Coordinate Reference System (CRS)*: 레이어에할당된 좌표계 를 표시합니다. 드롭다운 목록에서 최근 사용된 좌표계를 선택하거나  Select CRS 버튼을 (좌표계 선택기 참조) 클릭하면 레이어 좌표계를 변경할 수 있습니다. 레이어에 적용된 좌표계가 틀렸거나 적용된 좌표계가 없는 경우에만 변경하십시오.
- *Rendering*:
  - *Maximum (inclusive)* 및 *Minimum (exclusive)* 축척을 설정해서 피처가 보이게 될 축척 범위를 정의할 수 있습니다. 이 범위를 벗어나면, 피처를 숨깁니다.  Set to current canvas scale 버튼을 클릭하면 현재 맵 캔버스의 축척을 가시성 범위의 한계값으로 설정할 수 있습니다. 자세한 내용은가시성 축척 선택기 을 참조하세요.
  - *Linked layer*: allows you to optionally set a linked visibility layer for the annotation layer. If set, then the annotations will only be drawn when the linked layer is visible in the map.

  - *Opacity*: 이 도구를 통해 맵 캔버스에서 아래에 있는 레이어를 가시화할 수 있습니다. 슬라이드 바를 통해 사용자 벡터 레이어의 가시성을 필요에 따라 조정하십시오. 슬라이드 바 옆에 있는 메뉴에서 가시성을 정확한 백분율로 설정할 수도 있습니다.
  - *Layer* 수준의 *Blending mode*: 이 도구들을 통해 사용자가 그래픽 프로그램에서만 가능하다고 알고 있던 특별한 렌더링 효과를 줄 수 있습니다. 위 그리고 아래에 있는 레이어의 픽셀을혼합 모드 에서 설명한 설정으로 혼합합니다.
  - *Draw Effects* 버튼: 레이어의 모든 피처에 대해그리기 효과 를 줍니다.

피처 주석의 *Symbology* 속성에서 이 옵션 가운데 몇몇에 접근할 수 있습니다.

### 대화형 피처 작업

The Feature annotations have dedicated tools for creation depending on their type.

표 7.4: Creating feature annotation

| Family / based on | Annotation tool | 절차 |
|---|---|---|
| 도형 | Create Polygon Annotation | This is like digitizing a point, line or polygon vector feature. All the usual shortcuts for creating features apply when creating annotation items. A line or polygon annotation is drawn by left-clicking once for each vertex, with a final right mouse click to complete the shape. Snapping can be enabled while you draw, you can use the *Advanced Digitizing Tools* to precisely place vertices, and even switch the *drawing tools* to for example the streaming mode for completely free-form shapes. |
| | Create Line Annotation | |
| | Create Marker Annotation | |
| Text | Create Text Annotation at Point | Left-click on the map canvas. |
| | Create Text Annotation along Line | Left-click to add vertices to the text basement line geometry, and right-click to end. As above, digitizing features capabilities are also available. |
| | Create Text Annotation In Rectangle | Draw the rectangle box of the text, by left-clicking to add a first corner and left-clicking again to add the opposite one. |
| Picture | Create Picture Annotation | Draw the rectangle box of the picture, by left-clicking to add a first corner and left-clicking again to add the opposite one. Then pick an image or SVG file from the file explorer. |

일반 레이어와는 달리, 사용자가 피처를 선택하기 전에 주석 레이어를 활성화시키지 않아도 됩니다. 그냥  Modify Annotations 도구를 선택하면 어떤 피처 주석과도 대화형 작업을 할 수 있습니다:

- **선택**: 주석을 왼쪽 클릭하십시오. 기본적으로 주석은 생성 순서대로 렌더링됩니다. 즉 최근에 생성된 주석이 더 전에 생성된 주석 위에 배치될 것이라는 뜻입니다. 사용자가 위에 배치되길 바라는 주석을 선택하려면 피처의 Z 인덱스 속성을 조정해야 할 수도 있습니다.
- **이동**: 선택한 주석 항목을 왼쪽 클릭하면 주석을 움직일 수 있습니다. 오른쪽 클릭 또는 `Esc` 키를 누르면 이동을 취소하고, 다시 한번 왼쪽 클릭하면 주석이 움직인 위치로 이동합니다. 화살표 키를 눌러 이동을 제어할 수도 있습니다:
  - `Shift+key`: 이동 거리가 커집니다.

  - `Alt+key`: `1 px` 단위로 이동합니다.

- **Geometry modification**: move over an annotation and purple squares are displayed on nodes of its underlying geometry. Left-click on the square, move and click again. For linear or polygonal annotations, double-clicking a segment will add a new vertex.
- **삭제**: 주석을 선택한 상태에서 `Del` 또는 `Backspace` 키를 누르면 선택한 주석을 삭제할 것입니다.
- 피처 심볼 변경

### 피처 심볼

A selected annotation will display its *Symbology* properties in the *Layer styling* panel.

- For polygon, polyline and marker annotations, you can modify the appearance using full capabilities of the *symbol properties*
- For text-based annotations, an area with text editing tools helps you construct the string to display. It can be multiline, use HTML formatting and rely on QGIS expression functions. You can moreover modify the appearance using full capabilities of the *text format properties*.

  Depending on the type of annotation, additional options are available.

  - For text annotation at point, it is also possible to:
    * set the text *Alignment* (left, center or right of the anchor point)
    * configure whether the text should *Ignore map rotation* or *Rotate with map*. In both cases, a custom *Angle* can be set for the feature orientation.
  - 라인을 따라 텍스트 주석을 생성하는 경우, *Offset from line* 라인으로부터의 오프셋을 사용자가 선택한 단위로 환경설정하십시오.
  - For text annotation in rectangle, it is also possible to:
    * set the text *Horizontal alignment* (left, center, right or justify) and *Vertical alignment* (top, vertical center or bottom) in the rectangle box
    * configure for the rectangle, the *Margins* distance in the units of your choice, the *Frame* and *Background* colors.
- Picture annotations allow you to display a *Raster image* or an *SVG image*, using the 원격 또는 내장 파일 선택기. The placement and size of the picture can be set as:
  - *Scale dependent size*: the picture scales with the map, appearing larger when zooming in and smaller when zooming out. The size is defined by a single reference size in map units (e.g. meters or degrees, depending on the layer's CRS). With  *Lock aspect ratio*, you can keep the annotation size proportional to the embedded picture size, or stretch the picture to fill the annotation extent.
  - *Fixed size*: the picture annotation maintains a constant size in screen or output units (e.g. pixels or millimeters), regardless of the map scale. The *Width* and *Height* of the annotation can be filled separatey or adjusted proportionally to the embedded picture size.
  - *Relative to map*: the picture annotation will always be rendered at the same position relative to the map canvas bounds, regardless of the map scale. The *Width* and *Height* of the annotation can be filled separatey or adjusted proportionally to the embedded picture size.

  Moreover, you can enable display of the *Frame* and *Background* of the rectangle box, and configure them using *fill symbols*.
- For text at point, text inside rectangle and picture annotations, you can *Show callout* when the picture or text anchor point is offset from its default placement. To create a callout for an annotation:

1. Select the annotation item
2. Move the anchor point, i.e., click the central and green `X` node and click at the new placement
3. In the annotation properties, press the ⋯ button to configure the *callout properties*.

> **팁**
>
> Applying a balloon callout to a text or picture annotation will make you get their old-style balloon annotation display.

- Configure a *Reference scale*: indicates the map scale at which symbol or text sizes which use paper-based units (such as millimeters or points) relate to. The sizes will be scaled accordingly whenever the map is viewed at a different scale. For instance, a line feature wide of 2mm at 1:2000 *Reference scale* will be rendered using 4mm when the map is viewed at 1:1000.
- *Z-index* 설정: 높은 Z 인덱스를 가진 피처를 낮은 Z 인덱스를 가진 피처 위에 그립니다. 피처 보이기 및 선택 양쪽에 모두 편리한 설정입니다.
- 몇몇레이어 렌더링 설정을 수정하십시오.

### 말풍선 주석

*Edit* → *Add Annotation* → 메뉴 항목을 선택하면 또는 *Annotations Toolbar* 를 통해 말풍선 주석을 추가할 수 있습니다:

- HTML Annotation to display a HTML-formatted string or a whole `HTML` file as an annotation.
- Form Annotation: useful to display attributes of a vector layer in a customized `.ui` file. This is similar to the *custom attribute forms*, but displayed in an annotation item.

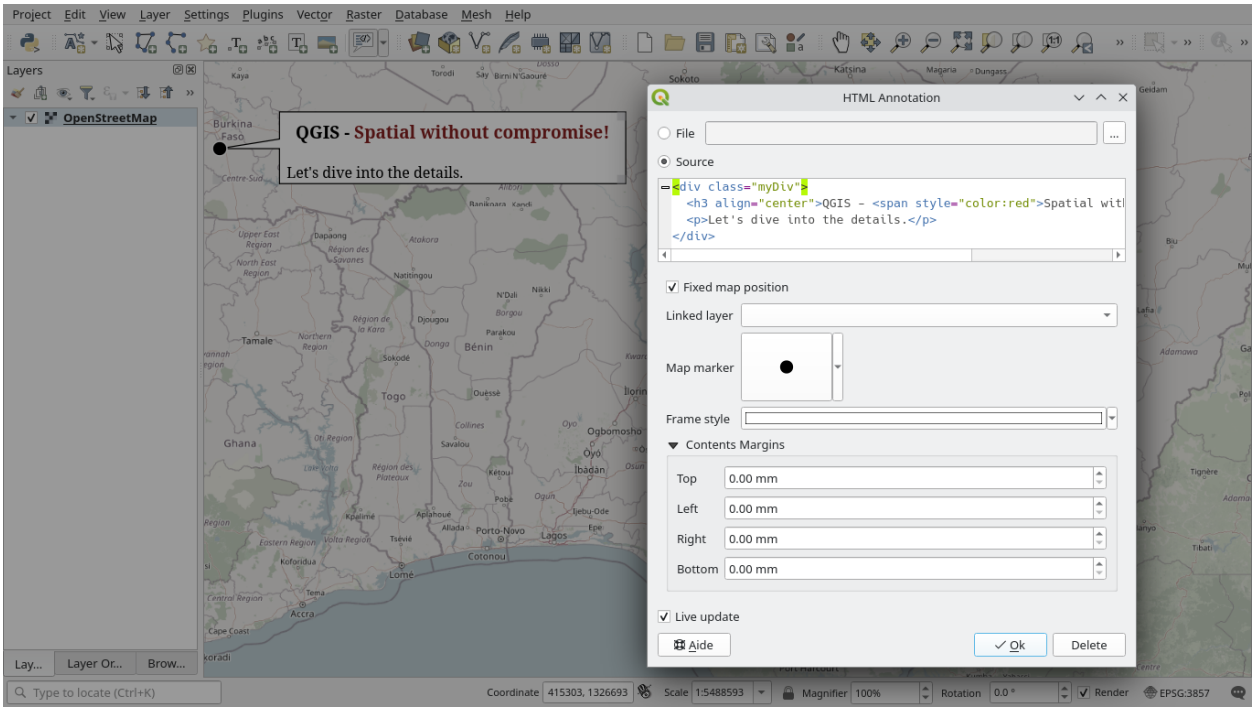



그림 7.13: Example of a HTML balloon annotation

말풍선 주석을 추가하려면, 대응하는 도구를 선택한 다음 맵 캔버스를 클릭하십시오. 비어 있는 말풍선이 추가됩니다. 이 말풍선을 더블클릭하면 여러 옵션을 담은 대화창이 열립니다. 모든 주석 유형에 대해 이 대화창은 거의 동일합니다:

- *File*: Fill it with the path to a `.html` or `.ui` file, depending on the type of annotation, to serve as contents of the annotation.
- *Source*: Only available for HTML annotations, it allows to serve the annotation with a plain or HTML-formatted text instead of a file.
- *Fixed map position*: when unchecked, the balloon placement is based on a screen position (instead of the map), meaning that it's always shown regardless of the map canvas extent.
- *Linked layer*: 주석을 맵 레이어와 연결해서 해당 레이어가 가시화될 때만 주석을 가시화합니다.
- *Map marker*: *QGIS 심볼* 을 사용해서 말풍선 고정 위치 (anchor position) 에 표시될 심볼을 설정합니다. (*Fixed map position* 옵션을 체크한 경우에만 심볼을 표시합니다.)
- *Frame style*: QGIS 심볼을 통해 주석 프레임의 배경색, 투명도, 획 색상 또는 말풍선의 너비를 설정합니다.
- *Contents margins*: 주석 프레임의 내부 여백을 설정합니다.
- *Live update*: 사용자가 변경하는 미리보기를 실시간으로 볼 수 있게 해줍니다.

말풍선 주석 도구가 활성화되어 있는 동안 다음 작업도 할 수 있습니다:

- 주석을 선택할 수 있습니다.
- 주석 크기를 조정할 수 있습니다.
- 맵 위치를 (맵 마커를 드래그해서) 이동시키거나 또는 말풍선만 이동시키는 방법으로 주석을 이동시킬 수 있습니다.
- 주석을 삭제할 수 있습니다: 주석을 선택한 다음 `Del` 키 또는 `Backspace` 키를 누르거나, 또는 주석을 더블클릭한 다음 속성 대화창에 있는 *Delete* 버튼을 클릭하십시오.
- 주석을 오른쪽 클릭하면 다음 컨텍스트 메뉴를 사용할 수 있습니다:
  - *Copy coordinate*: 주석의 맵 마커 좌표를 여러 좌표계로 복사할 수 있습니다.
  - *Edit*: 주석 속성을 편집할 수 있습니다. 주석을 더블클릭하는 것과 동일합니다.
  - *Delete*: 주석을 삭제할 수 있습니다.

### 7.1.8 측정

#### 일반 정보

QGIS 는 도형을 측정하는 네 가지 방법을 제공하고 있습니다:

- 쌍방향 측정 도구인 과,
- Field Calculator 에서 측정하는 방법,
- 피처 식별 도구에서 파생된 측정 방법,
- 그리고 *Vector → Geometry Tools → Export/Add Geometry Columns* 메뉴의 벡터 분석 도구가 있습니다.

측정 작업은 (UTM 같은) 투영좌표계 및 투영체가 정의되지 않은 데이터 안에서 이루어집니다. 처음 세 가지 측정 도구의 습성은 일반 투영 설정과 동일합니다:

- 다른 대부분의 GIS 와는 달리, 기본 측정법은 *Project* ▸ *Properties…* ▸ *General* 에서 정의한 타원체를 이용합니다. 프로젝트 용으로 지리 좌표계를 정의한 경우, 투영 좌표계를 정의한 경우 둘 다 타원체 상에서 측정합니다.
- 데카르트 수학을 이용해서 투영/평면 면적 또는 거리를 계산하려면, *Project* ▸ *Properties…* ▸ *General* 메뉴에서 측정 타원체를"None/Planimetric"으로 설정해야 합니다. 하지만 데이터 및 프로젝트 용으로 지리 (= 비투영) 좌표계를 정의했다면, 타원체 상에서 면적 및 거리를 측정할 것입니다.

그러나 식별 도구 또는 필드 계산기 둘 다 측정 작업 전에 사용자 데이터를 프로젝트 좌표계로 변환하지 않을 겁니다. 측정 전에 좌표계를 변환하도록 하고 싶다면, *Vector* ▸ *Geometry Tools* ▸ *Add Geometry Attributes…*메뉴의 벡터 분석 도구를 이용해야 합니다. 이때 사용자가 타원체 측정법을 선택한 경우를 제외하면 기본적으로 평면 상에서 측정하게 됩니다.

### 길이, 면적, 방향 및 각도를 쌍방향으로 측정하기

측정을 시작하려면 속성 툴바에 있는  아이콘을 클릭하십시오. 이 아이콘 옆에 있는 아래쪽 화살표를 누르면  길이,  면적,  방향, 또는  각도로 전환할 수 있습니다. 이 대화창에서 쓰이는 기본 단위는 *Project* ▸ *Properties…* ▸ *General* 메뉴에서 설정된 단위입니다.

*Measure Line* 과 the *Measure Area* 의 경우,  *Cartesian* 데카르트 평면 상 또는  *Ellipsoidal* 타원체 표면 상 가운데 하나로 측정할 수 있습니다.

> **참고**
>
> **측정 도구 환경 설정**
>
> 길이 또는 면적 측정 시, 위젯 하단에 있는 *Configuration* 버튼을 클릭하면 *Settings* ▸ *Options* ▸ *Map Tools* 메뉴와 동일하게 고무줄 색상, 측정 정밀도 및 단위 습성을 정의할 수 있습니다. 사용자가 선호하는 측정 단위 또는 각도 단위를 선택할 수도 있지만, *Project* ▸ *Properties…* ▸ *General* 메뉴 그리고 측정 위젯에서 선택한 옵션이 현재 프로젝트에서 사용자가 선택한 단위를 대체한다는 사실을 유념해야 합니다.

모든 측정 모듈은 디지타이즈 작업 모듈의 스냅 설정 (스냅 허용 오차 및 검색 반경 설정 참조) 을 이용합니다. 따라서 라인 피처를 따라서든 폴리곤 피처를 돌아서든, 정확히 측정하려면 먼저 레이어 스냅 허용 오차를 설정해야 합니다. 그러면 측정 도구 사용 시 (허용 오차 설정 범위 내의) 각 마우스 클릭이 해당 레이어에 스냅할 것입니다.

 Measure Line 도구는 지정한 포인트들 사이의 거리를 측정합니다. 이 도구는 사용자가 맵 상에 포인트를 클릭할 수 있게 해줍니다. 각 선분의 길이는 물론 전체 길이가 측정창에 표시됩니다. 측정창에서 사용자가 클릭한 모든 포인트들과 거리들의 좌표를 볼 수 있을 것입니다. 첫 행은 사용자의 시작 포인트를 나타내기 때문에 좌표만 담고 있을 것이라는 점을 기억하십시오. 이제 *Copy* 버튼을 클릭하면 사용자의 라인 측정값들을 클립보드로 한번에 복사할 수 있습니다. *Configuration* 버튼을 클릭하면복사 옵션 을 설정할 수 있는 *Measure Tool Copy Settings* 가 열릴 것입니다. 측정을 마치려면 오른쪽 클릭하면 됩니다.

측정 도구로 작업하는 동안 전체 거리 옆에 있는 드롭다운 목록을 사용해서측정 단위 를 대화형으로 변경할 수 있다는 사실을 기억하십시오. 새 프로젝트를 생성하거나 또다른 프로젝트를 열기 전까지 위젯에 이렇게 선택한 단위를 유지시킬 것입니다.

대화창의 *Info* 부분은 사용할 수 있는 좌표계 설정에 따라 어떻게 계산하는지 설명합니다.

 Measure Area: 면적도 측정할 수 있습니다. 누적 면적 크기가 측정창에 나타납니다. 그리기를 마치려면 오른쪽 클릭하면 됩니다. 서로 다른면적 단위 사이를 전환할 수 있는 기능은 물론이고, 정보 부분도 볼 수 있습니다.

 Measure Bearing 도구로 방향도 측정할 수 있습니다. 커서가 십자가 형태로 바뀝니다. 사용자가 측정하려는 방향의 첫 번째 포인트를 클릭한 다음, 커서를 움직여서 원하는 포인트를 클릭하십시오. 측정값이 팝업 대화창에 표시됩니다.

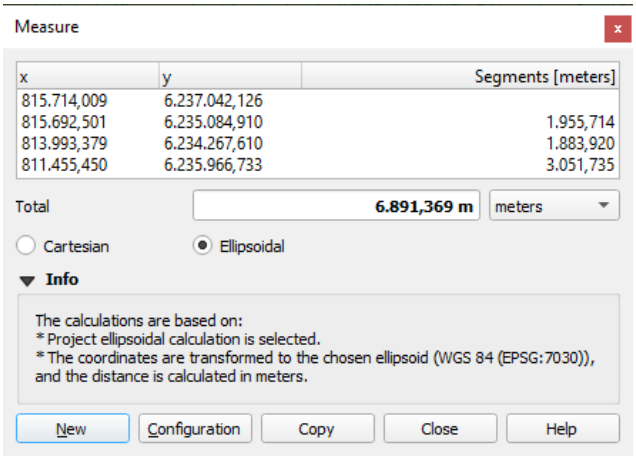



그림 7.14: 거리 측정

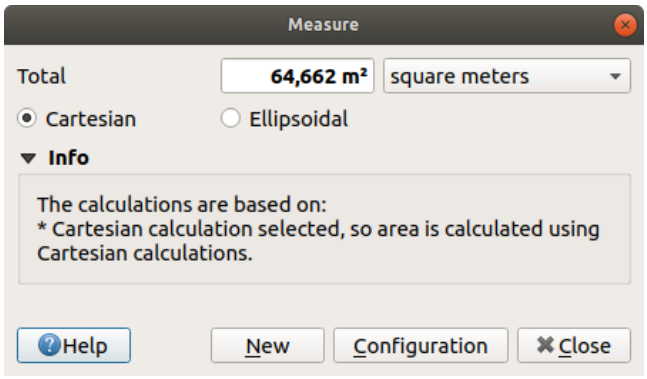



그림 7.15: 면적 측정

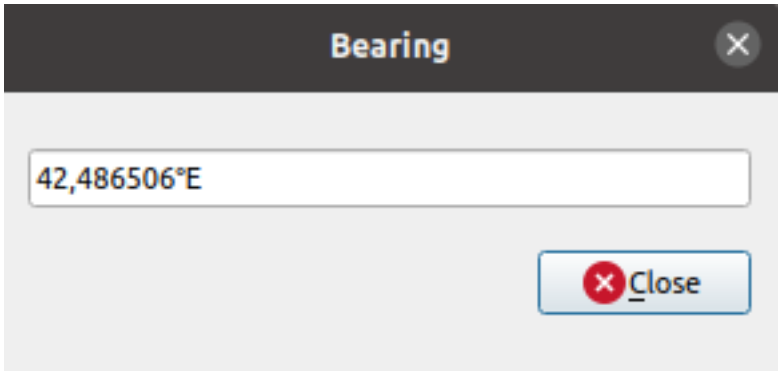



그림 7.16: 방향 측정

Measure Angle 도구로 각도도 측정할 수 있습니다. 커서가 십자가 형태로 바뀝니다. 사용자가 측정하려는 각도의 첫 번째 선분을 그린 다음, 커서를 움직여서 원하는 각도를 그리십시오. 측정값이 팝업 대화창에 표시됩니다.

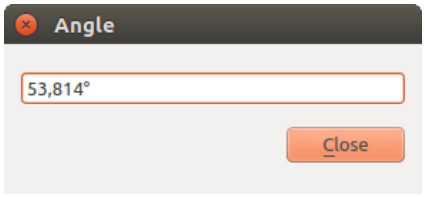



그림 7.17: 각도 측정

### 7.1.9 추가 맵 뷰 설정하기

*Layers* 패널의 현재 상태로부터 뷰 내용을 분기시킬 수 있는 추가적인 맵 뷰를 열 수도 있습니다. 새 맵 뷰를 추가하려면 *View* *New Map View* 메뉴 항목을 선택하십시오. 주 맵 뷰의 렌더링을 모방하는 새 플로팅 위젯 (floating widget) 이 QGIS 에 추가될 것입니다. 사용자가 필요한 개수만큼 맵 뷰를 추가할 수 있습니다. 이 추가 맵 뷰들을 나란히 배열하거나 아래 위로 쌓아놓는 방식으로 계속 부유 (float) 하는 상태로 유지할 수 있습니다.

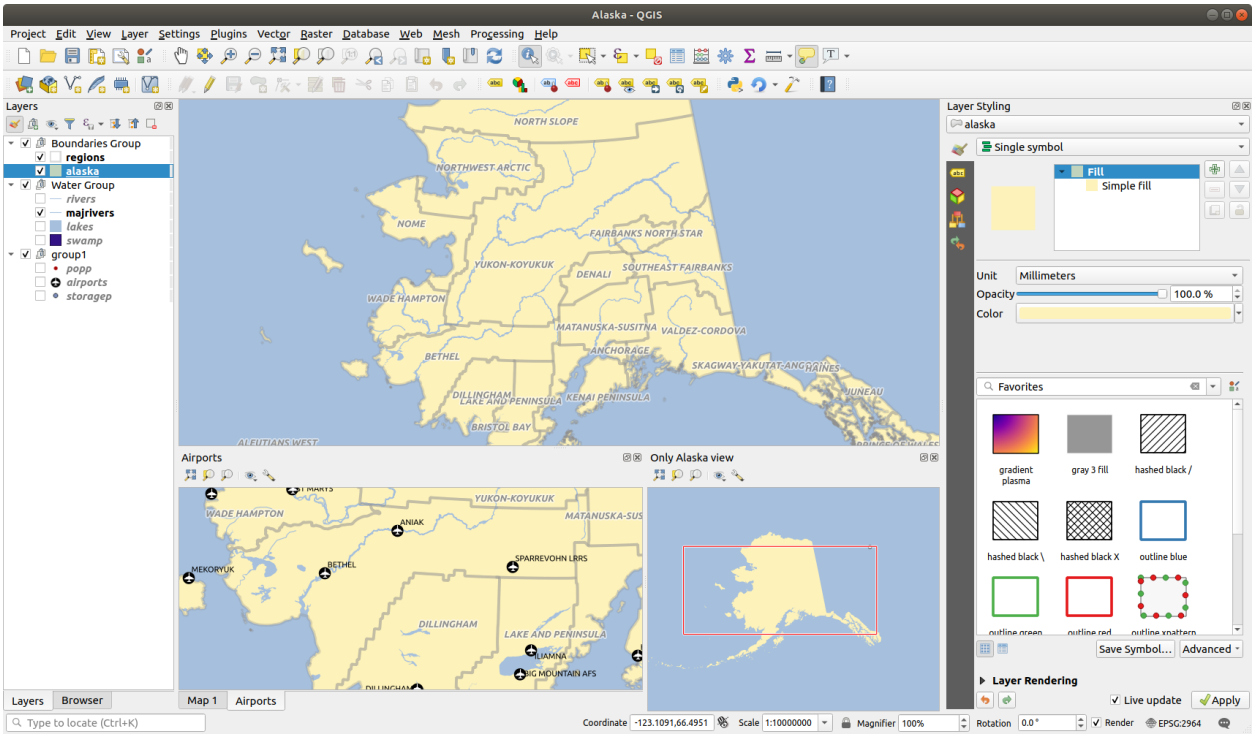



그림 7.18: 서로 다른 설정을 가진 여러 개의 맵 뷰들

추가된 맵 캔버스의 상단에는 다음 기능을 가진 툴바가 존재합니다:

- Zoom Full, Zoom to Selection 및 Zoom to Layer(s): 각각 맵 뷰의 전체 범위로 확대/축소, 선택 피처 (들) 의 범위로 확대/축소, 선택한 레이어 (들) 의 범위로 확대/축소해서 맵 뷰 내부를 탐색할 수 있습니다.
- Set View Theme: 맵 뷰 안에 표시되는맵 테마 를 선택할 수 있습니다. 테마를 `(none)` 으로 설정하면, 뷰가 *Layers* 패널의 변경 사항을 반영할 것입니다.

- View settings: 맵 뷰의 다음 설정들을 정의합니다:
  - *Synchronize view center with main map*: 축척을 변경하지 않으면서 맵 뷰들의 중앙을 동기화합니다. 이 옵션을 활성화하면 메인 맵의 중앙을 따르는 오버뷰 형식 또는 확대 맵을 사용할 수 있습니다.
  - *Synchronize view to selection*: 선택한 피처로 확대/축소 기능과 동일합니다.
  - *Scale*: 맵 뷰의 축척을 설정합니다.
  - *Rotation*: 마커의 기울기를 설정할 수 있습니다.
  - *Magnification*: 맵 뷰의 확대 비율을 설정합니다.
  - *Synchronize scale*: 메인 맵 축척과 동기화합니다. 이 옵션을 활성화하면 *Scale factor* 를 적용할 수 있습니다. 이를 테면 언제나 메인 맵 축척의 2 배 축척인 뷰를 설정할 수 있게 됩니다.
  - *Show annotations*: 맵 뷰에 주석을 표시합니다.
  - *Show cursor position*: 맵 뷰에 커서 위치를 표시합니다.
  - *Show main canvas extent*: 맵 뷰에 메인 캔버스의 범위를 표시합니다.
  - *Show labels*: 이 옵션을 해제하면 표시된 레이어의 속성에 라벨이 설정되어 있더라도 맵 뷰에 라벨을 보이지 않게 할 수 있습니다.
  - *Change map CRS*…: 맵 뷰의 좌표계를 변경합니다.
  - *Rename view*…: 맵 뷰의 이름을 변경합니다.

### 7.1.10 맵 뷰 내보내기

인쇄 조판기 또는 보고서 의 고급 기능을 이용하면 사용자가 만든 맵을 조판해서 여러 포맷으로 내보낼 수 있습니다. 물론 현재 렌더링을 조판하지 않은 채로 그대로 내보낼 수도 있습니다. 이런 맵의 즉석"스크린샷"에는 몇 가지 편리한 장점이 있습니다.

맵 캔버스를 현재 렌더링 그대로 내보내려면:

1. *Project* → *Import/Export* 메뉴 항목을 선택하십시오.
2. 사용자의 산출물 포맷에 따라, 다음 중 하나를 선택하십시오:
   - *Export Map to Image*…: 맵을 이미지로 내보냅니다.
   - 또는 *Export Map to PDF*…: 맵을 PDF 파일로 내보냅니다.

이 2 개의 도구는 일반적인 옵션 모음을 제공합니다. 다음 대화창이 열리면:

1. *Extent*: 내보낼 범위를 선택합니다. 현재 뷰의 범위 (기본값) 일 수도 있고, 레이어의 범위 또는 맵 캔버스 위에 그린 사용자 지정 범위일 수도 있습니다. 선택한 영역의 좌표가 표시되며, 직접 편집할 수 있습니다.
2. *Scale*: 맵의 축척을 입력하거나사전에 정의된 축척 가운데 선택할 수 있습니다. 축척을 변경하면 내보낼 범위의 크기를 (중앙을 기준으로) 다시 조정할 것입니다.
3. *Resolution*: 산출물의 해상도를 설정합니다.
4. *Output width* 와 *Output height*: 이미지의 너비와 높이를 픽셀 단위로 조정합니다. 기본적으로 현재 해상도와 범위를 기준으로 하지만, 이들을 사용자 지정하면 맵 범위의 크기를 (중앙을 기준으로) 다시 조정할 것입니다. 크기 비율을 고정시킬 수 있는데, 캔버스 위에 사용자 지정 범위를 그릴 경우 매우 편리할 수도 있습니다.
5. *Draw active decorations*: 이 옵션을 활성화하면지도 장식 (축척 막대, 제목, 그리드, 방위표 등등) 을 맵과 함께 내보낼 수 있습니다.

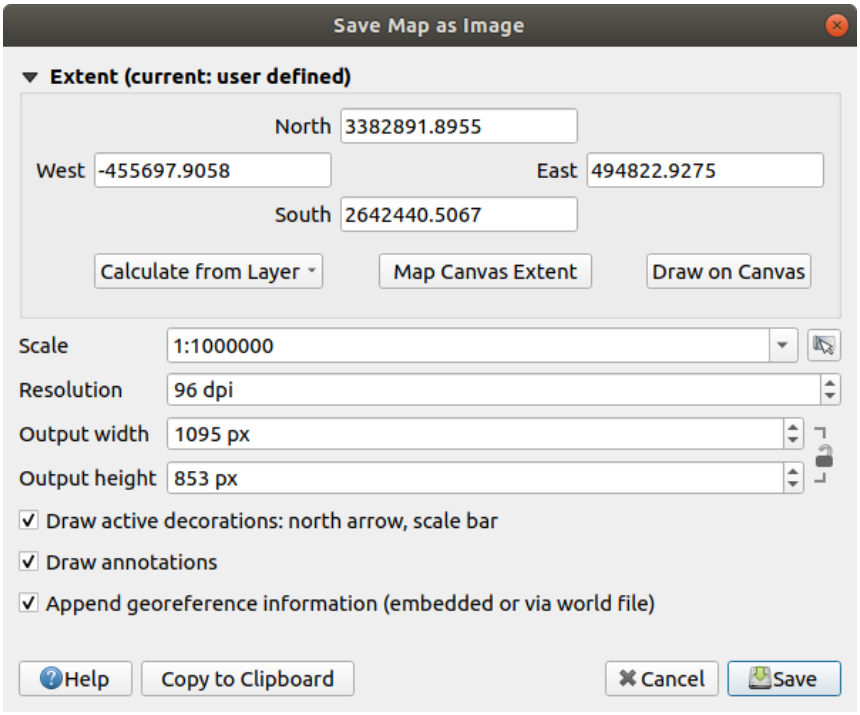



그림 7.19: 맵을 이미지로 저장 대화창

6. *Draw annotations*: 이 옵션을 활성화하면 어떤주석 도 내보낼 수 있습니다.

7. *Append georeference information (embedded or via world file)*: 산출물 포맷에 따라, 사용자 이미지가 저장되는 폴더에 이미지와 동일한 기반 이름을 가진 월드 파일 (`PNG` 이미지의 경우 확장자 `PNGW`, `JPG` 이미지의 경우 확장자 `JPGW`, …) 을 저장합니다. `PDF` 포맷의 경우 PDF 파일에 정보를 내장시킵니다.

8. PDF 로 내보낼 경우, *Save map as PDF*…대화창에서 더 많은 옵션을 설정할 수 있습니다:

   - *Export RDF metadata*: 문서의 제목, 작성자, 날짜, 설명 등등을 RDF 메타데이터로 내보냅니다.
   - *Create Geospatial PDF*: Generate a georeferenced PDF file. You can:
     - *Format*: 지리공간 PDF(Geospatial PDF) 포맷을 선택합니다.
     - *Include vector feature information*: 산출물 지리공간 PDF 파일에 있는 맵 안에 가시화된 피처들의 모든 도형 및 속성 정보를 지리공간 PDF 파일에 내장시킬 것입니다.

     > **참고**
     >
     > 지리공간 PDF 파일을 데이터소스로 이용할 수도 있습니다. QGIS 의 지리공간 PDF 지원에 대해 더 자세한 정보를 알고 싶다면 https://north-road.com/2019/09/03/qgis-3-10-loves-geospatialpdf/ 페이지를 참조하세요.

   - *Rasterize map*: 맵을 래스터화합니다.
   - *Simplify geometries to reduce output file size*: 맵을 내보내는 도중, 내보내기 해상도에서 눈에 띄지 않는 불필요한 꼭짓점들을 제거해서 도형을 단순화시킬 것입니다. (예를 들어 내보내기 해상도가 `300 dpi` 인 경우 서로의 거리가 `1/600 inch` 미만인 꼭짓점을 제거할 것입니다.) 이렇게 하면 내보내기 파일의

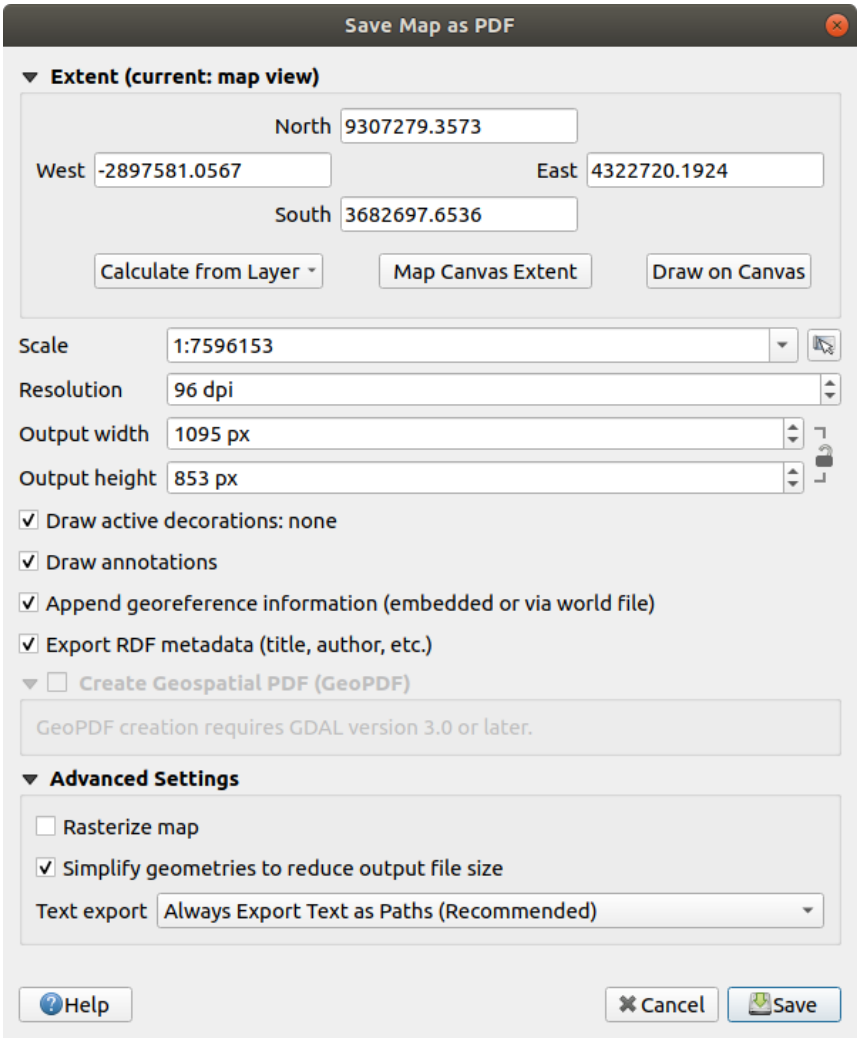



그림 7.20: 맵을 PDF 로 저장 대화창

용량 및 복잡도를 줄일 수 있습니다. (다른 응용 프로그램에서 대용량 파일을 불러올 수 없을 수도 있기 때문입니다.)

- Set the *Text export*: controls whether text labels are always or preferably exported as *text or outline objects*.

9. *Save*: 파일의 위치, 이름, 포맷을 선택합니다.

   이미지로 내보내는 경우, *Copy to clipboard* 버튼을 클릭해서 대화창에서 설정한 대로의 예상 산출물을 클립보드로 복사한 다음 리브레오피스 나 GIMP 등 다른 응용 프로그램에 맵을 붙여넣기할 수도 있습니다.

## 7.2 3 차원 맵 뷰

QGIS 는 3 차원 맵 뷰를 통해 3 차원 가시화를 지원하고 있습니다. *View → 3D Map Views →* 메뉴를 통해 3 차원 맵 뷰를 생성, 관리하고 열 수 있습니다:

1. *New 3D Map View* 메뉴 항목을 클릭하면 새 3 차원 맵 뷰를 생성할 수 있습니다. 도킹 가능한 QGIS 플로팅 패널이 나타날 것입니다. (그림 7.21 을 참조하세요.) 새 3 차원 맵 뷰는 2 차원 주 맵 캔버스와 동일한 범위와 뷰를 가지고 있으며, 뷰를 3 차원으로 전환하기 위한 탐색 도구 집합을 제공합니다.
2. *Manage 3D Map Views* 메뉴 항목을 클릭하면 3 차원 맵 뷰 관리자로 들어갑니다. 여기에서 3 차원 맵 뷰를 열고, 복제하고, 제거하고, 재명명할 수 있습니다.
3. 3 차원 맵 뷰를 하나 이상 생성했다면, *3D Map Views* 메뉴를 통해 그 목록이 표시됩니다. 이 메뉴를 클릭하면 목록을 켜고 끌 수 있습니다. 목록을 끈 상태라도, 프로젝트를 저장하면 목록도 저장될 것입니다.

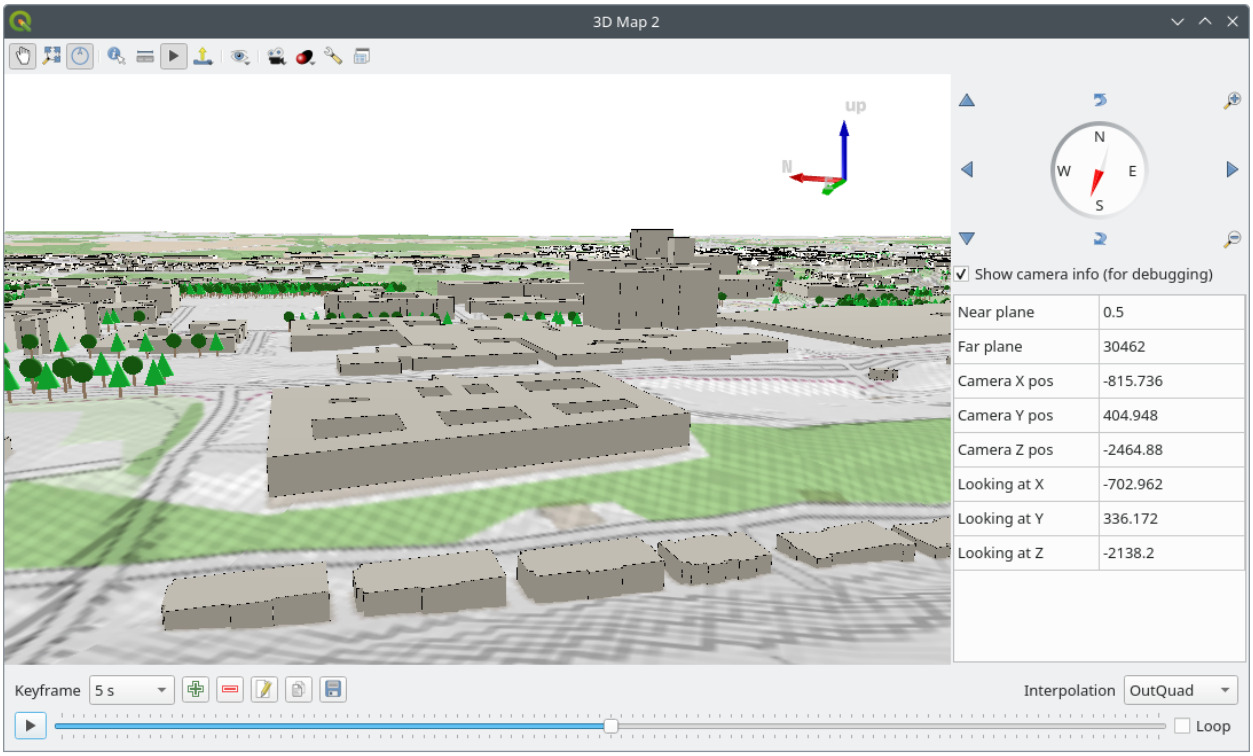



그림 7.21: 3D 맵 뷰 대화창

3 차원 맵 뷰 패널의 최상단에 다음과 같은 도구들이 있습니다:

- Camera Control: 카메라의 각도와 방향을 동일하게 유지하면서 뷰를 이동합니다.
- Zoom Full: 뷰를 전체 레이어 범위로, 또는참조 범위 를 설정한 경우 참조 범위로 확대/축소합니다.
- Toggle On-Screen Notification: (맵 뷰를 더 쉽게 조정하기 위한) 탐색 위젯을 켜고 끕니다.
- Identify: 지형에서 클릭한 포인트 또는 클릭한 3 차원 피처 (들) 에 대한 정보를 반환합니다. 더 자세한 정보는피처 식별 를 참조하세요.
- Measurement Line: measures the horizontal, vertical, and total 3D distance between points
  - Click to add points
  - Right-click to finish the measurement
  - Right-clicking twice or pressing *New* will reset the table
- Animations: 애니메이션 플레이어 위젯을 켜고 끕니다.
- Export 메뉴에서 신 (scene) 을 여러 포맷으로 내보낼 수 있습니다:
  - Save as Image…: 현재 뷰를 이미지 파일 포맷으로 내보냅니다.
  - Export 3D Scene: 현재 뷰를 (`.obj` 파일 형식의) 3 차원 신 (scene) 으로 내보내, 블렌더 등의 응용 프로그램에서 후처리할 수 있게 해줍니다. 지형 및 벡터 피처를 3 차원 객체로 내보냅니다. 레이어속성 및 맵 뷰환경 설정 을 무시하는 내보내기 설정에는 다음이 포함됩니다:
    - *Scene name* 및 *Folder*: 신 이름 및 대상 폴더
    - *Terrain resolution*: 지형 해상도
    - *Terrain texture resolution*: 지형 텍스처 해상도
    - *Model scale*: 모델 축척
    - *Smooth edges*: 경계 평탄화 옵션
    - *Export normals*: 법선 (□□) 내보내기 옵션
    - *Export textures*: 텍스처 내보내기 옵션
- Set View Theme: 미리 정의된맵 테마 에서 맵 뷰에 표시될 레이어 집합을 선택할 수 있도록 해줍니다.
- Camera 메뉴에서 2 차원 뷰와 3 차원 뷰 간의 관계를 제어할 수 있습니다:
  - *2D map view follows 3D camera* / *3D camera follows 2D Map view*: 각각 2 차원 맵 뷰가 3 차원 카메라를 따르도록 그리고/또는 3 차원 카메라가 2 차원 맵 뷰를 따르도록 뷰를 동기화합니다.
  - *Show visible camera area in 2D map view*: 2 차원 맵 뷰에 카메라의 가시 영역을 표시합니다.
  - *Set 3D scene on 2D map view*: 2 차원 맵 뷰 위에 그린 범위와 교차하는 지형 및 피처들만 표시하도록 3 차원 신을 잘라냅니다. 일반 환경설정 탭에서 더 많은 옵션을 사용할 수 있습니다.
  - Cross Section Tool creates a cross section in 3D scene by defining an area on the terrain in the 2D map canvas. Click to set the start point, move the mouse to define the line direction and click again to set the end point. You can adjust the size of the section by moving the mouse after selecting the start and end points. Press `Esc` or right-click to reset the selection.
  - Use Disable Cross Section to remove the cross section.

- Effects 는 3 차원 렌더링에음영 표시, 아이돔 *(Eye Dome)* 조명, 또는앰비언트 어클루전 *(ambient occlusion)* 과 같은 시각 효과를 추가합니다.
- The Configure··· button opens the dialog to configure the 3D map view *settings*.
- Dock 3D Map View: 도킹된 위젯에서 최상위 창으로 변환합니다.

### 7.2.1 신 (scene) 환경설정

The 3D map view opens with some default settings you can customize. To do so, press the Configure··· button at the top of the 3D canvas panel to open the *3D configuration* window.

3 차원 환경 설정 대화창에는 3 차원 신을 정밀 조정할 수 있는 다양한 옵션들이 있습니다:

#### 일반 (General) 그룹

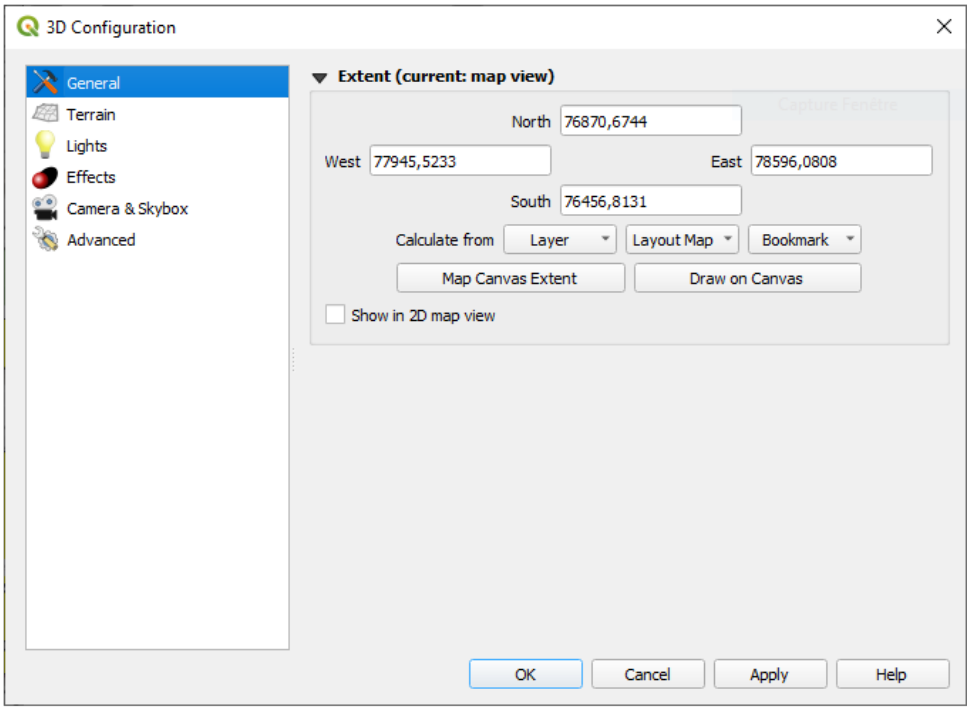



그림 7.22: 3 차원 맵 일반 환경설정 대화창

*General* 탭에서 다음을 설정할 수 있습니다:

- 공간 범위 선택기 를 이용해서 3 차원 신의 내용을 2 차원 맵 범위로 제한할 수 있습니다: 지형 및 피처들을 지정한 범위에 맞춰 잘라낸 다음 해당 범위 안에 들어오는 부분들만 신에 불러옵니다. 특히 지형의 수직 축척 설정을 표고차를 강조하기 위해 사용하는 경우, 카메라가 신의 내용 밑으로 배치되지 않도록 지형의 표고 범위, 메시 및 점구름 레이어들도 연산에 포함시킵니다.

  3 차원 맵 뷰에서 Zoom full 버튼을 클릭할 때, 설정 범위를 참조 범위로도 이용합니다.

- 주 맵 캔버스에 3 차원 신의 현재 범위에 대응하는 고무줄을 표시하고 싶다면 *Show in 2D map view* 옵션을 체크하면 됩니다.

### 지형

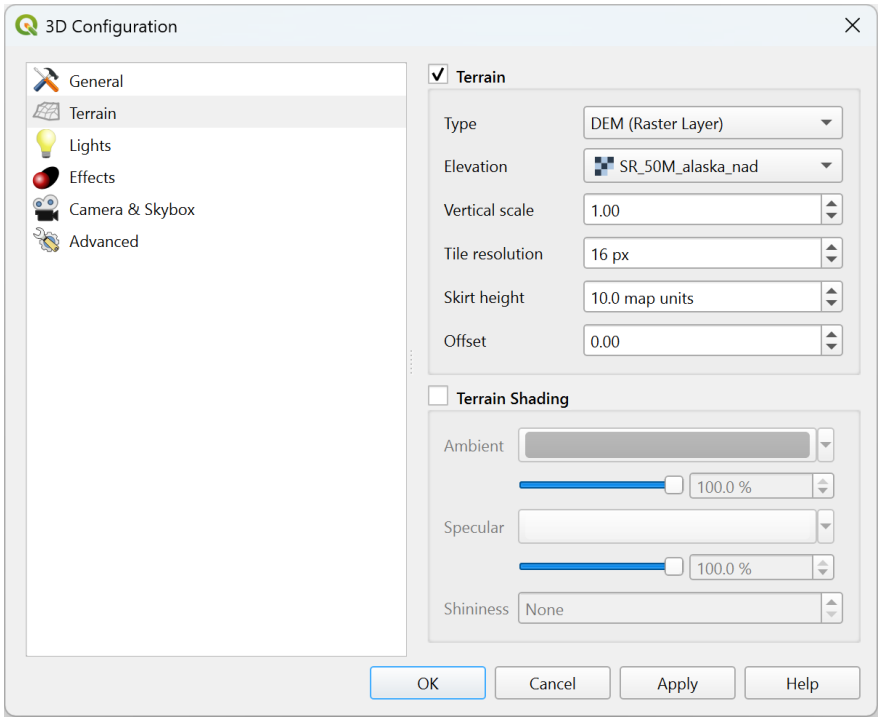



그림 7.23: 3 차원 맵 지형 환경설정 대화창

- *Terrain*: 자세한 내용으로 들어가기 전에, 3 차원 뷰에서의 지형은 지형 타일들의 위계로 표현되기 때문에 카메라가 지형에 가까이 갈수록 세부 사항이 충분하지 않은 기존 타일은 더 자세한 정보를 가진 더 작은 타일로 대체된다는 사실을 기억해야 합니다. 각 타일은 표고 래스터 레이어에서 추출한 메시 도형과 2 차원 맵 레이어에서 추출한 텍스처로 이루어져 있습니다.
  - *Type*: 표고 지형의 유형을 다음 유형들 가운데 하나로 선택할 수 있습니다:
    - *Flat terrain*: 평면 지형
    - *DEM (Raster Layer)*: DEM 을 불러옵니다.
    - *Online* 서비스: Mapzen 도구로 생성한 표고 타일 을 불러올 수 있습니다. 자세한 내용은 https://registry.opendata.aws/terrain-tiles/ 를 참조하세요.
    - *Mesh*: 메시 데이터셋을 불러옵니다.
    - *Quantized Mesh*: 프로젝트에서 양자화 (유한 집합에 대량의 입력값을 매핑하는 과정) 된 메시 레이어를 불러옵니다.
  - *Elevation*: 지형을 생성하기 위해 사용되는 래스터 또는 메시 레이어를 지정합니다. 래스터 레이어는 표고를 나타내는 밴드를 담고 있어야 합니다. 메시 레이어의 경우, 꼭짓점의 Z 값을 사용합니다.
  - *Vertical scale*: 수직축 용 축척 인자를 설정합니다. 축척을 증가시키면 지형의 높이를 과장하게 될 것입니다.
  - *Tile resolution*: 각 타일에 사용하기 위해 지형 래스터 레이어에서 얼마나 많은 샘플을 추출할지 설정합니다. 16px 이란 값은 각 타일의 도형이 16x16 픽셀의 표고 샘플로 이루어질 것이라는 의미입니다. 해상도 값이 높을수록 더 자세한 지형 타일을 생성하지만, 그에 비례해서 렌더링 복잡도도 더 높아집니다.

  - *Skirt height*: 가끔 지형 타일들 사이에서 작은 틈이 보일 수도 있습니다. 이 값을 높이면 지형 타일 주위로 수직 장벽 ("스커트") 을 추가해서 틈을 숨길 것입니다.
  - *Offset*: 지형을 위아래로 이동시킵니다. 예를 들어 신에 있는 다른 객체들의 지반 (ground level) 을 기준으로 지형 표고를 조정할 수 있습니다.

    지형의 높이와 (예를 들어 상대 수직 높이만 이용하는 점구름 (point cloud) 같은) 사용자 신에 있는 레이어의 높이가 불일치하는 경우 이 옵션이 유용할 수 있습니다. 이런 경우 지형 표고를 사용자 신에 있는 객체들의 표고와 일치하도록 조정하면 탐색 경험이 향상될 것입니다.

- 메시 레이어를 지형으로 사용하는 경우, (와이어프레임 표시, 삼각형 평활화, 상세 사항 수준 등을 설정하는) *Triangles settings* 및 *Rendering colors settings* 를 (렌더링 색상을 균일하게 할지 또는색상표 기준 으로 할지를) 환경 설정할 수 있습니다. 메시 레이어 *3* 차원 속성 에서 자세한 내용을 설명하고 있습니다.
- ☐ *Terrain shading*: 사용자가 지형 렌더링 방법을 선택할 수 있습니다.
  - 셰이딩 (shading) 비활성화 - 맵 텍스처만을 기준으로 지형 색상을 결정합니다.
  - 셰이딩 활성화 - 퐁 셰이딩 모델 (Phong's shading model) 을 이용, 맵 텍스처, 지형 법선 벡터 (normal vector), 신 조명 (들) 및 지형 소재 (terrain material) 의 *Ambient* 및 *Specular* 색상과 *Shininess* 를 계산해서 지형 색상을 결정합니다.

### 조명

*Lights* 탭에서 ✚ 메뉴를 클릭하면

- *Point lights* 를 최대 8 개 추가할 수 있습니다: 점조명 (point light) 은 마치 광구 (☐☐) 처럼 모든 방향으로 빛을 발해 일정 영역을 빛으로 채웁니다. 객체가 빛에 가까울수록 밝게 보이고, 객체가 빛에서 멀어질수록 어둡게 보입니다. 각 점조명의 위치 (*X*, *Y* 및 *Z*), 색상 (*Color*), 광도 (*Intensity*) 및 감쇠율 (*Attenuation*) 을 설정할 수 있습니다.
- *Directional lights* 를 최대 4 개 추가할 수 있습니다: 지향조명 (directional light) 은 사용자 객체에서 아주 멀리 떨어진 거대한 손전등과 비슷한 조명을 모방합니다. 마치 태양처럼 언제나 초점이 맞춰진, 절대 감쇠하지 않는 조명이죠. 지향조명은 단일 방향으로 평행한 광선을 발하지만 그 빛은 영원히 나아갑니다. 각 지향조명의 방향 (*Azimuth*), 고도 (*Altitude*), 색상 (*Color*) 및 광도 (*Intensity*) 를 설정할 수 있습니다.

### 시각 효과

- ☐ *Show shadows* 옵션을 체크하면 사용자 신에 음영을 표시하는데, 다음을 지정해야 합니다:
  - *Directional light*: 지향 조명의 방향을 지정합니다.
  - *Shadow rendering maximum distance*: 너무 멀리 있는 객체의 음영까지 렌더링하는 일을 피하려면, 특히 카메라가 지평선을 따라 위를 바라보고 있는 경우 최대 거리를 지정해야 합니다.
  - *Shadow bias*: 맵 크기의 차이 때문에 일부 영역이 다른 영역보다 더 어둡게 보이는 자체음영 (self-shadow) 효과를 피하려면 음영 편향을 설정해야 합니다. 값이 낮을수록 좋습니다.
  - *Shadow map resolution*: 음영을 더 선명하게 만듭니다. 해상도 파라미터를 너무 높게 설정하면 성능에 악영향을 미칠 수도 있습니다.
- ☐ *Show Eye Dome Lighting* (EDL): 심도 인식 (depth perception) 을 향상시켜주는 후처리 효과입니다. 각 픽셀의 심도 (카메라로부터의 거리) 를 이웃 픽셀들의 심도와 비교한 다음 해당 심도의 차이에 따라 강조시킵니다. 경계를 돋보이게 만드는 효과가 있습니다. 신 전체에 영향을 끼치며, 화면 공간 앰비언트 어클루전 *(Screen Space Ambient Occlusion)* 과 함께 사용할 수 있습니다. 다음 파라미터들을 제어할 수 있습니다:
  - *Lighting strength*: 심도를 보다 더 인식할 수 있도록 대조를 증가시킵니다.

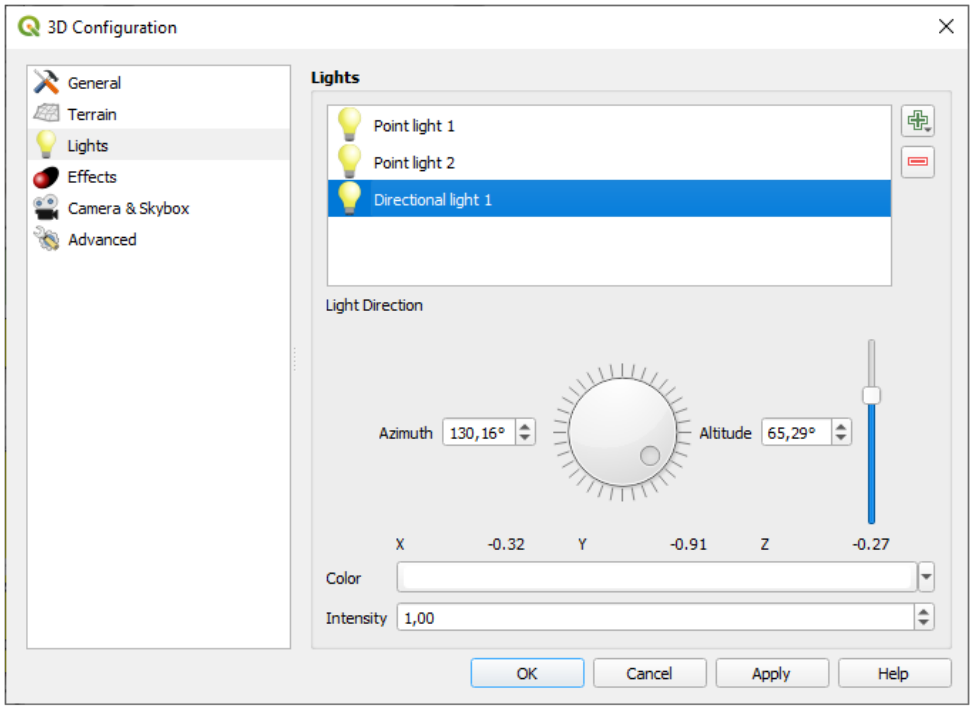



그림 7.24: 3 차원 맵 조명 환경 설정 대화창

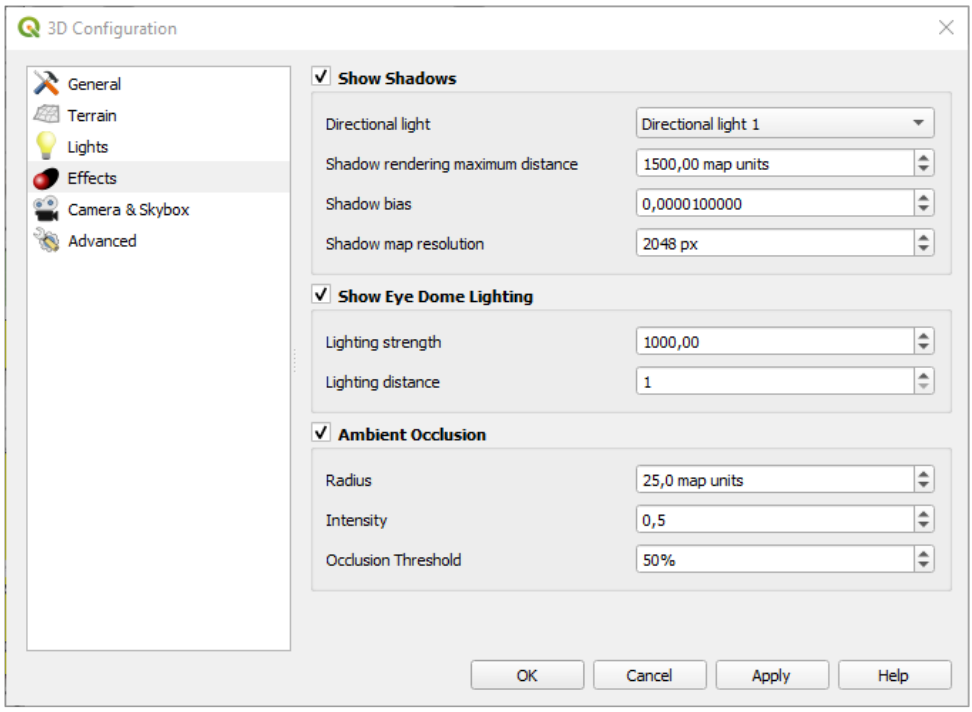



그림 7.25: 3 차원 맵 시각 효과 환경설정 대화창

  - *Lighting distance*: 사용된 픽셀이 중심 픽셀로부터 벗어난 거리를 나타내어 경계를 더 진하게 보이도록 합니다.

- *Ambient Occlusion* (SSAO): 화면 공간 앰비언트 어클루전도 앰비언트 조명에 보다 덜 노출된 영역에 더 어두운 음영을 적용해서 심도 인식을 향상시키는 후처리 효과입니다. 신 전체에 영향을 끼치며, 아이돔 조명과 함께 사용할 수 있습니다. 다음 파라미터들을 제어할 수 있습니다:
  - *Radius*: 앰비언트 어클루전을 얼마나 멀리까지 계산할지 지정합니다.
  - *Intensity*: 효과를 얼마나 강력하게 적용할지 (값이 높을수록 더 어두워집니다) 지정합니다.
  - *Occlusion threshold*: 효과를 나타내기 위해 이웃 포인트들을 얼마나 많이 어둡게 해야 할지 지정합니다. (값이 50% 미만일 경우 산출물이 더 어두워질 것이지만, 어클루전 범위가 더 늘어날 수도 있습니다.)

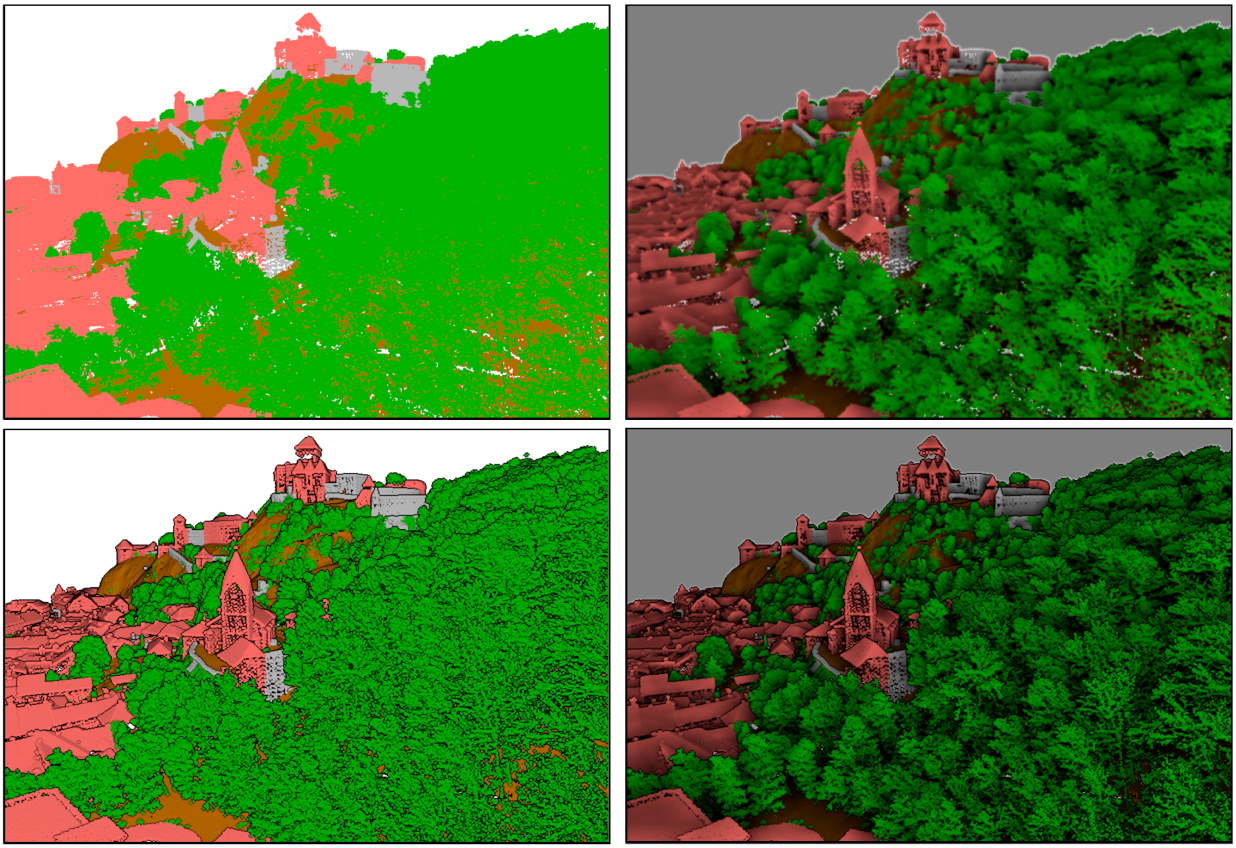


그림 7.26: 3 차원 맵에서 아이돔 조명 (EDL) 그리고/또는 화면 공간 앰비언트 어클루전 (SSAO) 을 이용한 점구름 렌더링

위에서부터 왼쪽에서 오른쪽으로: 적용 효과 없음–SSAO 만–EDL 만–SSAO 와 EDL 둘 다

### 카메라와 스카이박스

이 탭에서는 카메라, 3 차원 축, 탐색 동기화 및 스카이박스 같은 서로 다른 파라미터를 제어할 수 있습니다.

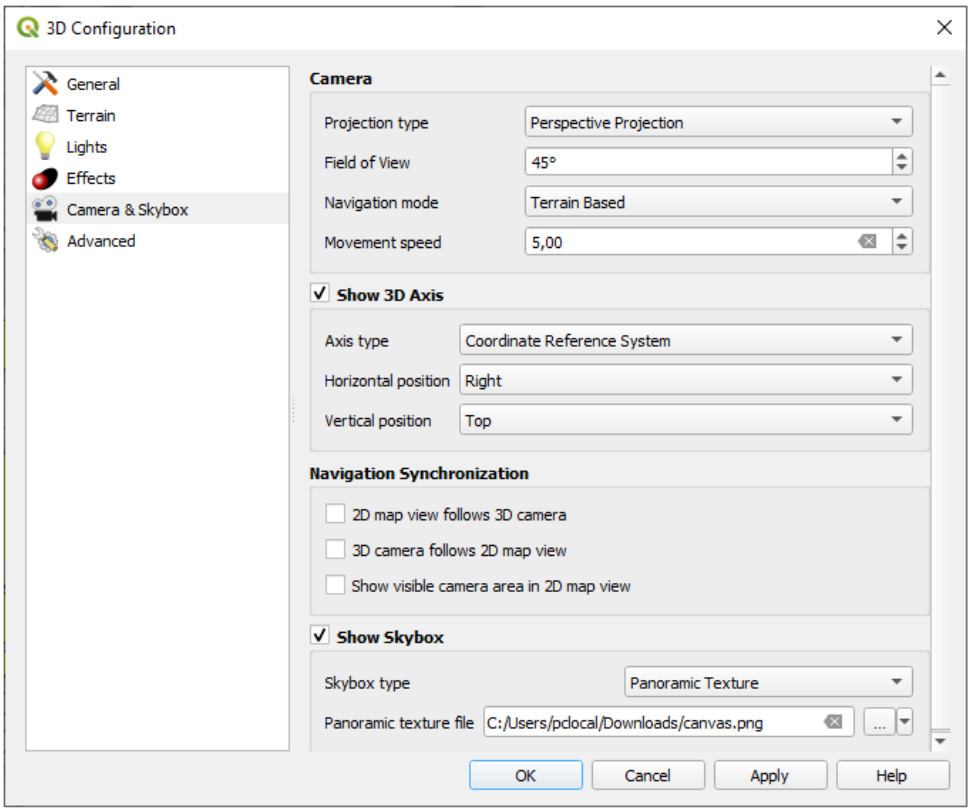
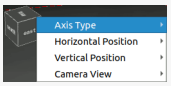



그림 7.27: 3 차원 맵 카메라 환경 설정 대화창

- *Camera* 파라미터 그룹은 *Settings* → *Options* → *3D* 대화창에서 정의한기본 카메라 설정 가운데 일부를 무시합니다.
- *Show 3D Axis*: 3 차원 축 도구를 활성화하려면 이 옵션을 체크하십시오. 이 파라미터 그룹은 축 유형 및 축 위치를 설정할 수 있게 해줍니다.
  - 축 유형을 *Coordinate Reference System* 유형으로 설정하면 직교 축 (orthogonal axis) 을 표현할 것입니다.
  - 축 유형을 *Cube* 유형으로 설정하면 3 차원 정육면체를 표현할 것입니다. 정육면체를 이용하면 카메라 뷰를 변경할 수 있습니다. 예를 들어, *north* 면을 클릭하면 카메라가 북쪽에서 바라보도록 설정됩니다.

> **팁**
>
> 3 차원 축의 위치 및 유형, 그리고 카메라 뷰를 빠르게 설정하려면 3 차원 축을 오른쪽 클릭하면 됩니다.
>
> 
> 
>
> 그림 7.28: 3 차원 축 컨텍스트 메뉴

- *Navigation Synchronization* 파라미터 그룹은 2 차원 뷰가 3 차원 카메라 위치를 따르도록 또는 3 차원 카메라 위치가 2 차원 뷰를 따르도록 하는 또는 쌍방향으로 동기화하는 옵션을 제공합니다. 마지막 옵션을 선택하면

2 차원 맵 뷰 위에 3 차원 카메라의 시야 범위를 표시합니다.

- *Show skybox*: 신에 스카이박스 렌더링을 활성화하려면 이 옵션을 체크하십시오. 스카이박스 유형은 다음 가운데 하나로 설정할 수 있습니다:
  - *Panoramic texture*: 하늘 풍경을 360° 제공하는 단일 파일입니다.
  - *Distinct faces*: 신을 담고 있는 육면체의 각 면을 각각의 텍스처 파일로 제공합니다.

  스카이박스의 텍스처 이미지 파일은 디스크에 저장된 파일일 수도, 원격 URL 일 수도, 또는 프로젝트에 내장되어 있을 수도 있습니다. (프로젝트 내장 파일에 대한 더 자세한 내용은내장 파일 선택기 를 참조하세요.)

### 고급

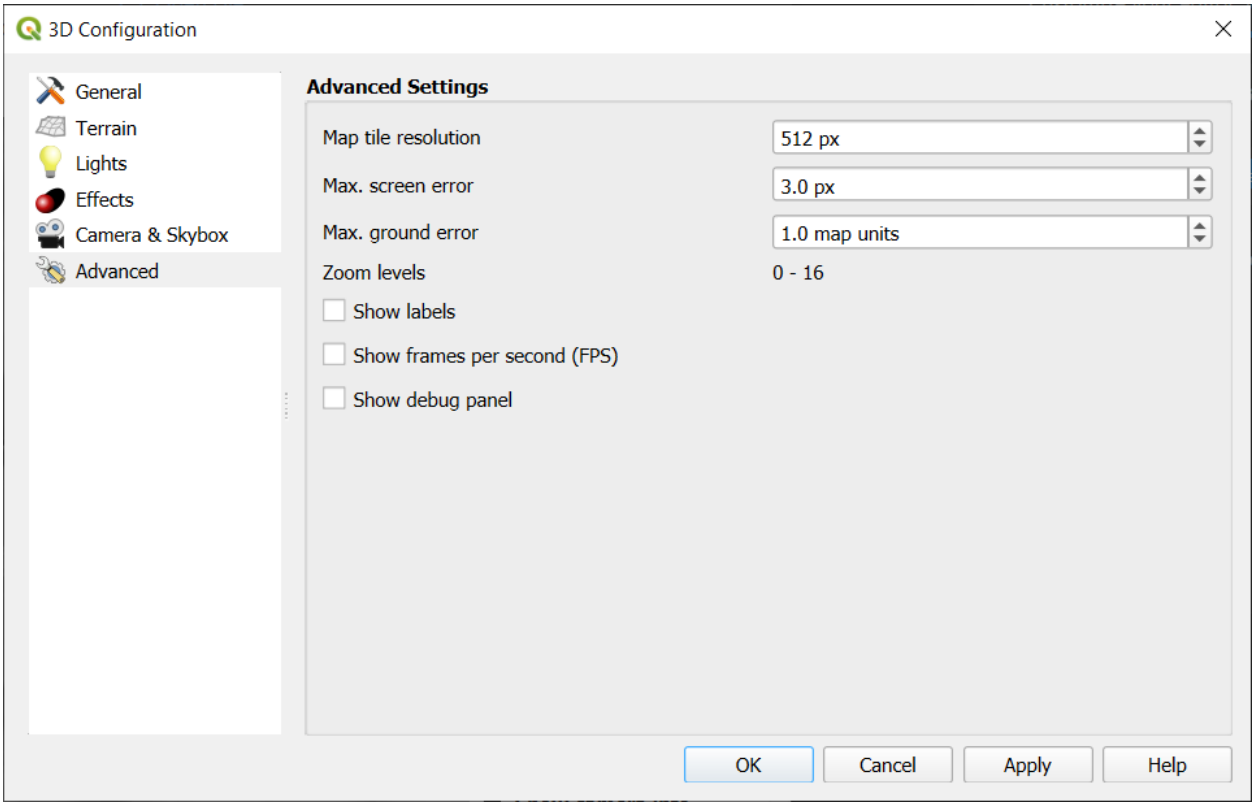



그림 7.29: 3 차원 맵 고급 환경설정 대화창

- *Map tile resolution*: 2 차원 맵 이미지의 너비와 높이를 지형 타일의 텍스처로 이용합니다. 256px 이란 각 타일을 256x256 픽셀 크기의 이미지로 렌더링할 것이라는 의미입니다. 픽셀값이 높을수록 더 자세한 지형 타일을 생성하지만, 그에 비례해서 렌더링 복잡도도 더 높아집니다.
- *Max. screen error*: 지형 타일을 더 자세한 타일로 (그리고 그 반대로) 교체하는 역치 (threshold) 를 결정합니다. 즉 3 차원 뷰가 얼마나 빨리 더 상세한 타일을 사용할 것인가를 결정하는 것이죠. 이 값이 낮을수록 신의 세부 사항이 더 자세해지지만, 그에 반비례해서 렌더링 복잡도는 더 높아집니다.
- *Max. ground error*: 지형 타일이 더 상세한 타일로 더이상 나눠지지 않는 해상도를 결정합니다. (타일을 나눠봤자 더 자세한 사항이 나오지도 않겠지요.) 이 값은 타일 위계의 심도 (depth) 를 제한합니다. 이 값이 낮을수록 위계가 늘어나지만, 그에 반비례해서 렌더링 복잡도는 더 높아집니다.

- *Zoom levels*: 확대/축소 단계의 개수를 보여줍니다. (*Map tile resolution* 과 *Max. ground error* 값에 따라 달라집니다.)
- ☐ *Show labels*: 맵 라벨을 켜고 끕니다.
- ☐ *Show frames per second (FPS)*: 초당 프레임 수 (frames per second) 를 표시합니다.
- ☐ *Show debug panel*: allows to show or hide a panel next to the 3D map view displaying options to better control map navigation and debug information:
  - ☐ *Show map tile info*: 지형 타일에 대한 경계 (border) 및 타일 개수를 포함하는 정보를 켜고 끕니다. (지형에 문제가 있을 때 파악하고 해결하는 데 유용합니다.)
  - ☐ *Show bounding boxes*: 지형 타일의 3D 경계 상자 (bounding box) 를 켜고 끕니다. (지형에 문제가 있을 때 파악하고 해결하는 데 유용합니다.)
  - ☐ *Show camera's view center* : 카메라 뷰의 중앙 표시를 켜고 끕니다.
  - ☐ *Show camera's rotation center*: 카메라의 회전 중심 표시를 켜고 끕니다.
  - ☐ *Show light sources*: 광원 위치에 구체를 표시해서 신 내용에 따라 광원을 더 쉽게 배치/재배치할 수 있습니다.
  - ☐ *Stop scene updates*
  - ☐ *Show debug overlay*: visual overlay which displays some useful debugging and profiling information. This allows in particular to quickly see the frame graph and the scene graph
  - ☐ *Debug Shadow Map*: (문제점 파악을 위해) 음영을 드리우는 광원의 시점에서 신을 적-흑 이미지로 렌더링합니다. 위젯은 3 차원 맵 뷰에 비례하는 *Size* 로 설정되며, *Corner* 에 도킹됩니다.
  - ☐ *Debug Depth Map*: (문제점 파악을 위해) 신의 심도 맵을 픽셀이 가까울수록 더 어두워지는 이미지로 렌더링합니다. 위젯은 3 차원 맵 뷰에 비례하는 *Size* 로 설정되며, *Corner* 에 도킹됩니다.
  - *Show camera info*:
    - ☐ *Far plane*: controls how far from the camera rendering stops
    - ☐ *Near plane*: controls how close to the camera rendering starts
    - ☐ *Camera X/Y/Z pos*: sets the camera's position in 3D space
    - ☐ *Looking at X/Y/Z*: sets the target point the camera is looking at

> **참고**
>
> When your 3D map view is open in a standalone window, you can use `Ctrl+Shift+d` keyboard shortcut to access the debug panel.

### 7.2.2 탐색 옵션

맵 뷰를 3 차원으로 탐색하려면:

- 지형 기울이기 (창의 중앙을 가로지르는 수평 축을 기준으로 회전)
  - Tilt up 및 Tilt down 도구를 클릭하십시오.
  - `Shift` 키를 누른 채 위쪽/아래쪽 화살표 키를 사용하십시오.
  - 마우스의 가운데 버튼을 누른 채 마우스를 앞쪽/뒤쪽으로 드래그하십시오.
  - `Shift` 키와 마우스 왼쪽 버튼을 함께 누른 채 마우스를 앞쪽/뒤쪽으로 드래그하십시오.
- 지형 회전하기 (창의 중앙을 가로지르는 수직 축을 기준으로 회전)
  - 탐색 위젯의 나침반을 보고자 하는 시선의 방향으로 돌립니다.
  - `Shift` 키를 누른 채 왼쪽/오른쪽 화살표 키를 사용하십시오.
  - 마우스의 가운데 버튼을 누른 채 마우스를 왼쪽/오른쪽으로 드래그하십시오.
  - `Shift` 키와 마우스 왼쪽 버튼을 함께 누른 채 마우스를 왼쪽/오른쪽으로 드래그하십시오.
- 수평면 기준으로 카메라 위치 (및 뷰 중앙) 변경하기
  - Camera control 버튼을 활성화한 상태에서 마우스 왼쪽 버튼을 누른 채 마우스를 드래그합니다.
  - 탐색 위젯의 방향 화살표를 클릭합니다.
  - 위쪽/아래쪽/왼쪽/오른쪽 화살표 키를 사용해서 카메라를 각각 앞쪽, 뒤쪽, 왼쪽, 오른쪽으로 이동합니다.
- 카메라 고도 변경하기: `Page Up`/`Page Down` 키를 누릅니다.
- 카메라 방향 변경하기 (카메라의 위치는 변하지 않지만 뷰 중앙 포인트가 이동)
  - `Ctrl` 키를 누른 채 화살표 키를 사용하면 카메라 방향을 위쪽, 아래쪽, 왼쪽, 오른쪽으로 돌립니다.
  - `Ctrl` 키와 마우스 왼쪽 버튼을 함께 누른 채 마우스를 드래그하십시오.
- 확대/축소
  - 탐색 위젯의 Zoom In 및 Zoom Out 도구를 클릭합니다.
  - 마우스 휠을 돌립니다. (`Ctrl` 키를 누른 채 돌리면 더 정밀하게 확대/축소할 수 있습니다.)
  - 마우스 오른쪽 버튼을 누른 채 드래그하면 확대 (아래쪽) 하거나 축소 (위쪽) 할 수 있습니다.

카메라 뷰를 초기화하려면 3 차원 맵 뷰 패널 최상단에 있는 Zoom Full 버튼을 클릭하십시오.

### 7.2.3 애니메이션 생성하기

애니메이션은 특정 시간대 카메라 위치의 키프레임 모음을 기반으로 합니다. 애니메이션을 생성하려면:

1. Animations 도구를 켜면 애니메이션 플레이어 위젯이 나타납니다.
2. Add keyframe 버튼을 클릭하고 *Keyframe time* 을 초 단위로 입력하면, *Keyframe* 콤보박스가 타임셋 (time set) 을 표시합니다.
3. 탐색 도구를 이용해서 현재 키프레임 시간과 관련된 위치로 카메라를 이동합니다.
4. 이전 단계를 반복해서 필요한 만큼 많은 (시간과 위치를 가진) 키프레임을 추가합니다.

5. 애니메이션을 미리보기하려면 ▶ 버튼을 클릭하십시오. QGIS 가 타임셋에 맞춰 카메라 위치/회전을 사용해 이를 각 키프레임 사이에 삽입해서 신 (scene) 을 생성할 것입니다. 애니메이션을 위한 다양한 *Interpolation* 모드를 사용할 수 있습니다. (linear, inQuad, outQuad, inCirc 등등…더 자세한 내용은 https://doc.qt.io/archives/qt-5.15/qeasingcurve.html#EasingFunction-typedef 를 참조하세요.)

   시간 슬라이드 바를 움직여서 애니메이션을 미리보기할 수도 있습니다. *Loop* 체크박스를 체크하면 애니메이션을 반복 실행할 것입니다. 이때 ▶ 버튼을 누르면 애니메이션을 정지합니다.

Export animation frames 버튼을 클릭하면 신을 나타내는 이미지 모음을 생성합니다. *Template* 에서 파일명을, *Output directory* 에서 저장할 디렉터리를 지정할 수 있을 뿐만 아니라, *Frames per second* 에서 초당 프레임 개수를, *Output width* 와 *Output height* 에서 산출 이미지의 너비와 높이도 설정할 수 있습니다.

### 7.2.4 3 차원 벡터 레이어

벡터 레이어 속성 패널의 *3D View* 부분에 있는 *Enable 3D Renderer* 를 활성화하면 3 차원 맵 뷰에 표고값을 가진 벡터 레이어를 표시할 수 있습니다. 3 차원 벡터 레이어의 렌더링을 조정할 수 있는 몇 가지 옵션을 설정할 수 있습니다.

## 7.3 표고 단면 뷰

*Elevation Profile* 패널은 라인을 따라 표고 데이터를 가시화하기 위한, 사이드 뷰 용 그리기 (plotting) 도구입니다. 이 도구는 벡터, 래스터, 메시 및 점구름 레이어를 지원하며, 2 차원 또는 3 차원 유형 데이터를 지원할 수 있습니다.

To add an elevation profile view, go to *View* *Elevation Profile* menu. You can add as many profile views as you want, and they can be docked, piled on top of each other, or floating.

### 7.3.1 인터페이스

*Elevation Profile* 패널 최상단에 있는 툴바에 다음과 같은 도구들이 있습니다:

표 7.5: 표고 단면 뷰 툴바

| 도구 | 단축키 | 설명 |
|---|---|---|
| Add Layers | `Ctrl` 키 누른 채 드래그 & 드롭 | 도표에 새 레이어를 추가할 수 있게 해줍니다. |
| Show Layer Tree | | 단면 뷰에서의 렌더링을 환경설정하기 위한 프로젝트 레이어들의 목록을 표시하거나 숨깁니다. |
| Capture Curve | | 단면 곡선을 표현하기 위해 맵 캔버스 위에 라인을 대화방식으로 (interactively) 그립니다. |
| Capture Curve From Feature | | 맵 캔버스에 있는 기존 라인 피처를 선택해서 해당 라인을 따라 단면 곡선을 생성합니다. |
| Nudge Left | `Ctrl+Alt+,` | 캡처한 라인을 맵 상에서 왼쪽으로 천천히 (예를 들어 표고를 기반으로 하는 최적의 단면선을 찾기 위해) 이동시킬 수 있습니다. |

다음 페이지에 계속

표 7.5 – 이전 페이지에서 계속

| 도구 | 단축키 | 설명 |
|---|---|---|
| Nudge Right | `Ctrl+Alt+.` | 캡처한 라인을 맵 상에서 오른쪽으로 천천히 (예를 들어 표고를 기반으로 하는 최적의 단면선을 찾기 위해) 이동시킬 수 있습니다. |
| Clear | | *Elevation Profile* 뷰에서 단면선 및 모든 표시된 도표 (plot) 를 제거합니다. |
| Identify Features | | 도표 (plot) 캔버스에 있는 피처를 단일 클릭 또는 클릭 & 드래그로 직사각형을 그리는 방법 가운데 하나를 통해 식별합니다. 식별 결과는 표준식별 결과 도킹 창에 표시됩니다. |
| Pan | `Space` | 클릭 & 드래그로 도표 캔버스를 이동 (pan) 시킵니다. 마우스 가운데 버튼을 누른 채 이동시킬 수도 있습니다. |
| Zoom X Axis | | 수직 비율을 유지하면서 수평 축을 따라 확대/축소합니다. |
| zoom | `Ctrl+Space` | 도표 캔버스 위를 클릭하거나 직사각형을 클릭 & 드래그해서 확대합니다. `Alt` 키를 누른 채 클릭하면 축소합니다. |
| Zoom Full | | *Elevation Profile* 뷰를 캡처 라인의 범위로 확대/축소합니다. |
| Enable Snapping | | 단면 뷰에 있는 도표 피처의 변 또는 꼭짓점에 스냅할 수 있게 해줍니다. 좌표 획득 또는 거리 측정을 정확하게 할 수 있게 해주는 편리한 도구입니다. |
| Measure Distances | | 수평 및 수직 거리를 측정합니다. |
| Export As PDF | | 도표를 PDF 로 (고품질 벡터 객체로) 내보냅니다. |
| Export As Image | | 도표를 몇 가지의 이미지 포맷으로 내보낼 수 있습니다. |
| Export Results | | Exports plots as 3D features, 2D Profile and Distance/Elevation Table. (The results can be saved as DXF files, CSV files or any of the standard writable vector spatial formats). More details at *Exporting from the elevation profile*. |
| Options | | 표고 단면선을 환경설정할 수 있습니다. |
| *Lock distance/elevation scales* | | 수평 및 수직 척도를 항상 동등하게 (따라서 예를 들면 단면 뷰에서 45° 경사가 45° 경사로 나타나도록) 유지합니다. |
| *Distance units* | | 단면 도표에서 거리를 맵 캔버스 단위가 아닌 다른 단위들로 렌더링할 수 있도록 해줍니다. |
| *Tolerance* | | Sets how far from the actual profile line a feature (vector point, line or polygon, point cloud) can reside to be included in the results. Uses the map units and is ignored by other layer or geometry types. For some layer types (e.g. points, lines, polygons), a custom tolerance can be set in the layer's Elevation properties tab, which replaces the global tolerance and allows more control over feature inclusion based on distance to the profile line. |
| *Rename Profile…* | | Allows to rename the profile view. |
| Dock Elevation Profile View | | 뷰를 도킹 상태 또는 플로트 상태로 변환합니다. |

좌하단에 있는 *Show Layer Tree* 버튼을 누르면 *Layers* 패널의 복사본을 표시할 수 있습니다. 하지만 이 복사본은 독립적인 위젯으로, 가시화 레이어 집합을 자신만의 사용자 지정 위계 순서 (stack order) 로 설정할 수 있습니다.

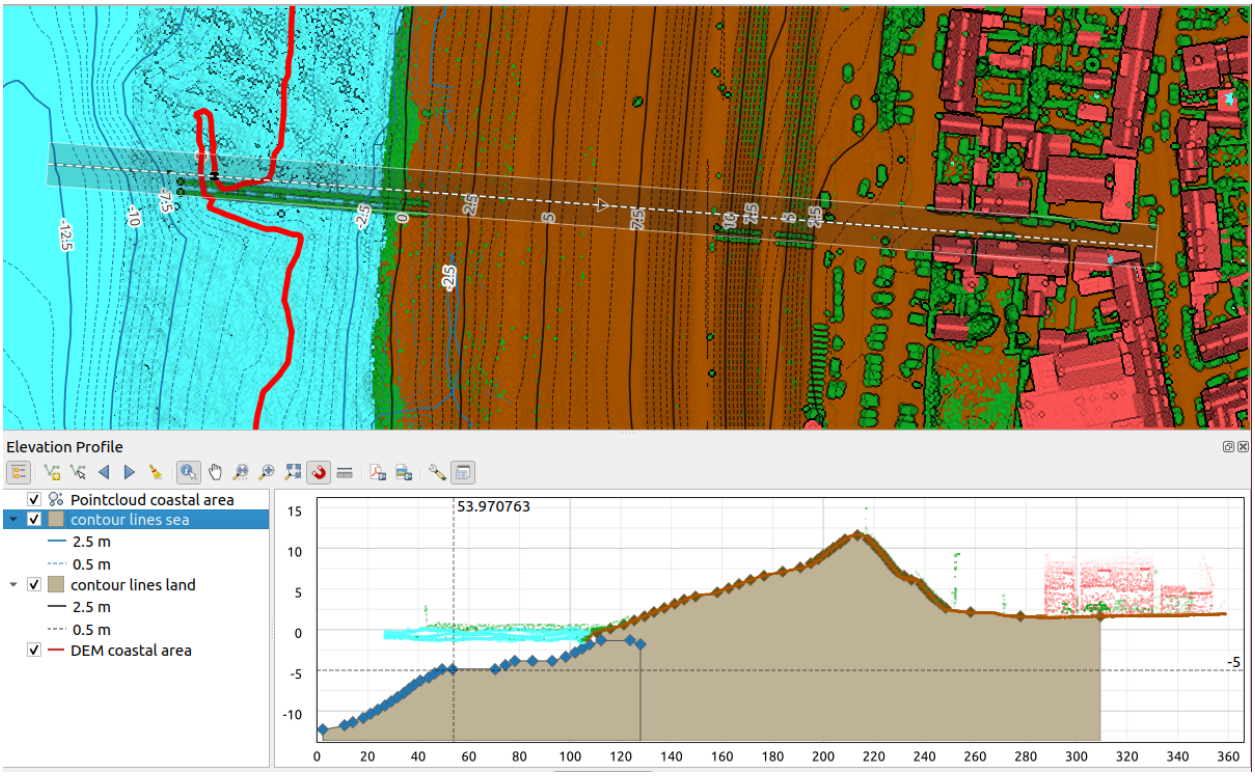



그림 7.30: 주 맵 캔버스 아래 내장된 표고 단면 대화창
이 그림은 방파제 뒤에 마을이 있는 해안선을 보여주고 있습니다. 표고 단면선 (elevation profile line) 에 더 큰 허용오차를 적용하면 더 많은 점구름 피처들을 반환합니다.

사용자는 이 패널을 통해 도표 캔버스 안에서 레이어 렌더링 및 레이어 습성을 제어할 수 있습니다:

- 도표 캔버스에 레이어를 렌더링할지 여부를 설정하려면 레이어 이름 옆에 있는 상자를 체크/체크 해제하십시오.
- 레이어 순서를 변경하려면 레이어를 위 또는 아래로 드래그 & 드롭하십시오.
- 단면 뷰에서의 레이어 스타일 렌더링: 레이어를 더블클릭하거나, 오른쪽 클릭한 다음 *Properties…*메뉴를 선택하면 레이어의 *Elevation* 속성 탭이 열려 표고 속성을 환경설정할 수 있습니다. 레이어 위에 마우스 커서를 가져가면 표고 설정들을 요약한 내용이 툴팁으로 표시됩니다.

레이어 트리 오른쪽에 있는 도표 캔버스 (plot canvas) 가 활성화된 레이어들의 표고 단면을 미리보기할 수 있는 메인 창입니다. 도표 캔버스는 수평 축이 단면선의 길이를 표시하고 수직 축이 관찰된 피처의 Z 표고를 표시하는 격자 그리드 기반입니다. 이 안에서 최상단의 도구들을 사용해서 확대/축소, 이동, 측정, 피처 식별 등등의 대화형 작업을 할 수도 있습니다.

### 7.3.2 표고 단면 생성하기

단면 (profile) 뷰를 생성하려면, 다음과 같이 하면 됩니다:

1. *View* →  *Elevation Profile* 메뉴를 선택하십시오. *Elevation profile* 패널이 열립니다.
2. 다음 그리기 도구 가운데 하나를 선택해서 이를 따라 지형과 피처를 렌더링할 단면선 (profile line) 을 생성하십시오:
   -  Capture Curve: 주 맵 캔버스를 왼쪽 클릭해서 꼭짓점들을 추가하고, 라인을 종료하려면 오른쪽 클릭하십시오. 해당 라인을 단면선으로 사용하게 될 것입니다.
   -  Capture Curve From Feature: 맵 캔버스에 있는 라인 피처를 클릭해서 단면선으로 선택하십시오. 클릭한 포인트에 여러 피처들이 존재하는 경우 해당 피처들 가운데 하나를 선택할 수 있게 해주는 팝업 메뉴가 나타날 것입니다.

   스냅 작업 옵션, 트레이스 작업, 디지타이즈 기법, 또는 고급디지타이즈 작업 패널 과 같은 모든 라인 디지타이즈 작업 케이퍼빌리티를 사용할 수 있습니다.

   이 단계에서 도표 캔버스가 몇몇 피처를 렌더링하기 시작할 수도 있습니다.
3. 다음 단계는 사용자가 가시화하고자 하는 레이어들의 표고 속성을 환경설정하는 것입니다.
   1.  Show Layer Tree 버튼을 클릭해서 레이어 목록을 보이게 하십시오. 기본적으로 단면 도구에 프로젝트의 모든 레이어를 불러와 참조시키지는 않습니다. 예를 들어 속성에 표고 데이터를 가지고 있다고 표시되지 않은 래스터 레이어는 무시합니다.  Add layers 버튼을 클릭하면 도표에 추가할 수 있지만 지금은 도표에 없는, 필터링된 레이어들의 목록이 열릴 것입니다. 선택한 레이어를 가진 대화창을 적용하면 자동으로 해당 레이어가 표고 데이터를 가지고 있다고 표시하고 즉시 도표에 추가할 것입니다.

      > **팁**
      >
      > `Ctrl` 키를 누른 채 드래그 & 드롭하는 방법으로 주 레이어 트리에서 표고 단면 도구의 레이어 트리로 레이어를 추가할 수 있습니다.

   2. 사용자가 관심이 있는 레이어들의 가시성을 켜고 끄십시오. 단면 뷰에는 가시화된 레이어들만 렌더링됩니다. 이때 주 *Layers* 패널에서 가시화/비가시화된 레이어와는 다른 레이어를 가시화/비가시화시킬 수 있습니다.

3. 레이어 이름을 더블클릭하거나, 오른쪽 클릭한 다음 *Properties* 메뉴를 선택하십시오. 해당 레이어의 *Elevation* 속성 탭이 열립니다. 이 탭에서 단면 뷰에 각 피처 또는 지형을 어떻게 렌더링해야 할지를 환경설정합니다. 환경설정할 수 있는 속성은 레이어 유형에 따라 달라집니다:

   - 래스터 표고 속성
   - 벡터 표고 속성
   - 점구름 표고 속성
   - 메시 표고 속성

   활성화된 레이어의 지형 또는 피처의 표고 속성을 환경설정하면, 단면 뷰는 곧바로 지정한 단면 곡선과 공간교차하는 지형 또는 피처를 렌더링하기 시작합니다.

4. Under Options drop-down menu, you can:

   - set the *Tolerance* value. This value is used to create a flat buffer around the elevation profile line, visible in the main map canvas. Any visible feature (point, line, polygon, etc.) overlapping that buffer will be captured in the plot canvas.
   - select the *Show Subsections Indicator* to display vertical lines at the distance of the vertices of the profile curve. This highlights the position of the vertices of the curve on the Elevation Profile.
   - select the *Subsections Symbology* to set the color, opacity, width and style of the Subsections Indicator lines.

> **참고**
>
> **Limitations with polygon extrusion**
>
> Geometry extrusion can be set in the *Elevation* properties of a layer, and rendered in the profile view. When tolerance is enabled, it is however not trivial to render extruded polygons, thus, for now, polygon extrusion is ignored.

### 7.3.3 대화형 단면 뷰 작업

표고 단면선을 생성하고 나면, 표고 캔버스가 단면선 전체 범위로 확대/축소됩니다. X 축 상에서는 단면의 길이를 볼 수 있고 Y 축 상에서는 캡처된 최저 높이와 최고 높이 사이의 높이 범위를 볼 수 있습니다. 이때 길이와 높이는 둘 다 맵 단위입니다.

표고 단면 뷰 안에서 마우스 커서를 움직일 때 십자로 교차하는 점선 2 개가 나타날 것입니다:

- 수직선은 높이 정보를 보여줍니다.
- 수평선은 표고 단면선의 시작점으로부터의 거리를 보여줍니다.

표고 단면 캔버스 안에서 마우스 커서를 움직일 때, 주 맵 캔버스 상의 표고 단면선을 따라 검은 점이 움직이는 것을 볼 수 있습니다. 그 방향을 나타내는 화살표가 단면선 가운데 표시됩니다.

주 맵 캔버스의 경우, QGIS 가 도표 캔버스 상을 탐색할 수 있는 수단을 제공합니다:

- Pan: 표고 단면 범위를 사용자가 원하는 어느 방향으로든 이동시킵니다. `Space` 키를 누른 채 마우스를 움직이면 도표 캔버스의 범위도 이동됩니다.
- Zoom X Axis: 수직 (표고) 축의 비율을 동일하게 유지한 채 수평 축을 따라 확대합니다. 왼쪽 클릭하면 클릭한 포인트를 축의 중심으로 X 축을 따라 도표를 늘립니다. 또는 직사각형을 드래그하면 X 축을 따라 도표를 직사각형의 너비로 늘립니다. `Alt` 키를 누른 채 Zoom X Axis 도구를 사용하면 X 축을 따라 축소합니다.

-  Zoom: (왼쪽 클릭을 사용하면) 특정 포인트로 확대하거나 (어떤 영역을 직사각형으로 드래그하면) 특정 영역으로 확대합니다. `Alt` 키를 누른 채  Zoom 도구를 사용하면 축소합니다. 이 도구를 `Ctrl` 키와 함께 사용하면 좀 더 부드럽게 확대 또는 축소할 수 있습니다.
-  Zoom Full: 단면선의 전체 범위를 반환된 모든 피처와 함께 보여주는, 첫 단계에서 사용된 기본 확대/축소 수준으로 돌아갑니다. 확대/축소 수준을 리셋하고자 할 때 사용하십시오.

도표 캔버스에 표시된 요소들과 대화형으로 작업할 수도 있습니다:

-  Enable snapping 버튼을 누르면 피처의 포인트, 꼭짓점 또는 변을 정확하게 획득할 수 있습니다. 이렇게 획득한 정보는 정확한 측정 또는 좌표 보고서에 쓰입니다.
-  Identify Features: 레이어 트리에서 가시화된 레이어의 피처를 식별합니다. 단면 뷰에 있는 피처 여러 개를 한번에 쿼리하려면 해당 피처들을 모두 포함하는 직사각형을 드래그하면 됩니다. 호환되는 (예: 벡터, 점구름) 포맷인 경우 해당 피처가 주 맵 캔버스에서 강조될 것입니다.
-  Measurement Distances: 두 포인트 사이의 수평 *Distance*, 그리고 *Elevation* 및 *Total distance* 를 맵 단위로 보고시키려면 도표 캔버스에서 포인트 2 개를 클릭하거나 또는 선택하십시오.
-  Nudge Left 및  Nudge right: 맵 캔버스에서 표고 단면선의 위치를 왼쪽 또는 오른쪽으로 이동시킵니다. 도표 캔버스가 단면선 버퍼와 중첩하는 피처 및 지형을 보여주도록 다시 그려질 것입니다. 표고 단면선은  Options 메뉴에 있는 *Tolerance* 값을 이용해서 옆으로 움직입니다.

> **경고**
>
> 현재 표고 단면 뷰 또는 프로젝트를 닫으면 프로젝트에서 뷰가 제거됩니다.

표고 단면 뷰에 대해 더 자세히 알고 싶다면, 나이얼 도슨 (Nyall Dawson) 의 QGIS elevation profile/cross section tool –a deep dive! 발표를 시청해보십시오.

### 7.3.4 Exporting from the elevation profile

The top toolbar of the *Elevation profile* dialog provides a variety of tools to export an elevation profile, allowing you to reuse the plots in another software:

-  Export As PDF: Exports plots to PDF (as high quality vector objects). Following options are requested for generating the output file:
  - The *Page size*, selecting a page from a predefined list or setting custom dimensions, and the page orientation
  - The *Chart ranges* to control the extent of the chart to export:
    - X 축 상에서 단면 곡선 시작 포인트로부터의 *Minimum distance* 최단 거리 및 *Maximum distance* 최장 거리
    - Y 축 상에서 *Minimum elevation* 최저 표고 및 *Maximum elevation* 최고 표고
  - The *Distance axis settings*: helps you configure the horizontal graduation of the grid over which the elevation profile is rendered. You can customize the *Label interval*, the *Major grid line interval* and the *Minor grid line interval*.
  - The *Elevation axis settings*, same as above, for vertical graduation

-  Export As Image: Exports plots to an image file format. Specific *Export options* such as the *Image width* and *Image height* are requested, along with the *Chart ranges*, *Distance axis settings* and *Elevation axis settings* previously mentioned.
-  Export results: Exports the profile results as a set of features of a vector layer (`.DXF`, `.CSV`, `.SHP`, `.GPKG`, ⋯ ). Depending on the output format, multiple layers may be returned when the geometry types of exported features differ.
  - *Export 3D features*: Exports profile lines as 3D features, with elevation values taken from the elevation slices and stored in exported geometry Z values.
  - *Export 2D profile*: Exports profiles as 2D profile lines, with elevation stored in exported geometry Y dimension and distance in X dimension (as shown in the elevation profile widget).
  - *Export distance/elevation table*: Exports profiles as a table of sampled distance vs elevation values.

CHAPTER 8

# 일반 도구

## 8.1 컨텍스트 도움말

특정 문제에 대해 도움말이 필요한 경우, 대부분의 대화창에 있는 *Help* 버튼을 통해 최신 사용자 지침서에서 대응하는 페이지를 볼 수 있습니다. 이때 외부 플러그인의 경우 전용 웹페이지로 이동할 수도 있다는 사실을 기억하십시오.

## 8.2 패널

QGIS 는 작업할 수 있는 패널 여러 개를 기본적으로 제공합니다. 이 패널들 가운데 일부는 이 절에서 설명하고 있으며, 이 문서의 다른 부분들에서 다른 패널들의 정보를 찾아볼 수도 있습니다. *View* → *Panels* → 메뉴에서 QGIS 가 제공하는 기본 패널들의 전체 목록을 볼 수 있으며, 패널 에서 이를 언급하고 있습니다.

### 8.2.1 레이어 패널

맵  범례 라고도 불리는 *Layers* 패널에는 프로젝트의 모든 레이어 목록이 표시되어, 사용자가 레이어의 가시성을 관리하며 맵의 모양을 정할 수 있습니다. `Ctrl+1` 단축키를 누르면 이 패널을 숨기거나 나타나게 할 수 있습니다.

QGIS 는 프로젝트에 레이어를 추가하는 다양한 방법을 제공하고 있습니다:

- 데이터 소스 관리자 대화창의 전용 데이터 제공자 탭에서 *Add* 버튼을 누릅니다.
- QGIS 탐색기 패널 또는데이터베이스 관리자 에서 파일 및 레이어를 더블클릭하거나, QGIS 로 드래그 & 드롭하거나, 또는 컨텍스트 메뉴를 사용합니다.
- 운영체제의 파일 탐색기에서 QGIS 로 파일을 드래그 & 드롭합니다.

이 모든 시나리오에서, 사용자는 한번에 하나 또는 여러 레이어를 열 수 있습니다. 새 레이어는 *Layers* 패널에 추가됩니다:

1. *Layers* 패널로 드래그 & 드롭하는 경우, 정확히 마우스 버튼을 놓은 위치에 레이어를 추가합니다.

2. 드래그 & 드롭 방법이 아닌 경우, 전체 수준새 레이어 추가 시 쓰이는 습성 설정을 따르는 위치에 레이어를 추가합니다.
3. 여러 레이어를 추가하는 경우, 다음과 같은 (위에서 아래 방향) 논리에 따라 레이어들을 피처들이 가능한 한 많이 보이게 할 수 있도록 하는 논리적인 순서로 정렬합니다:
   - 벡터 포인트 레이어
   - 벡터 라인 레이어
   - 벡터 폴리곤 레이어
   - 점구름 레이어
   - 메시 레이어
   - 래스터 레이어

*Layers* 패널 상단에 있는 툴바의 기능은 다음과 같습니다:

- Open the layer styling dock (F7): 레이어 스타일 작업 패널을 켜고 끕니다.
- Add new group: 그룹 및 레이어와 대화형 작업 을 참조하세요.
- Manage Map Themes: 레이어의 가시성을 제어하고 서로 다른맵 테마 에 레이어를 배치합니다.
- 범례 트리에 있는 레이어를 필터링합니다:
  - *Filter Legend by Map Content*: 레이어들 가운데, 가시화되어 있고 레이어 피처가 현재 맵 캔버스와 교차하는 레이어만 레이어 패널에서 레이어 스타일을 렌더링합니다. 그렇지 않은 경우 레이어에 일반 NULL 심볼을 적용합니다. 레이어 심볼을 기반으로 하는 이 습성은 사용자의 관심 지역을 커버하는 레이어의 피처 유형을 식별하는 데 유용합니다.
  - *Show Private Layers*: 프로젝트 설정을 수정하지 않고서도 *Layers* 패널에 있는프라이빗 레이어 *(private layer)* 를 표시하고 대화형 작업을 할 수 있는 유용한 단축키입니다.
  - *Show Broken Layers Only*: only layers with broken data sources are displayed.
- Filter Legend by Expression: 선택한 레이어 트리에서, 조건을 만족시키는 피처가 없는 레이어의 스타일을 제거하는 표현식을 적용하도록 해줍니다. 예를 들어 기준으로 설정한 다른 레이어의 영역/피처 내부에 있는 피처만 강조하도록 할 수도 있습니다. 드롭다운 목록에서 현재 적용된 표현식을 편집하고 삭제할 수 있습니다.
- Expand All 또는 Collapse All: 레이어 패널에 있는 모든 레이어 및 그룹을 펼치거나 접을 수 있습니다.
- Remove Layer/Group: 현재 선택한 레이어나 그룹을 제거합니다.

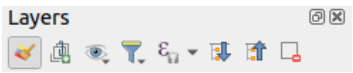



그림 8.1: 레이어 패널의 레이어 툴바

**참고**

레이어 패널을 관리하는 도구들은 인쇄 조판기의 맵과 범례 항목을 배치하는 데에도 쓸 수 있습니다.

### 맵 테마 환경설정하기

 Manage Map Themes 드롭다운 버튼은 *Layers* 패널에 있는 레이어들의 가시성을 조작할 수 있는 편리한 도구들을 제공합니다.

-  *Show All Layers*
-  *Hide All Layers*
-  *Show Selected Layers*
-  *Hide Selected Layers*
-  *Toggle Selected Layers*: 패널에서 처음 선택한 레이어의 가시성을 변경하고, 해당 상태를 다른 선택한 레이어들에 적용합니다. `Space` 단축키를 통해서도 접근할 수 있습니다.
- *Toggle Selected Layers Independently*: 선택한 레이어 각각의 가시성 상태를 변경합니다.
-  *Hide Deselected Layers*

레이어 가시성을 간단하게 조작할 수 있을 뿐만 아니라,  Manage Map Themes 메뉴는 범례에서 **맵 테마** 를 환경설정할 수 있으며 맵 테마를 변경할 수도 있습니다. 맵 테마는 다음과 같은 정보를 기록하는 현재 맵의 **스냅샷** 입니다:

- *Layers* 패널에서 보이도록 설정된 레이어들
- **그리고** 각 가시 레이어의:
  - 레이어에 적용된스타일 관련 사항
  - *Layers* 패널에서 노드 항목이 체크된 레이어 같은, 스타일의 가시 범주. 이는 단일 심볼 렌더링이 아니라심볼 에 적용됩니다.
  - 레이어 노드 (들) 및 레이어가 위치한 그룹 (들) 의 펼쳐진/접힌 상태

맵 테마를 생성하려면:

1. 표시하고자 하는 레이어를 체크하십시오.
2. 평소대로 레이어 속성 (심볼, 다이어그램, 라벨 등등) 을 환경설정하십시오.
3. 하단에 있는 *Style* 메뉴를 펼친 다음 *Add…*를 클릭해서 설정을프로젝트에 내장된 새 스타일 로 저장하십시오.

> **참고**
>
> 맵 테마는 현재 속성의 상세 사항을 기억하지 않습니다. 스타일 이름 참조 정보만 저장하기 때문에, 이 스타일이 활성화된 상태에서 (예를 들어 심볼 렌더링을 변경하는 것과 같이) 레이어에 수정 사항을 적용할 때마다 맵 테마는 새 정보로 업데이트됩니다.

4. 다른 레이어에 대해서도 필요한 만큼 이전 단계를 반복합니다.
5. 적용할 수 있다면, *Layers* 패널에서 그룹 또는 가시 레이어 노드를 펼치거나 접습니다.
6. 패널 상단에 있는  Manage Map Themes 버튼을 클릭한 다음, *Add Theme…*를 클릭해서 테마를 추가합니다.
7. 맵 테마의 이름을 입력한 다음 *OK* 를 클릭합니다.

새 테마가  드롭다운 메뉴의 목록 하단에 추가됩니다.

사용자가 필요한 만큼 제한없이 맵 테마를 생성할 수 있습니다. 위와 같이 정의한 기존 맵 테마 내용 가운데 맵 범례에 있는 현재 조합 (가시 레이어, 활성 스타일, 맵 범례 노드) 과 일치하지 않는 것이 하나라도 있다면, *Add*

*Theme*…를 클릭해서 새 맵 테마를 생성하거나, *Replace Theme* 메뉴를 사용해서 맵 테마를 업데이트하십시오. *Rename Current Theme*…버튼으로 활성화된 맵 테마의 이름을 변경할 수도 있고, *Remove Current Theme* 버튼을 클릭하면 활성 테마를 삭제할 수 있습니다.

맵 테마는 미리 환경설정해놓은 서로 다른 조합들을 빠르게 바꾸는 데 유용합니다. 목록에서 맵 테마를 선택하면 해당 테마의 조합을 복구합니다. 인쇄 조판기에서도 환경설정해놓은 테마에 접근할 수 있기 때문에, 특정 테마에 기반하면서도 현재 주 캔버스의 렌더링에 독립적인 서로 다른 맵 항목들을 생성할 수 있습니다. (맵 항목 레이어 참조)

### 레이어 패널의 컨텍스트 메뉴 개요

툴바 아래 있는 레이어 패널의 주요 요소는 프로젝트에 추가된 – 그룹으로 묶여 있을 수도 있는 – 모든 레이어 목록을 담고 있는 프레임입니다. 레이어 옆의 체크박스가 체크되어 있다면, 축척 기반 가시성 을 설정하지 않은 이상 맵 캔버스 범위와 중첩하는 해당 레이어의 내용이 표시된다는 의미입니다. 범례에서 레이어를 선택한 다음 위아래로 드래그하면 Z 순서 (Z-ordering) 를 변경할 수 있습니다. Z 순서란 범례 위쪽에 더 가까운 레이어가 범례 아래쪽에 있는 레이어 위에 렌더링된다는 뜻입니다. 또한 레이어 또는 레이어 그룹을 여러 QGIS 인스턴스들 사이로 드래그할 수도 있습니다.

> **참고**
>
> 레이어 정렬 패널을 통해 범례의 Z 순서를 무시하게 할 수 있습니다.

Depending on the item selected in the panel, a right-click shows a dedicated set of options presented below. Some of these entries are not available when multiple selections are made (e.g., *Filter*, *Rename*, *Properties*).

표 8.1: *Layers* 패널 항목의 컨텍스트 메뉴

| 옵션 | 그룹 | 벡터 레이어 | 래스터 레이어 | 메시 레이어 | 점구름 레이어 | 3D 레이어 |
|---|---|---|---|---|---|---|
| *Zoom to Layer(s)/Group* | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| *Zoom to Selection* | | ✓ | | | | |
| *Show in Overview* | | ✓ | ✓ | ✓ | ✓ | ✓ |
| *Show Feature Count* | | ✓ | | | | |
| *Show Label* | | ✓ | | ✓ | | |
| *Copy Layer/Group* | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| *Rename Layer/Group* | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| *Zoom to Native Resolution (100%)* | | | ✓ | | | |
| *Stretch Using Current Extent* | | | ✓ | | | |
| *Create Raster Attribute Table* | | | ✓ | | | |
| *Open Raster Attribute Table* | | | ✓ | | | |
| *Load Raster Attribute Table from VAT.DBF* | | | ✓ | | | |
| *Update SQL Layer*… | | ✓ | | | | |
| *Execute SQL*… | | ✓ | | | | |

다음 페이지에 계속

표 8.1 – 이전 페이지에서 계속

| 옵션 | 그룹 | 벡터 레이어 | 래스터 레이어 | 메시 레이어 | 점구름 레이어 | 3D 레이어 |
|---|---|---|---|---|---|---|
| *Edit Virtual Layer…* | | ✓ | | | | |
| *Add Group* | ✓ | | | | | |
| *Duplicate Layer* | | ✓ | ✓ | ✓ | ✓ | ✓ |
| *Remove Layer/Group…* | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| *Move Out of Group* | | ✓ | ✓ | ✓ | ✓ | |
| *Move to Top* | ✓ | ✓ | ✓ | ✓ | ✓ | |
| *Move to Bottom* | ✓ | ✓ | ✓ | ✓ | ✓ | |
| *Check and all its Parents* | | ✓ | ✓ | ✓ | ✓ | |
| *Group Selected* | | ✓ | ✓ | ✓ | ✓ | |
| *Open Attribute Table* | | ✓ | | | | |
| *Toggle Editing* | | ✓ | | ✓ | | |
| *Current Edits* | | ✓ | | ✓ | | |
| *Filter…* | | ✓ | ✓ | | ✓ | |
| *Change Data Source…* | | ✓ | ✓ | ✓ | ✓ | |
| *Repair Data Source…* | | ✓ | ✓ | ✓ | ✓ | |
| *Actions on selections* (편집 모드에서) | | ✓ | | | | |
| *Duplicate Feature* | | ✓ | | | | |
| *Duplicate Feature and Digitize* | | ✓ | | | | |
| *Set Layer Scale Visibility…* | | ✓ | ✓ | ✓ | ✓ | ✓ |
| *Zoom to Visible Scale* | | ✓ | ✓ | ✓ | ✓ | |
| *Layer CRS* | | ✓ | ✓ | ✓ | ✓ | ✓ |
| *Set Project CRS from Layer* | | ✓ | ✓ | ✓ | ✓ | ✓ |
| *Set to..* (최근 사용한 좌표계) | | | | ✓ | ✓ | ✓ |
| *Set Layer CRS…* | | ✓ | ✓ | ✓ | ✓ | ✓ |
| *Set Group CRS…* | ✓ | | | | | |
| *Set Group WMS Data…* | ✓ | | | | | |
| *Mutually Exclusive Group* | ✓ | | | | | |
| *Check and all its children (Ctrl-click)* | ✓ | | | | | |
| *Uncheck and all its children (Ctrl-click)* | ✓ | | | | | |
| *Make Permanent* | | ✓ | | | | |
| *Export* | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| *Save As…* | | | ✓ | | | |
| *Save Features As…* | | ✓ | | | | |

다음 페이지에 계속

표 8.1 – 이전 페이지에서 계속

| 옵션 | 그룹 | 벡터 레이어 | 래스터 레이어 | 메시 레이어 | 점구름 레이어 | 3D 레이어 |
|---|---|---|---|---|---|---|
| *Save Selected Features As…* | | ☑ | | | | |
| *Save As Layer Definition File…* | ☑ | ☑ | ☑ | ☑ | ☑ | ☑ |
| *Save As QGIS Layer Style File…* | | ☑ | ☑ | ☑ | ☑ | ☑ |
| *Styles* | | ☑ | ☑ | ☑ | ☑ | ☑ |
| *Copy Style* | | ☑ | ☑ | ☑ | ☑ | ☑ |
| *Paste Style* | ☑ | ☑ | ☑ | ☑ | ☑ | |
| *Add…* | | ☑ | ☑ | ☑ | ☑ | ☑ |
| *Rename Current…* | | ☑ | ☑ | ☑ | ☑ | ☑ |
| *Edit symbol…* | | ☑ | | | | |
| *Copy Symbol* | | ☑ | | | | |
| *Paste Symbol* | | ☑ | | | | |
| *Add Layer Notes…* | | ☑ | ☑ | ☑ | ☑ | ☑ |
| *Edit Layer Notes…* | | ☑ | ☑ | ☑ | ☑ | |
| *Remove Layer Notes* | | ☑ | ☑ | ☑ | ☑ | |
| *Properties…* | | ☑ | ☑ | ☑ | ☑ | ☑ |

GRASS 벡터 레이어의 경우, Toggle editing 을 사용할 수 없습니다. GRASS 벡터 레이어 편집에 대한 정보는*GRASS 벡터 레이어 디지타이즈 및 편집하기* 를 참조하십시오.

### 그룹 및 레이어와 대화형 작업

범례 창에 있는 레이어들을 그룹화할 수 있습니다. 그룹화하는 데에는 다음과 같은 서로 다른 방법들이 있습니다:

1. 아이콘을 클릭해서 새 그룹을 추가합니다. 그룹의 이름을 입력한 다음 `Enter` 를 누르십시오. 이제 기존 레이어를 클릭한 채로 그룹으로 드래그하십시오.
2. 하나 이상의 레이어를 선택한 다음 아이콘을 누르십시오. 선택한 레이어가 새 그룹으로 자동 삽입됩니다.
3. 레이어 몇 개를 선택하고, 범례 창을 오른쪽 클릭한 다음 *Group Selected* 를 선택하십시오. 새 그룹이 만들어지고 선택한 레이어들이 이 그룹으로 자동 이동될 것입니다.

레이어를 그룹에서 꺼내려면, 레이어를 밖으로 드래그하거나 또는 오른쪽 클릭한 다음 *Move Out of Group* 을 선택하면 됩니다. 레이어가 원래 있던 그룹과 동일한 수준으로 이동합니다. 또, 그룹을 다른 그룹 안에 넣을 수도 있습니다. 레이어가 그룹 안의 그룹에 있는 경우, *Move Out of Group* 메뉴는 레이어를 모든 그룹에서 꺼낼 것입니다.

레이어 패널 최상단으로 그룹 또는 레이어를 이동시키려면, 최상단으로 드래그하거나 *Move to Top* 메뉴를 선택하십시오. 그룹 안에 들어 있는 레이어에 이 메뉴 옵션을 사용하면, 해당 레이어가 들어 있는 현재 그룹 안에서 최상단으로 이동합니다. *Move to Bottom* 메뉴는 레이어와 그룹을 동일한 논리에 따라 최하단으로 이동시킵니다.

그룹의 체크박스를 클릭하면 그룹에서 체크된 레이어를 표시하거나 숨길 것입니다. `Ctrl` 키를 누른 채 체크박스를 클릭하면 그룹 및 하위그룹에 있는 모든 레이어를 켜거나 끕니다.

체크된/체크 해제된 레이어를 `Ctrl` 키를 누른 채 클릭하면 해당 레이어와 모든 부모 레이어를 체크 해제/체크합니다.

**Mutually Exclusive Group** 옵션을 활성화하면 한 번에 한 레이어만 보이는 그룹을 생성할 수 있습니다. 그룹 내부의 어떤 레이어를 보이도록 설정할 때마다 다른 레이어들은 보이지 않도록 꺼질 것입니다.

왼쪽 마우스 버튼으로 레이어를 선택할 때 `Ctrl` 키를 누르고 있으면 하나 이상의 레이어 또는 그룹을 동시에 선택할 수 있습니다. 그 다음 선택한 모든 레이어를 동시에 새 그룹으로 이동시킬 수 있습니다.

`Ctrl` 키로 항목 몇 개를 선택한 다음 `Ctrl+D` 를 누르면 하나 이상의 레이어 또는 그룹을 한 번에 삭제할 수도 있습니다. 이렇게 하면, 선택한 모든 레이어 또는 그룹이 레이어 목록에서 한꺼번에 제거될 것입니다.

#### 표시자 (indicator) 아이콘을 통한 레이어 및 그룹에 대한 상세 정보

*Layers* 패널의 레이어 또는 그룹 옆에 있는 서식 또는 아이콘이 레이어/그룹에 대한 상세 정보를 제공하기 위해 변경되는 경우가 종종 있습니다. 다음과 같은 요소로 나타납니다:

-  : 레이어가 편집 상태여서 사용자가 데이터를 수정할 수 있다는 것을 나타냅니다.
-  : 편집 중인 레이어에 일부 저장되지 않은 변경 사항이 있다는 것을 나타냅니다.
-  : 레이어에필터 가 적용되어 있다는 사실을 나타냅니다. 이 아이콘 위에 마우스 커서를 가져가면 필터 표현식을 볼 수 있으며, 더블클릭하면 쿼리를 업데이트할 수 있습니다.
-  : 프로젝트에필수적 인, 따라서 제거할 수 없는 레이어를 식별합니다.
-  : 내장된 그룹 또는 레이어 및 원래 프로젝트 파일의 경로를 식별합니다.
-  : 프로젝트 파일을 열 때 데이터소스를 사용할 수 없는 레이어를 식별합니다. (깨진 파일 경로 처리 를 참조하세요.) 아이콘을 클릭하거나, 레이어의 컨텍스트 메뉴에서 *Repair Data Source*…항목을 선택하면 소스 경로를 업데이트할 수 있습니다.
-  : 레이어가임시 스크래치 레이어 이며 현재 프로젝트를 닫으면 레이어의 내용이 사라질 것이라는 사실을 알려줍니다. 데이터 손실을 피하고 레이어를 항구적으로 만들려면, 아이콘을 클릭해서 레이어를 QGIS 가 지원하는 GDAL 벡터 유형 가운데 하나로 저장하십시오.
-  : 오프라인 편집 작업 모드 에 쓰이는 레이어를 식별합니다.
-  : 좌표계가 없거나, 알려지지 않은 좌표계를 가진 레이어를 식별합니다.
-  : 내재적으로 정밀도가 떨어지는 좌표계에 좌표를 저장하고 있는 레이어를 나타냅니다. (이 표시자 아이콘을 활성화하려면대응 설정 이 필요합니다.)
-  : 캔버스 애니메이션이 제어하는 시계열 레이어를 식별합니다.
-  : 관련노트 를 가지고 있는 레이어를 식별합니다.
- 맵 캔버스의 현재 축척이 레이어의 가시성 축척 범위를 벗어나는 경우 (레이어의 *Rendering* 속성에 설정된 대로) 해당 레이어 이름이 회색으로 바뀝니다. 레이어 컨텍스트 메뉴의 *Zoom to Visible Scale* 항목을 선택하면 맵을 해당 레이어의 가장 가까운 가시성 축척으로 확대/축소합니다.

### 그룹 작업을 통한 레이어 렌더링 제어

그룹은 프로젝트의 트리 내부에 레이어들을 구조화하는 수단이기도 하지만, 그룹을 구성하는 레이어들을 어떻게 렌더링할지에 영향을 미칠 수도 있습니다. 예를 들면 맵을 렌더링하는 동안 그룹 레이어들을 평탄화된 단일 객체로 취급한다던지 말이죠.

그룹을 선택할 때마다 *Layer Styling* 패널에서 이런 렌더링을 위한 옵션을 설정할 수 있습니다. Symbology 탭에서 *Render Layers as a Group* 을 체크하면 그룹의 하위 레이어들의 모양을 개별 레이어들이 아닌 단일 객체로 제어할 수 있는 옵션 집합이 활성화됩니다.

- *Opacity*: 다른 하위 레이어가 가리고 있는 하위 레이어의 피처를 가려진 채로 유지하고, 불투명도는"그룹 전체"에 적용됩니다.

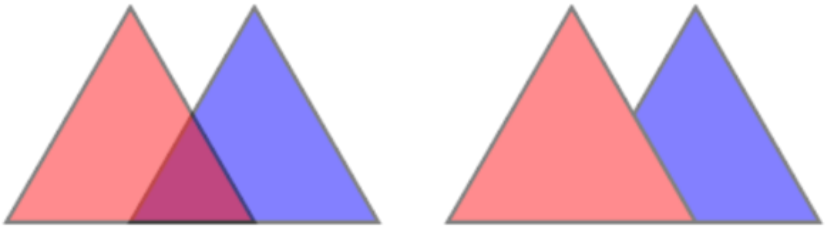


그림 8.2: 개별 레이어에 불투명도 설정 | 그룹에 불투명도 설정
왼쪽 이미지는 50% 불투명도로 렌더링된 레이어 2 개를 보여주고 있습니다. (아래에 있는 피처가 보이기는 하지만, 위에 있는 빨간 레이어가 50% 투명도로 반쯤 가리고 있습니다.) 오른쪽 이미지는 그룹에 불투명도를 설정한 결과를 보여줍니다. (위에 있는 빨간 레이어가 아래에 있는 파란 레이어의 부분을 완전히 가린 상태로 50% 불투명도를 적용해서 렌더링한 결과입니다.)

- *Blend modes*: 불투명도와 마찬가지로, 그룹 전체에 (곱하기, 중첩 등등의) 혼합 모드 를 설정하면 먼저 위에 있는 레이어가 아래 있는 레이어를 가린 상태로 그룹의 하위 레이어들의 피처를 평탄화합니다. 그 다음 평탄화된 그룹과 그룹 아래 있는 레이어들을 혼합해서 렌더링합니다.
  - 하위 레이어에 할당된 혼합 모드가 존재하는 경우, 평탄화 이전에 적용되지만 해당 그룹의 다른 하위 레이어들에만 영향을 미치도록 제한됩니다. 그룹 아래에 있는 다른 레이어들에는 영향을 미치지 않습니다.
  - 하위 레이어의 *Symbology* 탭에 그룹의 하위 레이어들에 대해 사용할 수 있는, 렌더링 시 다른 하위 레이어에"자르기 (clip)"스타일의 작업을 수행하는혼합 모드 옵션이 몇 개 존재합니다. 예를 들어 한 레이어의 렌더링을 다른"마스크"레이어의 렌더링으로 자를 수 있습니다.
- *Layer effects*: 하위 레이어들의 평탄화된 렌더링에만그리기 효과 를 적용합니다. 즉, 예를 들어 그룹에 적용된 그림자 생성 (Drop Shadow) 효과는 가려진 하위 레이어에는 나타나지 않을 것입니다.

그룹에 *Render layers as a group* 옵션을 설정한 경우, 그룹만 *Layer Order* 패널 목록에 표시될 것입니다. 목록 순서가 그룹 레이어의 배치에 따라 결정되기 때문에, 이 순서 목록에는 그룹의 하위 레이어들은 표시되지 않을 것입니다.

### 레이어 스타일 편집하기

*Layers* 패널에서 사용자가 레이어 렌더링을 빠르고 쉽게 변경할 수 있는 단축키들이 있습니다.

레이어를 오른쪽 클릭한 다음 컨텍스트 메뉴 목록에서 *Styles* 를 선택하십시오:

- 현재 해당 레이어에 사용할 수 있는스타일 을 볼 수 있습니다. 사용자가 레이어에 대해 많은 스타일들을 정의했다면, 한 스타일에서 다른 스타일로 변환할 때마다 맵 캔버스 상에서 사용자의 레이어 렌더링이 자동으로 업데이트될 것입니다.

- 현재 스타일의 일부분 또는 전부를 복사해서, 이 스타일을 적용할 수 있는 다른 레이어에 붙여넣을 수 있습니다.
- *Rename current*…: 현재 스타일의 이름을 재명명합니다.
- *Add*: 새 스타일을 추가합니다. (실제로는 현재 스타일의 복사본을 추가합니다.)
- *Remove current*: 현재 스타일을 제거합니다. (스타일 여러 개가 적용되어 있는 경우에만 사용할 수 있는 메뉴입니다.)

> **팁**
>
> **손쉽게 레이어 스타일 공유**
>
> 컨텍스트 메뉴에서 레이어의 스타일을 복사한 다음 레이어 그룹 또는 선택 집합에 붙여넣으십시오: 해당 스타일이 원본 레이어와 동일한 (벡터, 래스터, 메시, 점구름 등등) 유형의 레이어 모두에 적용됩니다. 다만 벡터 레이어의 경우 동일한 (포인트, 라인 또는 폴리곤) 도형 유형을 가진 레이어에만 적용됩니다.

피처 분류에 기반한 심볼을 (예를 들어 벡터 레이어의 경우범주 *(categorized)*, 등급 *(graduated)*, 또는규칙 기반 *(rule-based)* 심볼을, 또는 점구름 레이어의 경우범주화 *(classification)* 심볼을) 사용하는 경우, *Layers* 패널에 있는 클래스 항목을 오른쪽 클릭하면 해당 클래스의 (그리고 해당 클래스 피처의) 가시성을 한번에 편집할 수 있습니다. 즉 특정 클래스 피처들의 가시성을 하나하나 체크하거나 해제하지 않아도 됩니다:

- *Toggle Items*: 해당 클래스 피처들을 켜고끕니다.
- *Show All Items*: 모든 피처를 표시합니다.
- *Hide All Items*: 모든 피처를 숨깁니다.

벡터 레이어의 경우, 클래스 지엽 (leaf) 항목의 컨텍스트 메뉴를 통해 다음 메뉴 항목에 접근할 수 있습니다:

- *Select features*: 레이어에서 해당 클래스와 일치하는 모든 피처를 선택합니다.
- *Show in attribute table*: 해당 클래스와 일치하는 피처만 필터링한 속성 테이블을 엽니다.
- **색상환 (Color Wheel)** 을 이용해서심볼 색상 을 업데이트할 수 있습니다. 사용자 편의를 위해 색상환 하단에서 최근 사용한 색상을 선택할 수도 있습니다.
- *Edit Symbol*…: 피처 심볼 (심볼, 크기, 색상, …) 을 변경할 수 있는심볼 선택기 대화창을 엽니다.
- *Copy Symbol*: 심볼을 복사합니다.
- *Paste Symbol*: 심볼을 붙여넣습니다.

> **팁**
>
> 범주의 지엽 (leaf) 항목을 더블 클릭하면 *Symbol Selector* 대화창을 엽니다.

### 8.2.2 레이어 스타일 작업 패널

*Layer Styling* 패널은 (`Ctrl+3` 키 조합으로 활성화할 수 있으며) *Layer Properties* 대화창의 기능 가운데 일부를 실행하는 단축키입니다. 이 패널은 레이어 속성 대화창을 열지 않고도 레이어의 렌더링 방식과 습성을 정의해서 그에 따라 시각화할 수 있는 빠르고 손쉬운 방법을 제공합니다.

레이어 스타일 작업 패널은"모달 (modal)"대화상자가 레이어 속성 대화창을 가리는 일을 방지할 뿐만 아니라, 이미 대부분의 스타일 기능 (색상 선택기, 효과 속성, 규칙 편집, 라벨 대체 등등) 을 내장하고 있기 때문에 대화창들이 화면을 덮어버리는 일도 방지해줍니다. 예를 들어, 레이어 스타일 작업 패널 내부의 색상 버튼을 클릭하면 색상 선택기가 개별 대화창이 아니라 레이어 스타일 패널 안에서 열립니다.

레이어 패널에 있는 현재 레이어들의 드롭다운 목록에서 항목을 선택한 다음:

- 활성화된 항목에 따라:
  - 그룹의 경우 *Symbology* 를 설정합니다. (그룹 작업을 통한 레이어 렌더링 제어 을 참조하세요.)
  - 래스터 레이어의 경우 레이어의 *Symbology*, *Transparency*, *Histogram* 을 설정합니다. 이 옵션들은래스터 속성 대화창 과 동일합니다.
  - 벡터 레이어의 경우 레이어의 *Symbology*, *Labels*, *Mask* 및 *3D View* 를 설정합니다. 이 옵션들은벡터 속성 대화창 과 동일하며 제 3 자 플러그인이 도입하는 사용자 지정 속성으로 확장될 수 있습니다.
  - *Symbology*, *Labels* and *3D View* properties for mesh layer. These options are the same as in the 메시 데이터셋 속성.
  - *Symbology*, *3D View* and *Elevation* properties for point cloud or 3D Tiles layer. These options are the same as in the 점구름 속성 or *3D* 타일 속성.
  - *Symbology* and *Labels* properties for vector tiles layer. These options are the same as in the 벡터 타일 데이터셋 속성.
  - *Symbology* properties for annotations layer. These options are the same as in the *feature annotation properties*.
- 전체 수준 맵 음영 효과 속성을 활성화하고 환경설정합니다.
- *Style Manager* 에서 관련 스타일 (들) 을 관리합니다. (자세한 내용은사용자 지정 스타일 관리하기 를 참조하세요.)
- 사용자가 현재 프로젝트의 레이어 스타일에 적용한 변경 *History* 전체를 추적합니다. 따라서 이력 목록에 있는 어떤 상태든 선택하고 *Apply* 버튼을 클릭하면 해당 상태 이후의 변경 사항을 취소하고 해당 상태를 복구할 수 있습니다.

이 패널의 또다른 강력한 기능은 *Live update* 체크박스입니다. 이 옵션을 활성화하면 사용자가 변경 사항을 만드는 대로 맵 캔버스에 해당 변경 사항을 즉시 렌더링하기 때문에, 더 이상 *Apply* 버튼을 클릭할 필요가 없습니다.

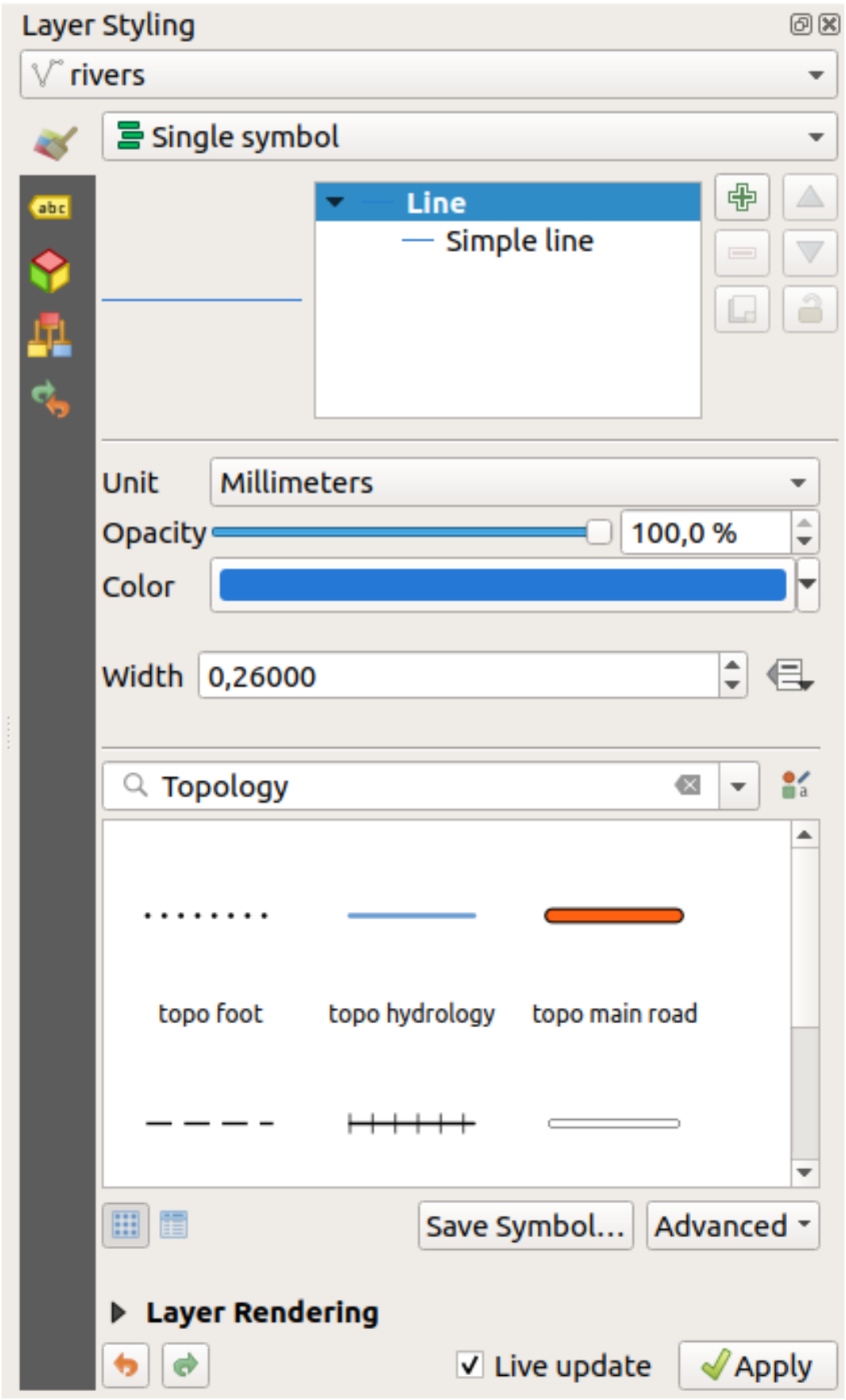



그림 8.3: 레이어 스타일 작업 패널에서 레이어 심볼 정의하기

### 8.2.3 레이어 순서 패널

QGIS 맵 캔버스에 보여지는 레이어는 기본적으로 *Layers* 패널에 있는 레이어 순서를 따라 렌더링됩니다. 패널에서 레이어가 위에 있을수록 맵 뷰에서도 위에 렌더링됩니다. (따라서 더 잘 보입니다.)

*View* → *Panels* → 메뉴에서 또는 `Ctrl+9` 키 조합으로 활성화할 수 있는 *Layer Order* 패널을 통해 레이어를 렌더링하는 순서를 레이어 패널에 있는 순서에 독립적으로 정의할 수 있습니다. 레이어 목록 아래 있는 *Control rendering order* 옵션을 체크하고 사용자가 원하는 대로 패널에 있는 레이어들을 재정렬하십시오. 예를 들어 그림 8.4 를 보십시오. 레이어 패널에 있는 레이어 위치에도 불구하고 `airports` 피처가 `alaska` 폴리곤 위에 표시되고 있습니다.

*Control rendering order* 옵션을 해제하면 기본 동작대로 레이어 페널의 순서를 따르게 됩니다.

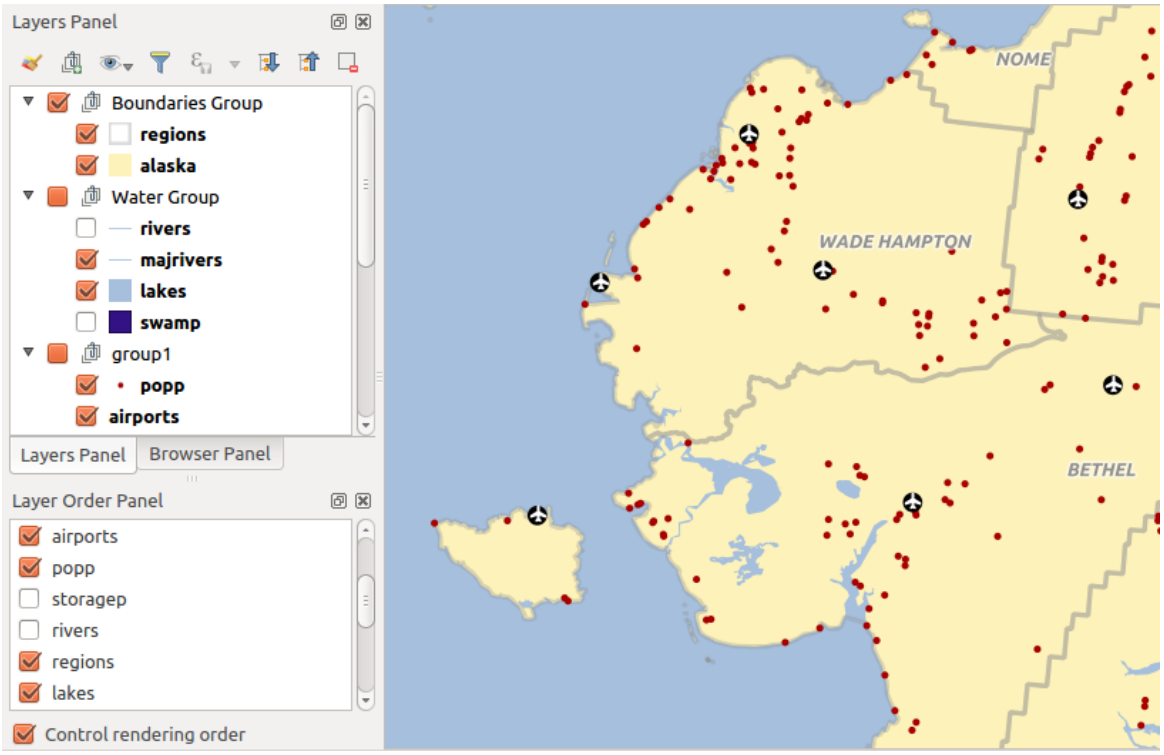



그림 8.4: 범례에 독립적인 레이어 순서 정의하기

### 8.2.4 오버뷰 패널

*Overview* 패널 (`Ctrl+8`) 은 일부 레이어들의 전체 범위를 보여주는 맵입니다. 오버뷰 맵은 *Layer* 메뉴 또는 레이어 컨텍스트 메뉴에서 *Show in Overview* 옵션을 켠 레이어들로 채워집니다. 이 뷰 안에는현재 맵 범위를 나타내는 빨간 사각형이 있습니다. 이 사각형을 통해 사용자가 현재 맵의 어느 영역을 보고 있는지 한 눈에 알 수 있습니다. 오버뷰에서 현재 범위를 보여주는 빨간색 사각형을 클릭 & 드래그하면, 그에 따라 주 맵 뷰도 자동적으로 업데이트됩니다.

맵 오버뷰가 표시하는 레이어에 라벨이 적용돼 있더라도 맵 오버뷰에는 라벨이 렌더링되지 않는다는 사실을 기억하십시오.

### 8.2.5 로그 메시지 패널

몇몇 작업을 불러오거나 공간 처리할 때,  *Log Messages Panel* 을 통해 서로 다른 탭에 나타나는 메시지를 추적하고 지켜볼 수 있습니다. 하단 상태 바의 가장 오른쪽에 있는 아이콘을 클릭하면 이 패널을 활성화할수 있습니다.

### 8.2.6 실행 취소/재실행 패널

*Undo/Redo* 패널 (`Ctrl+5`) 은 편집 중인 각 레이어에 대해 실행된 모든 작업 목록을 표시해서, 위쪽에 있는 작업을 선택하는것만으로도 작업 모음을 빠르게 실행 취소시킬 수 있습니다. 더 자세한 내용은실행 취소 및 재실행 편집 을 참조하세요.

### 8.2.7 통계 요약 패널

*Statistics* 패널 (`Ctrl+6`) 은 모든 벡터 레이어에 대한 요약된 정보를 표시할 수 있습니다. 이 패널에서 선택할 수있는 것은 다음과 같습니다:

- 통계를 계산할 벡터 레이어: 상단에 있는 드롭다운 메뉴에서 선택하거나, 또는 통계 드롭다운 목록 하단에 있는  *Follow selected layer* 를 체크하면 *Layers* 패널에 활성화되어 있는 레이어와 동기화시킬 수 있습니다.
- 사용할 필드 또는  표현식: 각 레이어에 대해 마지막 입력 내용을 기억해서 레이어 재선택 시 자동으로 계산합니다.
- 대화창 우하단의 드롭다운 버튼을 통해 반환할 통계 정보. 레이어 필드 유형에 따라 (또는 표현식의 값 유형에 따라) 사용할 수 있는 통계 정보는 다음과 같습니다:

표 8.2: 각 필드 유형에 따라 사용할 수 있는 통계

| 통계 정보 | 문자열 | 정수 | 실수 | 날짜 |
|---|---|---|---|---|
| 개수 | ☑ | ☑ | ☑ | ☑ |
| 고유 값 (Distinct Value) 개수 | ☑ | | | ☑ |
| 누락 값 (Missing value) 개수 | ☑ | ☑ | ☑ | ☑ |
| Sum | | ☑ | ☑ | |
| 평균 | | ☑ | ☑ | ☑ |
| 표준 편차 | | ☑ | ☑ | |
| 표본 표준 편차 | | ☑ | ☑ | |
| 최소값 | ☑ | ☑ | ☑ | ☑ |
| 최대값 | ☑ | ☑ | ☑ | ☑ |
| 범위 | | ☑ | ☑ | ☑ |
| 희귀값 (Minority) | ☑ | ☑ | ☑ | |
| 최빈값 (Majority) | ☑ | ☑ | ☑ | |
| 다양도 (Variety) | | ☑ | ☑ | |
| 제 1 사분위수 (First Quartile) | | ☑ | ☑ | |
| 제 3 사분위수 (Third Quartile) | | ☑ | ☑ | |
| 사분위간 범위 (Inter Quartile Range) | | ☑ | ☑ | |
| 최소 길이 | ☑ | | | |
| 최대 길이 | ☑ | | | |
| 평균 길이 | ☑ | | | |

통계 요약 정보는:

- 전체 레이어의 통계를 반환하거나, ☑ *Selected features only* 옵션으로 선택한 피처에 대한 통계만 반환할 수도 있습니다.
- 클립보드로 복사한 다음 다른 응용 프로그램에 테이블로 붙여넣을 수 있습니다.
- 기저 데이터소스가 바뀔 경우 (예를 들어 피처/필드를 새로 추가하거나, 피처/필드를 제거하거나, 속성을 수정하는 경우) 버튼을 통해 통계를 다시 계산할 수 있습니다.

### 8.2.8 디버그/개발 도구 패널

*Debugging/Development Tools* (`F12`) 패널은 QGIS 내부 작업을 처리하고 디버그하기 위한 통합 패널입니다. 다음 탭들 아래 사용할 수 있는 도구들이 모여 있습니다:

- *API Documentation*: embeds developer oriented documentation
- *Network Logger*: records and displays network requests and interactions
- *Query Logger*: logs SQL commands issued by data providers and duration of execution
- *Profiler*: provides load times for every actions in order to detect causes of slow down

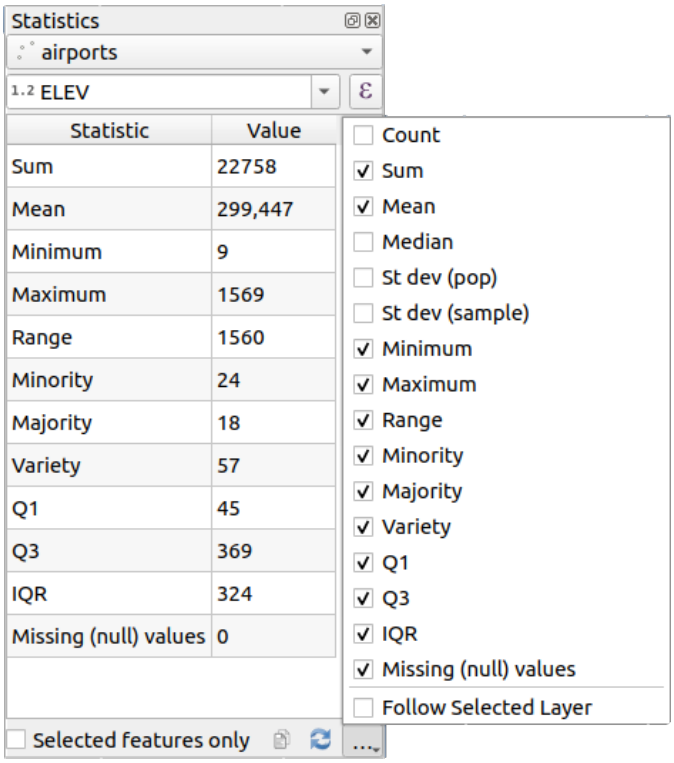



그림 8.5: 필드에 대한 통계 정보 종류 표시

**참고**

플러그인 개발자는 `registerDevToolWidgetFactory` 메소드를 이용해서 패널에 자신의 플러그인을 디버그하고 개발하기 위한 사용자 지정 탭을 추가할 수 있습니다.

### API Documentation

The *API Documentation* tab can display either the PyQGIS API documentation or the Qt API documentation. These are embedded in the widget, allowing you to browse and search the docs without leaving QGIS. It is also the place you can display specific help page for objects or classes in your code sample; from the Python interactive console or text editor dialog, select or hover a text and press `F1` or right-click and select Contextual help to get its API help. If you wish to read the active page in the system web browser, then press Open in web browser.

Read *contextual help* for details on the widget configuration.

### 네트워크 로그 관리자

*Network Logger* 탭은 네트워크 요청은 물론 요청 및 응답 상태, 헤더, 오류, SSL 환경설정 오류, 타임아웃, 캐시 상태 등등 유용한 상세 정보 전부를 기록하고 표시합니다.

최상단 툴바에서 다음과 같은 작업을 할 수 있습니다:

- *Record Log*: 로그 작업을 시작하거나 중단할 것입니다.
- *Clear Log*: 로그 이력을 초기화할 것입니다.
- *Save Log*…: 로그가 민감한 정보이기 때문에 기밀로 취급해야 한다는 경고를 띄운 뒤, 로그를 저장할 수 있게 합니다.
- *Settings* 드롭다운 메뉴를 클릭하면 성공한 요청을 표시할지 (*Show Successful Requests*), 타임아웃을 표시할지 (*Show Timeouts*), 캐시에서 서비스된 응답을 표시할지 (*Show Replies Served from Cache*) 선택할 수 있습니다.
- *Disable cache*: 모든 요청을 수행해야 하도록 캐시를 비활성화시킬 것입니다.
- *Filter requests*: URL 문자열 부분집합 또는 상태 요청을 기반으로 요청을 필터링합니다.

요청을 오른쪽 클릭하면 다음과 같은 작업을 할 수 있습니다:

- *Open URL*: 사용자의 기본 브라우저에 요청 URL 을 열 것입니다.
- *Copy URL*: URL 을 복사할 것입니다.
- *Copy As cURL*: URL 을 터미널에서 사용할 수 있는 서식으로 복사할 것입니다.
- *Copy as JSON*: 버그 보고서 또는 원격 지원에 쉽게 붙여넣을 수 있도록, 클립보드에 트리 값들을 JSON 문자열로 복사할 것입니다.

### 쿼리 로그 관리자

*Query Logger* 탭은 데이터 제공자가 전송한 SQL 명령 그리고 백엔드 데이터베이스로의 연결 API 를 QGIS 가 측정한 (예를 들어 명령을 전송한 클라이언트에서의) 실행 시간과 함께 로그로 기록합니다. QGIS 알고리즘 또는 플러그인을 디버그 또는 개발하는 도중 특정 레이어의 성능을 조사하는 경우 이 로그가 유용할 수 있습니다.

최상단 툴바에서 다음과 같은 작업을 할 수 있습니다:

- *Record Log*: 로그 작업을 시작하거나 중단할 것입니다.
- *Clear Log*: 로그 이력을 초기화할 것입니다.
- *Save Log*…: 로그가 민감한 정보이기 때문에 기밀로 취급해야 한다는 경고를 띄운 뒤, 로그를 저장할 수 있게 합니다.
- *Filter queries*: 쿼리 문자열 부분집합 또는 제공자 유형, 시작 시간, 개시자 (initiator) 등과 같은 상세 정보를 기반으로 쿼리를 필터링합니다.

보고된 쿼리를 오른쪽 클릭하면:

- *Copy SQL*: QGIS 가 데이터베이스에 호출한 SQL 명령을 복사할 것입니다.
- *Copy as JSON*: 버그 보고서 또는 원격 지원에 쉽게 붙여넣을 수 있도록, 클립보드에 트리 값들을 JSON 문자열로 복사할 것입니다.

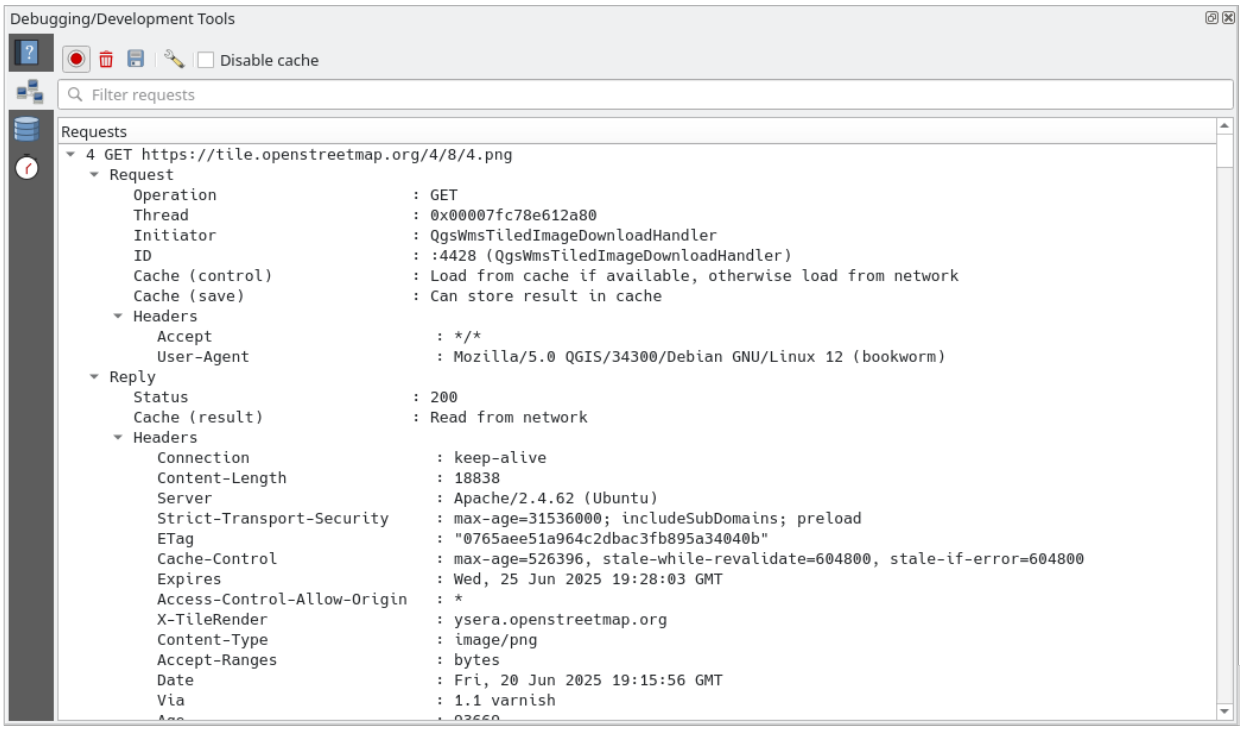



그림 8.6: GET 요청에 대한 네트워크 로그 관리자 산출물

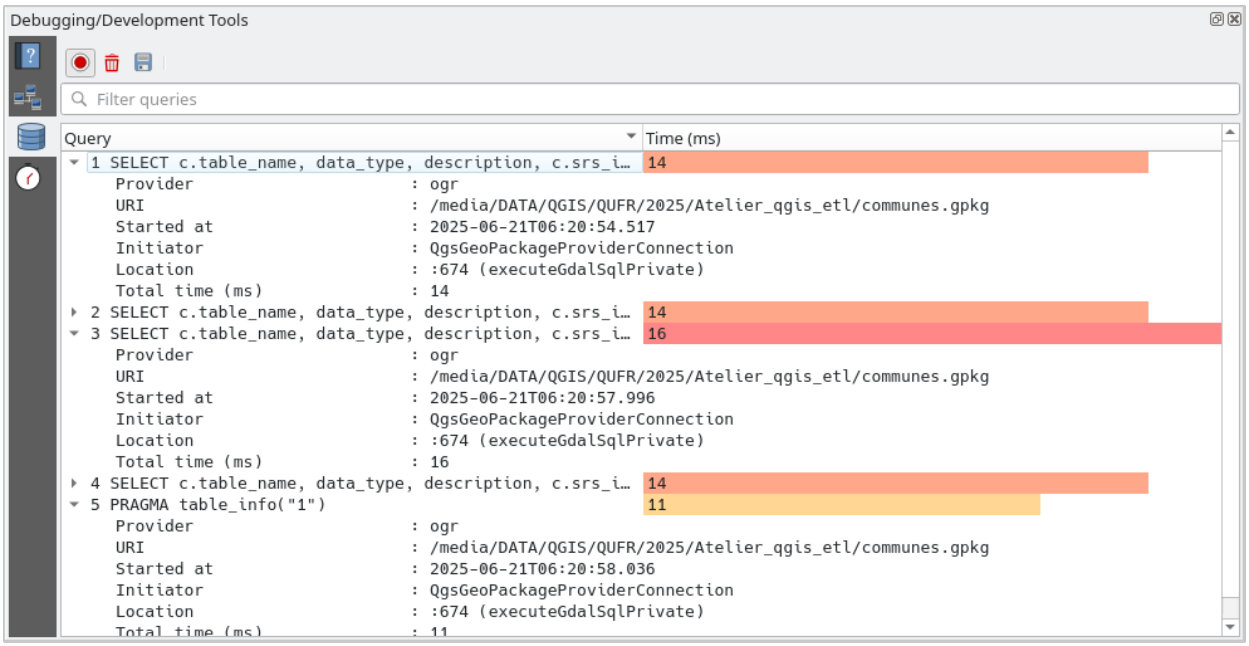



그림 8.7: 쿼리 로그 관리자 산출물

### 프로파일 관리자

*Profiler* 탭에서는 사용자가 요청한 액션과 관련된 모든 단일 작업들에 대한 불러오기 시간을 확인할 수 있습니다. 이 작업들은 그 맥락에 따라 읽어오기, 메뉴, 맵 캔버스 또는 3 차원 뷰 생성, 맵 레이어 참조 해제, 북마크 또는 레이아웃 불러오기 등등의 설정일 수 있습니다. 이 정보로 불러오기 시간이 느려진 원인을 식별할 수 있습니다.

Default supported actions can be selected from the *Category* drop-down menu:

- *Startup*: QGIS 의 구동 시간이 길어지는 경우 사용자가 문제를 진단 (및 해결) 할 수 있도록 도와줍니다.
- *Project Load*: 프로젝트를 불러오는 시간이 느릴 경우 그 원인을 식별하기 위해 프로젝트 불러오기의 여러 단계를 자세히 분석할 수 있도록 해줍니다.
- *Map Render*: 이 도구는 맵 렌더링의 문제점을 파악해서 어떤 레이어가 맵을 다시 그리는 시간을 늘어나게 하는지를 정확하게 추적할 수 있게 해줍니다.

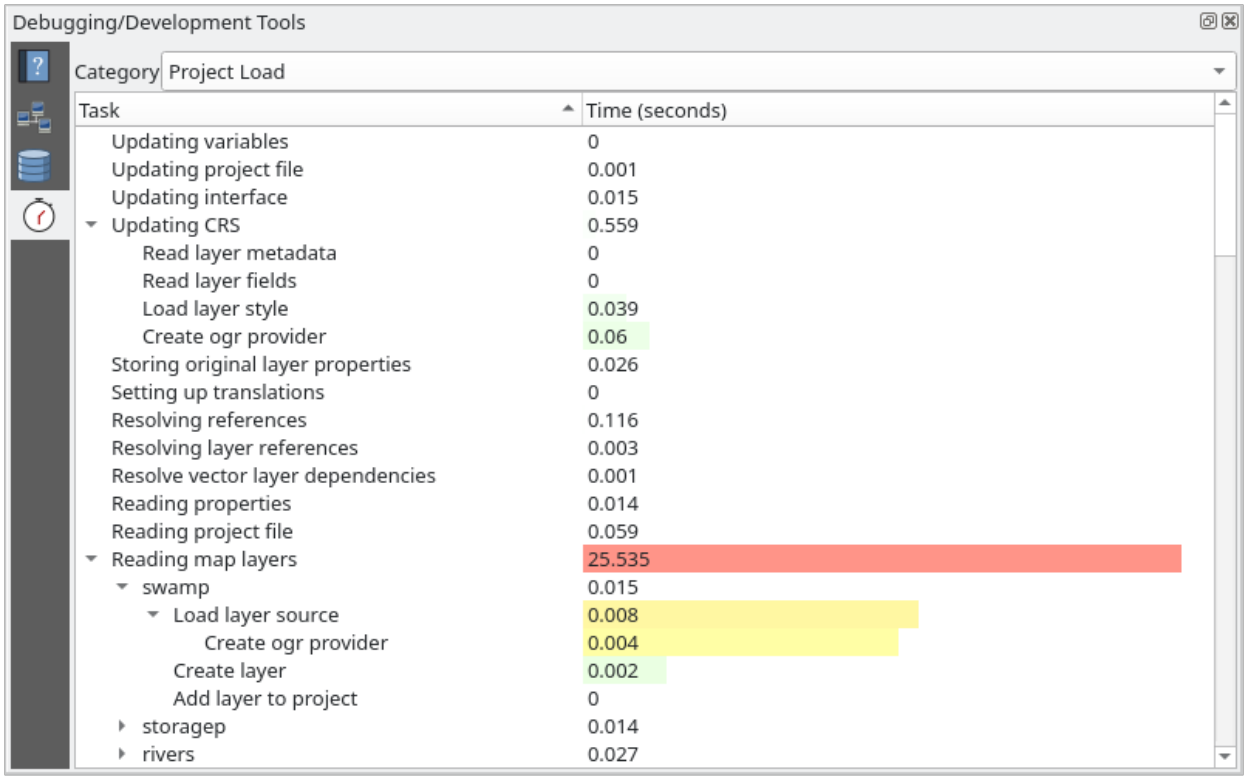



그림 8.8: QGIS 구동에 대한 프로파일 관리자

## 8.3 외부 프로젝트로부터 레이어를 내포시키기

서로 다른 프로젝트들에 있는 여러 레이어들을 동일한 스타일로 유지하고자 할 경우가 있습니다. 이 레이어들을 위한기본 스타일 을 생성할 수도 있고, 또는 다른 프로젝트로부터 레이어들을 내포시켜 수고를 줄일 수도 있습니다.

기존 프로젝트로부터 레이어 및 그룹을 내포시키면 스타일과 관련해 몇몇 장점이 생깁니다:

- 모든 유형의 (벡터 또는 래스터, 로컬 또는 온라인 등등) 레이어를 추가할 수 있습니다.
- 그룹 및 레이어를 불러오면, 서로 다른 사용자 프로젝트에 있는"배경"레이어들의 트리 구조를 동일하게 유지할 수 있습니다.
- 내포한 레이어를 편집할 수 있지만, 심볼, 라벨, 양식, 기본값, 동작 등과 같은 레이어 속성을 변경할 수는 없습니다. 이 습성은 프로젝트 전체에서 일관성을 보장합니다.
- 원본 프로젝트에 있는 항목을 수정하면, 변경 사항이 다른 모든 프로젝트로 전파됩니다.

사용자 프로젝트에 다른 프로젝트 파일의 내용을 삽입하고자 할 경우, *Layer* → *Embed Layers and Groups* 메뉴를 선택한 다음:

1. …버튼을 클릭해서 프로젝트를 찾으십시오. 프로젝트의 내용을 (그림 8.9 참조) 볼 수 있습니다.
2. `Ctrl` (또는 Cmd) 키를 누른 채 사용자가 가져오려 하는 레이어 및 그룹을 클릭하십시오.
3. *OK* 를 클릭합니다.

*Layers* 패널에 선택한 레이어 및 그룹을 삽입하고 맵 캔버스에 가시화할 것입니다. 내포된 항목의 이름 옆에 아이콘이 추가되고, 그 위에 마우스를 가져가면 원본 프로젝트 파일 경로를 보여주는 툴팁을 표시합니다.

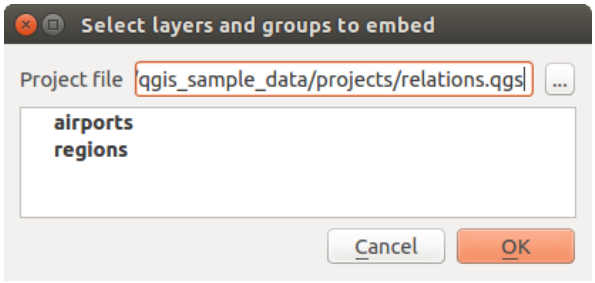



그림 8.9: 삽입할 레이어 및 그룹 선택

다른 모든 레이어와 마찬가지로, 내포 레이어를 오른쪽 클릭하고 Remove 를 선택하면 프로젝트에서 제거할 수 있습니다.

> **팁**
>
> **내포 레이어의 렌더링 변경**
>
> 원본 프로젝트 파일을 변경하지 않는 이상, 내포 레이어의 렌더링을 변경할 수는 없습니다. 하지만, 내포 레이어를 오른쪽 클릭하고 *Duplicate* 를 선택하면 완전히 동일하며 원본 프로젝트에 독립적인 레이어를 생성합니다. 그 다음 링크된 레이어를 안전하게 제거할 수 있습니다.

## 8.4 대화형 피처 작업

### 8.4.1 피처 선택

QGIS 는 맵 캔버스에서 피처를 선택할 수 있는 여러 도구를 제공하고 있습니다. *Edit → Select* 메뉴 또는 *Selection toolbar* 에서 선택 도구를 쓸 수 있습니다.

> **참고**
>
> 선택 도구는 현재 활성화된 레이어 상에서 동작합니다.

#### 맵 캔버스에서 직접 선택하기

하나 이상의 피처를 마우스로 선택하려면, 다음 도구들 가운데 하나를 쓸 수 있습니다:

-  Select Features by area or single click
-  Select Features by Polygon
-  Select Features by Freehand
-  Select Features by Radius

> **참고**
>
 Select Features by Polygon 도구를 제외하면, 이 수동 선택 도구들을 통해 맵 캔버스에서 피처 (들) 를 클릭 한 번으로 선택할 수 있습니다.

> **참고**
>
활성화 레이어에 있는 중첩된 피처를 선택하기 위해 (어떤 레이어에 있든) 기존 폴리곤을 쓰려면  Select Features by Polygon 도구를 사용하십시오. 폴리곤을 오른쪽 클릭하면 클릭한 포인트를 담고 있는 모든 폴리곤들의 목록을 보여주는 컨텍스트 메뉴가 나타납니다. 이 목록에서 원하는 폴리곤을 선택하십시오. 활성화된 레이어에 있는 모든 중첩 피처들이 선택될 것입니다.

> **팁**
>
사용자가 가장 마지막으로 선택했던 피처를 다시 선택하려면 *Edit → Select → Reselect Features* 메뉴를 사용하십시오. 사용자가 고생해서 선택해 놓은 것들이 실수로 다른 곳을 클릭해버리는 바람에 선택 해제됐을 경우 매우 유용합니다.

While using the  *Select Feature(s)* tool, holding `Shift` or `Ctrl` (`Cmd` on macOS) toggles whether a feature is selected (i.e., either adds to the current selection or removes from it).

다른 도구들의 경우, 다음 키를 누르면 서로 다른 습성이 발현됩니다:

- `Shift`: 현재 선택 집합에 피처를 추가

- `Ctrl`/`Cmd`: subtract features from the current selection
- `Ctrl+Shift`/`Cmd+Shift`: intersect with the current selection, i.e., only keep overlapping features from the current selection
- `Alt`: select features that are totally within the selection shape. Combined with `Shift` or `Ctrl`/`Cmd` keys, you can add or subtract features to/from the current selection.

### 자동 선택

속성 테이블 에서도 대부분 쓸 수 있는 다음 선택 도구들은 피처의 속성 또는 선택 상태를 바탕으로 선택 작업을 수행합니다 (속성 테이블과 맵 캔버스는 동일한 정보를 표시하기 때문에, 속성 테이블에서 피처 1 개를 선택하면 맵 캔버스에서도 선택될 것이라는 점을 기억하십시오):

-  Select Features By Expression… or press `Ctrl+F3` to select features using the expression dialog
-  Select Features By Value… 도구 또는 `F3` 키로 값을 이용해서 피처를 선택할 수 있습니다.
-  Deselect Features from All Layers 도구 또는 `Ctrl+Alt+A` 조합 키로 모든 레이어에서 선택한 모든 피처들을 선택 해제할 수 있습니다.
-  Deselect Features from the Current Active Layer 도구 또는 `Ctrl+Shift+A` 키
-  Select All Features 도구 또는 `Ctrl+A` 조합 키로 현재 레이어에 있는 모든 피처를 선택할 수 있습니다.
-  Invert Feature Selection 도구로 현재 레이어의 선택 집합을 반전시킬 수 있습니다.
-  Select by Location 도구로 (동일 또는 다른 레이어에 있는) 다른 피처들과의 공간 관계성을 바탕으로 피처를 선택할 수 있습니다. (위치로 선택하기 참조)
-  Select within distance 도구로, 참조된 피처로부터 지정한 최대 거리 안에 존재하는 모든 피처를 선택할 수 있습니다. (거리 안에서 선택하기 참조)

예를 들어 QGIS 예시 데이터의 `regions.shp` 파일에서 구 (borough) 인 지역을 찾고자 하는 경우:

1.  Select features using an Expression 아이콘을 클릭하고
2. *Fields and Values* 그룹을 펼친 다음
3. 쿼리하려는 필드를 (“TYPE_2”) 더블 클릭하고
4. 오른쪽에 나타나는 패널에 있는 *All Unique* 버튼을 클릭하면
5. 나타나는 목록에서‘Borough’를 더블 클릭하고 *Expression* 편집기 란에 다음 쿼리를 입력하십시오:

```
"TYPE_2"  =  'Borough'
```

6. *Select Features* 를 클릭합니다.

표현식 작성기 대화창에서, *Function list → Recent (Selection)* 메뉴를 이용해서 이전에 사용했던 표현식을 선택할 수도 있습니다. 이 대화창은 최근 사용한 표현식을 20 개까지 저장하고 있습니다. 더 자세한 정보 및 몇몇 예시를 살펴보려면표현식 을 참조하세요.

> **팁**
>
> **사용자 선택 집합을 새 파일로 저장**

*Edit ▸ Copy Features* 와 *Edit ▸ Paste Features as* 메뉴를 통해 선택한 피처를 각각 원하는 유형의 **New Temporary Scratch Layer** 또는 **New Vector Layer** 로 저장할 수 있습니다.

### 값으로 피처 선택

이 선택 도구는 레이어의 피처 양식을 열어 사용자가 각 필드에 대해 어떤 값을 검색할지, 대소문자를 구분해서 검색할지, 어떤 연산을 사용할지 선택할 수 있게 해줍니다. 이 도구는 자동적으로 검색란을 기존 값으로 채워주는 자동 완성 기능도 제공합니다.

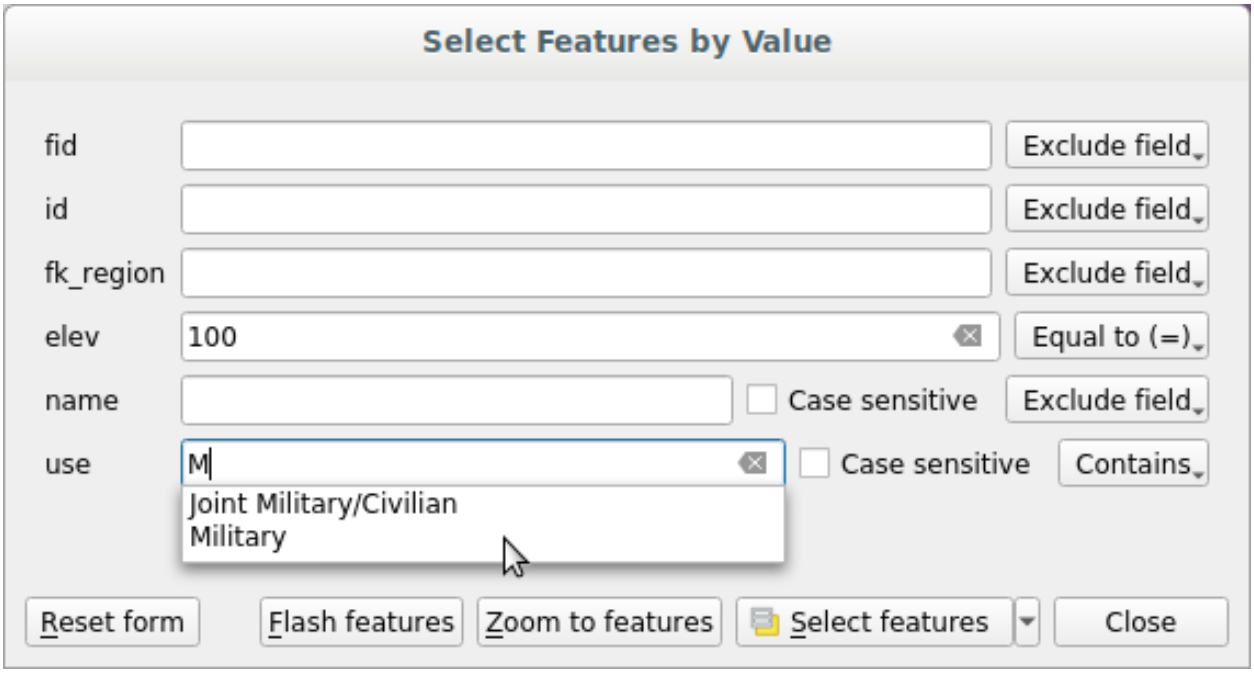



그림 8.10: 양식 대화창을 이용해 피처 필터링/선택

각 필드 옆에, 검색 습성을 제어할 수 있는 옵션을 나열한 드롭다운 목록이 있습니다:

표 8.3: 데이터 유형 별 쿼리 연산자

| 필드 검색 옵션 | 문자열 | Numeric | 날짜 |
|---|---|---|---|
| *Exclude Field* 검색에서 제외할 필드 | ☑ | ☑ | ☑ |
| *Equal to (=)* | ☑ | ☑ | ☑ |
| *Not equal to (≠)* | ☑ | ☑ | ☑ |
| *Greater than (>)* | | ☑ | ☑ |
| *Less than (<)* | | ☑ | ☑ |
| *Greater than or equal to (≥)* | | ☑ | ☑ |
| *Less than or equal to (≤)* | | ☑ | ☑ |
| *Between (inclusive)* | | ☑ | ☑ |
| *Not between (inclusive)* | | ☑ | ☑ |
| *Contains* | ☑ | | |
| *Does not contain* | ☑ | | |
| *Is missing (null)* | ☑ | ☑ | ☑ |
| *Is not missing (not null)* | ☑ | ☑ | ☑ |
| *Starts with* | ☑ | | |
| *Ends with* | ☑ | | |

문자열 비교의 경우, 대소문자를 구분하는 ☑ *Case sensitive* 옵션도 사용할 수 있습니다.

모든 검색 옵션을 설정한 다음, *Select features* 버튼을 눌러 일치하는 피처들을 선택할 수 있습니다. 드롭다운 옵션은 다음과 같습니다:

- *Select features*
- *Add to current selection*
- *Remove from current selection*
- *Filter current selection*

*Reset form* 버튼을 눌러 모든 검색 옵션을 초기화할 수도 있습니다.

조건을 설정하고 나면, 다음 가운데 하나를 할 수 있습니다:

- *Zoom to features*: 피처를 미리 선택하지 않고도 맵 캔버스에서 조건과 일치하는 피처로 확대/축소할 수 있습니다.
- *Flash features*: 조건과 일치하는 피처를 강조합니다. 선택 또는 식별 도구를 사용하지 않고도 피처를 식별할 수 있는 간편한 방법입니다. 강조 효과는 맵 캔버스 범위를 변경하지 않으며, 피처가 현재 맵 캔버스 경계 안에 있는 경우에만 강조가 가시화될 것이라는 점을 기억하십시오.

### 8.4.2 피처 식별

 Identify Features 도구는 맵 캔버스와의 쌍방향 액션을 통해 팝업창에서 피처 또는 픽셀에 대한 정보를 볼 수 있게 해줍니다. 이 도구를 사용하면 QGIS 가 지원하는 거의 모든 (벡터, 래스터, 메시, 점구름, WMS, WFS, …) 레이어 유형을 쿼리할 수 있습니다. 요소 하나를 식별하려면 다음 가운데 하나를 이용하십시오:

- *View* → *Identify Features* 메뉴
- `Ctrl+Shift+I` (또는 `Cmd+Shift+I`) 키 조합
- *Attributes* 툴바에 있는  Identify Features 버튼

그 다음 활성 레이어의 피처 또는 픽셀 하나를 클릭하십시오. 맵 캔버스에서 식별된 항목을 강조시키는 동시에 해당 항목에 대한 상세 정보를 볼 수 있는 *Identify Results* 대화창이 열립니다. 이 대화창에는 고급 환경설정을 위한 일련의 버튼들도 있습니다.

#### 식별 결과 대화창

아래에서 위로:

- *View* 는 대화창의 일반적인 측면과 결과물의 유형을 제어합니다. 다음 가운데 하나로 설정할 수 있습니다:
  - **Tree**: this is the default view, and returns the results in a tree-structure where the first item is the name of the layer and its children are its identified element(s).
  - **Table**: 래스터 기반 레이어 전용으로, 결과물을 `Layer`, `FID`, `Attribute` 및 `Value` 열을 가진 테이블로 볼 수 있게 해줍니다.
  - **Graph**: 래스터 기반 레이어 전용입니다.
- *Mode* 는 결과물을 반환할 수 있는 레이어를 선택할 수 있게 해줍니다. 이런 레이어는 맵 캔버스에서 데이터를 볼 수 있도록 가시화 설정이 되어 있어야 하는데, *Project properties* → *Data Sources* → *Layers capabilities* 메뉴에서식별 가능 하도록 설정해줘야 합니다. 다음과 같은 모드를 사용할 수 있습니다:
  - **Current layer**: *Layers* 패널에서 선택한 레이어 (들) 만 결과를 반환합니다. 그룹을 선택한 경우, 그룹의 지엽 (leaf) 레이어들로부터 결과를 고릅니다.
  - **Top down, stop at first**: 마우스 커서 아래 있는 레이어의 피처 또는 픽셀 가운데 가장 위에 있는 피처 또는 픽셀로부터 결과를 반환합니다.
  - **Top down**: 마우스 커서 아래 있는 레이어의 피처 또는 픽셀로부터 결과를 반환합니다.
  - **Layer selection**: 컨텍스트 메뉴를 열어서 사용자가 피처를 식별할 레이어를 선택할 수 있습니다. 마우스 커서 아래 있는 피처가 1 개뿐인 경우, 자동으로 결과를 표시할 것입니다.
- In the upper part, the information widget: when you identify a data in the map canvas, this is the place where the *Identify Results* dialog will list details about the clicked (or hovered over, depending on the tool in use) items. Their formatting relies on the *selected view*.

  The information displayed by the identify tool will depend on the type of layer you have selected, whether it is:
  - a *vector layer* (including vector tiles or point cloud data),
  - a *raster layer*,
  - a *mesh layer*.
- When in a tree view, the upper part also displays a handful of tools above the results:
  -  Open Form: 현재 피처의 양식을 엽니다.

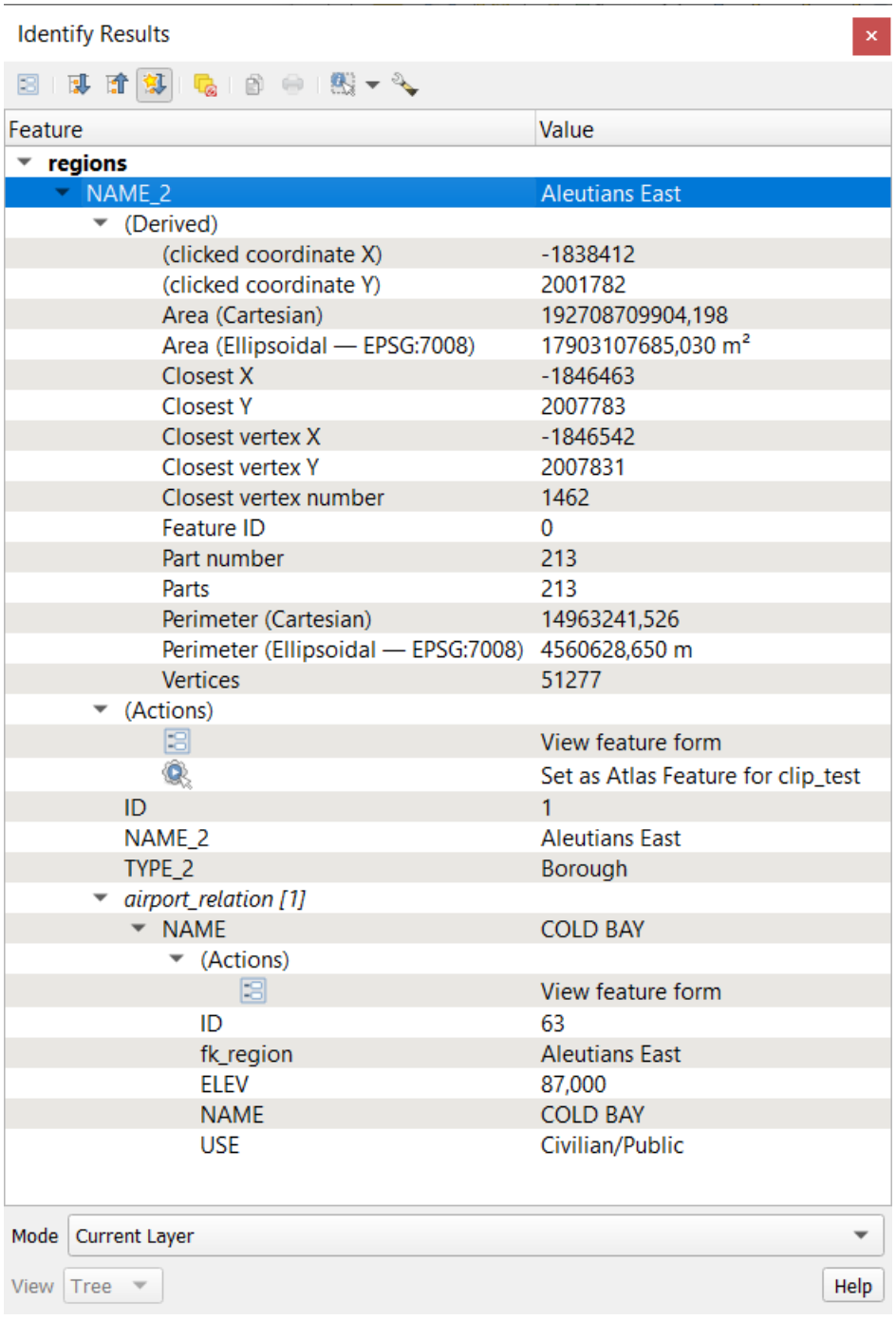



그림 8.11: 식별 결과 대화창

  -  Expand tree: 트리를 펼칩니다.
  -  Collapse tree: 트리를 접습니다.
  -  Expand New Results by Default: 다음 식별할 피처의 정보를 접을지 펼칠지 여부를 정의합니다.
  -  Clear Results: 결과를 초기화합니다.
  -  Copy the identified feature to clipboard: 식별된 피처 정보를 클립보드로 복사하기 때문에, 스프레드시트에 붙여넣는 데 적합합니다.
  -  Print selected HTML response: 결과물의 텍스트 기반 서식을 용지에 인쇄하거나 `.PDF` 파일로 저장합니다.
  - 대화형 식별 도구: 맵 캔버스에서 식별할 피처 또는 픽셀을 선택하기 위한 도구들의 드롭다운 메뉴입니다.
  -  Identify Settings 아래에서 다음을 활성화할 수 있습니다:
    *  *Auto open form for single feature results*: 이 옵션을 체크하면, 피처를 하나 식별할 때마다 해당 피처의 속성을 보여주는 양식이 열립니다. 피처 속성을 빠르게 편집할 수 있는 편리한 방법입니다.
    *  *Hide derived attributes from results*: 이 옵션을 체크하면, 레이어에 실제로 정의된 필드만 보여줍니다.
    *  *Hide NULL values from results*: 이 옵션을 체크하면, 결과에 NULL 값을 보여주지 않습니다.
    *  *Show Relations* to show relations in the vector layer *Identify results tree*
  -  Help: 현재 문서에 접근할 수 있습니다.

### 피처 식별 도구 이용하기

기본 보기 (*View: Tree*) 상태에서, *Identify Results* 패널은 쿼리할 레이어와 대화형 작업을 할 수 있는 몇몇 도구를 제공합니다. 대상 레이어 선택기 와 함께 이 도구들을 영리하게 조합하면 식별 작업을 크게 향상시킬 수 있습니다:

-  Identify Feature(s): 피처 (들) 를 클릭 한 번 또는 클릭 & 드래그로 식별합니다.
  - 클릭 한 번 또는 클릭 & 드래그: 대상 레이어에서 중첩하는 피처를 반환합니다.
  - 오른쪽 클릭: 컨텍스트 레이어에 대상 레이어에서 중첩하는 피처를 레이어 별로 그룹화시켜 목록화합니다. 그러면 다음 작업을 선택할 수 있습니다:
    * 특정 피처의 결과물을 볼 수 있습니다.
    * 특정 레이어에 있는 모든 피처의 결과물을 볼 수 있습니다.
    * 벡터 레이어의 경우, 반환된 피처로 필터링된 속성 테이블을 열 수도 있습니다.
    * 또는 반환된 피처를 모두 볼 수 있습니다.
-  Identify Features on Mouse over: 맵 캔버스 상에서 마우스를 옮겨 대상 레이어에 있는 항목 위로 가져가면 해당 항목을 강조하고 결과물 패널에 반환합니다.
-  Identify Features by Polygon: 사용자가 그린 또는 선택한 폴리곤과 중첩하는 항목을 반환합니다.
  - 폴리곤을 (왼쪽 클릭으로 포인트를 추가하고 오른쪽 클릭으로 폴리곤을 닫아서) 그리면, 대상 레이어에서 해당 폴리곤과 중첩하는 피처를 모두 강조하고 결과물 패널에 반환합니다.

  - 오른쪽 클릭하면 프로젝트에서 클릭한 위치 아래 있는 모든 가시 폴리곤 피처의 목록을 볼 수 있습니다. 항목을 하나 고르면 QGIS 가 대상 레이어에서 선택한 폴리곤과 중첩하는 피처를 모두 반환할 것입니다.

-  Identify Features by Freehand: 사용자가 자유롭게 그린 폴리곤과 중첩하는 항목을 반환합니다. 폴리곤을 (왼쪽 클릭으로 시작해서 커서를 이동시켜 면의 형태를 잡고 오른쪽 클릭으로 폴리곤을 닫아서) 그리십시오. 대상 레이어에서 중첩하는 모든 피처를 강조하고 결과물 패널에 반환합니다.

-  Identify Features by Radius returns items overlapping a drawn circle. Draw a circle (left-click to indicate the center point, move the pointer to shape the area or enter the radius in the pop-up text box and left-click or press `Enter` to validate the circle). All the overlaying features from target layers are highlighted and returned in the results panel.

> **팁**
>
> **쿼리할 레이어를 피처 식별 도구로 필터링하기**
>
> *Project* → *Properties…* → *Data Sources* → *Layer Capabilities* 메뉴에서 레이어 옆에 있는 *Identifiable* 열을 체크 해제하면,  Identify Features 도구를 사용할 때 해당 레이어를 쿼리 대상에서 제외합니다. 사용자가 관심을 가진 레이어에서만 피처를 반환받을 수 있는 간편한 방법입니다.

### 결과물의 컨텍스트 메뉴

Other functions can be found in the context menu of the identified item. For example, depending on the layer type, you can:

- interact with the individual identified item:
  - *View feature form* or *Edit feature form*, for vector layers
  - *Zoom to feature*: Zooms the map canvas to the extent of the identified feature or pixel
  - *Identify feature*: Restarts and recenters the search results on a *referenced or referencing vector feature* of the identified feature. Only available if  *Show Relations* is checked in Identify Settings.
  - *Copy feature*: Copies the vector geometry and attributes, or the pixel geometry
  - *Toggle feature selection*: Adds or removes identified feature to/from the selection
  - *Copy attribute value*: Copies the value in the *Value* column of the tree item you click on
  - *Copy feature attributes*: Copies the attribute names and values of the identified item
  - *Select features by attribute value*: Selects all vector features in the layer that match the selected attribute value
- control the rendering of the layers, the results panel or the map canvas:
  - *Clear result*: Removes results in the window
  - *Clear highlights*: Removes items highlighted on the map
  - *Highlight all*: Highlights in the map canvas all the items identified in the results dialog, regardless of their layer
  - *Highlight layer*: Highlights in the map canvas all the items identified in the layer of the selected entry
  - *Activate layer*: Selects the layer of the identified item in the *Layers* panel
  - *Layer properties*: Opens the layer properties window of the identified item
  - *Expand all* or *Collapse all* the results tree

## 8.5 레이어 속성 저장 및 공유

### 8.5.1 사용자 지정 스타일 관리하기

맵 캔버스에 벡터 레이어를 추가하면, QGIS 는 레이어 피처를 렌더링하는 데 기본값으로 랜덤한 심볼/색상을 이용합니다. 하지만 *Project* → *Properties…* → *Default styles* 에서 기본 심볼을 설정해서, 새로 추가되는 레이어마다 그 도형 유형에 따라 적용시킬 수 있습니다.

그러나 대부분의 경우, 레이어에 자동적으로 또는 수동으로 (수고를 덜 들이고) 적용할 수 있는, 사용자가 지정한 좀 더 복잡한 스타일을 선호할 겁니다. 레이어 속성 대화창 하단에 있는 *Style* 메뉴를 통해 이런 스타일을 정의할 수 있습니다. 이 메뉴는 스타일을 생성하고, 불러오고, 관리할 수 있는 기능을 제공합니다.

A style stores various information set in the layer or project properties dialog to render or interact with the layer. This includes:

표 8.4: Components of stored style and their references

| Category | Vector | Raster |
|---|---|---|
| Layer Configuration | *Layers capabilities* | |
| | 표시 속성 | 표시 속성 |
| Symbology | 심볼 속성 | 심볼 속성 ([1]) |
| 3D Symbology | *3* 차원 뷰 속성 | N/A |
| Labels | 라벨 속성 | N/A |
| Fields | 필드 속성 and 제약조건 | N/A |
| Attributes Form | 속성 양식 속성 | N/A |
| Actions | 액션 속성 | N/A |
| Map Tips | 표시 속성 | 표시 속성 |
| Diagrams | 도표 속성 | N/A |
| Attribute Table Configuration | 속성 테이블 작업 | |
| Rendering | 렌더링 속성 | 렌더링 속성 ([1]) |
| Custom Properties | 정보 속성 | 정보 속성 |
| Geometry Options | 디지타이즈 작업 속성 | N/A |
| Relations | 여러 레이어 간 관계 설정하기 | N/A |
| Temporal Properties | 시계열 속성 | 시계열 속성 |
| Legend Settings | 범례 속성 | N/A |
| Elevation Properties | 표고 속성 | 표고 속성 |
| Notes | 레이어 노트 | |

불러온 레이어에 적용되는 스타일의 기본 이름은 `default` 입니다. 사용자 레이어에 대해 이상적이고 적절한 렌더링 옵션을 설정하고 나면, *Style* 콤보박스를 클릭해서 다음 저장 옵션 가운데 하나를 선택할 수 있습니다:

- **Rename Current**: 활성화된 스타일을 재명명하고 현재 옵션으로 업데이트합니다.

[1] For raster layers, symbology and rendering items are melted together, meaning that saving/loading one would also save/load items of/from the other.

- **Add**: 현재 옵션으로 새 스타일을 생성합니다. 기본적으로 QGIS 프로젝트 파일 안에 저장될 겁니다. 스타일을 다른 파일로 또는 데이터베이스에 저장하는 방법은 다음 항을 참조하세요.
- **Remove**: 해당 레이어에 하나 이상의 스타일이 정의돼 있는 경우, 원하지 않는 스타일을 삭제합니다.

스타일 드롭다운 목록 하단에 레이어에 대한 스타일 집합이 있는데, 활성화된 스타일은 체크 표시가 돼 있습니다.

사용자가 레이어 속성 대화창을 승인할 때마다, 활성화된 스타일이 사용자가 변경한 사항으로 업데이트된다는 점을 기억하십시오.

레이어 하나에 대해 원하는 만큼 많은 스타일을 생성할 수 있지만, 한 번에 스타일 하나만 활성화할 수있습니다. 맵 테마 와 결합하면, 맵 범례에서 어떤 레이어도 복사할 필요가 없이 복잡한 프로젝트를 관리할 수 있는 빠르고 강력한 방법을 얻게 됩니다.

**참고**

사용자가 레이어 속성을 수정할 때마다 활성화된 스타일에 변경 사항을 저장한다는 사실은, 사용자가맵 테마 에서 사용되는 스타일을 실수로 변경하는 일 없이 언제나 올바른 스타일을 편집하고 있다는 점을 보장합니다.

**팁**

**레이어 컨텍스트 메뉴에서 스타일 관리하기**

Right-click on the layer in the *Layers* panel to copy, paste, add or rename layer styles. For both vector and raster layers, you can choose which style categories to copy/paste. Choose *All Style Categories* or one of the available categories, depending on the layer type. Note that for raster layers, the *Symbology* and *Rendering* categories are always copied together.

### 8.5.2 스타일을 파일 또는 데이터베이스로 저장하기

기본적으로 프로젝트 파일에 *Style* 콤보박스에서 생성한 스타일을 저장하고, 프로젝트에 있는 레이어에서 레이어로 복사하고 붙여넣을 수 있지만, 다른 프로젝트에서 불러올 수 있도록 프로젝트 외부에 저장할 수도 있습니다.

#### 텍스트 파일로 저장

 *Style* → *Save Style* 메뉴를 클릭하면, 스타일을 다음 텍스트 파일로 저장할 수 있습니다:

- QGIS 레이어 스타일 파일 (`.qml`)
- 벡터 레이어만 지원하는 SLD 파일 (`.sld`)

파일 기반 유형 레이어 (.shp, .tab 등등) 의 경우, *Save as Default* 메뉴가 레이어와 함께 (동일한 이름으로) `.qml` 파일을 생성합니다. 모든 렌더링 작업자 유형–단일 심볼, 범주, 등급 또는 규칙 기반–이 SLD 파일을 내보낼 수 있습니다. 그러나 SLD 를 가져오는 경우, 단일 심볼 또는 규칙 기반 렌더링 작업자 가운데 하나를 생성합니다. 즉 범주 또는 등급 스타일이 규칙 기반으로 변환된다는 뜻입니다. 범주 또는 등급 렌더링 작업자를 그대로 가져오려면 QML 포맷을 이용해야 합니다. 그런 반면, 스타일을 손쉽게 규칙 기반으로 변환하는 이런 방법이 매우 유용한 경우도 있습니다.

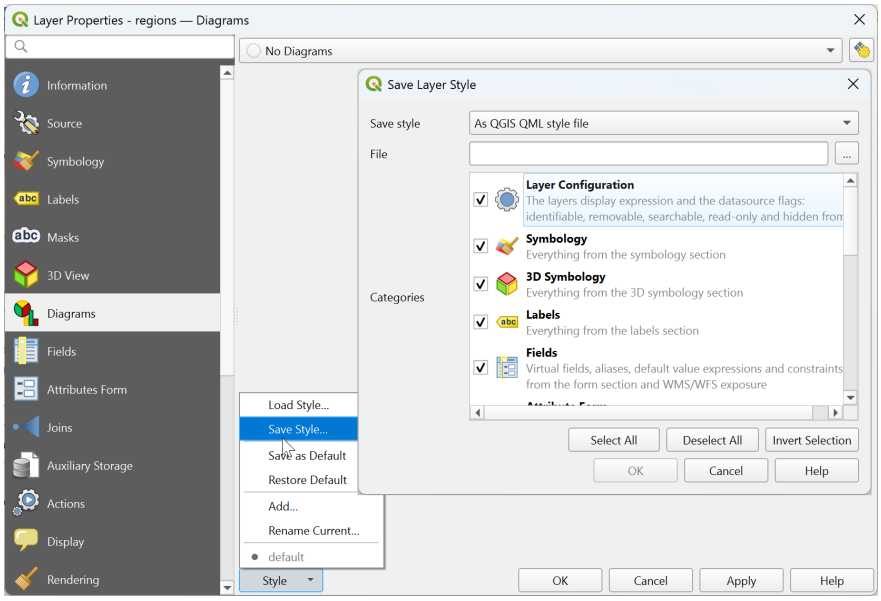



그림 8.12: 벡터 레이어 스타일 콤보박스 옵션들

### 데이터베이스에 저장

Vector layer styles can also be stored in a database if the layer datasource is a database provider. Supported formats are PostgreSQL, GeoPackage, SpatiaLite, MS SQL Server and Oracle. The layer style is saved inside a table (named `layer_styles`) in the database. Click on *Save Style…* → *Save in database* then fill in the dialog to define a style name, add a description, a `.ui` file if applicable and to check if the style should be the default style.

데이터베이스의 단일 테이블에 여러 스타일을 저장할 수 있습니다. 하지만 각 테이블은 오직 하나의 기본 스타일만 보유할 수 있습니다. 레이어 데이터베이스 또는사용자 프로파일 디렉터리에 있는 SQLite 로컬 데이터베이스인 `qgis.db` 에 기본 스타일을 저장할 수 있습니다.

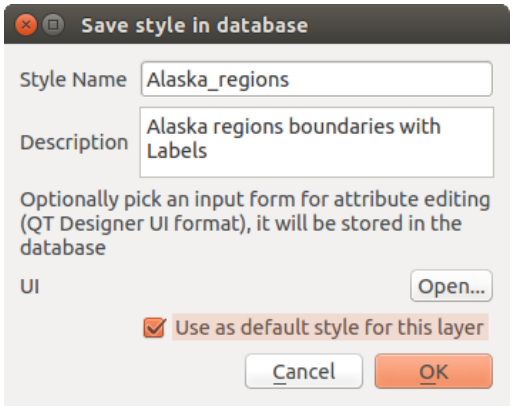



그림 8.13: 데이터베이스 대화창에서 스타일 저장

**팁**

**데이터베이스들 사이에서 스타일 파일 공유하기**

레이어가 데이터베이스 기반일 경우에만 해당 데이터베이스에 사용자 스타일을 저장할 수 있습니다.

데이터베이스 사이에서 (예를 들어 Oracle 레이어 스타일과 MS SQL 서버 스타일은) 호환되지 않습니다. 데이터베이스들 사이에서 스타일을 공유하려면 암호화되지 않은 텍스트 파일을 대신 이용하십시오.

**참고**

PostgreSQL 데이터베이스 백업에서 `layer_styles` 테이블을 복구하는 데 문제가 있을 수도 있습니다. 이런 문제를 해결하려면*QGIS layer_style 테이블 및 데이터베이스 백업* 을 참조하세요.

#### 스타일 불러오기

QGIS 에 레이어를 불러올 때, 해당 레이어의 기본 스타일이 이미 존재하는 경우, QGIS 는 해당 레이어를 레이어 기본 스타일로 불러옵니다. 또 *Style* ▸ *Restore Default* 메뉴 옵션을 클릭하면 기본 스타일 파일을 찾아 불러와서 해당 레이어의 현재 스타일을 교체합니다.

*Style* ▸ *Load Style* 메뉴를 사용하면 레이어에 저장된 모든 스타일을 적용할 수 있습니다. 텍스트 파일 스타일 (`.sld` 또는 `.qml`) 은 레이어 유형에 상관없이 적용할 수 있는 반면, 데이터베이스에 저장된 스타일은 레이어가 동일 데이터베이스 기반이거나, QGIS 로컬 데이터베이스에 저장된 스타일인 경우에만 불러올 수 있습니다.

*Database Styles Manager* 대화창은 데이터베이스에 있는 레이어 관련 스타일은 물론 저장돼 있는 다른 모든 스타일의 목록을 이름 및 설명과 함께 표시합니다.

**팁**

**프로젝트 내부에서 손쉽게 레이어 스타일 공유**

프로젝트 내부에서 파일 또는 데이터베이스 스타일을 가져오지 않고서도 레이어 스타일을 공유할 수 있습니다. *Layers Panel* 에 있는 레이어를 오른쪽 클릭한 다음 *Styles* 콤보박스에서 레이어의 스타일을 복사해서 그룹 또는 레이어 집합으로 붙여 넣을 수 있습니다. 이렇게 하면 원본 레이어와 동일한 (벡터 또는 래스터) 유형인 모든 레이어에, 그리고 벡터 레이어인 경우 모든 동일한 도형 유형 (포인트, 라인, 폴리곤) 에 스타일을 적용합니다.

### 8.5.3 레이어 정의 파일

활성화된 레이어의 컨텍스트 메뉴에서 *Export* ▸ *Save As Layer Definition File*…을 선택하면 레이어 정의를 `Layer Definition File` (`.qlr`) 로 저장할 수 있습니다. 레이어 정의 파일 (`.qlr`) 은 레이어의 데이터소스를 가리키는 참조 정보와 레이어 스타일을 담고 있습니다. `.qlr` 파일은 탐색기 패널에 표시되며, 레이어 패널에 레이어를 (저장된 스타일과 함께) 추가하는 데 사용할 수 있습니다. 시스템 파일 관리자에서 맵 캔버스로 `.qlr` 파일을 드래그 & 드롭할 수도 있습니다.

## 8.6 사용자 데이터를 문서로 작성

QGIS 에서 레이어 데이터를 표시하고 심볼화하는 것은 물론, 다음을 입력할 수도 있습니다:

- **metadata**: information to help people find and understand the dataset, how they can access and use it…these are properties of the datasource and can live out of the QGIS project. Metadata can also be defined for the project file itself.
- **노트**: 현재 프로젝트의 레이어와 관련된 지시 사항 및 코멘트입니다.

### 8.6.1 메타데이터

In the project or layer properties dialog, the *Metadata* tab provides you with options to create and edit a metadata report on the element. Information to fill concern:

- *Identification*: 데이터셋의 데이터 기본 속성 (상위 데이터셋, 식별자, 제목, 요약, 언어 등등)
- *Categories*: 데이터가 속한 카테고리. **ISO(국제 표준화 기구)** 카테고리 외에, 사용자가 사용자 지정 카테고리를 추가할 수 있습니다.
- *Keywords*: 데이터 및 관련 개념을 검색하기 위한, 표준 기반 어휘를 준수하는 키워드
- *Access*: 데이터셋에 접근하기 위한 정보 (사용 허가, 권한, 비용 및 제약 사항)
- *Extent*: 데이터셋의 공간 범위 (좌표계, 맵 범위, 고도) 또는 시계열 범위
- *Contact*: 데이터셋 소유자 (들) 의 연락처 정보
- *Links*: 보조 리소스 및 관련 정보를 가리키는 링크
- *History*: 데이터셋 이력

입력한 정보들의 요약이 *Validation* 탭에 표시되어, 이 양식과 관련된 잠재적인 문제점들을 식별할 수 있게 해줍니다. 확인 후 수정하거나 무시하면 됩니다.

메타데이터는 기본적으로 프로젝트 파일에 저장되며, *Metadata* 드롭다운 메뉴에 메타데이터를 `.qmd` 파일로 저장/불러오기하기 위한 그리고"기본"위치에 저장/불러오기 위한 옵션들이 있습니다.

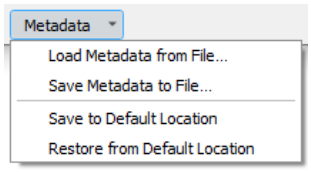



그림 8.14: 메타데이터 저장/불러오기 옵션들

*Save to Default Location* 및 *Restore from Default Location* 에 쓰이는"기본"위치는 기저 데이터소스와 그 환경설정에 따라 달라집니다.

- PostgreSQL 데이터소스의 경우 *Allow saving/loading QGIS layer metadata in the database* 환경설정 옵션이 체크되어 있다면 데이터베이스의 전용 테이블 안에 메타데이터를 저장합니다.
- GeoPackage 데이터소스의 경우 *Save to Default Location* 메뉴 항목이 항상 GeoPackage 의 내부 메타데이터 테이블에 메타데이터를 저장합니다.

  메타데이터가 PostgreSQL 또는 GeoPackage 내부 테이블에 저장되는 경우 브라우저 및레이어 메타데이터 검색 패널 에서 메타데이터를 검색하고 필터링할 수 있습니다.
- 다른 모든 파일 기반 데이터소스의 경우 *Save to Default Location* 메뉴 항목이 데이터소스 파일 옆에 메타데이터를 `.qmd` 파일로 저장합니다.
- 그 이외의 모든 경우, *Save to Default Location* 메뉴 항목은 로컬 `.sqlite` 데이터베이스에 메타데이터를 저장합니다.

### 8.6.2 레이어 노트

레이어 노트를 통해 현재 프로젝트 안에 있는 레이어를 문서로 작성할 수 있습니다. 할 일 목록, 지시 사항, 경고 등등과 같은 프로젝트 사용자를 위한 중요한 메시지를 노트에 저장할 수 있습니다.

*Layers* 패널에서, 레이어의 컨텍스트 메뉴로부터 *Add layer notes*…항목을 선택하면 열리는 대화창에서 필요한 텍스트를 입력하십시오.

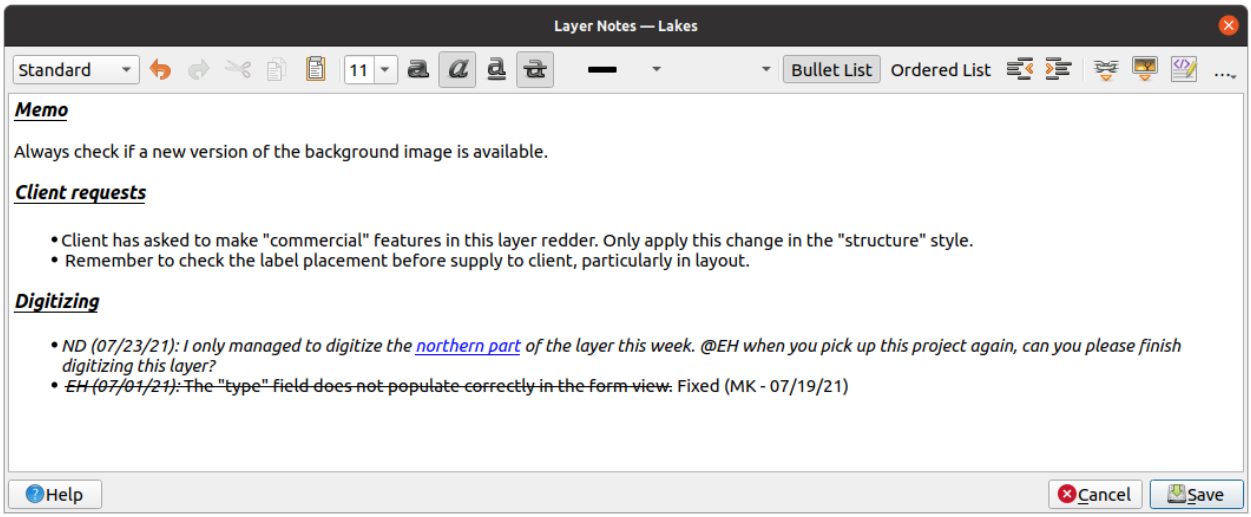



그림 8.15: 레이어에 노트를 추가하기

*Add layer notes* 대화창은 HTML 기반 여러 줄 (multiline) 텍스트 상자로, 다음을 위한 완전한 도구 집합을 제공합니다:

- 텍스트 처리: 잘라내기, 복사, 붙여넣기, 취소, 재실행
- 콘텐츠 전체 또는 일부에 적용되는 문자 서식: 글꼴 크기 및 색상, 굵은 글씨체, 기울인 글씨체, 밑줄, 취소선, 배경색, URL 강조
- 문단 구조: 글머리 기호 (bullet) 목록, 번호 매기기 목록, 들여쓰기, 사전 정의된 머리글
- 파일 삽입 (드래그 & 드롭으로도 가능)
- HTML 코딩으로 편집

툴바의 맨 오른쪽에 있는 …드롭다운 메뉴에서 다음 작업을 선택할 수 있습니다:

- *Remove all formatting*: 모든 서식을 제거
- *Remove character formatting*: 문자 서식을 제거
- *Clear all content*: 모든 내용을 지우기

*Layers* 패널에서 노트를 가진 레이어 옆에는 아이콘이 붙는데, 이 아이콘 위에 마우스 커서를 가져가면 노트를 표시합니다. 이 아이콘을 클릭하면 노트를 편집할 수 있습니다. 물론 레이어를 오른쪽 클릭해서 레이어 노트를 편집하거나 (*Edit layer note*…) 제거할 수 있습니다 (*Remove layer note*).

> **참고**
>
> 노트는레이어 스타일 의 일부분이기 때문에 `.qml` 또는 `.qlr` 파일로 저장할 수 있습니다. 또한 레이어 스타일을 복사-붙여넣기함으로써 어떤 레이어에서 다른 레이어로 옮길 수도 있습니다.

## 8.7 값을 변수로 저장

QGIS 에서는 표현식에 반복적으로 쓰이는 (예를 들어 프로젝트 제목 또는 사용자 이름 같은) 유용한 값을 변수에 저장할 수 있습니다. 사용자는 응용 프로그램의 전체 수준, 프로젝트 수준, 레이어 수준, 공간 처리 모델 설계자 수준, 조판 수준, 그리고 조판 항목 수준에서 변수를 정의할 수 있습니다. CSS 의 종속적 규칙과 마찬가지로, 변수도 덮어쓸 수 있습니다. 예를 들면 프로젝트 수준 변수는 동일한 이름을 가진 모든 응용 프로그램 전체 수준 변수를 덮어쓸 것입니다. 이런 변수를 이용해서 텍스트 문자열을 작성하거나, 변수 앞에 @ 문자를 붙여 사용자 지정 표현식을 작성할 수 있습니다. 예를 들면 인쇄 조판 안에 다음과 같은 내용을 가진 라벨을 생성하면:

```
This map was made using QGIS [% @qgis_version %]. The project file for this
map is: [% @project_path %]
```

다음과 같은 라벨로 렌더링됩니다:

```
This map was made using QGIS 3.4.4-Madeira. The project file for this map is:
/gis/qgis-user-conference-2019.qgs
```

사전 설정된 읽기 전용 변수 이외에도, 앞에서 언급한 모든 수준에서 사용자 지정 변수를 정의할 수 있습니다. 다음 변수들을 관리할 수 있습니다:

- *Settings* ▸ *Options* 메뉴에서 **전체 수준 변수**
- *Project Properties* 대화창에서 **프로젝트 수준 변수** (프로젝트 속성 참조)
- *Layer Properties* 대화창에서 **벡터 레이어 수준 변수** (벡터 속성 대화창 참조)
- *Model Designer* 대화창에서 **모델 설계자 변수** (모델 설계자 참조)
- 인쇄 조판기의 *Layout* 패널에서 **조판기 수준 변수** (조판 패널 참조)
- 인쇄 조판기의 *Item Properties* 패널에서 **조판 항목 수준 변수** (조판기 항목 공통 옵션 참조)

편집 가능한 변수와 구별하기 위해, 읽기 전용 변수의 이름 및 값을 이탤릭체로 표시합니다. 다른 한편으로, 하위 수준 변수가 덮어쓰는 상위 수준 변수엔 취소선이 그어집니다.

> **참고**
>
> 나이얼 도슨 (Nyall Dawson) 의 블로그 게시글, QGIS 2.12 에서의 변수 탐구 1 편, 2 편 그리고 3 편 에서 더 자세한 정보와 예시들을 찾아볼 수 있습니다.

## 8.8 일반 위젯

사용자가 QGIS 에서 자주 사용해야 할 옵션이 몇 개 있습니다. QGIS 는 사용자 편의를 위해 다음에 소개하는 특별 위젯들을 제공합니다.

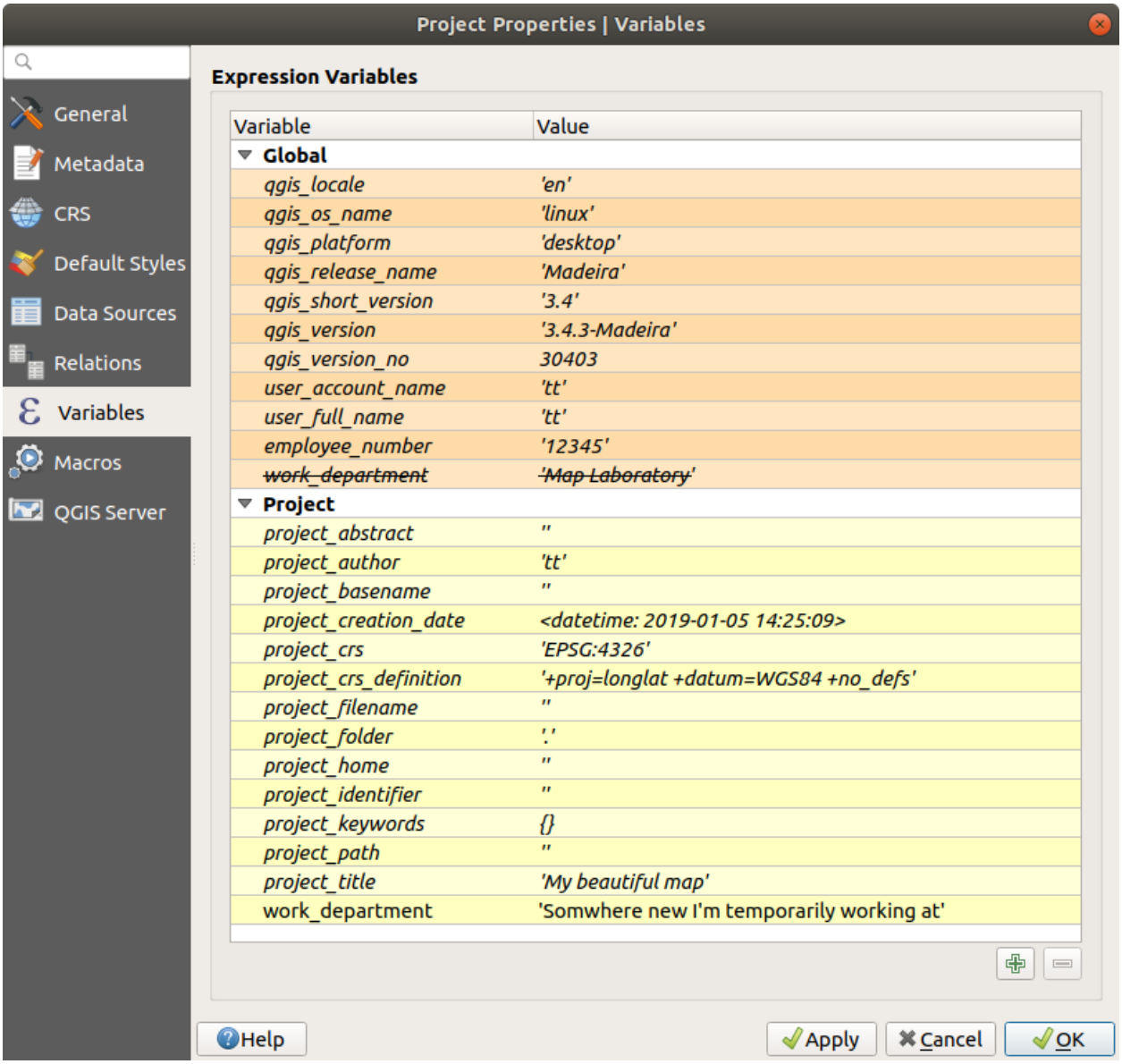



그림 8.16: 프로젝트 수준 변수 편집기

### 8.8.1 색상 선택기

#### 색상 대화창

사용자가 색상을 선택하기 위해  아이콘을 클릭할 때마다 *Select Color* 대화창이 열립니다. *Settings → Options… → General* 메뉴에 있는 *Use native color chooser dialogs* 파라미터 체크박스의 상태에 따라 이 대화창의 기능이 달라집니다. 체크된 상태라면, QGIS 가 실행 중인 OS 의 색상 대화창이 열립니다. 해제된 경우엔 QGIS 전용 색상 선택기가 열립니다.

전용 색상 선택기 대화창은 사용자가 색상을 선택할 수 있도록 Color ramp 색상표, Color wheel 색상환, Color swatches 색상 견본표 또는 Color picker 색상 추출기 탭 4 개를 보유하고 있습니다. 앞의 두 탭을 통해 가능한 모든 색상 조합을 탐색하고 항목에 사용자 선택 사항을 적용할 수 있습니다.

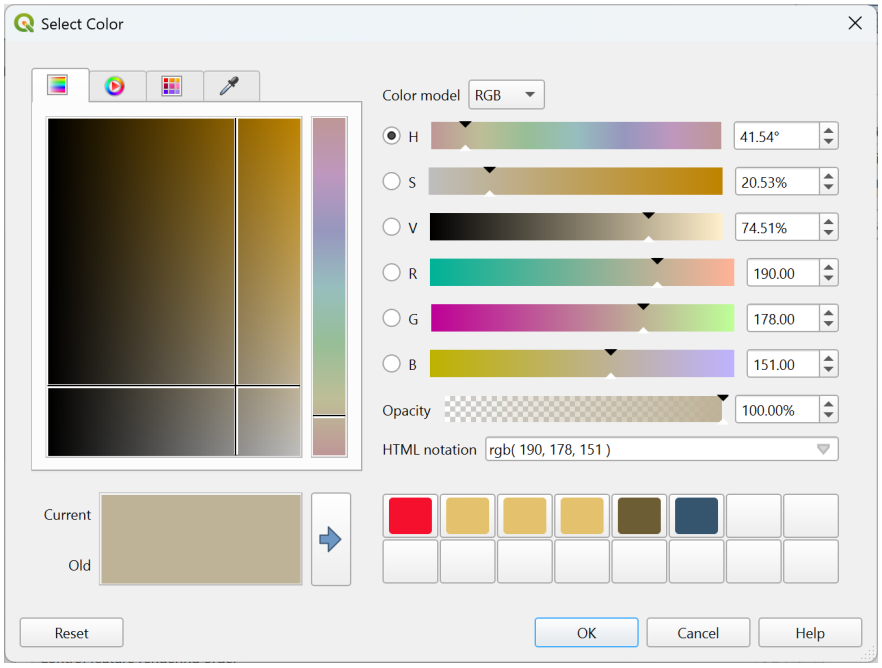



그림 8.17: 색상 선택기의 색상표 탭

Color swatches 탭에 있는 색상표 목록에서 색상을 선택할 수 있습니다. (자세한 내용은색상 설정 을 참조하세요.) 프레임 하단에 있는 Add current color 및 Remove selected color 버튼으로 *Recent colors* 색상표를 제외한 모든 색상표를 수정할 수 있습니다.

색상표 콤보박스 옆에있는 …버튼을 클릭하면 다음 옵션들을 선택할 수 있습니다:

- 색상을 복사, 붙여넣기, 불러오기 또는 내보내기
- 색상표를 생성, 불러오기 또는 제거하기
- *Show in Color Buttons* 옵션을 활성화시키면 색상 선택기 위젯 (그림 8.19 참조) 에 사용자 지정 색상표를 추가합니다.

사용자가 QGIS UI 의 어디에서라도 또는 다른 응용 프로그램에서라도 마우스 커서 아래에 있는 색상 표본을 추출할 수 있는 Color picker 색상 추출기를 이용하는 옵션도 있습니다. 탭이 활성화된 상태에서 스페이스 바를 누르고, 원하는 색상 위로 마우스를 가져가서 클릭하거나 스페이스 바를 다시 누르십시오. *Sample Color* 버튼을 눌러도 색상 추출기가 활성화됩니다.

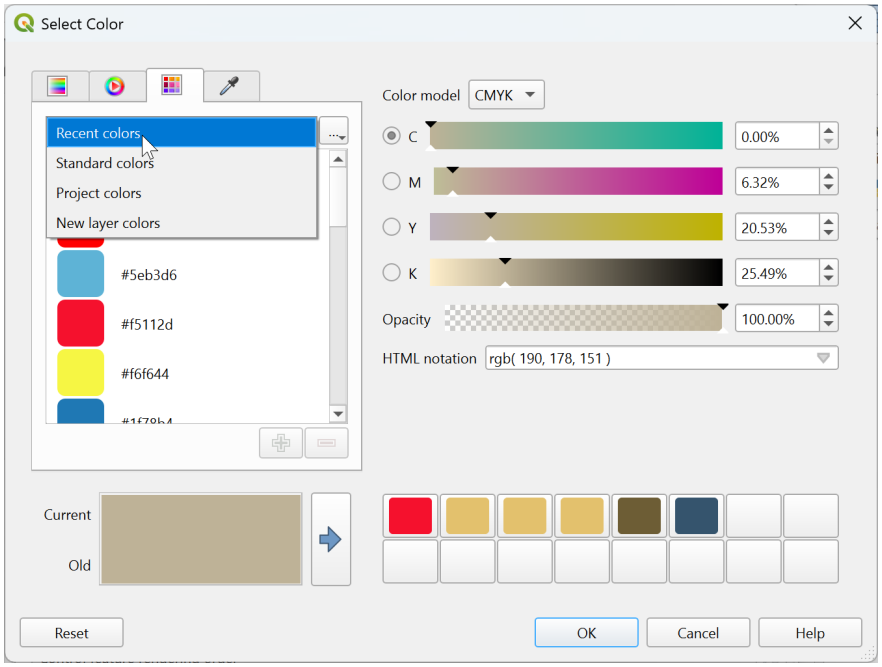



그림 8.18: 색상 선택기의 색상 견본표 탭

Whatever method you use, the selected color is always described through options of a *Color model* that defaults to the *project color model*:

- *RGB*: color is set using sliders for `HSV` (Hue, Saturation, Value) or `RGB` (Red, Green, Blue) values.
- *CMYK* (Cyan, Magenta, Yellow, Black)

색상환 (color wheel) 이나 색상표 (color ramp) 또는 모든 색상 파라미터 슬라이드 바를 클릭하는 것만으로도 쉽게 색상을 조정할 수 있습니다. 색상 파라미터들 옆에 있는 스핀박스를 통해 또는 해당 슬라이드 바에 커서를 가져가서 마우스 휠을 돌리면 이런 파라미터들을 변경할 수 있습니다. *HTML notation* 에 색상을 입력할 수도 있습니다. 마지막으로, 투명도 수준을 설정하기 위한 *Opacity* 슬라이드 바가 있습니다.

이 대화창에서 (오브젝트에 적용된) *Old* 색상과 (선택한) *Current* 색상을 시각적으로 비교해볼 수도 있습니다.

이렇게 선택한 색상을 손쉽게 쓸 수 있도록 슬롯에 드래그 & 드롭해서 또는 Add color to swatch 버튼을 클릭해서 저장할 수 있습니다.

> **팁**
>
> **빠른 색상 조정**
>
> 색상 선택기 위젯을 다른 색상 선택기 위젯으로 드래그 & 드롭하면 해당 위젯의 색상을 적용합니다.

### 색상 드롭다운 단축키

 색상 버튼 오른쪽에 있는 드롭다운 화살표를 클릭하면 빠른 색상 선택을 위한 위젯을 표시합니다. 이 단축키로 다음 기능에 접근할 수 있습니다:

- 색상을 선택하기 위한 색상환
- 색상 투명도를 변경하기 위한 알파값 슬라이드 바
- 이전에 *Show in Color Buttons* 에 설정된 색상표
- 현재 색상을 복사해서 다른 위젯으로 붙여넣기
- 사용자의 컴퓨터 스크린 어느 곳에서든 색상 선택
- 색상 선택기 대화창에서 색상 선택
- 빠른 색상 조정을 위해 한 위젯에서 다른 위젯으로 색상을 드래그 & 드롭

> **팁**
>
> 색상 선택기 위젯 위에서 마우스 휠을 돌리면 관련 색상의 불투명도를 빠르게 조정할 수 있습니다.

> **참고**
>
> 데이터 정의 무시 속성을 통해 색상 위젯이프로젝트 색상 으로 설정돼 있는 경우, 앞에서 설명한 색상 변경 기능들을 사용할 수 없습니다. 색상을 변경하려면 먼저 *Unlink color* 로 색상을 해제하거나 *Clear* 로 색상 정의를 초기화해야 합니다.

### 색상표 드롭다운 단축키

색상표는 하나 이상의 피처에 색상 집합을 적용할 수 있는 실용적인 방법입니다. 색상표 설정 절에서 색상표를 생성하는 방법을 설명하고 있습니다. 색상의 경우,  색상표 버튼을 클릭하면 그에 대응하는 유형의 색상표 대화창이 열려 그 속성을 변경할 수 있습니다.

옵션 오른쪽에 있는 드롭다운 메뉴를 통해 다음과 같은 더 다양한 색상 모음 및 옵션에 빠르게 접근할 수 있습니다:

- *Invert Color Ramp*: 반전 색상표
- *Clear Current Ramp*: 위젯에 할당된 모든 색상표를 설정 해제 (몇몇 맥락에서만 사용할 수 있습니다.)
- *Random Colors*: 몇몇 맥락에서만 (예를 들어 레이어 심볼에 색상표를 사용하는 경우) 사용할 수 있는 이 항목을 체크하면 임의의 색상을 가진 색상표를 생성하고 적용합니다. 현재 색상표가 만족스럽지 않은 경우 새로운 임의 색상표를 재생성하는 *Shuffle random colors* 메뉴 항목도 활성화시킵니다.
- *Style Manager* 대화창에서 **Favorites** 로 표시된 `gradient` 또는 `catalog: cpt-city` 색상표의 미리보기
- *All Color Ramps*: 호환 가능한 색상표 데이터베이스에 접근
- *Create New Color Ramp*…: 현재 위젯에서 사용할 수 있는 모든 지원 색상표 유형을 생성 (이 색상표를 라이브러리에 저장하지 않으면 현재 위젯에서밖에 사용할 수 없다는 사실을 기억하십시오.)
- *Edit Color Ramp*…: 전체 색상표 버튼을 클릭하는 것과 동일
- *Save Color Ramp*…: 스타일 라이브러리에 현재 색상표를 사용자 지정 사항과 함께 저장

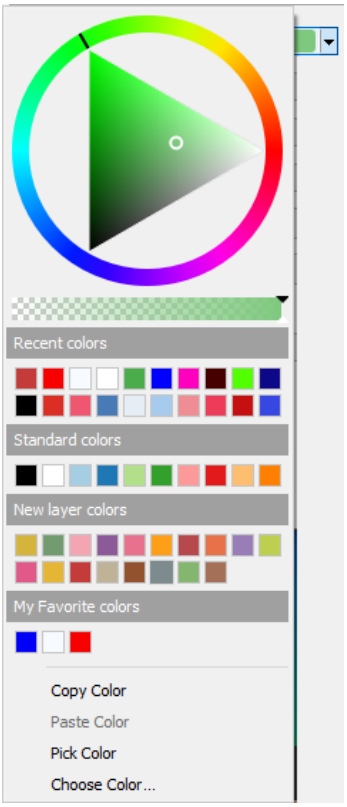



그림 8.19: 빠른 색상 선택기 메뉴

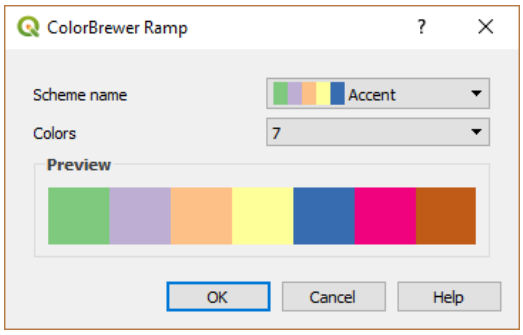



그림 8.20: 컬러브루어 (colorbrewer) 색상표 사용자 지정하기

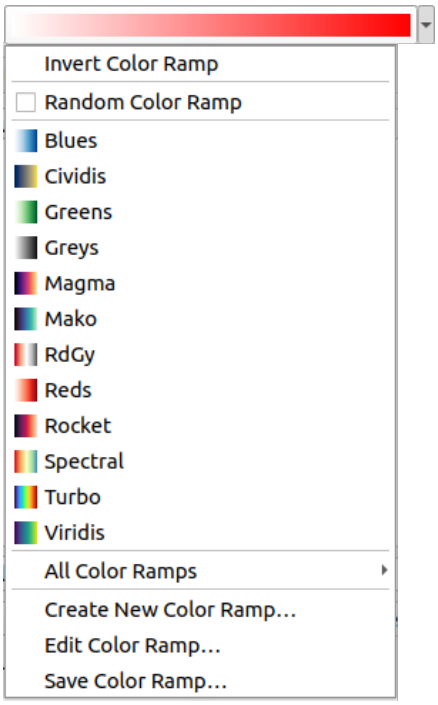



그림 8.21: 빠른 색상표 선택 위젯

### 8.8.2 심볼 위젯

*Symbol* 선택기 위젯은 피처의 심볼 속성을 설정하고자 할 때 편리한 단축키입니다. 드롭다운 화살표를 클릭하면색상 드롭다운 위젯 의 기능과 함께, 다음과 같은 심볼 옵션이 표시됩니다:

- *Configure Symbol*…: 심볼 선택기 위젯 버튼을 클릭하는 것과 동일합니다. 이 메뉴는심볼 파라미터 를 설정하는 대화창을 엽니다.
- *Copy Symbol*: 현재 항목의 심볼을 복사합니다.
- *Paste Symbol*: 현재 항목에 클립보드의 심볼을 붙여넣습니다. 심볼을 빠르게 환경설정할 수 있습니다.
- *Clear Current Symbol*: 위젯에 할당된 모든 심볼을 설정 해제 (몇몇 맥락에서만 사용할 수 있습니다.)

> **팁**
>
> 마커 또는 라인 심볼 위젯 위에서 마우스 휠을 돌리면 관련 심볼의 크기를 빠르게 조정할 수 있습니다.

### 8.8.3 원격 또는 내장 파일 선택기

파일 선택기 위젯과 함께 …버튼 옆에 드롭다운 화살표가 표시되는 경우가 가끔 있습니다. 일반적으로 다음을 사용할 때 이 드롭다운 화살표가 표시됩니다:

- 심볼 또는 라벨에 SVG 파일을 사용하는 경우
- 심볼, 라벨, 텍스처, 또는 장식을 사용자 지정하기 위해 래스터 이미지를 사용하는 경우

이 화살표를 누르면 다음 작업을 할 수 있는 메뉴가 표시될 것입니다:

- *Select File…*: 파일 시스템으로부터 파일을 불러옵니다. 파일 경로를 통해 파일을 식별하며, QGIS 가 해당 이미지를 표시하려면 이 경로를 확인해야 합니다.
- *From URL…*: 원격 URL 로부터 파일을 불러옵니다. 위와 마찬가지로, 원격 리소스를 성공적으로 검색한 경우에만 이미지를 불러올 것입니다.
- *Embed File…*: 현재 프로젝트, 스타일 데이터베이스, 또는 인쇄 조판 템플릿 안에 파일을 내장시킵니다. 그러면 언제나 이 파일을 해당 항목의 일부로서 렌더링합니다. 사용자 지정 심볼을 가진 독립 프로젝트를 생성할 수 있는 편리한 방법으로, 서로 다른 사용자 및 QGIS 설치본 간에 쉽게 공유할 수 있습니다.
- *Extract Embedded File…*: 위젯에서 내장 파일을 추출해서 디스크 상에 저장합니다.

### 8.8.4 가시성 축척 선택기

가시성 축척 선택기는 맵 캔버스에 어떤 요소를 어떤 축척에서 가시화시킬지를 제어하는 옵션들을 제공합니다. 지정한 축척 범위를 벗어나면 해당 요소를 표시하지 않습니다. 예를 들면 레이어, 라벨, 또는 도표에 대해 각각의 *Rendering* 속성 탭에서 가시성 축척을 적용시킬 수 있습니다.

1. *Scale dependent visibility* 체크박스를 체크하십시오.
2. *Minimum (exclusive)* 란에 원하는 가장 축소된 축척을 입력합니다. 값을 직접 입력하거나사전 정의 축척으로부터 선택할 수 있습니다.
3. 그리고/또는 *Maximum (inclusive)* 란에 원하는 가장 확대된 축척을 입력하십시오.

   축척 입력란 옆에 있는 Set to current canvas scale 버튼을 클릭하면 가시성 범위의 경계값을 현재 맵 캔버스의 축척으로 설정합니다. 이 버튼 옆의 화살표를 누르면 조판의 맵 축척값을 사용할 수 있습니다.

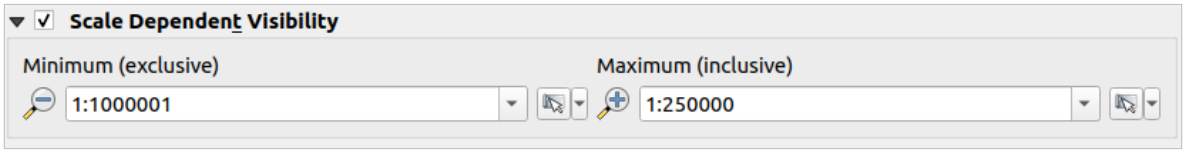



그림 8.22: 가시성 축척 선택기 위젯

### 8.8.5 공간 범위 선택기

*Extent* 선택기 위젯은 사용자가 레이어에 할당할, 또는 액션을 실행할 공간 범위를 선택하고자 할 때 편리하게 사용할 수 있는 지름길입니다. 그 맥락에 따라, 다음 가운데 하나를 선택할 수 있습니다:

- *Current Layer Extent*: 예를 들면 레이어를 내보내는 경우 현재 레이어 범위를 사용합니다.
- *Calculate from Layer*: 현재 프로젝트에 불러온 레이어의 범위를 사용합니다.
- *Map Canvas Extent*: 현재 맵 캔버스의 범위를 사용합니다.
- *Draw on Canvas*: 캔버스에 사각형을 그려 그 꼭짓점 좌표들을 사용합니다.
- *Calculate from Bookmark*: 저장되어 있는북마크 의 범위를 사용합니다.
- *Calculate from Layout Map*: 조판 맵 의 범위를 사용합니다.
- 좌표를 `xmin, xmax, ymin, ymax` 로 입력하거나 편집합니다.

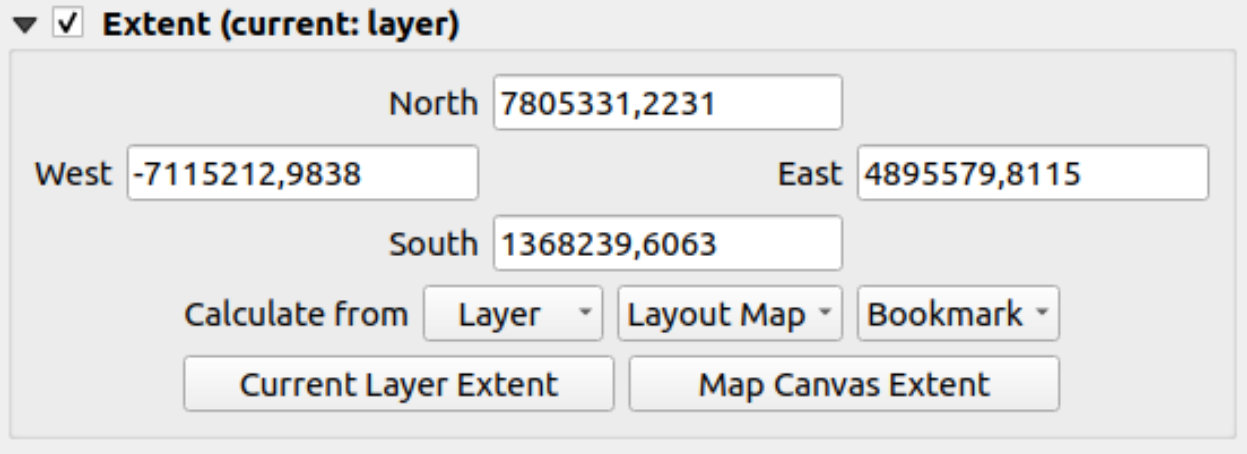



그림 8.23: 공간 범위 선택기 위젯

### 8.8.6 글꼴 선택기

*Font* 선택기 위젯은 텍스트 정보 (피처 라벨, 장식 라벨, 맵 범례 텍스트 등등) 의 글꼴 속성을 설정하고자 할 때 편리한 단축키입니다. 드롭다운 화살표를 클릭하면 다음과 같은 옵션의 일부 또는 전부가 표시됩니다:

- *Clear Current Text Format*: 위젯에 할당된 모든 텍스트 서식을 설정 해제 (몇몇 맥락에서만 사용할 수 있습니다.)
- *Font Size*: 관련 단위로 된 글꼴 크기
- *Recent Fonts*: (맨 위에) 활성화 글꼴이 체크되어 있는 최근 사용 글꼴 목록
- *Configure Format*…: 글꼴 선택기 위젯을 클릭하는 것과 동일합니다. 색상, 불투명도, 방향, HTML 태그, 버퍼, 배경, 음영 등등과 같은 고급 서식 옵션을 제공하는텍스트 서식 대화창을 엽니다.
- *Copy Format*: 텍스트의 서식을 복사합니다.
- *Paste Format*: 텍스트에 클립보드의 서식을 붙여넣습니다. 텍스트 서식을 빠르게 환경설정할 수 있습니다.
- 색상 위젯: 색상을 쉽게 설정할 수 있습니다.

> **팁**
>
> 글꼴 선택기 위젯 위에서 마우스 휠을 돌리면 관련 텍스트의 글꼴 크기를 빠르게 조정할 수 있습니다.

### 8.8.7 단위 선택기

QGIS 의 항목 (라벨, 심볼, 조판기 요소 등등) 들의 크기 속성이 꼭 프로젝트 단위 또는 특정 레이어의 단위일 필요는 없습니다. 광범위한 속성 집합의 경우, *Unit* 선택기 드롭다운 메뉴를 통해 사용자가 원하는 (스크린 해상도, 용지 크기, 또는 지형에 기반한) 렌더링에 따라 단위 값을 조정할 수 있습니다. 다음과 같은 단위를 사용할 수 있습니다:

- *Millimeters*
- *Points*
- *Pixels*
- *Inches*




그림 8.24: 글꼴 선택기 드롭다운 메뉴

- *Percentage*: 일부 속성을 다른 속성의 백분율로 설정할 수 있습니다. 예를 들면, 텍스트 서식의 경우 글꼴 크기가 변하는 대로 (버퍼 크기, 음영 반경 등등의) 구성 요소들의 크기가 고정된 채가 아니라 어울리게 변하도록 생성하는 데 유용합니다. 즉 사용자가 글꼴 크기를 변경할 때 이런 구성 요소들의 크기도 조정해야 할 필요가 없습니다.
- *Meters at Scale*: 이 단위를 선택하면 기저 맵 단위에 상관없이 (예를 들어 맵 단위가 인치, 피트, 또는 지리도 (geographic degree) 등이더라도) 항상 크기를 미터 단위로 설정할 수 있습니다. 현재 프로젝트의 타원체 설정을 기반으로 미터 단위의 크기를 계산하고, 현재 맵 범위의 중심에서 미터 단위의 거리 투영을 계산합니다. 투영 좌표계를 사용하는 맵의 경우 투영 단위를 이용해서 계산합니다. 지리도 (위도/경도) 기반 좌표계를 사용하는 맵의 경우 맵의 수직 축척 용 타원체 계산을 이용해서 대략적인 미터 단위 크기를 계산합니다.
- *Map Units*: 맵 뷰 축척에 따라 크기를 조정합니다. 이렇게 하면 너무 크거나 너무 작은 값이 될 수 있기 때문에, 입력란 옆에 있는 🔧 버튼을 사용해서 크기를 다음과 같은 기준을 기반으로 하는 값의 범위로 제한하십시오:
  - *Minimum scale* 및 *Maximum scale*: 두 축척 제한값 가운데 어느 한쪽에 도달할 때까지 맵 뷰 축척을 기반으로 값을 조정합니다. 축척이 제한 범위를 넘어서면, 가장 가까운 축척 제한에서의 값을 유지합니다.
  - 그리고/또는 mm 단위로 된 *Minimum size* 및 *Maximum size*: 두 축척 제한값 가운데 어느 한쪽에 도달할 때까지 맵 뷰 축척을 기반으로 값을 조정합니다. 축척이 제한 범위를 넘어서면 제한에서의 크기를 유지합니다.

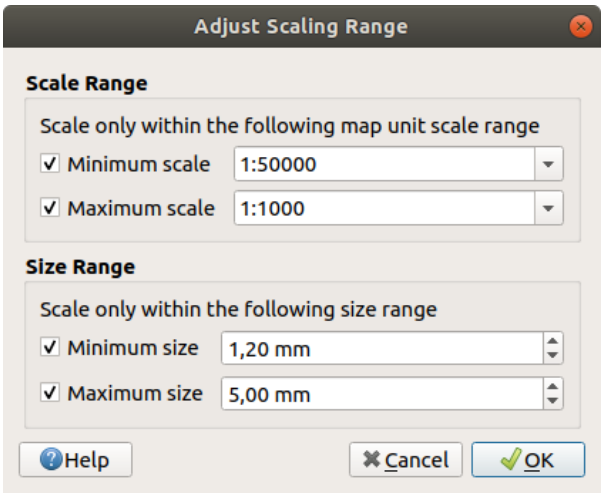



그림 8.25: 축척 범위 조정 대화창

### 8.8.8 숫자 서식 작업

Numeric formatters allow formatting of numeric values for display, using a variety of different formatting techniques (for instance scientific notation, currency values, percentage values, custom formatting, etc). One use of this is to set text in a layout scale bar or table, a color ramp legend, an elevation profile plot, ⋯

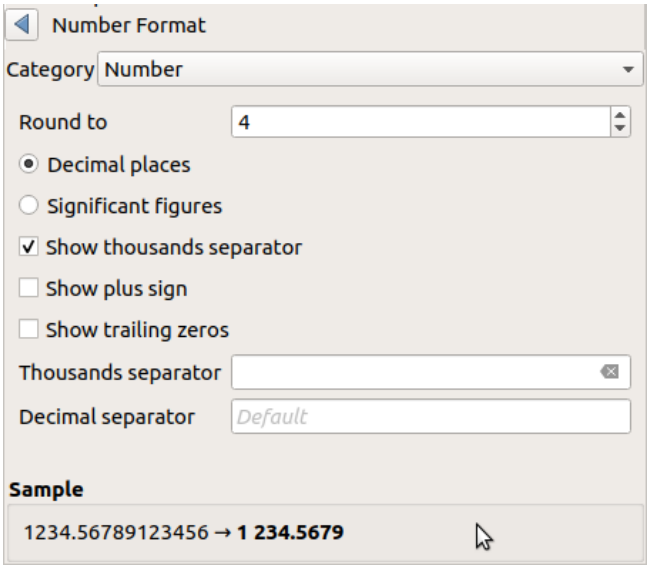



그림 8.26: 숫자값 서식 작업

서로 다른 서식 범주를 지원합니다. 대부분의 경우, 다음 숫자 옵션들의 일부 또는 전부를 설정할 수 있습니다:

- *Show thousands separator*: 천 단위 구분자를 활성화합니다.
- *Show plus sign*: 양수의 경우 플러스 표시를 추가합니다.
- *Show trailing zeros*: 소수점 오른쪽의 후행 0 을 표시합니다.

그러나 다음 범주들의 경우 사용자 지정 설정을 할 수도 있습니다:

- *General*: 기본 범주입니다. 어떤 설정도 할 수 없으며, 상위 위젯 속성에서 설정된 대로 또는 전체 수준 설정을 사용해서 숫자값을 표시합니다.
- *Custom expression*: allows you to craft a custom *QGIS expression* to format numbers. The expression can use the `@value` variable to retrieve the value to be formatted, and then use any standard QGIS expression function to format this as desired.
- *Number*
  - *Decimal places* 의 자체 정의된 자릿수대로, 또는 *Significant figures* 값대로 값을 반올림 *Round to* 할 수 있습니다.
  - *Thousands separator* 및 *Decimal separator* 를 사용자 지정할 수 있습니다.
- *Bearing*: 다음을 사용해서 방향/방위의 텍스트 표현을 설정합니다:
  - *Format*: 사용할 수 있는 값의 범위는 `0 to 180°, with E/W suffix`, `-180 to +180°` 및 `0 to 360°` 입니다.
  - *Decimal places*: 소수점 이하 자릿수를 설정합니다.
- *Currency*: 통화 가치의 텍스트 표현을 설정합니다.
  - *Prefix*
  - *Suffix*
  - *Decimal places*: 소수점 이하 자릿수를 설정합니다.
- *Fraction*: 십진수 (decimal value) 값의 대중적인 분수 표현을 설정합니다. (예: *0.5* 대신 *1/2*)
  - ☐ *Use unicode super/subscript*: 유니코드 위/아래 첨자를 표시합니다. (예: 1/2 대신 $^{1}/_{2}$)
  - ☐ *Use dedicated Unicode characters*: 전용 유니코드 문자를 사용합니다.
  - *Thousands separator*: 천 단위 구분자를 사용자 지정합니다.
- *Percentage*: 다음과 같은 설정과 함께 값에 `%` 를 추가합니다:
  - *Decimal places*: 소수점 이하 자릿수를 설정합니다.
  - *Scaling*: 실제 값이 이미 백분율로 표시된 값인지 (이때 값을 그대로 유지합니다) 또는 분수인지 (값을 변환합니다) 표시합니다.
- *Scientific*: `2.56e+03` 과 같은 형식의 과학적 표기법을 사용합니다. 소수점 이하 자릿수 *Decimal places* 를 설정할 수 있습니다.

*Sample* 부분에서 이런 설정들의 실시간 미리보기를 볼 수 있습니다.

### 8.8.9 혼합 모드

QGIS 는 사용자가 이전에 그래픽 프로그램으로밖에 몰랐을 특별한 렌더링 효과를 위한 여러 옵션과 함께 이 도구를 제공합니다. 레이어나 피처에 적용할 수 있지만, 인쇄 조판기 항목에도 다음 혼합 모드를 적용할 수 있습니다:

- *Normal*: 표준 혼합 모드로, 위에 있는 픽셀의 알파 채널을 이용해서 밑에 있는 픽셀과 혼합합니다. 색상은 섞이지 않습니다.
- *Lighten*: 위에 있는 픽셀과 밑에 있는 픽셀로부터 각 요소의 최대값을 선택합니다. 이 모드의 산출물은 경계가 도드라져 보기 안 좋은 경향이 있다는 사실을 주의하십시오.
- *Screen*: 원본의 밝은 픽셀은 산출물에 렌더링되는 반면, 어두운 픽셀은 렌더링되지 않습니다. 이 모드는 한 항목의 질감을 다른 항목과 섞는 데 가장 유용합니다. (예를 들어 어떤 레이어에 질감을 부여하는 데 음영기복도를 사용할 수 있습니다.)

- *Dodge*: 위에 있는 픽셀의 밝기를 바탕으로 밑에 있는 픽셀의 명도와 채도를 올립니다. 따라서 위에 있는 픽셀이 밝을수록 밑에 있는 픽셀의 채도와 명도가 증가합니다. 위에 있는 픽셀이 너무 밝지 않을 때 가장 유용합니다. 너무 밝을 경우 그 효과가 극단적일 것입니다.
- *Addition*: 이 혼합 모드는 한 항목의 픽셀값을 다른 항목의 픽셀값에 더합니다. (RGB 값이) 최대값을 초과하는 경우 하얀색으로 표시합니다. 피처를 강조하려 할 때 적당한 모드입니다.
- *Darken*: 위에 있는 픽셀과 밑에 있는 픽셀 요소 가운데 최저값을 유지합니다. Lighten 모드와 마찬가지로 보기에 안 좋은 경향이 있습니다.
- *Multiply*: 위에 있는 항목의 각 픽셀의 값을 밑에 있는 항목의 픽셀의 대응 값과 곱합니다. 산출물이 어둡게 보이게 됩니다.
- *Burn*: 위에 있는 항목의 색상이 어두울수록 밑에 있는 항목도 어두워집니다. 밑에 있는 레이어에 색상을 부여하고 조정하는 데 이 모드를 쓸 수 있습니다.
- *Overlay*: 이 모드는 Multiply 와 Screen 혼합 모드를 조합합니다. 밝은 부분은 더 밝고 어두운 부분은 더 어두운 산출물을 생성합니다.
- *Soft light*: Overlay 모드와 매우 비슷하지만, Multiply/Screen 을 이용하는 대신 Burn/Dodge 모드의 색상을 이용합니다. 이미지 위에 부드러운 빛을 비추는 것 같은 효과를 낸다고 합니다.
- *Hard light*: 이 모드도 Overlay 모드와 매우 비슷합니다. 이미지 위에 아주 강렬한 빛을 비추는 것 같은 효과를 낸다고 합니다.
- *Difference*: 이 모드는 밑에 있는 픽셀값에서 위에 있는 픽셀값을 빼거나 그 반대로 계산해서 항상 양의 값을 남깁니다. 검정색은 혼합해도 아무 변화가 없습니다. 모든 색상과의 차가 0 이기 때문입니다.
- *Subtract*: 이 혼합 모드는 한 항목의 픽셀값에서 다른 항목의 픽셀값을 뺍니다. 음의 값이 나올 경우 검정색으로 표시합니다.

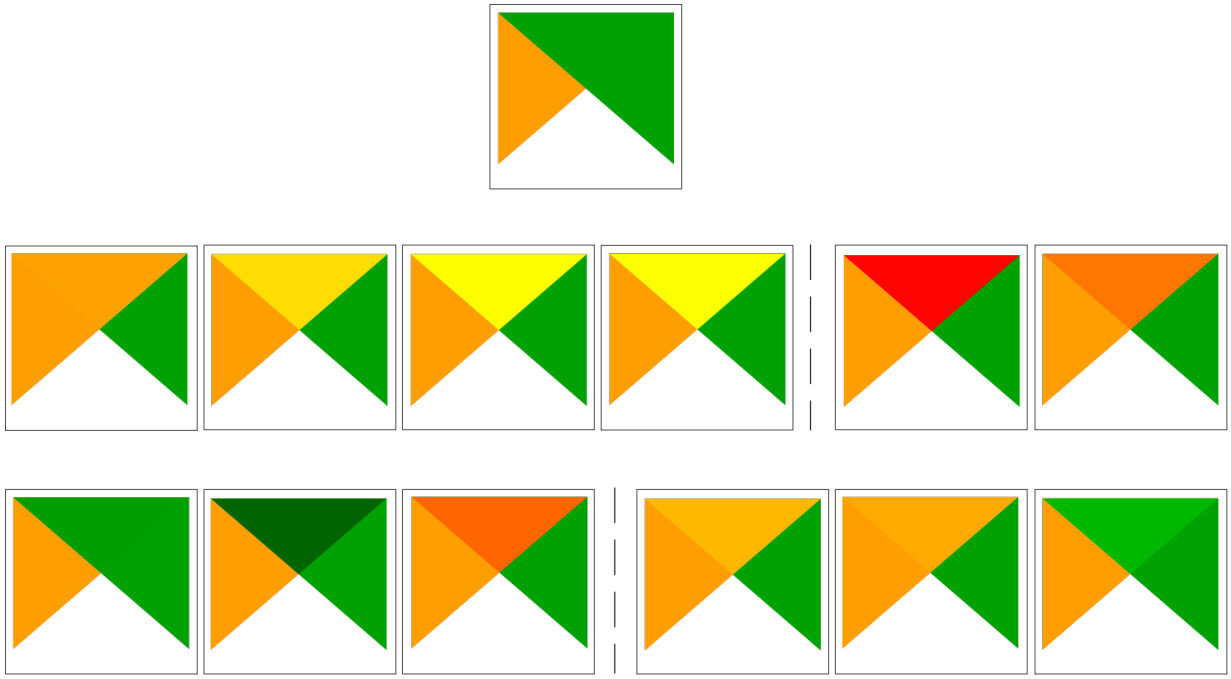


그림 8.27: 주황색 피처 위에 있는 녹색 피처에 적용된 혼합 모드의 예시들
위에서 아래, 왼쪽에서 오른쪽 방향으로: Lighten, Screen, Dodge, Addition –Difference, Subtract –Darken, Multiply, Burn – Overlay, Soft light, Hard light

레이어가레이어들을 그룹으로 렌더링하는 그룹의 일부분인 경우, 렌더링에 추가적인 혼합 모드를 사용할 수 있습니다. 이 추가 모드는 어떤 레이어의 렌더링을 두 번째"마스크"레이어의 렌더링으로 잘라낼 수 있는 방법을 제공합니다.

- *Masked By Below*: 상위 픽셀을 산출하지만, 상위 픽셀의 불투명도를 하위 픽셀의 불투명도만큼 감소시킵니다.
- *Mask Below*: 하위 픽셀을 산출하지만, 하위 픽셀의 불투명도를 상위 픽셀의 불투명도만큼 감소시킵니다.
- *Inverse Masked By Below*: 상위 픽셀을 산출하지만, 상위 픽셀의 불투명도를 하위 픽셀의 불투명도의 역 (투명도) 만큼 감소시킵니다.
- *Inverse Mask Below*: 하위 픽셀을 산출하지만, 하위 픽셀의 불투명도를 상위 픽셀의 불투명도의 역 (투명도) 만큼 감소시킵니다.
- *Paint Inside Below*: 상위 픽셀을 하위 픽셀 위에 혼합시키지만, 상위 픽셀의 불투명도를 하위 픽셀의 불투명도만큼 감소시킵니다.
- *Paint Below Inside*: 하위 픽셀을 상위 픽셀 위에 혼합시키지만, 하위 픽셀의 불투명도를 상위 픽셀의 불투명도만큼 감소시킵니다.

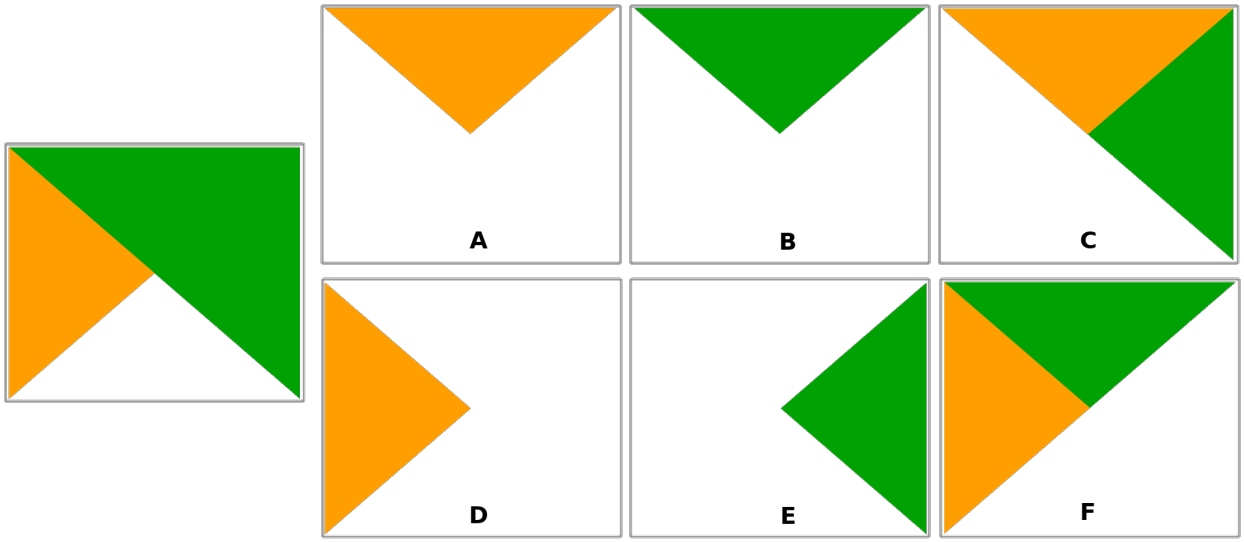



그림 8.28: 그룹 내에서 주황색 레이어 위에 있는 녹색 레이어에 적용된 혼합 잘라내기 모드의 예시들
**A**: Mask Below **B**: Masked By Below **C**: Paint Below Inside **D**: Inverse Mask Below **E**: Inverse Masked By Below **F**: Paint Inside Below

### 8.8.10 데이터 정의 무시 설정

벡터 레이어 속성 대화창 또는 인쇄 조판기의 설정에는 많은 옵션들이 있지만, 그 중에서도 Data defined override 아이콘이 있습니다. 레이어 속성 또는 항목 설정에 기반한표현식, 미리 작성한 또는 사용자 지정한 함수, 그리고변수를 사용하면 이 도구로 파라미터에 대한 동적인 값을 설정할 수 있습니다. 이 옵션을 활성화하면, 일반적인 값 (체크박스, 텍스트 상자, 슬라이드 바 등등) 에 상관없이 파라미터에 이 위젯이 반환하는 값을 적용합니다.

### 데이터 정의 무시 위젯

Data defined override 아이콘을 클릭하면 다음과 같은 항목들을 표시합니다:

- *Description*…은 해당 옵션 활성화 여부, 입력해야 할 값, 무결한 입력 유형 그리고 현재 정의를 설명합니다. 위젯에 마우스를 가져가도 이 정보들을 팝업으로 표시합니다.
- *Store data in the project*: 보조 저장소 속성 메커니즘을 사용해서 속성을 저장할 수 있는 버튼입니다.
- *Field type*: 무결한 입력 유형과 일치하는 레이어의 필드로부터 선택하는 항목입니다.
- *Color*: 위젯이 색상 속성과 링크되어 있는 경우, 이 메뉴를 사용하면 정의된 색상을 현재프로젝트 색상 스키마의 일부로서 접근할 수 있습니다.
- *Variable*: 사용할 수 있는 사용자 정의변수 에 접근할 수 있는 메뉴입니다.
- *Edit*…: *Expression String Builder* 대화창을 열어 표현식을 생성하거나 편집해서 적용할 수 있는 버튼입니다. 표현식을 정확히 작성할 수 있도록, 대화창에 기대되는 산출물의 서식을 알려주는 메시지를 표시합니다.
- *Paste* 및 *Copy* 버튼
- *Clear* 버튼은 설정을 초기화합니다.
- 수치 및 색상 속성의 경우, *Assistant*…메뉴를 사용해서 속성에 피처 데이터를 어떻게 적용시킬지 재조정할 수 있습니다. (자세한 내용은다음 절 을 참조하세요.)

**팁**

**오른쪽 클릭으로 데이터 정의 무시를 (비) 활성화하기**

데이터 정의 무시 옵션을 정확하게 설정했다면 노란색 또는 아이콘이 표시됩니다. 옵션이 정확하지 않다면 빨간색 또는 아이콘이 표시됩니다.

위젯을 오른쪽 클릭하는 것만으로도 환경설정된 Data-defined override 버튼을 활성화시키거나 비활성화시킬 수 있습니다.

**팁**

**Use middle-click to create or edit the expression to apply**

You can directly open the *Expression String Builder* dialog to create or edit the expression to apply by simply clicking the Data-defined override widget with the middle mouse button.

### 데이터 정의 어시스턴트 인터페이스 사용하기

When the Data-defined override button is associated with a size, a width, a rotation, an opacity or a color property, it has an *Assistant*… option that helps you change how the data is applied to the parameter for each feature. Basically, QGIS will scale a range of input values over a range of output values, and render each feature with their corresponding ouput value.

The assistant allows you to:

- Define the *Input* data, i.e.:
  - *Source*: 필드 또는 표현식 을 이용해서 표현할 수 있는 속성

  - the range of values to represent: you can manually enter the values or use the  Fetch value range from layer button to fill these fields automatically with the minimum and maximum values returned by the *Source* expression applied to your data. Features whose values are out of the given range will be represented with the property of the closest limit of the output range.

-  *Apply transform curve*: by default, output values are applied to input features following a specific scaling method (see below for setting). You can override this logic: enable the transform option, click on the graphic to add break point(s) and drag the point(s) to apply a custom distribution.
- *Output* 값을 정의할 수 있습니다: 이 옵션은 정의할 파라미터에 따라 달라집니다. 다음 전체 수준 값을 설정할 수 있습니다:
  - for color: a linear interpolation of source values is done over a given *color ramp* in order to assign each symbol the correct color. A color to use for symbols with NULL values can also be given.
  - for rotation: a linear interpolation is done over a range of minimum and maximum angles in order to assign each symbol the correct rotation. An angle to use for symbols with NULL values can also be given.
  - for opacity: a polynomial interpolation is done over a range of minimum and maximum opacity values. The *Exponent* field dictates the way source values are mapped to the output range. Power values greater than `1` will cause the output values distribution to start slowly before accelerating as the source values approach their maximum. Smaller exponents (less than `1`) will do the opposite. An opacity value to use for symbols with NULL values can also be given.
  - for width (line symbols): an interpolation is done over a range of minimum and maximum values in order to assign each symbol the correct width. The *Scale method* field controls how the interpolation is done and what the output value represents.
    * **Linear**: linearly interpolates the width of the symbol between the *Size from* and the *Size to* when the *Source* expression value varies from *value from* to *value to*. Symbols width are proportional.
    * **Exponential**: Also interpolates the width, but allows to control the exact power used for scaling through the *Exponent* field. Power values greater than `1` will cause the output values distribution to start slowly before accelerating as the source values approach their maximum. Smaller exponents (less than `1`) will do the opposite.

    The width to use for symbols with NULL values can also be given.
  - for size (point symbols): an interpolation is done over a range of minimum and maximum values in order to assign each symbol the correct size "diameter". The *Scale method* field controls how the interpolation is done and what the output value represents.
    * **Radius**: linearly interpolates the diameter of the symbol between the *Size from* and the *Size to* when the *Source* expression value varies from *value from* to *value to*. Symbols diameter are proportional.
    * **Surface**: scales the symbol diameter according to the square root (power of `0.5`) of the *Source* expression value to linearly interpolate the area of the symbol between the *Size from* and the *Size to* when the *Source* expression value varies from *value from* to *value to*. Symbols area are proportional. This method reflects human perception of size variation better than the **radius** one.
    * **Flannery**: Applies a compensation factor (power of `0.57`) to the area interpolation, in order to mitigate the eyes' perceptual underestimation of proportional areas. Maybe relevant only for circular symbols.
    * **Exponential**: Also interpolates the area, but allows to control the exact power used for scaling through the *Exponent* field. Power values greater than `1` will cause the output values distribution to start slowly before accelerating as the source values approach their maximum. Smaller exponents (less than `1`) will do the opposite.

    The size to use for symbols with NULL values can also be given.

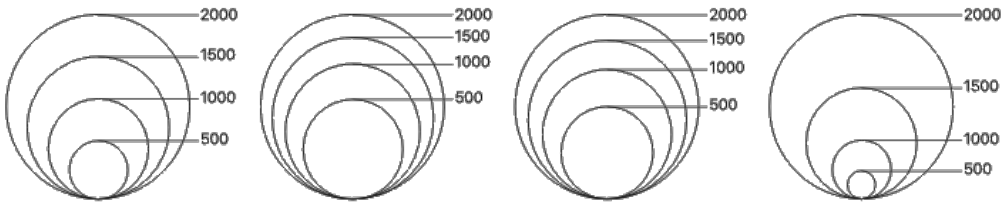



그림 8.29: Scaling values using different methods (from left to right): Radius - Surface - Flannery - Exponential of 2

적용이 가능한 속성인 경우, 대화창 오른쪽에 실시간으로 업데이트되는 미리보기를 표시해서 사용자가 값 조정 작업을 제어할 수 있도록 해줍니다.

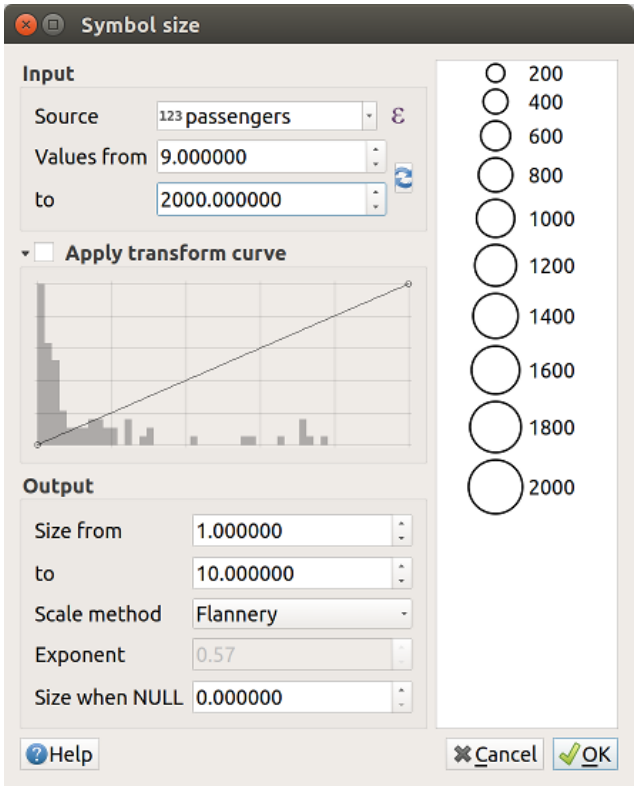



그림 8.30: 'passengers'필드의 값을 기반으로 피처 크기 조정하기

The values presented in the varying size assistant above will set the size 'Data-defined override'with the editable formula below:

```
coalesce(scale_polynomial("passengers", 9, 2000, 1, 10, 0.57), 0)
```

### 8.8.11 인증

Many data providers in QGIS (such as databases, web services or file-based services) require credentials in order to connect. The authentication widget in QGIS helps users supply these credentials securely, and it offers two ways to do so:

- entering basic credentials directly,
- or selecting an existing authentication configuration stored in the encrypted authentication database.

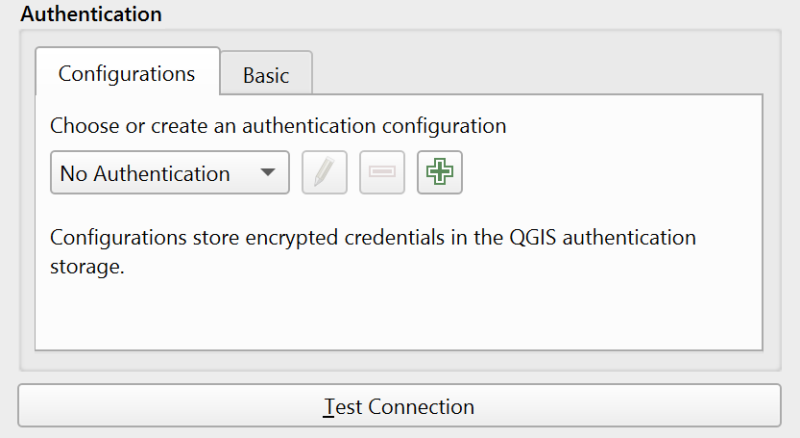



그림 8.31: Authentication widget

#### Basic

When setting authentication credentials, the *Basic* tab allows you to set:

- *User name*: User name used to connect to the data source.
- *Password*: Password used to connect to the data source.

*Convert to configuration* will save the user name and password in a new authentication configuration.

You can save any or both of the `User name` and `Password` parameters, in which case they will be used by default each time you need to connect to this database or service. If not saved, you'll be prompted to supply the credentials to connect to the database or service in next QGIS sessions. You can cancel the prompt, in which case the connection will fail. Click the *Ignore for session* button to avoid being prompted again during the current QGIS session. It is also possible to *Ignore for 10 seconds*. This will cancel the prompt but you need to wait 10 seconds before you can attempt to connect again. The connection parameters you entered are stored in a temporary internal cache and returned whenever a username/password for the same database or service is requested, until you end the current QGIS session.

> **경고**
>
> **QGIS User Settings and Security**
>
> In the *Authentication* tab, saving **username** and **password** will keep unprotected credentials in the connection configuration. Those **credentials will be visible** if, for instance, you share the project file with someone. Therefore, it is advisable to save your credentials in an *Authentication configuration* instead (*Configurations* tab - See 인증 시스템 for more details) or in a service connection file (see *PostgreSQL Service connection file* for example).

### Configurations

QGIS can store credentials under *authentication configurations*. Each configuration safely saves your login details in a portable encrypted database, so you can use them again without typing them every time. These configurations can be safely referenced by their ID tokens in projects and settings.

The authentication configuration selector allows users to choose from existing authentication configurations or create new ones. It provides options to:

- Select an existing authentication configuration from a drop-down list.
-  Create a new authentication configuration
-  Delete selected configuration
-  Edit selected configuration

For more information on the authentication system, see 인증 시스템.

CHAPTER 9

# 표현식으로 레벨업

## 9.1 표현식

레이어 데이터 및 사전 생성된 또는 사용자가 정의한 함수를 기반으로, **표현식** 은 도형 스타일, 라벨의 내용 또는 위치, 도표의 값, 조판기 항목의 높이 등을 동적으로 변경하고, 피처를 선택하고, 가상 필드 등을 생성하기 위해 속성값, 도형 및 변수를 처리할 수 있는 강력한 방법을 제공합니다.

> **참고**
>
> 함수 목록 에서 표현식 작성을 위한 기본 함수와 변수 목록을 자세한 정보 및 예시와 함께 찾아볼 수 있습니다.

### 9.1.1 표현식 문자열 작성기

*Expression string builder* 는 표현식을 생성하기 위한 주 대화창으로, QGIS 의 여러 부분에서 이용할 수 있는데 특히 다음과 같은 경우에 접근할 수 있습니다:

- 버튼을 클릭하는 경우
- Select By Expression… 도구로피처를 선택 하는 경우
- Field calculator 도구 등으로속성을 편집 하는 경우
- Data defined override 도구로 심볼, 라벨 또는 조판기 항목 파라미터를 처리하는 경우 (데이터 정의 무시 설정 참조)
- 도형 생성기 심볼 레이어를 생성하는 경우
- 지리 정보를 처리 하는 경우

표현식 작성기 대화창에서 다음 탭들을 선택할 수 있습니다:

- *Expression* 탭: 사전 정의 함수 목록 을 이용해서 사용할 표현식을 작성하고 확인할 수 있습니다.
- 함수 편집기 탭: 사용자 지정 함수를 생성해서 함수 목록을 확장할 수 있습니다.

### 인터페이스

*Expression* 탭은 함수, 레이어 필드 및 값을 이용해서 표현식을 작성할 수 있는 주 인터페이스를 제공합니다. 이 탭에 다음 작업을 할 수 있는 위젯들이 있습니다:

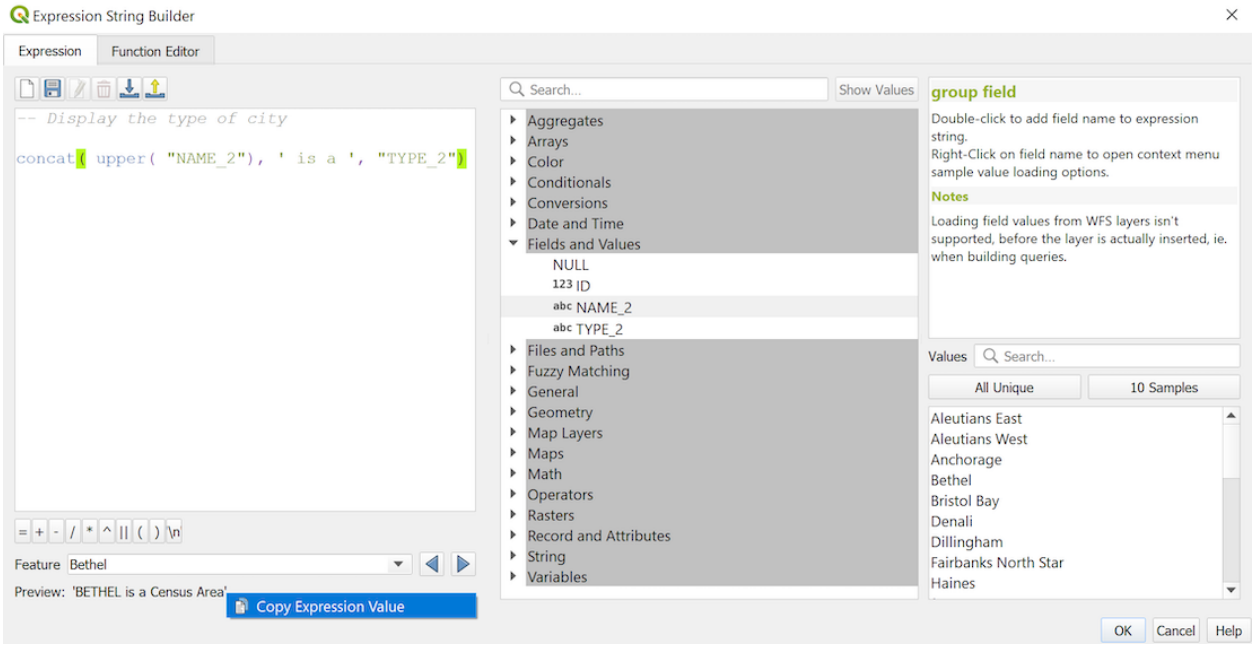



그림 9.1: 표현식 탭

- 표현식을 입력하거나 붙여넣을 수 있는 표현식 편집기 영역이 있습니다. 표현식 작성 속도를 향상시키기 위해 자동완성 기능을 사용할 수 있습니다:
  - 입력 텍스트에 대응하는 변수, 함수명 및 필드명을 아래쪽에 표시합니다. `Up` 및 `Down` 화살표를 사용해서 항목을 탐색한 다음 `Tab` 을 누르거나 원하는 항목을 클릭하기만 해도 표현식에 삽입할 수 있습니다.
  - 파라미터 입력 시 함수 파라미터가 표시됩니다.

  또한 QGIS 는 표현식이 올바른지 점검하고 다음을 사용해서 모든 오류를 강조합니다:
  - 밑줄 *(Underline)*: 알 수 없는 함수, 틀렸거나 무결하지 않은 인자 (argument) 를 표시합니다.
  - 마커 *(Marker)*: 단일 위치에 있는 다른 모든 오류를 (예를 들어 빠진 괄호, 예상 밖의 문자 등을) 표시합니다.

> **팁**
>
> **사용자 표현식을 주석으로 문서화**
>
> 복잡 표현식을 사용할 때, 사용자가 기억하기 쉽도록 텍스트를 여러 줄 또는 인라인 (inline) 주석 가운데 하나로 추가하는 편이 좋습니다.

```
/*
Labels each region with its highest (in altitude) airport(s)
and altitude, eg 'AMBLER : 264m' for the 'Northwest Artic' region
*/
with_variable(
  'airport_alti', -- stores the highest altitude of the region
  aggregate(
    'airports',
    'max',
    "ELEV", -- the field containing the altitude
    -- and limit the airports to the region they are within
    filter := within( @geometry, geometry( @parent ) )
  ),
    aggregate( -- finds airports at the same altitude in the region
      'airports',
      'concatenate',
      "NAME",
      filter := within( @geometry, geometry( @parent ) )
        and "ELEV" = @airport_alti
    )
    || ' : ' || @airport_alti || 'm'
    -- using || allows regions without airports to be skipped
)
```

- 표현식 편집기 위에 있는 도구 모음을 사용할 수 있습니다:
  -  Clear the expression editor: 표현식 편집기 초기화
  - 사용자 표현식 생성 및 관리
- 표현식 편집기 아래에는:
  - 표현식 작성에 도움이 되는 기본 연산자 모음이 있습니다.
  - 피처 속성을 데이터로 정의하는 경우 산출물의 예상 포맷을 알려줍니다.
  - *Output preview*: 기본적으로, 표현식이 레이어의 첫 번째 피처를 평가한 산출물의 미리보기를 (최대 60 글자까지) 실시간으로 표시합니다. 60 글자를 초과하는 산출물 미리보기 텍스트를 보려면 마우스 커서를 텍스트 위로 가져가십시오. 산출물 미리보기를 전부 담은 툴팁이 뜰 것입니다. 클립보드에 산출물 미리보기 텍스트를 복사하려면, 산출물 미리보기 텍스트를 오른쪽 클릭한 다음  *Copy Expression Value* 를 선택하면 됩니다.

    *Feature* 콤보박스를 이용하면 레이어의 다른 피처를 탐색해서 미리보기를 구할 수 있습니다. (콤보박스에 표시되는 값은 레이어의표시명 속성에서 가져옵니다.)

    오류가 발생하는 경우, 이를 알려주는 하이퍼링크를 통해 오류의 자세한 내용을 볼 수 있습니다.
- 함수 선택키는 그룹으로 정리된 함수, 변수, 필드 등등의 목록을 표시합니다. 목록을 필터링하는 검색란을 통해 특정 함수 또는 필드를 빨리 찾을 수 있습니다. 항목을 더블클릭하면 표현식 편집기에 해당 항목을 추가합니다.
- 도움말 패널이 함수 선택기에서 선택한 함수 별로 도움말을 표시합니다.

> **팁**
>
> 표현식에 있는 함수명을 `Ctrl+Click` 하면 자동적으로 대화창에 도움말을 표시합니다.

함수 선택기에서 필드를 선택하면 나타나는 필드의 값 위젯을 통해 다음과 같은 피처 속성을 가져올 수 있습니다:

- 특정 필드 값 탐색
- *All Unique* 또는 *10 Samples* 값 목록을 표시. 오른쪽 클릭으로도 표시할 수 있습니다.

  필드가 또다른 레이어나 값 모음과 매핑된 경우, 예를 들어필드 위젯 이 *RelationReference*, *ValueRelation* 또는 *ValueMap* 유형인 경우 (참조된 레이어, 테이블 또는 목록으로부터) 매핑된 필드의 모든 값을 목록화할 수 있습니다. 여기에 더해, *Only show values in use* 옵션을 체크하면 이 목록을 현재 필드에서 사용중인 값만 표시하도록 필터링할 수 있습니다.

위젯에 있는 필드 값을 더블 클릭하면 표현식 편집기에 해당 값을 추가합니다.

**팁**

대화창에서 함수 도움말이나 필드값을 표시하는 오른쪽 패널을 접을 수 (보이지 않게 할 수) 있습니다. 이 패널을 다시 펼치려면 *Show Values* 또는 *Show Help* 버튼을 클릭하세요.

### 표현식 작성하기

QGIS 표현식은 피처를 선택하거나 값을 설정하는 데 쓰입니다. QGIS 에서 표현식을 작성하려면 몇몇 규칙을 따라야 합니다:

1. **대화창이 맥락을 정의한다**: 사용자가 SQL 에 익숙하다면, *select features from layer where condition* 또는 *update layer set field = new_value where condition* 유형의 쿼리를 알고 있을 것입니다. QGIS 표현식도 이런 모든 정보를 필요로 하지만 사용자가 표현식 작성기 대화창을 열기 위해 사용하는 도구가 이 정보의 일부를 제공합니다. 예를 들어, `height` 필드를 가진 `buildings` 레이어를 지정한다고 할 때:

   - Select by expression 도구를 클릭한다는 것은"건물 레이어에서 피처를 선택"하려 한다는 뜻입니다. 사용자가 표현식 텍스트 위젯에 입력해야 하는 정보는 **조건** 뿐입니다. 예를 들면 높이가 20 을 초과하는 건물을 선택하려면 `"height" > 20` 이라고 입력하면 됩니다.
   - 이렇게 선택 집합을 만든 다음, Field calculator 버튼을 누르고 *Update existing field* 에"height"를 선택하면 이미"update buildings set height = ??? where height > 20"명령어를 지정한 것입니다. 이 경우 사용자가 지정해야 하는 나머지 부분은 **new value** 뿐입니다. 예를 들어 표현식 편집기 텍스트란에 `50` 을 입력해서 이전에 선택했던 건물의 높이를 설정하십시오.

2. **따옴표에 주의하라**: 작은따옴표는 문자 그대로 반환하기 때문에, 작은따옴표 사이에 있는 텍스트는 문자열로 (`'145'`) 해석됩니다. 큰따옴표는 해당 텍스트의 값을 반환하므로 필드 용으로 (`"myfield"`) 사용하십시오. 필드는 따옴표 없이도 (`myfield`) 사용할 수 있습니다. 숫자는 따옴표를 사용하지 않습니다 (`3.16`).

**참고**

필드명으로 보통 문자열을 인자로 받는 함수에서는 다음과 같이 입력하고:

```
attribute( @atlas_feature, 'height' ) -- returns the value stored in the
↪"height" attribute of the current atlas feature
```

다음처럼 입력하면 안 됩니다:

```
attribute( @atlas_feature, "height" ) -- fetches the value of the attribute␣
↪named "height" (e.g. 100), and use that value as a field
                                    -- from which to return the atlas␣
↪feature value. Probably wrong as a field named "100" may not exist.
```

> **팁**
>
> **표현식을 읽기 쉽게 하려면 명명된 파라미터를 이용**
>
> 어떤 함수들은 설정하는 데 여러 파라미터가 필요합니다. 표현식 엔진은 명명 파라미터를 지원하고 있습니다. 즉 암호 같은 표현식 `clamp(1, 2, 9)` 을 작성하는 대신, `clamp(min:=1, value:=2, max:=9)` 같은 표현식을 사용할 수 있다는 뜻입니다. 이렇게 하면 인자들도 `clamp(value:=2, max:=9, min:=1)` 처럼 위치를 바꿀 수 있습니다. 명명 파라미터를 이용하면 표현식 함수의 인자들이 어떤 의미인지 명확해집니다. 시간이 지난 뒤에 표현식을 해석하려 할 때 유용하겠죠!

### 몇몇 표현식 용례

- 필드 계산기에서 기존"total_pop"과"area_km2"필드를 이용, "pop_density"필드를 계산합니다:

```
"total_pop" / "area_km2"
```

- 피처의 면적에 따라 피처에 라벨을 추가하거나 범주화합니다:

```
CASE WHEN $area > 10 000 THEN 'Larger' ELSE 'Smaller' END
```

- "pop_density"값에 따른 범주로"density_level"필드를 업데이트합니다:

```
CASE WHEN "pop_density" < 50 THEN 'Low population density'
     WHEN "pop_density" >= 50 and "pop_density" < 150 THEN 'Medium population␣
↪density'
     WHEN "pop_density" >= 150 THEN 'High population density'
END
```

- 평균 집값이 제곱미터 당 1 만 유로 이하인지 초과인지를 기준으로 모든 피처에 범주 스타일을 적용합니다:

```
"price_m2" > 10000
```

- *Select By Expression*…도구를 통해"High population density"지역인 동시에 평균 집값이 제곱미터 당 1 만 유로를 초과하는 지역을 나타내는 모든 피처를 선택합니다:

```
"density_level" = 'High population density' and "price_m2" > 10000
```

  이런 표현식을 이용해서 어떤 피처에 라벨을 적용할지 또는 맵에 표시할지 정의할 수도 있습니다.

- "lands"레이어의 자연 구역과 중첩하는 피처를 선택합니다:

```
overlay_intersects( layer:='lands', filter:="zone_type"='Natural' )
```

- 각 피처가 담고 있는 건물의 개수를 셉니다:

```
array_length( overlay_contains( layer:='buildings', expression:=@id ) )
```

- 도형 생성기를 통해 레이어 별로 서로 다른 (유형의) 심볼을 생성합니다:

```
point_on_surface( @geometry )
```

- 포인트 피처를 지정, 포인트 도형 주위에 닫힌 라인을 (`make_line` 을 이용해서) 생성합니다:

```
make_line(
  -- using an array of points placed around the original
  array_foreach(
    -- list of angles for placing the projected points (every 90°)
    array:=generate_series( 0, 360, 90 ),
    -- translate the point 20 units in the given direction (angle)
    expression:=project( @geometry, distance:=20, azimuth:=radians( @element ) )
  )
)
```

- 인쇄 조판기 라벨에, 조판"Map 1"항목 안에 있는"airports"피처들의 이름을 표시합니다:

```
with_variable( 'extent',
               map_get( item_variables( 'Map 1' ), 'map_extent' ),
               aggregate( 'airports', 'concatenate', "NAME",
                          intersects( @geometry, @extent ), ' ,'
                        )
             )
```

#### 표현식 저장하기

표현식 편집기 프레임 위에 있는 Add current expression to user expressions 버튼을 사용해서 사용자가 자주 사용하는 중요한 표현식을 저장할 수 있습니다. 가운데 패널에 있는 **사용자 표현식** 그룹에서 사용할 수 있습니다. 이 사용자 표현식은사용자 프로파일 (<userprofile>/QGIS/QGIS3.ini 파일) 에 저장되며 현재 사용자 프로파일의 모든 프로젝트 안에서의 모든 표현식 대화창에서 사용할 수 있습니다.

표현식 편집기 프레임 위에 있는 도구 모음을 통해 사용자 표현식을 관리할 수 있습니다:

- Add the current expression to user expressions: 사용자 프로파일에 표현식을 저장합니다. 더 쉽게 식별하기 위한 라벨과 도움말 텍스트도 추가할 수 있습니다.
- Edit selected expression from user expressions: 사용자 표현식에서 선택한 표현식은 물론 그 라벨과 도움말을 편집합니다.
- Remove selected expression from user expressions: 사용자 표현식에서 선택한 표현식을 제거합니다.
- Import user expressions: .json 파일에서 현재 사용자 프로파일 폴더로 사용자 표현식을 가져옵니다.
- Export user expressions: 사용자 프로파일 QGIS3.ini 파일에 있는 모든 사용자 표현식을 .json 파일로 공유합니다.

### 9.1.2 함수 편집기

*Function Editor* 탭에서 사용자 고유의 함수를 파이썬 언어로 작성할 수 있습니다. 이 탭은 미리 정의된 함수로는 커버할 수 없는 특별한 요구 사항을 쉽고 편하게 처리할 수 있는 방법을 제공합니다.

새 함수를 생성하려면:

1. New File 버튼을 누르십시오.
2. Select where to store the new function. You can choose to save it either in a *Function file* or in *Project functions*.

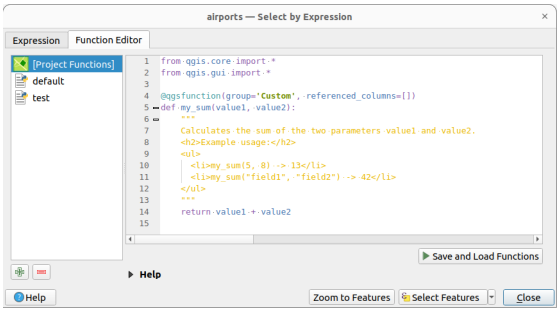



그림 9.2: 함수 편집기 탭

- If you select *Function file*, you need to enter a name for the Python `.py` file, which is based on a QGIS template file and is stored in the `/python/expressions` folder under the active *user profile* directory, thus available for all projects you may open with that profile.

  Custom Python functions stored as Function files are auto loaded whenever you start a new project under the user profile. Be aware that for sharing such Python functions, you need to share the `.py` file in the `/python/expressions` folder.

- If you select *Project functions*, the new function is stored in the project file and only available for that project.

  Custom Python functions stored as Project functions can be easily shared by sharing the project file where they were saved. When opening a project with project functions, QGIS can load or ignore them depending on whether *embedded Python code is enabled*.

A new item is added in the left panel of the *Function Editor* tab.

3. The right panel displays the content of the file: a python script template.
4. Uncomment the code block and start editing it according to your needs. You can rely on code autocompletion, code formatting and typing helpers, as defined in *Python settings*. Online API documentation for Qt and PyQGIS objects is also accessible pressing the `F1` key.

   You will find in the *Help* widget general information for setting the @qgsfunction decorator.

5. *Save and Load Functions* 버튼을 누르십시오. *Expression* 탭의 함수 트리에 사용자가 작성한 함수를 추가합니다. 기본적으로 `Custom` 그룹 아래 추가될 것입니다.
6. 사용자의 새 함수를 즐겨보세요.
7. 함수를 향상시켜야 할 경우, *Function Editor* 탭을 선택해서 수정한 다음 *Save and Load Functions* 을 다시 누르면 파일에서 수정 사항을 사용할 수 있게 됩니다. 즉 어떤 표현식 탭에서도 사용할 수 있게 됩니다.

사용자 지정 함수를 삭제하려면:

1. *Function Editor* 탭을 활성화하고
2. 목록에서 함수를 선택한 다음
3. Press the Remove selected function. The function is removed from the list and, depending on the storage, the corresponding `.py` file is deleted from the user profile folder, or the Python functions are removed from the project file.

참고

When a project file with Project functions is closed, the corresponding Project functions are unloaded and are no longer available in the QGIS session.

### 예시

다음은 값 2 개를 받아 실행되는 사용자만의 `my_sum` 함수를 생성하는 방법에 대한 간단한 예시입니다.

```
from qgis.core import *
from qgis.gui import *

@qgsfunction(group='Custom', referenced_columns=[])
def my_sum(value1, value2):
    """
    Calculates the sum of the two parameters value1 and value2.
    <h2>Example usage:</h2>
    <ul>
      <li>my_sum(5, 8) -> 13</li>
      <li>my_sum("field1", "field2") -> 42</li>
    </ul>
    """
    return value1 + value2
```

`@qgsfunction` 장식자 (decorator) 는 다음 인자들을 받습니다:

- `group`: 이 인자는 표현식 대화창에서 이 함수가 목록화되어야 할 그룹을 나타냅니다.
- `usesgeometry`: Set this to `True` if the expression requires access to the features geometry. By default `False`.
- `handlesnull`: Set this to `True` if the expression has custom handling for NULL values. If `False` (default), the result will always be NULL as soon as any parameter is NULL.
- `referenced_columns`: An array of attribute names that are required to run the function. Defaults to `[QgsFeatureRequest.ALL_ATTRIBUTES]`. Specifying a subset of fields or an empty list will result in a faster execution.
- `params_as_list`: Set this to `True` to pass the function parameters as a list. If `False` (default), the function will receive the parameters as individual arguments.

The function itself takes any number and type of parameters you want to pass to your function. These should be set before the following optional arguments:

- `feature`: 현재 피처
- `parent`: `QgsExpression` 객체
- `context`: 이 인자가 마지막 위치에 존재하는 경우 `QgsExpressionContext` 객체를 담고 있을 것입니다. 이 객체는 표현식 변수 같은 다양한 추가 정보에 접근할 수 있게 해줍니다. 예를 들어 `context.variable('layer_id')` 와 같은 방식으로 말이죠.

If those arguments are present in the function signature, they will be automatically passed to the function, without the need to specify them in the expression.

이제 이렇게 정의한 예시 함수를 표현식 탭에서 사용할 수 있습니다:

Find more examples and additional arguments for custom functions at `qgsfunction`. Further information about creating Python code can be found in the PyQGIS-Developer-Cookbook.

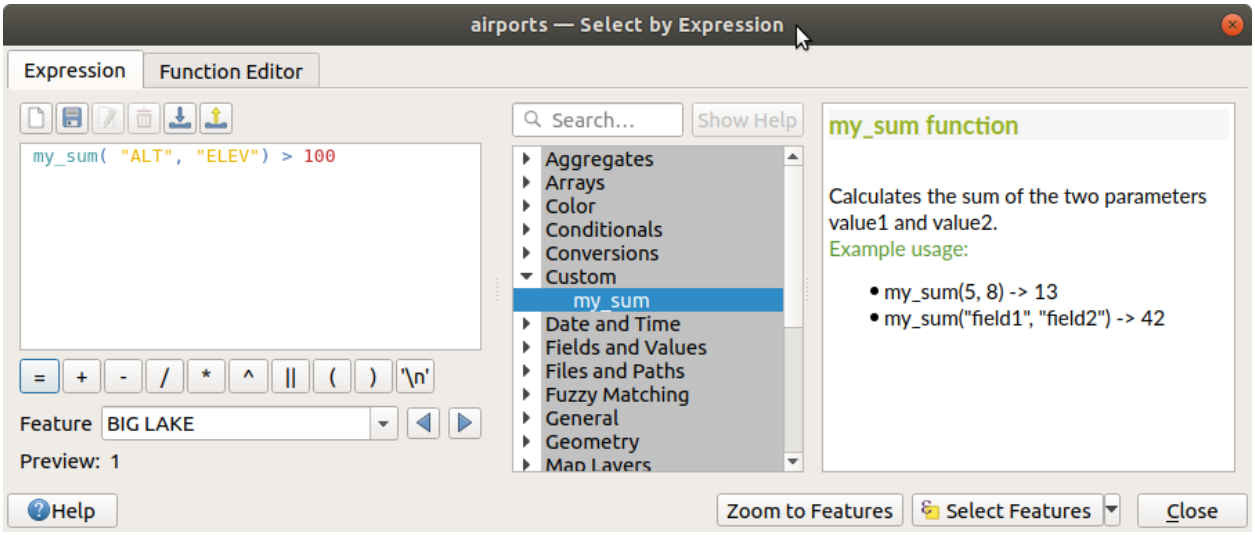



그림 9.3: 표현식 탭에 추가된 사용자 지정 함수

## 9.2 함수 목록

QGIS 에서 사용할 수 있는 함수, 연산자 및 변수의 목록을 다음에 범주별로 정리했습니다.

### 9.2.1 집계 함수

이 그룹은 레이어 및 필드에 있는 값들을 집계하는 함수를 담고 있습니다.

#### aggregate

다른 레이어의 피처를 이용해서 계산한 집계값을 반환합니다.

| | |
|---|---|
| 문법 | aggregate(layer, aggregate, expression, [filter], [concatenator=''], [order_by])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **layer** - 레이어 이름 또는 레이어 ID 가운데 하나를 나타내는 문자열<br>• **aggregate** - 계산할 집계에 대응하는 문자열. 무결한 옵션은 다음과 같습니다:<br>– count<br>– count_distinct<br>– count_missing<br>– minimum or min<br>– maximum or max<br>– sum<br>– mean<br>– median<br>– stdev<br>– stdevsample<br>– range<br>– minority<br>– majority<br>– q1: 제 1 사분위수 (First Quartile)<br>– q3: 제 3 사분위수 (Third Quartile)<br>– iqr: 사분위간 범위 (Inter Quartile Range)<br>– min_length: 최단 문자열 길이<br>– max_length: 최장 문자열 길이<br>– concatenate: 연결자 (concatenator) 로 문자열을 결합<br>– concatenate_unique: 연결자로 유일 문자열을 결합<br>– collect: 집계된 다중 부분 도형을 생성<br>– array_agg: 집계된 값의 배열을 생성<br>• **expression** - 집계할 하위 표현식 또는 필드명<br>• **filter** - 집계 용으로 사용되는 객체를 제한하기 위한 부가적인 필터 표현식입니다. 결합한 레이어에 있는 객체에서 필드 및 도형을 가져옵니다. 변수 @parent 를 통해 소스 객체에 접근할 수 있습니다.<br>• **concatenator** - '연결 (concatenate)'및'유일값 연결 (concatenate_unique)'집계를 위한 값을 결합하는 데 사용할 부가적인 문자열<br>• **order_by** - 집계 용으로 사용되는 객체를 정렬하기 위한 부가적인 필터 표현식입니다. 결합한 레이어에 있는 객체에서 필드 및 도형을 가져옵니다. 기본적으로, 지정되지 않은 순서로 피처를 반환할 것입니다. |
| 예제 | • `aggregate(layer:='rail_stations',aggregate:='sum', expression:="passengers")` → rail_stations 레이어의 passengers 필드에서 나온 모든 값의 합계<br>• `aggregate('rail_stations','sum', "passengers"/7)` → 값들의 합을 구하기 전에"passengers"필드를 7 로 나누어"passengers"의 일일 평균을 계산<br>• `aggregate(layer:='rail_stations',aggregate:='sum', expression:="passengers",filter:="class">3)` → "class"속성이 3 보다 큰 객체에서만"passengers"필드의 모든 값에 대한 합계<br>• `aggregate(layer:='rail_stations',aggregate:='concatenate', expression:="name", concatenator:=',')` → rail_stations 레이어에서 모든 객체에 대한 쉼표로 구분된 name 필드의 값 목록<br>• `aggregate(layer:='countries', aggregate:='max', expression:="code", filter:=intersects( @geometry, geometry(@parent) ) )` → 'countries'레이어 상에서 교차하는 국가의 국가 코드<br>• `aggregate(layer:='rail_stations',aggregate:='sum', expression:="passengers",filter:=contains( @atlas_geometry, @geometry ) )` → 현재 지도책 피처 안에 있는'rail_stations'레이어의'passengers' 필드에서 나온 모든 값의 합계<br>• `aggregate(layer:='rail_stations', aggregate:='collect', expression:=centroid(@geometry), filter:="region_name" = attribute(@parent,'name') )` → 현재 피처와 동일한 지역의'rail_stations'레이어의 중심점 도형을 집계 |

### array_agg

필드 또는 표현식에서 집계한 값들의 배열을 반환합니다.

| | |
|---|---|
| 문법 | array_agg(expression, [group_by], [filter], [order_by])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **expression** - 집계할 필드의 하위 표현식<br>• **group_by** - 그룹 집계 계산에 사용할 부가적인 표현식<br>• **filter** - 집계하는 데 사용되는 객체를 필터링하는 데 사용할 부가적인 표현식<br>• **order_by** - 집계하는 데 사용되는 객체를 정렬하는 데 사용하기 위한 부가적인 표현식. 기본적으로, 지정되지 않은 순서로 피처를 반환할 것입니다. |
| 예제 | • `array_agg("name",group_by:="state")` → 'state'필드로 그룹화된 이름 값 목록 |

### collect

표현식에서 나온 집계 도형들로 이루어진 다중 부분 도형을 반환합니다.

| | |
|---|---|
| 문법 | collect(expression, [group_by], [filter])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **expression** - 집계할 도형 표현식<br>• **group_by** - 그룹 집계 계산에 사용할 부가적인 표현식<br>• **filter** - 집계하는 데 사용되는 객체를 필터링하는 데 사용할 부가적인 표현식 |
| 예제 | • `collect( @geometry )` → 집계된 도형들로 이루어진 다중 부분 도형<br>• `collect( centroid(@geometry), group_by:="region", filter:="use" = 'civilian' )` → 피처의'region'값을 바탕으로 집계된'civilian'피처의 중심점 (centroid) |

### concatenate

필드 또는 표현식에서 나오는 집계된 문자열을 모두 구분자로 결합해서 반환합니다.

| | |
|---|---|
| 문법 | concatenate(expression, [group_by], [filter], [concatenator], [order_by])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **expression** - 집계할 필드의 하위 표현식<br>• **group_by** - 그룹 집계 계산에 사용할 부가적인 표현식<br>• **filter** - 집계하는 데 사용되는 객체를 필터링하는 데 사용할 부가적인 표현식<br>• **concatenator** - 값을 결합하는 데 사용할 부가적인 문자열. 기본적으로 비어 있습니다.<br>• **order_by** - 집계하는 데 사용되는 객체를 정렬하는 데 사용하기 위한 부가적인 표현식. 기본적으로, 지정되지 않은 순서로 피처를 반환할 것입니다. |
| 예제 | • `concatenate("town_name",group_by:="state",concatenator:=',')` → 'state'필드로 그룹화된'town_names'를 쉼표로 구분한 목록 |

### concatenate_unique

필드 또는 표현식에서 나오는 유일한 문자열을 모두 구분자로 결합해서 반환합니다.

| | |
|---|---|
| 문법 | concatenate_unique(expression, [group_by], [filter], [concatenator], [order_by])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **expression** - 집계할 필드의 하위 표현식<br>• **group_by** - 그룹 집계 계산에 사용할 부가적인 표현식<br>• **filter** - 집계하는 데 사용되는 객체를 필터링하는 데 사용할 부가적인 표현식<br>• **concatenator** - 값을 결합하는 데 사용할 부가적인 문자열. 기본적으로 비어 있습니다.<br>• **order_by** - 집계하는 데 사용되는 객체를 정렬하는 데 사용하기 위한 부가적인 표현식. 기본적으로, 지정되지 않은 순서로 피처를 반환할 것입니다. |
| 예제 | • `concatenate_unique("town_name",group_by:="state",concatenator:=',')` → 'state'필드로 그룹화된 유일한'town_names'를 쉼표로 구분한 목록 |

### count

일치하는 객체 수를 반환합니다.

| | |
|---|---|
| 문법 | count(expression, [group_by], [filter])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **expression** - 집계할 필드의 하위 표현식<br>• **group_by** - 그룹 집계 계산에 사용할 부가적인 표현식<br>• **filter** - 집계하는 데 사용되는 객체를 필터링하는 데 사용할 부가적인 표현식 |
| 예제 | • `count("stations",group_by:="state")` → 'state'필드로 그룹화된'stations' 개수 |

### count_distinct

고유값의 수를 반환합니다.

| | |
|---|---|
| 문법 | count_distinct(expression, [group_by], [filter])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **expression** - 집계할 필드의 하위 표현식<br>• **group_by** - 그룹 집계 계산에 사용할 부가적인 표현식<br>• **filter** - 집계하는 데 사용되는 객체를 필터링하는 데 사용할 부가적인 표현식 |
| 예제 | • `count_distinct("stations",group_by:="state")` → 'state'필드로 그룹화된 고유한'stations'개수 |

### count_missing

누락된 (NULL) 값의 개수를 반환합니다.

| | |
|---|---|
| 문법 | count_missing(expression, [group_by], [filter])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **expression** - 집계할 필드의 하위 표현식<br>• **group_by** - 그룹 집계 계산에 사용할 부가적인 표현식<br>• **filter** - 집계하는 데 사용되는 객체를 필터링하는 데 사용할 부가적인 표현식 |
| 예제 | • `count_missing("stations",group_by:="state")` → 'state'필드로 그룹화된 누락 (NULL) 'stations'개수 |

### iqr

필드 또는 표현식으로부터 계산한 중간 사분위수 범위 (inter quartile range) 를 반환합니다.

| | |
|---|---|
| 문법 | iqr(expression, [group_by], [filter])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **expression** - 집계할 필드의 하위 표현식<br>• **group_by** - 그룹 집계 계산에 사용할 부가적인 표현식<br>• **filter** - 집계하는 데 사용되는 객체를 필터링하는 데 사용할 부가적인 표현식 |
| 예제 | • `iqr("population",group_by:="state")` → 'state'필드로 그룹화된'population' 값의 중간 사분위수 범위 |

### majority

필드 또는 표현식에서 집계 다수 값 (가장 흔하게 나타나는 값) 을 반환합니다.

| | |
|---|---|
| 문법 | majority(expression, [group_by], [filter])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **expression** - 집계할 필드의 하위 표현식<br>• **group_by** - 그룹 집계 계산에 사용할 부가적인 표현식<br>• **filter** - 집계하는 데 사용되는 객체를 필터링하는 데 사용할 부가적인 표현식 |
| 예제 | • `majority("class",group_by:="state")` → 'state'필드로 그룹화된 범주에서 가장 흔하게 나타나는 값 |

### max_length

필드 또는 표현식에서 가장 긴 문자열의 길이를 반환합니다.

| | |
|---|---|
| 문법 | max_length(expression, [group_by], [filter])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **expression** - 집계할 필드의 하위 표현식<br>• **group_by** - 그룹 집계 계산에 사용할 부가적인 표현식<br>• **filter** - 집계하는 데 사용되는 객체를 필터링하는 데 사용할 부가적인 표현식 |
| 예제 | • `max_length("town_name",group_by:="state")` → 'state'필드로 그룹화된 'town_name'의 최장 길이 |

### maximum

필드 또는 표현식에서 나오는 집계 최대값을 반환합니다.

| | |
|---|---|
| 문법 | maximum(expression, [group_by], [filter])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **expression** - 집계할 필드의 하위 표현식<br>• **group_by** - 그룹 집계 계산에 사용할 부가적인 표현식<br>• **filter** - 집계하는 데 사용되는 객체를 필터링하는 데 사용할 부가적인 표현식 |
| 예제 | • `maximum("population",group_by:="state")` → 'state'필드로 그룹화된'population'의 최대값 |

### mean

필드 또는 표현식에서 나오는 집계 평균값을 반환합니다.

| | |
|---|---|
| 문법 | mean(expression, [group_by], [filter])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **expression** - 집계할 필드의 하위 표현식<br>• **group_by** - 그룹 집계 계산에 사용할 부가적인 표현식<br>• **filter** - 집계하는 데 사용되는 객체를 필터링하는 데 사용할 부가적인 표현식 |
| 예제 | • `mean("population",group_by:="state")` → 'state'필드로 그룹화된'population'의 평균값 |

### median

필드 또는 표현식에서 나오는 집계 중앙값을 반환합니다.

| | |
|---|---|
| 문법 | median(expression, [group_by], [filter])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **expression** - 집계할 필드의 하위 표현식<br>• **group_by** - 그룹 집계 계산에 사용할 부가적인 표현식<br>• **filter** - 집계하는 데 사용되는 객체를 필터링하는 데 사용할 부가적인 표현식 |
| 예제 | • `median("population",group_by:="state")` → 'state'필드로 그룹화된'population'의 중간값 |

### min_length

필드 또는 표현식에서 가장 짧은 문자열의 길이를 반환합니다.

| | |
|---|---|
| 문법 | min_length(expression, [group_by], [filter])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **expression** - 집계할 필드의 하위 표현식<br>• **group_by** - 그룹 집계 계산에 사용할 부가적인 표현식<br>• **filter** - 집계하는 데 사용되는 객체를 필터링하는 데 사용할 부가적인 표현식 |
| 예제 | • `min_length("town_name",group_by:="state")` → 'state'필드로 그룹화된 'town_name'의 최단 길이 |

### minimum

필드 또는 표현식에서 나오는 집계 최소값을 반환합니다.

| | |
|---|---|
| 문법 | minimum(expression, [group_by], [filter])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **expression** - 집계할 필드의 하위 표현식<br>• **group_by** - 그룹 집계 계산에 사용할 부가적인 표현식<br>• **filter** - 집계하는 데 사용되는 객체를 필터링하는 데 사용할 부가적인 표현식 |
| 예제 | • `minimum("population",group_by:="state")` → 'state'필드로 그룹화된'population'의 최소값 |

### minority

필드 또는 표현식에서 집계 소수 값 (가장 뜸하게 나타나는 값) 을 반환합니다.

| | |
|---|---|
| 문법 | minority(expression, [group_by], [filter])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **expression** - 집계할 필드의 하위 표현식<br>• **group_by** - 그룹 집계 계산에 사용할 부가적인 표현식<br>• **filter** - 집계하는 데 사용되는 객체를 필터링하는 데 사용할 부가적인 표현식 |
| 예제 | • `minority("class",group_by:="state")` → 'state'필드로 그룹화된 범주에서 가장 뜸하게 나타나는 값 |

### q1

필드 또는 표현식으로부터 계산한 첫 번째 사분위수 (quartile) 를 반환합니다.

| | |
|---|---|
| 문법 | q1(expression, [group_by], [filter])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **expression** - 집계할 필드의 하위 표현식<br>• **group_by** - 그룹 집계 계산에 사용할 부가적인 표현식<br>• **filter** - 집계하는 데 사용되는 객체를 필터링하는 데 사용할 부가적인 표현식 |
| 예제 | • `q1("population",group_by:="state")` → 'state'필드로 그룹화된'population'의 첫 번째 사분위수 값 |

### q3

필드 또는 표현식으로부터 계산한 세 번째 사분위수를 반환합니다.

| | |
|---|---|
| 문법 | q3(expression, [group_by], [filter])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **expression** - 집계할 필드의 하위 표현식<br>• **group_by** - 그룹 집계 계산에 사용할 부가적인 표현식<br>• **filter** - 집계하는 데 사용되는 객체를 필터링하는 데 사용할 부가적인 표현식 |
| 예제 | • `q3("population",group_by:="state")` → 'state'필드로 그룹화된'population'의 세 번째 사분위수 값 |

### range

필드 또는 표현식에서 나오는 값들의 (최대 ~ 최소) 집계 범위를 반환합니다.

| | |
|---|---|
| 문법 | range(expression, [group_by], [filter])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **expression** - 집계할 필드의 하위 표현식<br>• **group_by** - 그룹 집계 계산에 사용할 부가적인 표현식<br>• **filter** - 집계하는 데 사용되는 객체를 필터링하는 데 사용할 부가적인 표현식 |
| 예제 | • `range("population",group_by:="state")` → 'state'필드로 그룹화된'population'값들의 범위 |

### relation_aggregate

레이어 관계에서 나오는 파생 객체들 가운데 일치하는 모든 객체를 사용해서 계산한 집계값을 반환합니다.

| | |
|---|---|
| 문법 | relation_aggregate(relation, aggregate, expression, [concatenator=”], [order_by])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **relation** - 관계 ID 를 표현하는 문자열<br>• **aggregate** - 계산할 집계에 대응하는 문자열. 무결한 옵션은 다음과 같습니다:<br>– count<br>– count_distinct<br>– count_missing<br>– minimum or min<br>– maximum or max<br>– sum<br>– mean<br>– median<br>– stdev<br>– stdevsample<br>– range<br>– minority<br>– majority<br>– q1: 제 1 사분위수 (First Quartile)<br>– q3: 제 3 사분위수 (Third Quartile)<br>– iqr: 사분위간 범위 (Inter Quartile Range)<br>– min_length: 최단 문자열 길이<br>– max_length: 최장 문자열 길이<br>– concatenate: 연결자 (concatenator) 로 문자열을 결합<br>– concatenate_unique: 연결자로 유일 문자열을 결합<br>– collect: 집계된 다중 부분 도형을 생성<br>– array_agg: 집계된 값의 배열을 생성<br>• **expression** - 집계할 하위 표현식 또는 필드명<br>• **concatenator** - ‘연결 (concatenate)’집계를 위한 값을 결합하는 데 사용할 부가적인 문자열<br>• **order_by** - 집계 용으로 사용되는 객체를 정렬하기 위한 부가적인 표현식입니다. 결합한 레이어에 있는 객체에서 필드 및 도형을 가져옵니다. 기본적으로, 지정되지 않은 순서로 피처를 반환할 것입니다. |
| 예제 | • `relation_aggregate(relation:='my_relation', aggregate:='mean',expression:="passengers")` → ‘my_relation’관계를 사용해서 매칭한 모든 파생 객체의 평균값<br>• `relation_aggregate('my_relation','sum', "passengers"/7)` → ‘my_relation’관계를 사용해서 매칭한 모든 파생 객체에 대해‘passengers’필드의 합을 7 로 나눈 값<br>• `relation_aggregate('my_relation','concatenate', "towns", concatenator:=',')` → ‘my_relation’관계를 사용해서 매칭한 모든 파생 객체에 대한 ‘towns’필드의 쉼표 구분 목록<br>• `relation_aggregate('my_relation','array_agg', "id")` → ‘my_relation’관계를 사용해서 매칭한 모든 파생 객체에서 나온 ID 필드의 목록 |

더 읽어볼 거리: 여러 레이어 간 관계 설정하기

#### stdev

필드 또는 표현식에서 집계 표준 편차값을 반환합니다.

| | |
|---|---|
| 문법 | stdev(expression, [group_by], [filter])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **expression** - 집계할 필드의 하위 표현식<br>• **group_by** - 그룹 집계 계산에 사용할 부가적인 표현식<br>• **filter** - 집계하는 데 사용되는 객체를 필터링하는 데 사용할 부가적인 표현식 |
| 예제 | • `stdev("population",group_by:="state")` → 'state'필드로 그룹화된'population'값의 표준 편차 |

#### sum

필드 또는 표현식에서 집계 적산값을 반환합니다.

| | |
|---|---|
| 문법 | sum(expression, [group_by], [filter])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **expression** - 집계할 필드의 하위 표현식<br>• **group_by** - 그룹 집계 계산에 사용할 부가적인 표현식<br>• **filter** - 집계하는 데 사용되는 객체를 필터링하는 데 사용할 부가적인 표현식 |
| 예제 | • `sum("population",group_by:="state")` → 'state'필드로 그룹화된'population'값의 합계 |

### 9.2.2 배열 함수

이 함수 그룹은 (목록 데이터 구조라고도 하는) 배열을 생성하고 처리하는 함수들을 담고 있습니다. 배열 내부의 값의 순서가 중요합니다. 키-값 쌍이 중요하지 않고 값을 값의 키로 식별하는'맵'데이터 구조 와는 다릅니다.

#### array

파라미터로써 전달된 모든 값들을 담고 있는 배열을 반환합니다.

| | |
|---|---|
| 문법 | array(value1, value2, ⋯) |
| 인자 | • **value** - 값 |
| 예제 | • `array(2,10)` → [ 2, 10 ]<br>• `array(2,10)[0]` → 2 |

### array_all

배열이 지정한 배열의 모든 값을 담고 있는 경우 참을 반환합니다.

| | |
|---|---|
| 문법 | array_all(array_a, array_b) |
| 인자 | • **array_a** - 배열<br>• **array_b** - 검색할 값들의 배열 |
| 예제 | • `array_all(array(1,2,3),array(2,3))` → TRUE<br>• `array_all(array(1,2,3),array(1,2,4))` → FALSE |

### array_append

지정한 값을 맨 끝에 추가한 배열을 반환합니다.

| | |
|---|---|
| 문법 | array_append(array, value) |
| 인자 | • **array** - 배열<br>• **value** - 추가할 값 |
| 예제 | • `array_append(array(1,2,3),4)` → [ 1, 2, 3, 4 ] |

### array_cat

지정한 배열들을 모두 연결해서 담고 있는 배열을 반환합니다.

| | |
|---|---|
| 문법 | array_cat(array1, array2, ⋯) |
| 인자 | • **array** - 배열 |
| 예제 | • `array_cat(array(1,2),array(2,3))` → [ 1, 2, 2, 3 ] |

### array_contains

배열이 지정한 값을 담고 있는 경우 참을 반환합니다.

| | |
|---|---|
| 문법 | array_contains(array, value) |
| 인자 | • **array** - 배열<br>• **value** - 검색할 값 |
| 예제 | • `array_contains(array(1,2,3),2)` → TRUE |

### array_count

배열에서 지정한 값이 나타나는 횟수를 셉니다.

| | |
|---|---|
| 문법 | array_count(array, value) |
| 인자 | • **array** - 배열<br>• **value** - 개수를 셀 값 |
| 예제 | • `array_count(array('a', 'b', 'c', 'b'), 'b')` → 2 |

### array_distinct

지정한 배열의 개별 (distinct) 값들을 담고 있는 배열을 반환합니다.

| | |
|---|---|
| 문법 | array_distinct(array) |
| 인자 | • **array** - 배열 |
| 예제 | • `array_distinct(array(1,2,3,2,1))` → [ 1, 2, 3 ] |

### array_filter

표현식이 참이라고 평가한 항목만 보유한 배열을 반환합니다.

| | |
|---|---|
| 문법 | array_filter(array, expression, [limit=0])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **array** - 배열<br>• **expression** - 각 항목을 평가할 표현식입니다. 변수 *@element* 는 현재 값으로 대체될 것입니다.<br>• **limit** - 반환할 요소들의 최대 개수. 모든 값을 반환하게 하려면 0 을 사용하십시오. |
| 예제 | • `array_filter(array(1,2,3),@element < 3)` → [ 1, 2 ]<br>• `array_filter(array(1,2,3),@element < 3, 1)` → [ 1 ] |

### array_find

배열 안에 있는 값의 가장 낮은 인덱스를 반환합니다. (첫 번째 인덱스는 0 입니다.) 값을 찾을 수 없는 경우 -1 을 반환합니다.

| | |
|---|---|
| 문법 | array_find(array, value) |
| 인자 | • **array** - 배열<br>• **value** - 검색할 값 |
| 예제 | • `array_find(array('a', 'b', 'c'), 'b')` → 1<br>• `array_find(array('a', 'b', 'c', 'b'), 'b')` → 1 |

### array_first

배열의 첫 번째 값을 반환합니다.

| | |
|---|---|
| 문법 | array_first(array) |
| 인자 | • **array** - 배열 |
| 예제 | • `array_first(array('a','b','c'))` → 'a' |

### array_foreach

지정한 표현식으로 각 항목을 평가한 배열을 반환합니다.

| | |
|---|---|
| 문법 | array_foreach(array, expression) |
| 인자 | • **array** - 배열<br>• **expression** - an expression to evaluate on each item. The variable *@element* will be replaced by the current value and the variable *@counter* by the current index (starting with 0). |
| 예제 | • `array_foreach(array('a','b','c'),upper(@element))` → [ 'A', 'B', 'C' ]<br>• `array_foreach(array(1,2,3),@element + 10)` → [ 11, 12, 13 ]<br>• `array_foreach(array(1,2,3),@element + @counter)` → [ 1, 3, 5 ] |

### array_get

배열의 (0 부터 시작하는) N 번째 값 또는 (마지막이 -1 인) -N 번째 값을 반환합니다.

| | |
|---|---|
| 문법 | array_get(array, pos) |
| 인자 | • **array** - 배열<br>• **pos** - 가져올 인덱스 (0 기반) |
| 예제 | • `array_get(array('a','b','c'),1)` → 'b'<br>• `array_get(array('a','b','c'),-1)` → 'c' |

**힌트**

배열에서 값을 가져오기 위해인덱스 연산자 *([])* 를 사용할 수도 있습니다.

### array_insert

지정한 위치에 지정한 값을 삽입한 배열을 반환합니다.

| | |
|---|---|
| 문법 | array_insert(array, pos, value) |
| 인자 | • **array** - 배열<br>• **pos** - 추가할 위치 (0 기반)<br>• **value** - 추가할 값 |
| 예제 | • `array_insert(array(1,2,3),1,100)` → [ 1, 100, 2, 3 ] |

### array_intersect

array1 의 요소 중 적어도 하나 이상이 array2 에 있으면 참을 반환합니다.

| | |
|---|---|
| 문법 | array_intersect(array1, array2) |
| 인자 | • **array1** - 배열<br>• **array2** - 또다른 배열 |
| 예제 | • `array_intersect(array(1,2,3,4),array(4,0,2,5))` → TRUE |

### array_last

배열의 마지막 값을 반환합니다.

| | |
|---|---|
| 문법 | array_last(array) |
| 인자 | • **array** - 배열 |
| 예제 | • `array_last(array('a','b','c'))` → 'c' |

### array_length

배열의 요소 개수를 반환합니다.

| | |
|---|---|
| 문법 | array_length(array) |
| 인자 | • **array** - 배열 |
| 예제 | • `array_length(array(1,2,3))` → 3 |

### array_majority

배열에서 가장 많이 나타나는 값을 반환합니다.

| | |
|---|---|
| 문법 | array_majority(array, [option='all'])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **array** - 배열<br>• **option='all'** - 반환값 처리를 지정하는 문자열. 무결한 옵션은 다음과 같습니다:<br>– all: 기본값으로, 배열에서 가장 많이 나타나는 값을 모두 반환합니다.<br>– any: 가장 많이 나타나는 값 가운데 하나를 반환합니다.<br>– median: 가장 많이 나타나는 값들의 중앙값을 반환합니다. 숫자가 아닌 값은 무시합니다.<br>– real_majority: 배열 크기의 절반을 초과해서 나타나는 값을 반환합니다. |
| 예제 | • `array_majority(array(0,1,42,42,43), 'all')` → [ 42 ]<br>• `array_majority(array(0,1,42,42,43,1), 'all')` → [ 42, 1 ]<br>• `array_majority(array(0,1,42,42,43,1), 'any')` → 1 or 42<br>• `array_majority(array(0,1,1,2,2), 'median')` → 1.5<br>• `array_majority(array(0,1,42,42,43), 'real_majority')` → NULL<br>• `array_majority(array(0,1,42,42,43,42), 'real_majority')` → NULL<br>• `array_majority(array(0,1,42,42,43,42,42), 'real_majority')` → 42 |

### array_max

배열의 최대값을 반환합니다.

| | |
|---|---|
| 문법 | array_max(array) |
| 인자 | • **array** - 배열 |
| 예제 | • `array_max(array(0,42,4,2))` → 42 |

### array_mean

배열에 있는 숫자값들의 평균을 반환합니다. 배열에서 숫자가 아닌 값은 무시합니다.

| | |
|---|---|
| 문법 | array_mean(array) |
| 인자 | • **array** - 배열 |
| 예제 | • `array_mean(array(0,1,7,66.6,135.4))` → 42<br>• `array_mean(array(0,84,'a','b','c'))` → 42 |

### array_median

배열에 있는 숫자값들의 중앙값을 반환합니다. 배열에서 숫자가 아닌 값은 무시합니다.

| | |
|---|---|
| 문법 | array_median(array) |
| 인자 | • **array** - 배열 |
| 예제 | • `array_median(array(0,1,42,42,43))` → 42<br>• `array_median(array(0,1,2,42,'a','b'))` → 1.5 |

### array_min

배열의 최소값을 반환합니다.

| | |
|---|---|
| 문법 | array_min(array) |
| 인자 | • **array** - 배열 |
| 예제 | • `array_min(array(43,42,54))` → 42 |

### array_minority

배열에서 가장 적게 나타나는 값을 반환합니다.

| | |
|---|---|
| 문법 | array_minority(array, [option='all'])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **array** - 배열<br>• **option='all'** - 반환값 처리를 지정하는 문자열. 무결한 옵션은 다음과 같습니다:<br>– all: 기본값으로, 배열에서 가장 적게 나타나는 값을 모두 반환합니다.<br>– any: 가장 적게 나타나는 값 가운데 하나를 반환합니다.<br>– median: 가장 적게 나타나는 값들의 중앙값을 반환합니다. 숫자가 아닌 값은 무시합니다.<br>– real_majority: 배열 크기의 절반 미만으로 나타나는 값을 반환합니다. |
| 예제 | • `array_minority(array(0,42,42), 'all')` → [ 0 ]<br>• `array_minority(array(0,1,42,42), 'all')` → [ 0, 1 ]<br>• `array_minority(array(0,1,42,42,43,1), 'any')` → 0 or 43<br>• `array_minority(array(1,2,3,3), 'median')` → 1.5<br>• `array_minority(array(0,1,42,42,43), 'real_minority')` → [ 42, 43, 0, 1 ]<br>• `array_minority(array(0,1,42,42,43,42), 'real_minority')` → [ 42, 43, 0, 1 ]<br>• `array_minority(array(0,1,42,42,43,42,42), 'real_minority')` → [ 43, 0, 1 ] |

### array_prepend

지정한 값을 맨 앞에 추가한 배열을 반환합니다.

| | |
|---|---|
| 문법 | array_prepend(array, value) |
| 인자 | • **array** - 배열<br>• **value** - 추가할 값 |
| 예제 | • `array_prepend(array(1,2,3),0)` → [ 0, 1, 2, 3 ] |

### array_prioritize

어떤 배열을 다른 배열에 지정된 순서로 재정렬해서 반환합니다. 첫 번째 배열에는 있지만 두 번째 배열에는 없는 값은 산출물의 마지막에 추가될 것입니다.

| 문법 | array_prioritize(array, array_prioritize) |
| --- | --- |
| 인자 | • **array** - 배열<br>• **array_prioritize** - 우선 순위로 정렬된 값을 가진 배열 |
| 예제 | • `array_prioritize(array(1, 8, 2, 5), array(5, 4, 2, 1, 3, 8))` → [ 5, 2, 1, 8 ]<br>• `array_prioritize(array(5, 4, 2, 1, 3, 8), array(1, 8, 6, 5))` → [ 1, 8, 5, 4, 2, 3 ] |

### array_remove_all

모든 항목에서 지정한 값을 제거한 배열을 반환합니다.

| 문법 | array_remove_all(array, value) |
| --- | --- |
| 인자 | • **array** - 배열<br>• **value** - 제거할 값 |
| 예제 | • `array_remove_all(array('a','b','c','b'),'b')` → [ 'a', 'c' ] |

### array_remove_at

지정한 인덱스 위치의 항목을 제거한 배열을 반환합니다. (첫 번째 요소가 0 인) 양의 인덱스와 (마지막 요소가 -1 인) 음의 인덱스를 지원합니다.

| 문법 | array_remove_at(array, pos) |
| --- | --- |
| 인자 | • **array** - 배열<br>• **pos** - 제거할 위치 (0 기반) |
| 예제 | • `array_remove_at(array(1, 2, 3), 1)` → [1, 3 ]<br>• `array_remove_at(array(1, 2, 3), -1)` → [1, 2 ] |

### array_replace

배열을 지정한 값, 배열, 또는 값들의 맵으로 치환해서 반환합니다.

**값 및 배열 변이형**

지정한 값 또는 값들의 배열을 또 다른 값 또는 값들의 배열로 치환해서 반환합니다.

| | |
|---|---|
| 문법 | array_replace(array, before, after) |
| 인자 | • **array** - 입력 배열<br>• **before** - 치환할 값 또는 값들의 배열<br>• **after** - 치환에 사용할 값 또는 값들의 배열 |
| 예제 | • `array_replace(array('QGIS','SHOULD','ROCK'),'SHOULD','DOES')` → [ 'QGIS', 'DOES', 'ROCK' ]<br>• `array_replace(array(3,2,1),array(1,2,3),array(7,8,9))` → [ 9, 8, 7 ]<br>• `array_replace(array('Q','G','I','S'),array('Q','S'),'-')` → [ '-', 'G', 'I', '-' ] |

#### 맵 변이형

지정한 맵 키를 짝지어진 값으로 치환한 배열을 반환합니다.

| | |
|---|---|
| 문법 | array_replace(array, map) |
| 인자 | • **array** - 입력 배열<br>• **map** - 키와 값을 담고 있는 맵 |
| 예제 | • `array_replace(array('APP', 'SHOULD', 'ROCK'),map('APP', 'QGIS','SHOULD','DOES'))` → [ 'QGIS', 'DOES', 'ROCK' ] |

Further reading: *replace*, *regexp_replace*

### array_reverse

입력 배열의 값들의 순서를 역전시킨 배열을 반환합니다.

| | |
|---|---|
| 문법 | array_reverse(array) |
| 인자 | • **array** - 배열 |
| 예제 | • `array_reverse(array(2,4,0,10))` → [ 10, 0, 4, 2 ] |

### array_slice

배열의 일부를 반환합니다. start_pos 와 end_pos 인수로 반환할 일부를 정의합니다.

| | |
|---|---|
| 문법 | array_slice(array, start_pos, end_pos) |
| 인자 | • **array** - 배열<br>• **start_pos** - 반환할 일부의 시작 위치의 (0 기반) 인덱스입니다. start_pos 인덱스는 반환할 일부에 포함됩니다. start_pos 가 음수일 경우, 목록의 마지막부터 인덱스를 셉니다. (-1 기반)<br>• **end_pos** - 반환할 일부의 마지막 위치의 (0 기반) 인덱스입니다. end_pos 인덱스는 반환할 일부에 포함됩니다. end_pos 가 음수일 경우, 목록의 마지막부터 인덱스를 셉니다. (-1 기반) |
| 예제 | • `array_slice(array(1,2,3,4,5),0,3)` → [ 1, 2, 3, 4 ]<br>• `array_slice(array(1,2,3,4,5),0,-1)` → [ 1, 2, 3, 4, 5 ]<br>• `array_slice(array(1,2,3,4,5),-5,-1)` → [ 1, 2, 3, 4, 5 ]<br>• `array_slice(array(1,2,3,4,5),0,0)` → [ 1 ]<br>• `array_slice(array(1,2,3,4,5),-2,-1)` → [ 4, 5 ]<br>• `array_slice(array(1,2,3,4,5),-1,-1)` → [ 5 ]<br>• `array_slice(array('Dufour','Valmiera','Chugiak','Brighton'),1,2)` → [ 'Valmiera', 'Chugiak' ]<br>• `array_slice(array('Dufour','Valmiera','Chugiak','Brighton'),-2,-1)` → [ 'Chugiak', 'Brighton' ] |

### array_sort

지정한 배열의 요소들을 정렬시킨 배열을 반환합니다.

| | |
|---|---|
| 문법 | array_sort(array, [ascending=true])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **array** - 배열<br>• **ascending** - 배열을 내림차순으로 정렬하려면 이 파라미터를 false 로 설정하십시오. |
| 예제 | • `array_sort(array(3,2,1))` → [ 1, 2, 3 ] |

### array_sum

배열에 있는 숫자값들의 합을 반환합니다. 배열에서 숫자가 아닌 값은 무시합니다.

| | |
|---|---|
| 문법 | array_sum(array) |
| 인자 | • **array** - 배열 |
| 예제 | • `array_sum(array(0,1,39.4,1.6,'a'))` → 42.0 |

### array_to_string

배열 요소들을 구분자로 구분하고 빈 값은 선택적인 문자열로 대체한 문자열로 연결합니다.

| | |
|---|---|
| 문법 | array_to_string(array, [delimiter=','], [empty_value=''])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **array** - 입력 배열<br>• **delimiter** - 연결된 배열 요소들을 분리하는 데 사용하는 문자열 구분자<br>• **empty_value** - 빈 (길이가 0 인) 값을 대체하는 데 사용할 선택적인 문자열 |
| 예제 | • `array_to_string(array('1','2','3'))` → '1,2,3'<br>• `array_to_string(array(1,2,3),'-')` → '1-2-3'<br>• `array_to_string(array('1','','3'),',','0')` → '1,0,3' |

Further reading: *string_to_array*

### generate_series

이어지는 일련의 숫자를 담고 있는 배열을 생성합니다.

| | |
|---|---|
| 문법 | generate_series(start, stop, [step=1])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **start** - 시퀀스 (일련의 숫자) 의 첫 번째 값<br>• **stop** - 시퀀스가 도달하면 중지하는 값<br>• **step** - 값 사이의 증분으로 사용되는 값 |
| 예제 | • `generate_series(1,5)` → [ 1, 2, 3, 4, 5 ]<br>• `generate_series(5,1,-1)` → [ 5, 4, 3, 2, 1 ] |

### geometries_to_array

도형 하나를 배열에 더 단순한 도형들로 분할합니다.

| | |
|---|---|
| 문법 | geometries_to_array(geometry) |
| 인자 | • **geometry** - 입력 도형 |
| 예제 | • `geometries_to_array(geom_from_wkt('MultiPoint (1 2, 5 21)'))` → An array containing 'Point (1 2)'and 'Point (5 21)'<br>• `geometries_to_array(geom_from_wkt('GeometryCollection (Polygon ((5 8, 4 1, 3 2, 5 8)),LineString (3 2, 4 2))'))` → 폴리곤 도형 하나와 라인 도형 하나로 이루어진 배열<br>• `geom_to_wkt(geometries_to_array(geom_from_wkt('GeometryCollection (Polygon ((5 8, 4 1, 3 2, 5 8)),LineString (3 2, 4 2))'))[0])` → 'Polygon ((5 8, 4 1, 3 2, 5 8))'<br>• `geometries_to_array(geom_from_wkt('MULTIPOLYGON(((5 5,0 0,0 10,5 5)),((5 5,10 10,10 0,5 5))'))` → 폴리곤 도형 2 개로 이루어진 배열 |

### regexp_matches

그룹을 캡처해서 캡처된 모든 문자열의 배열을, 문자열을 대상으로 지정한 정규 표현식에 그룹 자체가 나타나는 순서대로 반환합니다.

| | |
|---|---|
| 문법 | regexp_matches(string, regex, [empty_value=''])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **string** - 정규 표현식에 대해 그룹을 캡처해올 문자열<br>• **regex** - 그룹을 캡처하는 데 사용된 정규 표현식<br>• **empty_value** - 빈 (길이가 0 인) 값을 대체하는 데 사용할 선택적인 문자열 |
| 예제 | • `regexp_matches('QGIS=>rocks','(.*)=>(.*)')` → [ 'QGIS', 'rocks']<br>• `regexp_matches('key=>','(.*)=>(.*)','empty value')` → [ 'key', 'empty value'] |

Further reading: *substr*, *regexp_substr*

### string_to_array

문자열을 지정한 구분자와 빈 값을 위한 선택적인 문자열을 사용해서 배열로 분해합니다.

| | |
|---|---|
| 문법 | string_to_array(string, [delimiter=','], [empty_value=''])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **string** - 입력 문자열<br>• **delimiter** - 입력 문자열을 분할하는 데 사용한 문자열 구분자<br>• **empty_value** - 빈 (길이가 0 인) 값을 대체하는 데 사용할 선택적인 문자열 |
| 예제 | • `string_to_array('1,2,3',',')` → [ '1', '2', '3']<br>• `string_to_array('1,,3',',','0')` → [ '1', '0', '3'] |

Further reading: *array_to_string*

### 9.2.3 색상 함수

이 그룹은 색상을 처리하기 위한 함수를 담고 있습니다.

#### color_cmyk

색상의 청록색, 자홍색, 황색, 흑색 요소를 기반으로 색상의 문자열 표현을 반환합니다.

| | |
|---|---|
| 문법 | color_cmyk(cyan, magenta, yellow, black) |
| 인자 | • **cyan** - 색상의 청록색 (cyan) 요소를 0 에서 100 사이의 백분율 정수값으로 입력<br>• **magenta** - 색상의 자홍색 (magenta) 요소를 0 에서 100 사이의 백분율 정수값으로 입력<br>• **yellow** - 색상의 황색 (yellow) 요소를 0 에서 100 사이의 백분율 정수값으로 입력<br>• **black** - 색상의 흑색 (black) 요소를 0 에서 100 사이의 백분율 정수값으로 입력 |
| 예제 | • `color_cmyk(100,50,0,10)` → '0,115,230' |

#### color_cmyka

색상의 청록색, 자홍색, 황색, 흑색 및 알파 (투명도) 요소를 기반으로 색상의 문자열 표현을 반환합니다.

| | |
|---|---|
| 문법 | color_cmyka(cyan, magenta, yellow, black, alpha) |
| 인자 | • **cyan** - 색상의 청록색 (cyan) 요소를 0 에서 100 사이의 백분율 정수값으로 입력<br>• **magenta** - 색상의 자홍색 (magenta) 요소를 0 에서 100 사이의 백분율 정수값으로 입력<br>• **yellow** - 색상의 황색 (yellow) 요소를 0 에서 100 사이의 백분율 정수값으로 입력<br>• **black** - 색상의 흑색 (black) 요소를 0 에서 100 사이의 백분율 정수값으로 입력<br>• **alpha** - 투명도를 0(완전히 투명) 에서 255(완전히 불투명) 사이의 정수값으로 입력 |
| 예제 | • `color_cmyka(100,50,0,10,200)` → '0,115,230,200' |

### color_cmykf

Returns a color object based on its cyan, magenta, yellow, black and alpha components.

| | |
|---|---|
| 문법 | color_cmykf(cyan, magenta, yellow, black, [alpha=1.0])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **cyan** - cyan component as a float value from 0.0 to 1.0<br>• **magenta** - magenta component as a float value from 0.0 to 1.0<br>• **yellow** - yellow component as a float value from 0.0 to 1.0<br>• **black** - black component as a float value from 0.0 to 1.0<br>• **alpha** - alpha component as a float value from 0.0 to 1.0 |
| 예제 | • `color_cmykf(1,0.9,0.81,0.62)` → CMYKA: 1.00,0.90,0.81,0.62,1.00 |

### color_grayscale_average

Applies a grayscale filter to a color and returns it. Returned type is the same as color argument, i.e. a color string representation or a color object.

| | |
|---|---|
| 문법 | color_grayscale_average(color) |
| 인자 | • **color** - a color string representation or a color object |
| 예제 | • `color_grayscale_average('255,100,50')` → '135,135,135,255'<br>• `color_grayscale_average(color_cmykf(0.6,0.5,0.1,0.8))` → CMYKA: 0.40,0.40,0.40,0.80,1.00 |

### color_hsl

색상의 색조, 채도, 명도 속성을 기반으로 색상의 문자열 표현을 반환합니다.

| | |
|---|---|
| 문법 | color_hsl(hue, saturation, lightness) |
| 인자 | • **hue** - 색상의 색조를 0 에서 360 사이의 정수값으로 입력<br>• **saturation** - 색상의 채도 백분율을 0 에서 100 사이의 정수값으로 입력<br>• **lightness** - 색상의 명도 백분율을 0 에서 100 사이의 정수값으로 입력 |
| 예제 | • `color_hsl(100,50,70)` → '166,217,140' |

### color_hsla

색상의 색조, 채도, 명도 그리고 알파 (투명도) 속성을 기반으로 색상의 문자열 표현을 반환합니다.

| | |
|---|---|
| 문법 | color_hsla(hue, saturation, lightness, alpha) |
| 인자 | • **hue** - 색상의 색조를 0 에서 360 사이의 정수값으로 입력<br>• **saturation** - 색상의 채도 백분율을 0 에서 100 사이의 정수값으로 입력<br>• **lightness** - 색상의 명도 백분율을 0 에서 100 사이의 정수값으로 입력<br>• **alpha** - 투명도를 0(완전히 투명) 에서 255(완전히 불투명) 사이의 정수값으로 입력 |
| 예제 | • `color_hsla(100,50,70,200)` → '166,217,140,200' |

### color_hslf

Returns a color object based on its hue, saturation, and lightness attributes.

| | |
|---|---|
| 문법 | color_hslf(hue, saturation, lightness, [alpha=1.0])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **hue** - hue of the color, as a float value from 0.0 to 1.0<br>• **saturation** - saturation of the color as a float value from 0.0 to 1.0<br>• **lightness** - lightness of the color as a float value from 0.0 to 1.0<br>• **alpha** - alpha component as a float value from 0.0 to 1.0 |
| 예제 | • `color_hslf(0.3,0.52,0.7)` → HSLA: 0.30,0.52,0.70,1.00 |

### color_hsv

색상의 색상, 채도, 명도 속성을 기반으로 색상의 문자열 표현을 반환합니다.

| | |
|---|---|
| 문법 | color_hsv(hue, saturation, value) |
| 인자 | • **hue** - 색상의 색조를 0 에서 360 사이의 정수값으로 입력<br>• **saturation** - 색상의 채도 백분율을 0 에서 100 사이의 정수값으로 입력<br>• **value** - 색상의 명도 백분율을 0 에서 100 사이의 정수값으로 입력 |
| 예제 | • `color_hsv(40,100,100)` → '255,170,0' |

### color_hsva

색상의 색조, 채도, 명도 그리고 알파 (투명도) 속성을 기반으로 색상의 문자열 표현을 반환합니다.

| | |
|---|---|
| 문법 | color_hsva(hue, saturation, value, alpha) |
| 인자 | • **hue** - 색상의 색조를 0 에서 360 사이의 정수값으로 입력<br>• **saturation** - 색상의 채도 백분율을 0 에서 100 사이의 정수값으로 입력<br>• **value** - 색상의 명도 백분율을 0 에서 100 사이의 정수값으로 입력<br>• **alpha** - 투명도를 0(완전히 투명) 에서 255(완전히 불투명) 사이의 정수값으로 입력 |
| 예제 | • `color_hsva(40,100,100,200)` → '255,170,0,200' |

### color_hsvf

Returns a color object based on its hue, saturation, and value attributes.

| | |
|---|---|
| 문법 | color_hsvf(hue, saturation, value, [alpha=1.0])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **hue** - hue of the color, as a float value from 0.0 to 1.0<br>• **saturation** - saturation of the color as a float value from 0.0 to 1.0<br>• **value** - value of the color as as a float value from 0.0 to 1.0<br>• **alpha** - alpha component as a float value from 0.0 to 1.0 |
| 예제 | • `color_hsvf(0.4,1,0.6)` → HSVA: 0.40,1.00,0.60,1.00 |

### color_mix

Returns a color mixing the red, green, blue, and alpha values of two provided colors based on a given ratio. Returned type is the same as color arguments, i.e. a color string representation or a color object.

| | |
|---|---|
| 문법 | color_mix(color1, color2, ratio) |
| 인자 | • **color1** - a color string or a color object<br>• **color2** - a color string or a color object<br>• **ratio** - 비율 |
| 예제 | • `color_mix_rgb('0,0,0','255,255,255',0.5)` → '127,127,127,255'<br>• `color_mix(color_cmykf(0.9,0.9,0.9,0.9),color_cmykf(0.1,0.1,0.1,0.1),0.5)` → CMYKA: 0.50,0.50,0.50,0.50,1.00 |

### color_mix_rgb

지정한 두 색상의 적색, 녹색, 청색 그리고 알파 (투명도) 값을 지정한 비율로 혼합한 색상을 표현하는 문자열을 반환합니다.

| | |
|---|---|
| 문법 | color_mix_rgb(color1, color2, ratio) |
| 인자 | • **color1** - 색상 문자열<br>• **color2** - 색상 문자열<br>• **ratio** - 비율 |
| 예제 | • `color_mix_rgb('0,0,0','255,255,255',0.5)` → '127,127,127,255' |

### color_part

Returns a specific component from a color string or color object, e.g., the red component or alpha component.

| | |
|---|---|
| 문법 | color_part(color, component) |
| 인자 | • **color** - a color string or a color object<br>• **component** - 반환할 색상 요소에 대응하는 문자열입니다. 사용할 수 있는 옵션은 다음과 같습니다:<br>– red: RGB 적색 요소 (0-255)<br>– green: RGB 녹색 요소 (0-255)<br>– blue: RGB 청색 요소 (0-255)<br>– alpha: 알파 (투명도) 값 (0-255)<br>– hue: HSV 색조 (0-360)<br>– saturation: HSV 채도 (0-100)<br>– value: HSV 명도 (0-100)<br>– hsl_hue: HSL 색조 (0-360)<br>– hsl_saturation: HSL 채도 (0-100)<br>– lightness: HSL 명도 (0-100)<br>– cyan: CMYK 청록색 요소 (0-100)<br>– magenta: CMYK 자홍색 요소 (0-100)<br>– yellow: CMYK 황색 요소 (0-100)<br>– black: CMYK 흑색 요소 (0-100) |
| 예제 | • `color_part('200,10,30','green')` → 10<br>• `to_int(color_part(color_cmykf(0.1,0.2,0.3,0.9),'black'))` → 90 |

### color_rgb

색상의 적색, 녹색, 청색 요소를 기반으로 색상의 문자열 표현을 반환합니다.

| | |
|---|---|
| 문법 | color_rgb(red, green, blue) |
| 인자 | • **red** - 적색 요소를 0 에서 255 사이의 정수값으로 입력<br>• **green** - 녹색 요소를 0 에서 255 사이의 정수값으로 입력<br>• **blue** - 청색 요소를 0 에서 255 사이의 정수값으로 입력 |
| 예제 | • `color_rgb(255,127,0)` → '255,127,0' |

### color_rgba

색상의 적색, 녹색, 청색 그리고 알파 (투명도) 요소를 기반으로 색상의 문자열 표현을 반환합니다.

| | |
|---|---|
| 문법 | color_rgba(red, green, blue, alpha) |
| 인자 | • **red** - 적색 요소를 0 에서 255 사이의 정수값으로 입력<br>• **green** - 녹색 요소를 0 에서 255 사이의 정수값으로 입력<br>• **blue** - 청색 요소를 0 에서 255 사이의 정수값으로 입력<br>• **alpha** - 투명도를 0(완전히 투명) 에서 255(완전히 불투명) 사이의 정수값으로 입력 |
| 예제 | • `color_rgba(255,127,0,200)` → '255,127,0,200' |

### color_rgbf

Returns a color object based on its red, green, blue and alpha components.

| | |
|---|---|
| 문법 | color_rgbf(red, green, blue, [alpha=1.0])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **red** - red component as a float value from 0.0 to 1.0<br>• **green** - green component as a float value from 0.0 to 1.0<br>• **blue** - blue component as a float value from 0.0 to 1.0<br>• **alpha** - alpha component as a float value from 0.0 to 1.0 |
| 예제 | • `color_rgbf(1.0,0.5,0)` → RGBA: 1.00,0.50,0.00,1.00 |

### create_ramp

색상 문자열 및 단계의 맵에서 나온 그레이디언트 색상표를 반환합니다.

| | |
|---|---|
| 문법 | create_ramp(map, [discrete=false])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **map** - 색상 문자열과 간격의 맵<br>• **discrete** - 개별 색상표를 생성하려면 이 파라미터를 참으로 설정하시오. |
| 예제 | • `ramp_color(create_ramp(map(0,'0,0,0',1,'255,0,0')),1)` → '255,0,0,255' |

### darker

Returns a darker (or lighter) color. Returned type is the same as color arguments, i.e. a color string representation or a color object.

| | |
|---|---|
| 문법 | darker(color, factor) |
| 인자 | • **color** - a color string or a color object<br>• **factor** - 감광 (darkening) 인자에 대응하는 정수:<br>– 이 인자가 100 을 초과하는 경우, 함수가 더 어두운 색상을 반환합니다. (예를 들어 이 인자를 200 으로 설정하면 휘도가 절반인 색상을 반환합니다.)<br>– 이 인자가 100 미만인 경우 더 밝은 색상을 반환하지만, 이런 목적을 위해서는 lighter() 함수를 사용할 것을 권장합니다.<br>– 이 인자가 0 또는 음수인 경우, 불특정 값을 반환합니다. |
| 예제 | • `darker('200,10,30', 200)` → '100,5,15,255' |

더 읽어볼 거리: *lighter*

### lighter

Returns a lighter (or darker) color. Returned type is the same as color arguments, i.e. a color string representation or a color object.

| | |
|---|---|
| 문법 | lighter(color, factor) |
| 인자 | • **color** - a color string or a color object<br>• **factor** - 증광 (lightening) 인자에 대응하는 정수:<br>– 이 인자가 100 을 초과하는 경우, 함수가 더 밝은 색상을 반환합니다. (예를 들어 이 인자를 150 으로 설정하면 휘도가 50% 더 높은 색상을 반환합니다.)<br>– 이 인자가 100 미만인 경우 더 어두운 색상을 반환하지만, 이런 목적을 위해서는 darker() 함수를 사용할 것을 권장합니다.<br>– 이 인자가 0 또는 음수인 경우, 불특정 값을 반환합니다. |
| 예제 | • `lighter('200,10,30', 200)` → '255,158,168,255' |

더 읽어볼 거리: *darker*

### project_color

프로젝트의 색상 스키마에서 색상을 반환합니다.

| | |
|---|---|
| 문법 | project_color(name) |
| 인자 | • **name** - 색상 이름 |
| 예제 | • `project_color('Logo color')` → '20,140,50' |

더 읽어볼 거리: 프로젝트 색상 설정하기

### project_color_object

Returns a color from the project's color scheme. Contrary to project_color which returns a color string representation, project_color_object returns a color object.

| | |
|---|---|
| 문법 | project_color_object(name) |
| 인자 | • **name** - 색상 이름 |
| 예제 | • `project_color_object('Logo color')` → RGBA: 0.08,0.55,0.20,1.00 |

### ramp_color

색상표에서 색상을 표현하는 문자열을 반환합니다.

#### 저장된 색상표 변수

저장한 색상표에서 색상을 표현하는 문자열을 반환합니다

| | |
|---|---|
| 문법 | ramp_color(ramp_name, value) |
| 인자 | • **ramp_name** - 색상표의 이름 문자열, 예를 들면'Spectral'<br>• **value** - 0 에서 1 사이의 실수로부터 색상을 선택하는 색상표 상의 위치 |
| 예제 | • `ramp_color('Spectral',0.3)` → '253,190,115,255' |

**참고**

사용 가능한 색상표는 QGIS 설치본마다 다릅니다. QGIS 프로젝트를 서로 다른 설치본 간에 이동하는 경우, 이 함수가 예상 결과를 제공하지 않을 수도 있습니다.

#### 표현식으로 생성된 색상표 변종

표현식으로 생성된 색상표에서 색상을 표현하는 문자열을 반환합니다.

| | |
|---|---|
| 문법 | ramp_color(ramp, value) |
| 인자 | • **ramp** - 색상표<br>• **value** - 0 에서 1 사이의 실수로부터 색상을 선택하는 색상표 상의 위치 |
| 예제 | • `ramp_color(create_ramp(map(0,'0,0,0',1,'255,0,0')),1)` → '255,0,0,255' |

더 읽어볼 거리: 색상표 설정, 색상표 드롭다운 단축키

### ramp_color_object

Returns a color object from a color ramp. Contrary to ramp_color which returns a color string representation, ramp_color_object returns a color object.

#### 저장된 색상표 변수

Returns a color object from a saved ramp

| 문법 | ramp_color_object(ramp_name, value) |
| --- | --- |
| 인자 | • **ramp_name** - 색상표의 이름 문자열, 예를 들면'Spectral'<br>• **value** - 0 에서 1 사이의 실수로부터 색상을 선택하는 색상표 상의 위치 |
| 예제 | • `ramp_color_object('Spectral',0.3)` → RGBA: 0.99,0.75,0.45,1.00 |

**참고**

사용 가능한 색상표는 QGIS 설치본마다 다릅니다. QGIS 프로젝트를 서로 다른 설치본 간에 이동하는 경우, 이 함수가 예상 결과를 제공하지 않을 수도 있습니다.

#### 표현식으로 생성된 색상표 변종

Returns a color object from an expression-created ramp

| 문법 | ramp_color_object(ramp, value) |
| --- | --- |
| 인자 | • **ramp** - 색상표<br>• **value** - 0 에서 1 사이의 실수로부터 색상을 선택하는 색상표 상의 위치 |
| 예제 | • `ramp_color_object(create_ramp(map(0,color_rgbf(0,0,0),1, color_rgbf(1,0,0))),1)` → RGBA: 1.00,0.00,0.00,1.00 |

### set_color_part

Sets a specific color component for a color string or a color object, e.g., the red component or alpha component.

| | |
|---|---|
| 문법 | set_color_part(color, component, value) |
| 인자 | • **color** - a color string or a color object<br>• **component** - 설정할 색상 요소에 대응하는 문자열입니다. 사용할 수 있는 옵션은 다음과 같습니다:<br>– red: RGB 적색 요소 (0-255)<br>– green: RGB 녹색 요소 (0-255)<br>– blue: RGB 청색 요소 (0-255)<br>– alpha: 알파 (투명도) 값 (0-255)<br>– hue: HSV 색조 (0-360)<br>– saturation: HSV 채도 (0-100)<br>– value: HSV 명도 (0-100)<br>– hsl_hue: HSL 색조 (0-360)<br>– hsl_saturation: HSL 채도 (0-100)<br>– lightness: HSL 명도 (0-100)<br>– cyan: CMYK 청록색 요소 (0-100)<br>– magenta: CMYK 자홍색 요소 (0-100)<br>– yellow: CMYK 황색 요소 (0-100)<br>– black: CMYK 흑색 요소 (0-100)<br>• **value** - 앞에 나열된 범위를 따르는, 색상 요소의 새 값 |
| 예제 | • `set_color_part('200,10,30','green',50)` → '200,50,30,255'<br>• `set_color_part(color_cmykf(0.21,0,0.92,0.70),'black',100)` → CMYKA: 0.21,0.00,0.92,1.00,1.00 |

### 9.2.4 조건 함수

이 그룹은 표현식에서 조건 검사를 처리하는 함수를 담고 있습니다.

#### CASE

일련의 조건들을 평가해서 처음으로 만족하는 조건에 대한 결과를 반환하는 데 CASE 를 사용합니다. 조건들을 연속적으로 평가하면서, 조건이 참인 경우 평가를 종료하고 대응하는 결과를 반환합니다. 어떤 조건도 참이 아닌 경우, ELSE 구문에 있는 값을 반환합니다. 또한, ELSE 구문을 설정하지 않았는데 어떤 조건도 참이 아닌 경우, NULL 을 반환합니다.

CASE

WHEN *condition* THEN *result*

[ ⋯n ]

[ ELSE *result* ]

END

[ ] 괄호는 선택적인 요소를 표시합니다.

| | |
|---|---|
| 인자 | • **WHEN condition** - 평가할 조건 표현식<br>• **THEN result** - *condition* 이 참으로 평가되는 경우 *result* 를 평가하고 반환합니다.<br>• **ELSE result** - 앞의 조건들이 모두 거짓으로 평가되는 경우 *result* 를 평가하고 반환합니다. |
| 예제 | • `CASE WHEN "name" IS NULL THEN 'None' END` → "name"필드가 NULL 인 경우'None'문자열을 반환합니다.<br>• `CASE WHEN $area > 10000 THEN 'Big property' WHEN $area > 5000 THEN 'Medium property' ELSE 'Small property' END` → 면적이 10000 을 초과하는 경우'Big property'문자열을, 면적이 5000 과 10000 사이인 경우'Medium property'를, 그 미만인 경우'Small property'를 반환합니다. |

### coalesce

표현식 목록에서 NULL 이 아닌 첫 번째 값을 반환합니다.

이 함수에는 인수 개수에 대한 제약이 없습니다.

| | |
|---|---|
| 문법 | coalesce(expression1, expression2, ⋯) |
| 인자 | • **expression** - 유형에 상관없이 무결한 모든 표현식 또는 값입니다. |
| 예제 | • `coalesce(NULL, 2)` → 2<br>• `coalesce(NULL, 2, 3)` → 2<br>• `coalesce(7, NULL, 3*2)` → 7<br>• `coalesce("fieldA", "fallbackField", 'ERROR')` → fieldA 가 NULL 이 아닐 경우 fieldA 의 값, NULL 인 경우"fallbackField"의 값, 또는 둘 다 NULL 인 경우 문자열'ERROR' |

### if

조건을 검증해서 결과에 따라 서로 다른 결과를 반환합니다.

| | |
|---|---|
| 문법 | if(condition, result_when_true, result_when_false) |
| 인자 | • **condition** - 확인해야 할 조건<br>• **result_when_true** - 조건이 참이거나, 거짓으로 변환되지 않는 또다른 값일 경우 반환될 결과입니다.<br>• **result_when_false** - 조건이 거짓이거나, 0 또는"처럼 거짓으로 변환되는 또다른 값일 경우 반환될 결과입니다. NULL 또한 거짓으로 변환될 것입니다. |
| 예제 | • `if( 1+1=2, 'Yes', 'No' )` → 'Yes'<br>• `if( 1+1=3, 'Yes', 'No' )` → 'No'<br>• `if( 5 > 3, 1, 0)` → 1<br>• `if( '', 'It is true (not empty)', 'It is false (empty)' )` → 'It is false (empty)'<br>• `if( ' ', 'It is true (not empty)', 'It is false (empty)' )` → 'It is true (not empty)'<br>• `if( 0, 'One', 'Zero' )` → 'Zero'<br>• `if( 10, 'One', 'Zero' )` → 'One' |

### nullif

value1 이 value2 와 동일한 경우 NULL 값을 반환합니다; 그렇지 않은 경우 value1 을 반환합니다. 조건에 따라 값을 NULL 로 대체하는 데 사용할 수 있습니다.

| | |
|---|---|
| 문법 | nullif(value1, value2) |
| 인자 | • **value1** - 사용해야 하거나 NULL 로 대체되어야 하는 값입니다.<br>• **value2** - NULL 대체를 촉발할 제어 값입니다. |
| 예제 | • `nullif('(none)', '(none)')` → NULL<br>• `nullif('text', '(none)')` → 'text'<br>• `nullif("name", '')` → 이름이 빈 문자열 (또는 이미 NULL) 인 경우 NULL 로 대체, 그렇지 않은 경우 이름을 유지합니다. |

#### regexp_match

유니코드 문자열 내부에서 정규 표현식과 첫번째로 일치하는 위치를 반환하거나, 하위 문자열을 찾지 못했을 경우 0 을 반환합니다.

| | |
|---|---|
| 문법 | regexp_match(input_string, regex) |
| 인자 | • **input_string** - 정규 표현식에 대해 테스트할 문자열<br>• **regex** - 테스트시킬 정규 표현식입니다. 백슬래시 문자는 이중으로 이스케이프시켜야만 합니다. (예: 공백 문자와 매칭하려면"\\s", 또는 단어 경계 (word boundary) 를 일치시키려면"\\b") |
| 예제 | • `regexp_match('QGIS ROCKS','\\sROCKS')` → 5<br>• `regexp_match('Budač','udač\\b')` → 2 |

#### try

표현식을 시도하고 오류가 없는 경우 그 값을 반환합니다. 표현식이 오류를 반환하는 경우, 대체값을 지정했다면 대체값을 반환하고, 그렇지 않다면 함수가 NULL 을 반환할 것입니다.

| | |
|---|---|
| 문법 | try(expression, [alternative])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **expression** - 실행해야 할 표현식<br>• **alternative** - 표현식이 오류를 반환하는 경우 반환될 결과입니다. |
| 예제 | • `try( to_int( '1' ), 0 )` → 1<br>• `try( to_int( 'a' ), 0 )` → 0<br>• `try( to_date( 'invalid_date' ) )` → NULL |

### 9.2.5 변환 함수

이 함수 그룹은 어떤 데이터 유형을 다른 데이터 유형으로 변환하는 함수를 담고 있습니다. (예: 문자열을 정수형으로/정수형을 문자열로, 바이너리를 문자열로/문자열을 바이너리로, 문자열을 날짜로 등등)

#### from_base64

Base64 인코딩된 문자열을 바이너리 값으로 디코딩합니다.

| | |
|---|---|
| 문법 | from_base64(string) |
| 인자 | • **string** - 해독할 문자열 |
| 예제 | • `from_base64('UUdJUw==')` → 'QGIS' |

### hash

지정한 메소드로 문자열로부터 해시를 생성합니다. 1 바이트 (8 비트) 를 16 진수"숫자 (digit)"2 개로 표현하기 때문에, 'md4'(16 바이트) 는 16 * 2 = 32 문자 길이 16 진수 문자열을 생성하고'keccak_512'(64 바이트) 는 64 * 2 = 128 문자 길이 16 진수 문자열을 생성합니다.

| | |
|---|---|
| 문법 | hash(string, method) |
| 인자 | • **string** - 해시시킬 문자열<br>• **method** - 'md4', 'md5', 'sha1', 'sha224', 'sha384', 'sha512', 'sha3_224', 'sha3_256', 'sha3_384', 'sha3_512', 'keccak_224', 'keccak_256', 'keccak_384', 'keccak_512'와 같은 해시 메소드 |
| 예제 | • `hash('QGIS', 'md4')` → 'c0fc71c241cdebb6e888cbac0e2b68eb'<br>• `hash('QGIS', 'md5')` → '57470aaa9e22adaefac7f5f342f1c6da'<br>• `hash('QGIS', 'sha1')` → 'f87cfb2b74cdd5867db913237024e7001e62b114'<br>• `hash('QGIS', 'sha224')` → '4093a619ada631c770f44bc643ead18fb393b93d6a6af1861fcfece0'<br>• `hash('QGIS', 'sha256')` → 'eb045cba7a797aaa06ac58830846e40c8e8c780bc0676d3393605fae50c053<br>• `hash('QGIS', 'sha384')` → '91c1de038cc3d09fdd512e99f9dd9922efadc39ed21d3922e69a4305cc2550<br>• `hash('QGIS', 'sha512')` → 'c2c092f2ab743bf8edbeb6d028a745f30fc720408465ed369421f0a4e20fa5e<br>• `hash('QGIS', 'sha3_224')` → '467f49a5039e7280d5d42fd433e80d203439e338eaabd701f0d6c17d'<br>• `hash('QGIS', 'sha3_256')` → '540f7354b6b8a6e735f2845250f15f4f3ba4f666c55574d9e9354575de0<br>• `hash('QGIS', 'sha3_384')` → '96052da1e77679e9a65f60d7ead961b28797782314478638 6eb43647b0<br>• `hash('QGIS', 'sha3_512')` → '900d079dc69761da113980253aa8ac0414a8bd6d09879a916228f87437<br>• `hash('QGIS', 'keccak_224')` → '5b0ce6acef8b0a121d4ac4f3eaa8503c799ad4e26a3392d1fb201478'<br>• `hash('QGIS', 'keccak_256')` → '991c520aa6815392de24087f61b2ae0fd56abbfeee4a8ca019c1011d3<br>• `hash('QGIS', 'keccak_384')` → 'c57a3aed9d856fa04e5eeee9b62b6e027cca81ba574116d3cc1f0d48a |

### md5

문자열로부터 md5 해시를 생성합니다.

| | |
|---|---|
| 문법 | md5(string) |
| 인자 | • **string** - 해시시킬 문자열 |
| 예제 | • `md5('QGIS')` → '57470aaa9e22adaefac7f5f342f1c6da' |

### sha256

문자열로부터 sha256 해시를 생성합니다.

| | |
|---|---|
| 문법 | sha256(string) |
| 인자 | • **string** - 해시시킬 문자열 |
| 예제 | • `sha256('QGIS')` → 'eb045cba7a797aaa06ac58830846e40c8e8c780bc0676d3393605fae50c05309' |

### to_base64

Base64 인코딩을 사용해서 바이너리 값을 문자열로 인코딩합니다.

| | |
|---|---|
| 문법 | to_base64(value) |
| 인자 | • **value** - 인코딩할 바이너리 값 |
| 예제 | • `to_base64('QGIS')` → 'UUdJUw==' |

### to_bool

Converts a given value to a boolean. The function will return false if the value is NULL, an empty string, an empty list, or 0.

| | |
|---|---|
| 문법 | to_bool(value) |
| 인자 | • **value** - value to convert to boolean |
| 예제 | • `to_bool('')` → false<br>• `to_bool('123')` → true<br>• `to_bool('false')` → true<br>• `to_bool(0)` → false<br>• `to_bool(1)` → true<br>• `to_bool(null)` → false |

### to_date

문자열을 날짜 객체로 변환합니다. 문자열을 파싱하기 위해 선택적인 서식 문자열을 지정할 수 있습니다; 해당 서식에 대한 추가 문서를 보고 싶다면 QDate::fromString 또는 format_date 함수 문서를 참조하세요. 기본적으로 현재 QGIS 사용자 로케일을 사용합니다.

| | |
|---|---|
| 문법 | to_date(string, [format], [language])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **string** - 날짜 값을 나타내는 문자열<br>• **format** - 문자열을 날짜 유형으로 변환하는 데 쓰이는 서식<br>• **language** - 문자열을 날짜 유형으로 변환하는 데 사용된 언어 (소문자, 2-3 문자, ISO 639 언어 코드); 기본적으로 현재 QGIS 사용자 로케일을 사용합니다. |
| 예제 | • `to_date('2012-05-04')` → 2012-05-04<br>• `to_date('June 29, 2019','MMMM d, yyyy')` → 2019-06-29, 현재 로케일이 'June'이라는 이름을 6 번째 달로 사용하는 경우가 아니라면 오류가 발생합니다.<br>• `to_date('29 juin, 2019','d MMMM, yyyy','fr')` → 2019-06-29 |

Further reading: *format_date*

### to_datetime

문자열을 날짜 & 시간 객체로 변환합니다. 문자열을 파싱하기 위해 선택적인 서식 문자열을 지정할 수 있습니다; 해당 서식에 대한 추가 문서를 보고 싶다면 QDate::fromString, QTime::fromString 또는 format_date 함수 문서를 참조하세요. 기본적으로 현재 QGIS 사용자 로케일을 사용합니다.

| | |
|---|---|
| 문법 | to_datetime(string, [format], [language])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **string** - 날짜 & 시간 값을 나타내는 문자열<br>• **format** - 문자열을 날짜 & 시간 유형으로 변환하는 데 쓰이는 서식<br>• **language** - 문자열을 날짜 & 시간 유형으로 변환하는 데 사용된 언어 (소문자, 2-3 문자, ISO 639 언어 코드); 기본적으로 현재 QGIS 사용자 로케일을 사용합니다. |
| 예제 | • `to_datetime('2012-05-04 12:50:00')` → 2012-05-04T12:50:00<br>• `to_datetime('June 29, 2019 @ 12:34','MMMM d, yyyy @ HH:mm')` → 2019-06-29T12:34, 현재 로케일이'June'이라는 이름을 6 번째 달로 사용하는 경우가 아니라면 오류가 발생합니다.<br>• `to_datetime('29 juin, 2019 @ 12:34','d MMMM, yyyy @ HH:mm', 'fr')` → 2019-06-29T12:34 |

Further reading: *format_date*

### to_decimal

도, 분, 초 좌표를 십진수 좌표로 변환합니다.

| | |
|---|---|
| 문법 | to_decimal(value) |
| 인자 | • **value** - 도, 분, 초 문자열 |
| 예제 | • `to_decimal('6°21\'16.445')` → 6.3545680555 |

### to_dm

좌표를 도, 분으로 변환합니다.

| | |
|---|---|
| 문법 | to_dm(coordinate, axis, precision, [formatting=])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **coordinate** - 경도 또는 위도 값<br>• **axis** - 좌표의 축. 'x'또는'y'입니다.<br>• **precision** - 소수점 이하 자릿수<br>• **formatting** - 서식 유형을 지정합니다. 받을 수 있는 값은 NULL(기본값), 'aligned'또는 'suffix'입니다. |
| 예제 | • `to_dm(6.1545681, 'x', 3)` → 6°9.274′<br>• `to_dm(6.1545681, 'y', 4, 'aligned')` → 6°09.2741′N<br>• `to_dm(6.1545681, 'y', 4, 'suffix')` → 6°9.2741′N |

### to_dms

좌표를 도, 분, 초로 변환합니다.

| | |
|---|---|
| 문법 | to_dms(coordinate, axis, precision, [formatting=])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **coordinate** - 경도 또는 위도 값<br>• **axis** - 좌표의 축. 'x'또는'y'입니다.<br>• **precision** - 소수점 이하 자릿수<br>• **formatting** - 서식 유형을 지정합니다. 받을 수 있는 값은 NULL(기본값), 'aligned'또는 'suffix'입니다. |
| 예제 | • `to_dms(6.1545681, 'x', 3)` → 6°9′16.445″<br>• `to_dms(6.1545681, 'y', 4, 'aligned')` → 6°09′16.4452″N<br>• `to_dms(6.1545681, 'y', 4, 'suffix')` → 6°9′16.4452″N |

### to_int

Converts a string to integer number. If a value cannot be converted to integer the expression is invalid (e.g '123asd'is invalid).

| | |
|---|---|
| 문법 | to_int(string) |
| 인자 | • **string** - 정수형 숫자로 변환할 문자열 |
| 예제 | • `to_int('123')` → 123 |

### to_interval

문자열을 간격 유형으로 변환합니다. 날짜 유형의 월, 일, 시 등을 추출하는 데 사용할 수 있습니다.

| | |
|---|---|
| 문법 | to_interval(string) |
| 인자 | • **string** - 간격을 나타내는 문자열. 사용할 수 있는 서식은 {n} days {n} hours {n} months 를 포함합니다. |
| 예제 | • `to_interval('1 day 2 hours')` → interval: 1.08333 days<br>• `to_interval( '0.5 hours' )` → interval: 30 minutes<br>• `to_datetime('2012-05-05 12:00:00') - to_interval('1 day 2 hours')` → 2012-05-04T10:00:00 |

### to_real

Converts a string to a real number. If a value cannot be converted to real the expression is invalid (e.g '123.56asd'is invalid). Numbers are rounded after saving changes if the precision is smaller than the result of the conversion.

| | |
|---|---|
| 문법 | to_real(string) |
| 인자 | • **string** - 실수형 숫자로 변환할 문자열 |
| 예제 | • `to_real('123.45')` → 123.45 |

#### to_string

Converts a number to string. The conversion is not locale-aware, see 'format_number' for a locale-aware alternative.

| | |
|---|---|
| 문법 | to_string(number) |
| 인자 | • **number** - 정수형 또는 실수형 값. 문자열로 변환할 숫자입니다. |
| 예제 | • `to_string(1.23)` → '1.23' |

Further reading: *format_number*

#### to_time

문자열을 시간 객체로 변환합니다. 문자열을 파싱하기 위해 부가적인 서식 문자열을 지정할 수 있습니다; 해당 서식에 대한 추가 문서를 보고 싶다면 QTime::fromString 을 참조하세요.

| | |
|---|---|
| 문법 | to_time(string, [format], [language])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **string** - 시간 값을 나타내는 문자열<br>• **format** - 문자열을 시간 유형으로 변환하는 데 쓰이는 서식<br>• **language** - 문자열을 시간 유형으로 변환하는 데 사용된 언어 (소문자, 2-3 문자, ISO 639 언어 코드) |
| 예제 | • `to_time('12:30:01')` → 12:30:01<br>• `to_time('12:34','HH:mm')` → 12:34:00<br>• `to_time('12:34','HH:mm','fr')` → 12:34:00 |

Further reading: *format_date*

### 9.2.6 CRS Functions

This group contains functions to operate on coordinate reference system objects.

#### crs_from_text

Creates a coordinate reference system from a string definition. The string can represent an authority ID, a WKT definition, or a PROJ string definition of the CRS.

| | |
|---|---|
| 문법 | crs_from_text(definition) |
| 인자 | • **definition** - CRS definition |
| 예제 | • `crs_from_text('EPSG:3857')` → crs value 'EPSG:3857 - WGS 84 / Pseudo-Mercator'<br>• `crs_from_text('PROJ:+proj=merc +a=6378137 +b=6378137 +lat_ts=0 +lon_0=0 +x_0=0 +y_0=0 +k=1 +units=m +nadgrids=@null +wktext +no_defs +type=crs')` → crs value 'EPSG:3857 - WGS 84 / Pseudo-Mercator' |

#### crs_to_authid

Returns the authority ID string for a coordinate reference system.

| | |
|---|---|
| 문법 | crs_to_authid(crs) |
| 인자 | • **crs** - crs value |
| 예제 | • `crs_to_authid(crs_from_text('PROJ:+proj=merc +a=6378137 +b=6378137 +lat_ts=0 +lon_0=0 +x_0=0 +y_0=0 +k=1 +units=m +nadgrids=@null +wktext +no_defs +type=crs'))` → 'EPSG:3857' |

### 9.2.7 사용자 지정 함수

이 그룹은 사용자가 생성한 함수를 담고 있습니다. 자세한 내용은함수 편집기 를 참조하세요.

### 9.2.8 날짜 및 시간 함수

이 함수 그룹은 날짜 및 시간 데이터를 처리하기 위한 함수들을 담고 있습니다. 이 그룹은변환 함수 그룹과 몇몇 함수를 (to_date, to_time, to_datetime, to_interval) 그리고 ref:*string_functions* 그룹과도 함수를 (format_date) 공유합니다.

> **참고**
>
> **필드에 날짜, 날짜 & 시간, 간격 유형을 저장하기**
>
> 데이터소스 제공자에 따라 날짜, 시간, 날짜 & 시간 값을 필드에 직접 저장할 수 있습니다. (예를 들어 Shapefile 은 날짜 유형은 받아들이지만 날짜 & 시간 또는 시간 유형은 받아들이지 못 합니다.) 다음은 이 제약 사항을 넘어서기 위한 몇 가지 제안입니다.
>
> - *format_date()* 함수를 사용하면 텍스트 유형 필드에 날짜, 날짜 & 시간, 시간 을 저장할 수 있습니다.
> - 간격 을 날짜 추출 함수 중 하나로 처리한 다음 (예: *day()* 함수는 날짜로 표현된 간격을 반환합니다) 정수 또는 실수 유형 필드에 저장할 수 있습니다.

### age

두 날짜 또는 두 날짜 & 시간 사이의 차이를 간격 유형으로 반환합니다.

그 차를 `Interval` 로 반환하는데, 쓸만한 정보를 추출하려면 다음 함수들 가운데 하나와 함께 사용되어야 합니다:

- `year`
- `month`
- `week`
- `day`
- `hour`
- `minute`
- `second`

| | |
|---|---|
| 문법 | age(datetime1, datetime2) |
| 인자 | • **datetime1** - 뒷 날짜를 나타내는 문자열, 날짜 또는 날짜 & 시간<br>• **datetime2** - 앞 날짜를 나타내는 문자열, 날짜 또는 날짜 & 시간 |
| 예제 | • `day(age('2012-05-12','2012-05-02'))` → 10<br>• `hour(age('2012-05-12','2012-05-02'))` → 240 |

### datetime_from_epoch

협정 세계시 (Qt.UTC) 1970-01-01T00:00:00.000 부터 셈한 밀리초 (msec) 의 개수를 Qt.LocalTime 으로 변환한 날짜 및 시간인 날짜 & 시간을 반환합니다.

| | |
|---|---|
| 문법 | datetime_from_epoch(int) |
| 인자 | • **int** - 숫자 (밀리초) |
| 예제 | • `datetime_from_epoch(1483225200000)` → 2017-01-01T00:00:00 |

### day

날짜 유형에서 일 (□) 부분을, 또는 간격 유형에서 일수를 추출합니다.

**날짜 변이형**

날짜 또는 날짜 & 시간에서 일 부분을 추출합니다.

| | |
|---|---|
| 문법 | day(date) |
| 인자 | • **date** - 날짜 또는 날짜 & 시간 값 |
| 예제 | • `day('2012-05-12')` → 12 |

#### 간격 변이형

길이를 간격의 일 단위로 계산합니다.

| | |
|---|---|
| 문법 | day(interval) |
| 인자 | • **interval** - 일 수를 반환할 간격 값 |
| 예제 | • `day(to_interval('3 days'))` → 3<br>• `day(to_interval('3 weeks 2 days'))` → 23<br>• `day(age('2012-01-01','2010-01-01'))` → 730 |

### day_of_week

지정한 날짜 또는 날짜 & 시간에서 해당 주의 요일을 반환합니다. 반환될 값 범위는 0 에서 6 이며, 0 은 일요일, 6 은 토요일입니다.

| | |
|---|---|
| 문법 | day_of_week(date) |
| 인자 | • **date** - 날짜 또는 날짜 & 시간 값 |
| 예제 | • `day_of_week(to_date('2015-09-21'))` → 1 |

### epoch

유닉스 원기 (unix epoch) 와 지정한 날짜값 사이의 간격을 밀리초 단위로 반환합니다.

| | |
|---|---|
| 문법 | epoch(date) |
| 인자 | • **date** - 날짜 또는 날짜 & 시간 값 |
| 예제 | • `epoch(to_date('2017-01-01'))` → 1483203600000 |

### format_date

날짜 유형 또는 문자열을 사용자 정의 문자열 서식으로 변환합니다. Qt 날짜 & 시간 서식 문자열을 사용하십시오. QDateTime::toString 을 참조하세요.

| | |
|---|---|
| 문법 | format_date(datetime, format, [language])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **datetime** - 날짜, 시간 또는 날짜 & 시간 값<br>• **format** - 문자열을 서식화하는 데 쓰이는 문자열 템플릿 |

| 표현식 | 산출물 |
|---|---|
| d | 앞에 0 이 없는 숫자 일 (1 에서 31) |
| dd | 앞에 0 이 있는 숫자 일 (01 에서 31) |
| ddd | 축약된 현지 요일 이름 (예: 'Mon'에서'Sun') |
| dddd | 축약하지 않은 현지 요일 이름 (예: 'Monday'에서'Sunday') |
| M | 앞에 0 이 없는 숫자 월 (1-12) |
| MM | 앞에 0 이 있는 숫자 월 (01-12) |
| MMM | 축약된 현지 월 이름 (예: 'Jan'에서'Dec') |
| MMMM | 축약하지 않은 현지 월 이름 (예: 'January'에서'December') |
| yy | 두 자리 숫자 연도 (00-99) |
| yyyy | 네 자리 숫자 연도 |

다음 표현식은 서식 문자열의 시간 부분에 쓰일 수도 있습니다:

| 표현식 | 산출물 |
|---|---|
| h | 앞에 0 이 없는 시간 (0 에서 23 또는 AM/PM 표시가 있는 경우 1 에서 12) |
| hh | 앞에 0 이 있는 시간 (00 에서 23 또는 AM/PM 표시가 있는 경우 01 에서 12) |
| H | 앞에 0 이 없는 시간 (AM/PM 표시가 있는 경우에도 0 에서 23) |
| HH | 앞에 0 이 있는 시간 (AM/PM 표시가 있는 경우에도 00 에서 23) |
| m | 앞에 0 이 없는 분 (0 에서 59) |
| mm | 앞에 0 이 있는 분 (00 에서 59) |
| s | 앞에 0 이 없는 초 (0 에서 59) |
| ss | 앞에 0 이 있는 초 (00 에서 59) |
| z | 뒤에 0 이 없는 밀리초 (0 에서 999) |
| zzz | 뒤에 0 이 있는 밀리초 (000 에서 999) |
| AP or A | AM/PM 시간으로 해석합니다. *AP* 는'AM'또는'PM'이어야만 합니다. |
| ap or a | AM/PM 시간으로 해석합니다. *ap* 는'am'또는'pm'이어야만 합니다. |

• **language** - 날짜 유형을 사용자 지정 문자열로 서식화하는 데 사용된 언어 (소문자, 2-3 문자, ISO 639 언어 코드); 기본적으로 현재 QGIS 사용자 로케일을 사용합니다.

예제

- `format_date('2012-05-15','dd.MM.yyyy')` → '15.05.2012'
- `format_date('2012-05-15','d MMMM yyyy','fr')` → '15 mai 2012'
- `format_date('2012-05-15','dddd')` → 'Tuesday', 현재 로케일이 영어 변이형인 경우
- `format_date('2012-05-15 13:54:20','dd.MM.yy')` → '15.05.12'
- `format_date('13:54:20','hh:mm AP')` → '01:54 PM'

### hour

날짜 & 시간 또는 시간에서 시 부분을, 또는 간격에서 시의 개수를 추출합니다.

#### 시간 변이형

시간 또는 날짜 & 시간에서 시 부분을 추출합니다.

| 문법 | hour(datetime) |
|---|---|
| 인자 | • **datetime** - 시간 또는 날짜 & 시간 값 |
| 예제 | • `hour( to_datetime('2012-07-22 13:24:57') )` → 13 |

#### 간격 변이형

길이를 간격의 시간 단위로 계산합니다.

| 문법 | hour(interval) |
|---|---|
| 인자 | • **interval** - 시간 개수를 반환할 간격 값 |
| 예제 | • `hour(to_interval('3 hours'))` → 3<br>• `hour(age('2012-07-22T13:00:00','2012-07-22T10:00:00'))` → 3<br>• `hour(age('2012-01-01','2010-01-01'))` → 17520 |

### make_date

년, 월, 일 숫자로부터 날짜값을 생성합니다.

| 문법 | make_date(year, month, day) |
|---|---|
| 인자 | • **year** - 년도 번호. 1 에서 99 까지의 년도를 그대로 해석합니다. 0 년도는 무결하지 않습니다.<br>• **month** - 월 번호, 1=January<br>• **day** - 일 번호, 월의 첫날을 1 로 시작 |
| 예제 | • `make_date(2020,5,4)` → date value 2020-05-04 |

### make_datetime

년, 월, 일, 시, 분, 초 숫자로부터 날짜 & 시간 값을 생성합니다.

| | |
|---|---|
| 문법 | make_datetime(year, month, day, hour, minute, second) |
| 인자 | • **year** - 년도 번호. 1 에서 99 까지의 년도를 그대로 해석합니다. 0 년도는 무결하지 않습니다.<br>• **month** - 월 번호, 1=January<br>• **day** - 일 번호, 월의 첫날을 1 로 시작<br>• **hour** - 시간 번호<br>• **minute** - 분<br>• **second** - 초 (밀리초를 포함한 분수값) |
| 예제 | • `make_datetime(2020,5,4,13,45,30.5)` → 날짜 & 시간 값 2020-05-04 13:45:30.500 |

### make_interval

년, 월, 주, 일, 시, 분, 초 값으로부터 간격값을 생성합니다.

| | |
|---|---|
| 문법 | make_interval([years=0], [months=0], [weeks=0], [days=0], [hours=0], [minutes=0], [seconds=0])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **years** - 연도 개수 (1 년 길이를 365.25 일로 가정)<br>• **months** - 월 개수 (1 개월 길이를 30 일로 가정)<br>• **weeks** - 주 개수<br>• **days** - 일 개수<br>• **hours** - 시간 개수<br>• **minutes** - 분 개수<br>• **seconds** - 초 개수 |
| 예제 | • `make_interval(hours:=3)` → interval: 3 hours<br>• `make_interval(days:=2, hours:=3)` → interval: 2.125 days<br>• `make_interval(minutes:=0.5, seconds:=5)` → interval: 35 seconds |

### make_time

시, 분, 초 숫자로부터 시간값을 생성합니다.

| 문법 | make_time(hour, minute, second) |
| --- | --- |
| 인자 | • **hour** - 시간 번호<br>• **minute** - 분<br>• **second** - 초 (밀리초를 포함한 분수값) |
| 예제 | • `make_time(13,45,30.5)` → time value 13:45:30.500 |

### minute

날짜 & 시간 또는 시간에서 분 부분을, 또는 간격에서 분의 개수를 추출합니다.

**시간 변이형**

시간 또는 날짜 & 시간에서 분 부분을 추출합니다.

| 문법 | minute(datetime) |
| --- | --- |
| 인자 | • **datetime** - 시간 또는 날짜 & 시간 값 |
| 예제 | • `minute( to_datetime('2012-07-22 13:24:57') )` → 24 |

**간격 변이형**

길이를 간격의 분 단위로 계산합니다.

| 문법 | minute(interval) |
| --- | --- |
| 인자 | • **interval** - 분의 개수를 반환할 간격 값 |
| 예제 | • `minute(to_interval('3 minutes'))` → 3<br>• `minute(age('2012-07-22T00:20:00','2012-07-22T00:00:00'))` → 20<br>• `minute(age('2012-01-01','2010-01-01'))` → 1051200 |

### month

날짜 유형에서 월 부분을, 또는 간격 유형에서 개월수를 추출합니다.

**날짜 변이형**

날짜 또는 날짜 & 시간에서 월 부분을 추출합니다.

| | |
|---|---|
| 문법 | month(date) |
| 인자 | • **date** - 날짜 또는 날짜 & 시간 값 |
| 예제 | • `month('2012-05-12')` → 05 |

#### 간격 변이형

길이를 간격의 월 단위로 계산합니다.

| | |
|---|---|
| 문법 | month(interval) |
| 인자 | • **interval** - 개월수를 반환할 간격 값 |
| 예제 | • `month(to_interval('3 months'))` → 3<br>• `month(age('2012-01-01','2010-01-01'))` → 4.03333 |

### now

현재 날짜와 시간을 반환합니다. 이 함수는 정적이며 평가하는 동안 일관된 결과를 반환할 것입니다. 반환된 시간은 표현식이 준비된 시간입니다.

| | |
|---|---|
| 문법 | now() |
| 예제 | • `now()` → 2012-07-22T13:24:57 |

### second

날짜 & 시간 또는 시간에서 초 부분을, 또는 간격에서 초의 개수를 추출합니다.

#### 시간 변이형

시간 또는 날짜 & 시간에서 초 부분을 추출합니다.

| | |
|---|---|
| 문법 | second(datetime) |
| 인자 | • **datetime** - 시간 또는 날짜 & 시간 값 |
| 예제 | • `second( to_datetime('2012-07-22 13:24:57') )` → 57 |

#### 간격 변이형

길이를 간격의 초 단위로 계산합니다.

| 문법 | second(interval) |
| --- | --- |
| 인자 | • **interval** - 초의 개수를 반환할 간격 값 |
| 예제 | • `second(to_interval('3 minutes'))` → 180<br>• `second(age('2012-07-22T00:20:00','2012-07-22T00:00:00'))` → 1200<br>• `second(age('2012-01-01','2010-01-01'))` → 63072000 |

### to_date

문자열을 날짜 객체로 변환합니다. 문자열을 파싱하기 위해 선택적인 서식 문자열을 지정할 수 있습니다; 해당 서식에 대한 추가 문서를 보고 싶다면 QDate::fromString 또는 format_date 함수 문서를 참조하세요. 기본적으로 현재 QGIS 사용자 로케일을 사용합니다.

| 문법 | to_date(string, [format], [language])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| --- | --- |
| 인자 | • **string** - 날짜 값을 나타내는 문자열<br>• **format** - 문자열을 날짜 유형으로 변환하는 데 쓰이는 서식<br>• **language** - 문자열을 날짜 유형으로 변환하는 데 사용된 언어 (소문자, 2-3 문자, ISO 639 언어 코드); 기본적으로 현재 QGIS 사용자 로케일을 사용합니다. |
| 예제 | • `to_date('2012-05-04')` → 2012-05-04<br>• `to_date('June 29, 2019','MMMM d, yyyy')` → 2019-06-29, 현재 로케일이 'June'이라는 이름을 6 번째 달로 사용하는 경우가 아니라면 오류가 발생합니다.<br>• `to_date('29 juin, 2019','d MMMM, yyyy','fr')` → 2019-06-29 |

Further reading: *format_date*

### to_datetime

문자열을 날짜 & 시간 객체로 변환합니다. 문자열을 파싱하기 위해 선택적인 서식 문자열을 지정할 수 있습니다; 해당 서식에 대한 추가 문서를 보고 싶다면 QDate::fromString, QTime::fromString 또는 format_date 함수 문서를 참조하세요. 기본적으로 현재 QGIS 사용자 로케일을 사용합니다.

| | |
|---|---|
| 문법 | to_datetime(string, [format], [language])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **string** - 날짜 & 시간 값을 나타내는 문자열<br>• **format** - 문자열을 날짜 & 시간 유형으로 변환하는 데 쓰이는 서식<br>• **language** - 문자열을 날짜 & 시간 유형으로 변환하는 데 사용된 언어 (소문자, 2-3 문자, ISO 639 언어 코드); 기본적으로 현재 QGIS 사용자 로케일을 사용합니다. |
| 예제 | • `to_datetime('2012-05-04 12:50:00')` → 2012-05-04T12:50:00<br>• `to_datetime('June 29, 2019 @ 12:34','MMMM d, yyyy @ HH:mm')` → 2019-06-29T12:34, 현재 로케일이'June'이라는 이름을 6 번째 달로 사용하는 경우가 아니라면 오류가 발생합니다.<br>• `to_datetime('29 juin, 2019 @ 12:34','d MMMM, yyyy @ HH:mm', 'fr')` → 2019-06-29T12:34 |

Further reading: *format_date*

### to_interval

문자열을 간격 유형으로 변환합니다. 날짜 유형의 월, 일, 시 등을 추출하는 데 사용할 수 있습니다.

| | |
|---|---|
| 문법 | to_interval(string) |
| 인자 | • **string** - 간격을 나타내는 문자열. 사용할 수 있는 서식은 {n} days {n} hours {n} months 를 포함합니다. |
| 예제 | • `to_interval('1 day 2 hours')` → interval: 1.08333 days<br>• `to_interval( '0.5 hours' )` → interval: 30 minutes<br>• `to_datetime('2012-05-05 12:00:00') - to_interval('1 day 2 hours')` → 2012-05-04T10:00:00 |

### to_time

문자열을 시간 객체로 변환합니다. 문자열을 파싱하기 위해 부가적인 서식 문자열을 지정할 수 있습니다; 해당 서식에 대한 추가 문서를 보고 싶다면 QTime::fromString 을 참조하세요.

| | |
|---|---|
| 문법 | to_time(string, [format], [language])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **string** - 시간 값을 나타내는 문자열<br>• **format** - 문자열을 시간 유형으로 변환하는 데 쓰이는 서식<br>• **language** - 문자열을 시간 유형으로 변환하는 데 사용된 언어 (소문자, 2-3 문자, ISO 639 언어 코드) |
| 예제 | • `to_time('12:30:01')` → 12:30:01<br>• `to_time('12:34','HH:mm')` → 12:34:00<br>• `to_time('12:34','HH:mm','fr')` → 12:34:00 |

Further reading: *format_date*

### week

날짜 유형에서 주 (☐) 번호를, 또는 간격 유형에서 주의 개수를 추출합니다.

**날짜 변이형**

날짜 또는 날짜 & 시간에서 주의 번호를 추출합니다.

| | |
|---|---|
| 문법 | week(date) |
| 인자 | • **date** - 날짜 또는 날짜 & 시간 값 |
| 예제 | • `week('2012-05-12')` → 19 |

**간격 변이형**

길이를 간격의 주 단위로 계산합니다.

| | |
|---|---|
| 문법 | week(interval) |
| 인자 | • **interval** - 개월수를 반환할 간격 값 |
| 예제 | • `week(to_interval('3 weeks'))` → 3<br>• `week(age('2012-01-01','2010-01-01'))` → 104.285 |

### year

날짜 유형에서 연도 부분을, 또는 간격 유형에서 연수를 추출합니다.

**날짜 변이형**

날짜 또는 날짜 & 시간에서 연도 부분을 추출합니다.

| | |
|---|---|
| 문법 | year(date) |
| 인자 | • **date** - 날짜 또는 날짜 & 시간 값 |
| 예제 | • `year('2012-05-12')` → 2012 |

**간격 변이형**

길이를 간격의 연 단위로 계산합니다.

| | |
|---|---|
| 문법 | year(interval) |
| 인자 | • **interval** - 년의 개수를 반환할 간격 값 |
| 예제 | • `year(to_interval('3 years'))` → 3<br>• `year(age('2012-01-01','2010-01-01'))` → 1.9986 |

**다음은 몇몇 예시입니다:**

이런 함수들 외에도, - (빼기) 연산자를 통해 날짜, 날짜 & 시간 또는 시간 유형을 뺄셈하면 간격 유형을 반환할 것입니다.

+ (더하기) 또는 - (빼기) 연산자를 이용해서 날짜, 날짜 & 시간 또는 시간 유형에 간격 유형을 더하거나 빼면 날짜 & 시간 유형을 반환합니다.

- QGIS 3.0 배포일까지 며칠 남았는지 알아보려면:

```
to_date('2017-09-29') - to_date(now())
-- Returns <interval: 203 days>
```

- 같은 내용을 시간 유형으로:

```
to_datetime('2017-09-29 12:00:00') - now()
-- Returns <interval: 202.49 days>
```

- 현재부터 100 일 후의 날짜 & 시간을 얻으려면:

```
now() + to_interval('100 days')
-- Returns <datetime: 2017-06-18 01:00:00>
```

### 9.2.9 필드 및 값

활성 레이어로부터 나온 필드 목록, 그리고 특수값을 담고 있습니다. 필드 목록에는 데이터셋에 저장된 필드, 가상 필드 및보조 필드 는 물론결합 으로 생성된 필드도 포함됩니다.

사용자 표현식에 필드명을 추가하려면 필드명을 더블클릭하십시오. 필드명을 (가급적 큰따옴표 안에) 입력하거나, 또는 필드의별명 을 입력해도 됩니다.

표현식에 사용할 필드값을 가져오려면, 적절한 필드를 선택한 다음 나타나는 위젯에서 *10 Samples* 와 *All Unique* 가운데 하나를 선택하십시오. 요청한 값들이 표시되는 목록 맨 위에 있는 *Search* 란을 통해 결과를 필터링할 수 있습니다. 필드를 오른쪽 클릭해서도 표본 값에 접근할 수 있습니다.

작성 중인 표현식에 값을 추가하려면, 목록에 있는 값을 더블클릭하십시오. 해당 값이 문자열 유형인 경우 작은따옴표를 추가해야 합니다. 그 외 유형은 따옴표가 필요하지 않습니다.

#### NULL

NULL 값과 동등합니다.

| 문법 | NULL |
|---|---|
| 예제 | • `NULL` → NULL 값 |

> **참고**
>
> NULL 에 대해 테스트하려면 *IS NULL* 또는 *IS NOT NULL* 표현식을 이용하십시오.

### 9.2.10 파일 및 경로 함수

이 그룹은 파일 및 경로 이름을 처리하는 함수를 담고 있습니다.

#### base_file_name

디렉터리 또는 파일 확장자 없이 파일의 기본 이름 (base name) 을 반환합니다.

| 문법 | base_file_name(path) |
|---|---|
| 인자 | • **path** - 파일 경로 또는 맵 레이어 값. 맵 레이어 값이 지정된 경우 레이어의 파일 소스를 사용할 것입니다. |
| 예제 | • `base_file_name('/home/qgis/data/country_boundaries.shp')` → 'country_boundaries' |

### exif

이미지 파일로부터 Exif(Exchangeable image file format) 태그 값을 가져옵니다.

| | |
|---|---|
| 문법 | exif(path, [tag])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **path** - 이미지 파일 경로 또는 맵 레이어 값. 맵 레이어 값이 지정된 경우 레이어의 파일 소스를 사용할 것입니다.<br>• **tag** - 반환할 태그. 지정하지 않으면 모든 Exif 태그 값을 가진 맵을 반환할 것입니다. |
| 예제 | • `exif('/my/photo.jpg','Exif.Image.Orientation')` → 0 |

### file_exists

파일 경로가 실재하는 경우 참을 반환합니다.

| | |
|---|---|
| 문법 | file_exists(path) |
| 인자 | • **path** - 파일 경로 또는 맵 레이어 값. 맵 레이어 값이 지정된 경우 레이어의 파일 소스를 사용할 것입니다. |
| 예제 | • `file_exists('/home/qgis/data/country_boundaries.shp')` → TRUE |

### file_name

디렉터리를 제외한 (파일 확장자 포함) 파일 이름을 반환합니다.

| | |
|---|---|
| 문법 | file_name(path) |
| 인자 | • **path** - 파일 경로 또는 맵 레이어 값. 맵 레이어 값이 지정된 경우 레이어의 파일 소스를 사용할 것입니다. |
| 예제 | • `file_name('/home/qgis/data/country_boundaries.shp')` → 'country_boundaries.shp' |

### file_path

파일 경로의 디렉터리 요소를 반환합니다. 파일 이름을 포함하지 않습니다.

| | |
|---|---|
| 문법 | file_path(path) |
| 인자 | • **path** - 파일 경로 또는 맵 레이어 값. 맵 레이어 값이 지정된 경우 레이어의 파일 소스를 사용할 것입니다. |
| 예제 | • `file_path('/home/qgis/data/country_boundaries.shp')` → '/home/qgis/data' |

### file_size

파일의 (바이트 단위) 용량을 반환합니다.

| | |
|---|---|
| 문법 | file_size(path) |
| 인자 | • **path** - 파일 경로 또는 맵 레이어 값. 맵 레이어 값이 지정된 경우 레이어의 파일 소스를 사용할 것입니다. |
| 예제 | • `file_size('/home/qgis/data/country_boundaries.geojson')` → 5674 |

### file_suffix

파일 경로에서 파일 접미어 (확장자) 를 반환합니다.

| | |
|---|---|
| 문법 | file_suffix(path) |
| 인자 | • **path** - 파일 경로 또는 맵 레이어 값. 맵 레이어 값이 지정된 경우 레이어의 파일 소스를 사용할 것입니다. |
| 예제 | • `file_suffix('/home/qgis/data/country_boundaries.shp')` → 'shp' |

#### is_directory

경로가 디렉터리에 대응하는 경우 참을 반환합니다.

| | |
|---|---|
| 문법 | is_directory(path) |
| 인자 | • **path** - 파일 경로 또는 맵 레이어 값. 맵 레이어 값이 지정된 경우 레이어의 파일 소스를 사용할 것입니다. |
| 예제 | • `is_directory('/home/qgis/data/country_boundaries.shp')` → FALSE<br>• `is_directory('/home/qgis/data/')` → TRUE |

#### is_file

경로가 파일에 대응하는 경우 참을 반환합니다.

| | |
|---|---|
| 문법 | is_file(path) |
| 인자 | • **path** - 파일 경로 또는 맵 레이어 값. 맵 레이어 값이 지정된 경우 레이어의 파일 소스를 사용할 것입니다. |
| 예제 | • `is_file('/home/qgis/data/country_boundaries.shp')` → TRUE<br>• `is_file('/home/qgis/data/')` → FALSE |

### 9.2.11 양식 함수

이 함수 그룹은 속성 양식 맥락에서만, 예를 들면필드 위젯 설정에서 실행되는 함수들을 담고 있습니다.

#### current_parent_value

내장 양식 맥락에서만 사용할 수 있는 이 함수는 현재 편집되고 있는 부모 양식에 있는 필드의 현재 저장되지 않은 값을 반환합니다. 현재 편집되고 있는 또는 부모 레이어에 아직 추가되지 않은 피처에 대한 부모 피처의 실제 속성값과는 다를 것입니다. 값-관계 위젯 필터 표현식에서 사용되는 경우, 이 함수를 양식을 내장 맥락에서 사용하지 않는 경우 레이어에서 실제 부모 피처를 가져올 수 있는'coalesce()'형태로 감싸야 합니다.

| | |
|---|---|
| 문법 | current_parent_value(field_name) |
| 인자 | • **field_name** - 현재 부모 양식에 있는 필드 이름 |
| 예제 | • `current_parent_value( 'FIELD_NAME' )` → 부모 양식에 있는'FIELD_NAME' 필드의 현재 값입니다. |

#### current_value

현재 편집중인 양식 또는 테이블 행에 있는 필드의 현재 저장되지 않은 값을 반환합니다. 이 객체의 속성은 현재 편집 중이거나 아직 레이어에 추가되지 않은 객체의 실제 속성 값과 다릅니다.

| | |
|---|---|
| 문법 | current_value(field_name) |
| 인자 | • **field_name** - 현재 양식 또는 테이블 행의 필드 이름 |
| 예제 | • `current_value( 'FIELD_NAME' )` → 'FIELD_NAME'필드의 현재 값입니다. |

### 9.2.12 퍼지 매칭 함수

이 그룹은 값들을 퍼지 비교하기 위한 함수를 담고 있습니다.

#### hamming_distance

두 문자열 사이의 해밍 거리 (Hamming distance) 를 반환합니다. 이는 입력 문자열들 내에서 대응하는 위치에 있는 문자들이 서로 다른 경우의 개수와 같습니다. 입력 문자열은 동일한 길이여야 하며 대소문자를 구분해서 비교합니다.

| | |
|---|---|
| 문법 | hamming_distance(string1, string2) |
| 인자 | • **string1** - 문자열<br>• **string2** - 문자열 |
| 예제 | • `hamming_distance('abc','xec')` → 2<br>• `hamming_distance('abc','ABc')` → 2<br>• `hamming_distance(upper('abc'),upper('ABC'))` → 0<br>• `hamming_distance('abc','abcd')` → NULL |

#### levenshtein

두 문자열 사이의 레벤시테인 편집 거리 (Levenshtein edit distance) 를 반환합니다. 이는 어떤 문자열을 다른 문자열로 변경하는 데 필요한 문자 편집 (삽입, 삭제 또는 치환) 의 최소 횟수와 같습니다.

레벤시테인 거리란 두 문자열 사이의 유사성을 측정하는 방법입니다. 거리가 짧을수록 두 문자열이 더 유사하고, 거리가 멀수록 두 문자열이 더 다르다는 의미입니다. 이 거리는 대소문자를 구분합니다.

| 문법 | levenshtein(string1, string2) |
| --- | --- |
| 인자 | • **string1** - 문자열<br>• **string2** - 문자열 |
| 예제 | • `levenshtein('kittens','mitten')` → 2<br>• `levenshtein('Kitten','kitten')` → 1<br>• `levenshtein(upper('Kitten'),upper('kitten'))` → 0 |

### longest_common_substring

두 문자열 사이의 가장 긴 공통 하위 문자열을 반환합니다. 이 하위 문자열은 두 입력 문자열의 하위 문자열 가운데 동일하면서 가장 긴 문자열입니다. 예를 들어, "ABABC"와"BABCA"의 가장 긴 공통 하위 문자열은"ABC"입니다. 하위 문자열은 대소문자를 구분합니다.

| 문법 | longest_common_substring(string1, string2) |
| --- | --- |
| 인자 | • **string1** - 문자열<br>• **string2** - 문자열 |
| 예제 | • `longest_common_substring('ABABC','BABCA')` → 'BABC'<br>• `longest_common_substring('abcDeF','abcdef')` → 'abc'<br>• `longest_common_substring(upper('abcDeF'),upper('abcdex'))` → 'ABCDE' |

### soundex

문자열의 사운덱스 표현 (Soundex representation) 을 반환합니다. 사운덱스란 유사 발음 검색 알고리즘으로, 문자열이 서로 비슷하게 발음되는 경우 동일한 사운덱스 코드로 표현될 것입니다.

| 문법 | soundex(string) |
| --- | --- |
| 인자 | • **string** - 문자열 |
| 예제 | • `soundex('robert')` → 'R163'<br>• `soundex('rupert')` → 'R163'<br>• `soundex('rubin')` → 'R150' |

### 9.2.13 일반 함수

이 그룹은 일반으로 분류된 함수를 담고 있습니다.

#### env

환경 변수를 받아 그 내용을 문자열로 반환합니다. 변수를 찾을 수 없는 경우, NULL 을 반환할 것입니다. 드라이브 문자 또는 경로 접두어 같은 시스템 전용 환경 설정을 주입하는 데 편리합니다. 환경 변수의 정의는 운영 체제에 따라 달라지기 때문에, 이를 어떻게 설정할 수 있는지에 대해 시스템 관리자 또는 운영 체제 문서를 확인해보십시오.

| | |
|---|---|
| 문법 | env(name) |
| 인자 | • **name** - 추출해야 할 환경 변수의 이름입니다. |
| 예제 | • `env( 'LANG' )` → 'en_US.UTF-8'<br>• `env( 'MY_OWN_PREFIX_VAR' )` → 'Z:'<br>• `env( 'I_DO_NOT_EXIST' )` → NULL |

#### eval

문자열로 전달된 표현식을 평가합니다. 컨텍스트 변수 또는 필드로 전달되는 동적 파라미터를 확장하는 데 유용합니다.

| | |
|---|---|
| 문법 | eval(expression) |
| 인자 | • **expression** - 표현식 문자열 |
| 예제 | • `eval('\'nice\'')` → 'nice'<br>• `eval(@expression_var)` → [@expression_var 평가 결과라면 무엇이든…] |

#### eval_template

문자열로 넘겨진 템플릿을 평가합니다. 컨텍스트 변수 또는 필드로 전달되는 동적 파라미터를 확장하는 데 유용합니다.

| | |
|---|---|
| 문법 | eval_template(template) |
| 인자 | • **template** - 템플릿 문자열 |
| 예제 | • `eval_template('QGIS [% upper(\'rocks\') %]')` → QGIS ROCKS |

### is_layer_visible

지정한 레이어가 현재 보이면 참을 반환합니다.

| | |
|---|---|
| 문법 | is_layer_visible(layer) |
| 인자 | • **layer** - 레이어 이름 또는 레이어 ID 가운데 하나를 나타내는 문자열 |
| 예제 | • `is_layer_visible('baseraster')` → TRUE |

### mime_type

바이너리 데이터의 MIME 유형을 반환합니다.

| | |
|---|---|
| 문법 | mime_type(bytes) |
| 인자 | • **bytes** - 바이너리 데이터 |
| 예제 | • `mime_type('<html><body></body></html>')` → text/html<br>• `mime_type(from_base64('R0lGODlhAQABAAAAACH5BAEKAAEALAAAAAABAAEAAAIAOw=='))` → image/gif |

### var

지정한 변수 내부에 저장된 값을 반환합니다.

| | |
|---|---|
| 문법 | var(name) |
| 인자 | • **name** - 변수 이름 |
| 예제 | • `var('qgis_version')` → '2.12' |

더 읽어볼 거리: 기본변수 목록

### with_variable

이 함수는 제 3 의 인수로 제공될 모든 표현식 코드를 위한 변수를 설정합니다. 동일한 계산 값을 서로 다른 위치에서 사용해야 하는 복잡한 표현식에서만 유용합니다.

| | |
|---|---|
| 문법 | with_variable(name, value, expression) |
| 인자 | • **name** - 설정할 변수의 이름<br>• **value** - 설정할 값<br>• **expression** - 변수를 사용할 수 있는 표현식 |
| 예제 | • `with_variable('my_sum', 1 + 2 + 3, @my_sum * 2 + @my_sum * 5)` → 42 |

### 9.2.14 도형 함수

이 그룹은 도형 객체를 대상으로 하는 (예: buffer, transform, $area 등) 함수를 담고 있습니다.

#### affine_transform

도형을 아핀 변환해서 반환합니다. 이 도형의 공간 좌표계에서 아핀 변환을 계산합니다. 축척, 기울기 (회전), 번역 (translation) 순서로 연산합니다. Z 또는 M 오프셋이 존재하지만 도형에는 존재하지 않는 경우 도형에 추가할 것입니다.

| | |
|---|---|
| 문법 | affine_transform(geometry, delta_x, delta_y, rotation_z, scale_x, scale_y, [delta_z=0], [delta_m=0], [scale_z=1], [scale_m=1])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **geometry** - 도형<br>• **delta_x** - X 축 번역<br>• **delta_y** - Y 축 번역<br>• **rotation_z** - Z 축을 중심으로 반시계 방향 도 단위 회전<br>• **scale_x** - X 축 축척 인자<br>• **scale_y** - Y 축 축척 인자<br>• **delta_z** - Z 축 번역<br>• **delta_m** - M 축 번역<br>• **scale_z** - Z 축 축척 인자<br>• **scale_m** - M 축 축척 인자 |
| 예제 | • `geom_to_wkt(affine_transform(geom_from_wkt('LINESTRING(1 1, 2 2)'), 2, 2, 0, 1, 1))` → 'LineString (3 3, 4 4)'<br>• `geom_to_wkt(affine_transform(geom_from_wkt('POLYGON((0 0, 0 3, 2 2, 0 0))'), 0, 0, -90, 1, 2))` → 'Polygon ((0 0, 6 0, 4 -2, 0 0))'<br>• `geom_to_wkt(affine_transform(geom_from_wkt('POINT(3 1)'), 0, 0, 0, 1, 1, 5, 0))` → 'PointZ (3 1 5)' |

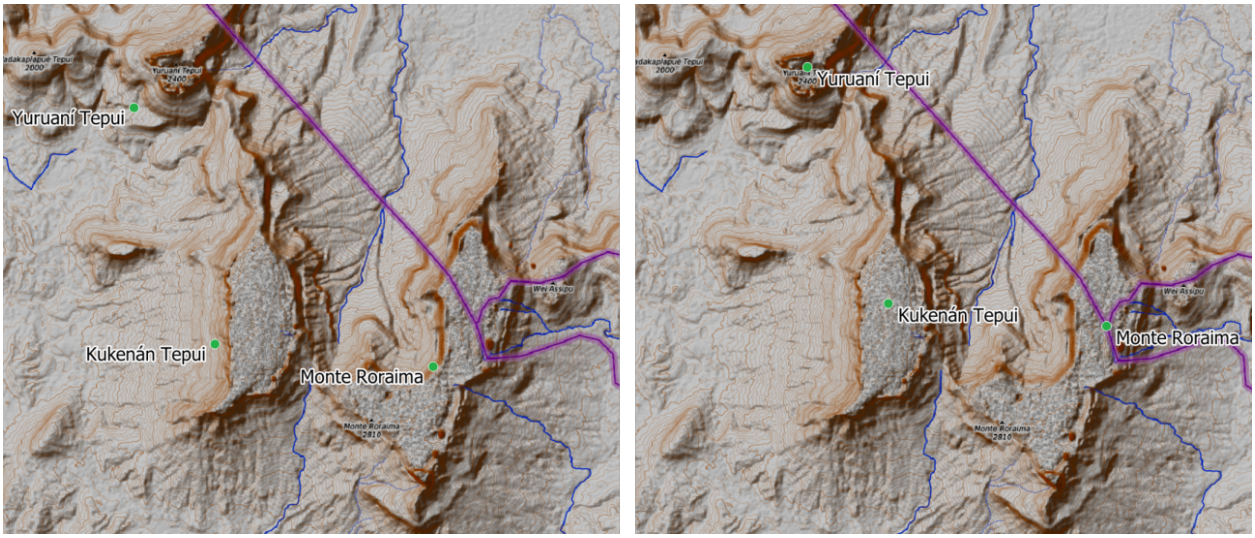



그림 9.4: 벡터 포인트 레이어 (녹색 점) 의 아핀 변형 (좌표 이동) 전 (왼쪽) 과 후 (오른쪽).

### angle_at_vertex

라인스트링 도형 상에 지정한 꼭짓점에서 도형에 대한 이등분선 각도 (평균 각도) 를 반환합니다. 여기서 각도는 진북에서 시계 방향으로 측정한 도 단위입니다.

| | |
|---|---|
| 문법 | angle_at_vertex(geometry, vertex) |
| 인자 | • **geometry** - 라인스트링 도형<br>• **vertex** - 0 부터 시작하는 꼭짓점 인덱스; 음의 값인 경우, 선택한 꼭짓점 인덱스는 꼭짓점 총 개수에서 절대값을 뺀 값이 될 것입니다 |
| 예제 | • `angle_at_vertex(geometry:=geom_from_wkt('LineString(0 0, 10 0, 10 10)'),vertex:=1)` → 45.0 |

### apply_dash_pattern

입력 도형에 점선 (dash) 패턴을 적용해서 각 라인/고리를 따라 지정한 패턴을 그린 멀티라인스트링 도형을 반환합니다.

| | |
|---|---|
| 문법 | apply_dash_pattern(geometry, pattern, [start_rule=no_rule], [end_rule=no_rule], [adjustment=both], [pattern_offset=0])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **geometry** - 도형 ([멀티] 라인스트링 또는 [멀티] 폴리곤 지정 가능)<br>• **pattern** - 점선 및 틈의 길이를 표현하는 숫자로 이루어진 배열로 정의된 점선 패턴. 배열이 담고 있는 요소의 개수가 짝수여야만 합니다.<br>• **start_rule** - 패턴의 시작을 제약하기 위한 선택적인 규칙. 무결한 값은'no_rule', 'full_dash', 'half_dash', 'full_gap', 'half_gap'입니다.<br>• **start_rule** - 패턴의 마지막을 제약하기 위한 선택적인 규칙. 무결한 값은'no_rule', 'full_dash', 'half_dash', 'full_gap', 'half_gap'입니다.<br>• **adjustment** - 원하는 패턴 규칙에 맞추기 위해 패턴의 어느 부분을 조정할지를 지정하기 위한 선택적인 규칙. 무결한 값은'both', 'dash', 'gap'입니다.<br>• **pattern_offset** - 패턴을 따라 시작할 특정 거리를 지정하는 선택적인 거리 |
| 예제 | • `geom_to_wkt(apply_dash_pattern(geom_from_wkt('LINESTRING(1 1, 10 1)'), array(3, 1)))` → MultiLineString ((1 1, 4 1),(5 1, 8 1),(9 1, 10 1, 10 1))<br>• `geom_to_wkt(apply_dash_pattern(geom_from_wkt('LINESTRING(1 1, 10 1)'), array(3, 1), start_rule:='half_dash'))` → MultiLineString ((1 1, 2.5 1),(3.5 1, 6.5 1),(7.5 1, 10 1, 10 1)) |

### $area

현재 객체의 면적을 반환합니다. 이 함수는 현재 프로젝트의 타원체 설정과 면적 단위 설정을 따라 면적을 계산합니다. 예를 들어 프로젝트에 타원체를 설정했다면 타원체 기반으로 면적을 계산하고, 타원체를 설정하지 않았다면 평면 상에서 면적을 계산합니다.

| | |
|---|---|
| 문법 | $area |
| 예제 | • `$area` → 42 |

### area

도형 폴리곤 객체의 면적을 계산합니다. 언제나 해당 도형의 공간 좌표계 (SRS) 안에서 평면 측량해서 계산하므로, 반환한 면적의 단위가 SRS 용 단위와 일치할 것입니다. 이것이 $area 함수가 수행하는 계산과 다른 점인데, $area 함수는 프로젝트의 타원체 및 면적 단위 설정을 기반으로 타원체 상에서 계산을 수행할 것입니다.

| | |
|---|---|
| 문법 | area(geometry) |
| 인자 | • **geometry** - 폴리곤 도형 객체 |
| 예제 | • `area(geom_from_wkt('POLYGON((0 0, 4 0, 4 2, 0 2, 0 0))'))` → 8.0 |

### azimuth

'point_a'의 수직선에서'point_b'로 시계 방향으로 측정한 진북 기준 방위각을 라디안 단위 각도로 반환합니다.

| | |
|---|---|
| 문법 | azimuth(point_a, point_b) |
| 인자 | • **point_a** - 포인트 도형<br>• **point_b** - 포인트 도형 |
| 예제 | • `degrees( azimuth( make_point(25, 45), make_point(75, 100) ) )` → 42.273689<br>• `degrees( azimuth( make_point(75, 100), make_point(25,45) ) )` → 222.273689 |

### bearing

'point_a'의 수직선에서'point_b'로 시계 방향으로 측정한 진북 기준 방위를 라디안 단위 각도로 반환합니다.

| | |
|---|---|
| 문법 | bearing(point_a, point_b, [source_crs], [ellipsoid])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **point_a** - 포인트 도형<br>• **point_b** - 포인트 도형<br>• **source_crs** - an optional string or CRS object representing the source CRS of the points. By default the current layer's CRS is used.<br>• **ellipsoid** - an optional string representing the acronym or the authority ID (e.g., 'EPSG:7030') of the ellipsoid on which the bearing should be measured. By default the current project's ellipsoid setting is used. |
| 예제 | • `degrees( bearing( make_point(16198544, -4534850), make_point(18736872, -1877769), 'EPSG:3857', 'EPSG:7030') )` → 49.980071<br>• `degrees( bearing( make_point(18736872, -1877769), make_point(16198544, -4534850), 'EPSG:3857', 'WGS84') )` → 219.282386 |

### boundary

입력 도형의 경계 (예: 도형의 위상 경계) 를 조합한 닫힌 경계를 반환합니다. 예를 들면, 폴리곤 도형은 내부에 있는 각 고리 별로 라인스트링으로 이루어진 경계를 가지게 될 것입니다. 포인트 또는 도형 집합 같은 일부 도형 유형들은 정의된 경계를 보유하고 있지 않기 때문에 NULL 을 반환할 것입니다.

| 문법 | boundary(geometry) |
| --- | --- |
| 인자 | • **geometry** - 도형 |
| 예제 | • `geom_to_wkt(boundary(geom_from_wkt('Polygon((1 1, 0 0, -1 1, 1 1))')))` → 'LineString(1 1,0 0,-1 1,1 1)'<br>• `geom_to_wkt(boundary(geom_from_wkt('LineString(1 1,0 0,-1 1)')))` → 'MultiPoint ((1 1),(-1 1))' |

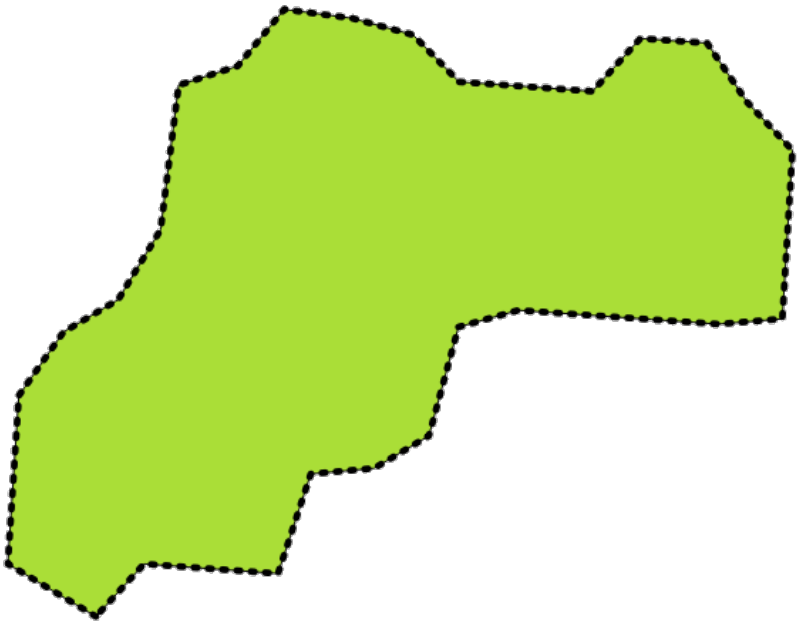


그림 9.5: 소스 폴리곤 레이어의 경계 (검은 점선)

더 읽어볼 거리: 경계 알고리즘

### bounds

입력 도형의 경계 상자 (bounding box) 를 표현하는 도형을 반환합니다. 해당 도형의 공간 좌표계 상에서 경계 상자를 계산합니다.

| 문법 | bounds(geometry) |
| --- | --- |
| 인자 | • **geometry** - 도형 |
| 예제 | • `bounds(@geometry)` → 현재 피처 도형의 경계 상자<br>• `geom_to_wkt(bounds(geom_from_wkt('Polygon((1 1, 0 0, -1 1, 1 1))')))` → 'Polygon ((-1 0, 1 0, 1 1, -1 1, -1 0))' |

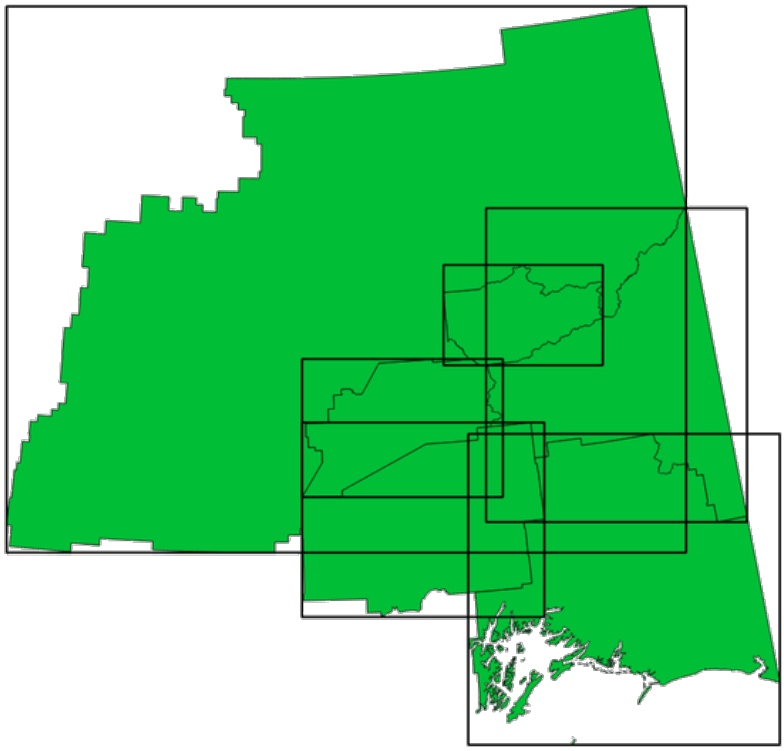


그림 9.6: 각 폴리곤 피처의 경계 상자를 표현하는 검은 선

더 읽어볼 거리: 경계 상자 알고리즘

### bounds_height

도형의 경계 상자 (bounding box) 의 높이를 표현하는 도형을 반환합니다. 해당 도형의 공간 좌표계 상에서 경계 상자를 계산합니다.

| | |
|---|---|
| 문법 | bounds_height(geometry) |
| 인자 | • **geometry** - 도형 |
| 예제 | • `bounds_height(@geometry)` → 현재 피처 도형의 경계 상자의 높이<br>• `bounds_height(geom_from_wkt('Polygon((1 1, 0 0, -1 1, 1 1))'))` → 1 |

### bounds_width

도형의 경계 상자 (bounding box) 의 너비를 표현하는 도형을 반환합니다. 해당 도형의 공간 좌표계 상에서 경계 상자를 계산합니다.

| | |
|---|---|
| 문법 | bounds_width(geometry) |
| 인자 | • **geometry** - 도형 |
| 예제 | • `bounds_width(@geometry)` → 현재 피처 도형의 경계 상자의 너비<br>• `bounds_width(geom_from_wkt('Polygon((1 1, 0 0, -1 1, 1 1))'))` → 2 |

### buffer

도형으로부터의 거리가 지정한 거리 이하인 모든 포인트들을 표현하는 도형을 반환합니다. 해당 도형의 공간 좌표계 상에서 거리를 계산합니다.

| | |
|---|---|
| 문법 | buffer(geometry, distance, [segments=8], [cap='round'], [join='round'], [miter_limit=2])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **geometry** - 도형<br>• **distance** - 레이어 단위의 버퍼 거리<br>• **segments** - 둥근 결합 스타일을 사용한 경우 사분원을 나타내는 데 사용할 선분의 개수입니다. 숫자가 클수록 더 많은 노드를 가진 매끈한 버퍼를 생성합니다.<br>• **cap** - 버퍼 용 마지막 선끝 스타일. 무결한 값은'round', 'flat'또는'square'입니다.<br>• **join** - 버퍼용 결합 스타일. 무결한 값은'round', 'bevel'또는'miter'입니다.<br>• **miter_limit** - 마이터 거리 제한. 결합 스타일을'miter'로 설정한 경우 사용할 수 있습니다. |
| 예제 | • `buffer(@geometry, 10.5)` → 현재 피처 도형에 버퍼를 10.5 단위만큼 적용한 폴리곤 |

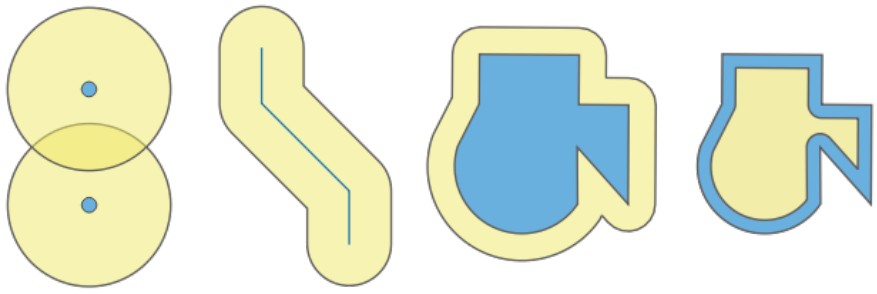


그림 9.7: 버퍼 (노란색) –양의 버퍼를 가진 포인트, 라인, 폴리곤 | 음의 버퍼를 가진 폴리곤

더 읽어볼 거리: 버퍼 알고리즘

### buffer_by_m

버퍼 반경이 라인 꼭짓점의 M 값에 따라 고르게 변하도록 라인 도형을 따라 버퍼를 생성합니다.

| | |
|---|---|
| 문법 | buffer_by_m(geometry, [segments=8])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **geometry** - 입력 도형. M 값을 가진 (멀티) 라인 도형이어야만 합니다.<br>• **segments** - 버퍼에서 사분면의 원호에 포함되는 선분 수 |
| 예제 | • `buffer_by_m(geometry:=geom_from_wkt('LINESTRINGM(1 2 0.5, 4 2 0.2)'),segments:=8)` → 반경이 0.5 에서 시작하여 라인스트링 도형을 따라 반경이 0.2 가 되는 가변 너비 버퍼 |

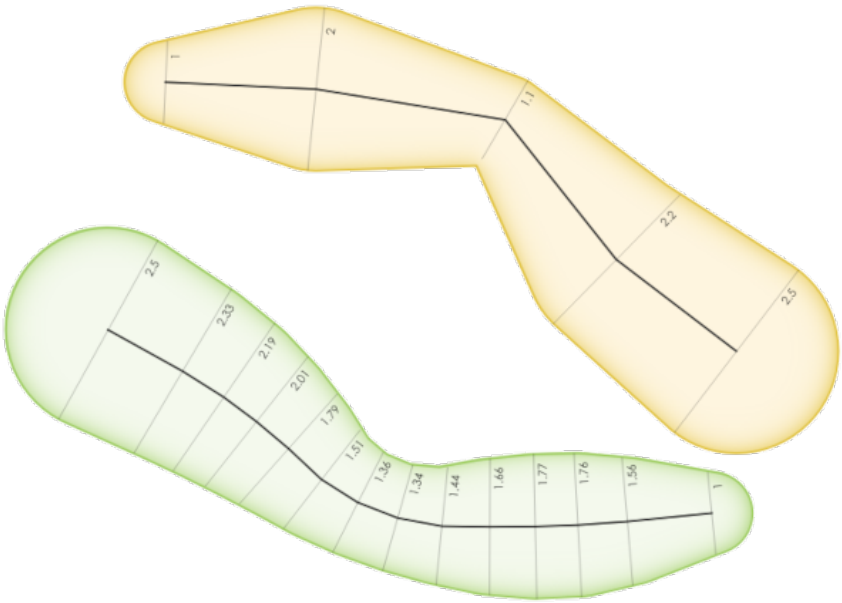



그림 9.8: 라인 피처의 꼭짓점에 M 값을 사용해서 버퍼 생성하기

더 읽어볼 거리: 변동 너비 버퍼 *(M 값으로)* 알고리즘

### centroid

도형의 기하학적 중심을 반환합니다.

| | |
|---|---|
| 문법 | centroid(geometry) |
| 인자 | • **geometry** - 도형 |
| 예제 | • `centroid(@geometry)` → 포인트 도형 |

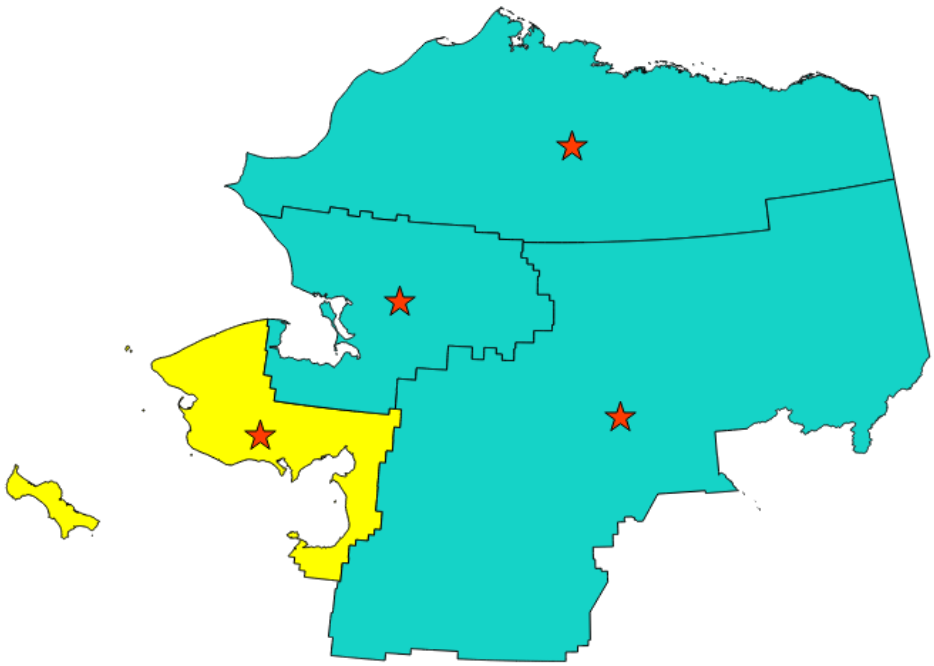


그림 9.9: 입력 레이어의 피처의 중심을 표현하는 빨간색 별

더 읽어볼 거리: 중심 알고리즘

### close_line

입력 라인스트링이 이미 닫힌 상태가 아닌 경우, 라인의 마지막 포인트에 첫 번째 포인트를 붙인 닫힌 라인스트링을 반환합니다. 도형이 라인스트링 또는 멀티라인스트링이 아닌 경우 NULL 을 반환할 것입니다.

| 문법 | close_line(geometry) |
|---|---|
| 인자 | • **geometry** - 라인스트링 도형 |
| 예제 | • `geom_to_wkt(close_line(geom_from_wkt('LINESTRING(0 0, 1 0, 1 1)')))` → 'LineString (0 0, 1 0, 1 1, 0 0)'<br>• `geom_to_wkt(close_line(geom_from_wkt('LINESTRING(0 0, 1 0, 1 1, 0 0)')))` → 'LineString (0 0, 1 0, 1 1, 0 0)' |

### closest_point

geometry2 와 가장 가까이 있는 geometry1 의 포인트를 반환합니다.

| | |
|---|---|
| 문법 | closest_point(geometry1, geometry2) |
| 인자 | • **geometry1** - 가장 가까운 점을 찾기 위한 기준 도형<br>• **geometry2** - 가장 가까운 점을 찾기 위한 대상 도형 |
| 예제 | • `geom_to_wkt(closest_point(geom_from_wkt('LINESTRING (20 80, 98 190, 110 180, 50 75 )'),geom_from_wkt('POINT(100 100)')))` → 'Point(73.0769 115.384)' |

### collect_geometries

도형 집합을 다중 부분 도형 객체로 수집합니다.

**인자 변이형 (variant) 목록**

도형 부분을 함수에 대한 개별 인자로 지정합니다.

| | |
|---|---|
| 문법 | collect_geometries(geometry1, geometry2, ⋯) |
| 인자 | • **geometry** - 도형 |
| 예제 | • `geom_to_wkt(collect_geometries(make_point(1,2), make_point(3,4), make_point(5,6)))` → 'MultiPoint ((1 2),(3 4),(5 6))' |

**배열 변이형**

도형 부분을 도형 부분의 배열로 지정합니다.

| | |
|---|---|
| 문법 | collect_geometries(array) |
| 인자 | • **array** - 도형 객체의 배열 |
| 예제 | • `geom_to_wkt(collect_geometries(array(make_point(1,2), make_point(3,4), make_point(5,6))))` → 'MultiPoint ((1 2),(3 4),(5 6))' |

더 읽어볼 거리: 도형 모으기 알고리즘

### combine

두 도형의 조합을 반환합니다.

| | |
|---|---|
| 문법 | combine(geometry1, geometry2) |
| 인자 | • **geometry1** - 도형<br>• **geometry2** - 도형 |
| 예제 | • `geom_to_wkt( combine( geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ), geom_from_wkt( 'LINESTRING(3 3, 4 4, 2 1)' ) ) )` → 'MULTILINESTRING((4 4, 2 1), (3 3, 4 4), (4 4, 5 5))'<br>• `geom_to_wkt( combine( geom_from_wkt( 'LINESTRING(3 3, 4 4)' ), geom_from_wkt( 'LINESTRING(3 3, 6 6, 2 1)' ) ) )` → 'LINESTRING(3 3, 4 4, 6 6, 2 1)' |

### concave_hull

도형에 있는 모든 포인트를 담고 있는 오목한 폴리곤을 반환합니다.

| | |
|---|---|
| 문법 | concave_hull(geometry, target_percent, [allow_holes=False])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **geometry** - 도형<br>• **target_percent** - 이 함수가 도달하려 하는 볼록 껍질 면적의 백분율. target_percent 가 1 이면 볼록 껍질과 동일한 결과를 산출합니다. target_percent 가 0 에서 0.99 사이라면 볼록 껍질보다 적은 면적을 산출합니다.<br>• **allow_holes** - optional argument specifying whether to allow holes within the output geometry. Defaults to FALSE, set to TRUE to allow including holes in the output geometry. |
| 예제 | • `geom_to_wkt(concave_hull(geom_from_wkt('MULTILINESTRING((106 164,30 112,74 70,82 112,130 94,130 62,122 40,156 32,162 76,172 88),(132 178,134 148,128 136,96 128,132 108,150 130,170 142,174 110,156 96,158 90,158 88),(22 64,66 28,94 38,94 68,114 76,112 30,132 10,168 18,178 34,186 52,184 74,190 100,190 122,182 148,178 170,176 184,156 164,146 178,132 186,92 182,56 158,36 150,62 150,76 128,88 118))'), 0.99))` → 'Polygon ((30 112, 36 150, 92 182, 132 186, 176 184, 190 122, 190 100, 186 52, 178 34, 168 18, 132 10, 112 30, 66 28, 22 64, 30 112))' |

더 읽어볼 거리: *convex_hull*, 오목 껍질 알고리즘

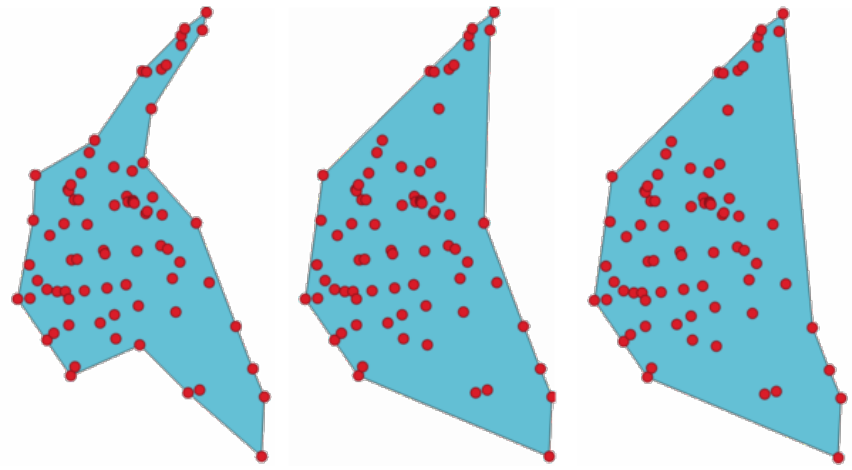


그림 9.10: target_percent 파라미터가 증가하는 오목 껍질

### contains

도형이 또다른 도형을 담고 있는지 검증합니다. geometry2 의 포인트 가운데 geometry1 외부에 있는 포인트가 하나도 없고, geometry2 내부의 포인트 가운데 최소한 포인트 1 개가 geometry1 내부에 있는 경우에만 참을 반환합니다.

| 문법 | contains(geometry1, geometry2) |
|---|---|
| 인자 | • **geometry1** - 도형<br>• **geometry2** - 도형 |
| 예제 | • `contains( geom_from_wkt( 'POLYGON((0 0, 0 1, 1 1, 1 0, 0 0))' ), geom_from_wkt( 'POINT(0.5 0.5 )' ) )` → TRUE<br>• `contains( geom_from_wkt( 'POLYGON((0 0, 0 1, 1 1, 1 0, 0 0))' ), geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ) )` → FALSE |

더 읽어볼 거리: *overlay_contains*

### convex_hull

도형의 볼록 껍질 (convex hull) 을 반환합니다. 볼록 껍질이란 선택 집합에 있는 모든 도형을 감싸는 최소 볼록 도형을 말합니다.

| 문법 | convex_hull(geometry) |
|---|---|
| 인자 | • **geometry** - 도형 |
| 예제 | • `geom_to_wkt( convex_hull( geom_from_wkt( 'LINESTRING(3 3, 4 4, 4 10)' ) ) )` → 'POLYGON((3 3, 4 10, 4 4, 3 3))' |

더 읽어볼 거리: *concave_hull*, 볼록 껍질 알고리즘

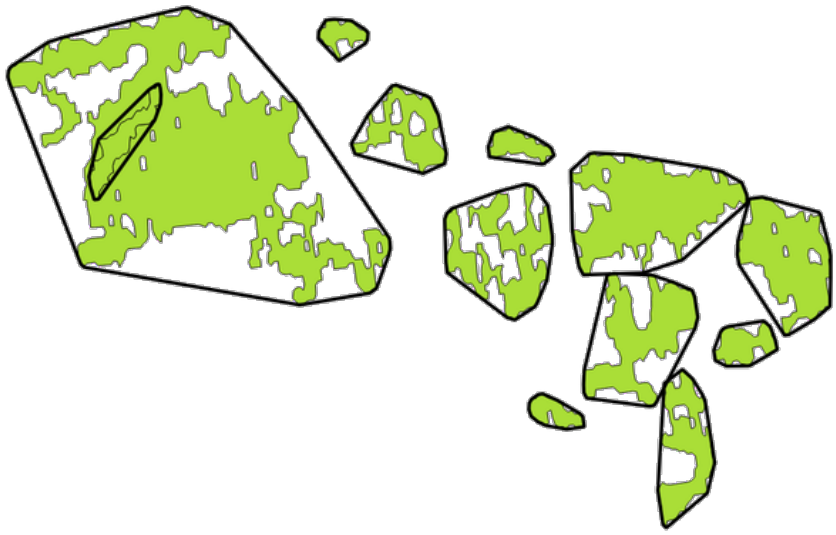


그림 9.11: 각 피처의 볼록 껍질을 표현하는 검은 선

### crosses

도형이 다른 도형과 공간교차 (cross) 하는지 검증합니다. 입력 도형들의 내부 포인트 전부가 아니라 일부가 일치할 경우 참을 반환합니다.

| | |
|---|---|
| 문법 | crosses(geometry1, geometry2) |
| 인자 | • **geometry1** - 도형<br>• **geometry2** - 도형 |
| 예제 | • `crosses( geom_from_wkt( 'LINESTRING(3 5, 4 4, 5 3)' ), geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ) )` → TRUE<br>• `crosses( geom_from_wkt( 'POINT(4 5)' ), geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ) )` → FALSE |

더 읽어볼 거리: *overlay_crosses*

### densify_by_count

폴리곤 또는 라인 레이어 도형을 입력받아 원본보다 더 많은 꼭짓점을 가지는 새 레이어 도형을 생성합니다.

| | |
|---|---|
| 문법 | densify_by_count(geometry, vertices) |
| 인자 | • **geometry** - 도형 ([멀티] 라인스트링 또는 [멀티] 폴리곤 지정 가능)<br>• **vertices** - (선분 당) 추가할 꼭짓점 개수 |
| 예제 | • `geom_to_wkt(densify_by_count(geom_from_wkt('LINESTRING(1 1, 10 1)'), 3))` → LineString (1 1, 3.25 1, 5.5 1, 7.75 1, 10 1) |

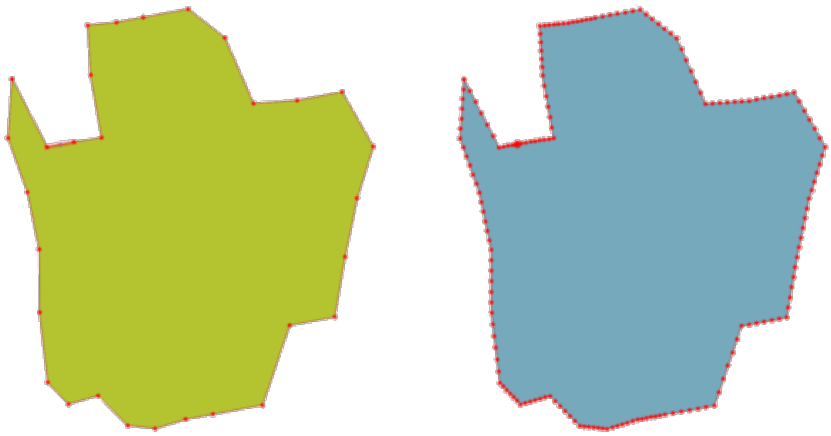


그림 9.12: 치밀화 이전과 이후의 꼭짓점을 표시하는 빨간 포인트

더 읽어볼 거리: 개수로 치밀화하기 알고리즘

### densify_by_distance

폴리곤 또는 라인 레이어 도형을 입력받아 가장자리 위에 지정한 최장 간격 거리 위치에 추가적인 꼭짓점들을 추가해서 치밀화시킨 새 레이어 도형을 생성합니다.

| | |
|---|---|
| 문법 | densify_by_distance(geometry, distance) |
| 인자 | • **geometry** - 도형 ([멀티] 라인스트링 또는 [멀티] 폴리곤 지정 가능)<br>• **distance** - 산출 도형에서 꼭짓점 사이의 최장 간격 거리 |
| 예제 | • `geom_to_wkt(densify_by_distance(geom_from_wkt('LINESTRING(1 1, 10 1)'), 4))` → LineString (1 1, 4 1, 7 1, 10 1) |

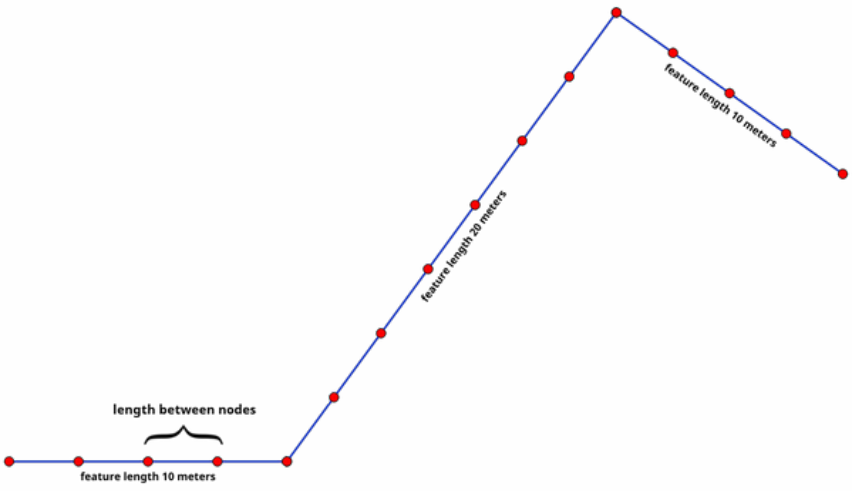



그림 9.13: 지정한 간격으로 도형 치밀화

더 읽어볼 거리: 간격으로 치밀화하기 알고리즘

### difference

geometry2 와 교차하지 않는 geometry1 의 부분을 표현하는 도형을 반환합니다.

| | |
|---|---|
| 문법 | difference(geometry1, geometry2) |
| 인자 | • **geometry1** - 도형<br>• **geometry2** - 도형 |
| 예제 | • `geom_to_wkt( difference( geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ), geom_from_wkt( 'LINESTRING(3 3, 4 4)' ) ) )` → 'LINESTRING(4 4, 5 5)' |

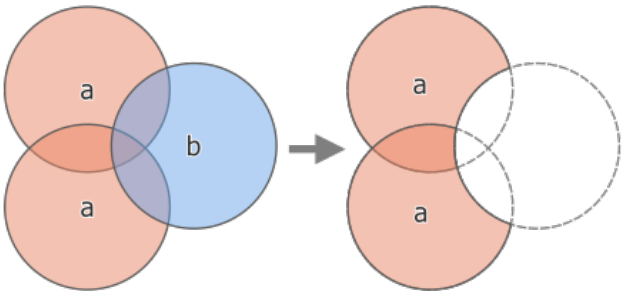



그림 9.14: Difference operation between a two-features input layer 'a' and a single feature overlay layer 'b' (left) - resulting in a new layer with the modified 'a' features (right)

더 읽어볼 거리: 차감하기 *(Difference)* 알고리즘

### disjoint

도형이 공간적으로 교차하지 않는지 여부를 테스트합니다. 도형들이 서로 어떠한 공간도 공유하지 않으면 참을 반환합니다.

| | |
|---|---|
| 문법 | disjoint(geometry1, geometry2) |
| 인자 | • **geometry1** - 도형<br>• **geometry2** - 도형 |
| 예제 | • `disjoint( geom_from_wkt( 'POLYGON((0 0, 0 1, 1 1, 1 0, 0 0 ))' ), geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ) )` → TRUE<br>• `disjoint( geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ), geom_from_wkt( 'POINT(4 4)' ))` → FALSE |

더 읽어볼 거리: *overlay_disjoint*

### distance

두 도형 사이의 (공간 좌표계 기반) 최단 거리를 투영체 단위로 반환합니다.

| | |
|---|---|
| 문법 | distance(geometry1, geometry2) |
| 인자 | • **geometry1** - 도형<br>• **geometry2** - 도형 |
| 예제 | • `distance( geom_from_wkt( 'POINT(4 4)' ), geom_from_wkt( 'POINT(4 8)' ) )` → 4 |

### distance_to_vertex

도형을 따라 지정한 꼭짓점까지의 거리를 반환합니다.

| | |
|---|---|
| 문법 | distance_to_vertex(geometry, vertex) |
| 인자 | • **geometry** - 라인스트링 도형<br>• **vertex** - 0 부터 시작하는 꼭짓점 인덱스; 음의 값인 경우, 선택한 꼭짓점 인덱스는 꼭짓점 총 개수에서 절대값을 뺀 값이 될 것입니다 |
| 예제 | • `distance_to_vertex(geometry:=geom_from_wkt('LineString(0 0, 10 0, 10 10)'),vertex:=1)` → 10.0 |

### end_point

도형에서 마지막 노드를 반환합니다.

| | |
|---|---|
| 문법 | end_point(geometry) |
| 인자 | • **geometry** - 도형 객체 |
| 예제 | • `geom_to_wkt(end_point(geom_from_wkt('LINESTRING(4 0, 4 2, 0 2)')))` → 'Point (0 2)' |

더 읽어볼 거리: *start_point*, 특정 꼭짓점 추출하기 알고리즘



그림 9.15: 라인 피처의 종단 포인트

### exif_geotag

이미지 파일의 Exif 지오태그로부터 포인트 도형을 생성합니다.

| | |
|---|---|
| 문법 | exif_geotag(path) |
| 인자 | • **path** - 이미지 파일 경로 또는 맵 레이어 값. 맵 레이어 값이 지정된 경우 레이어의 파일 소스를 사용할 것입니다. |
| 예제 | • `geom_to_wkt(exif_geotag('/my/photo.jpg'))` → 'Point (2 4)' |

### extend

라인스트링 도형의 시작과 끝을 지정한 양만큼 연장합니다. 라인에 있는 첫 번째와 마지막 선분의 방향을 사용해서 라인을 연장합니다. 멀티라인스트링의 경우, 모든 부분을 연장합니다. 거리 단위는 해당 도형의 공간 좌표계의 단위입니다.

| | |
|---|---|
| 문법 | extend(geometry, start_distance, end_distance) |
| 인자 | • **geometry** - (멀티) 라인스트링 도형<br>• **start_distance** - 라인 시작점을 연장할 거리<br>• **end_distance** - 라인 종단점을 연장할 거리 |
| 예제 | • `geom_to_wkt(extend(geom_from_wkt('LineString(0 0, 1 0, 1 1)'), 1,2))` → 'LineString (-1 0, 1 0, 1 3)'<br>• `geom_to_wkt(extend(geom_from_wkt('MultiLineString((0 0, 1 0, 1 1), (2 2, 0 2, 0 5))'),1,2))` → 'MultiLineString ((-1 0, 1 0, 1 3),(3 2, 0 2, 0 7))' |

더 읽어볼 거리: 라인 연장하기 알고리즘

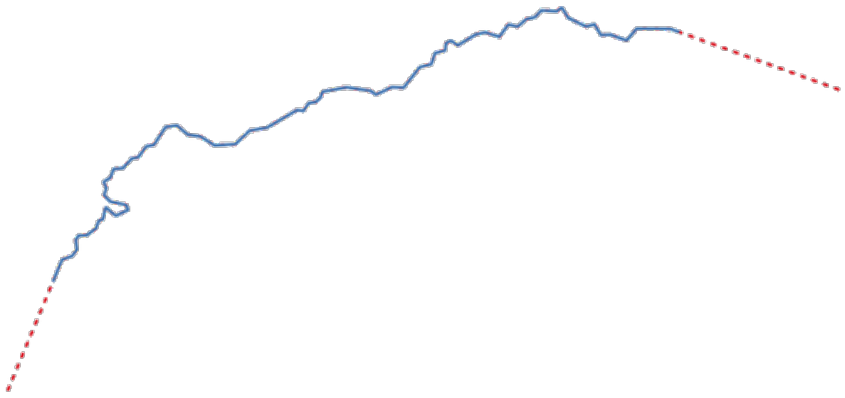


그림 9.16: 원본 레이어의 처음과 마지막 연장을 표현하는 빨간 점선

### exterior_ring

폴리곤 도형의 외곽 고리를 표현하는 라인스트링을 반환합니다. 도형이 폴리곤이 아닌 경우 NULL 을 반환할 것입니다.

| | |
|---|---|
| 문법 | exterior_ring(geometry) |
| 인자 | • **geometry** - 폴리곤 도형 |
| 예제 | • `geom_to_wkt(exterior_ring(geom_from_wkt('POLYGON((-1 -1, 4 0, 4 2, 0 2, -1 -1),( 0.1 0.1, 0.1 0.2, 0.2 0.2, 0.2, 0.1, 0.1 0.1))')))` → 'LineString (-1 -1, 4 0, 4 2, 0 2, -1 -1)' |

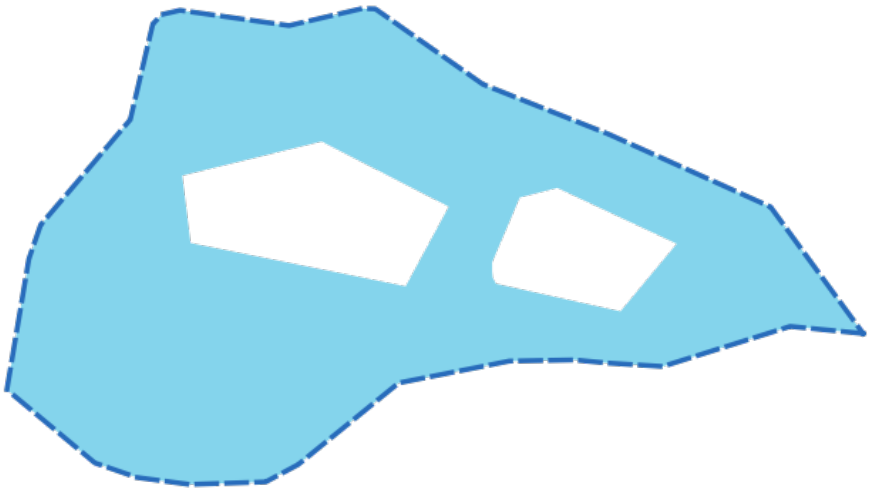


그림 9.17: 폴리곤의 외곽 고리를 표현하는 파선 (dashed line)

Further reading: 경계 algorithm, *interior_ring_n*

### extrude

입력 [멀티] 곡선 또는 [멀티] 라인스트링을 지정한 X 및 Y 좌표로 연장한 압출 (extruded) 버전을 반환합니다.

| 문법 | extrude(geometry, x, y) |
|---|---|
| 인자 | • **geometry** - 만곡 또는 라인스트링 도형<br>• **x** - x 확장, 숫자 값<br>• **y** - y 확장, 숫자 값 |
| 예제 | • `geom_to_wkt(extrude(geom_from_wkt('LineString(1 2, 3 2, 4 3)'), 1, 2))` → 'Polygon ((1 2, 3 2, 4 3, 5 5, 4 4, 2 4, 1 2))'<br>• `geom_to_wkt(extrude(geom_from_wkt('MultiLineString((1 2, 3 2), (4 3, 8 3))'), 1, 2))` → 'MultiPolygon (((1 2, 3 2, 4 4, 2 4, 1 2)),((4 3, 8 3, 9 5, 5 5, 4 3)))' |

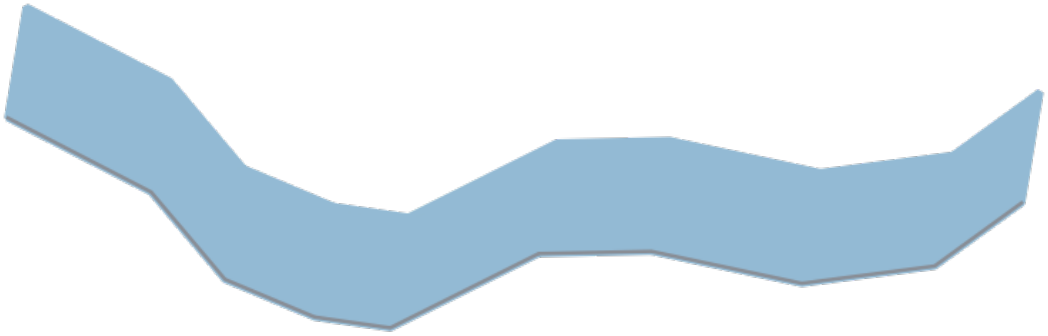


그림 9.18: 라인을 X 및 Y 방향 오프셋으로 압출해서 폴리곤 생성하기

### flip_coordinates

도형의 x 와 y 좌표값을 뒤바꾼 복사본을 반환합니다. 실수로 위도와 경도 값이 바뀐 도형을 복구하는데 유용합니다.

| 문법 | flip_coordinates(geometry) |
|---|---|
| 인자 | • **geometry** - 도형 |
| 예제 | • `geom_to_wkt(flip_coordinates(make_point(1, 2)))` → 'Point (2 1)'<br>• `geom_to_wkt(flip_coordinates(geom_from_wkt('LineString(0 2, 1 0, 1 6)')))` → 'LineString (2 0, 0 1, 6 1)' |

더 읽어볼 거리: *X* 좌표와 *Y* 좌표 바꾸기 알고리즘

### force_polygon_ccw

도형이 외곽 고리의 방향이 반시계 방향이고 내곽 고리의 방향은 시계 방향인 관습을 준수하도록 강제합니다.

| | |
|---|---|
| 문법 | force_polygon_ccw(geometry) |
| 인자 | • **geometry** - 도형. 폴리곤이 아닌 도형은 그대로 반환됩니다. |
| 예제 | • `geom_to_wkt(force_polygon_ccw(geometry:=geom_from_wkt('Polygon ((-1 -1, 0 2, 4 2, 4 0, -1 -1))')))` → 'Polygon ((-1 -1, 4 0, 4 2, 0 2, -1 -1))' |

더 읽어볼 거리: *force_polygon_cw*, *force_rhr*

### force_polygon_cw

도형이 외곽 고리의 방향이 시계 방향이고 내곽 고리의 방향은 반시계 방향인 관습을 준수하도록 강제합니다.

| | |
|---|---|
| 문법 | force_polygon_cw(geometry) |
| 인자 | • **geometry** - 도형. 폴리곤이 아닌 도형은 그대로 반환됩니다. |
| 예제 | • `geom_to_wkt(force_polygon_cw(geometry:=geom_from_wkt('POLYGON((-1 -1, 4 0, 4 2, 0 2, -1 -1))')))` → 'Polygon ((-1 -1, 0 2, 4 2, 4 0, -1 -1))' |

더 읽어볼 거리: *force_polygon_ccw*, *force_rhr*

### force_rhr

도형이 오른손 규칙을 따르도록 강제합니다. 오른손 규칙이란 폴리곤 경계 안의 영역이 경계의 오른쪽에 있다는 의미입니다. 자세히 말하자면, 외곽 고리는 시계 방향이며 내곽 고리는 반시계 방향입니다. 오른손 규칙의 정의가 몇몇 맥락에서 일관성이 없는 경우가 있기 때문에, 되도록이면 force_polygon_cw 함수만 사용하는 편이 좋습니다.

| | |
|---|---|
| 문법 | force_rhr(geometry) |
| 인자 | • **geometry** - 도형. 폴리곤이 아닌 도형은 그대로 반환됩니다. |
| 예제 | • `geom_to_wkt(force_rhr(geometry:=geom_from_wkt('POLYGON((-1 -1, 4 0, 4 2, 0 2, -1 -1))')))` → 'Polygon ((-1 -1, 0 2, 4 2, 4 0, -1 -1))' |

더 읽어볼 거리: 오른손 법칙 강제하기 알고리즘, *force_polygon_ccw*, *force_polygon_cw*

### geom_from_gml

도형의 GML 표현으로부터 생성된 도형을 반환합니다.

| | |
|---|---|
| 문법 | geom_from_gml(gml) |
| 인자 | • **gml** - 도형의 GML 표현 (문자열) |
| 예제 | • `geom_from_gml('<gml:LineString srsName="EPSG:4326"><gml:coordinates>4, 4 5,5 6,6</gml:coordinates></gml:LineString>')` → 라인 도형 객체 |

### geom_from_wkb

WKB(Well-Known Binary) 표현으로부터 생성된 도형을 반환합니다.

| | |
|---|---|
| 문법 | geom_from_wkb(binary) |
| 인자 | • **binary** - 도형의 WKB(Well-Known Binary) 표현 (바이너리 BLOB(Binary Large Object)) |
| 예제 | • `geom_from_wkb( geom_to_wkb( make_point(4,5) ) )` → 포인트 도형 객체 |

### geom_from_wkt

WKT(well-known text) 표현으로부터 생성된 도형을 반환합니다.

| | |
|---|---|
| 문법 | geom_from_wkt(text) |
| 인자 | • **text** - 도형의 WKT(well-known text) 표현 |
| 예제 | • `geom_from_wkt( 'POINT(4 5)' )` → 도형 객체 |

### geom_to_wkb

도형의 WKB(Well-Known Binary) 표현을 반환합니다

| | |
|---|---|
| 문법 | geom_to_wkb(geometry) |
| 인자 | • **geometry** - 도형 |
| 예제 | • `geom_to_wkb( @geometry )` → 도형 객체를 담고 있는 바이너리 BLOB(Binary Large Object) |

### geom_to_wkt

SRID 메타데이터를 제외한 도형의 WKT(well-known text) 표현을 반환합니다.

| | |
|---|---|
| 문법 | geom_to_wkt(geometry, [precision=8])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **geometry** - 도형<br>• **precision** - 숫자 정밀도 |
| 예제 | • `geom_to_wkt( make_point(6, 50) )` → 'POINT(6 50)'<br>• `geom_to_wkt(centroid(geom_from_wkt('Polygon((1 1, 0 0, -1 1, 1 1))')))` → 'POINT(0 0.66666667)'<br>• `geom_to_wkt(centroid(geom_from_wkt('Polygon((1 1, 0 0, -1 1, 1 1))')), 2)` → 'POINT(0 0.67)' |

### $geometry

현재 피처의 도형을 반환합니다. 다른 함수들의 처리 과정에 사용할 수 있습니다. **경고: 이 함수는 중요도가 떨어져 더 이상 사용되지 않고 앞으로는 사라지게 될 것입니다. 이 함수를 대체하는 @geometry 변수를 사용할 것을 권장합니다.**

| | |
|---|---|
| 문법 | $geometry |
| 예제 | • `geom_to_wkt( $geometry )` → 'POINT(6 50)' |

### geometry

객체의 도형을 반환합니다.

| | |
|---|---|
| 문법 | geometry(feature) |
| 인자 | • **feature** - 피처 객체 |
| 예제 | • `geometry( @feature )` → 현재 피처의 도형. @geometry 를 사용하는 편이 좋습니다.<br>• `geom_to_wkt( geometry( get_feature_by_id( 'streets', 1 ) ) )` → "streets"레이어의 ID 1 인 피처의 WKT 에 있는 도형. 예: 'POINT(6 50)'<br>• `intersects( @geometry, geometry( get_feature( 'streets', 'name', 'Main St.' ) ) )` → 현재 피처가"streets"레이어의'Main St.'라는 이름의 피처와 공간적으로 교차하는 경우 TRUE |

### geometry_n

도형 집합에서 특정 도형을 반환하거나, 입력 도형이 집합이 아닌 경우 NULL 을 반환합니다. 다중 부분 도형에서도 한 부분을 반환합니다.

| | |
|---|---|
| 문법 | geometry_n(geometry, index) |
| 인자 | • **geometry** - 도형 집합<br>• **index** - 집합의 첫 번째 도형이 1 인, 반환할 도형 인덱스 |
| 예제 | • `geom_to_wkt(geometry_n(geom_from_wkt('GEOMETRYCOLLECTION(POINT(0 1), POINT(0 0), POINT(1 0), POINT(1 1))'),3))` → 'Point (1 0)' |

### geometry_type

도형의 유형을 설명하는 문자열 (Point, Line, 또는 Polygon) 값을 반환합니다.

| | |
|---|---|
| 문법 | geometry_type(geometry) |
| 인자 | • **geometry** - 도형 |
| 예제 | • `geometry_type( geom_from_wkt( 'LINESTRING(2 5, 3 6, 4 8)') )` → 'Line'<br>• `geometry_type( geom_from_wkt( 'MULTILINESTRING((2 5, 3 6, 4 8), (1 1, 0 0))') )` → 'Line'<br>• `geometry_type( geom_from_wkt( 'POINT(2 5)') )` → 'Point'<br>• `geometry_type( geom_from_wkt( 'POLYGON((-1 -1, 4 0, 4 2, 0 2, -1 -1))') )` → 'Polygon' |

### hausdorff_distance

두 도형 사이의 하우스도르프 거리 (Hausdorff distance) 를 반환합니다. 하우스도르프 거리란 기본적으로 두 도형이 얼마나 닮았는지 또는 닮지 않았는지를 나타내는 척도입니다. 거리가 짧을수록 도형들이 더 닮았다는 의미입니다.

이 함수는 선택적인 치밀화 분수 (densify fraction) 인자와 함께 실행할 수 있습니다. 이 인자를 지정하지 않으면, 표준 하우스도르프 거리의 근사치를 사용합니다. 이 근사치는 활용도가 높은 사례들의 대다수에 대해 정확하거나 충분히 가까운 값입니다. 이에 대한 예는 다음과 같습니다:

- 서로에 대해 대충 평행하며 대충 길이도 같은 라인스트링들 사이의 거리를 계산합니다. 이 사례는 일치하는 선형 네트워크들에서 찾아볼 수 있습니다.
- 도형의 유사성을 검증합니다.

이 방법으로 제공되는 기본 근사값이 충분하지 않은 경우, 선택적인 치밀화 분수 (densify fraction) 인자를 지정하십시오. 이 인자를 지정하면 개별 하우스도르프 거리를 계산하기 전에 선분 밀도를 늘리는 작업을 수행합니다. 이 파라미터는 각 선분을 치밀화하는 분수를 설정합니다. 각 선분은 동일한 길이의 하위 선분으로 분할되는데, 이때 전체 길이의 분수가 지정한 분수에 가장 가까워집니다. 치밀화 분수 파라미터의 값을 줄이면 반환된 거리가 도형에 대한 실제 하우스도르프 거리에 근접하게 됩니다.

| | |
|---|---|
| 문법 | hausdorff_distance(geometry1, geometry2, [densify_fraction])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **geometry1** - 도형<br>• **geometry2** - 도형<br>• **densify_fraction** - 치밀화 분수의 양 |
| 예제 | • `hausdorff_distance( geometry1:= geom_from_wkt('LINESTRING (0 0, 2 1)'),geometry2:=geom_from_wkt('LINESTRING (0 0, 2 0)'))` → 2<br>• `hausdorff_distance( geom_from_wkt('LINESTRING (130 0, 0 0, 0 150)'),geom_from_wkt('LINESTRING (10 10, 10 150, 130 10)'))` → 14.142135623<br>• `hausdorff_distance( geom_from_wkt('LINESTRING (130 0, 0 0, 0 150)'),geom_from_wkt('LINESTRING (10 10, 10 150, 130 10)'),0.5)` → 70.0 |

### inclination

'point_a'에서'point_b'로 가는 경사를 천정 (zenith, 0) 부터 천저 (nadir, 180) 범위에서 측정해서 반환합니다.

| | |
|---|---|
| 문법 | inclination(point_a, point_b) |
| 인자 | • **point_a** - 포인트 도형<br>• **point_b** - 포인트 도형 |
| 예제 | • `inclination( make_point( 5, 10, 0 ), make_point( 5, 10, 5 ) )` → 0.0<br>• `inclination( make_point( 5, 10, 0 ), make_point( 5, 10, 0 ) )` → 90.0<br>• `inclination( make_point( 5, 10, 0 ), make_point( 50, 100, 0 ) )` → 90.0<br>• `inclination( make_point( 5, 10, 0 ), make_point( 5, 10, -5 ) )` → 180.0 |

### interior_ring_n

폴리곤 도형에서 특정 내곽 고리를 반환하거나, 도형이 폴리곤이 아닌 경우 NULL 을 반환합니다.

| | |
|---|---|
| 문법 | interior_ring_n(geometry, index) |
| 인자 | • **geometry** - 폴리곤 도형<br>• **index** - 첫 번째 내곽 고리가 1 인, 반환할 내곽 인덱스 |
| 예제 | • `geom_to_wkt(interior_ring_n(geom_from_wkt('POLYGON((-1 -1, 4 0, 4 2, 0 2, -1 -1),(-0.1 -0.1, 0.4 0, 0.4 0.2, 0 0.2, -0.1 -0.1),(-1 -1, 4 0, 4 2, 0 2, -1 -1))'),1))` → 'LineString (-0.1 -0.1, 0.4 0, 0.4 0.2, 0 0.2, -0.1 -0.1)' |

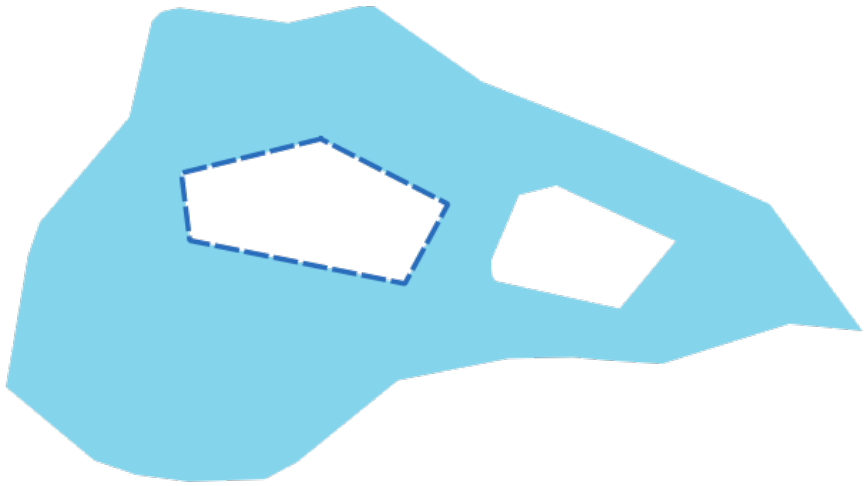


그림 9.19: The dashed line represents a specific interior ring of the polygon

Further reading: *exterior_ring*

### intersection

두 도형이 공유하는 부분을 표현하는 도형을 반환합니다.

| | |
|---|---|
| 문법 | intersection(geometry1, geometry2) |
| 인자 | • **geometry1** - 도형<br>• **geometry2** - 도형 |
| 예제 | • `geom_to_wkt( intersection( geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ), geom_from_wkt( 'LINESTRING(3 3, 4 4)' ) ) )` → 'LINESTRING(3 3, 4 4)'<br>• `geom_to_wkt( intersection( geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ), geom_from_wkt( 'MULTIPOINT(3.5 3.5, 4 5)' ) ) )` → 'POINT(3.5 3.5)' |

더 읽어볼 거리: 교차 *(Intersection)* 알고리즘

### intersects

도형들이 서로 교차하는지 검증합니다. 도형들이 공간적으로 교차하는 경우 (공간의 어떤 부분이라도 공유하는 경우) 참을 반환하고 교차하지 않는 경우 거짓을 반환합니다.

| | |
|---|---|
| 문법 | intersects(geometry1, geometry2) |
| 인자 | • **geometry1** - 도형<br>• **geometry2** - 도형 |
| 예제 | • `intersects( geom_from_wkt( 'POINT(4 4)' ), geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ) )` → TRUE<br>• `intersects( geom_from_wkt( 'POINT(4 5)' ), geom_from_wkt( 'POINT(5 5)' ) )` → FALSE |

더 읽어볼 거리: *overlay_intersects*

### intersects_bbox

도형의 경계 상자가 다른 도형의 경계 상자와 교차하는지 검증합니다. 도형들의 경계 상자가 공간적으로 교차하는 경우 (공간의 어떤 부분이라도 공유하는 경우) 참을 반환하고 교차하지 않는 경우 거짓을 반환합니다.

| 문법 | intersects_bbox(geometry1, geometry2) |
| --- | --- |
| 인자 | • **geometry1** - 도형<br>• **geometry2** - 도형 |
| 예제 | • `intersects_bbox( geom_from_wkt( 'POINT(4 5)' ), geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ) )` → TRUE<br>• `intersects_bbox( geom_from_wkt( 'POINT(6 5)' ), geom_from_wkt( 'POLYGON((3 3, 4 4, 5 5, 3 3))' ) )` → FALSE |

### is_closed

라인스트링이 닫힌 (시작점과 종단점이 일치하는) 경우 참을, 라인스트링이 닫히지 않은 경우 거짓을, 도형이 라인스트링이 아닌 경우 NULL 을 반환합니다.

| 문법 | is_closed(geometry) |
| --- | --- |
| 인자 | • **geometry** - 라인스트링 도형 |
| 예제 | • `is_closed(geom_from_wkt('LINESTRING(0 0, 1 1, 2 2)'))` → FALSE<br>• `is_closed(geom_from_wkt('LINESTRING(0 0, 1 1, 2 2, 0 0)'))` → TRUE |

### is_empty

도형이 비어 있는 (좌표가 없는) 경우 참을, 도형이 비어 있지 않은 경우 거짓을, 도형이 없는 경우 NULL 을 반환합니다. is_empty_or_null 함수도 참조하세요.

| 문법 | is_empty(geometry) |
| --- | --- |
| 인자 | • **geometry** - 도형 |
| 예제 | • `is_empty(geom_from_wkt('LINESTRING(0 0, 1 1, 2 2)'))` → FALSE<br>• `is_empty(geom_from_wkt('LINESTRING EMPTY'))` → TRUE<br>• `is_empty(geom_from_wkt('POINT(7 4)'))` → FALSE<br>• `is_empty(geom_from_wkt('POINT EMPTY'))` → TRUE |

더 읽어볼 거리: *is_empty_or_null*

### is_empty_or_null

도형이 NULL 이거나 비어 있는 (좌표가 없는) 경우 참을 반환하고, 그렇지 않은 경우 거짓을 반환합니다. 이 함수는 '@geometry IS NULL 또는 is_empty(@geometry)'표현식과 비슷합니다.

| | |
|---|---|
| 문법 | is_empty_or_null(geometry) |
| 인자 | • **geometry** - 도형 |
| 예제 | • `is_empty_or_null(NULL)` → TRUE<br>• `is_empty_or_null(geom_from_wkt('LINESTRING(0 0, 1 1, 2 2)'))` → FALSE<br>• `is_empty_or_null(geom_from_wkt('LINESTRING EMPTY'))` → TRUE<br>• `is_empty_or_null(geom_from_wkt('POINT(7 4)'))` → FALSE<br>• `is_empty_or_null(geom_from_wkt('POINT EMPTY'))` → TRUE |

더 읽어볼 거리: *is_empty*, *NULL*

### is_multipart

도형이 멀티 유형인 경우 참을 반환합니다.

| | |
|---|---|
| 문법 | is_multipart(geometry) |
| 인자 | • **geometry** - 도형 |
| 예제 | • `is_multipart(geom_from_wkt('MULTIPOINT ((0 0),(1 1),(2 2))'))` → TRUE<br>• `is_multipart(geom_from_wkt('POINT (0 0)'))` → FALSE |

### is_valid

도형이 무결한 경우, 즉 도형이 2 차원에서 OGC 규칙에 따라 잘 형성된 경우 참을 반환합니다.

| | |
|---|---|
| 문법 | is_valid(geometry) |
| 인자 | • **geometry** - 도형 |
| 예제 | • `is_valid(geom_from_wkt('LINESTRING(0 0, 1 1, 2 2, 0 0)'))` → TRUE<br>• `is_valid(geom_from_wkt('LINESTRING(0 0)'))` → FALSE |

더 읽어볼 거리: *make_valid*, 무결성 점검하기 알고리즘

### $length

라인스트링의 길이를 반환합니다. 만약 폴리곤의 둘레 길이가 필요하다면, $perimeter 를 대신 사용하십시오. 이 함수는 현재 프로젝트의 타원체 설정과 거리 단위 설정을 따라 거리를 계산합니다. 예를 들어 프로젝트에 타원체를 설정했다면 타원체 기반으로 거리를 계산하고, 타원체를 설정하지 않았다면 평면 상에서 거리를 계산합니다.

| | |
|---|---|
| 문법 | $length |
| 예제 | • `$length` → 42.4711 |

### length

문자열의 문자 개수 또는 라인스트링 도형의 길이를 반환합니다.

#### 문자열 변이형

문자열에 있는 문자의 개수를 반환합니다.

| | |
|---|---|
| 문법 | length(string) |
| 인자 | • **string** - 길이를 셀 문자열 |
| 예제 | • `length('hello')` → 5 |

#### 도형 변이형

라인 도형 객체의 길이를 계산합니다. 언제나 해당 도형의 공간 좌표계 (SRS) 에서 평면 측량해서 계산하므로, 반환한 길이의 단위가 SRS 용 단위와 일치할 것입니다. 이것이 $length 함수가 수행하는 계산과 다른 점인데, $length 함수는 프로젝트의 타원체 및 거리 단위 설정을 기반으로 타원체 상에서 계산을 수행할 것입니다.

| | |
|---|---|
| 문법 | length(geometry) |
| 인자 | • **geometry** - 라인 도형 객체 |
| 예제 | • `length(geom_from_wkt('LINESTRING(0 0, 4 0)'))` → 4.0 |

더 읽어볼 거리: *straight_distance_2d*

### length3D

라인 도형 객체의 3 차원 길이를 계산합니다. 도형이 3 차원 라인 객체가 아닌 경우 2 차원 길이를 반환합니다. 언제나 해당 도형의 공간 좌표계 (SRS) 에서 평면 측량해서 계산하므로, 반환한 길이의 단위가 SRS 용 단위와 일치할 것입니다. 이것이 $length 함수가 수행하는 계산과 다른 점인데, $length 함수는 프로젝트의 타원체 및 거리 단위 설정을 기반으로 타원체 상에서 계산을 수행할 것입니다.

| 문법 | length3D(geometry) |
| --- | --- |
| 인자 | • **geometry** - 라인 도형 객체 |
| 예제 | • `length3D(geom_from_wkt('LINESTRINGZ(0 0 0, 3 0 4'))` → 5.0 |

### line_interpolate_angle

라인스트링 도형을 따라 지정한 거리에서 도형과 평행한 각도를 반환합니다. 여기서 각도는 진북에서 시계 방향으로 측정한 도 단위입니다.

| 문법 | line_interpolate_angle(geometry, distance) |
| --- | --- |
| 인자 | • **geometry** - 라인스트링 도형<br>• **distance** - 라인을 따라 각도를 보간할 거리 |
| 예제 | • `line_interpolate_angle(geometry:=geom_from_wkt('LineString(0 0, 10 0)'),distance:=5)` → 90.0 |

### line_interpolate_point

라인스트링 도형을 따라 지정한 거리로 보간한 포인트를 반환합니다.

| 문법 | line_interpolate_point(geometry, distance) |
| --- | --- |
| 인자 | • **geometry** - 라인스트링 도형<br>• **distance** - 라인을 따라 보간할 거리 |
| 예제 | • `geom_to_wkt(line_interpolate_point(geometry:=geom_from_wkt('LineString(0 0, 8 0)'), distance:=5))` → 'Point (5 0)'<br>• `geom_to_wkt(line_interpolate_point(geometry:=geom_from_wkt('LineString(0 0, 1 1, 2 0)'), distance:=2.1))` → 'Point (1.48492424 0.51507576)'<br>• `geom_to_wkt(line_interpolate_point(geometry:=geom_from_wkt('LineString(0 0, 1 0)'), distance:=2))` → NULL |

더 읽어볼 거리: 라인에 포인트를 보간하기 알고리즘

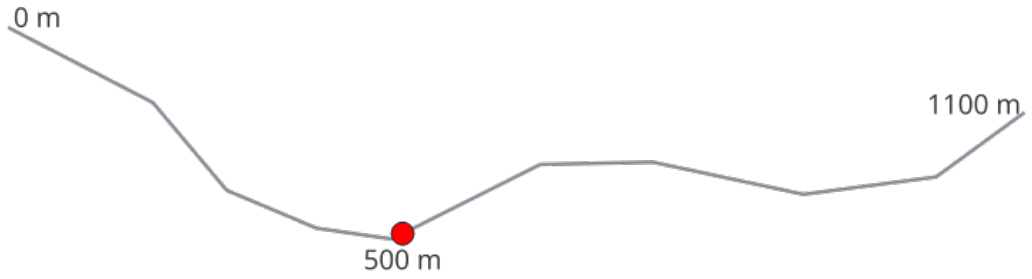



그림 9.20: 라인의 시작에서 500m 거리에 보간된 포인트

### line_interpolate_point_by_m

Returns the point interpolated by a matching M value along a linestring geometry.

| | |
|---|---|
| 문법 | line_interpolate_point_by_m(geometry, m, [use_3d_distance=false])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **geometry** - 라인스트링 도형<br>• **m** - an M value<br>• **use_3d_distance** - controls whether 2D or 3D distances between vertices should be used during interpolation (this option is only considered for lines with z values) |
| 예제 | • `geom_to_wkt(line_interpolate_point_by_m(geom_from_wkt('LineStringM(0 0 0, 10 10 10)'), m:=5))` → 'Point (5 5)' |

### line_locate_m

Returns the distance along a linestring corresponding to the first matching interpolated M value.

| | |
|---|---|
| 문법 | line_locate_m(geometry, m, [use_3d_distance=false])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **geometry** - 라인스트링 도형<br>• **m** - an M value<br>• **use_3d_distance** - controls whether 2D or 3D distances between vertices should be used during interpolation (this option is only considered for lines with z values) |
| 예제 | • `line_locate_m(geometry:=geom_from_wkt('LineStringM(0 0 0, 10 10 10)'),m:=5)` → 7.07106 |

### line_locate_point

라인스트링을 따라 라인스트링이 지정한 포인트 도형에 가장 가까워지는 위치에 상응하는 거리를 반환합니다.

| | |
|---|---|
| 문법 | line_locate_point(geometry, point) |
| 인자 | • **geometry** - 라인스트링 도형<br>• **point** - 라인스트링 상에서 가장 가까운 위치를 찾아야 할 포인트 도형 |
| 예제 | • `line_locate_point(geometry:=geom_from_wkt('LineString(0 0, 10 0)'),point:=geom_from_wkt('Point(5 0)'))` → 5.0 |

### line_merge

입력 도형에서 연결된 모든 라인스트링을 단일 라인스트링으로 병합한 라인스트링 또는 멀티라인스트링 도형을 반환합니다. 이 함수는 입력 도형이 라인스트링/멀티라인스트링이 아닌 경우 NULL 을 반환할 것입니다.

| | |
|---|---|
| 문법 | line_merge(geometry) |
| 인자 | • **geometry** - 라인스트링/멀티라인스트링 도형 |
| 예제 | • `geom_to_wkt(line_merge(geom_from_wkt('MULTILINESTRING((0 0, 1 1),(1 1, 2 2))')))` → 'LineString(0 0,1 1,2 2)'<br>• `geom_to_wkt(line_merge(geom_from_wkt('MULTILINESTRING((0 0, 1 1),(11 1, 21 2))')))` → 'MultiLineString((0 0, 1 1),(11 1, 21 2)' |

### line_substring

지정한 시작점과 종단점 거리 사이에 떨어지는 라인 (또는 곡선) 도형의 부분을 반환합니다 (라인의 시작에서부터 측정). Z 및 M 값은 기존 값에서 선형 보간됩니다.

| | |
|---|---|
| 문법 | line_substring(geometry, start_distance, end_distance) |
| 인자 | • **geometry** - 라인스트링 또는 곡선 도형<br>• **start_distance** - 하위 스트링의 시작점까지의 거리<br>• **end_distance** - 하위 스트링의 종단점까지의 거리 |
| 예제 | • `geom_to_wkt(line_substring(geometry:=geom_from_wkt('LineString(0 0, 10 0)'),start_distance:=2,end_distance:=6))` → 'LineString (2 0,6 0)' |

더 읽어볼 거리: 라인 부스트링 생성하기 알고리즘




그림 9.21: 시작 거리를 0m 로, 종단 거리를 250m 로 설정한 부스트링 라인

### m

포인트 도형의 (측정) M 값을 반환합니다.

| 문법 | m(geometry) |
|---|---|
| 인자 | • **geometry** - 포인트 도형 |
| 예제 | • `m( geom_from_wkt( 'POINTM(2 5 4)' ) )` → 4 |

### m_at

도형의 M 좌표를 반환하거나, 도형에 M 값이 없는 경우 NULL 을 반환합니다.

| 문법 | m_at(geometry, vertex) |
|---|---|
| 인자 | • **geometry** - 도형 객체<br>• **vertex** - 도형의 꼭짓점의 인덱스 (0 부터 인덱스 시작; 음의 값은 -1 부터 시작하며 마지막 인덱스부터 적용) |
| 예제 | • `m_at(geom_from_wkt('LineStringZM(0 0 0 0, 10 10 0 5, 10 10 0 0)'), 1)` → 5 |

### m_max

도형의 최대 (측정) M 값을 반환합니다.

| 문법 | m_max(geometry) |
|---|---|
| 인자 | • **geometry** - M 값을 담고 있는 도형 |
| 예제 | • `m_max( make_point_m( 0,0,1 ) )` → 1<br>• `m_max(make_line( make_point_m( 0,0,1 ), make_point_m( -1,-1,2 ), make_point_m( -2,-2,0 ) ) )` → 2 |

### m_min

도형의 최소 (측정) M 값을 반환합니다.

| | |
|---|---|
| 문법 | m_min(geometry) |
| 인자 | • **geometry** - M 값을 담고 있는 도형 |
| 예제 | • `m_min( make_point_m( 0,0,1 ) )` → 1<br>• `m_min(make_line( make_point_m( 0,0,1 ), make_point_m( -1,-1,2 ), make_point_m( -2,-2,0 ) ) )` → 0 |

### main_angle

도형을 완전히 덮는 지향 (oriented) 최소 경계 직사각형의 (북쪽에서 시계 방향으로 도 단위인) 장축 각도를 반환합니다.

| | |
|---|---|
| 문법 | main_angle(geometry) |
| 인자 | • **geometry** - 도형 |
| 예제 | • `main_angle(geom_from_wkt('Polygon ((321577 129614, 321581 129618, 321585 129615, 321581 129610, 321577 129614))'))` → 38.66 |

### make_circle

원형 폴리곤을 생성합니다.

| | |
|---|---|
| 문법 | make_circle(center, radius, [segments=36])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **center** - 원의 중심 포인트<br>• **radius** - 원의 반지름<br>• **segments** - 폴리곤 선분화를 위한 선택적인 인자. 기본값은 36 |
| 예제 | • `geom_to_wkt(make_circle(make_point(10,10), 5, 4))` → 'Polygon ((10 15, 15 10, 10 5, 5 10, 10 15))'<br>• `geom_to_wkt(make_circle(make_point(10,10,5), 5, 4))` → 'PolygonZ ((10 15 5, 15 10 5, 10 5 5, 5 10 5, 10 15 5))'<br>• `geom_to_wkt(make_circle(make_point(10,10,5,30), 5, 4))` → 'PolygonZM ((10 15 5 30, 15 10 5 30, 10 5 5 30, 5 10 5 30, 10 15 5 30))' |

### make_ellipse

타원형 폴리곤을 생성합니다.

| | |
|---|---|
| 문법 | make_ellipse(center, semi_major_axis, semi_minor_axis, azimuth, [segments=36])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **center** - 타원의 중심 포인트<br>• **semi_major_axis** - 타원의 긴반지름<br>• **semi_minor_axis** - 타원의 짧은반지름<br>• **azimuth** - 타원의 방향<br>• **segments** - 폴리곤 선분화를 위한 선택적인 인자. 기본값은 36 |
| 예제 | • `geom_to_wkt(make_ellipse(make_point(10,10), 5, 2, 90, 4))` → 'Polygon ((15 10, 10 8, 5 10, 10 12, 15 10))'<br>• `geom_to_wkt(make_ellipse(make_point(10,10,5), 5, 2, 90, 4))` → 'PolygonZ ((15 10 5, 10 8 5, 5 10 5, 10 12 5, 15 10 5))'<br>• `geom_to_wkt(make_ellipse(make_point(10,10,5,30), 5, 2, 90, 4))` → 'PolygonZM ((15 10 5 30, 10 8 5 30, 5 10 5 30, 10 12 5 30, 15 10 5 30))' |

### make_line

일련의 포인트 도형들로부터 라인 도형을 생성합니다.

**인자 변이형 (variant) 목록**

라인 꼭짓점을 함수에 대한 개별 인자로 지정합니다.

| | |
|---|---|
| 문법 | make_line(point1, point2, ⋯) |
| 인자 | • **point** - 포인트 도형 (또는 포인트 배열) |
| 예제 | • `geom_to_wkt(make_line(make_point(2,4),make_point(3,5)))` → 'LineString (2 4, 3 5)'<br>• `geom_to_wkt(make_line(make_point(2,4),make_point(3,5), make_point(9,7)))` → 'LineString (2 4, 3 5, 9 7)' |

**배열 변이형**

라인 꼭짓점을 포인트 배열로 지정합니다.

| | |
|---|---|
| 문법 | make_line(array) |
| 인자 | • **array** - 포인트의 배열 |
| 예제 | • `geom_to_wkt(make_line(array(make_point(2,4),make_point(3,5), make_point(9,7))))` → 'LineString (2 4, 3 5, 9 7)' |

### make_point

X 와 Y (그리고 부가적인 Z 및 M) 값으로부터 포인트 도형을 생성합니다.

| | |
|---|---|
| 문법 | make_point(x, y, [z], [m])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **x** - 포인트의 X 좌표<br>• **y** - 포인트의 Y 좌표<br>• **z** - 포인트의 부가적인 Z 좌표<br>• **m** - 포인트의 부가적인 M 값 |
| 예제 | • `geom_to_wkt(make_point(2,4))` → 'Point (2 4)'<br>• `geom_to_wkt(make_point(2,4,6))` → 'PointZ (2 4 6)'<br>• `geom_to_wkt(make_point(2,4,6,8))` → 'PointZM (2 4 6 8)' |

### make_point_m

X, Y 좌표와 M 값으로 포인트 도형을 생성합니다.

| | |
|---|---|
| 문법 | make_point_m(x, y, m) |
| 인자 | • **x** - 포인트의 X 좌표<br>• **y** - 포인트의 Y 좌표<br>• **m** - 포인트의 M 값 |
| 예제 | • `geom_to_wkt(make_point_m(2,4,6))` → 'PointM (2 4 6)' |

### make_polygon

외곽 고리와 부가적인 일련의 내곽 고리 도형으로부터 폴리곤 도형을 생성합니다.

| | |
|---|---|
| 문법 | make_polygon(outerRing, [innerRing1], [innerRing2], ⋯)<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **outerRing** - 폴리곤의 외곽 고리가 될 닫힌 라인 도형<br>• **innerRing** - 폴리곤의 내곽 고리가 될 선택적인 닫힌 라인 도형 |
| 예제 | • `geom_to_wkt(make_polygon(geom_from_wkt('LINESTRING( 0 0, 0 1, 1 1, 1 0, 0 0 )')))` → 'Polygon ((0 0, 0 1, 1 1, 1 0, 0 0))'<br>• `geom_to_wkt(make_polygon(geom_from_wkt('LINESTRING( 0 0, 0 1, 1 1, 1 0, 0 0 )'),geom_from_wkt('LINESTRING( 0.1 0.1, 0.1 0.2, 0.2 0.2, 0.2 0.1, 0.1 0.1 )'),geom_from_wkt('LINESTRING( 0.8 0.8, 0.8 0.9, 0.9 0.9, 0.9 0.8, 0.8 0.8 )')))` → 'Polygon ((0 0, 0 1, 1 1, 1 0, 0 0),(0.1 0.1, 0.1 0.2, 0.2 0.2, 0.2 0.1, 0.1 0.1),(0.8 0.8, 0.8 0.9, 0.9 0.9, 0.9 0.8, 0.8 0.8))' |

### make_rectangle_3points

포인트 3 개로부터 직사각형 폴리곤을 생성합니다.

| | |
|---|---|
| 문법 | make_rectangle_3points(point1, point2, point3, [option=0])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **point1** - 첫 번째 포인트<br>• **point2** - 두 번째 포인트<br>• **point3** - 세 번째 포인트<br>• **option** - 직사각형을 구성하기 위한 부가적인 인자입니다. 이 값은 기본적으로 0 입니다. 0(거리) 또는 1(투영) 이 될 수 있습니다. 거리 옵션: 두 번째 거리는 두 번째와 세 번째 포인트 사이의 거리와 동일합니다. 투영 옵션: 두 번째 거리는 선분 또는 선분의 연장 위에 있는 세 번째 포인트의 수직 투영의 거리와 동일합니다. |
| 예제 | • `geom_to_wkt(make_rectangle_3points(make_point(0, 0), make_point(0,5), make_point(5, 5), 0))` → 'Polygon ((0 0, 0 5, 5 5, 5 0, 0 0))'<br>• `geom_to_wkt(make_rectangle_3points(make_point(0, 0), make_point(0,5), make_point(5, 3), 1))` → 'Polygon ((0 0, 0 5, 5 5, 5 0, 0 0))' |

### make_regular_polygon

정다각형 폴리곤을 생성합니다.

| | |
|---|---|
| 문법 | make_regular_polygon(center, radius, number_sides, [circle=0])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **center** - 정다각형 폴리곤의 중심<br>• **radius** - 두 번째 포인트. 정다각형 폴리곤이 내접되는 경우 첫 번째가 됩니다. 정다각형 폴리곤이 외접되는 경우 첫 번째 변의 중간점이 됩니다.<br>• **number_sides** - 정다각형 폴리곤의 변/경계선의 개수<br>• **circle** - 정다각형 폴리곤을 구성하기 위한 선택적인 인자. 기본값은 0 입니다. 0(내접) 또는 1(외접) 이 될 수 있습니다. |
| 예제 | • `geom_to_wkt(make_regular_polygon(make_point(0,0), make_point(0,5), 5))` → 'Polygon ((0 5, 4.76 1.55, 2.94 -4.05, -2.94 -4.05, -4.76 1.55, 0 5))'<br>• `geom_to_wkt(make_regular_polygon(make_point(0,0), project(make_point(0,0), 4.0451, radians(36)), 5))` → 'Polygon ((0 5, 4.76 1.55, 2.94 -4.05, -2.94 -4.05, -4.76 1.55, 0 5))' |

### make_square

대각선으로부터 정사각형을 생성합니다.

| | |
|---|---|
| 문법 | make_square(point1, point2) |
| 인자 | • **point1** - 대각선의 첫 번째 포인트<br>• **point2** - 대각선의 마지막 포인트 |
| 예제 | • `geom_to_wkt(make_square( make_point(0,0), make_point(5,5)))` → 'Polygon ((0 0, -0 5, 5 5, 5 0, 0 0))'<br>• `geom_to_wkt(make_square( make_point(5,0), make_point(5,5)))` → 'Polygon ((5 0, 2.5 2.5, 5 5, 7.5 2.5, 5 0))' |

### make_triangle

삼각형 폴리곤을 생성합니다.

| | |
|---|---|
| 문법 | make_triangle(point1, point2, point3) |
| 인자 | • **point1** - 삼각형의 첫 번째 포인트<br>• **point2** - 삼각형의 두 번째 포인트<br>• **point3** - 삼각형의 세 번째 포인트 |
| 예제 | • `geom_to_wkt(make_triangle(make_point(0,0), make_point(5,5), make_point(0,10)))` → 'Triangle ((0 0, 5 5, 0 10, 0 0))'<br>• `geom_to_wkt(boundary(make_triangle(make_point(0,0), make_point(5,5), make_point(0,10))))` → 'LineString (0 0, 5 5, 0 10, 0 0)' |

### make_valid

무결한 도형을, 또는 도형을 무결하게 수정할 수 없는 경우 비어 있는 도형을 반환합니다.

| | |
|---|---|
| 문법 | make_valid(geometry, [method=structure], [keep_collapsed=false])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **geometry** - 도형<br>• **method** - 수정 알고리즘. 'structure'또는'linework'가운데 하나일 수도 있습니다. 'linework'옵션은 모든 고리를 노드화된 라인 집합으로 결합한 다음 해당 라인워크로부터 무결한 폴리곤을 추출합니다. 'structure'메소드는 먼저 모든 고리를 무결하게 만든 다음 셸 (shell) 들을 병합하고, 무결한 산출물을 생성하기 위해 셸에서 구멍을 뺍니다. 구멍 및 셸이 정확하게 분류되어 있다고 가정합니다.<br>• **keep_collapsed** - 참으로 설정한 경우, 더 낮은 차원으로 무너진 요소를 – 예를 들어 라인으로 무너진 고리 또는 포인트로 무너진 라인을–유지할 것입니다. |
| 예제 | • `geom_to_wkt(make_valid(geom_from_wkt('POLYGON((3 2, 4 1, 5 8, 3 2, 4 2))')))` → 'Polygon ((3 2, 5 8, 4 1, 3 2))'<br>• `geom_to_wkt(make_valid(geom_from_wkt('POLYGON((3 2, 4 1, 5 8, 3 2, 4 2))'), 'linework'))` → 'GeometryCollection (Polygon ((5 8, 4 1, 3 2, 5 8)),LineString (3 2, 4 2))'<br>• `geom_to_wkt(make_valid(geom_from_wkt('POLYGON((3 2, 4 1, 5 8, 3 2))'), method:='linework'))` → 'Polygon ((3 2, 4 1, 5 8, 3 2))'<br>• `make_valid(geom_from_wkt('LINESTRING(0 0)'))` → 비어 있는 도형 |

더 읽어볼 거리: *is_valid*, 도형 고치기 알고리즘

### minimal_circle

도형의 최소 외함 원을 반환합니다. 최소 외함 원이란 도형 집합에 있는 모든 도형을 감싸는 최소 원을 말합니다.

| | |
|---|---|
| 문법 | minimal_circle(geometry, [segments=36])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **geometry** - 도형<br>• **segments** - 폴리곤 선분화를 위한 선택적인 인자. 기본값은 36 |
| 예제 | • `geom_to_wkt( minimal_circle( geom_from_wkt( 'LINESTRING(0 5, 0 -5, 2 1)' ), 4 ) )` → 'Polygon ((0 5, 5 -0, -0 -5, -5 0, 0 5))'<br>• `geom_to_wkt( minimal_circle( geom_from_wkt( 'MULTIPOINT(1 2, 3 4, 3 2)' ), 4 ) )` → 'Polygon ((3 4, 3 2, 1 2, 1 4, 3 4))' |

더 읽어볼 거리: 최소 외함 원 알고리즘

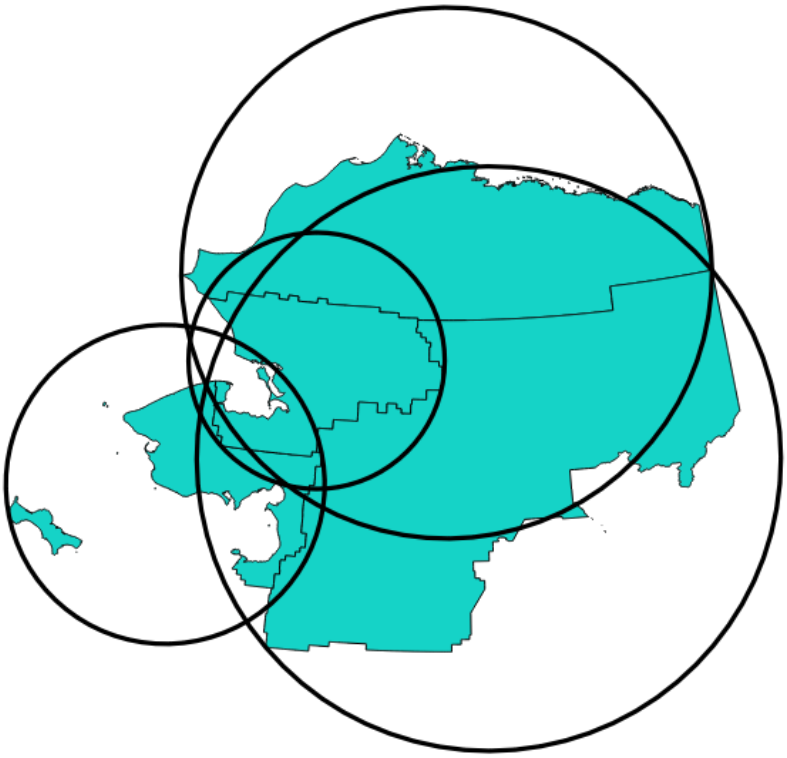


그림 9.22: 각 피처의 최소 외함 원

### nodes_to_points

입력 도형에 있는 모든 노드로 이루어진 멀티포인트 도형을 반환합니다.

| | |
|---|---|
| 문법 | nodes_to_points(geometry, [ignore_closing_nodes=false])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **geometry** - 도형 객체<br>• **ignore_closing_nodes** - 라인 또는 폴리곤 고리를 닫는 중복 노드들을 포함할지 여부를 지정하는 부가적인 인자입니다. 기본값은 거짓으로, 산출 집합에 이 중복 노드들이 포함되는 걸 피하려면 참으로 설정하세요. |
| 예제 | • `geom_to_wkt(nodes_to_points(geom_from_wkt('LINESTRING(0 0, 1 1, 2 2)')))` → 'MultiPoint ((0 0),(1 1),(2 2))'<br>• `geom_to_wkt(nodes_to_points(geom_from_wkt('POLYGON((-1 -1, 4 0, 4 2, 0 2, -1 -1))'),true))` → 'MultiPoint ((-1 -1),(4 0),(4 2),(0 2))' |

더 읽어볼 거리: 꼭짓점 추출하기 알고리즘

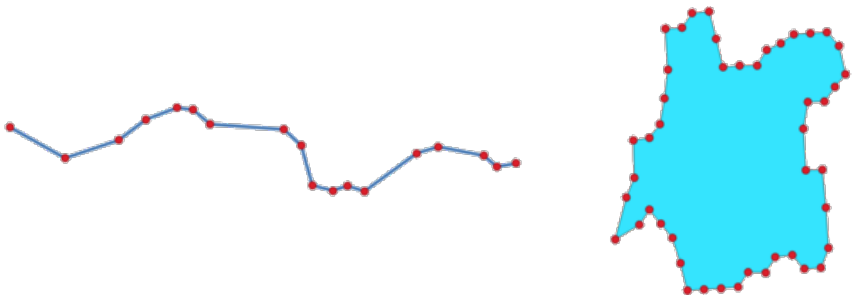


그림 9.23: 꼭짓점들로부터 추출한 멀티포인트 피처

### num_geometries

도형 집합에 있는 도형의 개수를, 또는 다중 부분 도형에 있는 부분들의 개수를 반환합니다. 이 함수는 입력 도형이 집합이 아닌 경우 NULL 을 반환합니다.

| 문법 | num_geometries(geometry) |
|---|---|
| 인자 | • **geometry** - 도형 집합 또는 다중 부분 도형 |
| 예제 | • `num_geometries(geom_from_wkt('GEOMETRYCOLLECTION(POINT(0 1), POINT(0 0), POINT(1 0), POINT(1 1))'))` → 4<br>• `num_geometries(geom_from_wkt('MULTIPOINT((0 1), (0 0), (1 0))'))` → 3 |

### num_interior_rings

폴리곤 또는 도형 집합에 있는 내곽 고리의 개수를 반환하거나, 또는 입력 도형이 폴리곤 또는 집합이 아닌 경우 NULL 을 반환합니다.

| 문법 | num_interior_rings(geometry) |
|---|---|
| 인자 | • **geometry** - 입력 도형 |
| 예제 | • `num_interior_rings(geom_from_wkt('POLYGON((-1 -1, 4 0, 4 2, 0 2, -1 -1),(-0.1 -0.1, 0.4 0, 0.4 0.2, 0 0.2, -0.1 -0.1))'))` → 1 |

### num_points

도형의 꼭짓점 개수를 반환합니다.

| | |
|---|---|
| 문법 | num_points(geometry) |
| 인자 | • **geometry** - 도형 |
| 예제 | • `num_points(@geometry)` → 현재 피처 도형에 있는 꼭짓점의 개수 |

### num_rings

폴리곤 또는 도형 집합에 있는 (외곽 고리를 포함하는) 고리의 개수를 반환하거나, 또는 입력 도형이 폴리곤 또는 집합이 아닌 경우 NULL 을 반환합니다.

| | |
|---|---|
| 문법 | num_rings(geometry) |
| 인자 | • **geometry** - 입력 도형 |
| 예제 | • `num_rings(geom_from_wkt('POLYGON((-1 -1, 4 0, 4 2, 0 2, -1 -1),(-0.1 -0.1, 0.4 0, 0.4 0.2, 0 0.2, -0.1 -0.1))'))` → 2 |

### offset_curve

라인스트링 도형을 한쪽으로 오프셋시켜 형성된 도형을 반환합니다. 거리 단위는 해당 도형의 공간 좌표계의 단위입니다.

| | |
|---|---|
| 문법 | offset_curve(geometry, distance, [segments=8], [join=1], [miter_limit=2.0])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **geometry** - (멀티) 라인스트링 도형<br>• **distance** - 오프셋 거리입니다. 양의 값이면 라인 좌측으로 버퍼를 적용하고, 음의 값이면 라인 우측으로 적용할 것입니다<br>• **segments** - 둥근 결합 스타일을 사용한 경우 사분원을 나타내는 데 사용할 선분의 개수입니다. 숫자가 클수록 더 많은 노드를 가진 매끈한 라인을 생성합니다.<br>• **join** - 모서리 부분의 결합 스타일. 1 = 둥글게, 2 = 마이터, 그리고 3 = 비스듬하게<br>• **miter_limit** - 매우 뾰족한 모서리에 사용된 마이터 비율에 대한 제한 (마이터 결합만 사용하는 경우) |
| 예제 | • `offset_curve(@geometry, 10.5)` → 라인 왼쪽으로 10.5 단위만큼 오프셋<br>• `offset_curve(@geometry, -10.5)` → 라인 오른쪽으로 10.5 단위만큼 오프셋<br>• `offset_curve(@geometry, 10.5, segments:=16, join:=1)` → 라인 왼쪽으로 10.5 단위만큼 오프셋, 선분을 더 많이 사용할 수록 더 매끈한 만곡 도형을 생성합니다.<br>• `offset_curve(@geometry, 10.5, join:=3)` → 라인 왼쪽으로 10.5 단위만큼 오프셋, 'bevel'결합 사용 |

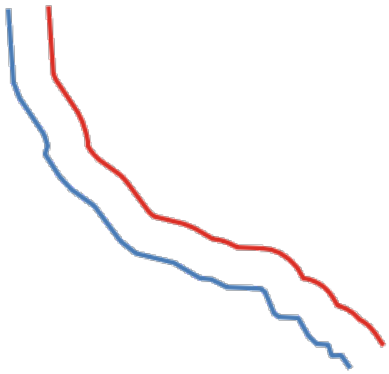


그림 9.24: 소스 피처는 파란색, 오프셋 피처는 빨간색

더 읽어볼 거리: 라인 오프셋시키기 알고리즘

### order_parts

지정한 기준으로 멀티 도형의 부분들을 정렬합니다.

| | |
|---|---|
| 문법 | order_parts(geometry, orderby, [ascending=true])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **geometry** - 멀티 유형 도형<br>• **orderby** - 정렬 기준을 정의하는 표현식 문자열<br>• **ascending** - 불 (boolean), 오름차순의 경우 참, 내림차순의 경우 거짓 |
| 예제 | • `geom_to_wkt(order_parts(geom_from_wkt('MultiPolygon (((1 1, 5 1, 5 5, 1 5, 1 1)),((1 1, 9 1, 9 9, 1 9, 1 1)))'), 'area(@geometry)', False))` → 'MultiPolygon (((1 1, 9 1, 9 9, 1 9, 1 1)),((1 1, 5 1, 5 5, 1 5, 1 1)))'<br>• `geom_to_wkt(order_parts(geom_from_wkt('LineString(1 2, 3 2, 4 3)'), '1', True))` → 'LineString(1 2, 3 2, 4 3)' |

### oriented_bbox

입력 도형의 지향된 (oriented) 최소 경계 상자를 표현하는 도형을 반환합니다.

| | |
|---|---|
| 문법 | oriented_bbox(geometry) |
| 인자 | • **geometry** - 도형 |
| 예제 | • `geom_to_wkt( oriented_bbox( geom_from_wkt( 'MULTIPOINT(1 2, 3 4, 3 2)' ) ) )` → 'Polygon ((3 2, 3 4, 1 4, 1 2, 3 2))' |

더 읽어볼 거리: 기울어진 최소 경계 상자 알고리즘

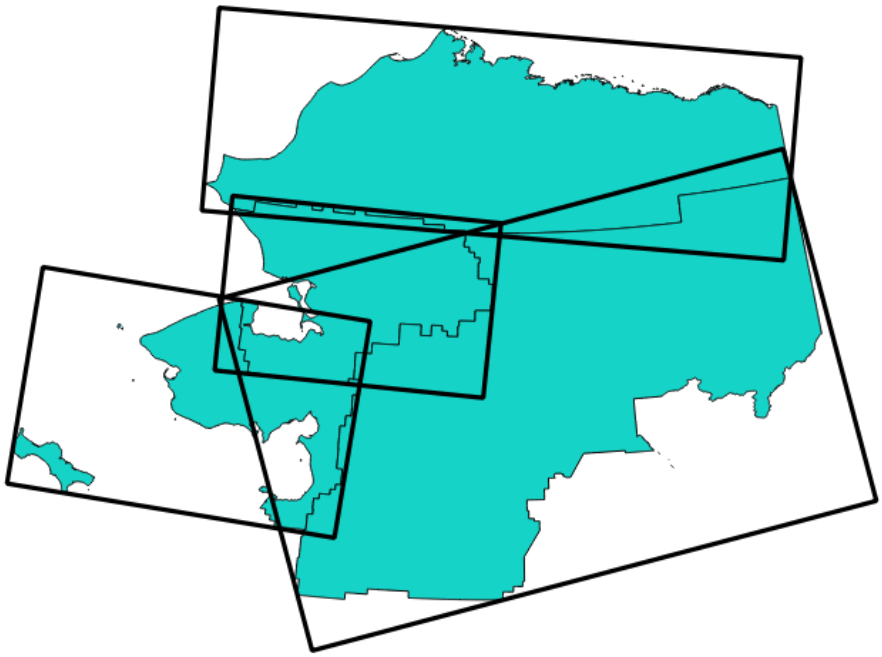


그림 9.25: 기울어진 최소 경계 상자

### overlaps

도형이 다른 도형과 중첩하는지 검증합니다. 도형들이 동일한 차원이며 공간을 공유하지만 서로를 완전히 담고 있지 않은 경우 참을 반환합니다.

| | |
|---|---|
| 문법 | overlaps(geometry1, geometry2) |
| 인자 | • **geometry1** - 도형<br>• **geometry2** - 도형 |
| 예제 | • `overlaps( geom_from_wkt( 'LINESTRING(3 5, 4 4, 5 5, 5 3)' ), geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ) )` → TRUE<br>• `overlaps( geom_from_wkt( 'LINESTRING(0 0, 1 1)' ), geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ) )` → FALSE |

### overlay_contains

현재 피처가 공간적으로 대상 레이어의 피처를 적어도 하나 이상 담고 있는지 여부, 또는 현재 피처가 담고 있는 대상 레이어의 피처에 대한 표현식 기반 결과물의 배열을 반환합니다.

기저 GEOS "Contains"서술부에 대해 자세히 알고 싶다면, PostGIS ST_Contains 함수의 설명을 읽어보세요.

| | |
|---|---|
| 문법 | overlay_contains(layer, [expression], [filter], [limit], [cache=false])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **layer** - 중첩을 검증할 레이어<br>• **expression** - 대상 레이어에서 나온 피처에 대해 평가할 부가적인 표현식입니다. 이 표현식을 설정하지 않은 경우, 이 함수는 최소 1 개의 매칭이 있는지를 나타내는 불(boolean) 을 반환할 것입니다.<br>• **filter** - 검증할 대상 피처를 필터링할 선택적인 표현식입니다. 이 표현식을 설정하지 않은 경우, 모든 피처를 검증할 것입니다.<br>• **limit** - 매칭하는 피처의 개수를 제한할 선택적인 정수입니다. 이 파라미터를 설정하지 않은 경우, 매칭하는 모든 피처를 반환할 것입니다.<br>• **cache** - 이 파라미터를 참으로 설정하면 로컬 공간 인덱스를 작성합니다. (대부분의 경우 바람직하지 않습니다. 사용자가 현저하게 느린 데이터 제공자와 작업하는 게 아니라면 말이죠.) |
| 예제 | • `overlay_contains('regions')` → TRUE, 현재 피처가'regions'를 공간적으로 담고 있는 경우<br>• `overlay_contains('regions', filter:= population > 10000)` → TRUE, 현재 피처가 인구 10,000 명을 초과하는'regions'를 공간적으로 담고 있는 경우<br>• `overlay_contains('regions', name)` → 현재 피처에 담겨 있는'regions'의 이름들의 배열<br>• `array_to_string(overlay_contains('regions', name))` → 현재 피처에 담겨 있는'regions'의 이름들을 쉼표로 구분한 목록인 문자열<br>• `array_sort(overlay_contains(layer:='regions', expression:="name", filter:= population > 10000))` → 현재 피처에 담겨 있으며 인구 10,000 명을 초과하는'regions'의 이름들을 정렬한 배열<br>• `overlay_contains(layer:='regions', expression:= geom_to_wkt(@geometry), limit:=2)` → 현재 피처에 담겨 있는 최대 2 개까지의'regions'의 (WKT 서식) 도형들의 배열 |

더 읽어볼 거리: *contains*, 배열 조작, 위치로 선택하기 알고리즘

### overlay_crosses

현재 피처가 대상 레이어의 피처를 적어도 하나 이상 공간 교차하고 있는지 여부, 또는 현재 피처가 공간 교차하는 대상 레이어의 피처에 대한 표현식 기반 결과물의 배열을 반환합니다.

기저 GEOS "Crosses"서술부에 대해 자세히 알고 싶다면, PostGIS ST_Crosses 함수의 설명을 읽어보세요.

| | |
|---|---|
| 문법 | overlay_crosses(layer, [expression], [filter], [limit], [cache=false])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **layer** - 중첩을 검증할 레이어<br>• **expression** - 대상 레이어에서 나온 피처에 대해 평가할 부가적인 표현식입니다. 이 표현식을 설정하지 않은 경우, 이 함수는 최소 1 개의 매칭이 있는지를 나타내는 불 (boolean) 을 반환할 것입니다.<br>• **filter** - 검증할 대상 피처를 필터링할 선택적인 표현식입니다. 이 표현식을 설정하지 않은 경우, 모든 피처를 검증할 것입니다.<br>• **limit** - 매칭하는 피처의 개수를 제한할 선택적인 정수입니다. 이 파라미터를 설정하지 않은 경우, 매칭하는 모든 피처를 반환할 것입니다.<br>• **cache** - 이 파라미터를 참으로 설정하면 로컬 공간 인덱스를 작성합니다. (대부분의 경우 바람직하지 않습니다. 사용자가 현저하게 느린 데이터 제공자와 작업하는 게 아니라면 말이죠.) |
| 예제 | • `overlay_crosses('regions')` → TRUE, 현재 피처가'regions'를 공간 교차하는 경우<br>• `overlay_crosses('regions', filter:= population > 10000)` → TRUE, 현재 피처가 인구 10,000 명을 초과하는'regions'를 공간 교차하는 경우<br>• `overlay_crosses('regions', name)` → 현재 피처를 공간 교차하는'regions'의 이름들의 배열<br>• `array_to_string(overlay_crosses('regions', name))` → 현재 피처를 공간 교차하는'regions'의 이름들을 쉼표로 구분한 목록인 문자열<br>• `array_sort(overlay_crosses(layer:='regions', expression:="name", filter:= population > 10000))` → 현재 피처를 공간 교차하며 인구 10,000 명을 초과하는'regions'의 이름들을 정렬한 배열<br>• `overlay_crosses(layer:='regions', expression:= geom_to_wkt(@geometry), limit:=2)` → 현재 피처와 공간 교차하는 최대 2 개까지의'regions'의 (WKT 서식) 도형들의 배열 |

더 읽어볼 거리: *crosses*, 배열 조작, 위치로 선택하기 알고리즘

### overlay_disjoint

현재 피처가 공간적으로 대상 레이어의 모든 피처와 공간적으로 분리되어 있는지 여부, 또는 현재 피처와 분리되어 있는 대상 레이어의 피처에 대한 표현식 기반 결과물의 배열을 반환합니다.

기저 GEOS "Disjoint"서술부에 대해 자세히 알고 싶다면, PostGIS ST_Disjoint 함수의 설명을 읽어보세요.

| | |
|---|---|
| 문법 | overlay_disjoint(layer, [expression], [filter], [limit], [cache=false])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **layer** - 중첩을 검증할 레이어<br>• **expression** - 대상 레이어에서 나온 피처에 대해 평가할 부가적인 표현식입니다. 이 표현식을 설정하지 않은 경우, 이 함수는 최소 1 개의 매칭이 있는지를 나타내는 불 (boolean) 을 반환할 것입니다.<br>• **filter** - 검증할 대상 피처를 필터링할 선택적인 표현식입니다. 이 표현식을 설정하지 않은 경우, 모든 피처를 검증할 것입니다.<br>• **limit** - 매칭하는 피처의 개수를 제한할 선택적인 정수입니다. 이 파라미터를 설정하지 않은 경우, 매칭하는 모든 피처를 반환할 것입니다.<br>• **cache** - 이 파라미터를 참으로 설정하면 로컬 공간 인덱스를 작성합니다. (대부분의 경우 바람직하지 않습니다. 사용자가 현저하게 느린 데이터 제공자와 작업하는 게 아니라면 말이죠.) |
| 예제 | • `overlay_disjoint('regions')` → TRUE, 현재 피처가 모든'regions'과 공간적으로 분리되어 있는 경우<br>• `overlay_disjoint('regions', filter:= population > 10000)` → TRUE, 현재 피처가 인구 10,000 명을 초과하는 모든'regions'과 공간적으로 분리되어 있는 경우<br>• `overlay_disjoint('regions', name)` → 현재 피처와 공간적으로 분리되어 있는'regions'의 이름들의 배열<br>• `array_to_string(overlay_disjoint('regions', name))` → 현재 피처와 공간적으로 분리되어 있는'regions'의 이름들을 쉼표로 구분한 목록인 문자열<br>• `array_sort(overlay_disjoint(layer:='regions', expression:="name", filter:= population > 10000))` → 현재 피처와 공간적으로 분리되어 있으며 인구 10,000 명을 초과하는'regions'의 이름들을 정렬한 배열<br>• `overlay_disjoint(layer:='regions', expression:=geom_to_wkt(@geometry), limit:=2)` → 현재 피처와 공간적으로 분리되어 있는 최대 2 개까지의'regions'의 (WKT 서식) 도형들의 배열 |

더 읽어볼 거리: *disjoint*, 배열 조작, 위치로 선택하기 알고리즘

### overlay_equals

Returns whether the current feature's geometry is exactly equal to at least one feature's geometry from a target layer, or an array of expression-based results for the features in the target layer whose geometry is exactly equal to the current feature's geometry. Note that the order of vertices matters.

| | |
|---|---|
| 문법 | overlay_equals(layer, [expression], [filter], [limit], [cache=false])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **layer** - 중첩을 검증할 레이어<br>• **expression** - 대상 레이어에서 나온 피처에 대해 평가할 부가적인 표현식입니다. 이 표현식을 설정하지 않은 경우, 이 함수는 최소 1 개의 매칭이 있는지를 나타내는 불(boolean) 을 반환할 것입니다.<br>• **filter** - 검증할 대상 피처를 필터링할 선택적인 표현식입니다. 이 표현식을 설정하지 않은 경우, 모든 피처를 검증할 것입니다.<br>• **limit** - 매칭하는 피처의 개수를 제한할 선택적인 정수입니다. 이 파라미터를 설정하지 않은 경우, 매칭하는 모든 피처를 반환할 것입니다.<br>• **cache** - 이 파라미터를 참으로 설정하면 로컬 공간 인덱스를 작성합니다. (대부분의 경우 바람직하지 않습니다. 사용자가 현저하게 느린 데이터 제공자와 작업하는 게 아니라면 말이죠.) |
| 예제 | • `overlay_equals('regions')` → TRUE if the current feature's geometry is exactly the same as the geometry of a region<br>• `overlay_equals('regions', filter:= population > 10000)` → TRUE if the current feature's geometry is exactly the same as the geometry of a region whose population is greater than 10000<br>• `overlay_equals('regions', name)` → an array of names, for the regions exactly equal to the current feature<br>• `array_to_string(overlay_equals('regions', name))` → a string as a comma separated list of names, for the regions exactly equal to the current feature<br>• `array_sort(overlay_equals(layer:='regions', expression:="name", filter:= population > 10000))` → an ordered array of names, for the regions exactly equal to the current feature and with a population greater than 10000<br>• `overlay_equals(layer:='regions', expression:=geom_to_wkt(@geometry), limit:=2)` → an array of geometries (in WKT), for up to two regions exactly equal to the current feature |

**참고**

This function requires exact equality of geometries, meaning that the geometries 'LINESTRING( 0 0, 1 1 )'and 'LINESTRING( 1 1, 0 0 )'are different.

더 읽어볼 거리: 배열 조작, 위치로 선택하기 알고리즘

### overlay_intersects

현재 피처가 대상 레이어의 피처를 적어도 하나 이상 공간적으로 교차하고 있는지 여부, 또는 현재 피처가 교차하는 대상 레이어의 피처에 대한 표현식 기반 결과물의 배열을 반환합니다.

기저 GEOS "Intersects"서술부에 대해 자세히 알고 싶다면, PostGIS ST_Intersects 함수의 설명을 읽어보세요.

| | |
|---|---|
| 문법 | overlay_intersects(layer, [expression], [filter], [limit], [cache=false], [min_overlap], [min_inscribed_circle_radius], [return_details], [sort_by_intersection_size])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **layer** - 중첩을 검증할 레이어<br>• **expression** - 대상 레이어에서 나온 피처에 대해 평가할 부가적인 표현식입니다. 이 표현식을 설정하지 않은 경우, 이 함수는 최소 1 개의 매칭이 있는지를 나타내는 불 (boolean) 을 반환할 것입니다.<br>• **filter** - 검증할 대상 피처를 필터링할 선택적인 표현식입니다. 이 표현식을 설정하지 않은 경우, 모든 피처를 검증할 것입니다.<br>• **limit** - 매칭하는 피처의 개수를 제한할 선택적인 정수입니다. 이 파라미터를 설정하지 않은 경우, 매칭하는 모든 피처를 반환할 것입니다.<br>• **cache** - 이 파라미터를 참으로 설정하면 로컬 공간 인덱스를 작성합니다. (대부분의 경우 바람직하지 않습니다. 사용자가 현저하게 느린 데이터 제공자와 작업하는 게 아니라면 말이죠.)<br>• **min_overlap** - 선택적인 제외 필터를 정의합니다:<br>– 폴리곤의 경우, 현재 피처의 제곱 단위의 최소 교차 영역 면적입니다. 교차가 여러 폴리곤에서 일어나는 경우, 이 값 이상의 면적을 가진 폴리곤이 최소 하나 있다면 교차 영역을 반환할 것입니다.<br>– 라인의 경우, 현재 피처 단위의 최단 교차 영역 길이입니다. 교차가 여러 라인에서 일어나는 경우, 이 값 이상의 길이를 가진 라인이 최소 하나 있다면 교차 영역을 반환할 것입니다.<br>• **min_inscribed_circle_radius** - 선택적인 (폴리곤 전용) 제외 필터를 정의합니다: 교차 영역 최대 내접원의 현재 피처 단위 최단 반경입니다. 교차가 여러 폴리곤에서 일어나는 경우, 이 값 이상의 최대 내접원 반경을 가진 폴리곤이 최소 하나 있다면 교차 영역을 반환할 것입니다.<br>기저 GEOS 서술부에 대해 자세히 알고 싶다면, PostGIS ST_MaximumInscribedCircle 함수의 설명을 읽어보세요.<br>이 인자를 사용하려면 GEOS 3.9 버전 이상이 필요합니다.<br>• **return_details** - Set this to true to return a list of maps containing (key names in quotes) the feature 'id', the expression 'result'and the 'overlap'value (of the largest element in case of multipart). The 'radius'of the maximum inscribed circle is also returned when the target layer is a polygon. Only valid when used with the expression parameter<br>• **sort_by_intersection_size** - 표현식과 함께 사용하는 경우에만 무결합니다. 이 인자를 'des'로 설정하면 중첩 값이 내림차순으로 정렬된 산출물을 반환합니다. 'asc'로 설정하면 오름차순으로 정렬된 산출물을 반환합니다. |
| 예제 | • `overlay_intersects('regions')` → TRUE, 현재 피처가'regions'와 공간적으로 교차하는 경우<br>• `overlay_intersects('regions', filter:= population > 10000)` → TRUE, 현재 피처가 인구 10,000 명을 초과하는'regions'를 공간적으로 교차하는 경우<br>• `overlay_intersects('regions', name)` → 현재 피처를 교차하는'regions'의 이름들의 배열<br>• `array_to_string(overlay_intersects('regions', name))` → 현재 피처를 교차하는'regions'의 이름들을 쉼표로 구분한 목록인 문자열<br>• `array_sort(overlay_intersects(layer:='regions', expression:="name", filter:= population > 10000))` → 현재 피처를 교차하며 인구 10,000 명을 초과하는'regions'의 이름들을 정렬한 배열<br>• `overlay_intersects(layer:='regions', expression:=geom_to_wkt(@geometry), limit:=2)` → 현재 피처와 교차하는 최대 2 개까지의'regions'의 (WKT 서식) 도형들의 배열<br>• `overlay_intersects(layer:='regions', min_overlap:=0.54)` → TRUE, 현재 피처가'regions'와 공간적으로 교차하고 교차 영역 면적이 (멀티폴리곤의 경우 최소한 하나 이상의 부분의 면적이) 0.54 이상인 경우<br>• `overlay_intersects(layer:='regions', min_inscribed_circle_radius:=0.54)` → TRUE, 현재 피처가'regions'와 공간적으로 교차하고 교차 영역의 (다중 부분 도형의 경우 최소한 하나 이상의 부분의) 최대 내접원의 반경이 0.54 이상인 경우<br>• `overlay_intersects(layer:='regions', expression:=geom_to_wkt(@geometry), return_details:=true)` → 'id', 'result', |

더 읽어볼 거리: *intersects*, 배열 조작, 위치로 선택하기 알고리즘

### overlay_nearest

현재 피처에서 지정한 거리 안에 대상 레이어의 피처 (들) 이 있는지 여부, 또는 현재 피처에서 지정한 거리 안에 있는 대상 레이어의 피처에 대한 표현식 기반 결과물의 배열을 반환합니다.

주의: 대용량 레이어의 경우 이 함수는 메모리를 많이 잡아먹고 느릴 수도 있습니다.

| | |
|---|---|
| 문법 | overlay_nearest(layer, [expression], [filter], [limit=1], [max_distance], [cache=false])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **layer** - 대상 레이어<br>• **expression** - 대상 레이어에서 나온 피처에 대해 평가할 부가적인 표현식입니다. 이 표현식을 설정하지 않은 경우, 이 함수는 최소 1 개의 매칭이 있는지를 나타내는 불 (boolean) 을 반환할 것입니다.<br>• **filter** - 검증할 대상 피처를 필터링할 선택적인 표현식입니다. 이 표현식을 설정하지 않은 경우, 대상 레이어에 있는 모든 피처를 검증할 것입니다.<br>• **limit** - 매칭하는 피처의 개수를 제한할 선택적인 정수입니다. 이 파라미터를 설정하지 않은 경우, 가장 가까이 있는 피처만 반환할 것입니다. -1 로 설정하면, 매칭하는 모든 피처를 반환합니다.<br>• **max_distance** - 매칭하는 피처를 검색할 거리를 제한하는 선택적인 파라미터입니다. 설정하지 않은 경우, 대상 레이어에 있는 모든 피처를 사용할 것입니다.<br>• **cache** - 이 파라미터를 참으로 설정하면 로컬 공간 인덱스를 작성합니다. (대부분의 경우 바람직하지 않습니다. 사용자가 현저하게 느린 데이터 제공자와 작업하는 게 아니라면 말이죠.) |
| 예제 | • `overlay_nearest('airports')` → TRUE, "airports"레이어에 매칭하는 피처가 최소한 하나 이상 있는 경우<br>• `overlay_nearest('airports', max_distance:= 5000)` → TRUE, 현재 피처에서 5,000 맵 단위 거리 안에'airport'레이어의 피처가 있는 경우<br>• `overlay_nearest('airports', name)` → 현재 피처에 가장 가까이 있는'airport' 레이어의 피처 이름을 배열로 반환합니다.<br>• `array_to_string(overlay_nearest('airports', name))` → 현재 피처에 가장 가까이 있는'airport'레이어의 피처 이름을 문자열로 반환합니다.<br>• `overlay_nearest(layer:='airports', expression:= name, max_distance:= 5000)` → 현재 피처의 5,000 맵 단위 거리 안에 있는'airport' 레이어의 피처 가운데 가장 가까이 있는 피처의 이름을 배열로 반환합니다.<br>• `overlay_nearest(layer:='airports', expression:="name", filter:= "Use"='Civilian', limit:=3)` → 현재 피처에 가장 가까이 있는 최대 3 개까지의 민간 공항을 거리 순으로 정렬한 공항 이름의 배열을 반환합니다.<br>• `overlay_nearest(layer:='airports', expression:="name", limit:= -1, max_distance:= 5000)` → 현재 피처에서 5,000 맵 단위 거리 안에 있는 모든'airport'레이어의 피처를 거리 순으로 정렬한 공항 이름의 배열을 반환합니다. |

더 읽어볼 거리: 배열 조작, 최근접으로 속성 결합하기 알고리즘

### overlay_touches

현재 피처가 대상 레이어의 피처를 적어도 하나 이상 공간적으로 접하고 있는지 여부, 또는 현재 피처가 접하고 있는 대상 레이어의 피처에 대한 표현식 기반 결과물의 배열을 반환합니다.

기저 GEOS "Touches"서술부에 대해 자세히 알고 싶다면, PostGIS ST_Touches 함수의 설명을 읽어보세요.

| | |
|---|---|
| 문법 | overlay_touches(layer, [expression], [filter], [limit], [cache=false])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **layer** - 중첩을 검증할 레이어<br>• **expression** - 대상 레이어에서 나온 피처에 대해 평가할 부가적인 표현식입니다. 이 표현식을 설정하지 않은 경우, 이 함수는 최소 1 개의 매칭이 있는지를 나타내는 불(boolean) 을 반환할 것입니다.<br>• **filter** - 검증할 대상 피처를 필터링할 선택적인 표현식입니다. 이 표현식을 설정하지 않은 경우, 모든 피처를 검증할 것입니다.<br>• **limit** - 매칭하는 피처의 개수를 제한할 선택적인 정수입니다. 이 파라미터를 설정하지 않은 경우, 매칭하는 모든 피처를 반환할 것입니다.<br>• **cache** - 이 파라미터를 참으로 설정하면 로컬 공간 인덱스를 작성합니다. (대부분의 경우 바람직하지 않습니다. 사용자가 현저하게 느린 데이터 제공자와 작업하는 게 아니라면 말이죠.) |
| 예제 | • `overlay_touches('regions')` → TRUE, 현재 피처가'regions'를 공간적으로 접하는 경우<br>• `overlay_touches('regions', filter:= population > 10000)` → TRUE, 현재 피처가 인구 10,000 명을 초과하는'regions'를 공간적으로 접하는 경우<br>• `overlay_touches('regions', name)` → 현재 피처를 접하고 있는'regions'의 이름들의 배열<br>• `array_to_string(overlay_touches('regions', name))` → a string as a comma separated list of names, for the regions touched by the current feature<br>• `array_sort(overlay_touches(layer:='regions', expression:="name", filter:= population > 10000))` → 현재 피처를 접하며 인구 10,000 명을 초과하는'regions'의 이름들을 정렬한 배열<br>• `overlay_touches(layer:='regions', expression:=geom_to_wkt(@geometry), limit:=2)` → 현재 피처와 접하는 최대 2 개까지의 'regions'의 (WKT 서식) 도형들의 배열 |

더 읽어볼 거리: *touches*, 배열 조작, 위치로 선택하기 알고리즘

### overlay_within

현재 피처가 공간적으로 적어도 하나 이상의 대상 레이어의 피처 내부에 있는지 여부, 또는 현재 피처를 담고 있는 대상 레이어의 피처에 대한 표현식 기반 결과물의 배열을 반환합니다.

기저 GEOS "Within"서술부에 대해 자세히 알고 싶다면, PostGIS ST_Within 함수의 설명을 읽어보세요.

| | |
|---|---|
| 문법 | overlay_within(layer, [expression], [filter], [limit], [cache=false])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **layer** - 중첩을 검증할 레이어<br>• **expression** - 대상 레이어에서 나온 피처에 대해 평가할 부가적인 표현식입니다. 이 표현식을 설정하지 않은 경우, 이 함수는 최소 1 개의 매칭이 있는지를 나타내는 불 (boolean) 을 반환할 것입니다.<br>• **filter** - 검증할 대상 피처를 필터링할 선택적인 표현식입니다. 이 표현식을 설정하지 않은 경우, 모든 피처를 검증할 것입니다.<br>• **limit** - 매칭하는 피처의 개수를 제한할 선택적인 정수입니다. 이 파라미터를 설정하지 않은 경우, 매칭하는 모든 피처를 반환할 것입니다.<br>• **cache** - 이 파라미터를 참으로 설정하면 로컬 공간 인덱스를 작성합니다. (대부분의 경우 바람직하지 않습니다. 사용자가 현저하게 느린 데이터 제공자와 작업하는 게 아니라면 말이죠.) |
| 예제 | • `overlay_within('regions')` → TRUE, 현재 피처가 공간적으로'regions'의 피처 내부에 있는 경우<br>• `overlay_within('regions', filter:= population > 10000)` → TRUE, 현재 피처가 공간적으로 인구 10,000 명을 초과하는'regions'내부에 있는 경우<br>• `overlay_within('regions', name)` → 현재 피처를 담고 있는'regions'의 이름들의 배열<br>• `array_to_string(overlay_within('regions', name))` → 현재 피처를 담고 있는'regions'의 이름들을 쉼표로 구분한 목록인 문자열<br>• `array_sort(overlay_within(layer:='regions', expression:="name", filter:= population > 10000))` → 현재 피처를 담고 있으며 인구 10,000 명을 초과하는'regions'의 이름들을 정렬한 배열<br>• `overlay_within(layer:='regions', expression:=geom_to_wkt(@geometry), limit:=2)` → 현재 피처를 담고 있는 최대 2 개까지의'regions'의 (WKT 서식) 도형들의 배열 |

더 읽어볼 거리: *within*, 배열 조작, 위치로 선택하기 알고리즘

### $perimeter

현재 객체의 둘레 길이를 반환합니다. 이 함수로 계산된 둘레는 현재 프로젝트의 타원체 설정과 거리 단위 설정을 따릅니다. 예를 들어 타원체가 프로젝트에 설정되면 계산된 둘레는 타원체 기반이 되고 타원체가 설정되지 않으면 계산된 둘레는 평면상의 측정이됩니다.

| | |
|---|---|
| 문법 | $perimeter |
| 예제 | • `$perimeter` → 42 |

### perimeter

도형 폴리곤 객체의 둘레를 계산합니다. 언제나 해당 도형의 공간 좌표계 (SRS) 안에서 평면 측량해서 계산하므로, 반환한 둘레의 단위가 SRS 용 단위와 일치할 것입니다. 이것이 $perimeter 함수가 수행하는 계산과 다른 점인데, $perimeter 함수는 프로젝트의 타원체 및 거리 단위 설정을 기반으로 타원체 상에서 계산을 수행할 것입니다.

| | |
|---|---|
| 문법 | perimeter(geometry) |
| 인자 | • **geometry** - 폴리곤 도형 객체 |
| 예제 | • `perimeter(geom_from_wkt('POLYGON((0 0, 4 0, 4 2, 0 2, 0 0))'))` → 12.0 |

### point_n

도형에서 지정한 노드를 반환합니다.

| | |
|---|---|
| 문법 | point_n(geometry, index) |
| 인자 | • **geometry** - 도형 객체<br>• **index** - 1 부터 시작하는 반환할 노드 인덱스; 음의 값인 경우, 선택한 꼭짓점 인덱스는 꼭짓점 총 개수에서 절대값을 뺀 값이 될 것입니다 |
| 예제 | • `geom_to_wkt(point_n(geom_from_wkt('POLYGON((0 0, 4 0, 4 2, 0 2, 0 0))'),2))` → 'Point (4 0)' |

더 읽어볼 거리: 특정 꼭짓점 추출하기 알고리즘

### point_on_surface

도형의 표면 상에 있다고 보장된 포인트를 반환합니다.

| | |
|---|---|
| 문법 | point_on_surface(geometry) |
| 인자 | • **geometry** - 도형 |
| 예제 | • `point_on_surface(@geometry)` → 포인트 도형 |

더 읽어볼 거리: 표면에 포인트 생성하기 알고리즘

### pole_of_inaccessibility

표면에 대한 도달불능극 근사치를 계산합니다. 도달불능극이란 표면의 경계로부터 가장 멀리 떨어져 있는 내부 포인트를 말합니다. 이 함수는'polylabel'알고리즘 (Vladimir Agafonkin, 2016) 을 사용하는데, 이 알고리즘은 지정한 허용 오차 안에서 진짜 도달불능극을 확실히 찾을 수 있는 반복 접근법입니다. 허용 오차가 정밀할수록 더 많이 반복하기 때문에 계산 시간이 더 걸릴 것입니다.

| | |
|---|---|
| 문법 | pole_of_inaccessibility(geometry, tolerance) |
| 인자 | • **geometry** - 도형<br>• **tolerance** - 반환된 포인트와 진극 위치 사이의 최장 거리 |
| 예제 | • `geom_to_wkt(pole_of_inaccessibility( geom_from_wkt('POLYGON((0 1, 0 9, 3 10, 3 3, 10 3, 10 1, 0 1))'), 0.1))` → 'Point(1.546875 2.546875)' |



그림 9.26: 도달불능극

더 읽어볼 거리: 도달불능극 알고리즘

### project

라디안 단위 거리, 방향 (방위각), 그리고 표고를 이용해서 시작점에서부터 투영된 포인트를 반환합니다.

| | |
|---|---|
| 문법 | project(point, distance, azimuth, [elevation])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **point** - 시작점<br>• **distance** - 투영할 거리<br>• **azimuth** - 0 이 진북에 해당하는 시계 방향 라디안 단위 방위각<br>• **elevation** - 라디안 단위의 경사 (inclination) 각도 |
| 예제 | • `geom_to_wkt(project(make_point(1, 2), 3, radians(270)))` → 'Point(-2, 2)' |

더 읽어볼 거리: 포인트 투영하기 (데카르트) 알고리즘

### relate

두 도형 간의 관계의 DE-9IM(Dimensional Extended 9 Intersection Model) 표현을 검증합니다.

**관계 변이형**

두 도형 간의 관계의 DE-9IM(Dimensional Extended 9 Intersection Model) 표현을 반환합니다.

| | |
|---|---|
| 문법 | relate(geometry, geometry) |
| 인자 | • **geometry** - 도형<br>• **geometry** - 도형 |
| 예제 | • `relate( geom_from_wkt( 'LINESTRING(40 40,120 120)' ), geom_from_wkt( 'LINESTRING(40 40,60 120)' ) )` → 'FF1F00102' |

**패턴 매칭 변이형**

두 도형 간의 DE-9IM 관계가 지정된 패턴과 일치하는지 여부를 테스트합니다.

| | |
|---|---|
| 문법 | relate(geometry, geometry, pattern) |
| 인자 | • **geometry** - 도형<br>• **geometry** - 도형<br>• **pattern** - 매칭시킬 DE-9IM 패턴 |
| 예제 | • `relate( geom_from_wkt( 'LINESTRING(40 40,120 120)' ), geom_from_wkt( 'LINESTRING(40 40,60 120)' ), '**1F001**' )` → TRUE |

### reverse

Reverses the direction of a line string or reverses a string of text.

**문자열 변이형**

Reverses the order of characters in a string.

| | |
|---|---|
| 문법 | reverse(string) |
| 인자 | • **string** - string to reverse |
| 예제 | • `reverse('hello')` → 'olleh' |

**도형 변이형**

라인스트링의 꼭짓점 순서를 역전시켜 라인스트링의 방향을 반전시킵니다.

| 문법 | reverse(geometry) |
| --- | --- |
| 인자 | • **geometry** - 도형 |
| 예제 | • `geom_to_wkt(reverse(geom_from_wkt('LINESTRING(0 0, 1 1, 2 2)')))` → 'LINESTRING(2 2, 1 1, 0 0)' |

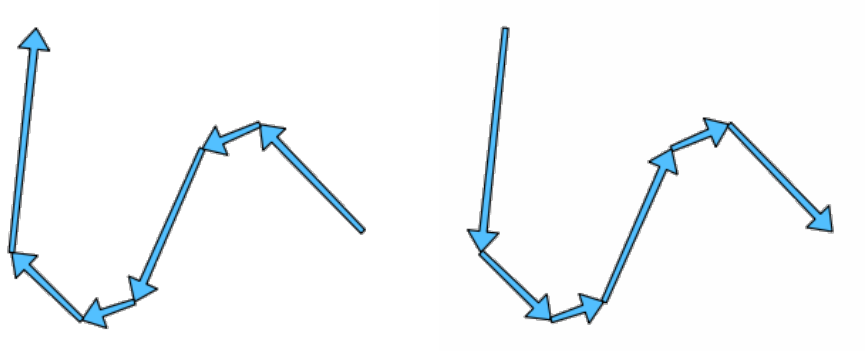


그림 9.27: 라인 방향 반전시키기

더 읽어볼 거리: 라인 방향 반전시키기 알고리즘

### rotate

도형의 기울인 (rotated) 버전을 반환합니다. 해당 도형의 공간 좌표계에서 계산합니다.

| 문법 | rotate(geometry, rotation, [center=NULL], [per_part=false])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| --- | --- |
| 인자 | • **geometry** - 도형<br>• **rotation** - 시계 방향 도 단위 회전<br>• **center** - 회전 중심 포인트. 이를 지정하지 않으면, 도형 경계 상자의 중심을 사용합니다.<br>• **per_part** - 부분 별로 기울기를 적용합니다. 참으로 설정하면, 입력 도형이 다중 부분 도형이고 기울기 중심 포인트를 명확하게 지정하지 않은 경우 각 부분의 경계 상자의 중심을 기준으로 기울기를 적용할 것입니다. |
| 예제 | • `rotate(@geometry, 45, make_point(4, 5))` → 도형을 (4, 5) 포인트 기준으로 시계 방향 45 도 회전시킵니다.<br>• `rotate(@geometry, 45)` → 도형을 도형 경계 상자의 중심을 기준으로 시계 방향 45 도 회전시킵니다. |

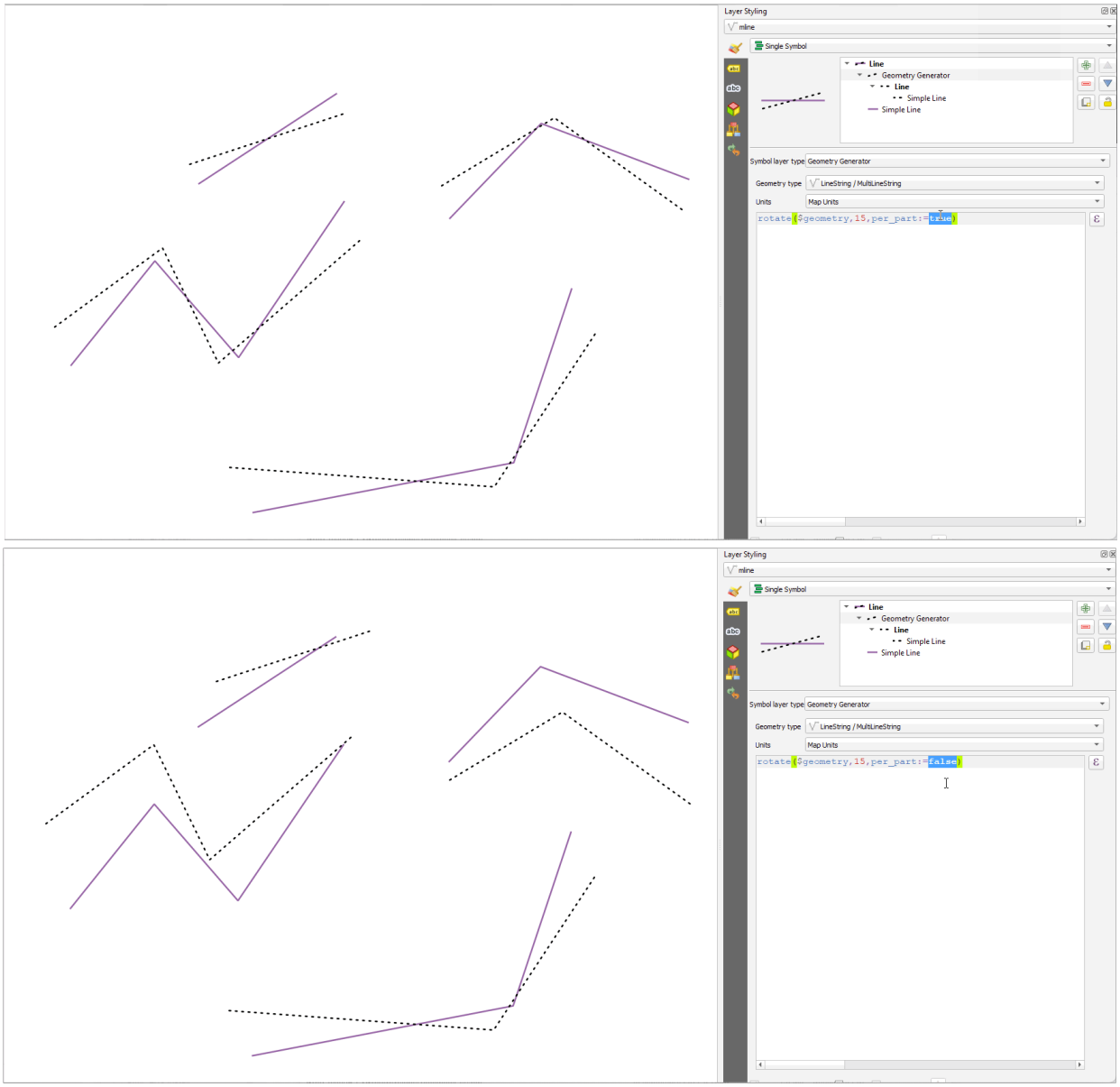



그림 9.28: 피처 기울이기

### roundness

Calculates how close a polygon shape is to a circle. The function returns 1 when the polygon shape is a perfect circle and 0 when it is completely flat.

| | |
|---|---|
| 문법 | roundness(geometry) |
| 인자 | • **geometry** - 폴리곤 |
| 예제 | • `round(roundness(geom_from_wkt('POLYGON(( 0 0, 0 1, 1 1, 1 0, 0 0))')), 3)` → 0.785<br>• `round(roundness(geom_from_wkt('POLYGON(( 0 0, 0 0.1, 1 0.1, 1 0, 0 0))')), 3)` → 0.260 |

더 읽어볼 거리: 둥글기 알고리즘

### scale

도형의 크기를 조정한 (scaled) 버전을 반환합니다. 해당 도형의 공간 좌표계에서 계산합니다.

| | |
|---|---|
| 문법 | scale(geometry, x_scale, y_scale, [center])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **geometry** - 도형<br>• **x_scale** - X 축 크기 조정 인자<br>• **y_scale** - Y 축 크기 조정 인자<br>• **center** - 크기 조정 중심 포인트. 이를 지정하지 않으면, 도형 경계 상자의 중심을 사용합니다. |
| 예제 | • `scale(@geometry, 2, 0.5, make_point(4, 5))` → 도형 크기를 (4, 5) 포인트 기준으로 수평으로 2 배, 수직으로 0.5 배 조정합니다.<br>• `scale(@geometry, 2, 0.5, make_point(4, 5))` → 도형 크기를 도형 경계 상자의 중심을 기준으로 수평으로 2 배, 수직으로 0.5 배 조정합니다. |

### segments_to_lines

입력 도형에 있는 모든 선분을 표현하는 라인으로 이루어진 멀티라인 도형을 반환합니다.

| | |
|---|---|
| 문법 | segments_to_lines(geometry) |
| 인자 | • **geometry** - 도형 객체 |
| 예제 | • `geom_to_wkt(segments_to_lines(geom_from_wkt('LINESTRING(0 0, 1 1, 2 2)')))` → 'MultiLineString ((0 0, 1 1),(1 1, 2 2))' |

더 읽어볼 거리: 라인 조각내기 알고리즘

### shared_paths

두 입력 도형이 공유하는 경로를 담고 있는 집합을 반환합니다. 같은 방향으로 진행되는 경로가 집합의 첫 번째 요소이고 반대 방향으로 진행되는 경로가 두 번째 요소입니다. 경로 자체는 첫 번째 도형의 방향으로 지정됩니다.

| 문법 | shared_paths(geometry1, geometry2) |
|---|---|
| 인자 | • **geometry1** - 라인스트링/멀티라인스트링 도형<br>• **geometry2** - 라인스트링/멀티라인스트링 도형 |
| 예제 | • `geom_to_wkt(shared_paths(geom_from_wkt('MULTILINESTRING((26 125,26 200,126 200,126 125,26 125),(51 150,101 150,76 175,51 150))'),geom_from_wkt('LINESTRING(151 100,126 156.25,126 125,90 161, 76 175)')))` → 'GeometryCollection (MultiLineString ((126 156.25, 126 125),(101 150, 90 161),(90 161, 76 175)),MultiLineString EMPTY)'<br>• `geom_to_wkt(shared_paths(geom_from_wkt('LINESTRING(76 175,90 161,126 125,126 156.25,151 100)'), geom_from_wkt('MULTILINESTRING((26 125,26 200,126 200,126 125,26 125),(51 150,101 150,76 175,51 150))')))` → 'GeometryCollection (MultiLineString EMPTY,MultiLineString ((76 175, 90 161),(90 161, 101 150),(126 125, 126 156.25)))' |

### shortest_line

도형 2 에 도형 1 을 결합하는 최단 라인을 반환합니다. 산출된 라인은 도형 1 에서 시작돼 도형 2 에서 끝납니다.

| 문법 | shortest_line(geometry1, geometry2) |
|---|---|
| 인자 | • **geometry1** - 최단 라인이 시작될 도형<br>• **geometry2** - 최단 라인이 끝날 도형 |
| 예제 | • `geom_to_wkt(shortest_line(geom_from_wkt('LINESTRING (20 80, 98 190, 110 180, 50 75 )'),geom_from_wkt('POINT(100 100)')))` → 'LineString(73.0769 115.384, 100 100)' |

### simplify

거리 기반 한계값을 (예를 들면 더글러스-패커 알고리즘을) 사용해서 노드를 제거해 도형을 단순화합니다. 이 알고리즘은 도형 내부의 큰 편차를 보전하고 거의 직선에 가까운 선분의 꼭짓점 개수를 줄입니다.

| | |
|---|---|
| 문법 | simplify(geometry, tolerance) |
| 인자 | • **geometry** - 도형<br>• **tolerance** - 제거할 포인트에 대한 직선 선분의 최대 편차 |
| 예제 | • `geom_to_wkt(simplify(geometry:=geom_from_wkt('LineString(0 0, 5 0.1, 10 0)'),tolerance:=5))` → 'LineString(0 0, 10 0)' |



그림 9.29: 왼쪽에서 오른쪽으로: 소스 레이어와 증가하는 단순화 허용 오차

더 읽어볼 거리: 단순화 알고리즘

### simplify_vw

면적 기반 한계값을 (예를 들면 비쉬왈링감-와이어트 알고리즘을) 사용해서 노드를 제거해 도형을 단순화합니다. 이 알고리즘은 도형에서 작은 면을 - 예를 들어 좁은 돌기 (narrow spike) 또는 거의 직선에 가까운 선분을 - 생성하는 꼭짓점을 제거합니다.

| | |
|---|---|
| 문법 | simplify_vw(geometry, tolerance) |
| 인자 | • **geometry** - 도형<br>• **tolerance** - 노드가 제거될 노드에 의해 생성된 최대 면적 측정 |
| 예제 | • `geom_to_wkt(simplify_vw(geometry:=geom_from_wkt('LineString(0 0, 5 0, 5.01 10, 5.02 0, 10 0)'),tolerance:=5))` → 'LineString(0 0, 10 0)' |

더 읽어볼 거리: 단순화 알고리즘

### single_sided_buffer

라인스트링 도형의 한쪽에만 버퍼를 적용해서 형성된 도형을 반환합니다. 거리 단위는 해당 도형의 공간 좌표계의 단위입니다.

| | |
|---|---|
| 문법 | single_sided_buffer(geometry, distance, [segments=8], [join=1], [miter_limit=2.0])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **geometry** - (멀티) 라인스트링 도형<br>• **distance** - 버퍼 거리. 양의 값이면 라인 좌측으로 버퍼를 적용하고, 음의 값이면 라인 우측으로 적용할 것입니다.<br>• **segments** - 둥근 결합 스타일을 사용한 경우 사분원을 나타내는 데 사용할 선분의 개수입니다. 숫자가 클수록 더 많은 노드를 가진 매끈한 버퍼를 생성합니다.<br>• **join** - 모서리 부분의 결합 스타일. 1 = 둥글게, 2 = 마이터, 그리고 3 = 비스듬하게<br>• **miter_limit** - 매우 뾰족한 모서리에 사용된 마이터 비율에 대한 제한 (마이터 결합만 사용하는 경우) |
| 예제 | • `single_sided_buffer(@geometry, 10.5)` → 라인 왼쪽으로 10.5 단위만큼 버퍼 생성<br>• `single_sided_buffer(@geometry, -10.5)` → 라인 오른쪽으로 10.5 단위만큼 버퍼 생성<br>• `single_sided_buffer(@geometry, 10.5, segments:=16, join:=1)` → 라인 왼쪽으로 10.5 단위만큼 버퍼 적용, 선분을 더 많이 사용할 수록 더 매끈한 버퍼를 생성합니다.<br>• `single_sided_buffer(@geometry, 10.5, join:=3)` → 라인 왼쪽으로 10.5 단위만큼 버퍼 적용, 'bevel'결합 사용 |



그림 9.30: 동일 라인 벡터 레이어의 왼쪽 대 오른쪽 버퍼

더 읽어볼 거리: 한쪽 버퍼 생성하기 알고리즘

### sinuosity

만곡 도형의 굽이 (sinuosity) 를 반환합니다. 굽이란 만곡 도형의 길이와 그 두 종단점의 (2 차원) 직선 거리의 비율입니다.

| | |
|---|---|
| 문법 | sinuosity(geometry) |
| 인자 | • **geometry** - 입력 만곡 도형 (원호스트링, 라인스트링) |
| 예제 | • `round(sinuosity(geom_from_wkt('LINESTRING(2 0, 2 2, 3 2, 3 3)')), 3)` → 1.265<br>• `sinuosity(geom_from_wkt('LINESTRING( 3 1, 5 1)'))` → 1.0 |

### smooth

도형의 모서리를 둥글게 다듬는 노드를 추가하여 도형을 매끄럽게 만듭니다. 입력 도형에 Z 또는 M 값이 포함된 경우 이 값들도 평활화하고, 산출 도형은 입력 도형과 동일한 차원을 유지할 것입니다.

| | |
|---|---|
| 문법 | smooth(geometry, [iterations=1], [offset=0.25], [min_length=-1], [max_angle=180])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **geometry** - 도형<br>• **iterations** - 반복해서 적용할 평활화 횟수입니다. 숫자가 클수록 더 매끄럽지만 더 복잡한 도형을 생성합니다.<br>• **offset** - 평활화된 도형이 얼마나 엄격하게 원본 도형을 따를지 제어하는 0 에서 0.5 사이의 값입니다. 값이 적을수록 더 평활화되고, 값이 클수록 덜 평활화됩니다.<br>• **min_length** - 평활화 작업에 적용할 최소 선분 길이입니다. 도형의 짧은 선분에 추가 노드를 과도하게 배치하는 일을 피하려면 이 파라미터를 사용하면 됩니다.<br>• **max_angle** - 노드에서 평활화 작업에 적용할 최대 각도 (0-180) 입니다. 최대 각도를 줄이면 도형에서 일부러 뾰족하게 만든 모서리를 보전할 수 있습니다. 예를 들어, 값을 80 도로 지정하면 도형에 있는 직각을 유지할 것입니다. |
| 예제 | • `geom_to_wkt(smooth(geometry:=geom_from_wkt('LineString(0 0, 5 0, 5 5)'),iterations:=1,offset:=0.2,min_length:=-1, max_angle:=180))` → 'LineString (0 0, 4 0, 5 1, 5 5)' |



그림 9.31: 반복 횟수를 늘릴수록 더 매끄러워지는 도형

더 읽어볼 거리: 평탄화 알고리즘

### square_wave

도형의 경계선을 따라 정사각/직사각 파형을 구성합니다.

| | |
|---|---|
| 문법 | square_wave(geometry, wavelength, amplitude, [strict=False])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **geometry** - 도형<br>• **wavelength** - 정사각 파형의 파장<br>• **amplitude** - 정사각 파형의 진폭<br>• **strict** - 파장 인자는 기본적으로"최장 파장"으로 취급되는데, 이때 도형의 경계선을 따라 정확한 개수의 정사각 파형을 생성하도록 실제 파장은 동적으로 조정될 것입니다. `strict` 인자를 참으로 설정하는 경우 파장을 변경 없이 사용하기 때문에 최종 파형에 쓰이는 패턴이 완전하지 않을 수도 있습니다. |
| 예제 | • `square_wave(geom_from_wkt('LineString(0 0, 10 0)'), 3, 1)` → 라인스트링을 따라 파장이 3 이고 진폭이 1 인 정사각 파형을 생성 |

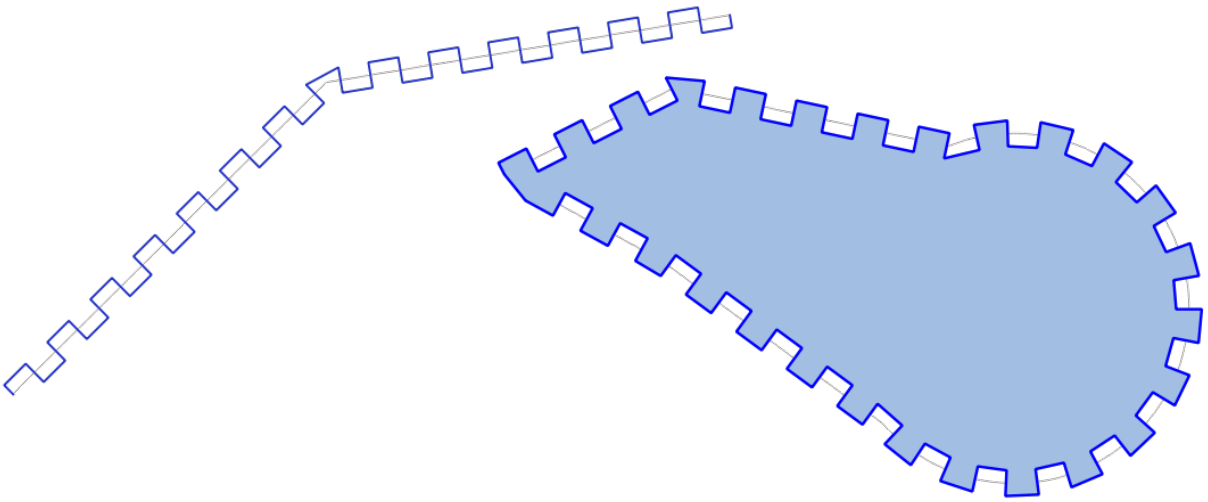


그림 9.32: 피처에 정사각 파형 심볼 적용하기

### square_wave_randomized

도형의 경계선을 따라 임의의 정사각/직사각 파형을 구성합니다.

| | |
|---|---|
| 문법 | square_wave_randomized(geometry, min_wavelength, max_wavelength, min_amplitude, max_amplitude, [seed=0])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **geometry** - 도형<br>• **min_wavelength** - 파형의 최단 파장<br>• **max_wavelength** - 파형의 최장 파장<br>• **min_amplitude** - 파형의 최저 진폭<br>• **max_amplitude** - 파형의 최고 진폭<br>• **seed** - 파형 생성을 위한 임의 뿌리값 (random seed) 을 지정합니다. 뿌리값이 0 이면 완전히 임의의 파형 집합을 생성할 것입니다. |
| 예제 | • `square_wave_randomized(geom_from_wkt('LineString(0 0, 10 0)'), 2, 3, 0.1, 0.2)` → 라인스트링을 따라 파장이 2 에서 3 사이이고 진폭이 0.1 에서 0.2 사이의 임의 크기의 정사각 파형을 생성 |

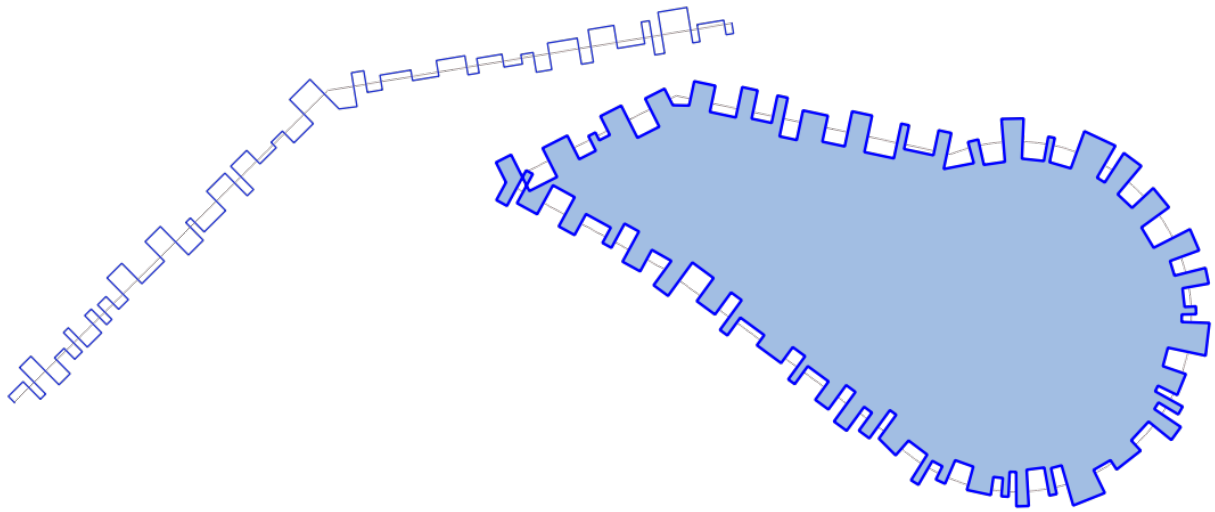


그림 9.33: 피처에 임의의 정사각 파형 심볼 적용하기

### start_point

도형에서 첫 번째 노드를 반환합니다.

| | |
|---|---|
| 문법 | start_point(geometry) |
| 인자 | • **geometry** - 도형 객체 |
| 예제 | • `geom_to_wkt(start_point(geom_from_wkt('LINESTRING(4 0, 4 2, 0 2)')))` → 'Point (4 0)' |

더 읽어볼 거리: *end_point*, 특정 꼭짓점 추출하기 알고리즘



그림 9.34: 라인 피처의 시작 포인트

### straight_distance_2d

도형의 첫 번째와 마지막 꼭짓점 사이의 직교/유클리드 거리를 반환합니다. 만곡 도형 (원호스트링, 라인스트링)이어야 합니다.

| | |
|---|---|
| 문법 | straight_distance_2d(geometry) |
| 인자 | • **geometry** - 도형 |
| 예제 | • `straight_distance_2d(geom_from_wkt('LINESTRING(1 0, 1 1)'))` → 1<br>• `round(straight_distance_2d(geom_from_wkt('LINESTRING(1 4, 3 5, 5 0)')), 3)` → 5.657 |

더 읽어볼 거리: *length*

### sym_difference

두 도형이 교차하지 않는 부분들을 표현하는 도형을 반환합니다.

| | |
|---|---|
| 문법 | sym_difference(geometry1, geometry2) |
| 인자 | • **geometry1** - 도형<br>• **geometry2** - 도형 |
| 예제 | • `geom_to_wkt( sym_difference( geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ), geom_from_wkt( 'LINESTRING(3 3, 8 8)' ) ) )` → 'LINESTRING(5 5, 8 8)' |

더 읽어볼 거리: 대칭 차감 알고리즘

### tapered_buffer

라인 도형을 따라 버퍼의 지름이 라인의 길이에 걸쳐 균등하게 달라지는 버퍼를 생성합니다.

| | |
|---|---|
| 문법 | tapered_buffer(geometry, start_width, end_width, [segments=8])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **geometry** - 입력 도형. 반드시 (멀티) 라인 도형이어야만 합니다.<br>• **start_width** - 라인 시작점에서의 버퍼 너비<br>• **end_width** - 라인 종단점에서의 버퍼 너비<br>• **segments** - 버퍼에서 사분면의 원호에 포함되는 선분 수 |
| 예제 | • `tapered_buffer(geometry:=geom_from_wkt('LINESTRING(1 2, 4 2)'),start_width:=1,end_width:=2,segments:=8)` → 반경이 1 에서 시작하여 라인스트링 도형을 따라 반경이 2 가 되는 테이퍼형 (tapered) 버퍼 |

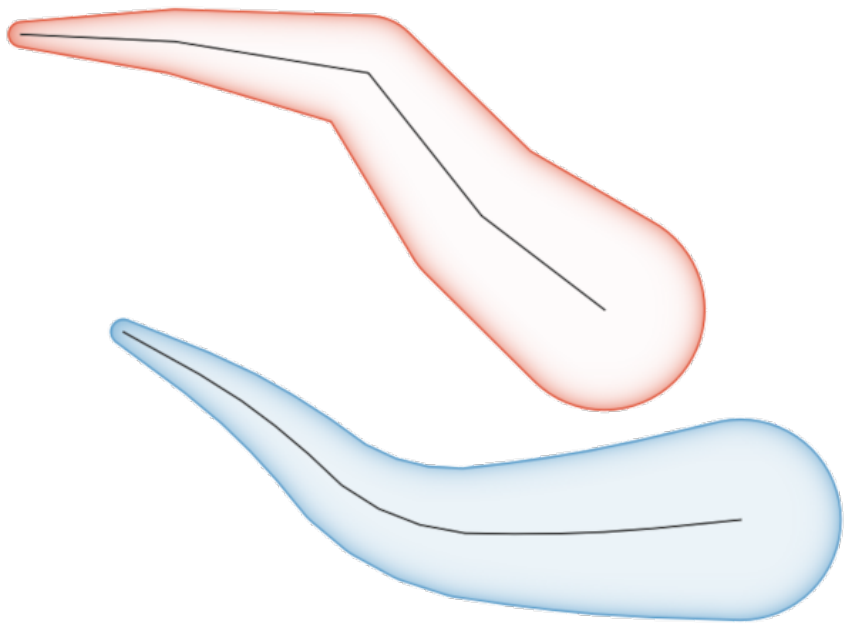


그림 9.35: 라인 피처의 테이퍼형 버퍼

더 읽어볼 거리: 줄어드는 버퍼 생성하기 알고리즘

### touches

도형이 다른 도형과 접하는지 검증합니다. 도형들이 최소한 포인트 1 개를 공유하지만 각 도형의 내부가 교차하지 않는 경우 참을 반환합니다.

| | |
|---|---|
| 문법 | touches(geometry1, geometry2) |
| 인자 | • **geometry1** - 도형<br>• **geometry2** - 도형 |
| 예제 | • `touches( geom_from_wkt( 'LINESTRING(5 3, 4 4)' ), geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ) )` → TRUE<br>• `touches( geom_from_wkt( 'POINT(4 4)' ), geom_from_wkt( 'POINT(5 5)' ) )` → FALSE |

더 읽어볼 거리: *overlay_touches*

### transform

원본 좌표계에서 대상 좌표계로 변환한 도형을 반환합니다.

| | |
|---|---|
| 문법 | transform(geometry, source_auth_id, dest_auth_id) |
| 인자 | • **geometry** - 도형<br>• **source_auth_id** - the source CRS definition or CRS object<br>• **dest_auth_id** - the destination CRS definition or CRS object |
| 예제 | • `geom_to_wkt( transform( make_point(488995.53240249, 7104473.38600835), 'EPSG:2154', 'EPSG:4326' ) )` → 'POINT(0 51)' |

더 읽어볼 거리: 레이어 재투영하기 알고리즘

### translate

도형의 이동 (translated) 버전을 반환합니다. 해당 도형의 공간 좌표계 상에서 계산합니다.

| | |
|---|---|
| 문법 | translate(geometry, dx, dy) |
| 인자 | • **geometry** - 도형<br>• **dx** - Δx<br>• **dy** - Δy |
| 예제 | • `translate(@geometry, 5, 10)` → 원래 도형과 동일한 유형의 도형 |

더 읽어볼 거리: 이동시키기 *(Translate)* 알고리즘

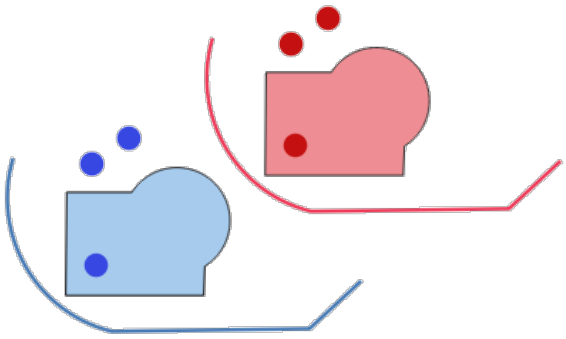


그림 9.36: 도형 번역하기

### triangular_wave

도형의 경계선을 따라 삼각 파형을 구성합니다.

| | |
|---|---|
| 문법 | triangular_wave(geometry, wavelength, amplitude, [strict=False])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **geometry** - 도형<br>• **wavelength** - 삼각 파형의 파장<br>• **amplitude** - 삼각 파형의 진폭<br>• **strict** - 파장 인자는 기본적으로"최장 파장"으로 취급되는데, 이때 도형의 경계선을 따라 정확한 개수의 삼각 파형을 생성하도록 실제 파장은 동적으로 조정될 것입니다. `strict` 인자를 참으로 설정하는 경우 파장을 변경 없이 사용하기 때문에 최종 파형에 쓰이는 패턴이 완전하지 않을 수도 있습니다. |
| 예제 | • `triangular_wave(geom_from_wkt('LineString(0 0, 10 0)'), 3, 1)` → 라인스트링을 따라 파장이 3 이고 진폭이 1 인 삼각 파형을 생성 |

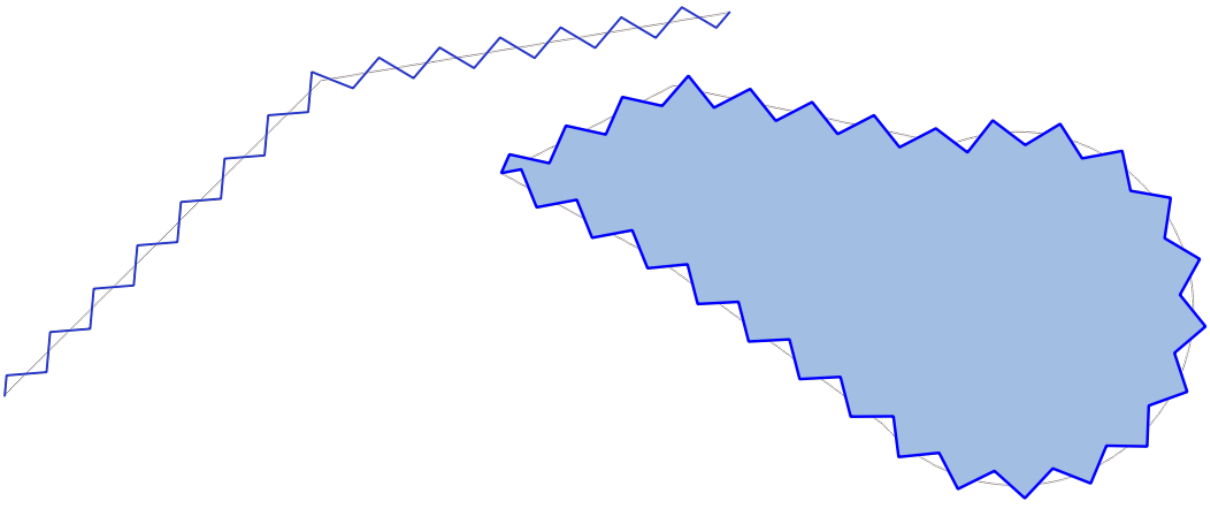


그림 9.37: 피처에 삼각 파형 심볼 적용하기

### triangular_wave_randomized

도형의 경계선을 따라 임의의 삼각 파형을 구성합니다.

| | |
|---|---|
| 문법 | triangular_wave_randomized(geometry, min_wavelength, max_wavelength, min_amplitude, max_amplitude, [seed=0])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **geometry** - 도형<br>• **min_wavelength** - 파형의 최단 파장<br>• **max_wavelength** - 파형의 최장 파장<br>• **min_amplitude** - 파형의 최저 진폭<br>• **max_amplitude** - 파형의 최고 진폭<br>• **seed** - 파형 생성을 위한 임의 뿌리값 (random seed) 을 지정합니다. 뿌리값이 0 이면 완전히 임의의 파형 집합을 생성할 것입니다. |
| 예제 | • `triangular_wave_randomized(geom_from_wkt('LineString(0 0, 10 0)'), 2, 3, 0.1, 0.2)` → 라인스트링을 따라 파장이 2 에서 3 사이이고 진폭이 0.1 에서 0.2 사이의 임의 크기의 삼각 파형을 생성 |

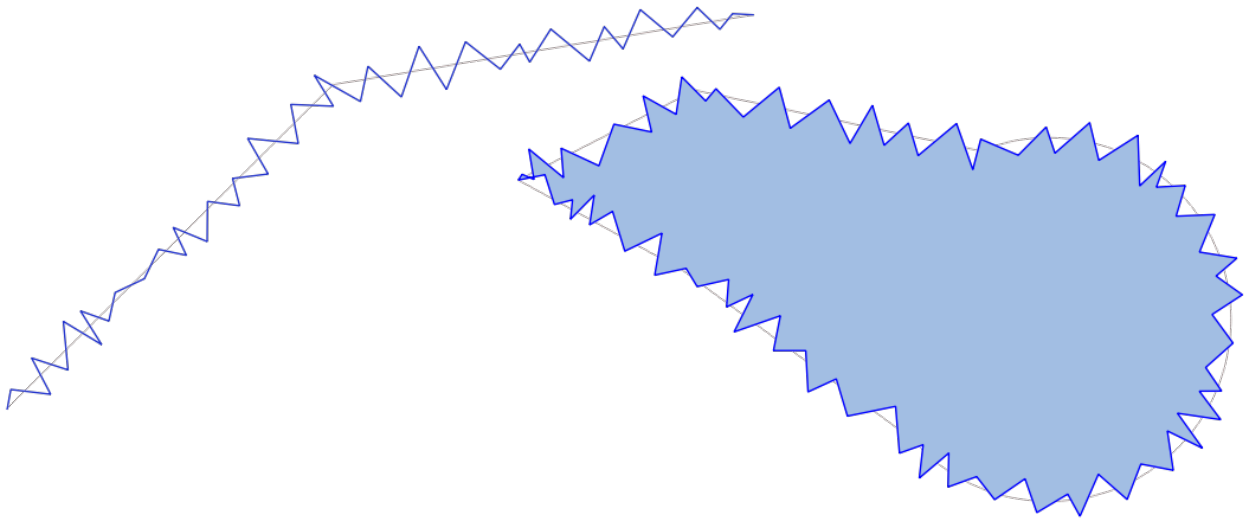


그림 9.38: 피처에 임의의 삼각 파형 심볼 적용하기

### union

도형들의 모든 포인트를 통합한 집합을 표현하는 도형을 반환합니다.

| | |
|---|---|
| 문법 | union(geometry1, geometry2) |
| 인자 | • **geometry1** - 도형<br>• **geometry2** - 도형 |
| 예제 | • `geom_to_wkt( union( make_point(4, 4), make_point(5, 5) ) )` → 'MULTIPOINT(4 4, 5 5)' |

### wave

도형의 경계선을 따라 둥근 (사인 곡선 같은) 파형을 구성합니다.

| | |
|---|---|
| 문법 | wave(geometry, wavelength, amplitude, [strict=False])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **geometry** - 도형<br>• **wavelength** - 사인 곡선 같은 파형의 파장<br>• **amplitude** - 사인 곡선 같은 파형의 진폭<br>• **strict** - 파장 인자는 기본적으로"최장 파장"으로 취급되는데, 이때 도형의 경계선을 따라 정확한 개수의 파형을 생성하도록 실제 파장은 동적으로 조정될 것입니다. `strict` 인자를 참으로 설정하는 경우 파장을 변경 없이 사용하기 때문에 최종 파형에 쓰이는 패턴이 완전하지 않을 수도 있습니다. |
| 예제 | • `wave(geom_from_wkt('LineString(0 0, 10 0)'), 3, 1)` → 라인스트링을 따라 파장이 3 이고 진폭이 1 인 사인 곡선 같은 파형을 생성 |

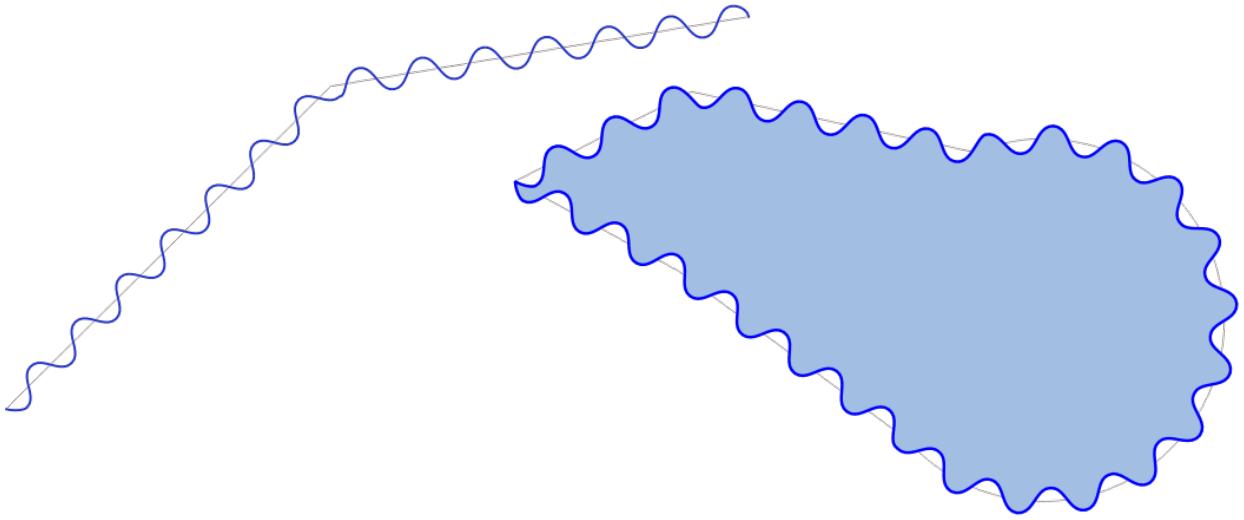


그림 9.39: 피처에 둥근 파형 심볼 적용하기

### wave_randomized

도형의 경계선을 따라 임의의 만곡 (사인 곡선 같은) 파형을 구성합니다.

| | |
|---|---|
| 문법 | wave_randomized(geometry, min_wavelength, max_wavelength, min_amplitude, max_amplitude, [seed=0])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **geometry** - 도형<br>• **min_wavelength** - 파형의 최단 파장<br>• **max_wavelength** - 파형의 최장 파장<br>• **min_amplitude** - 파형의 최저 진폭<br>• **max_amplitude** - 파형의 최고 진폭<br>• **seed** - 파형 생성을 위한 임의 뿌리값 (random seed) 을 지정합니다. 뿌리값이 0 이면 완전히 임의의 파형 집합을 생성할 것입니다. |
| 예제 | • `wave_randomized(geom_from_wkt('LineString(0 0, 10 0)'), 2, 3, 0.1, 0.2)` → 라인스트링을 따라 파장이 2 에서 3 사이이고 진폭이 0.1 에서 0.2 사이의 임의 크기의 만곡 파형을 생성 |

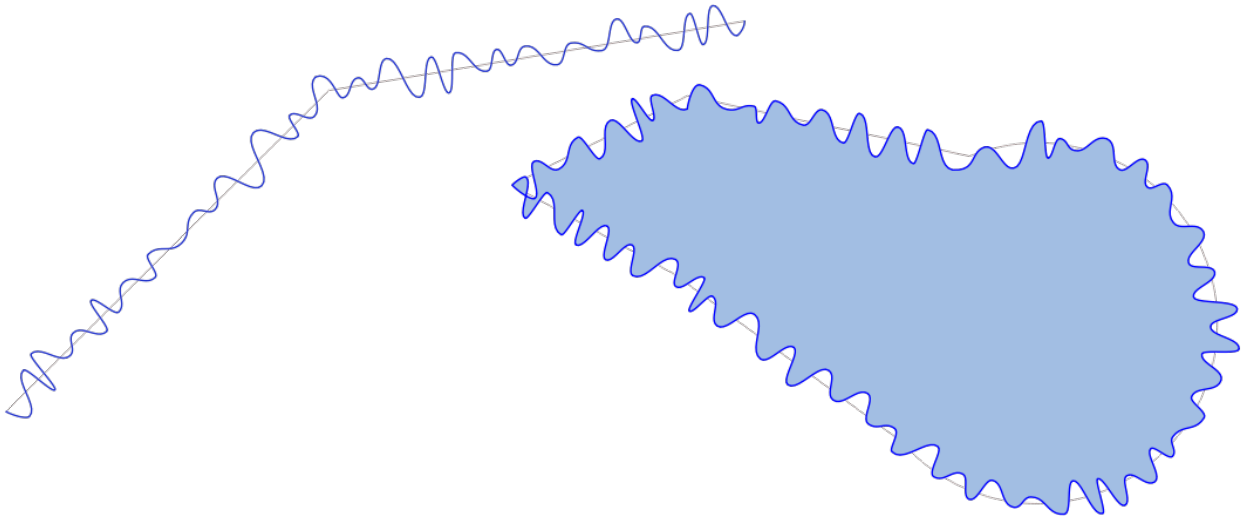


그림 9.40: 피처에 임의의 둥근 파형 심볼 적용하기

### wedge_buffer

포인트 도형에서 나온 쐐기 모양의 버퍼를 반환합니다.

| | |
|---|---|
| 문법 | wedge_buffer(center, azimuth, width, outer_radius, [inner_radius=0.0])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **center** - 버퍼의 중심 포인트 (원점). 포인트 도형이어야만 합니다.<br>• **azimuth** - 쐐기의 중선에서 포인트까지의 (도 단위) 각도<br>• **width** - 버퍼의 (도 단위) 너비. 쐐기는 방위각 방향의 양쪽으로 각도 너비의 반씩 확장될 것입니다.<br>• **outer_radius** - 버퍼 외부 반경<br>• **inner_radius** - 선택적인 버퍼 내부 반경 |
| 예제 | • `wedge_buffer(center:=geom_from_wkt('POINT(1 2)'), azimuth:=90,width:=180,outer_radius:=1)` → 포인트 (1,2) 를 중심으로 동쪽을 향한 너비 180 도, 외부 반경 1 인 쐐기 모양 버퍼 |

더 읽어볼 거리: 쐐기 버퍼 생성하기 알고리즘

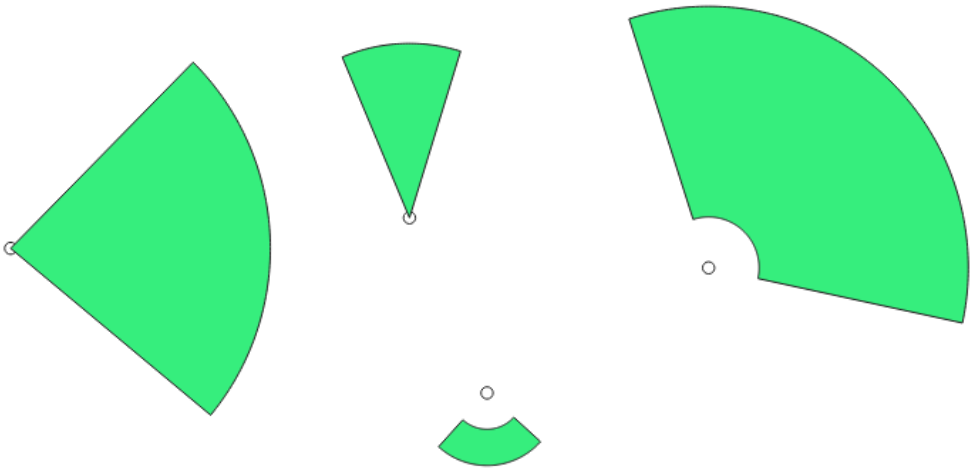


그림 9.41: 쐐기형 버퍼를 적용한 피처들

### within

도형이 다른 도형 내부에 있는지 검증합니다. 도형 1 이 도형 2 내부에 완전히 들어가 있는 경우 참을 반환합니다.

| | |
|---|---|
| 문법 | within(geometry1, geometry2) |
| 인자 | • **geometry1** - 도형<br>• **geometry2** - 도형 |
| 예제 | • `within( geom_from_wkt( 'POINT( 0.5 0.5)' ), geom_from_wkt( 'POLYGON((0 0, 0 1, 1 1, 1 0, 0 0))' ) )` → TRUE<br>• `within( geom_from_wkt( 'POINT( 5 5 )' ), geom_from_wkt( 'POLYGON((0 0, 0 1, 1 1, 1 0, 0 0 ))' ) )` → FALSE |

더 읽어볼 거리: *overlay_within*

### $x

현재 포인트 피처의 X 좌표를 반환합니다. 피처가 멀티포인트 피처인 경우, 첫 번째 포인트의 X 좌표를 반환할 것입니다. **경고: 이 함수는 중요도가 떨어져 더 이상 사용되지 않고 앞으로는 사라지게 될 것입니다. 이 함수를 대체하는 x() 함수를 @geometry 변수와 함께 사용할 것을 권장합니다.**

| | |
|---|---|
| 문법 | $x |
| 예제 | • `$x` → 42 |

더 읽어볼 거리: *x*

#### x

포인트 도형의 X 좌표를, 또는 포인트가 아닌 도형의 경우 중심점 (centroid) 의 X 좌표를 반환합니다.

| | |
|---|---|
| 문법 | x(geometry) |
| 인자 | • **geometry** - 도형 |
| 예제 | • `x( geom_from_wkt( 'POINT(2 5)' ) )` → 2<br>• `x( @geometry )` → 현재 피처 중심점의 X 좌표 |

### $x_at

현재 피처 도형의 X 좌표를 반환합니다. **경고: 이 함수는 중요도가 떨어져 더 이상 사용되지 않고 앞으로는 사라지게 될 것입니다. 이 함수를 대체하는 x_at 함수를 @geometry 변수와 함께 사용할 것을 권장합니다.**

| | |
|---|---|
| 문법 | $x_at(vertex) |
| 인자 | • **vertex** - 현재 도형의 꼭짓점의 인덱스 (0 부터 인덱스 시작; 음의 값은 -1 부터 시작하며 마지막 인덱스부터 적용) |
| 예제 | • `$x_at(1)` → 5 |

더 읽어볼 거리: *x_at*

### x_at

도형의 X 좌표를 반환합니다.

| | |
|---|---|
| 문법 | x_at(geometry, vertex) |
| 인자 | • **geometry** - 도형 객체<br>• **vertex** - 도형의 꼭짓점의 인덱스 (0 부터 인덱스 시작; 음의 값은 -1 부터 시작하며 마지막 인덱스부터 적용) |
| 예제 | • `x_at( geom_from_wkt( 'POINT(4 5)' ), 0 )` → 4 |

### x_max

도형의 최대 X 좌표를 반환합니다. 해당 도형의 공간 좌표계 상에서 좌표를 계산합니다.

| | |
|---|---|
| 문법 | x_max(geometry) |
| 인자 | • **geometry** - 도형 |
| 예제 | • `x_max( geom_from_wkt( 'LINESTRING(2 5, 3 6, 4 8)') )` → 4 |

### x_min

도형의 최소 X 좌표를 반환합니다. 해당 도형의 공간 좌표계 상에서 좌표를 계산합니다.

| | |
|---|---|
| 문법 | x_min(geometry) |
| 인자 | • **geometry** - 도형 |
| 예제 | • `x_min( geom_from_wkt( 'LINESTRING(2 5, 3 6, 4 8)') )` → 2 |

### $y

현재 포인트 피처의 Y 좌표를 반환합니다. 피처가 멀티포인트 피처인 경우, 첫 번째 포인트의 Y 좌표를 반환할 것입니다. **경고: 이 함수는 중요도가 떨어져 더 이상 사용되지 않고 앞으로는 사라지게 될 것입니다. 이 함수를 대체하는 y() 함수를 @geometry 변수와 함께 사용할 것을 권장합니다.**

| | |
|---|---|
| 문법 | $y |
| 예제 | • `$y` → 42 |

더 읽어볼 거리: *y*

### y

포인트 도형의 Y 좌표를, 또는 포인트가 아닌 도형의 경우 중심점 (centroid) 의 Y 좌표를 반환합니다.

| | |
|---|---|
| 문법 | y(geometry) |
| 인자 | • **geometry** - 도형 |
| 예제 | • `y( geom_from_wkt( 'POINT(2 5)' ) )` → 5<br>• `y( @geometry )` → 현재 피처 중심점의 Y 좌표 |

### $y_at

현재 피처 도형의 Y 좌표를 반환합니다. **경고: 이 함수는 중요도가 떨어져 더 이상 사용되지 않고 앞으로는 사라지게 될 것입니다. 이 함수를 대체하는 y_at 함수를 @geometry 변수와 함께 사용할 것을 권장합니다.**

| | |
|---|---|
| 문법 | $y_at(vertex) |
| 인자 | • **vertex** - 현재 도형의 꼭짓점의 인덱스 (0 부터 인덱스 시작; 음의 값은 -1 부터 시작하며 마지막 인덱스부터 적용) |
| 예제 | • `$y_at(1)` → 2 |

더 읽어볼 거리: *y_at*

### y_at

도형의 Y 좌표를 반환합니다.

| | |
|---|---|
| 문법 | y_at(geometry, vertex) |
| 인자 | • **geometry** - 도형 객체<br>• **vertex** - 도형의 꼭짓점의 인덱스 (0 부터 인덱스 시작; 음의 값은 -1 부터 시작하며 마지막 인덱스부터 적용) |
| 예제 | • `y_at( geom_from_wkt( 'POINT(4 5)' ), 0 )` → 5 |

### y_max

도형의 최대 Y 좌표를 반환합니다. 해당 도형의 공간 좌표계 상에서 좌표를 계산합니다.

| | |
|---|---|
| 문법 | y_max(geometry) |
| 인자 | • **geometry** - 도형 |
| 예제 | • `y_max( geom_from_wkt( 'LINESTRING(2 5, 3 6, 4 8)') )` → 8 |

### y_min

도형의 최소 Y 좌표를 반환합니다. 해당 도형의 공간 좌표계 상에서 좌표를 계산합니다.

| | |
|---|---|
| 문법 | y_min(geometry) |
| 인자 | • **geometry** - 도형 |
| 예제 | • `y_min( geom_from_wkt( 'LINESTRING(2 5, 3 6, 4 8)') )` → 5 |

### $z

현재 포인트 피처가 3 차원일 경우 포인트의 Z 값을 반환합니다. 피처가 멀티포인트 피처인 경우, 첫 번째 포인트의 Z 값을 반환할 것입니다. **경고: 이 함수는 중요도가 떨어져 더 이상 사용되지 않고 앞으로는 사라지게 될 것입니다. 이 함수를 대체하는 z() 함수를 @geometry 변수와 함께 사용할 것을 권장합니다.**

| | |
|---|---|
| 문법 | $z |
| 예제 | • `$z` → 123 |

### z

포인트 도형의 Z 좌표를 반환하거나, 도형에 Z 값이 없는 경우 NULL 을 반환합니다.

| | |
|---|---|
| 문법 | z(geometry) |
| 인자 | • **geometry** - 포인트 도형 |
| 예제 | • `z( geom_from_wkt( 'POINTZ(2 5 7)' ) )` → 7 |

### z_at

도형의 Z 좌표를 반환하거나, 도형에 Z 값이 없는 경우 NULL 을 반환합니다.

| | |
|---|---|
| 문법 | z_at(geometry, vertex) |
| 인자 | • **geometry** - 도형 객체<br>• **vertex** - 도형의 꼭짓점의 인덱스 (0 부터 인덱스 시작; 음의 값은 -1 부터 시작하며 마지막 인덱스부터 적용) |
| 예제 | • `z_at(geom_from_wkt('LineStringZ(0 0 0, 10 10 5, 10 10 0)'), 1)` → 5 |

#### z_max

도형의 최대 Z 좌표를 반환하거나, 도형에 Z 값이 없는 경우 NULL 을 반환합니다.

| | |
|---|---|
| 문법 | z_max(geometry) |
| 인자 | • **geometry** - Z 좌표를 가진 도형 |
| 예제 | • `z_max( geom_from_wkt( 'POINT ( 0 0 1 )' ) )` → 1<br>• `z_max( geom_from_wkt( 'MULTIPOINT ( 0 0 1 , 1 1 3 )' ) )` → 3<br>• `z_max( make_line( make_point( 0,0,0 ), make_point( -1,-1,-2 ) ) )` → 0<br>• `z_max( geom_from_wkt( 'LINESTRING( 0 0 0, 1 0 2, 1 1 -1 )' ) )` → 2<br>• `z_max( geom_from_wkt( 'POINT ( 0 0 )' ) )` → NULL |

#### z_min

도형의 최소 Z 좌표를 반환하거나, 도형에 Z 값이 없는 경우 NULL 을 반환합니다.

| | |
|---|---|
| 문법 | z_min(geometry) |
| 인자 | • **geometry** - Z 좌표를 가진 도형 |
| 예제 | • `z_min( geom_from_wkt( 'POINT ( 0 0 1 )' ) )` → 1<br>• `z_min( geom_from_wkt( 'MULTIPOINT ( 0 0 1 , 1 1 3 )' ) )` → 1<br>• `z_min( make_line( make_point( 0,0,0 ), make_point( -1,-1,-2 ) ) )` → -2<br>• `z_min( geom_from_wkt( 'LINESTRING( 0 0 0, 1 0 2, 1 1 -1 )' ) )` → -1<br>• `z_min( geom_from_wkt( 'POINT ( 0 0 )' ) )` → NULL |

### 9.2.15 조판 함수

이 그룹은 인쇄 조판기 항목 속성을 처리하는 함수를 담고 있습니다.

#### item_variables

해당 인쇄 조판기 내부에 있는 조판기 항목에서 나온 변수들의 맵을 반환합니다.

| 문법 | item_variables(id) |
| --- | --- |
| 인자 | • **id** - 조판 항목 ID |
| 예제 | • `map_get( item_variables('Map 0'), 'map_scale')` → 현재 인쇄 조판기에 있는'Map 0'항목의 축척 |

더 읽어볼 거리: 기본변수 목록

#### map_credits

조판, 또는 특정 조판 맵 항목에 표시되는 레이어들에 대한 크레디트 (사용 권한) 문자열 목록을 반환합니다.

| 문법 | map_credits([id], [include_layer_names=false], [layer_name_separator=': '])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| --- | --- |
| 인자 | • **id** - 맵 항목 ID 입니다. 지정하지 않는 경우 조판에 있는 모든 맵의 레이어들을 사용할 것입니다.<br>• **include_layer_names** - 이 인자를 참으로 설정하면 레이어 크레디트 문자열 앞에 레이어 이름을 포함시킵니다.<br>• **layer_name_separator** - include_layer_names 인자가 참인 경우 레이어 이름과 레이어 크레디트 문자열 사이에 삽입할 문자열입니다. |
| 예제 | • `array_to_string( map_credits() )` → 조판에 있는 모든 맵 항목에 표시된 모든 레이어들에 대한, 'CC-BY-NC, CC-BY-SA'같은 레이어 크레디트를 쉼표로 구분한 목록<br>• `array_to_string( map_credits( 'Main Map' ) )` → 'Main Map' 조판 항목에 표시된 레이어들에 대한, 'CC-BY-NC, CC-BY-SA'같은 레이어 크레디트를 쉼표로 구분한 목록<br>• `array_to_string( map_credits( 'Main Map', include_layer_names := true, layer_name_separator := ': ' ) )` → 'Main Map'조판 항목에 표시된 레이어들에 대한, 'Railway lines: CC-BY-NC, Basemap: CC-BY-SA'처럼 레이어 이름과 레이어 크레디트를 쉼표로 구분한 목록 |

이 함수를 사용하려면 레이어의메타데이터 속성 접근 권한이 필요합니다.

### 9.2.16 맵 레이어

이 그룹은 현재 프로젝트에서 사용할 수 있는 레이어 목록과, 각 레이어 별로 사용할 수 있는 (데이터셋에 저장된 필드, 가상 또는 보조 필드는 물론 결합으로 생성된) 필드를 담고 있습니다. 필드 및 값 에서 설명한 것과 동일한 방법으로 이런 필드를 쌍방향으로 작업할 수 있지만, 필드가 활성 레이어에 속하지 않은 경우 더블클릭하면 표현식에 이름을 필드 참조로서가 아니라 (작은 따옴표로 싸인) 문자열로 추가할 것입니다. 이 습성으로 인해, 예를 들어집계, 속성 또는공간 쿼리 수행 시, 서로 다른 레이어를 참조하는 표현식을 더 편리하게 작성할 수 있습니다.

또 레이어를 처리하기 위한 몇몇 편리한 함수들도 제공합니다.

### decode_uri

레이어를 가져와서 기저 데이터 제공자의 URI 를 디코딩합니다. 데이터 제공자 유형에 따라 어떤 데이터를 사용할 수 있는지가 달라집니다.

| | |
|---|---|
| 문법 | decode_uri(layer, [part])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **layer** - URI 를 디코딩해야 할 레이어입니다.<br>• **part** - 반환할 URI 부분. 지정하지 않으면 모든 URI 부분을 가진 맵을 반환할 것입니다. |
| 예제 | • `decode_uri(@layer)` → {'layerId': '0', 'layerName': '', 'path': '/home/qgis/shapefile.shp'}<br>• `decode_uri(@layer)` → {'layerId': NULL, 'layerName': 'layer', 'path': '/home/qgis/geopackage.gpkg'}<br>• `decode_uri(@layer, 'path')` → 'C:\my_data\qgis\shape.shp' |

### layer_property

일치하는 레이어 속성 또는 메타데이터 값을 반환합니다.

| 문법 | layer_property(layer, property) |
|---|---|
| 인자 | • **layer** - 레이어 이름 또는 레이어 ID 가운데 하나를 나타내는 문자열<br>• **property** - 반환할 속성에 대응하는 문자열입니다. 사용할 수 있는 옵션은 다음과 같습니다:<br>– name: 레이어 이름<br>– id: 레이어 ID<br>– title: 메타데이터 제목 문자열<br>– abstract: 메타데이터 요약 문자열<br>– keywords: 메타데이터 키워드<br>– data_url: 메타데이터 URL<br>– attribution: 메타데이터 속성 문자열<br>– attribution_url: 메타데이터 속성 URL<br>– source: 레이어 소스<br>– min_scale: 레이어 가시화 최저 축척<br>– max_scale: 레이어 가시화 최고 축척<br>– is_editable: 레이어 편집 모드가 켜져 있는지 여부<br>– crs: 레이어 좌표계<br>– crs_definition: 레이어 좌표계 전체 정의<br>– crs_description: 레이어 좌표계 설명<br>– crs_ellipsoid: 레이어 좌표계 타원체의 머리글자<br>– extent: 레이어 범위 (도형 객체로서)<br>– distance_units: 레이어 거리 단위<br>– type: 레이어 유형, 예: 벡터 또는 래스터<br>– storage_type: 저장소 포맷 (벡터 레이어 전용)<br>– geometry_type: 도형 유형, 예: 포인트 (벡터 레이어 전용)<br>– feature_count: 레이어에 있는 피처의 대략적인 개수 (벡터 레이어 전용)<br>– path: 레이어 데이터소스를 가리키는 파일 경로. 파일 기반 레이어에만 사용할 수 있습니다. |
| 예제 | • `layer_property('streets','title')` → 'Basemap Streets'<br>• `layer_property('airports','feature_count')` → 120<br>• `layer_property('landsat','crs')` → 'EPSG:4326' |

더 읽어볼 거리: 벡터, 래스터 및메시 레이어 속성

### load_layer

레이어를 소스 URI 와 제공자 이름으로 불러옵니다.

| 문법 | load_layer(uri, provider) |
|---|---|
| 인자 | • **uri** - 레이어 소스 URI 문자열<br>• **provider** - 레이어 데이터 제공자 이름 |
| 예제 | • `layer_property(load_layer('c:/data/roads.shp', 'ogr'), 'feature_count')` → c:/data/roads.shp 벡터 레이어의 피처 개수 |

### 9.2.17 맵 함수

이 그룹은 (딕셔너리 객체, 키-값 쌍, 또는 연관 배열 (associative array) 이라고도 하는) 맵 데이터 구조의 키와 값을 생성하고 처리하는 함수들을 담고 있습니다. 값들의 순서가 중요한목록 데이터 구조 와는 달리, 맵 객체에 있는 키-값 쌍은 중요하지 않고 값을 값의 키로 식별합니다.

#### from_json

JSON 서식 문자열을 불러옵니다.

| | |
|---|---|
| 문법 | from_json(string) |
| 인자 | • **string** - JSON 문자열 |
| 예제 | • `from_json('{"1":"one","2":"two"}')` → { '1': 'one', '2': 'two'}<br>• `from_json('[1,2,3]')` → [1,2,3] |

#### hstore_to_map

HStore 서식 문자열로부터 맵을 생성합니다.

| | |
|---|---|
| 문법 | hstore_to_map(string) |
| 인자 | • **string** - 입력 문자열 |
| 예제 | • `hstore_to_map('qgis=>rocks')` → { 'qgis': 'rocks'} |

#### map

파라미터 쌍으로 전달된 모든 키와 값을 담고 있는 맵을 반환합니다.

| | |
|---|---|
| 문법 | map(key1, value1, key2, value2, ⋯) |
| 인자 | • **key** - 키 (문자열)<br>• **value** - 값 |
| 예제 | • `map('1','one','2', 'two')` → { '1': 'one', '2': 'two'}<br>• `map('1','one','2', 'two')['1']` → 'one' |

### map_akeys

맵의 모든 키를 배열로 반환합니다.

| | |
|---|---|
| 문법 | map_akeys(map) |
| 인자 | • **map** - 맵 |
| 예제 | • `map_akeys(map('1','one','2','two'))` → [ '1', '2'] |

### map_avals

맵의 모든 값을 배열로 반환합니다.

| | |
|---|---|
| 문법 | map_avals(map) |
| 인자 | • **map** - 맵 |
| 예제 | • `map_avals(map('1','one','2','two'))` → [ 'one', 'two'] |

### map_concat

지정한 맵의 모든 항목들을 담고 있는 맵을 반환합니다. 두 맵이 동일한 키를 담고 있는 경우, 두 번째 맵의 값을 취합니다.

| | |
|---|---|
| 문법 | map_concat(map1, map2, …) |
| 인자 | • **map** - 맵 |
| 예제 | • `map_concat(map('1','one', '2','overridden'),map('2','two', '3','three'))` → { '1': 'one', '2': 'two', '3': 'three'} |

### map_delete

지정한 키와 그에 대응하는 값을 삭제한 맵을 반환합니다.

| | |
|---|---|
| 문법 | map_delete(map, key) |
| 인자 | • **map** - 맵<br>• **key** - 삭제할 키 |
| 예제 | • `map_delete(map('1','one','2','two'),'2')` → { '1': 'one'} |

### map_exist

지정한 키가 맵에 실재하는 경우 참을 반환합니다.

| | |
|---|---|
| 문법 | map_exist(map, key) |
| 인자 | • **map** - 맵<br>• **key** - 검색할 키 |
| 예제 | • `map_exist(map('1','one','2','two'),'3')` → FALSE |

### map_get

지정한 키에 대응하는 맵의 값을 반환합니다. 키가 실재하지 않는 경우 NULL 을 반환합니다.

| | |
|---|---|
| 문법 | map_get(map, key) |
| 인자 | • **map** - 맵<br>• **key** - 검색할 키 |
| 예제 | • `map_get(map('1','one','2','two'),'2')` → 'two'<br>• `map_get( item_variables('Map 0'), 'map_scale')` → 현재 인쇄 조판의 ('Map 0'항목이 있는 경우) 'Map 0'항목의 축척 |

**힌트**

맵에서 값을 가져오기 위해인덱스 연산자 *([])* 를 사용할 수도 있습니다.

### map_insert

추가한 키/값을 가진 맵을 반환합니다. 키가 이미 존재하는 경우, 그 값을 무시합니다.

| | |
|---|---|
| 문법 | map_insert(map, key, value) |
| 인자 | • **map** - 맵<br>• **key** - 추가할 키<br>• **value** - 추가할 값 |
| 예제 | • `map_insert(map('1','one'),'3','three')` → { '1': 'one', '3': 'three' }<br>• `map_insert(map('1','one','2','overridden'),'2','two')` → { '1': 'one', '2': 'two' } |

### map_prefix_keys

모든 키 앞에 지정한 문자열의 접두어를 붙인 맵을 반환합니다.

| | |
|---|---|
| 문법 | map_prefix_keys(map, prefix) |
| 인자 | • **map** - 맵<br>• **prefix** - 문자열 |
| 예제 | • `map_prefix_keys(map('1','one','2','two'), 'prefix-')` → { 'prefix-1': 'one', 'prefix-2': 'two'} |

### map_to_hstore

맵 요소들을 HStore 서식 문자열로 병합합니다.

| | |
|---|---|
| 문법 | map_to_hstore(map) |
| 인자 | • **map** - 입력 맵 |
| 예제 | • `map_to_hstore(map('1','one','2','two'))` → "'1"=>"one"',"'2"=>"two"' |

### map_to_html_dl

맵 요소들을 HTML 정의 목록 문자열로 병합합니다.

| | |
|---|---|
| 문법 | map_to_html_dl(map) |
| 인자 | • **map** - 입력 맵 |
| 예제 | • `map_to_html_dl(map('1','one','2','two'))` → <dl><dt>1</dt><dd>one</dd><dt>2</dt><dd>two</dd></dl> |

### map_to_html_table

맵 요소들을 HTML 테이블 문자열로 병합합니다.

| | |
|---|---|
| 문법 | map_to_html_table(map) |
| 인자 | • **map** - 입력 맵 |
| 예제 | • `map_to_html_table(map('1','one','2','two'))` → <table><thead><tr><th>1</th><th>2</th></tr></thead><tbody><tr><td>one</td><td>two</td></tr></tbody>< |

#### to_json

맵, 배열, 또는 다른 값으로부터 JSON 서식 문자열을 생성합니다.

| | |
|---|---|
| 문법 | to_json(value) |
| 인자 | • **value** - 입력 값 |
| 예제 | • `to_json(map('1','one','2','two'))` → {"1":"one","2":"two"}<br>• `to_json(array(1,2,3))` → [1,2,3] |

#### url_encode

맵으로부터 URL 이 인코딩된 문자열을 반환합니다. 모든 문자를 적절히 인코딩된 양식으로 변형시켜 SQL 서식을 완전하게 준수하는 쿼리 문자열을 생성합니다.

'+'더하기 기호는 변환되지 않는다는 사실을 기억하십시오.

| | |
|---|---|
| 문법 | url_encode(map) |
| 인자 | • **map** - 맵 |
| 예제 | • `url_encode(map('a&+b', 'a and plus b', 'a=b', 'a equals b'))` → 'a%26+b=a%20and%20plus%20b&a%3Db=a%20equals%20b' |

### 9.2.18 수학 함수

이 그룹은 (제곱근, 삼각함수 등의) 수학 함수를 담고 있습니다.

#### abs

숫자의 절댓값을 반환합니다.

| | |
|---|---|
| 문법 | abs(value) |
| 인자 | • **value** - 숫자 |
| 예제 | • `abs(-2)` → 2 |

### acos

값의 시컨트 (역 코사인) 를 라디안 단위로 반환합니다.

| | |
|---|---|
| 문법 | acos(value) |
| 인자 | • **value** - 라디안 단위 각도의 코사인 |
| 예제 | • `acos(0.5)` → 1.0471975511966 |

### asin

값의 코시컨트 (역 사인) 를 라디안 단위로 반환합니다.

| | |
|---|---|
| 문법 | asin(value) |
| 인자 | • **value** - 라디안 단위 각도의 사인 |
| 예제 | • `asin(1.0)` → 1.5707963267949 |

### atan

값의 코탄젠트 (역 탄젠트) 를 라디안 단위로 반환합니다.

| | |
|---|---|
| 문법 | atan(value) |
| 인자 | • **value** - 라디안 단위 각도의 탄젠트 |
| 예제 | • `atan(0.5)` → 0.463647609000806 |

### atan2

결과물의 사분면을 결정하기 위해 두 인자의 사인 값을 이용해서 dy/dx 값의 코탄젠트 (역 탄젠트) 를 반환합니다.

| | |
|---|---|
| 문법 | atan2(dy, dx) |
| 인자 | • **dy** - Y 좌표 차 (difference)<br>• **dx** - X 좌표 차 (difference) |
| 예제 | • `atan2(1.0, 1.732)` → 0.523611477769969 |

### ceil

숫자를 올림합니다.

| | |
|---|---|
| 문법 | ceil(value) |
| 인자 | • **value** - 숫자 |
| 예제 | • `ceil(4.9)` → 5<br>• `ceil(-4.9)` → -4 |

### clamp

입력값을 지정한 범위로 제한합니다.

| | |
|---|---|
| 문법 | clamp(minimum, input, maximum) |
| 인자 | • **minimum** - 입력 받을 수 있는 최소값<br>• **input** - 최소 및 최대 로 지정된 범위로 제한될 값<br>• **maximum** - 입력 받을 수 있는 최대값 |
| 예제 | • `clamp(1,5,10)` → 5<br>*input* 이 1 과 10 사이이기 때문에, 그대로 반환됩니다.<br>• `clamp(1,0,10)` → 1<br>*input* 이 최소값 1 미만이기 때문에, 함수가 1 을 반환합니다<br>• `clamp(1,11,10)` → 10<br>*input* 이 최대값 10 을 초과하기 때문에, 함수가 10 을 반환합니다 |

### cos

각도의 코사인을 반환합니다.

| | |
|---|---|
| 문법 | cos(angle) |
| 인자 | • **angle** - 라디안 단위 각도 |
| 예제 | • `cos(1.571)` → 0.000796326710733263 |

### degrees

라디안 단위를 도 단위로 변환합니다.

| | |
|---|---|
| 문법 | degrees(radians) |
| 인자 | • **radians** - 숫자 값 |
| 예제 | • `degrees(3.14159)` → 180<br>• `degrees(1)` → 57.2958 |

### exp

값의 지수 (exponential) 를 반환합니다.

| | |
|---|---|
| 문법 | exp(value) |
| 인자 | • **value** - 지수 (exponential) 를 반환할 숫자 |
| 예제 | • `exp(1.0)` → 2.71828182845905 |

### floor

숫자를 내림합니다.

| | |
|---|---|
| 문법 | floor(value) |
| 인자 | • **value** - 숫자 |
| 예제 | • `floor(4.9)` → 4<br>• `floor(-4.9)` → -5 |

### ln

값의 자연로그를 반환합니다.

| | |
|---|---|
| 문법 | ln(value) |
| 인자 | • **value** - 숫자 값 |
| 예제 | • `ln(1)` → 0<br>• `ln(2.7182818284590452354)` → 1 |

### log

전달된 값 및 밑 (base) 의 로그 값을 반환합니다.

| | |
|---|---|
| 문법 | log(base, value) |
| 인자 | • **base** - 양수<br>• **value** - 양수 |
| 예제 | • `log(2, 32)` → 5<br>• `log(0.5, 32)` → -5 |

### log10

전달된 표현식의 상용로그 (밑이 10 인 로그) 값을 반환합니다.

| | |
|---|---|
| 문법 | log10(value) |
| 인자 | • **value** - 양수 |
| 예제 | • `log10(1)` → 0<br>• `log10(100)` → 2 |

### max

값의 집합에서 가장 큰 값을 반환합니다.

| | |
|---|---|
| 문법 | max(value1, value2, ⋯) |
| 인자 | • **value** - 숫자 |
| 예제 | • `max(2,10.2,5.5)` → 10.2<br>• `max(20.5,NULL,6.2)` → 20.5 |

### min

값의 집합에서 가장 작은 값을 반환합니다.

| | |
|---|---|
| 문법 | min(value1, value2, ⋯) |
| 인자 | • **value** - 숫자 |
| 예제 | • `min(20.5,10,6.2)` → 6.2<br>• `min(2,-10.3,NULL)` → -10.3 |

### pi

계산을 위한 파이 (π) 값을 반환합니다.

| | |
|---|---|
| 문법 | pi() |
| 예제 | • `pi()` → 3.14159265358979 |

### radians

도 단위를 라디안 단위로 변환합니다.

| | |
|---|---|
| 문법 | radians(degrees) |
| 인자 | • **degrees** - 숫자 값 |
| 예제 | • `radians(180)` → 3.14159<br>• `radians(57.2958)` → 1 |

### rand

최소 및 최대 인자가 지정하는 범위 안에서 임의의 정수를 반환합니다. (최소값, 최대값 포함) 시드를 지정한 경우, 시드에 따라 항상 동일한 값을 반환할 것입니다.

| | |
|---|---|
| 문법 | rand(min, max, [seed=NULL])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **min** - 원하는 가장 작은 난수를 나타내는 정수<br>• **max** - 원하는 가장 큰 난수를 나타내는 정수<br>• **seed** - 시드로 사용할 값 |
| 예제 | • `rand(1, 10)` → 8 |

### randf

최소 및 최대 인자가 지정하는 범위 안에서 임의의 부동소수점형 실수를 반환합니다. (최소값, 최대값 포함) 시드를 지정한 경우, 시드에 따라 항상 동일한 값을 반환할 것입니다.

| | |
|---|---|
| 문법 | randf([min=0.0], [max=1.0], [seed=NULL])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **min** - 원하는 가장 작은 난수를 나타내는 실수<br>• **max** - 원하는 가장 큰 난수를 나타내는 실수<br>• **seed** - 시드로 사용할 값 |
| 예제 | • `randf(1, 10)` → 4.59258286403147 |

### round

숫자를 소수점 이하 자릿수로 반올림합니다.

| | |
|---|---|
| 문법 | round(value, [places=0])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **value** - 반올림할 실수<br>• **places** - 소수점 이하 자리를 반올림할 자릿수를 나타내는 선택적인 정수입니다. 음수값도 가능합니다. |
| 예제 | • `round(1234.567, 2)` → 1234.57<br>• `round(1234.567)` → 1235<br>• `round(1234.567, -1)` → 1230 |

### scale_exponential

지정한 값을 지수 곡선 (exponential curve) 을 이용해서 입력 범위에서 출력 범위로 변형합니다. 이 함수를 사용하면 지정한 산출 범위로 값을 부드럽게 조정 (ease in or ease out) 할 수 있습니다.

| | |
|---|---|
| 문법 | scale_exponential(value, domain_min, domain_max, range_min, range_max, exponent) |
| 인자 | • **value** - 입력 범위에 있는 값. 이 함수는 산출 범위에 대응하는 조정된 값을 반환할 것입니다.<br>• **domain_min** - 입력 값이 취해야 할 가장 작은 값인 입력 범위의 최소값을 지정합니다.<br>• **domain_max** - 입력 값이 취해야 할 가장 큰 값인 입력 범위의 최대값을 지정합니다.<br>• **range_min** - 함수가 산출해야 할 가장 작은 값인 산출 범위의 최소값을 지정합니다.<br>• **range_max** - 함수가 산출해야 할 가장 큰 값인 산출 범위의 최대값을 지정합니다.<br>• **exponent** - 입력값이 산출 범위에 매핑되는 방식을 지정하는 (0 보다 큰) 양의 값입니다. 큰 지수는 입력 값이 범위 최대값에 도달할 때 가속하기 전에 천천히 시작하여 산출값을 부드럽게 조정 (ease in) 되도록 만듭니다. 지수가 작으면 (1 미만) 산출값이 부드럽게 조정 (ease out) 되고 매핑이 빠르게 시작되지만 범위 최대 값에 도달하면 속도가 느려집니다. |
| 예제 | • `scale_exponential(5,0,10,0,100,2)` → 3.030<br>지수 2 를 사용해서 부드럽게 조정 (ease in)<br>• `scale_exponential(3,0,10,0,100,0.5)` → 87.585<br>지수 0.5 를 사용해서 부드럽게 조정 (ease out) |

### scale_linear

지정한 값을 선형 보간 (linear interpolation) 을 이용해서 입력 범위에서 출력 범위로 변형합니다.

| | |
|---|---|
| 문법 | scale_linear(value, domain_min, domain_max, range_min, range_max) |
| 인자 | • **value** - 입력 범위에 있는 값. 이 함수는 산출 범위에 대응하는 조정된 값을 반환할 것입니다.<br>• **domain_min** - 입력 값이 취해야 할 가장 작은 값인 입력 범위의 최소값을 지정합니다.<br>• **domain_max** - 입력 값이 취해야 할 가장 큰 값인 입력 범위의 최대값을 지정합니다.<br>• **range_min** - 함수가 산출해야 할 가장 작은 값인 산출 범위의 최소값을 지정합니다.<br>• **range_max** - 함수가 산출해야 할 가장 큰 값인 산출 범위의 최대값을 지정합니다. |
| 예제 | • `scale_linear(5,0,10,0,100)` → 50<br>• `scale_linear(0.2,0,1,0,360)` → 72<br>0 에서 1 사이의 값을 0 에서 360 사이의 각도로 조정<br>• `scale_linear(1500,1000,10000,9,20)` → 9.6111111<br>1,000 명에서 10,000 명 사이에서 변화하는 인구를 글꼴 크기 9 에서 20 사이로 조정 |

### scale_polynomial

지정한 값을 다항식 곡선 (polynomial curve) 을 이용해서 입력 범위에서 출력 범위로 변형합니다. 이 함수를 사용하면 지정한 산출 범위로 값을 부드럽게 조정 (ease in or ease out) 할 수 있습니다.

| | |
|---|---|
| 문법 | scale_polynomial(value, domain_min, domain_max, range_min, range_max, exponent) |
| 인자 | • **value** - 입력 범위에 있는 값. 이 함수는 산출 범위에 대응하는 조정된 값을 반환할 것입니다.<br>• **domain_min** - 입력 값이 취해야 할 가장 작은 값인 입력 범위의 최소값을 지정합니다.<br>• **domain_max** - 입력 값이 취해야 할 가장 큰 값인 입력 범위의 최대값을 지정합니다.<br>• **range_min** - 함수가 산출해야 할 가장 작은 값인 산출 범위의 최소값을 지정합니다.<br>• **range_max** - 함수가 산출해야 할 가장 큰 값인 산출 범위의 최대값을 지정합니다.<br>• **exponent** - 입력값이 산출 범위에 매핑되는 방식을 지정하는 (0 보다 큰) 양의 값입니다. 큰 지수는 입력 값이 범위 최대값에 도달할 때 가속하기 전에 천천히 시작하여 산출값을 부드럽게 조정 (ease in) 되도록 만듭니다. 지수가 작으면 (1 미만) 산출값이 부드럽게 조정 (ease out) 되고 매핑이 빠르게 시작되지만 범위 최대 값에 도달하면 속도가 느려집니다. |
| 예제 | • `scale_polynomial(5,0,10,0,100,2)` → 25<br>지수 2 를 사용해서 부드럽게 조정 (ease in)<br>• `scale_polynomial(3,0,10,0,100,0.5)` → 54.772<br>지수 0.5 를 사용해서 부드럽게 조정 (ease out) |

### sin

각도의 사인을 반환합니다.

| | |
|---|---|
| 문법 | sin(angle) |
| 인자 | • **angle** - 라디안 단위 각도 |
| 예제 | • `sin(1.571)` → 0.999999682931835 |

### sqrt

값의 제곱근을 반환합니다.

| | |
|---|---|
| 문법 | sqrt(value) |
| 인자 | • **value** - 숫자 |
| 예제 | • `sqrt(9)` → 3 |

#### tan

각도의 탄젠트를 반환합니다.

| | |
|---|---|
| 문법 | tan(angle) |
| 인자 | • **angle** - 라디안 단위 각도 |
| 예제 | • `tan(1.0)` → 1.5574077246549 |

### 9.2.19 메시 함수

이 그룹은 메시 관련 값들을 계산하거나 반환하는 함수를 담고 있습니다.

#### $face_area

현재 메시 면 (mesh face) 의 면적을 반환합니다. 이 함수는 현재 프로젝트의 타원체 설정과 면적 단위 설정을 따라 면적을 계산합니다. 예를 들어 프로젝트에 타원체를 설정했다면 타원체 기반으로 면적을 계산하고, 타원체를 설정하지 않았다면 평면 상에서 면적을 계산합니다.

| | |
|---|---|
| 문법 | $face_area |
| 예제 | • `$face_area` → 42 |

#### $face_index

현재 메시 면의 인덱스를 반환합니다.

| | |
|---|---|
| 문법 | $face_index |
| 예제 | • `$face_index` → 4581 |

#### $vertex_as_point

현재 꼭짓점을 포인트 도형으로 반환합니다.

| | |
|---|---|
| 문법 | $vertex_as_point |
| 예제 | • `geom_to_wkt( $vertex_as_point )` → 'POINT(800 1500 41)' |

#### $vertex_index

현재 메시 꼭짓점의 인덱스를 반환합니다.

| | |
|---|---|
| 문법 | $vertex_index |
| 예제 | • `$vertex_index` → 9874 |

#### $vertex_x

현재 메시 꼭짓점의 X 좌표를 반환합니다.

| | |
|---|---|
| 문법 | $vertex_x |
| 예제 | • `$vertex_x` → 42.12 |

#### $vertex_y

현재 메시 꼭짓점의 Y 좌표를 반환합니다.

| | |
|---|---|
| 문법 | $vertex_y |
| 예제 | • `$vertex_y` → 12.24 |

#### $vertex_z

현재 메시 꼭짓점의 Z 값을 반환합니다.

| | |
|---|---|
| 문법 | $vertex_z |
| 예제 | • `$vertex_z` → 42 |

### 9.2.20 연산자

이 그룹은 +, -, * 같은 연산자를 담고 있습니다. 다음에 설명하는 수학 함수들 대부분이, 입력 값 가운데 하나가 NULL 인 경우 NULL 을 반환한다는 점을 유념하십시오.

### %

나눗셈의 나머지입니다. 나눔수 (dividend) 의 부호를 가집니다.

| | |
|---|---|
| 문법 | a % b |
| 인자 | • **a** - 값<br>• **b** - 값 |
| 예제 | • `9 % 2` → 1<br>• `9 % -2` → 1<br>• `-9 % 2` → -1<br>• `5 % NULL` → NULL |

### *

두 값을 곱한 값을 반환합니다.

| | |
|---|---|
| 문법 | a * b |
| 인자 | • **a** - 값<br>• **b** - 값 |
| 예제 | • `5 * 4` → 20<br>• `5 * NULL` → NULL |

### +

두 값을 더한 값을 반환합니다. 두 값 가운데 하나가 NULL 인 경우 NULL 을 산출할 것입니다.

| | |
|---|---|
| 문법 | a + b |
| 인자 | • **a** - 값<br>• **b** - 값 |
| 예제 | • `5 + 4` → 9<br>• `5 + NULL` → NULL<br>• `'QGIS ' + 'ROCKS'` → 'QGIS ROCKS'<br>• `to_datetime('2020-08-01 12:00:00') + '1 day 2 hours'` → 2020-08-02T14:00:00 |

더 읽어볼 거리: *concat*, ||

### -

한 값에서 다른 값을 뺀 값을 반환합니다. 두 값 가운데 하나가 NULL 인 경우 NULL 을 산출할 것입니다.

| | |
|---|---|
| 문법 | a - b |
| 인자 | • **a** - 값<br>• **b** - 값 |
| 예제 | • `5 - 4` → 1<br>• `5 - NULL` → NULL<br>• `to_datetime('2012-05-05 12:00:00') - to_interval('1 day 2 hours')` → 2012-05-04T10:00:00 |

### /

한 값을 다른 값으로 나눈 값을 반환합니다.

| | |
|---|---|
| 문법 | a / b |
| 인자 | • **a** - 값<br>• **b** - 값 |
| 예제 | • `5 / 4` → 1.25<br>• `5 / NULL` → NULL |

### //

한 값을 다른 값으로 나눈 값에서 소수점 자리를 버린 값을 반환합니다.

| | |
|---|---|
| 문법 | a // b |
| 인자 | • **a** - 값<br>• **b** - 값 |
| 예제 | • `9 // 2` → 4 |

### <

두 값을 비교해서 왼쪽 값이 오른쪽 값 미만인 경우 1 로 평가합니다.

| | |
|---|---|
| 문법 | a < b |
| 인자 | • **a** - 값<br>• **b** - 값 |
| 예제 | • `5 < 4` → FALSE<br>• `5 < 5` → FALSE<br>• `4 < 5` → TRUE |

### <=

두 값을 비교해서 왼쪽 값이 오른쪽 값 이하인 경우 1 로 평가합니다.

| | |
|---|---|
| 문법 | a <= b |
| 인자 | • **a** - 값<br>• **b** - 값 |
| 예제 | • `5 <= 4` → FALSE<br>• `5 <= 5` → TRUE<br>• `4 <= 5` → TRUE |

### <>

두 값을 비교해서 두 값이 동등하지 않은 경우 1 로 평가합니다.

| | |
|---|---|
| 문법 | a <> b |
| 인자 | • **a** - 값<br>• **b** - 값 |
| 예제 | • `5 <> 4` → TRUE<br>• `4 <> 4` → FALSE<br>• `5 <> NULL` → NULL<br>• `NULL <> NULL` → NULL |

#### =

두 값을 비교해서 두 값이 동등한 경우 1 로 평가합니다.

| | |
|---|---|
| 문법 | a = b |
| 인자 | • **a** - 값<br>• **b** - 값 |
| 예제 | • `5 = 4` → FALSE<br>• `4 = 4` → TRUE<br>• `5 = NULL` → NULL<br>• `NULL = NULL` → NULL |

#### >

두 값을 비교해서 왼쪽 값이 오른쪽 값을 초과하는 경우 1 로 평가합니다.

| | |
|---|---|
| 문법 | a > b |
| 인자 | • **a** - 값<br>• **b** - 값 |
| 예제 | • `5 > 4` → TRUE<br>• `5 > 5` → FALSE<br>• `4 > 5` → FALSE |

#### >=

두 값을 비교해서 왼쪽 값이 오른쪽 값 이상인 경우 1 로 평가합니다.

| | |
|---|---|
| 문법 | a >= b |
| 인자 | • **a** - 값<br>• **b** - 값 |
| 예제 | • `5 >= 4` → TRUE<br>• `5 >= 5` → TRUE<br>• `4 >= 5` → FALSE |

### AND

조건 a 와 조건 b 가 참이면 참을 반환합니다.

| | |
|---|---|
| 문법 | a AND b |
| 인자 | • **a** - 조건<br>• **b** - 조건 |
| 예제 | • `TRUE AND TRUE` → TRUE<br>• `TRUE AND FALSE` → FALSE<br>• `4 = 2+2 AND 1 = 1` → TRUE<br>• `4 = 2+2 AND 1 = 2` → FALSE |

### BETWEEN

값이 지정한 범위 내에 있는 경우 참을 반환합니다. 이 범위는 경계 값을 포함한다고 간주합니다. 범위 외에 있는지를 테스트하려면 NOT BETWEEN 함수를 사용하면 됩니다.

| | |
|---|---|
| 문법 | value BETWEEN lower_bound AND higher_bound |
| 인자 | • **value** - 범위와 비교할 값. 문자열, 숫자, 또는 날짜일 수 있습니다.<br>• **lower_bound AND higher_bound** - 범위의 경계 값들 |
| 예제 | • `'B' BETWEEN 'A' AND 'C'` → TRUE<br>• `2 BETWEEN 1 AND 3` → TRUE<br>• `2 BETWEEN 2 AND 3` → TRUE<br>• `'B' BETWEEN 'a' AND 'c'` → FALSE<br>• `lower('B') BETWEEN 'a' AND 'b'` → TRUE |

**참고**

`value BETWEEN lower_bound AND higher_bound` 는 `value >= lower_bound AND value <= higher_bound` 와 동일합니다.

더 읽어볼 거리: *NOT BETWEEN*

### ILIKE

첫 번째 파라미터가 대소문자를 구분하지 않는 지정 패턴과 일치하는 경우 참을 반환합니다. 대소문자를 구분해도 일치하는지를 보려면 ILIKE 대신 LIKE 를 사용하면 됩니다. 숫자값도 입력할 수 있습니다.

| | |
|---|---|
| 문법 | string/number ILIKE pattern |
| 인자 | • **string/number** - 검색할 문자열<br>• **pattern** - 비교할 패턴. 와일드카드 문자로'%', 단일 와일드카드 문자로'_'를 사용할 수 있으며, 이런 특수 문자를 이스케이프시키려면'\\'를 사용하면 됩니다. |
| 예제 | • `'A' ILIKE 'A'` → TRUE<br>• `'A' ILIKE 'a'` → TRUE<br>• `'A' ILIKE 'B'` → FALSE<br>• `'ABC' ILIKE 'b'` → FALSE<br>• `'ABC' ILIKE 'B'` → FALSE<br>• `'ABC' ILIKE '_b_'` → TRUE<br>• `'ABC' ILIKE '_B_'` → TRUE<br>• `'ABCD' ILIKE '_b_'` → FALSE<br>• `'ABCD' ILIKE '_B_'` → FALSE<br>• `'ABCD' ILIKE '_b%'` → TRUE<br>• `'ABCD' ILIKE '_B%'` → TRUE<br>• `'ABCD' ILIKE '%b%'` → TRUE<br>• `'ABCD' ILIKE '%B%'` → TRUE<br>• `'ABCD%' ILIKE 'abcd\\%'` → TRUE<br>• `'ABCD' ILIKE '%B\\%'` → FALSE |

### IN

값들의 목록에서 값을 찾은 경우 참을 반환합니다.

| | |
|---|---|
| 문법 | a IN b |
| 인자 | • **a** - 값<br>• **b** - 값들의 목록 |
| 예제 | • `'A' IN ('A','B')` → TRUE<br>• `'A' IN ('C','B')` → FALSE |

### IS

a 가 b 와 동일한 경우 참을 반환합니다.

| | |
|---|---|
| 문법 | a IS b |
| 인자 | • **a** - 모든 값<br>• **b** - 모든 값 |
| 예제 | • `'A' IS 'A'` → TRUE<br>• `'A' IS 'a'` → FALSE<br>• `4 IS 4` → TRUE<br>• `4 IS 2+2` → TRUE<br>• `4 IS 2` → FALSE<br>• `@geometry IS NULL` → 0, 사용자 도형이 NULL 이 아닌 경우 |

### IS NOT

a 와 b 가 동일하지 않은 경우 참을 반환합니다.

| | |
|---|---|
| 문법 | a IS NOT b |
| 인자 | • **a** - 값<br>• **b** - 값 |
| 예제 | • `'a' IS NOT 'b'` → TRUE<br>• `'a' IS NOT 'a'` → FALSE<br>• `4 IS NOT 2+2` → FALSE |

### LIKE

첫 번째 파라미터가 지정 패턴과 일치하는 경우 참을 반환합니다. 숫자 값도 입력할 수 있습니다.

| | |
|---|---|
| 문법 | string/number LIKE pattern |
| 인자 | • **string/number** - 값<br>• **pattern** - 비교할 패턴. 와일드카드 문자로'%', 단일 와일드카드 문자로'_'를 사용할 수 있으며, 이런 특수 문자를 이스케이프시키려면'\\'를 사용하면 됩니다. |
| 예제 | • `'A' LIKE 'A'` → TRUE<br>• `'A' LIKE 'a'` → FALSE<br>• `'A' LIKE 'B'` → FALSE<br>• `'ABC' LIKE 'B'` → FALSE<br>• `'ABC' LIKE '_B_'` → TRUE<br>• `'ABCD' LIKE '_B_'` → FALSE<br>• `'ABCD' LIKE '_B%'` → TRUE<br>• `'ABCD' LIKE '%B%'` → TRUE<br>• `'1%' LIKE '1\\%'` → TRUE<br>• `'1_' LIKE '1\\%'` → FALSE |

### NOT

조건을 부정합니다.

| | |
|---|---|
| 문법 | NOT a |
| 인자 | • **a** - 조건 |
| 예제 | • `NOT 1` → FALSE<br>• `NOT 0` → TRUE |

### NOT BETWEEN

값이 지정한 범위 내에 없을 경우 참을 반환합니다. 이 범위는 경계 값을 포함한다고 간주합니다.

| | |
|---|---|
| 문법 | value NOT BETWEEN lower_bound AND higher_bound |
| 인자 | • **value** - 범위와 비교할 값. 문자열, 숫자, 또는 날짜일 수 있습니다.<br>• **lower_bound AND higher_bound** - 범위의 경계 값들 |
| 예제 | • `'B' NOT BETWEEN 'A' AND 'C'` → FALSE<br>• `1.0 NOT BETWEEN 1.1 AND 1.2` → TRUE<br>• `2 NOT BETWEEN 2 AND 3` → FALSE<br>• `'B' NOT BETWEEN 'a' AND 'c'` → TRUE<br>• `lower('B') NOT BETWEEN 'a' AND 'b'` → FALSE |

참고

`value NOT BETWEEN lower_bound AND higher_bound` 는 `value < lower_bound OR value > higher_bound` 와 동일합니다.

더 읽어볼 거리: *BETWEEN*

### OR

조건 a 또는 조건 b 가 참이면 참을 반환합니다.

| | |
|---|---|
| 문법 | a OR b |
| 인자 | • **a** - 조건<br>• **b** - 조건 |
| 예제 | • `4 = 2+2 OR 1 = 1` → TRUE<br>• `4 = 2+2 OR 1 = 2` → TRUE<br>• `4 = 2 OR 1 = 2` → FALSE |

### []

인덱스 연산자로, 배열 또는 맵 값으로부터 요소를 반환합니다.

| | |
|---|---|
| 문법 | [index] |
| 인자 | • **index** - 배열 인덱스 또는 맵 키 값 |
| 예제 | • `array(1,2,3)[0]` → 1<br>• `array(1,2,3)[2]` → 3<br>• `array(1,2,3)[-1]` → 3<br>• `map('a',1,'b',2)['a']` → 1<br>• `map('a',1,'b',2)['b']` → 2 |

더 읽어볼 거리: *array_get*, *map_get*

### ^

a 의 b 거듭제곱을 반환합니다.

| 문법 | a ^ b |
| --- | --- |
| 인자 | • **a** - 값<br>• **b** - 값 |
| 예제 | • `5 ^ 4` → 625<br>• `5 ^ NULL` → NULL |

### ||

두 값을 문자열 하나로 결합시킵니다.

두 값 가운데 하나가 NULL 인 경우 NULL 을 반환할 것입니다. 다른 습성을 원한다면 CONCAT 함수를 찾아보십시오.

| 문법 | a \|\| b |
| --- | --- |
| 인자 | • **a** - 값<br>• **b** - 값 |
| 예제 | • `'Here' || ' and ' || 'there'` → 'Here and there'<br>• `'Nothing' || NULL` → NULL<br>• `'Dia: ' || "Diameter"` → 'Dia: 25'<br>• `1 || 2` → '12' |

더 읽어볼 거리: *concat*, *+*

### ~

문자열 값이 정규 표현식과 일치하는 경우 참을 반환합니다. 백슬래시 문자는 이중으로 이스케이프시켜야만 합니다. (예: 공백 문자와 일치시키려면"\\s")

| 문법 | string ~ regex |
| --- | --- |
| 인자 | • **string** - 문자열 값<br>• **regex** - 정규 표현식. 백슬래시 문자는"\\d"처럼 이스케이프시켜야만 합니다. |
| 예제 | • `'hello' ~ 'll'` → TRUE<br>• `'hello' ~ '^ll'` → FALSE<br>• `'hello' ~ 'llo$'` → TRUE<br>• `'abc123' ~ '\\d+'` → TRUE |

더 읽어볼 거리: *regexp_match*

### 9.2.21 공간 처리 함수

이 그룹은 공간 처리 알고리즘을 대상으로 실행되는 함수를 담고 있습니다.

#### parameter

공간 처리 알고리즘의 입력 파라미터의 값을 반환합니다.

| | |
|---|---|
| 문법 | parameter(name) |
| 인자 | • **name** - 대응하는 입력 파라미터의 이름 |
| 예제 | • `parameter('BUFFER_SIZE')` → 5.6 |

### 9.2.22 래스터 함수

이 그룹은 래스터 레이어를 대상으로 실행되는 함수를 담고 있습니다.

#### raster_attributes

지정한 래스터 값과 일치하는 속성 테이블 항목에서 나온 필드 이름을 키로 그리고 래스터 속성 테이블 값을 값으로 가진 맵을 반환합니다.

| | |
|---|---|
| 문법 | raster_attributes(layer, band, value) |
| 인자 | • **layer** - 래스터 레이어의 이름 또는 ID<br>• **band** - 관련 속성 테이블 검색에 대한 밴드 번호<br>• **value** - 래스터 값 |
| 예제 | • `raster_attributes('vegetation', 1, raster_value('vegetation', 1, make_point(1,1)))` → {'class': 'Vegetated', 'subclass': 'Trees'}<br>• `raster_attributes('vegetation', 1, raster_value('vegetation', 1, @layer_cursor_point))` → {'class': 'Vegetated', 'subclass': 'Trees'} |

#### raster_statistic

래스터 레이어의 통계를 반환합니다.

| | |
|---|---|
| 문법 | raster_statistic(layer, band, property) |
| 인자 | • **layer** - 래스터 레이어 이름 또는 레이어 ID 가운데 하나를 나타내는 문자열<br>• **band** - 래스터 레이어의 1 로 시작하는 밴드 번호를 나타내는 정수<br>• **property** - 반환할 속성에 대응하는 문자열입니다. 사용할 수 있는 옵션은 다음과 같습니다:<br>– min: 최소값<br>– max: 최대값<br>– avg: 평균값<br>– stdev: 값들의 표준 편차<br>– range: 값의 범위 (최대 - 최소)<br>– sum: 래스터에서 나온 모든 값들의 합계 |
| 예제 | • `raster_statistic('lc',1,'avg')` → 'lc'래스터 레이어의 밴드 1 의 평균값<br>• `raster_statistic('ac2010',3,'min')` → 'ac2010'래스터 레이어의 밴드 3 의 최소값 |

#### raster_value

입력 포인트 위치의 래스터 값을 반환합니다.

| | |
|---|---|
| 문법 | raster_value(layer, band, point) |
| 인자 | • **layer** - 래스터 레이어의 이름 또는 ID<br>• **band** - 값을 가져올 밴드 번호<br>• **point** - 포인트 도형 (하나 이상의 부분을 가진 다중 부분 도형의 경우, NULL 값을 반환할 것입니다) |
| 예제 | • `raster_value('dem', 1, make_point(1,1))` → 25<br>• `raster_value('ndvi', 2, @layer_cursor_point)` → 25 |

### 9.2.23 레코드 및 속성 함수

이 그룹은 레코드 식별자를 대상으로 하는 함수를 담고 있습니다.

#### attribute

피처로부터 속성을 반환합니다.

**변이형 1**

현재 피처에서 나온 속성의 값을 반환합니다.

| | |
|---|---|
| 문법 | attribute(attribute_name) |
| 인자 | • **attribute_name** - 반환할 속성의 이름 |
| 예제 | • `attribute( 'name' )` → 현재 피처의'name'속성에 저장된 값 |

**변이형 2**

대상 피처 및 속성 이름을 지정할 수 있게 해줍니다.

| | |
|---|---|
| 문법 | attribute(feature, attribute_name) |
| 인자 | • **feature** - 피처<br>• **attribute_name** - 반환할 속성의 이름 |
| 예제 | • `attribute( @atlas_feature, 'name' )` → 현재 지도책 피처의'name'속성에 저장된 값 |

### attributes

피처의 모든 속성을 담고 있는, 필드명을 맵 키로 가진 맵을 반환합니다.

**변이형 1**

현재 피처의 모든 속성의 맵을 반환합니다.

| | |
|---|---|
| 문법 | attributes() |
| 예제 | • `attributes()['name']` → 현재 피처의'name'속성에 저장된 값 |

**변이형 2**

대상 피처를 지정할 수 있게 해줍니다.

| | |
|---|---|
| 문법 | attributes(feature) |
| 인자 | • **feature** - 피처 |
| 예제 | • `attributes( @atlas_feature )['name']` → 현재 지도책 피처의'name'속성에 저장된 값 |

더 읽어볼 거리: 맵 함수

### $currentfeature

현재 피처를 평가해서 반환합니다. 'attribute'함수와 함께 사용하면 현재 피처에서 나온 속성값을 평가할 수 있습니다. **경고: 이 함수는 중요도가 떨어져 더 이상 사용되지 않고 앞으로는 사라지게 될 것입니다. 이 함수를 대체하는 @feature 변수를 사용할 것을 권장합니다.**

| | |
|---|---|
| 문법 | $currentfeature |
| 예제 | • `attribute( $currentfeature, 'name' )` → 현재 피처의'name'속성에 저장된 값 |

### display_expression

레이어에서 지정한 피처에 대한 표시 (display) 표현식을 반환합니다. 기본적으로 표현식을 평가합니다. 인자를 0 개, 1 개 또는 그 이상 사용할 수 있습니다. 자세한 내용은 다음을 참조하세요.

#### 파라미터 없음

파라미터 없이 호출하는 경우, 이 함수는 현재 레이어에 있는 현재 피처의 표시 표현식을 평가할 것입니다.

| | |
|---|---|
| 문법 | display_expression() |
| 예제 | • `display_expression()` → 현재 레이어에 있는 현재 피처의 표시 표현식 |

#### '피처'1 개 파라미터

'feature'파라미터 1 개로만 호출하는 경우, 이 함수는 현재 레이어에서 지정한 피처를 평가할 것입니다.

| | |
|---|---|
| 문법 | display_expression(feature) |
| 인자 | • **feature** - 평가해야 할 피처 |
| 예제 | • `display_expression(@atlas_feature)` → 현재 지도책 피처의 표시 표현식 |

#### 레이어와 피처 파라미터

함수를 레이어와 피처와 함께 호출하는 경우, 지정한 레이어에서 지정한 피처를 평가할 것입니다.

| | |
|---|---|
| 문법 | display_expression(layer, feature, [evaluate=true])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **layer** - 레이어 (또는 레이어 ID 또는 이름)<br>• **feature** - 평가해야 할 피처<br>• **evaluate** - 표현식을 평가해야만 하는지 여부. 거짓인 경우, 표현식을 문자 그대로의 문자열로 반환할 것입니다. (이 문자열을 나중에'eval'함수로 평가할 수도 있습니다.) |
| 예제 | • `display_expression( 'streets', get_feature_by_id('streets', 1))` → 'streets'레이어에 있는 ID 1 인 피처의 표시 표현식<br>• `display_expression('a_layer_id', @feature, 'False')` → 지정한 피처의 평가되지 않은 표시 표현식 |

### feature_id

피처의 유일 ID 를 반환하거나, 피처가 무결하지 않은 경우 NULL 을 반환합니다.

| | |
|---|---|
| 문법 | feature_id(feature) |
| 인자 | • **feature** - 피처 객체 |
| 예제 | • `feature_id( @feature )` → 현재 피처의 ID |

더 읽어볼 거리: *get_feature_by_id*

### get_feature

지정한 속성값과 일치하는 레이어의 첫 번째 피처를 반환합니다.

**단일 값 변이형**

레이어 ID 와 함께, 단일 열과 값을 지정합니다.

| | |
|---|---|
| 문법 | get_feature(layer, attribute, value) |
| 인자 | • **layer** - 레이어 이름 또는 ID<br>• **attribute** - 일치 여부에 사용할 속성명<br>• **value** - 일치시킬 속성값 |
| 예제 | • `get_feature('streets','name','main st')` → "name"필드에"main st"값을 가진"streets"레이어에서 검색된 첫 번째 피처 |

**맵 변이형**

레이어 ID 와 함께, 열 (키) 와 그에 대응하는 값을 담고 있는 맵을 지정합니다.

| 문법 | get_feature(layer, attribute) |
| --- | --- |
| 인자 | • **layer** - 레이어 이름 또는 ID<br>• **attribute** - 사용할 열과 값 쌍을 담고 있는 맵 |
| 예제 | • `get_feature('streets',map('name','main st','lane_num','4'))` → "name"필드에"main st"값을 가지며"lane_num"필드에"4"값을 가진, "streets" 레이어에서 검색된 첫 번째 피처 |

### get_feature_by_id

레이어에서 ID 를 가진 피처를 반환합니다.

| 문법 | get_feature_by_id(layer, feature_id) |
| --- | --- |
| 인자 | • **layer** - 레이어, 레이어 이름 또는 레이어 ID<br>• **feature_id** - 반환해야 할 피처의 ID |
| 예제 | • `get_feature_by_id('streets', 1)` → "streets"레이어에서 ID 가 1 인 피처 |

더 읽어볼 거리: *feature_id*

### $id

현재 행의 피처 ID 를 반환합니다. **경고: 이 함수는 중요도가 떨어져 더 이상 사용되지 않고 앞으로는 사라지게 될 것입니다. 이 함수를 대체하는 @id 변수를 사용할 것을 권장합니다.**

| 문법 | $id |
| --- | --- |
| 예제 | • `$id` → 42 |

더 읽어볼 거리: *feature_id*, *get_feature_by_id*

### is_attribute_valid

특정 피처 속성이 모든 제약조건을 만족시킬 경우 참을 반환합니다.

| | |
|---|---|
| 문법 | is_attribute_valid(attribute, [feature], [layer], [strength])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **attribute** - 속성 이름<br>• **feature** - 피처입니다. 설정하지 않을 경우 현재 피처를 사용할 것입니다.<br>• **layer** - 벡터 레이어입니다. 설정하지 않을 경우 현재 레이어를 사용할 것입니다.<br>• **strength** - 특정 제약조건 유형으로 좁히려면'강한 (hard)'또는'약한 (soft)'조건을 설정하십시오. 설정하지 않을 경우, 강한 제약조건 또는 약한 제한조건 가운데 하나라도 만족시키지 못하면 함수가 거짓을 반환할 것입니다. |
| 예제 | • `is_attribute_valid('HECTARES')` → TRUE, "HECTARES"필드에 있는 현재 피처의 값이 모든 제약조건을 만족시킬 경우<br>• `is_attribute_valid('HOUSES',get_feature('my_layer', 'FID', 10), 'my_layer')` → FALSE, 'my_layer'레이어에서"FID"가 10 인 피처의"HOUSES" 필드에 있는 값이 제약조건을 모두 만족시키지 못할 경우 |

더 읽어볼 거리: 제약조건

### is_feature_valid

피처가 모든 필드 제약조건을 만족시킬 경우 참을 반환합니다.

| | |
|---|---|
| 문법 | is_feature_valid([feature], [layer], [strength])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **feature** - 피처입니다. 설정하지 않을 경우 현재 피처를 사용할 것입니다.<br>• **layer** - 벡터 레이어입니다. 설정하지 않을 경우 현재 레이어를 사용할 것입니다.<br>• **strength** - 특정 제약조건 유형으로 좁히려면'강한 (hard)'또는'약한 (soft)'조건을 설정하십시오. 설정하지 않을 경우, 강한 제약조건 또는 약한 제한조건 가운데 하나라도 만족시키지 못하면 함수가 거짓을 반환할 것입니다. |
| 예제 | • `is_feature_valid(strength:='hard')` → TRUE, 현재 피처의 모든 필드가 강한 제약조건을 만족시킬 경우<br>• `is_feature_valid(get_feature('my_layer', 'FID', 10), 'my_layer')` → FALSE, 'my_layer'레이어에서"FID"가 10 인 피처의 모든 필드가 제약조건을 모두 만족시키지 못할 경우 |

더 읽어볼 거리: 제약조건

### is_selected

피처를 선택한 경우 참을 반환합니다. 인자를 0 개, 1 개 또는 그 이상 사용할 수 있습니다. 자세한 내용은 다음을 참조하세요.

#### 파라미터 없음

파라미터 없이 호출하는 경우, 현재 레이어에서 현재 피처를 선택했다면 이 함수는 참을 반환할 것입니다.

| 문법 | is_selected() |
|---|---|
| 예제 | • `is_selected()` → TRUE, 현재 레이어에서 현재 피처를 선택한 경우 |

#### '피처'1 개 파라미터

'feature'파라미터 1 개로만 호출하는 경우, 현재 레이어에서 지정한 피처를 선택했다면 이 함수는 참을 반환합니다.

| 문법 | is_selected(feature) |
|---|---|
| 인자 | • **feature** - 선택했는지 검증해야 할 피처 |
| 예제 | • `is_selected(@atlas_feature)` → TRUE, 현재 지도책 피처를 선택한 경우<br>• `is_selected(get_feature('streets', 'name', 'Main St.'))` → TRUE, 활성화된"streets"레이어에서 유일한"Main St."이름을 가진 피처를 선택한 경우<br>• `is_selected(get_feature_by_id('streets', 1))` → TRUE, 활성화된 "streets"레이어에서 ID 가 1 인 피처를 선택한 경우 |

#### 파라미터 2 개

함수를 레이어 및 피처와 함께 호출하는 경우, 지정한 레이어에서 지정한 피처를 선택했다면 이 함수는 참을 반환할 것입니다.

| 문법 | is_selected(layer, feature) |
|---|---|
| 인자 | • **layer** - 선택됐는지 검증할 레이어 (또는 레이어 ID 또는 이름)<br>• **feature** - 선택했는지 검증해야 할 피처 |
| 예제 | • `is_selected( 'streets', get_feature('streets', 'name', "street_name"))` → TRUE, 현재 건물의 거리를 선택한 경우 (건물 레이어가 'street_name'이라는 필드명을 가지고 있고'streets'레이어가 유일값들을 담고 있는'name' 이라는 필드를 가고 있다고 가정합니다.)<br>• `is_selected( 'streets', get_feature_by_id('streets', 1))` → TRUE, "streets"레이어에서 ID 가 1 인 피처를 선택한 경우 |

### maptip

레이어에서 지정한 피처에 대한 맵 도움말 (maptip) 을 반환합니다. 기본적으로 표현식을 평가합니다. 인자를 0 개, 1 개 또는 그 이상 사용할 수 있습니다. 자세한 내용은 다음을 참조하세요.

#### 파라미터 없음

파라미터 없이 호출하는 경우, 이 함수는 현재 레이어에 있는 현재 피처의 맵 도움말을 평가할 것입니다.

| | |
|---|---|
| 문법 | maptip() |
| 예제 | • `maptip()` → 현재 레이어에 있는 현재 피처의 맵 도움말 |

#### '피처'1 개 파라미터

'feature'파라미터 1 개로만 호출하는 경우, 이 함수는 현재 레이어에서 지정한 피처를 평가할 것입니다.

| | |
|---|---|
| 문법 | maptip(feature) |
| 인자 | • **feature** - 평가해야 할 피처 |
| 예제 | • `maptip(@atlas_feature)` → 현재 지도책 피처의 맵 도움말 |

#### 레이어와 피처 파라미터

함수를 레이어와 피처와 함께 호출하는 경우, 지정한 레이어에서 지정한 피처를 평가할 것입니다.

| | |
|---|---|
| 문법 | maptip(layer, feature, [evaluate=true])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **layer** - 레이어 (또는 레이어 ID 또는 이름)<br>• **feature** - 평가해야 할 피처<br>• **evaluate** - 표현식을 평가해야만 하는지 여부. 거짓인 경우, 표현식을 문자 그대로의 문자열로 반환할 것입니다. (나중에'eval_template'함수로 평가할 수도 있습니다.) |
| 예제 | • `maptip('streets', get_feature_by_id('streets', 1))` → 'streets' 레이어에 있는 ID 1 인 피처의 맵 도움말<br>• `maptip('a_layer_id', @feature, 'False')` → 지정한 피처의 평가되지 않은 맵 도움말 |

### num_selected

지정한 레이어에서 선택한 객체들의 개수를 반환합니다. 기본적으로 표현식을 평가한 레이어 상에서 작동합니다.

| | |
|---|---|
| 문법 | num_selected([layer=current layer])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **layer** - 선택됐는지 검증할 레이어 (또는 레이어 ID 또는 이름) |
| 예제 | • `num_selected()` → 현재 레이어에서 선택한 피처 개수<br>• `num_selected('streets')` → 'streets'레이어에서 선택한 피처 개수 |

### represent_attributes

속성 이름이 키이고 환경 설정된 표현 값이 값인 키-값 쌍을 가진 맵을 반환합니다. 속성의 표현 값은 각 속성에 대해 환경 설정된 위젯 유형에 따라 달라집니다. 인자가 0 개, 1 개, 또는 그 이상일 수 있습니다. 자세한 내용은 다음을 참조하세요.

#### 파라미터 없음

파라미터 없이 호출하는 경우, 이 함수는 현재 레이어에 있는 현재 피처의 속성 표현 값을 반환할 것입니다.

| | |
|---|---|
| 문법 | represent_attributes() |
| 예제 | • `represent_attributes()` → 현재 피처의 속성 표현 값 |

#### '피처'1 개 파라미터

'feature'파라미터 1 개로만 호출하는 경우, 이 함수는 현재 레이어에서 지정한 피처의 속성 표현 값을 반환할 것입니다.

| | |
|---|---|
| 문법 | represent_attributes(feature) |
| 인자 | • **feature** - 평가해야 할 피처 |
| 예제 | • `represent_attributes(@atlas_feature)` → 현재 레이어에서 지정한 피처의 속성 표현 값 |

#### 레이어와 피처 파라미터

'layer'와'feature'파라미터 2 개로 호출하는 경우, 이 함수는 지정한 레이어에서 지정한 피처의 속성 표현 값을 반환할 것입니다.

| 문법 | represent_attributes(layer, feature) |
| --- | --- |
| 인자 | • **layer** - 레이어 (또는 레이어 ID 또는 이름)<br>• **feature** - 평가해야 할 피처 |
| 예제 | • `represent_attributes('atlas_layer', @atlas_feature)` → 지정한 레이어에서 지정한 피처의 속성 표현 값 |

더 읽어볼 거리: *represent_value*

### represent_value

필드값에 대해 구성된 표현값을 반환합니다. 구성된 위젯 유형에 따라 다릅니다. 흔히'Value Map'위젯에 유용합니다.

| 문법 | represent_value(value, [fieldName])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| --- | --- |
| 인자 | • **value** - 해석되어야 할 값입니다. 대부분 필드일 겁니다.<br>• **fieldName** - 위젯 환경 설정을 불러와야 할 필드 이름입니다. |
| 예제 | • `represent_value("field_with_value_map")` → 값에 대한 설명<br>• `represent_value('static value', 'field_name')` → 통계값에 대한 설명 |

더 읽어볼 거리: 위젯 유형, *represent_attributes*

### sqlite_fetch_and_increment

SQLite 데이터베이스에서 자동으로 증가하는 값을 관리합니다.

SQLite 기본값들은 삽입할 수 있을 뿐 사전에 불러올 수는 없습니다.

이런 습성 때문에, 데이터베이스에 행을 생성하기 전에 AUTO_INCREMENT 를 통해 증가하는 기본 키를 획득할 수 없게 됩니다. 주의: PostgreSQL 의 경우, *evaluate default values* 옵션을 통해 사전에 기본 키를 가져올 수 있습니다.

관계를 가진 새 피처를 추가할 때, 부모 양식이 아직 열려 있어서 부모 피처가 커밋되기 전에 이미 부모에 대한 자식을 추가할 수 있다면 참 편리하겠지요.

이런 제약을 해결하기 위해, GeoPackage 같은 포맷에 기반한 SQLite 에 있는 개별 테이블에 일련의 값들을 관리하는 데 이 함수를 사용할 수 있습니다.

시퀀스 ID(filter_attribute 및 filter_value) 에 대해 시퀀스 테이블을 필터링하고, id_field 의 현재 값을 1 씩 증가시킨 다음 증가된 값을 반환할 것입니다.

추가적인 열들이 지정할 값들을 요구하는 경우, 이 목적을 위해 default_values 맵을 사용할 수 있습니다.

주의

이 함수는 대상 SQLite 테이블을 수정합니다. 이 함수는 속성에 대한 기본값 환경 설정을 위해 활용되도록 만들어졌습니다.

데이터베이스 파라미터가 레이어이고 레이어가 트랜잭션 모드인 경우, 트랜잭션 생애 주기 동안 한번만 값을 가져와 캐시에 저장하고 증가시킬 것입니다. 따라서 동일한 데이터베이스에서 여러 프로세스를 병렬 작업하는 것은 안전하지 않습니다.

| | |
|---|---|
| 문법 | sqlite_fetch_and_increment(database, table, id_field, filter_attribute, filter_value, [default_values])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **database** - SQLite 파일 또는 GeoPackage 레이어를 가리키는 경로<br>• **table** - 시퀀스를 관리하는 테이블의 이름<br>• **id_field** - 현재 값을 담고 있는 필드의 이름<br>• **filter_attribute** - 이 시퀀스의 고유 식별자를 담고 있는 필드의 이름입니다. UNIQUE 인덱스를 가지고 있어야만 합니다.<br>• **filter_value** - 사용할 시퀀스의 이름<br>• **default_values** - 테이블에 있는 추가적인 열들의 기본값을 가진 맵입니다. 이 값들은 인용 부호로 묶여 있어야 합니다. 이 파라미터에 함수를 입력할 수 있습니다. |
| 예제 | • `sqlite_fetch_and_increment(@layer, 'sequence_table', 'last_unique_id', 'sequence_id', 'global', map('last_change', 'date(''now'')', 'user', '''' || @user_account_name || ''''))` → 0<br>• `sqlite_fetch_and_increment(layer_property(@layer, 'path'), 'sequence_table', 'last_unique_id', 'sequence_id', 'global', map('last_change', 'date(''now'')', 'user', '''' || @user_account_name || ''''))` → 0 |

더 읽어볼 거리: 데이터소스 속성, 여러 레이어 간 관계 설정하기

### uuid

QUuid::createUuid 메소드를 사용해서 각 행에 대한 범용 고유 식별자 (UUID) 를 생성합니다.

| | |
|---|---|
| 문법 | uuid([format='WithBraces'])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **format** - UUID 의 서식이 될 서식. 'WithBraces', 'WithoutBraces'또는'Id128'가운데 하나로 지정할 수 있습니다. |
| 예제 | • `uuid()` → '{0bd2f60f-f157-4a6d-96af-d4ba4cb366a1}'<br>• `uuid('WithoutBraces')` → '0bd2f60f-f157-4a6d-96af-d4ba4cb366a1'<br>• `uuid('Id128')` → '0bd2f60ff1574a6d96afd4ba4cb366a1' |

### 9.2.24 관계

이 그룹은 현재 프로젝트에서 사용할 수 있는관계 목록을 그 설명과 함께 담고 있습니다. 이 목록을 통해 표현식을 작성하거나 (예: *relation_aggregate* 함수) 양식을 사용자 정의하는 데 필요한 관계 ID 에 빠르게 접근할 수 있습니다.

### 9.2.25 센서 함수

이 그룹은 센서와 대화형 작업을 하기 위한 함수를 담고 있습니다.

#### sensor_data

등록된 센서에서 마지막으로 캡처된 값을 (또는 여러 개의 값을 보고하는 센서의 경우 값들을 맵으로) 반환합니다.

| | |
|---|---|
| 문법 | sensor_data(name, [expiration])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **name** - 센서 이름<br>• **expiration** - 마지막으로 캡처된 값 이후 허용되는 최대 밀리 초 |
| 예제 | • `sensor_data('geiger_1')` → '2000' |

### 9.2.26 문자열 함수

이 그룹은 문자열을 대상으로 하는 (예: 치환, 대문자 변환 등) 함수를 담고 있습니다.

#### ascii

문자열의 첫 번째 문자와 관련된 유니코드 코드를 반환합니다.

| | |
|---|---|
| 문법 | ascii(string) |
| 인자 | • **string** - 유니코드 코드로 변환할 문자열 |
| 예제 | • `ascii('Q')` → 81 |

#### char

유니코드 코드와 관련된 문자를 반환합니다.

| | |
|---|---|
| 문법 | char(code) |
| 인자 | • **code** - 유니코드 코드 번호 |
| 예제 | • `char(81)` → 'Q' |

### concat

여러 문자열을 하나로 연결합니다. NULL 값은 빈 문자열로 변환되고, (숫자 같은) 다른 값은 문자열로 변환됩니다.

| | |
|---|---|
| 문법 | concat(string1, string2, ⋯) |
| 인자 | • **string** - 문자열 값 |
| 예제 | • `concat('sun', 'set')` → 'sunset'<br>• `concat('a','b','c','d','e')` → 'abcde'<br>• `concat('Anno ', 1984)` → 'Anno 1984'<br>• `concat('The Wall', NULL)` → 'The Wall' |

#### 필드 연결에 대해

특수한 몇몇 특성을 가진 `||` 또는 + 가운데 하나를 사용해서도 문자열 또는 필드값을 연결할 수 있습니다:

- + 연산자는 덧셈 표현식이기도 합니다. 따라서 정수형 (필드 또는 숫자값) 피연산자가 있을 경우 오류를 낼 가능성이 있으므로 `||` 연산자 또는 `concat` 함수를 사용하는 편이 좋습니다:

```
'My feature id is: ' + "gid" => triggers an error as gid returns an integer
```

- When any of the arguments is a NULL value, either `||` or + will return a NULL value. To return the other arguments regardless the NULL value, you may want to use the `concat` or `array_to_string` function:

```
'My feature id is: ' + NULL ==> NULL
'My feature id is: ' || NULL => NULL
concat('My feature id is: ', NULL) => 'My feature id is: '
array_to_string( array('My feature id is: ', NULL) ) => 'My feature id is: '
```

further reading: *array_to_string*, ||, +

### format

지정한 인자를 이용해서 문자열 서식을 변경합니다.

| | |
|---|---|
| 문법 | format(string, arg1, arg2, ⋯) |
| 인자 | • **string** - 인자를 위한 자리 표시자 (placeholder) %1, %2 등등을 가진 문자열. 자리 표시자를 반복 사용할 수 있습니다. 가장 낮은 번호를 가진 자리 표시자는 arg1 으로 치환되고, 그 다음은 arg2 로⋯치환됩니다.<br>• **arg** - 모든 유형, 제한 없는 개수의 인자 |
| 예제 | • `format('This %1 a %2','is', 'test')` → 'This is a test'<br>• `format('This is %2','a bit unexpected but 2 is lowest number in string','normal')` → 'This is a bit unexpected but 2 is lowest number in string' |

### format_date

날짜 유형 또는 문자열을 사용자 정의 문자열 서식으로 변환합니다. Qt 날짜 & 시간 서식 문자열을 사용하십시오. QDateTime::toString 을 참조하세요.

문법

format_date(datetime, format, [language])
[] 괄호는 선택적인 인자를 표시합니다

인자

- **datetime** - 날짜, 시간 또는 날짜 & 시간 값
- **format** - 문자열을 서식화하는 데 쓰이는 문자열 템플릿

| 표현식 | 산출물 |
|---|---|
| d | 앞에 0 이 없는 숫자 일 (1 에서 31) |
| dd | 앞에 0 이 있는 숫자 일 (01 에서 31) |
| ddd | 축약된 현지 요일 이름 (예: 'Mon'에서'Sun') |
| dddd | 축약하지 않은 현지 요일 이름 (예: 'Monday'에서'Sunday') |
| M | 앞에 0 이 없는 숫자 월 (1-12) |
| MM | 앞에 0 이 있는 숫자 월 (01-12) |
| MMM | 축약된 현지 월 이름 (예: 'Jan'에서'Dec') |
| MMMM | 축약하지 않은 현지 월 이름 (예: 'January'에서'December') |
| yy | 두 자리 숫자 연도 (00-99) |
| yyyy | 네 자리 숫자 연도 |

다음 표현식은 서식 문자열의 시간 부분에 쓰일 수도 있습니다:

| 표현식 | 산출물 |
|---|---|
| h | 앞에 0 이 없는 시간 (0 에서 23 또는 AM/PM 표시가 있는 경우 1 에서 12) |
| hh | 앞에 0 이 있는 시간 (00 에서 23 또는 AM/PM 표시가 있는 경우 01 에서 12) |
| H | 앞에 0 이 없는 시간 (AM/PM 표시가 있는 경우에도 0 에서 23) |
| HH | 앞에 0 이 있는 시간 (AM/PM 표시가 있는 경우에도 00 에서 23) |
| m | 앞에 0 이 없는 분 (0 에서 59) |
| mm | 앞에 0 이 있는 분 (00 에서 59) |
| s | 앞에 0 이 없는 초 (0 에서 59) |
| ss | 앞에 0 이 있는 초 (00 에서 59) |
| z | 뒤에 0 이 없는 밀리초 (0 에서 999) |
| zzz | 뒤에 0 이 있는 밀리초 (000 에서 999) |
| AP or A | AM/PM 시간으로 해석합니다. *AP* 는'AM'또는'PM'이어야만 합니다. |
| ap or a | AM/PM 시간으로 해석합니다. *ap* 는'am'또는'pm'이어야만 합니다. |

- **language** - 날짜 유형을 사용자 지정 문자열로 서식화하는 데 사용된 언어 (소문자, 2-3 문자, ISO 639 언어 코드); 기본적으로 현재 QGIS 사용자 로케일을 사용합니다.

예제

- `format_date('2012-05-15','dd.MM.yyyy')` → '15.05.2012'
- `format_date('2012-05-15','d MMMM yyyy','fr')` → '15 mai 2012'
- `format_date('2012-05-15','dddd')` → 'Tuesday', 현재 로케일이 영어 변이형인 경우
- `format_date('2012-05-15 13:54:20','dd.MM.yy')` → '15.05.12'
- `format_date('13:54:20','hh:mm AP')` → '01:54 PM'

### format_number

천 단위 로케일 구분자로 구분된 서식을 가진 숫자를 반환합니다. 기본적으로 현재 QGIS 사용자 로케일을 사용합니다. 또한 소수점 이하 자릿수도 지정한 자릿수대로 잘라 맞춥니다.

| | |
|---|---|
| 문법 | format_number(number, [places=0], [language], [omit_group_separators=false], [trim_trailing_zeroes=false])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **number** - 서식화할 숫자<br>• **places** - 문자열을 잘라 맞출 십진 자릿수를 나타내는 정수입니다.<br>• **language** - 숫자 유형을 문자열로 서식화하는 데 사용된 언어 (소문자, 2-3 문자, ISO 639 언어 코드); 기본적으로 현재 QGIS 사용자 로케일을 사용합니다.<br>• **omit_group_separators** - 이 인자를 참으로 설정하면 문자열에 그룹 구분자를 포함시키지 않을 것입니다.<br>• **trim_trailing_zeroes** - 이 인자를 참으로 설정하면 문자열에서 소수점 뒤의 후행 0 들을 잘라낼 것입니다. |
| 예제 | • `format_number(10000000.332,2)` → '10,000,000.33', 현재 로케일이 예를 들어 영어 변이형인 경우<br>• `format_number(10000000.332,2,'fr')` → '10 000 000,33' |

### left

문자열 가장 왼쪽에 있는 *n* 개의 문자를 담은 하위 문자열을 반환합니다.

| | |
|---|---|
| 문법 | left(string, length) |
| 인자 | • **string** - 문자열<br>• **length** - 정수. 문자열의 왼쪽을 기준으로 반환할 문자의 개수입니다. |
| 예제 | • `left('Hello World',5)` → 'Hello' |

Further reading: *right*

### length

문자열의 문자 개수 또는 라인스트링 도형의 길이를 반환합니다.

**문자열 변이형**

문자열에 있는 문자의 개수를 반환합니다.

| | |
|---|---|
| 문법 | length(string) |
| 인자 | • **string** - 길이를 셀 문자열 |
| 예제 | • `length('hello')` → 5 |

#### 도형 변이형

라인 도형 객체의 길이를 계산합니다. 언제나 해당 도형의 공간 좌표계 (SRS) 에서 평면 측량해서 계산하므로, 반환한 길이의 단위가 SRS 용 단위와 일치할 것입니다. 이것이 $length 함수가 수행하는 계산과 다른 점인데, $length 함수는 프로젝트의 타원체 및 거리 단위 설정을 기반으로 타원체 상에서 계산을 수행할 것입니다.

| | |
|---|---|
| 문법 | length(geometry) |
| 인자 | • **geometry** - 라인 도형 객체 |
| 예제 | • `length(geom_from_wkt('LINESTRING(0 0, 4 0)'))` → 4.0 |

### lower

문자열을 소문자로 변환합니다.

| | |
|---|---|
| 문법 | lower(string) |
| 인자 | • **string** - 소문자로 변환할 문자열 |
| 예제 | • `lower('HELLO World')` → 'hello world' |

Further reading: *upper*, *title*

### lpad

채우기 문자를 사용하여 왼쪽에서 지정된 너비까지 채운 문자열을 반환합니다. 대상 너비가 문자열의 길이보다 작은 경우 문자열이 잘립니다.

| | |
|---|---|
| 문법 | lpad(string, width, fill) |
| 인자 | • **string** - 채울 문자열<br>• **width** - 새 문자열의 길이<br>• **fill** - 남은 공백을 채울 문자 |
| 예제 | • `lpad('Hello', 10, 'x')` → 'xxxxxHello'<br>• `lpad('Hello', 3, 'x')` → 'Hel' |

Further reading: *rpad*

### ltrim

문자열의 시작에서부터 지정한 문자들만 (기본값은 공백) 담고 있는 가장 긴 문자열을 제거합니다.

| | |
|---|---|
| 문법 | ltrim(string, [characters='']) <br> [] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **string** - 다듬을 문자열 <br> • **characters** - 잘라낼 문자열 |
| 예제 | • `ltrim(' hello world ')` → 'hello world ' <br> • `ltrim('zzzytest', 'xyz')` → 'test' |

더 읽어볼 거리: *rtrim*, *trim*

### regexp_match

유니코드 문자열 내부에서 정규 표현식과 첫번째로 일치하는 위치를 반환하거나, 하위 문자열을 찾지 못했을 경우 0 을 반환합니다.

| | |
|---|---|
| 문법 | regexp_match(input_string, regex) |
| 인자 | • **input_string** - 정규 표현식에 대해 테스트할 문자열 <br> • **regex** - 테스트시킬 정규 표현식입니다. 백슬래시 문자는 이중으로 이스케이프시켜야만 합니다. (예: 공백 문자와 매칭하려면"\\s", 또는 단어 경계 (word boundary) 를 일치시키려면"\\b") |
| 예제 | • `regexp_match('QGIS ROCKS','\\sROCKS')` → 5 <br> • `regexp_match('Budač','udač\\b')` → 2 |

Further reading: *strpos*

### regexp_replace

지정한 정규 표현식을 치환한 문자열을 반환합니다.

| | |
|---|---|
| 문법 | regexp_replace(input_string, regex, replacement) |
| 인자 | • **input_string** - 일치하는 경우 치환할 문자열<br>• **regex** - 치환할 정규 표현식입니다. 백슬래시 문자는 이중으로 이스케이프시켜야만 합니다. (예: 공백 문자와 매칭하려면"\\s") <br>• **replacement** - 지정한 정규 표현식과 일치하는 문자열을 치환할 문자열입니다. 캡처된 그룹은 \\1, \\2 등을 사용하여 대체 문자열에 삽입할 수 있습니다. |
| 예제 | • `regexp_replace('QGIS SHOULD ROCK','\\sSHOULD\\s',' DOES ')` → 'QGIS DOES ROCK'<br>• `regexp_replace('ABC123','\\d+','')` → 'ABC'<br>• `regexp_replace('my name is John','(.*) is (.*)','\\2 is \\1')` → 'John is my name' |

Further reading: *replace*, *array_replace*

### regexp_substr

문자열에서 지정한 정규 표현식과 일치하는 부분을 반환합니다.

| | |
|---|---|
| 문법 | regexp_substr(input_string, regex) |
| 인자 | • **input_string** - 일치하는 경우 검색할 문자열<br>• **regex** - 일치시킬 정규 표현식. 백슬래시 문자는 이중으로 이스케이프시켜야만 합니다. (예: 공백 문자와 일치시키려면"\\s") |
| 예제 | • `regexp_substr('abc123','(\\d+)')` → '123' |

Further reading: *substr*, *regexp_matches*

### repeat

Repeats a string a specified number of times.

| | |
|---|---|
| 문법 | repeat(string, number) |
| 인자 | • **string** - string to repeat<br>• **number** - number of times to repeat string |
| 예제 | • `repeat('Hello', 3)` → 'HelloHelloHello' |

### replace

문자열을 지정한 문자열, 배열, 또는 문자열의 맵으로 치환해서 반환합니다.

#### 문자열 및 배열 변이형

지정한 문자열 또는 문자열 배열을 문자열 또는 문자열 배열로 치환해서 반환합니다.

| | |
|---|---|
| 문법 | replace(string, before, after) |
| 인자 | • **string** - 입력 문자열<br>• **before** - 치환할 문자열 또는 문자열 배열<br>• **after** - 치환에 사용할 문자열 또는 문자열 배열 |
| 예제 | • `replace('QGIS SHOULD ROCK','SHOULD','DOES')` → 'QGIS DOES ROCK'<br>• `replace('QGIS ABC',array('A','B','C'),array('X','Y','Z'))` → 'QGIS XYZ'<br>• `replace('QGIS',array('Q','S'),'')` → 'GI' |

#### 맵 변이형

지정한 맵 키를 짝지은 값으로 치환한 문자열을 반환합니다. 더 긴 맵 키를 더 먼저 평가합니다.

| | |
|---|---|
| 문법 | replace(string, map) |
| 인자 | • **string** - 입력 문자열<br>• **map** - 키와 값을 담고 있는 맵 |
| 예제 | • `replace('APP SHOULD ROCK',map('APP','QGIS','SHOULD','DOES'))` → 'QGIS DOES ROCK'<br>• `replace('forty two',map('for','4','two','2','forty two','42'))` → '42' |

Further reading: *regexp_replace*, *array_replace*

### reverse

Reverses the direction of a line string or reverses a string of text.

#### 문자열 변이형

Reverses the order of characters in a string.

| | |
|---|---|
| 문법 | reverse(string) |
| 인자 | • **string** - string to reverse |
| 예제 | • `reverse('hello')` → 'olleh' |

#### 도형 변이형

라인스트링의 꼭짓점 순서를 역전시켜 라인스트링의 방향을 반전시킵니다.

| | |
|---|---|
| 문법 | reverse(geometry) |
| 인자 | • **geometry** - 도형 |
| 예제 | • `geom_to_wkt(reverse(geom_from_wkt('LINESTRING(0 0, 1 1, 2 2)')))` → 'LINESTRING(2 2, 1 1, 0 0)' |

### right

문자열 가장 오른쪽에 있는 *n* 개의 문자를 담은 하위 문자열을 반환합니다.

| | |
|---|---|
| 문법 | right(string, length) |
| 인자 | • **string** - 문자열<br>• **length** - 정수. 문자열의 오른쪽을 기준으로 반환할 문자의 개수입니다. |
| 예제 | • `right('Hello World',5)` → 'World' |

Further reading: *left*

### rpad

채우기 문자를 사용하여 오른쪽에서 지정된 너비까지 채운 문자열을 반환합니다. 대상 너비가 문자열의 길이보다 작은 경우 문자열이 잘립니다.

| | |
|---|---|
| 문법 | rpad(string, width, fill) |
| 인자 | • **string** - 채울 문자열<br>• **width** - 새 문자열의 길이<br>• **fill** - 남은 공백을 채울 문자 |
| 예제 | • `rpad('Hello', 10, 'x')` → 'Helloxxxxx'<br>• `rpad('Hello', 3, 'x')` → 'Hel' |

Further reading: *lpad*

### rtrim

문자열의 끝에서부터 지정한 문자들만 (기본값은 공백) 담고 있는 가장 긴 문자열을 제거합니다.

| | |
|---|---|
| 문법 | rtrim(string, [characters=''])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **string** - 다듬을 문자열<br>• **characters** - 잘라낼 문자열 |
| 예제 | • `rtrim(' hello world ')` → 'hello world'<br>• `rtrim('testxxzx', 'xyz')` → 'test' |

더 읽어볼 거리: *ltrim*, *trim*

### strpos

또다른 문자열 안에 있는 하위 문자열 가운데 첫 번째 일치하는 위치를, 또는 하위 문자열을 찾을 수 없는 경우 0 을 반환합니다.

| | |
|---|---|
| 문법 | strpos(haystack, needle) |
| 인자 | • **haystack** - 검색 대상 문자열<br>• **needle** - 검색할 문자열 |
| 예제 | • `strpos('HELLO WORLD','WORLD')` → 7<br>• `strpos('HELLO WORLD','GOODBYE')` → 0 |

더 읽어볼 거리: *regexp_match*

### substr

문자열의 부분을 반환합니다.

| | |
|---|---|
| 문법 | substr(string, start, [length])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **string** - 전체 입력 문자열<br>• **start** - 추출할 시작 위치를 나타내는 1 부터 시작하는 정수; 시작 위치가 음수이면 반환되는 문자열이 문자열 마지막에서 시작 값을 뺀 위치에서 시작합니다.<br>• **length** - 추출할 문자열의 길이를 나타내는 정수. 길이가 음수이면 반환 문자열은 문자열의 끝에서 지정된 길이의 문자를 생략합니다. |
| 예제 | • `substr('HELLO WORLD',3,5)` → 'LLO W'<br>• `substr('HELLO WORLD',6)` → 'WORLD'<br>• `substr('HELLO WORLD',-5)` → 'WORLD'<br>• `substr('HELLO',3,-1)` → 'LL'<br>• `substr('HELLO WORLD',-5,2)` → 'WO'<br>• `substr('HELLO WORLD',-5,-1)` → 'WORL' |

Further reading: *regexp_substr*, *regexp_matches*

### title

문자열의 모든 단어를 제목 서식 (모든 단어의 첫 문자가 대문자, 나머지 문자는 소문자) 으로 변환합니다.

| | |
|---|---|
| 문법 | title(string) |
| 인자 | • **string** - 제목 서식으로 변환할 문자열 |
| 예제 | • `title('hello WOrld')` → 'Hello World' |

Further reading: *lower*, *upper*

### to_string

Converts a number to string. The conversion is not locale-aware, see 'format_number'for a locale-aware alternative.

| | |
|---|---|
| 문법 | to_string(number) |
| 인자 | • **number** - 정수형 또는 실수형 값. 문자열로 변환할 숫자입니다. |
| 예제 | • `to_string(1.23)` → '1.23' |

Further reading: *format_number*

### trim

문자열에서 맨 앞과 맨 뒤의 모든 빈 자리 (공백, 탭 등) 를 제거합니다.

| | |
|---|---|
| 문법 | trim(string) |
| 인자 | • **string** - 다듬을 문자열 |
| 예제 | • `trim(' hello world ')` → 'hello world' |

더 읽어볼 거리: *ltrim*, *rtrim*

### upper

문자열을 대문자로 변환합니다.

| | |
|---|---|
| 문법 | upper(string) |
| 인자 | • **string** - 대문자로 변환할 문자열 |
| 예제 | • `upper('hello WOrld')` → 'HELLO WORLD' |

Further reading: *lower*, *title*

### wordwrap

최대/최소 문자 개수로 행갈이한 문자열을 반환합니다.

| | |
|---|---|
| 문법 | wordwrap(string, wrap_length, [delimiter_string])<br>[] 괄호는 선택적인 인자를 표시합니다 |
| 인자 | • **string** - 행갈이할 문자열<br>• **wrap_length** - 정수. wrap_length 가 양수인 경우 행갈이할 문자의 이상적인 최대 개수를 나타내는 숫자이고, 음수인 경우 행갈이할 문자의 최소 개수를 나타냅니다.<br>• **delimiter_string** - 새 줄로 행갈이할 선택적인 구분자 문자열 |
| 예제 | • `wordwrap('UNIVERSITY OF QGIS',13)` → 'UNIVERSITY OF<br>QGIS'<br>• `wordwrap('UNIVERSITY OF QGIS',-3)` → 'UNIVERSITY<br>OF QGIS' |

### 9.2.27 사용자 표현식

이 그룹은사용자 표현식 으로 저장된 표현식들을 담고 있습니다.

### 9.2.28 변수

이 그룹은 응용 프로그램, 프로젝트 파일 및 기타 설정과 관련된 동적 변수를 담고 있습니다. 즉 다음 맥락에 따라 일부 변수를 사용하지 못 할 수도 있다는 뜻입니다:

-  Select by expression 대화창에서
-  Field calculator 대화창에서
- 레이어 속성 대화창에서
- 인쇄 조판기에서

표현식에 다음 변수들을 사용하려면, 앞에 `@` 문자를 (예: `@row_number`) 붙여야 합니다:

| 변수 | 설명 |
|---|---|
| algorithm_id | 알고리즘의 유일 ID |
| animation_end_time | 애니메이션의 전체 시계열 시간 범위의 종단 (날짜 & 시간 값으로) |
| animation_interval | 애니메이션의 전체 시계열 시간 범위의 기간 (간격 값으로) |
| animation_start_time | 애니메이션의 전체 시계열 시간 범위의 시작 (날짜 & 시간 값으로) |
| atlas_feature | 현재 지도책 피처 (피처 객체로써) |
| atlas_featureid | 현재 지도책 피처 ID |
| atlas_featurenumber | 현재 조판기에 있는 지도책 피처의 번호 |
| atlas_filename | 현재 지도책 파일 이름 |
| atlas_geometry | 현재 지도책 피처 도형 |
| atlas_layerid | 현재 지도책 커버리지 레이어 ID |
| atlas_layername | 현재 지도책 커버리지 레이어 이름 |
| atlas_pagename | 현재 지도책 페이지 이름 |
| atlas_totalfeatures | 지도책에 있는 피처의 총 개수 |
| band | The number of the band in the raster layer |
| band_description | The description of the band in the raster layer |
| band_name | The name of the band in the raster layer |
| canvas_cursor_point | 캔버스 상에서의 프로젝트의 지리 좌표로 된 마지막 커서 위치 |
| cluster_color | 군집 내부 심볼의 색상, 또는 심볼이 혼합 색상인 경우 NULL 값 |
| cluster_size | 군집 내부에 담겨 있는 심볼의 개수 |
| current_feature | 현재 속성 양식 또는 테이블 행에서 편집 중인 피처 |
| current_geometry | 현재 양식 또는 테이블 행에서 편집 중인 피처의 도형 |
| current_parent_feature | 부모 양식에서 현재 편집하고 있는 피처를 나타냅니다. 내장된 양식 맥락에서만 사용할 수 있습니다. |
| current_parent_geometry | 부모 양식에서 현재 편집하고 있는 피처의 도형을 나타냅니다. 내장된 양식 맥락에서만 사용할 수 있습니다. |
| form_mode | 양식이 사용되는 목적입니다. AddFeatureMode, SingleEditMode, MultiEditMode, SearchMode, AggregateSearchMode 또는 IdentifyMode 같은 문자열로 입력합니다. |
| feature | 평가되는 현재 피처입니다. 현재 피처의 속성값을 평가하기 위해'attribute' 함수와 함께 사용할 수 있습니다. |
| frame_duration | 각 애니메이션 프레임의 시계열 기간 (간격 값으로) |
| frame_number | 애니메이션 재생 시 현재 프레임 번호 |

다음 페이지에 계속

표 9.1 – 이전 페이지에서 계속

| 변수 | 설명 |
|---|---|
| frame_rate | 애니메이션 재생 시 초당 프레임 개수 |
| fullextent_maxx | (모든 레이어를 포함하는) 전체 캔버스 범위의 최대 X 값 |
| fullextent_maxy | (모든 레이어를 포함하는) 전체 캔버스 범위의 최대 Y 값 |
| fullextent_minx | (모든 레이어를 포함하는) 전체 캔버스 범위의 최소 X 값 |
| fullextent_miny | (모든 레이어를 포함하는) 전체 캔버스 범위의 최소 Y 값 |
| geometry | 현재 피처에서 평가된 도형 |
| geometry_part_count | 렌더링된 피처의 도형에 있는 부분의 개수 |
| geometry_part_num | 렌더링 중인 피처의 현재 도형 부분의 번호 |
| geometry_point_count | 렌더링된 도형의 부분에 있는 포인트의 개수 |
| geometry_point_num | 현재 렌더링된 도형의 부분에 있는 포인트의 번호 |
| geometry_ring_num | 렌더링되는 피처에서 현재 도형 고리 번호입니다. (폴리곤 피처 전용입니다.) 외곽 고리의 번호가 0 입니다. |
| grid_axis | 현재 그리드 주석 축 (예: 경도는'x', 위도는'y') |
| grid_number | 현재 그리드 주석 값 |
| id | 평가된 현재 피처의 ID |
| item_id | 조판기 항목 사용자 ID (유일해야 하는 것은 아닙니다) |
| item_uuid | 조판기 항목 유일 ID |
| layer | 현재 레이어 |
| layer_crs | 현재 레이어의 좌표계 기관 ID |
| layer_crs_ellipsoid | 현재 레이어 좌표계의 타원체 기관 ID |
| layer_cursor_point | Point geometry under the mouse position in map canvas (or the GetFeatureInfo position in context of QGIS Server), in active layer's CRS |
| layer_id | 현재 레이어의 ID |
| layer_ids | 현재 프로젝트에 있는 모든 맵 레이어의 ID 목록 |
| layer_name | 현재 레이어의 이름 |
| layer_vertical_crs | The Identifier for the vertical coordinate reference system of the layer (e.g., 'EPSG:5703') |
| layer_vertical_crs_definition | The full definition of the vertical Coordinate reference system of the layer |
| layer_vertical_crs_description | The name of the vertical Coordinate reference system of the layer |
| layer_vertical_crs_wkt | The WKT definition of the vertical Coordinate reference system of the current layer |
| layers | 현재 프로젝트에 있는 모든 맵 레이어의 목록 |
| layout_dpi | 조판 해상도 (DPI) |
| layout_name | 조판 이름 |
| layout_numpages | 조판한 페이지 수 |
| layout_page | 조판기에 있는 현재 항목의 페이지 번호 |
| layout_pageheight | 조판에서 활성화된 페이지 높이 (표준 용지 크기의 경우 밀리미터 단위, 또는 사용자 지정 용지 크기의 경우 사용되는 모든 단위) |
| layout_pageoffsets | 각 페이지의 상단 Y 좌표 배열. 페이지 크기가 변경될 수도 있다는 맥락에서 페이지에 항목을 동적으로 배치할 수 있게 해줍니다. |
| layout_pagewidth | 조판에서 활성화된 페이지 너비 (표준 용지 크기의 경우 밀리미터 단위, 또는 사용자 지정 용지 크기의 경우 사용되는 모든 단위) |
| legend_column_count | 범례에 있는 열의 개수 |
| legend_filter_by_map | 맵이 범례 내용을 필터링했는지 여부를 표시 |
| legend_filter_out_atlas | 범례에서 지도책을 필터링했는지 여부를 표시 |
| legend_split_layers | 범례에서 레이어를 나눌 수 있는지 여부를 표시 |
| legend_title | 범례의 제목 |
| legend_wrap_string | 범례 텍스트 행갈이에 쓰이는 문자 (들) |
| map_crs | 현재 맵의 좌표계 |
| map_crs_acronym | 현재 맵 좌표계의 약어 |

다음 페이지에 계속

표 9.1 – 이전 페이지에서 계속

| 변수 | 설명 |
|---|---|
| map_crs_definition | 현재 맵 좌표계의 완전한 정의 |
| map_crs_description | 현재 맵 좌표계의 이름 |
| map_crs_ellipsoid | 현재 맵 좌표계 타원체의 약어 |
| map_crs_proj4 | 현재 맵 좌표계의 PROJ4 정의 |
| map_crs_projection | 맵의 좌표계가 사용하는 투영 메소드를 설명하는 이름 (예: 'Albers Equal Area' ) |
| map_crs_wkt | 현재 맵 좌표계의 WKT 정의 |
| map_end_time | 맵의 시계열 시간 범위의 종단 (날짜 & 시간 값으로) |
| map_extent | 맵의 현재 범위를 표현하는 도형 |
| map_extent_center | 맵의 중앙에 있는 포인트 피처 |
| map_extent_height | 맵의 현재 높이 |
| map_extent_width | 맵의 현재 너비 |
| map_id | 현재 맵의 대상 (destination) ID. 캔버스 렌더링인 경우'canvas', 조판기 맵 렌더링인 경우 해당 항목의 ID 가 됩니다. |
| map_interval | 맵의 시계열 시간 범위의 기간 (간격 값으로) |
| map_layer_ids | 맵에 가시화된 맵 레이어의 ID 목록 |
| map_layers | 맵에 가시화된 맵 레이어의 목록 |
| map_rotation | 맵의 현재 기울기 |
| map_scale | 맵의 현재 축척 |
| map_start_time | 맵의 시계열 시간 범위의 시작 (날짜 & 시간 값으로) |
| map_units | 맵 측정 단위 |
| map_z_range_lower | Lower elevation of the map's elevation range |
| map_z_range_upper | Upper elevation of the map's elevation range |
| model_path | 현재 모델의 (파일 이름 포함) 전체 경로 (또는 모델이 프로젝트에 내장된 경우 프로젝트 경로) |
| model_folder | 현재 모델을 담고 있는 폴더 (또는 모델이 프로젝트에 내장된 경우 프로젝트 폴더) |
| model_name | 현재 모델의 이름 |
| model_group | 현재 모델의 그룹 |
| notification_message | 제공자가 전송하는 알림 메시지의 내용 (제공자 알림이 촉발하는 액션에 대해서만 사용할 수 있습니다.) |
| parent | 부모 레이어에 있는 현재 피처를 참조. 집계 함수를 필터링하는 경우 해당 피처의 속성 및 도형에 접근할 수 있습니다. |
| plot_axis | The associated plot axis, e.g. 'x'or 'y'. |
| plot_axis_value | The current value for the plot axis. |
| project_abstract | 프로젝트 요약. 프로젝트 메타데이터에서 가져옵니다. |
| project_area_units | 현재 프로젝트에서 쓰이는 면적 단위. 도형 면적 계산 시 사용됩니다. |
| project_author | 프로젝트 저자. 프로젝트 메타데이터에서 가져옵니다. |
| project_basename | 현재 프로젝트 파일 이름의 (경로 및 확장자를 제외한) 기본 이름 |
| project_creation_date | 프로젝트 생성 날짜. 프로젝트 메타데이터에서 가져옵니다. |
| project_crs | Identifier for the coordinate reference system of the project (e.g., 'EPSG:4326') |
| project_crs_arconym | 프로젝트 좌표계의 약어 |
| project_crs_definition | 프로젝트 좌표계의 완전한 정의 |
| project_crs_description | 프로젝트 좌표계의 설명 |
| project_crs_ellipsoid | 프로젝트 좌표계의 타원체 |
| project_crs_proj4 | 프로젝트 좌표계의 PROJ4 표현 |
| project_crs_wkt | 프로젝트 좌표계의 WKT 표현 |
| project_distance_units | 현재 프로젝트에서 쓰이는 거리 단위. 도형 및 거리의 길이 계산 시 사용됩니다. |

다음 페이지에 계속

표 9.1 – 이전 페이지에서 계속

| 변수 | 설명 |
|---|---|
| project_ellipsoid | 현재 프로젝트의 타원체 이름. 측지 (geodetic) 면적 또는 도형 길이 계산 시 사용됩니다. |
| project_filename | 현재 프로젝트의 파일 이름 |
| project_folder | 현재 프로젝트가 있는 폴더 |
| project_home | 현재 프로젝트의 홈 경로 |
| project_identifier | 프로젝트 식별자. 프로젝트 메타데이터에서 가져옵니다. |
| project_keywords | 프로젝트 키워드. 프로젝트 메타데이터에서 가져옵니다. |
| project_last_saved | 프로젝트를 마지막으로 저장한 날짜 & 시간 |
| project_path | 현재 프로젝트의 (파일 이름을 포함한) 전체 경로 |
| project_title | 현재 프로젝트의 제목 |
| project_units | 프로젝트 좌표계의 단위 |
| project_vertical_crs | Identifier for the vertical Coordinate reference system of the project (e.g., 'EPSG:5703') |
| project_vertical_crs_definition | The full definition of the vertical coordinate reference system of the project |
| project_vertical_crs_description | The description of the vertical coordinate reference system of the project |
| project_vertical_crs_wkt | The WKT (well known text) representation of the vertical coordinate reference system of the project |
| qgis_locale | QGIS 의 현재 언어 |
| qgis_os_name | 현재 운영체제의 이름을 반환 (예: 'windows', 'linux'또는'osx') |
| qgis_platform | QGIS 의 플랫폼 (예: 'desktop'또는'server') |
| qgis_release_name | 현재 QGIS 의 배포명 |
| qgis_short_version | 현재 QGIS 의 버전 문자열 (ShortString 유형) |
| qgis_version | 현재 QGIS 의 버전 문자열 |
| qgis_version_no | 현재 QGIS 의 버전 숫자 |
| row_number | 현재 행의 번호를 저장 |
| snapping_results | 피처 디지타이즈 작업 도중 스냅 작업 결과물에 접근 가능 (피처 추가 시에만 사용할 수 있습니다.) |
| scale_value | 현재 축척 막대 거리 값 |
| selected_file_path | 외부 저장 시스템에 파일을 업로드하는 경우 파일 선택기 위젯에서 선택한 파일 경로 |
| symbol_angle | 피처를 렌더링하는 데 쓰이는 심볼의 각도 (마커 심볼의 경우에만 무결합니다.) |
| symbol_color | 피처를 렌더링하는 데 쓰이는 심볼의 색상 |
| symbol_count | (조판기 범례에 표시되는) 심볼이 표현하는 피처의 개수 |
| symbol_frame | 프레임 번호 (애니메이션화된 심볼 전용) |
| symbol_id | (조판기 범례에 표시되는) 심볼의 내부 ID |
| symbol_label | (사용자가 정의한 라벨 또는 기본 자동 생성된 라벨 가운데 하나로, 조판기 범례에 표시되는) 심볼 용 라벨 |
| symbol_layer_count | 심볼에 있는 심볼 레이어의 총 개수 |
| symbol_layer_index | 현재 심볼 레이어 인덱스 |
| symbol_marker_column | 마커에 대한 열 번호 (포인트 패턴 채우기에 대해서만 무결합니다) |
| symbol_marker_row | 마커에 대한 행 번호 (포인트 패턴 채우기에 대해서만 무결합니다) |
| user_account_name | 현재 사용자의 운영체제 계정명 |
| user_full_name | 현재 사용자의 운영체제 사용자명 |
| value | 현재 값 |
| vector_tile_zoom | 렌더링 중인 맵의 (현재 맵 축척으로부터 파생시킨) 정확한 벡터 타일 확대/축소 수준. 일반적으로 [0, 20] 범위 안의 값입니다. @zoom_level 과는 달리, 이 변수는 정수형 확대/축소 수준 사이의 보간값으로 쓰일 수 있는 부동소수점형 값입니다. |

다음 페이지에 계속

표 9.1 – 이전 페이지에서 계속

| 변수 | 설명 |
|---|---|
| with_variable | 표현식 내부에서 활용하기 위한 변수를 설정할 수 있고, 동일한 값을 반복해서 재계산하는 일을 피할 수 있습니다. |
| zoom_level | 렌더링 중인 맵의 (현재 맵 축척으로부터 파생시킨) 벡터 타일 확대/축소 수준. 일반적으로 [0, 20] 범위 안의 값입니다. |

**다음은 몇몇 예시입니다:**

- 조판기 중심에 있는 맵 항목의 X 좌표를 반환합니다:

```
x( map_get( item_variables( 'map1'), 'map_extent_center' ) )
```

- 현재 레이어에 있는 각 피처 별로, 피처와 중첩하는 공항 피처의 개수를 반환합니다:

```
aggregate( layer:='airport', aggregate:='count', expression:="code",
               filter:=intersects( $geometry, geometry( @parent ) ) )
```

- 라인에서 첫번째로 스냅하는 포인트의 object_id 를 가져옵니다:

```
with_variable(
  'first_snapped_point',
  array_first( @snapping_results ),
  attribute(
    get_feature_by_id(
      map_get( @first_snapped_point, 'layer' ),
      map_get( @first_snapped_point, 'feature_id' )
    ),
    'object_id'
  )
)
```

### 9.2.29 최근 함수

이 그룹은 최근 사용한 함수를 담고 있습니다. 그 맥락에 따라 (피처 선택, 필드 계산기, 일반) 최근 적용된 표현식이 대응하는 목록에 (10 개까지) 추가되며, 가장 최근에 사용한 함수가 맨 앞으로 오도록 정렬합니다. 이 목록을 통해 이전에 사용했던 표현식을 빠르게 찾아서 다시 적용할 수 있습니다.

CHAPTER 10

# 스타일 라이브러리

## 10.1 스타일 관리자

### 10.1.1 스타일 관리자 대화창

*Style Manager* 에서 일반적인 스타일 항목을 관리하고 생성할 수 있습니다. 여기서 일반적인 스타일 항목이란 피처, 레이어, 또는 인쇄 조판기에 심볼을 적용하는 데 사용할 수 있는 심볼, 색상표, 텍스트 서식, 또는 라벨 설정을 말합니다. 이 항목들은 활성화된사용자 프로파일 의 `symbology-style.db` 데이터베이스에 저장되어 해당 프로파일로 열리는 모든 프로젝트 파일이 공유합니다. *Style Manager* 대화창의 내보내기/가져오기 기능 덕분에 스타일 항목들을 다른 사용자들과도 공유할 수 있습니다.

다음 방법들로 모달리스 대화창을 열 수 있습니다:

- *Settings* → *Style Manager*…메뉴로
- 프로젝트 툴바에 있는 Style Manager 버튼으로
- (심볼 환경 설정 작업 또는텍스트 서식 설정 작업 중) 벡터 *Layer Properties* 메뉴로 열리는 대화창의 Style Manager 버튼으로

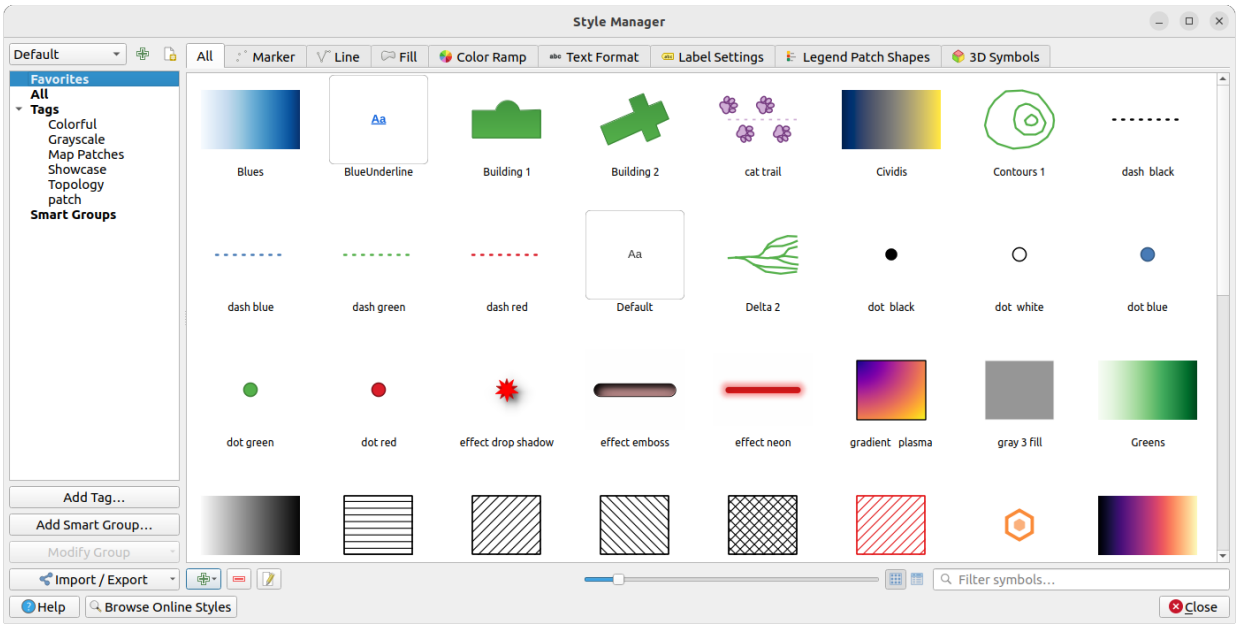



그림 10.1: 스타일 관리자

## 스타일 항목 정리

*Style Manager* 대화창의 좌상단 구석에 있는 드롭다운 메뉴에서 연결하고자 하는 스타일 데이터베이스를 선택할 수 있습니다. *Default* 를 선택하는 경우 사용할 수 있는 모든 기본 스타일 항목들과 사용자가 해당 데이터베이스에 저장한 심볼들을 찾아볼 수 있는 기본 스타일 데이터베이스에 연결할 것입니다. *Project Styles* 를 선택하면 사용자가 해당 데이터베이스에 저장한 심볼들만 찾아볼 수 있는 프로젝트 스타일 데이터베이스에 연결할 것입니다. 사용자가 스타일 데이터베이스를 더 생성할 때마다 드롭다운 메뉴의 목록에 추가될 것입니다. 프로젝트에 기존 스타일 데이터베이스를 추가 Add existing style database to project 하고 새 스타일 데이터베이스를 생성 Create new style database 할 수 있는 옵션도 존재합니다. (스타일 데이터베이스 에서 더 자세한 내용을 볼 수 있습니다.)

각 스타일 데이터베이스 별로, 다음과 같은 여러 카테고리로 정리해서 왼쪽 패널에 목록으로 만들 수 있습니다:

- **Favorites**: 항목을 환경 설정할 때 기본적으로 표시되는 카테고리로, 확장할 수 있는 항목 집합을 보여줍니다.
- **All**: 활성화된 유형의 사용할 수 있는 모든 항목들의 목록을 표시합니다.
- **Tags**: 항목을 식별하는 데 사용할 수 있는 라벨 목록을 표시합니다. 항목 하나에 1 개 이상의 태그를 붙일 수 있습니다. 목록에서 태그를 선택하면 탭을 업데이트해서 선택한 태그를 가지고 있는 항목만 표시합니다. 다음에 항목 집합에 붙일 수 있는 새 태그를 생성하려면 *Add Tag*…버튼을 사용하거나 모든 태그 컨텍스트 메뉴에서 *Add Tag*…를 선택하면 됩니다.
- **Smart Group**: 스마트 그룹은 조건 집합에 따라 그룹의 심볼을 동적으로 가져옵니다. (그림 10.2 를 참조하세요.) 스마트 그룹을 생성하려면 *Add Smart Group*…버튼을 클릭하십시오. 대화창에서 선택할 항목을 필터링할 (특정 태그가 붙어 있거나, 이름에 특정 문자열이 있거나 등등) 표현식을 입력할 수 있습니다. 입력한 조건 (들) 을 만족하는 모든 심볼, 색상표, 텍스트 서식, 또는 라벨 설정을 자동적으로 스마트 그룹에 추가합니다.

태그와 스마트 그룹은 상호 배타적이지 않습니다. 사용자의 스타일 요소들을 정리하는 서로 다른 두 가지 방법일 뿐입니다. 입력한 제한 조건을 만족하는 항목들을 자동적으로 가져오는 스마트 그룹과는 달리, 태그는 사용자가 직접 입력합니다. 이런 카테고리를 편집하려면:

- 항목을 선택하고 오른쪽 클릭해서 *Add to Tag* 를 선택한 다음 태그 이름을 선택하거나 새 태그를 선택하십시오.

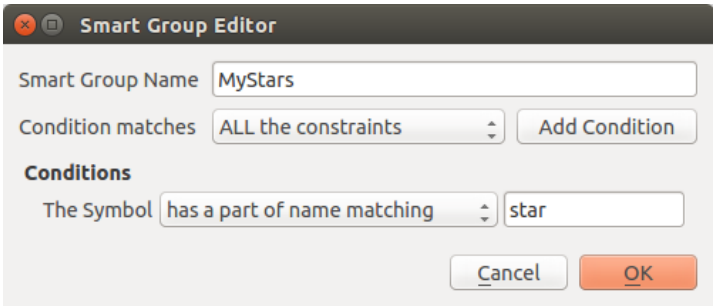



그림 10.2: 스마트 그룹 생성하기

- 또는 태그를 선택하고 *Modify group…* → *Attach Selected Tag to Symbols* 를 클릭하십시오. 각 항목 옆에 사용자가 선택하거나 선택 해제할 수 있는 체크박스가 나타납니다. 선택 작업을 완료하면 *Modify group…* → *Finish Tagging* 을 클릭하십시오.
- 아니면 스마트 그룹을 선택하고 *Modify group…* → *Edit smart group…*을 클릭한 다음 *Smart Group Editor* 대화창에서 새 제한 조건 집합을 환경 설정하십시오. 스마트 그룹의 컨텍스트 메뉴에서도 이 옵션을 사용할 수 있습니다.

태그 또는 스마트 그룹을 제거하려면 해당 항목을 오른쪽 클릭한 다음 *Remove* 버튼을 누르십시오. 이렇게 해도 카테고리에 모여 있던 항목은 삭제되지 않는다는 사실을 기억하세요.

*Style Manager* 대화창의 중심에는 각 항목의 미리보기를 탭으로 정리해 보여주는 프레임이 있습니다:

- *All* 탭은 미리 정의된 색상표와 텍스트 서식은 물론 점형, 선형, 면형 심볼과 라벨 설정의 전체 집합을 표시합니다.
- *Marker* 탭은 점형 심볼만 표시합니다.
- *Line* 탭은 선형 심볼만 표시합니다.
- *Fill* 탭은 면형 심볼만 표시합니다.
- *Color ramp* 탭은 색상표를 표시합니다.
- *Text format* 탭에서는 텍스트의 글꼴, 색상, 버퍼, 그림자 및 배경을 저장하는텍스트 서식 을 관리합니다. (예를 들어 조판기에서 사용할 수 있는 라벨 설정에서 서식 작업 부분 전체를 관리할 수 있습니다.)
- *Label settings* 탭은 라벨 배치, 우선 순위, 호출 (callout), 렌더링 등과 같은 레이어 유형에 특화된 몇몇 설정과 텍스트 서식을 포함하는라벨 설정 을 관리합니다.
- *Legend Patch Shapes* 탭은 *Marker*, *Line* 및 *Fill* 도형을 포함하는 사용자 지정 범례 패치 형태를 관리합니다.
- *3D Symbols* 탭은*3* 차원 맵 뷰 에서 렌더링될 피처를 위해*3* 차원 속성 (돌출 (extrusion), 음영, 표고 등등)으로 심볼을 환경 설정합니다.

하단 오른쪽에 있는 *Icon View* 또는 *List View* 에서 스타일을 배열할 수 있습니다. 두 뷰 모두 툴팁이 스타일의 더 큰 인스턴스를 표시합니다. 아이콘 왼쪽에 있는 섬네일 크기 슬라이드 바를 이용해서 더 나은 심볼 미리보기를 위해 대화창에서의 실제 섬네일 크기를 조정할 수 있답니다!

### 항목 추가, 편집, 제거

앞에서 설명했듯이, 스타일 요소는 활성화된 카테고리에 따라 (태그, 스마트 그룹, 즐겨찾기 등등) 그 내용이 달라지는 여러 탭에 정리되어 있습니다. 활성화된 탭에서 다음과 같은 작업을 할 수 있습니다:

- 새 항목 추가: Add item 버튼을 클릭한 다음심볼, 색상표, 또는텍스트 서식 및 라벨 작성기의 설명을 따라서 항목을 환경 설정합니다.
- 기존 항목 수정: 항목을 선택하고 Edit item 버튼을 클릭한 다음 앞에서 설명한 대로 환경 설정합니다.
- 기존 항목 삭제: 더 이상 필요하지 않은 요소를 삭제하려면, 해당 항목을 선택하고 Remove item 버튼을 클릭하십시오. (오른쪽 클릭으로도 가능합니다.) 로컬 데이터베이스에서 해당 항목을 삭제할 것입니다.

*All* 탭에서 모든 항목 유형에 대해 이 옵션들을 사용할 수 있다는 사실을 기억하십시오.

여러 항목들을 선택한 다음 오른쪽 클릭하면:

- *Add to Favorites*: 즐겨찾기에 추가합니다.
- *Remove from Favorites*: 즐겨찾기에서 제거합니다.
- *Add to Tag* : 적절한 태그를 선택하거나 새 태그를 생성해서 항목에 추가합니다. 현재 붙어 있는 태그는 체크되어 있습니다.
- *Clear Tags*: 심볼에서 모든 태그를 떼어냅니다.
- *Remove Item(s)*: 항목 (들) 을 제거합니다.
- *Edit Item*: 사용자가 오른쪽 클릭한 항목을 편집하고 변경 사항을 적용합니다.
- *Copy Item*: 항목을 클립보드로 복사합니다.
- *Paste Item* ⋯: 클립보드에 있는 항목을 스타일 관리자의 카테고리 가운데 하나 또는 QGIS 의 (심볼 또는 색상 버튼) 어딘가로 붙여넣습니다.
- *Export Selected Symbol(s) as PNG*⋯: 선택한 심볼 (들) 을 PNG 파일로 내보냅니다. (심볼에 대해서만 사용할 수 있습니다.)
- *Export Selected Symbol(s) as SVG*⋯: 선택한 심볼 (들) 을 SVG 파일로 내보냅니다. (심볼에 대해서만 사용할 수 있습니다.)

### 스타일 항목 공유

스타일 관리자 대화창 좌하단에 있는 *Import/Export* 도구는 심볼, 색상표, 텍스트 서식 및 라벨 설정을 쉽게 다른 사용자들과 공유할 수 있는 옵션을 제공합니다. 항목을 오른쪽 클릭해도 이 옵션들을 사용할 수 있습니다.

### 항목 내보내기

항목 집합을 `.XML` 파일로 내보낼 수 있습니다:

1. *Import/Export* 드롭다운 메뉴를 펼친 다음 *Export Item(s)*⋯을 선택하십시오.
2. 통합하고자 하는 항목들을 선택하십시오. 마우스로 선택하거나 이전에 설정한 태그 또는 그룹을 사용해서 선택 집합을 만들 수 있습니다.
3. 준비가 끝나면 *Export* 를 클릭하십시오. 저장한 파일의 위치를 나타내는 메시지가 표시될 것입니다. XML 포맷은 선택한 모든 항목들을 담고 있는 단일 파일을 생성합니다. 이후 또다른 사용자의 스타일 라이브러리로 이 파일을 가져올 수 있습니다.

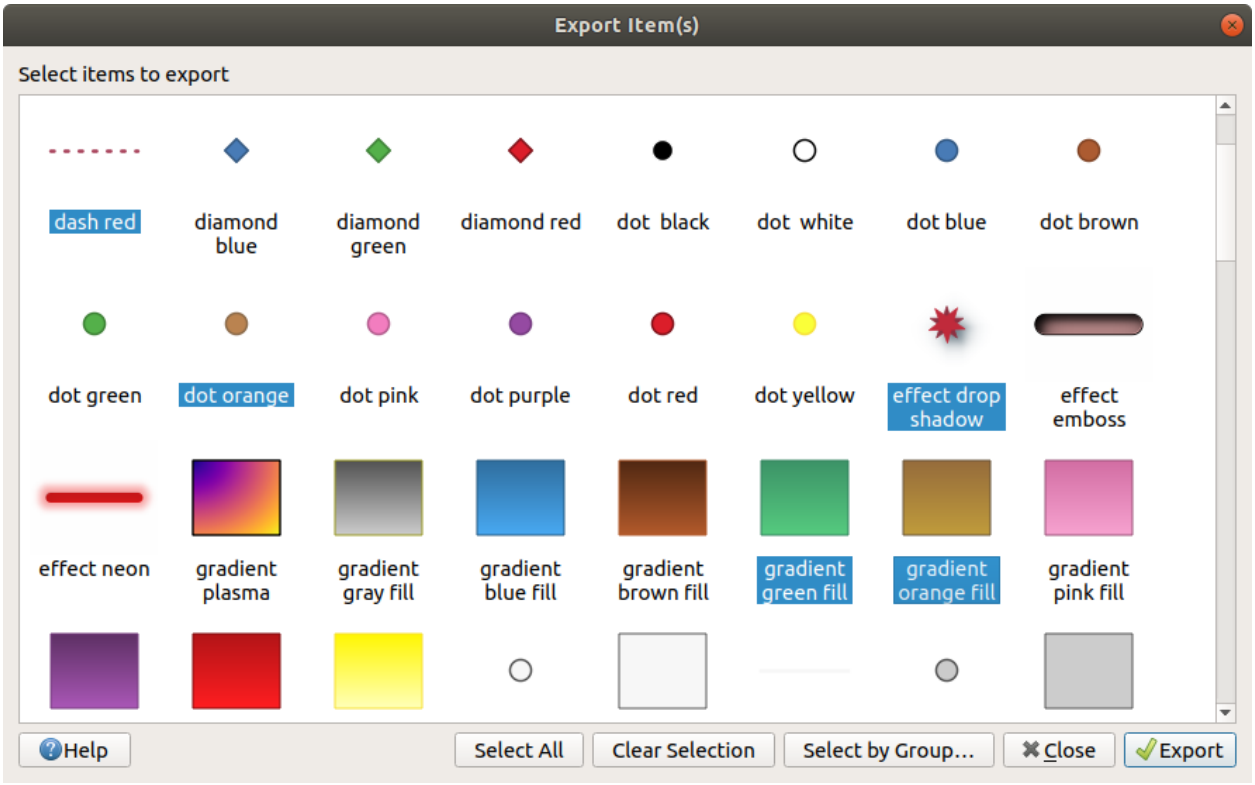



그림 10.3: 스타일 항목 내보내기

심볼을 선택한 경우, 선택한 심볼들을 `.PNG` 또는 `.SVG` 이미지 파일로도 내보낼 수 있습니다. `.PNG` 또는 `.SVG` 로 내보내면 (다른 스타일 항목 유형은 이 두 파일로 내보낼 수 없습니다) 지정한 폴더에 선택한 각 심볼의 파일을 생성합니다. 또다른 사용자의 *Settings* → *Options* → *System* 메뉴에서 SVG 폴더를 *SVG paths* 에 추가하면, 그 사용자는 이 모든 심볼에 직접 접근할 수 있습니다.

### 항목 가져오기

새 항목들을 가져와서 사용자의 스타일 라이브러리를 확장할 수 있습니다:

1. *Import/Export* 드롭다운 메뉴를 펼친 다음 대화창의 좌하단에 있는 *Import Item(s)* 을 선택하십시오.
2. 새로 열리는 대화창에서 스타일 항목의 소스를 정의하십시오. (사용자 디스크에 있는 `.xml` 파일이 될 수도 있고, 인터넷 URL 이 될 수도 있습니다.)
3. *Add to favorites*: 가져올 항목을 즐겨찾기에 추가할지 여부를 설정하십시오.
4. *Do not import embedded tags*: 이 옵션을 체크하면 항목에 붙어 있는 태그를 가져오지 않습니다.
5. *Additional tag(s)*: 새 항목에 추가할 태그의 이름을 정의합니다.
6. 미리보기에서 사용자 라이브러리에 추가하고자 하는 심볼을 선택하십시오.
7. 마지막으로 *Import* 를 클릭하십시오.

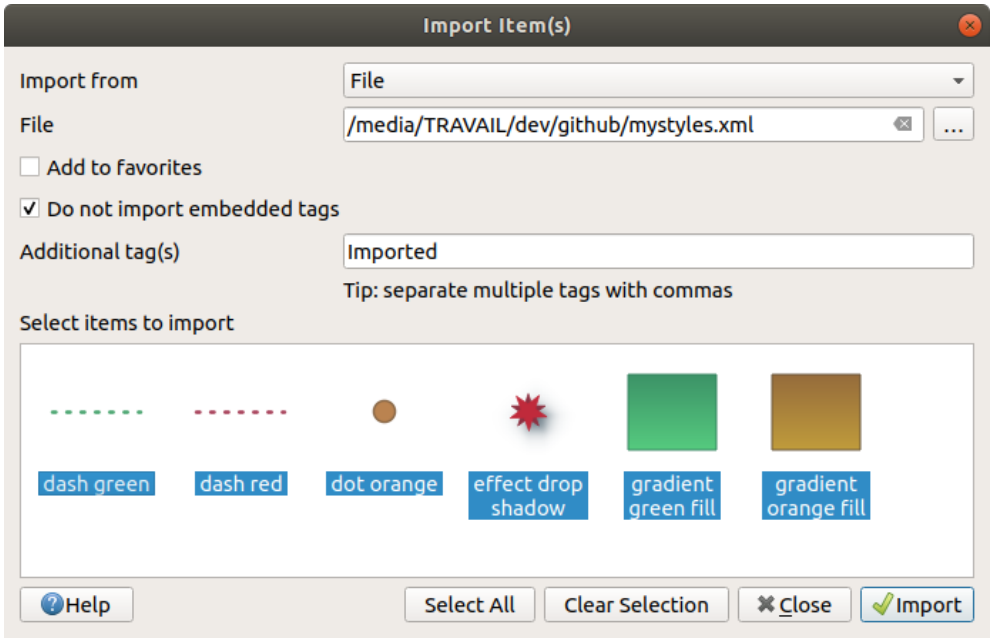



그림 10.4: 스타일 항목 가져오기

### 탐색기 패널 활용하기

*Browser* 패널에서 활성화된 사용자 프로파일 스타일 데이터베이스로 스타일 항목을 곧바로 가져올 수도 있습니다:

1. 탐색기에서 스타일 `.xml` 파일을 선택하십시오.
2. 맵 캔버스로 드래그 & 드롭하거나 맵 캔버스를 오른쪽 클릭한 다음 *Import Style*…을 선택하십시오.
3. 항목 가져오기 를 따라서 *Import Items* 대화창을 작성하십시오.
4. *Import* 버튼을 클릭하면 스타일 데이터베이스에 선택한 스타일 항목이 추가됩니다.

탐색기에서 스타일 파일을 더블클릭하면 파일이 담고 있는 항목을 보여주는 *Style Manager* 대화창이 열립니다. 이 항목들 가운데 선택한 다음 *Copy to Default Style*…버튼을 클릭하면 활성화된 스타일 데이터베이스로 선택한 항목을 가져옵니다. 이렇게 가져온 항목에 태그를 붙일 수 있습니다. 탐색기에서 스타일 파일을 오른쪽 클릭한 다음 *Open Style*…메뉴를 선택해도 동일한 작업을 할 수 있습니다.

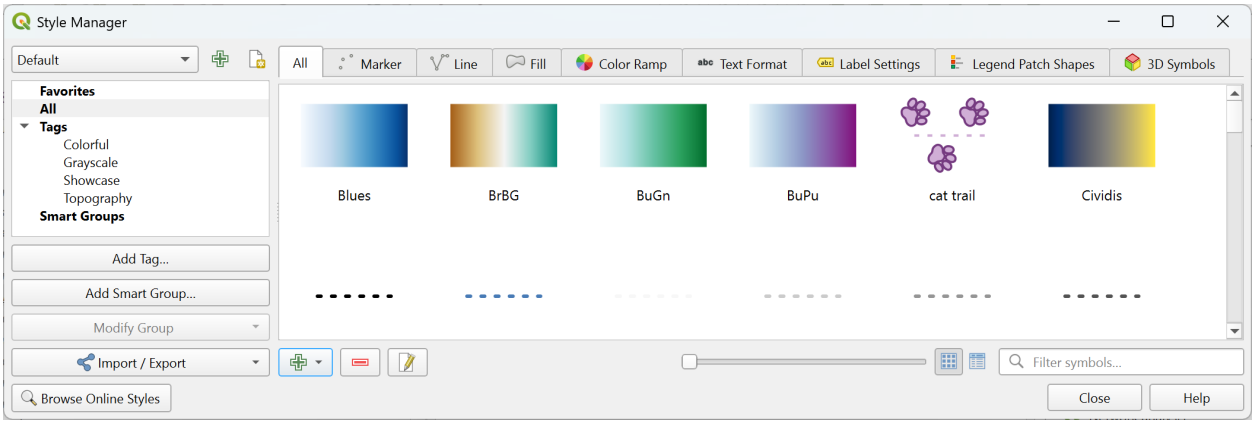



그림 10.5: 스타일 항목 파일 열기

이 대화창에서 단일 심볼을 `.PNG` 또는 `.SVG` 파일로 내보낼 수도 있습니다.

#### 온라인 저장소 이용하기

The QGIS project maintains a repository with a collection of styles shared by QGIS users. This is available at https://hub.qgis.org/styles and can be accessed from the *Style Manager* dialog, pressing the  *Browse Online Styles* button at the bottom.

이 저장소에서 다음과 같은 작업을 할 수 있습니다:

1. 스타일 유형 또는 이름을 바탕으로 모든 스타일 항목을 탐색하고 검색할 수 있습니다.
2. 스타일 파일을 다운로드한 다음 압축 해제할 수 있습니다.
3. 앞에서 설명했던 모든 가져오기 메소드를 이용해서 QGIS 에 있는 사용자 스타일 데이터베이스에 `.xml` 기반 파일을 불러올 수 있습니다.

### 10.1.2 색상표 설정

*Style Manager* 대화창의 색상표 탭은 사용자가 왼쪽 패널에서 선택한 카테고리를 기반으로 한 여러 색상표의 미리보기를 볼 수 있게 해줍니다.

사용자 지정 색상표를 생성하려면, 색상표 탭으로 가서  Add item 버튼을 클릭한 다음 드롭다운 목록에서 다음 색상표 유형 가운데 하나를 선택하십시오:

- *Gradient*: 시작 색상과 끝 색상을 지정해서 **연속적 (continuous)** 이거나 **불연속적 (discrete)** 일 수 있는 색상표를 생성합니다. 색상표 미리보기를 더블클릭해서 사용자가 원하는 만큼 많은 중간 색상 지점 (intermediate color stop) 을 추가할 수 있습니다. 색상 지점 표시자를 클릭하면 *Gradient stop* 부분에서 다음과 같은 작업을 할 수 있습니다:
  - 색상표 시작 지점으로부터 색상 지점의 *Relative position* 을 조정할 수 있습니다. 마우스 커서 또는 화살표 키를 눌러 표시자를 드래그할 수도 있습니다. (더 크게 움직이려면 `Shift` 키를 누른 채 마우스 또는 화살표 키를 사용하십시오.)
  - 색상들 사이를 보간하는 경우 사용할 색상 모델을 *RGB*, *HSL* 또는 *HSV* 가운데 하나로 지정할 수 있습니다. 어떤 상황에서는 이 옵션이 낮은 채도의 중간 톤을 억제해서 시각적으로 좀 더 보기 좋은 그레이디언트를 생성하는 데 도움이 될 수 있습니다.
  - *HSL* 또는 *HSV* 색상 사양의 **색상 (hue)** 구성 요소에 대해 보간법이 따라야 할 방향을 *Clockwise* 또는 *Counterclockwise* 가운데 하나로 설정할 수 있습니다.
  - 색상 속성 을 설정할 수 있습니다.
  - *Delete stop* 버튼 또는 `DEL` 키를 눌러서 색상 지점을 제거할 수 있습니다.

  *Plots* 부분은 색상 지점의 위치 또는 불투명도 및 HSL 구성 요소를 변경해서 색상표를 설계할 수 있는 또다른 그래픽적 방식을 제공합니다.

  > **힌트**
  >
  > 색상표에서 그레이디언트 색상표 미리보기로 어떤 색상을 드래그 & 드롭하면 새로운 색상 지점이 추가됩니다.

- *Color presets*: 사용자가 선택한 색상 목록으로 이루어진 색상표를 생성할 수 있습니다.

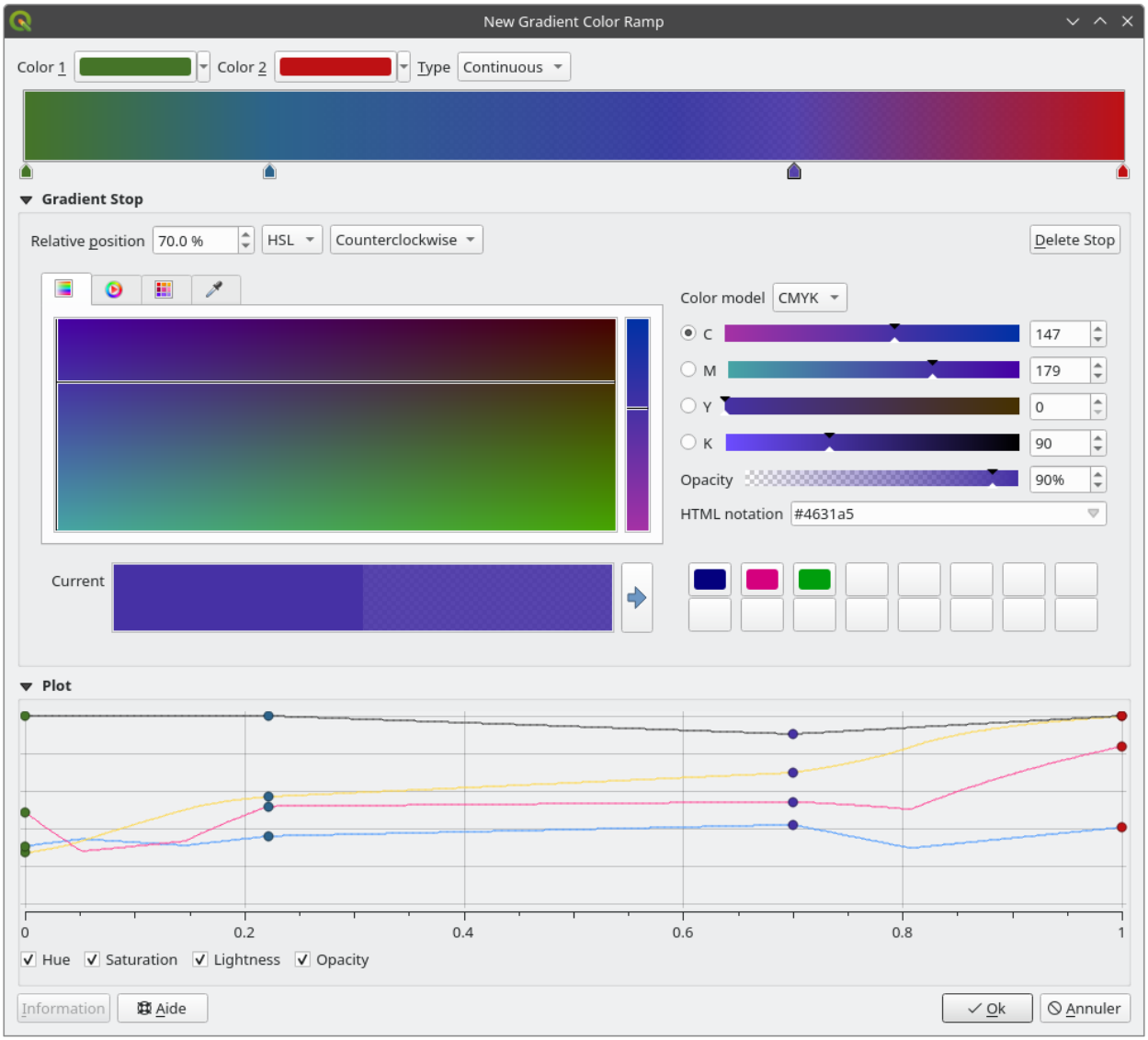



그림 10.6: 복수의 색상 지점을 지정한 사용자 지정 그레이디언트 색상표의 예시

- *Random*: *Hue* 색상, *Saturation* 채도, *Value* 명도 값의 범위 그리고 *Opacity* 투명도와 색상 개수 (*Classes*) 를 바탕으로 랜덤한 색상 집합을 생성합니다.
- *Catalog: ColorBrewer*: 색상표에서 색상 개수를 사용자 지정할 수 있는, 사전 정의된 비연속 색상그레이디언트의 집합입니다.
- *Catalog: cpt-city*: *Save as standard gradient* 메뉴를 사용해서 로컬에 저장되는 색상 그레이디언트의 전체 카탈로그를 엽니다. *cpt-city* 옵션은'별도의 설정 없이 바로 사용할 수 있는 (out of the box)'테마를 포함한 수백 개의 테마를 가진 새 대화창을 엽니다.

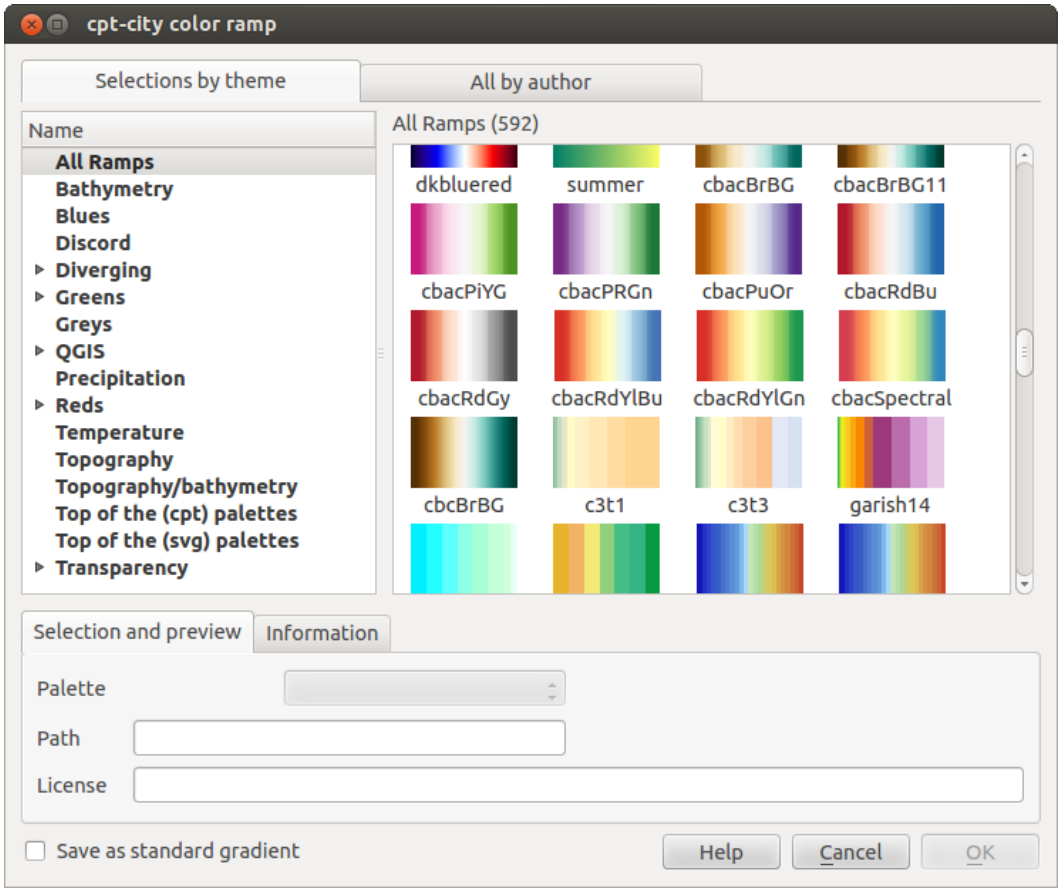



그림 10.7: 색상표 수백 개를 가진 cpt-city 대화창

### 10.1.3 범례 패치 형태 생성하기

새 범례 패치 형태를 생성하려면, *Legend Patch Shapes* 탭으로 가서 Add item 버튼을 누르십시오. 도형 유형을 선택할 수 있는 드롭다운 목록이 열립니다:

- *Marker Legend Patch Shape*…: 포인트 도형 유형을 사용합니다.
- *Line Legend Patch Shape*…: 라인 도형 유형을 사용합니다.
- *Fill Legend Patch Shape*…: 폴리곤 도형 유형을 사용합니다.

세 옵션 모두 동일한 대화창을 엽니다.

선택한 도형 유형에 따라 형태 유형과 표시되는 범례 패치 형태만 달라집니다. 다음 옵션을 사용할 수 있습니다:

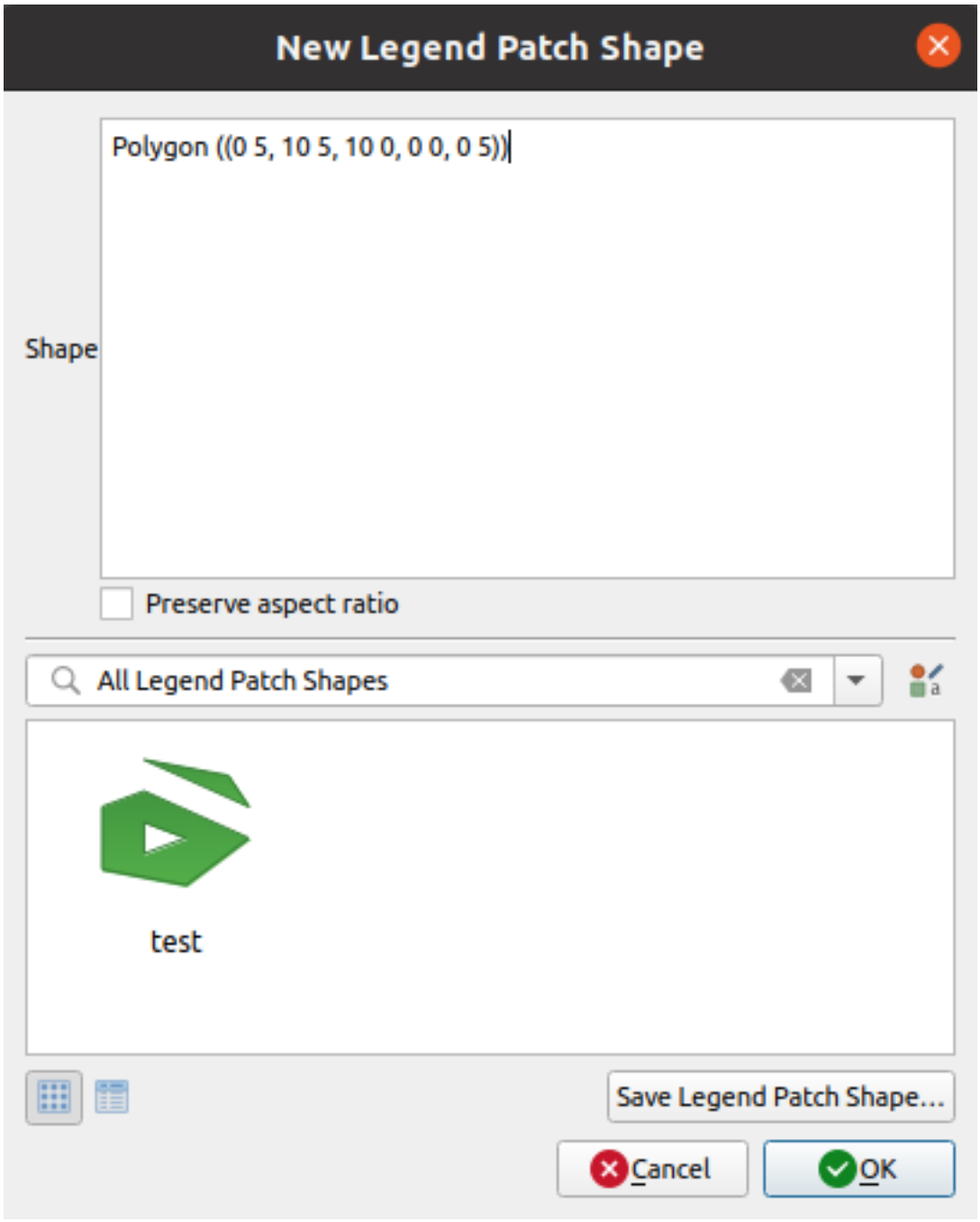



그림 10.8: 새 범례 패치 형태 생성하기

- *Shape*: 범례 패치 형태의 형태를 WKT 문자열로 정의합니다. 단일 도형 및 다중 부분 도형을 사용할 수 있지만, 도형집합 (GeometryCollection) 은 사용할 수 없습니다.
- *Preserve aspect ratio*: 가로 세로 비율을 유지합니다.
- 사용할 수 있는 범례 패치 형태를 태그로 필터링한 *Icon View* 또는 *List View*

새 형태를 정의했다면 *Save Legend Patch Shape*…또는 *OK* 를 클릭하면 됩니다. 둘 다 동일한 대화창을 열 것입니다.

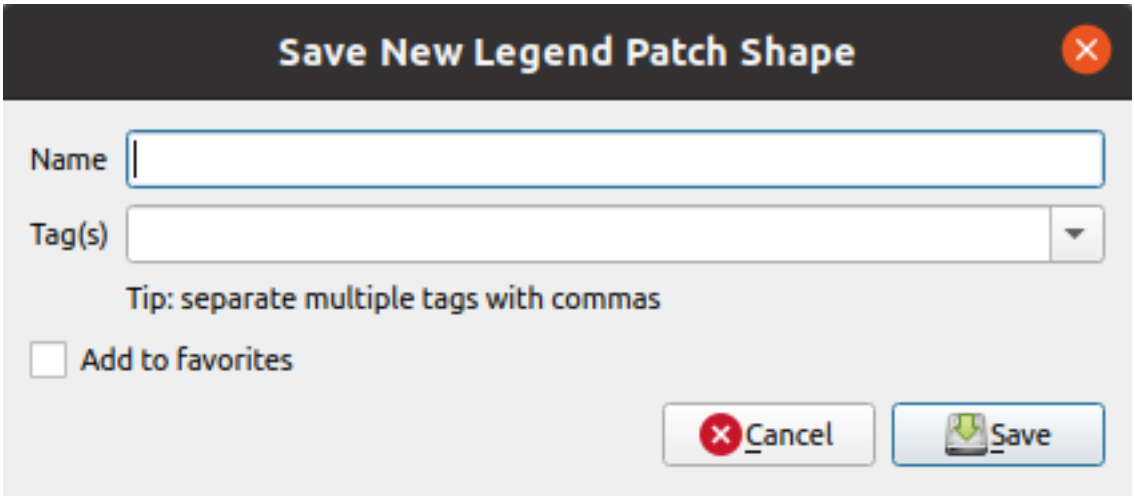



그림 10.9: 새 범례 패치 형태 저장하기

이 대화창에서 이름, 형태를 설명하는 태그, 그리고 즐겨찾기에 추가해야 할지 여부를 선택해야 합니다.

*Save* 버튼을 클릭하면 목록에 형태를 추가한 다음 계속 새 형태를 생성할 수 있도록 *New Legend Patch Shape* 대화창으로 돌아갑니다.

## 10.2 심볼 선택기

심볼 선택기는 심볼을 디자인하기 위한 주 대화창입니다. 마커 (marker), 라인 (line) 또는 채우기 (fill) 심볼을 생성하거나 편집할 수 있습니다.

다음은 심볼 선택기 대화창을 구성하는 주 요소 두 가지입니다:

- 심볼 트리: 이후에 결합해서 새 전체 수준 심볼을 형성하는 심볼 레이어를 표시합니다.
- 설정: 트리에서 선택한 심볼 레이어의 환경을 설정합니다.

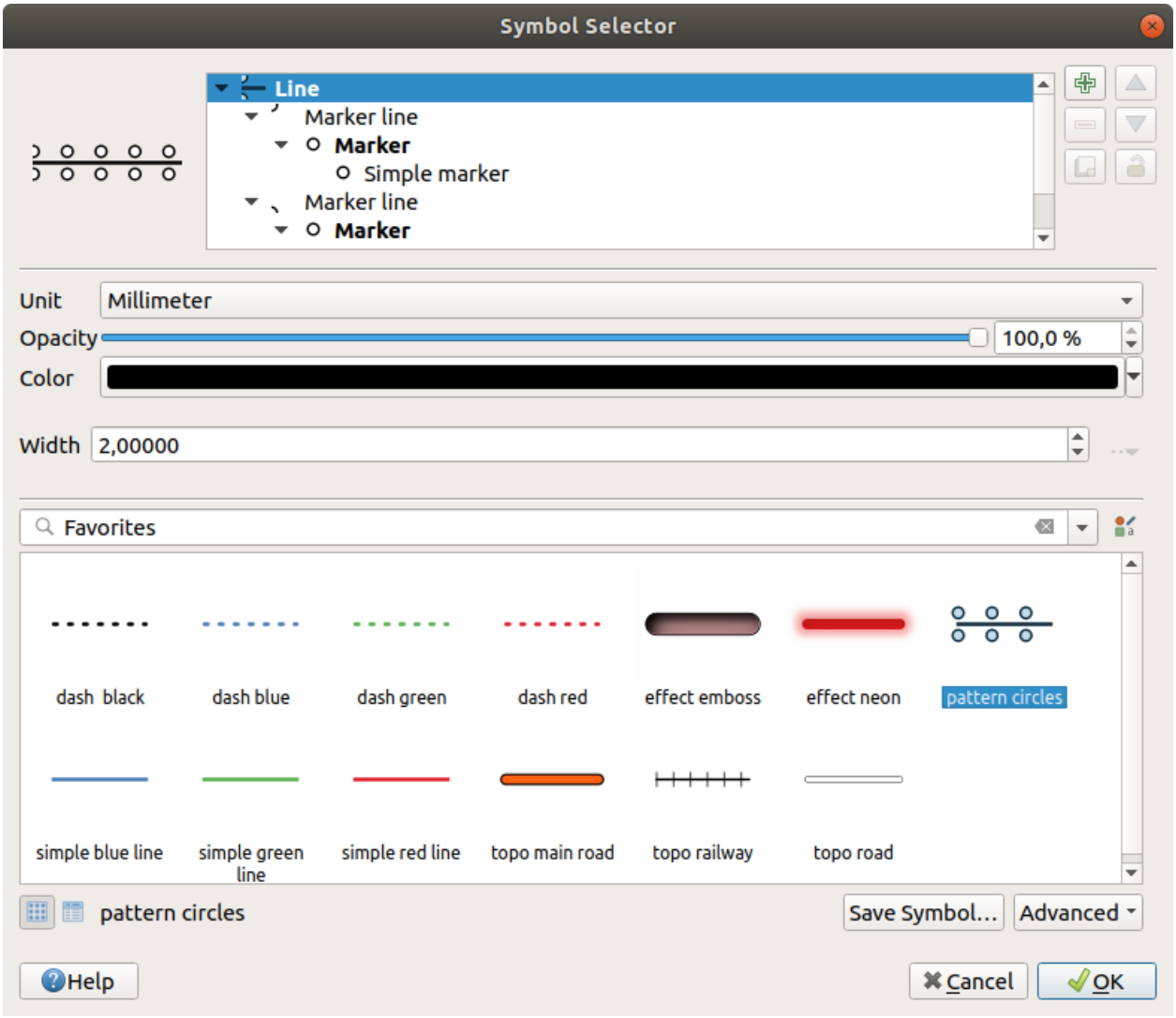



그림 10.10: 라인 심볼 디자인하기

### 10.2.1 심볼 레이어 트리

심볼은 여러 *Symbol layers* 로 이루어질 수 있습니다. 심볼 트리는 이후에 결합해서 새 전체 수준 심볼을 형성하는 이 심볼 레이어들의 적층 (overlay) 을 표시합니다. 게다가, 심볼 속성을 변경하자마자 동적인 심볼 표현이 업데이트됩니다.

심볼 트리 항목에서 선택한 수준에 따라, 트리 관리 작업을 도와주는 다양한 도구가 활성화되어 사용할 수 있습니다:

-  Add new symbol layer: 원하는 만큼 심볼을 겹칠 수 있습니다.
-  Remove the selected symbol layer: 선택한 심볼 레이어를 제거합니다.
-  Lock colors of symbol layer: 잠긴 색상은 사용자가 전체 (또는 그 이상) 심볼 수준에서 색상을 변경하더라도 영향을 받지 않습니다.
-  Duplicate a (group of) symbol layer(s): 심볼 레이어 (들의 그룹) 를 복제합니다.
- 심볼 레이어를 위 또는 아래로 이동시킬 수 있습니다.

### 10.2.2 심볼 환경 설정

QGIS 는 심볼, 그 다음 심볼 레이어라는 두 단계를 거쳐 심볼의 환경을 설정합니다.

#### 심볼

트리의 최상 수준에서, 심볼은 레이어 도형에 따라 **마커**, **라인** 또는 **채우기** 유형이 될 수 있습니다. 각 심볼은 하나 이상의 (어느 유형이든) 심볼 또는 심볼 레이어를 내장할 수 있습니다.

전체 수준 심볼에 적용되는 다음 몇몇 파라미터를 설정할 수 있습니다:

- *Unit*: **밀리미터**, **포인트**, **픽셀**, **축척 미터**, **맵 단위** 또는 **인치** 가운데 하나를 선택할 수 있습니다. (자세한 내용은단위 선택기 를 참조하세요.)
- *Opacity*: 투명도를 설정할 수 있습니다.
- *Color*: 사용자가 이 파라미터를 변경하면, 잠기지 않은 모든 하위 심볼 색상에 그 색상값을 반영합니다.
- *Size* 및 *Rotation*: 마커 심볼 전용의 크기 및 기울기 파라미터입니다.
- *Width*: 라인 심볼 전용의 선의 폭 파라미터입니다.

> **팁**
>
> 심볼 수준에서 심볼이 내장한 모든심볼 레이어 차원의 크기를 비례에 맞게 재조정하려면 (마커 심볼 전용) *Size* 나 (라인 심볼 전용) *Width* 속성을 사용하십시오.

> **참고**
>
> 스타일 관리자 대화창에서 심볼을 설정할 때 폭, 크기 또는 기울기 파라미터 옆에 있는데이터 정의 무시 버튼은 비활성화돼 있습니다. 심볼이 레이어와 연결된 다음에야, 이 버튼으로비례 또는 다변량 분석 렌더링을 생성할 수 있습니다.

- 심볼 라이브러리 미리보기: 동일 유형의 심볼들이 표시되며, 위에 있는 편집 가능한 드롭다운 목록을 통해 양식 없는 텍스트 또는카테고리 로 동일 유형의 심볼들을 필터링할 수 있습니다. 또 Style Manager 버튼을 사용해서 심볼 목록을 업데이트하고 스타일 관리자라는 이름의 유래가 된 대화창을 열 수 있습니다. 이 대화창에서스타일 관리자 절에서 설명하는 모든 기능을 사용할 수 있습니다.

  심볼은 다음 가운데 한 가지 방법으로 표시됩니다:

  - 프레임 하단에 있는 List View 버튼을 통한 (썸네일, 이름 및 관련 태그를 함께 표시하는) 아이콘 목록
  - 또는 Icon View 버튼을 통한 아이콘 미리보기

- *Save Symbol* 버튼을 클릭하면 *Save New Symbol* 대화창이 열립니다. 이 대화창에서 편집 중인 심볼을 *Destination* 어디에 추가할지 선택할 수 있고, *Name* 이름을 지정할 수 있으며, *Tag(s)* 태그를 추가할 수 있습니다. 사용자의 새 심볼을 Add to favorites 즐겨찾기에 추가할 수 있는 옵션도 있습니다.

- *Advanced* 옵션을 통해 다음 작업을 할 수 있습니다:

  - *Clip features to canvas extent*: 라인 및 채우기 심볼 전용 설정으로, 캔버스 범위를 따라 피처를 잘라냅니다.
  - *Force right-hand rule orientation*: 채우기 심볼 전용으로, 채우기 심볼의 고리 방향을 표준"오른손 법칙"을 따르도록 강제 렌더링할 수 있습니다. (예를 들어 외부 고리가 시계 방향, 내부 고리가 모두 반시계 방향인 폴리곤인 경우에 말이죠.)

    이 방향 고정은 렌더링 중에만 적용되고, 원본 피처 도형을 변경하지는 않습니다. 이렇게 설정하면 렌더링 중인 데이터셋과 개별 피처의 고리 방향과 상관없이, 일관된 외형을 가진 채우기 심볼을 생성할 수 있습니다.
  - for marker symbols, the *Buffer settings*⋯ enables addition of a halo effect around the marker symbol in order to make it more readable against different backgrounds. The buffer is calculated and drawn using the shape of ALL the symbol layers in the marker. You can adjust the *Size*, *Join style* and *Symbol* of display.
  - for all symbol types, enable and configure a distance for the map canvas *Extent buffer*, or the *Animation settings*.
  - Depending on the *symbology* of the layer the symbol is being applied to, additional settings are available in the *Advanced* menu:
    * 심볼 수준⋯: 심볼 렌더링 순서를 정의합니다.
    * 데이터 정의 크기 범례
    * *Match to Saved Symbols*⋯및 *Match to Symbols from File*⋯: 자동적으로범주에 심볼을 할당 합니다.

### 심볼 레이어

트리의 낮은 수준에서, 심볼 레이어를 사용자 지정할 수 있습니다. 사용할 수 있는 심볼 레이어 유형은 상위 심볼 유형을 따릅니다. 심볼 렌더링을 향상시키기 위해 심볼 레이어에 그리기 효과 를 적용할 수 있습니다.

모든 심볼 레이어 유형의 모든 옵션을 다 설명하는 것이 불가능하기 때문에, 특별하고 중요한 옵션들만 설명하겠습니다.

### 공통 파라미터

심볼 레이어가 마커, 라인, 채우기 하위 유형인지 상관없이, 심볼 레이어를 작성하는 데 다음 공통 옵션 및 위젯을 쓸 수 있습니다:

- 색상 선택기 위젯: 색상을 쉽게 조정할 수 있습니다.
- *Units*: **밀리미터**, **포인트**, **픽셀**, **축척 미터**, **맵 단위** 또는 **인치** 가운데 하나를 선택할 수 있습니다. (자세한 내용은단위 선택기 를 참조하세요.)
- Data-defined override 위젯: 거의 모든 옵션들 옆에 있으며, 각 심볼을 사용자 지정하는 능력을 확장합니다. (자세한 내용은데이터 정의 무시 설정 을 참조하세요.)
- *Enable symbol layer*: 이 옵션은 심볼 레이어의 가시성을 제어합니다. 심볼 레이어를 비활성화하면 심볼 렌더링 시 그리지 않지만 심볼 내에 저장합니다. 심볼 레이어를 숨길 수 있다는 사실은 사용자 심볼을 디자인하는 데 유용합니다. 디자인 도중 심볼 레이어를 제거하지 않아도 되기 때문입니다. 이때 데이터 정의 무시를 사용하면 (예를 들어 피처 속성을 이용한) 표현식을 기반으로 서로 다른 심볼 레이어들을 숨기거나 표시할 수 있습니다.
- *Draw effects* 버튼: 효과 렌더링 을 이용할 수 있습니다.

> **참고**
>
> 다음에 설명할 내용이 심볼 레이어의 유형은 피처 도형에 종속된다고 가정하고 있긴 하지만, 심볼 레이어를 다른 심볼 레이어 안에 삽입할 수 있다는 사실을 기억하십시오. 이 경우, 낮은 수준의 심볼 레이어 파라미터 (위치, 오프셋 등등) 가 피처 도형 자체가 아니라 높은 수준의 심볼에 종속될 수도 있습니다.

### 마커 심볼

포인트 도형 피처에 어울리는 마커 심볼은 다음과 같은 여러 *Symbol layer types* 를 가지고 있습니다:

표 10.1: List of marker symbol layer types

| | | |
|---|---|---|
| *Simple marker* | *Animated marker* | *Ellipse marker* |
| *Filled marker* | *Font marker* | *Geometry generator* |
| *Mask* | *Raster image marker* | *Vector Field marker* |
| *SVG marker* | | |

- **단순 마커** (기본값)

  단순 마커 심볼 레이어 유형은 다음과 같은 속성을 가지고 있습니다:

  - *Size*: 크기를 다양한 지원 단위로 설정할 수 있습니다.
  - *Fill color*: 채우기 색상을 설정할 수 있습니다.
  - *Stroke color*, *Stroke style*, *Stroke size*: 사전 정의된 목록에서 획 (stroke) 색상 및 스타일을 선택하고, 획의 굵기도 설정할 수 있습니다.
  - *Join style*: 결합 스타일을 **Bevel**, **Miter** 또는 **Round** 가운데 하나로 설정할 수 있습니다.
  - *Cap style*: 선끝 스타일을 **Square**, **Flat** 또는 **Round** 가운데 하나로 설정할 수 있습니다.
  - *Rotation*: 마커의 기울기를 설정할 수 있습니다.
  - *Offset*: 피처로부터 *X*, *Y* 방향으로 오프셋을 설정할 수 있습니다.

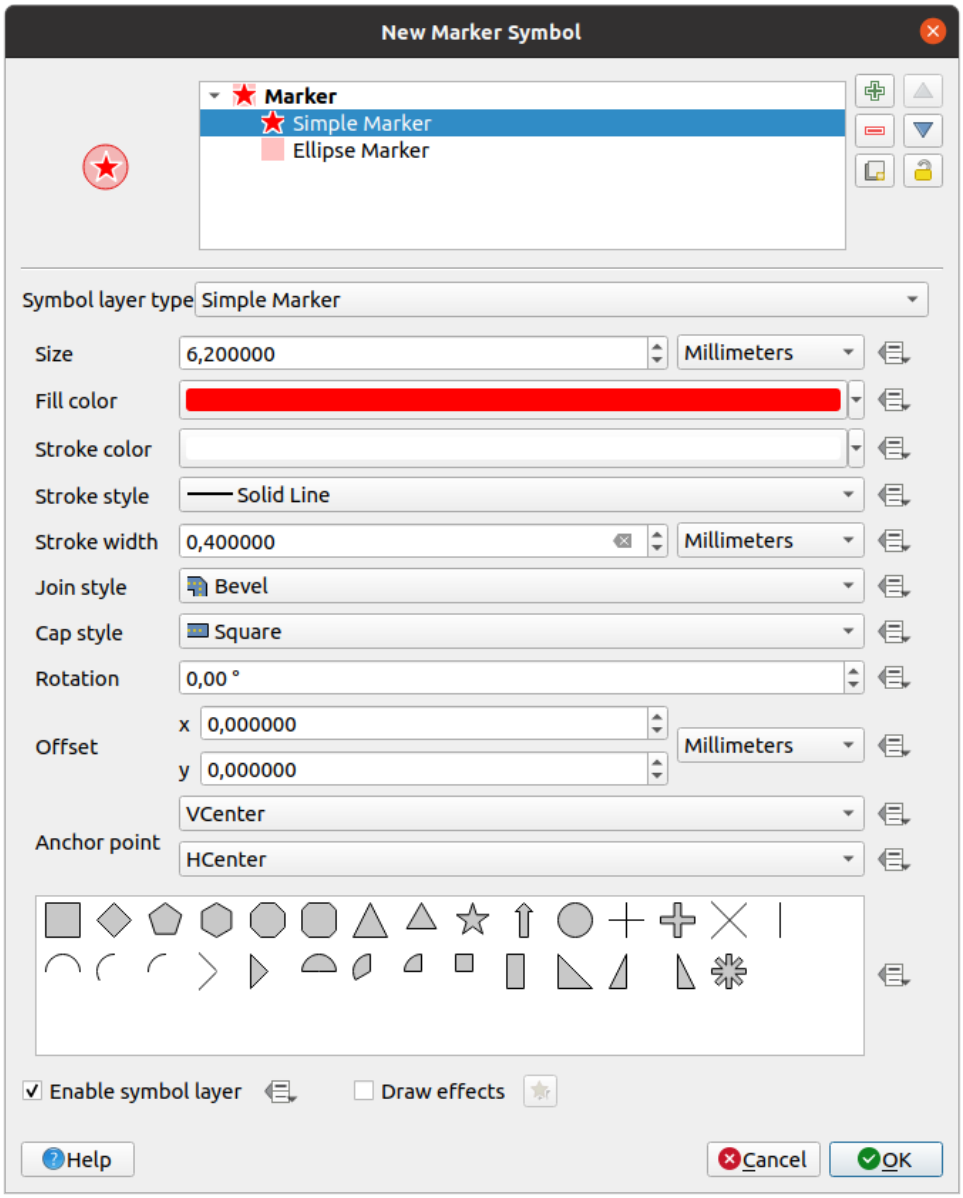



그림 10.11: 단순 마커 심볼 디자인하기

  - *Anchor point*: 심볼 위에 사분면 포인트를 배치 원점으로 정의할 수 있습니다. 앵커 포인트는 *Offset* 이 적용되는 포인트입니다.

- **애니메이션화 마커** (애니메이션화 마커 참조)
- **타원체 마커**: 단순 마커 심볼 레이어로, 너비 및 높이를 사용자 지정할 수 있습니다.
- **채운 마커 (filled marker)**: 단순 마커 심볼 레이어와 비슷하지만, 마커를 렌더링하는 데채우기 하위 심볼 을 이용한다는 점이 다릅니다. 이 마커를 사용하면 마커를 렌더링하는 데 그레이디언트 또는 부풀리기 (shape-burst) 같은 모든 기존 QGIS 채우기 (및 획) 스타일을 사용할 수 있습니다.
- **글꼴 마커 (font marker)**: 단순 마커 심볼 레이어와 비슷하지만, 마커를 렌더링하는 데 설치된 글꼴을 이용한다는 점이 다릅니다. 이 마커의 추가적인 속성은 다음과 같습니다:
  - *Font family*: 글꼴 집합
  - *Font style*: 글꼴 스타일
  - *Character(s)*: 심볼로서 표시될 텍스트를 나타냅니다. 텍스트를 직접 입력하거나, 글꼴 문자 집합 위젯에서 텍스트 문자를 선택할 수 있습니다. 또한 *Preview* 에서 텍스트 문자를 선택한 설정대로 실시간 미리보기할 수도 있습니다.
- **도형 생성기 (geometry generator)** (도형 생성기 참조)
- **마스크 (mask)**: 마스크 심볼의 하위 심볼이 마스크 형태를 정의합니다. 마스크 심볼의 색상 속성은 무시되며, 투명도만 적용될 것입니다. 마스크 심볼이 색상이 고정된 라벨 또는 다른 심볼과 겹치는 경우 유용한 습성입니다. 이렇지 않으면 겹치는 라벨 또는 심볼을 읽기 어렵기 때문입니다. 더 자세한 내용은마스크 속성 을 참조하세요.
- **래스터 이미지 마커 (Raster image marker)**: 이미지 (`PNG`, `JPG`, `BMP` 등등) 를 마커 심볼로 사용합니다. 이미지는 디스크에 저장된 파일, 원격 URL, 스타일 데이터베이스에 내장된 이미지 (자세한 내용은내장 파일 선택기 참조) 일 수 있으며, 또는 Base64 문자열로 인코딩되어 있을 수도 있습니다. 이미지의 너비와 높이를 각각 독립적으로 설정할 수 있고, Lock aspect ratio 를 이용해서 설정할 수도 있습니다. 이미지 크기를 모든흔히 쓰이는 단위 로 설정할 수 있고, 또는 이미지 원본 크기의 백분율로 (너비를 기준으로 크기 조정해서) 설정할 수도 있습니다.
- **벡터장 마커 (vector field marker)** (벡터장 마커 참조)
- **SVG 마커**: 사용자의 (*Settings* ▸ *Options*⋯▸ *System* 메뉴에 설정된) SVG 경로로부터 마커 심볼로 렌더링할 이미지를 제공받습니다. 심볼의 너비와 높이를 각각 독립적으로 설정할 수 있고, Lock aspect ratio 를 이용해서 설정할 수도 있습니다. 각 SVG 파일의 색상과 획도 조정할 수 있습니다. 이미지는 디스크에 저장된 파일, 원격 URL, 스타일 데이터베이스에 내장된 이미지 (자세한 내용은내장 파일 선택기 참조) 일 수 있으며, 또는 Base64 문자열로 인코딩되어 있을 수도 있습니다.

  *Dynamic SVG parameters* 를 통해 심볼을 설정할 수도 있습니다. SVG 심볼을 파라미터화하려면파라미터화할 수 있는 *SVG* 를 참조하세요.

  > **참고**
  >
  > SVG 버전 요구사항
  >
  > QGIS 는 SVG Tiny 1.2 profile 을 따라 SVG 파일을 렌더링합니다. 이 프로파일은 휴대폰, PDA, 노트북, 데스크탑컴퓨터를 아우르는 광범위한 기기에서 실행되도록 설계되어 있기 때문에, SVG 의 성능을 확장하기 위한 새 피처들과 함께 SVG 1.1 전체에 포함된 피처들의 서브셋을 포함하고 있습니다.
  >
  > 이 사양에 포함되지 않은 일부 피처는 QGIS 에서 정확하게 렌더링되지 않을 수도 있습니다.

### 라인 심볼

라인 도형 피처에 어울리는 라인 심볼은 다음과 같은 여러 *Symbol layer types* 를 가지고 있습니다:

표 10.2: List of line symbol layer types

| | | |
|---|---|---|
| *Simple line* | *Arrow* | *Filled line* |
| *Geometry generator* | *Hashed line* | *Interpolated line* |
| *Linear referencing* | *Lineburst* | *Marker line* |
| *Raster line* | | |

- **단순 라인** (기본값)

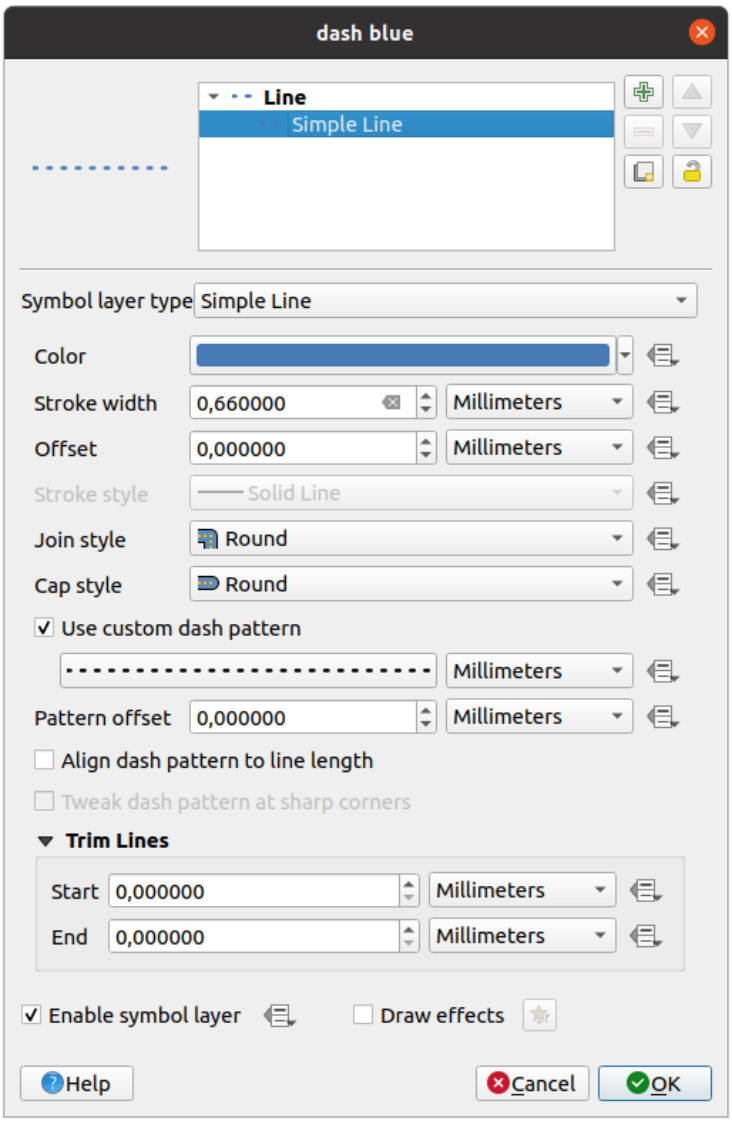



그림 10.12: 단순 라인 심볼 디자인하기

단순 라인 심볼 레이어 유형의 속성은 많은 부분단순 마커 심볼 과 동일하며, 추가로:

- *Use custom dash pattern*: *Stroke style* 설정을 무시하고 사용자 지정 점선 패턴으로 대체할 수 있습니다.

사용자가 선택한 단위로 모델을 형성하는 일련의 점선과 틈의 길이를 정의해야 합니다. 대화창 하단에 패턴의 전체 길이를 표시합니다.

  - *Pattern offset*: 점선/틈을 라인 상의 모서리를 고려해서 더 나은 위치에 배치할 수 있도록 점선/틈의 위치를 조정할 수 있습니다. (또한 인접한 점선 패턴 경계를"정렬"하는 데에도 잠재적으로 이용할 수 있습니다.)
  - *Align dash pattern to line length*: 라인의 끝이 점선 패턴의 틈이 아니라 완전한 점선 요소로 끝나도록 점선 패턴 길이를 조정합니다.
  - *Tweak dash pattern at sharp corners*: 완전한 점선 요소가 뾰족한 모서리에 걸치도록–뾰족한 모서리가 드러나도록–점선 패턴의 배치를 동적으로 조정합니다. 이 옵션은 *Align dash pattern to line length* 옵션을 체크한 상태에서만 동작합니다.
  - *Start* 그리고/또는 *End* 에서 *Trim lines*: 라인을 그릴 때 실제 라인스트링에서 처음 x 밀리미터 및 마지막 y 밀리미터를 다듬어서 렌더링할 수 있게 해줍니다. 이 옵션은 전체 라인 길이의 백분율을 포함하는단위 의 범위를 지원하며, 추가적인 제어를 위한 데이터 정의를 할 수도 있습니다. 예를 들어 라인 레이어가 라인의 시작과 종단에 배치된 마커 심볼 레이어와 중첩해서는 안 되는 복잡 심볼을 생성하는 경우, 시작/종단 다듬기 (trim) 거리를 사용할 수 있습니다.

- **화살표 (arrow)**: 라인을 만곡 (또는 직선) 화살표로 그립니다. 라인의 한쪽 또는 양쪽에 화살촉 표시를 할 수 있는데, 다음과 같은 옵션 (및 데이터 정의) 으로 화살표를 환경 설정할 수 있습니다:
  - *Head type*: 화살촉 유형
  - *Arrow type*: 화살 유형
  - *Arrow width*: 화살대 너비
  - *Arrow width at start*: 화살대 시작점 너비
  - *Head length*: 화살촉 길이
  - *Head thickness*: 화살촉 두께
  - *Offset*: 오프셋

  (라인 피처가 적어도 꼭짓점 3 개를 가지고 있어야만 하는) *Curved arrows* 만곡 화살표를 새 선분마다 *Repeat arrow on each segment* 반복해서 생성할 수 있습니다. 또 화살표 본체를 렌더링하는 데 그레이디언트나 부풀리기 같은채우기 심볼 도 이용합니다. 도형 생성기와 결합하면, 이 심볼 레이어 유형으로 흐름도를 표현할 수 있습니다.
- **도형 생성기 (geometry generator)** (도형 생성기 참조)
- **보간된 라인**: *Stroke width* 그리고/또는 *Color* 를 고정해서 (*Fixed width* 및 *Single color* 파라미터를 지정해서) 또는 도형을 따라 변화하도록 라인을 렌더링할 수 있습니다. 변화하도록 렌더링하는 경우, 다음을 입력해야 합니다:
  - *Start value* 및 *End value*: 피처 도형의 극단에서 보간을 위해 사용될 값입니다. 고정값일 수도 있고, 피처 속성값일 수도 있으며, 표현식을 기반으로 한 값일 수도 있습니다.
  - *Min. value* 및 *Max. value*: 보간을 수행하는 범위를 지정하는 값입니다. Load 버튼을 누르면 레이어에 적용된 최소 및 최대 시작/종단 값을 바탕으로 자동 입력됩니다.
  - 획 옵션 전용:
    * *Min. width* 및 *Max. width*: 변화하는 너비의 범위를 정의합니다. *Min. width* 는 *Min. value* 에, *Max. width* 는 *Max. value* 에 할당됩니다. 너비의단위 도 정의할 수 있습니다.
    * *Use absolute value*: 보간에 절대값만 고려합니다. (음의 값을 양의 값으로 사용합니다.)

    * *Ignore out of range*: 기본적으로, 피처의 `[start value - end value]` 범위가 `[min. value - max. value]` 범위에 포함되지 않는 경우 피처 도형에서 범위 밖에 있는 부분은 최소 또는 최대 너비로 렌더링됩니다. 이 옵션을 체크하면 범위 밖 부분을 전혀 렌더링하지 않습니다.
  - 변화하는 색상의 경우, 색상표 범주화 의 모든 보간 메소드를 사용할 수 있습니다.

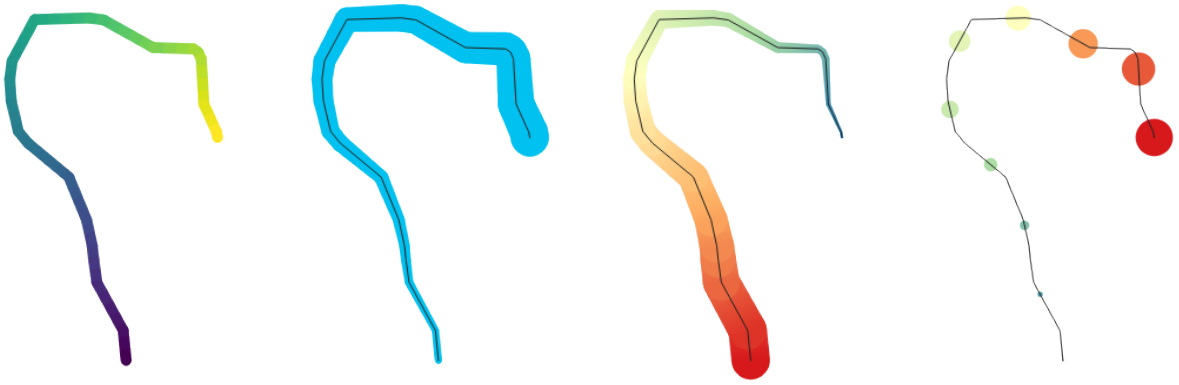


그림 10.13: 보간된 라인의 예시들

- **마커 라인**: 라인의 길이만큼마커 심볼 을 반복합니다.
  - *Marker placement*: (첫 번째 꼭짓점에서 시작하는) 규칙적인 간격 설정 그리고/또는 (첫 번째 또는 마지막 꼭짓점, 내부 꼭짓점, 라인 또는 각 선분의 중앙 포인트, 또는 모든 만곡 포인트의) 라인 도형 속성을 이용해서 마커 배치를 설정할 수 있습니다.
  - *Place on every part extremity*: 첫 번째 또는 마지막 꼭짓점 배치를 활성화한 경우, 이 옵션을 통해 다중 부분 도형이 가진 모든 부분의 첫 번째 또는 마지막 꼭짓점에서 마커를 렌더링할 수도 있습니다.
  - *Offset along line*: 라인을 따라 사용자가 선택한 (밀리미터, 포인트, 맵 단위, 축척 미터 (meters at scale), 백분율 등등의) 단위 로 마커 배치를 오프셋시킬 수도 있습니다:
    * 양의 값은 마커 심볼을 (*On first vertex* 및 *With interval* 배치를 선택한 경우) 라인 방향으로 오프셋시키고, (*On last vertex* 배치를 선택한 경우) 라인 반대 방향으로 오프셋시킵니다.
    * 닫히지 않은 라인 상에서, 음의 값은 (*On first vertex* 및 *On last vertex* 배치를 선택한 경우) 오프셋을 발생시키지 않으며, 또는 (마지막 꼭짓점으로부터) 심볼을 라인 반대 방향으로 오프셋시킵니다.
    * 닫힌 고리인 경우, QGIS 는 오프셋을 고리를 따라 (라인 방향 또는 반대 방향으로) 계속 반복되는 루프 (loop) 로 취급합니다. 예를 들어 오프셋을 150% 로 (또는 -10% 또는 -110% 로) 설정하면 이 오프셋을 닫힌 고리 길이의 50% 로 (또는 90% 로) 취급하게 됩니다.
  - *Rotate marker to follow line direction* 옵션을 통해 각 마커 심볼의 방향이 라인 방향을 따라야 하는지 아닌지 설정합니다.

    라인이 서로 다른 방향의 선분이 연속된 경우가 많기 때문에, 마커의 기울기를 라인을 따라 지정한 거리에서의 평균값으로 계산합니다. 예를 들어 *Average angle over* 속성을 `4mm` 로 설정하면, 심볼을 배치한 위치에서 앞/뒤 `2mm` 거리에 있는 포인트를 해당 마커 심볼의 라인 각도를 계산하는 데 사용한다는 뜻입니다. 이렇게 하면 전체 라인 방향에서 살짝 벗어난 국지적 편차를 평탄하게 만드는 (또는 제거하는) 효과를 볼 수 있습니다. 즉 마커 라인 심볼의 시각적 방향이 매우 보기 좋게 됩니다.
  - *Line offset*: 라인 피처로부터 마커 심볼을 오프셋시킬 수도 있습니다.
- **해시 라인 (hashed line)**: 라인 심볼의 길이만큼 라인 선분 (해시) 를 반복합니다. 이때 각 선분을 렌더링하는 데 라인 하위 심볼을 사용합니다. 다시 말해, 해시 라인은 마커 심볼을 선분으로 대체한 마커 라인이라는 뜻입니다. 따라서 해시 라인은 마커 라인 심볼과동일한 속성 을 다음 속성과 함께 가지고 있습니다:

  - *Hash length*: 해시의 길이
  - *Hash rotation*: 해시의 기울기

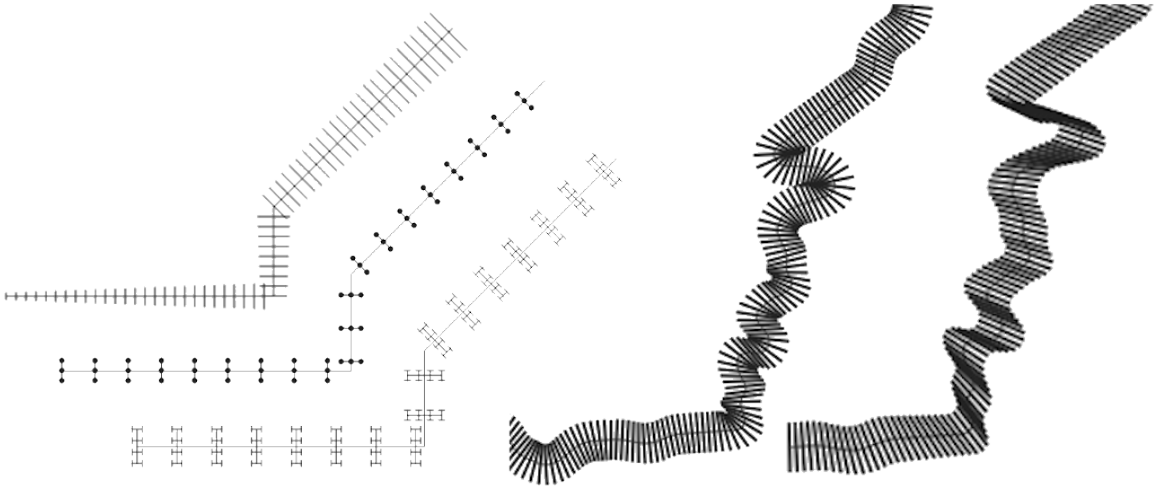


그림 10.14: 해시 라인의 예시

- **래스터 라인**: 라인 피처 형태의 길이를 따라 래스터 이미지를 반복해서 렌더링합니다. *Stroke width*, *Offset*, *Join style*, *Cap style* 및 *Opacity* 를 조정할 수 있습니다.
- **Linear referencing**: allows placing text labels at regular intervals along a line, or at positions corresponding to existing vertices. Positions can be calculated using Cartesian distances, or interpolated from z/m values. Parameters include:
  - *Measure placement*: Labels can be placed at regular cartesian 2d distances, at regular linearly interpolated spacing calculated using the Z or M values in geometries, or at existing vertices. A fixed or data-defined *Interval* is given.
  - *Quantity*: Label displayed at each position represent the running distance along the line, or the linearly interpolated Z or M value. Labels are rendered using the full range of *text* and *number* formatting functionalities available in QGIS.

> **참고**
>
> When using the distance-based placement or labels, the distances are calculated using 2D only. Cartesian calculations are based on the original layer CRS.

  - *Skip multiples of*: If set, labels which are a multiple of this value will be skipped over. This allows construction of complex referencing labels, e.g. where a symbol has two linear referencing symbol layers, one set to label every 100m in a small font, skipping multiples of 1000, and a second set to label every 1000m in a big bold font.
  - *Average angle over*: Labels are rendered using an angle calculated by averaging the linestring, so sharp tiny jaggies don't result in unslightly label rotation.
  - *Show marker symbols*, at referenced points in the line feature, using a full QGIS marker symbol. This allows e.g. showing a cross-hatch at the labeled point, for a "ruler"style line.
- **라인버스트 (lineburst)**: 라인의 너비를 따라 그레이디언트를 렌더링합니다. *Two color* 또는 *Color ramp* 가운데 하나를 선택할 수 있고, *Stroke width*, *Offset*, *Join style*, *Cap style* 을 조정할 수 있습니다.

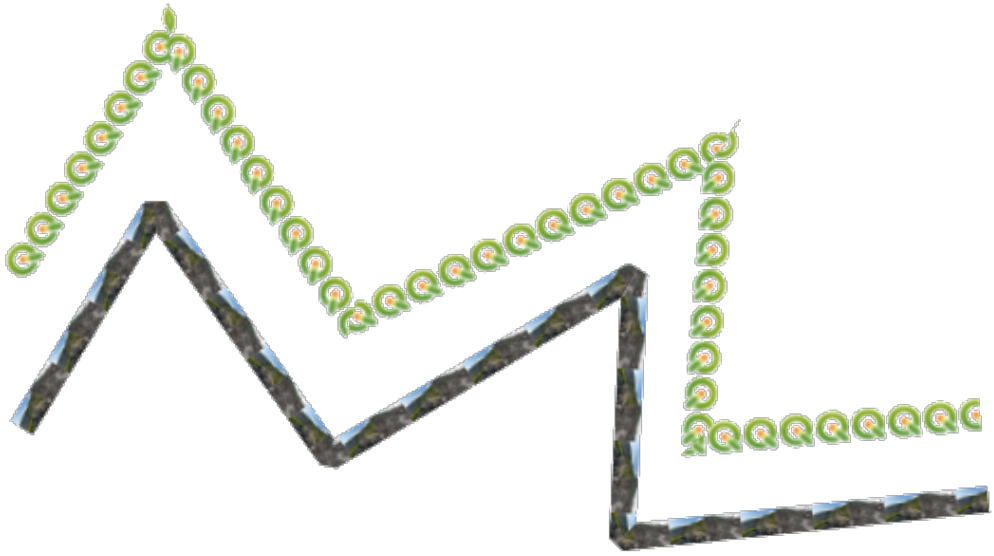


그림 10.15: 래스터 라인의 예시

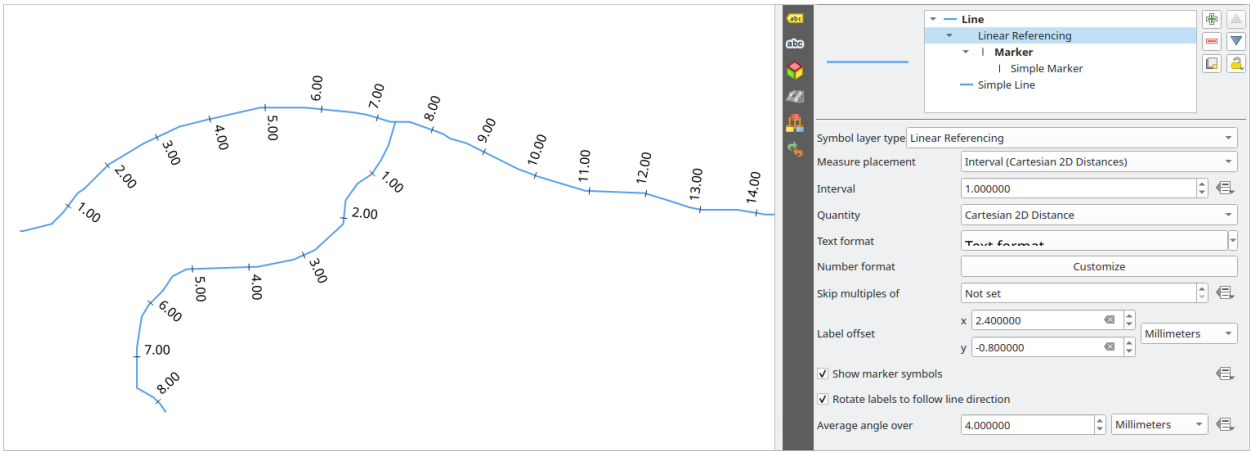



그림 10.16: Drawing linear labels along line feature



그림 10.17: 라인버스트 라인의 예시

- **Filled line**: renders the interior of the lines using a fill symbol allowing for lines filled with gradients, line hatches, etc. The *Stroke width*, *Offset*, *Join style*, *Cap style* can be adjusted.

### 채우기 심볼

폴리곤 도형 피처에 어울리는 채우기 심볼은 다음과 같은 여러 *Symbol layer types* 를 가지고 있습니다:

표 10.3: List of fill symbol layer types

| For polygon interior | For polygon ring |
|---|---|
| *Simple fill* | *Outline: Arrow* |
| *Centroid fill* | *Outline: Filled line* |
| *Geometry generator* | *Outline: Hashed line* |
| *Gradient fill* | *Outline: Interpolated line* |
| *Line pattern fill* | *Outline: Linear referencing* |
| *Point pattern fill line* | *Outline: Lineburst* |
| *Random marker fill* | *Outline: Marker line* |
| *Raster image* | *Outline: Raster line* |
| *SVG fill* | *Outline: Simple line* |
| *Shapeburst fill* | |

- **단순 채우기** (기본값): 폴리곤을 단일 색상으로 채웁니다.
- **중심 채우기 (centroid fill)**: 가시화된 피처의 중심 (centroid) 에마커 심볼 을 배치합니다. 하지만 마커의 위치가 피처의 실제 중심이 아닐 수도 있습니다. 중심 계산 시 맵 캔버스에 보이는 영역으로 폴리곤 (들) 이 잘려서 렌더링되기도 하고 폴리곤 내부의 구멍을 무시하기도 하기 때문입니다. 정확한 중심에 마커 심볼을 배치하려면도형 생성기 심볼 을 이용하십시오.

  다음 작업을 할 수 있습니다:

  - *Force placement of markers inside polygons*: 마커를 폴리곤 내부로 강제 배치할 수 있습니다.
  - *Draw markers on every part of multi-part features*: 다중 부분 피처의 모든 부분에 마커를 그리거나, 가장 큰 부분에만 포인트를 배치할 수 있습니다.
  - 현재 피처 도형과 겹치는 부분들을 유지하면서 (*Clip markers to polygon boundary*) 또는 심볼이 손해 입는

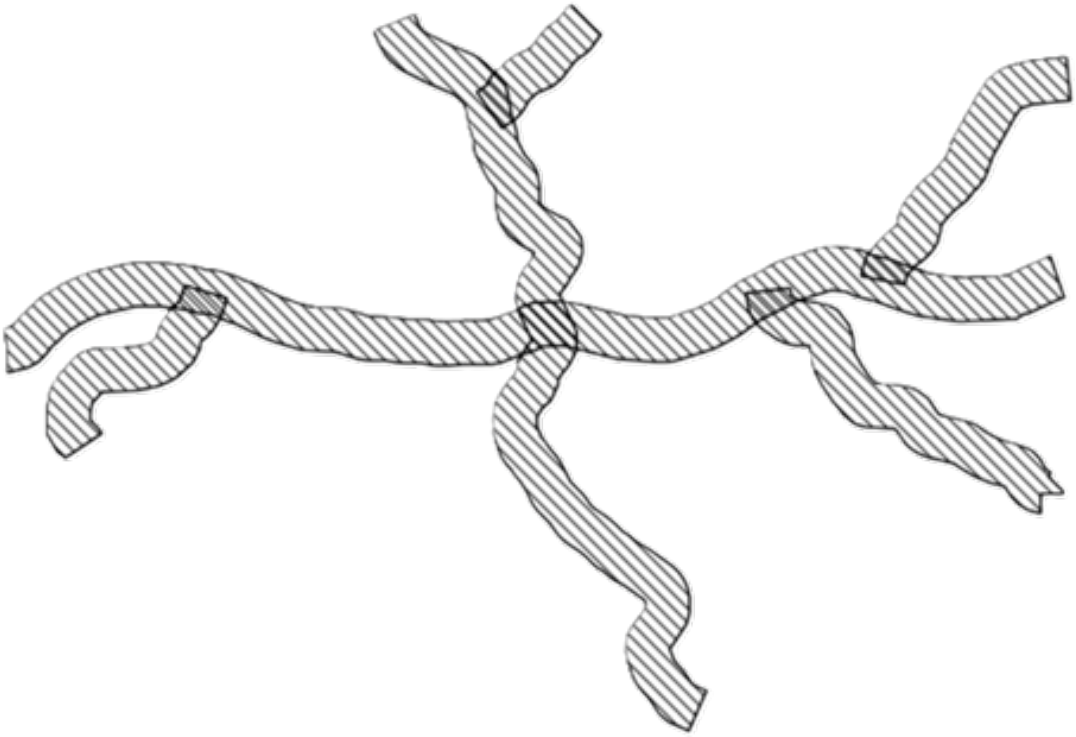


그림 10.18: Examples of filled lines

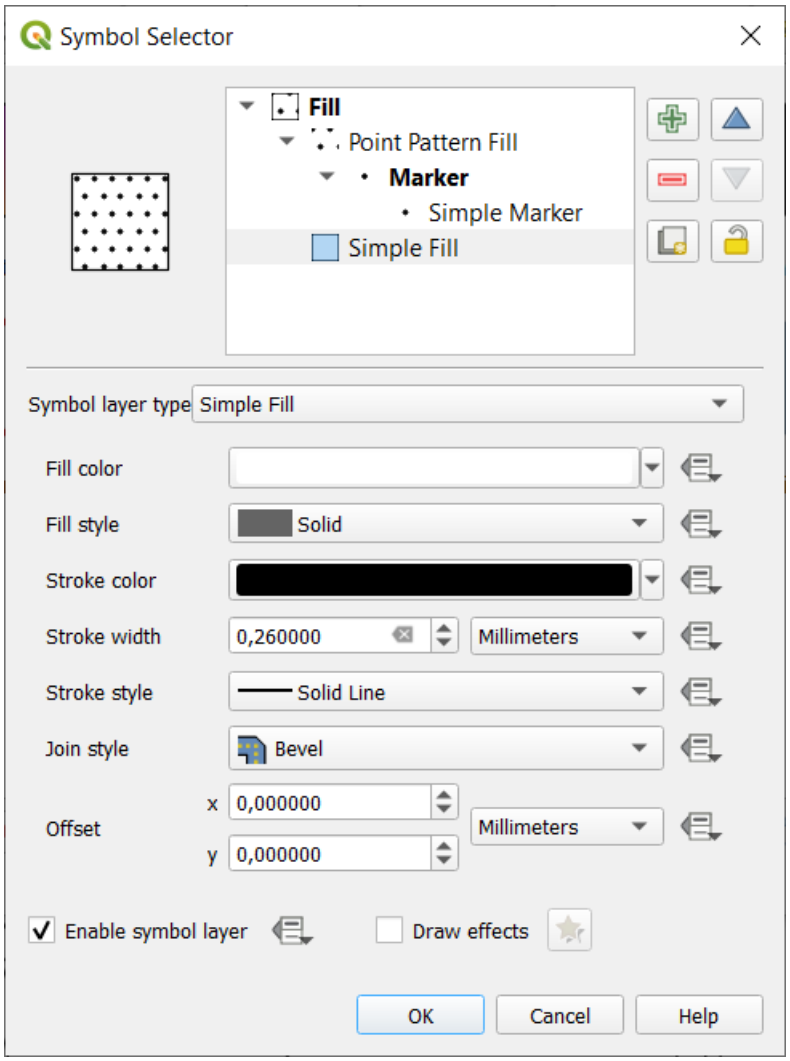



그림 10.19: 단순 채우기 심볼 디자인하기

  - *Offset*: 피처 경계선으로부터의 라인 오프셋 거리를 정의할 수 있습니다.
  - *Clipping*: 채우기 안에 있는 라인들을 폴리곤 형태에 맞춰 어떻게 잘라낼 것인지 제어할 수 있습니다. 다음 옵션들을 사용할 수 있습니다:
    * *Clip During Render Only*: 피처의 경계 상자 전체를 커버하는 라인을 생성한 다음 렌더링하는 동안 잘라냅니다. 라인의 (시작 및 종단) 양극단이 가시화되지 않을 것입니다.
    * *Clip Lines Before Render*: 렌더링 전에 라인을 폴리곤의 정확한 형태에 맞춰 잘라냅니다. 라인의 (선끝 스타일, 시작/종단 마커 라인 객체 등등을 포함하는) 양극단이 가시화되며, (라인 심볼 설정에 따라) 폴리곤 바깥으로 삐져나가는 경우가 있을 수도 있습니다.
    * *No Clipping*: 어떤 잘라내기도 수행하지 않습니다. 라인이 피처의 경계 상자 전체를 커버할 것입니다.

> **참고**
>
> When exporting maps to vector formats such as PDF or SVG, or when the line symbol in the pattern has data-defined properties, QGIS renders the line pattern fill using a line-by-line vector-based approach. This results in smaller output file sizes, higher quality exports, and improved editability of the pattern lines in external vector graphics applications.

- **포인트 패턴 채우기**: 마커 심볼 의 그리드 패턴으로 폴리곤을 채웁니다. 다음을 설정할 수 있습니다:
  - *Alignment*: 피처 (들) 를 기준으로 패턴을 어떻게 배치할지 정의할 수 있습니다:
    * *Align pattern to feature*: 각 피처 내부에 마커 라인을 렌더링합니다.
    * *Align pattern to map extent*: 마커가 피처 위로 보기 좋게 정렬될 수 있게 전체 맵 범위에 걸쳐 패턴을 렌더링합니다.

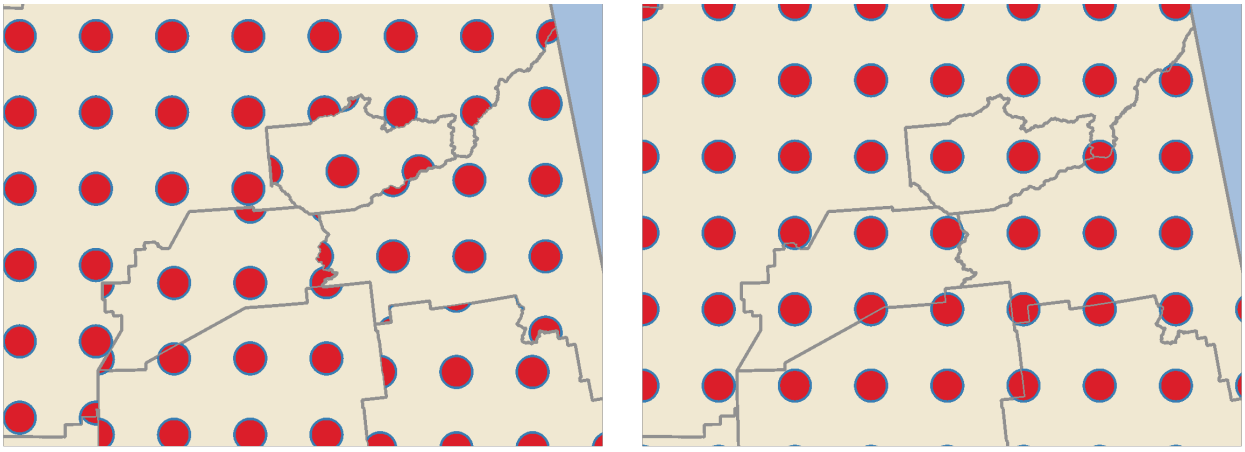


그림 10.20: 피처에 포인트 패턴 정렬하기 (왼쪽) | 전체 맵 범위에 정렬하기 (오른쪽)

  - *Distance*: 연속되는 마커 사이의 수평 (*Horizontal*) 및 수직 (*Vertical*) 거리를 정의할 수 있습니다.
  - *Displacement*: 한 열에서 (또는 행에서) 연속되는 마커 사이의 *Horizontal* (또는 *Vertical*) 정렬의 오프셋을 정의할 수 있습니다.
  - *Offset*: 피처 경계로부터의 수평 (*Horizontal*) 및 수직 (*Vertical*) 거리를 정의할 수 있습니다.
  - *Clipping*: 채우기 안에 있는 마커들을 폴리곤 형태에 맞춰 어떻게 잘라낼 것인지 제어할 수 있습니다. 다음 옵션들을 사용할 수 있습니다:

  * *Clip to shape*: 폴리곤 내부 부분만 가시화되도록 마커를 잘라냅니다.
  * *Marker centroid within shape*: 마커 중심이 폴리곤 내부에 들어오는 경우에만 렌더링합니다. 마커가 폴리곤 외부에 렌더링되더라도 잘라내지 않을 것입니다.
  * *Marker completely within shape*: 폴리곤 내부에 완전히 들어오는 마커만 렌더링합니다.
  * *No clipping*: 폴리곤과 조금이라도 교차하는 모든 마커를 렌더링합니다. (명확하게 하자면"폴리곤과 조금이라도 교차하는 경계 상자를 가진 모든 마커"입니다.)

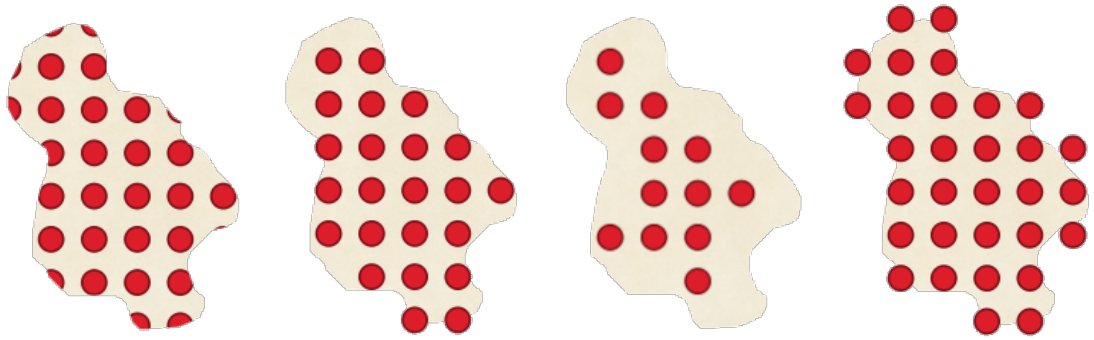


그림 10.21: 채우기 마커 잘라내기 - (왼쪽에서 오른쪽으로) *Clip to shape*, *Marker centroid within shape*, *Marker completely within shape*, *No clipping*

  - *Rotation*: 전체 패턴을 시계 방향으로 기울일 수 있습니다.
  - *Randomize pattern*: 이 설정 그룹은 포인트 패턴 채우기의 각 포인트를 수평으로 (*Horizontally*) 또는 수직으로 (*Vertically*) 지정한 최대 거리까지 임의 이동시킬 수 있습니다. 밀리미터, 포인트, 맵 단위, 또는 (패턴 너비 또는 높이를 기준으로 하는) "백분율"처럼, 지원하는 모든 단위로 최대 오프셋을 지정할 수 있습니다.

    심볼 패턴이 맵을 새로고침할 때마다"점프"해서 위치를 바꾸는 일을 방지하려면 선택적인 임의의 숫자 뿌리값 (random number seed) 을 설정하면 됩니다. 데이터 정의 무시도 지원합니다.

    **참고**

    *Randomize pattern* 과랜덤 마커 채우기 심볼 유형의 가장 큰 차이점은 포인트 패턴의 임의 오프셋이 마커를"준규칙적으로 (quasi-regular)"배치할 수 있게 해준다는 점입니다. 왜냐하면 패턴의 포인트들이 실질적으로 그리드에 묶여 있어, 비어 있는 부분이나 마커들이 겹치는 일이 없는, 임의에 반쯤 가까운 채우기를 생성할 수 있기 때문입니다. (랜덤 마커 채우기는 다릅니다. 이 채우기는 언제나 포인트를 완전히 임의로 배치할 것이기 때문에, 시각적으로 포인트들이 뭉쳐 있거나 보기에 좋지 않은 비어 있는 부분이 발생하는 경우가 생길 수도 있습니다.)

- **랜덤 마커 채우기**: 폴리곤 경계선 안의 랜덤한 위치에마커 심볼 을 배치해서 폴리곤을 채웁니다. 다음을 설정할 수 있습니다:
  - *Count method*: 렌더링할 마커 심볼의 개수를 절대 개수로 또는 치밀도 기반으로 간주할지 여부를 선택할 수 있습니다.
  - *Point count*: 렌더링할 마커 심볼의 개수를 정의할 수 있습니다.
  - *seed*: 일관적인 배치를 위한 선택적인 임의의 숫자 뿌리값을 정의할 수 있습니다.
  - *Density area*: 치밀도 기반 개수 메소드를 선택한 경우, 마커의 채우기 치밀도가 맵을 새로고침할 때마다 서로 다른 축척 및 확대/축소 수준에서도 동일하게 유지되도록 할 수 있습니다. (또한 QGIS 서버와 타일 기반 렌더링을 사용하는 경우에도 보기 좋게 임의 배치할 수 있게 해줍니다.)

  - *Clip markers to polygon boundary*: 폴리곤 경계 가까이 렌더링된 마커를 폴리곤 경계선에 맞춰 잘라낼지 말지를 설정할 수 있습니다.

- **Raster image fill**: fills the polygon with tiles from a raster image (`PNG` `JPG`, `BMP` ⋯). The image can be a file on the disk, a remote URL or an embedded file encoded as a string (*more details*). Options include (data defined) opacity, size, coordinate mode (object or viewport), rotation and offset. Under the option Size you can independently adjust the width and height of the fill pattern, enabling stretched raster fills in either the horizontal or vertical directions. The image width and height can be set using any of the *common units* or as a percentage of the original size.
- **SVG 채우기**: *SVG 마커* 의 크기를 지정해서 (*Texture width*) 폴리곤을 채웁니다.
- **부풀리기 (shapeburst) 채우기**: 이 옵션은 그레이디언트 채우기에 버퍼를 적용하는데, 폴리곤의 경계선에서 중심을 향해 그레이디언트를 그립니다. 경계선에서 음영을 그릴 거리, 색상표 사용 또는 단순한 2 색상 그레이디언트 사용, (선택적인) 채우기의 번짐 효과, 그리고 오프셋 파라미터 등을 설정할 수 있습니다.
- **Outline: Arrow**: uses a line *arrow symbol* layer to represent the polygon boundary (*Rings*) which can be the interior rings only, the exterior ring only or all the rings. The settings for the outline arrow are the same as for arrow line symbols.
- **Outline: Filled line**: uses a *filled line symbol* layer to represent the polygon boundary (*Rings*) which can be the interior rings only, the exterior ring only or all the rings. The settings are the same as for filled line symbols.
- **Outline: Hashed line**: uses a *hash line symbol* layer to represent the polygon boundary (*Rings*) which can be the interior rings only, the exterior ring only or all the rings. The other settings are the same as for hashed line symbols.
- **Outline: Interpolated line**: uses an *interpolated line symbol* layer to represent the polygon boundary (*Rings*) which can be the interior rings only, the exterior ring only or all the rings. The other settings are the same as for interpolated line symbols.
- **Outline: Linear referencing**: uses a *linear referencing symbol* layer to represent the polygon boundary (*Rings*) which can be the interior rings only, the exterior ring only or all the rings. The other settings are the same as for linear referencing symbols.
- **Outline: Lineburst**: uses a *lineburst symbol* layer to represent the polygon boundary (*Rings*) which can be the interior rings only, the exterior ring only or all the rings. The other settings are the same as for lineburst symbols.
- **Outline: Marker line**: uses a *marker line symbol* layer to represent the polygon boundary (*Rings*) which can be the interior rings only, the exterior ring only or all the rings. The other settings for the outline marker line are the same as for marker line symbols.
- **Outline: Raster line**: uses a *raster line symbol* layer to represent the polygon boundary (*Rings*) which can be the interior rings only, the exterior ring only or all the rings. The other settings for the outline hashed line are the same as for hashed line symbols.
- **Outline: Simple line**: uses a *simple line symbol* layer to represent the polygon boundary (*Rings*) which can be the interior rings only, the exterior ring only or all the rings. The *Draw line only inside polygon* option displays the polygon borders inside the polygon and can be useful to clearly represent adjacent polygon boundaries. The other settings for the outline simple line are the same as for simple line symbols.

**참고**

도형 유형이 폴리곤인 경우, 라인/폴리곤을 자동적으로 캔버스 범위에 맞춰 자르는 옵션을 비활성화할 수 있습니다. 어떤 경우 이 자르기 옵션이 심볼 스타일에 방해가 되기도 하기 때문입니다. (예를 들어 중심 채우기 유형은 중심이 언제나 실제 피처의 중심이어야만 합니다.)

### 파라미터화할 수 있는 SVG

*SVG marker* 의 색상을 변경할 수 있습니다. 채우기 색상의 경우 `param(fill)` 자리 표시자를, 채우기 불투명도의 경우 `param(fill-opacity)` 를, 획의 색상 및 불투명도의 경우 각각 `param(outline)` 및 `param(outline-opacity)` 를 그리고 획 너비의 경우 `param(outline-width)` 를 추가해야 합니다. 이 자리 표시자들 뒤에 다음과 같은 선택적인 기본값이 따라올 수 있습니다:

```
<svg width="100%" height="100%">
<rect fill="param(fill) #ff0000" fill-opacity="param(fill-opacity) 1" stroke=
↪"param(outline) #00ff00" stroke-opacity="param(outline-opacity) 1" stroke-width=
↪"param(outline-width) 10" width="100" height="100">
</rect>
</svg>
```

더 일반적으로는, `param(param_name)` 을 사용하면 SVG 를 자유롭게 파라미터화할 수 있습니다. 이 파라미터는 속성값 또는 노트 텍스트 가운데 하나로 사용할 수 있습니다:

```
<g stroke-width=".265" text-anchor="middle" alignment-baseline="param(align)">
  <text x="98" y="147.5" font-size="6px">param(text1)</text>
  <text x="98" y="156.3" font-size="4.5px">param(text2)</text>
</g>
```

그 다음, *Dynamic SVG parameters* 테이블에서 이 파라미터를 표현식으로서 정의할 수 있습니다.

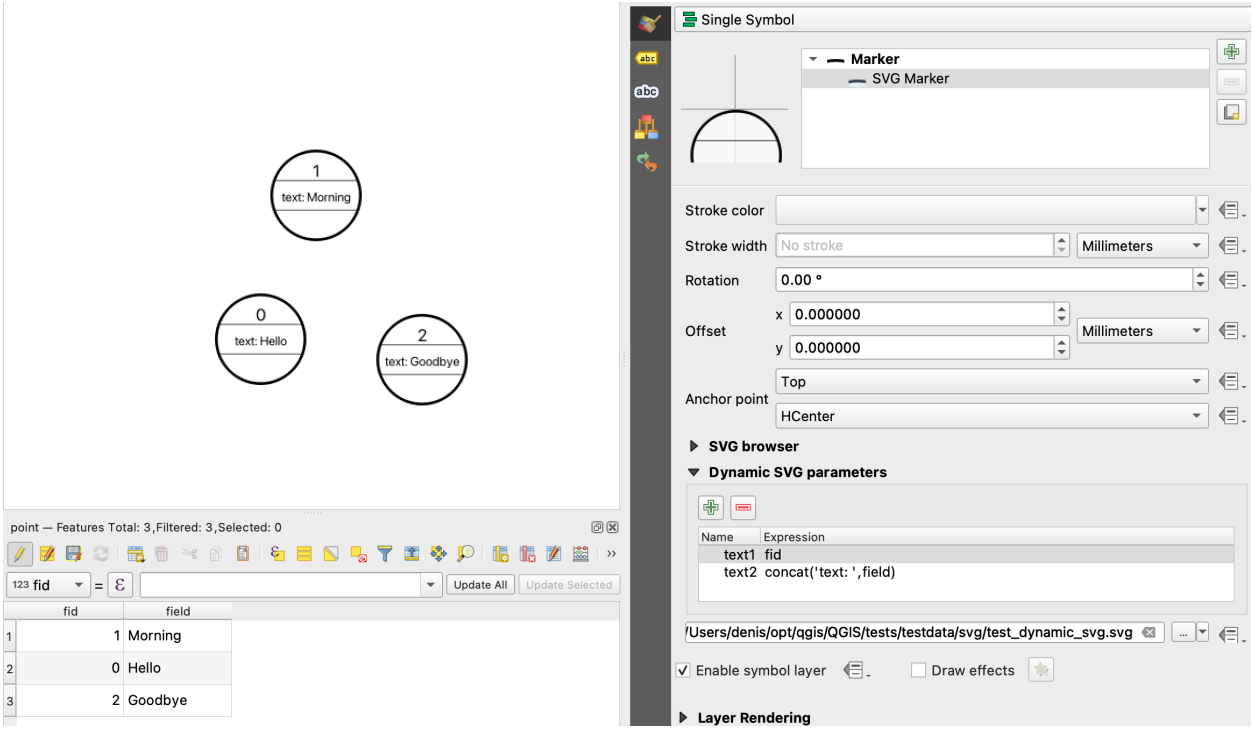



그림 10.22: 동적 SVG 파라미터 테이블

> **참고**
>
> QGIS 는 완전한 SVG 노드를 찾고 있기 때문에, 사용자 파라미터가 좀 더 복잡한 노드 안에 있는 경우 완전한 노드를 표현식으로 주입해줘야 합니다. 예를 들면 `transform="rotate(param(angle)"` 는 동작하지

않을 것입니다. 대신 `transform="param(rotation)"` 을 사용하고 `rotation` 파라미터를 `'rotate(' || coalesce(my_field, 0) || ')'` 표현식으로 정의해야 할 것입니다.

### 도형 생성기

모든 심볼 유형에 사용할 수 있는 *geometry generator* 심볼 레이어는 렌더링 과정에서 도형을 실시간으로 생성하기 위해표현식 문법 을 사용할 수 있습니다. 산출 도형은 원본 *Geometry type* 과 일치할 필요가 없으며, 각각의 심볼 레이어 위에 서로 다르게 수정된 심볼 레이어를 몇 개 추가할 수도 있습니다.

*Units* 속성을 설정할 수 있습니다: 도형 생성기 심볼이 레이어에 적용되지 않은 경우 (예를 들어 도형 생성기 심볼이 조판 항목에 사용된 경우) 이 속성을 통해 생산되는 산출물을 더 정밀하게 제어할 수 있습니다.

다음은 몇몇 예시입니다:

- 심볼을 피처의 중심점으로 렌더링하기

```
centroid( $geometry )
```

- 포인트 피처로부터 100 맵 단위 거리 안에 들어오는 피처들을 가시적으로 중첩시키기, 예: 포인트 주변에 100 미터 버퍼 생성하기

```
buffer( $geometry, 100 )
```

- 멀티포인트 도형 생성기로 사용하는 경우 중심 피처 포인트를 둘러싸는 포인트들의 방사형 효과 (radial effect) 생성하기

```
collect_geometries(
  array_foreach(
    generate_series( 0, 330, 30 ),
    project( $geometry, 3, radians( @element ) )
  )
)
```

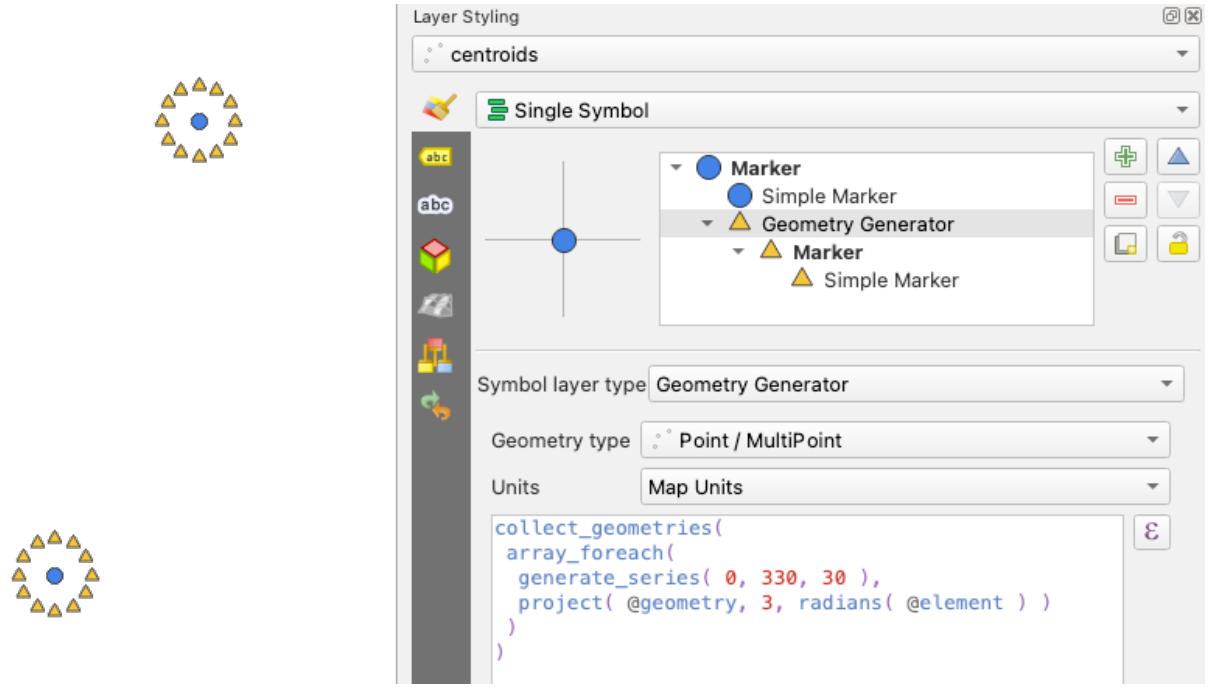
- 중심 피처 포인트를 둘러싸는 포인트들의 방사형 효과 (radial effect) 생성하기. 포인트 개수는 필드에 따라 달라집니다.

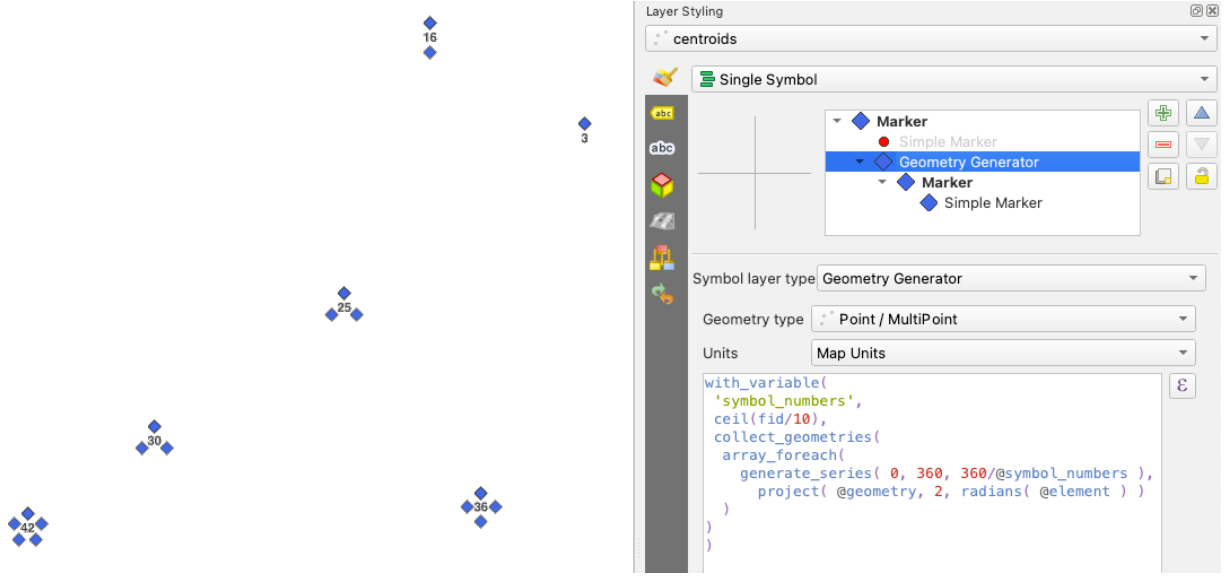
```
with_variable(
  'symbol_numbers',
  ceil(fid/10),
  collect_geometries(
    array_foreach(
      generate_series( 0, 360, 360/@symbol_numbers ),
      project( $geometry, 2, radians( @element ) )
    )
  )
)
```

- 두 레이어의 피처들을 연결하는 만곡 화살표 라인을 두 레이어의관계 를 바탕으로 생성하기

```
collect_geometries(
  with_variable(
    'destination_points',
    relation_aggregate( 'the_relation_id', 'array_agg', centroid( $geometry ) ),
    array_foreach(
      @destination_points,
```

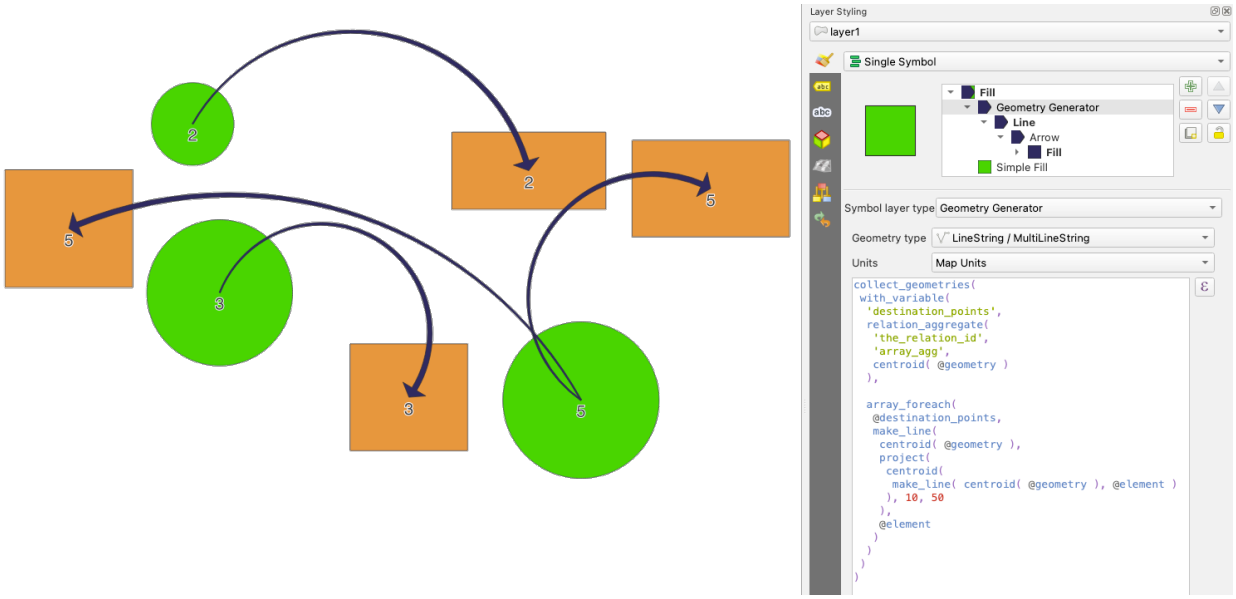
(다음 페이지에 계속)


Layer Styling
centroids
Single Symbol
Marker
Simple Marker
Geometry Generator
Marker
Simple Marker
Symbol layer type
Geometry Generator
Geometry type
Point / MultiPoint
Units
Map Units
collect_geometries(
array_foreach(
generate_series( 0, 330, 30 ),
project( @geometry, 3, radians( @element ) )
)
)


16
3
25
30
42
36
Layer Styling
centroids
Single Symbol
Marker
Simple Marker
Geometry Generator
Marker
Simple Marker
Symbol layer type
Geometry Generator
Geometry type
Point / MultiPoint
Units
Map Units
with_variable(
'symbol_numbers',
ceil(fid/10),
collect_geometries(
array_foreach(
generate_series( 0, 360, 360/@symbol_numbers ),
project( @geometry, 2, radians( @element ) )
)
)
)

(이전 페이지에서 계속)

```
      make_line(
        centroid( @geometry ),
        project(
          centroid( make_line( centroid( @geometry ), @element ) ),
          10,
          50
        ),
        @element
      )
    )
  )
)
```




### 벡터장 마커

지구의 변형, 조류 (□□) 의 흐름 등과 같은 벡터장 (vector field) 데이터를 표시하는 데 벡터장 마커를 이용합니다. 이 마커는 선택한 데이터 포인트의 속성에 따라 축척 및 방향이 정의된 라인으로 (화살표가 낫습니다) 벡터를 표시합니다. 포인트 데이터를 렌더링하는 데에만 이 마커를 사용할 수 있습니다. 라인 및 폴리곤 레이어를 이 심볼로 그릴 수는 없습니다.

벡터장은 데이터의 속성으로 정의되는데, 다음 방법 가운데 하나로 벡터장을 표현할 수 있습니다:

- **데카르트** 요소 (벡터장의 `x` 및 `y` 요소)
- **극 (polar)** 좌표: 이 경우, 속성이 `Length` 와 `Angle` 을 정의합니다. 각도는 북쪽에서 시계 방향으로 또는 동쪽에서 반시계 방향으로 측정될 수도 있고, 도 단위 또는 라디안 단위일 수도 있습니다.
- **높이만 있는 (height only)** 데이터: 데이터의 속성을 이용해서 척도를 조정한 수직 화살표를 표시합니다. 예를 들어 지구 변형의 수직적 요소를 표시하는 데 적합합니다.

벡터장을 살펴보기 위해 벡터장의 규모 (magnitude) 를 적절한 크기로 늘리거나 줄일 수 있습니다.

#### 애니메이션화 마커

애니메이션화 마커 심볼 유형은 사용자 맵 상에 포인트를 `.GIF`, `.WebP`, `.MNG` 등등의 애니메이션 파일을 사용해서 표현할 수 있게 해줍니다. 다음을 지정할 수 있습니다:

- *File*: 파일 경로
- *Frame rate*: 초당 보이는 단계의 개수로, 애니메이션이 얼마나 빠르게 재생되는지를 나타냅니다.
- *Size*: 모든지원 단위 로 설정할 수 있는 크기
- *Opacity*: 불투명도
- *Rotation*: 회전 각도
- *Offset*: 마커 위치로부터 *x* 와 *y* 방향으로 오프셋을 설정할 수 있습니다.
- *Anchor point*: 앵커 포인트

애니메이션화 심볼은 다음 두 가지 방법으로 처리할 수 있습니다:

- **사용자 맵이 애니메이션으로써 환경설정되지 않은 경우** (예를 들면 애니메이션이 없는 표준 QGIS 프로젝트인 경우) 애니메이션화 마커의 프레임이 현재 타임스탬프만으로 결정될 것입니다.
- **사용자 맵이**애니메이션으로써 환경설정 **된 경우** 애니메이션화 마커가 애니메이션의 타임라인과 동기화될 것입니다. 다시 말해 애니메이션이 일시정지됐을 때 애니메이션화 마커도 일시정지될 것이고, 애니메이션을 다시 시작하면 그와 함께 다시 진행될 것이라는 뜻입니다. 시계열 애니메이션에 지정된 프레임 레이트에 따라 맵도 다시 그려질 것입니다. 이 모드는 시계열 제어기를 사용해서 애니메이션을 내보낼 때에도 적용됩니다.

## 10.3 라벨 설정

Labels are textual information you can display on maps. They add details you could not necessarily represent using symbols, and may refer to vector features, raster cells, mesh elements, or simple annotations on the map⋯Two types of text-related items are available in QGIS:

- *Text Format*: 글꼴, 크기, 색상, 음영, 배경, 버퍼 등등 텍스트의 모양을 정의합니다.

  주로글꼴 위젯을 통해, 맵 위에 놓일 텍스트 (조판/맵 제목, 장식, 축척 막대 등등) 를 렌더링하는 데 이런 옵션들을 사용할 수 있습니다.

  *Text Format* 항목을 생성하려면:

  1. *Style Manager* 대화창을 열고
  2. *Text format* 탭을 선택한 다음
  3. Add item 버튼을 클릭하면 됩니다. 환경 설정 을 위한 *Text Format* 대화창이 열립니다. 다른 속성과 마찬가지로, 이 속성들을데이터 정의 할 수 있습니다.

- *Label Settings*: extend the text format settings with properties related to the location or the interaction with other texts or features (*callouts*, *placement*, *overlay, scale visibility*, *mask* ⋯).

  They are used to configure smart labelling for vector and mesh layers through the *Labels* tab of the vector or mesh *Layer Properties* dialog or *Layer Styling* panel or using the Layer Labeling Options button of the *Label toolbar*.

  *Label Settings* 항목을 생성하려면:

  1. *Style Manager* 대화창을 열고

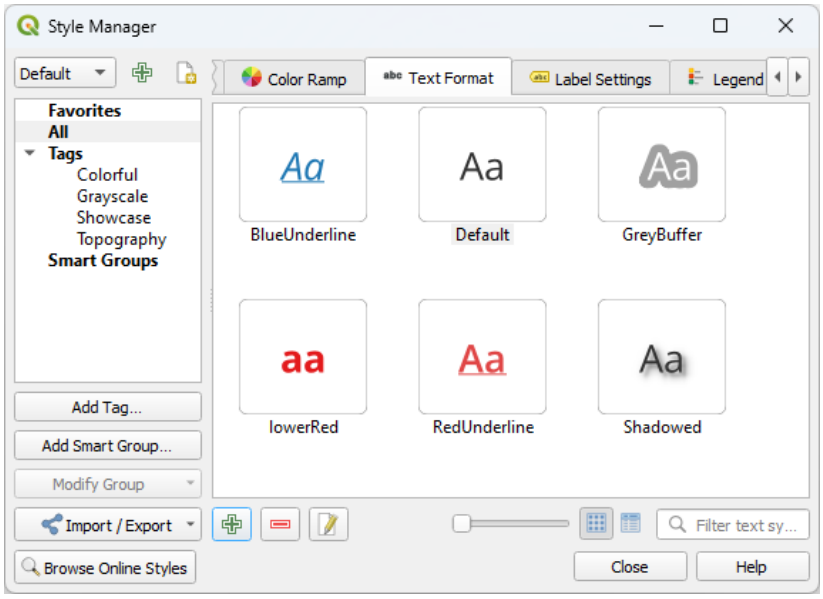



그림 10.23: 스타일 관리자 대화창의 텍스트 서식

2. *Label Settings* 탭을 선택한 다음

3. Add item 메뉴를 클릭하고 사용자가 라벨을 추가하려는 피처의 도형 유형에 해당하는 항목을 선택하십시오.

다음과 같은 속성과 함께 *Label Settings* 대화창이 열립니다. 다른 속성과 마찬가지로, 이 속성들을데이터 정의할 수 있습니다.

### 10.3.1 라벨 텍스트 서식 설정

Whether you are configuring a *Text Format* or *Label Settings* item, and depending on the type of layer you are configuring, you will be given the following options:

표 10.4: Label properties by data type

| 속성 탭 | 텍스트 서식 | 라벨 설정 | Vector or mesh layer | raster layer |
|---|---|---|---|---|
| *Text* | ☑ | ☑ | ☑ | ☑ |
| *Formatting* | ☑ | ☑ | ☑ | ☑ |
| *Buffer* | ☑ | ☑ | ☑ | ☑ |
| *Mask* | | ☑ | ☑ | |
| *Background* | ☑ | ☑ | ☑ | ☑ |
| *Shadow* | ☑ | ☑ | ☑ | ☑ |
| *Callout* | | ☑ | ☑ | |
| *Placement* | | ☑ | ☑ | ☑ |
| *Rendering* | | ☑ | ☑ | ☑ |

**주의**

While for legibility, “feature”is the name used below to indicate the item being labeled, depending on the underlying layer type, it can be replaced by “pixel”, “face”or “vertex”.



텍스트 탭

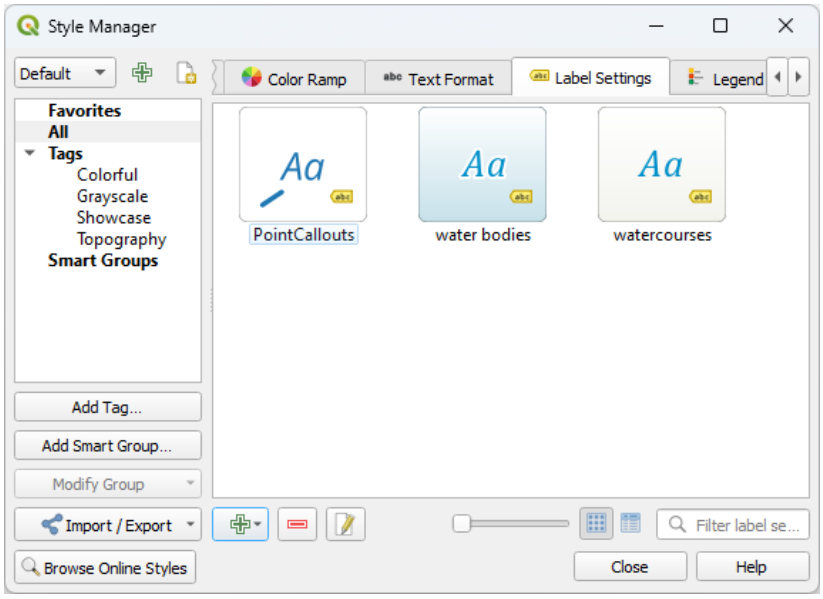



그림 10.24: 스타일 관리자 대화창에서 라벨 설정

설정할 수 있고, 스타일 데이터베이스에 새 항목을 추가할 수도 있습니다. *Save format*…또는 *Save settings*…버튼을 누르고 이름 및 태그 (들) 를 입력하면 됩니다.

> **참고**
>
> *Label Settings* 항목을 환경 설정하는 이 위젯에서 텍스트 서식 항목도 설정할 수 있습니다. 항목을 선택하면 라벨의 현재텍스트 속성 을 한번에 덮어씁니다. 마찬가지로 텍스트 서식을 생성하거나 덮어쓸 수 있습니다.

### Allow HTML Formatting

With *Allow HTML Formatting* enabled, you need to provide the HTML code in the *Value* field. Use whitespaces instead of tabs for any kind of indentation. The expression is parsed and any supported HTML tag overrides its corresponding setting in the labels properties. Because it is impossible to list and detail every HTML tag and CSS property that QGIS currently supports, we invite you to explore and test in your labels the ones supported by the underlying Qt library.

**Examples of supported HTML tags:**

- Text formatting, such as italic or bold, e.g.:

```
<i>QGIS</i> <b>rocks!</b>
```

- Superscript and subscript, where the text will be vertically $^{\text{super}}$ or $_{\text{sub}}$ aligned and automatically sized to 2/3 of the parent font size. You can also set a fixed font size for the superscript/subscript by including css rules, e.g.:

```
<sup style="font-size:33pt">my superscript text</sup>
```

- Text horizontal alignment, using either HTML the `align="xxx"` attribute or `center` tag. For HTML5 compatibility, prefer using the CSS property `text-align`.

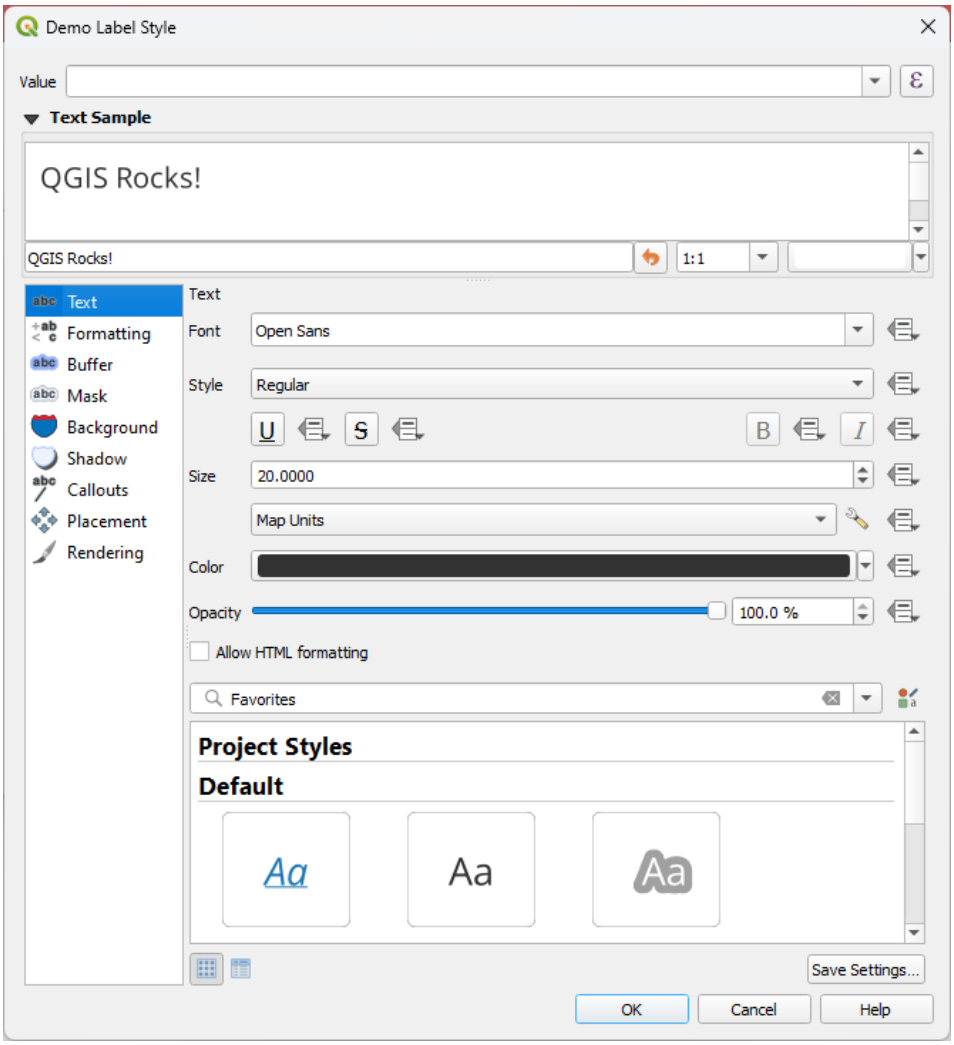



그림 10.25: 라벨 설정 - 텍스트 탭

> **주의**
>
> Horizontal alignment can not be used for curved labels.

- Structuring a block of text using header tags (such as `h1`, `h2`, ⋯), or paragraphs (with `p`, `div`, `br`):

```
<div class="myDiv">
  <h2>QGIS always rocks!!</h2>
  <p align="center">Let's dive into details of its nice features.</p>
</div>
```

- Image insertion: any image format readable by QGIS can be used in HTML label content. It can be served from local file paths, HTTP links, or base64 embedded content, using the `src="xxx"` attribute. Image sizes can be specified via the `width="##"` and `height="##"` attributes, in `points` unit. If width or height is not specified it will automatically be calculated from the original image size. Images are placed inline only, not as floating images, and not on curved text labels.

```
<img src="qgis.png" width=40 height=60>
```

**Examples of supported CSS properties:**

- Font properties (`color`, `font-family`, `font-size`, `font-weight`, `font-style`, `word-spacing`). Note that `word-spacing` will always use unit points.
- Text decorations such as underline, overline and line-through (`text-decoration`)
- Text alignment (`vertical-align`, `text-align`). Horizontal alignment can not be used for curved labels.
- Line height, in `points` or `percent` unit, e.g. "line-height: 40pt"or "line-height: 40%"
- Background properties such as `background-color` and `background-image`. They are supported for block type items (e.g. `div`) or inline items (e.g. `span`). For images, the CSS should be formatted as `background-image: url(xx)` and supports local file paths, HTTP links, or base64 embedded content.

> **주의**
>
> Backgrounds are not supported for curved text and are always rendered above any background shape for the label, and below drop shadows/buffers properties.

- Margin properties, available for block type items only, such as `div`, `p`, `h1`,⋯They can be specified in either the longhand or shorthand way, in `points` unit only. Negative values can be set for the bottom margin.

```
<div class="myDiv">
  <h2 style="margin-left: 5pt; margin-right: 10pt">QGIS still rocks...</h2>
  <p style="margin: 5pt 0pt -10pt 0pt">Thanks to you!!</p>
</div>
```

CSS properties can be set on HTML tags with the `style` attribute. The HTML tag `span` does not apply any formatting to text by itself and is ideal if you just want to apply CSS styling. A CSS property name and its value are separated by a colon (`:`). Multiple CSS properties are separated by semicolon (`;`), e.g.:

```
<span style="text-decoration:underline;text-align:center;color:blue;word-spacing:20">
↪I will be displayed as blue underlined and centered text with increased space␣
↪between words</span>
```

다음은 (동일한 라벨에 서로 다른 색상 및 밑줄을 적용한) HTML 기반 표현식과 렌더링의 예시입니다:

```
format(
  '<span style="color:blue">%1</span> ( <span style="color:red"><u>%2 ft</u></span> )
↪',
  title( lower( "Name" ) ),
  round($length)
)
```

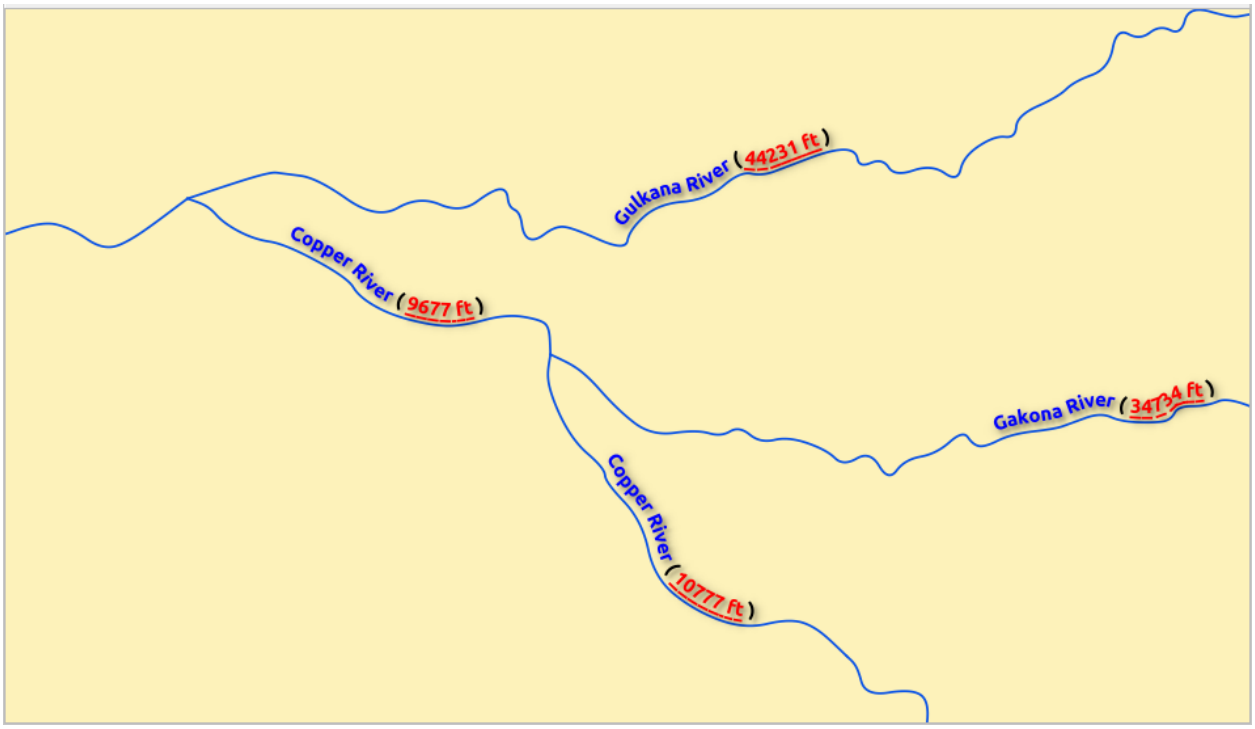



그림 10.26: HTML 서식을 활성화한 라벨 작업

### 서식 탭

*Formatting* 탭에서 다음 항목을 설정할 수 있습니다:

- 텍스트의 대소문자 스타일을 변경하려면 *Type case* 옵션을 사용하십시오. 텍스트를 다음과 같이 렌더링할 수 있습니다:
  - *No change*: 변경 사항 없음
  - *All uppercase*: 모두 대문자로 변경
  - *All lowercase*: 모두 소문자로 변경
  - *Small Caps*: renders lowercase characters as small caps.
  - *All Small Caps*: renders all characters as small caps regardless of their original case.
  - *Title case*: 원본 텍스트가 대/소문자만 사용하는 경우, 각 단어의 첫 문자를 대문자로, 다른 문자들을 소문자로 조정합니다. 원본 텍스트가 대/소문자를 섞어 사용하는 경우, 첫 문자를 제외한 다른 문자들은 변경하지 않습니다.
  - *Force first letter to capital*: 텍스트의 각 단어의 첫 문자를 대문자로 변경하고, 다른 문자들은 변경하지 않습니다.

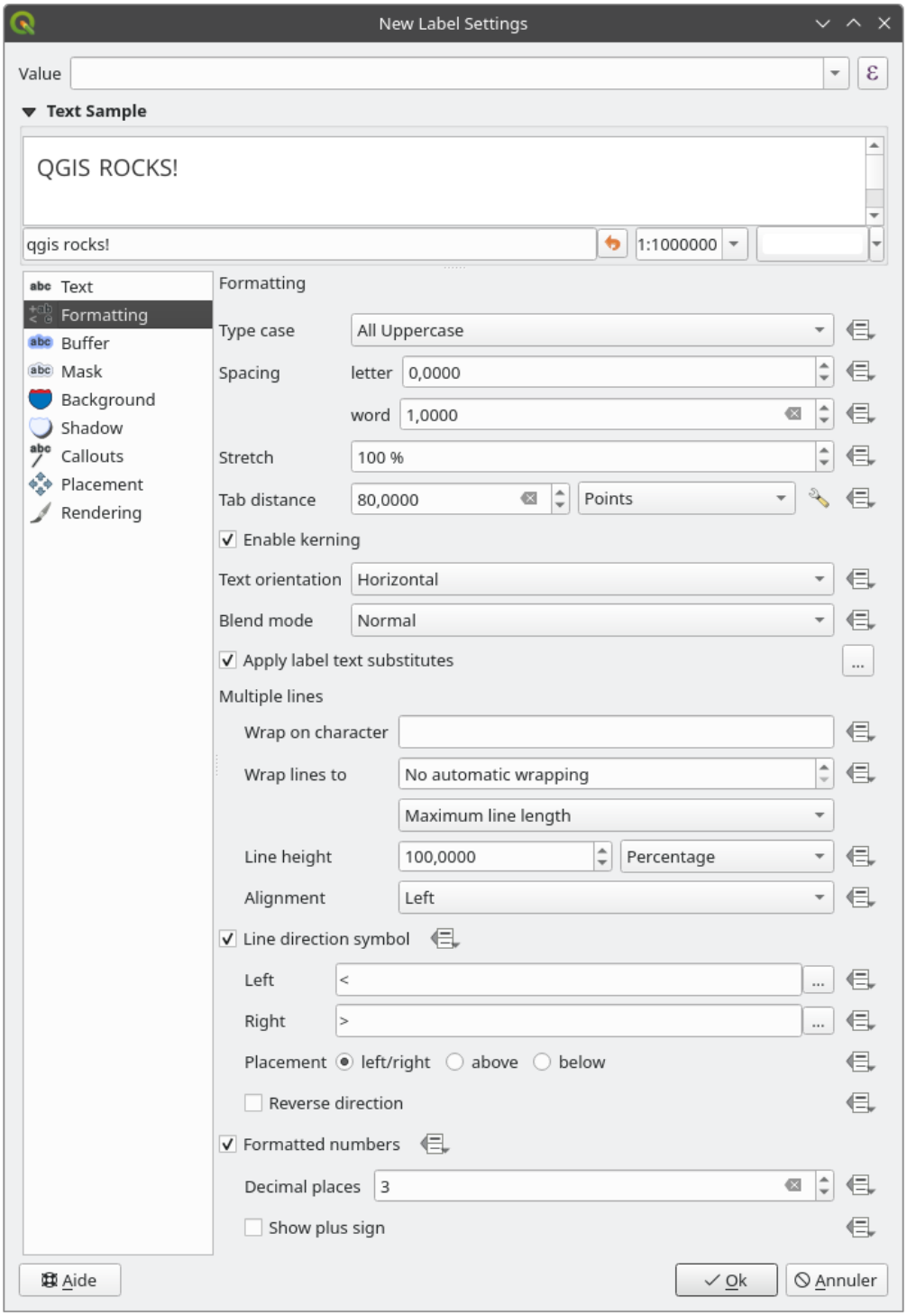



그림 10.27: 라벨 설정 - 서식 탭

- 단어들 사이의 여백, 개별 문자 사이의 여백을 변경하려면 *Spacing* 옵션을 사용하십시오.
- With *Tab distance* you can adjust the replacement spacing for `Tab` characters in the label, in any supported unit. This allows e.g. to properly separate or vertically align elements of a layout legend text.

  Pressing the Configure tab stops button on the right, you can add a list of tab positions, instead of a single distance. This allows e.g. the creation of table-like multiline labels, where texts are split over different "columns"and lined up nicely, based on the `Tab` characters.
- *Stretch* 비율: 인자를 통해 텍스트를 수평 방향으로 늘리거나 좁힐 수 있습니다. 라벨에 약간의 추가적인 텍스트를 맞추기 위해 글꼴 너비를 정밀 조정하는 데 편리합니다.
- *Enable kerning*: 텍스트 글꼴의 장식 꼬리 (f, y 등 활자의 몸체에서 돌출한 부분) 를 활성화합니다.
- *Text orientation*: 텍스트의 방향을 *Horizontal* 수평 또는 *Vertical* 수직으로 설정할 수 있습니다. 라벨 설정 시 (예를 들어, 라인 피처를평행 배치 모드로 적절히 라벨 작업하기 위해) *Rotation-based* 를 선택할 수도 있습니다.
- 사용자 라벨이 밑에 있는 맵 객체와 어떻게 혼합될지 설정하려면 *Blend mode* 옵션을 사용하십시오. (자세한 내용은혼합 모드 를 참조하세요.)
- *Apply label text substitutes*: 이 옵션은 객체 수준에 따라 라벨 텍스트를 대체할 (예를 들어 도로 유형을 축약한다던지 하는) 텍스트 목록을 지정할 수 있습니다. 맵 상에 라벨을 표시할 때 대체 텍스트를 사용합니다. 대체 텍스트를 재사용하거나 쉽게 공유하기 위해 대체 텍스트 목록을 내보내거나 가져올 수도 있습니다.
- *Multiple lines* 환경 설정:
  - *Wrap on character* 옵션을 통해 강제로 텍스트를 줄바꿈하는 문자를 정의할 수 있습니다.
  - *Wrap lines to* 옵션을 사용하면 자동 줄바꿈을 위한 이상적인 행 길이를 설정할 수 있습니다. 행 길이는 *Maximum line length* 또는 *Minimum line length* 가운데 하나로 표현할 수 있습니다.
  - *Line Height* 결정: 줄 높이 값을 *Millimeters*, *Points*, *Pixels*, *Percentage*, 또는 *Inches* 단위 가운데 하나로 설정할 수 있습니다. 줄 높이를 백분율로 설정하면, 해당 글꼴 계열의 기본 텍스트 줄 간격의 백분율을 의미합니다. 줄 높이는 일반적으로 텍스트 크기의 1.2 에서 1.5 배입니다.
  - *Alignment* 서식: 사용할 수 있는 전형적인 값은 *Left* 왼쪽 정렬, *Right* 오른쪽 정렬, *Justify* 양쪽 정렬, 그리고 *Center* 가운데 정렬입니다.

    포인트 라벨 속성을 설정할 때, *Follow label placement* 옵션을 선택하면 라벨 배치에 따라 텍스트를 정렬할 수도 있습니다. 이 경우, 포인트에 대한 라벨의 최종 배치에 따라 텍스트를 정렬할 것입니다. 예를 들어 라벨이 포인트 왼쪽에 위치한다면 라벨 텍스트를 오른쪽 정렬하고, 오른쪽에 위치한다면 왼쪽 정렬할 것입니다.

  **참고**

  곡선 기반라벨 배치 는 아직 *Multiple lines* 서식을 지원하지 않습니다. 지원하게 될 때 이 옵션은 비활성화될 것입니다.
- 라인 라벨의 경우 라인의 방향을 결정할 수 있는 *Line direction symbol* 를 *Left* 또는 *Right* 를 가리키는 심볼로 포함시킬 수 있습니다. *Placement* 탭에서 이 심볼을 **Curved** 또는 **Parallel** 배치 옵션과 함께 사용하면 더 좋습니다. 심볼 위치를 설정할 수 있는 옵션, 그리고 방향을 역전시킬 수 있는 *Reverse direction* 옵션도 존재합니다
- 숫자 텍스트 서식을 설정하려면 *Formatted numbers* 옵션을 활성화시키십시오. *Decimal places* 를 통해 소수점 숫자 개수를 설정할 수 있습니다. 기본값은 소수점 이하 3 자리입니다. *Show plus sign* 을 활성화하면 양수 앞에 플러스 표시를 추가할 수 있습니다.

### 버퍼 탭

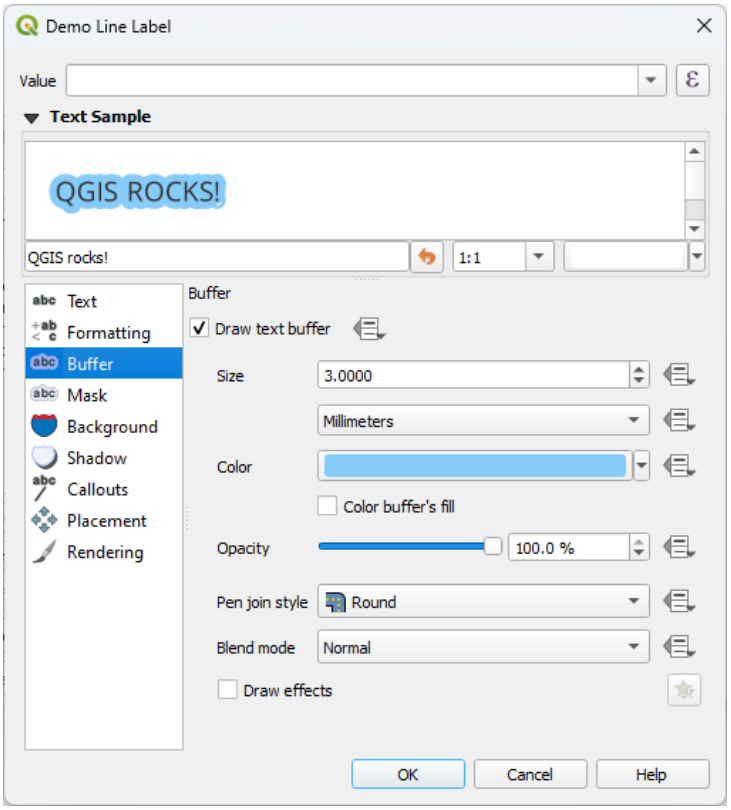



그림 10.28: 라벨 설정 - 버퍼 탭

라벨 주위에 버퍼 (테두리) 를 생성하려면, *Buffer* 탭에서 *Draw text buffer* 체크박스를 활성화하십시오. 다음을 설정할 수 있습니다:

- *Size*: 버퍼의 크기를 모든지원 단위 로 설정할 수 있습니다.
- *Color*: 버퍼의 색상을 설정할 수 있습니다.
- *Color buffer's fill*: 버퍼는 라벨의 외곽선으로부터 바깥쪽으로 확장하기 때문에 이 옵션을 활성화하면 라벨 내부를 채웁니다. 라벨 텍스트 밑에 뭐가 있는지 볼 수 있게 해주는 투명도 값을 가진 라벨 또는 특수한 혼합 모드를 사용하고 있을 경우 이 채우기 옵션이 의미가 있을 수도 있습니다. 완전히 투명한 라벨을 사용하는 동안 이 체크박스를 비활성화시키면, 외곽선으로 표시된 텍스트 라벨을 생성할 수 있습니다.
- *Opacity*: 버퍼의 투명도를 선택할 수 있습니다.
- *Pen join style* 적용: *Round* 원호, *Miter* 직각 또는 *Bevel* 경사 가운데 하나를 선택할 수 있습니다.
- 사용자 라벨의 버퍼가 밑에 있는 맵 객체와 어떻게 혼합될지 설정하려면 *Blend mode* 옵션을 사용하십시오. (자세한 내용은혼합 모드 를 참조하세요.)
- 고급 그리기 효과 를 추가하려면 *Draw effects* 옵션을 체크하십시오. 외곽 빛내기 (glow) 나 흐리기 (blur) 등을 통해 텍스트 가독성을 향상시킬 수 있습니다.

### 배경 탭

*Background* 탭에서 각 라벨 밑에 위치한 형태를 환경 설정할 수 있습니다. 배경을 추가하려면, *Draw Background* 체크박스를 활성화한 다음 *Shape* 유형을 선택하십시오. 다음 가운데 하나를 선택할 수 있습니다:

- *Rectangle*, *Square*, *Circle* 또는 *Ellipse*: 채우기 심볼 의 완전한 속성을 사용하는 직사각형, 정사각형, 원, 또는 타원 같은 정규 형태
- *SVG*: 파일 또는 URL 로 불러오는, 또는 프로젝트 또는 스타일 데이터베이스에 내장된 (자세한 내용은내장 파일 선택기 참조) SVG 심볼
- 심볼 라이브러리 에서 생성하거나 선택할 수 있는 *Marker Symbol*

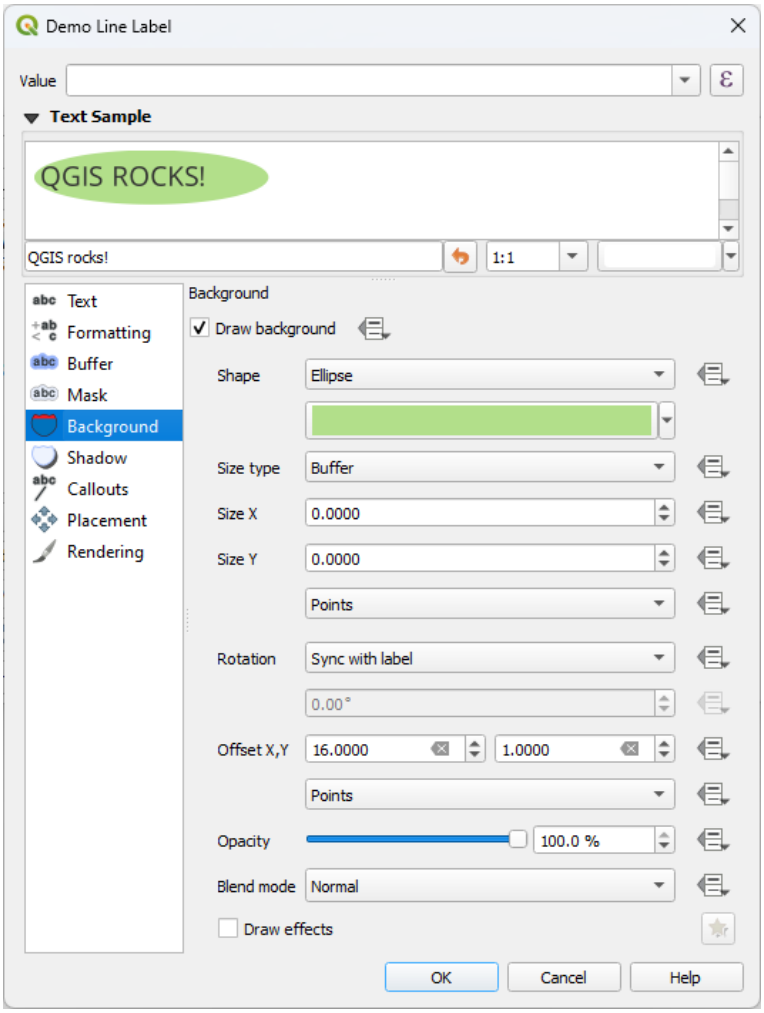



그림 10.29: 라벨 설정 - 배경 탭

선택한 형태에 따라 다음 속성 가운데 일부를 환경 설정해야 합니다:

- *Size type*: 프레임의 크기 유형을 선택할 수 있습니다:
  - *Fixed*: 텍스트 크기에 상관없이 모든 라벨에 동일한 크기를 적용합니다.
  - *Buffer*: 또는 텍스트의 경계 상자 위에 버퍼를 적용합니다.
- *Size*: 프레임의 크기를 X 및 Y 방향으로, 모든지원 단위 로 설정합니다.

- *Rotation*: 배경의 기울기를 *Sync with label*, *Offset of label* 그리고 *Fixed* 가운데 하나로 설정합니다. 뒤의 두 옵션은 도 단위로 각도를 지정해줘야 합니다.
- *Offset X,Y*: 배경 항목을 X 그리고/또는 Y 방향으로 이동시킵니다.
- *Radius X,Y*: 배경 형태의 모서리를 둥글게 만듭니다. (직사각형 및 정사각형 형태에만 적용할 수 있습니다.)
- *Opacity*: 배경의 투명도를 설정할 수 있습니다.
- *Blend mode*: 렌더링 시 배경과 다른 항목을 혼합할 수 있습니다. (혼합 모드 참조)
- SVG 심볼의 경우, 기본 속성 (*Load symbol parameters*) 을 이용하거나 사용자 지정 *Fill color*, *Stroke color* 및 *Stroke width* 를 설정할 수 있습니다.
- *Draw effects*: 텍스트 가독성을 향상시키기 위한 고급 그리기 효과 를 추가할 수 있습니다.

### 그림자 탭

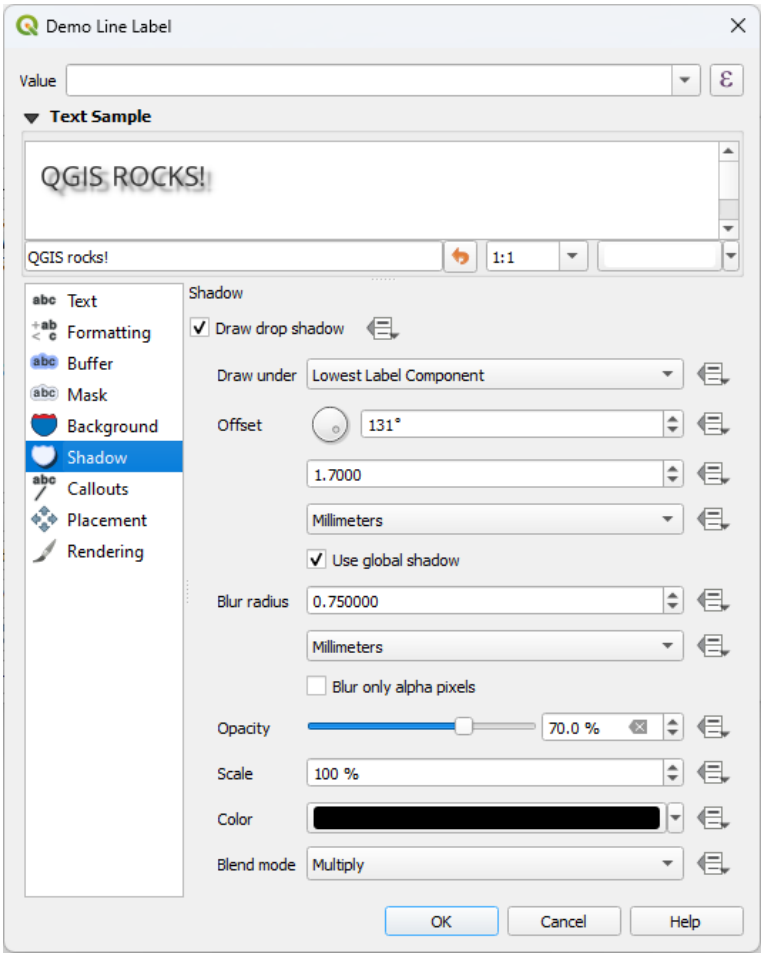



그림 10.30: 라벨 설정 - 음영 탭

텍스트에 그림자를 추가하려면, *Shadow* 탭을 선택한 다음 *Draw drop shadow* 옵션을 활성화하십시오. 다음을 설정할 수 있습니다:

- *Draw under* 옵션으로 그림자를 생성할 항목을 지정하십시오. *Text* 자체, *Buffer* 또는 *Background* 와 같은 특정 요소, 또는 *Lowest label component* 가운데 하나를 선택할 수 있습니다.

- Set the shadow's *Offset* from the item being shadowed, ie:
  - 각도: 그림자가 질 각도를 기저 항목의 방향에 따라 시계 방향으로 설정합니다.
  - The distance of offset from the item being shadowed
  - 오프셋 거리의 단위를 선택합니다.

  ☑ *Use global shadow* 체크박스를 체크하면, 각도의 영점이 라벨 항목의 방향을 따르지 않고 언제나 북쪽을 향하도록 고정됩니다.
- *Blur radius*: 그림자의 모양을 조정할 수 있습니다. 사용자가 선택한 단위의 숫자를 높이면 그림자가 옅어집니다.
- *Opacity*: 그림자의 투명도를 선택할 수 있습니다.
- *Scale* 인자를 사용해서 그림자의 크기를 재조정할 수 있습니다.
- *Color*: 그림자의 색상을 선택할 수 있습니다.
- 사용자 라벨의 그림자가 밑에 있는 맵 객체와 어떻게 혼합될지 설정하려면 *Blend mode* 옵션을 사용하십시오. (자세한 내용은혼합 모드 를 참조하세요.)

### 10.3.2 라벨의 상호 작용을 환경 설정하기

앞에서 설명한 텍스트 서식 설정 이외에도, 라벨이 다른 라벨 또는 피처와 어떻게 상호 작용할지 설정할 수 있습니다.

#### 마스크 탭

abc *Mask* 탭에서 라벨 주변에 마스크 영역을 정의할 수 있습니다. 심볼들이 중첩하는데 라벨들이 비슷한 색상이기 때문에 라벨들이 잘 보이도록 하고 싶을 때 매우 유용한 기능입니다. 라벨 마스크는 마스크에 설정된 경계 안쪽에 지정한 피처를 그리지 않도록 막습니다. 예를 들면, 라벨의 2mm 안쪽에 지정한 레이어를 그리지 않지만 또다른 레이어의 피처는 보이도록 라벨 마스크를 설정할 수 있습니다. 라벨 마스크는 다른 피처를 덮고 있는 라벨의 가독성을 제어할 수 있게 해준다는 점에서 라벨 버퍼와 비슷합니다. 둘 사이의 차이점은 라벨 버퍼는 아래에 있는 모든 피처들 위에 생성되지만, 라벨 마스크는 다른 레이어들이 그려지지 않도록 선택할 수 있다는 점입니다.

라벨에 마스크 효과를 생성하려면:

1. abc 탭에서 ☑ *Enable mask* 체크박스를 활성화하십시오.
2. 그러면 다음을 설정할 수 있습니다:
   - *Size*: 마스크의 크기를지원 단위 로 설정할 수 있습니다.
   - *Opacity*: 라벨 주위의 마스크 영역의 투명도를 설정할 수 있습니다.
   - *Pen Join Style*
   - ☑ *Draw effects* 체크박스를 체크하면그리기 효과 를 설정할 수 있습니다.
3. 중첩하는 레이어의 속성 가운데 abc *Mask* 탭에서 이 마스크 형태를 마스크 소스로 선택하십시오. (마스크 속성 참조)

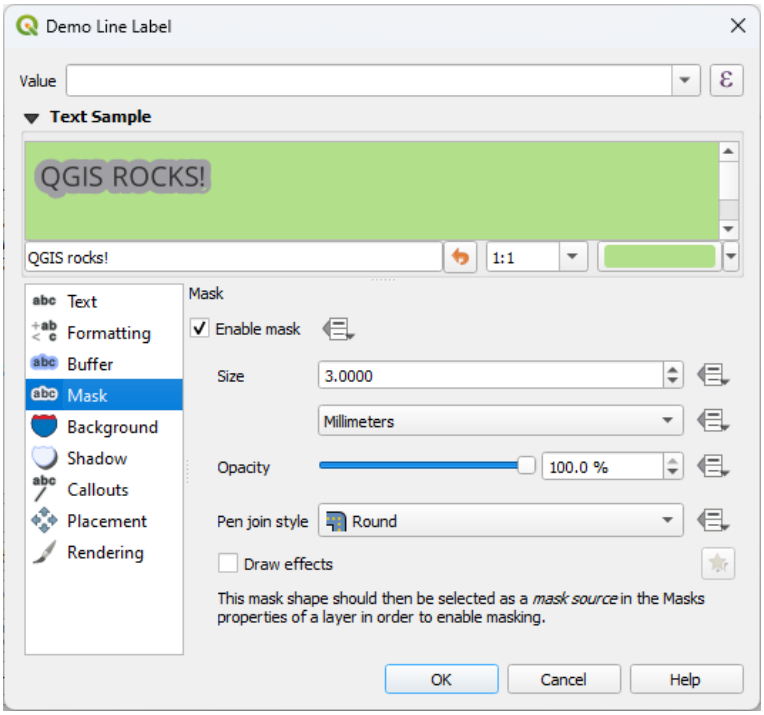



그림 10.31: 라벨 설정 - 마스크 탭 (텍스트 예시가 또다른 레이어가 제외되어 있음을 나타내는 녹색 배경을 보여주고 있습니다)

### 호출 탭

복잡한 맵 위에 라벨을 배치할 때 흔히 **호출 (callout)** 을 사용하곤 합니다. 관련 피처에서 떨어진 (또는 외부에 배치된) 라벨을 피처와 라벨을 연결하는 동적 라인으로 식별합니다. 두 종단점 가운데 하나를 (라벨 또는 피처 가운데 하나가) 이동시키면, 연결자 (connector) 의 형태를 다시 계산합니다.

라벨에 호출을 추가하려면, *Callouts* 탭을 선택한 다음 *Draw callouts* 옵션을 활성화하십시오. 다음을 설정할 수 있습니다:

1. 연결자의 *Style* 을 다음 가운데 선택합니다:
   - *Simple lines*: 직선, 최단 경로
   - *Manhattan style*: 90° 꺾인 라인
   - *Curved lines*: 만곡 라인
   - *Balloons*: 라벨을 감싸고 피처를 가리키는 말풍선. 모서리를 둥글게 만들 수 있습니다.
2. Set the properties of the callout. The following table shows the different properties, with description and compatible connector style.

표 10.5: Label callout properties

| Property | Style of callout | 설명 |
|---|---|---|
| *Fill style* | Balloons | A *fill symbol* with full display capabilities, including layer effects, data-defined settings, ⋯for drawing the balloon shape. |
| *Corner radius* | | Corner radius of the speech bubble |

다음 페이지에 계속

표 10.5 – 이전 페이지에서 계속

| Property | Style of callout | 설명 |
|---|---|---|
| *Wedge width* | | Sets how large the bubble speech connection with feature's pointer should be |
| *Margins* | | Margins around the label's text, in the unit of your choice |
| *Line style* | All but balloons | A *line symbol* with full display capabilities, including layer effects, data-defined settings, ⋯for drawing the connector line. |
| *Curvature* | Curved lines | The percentage of curvature of the connection line |
| *Orientation* | | Orientation, starting from the label to the feature. It can be *Clockwise*, *Counter-clockwise*, or *Automatic* (determining an optimal orientation for each label). |
| *Minimum length* | All but balloons | Minimum length of the connector line |
| *Draw lines to all feature parts* | | In case of a multi-part feature, indicates whether a connector line should be drawn from the label to each geometry part. |
| *Label anchor point* | | Controls where the connector line should join to the label text. Available options:<br>• *Closest point*: 가장 가까운 포인트에 연결합니다.<br>• *Centroid*: 중심점<br>• 가장자리에 있는 고정 위치 (*Top left*, *Top center*, *Top right*, *Left middle*, *Right middle*, *Bottom left*, *Bottom center* and *Bottom right*) 에 연결합니다. |
| *Offset from label area* | | Controls the distance from the label anchor point (where the callout line ends). This avoids drawing lines right up against the text. |
| *Offset from feature* | 모두 | Controls the distance from the feature (or its anchor point if a polygon) where callout lines end. E.g., this avoids drawing lines right up against the edges of the features. |
| *Feature anchor point* | | Where the connector line ends on the (polygon) feature. Available options:<br>• *Pole of inaccessibility*: 도달불능극<br>• *Point on exterior*: 외곽 포인트<br>• *Point on surface*: 표면 포인트<br>• *Centroid*: 중심점 |
| *End point marker* | Balloons | A *marker symbol* with full display capabilities including layer effects, data-defined, ⋯for rendering a marker symbol below the endpoint of the balloon callout. |
| *Blend mode* | 모두 | Controls the *blending* of the callout. |

*Data defined placement* 프레임에서, 호출의 (라벨 쪽) *Origin* 그리고/또는 (피처 쪽) *Destination* 포인트 좌표를 제어할 수 있습니다. 라벨 작업 툴바 에 있는 Move Label, Diagram or Callout 도구로도 호출의 배치를 직접 제어할 수 있습니다. 이런 방법으로 각 호출의 시작점 및 종단점을 이동시킬 수 있습니다. 마우스 커서를 가까이 가져가면 노드가 강조될 것입니다. 필요한 경우, 호출을 이동시키는 동안 `Shift` 키를 누르고 있으면 두 호출 포인트 사이의 각도가 15 도씩 늘어나는 방식으로 포인트를 스냅시킬 것입니다.

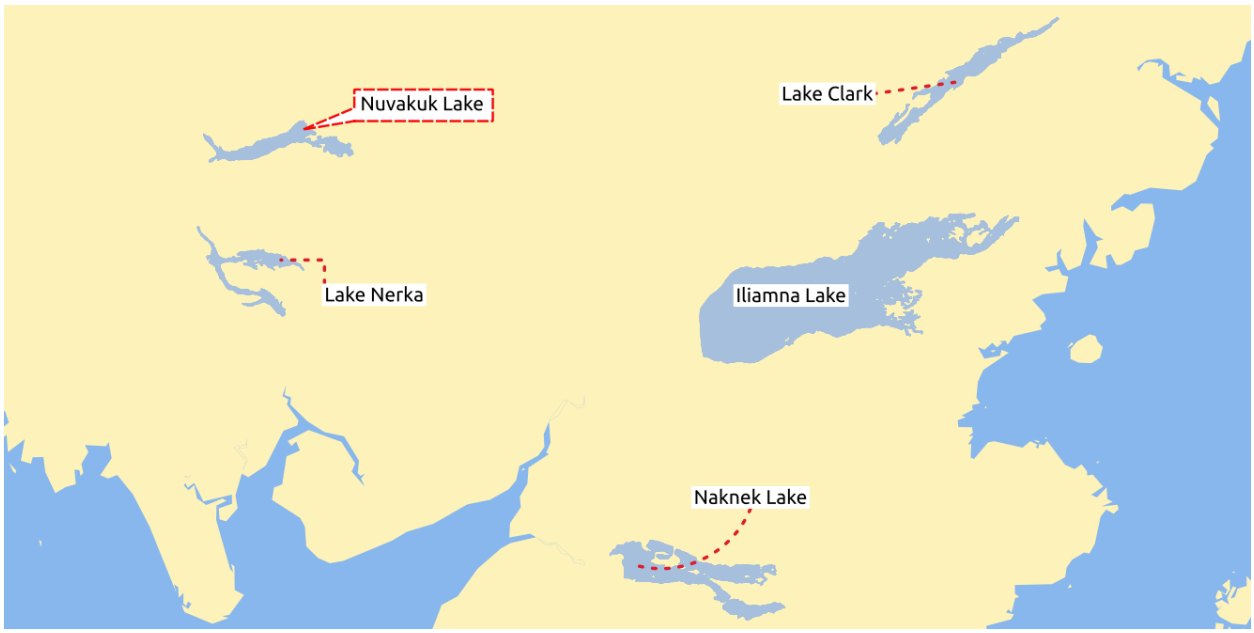



그림 10.32: 다양한 호출 설정을 가진 라벨

### 배치 탭

Choose the *Placement* tab for configuring label placement and labeling priority. Note that the placement options differ according to the type of vector or mesh layer, namely point, line or polygon, and are affected by the global *PAL setting*.

### 포인트 레이어에서의 배치

사용할 수 있는 포인트 라벨 배치 모드는 다음과 같습니다:

- *Cartographic*: 이상적인 지도 제작법의 배치 규칙을 따라 포인트 피처와 시각적으로 더 잘 어울리는 위치에 포인트 라벨을 생성합니다. 라벨을 다음과 같이 배치할 수 있습니다:
  - 포인트 피처 자체로부터 또는 (*Distance offset from* 옵션에서 설정하는) 피처를 표현하는 데 쓰인 심볼의 경계로부터지원 단위 로 설정한 *Distance* 에 라벨을 배치할 수 있습니다. 심볼 크기가 고정되지 않은 경우, 예를 들어 데이터 정의 크기로 설정됐거나 또는범주 렌더링 작업자에서 서로 다른 심볼을 사용하는 경우 심볼 경계를 기준으로 하는 옵션이 매우 유용합니다.
  - within a *Maximum Distance* from the feature, which is an optional setting that allows you to control how far a label can be placed from the feature it's labeling. This works alongside the *Distance* setting to create a range for label placement, adding flexibility to position labels more effectively, especially on busy maps, ensuring they fit neatly around their corresponding features.
  - using the *Prioritize Placement* option, which decides what's more important when placing labels. There are two options:
    - *Prefer closer labels*: By default, labels are kept close to the feature.
    - *Prefer position ordering*: The label will try to stay in a specific position (like top left or top right), even if it's a bit farther away from the feature. The label only moves to other positions if there's no room within the maximum distance at your preferred position.

  - following a *Position priority* which dictates placement candidates for anchoring labels around and (centered) over the point feature, and the order in which the positions are tested. The default order, based on guidelines from Krygier and Wood (2011) and other cartographic textbooks, is as follows:

    1. 우상단
    2. 좌상단
    3. 우하단
    4. 좌하단
    5. 우중단
    6. 좌중단
    7. 약간 우측으로 쏠린 상단
    8. 약간 좌측으로 쏠린 하단

    Using the Data-defined override button, you can provide a comma separated list of placements in order of priority. This also allows only certain placements to be used, for certain features only, so e.g., for coastal features you can prevent labels being placed over the land.

- *Around Point*: labels are placed in a circle around the feature with an equal radius set in *Distance*. Additionally you can set *Maximum Distance* from the feature, to control how far a label can be placed from the feature it's labeling. The placement priority is clockwise from the "top right". The position can be constrained using the data-defined *Quadrant* option.
- *Offset from Point*: 포인트 피처에서 다양한 단위의 *Offset X,Y* 거리에, 또는 가급적 피처 위로 라벨을 배치할 수 있습니다. 배치를 제약하기 위해 데이터 정의 *Quadrant* 옵션을 사용할 수 있고, 라벨에 *Rotation* 을 할당할 수 있습니다.

### 라인 레이어에서의 배치

라인 레이어 용 라벨 모드는 다음을 포함합니다:

- *Parallel*: 피처를 표현하는 일반화된 라인에 평행하게 라벨을 그립니다. 라인 가운데 좀 더 직선에 가까운 부분 위로 들어가는 배치를 선호합니다. 다음을 정의할 수 있습니다:
  - *Allowed positions*: *Above line*, *On line*, *Below line* 및 *Line orientation dependent position* (라인의 좌측 또는 우측에 라벨 배치). 동시에 여러 옵션을 선택할 수 있습니다. 이런 경우, QGIS 가 최적의 라벨 위치를 찾을 것입니다.
  - *Distance*: 라벨과 라인 사이의 거리
- *Curved*: 라인 피처의 곡률에 따라 라벨을 그립니다. *Parallel* 모드에서 사용할 수 있는 파라미터는 물론, 라벨을 내부에 그릴지 또는 외부에 그릴지, 그리고 *Maximum angle between curved characters polygon* 을 통해 각 문자 사이의 최대 각도도 정의할 수 있습니다.
- *Horizontal*: 라인 피처의 길이를 따라 라벨을 수평으로 그립니다.

배치 모드 이외에도 다음을 설정할 수 있습니다:

- *Repeating Labels Distance*: 피처의 길이에 걸쳐 라벨을 여러 번 표시합니다. 라벨 사이의 거리를 `Millimeters`, `Points`, `Pixels`, `Meters at scale`, `Map Units` 및 `Inches` 단위로 정의할 수 있습니다.
- A *Label Overrun Distance* (수평 모드에서는 사용할 수 없음): 라인 피처의 종단 (또는 시작) 을 넘어 연장할 수 있는 라벨의 최대 허용 거리를 지정합니다. 이 값을 늘리면 짧은 라인 피처에 라벨을 표시할 수 있게 됩니다.
- *Label Anchoring*: 라벨이 지시하는 라인 피처에서의 라벨의 배치를 제어합니다. *Settings* ⋯를 클릭하면 다음 가운데 선택할 수 있습니다:

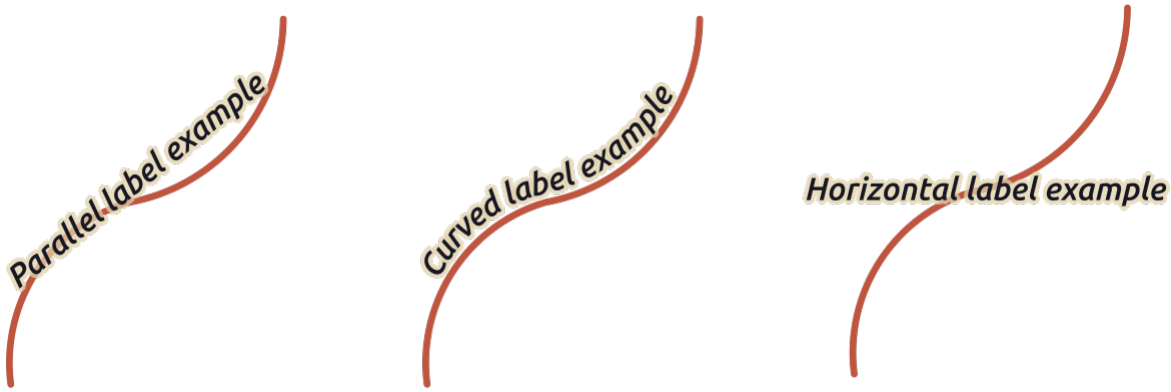



그림 10.33: 라인 피처의 라벨 배치 예시

- 라벨이 가까이 배치될 라인에서의 위치를 (비율로) 선택합니다. 데이터 정의를 할 수 있으며, 선택할 수 있는 값은 다음과 같습니다:
  - *Center of Line*
  - *Start of Line*
  - *End of Line*
  - 또는 *Custom…*
- *Clipping*: 라인 상에서 라벨 배치를 어떻게 계산할지를 결정합니다. 기본적으로 라인의 가시 영역만 사용하지만, 좀 더 일관적인 결과를 내기 위해 전체 영역을 사용할 수도 있습니다.
- *Anchor text*: 텍스트의 어느 (시작, 중앙, 또는 종단) 부분을 고정점에 정렬시킬지를 제어합니다. *Automatic* 앵커를 사용한다면:
  - 라인의 시작점에 가깝게 (0~25%) 고정된 라벨의 경우 라벨 텍스트의 **시작** 부분에 앵커를 배치할 것입니다.
  - 라인의 종단점에 가깝게 (75~100%) 고정된 라벨의 경우 라벨 텍스트의 **종단** 부분에 앵커를 배치할 것입니다.
  - 라인의 중앙에 가깝게 (25~75%) 고정된 라벨의 경우 라벨 텍스트의 **중앙** 부분에 앵커를 배치할 것입니다.
- *Placement Behavior*: *Preferred Placement Hint* 를 사용해서 라벨 고정점을 라벨 배치를 위한 힌트 정도로만 취급할 수 있습니다. *Strict* 를 선택하면 라벨을 라벨 고정점에 정확히 배치합니다.

### 폴리곤 레이어에서의 배치

폴리곤의 라벨을 배치하는 데 다음 모드 가운데 하나를 선택할 수 있습니다:

- *Offset from Centroid*: 라벨을 피처의 중심점에, 또는 피처의 중심점에서 (지원 단위 의) 고정 *Offset X,Y* 거리에 배치합니다. 맵 캔버스에 렌더링된 폴리곤의 부분을 기반으로 (*visible polygon*) 또는 사용자가 볼 수 있느냐에 상관 없이 폴리곤 전체를 기반으로 (*whole polygon*) 참조 중심점을 결정할 수 있습니다. 또한:
  - 중심점 포인트를 해당 폴리곤의 내부로 강제할 수 있고
  - 지정 사분면 내부에 라벨을 배치할 수 있으며

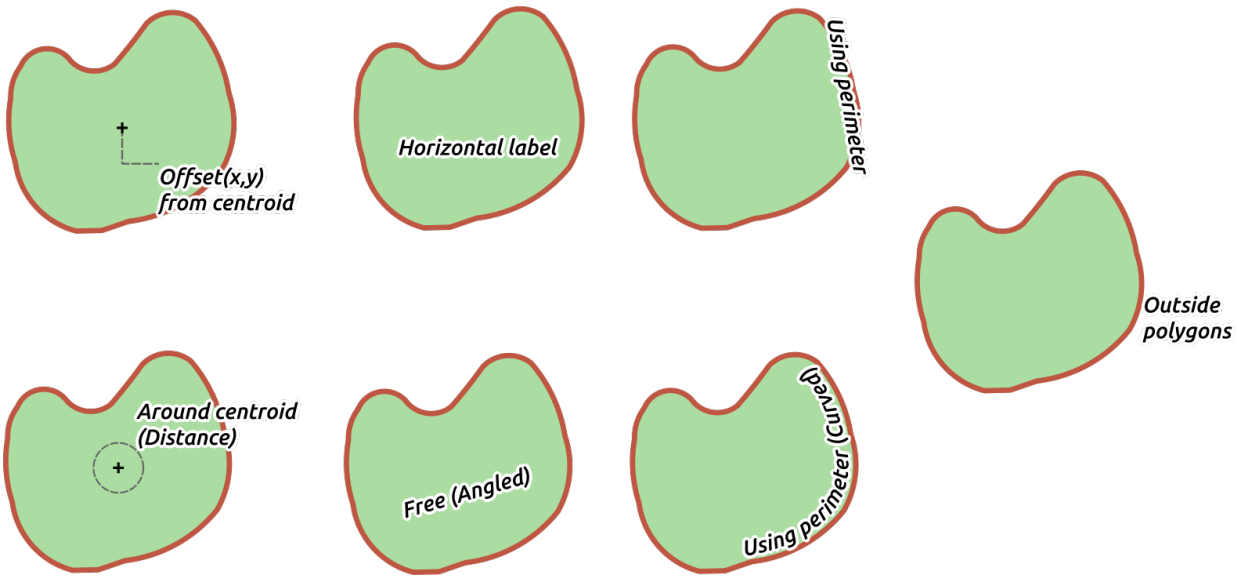



그림 10.34: 폴리곤 피처의 라벨 배치 예시

  - 기울기를 할당할 수 있습니다.
  - *Allow placing labels outside of polygons*: 라벨을 폴리곤 내부에 배치할 수 없는 경우 폴리곤 외부에 라벨을 배치합니다. 데이터 정의 속성 덕분에 외부 라벨을 허용할 수도, 방지할 수도, 또는 피처 별로 외부 라벨을 강제할 수도 있습니다.

- *Around Centroid*: 중심점 주변으로 사전 설정한 거리 안에 라벨을 배치합니다. 정확히 중심점에 배치하는 것을 선호합니다. 역시, 중심을 *visible polygon* 또는 *whole polygon* 기반으로 정의할 수 있으며, 중심점 포인트를 폴리곤 내부로 강제할지 여부도 정의할 수 있습니다.
- *Horizontal*: 폴리곤 내부의 최적 위치에 수평 라벨을 배치합니다. 폴리곤 경계에서 가장 먼 위치에 배치하는 것을 선호합니다. *Allow placing labels outside of polygons* 옵션으로 라벨을 폴리곤 외부에 배치할 수도 있습니다.
- *Free (Angled)*: 폴리곤 내부의 최적 위치에 기울인 라벨을 배치합니다. 기울기는 폴리곤의 방향을 따르며, 폴리곤 경계에서 가장 먼 위치에 배치하는 것을 선호합니다. *Allow placing labels outside of polygons* 옵션으로 라벨을 폴리곤 외부에 배치할 수도 있습니다.
- *Using Perimeter*: 폴리곤 경계선을 표현하는 일반화된 라인에 평행하게 라벨을 그립니다. 경계선 가운데 좀 더 직선에 가까운 부분 위로 들어가는 배치를 선호합니다. 다음을 정의할 수 있습니다:
  - *Allowed positions*: *Above line*, *On line*, *Below line* 및 *Line orientation dependent position* (폴리곤 경계선의 좌측 또는 우측에 라벨 배치). 동시에 여러 옵션을 선택할 수 있습니다. 이런 경우, QGIS 가 최적의 라벨 위치를 찾을 것입니다.
  - *Distance*: 라벨과 폴리곤 외곽선 사이의 거리
  - the *Repeating Labels Distance*: 경계선의 길이에 걸쳐 라벨을 여러 번 표시합니다.
- *Using Perimeter (Curved)*: 폴리곤 경계선의 곡률에 따라 라벨을 그립니다. *Using Perimeter* 모드에서 사용할 수 있는 파라미터는 물론, 라벨을 내부에 그릴지 또는 외부에 그릴지, 그리고 *Maximum angle between curved characters polygon* 을 통해 각 문자 사이의 최대 각도도 정의할 수 있습니다.
- *Outside Polygons*: 항상 폴리곤의 외부의 설정한 *Distance* 에 라벨을 배치합니다.

### 공통 배치 설정

몇몇 라벨 배치 설정을 모든 레이어 도형 유형에 대해 사용할 수 있습니다:

#### Label Spacing

The *Label Spacing* option allows you to set the margin around the label, preventing all other labels from being placed within that area. You can dynamically control *Margin around label* using the Data-defined override button. Choose unit from the unit selector, and there you can also *adjust scaling range*.

> **주의**
>
> This option is only available on QGIS installed with GEOS >= 3.10 (see *Help → About* menu).

#### Duplicate Labels

Use *Avoid duplicate labels* to improve label appearance in situations like road networks, where features such as dual carriageways or service roads can result in many identical labels appearing close together. When active, it prevents placing labels with the exact same text (case-sensitive) closer than the minimum distance defined in the *Settings*⋯ option. The setting applies across layers, so duplicate text from any layer will be considered.

> **주의**
>
> This option is only available on QGIS installed with GEOS >= 3.10 (see *Help → About* menu).

### 도형 생성기

*Geometry Generator* 프레임에서, 사용자는표현식 을 이용해서 라벨을 배치하고 렌더링하는 데 쓰이는 기저 도형을 수정할 수 있습니다. 도형을 동적으로 이동시키거나, 또다른 도형 (유형) 으로 변환시키는 데 유용할 수 있습니다.

도형 생성기를 사용하려면:

1. *Geometry generator* 옵션을 체크하십시오.
2. 도형 생성에 사용할 표현식을 입력하십시오.
3. 관련이 있는 경우, 표현식이 산출할 도형 유형을 선택하십시오: 배치 또는 렌더링 같은 도형 기반 라벨 설정을 새로운 도형 유형의 케이퍼빌리티와 일치하도록 업데이트합니다.

다음은 몇몇 활용 사례입니다:

- 또다른 필드인"label_position"에 저장되어 있는 도형을 사용
- 라벨 작업에 심볼로부터생성된 도형 도 사용
- 확대/축소 수준과 상관없이 거리/크기를 계산하기 위해 @map_scale 변수를 사용
- 만곡 배치 모드와 결합, 포인트 피처를 둘러싼 원호 라벨을 생성:

```
exterior_ring(make_circle($geometry, 20))
```

- 라인 피처의 시작점과 종단점에 라벨을 추가:

```
collect_geometries( start_point($geometry), end_point($geometry) )
```

- 라벨 배치를 위한 공간을 확보하기 위해 강을 나타내는 라인을 평활화:

```
smooth( $geometry, iterations:=30, offset:=0.25, min_length:=10 )
```

### 데이터 정의

*Data Defined* 그룹은 라벨 배치에 대해 피처 별로 직접 제어할 수 있는 기능을 제공합니다. 데이터 정의는 라벨의 속성 또는 다음을 설정할 표현식에 의존적입니다:

- *X* 및 *Y* 좌표
- 앞에서 설정한 사용자 지정 위치에서의 텍스트 정렬:
  - *Horizontal*: **Left**, **Center** 또는 **Right** 가운데 하나를 선택할 수 있습니다.
  - *Vertical*: **Bottom**, **Base**, **Half**, **Cap** 또는 **Top** 가운데 하나를 선택할 수 있습니다.
- the text *Rotation*. Rotation is defined as clockwise angle with 0° pointing in the direction of East for `Horizontal` oriented text and with 0° pointing in North direction for `Vertical` oriented text. Different units can be defined for the labeling rotation (e.g. `degrees`, `minutes of arc`, `turns`). QGIS prioritizes screen display rotation angles by default and always rotates text to be most readable on screen, so "upside-down" rotation angles are overridden while rendering. To change this behavior and force QGIS to render exact data-defined rotation angles there is an option in *Rendering* tab, under *Show upside-down labels*. Check the *Preserve data rotation values* entry if you want to keep the rotation value in the associated field and apply it to the label, whether the label is pinned or not. If unchecked, unpinning the label rotation is reset and its value cleared from the attribute table.

> **참고**
>
> 폴리곤 피처의 데이터 정의 기울기는 현재 *Around centroid* 배치 모드에서만 지원합니다.

> **참고**
>
> 데이터 정의 라벨 배치를 위한 라벨 맵 도구 (예: *Rotate label* 및 *Move label* 도구) 와 함께 표현식을 사용할 수는 없습니다. 위젯이 해당보조 저장소 필드 로 재설정될 것입니다.

### 우선 순위

*Priority* 부분에서 각 라벨의 배치 우선 순위를 정의할 수 있습니다. 예를 들어 서로 다른 도표 또는 라벨 후보들이 동일 위치에 존재하는 경우, 더 높은 우선 순위를 가진 항목이 표시되고 나머지 후보들은 표시되지 않을 수도 있습니다.

우선 순위는 더 높은 가중치를 가진방해물 피처 때문에 라벨이 누락되어도 될지 평가하는 데에도 쓰입니다.

### 방해물

어떤 맥락에서는 (예를 들어 라벨이 높은 밀도로 모여 있는 경우, 피처와 중첩하는 경우 등등) 라벨이 관계 없는 다른 피처를 중첩하는 위치에 배치될 수도 있습니다.

방해물이란 QGIS 가 그 위로 다른 피처의 라벨 또는 도표를 배치하지 않도록 하는 피처를 말합니다. *Obstacles* 부분에서 이 기능을 제어할 수 있습니다:

1. 레이어의 피처가 (동일 레이어에 있는 다른 피처의 항목을 포함한) 모든 라벨 및 도표의 방해물로서 동작해야 하는지 여부를 결정하려면 *Features act as obstacles* 옵션을 활성화하십시오.

   이 옵션 옆에 있는 Data-defined override 제어를 이용하면, 방해물로 사용할 피처 하위 집합을 전체 레이어 대신 선택할 수 있습니다.

2. 방해물의 가중치를 조정하려면 *Settings* 버튼을 사용하십시오.

   - 모든 잠재적인 방해물 피처에 대해 *Obstacle weight* 를 할당할 수 있습니다: 이 값보다 높은 배치 우선 순위를 가진 모든라벨 또는도표 는 방해물을 중첩할 수 있습니다. 이보다 낮은 우선 순위를 가진 라벨 또는 도표는, 다른 위치에 배치가 불가능한 경우, 누락될 것입니다.

     이 가중치도 데이터 정의할 수 있습니다. 즉 동일 레이어 안에서 특정 피처가 다른 피처보다 중첩될 가능성이 높게 만들 수 있습니다.

   - 폴리곤 레이어의 경우, 피처가 어떤 유형의 방해물인지 선택할 수 있습니다:

     - **Over the feature's interior**: 폴리곤 내부에 라벨이 배치되지 않도록 합니다. (라벨을 완전히 폴리곤 바깥에 배치하거나 폴리곤 내부에 살짝만 걸치도록 배치하는 것을 선호합니다.)
     - **Over the feature's boundary**: 폴리곤의 경계선을 가로질러 라벨이 배치되지 않도록 합니다. (라벨을 폴리곤 바깥에 또는 완전히 폴리곤 내부에 배치하는 것을 선호합니다.) 이 옵션은 피처가 전체 영역을 덮고 있는 (행정 구역, 범주 커버리지 등의) 레이어에 유용합니다. 이런 경우 피처 내부에 라벨을 배치해야 하고, 라벨이 피처들 사이의 경계선을 가로질러 배치되는 일을 피해야 더 보기에도 좋습니다.

### 렌더링 탭

*Rendering* 탭에서, 언제 라벨을 렌더링할 수 있는지 그리고 어떻게 다른 라벨 및 피처와 상호작용하는지를 조정할 수 있습니다.

### 라벨 옵션

*Label options* 에서는:

- 축척에 따라 그리고 픽셀 크기에 따라 가시성을 설정할 수 있는축척 기반 및 *Pixel size-based* 옵션이 있습니다.
- *Label z-index* 옵션은 라벨 렌더링 순서는 물론, 다른 레이어의 라벨과의 관계를 설정하는 것처럼 (데이터 정의 무시 표현식을 이용해서) 동일 레이어에 있는 다른 피처 라벨과의 관계성도 결정합니다. 더 낮은 Z 인덱스를 가진 (모든 레이어의) 라벨 위에 더 높은 Z 인덱스를 가진 라벨을 렌더링합니다.

  여기에, 라벨 2 개가 일치하는 Z 인덱스를 가지도록 논리가 수정됐을 경우:

  - 동일 레이어의 라벨인 경우, 더 큰 라벨 위에 더 작은 라벨을 그립니다.
  - 서로 다른 레이어의 라벨인 경우, 레이어 자체의 우선 순위 (예를 들어 맵 범례에서의 순서) 를 따릅니다.

> **참고**
>
> 이 설정은 다른 레이어의 피처 아래에 라벨을 그리도록 하는 것이 아니라, 그저 모든 레이어의 피처 위에 라벨을 어떤 순서로 렌더링할지 제어할 뿐입니다.

- *Allow inferior fallback placements*: QGIS 는 기본적으로 라벨을 사용자 설정에 따라 최적의 위치에 렌더링하려 시도합니다. 이 모드를 체크하면, 예를 들어 만곡 라벨 텍스트를 맞추기에는 라인이 너무 짧아서 피처 중심 포인트 바로 위에 라벨을 수평으로 배치할 수도 있는 경우처럼, 다른 선택지가 없는 경우 피처를 더 보기 안 좋은 위치로 배치하게 할 수도 있습니다.
- *Show label* 및 *Always Show* 에서 데이터 정의 표현식을 이용하면, 어떤 라벨을 렌더링해야 하는지를 미세 조정할 수 있습니다.
- Allow to *Show upside-down labels*: alternatives are **never**, **when rotation defined** or **always**.
  - **never** - default setting, screen readability is prioritized,
  - **when rotation defined** - label rotation should be defined under *Placement* tab, within the *Data Defined* group
  - **always** - upside-down labels are allowed

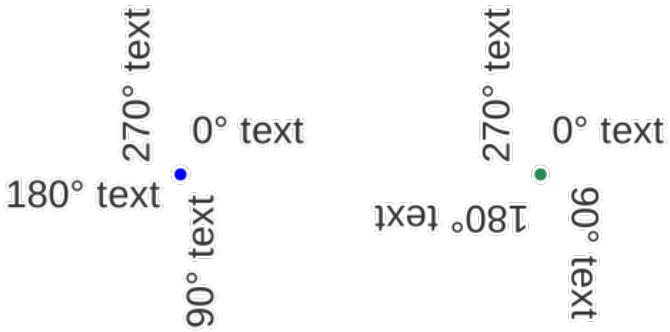



그림 10.35: Data defined label rotation with show upside-down labels option set to: “never”(left), “when rotation defined” (right)

- *Overlapping labels* 프레임에서는 레이어에 있는 피처에 대해 라벨이 겹쳐져도 되는지 여부를 그리고 각각의 라벨을 어떻게 처리해야 할지를 제어할 수 있습니다.
  - *Never overlap*: 라벨 일부가 누락되는 한이 있더라도 레이어의 라벨들을 절대로 겹치게 배치하지 않습니다.
  - *Allow overlaps if required*: 라벨을 다른 방식으로는 배치할 수 없는 경우, 겹치는 라벨을 렌더링합니다. 이 모드는 가능한 경우 라벨을 조금 덜 이상적인 위치로 이동시킬 것입니다. 예를 들어 라벨을 라인 또는 폴리곤의 중심으로부터 더 멀리 이동시켜 다른 라벨과 겹치지 않게 할 수 있다면, 그렇게 할 것입니다. 그러나 가능한 위치가 달리 없는 경우 라벨을 겹치게 렌더링합니다.
  - *Allow overlaps without penalty*: 한 라벨이 다른 라벨과 겹치거나 또는 가려도 아무 문제없습니다. 라벨을 좀 더 먼 위치로 이동시켜 서로 겹치지 않게 할 수 있더라도, 항상 최적의 배치를 (예를 들어 가장 중심에 가까운 배치를) 사용할 것입니다.

  겹치는 라벨과 차선의 배치를 둘 다 허용하면, 라벨이 레이어의 모든 피처에 표시되도록 보장할 것입니다… 가장 보기 좋은 렌더링이 아닐지라도 말이죠!

#### 피처 옵션

*Feature options* 에서는:

- 다중 부분 피처의 모든 부분에 라벨을 그리는 *Label every part of a multi-part features* 옵션을 선택할 수도 있고, *Limit number of features to be labeled to* 옵션을 선택해서 라벨을 그릴 피처의 개수를 제한할 수도 있습니다.
- 라인 및 폴리곤 레이어 둘 다 라벨을 그릴 피처의 최소 크기를 설정할 수 있는 *Suppress labeling of features smaller than* 옵션을 제공합니다.
- 폴리곤 피처의 경우, 라벨이 피처 내부에 완전히 들어가는지 여부를 바탕으로 표시할 라벨을 필터링할 수도 있습니다.
- 라인 피처의 경우, *Merge connected lines to avoid duplicate labels* 옵션을 선택하면배치 탭 에 있는 *Distance* 또는 *Repeat* 옵션과 결합해서 훨씬 보기 좋은 맵을 렌더링할 수 있습니다.

## 10.4 3 차원 심볼 생성

*Style Manager* 를 통해*3* 차원 맵 뷰 에서 렌더링될 모든 도형 유형의 3 차원 심볼을 생성하고 저장할 수 있습니다.

다른 항목들과 마찬가지로,  *3D Symbols* 탭을 선택하고  버튼 메뉴를 확장해서 다음을 생성할 수 있습니다:

- *3* 차원 포인트 심볼
- *3* 차원 라인 심볼
- *3* 차원 폴리곤 심볼

### 10.4.1 포인트 레이어

- 포인트 심볼에 사용할 서로 다른 유형의 3 차원 *Shape* 을 정의할 수 있습니다. 3 차원 형태는 주로 프로젝트 좌표계를 참조하는 단위를 가진 차원으로 정의됩니다. 다음과 같은 유형을 정의할 수 있습니다:
  - 반경 (*Radius*) 으로 정의되는 구 (*Sphere*) 형태
  - 반경 (*Radius*) 과 길이 (*Length*) 로 정의되는 원통 (*Cylinder*) 형태
  - 크기 (*Size*) 로 정의되는 정육면체 (*Cube*)
  - 상단 반경 (*Top radius*), 하단 반경 (*Bottom radius*) 그리고 길이 (*Length*) 로 정의되는 원뿔 (*Cone*) 형태
  - 크기 (*Size*) 로 정의되는 평면 (*Plane*)
  - 반경 (*Radius*) 과 작은 반경 (*Minor radius*) 으로 정의되는 *Torus* (원환면) 형태
  - 3 차원 모델 파일을 사용하는 *3D Model*: 지원 포맷은 파면 (wavefront) `.obj`, `.glTF` 및 `.fbx` 파일을 포함합니다. 모델은 디스크 상에 저장된 파일일 수 있고, 원격 URL 일 수도 있으며, 프로젝트에 내장 되어 있을 수도 있습니다. QGIS 허브 에서 커뮤니티가 생성한 모델들을 공유하고 있습니다.
  - 빌보드 높이 (*Billboard height*) 와 (일반적으로마커 심볼 을 바탕으로 하는) 빌보드 심볼 (*Billboard symbol*) 로 정의되는 빌보드 (*Billboard*). 이 심볼의 크기는 정적입니다. 3 차원 점구름 형태를 가시화하는 데 편리합니다.
- 고도 클램핑 (*Altitude clamping*) 은 *Absolute*, *Relative* 또는 *Terrain* 가운데 하나로 설정할 수 있습니다. 3 차원 벡터의 높이 값이 0 에서 시작하는 절대 측정값으로 제공되는 경우 *Absolute* 설정을 사용할 수 있습니다. *Relative* 또는 *Terrain* 설정은 기저 지형 표고에 지정한 표고값을 추가합니다.

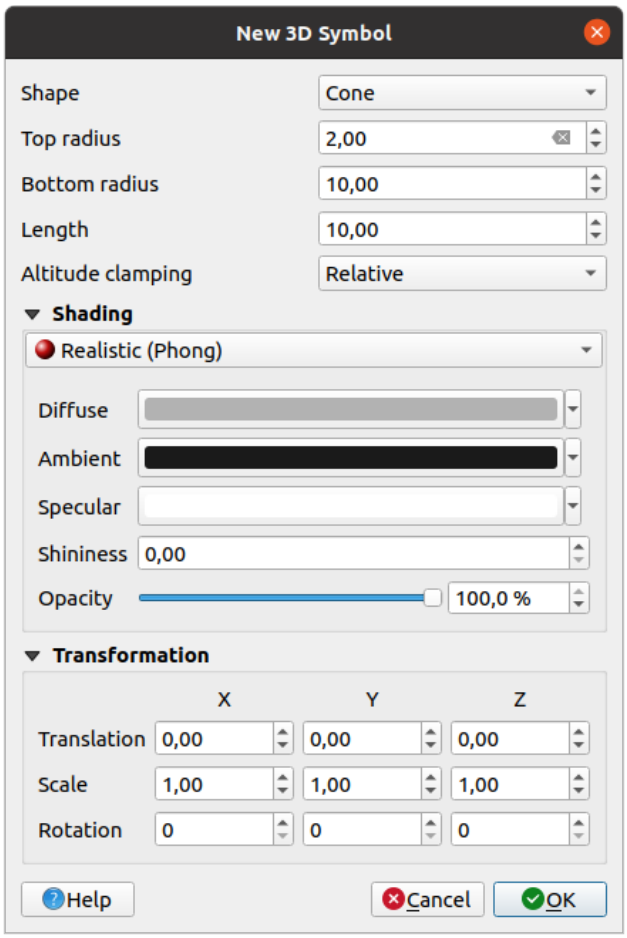



그림 10.36: 3 차원 포인트 심볼의 속성

- 음영 속성을 정의할 수 있습니다.
- *Transformations* 프레임에서 심볼에 아핀 변환을 적용할 수 있습니다:
  - *Translation*: x, y, z 축에서 객체를 이동시킵니다.
  - *Scale*: 3 차원 형태의 크기를 조정합니다.
  - *Rotation*: x, y, z 축을 중심으로 3 차원 형태를 회전시킵니다.

### 10.4.2 라인 레이어

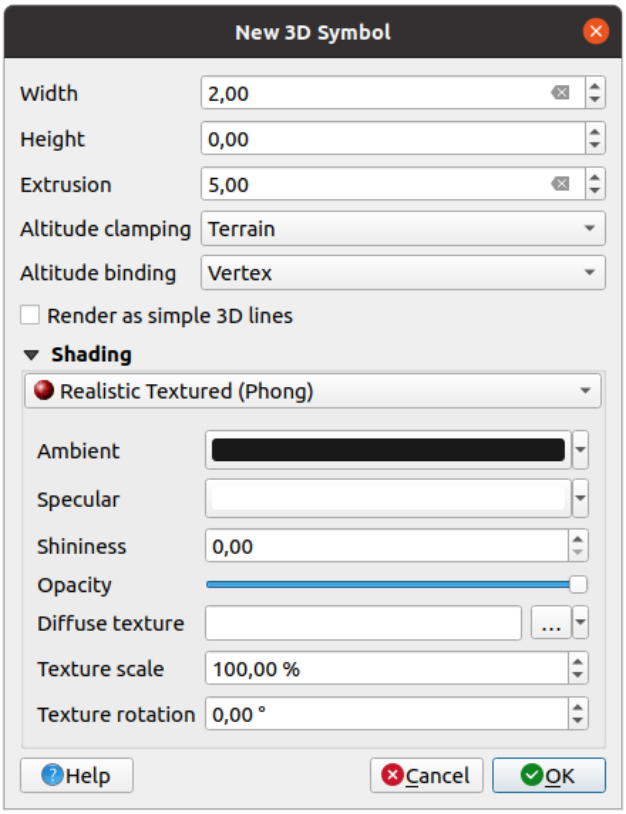



그림 10.37: 3 차원 라인 심볼의 속성

- *Width* 및 *Height* 설정 밑에서 벡터 라인의 *Extrusion* 을 정의할 수 있습니다. 라인이 Z 값을 가지지 않을 경우, 이 설정을 통해 3 차원 부피를 정의할 수 있습니다.
- 사용자가 래스터 표고 데이터 또는 기타 3 차원 벡터를 포함시킨 경우, *Altitude clamping* 을 통해 3 차원 라인의 기저 지표면에 상대적인 위치를 정의하십시오.
- *Altitude binding* 설정은 피처를 어떻게 지형에 고정시킬지를 정의합니다. 피처의 모든 *Vertex* 를 지형에 고정시키거나, 또는 *Centroid* 를 고정시킬 것입니다.
- *Render as simple 3D lines*: 단순 3 차원 라인으로 렌더링할 수도 있습니다.
- 음영 속성을 정의할 수 있습니다.

### 10.4.3 폴리곤 레이어

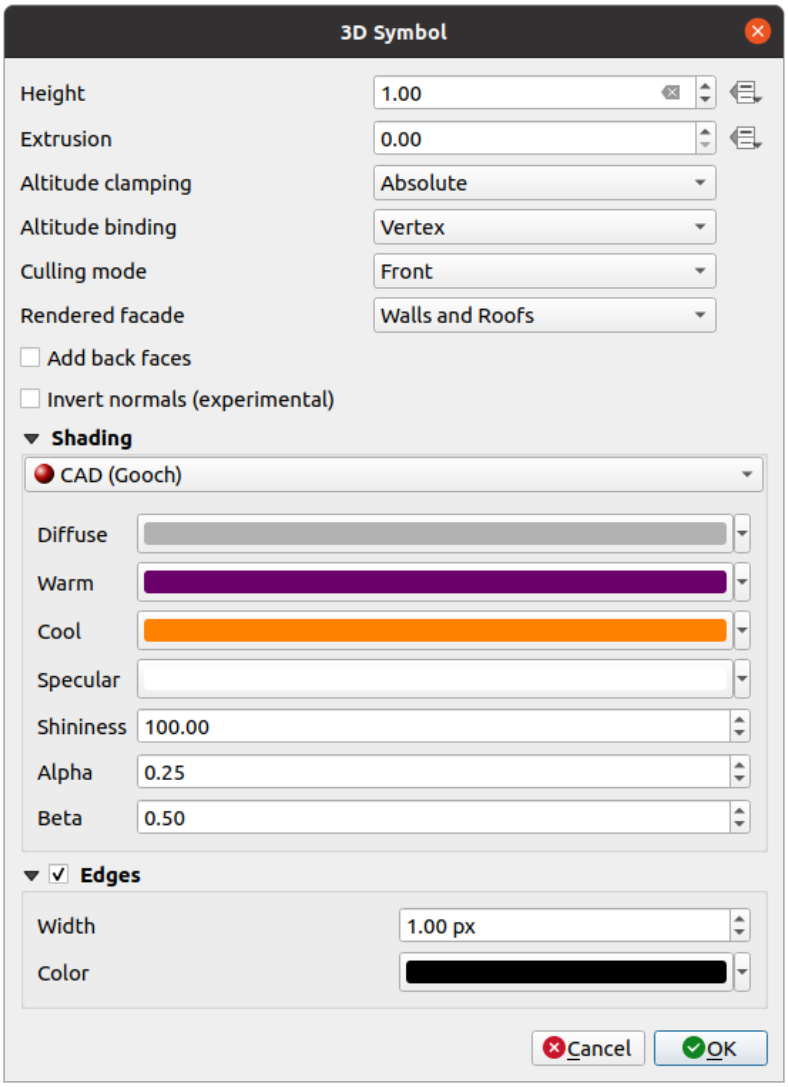



그림 10.38: 3 차원 폴리곤 심볼의 속성

- 다른 심볼들과 마찬가지로, *Height* 를 좌표계 단위로 정의할 수 있습니다. 또한 버튼을 사용해서 사용자 정의 표현식, 변수, 또는 속성 테이블의 항목으로 값을 덮어쓸 수도 있습니다.
- *Extrusion* 을 사용하면 폴리곤에서도 누락된 Z 값을 압출할 수 있습니다. 또 압출 과정에서 벡터 레이어의 값을 이용하기 위해 버튼을 사용하면 각 폴리곤 별로 서로 다른 결과값을 얻을 수 있습니다:
- 앞에서 설명한 대로 *Altitude clamping*, *Altitude binding* 을 정의할 수 있습니다.
- *Culling mode*: 심볼에 컬링 모드를 정의할 수 있습니다. 다음 가운데 하나로 설정할 수 있습니다:
  - *No Culling*: polygonZ/멀티패치 (multipatch) 데이터의 꼭짓점들이 일관성 있는 순서로 (예를 들어 시계 방향 또는 반시계 방향으로) 되어 있지 않은 경우 누락된 것으로 보이는 표면을 방지하는 데 이 모드가 도움이 될 수 있습니다.
  - *Front*: 정면 (front-face) 컬링
  - *Back*: 배면 (back-face) 컬링

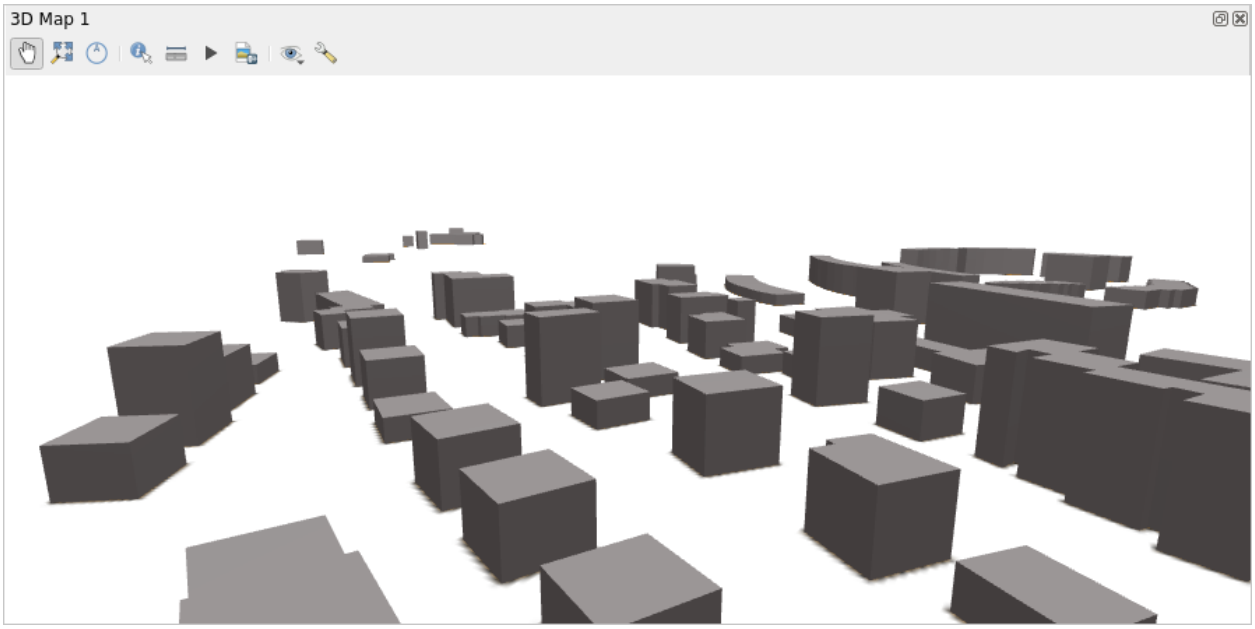



그림 10.39: 데이터 정의 압출

- *Rendered facade*: 표시할 면 (face) 을 결정합니다. *No facades*, *Walls*, *Roofs*, 또는 *Walls and roofs* 가운데 하나를 선택할 수 있습니다.
- ☑ *Add back faces*: 각 삼각형에 대해, 정확한 법선 (normal) 으로 정면 및 배면을 모두 생성합니다. 이때 꼭짓점 데이터의 개수가 늘어나 성능이 저하될 수도 있습니다. (예를 들어 꼭짓점 순서가 일관적이지 않은 데이터 때문에 발생한) 음영 문제를 해결하는 데 이 옵션을 사용할 수 있습니다.
- ☑ *Invert normals (experimental)*: 시계 방향/반시계 방향 정면 꼭짓점 순서를 해결하는 데 유용할 수 있습니다.
- 음영 속성을 정의할 수 있습니다.
- ☑ *Edges*: 이 옵션을 활성화하면 심볼 경계를 표시하며, 경계의 *Width* 및 *Color* 를 설정할 수 있습니다.

> **힌트**
>
> **3 차원 데이터를 최적으로 렌더링하기 위한 조합**
>
> *Culling mode*, *Add back faces* 그리고 *Invert normals* 모두 3 차원 데이터가 올바르게 보이지 않을 경우 3 차원 데이터의 모양을 수정하기 위한 설정들입니다. 일반적으로 데이터를 불러올 때 먼저 `culling mode=back` 과 `add back faces=disabled` 를 시도해보는 것이 좋습니다. 가장 효율적인 방법이기 때문입니다. 렌더링 결과가 올바르게 보이지 않는다면 `culling mode=no culling` 과 `add back faces=enabled` 를 시도해보십시오. 다른 조합들은 좀 더 고급 사용자를 위한 것으로, 입력 데이터셋이 얼마나 섞여 있느냐를 기반으로 하는 몇몇 시나리오에서만 유용합니다.

### 10.4.4 텍스처에 음영 만들기

음영 작업은 다른 상황에서는 신 (scene) 의 조명 때문에 보이지 않을 수도 있는 객체의 3 차원 세부 사항을 드러내는 데 도움이 됩니다. 궁극적으로, 피처를 시각화하기 위한 적절한 신 조명을 설정하느라 고민할 필요 없이 3 차원 심볼을 더 쉽게 작업할 수 있습니다.

QGIS 에서는 다양한 음영 기법을 사용하는데, 그 사용성은 심볼의 도형 유형에 따라 달라집니다:

- *Realistic (Phong)*: 거친 표면의 *Diffuse* 반사와 광택이 있는 (*Shininess*) 표면의 *Specular* 반사를 조합해서 표면이 빛을 반사하는 방식을 설명합니다. 신 전체에 확산되는 소량의 빛을 연산에 넣을 수 있는 *Ambient* 옵션도 포함하고 있습니다. 3 차원에서 반쯤 투명한 객체를 렌더링하려면 *Opacity* 슬라이드 바를 이용하십시오. 더 자세한 내용은 퐁 (Phong) 반사 모델에 대한 설명 을 읽어보세요.
- *Realistic Textured (Phong)*: 이미지를 *Diffuse Texture* 로 이용한다는 점을 제외하면 *Realistic (Phong)* 과 동일합니다. 이미지는 디스크 상에 저장된 파일일 수도 있고, 원격 URL 일 수도 있으며, 프로젝트에 내장되어 있을 수도 있습니다. *Texture scale* 과 *Texture rotation* 을 설정해줘야 합니다. 반쯤 투명한 3 차원 객체를 렌더링하려면 *Opacity* 슬라이드 바를 이용하십시오.
- *CAD (Gooch)*: 이 기법은 경계선이 계속 강조되고 시각적으로 두드러져 보이도록 음영이 중간 톤으로만 발생하게 해줍니다. *Diffuse*, *Specular*, *Shininess* 옵션 외에, (조명을 바라보는 표면을 위한) *Warm* 색상 및 (조명 반대 방향을 바라보는 표면을 위한) *Cool* 색상을 정의해야 합니다. 또한 *Alpha* 와 *Beta* 속성이 확산 (diffuse) 색상이 차가운 색상과 따뜻한 색상에 각각 미치는 상대적 기여도를 제어합니다. 구치 음영법 (Gooch shading) 에 대한 설명 도 읽어보세요.
- *Metal Roughness*: 빛과 표면이 어떻게 상호작용하는지에 대한 정확한 재현을 제공하는 물리 기반 렌더링 재질입니다. 재질 설정에 *Base color*, *Metalness* 및 *Roughness* 옵션을 사용할 수 있습니다.
- *Embedded Textures*: 3 차원 모델 형태를 가지고 있는 내장 텍스처

### 10.4.5 응용 프로그램 예시

앞에서 설명한 설정들을 자세히 알아보려면, https://app.merginmaps.com/projects/saber/luxembourg/tree 페이지에 있는 파일들을 다운로드 받아보십시오.

CHAPTER 11

# 데이터소스 관리

## 11.1 데이터 열기

오픈소스 소프트웨어 생태계의 일원으로서, QGIS 는 서로 다른 라이브러리들을 기반으로 빌드되어, 자체 제공자와 결합해 다음과 같은 여러 포맷들을 읽고, 많은 경우 작성할 수도 있는 능력을 갖추고 있습니다:

- Vector data formats include GeoPackage, GML, GeoJSON, GPX, KML, Comma Separated Values, ESRI formats (Shapefile, Geodatabase…), MapInfo and MicroStation file formats, AutoCAD DWG/DXF, GRASS and many more…Read the complete list of supported vector formats.
- Raster data formats include GeoTIFF, JPEG, ASCII Gridded XYZ, MBTiles, R or Idrisi rasters, GDAL Virtual, SRTM, Sentinel Data, ERDAS IMAGINE, ArcInfo Binary Grid, ArcInfo ASCII Grid, and many more…Read the complete list of supported raster formats.
- Database formats include PostgreSQL, SQLite/SpatiaLite, Oracle, MS SQL Server, SAP HANA, MySQL…
- QGIS 제공자는 WM(T)S, WFS, WCS, CSW, XYZ 타일, ArcGIS 서비스 등의 웹 맵 및 데이터 서비스 지원도 처리하고 있습니다. 자세한 정보를 알고 싶다면*OGC / ISO* 프로토콜 작업 을 참조하세요.
- 압축 폴더에서 지원하는 파일을 읽어올 수 있고, QML 파일 (*QML* → *QGIS* 스타일 파일 포맷), 가상 (virtual) 및 메모리 레이어 같은 QGIS 자체 포맷도 사용할 수 있습니다.

More than 80 vector and 140 raster formats are supported by GDAL and QGIS native providers.

> **참고**
>
> 목록에 있는 포맷 가운데 여러 가지 이유로 인해 QGIS 에서 작동하지 않는 포맷이 있을 수도 있습니다. 예를 들어 일부 포맷은 외부 상표등록 라이브러리가 필요할 수도 있고, 사용자 OS 에 GDAL/OGR 을 설치했을 때 사용자가 원하는 유형을 지원하도록 빌드되지 않았을 수도 있습니다. 사용할 수 있는 포맷 목록을 얻으려면, QGIS 에서 (벡터의 경우) `ogrinfo --formats` 명령어를 실행하거나 (래스터의 경우) *Settings* → *Options* → *GDAL* 메뉴를 확인하십시오.

QGIS 는 데이터 유형에 따라 서로 다른 도구로 데이터셋을 열 수 있습니다. 주로 *Layer → Add Layer →* 메뉴나 (*View → Toolbars* 메뉴에서 활성화할 수 있는) *Manage Layers* 툴바를 사용합니다. 하지만 이 모든 도구들은 결국 *Data Source Manager Toolbar* 에 있는 Open Data Source Manager 버튼이나 `Ctrl+L` 조합키로 열 수 있는 단일한 *Data Source Manager* 대화창으로 연결됩니다. 이 *Data Source Manager* 대화창은 파일 기반 데이터는 물론 QGIS 가 지원하는 데이터베이스나 웹 서비스도 열 수 있는 통합 인터페이스를 제공합니다. (그림 11.1 참조)

이런 주 진입점 (entry point) 외에, 연결된 데이터베이스를 분석하고 처리할 수 있는 고급 성능을 제공하는 *DB Manager* 플러그인도 있습니다. DB 관리자 성능에 대한 자세한 정보는데이터베이스 관리자 플러그인 에서 찾아볼 수 있습니다.

사용자가 다양한 데이터 유형들을 열 수 있도록 해주는 다른 도구들, 자체 또는 제 3 자 플러그인들도 다수 존재합니다.

이 장에서는 QGIS 가 데이터를 불러올 때 기본적으로 제공하는 도구들에 대해서만 설명할 것입니다. 주로 *Data Source Manager* 대화창에 집중하겠지만, 각 탭을 설명하는 데 그치지 않고 데이터 제공자 또는 포맷 사양에 기반한 도구들에 대해서도 알아볼 것입니다.

### 11.1.1 탐색기 패널

*Browser* 는 프로젝트에 사용자 데이터를 쉽고 빠르게 추가할 수 있는 주된 방법 가운데 하나입니다. 다음 방법으로 사용할 수 있습니다:

- Open Data Source Manager 버튼을 누르면 (`Ctrl+L` 조합키를 누르면) *Data Source Manager* 탭이 활성화됩니다.
- QGIS 의 패널이므로, *View → Panels* (또는 *Settings → Panels*) 메뉴를 통해 또는 `Ctrl+2` 조합키를 눌러 열 수 있습니다.

어떻게 열든, *Browser* 를 통해 사용자 파일 시스템을 둘러보고 레이어 유형 (래스터, 벡터, 테이블) 또는 데이터소스 유형 (일반 또는 압축 파일, 데이터베이스, 웹 서비스) 에 상관없이 지리 데이터 (geodata) 를 관리할 수 있습니다.

#### 인터페이스 탐구

탐색기 패널 상단에 있는 아이콘의 기능은 다음과 같습니다:

- Add Selected Layers: 레이어의 컨텍스트 메뉴에서 **Add selected layer(s)** 를 선택해도 맵 캔버스에 데이터를 추가할 수 있습니다.
- Refresh: 탐색 트리를 새로고칩니다.
- Filter Browser: 특정 데이터를 검색합니다. 검색어 또는 임의 문자 기호 (wildcard) 를 입력하면 탐색기가 트리를 필터링해서 입력 문자열과 일치하는 데이터베이스 테이블, 파일명 또는 폴더를 가리키는 경로만 표시합니다. 일치하지 않는 다른 데이터 또는 폴더는 숨깁니다. 그림 11.2 의 Browser Panel(2) 예시를 참조하십시오. 필터링 작업은 대소문자를 구분할 수도 있고 안 할 수도 있습니다. 다음 옵션으로 필터링을 설정할 수도 있습니다:
  - *Normal*: 검색어를 포함하는 모든 항목을 반환합니다.
  - *Wildcard(s)*: 검색 문자열에서 위치를 지정하는 `?` 그리고/또는 `*` 문자를 이용해서 검색을 세밀하게 조정합니다.
  - *Regular expression*: 정규 표현식을 이용합니다.
- Collapse All: 탐색 트리 전체를 접습니다.

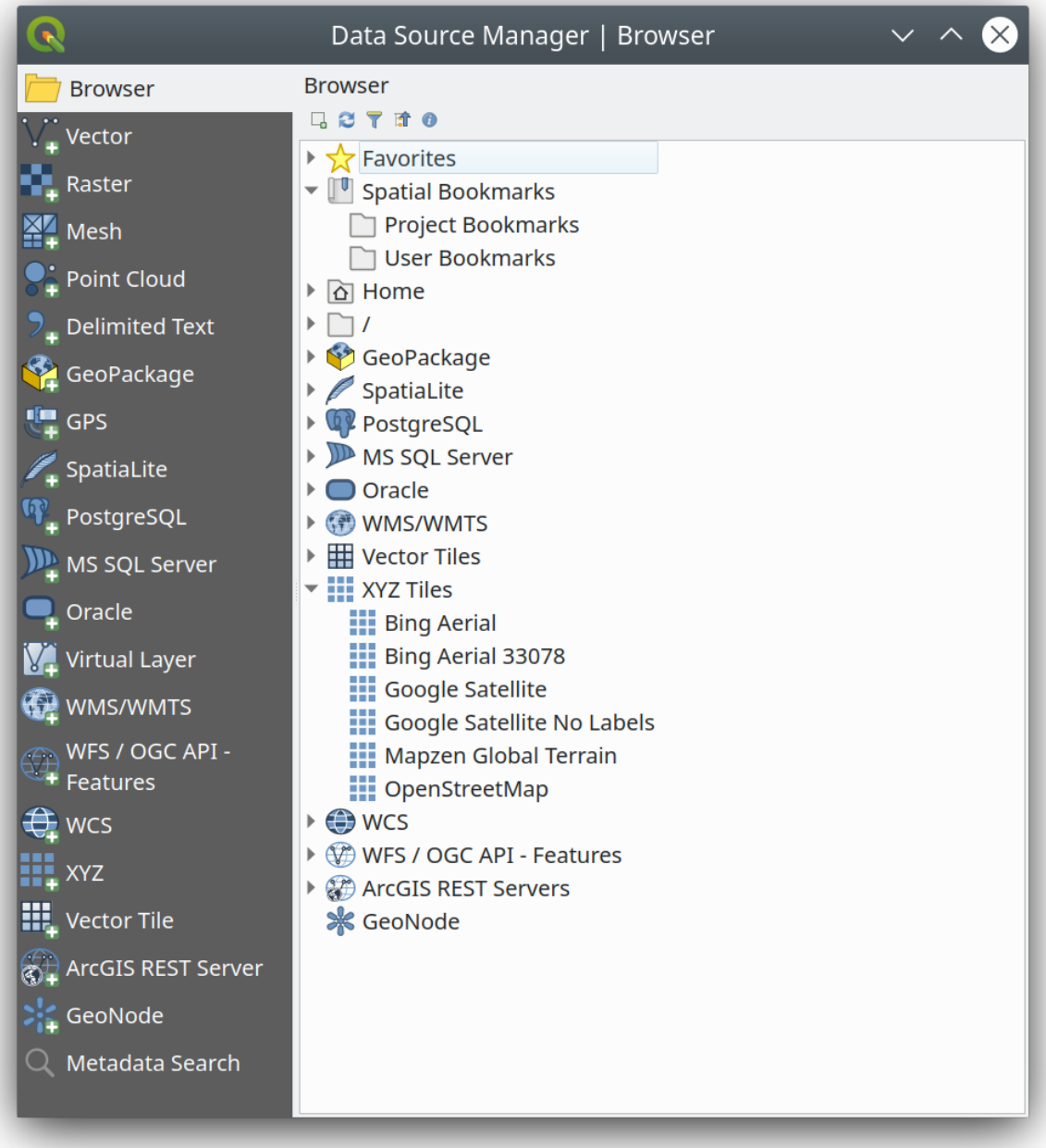



그림 11.1: QGIS 데이터소스 관리자 대화창

- Enable/disable properties widget: 속성 위젯을 켜고 끕니다. 활성화하면, 패널 하단에 선택한 항목의 (메타데이터가 있는 경우) 메타데이터를 표시하는 새 위젯을 추가합니다.

*Browser* 패널의 항목들은 위계에 따라 구성돼 있으며, 여러 최상위 수준 항목들이 있습니다:

1. *Favorites*: 자주 사용하는 위치를 가리키는 단축키를 배치할 수 있는 곳입니다.
2. *Spatial Bookmarks*: 자주 사용하는 맵 범위를 저장할 수 있는 곳입니다. (맵 상에서의 범위 북마크 작업 참조)
3. *Project Home*: 사용자 프로젝트와 관련된 (대부분의) 데이터가 저장돼 있는 폴더에 빠르게 접근할 수 있습니다. 기본값은 사용자의 프로젝트 파일이 위치한 디렉터리입니다.
4. *Home*: 파일 시스템의 홈 디렉터리 및 파일 시스템 루트 디렉터리
5. 연결된 로컬 또는 네트워크 드라이브
6. 그 다음 사용자 플랫폼과 기저 라이브러리에 따라 여러 컨테이너/데이터베이스 유형과 서비스 프로토콜이 나열됩니다:
   - *GeoPackage*
   - *SpatiaLite*
   - *PostgreSQL*
   - *SAP HANA*
   - *MS SQL Server*
   - *Oracle*
   - *WMS/WMTS*
   - *Vector Tiles*
   - *XYZ Tiles*
   - *WCS*
   - *WFS/OGC API-Features*
   - *ArcGIS REST Server*

### 탐색기 항목과 대화형 작업

탐색기는 드래그 & 드롭을 지원합니다. 탐색기 내부에서, 탐색기에서 캔버스 및 *Layers* 패널로, 그리고 *Layers* 패널에서 탐색기에 있는 (GeoPackage 같은) 레이어 컨테이너로 드래그 & 드롭할 수 있습니다.

탐색기에 있는 프로젝트 파일 항목을 해당 프로젝트가 담고 있는 (그룹 포함) 전체 레이어 트리를 표시하도록 펼칠 수 있습니다. 프로젝트 항목을 브라우저의 다른 모든 항목과 마찬가지로 취급하기 때문에, 탐색기 안에서 드래그 & 드롭하거나 (예를 들어 레이어 항목을 GeoPackage 파일로 복사하거나) 현재 프로젝트에 드래그 & 드롭 또는 더블 클릭을 통해 추가할 수도 있습니다.

*Browser* 패널에 있는 요소를 오른쪽 클릭하면 컨텍스트 메뉴가 열립니다.

파일 시스템 디렉터리 항목의 경우, 컨텍스트 메뉴에서 다음 옵션을 선택할 수 있습니다:

- *New* 메뉴 옵션을 선택하면 다음 선택한 항목에 생성합니다:
  - *Directory…*

  - *GeoPackage*…
  - *ShapeFile*…

- *Add as a Favorite*: 언제든 즐겨찾기 폴더를 재명명하거나 (*Rename favorite*…) 제거할 수 (*Remove favorite*) 있습니다.
- *Hide from Browser*: *Settings* → *Options* → *Data Sources* → *Hidden browser paths* 설정에서 숨긴 폴더들을 가시화시키도록 켜고 끌 수 있습니다.
- *Fast Scan this Directory*
- *Open Directory*
- *Open in Terminal*
- *Properties*…
- *Directory Properties*…

프로젝트에서 레이어로 동작할 수 있는 지엽 (leaf; 맨 마지막 끝 노드) 항목의 경우, 컨텍스트 메뉴에 지원하는 옵션을 표시할 것입니다. 예를 들어 데이터베이스나 서비스 기반이 아닌 벡터, 래스터 및 메시 데이터소스의 경우:

- *Export Layer* → *To File*…
- *Add Layer to Project*
- *Layer Properties*
- *Open with Data Source Manager*…
- *Manage* → *Rename* "*<name of file>*"…또는 *Delete* "*<name of file>*"…
- *Show in Files*
- *File Properties*

*Layer properties* 항목의 컨텍스트 메뉴는 (프로젝트에 레이어를 추가한 다음벡터 및래스터 레이어 속성에서 찾을 수 있는 것과 비슷한) 다음 옵션을 표시할 것입니다:

- 레이어 용 *Metadata*: 메타데이터 그룹의 컨텍스트 메뉴 옵션은 *Information from provider* (가능한 경우, *Path* 가 소스를 가리키는 하이퍼링크일 것입니다), *Identification*, *Extent*, *Access*, *Fields* (벡터 레이어 용), *Bands* (래스터 레이어 용), *Contacts*, *Links* (벡터 레이어 용), *References* (래스터 레이어 용), *History* 입니다.
- *Preview* 패널
- (*Attributes* 패널에 있는) 벡터 소스 용 속성 테이블

Use *Open with Data Source Manager*… to directly open and configure the data source in the *Data Source Manager* using the URI of your data source. This simplifies the process of adding a layer from the *Browser* by allowing you to set specific opening options for the data source. It is currently available for vector (including the dedicated GeoPackage entry), raster, and SpatiaLite data sources.

*Browser* 를 사용해서 프로젝트에 레이어를 추가하려면:

1. 앞에서 설명한대로 *Browser* 를 활성화하십시오. 사용자 파일 시스템, 데이터베이스 및 웹 서비스의 탐색 트리가 표시될 것입니다. 데이터베이스나 웹 서비스를 표시하려면 먼저 연결해야 할 수도 있습니다. (데이터베이스 또는 웹 서비스 절을 참조하세요.)
2. 목록에서 레이어를 찾으십시오.
3. 컨텍스트 메뉴를 사용해서 레이어 이름을 더블 클릭하거나맵 캔버스 로 레이어를 드래그 & 드롭하십시오. 레이어 패널 에 사용자 레이어를 추가하고 맵 캔버스 상에 표시할 것입니다.

> **팁**
>
> **탐색기에서 QGIS 프로젝트를 직접 열기**
>
> 프로젝트 이름을 더블 클릭하거나 맵 캔버스로 드래그 & 드롭하면 탐색기 패널에서 QGIS 프로젝트도 직접 열 수 있습니다.

파일을 불러왔다면, 맵 탐색 도구를 사용해서 레이어 주변을 확대/축소 및 이동할 수 있습니다. 레이어 스타일을 변경하려면 레이어 이름을 더블 클릭하거나, 범례에서 레이어 이름을 오른쪽 클릭하고 컨텍스트 메뉴에서 *Properties* 를 선택해서 *Layer Properties* 대화창을 여십시오. 벡터 레이어의 심볼을 설정하는 데 대한 자세한 내용을 알고 싶다면심볼 속성 절을 참조하세요.

탐색 트리에 있는 항목을 오른쪽 클릭하면 다음 작업을 할 수 있습니다:

- 파일 또는 테이블인 경우, 해당 항목의 메타데이터를 표시하거나 사용자의 프로젝트에 항목을 열 수 있습니다. 테이블의 경우 재명명하거나, 삭제하거나, 절단할 수도 (truncate) 있습니다.
- 폴더인 경우, 사용자의 즐겨찾기에 북마크해두고 탐색 트리에서 숨길 수 있습니다. *Settings* ▸ *Options* ▸ *Data Sources* 탭에서 이렇게 숨긴 폴더들을 관리할 수 있습니다.
- 사용자의공간 북마크 관리: 북마크를 생성하거나, `XML` 파일로 내보내거나 가져올 수 있습니다.
- 데이터베이스 또는 웹 서비스로의 연결을 생성할 수 있습니다.
- 스키마를 새로고침하거나, 재명명하거나 삭제할 수 있습니다.

단순히 드래그 & 드롭하는 것만으로도 데이터베이스로 파일을 불러들이거나 어떤 스키마/데이터베이스에서 다른 스키마/데이터베이스로 테이블을 복사할 수 있습니다. 드래그하는 동안 오래 스크롤을 해야 하는 일을 피하기 위해 두 번째 탐색 패널을 사용할 수 있습니다. 그냥 파일을 선택한 다음 한쪽 패널에서 다른 패널로 드래그 & 드롭하십시오.

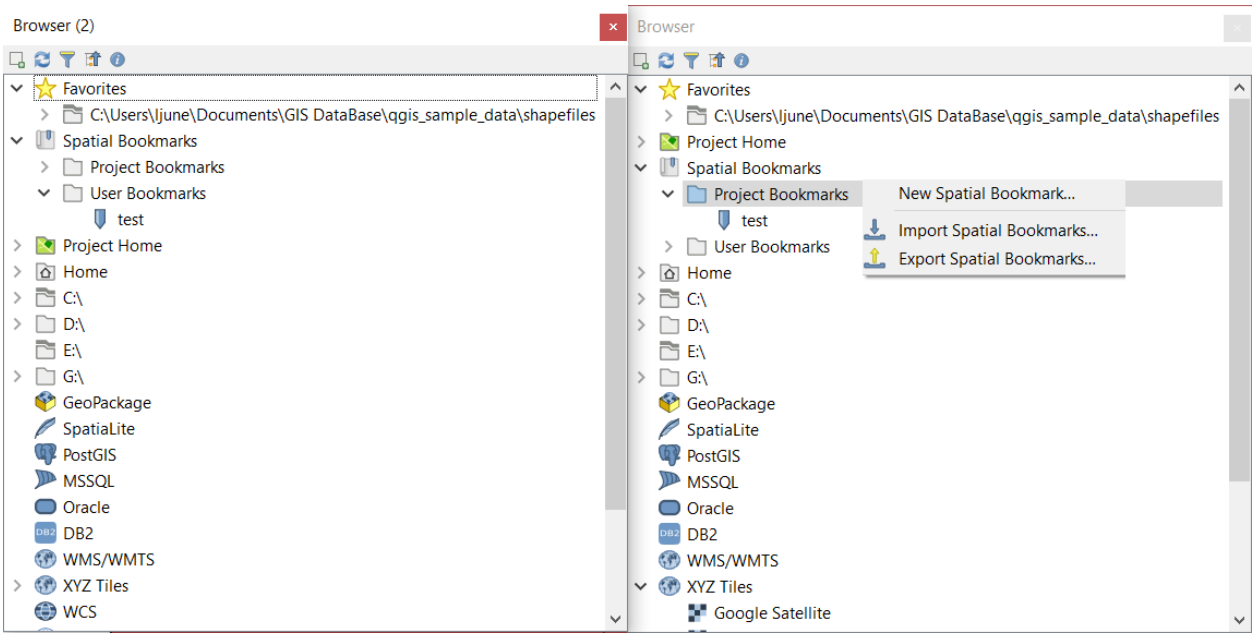



그림 11.2: 나란히 있는 QGIS 탐색기 패널

> **팁**

**사용자 OS 의 파일 탐색기에서 단순히 드래그 & 드롭하는 것만으로 QGIS 에 레이어 추가**

사용자의 운영체제 파일 탐색기에서 *Layers Panel* 또는 맵 캔버스로 파일 (들) 을 드래그 & 드롭해서 프로젝트에 추가할 수도 있습니다.

### 11.1.2 데이터베이스 관리자

The *DB Manager* Plugin is another tool for integrating and managing spatial database formats supported by QGIS (PostgreSQL, SpatiaLite, GeoPackage, Oracle Spatial, MS SQL Server, Virtual layers). It can be activated from the *Plugins* → *Manage and Install Plugins*… menu.

DB Manager 플러그인은 다음 여러 기능들을 제공합니다:

- 데이터베이스에 연결해서 그 구조 및 내용을 표시합니다.
- 데이터베이스의 테이블을 미리보기합니다.
- 맵 캔버스에 레이어를 더블 클릭 또는 드래그 & 드롭으로 추가합니다.
- QGIS 탐색기에서 또는 다른 데이터베이스로부터 데이터베이스에 레이어를 추가합니다.
- SQL 쿼리를 생성하고 맵 캔버스에 그 산출물을 추가합니다.
- 가상 레이어 를 생성합니다.

데이터베이스 관리자 플러그인 에서 데이터베이스 관리자 성능에 관한 더 자세한 정보를 설명하고 있습니다.

### 11.1.3 제공자 기반 불러오기 도구

QGIS 가 레이어를 추가하기 위해 제공하는 주 도구인 탐색기 패널 및 데이터베이스 관리자 이외에도, 데이터 제공자에 특화된 도구들이 존재합니다.

**참고**

일부외부 플러그인 도 QGIS 에서 특정 파일 포맷을 열기 위한 도구를 제공합니다.

#### 파일에서 레이어 불러오기

파일에서 레이어를 불러오려면:

1. *Data Source Manager* 대화창에 있는 레이어 유형 탭을 선택하십시오. 예를들면 Open Data Source Manager 버튼을 클릭하고 (또는 `Ctrl+L` 조합키를 누르고) 레이어 유형 탭을 선택하거나:
   - 벡터 데이터 (GML, ESRI 셰이프파일, Mapinfo 및 DXF 레이어 등) 의 경우: `Ctrl+Shift+V` 조합키를 누르거나, *Layer* → *Add Layer* → *Add Vector Layer* 메뉴 옵션을 선택하거나, Add Vector Layer 툴바 버튼을 클릭하십시오.
   - 래스터 데이터 (GeoTiff, MBTiles, GRIB(GRIdded Binary) 및 DWG 레이어 등) 의 경우: `Ctrl+Shift+R` 조합키를 누르거나, *Layer* → *Add Layer* → *Add Raster Layer* 메뉴 옵션을 선택하거나, Add Raster Layer 툴바 버튼을 클릭하십시오.

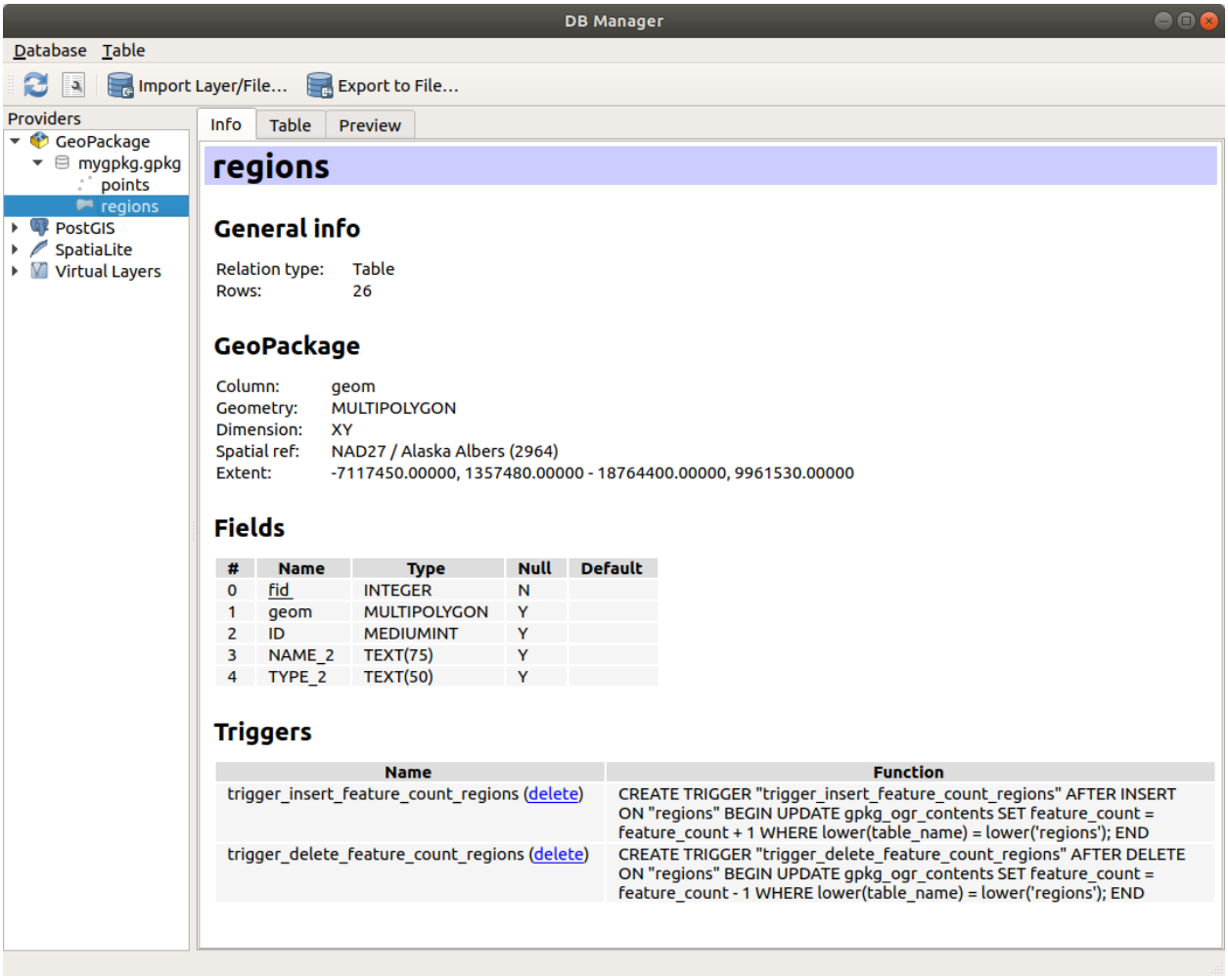



그림 11.3: 데이터베이스 관리자 대화창

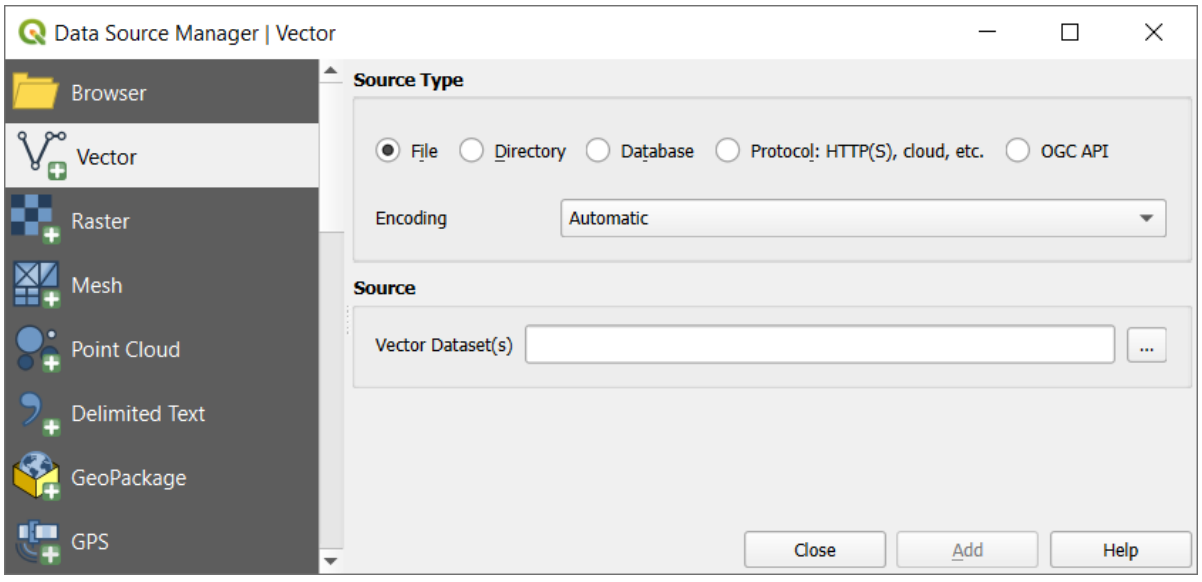



그림 11.4: 벡터 레이어 추가 대화창

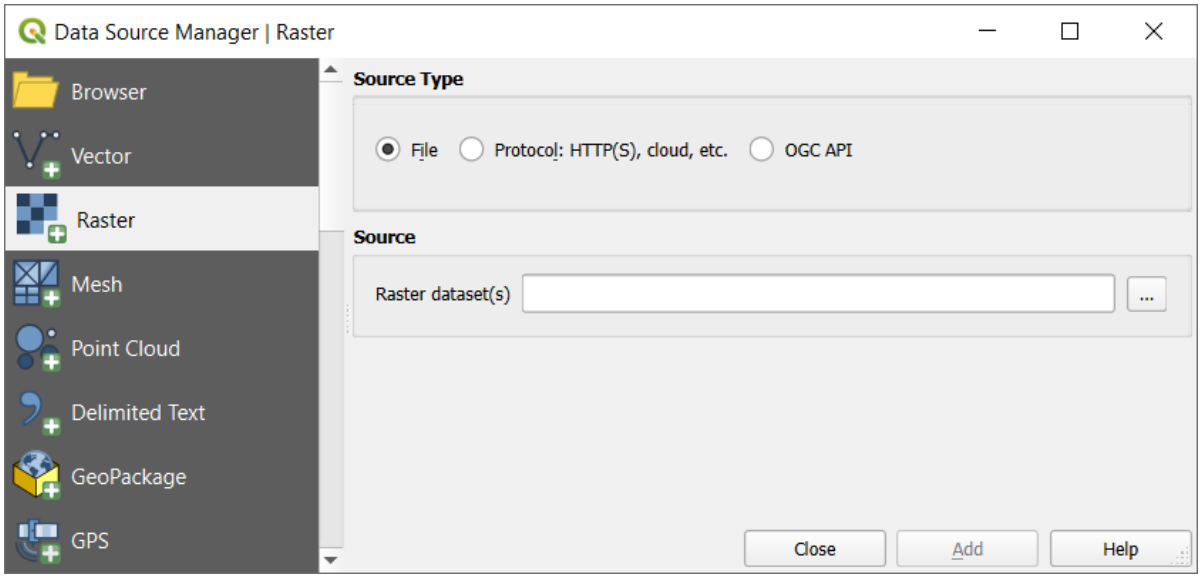



그림 11.5: 래스터 레이어 추가 대화창

2. *File* 소스 유형을 확인하십시오.
3. … Browse 버튼을 클릭하십시오.
4. 파일 시스템을 탐색해서 지원 데이터소스를 불러오십시오. 대화창에서 `Ctrl` 키를 누른 채 여러 항목을 클릭하거나, `Shift` 키를 누른 채 선택 범위의 첫 번째 항목과 마지막 항목을 클릭하면 한 번에 하나 이상의 레이어를 불러올 수 있습니다. 잘 검증된 포맷들만 포맷 필터에 표시됩니다. (풀다운 메뉴에서 맨 위에 있는) `All files` 를 선택하면 다른 포맷들도 불러올 수 있습니다.
5. Press *Open* to load the selected file into *Data Source Manager* dialog.

   Depending on the selected layer type, additional *Options* (encoding, geometry type, table filtering, file locking, data formatting …) are available for configuring. These options are described in detail in the specific GDAL vector or raster driver documentation. At the top of the options, a text with hyperlink will directly lead to the documentation of the appopriate driver for the selected file format.
6. *Add* 버튼을 클릭해서 파일을 QGIS 로 불러와 맵 뷰에 보이게 하십시오. 레이어 여러 개를 담고 있는 벡터 데이터셋을 추가하는 경우 *Select Items to Add* 대화창이 열릴 것입니다. 이 대화창에서, 사용자가 데이터셋에서 추가하고 싶은 특정 레이어를 선택할 수 있습니다. 또한 *Options* 부분에서 다음을 선택할 수도 있습니다:

   - *Add layers to a group*: 그룹에 레이어를 추가
   - *Show system and internal tables*: 시스템 테이블 및 내부 테이블 보이기
   - *Show empty vector layers*: 비어 있는 벡터 레이어 보이기

   그림 11.7 은 `alaska.shp` 파일을 불러온 후의 QGIS 입니다.

**참고**

MapInfo(예를 들면 `.tab`) 또는 Autocad(`.dxf`) 같은 일부 포맷은 파일 하나 안에 서로 다른 도형 유형을 함께 담을 수 있기 때문에 QGIS 에 이런 포맷을 불러올 경우 레이어 하나 당 도형 유형 하나만 보유할 수 있도록, 사용할 도형을 선택할 수 있는 대화창이 열립니다.

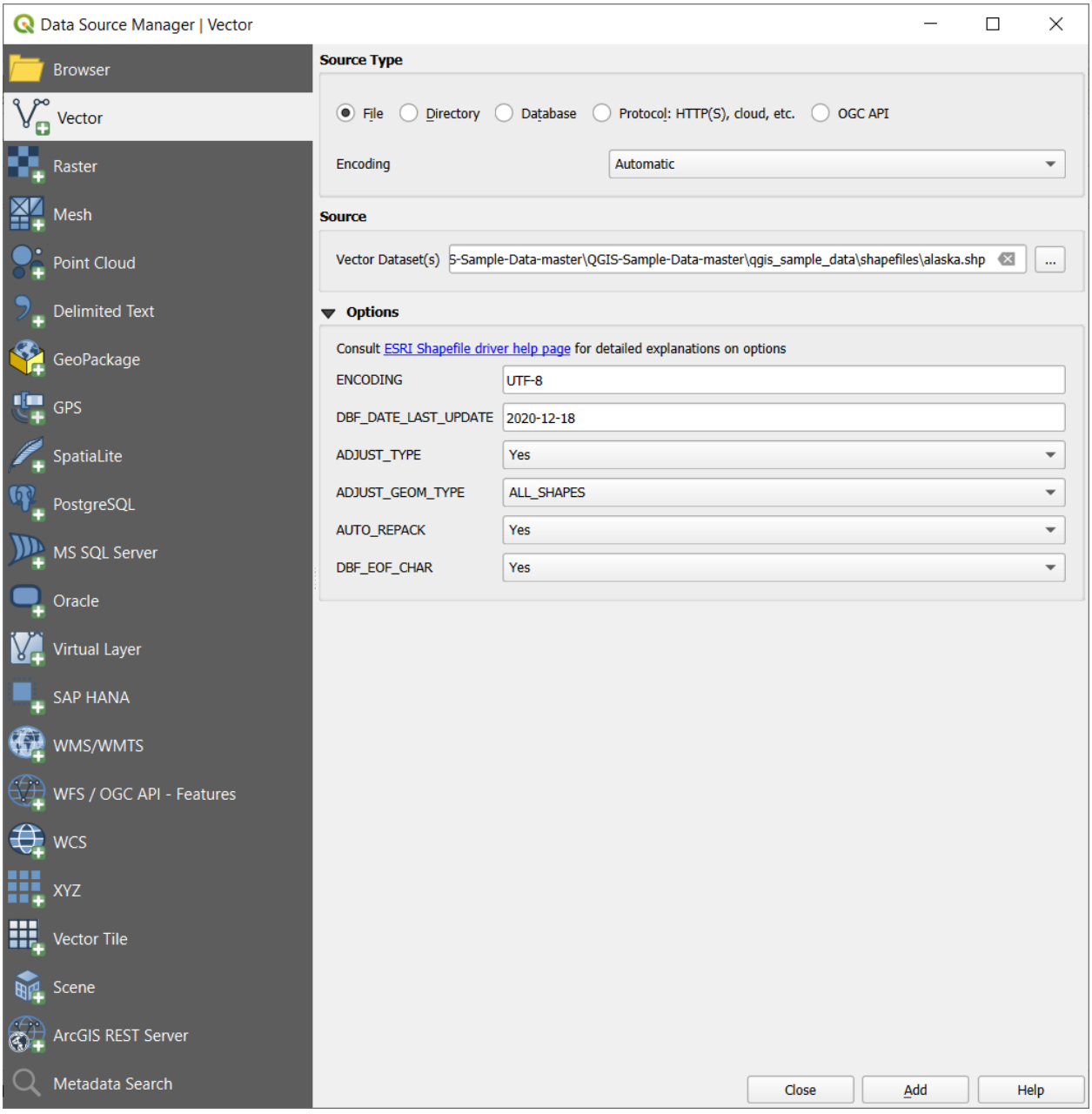



그림 11.6: 열기 옵션으로 셰이프파일 불러오기

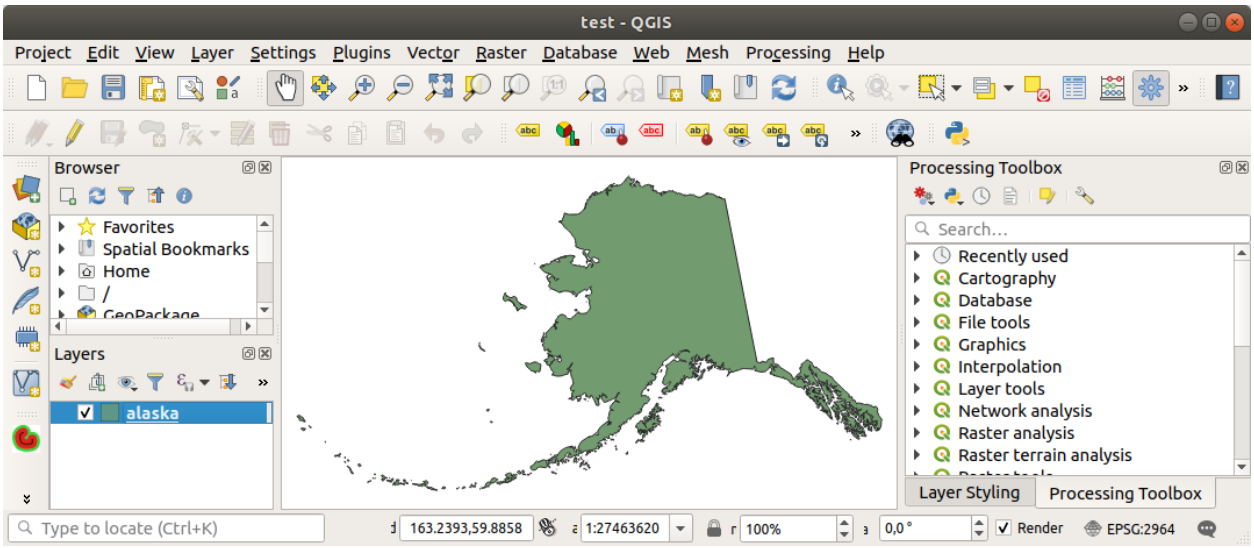



그림 11.7: 알래스카 셰이프파일을 불러들인 QGIS

Add Vector Layer 및 Add Raster Layer 탭에서는 *File* 이 아닌 소스 유형으로부터도 레이어를 불러올 수 있습니다:

- `ArcInfo Binary Coverage`, `UK. National Transfer Format` 과 같은 특정 벡터 포맷 뿐만 아니라 `US Census Bureau` 또는 `OpenfileGDB` 의 원시 (raw) TIGER 포맷도 불러올 수 있습니다. 이를 위해서는 *Source type* 으로 *Directory* 를 선택해야합니다. 이때 … Browse 를 누르면 대화창에서 디렉토리를 선택할 수 있습니다.
- *Database* 소스 유형로는 기존 데이터베이스 연결을 선택하거나 또는 선택한 데이터베이스 유형의 연결을 생성할 수 있습니다. `ODBC`, `Esri Personal Geodatabase`, `MS SQL` 서버 는 물론 `PostgreSQL` 또는 `MySQL` 등 많은 데이터베이스 유형을 선택할 수 있습니다.

  Pressing the *New* button opens the *Create a New OGR Database Connection* dialog whose parameters are among the ones you can find in 저장된 연결 생성하기. Pressing *Open* lets you select from the available tables.

- *Protocol: HTTP(S), cloud, etc.* 소스 유형은 로컬에 저장된 또는 네트워크 상에 있는 데이터를 엽니다. 이때 네트워크를 공개적으로 접속할 수도 있고, 상용 클라우드 저장소 서비스의 비공개 버킷 (private bucket) 을 사용할 수도 있습니다. 다음과 같은 프로토콜 유형을 지원합니다:
  - `HTTP/HTTPS/FTP`: *URI* 와 함께 필요한 경우인증 정보 를 입력해야 합니다.
  - Cloud storage such as `AWS S3`, `Google Cloud Storage`, `Microsoft Azure Blob`, `Microsoft Azure Data Lake Storage`, `Alibaba OSS Cloud`, and `Open Stack Swift Storage` supports direct control over VSI *Credential Options* when adding OGR vector or GDAL raster layers. You need to fill in the *Bucket or container* and the *Object key* first. After that, you can add the necessary *Credential Options*.

    When adding OGR vector or GDAL raster layers from the cloud based protocols, you can also set additional *Credential options* for that specific driver and bucket. When credential options are found in a layer's URI, they will also be automatically set. This allows different layers to use different credentials.
  - `GeoJSON` 또는 `GEOJSON - Newline Delimited` 포맷을 사용하거나 `CouchDB` 데이터베이스 기반의 (아직 실험 단계인) OGC `WFS 3` 를 지원하는 서비스: *URI* 와 함께 필요한 경우인증 정보 를 입력해야 합니다.
  - 모든 벡터 소스 유형에 대해 *Encoding* 을 정의하거나 또는 *Automatic* 설정을 이용할 수 있습니다.

- The *OGC API* source type allows you to access vector and raster data from servers that implement the OGC API standards. To use this option:

  1. Select *OGC API* from the *Data Source Manager* dialog.
  2. Enter the endpoint of the OGC API service you want to connect to. Note that you don't need to prefix the endpoint with "OGCAPI:".
  3. Click *Connect* to establish a connection to the server.

### 메시 레이어 불러오기

메시 (mesh) 란 일반적으로 시계열 및 기타 요소들을 가지고 있는 비구조적 그리드를 말합니다. 공간 요소는 2 차원 또는 3 차원 공간에 있는 꼭짓점, 변 (edge) 및 면 (face) 의 집합을 담고 있습니다. 메시 레이어에 대한 더 자세한 내용은메시 데이터 작업 을 참조하세요.

QGIS 에 메시 레이어를 추가하려면:

1. *Layer* 메뉴에서 선택하거나 Open Data Source Manager 버튼을 클릭해서 *Data Source Manager* 대화창을 여십시오.
2. 왼쪽 패널에 있는 *Mesh* 탭을 선택하십시오.
3. … Browse 버튼을 눌러 파일을 선택하십시오. 여러 유형 을 지원합니다.
4. 파일을 선택한 다음 *Add* 버튼을 누르십시오. 자체 메시 렌더링을 이용해서 레이어를 추가할 것입니다.
5. 선택한 파일이 메시 레이어를 많이 담고 있다면, 불러올 하위 레이어를 선택할 수 있는 대화창이 뜰 것입니다. 선택을 마친 다음 *OK* 버튼을 누르면 자체 메시 렌더링을 통해 레이어를 불러옵니다. 그룹 안에 선택한 레이어를 불러올 수도 있습니다.

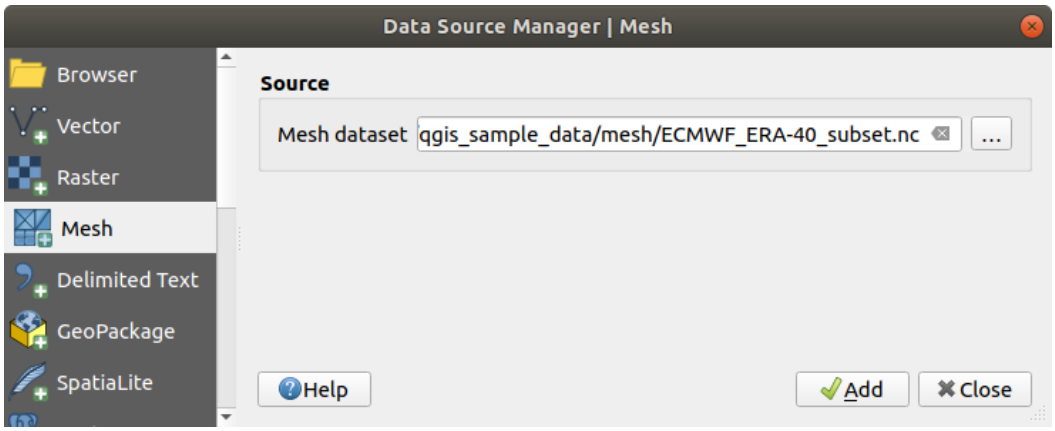



그림 11.8: 데이터소스 관리자의 메시 탭

### 구분 텍스트 파일 가져오기

앞에서 설명한 도구들을 이용해서 QGIS 에 구분 텍스트 파일 (예: `.txt`, `.csv`, `.dat`, `.wkt`) 을 불러올 수 있습니다. 이렇게 불러오면, 단순 테이블처럼 표시될 겁니다. 그런데 구분 텍스트 파일이 사용자가 가시화하려는 좌표/도형 데이터를 담고 있는 경우도 있습니다. 이것이 *Add Delimited Text Layer* 를 개발한 이유입니다.

1. Open Data Source Manager 아이콘을 클릭해서 *Data Source Manager* 대화창을 여십시오.
2. *Delimited Text* 탭을 선택하십시오.
3. … Browse 버튼을 클릭해서 가져올 구분 텍스트 파일을 (예를 들어 `qgis_sample_data/csv/elevp.csv` 파일을) 선택하십시오.
4. 다음에 설명하는 내용처럼 사용자 데이터셋 및 필요를 만족시키는 환경을 설정하십시오.

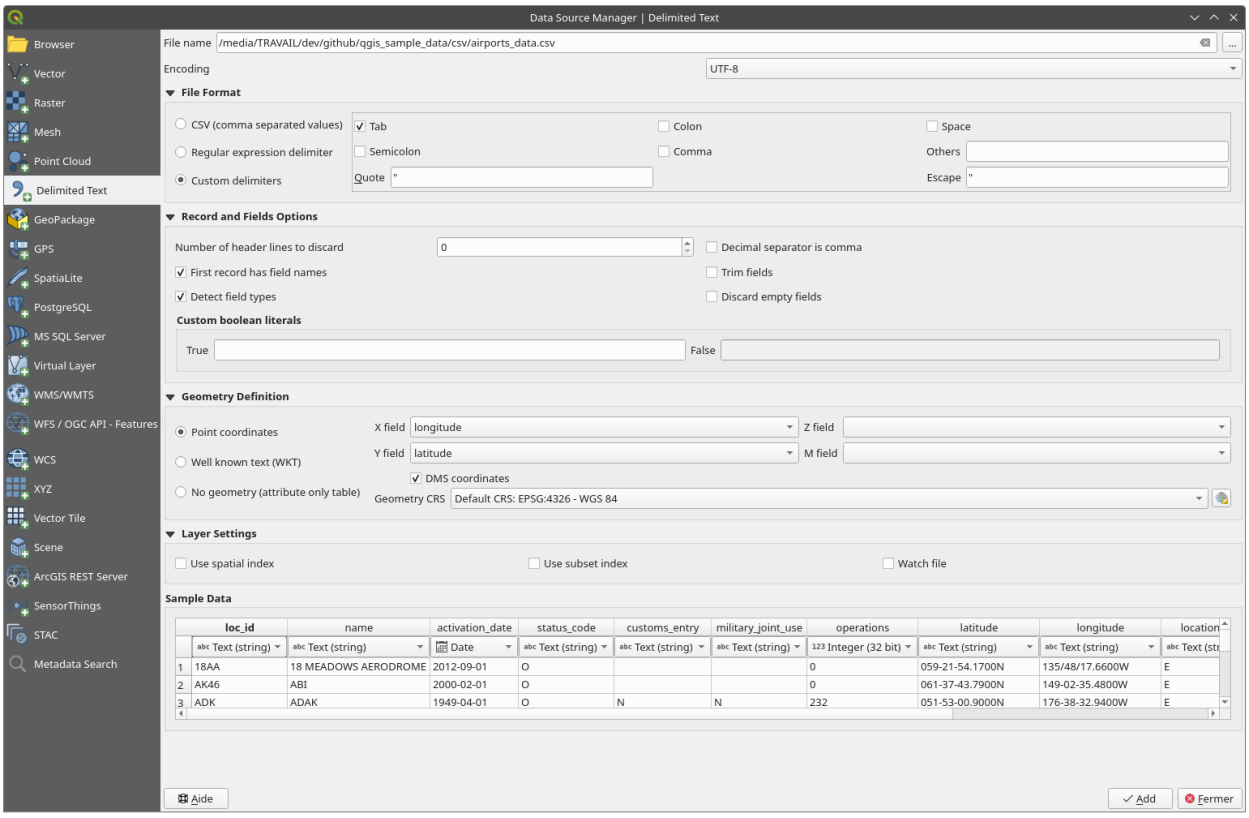



그림 11.9: 구분 텍스트 대화창

### 파일 포맷

파일을 선택하면, QGIS 가 가장 최근에 사용한 구분자로 필드와 행을 식별해서 파일을 파싱하려 할 겁니다. QGIS 가 파일을 제대로 파싱하게 하려면, 정확한 구분자를 선택하는 것이 중요합니다. 다음 옵션 가운데 하나를 선택해서 구분자를 지정할 수 있습니다:

- *CSV (comma separated values)*: 쉼표 문자를 구분자로 사용합니다.
- *Regular expression delimiter*: *Expression* 란에 구분자로 이용할 텍스트를 입력합니다. 예를 들어 구분자를 탭으로 변경하려면, (정규 표현식에서 탭 문자를 나타내는) `\t` 를 입력하세요.
- *Custom delimiters*: `comma`, `space`, `tab`, `semicolon` 등등 사전 정의된 여러 구분자들 가운데 선택합니다.

### 레코드와 필드

데이터 식별을 위한 몇몇 다른 편리한 옵션을 사용할 수 있습니다:

- *Number of header lines to discard*: 가져온 파일의 첫 줄이 빈 줄이거나 다른 서식으로 돼 있기 때문에 첫 줄을 파싱하지 않고자 하는 경우 편리합니다.
- *First record has field names*: 첫 줄에 있는 값들을 필드명으로 사용합니다. 이 옵션을 선택하지 않는다면 QGIS 는 필드명으로 `field_1`, `field_2`, …를 사용할 것입니다.
- *Detect field types*: 필드 유형을 자동으로 식별합니다. 이 옵션을 선택하지 않는다면 모든 속성을 텍스트 필드로 취급합니다.
- *Decimal separator is comma*: 소수 구분 기호를 쉼표로 강제할 수 있습니다.
- *Trim fields*: 필드에서 처음 및 마지막 부분의 공백을 잘라낼 수 있습니다.
- *Discard empty fields*: 비어 있는 필드를 무시할 수도 있습니다.
- *Custom boolean literals*: 불 (boolean) 값으로 탐지될 사용자 지정 문자열 쌍을 추가할 수 있습니다.

### 필드 유형 탐지

QGIS tries to detect the field types automatically (unless *Detect field types* is not checked) by examining the content of an optional sidecar CSVT file (see GeoCSV specification) and by scanning the whole file to make sure that all values can actually be converted without errors, the fall-back field type is text.

탐지된 필드 유형은 단순 데이터 미리보기 테이블의 파일명 아래 나타나며 필요한 경우 직접 변경할 수 있습니다.

다음과 같은 필드 유형들을 지원합니다:

- `Boolean`: 불 값으로 해석되는, 대소문자를 구분하지 않는 문자열 쌍은 `1`/`0`, `true`/`false`, `t`/`f`, `yes`/`no` 입니다.
- `Whole Number (integer)`: 정수형
- `Whole Number (integer - 64 bit)`: 64 비트 정수형
- `Decimal Number`: 십진수. 배정밀도 부동소수점형 숫자입니다.
- `Date`: 날짜
- `Time`: 시간

- `Date and Time`: 날짜 & 시간
- `Text`: 텍스트

### 도형 정의

파일 파싱 작업이 끝나면 *Geometry definition* 을 다음 가운데 하나로 설정하십시오.

- *Point coordinates*: 레이어가 포인트 도형 유형이며 해당 필드를 가지고 있다면 *X field*, *Y field*, (3 차원 데이터의 경우) *Z field* 를 입력하고, (측정 치수용) *M field* 도 지정하십시오. 좌표가 도/분/초로 정의된 경우, *DMS coordinates* 체크박스를 체크하십시오. Select CRS 위젯을 이용해서 적절한 *Geometry CRS* 를 지정하십시오.
- *Well known text (WKT)*: WKT 로 공간 정보를 표현하는 경우 이 옵션을 선택하십시오. WKT 도형을 담고 있는 *Geometry field* 를 선택하고 적절한 *Geometry field* 를 선택하거나 QGIS 가 자동으로 식별하게 하십시오. Select CRS 위젯을 이용해서 적절한 *Geometry CRS* 를 지정하십시오.
- 파일이 비공간 데이터를 담고 있는 경우, *No geometry (attribute only table)* 옵션을 활성화하면 파일을 서수 (□□) 테이블로 불러들일 것입니다.

### 레이어 설정

여기에 더해, 다음 옵션을 활성화시킬 수 있습니다:

- *Use spatial index*: 피처를 공간적으로 선택하고 화면에 표시하는 성능을 향상시킵니다.
- *Use subset index*: (레이어 속성에 하위 집합을 정의한 경우) 하위 집합 필터 의 성능을 향상시킵니다.
- *Watch file*: QGIS 실행 중에 다른 응용 프로그램이 파일을 변경하지는 않는지 감시합니다.

마지막으로 *Add* 를 클릭하면 맵에 레이어를 추가합니다. 이 예시에서는 프로젝트에 `Elevation` 이라는 이름의 포인트 레이어를 추가하며, 이 레이어는 QGIS 에서 다른 모든 맵 레이어와 마찬가지로 동작합니다. 이 레이어는 `.csv` 소스 파일을 쿼리한 결과물로 (즉 해당 파일과 연결돼 있기 때문에) 디스크 상에 공간 레이어를 만들려면저장 해야만 합니다.

### 샘플 데이터

파싱 작업자 속성을 설정하는 대로, 적용한 설정을 따라 샘플 데이터 미리보기를 업데이트합니다.

또한 샘플 데이터 테이블에서, 자동 결정된 열 유형을 무시할 수 있습니다.

## DXF 또는 DWG 파일 가져오기

탐색기 패널에서 `DXF` 및 `DWG` 파일을 드래그 & 드롭하기만 하면 QGIS 에 추가할 수 있습니다. 프로젝트에 추가하고 싶은 하위 레이어를 선택할 수 있는 창이 뜰 겁니다. 레이어 추가 시 스타일 속성은 랜덤하게 적용됩니다.

> **참고**
>
> 포인트, 라인 그리고/또는 폴리곤 등 여러 도형 유형을 담고 있는 DXF 파일의 경우, <*filename.dxf*> *entities* <*geometry type*> 형식으로 레이어를 명명합니다.

QGIS 에서 DXF/DWG 파일 구조 및 그 심볼을 유지하려면, 전용 *Project → Import/Export → Import Layers from DWG/DXF*…도구를 사용해야 할 수도 있습니다. 이 도구를 사용하면:

1. 그리기 파일 (drawing file) 에서 GeoPackage 데이터베이스로 요소들을 가져올 수 있습니다.
2. 프로젝트에 가져온 요소들을 추가할 수 있습니다.

*DWG/DXF Import* 대화창에서 그리기 파일의 내용을 가져오려면:

1. 가져올 DWG/DXF 그리기 파일 같은, *Source drawing* 의 위치를 입력하십시오.
2. 그리기 파일에 있는 데이터의 좌표계를 지정하십시오.
3. 데이터를 저장할 GeoPackage 파일 같은, *Target package* 의 위치를 입력하십시오. 기존 파일을 입력하면 해당 파일을 덮어 쓸 것입니다.
4. `blocks` 를 가져올 방법을 전용 콤보박스로 선택하십시오:
   - *Expand Block Geometries*: 그리기 파일에 있는 블록 (block) 을 정규 (normal) 요소로 가져옵니다.
   - *Expand Block Geometries and Add Insert Points*: 그리기 파일에 있는 블록을 정규 요소로 가져온 다음 삽입 포인트를 포인트 레이어로 추가합니다.
   - *Add Only Insert Points*: 블록 삽입 포인트를 포인트 레이어로 추가합니다.
5. 가져온 레이어를 `curved` 만곡 도형 유형으로 승격시키려면 ☑ *Use curves* 를 체크하십시오.
6. *Import* 버튼을 클릭해서 그리기 파일을 대상 GeoPackage 파일로 가져오십시오. 그리기 파일 내용이 GeoPackage 데이터베이스를 자동으로 채울 것입니다. 파일 용량에 따라 시간이 걸릴 수도 있습니다.

`.dwg` 또는 `.dxf` 데이터를 GeoPackage 데이터베이스로 가져왔다면, 가져온 파일에 있는 레이어의 목록이 대화창 프레임의 하단을 채울 것입니다. 이 목록에서 QGIS 프로젝트에 추가할 레이어를 선택할 수 있습니다:

1. 하단 프레임의 맨 위에서, 프로젝트에서 그리기 파일을 그룹화시킬 *Group name* 을 설정하십시오. 기본적으로 소스 그리기 파일의 파일 이름으로 설정됩니다.
2. 가시화할 레이어를 체크하십시오: 그리기 레이어의 포인트, 라인, 라벨 및 영역 피처를 위한 벡터 레이어들을 담고 있는 즉석 그룹에 선택한 각 레이어를 추가합니다. 선택한 레이어의 스타일은 *CAD 에서 원래 가지고 있던 모양을 닮게 될 것입니다.
3. 레이어를 열 때 가시화할 것인지 여부를 선택하십시오.
4. 모든 레이어를 단일 그룹으로 배치하려면 ☑ *Merge layers* 를 체크하십시오.
5. *OK* 를 클릭하면 QGIS 에서 레이어가 열립니다.

### 오픈스트리트맵 벡터 가져오기

오픈스트리트맵 프로젝트가 인기를 끌고 있습니다. 많은 나라에서 디지털 도로 지도 같은 무료 지리 데이터가 없기 때문입니다. GPS 데이터, 항공사진 또는 현지인의 지식으로 편집 가능하고 무료인 세계 지도를 생성하는 것이 OSM(OpenStreetMap) 프로젝트의 목표입니다. 이 목표를 돕기 위해, QGIS 는 OSM 데이터를 지원하고 있습니다.

*Browser Panel* 을 통해 맵 캔버스에 `.osm` 파일을 불러올 수 있는데, 이때 도형 유형을 바탕으로 하위 레이어를 선택할 수 있는 대화창이 열립니다. 불러온 레이어는 `.osm` 파일에 있는 모든 도형 유형의 데이터를 담고 있으며 `.osm` 파일 데이터 구조를 유지할 것입니다.

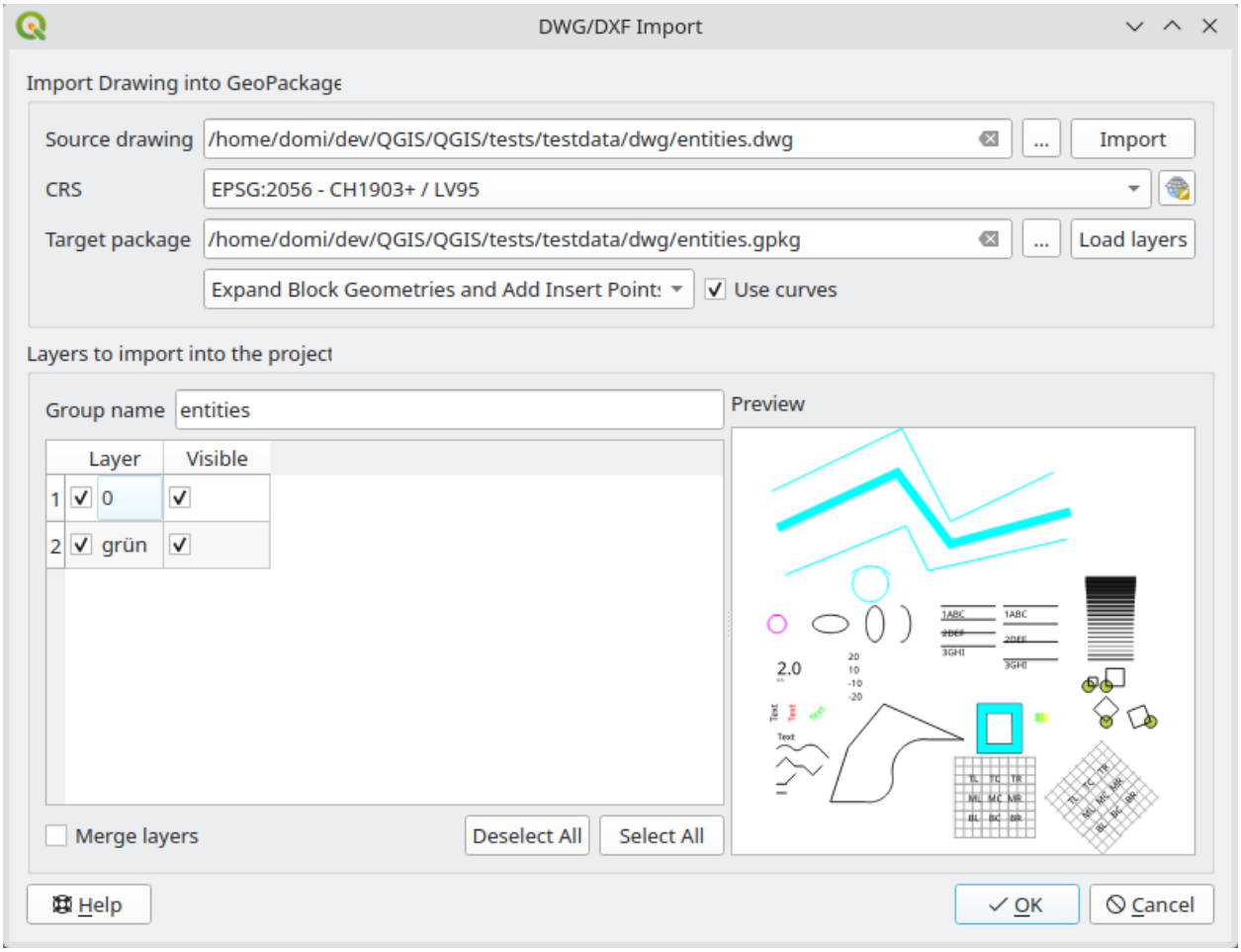



그림 11.10: DWG/DXF 파일 가져오기 대화창

### SpatiaLite 레이어

처음으로 SpatiaLite 데이터베이스에서 데이터를 불러오는 경우, 다음 가운데 한 방법으로 시작하십시오:

- Add SpatiaLite Layer 툴바 버튼을 클릭합니다.
- *Layer* → *Add Layer* 메뉴에서 *Add SpatiaLite Layer*…옵션을 선택합니다.
- `Ctrl+Shift+L` 조합키를 누릅니다.

드롭다운 메뉴에서 QGIS 가 이미 알고 있는 SpatiaLite 데이터베이스를 선택해서 연결할 수 있거나 또는 새 데이터베이스로의 새 연결을 정의할 수 있는 대화창이 열립니다. 새 연결을 정의하려면, *New* 를 클릭한 다음 파일 탐색기에서 사용자 SpatiaLite 데이터베이스를 지정하십시오. SpatiaLite 데이터베이스는 `.sqlite` 확장자를 가진 파일입니다.

QGIS 는 SpatiaLite 에서 편집할 수 있는 뷰도 지원합니다.

### GPS

GPS 데이터를 저장하기 위한 수십 개의 서로 다른 파일 포맷이 있습니다. QGIS 가 이용하는 포맷은 GPX(GPS eXchange format) 라고 불리는데, 동일 파일 내에 웨이포인트, 루트 및 트랙을 얼마든지 담을 수 있는 표준 교환(interchange) 포맷입니다.

… Browse 버튼으로 GPX 파일을 선택한 다음, 체크박스를 통해 해당 GPX 파일에서 불러오고자 하는 피처 유형을 선택하십시오. 각 피처 유형을 개별 레이어로 불러올 것입니다.

*GPS 데이터 작업* 에서 GPS 데이터 조작에 대해 더 자세히 설명하고 있습니다.

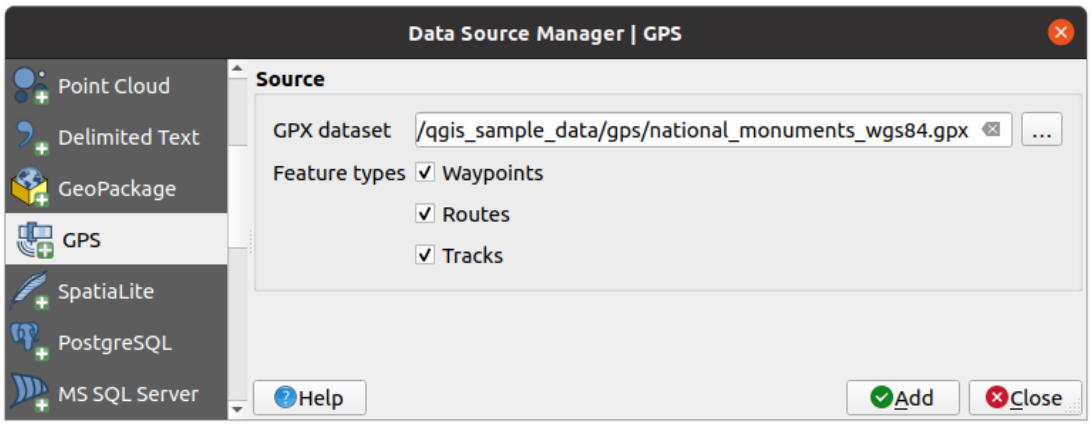



그림 11.11: GPS 데이터 불러오기 대화창

### GRASS

*GRASS GIS 통합* 에서 GRASS 벡터 데이터 작업을 설명하고 있습니다.

#### 데이터베이스 관련 도구

#### 저장된 연결 생성하기

QGIS 가 지원하는 데이터베이스 유형에서 테이블을 읽고 쓰기 위해, 사용자는 해당 데이터베이스로의 연결을 생성해야 합니다. *QGIS 탐색기 패널* 이 데이터베이스를 연결하고 사용하는 데 가장 간단하고 추천받는 방법이긴 하지만, QGIS 는 각 데이터베이스 유형에 연결하고 테이블을 불러올 수 있는 다른 도구들도 제공하고 있습니다:

- *Add PostgreSQL Layer*… or by typing `Ctrl+Shift+D`
- *Add Oracle Spatial Layer*…또는 `Ctrl+Shift+O` 단축키 입력
- *Add MS SQL Server Layer*
- *Add SAP HANA Spatial Layer*…또는 `Ctrl+Shift+G` 단축키 입력

*Manage Layers Toolbar* 또는 *Layer* ▸ *Add Layer* ▸ 메뉴를 통해 이 도구들에 접근할 수 있습니다. SpatiaLite 데이터베이스 연결 방법은*SpatiaLite 레이어* 에서 설명하고 있습니다.

> **팁**
>
> **QGIS 탐색기 패널에서 데이터베이스 연결 생성**
>
> 탐색 트리에서 대응하는 데이터베이스 유형을 선택해서 오른쪽 클릭한 다음, *Connect* 를 선택하면 데이터베이스 연결 대화창이 열릴 겁니다.

Most of the connection dialogs follow a common structure:

- a section with credentials information to connect to the database
- a section with options to tune which data can be requested in the database

### Connecting to PostgreSQL

The first time you use a PostgreSQL data source, you must create a connection to a database that contains the data. Press the appropriate button as exposed above, opening the *PostgreSQL* tab of the *Data Source Manager* dialog. To access the connection manager, click on the *New* button to display the *Create a New PostgreSQL Connection* dialog.

- *Name*: 해당 연결을 위한 이름입니다. *Database* 파라미터와 동일해도 됩니다.
- *Service*: Service parameter to be used alternatively to hostname/port (and potentially database). This can be defined in `pg_service.conf`. Check the *PostgreSQL Service connection file* section for more details.
- *Host*: 데이터베이스 호스트의 이름입니다. TCP/IP 연결을 열거나 호스트를 핑 (ping) 하는 데 쓰일 수도 있기 때문에 분해할 수 있는 (resolvable) 호스트명이어야만 합니다. 데이터베이스가 QGIS 와 동일한 컴퓨터에 있는 경우, 그냥 *localhost* 라고 입력하십시오.
- *Port*: Port number the PostgreSQL database server listens on. The default port for PostgreSQL is `5432`.
- *Database*: 데이터베이스의 이름입니다.
- *SSL mode*: SSL encryption setup. The following options are available:

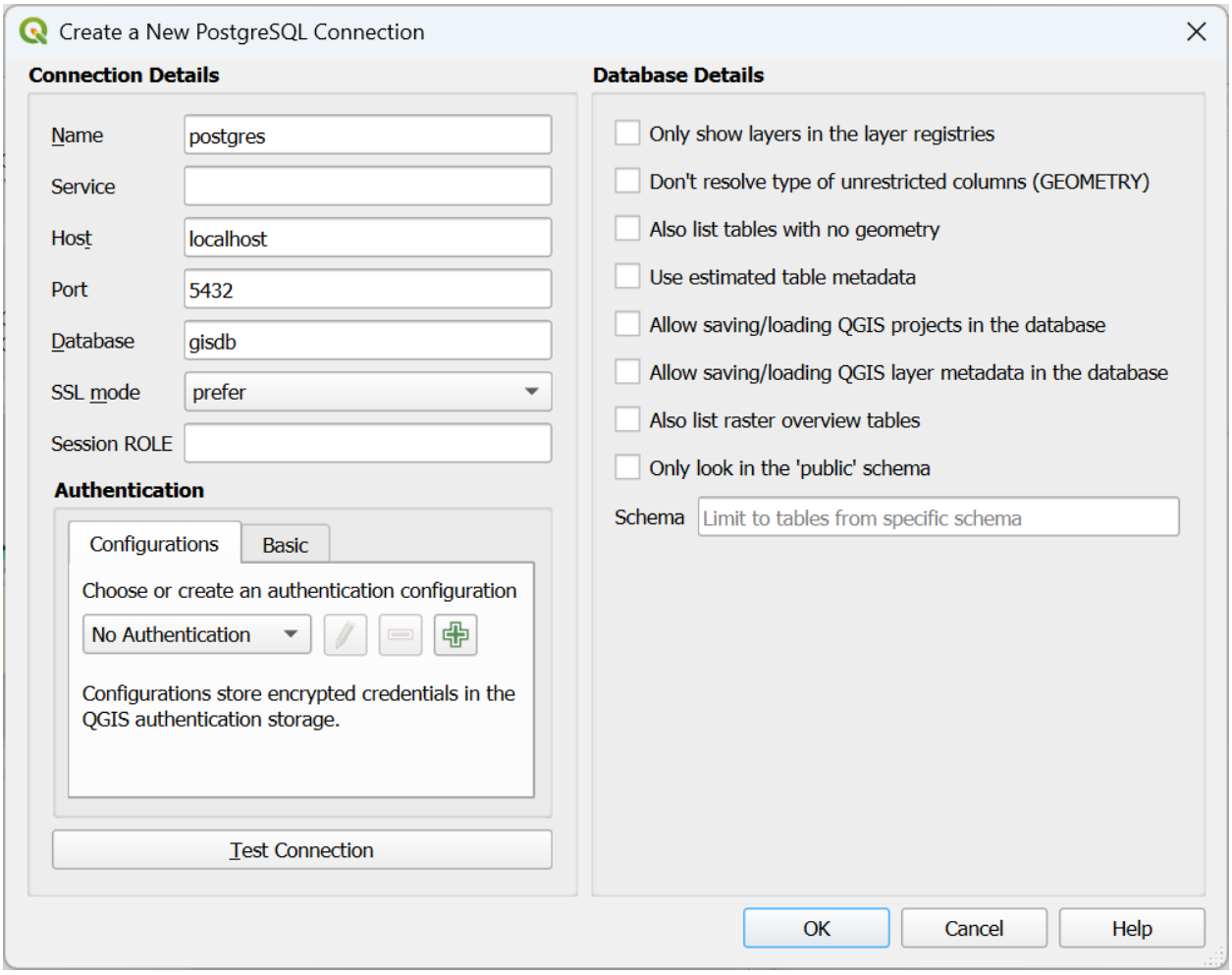



그림 11.12: Create a New PostgreSQL Connection Dialog

  - *Prefer* (기본값): 암호화에 신경을 쓰지는 않지만, 서버가 암호화를 지원한다면 암호화 오버헤드 (overhead) 를 감당할 용의가 있습니다.
  - *Require*: 내 데이터를 암호화하고 싶습니다. 오버헤드도 감당하겠습니다. 네트워크가 내가 원하는 서버에 항상 연결하도록 보장할 것을 믿습니다.
  - *Verify CA*: 내 데이터를 암호화하고 싶습니다. 오버헤드도 감당하겠습니다. 내가 신뢰하는 서버에 연결된다고 확신하고 싶습니다.
  - *Verify Full*: 내 데이터를 암호화하고 싶습니다. 오버헤드도 감당하겠습니다. 내가 지정한 신뢰하는 서버에 연결된다고 확신하고 싶습니다.
  - *Allow*: 보안에 신경을 쓰지는 않지만, 서버가 암호화를 고집한다면 암호화 오버헤드를 감당할 용의가 있습니다.
  - *Disable*: 보안에 신경을 쓰지도 않고, 암호화 오버헤드를 감당할 용의도 없습니다.
- *Session role*: used to set the current user identifier of the current session. This is useful to automatically give the ownership of a new object (table, view, function) to the session_role group and thus share ownership and associated rights with all members of the session_role group. Read more about session role.
- *Authentication*: For general details about the authentication dialog behavior, see 인증.

데이터베이스 유형에 따라 다음 체크박스들을 선택적으로 활성화할 수 있습니다:

- ☑ *Only show layers in the layer registries*: 레이어 레지스트리에만 레이어를 표시합니다.
- ☑ *Don't resolve type of unrestricted columns (GEOMETRY)*: 제약 조건이 없는 도형 열 유형을 분해하지 않습니다.
- ☑ *Also list tables with no geometry*: 도형이 없는 테이블도 기본적으로 목록에 표시할지 말지를 나타냅니다.
- ☑ *Use estimated table metadata*: When initializing layers, various queries may be needed to establish the characteristics of the geometries stored in the database table. When this option is checked, these queries examine only a sample of the rows and use the table statistics, rather than the entire table. This can drastically speed up operations on large datasets, but may result in incorrect characterization of layers (e.g. the feature count of filtered layers will not be accurately determined) and may even cause strange behaviour if columns that are supposed to be unique actually are not.
- ☑ *Allow saving/loading QGIS projects in the database* - 데이터베이스에 QGIS 프로젝트를 저장하고 불러올 수 있습니다. 자세한 내용은여기를 참조하세요.
- ☑ *Allow saving/loading QGIS layer metadata in the database* - 데이터베이스에 QGIS 레이어 메타데이터를 저장하고 불러올 수 있습니다. 자세한 내용은여기를 참조하세요.
- ☑ *Also list raster overview tables*
- ☑ *Only look in the 'public' schema*: '퍼블릭' 스키마만 검색합니다.
- *Schema*: Allows to specify a single schema to limit a connection to. When set, only tables from the matching schema will be included in the browser panel and data source select for the connection. This can be used to limit the database work required to populate tables for a connection pointing to a large database store.

Once all parameters and options are set, you can test the connection by clicking the *Test Connection* button or apply it by clicking the *OK* button.

**PostgreSQL Service connection file**

서비스 연결 파일을 이용하면 PostgreSQL 연결 파라미터들을 단일 서비스명으로 묶을 수 있습니다. 그러면 클라이언트가 해당 서비스명을 지정해서 관련 설정을 사용할 수 있습니다.

서비스 연결 파일명은 GNU/리눅스, 맥 OS 등 * 닉스 시스템에서는 `.pg_service.conf` 이며 윈도우에서는 `pg_service.conf` 입니다.

서비스 파일의 내용은 다음과 비슷합니다:

```
[water_service]
host=192.168.0.45
port=5433
dbname=gisdb
user=paul
password=paulspass

[wastewater_service]
host=dbserver.com
dbname=water
user=waterpass
```

**참고**

앞의 예시에는 서비스가 `water_service` 와 `wastewater_service` 2 개 있습니다. QGIS, pgAdmin 등에서 사용자가 연결하고자 하는 서비스명만 (괄호로 감싸지 않고) 설정해서 연결할 수 있습니다. `psql` 로 서비스를 이용하고자 하는 경우, `psql service=water_service` 명령어를 실행하하면 됩니다.

여기 에서 모든 PostgreSQL 파라미터들을 찾아볼 수 있습니다.

**참고**

서비스 파일에 비밀번호를 저장하고 싶지 않다면 .pg_pass 옵션을 이용할 수 있습니다.

**참고**

**QGIS 서버와 서비스**

서비스 파일과 QGIS 서버를 사용할 때, 서버 쪽에서도 반드시 서비스를 환경설정해야만 합니다. QGIS 서버 지침서 문서를 참조해보십시오.

GNU/리눅스, 맥 OS 등 * 닉스 운영체제 상에서 사용자 홈 디렉터리에 `.pg_service.conf` 파일을 저장하면 PostgreSQL 클라이언트가 해당 파일을 자동적으로 인식할 것입니다. 예를 들면, 로그인한 사용자가 `web` 일 경우, (다른 환경 변수를 지정하지 않고) 직접 작업하려면 `/home/web/` 디렉터리에 `.pg_service.conf` 파일을 저장해야 합니다.

You can specify the location of the service file by creating a `PGSERVICEFILE` environment variable (e.g. run the `export PGSERVICEFILE=/home/web/.pg_service.conf` command under your *nix OS to temporarily set the `PGSERVICEFILE` variable).

`pg_config --sysconfdir` 명령어로 `.pg_service.conf` 파일이 있는 폴더를 지정하거나, 서비스 연결 파일이 있는 디렉터리를 지정한 `PGSYSCONFDIR` 환경 변수를 추가해서 시스템 수준에서 (모든 사용자가) 서비스 연결 파일을 이용할 수 있도록 할 수도 있습니다. 동일한 이름의 서비스 정의가 사용자 파일과 시스템 파일 양쪽에 모두 있을 경우, 사용자 파일을 우선합니다.

> **경고**
>
> 윈도우의 경우 다음과 같은 위험 부담이 있습니다:
>
> - 서비스 파일을 `.pg_service.conf` 가 아니라 `pg_service.conf` 로 저장해야 합니다.
> - 서비스 파일이 제대로 동작하려면 유닉스 서식으로 저장해야 합니다. 한 가지 방법을 소개하자면, Notepad++ 로 파일을 연 다음 *Edit* → *EOL Conversion* → *UNIX Format* → *File save* 메뉴 옵션을 선택하면 됩니다.
> - 다양한 방법으로 환경 변수를 추가할 수 있습니다. 확실하다고 알려진 검증된 방법은 *Control Panel* → *System and Security* → *System* → *Advanced system settings* → *Environment Variables* 메뉴에서 `PGSERVICEFILE` 과 그 경로를, 예를 들어 `C:\Users\John\pg_service.conf` 를 입력해서 추가하는 것입니다.
> - 환경 변수를 추가한 다음 컴퓨터를 재시작해야 할 수도 있습니다.

### Oracle Spatial 데이터베이스를 연결하기

The spatial features in Oracle Spatial aid users in managing geographic and location data in a native type within an Oracle database. The connection dialog proposes:

- *Name*: A name for this connection. It can be the same as *Database*;
- *Database*: SID or SERVICE_NAME of the Oracle instance;
- *Host*: The name of the database host;
- *Port*: Port number the Oracle database server listens on. The default port is `1521`;
- *Options*: Oracle connection specific options (e.g. OCI_ATTR_PREFETCH_ROWS, OCI_ATTR_PREFETCH_MEMORY). The format of the options string is a semicolon separated list of option names or option=value pairs;
- *Workspace*: Workspace to switch to;
- *Authentication*: For general details about the authentication dialog behavior, see 인증.

다음 체크박스를 선택적으로 활성화할 수 있습니다:

- ☑ *Only look in metadata table*: `all_sdo_geom_metadata` 뷰에 있는 테이블만 표시하도록 제한합니다. 이 옵션을 활성화하면 공간 테이블의 초기 표시 속도를 향상시킬 수 있습니다.
- ☑ *Only look for user's tables*: 공간 테이블 검색 시 사용자가 소유한 테이블만 검색하도록 제한합니다.
- ☑ *Also list tables with no geometry*: 도형이 없는 테이블도 기본적으로 목록에 표시할지 말지를 나타냅니다.
- ☑ *Use estimated table statistics for the layer metadata*: 레이어 설정 시 Oracle 테이블을 위한 다양한 메타데이터가 필요합니다. 이 메타데이터는 테이블 행 개수, 도형 유형 및 도형 열에 있는 데이터의 공간적 범위 같은 정보를 포함합니다. 테이블의 행이 많을 경우, 이런 메타데이터를 알아내는 데 시간이 걸릴 수도 있습니다. 이 옵션을 활성화하면, 다음과 같은 신속한 테이블 메타데이터 작업을 수행합니다: `all_tables.num_rows` 로부터 행 개수를 결정합니다. 레이어 필터를 적용한 경우라도 언제나 SDO_TUNE.EXTENTS_OF 함수로 테이블 범위를 결정합니다. 테이블에 있는 NULL 이 아닌 도형 행 처음 100 줄로부터 테이블 도형을 결정합니다.
- ☑ *Only existing geometry types*: 기존 도형 유형만 목록에 표시하고 다른 유형을 추가하려 하지 않습니다.
- ☑ *Include additional geometry attributes*: 추가적인 도형 속성을 포함시킵니다.

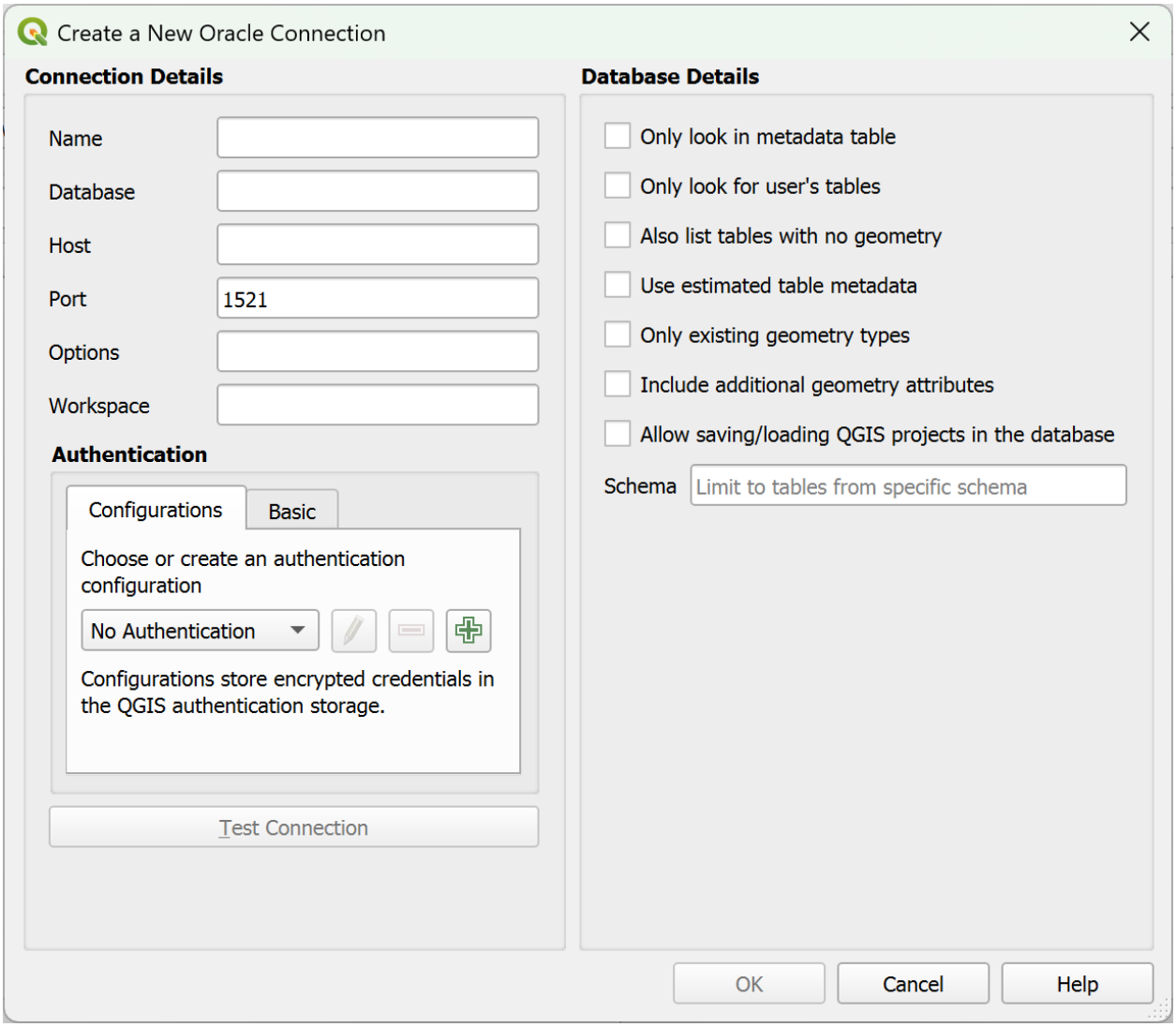



그림 11.13: Create a New Oracle Connection Dialog

- *Allow saving/loading QGIS projects in the database* - 데이터베이스에 QGIS 프로젝트를 저장하고 불러올 수 있습니다. 자세한 내용은여기를 참조하세요.
- *Schema*: Allows to specify a single schema to limit a connection to. When set, only tables from the matching schema will be included in the browser panel and data source select for the connection. This can be used to limit the database work required to populate tables for a connection pointing to a large database store.

> **팁**
>
> **Oracle 공간 레이어**
>
> 일반적으로, `USER_SDO_METADATA` 테이블의 항목이 Oracle 레이어를 정의합니다.
>
> 선택 도구가 정확하게 작동하도록 보장하려면, 사용자 테이블이 **기본 키 (primary key)** 를 가지고 있는 편이 좋습니다.

### MS SQL 서버를 연결하기

저장된 연결 생성하기 에서 언급했던 대로, QGIS 는 사용자가 MS SQL 서버 연결을 *Data Source Manager* 를 통해 생성할 수 있게 해줍니다.

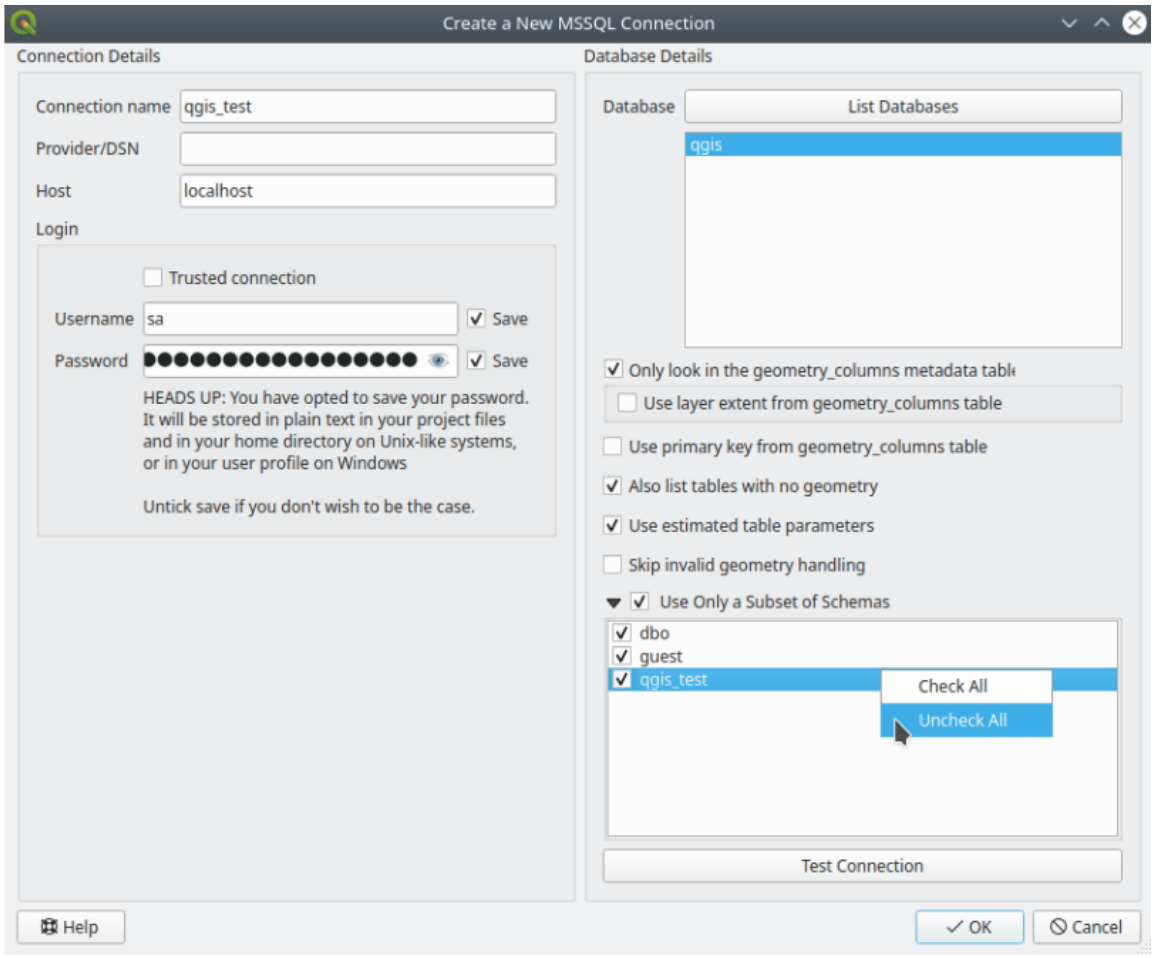



그림 11.14: MS SQL 서버 연결

새 MS SQL 서버 연결을 생성하려면, *Connection Details* 대화창에 다음 정보 가운데 일부를 제공해야 합니다:

- *Connection name*: 연결 이름
- *Provider/DSN*: 제공자/DSN(Data Source Name) 정보
- *Host*: 호스트 이름
- *Login*: 로그인 정보. *Save* 체크박스를 체크하면 사용자 자격 인증 정보를 저장할 수 있습니다.

*Database Details* 부분으로 가서 *List Databases* 버튼을 클릭하면 사용할 수 있는 데이터셋을 볼 수 있습니다. 원하는 데이터셋을 선택한 다음 *OK* 버튼을 누르십시오. 아니면 *Test Connection* 버튼을 클릭해서 연결 테스트를 수행할 수도 있습니다. *OK* 버튼을 클릭하고 나면 *Create a New MS SQL Server Connection* 대화창이 닫힐 것입니다. *Data Source Manager* 대화창에서 *Connect* 버튼을 누른 다음 레이어를 선택하고 *Add* 버튼을 클릭하십시오.

다음 옵션들을 선택적으로 활성화할 수 있습니다:

- *Only look in the geometry_columns metadata table*: restricts the available tables to the ones in the `geometry_columns` metadata table when scanning for tables. This can speed up the table scanning.
- *Use layer extent from geometry_columns table*: this option, dependent on the previous one, allows QGIS to skip extent calculation when loading layers and thus lowering the amount of time needed to load them. It relies on extent manually specified using additional QGIS-specific columns (`qgis_xmin`, `qgis_xmax`, `qgis_ymin`, `qgis_ymax`) in the `geometry_columns` table.
- *Use primary key from geometry_columns table*: allows QGIS to skip primary key calculation for views when loading them, thus lowering the amount of time needed to load them. It relies on names manually filled in a QGIS-specific `qgis_pkey` column set in the `geometry_columns` table. If more than one column is used for the primary key, they should be filled as comma separated values.
- *Also list table with no geometry*: 사용 가능 테이블 목록에 도형 열이 없는 테이블도 보이게 합니다.
- *Use estimated table parameters*: 평가된 테이블 메타데이터만 사용할 것입니다. 이 옵션을 체크하면 테이블 스캔 시간이 빨라지지만, 레이어 범위 같은 레이어 속성이 부정확해질 수도 있습니다.
- *Skip invalid geometry handling*: 무결하지 않은 도형을 가진 모든 레코드의 처리를 비활성화할 것입니다. 이 옵션을 체크하면 제공자의 속도가 빨라지지만, 테이블에 무결하지 않은 도형이 하나라도존재하는 경우 결과물이 예측하기 힘들어지고 누락된 레코드를 포함할 수도 있습니다. 사용자가 데이터베이스에 있는 모든 도형이 무결하며 새로 추가되는 도형 또는 테이블도 모두 무결할 것이라고 확신하는 경우에만 이 옵션을 체크하십시오.
- *Use only a Subset of Schemas*: MS SQL 연결에 대한 스키마를 필터링할 수 있게 해줍니다. 이 옵션을 체크하면, 체크한 스키마만 보이게 될 것입니다. 목록에 있는 어떤 스키마든 오른쪽 클릭으로 *Check* 또는 *Uncheck* 할 수 있습니다.

### SAP HANA 데이터베이스를 연결하기

> **참고**
>
> SAP HANA 데이터베이스에 연결하려면 SAP HANA 클라이언트가 필요합니다. SAP 개발 도구 웹사이트 에서 사용자 플랫폼 용 SAP HANA 클라이언트를 다운로드할 수 있습니다.

다음과 같은 파라미터를 입력할 수 있습니다:

- *Name*: 이 연결의 이름입니다.

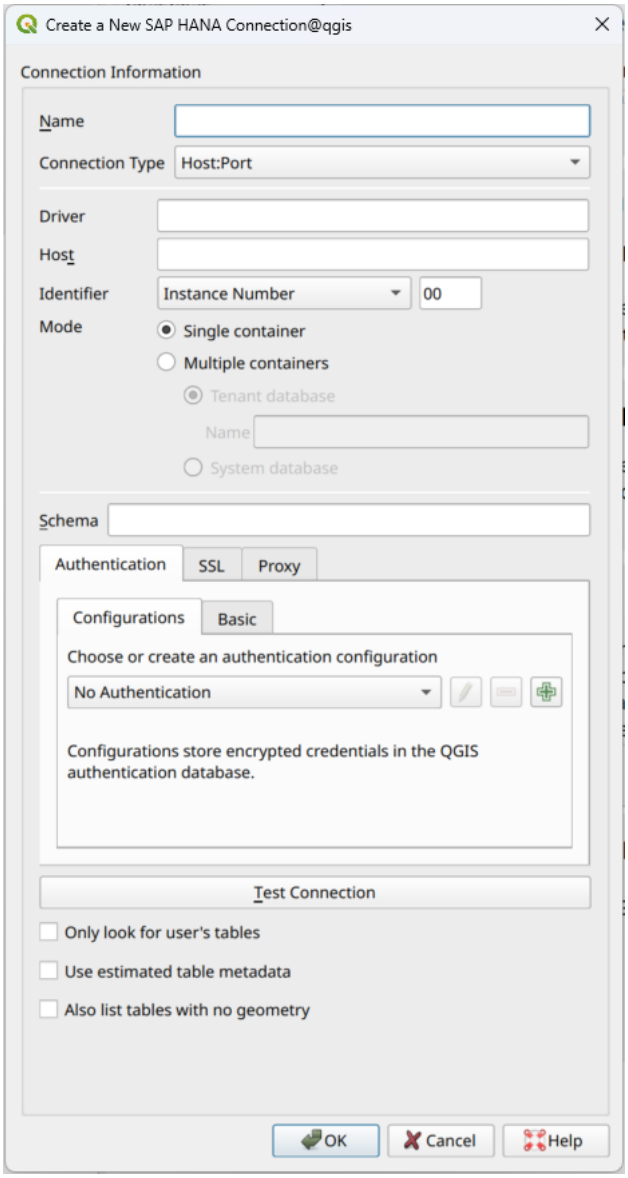



그림 11.15: 새 SAP HANA 연결 생성 대화창

- *Driver* : HANA ODBC 드라이버의 이름으로, QGIS 64 비트 버전을 이용하는 경우 `HDBODBC`, QGIS 32 비트 버전을 사용하는 경우 `HDBODBC32` 입니다. 적절한 드라이버 이름이 자동으로 입력됩니다.
- *Driver* : `/etc/odbcinst.ini` 파일에 등록된 SAP HANA ODBC 드라이버의 이름, 또는 SAP HANA ODBC 드라이버를 가리키는 전체 경로 가운데 하나입니다. SAP HANA 클라이언트 설치 파일은 기본적으로 `/usr/sap/hdbclient/libodbcHDB.so` 에 ODBC 드라이버를 설치할 것입니다.
- *Host*: 데이터베이스 호스트의 이름입니다.
- *Identifier*: 호스트 상에서 연결할 인스턴스를 식별합니다. *Instance Number* 또는 *Port Number* 가운데 하나일 수 있습니다. 인스턴스 번호는 두 자릿수이고, 포트 번호는 1 에서 65,535 사이의 숫자입니다.
- *Mode*: SAP HANA 인스턴스의 실행 모드를 지정합니다. *Identifier* 를 *Instance Number* 로 설정한 경우에만 이 설정을 작업에 포함할 것입니다. 데이터베이스가 컨테이너를 여러 개 호스팅하고 있다면 *Tenant database* 에서 지정한 이름을 가진 테넌트 (tenant) 에 연결하거나, 아니면 시스템 데이터베이스에 연결할 수 있습니다.
- *Schema*: 선택적인 파라미터입니다. 스키마 이름을 지정한 경우, QGIS 는 해당 스키마의 데이터만 검색할 것입니다. 스키마 란을 비워놓는 경우 QGIS 가 모든 스키마의 데이터를 검색할 것입니다.
- *Authentication*: For general details about the authentication dialog behavior, see 인증.
- *SSL Settings*:
  - *Enable TLS/SSL encryption*: TLS 1.1 ~ TLS1.2 암호화를 활성화합니다. 서버가 사용 가능한 가장 높은 버전을 선택할 것입니다.
  - *Provider*: SSL 통신에 사용되는 암호화 라이브러리 제공자 (cryptographic library provider) 를 지정합니다. *sapcrypto* 는 모든 플랫폼 상에서 작동할 것입니다. *openssl* 는 리눅스와 맥 상에서 작동할 것입니다. *mscrypto* 는 윈도우 상에서 작동할 것입니다. *commoncrypto* 가 작동하려면 CommonCryptoLib 을 설치해야 합니다.
  - *Validate SSL certificate*: 이 옵션을 체크하면, *Trust store file with public key* 에서 지정한 신뢰 저장소 (TrustStore) 를 이용해서 인증서를 인증할 것입니다.
  - *Override hostname in certificate*: 서버의 신원을 검증하기 위해 쓰이는 호스트 이름을 지정합니다. 여기에서 지정한 호스트 이름이 연결을 성립시켰던 호스트 이름 대신 서버의 신원을 검증합니다. 호스트 이름을 `*` 로 지정하면, 서버의 호스트 이름이 인증되지 않습니다. 다른 와일드카드 문자는 사용할 수 없습니다.
  - *Keystore file with private key*: 현재로서는 무시됩니다. 이 파라미터는 향후 사용자명과 비밀번호 대신 인증서를 통한 인증을 하게 될 수도 있습니다.
  - *Trust store file with public key*: OpenSSL 을 사용하는 경우 서버의 공공 인증서를 담고 있는 신뢰 저장소 파일을 가리키는 경로를 지정합니다. 일반적으로 신뢰 저장소는 루트 인증서 또는 서버의 공공 인증서를 서명한 인증 기관의 인증서를 담고 있습니다. 사용자가 CommonCryptoLib 또는 msCrypto 암호화 라이브러리를 사용하고 있다면 이 속성을 비워두십시오.
- *Only look for user's tables*: 이 옵션을 체크하면, QGIS 가 데이터베이스에 연결된 사용자가 소유한 테이블 및 뷰만 검색합니다.
- *Use estimated table metadata*: If checked, estimated table metadata will be used if available. For large tables, this avoids slow table loads and potentially expensive computations, but may result in incorrect layer properties such as layer extent. The fast extent estimation is available starting with QRC1/2024 and SP8 in HANA Cloud and HANA On-Premise respectively.
- *Also list tables with no geometries*: 이 옵션을 체크하면, QGIS 가 공간 열을 담고 있지 않은 테이블 및 뷰도 검색합니다.

> **팁**
>
> **SAP HANA 클라우드를 연결하기**
>
> SAP HANA 클라우드 인스턴스에 연결하고 싶다면 일반적으로 *Port Number* 를 `443` 으로 설정하고 *Enable TLS/SSL encryption* 옵션을 활성화해야만 합니다.

### 데이터베이스 레이어 불러오기

Once you have one or more connections defined to a database (see section 저장된 연결 생성하기), you can load layers from it. Of course, this requires that data are available. See section *PostgreSQL* 로 데이터 가져오기 for a discussion on importing data into a PostgreSQL database.

데이터베이스에서 레이어를 불러오려면 다음 단계를 거쳐야 합니다:

1. Open the corresponding tab of the database in the *Data Source Manager* dialog.
2. Choose the connection name from the drop-down list and press *Connect*.
3. The table below will be filled with your data grouped by schema, with a number of metadata information helpful for loading.
4. *Also list tables with no geometry* 체크박스를 활성화하거나 비활성화합니다.
5. *Search Options* 의 일부 옵션을 이용해서 사용자의 검색과 일치하는 테이블 목록을 줄일 수 있습니다. *Connect* 버튼을 누르기 전에 이 옵션을 설정하면, 데이터베이스를 불러오는 시간을 줄일 수도 있습니다.
6. 이용할 수 있는 레이어 목록에서 사용자가 추가하려 하는 레이어 (들) 를 찾습니다.
7. 그 레이어를 클릭해서 선택하십시오. `Shift` 또는 `Ctrl` 키를 누른 채 클릭하면 레이어 여러 개를 선택할 수 있습니다.

   Layers can be selected only if they have no warning icon at the left-hand side of their first column. This may indicate an issue to detect:

   - features geometry type: in which case you can select the appropriate one in the drop-dowmn list of the *Spatial Type* column
   - layer CRS: you can enter the correct code in the *SRID* column
   - layer's primary key, in order to unequivocally identify each feature: this can be fixed by selecting one or more attributes in the drop-down list at the corresponding *Feature id* column.

   > **팁**
   >
   > **Use the first column to store "primary keys"for views**
   >
   > Since PostgreSQL views don't support primary keys, a unique attribute or combination of attributes should always be selected. To help users to speed-up workflows, QGIS automatically selects the first attribute in the view. Therefore, users can define their views in a way that a unique column is in the first position of the view's definition. In this way, the view will be loaded with no extra interaction and the warning icon will never appear.

8. 가능한 경우, *Set Filter* 버튼을 눌러 (또는 레이어를 더블 클릭해서) *Query Builder* 대화창 (쿼리 작성기 참조) 을 연 다음 선택한 레이어에서 어떤 피처를 불러올 지 정의합니다. 필터 표현식은 `sql` 열에 표시됩니다. *Layer Properties* ▸ *General* ▸ *Provider Feature Filter* 프레임에서 이 제약 조건을 제거하거나 편집할 수 있습니다.

9. `Select at id` 열에 있는 체크박스는 기본적으로 활성화돼 있는데, 이를 통해 속성 없이 피처 ID 만 얻게 되기 때문에 대부분의 경우 데이터를 불러오는 속도가 향상됩니다.
10. *Add* 버튼을 클릭해서 맵에 레이어를 추가합니다.

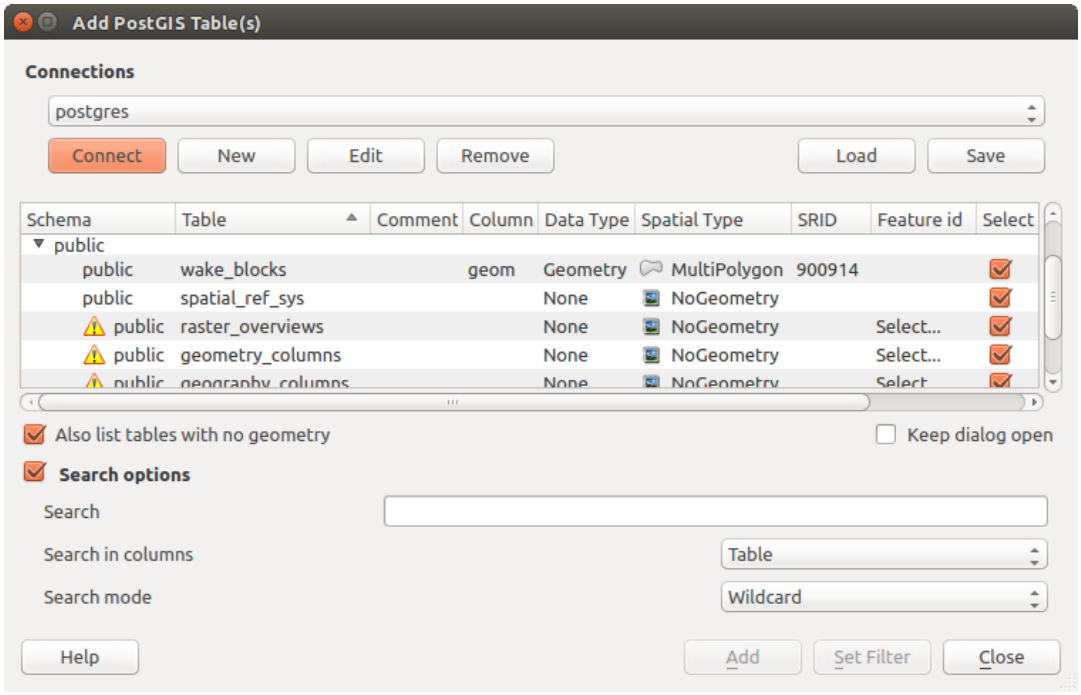



그림 11.16: Add PostgreSQL Table(s) Dialog

> **팁**
>
> **탐색기 패널을 사용해서 데이터베이스 테이블 불러오기 속도 향상**
>
> *Data Source Manager* 에서 데이터베이스 테이블을 추가하는 작업은 QGIS 가 각 테이블을 추가하기 전에 통계 및 속성을 (예를 들어 도형 유형 및 필드, 좌표계, 피처 개수) 가져오기 때문에 시간이 걸리는 경우가 있습니다. 이를 피하려면, 연결을 설정한 다음탐색기 패널 또는데이터베이스 관리자 를 사용해서 맵 캔버스로 데이터베이스 테이블을 드래그 & 드롭하는 편이 좋습니다.

### 11.1.4 레이어 메타데이터 검색 패널

QGIS 는 기본적으로 메타데이터 저장소를 지원하는 연결 또는 데이터 제공자에서 레이어 메타데이터를 검색합니다. (자세한 내용은데이터베이스에 메타데이터 저장하기 를 참조하세요.) *Metadata search* 패널에서 레이어를 레이어의 메타데이터로 탐색하고 (더블클릭하거나 *Add* 버튼을 눌러) 프로젝트에 추가할 수 있습니다. 레이어 목록을 다음으로 필터링할 수 있습니다:

- 텍스트: 메타데이터 속성 집합 (식별자, 제목, 요약) 을 검색합니다.
- 공간 범위: 현재프로젝트 범위 또는 맵 캔버스 범위를 사용해서 검색합니다.
- 레이어 (도형) 유형

> **참고**
>
> 메타데이터의 소스는 플러그인으로 확장시킬 수 있는 레이어 메타데이터 제공자 시스템을 통해 시행됩니다.

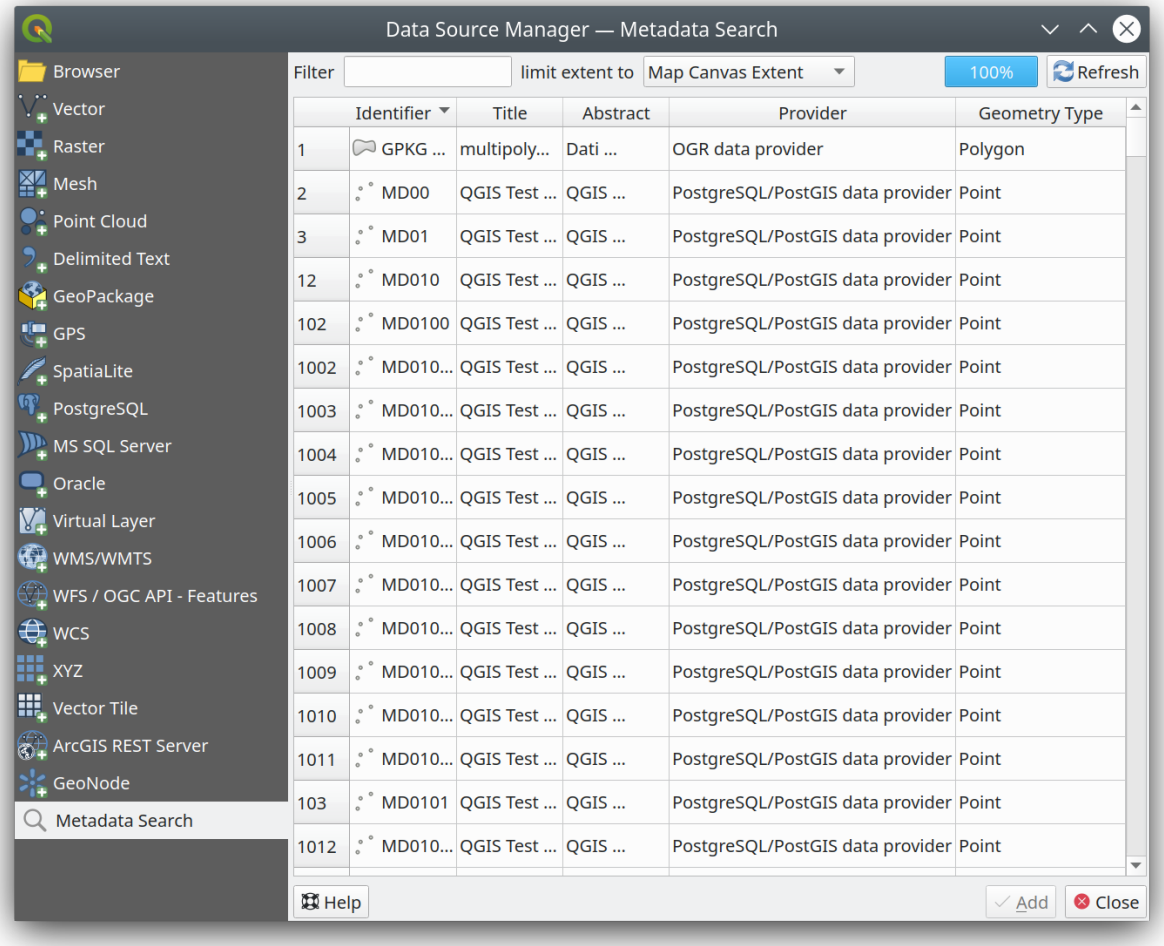



그림 11.17: 레이어 메타데이터 검색 패널

### 11.1.5 QGIS 사용자 지정 포맷

QGIS 는 2 개의 사용자 지정 포맷을 제안합니다:

- 임시 스크래치 레이어 (Temporary Scratch Layer): 프로젝트에 종속된 메모리 레이어 (자세한 내용은새 임시 스크래치 레이어 생성하기 참조)
- 가상 레이어 (Virtual Layer): 다른 레이어 (들) 에 대한 쿼리에서 산출된 레이어 (자세한 내용은가상 레이어 생성 참조)

### 11.1.6 QLR - QGIS 레이어 정의 파일

레이어 컨텍스트 메뉴의 *Export → Save As Layer Definition File*…메뉴를 통해 레이어 정의를레이어 정의 파일 (QLR - `.qlr`) 로 저장할 수 있습니다.

QLR 포맷은"완전한"QGIS 레이어를 다른 QGIS 사용자들과 공유할 수 있게 해줍니다. QLR 파일은 데이터소스를 가리키는 링크 및 레이어 스타일 작업에 필요한 모든 QGIS 스타일 정보를 담고 있습니다.

QLR 파일은 탐색기 패널에 표시되며, 레이어 패널에 레이어를 (저장된 레이어 스타일과 함께) 추가하는 데 사용할 수 있습니다. 시스템 파일 관리자에서 맵 캔버스로 QLR 파일을 드래그 & 드롭할 수도 있습니다.

Available actions for QLR files in the Browser Panel are:

- *Export Layer → To file*
- *Add Layer to Project*
- *Layer Properties*…

### 11.1.7 웹 서비스 연결

QGIS 를 통해 서로 다른 OGC 웹 서비스 유형 (WM(T)S, WFS(-T), WCS, CSW 등등) 에 접근할 수 있습니다. QGIS 서버 덕분에, 이 서비스들을 웹 상에 공개할 수도 있습니다. QGIS-Server-manual 에서 이런 성능에 대해 설명하고 있습니다.

#### 벡터 타일 서비스 사용하기

*Data Source Manager* 대화창의  *Vector Tiles* 탭 또는 *Browser* 패널에 있는 *Vector Tiles* 항목의 컨텍스트 메뉴를 통해 벡터 타일 서비스를 추가할 수 있습니다. *New Generic Connection*…또는 *New ArcGIS Vector Tile Service Connection*…가운데 하나로 서비스할 수 있습니다.

다음을 추가해서 서비스를 설정하십시오:

- *Name*: 서비스 이름입니다.
- *Style URL*: 맵박스 (MapBox) GL JSON 스타일 환경 설정을 가리키는 URL 입니다. 이 옵션을 지정하면, 연결로부터 레이어가 QGIS 에 연결될 때마다 해당 스타일을 적용할 것입니다. ArcGIS 벡터 타일 서비스 연결의 경우 이 URL 이 서버 환경 설정에 지정된 기본 스타일 환경 설정을 무시합니다.

  You can load vector tiles directly from a *Style URL*. The data source is automatically parsed from the style, and URLs with multiple sources are supported. That makes *Source URL* optional.
- the *Source URL*: of the type `http://example.com/{z}/{x}/{y}.pbf` for generic services and `http://example.com/arcgis/rest/services/Layer/VectorTileServer` for ArcGIS based services. The service must provide tiles in `.pbf` format.
- ☑ *Min. Zoom Level* 과 ☑ *Max. Zoom Level*: 벡터 타일은 피라미드 구조입니다. 이 옵션을 사용하면 타일 피라미드로부터 레이어를 개별적으로 생성할 수 있습니다. 그 다음 QGIS 가 이 레이어들을 이용해서 벡터 타일을 렌더링할 것입니다.

  (오픈스트리트맵 벡터 타일이 사용하는) 메르카토르 투영법의 경우, 확대/축소 0 수준이 1:500,000,000 축척에서 전세계를 표현하고 14 수준에서 1:35,000 축척을 표현합니다.
- the *authentication* configuration if necessary
- *Referer*: 지리참조자를 설정합니다.

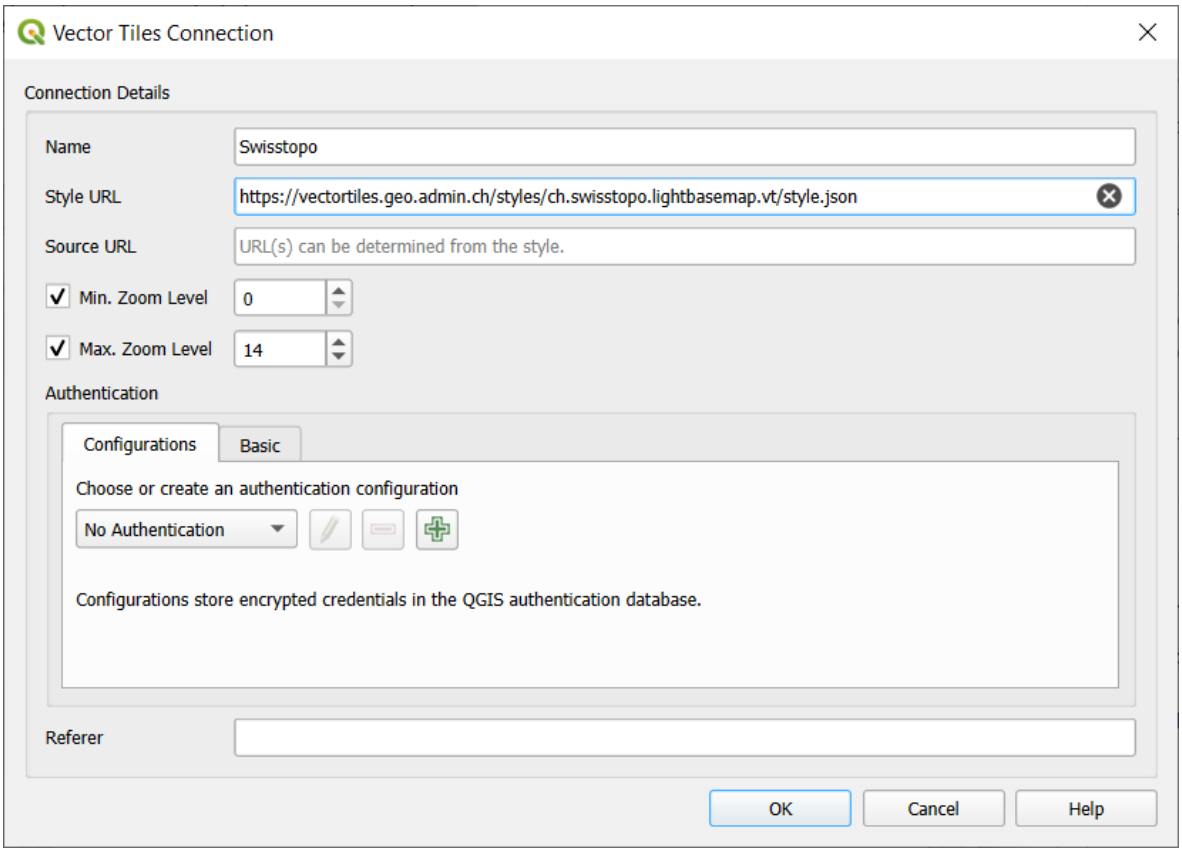



그림 11.18: Vector Tiles - Service configuration

그림 11.18 shows the dialog with the Vector Tiles service configuration.

*Data Source Manager* 대화창의 *Vector Tiles* 항목 또는 *Browser* 패널에 있는 *Vector Tiles* 의 컨텍스트 메뉴를 통해 환경 설정을 `.XML` 파일로 (*Save Connections*) 저장할 수 있습니다. 마찬가지로 파일로부터 환경 설정을 (*Load Connections*) 추가할 수도 있습니다.

벡터 타일 서비스로의 연결을 설정하고 나면 다음 작업을 할 수 있습니다:

- *Edit*: 벡터 타일 연결 설정을 편집할 수 있습니다.
- *Remove*: 연결을 제거할 수 있습니다.
- *Browser* 패널에서 항목을 오른쪽 클릭하면:
  - *Add layer to project*: 이 메뉴는 물론 더블 클릭으로도 레이어를 추가할 수 있습니다.
  - *Layer Properties…*: 레이어 속성을 표시해서 서비스가 제공하는 데이터의 메타데이터 및 미리보기에 접근할 수 있습니다. 프로젝트에 레이어를 불러온 후라면 더 많은 설정을 사용할 수 있습니다.

### XYZ 타일 서비스 사용하기

*Data Source Manager* 대화창의  *XYZ* 탭 또는 *Browser* 패널에 있는 *XYZ Tiles* 항목의 컨텍스트 메뉴를 통해 XYZ 타일 서비스를 추가할 수 있습니다. 기본적으로, QGIS 는 몇몇 기본값 및 사용할 준비가 된 XYZ 타일 서비스를 제공합니다:

-  *Mapzen Global Terrain*: 프로젝트 용 전세계 DEM 소스에 즉시 접근할 수 있습니다. https://registry.opendata.aws/terrain-tiles/ 에서 자세한 내용 및 리소스를 찾을 수 있습니다.
-  *OpenStreetMap*: 전세계 2 차원 맵에 접근할 수 있습니다. 그림 11.19 은 오픈스트리트맵 XYZ 타일 서비스의 환경 설정을 보여주는 대화창입니다.

새 서비스를 추가하려면, *New* 버튼을 (또는 탐색기 패널에서 *New Connection* 버튼을) 클릭하고 다음을 설정하십시오:

- *Name*: 서비스 이름입니다.
- *URL*: `http://example.com/{z}/{x}/{y}.png` 또는 `file:///local_path/{z}/{x}/{y}.png` 같이 추가하면 됩니다.
- the *authentication* configuration if necessary
- *Min. Zoom level* 과 *Max. Zoom level*
- *Referer*: 지리참조자를 설정합니다.
- *Tile Resolution*: *Unknown (not scaled)*, *Standard (256x256 / 96DPI)*, 또는 *High (512x512 / 192DPI)* 가운데 하나로 타일 해상도를 설정합니다.
- *Interpretation*: WMTS/XYZ 래스터 데이터셋을 사전 정의 인코딩 스키마를 따르는 단일 밴드 부동소수점 유형의 래스터 레이어로 변환합니다. *Default* (변환 없음), *MapTiler Terrain RGB* 및 *Terrarium Terrain RGB* 스키마를 지원합니다. 선택한 변환기가 각 픽셀에 대해 RGB 소스 값을 부동소수점형 값으로 번역할 것입니다. 타일 레이어를 불러오고 나면, 레이어를 단일 밴드 부동소수점형 래스터 레이어로 표시하고 QGIS 의 일반래스터 렌더링 작업자 를 통해 스타일 작업을 할 수 있습니다.

*OK* 버튼을 눌러 연결하십시오. 그러면 다음 작업을 할 수 있습니다:

- *Add*: 프로젝트에 새 레이어를 추가할 수 있습니다. 설정에서 지정한 이름으로 불러옵니다.
- *Edit*: XYZ 타일 연결 설정을 편집할 수 있습니다.
- *Remove*: 연결을 제거할 수 있습니다.
- *Browser* 패널에서 항목을 오른쪽 클릭하면:

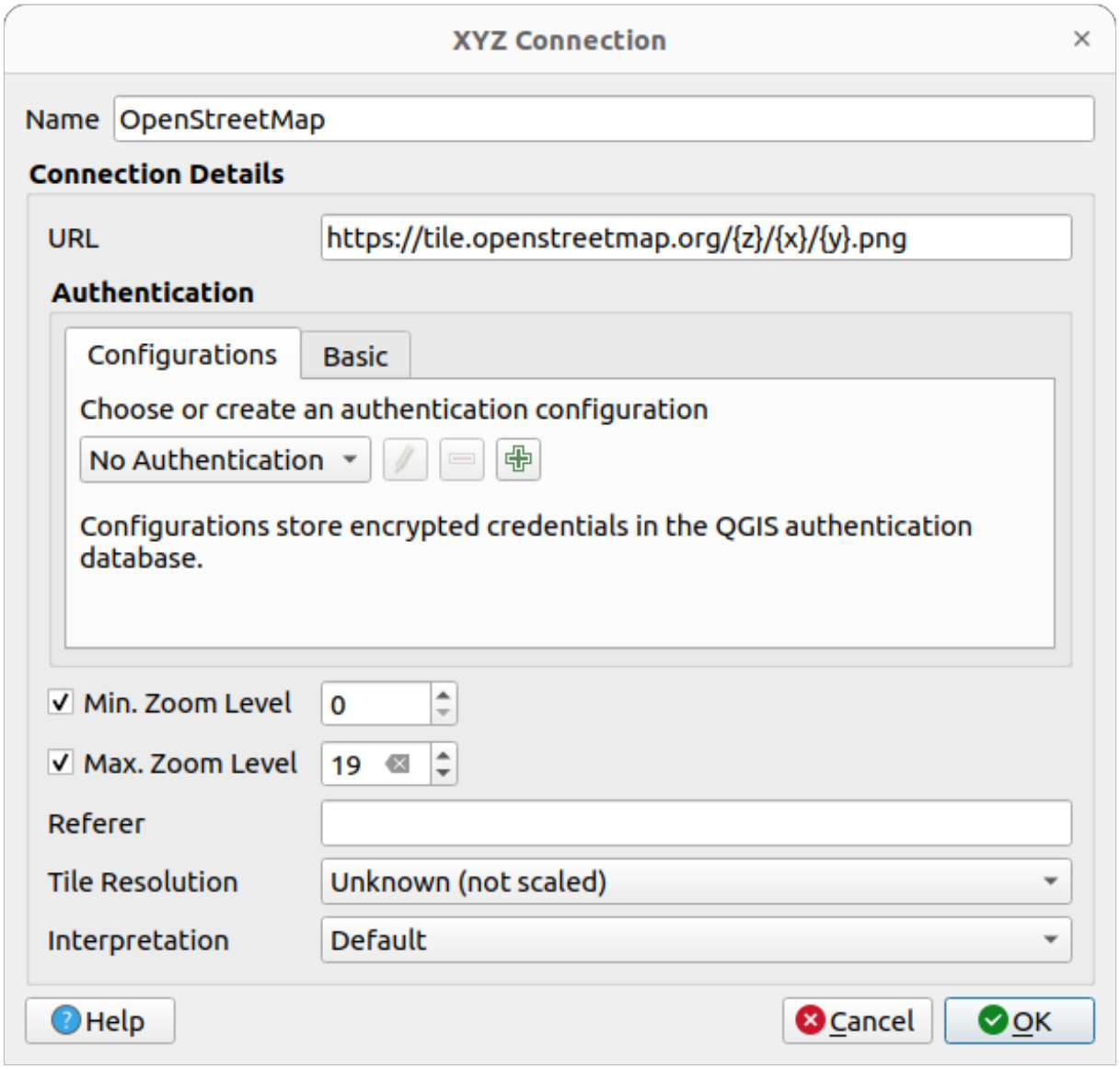



그림 11.19: XYZ 타일 - 오픈스트리트맵 환경 설정

  - *Export layer…* ▸ *To File*: XYZ 타일 레이어를래스터로 저장 합니다.
  - *Add layer to project*: 이 메뉴는 물론 더블 클릭으로도 레이어를 추가할 수 있습니다.
  - *Layer Properties…*: 레이어 속성을 표시해서 서비스가 제공하는 데이터의 메타데이터 및 미리보기에 접근할 수 있습니다. 프로젝트에 레이어를 불러온 후라면 더 많은 설정을 사용할 수 있습니다.

*Data Source Manager* 대화창의 *XYZ* 항목 또는 *Browser* 패널에 있는 *XYZ Tiles* 의 컨텍스트 메뉴를 통해 환경 설정을 `.XML` 파일로 (*Save Connections*) 저장할 수 있습니다. 마찬가지로 파일로부터 환경 설정을 (*Load Connections*) 추가할 수도 있습니다.

오픈스트리트맵의 XML 파일의 내용은 다음과 비슷하게 보입니다:

```
<!DOCTYPE connections>
<qgsXYZTilesConnections version="1.0">
  <xyztiles url="https://tile.openstreetmap.org/{z}/{x}/{y}.png"
   zmin="0" zmax="19" tilePixelRatio="0" password="" name="OpenStreetMap"
   username="" authcfg="" referer=""/>
</qgsXYZTilesConnections>
```

**팁**

**연결을 생성하지 않고 XYZ 타일 불러오기**

사용자 프로파일에 꼭 XYZ 타일 서비스 연결 설정을 저장하지 않더라도 프로젝트에 XYZ 타일을 추가할 수 있습니다. (예를 들면 그냥 한 번만 쓰면 되는 데이터셋 같은 경우 말이죠.) *Data Source Manager* ▸ *XYZ* 탭에서 *Connection Details* 그룹에 있는 속성을 편집하십시오. 앞의 *Name* 필드를 `Custom` 으로 바꿔줘야 합니다. *Add* 버튼을 클릭해서 프로젝트에 레이어를 불러오십시오. 레이어가 기본 이름인 `XYZ Layer` 로 명명될 것입니다.

XYZ 타일 서비스 예시:

- 흑백 오픈스트리트맵: *URL*: `http://tiles.wmflabs.org/bw-mapnik/{z}/{x}/{y}.png`, *Min. Zoom Level*: 0, *Max. Zoom Level*: 19
- 구글 지도: *URL*: `https://mt1.google.com/vt/lyrs=m&x={x}&y={y}&z={z}`, *Min. Zoom Level*: 0, *Max. Zoom Level*: 19
- 오픈웨더맵 기온 지도: *URL*: `http://tile.openweathermap.org/map/temp_new/{z}/{x}/{y}.png?appid={api_key}` *Min. Zoom Level*: 0, *Max. Zoom Level*: 19

### ArcGIS REST 서버 이용하기

An ArcGIS REST Server can host many different types of web services (feature service, map service, image service, … ). ArcGIS REST Servers can be added via the  *ArcGIS REST Server* tab of the *Data Source Manager* dialog or the contextual menu of the *ArcGIS REST Servers* entry in the *Browser* panel:

1. Press *New* (respectively *New Connection*) and provide:
   - a *Name*: A name for the connection.
   - the *URL*: Main address of the ArcGIS REST Server.
   - a *Prefix*: Used to specify the proxy prefix in the URL, which is necessary for some ArcGIS servers that use web proxy prefixes.
   - a *Community endpoint URL*: Endpoint URL for ArcGIS Online or Portal for ArcGIS, used to access content groups.

- a *Content endpoint URL*: Endpoint URL for the content service. This is used to access items in ArcGIS Online or Portal for ArcGIS.
- the *authentication* credentials if necessary.
- a *Referer*: The referer URL to be sent in the HTTP headers when making requests to the server. This may be required by some servers for authentication purposes.

> **참고**
>
> 탐색기 패널에 있는 콘텐츠 그룹을 통해, 대응하는 포털 종말점 URL 이 설정된 ArcGIS 피처 서비스 연결을 탐색해볼 수 있습니다.
>
> 연결에 포털 종말점이 설정되어 있는 경우, 브라우저에 있는 연결을 펼치면 보통 표시되는 서비스 전체 목록 대신"Groups"및"Services"폴더를 표시할 것입니다. 그룹 폴더를 펼치면 사용자가 멤버로 있는 모든 콘텐츠 그룹 목록을 표시할 것입니다. 이 콘텐츠 그룹들 각각을 펼쳐보면 해당 그룹에 속해 있는 서비스 항목을 표시합니다.

2. Then press *OK* to validate the configuration settings. These configurations can be saved to `.XML` file (*Save*). Likewise, they can be added from a file (*Load*).
3. Once a connection to an ArcGIS REST Server is set, it is possible to:
   - *Edit*: ArcGIS REST 서버 연결 설정을 편집할 수 있습니다.
   - *Remove*: 연결을 제거할 수 있습니다.
   - *Refresh*: 연결을 새로고침할 수 있습니다.
4. Press *Connect* to request the server and display its contents. They are organized in a tree structure whose nodes depend on the connection's endpoint.
5. Expand the tree to find and select the layers of interest. Their *Coordinate Reference System* is displayed at the bottom of the dialog.
6. For raster-based layers, you can select the *Image encoding* to use among a number of image formats advertized by the target service ( e.g., `PNG`, `JPG`, `GIF`, `SVG`, `SVGZ`, ⋯).
7. To add the selected layers to the map canvas, press *Add* button.

   Because layers can sometimes load and render slowly on the client side, applying a filter to restrict the features retrieved from the service can significantly improve performance, since only the filtered features are requested from the server. This can be done by:

   - checking *Only request features overlapping the current view extent*
   - for vector layers (feature service), pressing *Add with filter* to apply attribute-based filters to the layer with 표현식 문자열 작성기 functions.

     In the *Browser* panel, right-click the ArcGIS REST Server layer and select *Add Filtered Layer to Project* will also open the builder dialog.

Most of the above tools to connect and access layers on an ArcGIS REST Server are also available from the *Browser* panel, within the contextual menu of the target node item. Depending on the node, you can also:

- *View Service Info* which will open the default web browser and display information on the requested service.
- View the *Layer Properties*⋯ and get access to the metadata and a preview of the data provided by the service. More settings are available when the layer has been loaded into the project.

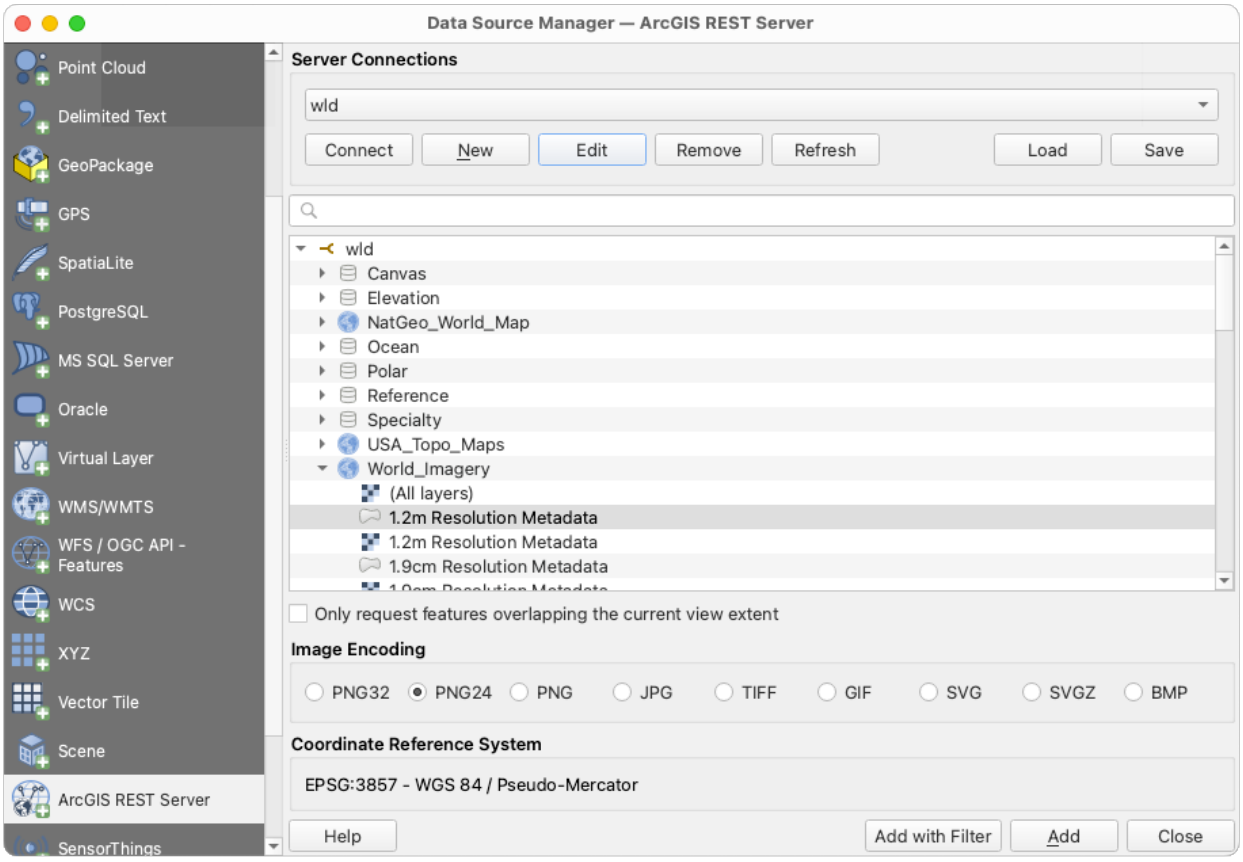



그림 11.20: Data Source Manager - ArcGIS REST Server

### Using 3D tiled scene services

QGIS supports multiple formats of 3D tiled datasets, grouped together as "tiled scenes". These include Cesium 3D Tiles and Quantized Mesh tiles.

To load a tiled scene dataset into QGIS, use the *Scene* tab in the *Data Source Manager* dialog.

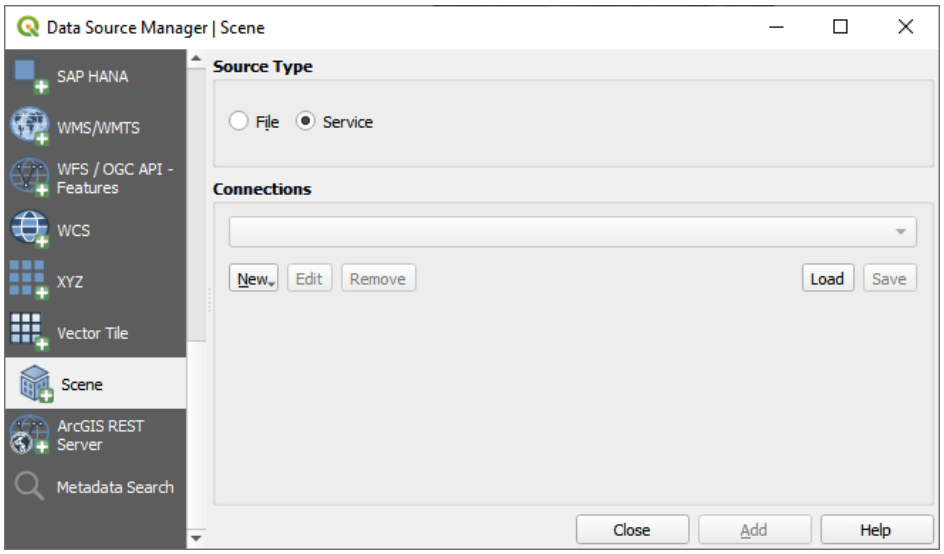



그림 11.21: 데이터소스 관리자 — 신 (scene)

Create a connection by clicking on *New*. You can add a *New Cesium 3D Tiles Connection* or a *New Quantized Mesh Connection*.

Choose a *Name* and set the *URL* to the URL of a layer description JSON file.

The URL may be remote (e.g. `http://example.com/tileset.json`) or local (e.g. `file:///path/to/tiles/tileset.json`).

The *authentication* configuration if necessary.

*Browser Panel* 에서도 서비스를 추가할 수 있습니다.

새 연결을 생성한 다음 *Add* 를 클릭하면 사용자 맵에 새 레이어를 추가할 수 있습니다.

### Using Cloud Connections

QGIS supports connections to cloud services like Alibaba Cloud OSS, Amazon S3, Google Cloud Storage, Microsoft Azure Blob Storage, Microsoft Azure Data Lake Storage, and OpenStack Swift Object Storage. You can load vector and raster data from these services into QGIS. Set up a new *Cloud* connection in the *Browser* panel by right-clicking on the *Cloud* entry and selecting *New Connection*. You will see a drop-down list of available cloud services. Select the service you want to connect to and fill in the required fields:

- *Name*: A name for the connection.
- *Bucket or Container*: The name of the bucket or container in the cloud service.
- *Object Key* (optional): The key of the object in the bucket or container.
- *Credentials*: The credentials to access the cloud service.

You can also choose to *Save Connection* to an XML file or *Load Connection* from an XML file.

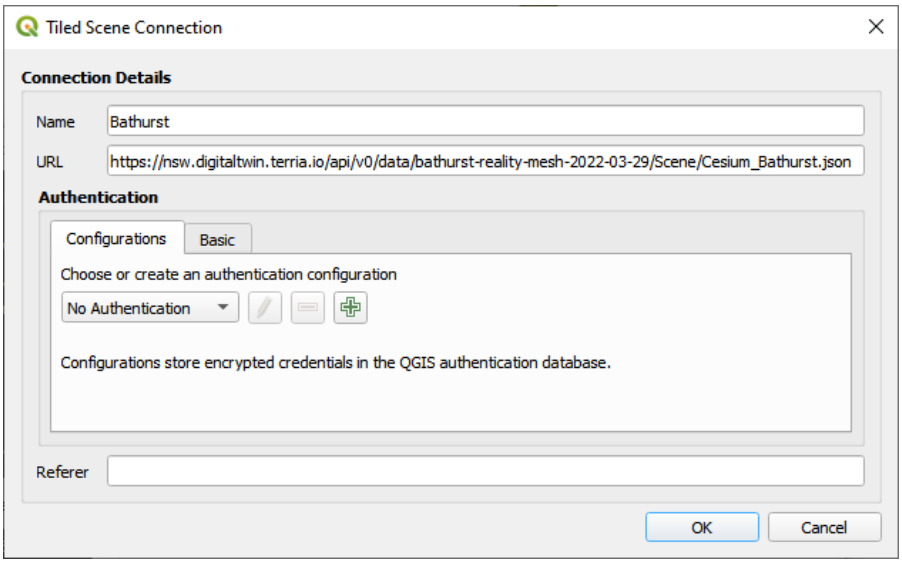



그림 11.22: 타일화된 신 연결

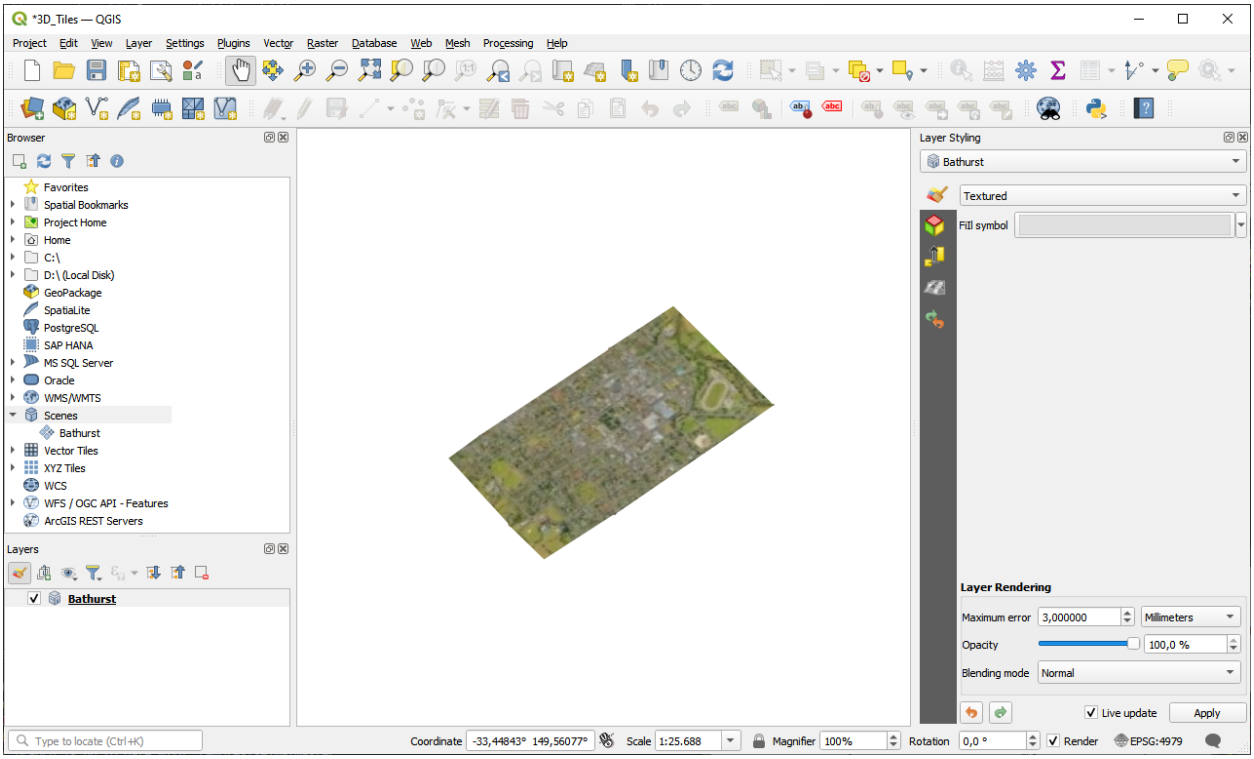



그림 11.23: 3D 타일 레이어 - 텍스처

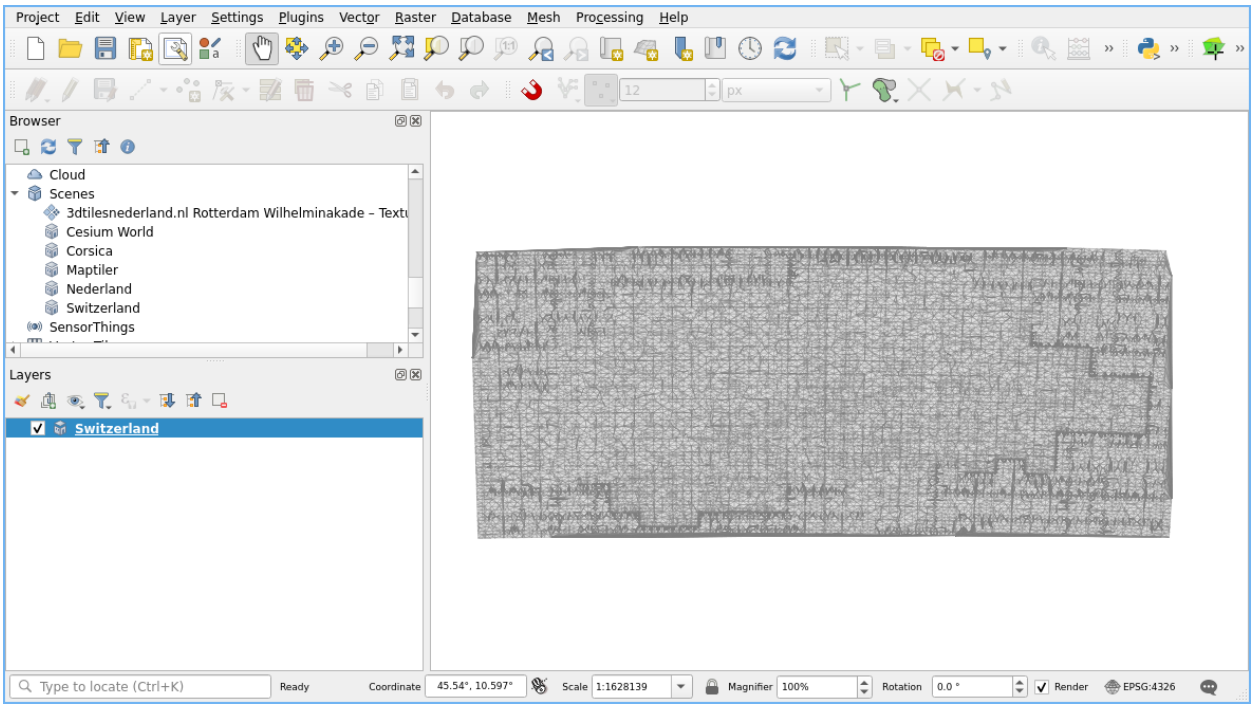



그림 11.24: Quantized Mesh layer

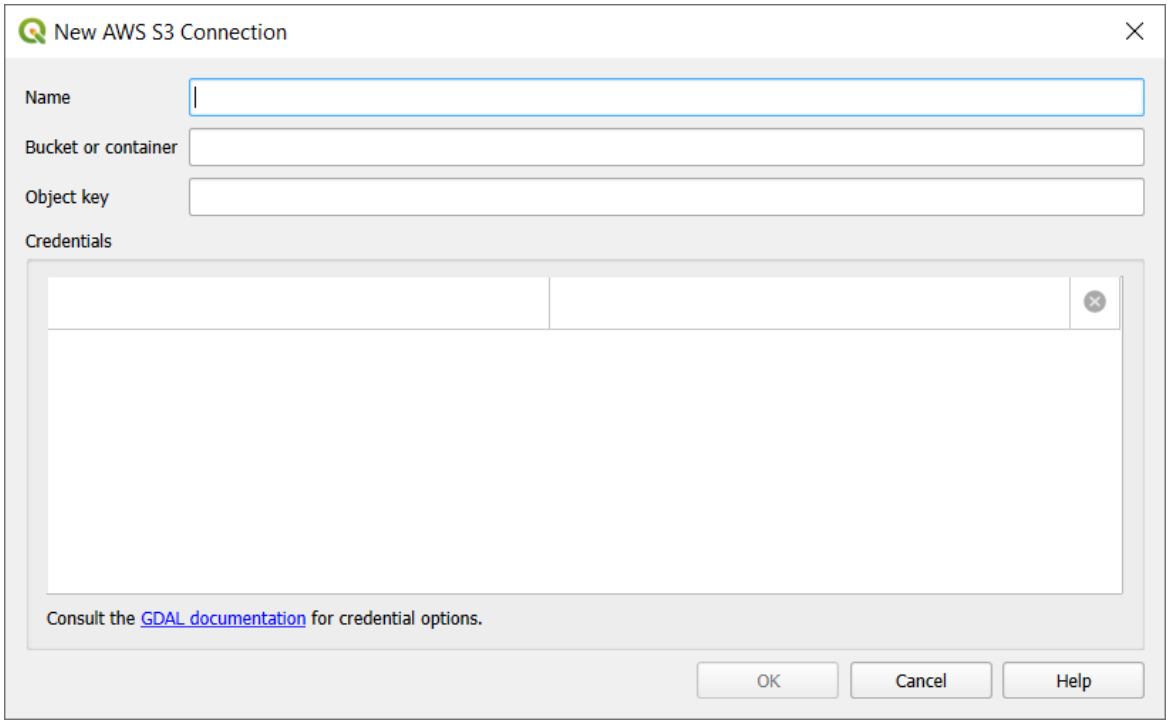



그림 11.25: Cloud Connection Dialog

## 11.2 레이어 생성

QGIS 는 다음과 같은 여러 방법을 통해 레이어를 생성할 수 있습니다:

- 아무것도 없이 처음부터 빈 레이어를 생성
- 기존 레이어로부터 생성
- 클립보드로부터 생성
- 하나 이상의 레이어를 바탕으로 하는 SQL 호환 쿼리의 산출물로서가상 레이어 를 생성

QGIS 는 서로 다른 포맷을 가져오기/내보내기할 수 있는 도구들도 제공합니다.

### 11.2.1 새 벡터 레이어 생성

QGIS 는 서로 다른 포맷들로 새 레이어를 생성할 수 있습니다. QGIS 는 GeoPackage, 셰이프파일, SpatiaLite, GPX 포맷 및 (메모리 레이어라고도 하는) 임시 스크래치 레이어를 생성할 수 있는 도구를 제공합니다. GRASS 플러그인 내부에서새 *GRASS* 레이어 를 생성할 수도 있습니다.

#### 새 GeoPackage 레이어 생성하기

To create a new GeoPackage layer, press the *New GeoPackage Layer…* button in the *Layer → Create Layer →* menu or from the *Data Source Manager* toolbar. You can also create a new GeoPackage layer through the *Browser Panel* by selecting the *Create Database and Layer…*. The *New GeoPackage Layer* dialog will be displayed as shown in 그림 11.26.

1. 첫 단계는 데이터베이스 파일의 위치를 가리키는 것입니다. *Database* 필드의 오른쪽에 있는 …버튼을 누른 다음 기존 GeoPackage 파일을 선택하거나 새 파일을 생성하면 됩니다. QGIS 가 사용자가 지정한 이름에 적합한 확장자를 자동적으로 추가할 것입니다.
2. 새 레이어/테이블의 이름을 (*Table name*) 입력합니다.
3. *Geometry type* 에서 도형 유형을 정의합니다. 도형이 없는 레이어가 아니라면 *Include Z dimension* 그리고/또는 *Include M values* 옵션을 지정할 수 있습니다.
4. 버튼으로 좌표계를 지정하십시오.

생성 중인 레이어에 필드를 추가하려면:

1. *Name*: 필드명을 입력하십시오.
2. *Type*: 데이터 유형을 선택하십시오. *Text data*, *Whole number* (integer 및 integer64 모두), *Decimal number*, *Date* 및 *Date and time*, *Binary (BLOB)* 및 *Boolean* 유형을 지원합니다.
3. *Maximum length*: 선택한 데이터 유형에 따라, 해당 값의 최대 길이를 입력하십시오.
4. *Add to Fields List* 버튼을 클릭하십시오.
5. 사용자가 추가해야 할 각 필드마다 앞의 단계를 반복합니다.
6. You can later change the fields order using the Move Up and Move Down buttons
7. 필요한 모든 필드를 정확하게 추가했다면 *OK* 를 클릭하십시오. QGIS 가 범례에 새 레이어를 추가할 것입니다. 기존 레이어 디지타이즈 작업 절에서 설명하는 대로 새 레이어를 편집할 수 있습니다.

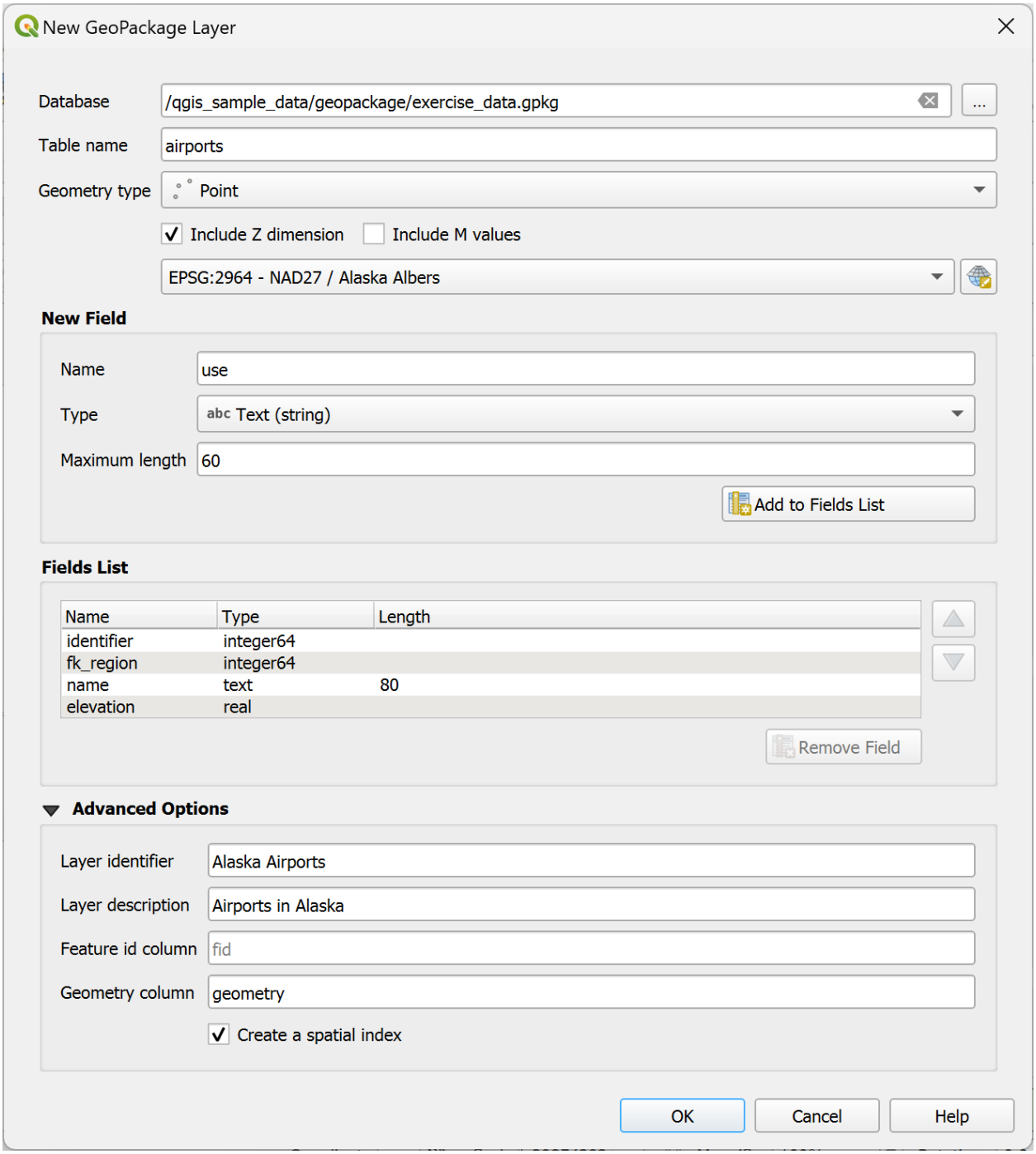



그림 11.26: 새 GeoPackage 레이어 생성 대화창

GeoPackage 레이어 생성 시 QGIS 는 레이어의 기본 키로 동작하는 `fid` 라는 이름의 *Feature id column* 을 기본적으로 생성합니다. 이 이름은 수정할 수 있습니다. 도형 필드가 있는 경우, 그 이름은 `geometry` 이며 *Create a spatial index* 옵션을 사용해서 공간 인덱스를 생성할 수 있습니다. 이 옵션들은 *Advanced Options* 아래에서 찾을 수 있으며, (사람이 해독할 수 있는 간단한 레이어명인) *Layer identifier* 와 *Layer description* 도 함께 있습니다.

GeoPackage 레이어는데이터베이스 관리자 를 통해 더 심도 있게 관리할 수 있습니다.

### 새 셰이프파일 레이어 생성하기

새 ESRI 셰이프파일 포맷 레이어를 생성하려면 *Layer* → *Create Layer* → 메뉴 또는 *Data Source Manager* 툴바에서 *New Shapefile Layer*…를 선택하십시오. 그림 11.27 와 같은 *New Shapefile Layer* 대화창이 열립니다.

1. *File name* 옆에 있는 …버튼을 사용해서 경로 및 파일명을 지정하십시오. QGIS 가 사용자가 지정한 이름에 적합한 확장자를 자동적으로 추가할 것입니다.
2. *File encoding*: 그 다음 데이터의 파일 인코딩을 설정하십시오.
3. 레이어의 *Geometry type* 을 선택하십시오: 도형 없음 (`.DBF` 포맷 파일로 산출), 포인트, 멀티포인트, 라인 또는 폴리곤
4. 도형이 추가적인 차원을 가져야 할지 지정하십시오: *None*, *Z (+ M values)* 또는 *M values*
5. 버튼으로 좌표계를 지정하십시오.

생성 중인 레이어에 필드를 추가하려면:

1. *Name*: 필드명을 입력하십시오.
2. Select the data *Type*. Only *Decimal number*, *Whole number*, *Text data*, *Date*, and *Boolean* data types attributes are supported.
3. *Length* 및 *Precision*: 선택한 데이터 유형에 따라, 해당 값의 길이 및 정밀도를 입력하십시오.
4. *Add to Fields List* 버튼을 클릭하십시오.
5. 사용자가 추가해야 할 각 필드마다 앞의 단계를 반복합니다.
6. You can later change the fields order using the Move Up and Move Down buttons
7. 필요한 모든 필드를 정확하게 추가했다면 *OK* 를 클릭하십시오. QGIS 가 범례에 새 레이어를 추가할 것입니다. 기존 레이어 디지타이즈 작업 절에서 설명하는 대로 새 레이어를 편집할 수 있습니다.

기본적으로 `id` 라는 첫번째 정수열을 추가하지만, 이를 제거할 수 있습니다.

### 새 SpatiaLite 레이어 생성하기

새 SpatiaLite 레이어를 생성하려면 *Layer* → *Create Layer* → 메뉴 또는 *Data Source Manager* 툴바에서 *New SpatiaLite Layer*…를 선택하십시오. 그림 11.28 과 같은 *New SpatiaLite Layer* 대화창이 열립니다.

1. 첫 단계는 데이터베이스 파일의 위치를 가리키는 것입니다. *Database* 필드의 오른쪽에 있는 …버튼을 누른 다음 기존 SpatiaLite 파일을 선택하거나 새 파일을 생성하면 됩니다. QGIS 가 사용자가 지정한 이름에 적합한 확장자를 자동적으로 추가할 것입니다.
2. 새 레이어용 이름을 (*Layer name*) 입력하십시오.
3. *Geometry type* 에서 도형 유형을 정의합니다. 도형이 없는 레이어가 아니라면 *Include Z dimension* 그리고/또는 *Include M values* 옵션을 지정할 수 있습니다.
4. 버튼으로 좌표계를 지정하십시오.

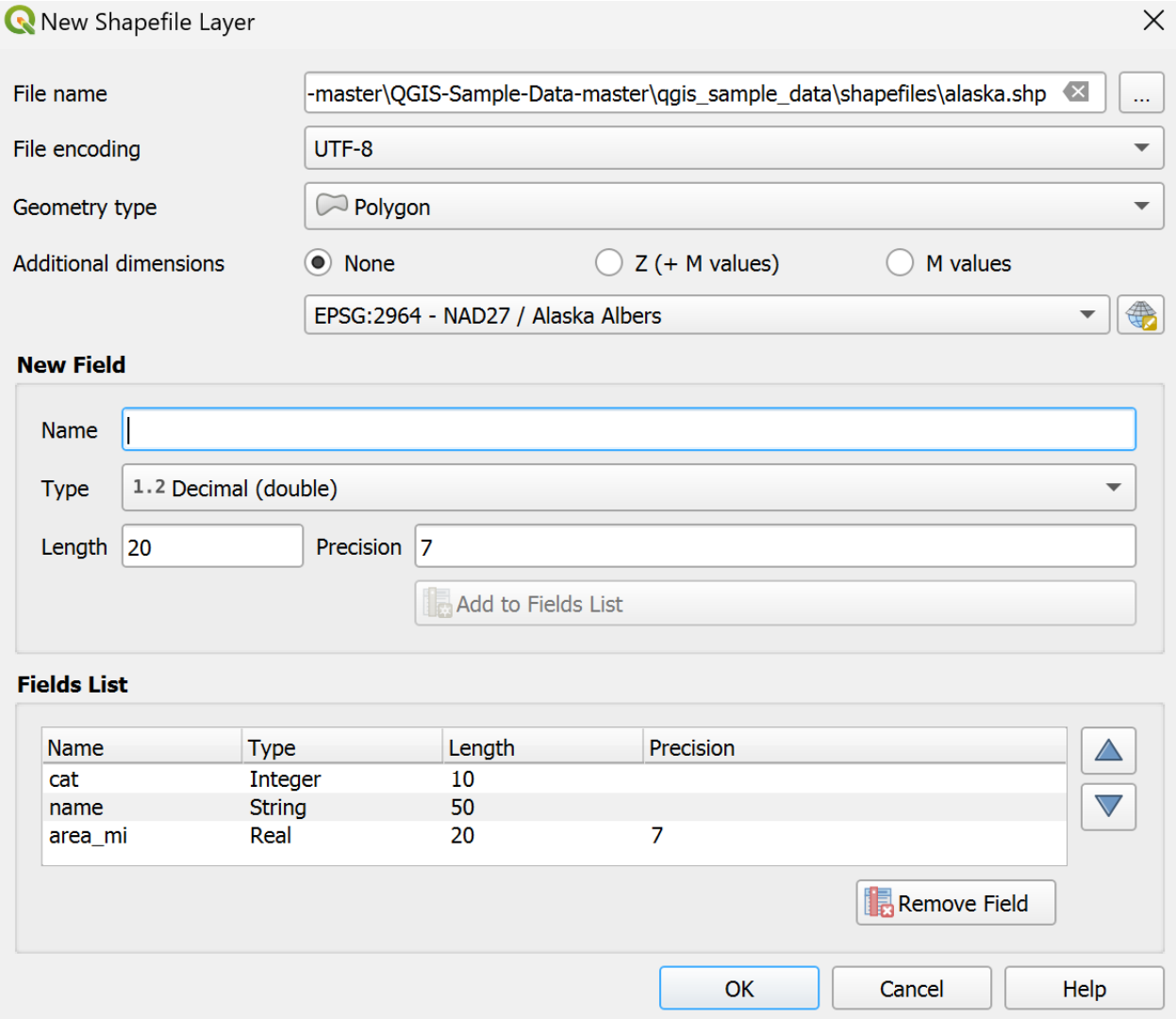



그림 11.27: 새 셰이프파일 레이어 생성 대화창

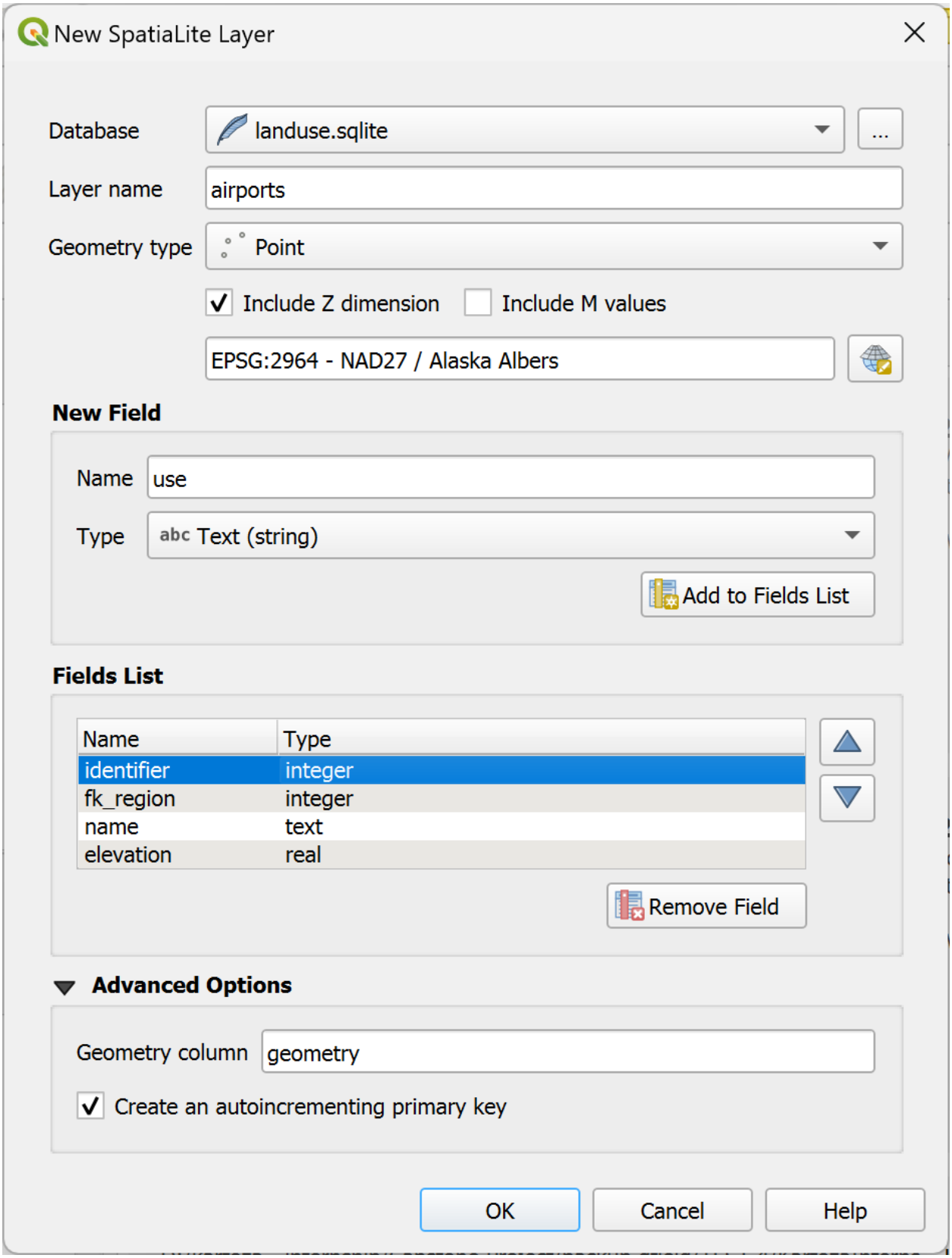



그림 11.28: 새 SpatiaLite 레이어 생성 대화창

생성 중인 레이어에 필드를 추가하려면:

1. *Name*: 필드명을 입력하십시오.
2. *Type*: 데이터 유형을 선택하십시오. *Text data*, *Whole number*, *Decimal number*, *Date* 및 *Date time* 유형을 지원합니다.
3. *Add to Fields List* 버튼을 클릭하십시오.
4. 사용자가 추가해야 할 각 필드마다 앞의 단계를 반복합니다.
5. You can later change the fields order using the Move Up and Move Down buttons
6. 필요한 모든 필드를 정확하게 추가했다면 *OK* 를 클릭하십시오. QGIS 가 범례에 새 레이어를 추가할 것입니다. 기존 레이어 디지타이즈 작업 절에서 설명하는 대로 새 레이어를 편집할 수 있습니다.

원한다면 *Advanced Options* 부분에 있는 *Create an autoincrementing primary key* 를 체크해서 오름차순 기본 키를 생성할 수 있습니다. (기본 이름이 `geometry` 인) *Geometry column* 을 재명명할 수도 있습니다.

SpatiaLite 레이어는데이터베이스 관리자 를 통해 더 심도 있게 관리할 수 있습니다.

### 새 메시 레이어 생성하기

새 메시 레이어를 생성하려면, *Layer* → *Create Layer* → 메뉴 또는 *Data Source Manager* 툴바에서 *New Mesh Layer …*를 선택하십시오. 그림 11.29 과 같은 *New Mesh Layer* 대화창이 열립니다.

1. 첫 단계는 메시 파일의 위치를 가리키는 것입니다. *File name* 란의 오른쪽에 있는 …버튼을 누른 다음 기존 메시 파일을 선택하거나 새 파일을 생성하면 됩니다.
2. *Layer name*: 레이어 이름을 입력하십시오. 레이어가 *Layers* 패널에 이 이름으로 표시될 것입니다.
3. *File format*: 파일 포맷을 선택하십시오. 현재 `2DM Mesh File (*.2dm)`, `Selafin File (*.slf)` 및 `UGRID (*.nc)` 메시 파일 포맷을 지원합니다.
4. 데이터셋에 할당할좌표계 를 선택하십시오.
5. 이 단계를 거치면 나중에 꼭짓점을 디지타이즈하고 데이터셋 그룹을 추가할 수 있는 비어 있는 레이어를 생성할 것입니다. 하지만 기존 메시 레이어로 레이어를 초기 설정할 수도 있습니다. 예를 들어 새 레이어를 다른 레이어의 꼭짓점 또는 면으로 채울 수 있습니다. 그 방법은 다음과 같습니다:
   1. *Initialize Mesh using* 을 체크하십시오.
   2. 그리고 *Mesh from the current project* 또는 *Mesh from a file* 가운데 하나를 선택하십시오. 선택한 메시 파일에 대한 정보를 확인할 수 있습니다.

   메시 레이어의 프레임만 새 레이어로 전송된다는 사실을 기억하십시오. 메시 레이어의 데이터셋은 복사되지 않습니다.

### 새 GPX 레이어 생성하기

새 GPX 파일을 생성하려면:

1. *Layer* 메뉴에서 *Create Layer* → *New GPX Layer…*메뉴 항목을 선택하십시오.
2. 대화창에서 새 파일을 저장할 경로와 새 파일의 이름을 입력한 다음 *Save* 버튼을 누르십시오.
3. 새 레이어 3 개가 *Layers* 패널에 추가됩니다:
   - 포인트 레이어 - 이 레이어는 이름, 표고, 코멘트, 설명, 소스, URL 및 URL 이름을 저장하는 필드들을 가지고 있으며, 위치 (`waypoints`) 를 디지타이즈할 수 있습니다.

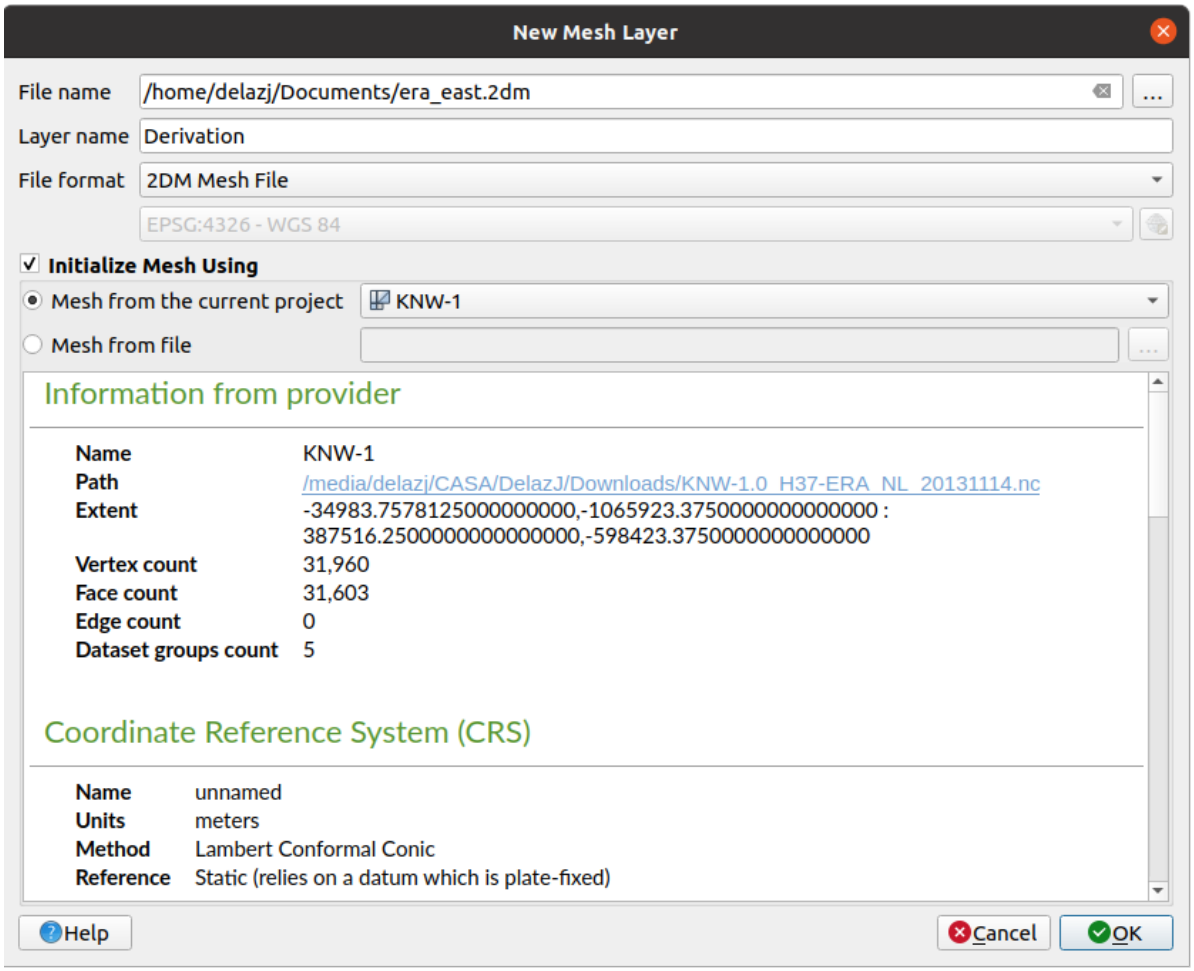



그림 11.29: 새 메시 레이어 생성 대화창

- 라인 레이어 - 이 레이어는 이름, 심볼, 번호, 코멘트, 설명, 소스, URL 및 URL 이름을 저장하는 필드들을 가지고 있으며, 계획된 경로 (`routes`) 를 구성하는 일련의 위치를 디지타이즈할 수 있습니다.
- 또다른 라인 레이어 - 이 레이어는 이름, 심볼, 번호, 코멘트, 설명, 소스, URL 및 URL 이름을 저장하는 필드들을 가지고 있으며, 시간 경과에 따른 수신기의 움직임을 추적 (`tracks`) 할 수 있습니다.

4. 이제기존 레이어 디지타이즈 작업 절에서 설명하는 대로 모든 레이어를 편집할 수 있습니다.

### 새 임시 스크래치 레이어 생성하기

임시 스크래치 레이어는 인 메모리 (in-memory) 레이어입니다. 디스크 상에 저장되지 않고 QGIS 종료 시 폐기될 레이어라는 뜻입니다. 일시적으로 필요한 피처를 저장하거나, 공간 처리 연산 도중의 매개 레이어어로서 사용하는데 편리합니다.

새 임시 스크래치 레이어를 생성하려면 *Layer* → *Create Layer* → 메뉴 또는 *Data Source Manager* 툴바에서 *New Temporary Scratch Layer…*를 선택하십시오. 그림 11.30 와 같은 *New Temporary Scratch Layer* 대화창이 열립니다.

1. *Layer name*: 레이어 이름을 입력하십시오.
2. *Geometry type*: 도형 유형을 선택하십시오. 다음과 같은 유형을 생성할 수 있습니다:
   - `No geometry`: 단순 테이블 역할을 하는 비도형 유형 레이어
   - `Point` 또는 `MultiPoint` 레이어
   - `LineString/CompoundCurve` 또는 `MultiLineString/MultiCurve` 레이어
   - `Polygon/CurvePolygon` 또는 `MultiPolygon/MultiSurface` 레이어
3. 도형 유형에 대해 데이터셋의 차원을 지정하십시오: *Include Z dimension* 그리고/또는 *Include M values* 를 가져야 할지 확인하십시오.
4. 버튼으로 좌표계를 지정하십시오.
5. 레이어에 필드를 추가하십시오. 다른 많은 포맷과는 달리, 임시 레이어는 어떤 필드도 없이 생성할 수 있습니다. 따라서 이 단계는 부가적입니다.
   1. *Name*: 필드명을 입력하십시오.
   2. *Type*: 데이터 유형을 선택하십시오. *Text*, *Whole number*, *Decimal number*, *Boolean*, *Date*, *Time*, *Date & Time* 및 *Binary (BLOB)* 유형을 지원합니다.
   3. *Length* 및 *Precision*: 선택한 데이터 유형에 따라, 해당 값의 길이 및 정밀도를 입력하십시오.
   4. *Add to Fields List* 버튼을 클릭하십시오.
   5. 사용자가 추가해야 할 각 필드마다 앞의 단계를 반복합니다.
   6. You can later change the fields order using the Move Up and Move Down buttons
6. 임시 스크래치 레이어 설정에 만족한다면 *OK* 를 클릭하십시오. QGIS 가 *Layers* 패널에 새 레이어를 추가할 것입니다. 기존 레이어 디지타이즈 작업 절에서 설명하는 대로 새 레이어를 편집할 수 있습니다.

미리 내용을 채운 임시 스크래치 레이어도 생성할 수 있습니다. 예를 들자면 클립보드 (클립보드로부터 새 레이어 생성 참조) 나공간 처리 알고리즘 의 결과물로부터 말이죠.

> **팁**
>
> **디스크 상에 메모리 레이어를 영구적으로 저장하기**

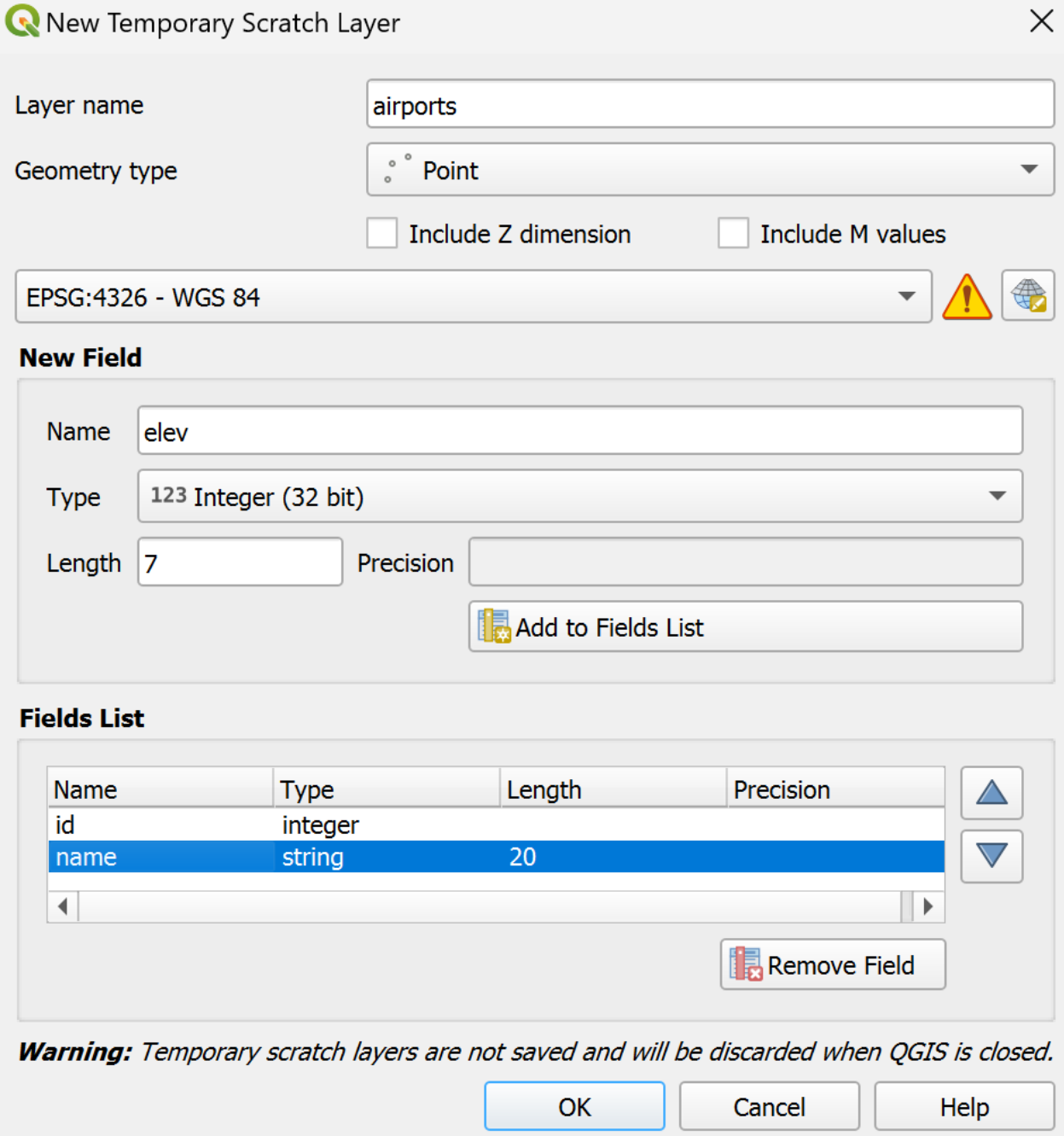



그림 11.30: 새 임시 스크래치 레이어 생성 대화창

임시 스크래치 레이어를 가진 프로젝트를 종료할 때 데이터 손실을 예방하려면, 해당 레이어를 QGIS 가 지원하는 모든 벡터 포맷으로 저장할 수 있습니다:

- 레이어 옆에 있는 표시 아이콘을 클릭하거나
- 레이어의 컨텍스트 메뉴에서 *Make permanent* 항목을 선택하거나
- 컨텍스트 메뉴의 *Export* → 항목 또는 *Layer* → *Save As…*메뉴를 선택합니다.

이 각 명령들은기존 레이어로부터 레이어 생성 절에서 설명하는 *Save Vector Layer as* 대화창을 엽니다. 대화창에서 저장한 파일이 *Layers* 패널에 있는 임시 레이어를 대체합니다.

### 11.2.2 기존 레이어로부터 레이어 생성

Layers (raster, vector and point cloud) can be saved in a different format and/or reprojected to a different coordinate reference system (CRS) using the *Layer* → *Save As…* menu or right-clicking on the layer in the *Layers panel* and selecting:

- *Export* → *Save As…* for raster and point cloud layers
- *Export* → *Save Features As…*또는 *Export* → *Save Selected Features As…*: 벡터 레이어용
- Drag and drop the layer from the layer tree to the PostgreSQL entry in the *Browser Panel*. Note that you must have a PostgreSQL connection in the *Browser Panel*.

#### 공통 파라미터

*Save Layer as…*대화창에는 레이어 저장 습성을 변경할 수 있는 파라미터가 몇 가지 있습니다. 다음은 래스터 및 벡터 레이어에 공통된 파라미터 가운데 일부입니다:

- *File name*: 디스크 상에 있는 파일의 위치입니다. 산출물 레이어를 참조할 수도 있고, 레이어를 저장하고 있는 컨테이너를 (예를 들면 GeoPackage, SpatiaLite 또는 오픈도큐먼트 스프레드시트 같은 데이터베이스 호환 포맷을) 참조할 수도 있습니다.
- *CRS*: 데이터를 재투영하기 위해 변경할 수 있습니다.
- *Extent*: 범위 선택기 위젯을 이용해서 입력물에서 내보내고자 하는 영역을 제한할 수 있습니다.
- *Add saved file to map*: 캔버스에 저장한 파일을 새 레이어로 추가합니다.

However, some parameters are specific to certain formats:

#### 래스터 특화 파라미터

내보내기 포맷에 따라, 일부 옵션을 사용하지 못 할 수도 있습니다:

- *Output mode*: **raw data** 또는 **rendered image** 가운데 하나를 선택할 수 있습니다.
- *Format*: GeoTiff, GeoPackage, MBTiles, Geospatial PDF, SAGA GIS Binary Grid, Intergraph Raster, ESRI .hdr Labelled 등등 GDAL 이 작성할 수 있는 모든 래스터 포맷으로 내보낼 수 있습니다.
- *Resolution*: 이미지의 해상도를 설정합니다.
- *Create Options*: 파일 생성 시 (파일 압축, 블록 크기, 색채 분석 colorimetry 등의) 고급 옵션을 사용할 수 있습니다. 산출물 포맷과 관련된미리 정의된 생성 프로파일 을 사용할 수도 있고, 또는 각 파라미터를 설정할 수도 있습니다.
- *Pyramids*: 피라미드 구조를 생성합니다.

- *VRT Tiles*: 사용자가 *Create VRT* 를 선택한 경우 이 옵션을 쓸 수 있습니다.
- *No data values*

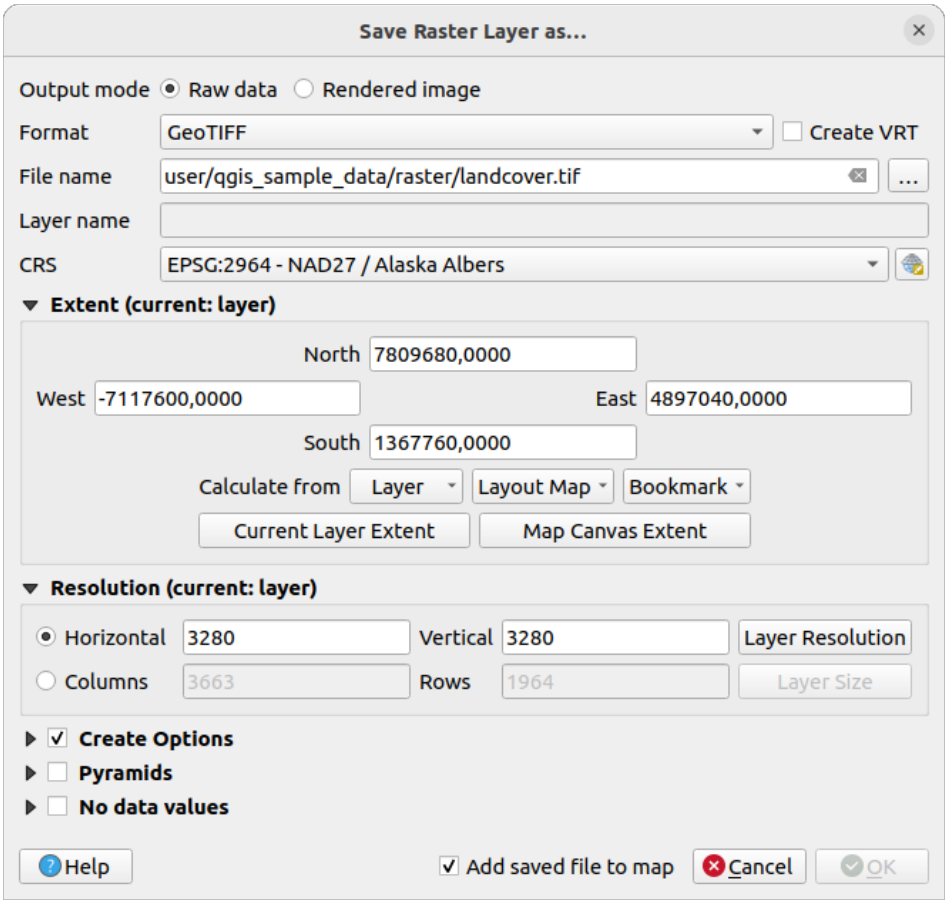



그림 11.31: 새 래스터 레이어로 저장하기

### 벡터 특화 파라미터

내보내기 포맷에 따라, 다음 가운데 일부 옵션을 사용할 수 있습니다:

- *Format*: GeoPackage, GML, ESRI 셰이프파일, AutoCAD DXF, ESRI FileGDB, Mapinfo TAB or MIF, SpatiaLite, CSV, KML, ODS 등등 GDAL 이 작성할 수 있는 모든 벡터 포맷으로 내보낼 수 있습니다.
- *Layer name*: *File name* 이 컨테이너 호환 포맷을 참조하는 경우 사용할 수 있는 이 옵션은 산출물 레이어를 나타냅니다.
- *Encoding*
- *Save only selected features*: 선택한 피처들만 내보냅니다.
- *Select fields to export and their export options*: 필드를 사용자 지정 이름과양식 위젯 설정으로 내보낼 수 있는 수단을 제공합니다:
  - 산출 레이어에 유지해야 할 필드를 선택하려면 *Name* 열 아래의 행을 체크하거나, *Select All* 또는 *Deselect All* 버튼을 누르십시오.

  - *Use aliases for exported name*: 이 옵션을 켜면 *Export name* 열을 대응하는 필드 별명으로 채우고, 끄면 필드의 원래 이름으로 리셋시킵니다. 셀을 더블클릭해서도 이름을 편집할 수 있습니다.
  - *Replace all selected raw field values by displayed values*: 속성 양식 사용자 지정 위젯을 사용 중이냐 아니냐에 따라, 선택한 원시 (raw) 필드 값을 표시된 값으로 치환할 수 있습니다. 예를 들어 필드에 `Value Map` 위젯이 적용된 경우, 산출 레이어는 원본 값 대신 설명 값을 담게 될 것입니다. *Replace with displayed values* 열에서 값을 필드 별로 치환할 수도 있습니다.

- *Persist layer metadata*: 소스 레이어에메타데이터 가 존재하는 모든 레이어에 메타데이터를 복사하고 저장할 것입니다:
  - 산출물이 GeoPackage 포맷인 경우, 새로 생성된 레이어에 메타데이터를 복사하고 저장할 것입니다.
  - 다른 포맷의 경우, 산출 레이어와 함께 메타데이터를 `.qmd` 파일로 저장할 것입니다. 하나 이상의 데이터셋을 지원하는 파일 기반 포맷은 (예: SpatiaLite, DXF 등등) 의도하지 않은 습성을 보일 수도 있다는 사실을 기억하십시오.
- *Symbology export*: 주로 DXF 내보내기에 사용되지만, DXF, KML, TAB 파일 포맷처럼 OGR 피처 스타일을 관리하는 (다음 참고 참조) 모든 파일 포맷으로 내보낼 수 있습니다.
  - **No symbology**: 데이터를 읽는 응용 프로그램의 기본 스타일
  - **Feature symbology**: OGR 피처 스타일로 스타일을 저장 (다음 참고 참조)
  - **Symbol Layer symbology**: OGR 피처 스타일로 저장 (다음 참고 참조) 하지만 심볼 레이어가 반복해서 이용한 심볼이 있을 경우 동일 도형을 반복해서 내보냅니다.
  - **Scale**: 최신 옵션에 척도 값을 적용할 수 있습니다.

> **참고**
>
> *OGR* 피처 스타일 이란 데이터에 스타일을 숨은 속성으로 직접 저장할 수 있는 방법입니다. 몇몇 포맷만이 이런 종류의 정보를 처리할 수 있습니다. KML, DXF 및 TAB 이 바로 그런 파일 포맷입니다. 고급 정보를 원한다면 OGR 피처 스타일 사양 문서를 읽어보면 됩니다.

- *Geometry*: 산출물 레이어의 도형 성능 환경을 설정할 수 있습니다.
  - *geometry type*: **Automatic** 으로 설정한 경우 피처의 원본 도형을 유지하고, 그렇지 않을 경우 제거하거나 아무 유형으로나 대체 (override) 합니다. 속성 테이블에 텅 빈 도형 열을 추가할 수 있고, 공간 레이어에서 도형 열을 제거할 수 있습니다.
  - *Force multi-type*: 레이어에 멀티도형 피처를 강제로 생성합니다.
  - *Include z-dimension*: 도형에 Z 차원을 포함시킵니다.

> **팁**
>
> 레이어 도형 유형을 대체 (override) 하는 옵션은 (`.csv` 파일 같은) 도형이 없는 테이블을 아무 도형 유형 (포인트, 라인, 폴리곤) 이나 가진 셰이프파일로 저장할 수 있기 때문에, Add Part 도구를 통해 행에 도형을 직접 추가할 수 있습니다.

- *Datasource Options*, *Layer Options* 또는 *Custom Options*: 산출물 포맷에 따라 고급 파라미터들을 환경 설정할 수 있습니다. 일부는데이터 포맷 및 필드 탐구 에서 설명하고 있지만, 전체 내용을 알고 싶다면 GDAL 드라이버 문서를 참조하세요. 각 파일 포맷은 고유의 맞춤 파라미터를 보유하고 있습니다. 예를 들어 `GeoJSON` 포맷의 경우, GDAL GeoJSON 문서를 살펴보십시오.

벡터 레이어를 기존 파일로 저장하는 경우, 사용자가 산출물 포맷 (Geopackage, SpatiaLite, FileGDB 등등) 의 성능에 따라 다음을 결정할 수 있습니다:

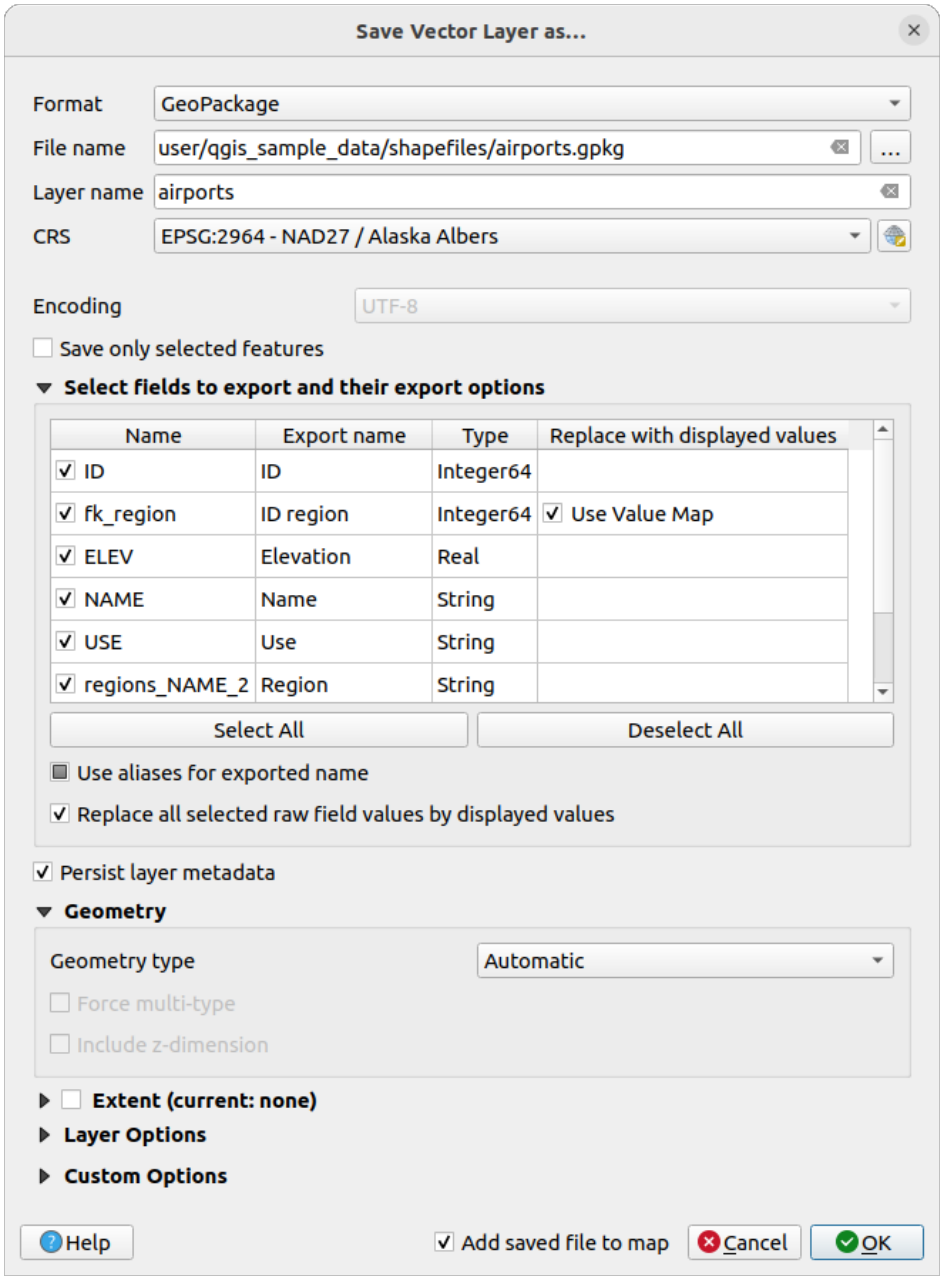



그림 11.32: 새 벡터 레이어로 저장하기

- 파일 전체를 덮어 쓰기
- 대상 레이어만 덮어 쓰기 (레이어명 설정 가능)
- 기존 대상 레이어에 피처 추가
- 새 필드가 있을 경우 필드와 함께 피처 추가

ESRI 셰이프파일, MapInfo .tab 같은 포맷의 경우도 피처를 추가할 수 있습니다.

### Point Cloud specific parameters

Similar to raster and vector layers, point cloud layers can be saved in a different format and/or reprojected to a different coordinate reference system (CRS). This allows you to export a point cloud layer to vector or point cloud formats. Current supported formats are: Temporary scratch (memory layer), GeoPackage, ESRI Shapefile, DXF and LAS/LAZ point cloud. In addition to the common parameters listed above, exporting point cloud layers includes the following options:

- *Filter by Polygon Layer*: Allows you to filter the point cloud data based on a polygon layer.
- *Elevation Range*: Enables filtering of the point cloud data based on a specified Z range.
- *Limit number of points*: Provides an option to limit the number of points exported from the point cloud layer.

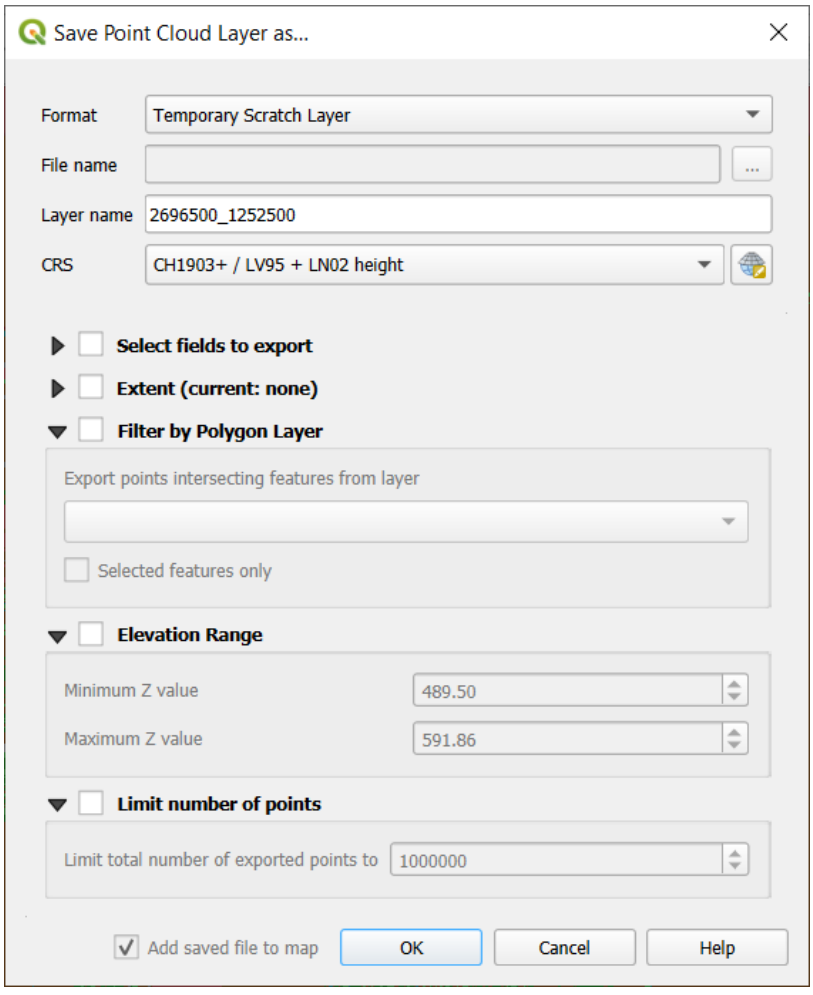



그림 11.33: Saving a point cloud layer as a new layer

### 11.2.3 새 DXF 파일 생성하기

단일 레이어를 `*.DXF` 등의 다른 포맷으로 내보낼 수 있는 옵션을 제공하는 *Save As*…대화창 이외에도, QGIS 는 레이어 여러 개를 단일 DXF 레이어로 내보낼 수 있는 또다른 도구를 제공하고 있습니다. *Project* → *Import/Export* → *Export Project to DXF*…메뉴로 이 도구를 이용할 수 있습니다.

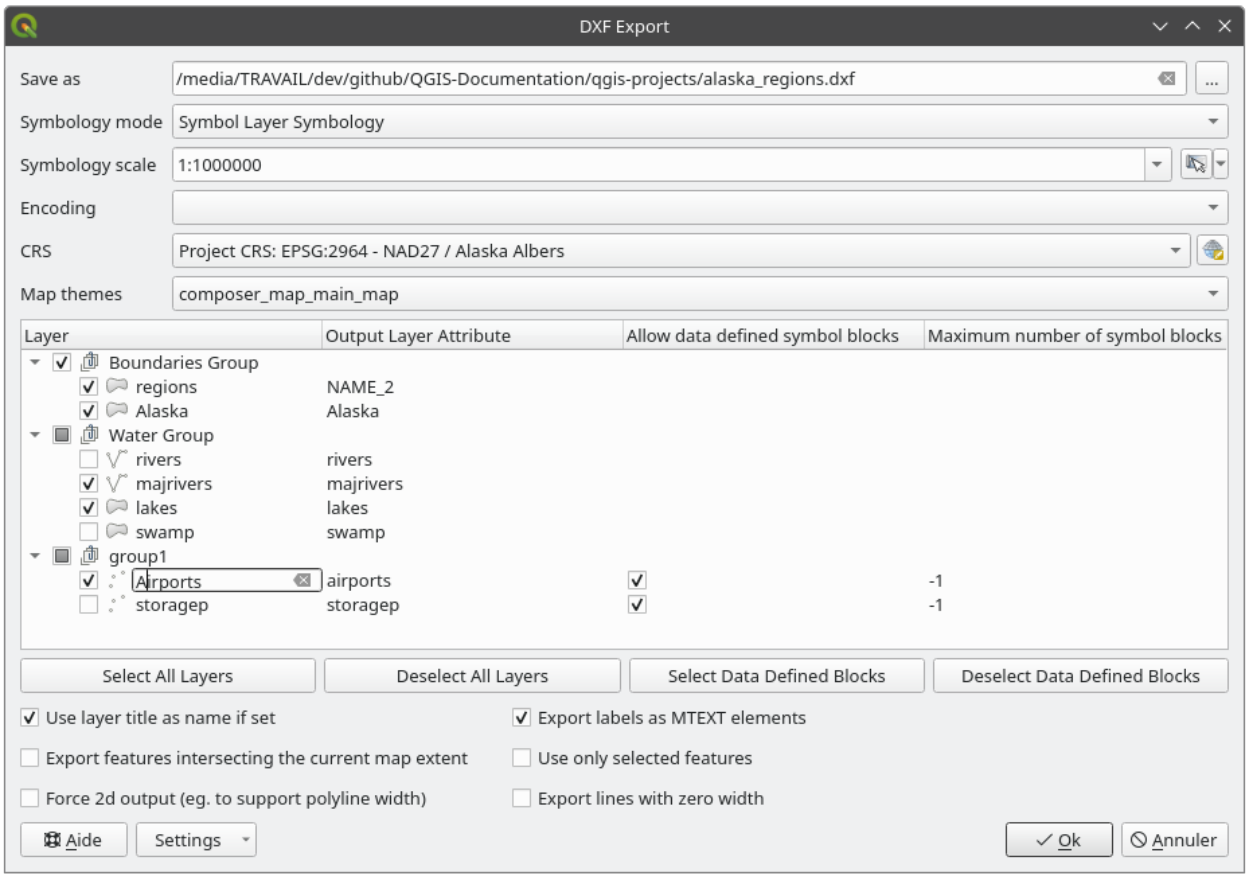



그림 11.34: 프로젝트를 DXF 로 내보내기 대화창

*DXF Export* 대화창에서 사용자는 다음 작업을 할 수 있습니다:

1. 대상 (destination) 파일 지정
2. 가능한 경우 심볼 모드 및 척도 선택 (*OGR* 피처 스타일 참고 참조)
3. *Encoding*: 데이터 인코딩 선택
4. *CRS*: 적용할 좌표계 선택: 선택한 레이어를 지정한 좌표계로 재투영합니다.
5. 테이블 위젯에서 체크하거나 기존맵 테마 에서 자동으로 고르는 방법으로 DXF 파일에 포함시킬 레이어를 선택합니다. *Select All* 및 *Deselect All* 버튼을 사용하면 내보낼 데이터를 신속하게 설정할 수 있습니다.

   For each layer, you can:

   - Override the output layer name without altering the original project layer. For this, click on the *Layer* name in the dialog and write the output name to use.
   - *Output layer attribute*: Choose whether to export all the features in a single DXF layer or rely on a field whose values are used to split the features into layers in the DXF output. In the latter case, each layer will take its name from the corresponding field value.

- *Allow data defined symbol blocks*:
- *Maximum number of symbol blocks*: creates symbol blocks up to the specified limit, starting with the ones containing the highest number of references. The other symbols are written as they are. `-1` means no limitation.

다음 체크박스들도 선택적으로 사용할 수 있습니다:

- ☐ *Use the layer title as name if set* instead of the layer name itself: the title is taken from the *metadata* or *server* properties of the layer;
- ☐ *Export features intersecting the current map extent*;
- ☐ *Force 2d output (eg. to support polyline width)*: 이 옵션을 활성화하면 산출물을 2 차원으로 강제합니다.
- ☑ *Export label as MTEXT elements* or TEXT elements;
- ☐ *Use only selected features*;
- ☐ *Export lines with zero width*: all the lines are exported with minimal width `0` (hairline) if enabled. This helps keep the lines minimal in the file regardless of the zoom level, and can be handy for doing further CAD-editing with the exported dxf, especially if there are many features next to each other on the map.

> **참고**
>
> The precedence for defining the output layer name is as follows:
>
> 1. The field value from *Output layer attribute*
> 2. The overridden name in *Layer* column
> 3. The *Use the layer title as name if set* option
> 4. The layer name

Current settings defined in the *DXF Export* dialog may be stored in an XML file for reusing them in other sessions. For this, the *Settings* combo box has two options: *Load Settings from File…* and *Save Settings to File…*.

### 11.2.4 클립보드로부터 새 레이어 생성

클립보드 상에 있는 피처를 새 레이어로 붙여넣을 수 있습니다. 이 작업을 하려면 피처를 몇 개 선택해서 클립보드로 복사한 다음, *Edit* ▸ *Paste Features as* ▸ 메뉴의 다음 메뉴 옵션 가운데 하나를 선택해서 새 레이어에 붙여넣으십시오:

- *New Vector Layer…*: *Save vector layer as…*대화창이 열립니다. (파라미터에 대한 설명은기존 레이어로부터 레이어 생성 참조)
- *Temporary Scratch Layer…*: 레이어 이름을 입력해야 합니다.

선택한 피처 및 그 속성을 가진 새 레이어를 생성합니다. (그리고 맵 캔버스에 추가합니다.)

> **참고**
>
> QGIS 안에서 선택하고 복사한 피처는 물론, 다른 응용 프로그램에서 — WKT(Well-Known Text) 를 이용해서 정의된 도형이라면 — 도형을 복사해도 클립보드로부터 레이어를 생성할 수 있습니다.

### 11.2.5 Creating SQL Query Layers

Beside loading an entire layer in a project or creating new layers from scratch or pasted features, you can also load layers generated on the fly from other layer(s). They are the result of a more or less advanced filter using SQL language, applied to stored layers regardless of their data provider or their availability in the active project. This means that temporary layers are not compatible with this feature. Depending on the provider, one or more layers can be used to write the query. The generated layer remains dependent on the layer(s) involved in the query and is loaded with the Filter icon next to it.

This feature is accessible:

- From the *Browser* panel, right-click on a supported data (plain layer, database connection, schema or table) and select *Execute SQL…* entry in the contextual menu.
- From the *Layers* panel, select a loaded layer, right-click and select *Execute SQL…* entry in the contextual menu.

This opens a window with a central text box widget where you can write SQL queries.

At the top of the dialog, a toolbar provides a set of tools to create, store and manipulate your queries:

- Open Queries…: fills the text editor widget with contents from an existing `.sql` file
- Save Queries… and Save Queries as… help you store the written query to a `.sql` file
- Statements of the query can be adjusted using the Cut, Copy and Paste buttons. Likewise, you can Undo or Redo your changes.
- The Find & Replace enables, at the bottom of the dialog, a widget allowing to look for a particular string in your SQL code. The search can be case sensitive, affect partial or whole word, rely on a regular expression. It is then possible to navigate through the found strings, replacing them one by one or all in a row.
- Use Clear SQL Editor to wipe the text editor.
- The History button opens a dialog storing previously run queries. More at *Query History*.
- As previously mentioned, queries can be saved as an `.sql` file stored on disk. Using the Store Current Query button, they can also be stored:
  - In the active *User Profile*, in the associated `QGIS3.ini` file, thus accessible in subsequent projects
  - or as part of the *Current Project*.

  Clicking an entry from the stored queries drop-down menu inserts that query in the expression being written. The stored entry can also be deleted, from the drop-down menu.

#### Running and loading queries as layer

In the central part of the *Execute SQL* dialog, you build your query using the SQL syntax supported by the underlying provider (e.g., OGR, GeoPackage, PostgreSQL).

By default, if opened from a layer entry, a sample SQL query is provided. Editing tools to select, cut, copy, paste, undo and redo are available as well from the contextual menu.

> **팁**
>
> **Finding the right SQL syntax for your dataset**

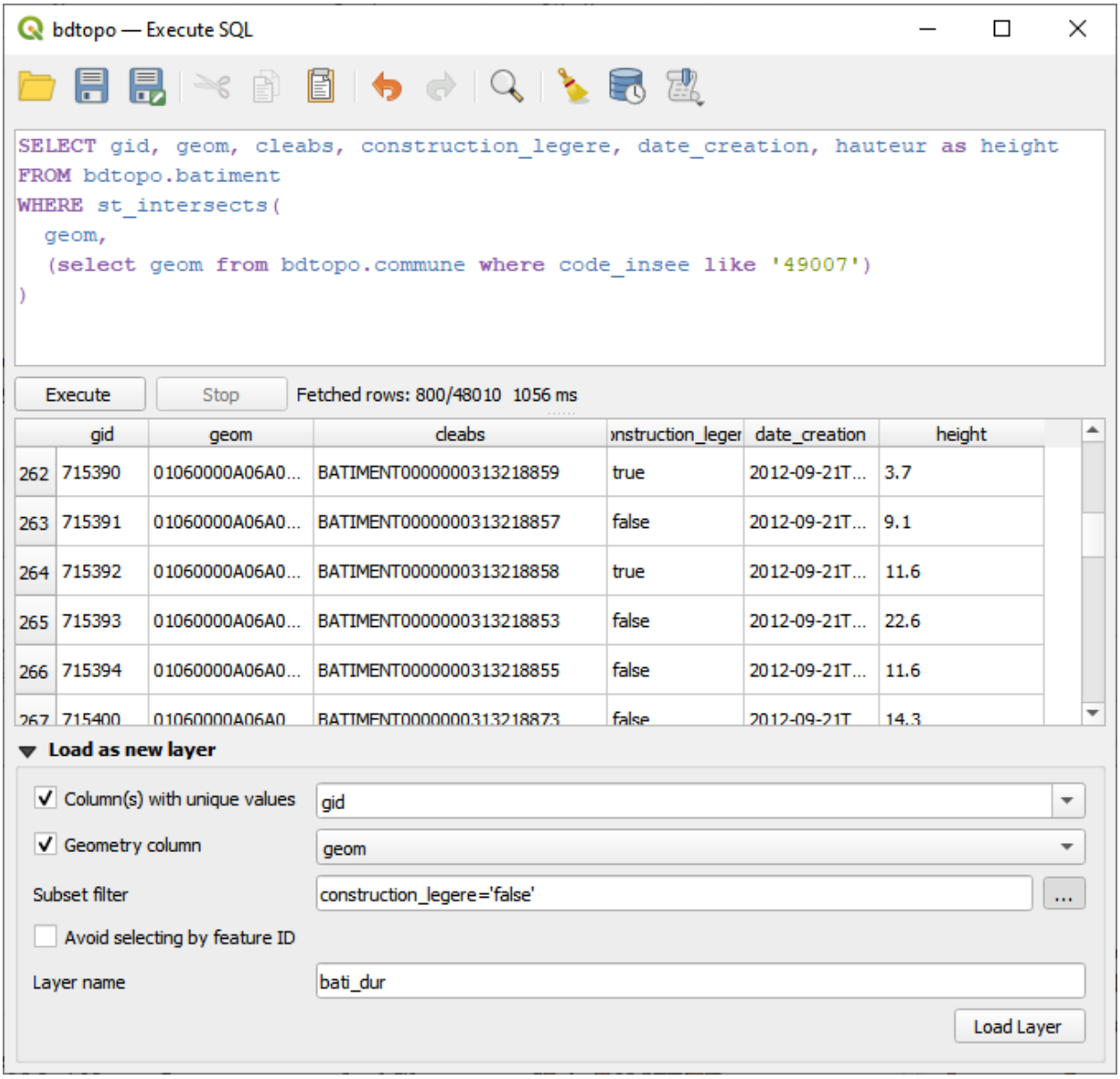



그림 11.35: Executing SQL queries in the Execute SQL window

To find the appropriate SQL dialect for your dataset, press the ⋯ button next to the *Subset filter* option in the bottom part of the dialog.

When ready, pressing the *Execute* button below the text area will run the query. It is possible to highlight a portion of the SQL to only execute that portion when pressing `Ctrl+R` or clicking the *Execute selection* button. Use the *Stop* button to abort the execution.

A successful query execution will display a table at the bottom of the dialog with returned features. You can select specific cells in the result set. Use the `Ctrl+C` shortcut to copy the selected cells to the clipboard. The copied data is available as a formatted table. This allows you to paste the data into other applications, such as spreadsheet where it will show up as a table.

The returned table can be loaded in QGIS expanding the *Load as new layer* group and configuring parameters (their availability depends on the layer provider):

- *Column(s) with unique values* to indicate primary key of the data,
- *Geometry column*: check the box to load the layer as a spatial one, and indicate the geometry field name
- *Subset filter*: allows to filter the results using a `WHERE` clause. It can be written in the text box, or built using the *Query builder* after you pressed ⋯. Make sure to use fields that are available in the SQL layer.
- *Avoid selecting by feature ID*
- *Layer name* in the project

At any time, in the *Layers panel*, you can adjust the output layer by right-clicking and selecting *Update SQL expression*⋯ . The *Update SQL* dialog opens, prefilled with the applied query that you can edit as you wish. Once ready, press *Update layer* and the layer will be modified in-place.

### Query History

The *Query History* dialog displays all the previously run queries, sorted by date and data provider type. It is accessible from the *Database*  *Query History*⋯ menu, or pressing the  History button from the *Execute SQL* dialog.

At the top of the dialog, a search box can be used to filter out the queries. The search is done through every strings displayed in the query history. If the dialog is opened from within the *Execute SQL* dialog, the search box is preceded by the *Execute SQL toolbar*.

Below, collapsible and sorted date entries list their related queries. Each query is identifiable by its SQL command, and when expanded, reveals:

- The *Connection*: the source of the data the query is based on
- The *Row count*: number of returned features
- The *Execution time*

Hover over an entry and a tooltip displays the full query. Likewise, selecting an entry also displayed it in the bottom part of the dialog. You can interact with the text, copying all or part of it.

Double-click the SQL command entry and the *Execute SQL* dialog opens, filled with the clicked query. Any query being written is overwritten. Right-click the SQL command entry and you can either:

- *Load SQL Command*⋯, loads the target command into the *Execute SQL* dialog, replacing any existing query. It is the same as double clicking the entry.
- *Copy SQL Command* and paste it wherever you want.

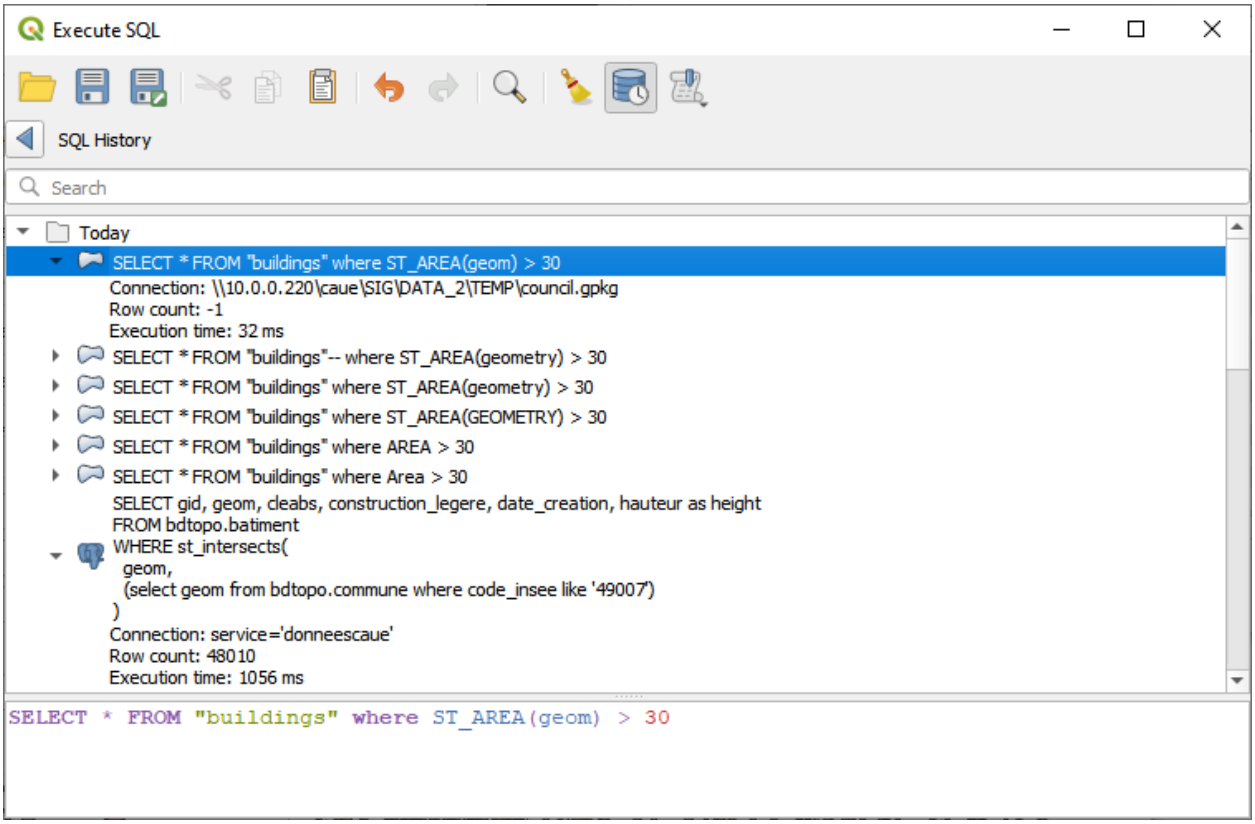



그림 11.36: History of the executed SQL queries

### 11.2.6 가상 레이어 생성

가상 레이어는 특별한 종류의 벡터 레이어입니다. QGIS 가 지원하는 여러 다른 벡터 레이어 포맷을 대상으로 한 SQL 쿼리의 산출물로 레이어를 정의할 수 있게 해줍니다. 가상 레이어는 자체적으로 데이터를 보유하지 않으며, 뷰로 표시할 수 있습니다.

가상 레이어를 생성하려면, 다음 방법으로 가상 레이어 생성 대화창을 여십시오:

- *Layer* → *Add Layer* → 메뉴에서 *Add/Edit Virtual Layer* 항목을 선택하거나
- *Data Source Manager* 대화창에서 *Add Virtual Layer* 탭을 선택하거나
- *DB Manager* 대화창의 트리를 사용합니다.

대화창에서 *Layer name* 및 SQL *Query* 를 지정할 수 있습니다. 쿼리는 불러온 벡터 레이어의 이름 (또는 ID) 을 테이블처럼 이용할 수 있으며, 테이블의 필드명을 열처럼 사용할 수도 있습니다.

예를 들면, `airports` 라는 레이어가 있을 경우 다음과 같은 SQL 쿼리를 통해 `public_airports` 라는 가상 레이어를 생성할 수 있습니다:

```
SELECT *
FROM airports
WHERE USE = "Civilian/Public"
```

`airports` 레이어의 기저 제공자가 어떤 것이든, 심지어 해당 제공자가 SQL 쿼리를 직접 지원하지 않는다고 해도, SQL 쿼리는 실행될 것입니다.

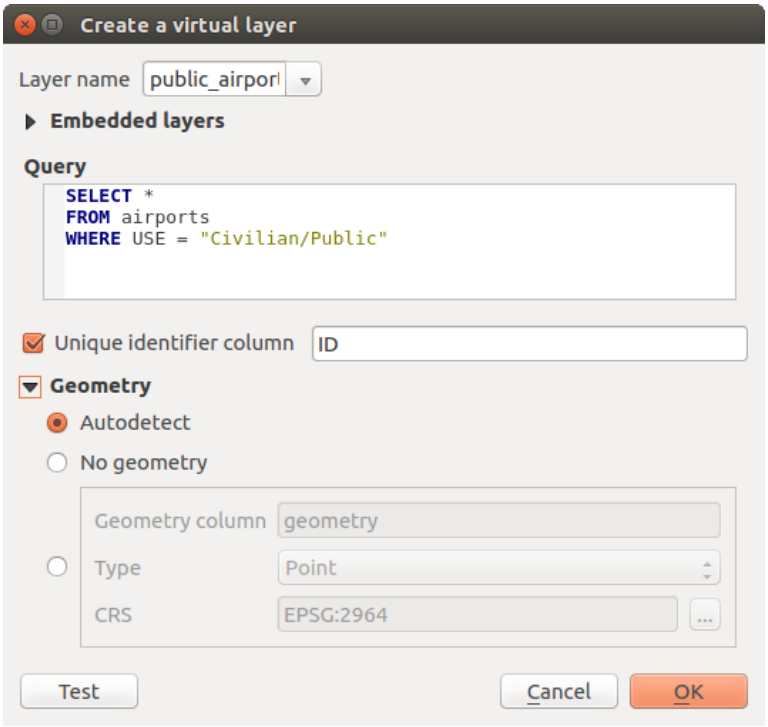



그림 11.37: 가상 레이어 생성 대화창

예를 들어 공항과 국가 정보를 결합시키려면, 다음과 같이 결합 (join) 및 복잡 쿼리를 생성할 수도 있습니다:

```
SELECT airports.*, country.population
FROM airports
JOIN country
ON airports.country = country.name
```

> **참고**
>
> 데이터베이스 관리자 플러그인 의 SQL 창을 이용해서 가상 레이어를 생성할 수도 있습니다.

### 쿼리에 사용하기 위해 레이어를 삽입하기

맵 캔버스에서 작업할 수 있는 벡터 레이어 이외에도, 사용자가 *Embedded layers* 목록에 레이어를 추가할 수 있습니다. 이렇게 추가한 레이어는 맵 캔버스 또는 레이어 패널에 표시될 필요 없이 사용자가 쿼리에 이용할 수 있습니다.

레이어를 삽입하려면, *Add* 를 클릭한 다음 *Local name*, *Provider*, *Encoding* 그리고 *Source* 를 가리키는 경로를 입력하십시오.

*Import* 버튼을 누르면 맵 캔버스에 불러온 레이어를 삽입 레이어 목록에 추가할 수 있습니다. 이렇게 하면 향후 기존 쿼리를 고장내지 않고 레이어 패널에서 해당 레이어를 제거할 수 있습니다.

### 지원하는 쿼리 언어

기저 엔진은 작업 수행에 SQLite 및 SpatiaLite 를 이용합니다.

즉 사용자가 로컬 설치한 SQLite 가 지원하는 모든 SQL 을 이용할 수 있다는 뜻입니다.

가상 레이어 쿼리에 SQLite 의 기능 및 SpatiaLite 의 공간 기능을 이용할 수도 있습니다. 예를 들어, 다음과 같은 쿼리를 통해 속성만 가지고 있는 레이어로부터 포인트 레이어를 생성할 수 있습니다:

```
SELECT id, MakePoint(x, y, 4326) as geometry
FROM coordinates
```

가상 레이어 쿼리에*QGIS 표현식 함수* 를 이용할 수도 있습니다.

레이어의 도형 열을 참조하려면, `geometry` 이름을 이용하십시오.

순수한 SQL 쿼리와는 반대로, 가상 레이어 쿼리의 모든 필드는 이름을 가지고 있어야만 합니다. 사용자가 이용하는 열이 계산식 또는 함수 호출의 산출물인경우 사용자 열을 명명하는 데 `as` 키워드를 사용해야 한다는 사실을 기억하십시오.

### 성능 문제

기본 파라미터를 설정하고 나면, 가상 레이어 엔진은 최선을 다해 쿼리의 서로 다른 열들의 유형을 — 도형 열이 존재할 경우 도형 열의 유형도 포함해서 — 탐지하려 할 것입니다.

이 탐지 작업은 가능한 경우 쿼리를 내부 검사하거나 또는 불가능할 경우 마지막 수단으로 쿼리의 첫 행 (LIMIT 1) 을 불러와서 이루어집니다. 레이어를 생성하기 위해서만 쿼리 결과의 첫 행을 불러오는 것은 성능 관련 이유 때문에 바람직하지 않을 수도 있습니다.

가상 레이어 생성 대화창의 파라미터는 다음과 같습니다:

- *Unique identifier column*: 이 옵션으로 QGIS 가 행 식별자로 이용할 수 있는 유일 정수 값들을 나타내는 쿼리 필드를 지정할 수 있습니다. 기본적으로, 자동 증가하는 정수 값이 쓰입니다. 유일 식별자 열을 정의하면 ID 로 행을 선택하는 작업 속도를 향상시킬 수 있습니다.

- *No geometry*: 이 옵션으로 가상 레이어가 모든 도형 필드를 강제로 무시하도록 만들 수 있습니다. 속성만 가지고 있는 레이어를 산출합니다.
- 도형 *Column*: 이 옵션으로 레이어의 도형으로 쓰일 열의 이름을 지정할 수 있습니다.
- 도형 *Type*: 이 옵션으로 가상 레이어의 도형 유형을 지정할 수 있습니다.
- 도형 *CRS*: 이 옵션으로 가상 레이어의 좌표계를 지정할 수 있습니다.

### 특별 주석

가상 레이어 엔진은 쿼리의 각 열의 유형을 결정하려 합니다. 실패할 경우, 열 유형을 결정하기 위해 쿼리의 첫 행을 불러옵니다.

몇몇 특별 주석을 이용하면 쿼리 안에서 특정 열의 유형을 직접 지정할 수 있습니다.

특별 주석의 문법은 `/*:type*/` 입니다. 열 이름의 바로 뒤에 이 주석을 달아야 합니다. `type` 자리에 정수의 경우 `int`, 부동소수점 실수의 경우 `real`, 또는 문자열의 경우 `text` 를 넣을 수 있습니다.

다음은 예시입니다:

```
SELECT id+1 as nid /*:int*/
FROM table
```

도형 열의 유형 및 좌표계도 특별 주석으로 설정할 수 있습니다. 문법은 `/*:gtype:srid*/` 입니다. 이때 `gtype` 자리에 도형 유형 (`point`, `linestring`, `polygon`, `multipoint`, `multilinestring` 또는 `multipolygon`) 을 넣을 수 있으며, `srid` 자리엔 좌표계의 EPSG 코드를 나타내는 정수값을 넣을 수 있습니다.

### 인덱스 사용

가상 레이어를 통해 레이어를 호출하는 경우, 소스 레이어의 인덱스를 다음 방식으로 이용할 것입니다:

- 레이어의 기본 키 열에 대해 = 서술자 (predicate) 가 쓰인 경우, 기저 데이터 제공자에게 해당 특정 ID(FilterFid) 를 물을 것입니다.
- 다른 서술자 (>, <=, != 등등) 가 쓰이거나 기본 키가 없는 열에 대해 쓰인 경우, 기저 벡터 데이터 제공자를 요청하는 데 표현식에서 생성된 요청을 이용할 것입니다. 즉 인덱스가 존재하는 경우 데이터베이스 제공자에 대해 인덱스를 이용할 수도 있다는 뜻입니다.

요청 내부의 공간 서술자를 처리하고 공간 인덱스 이용을 요청하는 데 특화된 문법이 존재합니다. 각 가상 레이어에 대해 `_search_frame_` 이라는 숨은 열이 존재합니다. 이 열과 경계 상자와의 동일성을 비교할 수 있습니다. 다음은 그 예시입니다:

```
SELECT *
FROM vtab
WHERE _search_frame_=BuildMbr(-2.10,49.38,-1.3,49.99,4326)
```

이 공간 인덱스 문법과 함께 사용하는 경우, `ST_Intersects` 같은 공간 2 진 (binary) 서술자의 실행 속도가 현저히 빨라집니다.

## 11.3 지리참조자

The Georeferencer is a tool for aligning unreferenced raster or vector layers to known coordinate systems using Ground Control Points (GCPs). It supports exporting transformed rasters (GeoTIFF) and vectors (shapefiles, geopackages, etc.) or writing accompanying world files (for raster only). The basic approach to georeferencing a layer is to locate points on it for which you can accurately determine coordinates.

주요 기능

표 11.1: Georeferencer Tools

| 아이콘 | 목적 | 아이콘 | 목적 |
|---|---|---|---|
| | 래스터 열기 | | 벡터 열기 |
| | 지리참조 작업 시작 | | GDAL 스크립트 생성 |
| | GCP 포인트 불러오기 | | 다른 이름으로 GCP 포인트 저장 |
| | 변환 설정 | | |
| | Add GCP Point | | Delete GCP Point |
| | GCP 포인트 이동 | | Snapping Type |
| | Show Advanced Digitizing Dock | | 이동 |
| | 확대 | | 축소 |
| | 레이어로 확대/축소 | | 이전으로 확대/축소 |
| | 다음으로 확대/축소 | | QGIS 에 지리참조자 링크 |
| | 지리참조자에 QGIS 링크 | | 히스토그램 전체 균일화 |
| | 히스토그램 부분 균일화 | | |

### 11.3.1 Georeferencing raster layer

이미지 상에서 선택한 포인트와 대응하는 X 및 Y 좌표 (도분초 (dd mm ss.ss), 십진도 (dd.dd), 또는 투영 좌표 (mmmm.mm)) 에 대해서는 다음 두 가지 대안적인 절차를 사용할 수 있습니다:

- 래스터 자체 이미지 상에 좌표가 표시된 십자표가"작성돼 있는"경우가 있습니다. 이 경우 사용자가 좌표를 직접 입력할 수 있습니다.
- 이미 지리참조된 레이어를 활용하는 방법입니다. 이 레이어는 사용자가 지리참조시키려는 이미지 상에 가지고 있는 객체/피처와 동일한 객체/피처를 담고 있으며, 사용자가 이미지에 적용하려는 투영체를 사용하는 벡터 또는 래스터 데이터일 수 있습니다. 이런 경우, QGIS 맵 캔버스에 불러온 참조 데이터셋을 클릭해서 좌표를 입력할 수 있습니다.

이미지를 지리참조시키기 위한 일반 절차는 래스터 상에서 여러 포인트를 선택하고, 포인트 좌표를 지정한 다음, 적합한 변형 (transformation) 유형을 선택하는 것입니다. 지리참조자는 입력 파라미터 및 데이터를 기반으로 월드 파일 파라미터를 계산할 것입니다. 사용자가 좌표를 더 많이 제공할수록, 더 나은 결과를 보게 될 것입니다.

첫 번째 단계는 QGIS 를 구동하고 QGIS 메뉴 바에 나타나는 *Layer* *Georeferencer* 메뉴 옵션을 선택하는 것입니다. 그림 11.38 와 같은 지리참조자 플러그인 대화창이 열릴 것입니다.

이 예시를 위해 SDGS(South Dakota Geological Survey) 가 제공하는 사우스다코타 주의 지형도 (topo sheet) 를 사용할 것입니다. 이 지형도는 나중에 GRASS `spearfish60` 위치 데이터와 함께 가시화시킬 수 있습니다. https://grass.osgeo.org/sampledata/spearfish_toposheet.tar.gz 에서 해당 지형도를 다운로드할 수 있습니다.

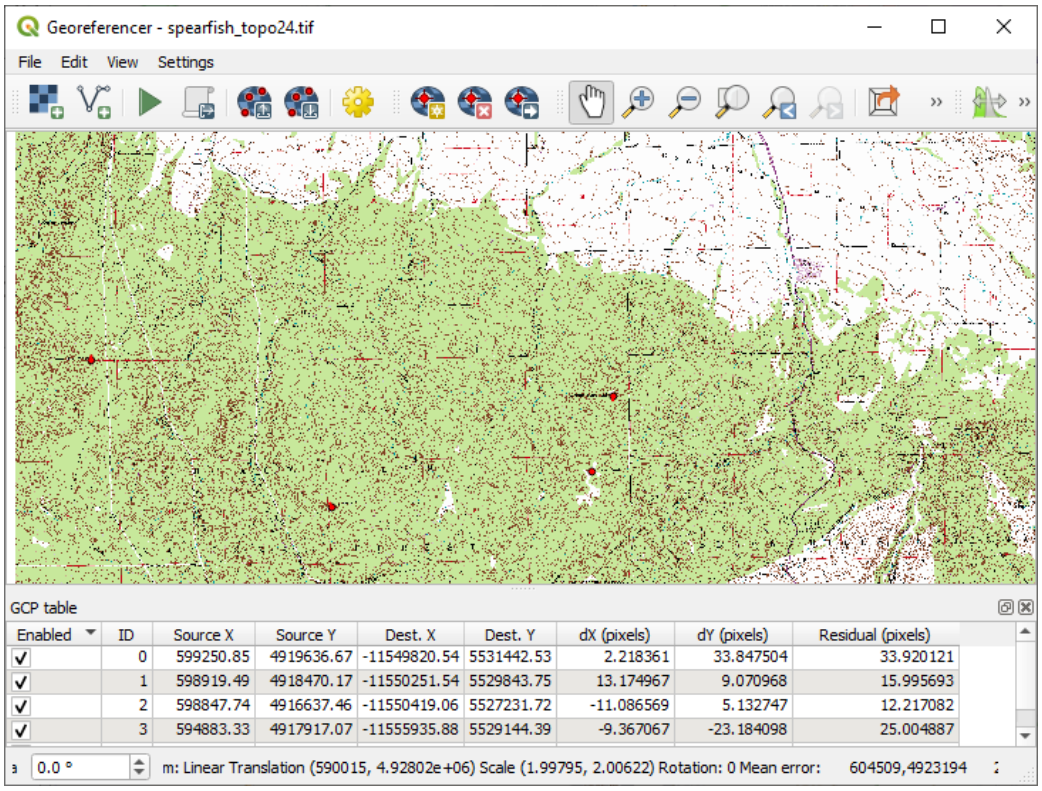



그림 11.38: 지리참조자 대화창

### 지상기준점 (GCP) 입력

1. 참조되지 않은 래스터를 지리참조시키려면, 먼저  버튼을 사용해서 해당 래스터를 불러와야만 합니다. 이 래스터는 대화창의 주 작업 영역에 나타날 것입니다. 래스터를 불러오고 나면, 참조 포인트를 입력할 수 있습니다.

2. Using the  Add GCP Point button, add points to the main working area and enter their coordinates (see Figure 그림 11.39). For this procedure you have the following options:

   - 래스터 이미지에 포인트를 클릭한 다음 X 및 Y 좌표를 포인트의 좌표계와 함께 직접 입력합니다.
   - 래스터 이미지에 포인트를 클릭한 다음  From map canvas 버튼을 눌러 QGIS 맵 캔버스에 이미 불러온 지리참조 맵의 도움을 받아 X 및 Y 좌표를 추가합니다. 좌표계는 자동적으로 설정될 것입니다.
   - 주 맵 캔버스에서 GCP 를 입력할 때, 주 캔버스에서 포인트를 선택하는 동안 지리참조자 창을 숨길 수 있는 옵션이 있습니다.  *From Map Canvas* 도구를 클릭한 다음  *Automatically hide georeferencer window* 옵션을 체크하면, 맵 캔버스 상에 포인트를 추가할 때까지 지리참조자 창을 숨길 것입니다. *Enter Map Coordinates* 대화창은 열린 채로 유지됩니다. 이 옵션을 체크 해제하면, 맵 캔버스 상에서 포인트를 선택하는 동안 두 창 모두 열려 있을 것입니다. 이 옵션은 지리참조자 창이 메인 인터페이스에 도킹되어 있는 경우 아무 영향도 미치지 않습니다.

3. 포인트 입력을 계속하십시오. 최소한 포인트 4 개를 입력해야 하며, 더 많은 좌표를 지정할수록 결과물이 더 나아질 것입니다. 관련 GCP 집합을 찾기 위해 작업 영역을 확대/축소하고 이동하기 위한 추가 도구들이 있습니다.

> **팁**
>
> To avoid constant switching between Pan, Add GCP point and Move GCP point buttons, you may use the keyboard arrow keys for moving and the mouse wheel for scaling the georeferenced map conveniently.

4. After you provide a few points, you can use the Link QGIS to Georeferencer and/or Link Georeferencer to QGIS buttons that will adjust, respectively, the map extent of the main QGIS window to the present view in Georeferencer and/or vice versa.

5. GCP 를 수정해야 하는 경우, 맵 캔버스와 지리참조 작업 창 둘 다에서 도구로 GCP 를 이동시킬 수 있습니다.

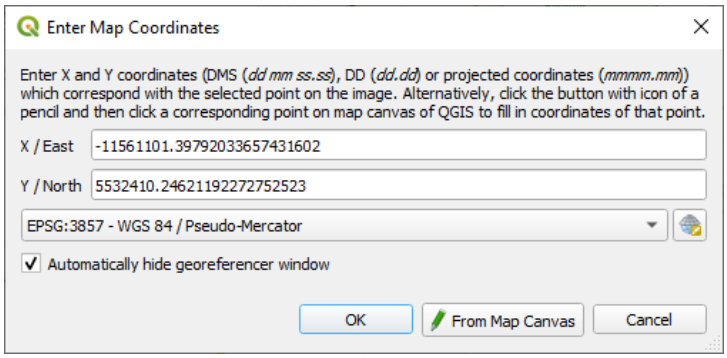



그림 11.39: 래스터 이미지에 포인트 추가

맵에 추가된 포인트들은 일반적으로 래스터 파일과 동일한 위치에 개별 텍스트 파일 (`[filename].points`)로 저장됩니다. 이 습성은 향후 지리참조자를 다시 열어 결과물을 최적화하기 위해 새 포인트를 추가하거나 기존 포인트를 삭제할 수 있게 해줍니다. 포인트 파일은 `mapX, mapY, pixelX, pixelY` 라는 서식으로 값을 담고 있습니다. Load GCP points 및 Save GCP points as 버튼을 통해 포인트 파일을 관리할 수 있습니다.

### 변형 설정

래스터 이미지에 사용자 GCP 를 추가했다면, 지리참조 과정을 위해 변형 (transformation) 설정을 정의해야 합니다.

### 사용 가능한 변형 알고리즘

사용자가 GCP 를 얼마나 많이 잡았는지에 따라, 입력 데이터의 유형 및 품질에 따라, 그리고 최종 산출물에 어느 정도의 기하학적 왜곡을 주어도 좋은지에 따라, 많은 변형 (transformation) 알고리즘을 사용할 수 있습니다.

현재, 다음 *Transformation type* 들을 사용할 수 있습니다:

- **선형 (Linear)** 알고리즘을 사용해서 월드 파일을 생성하는데, 다른 알고리즘과 달리 래스터 픽셀을 실제로 변형하지는 않습니다. 이 알고리즘으로 이미지 배치 작업 (변형 작업) 과 일괄 (uniform) 축척 작업을 할 수 있지만, 회전 또는 다른 변형을 할 수는 없습니다. 사용자의 이미지가 훌륭한 품질의 래스터 맵이고 알려진

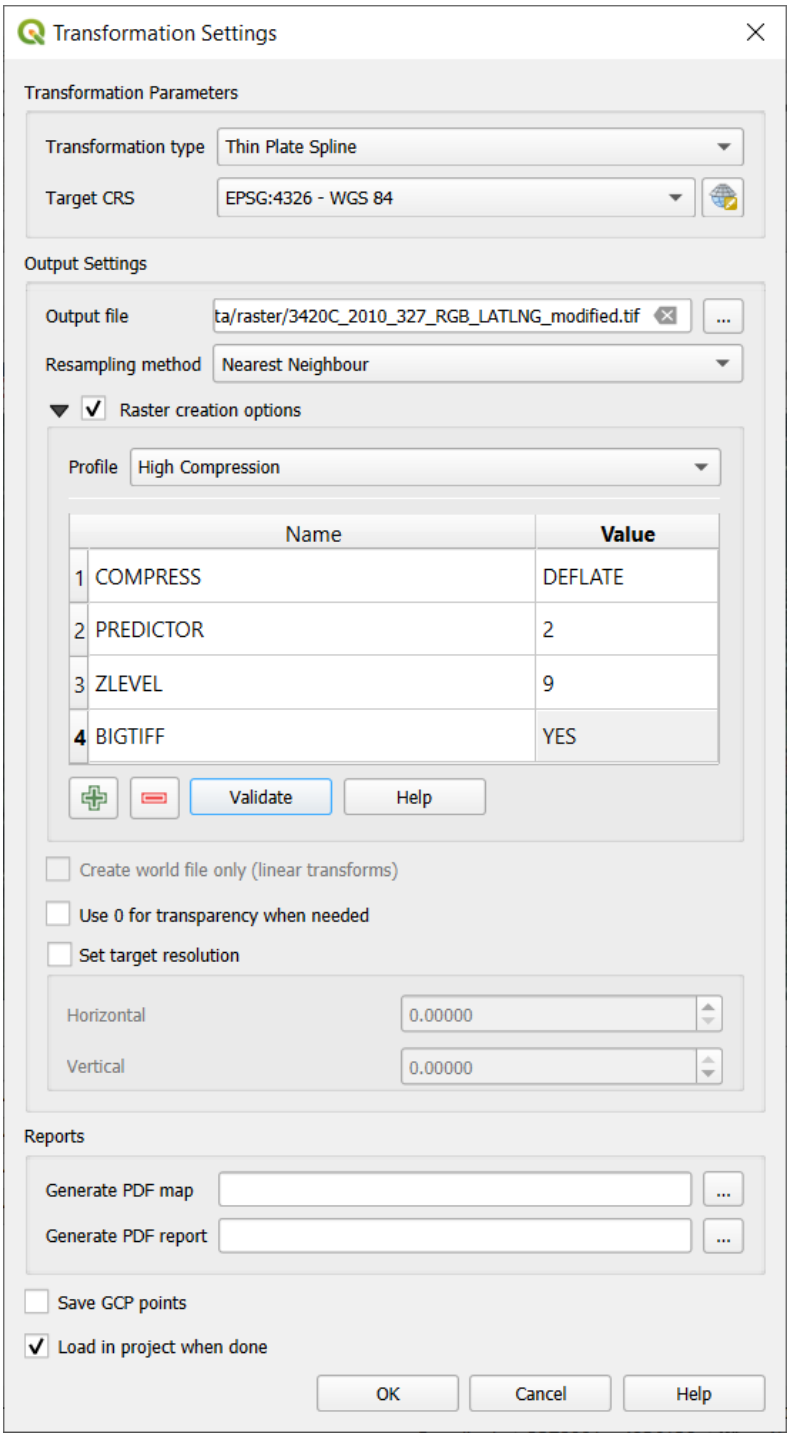



그림 11.40: 지리참조자 변형 설정 정의하기

좌표계를 쓰고 있지만, 지리참조 정보만 누락돼 있는 경우 가장 적합한 알고리즘입니다. 적어도 2 개 이상의 GCP 가 필요합니다.

- **헬메르트 (Helmert)** 변형으로도 회전이 가능합니다. 사용자의 래스터가 훌륭한 품질의 지역 맵이거나 정사보정 (orthorectified) 항공사진인 경우 특히 유용하지만, 사용자 좌표계에서 그리드 방향으로 정렬되지는 않습니다. 적어도 2 개 이상의 GCP 가 필요합니다.
- **다항식 1(Polynomial 1)** 알고리즘은 보다 일반적인 아핀 변환 (affine transformation) 을, 그 중에서도 특히 일괄 자르기 (uniform shear) 도 할 수 있습니다. 직선은 직선으로 (예: 동일 선상의 포인트는 동일 선상에) 남고, 평행한 라인들은 평행하게 남습니다. 이 알고리즘은 서로 다른 방향에서 서로 다른 지상 픽셀 크기로 도표화 (또는 데이터 수집) 됐을 수도 있는 지리참조 데이터 통계지도 (cartogram) 에 특히 유용합니다. 적어도 3 개 이상의 GCP 가 필요합니다.
- **다항식 2(Polynomial 2)**, **다항식 3(Polynomial 3)** 알고리즘은 아핀 변환만 사용하는 대신 더 일반적인 2 또는 3 단계 다항 방정식을 사용합니다. 즉 곡률 또는 다른 체계적인 이미지 왜곡을, 예를 들면 만곡 경계를 가진 항공사진 맵을 처리할 수 있습니다. 적어도 6 개의 (10 개 권장) GCP 가 필요합니다. 이미지 전체에 걸쳐 각도 및 지역 축척을 유지하거나 일괄 처리하지 않습니다. 특히 직선이 곡선이 될 수도 있고, 경계 부근 또는 GCP 에서 멀리 떨어진 위치에서 다항식을 데이터에 맞게 추정하느라 상당한 왜곡이 일어날 수도 있습니다.
- **투영 (Projective)** 알고리즘은 다항식 1 알고리즘을 다른 방향으로 일반화해서, 서로 평행하지 않은 2 개의 평면, 즉 이미지와 맵 캔버스 사이의 중심투영 (central projection) 을 표현하는 변형을 할 수 있습니다. 직선은 직선으로 남지만, 평행 상태는 보전되지 않으며 이미지 전체에 걸쳐 원근법에 따라 축척이 일관되게 변화합니다. 이 변형 유형은 훌륭한 품질의 맵을 (평면 스캔이 아니라) 비스듬하게 찍은 사진 또는 비스듬하게 찍은 항공사진을 지리참조하는 데 가장 유용합니다. 적어도 4 개의 GCP 가 필요합니다.
- 마지막으로, **박막 스플라인 (Thin Plate Spline; TPS)** 알고리즘은 지역적인 다항식 여러 개를 사용해서 지정한 GCP 에 일치하도록 래스터를"고무판 휘듯이"전체 지표 곡률을 최소화한 채로 변형할 수 있습니다. GCP 에서 멀리 떨어진 영역은 GCP 와 일치하도록 움직여서 산출될 것이지만, 그렇지 않은 경우 지역적인 왜곡은 최소화될 것입니다. TPS 는 망가지거나 왜곡된, 아니면 살짝 부정확한 맵 또는 서툴게 정사보정된 항공사진을 지리참조하는 데 가장 유용합니다. 또 알 수 없는 투영 유형 또는 파라미터를 가지고 있지만 정규 그리드 또는 조밀한 임시 (ad-hoc) GCP 집합을 참조 맵 레이어와 매칭할 수 있는 맵을 대략적으로 지리참조해서 그대로 재투영하는 데에도 유용합니다. 이 알고리즘은 기술적으로 최소한 10 개 이상의 GCP 를 필요로 하지만, 성공적인 실행을 위해서는 보통 더 많이 필요합니다.

TPS 를 제외한 모든 알고리즘에서 최소 필요 개수보다 더 많은 GCP 를 지정한 경우, 전체적인 잔여물 (residual) 오류가 최소화되도록 파라미터들이 맞춰질 것입니다. 이렇게 하면 포인트 위치 클릭 또는 좌표 입력 또는 다른 작은 지역 이미지 왜곡 등에서 일어나는 경미한 부정확함 등의 등록 오차 (registration error) 의 영향을 최소화하는 데 도움이 됩니다. 이런 부정확함을 상쇄해줄 다른 GCP 가 없다면, 경미한 오류 또는 왜곡이, 특히 지리참조된 이미지의 경계 부분에서는, 상당히 왜곡되어 나타날 수도 있습니다. 하지만 필요한 최소 GCP 보다 더 많이 지정하면, 산출물에서 대략적이나마 매칭될 것입니다. 이에 반해 TPS 알고리즘은 지정한 모든 GCP 를 정확하게 매칭하지만, 등록 오차로 인한 상당한 왜곡이 서로 가까이 있는 GCP 들 사이에서 나타날 수도 있습니다.

#### 리샘플링 방법 정의

입력 데이터 및 작업의 궁극적인 목표에 따라 서로 다른 리샘플링 유형을 선택해야 할 것입니다. 래스터의 통계를 변경하고 싶지 않다면 (Linear, Helmert, 또는 Polynomial 1 변형 이외의 알고리즘을 사용하기 때문에 일괄적이지 않은 기하학적 축척 작업이 일어나는 경우를 제외하면), '최근접 이웃'을 선택하는 편이 좋습니다. 반면에'3 차 리샘플링'을 선택하면, 시각적으로 좀 더 다듬어진 결과물을 산출할 것입니다.

다음 다섯 가지 리샘플링 메소드 가운데 하나를 선택할 수 있습니다:

1. 최근접 이웃 (Nearest neighbour)
2. 쌍선형 (2x2 커널)
3. 3 차 (4x4 커널)
4. 3 차 B-스플라인 (4x4 커널)

5. 란초시 (6x6 커널)

### Define the Raster creation options

When exporting a raster, *Raster creation options* allows you to define additional options that control how the output file is structured and compressed. See more at *Raster driver options*.

> **팁**
>
> Select an empty entry if you want to create your own custom combination of parameters.

#### 변형 설정 정의

지리참조된 산출 래스터를 위해 정의해야 할 몇몇 옵션들이 있습니다.

- *Create world file* 옵션은 선형 변형 유형을 선택한 경우에만 사용할 수 있습니다. 월드 파일을 생성한다는 것은 래스터 이미지가 실제로 변형되지 않는다는 의미이기 때문입니다. 이 경우, 새 월드 파일만 생성하기 때문에 *Output raster* 란은 활성화되지 않습니다.
- 다른 모든 변형 유형의 경우, *Output raster* 를 정의해야 합니다. 기본적으로, 원본 래스터 이미지와 동일한 폴더에 `[파일명]_modified` 라는 새 파일을 생성할 것입니다.
- 그 다음 단계로, 지리참조된 래스터를 위한 *Target CRS* (좌표계) 를 정의해야 합니다. (투영 작업 을 참조하세요.)
- 사용자가 원하는 경우, **PDF 지도** 는 물론 **PDF 보고서** 도 **생성** 할 수 있습니다. 이 보고서는 사용한 변형 파라미터, 잔여물의 이미지, 그리고 모든 GCP 및 그 평균 제곱근 오차 (Root Mean Square Error) 목록 관련 정보를 포함합니다.
- 또한, *Set Target Resolution* 옵션을 활성화하면 산출 래스터의 픽셀 해상도를 정의할 수 있습니다. 수평 및 수직 해상도의 기본값은 1 입니다.
- 0 값을 가진 픽셀을 투명하게 가시화해야 하는 경우, *Use 0 for transparency when needed* 옵션을 활성화하면 됩니다. 예시 지형도의 모든 하얀색 영역이 투명하게 변할 것입니다.
- *Save GCP Points* 옵션을 체크하면 산출 래스터 옆에 GCP 포인트를 파일로 저장할 것입니다.
- 마지막으로, *Load in project when done* 옵션을 활성화하면 변형 작업 종료시 QGIS 맵 캔버스에 산출 래스터를 자동적으로 불러올 것입니다.

#### 변형 작업 실행

모든 GCP 를 수집하고 모든 변형 설정을 정의했다면, 그냥 Start georeferencing 버튼을 눌러 새 지리참조 래스터를 생성하십시오.

### 11.3.2 Georeferencing vector layer

Georeferencing vector layers works similarly to raster georeferencing, but instead of matching image pixels, you match vector geometries (points, lines, or polygons) to known spatial references.

The standard procedure starts the same as for raster georeferencing: open QGIS and add a layer to the map canvas to use as a reference. This can be a georeferenced raster or vector layer, or a WMS layer.

Open the Georeferencer dialog from *Layer* → *Georeferencer*.

Start georeferencing by following these steps (in this example, we use an unreferenced `alaska.shp`):

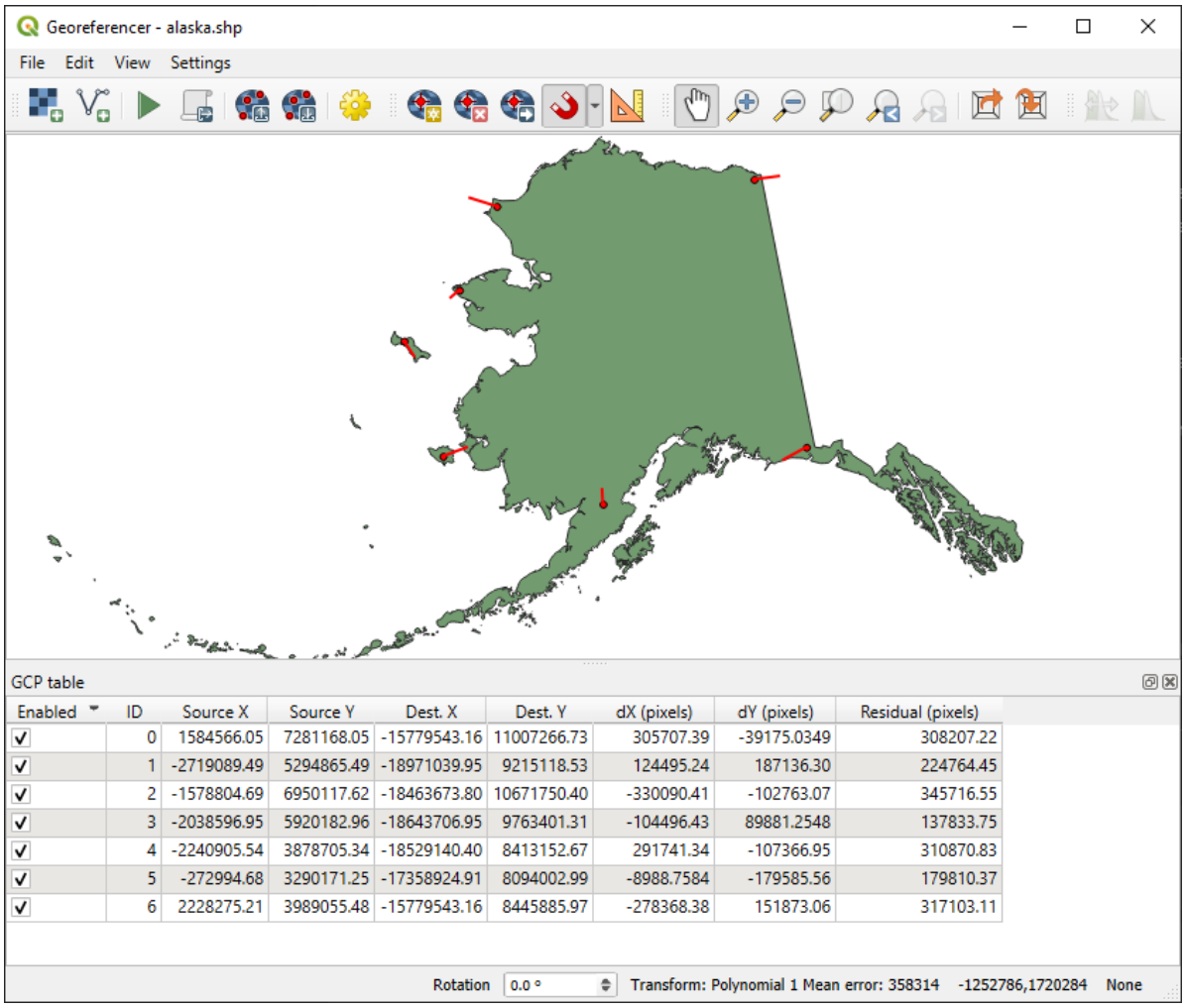



그림 11.41: Vector Georeferencer Dialog

1. Load the unreferenced vector layer using the button. The vector layer will appear in the main working area of the dialog.

2. Activate snapping by clicking the button and selecting the desired snapping type(s). This enables snapping to the reference layer when placing GCPs.

   You can also use the Advanced Digitizing dock to ensure high-precision point selection. For more information, refer to the *Advanced Digitizing* section.

3. Use the  Add GCP Point button to add a point to the working area. Enter its coordinates manually and set the CRS, or click the  From map canvas button to pick the coordinates from a georeferenced layer in the main QGIS map canvas. In that case, the CRS will be set automatically.
4. Define the transformation settings:
   - Select the *Transformation type*. The transformation algorithms are the same as those for raster georeferencing. See 변형 설정 for more details.
   - Define the *Target CRS* (Coordinate Reference System) for the georeferenced vector (see 투영 작업).
   - Set the output file format and path (e.g., GeoPackage, Shapefile). By default, a new file with suffix `_modified` will be created in the same folder as the original vector file.
   - Optionally, enable **Generate PDF map** and **Generate PDF report**. The report includes transformation parameters, GCP residuals, and a summary of RMS errors.
   - Enable  *Save GCP Points* to store GCPs in a file alongside the output vector layer.
   - Enable  *Load in project when done* to add the result directly to the map canvas.
5. Click  Start georeferencing to run the transformation and generate the georeferenced vector layer.

### 11.3.3 Show and adapt layer properties

Clicking on the *Source properties* option in the *Settings* menu opens the *Raster Layer properties* or *Vector Layer properties* depending on the type of layer you are georeferencing.

### 11.3.4 지리참조자 환경 설정

*Settings* → *Configure Georeferencer* 메뉴에서 (또는 `Ctrl+P` 단축키로) 지리참조자의 습성을 사용자 정의할 수 있습니다.

- *Point Tip*: 이 아래 있는 체크박스들을 사용해서 지리참조자 창과 주 맵 캔버스 양쪽에 GCP ID 와 X/Y 좌표를 켜고끌 수 있습니다.
- *Residual Units*: 잔여물 단위를 픽셀 단위 또는 맵 단위로 지정할지 여부를 제어합니다.
- *PDF Report*: 보고서를 내보내기 위해 여백 크기를 밀리미터 단위로 설정할 수 있습니다.
- *PDF Map*: 맵을 내보내기 위한 용지 크기를 선택할 수 있습니다.
- 마지막으로,  *Show Georeferencer window docked* 옵션을 활성화할 수 있습니다. 이 옵션을 체크하면 지리참조자 창을 최소화시킬 수 있는 개별 창이 아니라 QGIS 메인 창에 도킹시킬 것입니다.

## 11.4 데이터 포맷 및 필드 탐구

### 11.4.1 래스터 데이터

GIS 에서 래스터 데이터는 지표면 상, 위 또는 아래에 있는 객체/현상을 표현하는 개별 셀들의 매트릭스를 뜻합니다. 래스터 그리드의 각 셀은 모두 동일한 크기이며 대부분의 경우 직사각형입니다. (QGIS 에서는 언제나 직사각형일 겁니다.) 전형적인 래스터 데이터는 항공사진 또는 위성 이미지와, 표고 또는 기온 같은 모델 데이터 등의 원격탐사 데이터를 포함합니다.

벡터 데이터와는 달리, 래스터 데이터는 일반적으로 각 셀에 관련된 데이터베이스 레코드를 보유하지 않습니다. 래스터 데이터는 래스터 레이어의 픽셀 해상도 및 모서리 픽셀의 X/Y 좌표로 지오코딩돼 있습니다. 덕분에 QGIS 가 맵 캔버스에 데이터를 정확하게 배치할 수 있습니다.

QGIS 작업 시 래스터 데이터를 저장하는 데에는 GeoPackage 포맷이 편리합니다. 유명하고 강력한 GeoTiff 포맷도 훌륭한 대안이지요.

QGIS 는 GeoTiff 같은 래스터 레이어 또는 관련 월드 파일 *(world file)* 내부에 있는 지리참조 정보를 이용해서 데이터를 제대로 표시합니다.

### 11.4.2 벡터 데이터

Many of the features and tools available in QGIS work the same, regardless of the vector data source. However, because of the differences in format specifications (GeoPackage, ESRI Shapefile, MapInfo and MicroStation file formats, AutoCAD DXF, PostgreSQL, SpatiaLite, Oracle Spatial, MS SQL Server, SAP HANA Spatial databases and many more), QGIS may handle some of their properties differently. Support is provided by the GDAL vector drivers. This section describes how to work with some of these specifics.

> **참고**
>
> QGIS 는 Z 그리고/또는 M 값을 보유할 수도 있는 [멀티] 포인트, [멀티] 라인, [멀티] 폴리곤, CircularString, CompoundCurve, CurvePolygon, MultiCurve, MultiSurface 피처 유형을 지원합니다.
>
> 일부 드라이버가 CircularString, CompoundCurve, CurvePolygon, MultiCurve, MultiSurface 같은 일부 피처 유형을 지원하지 않는다는 사실도 알아두어야 합니다. 이 경우 QGIS 는 이 유형들을 변환할 것입니다.

#### GeoPackage

GeoPackage (GPKG) 포맷은 플랫폼 독립적이며, SQLite 데이터베이스 컨테이너로 실행되고, 벡터와 래스터 데이터 둘 다 저장할 수 있습니다. OGC(Open Geospatial Consortium) 가 정의한 이 포맷은 2014 년 공개되었습니다.

SQLite 데이터베이스에 다음을 저장하는 데 GeoPackage 를 사용할 수 있습니다:

- **벡터** 피처
- **이미지의 타일 매트릭스 집합** 및 **래스터** 맵
- 속성 (비공간 데이터)
- 확장자

GeoPackage 는 QGIS 3.8 버전부터 QGIS 프로젝트도 저장할 수 있습니다. GeoPackage 레이어는 JSON 필드를 보유할 수 있습니다.

GeoPackage 는 QGIS 의 벡터 데이터용 기본 포맷입니다.

### ESRI 셰이프파일 포맷

ESRI 셰이프파일 포맷은, 예를 들어 GeoPackage 및 SpatiaLite 와 비교하면 몇몇 제한이 존재하더라도, 지금도 QGIS 에서 가장 자주 이용되는 벡터 파일 포맷 가운데 하나입니다.

ESRI 셰이프파일 포맷 데이터셋은 파일 몇 개로 이루어져 있습니다. 다음은 필수적인 파일 3 개입니다:

1. `.shp`: 피처 도형을 담고 있는 파일
2. `.dbf`: dBase 서식으로 속성을 담고 있는 파일
3. `.shx`: 인덱스 파일

ESRI 셰이프파일 포맷 데이터셋은 `.prj` 확장자를 쓰는 파일도 포함할 수 있는데, 이 파일은 투영체 정보를 담고 있습니다. 투영체 파일이 있다면 매우 유용하지만, 필수적인 파일은 아닙니다. 셰이프파일 포맷 데이터셋은 추가 파일들을 포함할 수도 있습니다. 더 자세한 내용은 ESRI 기술 사양 문서 를 참조하세요.

GDAL 은 압축 ESRI 셰이프파일 포맷 (`shz` 및 `shp.zip`) 읽고 쓰기를 지원합니다.

**ESRI 셰이프파일 포맷 데이터셋 성능 향상**

ESRI 셰이프파일 포맷 데이터셋 렌더링 성능을 향상시키려면, 공간 인덱스를 생성하면 됩니다. 공간 인덱스가 있다면 확대/축소 및 이동 속도도 향상됩니다. QGIS 가 이용하는 공간 인덱스 파일의 확장자는 `.qix` 입니다.

다음 단계를 통해 인덱스를 생성하십시오:

1. ESRI 셰이프파일 포맷 데이터셋을 불러옵니다. (탐색기 패널 참조)
2. 범례에 있는 레이어명을 더블클릭하거나, 또는 오른쪽 클릭한 다음 컨텍스트 메뉴에서 *Properties* 를 선택해서 *Layer Properties* 대화창을 엽니다.
3. *Source* 탭에서 *Create Spatial Index* 버튼을 클릭하십시오.

**.prj 파일을 불러올 때의 문제**

`.prj` 파일을 보유한 ESRI 셰이프파일 포맷 데이터셋을 불러올 때 QGIS 가 해당 파일에서 좌표계를 읽어들일 수 없는 경우, 해당 레이어의 *Layer Properties* → *Source* 탭에서  Select CRS 버튼을 클릭한 다음 적절한 투영체를 직접 정의해줘야 합니다. `.prj` 파일이 QGIS 에서 사용되는, *CRS* 대화창 목록에 있는 완전한 투영체 파라미터를 제공하지 못 하는 경우가 많기 때문입니다.

같은 이유로, QGIS 에서 새 ESRI 셰이프파일 포맷 데이터셋을 생성할 때, ESRI 소프트웨어와 호환되는 제한적인 투영체 파라미터를 담고 있는 `.prj` 파일과, 사용 좌표계의 완전한 파라미터를 제공하는 `.qpj` 파일, 2 개의 서로 다른 투영체 파일을 생성합니다. `.qpj` 파일이 존재하는 경우, QGIS 는 `.prj` 파일 대신 `.qpj` 파일을 이용할 것입니다.

### 구분 텍스트 파일

구분 (delimited) 텍스트 파일은 그 단순성과 가독성 때문에 — 평범한 텍스트 편집기에서도 데이터를 살펴보고 편집할 수 있기 때문에 — 매우 흔하고 널리 쓰입니다. 구분 텍스트 파일은 각 열을 지정된 문자로 구분하고 각 행을 줄바꿈으로 구분하는 테이블형 데이터입니다. 첫 행은 보통 열 이름을 담고 있습니다. CSV(Comma Separated Values) 는 각 열을 쉼표로 구분하는 흔한 구분 텍스트 파일 유형입니다. 구분 텍스트 파일은 위치 정보를 담을 수도 있습니다. (구분 텍스트 파일에 지리 정보 저장하기 참조)

QGIS 는 구분 텍스트 파일을 레이어 또는 일반 테이블로 불러올 수 있습니다. (탐색기 패널 또는구분 텍스트 파일 가져오기 를 참조하세요.) 먼저 파일이 다음 조건을 만족시키는지 확인하십시오:

1. 파일이 필드명을 담은 구분된 헤더 (header) 행을 보유해야만 합니다. 헤더 행은 데이터의 첫 줄 (이상적으로는 텍스트 파일의 첫 행) 이어야만 합니다.
2. 도형을 활성화해야 하는 경우, 파일이 도형 정의 필드 (들) 를 포함해야만 합니다. 이 필드 (들) 의 이름은 어떤 것이라도 가능합니다.

3. 도형이 좌표로 정의된 경우, X 와 Y 좌표 필드는 숫자로 지정돼야만 합니다. 어떤 좌표계인지는 중요하지 않습니다.
4. 문자열이 아닌 데이터를 CSV 파일로 저장하는 경우 CSVT 파일을 이용하면 됩니다. (필드 서식 제어에 *CSVT 파일 사용하기* 를 참조하세요.)

무결한 텍스트 파일의 예로, QGIS 예시 데이터셋 (예시 데이터 다운로드 참조) 에 들어 있는 다음 `elevp.csv` 표고 포인트 데이터 파일을 불러들이겠습니다:

```
X;Y;ELEV
-300120;7689960;13
-654360;7562040;52
1640;7512840;3
[...]
```

이 텍스트 파일에 대해 알아두어야 할 점은 다음과 같습니다:

1. 이 예시 텍스트 파일은 구분자로 `;` (쌍반점) 을 씁니다. (어떤 문자라도 필드를 구분하는 데 쓸 수 있습니다.)
2. 첫 줄이 헤더 행입니다. `X`, `Y` 및 `ELEV` 필드를 담고 있습니다.
3. 텍스트 필드를 구분하는 데 큰따옴표 (`"`) 를 쓰지 않습니다.
4. X 좌표는 `X` 필드에 담겨 있습니다.
5. Y 좌표는 `Y` 필드에 담겨 있습니다.

### 구분 텍스트 파일에 지리 정보 저장하기

구분 텍스트 파일은 다음과 같은 주된 양식 2 개로 지리 정보를 담을 수 있습니다:

- 포인트 도형 데이터의 경우, 분리된 열에 (예: `Xcol`, `Ycol` 등등) 좌표로 저장
- 모든 도형 유형의 경우, 단일 열에 도형의 WKT(Well-Known Text) 표현으로 저장

만곡 도형 (CircularString, CurvePolygon 및 CompoundCurve) 피처를 지원합니다. 다음은 구분 텍스트 파일에 도형을 WKT 로 코딩한 도형 유형의 예시입니다:

```
Label;WKT_geom
LineString;LINESTRING(10.0 20.0, 11.0 21.0, 13.0 25.5)
CircularString;CIRCULARSTRING(268 415,227 505,227 406)
CurvePolygon;CURVEPOLYGON(CIRCULARSTRING(1 3, 3 5, 4 7, 7 3, 1 3))
CompoundCurve;COMPOUNDCURVE((5 3, 5 13), CIRCULARSTRING(5 13, 7 15,
  9 13), (9 13, 9 3), CIRCULARSTRING(9 3, 7 1, 5 3))
```

구분 텍스트 파일은 도형의 Z 및 M 좌표도 지원합니다:

```
LINESTRINGZ(10.0 20.0 30.0, 11.0 21.0 31.0, 11.0 22.0 30.0)
```

#### 필드 서식 제어에 CSVT 파일 사용하기

CSV 파일을 불러올 때, 달리 지정하지 않는 이상 GDAL 드라이버는 모든 필드가 (텍스트 등의) 문자열이라고 가정합니다. GDAL 드라이버에 (그리고 QGIS 에) 서로 다른 열들이 어떤 데이터 유형인지 알려준 다음 CSVT 파일을 생성할 수 있습니다:

| 유형 | 이름 | 예시 |
|---|---|---|
| 범자연수 (Whole number) | 정수 | 4 |
| 불 (Boolean) | Integer(Boolean) | true |
| 십진수 (Decimal number) | Real | 3.456 |
| 날짜 | Date (YYYY-MM-DD) | 2016-07-28 |
| 시간 | Time (HH:MM:SS+nn) | 18:33:12+00 |
| 날짜 & 시간 | DateTime (YYYY-MM-DD HH:MM:SS+nn) | 2016-07-28 18:33:12+00 |
| CoordX | CoordX | 8.8249 |
| CoordY | CoordY | 47.2274 |
| Point(X) | Point(X) | 8.8249 |
| Point(Y) | Point(Y) | 47.2274 |
| WKT | WKT | POINT(15 20) |

CSVT 파일은 데이터 유형을 따옴표로 표현하고 쉼표 등으로 구분하는, 다음과 같은 **한 줄** 짜리 평문 텍스트 파일입니다:

```
"Integer","Real","String"
```

각 열의 길이 (width) 와 정밀도도 다음과 같이 설정할 수 있습니다:

```
"Integer(6)","Real(5.5)","String(22)"
```

이 파일은 `.csv` 파일과 동일한 폴더에 저장됩니다. 파일명은 CSV 파일과 동일해야 하지만, 확장자는 `.csvt` 를 씁니다.

*You can find more information at* GDAL CSV Driver.

> **팁**
>
> **필드 유형 탐지**
>
> 데이터 유형을 말해주는 CSVT 파일을 사용하는 대신, QGIS 는 필드 유형을 자동으로 탐지하고 해당 필드 유형으로 변경시킬 수 있는 기능을 제공하고 있습니다.

### PostgreSQL Database

In order to process geographical information in a PostgreSQL database, the PostGIS extension must first be installed. PostGIS extends the capabilities of the PostgreSQL relational database by adding support for storing, indexing, and querying geospatial data.

To enable PostGIS in your database you have to activate the extension in your database (open database and run `CREATE EXTENSION postgis;`). PostGIS enabled databases are also often named "PostGIS layer". Using PostGIS, vector functions such as select and identify work more accurately than they do with GDAL layers in QGIS.

> **팁**
>
> **PostGIS 레이어**
>
> 일반적으로, geometry_columns 테이블의 항목이 PostGIS 레이어를 식별합니다. QGIS 는 geometry_columns 테이블에 항목이 없는 레이어를 불러올 수 있습니다. 테이블과 뷰 둘 다 말입니다. 뷰 생성에 대한 정보는 PostgreSQL 사용자 지침서를 참조하십시오.

QGIS 가 PostgreSQL 레이어에 접근하는 몇몇 방법에 대해 자세히 설명하겠습니다. 대부분의 경우 QGIS 는 불러올 수 있는 데이터베이스 테이블 목록을 제공하고, 사용자 요청에 따라 테이블을 불러올 것입니다. 하지만 QGIS 로 PostgreSQL 테이블을 불러오는 데 문제가 있을 경우, 다음 내용을 통해 QGIS 메시지를 이해해서 PostgreSQL 테이블 또는 뷰 정의를 수정하면 QGIS 가 테이블 또는 뷰를 불러올 수 있을 것입니다.

> **참고**
>
> PostgreSQL 데이터베이스도 QGIS 프로젝트를 저장할 수 있습니다.

### 기본 키

QGIS 가 PostgreSQL 레이어를 불러오려면, 레이어에 유일 (unique) 키로 이용할 수 있는 열이 있어야 합니다. 테이블의 경우, 일반적으로 테이블에 기본 키 (primary key) 또는 유일 제약조건 열이 있어야 한다는 뜻입니다. QGIS 에서 이 열은 int4(4 바이트 정수) 유형이어야 합니다. CTID 열을 기본 키로 대신 쓸 수도 있습니다. 테이블에 이런 항목들이 없는 경우, OID 열을 대신 사용할 것입니다. 열을 인덱스 작업하면 성능이 향상됩니다. (PostgreSQL 은 기본 키를 자동적으로 인덱스 작업한다는 사실을 기억하십시오.)

QGIS 의 *Select at id* 체크박스는 기본적으로 활성화돼 있습니다. 이 옵션을 활성화하면 속성 없이 ID 를 가져오는데, 대부분의 경우 훨씬 빠릅니다.

### 뷰

PostgreSQL 레이어가 뷰인 경우 요구 사항은 동일하지만, 뷰가 언제나 기본 키 또는 유일 제약조건 열을 보유하고 있는 것은 아닙니다. QGIS 에 뷰를 불러오려면 먼저 대화창에서 기본 키 항목을 (정수 유형으로) 정의해야 합니다. 뷰에 적당한 열이 없는 경우, QGIS 는 레이어를 불러오지 못 합니다. 이 경우 적당한 열을 (기본 키 또는 유일 제약조건을 가진 정수 유형, 인덱스 작업 추천) 포함하도록 뷰를 수정해야 합니다.

테이블인 경우, *Select at id* 체크박스가 기본적으로 활성화돼 있습니다. (이 체크박스의 의미는 바로 앞에서 설명했습니다.) 사용자가 대용량 뷰를 이용하는 경우 이 옵션을 비활성화하는 편이 좋습니다.

> **참고**
>
> **PostgreSQL 외래 테이블**
>
> PostgreSQL 제공자는 PostgreSQL 외래 (foreign) 테이블을 분명하게 지원하지 않으며, 뷰처럼 처리할 것입니다.

### QGIS layer_style 테이블 및 데이터베이스 백업

If you want to make a backup of your PostgreSQL database using the `pg_dump` and `pg_restore` commands, and the default layer styles as saved by QGIS fail to restore afterwards, you need to set the XML option to `DOCUMENT` before the restore command:

1. `layer_style` 테이블을 평문으로 백업하십시오.
2. 텍스트 편집기에서 백업 파일을 여십시오.
3. `SET xmloption = content;` 줄을 `SET XML OPTION DOCUMENT;` 로 변경하십시오.
4. 파일을 저장하십시오.
5. PostgreSQL 의 psql 을 이용해서 새 데이터베이스에 테이블을 복원하십시오.

### 데이터베이스 쪽에서 필터링

QGIS 는 서버 쪽에 있는 피처를 필터링할 수 있습니다. *Settings* → *Options* → *Data Sources* → *Execute expressions on server-side if possible* 체크박스를 활성화하면 됩니다. 이 옵션을 활성화하면 데이터베이스에 지원하는 표현식만 전송할 것입니다. 지원하지 않는 연산자 또는 함수를 이용한 표현식은 자연스럽게 폴백 (fallback) 되어 로컬에서 평가될 것입니다.

### PostgreSQL 데이터 유형의 지원

PostgreSQL 제공자는 정수형, 부동소수점 실수형, 불린 (boolean), 2 진 객체 (binary object), 문자형 (varchar), 도형형, 타임스탬프 (timestamp), 배열 (array), HStore, JSON 등의 데이터 유형을 지원합니다.

### PostgreSQL 로 데이터 가져오기

Data can be imported into PostgreSQL using several tools, including the Browser, DB Manager plugin and the command line tools shp2pgsql and ogr2ogr.

### DB 관리자

QGIS 는 DB Manager 라는 핵심 플러그인을 내장하고 있습니다. 이 도구는 데이터를 불러오는 데 쓸 수 있으며, 스키마를 지원하기도 합니다. 더 자세한 내용은데이터베이스 관리자 플러그인 을 참조하세요.

### shp2pgsql

PostGIS 는 PostGIS 호환 데이터베이스로 셰이프파일 포맷 데이터셋을 불러올 수 있는 **shp2pgsql** 이라는 기능을 포함하고 있습니다. 예를 들어, `gis_data` 라는 PostgreSQL 데이터베이스로 `lakes.shp` 이라는 셰이프파일 포맷 데이터셋을 가져오려면, 다음 명령어를 사용하십시오:

```
shp2pgsql -s 2964 lakes.shp lakes_new | psql gis_data
```

이 명령어는 `gis_data` 데이터베이스에 `lakes_new` 라는 새 레이어를 생성합니다. 이 새 레이어는 SRID(Spatial Reference IDentifier) 2964 를 적용받게 됩니다. 공간 참조 시스템 및 투영체에 대한 더 자세한 내용은투영 작업 을 참조하세요.

> **팁**
>
> **Exporting datasets from PostgreSQL**
>
> There is also a tool for exporting postgreSQL datasets wth geographic data to Shapefile format: **pgsql2shp**. It is shipped within your PostGIS distribution.

### ogr2ogr

In addition to **shp2pgsql** and **DB Manager**, there is another tool for feeding geographical data in PostgreSQL: **ogr2ogr**. It is part of your GDAL installation.

To import a Shapefile format dataset into PostgreSQL, do the following:

```
ogr2ogr -f "PostgreSQL" PG:"dbname=postgis host=myhost.de user=postgres
password=topsecret" alaska.shp
```

This will import the Shapefile format dataset `alaska.shp` into the database *postgis* using the user *postgres* with the password *topsecret* on the host server *myhost.de*.

Note that GDAL must be built with PostgreSQL. You can verify this by typing (in  ):

```
ogrinfo --formats | grep -i post
```

PostgreSQL 의 기본 메소드인 **INSERT INTO** 대신 **COPY** 명령어를 더 선호하는 경우, 다음 명령어로 (적어도  와  에서는) 다음 환경 변수를 내보낼 수 있습니다:

```
export PG_USE_COPY=YES
```

**ogr2ogr** 도구는 **shp2pgsl** 처럼 공간 인덱스를 생성하지 않습니다. 셰이프파일 포맷 데이터셋을 불러온 후에 (다음성능 향상시키기 에서 설명하는 대로) 추가 단계로 일반적인 SQL 명령어 **CREATE INDEX** 를 이용해서 직접 생성해야 합니다.

#### 성능 향상시키기

Retrieving features from a PostgreSQL database can be time-consuming, especially over a network. You can improve the drawing performance of PostgreSQL layers by ensuring that a spatial index exists on each layer in the database. PostGIS supports creation of a GiST (Generalized Search Tree) index to speed up spatial searching (GiST index information is taken from the PostGIS documentation available at https://postgis.net).

> **팁**
>
> DB 관리자를 이용해서 사용자 레이어에 인덱스를 생성할 수 있습니다. 먼저 레이어를 선택한 다음 *Table → Edit table* 메뉴를 클릭하고, *Indexes* 탭에서 *Add Spatial Index* 를 클릭하십시오.

GiST 인덱스를 생성하기 위한 문법은 다음과 같습니다:

```
CREATE INDEX [indexname] ON [tablename]
  USING GIST ( [geometryfield] GIST_GEOMETRY_OPS );
```

대용량 테이블의 경우 인덱스 생성 작업이 오래 걸릴 수 있다는 점을 기억하십시오. 인덱스가 생성되고 나면, `VACUUM ANALYZE` 명령을 실행해야 합니다. 더 자세한 내용은 PostGIS 문서 (참고 문헌 및 웹사이트 에 있는 POSTGIS-PROJECT) 를 참조하세요.

다음은 GiST 인덱스를 생성하는 예시입니다:

```
gsherman@madison:~/current$ psql gis_data
Welcome to psql 8.3.0, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help with psql commands
       \g or terminate with semicolon to execute query
       \q to quit

gis_data=# CREATE INDEX sidx_alaska_lakes ON alaska_lakes
gis_data-# USING GIST (geom GIST_GEOMETRY_OPS);
CREATE INDEX
gis_data=# VACUUM ANALYZE alaska_lakes;
VACUUM
gis_data=# \q
gsherman@madison:~/current$
```

### SpatiaLite 레이어

벡터 레이어를 SpatiaLite 포맷으로 저장하고 싶다면, 기존 레이어로부터 레이어 생성 에 있는 지침을 따르면 됩니다. *Format* 을 `SpatiaLite` 으로 선택하고 *File name* 과 *Layer name* 을 둘 다 입력하십시오.

Also, you can select `SQLite` as format and then add `SPATIALITE=YES` in the *Custom Options* → *Data source* field. This tells GDAL to create a SpatiaLite database. See also https://gdal.org/en/latest/drivers/vector/sqlite.html.

QGIS 는 SpatiaLite 에서 편집할 수 있는 뷰도 지원합니다. SpatiaLite 데이터를 관리하고자 한다면, 핵심 플러그인인*DB* 관리자 를 이용할 수도 있습니다.

새 SpatiaLite 레이어를 생성하고 싶다면, 새 *SpatiaLite* 레이어 생성하기 를 참조해주세요.

### GeoJSON 특화 파라미터

GeoJSON 으로레이어를 내보내는 경우, 특화된 몇몇 *Layer Options* 를 사용할 수 있습니다. 이 특화 옵션은 해당 파일의 작성을 책임지는 GDAL 이 제공합니다.

- *COORDINATE_PRECISION*: 좌표 작성 시 소수점 뒤에 올 최대 자릿수입니다. 기본값은 15 입니다. (참고: 위경도 좌표의 경우 6 이면 충분하다고 간주합니다.) 잘라내기 (truncation) 기능이 이 자릿수 뒤의 0 들을 제거할 것입니다.
- *RFC7946* by default GeoJSON 2008 will be used. If set to YES, the updated RFC 7946 standard will be used. Default is NO (thus GeoJSON 2008). See https://gdal.org/en/latest/drivers/vector/geojson.html#rfc-7946-write-support for the main differences, in short: only EPSG:4326 is allowed, other crs's will be transformed, polygons will be written such as to follow the right-hand rule for orientation, values of a "bbox"array are [west, south, east, north], not [minx, miny, maxx, maxy]. Some extension member names are forbidden in FeatureCollection, Feature and Geometry objects, the default coordinate precision is 7 decimal digits
- *WRITE_BBOX*: YES 로 설정하면 피처 및 피처 집합 수준에서 도형의 경계 상자를 포함합니다.

Besides GeoJSON there is also an option to export to "GeoJSON - Newline Delimited"(see https://gdal.org/en/latest/drivers/vector/geojsonseq.html). Instead of a FeatureCollection with Features, you can stream one type (probably only Features) sequentially separated with newlines.

GeoJSON - 새줄 문자 구분도 몇몇 특화된 레이어 옵션을 사용할 수 있습니다:

- *COORDINATE_PRECISION*: 앞의 설명 참조 (GeoJSON 과 동일)
- *RS*: 레코드를 RS=0x1E 문자로 시작할지 여부를 결정합니다. 피처를 어떻게 구분하는가의 차이입니다: 새줄 (LF) 문자만으로 구분할 수도 있고 (새줄 문자 구분 JSON, geojsonl) 또는 접두어로 레코드 구분자 (RS) 문자를 붙일 수도 (GeoJSON 텍스트 시퀀스, geojsons) 있습니다. 기본값은 NO 입니다. 확장자를 지정하지 않을 경우 파일에 `.json` 확장자를 붙입니다.

### SAP HANA 공간 레이어

QGIS 가 SAP HANA 레이어에 접근하는 몇몇 방법에 대해 자세히 설명하겠습니다. 대부분의 경우 QGIS 는 불러올 수 있는 데이터베이스 테이블 및 뷰 목록을 제공하고, 사용자 요청에 따라 테이블을 불러올 것입니다. 하지만 QGIS 로 SAP HANA 테이블 및 뷰를 불러오는 데 문제가 있을 경우, 다음 내용을 통해 기저 원인을 이해하고 문제점을 해결할 수 있을 것입니다.

#### 피처 식별

QGIS 의 모든 편집 작업 케이퍼빌리티를 이용하고 싶다면, QGIS 가 레이어에 있는 각 피처를 확실하게 식별할 수 있어야만 합니다. QGIS 는 내부적으로 64 비트 부호 있는 정수형을 사용해서 피처를 식별하는 데, 이때 음수 범위를 특수 목적 용으로 예약해 둡니다.

따라서 SAP HANA 제공자가 QGIS 의 피처 편집 작업 케이퍼빌리티를 완전하게 지원하려면 양의 64 비트 정수형에 매핑될 수 있는 유일 키가 필요하게 됩니다. 이런 매핑을 할 수 없는 경우, 그래도 피처를 볼 수는 있겠지만 편집 작업은 할 수 없을지도 모릅니다.

#### 테이블 추가하기

테이블을 레이어로 추가하는 경우 SAP HANA 제공자는 테이블의 기본 키를 이용해서 유일 FID(Feature ID) 에 매핑시킵니다. 따라서 피처 편집 작업을 완전하게 지원하려면 사용자의 테이블 정의에 기본 키가 있어야 합니다.

SAP HANA 제공자가 다중 열 기본 키를 지원하긴 하지만, 최고의 성능을 보이길 원한다면 사용자의 기본 키가 `INTEGER` 유형의 단일 열이어야 합니다.

#### 뷰 추가하기

뷰를 레이어로 추가하는 경우 SAP HANA 제공자는 피처를 확실하게 식별하는 열을 자동으로 식별하지 못 합니다. 게다가 몇몇 뷰는 읽기 전용으로 편집할 수 없기도 합니다.

피처 편집 작업을 완전하게 지원하려면, 뷰가 업데이트 가능해야 하며 (해당 뷰에 대한 `SYS.VIEWS` 시스템 뷰에 있는 `IS_READ_ONLY` 열을 확인하십시오) 사용자가 QGIS 에 피처를 식별하는 하나 이상의 열을 직접 지정해줘야만 합니다. *Layer → Add Layer → Add SAP HANA Spatial Layer* 메뉴에서 *Feature id* 열에 있는 열을 선택하면 열을 지정할 수 있습니다. 최고의 성능을 바란다면, *Feature id* 값이 단일 `INTEGER` 열이어야 합니다.

### 11.4.3 경도 180° 선을 가로지르는 레이어

많은 GIS 패키지들이 경도 180° 선을 가로지르는 지리 참조 시스템 (위도/경도) 으로 레이어를 감싸고 있지 않습니다. 그 결과, QGIS 에서 이런 레이어를 열었을 때, 서로 가까이 있어야 하지만 멀리 떨어져 있는 두 위치를 보게 됩니다. 그림 11.42 에서, 맵 캔버스 왼쪽 구석에 있는 작은 점들 (채텀 제도) 은 그리드 내부, 뉴질랜드 본도의 오른쪽에 있어야 합니다.

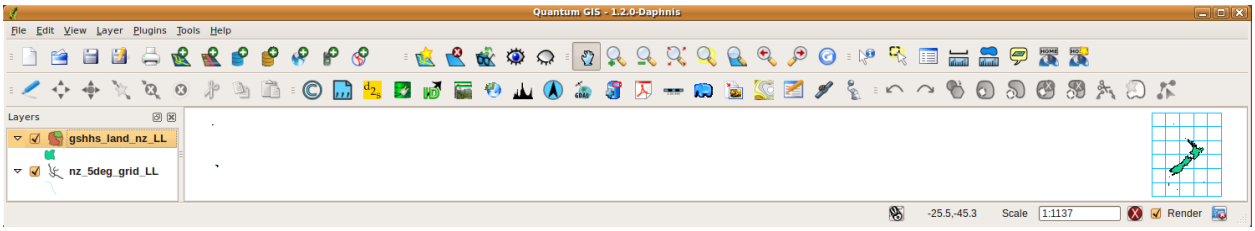



그림 11.42: 경도 180° 선을 가로지르는 위도/경도 맵

#### Solving in PostgreSQL

A work-around is to transform the longitude values using PostgreSQL and the ST_ShiftLongitude function. This function reads every point/vertex in every component of every feature in a geometry, and shifts its longitude coordinate from -180..0° to 180..360° and vice versa if between these ranges. This function is symmetrical so the result is a 0..360° representation of a -180..180° data and a -180..180° representation of a 0..360° data.

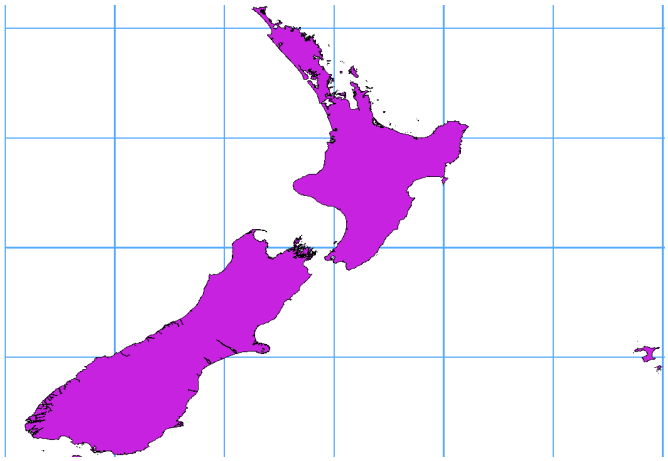


그림 11.43: 경도 180° 선을 가로지르는 데이터에 **ST_ShiftLongitude** 함수를 적용

1. Import data into PostgreSQL (*PostgreSQL* 로 데이터 가져오기) using, for example, the DB Manager plugin.
2. Use the PostgreSQL command line interface to issue the following command:

```
-- In this example, "TABLE" is the actual name of your PostgreSQL table
update TABLE set geom=ST_ShiftLongitude(geom);
```

3. 모든 작업이 제대로 됐다면, 업데이트한 피처의 개수를 확인하는 메시지를 볼 수 있을 겁니다. 그리고 나서야 맵을 불러와서 바뀐 모습 (그림 11.43) 을 볼 수 있습니다.

CHAPTER 12

# 벡터 데이터 작업

## 12.1 벡터 속성 대화창

벡터 레이어의 *Layer Properties* 대화창은 맵에서 레이어의 피처가 어떻게 보이는지 (심볼, 라벨, 도표) 및 마우스와의 상호 작용 (액션, 맵 도움말, 양식 디자인) 을 관리하는 일반 설정을 제공합니다. 또 레이어에 대한 정보를 제공하기도 합니다.

*Layer Properties* 대화창을 열려면:

- *Layers* 패널에서 레이어를 더블 클릭하거나, 오른쪽 클릭한 다음 컨텍스트 메뉴에서 *Properties*…를 선택하십시오.
- 레이어를 선택한 다음 *Layer* ▷ *Layer Properties*…메뉴를 클릭하십시오.

벡터 *Layer Properties* 대화창은 다음 탭들로 이루어져 있습니다:

| | | |
|---|---|---|
| 정보 | 소스 | *Symbology* ([1]) |
| *Labels* ([Page 582, 1]) | *Mask* ([Page 582, 1]) | *3D View* ([Page 582, 1]) |
| 도표 | 필드 | 속성 양식 |
| 결합 | 보조 저장소 | 액션 |
| 표시 | 렌더링 | 시계열 |
| 변수 | 표고 | 메타데이터 |
| 의존성 | 범례 | *QGIS* 서버 |
| 디지타이즈 작업 | *External plugins* ([2]) tabs | |

> **팁**
>
> **레이어 스타일의 전체 또는 일부 속성 공유**
>
> 대화창 하단에 있는 *Style* 메뉴를 통해 여러 대상 (파일, 클립보드, 데이터베이스 등) 에서/으로 이런 전체 또는 일부 속성들을 가져오거나 내보낼 수 있습니다. 사용자 지정 스타일 관리하기 를 참조하세요.

> **참고**
>
> 원본 프로젝트 파일에서 삽입 레이어 (외부 프로젝트로부터 레이어를 내포시키기 참조) 의 속성들 (심볼, 라벨, 액션, 기본값, 양식 등등) 을 읽어오기 때문에, 이 습성을 방해할 수도 있는 변경 사항이 적용되는 일을 피하기 위해 삽입 레이어에 대해 레이어 속성 대화창을 사용할 수 없게 돼 있습니다.

### 12.1.1 정보 속성

*Information* 탭은 읽기 전용으로, 현재 레이어에 대한 요약 정보 및 메타데이터를 한 눈에 볼 수 있는 흥미로운 장소입니다. 다음과 같은 정보를 제공합니다:

- 프로젝트 이름, 소스 경로, 보조 파일 목록, 최신 저장 시간 및 용량, 사용하는 제공자 등의 일반 정보
- custom properties, used to store in the active project additional information about the layer. Default custom properties may include *layer notes*, *legend widgets*, *layer variables*, *form properties*…More custom properties can be created and managed using PyQGIS, specifically through the setCustomProperty() method.
- 레이어 제공자 기반 정보: 저장소 포맷, 도형 유형, 데이터소스 인코딩, 범위, 피처 개수 등등
- 좌표계: 이름, 단위, 메소드, 정확도, 참조 (예: 정적인지 동적인지 여부)
- 채워진 메타데이터 로부터 선택: 접근, 범위, 링크, 연락처, 이력, …
- 레이어의 도형 관련 정보 (공간 범위, 좌표계 등등) 또는 속성 (필드 개수, 각 속성의 특성 등등)

### 12.1.2 소스 속성

*Source* 탭에서 벡터 레이어에 대한 일반 설정을 정의할 수 있습니다.

#### 설정

- *Layer name*: (*Layers Panel* 에서 표현식 등을 이용해서, 또는 인쇄 조판 범례에서, 또는…) 프로젝트 내의 레이어를 식별하기 위해 쓰일, 레이어 파일명과는 다른 레이어 이름을 설정할 수 있습니다.
- *Data source encoding*: QGIS 가 정확하게 탐지하지 못 한 경우, 데이터 포맷에 따라 데이터소스 인코딩을 선택하십시오.

[1] Also available in the *Layer styling panel*
[2] *External plugins* you install can optionally add tabs to this dialog. Those are not presented in this document. Refer to their documentation.

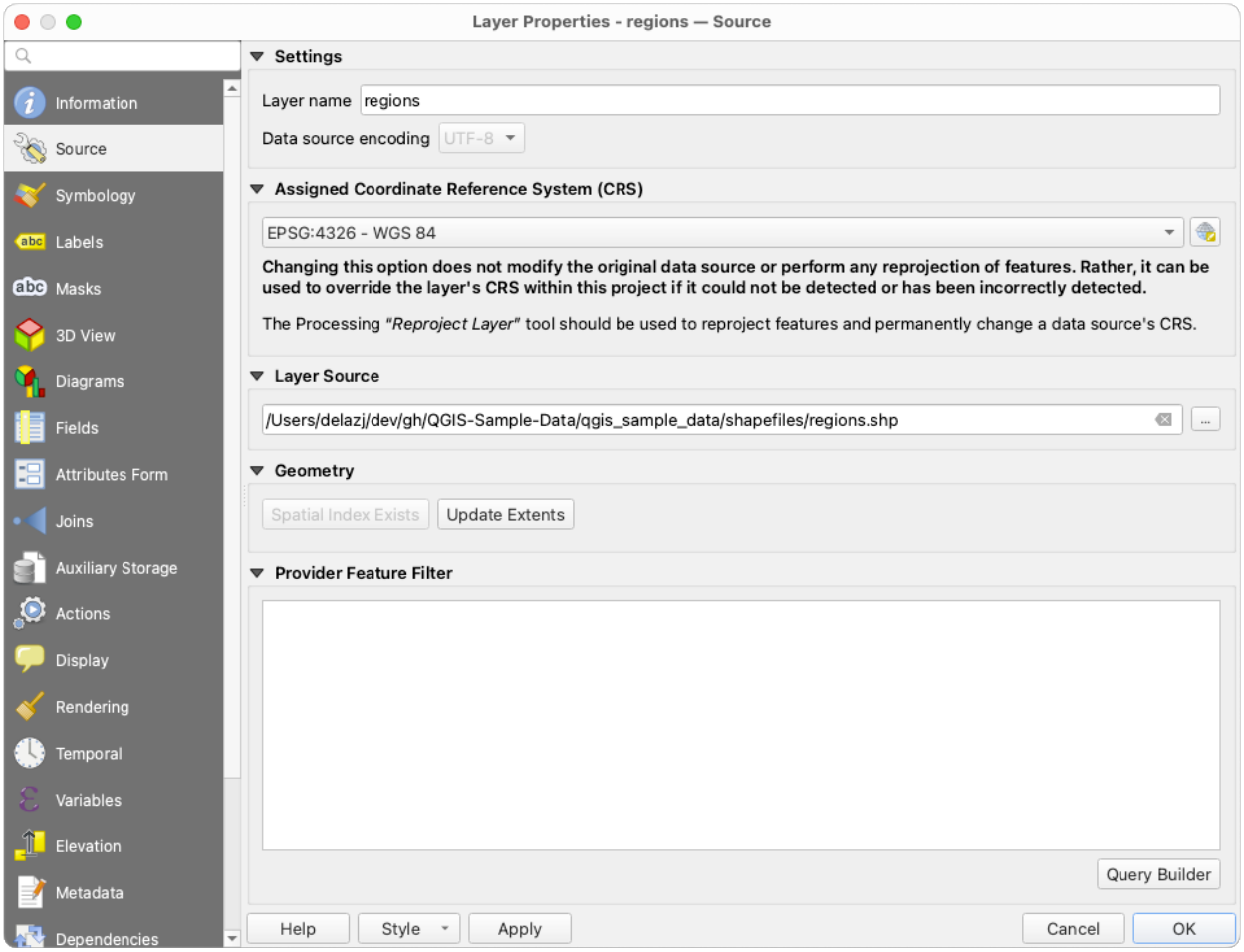



그림 12.1: 벡터 레이어 속성 대화창의 소스 탭

### Coordinate Reference System

- 레이어의할당 좌표계 를 표시합니다. 드롭다운 목록에서 최근 사용한 좌표계를 선택하거나 Select CRS 버튼을 (좌표계 선택기 참조) 클릭해서 레이어 좌표계를 변경할 수 있습니다. 레이어에 적용된 좌표계가 틀렸거나 또는 적용된 좌표계가 없는 경우에만 변경하십시오. 사용자 데이터를 다른 좌표계로 재투영하고 싶은 경우, 공간 처리 재투영 알고리즘을 이용하거나새 레이어로 저장 하는 편이 좋습니다.

### Layer source and Geometry

- Depending on the data provider, a *Layer source* group indicates the path to the source of the dataset and allows for replacing the loaded layer:
  - When the layer is stored as file on disk, edit the path shown in the text box or press ⋯ Browse to select another file on the disk. Both layers do not need to share attribute fields, geometry type or file formats.
  - When the layer is provided by an ArcGIS Feature service, it is possible to modify its *authentication settings*, while keeping unchanged the details for connecting to the service.
- *Create spatial index* (only for OGR-supported formats): helps speeding layer rendering and features' geometry retrieval.
- *Update extents*: 레이어의 범위 정보를 업데이트합니다.

#### 쿼리 작성기

레이어 속성 대화창의 *Source* 탭의 *Provider Feature Filter* 부분 하단에 있는 *Query Builder* 버튼을 누르면 쿼리 작성기 대화창을 열 수 있습니다.

쿼리 작성자는 SQL 호환 WHERE 절을 이용해서 레이어에 있는 피처의 하위 집합을 정의하고 그 결과를 메인 창에 표시할 수 있는 인터페이스를 제공합니다. 쿼리를 비활성화하지 않는 한, 프로젝트에서 쿼리 결과에 상응하는 피처만 사용할 수 있습니다.

`Query Builder` 에서 필터를 정의하는 데 하나 이상의 레이어 속성을 사용할 수 있습니다. 그림 12.2 에서 하나 이상의 속성을 사용하는 방법을 볼 수 있습니다. 이 예시에서, 필터는 AND, OR 및 NOT 연산자와 괄호를 이용해서 다음

- `toa` (`DateTime` 필드: `cast("toa" as character) > '2017-05-17'` 그리고 `cast("toa" as character) < '2019-12-24T18:00:00'`)
- `name` (`String` 필드: `"name" > 'S'`)
- `FID` (`Integer` 필드: `FID > 10`)

속성들을 결합합니다. 이 문법은 (`toa` 필드 용 날짜 & 시간 유형을 포함해서) GeoPackage 데이터셋용입니다.

데이터 제공자 (GDAL, PostgreSQL, MS SQL 서버 등등) 수준에서 필터를 생성하기 때문에, 문법은 데이터 제공자를 따릅니다. (예를 들어 ESRI 셰이프파일 포맷은 날짜 & 시간 유형을 지원하지 않습니다.) 완전한 표현식은 다음과 같습니다:

```
cast("toa" as character) > '2017-05-17' AND
cast("toa" as character) < '2019-12-24T18:00:00' AND
NOT ("name" > 'S' OR FID > 10)
```

*Layer* 메뉴 또는 레이어의 컨텍스트 메뉴에서 *Filter*⋯옵션을 이용해서도 *Query Builder* 대화창을 열 수 있습니다. 대화창의 *Fields*, *Values* 및 *Operators* 부분에서 *Provider specific filter expression* 상자에 노출되는 SQL 호환 쿼리를 작성할 수 있습니다.

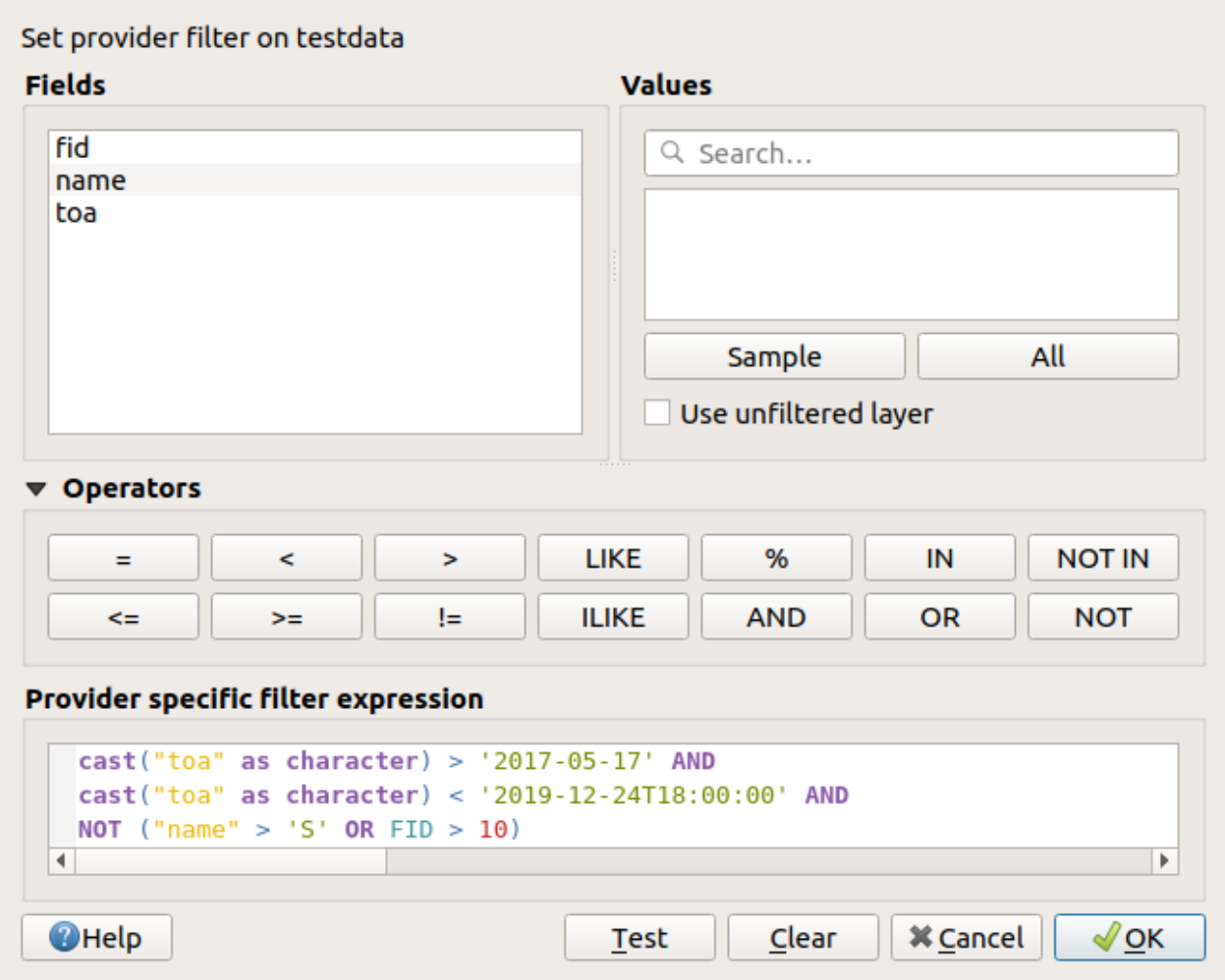

Set provider filter on testdata
Fields
fid
name
toa
Values
Search...
Sample
All
Use unfiltered layer
Operators
=
<
>
LIKE
%
IN
NOT IN
<=
>=
!=
ILIKE
AND
OR
NOT
Provider specific filter expression
cast("toa" as character) > '2017-05-17' AND
cast("toa" as character) < '2019-12-24T18:00:00' AND
NOT ("name" > 'S' OR FID > 10)
Help
Test
Clear
Cancel
OK

그림 12.2: 쿼리 작성기

**필드** 목록은 레이어의 모든 필드를 담고 있습니다. SQL WHERE 절 란에 속성 열을 추가하려면, 필드 목록에 있는 이름을 더블 클릭하거나 SQL 란에 직접 입력하십시오.

**값** 프레임은 현재 선택한 필드의 값들을 목록화합니다. 필드의 모든 유일 값들을 표시하려면, *All* 버튼을 클릭하십시오. 열의 처음 유일한 값 25 개만 표시하려면, *Sample* 버튼을 클릭하십시오. SQL WHERE 절 란에 값을 추가하려면, 값 목록에 있는 이름을 더블 클릭하십시오. 값 프레임 상단에 있는 검색란을 사용하면 목록에 있는 속성 값을 쉽게 탐색하고 찾을 수 있습니다.

**연산자** 부분은 사용할 수 있는 모든 연산자를 담고 있습니다. SQL WHERE 절 란에 연산자를 추가하려면, 적절한 버튼을 클릭하십시오. 관계 연산자 (=, > 등등), 문자열 비교 연산자 (`LIKE`), 그리고 논리 연산자 (`AND`, `OR` 등등)를 사용할 수 있습니다.

*Test* 버튼은 사용자의 쿼리를 점검하고 현재 쿼리를 만족하는 피처의 개수를 보여주는 메시지 창을 엽니다. *Clear* 버튼을 누르면 SQL 쿼리를 모두 삭제하고 레이어를 원래 상태로 (예를 들면 모든 피처를 다시 불러와) 되돌립니다. 쿼리를 `.QQF` 파일로 저장 *Save*…할 수도 있고, 파일에서 쿼리를 대화창으로 불러올 *Load*…수도 있습니다.

필터를 적용하면, QGIS 는 쿼리로 선택된 하위 집합이 마치 완전한 레이어인 것처럼 취급합니다. 예를 들면, 사용자가 앞의 그림처럼'Borough'에 대해 필터를 적용한 경우 (`"TYPE_2" = 'Borough'`) `Anchorage` 를 표시하거나 쿼리하거나 저장하거나 편집할 수 없습니다. `Anchorage` 는'Municipality'이기 때문에 하위 집합에 속하지 않기 때문입니다.

**팁**

**레이어 패널은 필터링된 레이어를 표시합니다**

*Layers* 패널에서 필터링된 레이어에 마우스를 가져가면 레이어 옆에 쿼리가 사용됐다는 사실을 알려주는  Filter 아이콘이 나타납니다. 이 아이콘을 더블 클릭하면 *Query Builder* 대화창이 열려 쿼리를 편집할 수 있습니다. 쿼리 작성기 대화창은 *Layer* → *Filter*…메뉴를 통해서 열 수도 있습니다.

### 12.1.3 심볼 속성

 *Symbology* 탭은 사용자 벡터 데이터를 렌더링하고 심볼 작업을 하기 위한 종합 도구를 제공합니다. 모든 벡터 데이터에 공통적으로 쓰이는 도구는 물론 서로 다른 벡터 데이터 유형에 맞춰 특화된 심볼 작업 도구도 이용할 수 있습니다. 하지만 모든 데이터 유형이 다음과 같은 대화창 구조를 공유합니다: 상단에는 범주화 및 피처에 적용할 심볼을 준비하는 데 쓰이는 위젯이, 하단에는레이어 렌더링 위젯이 있습니다.

**팁**

**서로 다른 레이어 스타일로 재빨리 바꾸기**

*Layer Properties* 대화창 하단에 있는 *Styles* → *Add* 메뉴를 이용하면, 스타일을 원하는 대로 얼마든지 저장할 수 있습니다. 스타일이란 레이어의 모든 (심볼, 라벨, 도표, 필드 양식, 액션 등등) 속성을 사용자가 원하는대로 조합한 것입니다. 그리고 나서, *Layers Panel* 에 있는 레이어의 컨텍스트 메뉴에서 스타일을 바꾸기만 하면, 사용자 데이터의 모습이 자동적으로 변경될 겁니다.

**팁**

**벡터 심볼 내보내기**

QGIS 벡터 심볼을 구글 `*.kml`, `*.dxf` 및 MapInfo `*.tab` 파일로 내보낼 수 있는 옵션이 존재합니다. 레이어를 오른쪽 클릭해서 컨텍스트 메뉴를 열어, *Save As*…를 선택하고 산출물 파일명 및 포맷을 지정하면 됩니다.

대화창에서는 *Symbology export* 메뉴의 *Feature symbology* 또는 *Symbol layer symbology* 메뉴 옵션을 통해 심볼을 저장하십시오. 심볼 레이어를 사용해본 경험이 있다면, 두 번째 방법을 이용하는 편이 좋습니다.

### 피처 렌더링

렌더링 작업자 (renderer) 는 정확한 심볼과 함께 피처를 그리는 일을 책임집니다. 레이어 도형 유형에 상관없이, 렌더링 작업자에는 단일 심볼, 범주, 등급, 규칙 기반이라는 공통 유형이 4 개 있습니다. 포인트 레이어의 경우, 포인트 변위 (displacement), 포인트 군집 및 열지도 렌더링 작업자를 이용할 수 있는 반면 폴리곤 레이어의 경우 그 외에도 병합된 피처, 역 (inverted) 폴리곤 및 2.5 차원 렌더링 작업자를 통해 렌더링할 수도 있습니다.

연속 색상 렌더링 작업자라는 건 없습니다. 연속 색상 렌더링 작업자란 사실 등급 렌더링 작업자의 특수한 경우일 뿐이기 때문입니다. 심볼 및 색상표를 지정하면 심볼 색상을 적절하게 설정하는 범주 및 등급 렌더링 작업자를 생성할 수 있습니다. 각 데이터 유형 (포인트, 라인 및 폴리곤) 별로, 해당 벡터 심볼 레이어 유형이 존재합니다. 어떤 렌더링 작업자를 선택하느냐에 따라, 대화창에 서로 다른 부분들이 추가될 것입니다.

**참고**

벡터 레이어의 스타일을 설정할 때 렌더링 작업자 유형을 변경해도, 심볼에 대한 사용자 설정은 유지될 것입니다. 다만 단 한 번 변경하는 경우에만 유지된다는 점을 기억해주십시오. 렌더링 작업자 유형을 계속 변경하다보면 심볼 설정이 사라지게 됩니다.

### 단일 심볼 렌더링 작업자

 *Single Symbol* 렌더링 작업자는 레이어의 모든 피처를 사용자가 정의한 단일 심볼을 이용해서 렌더링합니다. 심볼 표현에 대한 자세한 정보는심볼 선택기 를 참조하세요.

### 무 (□) 심볼 렌더링 작업자

 *No Symbols* 렌더링 작업자는 모든 피처를 동일하게 렌더링하는 단일 심볼 렌더링 작업자의 특수한 용례입니다. 이 렌더링 작업자를 이용하면, 피처에 적용된 어떤 심볼도 렌더링하지 않지만 라벨, 도표 및 기타 심볼이 아닌 부분들은 렌더링될 것입니다.

이 렌더링 작업자를 이용하더라도 맵 캔버스에서 레이어에 있는 피처를 선택할 수 있는데, 선택한 피처를 기본 심볼로 렌더링할 것입니다. 편집 중인 피처도 표시할 것입니다.

이 렌더링 작업자는 오직 라벨 또는 도표만 보이길 원하는 레이어를 표시하기 위한 간편한 편법입니다. 다른 방법을 쓴다면 완전히 투명한 면/경계선을 보유한 심볼을 이용해서 렌더링해야만 하겠지요.

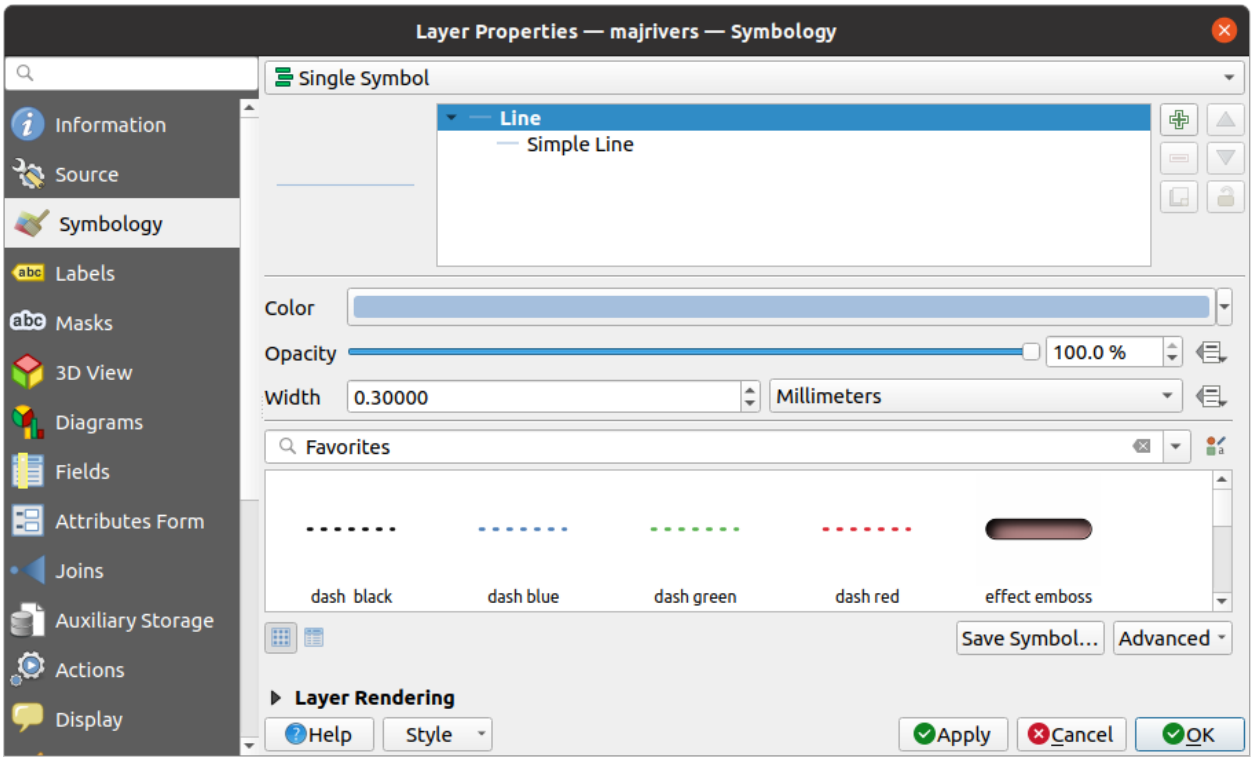



그림 12.3: 단일 심볼 라인 속성

### 범주 렌더링 작업자

*Categorized* 렌더링 작업자는 필드 또는 표현식의 개별 값을 속성에 반영하는 사용자 지정 심볼을 이용해서 레이어의 피처를 렌더링합니다.

레이어에 대해 범주 심볼을 사용하려면:

1. 범주화의 *Value* 를 선택하십시오: 이 값은 기존 필드일 수도 있고, 사용자가 입력란에 입력하거나 또는 관련 ε 버튼으로 작성할 수 있는표현식 일 수도 있습니다. (예를 들어 사용자의 범주화 기준이 하나 이상의 속성으로부터 파생되는 경우) 범주화 작업에 표현식을 이용하면 심볼 범주화만을 위한 필드를 생성하지 않아도 됩니다.

   피처 범주화를 위해 사용되는 표현식은 어떤 종류든 가능합니다. 예를 들어:

   - 비교 표현식이 될 수 있습니다. 이 경우, QGIS 는 `1` (**True**) 이나 `0` (**False**) 을 반환합니다. 다음은 그 예시입니다:

     ```
     myfield >= 100
     $id = @atlas_featureid
     myfield % 2 = 0
     within( $geometry, @atlas_geometry )
     ```

   - 서로 다른 필드를 결합할 수 있습니다:

     ```
     concat( field_1, ' ', field_2 )
     ```

   - 필드들의 값을 계산할 수 있습니다:

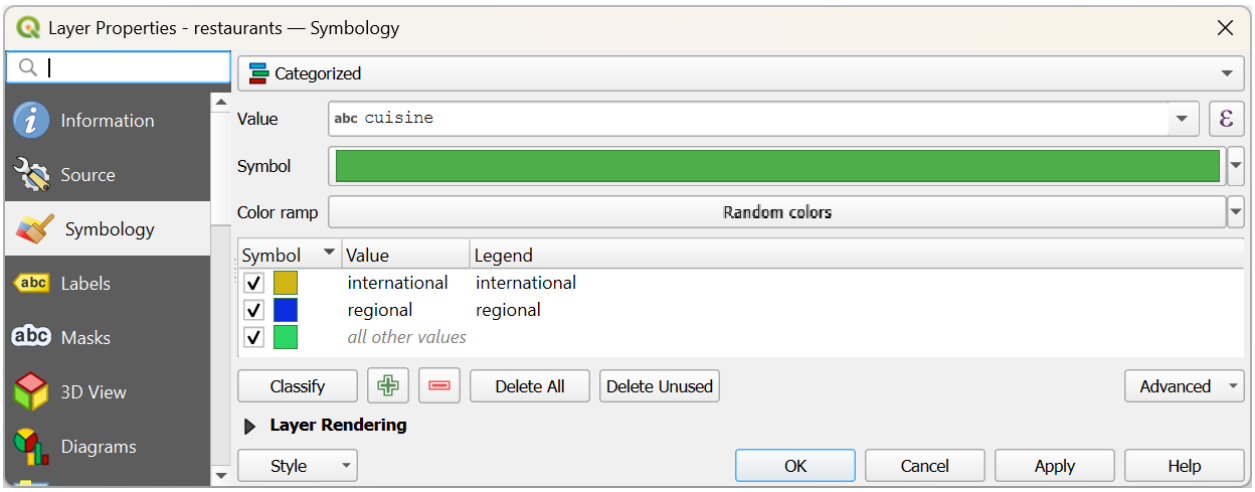



그림 12.4: 범주 심볼 작업 옵션들

```
myfield % 2
year( myfield )
field_1 + field_2
substr( field_1, -3 )
```

- 선형 (linear) 값들을 개별 범주들로 변환시킬 수 있습니다:

```
CASE WHEN x > 1000 THEN 'Big' ELSE 'Small' END
```

- 여러 개별 값들을 단일 범주로 결합할 수 있습니다:

```
CASE
WHEN building IN ('residence', 'mobile home') THEN 'residential'
WHEN building IN ('commercial', 'industrial') THEN 'Commercial and Industrial'
END
```

> **팁**
>
> 객체를 범주화하는 데 모든 종류의 표현식을 사용할 수 있지만, 일부 복잡 표현식의 경우규칙 기반 렌더링 작업자 를 사용하는 편이 더 간단할 수도 있습니다.

2. 모든 범주용 기본 심볼로 쓰이게 될심볼 을 환경 설정하십시오.
3. Indicate the *Color ramp*, i.e. the range of colors from which the color applied to each symbol is selected.

   색상표 위젯 의 공통 옵션 외에, 범주에 *Random Color Ramp* 임의의 색상표를 적용할 수 있습니다. 사용자가 만족하지 못 하는 경우 *Shuffle Random Colors* 를 클릭하면 새로운 임의의 색상 집합을 재생성할 수 있습니다.
4. 그리고 *Classify* 버튼을 클릭해서 주어진 필드나 표현식의 개별 값으로부터 범주들을 생성하십시오.
5. For each class, you can edit the *Legend* column to a more meaningful label (used in the *Layers* panel and the print layout).

> **참고**
>
> Keep in mind that some values may use widgets that do not display the actual value stored in the field. For example, a checkbox widget may store `1` and `0` for checked and unchecked states, while displaying `True` and `False` labels. In this case, to categorize features based on the checkbox state, you need to use the stored values (`1` and `0`) in the expression. QGIS will automatically use the display value for the legend column.

6. 실시간 업데이트 를 사용하는 중이 아니며 맵 캔버스 상의 각 피처를 각 피처의 범주 심볼로 렌더링할 예정인 경우 *Apply* 를 눌러 변경 사항을 적용하십시오.

   QGIS 는 기본적으로 목록에 *all other values* 범주를 추가합니다. 처음 시작 시에는 비어 있지만, 이 범주는 다른 범주에 들어가지 않는 모든 피처를 위한 기본 범주입니다. (예를 들어 범주화 필드/표현식을 위한 새 값을 가진 피처를 생성하는 경우처럼 말이죠.)

기본 범주를 심도 있게 미세 조정할 수 있습니다:

- You can Add new categories, Remove selected categories, *Delete All* of them or *Delete Unused* categories.
- 범주명 왼쪽에 있는 체크박스를 해제하면 각 범주를 비활성화시킬 수 있습니다. 대응하는 피처는 맵 상에서 숨겨집니다.
- 드래그 & 드롭으로 범주의 순서를 조정할 수 있습니다.
- 범주의 심볼, 값 또는 범례를 변경하려면, 해당 범주를 더블 클릭하면 됩니다.

선택한 항목 (들) 을 오른쪽 클릭하면 나오는 컨텍스트 메뉴에는:

- *Copy Symbol* 및 *Paste Symbol*: 항목의 모습을 다른 항목에 적용할 수 있는 편리한 방법입니다.
- *Change Color*…: 선택한 심볼 (들) 의 색상을 변경합니다.
- *Change Opacity*…: 선택한 심볼 (들) 의 투명도를 변경합니다.
- *Change Output Unit*…: 선택한 심볼 (들) 의 산출물 단위를 변경합니다.
- *Change Width*…: 선택한 라인 심볼 (들) 의 폭을 변경합니다.
- *Change Size*…: 선택한 폴리곤 심볼 (들) 의 크기를 변경합니다.
- *Change Angle*…: 선택한 폴리곤 심볼 (들) 의 기울기를 변경합니다.
- *Merge Categories*: 선택한 여러 범주들을 단일 그룹으로 병합합니다. 이렇게 하면 수많은 범주들을 가진 레이어들의 스타일 작업을 좀 더 단순하게 할 수 있습니다. 수많은 개별 범주들을 더 적고 더 관리하기 쉬운 범주 집합으로 그룹화해서 여러 값들에 적용시키는 것도 가능할 것입니다.

> **팁**
>
> 목록에서 선택한 범주 가운데 가장 위에 있는 심볼이 병합된 범주들 용으로 쓰일 심볼이기 때문에, 병합 전에 사용자가 재사용하고자 하는 심볼을 가진 범주를 제일 위로 이동시키는 편이 좋습니다.

- *Unmerge Categories*: 이전에 병합한 범주들을 다시 개별 범주들로 분리합니다.

생성된 범주들은 *Layers* 패널에 있는 트리 위계에도 나타납니다. 레이어 패널에서 항목을 하나 더블클릭하면 할당된 심볼을 편집할 수 있습니다. 항목을 오른쪽 클릭하면더 많은 옵션 에 접근할 수 있습니다.

*Advanced* 메뉴는 범주화 작업의 속도를 향상시키거나 심볼 렌더링을 미세 조정할 수 있는 옵션을 제공합니다:

- *Match to saved symbols*: 각 범주에 범주의 범주화 값을 나타내는 이름을 가진 심볼을심볼 라이브러리 를 사용해서 할당합니다.

- *Match to symbols from file*…: 심볼을 보유한 파일을 지정해서, 각 범주에 범주의 범주화 값을 나타내는 이름을 가진 심볼을 할당합니다.
- 심볼 수준…: 심볼 렌더링 순서를 정의합니다.

### 등급 렌더링 작업자

 *Graduated* 렌더링 작업자는 선택한 피처의 속성이 어떤 범주에 할당되는지를 반영하는 사용자 지정 심볼의 색상 또는 크기를 이용해서 레이어의 모든 피처를 렌더링합니다.

범주 렌더링 작업자처럼, 등급 렌더링 작업자는 지정한 열로부터 심볼의 기울기 및 크기 척도를 설정할 수 있습니다.

또한 범주 렌더링 작업자처럼, 다음 옵션들을 설정할 수 있습니다:

- 범주화의 *Value*: 이 값은 기존 필드일 수도 있고, 사용자가 입력란에 입력하거나 또는 관련  버튼으로 작성할 수 있는표현식 일 수도 있습니다. (예를 들어 사용자의 범주화 기준이 하나 이상의 속성으로부터 파생되는 경우) 범주화 작업에 표현식을 이용하면 심볼 범주화만을 위한 필드를 생성하지 않아도 됩니다.
- 심볼 (심볼 선택기 대화창을 이용합니다.)
- 범례 양식 및 정밀도
- 심볼 변경 방법: 색상 또는 크기
- 색상 방법을 선택했을 경우 색상 (색상표 목록 상자를 이용합니다.)
- 크기 (크기 범위 및 해당 단위를 이용합니다.)

그 다음 할당된 필드 또는 표현식에서 나온 값의 대화형 히스토그램을 표시하는 히스토그램 탭을 이용할 수 있습니다. 히스토그램 위젯을 통해 범주 단계 (class break) 를 이동하거나 추가할 수 있습니다.

> **참고**
>
> 통계 요약 패널을 통해 사용자 벡터 레이어에 관한 정보를 더 많이 얻을 수 있습니다. 통계 요약 패널 을 참조하세요.

범주 탭으로 돌아가면, 범주의 개수는 물론 각 범주에 있는 범주화 기능 모드도 (모드 목록을 이용해서) 설정할 수 있습니다. 다음 모드들을 선택할 수 있습니다:

- 동일 개수 (Equal Count; Quantile): 각 범주가 동일한 개수의 요소를 가지게 됩니다. (상자 수염 그림 개념입니다.)
- 동일 간격 (Equal Interval): 각 범주가 동일한 크기를 가지게 됩니다. (예를 들어 1 에서 16 까지의 값을 범주 4 개로 설정하면, 각 범주의 크기는 4 가 됩니다.)
- 고정 간격 (Fixed Interval): 각 범주가 고정된 값의 범위를 가지게 됩니다. (예를 들어 1 에서 16 까지의 값을 간격 크기 4 로 설정하면, 범주들이 1-4, 5-8, 9-12 그리고 13-16 이 됩니다.)
- 대수 척도 (Logarithmic Scale): 값이 광범위한 데이터에 적합합니다. 적은 값에는 촘촘한 범주를, 큰 값에는 넓은 범주를 할당합니다. (예를 들어 [0..100] 범위의 십진수를 범주 2 개로 설정하면, 처음 범주는 0 에서 10 까지, 다음 범주는 10 에서 100 까지가 될 것입니다.)
- Natural Breaks (Jenks): 각 범주 내부의 분산량은 최소화되는 반면 범주 간의 분산량은 최대화되는 모드입니다.
- Pretty Breaks: x 값의 범위를 커버하면서 동일한 간격으로 보기 좋게 분포된 n+1 개의 일련의 값들을 계산합니다. 10 의 거듭제곱을 1, 2 또는 5 로 곱한 수를 값으로 선택합니다. (R 통계 환경의 pretty 에 기반한 모드입니다.)

- 표준 편차: 값들의 표준 편차에 따라 범주를 생성합니다.

*Symbology* 탭의 가운데에 있는 목록 상자는 범주가 렌더링될 범위, 라벨, 심볼을 포함하는 범주 목록을 담고 있습니다.

**Classify** 버튼을 클릭해서 선택한 모드를 통해 범주를 생성하십시오. 범주명 왼쪽에 있는 체크박스를 해제하면 각 범주를 비활성화시킬 수 있습니다.

범주의 심볼, 값 그리고/또는 라벨을 변경하려면, 그냥 사용자가 원하는 항목을 더블 클릭하면 됩니다.

선택한 항목 (들) 을 오른쪽 클릭하면 나오는 컨텍스트 메뉴에는:

- *Copy Symbol* 및 *Paste Symbol*: 항목의 모습을 다른 항목에 적용할 수 있는 편리한 방법입니다.
- *Change Color*…: 선택한 심볼 (들) 의 색상을 변경합니다.
- *Change Opacity*…: 선택한 심볼 (들) 의 투명도를 변경합니다.
- *Change Output Unit*…: 선택한 심볼 (들) 의 산출물 단위를 변경합니다.
- *Change Width*…: 선택한 라인 심볼 (들) 의 폭을 변경합니다.
- *Change Size*…: 선택한 폴리곤 심볼 (들) 의 크기를 변경합니다.
- *Change Angle*…: 선택한 폴리곤 심볼 (들) 의 기울기를 변경합니다.

그림 12.5 는 QGIS 예시 데이터셋의 `major_rivers` 레이어에 대한 등급 렌더링 작업자 대화창의 예시입니다.

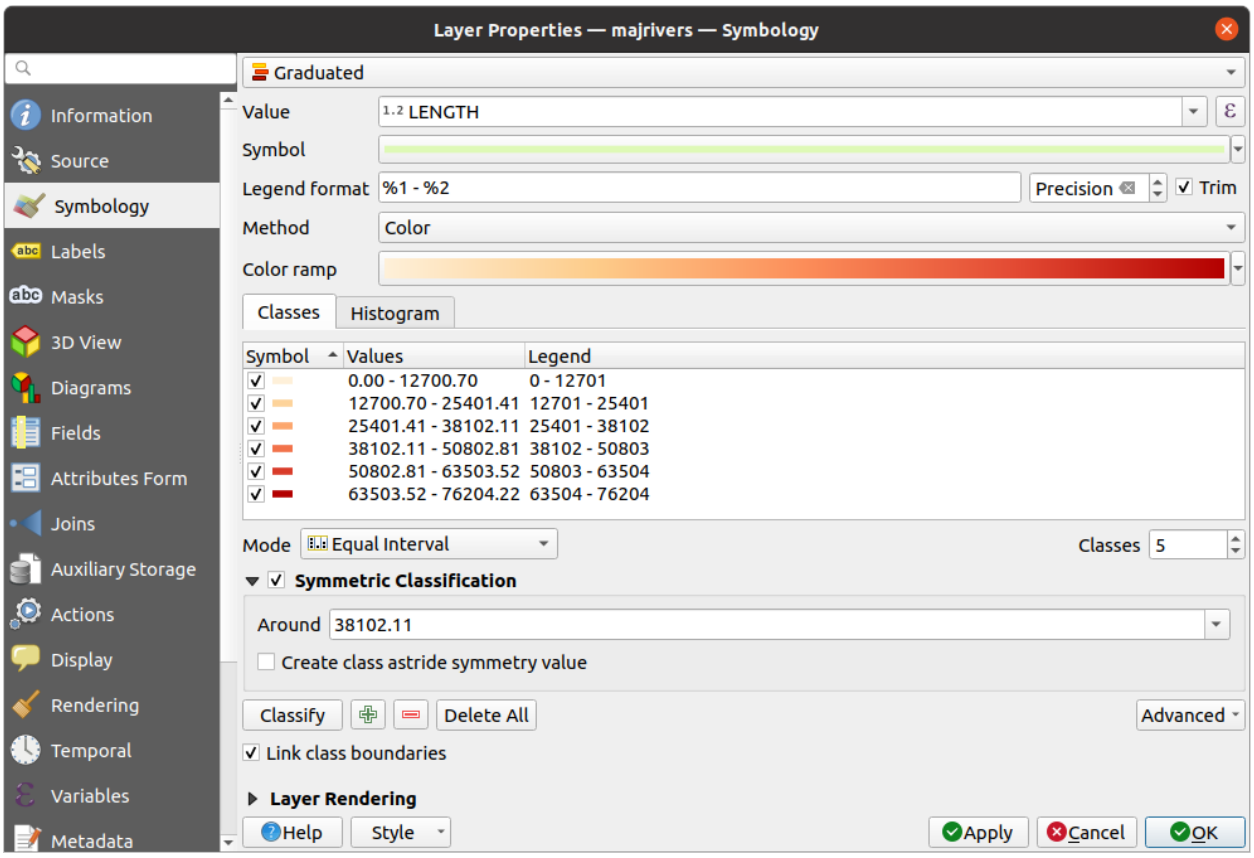



그림 12.5: 등급 심볼 작업 옵션들

생성된 범주들은 *Layers* 패널에 있는 트리 위계에도 나타납니다. 레이어 패널에서 항목을 하나 더블클릭하면 할당된 심볼을 편집할 수 있습니다. 항목을 오른쪽 클릭하면더 많은 옵션 에 접근할 수 있습니다.

### 비례 심볼 및 다변량 분석

비례 심볼 및 다변량 분석은 심볼 렌더링 작업 드롭다운 목록에서 선택할 수 있는 렌더링 유형이 아닙니다. 하지만 앞에서 소개한 렌더링 옵션에 적용된데이터 정의 무시 옵션을 통해, QGIS 는 사용자의 포인트 및 라인 데이터를 비례 심볼로 표현할 수 있습니다.

#### 비례 심볼 생성하기

비례 렌더링을 적용하려면:

1. 먼저 레이어에단일 심볼 렌더링 작업자 를 적용합니다.
2. 그리고 피처에 적용할 심볼을 설정합니다.
3. 심볼 트리의 상단에 있는 항목을 선택한 다음, (포인트 레이어의) *Size* 또는 (라인 레이어의) *Width* 옵션 옆에 있는 Data-defined override 버튼 을 클릭하십시오.
4. 필드를 선택하거나 표현식을 입력하면, QGIS 가 각 피처별로 속성에 산출 값을 적용하고 맵 캔버스에 있는 심볼의 크기를 비례에 맞춰 재조정할 것입니다.

   필요하다면 메뉴의 *Size assistant*…옵션을 사용해서 심볼 크기 재조정에 몇몇 (지수, 플래너리 등등) 변환을 적용하십시오. (자세한 내용은데이터 정의 어시스턴트 인터페이스 사용하기 를 참조하세요.)

레이어 패널 과인쇄 조판기 범례 항목 에 비례 심볼을 표시하도록 선택할 수 있습니다: *Symbology* 탭의 주 대화창 하단에 있는 *Advanced* 드롭다운 목록을 펼치고 **Data-defined size legend**…를 선택하면 범례 항목을 환경 설정할 수 있습니다. (자세한 내용은데이터 정의 크기 범례 를 참조하세요.)

#### 다변량 분석 생성하기

다변량 분석 렌더링은 2 개 이상의 변수들 사이의 관계를 평가할 수 있습니다. 예를 들면 변수 하나를 크기로 표현하고, 다른 변수를 색상표로 표현할 수 있습니다.

QGIS 에서 다변량 분석을 생성하는 가장 간단한 방법은 다음과 같습니다:

1. 먼저 모든 범주에 동일한 심볼 유형을 이용해서 레이어 상에 범주 또는 등급 렌더링을 적용하십시오.
2. 그 다음 범주에 대해 비례 심볼을 적용하십시오:
   1. 범주화 프레임 상단에 있는 *Change* 버튼을 클릭하면심볼 선택기 대화창이 열립니다.
   2. 앞에서 설명한 대로 데이터 정의 무시 위젯을 통해 심볼 레이어의 크기 또는 너비 척도를 재조정하십시오.

비례 심볼과 마찬가지로, 데이터 정의 크기 범례 를 사용해서 레이어 트리에 있는 범주 또는 등급 범주 심볼 위에 척도를 조정한 심볼을 추가할 수 있습니다. 인쇄 조판기 범례 항목에서도 이 두 표현을 모두 쓸 수 있습니다.

### 규칙 기반 렌더링 작업자

규칙이란 피처에 특정 렌더링 설정을 적용하기 위해 피처의 속성 (attribute) 또는 속성 (property) 에 따라 피처를 구분하는 데 쓰이는 QGIS 표현식 을 말합니다. 규칙은 내포될 수 있으며, 피처가 모든 상위 내포 수준 (들) 에 속하는 경우 피처는 범주에 속합니다.

따라서 *Rule-based* 렌더링 작업자는 선택한 피처를 세분화된 범주에 할당하는지를 반영하는 측면을 가진 심볼을 이용해서 레이어의 모든 피처를 렌더링하도록 설계되었습니다.

규칙을 생성하려면:

1. 기존 행을 더블클릭해서 활성화시키거나 (QGIS 는 기본적으로 렌더링 모드가 활성화될 때 규칙을 가지지 않은 심볼을 추가합니다) 또는 Edit rule 또는 Add rule 버튼을 클릭하십시오.

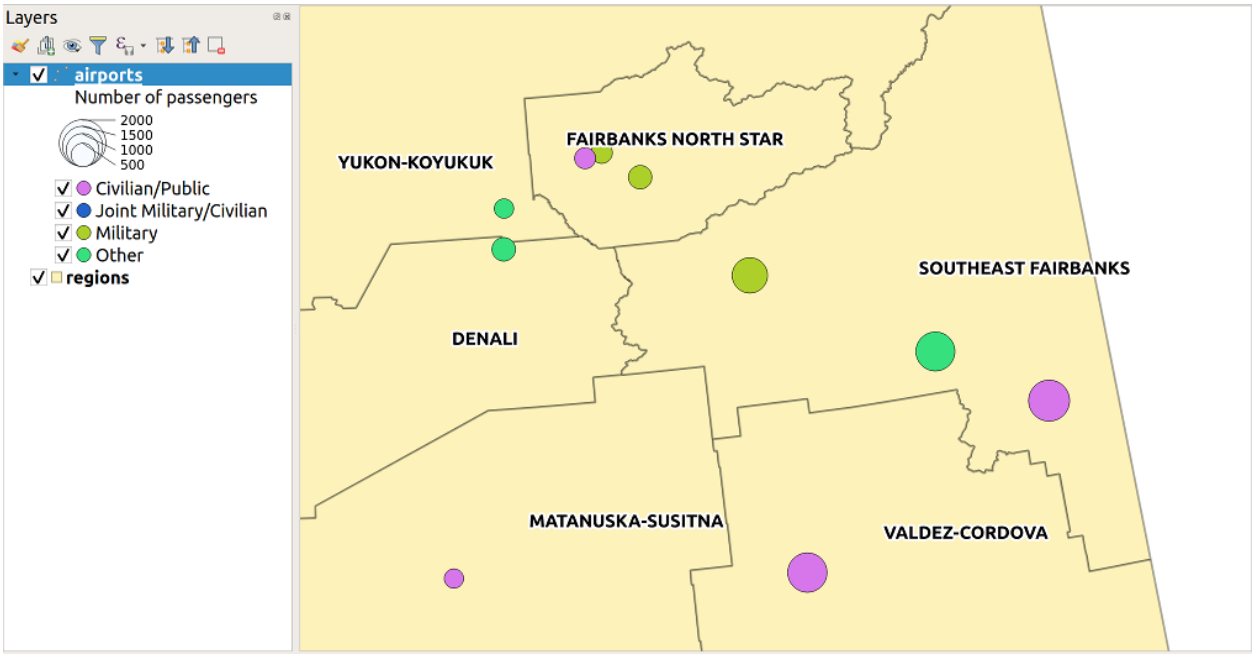



그림 12.6: 크기 척도를 조정한 범례를 가진 다변량 분석의 예시

2. *Edit Rule* 대화창이 열리면 사용자가 각 규칙을 쉽게 식별하도록 해주는 라벨을 정의할 수 있습니다. *Layers Panel* 은 물론 인쇄 작성자 범례도 이 라벨을 표시할 것입니다.
3. *Filter* 옵션 옆에 있는 텍스트란에 표현식을 직접 입력하거나 그 옆에 있는 버튼을 클릭해서 표현식 문자열 작성자 대화창을 여십시오.
4. 제공되는 함수들과 레이어 속성을 사용해서 사용자가 가져오고 싶은 피처를 필터링하기 위한표현식 을 작성하십시오. 쿼리 결과를 확인하려면 *Test* 버튼을 클릭하십시오.
5. 규칙 설명을 완전하게 하려면 더 긴 라벨을 입력할 수 있습니다.
6. *Scale Range* 옵션을 사용해서 규칙을 적용해야 할 축척을 설정할 수 있습니다.
7. 마지막으로 이 피처들에사용할 심볼 을 환경 설정하십시오.
8. *OK* 를 클릭하십시오.

규칙을 요약한 새로운 행이 레이어 속성 대화창에 추가됩니다. 앞에서 설명한 단계를 따라 필요한 만큼 많은 규칙을 생성할 수도 있고 기존 규칙을 복사해서 붙여넣을 수도 있습니다. 규칙을 내포시키고 하위 범주에서 상위 규칙 피처들을 개선하려면 규칙을 다른 규칙 위로 드래그 & 드롭하십시오.

규칙 기반 렌더링 작업자를 범주 또는 등급 렌더링 작업자와 결합시킬 수 있습니다. 규칙을 하나 선택하면, *Refine selected rules* 드롭다운 메뉴를 이용해서 하위 범주에 해당 규칙이 적용된 피처를 정리할 수 있습니다. 세분화된 범주는 트리 계층에 규칙의 하위 항목처럼 나타나며, 상위 범주와 마찬가지로 사용자가 각 범주의 심볼과 규칙을 설정할 수 있습니다. 다음을 기반으로 규칙 세분화를 자동화할 수 있습니다:

- **축척 (scales)**: 이 옵션은 축척 목록을 지정해서 서로 다른 사용자 정의 축척 범위가 적용되는 범주 집합을 생성합니다. 각각의 새로운 축척 기반 범주가 고유한 심볼과 정의 (definition) 표현식을 가질 수 있습니다. 축척 기반 범주는 예를 들어 동일한 피처를 서로 다른 축척에서 서로 다른 심볼로 표시하거나, 또는 축척에 따라 특정 피처 집합만 표시하는 데 (예를 들면 대축척에서 지방 공항을 표시하고 소축척에서는 국제 공항을 표시하는 데) 편리한 방법입니다.
- **범주 (categories)**: 선택한 규칙에 해당하는 피처에범주 렌더링 작업자 를 적용합니다.
- 또는 **범위 (ranges)**: 선택한 규칙에 해당하는 피처에등급 렌더링 작업자 를 적용합니다.

세분화된 범주는 트리 계층에 규칙의 하위 항목처럼 나타나며, 상위 범주와 마찬가지로 사용자가 각 범주의 심볼을 설정할 수 있습니다. 내포된 규칙의 심볼은 층층이 쌓이기 때문에 신중하게 선택하십시오. 이 층층이 쌓인 심볼 가운데 특정 심볼의 렌더링을 방지하기 위해 *Edit rule* 대화창에서 *Symbols* 를 체크 해제할 수도 있습니다.

*Edit rule* 대화창에서, 모든 규칙을 직접 작성하지 않고서도 동일 수준에서 어떤 규칙과도 일치하지 않는 모든 피처를 걸러내는 *Else* 옵션을 사용할 수 있습니다. *Layer Properties* → *Symbology* → *Rule-based* 대화창의 규칙 *(Rule)* 열에 `Else` 를 입력해도 동일한 결과를 얻을 수 있습니다.

선택한 항목 (들) 을 오른쪽 클릭하면 나오는 컨텍스트 메뉴에는:

- *Copy* 와 *Paste*: 기존 항목 (들) 을 기반으로 새 항목 (들) 을 생성할 수 있는 편리한 방법입니다.
- *Copy Symbol* 및 *Paste Symbol*: 항목의 모습을 다른 항목에 적용할 수 있는 편리한 방법입니다.
- *Change Color*…: 선택한 심볼 (들) 의 색상을 변경합니다.
- *Change Opacity*…: 선택한 심볼 (들) 의 투명도를 변경합니다.
- *Change Output Unit*…: 선택한 심볼 (들) 의 산출물 단위를 변경합니다.
- *Change Width*…: 선택한 라인 심볼 (들) 의 폭을 변경합니다.
- *Change Size*…: 선택한 폴리곤 심볼 (들) 의 크기를 변경합니다.
- *Change Angle*…: 선택한 폴리곤 심볼 (들) 의 기울기를 변경합니다.
- *Refine Current Rule*: 현재 규칙을 **축척**, **범주** 또는 **범위** 로 미세 조정할 수 있는 하위 메뉴를 엽니다. 대화창 하단에서대응하는 메뉴 를 선택하는 것과 동일합니다.

규칙 기반 렌더링 작업자 대화창에서 한 행을 체크 해제하면 맵 캔버스에서 특정 규칙과 내포된 규칙을 가진 피처를 숨깁니다.

생성된 규칙들은 맵 범례에 있는 트리 위계에도 나타납니다. 맵 범례에서 항목을 하나 더블클릭하면 할당된 심볼을 편집할 수 있습니다. 항목을 오른쪽 클릭하면더 많은 옵션 에 접근할 수 있습니다.

그림 12.7 은 QGIS 예시 데이터셋의 `rivers` 레이어에 대한 규칙 기반 렌더링 작업자 대화창의 예시입니다.

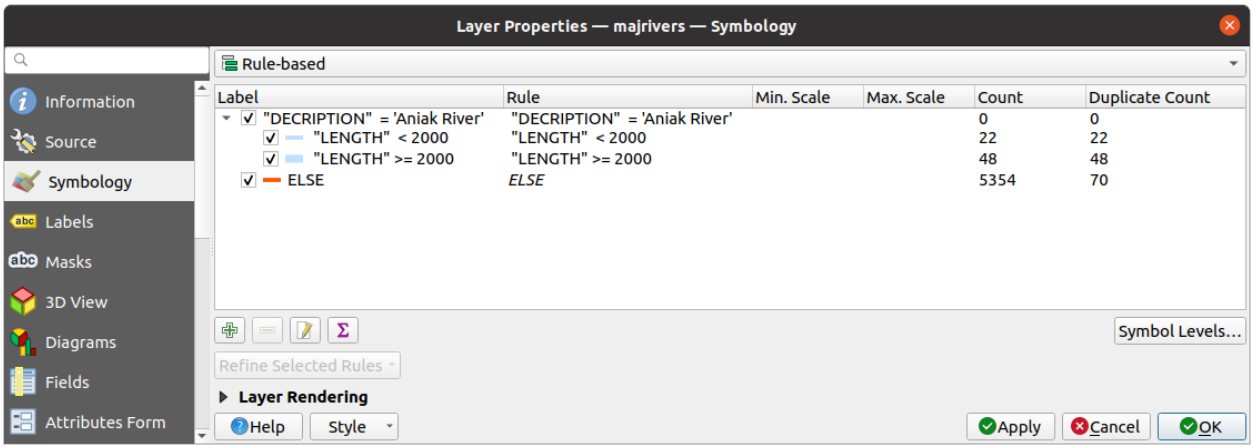



그림 12.7: 규칙 기반 심볼 작업 옵션들

### 포인트 변위 렌더링 작업자

 *Point Displacement* 렌더링 작업자는 서로 지정한 허용 오차 거리에 들어오는 포인트 피처들을 받아서, 다양한 배치 방법에 따라 그 무게중심 (barycenter) 주위에 심볼들을 배치합니다. 이 렌더링 작업자는 (예를 들어 같은 건물 안에 있는 오락시설들처럼) 포인트 레이어의 피처들이 동일한 위치에 존재하더라도 모두 가시화시킬 수 있는 편리한 방법이 될 수 있습니다.

포인트 변위 렌더링 작업자를 환경 설정하려면, 다음 작업을 해줘야 합니다:

1. *Center symbol*: 중심에 있는 가상 포인트가 어떻게 보일지를 결정하는 중심 심볼을 설정합니다.
2. *Renderer*: 레이어의 피처를 어떻게 분류할지를 결정하는 (단일, 범주, 규칙 기반 등등의) 렌더링 작업자 유형을 선택합니다.
3. *Renderer Settings*…버튼을 눌러서 선택한 렌더링 작업자에 따라 피처의 심볼을 환경 설정하십시오.
4. *Distance*: 서로 근접한 피처들을 중첩해 있다고 간주하고 동일한 가상 포인트 주위로 이동시킬 허용 오차 거리를 설정하십시오. 흔히 쓰이는 심볼 단위들을 지원합니다.
5. *Placement methods*: 배치 방법을 환경설정합니다:
   - **고리 (Ring)**: 표시할 피처 개수에 따라 반경이 달라지는 원 위에 모든 피처를 배치합니다.
   - **동심 고리 (Concentric rings)**: 동심원 집합을 이용해서 피처를 표시합니다.
   - **그리드 (Grid)**: 각 교차점에 포인트 심볼을 가진 정규 그리드를 생성합니다.
6. 이동된 심볼은 *Displacement lines* 상에 배치됩니다. 변위 라인 (displacement line) 의 최소 간격은 포인트 심볼 렌더링 작업자의 설정을 따르지만, 그래도 *Stroke width*, *Stroke color* 및 *Size adjustment* 와 같은 일부 설정을 사용자 지정할 수 있습니다. (예를 들어 이런 사용자 지정 설정을 통해 렌더링된 포인트 사이의 간격을 늘릴 수 있습니다.)
7. *Labels* 그룹 옵션을 이용해서 포인트 라벨 작업을 수행할 수 있습니다. 라벨은 피처의 실제 위치가 아니라 이동된 심볼 근처에 배치됩니다.
   1. *Label attribute*: 라벨 작업에 사용할 레이어의 필드를 선택합니다.
   2. *Label font*: 라벨 글꼴의 속성 및 크기를 설정합니다.
   3. *Label color*: 색상을 선택합니다.
   4. *Label distance factor*: 각 포인트 피처에 대해, 심볼 중심으로부터 심볼의 대각선 크기에 비례해서 라벨을 오프셋시킬 거리 인자를 설정합니다.
   5. ☐ *Use scale dependent labeling*: 지정한 *Minimum map scale* 을 초과하는 축척에서만 라벨을 표시하고자 하는 경우 이 옵션을 체크하십시오.

> **참고**
>
> 포인트 변위 렌더링 작업자는 피처 도형을 변경하지 않습니다. 즉 포인트가 제자리를 벗어나지 않는다는 뜻입니다. 포인트들은 초기 위치에 그대로 자리합니다. 그저 렌더링 목적을 위해 시각적으로만 바뀔 뿐입니다. 변위된 피처를 생성하고자 하는 경우포인트 변위시키기 공간 처리 알고리즘을 대신 사용하십시오.

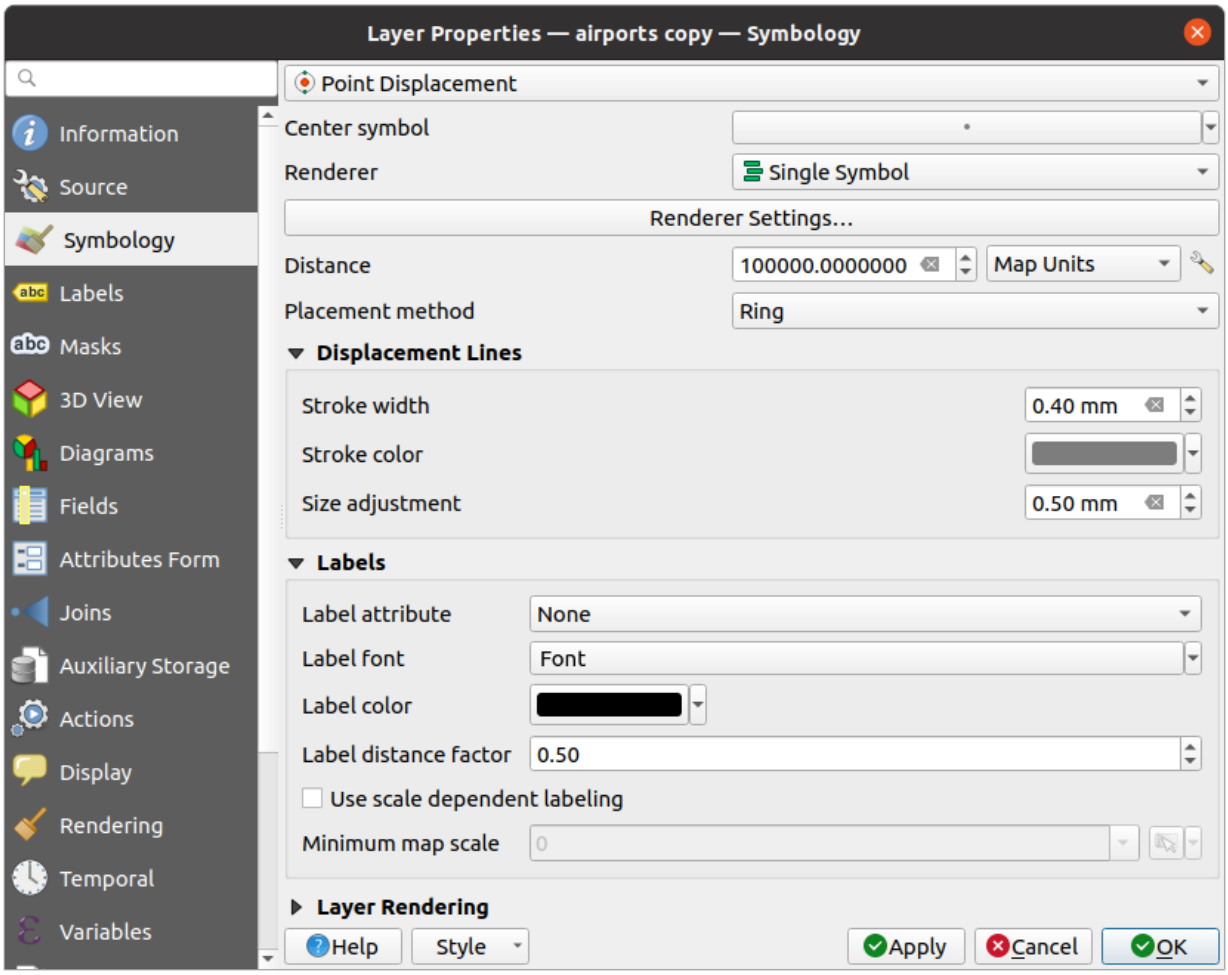



그림 12.8: 포인트 변위 대화창

### 포인트 군집 렌더링 작업자

최근접 또는 중첩 포인트 피처 배치를 과장하는 *Point Displacement* 렌더링 작업자와는 달리, *Point Cluster* 렌더링 작업자는 근접 포인트들을 단일하게 렌더링된 마커 심볼 하나로 그룹화합니다. 서로 특정 거리 안에 있는 포인트들을 단일 심볼로 병합하는 것이죠. 단순히 검색 거리 안에 있는 첫번째 그룹에 할당하기 보다는, 형성되는 그룹 가운데 가장 가까운 그룹을 기반으로 포인트 집합체를 생성합니다.

주 대화창에서 다음과 같은 작업을 할 수 있습니다:

1. *Cluster symbol* 에서 포인트 군집을 표현하기 위한 심볼을 설정할 수 있습니다. 기본 렌더링은 글꼴 마커 심볼 레이어 상의 `@cluster_size` 변수 의 도움으로 합쳐진 피처들의 개수를 표시합니다.
2. *Renderer*: 예를 들어 레이어의 피처를 어떻게 분류할지를 결정하는 (단일, 범주, 규칙 기반 등등의) 렌더링 작업자 유형을 선택할 수 있습니다.
3. *Renderer Settings*…버튼을 누르면 평소와 같이 피처의 심볼을 환경 설정할 수 있습니다. 이 심볼은 군집화되지 않은 피처에만 표시되며, 군집화된 피처에는 *Cluster symbol* 이 적용된다는 사실을 기억하십시오. 또한 군집에 있는 모든 포인트 피처가 동일한 렌더링 범주에 속해 있기 때문에 동일한 색상이 적용될 경우, 이 색상은 군집의 `@cluster_color` 변수를 표현합니다.
4. *Distance*: 피처들이 군집하는지 판단하는 데 고려할 최대 거리를 설정할 수 있습니다. 흔히 쓰이는 심볼 단위들을 지원합니다.

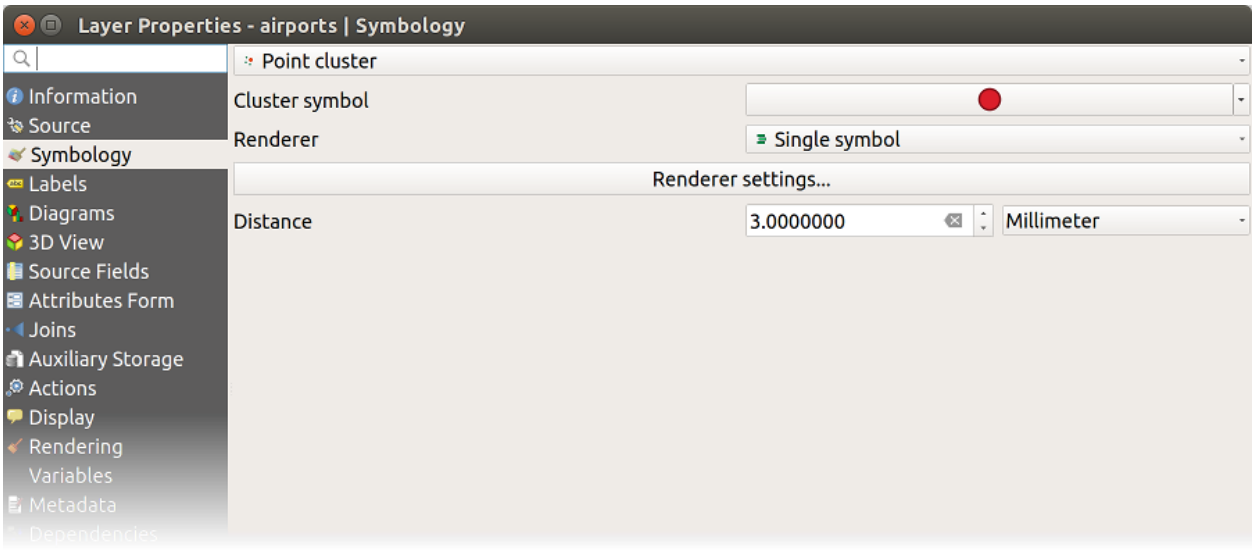



그림 12.9: 포인트 군집 대화창

**참고**

포인트 군집 렌더링 작업자는 피처 도형을 변경하지 않습니다. 즉 포인트가 제자리를 벗어나지 않는다는 뜻입니다. 포인트들은 초기 위치에 그대로 자리합니다. 그저 렌더링 목적을 위해 시각적으로만 바뀔 뿐입니다. 군집 기반 피처를 생성하고자 하는 경우*k*-평균 군집 형성 또는*DBSCAN* 군집 형성 공간 처리 알고리즘을 대신 사용하십시오.

### 병합 피처 렌더링 작업자

*Merged Features* 렌더링 작업자는 복잡 심볼 또는 중첩하는 피처들을 균일하고 연속적인 지도 제작 심볼로 표현하기 위해 렌더링 전에 면 (area) 과 선 (line) 피처들을 단일 객체로"융합 (dissolve)"시킬 수 있습니다.

### 반전 폴리곤 렌더링 작업자

*Inverted Polygon* 렌더링 작업자는 레이어에 있는 폴리곤의 바깥을 채울 심볼을 지정할 수 있습니다. 앞에서와 마찬가지로 단일 심볼, 등급, 범주, 규칙 기반 또는 2.5D 같은 하위 렌더링 작업자를 선택할 수 있습니다.

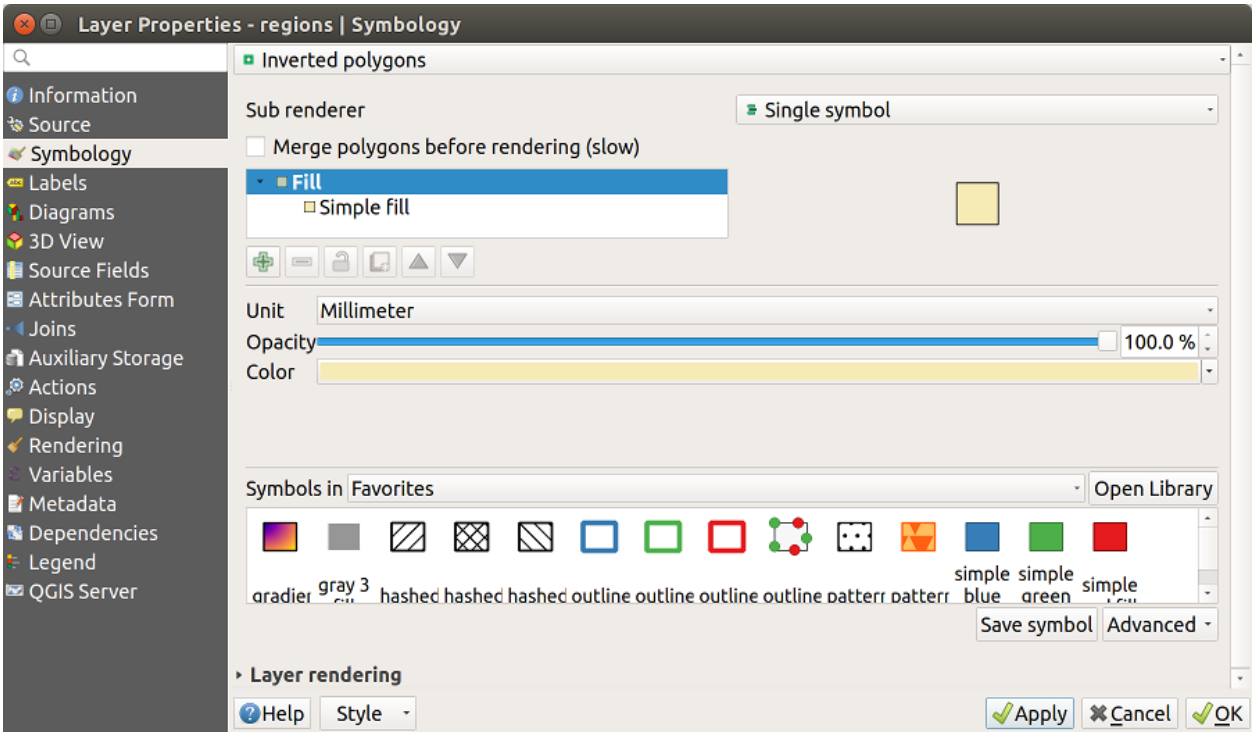



그림 12.10: 반전 폴리곤 대화창

### 열지도 렌더링 작업자

With the *Heatmap* renderer you can create live dynamic heatmaps for (multi)point layers. You can specify the heatmap *Radius* in millimeters, points, pixels, map units or inches, choose and edit a *Color ramp* for the heatmap style and use a slider for selecting a trade-off between render speed and quality. You can also define a *Maximum value* limit and *Weight points by* using a field or an expression.

Use Data defined override to dynamically control *Radius* and *Maximum value* based on the attributes of your data. For example, the radius of a heatmap point could be determined by its population attribute, or the maximum value could be based on a temporal range.

When adding or removing a feature the heatmap renderer updates the heatmap style automatically. The *Color ramp* will be shown as a legend bar and in the *Legend settings* you can set the *Labels* for the *Maximum* and *Minimum* values. You can also change the orientation and direction of the legend in the *Layout*.

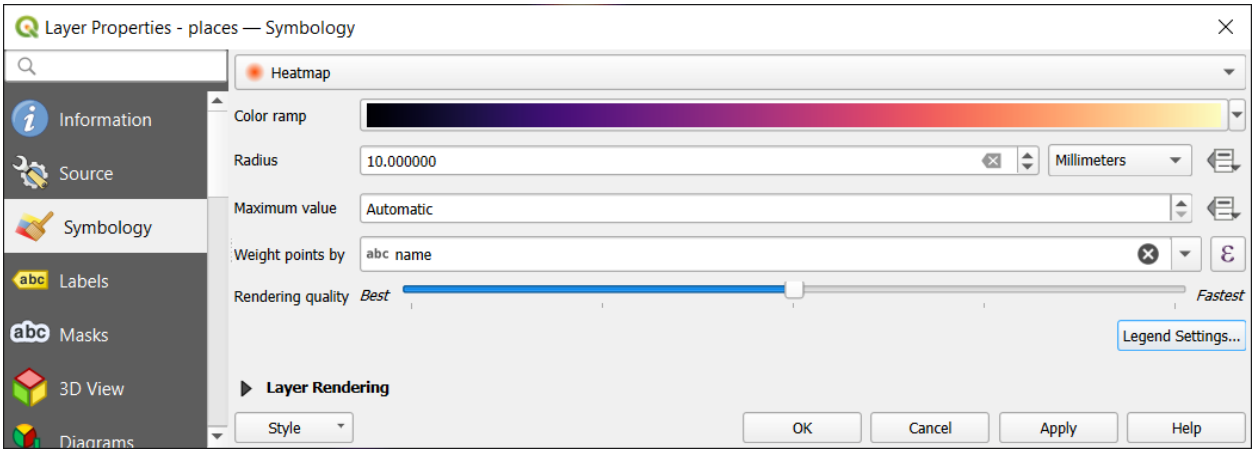



그림 12.11: 열지도 대화창

### 2.5D 렌더링 작업자

*2.5D* 렌더링 작업자를 이용해서 사용자 레이어의 피처에 2.5D 효과를 줄 수 있습니다. 먼저 *Height* 값 (맵 단위)을 설정하십시오. 고정값, 사용자 레이어의 필드 가운데 하나, 또는 표현식으로 설정할 수 있습니다. 또 시각의 방향을 (0° 는 서쪽으로, 값이 올라갈수록 반시계 방향으로 돕니다) 재현하려면 *Angle* (도 단위) 을 설정해야 합니다. *Roof Color* 및 *Wall Color* 을 설정하려면 고급 환경 설정 옵션을 사용하십시오. 만약 피처의 벽에 태양광 효과를 주고 싶다면, *Shade walls based on aspect* 옵션을 체크하도록 하십시오. *Color* 및 *Size* 값 (맵 단위) 을 설정하면 그림자 효과를 줄 수도 있습니다.

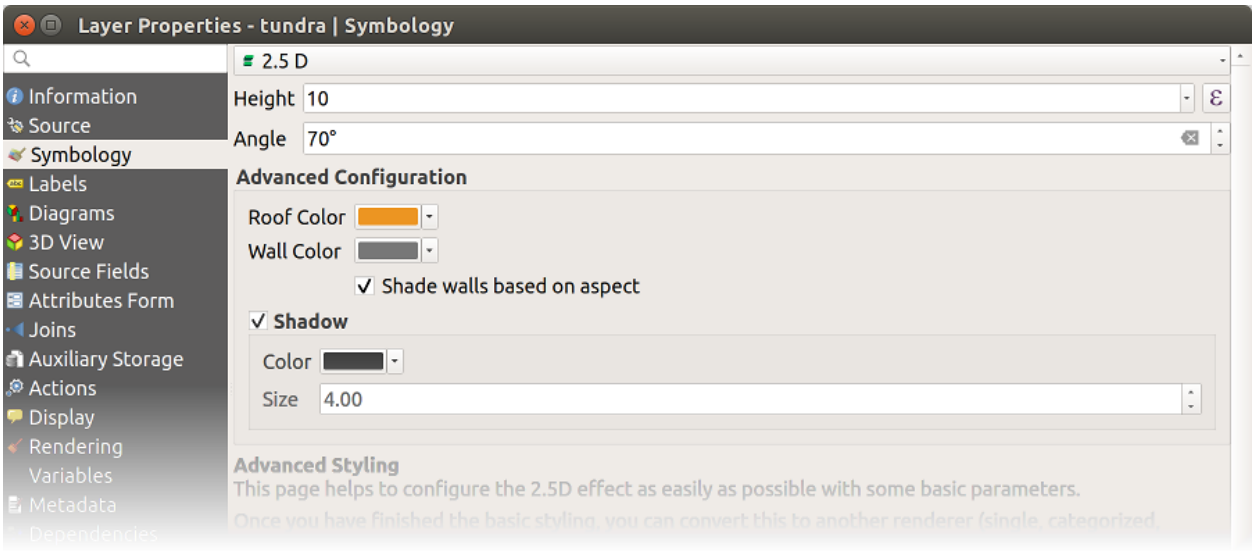



그림 12.12: 2.5D 대화창

> **팁**
>
> **다른 렌더링 작업자에서 2.5D 효과 이용하기**

2.5D 렌더링 작업자에서 기본 스타일 설정을 마치고 나면, 다른 (단일 심볼, 범주, 등급) 렌더링 작업자로 변환시킬 수 있습니다. 2.5D 효과가 지속되는 동시에 다른 렌더링 작업자의 모든 특정 옵션도 쓸 수 있어 2.5D 효과를 정밀하게 조정할 수 있습니다. (예를 들어 멋진 2.5D 표현으로 범주 심볼을 그리거나 2.5D 심볼에 몇몇 기타 스타일을 추가할 수도 있습니다.) 그림자 및"건물"자체가 주변의 다른 피처를 가리지 않도록 하려면, 심볼 수준 (*Advanced* ▸ *Symbol levels*…) 을 활성화해야 할 수도 있습니다. 2.5D 높이 및 각도 값은 레이어의 변수에 저장되기 때문에, 이후 레이어 속성 대화창의 변수 탭에서 편집할 수 있습니다.

### 내장 렌더링 작업자

*Embedded Symbols* 렌더링 작업자는 제공된 데이터소스의'자체 (native)'심볼을 표시할 수 있습니다. 주로 사전 정의된 심볼을 가지고 있는 KML 및 TAB 데이터셋이 이에 해당합니다.

### 레이어 렌더링

심볼 탭에서 레이어의 모든 피처에 대해 일괄적으로 적용되는 다음 옵션들을 설정할 수도 있습니다:

- *Opacity*  : 이 도구를 통해 맵 캔버스에서 아래에 있는 레이어를 가시화할 수 있습니다. 슬라이드 바를 통해 사용자 벡터 레이어의 가시성을 필요에 따라 조정하십시오. 슬라이드 바 옆에 있는 메뉴에서 가시성을 정확한 백분율로 설정할 수도 있습니다.
- *Layer* 및 *Feature* 수준의 *Blending mode*: 이 도구들을 통해 사용자가 그래픽 프로그램에서만 가능하다고 알고 있던 특별한 렌더링 효과를 줄 수 있습니다. 위 그리고 아래에 있는 레이어의 픽셀을혼합 모드 에서 설명한 설정으로 혼합합니다.
- *Draw Effects* 버튼: 레이어의 모든 피처에 대해그리기 효과 를 줍니다.
- *Control feature rendering order*: 피처 속성을 이용해서 피처를 어떤 순서로 렌더링해야 하는지에 대한 Z 순서를 정의할 수 있습니다. 체크박스를 체크한 다음 옆에 있는  버튼을 클릭하십시오. *Define Order* 대화창이 열리는데, 다음 작업들을 할 수 있습니다:

  1. 레이어 피처에 적용시킬 필드를 선택하거나 표현식을 작성합니다.
  2. 불러온 피처들을 어떤 순서로 배열할지 설정합니다. 예를 들어 **오름차순 (Ascending)** 을 선택했다면, 높은 값을 가진 피처 아래에 낮은 값을 가진 피처를 렌더링합니다.
  3. NULL 값을 반환하는 피처를 언제, 즉 **맨 처음** (아래) 또는 **마지막** (위) 으로 렌더링할지 정의합니다.
  4. 사용자가 사용하고자 하는 규칙에 따라 앞의 단계를 얼마든지 반복하십시오.

  첫 번째 규칙은 레이어에 있는 모든 피처에 적용돼, 반환 값에 따라 피처들을 Z 순서로 배열합니다. 그 다음 규칙은 동일한 (NULL 값 포함) 값을 가진, 즉 동일한 Z 수준의 각 피처 그룹에 적용돼 각 그룹에 있는 항목들을 배열합니다. 그 다음 규칙은 다시…

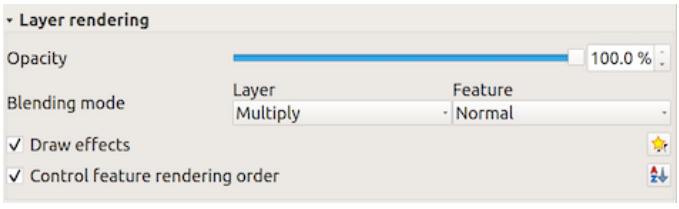



그림 12.13: 레이어 렌더링 옵션들

### 기타 설정

#### 심볼 수준

적층 (□□) 심볼 레이어를 지원하는 (열지도 제외) 렌더링 작업자의 경우 각 심볼 수준의 렌더링 순서를 조정하는 옵션이 존재합니다.

대부분의 렌더링 작업자의 경우, 저장된 심볼 목록 하단에 있는 *Advanced* 버튼을 클릭한 다음 *Symbol levels* 를 선택하면 심볼 수준 옵션에 접근할 수 있습니다. 규칙 기반 렌더링 작업자 의 경우 *Symbols Levels*…버튼으로 이 옵션에 직접 접근할 수 있는 반면, 포인트 변위 렌더링 작업자 의 경우 *Rendering settings* 대화창에 동일한 버튼이 있습니다.

심볼 수준을 활성화하려면, *Enable symbol levels* 를 체크하십시오. 복합 심볼, 라벨, 그리고 열로 나누어진 개별 심볼 레이어의 작은 표본이 번호와 함께 각 행에 표시됩니다. 이 번호들은 심볼 레이어를 그릴 렌더링 순서 수준을 나타냅니다.  번호 숫자가 낮을수록 먼저 그려져 아래에 남고, 높을수록 나중에 그려져 다른 레이어 위에 있게 됩니다.

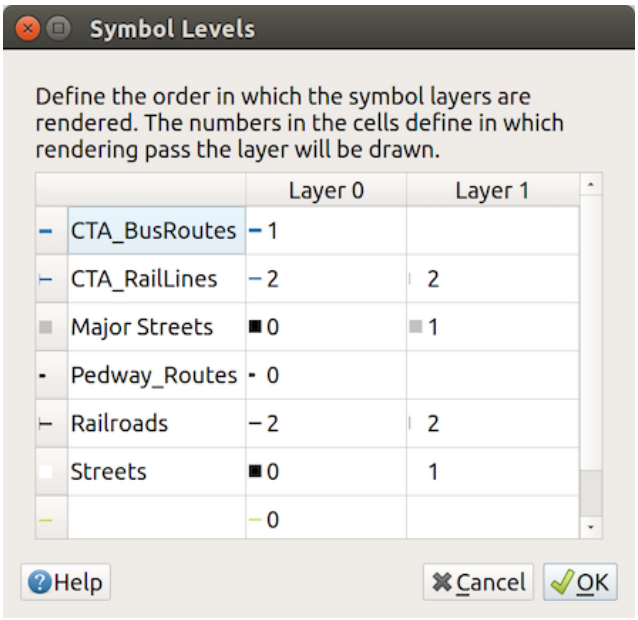



그림 12.14: 심볼 수준 대화창

> **참고**
>
> 심볼 수준을 비활성화하면, 심볼 각각의 피처 순서에 따라 완전한 심볼을 그립니다. 위에 있는 심볼은 아래 있는 심볼을 혼란스럽게 만들 뿐입니다. 게다가, 비슷한 심볼들이 서로"병합"되지 않을 것입니다.

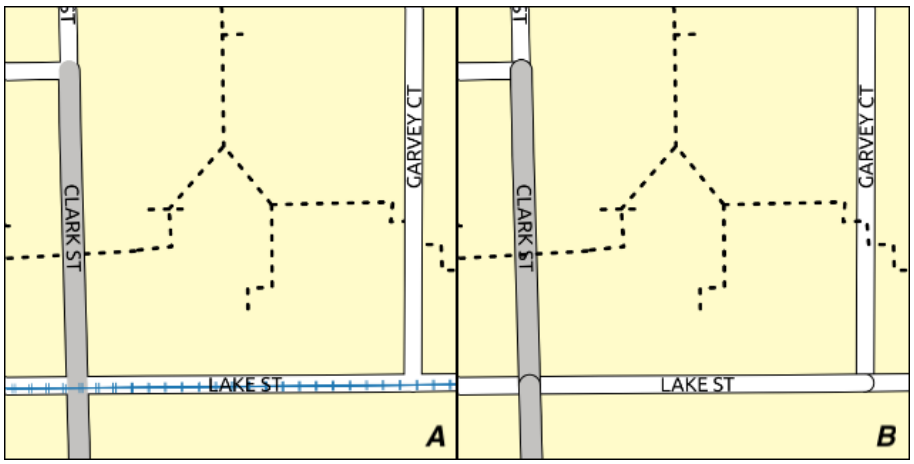



그림 12.15: 심볼 수준 (A) 활성화 (B) 비활성화 비교

### 데이터 정의 크기 범례

레이어를비례 심볼 또는 다변량 렌더링 으로 렌더링하거나 레이어에척도를 적용한 크기 도표 를 적용한 경우, 레이어 패널 과인쇄 조판기 범례 둘 다에 척도를 적용한 심볼을 표시하도록 할 수 있습니다.

*Data-defined Size Legend* 대화창을 활성화시켜 심볼을 렌더링하려면, 저장된 심볼 목록 아래 있는 *Advanced* 버튼을 눌러 해당 옵션을 선택하십시오. 도표의 경우, *Legend* 탭에서 해당 옵션을 사용할 수 있습니다. 이 대화창에서 다음과 같은 옵션을 설정할 수 있습니다:

- 범례 유형을 선택합니다: *Legend not enabled*, *Separated legend items* 그리고 *Collapsed legend* 가운데 하나를 선택할 수 있습니다. 마지막 옵션을 선택했다면 범례 항목을 **하단 (Bottom)** 정렬할지 **중앙 (Center)** 정렬할지 선택할 수 있습니다.
- 범례 표현에사용할 심볼 을 미리보기할 수 있습니다.
- 범례에 제목을 삽입합니다.
- 사용할 범주의 크기를 재조정합니다: QGIS 는 기본적으로 범례의 범주를 (Natural Pretty Breaks 기반으로) 5 단계로 나눕니다. 그러나 *Manual size classes* 옵션을 사용하면 사용자 지정 범주를 적용시킬 수 있습니다. 및 버튼을 사용해서 사용자 지정 범주의 값과 라벨을 설정하십시오.
- 범례가 접혀 있는 경우:
  - *Align symbols*: 심볼을 가운데 또는 하단으로 정렬시킬 수 있습니다.
  - 심볼에서 해당 범례 텍스트까지를 가리키는 *Line symbol* 수평 지시선을 환경 설정할 수 있습니다.

대화창의 오른쪽 패널에 범례의 미리보기를 표시하고, 사용자가 파라미터를 설정하는 대로 업데이트합니다.

> **참고**
>
> 현재, 레이어 심볼을 위한 데이터 정의 크기 범례는 단일 심볼, 범주 또는 등급 심볼을 사용하는 포인트 레이어에만 적용할 수 있습니다.

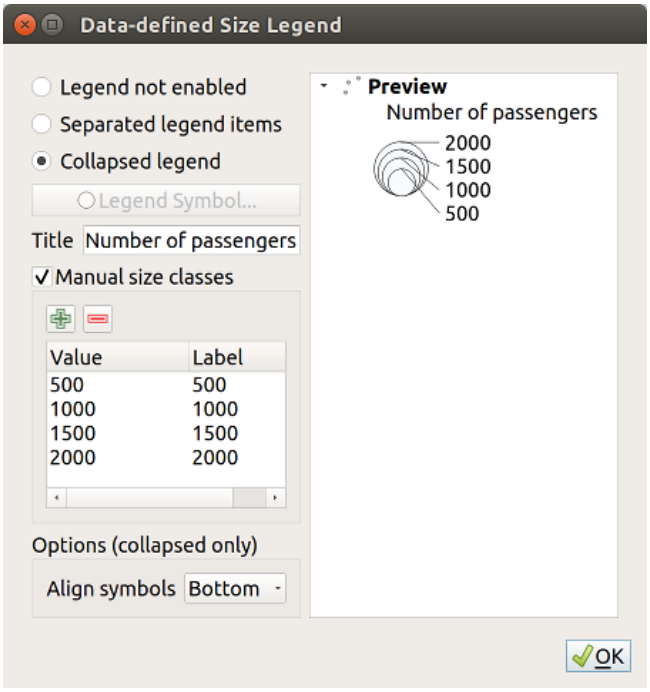



그림 12.16: 크기 척도 범례 설정하기

### 애니메이션 설정

*Animation settings* 패널을 활용하면 어떤 심볼이든애니메이션화된 심볼 로 만들 수 있습니다. 이 패널에서 심볼에 대한 애니메이션을 활성화하고 해당 심볼을 다시 그리기 위한 특정 프레임 레이트 (frame rate) 를 설정할 수 있습니다.

1. 심볼의 최고 수준에서, 대화창 우하단에 있는 *Advanced* 메뉴를 선택하십시오.
2. *Animation settings* 옵션을 찾으십시오.
3. *Is Animated* 를 체크해서 심볼에 대한 애니메이션을 활성화시키십시오.
4. *Frame rate* 를 환경설정하십시오. 예를 들면 애니메이션의 속도를 조절할 수 있습니다.
5. 이제 모든 하위 심볼 데이터 정의 속성에서 어떤 속성을 애니메이션화시키고 싶다면 `@symbol_frame` 변수를 사용하면 됩니다.

예를 들어 심볼의 회전을 데이터 정의 표현식 `@symbol_frame % 360` 로 설정하면, 시간이 지나면서 심볼이 회전할 것입니다. 이때 심볼의 프레임 레이트가 회전 속도를 정하게 됩니다:

### Extent buffer

You may set an extent buffer for a *symbol*. This means that a buffer is applied to the current map extent so that if a feature is outside of the actual map extent but inside the buffered extent it will still be rendered. This is useful for example with symbols which use the geometry generator where you would like to still see the generated geometries even if the actual feature is outside of the map extent.

To edit the extent buffer you can utilize the *Extent buffer* panel.

1. 심볼의 최고 수준에서, 대화창 우하단에 있는 *Advanced* 메뉴를 선택하십시오.
2. Find *Extent buffer* option
3. In the new panel you can set the buffer distance

The buffer distance units can be changed. You can also control the distance value by using the data defined override widget. For example you can change the value based on the current map scale `if(@map_scale > 50000, 5000, 0)`:

### 그리기 효과

레이어 렌더링을 향상시키고 최종 맵을 다른 소프트웨어로 렌더링하는 일을 피하기 (또는 적어도 줄이기) 위해, QGIS 는 또다른 강력한 기능을 제공합니다: *Draw Effects* 옵션은 사용자가 벡터 레이어의 가시화를 직접 조정할 수 있도록 그리기 효과를 추가합니다.

이 옵션은 *Layer Properties* ▸ *Symbology* 대화창의레이어 렌더링 그룹 (전체 레이어에 적용) 또는심볼 레이어 속성 (해당하는 피처에 적용) 에 있습니다. 이 두 옵션을 함께 사용할 수도 있습니다.

*Draw effects* 옵션을 체크한 다음 Customize effects 버튼을 클릭하면 그리기 효과를 활성화시킬 수 있습니다. *Effect Properties* 대화창이 열리는데 (그림 12.17 참조) 다음과 같은 유형의 효과들을 사용자 지정 옵션과 함께 쓸 수 있습니다:

- **소스 (Source)**: 레이어 속성 환경 설정에 따라 피처의 원본 스타일을 그립니다. 해당 스타일의혼합 모드 와그리기 모드 는 물론, *Opacity* 도 조정할 수 있습니다. 이 세 가지는 모든 효과 유형의 공통 속성입니다.

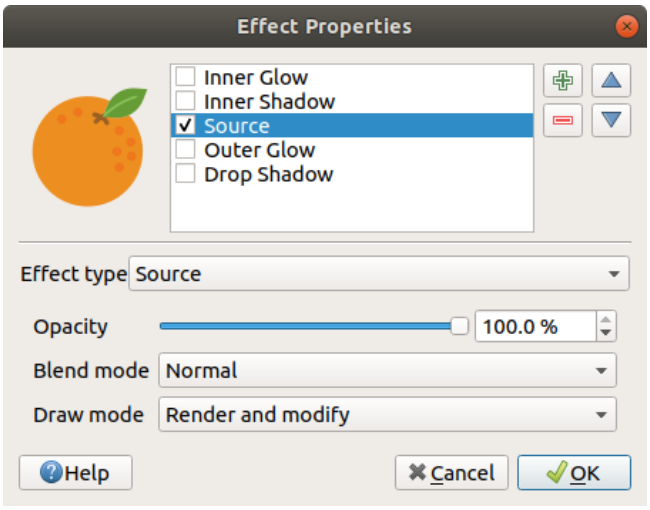



그림 12.17: 그리기 효과: 소스 대화창

- **흐리기 (Blur)**: 벡터 레이어에 흐리기 효과를 줍니다. 사용자가 *Blur type* (*Stack blur (fast)* 또는 *Gaussian blur (quality)*) 및 *Blur strength* 를 변경할 수 있습니다.

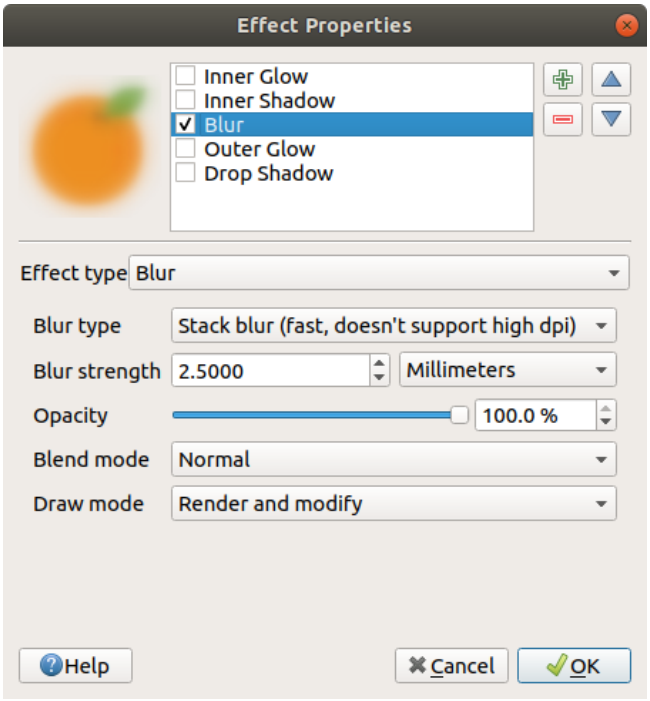



그림 12.18: 그리기 효과: 흐리기 대화창

- **색상화 (Colorise)**: 이 효과를 통해 단일 색조를 이용한 어떤 스타일을 만들 수 있습니다. 기반은 언제나 심볼의 회색조 버전으로, 다음을 설정할 수 있습니다:
  - *Grayscale* 을 통해 회색조 심볼을 어떻게 생성할지 ('By lightness', 'By luminosity', 'By average' 그리고'Off'옵션 가운데 하나를) 선택할 수 있습니다.
  - *Colorise* 를 활성화한 경우, 다른 색상을 혼합하고 색상의 강도를 설정할 수 있습니다.
  - 생성되는 심볼의 *Brightness* (명도), *Contrast* (대비) 그리고 *Saturation* (채도) 수준을 조정할 수 있습니다.
- **그림자 생성 (Drop Shadow)**: 이 효과를 이용하면 객체에 그림자를 추가하는데, 마치 차원을 하나 더한 것처럼 보입니다. 그림자의 방향 및 소스 객체와의 근접도를 결정하는 *Offset* 각도 및 거리를 변경해서 이 효과를 사용자 지정할 수 있습니다. *Drop Shadow* 는 그림자 효과의 *Blur radius* 및 *Color* 를 변경할 수 있는 옵션도 보유하고 있습니다.
- **내부 그림자 (Inner Shadow)**: 이 효과는 *Drop Shadow* 효과와 비슷하지만, 피처 경계선 내부에 그림자 효과를 추가합니다. *Drop Shadow* 효과와 동일한 사용자 지정 옵션을 설정할 수 있습니다.
- **내부 광원 (Inner Glow)**: 객체 내부에 광원 효과를 추가합니다. 광원의 *Spread* (너비) 또는 *Blur radius* 를 조정해서 이 효과를 사용자 지정할 수 있습니다. 흐리기 반경은 사용자가 흐리기 효과를 주려 하는 위치가 피처 경계선에 얼마나 근접해 있는지를 지정합니다. 또한, 광원의 색상을 *Single color* 또는 *Color ramp* 로 사용자 지정할 수 있는 옵션도 있습니다.
- **외부 광원 (Outer Glow)**: 이 효과는 *Inner Glow* 효과와 비슷하지만, 피처 경계선 외부에 광원 효과를 추가합니다. *Inner Glow* 효과와 동일한 사용자 지정 옵션을 설정할 수 있습니다.
- **변형 (Transform)**: 이 효과는 심볼의 형태를 변형시킬 수 있습니다. 사용자 지정을 위해 사용할 수 있는 첫 번째 옵션은 *Reflect horizontal* 및 *Reflect vertical* 으로, 수평 그리고/또는 수직 축을 기준으로 거울상을 실제로 생성합니다. 나머지 옵션 4 개는 다음과 같습니다:

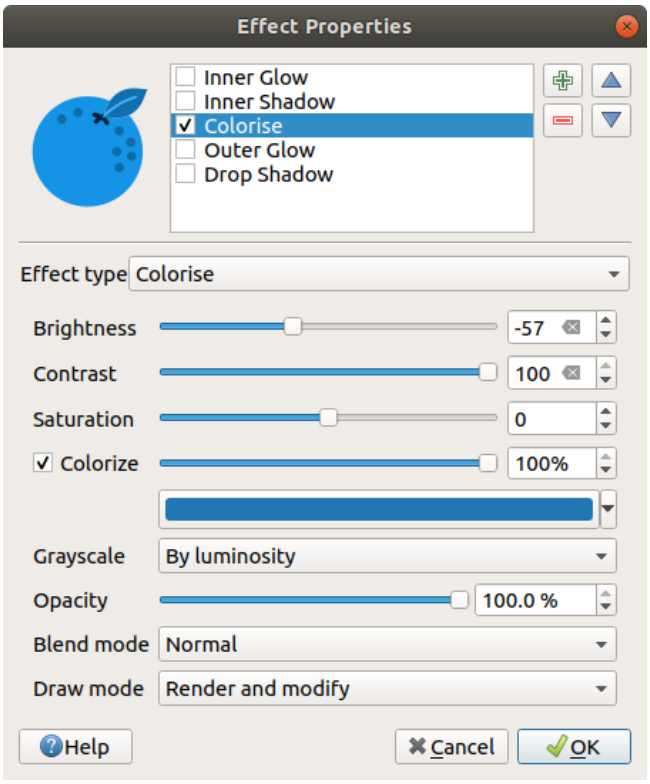



그림 12.19: 그리기 효과: 색상화 대화창

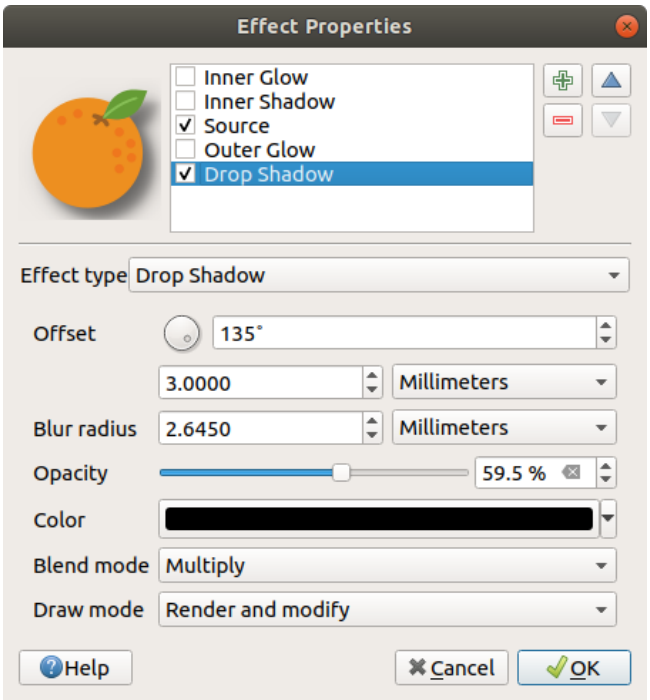



그림 12.20: 그리기 효과: 그림자 생성 효과 대화창

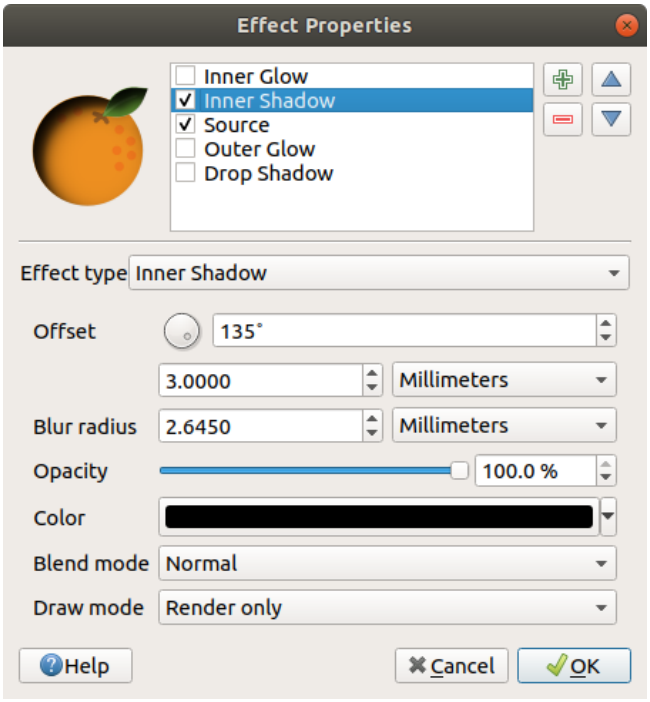



그림 12.21: 그리기 효과: 내부 그림자 대화창

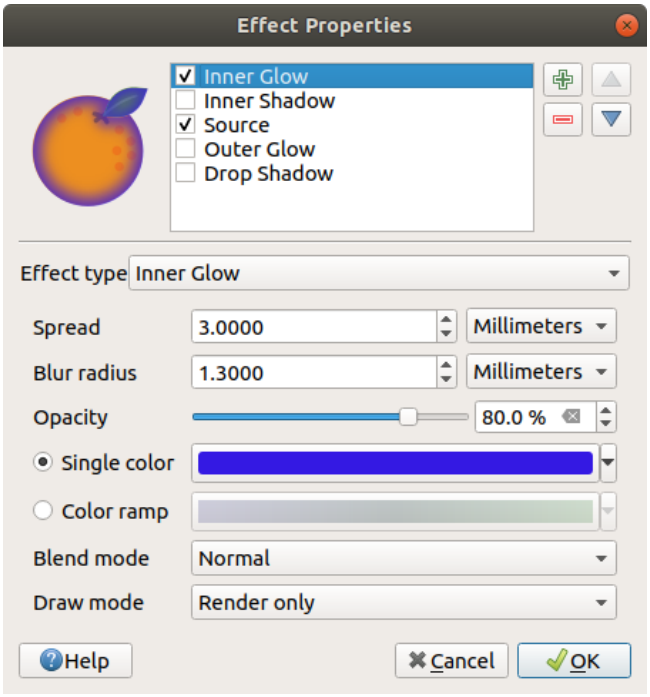



그림 12.22: 그리기 효과: 내부 광원 대화창

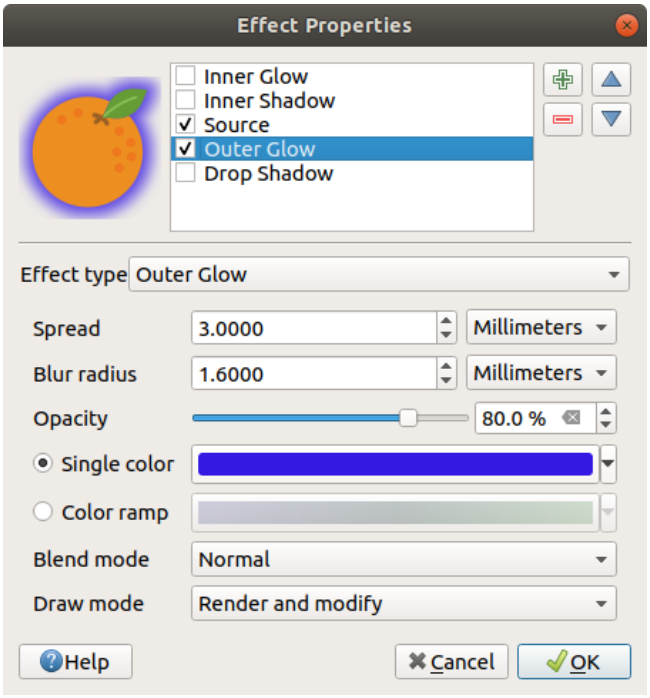



그림 12.23: 그리기 효과: 외부 광원 대화창

- *Shear X,Y*: X 그리고/또는 Y 축을 따라 피처를 기울입니다.
- *Scale X,Y*: X 그리고/또는 Y 축을 따라 피처를 지정한 백분율로 확대하거나 축소합니다.
- *Rotation*: 피처의 중심점을 기준으로 피처를 회전시킵니다.
- *Translate X,Y*: X 그리고/또는 Y 축을 기준으로 지정한 거리에 따라 피처의 위치를 변경합니다.

하나 이상의 그리기 효과를 동시에 줄 수 있습니다. 효과 목록에 있는 체크박스를 통해 어떤 효과를 활성화/비활성화시켜보십시오. *Effect type* 옵션을 이용하면 선택한 효과 유형을 변경할 수 있습니다. Move up 및 Move down 버튼으로 효과의 순서를 바꿀 수 있고, Add new effect 와 Remove effect 버튼을 통해 효과를 추가하거나 제거할 수도 있습니다.

모든 그리기 효과 유형에 사용할 수 있는 공통 옵션이 몇 개 있습니다. *Opacity* 및 *Blend mode* 옵션은레이어 렌더링에서 설명하는 옵션과 비슷하게 동작하며, 변형 효과를 제외한 모든 그리기 효과에 사용할 수 있습니다.

모든 그리기 효과에서 *Draw mode* 옵션을 쓸 수도 있는데, 심볼을 렌더링할지 그리고/또는 수정할지를 다음 몇몇 규칙에 따라 선택할 수 있습니다:

- 그리기 효과는 위에서 아래 순서로 렌더링합니다.
- *Render only* 모드는 효과를 가시화할 것이라는 뜻입니다.
- *Modifier only* 모드는 효과를 가시화하지는 않지만 적용된 변경 사항을 다음 (바로 아래에 있는) 효과로 넘길 것이라는 의미입니다.
- *Render and Modify* 모드는 효과를 가시화하고 적용된 변경 사항을 다음 효과로 넘길 것입니다. 어떤 효과가 효과 목록 제일 위에 있거나 또는 바로 위에 있는 효과가 수정 모드가 아닌 경우, (소스 효과와 비슷하게) 레이어 속성에서 가져온 원본 소스 심볼에 해당 효과를 줄 것입니다.

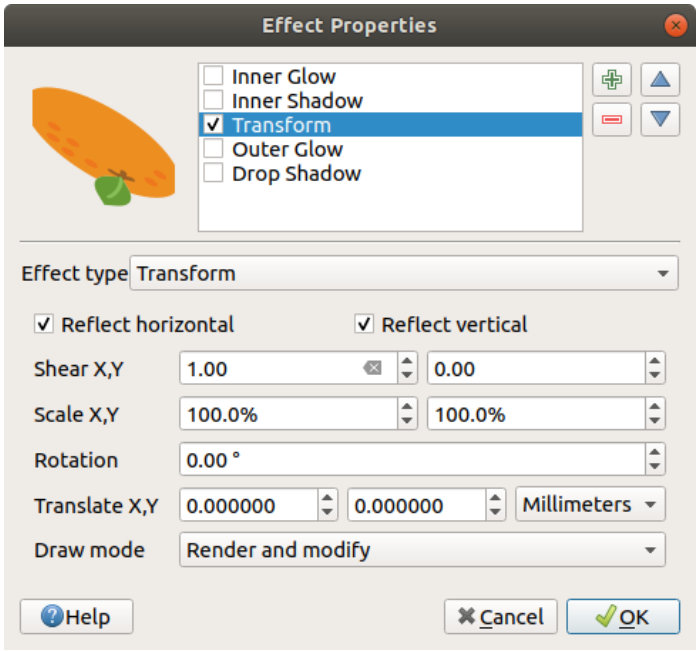



그림 12.24: 그리기 효과: 변형 대화창

### 12.1.4 라벨 속성

*Labels* 속성 대화창은 벡터 레이어에 대해 스마트 라벨 작업 환경을 설정하기 위해 필요한 그리고 적절한 모든 기능을 제공합니다. *Layer Styling* 패널에서 또는 **라벨 툴바** 의 Layer Labeling Options 아이콘을 통해 이 대화창을 열 수 있습니다.

At the top of the dialog, you have:

- a combobox for selecting the appropriate labeling method for the active layer
- the Configure project labeling rules button: helps you control interactions between labels and features across the layers in the project. More details at *Configuring project labeling rules*.
- the Automated placement settings (applies to all layers) button: configure general properties on label placement and conflicts resolution. More details at 자동화된 배치 엔진 설정하기.

먼저 드롭다운 목록에서 라벨 작업 방식을 선택합니다. 다음 4 개의 옵션을 사용할 수 있습니다:

- *No labels*: 기본값입니다. 레이어에 어떤 라벨도 표시하지 않습니다.
- *Single labels*: 맵 상에 라벨을 단일 속성 또는 표현식을 이용해서 표시합니다.
- 규칙 기반 라벨 작업
- *Blocking*: 어떤 레이어의 라벨을 렌더링하지 않고 해당 레이어가 다른 레이어를 가리도록만 설정할 수 있습니다.

다음으로 *Single labels* 옵션을 선택했다고 가정하겠습니다. 다음 대화창이 열립니다:

대화창 상단에 *Value* 드롭다운 목록이 활성화됩니다. 이 목록에서 라벨 작업에 사용할 속성 열을 선택할 수 있습니다. 디스플레이 필드 를 기본값으로 사용합니다. 표현식을 바탕으로 라벨을 정의하고 싶다면 ε 을 클릭하십시오–표현식을 기반으로 라벨 정의하기 를 참조하세요.

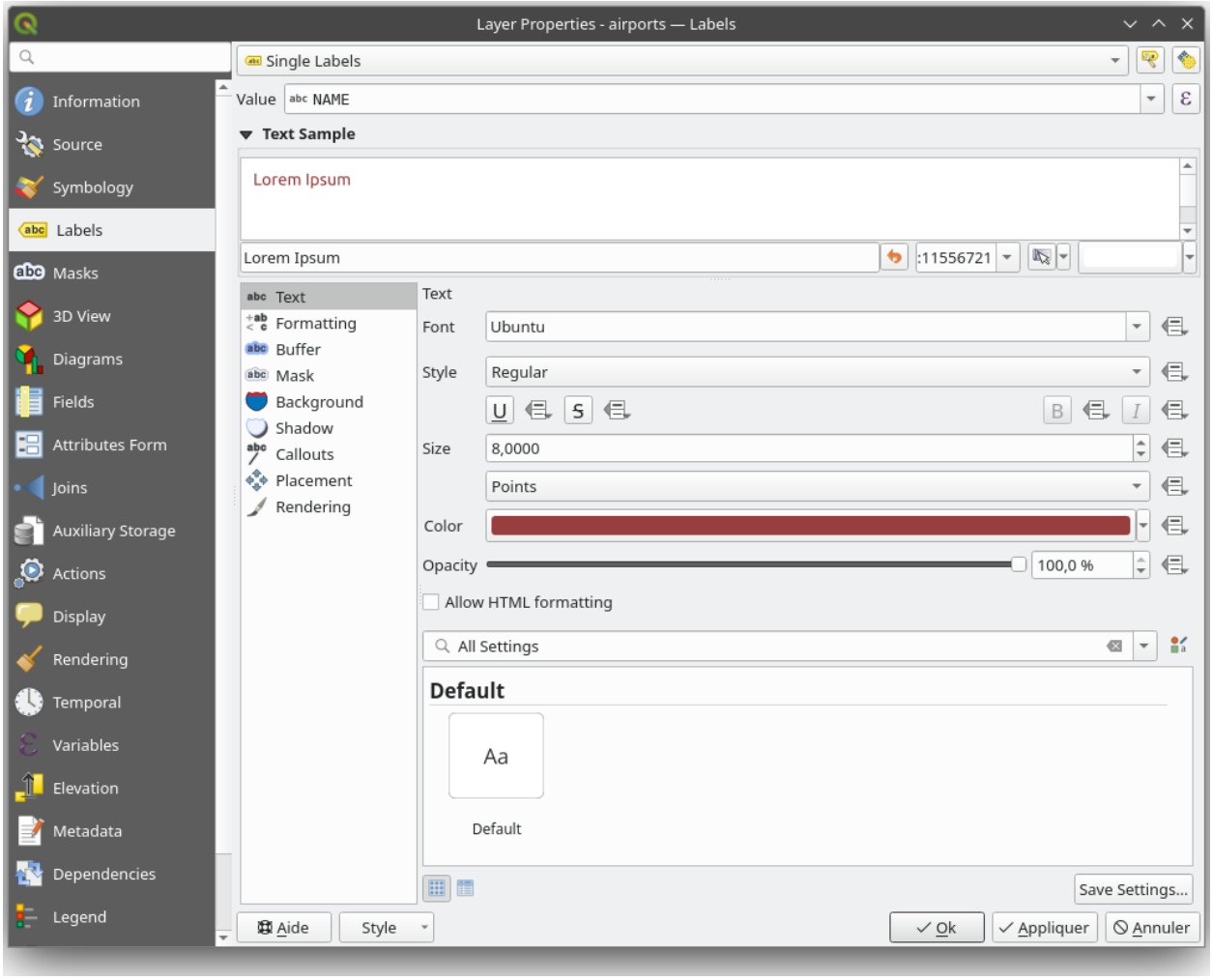



그림 12.25: 레이어 라벨 작업 설정 - 단일 라벨

> **참고**
>
> 범례 탭에서 라벨을 활성화시키면 범례에 라벨을 라벨 서식과 함께 항목으로 표시할 수 있습니다.

여러 탭에 라벨을 사용자 지정할 수 있는 다음 옵션들이 표시됩니다:

- abc 텍스트
- 서식 작업
- abc 버퍼
- abc 마스크
- 배경
- 그림자
- abc 호출
- 배치
- 렌더링

라벨 설정 에서 각 속성을 설정하는 방법을 설명하고 있습니다.

### Configuring project labeling rules

Pressing the Configure project labeling rules button next to the labeling method drop-down selector, you can create rules that controls how labels from a layer can interact with labels or features from another layer.

1. Press the Add rule button and in the drop-down menu, select one of the rule types:
   - *Prevent labels overlapping features*: prevents labels being placed overlapping features from a different layer.
   - *Pull labels towards features*: prevents labels being placed too far from features from a different layer. The maximum distance can be set in the unit of your choice.
   - *Push labels away from features*: prevents labels being placed too close to features from a different layer. The minimum distance can be set in the unit of your choice, as well as the *rule's priority* (The highest-priority rules are more important to respect in the event of a label placement conflict).
   - *Push labels away from other labels*: prevents labels being placed too close to labels from a different layer.

   > **주의**
   >
   > The last three options are only available on QGIS installed with GEOS >= 3.10 (see *Help → About* menu).

2. Fill the properties at your will; you can provide a more meaningful name to the rule.
3. *OK* 를 클릭합니다.
4. Add as many rules as necessary.
5. If necessary, press Edit rule to modify the selected rule or Remove rule to delete it from the project.

The set rules are available from any layer *Labels* properties tab, pressing the Configure project labeling rules button. You can temporary enable or disable any of them, using the checkbox next to the name. Hover over a rule to preview its details.

### 자동화된 배치 엔진 설정하기

자동화된 배치 설정을 사용하면 라벨의 프로젝트 수준 및 자동화된 습성 환경을 설정할 수 있습니다. *Labels* 탭 우상단에 있는 Automated placement settings (applies to all layers) 버튼을 클릭하면 다음 옵션을 가지고 있는 대화창이 열립니다:

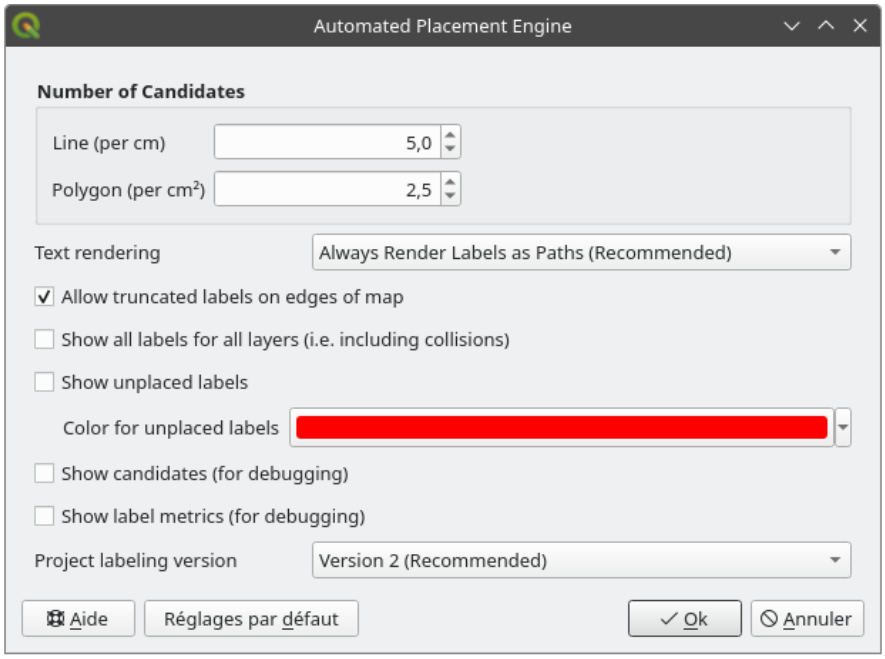



그림 12.26: 라벨 배치 자동화 엔진

- *Number of candidates*: 라인 및 폴리곤 피처의 크기를 바탕으로 배치할 수 있는 라벨의 개수를 계산해서 피처에 할당합니다. 피처가 더 길거나 넓을수록 더 많은 후보를 가지게 되므로 해당 피처의 라벨이 충돌할 위험을 덜고 더 잘 배치할 수 있습니다.
- *Text rendering*: sets the default value for label rendering widgets when *exporting a map canvas* or *a layout* to PDF or SVG. If *Always render labels as text* is selected then labels can be edited in external applications (e.g. Inkscape) as normal text. BUT the side effect is that the rendering quality is decreased, and there are issues with rendering when certain text settings like buffers are in place. That's why *Always render labels as paths (recommended)* which exports labels as outlines but guarantees complete compatibility with the full range of formatting options available, is recommended. With *Prefer rendering labels as text* option, labels are rendered as text objects, unless doing so results in rendering artifacts or poor quality rendering (depending on text format settings).

> **참고**
>
> When rendering labels as text to a vector based device (e.g. PDF or SVG), care must be taken to ensure that the required fonts are available to users when opening the created files, or default fallback fonts will be used to display the output instead. (Although PDF exports MAY automatically embed some fonts when possible, depending on the user's platform).

- *Allow truncated labels on edges of map*: 일부가 맵 범위 바깥으로 벗어나는 라벨을 렌더링할지 여부를 제어합니다. 이 옵션을 체크하면 (라벨을 완전히 가시 영역 안으로 배치할 방법이 없는 경우) 이런 라벨을 표시할 것입니다. 체크를 해제하면 일부만 보이는 라벨을 무시할 것입니다. 이 설정은조판기 맵 항목 에서의 라벨 표시에는 아무 영향도 주지 않는다는 사실을 기억하십시오.
- *Show all labels for all layers (i.e. including colliding objects)*: 이 옵션을 레이어별로 설정할 수도 있다는 사실을 기억하십시오. (렌더링 탭 참조)
- *Show unplaced labels*: 어떤 중요한 라벨이 (예를 들면 중첩 또는 다른 제약조건 때문에) 맵에서 없어졌는지 여부를 확인할 수 있습니다. 이런 라벨은 사용자 지정 색상으로 표시됩니다.
- *Show candidates (for debugging)*: 맵 위에 라벨 배치용으로 생성된 모든 후보를 보여주는 상자들을 그릴지 여부를 제어합니다. 옵션 이름대로, 이 옵션은 디버깅 및 서로 다른 라벨 작업 설정이 주는 효과를 점검하는 데에만 유용합니다. 사용자가 직접라벨 툴바 의 도구를 사용해서 더 나은 배치 작업을 하는 경우 이 옵션이 편리할 수도 있습니다.
- *Show label metrics (for debugging)*: displays the text bounds of the label in red and baselines in blue
- *Project labeling version*: QGIS 는 두 가지 서로 다른 라벨 배치 자동화 버전을 지원합니다:
  - *Version 1*: (QGIS 3.10 이하 버전이 사용하는, 그리고 이전 버전에서 생성된 프로젝트 파일을 QGIS 3.12 이후 버전에서 열었을 경우 사용되는) 이전 체계입니다. 버전 1 은 라벨 및 방해물의 우선 순위를 "대강의 안내"로만 취급하기 때문에, 이 버전에서는 우선 순위가 낮은 라벨이 우선 순위가 높은 방해물 위로 배치될 수도 있습니다. 따라서 이 버전을 사용할 경우 원하는 라벨 작업 산출물을 얻기 어렵게 되므로, 옛날 프로젝트와의 호환성만을 위해 사용하는 것을 추천합니다.
  - *Version 2 (*추천*)*: QGIS 3.12 이후 버전에서 생성된 새 프로젝트에서 사용되는 기본 체계입니다. 버전 2 에서는 라벨이방해물 을 중첩할 수 있는 경우를 규정하는 논리를 재작업했습니다. 이 새로운 논리는 어떤 라벨도 스스로의 우선 순위와 비교해 더 큰 방해물 가중치를 가진 어떤 방해물도 중첩할 수 없도록 하고 있습니다. 그 결과, 이 버전은 훨씬 예상 가능하고 쉽게 이해할 수 있는 라벨 작업 산출물을 내놓습니다.

### 규칙 기반 라벨 작업

규칙 기반 렌더링 에서처럼, 규칙 기반 라벨 작업으로 한 번에 여러 라벨을 환경 설정할 수 있고 표현식 필터와 축척 범위를 바탕으로 선택적으로 적용할 수 있습니다.

규칙을 생성하려면:

1. *Labels* 탭에서 주 드롭다운 목록에 있는 **규칙 기반 라벨 작업** 을 선택하십시오.
2. 대화창 하단에 있는 Add rule 버튼을 클릭하십시오.
3. 새 대화창이 나타나면 다음을 입력하십시오:
   - *Description*: *Labels* 탭에 있는 규칙을 식별하기 위해 쓰이는, 그리고 인쇄 조판 범례에라벨 범례 항목 으로 쓰이는 텍스트입니다.
   - *Filter*: 라벨 설정을 적용할 피처를 선택하기 위한 표현식입니다.
   - 기존에 설정된 규칙이 존재하는 경우, *Else* 옵션을 사용하면 같은 그룹 안에 있는 규칙의 어떤 필터와도 일치하지 않는 피처를 모두 선택할 수 있습니다.
4. 라벨 규칙이 적용되어야 할축척 범위 를 설정할 수 있습니다.
5. *Labels* 그룹 프레임에서 사용할 수 있는 옵션들은 일반적인라벨 설정들 입니다. 이들을 환경 설정한 다음 *OK* 버튼을 누르십시오.

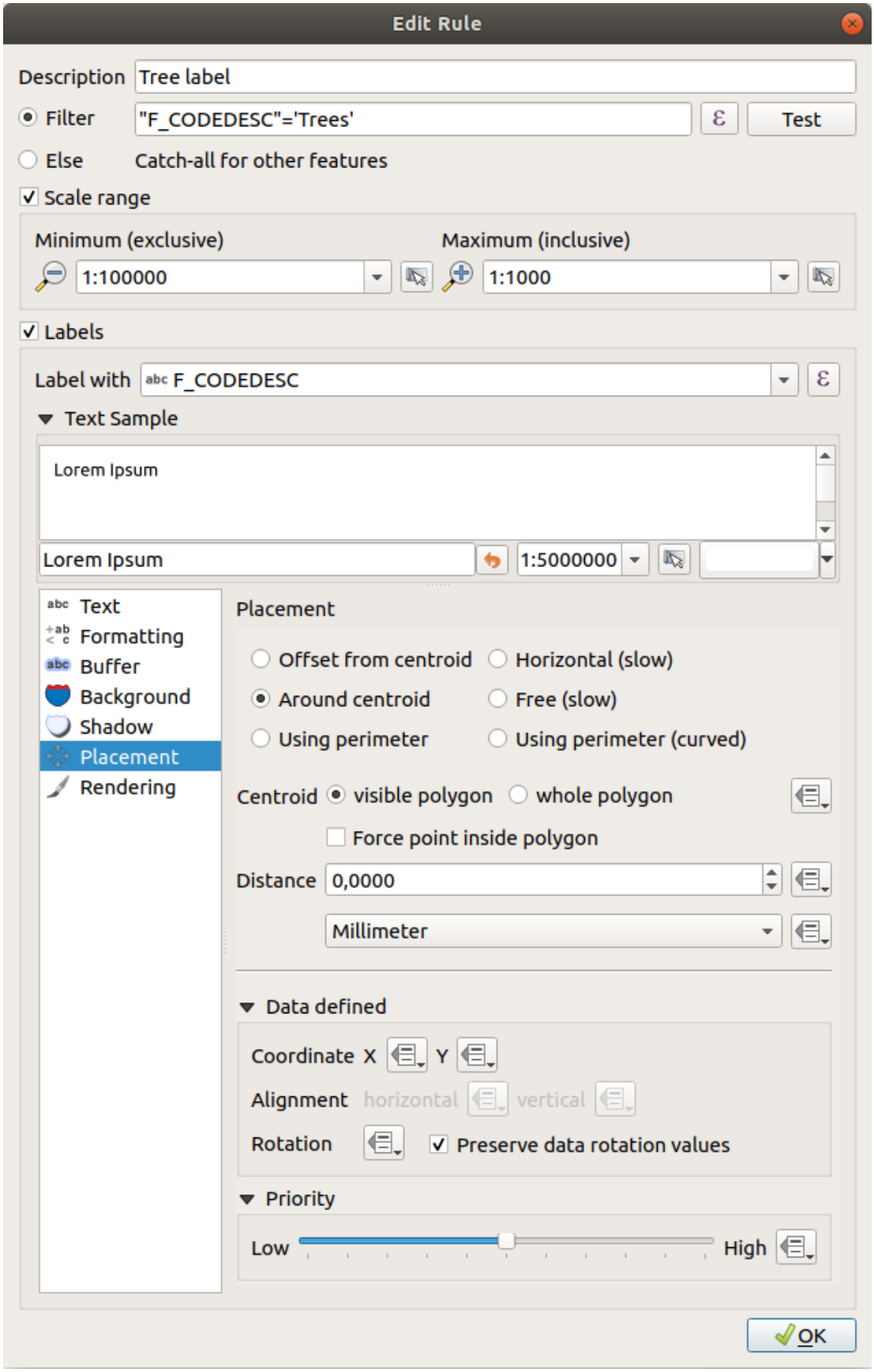

Edit Rule
Description
Tree label
Filter
"F_CODEDESC"='Trees'
Test
Else
Catch-all for other features
Scale range
Minimum (exclusive)
1:100000
Maximum (inclusive)
1:1000
Labels
Label with
F_CODEDESC
Text Sample
Lorem Ipsum
Lorem Ipsum
1:5000000
Text
Formatting
Buffer
Background
Shadow
Placement
Rendering
Placement
Offset from centroid
Horizontal (slow)
Around centroid
Free (slow)
Using perimeter
Using perimeter (curved)
Centroid
visible polygon
whole polygon
Force point inside polygon
Distance
0,0000
Millimeter
Data defined
Coordinate X
Y
Alignment
horizontal
vertical
Rotation
Preserve data rotation values
Priority
Low
High
OK

그림 12.27: 규칙 설정

기존 규칙들을 요약한 내용이 주 대화창에 (그림 12.28 참조) 표시됩니다. 여러 개의 규칙을 추가할 수 있고, 드래그 & 드롭으로 이 규칙들을 재정렬하거나 서로 겹치게 배열할 수 있습니다. 규칙을 버튼으로 제거할 수도 있고, 버튼 또는 더블클릭으로 편집할 수도 있습니다.

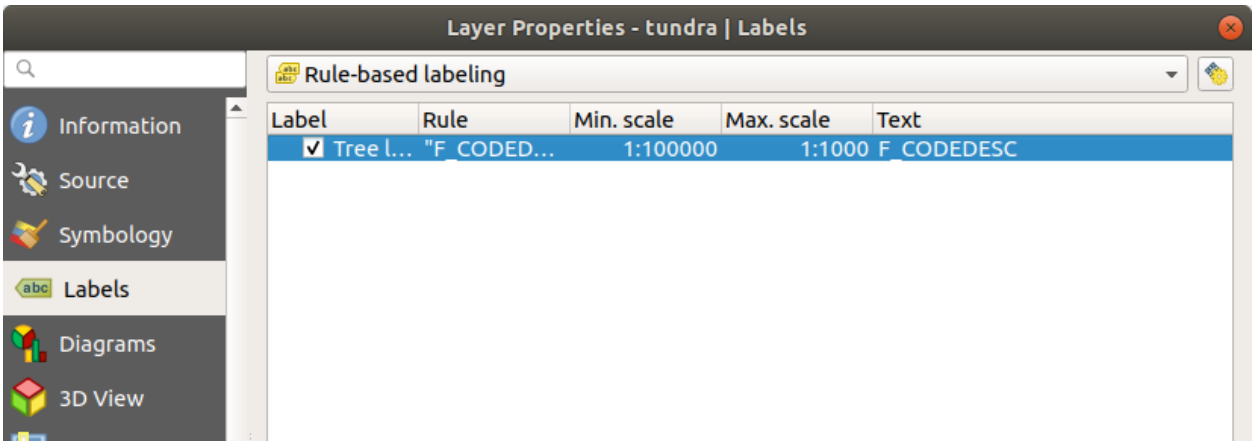



그림 12.28: 규칙 기반 라벨 작업 패널

### 표현식을 기반으로 라벨 정의하기

단일 라벨 또는 규칙 기반 라벨 작업 가운데 어떤 유형을 선택하든, QGIS 는 라벨 피처에 표현식을 이용할 수 있습니다.

*Single labels* 방법을 사용한다고 가정하고, 레이어 속성 대화창의 *Labels* 탭에 있는 *Value* 드롭다운 목록 옆에 있는 아이콘을 클릭하십시오.

그림 12.29 는 알래스카 수목 레이어를 수목 유형 및 면적으로 라벨 작업하기 위한 표현식의 예시를 보여주고 있습니다. 'VEGDESC'필드, 몇몇 설명 텍스트, 그리고 `format_number()` 함수와 `$area` 를 결합해서 라벨을 더 보기 좋게 만듭니다.

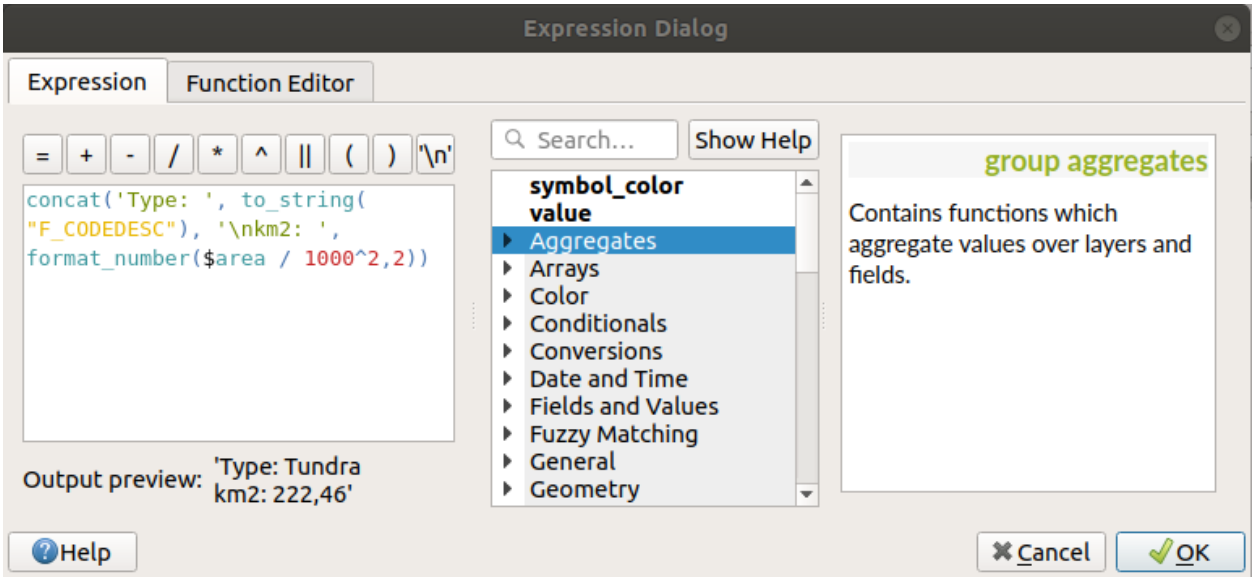



그림 12.29: 라벨 작업에 표현식 이용하기

표현식 기반 라벨 작업은 쉽습니다. 사용자가 주의해야 할 점은 다음 세 가지뿐입니다:

- 모든 요소들 (문자열, 필드, 함수) 을 `concat`, `+` 또는 `||` 같은 문자열 연결 함수로 결합시켜야 합니다. (NULL 또는 숫자 값을 쓰는 경우) 이런 모든 도구를 써도 사용자의 필요를 충족시키지 못 하는 경우가 있을 것이라는 점을 기억하십시오.
- 문자열은'작은따옴표'안에 작성해야 합니다.
- 필드는"큰따옴표"안에, 또는 아무 따옴표도 없이 작성해야 합니다.

예시를 몇 가지 살펴보겠습니다:

1. 'name'및'place'필드 2 개를 쉼표로 구분하는 라벨:

```
"name" || ', ' || "place"
```

   다음을 반환합니다:

```
John Smith, Paris
```

2. 'name'및'place'필드 2 개와 다른 텍스트를 바탕으로 하는 라벨:

```
'My name is ' + "name" + 'and I live in ' + "place"
'My name is ' || "name" || 'and I live in ' || "place"
concat('My name is ', name, ' and I live in ', "place")
```

   다음을 반환합니다:

```
My name is John Smith and I live in Paris
```

3. 'name'및'place'필드 2 개와 다른 텍스트를 서로 다른 연결 함수로 결합하는 라벨:

```
concat('My name is ', name, ' and I live in ' || place)
```

   다음을 반환합니다:

```
My name is John Smith and I live in Paris
```

   또는'place'필드가 NULL 값인 경우 다음을 반환합니다:

```
My name is John Smith
```

4. 'name'및'place'필드 2 개와 설명 텍스트를 바탕으로 하는 여러 줄의 라벨:

```
concat('My name is ', "name", '\n' , 'I live in ' , "place")
```

   다음을 반환합니다:

```
My name is John Smith
I live in Paris
```

5. 필드와 `$area` 함수로 장소명 및 변환 단위의 반올림된 면적을 바탕으로 하는 라벨:

```
'The area of ' || "place" || ' has a size of '
|| round($area/10000) || ' ha'
```

   다음을 반환합니다:

```
The area of Paris has a size of 10500 ha
```

6. CASE ELSE 조건을 생성해봅시다. 'population'필드의 인구 값이 5 만 명 이하일 경우'town', 초과할 경우'city'로 라벨을 생성하려면:

```
concat('This place is a ',
CASE WHEN "population" <= 50000 THEN 'town' ELSE 'city' END)
```

다음을 반환합니다:

```
This place is a town
```

7. 도시의 이름만 표시하고 다른 피처의 라벨은 표시하지 않으려면 ("city"의 맥락은 바로 앞의 예시를 참조하세요):

```
CASE WHEN "population" > 50000 THEN "NAME" END
```

다음을 반환합니다:

```
Paris
```

표현식 작성기에서 볼 수 있듯이, QGIS 는 사용자 데이터의 라벨을 위한 단순 및 복잡 표현식을 생성하는 데 쓸 수 있는 함수를 수백 개나 제공하고 있습니다. 표현식에 관한 자세한 정보 및 예시를 살펴보려면표현식 을 참조하세요.

### 라벨 작업에 데이터 정의 무시 이용하기

Data defined override 기능을 통해, 속성 테이블의 항목 또는 그에 기반한 표현식이 라벨 작업 설정을 무시하게 만들 수 있습니다. 앞에서 설명한 라벨 작업 옵션 대부분의 값을 설정하는 데 이 기능을 이용할 수 있습니다.

예를 들어 알래스카 QGIS 예시 데이터셋을 사용해서 `airports` 레이어에 군수 목적을 나타내는 `USE` 를 기반으로 공항 이름 라벨을 붙여봅시다. 이를테면 공항에 접근할 수 있는 것이:

- 군대일 경우 회색, 크기 8 로 표시하고
- 다른 경우 파란색, 크기 10 으로 표시합니다.

이렇게 하려면 레이어의 `NAME` 필드에 대해 라벨 작업을 활성화한 다음 (라벨 설정 참조):

1. *Text* 탭을 선택하십시오.
2. *Size* 속성 옆에 있는 아이콘을 클릭하십시오.
3. *Edit…*를 선택해서 다음과 같이 입력하십시오:

```
CASE
  WHEN "USE" like '%Military%' THEN 8 -- because compatible values are 'Military'
                                      -- and 'Joint Military/Civilian'
  ELSE 10
END
```

4. *OK* 를 눌러 검증하십시오. 대화창이 닫히고, 버튼이 규칙이 실행 중임을 뜻하는 으로 바뀝니다.
5. 그 다음 색상 속성 옆에 있는 버튼을 클릭하고, 다음과 같이 입력한 후 검증하십시오:

```
CASE
  WHEN "USE" like '%Military%' THEN '150, 150, 150'
  ELSE '0, 0, 255'
END
```

라벨의 어떤 속성이든, 이와 같이 원하는 대로 사용자 지정할 수 있습니다. 데이터 정의 무시 설정 에서 Data-define override 위젯의 설명 및 조작법에 대한 자세한 내용을 참조하세요.

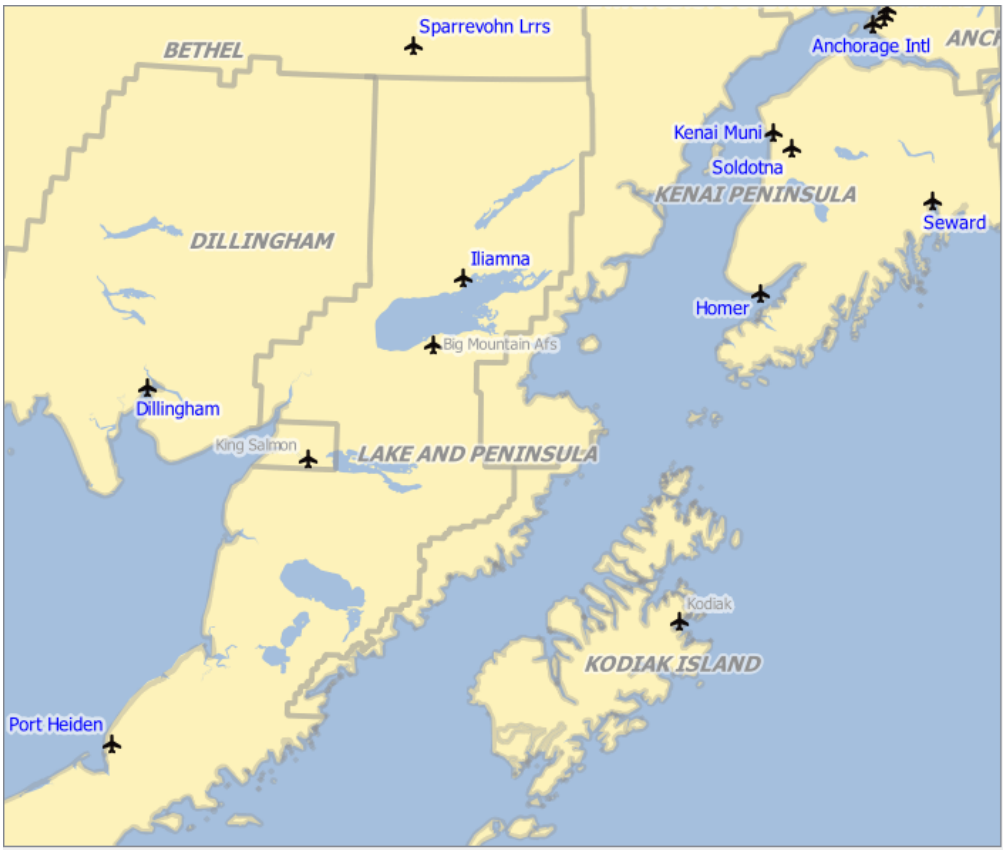



그림 12.30: 공항의 속성을 기반으로 서식을 작업한 공항 라벨

**팁**

**데이터 정의 무시를 사용해서 다중부분 피처의 모든 부분에 라벨 붙이기**

사용자의 라벨 속성과는 별개로 다중부분 (multi-part) 피처용 라벨 작업을 설정할 수 있는 옵션이 존재합니다. 렌더링, `Feature options` 를 선택하고, *Label every part of multipart-features* 옆에 있는 Data-define override 버튼을 클릭한 다음데이터 정의 무시 설정 에서 설명하는 대로 라벨을 정의하십시오.

### 라벨 툴바

*Label Toolbar* 는 (라벨호출 을 포함하는) 라벨 또는 도표 속성을 조작할 수 있는 몇몇 도구를 제공합니다:



그림 12.31: 라벨 툴바

- Highlight Pinned Labels, Diagrams and Callouts: 라벨 항목이 속한 벡터 레이어가 편집 가능 상태인 경우 녹색으로 강조되고, 편집 가능 상태가 아니라면 파란색을 유지합니다.
- Toggle Display of Unplaced Labels: 어떤 중요한 라벨이 (예를 들면 중첩 또는 다른 제약조건 때문에) 맵에서 누락되었는지 여부를 확인할 수 있습니다. 이런 라벨은 사용자 지정이 가능한 색상으로 표시됩니다. (자동화된 배치 엔진 설정하기 를 참조하세요.)
- Pin/Unpin Labels and Diagrams: 어떤 영역을 클릭하거나 드래그하면, 해당 영역에 있는 항목들에 핀을 꽂습니다. `Shift` 키를 누른 채 어떤 영역을 클릭하거나 드래그하면, 해당 영역의 항목들에 꽂혀 있는 핀을 제거합니다. 마지막으로, `Ctrl` 키를 누른 채 어떤 영역을 클릭하거나 드래그하면 해당 영역의 항목들의 핀 상태를 반전시킬 수 있습니다.
- Show/Hide Labels and Diagrams: 항목을 클릭하거나 kbd:*Shift* 키를 누른 채 어떤 영역을 클릭 & 드래그하면 해당 항목을 숨깁니다. 항목이 숨겨져 있는 경우, 그냥 해당 피처를 한번 클릭하면 가시성을 복원합니다. 영역을 드래그하면 해당 영역에 있는 모든 항목의 가시성을 복원할 것입니다.
- Move a Label, Diagram or Callout: 항목을 클릭해서 선택한 다음 원하는 위치를 클릭해서 해당 위치로 옮깁니다. 새로운 좌표는보조 필드 에 저장됩니다. 이 도구로 항목을 선택한 다음 `Delete` 키를 누르면 저장된 위치값을 삭제할 것입니다.
- Rotate a Label: 라벨을 클릭해서 선택한 다음 원하는 기울기가 적용되도록 다시 클릭하십시오. 마찬가지로, 새 각도는 보조 필드에 저장됩니다. 이 도구로 라벨을 선택한 다음 `Delete` 키를 누르면 해당 라벨의 기울기값을 삭제할 것입니다.
- Change Label Properties: 이 아이콘을 누른 다음 라벨을 클릭하면, 해당 라벨의 속성을 변경할 수 있는 대화창이 열립니다. 라벨 그 자체, 라벨의 좌표, 각도, 글꼴, 크기, 멀티라인 정렬 등등 그 속성이 필드에 매핑돼 있는 한 해당 속성을 변경할 수 있습니다. 여기에서 피처의 모든 부분에 라벨을 붙이는 *Label every part of a feature* 옵션을 설정할 수 있습니다.

**경고**

**라벨 도구는 현재 필드 값을 덮어 씁니다**

라벨 작업을 사용자 지정하는 데 *Label toolbar* 를 사용하면 매핑된 필드에 새 속성값을 실제로 작성합니다. 따라서, 사용자가 나중에 필요할 수도 있는 데이터를 무심코 덮어 쓰지 않도록 주의하십시오!

**참고**

기저 데이터소스를 수정하지 않은 채 라벨 작업 (위치 등등) 을 사용자 지정하는 데보조 저장소 속성 메커니즘을 사용할 수도 있습니다.

#### 맵 캔버스에서 라벨을 사용자 지정하기

*Label Toolbar* 와 데이터 정의 무시 설정을 결합하면 맵 캔버스에 있는 라벨을 조작할 (이동시킬, 편집할, 기울기를 적용할) 수 있습니다. 이제 Move Label, Diagram or Callout 기능에 데이터 정의 무시 기능을 이용하는 예시를 설명할 것입니다. (그림 12.32 를 참조하세요.)

1. QGIS 예시 데이터셋으로부터 `lakes.shp` 레이어를 가져옵니다.
2. 레이어를 더블클릭해서 레이어 속성 대화창을 엽니다. *Labels* 탭을 선택하고 *Placement* 를 클릭한 다음 *Offset from centroid* 를 선택합니다.
3. *Data defined* 항목을 찾아 아이콘을 클릭해서 *Coordinate* 의 필드 유형을 정의합니다. X 값으로 `xlabel` 을, Y 값으로 `ylabel` 을 선택합니다. 이제 아이콘이 노란색으로 강조됐을 겁니다.

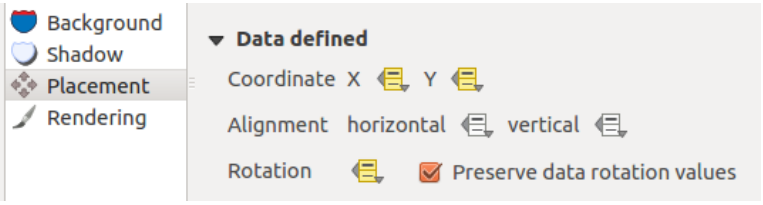



그림 12.32: 데이터 정의 무시를 통한 벡터 폴리곤 레이어의 라벨 작업

4. 호수로 확대/축소합니다.
5. Toggle Editing 버튼으로 레이어를 편집 가능한 상태로 설정합니다.
6. 라벨 툴바의 아이콘을 클릭합니다. 이제 또다른 위치로 라벨을 직접 옮길 수 있습니다. (그림 12.33 을 참조하세요.) 라벨의 새 위치가 속성 테이블의 `xlabel` 및 `ylabel` 열에 저장됩니다.
7. 다음을 사용해서 각 호수와 옮긴 라벨을 연결하는 라인을 추가할 수도 있습니다:
   - 라벨의호출 속성
   - 또는 다음 표현식과 함께도형 생성기 심볼 레이어 를 사용하는 경우:

```
make_line( centroid( $geometry ), make_point( "xlabel", "ylabel" ) )
```

> **참고**
>
> 편집할 수 있는 기저 데이터소스가 없어도보조 저장소 속성 메커니즘을 데이터 정의 속성과 함께 사용할 수도 있습니다.

### 12.1.5 도표 속성

The *Diagrams* tab allows you to add a graphic overlay to a vector layer (see 그림 12.34). This dialog can also be accessed from the *Layer Styling* panel, or using the Layer Diagram Options button of the **Labels toolbar**.

현재 도표의 핵심 구현은 다음을 지원하고 있습니다:

- *No diagrams*: 기본값입니다. 피처 위로 어떤 도표도 표시하지 않습니다.

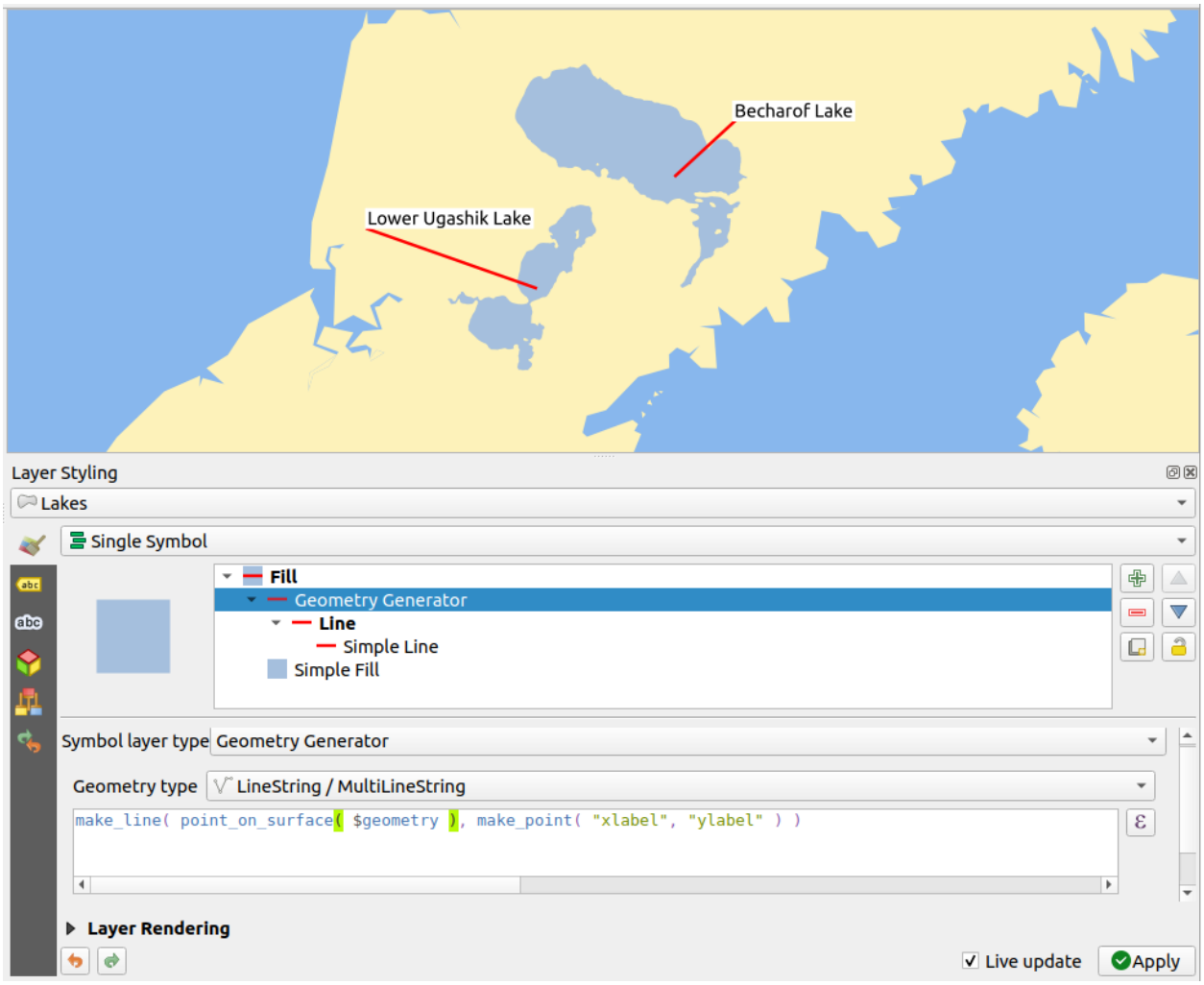



그림 12.33: 이동시킨 라벨

-  *Pie chart*: 원형 통계 그래픽인 원형 차트로, 숫자로 나타낸 비율을 보여주는 조각들로 나누어집니다. 각 조각의 원호 길이는 각 조각이 나타내는 양과 비례합니다.
- abc *Text diagram*, a horizontally divided circle showing statistic values inside;
-  *Histogram*, bars of varying colors for each attribute aligned next to each other;
-  *Stacked bars*, stacks bars of varying colors for each attribute on top of each other vertically or horizontally;
-  *Stacked diagram*, stacks diagrams of equal or varying types, next to each other, vertically or horizontally. More details at *Stacked Diagrams*.

*Diagrams* 탭의 우상단에 있는  Automated placement settings (applies to all layers) 버튼을 클릭하면 맵 상에 있는 도표라벨의 배치 를 제어할 수 있습니다.

> **팁**
>
> **도표 유형을 재빨리 바꾸기**
>
> 서로 다른 도표 유형들의 설정이 거의 동일하다는 점을 고려하면, 사용자가 도표를 구성할 때 도표 유형을 쉽게 변경해서 어떤 유형이 사용자 데이터를 아무 손실 없이 적절하게 표현하는지 확인할 수 있습니다.

각 도표 유형 별로, 속성은 다음과 같은 여러 탭으로 나누어져 있습니다:

- 속성
- 렌더링
- 크기
- 배치
- 옵션
- 범례

### 속성

*Attributes* 탭은 도표에 어떤 변수를 표시할지 정의합니다.  add item 버튼을 이용해서'Assigned Attributes'패널로 원하는 필드를 선택해 넣으십시오. 표현식 으로 생성한 속성도 사용할 수 있습니다.

어떤 행이든 클릭 & 드래그하면 위 아래로 이동시켜 속성이 표시되는 방식을 정렬시킬 수 있습니다. 항목을 더블 클릭해서'Legend'열에 있는 라벨을 변경하거나, 속성 색상을 변경할 수도 있습니다.

이 라벨은 인쇄 조판기 또는 레이어 트리의 범례에 표시되는 기본 텍스트입니다.

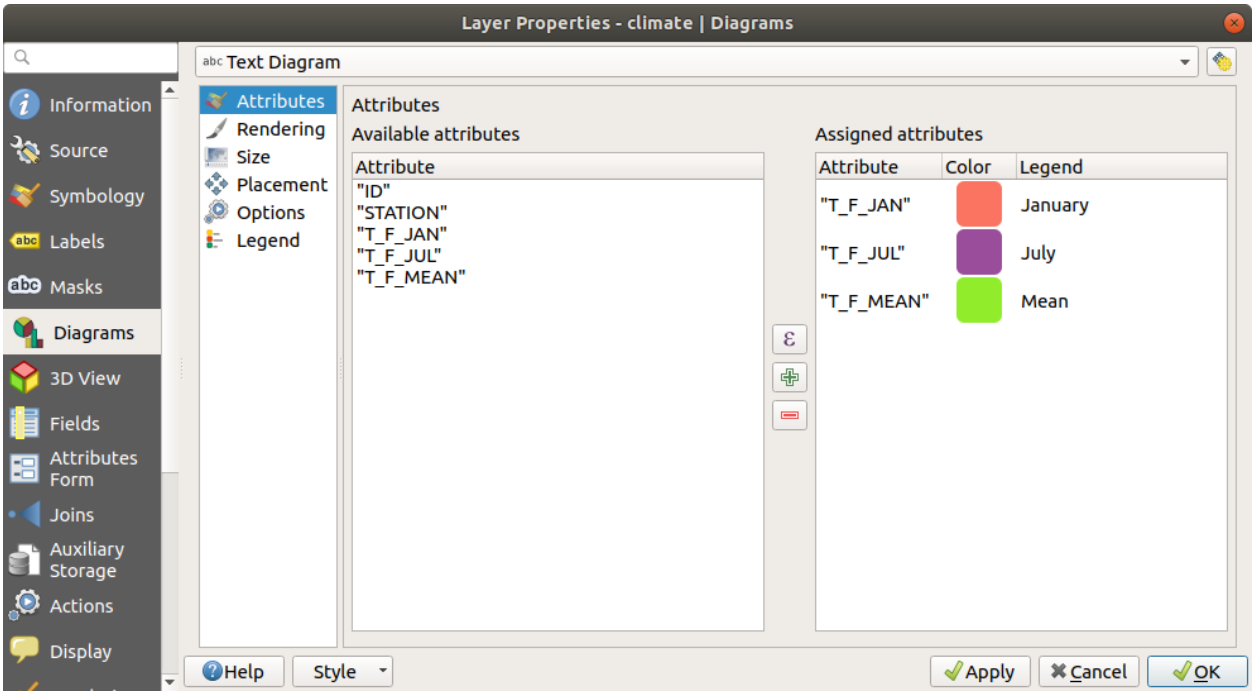



그림 12.34: 도표 속성 - 속성 탭

### 렌더링

*Rendering* 탭은 도표가 어떻게 보일지를 정의합니다. 이 탭은 통계값들을 건드리지 않는, 다음과 같은 일반 설정들을 제공합니다:

- 그래픽의 투명도, 외곽선 너비 및 색상
- 도표 유형에 따라:
  - 히스토그램과 누적 막대 그래프의 경우, 막대의 너비 및 막대 사이의 간격을 설정할 수 있습니다. 누적 막대의 경우 간격을 0 으로 설정하는 편이 좋습니다. 여기에, 라인 심볼 속성 을 사용하면 *Axis line symbol* 을 맵 캔버스에 보이게 하고 사용자 지정할 수 있습니다.
  - 텍스트 도표의 경우, 원의 배경 색상 및 텍스트 용글꼴 을 설정할 수 있습니다.
  - 원형 차트의 경우, 첫 번째 조각의 *Start angle* 과 그 (시계 방향 또는 반시계 방향) *Direction* 을 설정할 수 있습니다.
- 그래픽에그리기 효과 의 사용

이 탭에서 도표의 가시성을 서로 다른 옵션들을 통해 관리 및 미세 조정할 수 있습니다:

- *Diagram z-index*: controls how diagrams are drawn on top of each other and on top of labels. A diagram with a high index is drawn over other diagrams and labels;
- *Show all diagrams*: 서로 중첩되더라도 모든 도표를 표시합니다.
- *Show diagram*: 특정 도표만 렌더링되도록 할 수 있습니다.
- *Always Show*: 특정 도표가 다른 도표 또는 맵 라벨과 중첩하더라도 언제나 렌더링되도록 선택합니다.
- 축척에 따른 가시성 을 설정합니다.

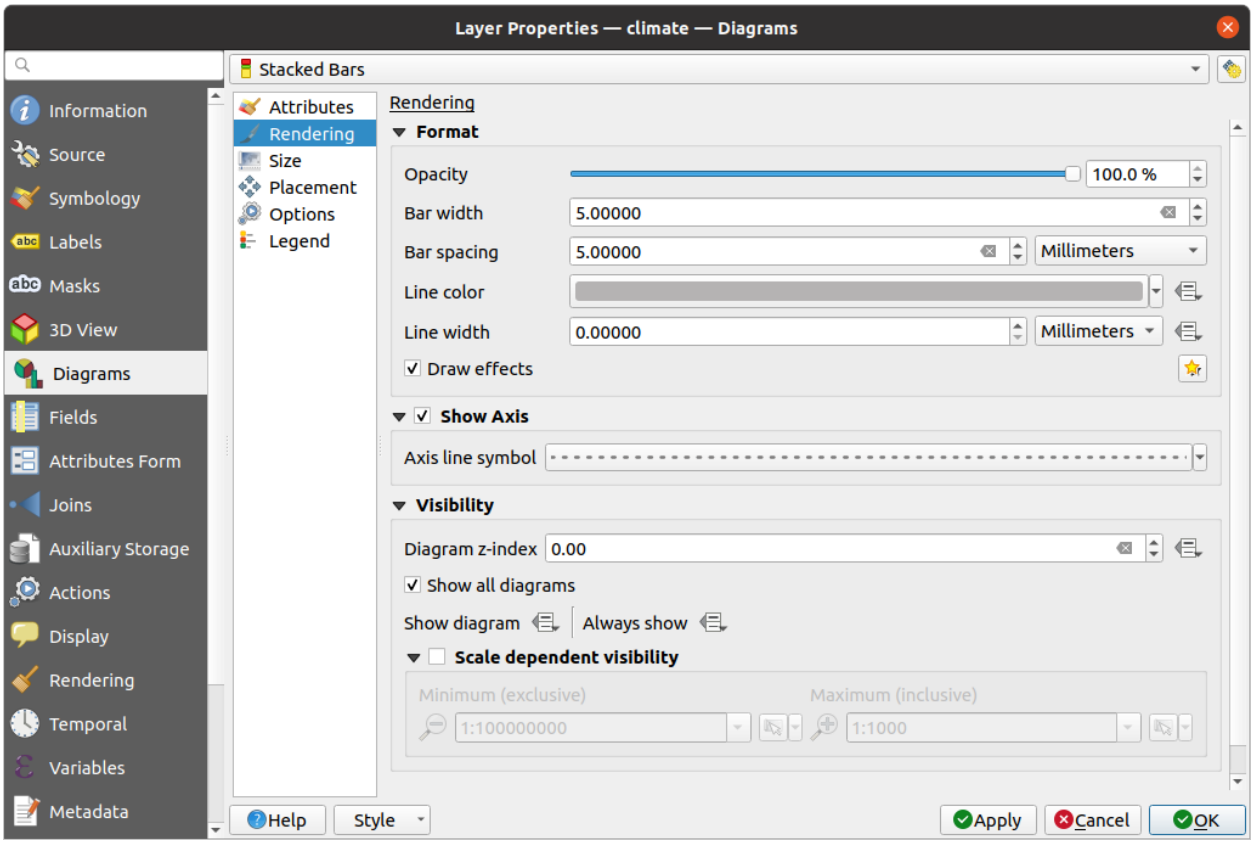



그림 12.35: 도표 속성 - 렌더링 탭

### 크기

*Size* 탭은 선택한 통계를 어떻게 표현할지 설정합니다. 도표의 크기단위 는'Millimeters', 'Points', 'Pixels', 'Map Units' 또는'Inches'입니다. 다음 가운데 하나를 선택할 수 있습니다:

- *Fixed size*: 모든 피처의 그래픽을 유일한 크기로 표현합니다. (히스토그램에는 적용되지 않습니다.)
- *Scaled size*: 레이어 속성을 이용하는 표현식을 기반으로 크기를 결정합니다:
  1. *Attribute* 란에서 필드를 선택하거나 표현식을 작성한 다음
  2. *Find* 를 눌러 해당 속성의 *Maximum value* 를 반환받거나, 위젯에 사용자 지정 값을 입력하십시오.
  3. 히스토그램과 누적 막대 그래프의 경우, 속성의 *Maximum value* 를 나타내는 데 사용되는 *Bar length* 값을 입력하십시오. 각 피처 별로 이 비율을 유지하기 위해 막대 길이를 선형적으로 축척 조정할 것입니다.
  4. 원형 차트와 텍스트 도표의 경우, 속성의 *Maximum value* 를 나타내는 데 사용되는 *Size* 값을 입력하십시오. 각 피처 별로 이 비율을 (0 에서부터) 유지하기 위해 원의 면적 또는 지름을 선형적으로 축척 조정할 것입니다. 하지만 작은 도표의 경우 *Minimum size* 를 설정할 수 있습니다.

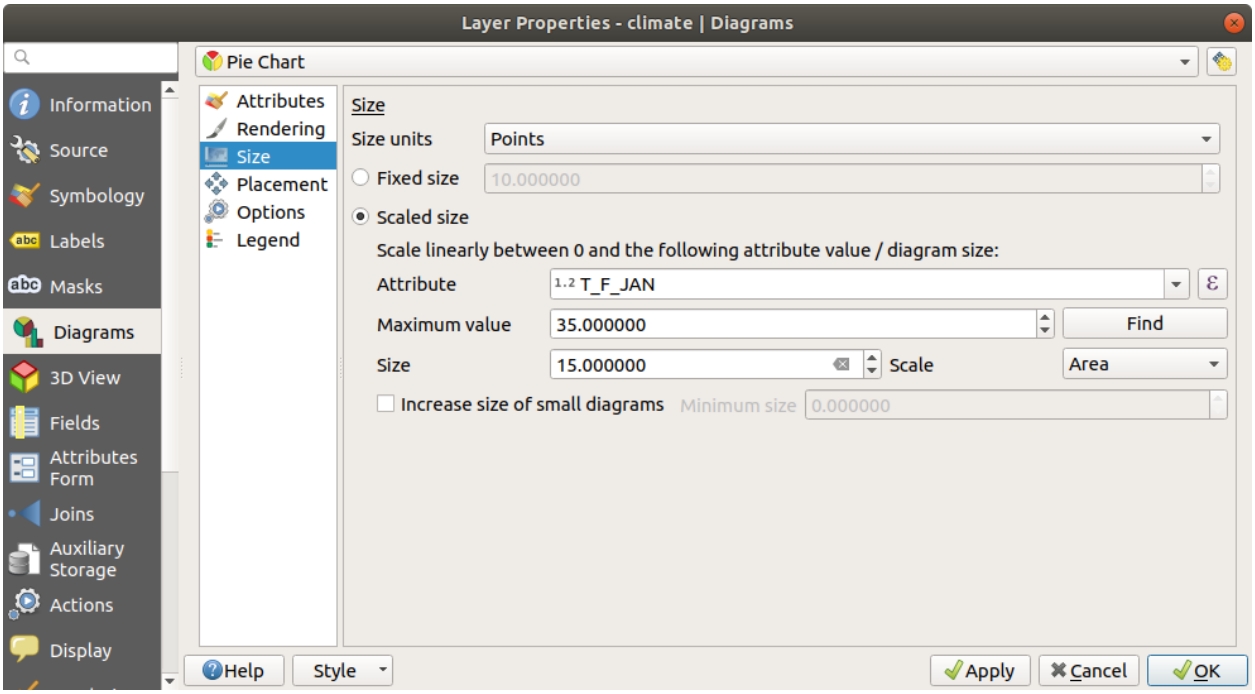



그림 12.36: 도표 속성 - 크기 탭

### 배치

*Placement* 탭에서 도표의 위치를 정의합니다. 레이어 도형 유형에 따라, 배치를 위한 서로 다른 옵션을 제공하고 있습니다 (자세한 내용은배치 참조):

- *Around point* 또는 *Over point*: 포인트 도형 용입니다. 전자는 적용할 반경도 설정해야 합니다.
- *Around line* 또는 *Over line*: 라인 도형 용입니다. 포인트 객체와 마찬가지로, 전자는 어느 한 쪽으로의 거리를 설정해야 합니다. 사용자는 객체를 기준으로 (라인의'above', 'on'그리고/또는'below') 도표 배치를 지정할 수 있습니다. 한 번에 여러 옵션을 함께 선택할 수도 있습니다. 이 경우, QGIS 가 도표의 최적 위치를 찾을 것입니다. 라인 방향을 이용해서 도표의 위치를 설정할 수도 있다는 점을 기억하십시오.

- *Around centroid* (설정된 *Distance* 에), *Over centroid*, *Using perimeter* 및 *Inside polygon*: 폴리곤 객체 용 옵션입니다.

*Coordinate* 그룹: 피처 별로 도표 배치를 직접 제어할 수 있습니다. 피처의 속성 또는 표현식을 사용해서 *X* 및 *Y* 좌표를 설정합니다. 라벨 및 도표 이동 도구를 사용해서 좌표 정보를 채울 수도 있습니다.

In the *Priority* section, you can define the placement priority rank of each diagram, i.e. if there are different diagrams or labels candidates for the same location, the item with the higher priority will be displayed and the others could be left out.

*Discourage diagrams and labels from covering features* defines features to use as *obstacles*, i.e. QGIS will try to not place diagrams nor labels over these features. The priority rank is then used to evaluate whether a diagram could be omitted due to a greater weighted obstacle feature.

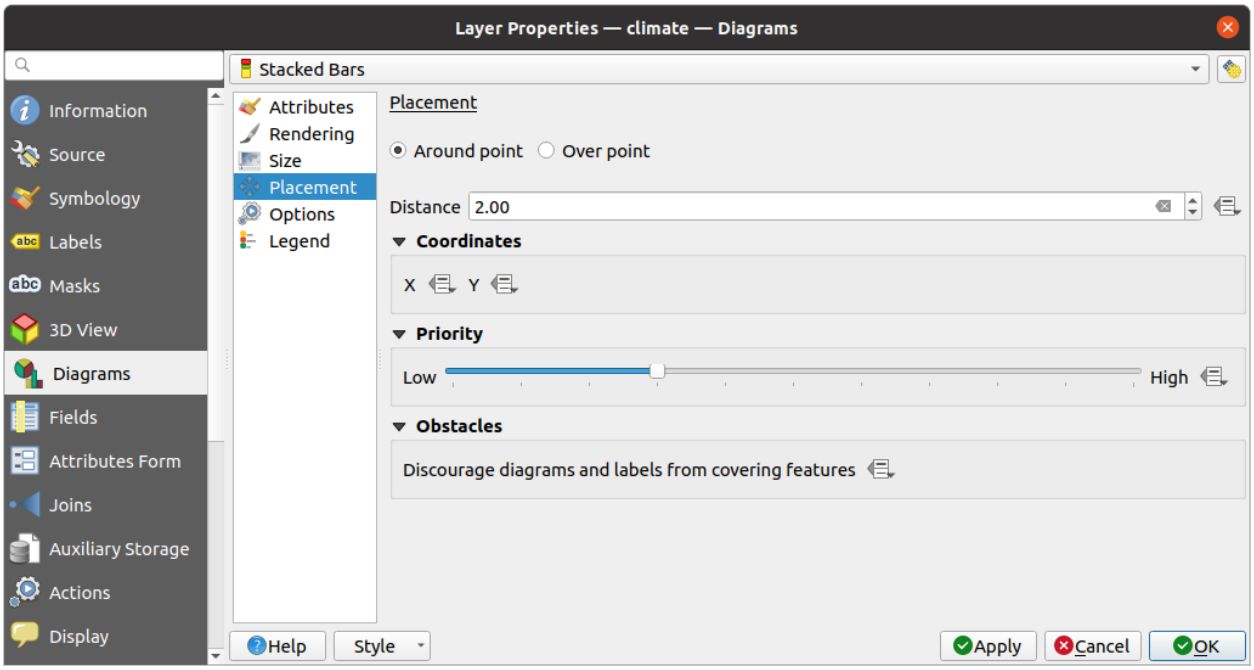



그림 12.37: 도표 속성 - 배치 탭

### 옵션

*Options* 탭에서 히스토그램과 누적 막대 그래프를 설정할 수 있습니다. 수평 및 수직 도표에 대해 *Bar orientation* 이 *Up*, *Down*, *Right* 또는 *Left* 가 되어야 할지 선택할 수 있습니다.

### 범례

*Legend* 탭에서, 레이어 패널 및인쇄 조판기 범례 의 레이어 심볼 옆에 도표 항목을 표시할지 여부를 선택할 수 있습니다:

- *Show legend entries for diagram attributes*: 이 옵션을 체크하면 범례에 *Attributes* 탭에서 할당했던 `Color` 및 `Legend` 속성을 표시합니다.
- 그리고 도표에척도 크기 가 사용된 경우, *Legend Entries for Diagram Size*…버튼을 클릭하면 범례에 있는 도표 심볼 속성의 환경을 설정할 수 있습니다. 이 버튼은 *Data-defined Size Legend* 대화창을 여는데, 데이터 정의 크기 범례 에서 이 대화창의 옵션을 설명하고 있습니다.

설정을 마치면, 인쇄 조판기 범례의 레이어 심볼 옆에도 도표 범례 항목이 (색상 및 도표 크기 속성이) 표시됩니다.

### Stacked Diagrams

Stacked diagrams allow users to create complex diagrams like population pyramids, where two subdiagrams, namely histograms, are located side by side and displayed horizontally.

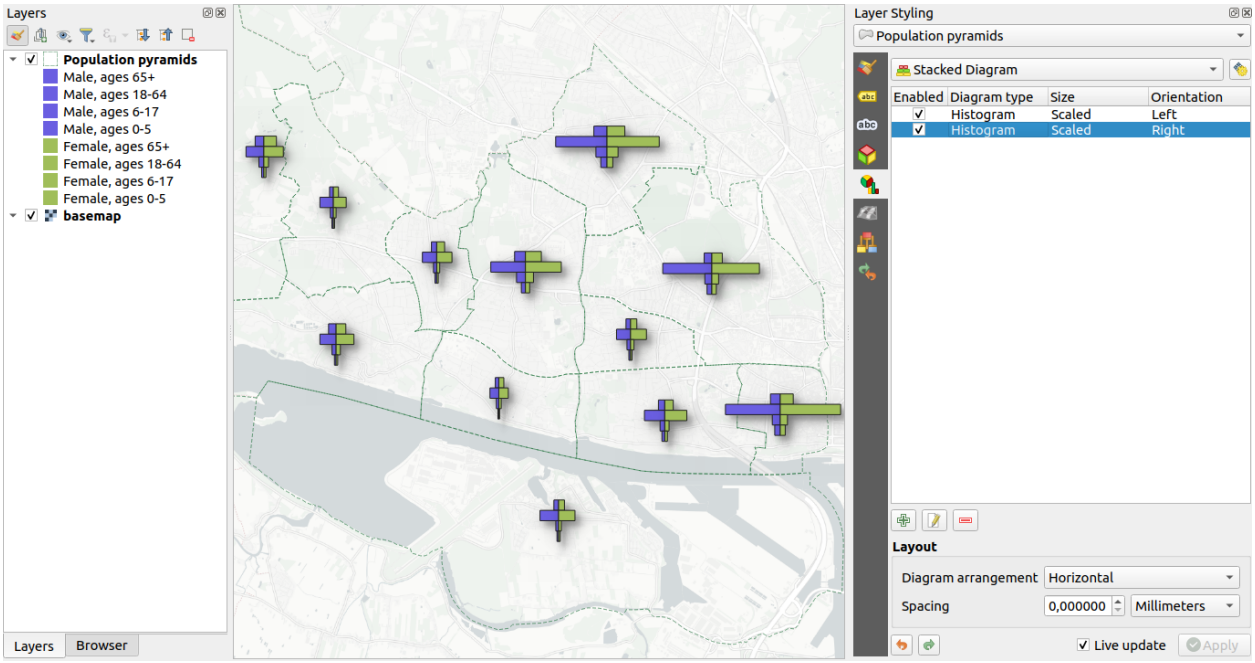



그림 12.38: Population pyramids built for each layer feature

Multi-temporal diagrams can also be constructed as stacked diagrams. The number of subdiagrams, as well as the spacing between them can be configured.

Moreover, subdiagrams can have different types (e.g., a pie chart alongside a histogram) and have their own independent settings like *Attributes*, *Rendering*, *Size*, *Options* and *Legend*.

*Placement* settings in a stacked diagram, as well as some visibility settings (located in the *Rendering* tab), are determined by the placement and visibility settings of the first subdiagram in the stack.

Finally, subdiagram ordering is given by the item ordering in the Stacked Diagram's list. The first subdiagram appears to the left in a horizontal stacked diagram, or in the upper part of a vertical one.

## 12.1.6 마스크 속성

*Masks* 탭에서 다른 모든 레이어의 심볼 레이어 또는 라벨과 중첩하는 현재 레이어 심볼을 환경 설정할 수 있습니다. 즉 중첩하는 경우 색상이 비슷해 구분하기 힘든 심볼 및 라벨의 가독성을 향상시킬 수 있다는 뜻입니다. 항목 주변에 사용자 지정한 투명 마스크를 추가해서 현재 레이어의 심볼 레이어의 일부분을"숨깁니다".

활성화된 레이어에 마스크를 적용하려면, 먼저 프로젝트에서마스크 심볼 레이어 또는마스크 라벨 을 켜야 합니다. 그 다음 *Masks* 탭에서 다음을 확인하십시오:

- *Masked symbol layers*: 현재 레이어의 모든 심볼 레이어를 트리 구조로 목록화합니다. 이 목록에서 선택한 마스크 소스와 중첩하는 경우 투명하게"누락"시키고자 하는 심볼 레이어 항목을 선택할 수 있습니다.
- *Mask sources* 탭: 프로젝트에 정의되어 있는 모든 마스크 라벨과 마스크 심볼 레이어들의 목록입니다. 선택한 마스크 심볼 레이어를 덮는 마스크를 생성할 항목을 선택하십시오.

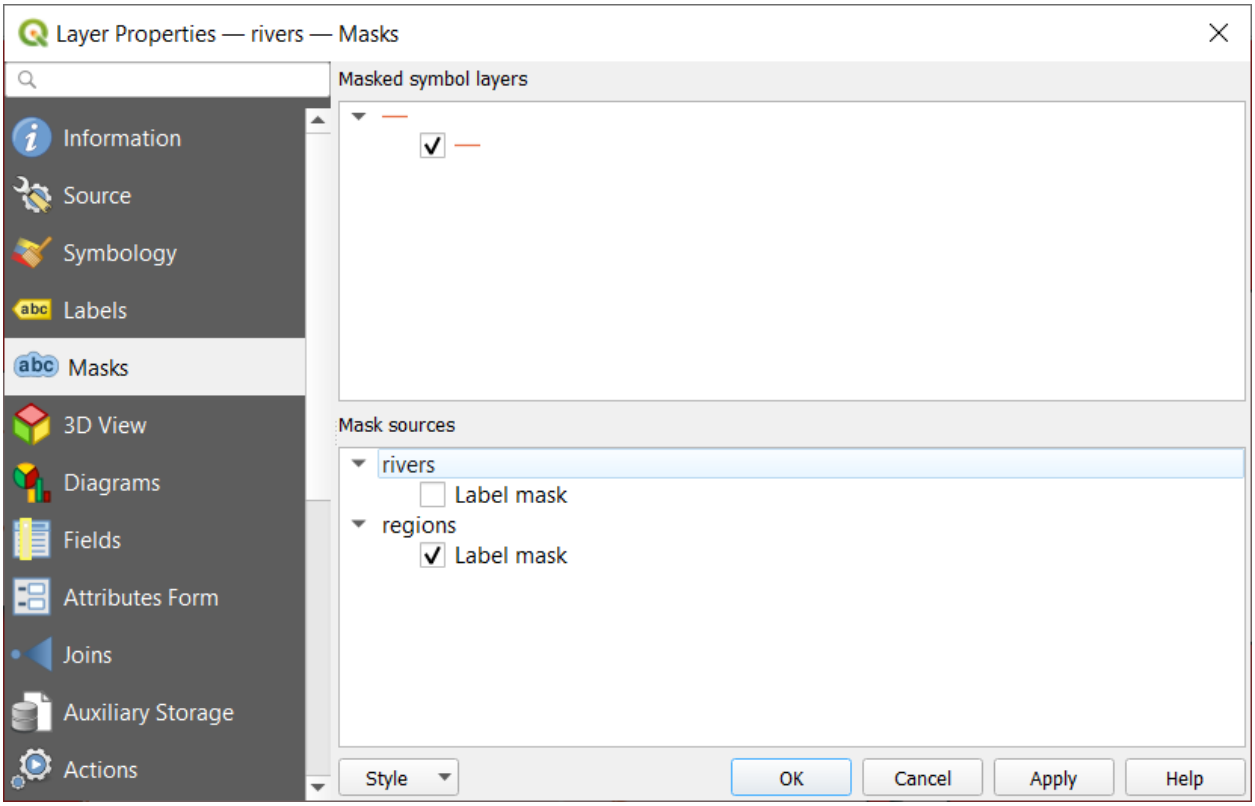



그림 12.39: 레이어 속성 - 마스크 탭

### 12.1.7 3 차원 뷰 속성

*3D View* 탭은*3D* 맵 뷰 도구에서 묘사되어야 할 벡터 레이어를 위한 설정을 제공합니다.

레이어를 3 차원으로 표시하려면, 탭 상단에 있는 콤보박스에서 다음 가운데 하나를 선택하십시오:

- *Single symbol*: 속성이데이터 정의 되었을 수도 있고 아닐 수도 있는 공통 3D 심볼을 사용해서 피처를 렌더링합니다. 각 레이어 도형 유형에 대한 자세한 내용은*3D* 심볼 설정하기 를 읽어보십시오.
- *Rule-based*: 표현식 기반과 축척 범위를 기반으로 여러 심볼의 환경을 정의하고 선택적으로 적용시킬 수 있습니다. 그 방법에 대한 자세한 내용은규칙 기반 렌더링 을 참조하세요.

> **주의**
>
> **심볼 표고와 지형 설정에는 *Elevation* 탭을 사용하는 편이 좋습니다.**
>
> *3D View* 탭에서 피처의 표고 및 고도와 관련된 (*Altitude clamping*, *Altitude binding*, *Extrusion* 또는 *Height* 등의) 속성은 레이어의표고 속성 으로부터 기본값을 상속받기 때문에, *Elevation* 탭 안에서 설정하는 편이 좋을 것입니다.

더 나은 성능을 위해 배후에서 멀티스레드 작업을 통해 벡터 레이어의 데이터를 불러온 다음, 타일로 렌더링합니다. 이때 3D 뷰 탭의 *Layer rendering* 부분에서 타일의 크기를 제어할 수 있습니다.

- *Zoom levels count*: 사분 트리 (quadtree) 의 깊이를 결정합니다. 예를 들면 확대/축소 수준이 1 인 경우 전체 레이어를 단일 타일로 나타냅니다. 확대/축소 수준이 3 이라면 지엽 (leaf) 수준에서 타일이 16 개가 됩니다. (모든 추가적인 확대/축소 수준은 타일 개수가 4 배수가 됩니다.) 기본값은 3 이며, 최대값은 8 입니다.

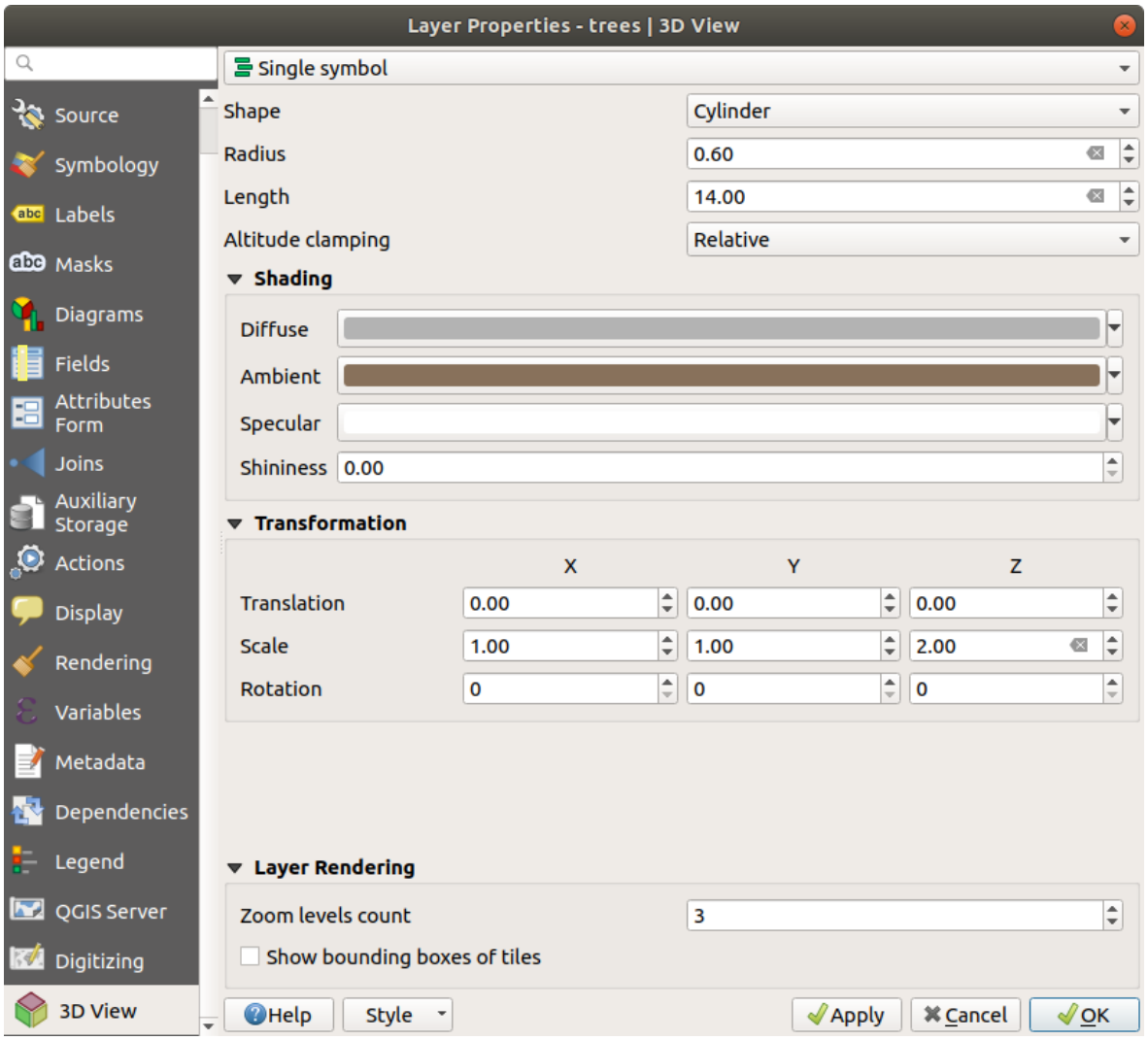



그림 12.40: 포인트 레이어의 3 차원 속성

- *Show bounding boxes of tiles*: 표시되어야 할 타일이 표시되지 않는 문제가 있을 경우 특히 유용한 옵션입니다.

### 12.1.8 필드 속성

*Fields* 탭은 레이어 관련 필드에 대한 정보를 제공하며, 레이어 필드를 정리할 수 있습니다.

Toggle editing mode 버튼으로 레이어를편집 가능 모드로 만들 수 있습니다. 레이어가 편집 가능 모드일 경우, New field 및 Delete field 버튼을 사용해서 레이어 구조를 수정할 수 있습니다.

새 필드 New field 를 생성할 때, 주석 (comment) 편집을 허용하는 데이터소스에 대해서만 *Comment* 옵션을 사용할 수 있습니다. 또한, 지원하는 OGR 포맷 (지오패키지와 ESRI 파일 지오데이터베이스) 의 경우 *Add Field* 대화창 안에서 별명 (alias) 을 설정할 수도 있습니다.

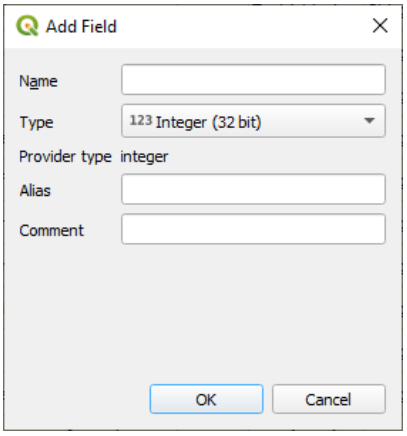



그림 12.41: 필드 추가 대화창

필드명을 더블클릭하면 필드를 재명명할 수도 있습니다. PostgreSQL, Oracle, 메모리 레이어 및 QGIS 버전에 따라 달라지는 몇몇 GDAL 레이어와 같은 데이터 제공자만 이 기능을 지원합니다.

기저 데이터소스 또는양식 속성 에서 설정한 경우, 필드의 별명 (alias) 도 표시됩니다. 별명이란 피처 양식 또는 속성 테이블에서 사용할 수 있는 사람이 읽을 수 있는 필드명입니다. 별명은 프로젝트 파일에 저장됩니다.

가상 필드 와보조 저장소 그리고 데이터셋에 포함되어 있는 필드들 이외에도, *Fields* 탭은 모든결합된 레이어 의 필드도 목록화합니다. 필드가 어디에 속해 있느냐에 따라 서로 다른 배경색이 적용됩니다.

대화창은 목록의 각 필드에 대해 필드의 *Type*, *Type name*, *Length* 및 *Precision* 같은 읽기 전용 특성도 목록화합니다.

데이터 제공자에 따라, 예를 들면 필드 생성 시에, 필드에 코멘트를 결합시킬 수 있습니다. 데이터 제공자가 이 정보를 가져와 *Comment* 열에 표시하며, 이후 피처 양식에 있는 필드 라벨 위에 마우스를 가져가면 코멘트를 표시합니다.

*Configuration* 열에서, 필드가 특정 상황에서 어떤 습성을 보여야 하는지 설정할 수 있습니다:

- `Not searchable`: 위치 검색란 에서 해당 필드를 쿼리하지 못 하도록 하려면 이 옵션을 선택하십시오.
- `Do not expose via WMS`: QGIS 서버에서 해당 레이어를 WMS 로 서비스하고 있는 경우 해당 필드를 표시하지 못 하도록 하려면 이 옵션을 선택하십시오.
- `Do not expose via WFS`: QGIS 서버에서 해당 레이어를 WFS 로 서비스하고 있는 경우 해당 필드를 표시하지 못 하도록 하려면 이 옵션을 선택하십시오.

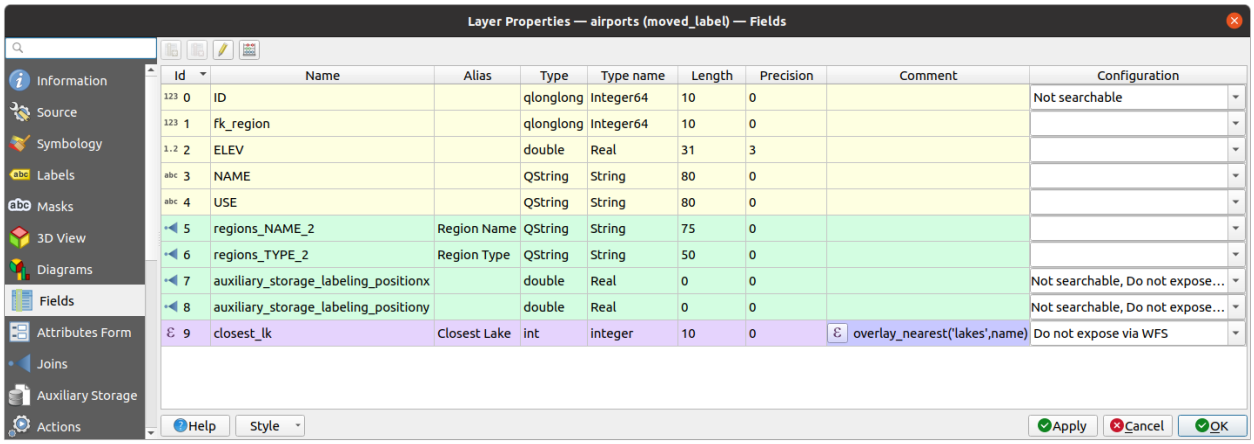



그림 12.42: 필드 속성 탭

### 12.1.9 속성 양식 속성

The *Attributes Form* tab helps you set up the form to display when creating new features, editing or querying existing one. This affects them in both *table and form views*. You can define:

- the look and the behavior (label, visibility, widget, constraints⋯) of each field in the current layer, the *joined layers* and the *relational ones*;
- the placement of enabled *actions*;
- the form's structure (custom or autogenerated);
- extra HTML, QML, text widgets to enhance the form;
- 양식 또는 필드 위젯과 대화형 작업을 처리하기 위한 파이썬 추가 논리

대화창 우상단에서 새 피처 생성시 양식을 기본적으로 열지 여부를 설정할 수 있습니다. *Settings* ▸ *Options* ▸ *Digitizing* 메뉴의 *Suppress attribute form pop-up after feature creation* 옵션을 통해 레이어별 또는 전체 수준으로 환경 설정할 수 있습니다.

#### 사용자 데이터용 양식을 사용자 지정하기

Identify Features 도구로 피처를 클릭하거나 또는 속성 테이블을 양식 뷰 *(form view)* 모드로 바꾸는 경우, 기본적으로 QGIS 는 기본 양식을 미리 정의된 위젯과 함께 표시합니다. (일반적으로 스핀박스 (spinbox) 와 텍스트 상자로 이루어집니다. 지정된 행에 각 필드를 위젯 옆에 있는 필드 라벨로 표현합니다.) 만약 레이어에관계 가 설정돼 있다면, 양식의 하단에 내장된 프레임에 참조 레이어의 필드를 동일한 기본 구조대로 표시합니다.

이 렌더링은 *Layer properties* ▸ *Attributes Form* 탭에 있는 *Attribute editor layout* 설정의 기본값 `Autogenerate` 의 결과입니다. 이 속성은 서로 다른 세 가지 값을 선택할 수 있습니다:

- `Autogenerate`: 양식에 대해"1 행 - 1 필드"라는 기본 구조를 유지하지만 대응하는 각 위젯을 사용자 지정할 수 있습니다.
- `Drag-and-drop designer`: 위젯 사용자 지정 외에도 양식 구조를 더 복잡하게, 예를 들어 그룹 및 탭에 위젯을 내장시키도록 만들 수 있습니다.
- `Provide ui file`: Qt 설계자 파일을 사용할 수 있게 해줍니다. 따라서 잠재적으로 더 복잡하고 완전한 기능을 가진 템플릿을 피처 양식으로 활용할 수 있습니다.

Just below the *Attribute editor layout* setting, a search box allows you to filter fields, relations, containers and other available widgets.

For items that support aliases, like fields or relations, the filter targets both names and aliases.

The Show field aliases instead of names button allows you to toggle field aliases and field names.

When showing aliases, fields without an alias set will have their names displayed in light gray to help identify them easily.

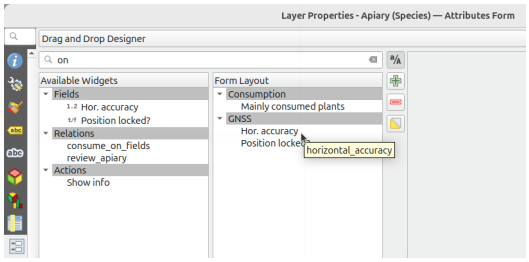



그림 12.43: Search fields using aliases and names in Attributes Form page

### 자동 생성 양식

When the `Autogenerate` option is on, the *Available widgets* panel displays lists of fields (from the current layer, the join and relation layers) and the enabled actions that would be shown in the form. Select a field and you can configure its appearance and behavior in the right panel:

- 필드에사용자 지정 라벨과 자동화 점검 을 추가할 수 있습니다.
- 사용할특정 위젯 을 설정할 수 있습니다.

In this editor mode, only the properties can be viewed for the actions. Moreover, they will be displayed through an *Actions* drop-down menu in the feature form, and following the *Show in attribute table* property in other contexts.

### 드래그 & 드롭 설계자

드래그 & 드롭 설계자는 그림 12.44 에서 볼 수 있는 바와 같이 (HTML/QML 위젯 또는 레이어에 정의된액션 처럼) 특정 필드에 직접 링크되어 있지 않은 속성 필드 또는 다른 위젯을 나타내기 위한 컨테이너 (탭 또는 그룹) 를 몇 개 가지고 있는 양식을 생성할 수 있게 해줍니다.

1. *Select attribute layout editor* 콤보박스에서 `Drag and drop designer` 를 선택하면, *Available widgets* 패널 옆에 있는 *Form Layout* 패널을 기존 필드로 채워서 활성화합니다. 선택한 필드는 세 번째 패널에 자신의속성 을 표시합니다.
2. 사용자의 *Form Layout* 패널에서 사용하지 않고자 하는 필드를 선택한 다음 버튼을 눌러 제거하십시오. Invert selection 버튼으로 선택한 필드를 반전시킬 수 있습니다.
3. 필드를 첫 번째 패널에서 *Form Layout* 패널로 드래그 & 드롭해서 다시 추가시키십시오. 같은 필드를 중복해서 추가할 수 있습니다.
4. *Form Layout* 패널에 있는 필드를 드래그 & 드롭하면 필드 위치를 재배열할 수 있습니다.
5. 동일 카테고리에 속한 관련 필드에 컨테이너를 추가하면 더 나은 양식 구조를 만들 수 있습니다.
    1. 먼저 Add a new tab or group to the form layout 아이콘을 클릭해서 필드 및 기타 그룹을 표시할 탭을 생성하십시오.

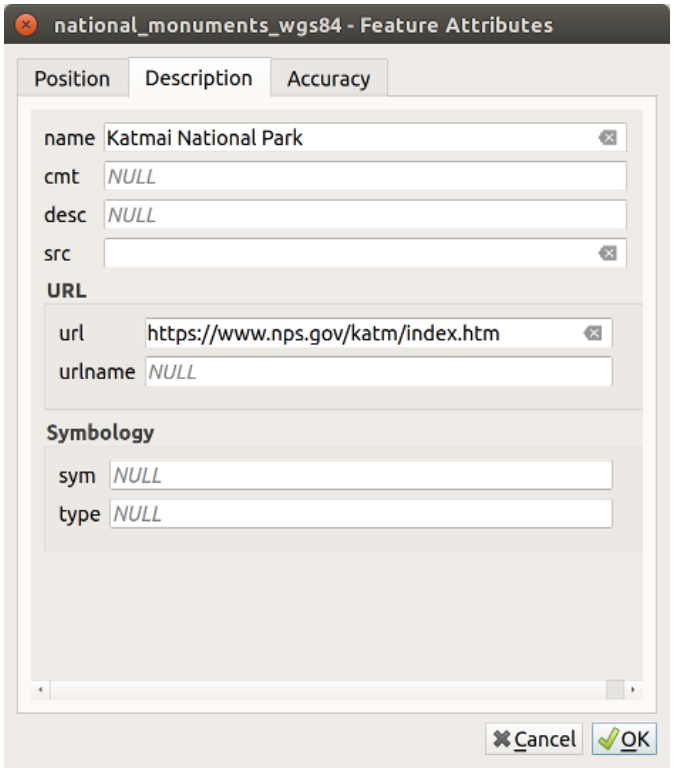



그림 12.44: 탭과 명명된 그룹을 가진 내장 양식 생성의 결과물

2. Then set the properties of the container, i.e.:
   - *Label*: 컨테이너에 적용될 제목입니다.
   - *Container Type*: 컨테이너 유형은 *Tab*, *Group box in container* (탭 또는 또다른 그룹 안에서 접을 수 있는 그룹 상자) 또는 *Row* (위젯의 개수에 따라 열 개수를 자동으로 결정하는, 사용자 위젯을 수평 행으로 배열할 수 있게 해주는 컨테이너 유형) 가운데 하나로 설정할 수 있습니다.
   - *Within*: 이 부가적인 기능을 이용하면 새 (*Group box in container* 또는 *Row* 유형) 컨테이너를 내장시킬 기존 컨테이너를 선택할 수 있습니다.
   - *Number of columns*: 내장된 필드가 배포되어야 할 열의 개수입니다.

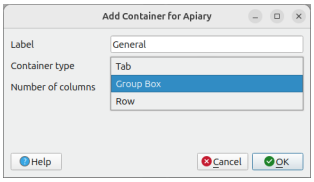



그림 12.45: **속성 편집기 조판** 으로 컨테이너를 생성하는 대화창

이후에 항목을 선택한 다음 세 번째 패널에서 다음 설정을 하면 이 속성들과 기타 속성들을 업데이트할 수 있습니다:

- 컨테이너의 라벨을 숨기기 또는 표시하기
- 컨테이너 이름을 변경하기
- 열의 개수를 설정하기

- 컨테이너의 가시성을 제어하는 표현식을 입력하기. 양식의 값들이 변경될 때마다 표현식을 다시 평가하고 그에 따라 탭 또는 그룹 상자를 표시하거나 숨길 것입니다.
- *Show as Group Box*: 이 옵션을 체크하면 탭을 최상위 그룹 상자로 변환하고, 해제하면 그룹 상자를 탭으로 변환합니다.
- 그룹 상자의 경우, 양식을 열 때 설정하느냐 하지 않느냐에 따라 모든 피처를 *Collapsed* 접힌 상태로 표시하거나 또는 표현식 (*Control collapsed by expression*) 과 일치하는 피처만 접힌 상태로 표시하거나 할 것입니다.
- 컨테이너의 *Style* 모양을 사용자 지정 *Background color*, 라벨 색상 및 글꼴 속성으로 환경설정하기

3. Add a new tab or group to the form layout 아이콘을 다시 눌러서 사용자가 원하는 만큼 많은 컨테이너를 생성하고 내장시킬 수 있습니다.

6. 다음 단계는 각 컨테이너에 관련 필드를 할당하는 것입니다. 그냥 드래그 & 드롭하면 됩니다. 그룹 상자 및 탭도 같은 방법으로 옮길 수 있습니다.
7. 사용 중인 필드의위젯을 사용자 지정 하십시오.
8. 레이어가일대다 또는 다대다 관계 를 가진 경우 관계 이름을 *Available Widgets* 패널에서 *Form Layout* 패널로 드래그 & 드롭하십시오. 현재 레이어의 양식에 선택한 위치로 관련 레이어 속성 양식을 내장시킬 것입니다. 다른 항목의 경우, 관계 라벨을 선택하면 몇몇 속성을 환경 설정할 수 있습니다:
   - 관계 라벨을 숨기기 또는 표시하기
   - 링크 버튼 표시하기
   - 링크 해제 버튼 표시하기
9. 레이어가 `Layer` 또는 `Feature` 범위에 하나 이상의액션 을 활성화한 상태인 경우, *Actions* 아래에 해당 액션들을 목록화할 것입니다. 다른 필드와 마찬가지로, 이 액션을 *Form Layout* 패널로 드래그 & 드롭할 수 있습니다. 현재 레이어의 양식에 선택한 위치로 관련 액션을 내장시킬 것입니다.
10. *Other Widgets* 에서 (기타 위젯 참조) 하나 이상의 위젯을 추가해서 양식을 좀 더 사용자 지정해보십시오.
11. 레이어 속성 대화창의 *Apply* 버튼을 클릭합니다.
12. 피처 속성 양식을 (예를 들어 Identify features 도구를 사용해서) 열면, 새 양식을 표시할 것입니다.

### 기타 위젯

드래그 & 드롭 설계자는 특정 필드에 연결되지 않은 위젯을 몇 개 제공하고 있습니다. 이런 위젯을 이용해서 양식의 모양을 향상시키거나 또는 동적으로 계산된 값들을 표시할 수 있습니다.

- *HTML Widget*: HTML 페이지를 내장하고 있습니다. 이 HTML 소스가 동적으로 계산된 표현식의 결과물을 담고 있을 수도 있습니다.

  예를 들어 필드에 BLOB(Binary Large Object) 으로 저장된 이미지 (`photo` 라고 합시다) 를 표시하는 데 HTML 위젯을 사용할 수 있습니다:

  1. `Drag-and-drop designer` 모드에서 사용자의 *Form Layout* 에 *HTML Widget* 을 추가합니다.
  2. HTML 위젯을 환경설정하려면 *HTML Widget* 을 더블클릭하십시오.
  3. 기본 *Title* 을 변경하거나 숨기십시오.
  4. ε 버튼을 누른 다음 다음 QGIS 표현식을 입력하십시오:

```
'<img src= "' || to_base64("photo") || '">'
```

`photo` 를 사용자의 BLOB 필드 이름으로 반드시 바꾸십시오. 이 표현식은 BLOB 파일이 인코딩되어 있는 HTML 이미지 태그를 가진 문자열을 생성합니다.

5. 대화창의 *Apply* 버튼을 클릭한 다음 버튼을 누르십시오.
6. QGIS 가 사용자 표현식을 평가하기 위해 자동으로 HTML 서식과 함수들을 적용시켜 다음과 같은 코드를 생성할 것입니다:

```
<script>document.write(expression.evaluate("'<img src=\"
↪' || to_base64(\"photo\") || '\">'"));</script>
```

사용자 이미지의 미리보기가 오른쪽에 표시됩니다.

- *QML Widget*: embeds a Qt QML document, displaying graphical elements in the attribute form. Beside the custom *Title* of the added widget and whether it should be shown or not, you can select from predefined *QML code* elements:
  - *Free text*⋯: allows you to write from scratch or paste an existing code
  - *Rectangle*: provides a minimal code for displaying a rectangle
  - *Bar chart*: provides a minimal code for displaying a bar chart
  - *Pie chart*: provides a minimal code for displaying a pie chart

  You can extend the code with QML syntax, use layer fields or QGIS expressions that are dynamically calculated.

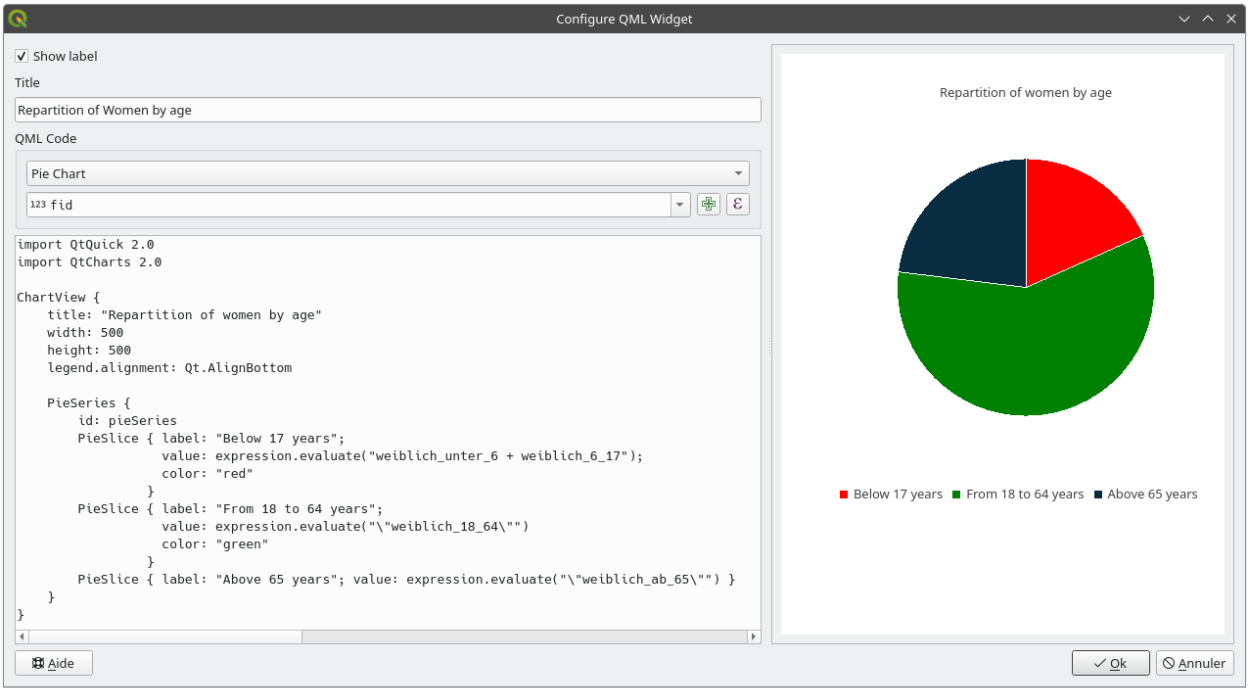



그림 12.46: Setting a QML graph to display in attribute form

- *Text Widget*: 기본 HTML 마크업 언어를 지원하며 동적으로 계산된 표현식 결과를 담고 있을 수도 있는 텍스트 위젯을 표시합니다.
- *Spacer Widget*: 두 위젯 사이의 수직 거리를 벌리기 위한 빈 투명 사각형을 삽입합니다.

> **팁**
>
> **동적 내용 표시**
>
> 앞에서 설명한 위젯들은 (*Spacer Widget* 은 제외) 양식에 있는 또다른 필드의 값이 변할 때마다 동적으로 변경되는 내용을 표시하기 위한 표현식을 지원합니다. 또다른 필드의 값을 조사하려면 표현식에 `current_value('field_name')` 함수를 사용하면 됩니다.

### 사용자 지정 UI 파일 사용하기

`Provide ui-file` 옵션은 Qt 설계자로 만든 복잡한 대화창을 사용할 수 있게 해줍니다. UI 파일을 이용하면 대화창을 엄청난 자유도로 생성할 수 있습니다. 레이어의 필드에 그래픽 객체 (텍스트 상자, 콤보박스 등등) 를 링크시키려면 객체를 필드와 동일한 이름으로 명명해야 한다는 사실을 기억하십시오.

사용할 파일을 가리키는 경로를 정의하려면 *Edit UI* 를 이용하십시오.

원격 서버에 UI 파일을 호스팅할 수도 있습니다. 이 경우, *Edit UI* 양식에 파일 경로 대신 URL 을 지정하십시오.

QGIS-training-manual-index-reference 의 새 양식 생성 에서 몇몇 예시를 찾아볼 수 있습니다. 더 심화된 내용을 알고 싶다면, https://woostuff.wordpress.com/2011/09/05/qgis-tips-custom-feature-forms-with-python-logic/ 을 참조하세요.

### 사용자 지정 함수로 사용자 양식 향상시키기

QGIS 양식은 대화창을 열 때 호출되는 파이썬 함수를 보유할 수 있습니다. 이 함수를 이용해서 사용자 대화창에 추가적인 논리를 추가하십시오. 다음 세 가지 서로 다른 방법으로 양식 코드를 지정할 수 있습니다:

- `load from the environment`: 예를 들어 `startup.py` 파일 또는 설치한 플러그인에 있는 함수를 이용합니다.
- `load from an external file`: 원격 파일용 URL 을 입력하거나 또는 사용자의 파일 시스템에서 파이썬 파일을 선택할 수 있는 파일 선택기를 엽니다.
- `provide code in this dialog`: 사용자가 사용할 함수를 직접 입력할 수 있는 파이썬 편집기를 엽니다.

세 가지 방법 모두 사용자가 호출할 함수명을 입력해야만 합니다. (다음 예시에서는 `open` 입니다.)

다음은 (`MyForms.py` 모듈에 있는) 예시입니다:

```
def open(dialog,layer,feature):
    geom = feature.geometry()
    control = dialog.findChild(QWidget,"My line edit")
```

파이썬 초기화 함수 (Python Init Function) 에서도 마찬가지로 `open` 으로 참조합니다.

### 필드 습성 환경 설정

*Attributes Form* 의 주요 부분에서 속성 테이블 또는 피처 양식에 있는 필드의 값을 입력하거나 표시하는 데 쓰이는 위젯의 유형을 설정할 수 있습니다. 사용자가 각 필드와 값 또는 각 필드에 추가할 수 있는 값의 범위와 어떻게 대화형 작업을 할지 정의할 수 있습니다.

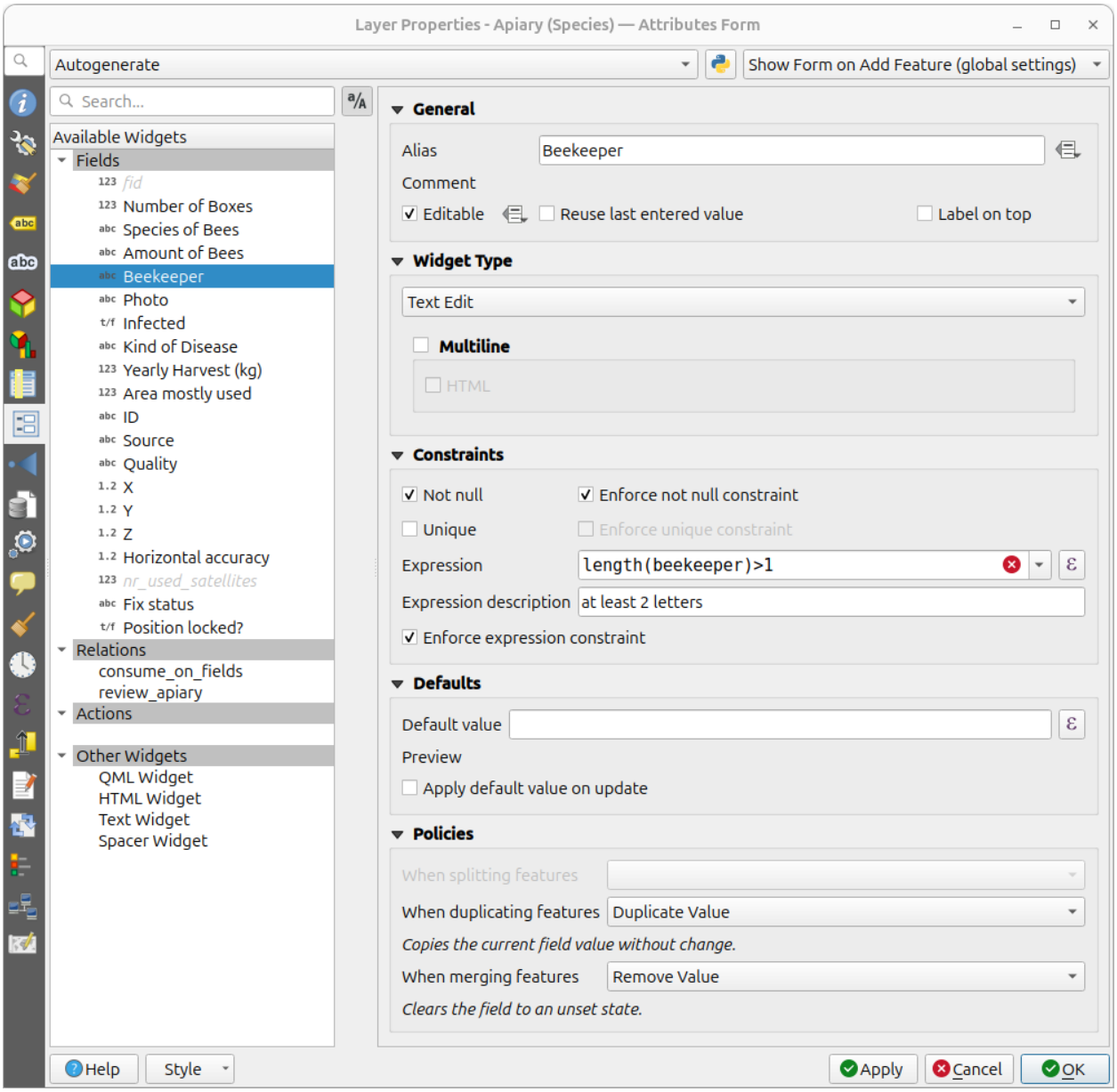



그림 12.47: 속성 열을 위한 편집 위젯을 선택할 수 있는 대화창

> **팁**
>
> **Copy and paste widget configuration**
>
> The field configuration can be copied to the clipboard and then pasted to another field via right click on a field in the *Available widgets* panel.

This comes handy to save time and to make sure that common settings are correctly applied to several related fields, e.g., when configuring latitude and longitude field widgets.

Copy and paste of widget configuration can be done between fields of the same layer, between fields from different layers in the QGIS project, or between fields from layers in different QGIS instances.

Note that neither aliases nor comments are copied, since those are specific to each particular field.

### 공통 설정

필드에 적용된 위젯의 유형에 상관없이, 필드를 편집할 수 있는지 없는지 그리고 필드를 어떻게 편집할 수 있는지를 제어하도록 설정할 수 있는 몇몇 공통 속성이 존재합니다.

### 위젯 표시

드래그 & 드롭 설계자 모드에서만 사용할 수 있는 이 그룹은 사용자가 필드에 할당된 위젯의 모양을 환경설정하는 데 도움을 줄 것입니다:

- *Show label*: 양식에 필드명을 표시해야 하는지 여부를 나타냅니다.
- *Override label color*: 필드의 라벨에 특정 색상을 적용합니다.
- *Override label font*: 필드의 라벨에 특정 글꼴 속성 (볼드체, 이탤릭체, 밑줄, 취소선, 또는 글꼴 계열) 을 적용합니다.
- *Size*: 속성 양식의 크기를 조정할 때 위젯의 크기를 그에 맞춰 어떻게 조정할지에 대한 방법을 제어할 수 있습니다.
  - *Horizontal stretch*: 수평 공간이 더 필요한 위젯에 대해 더 큰 수평 값을 설정합니다.
  - *Vertical stretch*: 양식의 크기를 조정할 때 위젯의 수직 크기를 어떻게 조정할지에 대한 방법을 결정합니다.

### 일반 옵션

- *Alias*: 필드용으로 쓰이는, 사람이 읽을 수 있는 이름입니다. 이 별명은 피처 양식, 속성 테이블 또는 *Identify results* 패널에 표시될 것입니다. 표현식 작성기 에서 필드명 대신 사용될 수도 있는데, 표현식을 쉽게 이해하고 검토할 수 있게 해줍니다.
- *Comment*: 필드의 코멘트를 읽기전용 상태로, *Fields* 탭에 표시된대로 표시합니다. 이 정보는 피처 양식에 있는 필드 라벨 위에 마우스를 가져가면 툴팁으로 표시됩니다.
- ☑ *Editable*: 이 설정을 선택 해제하면 레이어가 편집 모드 상태일지라도 필드를 읽기 전용으로 (사용자가 수정할 수 없도록) 설정합니다. 이 설정을 체크해도 제공자의 어떤 편집 제약도 무시하지 않는다는 사실을 기억하십시오. Data-defined override 버튼을 통해 데이터 정의 속성이 이 옵션을 제어하도록 할 수 있습니다.
- ☑ *Reuse last entered value*: 이 필드에 마지막으로 입력된 값을 기억해서 레이어에서 편집되는 다음 피처에 대한 기본값으로 사용합니다.
- ☑ *Label on top*: 피처 양식에 있는 위젯의 위 또는 옆에 필드명을 배치합니다.

### 제약조건

필드에 삽입되는 값을 제약할 수 있습니다. 다음과 같은 제약 조건을 설정할 수 있습니다:

- *Not null*: 사용자가 값을 입력해야 합니다.
- *Unique*: 삽입되는 값이 필드 전체에서 유일하도록 보장합니다.
- 사용자 지정 *expression* 기반: 예를 들어 *col0* 필드의 값이 오직 알파벳 문자만 입력받게 하려면 `not regexp_match(col0,'[^A-Za-z]')` 같은 표현식을 쓰면 됩니다. 사용자가 제약조건을 기억할 수 있도록 간단한 설명을 추가할 수 있습니다.

필드에 값을 추가하거나 필드값을 편집할 때마다, 기존 제약조건이 새 값을 받아서:

- 모든 요구사항을 만족하는 경우, 양식에 있는 필드 옆에 초록색 체크 표시가 나타납니다.
- 모든 요구사항을 만족하지 못하는 경우, 필드가 노란색 또는 주황색으로 바뀌고 그에 대응하는 가위표가 위젯 옆에 표시됩니다. 가위표 위에 마우스를 가져가면 해당 필드에 어떤 제약조건이 적용돼 있는지 알 수 있으므로 값을 수정하면 됩니다:
  - 만족하지 않는 제약조건이 강제가 아니기 때문에 (`soft constraint`) "틀린"값을 변경 사항으로 저장해도 될 때 노란색 가위표가 나타납니다.
  - 주황색 가위표는 무시할 수 없으며, 제약조건을 만족할 때까지 사용자의 수정 사항을 저장할 수 없습니다. *Enforce constraint* 옵션을 활성화했을 때 (`hard constraint`) 주황색 가위표가 나타납니다.

### 기본값

- *Default value*: 새 피처의 경우, 기본적으로 미리 정의된 값 또는표현식 기반 값 으로 필드를 자동적으로 채웁니다. 예를 들어:
  - 필드 생성시 피처의 X 좌표, 길이, 면적 또는 다른 어떤 도형 정보로 필드를 자동적으로 채우려면 `$x`, `$length`, `$area` 를 사용하면 됩니다.
  - 새 피처를 생성할 때마다 필드를 1 씩 증가하는 값으로 채우려면 `maximum("field")+1` 을 사용하면 됩니다.
  - 피처 생성 날짜 & 시간을 저장하려면 `now()` 를 사용하면 됩니다.
  - 표현식에변수 를 사용하면, 예를 들어 작업자 이름 (`@user_full_name`), 프로젝트 파일 경로 (`@project_path`) 등등을 더 쉽게 삽입할 수 있습니다.

  산출되는 기본값의 미리보기가 위젯 하단에 표시됩니다.

  > **참고**
  >
  > `Default value` 옵션은 생성 중인 피처의 다른 어떤 필드에 있는 값도 알지 못합니다. 때문에 이 값들을 결합하는 표현식을 사용하는 것은 불가능할 것입니다. 예를 들어 `concat(field1, field2)` 같은 표현식을 사용하면 작동하지 않을 수도 있습니다.

- *Apply default value on update*: 피처 속성 또는 도형이 변경될 때마다 기본값을 다시 계산합니다. 이 옵션으로 데이터를 마지막으로 수정한 사용자, 피처가 마지막으로 변경된 시간 등과 같은 값들을 손쉽게 저장할 수 있습니다.

### 정책

Field policies determine how values are assigned to fields during various editing operations:

**Split and Duplicate Policies**

These policies apply when *Splitting features* or *Duplicating features*:

- *Duplicate Values*: Keeps the existing value of the field for the new features.
- *Use Default Value*: Resets the field by recalculating its *default value*. If no default value clause exists, the existing value is kept for the new features.
- *Remove Value*: 필드 값을 지우고 설정되지 않은 상태로 만듭니다.
- *Use Ratio Geometries*: 모든 분할 피처의 필드 값들을 기존값에 분할된 부분들의 길이 또는 면적 비율을 곱하는 방식으로 다시 계산합니다.

**Merge Policies**

These policies determine initial values when *Merging features*:

- *Remove Value*: Clears the field to an unset state (data provider may populate default value).
- *Use Default Value*: Uses the default field value set in QGIS. (expressions use a dummy feature with the merged geometry; references to other fields won't work).
- *Use Sum* (numeric fields only): Sums values from all merged features.
- *Use Minimum Value* (numeric fields only): Uses smallest value from the selected features.
- *Use Maximum Value* (numeric fields only): Uses largest value from the selected features.
- *Average Weighted by Geometry* (numeric fields only): New values are computed as the weighted average of the source values.
- *Use Largest Feature*: Uses value from feature with the largest geometry.
- *Set to NULL*: Sets attribute to NULL.

### 편집 위젯

QGIS 는 필드 유형에 따라 자동적으로 기본 위젯 유형을 결정하고 필드에 할당합니다. 이후 사용자가 필드 유형과 호환되는 다른 어떤 위젯으로도 대체할 수 있습니다. 다음과 같은 위젯을 사용할 수 있습니다:

- **바이너리 (BLOB)**: 바이너리 필드에 대해서만 사용할 수 있습니다. 비어 있지 않다면 내장된 데이터 크기를 기본 라벨로 표시합니다. 라벨 옆에 있는 드롭다운 버튼을 누르면:
  - *Embed file*: 필드를 새로운 파일로 대체하거나 채울 수 있습니다.
  - *Clear contents*: 필드에 있는 모든 데이터를 제거할 수 있습니다.
  - *Save contents to file*: 데이터를 디스크 상의 파일로 내보낼 수 있습니다.

  드래그 & 드롭 양식에 예를 들어*QML 또는 HTML 위젯* 을 결합시키면, 내장된 바이너리 파일을 필드에서 미리보기할 수 있습니다.
- **Checkbox**: Displays a checkbox whose state defines the value to insert. Internally, the widget stores a value (e.g. 1 for checked, 0 for unchecked) while representing it visually as a toggled box. You can configure the stored values for the checked and unchecked states in the representation. It is also possible to display the checkbox state as a plain text using "True"/"False"labels. These settings influence how the values are shown in classification renderers.
- **범주화 (Classification)**: 레이어에범주 심볼 이 적용된 경우에만 사용할 수 있으며, 범주들의 값을 가지고 있는 콤보박스를 표시합니다.

- **색상 (Color)**: 색상을 선택할 수 있는색상 위젯 을 표시합니다. 색상값은 속성 테이블에 HTML 서식으로 저장됩니다.
- **날짜 & 시간 (Date/Time)**: 날짜, 시간, 또는 둘 다 입력하기 위한 캘린더 위젯을 열 수 있는 한 줄짜리 텍스트란을 표시합니다. 열이 텍스트 유형이어야만 합니다. 캘린더 팝업창 등의 사용자 지정 서식을 선택할 수 있습니다.
- **일람표 (Enumeration)**: 데이터베이스에서 가져온 미리 정의된 값들을 가지고 있는 콤보박스를 엽니다. 현재 PostgreSQL 제공자가 `enum` 유형인 필드에 대해서만 지원하고 있습니다.
- **Attachment**: Uses a "Open file"dialog to store file path in a relative or absolute mode. It can be used to display the document path as a hyperlink or render the document within a dedicated widget in the form. Supported document types are image, web page, audio and video, and the supported file formats depend on the Operating System. You can also configure an *external storage system* to fetch/store resources.

> **팁**
>
> **첨부 위젯의 상대 경로**
>
> 파일 탐색기를 통해 선택한 경로가 `.qgs` 프로젝트 파일과 동일한 디렉터리에 또는 그 아래 위치하는 경우, 해당 경로를 상대 경로로 변환합니다. 이 습성은 멀티미디어 정보가 첨부된 `.qgs` 프로젝트의 이동성 (portability) 을 향상시킵니다.

- **Geometry**: Compatible with the *Geometry* field type. For example, PostgreSQL and Memory layers can contain multiple geometry fields. QGIS uses the first geometry column as the layer geometry. Applying the Geometry widget to additional geometry fields allows lossless use of the data in the attribute form:
  - Copy in the formats WKT (Well-known text representation, e.g. `POINT (30 10)` ) or GeoJSON
  - Paste
  - Clear
- **숨김 (Hidden)**: 숨겨진 속성 열을 볼 수 없습니다. 사용자가 열의 내용을 확인할 수 없습니다.
- **키/값 (Key/Value)**: 단일 필드 안에 키/값 쌍의 집합을 저장하기 위한 열 2 개짜리 테이블을 표시합니다. 현재 PostgreSQL 제공자가 `hstore` 유형인 필드에 대해서만 지원하고 있습니다.
- **JSON 뷰**: 문법을 강조하는 텍스트 편집기 또는 트리 뷰에 JSON 데이터를 표시합니다. 현재 이 위젯은 읽기 전용입니다. 데이터를 어떻게 표시할지를 변경할 수 있는 옵션이 몇 가지 있습니다. '기본 뷰'는 위젯이 텍스트 모드로 나타나야 할지 또는 트리 모드로 나타나야 할지를 지정합니다. 'JSON 서식'은 다음 트리 뷰 전용 옵션 3 개를 가지고 있습니다:
  - 들여쓰기 (Indented): 데이터를 새줄 문자 (newline) 와 들여쓰기를 위한 공백 문자 4 개를 가진 인간이 읽을 수 있는 양식으로 표시합니다.
  - 조밀 (Compact): 데이터를 새줄 문자나 공백이 없는 한 줄 크기의 최적화된 문자열로 표시합니다.
  - 비활성화 (Disabled): 데이터를 제공자가 보내온 그대로 표시합니다.
- **목록 (List)**: 단일 필드 안에 서로 다른 값들을 추가하기 위한 열 1 개짜리 테이블을 표시합니다. 현재 PostgreSQL 제공자가 `array` 유형인 필드에 대해서만 지원하고 있습니다.
- **범위 (Range)**: 특정 범위에서 숫자 값을 설정할 수 있습니다. 슬라이드 바 또는 스핀박스 위젯 가운데 선택할 수 있습니다.

> **참고**

QGIS 는 GeoPackage 또는 ESRI 파일 지오데이터베이스처럼 사전 정의된 **범위 필드 도메인 (range Field Domains)** 을 가진 일부 레이어들을 자동으로 인식해서 관련 필드들에 **범위** 위젯을 할당할 것입니다. 이 위젯은 해당 도메인에 지정된 최소값과 최대값을 가질 것입니다.

- **Relation Reference**: This is the default widget assigned to the referencing field (i.e., the foreign key in the child layer) when a *relation* is set. It provides direct access to the parent feature's form which in turn embeds the list and form of its children. The number of entries in the widget can be limited for efficiency, and if limit is not set, all entries will be loaded. The field defined in the display expression is used as representable value in classification renderers.
- **텍스트 편집 (Text Edit)** (기본값): 간단한 텍스트 또는 여러 줄의 텍스트를 쓸 수 있는 텍스트 편집란을 엽니다. 여러 줄 유형을 선택한 경우 HTML 내용도 선택할 수 있습니다.
- **단일값 (Unique Values)**: 속성 테이블에서 이미 사용된 값들 가운데 하나를 선택할 수 있습니다. 'Editable'설정을 체크한 경우 자동 완성을 지원하는 한 줄짜리 텍스트 편집란을 표시하며, 체크하지 않은 경우 콤보박스를 표시합니다.
- **UUID 생성기 (UUID Generator)**: UUID(Universally Unique Identifiers) 필드가 비어 있을 경우 읽기 전용 UUID 를 생성합니다.
- **Value Map**: A combo box with predefined items. The value is stored in the attribute, the description is shown in the combo box. You can define values manually or load them from a layer or a CSV file. The description is used as the representable value, meaning it will be shown in classification renderers.

> **참고**
>
> QGIS 는 GeoPackage 또는 ESRI 파일 지오데이터베이스처럼 사전 정의된 **코딩된 필드 도메인 (coded Field Domains)** 을 가진 일부 레이어들을 자동으로 인식해서 관련 필드들에 **값 지도** 위젯을 할당할 것입니다.

- **Value Relation**: Offers values from a related table in a combobox. You can select layer, key column and value column. The key column is the stored value in the attribute table, while the value column is the label shown in the combobox. Value column is also the representable value that will be shown in the classification renderers. You may also choose to order the list of displayed values by the *Key* column (the default sort order), the *Value* column, or a specific *Field* of the layer. Enable the  *Descending order* if you wish the values to be sorted in reverse order.

  Several options are available to change the standard behavior: allow NULL value, allow multiple selections and use of auto-completer. The forms will display either a drop-down list or a line edit field when completer checkbox is enabled.

  PostgreSQL, GeoPackage 또는 SpatiaLite 데이터베이스에 저장되어 있는 레이어가 값 관계 위젯을 사용하도록 환경 설정되었지만 필요한 레이어가 아직 프로젝트에 불러와져 있지 않은 경우, QGIS 가 동일 데이터베이스/연결에서 해당 레이어를 자동으로 검색할 것입니다.

### 12.1.10 결합 속성



*Joins* 탭에서, 현재 레이어의 피처와 또다른 불러온 벡터 레이어의 (또는 테이블의) 피처를 결합시킬 수 있습니다. 결합은 레이어들이 1 대 1 관계로 공유하는 속성을 기반으로 합니다. 결합에 대해 더 자세히 알고 싶다면두 레이어 간 피처 결합하기 를 읽어보세요.

### 12.1.11 보조 저장소 속성

스타일 작업 및 라벨 작업을 사용자 지정하는 표준적인 방법은데이터 정의 무시 설정 에서 설명하는대로 데이터 정의 속성을 사용하는 것입니다. 하지만, 기저 데이터가 읽기 전용인 경우 이 방법을 사용할 수 없을 수도 있습니다. 게다가 이런 데이터 정의 속성을 환경 설정하는 것이 너무 오래 걸리거나 또는 바람직하지 않을 수도 있고요! 예를 들어라벨 툴바 의 맵 도구들을 완전하게 사용하고 싶다면 사용자의 원본 데이터소스에 있는 20 개 이상의 (X 및 Y 위치, 회전 각도, 글꼴 스타일, 색상 등등) 필드들을 추가하고 환경 설정해주어야 합니다.

보조 저장소 메커니즘은 이런 제한 및 불편한 환경 설정을 해결할 수 있습니다. 보조 필드는 — 편집 가능한 결합 덕분에 — SQLite 데이터베이스에 이런 (라벨, 도표, 심볼 등의) 데이터 정의 속성들을 자동으로 관리하고 저장할 수 있는 우회적인 방법입니다. 즉 편집 가능 상태가 아닌 레이어의 속성들을 저장할 수 있게 해줍니다.

벡터 레이어 속성 대화창에 보조 저장소를 관리하기 위한 탭이 있습니다:

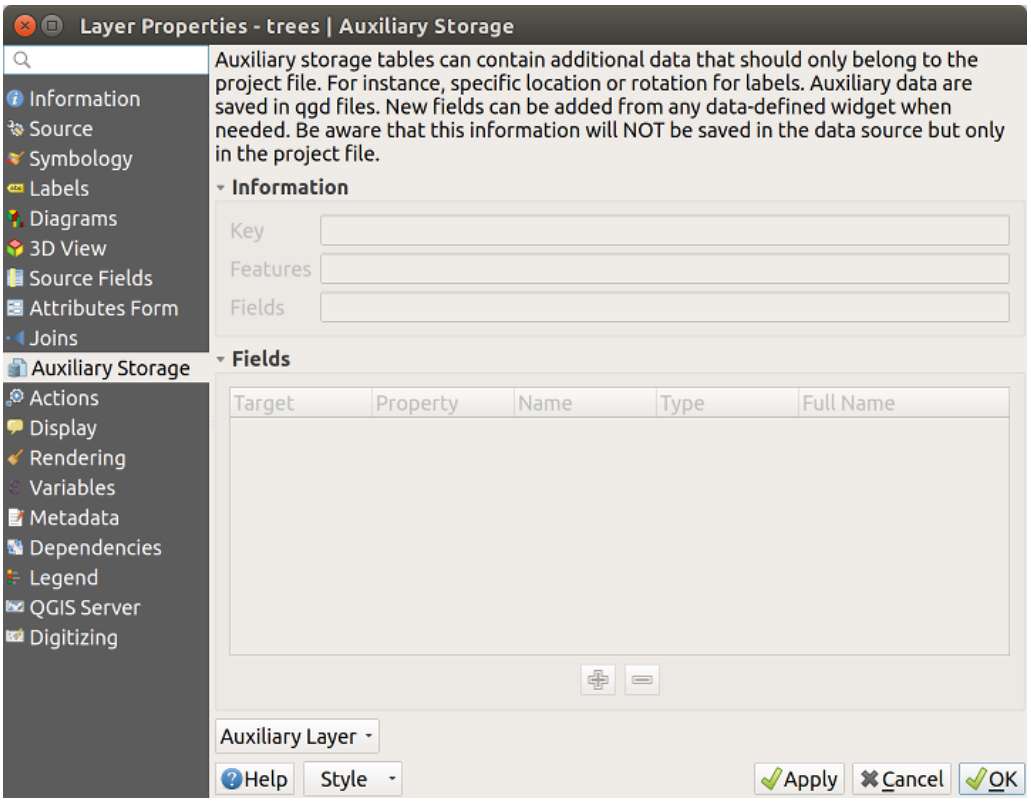



그림 12.48: 보조 저장소 탭

### 라벨 작업

편집 가능 상태가 아닌 데이터 정의 속성 덕분에 데이터소스를 사용자 지정할 수도 있다는 점을 고려하면, 라벨 작업을 활성화하면 언제라도라벨 툴바 에서 설명한 라벨 작업 맵 도구들을 사용할 수 있습니다.

사실, 보조 저장소 시스템은 SQLite 데이터베이스에 (보조 저장소 데이터베이스 참조) 이 속성들을 저장하기 위해 보조 레이어를 필요로 합니다. 라벨 작업 맵 도구가 활성화된 상태에서 맵을 처음으로 클릭할 때 보조 레이어 생성 과정이 실행됩니다. 그러면 (피처를 유일하게 식별하도록 보장하기 위해) 결합 작업에 사용할 기본 키를 선택할 수 있는 창이 표시됩니다:

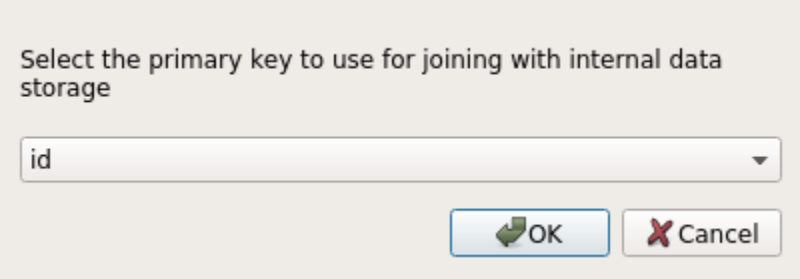



그림 12.49: 보조 레이어 생성 대화창

현재 데이터소스에서 보조 레이어의 환경을 설정하자마자, 탭에서 보조 레이어의 정보를 검색할 수 있습니다:

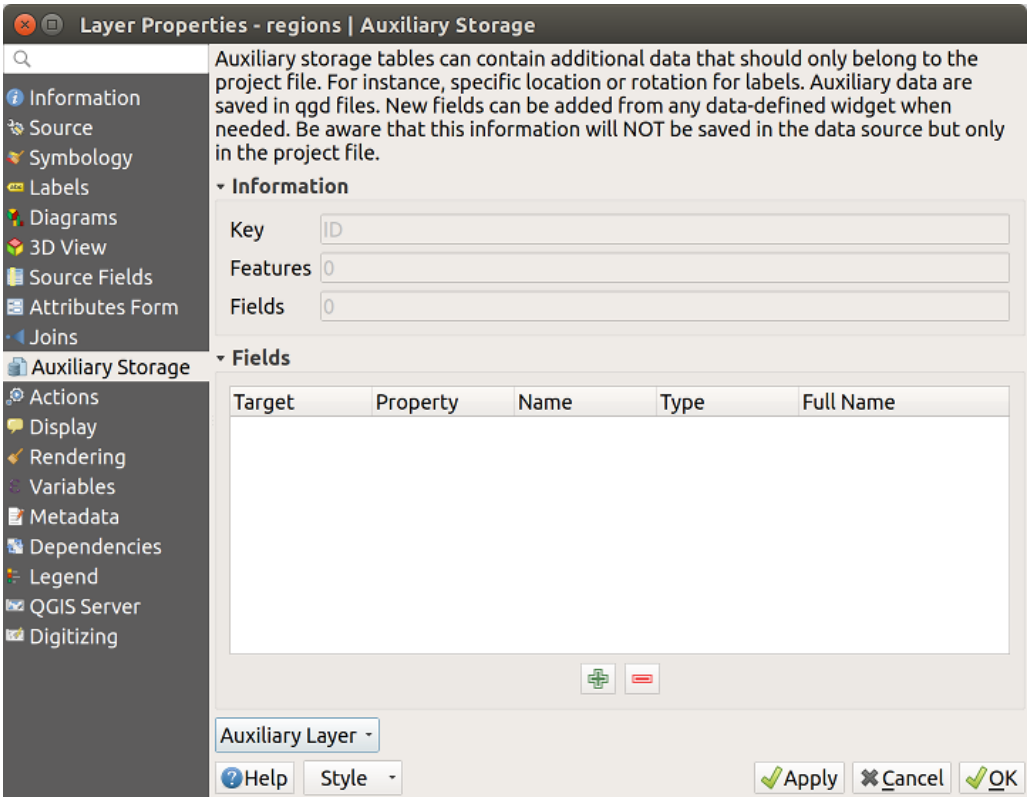



그림 12.50: 보조 레이어 키

이 보조 레이어는 다음과 같은 특성을 가집니다:

- 기본 키는 `ID` 이고,

- 보조 필드를 사용하는 피처는 0 개이며,
- 보조 필드도 0 개입니다.

이제 보조 레이어를 생성했으니, 레이어 라벨을 편집할 수 있습니다. Change Label 맵 도구를 활성화한 상태에서 라벨을 클릭하면, 크기, 색상 등의 스타일 작업 속성을 업데이트할 수 있습니다. 대응하는 데이터 정의 속성들을 다음과 같이 생성하고 가져올 수 있습니다:

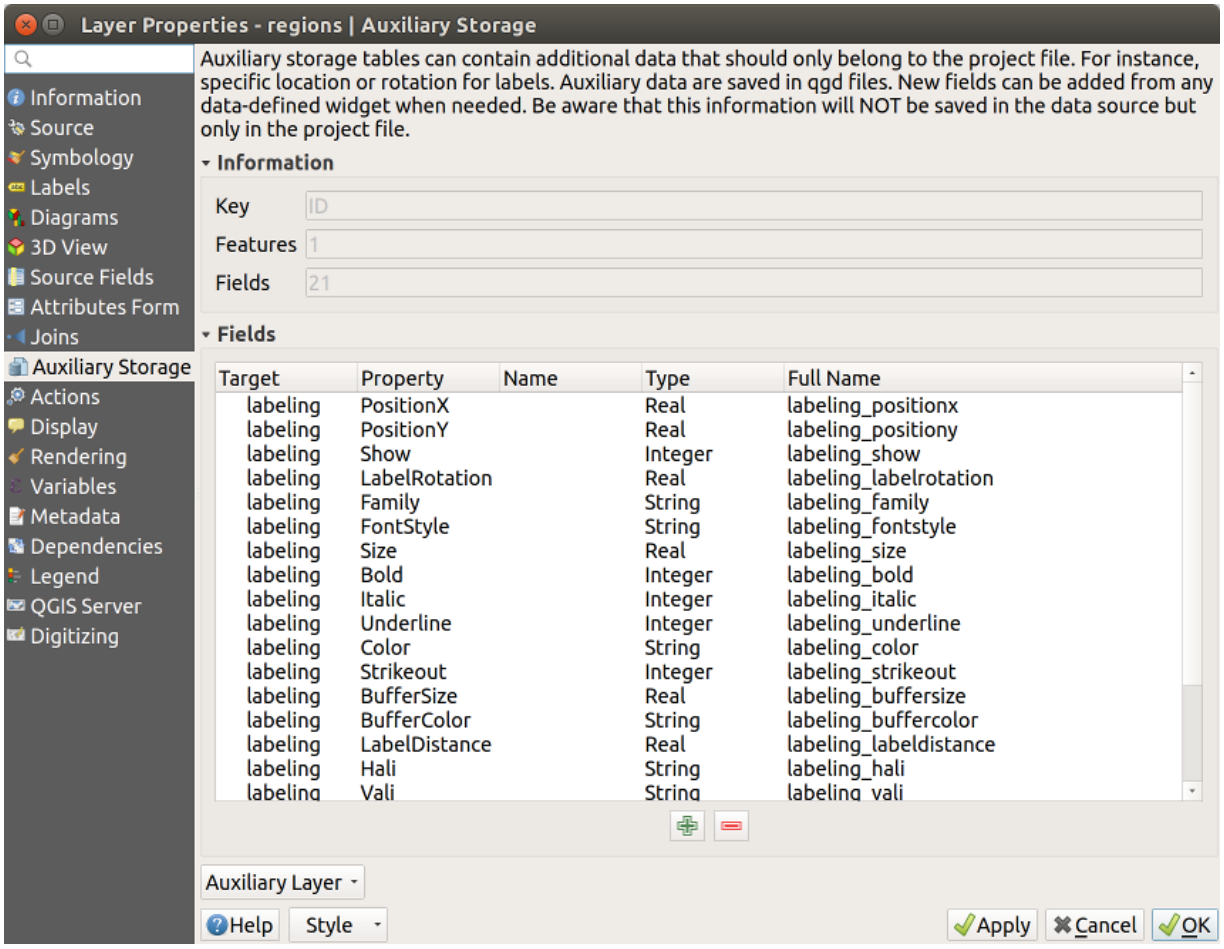



그림 12.51: 보조 필드

앞 그림에서 볼 수 있듯이, 라벨용으로 필드 21 개가 자동적으로 생성되고 환경 설정됐습니다. 예를 들면, 기저 SQLite 데이터베이스에서 `FontStyle` 보조 필드의 유형은 `String` 이며 이름은 `labeling_fontstyle` 입니다. 또 현재 보조 필드를 사용하고 있는 필드가 1 개있습니다.

*Labels* 속성 탭에 아이콘이 표시된 것을 눈치채셨나요? 데이터 정의 무시 옵션이 정확하게 설정되었다는 의미입니다:

그 외에도, Data-defined override 버튼으로 특정 속성을 위한 보조 필드를 생성할 수 있는 또다른 방법이 있습니다. *Store data in the project* 를 클릭하면, *Opacity* 필드를 위한 보조 필드를 자동적으로 생성합니다. 아직 보조 레이어를 생성하지 않은 상태에서 이 버튼을 클릭했다면, 보조 레이어 생성 대화창 창이 먼저 표시되어 결합 작업을 위한 기본 키를 선택해야 합니다.

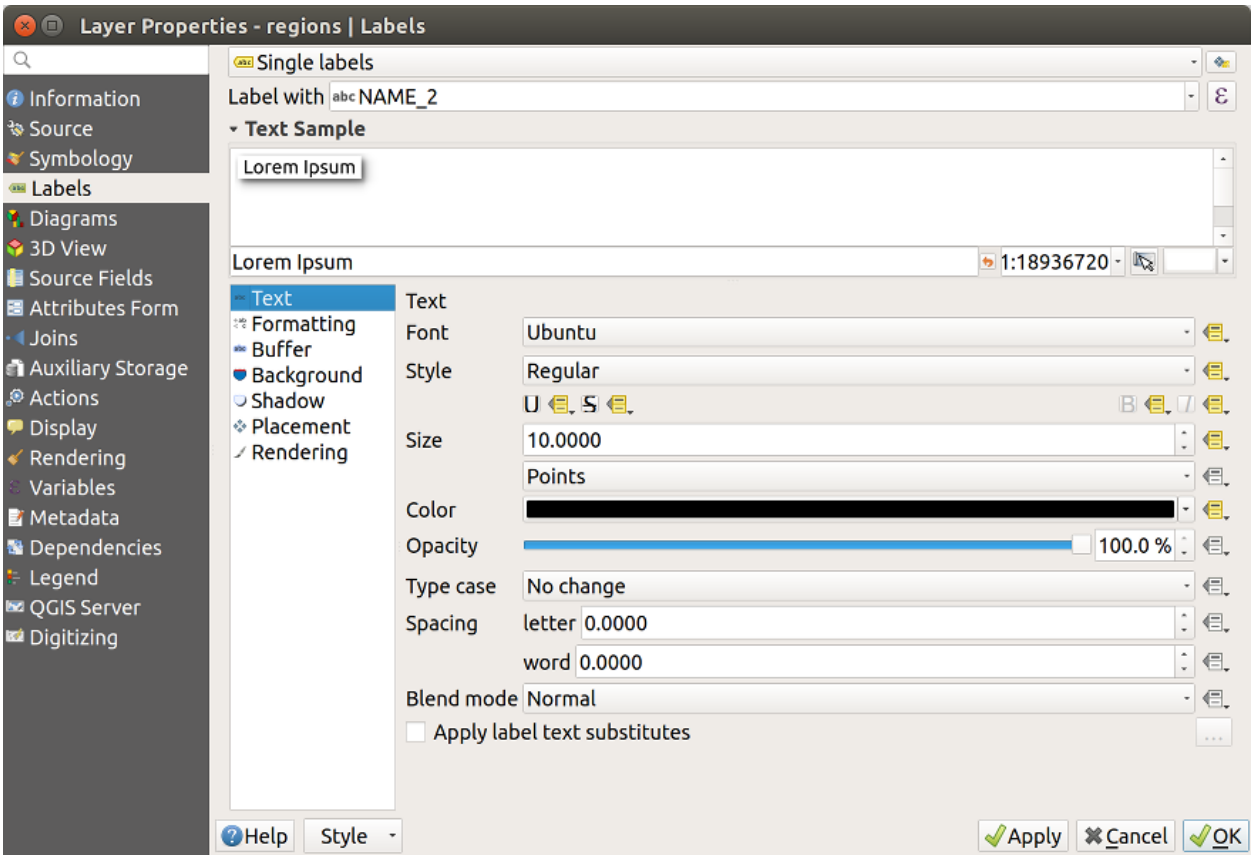



그림 12.52: 자동적으로 생성된 데이터 정의 속성들

### 심볼

앞에서 설명한 라벨 사용자 지정 작업을 위한 방법처럼, 보조 필드를 사용해서 심볼 및 도표의 스타일도 작업할 수 있습니다. 그러려면 먼저 Data-defined override 를 클릭한 다음 특정 속성을 위해 *Store data in the project* 를 선택하십시오. 예를 들어 *Fill color* 필드의 경우:

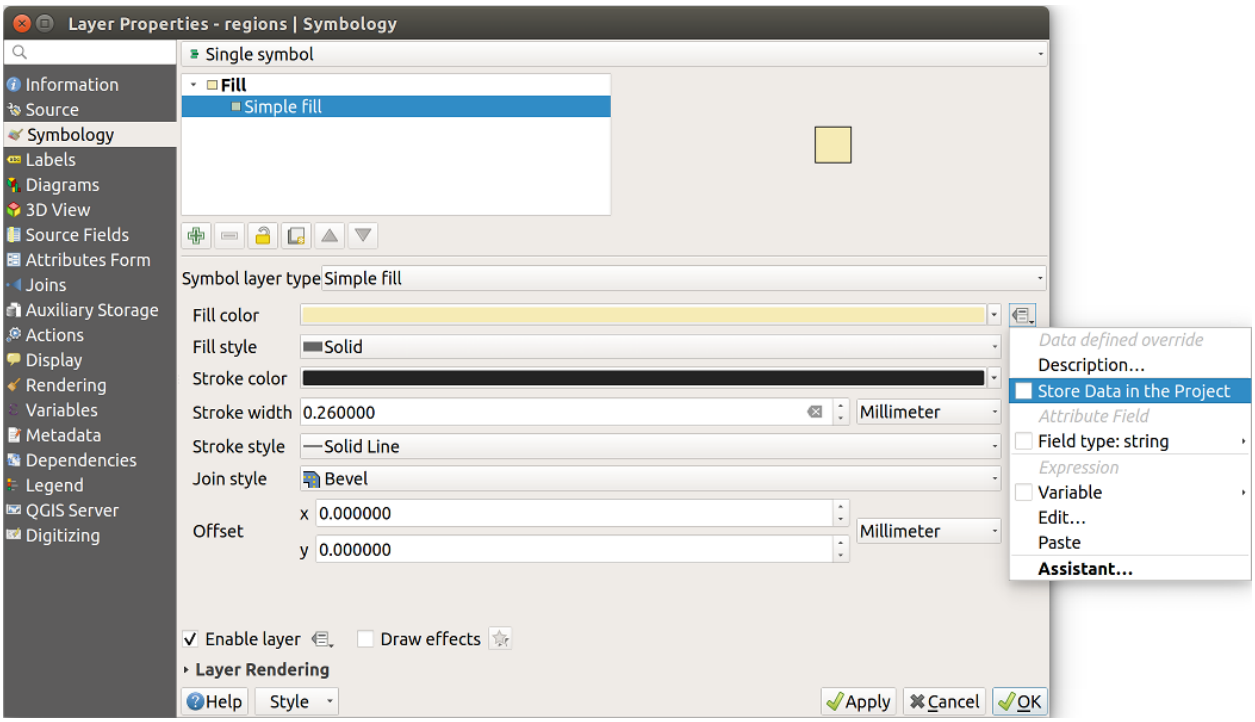



그림 12.53: 심볼을 위한 데이터 정의 속성 메뉴

각 심볼을 위한 서로 다른 (예를 들어 채우기 스타일, 채우기 색상, 획 색상 등등) 속성들이 있으므로, 속성을 표현하는 각 보조 필드에는 충돌을 피하기 위한 유일한 이름이 필요합니다. *Store data in the project* 를 선택하면 필드의 *Type* 을 표시하고 보조 필드용 유일 이름을 입력하도록 하는 창이 열립니다. 예를 들면, *Fill color* 보조 필드를 생성할 때 다음과 같은 창이 열립니다:

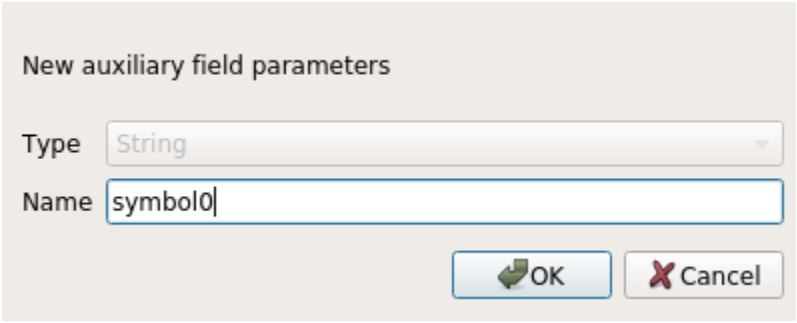



그림 12.54: 심볼을 위한 보조 필드의 이름

보조 필드를 생성하고 나면, 보조 저장소 탭에서 해당 필드를 검색할 수 있습니다:

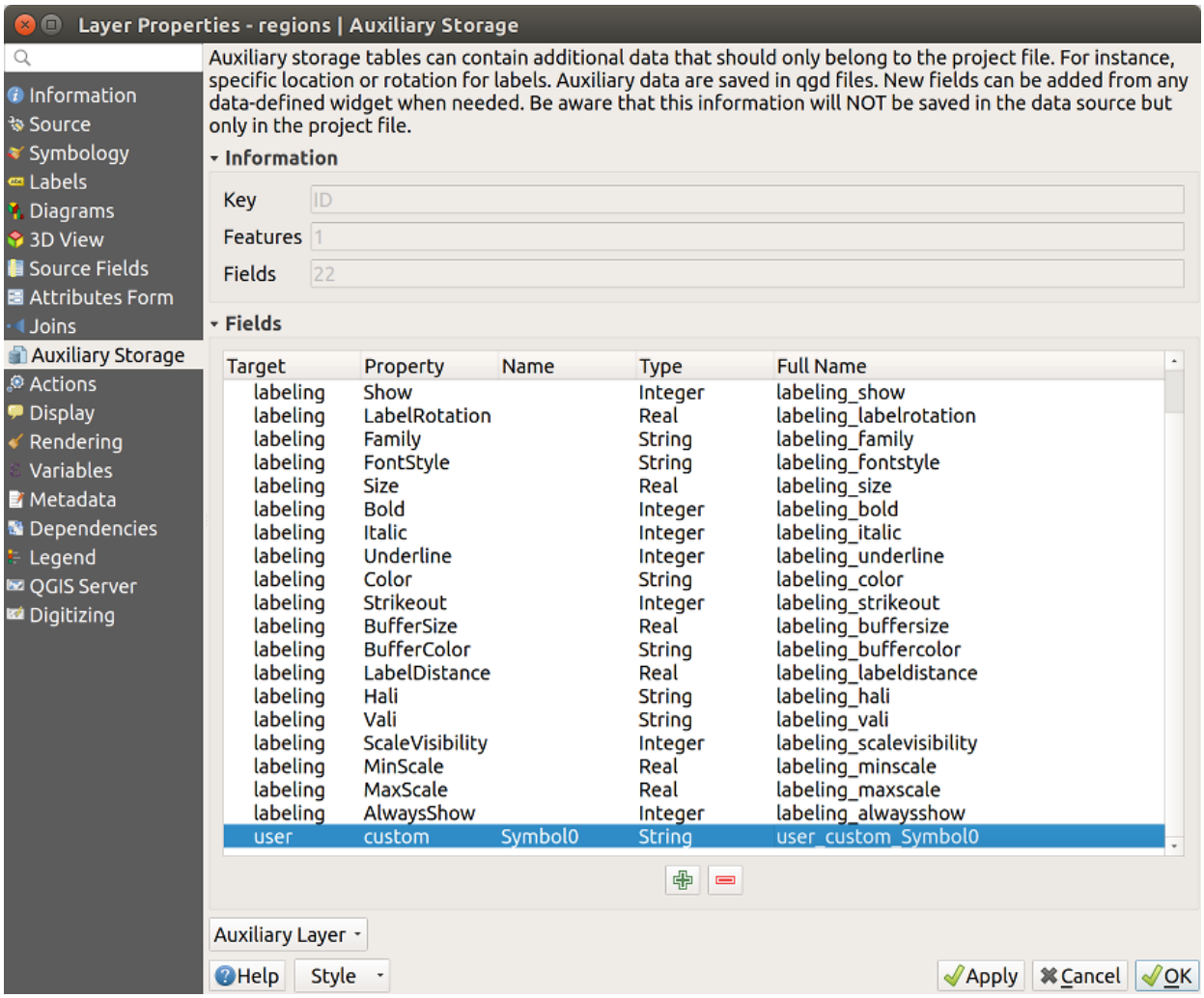



그림 12.55: 보조 필드 심볼

### 속성 테이블 및 위젯

속성 테이블 을 사용해서 보조 필드를 편집할 수 있습니다. 하지만, 모든 보조 필드가 처음부터 속성 테이블에 나타나지는 않습니다.

레이어의 심볼, 라벨, 모양, 또는 도표 속성을 표현하는 보조 필드들은 속성 테이블에 자동적으로 나타납니다. 그 예외는라벨 툴바 를 사용해서 수정할 수 있는 속성으로, 기본적으로 숨겨져 있습니다. `Color` 를 표현하는 보조 필드는 기본적으로 **색상** 위젯을 보유하도록 설정되는데, 그 외의 보조 필드는 기본적으로 **텍스트 편집** 위젯으로 설정됩니다.

라벨 툴바 를 사용해서 수정할 수 있는 속성을 표현하는 보조 필드는 기본적으로 속성 테이블에서 **숨겨져 (Hidden)** 있습니다. 이런 필드를 가시화하려면, 속성 양식 속성 탭 을 열고 보조 필드의 *Widget Type* 을 **Hidden** 에서 다른 관련 값으로 변경하십시오. 예를 들어, **auxiliary_storage_labeling_size** 를 **Text Edit** 으로 변경하거나 또는 **auxiliary_storage_labeling_color** 를 **Color** 위젯으로 변경하십시오. 이제 필드가 속성 테이블에서 가시화될 것입니다.

속성 테이블에 있는 보조 필드는 다음 그림처럼 나타날 것입니다:

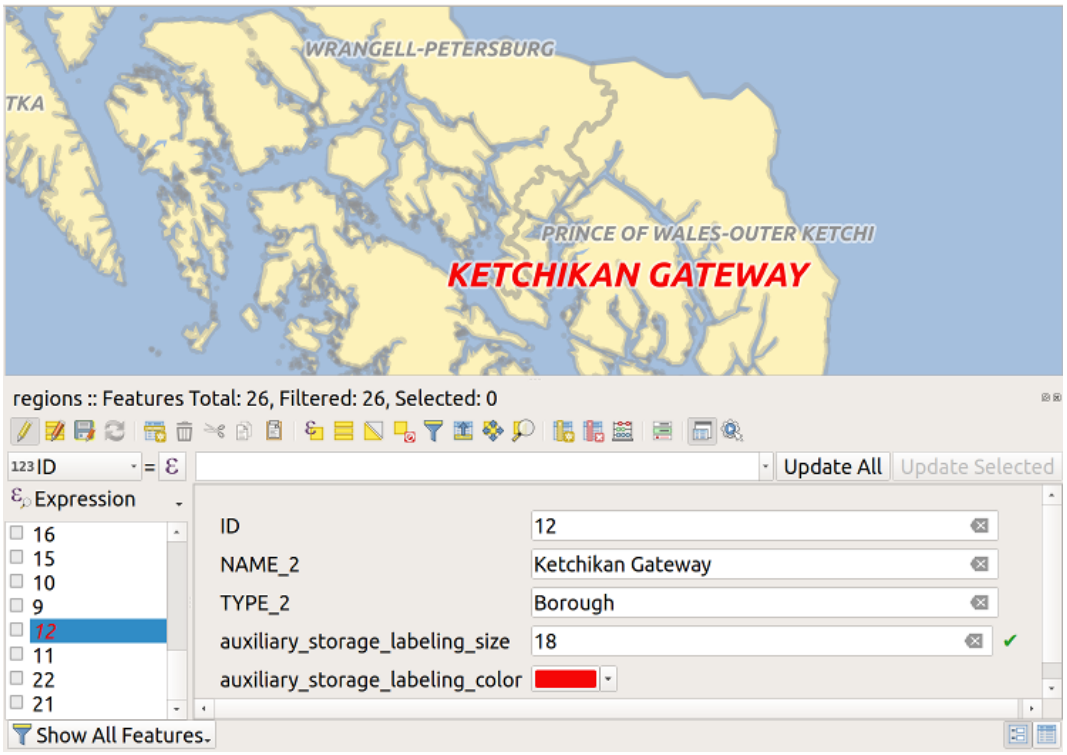



그림 12.56: 보조 필드를 보유한 양식

### 관리

*Auxiliary Layer* 메뉴에서 보조 필드를 관리할 수 있습니다:

이 경우 첫 항목 *Create* 가 비활성화돼 있습니다. 보조 레이어가 이미 생성됐기 때문입니다. 그러나 처음부터 작업하는 경우, 이 액션을 사용해서 보조 레이어를 생성할 수 있습니다. 라벨 작업 에서 설명했듯이, 기본 키가 필요할 것입니다.

*Clear* 액션은 모든 보조 필드를 유지하면서 그 내용들은 제거할 수 있습니다. 이렇게 하면 이 필드들을 사용하는 피처의 개수가 `0` 으로 떨어질 것입니다.

*Delete* 액션은 보조 레이어를 완전히 제거합니다. 다시 말해, 기저 SQLite 데이터베이스에서 대응하는 테이블을 삭제합니다. 사용자 지정한 속성들이 사라집니다.

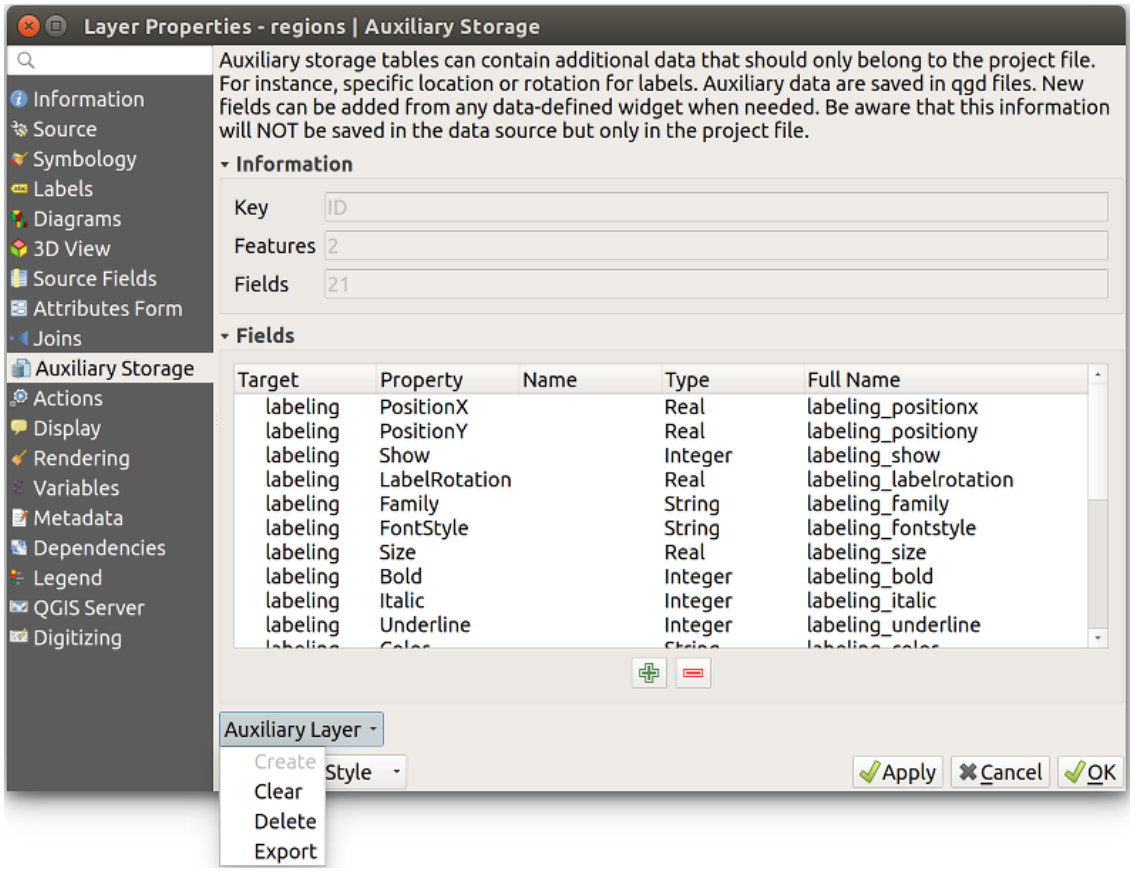



그림 12.57: 보조 레이어 관리

마지막으로 *Export* 액션은 보조 레이어를새 벡터 레이어 로 저장할 수 있습니다. 보조 저장소는 도형을 저장하지 않는다는 사실을 기억하십시오. 하지만 이 경우, 원본 데이터소스에서 도형들도 내보냅니다.

#### 보조 저장소 데이터베이스

사용자 프로젝트를 `.qgs` 포맷으로 저장하는 경우, 보조 저장소용으로 사용한 SQLite 데이터베이스도 동일한 위치에 저장되며 그 확장자는 `.qgd` 입니다.

사용자 편의를 위해, `.qgz` 압축 포맷을 대신 사용할 수도 있습니다. 이 경우 `.qgd` 및 `.qgs` 파일 둘 다 압축 파일에 들어갑니다.

### 12.1.12 액션 속성

*Actions* 탭은 객체의 속성을 기반으로 하는 액션을 수행할 수 있는 기능을 제공하고 있습니다. 예를 들면 객체의 속성에서 생성한 인자 (argument) 와 함께 프로그램을 실행하거나 또는 웹 보고 도구에 파라미터를 전달하는 등의 수많은 액션을 수행할 수 있습니다.

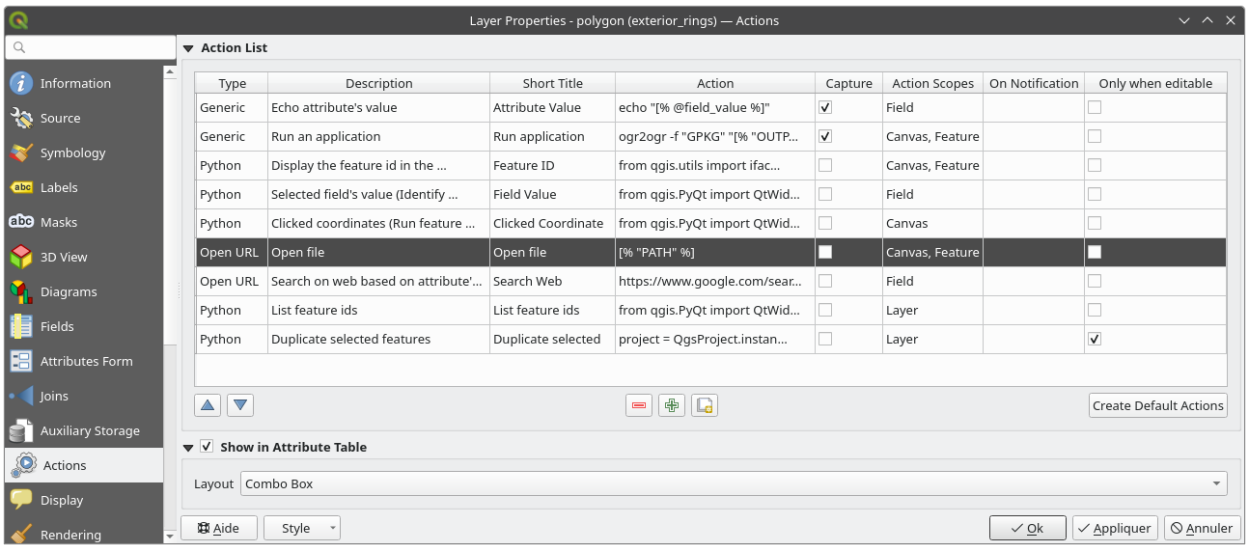



그림 12.58: 몇몇 예시 액션을 담고 있는 오버뷰 액션 대화창

외부 응용 프로그램을 자주 실행하거나 사용자 벡터 레이어에 있는 하나 이상의 값들을 기반으로 웹 페이지를 보려 하는 경우 액션이 유용합니다. 액션은 서로 다른 유형으로 나뉘며, 다음과 같이 사용할 수 있습니다:

- *Generic*, *macOS*, *Windows* 및 *Unix* 액션은 외부 프로세스를 실행합니다.
- *Python* 액션은 파이썬 표현식을 실행합니다.
- *Generic* 및 *Python* 액션은 어디에서나 가시화됩니다.
- *macOS*, *Windows* 및 *Unix* 액션은 각각 대응하는 플랫폼 상에서만 가시화됩니다. (예를 들어, 편집기를 여는 "편집"액션을 3 개 정의해도 사용자는 편집기를 실행하는 플랫폼의 전용"편집"액션만을 보고 실행할 수 있습니다.)
- *Open URL*: 지정된 URL 을 열기 위해 HTTP GET 요청을 이용합니다.
- *Submit URL (urlencoded or JSON)*: HTTP POST 요청을 이용한다는 점을 제외하면 *Open URL* 액션과 동일합니다. 본문이 무결한 JSON 인 경우"application/x-www-form-urlencoded"또는"application/json"을 사용해서 URL 에 데이터를 포스팅합니다.

다음은 액션 호출의 한 예시입니다:

```
http://localhost:8000?/[% url_encode(map('file', 'index.php')) %]
```

- *Submit URL (multipart)*: HTTP POST 요청을 이용한다는 점을 제외하면 *Open URL* 액션과 동일합니다. "multipart/form-data"를 이용해서 URL 에 데이터를 포스팅합니다.

이 대화창은 예시를 몇 개 포함하고 있습니다. *Create Default Actions* 버튼을 클릭하면 이 예시 액션들을 불러올 수 있습니다. 예시 액션을 편집하려면, 해당 행을 더블클릭하십시오. 예시 가운데 하나는 속성값을 기반으로 검색을 수행합니다. 다음 항에서 이 개념을 사용합니다.

*Show in Attribute Table* 옵션을 활성화하면 속성 테이블 대화창에 체크한 피처 범위 (feature-scoped) 액션들을 *Combo Box* 또는 *Separate Buttons* 가운데 하나로 표시할 수 있습니다. (열 환경설정하기 참조)

### 액션 정의하기

속성 액션을 정의하려면, 벡터 *Layer Properties* 대화창을 열고 *Actions* 탭을 선택하십시오. *Actions* 탭에서 

을 클릭하면 *Edit Action* 대화창이 열립니다.

*Type* 옵션으로 액션 유형을 선택하고 액션을 설명하는 이름을 부여하십시오. 액션 자체의 이름이 액션을 작동시켰을 때 실행될 응용 프로그램의 이름을 포함해야만 합니다. 하나 이상의 속성 필드를 응용 프로그램에 대한 인자로서 추가할 수 있습니다. 액션을 작동시켰을 때, 필드명 앞의 `%` 로 시작하는 어떤 문자 집합도 해당 필드의 값으로 대체될 것입니다. 특수 문자열 `%%` 는 식별 결과 또는 속성 테이블에서 선택한 필드의 값으로 대체될 것입니다. (액션 사용하기 항을 참조하세요.) 큰따옴표로 여러 텍스트를 묶으면 프로그램, 스크립트 또는 명령어에 대한 단일 인자로 만들 수 있습니다. 큰따옴표 앞에 역 슬래시가 있을 경우 이를 무시할 것입니다.

액션은 인자를 사용해서 단일 프로세스를 호출할 수 있기 때문에, (`&`, `&&`, `;`, `|` 같은) 불 연산자는 작동하지 않을 것입니다. 유닉스 같은 운영체제에서는 여러 개의 명령어들을 `bash -c` 를 통해 실행할 수 있습니다.

*Action Scopes* 는 어디에서 액션을 사용할 수 있어야 하는지를 정의할 수 있습니다. 다음 가운데 선택할 수 있습니다:

1. *Field*: 속성 테이블에 있는 셀, 피처 양식, 그리고 메인 툴바의 기본 액션 버튼을 오른쪽 클릭하면 액션을 사용할 수 있습니다.
2. *Feature*: 속성 테이블에 있는 셀을 오른쪽 클릭하면 액션을 사용할 수 있습니다.
3. *Canvas*: 툴바에 있는 메인 액션 버튼으로 액션을 사용할 수 있습니다.
4. *Form*: 드래그 & 드롭 모드를 이용해서 설계된 피처 양식에서만 액션을 사용할 수 있습니다.
5. *Layer*: 속성 테이블 툴바에 있는 액션 버튼으로 액션을 사용할 수 있습니다. 이 액션 유형은 단일 피처가 아니라 레이어 전체를 대상으로 한다는 점을 유의하십시오.

필드명이 다른 필드명의 하위 문자열인 경우 (예: `col1` 과 `col10`) 필드명 (과 `%` 문자) 을 꺾쇳괄호로 묶어서 (예: `[%col10]`) 그 사실을 나타내야 합니다. 이렇게 하면 필드명 `%col10` 을 뒤에 0 이 붙은 `%col1` 필드명과 혼동하는 일을 피할 수 있습니다. QGIS 가 `%` 문자열을 필드값으로 대체할 때 괄호를 제거할 것입니다. 대체된 필드값이 꺾쇠괄호로 묶여 있길 바란다면, `[[%col10]]` 처럼 괄호를 두 번 치십시오.

*Identify Features* 도구를 사용하면 *Identify Results* 대화창을 열 수 있습니다. 이 대화창에 레이어 유형에 관련된 정보를 담은 *(Derived)* 항목이 있습니다. 이 파생 필드의 이름을 `(Derived)` . 로 처리하면 다른 필드에 접근하는 것과 비슷한 방식으로 이 항목의 값에 접근할 수 있습니다. 예를 들어 포인트 레이어는 `X` 및 `Y` 필드를 보유하고 있는데, `%(Derived).X` 및 `%(Derived).Y` 로 이 필드들의 값을 사용할 수 있습니다. *Attribute Table* 대화창이 아니라 *Identify Results* 대화 상자에서만 이 파생 속성을 쓸 수 있습니다.

다음은 예시 액션 2 개입니다:

- `konqueror https://www.google.com/search?q=%nam`
- `konqueror https://www.google.com/search?q=%%`

첫 번째 예시는 캉커러 웹브라우저를 작동시키면서 열어야 할 URL 을 전달합니다. 이 URL 은 벡터 레이어에 있는 `nam` 필드의 값을 구글 검색합니다. 액션이 호출하는 응용 프로그램 또는 스크립트가 반드시 경로 안에 있어야만 한다는 사실을 기억하십시오. 그렇지 않을 경우 전체 경로를 입력해야만 합니다. 확실하게 하려면, 첫 번째 예시를 `/opt/kde3/bin/konqueror https://www.google.com/search?q=%nam` 으로 다시 작성하면 됩니다. 이렇게 하면 액션을 작동시켰을 때 캉커러 응용 프로그램이 확실하게 실행될 것입니다.

두 번째 예시는%% 기호를 사용하는데, 특정 필드에서 그 값을 불러오지 않는다는 뜻입니다. 액션을 작동시켰을 때, %% 기호는 식별 결과 또는 속성 테이블에서 선택한 필드의 값으로 대체될 것입니다.

### Duplicating Actions

QGIS allows you to duplicate existing actions. To duplicate an attribute action, open the vector *Layer Properties* dialog and click on the *Actions* tab. In the *Actions* tab, click the  Duplicate an action to open the *Duplicate Action* dialog. You must have selected at least one existing action in order to create a duplicate.

In the dialogue that appears, make any changes that are necessary. See 액션 정의하기 for further information. Once finished, press *OK* to create a duplicate of the action with any changes that you made. If you did not edit the description, or if you changed it to be identical to the description of any other existing action, "_1"will be added to the end of it.

#### 액션 사용하기

QGIS 는 사용자가 레이어 상에 활성화시킨 액션들을 실행시킬 수 있는 방법을 여러 개 제공하고 있습니다. 액션 설정에 따라, 다음과 같이 실행시킬 수 있습니다:

- *Attributes toolbar* 또는 *Attribute table* 대화창의  Run Feature Action 버튼의 드롭다운 메뉴를 통해
-  Identify Features 도구로 피처를 오른쪽 클릭했을 때 (자세한 내용은피처 식별 참조)
- *Identify Results* 패널의 *Actions* 부분에서
- *Attribute Table* 대화창에 있는 *Actions* 열의 항목으로

`%%` 표기를 사용하는 액션을 작동시키는 경우, *Identify Results* 대화창 또는 *Attribute Table* 대화창에서 응용 프로그램 또는 스크립트에 전달하려는 필드값을 오른쪽 클릭하십시오.

배시 (bash) 및 `echo` 명령어를 이용해 (따라서  또는 아마도  에서만 동작할 겁니다) 벡터 레이어에서 데이터를 추출해서 파일로 삽입하는 또다른 예시가 있습니다. 이 레이어는 수종명 `taxon_name`, 위도 `lat` 그리고 경도 `long` 필드를 가지고 있습니다. 서식지를 공간 선택 (spatial selection) 해서 (QGIS 맵 영역에서 노란색으로 표시된) 선택한 레코드의 필드값들을 텍스트 파일로 내보내려 합니다. 다음은 이 작업을 하기 위한 액션입니다:

```
bash -c "echo \"%taxon_name %lat %long\" >> /tmp/species_localities.txt"
```

서식지를 몇 개 선택해서 각각에 대해 액션을 실행한 다음, 산출 파일을 열면 다음과 비슷한 내용을 보게 될 겁니다:

```
Acacia mearnsii -34.0800000000 150.0800000000
Acacia mearnsii -34.9000000000 150.1200000000
Acacia mearnsii -35.2200000000 149.9300000000
Acacia mearnsii -32.2700000000 150.4100000000
```

`lakes` 레이어에 대한 구글 검색 액션을 생성하는 실습을 해보겠습니다. 먼저, 키워드 검색을 수행하기 위한 URL 을 알아야 합니다. 구글 사이트로 가서 간단한 검색을 한 다음, 사용자 브라우저의 주소창에서 URL 을 복사하면 쉽게 얻을 수 있습니다. 이렇게 QGIS 가 검색어인 구글 검색 URL 서식이 https://www.google.com/search?q=QGIS 라는 걸 알았습니다. 이때 `QGIS` 가 검색어입니다. 이 정보를 가지고 다음 단계로 넘어가겠습니다:

1. `lakes` 레이어를 불러왔는지 확인합니다.

2. 범례에 있는 레이어를 더블클릭하거나, 또는 오른쪽 클릭한 다음 컨텍스트 메뉴에서 *Properties* 를 선택해서 *Layer Properties* 대화창을 엽니다.
3. *Actions* 탭을 클릭합니다.
4.  Add a new action 아이콘을 클릭합니다.
5. *Open URL* 액션 유형을 선택하고,
6. 액션명을, 예를 들어 `Google Search` 를 입력합니다.
7. *Short Name*, 또는 *Icon* 도 따로 추가할 수 있습니다.
8. 액션의 *Action Scope* 를 선택하십시오. 더 자세한 내용은액션 정의하기 를 참조하세요. 이 예시의 경우 기본 설정을 유지합니다.
9. 액션에 구글 검색 용 URL `https://www.google.com/search?q=` 를 추가합니다. 검색어는 포함시키지 않습니다.
10. *Action* 란에 있는 텍스트가 이제 다음처럼 보여야 합니다:

    ```
    https://www.google.com/search?q=
    ```

11. `lakes` 레이어의 필드명을 담고 있는 드롭다운 박스를 클릭합니다. *Insert* 버튼 바로 왼쪽에 있습니다.
12. 드롭다운 박스에서 *NAMES* 를 선택한 다음 *Insert* 를 클릭합니다.
13. 이제 사용자 액션 텍스트가 다음과 같이 보일 겁니다:

    ```
    https://www.google.com/search?q=[%NAMES%]
    ```

14. 액션 생성을 완료하고 추가하려면, *OK* 버튼을 클릭합니다.

이제 액션을 완성했고, 사용할 준비가 되었습니다.

*Layer Properties* 대화창을 닫은 다음 관심 지역으로 확대하십시오. `lakes` 레이어가 활성화돼 있는지 확인한 다음 호수를 식별하십시오. 식별 결과 창에 생성한 액션이 다음과 같이 추가된 것을 확인할 수 있습니다:

이 액션을 클릭하면, 기본 브라우저를 구동시켜 URL https://www.google.com/search?q=Tustumena 로 이동합니다. 이 액션에 더 많은 속성 필드를 추가할 수도 있습니다. 액션 텍스트 마지막에 + 를 추가하고, 또다른 필드를 선택한 다음 *Insert Field* 버튼을 클릭하면 됩니다. 그런데 이 예제에는 검색할 의미가 있는 다른 필드가 없네요.

레이어 하나에 액션을 여러 개 정의할 수 있고, 각각의 액션이 *Identify Results* 대화창에 표시될 겁니다.

속성 테이블에서 행을 하나 선택하고 오른쪽 클릭한 다음, 컨텍스트 메뉴에서 액션을 선택해서 액션을 작동시킬 수도 있습니다.

액션의 쓸모는 무궁무진합니다. 예를 들면, 이미지 또는 사진의 위치를 파일명과 함께 담고 있는 포인트 레이어가 있을 경우, 이미지를 표시할 뷰어를 실행시키는 액션을 생성할 수 있습니다. 또는 구글 검색 예제에서와 동일한 방법으로 설정해서 속성 필드 또는 필드 결합에 대한 웹 기반 보고를 수행하는 액션을 사용할 수도 있습니다.

훨씬 복잡한 예제도 만들어볼 수 있습니다. 예를 들어, **파이썬** 액션을 이용해서 말이죠.

외부 응용 프로그램으로 파일을 여는 액션을 생성하는 경우 보통 절대 경로를 사용합니다만, 가끔 상대 경로를 쓰기도 하죠. 후자의 경우 외부 프로그램 실행 파일의 위치가 경로의 기준이 됩니다. 하지만 선택한 (셰이프파일 또는 SpatiaLite 같은 파일 기반) 레이어를 기준으로 하는 상대 경로를 사용해야 하는 경우라면 어떨까요? 다음 코드로 이 문제를 해결할 수 있습니다:

```
command = "firefox"
imagerelpath = "images_test/test_image.jpg"
layer = qgis.utils.iface.activeLayer()
import os.path
layerpath = layer.source() if layer.providerType() == 'ogr'
```

(다음 페이지에 계속)

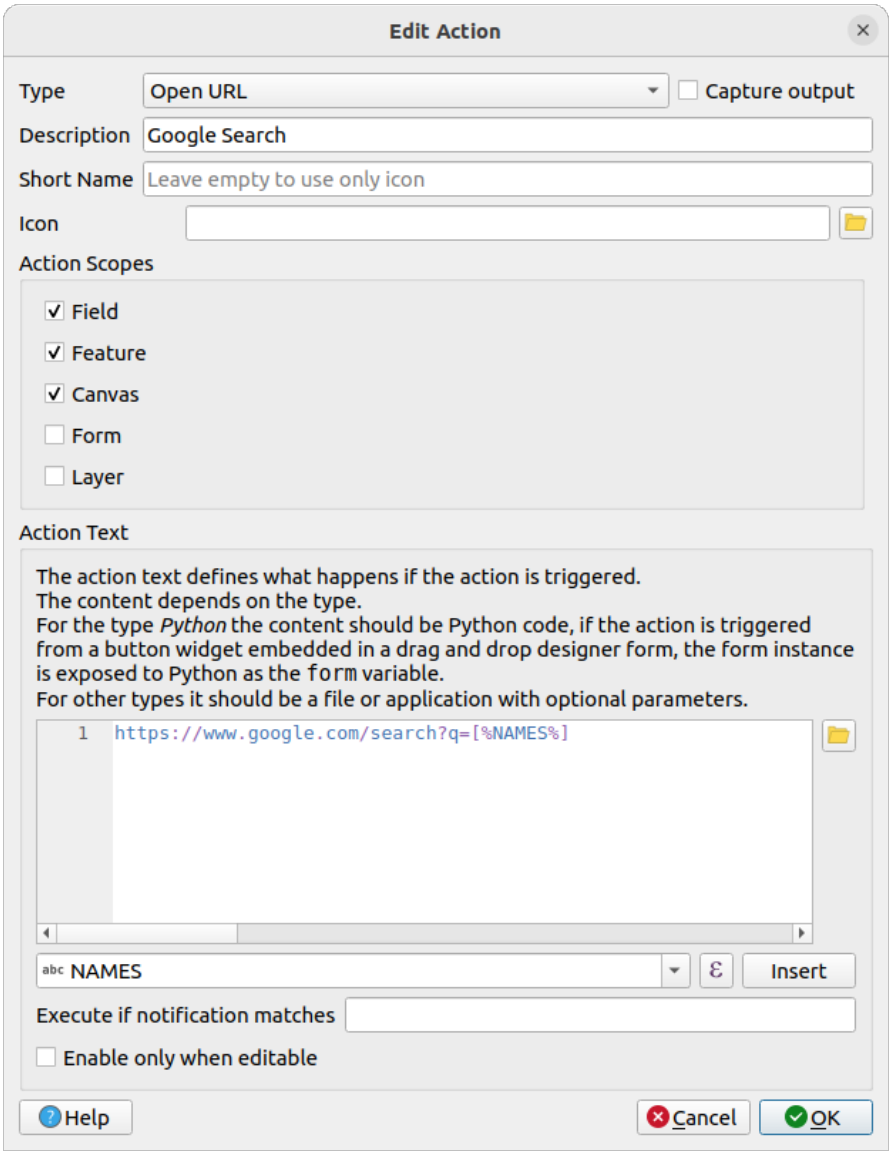



그림 12.59: 예시 액션 환경을 설정한 액션 편집 대화창

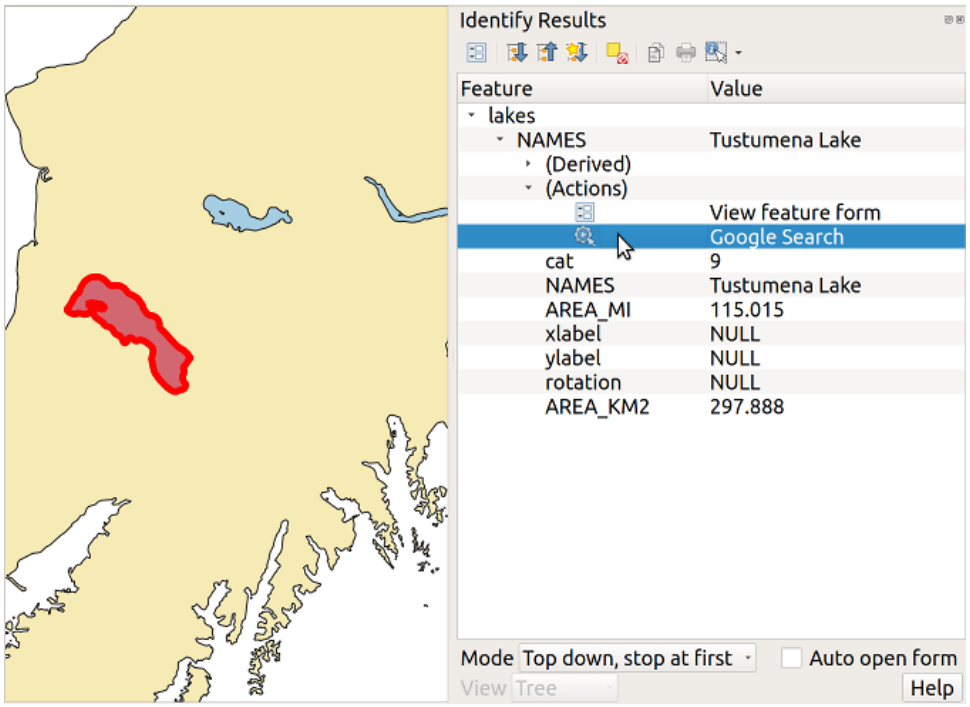



그림 12.60: 객체 선택 및 액션 선택

(이전 페이지에서 계속)

```
  else (qgis.core.QgsDataSourceURI(layer.source()).database()
    if layer.providerType() == 'spatialite' else None)
path = os.path.dirname(str(layerpath))
image = os.path.join(path,imagerelpath)
import subprocess
subprocess.Popen( [command, image ] )
```

이 액션이 파이썬 유형 가운데 하나이며 *command* 및 *imagerelpath* 변수를 사용자 필요에 맞춰 변경해야 한다는 점만 기억하면 됩니다.

그런데 상대 경로의 기준이 (저장한) 프로젝트 파일 위치여야 하는 경우는 또 어떨까요? 이 경우 파이썬 액션의 코드는 다음처럼 변합니다:

```
command = "firefox"
imagerelpath = "images_test/test_image.jpg"
projectpath = qgis.core.QgsProject.instance().fileName()
import os.path
path = os.path.dirname(str(projectpath)) if projectpath != '' else None
image = os.path.join(path, imagerelpath)
import subprocess
subprocess.Popen( [command, image ] )
```

파이썬 액션의 또다른 예시는 프로젝트에 새 레이어를 추가할 수 있게 해주는 액션입니다. 예를 들어, 다음 예시 코드들은 프로젝트에 각각 벡터와 래스터를 추가할 것입니다. 프로젝트에 추가되는 파일명 및 레이어에 할당되는 이름은 데이터를 기반으로 합니다. (*filename* 및 *layername* 은 액션이 작동한 벡터의 속성 테이블에 있는 열들의 이름입니다.)

```
qgis.utils.iface.addVectorLayer('/yourpath/[% "filename" %].shp',
  '[% "layername" %]', 'ogr')
```

래스터를 (이 예제에서는 TIF 이미지를) 추가하려면 다음 코드를 사용합니다:

```
qgis.utils.iface.addRasterLayer('/yourpath/[% "filename" %].tif',
  '[% "layername" %]')
```

### 12.1.13 표시 속성

*Display* 탭에서 피처 식별에 사용되는 필드의 환경을 설정할 수 있습니다:

- *Display name*: 필드 또는표현식 을 기반으로 하는 이름을 표시합니다. 레이어의 어떤 필드도 <name> 요소를 가지고 있지 않을 경우, 기본적으로 이 이름은 레이어의 첫 번째 필드로 설정됩니다.
  - 식별 도구 의 결과에서 피처 정보 최상단에 나타나는 라벨
  - 모든 레이어에서 피처를 찾을 때위치 탐색란 에서 쓰이는 필드
  - 속성 테이블양식 뷰 의 피처 식별자
  - 맵 또는 조판을 지리공간 PDF(Geospatial PDF) 같은 레이어화된 산출 포맷으로 내보내는 경우의 피처 식별자
  - 맵 도움말 (map tip) 정보, 예를 들어 Show Map Tips 아이콘을 누른 상태에서 활성화된 레이어의 피처 위에 마우스를 가져갔을 때 맵 캔버스에 표시되는 메시지. *Enable Map Tips* 옵션이 활성화되어 있고 어떤 *HTML Map Tip* 도 설정되어 있지 않을 때 활용할 수 있습니다.
- *Enable Map Tips*: 레이어에 맵 도움말을 표시할지 말지를 제어합니다.
- The *HTML Map Tip* provides a complex and full HTML text editor for map tips, mixing QGIS expressions and html styles and tags (multiline, fonts, images, hyperlink, tables, ⋯). You can check the result of your code sample in the *Preview* frame (also convenient for previewing the *Display name* output). Additionally, you can select and edit existing expressions using the *Insert/Edit Expression* button.

> **참고**
>
> Understanding the *Insert/Edit Expression* button behavior
>
> If you select some text within an expression (between "[%"and "%]"), or if no text is selected but the cursor is inside an expression, the whole expression will be automatically selected for editing. If the cursor or a selected text is outside an expression, the dialog opens with the selection.

맵 도움말을 표시하려면:

1. *View* → *Show Map Tips* 메뉴를 선택하거나 *Attributes Toolbar* 의 Show Map Tips 아이콘을 클릭하십시오.
2. 사용자가 대상으로 삼은 레이어가 활성화되어 있는지 그리고 *Enable Map Tips* 옵션이 체크되어 있는지 확인하십시오.
3. 마우스 커서를 피처 위로 가져가면, 해당 피처에 대응하는 정보가 표시될 것입니다.

맵 도움말은 크로스 레이어 (cross-layer) 기능입니다. 다시 말해 한번 활성화시키면 다시 끌 때까지 계속 프로젝트에서 맵 도움말이 활성화된 레이어들에 모두 적용된다는 뜻입니다.

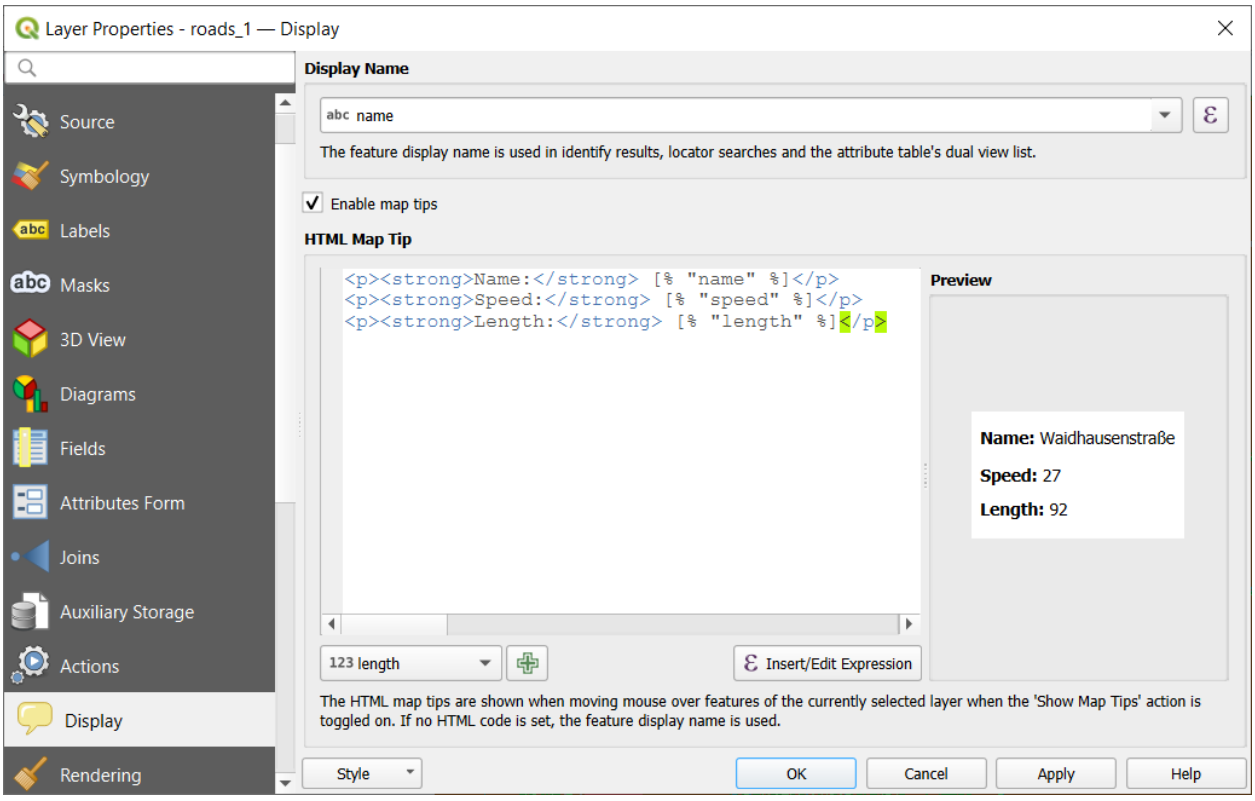



그림 12.61: 맵 도움말 용 HTML 코드




그림 12.62: HTML 코드로 생성된 맵 도움말

### 12.1.14 렌더링 속성

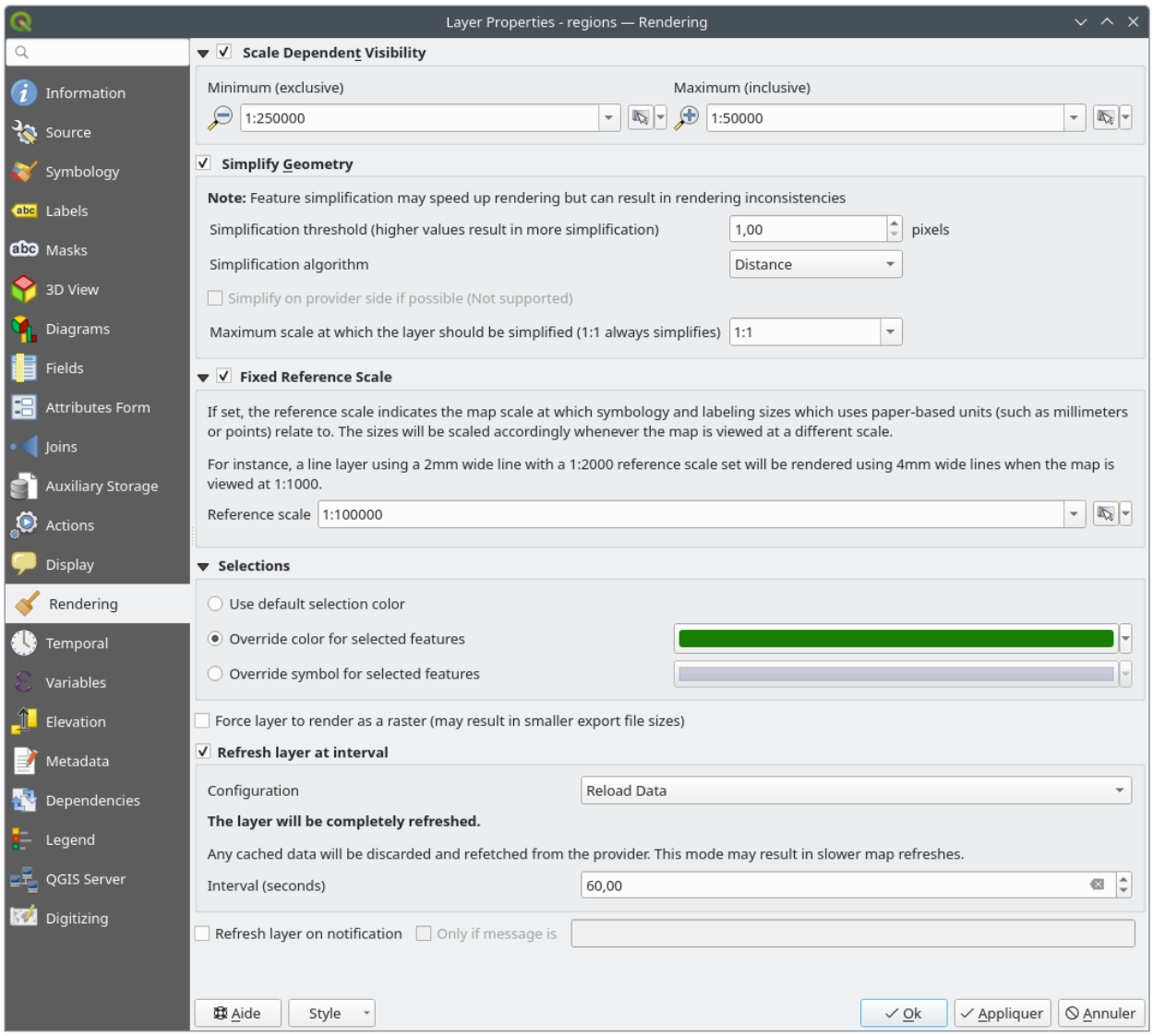



그림 12.63: 레이어 렌더링 속성 대화창

*Rendering* 탭에서 다음 설정을 할 수 있습니다:

- *Scale dependent visibility* 에서는 *Maximum (inclusive)* 및 *Minimum (exclusive)* 축척을 설정해서 피처가 보이게 될 축척 범위를 정의할 수 있습니다. 이 범위를 벗어나면, 피처를 숨깁니다. Set to current canvas scale 버튼을 클릭하면 현재 맵 캔버스의 축척을 가시성 범위의 한계값으로 설정할 수 있습니다. 자세한 내용은가시성 축척 선택기 을 참조하세요.

> **참고**
>
> *Layers* 패널에서 레이어에 대한 축척 의존 가시성을 활성화시킬 수도 있습니다. 레이어를 오른쪽 클릭한 다음 컨텍스트 메뉴에서 *Set Layer Scale Visibility* 를 선택하십시오.

- QGIS 는 실시간 피처 단순화를 지원합니다. 이 기능은 소축척에서 여러 복잡 피처를 그릴 때 렌더링 시간을 단축시켜 줍니다. 레이어 설정에서 ☑ *Simplify geometry* 옵션으로 이 기능을 활성화하거나 비활성화할 수 있습니다. 새로 추가하는 레이어에 대해 기본적으로 단순화를 활성화시키는 전체 수준 설정도 있습니다. (자세한 정보는전체 수준 단순화 를 참조하세요.)

  > **참고**
  >
  > 피처 단순화는 어떤 경우 사용자의 렌더링된 산출물에 오류를 남길 수도 있습니다. 폴리곤들 사이의 조각이거나, 오프셋 기반 심볼 레이어를 이용한 경우 부정확한 렌더링의 결과일 수도 있습니다.

- ☐ *Fixed reference scale* 을 설정하는 경우, 참조 축척은 (밀리미터 또는 포인트 같은) 용지 기반 단위를 사용하는 심볼 및 라벨 크기와 관련된 맵 축척을 나타냅니다. 맵의 축척을 변경할 때마다 그에 맞춰 크기를 조정할 것입니다.

  예를 들어 *Reference scale* 을 2 밀리미터 너비의 라인을 사용하는 1:2,000 으로 설정한 라인 레이어는 맵의 축척이 1:1,000 일 때 4 밀리미터 너비의 라인을 사용해서 렌더링될 것입니다.

- *Selections* 부분에서는 특정 레이어에 특정 색상 또는 심볼을 기본값으로 (*Project properties* → *General* → *Selection color*) 사용해야 할지 여부를 제어할 수 있습니다. 특정 심볼을 가진 선택 피처들의 가시성을 향상시키는 데 유용한 방법입니다:

  - *Use default selection color*: 선택 집합에 기본 색상을 적용합니다.
  - *Override color for selected features*: 예를 들어 레이어가 노란색을 기본 색상으로 사용하는 경우 표준 노란색 선택 집합을 보이지 않게 합니다.
  - *Override symbol for selected features*: 예를 들어 라인 레이어가 가느다란 심볼을 사용하는데 라인에 적용된 색상이 라인을 잘 보이지 않게 만드는 경우, 이 심볼을 무시하고 더 두꺼운 라인으로 만들면 도움이 될 수 있습니다. 또, 레이어가 래스터 심볼 또는 색상표를 가진 그레이디언트 채우기 (fill)/라인/부풀리기 (shapeburst) 심볼을 사용하는 경우 기본 선택 집합 색상은 전혀 적용되지 않습니다. 레이어에서 선택된 피처에 사용할 더 단순한 특정 심볼을 설정할 수 있으면 도움이 될 것입니다.

- 극도로 세밀한 레이어 (예: 수많은 노드를 지닌 폴리곤 레이어) 를 렌더링하면, 내보낸 파일에 수많은 노드가 전부 포함돼야 하기 때문에 조판기가 내보낸 PDF/SVG 포맷 파일의 용량이 터무니없이 커질 수도 있습니다. 또다른 프로그램에서 해당 파일을 열거나 편집할 때 아주 느려질 수도 있습니다.

  ☑ *Force layer to render as raster* 를 활성화하면 이런 레이어를 강제로 래스터화해서 내보내기 파일이 해당 레이어가 담고 있는 모든 노드를 포함할 필요가 없도록 합니다. 따라서 렌더링 속도도 향상됩니다.

  조판기에서 래스터로 내보내도록 강제해도 동일한 결과를 얻을 수 있지만, 이 방법은 모든 레이어를 래스터화하기 때문에 마지막 수단으로 남겨두어야 합니다. 아니면레이어 내보내기 설정 에 있는 도형 단순화 기능을 이용할 수도 있습니다.

- ☐ *Refresh layer at interval*: 이 옵션은 레이어를 얼마나 정기적으로 새로고침할 수 있는지 그리고 새로고침할지 여부를 제어합니다. 사용할 수 있는 환경설정 *Configuration* 옵션들은 다음과 같습니다:

  - *Reload data*: 레이어를 완전히 새로고침할 것입니다. 캐시되어 있던 데이터를 모두 삭제하고 제공자로부터 다시 받아올 것입니다. 이 모드는 맵 새로고침 속도를 저하할 수도 있습니다.
  - *Redraw layer only*: 이 모드는 애니메이션에 대해 또는 레이어의 스타일이 정기적인 간격으로 업데이트될 경우 유용합니다. 하나 이상의 레이어에 자동 업데이트 간격을 설정한 경우 여러번 새로고침하지 않도록 캔버스 업데이트를 연기합니다.

  연속되는 새로고침 사이의 간격을 초 단위로 *Interval (seconds)* 설정할 수도 있습니다.

- 데이터 제공자 (예: PostgreSQL) 에 따라, QGIS 외부에서 데이터소스에 변경 사항이 생겼을 경우 QGIS 에 알림을 전송할 수 있습니다. ☑ *Refresh layer on notification* 옵션을 사용해서 업데이트를 촉발하십시오. 특정 메시지에만 레이어를 새로고침하도록 ☑ *Only if message is* 텍스트란에 메시지를 설정할 수도 있습니다.

### 12.1.15 시계열 속성

*Temporal* 탭은 레이어의 렌더링을 시간에 따라 제어할 수 있는 옵션들을 제공합니다. 이런 동적 렌더링에는 맵 캔버스 전체에 적용되는시계열 탐색 을 활성화해야 할 필요가 있습니다.

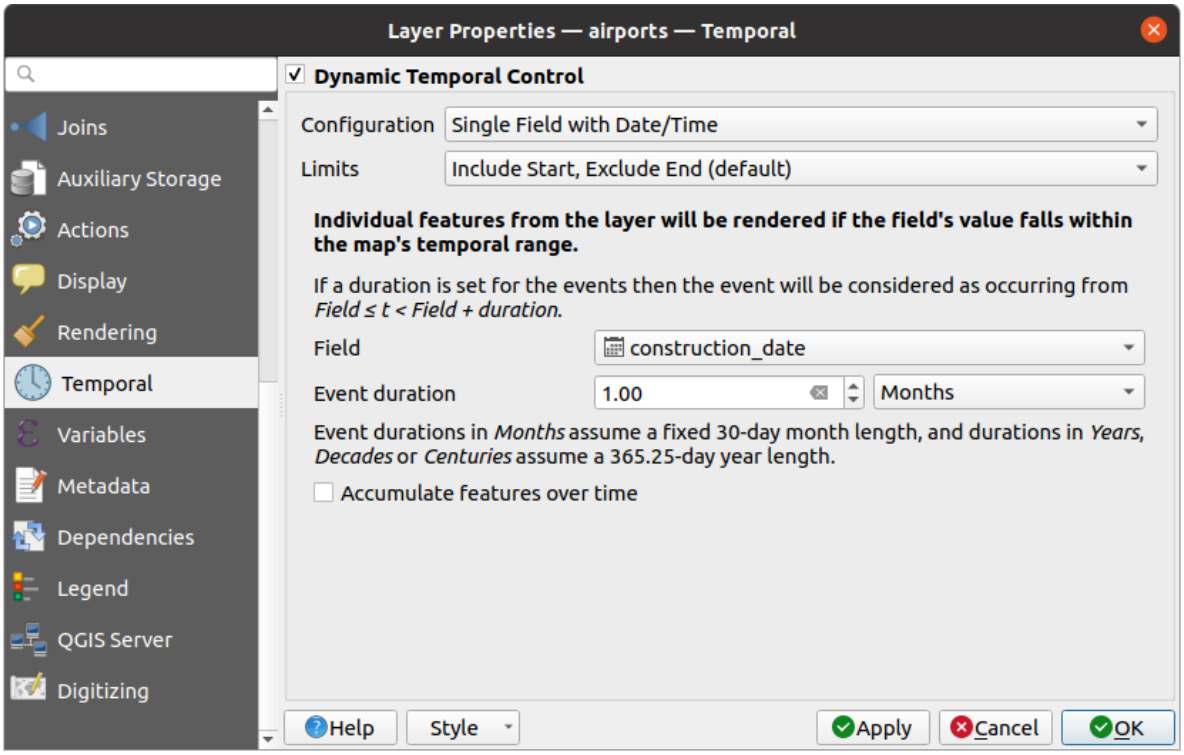



그림 12.64: 벡터 레이어 시계열 속성 대화창

벡터 레이어의 시계열 렌더링을 환경 설정하려면 *Dynamic Temporal Control* 옵션을 체크해야 합니다. 사용자 데이터셋의 구조에 따라, 다음 *Configuration* 옵션들 가운데 하나를 사용해야 할 수도 있습니다:

- *Fixed time range*: 맵 캔버스의 시계열 프레임이 지정한 *Start date* 와 *End date* 범위와 겹치는 경우 모든 피처를 렌더링합니다.
- *Single field with date/time*: 피처의 *Field* 값이 맵 캔버스의 시계열 프레임 안에 들어오는 경우 피처를 렌더링합니다. *Event duration* 을 설정할 수 있습니다. *Accumulate features over time* 옵션을 체크하면 맵의 시계열 범위 이전 또는 시계열 범위 안에 나타나는 모든 피처를 계속 렌더링할 것입니다. 이때 이벤트 기간 (event duration) 은 무시됩니다.
- *Separate fields for start and end date/time*: 피처의 *Start field* 와 *End field* 로 지정된 범위가 맵 캔버스의 시계열 프레임과 겹치는 경우 피처를 렌더링합니다.
- *Separate fields for start and event duration*: 피처의 *Start field* 와 *Event duration field* 로 정의된 범위가 맵 캔버스의 시계열 프레임과 겹치는 경우 피처를 렌더링합니다.
- *Start and end date/time from expressions*: 필드의 *Start expression* 과 *End expression* 으로 지정된 시간 범위가 맵 캔버스의 시계열 프레임과 겹치는 경우 피처를 렌더링합니다.
- *Redraw layer only*: 각각의 새로운 애니메이션 프레임에서 레이어를 새로 렌더링하지만 피처에 어떤 시간 기반 필터링도 적용하지 않습니다. 레이어가 렌더링 작업자 설정에 시간 기반 표현식을 사용하는 경우 (예: 데이터 정의 심볼) 유용합니다.

*Limits*: 피처의 시간 범위에 제한을 설정할 수도 있습니다:

- *Include start, exclude end*: 시작 시간을 포함하고, 중단 시간은 제외합니다.
- *Include start, include end*: 시작 시간과 중단 시간을 모두 포함합니다.

### 12.1.16 변수 속성

*Variables* 탭은 레이어 수준에서 쓸 수 있는 모든 변수들의 목록을 표시합니다. (전체 및 프로젝트 수준의 모든 변수를 포함합니다.)

또한 사용자가 레이어 수준 변수를 관리할 수도 있습니다. 버튼을 클릭해서 새 사용자 지정 레이어 수준 변수를 추가합니다. 마찬가지로, 목록에서 사용자 지정 레이어 수준 변수를 선택한 다음 버튼을 클릭해서 제거합니다. 일반 도구 장의값을 변수로 저장 절에서 변수에 관한 더 자세한 정보를 살펴볼 수 있습니다.

### 12.1.17 표고 속성

*Elevation* 탭은*3* 차원 맵 뷰 안에서의 레이어 표고 속성 및단면 도구 도표 *(profile tool charts)* 에서의 3 차원 맵 뷰의 모양을 제어할 수 있는 옵션들을 제공합니다. 그 중에서도 다음을 설정할 수 있습니다:

- *Vertical Reference System*: If the CRS of your vector layer is a compound one (including a Z dimension), then the vertical CRS used for the layer will be the vertical component of the layer CRS. In this case, you cannot manually set a different vertical CRS. If your layer CRS is horizontal (2D), then you can select a specific vertical CRS by clicking on the Select CRS. Vertical reference systems are supported for vector layers by:
  - *Elevation profiles*
  - *Identify Tool Results*
  - *3D map views*
- *Elevation Clamping*: 피처의 고도를 어떻게 표현할지 그리고 표현해야 할지 여부를 정의합니다:
  - *Clamped to terrain*: 피처의 모든 Z 값을 무시하고 지형의 높이로부터 표고를 직접 가져옵니다. 지형의 데이터 정의 *Offset* 값도 입력할 수 있습니다.
  - *Relative to terrain*: 피처의 모든 기존 Z 값을 지형 높이에 더합니다. *Scale* 인자를 지정한 다음 데이터 정의 *Offset* 을 이용해서 표고를 조정할 수 있습니다. 2 차원 도형 레이어에는 이 옵션을 사용할 수 없습니다.
  - *Absolute*: 지형 높이를 무시하고 피처의 Z 값으로부터 표고를 직접 가져옵니다. *Scale* 인자를 지정한 다음 데이터 정의 *Offset* 을 이용해서 표고를 조정할 수 있습니다. (Z 값이 없는) 2 차원 도형 레이어의 경우 데이터 정의 *Base height* 옵션을 대신 설정할 수 있습니다.
- *Enable extrusion*: 피처를 피처의 기반 위로 얼마나 수직 연장시킬지를 제어하려면 *Height* 를 설정하면 됩니다. 이 옵션은 건물 청사진 폴리곤 레이어 같은 2 차원 도형 레이어가 실제로는 3 차원 객체를 표현한다는 사실을 나타내는 데 편리합니다.
- *Custom tolerance*: allows you to override the global sampling tolerance defined in the elevation profile settings, specifically for this vector layer. When enabled, you can set *Tolerance* that determines how the layer is sampled along the profile line. This is useful when you need finer or coarser sampling than the global default, for example, to improve performance or accuracy for complex geometries.
- *Elevation Binding*: 지형에 의존하는 *Elevation clamping* 옵션을 라인 또는 폴리곤 레이어와 결합하는 경우에만 이 옵션을 사용할 수 있습니다. 이 옵션은 피처 표고를 지형 높이에 상대적으로 설정하는 방법을 제어합니다. 지형 높이는 다음과 같이 샘플링할 수 있습니다:
  - *Centroid*: 피처의 중심점에서 샘플링합니다. 각 꼭짓점의 Z 값에 중심점의 높이를 추가합니다.

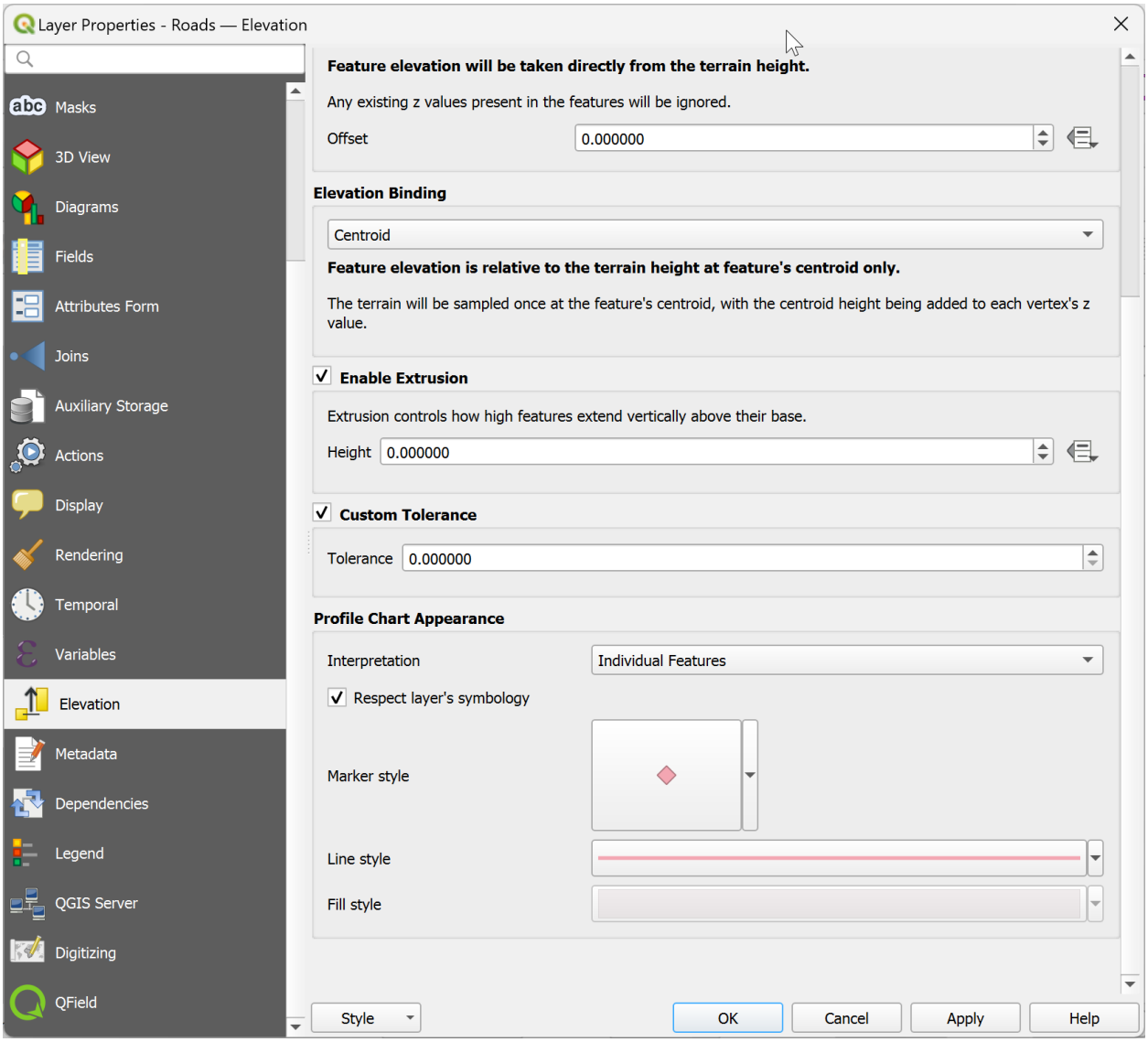



그림 12.65: 벡터 레이어 표고 속성 대화창

  - *Vertex*: 모든 개별 꼭짓점에서 샘플링한 다음 각 꼭짓점의 Z 값에 추가합니다.

- *Profile Chart Appearance*: 단면 도표를 그릴 때 피처를 어떻게 렌더링할지를 제어합니다. 두 가지 주요 *Interpretation* 모드를 사용할 수 있습니다:

  - *Individual features*: 단면 (cross section profile) 라인이 벡터 피처와 교차하는 개별 위치를 샘플링합니다. 이 교차 영역은 레이어 유형에 따라 그리고 압출 (extrusion) 이 적용되었는지에 따라 포인트, 라인, 또는 표면으로 표현될 수 있습니다.

    *Respect layer symbology* 옵션을 체크하면 단면 도표 상에 피처를 대응하는레이어 스타일 로 렌더링할 것입니다. (예를 들어 단면 도표 상에 범주화된 범주들을 가시화시킬 수 있습니다.) 단면 심볼 유형이 레이어의 렌더링 작업자 심볼 유형과 일치하지 않는 경우, 단면 심볼에 렌더링 작업자 심볼의 색상만 적용합니다.

    레이어 설정에 따라, 다음을 이용해서 단면 심볼을 사용자 지정 스타일로 표현할 수 있습니다:

    * 마커 스타일: 압출되지 않은 포인트 및 라인 피처의 경우, 그리고 단면 라인과 접하는 압출되지 않은 폴리곤 피처의 경우 사용자 지정 마커 스타일로 표현할 수 있습니다.
    * 라인 스타일: 압출된 포인트 및 라인 피처의 경우, 그리고 단면 라인과 교차하는 압출되지 않은 폴리곤 피처의 경우 사용자 지정 라인 스타일로 표현할 수 있습니다.
    * 채우기 스타일: 압출된 폴리곤 피처의 경우 사용자 지정 채우기 스타일로 표현할 수 있습니다.

  - *Continuous Surface (e.g. contours)*: 샘플링된 표고값들을 연속되는 라인으로 결합시켜 표고 도표를 개별 피처가 아닌 표면으로 렌더링할 것입니다. 이 옵션은 가시화를 향상시킬 수 있으며, 예를 들어 등고선 또는 측량 표고점과 같이 연속되는 표고 표면을 표현하는 벡터 레이어를 위해 개발되었습니다. 단면의 *Style* 을 다음과 같이 설정할 수 있습니다:

    * 라인 스타일 이 적용된 *Line*
    * 표고 곡선 위 (*Fill above*) 또는 아래 (*Fill below*) 에 채우기 심볼을 사용해서 렌더링한 표고 표면입니다. 표면 심볼은 다음을 이용해서 표현됩니다:
      - 채우기 스타일
      - *Limit*: 채우기 표면이 얼마나 높이 올라가게 될지 결정하는 최대 (제각기 최저) 고도를 설정합니다.

    이에 더해 *Show markers at sampled points* 옵션을 체크하면 설명선 (interpretation line) 위에 마커를 가시화시키고 마커에 *Marker style* 을 적용시킬 수 있습니다.

### 12.1.18 메타데이터 속성

*Metadata* 탭에 사용자 레이어에 대한 메타데이터 리포트를 생성하고 편집할 수 있는 옵션이 있습니다. 더 자세한 정보는메타데이터 를 참조하세요.

### 12.1.19 의존성 속성

*Dependencies* 탭에서 레이어 사이의 데이터 의존성을 선언할 수 있습니다. 레이어에서 — 사용자가 직접 수정하지 않고 — 데이터가 수정되면 다른 레이어의 데이터가 수정될 수도 있는 경우 데이터 의존성이 발생합니다. 예를 들면 레이어의 도형을 수정한 후 데이터베이스 트리거 또는 사용자 지정 PyQGIS 스크립트 작업이 또다른 레이어의 도형을 업데이트하는 경우를 말합니다.

*Dependencies* 탭에서 현재 레이어의 데이터를 외부적으로 변경할 수도 있는 어떤 레이어도 선택할 수 있습니다. 의존 레이어를 정확하게 지정하면, 의존 레이어가 변경됐을 때 QGIS 가 선택한 레이어의 캐시를 무효화할 수 있습니다.

### 12.1.20 범례 속성

*Legend* 속성 탭은레이어 패널 그리고/또는인쇄 조판기 범례 를 위한, 다음과 같은 옵션을 포함하는 고급 설정을 제공합니다:

- 레이어에 적용된 심볼에 따라, 범례에 딱히 인간이 읽을 수 있지도 않고 유용하지도 않은 항목들이 몇 개 나타날 수도 있습니다. *Legend placeholder image* 를 이용하면 *Layers* 패널과 인쇄 조판 범례에 표시되는 대체이미지를 선택 할 수 있습니다.
- *Show label legend*: 범례에 다양한 라벨 설정들의 오버뷰를 항목으로 표시합니다. 라벨 스타일 을 설명과 함께 미리보기할 수 있습니다.
- *Text on symbols*: 몇몇 경우 범례에 있는 심볼에 추가 정보를 추가하는 것이 유용할 수 있습니다. 이 프레임을 통해 레이어 심볼에서 사용되는 어떤 심볼이든, *Layers* 패널 및 인쇄 조판기 범례 양쪽에 있는 심볼 위에 텍스트를 표시하도록 할 수 있습니다. 테이블 위젯에 있는 심볼 옆에 텍스트를 입력하거나 *Set Labels from Expression* 버튼으로 테이블을 채우면 심볼과 텍스트를 매핑할 수 있습니다. *Text Format* 버튼의 글꼴 위젯과 색상 선택기 위젯을 통해 텍스트의 모양을 설정합니다.

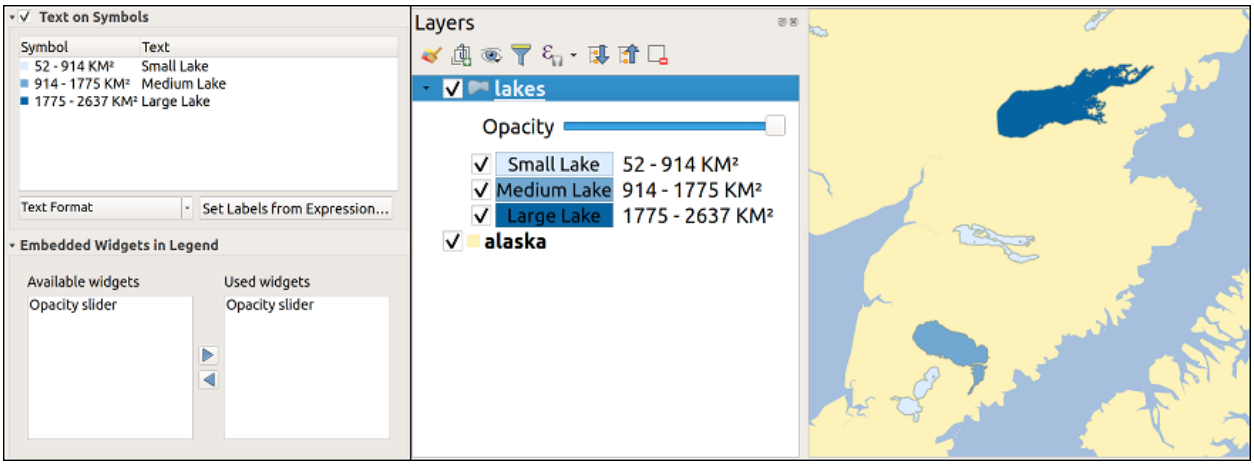



그림 12.66: 심볼 위에 텍스트 설정 (좌), *Layers* 패널에서 텍스트 렌더링 설정 (우)

- 레이어 패널에 있는 레이어 트리 안에 삽입할 수 있는 위젯 목록 가운데 위젯을 선택할 수 있습니다. 레이어 작업 시 자주 쓰이는 몇몇 액션 (투명도, 필터링, 선택, 스타일, 기타 등등의 설정) 에 빨리 접근하기 위한 방법으로 사용됩니다.

  QGIS 가 기본적으로 투명도 위젯을 제공하고는 있지만, 자체 위젯을 등록하고 플러그인이 관리하는 레이어에 사용자 지정 액션을 할당하는 플러그인으로 위젯 목록을 확장할 수 있습니다.

### 12.1.21 QGIS 서버 속성

*QGIS Server* 탭은 *Description*, *Attribution*, *Metadata URL*, 그리고 *Legend URL* 부분들로 이루어져 있습니다.

From the *Description* section, you can change the *Short name* used to reference the layer in requests (to learn more about short names, read services_basics_short_name). You can also add or edit a *Title*, an alternative *WFS Title* and *Abstract* for the layer, or define a *Keyword list* here. These keyword lists can be used in a metadata catalog. If you want to use a title from an XML metadata file, you have to fill in a link in the *Data URL* field.

XML 메타데이터 카탈로그에서 속성 데이터를 얻으려면 *Attribution* 프레임을 이용하십시오.

*Metadata URL* 프레임에서 XML 메타데이터 카탈로그를 가리키는 일반 경로를 추가할 수 있습니다. 이 정보는 후속 세션을 위해 QGIS 프로젝트 파일에 저장되어 QGIS 서버 용으로 쓰일 것입니다.

*Legend URL* 프레임에서 URL 란에 범례 이미지의 URL 을 입력할 수 있습니다. *Format* 드롭다운 옵션을 이용해서 적절한 이미지 포맷을 적용시킬 수 있습니다. 현재 PNG, JPG 및 JPEG 이미지 포맷을 지원합니다.

QGIS 서버에 대해 더 알고 싶다면, QGIS-Server-manual 를 읽어보십시오.

### 12.1.22 디지타이즈 작업 속성

 *Digitizing* 탭은 디지타이즈한 도형의 질을 보장하도록 하는 옵션들을 제공합니다.

#### 자동 수정

*Automatic Fixes* 부분의 옵션들은 추가 또는 수정되는 모든 도형의 꼭짓점에 직접 영향을 줍니다.  *Remove duplicate nodes* 옵션을 체크하면, 바로 앞뒤에 위치하며 정확히 동일한 좌표를 가진 모든 두 꼭짓점을 제거할 것입니다. *Geometry precision* 을 설정하면, 모든 꼭짓점이 환경 설정한 도형 정밀도의 가장 가까운 배수로 반올림될 것입니다. 이 반올림은 레이어 좌표 참조 시스템에서 일어납니다. Z 및 M 값은 반올림되지 않습니다. 디지타이즈 작업 도중 캔버스 상에 그리드가 여러 맵 도구들과 함께 나타납니다.

#### 도형 점검

*Geometry checks* 프레임에서 도형 별 추가 점검을 활성화시킬 수 있습니다. 어떤 도형을 수정한 직후 확인된 오류를 *Geometry validation* 패널을 통해 사용자에게 보고합니다. 점검을 통과하지 못 하는 한 레이어를 저장할 수 없습니다.  *Is valid* 옵션을 체크하면 도형 상에 자기 교차 (self intersection) 같은 오류가 있는지 기본 무결성 검사를 실행할 것입니다.

#### 위상 점검

*Topology checks* 프레임에서는 추가적인 지형 점검을 활성화시킬 수 있습니다. 사용자가 레이어를 저장할 때 지형 점검을 실행할 것입니다. *Geometry validation* 패널을 통해 확인된 오류를 보고할 것입니다. 점검을 통과하지 못 하는 한 레이어를 저장할 수 없습니다. 지형 점검은 수정된 피처의 경계 상자 영역에서 실행됩니다. 다른 피처가 같은 영역에 있을 수도 있기 때문에, 현재 편집 세션에서 발생한 오류는 물론 이 다른 피처와 관련된 지형 오류도 보고될 것입니다.

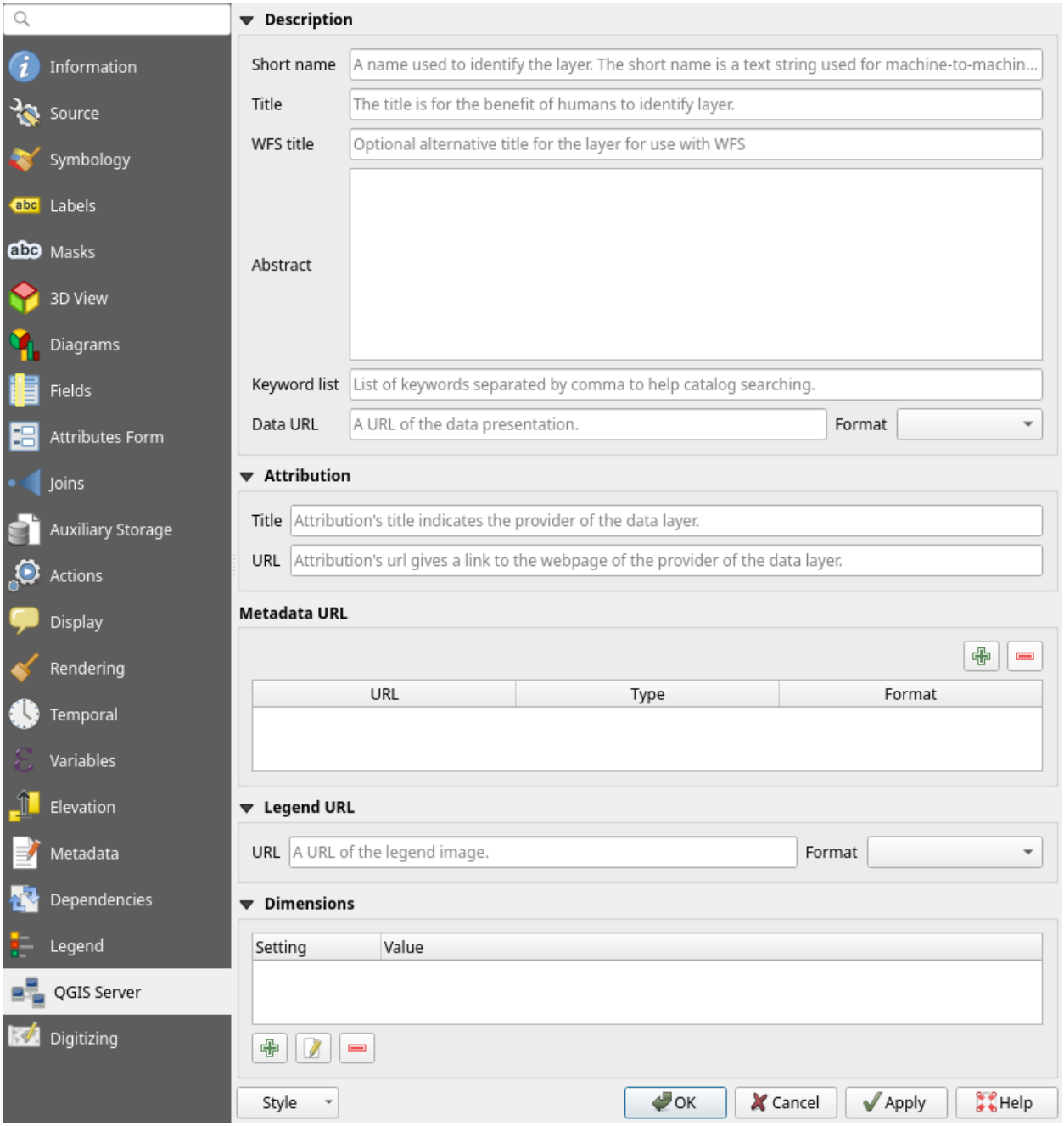



그림 12.67: 벡터 레이어 속성 대화창의 QGIS 서버 탭

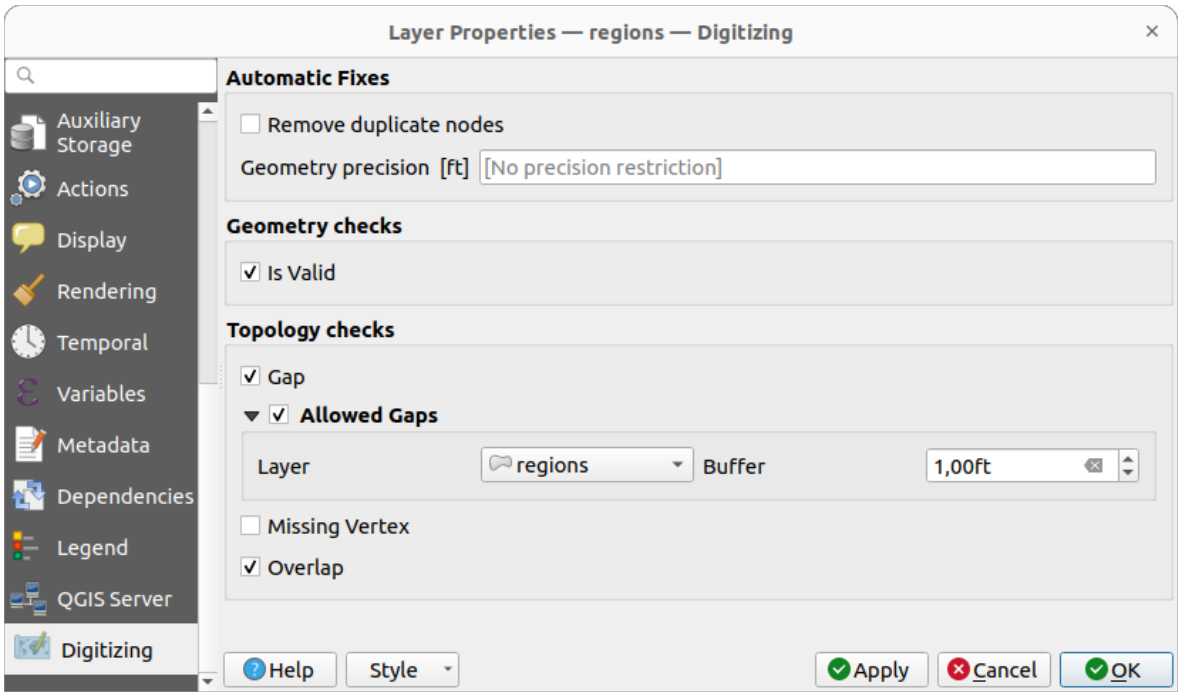



그림 12.68: 벡터 레이어 속성 대화창의 QGIS 디지타이즈 작업 탭

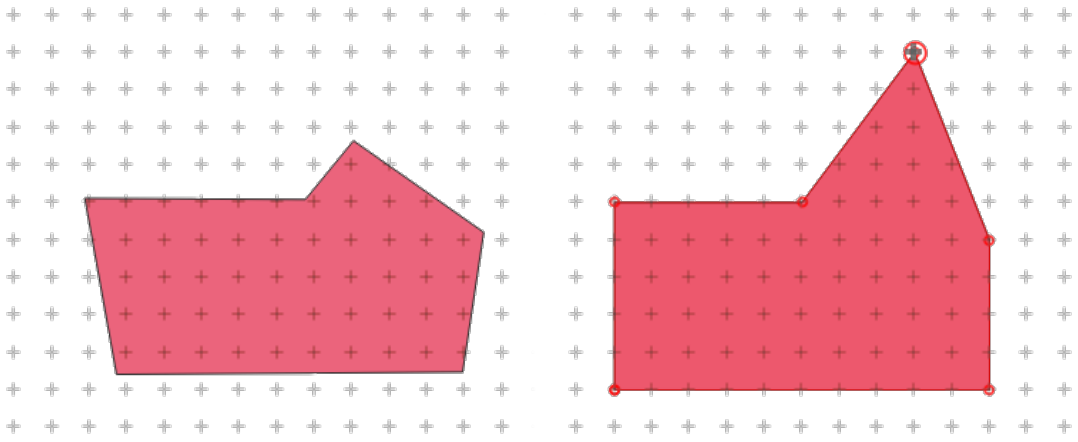
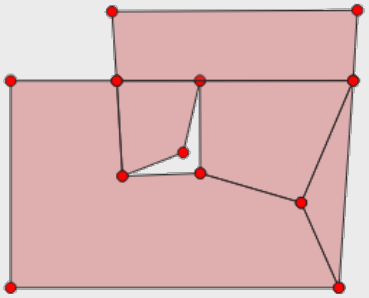
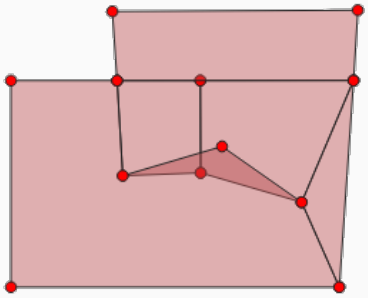
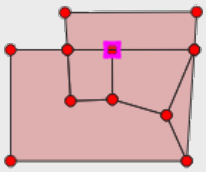


그림 12.69: 제일 위에 있는 꼭짓점을 이동시키면 그리드의 모든 꼭짓점들에 스냅됩니다

| 위상 점검 옵션 | 실제 예시 |
| --- | --- |
| *Gap*: 인접한 폴리곤들 사이의 틈을 점검할 것입니다. |  |
| *Overlap*: 인접한 폴리곤들 사이의 중첩을 점검할 것입니다. |  |
| *Missing vertex*: 인접한 폴리곤들이 경계선을 공유하는데 한쪽 경계에 있는 꼭짓점이 다른쪽 경계에 없는 경우 빠진 꼭짓점을 점검할 것입니다. |  |

### 틈 점검의 예외

틈만 없다면 폴리곤으로 완전히 덮히는 폴리곤 레이어의 영역 안에 있는 틈을 유지하는 것이 바람직한 경우가 종종 있습니다. 예를 들어 토지 이용 레이어는 호수를 나타내는, 받아들일 수 있는 구멍을 가지고 있을 수도 있습니다. 영역이 틈 점검에서 무시당하도록 정의할 수도 있습니다. 이런 영역 안에서는 틈이 허용되기 때문에, 이런 틈을 허용된 틈 *(Allowed Gap)* 영역이라고 부르도록 하겠습니다.

*Allowed Gaps* 아래 있는 틈 점검을 위한 옵션에서, 허용된 틈 레이어 *(Allowed Gaps layer)* 의 환경을 설정할 수 있습니다.

틈 점검을 실행할 때마다, 허용된 틈 레이어 에서 하나 또는 그 이상의 폴리곤으로 덮혀 있는 틈은 위상 오류로 보고되지 않습니다.

추가적인 *Buffer* 를 설정할 수도 있습니다. 이 버퍼는 허용된 틈 레이어 상의 각 폴리곤에 적용됩니다. 이렇게 버퍼를 설정하면 틈 경계의 외곽선을 자잘하게 변경하더라도 틈 점검이 민감하게 잡아내지 않도록 할 수 있습니다.

허용된 틈 을 활성화하면, 디지타이즈 도중 발생한 틈을 보고하는 도형 무결성 검증 도크 (dock) 에서 탐지된 틈 오류를 위한 추가 버튼 (*Add Allowed Gap*) 을 사용할 수 있습니다. *Add Allowed Gap* 버튼을 누르면, 허용된 틈 레이어 에 탐지된 틈의 도형을 가진 새 폴리곤을 삽입합니다. 이렇게 하면 한번에 틈을 허용할 수 있습니다.

#### 도형 검증 패널

앞에서 언급한 어떤 디지타이즈 작업 점검이라도 오류를 발견했을 때 *Geometry Validation* 패널이 나타날 것입니다. 이 대화창은 오류 목록 및 오류 설명을 표시하며, 사용자는 키보드의 화살표 키 또는 패널의 전용 화살표를 이용해서 목록을 살펴볼 수 있습니다.

레이어의 편집 내용을 저장하려면 먼저 모든 문제점을 해결해야 할 것입니다:

1. 오류를 선택한 다음 다음 작업을 할 수 있습니다:
   -  *Zoom to Feature(s)*: 피처 (들) 로 확대/축소
   -  *Zoom to problem*: 오류 위치로 확대/축소
2. 보통 사용하는디지타이즈 작업 도구 를 선택해서 문제점을 수정하십시오.

## 12.2 속성 테이블 작업

속성 테이블은 선택한 레이어의 피처에 관한 정보를 표시합니다. 테이블에 있는 각 행은 (도형을 지닌 또는 지니지 않은) 피처를 나타내며, 각 열은 피처에 관한 한 가지 특정 정보를 담고 있습니다. 테이블에 있는 피처를 검색하고, 선택하고, 이동시키거나 편집할 수도 있습니다.

### 12.2.1 서문: 공간 및 비공간 테이블

QGIS 는 공간 및 비공간 레이어를 로드할 수 있습니다. PostgreSQL, MS SQL 서버, SpatiaLite 및 오라클은 물론 현재 GDAL 에서 지원하는 테이블과 구분자로 구분된 텍스트도 포함합니다. 로드된 레이어는 모두 *Layers* 패널 안에 나열됩니다. 레이어를 공간적으로 활성화시킬 수 있는지 여부가 맵 상에서 레이어와 상호 작용할 수 있는지 여부를 결정합니다.

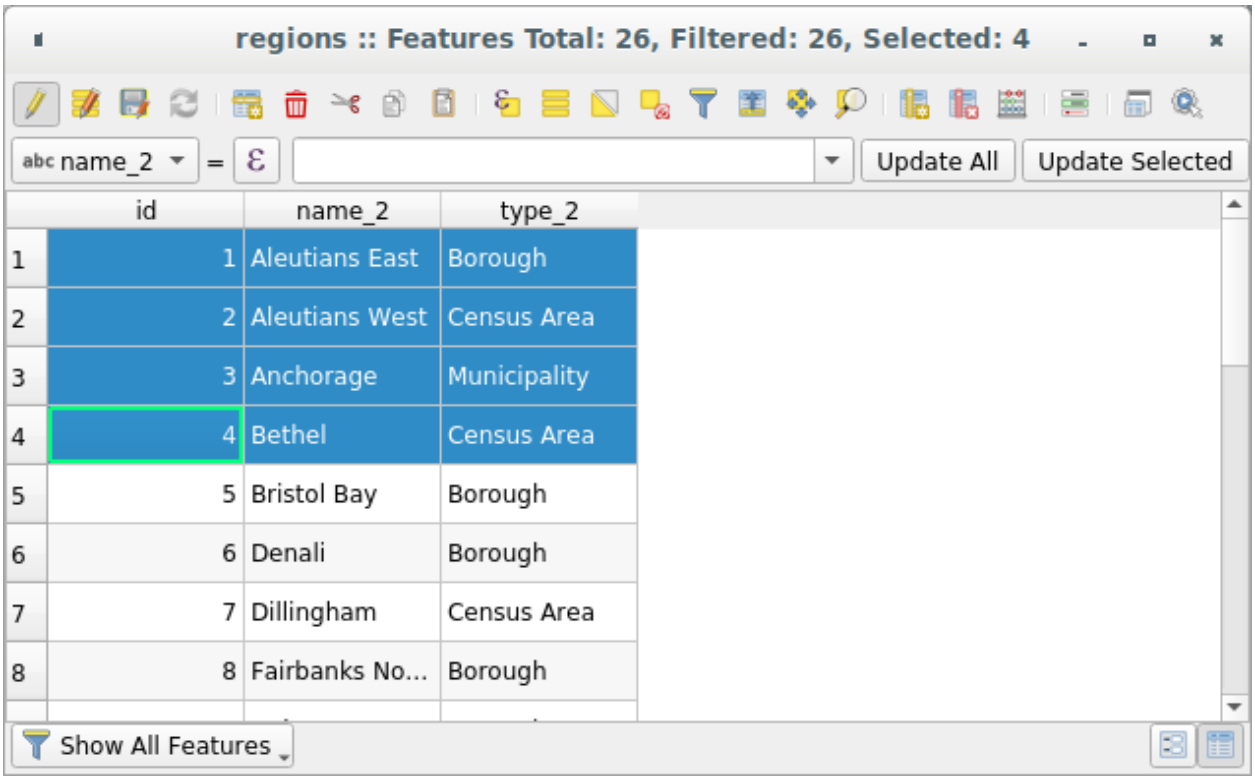
속성 테이블 뷰를 통해 비공간 테이블을 둘러보고 편집할 수 있습니다. 거기에, 비공간 테이블을 사용해서 필드 검색을 할 수도 있습니다. 예를 들면, 디지타이즈 작업 중에 비공간 테이블의 열을 이용해서 속성값 또는 허용되는 값의 범위를 정의한 다음 특정 벡터 레이어에 추가할 수 있습니다. 자세한 내용은속성 양식 속성 에 있는 편집 위젯을 자세히 살펴보세요.

### 12.2.2 속성 테이블 인터페이스 소개

벡터 레이어의 속성 테이블을 열려면, 레이어 패널 에서 해당 레이어를 클릭해서 활성화시키십시오. 그 다음 메인 *Layer* 메뉴에서  *Open Attribute Table* 메뉴 옵션을 선택하십시오. 레이어를 오른쪽 클릭한 다음 컨텍스트 메뉴에서  *Open Attribute Table* 을 선택하거나, 속성 툴바에 있는  *Open Attribute Table* 아이콘을 클릭해도 됩니다. 단축키를 선호한다면, `F6` 키를 누르면 속성 테이블이 열릴 것입니다. `Shift+F6` 키 조합을 누르면 선택한 피처로 필터링한 속성 테이블이 열리고, `Ctrl+F6` 키 조합을 누르면 가시화 피처로 필터링한 속성 테이블이 열릴 것입니다.

이렇게 하면 레이어의 피처 속성을 표시하는 새 창이 열립니다. (그림 12.70 참조) *Settings* → *Options* → *Data sources* 메뉴의 설정에 따라, 도킹 창 (docked window) 또는 일반 창에서 속성 테이블을 열 것입니다. 속성 테이블 제목에 레이어가 공간적으로 제한됐는지는 물론, 레이어에 있는 피처의 총 개수 및 현재 선택한/필터링한 피처의 개수도 표시됩니다.

속성 테이블 창 상단에 있는 아이콘들은 다음과 같은 기능을 제공합니다:




| | id | name_2 | type_2 |
|---|---|---|---|
| 1 | 1 | Aleutians East | Borough |
| 2 | 2 | Aleutians West | Census Area |
| 3 | 3 | Anchorage | Municipality |
| 4 | 4 | Bethel | Census Area |
| 5 | 5 | Bristol Bay | Borough |
| 6 | 6 | Denali | Borough |
| 7 | 7 | Dillingham | Census Area |
| 8 | 8 | Fairbanks No... | Borough |

그림 12.70: “regions”레이어의 속성 테이블

표 12.1: 사용할 수 있는 도구들

| 아이콘 | 라벨 | 목적 | 기본 단축키 |
|---|---|---|---|
| | Toggle editing mode | 편집 기능 활성화 | Ctrl+E |
| | Toggle multi edit mode | 여러 피처들의 여러 필드들을 업데이트 | |
| | Save Edits | 현재 수정 사항을 저장 | |
| | Reload the table | | |
| | Add feature | 도형 없는 새 피처를 추가 | |
| | Delete selected features | 레이어에서 선택한 피처를 제거 | |
| | Cut selected features to clipboard | | Ctrl+X |
| | Copy selected features to clipboard | | Ctrl+C |
| | Paste features from clipboard | 복사한 피처로부터 새 피처를 삽입 | Ctrl+V |
| | Select features using an Expression | | |
| | Select All | 레이어에 있는 모든 피처를 선택 | Ctrl+A |
| | Invert selection | 레이어에 있는 현재 선택을 반전 | Ctrl+R |
| | Deselect all | 현재 레이어에 있는 모든 피처를 선택 해제 | Ctrl+Shift+A |
| | 양식을 이용해서 피처 필터링/선택하기 | | Ctrl+F |

다음 페이지에 계속

표 12.1 – 이전 페이지에서 계속

| 아이콘 | 라벨 | 목적 | 기본 단축키 |
|---|---|---|---|
|  | Move selected to top | 선택한 행을 테이블 맨 위로 이동 | |
|  | Pan map to the selected rows | | `Ctrl+P` |
|  | Zoom map to the selected rows | | `Ctrl+J` |
|  | New field | 데이터소스에 새 필드를 추가 | `Ctrl+W` |
|  | Delete field | 데이터소스에서 필드를 제거 | |
|  | Organize columns | 속성 테이블에서 필드를 표시/숨기기 | |
|  | Open field calculator | 행에 있는 여러 피처들의 필드를 업데이트 | `Ctrl+I` |
|  | Conditional formatting | 테이블 서식 작업 활성화 | |
|  | Dock attribute table | 속성 테이블 도킹/언도킹을 토글 | |
|  | Actions | 레이어 관련 액션 목록 | |

**주의**

데이터 유형 및 사용자의 QGIS 버전과 함께 빌드한 GDAL 라이브러리에 따라, 일부 도구를 사용하지 못 할 수도 있습니다.

이 아이콘들 아래엔 (편집 모드 일 때만 활성화되는) 간편 필드 계산 막대가 있는데, 이 계산 막대를 통해 레이어에 있는 피처들 전부 또는 일부에 계산을 간편하게 적용할 수 있습니다. 이 막대는  Field Calculator (속성값 편집 참조)와 동일한표현식 을 이용합니다.

### 테이블 뷰 대 양식 뷰

QGIS 는 속성 테이블에서 데이터를 쉽게 수정할 수 있는 뷰 모드 2 개를 제공하고 있습니다:

-  Table view: 피처 여러 개의 값을 각 행이 피처를 나타내고 각 열이 필드를 나타내는 표 모드로 표시합니다. 열 헤더를 오른쪽 클릭하면테이블을 표시하는 방식을 환경 설정 할 수 있고, 셀을 오른쪽 클릭하면피처와 대화형 작업 을 할 수 있습니다.

  속성 테이블은 테이블 뷰 모드에서 수직/수평 스크롤 이동을 전환할 수 있는 `Shift+Mouse Wheel` 스크롤링을 지원합니다. 맥 OS 에서도 마우스를 트랙패드로 교체하면 이 기능을 사용할 수 있습니다.

-  Form view: 첫 번째 패널에피처 식별자 를 표시하고 두 번째 패널에는 첫 번째 패널에서 클릭한 식별자의 속성만 표시합니다. 첫 번째 패널 상단에 있는 드롭다운 메뉴에서 속성 (*Column preview*) 또는 표현식 (*Expression*) 을 이용해서"식별자"를 지정할 수 있습니다. 이 드롭다운 메뉴에서 최근 사용한 표현식 10 개를 다시 사용할 수도 있습니다. 양식 뷰는 레이어 필드 환경 설정을 (속성 양식 속성 참조) 사용합니다.

  첫 번째 패널 하단에 있는 화살표를 이용해서 피처 식별자를 탐색할 수 있습니다. 사용자가 탐색하는 대로 두 번째 패널의 피처 속성이 업데이트될 것입니다. 첫 번째 패널 하단의 다음 버튼들 가운데 하나로 맵 캔버스에 있는 활성 피처를 식별하거나 활성 피처로 이동할 수 있습니다:

  -  Highlight current feature: 맵 캔버스에 현재 피처가 가시화되어 있는 경우 현재 피처를 강조합니다.
  -  Automatically pan to current feature: 현재 피처로 자동 이동합니다.

- Zoom to current feature: 현재 피처로 확대/축소합니다.

대화창 우하단에 있는 대응하는 아이콘을 클릭해서 두 뷰 모드 사이를 전환할 수 있습니다.

*Settings* ▸ *Options* ▸ *Data Sources* 메뉴에서 속성 테이블이 열릴 때의 *Default view* 를 설정할 수도 있습니다. 'Remember last view', 'Table view'또는'Form view'옵션 가운데 하나를 선택할 수 있습니다.

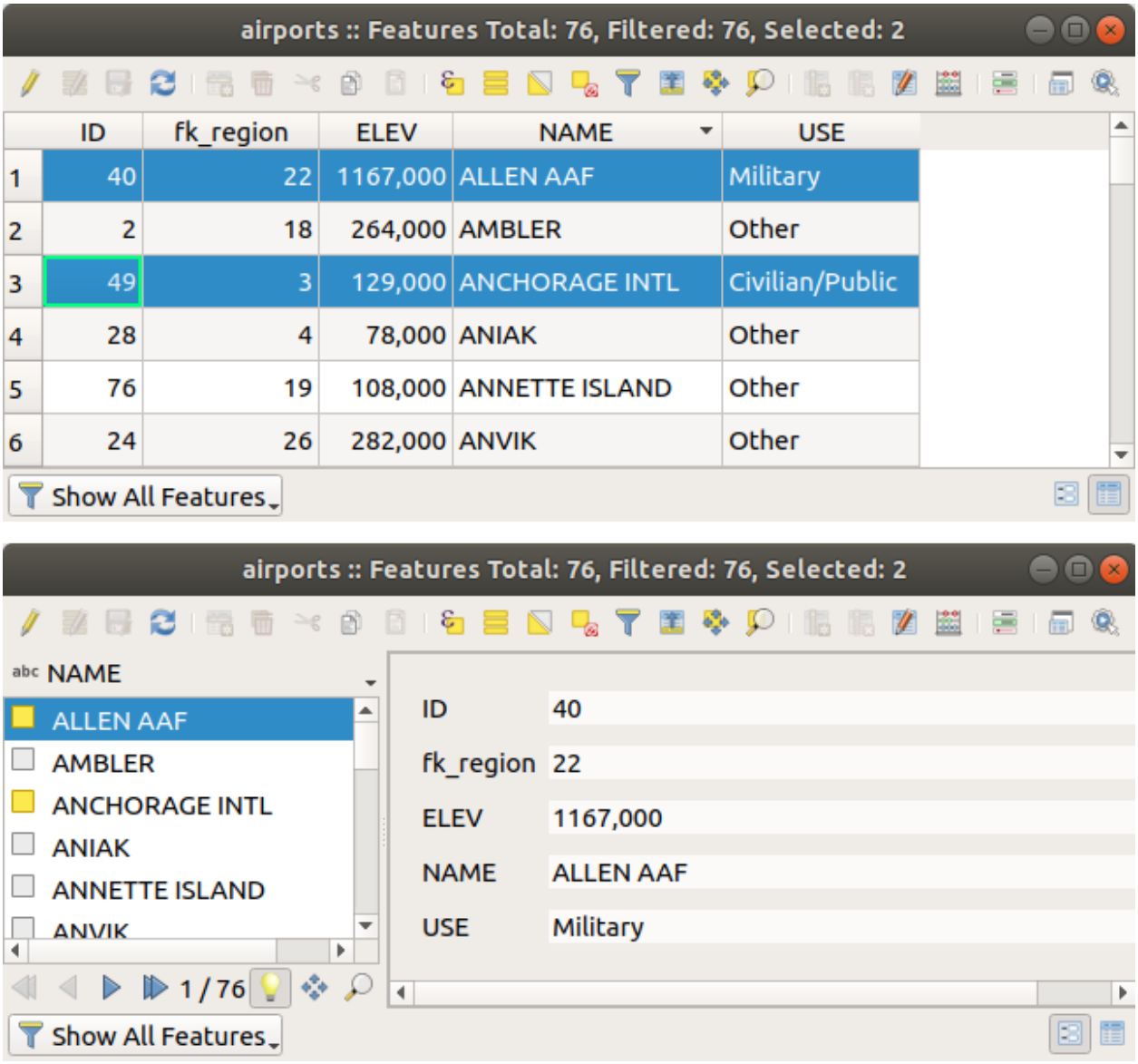



그림 12.71: 속성 테이블의 테이블 뷰 (위) 대 양식 뷰 (아래)

### 열 환경설정하기

테이블 뷰에서 열 헤더를 오른쪽 클릭하면, 다음을 제어할 수 있도록 해주는 도구에 접근할 수 있습니다:

- 열 (들) 의 너비
- 열 (들) 의 가시성 및 순서
- 데이터 정렬 순서

### 열 너비 설정하기

열 헤더를 오른쪽 클릭한 다음 다음 가운데 하나를 선택하면 열 너비를 설정할 수 있습니다:

- *Set width*…: 원하는 값을 입력할 수 있습니다. 기본적으로, 위젯에는 현재 값이 표시됩니다.
- *Set all column widths*…: 모든 열 너비를 동일한 값으로 설정할 수 있습니다.
- *Autosize*: 열에 가장 알맞은 너비로 조정할 수 있습니다.
- *Autosize all columns*: 모든 열을 가장 알맞은 너비로 조정할 수 있습니다.

열 헤더의 오른쪽 경계선을 드래그해서 열 너비를 변경할 수도 있습니다. 이 새 열 너비는 레이어에 저장되어, 다음에 속성 테이블을 열 때 복원될 것입니다.

In the *Data Source Settings*, you can choose to *Autosize all columns by default when opening attribute table*, which will make "Autosize All Columns"the default view every time attribute tables are opened in QGIS.

### 열 숨기기와 정리하기 및 액션 활성화하기

By right-clicking in a column header, you can choose to *Hide column* from the attribute table (in "table view"mode). For more advanced controls, press the Organize columns… button from the dialog toolbar or choose *Organize columns*… in a column header contextual menu.

In the new dialog, you can:

- check/uncheck columns you want to show or hide: a hidden column will disappear from every instance of the attribute table dialog until it is actively restored. It is also possible to:
  - choose *Show All* to display all the fields (columns) and actions in the table
  - choose *Hide All* to hide all the fields (columns) and actions in the table
  - use the *Toggle selection* to invert visibility of the current selection of columns. You can use *keyboard combination* for selecting multiple columns.
- 속성 테이블에서 열을 재정렬하기 위해 항목을 드래그 & 드롭할 수 있습니다. 이 작업은 테이블 렌더링을 변경하는 것이지 레이어 데이터소스의 필드 순서를 변경하는 것이 아니라는 점을 기억하십시오.
- 각 행에 활성화된 액션들의 드롭다운 박스 또는 버튼 목록을 표시하는 새로운 가상 *Actions* 열을 추가할 수 있습니다. 액션에 관한 자세한 정보를 알고 싶다면액션 속성 을 참조하세요.

### 행 정렬하기

열 헤더를 클릭하면, 어떤 열로든 행들을 정렬시킬 수 있습니다. 작은 화살표는 정렬 순서를 나타냅니다. (아래를 가리키는 화살표는 최상위 행으로부터 아래 방향으로 값을 내림차순으로 정렬하고, 위를 가리키는 화살표는 최상위 행으로부터 아래 방향으로 값을 오름차순으로 정렬한다는 의미입니다.) 열 헤더의 컨텍스트 메뉴에서 *Sort…* 항목을 선택하고 표현식을 작성해서 행들을 정렬시킬 수도 있습니다. 예를 들어 열 여러 개를 이용해서 행들을 정렬시키려면 `concat(col0, col1)` 같은 표현식을 작성하면 됩니다.

양식 뷰에서는,  *Sort by preview expression* 옵션을 통해 피처 식별자를 정렬할 수 있습니다.

행을 정렬하는 작업은 테이블 렌더링에만 영향을 줄 뿐 레이어 데이터소스 내부의 피처 순서를 변경하지는 않는다는 점을 기억하십시오.

> **팁**
>
> **서로 다른 유형의 열을 기반으로 정렬하기**
>
> 문자열과 숫자 유형의 열들을 기반으로 속성 테이블을 정렬하려 할 때, `concat("USE", "ID")` 표현식이 문자열 값을 (예를 들어 `'Borough105' < 'Borough6'`) 반환하기 때문에 예기치 못한 결과로 이어질 수도 있습니다. 대신 `concat("USE", lpad("ID", 3, 0))` 과 같은 표현식을 사용하면 `'Borough105' > 'Borough006'` 을 반환하므로 이런 오류를 해결할 수 있습니다.

### 테이블 셀의 조건부 서식 작업

조건부 서식을 설정하면, 피처에 대해 사용자 지정 조건을 적용시켜 사용자가 특별히 집중하고자 하는 속성 테이블 항목을 강조할 수 있습니다:

- 도형 (예를 들어 다중 부분 피처, 작은 면적을 가진 피처, 또는 지정한 맵 범위 안에 있는 피처 등등)
- 필드 값 (예를 들어 값을 한계값과 비교하거나, 비어 있는 셀을 식별하거나, 값을 복제하거나 등등)

테이블 뷰의 (양식 뷰에서는 활성화할 수 없습니다) 속성 창 우상단에 있는  Conditional formatting 버튼을 클릭하면 조건부 서식 패널을 활성화할 수 있습니다.

이 새 패널에서  *Field* 또는  *Full row* 의 서식을 렌더링하기 위한 새 규칙을 추가할 수 있습니다. 새 규칙을 추가하는 양식이 열리는데, 다음 항목을 정의할 수 있습니다:

- 규칙의 이름
- 표현식 작성기 함수를 이용한 조건
- 서식: 사전 정의된 서식 목록에서 선택하거나, 다음과 같은 속성을 기반으로 생성할 수도 있습니다:
  - 배경 및 텍스트 색상
  - 아이콘 사용 여부
  - 볼드체, 이탤릭체, 밑줄, 취소선
  - 폰트

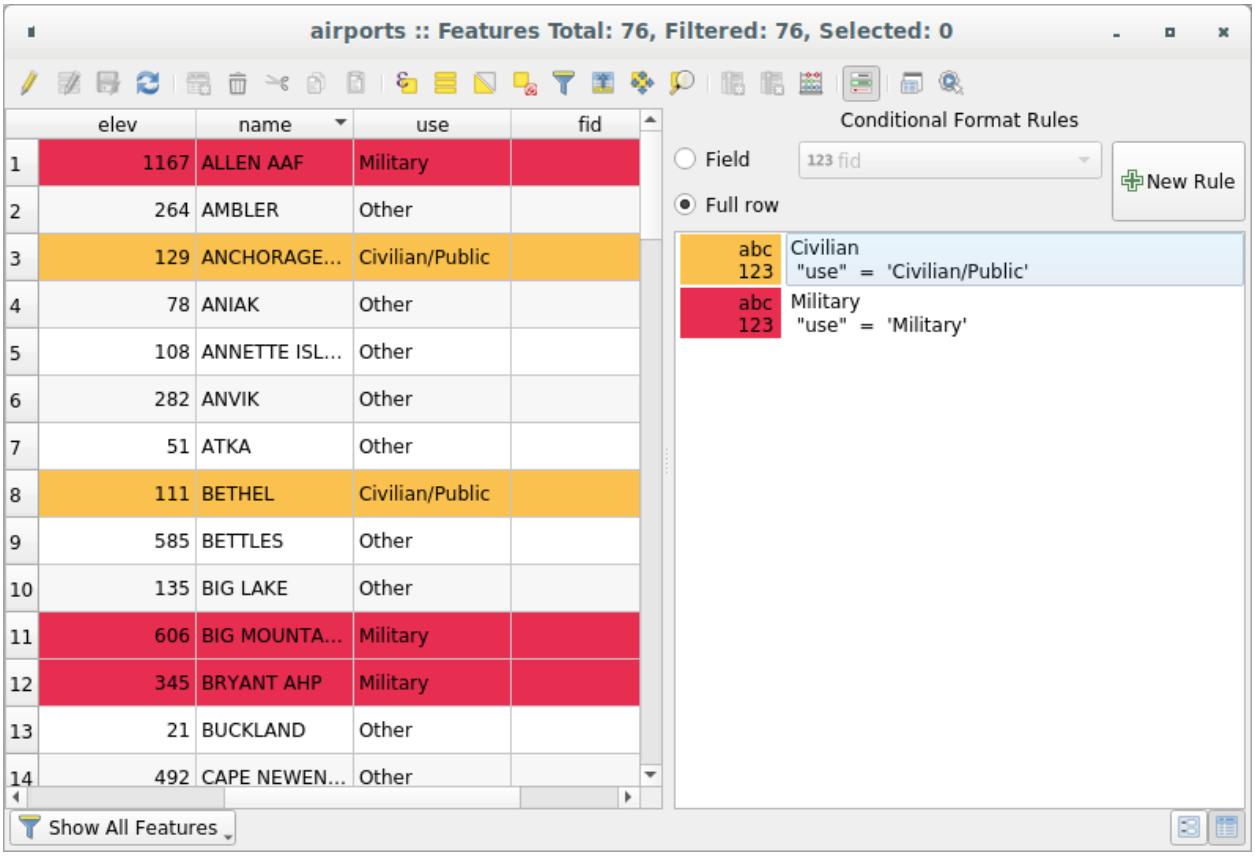



그림 12.72: 속성 테이블의 조건부 서식

### 12.2.3 속성 테이블의 피처와 상호작용

#### 피처 선택

테이블 뷰에서, 속성 테이블의 각 행은 레이어에 있는 유일 피처의 속성들을 표시합니다. 행을 선택하면 피처도 선택되며 마찬가지로 맵 캔버스에서 (도형을 사용할 수 있는 레이어인 경우) 피처를 선택하면 속성 테이블에서 해당 행이 선택됩니다. 맵 캔버스 (또는 속성 테이블) 에서 선택한 피처의 집합을 변경하는 경우, 그에 따라 속성 테이블 (또는 맵 캔버스) 에 있는 선택도 업데이트됩니다.

행 왼쪽에 있는 행 번호를 클릭하면 행을 선택할 수 있습니다. `Ctrl` 키를 누른채 클릭하면 **여러 행들** 을 선택 표시할 수 있습니다. `Shift` 키를 누른 채 행 왼쪽에 있는 행 헤더를 몇 개 클릭하면 **연속 선택** 을 할 수 있습니다. 현재 커서 위치와 클릭한 행 사이의 모든 행들이 선택될 것입니다. 테이블에 있는 셀을 클릭해서 속성 테이블에서 커서 위치를 이동해도 행 선택이 변경되지는 않습니다. 맵 캔버스에서 선택을 변경해도 속성 테이블에서 커서 위치가 이동되지 않습니다.

속성 테이블의 양식 뷰에서는, 기본적으로 왼쪽 패널에 표시된 필드의 값으로 (표시 속성 참조) 피처를 식별합니다. 패널 상단에 있는 드롭다운 목록을 통해 기존 필드를 선택하거나 또는 사용자 지정 표현식을 이용해서 이 식별자를 대체할 수 있습니다. 컨텍스트 메뉴에서 피처 목록을 정렬하도록 선택할 수도 있습니다.

오른쪽 패널에 피처의 속성을 표시하려면 왼쪽 패널에 있는 값을 클릭하십시오. 피처를 선택하려면, 식별자 왼쪽에 있는 정사각형 심볼 안을 클릭해야 합니다. 기본적으로, 심볼이 노랑색으로 바뀔 겁니다. 테이블 뷰와 마찬가지로, 앞에서 설명한 대로 조합키를 이용해서 여러 피처를 한꺼번에 선택할 수 있습니다.

마우스로 피처를 선택하는 것 외에도, 속성 테이블 툴바에서 다음과 같은 툴을 이용해서 피처의 속성에 기반한 자동 선택을 할 수 있습니다 (더 자세한 정보와 용례를 알고 싶다면자동 선택 항 및 그 다음 항을 참조하세요):

-  *Select By Expression…*
-  *Select Features By Value…*
-  *Deselect All Features from the Layer*
-  *Select All Features*
-  *Invert Feature Selection*

양식을 이용해서도 피처를 선택할 수 있습니다.

#### 피처 필터링

속성 테이블에서 피처를 선택하고 나면, 테이블에 선택한 레코드만 표시하기를 바랄 수도 있습니다. 속성 테이블 대화창 좌하단에 있는 드롭다운 목록에서 *Show Selected Features* 항목을 선택하면 간단하게 선택한 레코드만 볼 수 있습니다. 이 목록은 다음과 같은 필터를 제공하고 있습니다:

-  *Show All Features*
-  *Show Selected Features - Layer* 메뉴에서 *Open Attribute Table (Selected Features)* 항목을 선택하거나 *Attributes Toolbar* 를 이용하거나 `Shift+F6` 단축키를 누르는 것과 동일합니다.
-  *Show Features visible on map - Layer* 메뉴 또는 *Attributes Toolbar* 에서 *Open Attribute Table (Visible Features)* 항목을 선택하거나, 또는 `Ctrl+F6` 단축키를 누르는 것과 동일합니다.
-  *Show Features with Failing Constraints* - 제약조건 을 만족시키지 못 하는 피처만 표시하도록 피처를 필터링할 것입니다. 만족시키지 못 하는 제약조건이 강한 (hard) 조건인지 또는 약한 (soft) 조건인지에 따라, 해당 필드 값이 각각 어두운 주황색 또는 밝은 주황색 셀로 표시됩니다.

-  *Show Edited and New Features* - *Layer* 메뉴 또는 *Attributes Toolbar* 에서 *Open Attribute Table (Edited and New Features)* 항목을 선택하는 것과 동일합니다.
- *Field Filter* - 필드 값을 기반으로 필터링할 수 있습니다: 목록에서 열을 선택한 다음 값을 입력 또는 선택하고 `Enter` 키를 눌러 필터링합니다. 그러면 속성 테이블에 `num_field = value` 또는 `string_field ilike '%value%'` 표현식과 일치하는 피처만 표시합니다. 문자열을 좀 더 엄격하게 필터링하려면  *Case sensitive* 옵션을 체크하면 됩니다.
-  *Advanced filter (Expression)*: 표현식 작성기 대화창을 엽니다. 이 대화창에서 테이블 행들과 일치시킬복잡 표현식 을 생성할 수 있습니다. 예를 들어 하나 이상의 필드를 이용해서 테이블을 필터링할 수 있습니다. 이를 적용하면, 필터 표현식이 양식 하단에 표시될 것입니다.
-  *Stored filter expressions* : 사용자 속성 테이블을 필터링하는 데 자주 쓰이는저장된 표현식 을 불러옵니다.

양식을 이용해서도 피처를 필터링할 수 있습니다.

**참고**

속성 테이블에서 레코드를 필터링해도 레이어에서 피처를 필터링하지는 않습니다. 일시적으로 테이블에서 숨겨진 것뿐으로, 맵 캔버스에서 접근할 수도 있고 필터를 제거해서 접근할 수도 있습니다. 레이어에서 피처를 실제로 숨기는 필터를 원한다면, 쿼리 작성기 를 이용하십시오.

**팁**

`Show Features Visible on Map` 으로 데이터소스 필터링 업데이트

성능 향상을 목적으로, 속성 대화창이 열릴 때 속성 테이블에 표시되는 피처가 공간적으로 맵 캔버스 범위로 제한돼 있는 경우 (이 방법에 대해서는데이터소스 옵션 을 참조하세요) 새 캔버스 범위 상에서 *Show Features Visible on Map* 을 선택하면 공간 제약 조건을 업데이트합니다.

### 필터 표현식 저장하기

사용자가 속성 테이블 필터링에 사용한 표현식을 향후 호출을 위해 저장할 수 있습니다. *Field Filter* 또는 *Advanced Filter (expression)* 항목을 사용하는 경우, 속성 테이블 대화창 하단의 텍스트 위젯에 사용한 표현식을 표시합니다. 이 텍스트 란 옆에 있는  Save expression with text as name 아이콘을 클릭하면 프로젝트에 해당 표현식을 저장합니다. 이 아이콘 옆에 있는 드롭다운 메뉴를 클릭하면 (*Save expression as…*) 저장할 표현식의 이름을 사용자 지정할 수 있습니다. 저장한 표현식이 표시되고 나면, 아이콘이  으로 변화하고 드롭다운 메뉴에서 표현식을 편집하고 (*Edit the expression*) 이름을 새로 명명하거나, 또는 저장된 표현식을 삭제 (*Delete stored expression*) 할 수 있습니다.

필터 표현식은 프로젝트에 저장되며, 속성 테이블의 *Stored filter expressions* 메뉴를 통해 사용할 수 있습니다. 이 저장된 필터 표현식은 활성화된 사용자 프로파일의 모든 프로젝트가 공유하는사용자 표현식 과는 다릅니다.

### 양식을 이용해서 피처 필터링하기/선택하기

Filter/Select features using form 을 클릭하거나 `Ctrl+F` 조합키를 누르면 속성 테이블 대화창이 양식 뷰로 전환되고 모든 위젯도 해당 위젯의 검색형으로 대체될 것입니다.

이 다음부터 설명할 도구 기능은값으로 피처 선택 에서 설명한 것과 유사합니다. 이 항에서 모든 연산자 및 선택 모드에 대한 설명을 찾아볼 수 있습니다.

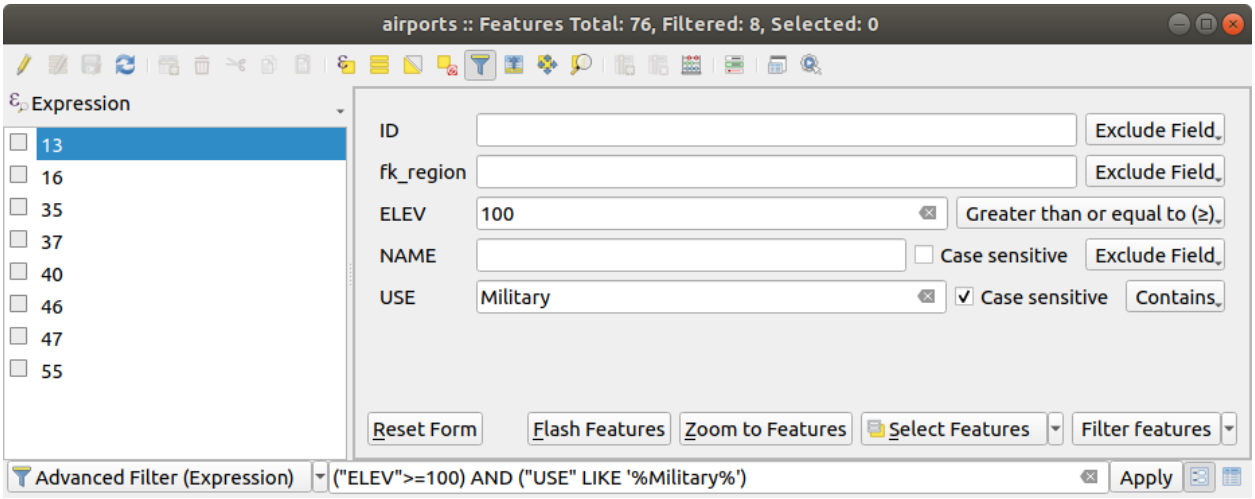



그림 12.73: 필터 양식으로 필터링한 속성 테이블

속성 테이블에서 피처를 필터링/선택하는 경우, 필터를 정의하고 미세 조정할 수 있는 *Filter features* 버튼이 존재합니다. 이 버튼을 사용하면 *Advanced filter (Expression)* 옵션을 촉발해서 양식 하단에 있는 편집 가능한 텍스트 위젯에 대응하는 필터 표현식을 표시합니다.

이미 필터링한 피처가 있는 경우, *Filter features* 버튼 옆에 있는 드롭다운 목록을 이용해서 필터를 미세 조정할 수 있습니다. 다음 옵션을 선택할 수 있습니다:

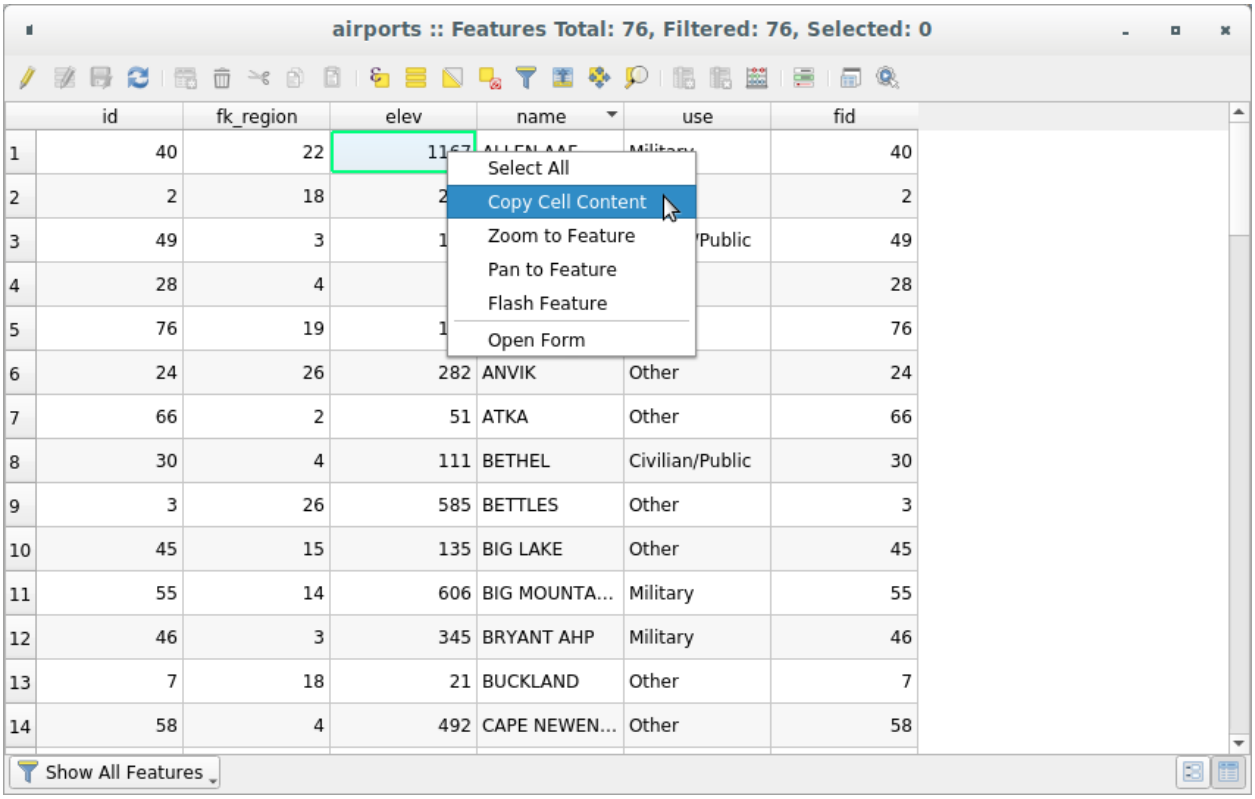
- *Filter within ("AND")*
- *Extend filter ("OR")*

필터를 적용 해제하려면, 좌하단의 풀다운 메뉴에서 *Show all features* 메뉴 옵션을 선택하거나 또는 표현식을 삭제한 다음 *Apply* 버튼을 클릭하거나 `Enter` 키를 누르십시오.

### 피처에 대한 더 많은 액션들

사용자가 속성 테이블에서 피처를 조작할 수 있는 가능성이 몇 가지 있습니다. 셀을 오른쪽 클릭하면 다음을 할 수 있습니다:

- *Select all*: 피처를 모두 선택 (`Ctrl+A`) 할 수 있습니다.
- *Copy cell content*: 클립보드로 셀의 내용을 복사할 수 있습니다.
- *Zoom to feature*: 피처를 먼저 선택하지 않고서도 피처로 확대/축소할 수 있습니다.
- *Pan to feature*: 피처를 먼저 선택하지 않고서도 피처로 이동할 수 있습니다.
- *Flash feature*: 맵 캔버스에 있는 피처를 강조할 수 있습니다.
- *Open form*: 클릭한 피처의 속성 테이블을 양식 뷰로 켜고 끕니다.
- 이전에 *Layer properties* → *Actions* 탭에서 활성화시켰던액션 목록 을 보이게 할 수 있습니다.




| | id | fk_region | elev | name | use | fid |
|---|---|---|---|---|---|---|
| 1 | 40 | 22 | 1167 | ALLEN AAF | Military | 40 |
| 2 | 2 | 18 | 2 | | | 2 |
| 3 | 49 | 3 | 1 | | /Public | 49 |
| 4 | 28 | 4 | | | | 28 |
| 5 | 76 | 19 | 1 | | | 76 |
| 6 | 24 | 26 | 282 | ANVIK | Other | 24 |
| 7 | 66 | 2 | 51 | ATKA | Other | 66 |
| 8 | 30 | 4 | 111 | BETHEL | Civilian/Public | 30 |
| 9 | 3 | 26 | 585 | BETTLES | Other | 3 |
| 10 | 45 | 15 | 135 | BIG LAKE | Other | 45 |
| 11 | 55 | 14 | 606 | BIG MOUNTA... | Military | 55 |
| 12 | 46 | 3 | 345 | BRYANT AHP | Military | 46 |
| 13 | 7 | 18 | 21 | BUCKLAND | Other | 7 |
| 14 | 58 | 4 | 492 | CAPE NEWEN... | Other | 58 |

그림 12.74: 셀 내용 복사 버튼

If you want to use attribute data in external programs (such as Excel, LibreOffice, or a custom web application), select one or more row(s) and use the Copy selected rows to clipboard button or press `Ctrl+C`. Moreover, in *Settings* → *Options* → *Data Sources* menu you can define the format to paste to with the *Copy features as* option. More details at 데이터소스 설정.

### 12.2.4 속성값 편집

속성 테이블에서 데이터를 수정하려면, 먼저 레이어를 편집 모드로 켜고꺼야 합니다. Toggle Editing 버튼을 클릭하십시오. 레이어 도형 유형 및 클립보드 상태에 따라 속성 테이블 상단 툴바에 몇몇 도구들이 더 활성화됩니다.

그러면 다음과 같은 방법으로 속성값을 편집할 수 있습니다:

- 셀 안에 새 값을 직접 입력합니다. 속성 테이블이 테이블 뷰 모드인지 양식 뷰 모드인지는 상관없습니다. 셀 별로, 피처 별로 변경할 수 있습니다.
- 필드 계산기 이용하기: 행에 이미 존재할 수도 있는 필드를 업데이트하거나, 여러 피처에 대해 필드를 생성합니다. 필드 계산기를 이용해서 가상 필드도 생성할 수 있습니다.
- 간편 필드 계산 막대 이용하기: 필드 계산기와 동일하지만 기존 필드에 대해서만 사용할 수 있습니다.
- 다중 편집 모드를 이용합니다. 행에서 여러 피처의 여러 필드를 업데이트합니다.

Putting the layer into edit mode will also allow you to Paste features from clipboard (`Ctrl+V`) Cut selected rows to clipboard (`Ctrl+X`) or Delete selected features. More details at 피처 잘라내기, 복사 및 붙여넣기.

### 필드 계산기 이용하기

속성 테이블에 있는 Field Calculator 아이콘으로 기존 속성값을 기반으로 계산하거나, 또는 — 예를 들어 도형 피처의 길이 또는 면적을 계산하는 — 함수를 정의할 수 있습니다. 계산 결과를 (가상 필드 일 수도 있는) 새 속성 필드에 저장할 수도 있고, 기존 필드에 있는 값을 업데이트하는 데 쓸 수도 있습니다.

편집을 지원하는 레이어라면 필드 계산기를 사용할 수 있습니다. 필드 계산기 아이콘을 클릭하면 대화창이 열립니다. (그림 12.75 참조) 레이어가 편집 모드가 아닌 경우, 필드 계산기가 계산을 하기 전에 경고 메시지를 표시한 다음 레이어를 편집 모드로 전환할 것입니다.

필드 계산기 대화창은표현식 작성기 대화창을 바탕으로 표현식을 정의하고 계산 결과를 기존 또는 새로 생성한 필드에 적용할 수 있는 완전한 인터페이스를 제공합니다. 필드 계산기 대화창을 이용하려면, 먼저 다음 가운데 어떤 작업을 하기 원하는지 선택해야만 합니다:

1. 레이어 전체에 계산 결과를 적용할지 또는 선택한 피처에만 계산 결과를 적용할지
2. 계산 결과를 추가할 새 필드를 생성할지 또는 기존 필드를 업데이트할지

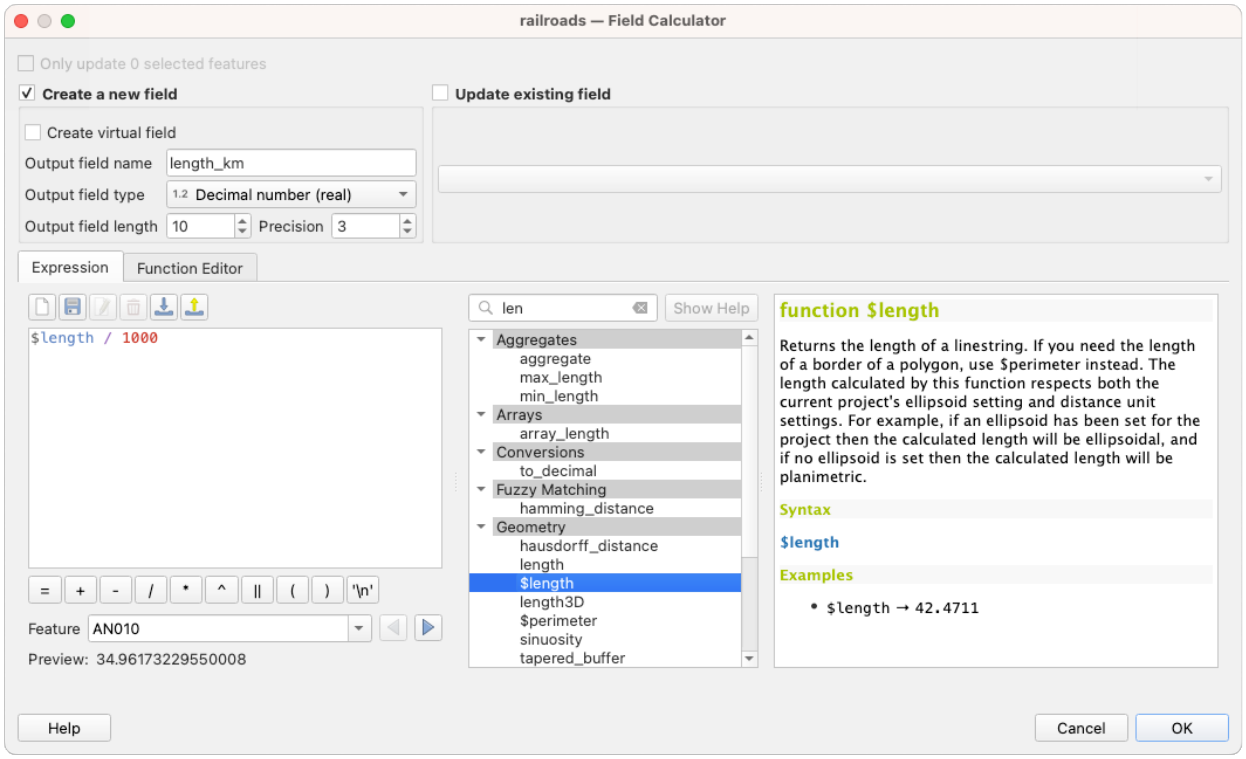



그림 12.75: 필드 계산기

새 필드를 추가하기로 결정한 경우, 필드명, 필드 유형 (정수형, 실수형, 날짜 또는 문자열), 그리고 필요한 경우 필드의 총 길이 및 정밀도를 입력해야 합니다. 예를 들면, 필드 길이를 10 으로 그리고 필드 정밀도를 3 으로 선택한 경우 소수점 앞에 7 자리, 소수점, 그리고 소수점 아래 3 자리를 입력할 수 있습니다.

다음은 *Expression* 탭을 이용하는 경우 필드 계산기가 어떻게 작동하는지 보여주는 짧은 예시입니다. QGIS 예시 데이터셋의 `railroads` 레이어의 길이를 킬로미터 단위로 계산하려 합니다:

1. QGIS 에 `railroads.shp` 셰이프파일을 불러온 다음 Open Attribute Table 아이콘을 클릭합니다.
2. Toggle editing mode 아이콘을 클릭한 다음 Field Calculator 아이콘으로 필드 계산기 대화창을 엽니다.

3. *Create a new field* 체크박스를 체크해서 계산 결과를 새로운 필드로 저장하도록 합니다.
4. *Output field name* 산출 필드명을 `length_km` 으로 설정하십시오.
5. *Output field type* 산출 필드 유형을 `Decimal number (real)` 로 선택하십시오.
6. *Output field length* 산출 필드 길이를 `10` 으로, *Precision* 정밀도를 `3` 으로 설정하십시오.
7. *Geometry* 그룹의 `$length` 를 더블클릭해서 필드 계산기 표현식 란에 도형의 길이를 추가하십시오. (표현식이 작성되는 대로 표현식 란 아래에 실시간으로 업데이트되는 산출물의 미리보기를 60 글자까지 볼 수 있습니다.)
8. 필드 계산기 표현식 란에 `/ 1000` 을 입력해서 표현식을 완성한 다음 *OK* 버튼을 클릭하십시오.
9. 이제 속성 테이블에서 새 필드 *length_km* 를 찾아볼 수 있습니다.

### 가상 필드 생성하기

가상 필드란 실시간으로 계산되는 표현식을 기반으로 한 필드를 말합니다. 즉 기저 파라미터가 변하자마자 그 값을 자동적으로 업데이트한다는 뜻입니다. 표현식은 레이어에 있는 모든 피처에 적용되며 한 번만 설정하면 됩니다. 기저 값이 변할 때마다 필드를 다시 계산해야 할 필요가 없습니다. 예를 들어, 사용자가 피처를 디지타이즈하는 과정에서 면적을 평가해야 하는 경우 또는 변할 수도 있는 (예: `now()` 함수 이용 시) 날짜들 사이의 기간을 자동으로 계산해야 하는 경우 가상 필드를 이용할 수 있습니다.

*Field calculator* 대화창을 통해 가상 필드를 생성하는데, 정규 필드와동일한 과정 을 따릅니다. *Create virtual field* 옵션을 체크하고 사용자 표현식이 생성하게 될 데이터와 호환되는 필드 유형을 사용해야 한다는 점만 기억하십시오.

레이어 속성 대화창의 *Fields* 탭을 통해 가상 필드를 편집합니다. (필드 속성 을 참조하세요.) *Comment* 열에서 필드를 정의하는 표현식을 볼 수 있으며, 그 옆에 있는 ε 버튼을 클릭하면 업데이트를 위한 표현식 편집기 창이 열립니다.

> **참고**
>
> **가상 필드 사용법**
>
> - 필드 생성 시에만 필드를 가상 필드라고 설정할 수 있습니다.
> - 가상 필드는 레이어 속성에 영구히 유지되지 않습니다. 가상 필드가 생성된 프로젝트의 프로젝트 파일에만 저장되고 사용할 수 있다는 뜻입니다.

### 간편 필드 계산 막대 사용하기

언제든 사용할 수 있는 필드 계산기와는 달리, 속성 테이블 상단에 있는 간편 필드 계산 막대는 레이어가 편집 모드인 경우에만 표시됩니다. 간편 필드 계산 막대는 표현식 엔진의 도움을 받아 이미 존재하는 필드를 더 빨리 편집할 수 있습니다:

1. 드롭다운 목록에서 업데이트할 필드를 선택하십시오.
2. 사용자가 표현식을 직접 입력하거나 또는 ε 표현식 버튼을 이용해서 표현식을 작성해서 텍스트란을 채우십시오.
3. 사용자의 필요에 따라 *Update All*, *Update Selected* 또는 *Update Filtered* 버튼 가운데 하나를 클릭하십시오.

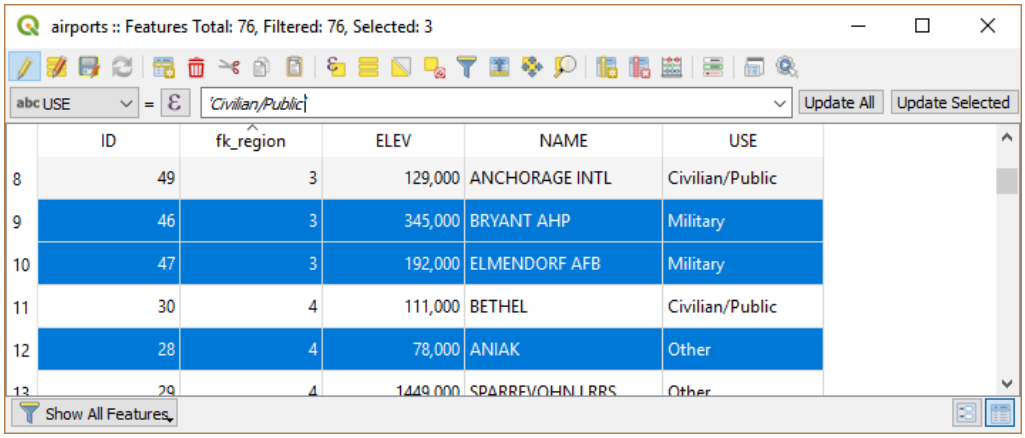



그림 12.76: 간편 필드 계산 막대

### 다중 필드 편집하기

이전 도구들과는 달리, 다중 편집 모드는 동시에 서로 다른 피처들의 여러 속성들을 편집할 수 있습니다. 레이어가 편집 모드로 전환되면, 다음과 같이 다중 편집 기능에 접근할 수 있습니다:

- 속성 테이블 대화창에 있는 툴바에서 Toggle multi edit mode 버튼을 클릭하십시오.
- *Edit* → *Modify attributes of selected features* 메뉴를 선택하십시오.

> **참고**
>
> 속성 테이블에 있는 도구와는 달리, *Edit* → *Modify Attributes of Selected Features* 메뉴 옵션을 선택하면 속성 변경 사항을 입력할 수 있는 모달 대화창이 열립니다. 즉, 실행 전에 피처를 선택해놓아야 합니다.

행에서 여러 필드를 편집하려면:

1. 편집하고자 하는 피처를 선택합니다.
2. 속성 테이블 툴바에서 버튼을 클릭하면 대화창이 양식 뷰로 전환될 것입니다. 이 단계에서 피처를 선택할 수도 있습니다.
3. 속성 테이블의 오른쪽에 선택한 피처의 필드들 (및 값들) 이 표시됩니다. 각 필드 옆에 현재 다중 편집 상태를 표시해주는 다음과 같은 새 위젯들이 나타납니다:
   - 선택한 피처들이 이 필드에 서로 다른 값들을 담고 있습니다. 텅 빈 것으로 표시되며 각 피처가 원본 값을 유지할 것입니다. 위젯의 드롭다운 목록에서 필드의 값을 리셋할 수 있습니다.
   - 선택한 모든 피처들이 이 필드에 동일한 값을 지니며 양식에 표시된 값이 유지될 것입니다.
   - 필드 편집이 끝나 선택한 모든 피처에 입력한 값을 적용할 것입니다. 대화창 상단에 사용자 변경 사항을 적용할지 또는 리셋할지를 묻는 메시지를 표시합니다.

   이 위젯 가운데 하나를 클릭하면 필드 별로 현재 값을 설정하거나 원본 값으로 리셋할 수 있습니다. 즉 필드 별로 사용자 변경 사항을 되돌릴 수 있다는 뜻입니다.
4. 사용자가 원하는 필드를 변경합니다.
5. 상단의 메시지 텍스트 또는 왼쪽 패널의 다른 모든 피처에 있는 **Apply changes** 버튼을 클릭합니다.

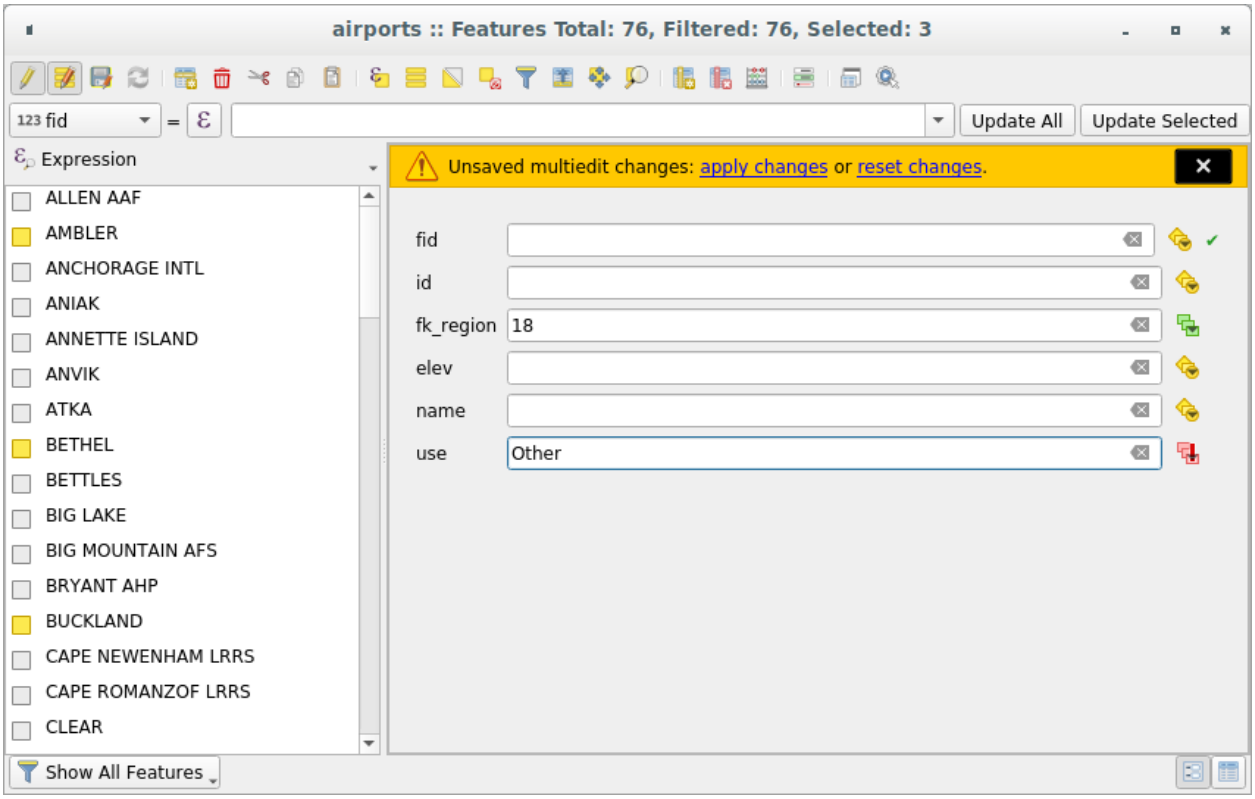



그림 12.77: 여러 피처의 필드들을 편집하기

**선택한 모든 피처** 에 변경 사항을 적용할 것입니다. 선택한 피처가 없는 경우, 전체 테이블을 사용자 변경 사항으로 업데이트합니다. 수정 작업은 단일 편집 명령어로 이루어집니다. 따라서 Undo 아이콘을 클릭하면 선택한 모든 피처의 속성 변경 사항을 한 번에 되돌릴 것입니다.

> **참고**
>
> 다중 편집 모드는 자동 생성된 드래그 & 드롭 양식에서만 사용할 수 있습니다. (사용자 데이터용 양식을 사용자 지정하기 을 참조하세요.) 사용자 UI 양식은 지원하지 않습니다.

### 12.2.5 식별 도구를 통해 피처 속성을 탐색하기

피처 식별 도구를 사용해서 맵 캔버스에 어떤 피처의 모든 속성을 표시할 수 있습니다. 속성 테이블에서 피처를 검색해야 할 필요 없이 피처의 모든 데이터를 빠르게 살펴보고 검증할 수 있는 방법입니다.

벡터 레이어에 *Identify features* 도구를 사용하려면 다음 단계를 수행하십시오:

1. 레이어 패널에서 벡터 레이어를 선택합니다.
2. 툴바에 있는 *Identify features* 도구를 클릭하거나, 또는 `Ctrl+Shift+I` 단축키를 누릅니다.
3. 맵 뷰에서 피처 하나를 클릭합니다.

The *Identify results* panel will display different features information in a tree view. There are two columns in the panel: *Feature* and *Value*. The first item is the name of the layer and its children are its identified feature(s). Each feature is identified by the *display name* field along with its value. Other information about the feature follows:

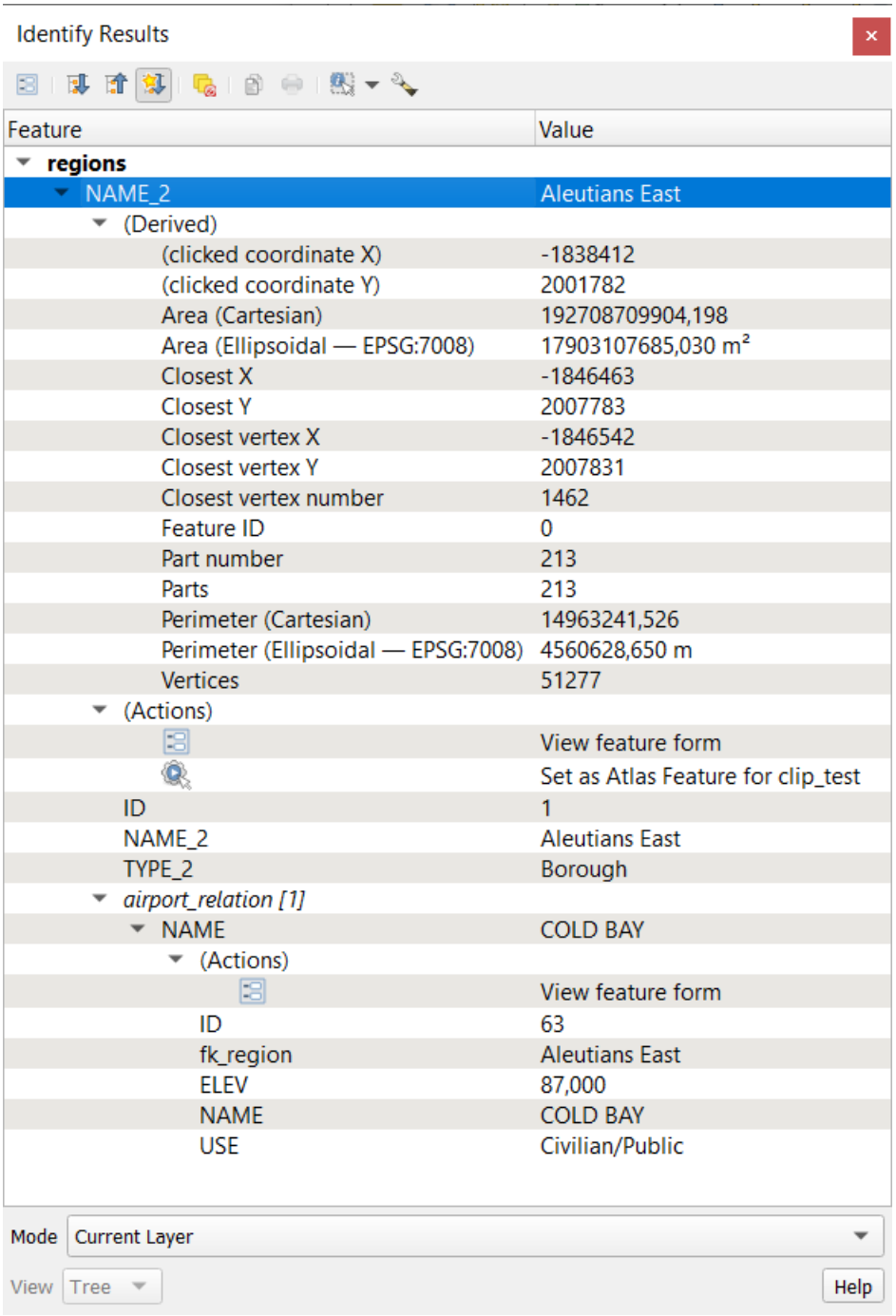



그림 12.78: Identify Results dialog

- **Derived** 부분 - 레이어에 있는 다른 정보로부터 계산한 또는 파생시킨 정보입니다. 예를 들어 폴리곤의 면적 또는 라인의 길이와 같은 정보입니다. 이 부분에서 다음과 같은 일반 정보를 찾아볼 수 있습니다:
  - Depending on the geometry type, the cartesian measurements of length, perimeter or area in the layer's CRS units. For 3D line vectors, the cartesian line length is available.
  - *Project Properties* 대화창에서 (*General* -> Measurements`) 설정된 타원체가 있는 경우, 도형 유형에 따라 타원체 상에서 지정한 단위로 측정된 피처의 길이, 둘레 또는 면적을 구할 수 있습니다.
  - 피처가 보유한 도형 부분의 개수와 클릭한 부분의 개수
  - 피처가 보유한 꼭짓점의 개수
  - Coordinate information, using the project properties *Coordinates display* settings:
    - `X` and `Y` coordinate values of the clicked point
    - the number of the closest vertex to the clicked point
    - `X` and `Y` coordinate values of the closest vertex (and `Z`/`M` if applicable)
    - if you click on a curved segment, the radius of that section is also displayed.
    - if both the vector layer and the project have vertical datums set and they differ, the `Z` value will be displayed for both datums.
- **Actions**: Actions can be added to the identify feature windows. The action is run by clicking on the action label. By default, only one action is added, namely `View feature form` for editing. You can define more actions in the layer's properties dialog (see 액션 속성).
- **Data attributes**: 사용자가 클릭한 피처의 속성 필드 및 값들의 목록입니다.

> **참고**
>
> Links in the feature's attributes are clickable from the *Identify Results* panel and will open in your default web browser.

- When a vector layer has defined *relations*, the *Identify Results* panel can display both **referenced** and **referencing** related features. To view these relations, ensure that the *Show relations* option is enabled in the *Identify Settings*. Available information for each related feature include:
  - the name of the relation and the linked layer
  - the entry in the referenced or referencing field, identifying the feature
  - the activated actions
  - the list of attribute fields and values

  You can expand related features to explore connected records through multiple levels, including many-to-many (n:m) and polymorphic relations. Only nodes you explicitly expand are loaded, preventing excessive nesting.

  To focus on a specific related record, right-click it and choose *Identify Feature*. This re-centers the identify results on the selected feature, starting a new identification tree from it and effectively limiting the visible nesting depth. Features already shown in ancestor nodes are automatically omitted to avoid duplicates or circular relations.

## 12.3 편집 작업

QGIS has various capabilities for editing OGR, SpatiaLite, PostgreSQL, MS SQL Server and Oracle Spatial vector layers and tables. They can be of 2D or 3D geometry type.

> **참고**
>
> GRASS 레이어 편집 과정은 조금 다릅니다. 자세한 내용은*GRASS 벡터 레이어 디지타이즈 및 편집하기* 를 참조하세요.

> **주의**
>
> **병행 편집**
>
> QGIS 는 사용자가 피처를 편집하는 도중에 다른 사용자가 동일한 피처를 편집하고 있지는 않은지 추적하지 않습니다. 마지막에 저장하는 변경 사항이 적용될 것입니다.

> **팁**
>
> **편집 내용 검증하기**
>
> *Layer Properties → Digitizing* 탭에서 레이어 수준의 지속 검증을 활성화시킬 수 있습니다. 자세한 내용은디지타이즈 작업 속성 을 참조하세요.

### 12.3.1 스냅 허용 오차 및 검색 반경 설정

QGIS 는 *Settings → Options… → Digitizing* 메뉴에서 편집 도구의 기본 습성을 환경 설정할 수 있는 파라미터를 여러 개 제공합니다. 자세한 내용은디지타이즈 작업 설정 을 참조하세요.

벡터 레이어 도형을 최적으로 정확하게 편집하려면, 피처 꼭짓점에 대해 적절한 스냅 허용 오차 및 검색 반경 값을 설정해줘야 합니다. *Snapping* 프레임에서 이런 스냅 허용 오차 및 검색 반경을 처리하는 관련 옵션들을 설정할 수 있습니다.

- **스냅 허용 오차 (Snapping tolerance)**: 사용자가 새 꼭짓점을 추가하거나 기존 꼭짓점을 이동시키는 경우, QGIS 는 사용자가 연결하고자 시도하는 가장 가까운 꼭짓점 또는 선분을 검색하는 데 스냅 허용 오차 거리를 사용합니다. 스냅 허용 오차 거리 안에 다른 꼭짓점 또는 선분이 없다면, QGIS 는 기존 꼭짓점 또는 선분에 스냅시키는 대신 사용자가 마우스 버튼을 놓은 위치에 꼭짓점을 배치할 것입니다.

  허용 오차 설정은 스냅 작업에 쓰이는 모든 도구에 영향을 미치며, 새 레이어 및 프로젝트에 기본적으로 적용됩니다. 하지만 레이어 수준에서 (스냅 작업 및 디지타이즈 작업 옵션 참조) 무시할 수 있습니다.

- **검색 반경 (Search radius)**: *Search radius for vertex edits* 는 사용자가 맵을 클릭할 때 QGIS 가 선택할 꼭짓점을 "검색"하기 위해 사용하는 거리입니다. 이 검색 반경 안에 다른 꼭짓점이 없을 경우, QGIS 는 편집 작업을 위해 어떤 꼭짓점도 찾아 선택하지 않을 것입니다.

스냅 허용 오차 및 검색 반경은 `map units` 또는 `pixels` 로 설정되기 때문에, 알맞은 설정값을 찾기 위해 시행착오를 거쳐야 할 수도 있습니다. 허용 오차를 너무 길게 설정하면, QGIS 가 틀린 꼭짓점에 스냅할 수도 있습니다. 특히 대량의 인접한 꼭짓점들을 다루고 있을 경우엔 말입니다. 검색 반경을 짧게 설정할수록, 사용자가 원하는 꼭짓점을 찾기 힘들어질 것입니다.

### 12.3.2 스냅 작업 및 디지타이즈 작업 옵션

*Project* ▸ *Snapping Options*…메뉴를 이용하면 프로젝트에서 전체 수준스냅 작업 및 디지타이즈 작업 옵션 을 (스냅 모드, 허용 오차 값, 단위 등등을) 무시할 수 있습니다. *Snapping and Digitizing Options* 에서 몇몇 다른 (스냅하는 레이어, 축척 제한, 위상 등등의) 속성도 환경 설정할 수 있습니다. *Snapping Toolbar* 에서 이런 기능들 대부분을 사용할 수 있습니다.

기본적으로 사용자가 Enable snapping 버튼 또는 S 키를 누를 때까지, 프로젝트에서 스냅 작업은 비활성 상태입니다. 이 툴바에서 스냅 모드, 허용 오차 값, 그리고 단위도 환경 설정할 수 있습니다.

#### 스냅 작업 속성

스냅을 활성화할 레이어 (들) 를 선택하는 데 다음 세 가지 옵션이 있습니다:

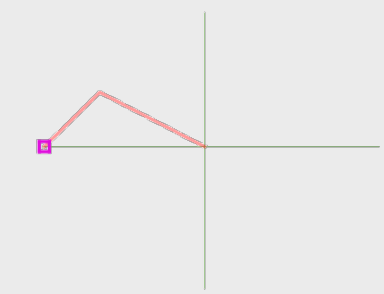
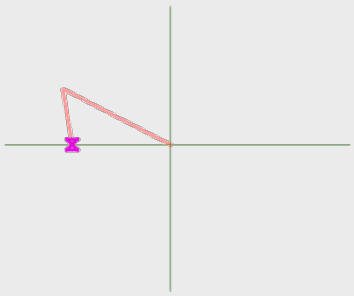
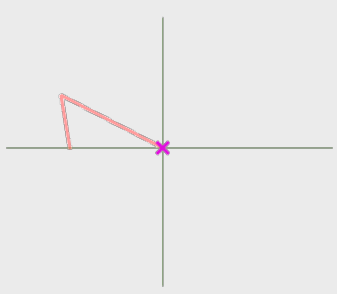
- *All layers*: 프로젝트에 있는 가시화된 모든 레이어를 선택합니다. 모든 꼭짓점 그리고/또는 선분에 포인터를 스냅시키는 빠르고 간단한 설정입니다. 대부분의 경우 이 스냅 모드를 사용하면 충분하지만, 수많은 벡터 레이어를 가진 프로젝트의 경우 사용에 주의를 요합니다. 성능에 영향을 줄 수도 있기 때문입니다.
- *Current layer*: 활성화된 레이어만 선택합니다. 편집 중인 레이어 내부의 위상 (topology) 의 일관성을 보장할 수 있는 편리한 방법입니다.
- *Advanced Configuration*: 레이어 별로 스냅 작업의 스냅 모드, 허용 오차 및 단위, 중첩 및 축척을 활성화시키고 조정할 수 있습니다. (그림 12.79 을 참조하세요.) 레이어를 편집하면서 꼭짓점을 또다른 레이어에 스냅해야 하는 경우, 대상 레이어가 체크되어 있는지 확인하고 스냅 허용 오차를 더 큰 값으로 늘리십시오. 스냅 작업 옵션 대화창에서 체크되어 있지 않은 레이어에는 스냅되지 않을 것입니다.

꼭짓점을 이동시키거나 생성할 때, 다음 스냅 작업 모드 가운데 하나를 선택할 수 있습니다:

- *Vertex*: 꼭짓점에 스냅합니다.
- *Segment*: 라인 또는 폴리곤 둘레를 따라 스냅합니다. 위상 편집 작업이 활성화된 경우, 스냅하는 위치에 새 꼭짓점을 추가합니다.
- *Area*: 스냅 포인트가 꼭 폴리곤의 경계 상만이 아니라 폴리곤의 영역 안에 들어가도록 보장합니다.
- *Centroid*: 피처 도형의 중심점에 스냅합니다. 다중부분 도형의 경우, 대상 포인트가 기존 부분들과 전혀 다를 수도 있습니다.
- *Middle of Segments*: 라인 또는 폴리곤 둘레의 중앙점에 스냅합니다.
- *Line Endpoints*: 라인 또는 폴리곤 둘레의 모든 부분의 시작 또는 끝 꼭짓점에 스냅합니다.

QGIS 는 스냅 의 유형에 따라 서로 다른 스냅 아이콘을 표시합니다:

표 12.2: 스냅 작업 아이콘

| | | |
|---|---|---|
|  |  |  |
| 꼭짓점에 스냅하기: 상자 아이콘 | 선분에 스냅하기: 모래시계 아이콘 | 교차점에 스냅하기: 가위표 아이콘 |

전체 수준 설정의 *Digitizing* 프레임에서 이 아이콘들의 색상을 변경할 수 있다는 사실을 기억하십시오.

허용 오차를 `map units` 또는 `pixels` 로 설정할 수 있습니다. `pixels` 로 설정하면, 맵 축척을 변경하더라도 일정하게 스냅할 수 있다는 장점이 있습니다. 일반적으로 10~12 픽셀이면 괜찮은 값이지만, 사용자 스크린의 해상도에 따라 달라집니다. 맵 단위로 설정하면 허용 오차를 실제 지표 거리와 비례하도록 만들 수 있습니다. 예를 들어 요소들 사이에 최소 거리를 두어야 한다면, 꼭짓점들을 서로 너무 가깝지 않게 추가하는 데 이 옵션이 유용할 것입니다.

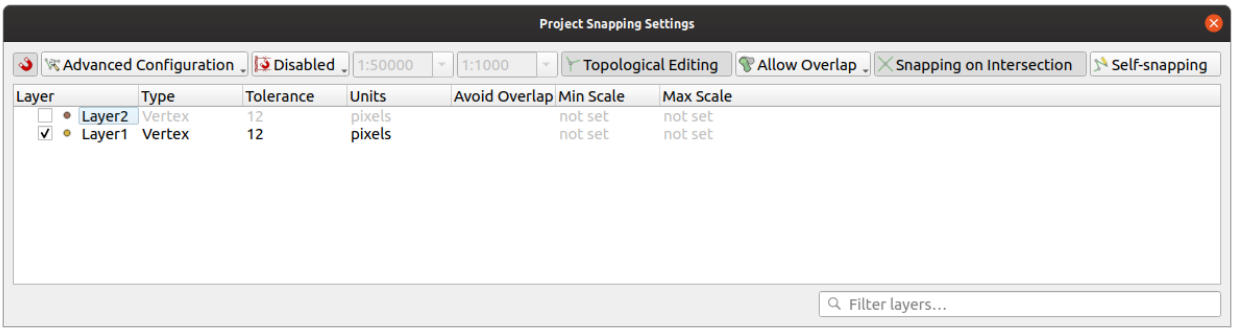



그림 12.79: 스냅 옵션 (고급설정 모드)

**참고**

기본적으로, 가시화된 (레이어 심볼이"No symbols"인 경우를 제외하고, 피처 스타일이 표시된) 피처에만 스냅할 수 있습니다. *Settings* ▸ *Options* ▸ *Digitizing* 탭에 있는 ☐ *Enable snapping on invisible features* 옵션을 체크하면 가시화되지 않은 피처에도 스냅할 수 있게 됩니다.

**팁**

**스냅 기본값 활성화**

*Settings* ▸ *Options* ▸ *Digitizing* 탭에서 모든 새 프로젝트가 기본적으로 스냅을 활성화시키도록 설정할 수 있습니다. 기본 스냅 모드, 허용 오차 값, 그리고 단위도 설정할 수 있습니다. 여기서 설정한 값들이 *Snapping Options* 대화창에 표시될 것입니다.

### 교차점 스냅 활성화

또다른 옵션은  *snapping on intersection* 을 선택하는 것입니다. 이 옵션은 스냅이 활성화된 레이어의 도형 교차점에 — 설령 교차점에 꼭짓점이 없는 경우라도 — 스냅할 수 있게 해줍니다.

### 축척 범위에 따라 스냅 제한하기

몇몇 경우 스냅 작업이 매우 느려질 수도 있습니다. 일부 레이어에 있는 피처의 용량 때문에 계산하고 유지해야 하는 인덱스가 많아지기 때문입니다. 이를 위해 맵 뷰가 적절한 축척 범위일 경우에만 스냅 작업을 활성화시키는 몇몇 파라미터가 존재합니다. 이 파라미터들을 설정하면 그리기 작업에 적절한 축척인 경우에만, 리소스를 많이 소비하더라도 스냅 작업에 관련된 인덱스 계산을 하도록 할 수 있습니다.

*Project* ▸ *Snapping Options*…메뉴에서 스냅 작업 축척 제한을 환경 설정합니다. 축척에 따른 스냅 작업 제한은 *Advanced Configuration* 모드에서만 사용할 수 있습니다.

축척 범위에 따라 스냅 작업을 제한하는 데 사용할 수 있는 모드는 세 가지가 있습니다:

- *Disabled*: 현재 맵 축척이 어떻게 되든 스냅 작업을 활성화합니다. 기본 모드입니다.
- *Global*: 스냅 작업이 제한되며 현재 맵 축척이 전체 수준 최저와 전체 수준 최고 값 사이인 경우에만 활성화됩니다. 이 모드를 선택할 경우 스냅 작업이 활성화되는 축척 범위를 환경 설정할 수 있는 두 위젯을 사용할 수 있습니다.
- *Per layer*: 각 레이어에 대해 스냅 작업 축척 범위 제한을 정의합니다. 이 모드를 선택할 경우 각 레이어에 대한 최저 및 최고 축척을 환경 설정할 수 있는 두 열을 사용할 수 있습니다.

이때 최저 및 최고 축척은 QGIS 의 관습을 따른다는 점을 기억하십시오: 최저 축척이란 가장"축소한"축척이며 최고 축척은 가장"확대한"축척입니다. "0"또는"설정하지 않음"으로 설정한 최저 및 최고 축척은 제한이 없는 것으로 간주합니다.

### 자체 스냅

 Self-snapping 옵션을 이용하면 편집 중인 도형에 스냅할 수 있습니다. 이 옵션을고급 디지타이즈 작업 패널 과 함께 사용하는 경우, 이전 경계 또는 꼭짓점을 기준으로 새 경계를 쉽게 디지타이즈할 수 있습니다. 자체 스냅 작업은 무결하지 않은 도형을 생성할 수도 있으므로 조심해서 사용하십시오.

### 사용자 지정 그리드 상에서 스냅하기

레이어 속성 대화창의 *Digitizing* 탭에서 레이어 별로 스냅 거리를 사용자 지정할 수도 있습니다. *Geometry precision* 거리를 설정하면, 맵 캔버스의 축척이 가시 축척에 근접한 경우 점으로 이루어진 그리드가 나타납니다. 이때 그리드의 점에 스냅할 수 있습니다. 도형을 추가 또는 수정하면 해당 도형의 모든 꼭짓점이 자동으로 가장 가까운 그리드의 노드에 스냅될 것입니다. 더 자세한 정보는디지타이즈 작업 속성 을 참조하세요.

### 12.3.3 위상 편집

이런 스냅 옵션 이외에도, *Snapping options*…대화창 (*Project* ▸ *Snapping options*) 및 *Snapping* 툴바에서 몇몇 다른 위상 기능들을 활성화/비활성화할 수 있습니다.

#### 위상 편집 활성화

 Topological editing 버튼은 공통 경계선을 공유하는 피처를 편집하고 유지보수할 수 있게 해줍니다. 이 옵션을 활성화하면, QGIS 가 피처들이 공유하는 경계선을'탐지'합니다. 공통 꼭짓점/선분을 이동시키면, QGIS 가 인접 피처 도형의 꼭짓점/선분도 이동시킬 것입니다.

위상 편집 작업은 서로 다른 레이어들의 피처들과 작동합니다. 해당 레이어들이 가시화되어 있고 편집 모드라면 말이죠.

Z 또는 M 값을 가진 레이어에서 위상 편집 작업은 연결에 쓰인 경계의 값을 기반으로 꼭짓점의 Z 또는 M 값을 보간할 것입니다.

#### 중첩 제어

중첩을 이용하면 선택한 레이어에서 기존 피처와 중첩하는 피처를 새로 그릴 수 없지만, 인접한 폴리곤을 디지타이즈하는 시간을 줄여주기도 합니다. 중첩 도구를 통해 이를 제어할 수 있습니다. 다음 세 가지 모드를 선택할 수 있습니다:

1.  *Allow Overlap* (기본값): 중첩을 허용합니다.
2.  *Avoid Overlap on Active Layer*: 편집 중인 레이어의 다른 피처와 중첩하지 못 하게 막습니다. 이웃 피처와 중첩하는 도형을 새로 디지타이즈하는 경우 QGIS 가 새 도형의 중첩 부분 (들) 을 잘라내고 기존 피처의 경계선에 스냅시킬 것입니다. 사용자가 경계선 상에 있는 공통 꼭짓점들을 디지타이즈하지 않아도 된다는 장점이 있습니다.
3.  *Follow Advanced Configuration*: *Advanced configuration* 뷰 모드에서 레이어 별로 중첩 설정을 할 수 있습니다.

> **참고**
>
> 기존 도형이 새 도형을 완전히 덮는 경우, 새 도형은 제거되며 QGIS 가 오류 메시지를 띄울 것입니다.

> **경고**
>
> ***Avoid overlap* 옵션을 조심해서 사용해야 합니다**
>
> 이 옵션은 모든 폴리곤 레이어에서 중첩하는 새 도형을 자르기 때문에, 더 이상 이 옵션이 필요하지 않게 되자마자 반드시 체크 해제해야 합니다. 그렇지 않을 경우 기대하지 않은 도형을 얻게 될 수 있습니다.

### 자동 투사

보통, 캡처 작업 맵 도구 (피처 추가, 부분 추가, 고리 추가, 재형성 및 분할) 이용시 사용자가 피처의 각 꼭짓점을 클릭해야 합니다. 자동 투사 (automatic tracing) 모드를 사용하면 디지타이즈 작업 과정의 속도를 향상시킬 수 있습니다. 디지타이즈 작업 도중 모든 꼭짓점을 직접 배치해야 할 필요가 없어지기 때문이죠:

1. (*Snapping* 툴바에 있는)  아이콘을 클릭하거나 `T` 키를 눌러  Tracing 도구를 활성화하십시오.
2. 사용자가 투사하고자 하는 피처의 꼭짓점이나 선분에스냅 하십시오.
3. 스냅하고 싶은 또다른 꼭짓점 또는 선분으로 마우스를 가져가면 일반적인 직선이 아니라 디지타이즈 작업 고무줄이 마지막으로 스냅한 포인트에서 현재 위치까지의 경로를 나타냅니다. 이 도구는 만곡 도형에서도 작동합니다.

   QGIS 는 사실 기저 피처 위상을 이용해서 두 포인트 사이의 최단 경로를 생성합니다. 투사 기능을 이용하려면 경로를 생성하기 위해 투사가 가능한 레이어의 스냅 기능이 활성화된 상태여야 합니다. 또 디지타이즈 작업 중 기존 꼭짓점 또는 선분에 스냅해야 하며, 두 노드가 기존 피처 경계를 통해 위상적으로 연결될 수 있는지도 확실히 해야 합니다. 그렇지 않으면 QGIS 가 두 노드를 연결할 수 없으므로 단일 직선을 투사하게 됩니다.
4. 클릭하면 QGIS 가 표시된 경로를 따라 중간 단계 꼭짓점들을 배치합니다.

피처 경계를 따라 투사하는 것이 아니라 피처 경계와 평행하는 경로를 디지타이즈하려면  Enable Tracing 아이콘을 펼쳐 *Offset* 옵션을 설정하십시오. 양의 값은 투사 방향의 왼쪽으로 새로 그리도록 이동시키고, 음의 값은 그 반대로 투사 방향의 오른쪽으로 새로 그리게 이동시킵니다.

**참고**

**최적의 투사를 위해 맵 축척 또는 스냅 설정을 조정하기**

맵 캔버스에 너무 많은 피처들이 있는 경우, 잠재적으로 너무 긴 투사 구조 준비 및 대용량 메모리 부담을 피하기 위해 투사를 비활성화시킵니다. 맵을 확대하거나 일부 레이어를 비활성화하면 다시 투사를 활성화시킵니다.

**참고**

**위상 포인트를 추가하지 않습니다**

이 도구는 *Topological editing* 옵션을 활성화한 상태이더라도 기존 폴리곤 도형에 포인트를 추가하지 않습니다. 편집 중인 레이어에 도형 정밀도를 활성화한 경우라면, 새로 그린 도형이 기존 도형과 정확하게 일치하지 않을 수도 있습니다.

**팁**

**`T` 키를 눌러 자동 투사를 빠르게 활성화/비활성화하기**

`T` 키를 누르면, (피처를 디지타이즈하는 도중이라도) 투사를 활성화/비활성화할 수 있습니다. 따라서 투사를 활성화한 상태로 피처의 일부분을 디지타이즈하고 다른 부분은 비활성화한 상태로 디지타이즈할 수도 있습니다. 투사를 비활성화해도 도구들은 아무 일 없이 작동합니다.

**팁**

**만곡 도형으로 투사 변환**

*Settings* ▸ *Options* ▸ *Digitizing* ▸ *Tracing* 메뉴를 사용하면 디지타이즈 작업 도중 만곡 도형을 생성할 수 있습니다. 디지타이즈 작업 옵션 을 참조하세요.

### 12.3.4 기존 레이어 디지타이즈 작업

기본적으로, QGIS 는 레이어를 읽기 전용으로 불러옵니다. 이 습성은 혹시라도 마우스가 미끄러져 레이어를 실수로 편집하는 일을 피하기 위한 예방책입니다. 하지만, 편집을 지원하는 데이터 제공자이며 (데이터 포맷 및 필드 탐구 참조) 기저 데이터소스가 쓰기 가능인 경우 (예: 파일이 읽기 전용이 아닌 경우) 모든 레이어를 편집할 수 있습니다.

> **팁**
>
> **프로젝트 내부에서 레이어 편집 권한 제한**
>
> *Project* ▸ *Properties…* ▸ *Data Sources* ▸ *Layers Capabilities* 테이블에서, 제공자의 권한 허가 여부에 상관없이 어떤 레이어든 읽기 전용으로 설정할 수 있습니다. 이것은 다중 사용자 환경에서 승인받지 않은 사용자가 실수로 레이어를 (예: 셰이프파일을) 편집해서 혹시라도 데이터를 변질시키는 일을 막기 위한 간편한 방법이 될 수 있습니다. 이 설정은 현재 프로젝트 안에서만 적용된다는 사실을 기억하십시오.

일반적으로, 벡터 레이어를 편집하기 위한 도구는고급 디지타이즈 작업 에서 설명하는 대로 디지타이즈 작업과 고급 디지타이즈 작업 툴바로 나뉘어져 있습니다. *View* ▸ *Toolbars* ▸ 메뉴에서 둘 다 선택하고 선택 해제할 수 있습니다.

기본 디지타이즈 작업 도구를 사용하면, 다음 기능들을 수행할 수 있습니다:

표 12.3: 벡터 레이어 기본 편집 툴바

| 도구 | 목적 | 도구 | 목적 |
|---|---|---|---|
| | 모든 레이어 또는 선택한 레이어에서 변경 사항을 동시에 저장, 되돌리기, 또는 취소 | | 활성 레이어 상태를 기반으로 선택한 레이어 (들) 의 편집 상태를 켜고 끄기 |
| | 활성 레이어에 편집 내용을 저장 | | |
| | 직선을 이용해서 디지타이즈 | | 곡선을 이용해서 디지타이즈 |
| | 자유로운 (freehand) 디지타이즈 활성화 | | 정다각형 폴리곤을 디지타이즈 |
| | 신규레코드 추가 | | 피처 추가: 포인트 캡처 |
| | 피처 추가: 라인 캡처 | | 피처 추가: 폴리곤 캡처 |
| | 꼭짓점 도구 (모든 레이어) | | 꼭짓점 도구 (현재 레이어) |
| *Show Vertex Editor* | 꼭짓점 편집기 패널을 자동으로 열지 여부를 설정 | | 선택한 모든 피처의 속성을 동시에 수정 |
| | 활성 레이어에서 선택한 피처를 삭제 | | 활성 레이어에서 피처를 잘라내기 |
| | 활성 레이어에서 선택한 피처를 복사 | | 활성 레이어에 피처를 붙여넣기 |
| | 활성 레이어에서 변경 사항을 되돌리기 | | 활성 레이어에서 변경 사항을 다시 실행 |

이 모든 디지타이즈 작업 도구를 쓰는 도중에 도구를 비활성화시키지 않고서도 맵 캔버스에서맵 뷰 탐구하기 할 수 있다는 사실을 기억하십시오.

모든 편집 작업 세션은 지정 레이어의 컨텍스트 메뉴, 속성 테이블 대화창, 디지타이즈 작업 툴바 또는 menuselection:*Layer* 메뉴에 있는 Toggle editing 옵션을 선택해야 시작합니다.

레이어를 편집 모드로 전환하면, 편집 작업 툴바에 있는 부가 도구 아이콘들이 활성화되며, *Settings* ▸ *Options…* ▸ *Digitizing* 탭에 있는 *Show markers only for selected features* 옵션을 체크하지 않은 경우 모든 피처의 꼭짓점에 마커가 나타날 것입니다.

> **팁**
>
> **정기적으로 저장**
>
> Save Layer Edits 아이콘으로 주기적으로 편집 내용을 저장하는 것을 잊지 마십시오. 이 아이콘을 클릭하면 사용자의 데이터소스가 모든 수정 사항을 받아들일 수 있는지도 확인할 것입니다.

### 도형 편집 기법

라인 또는 폴리곤 기반 레이어에 (주로 피처를 추가하고 분할하고 형태를 변경하는) 도형 그리기 도구를 활성화하면 새 꼭짓점을 추가하기 위한 기법을 선택할 수 있습니다:

- Digitize with Segment: 시작 포인트와 종단 포인트를 왼쪽 클릭해서 직선을 그립니다.
- Digitize with Curve: 연속하는 3 개의 (시작, 원호 위, 종단) 노드를 왼쪽 클릭으로 정의해서 곡선을 그립니다. 도형 유형이 곡선을 지원하지 않는 경우, 연속하는 짧은 선분들을 이용해서 곡률의 근사치를 표현합니다.
- Stream Digitizing: 자유 (freehand) 모드에서–예를 들어 맵 캔버스에서 커서의 이동 궤적 및 *Streaming Tolerance* 를 따라 노드들을 추가해서 – 라인을 그립니다. 스트리밍 허용 오차는 연속하는 두 꼭짓점 사이의 간격을 정의합니다. 현재 지원하는 단위는 픽셀 (px) 뿐입니다. 이 모드에서는 왼쪽 클릭으로 시작 포인트를, 그리고 오른쪽 클릭으로 종단 포인트를 정의하기만 하면 됩니다.
- Digitize Shape: 정다각형 폴리곤을 그리기 위해형태 디지타이즈 작업 툴바 에 있는 도구를 실행합니다.

디지타이즈 작업 모드를 바꿔도 선택한 기법은 유지됩니다. 한 도형을 그리는 동안 처음 세 모드 가운데 어느 모드든 조합할 수 있습니다.

### 피처 추가하기

레이어 유형에 따라, 툴바에 있는 Add Record, Add Point Feature, Add Line Feature 또는 Add Polygon Feature 아이콘을 통해 현재 레이어에 새 피처를 추가할 수 있습니다.

무 (☐) 도형 (geometryless) 피처를 추가하려면, Add Record 아이콘을 클릭해서 열리는 피처 양식에 속성을 입력하면 됩니다.

공간적으로 활성화된 도구로 피처를 생성하려면, 먼저 도형을 디지타이즈한 다음 그 속성을 입력하십시오. 도형을 디지타이즈하려면:

1. Digitize With Segment: (기본값인 동시에 선택적인) 직선 디지타이즈 도형 그리기 방식을 선택하십시오.
2. 맵 영역 위를 왼쪽 클릭해서 새 피처의 첫 번째 포인트를 생성하십시오. 포인트 피처의 경우 이것만으로도 충분하며, 필요한 경우 그 속성을 입력할 수 있는 피처 양식을 열 것입니다.

3. 라인 또는 폴리곤 도형의 경우, 캡처하고자 하는 추가 포인트 각각을 계속 왼쪽 클릭해나가십시오. 피처에 스냅 옵션, 그리드에 스냅 또는고급 디지타이즈 작업 패널을 이용해서 각 꼭짓점을 정확하게 배치할 수 있습니다.

   사용자가 노드들을 하나씩 클릭해서 그 사이에 직선을 그리는 도중, 라인 및 폴리곤을:

   - 자동으로 추적 해서 디지타이즈 작업의 속도를 올릴 수 있습니다. 사용자가 기존 피처를 따라 배치하는 꼭짓점들 사이에 연속되는 직선들을 생성할 것입니다.
   -  Stream Digitizing 또는 `R`: 사용자가 직접 자유롭게 디지타이즈할 수 있습니다.
   - `Ctrl+Shift+G` 또는  Digitize with Curve: 곡선을 그릴 수 있습니다.

   **참고**

   라인 또는 폴리곤 도형을 디지타이즈하는 동안, 도형 그리기 메소드 사이를 자유롭게 전환할 수 있습니다. 즉 직선, 자유로운 선, 그리고 곡선 부분이 섞인 피처를 생성할 수 있습니다.

4. 사용자가 잘못 추가했을 수도 있는 마지막 노드 (들) 를 되돌리려면 `Delete` 또는 `Backspace` 키를 누르십시오.
5. 포인트 추가 작업을 완료했다면, 맵 영역 어디든 오른쪽 클릭해서 해당 피처의 도형 입력 작업이 끝났다고 확인하십시오.

   **팁**

   **디지타이즈 작업 고무줄을 사용자 지정하기**

   폴리곤을 캡처하는 도중에 기본값인 빨강색 고무줄이 아래에 있는 — 사용자가 포인트를 캡처하고자 하는 — 피처 또는 장소를 가릴 수가 있습니다. *Settings* ▸ *Options* ▸ *Digitizing* 대화창에서 고무줄의 *Fill Color* 에 대한 투명도 (또는 알파 채널) 를 높이면 이 문제를 해결할 수 있습니다. 또는 *Don't update rubber band during node editing* 옵션을 체크해서 고무줄을 사용하지 않을 수도 있습니다.

6. 라인 피처의 경우 `Shift` 키를 누른 채 오른쪽 클릭하면 자동으로 라인을 닫을 것입니다.
7. 새 피처에 대한 정보를 입력할 수 있는 속성창이 나타날 것입니다. 그림 12.80 이 가상의 새로운 강에 대한 속성을 설정하는 내용입니다. 하지만 *Settings* ▸ *Options* 메뉴 아래의 *Digitizing* 메뉴에서 다음 작업을 할 수 있습니다:

   - *Suppress attributes pop-up windows after each created feature*: 속성 입력 양식 대화창을 열지 않도록 합니다.
   - *Reuse last entered attribute values*: 속성 입력 양식 대화창이 열릴 때 가장 최근에 입력한 속성값으로 자동 입력해서 변경할 값만 입력하면 되도록 합니다.

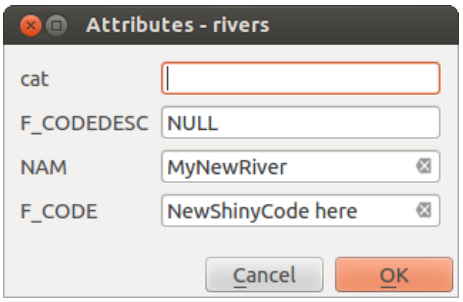



그림 12.80: 새 벡터 피처를 디지타이즈한 후의 속성 입력 양식 대화창

### 꼭짓점 도구

QGIS 는 벡터 피처 꼭짓점과 대화형 작업을 할 수 있는 도구를 2 개 제공합니다:

- Vertex Tool (Current Layer): (*Layers* 패널에서) 활성화된 레이어에 있는 피처에만 영향을 미칩니다.
- Vertex Tool (All Layers): 편집 가능한 레이어에 있는 모든 피처에 영향을 미칩니다. 활성 레이어를 전환하지 않고 피처를 편집하거나, 또는 여러 레이어를 (예를 들어 국경 및 각 국가의 지역 경계선을) 동시에 편집할 수 있습니다.

편집 가능한 모든 벡터 레이어에 대해, 꼭짓점 도구는 CAD 프로그램과 비슷하게 피처 꼭짓점을 조작할 수 있는 기능을 제공합니다. 여러 꼭짓점들을 동시에 선택해서 한 번에 이동, 추가, 또는 삭제시킬 수 있습니다. 꼭짓점 도구는 위상 편집 기능도 지원합니다. 꼭짓점 도구는 선택 집합을 유지하기 때문에, 어떤 작업이 끝나더라도 해당 피처 및 도구에 대한 선택 집합은 활성화 상태를 유지합니다.

> **팁**
>
> **Drawing a series of new vertices**
>
> The vertex tool does not support *automatic tracing* and is optimized for editing individual vertices and moving or deleting an arbitrary selection of multiple vertices. Try using the Reshape Features tool when you need to replace or insert a series of new vertices, especially if you want to trace other features.

*Settings* → *Options* → *Digitizing* → *Search Radius:* 1,00 검색 반경 속성을 0 보다 큰 숫자로 설정하는 것이 중요합니다. 그러지 않을 경우, QGIS 가 현재 어떤 꼭짓점을 편집 중인지 구분하지 못 하고 경고 메시지를 표시할 것입니다.

> **팁**
>
> **꼭짓점 마커**
>
> QGIS 는'반투명한 원 (Semi-transparent circle)', '십자표 (Cross)'및'없음 (None)'유형의 꼭짓점 마커를 지원합니다. 마커 스타일을 변경하려면 *Settings* 메뉴에서 *Options* 를 선택한 다음 *Digitizing* 탭으로 가서 적절한 항목을 선택하십시오.

### 기본 작업

레이어가 편집 모드 상태라고 가정하고, 꼭짓점 도구를 활성화하는 것으로 시작합니다. 꼭짓점 위에 커서를 가져가면 빨간색 원이 나타날 것입니다.

- **꼭짓점 선택**: 꼭짓점을 다음과 같은 방법으로 선택할 수 있습니다:
  - `Shift` 키를 누른 채 꼭짓점을 하나씩 클릭하기
  - 대상 꼭짓점을 둘러싸는 직사각형을 클릭 & 드래그하기
  - 대상 꼭짓점을 둘러싸는 폴리곤을 그리기: 꼭짓점 도구를 이용해서 `Alt` 키를 누른 채 폴리곤을 디지타이즈하기 시작하십시오. 이어지는 각 클릭마다 고무줄 폴리곤에 새 꼭짓점을 추가합니다. `Backspace` 또는 `Delete` 키를 누르면 마지막에 추가한 고무줄 꼭짓점을 제거합니다. `Esc` 키를 누르면 고무줄의 꼭짓점들을 모두 백스페이스/삭제하는 것과 마찬가지로 폴리곤 선택 모드를 취소합니다. 오른쪽 클릭하면 폴리곤 디지타이즈 작업을 완료하고 고무줄 폴리곤 내에 있는 모든 꼭짓점을 선택합니다.

  꼭짓점을 선택하면, 꼭짓점 색상이 파란색으로 바뀝니다. 현재 선택 집합에 더 많은 꼭짓점을 추가하려면, `Shift` 키를 누른 채 앞에서 설명한 대로 진행하십시오. 선택 집합에서 꼭짓점을 제거하려면 `Ctrl` 키를 누른 채 제거할 꼭짓점을 선택하십시오.

  > **팁**
  >
  > **피처 선택에 속박된 꼭짓점 도구**
  >
  > 서로 다른 피처에 (또는 레이어에) 걸쳐 있는 꼭짓점들을 동시에 선택할 수 있습니다. 피처들이 많이 모여 있는 위치에 있는 특정 피처의 꼭짓점을 찾고 있다면 먼저 해당 피처를 선택하십시오. 그 다음 꼭짓점 도구를 이용해서 꼭짓점 주위에 직사각형 또는 폴리곤 선택기를 그리십시오. 선택한 피처의 꼭짓점만 선택될 것입니다.
  >
  > 꼭짓점 편집기 패널에 해당 피처를 표시한 경우도 마찬가지입니다.

- **배치 (Batch) 꼭짓점 선택 모드**: `Shift+R` 조합키를 누르면 배치 선택 모드를 활성화시킬 수 있습니다. 한 번 클릭해서 첫 번째 꼭짓점을 선택한 다음, 또다른 꼭짓점에 마우스를 **클릭하지 말고** 가져가십시오. 이렇게 하면 두 꼭짓점 사이의 모든 꼭짓점을 (폴리곤의 경우) 최단 경로를 사용해서 동적으로 선택할 것입니다.

  `Ctrl` 키를 누르면 피처 경계선을 따라 최장 경로를 선택해서 선택을 반전할 것입니다. 두 번째 클릭으로 노드 선택을 완료할 수 있고, 또는 `Esc` 키를 누르면 배치 모드에서 벗어나게 됩니다.

- **꼭짓점 추가**: 라인 또는 폴리곤 도형에 꼭짓점을 추가하려면, `Shift` 키를 누른 채 선분 위에 있는 위치를 더블클릭하십시오.

  선분 위에 커서를 가져가면 선분 중앙에 가상의 새 노드가 나타납니다. 이 노드를 클릭한 다음, 커서를 새로운 위치로 가져가서 다시 클릭하면 새 꼭짓점을 추가합니다. 라인 도형의 경우 가상 노드가 양극단에도 나타납니다: 이 노드를 클릭하고, 이어지는 클릭을 수행한 다음 오른쪽 클릭으로 완료하십시오. 기존 라인을 쉽게 연장할 수 있는 방법입니다.

- **꼭짓점 삭제하기**: 꼭짓점을 선택한 다음 `Delete` 키를 누르십시오. 피처의 모든 꼭짓점을 삭제하면 – 데이터소스와 호환되는 경우–도형이 없는 피처를 생성합니다. 피처를 완전히 삭제하는 것이 아니라 도형 부분만 삭제한다는 점을 기억하십시오. 피처를 완전히 삭제하려면  Delete Selected 도구를 사용해야 합니다.

- **꼭짓점 이동**: 이동시키고자 하는 꼭짓점을 모두 선택하고, 선택한 꼭짓점 또는 경계를 하나 클릭한 다음, 원하는 새 위치를 클릭하십시오. 두 번째 클릭 전에피처에 스냅 케이퍼빌리티 및 거리, 각도, 정확한 X 및 Y 위치를 제약하는고급 디지타이즈 작업 패널 을 이용할 수 있습니다. 선택한 모든 꼭짓점이 전이될 것입니다.

  하지만그리드에 스냅 옵션이 활성화되어 있다면 선택한 꼭짓점이 전이된 위치에 가장 가까운 그리드 교차점에 스냅됩니다. 선택하지 않은 꼭짓점들도 가장 가까운 그리드 교차점으로 이동합니다. 이 옵션이 활성화된 경우 단순 이전을 할 수 없습니다.

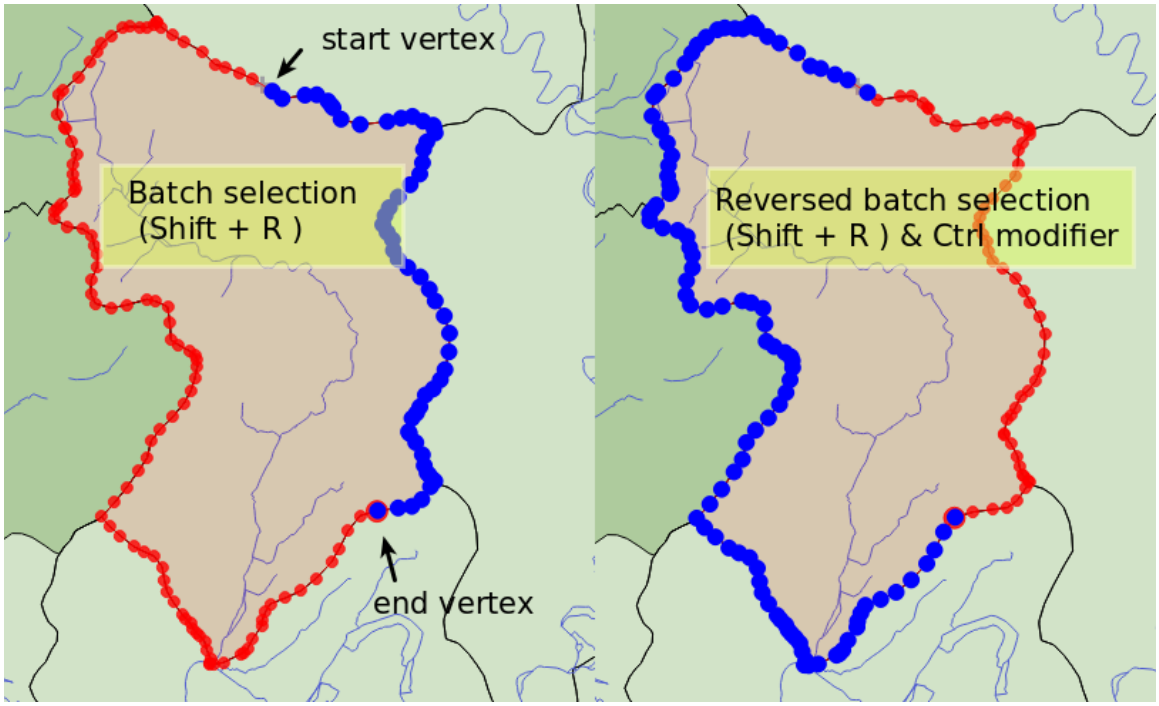



그림 12.81: `Shift+R` 로 배치 꼭짓점 선택하기

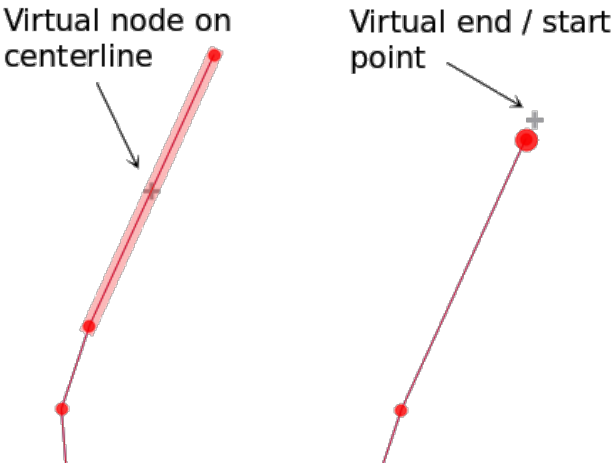



그림 12.82: 꼭짓점을 추가하기 위한 가상 노드

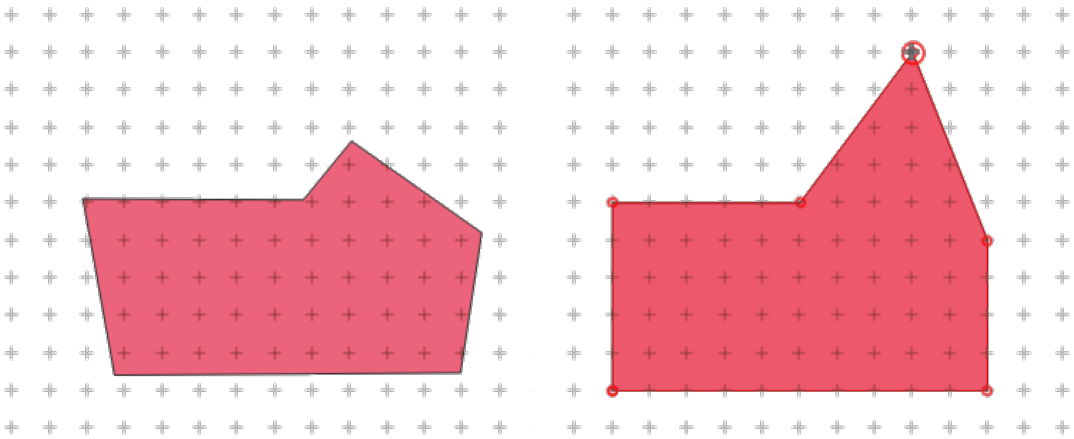


그림 12.83: 제일 위에 있는 꼭짓점을 이동시키면 그리드의 모든 꼭짓점들에 스냅됩니다

- **인접 선분을 곡선으로/곡선을 인접 선분으로 변환하기**: 변환하고자 하는 선분의 중앙 꼭짓점을 선택한 다음 O 키를 누르십시오. 꼭짓점이 곡선에 속해 있다면 곡선을 직선으로 변환할 것입니다. 꼭짓점이 두 개의 직선 사이에 있다면, 해당 직선 2 개를 곡선으로 변환할 것입니다. 라인의 첫 번째 또는 마지막 꼭짓점은 곡선의 중앙 꼭짓점으로 변환될 수 없습니다. 이 도구를 사용하려면 레이어가 만곡 도형 유형과 호환되어야만 합니다.

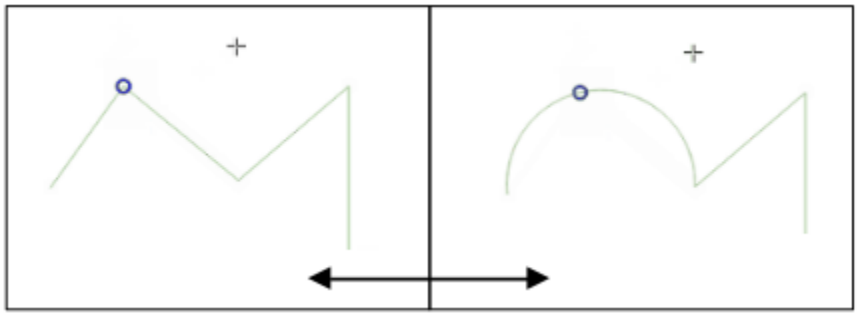


그림 12.84: O 문자 키로 곡선을 직선으로/직선을 곡선으로 변환하기

꼭짓점 도구로 변경한 사항 하나하나가 *Undo* 대화창에 개별 항목으로 저장됩니다. 이 기능이 활성화된 경우 모든 작업이 위상 편집 작업을 지원한다는 사실을 기억하십시오. 실시간 (on-the-fly) 투영도 지원합니다.

### 꼭짓점 편집기 패널

꼭짓점 도구를 활성화하면 *Vertex Editor* 패널도 열립니다. 피처를 오른쪽 클릭하면 패널에 해당 피처의 모든 꼭짓점 목록과 함께 꼭짓점의 *x*, *y*, (그리고 해당하는 경우 *z*, *m*) 그리고 (원호 도형인 경우 반경을 나타내는) *r* 을 채웁니다. 또한 이 피처를 편집 전용으로 지정합니다. 다시 말해 다른 피처의 편집을 비활성화시킨다는 의미입니다:

- 테이블에 있는 행을 하나 선택하면 맵 캔버스에서 대응하는 꼭짓점이 선택됩니다. 그 반대도 마찬가지입니다.
- 맵 캔버스 위를 클릭하거나 드래그하면 선택한 피처의 꼭짓점 및 선분만 선택하거나 이동시킬 것입니다.
- 테이블에서 좌표를 변경하면 꼭짓점 위치를 업데이트합니다. 꼭짓점의 Z 좌표 또는 M 값을 편리하게 편집할 수 있는 방법입니다.
- 행 여러 개를 선택한 다음 한번에 삭제할 수도 있습니다.
- 선택한 피처에만 새 꼭짓점을 추가할 수 있습니다.

사용자가 꼭짓점 도구와 쌍방향 작업을 할 때마다 (다른 패널을 가리거나 패널 배치를 방해할 수도 있기 때문에) *Vertex Editor* 패널에 즉시 반영되지 않기를 바란다면, 패널 상단에 있는 ☰ Options 메뉴의 *Auto-open table* 항목을 체크 해제하십시오. 그 다음 패널을 닫아도 됩니다. 패널을 다시 열려면, 패널 또는 툴바를 오른쪽 클릭하고 목록에서 꼭짓점 편집기 패널을 선택하거나, 또는 *Digitizing toolbar* 메뉴의 *Show vertex editor* 항목을 체크하면 됩니다.

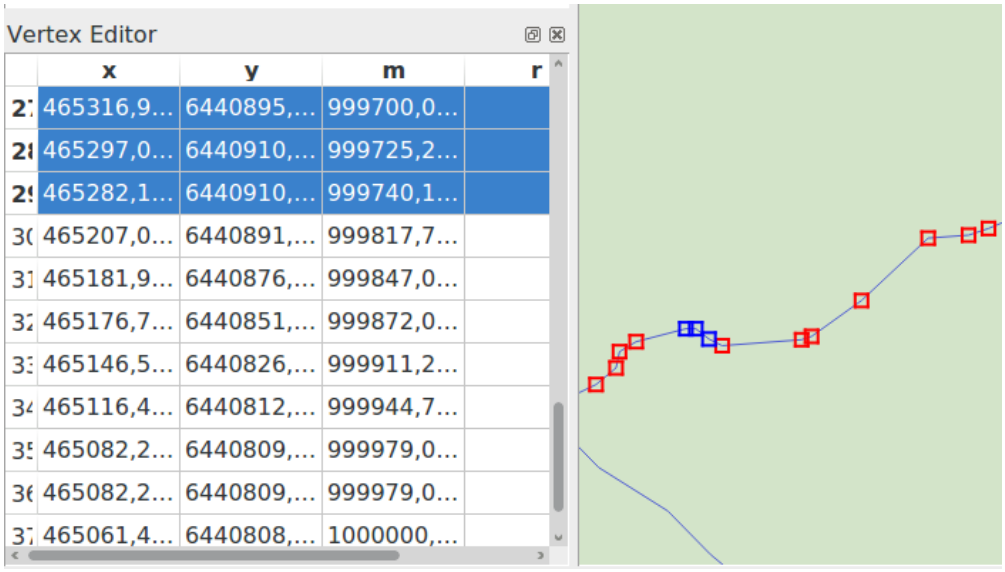



그림 12.85: 선택한 노드를 보여주는 꼭짓점 편집기 패널

### Z 좌표 또는 M 값 할당 규칙

3 차원 벡터 피처 또는 M 값을 가진 피처를 디지타이즈하는 작업은 (X, Y) 2 차원 레이어를 디지타이즈하는 작업과 크게 다르지 않습니다. 이 장에서 설명하는 도구 및 옵션을 그대로 사용할 수 있어 평면 환경에서 꼭짓점 또는 포인트를 배치할 수 있게 해줍니다. 그 다음 Z 좌표 (또는 M 값) 할당을 처리해야 할 수도 있습니다:

- QGIS 는 기본적으로 새 꼭짓점에 *Settings* ▸ *Options* ▸ *Digitizing* 탭에서 설정한 *Default Z value* (또는 *Default M value*) 를 할당할 것입니다. 고급 디지타이즈 작업 패널 을 사용하는 경우, *z* (또는 *m*) 위젯으로부터 값을 가져옵니다.
- 꼭짓점에 스냅할 때, 새 꼭짓점 또는 이동시킨 꼭짓점은 스냅된 꼭짓점의 Z 좌표 또는 M 값을 가집니다.
- 위상 편집 모드에서 선분에 스냅하면, 선분을 따라 새 꼭짓점의 Z 좌표 또는 M 값이 보간됩니다.
- *Advanced Digitizing Panel* 의 *z* (또는 *m*) 위젯이 🔒 잠겨 있는 경우, 이 위젯의 값이 꼭짓점에 적용되며 어떤 스냅된 꼭짓점 또는 선분의 Z 좌표 또는 M 값에도 우선합니다.

기존 피처의 Z 좌표 또는 M 값을 편집하려면꼭짓점 편집기 패널 을 이용하면 됩니다. 사용자 지정 Z 좌표 또는 M 값을 가진 피처를 생성하려면 *Advanced Digitizing Panel* 을 이용하는 편이 좋을 것입니다.

### 피처 잘라내기, 복사 및 붙여넣기

먼저  Toggle editing 을 통해 대상 레이어를 편집 모드로 설정해놓았다면, 동일한 QGIS 프로젝트의 레이어들 사이에서 선택한 피처를 자르고, 복사하고 붙여넣을 수 있습니다.

> **팁**
>
> **복사/붙여넣기를 이용해서 폴리곤을 라인으로, 라인을 폴리곤으로 변형하기**
>
> 라인 피처를 복사해서 폴리곤 레이어에 붙여넣는 경우, QGIS 가 대상 레이어에 그 경계가 라인 피처의 닫힌 도형에 대응하는 폴리곤으로 붙여넣습니다. 이는 동일한 데이터를 가진 서로 다른 도형 유형을 빠르게 생성하는 방법입니다.

외부 응용 프로그램에 피처를 텍스트로 붙여넣을 수도 있습니다. 피처가 도형 데이터를 OGC WKT(Well-Known Text) 서식으로 표현한 CSV 포맷인 경우에 말이죠. QGIS 외부의 WKT 및 GeoJSON 객체를 QGIS 내부의 레이어로 붙여넣는 것도 가능합니다.

복사/붙여넣기 기능이 유용한 경우란 어떤 경우일까요? 바로 한 번에 하나 이상의 레이어를 편집하는 중에 레이어들 간에 피처를 복사/붙여넣기하는 경우입니다. 어째서 이런 작업이 필요한 걸까요? QGIS 예시 데이터의 `big_lakes` 레이어에 있는 호수 5,000 개 전부가 아니라 한두 개만 있는 레이어에서 몇몇 작업을 해야 한다고 가정해봅시다. 새 레이어를 생성한 다음 해당 레이어에 필요한 호수만 복사/붙여넣기를 이용해서 넣을 수 있습니다.

새 레이어에 몇몇 호수를 복사해 넣는 다음 예제를 해보겠습니다:

1. 복사해오려는 레이어 (원본 레이어) 를 불러옵니다.
2. 복사해 넣으려는 레이어 (대상 레이어) 를 불러오거나 생성합니다.
3. 대상 레이어를 편집 모드로 설정합니다.
4. 범례에서 원본 레이어를 클릭해서 활성화시킵니다.
5.  Select Features by area or single click 도구를 이용해서 원본 레이어에서 피처 (들) 을 선택합니다.
6.  Copy Features 도구를 클릭합니다.
7. 범례에서 대상 레이어를 클릭해서 활성화시킵니다.
8.  Paste Features 도구를 클릭합니다.
9. 편집 모드를 해제하고 변경 사항을 저장합니다.

원본과 대상 레이어가 서로 다른 스키마를 가지고 있다면 (필드명과 유형이 동일하지 않다면) 어떻게 될까요? QGIS 는 일치하는 데이터를 붙여넣고 나머지는 무시합니다. 대상 레이어에 속성을 복사해 넣을 필요가 없는 경우, 필드 및 데이터 유형을 어떻게 구성하든 상관없습니다. 하지만 전체를 — 피처와 그 속성까지 — 복사하고자 하는 경우, 원본과 대상 레이어의 스키마가 일치하는지 확인해야 합니다.

When you paste features into another layer, QGIS checks the attribute values against the *constraints* defined in the destination layer (for example, `NOT NULL`, `Unique`, or expression constraints). If one or more pasted features contain invalid attribute values that do not meet these constraints, a dialog will appear, listing all affected features and fields. When pasting a single feature, you can decide whether to:

- Edit the invalid values directly in the dialog. The corrected values will be used when pasting.
- *Paste Anyway*: the feature is pasted as-is, even with invalid values.

When pasting multiple features, the dialog lists all invalid fields across all features, allowing bulk review and correction. You can decide per field or per feature whether to:

- Edit the invalid values directly in the dialog. The corrected values will be used when pasting.

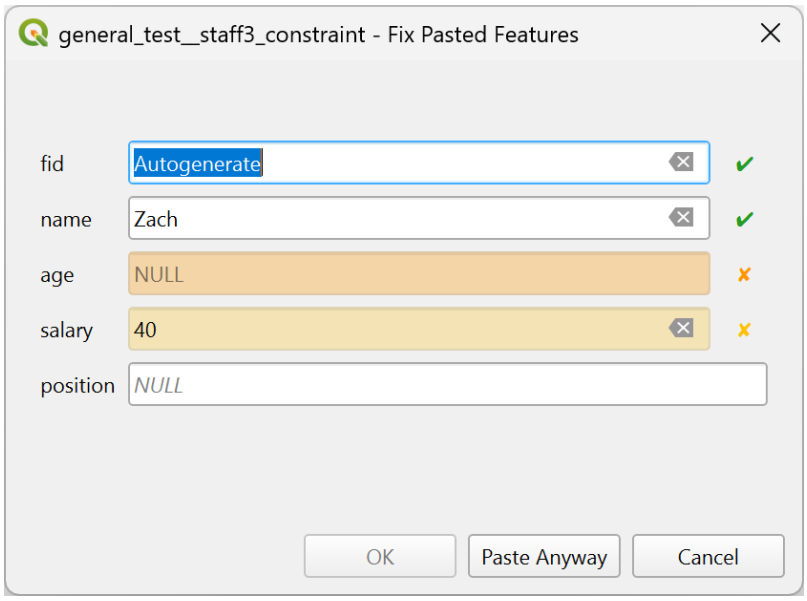



그림 12.86: Dialog for handling invalid attribute values when pasting a single feature

- *Discard All*: no features pasted.
- *Discard All Invalid*: pasted only features with valid attribute values.
- *Paste All (Including Invalid)*: all features are pasted as-is, even with invalid values.
- *Skip*: skip the specific feature, allowing proceeding with the rest.

**참고**

**붙여넣은 피처의 합치성**

원본과 대상 레이어가 동일한 투영체를 사용하고 있는 경우, 붙여넣은 피처가 원본 레이어와 동일한 도형을 가지게 될 것입니다. 하지만, 대상 레이어의 투영체가 다른 경우, QGIS 는 도형이 동일한지 보장할 수 없게 됩니다. 왜냐하면 서로 다른 투영체 간에 변환할 때 자잘한 반올림 오류가 발생하기 때문입니다.

**팁**

**문자열 속성을 또다른 속성으로 복사하기**

사용자의 속성 테이블에 문자열 유형의 새 열을 생성했는데, 더 긴 길이를 가진 다른 속성 열로 그 값을 붙여넣으려는 경우, 열의 길이가 대상 열의 길이로 늘어날 것입니다. GDAL 셰이프파일 드라이버가 삽입될 데이터의 길이를 동적으로 수용하기 위해 문자열 및 정수형 필드를 자동 확장하기 때문입니다.

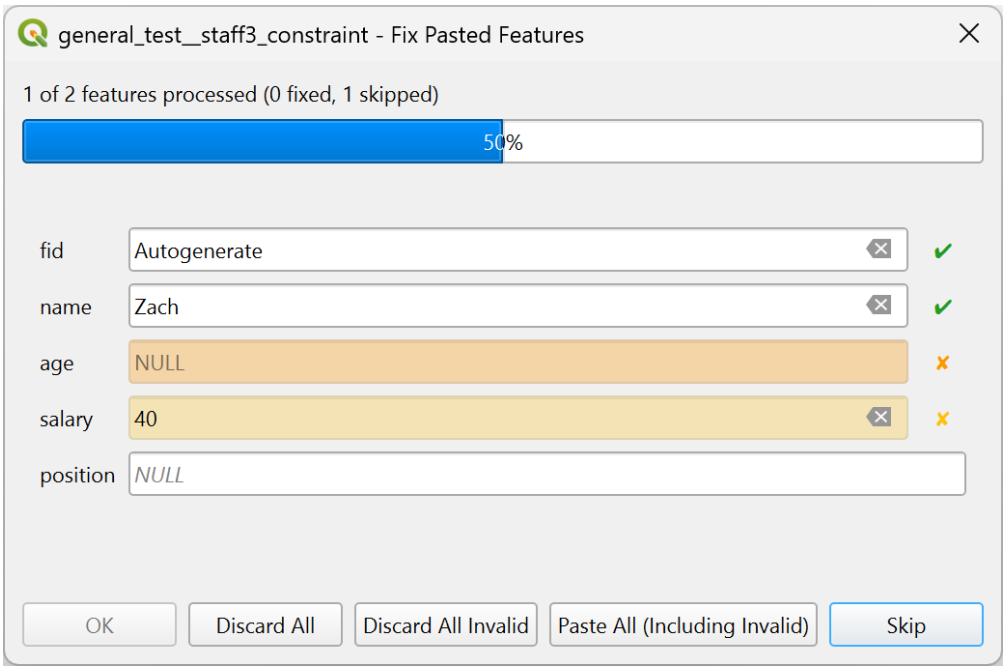



그림 12.87: Dialog for handling invalid attribute values when pasting multiple features

### 선택한 피처 삭제하기

어떤 피처 전체 (속성 및 도형) 를 삭제하려는 경우, 먼저 표준적인 Select Features by area or single click 도구를 사용해서 도형을 선택할 수 있습니다. 속성 테이블에서 선택할 수도 있습니다. 선택을 마치고 나면, `Delete` 또는 `Backspace` 키를 누르거나 Delete Selected 도구를 클릭해서 피처를 삭제하십시오. 선택한 피처 여러 개를 한 번에 삭제할 수 있습니다.

피처를 삭제하는 데 디지타이즈 작업 툴바에 있는 Cut Features 도구를 이용할 수도 있습니다. 이 도구는 피처를 사실상 삭제하지만"공간 클립보드"에 옮겨 놓기도 합니다. 즉, 삭제할 피처를 잘라내는 겁니다. 이후에 Paste Features 도구를 통해 되돌려 놓을 수 있습니다. 기초 수준의 실행 취소 능력이죠. 잘라내기, 복사 및 붙여넣기는 현재 선택한 피처를 대상으로 작동합니다. 즉 한 번에 하나 이상을 대상으로 작업할 수 있다는 뜻입니다.

### 실행 취소 및 재실행

Undo 및 Redo 도구는 벡터 편집 작업을 실행 취소하거나 재실행할 수 있습니다. 실행 취소/재실행 이력에 있는 모든 작업을 보여주는 고정 가능한 (dockable) 위젯도 있습니다. (그림 12.88 를 참조하세요.) 이 위젯은 기본적으로 표시되지 않습니다. 이 위젯을 표시하려면 툴바를 오른쪽 클릭한 다음 *Undo/Redo Panel* 을 활성화해야 합니다. 하지만 이 위젯이 표시되지 않은 경우에도 실행 취소/재실행 능력은 켜져 있습니다.

*Undo* 버튼을 클릭하거나 `Ctrl+Z` (또는|osx| `Cmd+Z`) 조합키를 누르면, 모든 피처와 속성의 상태가 되돌린 작업 이전의 상태로 돌아갑니다. 일반적인 벡터 편집 작업이 아닌 변경 사항 (예: 플러그인을 통한 변경 사항) 은 해당 변경 사항이 어떻게 수행됐는지에 따라 되돌릴 수도 있고 되돌리지 못 할 수도 있습니다.

실행 취소/재실행 이력 위젯을 이용하려면, 그냥 이력 목록에 있는 작업을 하나 선택하십시오. 모든 피처가 선택한 작업 직후의 상태로 돌아갈 것입니다.

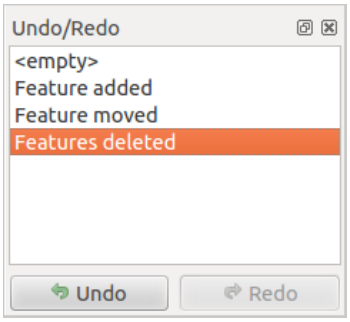



그림 12.88: 실행 취소/재실행 디지타이즈 작업 이력

### 편집한 레이어 저장하기

레이어가 편집 모드 상태인 경우, 모든 변경 사항이 QGIS 의 메모리에 남아 있습니다. 즉 변경 사항이 즉시 데이터소스 또는 디스크에 커밋 또는 저장되는 것이 아닙니다. 현재 레이어에 편집 사항을 저장하고 싶지만 편집 모드를 해제하지 않고 계속 편집 작업을 하고 싶은 경우, Save Layer Edits 아이콘을 클릭하면 됩니다. Toggle editing 아이콘으로 편집 모드를 끄는 경우 (또는 QGIS 를 종료하는 경우) 사용자의 변경 사항을 저장할 거냐 또는 무시할 거냐를 묻는 대화창이 열립니다.

변경 사항을 저장할 수 없는 경우 (예: 디스크에 남은 공간이 없거나, 범위를 벗어난 속성값이 있는 경우) QGIS 의 내부 메모리 상태가 보존됩니다. 사용자가 편집 사항을 조정해서 다시 저장할 수 있게 하기 위해서입니다.

> **팁**
>
> **데이터 무결성**
>
> 편집 작업을 시작하기 전에 항상 사용자 데이터소스를 백업하도록 권장합니다. QGIS 개발자들이 사용자 데이터의 무결성을 보존하기 위해 모든 노력을 다 하고 있지만, 어떤 보장도 해드릴 수 없기 때문입니다.

### 여러 레이어를 한 번에 저장하기

이 기능은 여러 레이어를 디지타이즈할 수 있게 해줍니다. 사용자가 여러 레이어에서 만든 모든 변경 사항을 저장하려면 Save for Selected Layers 를 선택하십시오. Rollback for Selected Layers 를 클릭하면, 선택한 모든 레이어에 대한 디지타이즈 작업을 되돌릴 수도 있습니다. 선택한 레이어에 대한 편집 작업을 종료하려면, Cancel for Selected Layer(s) 를 통해 쉽게 끝낼 수 있습니다.

프로젝트의 모든 레이어 편집 작업에서 동일한 기능들을 사용할 수 있습니다.

> **팁**
>
> **여러 레이어 변경 사항을 한 번에 편집, 저장 또는 되돌리는 데 트랜잭션 그룹 이용하기**
>
> 동일한 PostGreSQL 데이터베이스로부터 나온 레이어들을 작업하는 경우, *Project* → *Properties…* → *Data Sources* 메뉴의 *Automatically create transaction groups where possible* 옵션을 활성화시키면 레이어들의 습성 (편집 모드 시작/종료, 변경사항을 동시에 저장 또는 되돌리기) 을 동기화합니다.

### 12.3.5 고급 디지타이즈 작업

표 12.4: 벡터 레이어 고급 편집 툴바

| 아이콘 | 목적 | 아이콘 | 목적 |
|---|---|---|---|
|  | 고급 디지타이즈 작업 도구 활성화 | | |
|  | 피처 (들) 이동 |  | 피처 (들) 복사 및 이동 |
|  | 피처 (들) 회전 |  | 피처 단순화 |
|  | 피처 크기 조정 | | |
|  | 고리 추가 |  | 부분 추가 |
|  | 고리 채우기 |  | 방향 바꾸기 |
|  | 고리 삭제 |  | 부분 삭제 |
|  | 곡선 오프셋 |  | 피처 재형성 |
|  | 부분 분할 |  | 피처 분할 |
|  | 선택한 피처의 속성 병합 |  | 선택한 피처 병합 |
|  | 포인트 심볼 회전 |  | 포인트 심볼 오프셋 |
|  | 피처 다듬기 (Trim) 또는 연장 | | |

#### 피처 (들) 이동

 Move Feature(s) 도구는 기존 피처를 이동시킬 수 있습니다:

1. 이동시킬 피처 (들) 를 선택하십시오.
2. 맵 캔버스 상에서 변위의 원점을 나타내는 위치를 클릭하십시오. 정확한 포인트를 선택하기 위해 스냅 기능을 사용할 수 있습니다.

   원점 좌표를 정확하게 설정하기 위해고급 디지타이즈 작업 제약조건 을 이용할 수도 있습니다. 이런 경우:

   1. 먼저  버튼을 클릭해서 패널을 활성화시킵니다.
   2. `x` 를 입력한 다음 사용자가 원하는 원점에 대응하는 값을 입력합니다. 그리고 옵션 옆에 있는  버튼을 눌러 값을 고정시킵니다.
   3. `y` 좌표도 동일한 방법으로 설정합니다.
   4. 맵 캔버스를 클릭하면 지정한 좌표에 사용자의 원점을 배치합니다.
3. 맵 캔버스를 이동해서 변위의 대상 포인트를 가리키십시오. 계속 스냅 모드를 사용할 수도 있고, 또는 앞에서 설명한대로 고급 디지타이즈 작업 패널을 사용해서 전이 (□□) 의 종단점을 배치하기 위한 상호보완적인 `distance` 및 `angle` 배치 제약조건을 지정할 수도 있습니다.
4. 맵 캔버스를 클릭하십시오: 전체 피처가 새 위치로 이동합니다.

 Copy and Move Feature(s) 도구를 사용하면 피처 (들) 의 전이된 복사본을 마찬가지로 생성할 수 있습니다.

> **참고**
>
> *Move Feature(s)* 또는 *Copy and Move Feature(s)* 도구로 맵 캔버스를 처음 클릭했을 때 어떤 피처도 선택되지 않은 경우, 액션의 영향을 받는 것은 포인터 아래 있는 피처뿐입니다. 따라서 여러 피처를 이동시키려면 먼저 피처들을 선택해야 합니다.

### 피처 (들) 회전

맵 캔버스에 있는 하나 이상의 피처를 기울이려면 Rotate Feature(s) 도구를 사용하십시오:

1. Rotate Feature(s) 아이콘을 클릭합니다.
2. 기울이려는 피처를 클릭하십시오. 피처의 중심을 기울기 중심으로 참조하고, 기울인 피처의 미리보기를 표시하며, 현재 *Rotation* 각도를 표시하는 위젯이 열립니다.
3. 새 배치에 만족한다면 맵 캔버스를 클릭하십시오. 또는 텍스트란에 기울기 각도를 직접 입력해도 됩니다. *Snap to* ° 박스를 이용해서 기울기 값에 제약조건을 걸 수도 있습니다.
4. 여러 피처를 한번에 기울이려면, 먼저 피처들을 선택해야 합니다. 기본적으로 결합된 도형들의 중심을 기준으로 기울어질 것입니다.

기본적인 피처 중심점 대신 고정점 (anchor point) 을 사용할 수도 있습니다. `Ctrl` 키를 누른 채 맵 캔버스를 클릭하면, 해당 포인트를 새 기울기 중심점으로 사용할 것입니다.

맵을 클릭하기 전에 `Shift` 키를 누른 상태인 경우, 45 도씩 기울어질 것입니다. 이후 사용자 입력 위젯에서 각도를 수정할 수 있습니다.

피처 기울이기를 끝내려면, `ESC` 키를 누르거나 다시 Rotate Feature(s) 아이콘을 클릭하면 됩니다.

### 피처 크기 조정

Scale Feature 도구는 피처 회전 도구와 비슷하지만, 선택한 피처의 기울이기를 수행하는 대신 피처의 도형 크기를 재조정합니다. 앵커 포인트를 기준으로 크기를 변경하고, 맵 캔버스 상단 구석에 나타나는 위젯에서 크기 조정 비율을 직접 지정할 수 있습니다.

### 피처 단순화

Simplify Feature 도구는 도형의 무결성이 유지되는 한 꼭짓점 개수를 줄이거나 밀도를 높여서 라인 또는 폴리곤 도형의 형태를 대화형 작업으로 변형시킬 수 있습니다:

1. Simplify Feature 도구를 선택하십시오.
2. 피처를 클릭하거나 피처 위로 직사각형을 드래그하십시오.
3. 적용할 *Method* 를 정의할 수 있는 대화창이 열립니다. 다음과 같은 방법을 선택할 수 있습니다:
   - 도형을 단순화 하는, 즉 꼭짓점을 원본보다 줄이는 방법입니다. `Simplify by distance`, `Simplify by snapping to grid` 또는 `Simplify by area (Visvalingam)` 를 사용할 수 있습니다. 그 다음 단순화에 사용할 *Tolerance* 단위를 `Layer units`, `Pixels` 또는 `Map units` 가운데 하나로 지정해야 합니다. 허용 오차가 클수록 더 많은 꼭짓점을 삭제할 수 있습니다.

- 또는 `Smooth` 옵션의 도움으로 새 꼭짓점을 추가해서도형의 밀도를 높이는 방법도 있습니다. 기존의 각 꼭짓점에서 시작하는 두 선분 상의 선분 길이의 백분율을 나타내는 *Offset* 거리에 각각 꼭짓점을 배치합니다. *Iterations* 옵션에서 새 꼭짓점을 배치하는 과정의 반복 횟수도 설정할 수 있습니다. 반복 횟수가 높을수록 꼭짓점도 늘어나서 피처가 더 평탄화됩니다.

사용자가 사용한 설정은 편집 작업을 또는 프로젝트를 종료해도 저장될 것입니다. 따라서 다음에 피처를 단순화하는 경우에도 동일한 설정을 사용할 수 있습니다.

4. 대화창 하단에 적용될 수정 사항을 요약해서 (작업 전후의) 피처 및 꼭짓점 개수의 목록과 (변경되는 비율을) 함께 표시합니다. 또 맵 캔버스의 기존 피처 위에 예상 피처를 고무줄 색상으로 표시합니다.
5. 예상 피처가 사용자의 필요를 만족시킨다면, *OK* 를 클릭해서 수정 사항을 적용하십시오. 그렇지 않을 경우, *Cancel* 을 클릭하거나 맵 캔버스를 오른쪽 클릭하면 작업을 중단할 수 있습니다.

> **참고**
>
> 렌더링만을 위해 도형을 단순화하는 *Settings* → *Options* → *Rendering* 메뉴의 피처 단순화 옵션과는 달리, Simplify Feature 도구는 데이터소스에 있는 피처의 도형을 영구히 변경합니다.

### 부분 추가

Add Part 도구로 선택한 피처에 부분을 추가해서 멀티포인트, 멀티라인 또는 멀티폴리곤 피처를 생성할 수 있습니다. 기존 피처 외부에서 이 새 부분을 디지타이즈해야만 하는데, 이 기존 피처는 먼저 선택돼 있어야 합니다.

도형이 없는 피처에 도형을 추가하는 경우에도 Add Part 도구를 사용할 수 있습니다. 먼저, 속성 테이블에서 피처를 선택한 다음 Add Part 도구로 새 도형을 디지타이즈하십시오.

> **참고**
>
> **Order of vertices in polygon parts**
>
> Unlike the OGC standards, QGIS doesn't constrain vertices of the exterior boundary of a polygon feature to be ordered counterclockwise. Thus, you can find both directions in a layer. However, every parts of the same multipolygon feature will have their outer vertices ordered following the same direction.
>
> You can however use the 오른손 법칙 강제하기 algorithm to constrain features of a layer to have vertices of their outer boundaries ordered in the clockwise direction.

### 부분 삭제

Delete Part 도구는 다중 피처에서 부분을 삭제할 수 있습니다. (예를 들면 멀티폴리곤 피처에서 폴리곤을 삭제할 수 있습니다.) 이 도구는 포인트, 라인 및 폴리곤 모든 다중 부분 도형에 대해 작동합니다. 더우기, 피처의 도형 요소를 완전히 제거할 수도 있습니다. 부분을 삭제하려면, 그냥 대상 피처 안에 있는 부분을 클릭하면 됩니다.

### 고리 추가

툴바에 있는  Add Ring 아이콘을 클릭하면 고리 폴리곤을 생성할 수 있습니다. 기존 폴리곤 영역 내부에'구멍'으로 간주되는 내곽 폴리곤을 디지타이즈할 수 있다는 뜻으로, 이 경우 외곽 및 내곽 폴리곤 경계선 사이의 영역만 고리 폴리곤으로 남게 됩니다.

To add a ring:

1. Select the feature(s) to modify
2. Activate the  Add Ring tool
3. Draw a polygon within the selected geometries, using the aforementioned *techniques*. A hole appears in the selected geometries.
4. If no geometry is selected when the ring is drawn, then a hole is added to each of the polygons the ring is drawn over.

> **참고**
>
> **Order of vertices in polygon rings**
>
> Unlike the OGC standards, QGIS doesn't constrain vertices of the exterior boundary of a polygon feature to be ordered counterclockwise. Thus, you can find both directions in a layer. However, every rings of the same (multi)polygon feature will have their vertices ordered in the opposite direction to the outer boundary's.
>
> You can however use the 오른손 법칙 강제하기 algorithm to constrain features of a layer to have vertices of their outer boundaries ordered in the clockwise direction, and vertices of their interior rings ordered in the counter-clockwise direction.

### 고리 채우기

 Fill Ring 도구는 다른 폴리곤 안에 중첩되는 영역 없이 완전히 들어가는 폴리곤 피처를 생성할 수 있습니다. 즉 기존 피처 안에 있는 구멍을 덮는 새 피처를 말이죠. 이런 피처를 생성하려면:

1.  Fill Ring 도구를 선택하십시오.
2. 기존 피처 위로 새 폴리곤을 그리십시오: QGIS 가 새 폴리곤 도형에 (  Add Ring 도구를 사용하듯이) 고리를 추가한 다음 그 고리와 일치하는 도형을 가진 새 피처를 (  Add polygon feature 도구로 내곽 경계선을투사 하듯이) 생성합니다.
3. 또는, 고리가 피처에 이미 존재하는 경우 고리 위로 마우스를 가져가 `Shift` 키를 누른 채 왼쪽 클릭하십시오: 해당 위치에 구멍을 채우는 새 피처가 그려집니다.

   새 피처의 *Feature Attributes* 양식이"부모"피처의 값 그리고/또는필드 제약조건 과 함께 열립니다.

### 고리 삭제

Delete Ring 도구는 구멍 안쪽을 클릭해서 기존 폴리곤 내부의 고리를 삭제할 수 있습니다. 이 도구는 폴리곤 및 멀티폴리곤 피처에 대해서만 작동합니다. 폴리곤의 외곽 고리에 이 도구를 사용해도 아무것도 변경되지 않습니다.

### 피처 재형성

툴바에 있는 Reshape Features 도구를 사용해서 라인 및 폴리곤 피처를 재형성할 수 있습니다. 라인의 경우, 원본 라인을 첫 번째 교차점에서 마지막 교차점까지의 라인 부분으로 대체합니다.

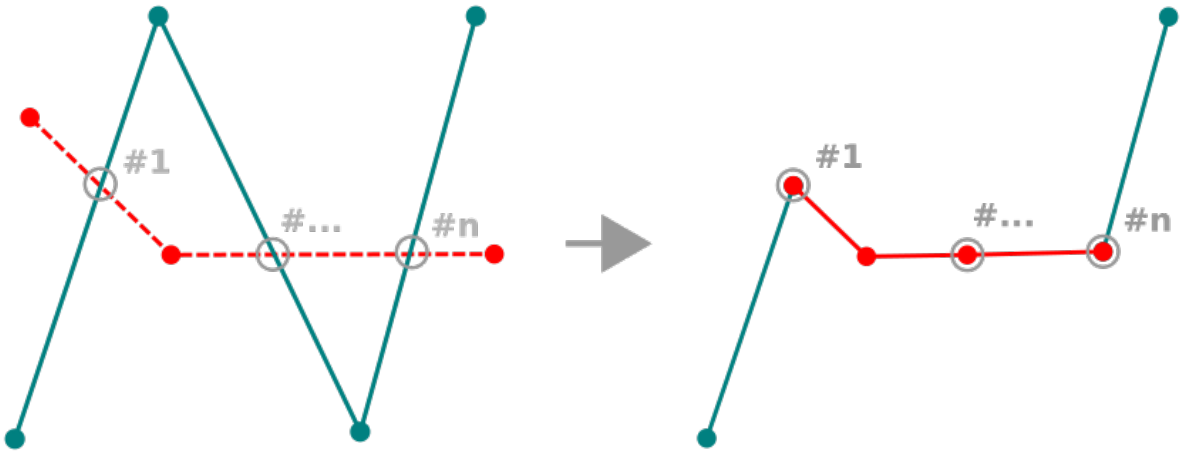



그림 12.89: 라인 재형성하기

**팁**

**재형성 도구로 라인스트링 도형을 연장시키기**

Reshape Features 도구를 이용해서 기존 라인의 첫 번째 또는 마지막 꼭짓점에 스냅한 다음 새 라인을 그리면 기존 라인스트링 도형을 늘릴 수 있습니다. 새 라인을 승인하면 피처 도형이 두 라인의 결합체로 변합니다.

폴리곤의 경우, 이 도구는 폴리곤의 경계선을 재형성합니다. 제대로 동작하려면, 재형성 도구의 라인이 폴리곤 경계선과 최소한 두 번 교차해야만 합니다. 라인을 그리려면, 맵 캔버스를 클릭해서 꼭짓점을 추가하십시오. 그리기를 마치려면, 오른쪽 클릭만 하면 됩니다. 라인의 경우와 마찬가지로, 첫 번째와 마지막 교차점 사이에 있는 선분만 처리됩니다. 폴리곤 내부에 있는 재형성 라인의 선분은 폴리곤을 잘라내고, 외부에 있는 선분은 폴리곤을 확장시킬 것입니다.

폴리곤의 경우, 재형성 작업이 때로 의도하지 않은 결과를 낼 수도 있습니다. 이 도구는 폴리곤의 자잘한 부분을 수정하는 데 유용하지, 본체를 수정하기는 힘듭니다. 또 재형성 라인이 여러 폴리곤 고리와 교차해서도 안 됩니다. 그런 경우 무결하지 않은 폴리곤이 생성되기 때문입니다.

**참고**

재형성 도구는 폴리곤 고리 또는 닫힌 라인의 시작점을 변경시킬 수도 있습니다. 즉, '두 번'나타나던 포인트가

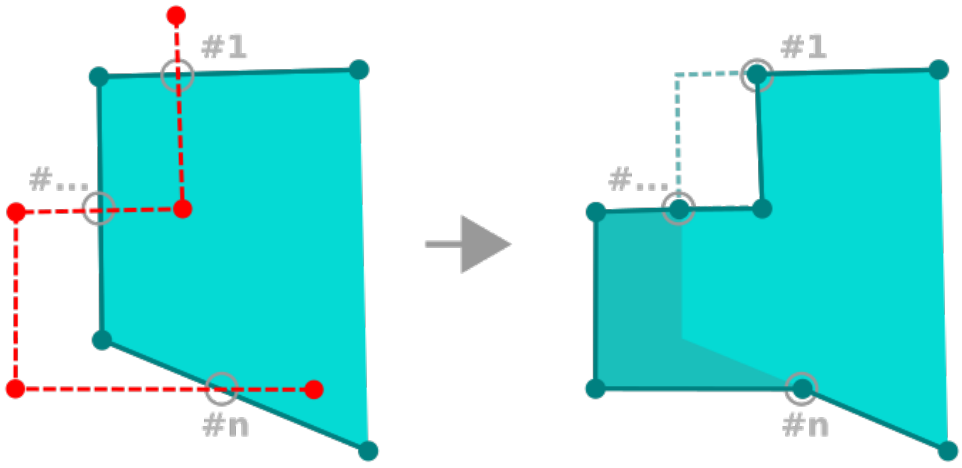



그림 12.90: 폴리곤 재형성하기

더 이상 그러지 않을 수도 있다는 뜻입니다. 대부분의 응용 프로그램에서 문제가 없을 수도 있지만, 고려해야 할 사항임에는 틀림없습니다.

### 곡선 오프셋

The Offset Curve tool creates parallel shifts of line layers. The tool can be applied to the edited layer (the geometries are modified) or also to background layers (in which case it creates copies of the lines / rings and adds them to the edited layer). It is thus ideally suited for the creation of distance line layers. The *User Input* dialog pops-up, showing the displacement distance and other settings.

To create a shift of a line or polygon layer, you must first go into editing mode and activate the Offset Curve tool. Then click on a feature to shift it. Move the mouse and click where wanted or enter the desired distance in the user input widget. Holding `Ctrl` during the 2nd click will make an offset copy. Your changes may then be saved with the Save Layer Edits tool.

For geometries on background layers make sure that snapping is on and hold `Ctrl` to select the geometry from the background. Also hold `Ctrl` when doing the second click. Geometries will be converted to the target layer geometry type.

QGIS options dialog (Digitizing tab then **Curve offset tools** section) or the icon in the *User Input* dialog allows you to configure *some parameters* like **Join style**, **Quadrant segments**, **Miter limit** and **End cap style**.

### 라인 반전

Changing the direction of a line geometry can be useful for cartographical purposes (e.g., for orienting symbols or labels that follow the line direction) or when preparing for network analysis where the direction of flow or travel matters.

This tool works only on line layers. For multipart lines, you can reverse individual parts without affecting the other parts of the feature.

라인 방향을 변경하려면:

1. Select a line layer in the *Layers* panel and make it editable.
2. Activate the reverse line tool by clicking  Reverse line in the *Advanced Digitizing* toolbar.
3. Move the pointer over a line feature. The part under the cursor is highlighted.
4. Click on the line: the direction of the line (or part) is reversed.
5. You can reverse other lines by clicking on them.

> **팁**
>
> For bulk line direction changes, use the 라인 방향 반전시키기 processing algorithm which can reverse all lines in a layer or a selection at once.

### 피처 분할

 Split Features 도구를 사용해서 피처를 두 개 이상의 독립적인 새 피처들로 분할할 수 있습니다. 예를 들면 분할된 각 도형은 속성 테이블의 새 행과 대응합니다.

라인 또는 폴리곤 피처를 분할하려면:

1.  Split Features 도구를 선택하십시오.
2. Draw a line across the feature(s) you want to split. If a selection is active, only selected features are split. The original feature is then assigned the biggest geometry resulting from the splitting, and new features are created for the remaining parts. Fields of the features are filled/updated according to the datasource provider rules or their *splitting policy*.
3. 이제 산출되는 모든 피처의 모든 속성을 평소대로 수정할 수 있습니다.

> **팁**
>
> **클릭 한 번으로 폴리라인을 새 피처들로 분할하기**
>
>  Split Features 도구로 폴리라인 피처의 기존 꼭짓점에 스냅하고 클릭하면, 해당 피처를 새 피처 2 개로 분할합니다.

### 부분 분할

QGIS 는 다중 부분 피처의 부분을 분할해서 부분의 개수를 늘릴 수 있습니다.  Split Parts 도구를 사용해서 분할하고 싶은 부분을 가로지르는 라인을 그리기만 하면 됩니다.

> **팁**
>
> **클릭 한 번으로 폴리라인을 새 부분들로 분할하기**
>
>  Split Parts 도구로 폴리라인 피처의 기존 꼭짓점에 스냅하고 클릭하면, 해당 피처를 동일한 피처에 속해 있는 새 폴리라인 2 개로 분할합니다.

### 선택한 피처 병합

 Merge Selected Features 도구는 기존 피처들을 합쳐서 새 피처를 생성할 수 있습니다. 기존 피처의 도형들을 병합해서 새 피처를 생성합니다. 기존 피처들에 공통 경계가 없는 경우, 멀티폴리곤/멀티폴리라인/멀티포인트 피처를 생성합니다.

1. 먼저, 결합하고자 하는 피처들을 선택하십시오.
2. 그 다음  Merge Selected Features 아이콘을 클릭하십시오.
3. In the new dialog, the *Merge* line at the bottom of the table shows the attributes of the resulting feature. If any *Merge policy* have been defined, they are automatically applied to populate this row. However, you can override these values manually. Modify the values in the following ways:
   - 대응하는 셀에서 값을 직접 대체합니다.
   - 테이블에서 행을 선택하고 *Take attributes from selected feature* 를 클릭하면 선택한 최초 피처의 값을 사용합니다.
   - *Take attributes from the largest geometry* 를 클릭하면 가장 긴 라인 피처, 가장 큰 폴리곤, 또는 가장 많은 부분들을 가진 멀티포인트의 속성을 사용합니다.
   - *Skip all fields* 를 클릭하면 빈 속성을 사용합니다.
   - 테이블 상단에 있는 드롭다운 메뉴를 펼친 다음 앞의 옵션들 가운데 하나를 선택하면 대응하는 필드에만 적용할 것입니다. 여기에서 초기 피처 속성을 (필드의 유형에 따라 최소값, 최대값, 중앙값, 합계, 개수, 연결 (concatenation) 등등을) 집계하도록 선택할 수도 있습니다. 전체 함수 목록을 알고 싶다면통계 요약 패널 을 참조하세요.

   > **참고**
   >
   > 레이어가 필드에 기본값 또는 절을 가지고 있을 경우, 이를 병합된 피처의 초기 값으로 사용합니다.

4. 수정 사항을 적용시키려면 *OK* 를 클릭하십시오. 레이어에 이전에 선택한 피처들을 대체하는 단일 [멀티] 피처를 생성합니다.

### 선택한 피처의 속성 병합

Merge Attributes of Selected Features 도구는 피처들의 경계를 병합하지 않고서도 피처에 동일한 속성을 적용할 수 있게 해줍니다. 이 도구의 대화창은 *Merge Selected Features* 도구의 대화창과 동일합니다. 다만 선택한 개체들의 속성은 동일하게 변하지만, 개체 도형은 그대로 유지된다는 점이 다릅니다.

### 포인트 심볼 회전

Rotate Point Symbols 도구를 통해 맵 캔버스에 있는 포인트 심볼의 기울기를 개별적으로 변경할 수 있습니다.

1. 먼저, 기울기 값을 저장할 필드를 지정해야 합니다. 심볼데이터 정의 기울기 속성 에 필드를 할당하면 됩니다:
   1. *Layer Properties* → *Symbology* 대화창에서 심볼 편집기 대화창을 찾으십시오.
   2. 심볼 레이어의 (되도록) 최상위 *Marker* 심볼의 *Rotation* 옵션 옆에 있는 *Data-defined override* 위젯을 클릭하십시오.
   3. *Field Type* 콤보박스에서 필드를 선택하십시오. 이 필드의 값들을 따라 각 피처의 심볼을 기울일 것입니다.

      기울기 값을 제어하기 위해보조 데이터 저장소 필드를 생성하려면 *Store data in project* 항목을 체크해도 됩니다.

> **참고**
>
> **반드시 동일한 필드가 모든 심볼 레이어에 할당되도록 하십시오**
>
> 최상위 심볼 트리에서 데이터 정의 기울기 필드를 설정하면 모든 심볼 레이어에 해당 필드를 적용합니다. 이는 *Rotate Point Symbols* 도구로 그래픽 심볼을 기울이기 위한 전제조건입니다. 실제로, 동일한 필드가 심볼 레이어의 기울기 속성에 첨부되지 않았을 경우, 이 도구는 작동하지 않을 것입니다.

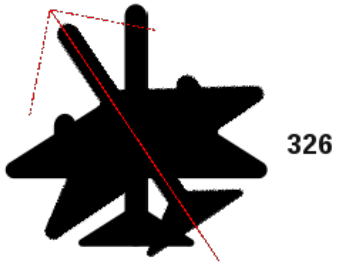



그림 12.91: 포인트 심볼 기울이기

2. 그 다음 맵 캔버스에 있는 포인트 심볼을 Rotate Point Symbols 도구로 클릭하십시오.
3. 마우스를 주위로 움직여보십시오. 기울기 값을 가진 빨간 화살표가 가시화될 것입니다. (그림 12.91 참조) 마우스를 움직이는 도중 `Ctrl` 키를 누르면, 기울기가 15 도 단계로 바뀔 것입니다.
4. 원하는 각도 값으로 기울어졌다면, 한번 더 클릭하십시오. 심볼을 이 새 기울기로 렌더링하고 이에 맞춰 관련 필드도 업데이트됩니다.

   심볼을 기울이지 않으려면 오른쪽 클릭하면 됩니다.

### 포인트 심볼 오프셋

 Offset Point Symbols 도구는 포인트 심볼이 맵 캔버스에 렌더링된 위치를 대화형으로 변경할 수 있습니다. 이 도구는  Rotate Point Symbols 도구와 비슷하게 동작합니다만, 심볼의 각 레이어마다 데이터 정의된 *Offset (X,Y)* 속성에 필드를 연결해야 한다는 점이 다릅니다. 이 필드는 심볼이 맵 캔버스에서 이동한 피처의 오프셋 좌표로 채워질 것입니다.

1. 심볼의 *Offset (X,Y)* 속성의 데이터 정의 위젯에 필드를 연결하십시오. 심볼이 다중 레이어로 이루어진 경우, 각 레이어마다 필드를 할당해야 할 수도 있습니다.
2.  Offset Point Symbols 도구를 선택하십시오.
3. 포인트 심볼을 클릭하십시오.
4. 새 위치로 이동시키십시오.
5. 한번 더 클릭하십시오. 심볼이 새 위치로 이동했습니다. 원래 위치로부터의 오프셋 값은 연결된 필드에 저장됩니다.

   심볼을 오프셋시키지 않으려면 오른쪽 클릭하면 됩니다.

**참고**

 Offset Point Symbols 도구는 포인트 심볼 자체를 이동시키지 않습니다. 그러려면  Vertex Tool (Current Layer) 또는  Move Feature 도구를 사용해야 합니다.

### 피처 다듬기/연장하기

The  Trim/Extend tool allows you to shorten or lengthen segments of a (multi)line or (multi)polygon geometry to converge with a selected segment (the cutting line). This results in a modified geometry with a vertex snapped to a reference segment or in its prolongation. Depending on how the selected geometries are placed in relation to each other, the tool will either:

- **다듬기 (trim)**: 라인 선분 또는 폴리곤 경계선에서 절단선을 넘어가는 부분을 제거합니다.
- **연장하기 (extend)**: 폴리곤 경계선 또는 라인 선분이 절단선에 스냅될 수 있도록 연장시킵니다.

기존 도형을 다듬거나 연장하려면:

1.  Trim/Extend 도구를 선택하십시오.
2. Click the reference segment, i.e., the segment with respect to which you want to extend or trim another segment. It doesn't need to be on the active layer. A red dotted line appears, visually extending the reference segment across the map canvas according to the active layer's CRS.
3. Hover over the target segment, i.e., the one you want to trim or extend. It does not need to be the last segment of the geometry, but has to be on the active layer. QGIS displays a preview of the feature's geometry with the segment limited to its intersection with the highlighted red dotted extending line.
   1. If an extension of the segment is required, click anywhere on the segment.
   2. In the case of a trim, click the part that should remain.
   3. In either case, the feature's geometry is updated accordingly. When both segments are in 3D, the tool performs an interpolation on the limit segment to get the Z value.
4. If you need to use the same reference segment for trimming or extending many features:
   1. Press `Shift` while selecting the reference segment.

2. Click consecutively on the segments to modify and each will be trimmed or extended accordingly.

> **참고**
>
> Snapping is automatically enabled when this tool is activated. Your original snapping settings will be restored once the tool is deactivated.

> **주의**
>
> Trim/Extend 도구를 사용하는 동안 수정된 도형에 주의를 기울이십시오. 입력 도형에 따라 무결하지 않은 도형을 생성해서 레이어를 저장하지 못 하게 될 수도 있습니다.

### 12.3.6 형태 디지타이즈 작업

*Shape Digitizing* 툴바는 정다각형 형태의 라인 또는 폴리곤 피처를 그릴 수 있는 일련의 도구들을 제공합니다. 이 툴바는 사용자가 *Digitizing Toolbar* 에서 선택할 수 있는 Digitize Shape 도형 그리기 방법 과 동기화됩니다. 이 툴바를 사용하려면:

1. 툴바 보이기: *View → Toolbars → Shape Digitizing*
2. 도형의 형태를 생성하거나 수정하는 도구를 선택하십시오. 예: Add line feature, Add polygon feature, Add part, Add ring, Reshape Features, ...
3. *Digitizing Toolbar* 에 Digitize with segment 버튼이 활성화됩니다. 이 작업이 처음이라면 *Shape Digitizing* 툴바의 도구들을 활성화시키기 위해 Digitize Shape 으로 전환해야 할 수도 있습니다.
4. 형태 디지타이즈 작업 도구를 고른 다음 그리십시오.

#### 반경으로 원호 라인스트링 그리기

Circular string by radius 버튼을 누르고 만곡 도형을 구성하는 노드 2 개와 반경을 지정하면 원호 도형을 가진 라인 또는 폴리곤 피처를 추가할 수 있습니다.

1. 두 번 왼쪽 클릭해서 도형을 이루는 포인트 2 개를 배치하십시오.
2. 맵 캔버스의 오른쪽 상단에 나타나는 *Radius* 위젯에 (두 포인트 사이의 거리에 대응하는) 현재 반경이 표시됩니다. 반경 란의 값을 사용자가 원하는 값으로 편집하십시오.
3. 커서를 움직이면 이 제약조건들과 일치하는 원호의 오버뷰를 표시합니다. 원하는 원호가 표시될 때 오른쪽 클릭해서 결정하십시오.
4. 또다른 원호를 그리려면 새 포인트를 추가하십시오.

> **참고**
>
> **만곡 도형은 호환되는 데이터 제공자에서만 만곡 도형으로 저장됩니다**

Although QGIS allows to digitize curved geometries within any editable data format, you need to be using a data provider (e.g. PostgreSQL, Oracle Spatial, memory layer, GML or WFS) that supports curves to have features stored as curved, otherwise QGIS segmentizes the circular arcs.

### 원 그리기

원을 그릴 수 있는 도구들의 집합이 있습니다. 다음은 그 도구들의 설명입니다.

원은 원호 스트링으로 변환됩니다. 따라서반경으로 원호 라인스트링 그리기 에서 설명한대로 데이터 제공자가 허용하는 경우 만곡 도형으로 저장될 것이고, 아닌 경우 QGIS 가 원호를 선분화할 것입니다.

-  Circle from 2 points: 포인트 2 개로 지름 및 원의 방향을 정의합니다. (왼쪽 클릭, 오른쪽 클릭)
-  Circle from 3 points: 포인트 3 개가 원의 경계선 상에 있다고 가정하고 원을 그립니다. (왼쪽 클릭, 왼쪽 클릭, 오른쪽 클릭)
-  Circle by a center point and another point: 중심 포인트와 원의 경계선 상에 있다고 가정한 포인트로 원을 그립니다. (왼쪽 클릭, 오른쪽 클릭) 고급 디지타이즈 작업 패널 과 함께 사용하는 경우, 첫 번째 클릭 이후 거리값을 설정하고 고정해서 이 도구를"Add circle from center and radius"도구로 사용할 수 있습니다.
-  Circle from 3 tangents: 선분 3 개에 접하는 원을 그립니다. **'선분에 스냅'옵션을 활성화해야만 한다는 사실을 기억하십시오.** (스냅 허용 오차 및 검색 반경 설정 을 참조하세요.) 선분을 클릭해서 접선 (tangent) 을 추가하십시오. 접선 2 개가 평행한 경우, 첫 번째 평행 접선을 클릭한 위치의 좌표를 이용해서 원의 배치를 결정합니다. 접선 3 개가 모두 평행하다면 오류 메시지를 띄우고 입력을 초기화합니다. (왼쪽 클릭, 왼쪽 클릭, 오른쪽 클릭)
-  Circle from 2 tangents and a point: 접선 3 개로 원을 그리는 도구와 유사하지만, 접선 2 개를 선택한 다음 반경을 입력하고 원하는 중심을 선택해야 한다는 점이 다릅니다.

### 타원 그리기

타원을 그릴 수 있는 도구들의 집합이 있습니다. 다음은 그 도구들의 설명입니다.

타원을 원호 (circular string) 로 변환할 수 없기 때문에, 타원은 언제나 선분화될 것입니다.

-  Ellipse from center and two points: 지정한 중심 (center), 긴 반지름 (major axis), 짧은 반지름 (minor axis) 으로 타원을 그립니다. (왼쪽 클릭, 왼쪽 클릭, 오른쪽 클릭)
-  Ellipse from center and a point: 중심과 모서리를 지정한 경계 상자에 내접하도록 타원을 그립니다. (왼쪽 클릭, 오른쪽 클릭)
-  Ellipse from extent: 서로 바라보는 두 모서리를 지정한 경계 상자에 내접하도록 타원을 그립니다. (왼쪽 클릭, 오른쪽 클릭)
-  Ellipse from foci: 두 초점 (foci) 을 나타내는 포인트 2 개와, 타원 경계선 위에 있는 포인트 1 개를 지정해서 타원을 그립니다. (왼쪽 클릭, 왼쪽 클릭, 오른쪽 클릭)

### 직사각형 그리기

직사각형을 그릴 수 있는 도구들의 집합이 있습니다. 다음은 그 도구들의 설명입니다.

- Rectangle from center and a point: 지정한 중심과 모서리로 직사각형을 그립니다. (왼쪽 클릭, 오른쪽 클릭)
- Rectangle from extent: 서로 바라보는 두 모서리를 지정해서 직사각형을 그립니다. (왼쪽 클릭, 오른쪽 클릭)
- Rectangle from 3 points (distance): 포인트 3 개를 지정해서 방향을 가진 직사각형을 그립니다. 첫 번째와 두 번째 포인트가 첫 변의 길이와 각도를 결정합니다. 세 번째 포인트는 다른 변의 길이를 결정합니다. 변들의 길이를 설정하기 위해고급 디지타이즈 작업 패널 을 사용할 수 있습니다. (왼쪽 클릭, 왼쪽 클릭, 오른쪽 클릭)
- Rectangle from 3 points (projected): 앞의 도구와 동일하지만, 두 번째 변의 길이를 첫 번째 변 위의 세 번째 포인트의 투영으로부터 계산합니다. (왼쪽 클릭, 왼쪽 클릭, 오른쪽 클릭)

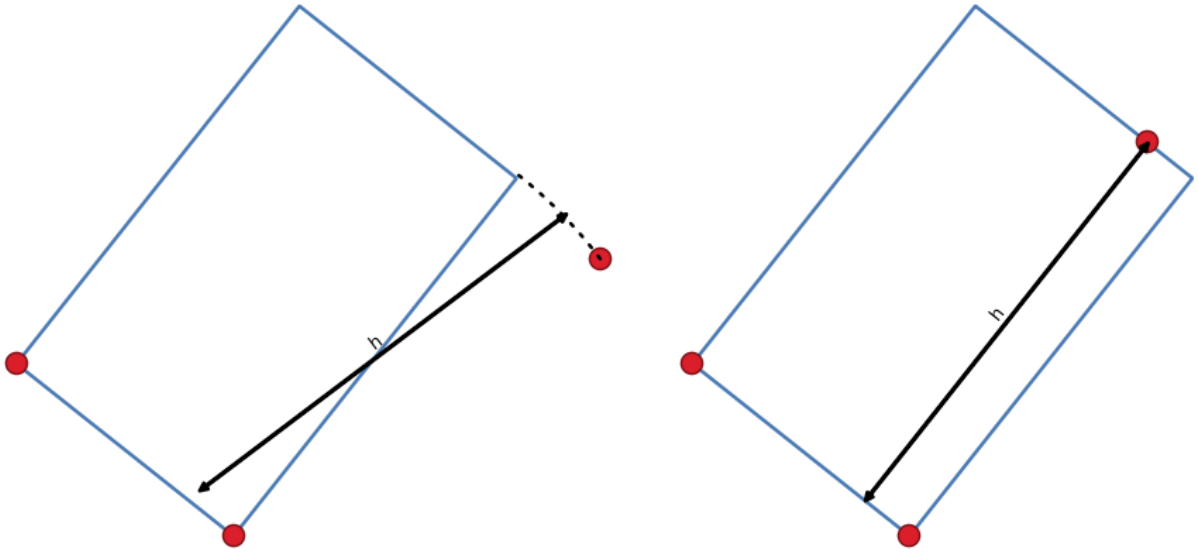



그림 12.92: 포인트 3 개로 직사각형 그리기: 거리 (오른쪽) 와 투영 (왼쪽)

### 정다각형 그리기

정다각형 (regular polygon) 을 그리기 위한 도구 집합이 있습니다. 다음은 그 도구들의 설명입니다. 첫 번째 포인트의 위치를 왼쪽 클릭해서 지정합니다. 다각형의 변의 개수를 설정할 수 있는 대화창이 나타납니다. 정다각형을 완성할 위치를 오른쪽 클릭하십시오.

- Regular polygon from two points: 포인트 2 개가 첫 번째 변의 길이 및 각도를 결정하는 정다각형을 그립니다.
- Regular polygon from center and a point: 지정한 중심 포인트로부터 정다각형을 그립니다. 두 번째 포인트는 변의 중점까지의 각도 및 거리를 결정합니다.
- Regular polygon from center and a corner: 앞의 도구와 동일하지만, 두 번째 포인트가 꼭짓점까지의 각도 및 거리를 결정합니다.

### 12.3.7 고급 디지타이즈 작업 패널

새로운 또는 기존 도형을 캡처, 재형성, 분할하는 경우 고급 디지타이즈 작업 패널을 이용할 수도 있습니다. 라인을 정확히 평행으로 또는 특정 각도에 직각으로 디지타이즈할 수도 있고, 라인을 지정 각도로 고정시킬 수도 있습니다. 게다가, X 및 Y 좌표는 물론 3 차원 피처의 경우 Z 좌표 또는 M 값을 직접 입력해서 사용자의 새 도형을 정밀하게 정의할 수도 있습니다.

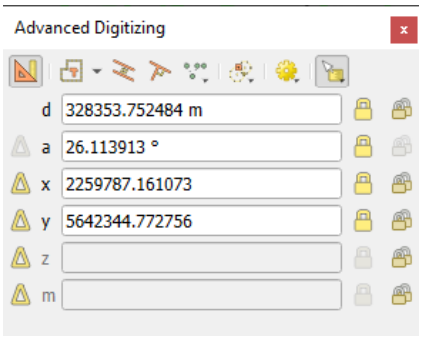



그림 12.93: 고급 디지타이즈 작업 패널

툴바를 오른쪽 클릭하거나, *View* ▸ *Panels* ▸ 메뉴로부터, 또는 `Ctrl+4` 키를 누르면 *Advanced Digitizing* 패널을 열 수 있습니다. 패널이 가시화되고 나면, Enable advanced digitizing tools 아이콘을 클릭해서 고급 디지타이즈 작업 도구를 활성화하십시오.

> **참고**
>
> 맵 뷰가 지리 좌표인 경우 이 도구들을 활성화할 수 없습니다.

고급 다지타이즈 작업 도구의 목적은 맵 캔버스에서 디지타이즈 작업 도중 마우스를 이동시킬 때 좌표, 길이 그리고 각도를 고정하는 것입니다.

또한 상대 또는 절대 참조로 제약 조건을 생성할 수도 있습니다. 상대 참조란 이전 꼭짓점 또는 선분에 따라 다음 꼭짓점 제약 조건의 값이 변한다는 뜻입니다.

#### 툴바

*Digitizing panel* 의 최상단에 다음과 같은 버튼들이 있습니다:

- Enable advanced digitizing tools: 고급 디지타이즈 작업 도구를 활성화합니다.
- Construction mode: 클릭한 위치를 거리, 각도, X, Y, Z, 또는 M 상대값을 고정시키기 위한 참조 포인트로 재사용하기 위해 캡처할 수 있습니다. 더 자세한 내용은작성 모드 를 참조하세요.
- Parallel: 기존 라인에 평행한 라인을 그립니다. 더 자세한 내용은평행 및 수직 라인 을 참조하세요.
- Perpendicular: 기존 라인에 수직인 라인을 그립니다. 더 자세한 내용은평행 및 수직 라인 을 참조하세요.
- Construction Tools: 작성 도구는 기존 요소의 추정 좌표를 기반으로 꼭짓점의 배치를 강제하는 옵션을 제공합니다:
  - *Line Extension*: 선분 위에 커서를 가져가면 맵 캔버스 전체에 해당 선분의 연장선인 보라색 점선을 표시합니다. 이 가상의 선 어디에나 꼭짓점을 스냅시킬 수 있습니다.

  -  *X/Y Point*: 꼭짓점 위에 커서를 가져가면 맵 캔버스 전체에 꼭짓점의 X 또는 Y 좌표를 따라 보라색 점선을 표시합니다. 이 가상의 선 어디에나 꼭짓점을 스냅시킬 수 있습니다. 서로 다른 두 꼭짓점 위에 커서를 가져가서 두 꼭짓점에 대한 가상의 좌표선을 생성한 다음 그 교차점에 새 꼭짓점을 스냅시킬 수도 있습니다.

-  2-circle Point Intersection: allows you to digitize a point or vertex at the intersection of two circles. (more at *2-circle point intersection*).

-  Snap to common angles: 커서를 움직이는 동안 다음 꼭짓점을 추가하기 위해 스냅할 수 있는 가상의 선을 표시합니다. 마지막으로 추가한 꼭짓점과 사전 설정 (0.1°, 0.5°, 1°, 5°, 10°, 15°, 18°, 22.5°, 30°, 45° 또는 90° 단계를 따르는) 목록에 있는 (이전 선분 기준 절대 또는 상대) 각도가 이 가상의 선을 정의합니다. 이 기능을 비활성화하려면 *Do not snap to common angles* 를 선택하십시오.

  정확한 디지타이즈 작업을 위해피처에 스냅하기 를 공통 각도에 스냅하기와 함께 사용할 수 있습니다. 스냅할 대상 요소를 유지하는 방법을 세밀하게 제어하기 위해, *Snapping priority* 항목에서 피처에 스냅하기를 우선할지 또는 공통 각도에 스냅하기를 우선할지 여부를 지정할 수 있습니다. 디지타이즈 작업 중에 한 방법에서 다른 방법으로 전환할 수 있기 때문에, 그 동안 다른 모든 스냅 작업 옵션을 비활성화시키지 않아도 됩니다. 디지타이즈 작업 도중 각도 목록을 순환시키려면 `N` (또는 `Shift+N`) 키를 누르십시오.

-  Floater settings: *Show floater* 옵션을 체크한 경우, 디지타이즈 작업 도중 디지타이즈 정보를 가진 컨텍스트 메뉴가 커서를 따라다니게 됩니다. 패널의 단축키 를 통해 그 값들에 접근할 수 있고, 무결성 검증 후에 값들을 편집하고 (`Enter` 키를 눌러)  Locked 고정시킬 수 있습니다. 메뉴의 하단에서 표시할 정보 유형을 선택할 수 있습니다:

  - *Show distance*
  - *Show angle*
  - *Show XY coordinates*
  - *Show Z value*
  - *Show M value*
  - *Show bearing/azimuth*
  - *Show common snapping angle*

툴바 아래, 기본적으로 맵 캔버스 상의 커서 위치 또는 이동을 반영하는 값을 표시하는 텍스트란 여러 개가 있습니다. 이 값들을 편집하면 사용자가 편집 중인 항목의 위치를 강제할 수 있습니다:

- *d*: 참조 위치–보통 마지막으로 편집한 꼭짓점–로부터의 거리
- *a*: 참조 위치–보통 마지막으로 편집한 꼭짓점–로부터의 (절대 또는 상대) 각도
- *x*: 포인터 (커서) 의 X 좌표
- *y*: 포인터 (커서) 의 Y 좌표
- *z*: 포인터 (커서) 아래 있는 꼭짓점 또는 선분의 기본 Z 값 또는 Z 좌표
- *m*: 포인터 (커서) 아래 있는 꼭짓점 또는 선분의 기본 M 값 또는 M 값

### 키보드 단축키

고급 디지타이즈 작업 패널의 활용 속도를 높이려면, 다음과 같은 키보드 단축키를 사용할 수 있습니다:

표 12.5: 고급 디지타이즈 작업 패널 도구의 키보드 단축키

| 키 | 간단한 설명 | Ctrl+ 또는 Alt+ 키조합 | Shift+ |
|---|---|---|---|
| D | 거리 설정 | 거리 고정 | |
| A | 각도 설정 | 각도 고정 | 마지막 선분에 상대적인 각도 켜고끄기 |
| X | X 좌표 설정 | X 좌표 고정 | 마지막 꼭짓점에 상대적인 X 켜고끄기 |
| Y | Y 좌표 설정 | Y 좌표 고정 | 마지막 꼭짓점에 상대적인 Y 켜고끄기 |
| Z | Z 좌표 설정 | Z 좌표 고정 | 마지막 꼭짓점에 상대적인 Z 켜고끄기 |
| M | M 값 설정하기 | M 값 고정 | 마지막 꼭짓점에 상대적인 M 켜고끄기 |
| C | 작성 모드 변경 | | |
| P | 수직 및 평행 모드 변경 | | |

**참고**

Z 좌표 또는 M 값 옵션은 레이어 도형 차원과 호환되는 경우에만 사용할 수 있습니다.

### 절대 참조 디지타이즈 작업

아무것도 없는 상태에서 새 도형을 그리는 경우, 지정한 좌표에서 꼭짓점 디지타이즈를 시작할 수 있다면 매우 편리합니다.

예를 들어 폴리곤 레이어에 새 피처를 추가하려면  버튼을 클릭하십시오. 사용자가 피처 편집을 시작하고자 하는 정확한 좌표를 다음과 같이 입력할 수 있습니다:

1. *x* 텍스트란을 클릭합니다. (또는 X 단축키를 누릅니다.)
2. 원하는 X 좌표를 입력한 다음 Enter 키를 누르거나 오른쪽에 있는  아이콘을 클릭하면 맵 캔버스 상의 X 좌표의 축에 마우스가 고정됩니다.
3. *y* 텍스트란을 클릭합니다. (또는 Y 단축키를 누릅니다.)
4. 원하는 Y 좌표를 입력한 다음 Enter 키를 누르거나 오른쪽에 있는  아이콘을 클릭하면 맵 캔버스 상의 Y 좌표의 축에 마우스가 고정됩니다.
5. 레이어가 Z 좌표 또는 M 값을 가지고 있다면, 그에 대응하는 *z* 또는 *m* 위젯이 활성화되어 *Settings* ▸ *Options* ▸ *Digitizing* 탭에 설정된 기본값을 표시합니다.
    1. *z* 또는 *m* 텍스트란을 클릭합니다. (또는 각각 Z 또는 M 키보드 단축키를 누릅니다.)
    2. 원하는 좌표를 입력한 다음 Enter 키를 누르거나 오른쪽에 있는  아이콘을 클릭하면 위젯의 값이 고정됩니다.

**참고**

기존 피처로부터 어떻게 Z 좌표 또는 M 값을 자동으로 결정하는지에 대한 자세한 내용을 알고 싶다면*Z 좌표 또는 M 값 할당 규칙* 을 읽어보십시오.

6. 파란색 점선 2 개와 녹색 십자표가 사용자가 입력한 정확한 좌표를 표시합니다. 맵 캔버스를 클릭하면 녹색 십자표 위치에 꼭짓점을 추가합니다.

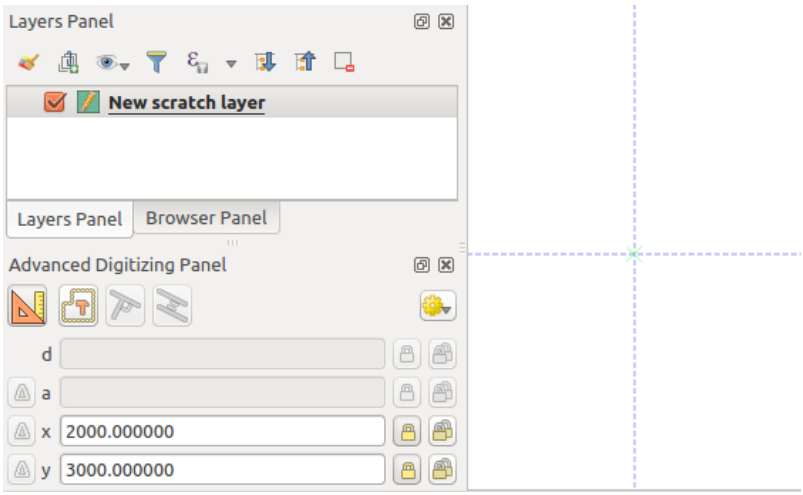



그림 12.94: 지정 좌표에서 그리기 시작

7. 다음 꼭짓점의 새로운 좌표쌍을 추가해서 앞의 과정을 반복할 수도 있고, 또는 또다른 (예를 들어 직선, 곡선 또는 자유) 디지타이즈 작업 모드 로 전환할 수도 있습니다.
8. 지정한 길이의 선분을 그리려면:
   1. *d (distance)* 텍스트란을 클릭하거나 `D` 키보드 단축키를 누르십시오.
   2. 거리 값을 (맵 단위로) 입력하십시오.
   3. `Enter` 키를 누르거나 오른쪽에 있는  버튼을 클릭하면 맵 캔버스에서 마우스 커서가 입력한 선분 길이로 고정됩니다. 거리 텍스트란에 입력한 값을 반경으로 하는 원이 마지막 꼭짓점을 둘러쌉니다. 이 원 위에 있는 십자표가 사용자가 클릭하는 다음 꼭짓점의 위치를 표시합니다.

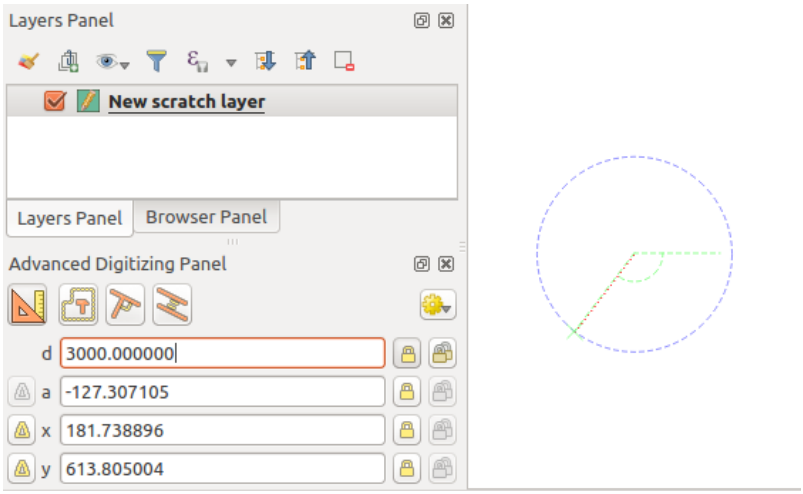



그림 12.95: 고정 길이 선분

9. 선분의 각도를 설정해서 꼭짓점 위치를 강제할 수도 있습니다. 앞에서 설명한 대로:
   1. *a (angle)* 텍스트란을 클릭하거나 `A` 키보드 단축키를 누르십시오.

2. 각도 값을 (도 단위로) 입력하십시오.

3. `Enter` 키를 누르거나 오른쪽에 있는 버튼을 클릭해서 각도를 고정하십시오. 맵 캔버스에 설정한 각도를 기반으로 기울어진, 마지막 꼭짓점을 지나가는 선이 나타납니다. 이 선 위에 있는 십자표가 사용자가 클릭하는 다음 꼭짓점의 위치를 표시합니다.

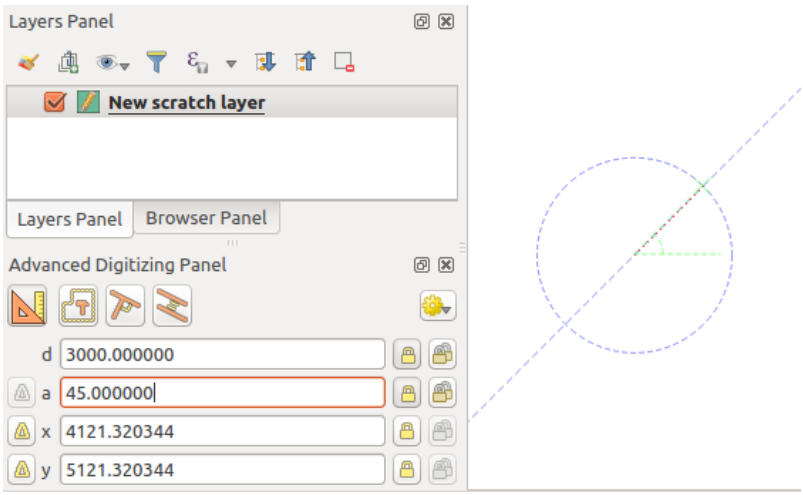



그림 12.96: 고정 각도 선분

**힌트**

`Ctrl+<key>` 또는 `Alt+<key>` 조합키를 누르면 대상 속성을 자동으로 고정하고 편집 란에 그 값을 입력합니다. 텍스트란에서 이 값을 수정한 다음 `Enter` 키를 누르면 끝입니다. Toggle floater 도구와 함께 이 조합키를 사용하면 키보드 디지타이즈 작업을 통해 많은 시간을 줄일 수 있습니다.

### 상대 참조 디지타이즈 작업

각도 또는 좌표의 절대값을 사용하는 대신, 직전에 디지타이즈한 꼭짓점 또는 선분에 상대적인 값을 사용할 수도 있습니다.

각도의 경우, *a* 텍스트란 왼쪽에 있는 버튼을 클릭해서 (또는 `Shift+A` 조합키를 눌러서) 이전 선분에 상대적인 각도를 켜고 끌 수 있습니다. 이 옵션을 켜면, 직전 선분과 마우스 포인터 사이의 각도를 측정합니다.

좌표의 경우, *x*, *y*, *z* 또는 *m* 텍스트란 왼쪽에 있는 버튼을 클릭해서 (또는 `Shift+<key>` 조합키를 눌러서) 이전 꼭짓점에 상대적인 좌표를 켜고 끌 수 있습니다. 이 옵션을 켜면, 직전 꼭짓점을 설정 좌표의 원점으로 삼아 좌표를 측정할 것입니다.

### 지속 고정

절대 참조 또는 상대 참조 디지타이즈 작업 모두, *Continuous lock* 버튼을 누르면 각도, 거리, X, Y, Z 및 M 제약조건을 계속 고정 상태로 만들 수 있습니다. 지속 고정을 사용하면 포인트 또는 꼭짓점 여러 개를 동일한 제약조건을 사용해서 디지타이즈할 수 있습니다.

### 작성 모드

Construction mode 아이콘을 클릭하거나 `C` 키보드 단축키를 누르면 작성 모드 를 켜고 끌 수 있습니다. 작성 모드에서는 맵 캔버스를 클릭해도 새 꼭짓점을 추가하지 않을 것입니다. 대신 클릭한 위치를 캡처한 다음 해당 위치를 거리, 각도, 또는 X, Y, Z, M 상대값을 고정시키기 위한 참조 포인트로 사용할 수 있게 해줍니다.

In the Construction mode you will find a drop-down menu where you can choose to:

- *Record Construction Guides*: All construction steps are rendered as dashed lines. Each step that you perform is visually represented, allowing you to trace the construction process. The guides are displayed as long as *Advanced Digitizing* remains active.
- *Show Construction Guides*: Allows you to make construction guides visible or hidden on the canvas. When enabled, all active guides are displayed, offering better spatial orientation and precision during the digitizing process.
- *Snap to Visible Construction Guides*: The guides are snap-able, allowing you to start new construction steps from any point along the existing guides.
- Choose *Clear Construction Guides* to remove all the guides from the canvas.

예를 들어, 기존 포인트에서 정확히 동일한 거리에 있는 포인트를 몇 개 그리는 데 작성 모드를 사용할 수 있습니다.

맵 캔버스 안에 기존 포인트가 존재하고 스냅 작업 모드를 정확하게 활성화한 경우, 기존 포인트로부터 지정한 거리 및 각도에 해당하는 위치에 다른 포인트를 쉽게 그릴 수 있습니다. 버튼 이외에도 Construction mode 아이콘을 클릭하거나 `C` 키보드 단축키를 눌러 작성 모드 를 활성화해줘야 합니다.

사용자가 해당 포인트로부터 거리를 계산하고자 하는 포인트 옆을 클릭하고 *d* 텍스트란을 클릭한 다음 (또는 `D` 단축키를 누른 다음) 원하는 거리를 입력하고 `Enter` 키를 눌러 다음과 같이 맵 캔버스에 마우스 위치를 고정하십시오:

새 포인트를 추가하기 전에, `C` 단축키를 눌러 작성 모드를 끄십시오. 이제 맵 캔버스를 클릭하면, 입력한 거리에 포인트가 배치될 것입니다.

각도 제약조건을 이용하면, 원래 포인트로부터 동일한 거리에 있지만 새로 추가한 포인트로부터 특정 각도에 있는 또다른 포인트를 생성할 수도 있습니다. Construction mode 아이콘을 클릭하거나 `C` 키보드 단축키를 눌러 작성 모드로 들어가십시오. 새로 추가한 포인트를 클릭한 다음 다른 포인트를 클릭해서 선분의 방향을 설정하십시오. 그리고 *d* 텍스트란을 클릭하고 (또는 `D` 단축키를 누르고) 원하는 거리를 입력한 다음 `Enter` 키를 누르십시오. *a* 텍스트란을 클릭하고 (또는 `A` 단축키를 누르고) 사용자가 원하는 각도를 입력한 다음 `Enter` 키를 누르십시오. 마우스의 커서 위치가 지정한 거리와 각도로 고정될 것입니다.

새 포인트를 추가하기 전에, `C` 단축키를 눌러 작성 모드를 끄십시오. 이제 맵 캔버스를 클릭하면, 입력한 거리에 포인트가 배치될 것입니다. 이 과정을 반복하면 여러 포인트를 추가할 수 있습니다.

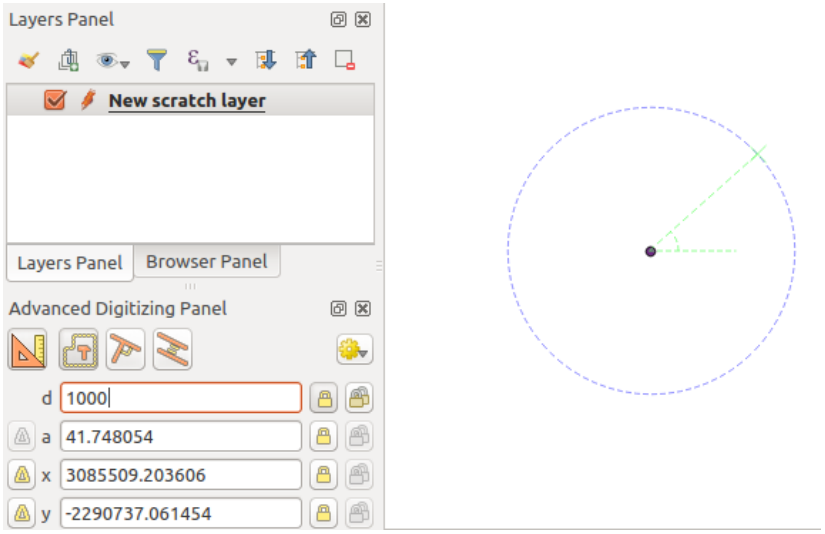



그림 12.97: 포인트로부터의 거리

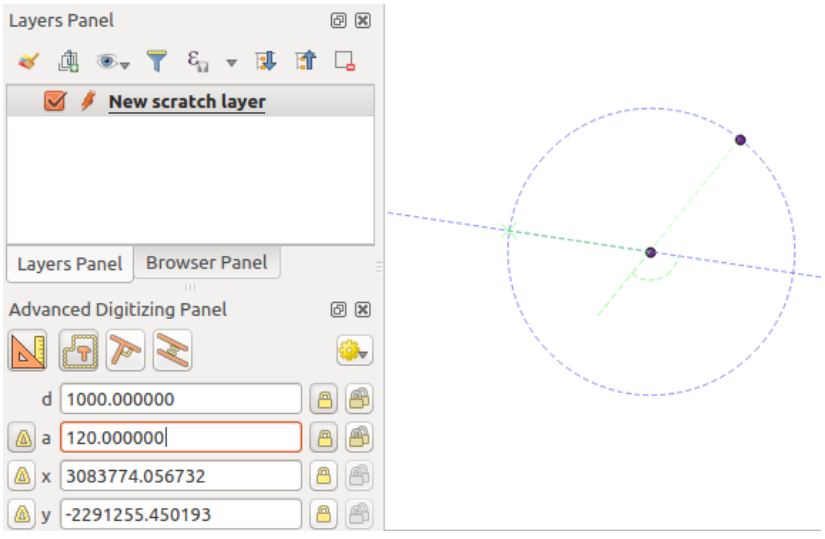



그림 12.98: 포인트들로부터의 거리 및 각도

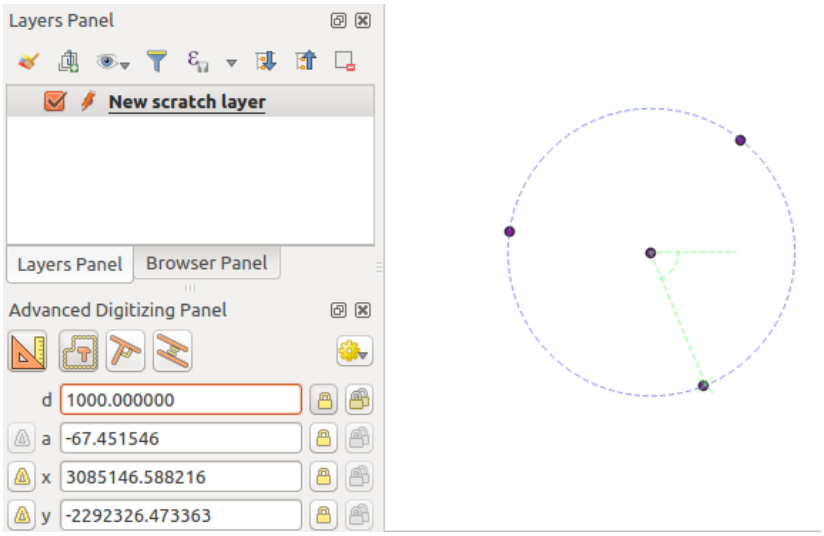



그림 12.99: 지정 거리 및 각도에 있는 포인트들

### 평행 및 수직 라인

앞에서 설명한 모든 도구는 Perpendicular 및 Parallel 도구와 함께 사용할 수 있습니다. 이 두 도구는 또다른 선분에 완벽하게 수직인 또는 평행한 선분을 그릴 수 있습니다. 대상 선분은 또다른 레이어 상에 있을 수도, 동일한 레이어 안에 있는 또다른 피처에 있을 수도, 또는 (자체 스냅 옵션 이 활성화된 상태라면) 디지타이즈 중인 피처의 선분일 수도 있습니다.

수직 선분을 그리려면:

1. 먼저 선분 꼭짓점 가운데 하나를 추가하십시오.
2. Perpendicular 아이콘을 클릭해서 (또는 `P` 키보드 단축키를 눌러서) 수직 선분 도구를 활성화하십시오.
3. 사용자가 수직 선분을 그리고자 하는 대상 선분을 클릭하십시오.
4. 대상 선분에 수직으로 이전 꼭짓점을 지나가는 가상 점선이 나타납니다. 다음 꼭짓점을 가상 점선 상의 위치로 강제하도록 각도 속성이 고정되고, 십자표가 가상 점선 상의 커서 위치를 나타냅니다. 원하는 위치를 클릭해서 새 꼭짓점을 배치하십시오.

평행 선분을 그리는 작업은 Parallel 아이콘을 클릭하는 (또는 `P` 키보드 단축키를 두 번 누르는) 것을 제외하면 앞의 단계들과 동일합니다.

이 두 도구는 사용자가 편집 작업을 하는 도중 올바른 수직 및 평행 각도를 찾아 해당 파라미터를 고정시킵니다. 편집 작업 중에 각도 파라미터를 고정 해제시키면 이 도구를 취소시킬 수 있습니다.

### 2-circle point intersection

To add a vertex at the intersection of two circles, follow these steps:

1. Click the 2-circle Point Intersection icon.
2. A dialog will open where you can define the parameters for *Circle #1* and *Circle #2*.
3. Click on the map canvas, and the tool will automatically calculate the *X* and *Y* coordinates for the centers of both circles.

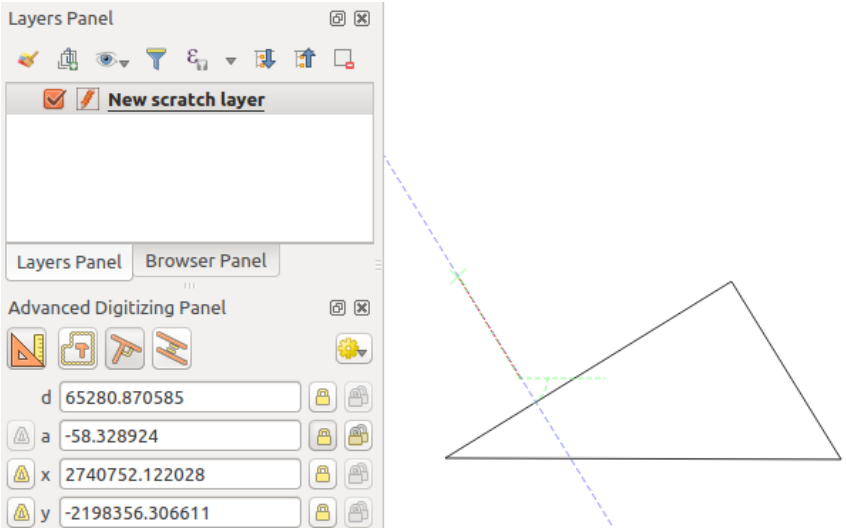



그림 12.100: 수직 디지타이즈 작업

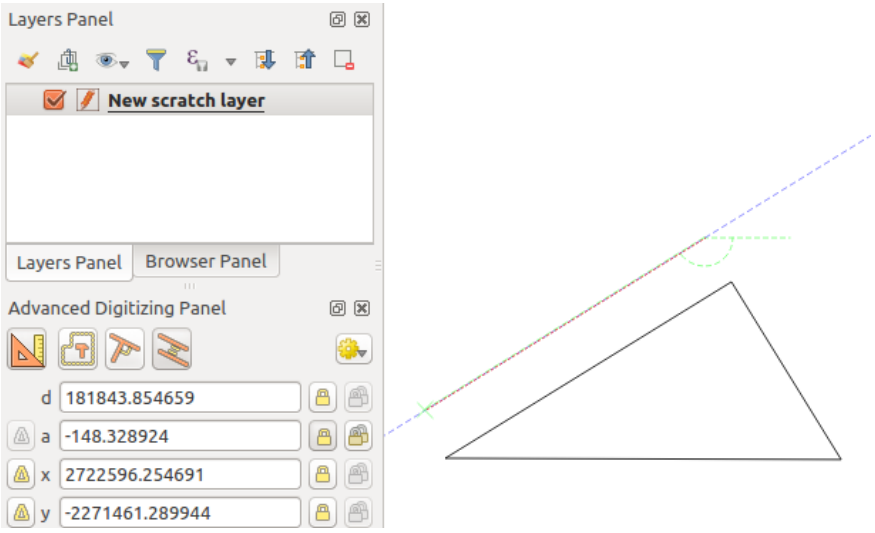



그림 12.101: 평행 디지타이즈 작업

4. Enter the distance *d* for each circle.
5. The tool will calculate and display the two intersection points of the circles.
6. Click one of the intersection points to add the new vertex.

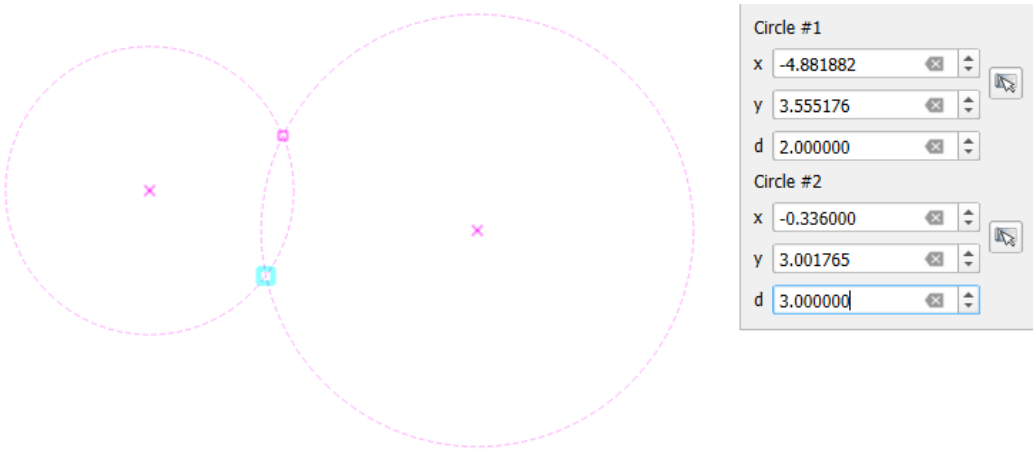



그림 12.102: 2-circle Point Intersection

### 12.3.8 제자리 (in-place) 공간 처리 레이어 수정자

공간 처리 메뉴 는 입력 피처 또는 입력 피처와 (동일 레이어 안에 있을 수도 있고 아닐 수도 있는) 다른 피처의 관계의 속성을 기반으로 새 피처를 생성하고 분석할 수 있는 수많은 도구를 제공합니다. 새 레이어를 산출물로 생성하는 것이 공통된 습성이지만, 몇몇 알고리즘은 입력 레이어를 수정할 수도 있습니다. 이를 사용하면 고급의 복잡한 작업을 통해 다중 피처 수정을 간편하게 자동화할 수 있습니다.

피처를 제자리에서 편집하려면:

1. *Layers* 패널에서 편집할 레이어를 선택하십시오.
2. 관심이 있는 피처를 선택하십시오. 이 단계를 건너뛰어도 되는데, 그럴 경우 레이어 전체에 수정 사항을 적용할 것입니다.
3. 공간 처리 툴박스 상단에 있는 Edit Features In-Place 버튼을 클릭하십시오. 알고리즘 목록을 필터링해서 제자리 수정을 지원하는 알고리즘만 표시할 것입니다. 예를 들면:
   - 레이어 수준에서가 아니라 피처 소스 수준에서 작동하는 알고리즘
   - 필드를 추가하거나 제거하는 등 레이어 구조를 변경하지 않는 알고리즘
   - 라인 레이어를 포인트 레이어로 바꾸는 등 도형 유형을 변경하지 않는 알고리즘
4. 사용자가 실행하려는 알고리즘을 찾아 더블클릭하십시오.

> **참고**
>
> 알고리즘이 (일반적인 입력/산출 레이어 파라미터를 제외하면) 사용자가 설정한 어떤 파라미터도 필요로 하지 않는 경우, 어떤 대화창도 열리지 않고 알고리즘이 즉시 실행됩니다.

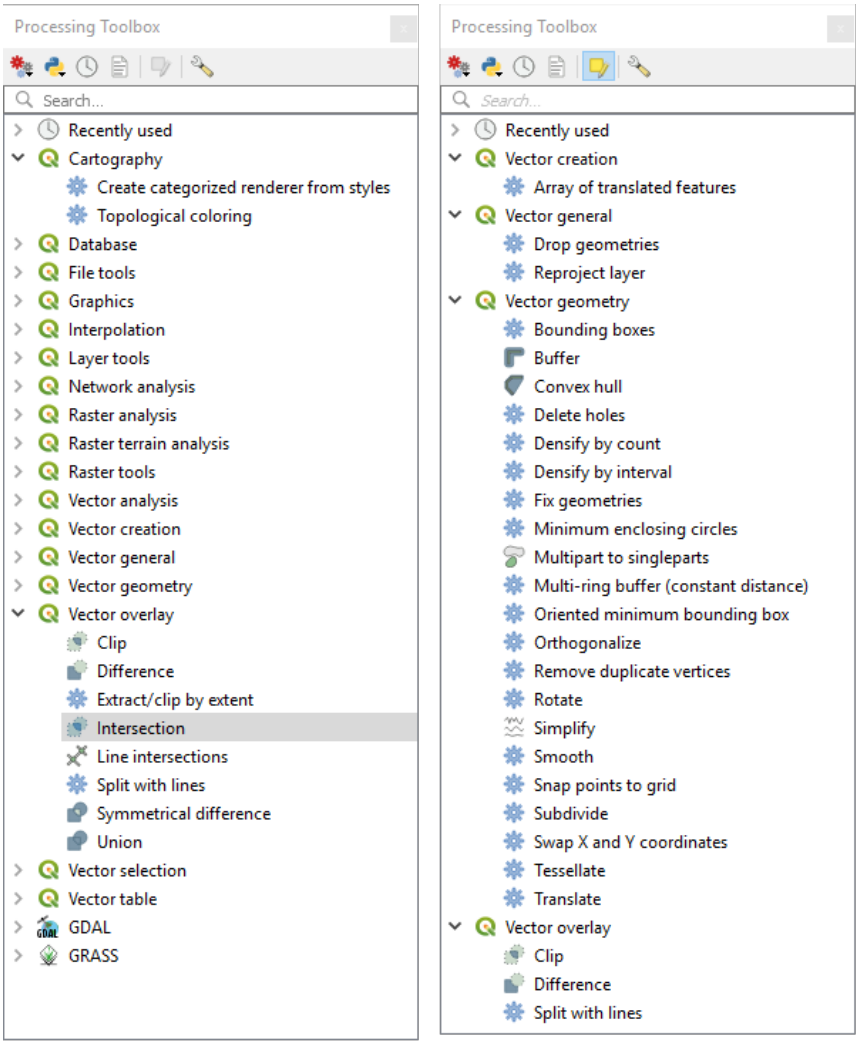



그림 12.103: 공간 처리 알고리즘: 전체 (왼쪽) 대 제자리 폴리곤 편집기 (오른쪽)

1. 일반적인 입력/산출 레이어 파라미터가 아닌 다른 파라미터가 필요한 경우, 알고리즘 대화창이 열립니다. 필요한 정보를 입력하십시오.
2. 활성화된 선택 집합이 있는지 여부에 따라 *Modify Selected Features* 또는 *Modify All Features* 를 클릭하십시오.

변경 사항을 레이어에 적용하고 편집 버퍼에 저장합니다. 레이어는 자동으로 편집 모드로 넘어가며, 레이어명 옆에 표시되는  아이콘은 저장하지 않은 수정 사항이 있다는 사실을 나타냅니다.

5. 다른 작업과 마찬가지로,  Save layer edits 버튼을 클릭하면 레이어에 변경 사항을 저장합니다. 물론  Undo 버튼을 클릭해서 모든 수정 사항을 되돌릴 수도 있습니다.

## 12.4 크로스 레이어 데이터 연결 및 편집 작업

서로 다른 레이어들의 데이터를 연결할 수 있는 기능은 GIS 소프트웨어의 필수적인 기능 가운데 하나입니다. 피처들 간의 공간 관계 또는 공유되는 속성을 기반으로 이런 연결을 만들 수 있습니다. QGIS 는 이러한 모든 결합들을 다룰 수 있는 다음과 같은 도구들을 제공합니다:

- 위치로 속성 결합하기, 최근접으로 속성 결합하기, 필드값으로 속성 결합하기 등등과 같이 데이터 연결의 결과로서 새 레이어를 생성할 수 있는 공간 처리 알고리즘들
- 데이터베이스 관리자 로부터 새 레이어를 생성하거나, 가상 레이어 로서 새 레이어를 생성할 수 있는 SQL 쿼리
- 지정한 레이어에 있는 피처들의 속성을 몇몇 일치하는 속성 (들) 을 바탕으로 또다른 레이어에 있는 피처들의 속성으로 일시적으로 확장시키는결합 속성 또는관계 설정

  결합 (join) 과 관계 (relation) 는 데이터베이스 테이블들에 저장되어 있는 데이터를 테이블의 내용을 합쳐서 가장 효율적으로 사용하기 위해 빌려온, 데이터베이스에서 사용되는 개념입니다. 서로 다른 레이어 (테이블) 들의 피처 (행) 들이 서로 연결될 수 있다는 뜻이죠. 서로 간에 일치하는 행의 개수는 어떤 값이든 (0 이든, 1 이든, 다수든) 될 수 있습니다.

### 12.4.1 두 레이어 간 피처 결합하기

QGIS 에서는 현재 레이어의 피처와 또다른 레이어의 피처를, 두 레이어가 공간적으로 활성화되어 있는지 그리고 도형의 유형이 일치하는지 관계 없이, **결합** 을 통해 연결시킬 수 있습니다. 이 결합은 두 레이어가 1 대 1 관계로 공유하는 속성을 기반으로 생성됩니다.

어떤 (다음에서 `target layer` 로 표기되는) 레이어 상에 결합을 생성하려면:

1. 레이어 메뉴에서 *Properties*  *Joins* 탭으로 가십시오.
2.  Add new join 아이콘을 클릭하면 *Add vector join* 대화창이 열립니다.
3. 대상 벡터 레이어와 연결하려는 *Join layer* 를 선택하십시오.
4. (`join layer` 로부터) *Join field* 를 그리고 (`target layer` 로부터) *Target field* 를 지정하십시오. 두 레이어에서 일치하는 피처를 찾는 데 쓰이는 필드들이기 때문에 두 필드가 공통된 값들을 가지고 있어야 합니다.
5. *OK* 를 누르면 *Join* 패널에 선택한 파라미터들의 요약이 추가됩니다.

앞의 단계들을 통해 생성된 결합은, 결합 레이어에서 **처음으로 일치하는 피처** 의 **모든** 속성을 대상 레이어의 피처에 추가시킵니다. 결합 과정에서 피처 쌍을 만드는 데에는 다음과 같은 논리가 쓰입니다:

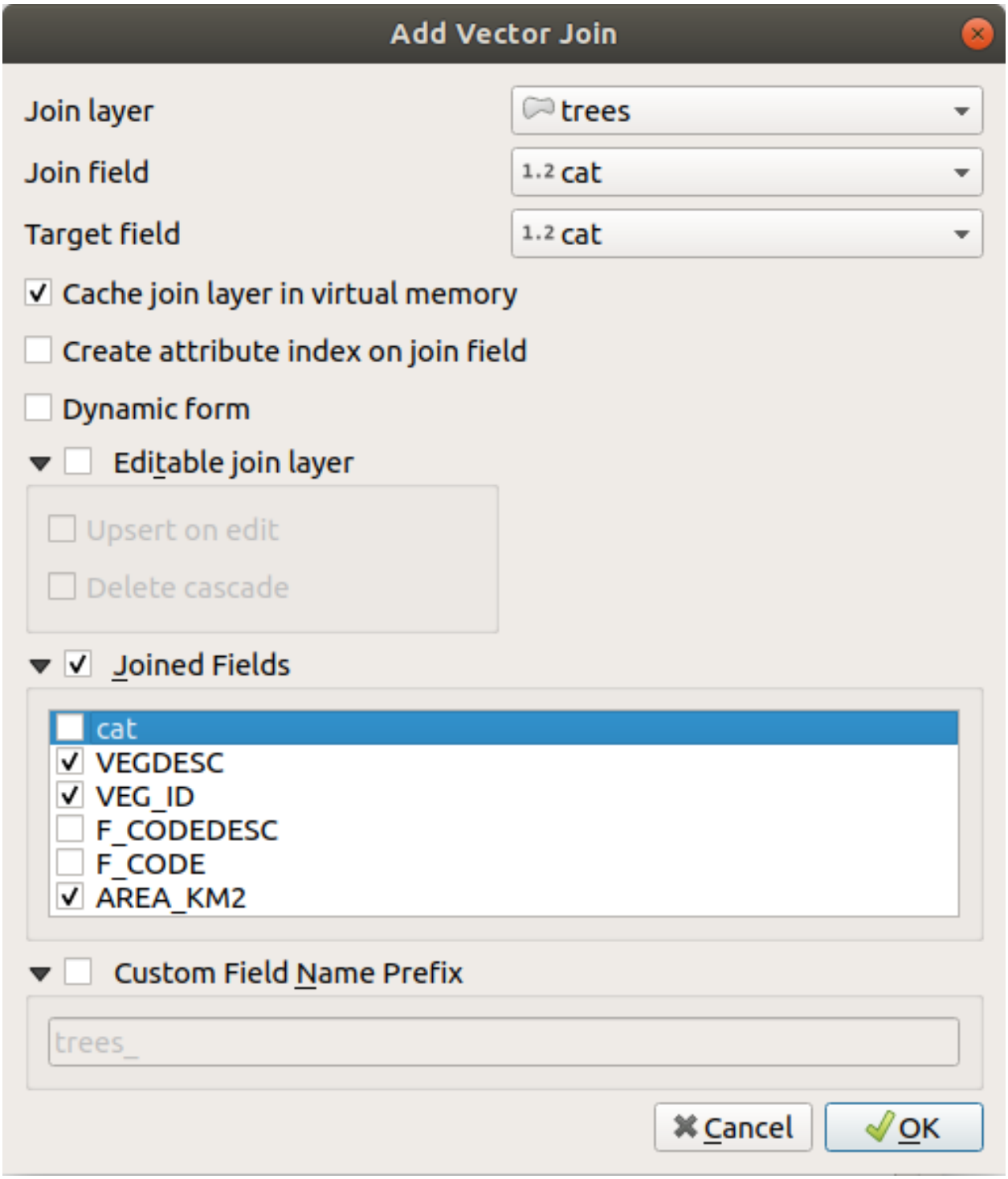



그림 12.104: 기존 벡터 레이어에 속성 테이블 결합

- 일치하는 피처가 있는지에 상관 없이, 대상 레이어 (target layer) 에 있는 모든 피처를 반환합니다.
- 대상 필드가 중복되는 값들을 담고 있을 경우, 이 피처들은 결합 레이어의 동일한 피처에 할당됩니다.
- 결합 필드가 중복되는 일치하는 값들을 담고 있을 경우, 맨 처음 가져온 피처만 선택합니다.

> **참고**
>
> QGIS 에서 결합은 일치하는 단일 필드에 기반하기 때문에 대부분의 경우 두 레이어의 두 필드들이 담고 있는 값들이 유일한지 확인해보는 편이 좋습니다.

QGIS 는 이 결합을 미세 조정할 수 있는 많은 옵션을 제공하고 있습니다:

- *Cache join layer in virtual memory*: 검색 속도를 향상시키기 위해, 메모리에 결합된 레이어의 (도형을 제외한) 값들을 저장할 수 있게 합니다.
- *Create attribute index on the join field*: 이 옵션을 활성화하면 검색 속도를 향상시키기 위해 결합 필드에 속성 인덱스를 생성합니다.
- *Dynamic form*: 결합 필드들을 *Target field* 를 따라서 실시간으로 동기화할 수 있습니다. 이 옵션을 활성화하면, 결합 필드에 있는 제약조건도 정확하게 업데이트됩니다. 이 옵션이 기본적으로 비활성화돼 있는 것은 피처나 결합이 많을 경우 처리 시간이 너무 오래 걸리기 때문이라는 사실을 기억하세요.
- 대상 레이어가 편집 가능 상태인 경우, 필드에 대한 정보를 알려주기 위한 몇몇 아이콘들이 속성 테이블의 필드 옆에 표시될 것입니다:
  - : 결합 레이어를 편집할 수 있도록 환경 설정되지 않았습니다. 대상 속성 테이블에서 결합 피처를 편집할 수 있도록 하고 싶다면, *Editable join layer* 옵션을 체크해야 합니다.
  - : 결합 레이어를 편집할 수 있도록 환경 설정돼 있지만, 현재 읽기 전용 상태입니다.
  - : 결합 레이어를 편집할 수 있지만, 동기화 메커니즘이 활성화되지 않았습니다. 대상 레이어에서 피처를 생성하면 결합 레이어에 피처를 자동적으로 추가하게 하려면, *Upsert on edit* 옵션을 체크해야 합니다. 마찬가지로, 대상 레이어에서 피처를 삭제하면 결합 레이어에서도 자동적으로 삭제하게 하려면 *Delete cascade* 옵션을 체크해야 합니다.
- *Joined fields*: 결합된 레이어의 모든 필드를 추가하는 대신, 하위 집합을 지정할 수 있습니다.
- *Custom field name prefix*: 이름의 충돌을 방지하기 위해 결합된 필드명에 사용자 지정 접두어를 추가합니다.

### 12.4.2 여러 레이어 간 관계 설정하기

두 레이어의 피처들 사이에 1 대 1 연결을 정의하는 결합과는 달리, 관계는 사용자가 두 개 이상의 레이어들의 여러 피처들 사이에 상호 연결 (interconnection) 을 작성할 수 있도록 해줍니다. 이와 같이, 관계란 프로젝트 수준의 설정이므로 *Project* → *Properties* → *Relations* 탭에서 설정합니다. 이 탭에서 다음을 설정할 수 있습니다:

- *Add relation*: 다음과 같은 유형의 관계를 추가할 수 있습니다:
  - *1* 대 다 관계
  - 다대다 관계
  - 액션 드롭다운 메뉴의 전용 도구들로 추가하거나 편집할 수 있는다형성 관계

> **참고**
>
> 비다형성 (non-polymorphic) 관계를 한번 생성하고 나면, 이 관계를 간단하게 편집할 수 있는 방법이 없습니다. 더블클릭으로 관계의 이름만 편집할 수 있을 뿐이죠. 이런 관계의 다른 모든 파라미터의 경우, 사용자가 제거한 다음 다시 생성해야만 할 것입니다.

- *Discover relations*: QGIS 는 지원하는 데이터베이스 포맷들 (PostgreSQL, 지오패키지, ESRI 파일 지오데이터베이스, …) 로부터 기존 관계를 발견할 수 있습니다. 이를 통해 관계를 보다 쉽게 정의할 수 있습니다.
- *Remove relation*: 현재 선택 집합에서 선택한 관계를 제거합니다.

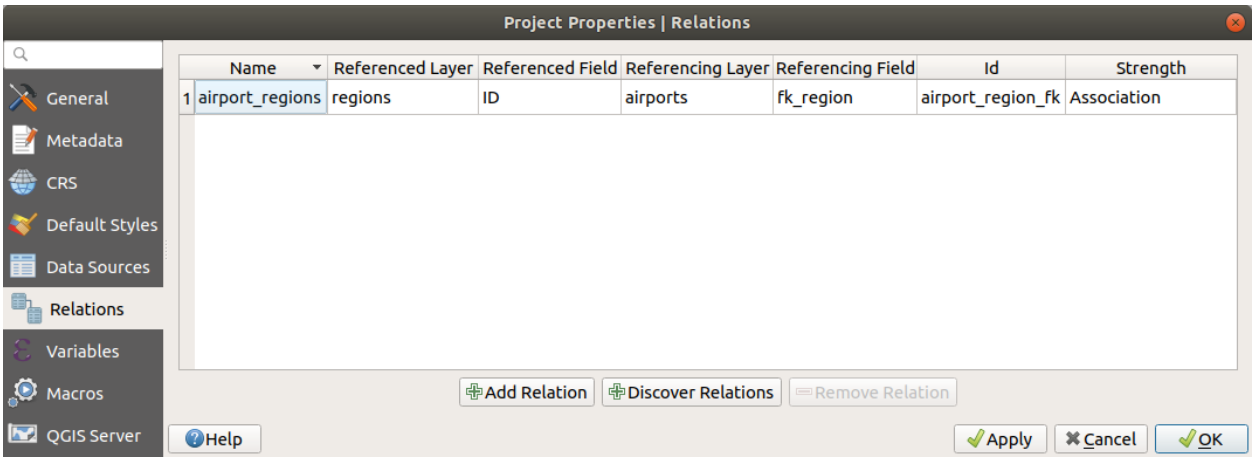



그림 12.105: 관계 탭

### 1 대다 (1-N) 관계

예를 들어 사용자에게 알래스카의 모든 지역을 담고 있고, 지역명, 지역 유형, (기본키로 동작하는) 유일 ID 관련 속성을 제공하는 (폴리곤) 레이어가 있다고 해보겠습니다.

그런데 알래스카 지역에 위치한 공항 관련 정보를 담고 있는 또다른 포인트 레이어 또는 테이블을 얻었고, 사용자가 이 공항들도 파악하고 싶다고 해보겠습니다. `regions` 레이어에 이 공항들을 추가하려면, 외래 키 (foreign key) 를 이용해서 일대다 관계를 생성해야 합니다. 왜냐하면 대부분의 지역에 공항이 몇 군데씩 있기 때문입니다.

#### 레이어와 키

QGIS 는 테이블과 벡터 레이어를 전혀 구분하지 않습니다. 기본적으로, 벡터 레이어는 도형을 지닌 테이블입니다. 따라서 사용자의 테이블을 벡터 레이어로 추가할 수 있습니다. 일대다 관계를 시연해보기 위해, 동일한 데이터셋에 있는 `regions` 및 `airports` 셰이프파일을 불러오겠습니다. 실제로는 각 공항이 딱 1 개 지역에만 속해 있는 반면 각 지역은 공항을 몇 개든 가지고 있을 수 있다는 뜻입니다. (전형적인 일대다 관계죠.)

공항을 설명하는 속성에 더해, 공항 레이어는 외래 키처럼 동작하는 또다른 `fk_region` 필드를 가지고 있습니다. (데이터베이스를 사용하는 경우, 이 외래 키 필드에 제약조건을 정의해놓는 편이 좋을 겁니다.) 이 `fk_region` 필드는 항상 지역의 ID 를 담고 있을 것입니다. 이 ID 를 해당 지역을 가리키는 포인터라고 볼 수도 있겠죠.

편집 작업을 위한 사용자 지정 편집 양식을 설계할 수 있도록 QGIS 에 레이어들 사이의 관계를 알려주면 끝입니다. QGIS 가 나머지 설정을 처리할 것입니다. 서로 다른 제공자와도 같은 방식으로 작업할 수 있습니다. (즉 셰이프파일 및 CSV 파일에도 동일하게 사용할 수 있습니다.)

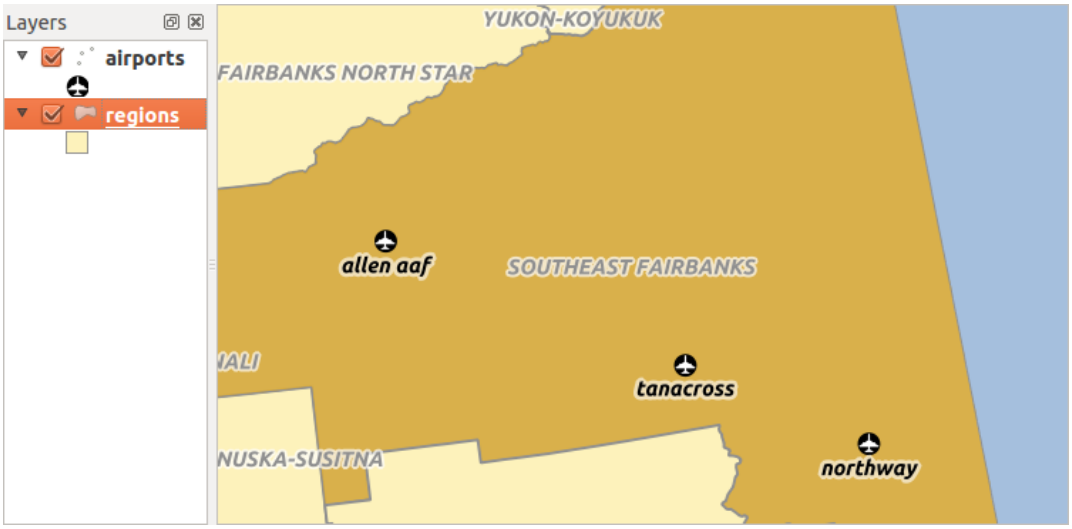



그림 12.106: 알래스카의 지역과 공항

### 1 대다 (1-N) 관계 정의하기

가장 먼저 할 일은 QGIS 에 두 레이어 간의 관계를 알려주는 것입니다. *Project* ▸ *Properties*…메뉴에서 이 작업을 할 수 있습니다. *Relations* 탭을 선택한 다음 *Add Relation* 버튼을 클릭하십시오.

- **이름 (Name)**: 제목으로 사용됩니다. 관계의 목적을 설명하는, 사람이 읽을 수 있는 문자열이어야 합니다. 이 예시에서는 그냥 **airport_relation** 라고 하겠습니다.
- **참조되는 레이어 (Referenced Layer) (부모)**: 부모 레이어로도 간주되는 이 레이어는 외래 키가 가리키는 기본 키를 가지고 있습니다. 즉 이 예시에서는 `regions` 레이어입니다. 참조되는 레이어의 기본 키를 정의해야 하는데, 이 예시에서는 `ID` 입니다.
- **참조하는 레이어 (Referencing Layer) (자식)**: 자식 레이어로도 간주되는 이 레이어가 외래 키 필드를 담고 있습니다. 이 예시에서는 `airports` 레이어입니다. 이 레이어의 경우 다른 레이어를 가리키는 참조하는 필드를 추가해야 합니다. 이 예시에서는 `fk_region` 입니다.

> **참고**
>
> 어떤 경우, 레이어에 있는 피처들을 유일하게 식별하기 위해 단일 필드 이상이 필요할 때가 있습니다. 이런 레이어와의 관계를 생성하는 데에는 **복합 키 (composite key)**, 예를 들어 일치하는 필드들의 단일 쌍 이상이 필요합니다. Add new field pair as part of a composite foreign key 버튼을 이용해서 필요한만큼 많은 쌍을 추가하십시오.

- **ID**: 내부 목적으로 사용되며, 유일한 값이어야 합니다. 사용자 지정 양식 을 작성하기 위해 필요할 수도 있습니다. 이 항목을 비워둘 경우, 자동적으로 생성되지만 사용자가 더 쉽게 다룰 수 있는 ID 를 직접 할당할 수도 있습니다.
- **관계 강도 (Relationship Strength)**: 상위 (부모) 와 하위 (자식) 레이어 사이의 관계 강도를 설정합니다. 기본 *Association* 유형은 상위 레이어가 하위 레이어에 단순하게 링크되어 있다는 의미인 반면, *Composition* 유형은 상위 피처를 복제할 때 하위 피처도 복제할 수 있으며 상위 레이어의 피처를 삭제하면 하위 피처도 삭제할 수 있는 등, 모든 수준에서 순차적 (cascade) 으로 결과물을 산출합니다. (다시 말해 하위의 하위의…하위 피처도 삭제할 수 있다는 뜻입니다.)

*Relations* 탭에서, *Discover Relation* 버튼을 눌러 불러온 레이어들의 제공자로부터 사용할 수 있는 관계를 가져올

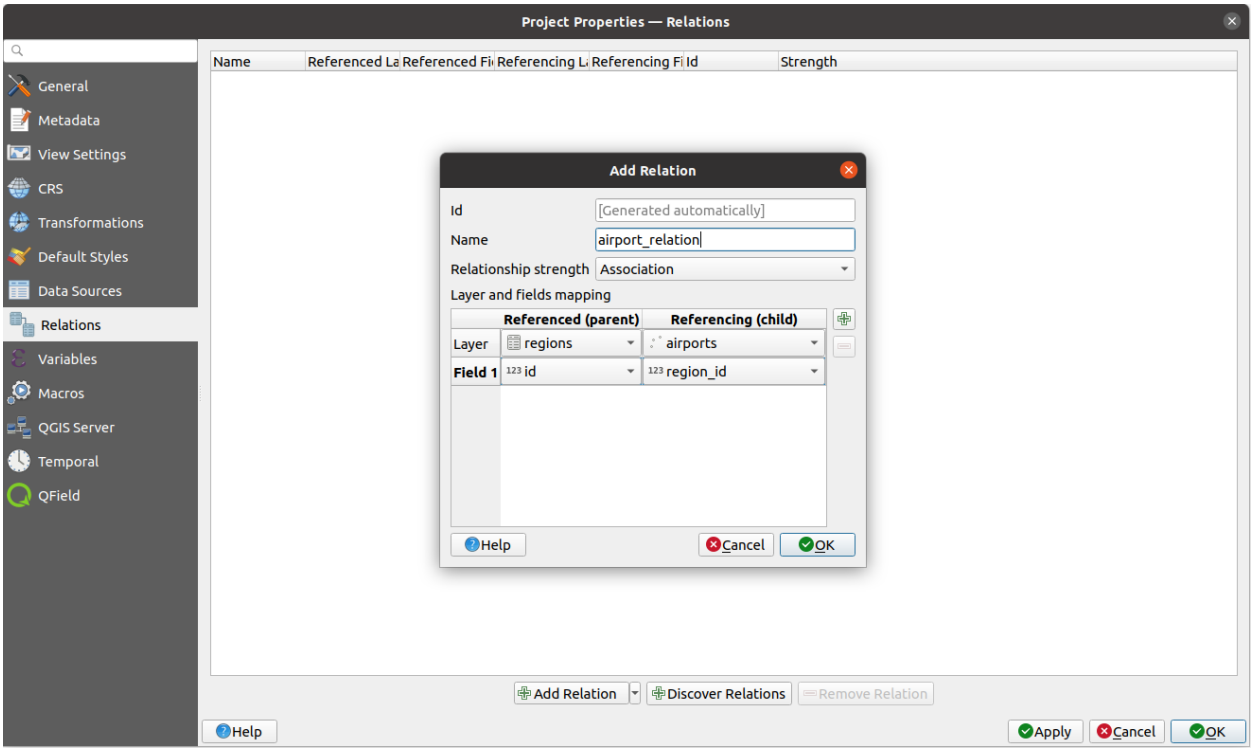



그림 12.107: 지역과 공항 레이어 사이에 관계 추가하기

수도 있습니다. 이 기능은 PostgreSQL 또는 SpatiaLite 같은 데이터 제공자에 저장된 레이어에 대해서 사용할 수 있습니다.

### 1 대다 (1-N) 관계용 양식

이제 QGIS 가 관계를 인지했으니, 관계를 이용해서 QGIS 가 생성하는 양식을 향상시킬 것입니다. 기본 양식 생성 방법 (자동 생성) 을 변경하지 않았으므로, 양식에 위젯 하나만 추가할 것입니다. 범례에서 `regions` 레이어를 선택한 다음 식별 도구를 써보십시오. 사용자 설정에 따라 양식이 바로 열릴 수도 있고, 식별 대화창의 액션을 통해 양식을 열어야 할 수도 있습니다.

그림에서 보이듯이, 이 특정 지역에 할당된 공항이 테이블에 전부 표시됩니다. 또 사용할 수 있는 버튼도 몇 개 있습니다. 이 버튼들을 간단히 설명하겠습니다.

- 버튼은 편집 모드를 켜고 끕니다. 현재 `regions` 레이어에 있는 피처의 피처 양식을 보고 있긴 하지만, 이 버튼은 `airports` 레이어의 편집 모드를 켜고 끈다는 점을 조심하십시오. 왜냐하면 테이블이 `airports` 레이어의 피처를 표시하고 있기 때문입니다.
- 버튼은 하위 레이어 (`airports`) 의 모든 편집 내용을 저장합니다.
- 버튼은 맵 캔버스에 공항 (`airports`) 도형을 디지타이즈할 수 있게 해주며, 기본적으로 현재 지역 (`regions`) 에 새 피처를 할당합니다. 도형 유형에 따라 아이콘이 바뀔 것이라는 사실을 기억하십시오.
- 버튼은 `airports` 레이어의 속성 테이블에 새 레코드를 추가하고, 기본적으로 현재 지역 (`regions`) 에 새 피처를 할당합니다. 나중에 *Add part* 디지타이즈 작업 도구로 도형을 그릴 수 있습니다.
- 버튼은 하위 레이어 안에 있는 하나 이상의 하위 피처를 복사하고 붙여넣기할 수 있습니다. 나중에 서로 다른 상위 피처에 이 하위 피처를 할당하거나 또는 하위 피처의 속성을 수정할 수 있습니다.

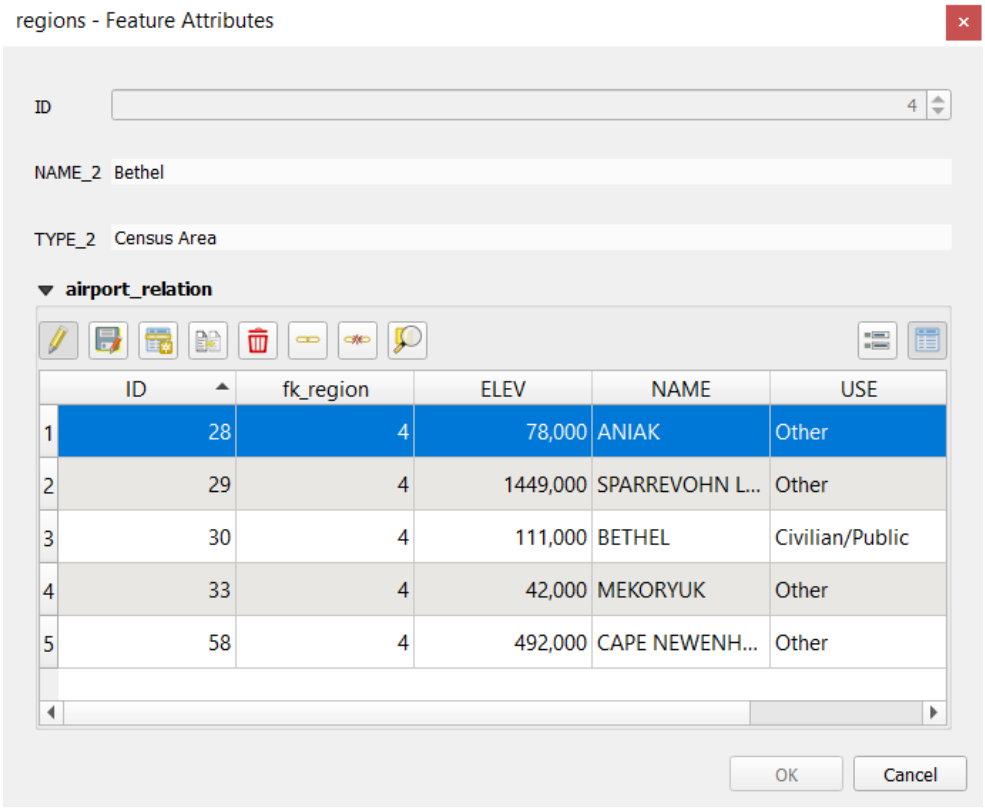



그림 12.108: “airports”와 관계가 생성된“regions”의 식별 대화창

- 버튼은 선택한 공항 (들) 을 영구 삭제합니다.
- 심볼은 나중에 현재 지역에 할당될 어떤 공항이든 선택할 수 있는 새 대화창을 엽니다. 사용자가 실수로 잘못된 지역에 공항을 생성한 경우 편리할 수도 있습니다.
- 심볼은 현재 지역에서 선택한 공항 (들) 을 링크 해제합니다. 즉 선택한 공항 (들) 을 실질적으로 할당되지 않은 상태로 남겨둔다는 (외래 키를 NULL 로 설정한다는) 뜻입니다.
- 버튼을 사용하면 선택한 자식 피처로 맵을 확대/축소할 수 있습니다.
- 오른쪽에 있는 및 두 버튼은 관련된 하위 피처들의테이블 뷰와 양식 뷰 를 전환합니다.

`regions` 피처에드래그 & 드롭 설계자 를 사용하는 경우, 어떤 도구를 사용할 수 있는지 선택할 수 있습니다. *Force hide form on add feature* 옵션을 이용해서 새 피처를 추가할 때 새 양식을 열지 여부를 결정할 수도 있습니다. 이 옵션이 정확하게 동작하기 위해 NULL 값을 가진 속성이 무결한 기본값을 반드시 가져오지 않아도 된다는 사실을 기억하십시오.

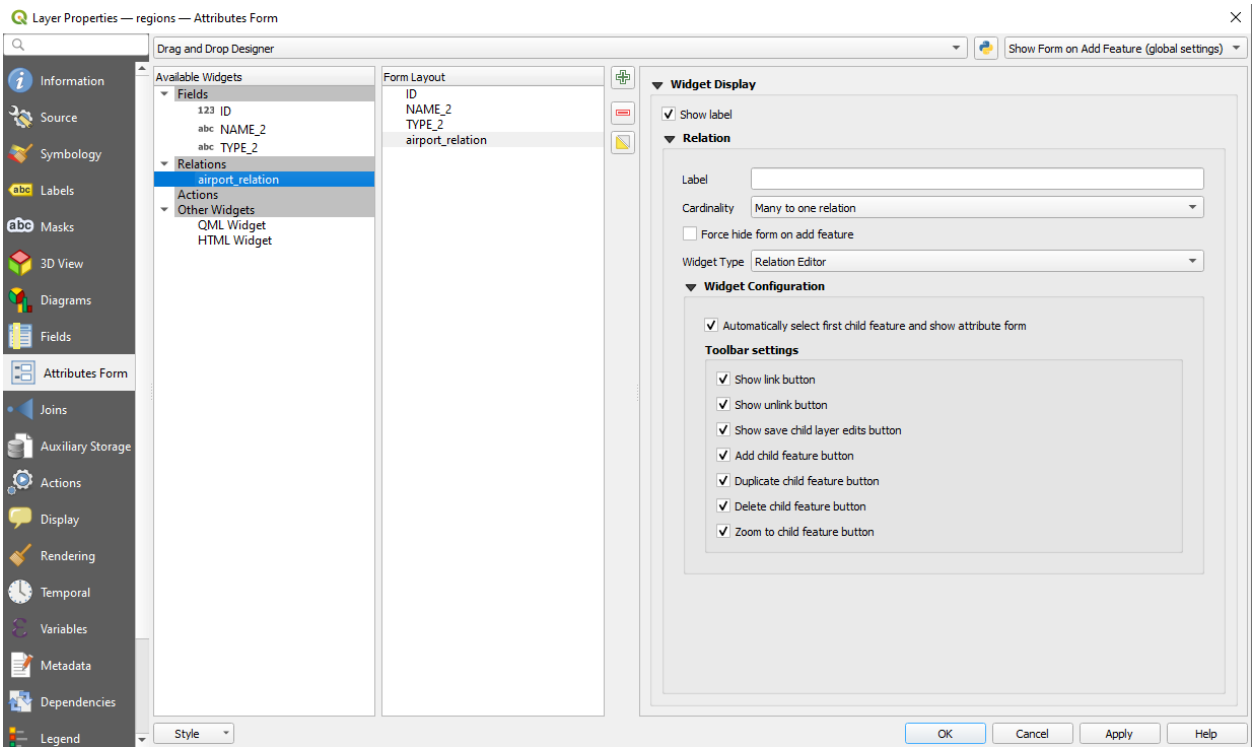



그림 12.109: 지역-공항 관계를 환경 설정하기 위한 드래그 & 드롭 설계자 도구

앞의 예시에서 참조하는 레이어가 도형을 보유하고 있기 (즉 그저 문자와 숫자를 조합한 테이블이 아니기) 때문에, 앞의 단계를 수행하면 레이어 속성 테이블에 대응 도형 피처가 없는 항목을 생성할 것입니다. 도형을 추가하려면:

1. *Open Attribute Table*: 참조하는 레이어의 속성 테이블을 여십시오.
2. 참조되는 레이어의 피처 양식 안에서 이전에 추가된 레코드를 선택하십시오.
3. Add Part 디지타이즈 작업 도구를 사용해서 선택한 속성 테이블 레코드에 도형을 첨부하십시오.

공항 테이블을 작업하는 경우, (관계를 생성하기 위해 사용된) `fk_region` 필드를 대상으로 관계 참조 위젯을 자동으로 설정합니다. 관계 참조 위젯 을 참조하세요.

공항 양식의 `fk_region` 필드 오른쪽에 버튼이 있을 겁니다. 이 버튼을 클릭하면 지역 레이어의 양식이 열릴 것입니다. 이 위젯에서 링크된 부모 피처의 양식을 쉽고 빠르게 열 수 있습니다.

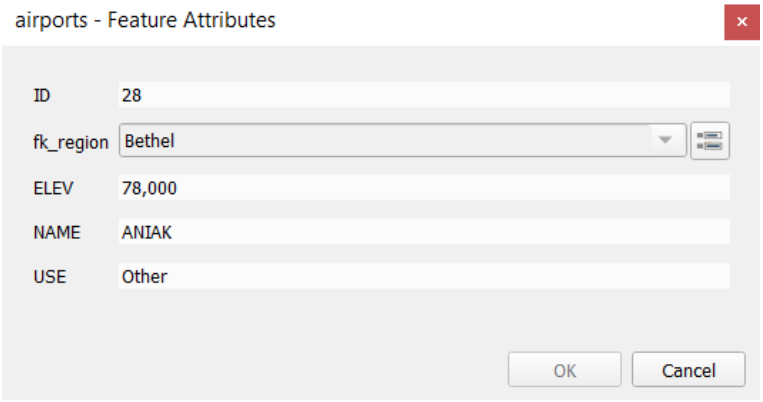
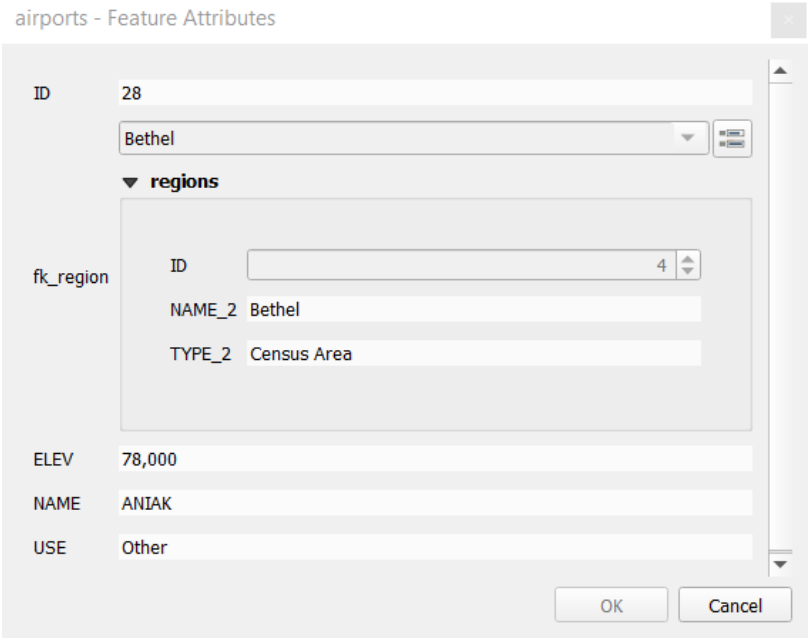



그림 12.110: "regions"와 관계가 생성된"airports"의 식별 대화창

관계 참조 위젯은 자식 레이어의 양식 안에 부모 레이어의 양식을 삽입할 수 있는 옵션도 가지고 있습니다. 공항 레이어의 *Properties* ▸ *Attributes Form* 메뉴를 선택하고, `fk_region` 필드를 선택한 다음 `Show embedded form` 옵션을 활성화하십시오.

지금 피처 대화창을 보고 있다면, 지역의 양식이 공항 양식 안에 삽입돼 있고 현재 공항을 다른 지역으로 할당할 수 있는 콤보박스도 있다는 것을 알 수 있을 겁니다.




게다가 공항 레이어의 편집 모드를 켜고 끌 경우, `fk_region` 필드도 자동 완성 기능을 지원합니다. 사용자가 입력하는 중에 지역 레이어의 `id` 필드의 모든 값을 보게 될 것입니다. 이때 공항 레이어의 *Properties* ▸ *Attributes Form* 메뉴에서 `Allow adding new features` 옵션을 선택했다면 버튼을 눌러 폴리곤을 디지타이징할 수 있습니다.

자식 피처의 속성을 기반으로 부모 레이어의 피처를 선택하기 위해값으로 피처 선택 도구에서 자식 레이어를 사용할 수도 있습니다.

그림 12.111 에서, 공항의 평균 해발고도가 500m 를 초과하는 모든 지역을 선택합니다.
이 양식에서 여러 서로 다른 집계 함수들을 사용할 수 있을 것입니다.

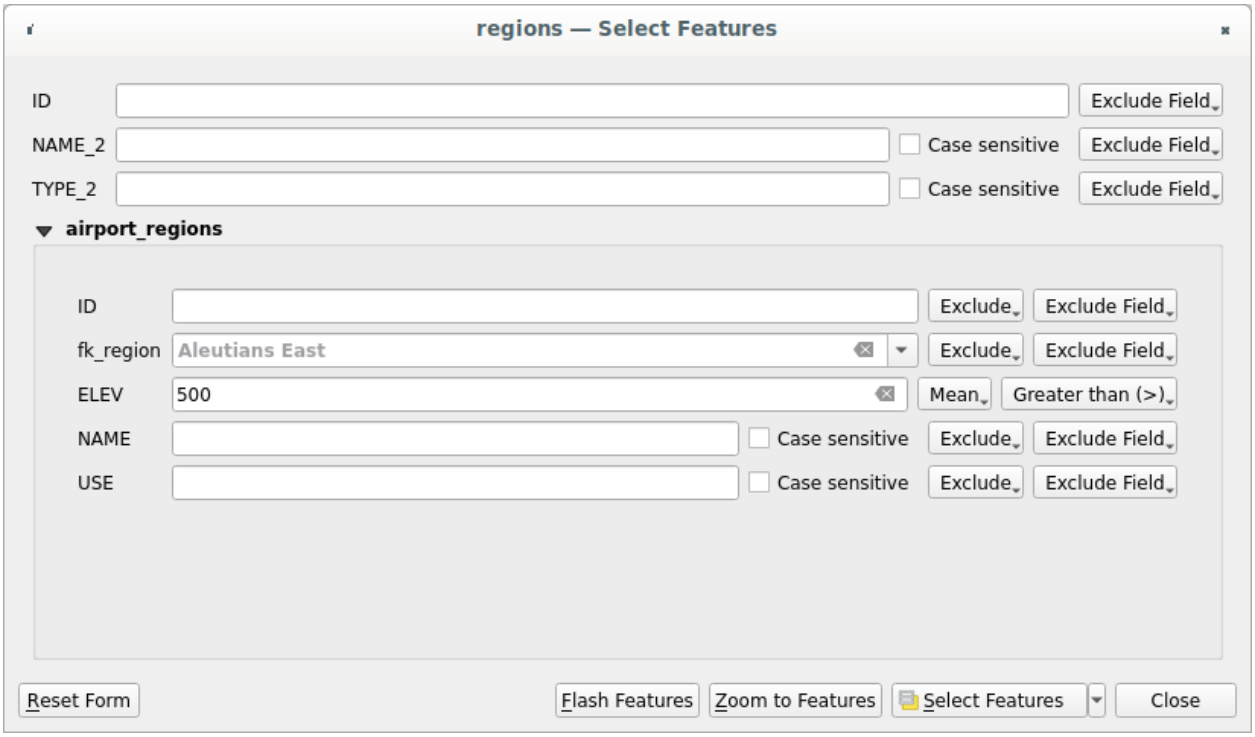



그림 12.111: 자식 값으로 부모 피처 선택

### 다대다 (N-M) 관계

N-M 관계란 두 테이블 간의 다대다 관계를 말합니다. `airports` 와 `airlines` 레이어를 예로 들 수 있습니다. 공항은 여러 항공사의 노선을 받아들이고 항공사는 여러 공항에 이착륙합니다.

This SQL code creates the three tables we need for an N-M relationship in a PostgreSQL schema named *locations*. You can run the code using the *Database → DB Manager…* for PostgreSQL or external tools such as pgAdmin. The airports table stores the `airports` layer and the airline table stores the `airlines` layer. In both tables few fields are used for clarity. The *tricky* part is the `airports_airlines` table. We need it to list all airlines for all airports (or vice versa). This kind of table is known as a *pivot table*. The *constraints* in this table force that an airport can be associated with an airline only if both already exist in their layers.

```
CREATE SCHEMA locations;

CREATE TABLE locations.airports
(
   id serial NOT NULL,
   geom geometry(Point, 4326) NOT NULL,
   airport_name text NOT NULL,
   CONSTRAINT airports_pkey PRIMARY KEY (id)
);

CREATE INDEX airports_geom_idx ON locations.airports USING gist (geom);
```

(다음 페이지에 계속)

(이전 페이지에서 계속)

```
CREATE TABLE locations.airlines
(
   id serial NOT NULL,
   geom geometry(Point, 4326) NOT NULL,
   airline_name text NOT NULL,
   CONSTRAINT airlines_pkey PRIMARY KEY (id)
);

CREATE INDEX airlines_geom_idx ON locations.airlines USING gist (geom);

CREATE TABLE locations.airports_airlines
(
   id serial NOT NULL,
   airport_fk integer NOT NULL,
   airline_fk integer NOT NULL,
   CONSTRAINT airports_airlines_pkey PRIMARY KEY (id),
   CONSTRAINT airports_airlines_airport_fk_fkey FOREIGN KEY (airport_fk)
      REFERENCES locations.airports (id)
      ON DELETE CASCADE
      ON UPDATE CASCADE
      DEFERRABLE INITIALLY DEFERRED,
   CONSTRAINT airports_airlines_airline_fk_fkey FOREIGN KEY (airline_fk)
      REFERENCES locations.airlines (id)
      ON DELETE CASCADE
      ON UPDATE CASCADE
      DEFERRABLE INITIALLY DEFERRED
 );
```

PostgreSQL 대신 GeoPackage 도 사용할 수 있습니다. 이 경우, 사용자가 *Database → DB Manager*…메뉴를 통해 세 테이블을 직접 생성할 수 있습니다. GeoPackage 에는 스키마가 존재하지 않기 때문에 접두어 *locations* 을 붙일 필요도 없습니다.

*Table → Create Table*…또는 *Table → Edit Table*…메뉴로는 `airports_airlines` 테이블의 외래 키 제약조건을 생성할 수 없기 때문에, *Database → SQL Window*…를 사용해서 생성해야 합니다. GeoPackage 는 *ADD CONSTRAINT* 선언문을 지원하지 않으므로 `airports_airlines` 테이블을 다음 두 단계를 거쳐 생성해야 합니다:

1. *Table → Create Table*…메뉴를 사용해서 `id` 필드만 보유한 테이블을 설정합니다.
2. *Database → SQL Window*…메뉴로 SQL 창을 열고, 다음 SQL 코드를 입력하고 실행합니다:

```
ALTER TABLE airports_airlines
   ADD COLUMN airport_fk INTEGER
   REFERENCES airports (id)
   ON DELETE CASCADE
   ON UPDATE CASCADE
   DEFERRABLE INITIALLY DEFERRED;

ALTER TABLE airports_airlines
   ADD COLUMN airline_fk INTEGER
   REFERENCES airlines (id)
   ON DELETE CASCADE
   ON UPDATE CASCADE
   DEFERRABLE INITIALLY DEFERRED;
```

그 다음 QGIS 에서, 앞에서 설명한 대로일대다 *(1-N)* 관계 2 개를 설정해야 합니다:

- `airlines` 테이블과 피벗 테이블 간의 관계
- `airports` 테이블과 피벗 테이블 간의 두 번째 관계

*Project* → *Properties* → *Relations* 메뉴에서 *Discover Relations* 메뉴 옵션을 사용하면 (PostgreSQL 인 경우에만) 이 작업을 더 쉽게 할 수 있습니다. QGIS 가 사용자 데이터베이스의 모든 관계를 자동으로 읽어들여, 사용자는 필요한 두 관계만 선택하면 됩니다. 먼저 QGIS 프로젝트에 세 테이블을 불러와야 한다는 것을 잊지마세요.

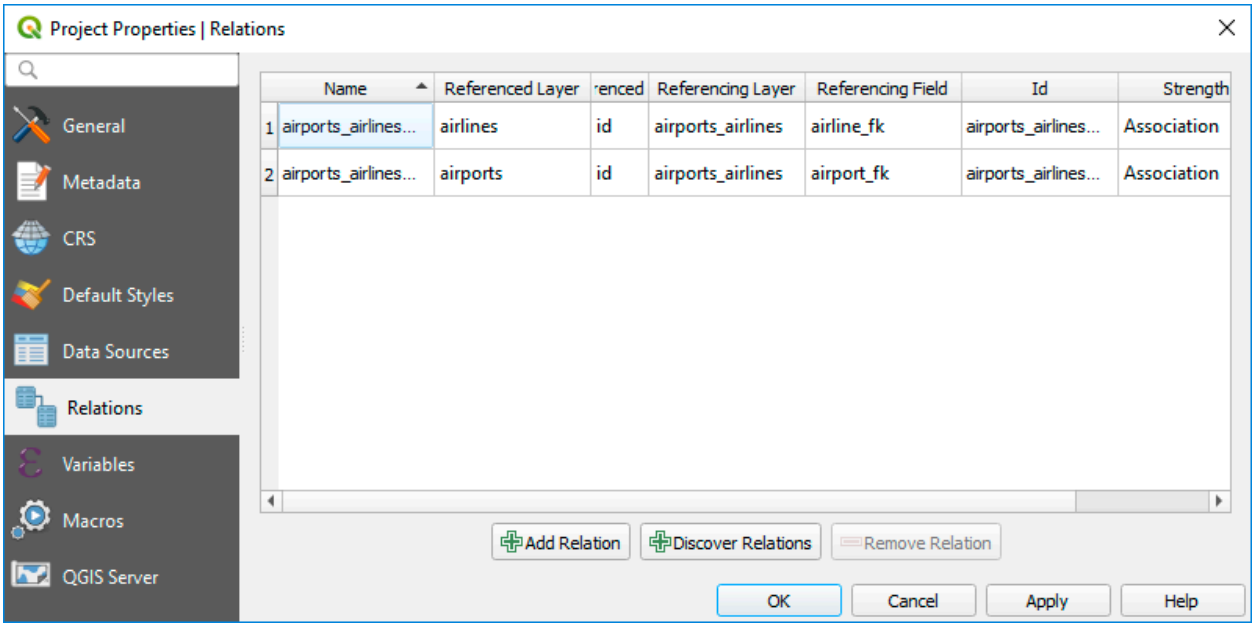



그림 12.112: 관계와 자동검색

사용자가 `airport` 또는 `airline` 을 제거하고자 하는 경우, QGIS 는 `airports_airlines` 테이블에서 관련 레코드 (들) 를 제거하지 않을 것입니다. 현재 예시에서처럼 피벗 테이블 생성 시 올바른 제약조건 을 지정한 경우 데이터베이스가 관련 레코드 (들) 를 제거할 것입니다.

> **참고**
>
> **다대다 관계를 자동 트랜잭션 (transaction) 그룹과 결합하기**
>
> 이런 맥락에서 작업하는 경우 *Project Properties* → *Data Sources* → 메뉴 옵션에서 트랜잭션 모드를 활성화시켜야 합니다. QGIS 가 모든 테이블 (`airlines`, `airports` 및 피벗 테이블) 에 있는 행 (들) 을 추가하거나 업데이트할 수 있어야 하기 때문입니다.

마지막으로 *Layer Properties* → *Attributes Form* 메뉴에서 `airports` 및 `airlines` 레이어에 대한 올바른 기수 (cardinality) 를 선택해야 합니다. 첫 번째 기수로 **airlines (id)** 옵션을, 두 번째 기수로는 **airports (id)** 옵션을 선택해야 합니다.

이제 *Add child feature* 또는 하위 양식에 있는 *Link existing child feature* 를 통해 공항을 항공사와 (또는 항공사를 공항과) 관련시킬 수 있습니다. 레코드가 `airports_airlines` 테이블에 자동적으로 삽입될 것입니다.

> **참고**
>
> **다대일 관계** 기수성 사용하기
>
> N-M 관계에서 피벗 테이블을 숨기는 것이 바람직하지 않을 때가 있습니다. 주로 관계가 확정된 후에야 값을 가질 수 있는 관계에 있는 속성이 존재하기 때문입니다. 사용자 테이블이 도형 필드를 보유하고 있는 경우 피벗 테이블의 외래 키 필드에 대해 *On map identification* 옵션을 (*Layer Properties* → *Attributes Form* → *Available widgets* → *Fields*) 활성화시켜보면 재미있을 수도요.

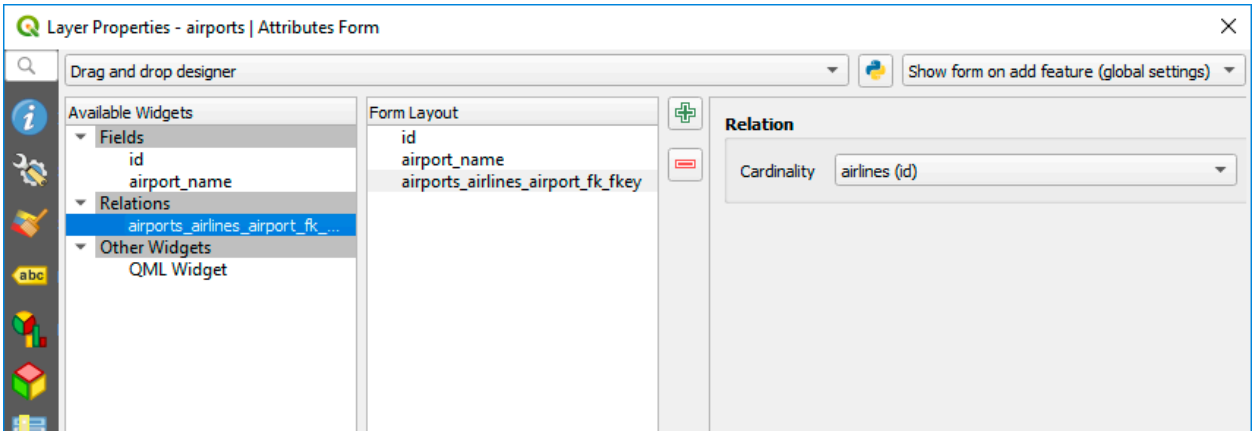



그림 12.113: 관계 기수성 설정

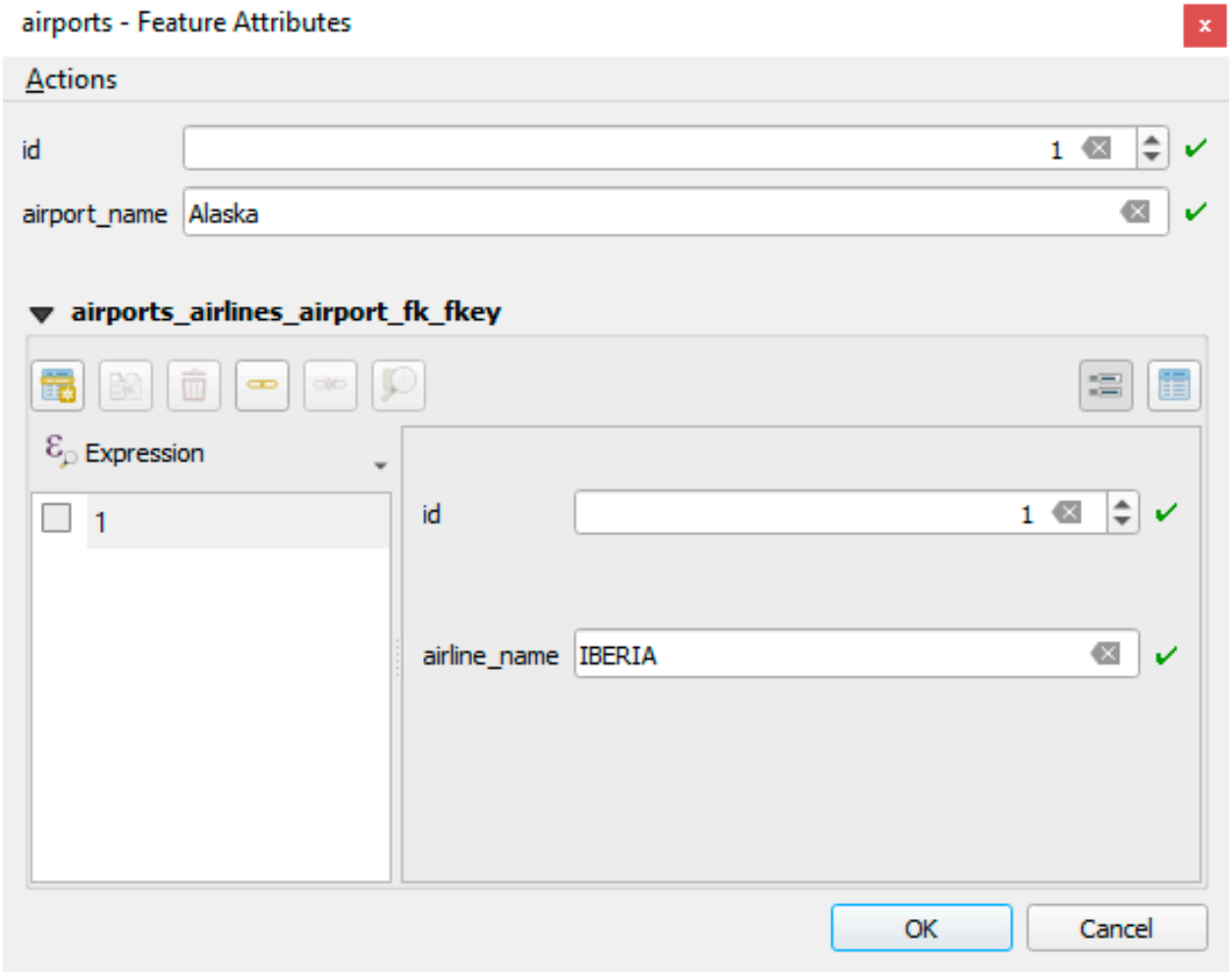



그림 12.114: 공항과 항공사 사이의 N-M 관계

> **참고**
>
> **피벗 테이블의 기본 키**
>
> 피벗 테이블의 기본 키에 여러 필드를 사용하는 일을 피하십시오. QGIS 는 단일 기본 키를 가정하기 때문에 `constraint airports_airlines_pkey primary key (airport_fk, airline_fk)` 같은 제약조건은 작동하지 않을 것입니다.

### 다형성 관계

#### 목적

다형성 관계 (polymorphic relation) 는 1 대다 (1-N) 관계의 특례로, 단일 (문서) 참조 레이어가 참조되는 레이어 여러 개의 피처를 담고 있는 경우입니다. 서로 다른 참조하는 레이어가 각각 참조되는 레이어를 필요로 하는 일반적인 관계와는 다릅니다. (문서) 참조하는 레이어에 참조되는 레이어를 식별하기 위한 정보를 저장하는 추가적인 `layer_field` 열을 추가하면 단일 (문서) 참조 레이어를 생성할 수 있습니다. 가장 단순한 형태의 경우, (문서) 참조 레이어는 그냥 이 필드에 참조되는 레이어의 레이어 이름을 삽입할 것입니다.

좀 더 정확하게 말하자면, 다형성 관계란 참조하는 레이어는 동일하지만 참조되는 레이어는 동적으로 정의되는 일반적인 관계들의 집합입니다. 레이어의 다형성 설정은 참조되는 레이어의 테이블 이름, 레이어 ID, 레이어 이름 같은 일부 속성과 일치해야 하는 표현식을 이용해서 정의할 수 있습니다.

공원에 방문해서 그곳에서 볼 수 있는 다양한 `plants` 와 `animals` 종들의 사진을 찍는다고 상상해보십시오. 각 식물 또는 동물과 관련된 사진이 여러 장일 것이기 때문에, 일반적인 일대다 관계를 이용해서 사진을 저장하는 경우 `animal_images` 와 `plant_images`, 2 개의 개별 테이블이 필요할 것입니다. 테이블이 2 개라면 문제가 되지 않을 수도 있겠지만, 버섯이나 새 등등에 대해서도 개별적으로 사진을 찍기를 원한다면 어떨까요.

Polymorphic relations solve this problem as all the referencing features are stored in the same table `documents`. For each feature the referenced layer is stored in the `referenced_layer` field and the referenced feature id in the `referenced_id` field.

#### 다형성 관계 정의하기

먼저, *Project* → *Properties*⋯메뉴를 통해 QGIS 에 레이어들 간의 다형성 관계가 존재한다는 사실을 알리십시오. *Relations* 탭으로 가서 *Add Relation* 버튼 옆에 있는 아래 방향 화살표를 클릭하면, 새로 나타나는 드롭다운 목록에서 *Add Polymorphic Relation* 옵션을 선택할 수 있습니다.

- **ID**: 내부 목적으로 사용되며, 유일한 값이어야 합니다. 사용자 지정 양식 을 작성하기 위해 필요할 수도 있습니다. 이 항목을 비워둘 경우, 자동적으로 생성되지만 사용자가 더 쉽게 다룰 수 있는 ID 를 직접 할당할 수도 있습니다.
- **Referencing Layer (Child)** also considered as child layer, is the one with the foreign key field on it. In our case, this is the `documents` layer. For this layer you need to add a referencing field which points to the other layer, so this is `referenced_id`.

> **참고**
>
> 어떤 경우, 레이어에 있는 피처들을 유일하게 식별하기 위해 단일 필드 이상이 필요할 때가 있습니다. 이런 레이어와의 관계를 생성하는 데에는 **복합 키 (composite key)**, 예를 들어 일치하는 필드들의 단일 쌍 이상이 필요합니다. Add new field pair as part of a composite foreign key 버튼을 이용해서 필요한만큼 많은 쌍을 추가하십시오.

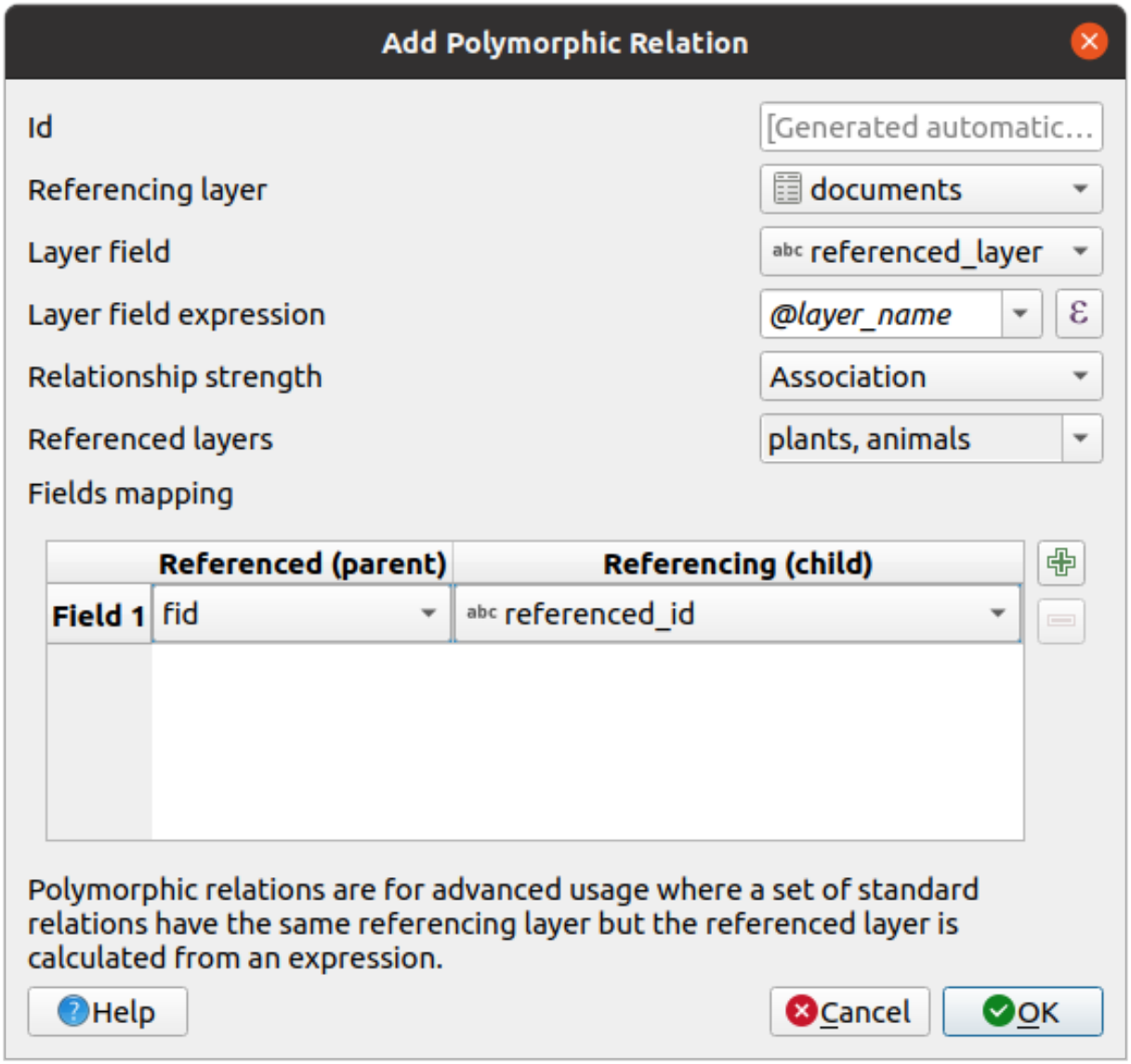



그림 12.115: `documents` 레이어를 참조하는 레이어로, `animals` 와 `plants` 레이어를 참조되는 레이어로 이용해서 다형성 관계를 추가합니다.

- **레이어 필드**: 평가된 레이어 표현식의 결과를 저장하는, 해당 피처가 속해 있는 참조하는 테이블인, 참조하는 테이블에 있는 필드입니다. 이번 예시에서는 `referenced_layer` 필드입니다.
- **레이어 표현식**: 이 표현식의 평가 결과가 레이어의 유일 식별자가 됩니다. 레이어 이름 `@layer_name`, 레이어 ID `@layer_id`, 레이어의 테이블 이름 `decode_uri(@layer, 'table')`, 또는 레이어를 유일하게 식별할 수 있다면 무엇이든 가능합니다.
- **관계 강도 (Relationship Strength)**: 상위 (부모) 와 하위 (자식) 레이어 사이에 생성된 관계의 강도를 설정합니다. 기본 *Association* 유형은 상위 레이어가 하위 레이어에 단순하게 링크되어 있다는 의미인 반면, *Composition* 유형은 상위 피처를 복제할 때 하위 피처도 복제할 수 있으며 상위 레이어의 피처를 삭제하면 하위 피처도 삭제할 수 있는 등, 모든 수준에서 순차적 (cascade) 으로 결과물을 산출합니다. (다시 말해 하위의 하위의…하위 피처도 삭제할 수 있다는 뜻입니다.)
- **참조되는 레이어**: 상위 (부모) 레이어로도 간주되는 참조된 레이어는 기본 키 (primary key) 가 가리키는 레이어이기 때문에, 이 예시에서는 `plants` 및 `animals` 레이어입니다. 드롭다운 메뉴에서 참조되는 레이어의 기본 키를 정의해야 합니다. 이 예시에서는 `fid` 입니다. 무결한 기본 키를 정의하려면 참조되는 레이어 모두가 이 이름을 가진 필드를 가지고 있어야 한다는 사실을 기억하십시오. 참조되는 레이어 가운데 하나에라도 이 이름을 가진 필드가 없다면, 다형성 관계를 저장할 수 없습니다.

다형성 관계를 추가하고 나면, *Edit Polymorphic Relation* 메뉴 항목을 통해 다형성 관계를 편집할 수 있습니다.

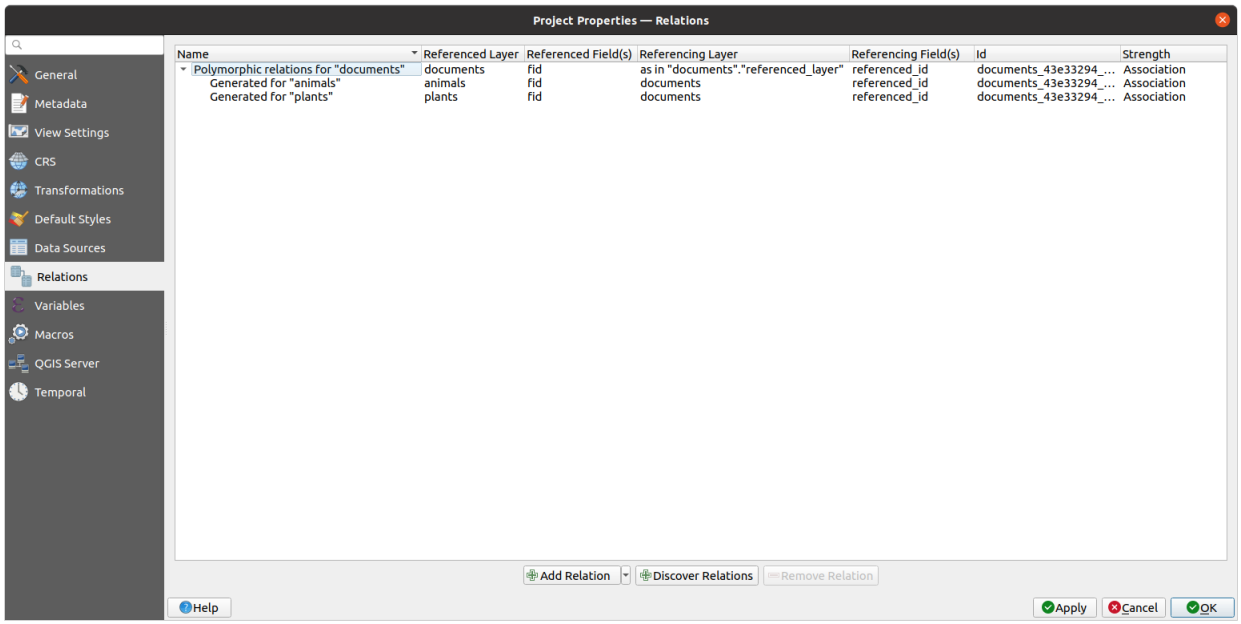



그림 12.116: 새로 생성한 다형성 관계와 이 관계의 동식물에 대한 하위 (자식) 관계의 미리보기

이 예시는 다음과 같은 데이터베이스 스키마를 이용합니다:

```
CREATE SCHEMA park;

CREATE TABLE park.animals
(
   fid serial NOT NULL,
   geom geometry(Point, 4326) NOT NULL,
   animal_species text NOT NULL,
   CONSTRAINT animals_pkey PRIMARY KEY (fid)
);

CREATE INDEX animals_geom_idx ON park.animals USING gist (geom);
```

(다음 페이지에 계속)

(이전 페이지에서 계속)

```
CREATE TABLE park.plants
(
   fid serial NOT NULL,
   geom geometry(Point, 4326) NOT NULL,
   plant_species text NOT NULL,
   CONSTRAINT plants_pkey PRIMARY KEY (fid)
);

CREATE INDEX plants_geom_idx ON park.plants USING gist (geom);

CREATE TABLE park.documents
(
   fid serial NOT NULL,
   referenced_layer text NOT NULL,
   referenced_id integer NOT NULL,
   image_filename text NOT NULL,
   CONSTRAINT documents_pkey PRIMARY KEY (fid)
);
```

Once relations are established, the *Identify Features* result panel will display related features, allowing you to view associated data directly.

### Dataset-stored Relationships

Other than the above mentioned relationships that are created in the project properties and apply only in the active project, QGIS can also read database-level relations, i.e., connections defined between layers in the dataset. This is supported for database providers such as PostgreSQL, SpatiaLite, GeoPackage, ESRI FileGeodatabase. When such parent and child layers are loaded in a project, QGIS can automatically detect the relation settings and fill appropriately the *Project* → *Project properties* → *Relations* tab. This can also be triggered using the *Discover relations* button.

For some data types (e.g., GeoPackage, SpatiaLite or ESRI FileGeodatabase), QGIS also allows to create and edit the relations between the tables in the database. This can be managed through the *Browser Panel*. From the *Browser panel*, right-click on the database entry, select *New relationship…*, and in the new dialog, fill the following options:

- *Name*: set the name of the new relationship.
- *Cardinality*: define the cardinality of the relationship.
- *Strength*: specify the strength of the relationship.
- Under the *Tables* group, set your *Parent* and *Child* tables.
- Under the *Fields* group, set your *Parent field* and *Child field*.
- Under the *Advanced* section, depending on the database type, you can set:
  - *Related table type*:
    - *features*: for standard feature tables.
    - *media*: for tables storing media files.
    - *simple_attributes*: for tables with simple attribute data.
    - *attributes*: for tables with complex attribute data.
    - *tiles*: for tables storing tile data.

- *Forward label*: a name to uniquely identify the relationship when navigating from the parent table to the child table.
- *Reverse label*: a name to uniquely identify the relationship when navigating from the child table to the parent table.

Applying the dialog will add a *Relationships* node under the database entry, with the new relationship as child item.

Editing or deleting existing relationships can be done by right-clicking on the relationship in the Browser Panel.

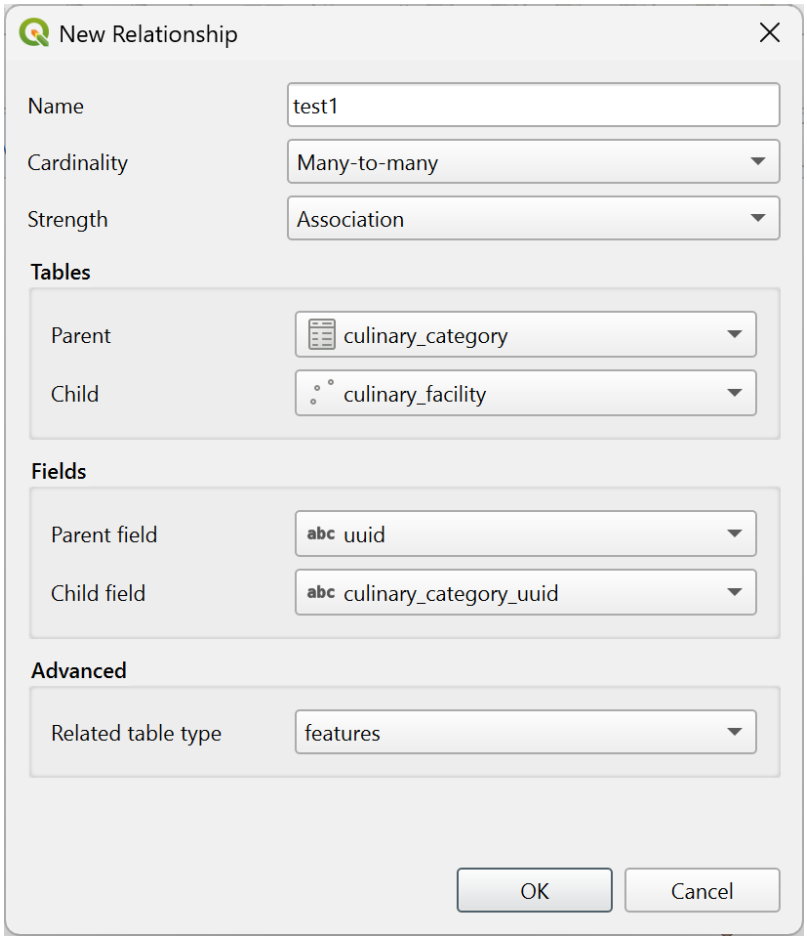



그림 12.117: Managing dataset-stored relationships in the Browser Panel

### 12.4.3 Storing and fetching an external resource

A field may target a resource stored on an external storage system. Attribute forms can be configured so they act as a client to an external storage system in order to store and fetch those resources, on users demand, directly from the forms.

#### Configuring an external storage

In order to setup an external storage, you have to first configure it from the vector *attribute form properties* and select the *Attachment* widget.




그림 12.118: Editing a WebDAV external storage for a given field

From the *Attachment* widget, you have to first select the *Storage type*:

- *Select Existing File*: The target URL already exists. When you select a resource, no store operation is achieved, the attribute is simply updated with the URL.
- *Simple Copy*: Stores a copy of the resource on a file disk destination (which could be a local or network shared file system) and the attribute is updated with the path to the copy.
- *WebDAV Storage*: The resource is pushed to a HTTP server supporting the WebDAV protocol and the attribute is updated with its URL. Nextcloud, Pydio or other file hosting software support this protocol.
- *AWS S3*: The resource is pushed to a server supporting AWS Simple Storage Service protocol and the attribute is updated with its URL. Amazon Web Service and MinIO hosting software support this protocol.

Then, you have to set up the *Store URL* parameter, which provides the URL to be used when a new resource needs to be stored. It's possible to set up an expression using the *data defined override widget* in order to have specific values according to feature attributes.

The variable **@selected_file_path** could be used in that expression and represent the absolute file path of the user selected file (using the file selector or drag'n drop).

참고

Using the **WebDAV** or **AWS S3** external storage, if the URL ends with a "/", it is considered as a folder and the selected file name will be appended to get the final URL.

If the external storage system needs to, it's possible to configure an *authentication*.

참고

To use the **AWS S3** external storage, you must use an **AWS S3** authentication type.

### Using an external storage

Once configured, you can select a local file using the button ⋯ when editing a feature's attribute. Depending on the configured *storage type*, the file will be stored on the external storage system (except if *Select existing file* has been selected) and the field will be updated with the new resource URL.

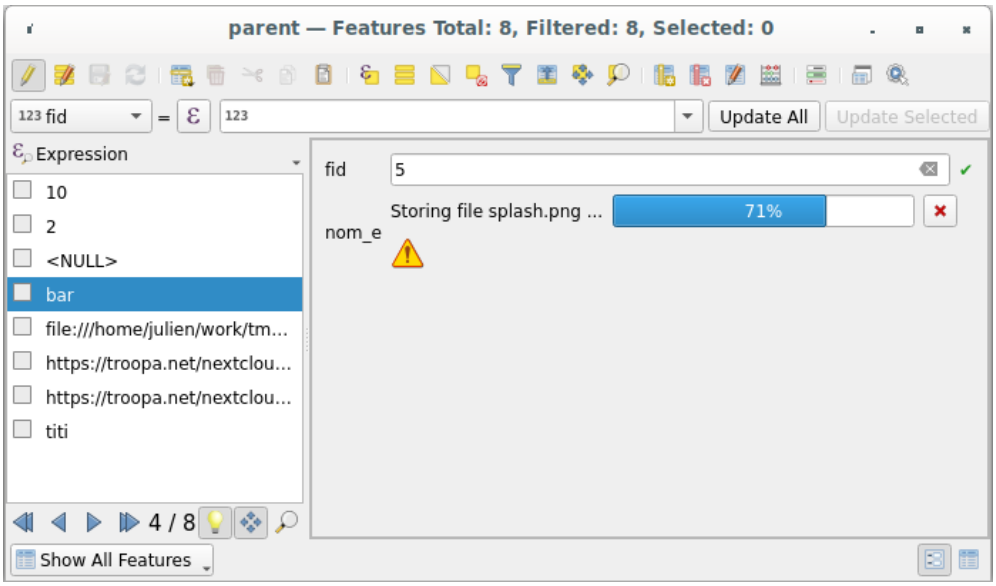



그림 12.119: Storing a file to a WebDAV external storage

참고

User can also achieve the same result if he drags and drops a file on the whole attachment widget.

Use the Cancel button to abort the storing process. It's possible to configure a viewer using the *Integrated document viewer* so the resource will be automatically fetched from the external storage system and displayed directly below the URL. The above icon indicates that the resource cannot be fetched from the external storage system. In that case, more details might appear in the 로그 메시지 패널.

CHAPTER 13

# 래스터 데이터 작업

## 13.1 래스터 속성 대화창

래스터 데이터는 픽셀 (또는 셀) 로 이루어져 있으며, 픽셀 하나가 값 하나를 가지고 있습니다. 래스터 데이터는 다음을 포함하는 다양한 유형의 데이터를 저장하는 데 쓰입니다:

- 위성 영상, 수치 (digital) 항공 사진, 스캔한 지도 같은 이미지 (imagery)
- 수치 표고 모델 (DEM; Digital Elevation Model), 수치 지형 모델 (DTM; Digital Terrain Model) 같은 표고 데이터
- 토지 피복 (land cover), 토양 유형, 강우량 및 다른 많은 정보 같은 기타 데이터 유형들

Raster data can be stored in several supported formats, including GeoTIFF, ERDAS Imagine, ArcInfo ASCII GRID, PostgreSQL Raster and others. See more at 데이터 열기.

래스터 레이어의 속성을 살펴보고 설정하려면, 맵 범례에 있는 레이어명을 더블클릭하거나 또는 레이어명을 오른쪽 클릭한 다음 컨텍스트 메뉴에서 *Properties* 를 선택하십시오. *Raster Layer Properties* 대화창이 열릴 것입니다.

이 대화창에는 다음과 같은 탭들이 있습니다:

| | | |
|---|---|---|
| 정보 | 소스 | *Symbology* [1] |
| *Transparency* [1] | *Labels* [1] | *Histogram* [1] |
| 렌더링 | 시계열 | 피라미드 |
| 표고 | 메타데이터 | *Legend* |
| 표시 | *Attribute Tables* | *QGIS* 서버 |
| *External plugins* [2] tabs | | |

[1] Also available in the *Layer styling panel*

[2] *External plugins* you install can optionally add tabs to this dialog. Those are not presented in this document. Refer to their documentation.

> **팁**
>
> **실시간 렌더링 업데이트**
>
> 레이어 스타일 작업 패널 은 레이어 속성 대화창의 공통 기능 가운데 일부를 제공하며, 레이어 스타일을 빠르게 환경 설정할 수 있고 맵 캔버스 상에서 사용자의 변경 사항을 살펴볼 수 있는 훌륭한 모달리스 위젯입니다.

> **참고**
>
> 원본 프로젝트 파일에서 삽입 레이어 (외부 프로젝트로부터 레이어를 내포시키기 참조) 의 속성들 (심볼, 라벨, 액션, 기본값, 양식 등등) 을 읽어오기 때문에, 이 습성을 방해할 수도 있는 변경 사항이 적용되는 일을 피하기 위해 삽입 레이어에 대해 레이어 속성 대화창을 사용할 수 없게 돼 있습니다.

### 13.1.1 정보 속성

*Information* 탭은 읽기 전용으로, 현재 레이어에 대한 요약 정보 및 메타데이터를 한 눈에 볼 수 있는 흥미로운 장소입니다. 다음과 같은 정보를 제공합니다:

- 프로젝트 이름, 소스 경로, 보조 파일 목록, 최신 저장 시간 및 용량, 사용하는 제공자 등의 일반 정보
- custom properties, used to store in the active project additional information about the layer. Default custom properties include *Identify/format*, which influences how the results from using the *Identify features* tool over a raster layer are formatted. More properties can be created and managed using PyQGIS, specifically through the setCustomProperty() method.
- 레이어 제공자 기반 정보: 범위, 너비 및 높이, 데이터 유형, GDAL 드라이버, 밴드 통계
- 좌표계: 이름, 단위, 메소드, 정확도, 참조 (예: 정적인지 동적인지 여부)
- 레이어 속성에서 읽어오는 정보: 데이터 유형, 범위, 너비/높이, 압축, 픽셀 크기, 밴드 통계, 열 및 행의 개수, 래스터의 NODATA 값, …
- 채워진 메타데이터 로부터 선택: 접근, 범위, 링크, 연락처, 이력, …

### 13.1.2 소스 속성

*Source* 탭은 선택한 래스터에 대한 다음과 같은 기본 정보를 표시합니다:

- *Layer name*: *Layers Panel* 에 표시될 레이어명
- *Coordinate Reference System*: 레이어의좌표계 를 표시합니다. 드롭다운 목록에서 최근 사용한 좌표계를 선택하거나 Select CRS 버튼을 (좌표계 선택기 참조) 클릭해서 레이어 좌표계를 변경할 수 있습니다. 레이어에 적용된 좌표계가 틀렸거나 또는 적용된 좌표계가 없는 경우에만 변경하십시오. 사용자 데이터를 다른 좌표계로 재투영하고 싶은 경우, 공간 처리 재투영 알고리즘을 이용하거나새 데이터셋으로 레이어를 저장 하는 편이 좋습니다.
- Depending on the data provider, a *Layer source* group indicates the path to the source of the dataset and allows for replacing the loaded layer:
  - When the layer is stored as file on disk, edit the path shown in the text box or press … Browse to select another file on the disk

- When the layer is provided by an ArcGIS MapServer service, it is possible to modify its *authentication settings*, while keeping unchanged the details for connecting to the service.
- For an XYZ remote layer, it is possible to modify at the layer level any of its *connection details* (URL, authentication settings, zoom levels, resolution, ⋯). Modifications are applied to the layer without altering the original connection settings.

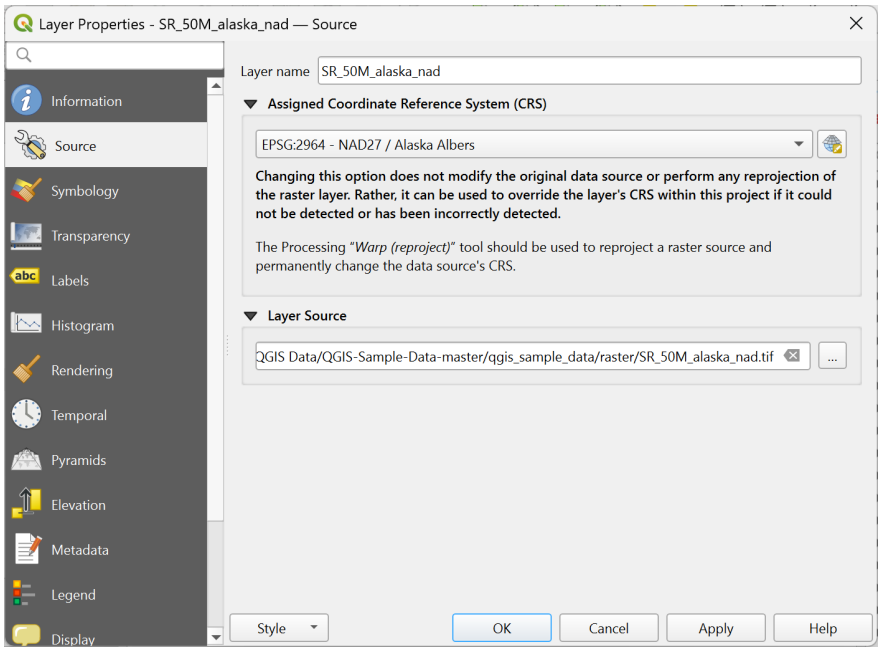



그림 13.1: 래스터 레이어 속성 — 소스 대화창

### 13.1.3 심볼 속성

래스터 레이어 심볼 탭은 세 부분으로 이루어져 있습니다:

- *Band rendering*: 사용할 렌더링 작업자 유형을 제어할 수 있습니다.
- *Layer rendering*: 렌더링된 데이터에 효과를 적용할 수 있습니다.
- *Resampling*: 맵의 렌더링을 최적화하기 위한 리샘플링 메소드를 선택할 수 있습니다.

#### 밴드 렌더링

QGIS 는 서로 다른 *Render types* 를 여럿 제공하고 있습니다. 데이터 유형 및 사용자가 강조하고자 하는 정보에 따라 렌더링 작업자를 선택합니다.

1. 다중 밴드 색상: 파일이 밴드 여러 개를 가지고 있는 경우 (예: 밴드 여러 개를 가진 위성 이미지)
2. 팔레트/유일값: 인덱스가 적용된 팔레트를 가지고 있는 단일 밴드 파일 (예: 디지털 지형도) 또는 래스터 레이어 렌더링을 위한 팔레트의 일반적인 활용을 위한 경우
3. 단일 밴드 회색조: 이미지 (의 밴드 하나) 를 회색조로 렌더링할 것입니다. QGIS 는 파일이 다중 밴드도 아니고 팔레트도 가지고 있지 않을 경우 (예: 음영기복도) 이 렌더링 작업자를 선택할 것입니다.
4. 단일 밴드 의사색상: 연속적인 팔레트 또는 색상표를 가지고 있는 파일의 경우 (예: 표고 지도) 이 렌더링 작업자를 사용할 수 있습니다.

5. *Single color* - the raster layer will be rendered with a single color.
6. 음영기복: 밴드로부터 음영기복을 생성합니다.
7. 등고선: 소스 래스터 밴드에 대해 등고선을 실시간으로 (on-the-fly) 생성합니다.

### 다중 밴드 색상

다중 밴드 색상 렌더링 작업자는 이미지에서 선택한 밴드 3 개를 렌더링합니다. 각 밴드는 색상 이미지의 빨간색 (R), 초록색 (G), 파란색 (B) 요소로 쓰입니다. QGIS 는 래스터의 각 밴드에서 *Min* 과 *Max* 값을 자동적으로 가져와 그에 따라 색상 작업을 조정합니다. 최소/최대 값 설정 부분에서 값의 범위를 제어할 수 있습니다.

값에 적용할 수 있는 *Contrast enhancement* 방법을'No enhancement', 'Stretch to MinMax', 'Stretch and clip to MinMax' 그리고'Clip to min max'가운데 선택할 수 있습니다.

> **참고**
>
> **대조 개선**
>
> GRASS 래스터를 추가하면, QGIS 전체 수준 옵션에서 어떤 값으로 설정돼 있건 상관없이, 자동적으로 *Contrast enhancement* 옵션이 항상 *stretch to min max* 로 설정될 것입니다.

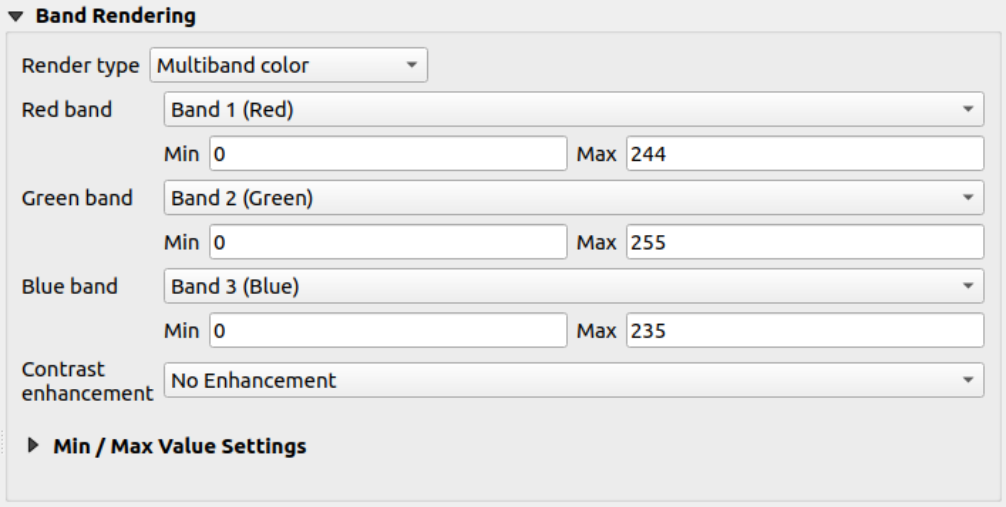



그림 13.2: 래스터 심볼 — 다중 밴드 색상 렌더링

> **팁**
>
> **다중 밴드 래스터의 단일 밴드 살펴보기**
>
> 다중 밴드 이미지의 단일 밴드를 (예를 들어 빨간색 밴드를) 살펴보고 싶다면, 초록색과 파란색 밴드를 *Not Set* 으로 설정해야 한다고 생각할 수도 있습니다. 그러나 그건 정확한 방법이 아닙니다. 빨간색 밴드만 표시하려면, 이미지 유형을단일 밴드 회색조 로 설정한 다음 *Gray band* 를 빨간색으로 선택하십시오.

### 팔레트/유일값

각 픽셀값에 특정 생상을 할당한 색상표를 가지고 있는 단일 밴드 파일을 위한 표준 렌더링 옵션입니다. 이런 경우, 팔레트를 자동적으로 렌더링합니다.

래스터의 각 유일값에 색상을 할당해서 모든 유형의 래스터 밴드에 사용할 수 있습니다.

어떤 색상을 변경하고 싶은 경우, 해당 색상을 더블클릭하기만 하면 *Select color* 대화창이 열립니다.

색상에 라벨도 적용할 수 있습니다. 이 라벨은 래스터 레이어의 범례에 표시됩니다.

색상표에서 선택한 행들을 오른쪽 클릭하면 다음 옵션을 가진 컨텍스트 메뉴가 나타납니다:

- *Change Color*…: 선택 집합의 색상을 변경합니다.
- *Change Opacity*…선택 집합의 투명도를 변경합니다.
- *Change Label*…: 선택 집합의 라벨을 변경합니다.

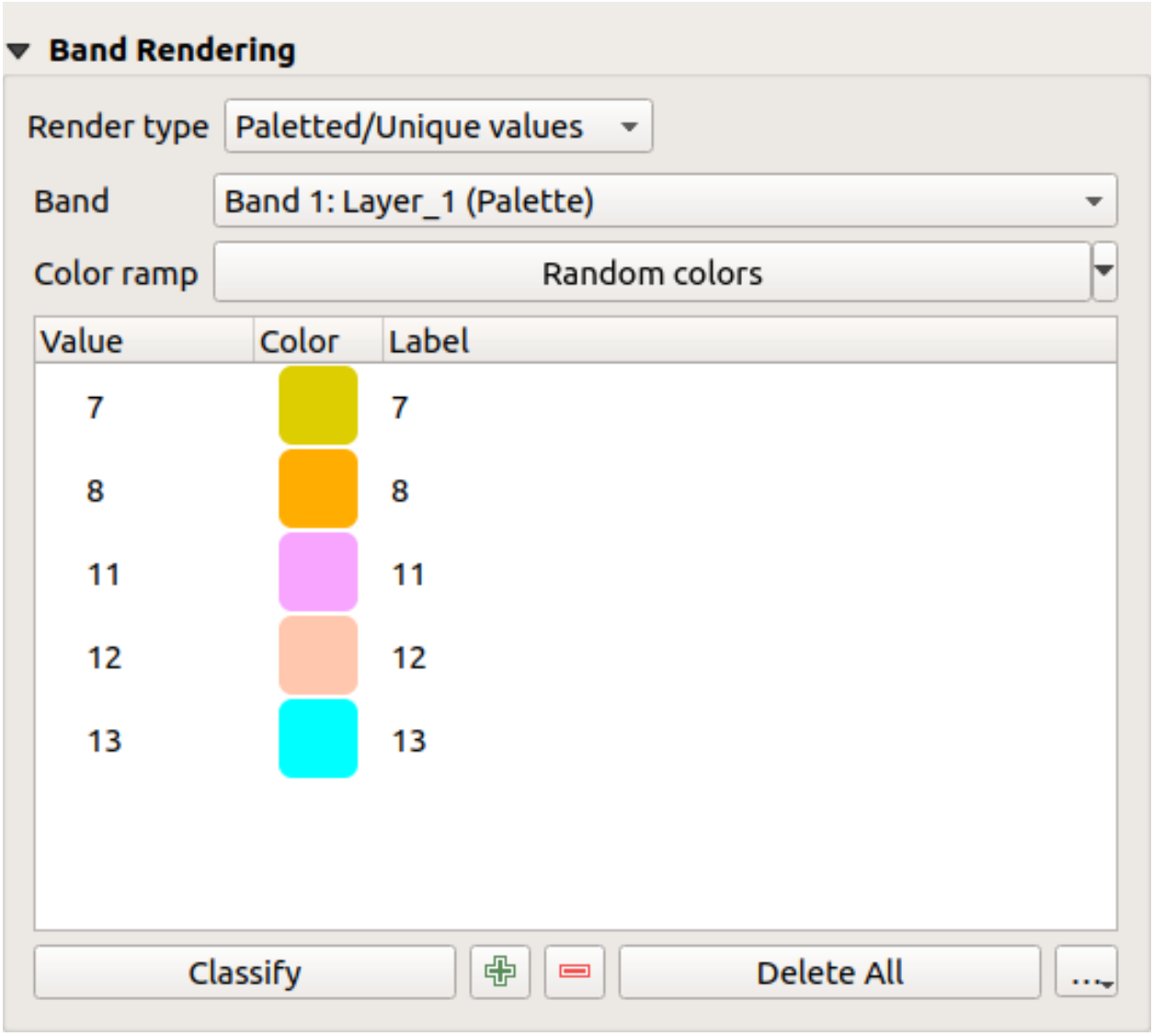



그림 13.3: 래스터 심볼 — 팔레트/유일값 렌더링

색상표 우하단에 있는 ⋯ (Advanced options) 버튼을 클릭하면 열리는 드롭다운 메뉴에서 색상표 불러오기 (*Load Color Map from File*⋯), 색상표 내보내기 (*Export Color Map to File*⋯), 그리고 범주 불러오기 (*Load Classes from Layer*) 기능을 사용할 수 있습니다.

### 단일 밴드 회색조

이 렌더링 작업자는 *Color gradient* 를 통해 오직 하나의 밴드를'Black to white'또는'White to black'그레이디언트로 이용해서 레이어를 렌더링할 수 있게 해줍니다. 최소/최대 값 설정 에서 색상으로 변환할 값의 범위를 (*Min* 및 *Max*) 변경할 수 있습니다.

값에 적용할 수 있는 *Contrast enhancement* 방법을'No enhancement', 'Stretch to MinMax', 'Stretch and clip to MinMax' 그리고'Clip to min max'가운데 선택할 수 있습니다.

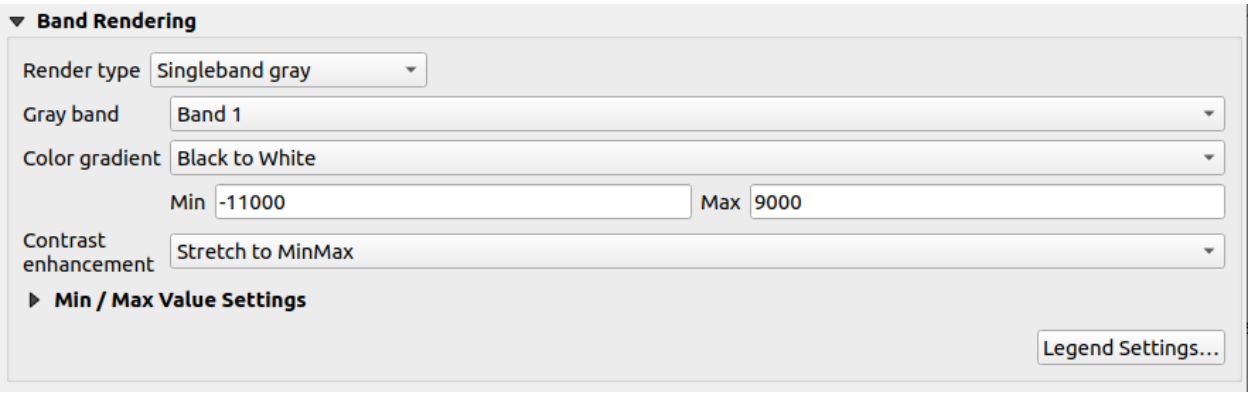



그림 13.4: 래스터 심볼 — 단일 밴드 회색조 렌더링

선택한 색상 그레이디언트와 (*Layers* 패널 및 조판범례 항목 에 있는) 레이어 범례를 기반으로 색상을 할당한 픽셀들을 연속 색상표를 이용해서 표시합니다. 이 설정을 미세 조정하고 싶다면 *Legend settings*⋯버튼을 누르십시오. 더 자세한 내용은래스터 범례 사용자 지정하기 를 참조하세요.

### 단일 밴드 의사색상

연속 팔레트를 포함하고 있는 단일 밴드 파일을 위한 렌더링 옵션입니다. 다중 밴드 래스터의 밴드 하나에 대한 색상표를 생성할 수도 있습니다.

레이어의 *Band* 와값의 범위 를 이용하면, 범주 내에 있는 픽셀에 표현 색상을 보간하고 할당할 수 있습니다. 자세한 내용은색상표 셰이더 범주화 를 참조하세요.

선택한 색상표와 (*Layers* 패널 및 조판범례 항목 에 있는) 레이어 범례를 기반으로 색상을 할당한 픽셀들을 연속 색상표를 이용해서 표시합니다. 이 설정을 미세 조정하고 싶거나, 또는 그 대신 구분된 범주들 (및 색상들) 을 가진 범례를 사용하고 싶다면 *Legend settings*⋯버튼을 누르십시오. 더 자세한 내용은래스터 범례 사용자 지정하기 를 참조하세요.

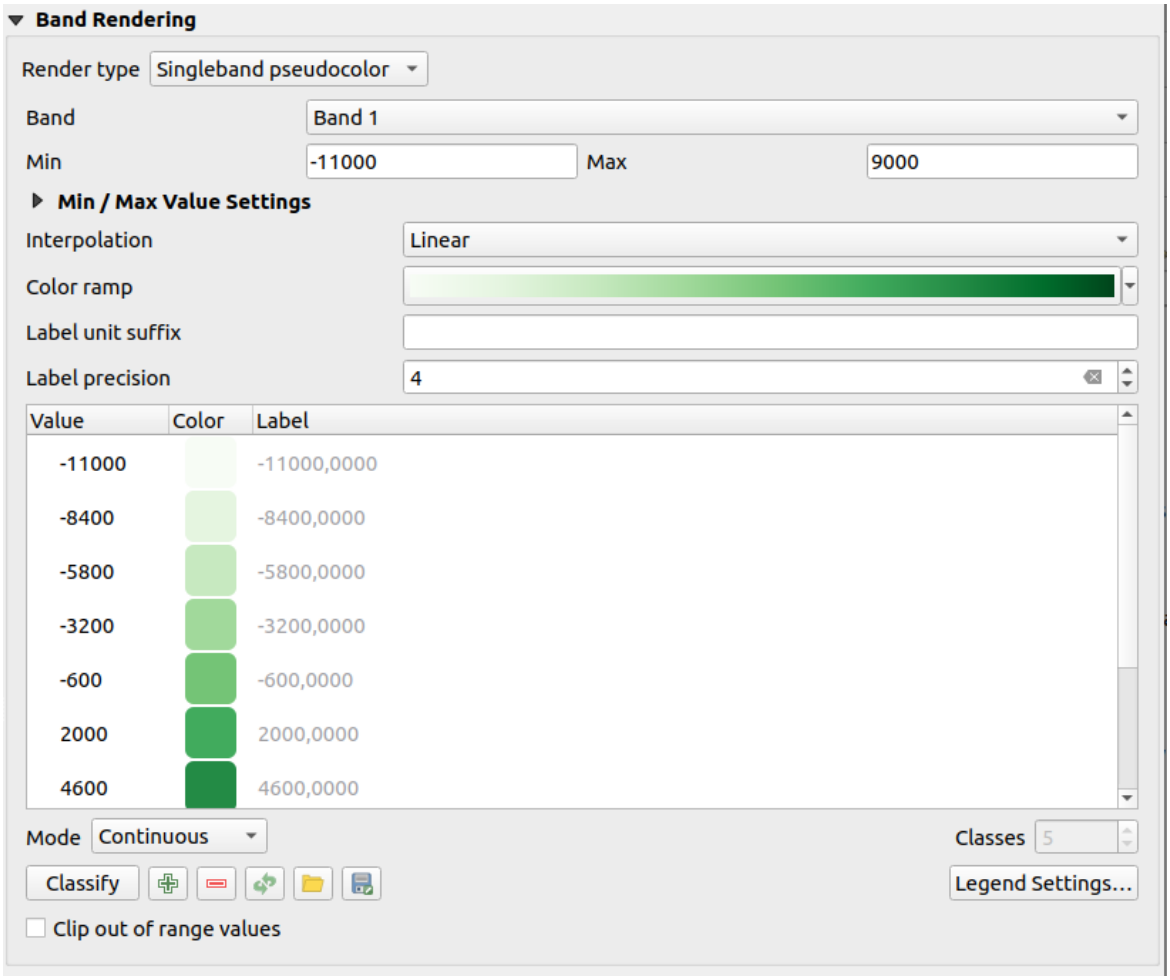



그림 13.5: 래스터 심볼 — 단일 밴드 의사색상 렌더링

### Single color

This renderer allows you to render a raster layer using *Single color*. This type of renderer is useful when you want to display a raster layer uniformly, without any variation in color based on pixel values.

The single color renderer can be used with both single-band and multiband raster layers. When used with multiband rasters, you can select which band to apply the single color to, effectively displaying that specific band uniformly across the entire layer.

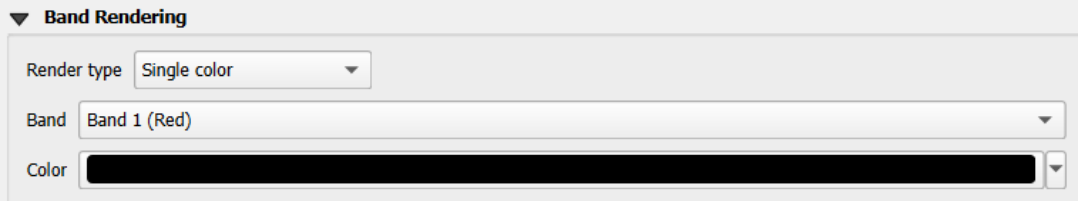



그림 13.6: Raster Symbology - Single color rendering

### 음영기복

음영기복을 사용해서 래스터 레이어의 밴드를 렌더링합니다.

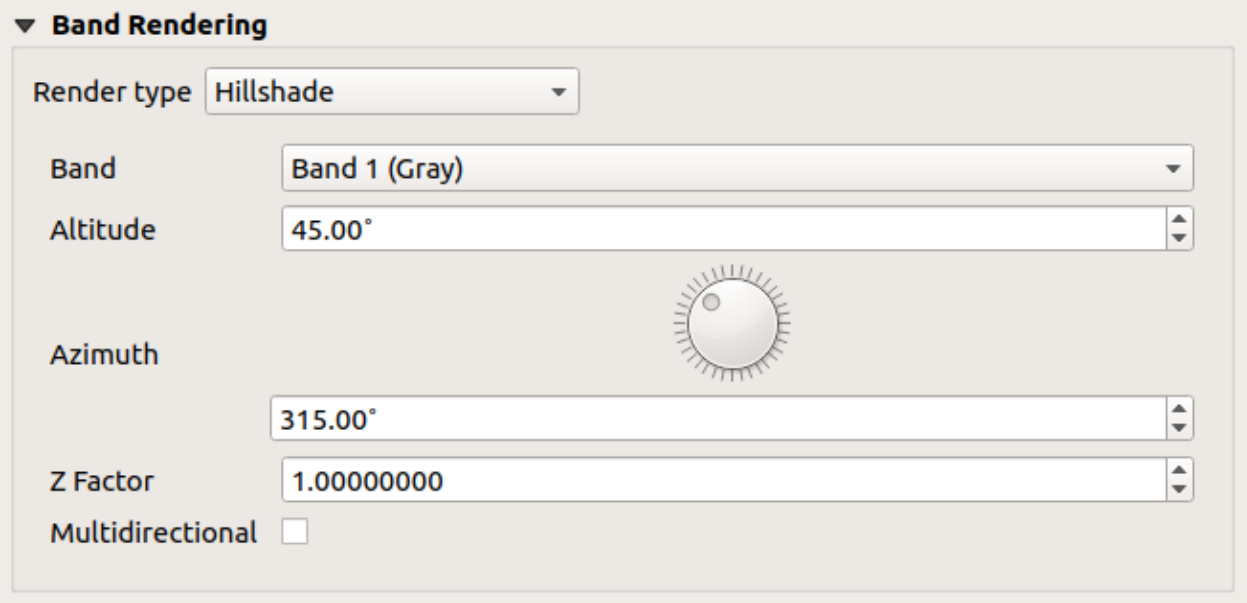



그림 13.7: 래스터 심볼 — 음영기복 렌더링

옵션:

- *Band*: 사용할 래스터 밴드
- *Altitude*: 조명 소스의 앙각 (elevation angle) (기본값: `45°`)
- *Azimuth*: 조명 소스의 방위각 (기본값: `315°`)
- *Z Factor*: 래스터 밴드의 값을 위한 척도 인자 (scaling factor) (기본값: `1`)
- *Multidirectional*: 다방향 음영기복을 사용할지 여부를 지정 (기본값: `off`)

### 등고선

이 렌더링 작업자는 소스 래스터 밴드로부터 실시간으로 (on-the-fly) 계산한 등고선을 그립니다.

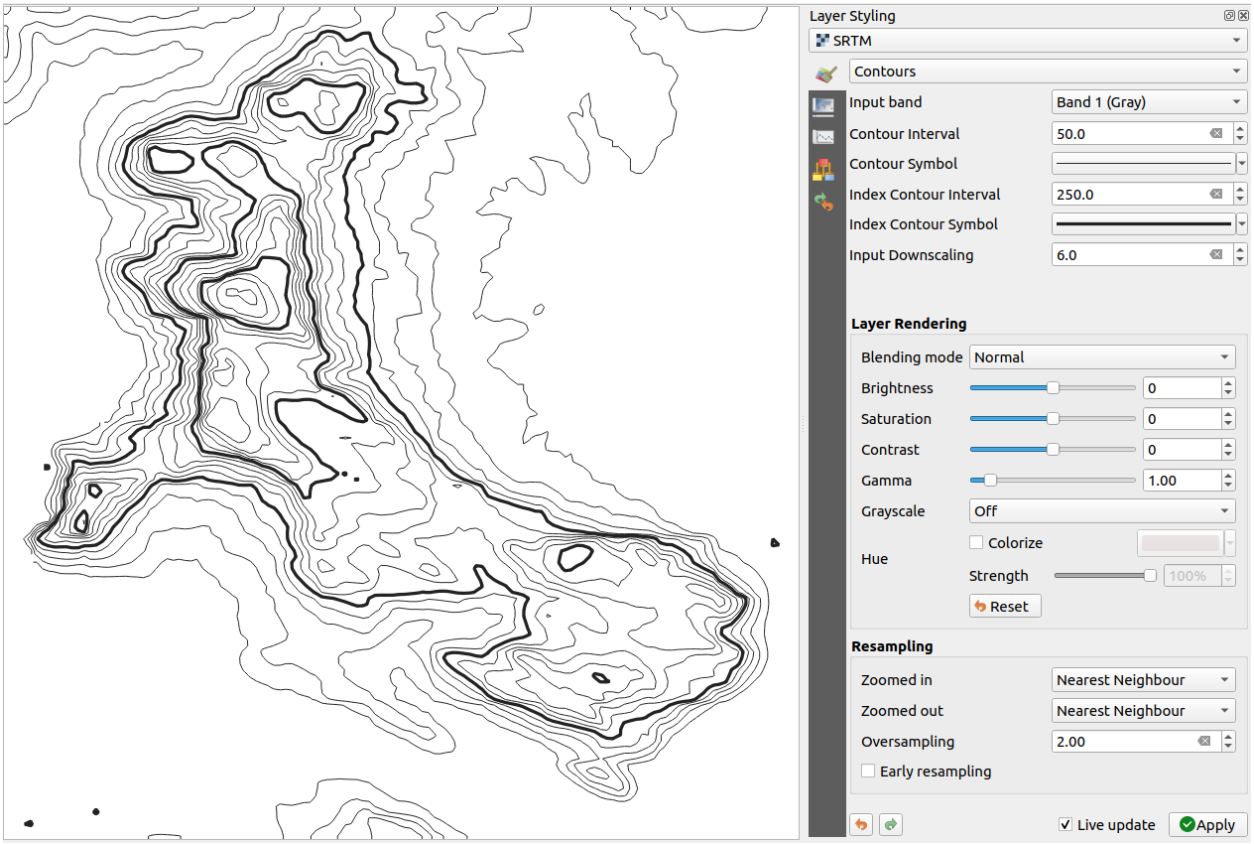



그림 13.8: 래스터 심볼 - 등고선 렌더링

옵션:

- *Input band*: 사용할 래스터 밴드
- *Contour interval*: 연속된 두 등고선 라인 사이의 거리
- *Contour symbol*: 등고선 라인에 공통으로 적용할심볼
- *Index contour interval*: 연속된 두 **색인 등고선 (index contour)** 사이의 거리로, 색인 등고선이란 식별하기 쉽도록 눈에 띄는 방식으로 표시된 라인을 말합니다. 일반적으로 다른 등고선 라인보다 더 진하게 인쇄되고, 라인을 따라 값 라벨이 붙습니다.
- *Index contour symbol*: 색인 등고선 라인에 적용할 심볼
- *Input downscaling*: 렌더링 작업자가 데이터 제공자에 전송할 요청을 얼마나 축소 (downscale) 할 것인지를 나타냅니다. (기본값은 `4.0` 입니다.)

  예를 들어, 입력 래스터 블록의 등고선을 산출 래스터 블록과 동일한 크기로 생성하는 경우, 생성된 라인이 너무 많은 세부 정보를 담고 있을 수 있습니다. 소스 래스터를 더 낮은 해상도로 요청하는"축소"인자로 이 세부 정보를 줄일 수 있습니다. 축소 인자를 10 으로 설정하면, 렌더링 작업자가 제공자에 1000x500 크기의 래스터 블록을 100x50 크기로 요청할 것입니다. 축소 인자가 클수록 등고선이 더 단순화됩니다. (일부 세부 정보는 누락되겠지만 말이죠.)

### 최소/최대 값 설정하기

QGIS 는 래스터의 밴드 (들) 의 *Min* 및 *Max* 값을 기본적으로 보고합니다. 너무 낮은 그리고/또는 너무 높은 몇몇 값들은 래스터를 렌더링하는 데 부정적인 영향을 줄 수 있습니다. *Min/Max Value Settings* 프레임에서 렌더링을 제어할 수 있습니다.

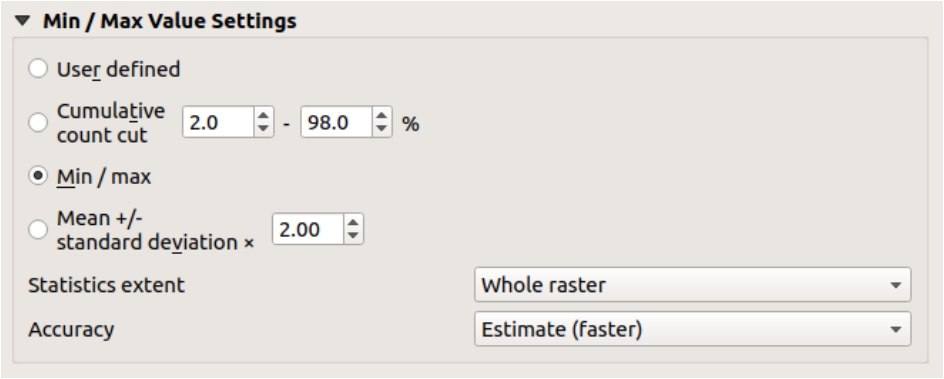



그림 13.9: 래스터 심볼 — 최소/최대 값 설정하기

다음과 같은 옵션을 사용할 수 있습니다:

- *User defined*: 밴드의 기본 *Min* 및 *Max* 값을 무시할 수 있습니다.
- *Cumulative count cut*: 특이값 (outlier) 을 제거합니다. 값의 표준 범위는 `2%` 부터 `98%` 까지지만, 사용자가 직접 조정할 수 있습니다.
- *Min / max*: 이미지 밴드에 있는 값의 전체 범위를 사용합니다.
- *Mean +/- standard deviation x*: 표준 편차 또는 다중 표준 편차 안에 떨어지는 값만을 대상으로 색상표를 생성합니다. 래스터 레이어에 래스터 렌더링에 부정적인 영향을 주는 비정상적으로 높은 값을 가진 셀이 한두 개 있을 때 유용합니다.

밴드의 최소/최대 값은 다음을 기반으로 계산됩니다:

- *Statistics extent*: *Whole raster*, *Current canvas* 또는 *Updated canvas* 가운데 선택할 수 있습니다. *Updated canvas* 는 렌더링에 사용되는 최소/최대 값이 캔버스 범위와 함께 변경될 것이라는 (동적으로 줄거나 늘어날 것이라는) 의미입니다.
- *Accuracy*: *Estimate (faster)* 또는 *Actual (slower)* 가운데 선택할 수 있습니다.

> **참고**
>
> 일부 설정의 경우, 위젯에 실제 최소/최대 값을 표시하기 위해 레이어 속성 대화창의 *Apply* 버튼을 눌러야 할 수도 있습니다.

### 색상표 셰이더 범주화

이 메소드를 이용하면 스칼라 (scalar) 데이터셋 (래스터 또는 메시 등고선) 을 그 값을 기반으로 범주화하고 표현할 수 있습니다. 색상표 와 범주의 개수를 지정하면, 범주 경계 값에 대한 중간 색상표 항목들을 생성합니다. 각 색상은 범주화 모드에 따라 값의 범위로부터 보간된 값으로 매핑됩니다. 그 다음 그 범주를 기반으로 스칼라 데이터셋 요소들에 색상을 할당합니다.

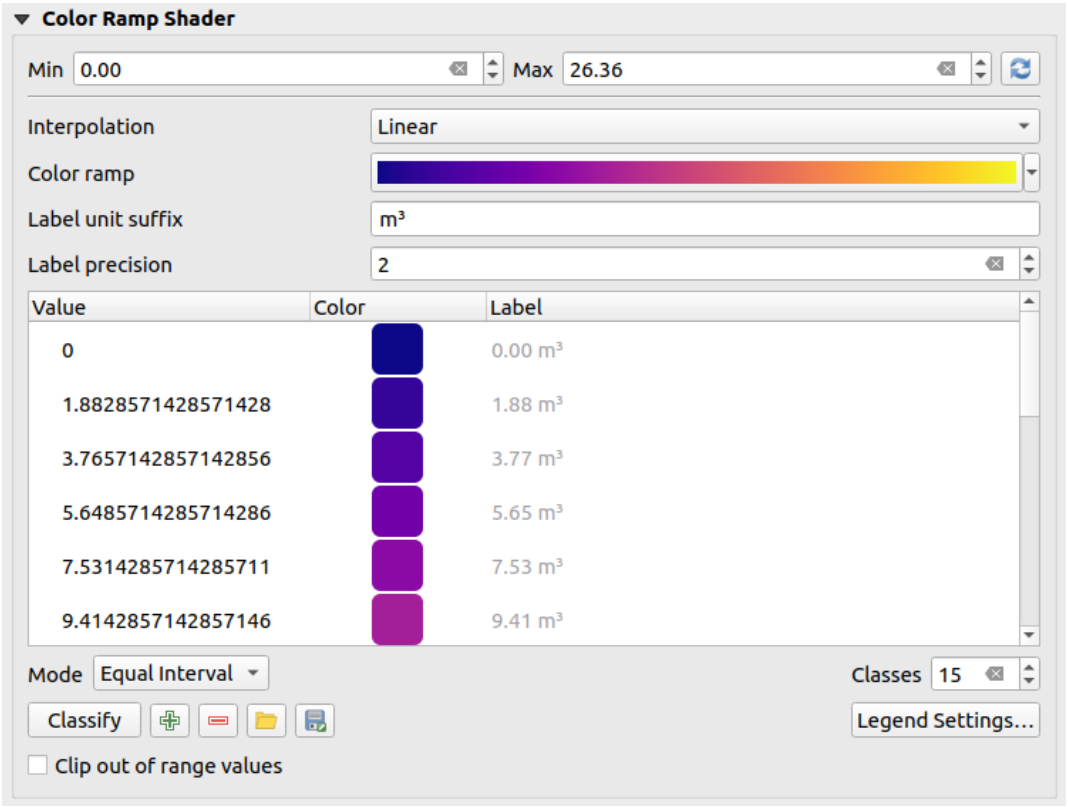



그림 13.10: 색상표 셰이더 (shader) 로 데이터셋을 범주화하기

1. *Min* 및 *Max* 값을 이용해서 범주 경계를 보간하기 때문에 이 값들을 반드시 정의해야만 합니다. 기본적으로 QGIS 가 데이터셋에서 최소/최대 값을 탐지하지만, 이 값들을 수정할 수 있습니다.
2. *Interpolation* 항목이 스칼라 요소에 색상을 어떻게 할당할지를 정의합니다:
   - *Discrete* (*Value* 열 헤더에 <= 심볼이 나타납니다): 동등하거나 더 높은 값을 가진, 가장 가까운 색상표 항목에서 색상을 선택합니다.
   - *Linear*: 픽셀 값보다 높거나 낮은 색상표 항목에서 색상을 선형적으로 보간합니다. 다시 말해 데이터셋의 각 값에 유일 색상을 대응시킨다는 뜻입니다.
   - *Exact* (*Value* 열의 헤더에 = 심볼이 나타납니다): 색상표 항목과 동등한 값을 가진 픽셀에만 색상을 적용합니다. 그렇지 않은 픽셀은 렌더링하지 않습니다.
3. *Color ramp* 위젯에서 데이터셋에 할당할 색상표를 선택할 수 있습니다. 이 위젯 에서와 같이, 새 색상표를 생성하거나 현재 선택한 색상표를 편집하고 저장할 수 있습니다. 색상표의 이름은 환경 설정에 저장될 것입니다.
4. *Label unit suffix* 란에서 범례에 있는 값 뒤에 라벨을 추가하고, *Label precision* 란에서는 표시할 소수점 이하 자릿수를 제어합니다.
5. 범주화 *Mode* 메뉴를 통해 범주 전체에 걸쳐 값을 어떻게 분포시킬지 정의할 수 있습니다:

- *Equal interval*: *Number of classes* 를 지정하면, 범주들이 모두 동일한 크기를 가지도록 경계 값 (limits value) 들을 정의합니다.
- *Continuous*: 색상표의 색상 지점 (stop) 으로부터 범주 개수 및 색상을 가져옵니다. 색상표의 색상 지점 분포를 따라 경계 값을 설정합니다.
- *Quantile*: *Number of classes* 를 지정하면, 각 범주가 동일한 개수의 요소를 가지도록 경계 값을 정의합니다. 메시 레이어 에는 사용할 수 없습니다.

6. 범주화 모드를 설정하고 나면 범주들을 범주화 (*Classify*) 하거나 미세 조정할 수 있습니다:

   -  Add values manually: 이 버튼은 테이블에 값을 추가합니다.
   -  Remove selected row: 이 버튼은 테이블에서 선택한 값을 삭제합니다.
   - *Value*: 이 열을 더블클릭하면 범주 값을 수정할 수 있습니다.
   - *Color*: 이 열을 더블클릭하면 *Change color* 대화창이 열리고, 해당 값에 적용할 색상을 선택할 수 있습니다.
   - *Label*: 이 열을 더블클릭하면 범주 라벨을 수정할 수 있습니다. 그러나 사용자가 피처 식별 도구를 사용할 때 이 값은 표시되지 않을 것입니다.
   - 색상표에서 선택한 행들을 오른쪽 클릭하면 해당 선택 집합에 대해 색상 (*Change Color…*) 및 불투명도 (*Change Opacity…*) 를 변경할 수 있습니다.

    Load color map from file 또는  Export color map to file 버튼을 이용해서 기존 색상표를 불러오거나 또는 다른 세션을 위해 색상표를 저장할 수도 있습니다.

7. *Interpolation* 드롭다운 메뉴에서 선형 보간을 선택한 경우, 다음도 환경 설정할 수 있습니다:

   -  *Clip out of range values*: 선형 메소드는 기본적으로 설정된 *Min* 값 미만의 (또는 설정된 *Max* 값을 초과하는) 데이터셋 값에 첫 번째 범주 (또는 마지막 범주) 색상을 할당합니다. 이 범위 밖의 값들을 렌더링하지 않고자 하는 경우 이 옵션을 체크하십시오.
   - *Legend settings*: *Layers* 패널의 범례 및조판 범례 항목 을 어떻게 표시할지 설정합니다. 더 자세한 내용은래스터 범례 사용자 지정하기 를 참조하세요.

### 래스터 범례 사용자 지정하기

래스터 또는 메시 레이어에 색상표를 적용시킬 때, 범주를 보여주는 범례를 표시하고자 할 수도 있습니다. QGIS 는 기본적으로 *Layers* 패널 및 조판의범례 항목 에 최소/최대 값을 가진 연속 색상표를 표시합니다. 범주화 위젯에 있는 *Legend settings* 버튼을 통해 이 색상표를 사용자 지정할 수 있습니다.

 *Use continuous legend*: 이 대화창에서 연속 색상 범례를 사용할지 여부를 설정할 수 있습니다. 이 옵션을 체크해제하면 서로 다른 범주들에 각각 대응하는 개별 색상들을 적용한 범례를 표시합니다. 단일 밴드 회색조 심볼을 사용하는 래스터에는 이 옵션을 활성화시킬 수 없습니다.

*Use continuous legend* 옵션을 체크하면 범례의 라벨 및 조판 속성 둘 다 환경 설정할 수 있습니다.

**라벨**

- *Prefix* 및 *Suffix*: 라벨에 접두어 및 접미어를 추가할 수 있습니다.
- *Minimum* 및 *Maximum*: 범례에 표시될 최소 및 최대 값을 수정할 수 있습니다.
- *Number format*: 숫자 서식을사용자 지정 할 수 있습니다.
- *Text format* : 인쇄 조판 범례에 쓰일 텍스트 서식을사용자 지정 할 수 있습니다.

**조판**

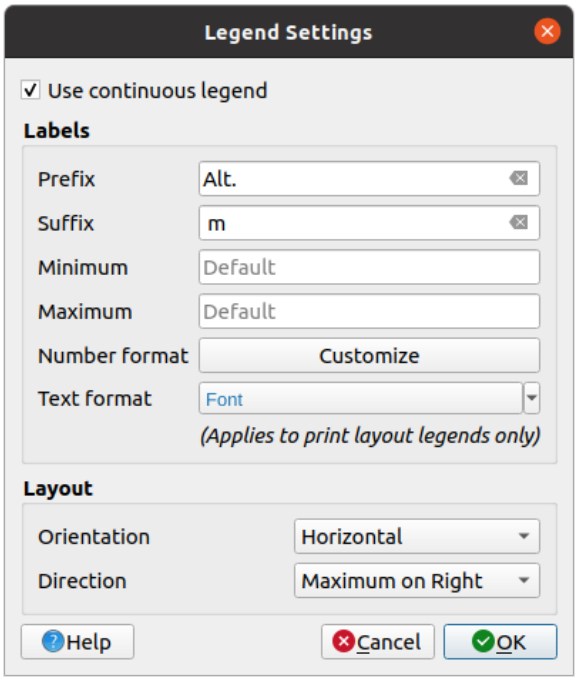



그림 13.11: 래스터 범례 수정하기

- *Orientation*: 범례 색상표의 방향을 제어합니다. **수직** 또는 **수평** 가운데 하나로 선택할 수 있습니다.
- *Direction*: 색상표의 방향에 대한 값의 방향을 제어할 수 있습니다.
  - 수직을 선택한 경우, **상단이 최대값 (Maximum on top)** 또는 **상단이 최소값 (Minimum on top)** 방향으로 표시할 수 있습니다.
  - 수평을 선택한 경우, **우측이 최대값 (Maximum on right)** 또는 **우측이 최소값 (Minimum on right)** 방향으로 표시할 수 있습니다.

### 레이어 렌더링

레이어 밴드 (들) 에 적용된 심볼 유형을 통해 전체 래스터 파일 (들) 에 대한 특수 렌더링 효과를 얻을 수 있습니다:

- 혼합 모드 가운데 하나를 사용할 수 있습니다.
- 색상에 사용자 지정 명도 (*Brightness*), 채도 (*Saturation*), 감마 (*Gamma*) 및 대조 (*Contrast*) 를 설정할 수 있습니다.
- *Invert colors* 옵션을 사용하면 레이어를 보색 (opposite color) 으로 렌더링할 수 있습니다. 예를 들어 원본 오픈스트리트맵 타일을 다크 모드로 전환하는 데 편리한 옵션입니다.
- *Grayscale* 옵션을 이용하면'명도 (By lightness)', '광도 (By luminosity)'또는'평균 (By average)'메소드 가운데 하나를 이용해서 레이어를 회색조로 변환할 수 있습니다.
- 색상표에서 색조 (*Hue*) 의 강도 (*Strength*) 를 색상화 (*Colorize*) 하고 조정할 수 있습니다.

레이어 렌더링에 적용된 모든 사용자 지정 변경 사항을 제거하려면 *Reset* 버튼을 누르십시오.

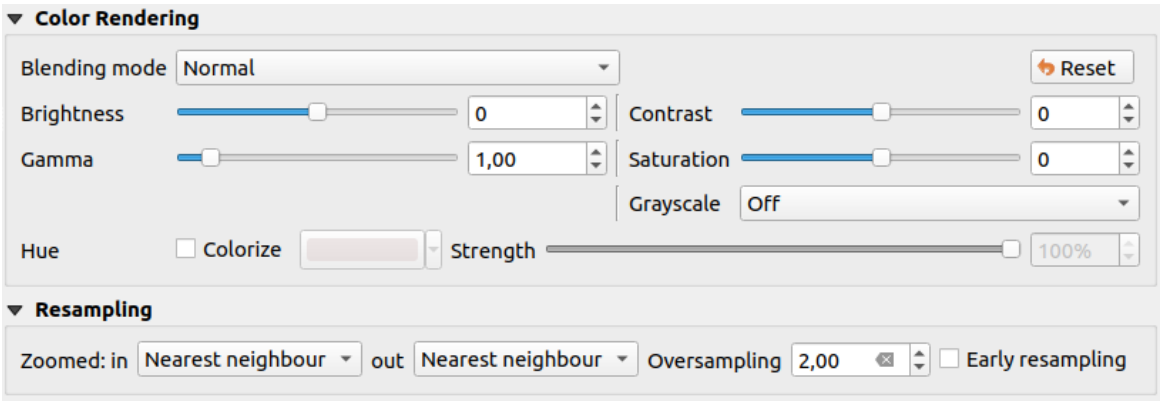



그림 13.12: 래스터 심볼 — 레이어 렌더링 및 리샘플링 설정하기

#### 리샘플링

*Resampling* 옵션의 효과는 사용자가 이미지를 확대/축소하는 경우 나타납니다. 리샘플링 모드로 맵의 모습을 최적화할 수 있습니다. 리샘플링 모드들은 기하학적 변형을 통해 새 회색조 값 매트릭스를 생성합니다.

'최근접 이웃 (Nearest neighbour)'방법을 적용하면, 확대 시 맵의 픽셀이 두드러져 보일 수 있습니다. '이중선형 (Bilinear) (2x2 커널)'또는'3 차 (Cubic) (4x4 커널)'방법을 사용하면 이 모습을 향상시킬 수 있는데, 이 방법들은 뚜렷한 경계를 흐릿하게 만듭니다. 이 흐리기 효과로 더 평탄화된 이미지를 생성할 수 있습니다. 예를 들자면 래스터 수치지형도에 이 방법을 적용할 수 있습니다.

*Early resampling*: 이 옵션을 체크하면 소스의 해상도가 밝혀져 있는 제공자 수준에서 래스터 렌더링을 계산하기 때문에, 확대 렌더링 시 QGIS 사용자 지정 스타일 작업을 더 잘 반영하게 됩니다. 타일 래스터를해석 *(interpretation)* 메소드 를 통해 불러온 경우 이 옵션이 매우 유용합니다.

### 13.1.4 투명도 속성

QGIS 는 래스터 레이어의 *Transparency* 투명도 수준을 설정할 수 있는 기능을 제공합니다.

현재 래스터 레이어를 통해 기저 레이어가 (있다면) 어느 정도로 가시화되어야 할지 설정하려면 *Global opacity* 슬라이드 바를 이용하십시오. 래스터 레이어들을 중첩시킨 경우 (예를 들어 음영기복도 위에 범주화 래스터 맵을 중첩시킨 경우) 매우 유용한 기능입니다. 이렇게 하면 맵을 좀 더 3 차원적으로 보이게 할 것입니다. 래스터의 불투명도는 데이터 정의로 설정할 수 있습니다. 예를 들어 또다른 레이어의 가시성에 따라, 또는 시계열 변수에 따라, 또는 지도책의 어느 페이지에 있느냐에 따라, 또는 다른 수많은 조건들에 따라 래스터의 불투명도는 바뀔 수 있습니다.

*No data value* 옵션을 체크하면, QGIS 가 렌더링 시 있는 그대로 연산에 넣을 수 있는 원본 소스의 NODATA 값을 (있다면) 보고합니다. 또한 사용자가 *Additional no data value* 란에 추가적인 NODATA 값으로 취급해야 할 래스터 값을 직접 입력할 수도 있습니다. *Display no data as* 색상 선택기를 통해 NODATA 픽셀에 기본 투명도 렌더링 대신 사용자 지정 색상을 적용할 수도 있습니다.

*Custom transparency options* 부분에서 투명도를 훨씬 융통성 있게 사용자 지정할 수 있습니다:

- *Transparency band* 옵션을 통해 전체 밴드에 적용될 투명도를 설정하십시오.
- 대응하는 투명도 수준으로 투명하게 할 픽셀 목록을 지정하십시오:
  1. *Add values manually* 버튼을 클릭합니다. 픽셀 목록에 새 행이 나타날 것입니다.
  2. For single-band based symbology (e.g. DEMs), enter the **From** and **To** values and adjust the **Percent Transparent** to apply.

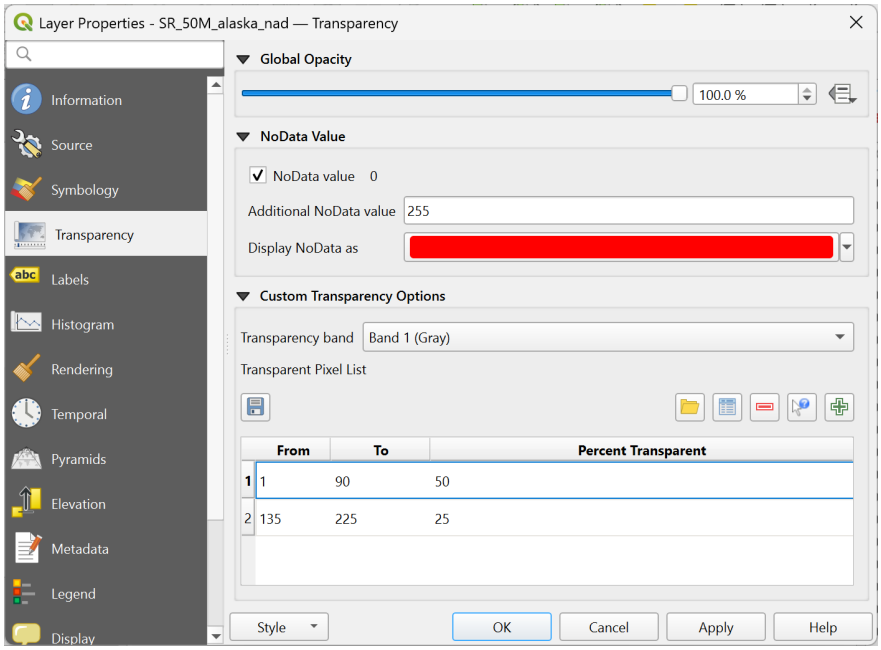



그림 13.13: 래스터 투명도

3. For multiband based symbology (e.g. RGB images) enter the **Red**, **Green** and **Blue** values of the pixel and adjust the **Percent Transparent** to apply. QGIS supports **Tolerance** for pixel values, when defining transparency. This means that pixels with color close to the specified RGB values can also be made transparent. Note that this feature applies only to multiband rasters.

4. 그 대신에, Add values from display 버튼으로 래스터에서 픽셀 값을 직접 가져올 수도 있습니다. 그 다음 투명도 값을 입력하십시오.

5. 더 많은 값을 사용자 지정 투명도로 조정하려면 앞 단계를 반복하십시오.

6. *Apply* 버튼을 클릭한 다음 맵을 살펴보십시오.

이처럼 사용자 지정 투명도를 설정하는 일은 꽤 쉬운 편이지만, 반복 작업을 오래 해야 할 수도 있습니다. 따라서, Export to file 아이콘을 이용해서 사용자의 투명도 목록을 파일로 저장할 수 있습니다. Import from file 아이콘은 사용자의 투명도 설정을 불러와 현재 래스터 레이어에 적용시킵니다.

### 13.1.5 Labels Properties

The *Labels* properties provides you with all the needed and appropriate capabilities to configure smart labeling on raster layers. This dialog can also be accessed from the *Layer Styling* panel.

At the top of the dialog, you have:

- a combobox for selecting the appropriate labeling method for the active layer
- the Configure project labeling rules button: helps you control interactions between labels and features across the layers in the project. More details at *Configuring project labeling rules*.
- the Automated placement settings (applies to all layers) button: configure general properties on label placement and conflicts resolution. More details at 자동화된 배치 엔진 설정하기.

The first step is to choose the labeling method from the drop-down list. Available methods are:

- *No labels*: the default value, showing no labels from the layer
- *Label with pixel values*: Show labels on the map using a band.

Using the *Label with pixel values* option, the following dialog opens.

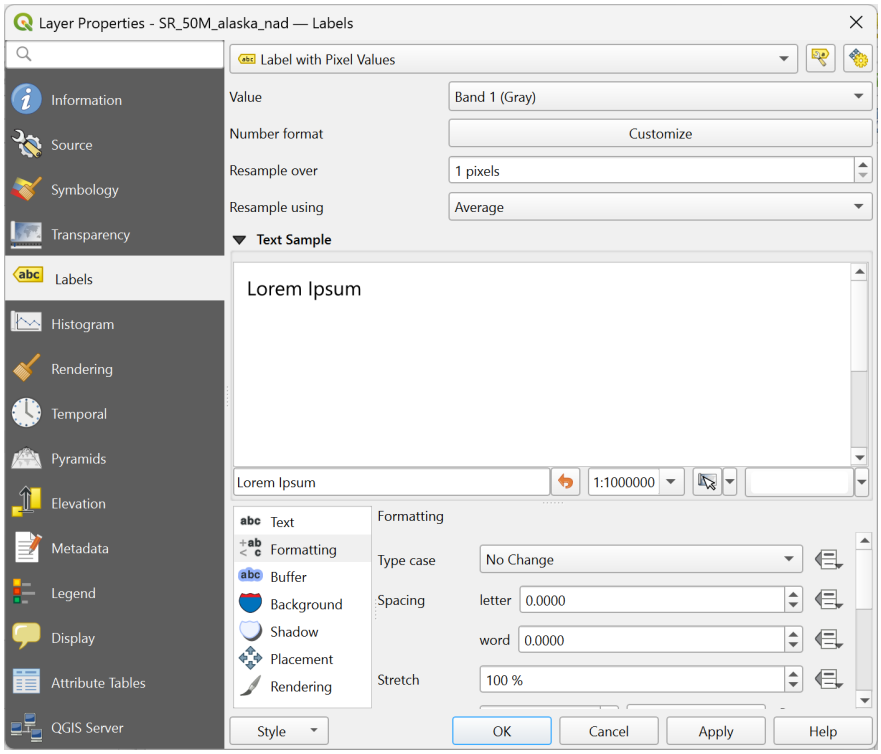



그림 13.14: Raster layer labeling settings

At the top of the dialog:

- A *Value* drop-down list allows you to select the band to take the values from
- Press *Customize* to configure a *proper number formatting*
- By default, the displayed value represents individual pixel band value. With *Resample over*, you can compute the value from the neighbouring pixels (setting `2` means `2*2=4` pixels) using a statistical method set in the *Resample using* widget.

Below are displayed options to customize the labels, under various tabs:

- *Text*
- *Formatting*
- *Buffer*
- *Background*
- *Shadow*
- *Placement*
- *Rendering*

Description of how to set each property is exposed at 라벨 설정.

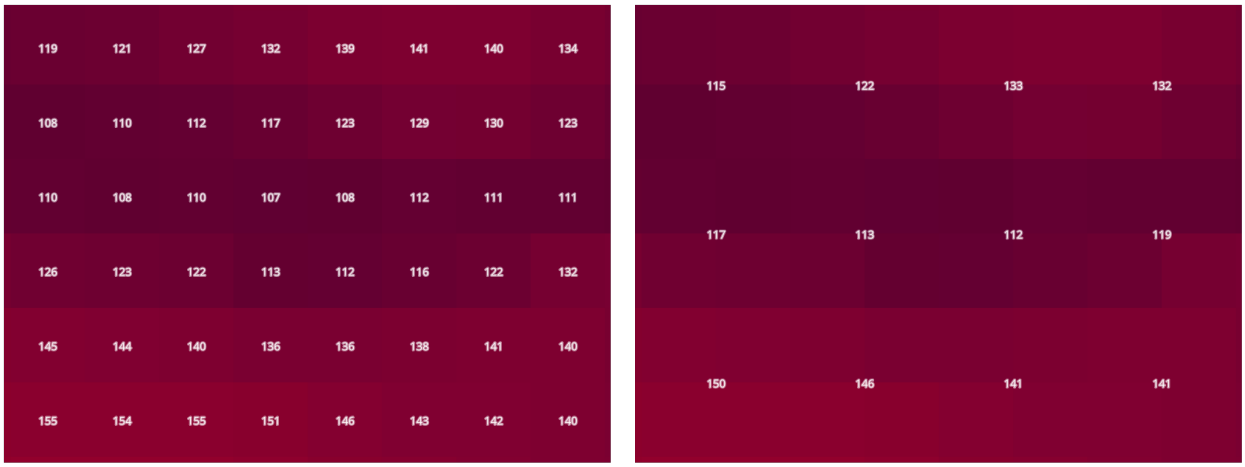



그림 13.15: Pixels labeled using various resampling options

### 13.1.6 히스토그램 속성

*Histogram* 탭에서 사용자 래스터에 있는 값의 분포를 살펴볼 수 있습니다. *Compute Histogram* 버튼을 클릭하면 히스토그램을 자동적으로 생성합니다. 모든 기존 밴드를 함께 표시할 것입니다. 아이콘을 클릭하면 히스토그램을 이미지로 저장할 수 있습니다.

히스토그램 하단의 드롭다운 메뉴에서 래스터 밴드를 선택하고, *Set min/max style for* 를 통해 해당 밴드의 최소/최대 스타일을 설정할 수 있습니다. *Prefs/Actions* 드롭다운 메뉴는 히스토그램을 사용자 지정할 수 있는 고급 옵션을 제공합니다:

- *Visibility* 옵션을 통해 개별 밴드의 히스토그램을 표시할 수 있는데, 먼저 *Show selected band* 옵션을 선택해야 합니다.
- *Min/max options* 에서'Always show min/max markers', 'Zoom to min/max'그리고'Update style to min/max' 가운데 하나를 선택할 수 있습니다.
- 밴드 (들) 의 최소 또는 최대값을 변경한 후, *Actions* 옵션에서'Reset'또는'Recompute histogram'가운데 하나를 선택해서 히스토그램을 다시 계산할 수 있습니다.

### 13.1.7 렌더링 속성

*Rendering* 탭에서 다음 설정을 할 수 있습니다:

- 레이어의 *Scale dependent visibility* 설정: *Maximum (inclusive)* 및 *Minimum (exclusive)* 축척을 설정해서 피처가 보이게 될 축척 범위를 정의할 수 있습니다. 이 범위를 벗어나면, 피처를 숨깁니다. Set to current canvas scale 버튼을 클릭하면 현재 맵 캔버스의 축척을 가시성 범위의 경계 값으로 설정할 수 있습니다. 자세한 내용은가시성 축척 선택기 을 참조하세요.

> **참고**
>
> *Layers* 패널에서 레이어에 대한 축척 의존 가시성을 활성화시킬 수도 있습니다. 레이어를 오른쪽 클릭한 다음 컨텍스트 메뉴에서 *Set Layer Scale Visibility* 를 선택하십시오.

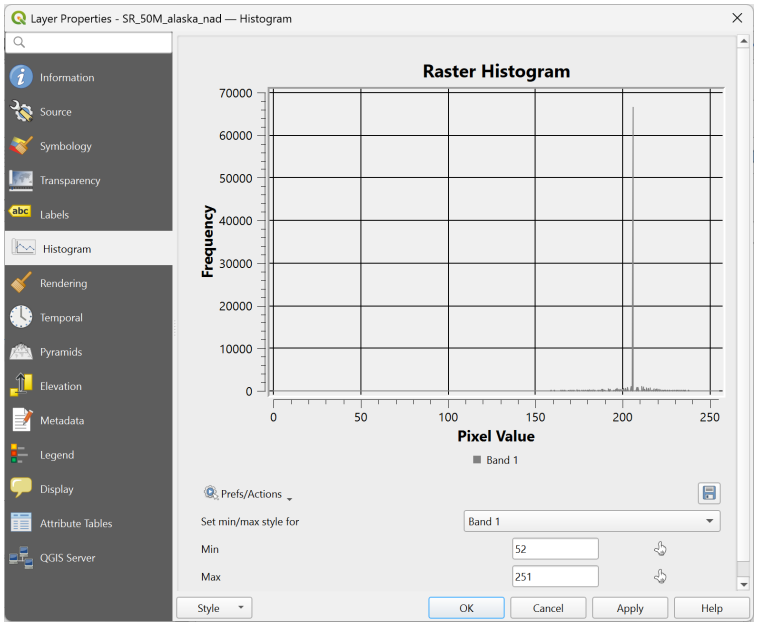



그림 13.16: 래스터 히스토그램

- *Refresh layer at interval*: 이 옵션은 레이어를 얼마나 정기적으로 새로고침할 수 있는지 그리고 새로고침할지 여부를 제어합니다. 사용할 수 있는 환경설정 *Configuration* 옵션들은 다음과 같습니다:
  - *Reload data*: 레이어를 완전히 새로고침할 것입니다. 캐시되어 있던 데이터를 모두 삭제하고 제공자로부터 다시 받아올 것입니다. 이 모드는 맵 새로고침 속도를 저하할 수도 있습니다.
  - *Redraw layer only*: 이 모드는 애니메이션에 대해 또는 레이어의 스타일이 정기적인 간격으로 업데이트될 경우 유용합니다. 하나 이상의 레이어에 자동 업데이트 간격을 설정한 경우 여러번 새로고침하지 않도록 캔버스 업데이트를 연기합니다.

  연속되는 새로고침 사이의 간격을 초 단위로 *Interval (seconds)* 설정할 수도 있습니다.

### 13.1.8 시계열 속성

*Temporal* 탭은 레이어의 렌더링을 시간에 따라 제어할 수 있는 옵션들을 제공합니다. 이런 동적 렌더링에는 맵 캔버스 전체에 적용되는시계열 탐색 을 활성화해야 할 필요가 있습니다.

*Dynamic Temporal Control* 옵션을 체크한 다음 레이어를 어떤 방식으로 다시 그릴지를 선택하십시오:

- *Automatic*: the rendering is controlled by the underlying data provider if it suppports temporal data handling. E.g. this can be used with WMS-T layers or PostgreSQL rasters.
- *Fixed Date/Time*: only show the raster layer at a single, specific date/time. This avoids having to enter the same value for both the start and end of the temporal range.
- *Fixed time range*: only show the raster layer if the animation time is within a *Start date* and *End date* range.
- *Fixed Time Range Per Band*: only shows a band when the current animation time is between its *Begin* and *End* date range. This option allows you to either manually set these time ranges for each band or use the ε button to automatically generate datetime values, enabling detailed temporal analysis and visualization. This mode is particularly useful for working with raster layers where each band corresponds to a specific time period, such as NetCDF files.

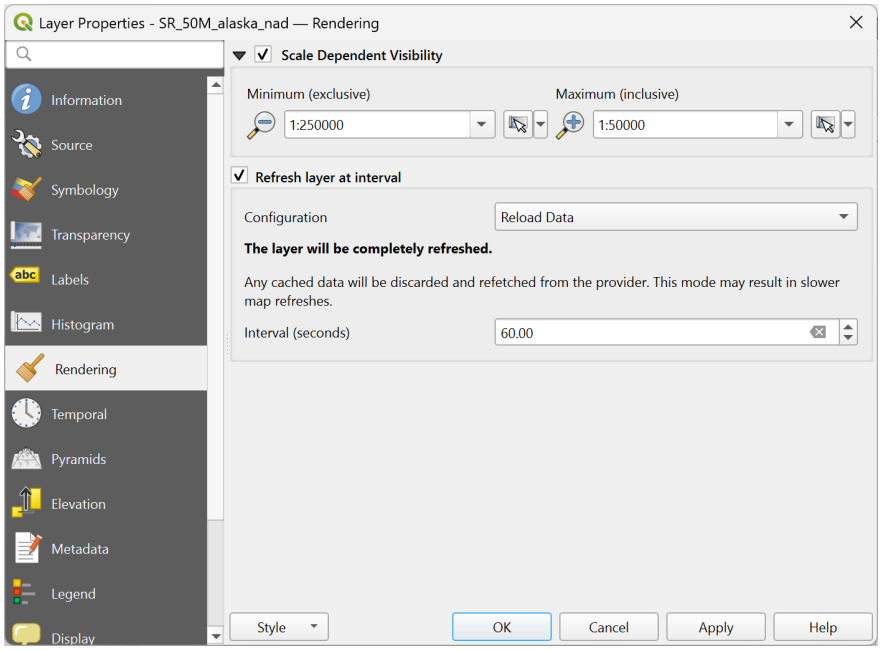



그림 13.17: 래스터 렌더링 속성

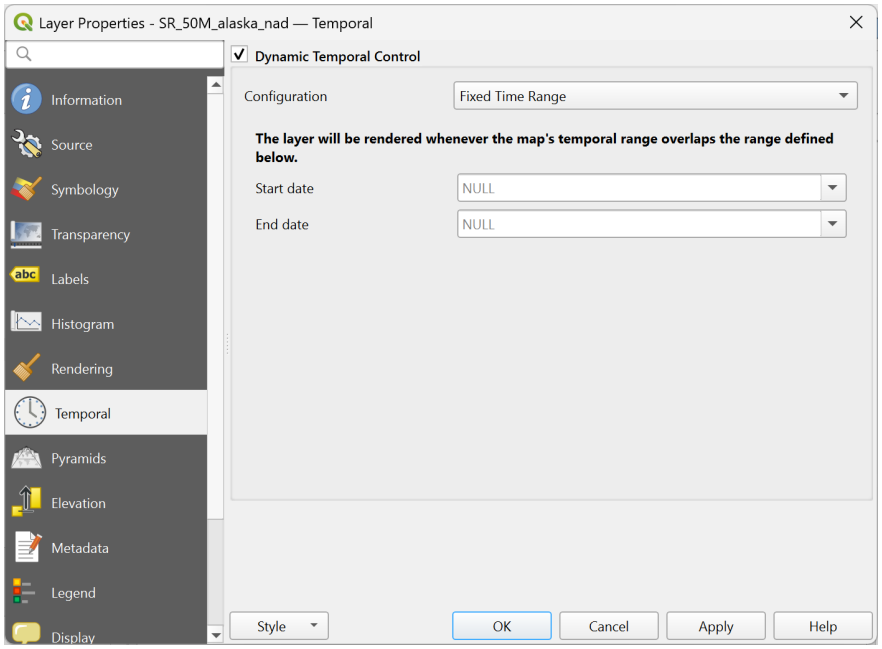



그림 13.18: 래스터 시계열 속성

- *Represents Temporal Values*: interprets each pixel in the raster layer as a datetime value. When this temporal mode is active, pixels that do not fall within the temporal range specified in the render context will be hidden, ensuring that only temporally relevant data is displayed. This mode is effective for:
  - Analyzing land use changes, like observing deforestation patterns.
  - Studying flooding by comparing water coverage across different times.
  - Evaluating movement costs in terrain analysis, for example, using GRASS GIS's r.walk tool to calculate travel costs across a landscape.
- *Redraw layer only*: 각각의 새로운 애니메이션 프레임에서 레이어를 새로 렌더링합니다. 레이어가 렌더링 작업자 설정에 시간 기반 표현식을 사용하는 경우 (예: 래스터 레이어를 페이드 인/페이드 아웃시키기 위해 데이터 정의된 렌더링 작업자 불투명도) 유용합니다.

### 13.1.9 피라미드 속성

고해상도 래스터 레이어는 QGIS 의 맵 탐색 속도를 느리게 할 수 있습니다. 이때 더 낮은 해상도를 가진 데이터 복사본들 (피라미드) 을 생성하면, QGIS 가 확대/축소 수준에 따라 가장 알맞는 해상도를 선택하기 때문에 성능을 크게 개선할 수 있습니다.

피라미드를 생성하려면 원본 데이터가 저장돼 있는 디렉터리에 쓰기 권한을 가지고 있어야만 합니다.

*Resolutions* 목록에서, 해상도를 클릭해서 피라미드 수준을 생성하려는 해상도를 선택하십시오.

*Overview format* 드롭다운 목록에서 **Internal (if possible)** 을 선택한 경우, QGIS 는 피라미드를 원본 파일 내부에 생성하려 시도할 것입니다.

> **참고**
>
> 피라미드 생성 작업이 원본 데이터 파일을 변경시킬 수도 있으며, 생성 후엔 제거할 수 없다는 점을 기억하십시오. 사용자 래스터의'피라미드가 아닌'버전을 보존하고 싶다면, 피라미드를 생성하기 전에 백업하십시오.

**External** 및 **External (Erdas Imagine)** 을 선택한 경우, 원본 래스터 파일이 있는 디렉터리에 피라미드를 동일한 파일명과 `.ovr` 확장자를 가진 파일로 생성할 것입니다.

피라미드를 계산하는 데 다음 몇몇 *Resampling methods* 를 사용할 수 있습니다:

- 최근접 이웃 (Nearest Neighbour)
- 평균 (Average)
- Gauss
- 3 차 (Cubic)
- 3 차 스플라인 (Cubic Spline)
- Laczos
- 모드 (Mode)
- 없음 (None)

마지막으로, *Build Pyramids* 버튼을 클릭해서 생성 과정을 시작하십시오.

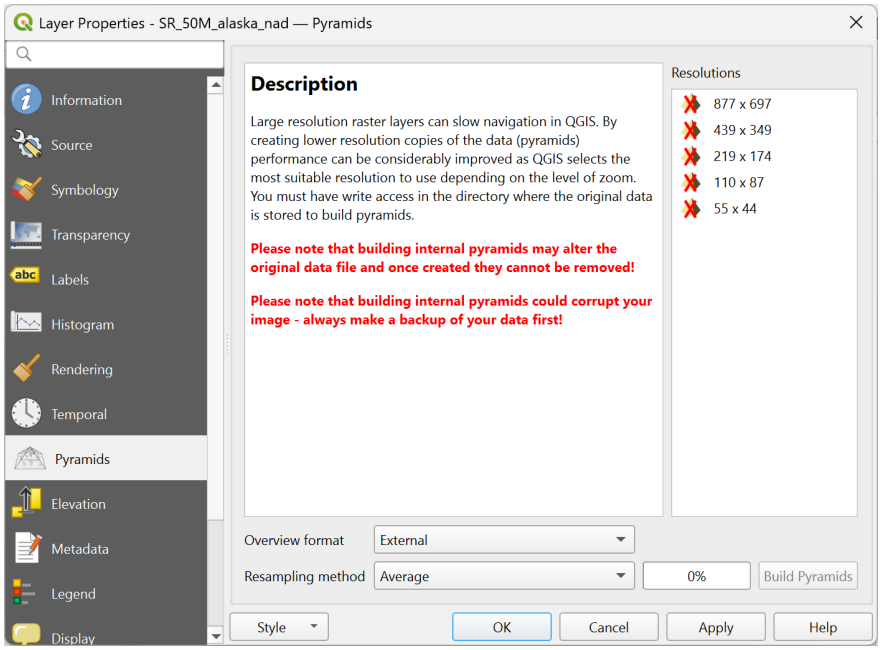



그림 13.19: 래스터 피라미드

### 13.1.10 표고 속성

The *Elevation* tab provides options to control the layer elevation properties within a *3D map view* and its appearance in the *profile tool charts*. Specifically, you can choose to *Disable* this configuration if the layer does not contain elevation data or you can set:

- *Represents Elevation Surface*: whether the raster layer represents a height surface (e.g DEM) and the pixel values should be interpreted as elevations. Choose this option if you want to display a raster in an *elevation profile view*. You will also need to fill in the *Band* to pick values from and can apply a *Scale* factor and an *Offset*.
- *Fixed Elevation Range*: The raster layer (or selected raster band) is associated with a fixed elevation range. This mode can be used when a layer has a single fixed elevation or a range (slice) of elevation values. If a range is specified, pixels will be extruded over this range. You can set the *Lower* and *Upper* elevation range values for the layer, and specify whether the lower or upper *Limits* are inclusive or exclusive.
- *Fixed Elevation Range Per Band*: Each band in the raster can have a fixed elevation range associated with it. This is designed for data sources that expose elevation-related data in bands, such as NetCDF files. For example, a raster with temperature data at different ocean depths. When rendering, the uppermost matching band will be selected and used for the layer's data. This feature is exposed as a user-editable table for raster bands with lower and upper values. Users can either populate the lower and upper values manually or use an *Expression* to auto-fill all band values based on expression. The expression-based fill allows you to design expressions that extract useful information from band names. For example, extracting the depth value from a band name like "Band 001: depth=-5500 (meters)".
- *Dynamic Elevation Range Per Band*: This mode calculates elevation ranges for raster bands dynamically using QGIS expressions. It's ideal for datasets where elevation values follow a consistent pattern across bands (like equally spaced vertical layers), eliminating the need to manually assign fixed elevations. Instead of entering individual values for each band, you define expressions for the *Lower* and *Upper* elevation bounds. These expressions can use variables like `@band`, `@band_name`, or `@band_description` automatically computing the elevation range based on each band's properties.

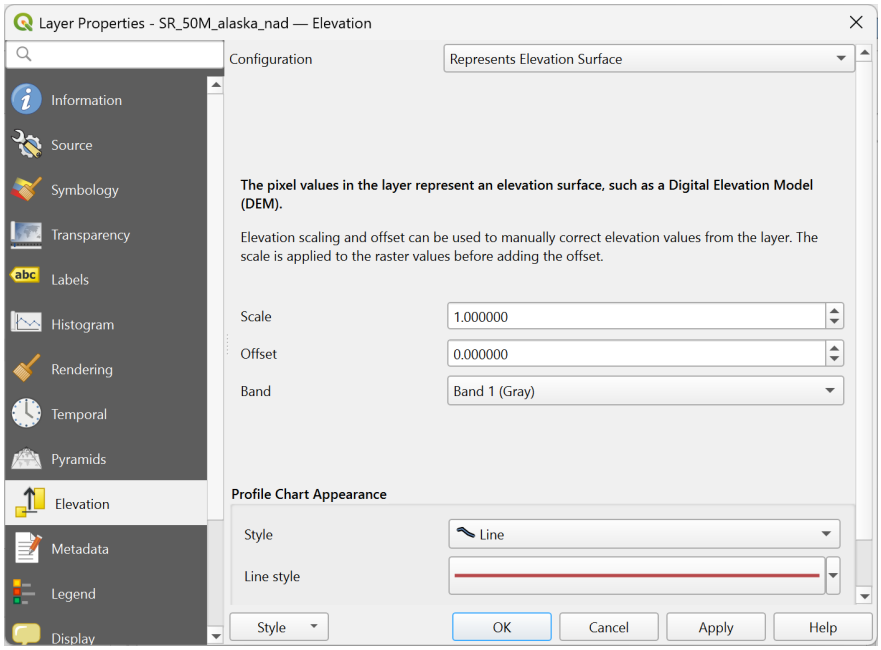



그림 13.20: 래스터 표고 속성

- *Profile Chart Appearance*: 단면 도표에 있는 래스터 표고 데이터의 렌더링을 제어합니다. 단면 *Style* 을 다음 가운데 하나로 설정할 수 있습니다:
  - 라인 스타일 이 적용된 *Line*
  - 표고 곡선 위 (*Fill above*) 또는 아래 (*Fill below*) 에 채우기 심볼을 사용해서 렌더링한 표고 표면입니다. 표면 심볼은 다음을 이용해서 표현됩니다:
    - 채우기 스타일
    - *Limit*: 채우기 표면이 얼마나 높이 올라가게 될지 결정하는 최대 (제각기 최저) 고도를 설정합니다.

### 13.1.11 메타데이터 속성

*Metadata* 탭에 사용자 레이어에 대한 메타데이터 리포트를 생성하고 편집할 수 있는 옵션이 있습니다. 더 자세한 정보는메타데이터 를 참조하세요.

### 13.1.12 범례 속성

*Legend* 탭은레이어 패널 그리고/또는인쇄 조판기 범례 를 위한, 다음과 같은 옵션을 포함하는 고급 설정을 제공합니다:

- 레이어에 적용된 심볼에 따라, 범례에 딱히 인간이 읽을 수 있지도 않고 유용하지도 않은 항목들이 몇 개 나타날 수도 있습니다. *Legend placeholder image* 를 이용하면 *Layers* 패널과 인쇄 조판 범례에 표시되는 대체이미지를 선택 할 수 있습니다.
- *Embedded widgets in Legend* 프레임은 레이어 패널에 있는 레이어 트리 안에 삽입할 수 있는 위젯 목록을 표시합니다. 레이어 작업 시 자주 쓰이는 몇몇 액션 (투명도, 필터링, 선택, 스타일, 기타 등등의 설정) 에 빨리 접근하기 위한 방법으로 사용됩니다.

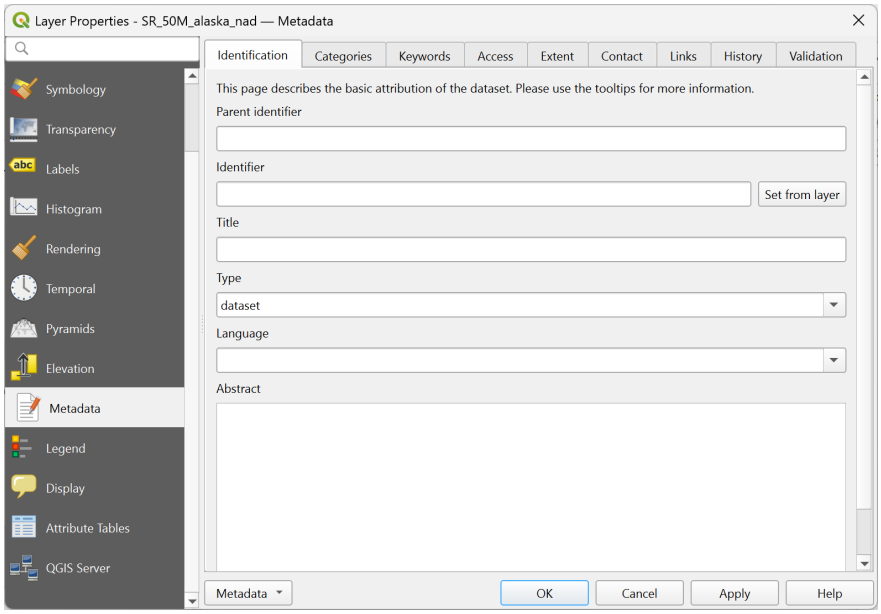



그림 13.21: 래스터 메타데이터

QGIS 가 기본적으로 투명도 위젯을 제공하고는 있지만, 자체 위젯을 등록하고 플러그인이 관리하는 레이어에 사용자 지정 액션을 할당하는 플러그인으로 위젯 목록을 확장할 수 있습니다.

### 13.1.13 표시 속성

*Display* 탭에서 픽셀 식별에 사용되는 HTML 맵 도움말을 환경설정할 수 있습니다:

- *Enable Map Tips* 옵션이 레이어에 맵 도움말을 표시할지 말지를 제어합니다.
- The *HTML Map Tip* provides a complex and full HTML text editor for map tips, mixing QGIS expressions and html styles and tags (multiline, fonts, images, hyperlink, tables, ⋯). You can check the result of your code sample in the *Preview* frame. You can also select and edit existing expressions using the *Insert/Edit Expression* button.

  You might look for expressions located in *Rasters* group or the `@layer_cursor_point` variable in the *Expressions* dialog.

> **참고**
>
> **Understanding the *Insert/Edit Expression* button behavior**
>
> If you select some text within an expression (between "[%"and "%]"), or if no text is selected but the cursor is inside an expression, the whole expression will be automatically selected for editing. If the cursor or a selected text is outside an expression, the dialog opens with the selection.

맵 도움말을 표시하려면:

1. *View* → *Show Map Tips* 메뉴를 선택하거나 *Attributes Toolbar* 의 Show Map Tips 아이콘을 클릭하십시오.
2. 사용자가 대상으로 삼은 레이어가 활성화되어 있는지 그리고 *Enable Map Tips* 옵션이 체크되어 있는지 확인하십시오.

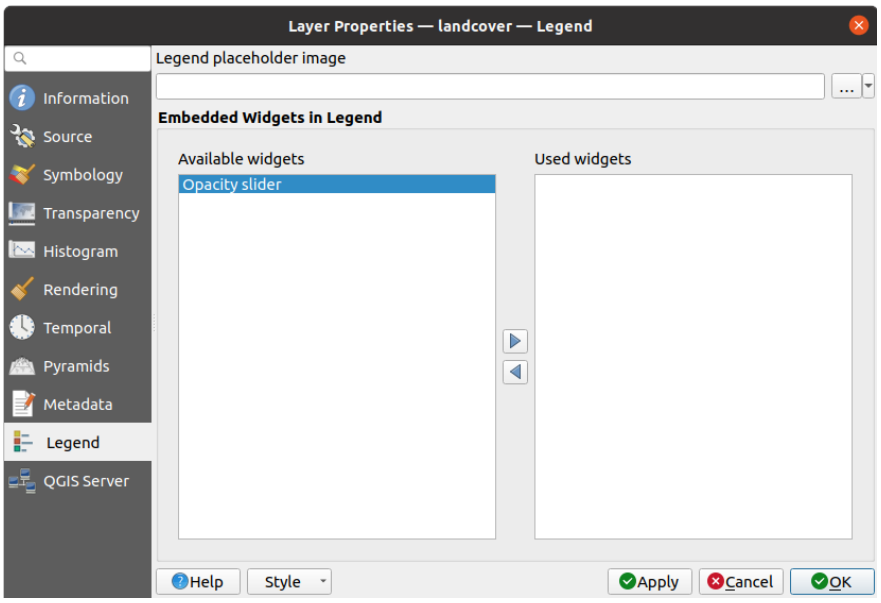



그림 13.22: 래스터 범례

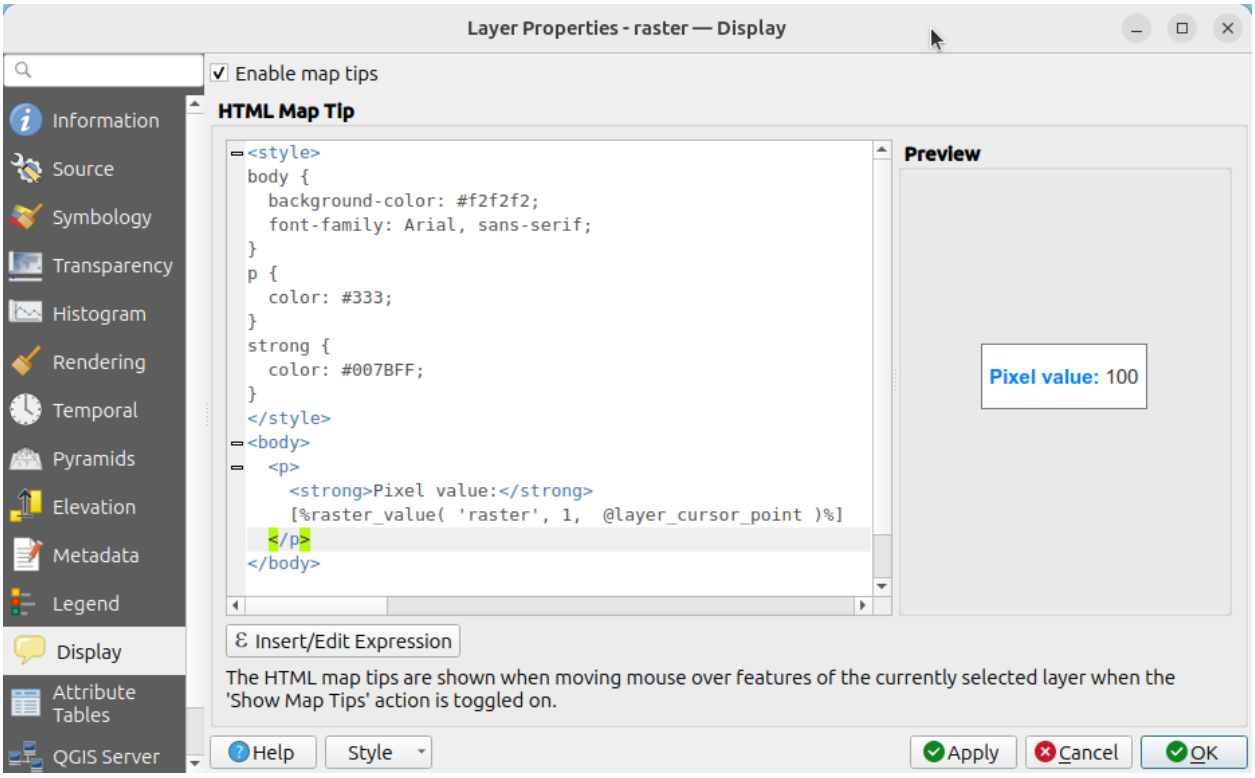



그림 13.23: 래스터 레이어의 맵 도움말

3. 마우스 커서를 픽셀 위로 가져가면, 해당 픽셀에 대응하는 정보가 표시될 것입니다.

맵 도움말은 크로스 레이어 (cross-layer) 기능입니다. 다시 말해 한번 활성화시키면 다시 끌 때까지 계속 프로젝트에서 맵 도움말이 활성화된 레이어들에 모두 적용된다는 뜻입니다.

### 13.1.14 QGIS 서버 속성

*QGIS Server* 탭에서는 QGIS 서버 가 게시한 데이터의 설정들을 환경 설정할 수 있습니다. 이런 환경 설정은 다음을 포함합니다:

- *Description*: provides information to describe the data, such as *Short name*, *Title*, *Abstract*, a *Keyword list*, and a *Data URL* whose *Format* can be in `text/html`, `text/plain` or `application/pdf`.
- *Attribution*: 데이터 제공자를 식별할 수 있는 *Title* 및 *URL* 을 제공합니다.
- *Metadata URL*: *Type* 이 `FGDC` 또는 `TC211` 일 수 있고 *Format* 이 `text/plain` 또는 `text/xml` 일 수 있는, 메타데이터에 대한 *URL* 목록을 제공합니다.
- *Legend URL*: *Format* 이 `image/png` 또는 `image/jpeg` 가운데 하나인, 범례에 대한 *URL* 을 제공합니다.

**참고**

사용자가 게시하려는 래스터 레이어를 다른 웹 서비스가 이미 제공하고 있는 경우, 고급속성들 을 설정할 수 있습니다.

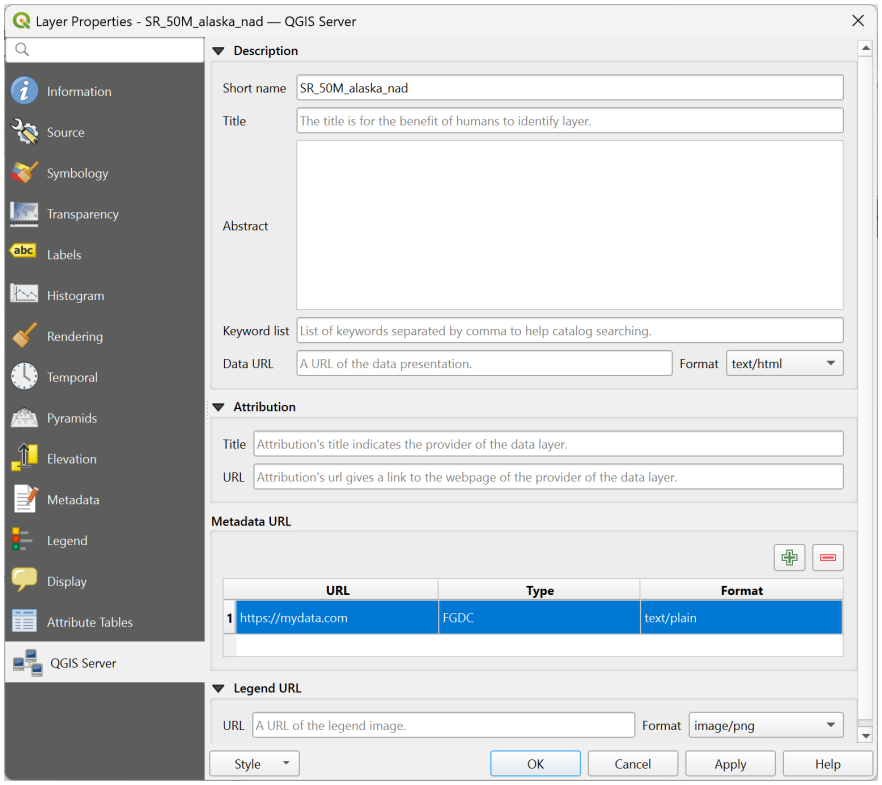



그림 13.24: 래스터 속성에서의 QGIS 서버

### 13.1.15 Attribute Tables

Similar to vector layers, raster layers in QGIS can have associated Raster Attribute Tables (RATs) that store additional information about raster pixel values.

A Raster Attribute Table links raster values (or value ranges) to descriptive attributes such as class names, pixel counts, and colors. RATs are commonly used for classified rasters, color tables, and histogram information, helping QGIS interpret how raster values should be displayed and classified.

The *Attribute Tables* tab allows you to view and edit raster attribute data. To access the attribute table of a raster layer, right-click the layer in the layers panel, open *Properties*⋯ and select *Attribute Tables*.

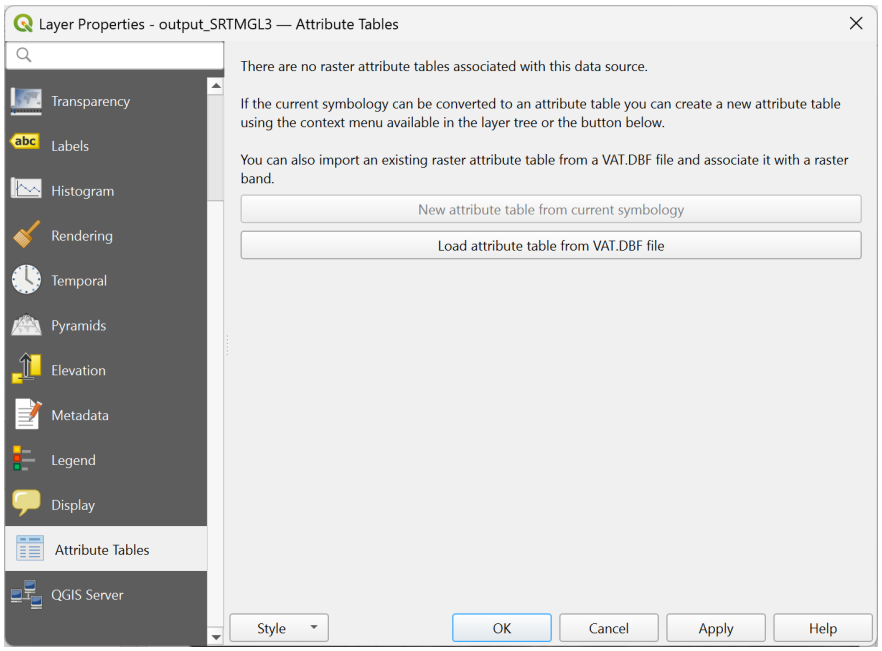



그림 13.25: Accessing the raster attribute table

When opening the *Attribute Tables* for a raster layer that does not have an associated Raster Attribute Table, QGIS displays a notice informing that no attribute table is currently linked to the data source. In this state, the interface provides two options to define and associate a new attribute table:

- *New attribute table from current symbology*: will appear if your raster layer has a classified symbology applied. Otherwise, this option is disabled. Applying classified symbology also triggers an option in the layer context menu to *Create Raster Attribute Table*. Click this button and set the following options in the *New Raster Attribute Table* dialog.
  - *Managed by the data provider*: the attribute table will be saved and managed by the data provider (if supported), overwriting any existing table for the raster band used by the current style. Depending on the data provider, the attribute table will be embedded in the main raster file or saved into a sidecar file managed by the data provider.
  - *Sidecar VAT.DBF file*: saves the attribute table into a sidecar VAT.DBF file. The resulting file will not be associated with any particular band.
- *Load attribute table from VAT.DBF file*: retrieves the attribute table from an external DBF file. *Load Raster Attribute Table* dialog will appear to let you browse and select the DBF file and associate it with the raster band. The action also can be done by right-clicking the raster layer in the layers panel and selecting *Load Raster Attribute Table*

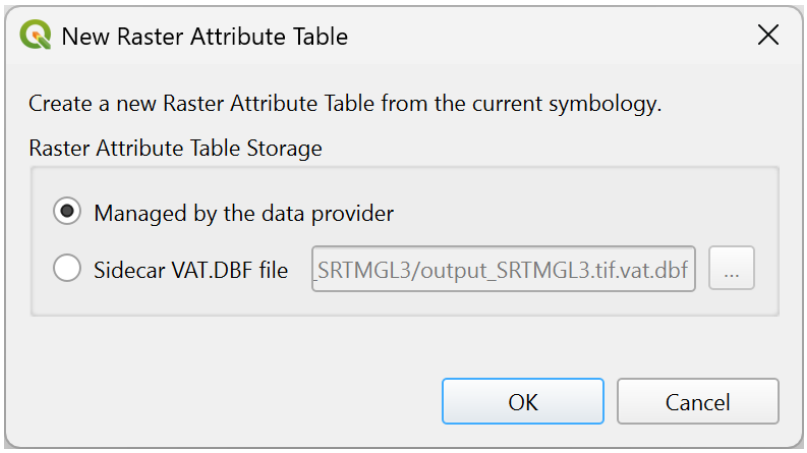



그림 13.26: Create new attribute table from current symbology

*from VAT.DBF*. *Open the newly loaded attribute table* allows you to directly open the attribute table editor after loading the table.

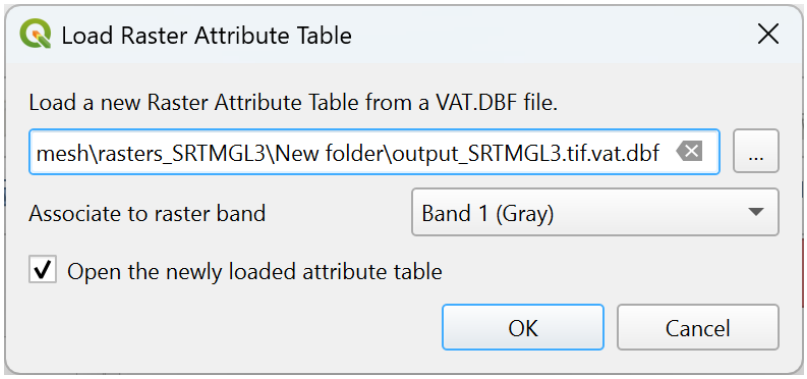



그림 13.27: Load attribute table from VAT.DBF file

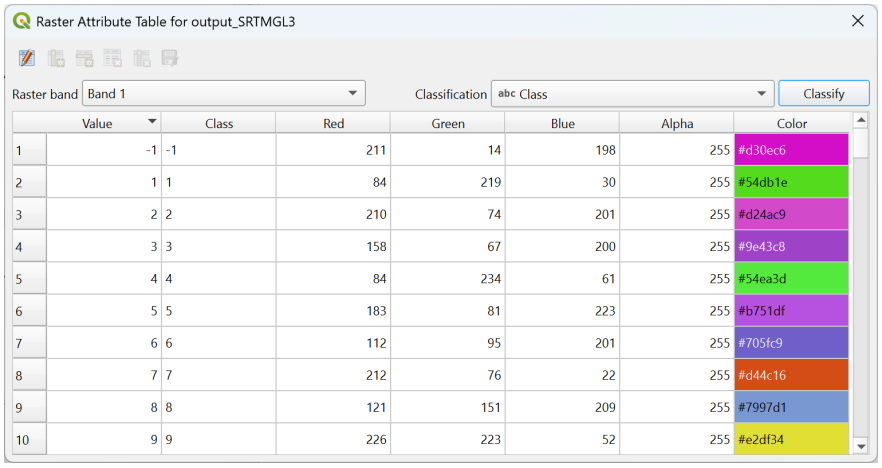
### Editing raster attribute tables

When set, a raster attribute table can be accessed from the *Attribute tables* tab of the layer properties dialog or by selecting *Open raster attribute table* from the layer contextual menu in *Layers* panel. This dialog shows a table whose rows correspond to the different pixel values or value ranges classified in a given band of the raster layer. Depending on the type of classification, a row is described by:

- for an exact values classification (e.g., paletted/unique values): the band name and the applied *Color*. Additional fields of the *Red*, *Green*, *Blue* and *Opacity* of the color are displayed.
- for range values classification (e.g., singleband pseudocolor): the range of values defined by their *Min*, *Max*, the associated label (*Class*) and color ramp interpolated over the values (*Color*). Additional fields of the minimum and maximum of *Red*, *Green*, *Blue* and *Opacity* of the color ramp are displayed.

You can also customize the raster attribute table by adding new fields for classification and group different value ranges accordingly, modifying value ranges and their associated colors or labels, adding new rows, ⋯The new attribute table can be used to update the current layer symbology.

At the top of the dialog, a set of tools allows to edit and save changes to the attribute table:




| | Value | Class | Red | Green | Blue | Alpha | Color |
|---|---|---|---|---|---|---|---|
| 1 | -1 | -1 | 211 | 14 | 198 | 255 | #d30ec6 |
| 2 | 1 | 1 | 84 | 219 | 30 | 255 | #54db1e |
| 3 | 2 | 2 | 210 | 74 | 201 | 255 | #d24ac9 |
| 4 | 3 | 3 | 158 | 67 | 200 | 255 | #9e43c8 |
| 5 | 4 | 4 | 84 | 234 | 61 | 255 | #54ea3d |
| 6 | 5 | 5 | 183 | 81 | 223 | 255 | #b751df |
| 7 | 6 | 6 | 112 | 95 | 201 | 255 | #705fc9 |
| 8 | 7 | 7 | 212 | 76 | 22 | 255 | #d44c16 |
| 9 | 8 | 8 | 121 | 151 | 209 | 255 | #7997d1 |
| 10 | 9 | 9 | 226 | 223 | 52 | 255 | #e2df34 |

그림 13.28: Raster Attribute Table

- Edit Attribute Table turns the table into edit mode. If no specific row or field is selected, only the Add Column… option is available.
- Add Column… opens the *Add Column* dialog to add a new column to the raster attribute table. The dialog lets you define:
  - *Column Type*, which can be:
    - *Standard column*
    - *Color RGBA*
    - *Color ramp (RGBA minimum and RGBA maximum)*
  - *Column Definition*, including:
    - *Name* of the column (mandatory)
    - *Usage*, which can be:
      - *Pixel Count*: stores the number of pixels belonging to a given value or value range.
      - *Generic*: general-purpose column with no predefined meaning.
      - *Name*: stores the class name or label associated with a raster value or range.
    - *Data type*, which can be:
      - *String*
      - *Integer*
      - *Long Integer*
      - *Double*
  - *Insertion point*, defining whether the new column is inserted *Before* or *After* a selected existing column.
- *Add Row*… opens the *Add Row* dialog to add a new row to the raster attribute table. Choose the *Insertion point* as *Before current row* or *After current row*.
- Remove Row removes the selected row from the raster attribute table.

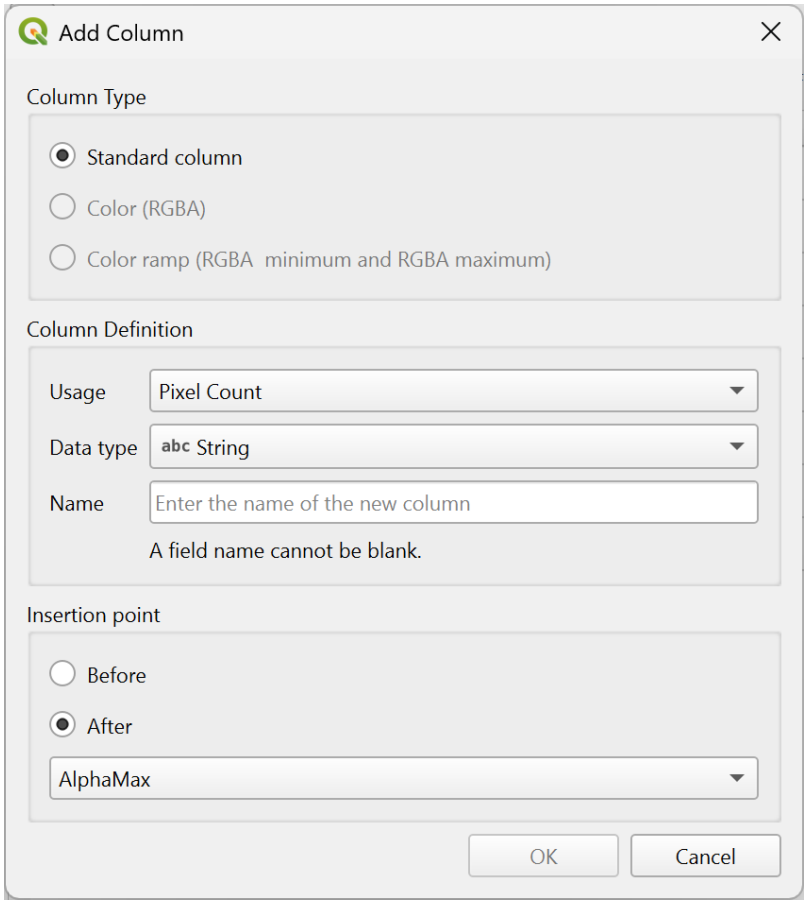



그림 13.29: Add Column dialog

-  Remove Column removes the selected column from the raster attribute table.
-  Save Changes saves any modifications made to the raster attribute table.

The next row provides options for controlling the displayed information in the table:

- *Raster band*: allows selecting which raster band the attribute table applies to. Raster layers may contain multiple bands (e.g., multispectral data). Each band can have its own Raster Attribute Table.
- *Classification*: allows selecting the fields used for classifying the raster layer. Keep in mind that the existing symbology for the raster will be replaced by a new symbology from the attribute table and any unsaved changes to the current symbology will be lost. Make sure to save your symbology style if needed before applying a classification.

### 13.1.16 래스터 셀 식별하기

 피처 식별 도구는 래스터 레이어에 있는 특정 포인트에 대한 정보를 볼 수 있게 해줍니다.

 *Identify features* 도구를 사용하려면:

1. 레이어 패널에서 래스터 레이어를 선택합니다.
2. 툴바에 있는 *Identify features* 도구를 클릭하거나, 또는 `Ctrl+Shift+I` 단축키를 누릅니다.
3. Click on the point in the raster layer that you want to identify. The pixel will get highlighted.

The Identify Results panel will open in its default `Tree` view and display information about the clicked point. Formatting of the results vary depending on the provider of the layer. For example:

- For a local raster layer: below the name of the layer, you have on the left the band(s) of the clicked pixel, and on the right their respective value.
- For a remote layer such as WMS, a *Format* menu allows you to select whether the information should be displayed as *HTML*, *Feature* or *Text*.

These values can also be rendered (from the *View* menu located at the bottom of the panel) in:

- `Table` 뷰 - 표에 식별된 피처와 그 값에 대한 정보를 정리합니다.
- `Graph` 뷰 - 그래프에 식별된 피처와 그 값에 대한 정보를 정리합니다.

픽셀 속성 아래에서 다음과 같은 *Derived* 파생 정보를 찾아볼 수 있습니다:

- 클릭한 포인트의 `X` 및 `Y` 좌표값
- Column and row of the point clicked (pixel) when compatible

## 13.2 래스터 분석

### 13.2.1 래스터 계산기

*Raster* 메뉴에 있는 *Raster Calculator* 는 기존 래스터 픽셀값을 기반으로 계산을 수행할 수 있습니다. (그림 13.30 참조) 계산 결과는 GDAL 지원 포맷의 새 래스터 레이어로 작성됩니다.

*Raster bands* 목록은 불러온 래스터 레이어들 가운데 사용할 수 있는 모든 레이어들을 담고 있습니다. 래스터 계산기 표현식 필드에 래스터를 추가하려면, 필드 목록에서 레이어 이름을 더블클릭하십시오. 그 다음 연산자를 이용해서 계산 표현식을 구성하거나, 또는 표현식란에 직접 입력할 수 있습니다.

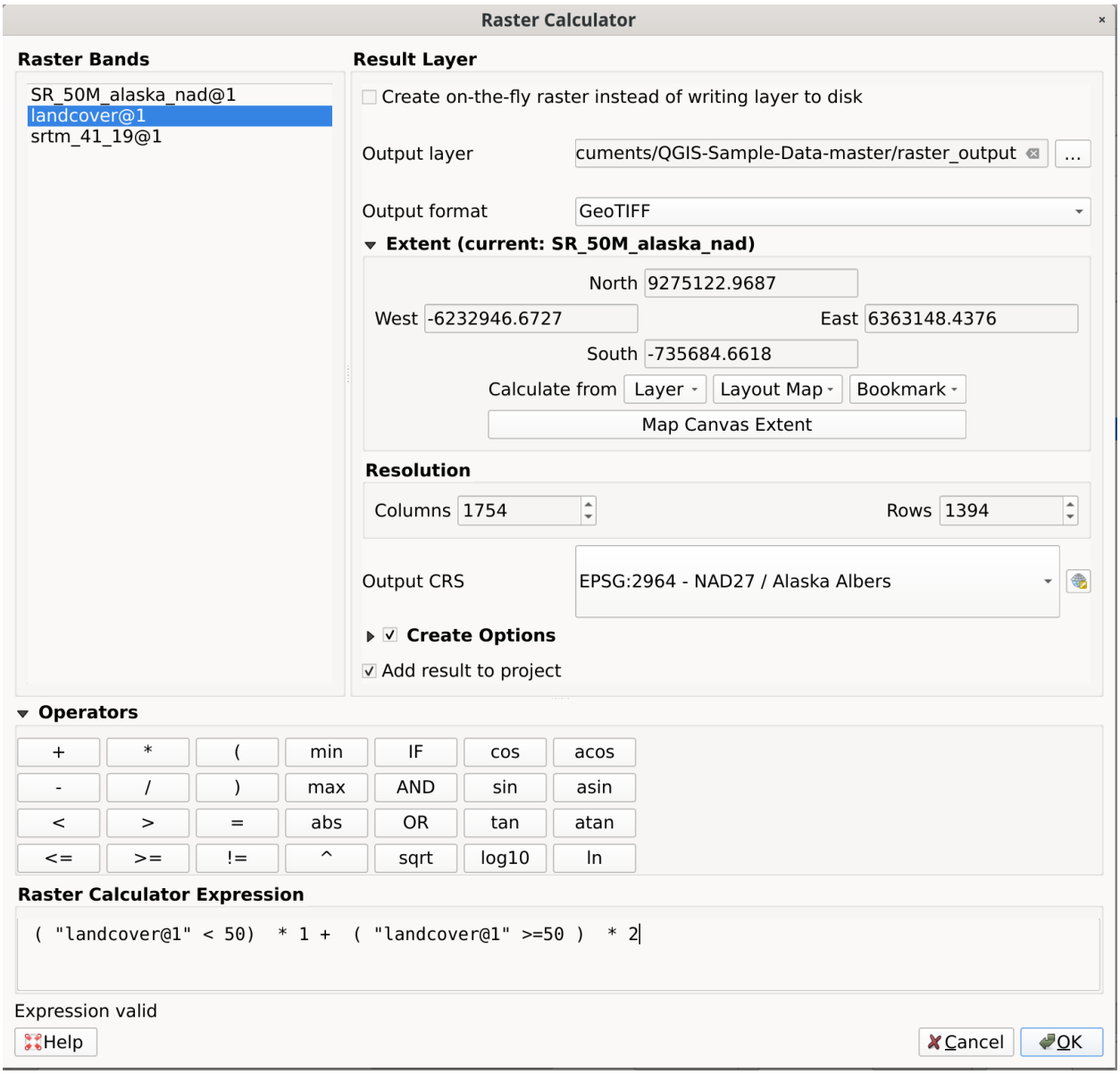



그림 13.30: 래스터 계산기

*Result layer* 부분에 산출 레이어를 정의해줘야 합니다. 다음을 정의할 수 있습니다:

- ☑ *Create on-the-fly raster instead of writing layer to disk*:
  - 이 옵션을 체크 해제하는 경우, 디스크 상에 산출물을 새 평문 텍스트 파일로 저장합니다. *Output layer* 경로 및 *Output format* 은 필수입니다.
  - 이 옵션을 체크하는 경우, 가상 래스터 레이어를 – 예를 들면 URI 로 정의되는 래스터 레이어를 레이어 픽셀을 실시간 (on-the-fly) 으로 계산해서 생성합니다. 디스크 상에 새 파일로 저장되는 것이 아니라, 이 가상 레이어는 계산에 사용된 래스터에 계속 연결되어 있기 때문에 계산에 사용된 래스터를 삭제하거나 이동시키면 가상 레이어에도 오류가 발생합니다. *Layer name* 을 지정할 수 있습니다. 지정하지 않으면 계산 표현식을 그대로 사용합니다. 프로젝트로부터 가상 레이어를 제거하면 삭제되며, *Export* → *Save as* …레이어 컨텍스트 메뉴를 이용하면 디스크 상에 파일로 저장할 수 있습니다.
- Define the *Spatial extent* of the calculation using the *extent selector* widget.
- 열 및 행 개수를 이용해서 레이어의 *Resolution* 을 설정할 수 있습니다. 입력 레이어의 해상도와 다른 경우, 값을 최근접 이웃 알고리즘으로 리샘플링할 것입니다.
- Enable the ☑ *Create Options* to define additional raster creation options. These options let you control how the output file is structured and compressed, including format, compression method, and other driver-specific settings. See more at *Raster driver options*.
- ☑ *Add result to project* 옵션을 체크하는 경우, 범례 영역에 산출 레이어를 자동으로 추가할 것이고, 추가 후 가시화시킬 수 있습니다. 가상 래스터의 경우 기본적으로 체크되어 있습니다.

*Operators* 부분에 사용할 수 있는 모든 연산자를 담고 있습니다. 래스터 계산기 표현식란에 연산자를 추가하려면, 적절한 버튼을 클릭하십시오. 산술 계산 (+, –, *, …) 및 삼각함수 (`sin`, `cos`, `tan`, …) 를 사용할 수 있습니다. 조건 표현식 (=, `!=`, <, >=, …) 은 거짓인 경우 0 을 또는 참인 경우 1 을 반환하기 때문에, 다른 연산자 및 함수와 함께 사용할 수 있습니다.

**더 보기**

래스터 계산기 및래스터 계산기 *(가상)* 알고리즘

### 래스터 계산기 표현식

#### 대화창

*Raster calculator expression* 대화창은 일련의 래스터 레이어들 사이에 픽셀을 계산하기 위한 표현식을 작성할 수 있는 수단을 제공합니다.

- *Layers*: 범례에 불러온 모든 래스터 레이어의 목록을 표시합니다. 이 목록을 사용해서 (레이어명을 더블 클릭하면) 표현식란에 해당 레이어를 추가할 수 있습니다. 래스터 레이어는 `layer_name@band_number` 처럼 이름 및 밴드 번호로 참조됩니다. 예를 들어 `DEM` 이라는 레이어의 첫 번째 밴드는 `DEM@1` 로 참조될 것입니다.
- *Operators*: 픽셀을 조작하기 위한 계산 연산자 여러 개를 담고 있습니다:
  - 산술: `+`, `–`, `*`, `sqrt`, `abs`, `ln`, …
  - 삼각함수: `sin`, `cos`, `tan`, …
  - 비교: `=`, `!=`, <, >=, …
  - 논리: `IF`, `AND`, `OR`, `(`, `)`
  - 통계: `min`, `max`

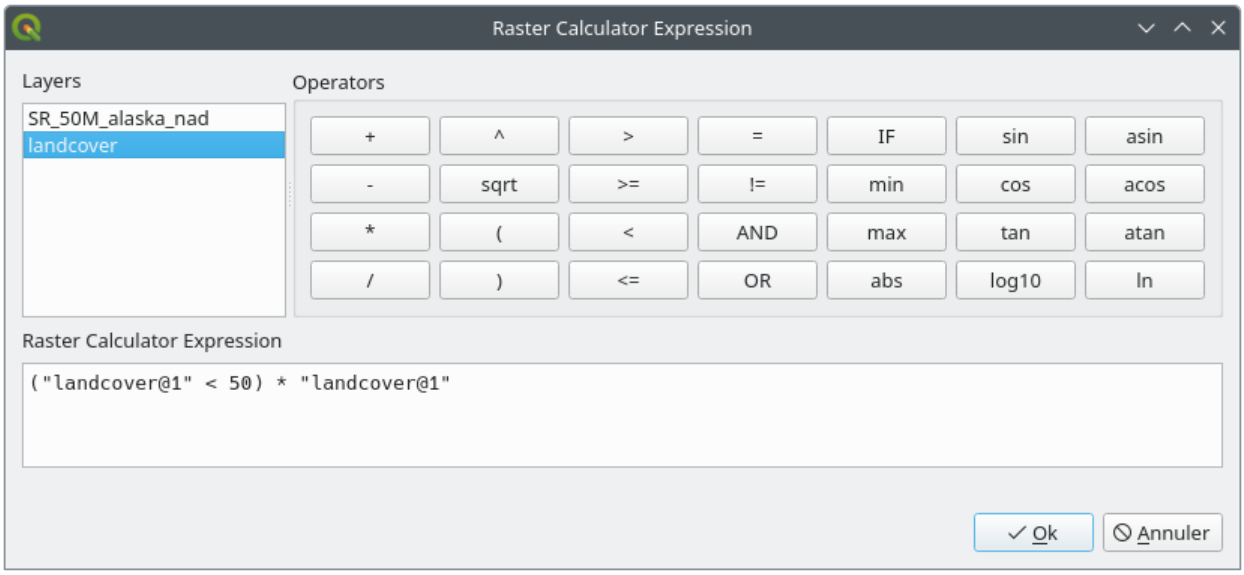



그림 13.31: 래스터 표현식 계산기

래스터 계산기 표현식 입력란에 연산자를 추가하려면, 알맞은 버튼을 클릭하십시오.

- *Raster calculator expression* 부분에 표현식을 작성합니다.

### 예제

#### 표고값을 미터 단위에서 피트 단위로 변환하기

미터 단위의 표고 래스터에서 피트 단위의 래스터를 생성하려면, 미터를 피트로 변환하는 인수 3.28 을 사용해야 합니다. 다음은 그 표현식입니다:

```
"elevation@1" * 3.28
```

#### 마스크 이용하기

래스터의 일부분을 가리고 싶은 경우 – 그러니까 예를 들어 표고 0 미터 이상에만 관심이 있을 경우 – 다음 표현식을 이용해서 마스크를 생성하고 그 결과를 래스터에 적용하는 작업을 한 번에 할 수 있습니다:

```
("elevation@1" >= 0) * "elevation@1"
```

즉 그 값이 0 이상인 모든 셀에 대해 조건 표현식이 1 을 반환하기 때문에 1 을 곱해도 원본 값을 유지합니다. 반면 값이 0 미만인 셀에 대해서는 0 을 반환하기 때문에 래스터 값이 0 이 됩니다. 이렇게 실시간으로 마스크를 생성합니다.

#### 래스터 범주화하기

래스터를 범주화하고 싶은 경우 – 그러니까 예를 들어 표고값을 2 단계로 나누고 싶은 경우, 다음 표현식을 이용해서 1 과 2 두 개의 값을 지닌 래스터를 한 번에 생성할 수 있습니다:

```
("elevation@1" < 50) * 1 + ("elevation@1" >= 50) * 2
```

즉 그 값이 50 미만인 모든 셀의 값은 1 이 되고, 50 이상인 모든 셀의 값은 2 가 됩니다.

또는 `IF` 연산자를 사용할 수 있습니다.

```
if ( elevation@1 < 50 , 1 , 2 )
```

CHAPTER 14

# 메시 데이터 작업

## 14.1 메시란 무엇일까요?

메시 (mesh) 란 일반적으로 시계열 및 기타 요소들을 가지고 있는 비구조적 그리드를 말합니다. 공간 요소는 2 차원 또는 3 차원 공간에 있는 꼭짓점, 변 (edge) 및 면 (face) 의 집합을 담고 있습니다:

- **꼭짓점**: XY(Z) 포인트 (레이어 좌표계)
- **변**: 연결된 두 꼭짓점
- **면**: 면이란 닫힌 도형 (closed shape) 을 형성하는 변 (edge) 의 집합을 말합니다. 일반적으로 삼각형 또는 사변형 (quadrilateral) 이며, 드물게 더 많은 꼭짓점을 가진 폴리곤도 면이 될 수 있습니다.

위의 정의에 따라서, 메시 레이어는 서로 다른 구조 유형들을 가질 수 있습니다:

- 1 차원 메시: 꼭짓점 및 변으로 이루어집니다. 변이란 꼭짓점 두 개를 잇는 선분으로, (스칼라 또는 벡터) 데이터를 할당시킬 수 있습니다. 예를 들면 도시 하수 시스템 모델링에 1 차원 메시 네트워크를 활용할 수 있습니다.
- 2 차원 메시: 삼각면, 정규 사분면 (regular quads) 또는 비구조적 사분면 (unstructured quads) 으로 이루어집니다.
- 3 차원 레이어드 메시: 수직 좌표를 통해 수직 방향 (수준) 으로 각각 압출 (extrusion) 된 2 차원 비구조적 메시 여러 개의 층위로 이루어집니다. 각 수직 수준에서 꼭짓점과 면의 위상은 동일합니다. 메시 정의는 (수직 수준 압출은) 일반적으로 시간에 따라 변화할 수도 있습니다. 3 차원 메시 데이터는 일반적으로 부피 중심 또는 어떤 파라미터 함수로 정의됩니다.

메시는 공간 구조에 관한 정보를 제공합니다. 거기에 더해, 메시는 모든 꼭짓점에 값을 할당하는 데이터셋 (그룹) 을 보유할 수 있습니다. 예를 들면, 다음 그림처럼 번호가 붙은 꼭짓점을 가진 삼각형 메시의 경우:

각 꼭짓점이 서로 다른 (일반적으로 여러 수량의) 데이터셋을 저장할 수 있으며, 이 데이터셋들은 시계열 차원도 보유할 수 있습니다. 따라서 단일 파일이 여러 데이터셋을 담고 있을 수도 있습니다.

다음 테이블을 보면 메시 데이터셋에 어떤 데이터를 저장할 수 있는지 알 수 있습니다. 테이블 열은 메시 꼭짓점들의 인덱스를 나타내며, 각 행은 데이터셋 하나를 나타냅니다. 데이터셋은 서로 다른 데이터 유형들을 보유할 수 있습니다. 이 테이블의 경우, 특정 시간 (t1, t2, t3) 에 표고 10m 에서 측정한 풍속을 저장하고 있습니다.

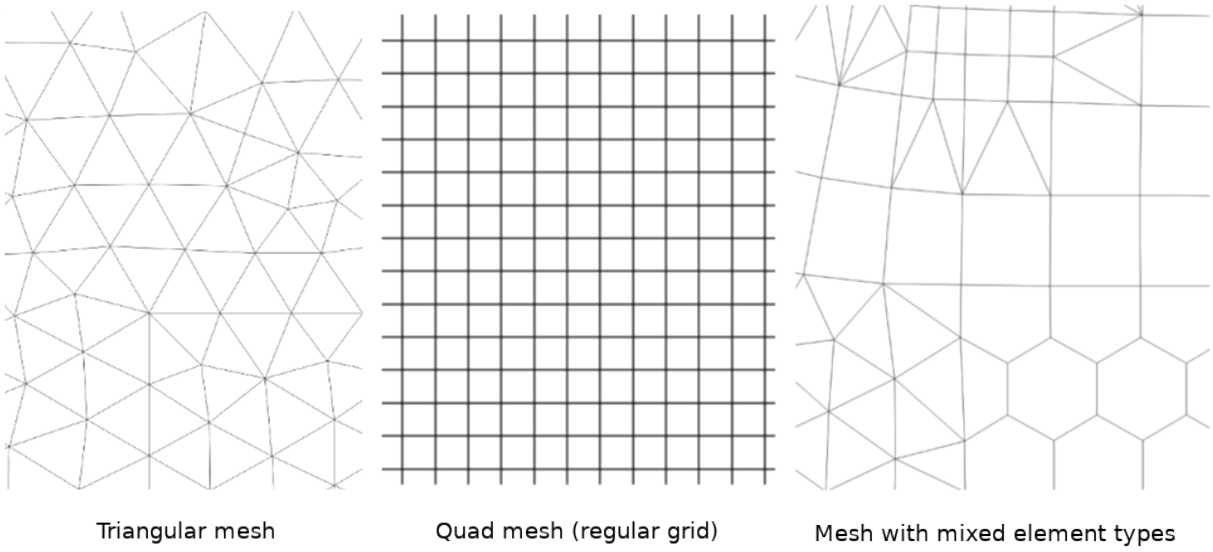



그림 14.1: 서로 다른 메시 유형

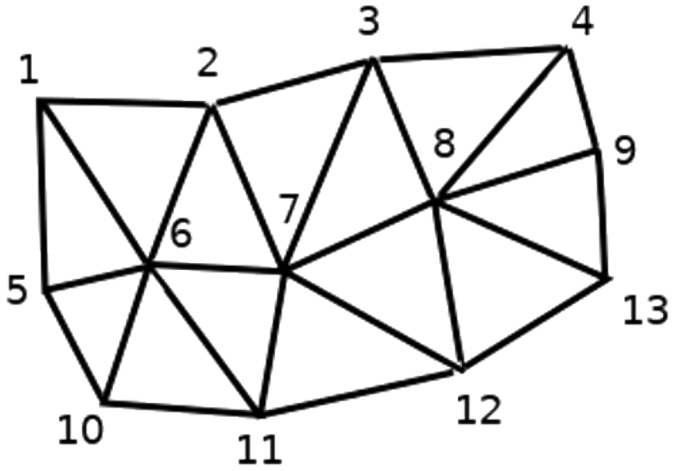



그림 14.2: 번호가 붙은 꼭짓점을 가진 삼각형 그리드

비슷한 방식으로, 메시 데이터셋이 각 꼭짓점에 벡터값도 저장할 수 있습니다. 예를 들어 지정한 타임 스탬프 순간의 풍향처럼 말이죠:

표 14.1: 메시 데이터셋의 예시

| 표고 10m 바람 | 1 | 2 | 3 | … |
|---|---|---|---|---|
| t1 시간의 표고 10m 풍속 | 17251 | 24918 | 32858 | … |
| t2 시간의 표고 10m 풍속 | 19168 | 23001 | 36418 | … |
| t3 시간의 표고 10m 풍속 | 21085 | 30668 | 17251 | … |
| … | … | … | … | … |
| t1 시간의 표고 10m 풍향 | [20,2] | [20,3] | [20,4.5] | … |
| t2 시간의 표고 10m 풍향 | [21,3] | [21,4] | [21,5.5] | … |
| t3 시간의 표고 10m 풍향 | [22,4] | [22,5] | [22,6.5] | … |
| … | … | … | … | … |

값에 색상을 할당하고 (단일 밴드 의사색상 래스터 렌더링 작업과 비슷합니다) 메시 위상에 따라 꼭짓점 사이의 데이터를 보간해서 데이터를 가시화할 수 있습니다. 몇몇 수량이 단순 스칼라 (scalar) 값이 아니라 2 차원 벡터인 (예: 풍향) 경우도 흔합니다. 이런 수량의 경우 방향을 나타내는 화살표를 표시하는 편이 바람직합니다.

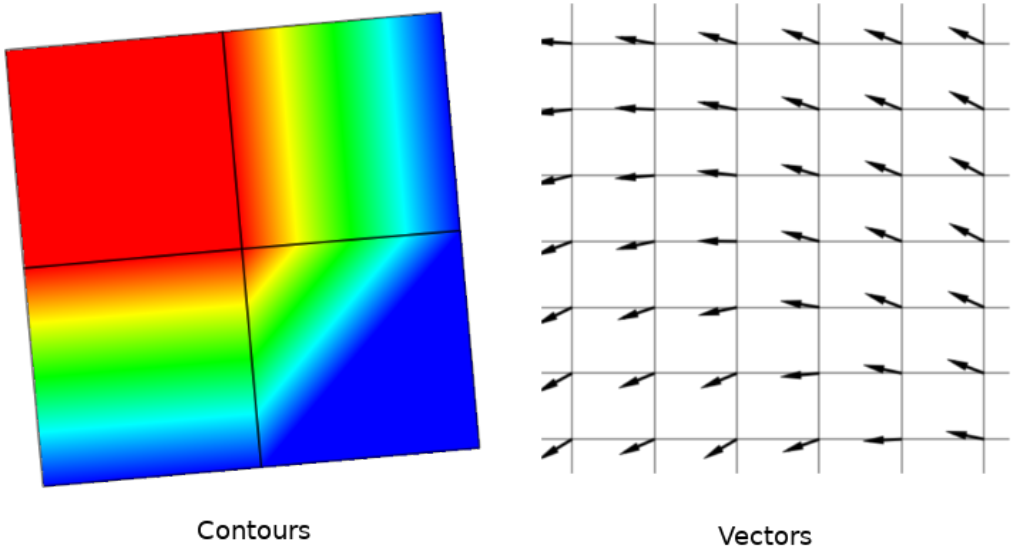



그림 14.3: 메시 데이터를 가시화할 수 있는 방법들

## 14.2 지원 포맷

QGIS accesses mesh data using the MDAL drivers, and natively supports a variety of formats. Whether QGIS can edit a mesh layer depends on the format and the mesh structure type.

*Data Source Manager* 대화창에 있는 *Mesh* 탭을 통해 QGIS 로 메시 데이터셋을 불러올 수 있습니다. 더 자세한 내용은메시 레이어 불러오기 를 읽어보세요.

## 14.3 메시 데이터셋 속성

메시 레이어의 *Layer Properties* 대화창은 레이어의 데이터셋 그룹과 각 그룹의 렌더링을 (활성 데이터셋 그룹, 심볼, 2 차원 및 3 차원 렌더링 등등을) 관리할 수 있는 일반 설정을 제공합니다. 또 레이어에 대한 정보를 제공하기도 합니다.

*Layer Properties* 대화창을 열려면:

- *Layers* 패널에서 레이어를 더블 클릭하거나, 오른쪽 클릭한 다음 컨텍스트 메뉴에서 *Properties*…를 선택하십시오.
- 레이어를 선택한 다음 *Layer* → *Layer Properties*…메뉴를 클릭하십시오.

메시 *Layer Properties* 대화창은 다음 탭들로 이루어져 있습니다:

표 14.2: 메시 레이어 속성 탭들

| | | |
|---|---|---|
| 정보 | 소스 | *Symbology* [1] |
| *Labels* [1] | *3D View* [1] | 렌더링 |
| 시계열 | 표고 | 메타데이터 |

> **참고**
>
> 대화창 하단에 있는 *Style* 메뉴를 이용해서 메시 레이어 속성 대부분을 `.qml` 파일로 저장하거나 `.qml` 파일로부터 불러올 수 있습니다. 자세한 내용은사용자 지정 스타일 관리하기 를 참조하세요.

### 14.3.1 정보 속성

*Information* 탭은 읽기 전용으로, 현재 레이어에 대한 요약 정보 및 메타데이터를 한 눈에 볼 수 있는 흥미로운 장소입니다. 다음과 같은 정보를 제공합니다:

- 프로젝트 이름, 소스 경로, 보조 파일 목록, 최신 저장 시간 및 용량, 사용하는 제공자 등의 일반 정보
- custom properties, used to store in the active project additional information about the layer. Default custom properties may include *layer notes*. More properties can be created and managed using PyQGIS, specifically through the setCustomProperty() method.
- 해당 레이어의 제공자 기반: 범위, 꼭짓점, 면, 변 그리고/또는 데이터셋 그룹 개수
- 좌표계: 이름, 단위, 메소드, 정확도, 참조 (예: 정적인지 동적인지 여부)
- 채워진메타데이터 로부터 추출: 접근, 범위, 링크, 연락처, 이력, …

[1] Also available in the *Layer styling panel*

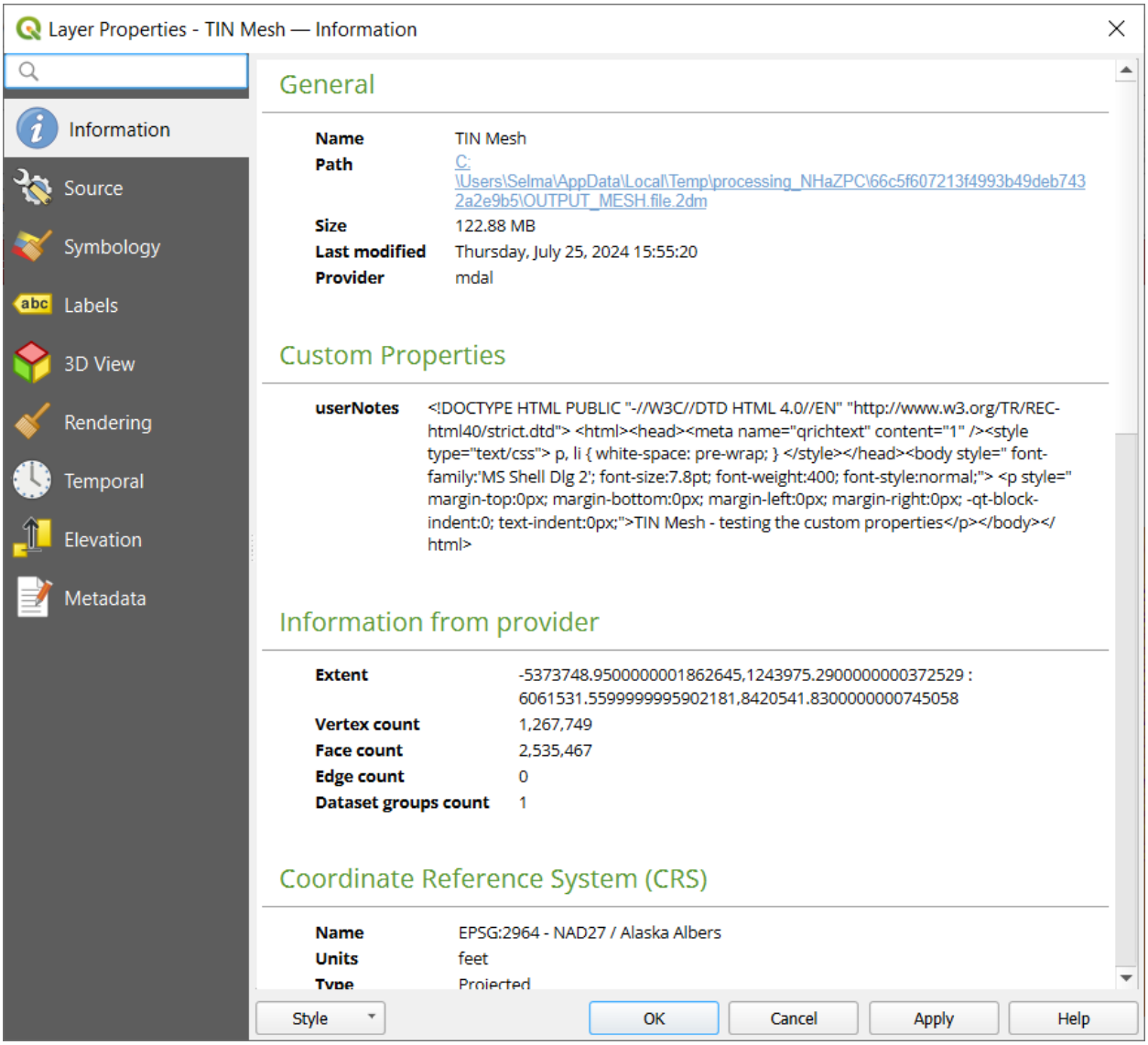



그림 14.4: 메시 레이어 정보 속성

### 14.3.2 소스 속성

*Source* 탭은 선택한 메시에 대한 다음과 같은 기본 정보를 표시합니다:

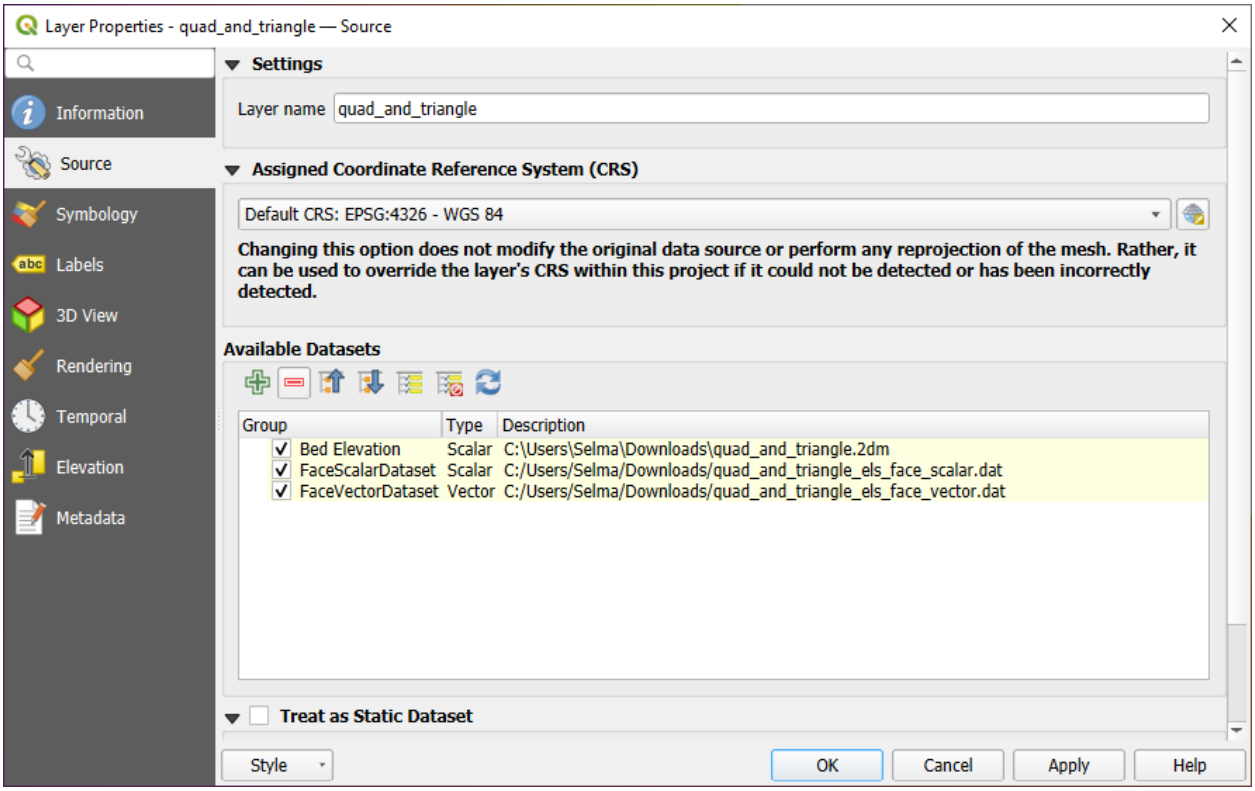



그림 14.5: 메시 레이어 소스 속성

- *Layers* 패널에 표시할 레이어명
- 좌표계 설정: 레이어에할당된 좌표계 를 표시합니다. 드롭다운 목록에서 최근 사용된 좌표계를 선택하거나 *Select CRS* 버튼을 (좌표계 선택기 참조) 클릭하면 레이어 좌표계를 변경할 수 있습니다. 레이어에 적용된 좌표계가 틀렸거나 적용된 좌표계가 없는 경우에만 변경하십시오.
- *Available datasets* 프레임은 메시 레이어에 있는 모든 데이터셋 그룹 (및 하위 그룹) 들의 목록을 담고 있습니다. 트리 뷰에서 각 그룹의 유형 및 설명을 볼 수 있습니다. (예를 들어 파일에 데이터를 저장하는) 정규 데이터셋과 (실시간으로 계산되는) 가상 데이터셋 둘 다 목록화됩니다.
  - Use the Assign Extra Dataset to Mesh button to add more groups to the current mesh layer. You can add dataset group to a mesh layer with the same name, but not from the same URI. Dataset group names are automatically renamed to "Original Name_Number".
  - Use Remove Extra Dataset from Mesh to remove additional datasets groups from the mesh layer. Note that only dataset groups not associated with the mesh source file can be removed.
  - Collapse all and Expand all the dataset tree, in case of embedded groups
  - 관심을 가진 데이터셋이 몇 개 되지 않는다면 다른 데이터셋들을 체크 해제해서 프로젝트에서 사용할 수 없게 만들 수도 있습니다.
  - 데이터셋 이름을 더블클릭하면 데이터셋을 재명명할 수 있습니다.

  - Reset to defaults: checks all the groups and renames them back to their original name in the provider.
  - 가상 데이터셋 그룹을 오른쪽 클릭하면:
    * *Remove dataset group*: 프로젝트에서 해당 데이터셋 그룹을 제거할 수 있습니다.
    * *Save dataset group as*…: 해당 데이터셋 그룹을 디스크 상에 어떤 지원 포맷으로든 파일로 저장할 수 있습니다. 이 새 파일은 프로젝트에 있는 현재 메시 레이어에 할당된 채로 유지됩니다.
- *Treat as static dataset* 그룹을 체크하면 메시 레이어를 렌더링하는 동안맵 시계열 탐색 속성을 무시할 수 있습니다. ( *Symbology* *Datasets* 탭에서 선택한) 각 활성 데이터셋 그룹에 대해:
  - *None* 으로 설정하면 해당 데이터셋 그룹을 하나도 표시하지 않습니다.
  - 예를 들어 시계열 데이터가 아닌"지층 표고 (bed elevation)"의 경우, *Display dataset* 으로 설정할 수 있습니다.
  - extract a particular date time: the dataset matching the provided time is rendered and stays fixed during map navigation.

### 14.3.3 심볼 속성

*Symbology* 버튼을 클릭해서 심볼 대화창을 활성화시키십시오. 심볼 속성은 몇 개의 탭으로 나뉘어져 있습니다:

- 데이터셋
- 등고선
- 벡터
- 렌더링
- 메시 스택 평균화 메소드

#### 데이터셋

Datasets 탭에서 레이어에 어떤 데이터셋을 사용할지 제어하고 설정합니다. 다음 항목들을 설정할 수 있습니다:

- 메시 데이터셋에서 사용할 수 있는 *Groups* 가 다음 가운데 어느 유형을 제공하는지:
  - 스칼라 (scalar) 데이터셋
  - 또는 벡터 데이터셋: 기본적으로 각 벡터 데이터셋은 그 규모 (magnitude) 를 표현하는 스칼라 데이터셋 하나를 자동적으로 생성합니다.

  데이터셋 이름 옆에 있는 아이콘을 클릭하면 표현할 그룹과 데이터 유형을 선택할 수 있습니다.
- *Selected dataset group(s) metadata* 에서 다음과 같은 상세 정보를 볼 수 있습니다:
  - 메시 유형: 변 (edge) 또는 면 (face)
  - 데이터 유형: 꼭짓점, 변, 면 또는 부피
  - 데이터셋 그룹이 벡터 유형인지 아닌지
  - 메시 레이어에 있는 원래 이름
  - 적용 가능한 경우, 단위

- 선택한 데이터셋에 사용할 수 있는혼합 모드

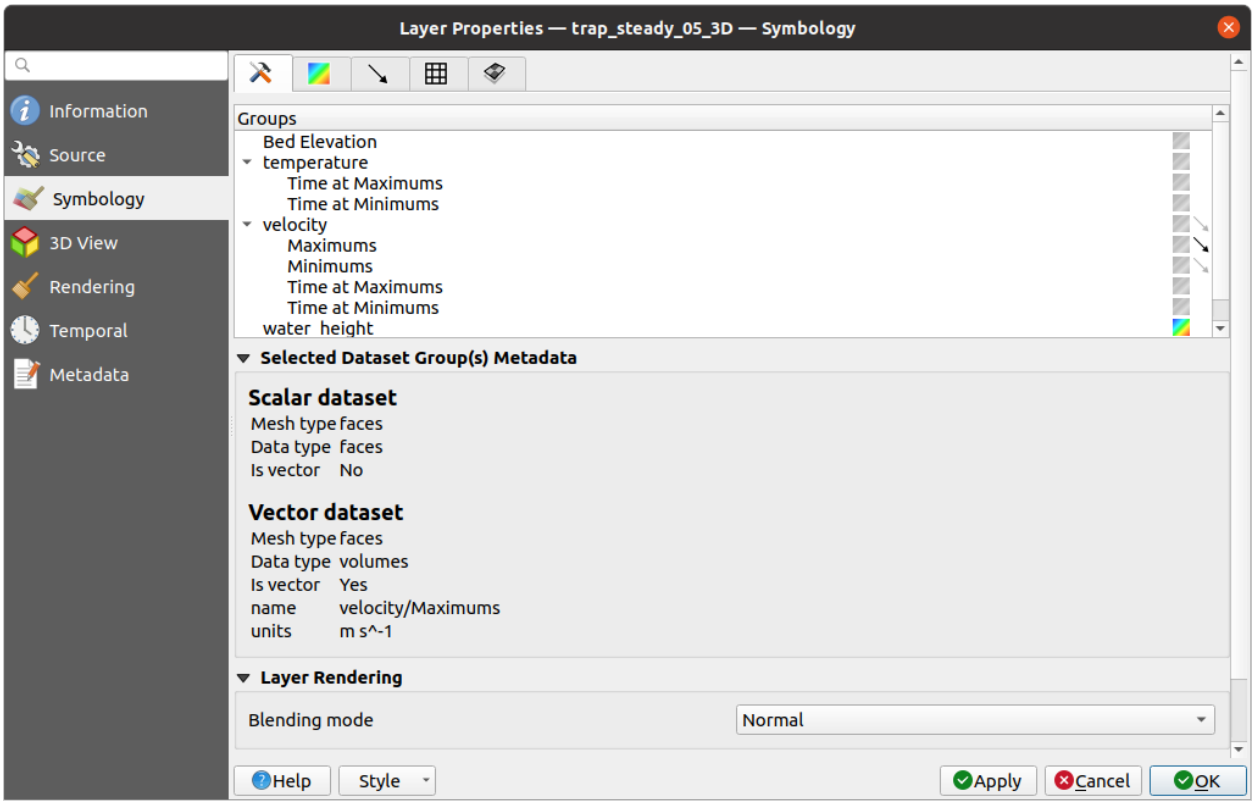



그림 14.6: 메시 레이어 데이터셋

선택한 벡터 그리고/또는 스칼라 그룹에 다음 탭들을 이용해서 심볼을 적용할 수 있습니다.

### 등고선 심볼

**참고**

Contours 탭은 *Datasets* 탭에서 스칼라 데이터셋을 선택한 경우에만 활성화됩니다.

Contours 탭에서는 다음 그림 14.7 에서 볼 수 있듯이 선택한 그룹의 현재 등고선 시각화 옵션을 확인하고 변경할 수 있습니다:

- 1 차원 메시의 경우, 변의 *Stroke width* 를 설정하십시오. 전체 데이터셋에 대해 고정 너비를 설정할 수도 있고, 아니면 도형에 따라 다른 너비를 설정할 수도 있습니다. (더 자세한 내용은보간된 라인 렌더링 작업자 를 참조하세요.)
- 2 차원 메시 유형의 경우, 슬라이드 바 또는 스핀박스를 사용해서 현재 그룹의 *Opacity* 를 설정할 수 있습니다.
- *User Defined* allows you to enter the range of values you want to represent on the current group: use Load to fetch the min and max values of the current group or enter custom values if you want to exclude some.
- Select *Min/Max* to set the renderer's minimum and maximum values based on the chosen extent.

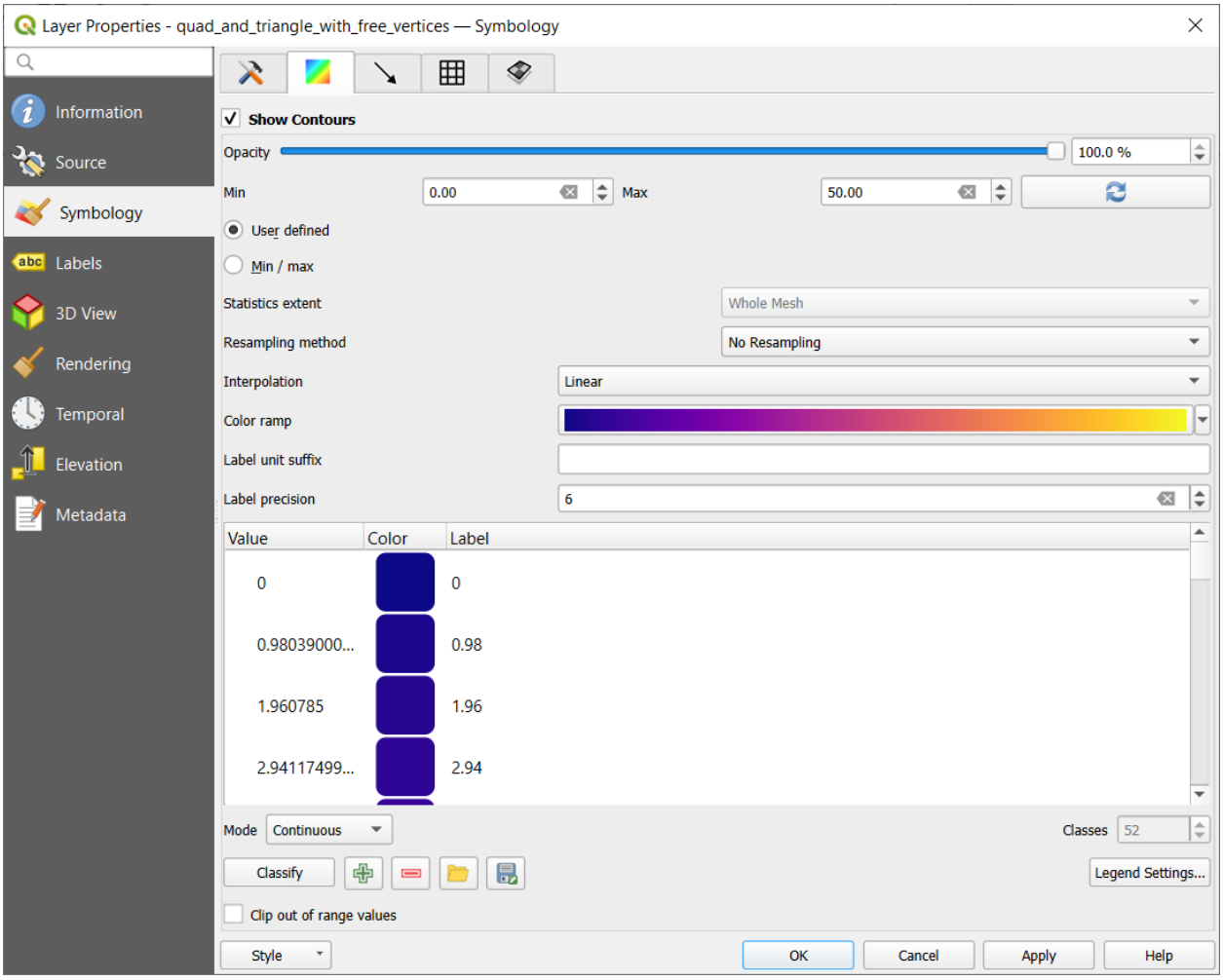



그림 14.7: 메시 레이어에서 등고선 스타일 작업하기

- *Statictics extent* can be *Whole mesh*, *Current canvas* or *Updated canvas*. *Updated canvas* means that min/max values used for the rendering will change with the canvas extent (dynamic stretching).
- 2 차원/3 차원 메시의 경우, *Neighbour average* 메소드를 이용해서 면들을 둘러싼 꼭짓점들에 대해 (또는 꼭짓점들을 둘러싼 면들로부터) 값을 보간하려면 *Resampling method* 를 선택하십시오. 데이터셋이 (면 각각에 대해) 꼭짓점들에 대해 정의되어 있는지에 따라, QGIS 는 꼭짓점들에 대한 값을 이용해서 기본 렌더링을 매끄럽게 유지하기 위해 (각각 *Neighbour average* 인) *None* 메소드를 이 설정의 기본값으로 사용합니다.
- 색상표 셰이더 *(color ramp shader)* 범주화를 이용해서 데이터셋을 범주화시키십시오.

### 벡터 심볼

**참고**

The Vectors tab can be activated only if a vector dataset has been selected in the *Datasets* tab. Click on the Vectors icon on the right side of the *Datasets* tab.

Vectors 탭에서는 다음 그림 14.8 에서 볼 수 있듯이 선택한 그룹의 현재 벡터 시각화 옵션을 확인하고 변경할 수 있습니다:

다양한 유형의 *Symbology* 를 이용해서 메시 벡터 데이터셋의 스타일을 작업할 수 있습니다:

- **화살표**: 벡터가 원시 데이터셋에서 (예를 들어 노드에서 또는 요소의 중심에서) 또는 사용자 정의 그리드에서 (그러니까 균등하게 분포되어) 정의된 대로, 동일한 위치에서 벡터를 화살표로 표현할 수 있습니다. 화살표의 길이는 원시 데이터에 정의된 벡터 규모에 비례하지만, 다양한 메소드로 크기를 조정할 수 있습니다.
- **유선 (□□; streamline)**: 벡터를 시작점 (start point) 에서 시작하는 유선으로 표현합니다. 이 시작점은 메시의 꼭짓점, 사용자 그리드, 또는 임의의 위치에서 시작할 수 있습니다.
- **추적 (trace)**: 유선보다 더 보기 좋은 애니메이션입니다. 물 속에 모래알을 임의로 던진 다음 해당 모래가 어디로 떠내려 가는지를 확인하는 것과 비슷한 효과를 보여줍니다.
- **Wind barbs**: vectors are represented with wind barbs, a common way to represent wind speed and direction.

선택한 심볼에 따라 사용할 수 있는 속성은 다음 표와 같습니다.

표 14.3: 벡터 심볼 속성의 사용 가능성 및 그 의미

| Label, Description and Properties | 화살표 | 유선 | 추적 | Wind Barbs |
|---|---|---|---|---|
| *Line width*: Width of the vector representation | ☑ | ☑ | ☑ | ☑ |
| *Coloring method*:<br>• *Single color*: 모든 벡터에 할당되는 단일 색상<br>• 또는색상표 셰이더 를 이용하는, 벡터 규모를 기반으로 하는 다양한 색상 | ☑ | ☑ | ☑ | ☑ |
| *Filter by magnitude*: Only vectors whose length for the selected dataset falls between a *Min* and *Max* range are displayed | ☑ | ☑ | | ☑ |
| *Display on user grid*: Places the vector on a grid with custom *X spacing* and *Y spacing* and interpolates their length based on neighbours | ☑ | ☑ | | ☑ |

다음 페이지에 계속

표 14.3 – 이전 페이지에서 계속

| Label, Description and Properties | 화살표 | 유선 | 추적 | Wind Barbs |
|---|---|---|---|---|
| *Head options*: *Length* and *Width* of the arrow head, as a percentage of its shaft length | ☑ | | | |
| *Arrow length*:<br>• **최소/최대값으로 정의**: 사용자가 화살표에 대한 최단/최장 길이를 지정하면, QGIS 가 기저 벡터의 규모를 기반으로 그 크기를 보간할 것입니다.<br>• **규모에 비례**: 화살표 길이가 기저 벡터의 규모에 비례합니다.<br>• **고정**: 모든 벡터를 동일한 길이로 표시합니다. | ☑ | | | |
| *Streamlines seeding method*:<br>• **메시/그리드 상**: 벡터를 표시하기 위해 사용자 그리드에 의존합니다.<br>• **임의**: 벡터를 특정 밀도에 따라 임의로 배치합니다. | | ☑ | | |
| *Particles count*: The amount of "sand"you want to throw into visualisation | | | ☑ | |
| *Max tail length*: The time until the particle fades out | | | ☑ | |
| *Wind Barbs*:<br>• **Length**: Specifies the length of the wind barbs. Choose the units from a drop-down menu.<br>• **Data Units**: Specifies the units of wind speed data. Wind barbs encode the wind in knots (nautical miles per hour). Therefore, you should select the units that your data are in, or use a custom *Multiplier* to convert your data units to knots. This ensures that the wind speed is accurately represented. | | | | ☑ |

### 렌더링

QGIS 는 Rendering 탭에서 메시 구조를 표시하고 사용자 지정할 수 있는 다양한 옵션을 제공합니다. 다음을 표현하기 위해 *Line width* 및 *Line color* 를 설정할 수 있습니다:

- 1 차원 메시의 경우, 변 (edge)
- 2 차원 메시의 경우:
  - *Native mesh rendering*: 레이어로부터 원본 면과 변을 표시
  - *Triangular mesh rendering*: 더 많은 변을 추가해서 면을 삼각형으로 표시

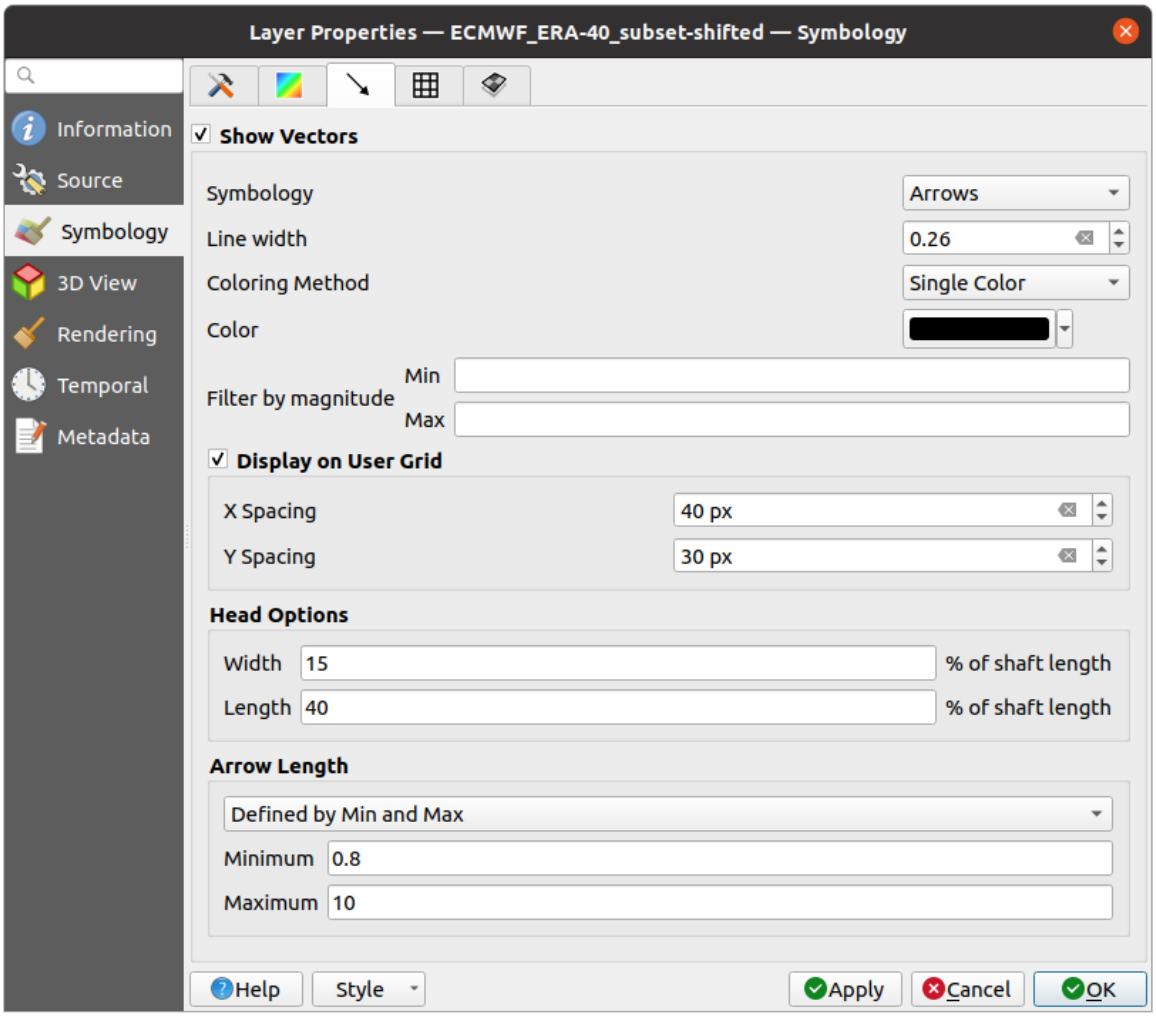



그림 14.8: 메시 레이어에서 화살표로 벡터 스타일 작업하기

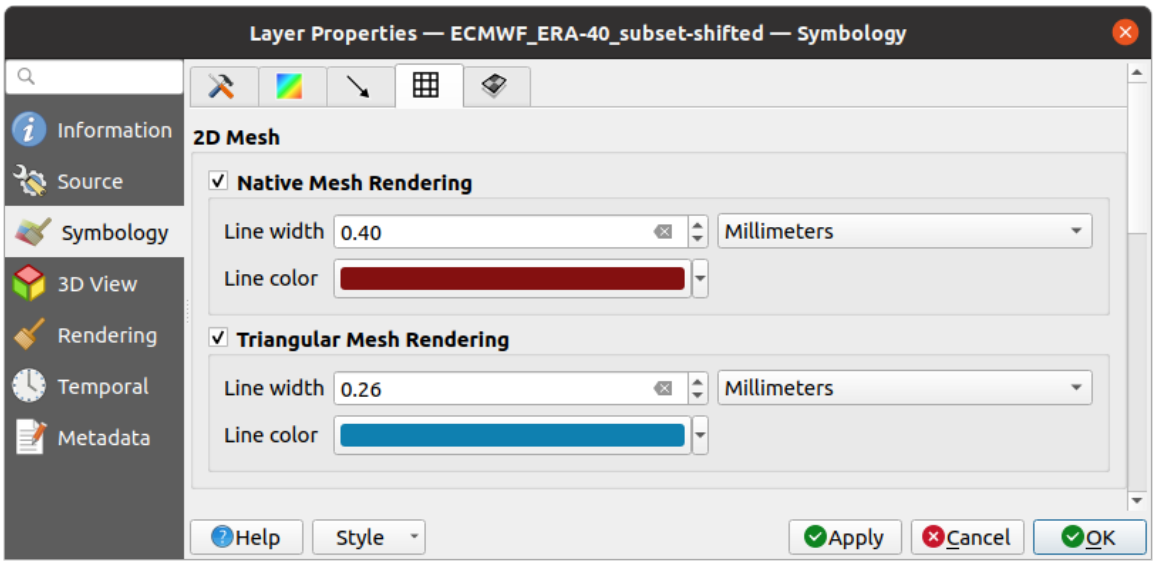



그림 14.9: 2 차원 메시 렌더링

#### 메시 스택 평균화 메소드

3 차원 적층 (□□) 메시는 수직 좌표를 이용해서 각각 수직 방향 (`levels`) 으로 압출 (extrude) 된 2 차원 비구조적 (unstructured) 메시 여러 개를 쌓아 올린 형태로 이루어져 있습니다. 꼭짓점 및 면의 위상은 각 꼭짓점 수준의 위상과 동일합니다. 일반적으로 기반 2 차원 메시 위에 규칙적으로 쌓아 올린 부피에 값을 저장합니다. 2 차원 캔버스에 3 차원 적층 메시를 가시화시키려면, (3 차원) 부피의 값을 메시 레이어에 표시할 수 있는 (2 차원) 면의 값으로 변환해야 합니다. Stacked mesh averaging method 가 이런 변환을 처리할 수 있는 다양한 평균화/보간 메소드를 제공합니다.

You can select the method to derive the 2D datasets and corresponding parameters (level index, depth or height values). For each method, an example of application is shown in the dialog but you can read more on the methods at https://fvwiki.tuflow.com/Depth_Averaging_Results.

### 14.3.4 라벨 속성

The *Labels* tab offers you dynamic labeling of mesh vertices and native mesh faces based on geometric properties and custom expressions. This dialog can also be accessed from the *Layer Styling* panel.

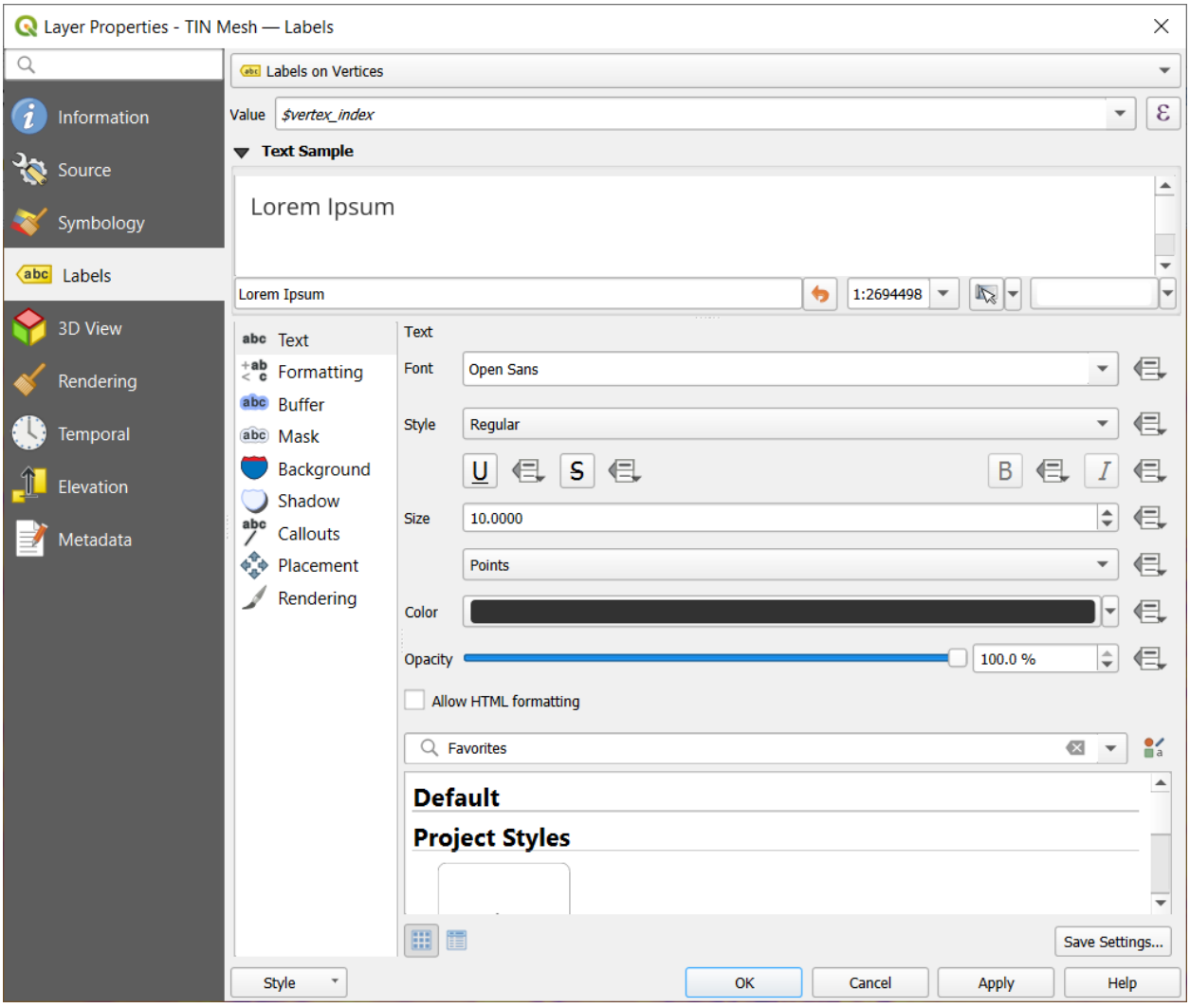



그림 14.10: Mesh Labels Properties

Choose the labeling method from the drop-down list. Available methods are:

- *No labels*: the default value, showing no labels from the layer.
- *Labels on vertices*: labels the mesh vertices as a point feature.
- *Labels on faces*: labels the mesh faces as a polygon feature.

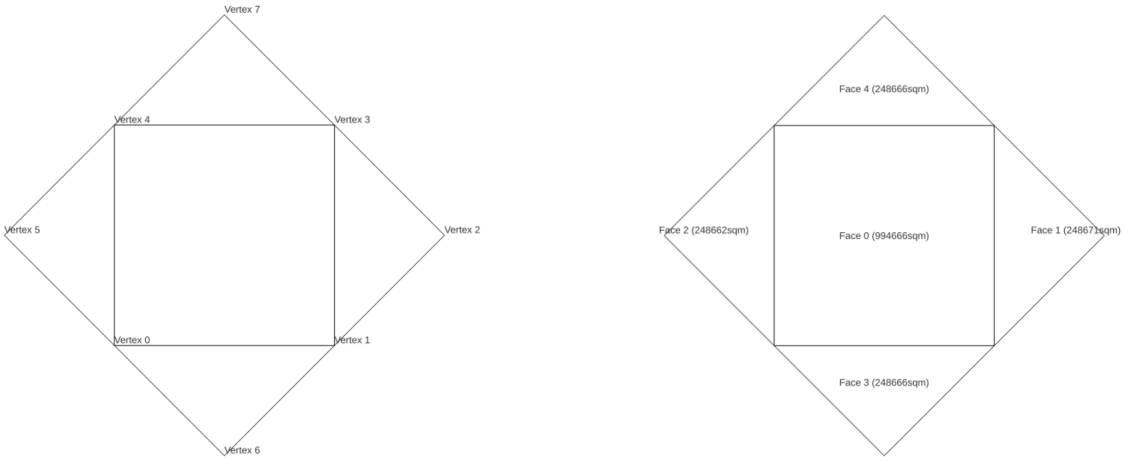



그림 14.11: Mesh Labeling methods

Once the labeling method has been selected, open Expression dialog and choose a *mesh function*. You can combine meshes functions with other functions from the *list* to modify the label.

여러 탭에 라벨을 사용자 지정할 수 있는 다음 옵션들이 표시됩니다:

- 텍스트
- 서식 작업
- 버퍼
- 마스크
- 배경
- 그림자
- 호출
- 배치
- 렌더링

라벨 설정 에서 각 속성을 설정하는 방법을 설명하고 있습니다.

### 14.3.5 3 차원 뷰 속성

메시 레이어의 꼭짓점 Z 값을 바탕으로 메시 레이어를*3 차원 맵 뷰의 지형* 으로 사용할 수 있습니다. *3D View* 속성 탭에서 동일한 3 차원 뷰에 메시 레이어의 데이터셋을 렌더링할 수도 있습니다. 따라서 꼭짓점의 수직 요소를 (수면의 수준 같은) 데이터셋 값과 동등하게 설정할 수 있고, 메시의 텍스처를 색상 음영을 이용해서 (예를 들어 속도 같은) 다른 데이터셋 값으로 렌더링하도록 설정할 수도 있습니다.

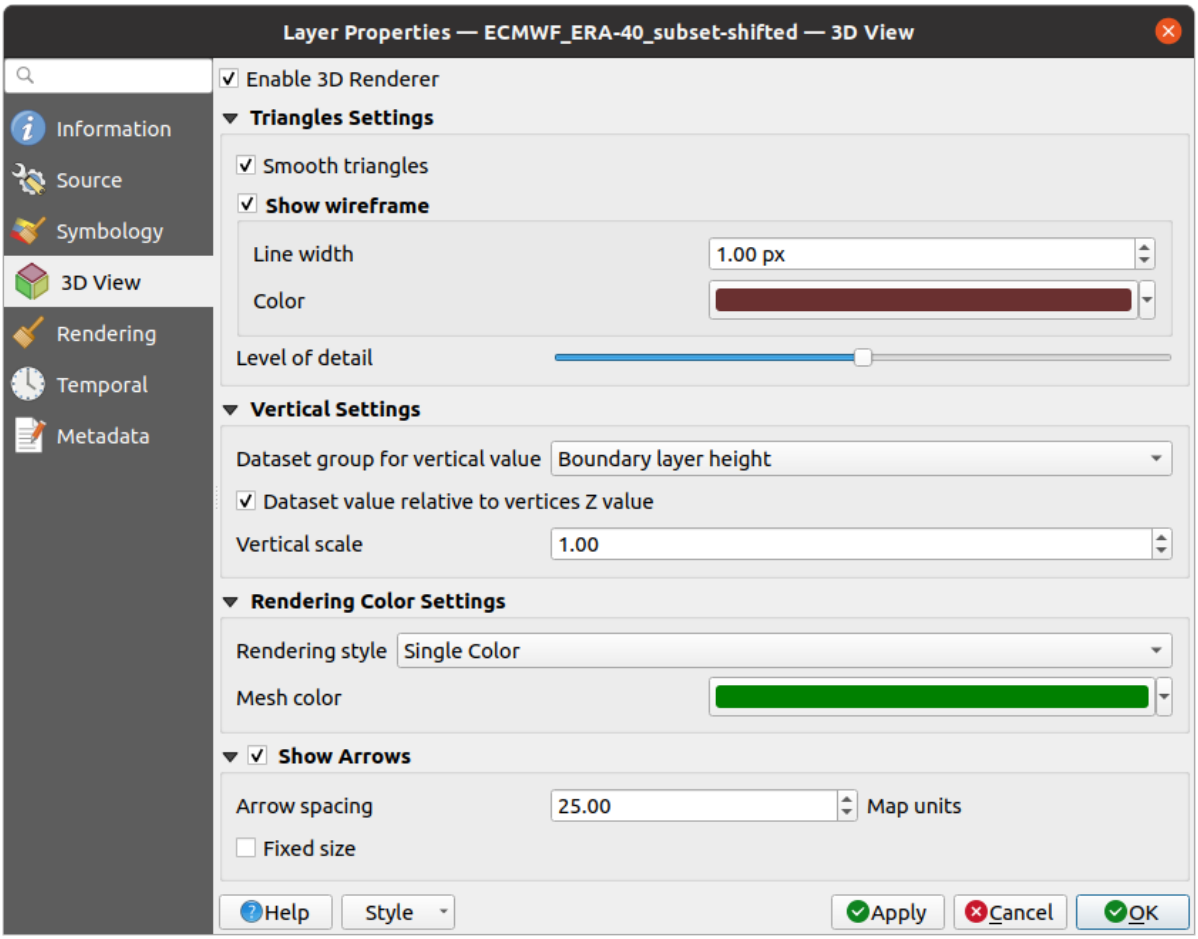



그림 14.12: 메시 데이터셋 3 차원 속성

*Enable 3D Renderer* 를 체크하면 다음과 같은 속성을 편집할 수 있습니다:

- *Triangle settings* 에서는:
  - *Smooth triangles*: 연속되는 삼각형들 사이의 각도를 부드럽게 처리해서 3 차원 렌더링을 개선합니다.
  - *Show wireframe*: 철사 구조 (wire frame) 의 *Line width* 및 *Color* 를 설정할 수 있습니다.
  - *Level of detail*: 렌더링할 메시 레이어를 어떻게단순화 시킬지를 제어합니다. 맨 오른쪽이 기반 메시이며, 왼쪽으로 갈수록 레이어를 더 단순화시켜 세부 정보를 줄여서 렌더링합니다. *Rendering* 탭의 *Simplify mesh* 옵션을 활성화시킨 경우에만 이 옵션을 사용할 수 있습니다.
- *Vertical settings* 에서는 렌더링된 삼각형 꼭짓점의 수직 요소의 습성을 제어할 수 있습니다:
  - *Dataset group for vertical value*: 메시의 수직 요소로 쓰이게 될 데이터셋 그룹을 지정합니다.

  - *Dataset value relative to vertices Z value*: 데이터셋 값을 절대 Z 좌표로 간주할지 아니면 꼭짓점의 자체 Z 값에 상대적인 값으로 간주할지 여부를 선택합니다.
  - *Vertical scale*: 데이터셋의 Z 값에 적용할 척도를 지정합니다.

- *Rendering color settings*: 등고선 심볼 (*2D contour color ramp shader*) 에 설정된 색상표 셰이더를 기반으로 할 수도 있는 렌더링 스타일 (*Rendering style*) 을 이용해서 렌더링할지, 또는 관련 *Mesh color* 인 단일 색상 (*Single color*) 으로 렌더링할지를 설정합니다.
- *Show arrows*: 벡터 2 차원 렌더링 에 쓰인 것과 동일한 벡터 데이터셋 그룹을 바탕으로 메시 레이어 데이터셋의 3 차원 개체에 화살표를 표시합니다. 화살표의 색상은 2 차원 색상 설정으로 표시됩니다. 화살표 사이의 간격 (*Arrow spacing*) 을 정의할 수도 있고, 화살표의 크기가 고정값 (*Fixed size*) 인지 아니면 축척에 따라 조정되는지도 정의할 수 있습니다. 화살표가 서로 중첩할 수는 없기 때문에, 이 간격 설정을 통해 화살표의 최대 크기도 정의합니다.

### 14.3.6 렌더링 속성

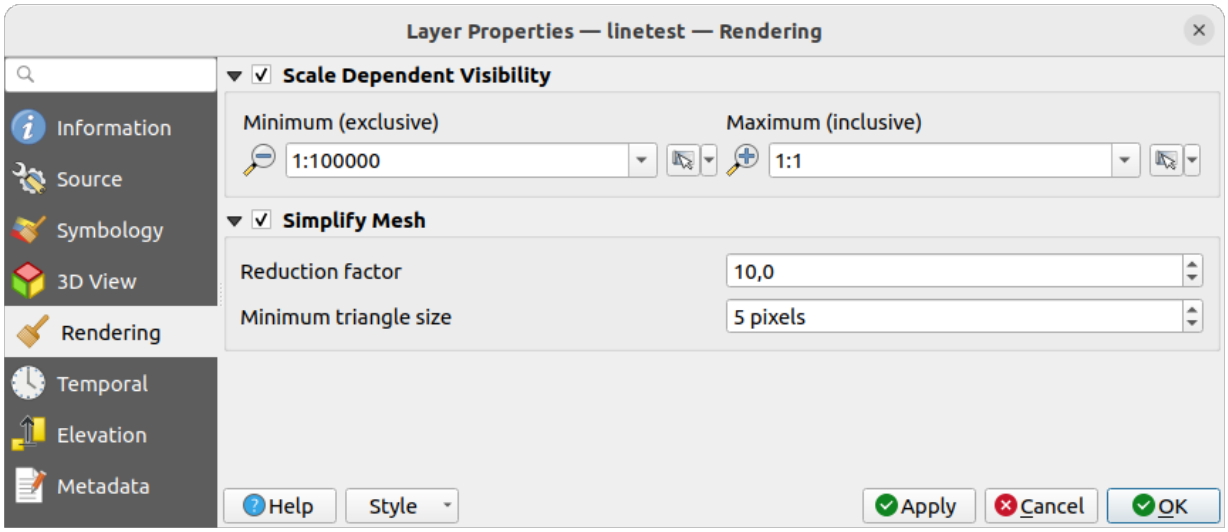



그림 14.13: 메시 렌더링 속성

*Scale dependent visibility* 그룹 박스에서는 *Maximum (inclusive)* 및 *Minimum (exclusive)* 축척을 설정해서 메시 요소가 보이게 될 축척 범위를 정의할 수 있습니다. 이 범위를 벗어나면, 메시 요소를 숨깁니다. Set to current canvas scale 버튼을 클릭하면 현재 맵 캔버스의 축척을 가시성 범위의 한계값으로 설정할 수 있습니다. 자세한 내용은가시성 축척 선택기 을 참조하세요.

> **참고**
>
> *Layers* 패널에서 레이어에 대한 축척 의존 가시성을 활성화시킬 수도 있습니다. 레이어를 오른쪽 클릭한 다음 컨텍스트 메뉴에서 *Set Layer Scale Visibility* 를 선택하십시오.

메시 레이어의 면이 수백만에 달할 수도 있기 때문에, 특히 모든 면을 보이기엔 너무 작은 뷰에 모든 면을 표시하는 경우 렌더링이 아주 느려지는 경우가 있을 수도 있습니다. 렌더링의 속도를 높이려면 메시 레이어를 단순화시켜 하나 이상의 메시가 서로 다른상세 수준 *(level of detail)* 을 표현하도록 하고 QGIS 가 메시 레이어를 렌더링하길 바라는 상세 수준을 선택하면 됩니다. 단순화된 메시는 삼각면만 담고 있다는 사실을 기억하십시오.

*Rendering* 탭에서 *Simplify mesh* 옵션을 체크한 다음 다음을 설정하십시오:

- *Reduction factor*: 단순화된 메시의 연속적인 수준의 생성을 제어합니다. 예를 들면 기반 메시의 면이 5 백만 개이고 축도 (□□) 인자가 10 인 경우, 처음으로 단순화된 메시의 면은 약 50 만 개, 두 번째로 단순화된 메시의 면은 5 만 개, 세 번째는 5 천 개, …일 것입니다. 축도 인자의 값이 높을수록 메시를 더 빠르게 단순화하지만 (즉 삼각면의 크기가 커지지만) 상세 수준의 개수는 적어집니다.
- *Minimum triangle size*: 표시가 허용되는 삼각형의 (픽셀 단위) 평균 크기를 설정합니다. 메시의 평균 크기가 이 값보다 적을 경우, 더 낮은 상세 수준의 메시를 렌더링할 것입니다.

### 14.3.7 시계열 속성

*Temporal* 탭은 레이어의 렌더링을 시간에 따라 제어할 수 있는 옵션들을 제공합니다. 이 옵션들은 활성화된 데이터셋 그룹의 시계열 값을 동적으로 표시할 수 있게 해줍니다. 이런 동적 렌더링에는 맵 캔버스 전체에 적용되는시계열 탐색 을 활성화해야 할 필요가 있습니다.

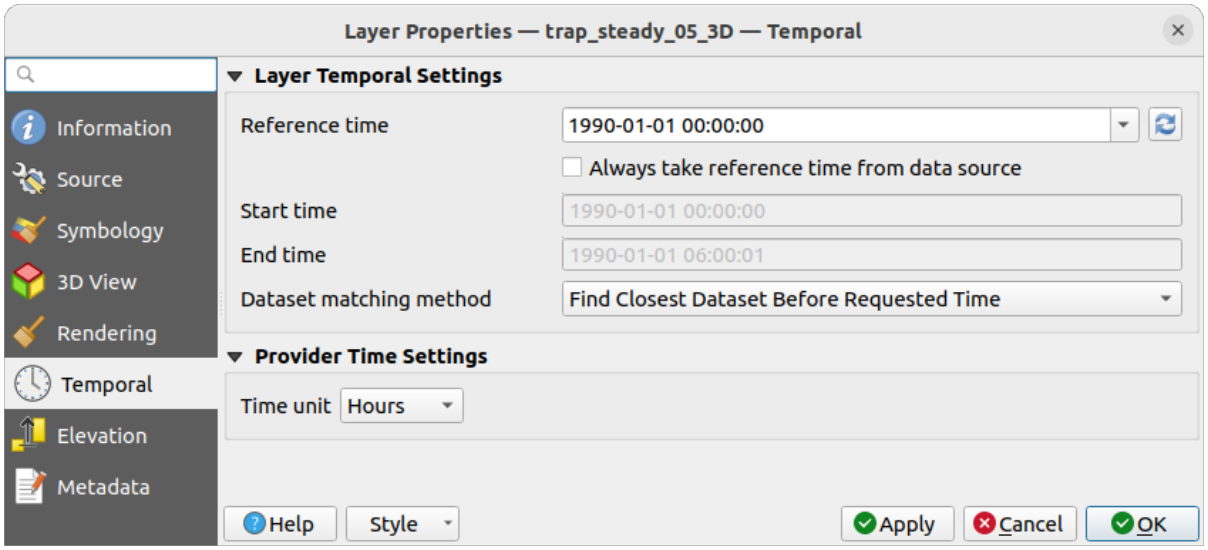



그림 14.14: 메시 시계열 속성

#### 레이어 시계열 설정

- *Reference time*: 데이터셋 그룹의 참조 (기준) 시간을 절대 날짜 & 시간 값으로 설정합니다. QGIS 는 기본적으로 소스 레이어를 파싱해서 레이어의 데이터셋 그룹에 있는 무결한 첫 번째 참조 시간을 반환합니다. 레이어에 참조 시간이 없다면 프로젝트의 시간 범위가 설정한 값을 사용하거나, 또는 현재 날짜로 되돌아갑니다. 그 다음 데이터셋의 내부 타임스탬프 단계를 기반으로 연산에 넣을 *Start time* 과 *End time* 을 계산합니다.

  *Reference time*: 참조 시간을 (그 다음 시간 범위를) 사용자 지정할 수 있으며, Reload from provider 버튼을 사용해서 변경 사항을 되돌릴 수도 있습니다. *Always take reference time from data source* 옵션을 체크하면, 레이어를 다시 불러올 때마다 또는 프로젝트를 다시 열 때마다 파일로부터 시간 속성을 업데이트시킬 수 있습니다.

- *Dataset matching method*: 지정한 시간에 표시할 데이터셋을 결정합니다. *Find closest dataset before requested time* 옵션은 요청 시간 이전에 가장 가까운 데이터셋을 검색하며, *Find closest dataset from requested time (after or before)* 옵션은 요청 시간 전후에 가장 가까운 데이터셋을 검색합니다.

#### 제공자 시간 설정

- 원시 데이터에서 *Time unit* 을 추출하거나, 또는 사용자 정의합니다. 시간 단위를 정의하면 맵 시간 탐색 도중 메시 레이어의 속도를 프로젝트의 다른 레이어들과 정렬시킬 수 있습니다. *Seconds*, *Minutes*, *Hours* 및 *Days* 단위를 지원합니다.

### 14.3.8 표고 속성

The *Elevation* tab provides options to control the layer elevation properties within a *3D map view* and *2D map view* and its appearance in the *profile tool charts*. Specifically, you can configure how heights from your dataset are interpreted:

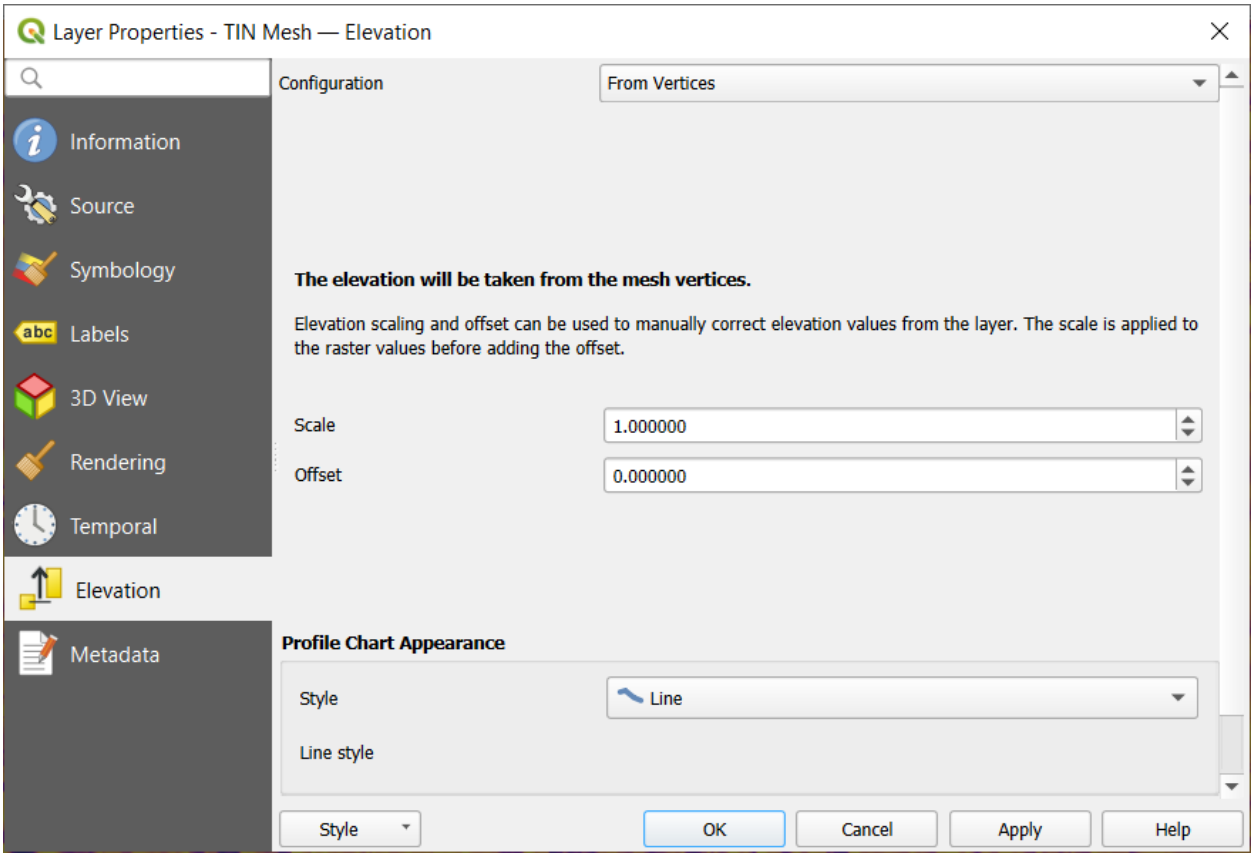



그림 14.15: 메시 표고 속성

- *From vertices*: how the mesh layer vertices Z values should be interpreted as terrain elevation. You can apply a *Scale* factor and an *Offset*. This setting is available for *3D map view* and *profile tool charts*.
- *Fixed Elevation Range*: the mesh layer is linked to a fixed elevation range. This mode is applicable when a layer has either a single fixed elevation or a range (slice) of elevation values. If a range is specified, mesh values will be extruded over this range. You can set the *Lower* and *Upper* elevation range values for the layer, and specify whether the lower or upper *Limits* are inclusive or exclusive. When enabled, the layer will only be visible in *elevation filtered 2D maps* when the layer's range is included in the map's Z range.
- *Fixed Elevation Range Per Group*: each group in the mesh layer is associated with a fixed elevation range. This mode can be used when a layer has elevation data exposed through different dataset groups. This feature is exposed as a user-editable table for dataset groups with lower and upper values. You can either populate the lower and upper values manually or use an *Expression* to auto-fill all group values based on an expression. When enabled, the layer will be filtered in the *2D map view*, displaying only values within the map filter ranges. In *3D map view* or *profile tool charts*, the full extent of the layer will be displayed, ignoring the filtering.

- *Profile Chart Appearance*: 단면 도표에 있는 메시 요소 표고의 렌더링을 제어합니다. 단면 *Style* 을 다음 가운데 하나로 설정할 수 있습니다:
  - 라인 스타일 이 적용된 *Line*
  - 표고 곡선 위 (*Fill above*) 또는 아래 (*Fill below*) 에 채우기 심볼을 사용해서 렌더링한 표고 표면입니다. 표면 심볼은 다음을 이용해서 표현됩니다:
    * 채우기 스타일
    * *Limit*: 채우기 표면이 얼마나 높이 올라가게 될지 결정하는 최대 (제각기 최저) 고도를 설정합니다.

### 14.3.9 메타데이터 속성

*Metadata* 탭에 사용자 레이어에 대한 메타데이터 리포트를 생성하고 편집할 수 있는 옵션이 있습니다. 더 자세한 정보는메타데이터 를 참조하세요.

## 14.4 Exploring mesh datasets

To identify mesh elements and their associated datasets, use the *Identify Features* tool from the *Attributes* toolbar. Click on a mesh element in the map canvas and the *Identify Results* panel opens, displaying information about the mesh layer and the identified element. The panel shows:

- Layer name
- Active dataset groups (scalar and/or vector) for the clicked element will be displayed first. On the other hand, inactive dataset groups are shown afterwards. For each group, the following derived information may also be shown, if applicable:
  - **Source** of the dataset (the file where it is stored).
  - **Time step** of the dataset value being displayed. For non-temporal dataset groups, this field remains empty.
  - **Vector x-component** and **Vector y-component** values for vector datasets.

Please note that when temporal navigation is disabled, the results only contain values from the *static dataset groups* defined.

- Geometry information shows the coordinates of the clicked point and may contain derived information in the following:
  - Centroid of the face.
  - Coordinates of a snapped vertex (if the click was snapped to a vertex).
  - Center of a snapped edge (if the click was snapped to an edge).

## 14.5 메시 레이어 편집

QGIS 는 처음부터 또는 기존 메시 레이어를 기반으로메시 레이어를 생성 할 수 있습니다. 새 레이어의 도형을 생성 또는 수정할 수 있으며, 나중에 해당 도형에 데이터셋을 할당할 수 있습니다. 기존 메시 레이어를 편집할 수도 있습니다. 편집 작업 대상 레이어는 도형만 가지고 있어야 하기 때문에, 편집 전에 모든 관련 데이터셋을 제거하거나 (편집 후 해당 데이터셋이 필요한 경우 데이터셋을 다시 할당할 수 있도록 준비해놓아야 합니다) 또는 레이어의 (도형만 가지고 있는) 복사본을 생성하도록 요청받게 될 것입니다.

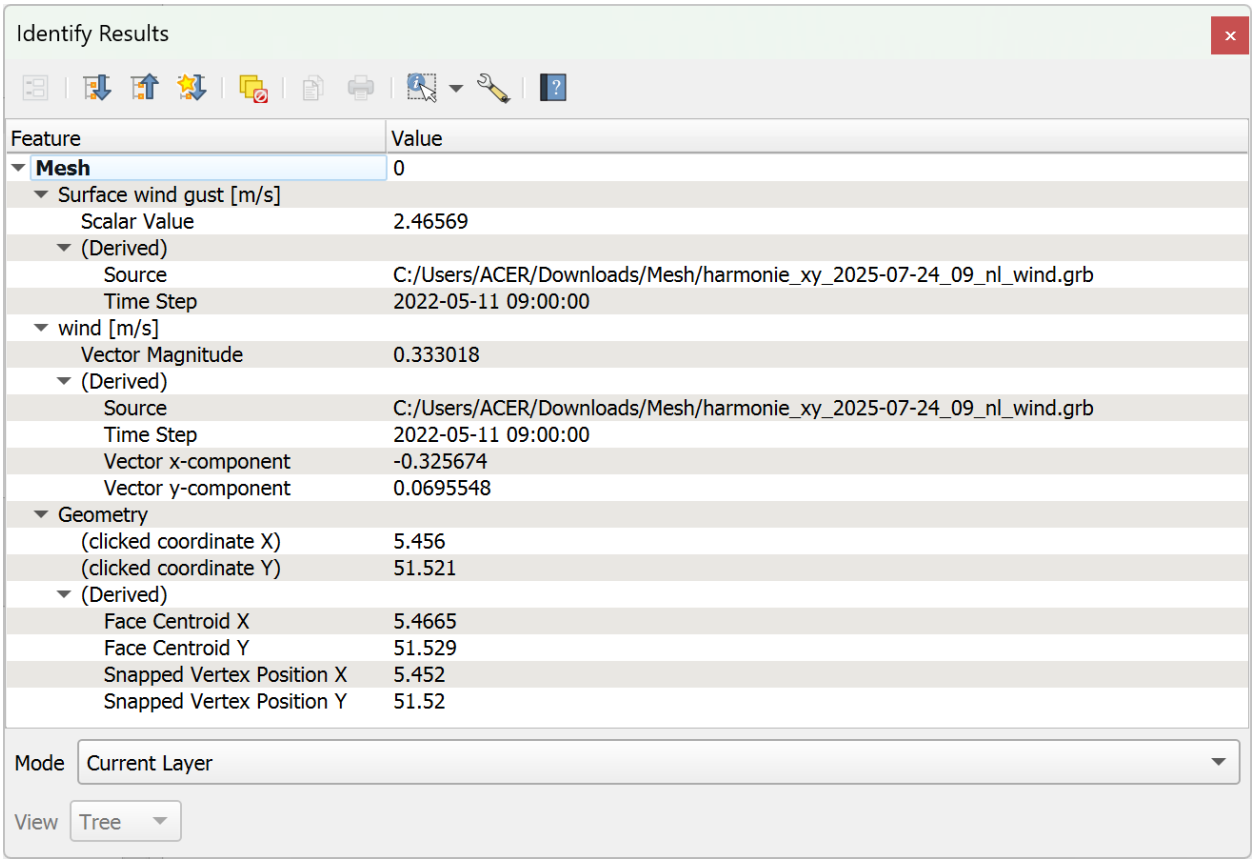



그림 14.16: Mesh Identify Results panel

> **참고**
>
> QGIS does not allow to digitize edges on mesh layers. Only vertices and faces are mesh elements that can be created. Also not all supported mesh formats can be edited in QGIS (see permissions).

### 14.5.1 메시 디지타이즈 작업 도구의 개요

다음 도구들을 이용해서 기반 메시 레이어 요소와 대화형 작업을 하거나 편집할 수 있습니다.

표 14.4: 메시 디지타이즈 작업을 위한 도구들

| Label | 목적 | 위치 |
|---|---|---|
|  Current Edits | 모든 레이어 또는 선택한 레이어에서 변경 사항을 동시에 저장, 되돌리기, 또는 취소 | *Digitizing* 툴바 |
|  Toggle to Edit | 레이어의 편집 모드를 켜고 끄기 | *Digitizing* 툴바 |
|  Save Edits | 레이어의 변경 사항을 저장 | *Digitizing* 툴바 |
|  Undo | 마지막 변경 사항 (들) 을 되돌리기 - `Ctrl+Z` | *Digitizing* 툴바 |
|  Redo | 마지막으로 되돌린 액션 (들) 을 다시 실행 - `Ctrl+Shift+Z` | *Digitizing* 툴바 |
|  Enable Advanced Digitizing tools | 고급 디지타이즈 작업 패널 을 켜고 끄기 | *Advanced Digitizing* 툴바 |
|  Reindex Faces and Vertices | 최적화를 위해 메시 요소의 개수를 다시 세고 인덱스를 재생성 | *Mesh* 메뉴 |
|  Digitize Mesh Elements | 꼭짓점 및 면을 선택/생성 | *Mesh* *Digitizing* 툴바 |
|  Select Mesh Elements by Polygon | 폴리곤을 그려서 그 안에 들어오는 꼭짓점 및 면을 선택 | *Mesh* *Digitizing* 툴바 |
|  Select Mesh Elements by Expressio | 표현식을 이용해서 꼭짓점 및 면을 선택 | *Mesh* *Digitizing* 툴바 |
|  Transform Vertices Coordinates | 선택한 꼭짓점의 좌표를 수정 | *Mesh* *Digitizing* 툴바 |
|  Force by Selected Geometries | 라인 도형을 이용해서 면을 분할하고 Z 값을 강제 | *Mesh* *Digitizing* 툴바 |

### 14.5.2 Z 값 할당 로직 탐구하기

메시 레이어의 편집 모드를 켠 경우, 맵 캔버스의 우상단에 *Vertex Z value* 위젯이 열립니다. 이 꼭짓점 Z 값은 기본적으로 *Settings* → *Options* → *Digitizing* 탭에 설정된 *Default Z value* 에 대응합니다. 꼭짓점 집합을 선택한 경우 이 위젯은 선택한 꼭짓점들의 평균 Z 값을 표시합니다.

편집 작업 도중 새 꼭짓점에 *Vertex Z value* 를 할당합니다. Z 값을 사용자 지정할 수도 있습니다. 위젯에서 값을 편집한 다음 `Enter` 키를 누르면 기본값을 무시하고 디지타이즈 작업 과정에 이 새로운 값을 사용할 것입니다. 위젯에 있는  아이콘을 클릭하면 Z 값을 기본값으로 리셋합니다.

### 할당 규칙

This is configured in the drop-down menu of the *Digitize mesh elements* tool as *New vertex Z value* option. Following methods are available:

표 14.5: Z value assignment methods

| Method | Description |
|---|---|
| **Project terrain** | Z value is always taken or calculated from the *project reference terrain*.<br>• If it is **Flat terrain**, the value corresponds to the *Terrain height* set in the project properties.<br>• If it is a **Mesh** layer, the value is interpolated from the surrounding vertices of the mesh.<br>• If it is a **DEM (raster layer)**, the value is assigned from the raster pixel value.<br>• If there is no terrain under the new vertex, the value falls back to the *Default Z value* set in *digitizing options*. |
| **Z widget** | Z value is taken from the *Vertex Z value* widget or the *Advanced Digitizing Panel Z* widget (if it is in Locked state). |
| **Prefer mesh, then Z widget** | Interpolates the Z value from the active mesh layer if the vertex is on an edge or face, otherwise falls back to the *Vertex Z value* or *Advanced Digitizing Panel Z* widget (if it is in Locked state). |
| **Prefer mesh, then terrain** | Interpolates the Z value from the active mesh layer if the vertex is on the edge or face, otherwise uses the project reference terrain value. |

The following detailed logic describes the behavior of the **Prefer mesh, then Z widget** strategy.

표 14.6: 새 꼭짓점에 Z 값을 할당하는 경우들

| 꼭짓점 생성 | Vertices selected? | 할당되는 값의 소스 | 할당되는 Z 값 |
|---|---|---|---|
| 어떤 면 또는 면의 변에도 연결되어 있지 않은"자유로운"꼭짓점 | X | *Vertex Z value* | 기본값 또는 사용자 정의 값 |
| | | *Advanced Digitizing Panel* (*z* 위젯이 Locked 상태인 경우) | *z* 위젯이 Locked 상태인 경우 *z* 위젯의 값 |
| | O | *Vertex Z value* | Average of the selected vertices or user-defined |
| 변 위에 있는 꼭짓점 | — | 메시 레이어 | 변의 꼭짓점들로부터 보간한 값 |
| 면 위에 있는 꼭짓점 | — | 메시 레이어 | 면의 꼭짓점들로부터 보간한 값 |
| 2 차원 벡터 피처에 스냅된 꼭짓점 | — | *Vertex Z value* | 기본값 또는 사용자 정의 값 |
| 3 차원 벡터 꼭짓점에 스냅된 꼭짓점 | — | 벡터 레이어 | 꼭짓점의 Z 값 |
| 3 차원 벡터 선분에 스냅된 꼭짓점 | — | 벡터 레이어 | 벡터 선분을 따라 보간된 값 |

> **참고**
>
> 고급 디지타이즈 작업 패널 을 활성화하고 어떤 메시 요소도 선택하지 않은 경우 *Vertex Z value* 위젯이 비활성화됩니다. 이때 고급 디지타이즈 작업 패널의 *z* 위젯이 Z 값 할당 규칙을 제어합니다.

#### 기존 꼭짓점의 Z 값을 수정하기

꼭짓점의 Z 값을 수정하는 가장 단순한 방법은 다음과 같습니다:

1. 하나 이상의 꼭짓점을 선택하십시오. *Vertex Z value* 위젯이 선택 집합의 평균 높이를 표시할 것입니다.
2. 위젯의 값을 변경하십시오.
3. `Enter` 키를 누르십시오. 입력한 값이 꼭짓점에 할당되고 다음 꼭짓점의 기본값이 됩니다.

꼭짓점의 Z 값을 변경하는 또다른 방법은 Z 값 케이퍼빌리티를 가진 벡터 레이어 피처로 꼭짓점을 이동시켜 스냅하는 것입니다. 하나 이상의 꼭짓점을 선택한 경우, Z 값을 이 방법으로 변경할 수 없습니다.

메시 꼭짓점 변형 대화창에서도 꼭짓점 선택 집합의 Z 값을 (꼭짓점의 X 또는 Y 좌표와 함께) 수정할 수 있습니다.

### 14.5.3 메시 요소 선택하기

To select mesh elements, you can use the following tools:

-  Digitize Mesh Elements to select different mesh elements, see more at *Digitize Mesh Elements* 사용하기.
-  Select Mesh Elements by Polygon to select different mesh elements by polygon, see more at *Select Mesh Elements by Polygon* 사용하기.
-  Select Mesh Elements by Expression to select different mesh elements by expression, you can choose to *Select by vertices* or *Select by faces*, see more at *Select Mesh Elements by Expression* 사용하기.
-  Select Isolated Vertices to select all vertices that are not part of any mesh face.
-  Select All Vertices to select all vertices of the mesh layer.

#### *Digitize Mesh Elements* 사용하기

 Digitize Mesh Elements 도구를 활성화하십시오. 요소 위로 마우스 커서를 가져가면 요소가 강조되고, 요소를 선택할 수 있습니다.

- 꼭짓점을 클릭하면 해당 꼭짓점이 선택됩니다.
- 면 또는 변의 중심에 있는 작은 정사각형을 클릭하면 해당 면 또는 변을 선택합니다. 연결되어 있는 꼭짓점들도 선택됩니다. 반대로 변 또는 면의 모든 꼭짓점을 선택하면 해당 요소도 선택됩니다.
- 직사각형을 그려 직사각형과 중첩하는 요소를 (면이 선택되는 경우 면의 모든 꼭짓점도) 선택합니다. 직사각형 안에 완전히 들어오는 요소만 선택하고자 하는 경우 `Alt` 키를 누른 채 직사각형을 그리십시오.
- 선택 집합에 요소를 추가하려면, `Shift` 키를 누른 채 요소를 선택하십시오.
- 선택 집합에서 요소를 제거하려면, `Ctrl` 키를 누른 채 제거할 요소를 다시 선택하십시오. 면을 선택 해제하면 면의 꼭짓점도 모두 선택 해제될 것입니다.

#### *Select Mesh Elements by Polygon* 사용하기

 Select Mesh Elements by Polygon 도구를 활성화한 다음:

- 메시 도형 위로 폴리곤을 그립니다. (왼쪽 클릭은 꼭짓점을 추가하고, `Backspace` 키는 마지막 꼭짓점을 취소하고, `Esc` 키는 폴리곤 그리기를 중단하며, 오른쪽 클릭은 폴리곤을 완성합니다.) 폴리곤과 일부분이라도 중첩하는 모든 꼭짓점 및 면이 선택될 것입니다. 폴리곤 안에 완전히 들어오는 요소만 선택하고자 하는 경우 폴리곤을 그리는 도중 `Alt` 키를 누르십시오.
- 벡터 레이어의 피처를 오른쪽 클릭하면 뜨는 목록에서 도형을 선택하면, 선택한 도형과 조금이라도 중첩하는 메시 레이어의 꼭짓점 및 면이 선택될 것입니다. `Alt` 키를 누른 채 선택하면 도형 안에 완전히 들어오는 요소만 선택됩니다.
- 선택 집합에 요소를 추가하려면, `Shift` 키를 누른 채 요소를 선택하십시오.
- 선택 집합에서 요소를 제거하려면, `Ctrl` 키를 누른 채 선택 폴리곤을 그리십시오.

#### *Select Mesh Elements by Expression* 사용하기

 Select Mesh Elements by Expression 은 메시 요소를 선택하기 위한 또다른 도구입니다. 이 도구를 선택하면 메시표현식 선택기 대화창 이 열리고 이 대화창에서 다음 작업을 할 수 있습니다:

1. 선택 방법을 선택할 수 있습니다:
   - *Select by vertices*: 꼭짓점에 입력한 표현식을 적용시켜 표현식을 만족하는 꼭짓점 및 꼭짓점과 최종적으로 연결된 변/면을 반환합니다.
   - *Select by faces*: 면에 입력한 표현식을 적용시켜 표현식을 만족하는 면 및 면과 최종적으로 연결된 변/꼭짓점을 반환합니다.
2. 선택 표현식을 작성할 수 있습니다. 메시 그룹 에서 사용할 수 있는 함수들을 선택한 선택 방법에 따라 필터링할 것입니다.
3. 선택 집합이 가져야 할 습성을 설정해서 다음을 눌러 쿼리를 실행할 수 있습니다:
   -  *Select*: 레이어에 있는 기존 선택 집합을 모두 대체합니다.
   -  *Add to current selection*: 현재 선택 집합에 선택한 메시 요소를 추가합니다.
   -  *Add to current selection*: 현재 선택 집합에서 선택한 메시 요소를 제거합니다.

### 14.5.4 메시 요소 수정하기

#### 꼭짓점 추가하기

메시 레이어에 꼭짓점을 추가하려면:

1.  Digitize mesh elements 버튼을 누르십시오.
2. A *Vertex Z value* widget is always available on the top right corner of the map canvas during digitizing. It allows setting or overriding the Z coordinate assigned to new vertices, depending on the selected *Z value assignment method*.
3. You can choose other method for assigning Z values to new vertices from the drop-down menu in the  Digitize mesh elements tool.
4. 그리고 다음과 같이 더블클릭하십시오:

- 면 바깥에: 어떤 면과도 연결되지 않은 꼭짓점은"자유"꼭짓점을 추가합니다. 레이어의 편집 모드가 켜져 있다면 이 꼭짓점은 빨간 점으로 표현됩니다.
- 기존 면 (들) 의 변 위에: 변 위에 꼭짓점을 추가하고, 접하는 면 (들) 을 새 꼭짓점과 연결된 삼각형들로 분할합니다.
- 면 안에: 면을 새 꼭짓점에 주변 꼭짓점들을 연결하는 변을 가진 삼각형들로 분할합니다.

With the *Refine neighboring faces when adding vertices* option enabled in the  Digitize mesh elements button drop-down menu, a check is applied to triangular faces that share at least one vertex with the face the new vertex is added to. If their edges do not satisfy Delaunay triangulation rules, they are flipped accordingly.

### 면 추가하기

메시 레이어에 면을 추가하려면:

1.  Digitize mesh elements 버튼을 누르십시오.
2. 맵 캔버스 우상단에 *Vertex Z value* 위젯이 나타납니다. 이 위젯의 값을 이후 추가할 꼭짓점에 할당하고자 하는 Z 좌표로 설정하십시오.
3. Hover over a vertex and click the small triangle that appears next to it.
4. 커서를 다음 꼭짓점 위치로 옮기십시오. 기존 꼭짓점에 스냅하거나 왼쪽 클릭으로 새 꼭짓점을 추가할 수 있습니다.
5. Proceed as above to add as many vertices as you wish for the face. Press `Backspace` button to undo the last vertex.
6. 마우스를 움직이는 동안 면의 형태를 나타내는 고무줄이 표시됩니다. 고무줄이 녹색이면 예상되는 면이 무결하다는 뜻으로, 오른쪽 클릭으로 메시에 면을 추가할 수 있습니다. 고무줄이 빨간색이면 (자체 교차, 기존 면 또는 꼭짓점과 중첩, 구멍 생성 등등 때문에) 면이 무결하지 않다는 뜻으로 메시에 추가할 수 없습니다. 이런 경우 일부 꼭짓점을 되돌려서 도형을 수정해야 할 것입니다.
7. 면 디지타이즈 작업을 중단하려면 `Esc` 키를 누르십시오.
8. 오른쪽 클릭으로 면을 완성하십시오.

### 메시 요소 제거하기

1. 대상 요소를 선택 하십시오.
2.  Digitize mesh elements 도구를 활성화하십시오.
3. 오른쪽 클릭한 다음 다음을 선택하십시오:
   - *Remove Selected Vertices and Fill Hole(s)* 또는 `Ctrl+Del` 키: 꼭짓점 및 연결된 면을 제거하고 이웃 꼭짓점들로부터 삼각형을 생성해서 구멍 (들) 을 채웁니다.
   - *Remove Selected Vertices Without Filling Hole(s)* 또는 `Ctrl+Shift+Del` 키: 꼭짓점 및 연결된 면을 제거하고 구멍 (들) 을 채우지 않습니다.
   - *Remove Selected Face(s)* 또는 `Shift+Del` 키: 면을 제거하지만 꼭짓점은 유지합니다.

선택하지 않은 채 단일 아이템 위에 커서를 가져가면 나타나는 컨텍스트 메뉴로부터 이 옵션들을 사용할 수도 있습니다.

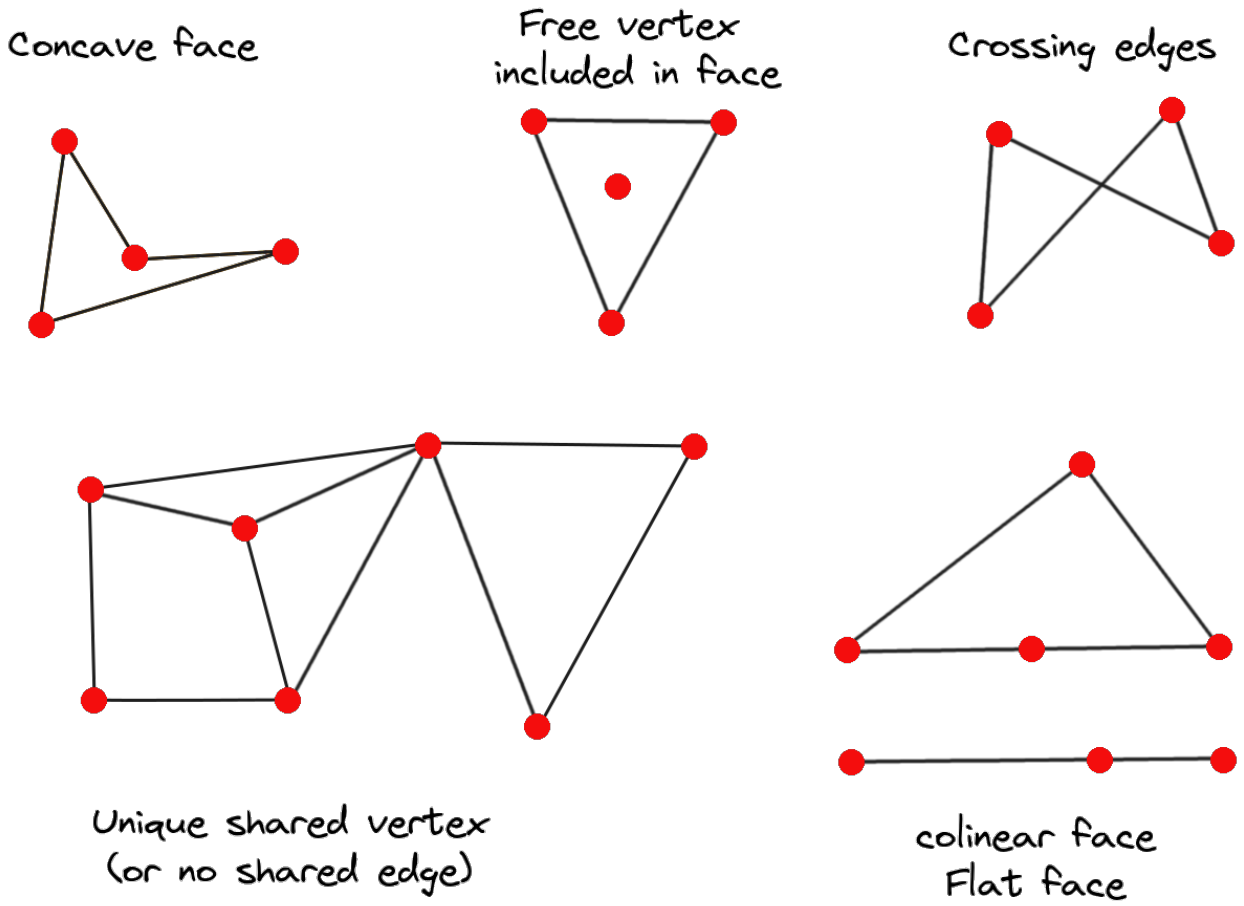



그림 14.17: 무결하지 않은 메시의 예시들

### 메시 요소 이동시키기

메시 레이어의 꼭짓점 및 면을 이동시키려면:

1. 대상 요소를 선택 하십시오.
2. Digitize mesh elements 도구를 활성화하십시오.
3. 요소 이동을 시작하려면, 면/변의 꼭짓점 하나 또는 중심점을 클릭하십시오.
4. 커서를 대상 위치로 움직이십시오. (벡터 피처에 스냅시킬 수 있습니다.)
5. 새 위치로 이동시켰을 때무결하지 않은 메시 를 생성하지 않는다면, 이동시킨 요소가 녹색으로 나타납니다. 다시 클릭해서 요소를 해당 위치로 옮기십시오. 모든 꼭짓점이 선택된 면의 위치가 변경되고, 이웃 면들의 형태도 그에 맞춰 변경됩니다.

### 메시 꼭짓점 변형시키기

Transform Vertices Coordinates 도구는 표현식을 통해 꼭짓점의 X, Y, 그리고/또는 Z 좌표를 편집해서 꼭짓점을 이동시키는 고급 방식을 제공합니다.

1. 좌표를 편집하고자 하는 꼭짓점을 선택하십시오.
2. Transform Vertices Coordinates 아이콘을 클릭하십시오. 대화창이 선택한 꼭짓점들의 개수와 함께 열립니다. 이 상태에서도 선택 집합에 꼭짓점을 추가하거나 제거할 수 있습니다.
3. 사용자가 수정하려는 속성에 따라, *X coordinate*, *Y coordinate* 그리고/또는 *Z value* 옵션을 체크해야 합니다.
4. 그 다음 텍스트란에 대상 위치를 입력하십시오. 숫자값을 입력할 수도 있고 ( Expression dialog 를 통해) 표현식을 작성할 수도 있습니다.
5. With the Import Coordinates of the Selected Vertex pressed, the X, Y and Z boxes are automatically filled with its coordinates whenever a single vertex is selected. You can adjust Z value by using *Get the Z value from the project terrain* button and in that case the Z box is deactivated. A convenient and quick way to adjust vertices individually.
6. *Preview Transform* 버튼을 누르면 꼭짓점의 새 위치를 시뮬레이션해서 변형된 메시를 미리 보기할 수 있습니다.
   - 미리 보기가 녹색인 경우 변형된 메시가 무결하다는 뜻이므로 변형 작업을 적용시킬 수 있습니다.
   - 미리 보기가 빨간색이면 변형된 메시가 무결하지 않다는 뜻이기 때문에 메시가 수정될 때까지 변형 작업을 적용시킬 수 없습니다.
7. 꼭짓점 집합에서 선택한 좌표를 수정하려면 *Apply Transform* 버튼을 누르십시오.

### 메시 도형 형태 변경

#### 컨텍스트 메뉴

1. Digitize mesh elements 도구를 활성화하십시오.
2. 메시 항목 (들) 을 선택하거나, 선택하지 마십시오.
3. 메시 요소 위에 커서를 가져가면 메시 요소가 강조됩니다.
4. 이를 오른쪽 클릭하면 다음 작업을 할 수 있습니다:
   - 항목 (들) 을 제거 할 수 있습니다.

- *Split Selected Face(s)* (*Split Current Face*): 사용자가 커서를 올려놓은 면, 또는 선택한 사각 (quad) 메시 면 각각을 삼각형 2 개로 분할합니다.
- *Delaunay Triangulation with Selected vertices*: 선택한 자유 꼭짓점들을 이용해서 삼각면을 작성합니다.
- *Refine Selected Face(s)* (*Refine Current Face*): 면의 각 변 중간에 추가된 꼭짓점을 바탕으로 면 하나를 4 개로 분할합니다. (삼각형은 삼각형들로, 사각형은 사각형들로 분할됩니다.) 또한 새 꼭짓점에 연결된 인접 면을 삼각 분할할 것입니다.

### 변 마커

 Digitize mesh elements 도구가 활성화된 상태에서 사용자가 변 위로 커서를 가져가면 해당 변이 강조되고 쌍방향 작업이 가능해집니다. 작업의 맥락에 따라, 다음과 같은 마커들을 사용하게 될 수도 있습니다:

- 변의 중앙에 나타나는 **정사각형**: 이 마커를 클릭하면 양 극단의 꼭짓점들을 선택합니다.
- 변 양쪽의 두 면을 병합할 수 있는 경우 나타나는 **십자표**: 이 마커를 클릭하면 변을 삭제하고 두 면을 병합합니다.
- 변이 두 삼각형 사이에 있는 경우 나타나는 **원**: 이 마커를 클릭하면 변을 뒤집습니다. 즉, 두 면의 다른"자유" 꼭짓점 2 개에 대신 연결합니다.

### *Force by Selected Geometries* 도구

 Force by Selected Geometries 도구는 라인 도형을 이용해서 절단선 (break line) 을 적용하는 고급 방식을 제공합니다. 절단선은 메시의 변을 절단선을 따라 강제로 분할할 것입니다. 작업이 완료되고 나면 절단선을 영구적인 것으로 간주하지 않기 때문에, 변들이 제약조건으로 작동하지 않을 것이고 다른 모든 변과 마찬가지로 수정할 수 있게 됩니다. 이 도구를 이용하면 예를 들어 강둑이나 도로의 경사면 경계 같은 정확한 라인으로 메시 레이어를 부분적으로 수정할 수 있습니다.

1.  Force by Selected Geometries 도구를 활성화하십시오.
2. "강제선 (forcing line)"으로 사용할 도형을 지정하십시오. 다음과 같은 도형을 지정할 수 있습니다:
   - 맵 캔버스에 있는 라인 또는 폴리곤 피처에서 선택: 벡터 피처를 오른쪽 클릭하면 나타나는 컨텍스트 메뉴의 목록에서 도형을 선택할 수 있습니다.
   - 메시 프레임 위에 그린 가상의 선: 왼쪽 클릭으로 꼭짓점을 추가하고, 오른쪽 클릭으로 선을 완성하십시오. *Vertex Z value* 위젯 또는 *Advanced Digitizing Panel* 이 열려 있는 경우 *z* 위젯을 통해 꼭짓점의 Z 값을 설정하십시오. 가상의 선의 꼭짓점이 메시의 꼭짓점 아니면 3 차원 벡터 피처의 꼭짓점 또는 선분에 스냅되어 있는 경우라면 새 꼭짓점이 스냅되어 있는 요소의 Z 값을 가질 것입니다.

 Force by Selected Geometries 도구의 드롭다운 메뉴에서 설정할 수 있는 옵션에 따라 달라지는 방식으로 라인 도형 또는 폴리곤의 경계선과 중첩하는 메시 면에 영향을 미칠 것입니다:

-  *Add new vertex on intersecting edges*: 이 옵션을 선택하면 강제선 (절단선) 이 변과 교차할 때마다 새 꼭짓점을 추가합니다. 이 옵션을 선택한 경우 강제선과 교차하는 모든 면을 강제선을 따라 분할하게 됩니다.

  이 옵션을 체크 해제하면, 강제선과 교차하는 면을 제거하고 기존 꼭짓점과 강제선의 꼭짓점만을 이용해서 삼각 분할한 면으로 대체합니다. (강제선과 교차하는 경계선의 변 위에도 새 꼭짓점을 추가합니다.)

- *Interpolate Z value from*: 새 꼭짓점의 Z 값을 어떻게 계산할지 설정합니다. 다음 가운데 선택할 수 있습니다:
  - *Mesh* 자체: 새 꼭짓점이 들어가는 면의 꼭짓점들로부터 새 꼭짓점의 Z 값을 보간합니다.
  - 또는 *Forcing line*: 3 차원 벡터 피처 또는 사용자가 그린 선이 강제선을 정의하는 경우 해당 도형에서 새 꼭짓점의 Z 값을 파생시킵니다. 2 차원 라인 피처의 경우, 새 꼭짓점의 Z 값은 *Vertex Z value* 입니다.

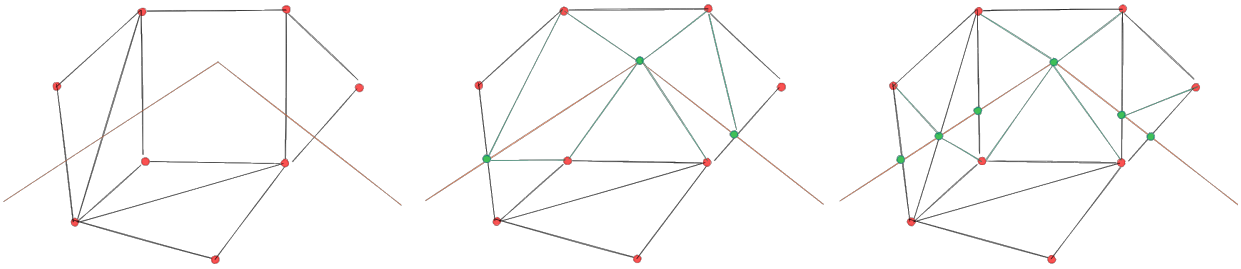


그림 14.18: 라인 도형으로 메시를 강제 분할 - 변들의 교차점에 새 꼭짓점을 추가하지 않은 (가운데) 결과물과 추가한 (오른쪽) 결과물

- *Tolerance*: 기존 메시 꼭짓점이 강제선에 이 허용 오차 값보다 가까이 있는 경우 강제선 위에 새 꼭짓점을 생성하지 않고 대신 기존 꼭짓점을 사용합니다. *Meters at Scale* 또는 *Map Units* 단위로 이 값을 설정할 수 있습니다. (자세한 내용은단위 선택기 를 참조하세요.)

### 14.5.5 메시 인덱스 재생성하기

편집 작업 도중 되돌리기/다시 실행 (undo/redo) 작업을 빠르게 하기 위해, QGIS 는 삭제된 요소를 위한 텅 빈 공간을 유지합니다. 이때 메모리 사용량이 늘어날 수도 있고 메시 구조 생성이 비효율적으로 바뀔 수도 있습니다.

*Mesh* | *Reindex Faces and Vertices* 도구는 이런 구멍을 제거하고 면과 꼭짓점의 인덱스가 연속적이고 어느 정도 합리적인 순서가 되도록 번호를 다시 매기도록 설계되었습니다. 이렇게 하면 면과 꼭짓점 사이의 관계를 최적화하고 계산의 효율을 향상시킵니다.

> **참고**
>
> *Reindex Faces and Vertices* 도구는 레이어를 저장하고 되돌리기/다시 실행 스택을 초기화하기 때문에 모든 되돌리기를 비활성화시킵니다.

## 14.6 메시 계산기

최상위 *Mesh* 메뉴의 *Mesh Calculator* 도구를 이용하면 기존 데이터셋 그룹에 대한 수학적이고 논리적인 계산을 수행해서 새로운 데이터셋 그룹을 생성할 수 있습니다. (그림 14.19 을 참조하세요.)

*Datasets* 프레임은 활성 메시 레이어의 모든 데이터셋 그룹의 목록을 담고 있습니다. 표현식에 데이터셋 그룹을 사용하려면 이 목록에서 데이터셋 그룹의 이름을 더블클릭하십시오. *Mesh calculator expression* 란에 선택한 데이터셋 그룹이 추가될 것입니다. 그러면 연산자를 이용해서 계산 표현식을 작성하거나 또는 사용자가 직접 표현식을 입력할 수 있습니다.

*Result Layer* 프레임에서 산출 레이어의 속성을 환경 설정할 수 있습니다:

- ☑ *Create on-the-fly dataset group instead of writing layer to disk*:
  - 이 옵션을 체크 해제하는 경우, 디스크 상에 산출물을 새 평문 텍스트 파일로 저장합니다. *Output File* 경로 및 *Output Format* 은 필수입니다.
  - 이 옵션을 체크하면 메시 레이어에 새 데이터셋 그룹을 추가할 것입니다. 이 데이터셋 그룹의 값들은 메모리에 저장되는 것이 아니라 필요한 경우 메시 계산기에 입력된 공식으로 각 데이터셋을 계산해서 나옵니다. 이런 가상 데이터셋 그룹은 프로젝트와 함께 저장되며, 필요한 경우 레이어의 *Source* 속성 탭에서 제거하거나 또는 영구적인 파일로 저장할 수 있습니다.

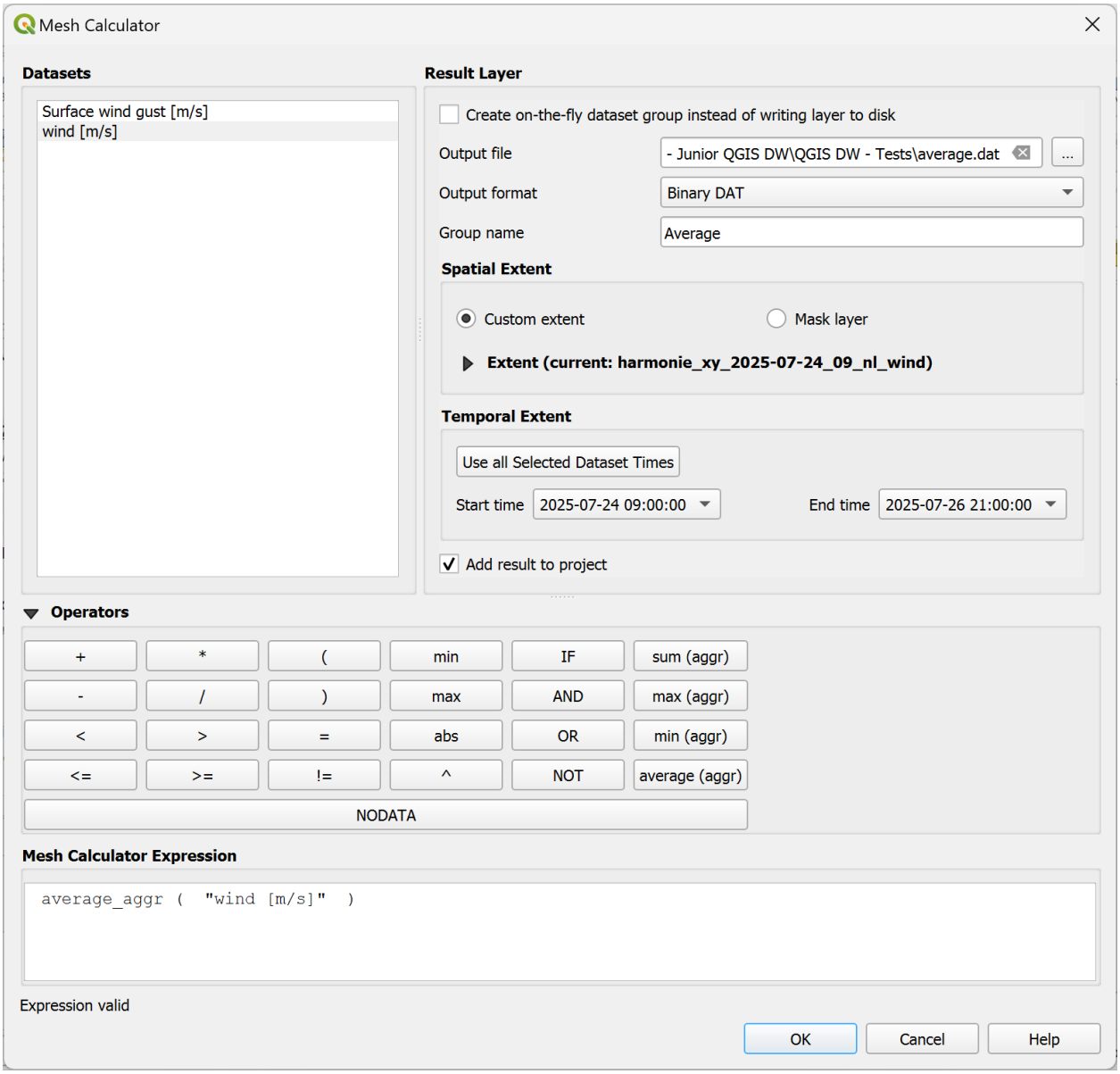



그림 14.19: 메시 계산기

둘 중 어떤 경우라도, 산출 데이터셋 그룹의 *Group Name* 을 지정해야 할 것입니다.

- 계산에 넣을 *Spatial extent* 는 다음 가운데 하나일 수 있습니다:
  - defined as a *Custom extent* using the *extent selector* widget.
  - 프로젝트의 폴리곤 레이어로 정의 (*Mask layer*): 폴리곤 피처 도형을 이용해서 메시 레이어 데이터셋을 잘라낼 수 있습니다.
- *Temporal extent*: 기존 데이터셋 그룹의 타임스탬프에서 선택한 *Start time* 및 *End time* 옵션으로 계산에 넣을 데이터셋 시계열 범위를 설정할 수 있습니다. *Use all selected dataset times* 버튼을 눌러 전체 시계열 범위를 가져와 *Start time* 및 *End time* 란을 채울 수도 있습니다.
- With the  *Add result to project* checkbox, the result layer will automatically be added to the legend area and can be visualized. Checked by default.

*Operators* 프레임이 사용할 수 있는 모든 연산자를 담고 있습니다. 메시 계산기 표현식란에 연산자를 추가하려면, 적절한 버튼을 클릭하십시오. 산술 계산 (+, –, *, ⋯) 및 통계 함수 (`min`, `max`, `sum (aggr)`, `average (aggr)`, ⋯) 를 사용할 수 있습니다. 조건 표현식 (=, !=, <, >=, `IF`, `AND`, `NOT`, ⋯) 은 거짓인 경우 0 을 또는 참인 경우 1 을 반환하기 때문에, 다른 연산자 및 함수와 함께 사용할 수 있습니다. 표현식에 `NODATA` 값도 사용할 수 있습니다.

*Mesh Calculator Expression* 위젯은 실행할 표현식을 표시하고 편집할 수 있게 해줍니다.

CHAPTER 15

# 벡터 타일 작업

## 15.1 벡터 타일이란?

벡터 타일이란 지리 데이터의 패킷으로, 웹을 통해 전송하기 위해 사전 정의된 대략 정사각형 모양의"타일"을 말합니다. 사전 렌더링된 래스터 맵 타일과 벡터 맵 타일을 결합한 것입니다. 벡터 타일 서버는 사전 렌더링된 맵 이미지 대신 각 타일의 경계선을 따라 자른 벡터 맵 데이터를 반환합니다. 이렇게 자른 타일은 피라미드 접근법에서 파생된 벡터 타일 서비스의 확대/축소 수준을 나타냅니다. 이 구조를 사용하면, 타일화되지 않은 벡터 맵보다 데이터 전송량이 줄어듭니다. 현재 확대/축소 수준에서 현재 맵 뷰에 보이는 데이터만 전송하면 되니까요. 또 타일화된 래스터 맵과 비교해도 – 벡터 데이터가 일반적으로 렌더링된 비트맵보다 훨신 용량이 작기 때문에 – 데이터 전송량이 훨씬 줄어듭니다. 벡터 타일에는 할당된 스타일 작업이 없기 때문에, 데이터를 표시하기 위해서는 QGIS 가 지도 제작 스타일을 적용해야 합니다.

## 15.2 지원 포맷

다음을 통해 벡터 타일을 지원할 수 있습니다:

- 원격 소스 (HTTP/S) - XYZ 템플릿과 함께 – 예: `type=xyz&url=http://example.com/{z}/{x}/{y}.pbf`
- 로컬 파일 - XYZ 템플릿과 함께 – 예: `type=xyz&url=file:///path/to/tiles/{z}/{x}/{y}.pbf`
- 로컬 MBTiles 데이터베이스 – 예: `type=mbtiles&url=file:///path/to/file.mbtiles`

*Data Source Manager* 대화창에 있는  *Vector Tile* 탭을 통해 QGIS 로 벡터 타일 데이터셋을 불러올 수 있습니다. 더 자세한 내용은벡터 타일 서비스 사용하기 를 읽어보세요.

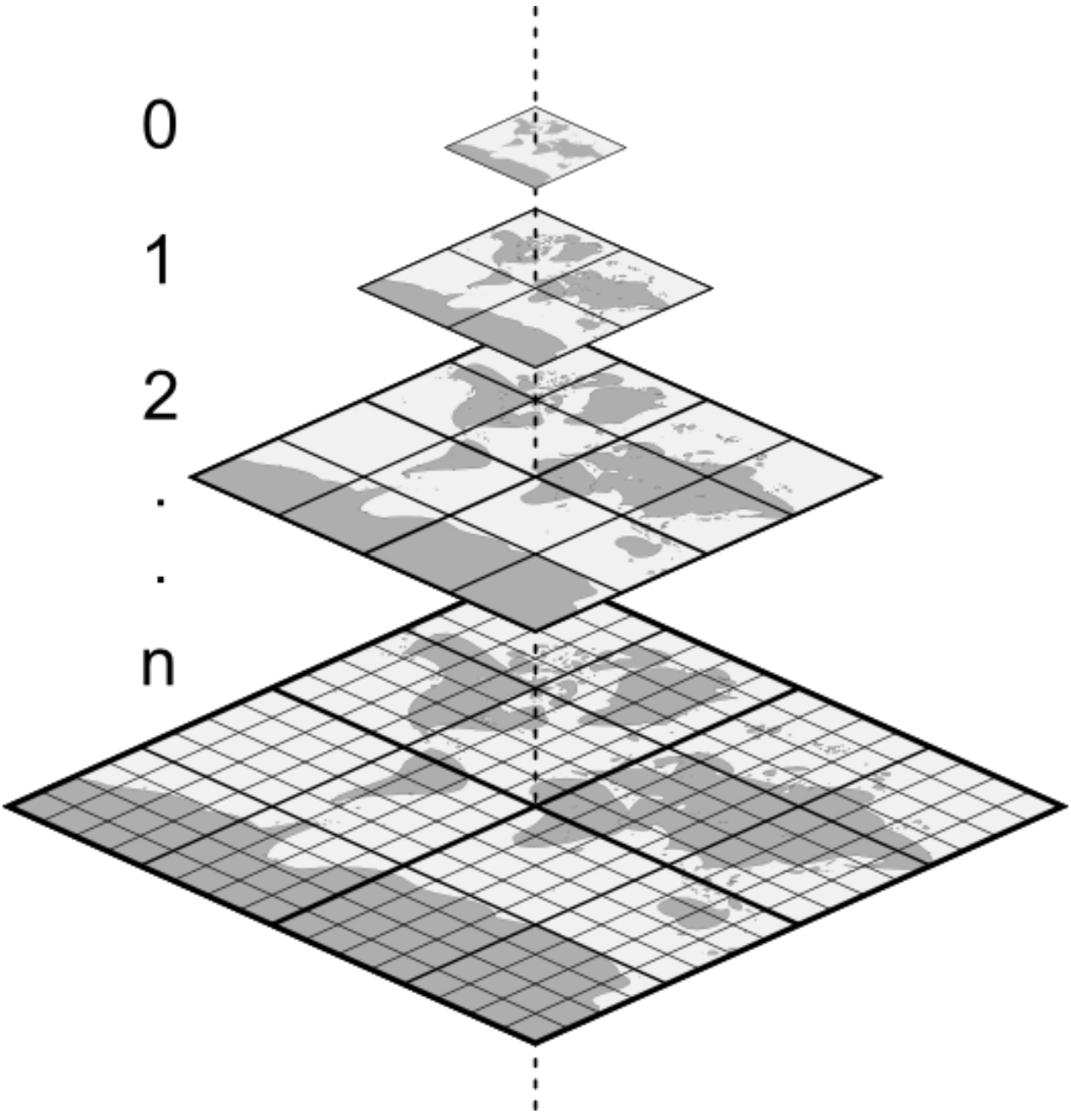
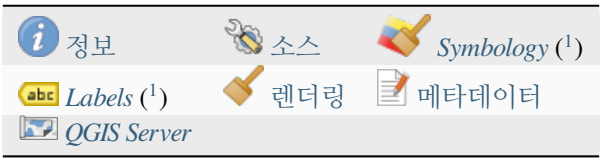



그림 15.1: 확대/축소 수준을 가진 벡터 타일의 피라미드 구조

## 15.3 벡터 타일 데이터셋 속성

벡터 타일 *Layer Properties* 대화창은 다음 탭들로 이루어져 있습니다:




### 15.3.1 정보 속성

*Information* 탭은 읽기 전용으로, 현재 레이어에 대한 요약 정보 및 메타데이터를 한 눈에 볼 수 있는 흥미로운 장소입니다. 다음과 같은 정보를 제공합니다:

- 레이어 제공자 기반: 이름, URI, 소스 유형 및 경로, 확대/축소 수준 개수
- custom properties, used to store in the active project additional information about the layer. More properties can be created and managed using PyQGIS, specifically through the setCustomProperty() method.
- the Coordinate Reference System: name, units, method, accuracy, reference (i.e., whether it is static or dynamic)
- 채워진 메타데이터 로부터 선택: 접근, 범위, 링크, 연락처, 이력, …

### 15.3.2 소스 속성

*Source* 탭은 선택한 벡터 타일에 대한 다음과 같은 기본 정보를 표시합니다:

- *Layer name*: *Layers Panel* 에 표시될 레이어명
- *Coordinate Reference System*: 레이어의좌표계 *(CRS)* 를 표시합니다. 드롭다운 목록에서 최근 사용된 좌표계를 선택하거나 Select CRS 버튼을 (좌표계 선택기 참조) 클릭하면 레이어 좌표계를 변경할 수 있습니다. 레이어 좌표계가 틀렸거나, 지정되지 않은 경우에만 변경하십시오.
- For file based vector tiles (`.vtpk` or `.mbtiles`), a *Layer source* group indicates the path to the source of the dataset and allows for replacing the loaded layer. Edit the path shown in the text box or press … Browse to select another file on the disk.

### 15.3.3 심볼 및 라벨 속성

#### 규칙 설정하기

벡터 타일 하나는 포인트, 라인, 폴리곤 도형들로 이루어져 있기 때문에 각각의 심볼들을 사용할 수 있습니다. (심볼 그리고/또는 라벨이 적용된) 지도 제작 스타일을 구현하려면 다음 작업 가운데 하나를 시도해볼 수 있습니다:

- 벡터 타일 연결 을 생성할 때 *Style URL* 을 이용하십시오. QGIS 에 레이어를 불러오는 즉시 *Symbology* 탭에 심볼이 나타날 것입니다.
- 또는 레이어 속성 아래 각각 대응하는 탭에서 사용자 지정 심볼 및 라벨을 작성하십시오. QGIS 는 기본적으로 피처의 도형 유형에 따라 동일한 심볼을 할당합니다.

[1] Also available in the *Layer styling panel*

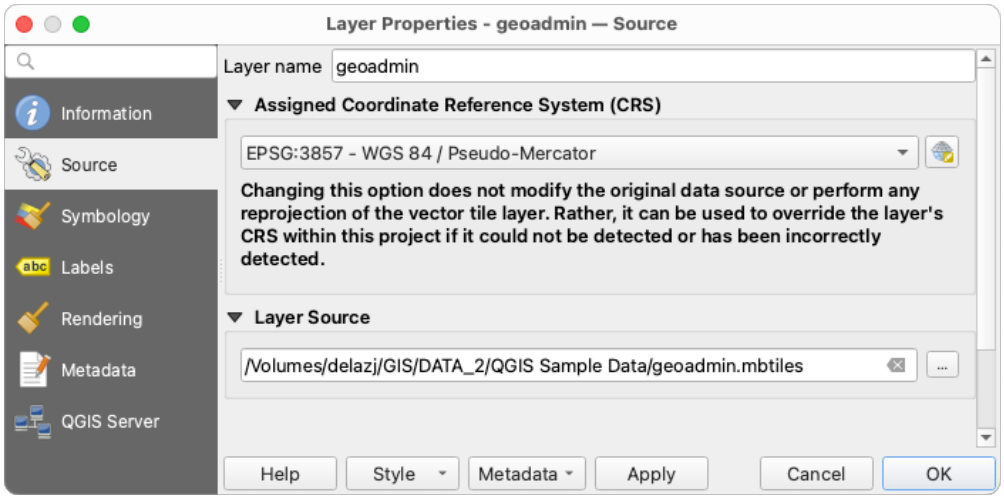



그림 15.2: 벡터 타일 속성 — 소스 대화창

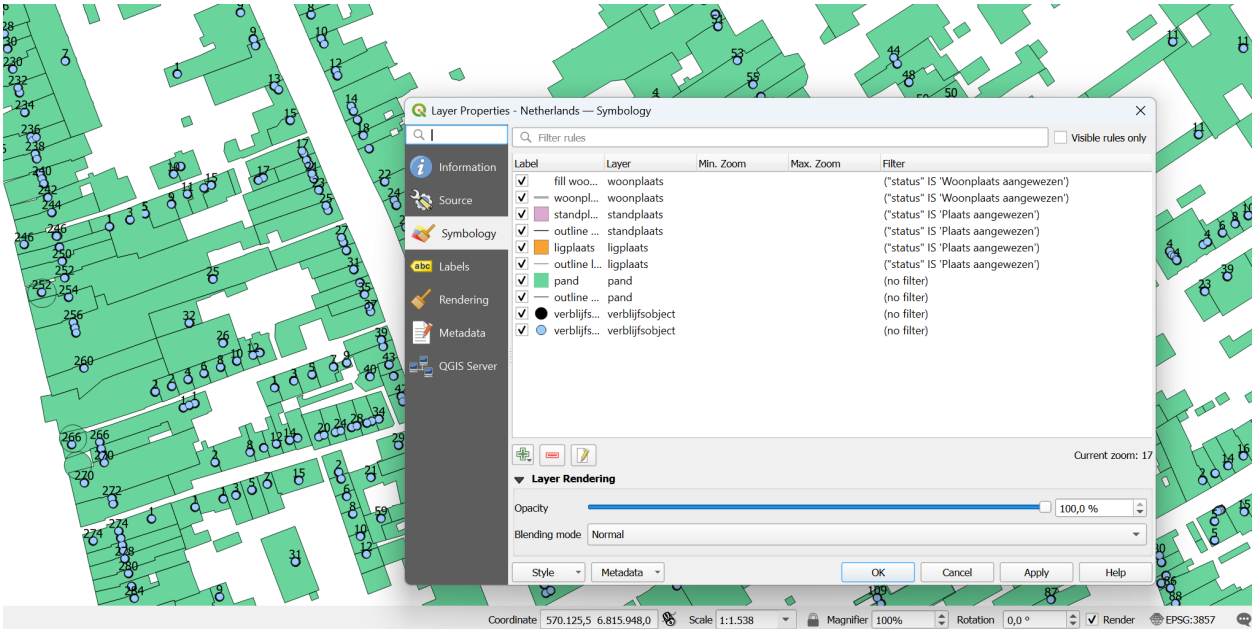



그림 15.3: 벡터 타일 레이어 심볼

이 두 작업에서 벡터 타일의 스타일 설정은 피처에 적용된, 다음과 같은 일련의규칙 을 따르게 됩니다:

- 벡터심볼 또는라벨
- *Label*: 규칙의 포괄적인 식별을 위한 제목입니다.
- `(all layers)`: 규칙이 모든 레이어에 적용되지 않은 경우, 규칙을 적용해야 할 특정 *Layer* 의 이름을 지정해야 합니다.
- *Min. Zoom* 및 *Max. Zoom*: 표시할 범위를 지정합니다. 심볼과 라벨을 확대/축소 수준에 따라 표시되도록, 또는 숨기도록 할 수 있습니다.
- *Filter*: 스타일을 적용할 피처를 식별하기 위한 QGIS 표현식입니다.

Add rule 버튼을 누르고 피처의 도형 유형에 맞춰 심볼 유형 (*Marker*, *Line*, *Fill*) 을 선택하면 각각의 규칙이 추가됩니다. 선택한 규칙을 제거 Remove selected rules 하거나 편집 Edit current rule 할 수도 있습니다.

하단에 *Current Zoom* 이 보입니다. 대화창 상단의 *Visible rules only* 옵션을 체크하면 현재 확대/축소 수준에서 가시화되는 규칙 목록으로만 필터링됩니다. 이렇게 하면 복잡한 벡터 스타일 작업을 쉽게 해주고, 문제가 생길 가능성이 있는 규칙을 찾기도 쉬워집니다. *Filter rules* 텍스트 상자도 *Label*, *Layer* 및 *Filter* 항목을 검색하여 규칙을 쉽게 찾을 수 있도록 도와줍니다.

그림 15.3 에서, 오픈스트리트맵의 `landuse` 레이어를 위한 스타일을 설정합니다. 가시성을 강화하기 위해 규칙 대부분을 끈 상태입니다.

### 레이어 렌더링

*Symbology* 탭에서 레이어의 모든 피처에 대해 일괄적으로 적용되는 다음 옵션들을 설정할 수도 있습니다:

- *Opacity*: 이 도구를 통해 맵 캔버스에서 아래에 있는 레이어를 가시화할 수 있습니다. 슬라이드 바를 통해 사용자 벡터 레이어의 가시성을 필요에 따라 조정하십시오. 슬라이드 바 옆에 있는 메뉴에서 가시성을 정확한 백분율로 설정할 수도 있습니다.
- *Blending mode*: 이 도구들을 통해 사용자가 그래픽 프로그램에서만 가능하다고 알고 있던 특별한 렌더링 효과를 줄 수 있습니다. 위 그리고 아래에 있는 레이어의 픽셀을혼합 모드 에서 설명한 설정으로 혼합합니다.

### 스타일

대부분의 탭 하단에서 사용할 수 있는 *Styles* 메뉴는 벡터 타일에 적용할 스타일을 저장하고, 불러오고, 생성하거나, 전환할 수 있는 단축키입니다. QGIS 에서는 벡터 타일의 스타일을 *QML* 파일로 저장할 수 있으며, 다음 가운데 하나로 가져올 수도 있습니다:

- *QML* 파일 (*QML* *QGIS* 스타일 파일 포맷)
- *MapBox GL Json* 스타일 환경설정 파일

더 자세한 내용은레이어 속성 저장 및 공유 을 참조하세요.

### 15.3.4 렌더링 속성

*Scale dependent visibility* 에서는 *Maximum (inclusive)* 및 *Minimum (exclusive)* 축척을 설정해서 피처가 보이게 될 축척 범위를 정의할 수 있습니다. 이 범위를 벗어나면, 피처를 숨깁니다. Set to current canvas scale 버튼을 클릭하면 현재 맵 캔버스의 축척을 가시성 범위의 한계값으로 설정할 수 있습니다. 자세한 내용은가시성 축척 선택기 을 참조하세요.

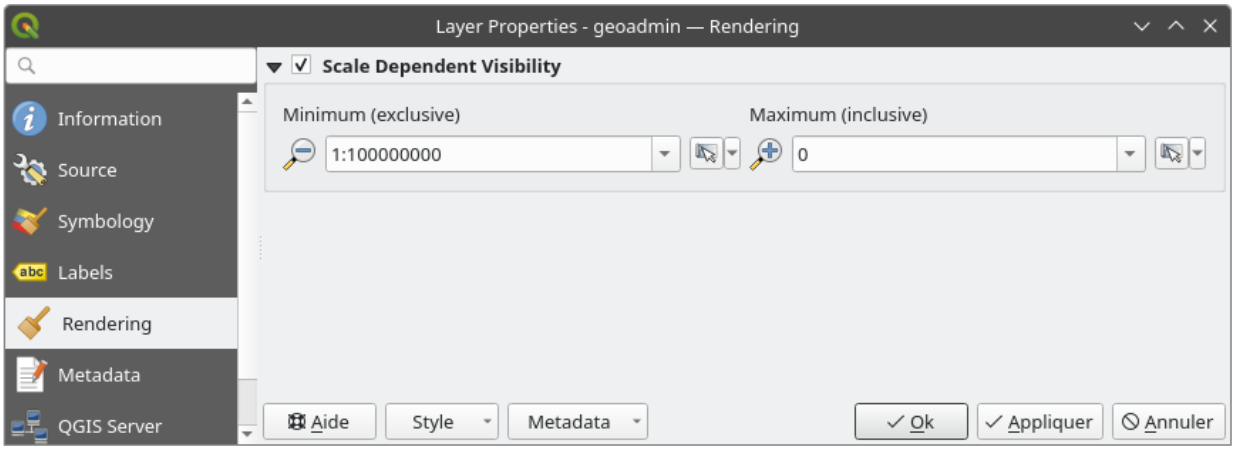



그림 15.4: 벡터 타일 속성 — 렌더링 대화창

### 15.3.5 메타데이터 속성

*Metadata* 탭에 사용자 레이어에 대한 메타데이터 리포트를 생성하고 편집할 수 있는 옵션이 있습니다. 더 자세한 정보는메타데이터 를 참조하세요.

## 15.4 QGIS Server Properties

The *QGIS Server* tab helps you configure settings of the data when published by QGIS Server. The configuration concerns:

- *Description*: provides information to describe the data, such as *Short name*, *Title*, *Abstract*, a *Keyword list*, and a *Data URL* whose *Format* can be in `text/html`, `text/plain` or `application/pdf`.
- *Attribution*: a *Title* and *URL* to identify who provides the data
- *Metadata URL*: a list of *URL* for the metadata that can be of `FGDC` or `TC211` *Type*, and in `text/plain` or `text/xml` *Format*
- *Legend URL*: a *URL* for the legend, in either `image/png` or `image/jpeg` *Format*

CHAPTER 16

# 점구름 작업

## 16.1 점구름 소개

**점구름이란 무엇일까요?**

점구름 (point cloud) 이란 수많은 (수십억, 때론 수조에 달하는) 개별 데이터 포인트로 이루어진 공간의 3 차원 이미지를 말합니다. 캡처 방법에 따라, 점구름은 일반적으로 색상값 또는 강도 (intensity) 같은 캡처 과정에서 나온 추가 속성도 가집니다. 이런 속성들을 이용하면, 예를 들어 점구름을 서로 다른 색상으로 표시할 수 있습니다. QGIS 에서는 경관의 (또는 또다른 공간의) 3 차원 이미지를 생성하는 데 점구름을 이용할 수 있습니다.

**지원 포맷**

QGIS supports the data formats Entwine Point Tile (EPT) and LAS/LAZ. To work with point clouds, QGIS always saves the data in EPT. EPT is a storage format that consists of several files stored in a common folder. To allow quick access to the data, EPT uses indexing. For more information on the EPT format, see entwine homepage

데이터 포맷이 LAS 또는 LAZ 인 경우, QGIS 가 데이터를 처음 불러올 때 EPT 로 변환할 것입니다. 파일 용량에 따라 시간이 걸릴 수도 있습니다. 이 과정에서 LAS/LAZ 파일이 있는 폴더에 `ept_` + `name_LAS/LAZ_file` 스키마를 따르는 하위 폴더를 하나 생성합니다. 이런 하위 폴더가 이미 존재하는 경우, QGIS 는 즉시 EPT 를 불러옵니다. (따라서 불러오기 시간을 단축합니다.)

**알아두면 좋을 정보**

QGIS 에서는 (아직) 점구름을 편집할 수 없습니다. 사용자의 점구름을 조작하고 싶다면 오픈소스 점구름 공간 처리 도구인 CloudCompare 를 사용하면 됩니다. 또한 포인트 데이터 추상 라이브러리 (Point Data Abstraction Library) (PDAL - GDAL 과 유사) 도 점구름을 편집할 수 있는 옵션을 제공합니다. (PDAL 은 명령 줄 전용 유틸리티입니다.)

수많은 데이터 포인트 개수 때문에, QGIS 에서 점구름의 속성 테이블을 표시할 수가 없습니다. 하지만  식별 도구 가 점구름을 지원하므로 단일 데이터 포인트조차도 속성을 모두 표시할 수 있습니다.

If you want to create a new layer, in the same or another supported format, from an existing point cloud layer, see *Creating new layers from an existing layer*.

## 16.2 점구름 속성

점구름의 *Layer Properties* 대화창에서 레이어와 레이어 렌더링에 대한 일반 설정을 할 수 있습니다. 이 대화창은 레이어 관련 정보도 제공합니다.

*Layer Properties* 대화창을 열려면:

- *Layers* 패널에서 레이어를 더블 클릭하거나, 오른쪽 클릭한 다음 컨텍스트 메뉴에서 *Properties…*를 선택하십시오.
- 레이어를 선택한 다음 *Layer* → *Layer Properties…*메뉴를 클릭하십시오.

점구름 *Layer Properties* 대화창은 다음 탭들로 이루어져 있습니다:




**참고**

속성 대화창 하단에 있는 *Style* 메뉴를 이용해서 점구름 레이어 속성 대부분을 `.qml` 파일로 저장하거나 `.qml` 파일로부터 불러올 수 있습니다. 자세한 내용은레이어 속성 저장 및 공유 를 참조하세요.

### 16.2.1 정보 속성

*Information* 탭은 읽기 전용으로, 현재 레이어에 대한 요약 정보 및 메타데이터를 한 눈에 볼 수 있는 흥미로운 장소입니다. 다음과 같은 정보를 제공합니다:

- 프로젝트 이름, 소스 경로, 최신 저장 시간 및 용량, 사용하는 제공자 등의 일반 정보
- custom properties, used to store in the active project additional information about the layer. Default custom properties may include *layer notes*. More properties can be created and managed using PyQGIS, specifically through the setCustomProperty() method.
- 레이어 제공자 기반: 범위 및 포인트 개수
- 좌표계: 이름, 단위, 메소드, 정확도, 참조 (예: 정적인지 동적인지 여부)
- 제공자가 전송하는 메타데이터: 생성 날짜, 버전, 데이터 포맷, X/Y/Z 척도, …
- (메타데이터를 편집할 수 있는) 메타데이터 탭에서 선택: 접근 정보, 범위, 링크, 연락처, 이력, …

[1] Also available in the *Layer styling panel*

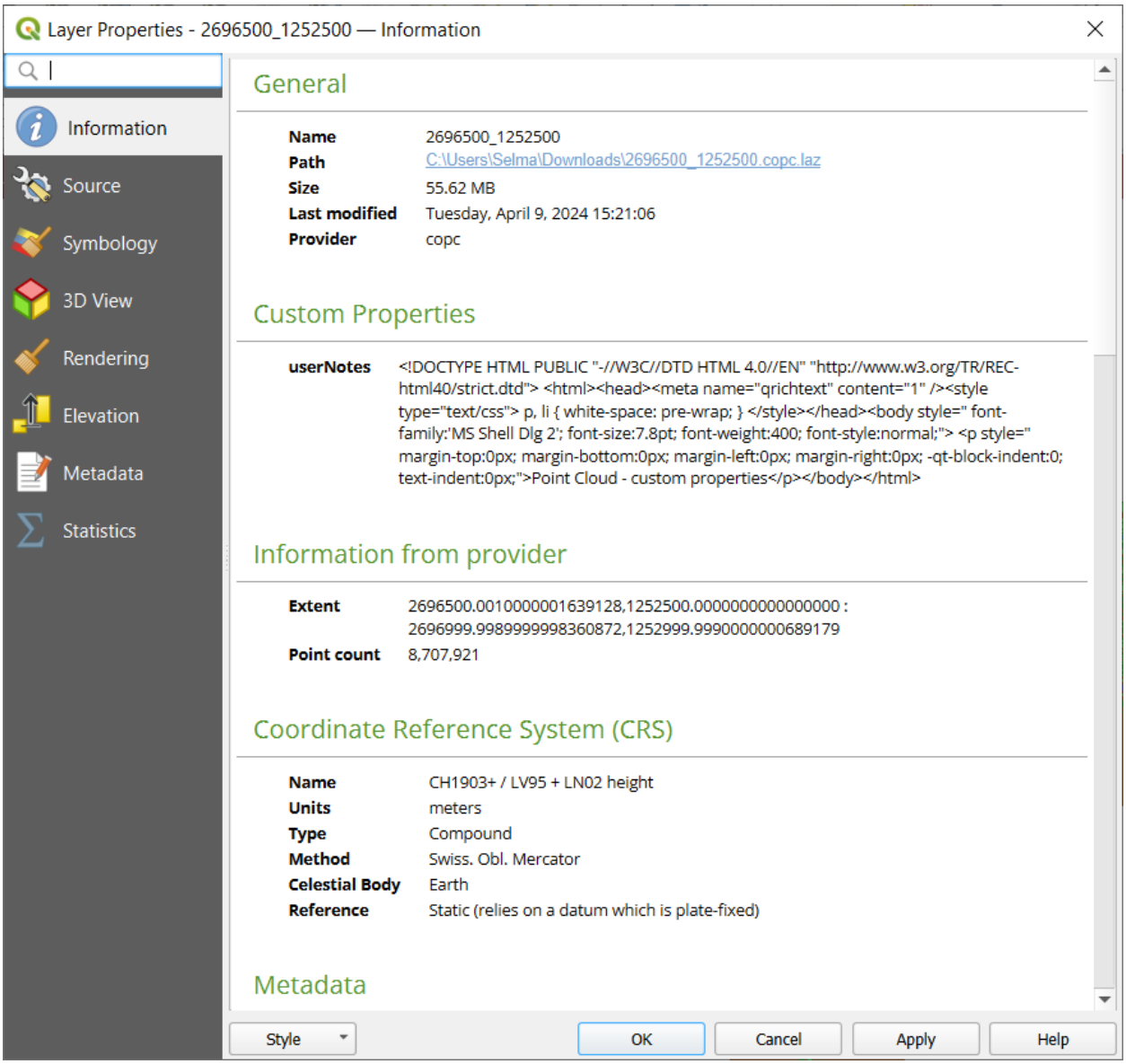



그림 16.1: 점구름 정보 탭

### 16.2.2 소스 속성

*Source* 탭에서 점구름 레이어에 대한 다음과 같은 기본 정보를 보고 편집할 수 있습니다:

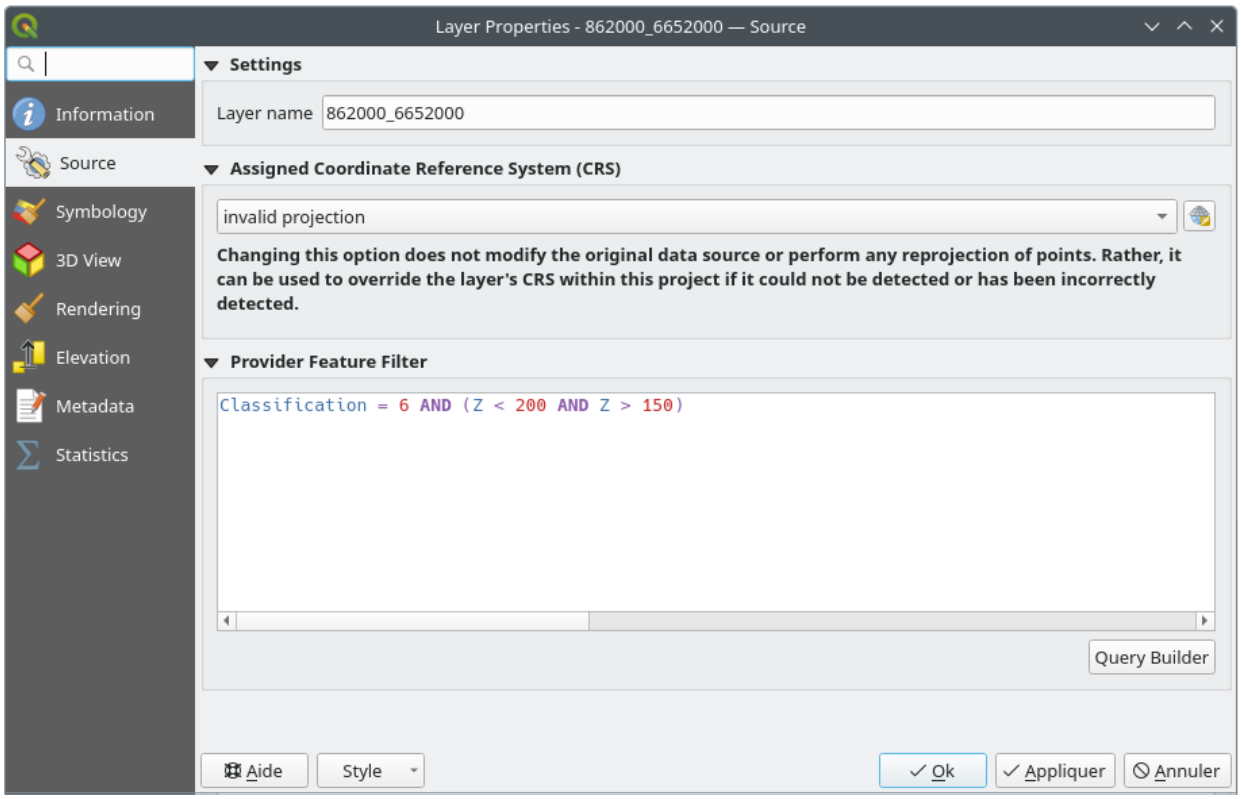



그림 16.2: 점구름 소스 탭

- *Settings*: (*Layers Panel* 에서 표현식 등을 이용해서, 또는 인쇄 조판 범례에서, 또는…) 프로젝트 내의 레이어를 식별하기 위해 쓰일, 레이어 파일명과는 다른 레이어 이름을 설정할 수 있습니다.
- *Assigned Coordinate Reference System (CRS)*: 레이어에 할당된좌표계 를 변경할 수 있습니다. 드롭다운 목록에서 최근 사용된 좌표계를 선택하거나 Select CRS 버튼을 (좌표계 선택기 참조) 클릭하면 레이어 좌표계를 변경할 수 있습니다. 레이어에 적용된 좌표계가 틀렸거나 적용된 좌표계가 없는 경우에만 변경하십시오.
- *Provider Feature Filter*: 현재 프로젝트에 사용할 데이터를 PDAL 데이터 제공자 수준에서 함수와 표현식을 사용해 하위 집합으로 제한할 수 있습니다. 필터를 설정하려면 하단에 있는 *Query Builder* 버튼을 클릭하십시오.

  이 대화창의 하단 부분 *Provider specific filter expression* 란에 제공자 전용 필터 표현식을 입력할 수 있습니다. 이 표현식은 다음을 사용해서 작성할 수 있습니다:

  - *Fields*: 필드 목록은 점구름 레이어의 모든 속성을 담고 있습니다. 표현식에 속성을 추가하려면, 속성 이름을 더블클릭하거나 텍스트란에 이름을 입력하기만 하십시오.
  - *Values*: 이 프레임의 목록은 현재 선택한 속성의 값 또는 통계를 키: 값 쌍으로 담고 있습니다. 표현식 텍스트 상자에 값을 추가하려면, 목록에서 값의 이름을 더블클릭하십시오. 속성의 유형에 따라 키 또는 값 가운데 하나가 표현식에 추가될 것입니다. 또는 표현식 텍스트 상자에 값을 입력하십시오.
  - *Operators*: 연산자 툴바는 사용할 수 있는 모든 연산자를 담고 있습니다. 표현식 텍스트 상자에 연산자를

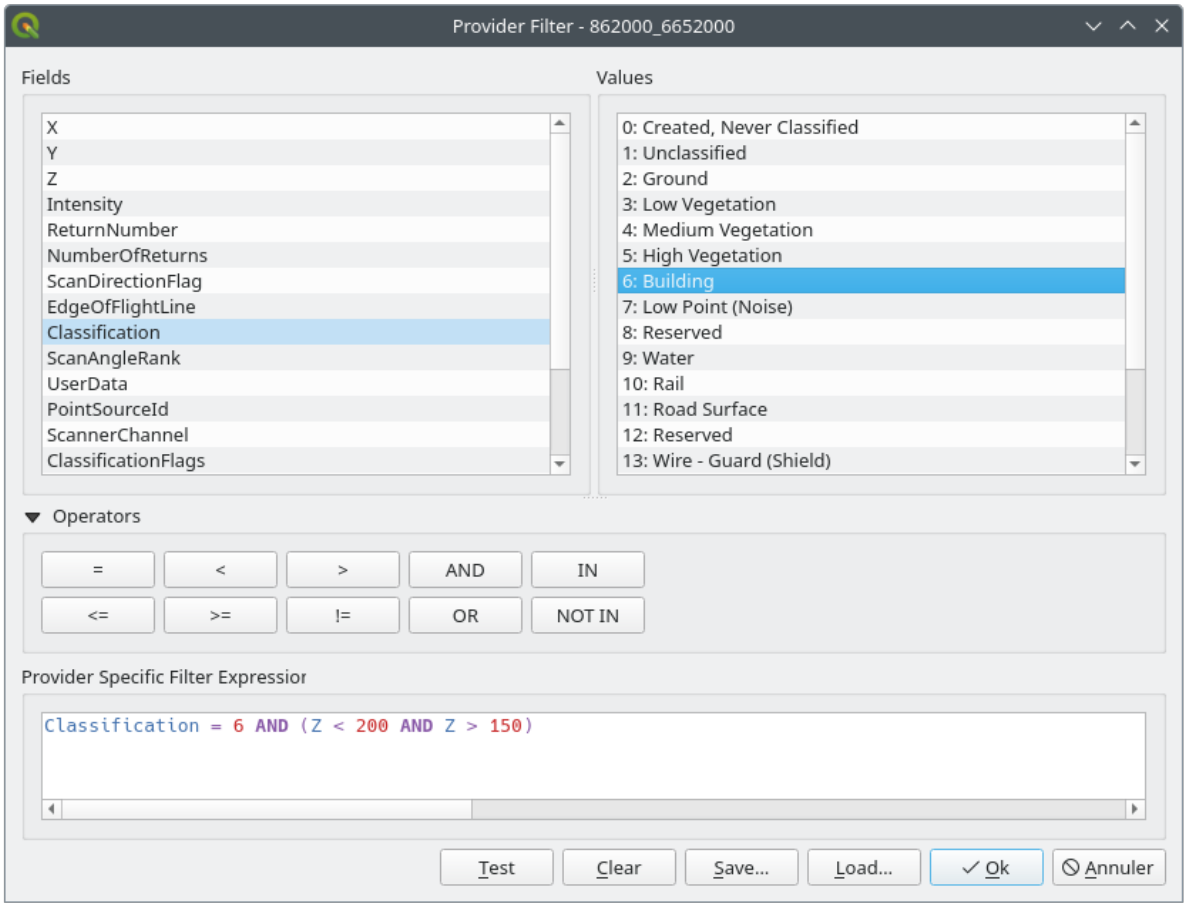



그림 16.3: 점구름 레이어를 특정 고도의 건물들로 필터링하기

추가하려면, 알맞은 버튼을 클릭하십시오. 관계 연산자 (=, > 등등) 와 논리 연산자 (AND, OR 등등) 를 사용할 수 있습니다.

*Test* 버튼을 클릭하면 사용자 쿼리의 문법을 점검합니다. *Clear* 버튼은 쿼리를 지우고 레이어를 원래 상태로 (예를 들면 레이어의 모든 포인트를 다시 불러와) 되돌립니다. 쿼리를 `.QQF` 파일로 *Save…*저장할 수도 있고, 파일에서 쿼리를 대화창으로 *Load…*불러올 수도 있습니다.

필터를 적용하면, QGIS 는 쿼리로 선택된 하위 집합이 마치 완전한 레이어인 것처럼 취급합니다. 예를 들면 사용자가앞의 필터 를 건물을 필터링하기 위해 적용한 경우, 식생 분류를 표시하거나 쿼리하거나 저장하거나 편집할 수 없습니다. 식생 분류는 이 하위 집합에 속하지 않기 때문입니다.

> **팁**
>
> **레이어 패널은 필터링된 레이어를 표시합니다**
>
> *Layers* 패널에서 필터링된 레이어에 마우스를 가져가면 레이어 옆에 쿼리가 사용됐다는 사실을 알려주는 Filter 아이콘이 나타납니다. 이 아이콘을 더블 클릭하면 *Query Builder* 대화창이 열려 쿼리를 편집할 수 있습니다. 쿼리 작성기 대화창은 *Layer* → *Filter…*메뉴를 통해서 열 수도 있습니다.

### 16.2.3 심볼 속성

*Symbology* 탭에서 점구름을 어떻게 렌더링할지 설정합니다. 이 탭의 윗부분에서 다양한 피처 렌더링 작업자를 설정할 수 있습니다. 아랫부분에서는 모든 피처 렌더링 작업자에 적용되는, 레이어 전체에 대한 일반 설정을 할 수 있습니다.

#### 피처 렌더링 유형

*Symbology* 탭 최상단에 있는 드롭다운 메뉴를 통해 점구름 렌더링을 위한 다양한 옵션들을 선택할 수 있습니다 (그림 16.4 참조):

- *Extent Only*: 데이터 범위의 경계 상자만 표시합니다. 이 옵션은 데이터의 범위를 살펴보는 데 유용합니다. 다른 경우들과 마찬가지로, *Symbol* 위젯 을 통해 사용자가 경계 상자에 적용하고 싶은 모든 속성을 (색상, 획, 불투명도, 하위 레이어 등등을) 환경설정할 수 있습니다.
- *Attribute by Ramp*: 데이터를 색상 그레이디언트로 렌더링합니다. 색상표 렌더링 작업자의 속성 을 참조하세요.
- *RGB*: 적색, 녹색 및 청색 색상값을 이용해서 데이터를 렌더링합니다. *RGB* 렌더링 작업자 를 참조하세요.
- *Classification*: 서로 다른 범주에 대해 서로 다른 색상을 이용해서 데이터를 렌더링합니다. 범주화 렌더링 작업자 를 참조하세요.

점구름을 불러올 때, QGIS 가 다음과 같은 로직을 따라 최적의 렌더링 작업자를 선택합니다:

- 데이터셋이 색상 정보 (적색, 녹색, 청색 속성) 을 담고 있는 경우 RGB 렌더링 작업자를 사용할 것입니다.
- 데이터셋이 `Classification` 속성을 담고 있는 경우 범주화 렌더링 작업자를 사용할 것입니다.
- 이 두 경우가 아니라면 Z 속성 기반 렌더링으로 돌아갈 것입니다.

사용자가 점구름의 속성을 모르는 경우, *Statistics* 탭 에서 점구름이 담고 있는 속성 그리고 값들이 분포된 범위를 표시하는 훌륭한 개요를 볼 수 있습니다.

For each renderer, you can improve the data display adjusting the *point symbol size* or enabling *surface triangulation*.

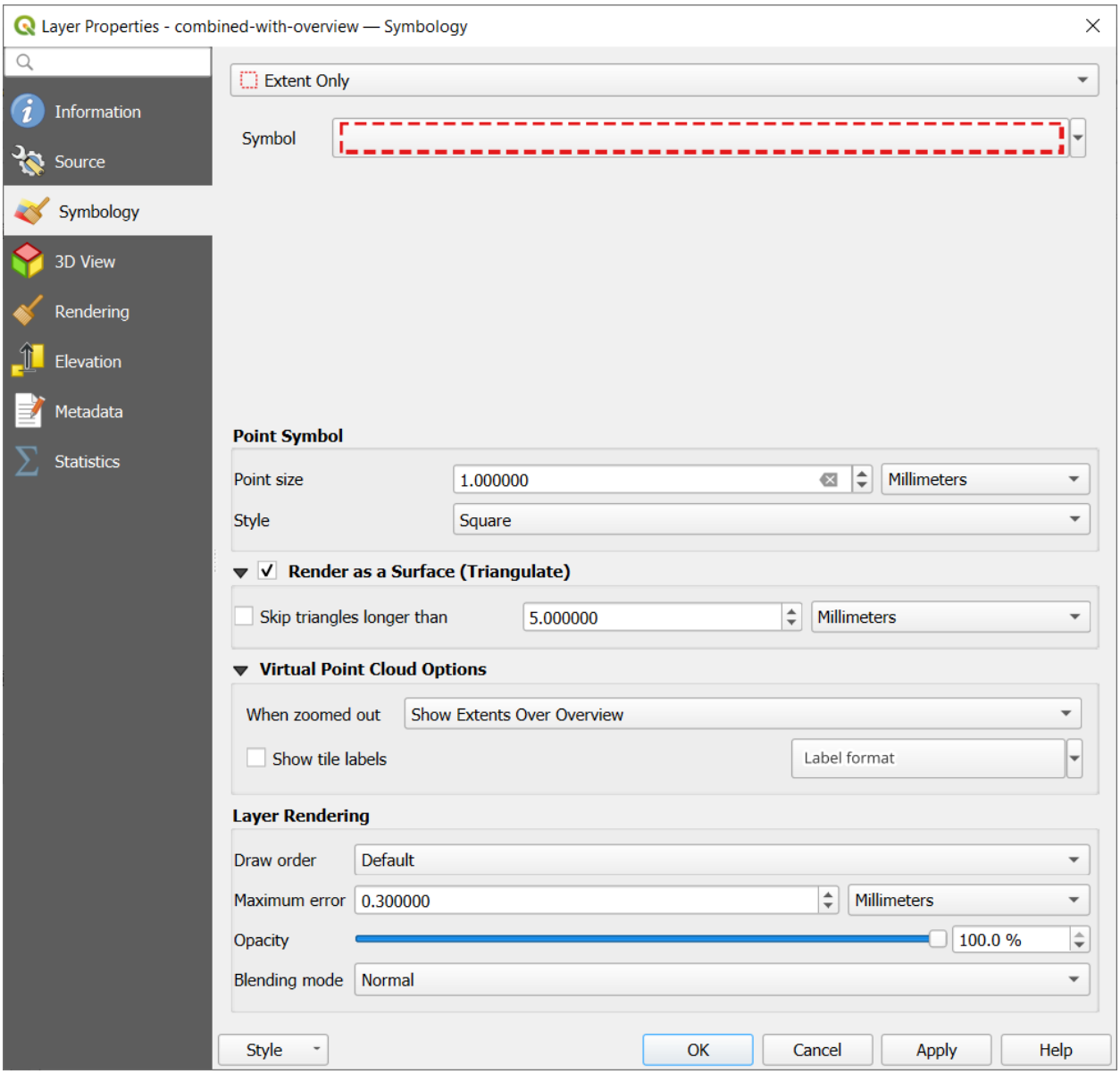



그림 16.4: 점구름 심볼 탭

### 색상표 렌더링 작업자의 속성

*Attribute by Ramp* 를 이용하면 데이터를 색상 그레이디언트 위로 숫자값에 따라 표시할 수 있습니다. 이런 숫자값은 이를테면 기존 강도 속성 또는 Z 값 등이 될 수 있습니다. 최소값 및 최대값에 따라, 다른 값들을 보간해서 색상 그레이디언트에 분포시킵니다. 개별 값 및 개별 값을 특정 색상에 할당한 것을"색상표 (color map)"라고 하며, 색상표는 표로 표시됩니다. 다음 그림 아래 다양한 설정 옵션을 설명합니다.

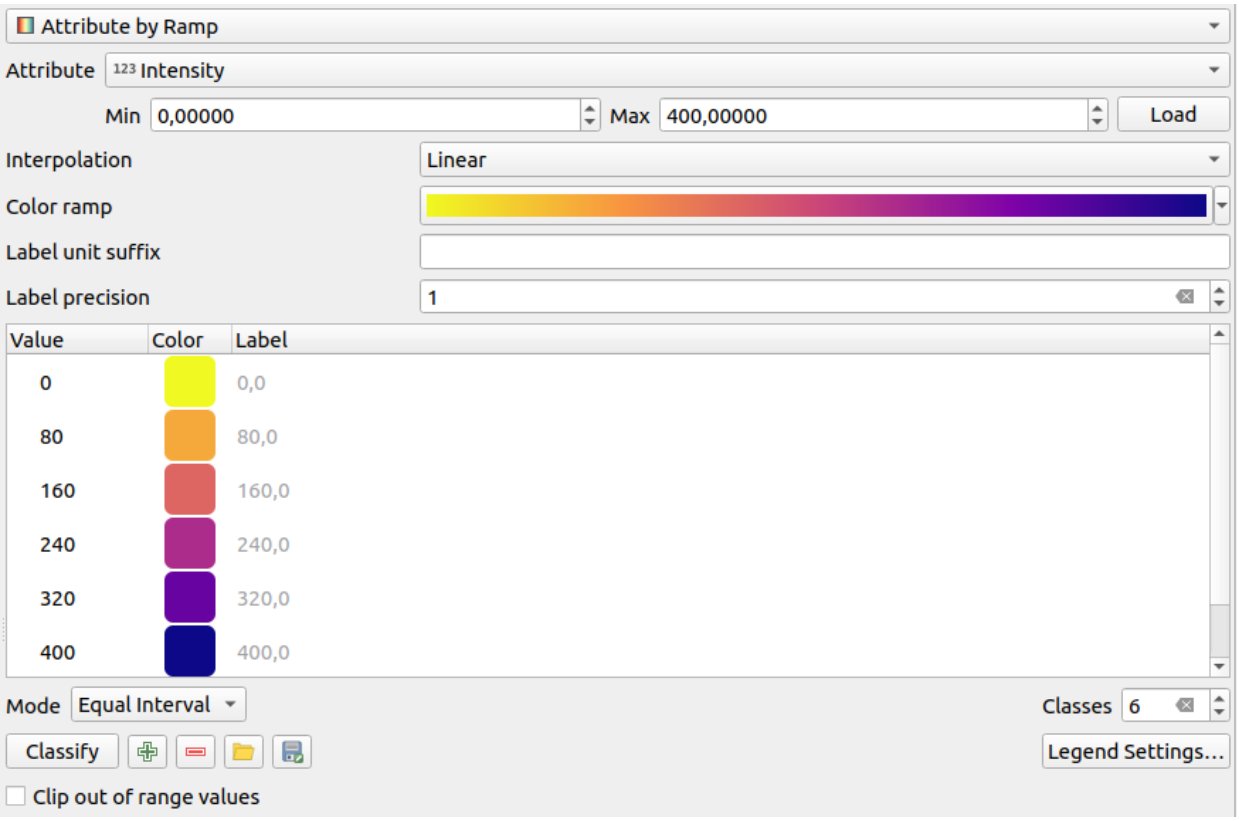



그림 16.5: 점구름 심볼 탭: 색상표의 속성

- *Min* 및 *Max* 란에서 색상표에 적용되는 범위를 정의합니다: *Min* 값은 색상표의 왼쪽 끝, *Max* 값은 오른쪽 끝을 나타내고, 그 사이의 값들은 보간됩니다. QGIS 는 기본적으로 선택한 속성에서 최소값과 최대값을 탐지하지만 이 두 값을 수정할 수도 있습니다. 값을 변경한 뒤에라도 *Load* 버튼을 누르면 기본값을 복원할 수 있습니다.
- *Interpolation* 항목이 색상에 값을 어떻게 할당할지를 정의합니다:
  - *Discrete* (*Value* 열 헤더에 <= 심볼이 나타납니다): 동등하거나 더 높은 값을 가진, 가장 가까운 색상표 항목에서 색상을 선택합니다.
  - *Linear*: 픽셀 값보다 높거나 낮은 색상표 항목에서 색상을 선형적으로 보간합니다. 다시 말해 데이터셋의 각 값에 유일 색상을 대응시킨다는 뜻입니다.
  - *Exact* (*Value* 열의 헤더에 = 심볼이 나타납니다): 색상표 항목과 동등한 값을 가진 픽셀에만 색상을 적용합니다. 그렇지 않은 픽셀은 렌더링하지 않습니다.
- *Color ramp* 위젯에서 데이터셋에 할당할 색상표를 선택할 수 있습니다. 이 위젯 에서와 같이, 새 색상표를 생성하거나 현재 선택한 색상표를 편집하고 저장할 수 있습니다.
- *Label unit suffix* 란에서 범례에 있는 값 뒤에 라벨을 추가하고, *Label precision* 란에서는 표시할 소수점 이하 자릿수를 제어합니다.

범주화 *Mode* 메뉴를 통해 범주 전체에 걸쳐 값을 어떻게 분포시킬지 정의할 수 있습니다:

- *Continuous*: 색상표의 색상 지점으로부터 범주 개수 및 색상을 가져옵니다. 색상표의 색상 지점 분포를 따라 경계 값을 설정합니다. (색상 지점에 대해 더 자세히 알고 싶다면색상표 설정 을 참조하세요.)
- *Equal interval*: 이 줄 끝에 있는 *Classes* 란에서 범주의 개수를 지정하면, 범주들이 모두 동일한 크기를 가지도록 경계 값 (limits value) 들을 정의합니다.

범주들을 자동으로 결정하고 색상표에 표시합니다. 그러나 이런 범주들을 직접 편집할 수도 있습니다:

- *Value*: 표에서 이 열을 더블클릭하면 범주 값을 수정할 수 있습니다.
- *Color*: 이 열을 더블클릭하면색상 선택기 위젯이 열리고, 해당 값에 적용할 색상을 선택할 수 있습니다.
- *Label*: 이 열을 더블클릭하면 범주 라벨을 수정할 수 있습니다.
- 색상표에서 선택한 행들을 오른쪽 클릭하면 해당 선택 집합에 대해 색상 (*Change Color*…) 및 불투명도 (*Change Opacity*…) 를 변경할 수 있습니다.

표 아래에 있는 *Classify* 버튼으로 기본 범주들을 복원하거나, Add 아이콘으로 값을 직접 추가하거나, Delete 아이콘으로 표에서 선택한 값을 삭제할 수 있습니다.

사용자 지정 색상표가 매우 복잡할 수도 있기 때문에, Load 아이콘으로 기존 색상표를 불러오거나 Save 아이콘으로 색상표를 다른 레이어에 사용하기 위해 (`txt` 파일로) 저장할 수 있습니다.

*Interpolation* 드롭다운 메뉴에서 *Linear* 를 선택한 경우, 다음도 환경 설정할 수 있습니다:

- *Clip out of range values*: 선형 메소드는 기본적으로 설정된 *Min* 값 미만의 (또는 설정된 *Max* 값을 초과하는) 데이터셋 값에 첫 번째 범주 (또는 마지막 범주) 색상을 할당합니다. 이 범위 밖의 값들을 렌더링하지 않고자 하는 경우 이 옵션을 체크하십시오.
- *Legend settings*: *Layers* 패널의 범례 및조판 범례 를 어떻게 표시할지 설정합니다. 사용자 지정 방식은 래스터 레이어의 사용자 지정 방식과 동일합니다. (더 자세한 내용은래스터 범례 사용자 지정하기 를 참조하세요.)

### RGB 렌더링 작업자

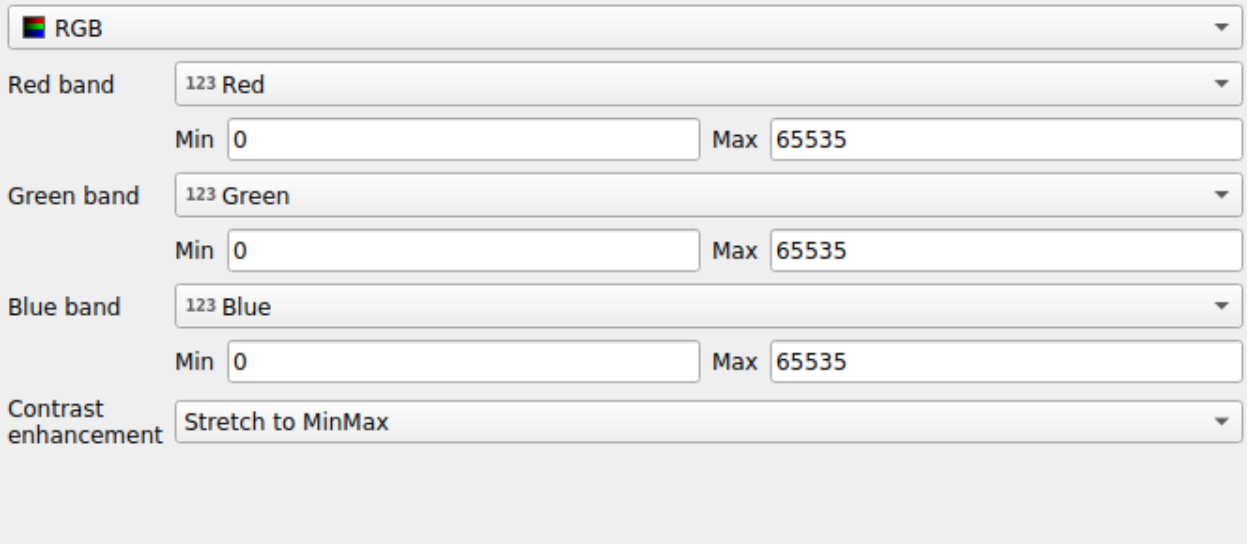



그림 16.6: 점구름 RGB 렌더링 작업자

*RGB* 렌더링 작업자를 이용하면, 점구름에서 선택한 속성 3 개를 적색, 녹색, 그리고 청색 요소로 사용할 것입니다. 속성들이 제대로 명명되어 있다면, QGIS 가 자동적으로 이 속성들을 선택해서 각 밴드의 *Min* 및 *Max* 값을 불러오고 이에 맞춰 색상 배율을 조정합니다. 하지만 이 값들을 직접 수정할 수도 있습니다.

값에 적용할 수 있는 *Contrast enhancement* 방법을 *No Enhancement*, *Stretch to MinMax*, *Stretch and Clip to MinMax* 그리고 *Clip to MinMax* 가운데 선택할 수 있습니다.

**참고**

*Contrast enhancement* 도구는 아직 개발 중입니다. 이 도구에 문제가 발생하는 경우 기본 설정인 *Stretch to MinMax* 를 사용해야 할 것입니다.

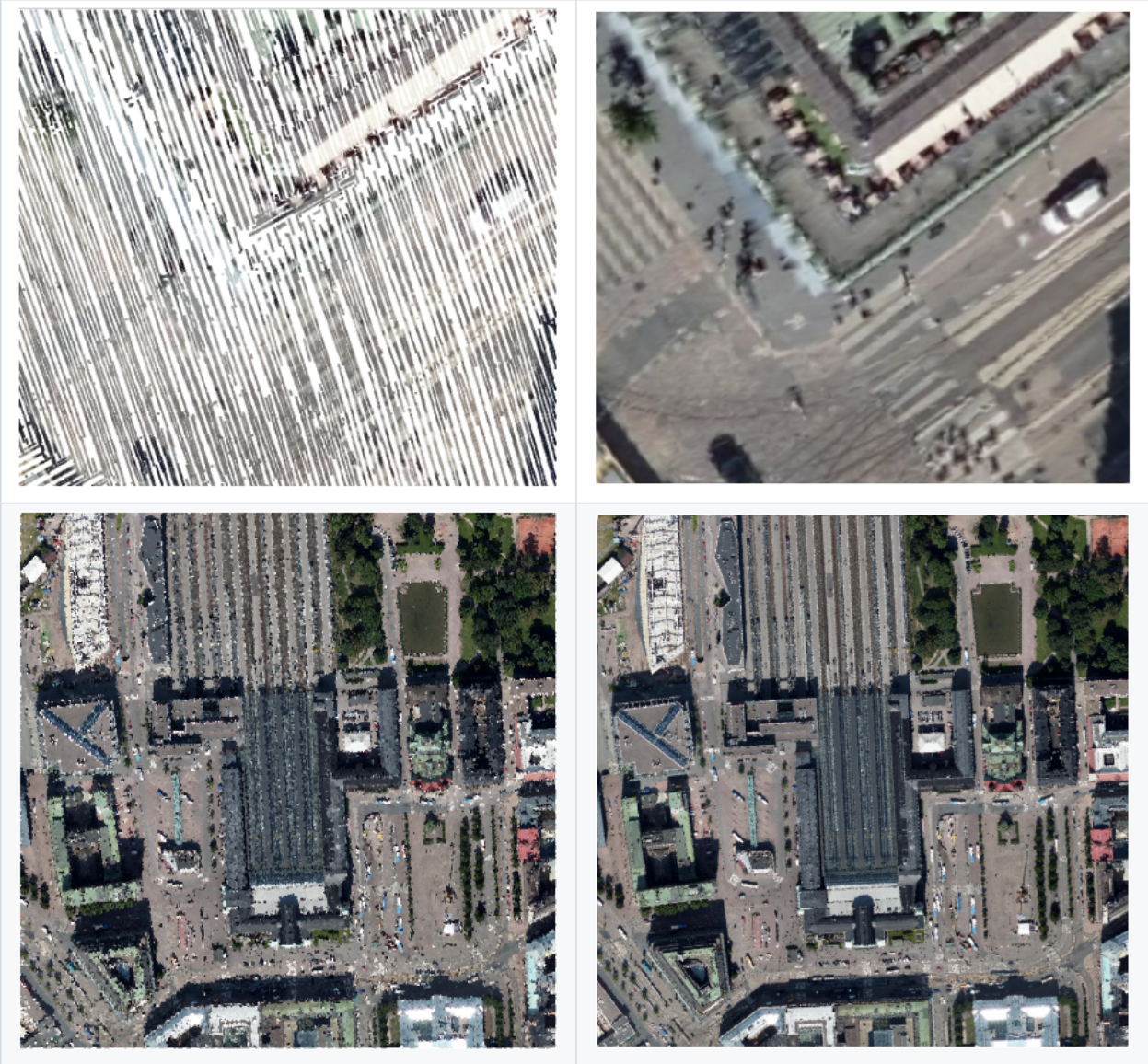


그림 16.7: Example of RGB renderer (left) combined with surface triangulation option (right)

### 범주화 렌더링 작업자

*Classification* 렌더링에서는 점구름을 속성 기반 색상으로 구별해서 표시합니다. 모든 유형의 (숫자값, 문자열, …) 속성을 이용할 수 있습니다. 점구름 데이터가 `Classification` 이라는 필드를 포함하고 있는 경우가 많습니다. 이 필드는 일반적으로, 식생 (□□) 과 같은, 후처리 과정에서 자동적으로 결정된 데이터를 포함하고 있습니다. *Attribute* 드롭다운 메뉴에서, 범주화에 사용할 필드를 속성 테이블로부터 선택할 수 있습니다. QGIS 는 기본적으로 LAS 사양의 정의를 이용합니다. (ASPRS 홈페이지 에 있는 PDF 에서'ASPRS 표준 포인트 범주'표를 참조하세요.) 하지만 데이터가 이 스키마를 비켜 갈 수도 있습니다. 의심이 든다면 데이터를 받은 인물 또는 기관에 범주의 정의를 문의해야 할 것입니다.

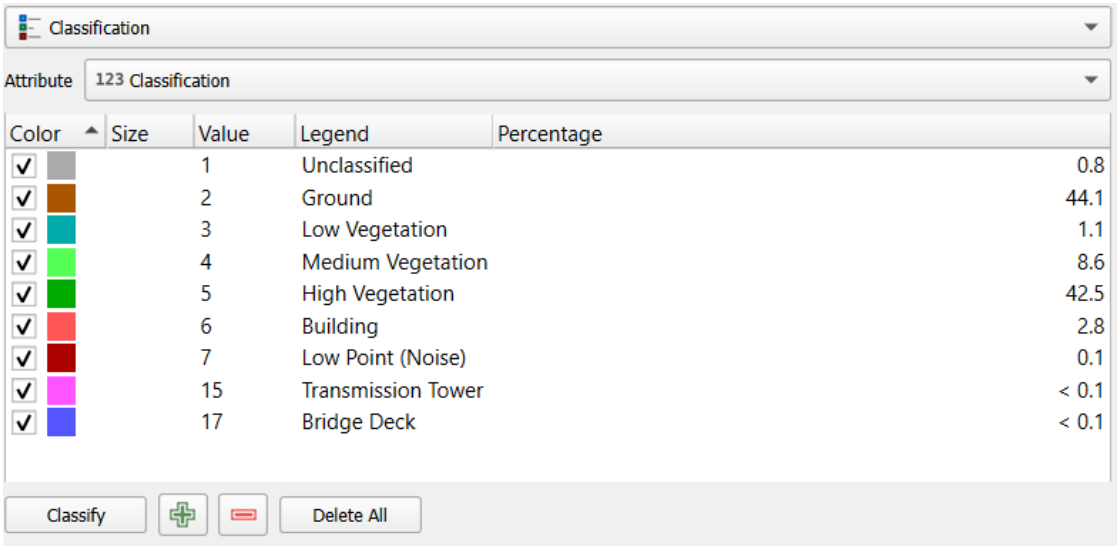



그림 16.8: 점구름 범주화 렌더링 작업자

In the table all used values are displayed with the corresponding color and legend. At the beginning of each row there is a check box; if it is unchecked, this value is no longer shown on the map. With double click in the table, you can modify:

- the *Color*: the 색상 선택기 widget opens
- the *Size*: assigning a size of `0` to a category will revert it to use the default *point size* set for the layer
- the *Value*
- the *Legend*
- the *Percentage* column will show you the representation of the category within the layer.

You can also right-click one or more rows to open a context menu with the options to change the color, opacity or point size.

테이블 아래에 있는 버튼으로 QGIS 가 생성한 기본 범주를 변경할 수 있습니다:

- *Classify* 버튼을 누르면 데이터를 자동으로 범주화할 수 있습니다: 속성에 나타나지만 아직 표에는 표시되지 않은 모든 값을 추가합니다.
- Add 또는 Delete 아이콘을 클릭하면 값을 직접 추가 또는 제거할 수 있습니다.
- *Delete All* 버튼을 누르면 표에서 모든 값을 제거합니다.

> **힌트**
>
> *Layers* 패널에서 레이어의 클래스 지엽 (leaf) 항목을 오른쪽 클릭하면 대응하는 피처들의 가시성을 빠르게 환경설정할 수 있습니다.

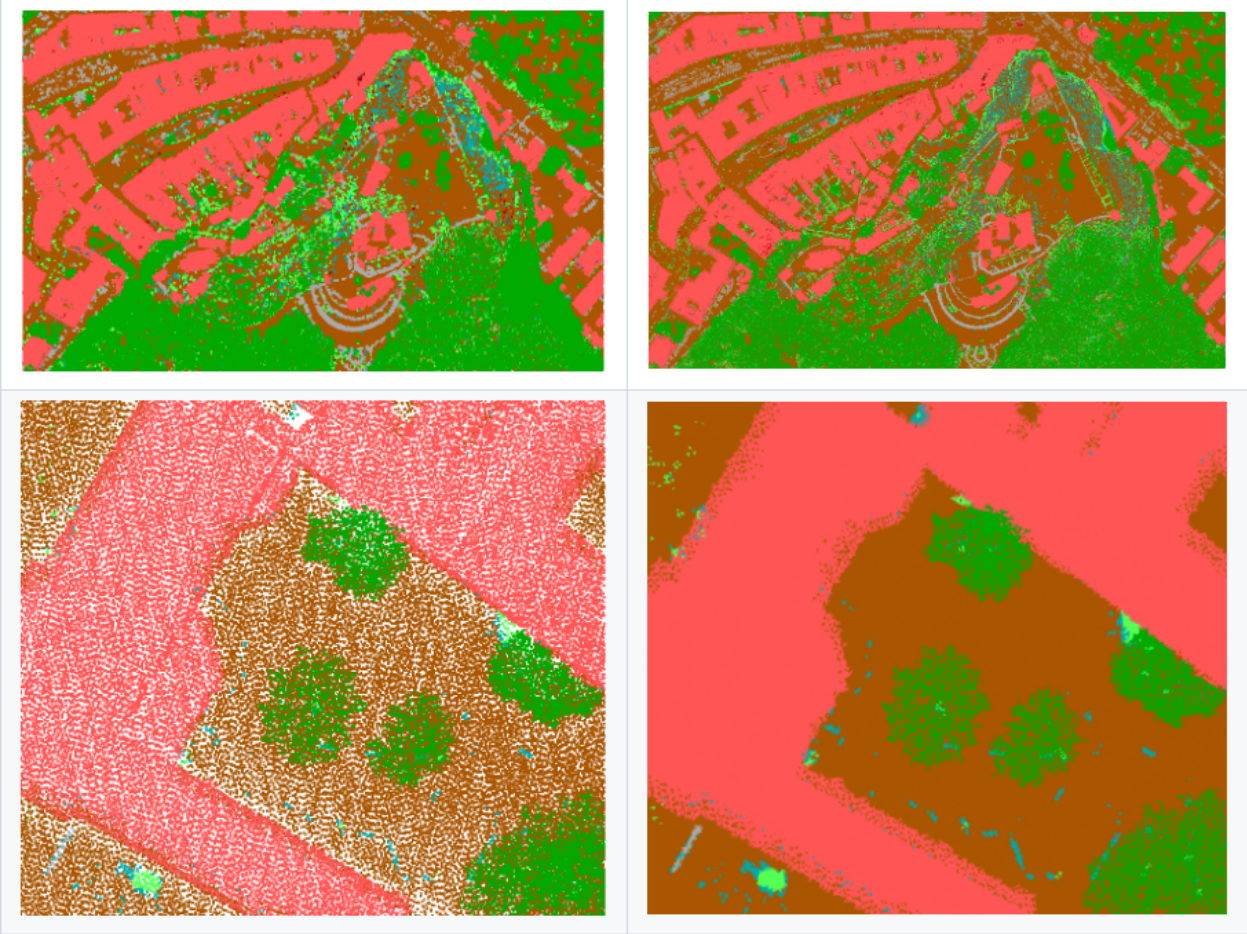


그림 16.9: Example of classification renderer (left) combined with surface triangulation option (right)

### 포인트 심볼

*Point Symbol* 프레임에서 각 데이터 포인트를 표시할 크기 및 단위 (예: 밀리미터, 픽셀, 인치 등등) 를 설정할 수 있습니다. 포인트 스타일도 *Circle* 또는 *Square* 가운데 하나로 선택할 수 있습니다.

### Render as a surface (Triangulate)

Check *Render as surface (Triangulate)* to enable the triangulation of the point cloud layer in the 2D view. This option allows rendering triangles instead of points. Each point keeps its color for interpolation in the triangle. You can control the horizontal length of computed triangles:

By checking the *Skip triangles longer than* option and setting up the threshold value, you can control the maximum lenght of a side of the triangles to consider in the horizontal plan. This can be particularly useful if you want to identify actual holes in the data.

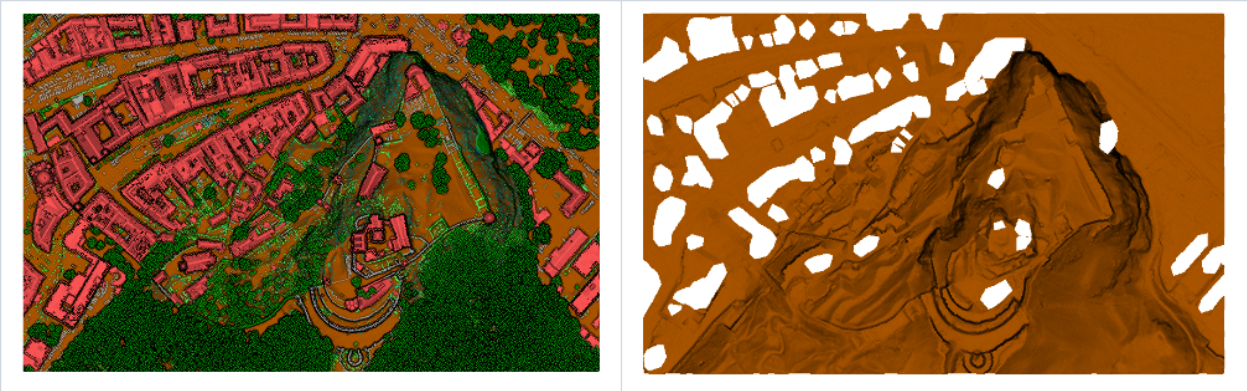


그림 16.10: Rendering data as a surface with map shading (left) and with map shading, filtering large triangles (right)

### Virtual Point Cloud Options

The *Virtual Point Cloud Options* are available only when the layer is a *virtual point cloud (VPC)*. QGIS renders the overview of the VPC when zoomed out, if an overview is present. This provides a seamless transition from the overview to the detailed display of individual point clouds as you zoom in.

You can control how the VPC is displayed when zoomed out using the available options:

- *Show Extents Only*: Only the extents of the underlying point clouds are displayed.
- *Show Overview Only*: Only the overview is displayed (if available).
- *Show Extents Over Overview*: The extents are displayed on top of the overview.

You can also choose to *Show tile labels* to display the tile names and set the label format.

#### 레이어 렌더링

*Layer Rendering* 프레임에서는 레이어 렌더링을 수정할 수 있는 다음 옵션들을 설정할 수 있습니다:

- *Draw order*: 2 차원 맵 캔버스 위에 점구름을 렌더링하는 순서가 점구름의 Z 값을 따라야 할지를 제어할 수 있습니다. 다음과 같은 순서로 렌더링할 수 있습니다:
  - *Default*: 포인트가 레이어에 저장되어 있는 순서대로 렌더링합니다.
  - *Bottom to top*: 하단에서 상단으로 (실제 정사영상으로 보이기 위해, 높은 Z 값을 가진 포인트가 낮은 Z 값을 가진 포인트를 가리도록) 렌더링합니다.
  - 또는 *Top to bottom*: 아래에서 위를 보는 장면이 나타나도록 렌더링합니다.

- *Maximum error*: 점구름은 보통 화면에 표시하기 위해 필요한 개수보다 훨씬 많은 포인트들을 담고 있습니다. 이 옵션을 통해 점구름을 얼마나 진하게 또는 연하게 표시할지를 설정합니다. ('포인트 사이의 최대 허용 간격'으로 이해해도 됩니다.) 큰 숫자를 (예: 5mm) 설정하면 포인트 사이의 틈이 보이게 될 것입니다. 작은 숫자를 (예: 0.1mm) 설정하면 표시에 필요하지 않은 포인트를 대량으로 강제 렌더링하게 되어 속도가 느려질 수 있습니다.
- *Opacity*: 이 도구를 통해 맵 캔버스에서 아래에 있는 레이어를 가시화할 수 있습니다. 슬라이드 바를 통해 사용자 레이어의 가시성을 필요에 따라 조정하십시오. 슬라이드 바 옆에 있는 메뉴에서 가시성을 정확한 백분율로 설정할 수도 있습니다.
- *Blending mode*: 이 도구를 통해 특별한 렌더링 효과를 줄 수 있습니다. 위 그리고 아래에 있는 레이어의 픽셀을혼합 모드 에서 설명한 설정으로 혼합합니다.
- *Eye dome lighting*: 아이돔 (Eye Dome) 조명은 보다 나은 심도 렌더링을 위해 맵 캔버스에 음영 효과를 적용합니다. 렌더링 품질은그리기 순서 속성에 따라 달라집니다. 이 순서를 *Default* 로 설정하면 최적의 결과를 얻지 못 할 수도 있습니다. 다음 파라미터들을 제어할 수 있습니다:
  - *Strength*: 심도를 보다 더 인식할 수 있도록 대조를 증가시킵니다.
  - *Distance*: 사용된 픽셀이 중심 픽셀로부터 벗어난 거리를 나타내어 경계를 더 진하게 보이도록 합니다.

### 16.2.4 3 차원 뷰 속성


*3D View* 탭에서는 3 차원 맵에 점구름을 렌더링하기 위한 설정을 할 수 있습니다.

#### 3 차원 렌더링 모드

탭 최상단에 있는 드롭다운 메뉴에서 다음과 같은 옵션들을 선택할 수 있습니다:

- *No Rendering*: 데이터를 표시하지 않습니다.
- *Follow 2D Symbology*: 3 차원 피처 렌더링을*2* 차원 할당 심볼 과 동기화합니다.
-  *Single Color*: 속성에 상관없이 모든 포인트를 동일한색상 으로 표시합니다.
-  *Attribute by Ramp*: 지정한 속성을 색상표로 보간한 다음 각각 일치하는 색상을 피처에 할당합니다. 색상표 렌더링 작업자의 속성 을 참조하세요.
-  *RGB*: 피처의 서로 다른 속성들을 이용해서 피처에 할당할 적, 녹, 청 색상 구성 요소를 설정합니다. *RGB* 렌더링 작업자 를 참조하세요.
-  *Classification*: 속성을 기반으로 포인트들을 색상에 따라 구분합니다. 범주화 렌더링 작업자 를 참조하세요.

#### 3 차원 포인트 심볼


*3D View* 탭의 아랫부분에 *Point Symbol* 프레임이 있습니다. 여기에서 모든 렌더링 작업자에 동일하게 적용되는, 레이어 전체에 대한 일반 설정을 할 수 있습니다. 다음과 같은 옵션을 설정할 수 있습니다:

- *Point size*: 각 데이터 포인트를 어느 정도의 (픽셀 단위) 크기로 표시할지 설정할 수 있습니다.
- *Maximum screen space error*: 이 옵션을 통해 점구름을 얼마나 진하게 또는 연하게 표시할지를 (픽셀 단위로) 설정합니다. 큰 숫자를 (예: 10) 설정하면 포인트 사이의 틈이 보이게 될 것입니다. 작은 숫자를 (예: 0) 설정하면 표시에 필요하지 않은 포인트를 대량으로 강제 렌더링하게 되어 속도가 느려질 수 있습니다. (더 자세한 내용은 *Symbology* 최대 오류 를 찾아보세요.)

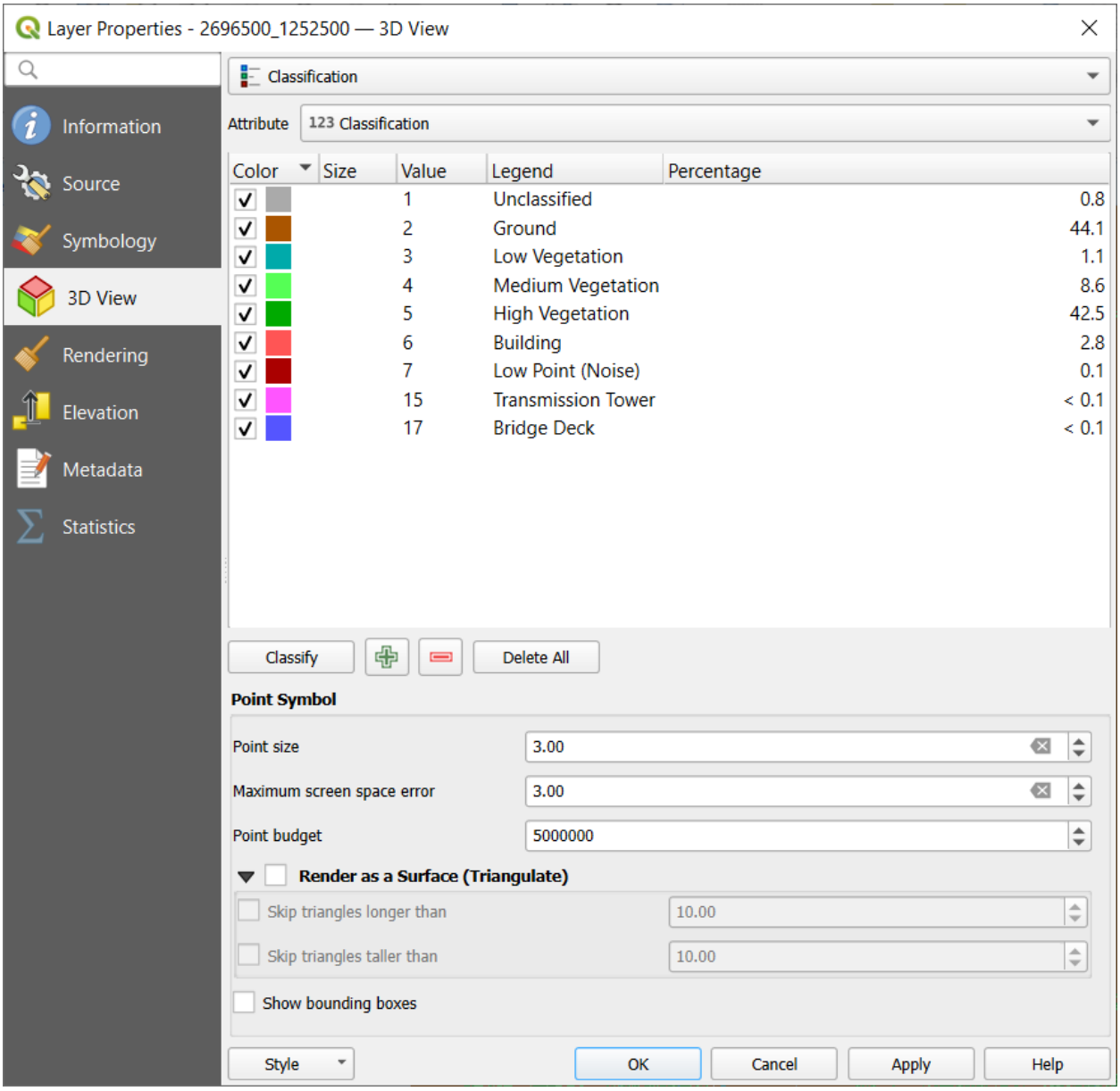



그림 16.11: 범주화 렌더링 작업자를 선택한 점구름 3 차원 뷰 탭

- *Point budget*: 렌더링이 너무 오래 걸리는 것을 피하려면, 렌더링할 최대 포인트 개수를 설정하면 됩니다.
- 3 차원에 점구름 레이어를 삼각분할로 계산한 단단한 표면 (solid surface) 을 이용해서 렌더링하려면 *Render as surface (Triangulate)* 를 체크하십시오. 계산된 삼각형들의 차원을 제어하려면:
  - *Skip triangles longer than*: 한계값을 초과하는 삼각형을 건너뜁니다. 이 한계값은 수평면에서 설정되며, 삼각형의 변들 가운데 가장 긴 길이에 대응합니다.
  - *Skip triangles taller than*: 한계값을 초과하는 삼각형을 건너뜁니다. 이 한계값은 수직면에서 설정되며, 삼각형의 변들 가운데 가장 긴 높이에 대응합니다.
- *Show bounding boxes*: 디버그 작업에 특히 유용한 옵션으로, 노드들의 경계 상자들을 계층 구조로 표시합니다.

### 16.2.5 렌더링 속성

*Scale dependent visibility* 그룹 박스에서는 *Maximum (inclusive)* 및 *Minimum (exclusive)* 축척을 설정해서 피처가 보이게 될 축척 범위를 정의할 수 있습니다. 이 범위를 벗어나면, 피처를 숨깁니다. Set to current canvas scale 버튼을 클릭하면 현재 맵 캔버스의 축척을 가시성 범위의 한계값으로 설정할 수 있습니다. 자세한 내용은가시성 축척 선택기 을 참조하세요.

> **참고**
>
> *Layers* 패널에서 레이어에 대한 축척 의존 가시성을 활성화시킬 수도 있습니다. 레이어를 오른쪽 클릭한 다음 컨텍스트 메뉴에서 *Set Layer Scale Visibility* 를 선택하십시오.

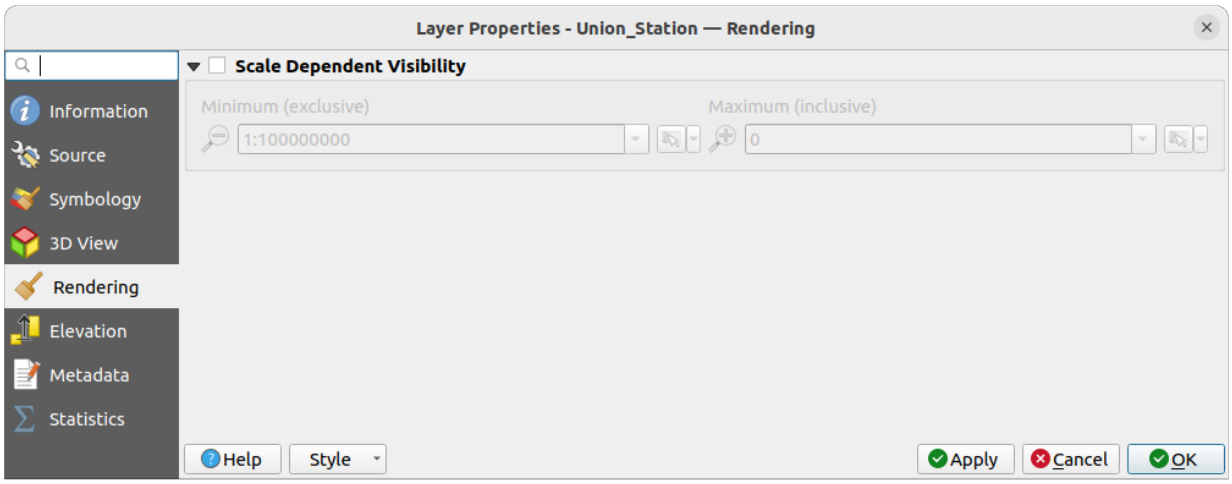



그림 16.12: 점구름 렌더링 탭

### 16.2.6 표고 속성

*Elevation* 탭에서 데이터의 Z 값을 수정할 수 있습니다. 3 차원 맵에서의 데이터의 표고 그리고단면 도구 도표 *(profile tool charts)* 에서의 모양을 조정하기 위해 Z 값을 정정해야 할 수도 있기 때문입니다. 다음 옵션들이 있습니다:

- *Vertical Reference System*: If the CRS of your point cloud layer is a compound one (including a Z dimension), then the vertical CRS used for the layer will be automatically derived from the vertical component of the layer's CRS. In this case, you cannot manually set a different vertical CRS, and the option to change it will be disabled. If your point cloud layer uses a horizontal (2D) CRS (though uncommon), you can manually select a specific vertical CRS by clicking on the Select CRS. Vertical reference systems are supported for point cloud layers in:
  - *Elevation profiles*
  - *Identify Tool Results*
  - *3D map views*
- *Elevation* 부분에서는:
  - *Scale*: 크기 조정 인자를 설정할 수 있습니다. 이 란에 `10` 을 입력하면, Z 값이 `5` 인 포인트를 높이 `50` 으로 표시합니다.
  - *Offset*: 이 란에 Z 수준에 대한 오프셋을 입력할 수 있습니다. 서로 다른 데이터소스들의 높이를 서로 일치시키는 데 유용합니다. 기본적으로, 데이터가 담고 있는 Z 값 가운데 가장 낮은 값을 오프셋 값으로 사용합니다. 이 줄의 끝에 있는 Refresh 버튼을 누르면 이 값을 복원할 수도 있습니다.
- *Profile Chart Accuracy* 부분에서는, 표고 단면에서 포인트들을 얼마나 진하게 또는 옅게 렌더링할 것인지를 *Maximum error* 로 제어할 수 있습니다. 더 높은 값을 설정하면 더 적은 포인트들을 더 빠르게 생성합니다.
- *Profile Chart Appearance* 부분에서는, 포인트 표시 방법을 제어할 수 있습니다:
  - *Point size*: 포인트를 렌더링할 크기를 지원 단위 (밀리미터, 맵 단위, 픽셀 등등) 로 설정할 수 있습니다.
  - *Style*: 포인트를 원 *Circle* 또는 정사각형 *Square* 으로 렌더링할지를 선택할 수 있습니다.
  - *Color*: 단면 뷰에서 가시화된 모든 포인트에 단일 색상을 적용시킬 수 있습니다.
  - *Respect layer's coloring*: 포인트를2 차원 심볼 을 통해 할당된 색상으로 표시하는 대신 레이어 색상 작업을 따르도록 하려면 이 옵션을 체크하십시오.
  - *Apply opacity by distance from curve effect*: 이 옵션을 체크하면 포인트가 단면 곡선에서 멀수록 포인트의 투명도가 올라갑니다.

### 16.2.7 메타데이터 속성

*Metadata* 탭에 사용자 레이어에 대한 메타데이터 리포트를 생성하고 편집할 수 있는 옵션이 있습니다. 더 자세한 정보는메타데이터 를 참조하세요.

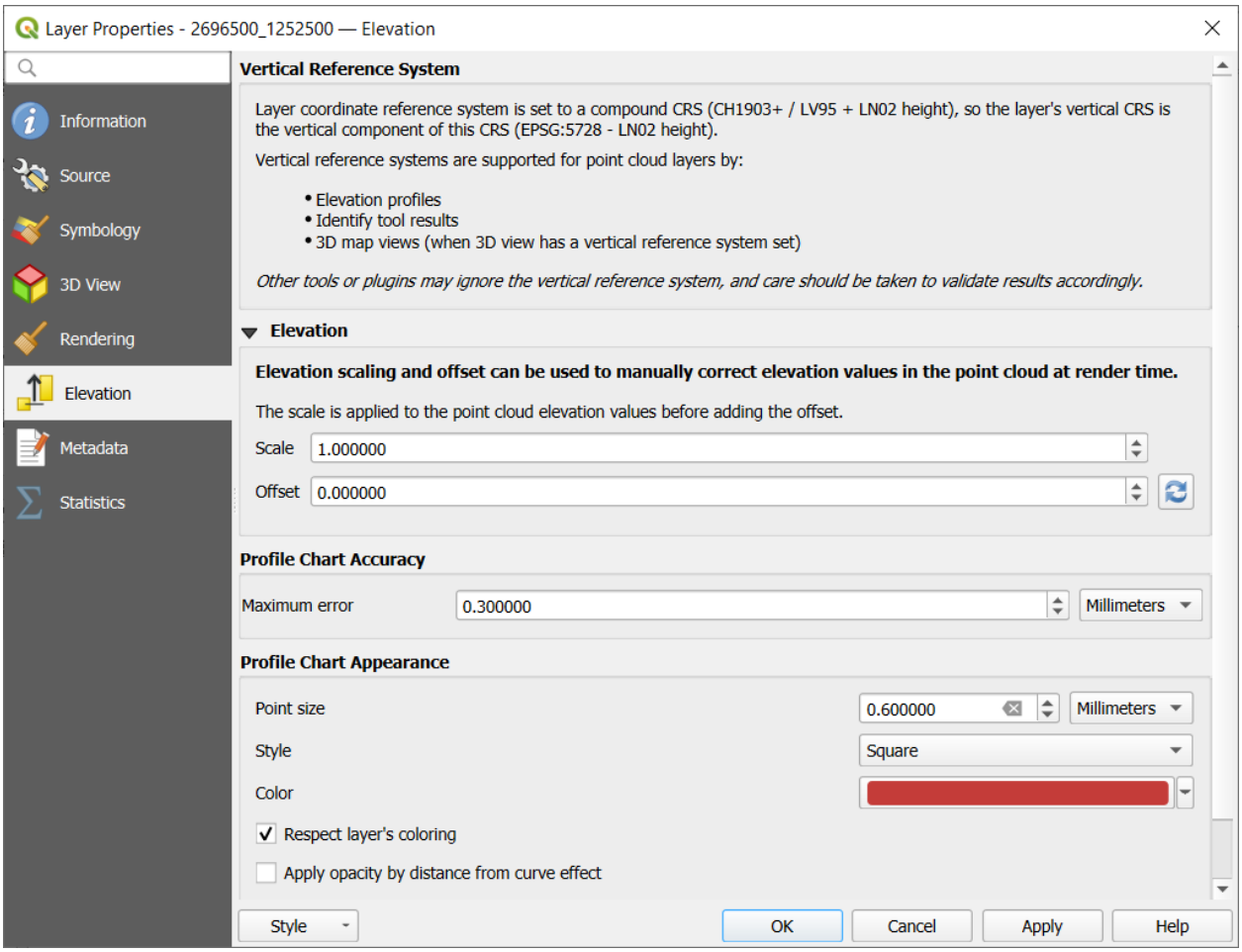



그림 16.13: 점구름 표고 탭

### 16.2.8 통계 속성

Σ *Statistics* 탭에서는 사용자의 점구름 및 그 분포에 대한 속성들의 개요를 볼 수 있습니다.

윗부분에 있는 *Attribute Statistics* 프레임은 점구름이 담고 있는 모든 속성을 목록으로 나열할 뿐만 아니라, *Minimum*, *Maximum*, *Mean*, *Standard Deviation* 같은 몇몇 통계값도 표시합니다.

*Classification* 속성이 존재하는 경우, 아랫부분에 또다른 표가 표시됩니다. 이 표는 해당 속성이 담고 있는 모든 값과 함께 값의 절대 개수 (*Count* ) 및 상대 존재비 (% abundance) 를 목록으로 나열합니다.

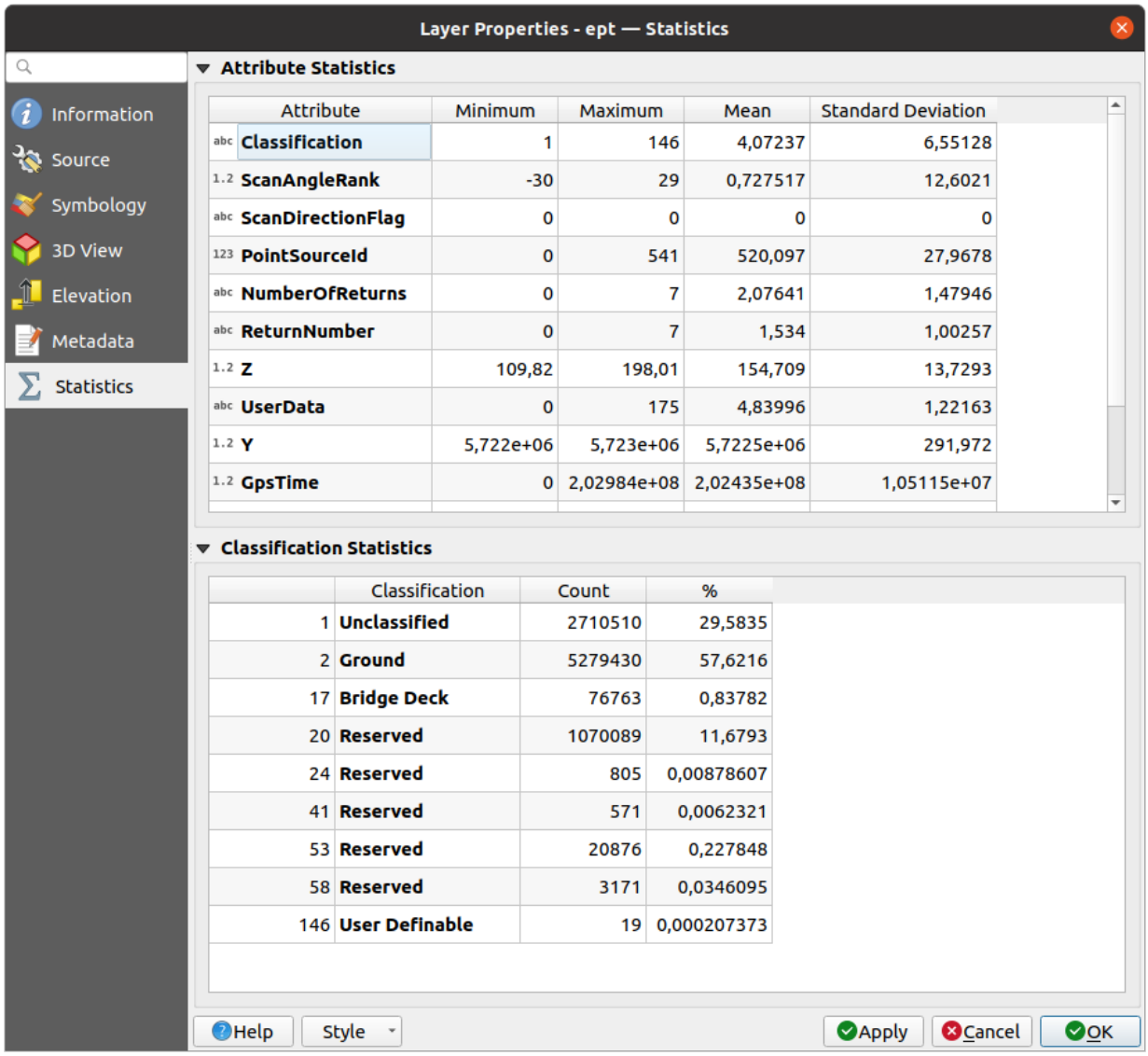



그림 16.14: 점구름 통계 탭

## 16.3 가상 점구름

라이다 (LiDAR) 가 광범위한 영역을 측정할 때, 포인트 수십조 개를 가진 테라바이트 단위의 데이터셋을 생성하는 경우가 많습니다. 이런 대용량 데이터셋을 단일 점구름 파일로 표현하는 경우 저장, 전송, 표시 및 분석이 어렵기 때문에 실용적이지 않습니다. 따라서 점구름 데이터는 일반적으로 정사각형 타일 (예: `1km x 1km`) 로 분할되어 저장 및 배포됩니다. 이때 각 타일은 좀 더 감당 가능한 (예: 압축 시 최대 200MB) 파일 크기가 됩니다.

데이터를 타일화하면 데이터 용량의 문제를 해결할 수 있지만, 단일 타일 안에 완전히 들어오지 않는 관심 지역을 처리하거나 볼 때 문제가 발생합니다. 사용자는 타일 여러 개를 연산하는 워크플로를 개발해야 하며, 산출물에서 원하지 않는 오류를 보고 싶지 않다면 타일 가장자리 가까이 있는 데이터를 처리할 때 특히 주의를 기울여야 합니다. 마찬가지로, 점구름 데이터를 볼 때 파일 여러 개를 불러와 동일한 심볼을 적용시키는 과정이 복잡해지고 느려지게 됩니다.

다음은 QGIS 에 점구름 타일 몇 개를 불러온 예시입니다. 각 타일의 스타일은 타일의 최소/최대 Z 값을 바탕으로 하기 때문에, 타일 가장자리에 눈으로도 볼 수 있는 오류를 발생시킵니다. 즉 각 레이어 별로 스타일 작업을 조정해야 합니다:

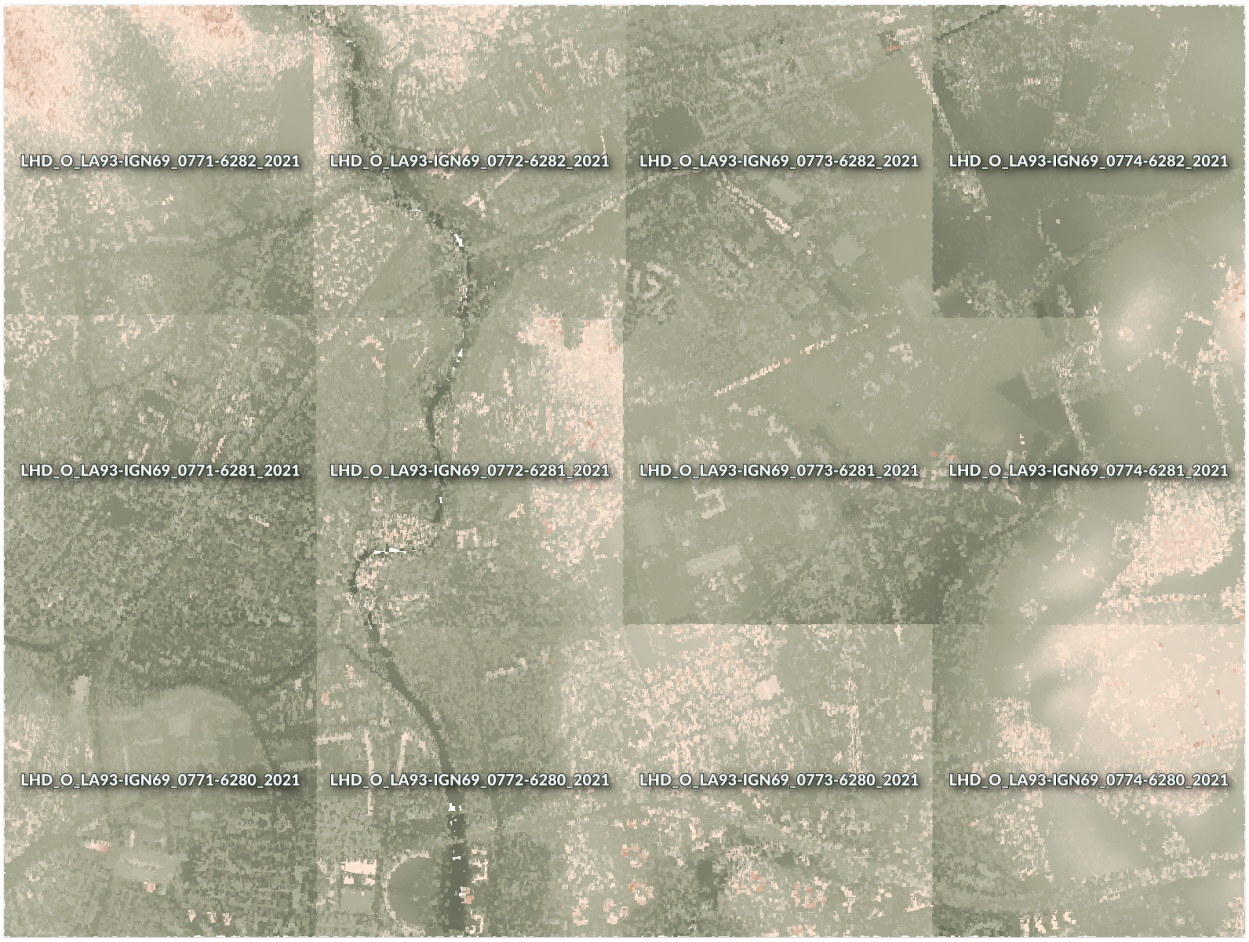



그림 16.15: 가장자리에 오류가 발생한 개별 점구름 타일 불러오기

GIS 업계에서는 가상 래스터라는 개념을 흔히 접할 수 있습니다. 가상 래스터란 실제 데이터를 가진 다른 래스터 파일을 참조하기만 하는 파일을 말합니다. GIS 소프트웨어는 이런 방법을 통해 수많은 파일들을 포함하고 있는 전체 데이터셋을 단일 래스터 레이어로 취급합니다. 가상 파일에 목록화된 모든 래스터를 표시하고 분석하는 편이 훨씬 쉬우니까요.

GDAL 에서 가상 래스터라는 개념을 빌려온 **가상 점구름 (VPC; Virtual Point Cloud)** 은 다른 점구름 파일들을 참조하는 파일 포맷입니다. 가상 점구름을 지원하는 소프트웨어는 타일화 데이터셋 전체를 단일 데이터 소스로 취급합니다.

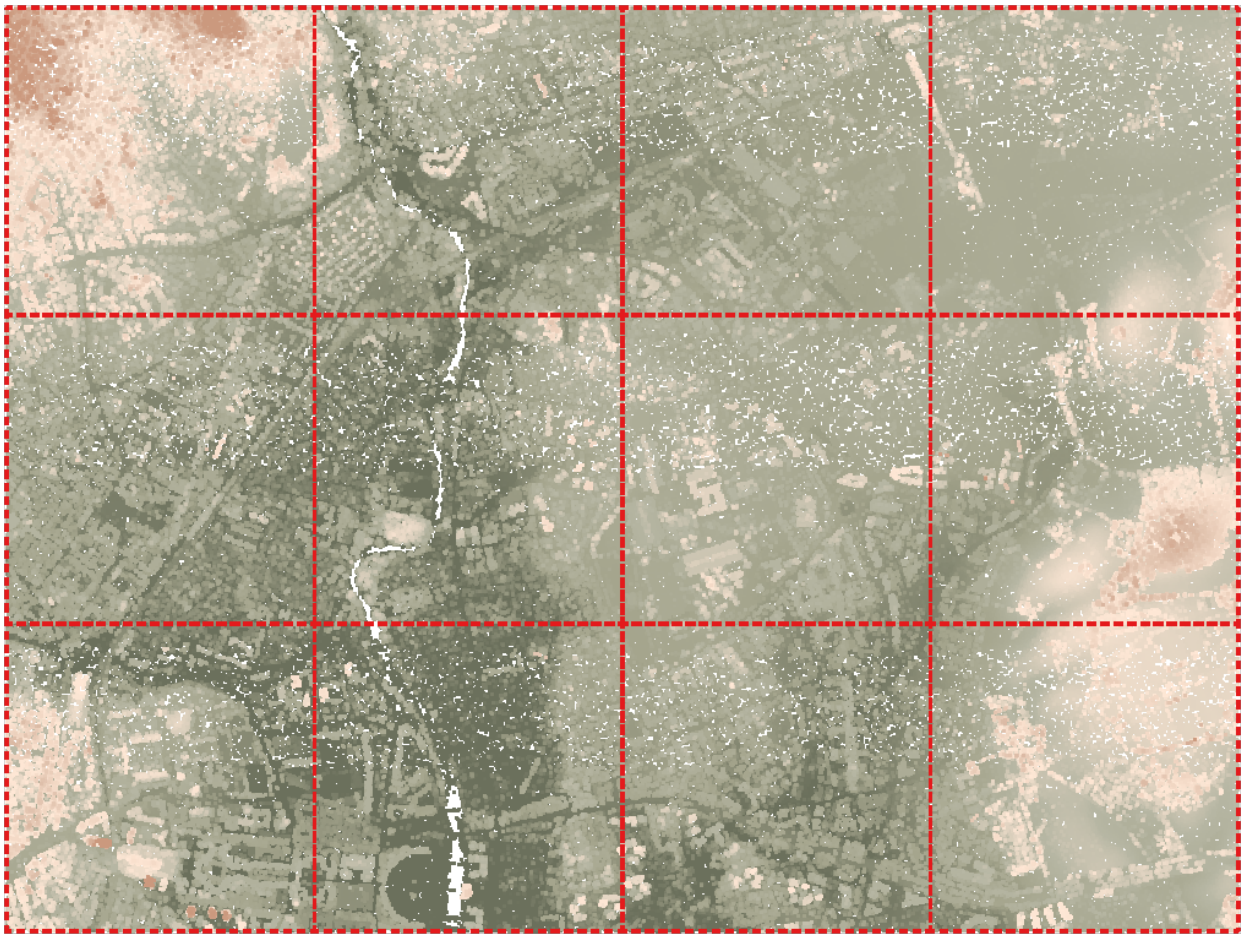


그림 16.16: 가상 점구름

가상 점구름을 표시하고 조작하는 편이 실제 점구름 파일보다 훨씬 유연하고 쉽습니다.

가상 점구름 파일의 핵심은 실제 데이터 파일 (예: `.LAS`, `.LAZ` 또는 `.COPC` 파일) 과 파일로부터 추출한 추가적인 메타데이터를 가리키는 참조를 담고 있는, `.vpc` 확장자를 가진 단순 JSON 파일입니다. 사용자가 직접 입력해서 VPC 파일을 작성할 수 있기는 해도, 다음과 같은 자동화 도구를 사용해서 생성하는 편을 강력하게 권장합니다:

- 공간 처리가상 점구름 *(VPC)* 작성하기 알고리즘
- PDAL 렌치 (wrench) 도구의 `build_vpc` 명령어

더 자세한 내용을 알고 싶다면, 최고의 예제와 (오버뷰 같은) 선택적인 확장 사양도 담고 있는 VPC 사양 을 참조해주십시오.

CHAPTER 17

# 3D 타일 작업

## 17.1 3D 타일이란?

3D tiles are specification for streaming and rendering large-scale 3D geospatial datasets. They use a hierarchical structure to efficiently manage and display 3D content, optimising performance by dynamically loading appropriate levels of detail. This technology is widely used in urban planning, architecture, simulation, gaming, and virtual reality, providing a standardised and interoperable solution for visualising complex geographical data.

See *Using 3D tiled scene services* for instructions on how to add these data sources to QGIS.

### 17.1.1 State of 3D Tiles Support

Currently, QGIS supports two formats of 3D tiles:

- "Cesium"3D Tiles are used primarily for complex 3D models of buildings or whole cities. Such datasets can be provided by cloud-based platforms such as Cesium Ion or Google (*Photorealistic 3D Tiles*).
- Quantized Mesh tiles are used for terrain elevation data.

Supported *3D Tiles* features:

- QGIS currently has partial support for 3D Tiles 1.0 and partial support for 3D Tiles 1.1.
- For 3D Tiles 1.0, the tile format Batched 3D Model (`b3dm`) is supported. The tile formats Instanced 3D Model (`i3dm`), Point Cloud (`pnts`) and Composite (`cmpt`) are not supported yet.
- For 3D Tiles 1.1, the `3DTILES_content_gltf` extension is supported, so tile contents may also be encoded as glTF 2.0 (text or binary). The `EXT_mesh_gpu_instancing` extension is not supported yet.
- Currently only explicit tiling is supported. Implicit tiling is not supported yet
- Currently only tilesets using EPSG:4978 are supported.



그림 17.1: 3D 타일의 예시

## 17.2 3D 타일 속성

3D 타일 *Layer Properties* 대화창은 다음 탭들로 이루어져 있습니다:




### 17.2.1 정보 속성

*Information* 탭은 읽기 전용으로, 현재 레이어에 대한 요약 정보 및 메타데이터를 한 눈에 볼 수 있는 흥미로운 장소입니다. 다음과 같은 정보를 제공합니다:

- 레이어 제공자 기반: 이름, URL, 소스 유형 및 경로, 확대/축소 수준 개수
- custom properties, used to store in the active project additional information about the layer. More properties can be created and managed using PyQGIS, specifically through the setCustomProperty() method.
- 좌표계: 이름, 단위, 메소드, 정확도, 참조 (예: 정적인지 동적인지 여부)
- 채워진 메타데이터 로부터 선택: 접근, 범위, 링크, 연락처, 이력, …

### 17.2.2 소스 속성

*Source* 탭은 선택한 3D 타일에 대한 다음과 같은 기본 정보를 표시합니다:

- *Layer name*: *Layers Panel* 에 표시될 레이어명
- *Coordinate Reference System*: 레이어의좌표계 *(CRS)* 를 표시합니다. 드롭다운 목록에서 최근 사용된 좌표계를 선택하거나 Select CRS 버튼을 (좌표계 선택기 참조) 클릭하면 레이어 좌표계를 변경할 수 있습니다. 레이어 좌표계가 틀렸거나, 지정되지 않은 경우에만 변경하십시오.

### 17.2.3 심볼 속성

레이어 스타일은 기본적으로 텍스처를 이용하지만, 드롭다운 메뉴에서 *Wireframe* 을 선택하면 신 (scene) 뒤에 와이어프레임 메시 (wireframe mesh) 가 보이도록 변경할 수 있습니다. 메시의 채우기 및 라인 심볼을 벡터 폴리곤과 유사하게 변경할 수도 있습니다. *Use texture colors* 를 체크하면 각각의 메시 요소를 전체 텍스처의 평균 색상값으로 렌더링할 것입니다. 대용량 데이터셋을 처리할 때 데이터를 빠르게 훑어보고자 하는 경우 이 옵션을 시도해볼 만합니다.

*Symbology* 탭에서 레이어의 모든 피처에 대해 일괄적으로 적용되는 다음 옵션들을 설정할 수도 있습니다:

- *Maximum error*: 이 파라미터는 3 차원 모델에 표시되는 상세 내용의 수준을 결정합니다. 3D 타일은 점구름과 마찬가지로 시각적 표현에 필요한 양보다 더 많은 정보를 담고 있는 경우가 많습니다. 이 설정을 조정하면 표시 밀도와 렌더링 속도 사이의 밸런스를 제어할 수 있습니다. 값이 클수록 (예: 5mm) 요소 사이사이에 눈에 띌 만한 틈이 생길 수도 있습니다. 반면에 값이 작을수록 (예: 0.1mm) 렌더링해야 할 상세 항목들의 개수가 과도해져 렌더링 과정의 속도가 느려질 가능성이 큽니다. 서로 다른 단위들 가운데 선택할 수 있으므로, 사용자의 특정한 수요에 맞춰 설정을 맞출 수 있습니다.

[1] Also available in the *Layer styling panel*

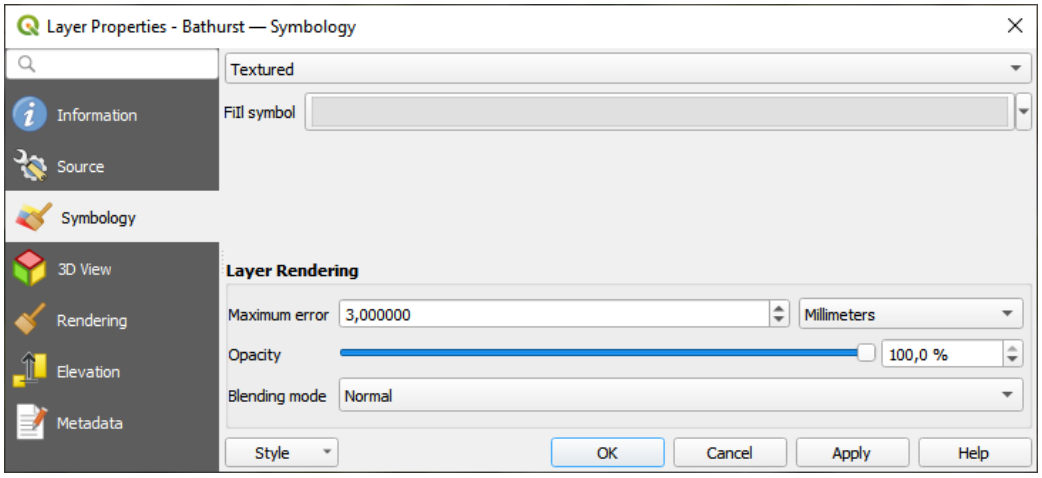



그림 17.2: 3D 타일 레이어 심볼

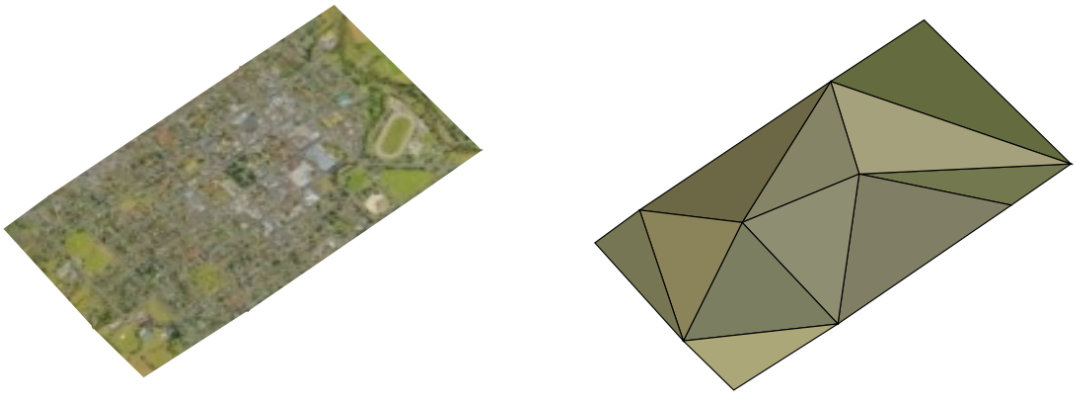


그림 17.3: 3D 타일 — 텍스처와 와이어프레임

- *Opacity*: Adjusts the visibility of the underlying layer on the map canvas using this tool. Use slider to tailor the visibility of your scene layer according to your preferences. Alternatively, specify the exact percentage of visibility through the menu next to the slider.
- *Blending mode*: 이 도구들을 통해 사용자가 그래픽 프로그램에서만 가능하다고 알고 있던 특별한 렌더링 효과를 줄 수 있습니다. 위 그리고 아래에 있는 레이어의 픽셀을혼합 모드 에서 설명한 설정으로 혼합합니다.

### 17.2.4 3 차원 뷰 속성

- *Maximum screen space error*: Determines the threshold for swapping terrain tiles with more detailed ones (and vice versa) - i.e. how soon the 3D view will use higher quality tiles. Lower numbers mean more details in the scene at the expenses of increased rendering complexity.
- *Show bounding boxes*: Shows 3D bounding boxes of the terrain tiles (useful for troubleshooting terrain issues).

데이터를 보려면 *New 3D map view* 를 열면 됩니다.

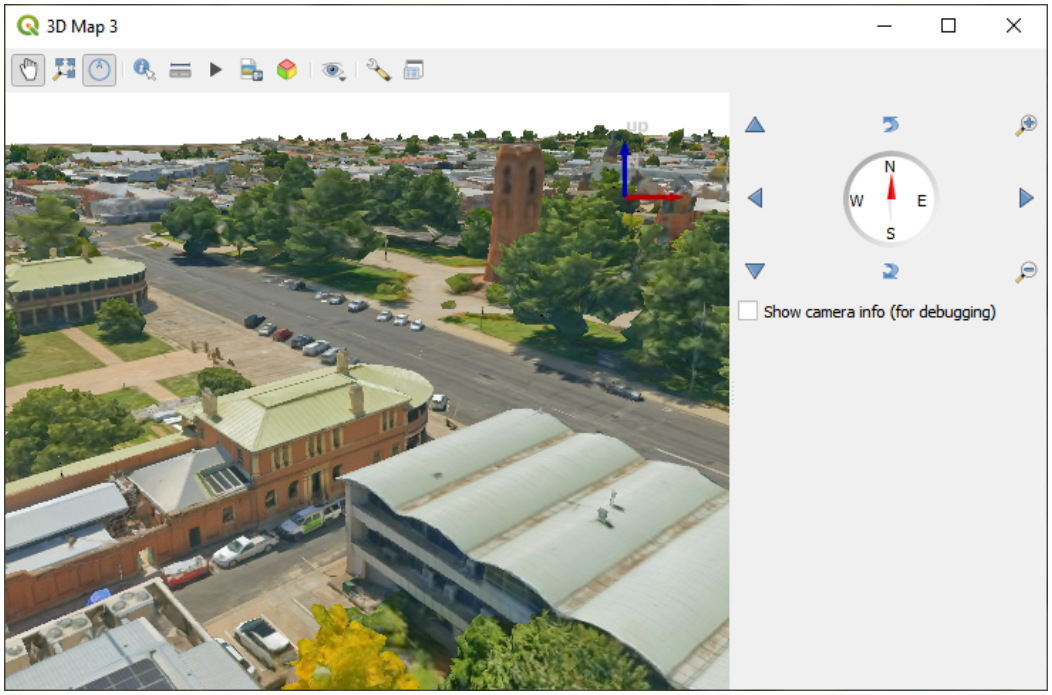



그림 17.4: 3 차원 맵 뷰

### 17.2.5 렌더링 속성

*Scale dependent visibility* 에서는 *Maximum (inclusive)* 및 *Minimum (exclusive)* 축척을 설정해서 피처가 보이게 될 축척 범위를 정의할 수 있습니다. 이 범위를 벗어나면, 피처를 숨깁니다. Set to current canvas scale 버튼을 클릭하면 현재 맵 캔버스의 축척을 가시성 범위의 한계값으로 설정할 수 있습니다. 자세한 내용은가시성 축척 선택기 을 참조하세요.

### 17.2.6 표고 속성

 *Elevation* 탭은*3* 차원 맵 뷰 안에서의 레이어 표고 속성을 제어할 수 있는 옵션들을 제공합니다. 그 중에서도 다음을 설정할 수 있습니다:

- *Elevation Surface*: 3D 레이어의 수직 Z 값을 어떻게 지형 표고로 해석해야 할지를 정의합니다. *Scale* 인자와 *Offset* 값을 적용할 수 있습니다.

### 17.2.7 메타데이터 속성

 *Metadata* 탭에 사용자 레이어에 대한 메타데이터 리포트를 생성하고 편집할 수 있는 옵션이 있습니다. 더 자세한 정보는메타데이터 를 참조하세요.

CHAPTER 18

# 맵 조판 작업

인쇄 조판기와 보고서를 사용해서 맵과 지도책을 생성하고, 인쇄하거나 또는 이미지, PDF 또는 SVG 파일로 저장할 수 있습니다.

## 18.1 인쇄 조판기 개요

The print layout provides growing layout and printing capabilities. It allows you to add elements such as the QGIS 2D or 3D map canvas, text labels, images, legends, scale bars, basic shapes, arrows, attribute and simple tables, elevation profiles and HTML frames···You can size, group, align, position and rotate each element and adjust their properties to create your layout.

The layout can be printed or exported to image formats, PostScript, PDF or to SVG. You can save the layout as a template and load it again in another session. Finally, generating several maps based on a template can be done through the atlas generator.

### 18.1.1 초보자를 위한 예시 세션

인쇄 조판기 작업을 시작하기 전에, QGIS 맵 캔버스에 래스터 또는 벡터 레이어를 몇 개 불러온 다음 레이어 속성을 사용자 편의에 맞춰 조정해야 합니다. 모든 것이 사용자 마음에 들도록 렌더링되고 심볼도 적용된 후에야, *Project* 툴바에 있는 New Print Layout 아이콘을 클릭하거나 *Project* → *New Print Layout* 메뉴 옵션을 선택하십시오. 새 조판기의 제목을 설정하라는 메시지가 뜰 것입니다.

맵을 어떻게 생성하는지 보고 싶다면 다음 지침을 따라주십시오.

1. 대화창 좌측에서, Add map 툴바 버튼을 클릭하고 왼쪽 마우스 버튼을 누른 채 조판기 캔버스 위에 직사각형을 드래그하십시오. 이 직사각형 안쪽에 QGIS 맵 뷰가 추가됩니다.

2. Add scalebar 툴바 버튼을 클릭하고 인쇄 조판기 캔버스를 왼쪽 클릭하십시오. 캔버스에 축척 막대가 추가됩니다.

3.  Add legend 툴바 버튼을 클릭하고 왼쪽 마우스 버튼을 누른 채 캔버스 위에 직사각형을 드래그하십시오. 이 직사각형 안쪽에 범례를 렌더링할 것입니다.
4.  Select/Move item 아이콘을 클릭해서 캔버스 상에 있는 지도를 선택한 다음 조금 움직여보십시오.
5. 지도 항목이 선택 상태인 동안 그 크기를 변경할 수도 있습니다. 지도 항목의 모서리에 있는 하얀색 정사각형을 클릭하고 왼쪽 마우스 버튼을 누른 채 새 위치로 드래그하면 지도의 크기를 변경합니다.
6. 좌하단에 있는 *Item Properties* 패널을 선택한 다음 방향 관련 옵션을 찾아, *Map orientation* 의 설정값을'15.00°' 로 변경하십시오. 지도 항목의 방향이 바뀐 것을 볼 수 있을 겁니다.
7. 이제 사용자가 만든 인쇄 조판을 인쇄하거나 또는 *Layout* 메뉴에 있는 내보내기 도구를 통해 이미지 포맷, PDF 또는 SVG 파일로 내보낼 수 있습니다.
8. 마지막으로,  Save Project 버튼을 클릭하면 프로젝트 파일 안에 사용자의 인쇄 조판을 저장할 수 있습니다.

조판기에 여러 요소들을 추가할 수 있습니다. 인쇄 조판기 캔버스에 하나 이상의 맵 뷰, 범례 또는 축척 막대를 놓을 수도 있고, 지도를 한 장 이상 조판할 수도 있습니다. 각 요소는 고유 속성을 보유하고 있으며, 지도의 경우 고유 범위를 보유하고 있습니다. 조판기 캔버스에서 어떤 항목을 제거하고 싶다면, `Delete` 또는 `Backspace` 키를 눌러 삭제할 수 있습니다.

### 18.1.2 조판 관리자

*Layout Manager* 란 프로젝트의 인쇄 조판을 관리할 수 있는 주 대화창입니다. 기존 인쇄 조판의 오버뷰 및 프로젝트의 보고서를 볼 수 있고, 다음과 같은 도구를 제공합니다:

- 조판을 검색할 수 있습니다.
- 새 인쇄 조판을 추가하고, 기초부터 보고서를 추가하고, 템플릿을 추가하거나 기존 템플릿을 복제할 수 있습니다.
- 인쇄 조판, 보고서, 템플릿을 재명명하거나 삭제할 수 있습니다.
- 프로젝트에서 인쇄 조판, 보고서, 템플릿을 열 수 있습니다.

조판 관리자 대화창을 열려면:

- QGIS 주 대화창에서 *Project* → *Layout Manager*…메뉴를 선택하거나 *Project Toolbar* 에 있는  Layout Manager 버튼을 클릭하십시오.
- 인쇄 조판기 또는 보고서 대화창에서 *Layout* → *Layout Manager*…메뉴를 선택하거나 *Layout Toolbar* 에 있는  Layout Manager 버튼을 클릭하십시오.

조판 관리자 상단에 있는 목록은 프로젝트에서 사용할 수 있는 모든 인쇄 조판 또는 보고서를 담고 있습니다. 다음과 같은 도구들이 있습니다:

- *Show*: 여러 보고서 그리고/또는 인쇄 조판 (들) 을 클릭 한 번으로 선택하고 열 수 있습니다. 이름을 더블클릭해서 열 수도 있습니다.
- *Duplicate*…: 선택한 인쇄 조판 또는 보고서를 복제합니다. (이미 항목 1 개를 선택한 경우에만 사용할 수 있습니다.) 선택한 항목을 템플릿으로 사용하는 새 대화창을 엽니다. 새 조판기를 위한 새 제목을 지정하라는 메시지가 뜰 것입니다.
- *Rename*…: 선택한 조판 또는 보고서를 재명명합니다. (이미 항목 1 개를 선택한 경우에만 사용할 수 있습니다.) 새 조판기를 위한 새 제목을 지정하라는 메시지가 뜰 것입니다.
- *Remove*…: 조판을 제거합니다. 프로젝트에서 선택한 인쇄 조판 (들) 을 삭제할 것입니다.

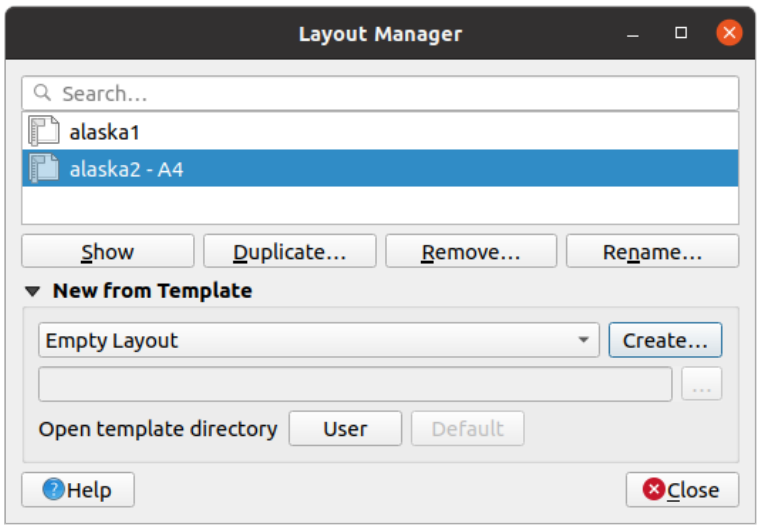



그림 18.1: 인쇄 조판 관리자

조판 관리자 하단에서 새 인쇄 조판을 생성하고, 기초부터 보고서를 생성하거나 템플릿을 생성할 수 있습니다. 기본적으로, QGIS 는 (프레임 하단에 있는 두 버튼으로 접근할 수 있는) 사용자 프로파일 *<user_profiles>* 폴더와 응용 프로그램 템플릿 디렉터리는 물론, *Settings* → *Options* → *Layouts* 메뉴에서 *Path(s) to search for extra print templates* 로 선언된 모든 폴더에서도 템플릿을 찾을 것입니다. 찾아낸 모든 템플릿은 콤보박스 목록에 표시됩니다. 항목을 하나 선택한 다음 *Create* 버튼을 클릭하면 새 보고서 또는 인쇄 조판을 생성합니다.

물론 사용자가 사용자 지정 폴더에 있는 조판 템플릿을 사용할 수도 있습니다. 이 경우 템플릿 드롭다운 목록에서 *specific* 을 선택한 다음, 템플릿을 탐색해서 *Create* 버튼을 클릭하십시오.

> **팁**
>
> **탐색기 패널에서 템블릿 기반 인쇄 조판을 생성하기**
>
> 어느 파일 탐색기에서든 인쇄 조판 템플릿 `.qpt` 파일을 맵 캔버스로 드래그 & 드롭하거나, 또는탐색기 패널 에서 해당 파일을 더블클릭하면 템플릿으로부터 새 인쇄 조판을 생성합니다.

### 18.1.3 인쇄 조판기의 메뉴, 도구 및 패널

인쇄 조판기를 열면 인쇄 옵션 사용 시 종이 표면을 나타내는 텅 빈 캔버스가 표시됩니다. 캔버스 좌측에는 각각 2D 또는 3D 맵 캔버스, 텍스트 라벨, 이미지, 범례, 축척 막대, 기본 도형, 화살표, 속성 또는 단순 테이블, HTML 프레임, 표고 단면 등등과 같은 지도 조판기 항목을 추가할 수 있는 버튼들이 있습니다. 이 툴바에는 조판의 뷰를 탐색하고 영역을 확대/축소하고 이동할 수 있는 버튼은 물론 조판 항목을 선택하고 맵 항목의 범위를 이동시킬 수 있는 버튼도 있습니다.

그림 18.2 는 어떤 항목도 추가하지 않은 상태의 인쇄 조판기 초기 뷰를 보여주고 있습니다.

캔버스 우측에는 패널 2 개가 있습니다. 위 패널은 *Items* 및 *Undo History* 패널을 담고 있고, 아래 패널은 *Layout*, *Item properties* 그리고 *Atlas generation* 패널을 담고 있습니다.

- *Items* 패널은 캔버스에 추가된 지도 조판 항목의 목록을 표시하고 전체 수준에서 조판 항목들과 대화형 작업을 할 수 있는 방법을 제공합니다. (자세한 내용은항목 패널 을 참조하세요.)
- *Undo History* 패널은 조판에 적용된 모든 변경 사항의 이력을 표시합니다. 이력을 클릭하면 특정 상태의 레이아웃으로 되돌리거나 넘어가는 실행 취소 및 재실행을 할 수 있습니다.

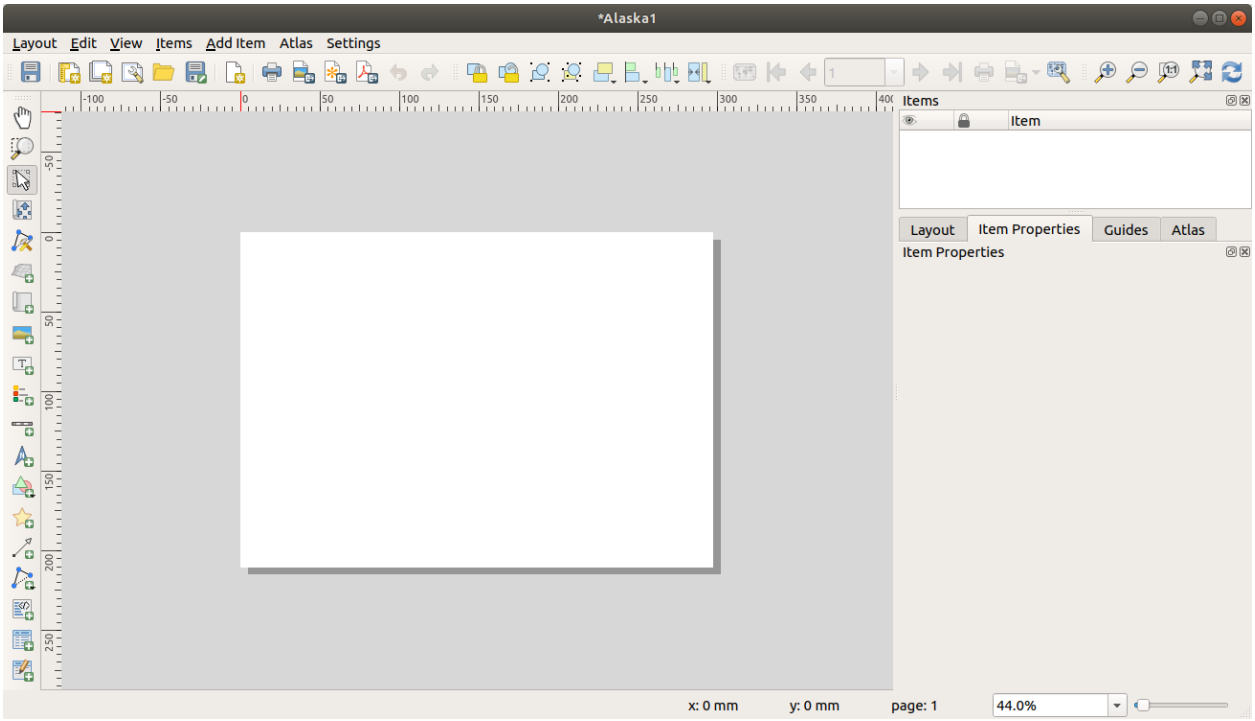



그림 18.2: 인쇄 조판기

- *Layout* 패널에서 조판을 내보내거나 조판기에서 작업하는 경우 적용되는 일반 파라미터들을 설정할 수 있습니다. (자세한 내용은조판 패널 을 참조하세요.)
- *Item Properties* 패널은 패널은 선택한 항목의 속성을 표시합니다. Select/Move item 아이콘을 클릭해서 캔버스 상에 있는 항목을 (예: 범례, 축척 막대 또는 라벨 등을) 선택한 다음, *Item Properties* 패널을 선택해서 선택한 항목에 대한 설정을 사용자 지정할 수 있습니다. (각 항목의 설정에 관한 자세한 정보는조판기 항목 을 참조하세요.)
- *Atlas* 패널에서 현재 조판 용 지도책을 생성하고, 지도 파라미터에 접근할 수 있습니다. (지도책 생성 활용에 관한 자세한 정보는지도책 생성 을 참조하세요.)

인쇄 조판기 대화창 하단에는 마우스 위치, 현재 페이지 번호, 확대/축소 수준을 설정할 수 있는 콤보박스, 선택한 항목이 있을 경우 선택한 항목의 개수 그리고 지도책 생성의 경우 피처 개수를 표시하는 상태 바가 있습니다.

인쇄 조판기 대화창 상단에는 메뉴 및 기타 툴바가 있습니다. 이 메뉴와 툴바의 아이콘으로 모든 인쇄 조판기 도구를 사용할 수 있습니다.

툴바를 오른쪽 클릭하거나, *View* → *Toolbars* → 또는 *View* → *Panels* → 메뉴를 통해 모든 툴바와 패널을 켜고 끌 수 있습니다.

### 메뉴 및 도구

#### 조판기 메뉴

*Layout* 메뉴는 조판을 관리할 수 있는 다음 액션을 제공합니다:

- 인쇄 조판기 대화창에서 바로 프로젝트 파일을 저장합니다.
- *New Layout*…버튼으로 새 비어 있는 인쇄 조판을 생성합니다.
- *Duplicate Layout*…버튼으로 현재 조판을 복제해서 새 인쇄 조판을 생성합니다.
- *Delete Layout*…버튼으로 현재 조판을 제거합니다.
- *Layout Manager*…버튼으로 조판 관리자 대화창을 엽니다.
- *Layouts* 메뉴에서 기존 인쇄 조판을 엽니다.

조판 설계를 완료하면, *Save as Template* 아이콘을 클릭해서 인쇄 조판기 세션의 현재 상태를 `.qpt` 템플릿 파일로 저장하고, 다른 세션/인쇄 조판기에서 *Add Items from Template* 아이콘을 클릭하면 그 항목들을 다시 불러올 수 있습니다.

*Layout* 메뉴에서, QGIS 가 생산한 지리 정보를 공유할 수 있는 – 보고서에 첨부하거나 발행할 수 있는 – 강력한 방법도 제공합니다. *Export as Image*…, *Export as PDF*…, *Export as SVG*…및 *Print*…같은 도구들 말입니다.

다음은 이 메뉴에서 사용할 수 있는 모든 도구들을 요긴한 정보와 함께 정리한 목록입니다.

표 18.1: "Layout"Menu Tools

| 도구 | 단축키 | 툴바 | 참조 |
|---|---|---|---|
| *Save Project* | `Ctrl+S` | *Layout* | *QGIS* 프로젝트 소개 |
| *New Layout* | `Ctrl+N` | *Layout* | 조판 관리자 |
| *Duplicate Layout* | | *Layout* | 조판 관리자 |
| *Delete Layout* | | | |
| *Layout Manager*… | | *Layout* | 조판 관리자 |
| *Layouts* | | | |
| *Layout Properties*… | | | 조판 패널 |
| *Rename Layout*… | | | |
| *Page Properties*… | | | 페이지 속성 작업 |
| *Add Pages*… | | *Layout* | 페이지 속성 작업 |
| *Add Items from Template* | | *Layout* | 조판기 항목 생성하기 |
| *Save as Template*… | | *Layout* | 조판 관리자 |
| *Export as Image*… | | *Layout* | 이미지로 내보내기 |
| *Export as SVG*… | | *Layout* | *SVG* 로 내보내기 |
| *Export as PDF*… | | *Layout* | *PDF* 로 내보내기 |
| *Printer Page Setup*… | `Ctrl+Shift+P` | | |

다음 페이지에 계속

표 18.1 – 이전 페이지에서 계속

| 도구 | 단축키 | 툴바 | 참조 |
|---|---|---|---|
| *Print…* | Ctrl+P | *Layout* | 출력물 생성 작업 |
| *Close* | Ctrl+Q | | |

### 편집 메뉴

*Edit* 메뉴는 레이아웃에 있는 항목을 위한 선택 도구, 복사/잘라내기/붙여넣기 및 실행 취소/재실행 기능과 같은 공통 액션을 포함하는 인쇄 조판 항목을 처리할 수 있는 도구를 제공합니다.

붙여넣기 액션을 사용하는 경우, 현재 마우스 위치에 항목을 붙여넣을 것입니다. *Edit* → *Paste in Place* 액션을 클릭하거나 또는 Ctrl+Shift+V 조합키를 누르면 현재 페이지에 원본 페이지에서의 위치와 동일한 위치로 항목을 붙여넣을 것입니다. 이 방법은 페이지에서 페이지로 동일한 위치에 항목을 복사/붙여넣기할 수 있도록 보장합니다.

다음은 이 메뉴에서 사용할 수 있는 모든 도구들을 요긴한 정보와 함께 정리한 목록입니다.

표 18.2: "Edit"Menu Tools

| 도구 | 단축키 | 툴바 | 참조 |
|---|---|---|---|
| *Undo (last change)* | Ctrl+Z | *Layout* | 실행 취소 (*Undo*) 이력 패널: 액션 되돌리기 및 복구 |
| *Redo (last reverted change)* | Ctrl+Y | *Layout* | 실행 취소 (*Undo*) 이력 패널: 액션 되돌리기 및 복구 |
| *Delete* | Del | | |
| *Cut* | Ctrl+X | | |
| *Copy* | Ctrl+C | | |
| *Paste* | Ctrl+V | | |
| *Paste in place* | Ctrl+Shift+V | | |
| *Select All* | Ctrl+A | | |
| *Deselect all* | Ctrl+Shift+A | | |
| *Invert Selection* | | | |
| *Select Next Item Below* | Ctrl+Alt+[ | | |
| *Select Next Item above* | Ctrl+Alt+] | | |
| *Pan Layout* | P | *Toolbox* | |
| *Zoom* | Z | *Toolbox* | |
| *Select/Move Item* | V | *Toolbox* | 대화형 조판기 항목 작업 |
| *Move Content* | C | *Toolbox* | 맵 항목 |
| *Edit Nodes Item* | | *Toolbox* | 노드 기반 도형 항목 |

### 뷰 메뉴

*View* 메뉴에서 탐색 도구를 사용할 수 있으며, 인쇄 조판기의 일반 습성을 환경 설정할 수 있습니다. 공통적인 확대/축소 도구 외에도, 다음과 같은 액션을 사용할 수 있습니다:

- Refresh view: (뷰가 일관성이 없는 상태인 경우) 뷰를 새로고칩니다.
- 항목을 이동시키거나 생성할 때 항목을 스냅할 수 있는그리드 를 활성화합니다. *Settings → Layout Options…* 메뉴 또는조판기 패널 에서 그리드를 설정할 수 있습니다.
- 항목을 이동시키거나 생성할 때 항목을 스냅할 수 있는가이드 를 활성화합니다. 가이드란 (조판기의 상단 또는 좌측에 있는) 자 (ruler) 를 클릭하면 생성할 수 있는 빨간색 선을 말합니다. 가이드를 원하는 위치로 드래그 & 드롭할 수 있습니다.
- *Smart Guides*: 사용자가 어떤 항목을 이동시키거나 재형성하는 동안 다른 조판 항목을 가이드로 이용해서 동적으로 스냅하게 합니다.
- *Clear Guides*: 모든 현재 가이드를 제거합니다.
- *Show Bounding box*: 사용자가 선택한 항목을 더 잘 식별할 수 있도록 선택한 항목 주위에 경계 상자를 표시합니다.
- *Show Rules*: 조판 주위에 자를 표시합니다.
- *Show Pages*: 용지를 표시하거나, 또는 투명하게 설정합니다. 조판기를 사용해서 인쇄용이 아닌 — 예를 들어 발표 자료 또는 기타 문서에 삽입하기 위한 — 조판을 생성하는 일은 흔합니다. 따라서 조판을 완전히 투명한 배경으로 내보내는 기능이 필요합니다. 다른 편집 소프트웨어 패키지에서는"무한 캔버스"라고 불리기도 합니다.

인쇄 조판기에서 마우스 휠 또는 상태 바에 있는 슬라이드 바 및 콤보박스를 통해 확대/축소 수준을 변경할 수 있습니다. 조판 영역에서 작업 도중 이동 모드로 전환해야 하는 경우, `Spacebar` 또는 마우스 휠을 누르고 있으면 됩니다. `Ctrl+Spacebar` 조합키를 누르는 동안에는 확대 모드로 전환되며, `Ctrl+Alt+Spacebar` 조합키를 누르는 동안에는 축소 모드로 전환됩니다.

*View* → 메뉴에서 패널과 툴바를 활성화시킬 수 있습니다. 조판과 대화형 작업을 하기 위해 사용할 수 있는 공간을 최대화하려면, *View → Toggle Panel Visibility* 옵션을 체크하거나 `Ctrl+Tab` 조합키를 누르십시오. 모든 패널이 숨겨집니다. 이 옵션을 체크 해제하면 이전에 가시화 상태이던 패널만 복구됩니다.

*View* → *Toggle Full Screen* 옵션을 체크하거나 `F11` 키를 누르면 전체 화면 모드로 전환, 대화형 작업을 할 수 있는 공간을 더 확보할 수도 있습니다.

표 18.3: "View"Menu Tools

| 도구 | 단축키 | 툴바 | 참조 |
|---|---|---|---|
| *Refresh* | `F5` | *Navigation* | |
| *Preview* → | | | |
| *Zoom In* | `Ctrl++` | *Navigation* | |
| *Zoom Out* | `Ctrl+-` | *Navigation* | |
| *Zoom to 100%* | `Ctrl+1` | *Navigation* | |
| *Zoom Full* | `Ctrl+0` | *Navigation* | |
| *Zoom to Width* | | | |
| *Show Grid* | `Ctrl+'` | | 가이드 및 그리드 |
| *Snap to Grid* | `Ctrl+Shift+'` | | 가이드 및 그리드 |

다음 페이지에 계속

표 18.3 – 이전 페이지에서 계속

| 도구 | 단축키 | 툴바 | 참조 |
|---|---|---|---|
| *Show Guides* | `Ctrl+;` | | 가이드 및 그리드 |
| *Snap to Guides* | `Ctrl+Shift+;` | | 가이드 및 그리드 |
| *Smart Guides* | `Ctrl+Alt+;` | | |
| *Manage Guides…* | | | 가이드 패널 |
| *Clear Guides* | | | 가이드 패널 |
| *Show Rulers* | `Ctrl+R` | | |
| *Show Bounding Boxes* | `Ctrl+Shift+B` | | |
| *Show Pages* | | | |
| *Toolbars* | | | 패널 및 툴바 |
| *Panels* | | | 패널 및 툴바 |
| *Toggle Full Screen* | `F11` | | 뷰 |
| *Toggle Panel Visibility* | `Ctrl+Tab` | | 뷰 |

### 항목 메뉴

*Items* 메뉴에서 조판에서의 항목 위치와 항목 사이의 관계를 환경 설정할 수 있습니다. (대화형 조판기 항목 작업를 참조하세요.)

표 18.4: “Items”Menu Tools

| 도구 | 단축키 | 툴바 | 참조 |
|---|---|---|---|
| *Group* | `Ctrl+G` | *Actions* | 항목 그룹화 |
| *Ungroup* | `Ctrl+Shift+G` | *Actions* | 항목 그룹화 |
| *Raise* | `Ctrl+]` | *Actions* | 정렬 및 분포 |
| *Lower* | `Ctrl+[` | *Actions* | 정렬 및 분포 |
| *Bring to Front* | `Ctrl+Shift+]` | *Actions* | 정렬 및 분포 |
| *Send to Back* | `Ctrl+Shift+[` | *Actions* | 정렬 및 분포 |
| *Lock Selected Items* | `Ctrl+L` | *Actions* | 항목 고정하기 |
| *Unlock All* | `Ctrl+Shift+L` | *Actions* | 항목 고정하기 |
| *Align Items* | | *Actions* | 정렬 및 분포 |
| *Distribute Items* | | *Actions* | 항목 이동 및 크기 조정 |
| *Resize* | | *Actions* | 항목 이동 및 크기 조정 |

### 항목 추가 메뉴

조판 항목을 생성하는 도구들입니다. 조판기 항목 에서 각 도구를 심도 있게 설명하고 있습니다.

표 18.5: "Add Item"Menu Tools

| 도구 | 툴바 | 참조 |
|---|---|---|
| *Add Map* | *Toolbox* | 맵 항목 |
| *Add 3D Map* | *Toolbox* | *3* 차원 맵 항목 |
| *Add Picture* | *Toolbox* | 그림 항목 |
| *Add Label* | *Toolbox* | 라벨 항목 |
| *Add Dynamic Text* | | 라벨 항목 |
| *Add Legend* | *Toolbox* | 범례 항목 |
| *Add Scale Bar* | *Toolbox* | 축척 막대 항목 |
| *Add North Arrow* | *Toolbox* | 방위표 항목 |
| *Add Shape* | *Toolbox* | 정규 도형 항목 |
| *Add Rectangle* | *Toolbox* | 정규 도형 항목 |
| *Add Ellipse* | *Toolbox* | 정규 도형 항목 |
| *Add Triangle* | *Toolbox* | 정규 도형 항목 |
| *Add Marker* | *Toolbox* | 마커 항목 |
| *Add Arrow* | *Toolbox* | 화살표 항목 |
| *Add Node Item* | *Toolbox* | 노드 기반 도형 항목 |
| *Add Polygon* | *Toolbox* | 노드 기반 도형 항목 |
| *Add Polyline* | *Toolbox* | 노드 기반 도형 항목 |
| *Add HTML* | *Toolbox* | *HTML* 프레임 항목 |
| *Add Attribute Table* | *Toolbox* | 속성 테이블 항목 |
| *Add Fixed Table* | *Toolbox* | 고정 테이블 항목 |
| *Add Elevation Profile* | *Toolbox* | 표고 단면 항목 |

### 지도책 메뉴

표 18.6: "Atlas"Menu Tools

| 도구 | 단축키 | 툴바 | 참조 |
|---|---|---|---|
| *Preview Atlas* | `Ctrl+ALt+/` | *Atlas* | 미리보기 및 지도책 생성 |
| *First Feature* | `Ctrl+<` | *Atlas* | 미리보기 및 지도책 생성 |
| *Previous Feature* | `Ctrl+,` | *Atlas* | 미리보기 및 지도책 생성 |

다음 페이지에 계속

표 18.6 – 이전 페이지에서 계속

| 도구 | 단축키 | 툴바 | 참조 |
|---|---|---|---|
| *Next Feature* | `Ctrl+.` | *Atlas* | 미리보기 및 지도책 생성 |
| *Last feature* | `Ctrl+>` | *Atlas* | 미리보기 및 지도책 생성 |
| *Print Atlas…* | | *Atlas* | 미리보기 및 지도책 생성 |
| *Export Atlas as Images…* | | *Atlas* | 미리보기 및 지도책 생성 |
| *Export Atlas as SVG…* | | *Atlas* | 미리보기 및 지도책 생성 |
| *Export Atlas as PDF…* | | *Atlas* | 미리보기 및 지도책 생성 |
| *Atlas Settings* | | *Atlas* | 지도책 생성 |

### 설정 메뉴

*Settings* ▸ *Layout Options…*메뉴는 QGIS 주 캔버스의 *Settings* ▸ *Options* ▸ *Layouts* 메뉴를 가리키는 단축키입니다. 이 메뉴에서 새로운 모든 인쇄 조판에 대해 기본값으로 사용될 몇몇 옵션을 설정할 수 있습니다:

- *Layout defaults*: 사용할 기본 글꼴을 지정할 수 있습니다.
- *Grid appearance*: 그리드 스타일 및 색상을 설정할 수 있습니다. 그리드 유형은 **점선 (Dot line)**, **실선 (Solid line)** 그리고 **십자 (Cross)** 가운데 하나를 선택할 수 있습니다.
- *Grid and guide defaults*: 그리드의 간격, 오프셋 및 허용 오차를 정의할 수 있습니다. (자세한 내용은가이드 및 그리드 를 참조하세요.)
- *Layout Paths*: 인쇄 템플릿을 검색할 사용자 지정 경로 목록을 관리할 수 있습니다.

*Settings* ▸ *Keyboard Shortcuts…*메뉴를 선택하면 인쇄 조판 인터페이스의단축키 관리자 를 사용할 수 있습니다.

### 컨텍스트 메뉴

인쇄 조판기 대화창에서 어디를 오른쪽 클릭하느냐에 따라, 다양한 기능을 가진 컨텍스트 메뉴를 열 수 있습니다:

- 메뉴 바 또는 툴바를 오른쪽 클릭하면 클릭 한 번으로 활성화 또는 비활성화할 수 있는 조판기 패널 및 툴바 목록을 표시합니다.
- 자를 오른쪽 클릭하면 *Show Guides* 가이드를 가시화하거나, *Snap to Guides* 가이드에 스냅하도록 설정하거나, *Manage Guides…* 가이드 패널 을 열거나 또는 *Clear Guides* 가이드를 제거할 수 있습니다. 자를 숨길 수도 있습니다.
- 인쇄 조판기 캔버스를 오른쪽 클릭하면:
  - 최근 변경 사항을 *Undo* 실행 취소하고 *Redo* 재실행할 수 있고, 또는 복사한 항목을 *Paste* 붙여넣을 수 있습니다. (선택한 항목이 없을 경우에만 가능합니다.)
  - 페이지를 클릭한 경우, 현재페이지 속성 패널을 추가로 열 수 있고, 또는 *Remove Page* 페이지를 제거할 수도 있습니다.
  - 선택한 항목을 클릭했다면 해당 항목을 잘라내거나 복사할 수 있는 것은 물론항목 속성 패널도 열 수 있습니다.
  - 하나 이상의 항목을 선택한 경우 선택 집합을 그룹화할 수도 있고, 그리고/또는 적어도 하나 이상의 그룹이 이미 선택 집합에 존재할 경우 그룹을 해제할 수도 있습니다.

- 모든 조판기 패널의 텍스트란이나 스핀박스 위젯 안을 오른쪽 클릭하면 그 내용을 수정할 수 있는 편집 옵션을 표시합니다.

### 조판 패널

*Layout* 패널에서 사용자의 인쇄 조판의 전체 수준 설정을 정의할 수 있습니다.

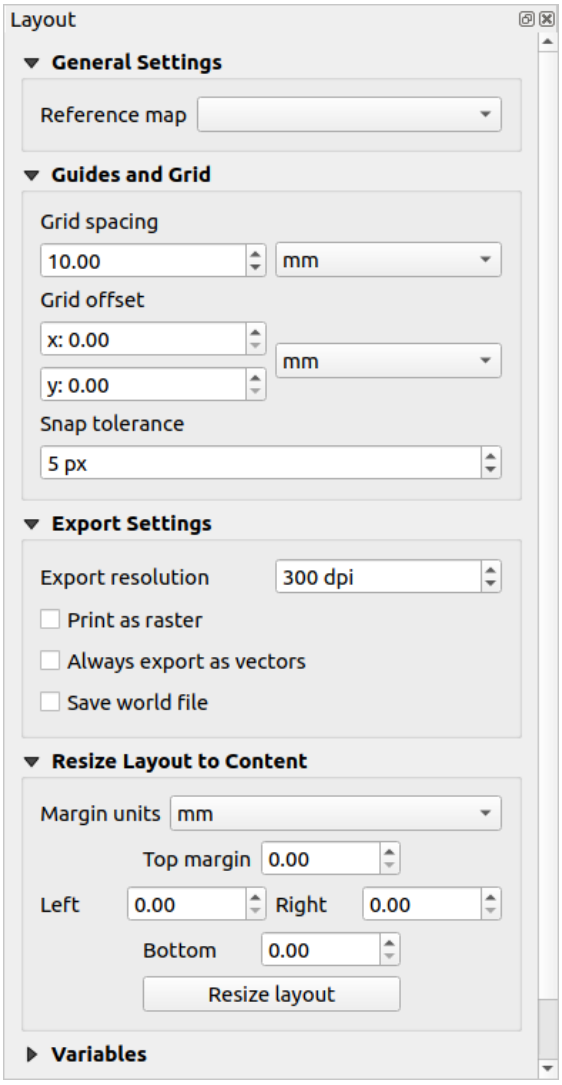



그림 18.3: 인쇄 조판기에서 조판 설정하기

### 일반 설정

인쇄 조판에서 하나 이상의 맵 항목을 사용할 수 있습니다. *Reference map* 은 조판의 마스터 맵으로 사용할 맵 항목을 나타냅니다. 조판에 맵 항목이 하나라도 존재하는 한 마스터 맵이 할당됩니다. 조판기는 인쇄 조판을 지리참조 포맷으로 내보내는 작업도 포함해서, 계산 단위 또는 축척을 사용하는 모든 속성 및 변수에 이 맵을 사용할 것입니다.

게다가, 축척 막대, 범례, 또는 방위표 같은 새 조판 항목의 (방향, 표시되는 레이어, 축척 등의) 설정은 기본적으로 조판 항목이 그려지는 맵 항목에 종속됩니다. 그리고 중첩하는 맵이 없는 경우 참조 맵에 떨어지게 됩니다.

### 가이드 및 그리드

사용자 조판 용지 위에 일부 항목을 정확하게 배치하는 데 도움이 되는 참조 표시를 할 수 있습니다. 다음 가운데 하나를 이 표시로 선택할 수 있습니다:

- 사용자가 원하는 위치에 놓을 수 있는 (**가이드** 라고 하는) 단순 수평 또는 수직선 (가이드 생성에 대한 정보는가이드 패널 을 참조하세요.)
- 또는 정규 **그리드**: 조판 위에 겹쳐지는 수평 및 수직선들의 망

이 그룹에서 *Grid spacing* 또는 *Grid offset* 같은 설정을 조정할 수 있고, 항목을 위해 사용할 수 있는 *Snap tolerance* 설정도 지정할 수 있습니다. 허용 오차란 항목을 이동, 크기 조정 또는 생성하는 동안 마우스 커서가 그리드 또는 가이드에서 해당 거리 안에 있는 경우 그리드 또는 가이드에 스냅하는 최장 거리를 말합니다.

*View* 메뉴에서 그리드 또는 가이드를 표시해야 할지 여부를 설정합니다. 이 메뉴에서, 조판 항목을 스냅하는 데 그리드 또는 가이드를 이용할지도 결정할 수 있습니다. 그리드 선과 가이드 선이 어떤 포인트의 허용 오차 내에 있는 경우, 언제나 가이드를 우선합니다. 가이드는 사용자가 직접 설정하기 때문입니다. (따라서, 가이드가 스냅하기에 너무나 바람직한 위치에 명백히 배치되었다고 가정하기 때문에 일반 그리드보다 우선해서 선택되어야 하는 것입니다.)

> **참고**
>
> *Settings* → *Layout Options* 메뉴에서도 앞에서 언급한 그리드 및 가이드의 파라미터들을 설정할 수 있습니다. 하지만 이 옵션들은 기본적으로 새 인쇄 조판기에만 적용됩니다.

### 내보내기 설정

*Export resolution* 옵션에서 모든 내보내기 지도에 적용될 해상도를 정의할 수 있습니다. 지도를 내보낼 때마다 이 설정을 무시할 수 있습니다.

몇몇 고급 렌더링 옵션 (혼합 모드, 효과 등등) 때문에, 조판 항목을 정확하게 내보내려면 먼저 래스터화해야 할 수도 있습니다. QGIS 는 다른 모든 항목도 래스터화하도록 강제하지 않고 개별적으로 래스터화할 것입니다. 이렇게 해야 PostScript 또는 PDF 로 인쇄하거나 저장하면서 가능한 한 많은 항목을 벡터로 유지할 수 있습니다. 예를 들어 레이어 투명도가 적용된 맵 항목이 라벨, 축척 막대 등도 래스터화하도록 강제하지 않을 것입니다. 하지만:

- ☑ *Print as raster* 를 체크하면 모든 항목을 강제로 래스터화할 수 있습니다.
- 또는 정반대의 옵션, 예를 들어 *Always export as vectors* 를 선택하면 호환 포맷으로 내보낼 때 모든 항목을 벡터로 유지하도록 강제할 수 있습니다. 몇몇 경우 이 옵션을 선택하면 산출물이 조판과 달라보이게 될 수도 있다는 사실을 기억하십시오.

포맷이 지원하는 경우 (예: `.TIF`, `.PDF`) 인쇄 조판을 내보내면 기본적으로 지리참조 파일을 (*General settings* 그룹에 있는 *Reference map* 항목을 기반으로) 생성합니다. 다른 포맷의 경우, 지리참조 산출물을 생성하려면 ☑ *Save world file* 옵션을 체크해서 월드 파일을 생성해야 합니다. 월드 파일은 내보낸 지도와 같은 디렉터리에 생성되며, 참조 맵

항목을 가진 페이지와 동일한 이름을 가지게 됩니다. 월드 파일은 내보낸 지도를 쉽게 지리 참조할 수 있는 정보를 담고 있습니다.

### 조판을 내용에 맞춰 크기 조정

이 그룹에서 *Resize page* 도구를 사용하면, 인쇄 조판의 현재 내용을 커버하는 범위를 가진 유일한 페이지 조판을 생성합니다. (잘라낸 경계 주위로 *margins* 여백을 설정할 수도 있습니다.)

이 습성은내용에 맞춰 자르기 옵션과는 다릅니다. 모든 항목들을 기존 모든 페이지들을 대체하는 유일한 실재하는 페이지 위에 배치하기 때문이죠.

### 변수

*Variables* 는 조판기 수준에서 사용할 수 있는 모든 변수의 목록을 표시합니다. (전체 및 프로젝트 수준의 모든 변수를 포함합니다.)

또한 사용자가 조판기 수준 변수를 관리할 수도 있습니다. 버튼을 클릭하면 새 사용자 지정 조판기 수준 변수를 추가합니다. 마찬가지로, 목록에서 사용자 지정 조판기 수준 변수를 선택한 다음 버튼을 클릭하면 선택한 변수를 제거합니다.

일반 도구 에서 변수 활용에 관한 더 자세한 정보를 살펴볼 수 있습니다.

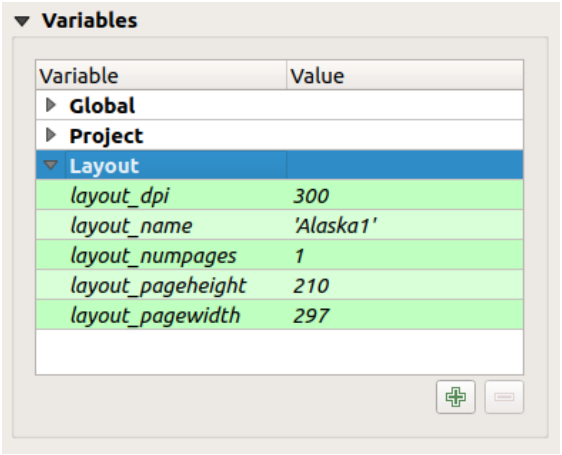



그림 18.4: 인쇄 조판기의 변수 편집기

### 페이지 속성 작업

조판을 여러 페이지로 구성할 수도 있습니다. 예를 들어, 첫 페이지에 맵 캔버스를 놓고, 두 번째 페이지에 레이어 관련 속성 테이블을 놓고, 세 번째 페이지에 사용자 조직의 웹사이트로 연결된 HTML 프레임을 놓을 수 있습니다. 또는 각 페이지에 여러 유형의 항목을 추가할 수 있습니다.

### 새 페이지 추가하기

더 깊게 들어가면, 페이지를 서로 다른 크기 그리고/또는 방향으로 설정해서 조판을 만들 수 있습니다. 페이지를 추가하려면, *Layout* 메뉴 또는 *Layout Toolbar* 에서 *Add Pages*…도구를 선택하십시오. *Insert Pages* 대화창이 열리고 다음을 입력해야 합니다:

- 삽입할 페이지 수
- 페이지 (들) 의 위치: 지정한 페이지의 앞 또는 뒤, 또는 인쇄 조판의 맨 마지막
- *Page size*: 미리 설정된 판형의 (`A4`, `B0`, `Legal`, `Letter`, `ANSI A`, `Arch A` 및 파생 판형은 물론 `1920x1080` 또는 `1024x768` 같은 해상도 유형의) 페이지 (*Orientation* 세로모드 (Portrait) 또는 가로모드 (Landscape) 속성도 포함)

  물론 페이지 크기를 사용자 지정 판형으로 정의할 수도 있습니다. `custom` 을 선택한 경우, 페이지의 *Width* 및 *Height* 를 (필요한 경우 고정된 비율로) 입력해야 합니다. 그리고 `mm`, `cm`, `px`, `pt`, `in`, `ft` 등등 가운데 사용할 단위도 선택해야 합니다. 단위를 전환할 때마다 입력한 값도 자동적으로 변환합니다.

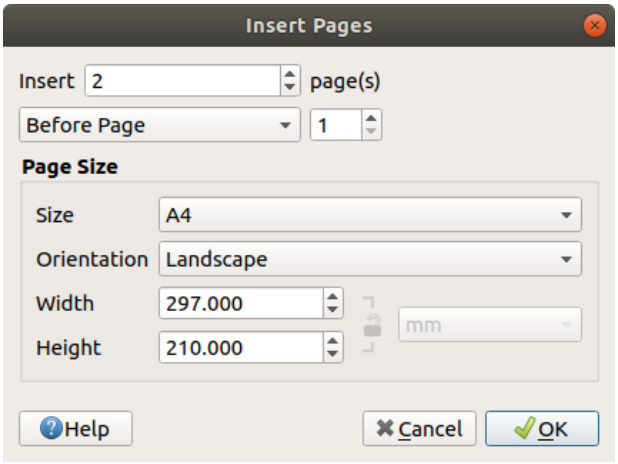



그림 18.5: 인쇄 조판기에서 새 페이지 생성하기

### 페이지 속성 업데이트하기

Any page can be later customized through the Page *Item Properties* panel. To access a page's properties, either:

- Left-click on an empty section of the page
- Right-click on a page and select *Page Properties*….
- Go to *Layout* → *Page properties*… while hovering over the page

The *Item Properties* panel opens with settings such as:

- 여러 페이지를 가진 조판의 경우, 활성 페이지 번호와 사용할 수 있는 페이지 개수를 표시합니다.
- 앞에서 설명한 *Page size* 를 설정할 수 있습니다. 데이터 정의 무시 옵션으로 각 속성을 수정할 수 있습니다. (활용례를 보려면지도책과 데이터 정의 무시 버튼 탐구 을 참조하세요.)
- 현재 페이지의 *Background* 배경을 사용자가 원하는색상 또는심볼 로 설정할 수 있습니다.
- the *Apply to all Pages* button updates the current page properties to the other existing pages in the layout;
- ☐ *Exclude page from exports* 옵션은조판기 산출물 에 현재 페이지를 그 내용과 함께 포함시켜야 할지 여부를 제어할 수 있습니다.

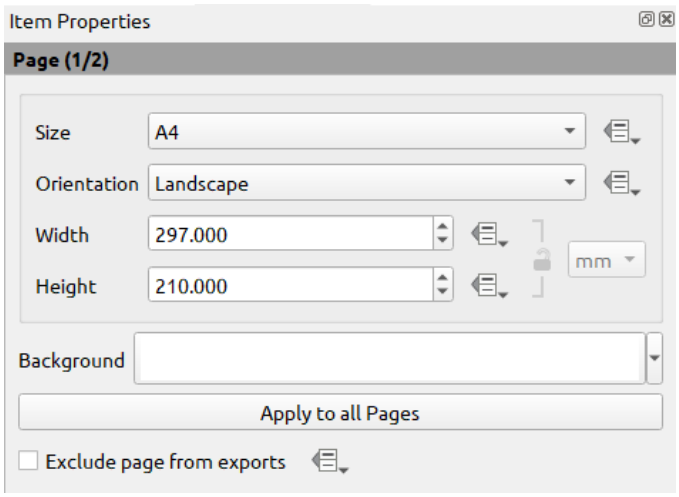



그림 18.6: 페이지 속성 대화창

### 가이드 패널

가이드란 사용자가 항목을 생성, 이동, 또는 크기 조정할 때 항목 배치 작업을 도와주기 위해 조판 페이지에 표시할 수 있는 수직 또는 수평 참조선을 말합니다. 가이드를 활성화시키려면 *View → Show Guides* 및 *View → Snap to Guides* 옵션을 체크하면 됩니다. 가이드를 생성하려면, 다음 두 가지 방법이 있습니다:

- *View → Show Rulers* 옵션을 설정한 경우, 자를 클릭 & 드래그한 다음 페이지 영역의 원하는 위치에서 마우스 버튼을 놓으십시오.
- 정밀도를 더 높이려면, *View → Toolbox* 에서 또는 페이지의 컨텍스트 메뉴에서 *Manage guides for page…*를 선택해서 *Guides* 패널을 선택하십시오.

*Guides* 패널에서, 지정한 위치에 스냅 선을 생성할 수 있습니다:

1. 가이드를 추가하고 싶은 *Page* 를 선택하십시오.
2. Add new guide 버튼을 클릭하고 수평 또는 수직 선의 좌표를 입력하십시오. 원점은 좌상단입니다. 서로 다른 단위를 선택할 수 있습니다.

   이 패널에서 기존 가이드의 위치를 정확한 좌표로 조정할 수도 있습니다. 기존 가이드를 더블클릭한 다음 대체값을 입력하십시오.
3. *Guides* 패널에 있는 목록은 현재 페이지용 항목만 담고 있습니다. 현재 페이지에만 가이드를 생성하거나 제거할 수 있습니다. 하지만, *Apply to All Pages* 버튼으로 현재 페이지의 가이드 환경 설정을 조판에 있는 다른 페이지로 복제할 수 있습니다.
4. 가이드를 삭제하려면, 가이드를 선택하고 Remove selected guide 버튼을 클릭하십시오. 현재 페이지에 있는 모든 가이드를 제거하려면 *Clear All Guides* 를 사용하면 됩니다.

> **팁**
>
> 기존 조판 항목에 스냅하기
>
> 새 항목을 이동, 크기 조정, 또는 생성할 때 가이드 및 그리드 외에 기존 항목을 스냅 참조물로 사용할 수 있습니다. 이를 **스마트 가이드** 라 하며, *View → Smart Guides* 옵션을 체크해야 사용할 수 있습니다. 마우스 커서가 어떤 항목의 경계에 가까이 갈 때마다 스냅 십자가 나타납니다.

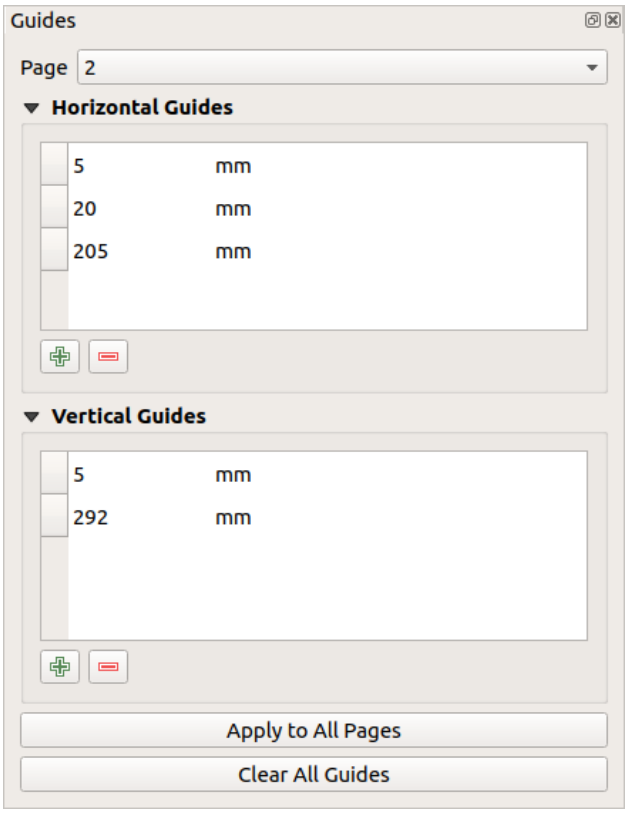

Guides
Page 2
Horizontal Guides
5 mm
20 mm
205 mm
Vertical Guides
5 mm
292 mm
Apply to All Pages
Clear All Guides

그림 18.7: 가이드 패널

### 항목 패널

*Items* 패널에는 항목의 선택 및 가시성을 관리하는 몇몇 옵션이 있습니다. 인쇄 조판기 캔버스에 추가된 (항목 그룹 도 포함하는) 모든 항목의 목록이 표시됩니다. 캔버스에서 항목을 선택하면 이 목록에 있는 대응하는 행도 선택되며, 이 목록에서 행을 선택해도 캔버스에 있는 대응하는 항목이 선택됩니다. 즉 다른 항목 아래에 위치한 항목을 쉽게 선택할 수 있습니다. 선택한 행은 볼드체로 표시된다는 사실을 기억하십시오. 다중 항목 선택도 가능합니다. 연속하는 항목들을 선택하려면 `Shift` 키를 누르고 있으면 되고, 연속하지 않는 항목들을 선택하려면 `Ctrl` 키를 누르고 있으면 됩니다.

선택한 항목에 모두에 대해, 다음 설정을 할 수 있습니다:

- : 항목을 보이도록 또는 보이지 않도록 설정합니다.
- : 항목의 위치를 고정하거나 고정을 해제합니다.
- 항목의 Z 위치를 정렬합니다. 목록에 있는 각 항목을 클릭 & 드래그해서 위아래로 이동시킬 수 있습니다. 목록의 상위에 있는 항목은 인쇄 조판기 캔버스에서 전경으로 나올 것입니다. 기본적으로, 새로 생성한 항목이 전경에 배치됩니다.
- 텍스트를 더블클릭하면 항목 ID 를 변경할 수 있습니다.
- 항목을 오른쪽 클릭하면 항목을 복사 또는 삭제할지, 또는 항목의속성 패널 을 열지를 선택할 수 있습니다.

어떤 항목을 위한 정확한 위치를 찾았다면, 열에 있는 체크박스를 체크해서 위치를 고정시킬 수 있습니다. 고정된 항목은 캔버스 상에서 선택할 수 **없습니다**. *Items* 패널에서 항목을 선택한 다음 체크박스를 체크 해제해서 고정된 항목을 고정 해제할 수 있습니다. 또는 툴바에 있는 아이콘을 클릭하면 고정된 모든 항목을 고정 해제할 수 있습니다.

### 실행 취소 (Undo) 이력 패널: 액션 되돌리기 및 복구

조판 과정 도중, 변경 사항을 되돌리고 복구할 수 있습니다. *Edit* 메뉴, *Layout* 툴바 또는 인쇄 조판 영역을 오른쪽 클릭할 때마다 표시되는 컨텍스트 메뉴에서 사용할 수 있는 되돌리기 및 복구 도구를 사용해서 말이죠.

- Revert last change
- Restore last change

*Undo history* 패널 안에서 (그림 18.8 참조) 마우스 클릭으로도 되돌리기/복구 작업을 할 수 있습니다. 이력 패널은 인쇄 조판기 내에서 수행된 최근 액션의 목록을 표시합니다. 사용자가 되돌리고 싶은 시점을 선택하고 새 액션을 실행하면, 선택한 액션 이후에 수행된 모든 액션이 제거될 것입니다.

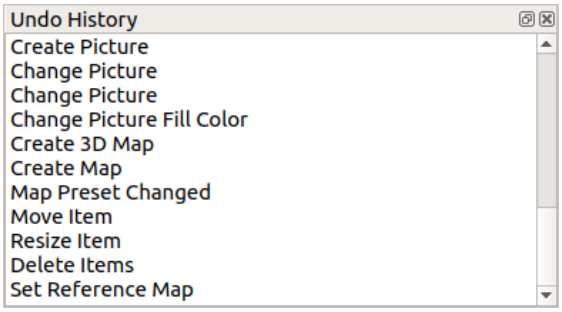



그림 18.8: 인쇄 조판기의 실행 취소 이력

## 18.2 조판기 항목

### 18.2.1 조판기 항목 공통 옵션

QGIS 는 맵을 조판할 수 있는 맵, 범례, 축척 막대, 사진, 표, 방위표, 이미지 유형 등과 같은 수많은 항목들을 제공합니다. 하지만 이 항목들은 다음에서 설명하는 바와 같이 몇몇 공통 옵션 및 습성을 공유하고 있습니다.

#### 조판기 항목 생성하기

서로 다른 도구들을 사용해서 기초부터 또는 기존 항목을 기반으로 새 항목을 생성할 수 있습니다.

기초부터 조판기 항목을 생성하려면:

1. *Add Item* 메뉴 또는 *Toolbox* 툴바 가운데 하나에서 대응하는 도구를 선택하십시오.
2. 그 다음:
   - 페이지를 클릭하면 열리는 *New Item Properties* 대화창에서 필요한 크기 및 배치 정보를 정의하십시오. (자세한 내용은위치 및 크기 를 참조하세요.)

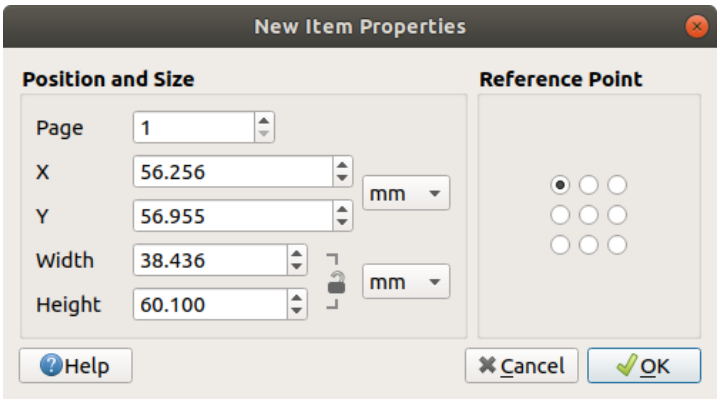



그림 18.9: 새 항목 속성 대화창

   - 또는 항목의 초기 크기 및 배치를 정의하기 위해 클릭 & 드래그할 수도 있습니다. 더 나은 위치를 지정하려면그리드와 가이드 를 사용해도 됩니다.

> **참고**
>
> 항목이 특정 형태를 취할 수 있으므로, 노드 또는 화살표 항목을 그리려면 클릭 한 번이나 클릭 & 드래그 방법으로는 부족합니다. 항목의 각 노드마다 클릭해서 배치해야 합니다. 더 자세한 내용은노드 기반 도형 항목 을 참조하세요.

또한:

1. *Toolbox* 툴바에서  Select/Move item 버튼으로 기존 항목을 선택할 수 있습니다.
2. 컨텍스트 메뉴 또는 *Edit* 메뉴를 사용해서 선택한 항목을 복사하거나 잘라내어 마우스 커서 위치에 새 항목으로 붙여넣을 수 있습니다.

*Paste in Place* (`Ctrl+Shift+V`) 명령어를 사용하면 어떤 페이지에서 다른 페이지로 항목을 복제하고, 새 페이지에 항목을 원본과 동일한 좌표에 배치할 수 있습니다.

게다가, *Layout* → *Add Items from Template*…명령어를 통해 인쇄 조판기 템플릿을 사용해서 (자세한 내용은조판 관리자 참조) 항목을 생성할 수 있습니다.

> **팁**
>
> **파일 탐색기를 통해 조판기 항목 추가**
>
> 사용자의 파일 탐색기 또는 *Browser* 패널에 있는 인쇄 조판기 템플릿을 인쇄 조판기 대화창으로 드래그 & 드롭하면, QGIS 가 조판에 해당 템플릿의 모든 항목을 자동으로 추가합니다.

### 대화형 조판기 항목 작업

완벽한 조판을 생성하기 위해 인쇄 조판 내부에 있는 각 항목을 이동시키고 크기를 조정할 수 있습니다. 이 두 작업 모두, 첫 단계는  Select/Move item 도구를 활성화해서 항목을 클릭하는 것입니다.

 Select/Move item 도구로 항목을 다중 선택할 수 있습니다. 사용자가 원하는 항목들 위로 클릭 & 드래그하거나, `Shift` 키를 누른 채 하나씩 클릭하십시오. 항목 선택을 해제하려면, `Shift` 키를 누른 채 항목을 클릭하십시오.

선택을 할 때마다, 상태 바에 선택한 항목 개수가 표시됩니다. *Edit* 메뉴에서 모든 항목 선택, 모든 선택 해제, 현재 선택 반전 등의 액션을 실행할 수 있습니다.

### 항목 이동 및 크기 조정

*View* → *Show Bounding Boxes* 옵션을 체크 해제하지 않는 이상, 선택한 항목의 경계선에 사각형들을 표시할 것입니다. 이 사각형 가운데 하나를 마우스로 움직이면, 움직이는 방향으로 항목의 크기를 조정할 것입니다. 크기 조정 도중 `Shift` 키를 누르면 가로세로 비율 (aspect ratio) 을 유지할 것입니다. `Alt` 키를 누르면 항목의 중심을 기준으로 크기를 조정할 것입니다.

조판기 항목을 이동하려면, 항목을 마우스로 왼쪽 클릭한 채 움직이십시오. 이동 방향을 수평 또는 수직 축으로 제한하려면 마우스를 움직이는 동안 `Shift` 키를 누르고 있으면 됩니다. 선택한 항목을 `Arrow keys` 화살표 키로 움직일 수도 있습니다. 화살표 키의 이동 속도가 너무 느릴 경우, `Shift` 키를 누르면 속도가 빨라집니다. 이동 정밀도를 높이고 싶다면 *Position and size* 속성 또는 앞에서 항목 생성에 대해 설명한대로 그리드/가이드 스냅을 사용하십시오.

여러 항목들을 동시에 크기 조정하거나 이동시키는 방법은 단일 항목의 경우와 동일합니다. 하지만 QGIS 는 선택한 항목들의 크기를 다음과 같은 서로 다른 규칙을 따라 자동으로 조정할 수 있는 고급 도구를 제공하고 있습니다:

- 각 항목의 높이를 선택한 항목의  최고 또는  최저 높이와 일치시킵니다.
- 각 항목의 너비를 선택한 항목의  최광 (□□) 또는  최협 (□□) 너비와 일치시킵니다.
- 항목 크기를  정사각형에 맞춰 조정: 각 항목을 정사각형 형태로 확장합니다.

항목 위치도 마찬가지로, 선택한 항목들의 위치를 다음을 기준으로 등거리로 분포하도록 만드는도구들 을 사용할 수 있습니다:

- 항목의 경계 (좌측, 우측, 상단 또는 하단)
- 항목의 수평 또는 수직 중심
- 항목 사이의 수평 또는 수직 간격

### 항목 그룹화

항목들을 그룹화하면 항목들의 집합을 단일 항목처럼 처리할 수 있습니다. 여러 항목들을 한꺼번에 쉽게 크기 조정, 이동, 삭제, 복사할 수 있습니다.

항목들의 그룹을 생성하려면, 하나 이상의 항목을 선택한 다음 *Items* 메뉴 또는 *Actions* 툴바 또는 컨텍스트 메뉴에 있는 *Group* 버튼을 누르십시오. *Items* 패널에 `Group` 이라는 행이 추가되어, 다른항목 패널 객체 와 마찬가지로 고정시키거나 숨길 수 있습니다. 캔버스 상에서 그룹화된 항목들을 **개별적으로** 선택할 수는 없습니다. 개별 항목을 직접 선택하고 항목 속성 패널에 접근하려면 항목 패널을 사용하십시오.

### 항목 고정하기

어떤 항목을 위한 정확한 위치를 찾고 나면, *Items* 메뉴 또는 *Actions* 툴바에 있는 *Lock selected items* 버튼을 누르거나, *Items* 패널에서 해당 항목의 옆에 있는 체크박스를 체크해서 위치를 고정시킬 수 있습니다. 고정된 항목은 캔버스 상에서 선택할 수 **없습니다**.

*Items* 패널에서 항목을 선택한 다음 체크박스를 체크 해제해서 고정된 항목을 고정 해제할 수 있습니다. 또는 툴바에 있는 아이콘들을 사용해도 됩니다.

### 정렬 및 분포

조판에 있는 항목들의 Z 위치를 올리거나 내리기 위한 도구는 Raise selected items 풀다운 메뉴에 있습니다. 인쇄 조판기 캔버스 상에서 항목을 선택한 다음 선택한 항목을 다른 항목들 위아래로 올리거나 내릴 수 있는 기능을 선택하십시오. *Items* 패널에 이 순서를 표시하고 있습니다. 또한, *Items* 패널의 이 목록에 있는 객체의 라벨을 클릭 & 드래그해서 객체의 순위를 올리거나 내릴 수도 있습니다.

Align selected items 풀다운 메뉴에서 (그림 18.10 참조) 몇몇 정렬 옵션을 사용할 수 있습니다. 정렬 기능을 사용하려면 먼저 항목을 선택한 다음, 정렬 아이콘 가운데 하나를 클릭하십시오:

- *Align Left* 좌측 또는 *Align Right* 우측 정렬
- *Align Top* 상단 정렬 또는 *Align Bottom* 하단 정렬
- *Align Center* 수평 중심 또는 *Align Center Vertical* 수직 중심 정렬

선택한 모든 항목이 공통 경계 상자로 정렬될 것입니다. 조판기 캔버스 상에서 항목을 이동시킬 때, 경계선, 중심점 또는 모서리가 정렬되는 경우 정렬 도우미 선이 표시됩니다.

조판 항목 배치를 향상시킬 수 있는 또다른 방법은 조판 페이지 전체에 걸쳐 항목 사이의 간격을 조정하는 것입니다. 항목을 선택한 다음 *Distribute Left Edges* 드롭다운 메뉴를 클릭해서 다음 가운데 하나로 설정하면 됩니다:

- *Distribute Left Edges* 또는 *Distribute Right Edges*: 항목의 왼쪽 경계 또는 오른쪽 경계를 등거리로 분포
- *Distribute Top edges* or *Distribute Bottom Edges*: 항목의 상단 경계 또는 하단 경계를 등거리로 분포
- *Distribute Horizontal Centers* 또는 *Distribute Vertical Centers*: 항목의 수평 중심 또는 수직 중심을 등거리로 분포
- *Distribute Horizontal Spacing Equally* 또는 *Distribute Vertical Spacing Equally*: 항목의 수평 또는 수직 사이에 동등한 간격을 추가

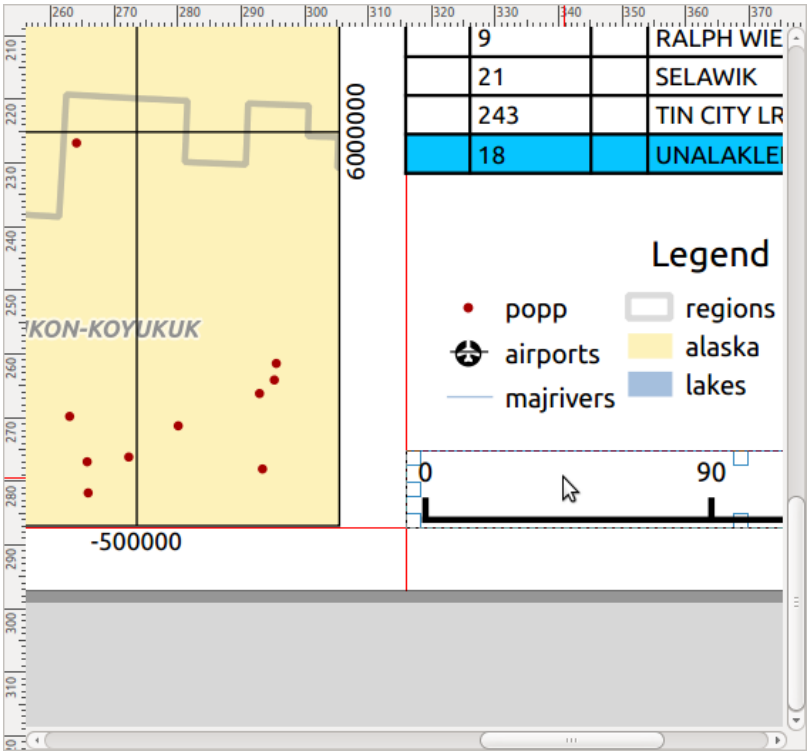



그림 18.10: 인쇄 조판기의 정렬 도우미 선

### 공통 항목 속성

조판기 항목은 *Item Properties* 패널 하단에 보이는 (그림 18.11 참조) 위치 및 크기, 기울기, 프레임, 배경, 항목 ID, 렌더링, 변수 그리고 렌더링이라는 일련의 공통 속성을 보유하고 있습니다.

> **참고**
>
> 대부분의 옵션 옆에 있는 Data defined override 아이콘은 해당 속성을표현식 또는변수 를 사용해서 레이어, 피처 속성, 도형 또는 다른 모든 조판기 항목 속성과 연결시킬 수 있다는 의미입니다. 더 자세한 내용은데이터 정의 무시 설정 을 참조하세요.

- *Position and size* 그룹에서 항목을 담고 있는 프레임의 크기와 위치를 정의할 수 있습니다. (더 자세한 내용은위치 및 크기 를 참조하세요.)
- *Rotation* 부분에서 항목의 기울기를 (도 단위로) 설정할 수 있습니다.
- *Frame* 그룹은 항목 주위에 프레임을 표시할지 숨길지 결정합니다. *Color*, *Thickness* 및 *Join style* 위젯을 사용해서 색상, 두께 및 결합 스타일 속성을 조정할 수 있습니다.
- *Background* 그룹에서 *Color* 배경 색상을 생성하거나 선택할 수 있습니다. 알파 필드 설정을 변경하면 투명도를 조정할 수 있습니다.
- *Item ID* 그룹은 다른 인쇄 조판기 항목과의 관계를 생성합니다. QGIS 서버 또는 다른 잠재적 웹 클라이언트들이 항목 ID 를 이용합니다. 사용자가 어떤 항목 (예: 지도 또는 라벨) 에 ID 를 설정하면, 웹 클라이언트가 해당 특정 항목에 대한 속성 (예: 라벨 텍스트) 을 설정하는 데이터를 전송할 수 있습니다. `GetProjectSettings` 명령어를 실행하면 현재 조판에서 사용할 수 있는 항목과 ID 목록을 표시할 것입니다.

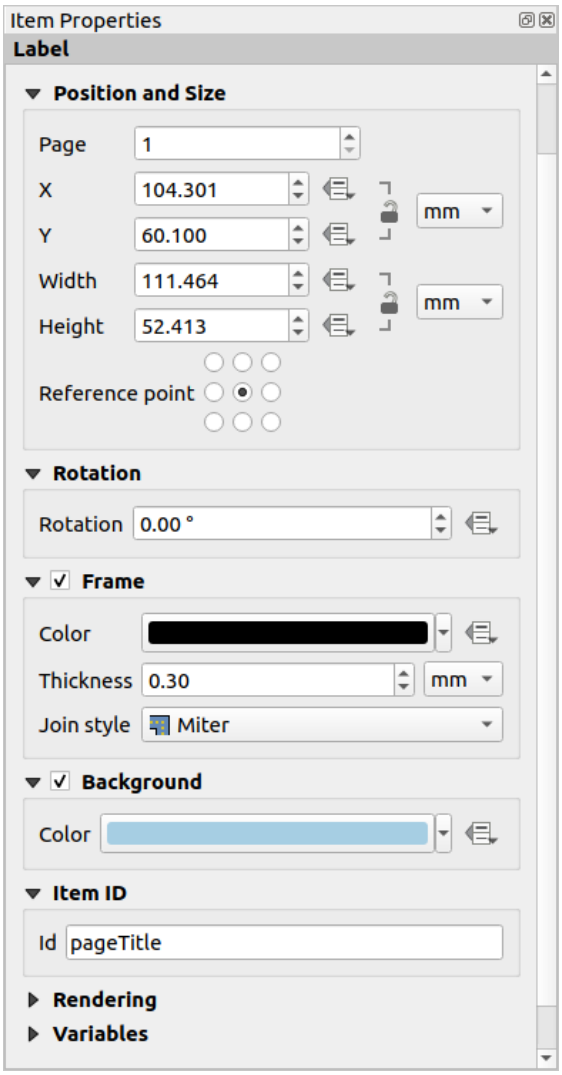



그림 18.11: 공통 항목 속성 그룹

- *Rendering* 모드에서 항목을 표시할지 여부 및 어떻게 표시할지를 설정할 수 있습니다. 예를 들면혼합 모드를 적용하거나, 항목의 투명도를 조정하거나, 또는 *Exclude item from exports* 설정으로 내보내기할 때 특정 항목을 제외시킬 수 있습니다.

### 위치 및 크기

새 항목 속성 대화창의 기능을 데이터 정의 기능을 사용해서 확장시키는 이 그룹은 항목을 정확하게 배치할 수 있게 해줍니다.

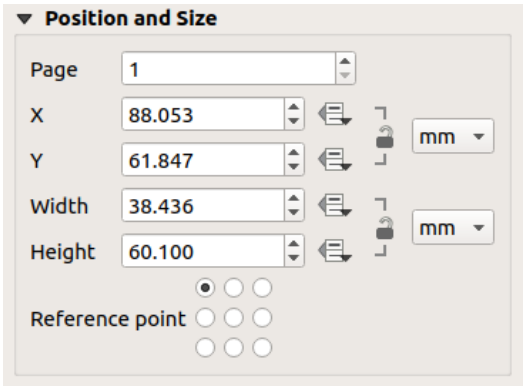



그림 18.12: 위치 및 크기

- 항목을 배치할 페이지의 실제 번호
- 항목이 참조할 지점 (reference point)
- 선택한 페이지에 있는 항목의 *Reference point* 의 *X* 및 *Y* 좌표. 버튼을 클릭하면 이 값들의 비율을 고정시킬 수 있습니다. 비율을 고정한 상태에서 이 두 값 가운데 하나를 위젯 또는 Select/Move item 도구를 사용해서 변경하면, 두 값 모두 영향을 받을 것입니다.
- 항목 경계 상자의 *Width* 및 *Height*. 좌표로 설정할 경우, 너비와 높이 비율을 고정할 수 있습니다.

### 렌더링 모드

QGIS 는 벡터 및 래스터 레이어와 마찬가지로 조판기 항목에 대한 고급 렌더링을 할 수 있습니다.

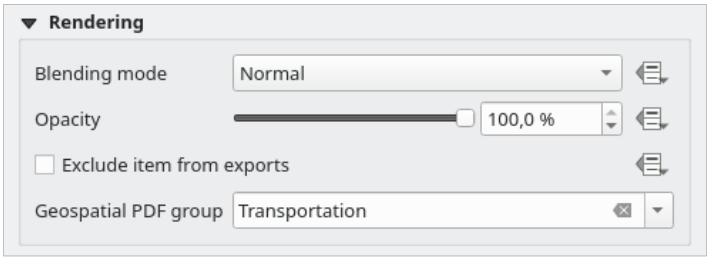



그림 18.13: 렌더링 모드

- *Blending mode*: 이 도구를 통해 그래픽 프로그램에서만 가능했던 특별한 효과를 줄 수 있습니다. 사용자의 위에 있는 항목과 아래에 있는 항목의 픽셀들을 설정한 모드에 따라 혼합할 수 있습니다. (각 효과에 대한 설명은혼합 모드 를 참조하세요.)

- *Opacity*: 이 도구를 통해 조판에서 다른 항목 아래에 있는 항목을 가시화할 수 있습니다. 슬라이드 바를 통해 사용자 항목의 가시성을 필요에 맞춰 조정하십시오. 슬라이드 바 옆에 있는 메뉴에서 가시성을 정확한 백분율로 설정할 수도 있습니다.
- *Exclude item from exports*: 데이터 정의 표현식과 함께 사용하면, 일부 내보내기에서 특정 항목을 보이지 않도록 만들 수 있습니다. 이 체크박스를 활성화하면, 선택한 항목이 인쇄물, PDF 파일 등등에 포함되지 않을 것입니다.
- *Geospatial PDF group*: When set, a matching layer tree group will be created in the exported Geospatial PDF and the item will only be visible when this group is checked. This allows content to be selectively displayed as a group by viewers of the Geospatial PDF. E.g., it can allow extra layout content such as descriptive labels or legends to only be shown when layers from the group are visible, making the file export much more flexible.

#### 변수

*Variables* 는 조판기 항목 수준에서 사용할 수 있는 모든 변수의 목록을 표시합니다. (전체, 프로젝트 및 조판기 수준의 모든 변수를 포함하고 있습니다.) 조판 맵 항목은 맵의 축척, 범위 등과 같은 값에 쉽게 접근할 수 있게 해주는 변수들을 위한 추가적인 *Map settings* 부분을 가지고 있습니다.

*Variables* 에서, 조판 항목 수준 변수를 관리할 수도 있습니다. 버튼을 클릭해서 새 사용자 지정 항목 수준 변수를 추가합니다. 마찬가지로, 목록에서 사용자 지정 항목 수준 변수를 선택한 다음 버튼을 클릭해서 제거합니다.

값을 변수로 저장 에서 변수 활용에 관한 더 자세한 정보를 살펴볼 수 있습니다.

### 18.2.2 맵 항목

맵 항목은 사용자가 맵 캔버스에 설계한 맵을 표시하는 메인프레임입니다. *Add Map* 도구를항목 생성 지침 을 따라 사용해서 맵 항목을 추가하십시오. 이후대화형 조판기 항목 작업 에서 설명한 내용과 동일한 방법으로 새 맵 항목을 수정할 수 있습니다.

기본적으로, 새 맵 항목은맵 캔버스 의 현재 상태를 그 범위와 가시화 레이어와 함께 표시합니다. *Item Properties* 패널에서 맵 항목을 사용자 지정할 수 있습니다. 공통 항목 속성 이외에도, 다음과 같은 기능을 가지고 있습니다:

#### 툴바

맵 항목의 *Item Properties* 패널은 다음 기능을 가진 툴바를 내장하고 있습니다:

- Update map preview: 맵 미리보기를 업데이트합니다.
- Set map canvas to match main canvas extent: 맵 캔버스의 범위를 주 캔버스 범위와 일치시킵니다.
- View current map extent in main canvas: 주 캔버스에 현재 맵의 범위를 표시합니다.
- Set map scale to match main canvas scale: 맵의 축척을 주 캔버스 축적과 일치시킵니다.
- Set main canvas to match current map scale: 주 캔버스의 축척을 현재 맵 축척과 일치시킵니다.
- Bookmarks: 맵 항목의 범위를 기존 공간 북마크와 일치시킵니다.
- Interactively edit map extent: 맵 항목 안에서 대화형으로 이동 및 확대/축소합니다.

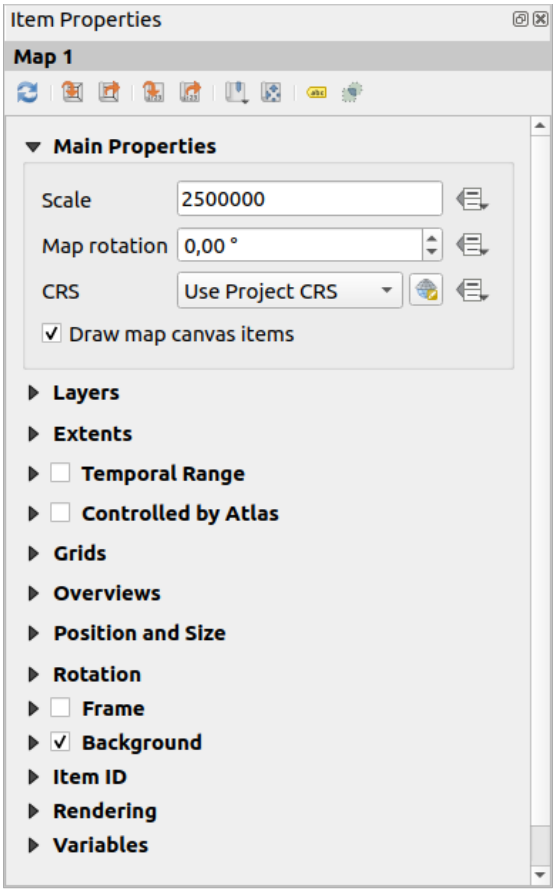



그림 18.14: 맵 항목 속성 패널

- Labeling settings: 조판기 맵 항목 범위에서 (배치, 가시화 등의) 피처 라벨 습성을 제어합니다:
  - *Margin from map edges*: 맵 경계로부터의 여백을 — 맵 항목의 범위로부터 데이터 정의할 수 있는 거리를 — 설정합니다. 이 여백에 어떤 라벨도 표시되어서는 안 됩니다.
  - *Allow truncated labels on edges of map*: 일부가 맵 항목이 허용한 범위 바깥으로 벗어나는 라벨을 렌더링할지 여부를 제어합니다. 이 옵션을 체크하면 (라벨을 완전히 가시 영역 안으로 배치할 방법이 없는 경우) 이런 라벨을 표시할 것입니다. 체크를 해제하면 일부만 보이는 라벨을 무시할 것입니다.
  - *Label blocking items*: **활성화된** 맵 항목에서 (축척 막대, 방위표, 삽입 지도 등과 같은) 다른 조판기 항목을 맵 라벨의 방해물로 설정할 수 있습니다. 이렇게 설정하면 어떤 맵 라벨도 다른 항목 아래 배치되지 않습니다 — 라벨 작업 엔진이 다른 항목과 겹치는 맵 라벨을 다른 위치에 배치하거나, 또는 그냥 무시하도록 할 것입니다.

    *Margin from map edges*: 맵 경계 여백을 설정하면, 맵 라벨을 체크한 조판기 항목으로부터 지정한 거리보다 가까이 배치하지 않을 것입니다.
  - *Show unplaced labels*: 어떤 라벨이 (예를 들면 다른 맵 라벨과의 충돌 또는 라벨을 배치할 공간이 부족하기 때문에) 조판기 맵에서 빠졌는지 여부를 확인할 수 있습니다. 이런 라벨은미리 정의된 색상으로 강조됩니다.
- Clipping settings: 맵 항목을 지도책 피처와 도형 및 폴리곤 항목에 맞춰 자릅니다.
  - *Clip to atlas feature*: 조판기 맵 항목을 자동으로 현재지도책 피처 에 맞춰 자르도록 결정할 수 있습니다.

    다음과 같은 서로 다른 자르기 모드를 사용할 수 있습니다:
    - *Clip During Render Only*: 지도책 피처를 벗어나는 벡터 피처 부분이 보이지 않도록 페인터 기반 자르기를 적용합니다.
    - *Clip Feature Before Render*: 지도책 피처를 벗어나는 부분이 있는 피처의 경계선이 지도책 피처의 경계선에 계속 보이도록, 피처를 렌더링하기 전에 자르기를 적용합니다.
    - *Render Intersecting Features Unchanged*: 현재 지도책 피처와 교차하는 모든 피처를, 피처의 도형을 자르지 않은 채 렌더링합니다.

    *Force labels inside atlas feature* 옵션을 활성화할 수 있습니다. *Clip all layers* 옵션으로 지도책 피처에 맞춰 모든 레이어를 자르고 싶지 않은 경우 *Clip selected layers* 옵션을 선택하면 됩니다.
  - *Clip to item*: 인쇄 조판기에서도형 또는폴리곤 항목을 사용해서 맵 항목의 모양을 변경할 수 있습니다. 이 옵션을 활성화하면 콤보박스에서 선택한 도형에 맞춰 자동으로 맵을 자를 것입니다. 이때 앞에서 설명한 자르기 모드들을 사용할 수 있고, 라벨을 자른 도형 안에만 표시하도록 강제할 수 있습니다.

### 주요 속성

맵 항목 *Item Properties* 패널의 *Main properties* 그룹은 (그림 18.14 참조) 다음 기능을 제공하고 있습니다:

- 맵 캔버스의 뷰를 수정한 경우, *Update Preview* 버튼을 클릭하면 맵 항목을 새로 렌더링합니다. 대부분의 경우 변경 사항이 생기면 자동적으로 맵 항목을 새로고침한다는 점을 기억하세요.
- *Scale* 란에서 맵 항목 축척을 직접 설정합니다.
- *Map rotation* 란에서 맵 항목의 내용물을 시계 방향 도 단위로 기울일 수 있습니다. 여기서 맵 캔버스 자체를 기울이는 것은 제한될 수 있습니다.
- *CRS* 목록에서 맵 항목의 내용물을 표시할 어떤좌표계 든 선택할 수 있습니다. 기본값은 `Use project CRS` 입니다.

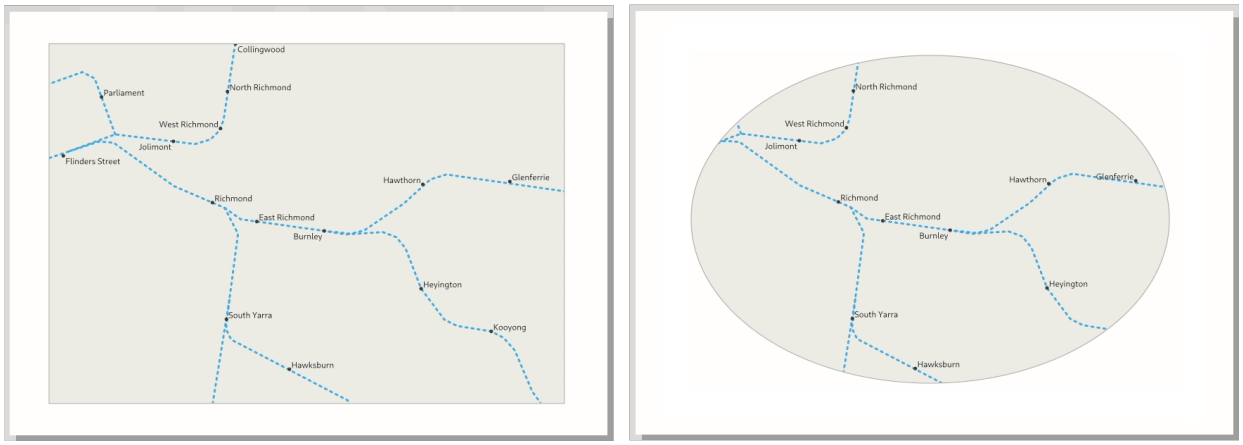



그림 18.15: 조판기 맵 항목을 도형에 맞춰 자르기

- *Draw map canvas items* 를 활성화하면 인쇄 조판에 주 맵 캔버스에 배치된주석 을 표시할 수 있습니다.

### 레이어

기본적으로, 맵 항목의 모습은 맵 캔버스 렌더링과 동기화됩니다. 즉 *Layers Panel* 에서 레이어 가시성을 켜고 끄거나 레이어 스타일을 수정하면 자동적으로 맵 항목에 적용됩니다. 다른 모든 항목과 마찬가지로 인쇄 조판에 맵 항목 여러 개를 추가하기를 원할 수도 있기 때문에, 서로 다른 축척에서 서로 다른 영역, 레이어 조합 등을 표시할 수 있으려면 이 동기화를 끊어야 합니다. *Layers* 속성 그룹에서 (그림 18.16 참조) 동기화를 끊을 수 있습니다.

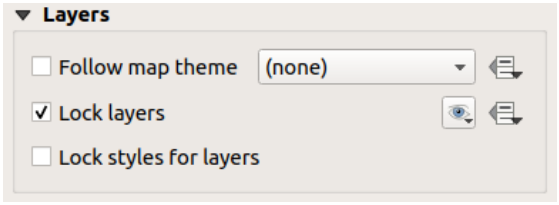



그림 18.16: 맵 레이어 그룹

맵 항목과 기존맵 테마 와의 일관성을 유지하고 싶다면 *Follow map theme* 옵션을 체크한 다음 드롭다운 목록에서 원하는 테마를 선택하십시오. QGIS 메인 창에서 테마를 (테마 대체 기능을 사용해서) 변경하면 자동적으로 맵 항목에 적용할 것입니다. 맵 테마를 선택했다면, *Lock styles for layers* 옵션이 비활성화됩니다. *Follow map theme* 옵션도 레이어 스타일을 (심볼, 라벨, 도표를) 업데이트하기 때문입니다.

맵 항목에 표시된 레이어를 현재 맵 캔버스 가시성 상태로 고정하려면 *Lock layers* 옵션을 체크하십시오. 이 옵션을 활성화하고 나면, QGIS 메인 창에서 레이어 가시성을 어떻게 변경하더라도 조판기의 맵 항목에 영향을 주지 않을 것입니다. 그럼에도 불구하고, 고정한 레이어의 스타일 및 라벨은 QGIS 메인 창의 상태에 따라 업데이트될 것입니다. *Lock styles for layers* 옵션을 사용하면 이를 방지할 수 있습니다.

현재 맵 캔버스를 사용하는 대신, 맵 항목의 레이어를 기존 맵 테마로도 고정할 수 있습니다. Set layer list from a map theme 버튼의 드롭다운 목록에서 맵 테마를 선택하면, *Lock layers* 옵션이 활성화됩니다. 이제 사용자가 또다른 맵 테마를 선택하거나 *Lock layers* 옵션을 체크 해제할 때까지 맵 항목에 맵 테마의 가시 레이어 집합을 적용합니다. 그 다음 *Navigation* 툴바의 Refresh view 버튼 또는 앞에서 설명한 *Update Preview* 버튼을 사용해서 뷰를 새로고침해야 할 수도 있습니다.

*Follow map theme* 옵션과는 달리, *Lock layers* 옵션을 활성화하고 맵 테마로 설정하면 QGIS 메인 창에서 (테마 대체 기능을 통해) 맵 테마를 업데이트하더라도 맵 항목의 레이어가 새로고침되지 않을 것이라는 점을 기억하십시오.

옵션 옆에 있는 아이콘을 이용하면, 맵 항목에서 고정된 레이어도데이터로 정의 할 수 있습니다. 이 아이콘을 이용할 경우, 드롭다운 목록에서 선택한 집합을 무시합니다. 대신 | 문자로 구분된 레이어 목록을 전송해야 합니다. 다음은 맵 항목이 `layer 1` 및 `layer 2` 레이어만 사용하도록 고정하는 예시입니다:

```
concat ('layer 1', '|', 'layer 2')
```

### 범위

맵 항목 속성 패널의 *Extents* 그룹은 다음 기능을 제공하고 있습니다. (그림 18.17 을 참조하세요):

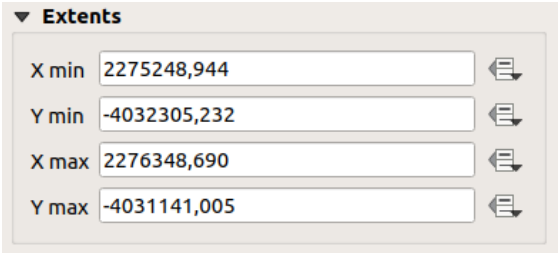



그림 18.17: 맵 범위 그룹

**Extents** 영역은 맵 항목에 표시된 `X` 및 `Y` 좌표를 출력합니다. 이 값들 각각을 직접 대체해서 출력되는 맵 캔버스 영역 그리고/또는 맵 항목 크기를 수정할 수 있습니다. 맵 항목 패널 상단에 있는 다음과 같은 도구들을 사용해서 범위를 수정할 수도 있습니다:

- Set map canvas to match main canvas extent: 맵 캔버스의 범위를 주 캔버스 범위와 일치시킵니다.
- Set map scale to match main canvas scale: 맵의 축척을 주 캔버스 축척과 일치시킵니다.

Move item content 도구로도 맵 항목의 범위를 수정할 수 있습니다. 맵 항목 안을 클릭 & 드래그해서 맵 항목의 현재 뷰를 축척을 유지한 채 변경하십시오. 도구를 활성화한 상태에서 마우스 휠로 확대 또는 축소하면, 표시된 맵의 축척을 수정할 수 있습니다. `Ctrl` 키를 누른 채 마우스를 움직이면 더 세밀하게 확대/축소할 수 있습니다.

## Elevation range

The *Elevation range* setting in the map item properties allows you to filter the content of certain layers based on a specific elevation range. While all layers remain visible, those that support elevation filtering (currently point clouds and raster DEMs) will have their data filtered. Only the parts of these layers that fall within the elevation range set by the *Lower* and *Upper* values will be displayed.

The *Elevation range* can be data-defined. This means you can set different elevation ranges for different features in the *atlas* or *report*.

### 시계열 범위

맵 항목 속성 패널의 *Temporal range* 그룹은 시계열 범위를 기반으로 맵 항목에 있는 레이어의 렌더링을 제어할 수 있는 옵션을 제공합니다. *Start* 및 *End* 날짜가 설정하는 시간 범위와 중첩하는 시계열 속성을 가진 레이어만 맵 항목에 표시합니다.

이와 관련된 데이터 정의 위젯을 이용하면 시계열 범위를 동적으로 만들어 시계열지도책 을 생성할 수 있습니다. 즉 그 내용이 시간을 기반으로 변화하는, 공간 영역이 고정된 자동화 맵을 생성할 수 있습니다. 예를 들면, 시작 및 종단 필드 쌍과 날짜 범위를 나타내는 여러 행을 가진 CSV 파일을 커버리지 레이어로 사용해서 맵 항목 속성에서 시간 범위와 지도책으로 제어를 모두 활성화하고 지도책 내보내기를 클릭합니다.

### 지도책으로 제어

*Controlled by atlas* 그룹 속성은 인쇄 조판기에서지도책 *(atlas)* 을 활성화한 경우에만 사용할 수 있습니다. 지도책으로 맵 항목을 제어하기를 원한다면 이 옵션을 활성화하십시오. 커버리지 레이어에 대해 반복하는 경우, 맵 항목의 범위가 다음 규칙을 따라 지도책 피처에 이동/확대/축소됩니다:

- *Margin around features*: 피처 주위에 맵 항목의 너비 또는 높이의 백분율로 지정한 상하좌우 여백을 유지하면서 피처를 최적의 축척으로 확대/축소합니다. 모든 피처에 대해 동일한 여백을 지정하거나, 또는 예를 들어 맵 축척에 따라변수를 설정 할 수 있습니다.
- *Predefined scale (best fit)*: 프로젝트 수준의사전 정의 축척 에서 지도책 피처가 최적의 크기가 되도록 피처를 확대/축소합니다.
- *Fixed scale*: 맵 항목의 축척을 동일하게 유지하면서 지도책 피처를 차례로 이동합니다. 동일한 크기의 (예: 그리드) 피처를 작업하거나 지도책 피처들의 서로 다른 크기를 강조할 때 적합합니다.

### 그리드

그리드를 사용하면, 맵 항목 투영체 또는 다른 투영체에서 맵 항목의 범위 또는 좌표와 관련된 정보를 사용자 맵 위에 추가할 수 있습니다. *Grids* 그룹에서 맵 항목에 여러 그리드를 추가할 수 있습니다.

- 및 버튼으로 선택한 그리드를 추가 또는 제거할 수 있습니다.
- 및 버튼으로 목록에 있는 그리드를 위아래로 이동시킬 수 있습니다. 즉 맵 항목 위에서 어떤 그리드를 다른 그리드의 위 또는 아래로 배치할 수 있습니다.

추가한 그리드를 재명명하려면 더블클릭하십시오.

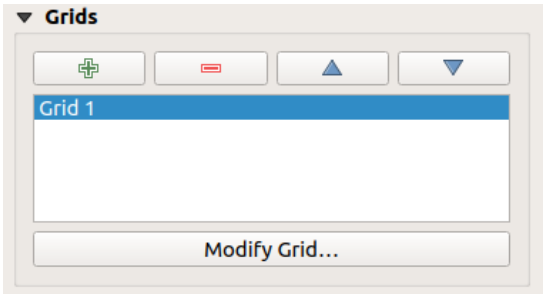



그림 18.18: 맵 그리드 그룹

그리드를 수정하려면, 그리드를 선택한 다음 *Modify Grid*⋯버튼을 클릭해서 *Map Grid Properties* 패널을 여십시오. 이 패널에 그리드 환경 설정 옵션들이 있습니다.

### 그리드 모양

*Map Grid Properties* 패널에서 *Grid enabled* 옵션을 체크하면 맵 항목 위에 그리드를 표시합니다.

다음과 같은 그리드 유형을 사용하도록 지정할 수 있습니다:

- **실선 (Solid)**: 그리드 프레임을 선으로 표시합니다. *Line style* 을색상 과심볼 선택기 위젯을 사용해서 사용자 지정할 수 있습니다.
- **십자 (Cross)**: 그리드 선의 교차 지점에 선분을 표시합니다. *Line style* 과 *Cross width* 를 설정할 수 있습니다.
- **마커 (Markers)**: 그리드 선의 교차 지점에 사용자 지정할 수 있는 마커 심볼만 표시합니다.
- 또는 **프레임과 주석만** 표시할 수도 있습니다.

그리드 유형 외에도, 다음을 정의할 수 있습니다:

- 그리드의 *CRS*: 기본적으로 맵 항목 좌표계를 따릅니다. Select CRS 버튼을 클릭하면 다른 좌표계로 설정할 수 있습니다.
- 그리드 참조에 사용할 *Interval* 유형:
  - **맵 단위 (Map Units)**: *X* 및 *Y* 방향으로 연속하는 그리드 참조 사이의 맵 내부 거리를 (그리드 좌표계 단위로) 설정합니다. 그리드 눈금의 개수는 맵 축척에 따라 달라질 것입니다.
  - **선분 너비 맞추기 (Fit Segment Width)** 를 선택하면 맵 범위를 기반으로"보기 좋은 (pretty)"그리드 간격을 동적으로 선택할 것입니다. *Minimum* 및 *Maximum* 거리 값을 사용자 지정할 수 있는 거리 범위 안에서 이 최적 간격을 계산합니다.
  - **밀리미터 (Millimeters)** 또는 **센티미터 (Centimeters)**: *X* 및 *Y* 방향으로 연속하는 그리드 참조 사이의 용지 상 거리를 설정합니다. 그리드 눈금의 개수는 맵 축척에 상관없이 동일할 것입니다.
- *Offset*: 맵 항목 경계로부터 *X* 그리고/또는 *Y* 방향으로 오프셋 거리를 정의할 수 있습니다.
- *Blend mode*: 그리드에혼합 모드 를 사용할 수 있는 경우, 그리드의 혼합 모드를 정의할 수 있습니다.

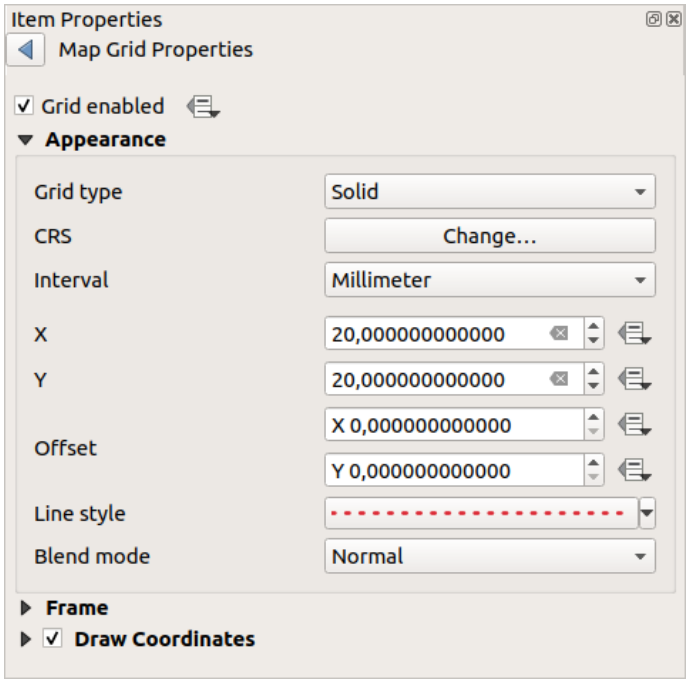



그림 18.19: 그리드 모양 대화창

### 그리드 프레임

맵을 담고 있는 프레임의 스타일을 `No Frame`, `Zebra`, `Zebra (nautical)`, `Interior ticks`, `Exterior ticks`, `Interior and Exterior ticks`, `Line border` 그리고 `Line border (nautical)` 옵션 가운데 하나로 선택할 수 있습니다.

가능한 경우, 연관 색상과 함께 *Frame size*, *Frame margin*, *Frame line thickness* 를 정의할 수 있고, *Frame fill colors* 도 설정할 수 있습니다.

기울인 맵 또는 재투영한 그리드를 작업하는 경우, 구획 (division) 부분의 값을 `Latitude/Y only` 및 `Longitude/X only` 로 설정하면, 그리드 프레임 양쪽에 위도/Y 및 경도/X 좌표가 섞이는 일을 방지할 수 있습니다. 또 그리드 프레임 양쪽에 각각 가시화 또는 비가시화되도록 선택할 수도 있습니다.

맵 항목 영역을 (*Main properties* 그룹에서) 기울였을 때 또는 그리드에 다른 좌표계를 적용시켰을 때, 그리드 선이 맵 항목의 변에 평행 또는 수직이 아니게 될 수도 있습니다. 이런 경우 내부 그리고/또는 외부 눈금 스타일을 적용한 그리드가 보기 나빠질 수 있습니다. ☑ *Follow grid rotation* 옵션을 체크하면 눈금을 그리드 선에 맞춰 정렬시킬 것입니다. 게다가 몇몇 속성들을 더 조정할 수 있습니다:

- *Ticks alignment*: 내부 그리고/또는 외부 눈금을 대응하는 그리드 선에 평행하게 만들 것입니다. 눈금을 다음 가운데 하나로 정렬시킬 수 있습니다:
  - **직교 (Orthogonal)**: 한 선에서 같은 쪽에 있는 눈금이 측면에 평행하게 끝납니다. 이렇게 설정하면 예를 들어 프레임에 작은 각도를 적용했을 때 몇몇 눈금이 더 길어질 수 있습니다.
  - **고정 길이 (Fixed length)**: 모든 눈금이 동일한 길이를 가지기 때문에, 눈금들이 정렬되지 않을 수도 있습니다.
- *Skip below angle*: 지정한 한계값 밑에서 프레임 경계와 교차하는 그리드 선에 눈금을 표시하지 않습니다.
- *Margin from map corner*: 맵 모서리에 너무 가까이 있는 눈금을 표시하지 않습니다. 경계와 중첩 그리고/또는 경계 바깥으로 나갈 수 있기 때문입니다.

### 좌표

☑ *Draw coordinates* 옵션을 활성화하면 맵 프레임에 좌표를 추가할 수 있습니다. 선택한그리드 간격 단위에 관련된 값을 표시합니다. 좌표 서식으로 주석 숫자 서식을 선택할 수 있는데, 이 서식은 실수 좌표인지 도/분/초 좌표인지, 접미사 여부, 정렬 여부 옵션을 가지고 있습니다. 표현식 대화창을 이용한 사용자 지정 서식도 설정할 수 있습니다.

그리드 프레임의 *Left*, *Right*, *Top* 및 *Bottom* 변 각각에 대해 다음을 지정할 수 있습니다:

- 좌표를 렌더링 할지 여부: **전부 보이기 (Show all)**, **위도/Y 만 보이기 (Show latitude/Y only)**, **경도/X 만 보이기 (Show longitude/X only)**, **비활성화 (Disabled)** 가운데 하나로 지정할 수 있습니다. 그리드 프레임의 각 변에 위도/Y 값만 또는 경도/X 값만 보이게 하면 기울인 맵 또는 재투영한 그리드를 작업할 때 양쪽에 위도/Y 및 경도/X 좌표가 섞이는 일을 방지할 수 있습니다.
- 그리드 프레임 기준 텍스트의 상대적 위치: **프레임 바깥 (Outside frame)** 또는 **프레임 안쪽 (Inside frame)** 가운데 하나로 지정할 수 있습니다.
- 주석의 배치 및 방향:
  - **수평 (Horizontal)**
  - **수직 오름차순 (Vertical ascending)**, **수직 내림차순 (Vertical descending)**
  - **경계 방향 (Boundary direction)**
  - 눈금 기반 프레임 사용 시 **눈금 위 (Above tick)**, **눈금에 (On tick)**, **눈금 아래 (Under tick)**

주석을 그리기 위한 *Font* 글꼴 속성 (글꼴, 크기, 색상, 버퍼, ⋯), *Distance to the map frame* 맵 프레임까지의 거리, 그리고 *Coordinate precision* 좌표 정밀도 (소수점 이하 자릿수) 도 정의할 수 있습니다.

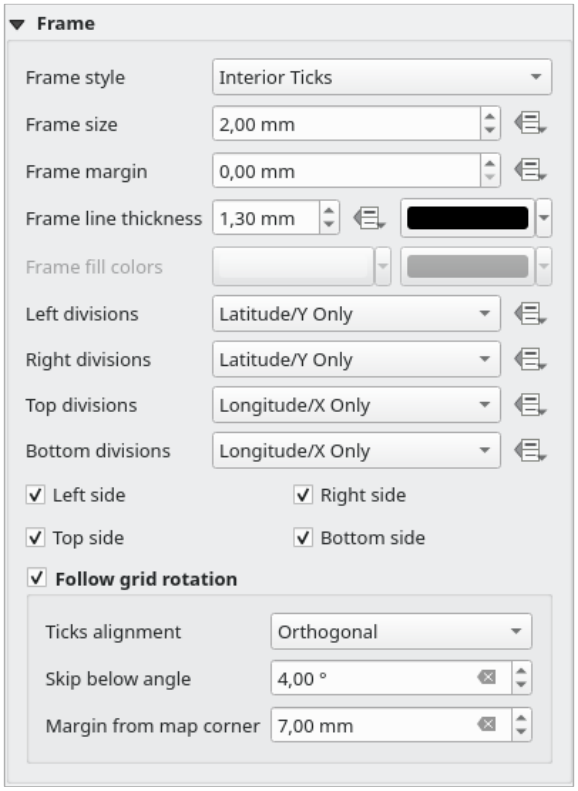



그림 18.20: 그리드 프레임 대화창

*Follow grid rotation*: 맵 영역을 회전시키거나 그리드를 재투영한 경우 사용할 수 있는 이 옵션은 주석의 배치를 조정할 수 있게 해줍니다. 선택한 배치 모드에 따라, 주석도 회전시킵니다:

- *Annotations alignment*: **직교** 또는 **고정 길이** 가운데 하나로 지정할 수 있습니다.
- *Skip below angle*: 지정한 한계값 밑에서 프레임 경계와 교차하는 그리드 선에 주석을 표시하지 않습니다.
- *Margin from map corner*: 맵 모서리에 너무 가까이 있는 주석을 표시하지 않습니다. 경계와 중첩 그리고/또는 경계 바깥으로 나갈 수 있기 때문입니다.

### 오버뷰

종종 인쇄 조판에 하나 이상의 맵을 추가해서, 어떤 맵 항목의 연구 영역을 또다른 맵 항목 위에 배치하고 싶을 때가 있습니다. 예를 들어 맵을 보는 사람들이 두 번째 맵에 표시된 더 넓은 지리적 맥락에서 해당 영역을 식별하기 쉽게 만들기 위해서 말이죠.

맵 항목 패널의 *Overviews* 그룹에서 서로 다른 맵 범위 2 개 사이의 링크를 생성하고, 다음과 같은 기능을 사용할 수 있습니다:

오버뷰를 생성하려면, 해당 맵 위에 다른 맵의 범위를 표시하고자 하는 맵을 선택한 다음 *Item Properties* 패널에서 *Overviews* 옵션을 펼치십시오. 그리고 버튼을 클릭해서 오버뷰를 추가하십시오.

이 오버뷰는 처음에'Overview 1'이라고 명명됩니다. (그림 18.22 를 참조하세요.) 이 그룹에서:

- 더블클릭하면 다른 이름으로 변경할 수 있습니다.
- 및 버튼을 이용해서 오버뷰를 추가하거나 제거할 수 있습니다.

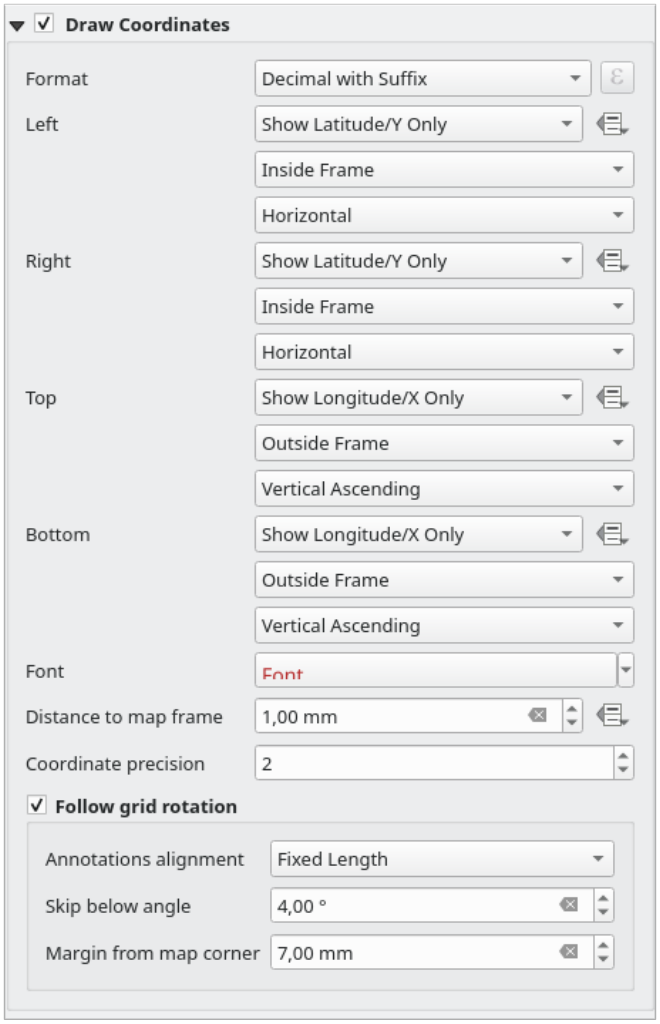



그림 18.21: 그리드 좌표 렌더링 대화창

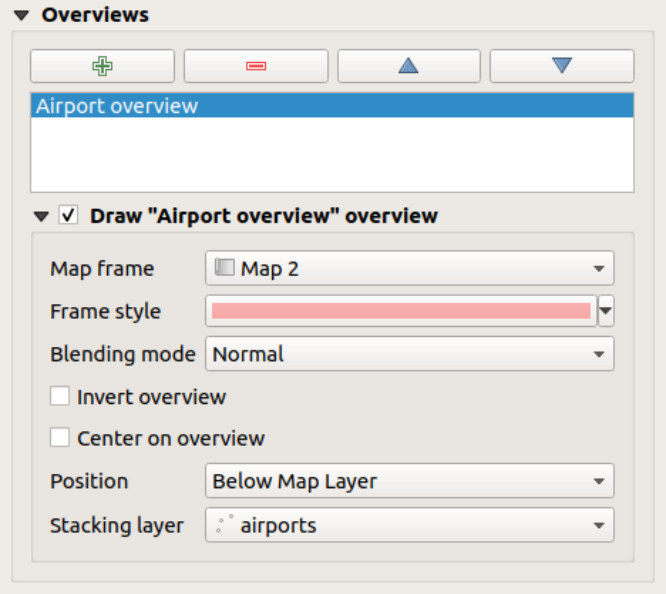



그림 18.22: 맵 오버뷰 그룹

- 및 버튼을 이용하면 목록 안에서 오버뷰를 이동시켜, 맵 항목에서 (두 오버뷰가 동일한스택 위치 *(stack position)* 에 있는 경우) 다른 오버뷰의 위 또는 아래 배치할 수 있습니다.

그 다음 목록에서 오버뷰 항목을 선택하고 *Draw* "*<name_overview>*"*overview* 옵션을 체크하면, 선택한 맵 프레임 상에 오버뷰를 렌더링합니다. 이 오버뷰를 다음과 같이 사용자 지정할 수 있습니다:

- *Map frame*: 현재 맵 항목 위에 범위를 표시할 맵 항목을 선택합니다.
- *Frame Style*: 오버뷰 프레임을 렌더링할심볼 속성 을 설정합니다.
- *Blending mode*: 서로 다른 투명도 혼합 모드를 설정할 수 있습니다.
- *Invert overview* 옵션을 활성화하면 참조되는 맵의 범위 주변에 마스크를 생성합니다. 참조되는 맵의 범위는 명확하게 표시되는 반면, 맵 항목의 다른 부분은 (채우기 색상을 설정했다면) 프레임 채우기 색상과 혼합되어 흐려집니다.
- *Center on overview* 옵션을 활성화하면 맵 항목의 내용물을 이동해서 맵의 중심에 오버뷰 프레임을 표시합니다. 오버뷰를 여러 개 추가한 경우, 1 개의 오버뷰 항목만 중심에 사용할 수 있습니다.
- *Position*: 맵 항목의 레이어 스택 가운데 정확히 어디에 오버뷰를 배치할지를 제어합니다. 예를 들면, 오버뷰 범위를 도로 같은 일부 레이어 아래, 기타 배경 레이어 위로 렌더링하도록 할 수 있습니다. 다음 옵션을 사용할 수 있습니다:
  - *Below map*: 오버뷰를 맵 항목 아래에 렌더링합니다.
  - *Below map layer* 또는 *Above map layer*: 오버뷰 프레임을 어떤 레이어 도형의 상대적인 아래 또는 위에 렌더링합니다. *Stacking layer* 옵션에서 해당 레이어를 선택할 수 있습니다.
  - *Below map labels*: 라벨이 언제나 맵 항목에 있는 모든 피처 도형 위에 렌더링된다는 점을 고려하면, 오버뷰 프레임을 모든 라벨 아래, 모든 도형 위로 배치합니다.
  - *Above map labels*: 오버뷰 프레임을 맵 항목에 있는 모든 도형과 라벨 위로 배치합니다.

### 18.2.3 3 차원 맵 항목

3 차원 맵 항목을 사용해서*3 차원 맵 뷰* 를 표시합니다. *Add 3D Map* 버튼을 클릭하고항목 생성 지침 을 따라 새 3 차원 맵 항목을 추가하십시오. 3 차원 맵 항목을 추가한 후대화형 조판기 항목 작업 에서 설명한 내용과 동일한 방법으로 수정할 수 있습니다.

새 3 차원 맵 항목은 기본적으로 비어 있습니다. *Item Properties* 패널에서 3 차원 맵 뷰의 속성을 설정하고 사용자 지정할 수 있습니다. 이 항목은공통 속성 이외에도 다음과 같은 기능을 가지고 있습니다 (그림 18.23 참조):

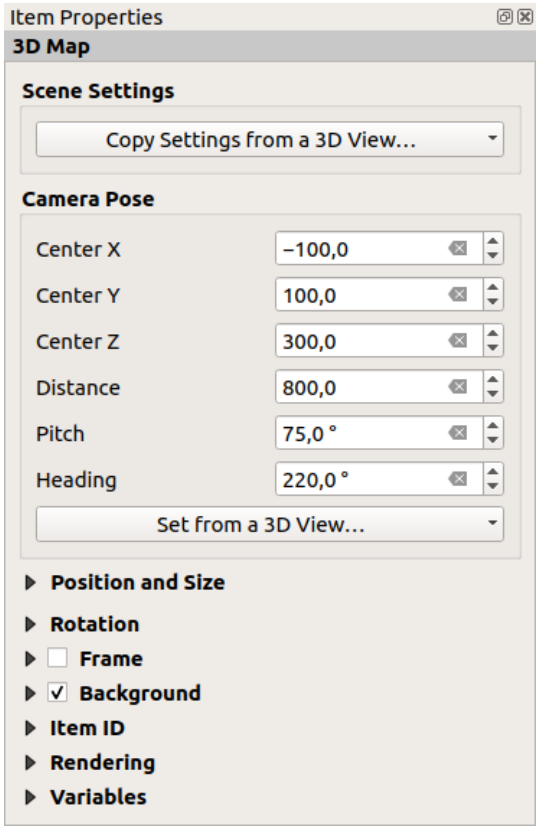



그림 18.23: 3 차원 맵 항목 속성

#### 신 (Scene) 설정

*Copy Settings from a 3D View*…버튼을 클릭해서 표시할 3 차원 맵 뷰를 선택하십시오.

선택한 3 차원 맵 뷰를 현재 환경 설정 (레이어, 지형, 조명, 카메라 위치 및 각도 등등) 으로 렌더링합니다.

#### 카메라 포즈

- *Center X*: 카메라가 바라보는 지점의 X 좌표를 설정합니다.
- *Center Y*: 카메라가 바라보는 지점의 Y 좌표를 설정합니다.
- *Center Z*: 카메라가 바라보는 지점의 Z 좌표를 설정합니다.
- *Distance*: 카메라 중심에서 카메라가 바라보는 지점까지의 거리를 설정합니다.
- *Pitch*: X 축 주위로 (수직 회전) 카메라의 회전 각도를 설정합니다. 0° 에서 360° 사이의 값을 지정할 수 있습니다. 0°: 지형을 바로 위에서 바라봅니다. 90°: 지형을 옆 방향에서 수평으로 바라봅니다. 180°: 바로 아래에서 바라봅니다. 270°: 위아래를 뒤집어 수평으로 바라봅니다. 360°: 바로 위에서 바라봅니다.
- *Heading*: Y 축 주위로 (수평 회전) 카메라의 회전 각도를 설정합니다. 0° 에서 360° 사이의 값을 지정할 수 있습니다. 0°/360°: 북쪽에서 바라봅니다. 90°: 서쪽에서 바라봅니다. 180°: 남쪽에서 바라봅니다. 270°: 동쪽에서 바라봅니다.

*Set from a 3D View*…풀다운 메뉴에서 3 차원 맵 항목을 3 차원 뷰의 파라미터로 채울 수 있습니다.

### 18.2.4 라벨 항목

*Label* 항목은 사용자의 맵을 이해할 수 있도록 돕는 텍스트로 장식할 수 있는 도구입니다. 라벨은 제목, 작성자, 데이터소스 또는 기타 모든 정보를 담을 수 있습니다. 항목 생성 지침 을 따라 *Add Label* 도구를 이용해서 라벨을 추가하고대화형 조판기 항목 작업 에서 설명한 내용과 동일한 방법으로 수정할 수 있습니다.

라벨 항목은 기본적으로 라벨의 *Item Properties* 패널에서 사용자 지정할 수 있는 기본 텍스트를 제공합니다. 이 항목은공통 속성 이외에도 다음과 같은 기능을 가지고 있습니다 (그림 18.24 참조):

#### 주요 속성

*Main properties* 그룹이 라벨 텍스트를 제공하는 장소입니다. 텍스트는 정적 텍스트일 수도, 표현식 함수 및 변수를 사용하는 동적 텍스트일 수도, 그리고/또는 HTML 서식일 수도 있습니다. 라벨의 동적 부분을 동적 텍스트로 해석하고 평가하려면 `[%` 와 `%]` 로 감싸야 합니다.

- 라벨에서 표현식을 사용하려면 *Insert/Edit Expression*…버튼을 클릭한 다음 평소와 같이 공식을 작성하면 됩니다. 그리고 대화창을 적용하면, QGIS 가 감싸는 괄호 문자들을 자동으로 추가합니다.

  **힌트**

  텍스트 상자에서 아무것도 선택하지 않은 채 *Insert/Edit Expression*…버튼을 클릭하면 기존 텍스트에 새 표현식이 추가될 것입니다. 하지만 기존 표현식을 수정하고 싶은 경우 사용자가 무엇을 선택했는지에 따라 이 습성은 변합니다: 표현식 안에 있는 ('[%'와'%]'사이에 있는) 텍스트 일부를 선택한 경우 또는 어떤 텍스트도 선택하지 않았지만 커서가 표현식 내부에 있는 경우, 표현식 전체가 선택될 것입니다. 다른 모든 경우에는 사용자가 선택한 것이 그대로 유지될 것입니다.

  일반적으로 맵은 몇몇 공통적인 텍스트 정보 (날짜, 제작자, 제목, 페이지 번호 등등) 로 채워지기 때문에, QGIS 에서는 대응하는 표현식 또는 변수에 직접 접근할 수 있습니다: 이런 표현식 또는 변수를 선택해서 사용자 라벨에 삽입하려면 *Dynamic text* 버튼을 누르십시오.

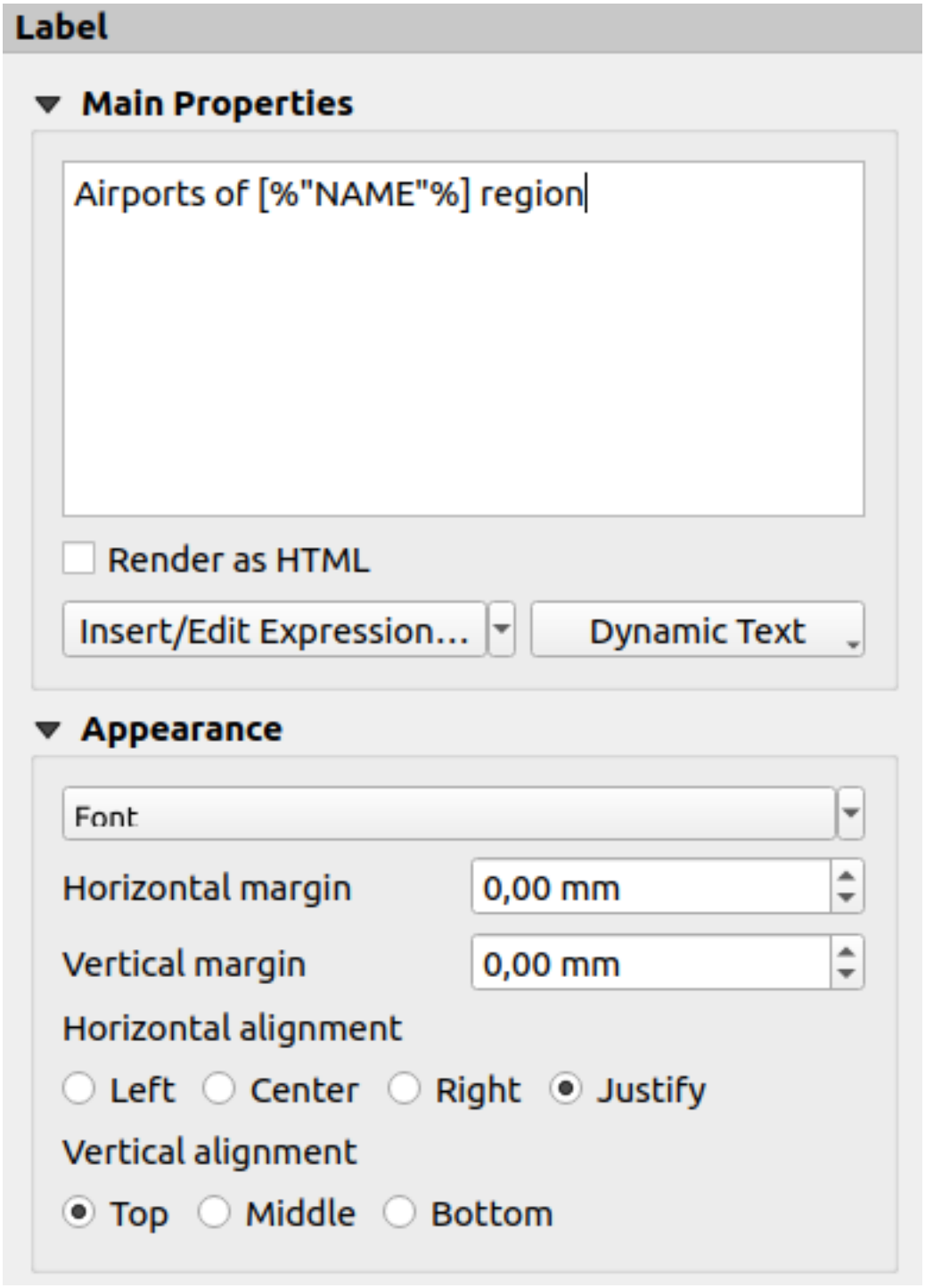



그림 18.24: 라벨 항목 속성 패널

> **팁**
>
> 최상위 메뉴 *Add Item* → *Add Dynamic Text* → 를 사용하면 선택한 사전 정의 표현식으로 채워진 새 라벨을 생성할 수 있습니다.

동적 텍스트를 정적 텍스트로 변환할 수도 있습니다: *Insert/Edit Expression*…버튼 옆에 있는 드롭다운 화살표를 눌러서 *Convert to Static* 를 선택하십시오. 라벨 내용 가운데 모든 동적 부분을 평가해서 현재 값으로 대체할 것입니다. 그 다음 필요한 경우 산출 텍스트를 직접 조정할 수 있습니다.

- 라벨을 HTML 코드로 해석할 수 있습니다: *Render as HTML* 체크박스를 체크하십시오. 이제 HTML 태그 또는 스타일, URL, 웹페이지를 가리키는 링크를 가진 클릭할 수 있는 이미지, 또는 좀 더 복잡한 HTML 코드 등등을 삽입할 수 있습니다.

다음 코드는 고급 라벨 작업을 위해 HTML 렌더링과 표현식을 결합하고 그림 18.25 를 산출할 것입니다:

```
<html>
 <head>
   <style>
      /* Define some custom styles, with attribute-based size */
      name {color:red; font-size: [% ID %]px; font-family: Verdana; text-shadow: grey 1px 0 10px;}
      use {color:blue;}
   </style>
 </head>

 <body>
   <!-- Information to display -->
   <u>Feature Information</u>
   <ul style="list-style-type:disc">
     <li>Feature Id: [% ID %]</li>
     <li>Airport: <name>[% NAME %]</name></li>
     <li>Main use: <use>[% USE %]</use></li>
   </ul>
   Last check: [% concat( format_date( "control_date", 'yyyy-MM-dd'), ' by <b><i>', @user_full_name, '</i></b>' ) %]

   <!-- Insert an image -->
   <p align=center><img src="path/to/logos/qgis-logo-made-with-color.svg" alt="QGIS icon" style="width:80px;height:50px;"</p>
 </body>
</html>
```




그림 18.25: HTML 스타일 작업으로 라벨에 영향 주기

### 모양

- *Font* 버튼을 눌러 텍스트의 글꼴과 스타일을 정의하십시오. *Label Font* 메뉴에서라벨 텍스트 서식 설정 에 대한 몇몇 설정을 사용할 수 있습니다.
- 서로 다른 수직 및 수평 여백을 mm 단위로 지정할 수 있습니다. 조판 항목의 경계로부터의 여백입니다. 라벨 경계 바깥쪽에 — 예를 들어 라벨 항목을 다른 항목과 정렬시키기 위해 — 라벨을 배치할 수 있습니다. 이 경우 여백에 음의 값을 사용해야 합니다.
- 텍스트 *alignment* 옵션은 사용자 라벨을 배치하기 위한 또다른 방법입니다. 다음 가운데 하나를 선택할 수 있습니다:
  - *Horizontal alignment* 의 경우 *Left*, *Center*, *Right* 또는 *Justify*
  - *Vertical alignment* 의 경우 *Top*, *Vertical Center*, *Bottom*

**팁**

**Resize the layout label item to fit its contents**

By default, the size of a layout label item does not depend on the contained text. It may be too large with useless blank space or too small, hiding part of the text. To automatically resize a label so it fits the text inside it perfectly, double-click one of the square handles on the label. The label will adjust its size to match the text, either expanding or shrinking as needed, while the opposite handle of the label remains fixed in place. This allows you to quickly ensure the label fits the text without manually resizing it. So e.g., double-clicking the left-middle handle causes the left side of the label to move.

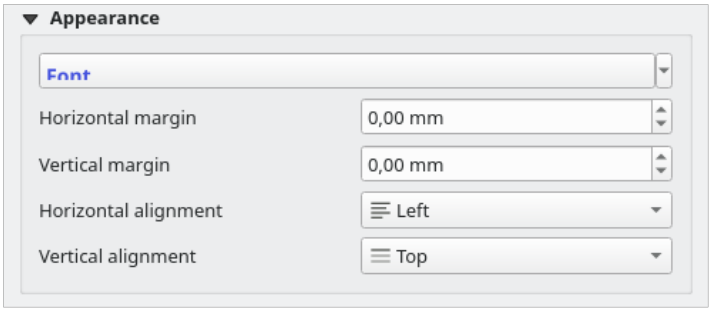



그림 18.26: 라벨 항목 모양 속성

### 라벨 항목에서의 표현식 탐구

다음은 사용자가 라벨 항목을 흥미로운 정보로 채우는 데 사용할 수 있는 몇몇 표현식 예시입니다. *Main properties* 프레임에서 코드를, 또는 적어도 계산식 부분을 `[%` 와 `%]` 로 둘러싸야 한다는 점을 기억하십시오.

- 제목을"field1"에 있는 현재 지도책 피처값과 함께 표시합니다:

```
'This is the map for ' || "field1"
```

  또는 *Main properties* 부분에서 작성하면:

```
This is the map for [% "field1" %]
```

- 공간 처리된 지도책 피처용 페이지 번호를 (예: `Page 1/10`) 추가합니다:

```
concat( 'Page ', @atlas_featurenumber, '/', @atlas_totalfeatures )
```

- 공항의 공통 속성을 기반으로 현재 지도책 지역 피처의 공항 이름을 반환합니다:

```
aggregate( layer := 'airports',
           aggregate := 'concatenate',
           expression := "NAME",
           filter := fk_regionId = attribute( @atlas_feature, 'ID' ),
           concatenator := ', '
         )
```

  또는, 속성 관계 를 설정한 경우:

```
relation_aggregate( relation := 'airports_in_region_relation',
                    aggregate := 'concatenate',
                    expression := "NAME",
                    concatenator := ', '
                  )
```

- 공항의 공간 관계를 기반으로 현재 현재 지도책 지역 피처가 담고 있는 공항 이름을 반환합니다:

```
aggregate( layer := 'airports',
           aggregate := 'concatenate',
           expression := "NAME",
           filter := contains( geometry( @parent ), $geometry ),
           concatenator := ', '
         )
```

  또는:

```
array_to_string( array:= overlay_contains( layer := 'airports',
                                           expression := "NAME" ),
                 delimiter:= ', '
               )
```

- `Map 1` 항목 범위의 하단 X 좌표를 반환합니다:

```
x_min( map_get( item_variables( 'Map 1' ), 'map_extent' ) )
```

- 현재 조판기 `Map 1` 항목에 있는 레이어명들을 추출해서, 한 줄 서식으로 표시합니다:

```
array_to_string(
 array_foreach(
  map_get( item_variables( 'Map 1' ), 'map_layers' ), -- retrieve the layers list
  layer_property( @element, 'name' ) -- retrieve each layer name
 ),
 '\n' -- converts the list to string separated by breaklines
)
```

- 조판 `Map 1` 항목에 레이어 목록을 사용 허가 문자열 (사용 권한) 과 함께 표시합니다. 먼저 레이어의메타데이터 접근 속성을 채워야 합니다.

```
array_to_string( map_credits( 'Map 1', true ) )
```

### 18.2.5 범례 항목

*Legend* 항목은 맵 상에서 사용되는 심볼의 의미를 설명하는 상자 또는 표입니다. 범례는 맵 항목에 귀속됩니다. 항목 생성 지침 을 따라 *Add Legend* 도구로 범례를 추가하고대화형 조판기 항목 작업 에서 설명한 내용과 동일한 방법으로 수정할 수 있습니다.

범례 항목은 기본적으로 범례의 *Item Properties* 패널에서 미세 조정할 수 있는 사용 가능한 모든 레이어를 표시합니다. 이 항목은공통 속성 이외에도 다음과 같은 기능을 가지고 있습니다 (그림 18.27 참조):

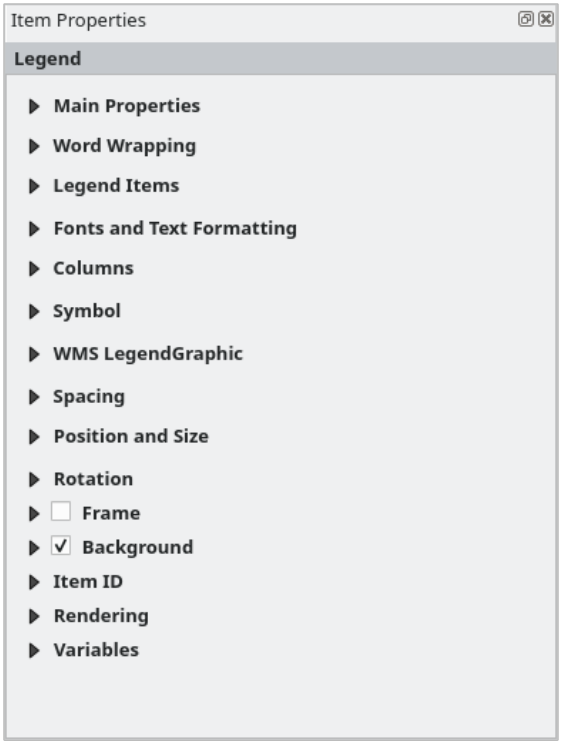



그림 18.27: 범례 항목 속성 패널

#### 주요 속성

범례 *Item Properties* 패널의 *Main properties* 그룹은 다음 기능을 제공하고 있습니다. (그림 18.28 을 참조하세요.):

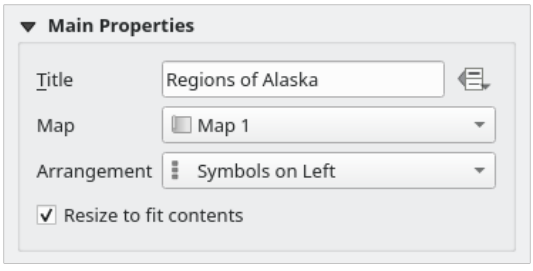



그림 18.28: 범례 주요 속성 그룹

주요 속성 그룹에서 다음 작업을 할 수 있습니다:

- 범례 *Title* 을 변경할 수 있습니다. 데이터 정의 무시 설정을 이용하면 동적인 제목을 만들 수 있습니다. 예를 들자면 지도책을 생성하는 경우 유용한 기능이죠.
- 현재 범례가 어떤 *Map* 항목을 참조할지 선택할 수 있습니다. 범례 항목을 렌더링할 맵이 기본적으로 선택됩니다. 기본 선택된 맵이 없다면, 참조 맵 으로 돌아가게 됩니다.

> **참고**
>
> 범례의 데이터 정의된 속성에서 연결된 맵 항목의 (@map_id, @map_scale, @map_extent 등의) 변수 에도 접근할 수 있습니다.

- 범례에 심볼과 텍스트를 어떻게 배치할지 설정할 수 있습니다: *Arrangement* 옵션을 *Symbols on left* 또는 *Symbols on right* 가운데 하나로 선택할 수 있습니다. 사용 중인 로케일에 따라 (우에서 좌로 기반인지 또는 그 반대인지에 따라) 기본값이 달라집니다.
- ☑ *Resize to fit contents* 옵션을 활성화하면 범례 크기를 그 내용물에 맞춰 자동적으로 조정합니다. 체크를 해제한 경우, 사용자가 설정한 크기로 고정되어 크기가 변하지 않을 것입니다. 내용물이 범례 크기에 맞지 않는 경우 잘려나가게 됩니다.

### Word wrapping

The *Word wrapping* group of the legend *Item Properties* panel provides the following functionalities:

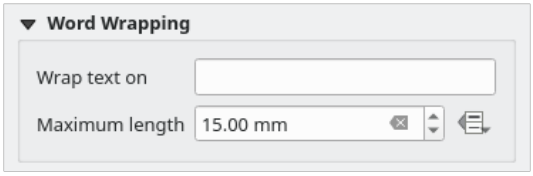



그림 18.29: Legend's word wrapping group

- 범례의 텍스트에 행갈이 문자를 설정할 수 있습니다. 해당 문자가 나타날 때마다 행을 바꿀 것입니다.
- Allow legend text to be automatically wrapped after a set line length (in millimeters), preventing very wide auto-generated legends. The *Maximum length* can be set as a static or data-defined value, convenient for dynamic layouts which adjust legend size based on for example page orientation or displayed features, or atlases where you may want to tweak the legend appearance on different pages.

  **Examples**:

  – Control legend width based on layout page orientation

  ```
  IF( @layout_pageheight >= @layout_pagewidth, 35, 80 )
  ```

  – Set the legend column width when a specific layer is displayed in the linked map

  ```
  CASE WHEN
    array_contains(
      map_get( item_variables('legend_1'), 'map_layer_ids' ),
      'a_specific_layer_id' )
  THEN 60
  END
  ```

### 범례 항목

범례 *Item Properties* 패널의 *Legend items* 그룹은 다음 기능을 제공하고 있습니다. (그림 18.30 을 참조하세요.):

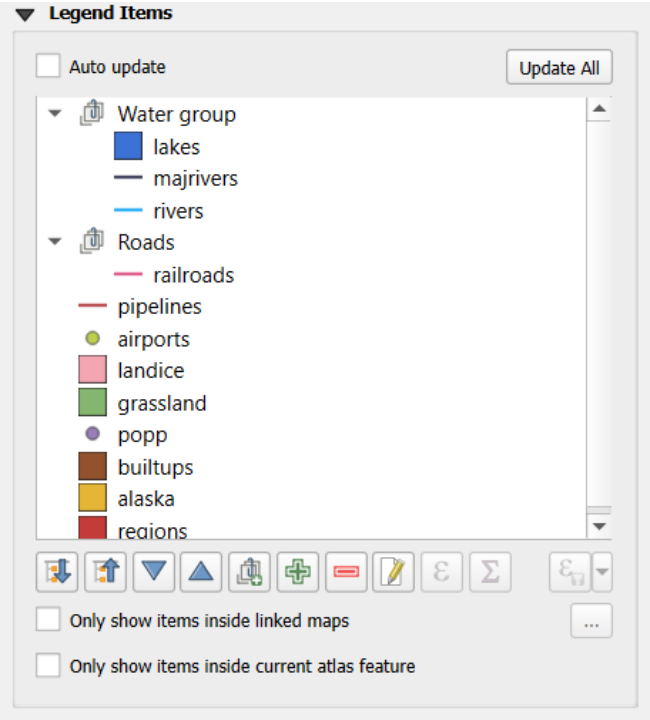



그림 18.30: 범례 항목 그룹

- *Auto update* 옵션을 활성화한 경우 범례를 자동적으로 업데이트할 것입니다. *Auto update* 옵션을 비활성화하면 범례 항목을 좀 더 사용자 마음대로 제어할 수 있습니다. 이 경우 범례 항목 목록 아래에 있는 아이콘들이 모두 활성화될 것입니다.
- 범례 항목 창은 모든 범례 항목의 목록을 담고 있으며, 항목 순서를 변경하고, 레이어를 그룹화하고, 목록에 있는 항목들을 제거 및 복구하고, 레이어 이름과 심볼을 편집하고, 필터를 추가할 수 있습니다.
  - Use the Expand all and Collapse all buttons to respectively expand or collapse all the groups and subgroups in the legend tree. Ensure that the *Auto update* is unchecked to use these buttons.
  - 및 버튼을 사용하거나'드래그 & 드롭'기능을 통해 항목 순서를 변경할 수 있습니다. 다만 WMS LegendGraphic 의 경우 순서를 변경할 수 없습니다.
  - 범례 그룹을 추가하려면 아이콘을 클릭하십시오.
  - 레이어를 추가하려면 아이콘을, 그리고 그룹, 레이어, 또는 심볼 범주를 제거하려면 아이콘을 사용하십시오.
  - 레이어, 그룹명, 또는 제목을 편집하려면, 먼저 범례 항목을 선택한 다음 아이콘을 클릭하십시오. 항목을 더블클릭해도 재명명할 수 있는 텍스트란이 열립니다.
  - 버튼을 누르면 표현식을 사용해서 선택한 레이어의 심볼 라벨을 각각 사용자 정의할 수 있습니다. (범례 라벨 데이터 정의하기 참조)
  - 아이콘을 클릭하면 벡터 레이어의 각 범주의 객체 개수의 합계를 표시합니다.

- Filter legend by expression 아이콘을 클릭하면, 범례에 레이어의 범례 항목 가운데 어떤 항목을 표시할지 필터링할 수 있습니다. 예를 들면 (규칙 기반 또는 범주 심볼에서 나온) 서로 다른 범례 항목을 가진 레이어를 사용하는 경우, 범례 트리에서 조건을 만족하는 피처가 없는 스타일을 제거하는 불 (boolean) 표현식을 지정할 수 있습니다. 다만 이런 경우에도 해당 피처는 조판기 맵 항목에 유지되고 표시된다는 사실을 기억하십시오.

범례 항목의 기본 습성이 *Layers* 패널 트리를 흉내내어 동일한 그룹, 레이어 및 심볼의 범주를 표시하기는 해도, 어떤 항목을 오른쪽 클릭하면 레이어명을 숨기거나 레이어를 그룹 또는 하위 그룹으로 옮길 수 있는 옵션을 제공합니다. 레이어에 몇몇 변경 사항을 만든 경우, 범례 항목의 컨텍스트 메뉴에서 *Reset to defaults* 액션을 선택하면 변경 사항들을 되돌릴 수 있습니다.

QGIS 메인 창에서 심볼을 변경한 다음, *Update All* 버튼을 클릭하면 인쇄 조판기의 범례 항목에 해당 변경 사항을 적용할 수 있습니다.

- *Only show items inside linked map* 옵션을 활성화하면, 링크된 맵에 가시화된 범례 항목들만 범례 목록에 표시될 것입니다. 맵이 하나 이상일 경우 …을 클릭해서 사용자 조판으로부터 다른 맵들을 선택할 수 있습니다. 이 도구는 *Auto-update* 옵션을 활성화한 상태에서만 사용할 수 있습니다.
- 폴리곤 객체를 보유한 지도책을 생성하는 동안, 현재 지도책 객체 바깥쪽에 놓인 범례 항목들을 필터링할 수 있습니다. *Only show items inside current atlas feature* 옵션을 체크하십시오.

### 범례 라벨 데이터 정의하기

도구는 지정한 레이어의 각 심볼 라벨에표현식 을 추가할 수 있습니다. 새 변수들 (`@symbol_label`, `@symbol_id` 및 `@symbol_count`) 로 범례 항목과 대화형 작업을 할 수 있습니다.

예를 들어 `type` 필드로 범주화한 `regions` 레이어를 지정하면, 범례에 있는 각 범주 단계에 `Borough (3) - 850ha` 와 같이 각 단계의 피처 개수 및 총면적을 추가할 수 있습니다.

1. 범례 트리에서 레이어 항목을 선택하십시오.
2. 버튼을 클릭해서 *Expression String Builder* 대화창을 여십시오.
3. 다음 표현식을 입력하십시오 (심볼 라벨은 편집되지 않았다고 가정합니다):

```
format( '%1 (%2) - %3ha',
        @symbol_label,
        @symbol_count,
        round( aggregate(@layer, 'sum', $area, filter:= "type"=@symbol_label)/
↪10000 )
      )
```

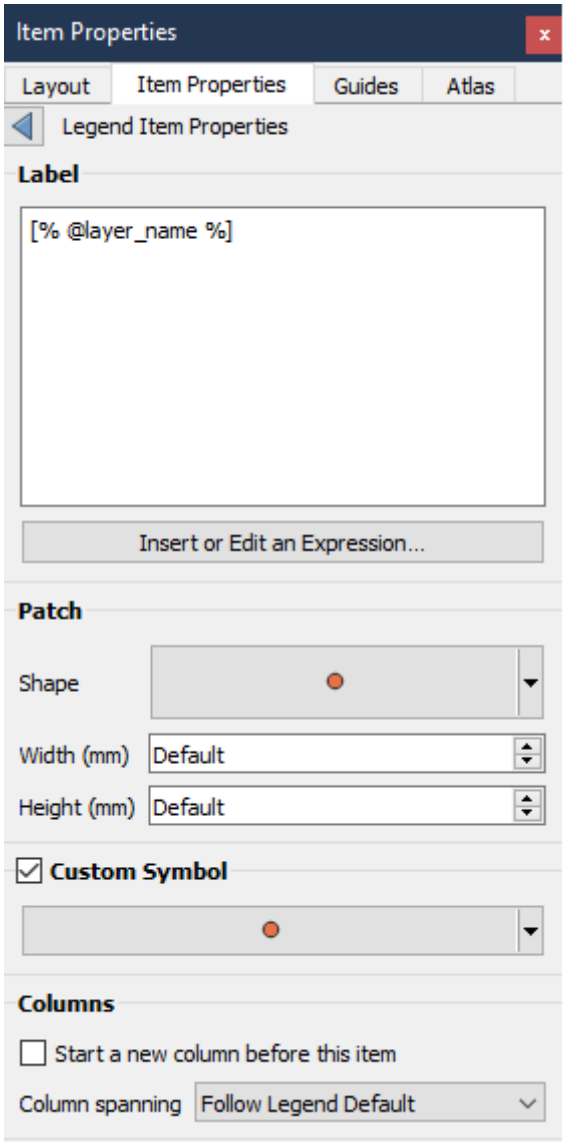
4. *OK* 를 클릭하십시오.

### 범례 항목 사용자 지정하기

*Legend Items Properties* 에서 범례 항목들도 개별적으로 사용자 지정할 수 있습니다. 그러나 이런 사용자 지정은 *Auto update* 를 비활성화한 상태에서만 가능합니다.

항목을 더블클릭하거나 Edit selected item properties 아이콘을 클릭하면 사용자 지정 작업을 더 세밀하게 할 수 있습니다.

*Label*


Item Properties
Layout
Item Properties
Guides
Atlas
Legend Item Properties
Label
[% @layer_name %]
Insert or Edit an Expression...
Patch
Shape
Width (mm)
Default
Height (mm)
Default
Custom Symbol
Columns
Start a new column before this item
Column spanning
Follow Legend Default

모든 항목 유형에 대해, *Insert or Edit an Expression* 을 사용해서 표현식을 입력하거나 삽입해서 라벨 텍스트를 수정할 수 있습니다. 항목 라벨의 어디에든 `[% 표현식%]` 표기법을 이용해서 표현식을 추가할 수도 있습니다.

*Columns*

*Legend Items Properties* 에서 레이어의 특정 항목 또는 모든 심볼 뒤에 열을 강제로 분할하도록 열 분할 습성을 제어할 수도 있습니다. 이 위젯에서는 레이어의 자동 분할과 그 하위 레이어 생성을 레이어 별로 허용하거나 막을 수도 있습니다.

*Patch*

심볼을 가진 항목의 경우, *Legend Items Properties* 에서 심볼이 차지할 수 있는 최대 높이와 너비를 지정할 수 있습니다.

벡터 심볼의 경우, 심볼에 대해 사용자 지정 형태를 지정할 수 있습니다. 일반적으로 단순 평면에 도형을 표현하는 표현식이 이런 형태를 정의하지만, 스타일 관리자에 이런 형태를 저장하고 나중에 가져올 수도 있습니다. 각 도형 유형의 기본 심볼 또한 스타일 관리자를 통해 제어할 수 있습니다.

*Custom Symbol*

벡터 심볼의 경우 사용자 지정 심볼도 지정할 수 있습니다. 범례에서 심볼을 강조하거나 실제 심볼 미리보기에서 심볼을 독립적으로 표시하기 위해 특정 심볼의 렌더링을 미세 조정하는 데 사용자 지정 심볼이 유용할 수 있습니다. 이런 사용자 지정 심볼은 범례 심볼을 무시하지만, 지정한 심볼 *Patch* 를 고려할 것입니다.

### 글꼴 및 텍스트 서식 설정

범례 *Item Properties* 패널의 *Fonts and text formatting* 그룹은 다음 기능을 제공하고 있습니다:

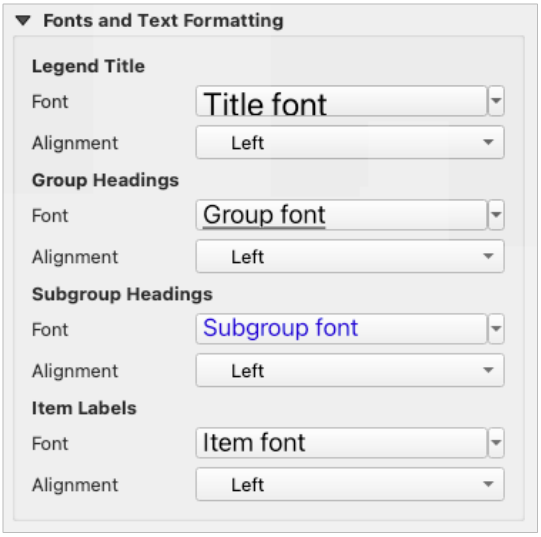



그림 18.31: 범례 글꼴 속성

- 범례 항목에 있는 범례 제목, 그룹, 하위 그룹 및 항목 (피처) 의 글꼴을글꼴 선택기 위젯을 사용해서 변경할 수 있습니다. 이 위젯은텍스트 서식 설정 (글꼴 간격 제어, 혼합 HTML 서식, 색상, 혼합 모드, 배경, 텍스트 버퍼, 음영 등등) 의 모든 기능을 제공합니다.
- 이 각 수준에서 텍스트 *Alignment* 방식을 *Left* (좌에서 우로 기반 로케일의 기본값), *Center* 또는 *Right* (우에서 좌로 기반 로케일의 기본값) 가운데 하나로 설정할 수 있습니다.

### 열

범례 *Item Properties* 패널의 *Columns* 그룹에서 범례 항목을 여러 열에 걸쳐 배열할 수 있습니다:

- *Count* 1,00 란에서 열 개수를 설정하십시오. 지도책 피처, 범례 내용물, 프레임 크기 등에 따라 이 값을 동적으로 설정할 수 있습니다.
- ☑ *Equal column widths* 옵션은 범례 열을 어떻게 조정해야 할지를 설정합니다.
- ☑ *Split layers* 옵션을 활성화하면 범주 또는 등급 레이어 범례를 여러 열에 나눌 수 있습니다.

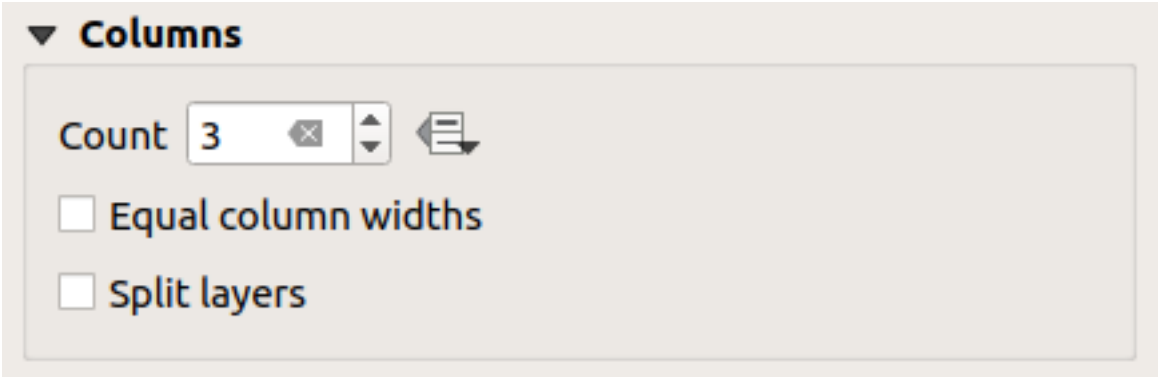



그림 18.32: 범례 열 설정

### 심볼

범례 *Item Properties* 패널의 *Symbol* 그룹에서 범례 라벨 옆에 표시되는 심볼의 크기를 환경 설정할 수 있습니다:

- *Symbol width* 및 *Symbol height*: 심볼의 너비 및 높이를 설정할 수 있습니다.
- 마커의 *Min symbol size* 및 *Max symbol size*: 마커 심볼의 최소/최대 크기를 설정할 수 있습니다. `0.00mm` 로 설정하면 설정값이 없다는 의미가 됩니다.
- ☑ *Draw stroke for raster symbols*: 래스터 레이어의 밴드 색상을 나타내는 심볼에 윤곽선 (outline) 을 추가할 수 있습니다. *Stroke color* 획의 색상과 *Tickness* 두께 둘 다 설정할 수 있습니다.

### WMS LegendGraphic

범례 *Item Properties* 패널의 *WMS LegendGraphic* 그룹은 다음 기능을 제공하고 있습니다. (그림 18.34 를 참조하세요.):

WMS 레이어를 추가한 다음 범례 항목을 삽입한 경우, WMS 서버에 WMS 범례를 제공하라는 요청을 전송할 것입니다. WMS 서버가 GetLegendGraphic 기능을 제공하는 경우에만 이 범례가 표시될 것입니다. WMS 범례 내용은 래스터 이미지로 제공될 것입니다.

*WMS LegendGraphic* 옵션에서 *Legend width* 및 *Legend height* 를 설정해서 WMS 범례 래스터 이미지의 너비 및 높이를 조정할 수 있습니다.

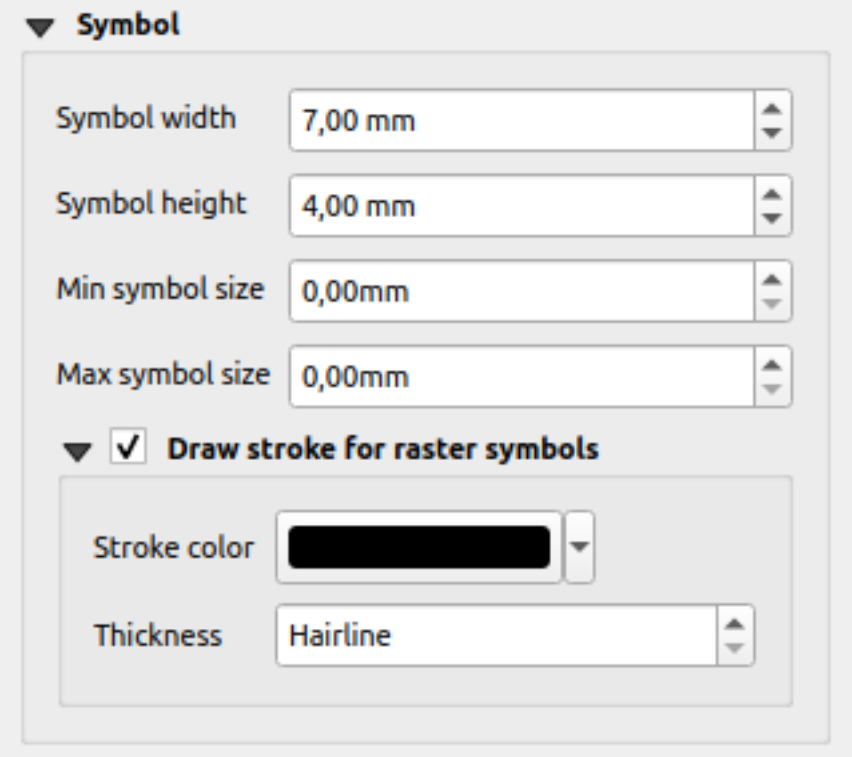



그림 18.33: 범례 심볼 환경 설정

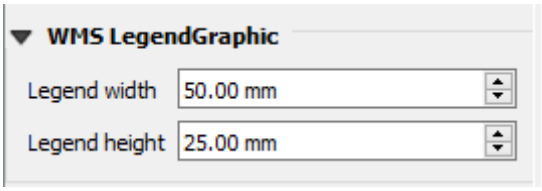
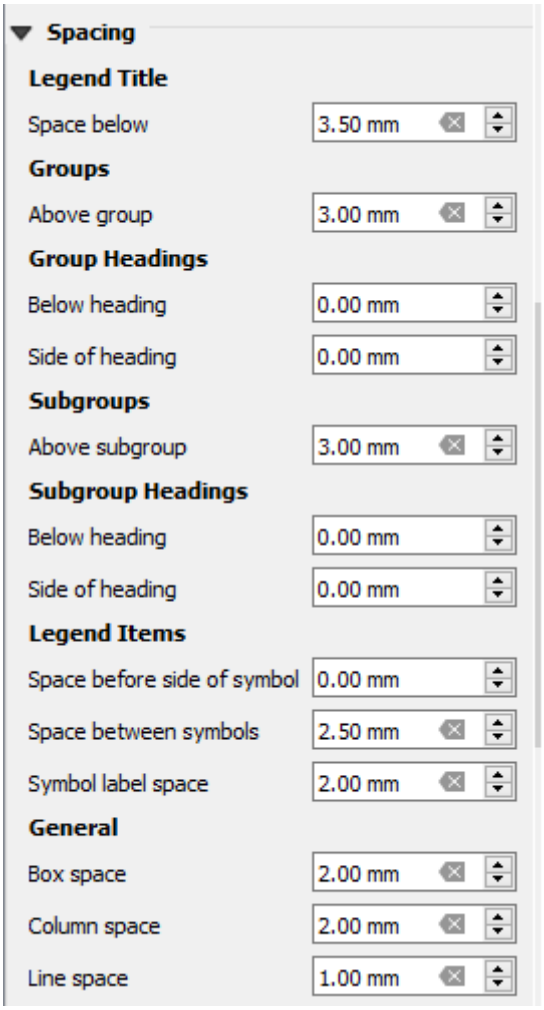



그림 18.34: WMS LegendGraphic

간격




*Spacing* 그룹은 범례 안에서의 간격을 사용자 지정할 수 있게 해줍니다. 간격은 범례에 있는 항목들의 그룹화와 그 관계를 나타내는 데 큰 도움을 줄 수 있습니다.

이 대화창에서 제목, 그룹, 하위 그룹, 심볼, 라벨, 입력란, 열 및 선의 앞 또는 주위 간격을 사용자 지정할 수 있습니다.

### 18.2.6 축척 막대 항목

축척 막대 항목은 맵 항목 상에 있는 피처의 크기와 피처들 사이의 거리를 시각적으로 나타냅니다. 축척 막대 항목을 사용하려면 맵 항목이 필요합니다. *Add Scale Bar* 도구를항목 생성 지침 을 따라 사용해서 새 축척 막대 항목을 추가하십시오. 이후대화형 조판기 항목 작업 에서 설명한 내용과 동일한 방법으로 새 맵 항목을 수정할 수 있습니다.

새 축척 막대 항목은 기본적으로 해당 항목이 그려지는 맵 항목의 축척을 표시합니다. 축척 막대를 그릴 맵 항목이 없는 경우, 참조 맵 을 사용합니다. *Item Properties* 패널에서 축척 막대를 사용자 지정할 수 있습니다. 이 항목은공통 속성 이외에도 다음과 같은 기능을 가지고 있습니다 (그림 18.35 참조):

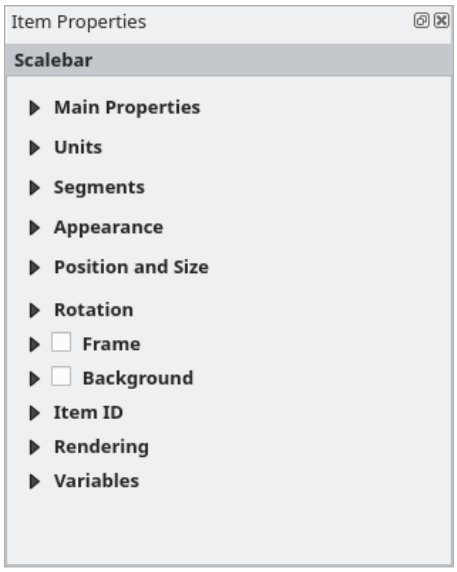



그림 18.35: 축척 막대 항목 속성 패널

### 주요 속성

축척 막대 *Item Properties* 패널의 *Main properties* 그룹은 다음 기능을 제공하고 있습니다. (그림 18.36 를 참조하세요.):




그림 18.36: 축척 막대 주요 속성 그룹

1. 먼저, 축척 막대를 첨부할 맵을 선택하십시오.
2. 그리고 축척 막대의 스타일을 선택하십시오. 다음 스타일들을 사용할 수 있습니다:
   - **Single box** 또는 **Double box** 스타일: 색상을 교차하는 상자들을 이은 한 줄 또는 두 줄 막대
   - **Middle**, **Up** 또는 **Down**: 축척 표시선이 축척 막대 가운데, 위쪽, 또는 아래쪽에 있음
   - **Stepped line**: 축척 막대를 표현하는 계단형 라인
   - **Hollow**: 선분마다 색상이 교체되며, 색상이 교체되는 선분들 가운데를 수평선이 지나는 단일 상자
   - **Numeric**: 축척 비율을 인쇄 (예: `1:50000`)
3. 속성을 적절하게 설정하십시오.

### 단위

축척 막대 *Item Properties* 패널의 *Units* 그룹은 표시될 단위와 텍스트 서식을 설정하기 위한 다음 기능을 제공하고 있습니다. (그림 18.37 를 참조하세요.):

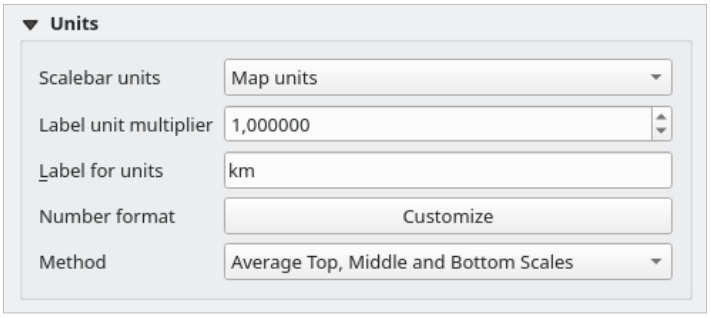



그림 18.37: 축척 막대 단위 그룹

- *Scalebar units*: 사용자가 원하는 단위를 선택하십시오. **Map Units** (기본값), **Meters**, **Feet**, **Miles** 또는 **Nautical Miles** 등등과 몇몇 파생형 가운데 하나를 선택할 수 있습니다. 단위 변환은 자동으로 이루어집니다.
- *Label unit multiplier*: 라벨 단위 당 축척 막대 단위가 얼마나 되는지 설정합니다. 예를 들어 축척 막대 단위를 "미터"로 설정한 경우, 곱수 (multiplier) 를 1,000 으로 설정하면 축척 막대 라벨이"킬로미터"가 될 것입니다.
- *Label for units*: 축척 막대 단위를 설명하는 데 쓰이는 텍스트를 — 예를 들어 `m` 또는 `km` — 정의합니다. 앞에서 말한 곱수를 반영하도록 맞춰야 합니다.
- *Customize*: *Number format* 옆에 있는 이 버튼을 누르면 축척 막대에서 천단위 구분자, 십진 자릿수, 과학적 기수법 등을 포함한 모든 숫자 서식 속성을 제어할 수 있습니다. (자세한 내용은숫자 서식 작업 을 참조하세요.) 현재 QGIS 로케일 밖에 있는 사용자를 위한 맵을 만들거나, 또는 로케일 기본값이 아닌 다른 스타일을 원하는 경우 (예: 로케일 기본값은 천단위 구분자를 감추는 것이지만 천단위 구분자를 사용하고 싶은 경우) 매우 유용합니다.
- Select the *Method* for scale bar calculation: Depending on the map CRS and extent, the distance reported in the scale bar may not be the same measured over the map. This option, which defaults to the *project's scale calculation method*, helps you specify the one you would like to use for this particular scale bar.

### 선분

축척 막대 *Item Properties* 패널의 *Segments* 그룹은 선분과 그 분할의 개수와 크기를 환경 설정하기 위한 다음 기능을 제공하고 있습니다. (그림 18.38 을 참조하세요.):

- *Segments*: 축척 막대의 `0` 의 좌측 및 우측에 얼마나 많은 분할을 그릴지 정의할 수 있습니다:
  - *Left*: 좌측에 있는 유일 선분의 분할 개수
  - *Right*: 우측에 있는 선분들의 분할 개수
- 분할 하나의 너비 또는 축척 막대 전체 길이에 대한 범위를 설정할 수 있습니다:
  - *Fixed width*: 분할 하나가 사용자의 축척 막대 단위로 어느 정도의 길이가 될지 설정하십시오.
  - *Fit segment width*: 또는 이 옵션에서 축척 막대 전체 크기를 `mm` 단위로 제한할 수도 있습니다. 이 경우, 맵 축척이 변경될 때마다 설정된 상한 및 하한 범위에 맞춰 축척 막대의 크기도 조정됩니다 (그리고 축척 라벨도 업데이트됩니다).
- *Height*: 축척 막대의 높이를 정의할 수 있습니다.

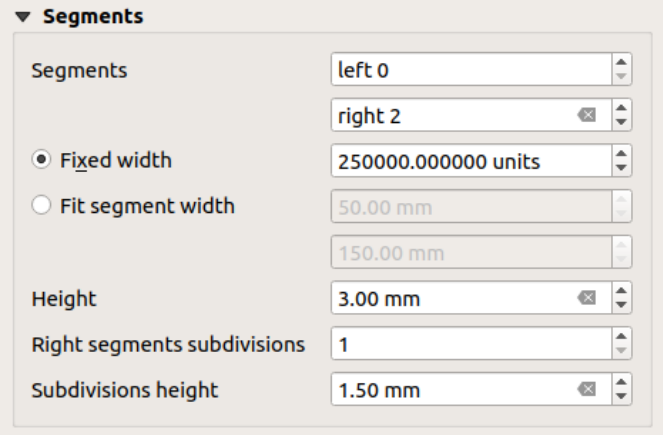



그림 18.38: 축척 막대 선분 그룹

- *Right segment subdivisions*: 축척 막대의 우측을 몇 개로 분할할지 정의할 수 있습니다. (*Line Ticks Down*, *Line Ticks Middle* 및 *Line Ticks Up* 축척 막대 스타일 전용)
- *Subdivision height*: 분할 선분의 높이를 정의할 수 있습니다.

### 모양

The *Appearance* group of the scale bar *Item Properties* panel provides the following functionalities:

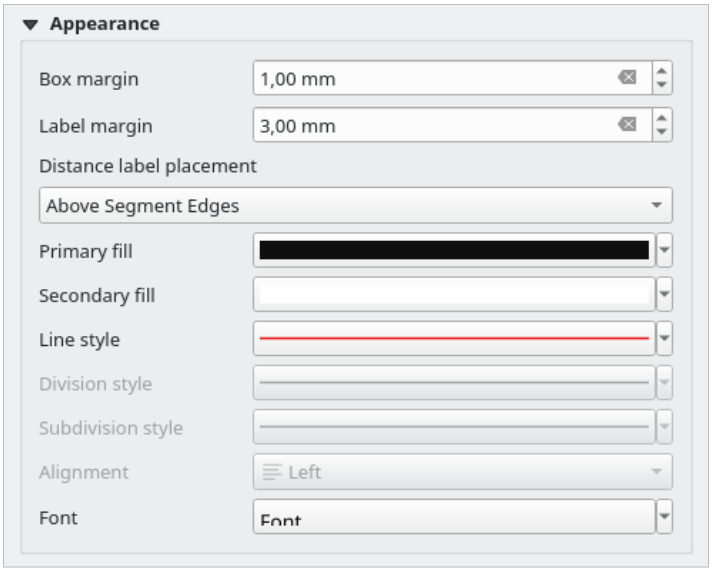



그림 18.39: Scale Bar Appearance group

축척 프레임 안에서 축척 막대를 어떻게 표시할지 정의할 수 있습니다.

- *Box margin*: 텍스트와 프레임 경계 사이의 여백을 설정합니다.
- *Label margin*: 텍스트와 축척 막대 렌더링 사이의 여백을 설정합니다.
- *Distance label placement* defines the position of the text relative to the scale bar segments: *Above Segment Edges*, *Above Segment Centers*, *Below Segment Edges* or *Below Segment Centers*

- *Primary fill* 및 *Secondary fill*: 축척 막대의 첫 번째 채우기와 두 번째 채우기를 (색상, 투명도, 패턴, 효과 등등) 채우기 심볼 속성 을 사용해서 그립니다. (*Single Box*, *Double Box* 및 *Hollow* 스타일 전용)
- *Line style*: 축척 막대의 라인을 (색상, 획, 결합 스타일, 선끝 스타일, 패턴, 효과 등등) 라인 심볼 속성 을 사용해서 그립니다. (*Numeric* 을 제외한 모든 스타일 전용)
- *Division style* 및 *Subdivision style*: *Line Ticks Up*, *Line Ticks Middle* 및 *Line Ticks Down* 축척 막대 스타일에서 각각 분할 및 하위분할 선분을 (색상, 획, 결합 스타일, 선끝 스타일, 패턴, 효과 등등) 라인 심볼 속성 을 사용해서 그립니다.
- *Alignment*: 텍스트를 축척 프레임의 좌측, 중앙, 또는 우측으로 정렬합니다. (**Numeric** 축척 막대 스타일 전용)
- *Font* 버튼을 클릭하면 축척 막대 라벨의 (크기, 글꼴, 색상, 자간, 음영, 배경 등등) 속성 을 설정할 수 있습니다.

축척 막대의 표시 속성 대부분이 데이터 정의된 속성을 가지고 있을 수도 있는 심볼에 의존하기 때문에, 데이터 정의된 축척 막대를 렌더링할 수도 있습니다.

**예시**: 축척 라벨의 볼드체에 적용된 다음 코드는 숫자가 500 의 배수인 경우 숫자를 볼드체로 표시할 것입니다:

```
-- returns True (or 1) if the value displayed on the bar
-- is a multiple of 500

@scale_value % 500 = 0
```

### 18.2.7 테이블 항목

사용자 맵을 장식하고 설명하는 데 테이블 항목을 사용할 수 있습니다:

- *Attribute table*: automatically extracts a subset of the attributes of a layer, based on predefined rules
- *Fixed table*: allows for creation of tables with contents manually entered (i.e. spreadsheet style), so that you can create completely custom tables.

#### 속성 테이블 항목

인쇄 조판기는 프로젝트의 모든 레이어의 속성을 표시할 수 있습니다. 항목 생성 지침 을 따라 *Add Attribute Table* 도구를 사용해서 새 속성 테이블 항목을 추가하십시오. 이후대화형 조판기 항목 작업 에서 설명한 내용과 동일한 방법으로 새 테이블 항목을 수정할 수 있습니다.

새 속성 테이블 항목은 기본적으로 (알파벳순으로 정렬된) 첫 번째 레이어의 첫 번째 행을 모든 필드와 함께 불러옵니다. 하지만 속성 테이블의 *Item Properties* 패널에서 테이블을 사용자 지정할 수 있습니다. 이 항목은공통 속성 이외에도 다음과 같은 기능을 가지고 있습니다 (그림 18.40 참조):

#### 주요 속성

속성 테이블의 *Main properties* 그룹은 다음 기능을 제공하고 있습니다. (그림 18.41 를 참조하세요.):

- *Source*: 기본적으로 프로젝트에 불러온 벡터 레이어들 가운데 *Layer* 를 선택할 수 있게 해주는 **Layer features** 만 선택할 수 있습니다.

  레이어 목록 가까이 있는 Data-defined override 버튼으로 테이블을 채우는 데 사용된 레이어를 동적으로 변경할 수 있습니다. 예를 들어 지도책 페이지마다 속성 테이블을 서로 다른 레이어 속성으로 채울 수 있습니다. 사용된 테이블 구조 (그림 18.44) 가 *Layer* 드롭다운 목록에 표시된 레이어의 테이블 구조이며 그대로 남아

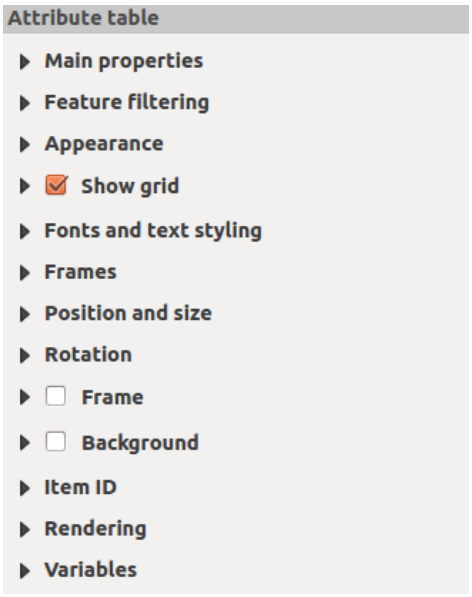



그림 18.40: 속성 테이블 항목 속성 패널

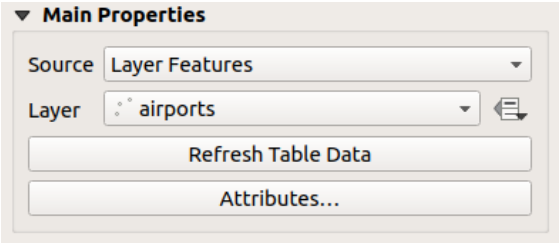



그림 18.41: 속성 테이블 주요 속성 그룹

있다는 사실을 기억하십시오. 즉 서로 다른 필드 (들) 을 가진 레이어에 데이터 정의 테이블을 설정하면 테이블에 비어 있는 열 (들) 이 생길 것입니다.

*Atlas* 패널에서 *Generate an atlas* 옵션을 활성화한 경우 (지도책 생성 참조) 다음과 같은 *Source* 2 개를 추가로 선택할 수 있습니다:

- **Current atlas feature** (그림 18.42 참조): 레이어를 선택할 수 있는 옵션은 표시되지 않으며, 테이블 항목은 지도책 커버리지 레이어의 현재 피처에서 나온 속성을 가진 행만 표시할 것입니다.
- **Relation children** (그림 18.43 참조): 관계명을 담은 옵션이 표시될 것입니다. 사용자가 지도책 커버리지 레이어를 부모 레이어로 사용해서관계 를 정의한 경우에만 이 옵션을 사용할 수 있으며, 테이블 항목은 지도책 커버리지 레이어의 현재 객체에서 나온 자식 행만 표시할 것입니다.

- *Refresh Table Data*: 속성 테이블의 실제 내용이 변경된 경우 테이블을 새로고침할 수 있습니다.

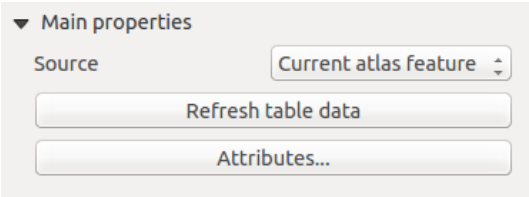



그림 18.42: 'Current atlas feature'에 대한 속성 테이블 주요 속성

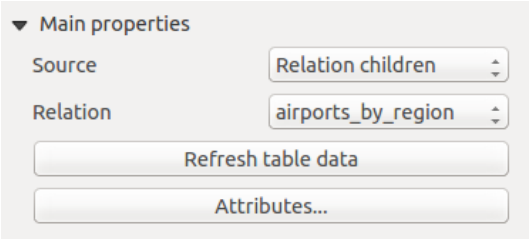



그림 18.43: 'Relation children'에 대한 속성 테이블 주요 속성

- *Attributes*⋯: *Select Attributes* 대화창을 엽니다. (그림 18.44 를 참조하세요.) 이 대화창에서 테이블에서 보이는 내용을 변경할 수 있습니다. 대화창 상단은 표시할 속성 목록을 담고 있으며, 하단에서 데이터가 정렬되는 방식을 설정할 수 있습니다.

  *Columns* 부분에서 다음 작업을 할 수 있습니다:

  - 행을 선택한 다음 및 버튼을 사용해서 이동시켜 목록에서 속성의 순서를 변경할 수 있습니다. 여러 행을 한 번에 선택해서 이동시킬 수 있습니다.
  - 버튼으로 새 속성을 추가할 수 있습니다. 테이블 하단에 비어 있는 새 행을 추가하는데, 속성 값이 될 필드를 선택하거나 또는 정규 표현식을 통해 새 속성을 생성할 수 있습니다.
  - 버튼으로 속성을 제거할 수 있습니다. 여러 행을 한 번에 선택해서 이동시킬 수 있습니다.
  - *Reset* 버튼을 눌러 속성 테이블을 기본 상태로 복구시킬 수 있습니다.
  - *Clear* 버튼을 사용하면 테이블을 비울 수 있습니다. 대용량 테이블에서 몇 개 안 되는 속성만 표시하고자 할 때 유용합니다. 각 행을 하나씩 제거하는 대신, 전체 테이블을 비운 다음 필요한 행들을 추가하는 것이 빠를 수도 있으니까요.
  - *Heading* 열에 사용자 지정 텍스트를 추가해서 셀 제목 (cell heading) 을 변경할 수 있습니다.
  - 테이블 셀 내부의 텍스트 위치를 지시하는 *Alignment* 열을 통해 셀 정렬을 관리할 수 있습니다.

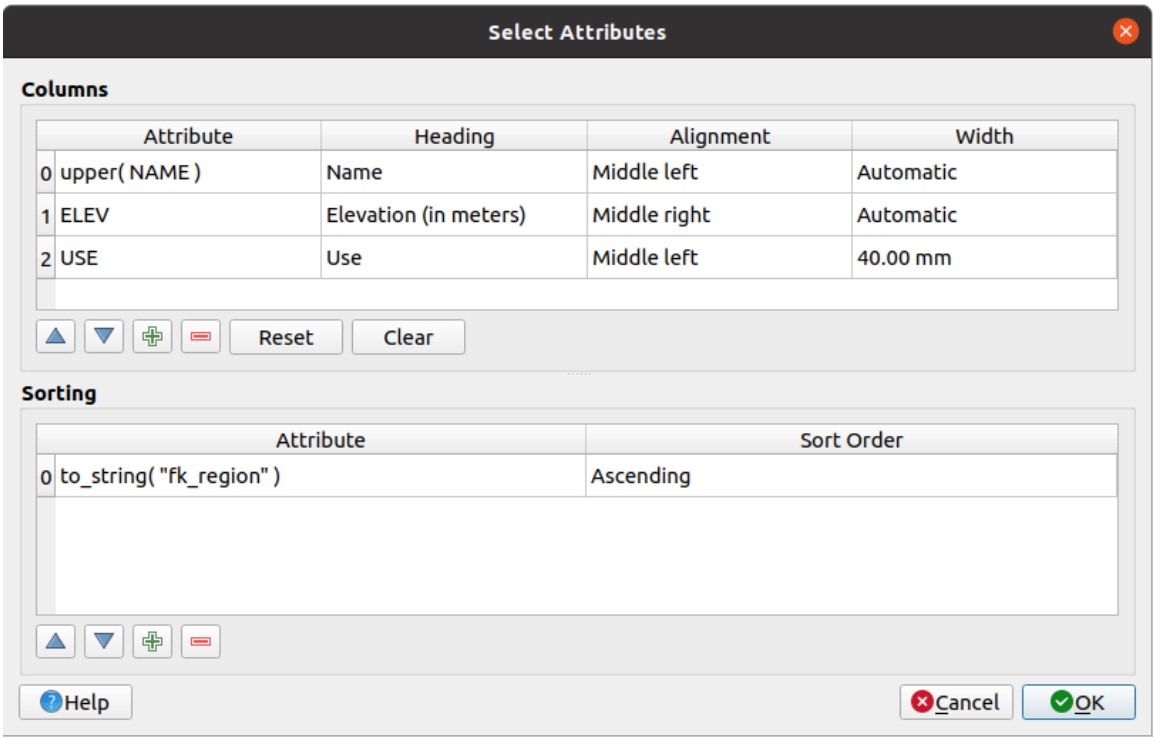



그림 18.44: 속성 테이블 속성 선택 대화창

- *width* 열에 사용자 지정 값을 추가하면 셀 너비를 직접 관리할 수 있습니다.

*Sorting* 부분에서 다음 작업을 할 수 있습니다:

- 테이블을 정렬시킬 속성을 추가할 수 있습니다: 버튼을 누르면 새로운 빈 행이 추가됩니다. *Attribute* 열에 필드 또는 표현식을 삽입한 다음 *Sort order* 를 **오름차순 (Ascending)** 또는 **내림차순 (Descending)** 으로 설정하십시오.
- 이 목록에서 행을 선택한 다음 및 버튼을 사용하면 속성 수준에서의 정렬 우선 순위를 변경할 수 있습니다. *Sort Order* 열에서 셀을 선택하면 속성 필드의 정렬 순서를 변경할 수 있습니다.
- 버튼을 사용하면 정렬한 목록에서 속성을 제거할 수 있습니다.

### 피처 필터링

속성 테이블의 *Feature filtering* 그룹은 다음 기능을 제공하고 있습니다. (그림 18.45 을 참조하세요.):

다음 작업을 할 수 있습니다:

- *Maximum rows* 옵션에서 표시할 최대 행 개수를 정의할 수 있습니다.
- *Remove duplicate rows from table* 옵션을 활성화하면 유일한 레코드만 표시할 수 있습니다.
- *Show only visible features within a map* 옵션을 활성화하고 대응하는 *Linked map* 을 선택하십시오. 링크된 맵의 가시화된 피처의 속성만 표시할 것입니다.
- *Generate an atlas* 옵션을 활성화한 경우에만 *Show only features intersecting Atlas feature* 를 활성화할 수 있습니다. 이 옵션을 활성화하면, 현재 지도책 피처와 교파하는 피처만 가진 테이블을 표시할 것입니다.

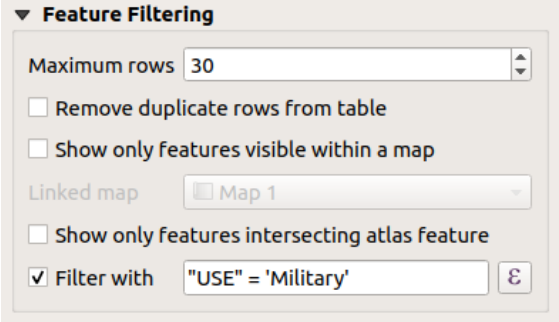



그림 18.45: 속성 테이블 피처 필터링 그룹

- *Filter with* 옵션을 활성화하면 입력 란에 필터를 직접 입력하거나 또는 옆에 있는 버튼을 눌러 정규 표현식을 삽입할 수 있습니다. 다음은 QGIS 예시 데이터셋에서 `airports` 레이어를 불러온 경우 사용할 수 있는 필터링 선언문의 몇몇 예시입니다:
  - `ELEV > 500`
  - `NAME = 'ANIAK'`
  - `NAME NOT LIKE 'AN%'`
  - `regexp_match( attribute( $currentfeature, 'USE' ) , '[i]')`

  마지막 정규 표현식은'USE'속성 필드에 `i` 문자가 있는 공항만 포함시킬 것입니다.

### 모양

속성 테이블의 *Appearance* 그룹은 다음 기능을 제공하고 있습니다. (그림 18.46 를 참조하세요.):

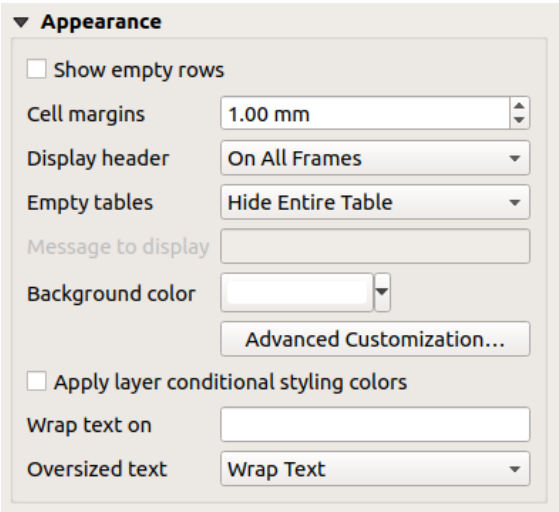



그림 18.46: 속성 테이블 모양 그룹

- *Show empty rows* 를 체크하면 속성 테이블에 빈 셀들이 표시됩니다. 사용자가 표시해야 할 산출물이 있는 경우 빈 셀들을 추가하는 데 이 옵션을 쓸 수도 있습니다!
- *Cell margins* 옵션에서 테이블의 각 셀에 있는 텍스트 주위의 여백을 정의할 수 있습니다.

- *Display header* 콤보박스 목록에서 헤더 표시를'On first frame', 기본 값인'On all frames', 또는'No header'가운데 하나로 설정할 수 있습니다.
- *Empty tables* 옵션은 산출물 집합이 비어 있는 경우 무엇을 표시할 것인지 제어합니다.
  - **Draw headers only**: *Display header* 옵션에서'No header'를 선택하지 않았다면, 헤더만 렌더링합니다.
  - **Hide entire table**: 테이블의 배경만 렌더링합니다. 테이블을 완전히 숨기려면 *Frames* 대화창에 있는 *Don't draw background if frame is empty* 옵션을 활성화해야 합니다.
  - **Show set message**: 헤더를 렌더링하고 모든 열들을 포괄하는 셀을 추가한 다음'No result'같은 메시지를 표시합니다. *Message to display* 옵션에서 이 메시지를 지정할 수 있습니다.
- *Message to display* 옵션은 *Empty tables* 에서 **Show set message** 를 선택한 경우에만 활성화됩니다. 여기에 입력한 메시지는 산출물이 빈 테이블인 경우 테이블의 첫 행에 표시될 것입니다.
- *Background color* 옵션에서색상 선택기 위젯을 사용해서 테이블의 배경색을 설정할 수 있습니다. *Advanced customization* 옵션으로 누르면 각 셀 별로 서로 다른 배경색을 정의할 수 있습니다. (그림 18.47 를 참조하세요.)

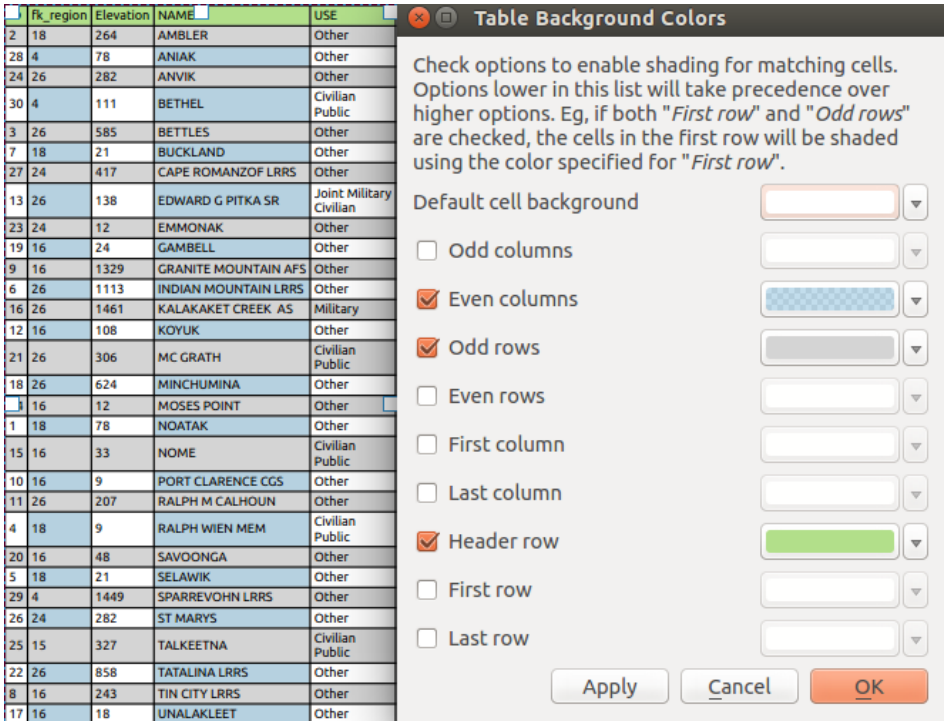



그림 18.47: 속성 테이블 고급 배경 대화창

- *Apply layer conditional styling colors*: 조판 속성 테이블 내부에 레이어가 가진 (배경색, 글꼴 계열 및 진한 글꼴, 기울인 글꼴, 취소선, 밑줄, 색상 등등의) 조건부 테이블 서식 을 적용합니다. 조건부 서식 규칙이 다른 조판 테이블 서식 설정을 우선합니다. 예를 들면 조건부 서식 규칙은 번갈아 바뀌는 행 색상 같은 기타 셀 배경색 설정을 무시할 것입니다.
- *Wrap text on* 옵션에서 셀 내부의 행갈이 문자를 정의할 수 있습니다.
- *Oversized text* 옵션에서 열에 대해 설정한 너비가 그 내용의 길이보다 짧을 경우의 습성을 정의할 수 있습니다. **Wrap text** 는 내용을 행갈이하고, **Truncate text** 는 열의 너비보다 긴 부분을 잘라버립니다.

> **참고**
>
> 테이블 공통 기능 에서 속성 테이블 항목의 속성을 자세히 설명하고 있습니다.

### 고정 테이블 항목

항목 생성 지침 을 따라 *Add Fixed Table* 도구를 사용해서 새 테이블 항목을 추가하면 테이블에 추가적인 맵 관련 정보를 직접 삽입할 수 있습니다. 이후대화형 조판기 항목 작업 에서 설명한 내용과 동일한 방법으로 새 테이블 항목을 수정할 수 있습니다.

기본적으로, 맵 조판기에 최소화된 열과 행 2 개가 나타납니다. *Item Properties* 패널에서 테이블을 사용자 지정해야 합니다. 공통 항목 속성 이외에도, 다음과 같은 기능을 가지고 있습니다:

### 주요 속성

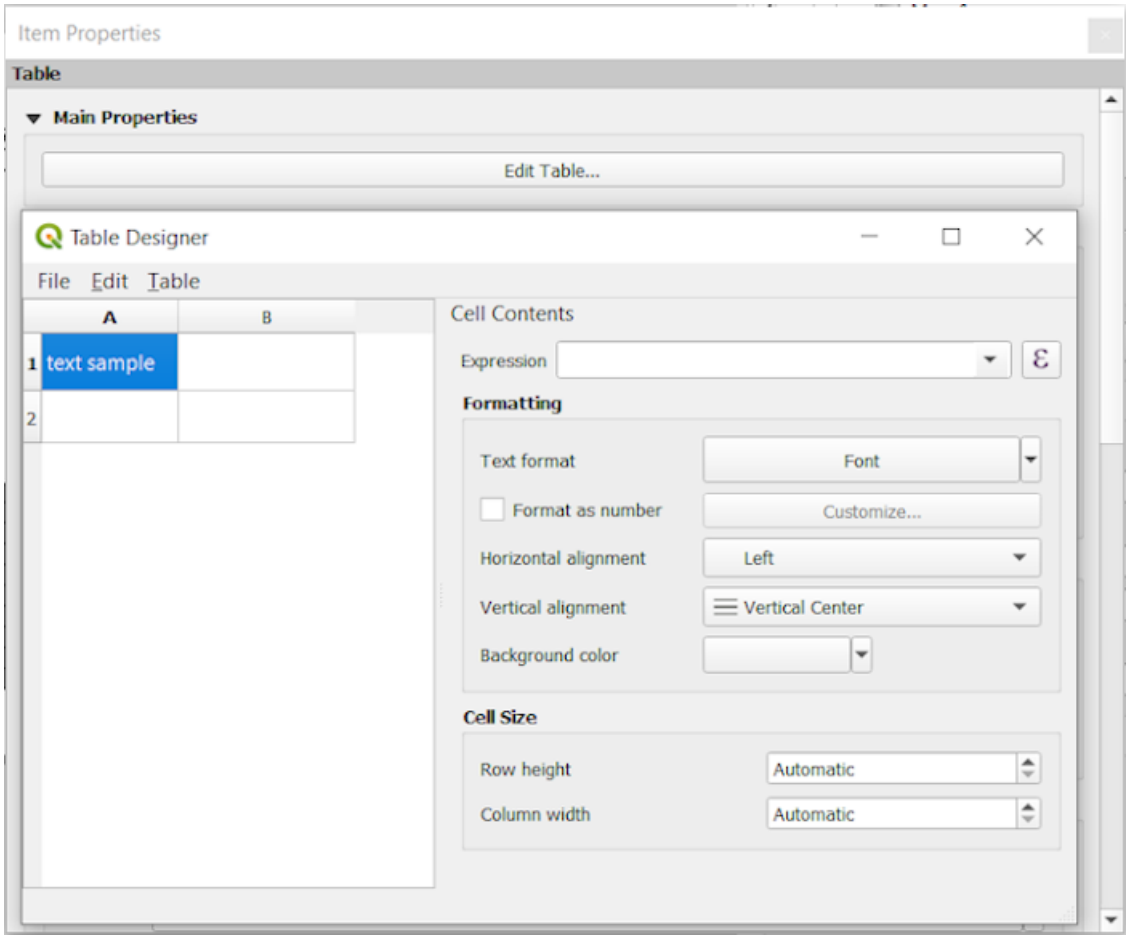



그림 18.48: 테이블 설계자를 띄운 고정 테이블 항목 속성 패널

In *Main properties*, press *Edit table*⋯ to open the *Table designer* dialog where you can build your table (double-clicking the table item will also open the editor dialog):

- On the left hand of the dialog, a table frame displays the active table item with its content and settings. Click or double-click on a cell and write a text to modify its contents. Press `Ctrl+Enter` to add linebreak (for a multiline text). The logic for selecting multiple cells in the table is the same as in the vector layer *attribute table*. Grab and move a row or column separator to resize the row above or the column before. A double-click on the separator will resize the row above or the column before to fit their contents. A right-click over column or row headers shows a contextual menu to insert columns or rows around the selection or to delete them.
- Through the menus on top of the dialog, it is possible to:
  - *File → Import Content From Clipboard*: it overrides contents of the table.
  - *File → Close editor*: it closes the *Table Designer* dialog.
  - From the *Edit* menu, you can work with selection functionalities for rows and columns:
    - *Select all* cells in the table
    - *Select columns* or *Select Rows*: when cells are selected in the table, you can extend the selection respectively to their column(s) or row(s)
    - *Clear Cells*: deletes the content of the selected cells
  - The *Table* menu is the place where you design the structure of the table. You can:
    - *Insert rows →* above or below the selection
    - *Insert columns →* before or after the selection
    - *Delete Rows* or *Delete Columns* of selected cells
    - *Merge selected cells*: select multiple cells in a rectangular shape and you can merge them and concatenate their contents into the top left cell (whose styling is also applied to the merged cell, unless there is a last row/column cell in the selection).
    - *Split selected cells*: select a merged cell and you can split it back to its individual cells. The current text is kept in the top left cell, and the other cells are filled with their contents before they get merged. They are also reapplied their original styling.
    - *Include Header Row* whose styling can only be controlled from the 글꼴 및 텍스트 스타일 widget
- Select the cells and, on the right of the table frame, you can provide the *Cell Contents*:
  - Either manually enter the contents of each cell, or use an *Expression* to automatically populate it.
  - Under the *Formatting* group, set specific options that apply to the selection and take precedence over the *global table content styling*:
    - *Text format*: 텍스트 서식을 선택할 수 있습니다.
    - *Format as number* 옵션을 활성화하면 숫자 서식을 선택할 수 있습니다 (몇몇 서식을 사용할 수 있습니다).
    - *Horizontal alignment* 및 *Vertical alignment*: 수평 및 수직 정렬 방식을 선택할 수 있습니다.
    - *Background color*: 셀의 배경색을 선택할 수 있습니다.
  - *Cell Size* 부분의 *Row height* 및 *Column width*: 셀의 행 높이와 열 길이를 정의할 수 있습니다.

### 모양

고정 테이블의 *Appearance* 그룹은 다음 기능을 제공하고 있습니다:

- *Show empty rows* 를 체크하면 속성 테이블을 빈 셀들로 채웁니다.
- *Cell margins* 옵션에서 테이블의 각 셀에 있는 텍스트 주위의 여백을 정의할 수 있습니다.
- *Display header* 콤보박스 목록에서 헤더 표시를'On first frame', 기본 값인'On all frames', 또는'No header'가운데 하나로 설정할 수 있습니다.
- *Background color* 옵션에서색상 선택기 위젯을 사용해서 테이블의 배경색을 설정할 수 있습니다. *Advanced customization* 옵션으로 누르면 각 셀 별로 서로 다른 배경색을 정의할 수 있습니다.
- *Oversized text* 옵션에서 열에 대해 설정한 너비가 그 내용의 길이보다 짧을 경우의 습성을 정의할 수 있습니다. **Wrap text** 는 내용을 행갈이하고, **Truncate text** 는 열의 너비보다 긴 부분을 잘라버립니다.

> **참고**
>
> 테이블 공통 기능 에서 고정 테이블 항목의 속성을 자세히 설명하고 있습니다.

## 테이블 공통 기능

### 그리드 표시

테이블 항목의 *Show grid* 그룹은 다음 기능을 제공하고 있습니다. (그림 18.49 을 참조하세요.):

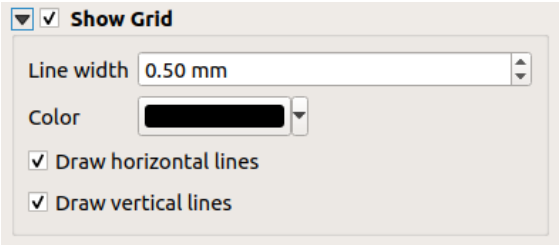



그림 18.49: 속성 테이블 그리드 표시 그룹

- *Show grid*: 테이블 셀의 윤곽선인 그리드를 표시하고 싶은 경우 이 옵션을 활성화하십시오. *Draw horizontal lines* 또는 *Draw vertical lines* 가운데 하나를 선택하거나, 둘 다 선택할 수도 있습니다.
- *Line width*: 그리드에 사용되는 라인의 두께를 설정합니다.
- *Color*: 색상 선택기 위젯을 사용해서 그리드의 색상을 설정합니다.

### 글꼴 및 텍스트 스타일

테이블 항목의 *Fonts and text styling* 그룹은 다음 기능을 제공하고 있습니다. (그림 18.50 을 참조하세요.):




그림 18.50: 속성 테이블 글꼴 및 텍스트 스타일 그룹

- 고급 (버퍼, 음영, 그리기 효과, 투명도, 배경, 색상 등등) 텍스트 설정 위젯을 사용해서 *Table heading* 및 *Table contents* 의 *Font* 속성을 정의할 수 있습니다. 텍스트 설정을 변경해도 *Appearance* 부분 또는 *Table Designer* 대화창에서 셀에 할당한 사용자 정의 폰트는 영향을 받지 않는다는 점을 기억하십시오. 기본 렌더링된 셀만 덮어씁니다.
- For *Table heading* you can additionally set the *Alignment* to `Follow column alignment` or override this setting by choosing `Left`, `Center` or `Right`. The column alignment is set using the *Select Attributes* dialog (see 그림 18.44).

### 프레임

테이블 항목 속성의 *Frames* 그룹은 다음 기능을 제공하고 있습니다. (그림 18.51 를 참조하세요.):

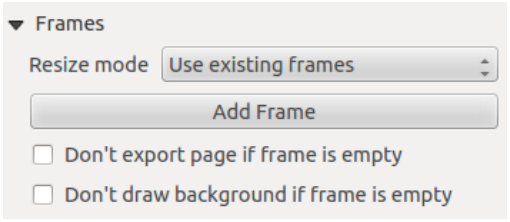



그림 18.51: 속성 테이블 프레임 그룹

- *Resize mode* 콤보박스 목록에서 속성 테이블의 내용을 어떻게 렌더링할지 선택할 수 있습니다:
  - `Use existing frames`: 첫 번째 프레임과 추가된 프레임에만 산출물을 표시합니다.
  - `Extend to next page`: 속성 테이블의 전체 집합을 표시하는 데 필요한 만큼 많은 프레임 (과 대응하는 페이지) 을 생성합니다. 조판 상에서 각 프레임을 이리저리 옮길 수 있습니다. 프레임의 크기를 조정하는 경우, 산출 테이블이 다른 프레임들 사이로 나뉠 것입니다. 마지막 프레임은 테이블에 맞춰 빈 부분을 잘라냅니다.
  - `Repeat until finished`: 'Extend to next page'옵션과 마찬가지로 필요한 만큼 프레임을 생성합니다. 다만 모든 프레임의 크기가 동일할 것입니다.
- *Add Frame* 버튼을 클릭하면 선택한 프레임과 동일한 크기의 새 프레임을 추가합니다. *Resize mode* 또는 `Use existing frames` 를 선택한 경우, 첫 번째 프레임에 다 들어가지 않은 산출 테이블을 다음 프레임으로 넘길 것입니다.

- *Don't export page if frame is empty* 옵션을 활성화하면, 테이블 프레임에 아무 내용도 없을 때 페이지를 내보낼 수 없게 됩니다. 즉 산출물에 다른 모든 조판기 항목들 — 맵, 축척 막대, 범례 등등 — 이 보이지 않을 것이라는 뜻입니다.
- *Don't draw background if frame is empty* 옵션을 활성화하면, 테이블 프레임에 아무 내용도 없을 때 배경을 렌더링할 수 없게 됩니다.

### 18.2.8 마커, 그림 및 방위표 항목


- 그림 항목
  - 주요 속성
  - 크기 및 배치
  - 이미지 기울이기
- 방위표 항목
- 마커 항목


인쇄 조판에서 맵 또는 범례 항목과 함께 사용자가 구현한 내용을 이미지 또는 주석으로 장식하고 싶을 수도 있습니다. QGIS 는 이를 달성하기 위한 다양한 도구를 제공합니다:

- 그림 항목: 조판을 이미지 래스터 또는 SVG 파일로 장식합니다. (예: 로고, 사진, 방위표, …)
- 방위표 *(north arrow)* 항목: 방위표 이미지로 사전 정의된 그림 항목입니다.
- 마커 항목: 조판을 QGIS 벡터심볼 로 장식합니다. 맵 항목 위에 마커를 배치하거나 고급 사용자 지정 범례를 생성하는 데 쓰일 수 있습니다.

#### 그림 항목

그림을 파일 관리자에서 맵 캔버스로 드래그하거나, `Ctrl+V` 조합키 또는 *Edit* ▷ *Paste* 메뉴를 이용해서 조판에 직접 붙여넣거나, 또는항목 생성 지침 을 따라 Add Picture 아이콘을 클릭해서 추가할 수 있습니다. 그 다음대화형 조판기 항목 작업 에서 설명한 대로 그림을 조작할 수 있습니다.

Add Picture 아이콘을 사용하는 경우, 그림 항목은 그림의 *Item Properties* 패널에서 사용자 지정할 수 있는 비어 있는 프레임이 될 것입니다. 공통 항목 속성 이외에도, 다음과 같은 기능을 가지고 있습니다:

#### 주요 속성

그림 항목은 두 가지 유형의 이미지를 지원합니다:

- *Raster Image*: 파일 선택기를 이용해서 데이터를 가져올 수 있습니다. … Browse 버튼을 이용해서, 또는 또는 텍스트란에 직접 경로를 입력해서 사용자 컴퓨터에 있는 파일을 선택하십시오. 그림을 가리키는 원격 URL 을 지정할 수도 있습니다. 조판에 관련 이미지를내장 시킬 수도 있습니다.

  data defined override 버튼을 클릭하면 피처 속성으로부터, 또는 정규 표현식을 이용해서 이미지 소스를 설정할 수 있습니다.

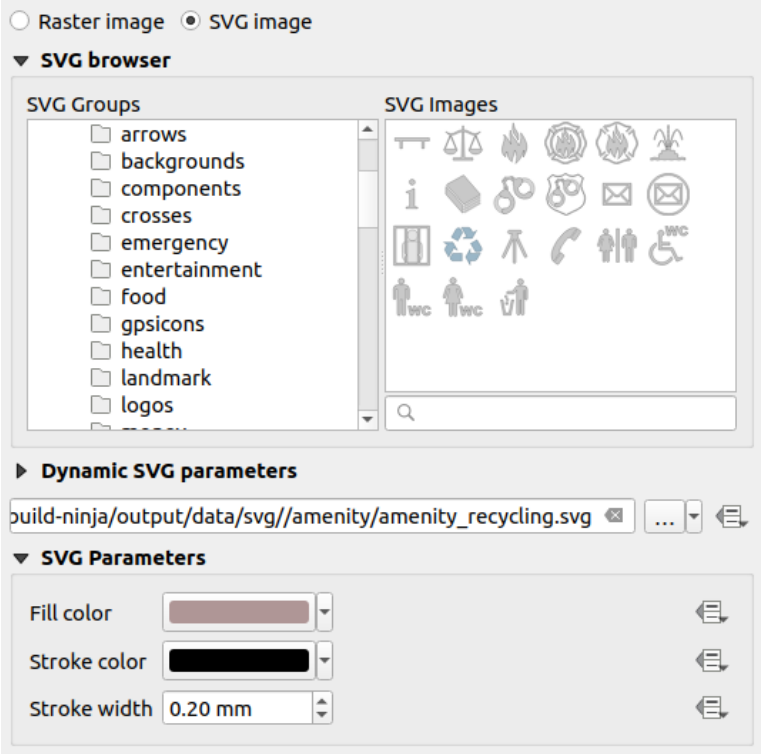



그림 18.52: 그림 항목 속성 패널

- *SVG Image*: 기본적으로 *Settings* → *Options* → *System* → *SVG Paths* 메뉴에서 제공하는 SVG 라이브러리를 이용합니다. 하지만 다른 파일을 이용할 수 있으며, 래스터 이미지와 동일한 규칙을 따라 파일을 선택할 수 있습니다. 물론 SVG 파라미터도 동적으로 설정할 수 있습니다.

  QGIS 가 기본으로 지원하는 `.SVG` 파일은 사용자 지정 가능합니다. 즉 원본과는 다른 *Fill color*, *Stroke color* (투명도 포함) 및 *Stroke width* 를 *SVG Parameters* 그룹에 있는 대응 기능을 사용해서 쉽게 적용할 수 있다는 뜻입니다. 물론 이 속성들을데이터 정의 할 수도 있습니다.

  이 속성들이 활성화되지 않은 `.SVG` 파일을 추가한 경우, 예를 들어 투명도를 지원하려면 파일에 다음 태그들을 추가해야 합니다:

  - `fill-opacity="param(fill-opacity)"`
  - `stroke-opacity="param(outline-opacity)"`

  더 자세한 내용은파라미터화할 수 있는 *SVG* 를 참조하세요.

> **참고**
>
> 조판 페이지에 (래스터 또는 SVG) 이미지 파일을 드래그 & 드롭하면 대응하는 설정을 가진 조판 그림 항목을 생성할 것입니다.

### 크기 및 배치




그림 18.53: 조판 그림 크기 및 배치 속성

*Resize mode* 옵션을 통해 프레임 크기가 변경될 경우 이미지를 어떻게 표시할지 설정할 수 있습니다:

- `Zoom`: 이미지의 종횡비를 유지한 채 이미지를 프레임에 맞춰 확대/축소합니다.
- `Stretch`: 프레임 내부에 맞춰 이미지를 늘립니다.
- `Clip`: 이 모드는 래스터 이미지 전용입니다. 이 모드는 크기 조정 없이 원본 이미지 크기를 유지한 채, 프레임으로 이미지를 잘라냅니다. 즉 프레임 내부에 들어오는 이미지의 일부만 보일 것이라는 뜻입니다.
- `Zoom and resize frame`: 이미지를 프레임에 맞춰 확대한 다음, 프레임 크기를 산출 이미지 크기에 맞춰 조정합니다.
- `Resize frame to image size`: 프레임 크기를 원본 이미지 크기에 맞춰 조정합니다. (이미지 크기를 조정하지 않습니다.)

선택한 *Resize mode* 에 따라 *Placement* 및 *Image rotation* 옵션이 비활성화될 수도 있습니다. *Placement* 옵션은 이미지 프레임 안에서의 이미지 위치를 (위/가운데/아래 및 왼쪽/가운데/오른쪽 정렬을) 선택할 수 있게 해줍니다.

### 이미지 기울이기

*Image rotation* 란을 이용하면 이미지를 기울일 수 있습니다. *Sync with map* 옵션을 활성화하면 이미지의 기울기와 선택한 맵 항목에 적용된 기울기를 동기화합니다. 이 옵션은 모든 그림을 방위표처럼 동작하게 만드는 편리한 기능입니다. *North alignment* 를 다음 가운데 하나로 선택할 수 있습니다:

- **도북 (□□, Grid north)**: 국가/지역 (national/local) 수준 그리드 상의 중앙 경선 (□□□□, central meridian) 에 평행하는 그리드 선의 방향
- **진북 (True north)**: 경도 자오선 (meridian of longitude) 의 방향

그림 기울기에 편각 (declination) *Offset* 도 적용할 수 있습니다.

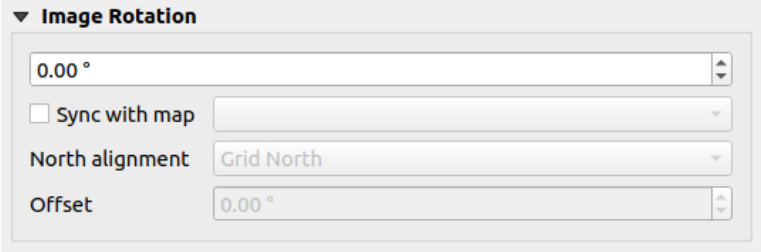



그림 18.54: 조판 그림 이미지 기울기 속성

### 방위표 항목

항목 생성 지침 을 따라 *Add North Arrow* 버튼으로 방위표 항목을 추가하십시오. 이후대화형 조판기 항목 작업 에서 설명한 내용과 동일한 방법으로 새 방위표 항목을 수정할 수 있습니다.

방위표는 이미지이기 때문에, *North Arrow* 항목은그림 항목 과 동일한 속성을 가지고 있습니다. 주요 차이점은 다음과 같습니다:

- 새 항목 추가시 빈 프레임 대신 기본 방위표를 사용합니다.
- 방위표 항목은 기본적으로 맵 항목과 동기화됩니다: *Sync with map* 속성이 방위표가 렌더링되는 맵입니다. 기본 선택된 맵이 없다면, 참조 맵 으로 돌아가게 됩니다.

**참고**

방위표 대부분은 진북 방향에 `N` 표시가 없습니다. `N` 을 북쪽이라는 의미로 사용하지 않는 언어들이 많기 때문입니다.

### 마커 항목

마커 항목을 추가하려면, Add Marker 아이콘을 클릭하고 페이지를 클릭하십시오. 기본 포인트 마커 심볼을 추가합니다. 그 다음대화형 조판기 항목 작업 에서 설명한 대로 마커를 조작할 수 있습니다. 그러나 다른 대부분의 항목들과는 달리, 내장 심볼 속성이 마커의 크기를 제어하기 때문에 마커 항목의 크기를 조정할 수 있다는 사실을 기억하십시오.

마커의 *Item Properties* 패널에서 마커 항목을 사용자 지정할 수 있습니다. 공통 항목 속성 이외에도, 다음과 같은 속성을 설정할 수 있습니다:

- 모든 심볼위젯 케이퍼빌리티 덕분에 *Symbol* 을 수정할 수 있습니다.
- 마커 항목의 기울기를 맵의 기울기와 동기화시켜 (이미지 기울이기 참조) 방위표처럼 동작하게 할 수 있습니다. 맵 기울기는 모든 기존 마커 심볼 수준의 기울기에 추가됩니다. (따라서 예를 들어 삼각형 마커를 90° 로 기울여 똑바로 위를 향하게 해야 하는 경우에도 방위표 모드에서 여전히 잘 작동할 것입니다!)

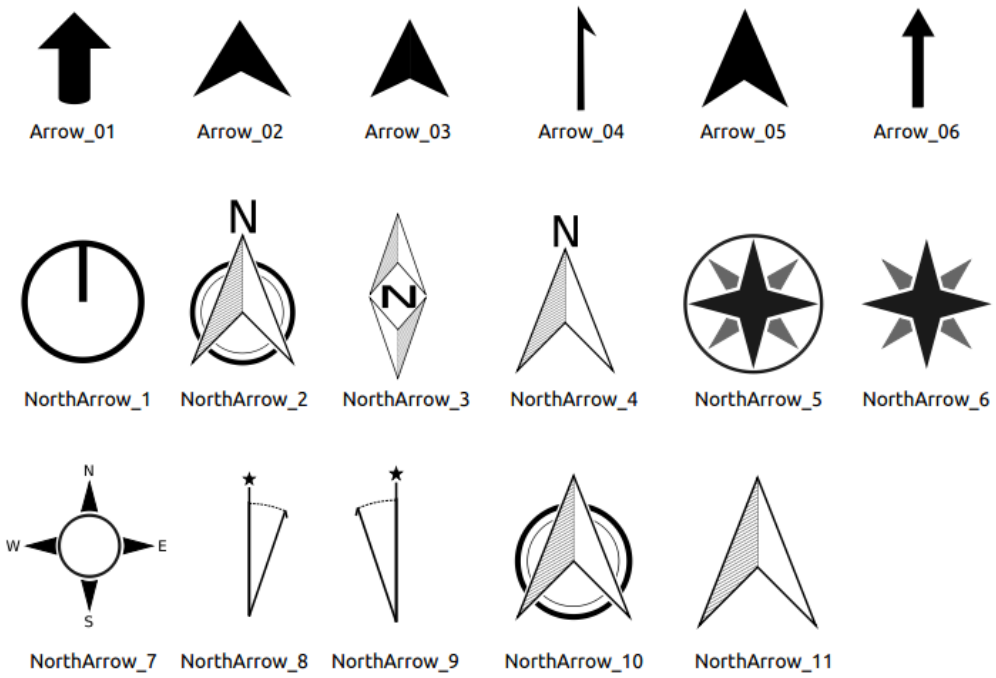



그림 18.55: SVG 라이브러리에서 선택하고 사용할 수 있는 방위표들

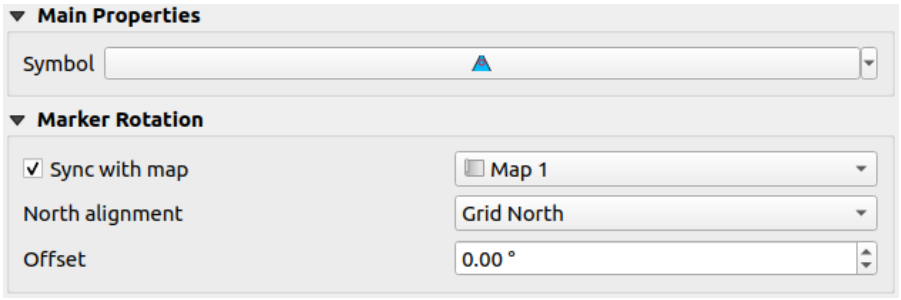



그림 18.56: 마커 항목 사용자 지정 속성

### 18.2.9 표고 단면 항목

표고 단면 항목은 조판에표고 단면 뷰 를 표시하는 데 쓰입니다.  *Add Elevation Profile* 버튼을 클릭하고항목 생성 지침 을 따라 새 표고 단면 항목을 추가하십시오. 표고 단면 항목을 추가한 후대화형 조판기 항목 작업 에서 설명한 내용과 동일한 방법으로 수정할 수 있습니다.

새 표고 단면 항목에는 해당 항목을 비어 있는 도표로 만들어주는 기본 설정이 있습니다. *Item Properties* 패널에서 그 속성을 사용자 지정할 수 있습니다. 공통 속성 이외에도, 다음과 같은 기능을 가지고 있습니다:

표고 단면의 *Item Properties* 패널은 상단에 다음 기능을 가진 툴바를 내장하고 있습니다:

-  Update elevation profile: 표고 단면 항목을 다시 렌더링합니다.
-  Copy from elevation profile: 표고 단면 뷰를 선택할 수 있는 드롭다운 메뉴입니다. 조판 표고 단면 항목에 뷰 설정을 적용한 다음 수정할 수 있습니다.

#### 레이어

*Layers* 그룹 아래에 있는 트리 뷰에서 단면 항목에 렌더링하고자 하는 레이어를 체크하십시오. 선택한 레이어의 *Elevation* 속성을 적절하게 환경설정하는 것을 잊지 마십시오.

#### 단면 곡선

-  *Controlled by atlas*: 현재 지도책 피처로부터단면 곡선 을 가져와서 사용자가 지도책 피처를 작업하는 동안 표고 단면 뷰를 업데이트할 것입니다. 현재 이 옵션은 활성 조판 지도책 또는 보고서에 대해 라인 도형 유형 커버리지 레이어를 사용해서 지원하고 있습니다.
- *Tolerance*: 데이터 정의할 수 있는 이 허용오차 거리는 가시 레이어의 피처가 조판 표고 뷰에 보이려면 단면 곡선으로부터 어느 정도의 거리 안에 들어와야 하는지를 제어합니다. 지금은 포인트 피처만 반환합니다.

#### Show Subsections Indicator

It is possible to highlight the position of the vertices of the curve on the Elevation Profile. Select the  *Show Subsections Indicator* to display vertical lines at the distances of the vertices. Use the *Lines* drop-down list to set the symbol, color, opacity and width of the Subsections Indicator lines.

#### 도표 범위

조판 표고 단면 항목이 반드시 그 바탕이 되는 표고 단면 뷰의 전체 범위를 표시해야 할 필요는 없습니다. 다음을 지정하면 렌더링할 영역을 제한할 수 있습니다:

- X 축 상에서 단면 곡선 시작 포인트로부터의 *Minimum distance* 최단 거리 및 *Maximum distance* 최장 거리
- Y 축 상에서 *Minimum elevation* 최저 표고 및 *Maximum elevation* 최고 표고

#### 거리 및 표고 축

*Distance axis* 및 *Elevation axis* 그룹에는 각각 X 및 Y 축 상에서 표고 단면 항목 위의 그리드를 정밀하게 조정할 수 있는 옵션이 있습니다:

- *Unit*: 이 거리 표시 단위를 설정하면 맵 캔버스 단위를 무시할 수 있습니다.
- *Major interval* 및 *Minor interval*: 축의 눈금을'주 간격'및'부 간격'으로 설정할 수 있습니다.
- *Major grid lines* 및 *Minor grid lines*: '주 그리드 선'및'부 그리드 선'에 각각 대응하는 라인 심볼을 적용할 수 있습니다.
- *Label interval*: 눈금 항목에 얼마나 규칙적으로 라벨을 배치할지 설정할 수 있습니다. 또한 *Label format* 라벨 서식과 *Label font* 라벨 글꼴도 설정할 수 있습니다.
- *Distance labels*: *Every value* 모든 값, *First value* 첫 번째 값, *Last value* 마지막 값, *First and last values* 첫 번째와 마지막 값 옆에 거리 단위 심볼을 배치해야 할지, 아니면 *None* 건너뛰어야 할지를 환경설정합니다.

#### 도표 영역

*Chart area* 아래에서 표고 단면 도면을 실제로 표시할 영역의 렌더링을 환경설정할 수 있습니다:

- *Background*: 배경 채우기 심볼
- *Border*: 경계 라인 심볼
- 표고 단면 항목 경계로부터의 여백

### 18.2.10 HTML 프레임 항목

웹사이트의 내용을 표시하는 프레임을 추가하거나, 또는 사용자 고유의 HTML 페이지를 생성하고 스타일을 적용할 수 있습니다. 항목 생성 지침 을 따라 *Add HTML* 도구로 HTML 프레임을 추가하고대화형 조판기 항목 작업 에서 설명한 내용과 동일한 방법으로 수정할 수 있습니다. HTML 프레임 생성 당시의 조판 내보내기 해상도가 HTML 축척을 제어한다는 점을 기억하십시오.

HTML *Item Properties* 패널에서 HTML 항목을 사용자 지정할 수 있습니다. 이 항목은공통 속성 이외에도 다음과 같은 기능을 가지고 있습니다 (그림 18.57 참조):

#### HTML 소스

HTML 프레임 *Item Properties* 패널의 *HTML Source* 그룹은 다음 기능을 제공하고 있습니다. (그림 18.58 을 참조하세요.):

- *URL* 옵션에서 사용자의 인터넷 브라우저에서 복사한 웹페이지의 URL 을 입력하거나, 또는 … Browse 버튼을 눌러 HTML 파일을 선택할 수 있습니다. 아니면 Data-defined override 버튼을 클릭해서 테이블에 있는 속성 필드의 내용에서 또는 정규 표현식을 사용해서 URL 을 지정할 수도 있습니다.
- *Source* 의 텍스트란에 텍스트를 몇몇 HTML 태그와 함께 입력할 수도 있고, 전체 HTML 페이지를 작성할 수도 있습니다.
- 현재 연도를 표시하고 싶은 경우, *Insert or Edit an Expression*…버튼을 클릭해서 소스 텍스트란에 `[%Year($now)%]` 같은 표현식을 추가하면 됩니다. 이 버튼은 *Source* 라디오 버튼을 선택한 경우에만 활성화됩니다. 표현식을 삽입한 후 HTML 프레임을 새로고침하기 전에 텍스트란 안쪽을 아무데나 한 번 클릭하십시오. 그러지 않으면 표현식이 사라질 것입니다.

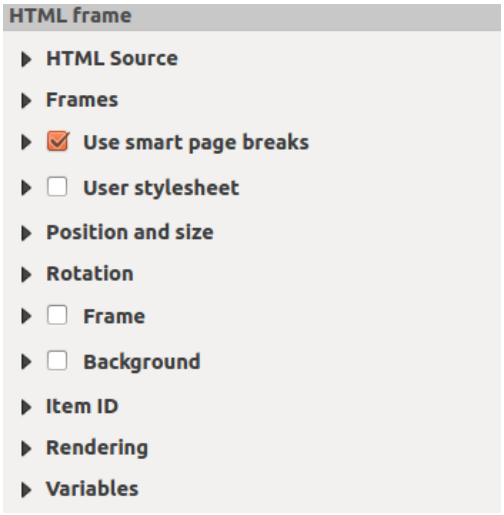



그림 18.57: HTML 프레임 항목 속성 패널

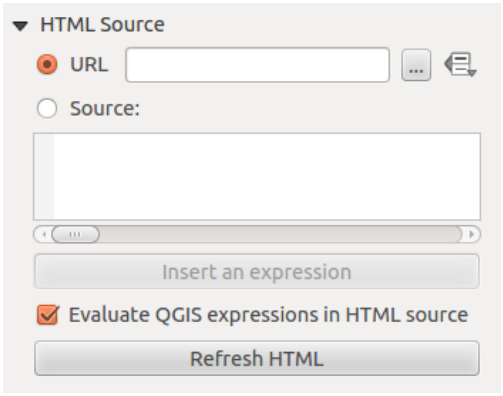



그림 18.58: HTML 프레임 항목의 HTML 소스 속성 그룹

- *Evaluate QGIS expressions in HTML code* 옵션을 활성화하면 사용자가 삽입한 표현식의 산출물을 볼 수 있습니다. 활성화하지 않으면 표현식 자체가 표시될 것입니다.
- *Refresh HTML* 버튼을 클릭하면 HTML 프레임 (들) 을 새로고침해서 변경 사항의 결과를 볼 수 있습니다.

### 프레임

HTML 프레임 *Item Properties* 패널의 *Frames* 그룹은 다음 기능을 제공하고 있습니다. (그림 18.59 을 참조하세요.):

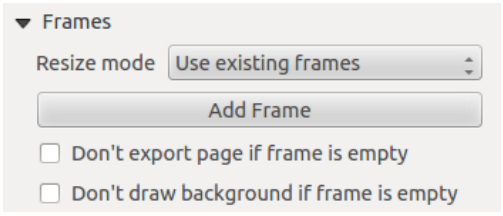



그림 18.59: HTML 프레임 항목의 프레임 속성 그룹

- *Resize mode* 옵션에서 HTML 의 내용을 어떻게 렌더링할지 선택할 수 있습니다:
  - `Use existing frames`: 첫 번째 프레임과 추가된 프레임에만 산출물을 표시합니다.
  - `Extend to next page`: 웹 페이지의 높이를 렌더링하는 데 필요한 만큼 많은 프레임 (과 대응하는 페이지) 을 생성합니다. 조판 상에서 각 프레임을 이리저리 옮길 수 있습니다. 프레임의 크기를 조정하는 경우, 웹 페이지가 다른 프레임들 사이로 나뉠 것입니다. 마지막 프레임은 웹 페이지에 맞춰 빈 부분을 잘라냅니다.
  - `Repeat on every page`: 모든 페이지에서 웹 페이지의 좌상단을 동일한 크기의 프레임 안에 반복해서 렌더링합니다.
  - `Repeat until finished`: 'Extend to next page'옵션과 마찬가지로 필요한 만큼 프레임을 생성합니다. 다만 모든 프레임의 크기가 동일합니다.
- *Add Frame* 버튼을 클릭하면 선택한 프레임과 동일한 크기의 또다른 프레임을 추가합니다. *Resize mode* 또는 `Use existing frames` 를 선택한 경우, 첫 번째 프레임에 다 들어가지 않은 HTML 페이지가 이 다음 프레임으로 넘어갑니다.
- *Don't export page if frame is empty* 옵션을 활성화하면, 프레임에 아무 HTML 내용물도 없을 때 페이지를 내보낼 수 없게 됩니다. 즉 산출물에 다른 모든 조판기 항목들 — 맵, 축척 막대, 범례 등등 — 이 보이지 않을 것이라는 뜻입니다.
- *Don't draw background if frame is empty* 옵션을 활성화하면, HTML 프레임에 아무 내용도 없을 때 프레임을 렌더링할 수 없게 됩니다.

### 스마트 페이지 나누기 사용 및 사용자 스타일시트

HTML 프레임 *Item Properties* 패널의 *Use smart page breaks* 및 *User style sheet* 그룹은 다음 기능을 제공하고 있습니다. (그림 18.60 을 참조하세요.):

- *Use smart page breaks* 옵션을 활성화하면 HTML 프레임의 내용이 텍스트 줄 중간에 나뉘는 일을 방지해서 다음 프레임으로 보기 좋고 매끄럽게 이어지게 해줍니다.
- HTML 내부의 어디에서 페이지를 나눌 것인지 계산하는 경우 *Maximum distance* 를 설정할 수 있습니다. 이 길이는 페이지 나누기 최적 위치를 계산한 후 프레임 하단에 허용되는 최장 빈 공간을 뜻합니다. 값을 크게 설정할수록 페이지 나누기 위치를 더 잘 선택할 수 있지만, 프레임 하단에 낭비되는 공간도 더 많아질 것입니다. *Use smart page breaks* 옵션을 활성화한 경우에만 이 길이를 설정할 수 있습니다.

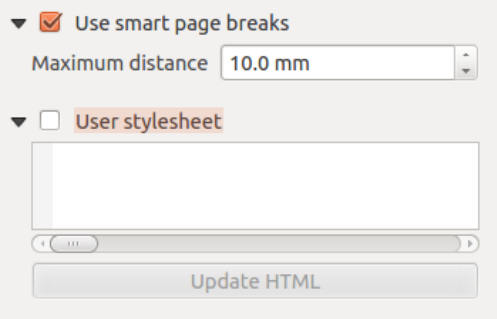



그림 18.60: HTML 프레임 항목의 스마트 페이지 나누기 사용 및 사용자 스타일시트 속성 그룹

- *User style sheet* 옵션을 활성화하면 주로 CSS(Cascading Style Sheet) 가 제공하는 HTML 스타일을 적용할 수 있습니다. 다음은 `<h1>` 헤더 태그의 색상을 초록색으로 설정하고 `<p>` 문단 태그에 포함된 텍스트의 글꼴 및 글꼴 크기를 설정하는 스타일 코드의 예시입니다.

```
h1 {color: #00ff00;
}
p {font-family: "Times New Roman", Times, serif;
   font-size: 20px;
}
```

- *Update HTML* 버튼을 클릭면 스타일시트 설정의 결과를 볼 수 있습니다.

### 18.2.11 도형 항목

QGIS 는 인쇄 조판 위에 정규 도형 또는 더 복잡한 도형을 그릴 수 있는 몇몇 도구를 제공하고 있습니다.

> **참고**
>
> 다른 인쇄 조판기 항목과는 달리, 프레임에 스타일을 적용하거나 도형 경계 프레임의 배경 색상을 설정할 수 없습니다. (기본적으로 투명하게 설정돼 있습니다.)

#### 정규 도형 항목

*Shape* 항목은 사용자 맵을 삼각형, 사각형, 타원 등과 같은 정규 도형으로 장식할 수 있는 도구입니다. Add Shape 도구를 통해 정규 도형을 추가할 수 있습니다. 이 도구에서 각각 Add Rectangle, Add Ellipse 그리고 Add Triangle 같은 특정 도구를 사용할 수 있습니다. 적절한 도구를 선택한 다음, 항목 생성 지침 을 따라 항목을 그릴 수 있습니다. 다른 조판기 항목과 마찬가지로, 대화형 조판기 항목 작업 에서 설명한 내용과 동일한 방식으로 정규 도형을 수정할 수 있습니다.

> **참고**
>
> 기본 도형 생성 시 `Shift` 키를 누른 채 클릭 & 드래그하면, 완벽한 원, 정사각형, 또는 정삼각형을 생성할 수 있습니다.

도형 *Item Properties* 패널에서 기본 도형 항목을 사용자 지정할 수 있습니다. 이 항목은공통 속성 이외에도 다음과 같은 기능을 가지고 있습니다 (그림 18.61 참조):

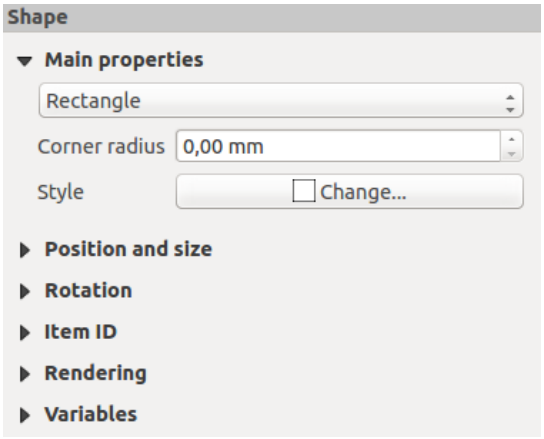



그림 18.61: 도형 항목 속성 패널

*Main properties* 그룹의 지정 프레임 안에 도형 항목 유형을 표시하며, 사용자가 각 (**Ellipse**, **Rectangle** 또는 **Triangle**) 유형을 전환할 수 있습니다.

고급심볼 및색상 선택기 위젯을 통해 도형의 스타일을 설정할 수 있습니다.

직사각형 도형의 경우, *Corner radius* 값을 다양한 단위로 설정할 수 있습니다. 모서리가 둥글어집니다.

### 노드 기반 도형 항목

*Add Shape* 도구가 단순하고 사전 정의된 도형 항목을 생성할 수 있게 해준다면, *Add Node Item* 도구는 좀 더 높은 수준의 사용자 지정 도형 항목을 생성할 수 있게 해줍니다. 폴리라인이나 폴리곤의 경우, 라인 또는 변 (side)을 원하는 만큼 많이 그릴 수 있고 *Edit Nodes Item* 도구를 통해 항목의 꼭짓점을 직접 개별적으로 수정할 수 있습니다. 대화형 조판기 항목 작업 에서 설명한 내용대로 도형 항목 자체도 수정할 수 있습니다.

노드 기반 도형을 추가하려면:

1. Add Node Item 아이콘을 클릭하십시오.
2. Add Polygon 또는 Add Polyline 도구 가운데 하나를 선택하십시오.
3. 왼쪽 클릭을 연속해서 사용자 도형에 노드를 추가하십시오. 선분을 그릴 때 `Shift` 키를 누르고 있으면, 45° 의 배수 방향을 따르도록 제한됩니다.
4. 작업을 종료했다면, 오른쪽 클릭으로 도형 생성을 끝내십시오.

노드 기반 도형의 *Item Properties* 패널에서 도형의 모양을 사용자 지정할 수 있습니다.

*Main properties* 부분에서, 고급심볼 및색상 선택기 위젯을 통해 도형의 스타일을 설정할 수 있습니다.

폴리라인 노드 기반 도형 항목의 경우, *Line markers* 를, 예를 들어 다음을 추가해서 파라미터화할 수도 있습니다:

- 시작 그리고/또는 종단 마커를 다음 옵션으로 추가합니다:
  - *None*: 단순 폴리라인을 그립니다.
  - *Arrow*: 사용자 지정할 수 있는 정규 삼각형 화살촉을 추가합니다.
  - *SVG* 마커: `.svg` 파일을 항목의 화살촉으로 사용합니다.

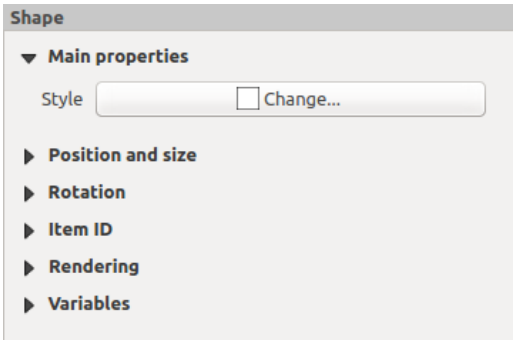



그림 18.62: 폴리곤 노드 도형 항목 속성 패널

- 화살촉을 사용자 지정합니다:
  - *Arrow stroke color*: 화살촉의 획 색상을 설정합니다.
  - *Arrow fill color*: 화살촉의 채우기 색상을 설정합니다.
  - *Arrow stroke width*: 화살촉의 획 너비를 설정합니다.
  - *Arrow head width*: 화살촉의 크기를 설정합니다.

SVG 이미지는 자동적으로 라인의 방향을 따라 기울어집니다. 대응하는 각각의 옵션으로 QGIS 사전 정의 SVG 이미지의 획 및 채우기 색상을 변경할 수 있습니다. 사용자 지정 SVG 는 다음지침 에 따라 태그를 몇 개 추가해야 할 수도 있습니다.

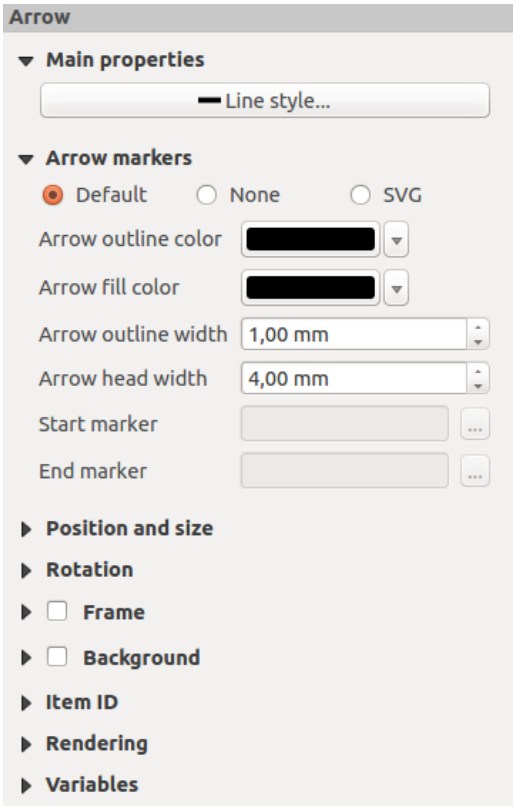



그림 18.63: 폴리라인 노드 도형 항목 속성 패널

### 화살표 항목

Add Arrow 도구는 폴리라인을 기본적으로 화살표를 활성화한 상태로 생성하는 단축키입니다. 즉폴리라인 노드 항목 과 동일한 속성과 습성을 가지고 있다는 의미입니다.

실제로, 예를 들어 서로 다른 인쇄 조판기 항목 2 개의 관계를 나타내기 위한 단순한 화살표를 추가하는 데 화살표 항목을 사용할 수 있습니다. 하지만, 방위표를 생성하고자 하는 경우 먼저이미지 항목 을 고려해야 합니다. 맵 항목과 동기화시켜 맵 항목을 따라 자동적으로 기울어지는 `.svg` 포맷의 방위표 집합 가운데 선택할 수 있기 때문입니다.

### 노드 항목 도형 편집하기

Edit Nodes Item 버튼을 클릭하면 노드 기반 도형을 편집할 수 있는 특수 모드로 전환됩니다. 이 모드를 활성화하면, 노드를 클릭해서 선택할 수 있습니다. (선택한 노드에는 마커가 표시됩니다.) 선택한 노드는 클릭 & 드래그해서 또는 방향키를 이용해서 이동시킬 수 있습니다. 또한 이 모드에서는 기존 도형에 노드를 추가할 수도 있습니다. 선분을 더블클릭하면 노드가 클릭한 위치에 추가됩니다. 마지막으로, `Del` 키를 누르면 현재 선택한 노드를 제거할 수 있습니다.

## 18.3 출력물 생성 작업

그림 18.64 는 앞에서 설명한 조판기 항목의 모든 유형을 포함하고 있는 인쇄 조판 예시를 보여주고 있습니다.

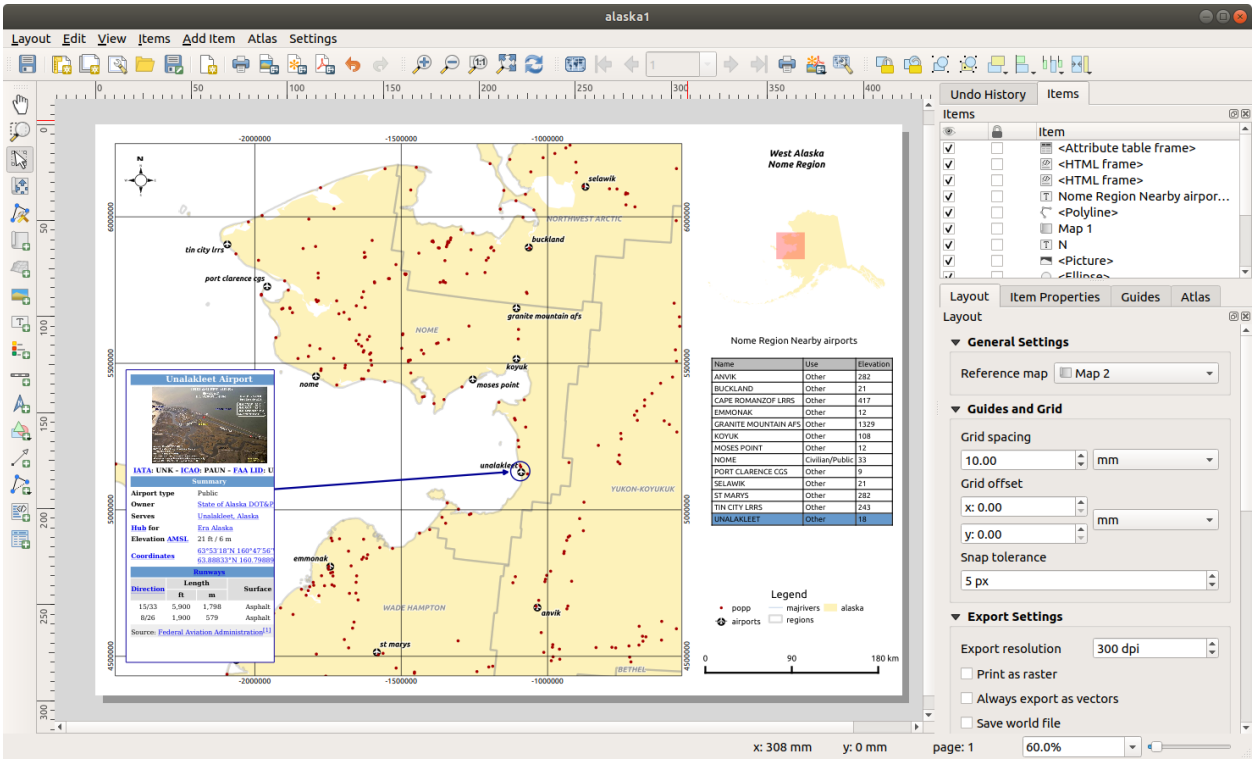



그림 18.64: 맵 뷰, 범례, 이미지, 축척 막대, 좌표, 텍스트 및 HTML 프레임을 추가한 인쇄 조판기

*Layout* 메뉴 또는 툴바에서 인쇄 조판을 서로 다른 파일 포맷으로 내보낼 수 있고, 그 해상도 (인쇄 품질) 및 용지 크기도 수정할 수 있습니다:

- Print 아이콘을 클릭하면 설치된 프린터 드라이버에 따라 조판을 연결된 프린터 또는 PostScript 파일로 인쇄할 수 있습니다.
- Export as image 아이콘을 클릭하면 인쇄 조판을 `PNG`, `BMP`, `TIF`, `JPG` 등과 같은 여러 이미지 포맷으로 내보낼 수 있습니다.
- Export as SVG 아이콘을 클릭하면 인쇄 조판을 `SVG` (Scalable Vector Graphic) 파일로 저장할 수 있습니다.
- Export as PDF 아이콘을 클릭하면 정의된 인쇄 조판을 `PDF` (Portable Document Format) 파일로 직접 저장할 수 있습니다.

### 18.3.1 내보내기 설정

인쇄 조판을 내보낼 때마다 QGIS 가 가장 적절한 산출물을 생성하기 위해 점검해야 하는 내보내기 설정 집합이 있습니다. 다음과 같은 환경 설정들입니다:

- *Export resolution* 내보내기 해상도, *Print as raster* 래스터로 인쇄, *Always export as vectors* 항상 벡터로 내보내기, 또는 *Save world file* 월드 파일 저장 같은, *Layout* 패널의내보내기 설정
- 페이지 항목 속성 패널의 *Exclude page from exports* 내보내기에서 선택 페이지 제외 설정
- 항목 속성 패널의 *Exclude item from exports* 내보내기에서 선택 항목 제외 설정

이에 더해, 조판에 몇 가지 사전 정의된 점검을 자동으로 적용합니다. 현재 이런 점검에는 맵 항목에 축척 막대가 정확하게 연결되었는지 테스트하기, 그리고 맵에 맵 오버뷰 항목도 정확하게 연결되었는지 테스트하기가 포함되어 있습니다. 이 점검이 실패하는 경우, 사용자에게 문제점을 알려주는 경고가 표시됩니다.

### 18.3.2 이미지로 내보내기

조판을 이미지로 내보내려면:

1. Export as image 아이콘을 클릭하십시오.
2. 조판을 내보내는 데 사용할 이미지 포맷, 폴더 및 파일명 (예: `myill.png`) 을 선택하십시오. 조판이 여러 페이지를 담고 있는 경우, 각 페이지를 사용자가 지정한 파일명 뒤에 페이지 번호를 붙인 파일로 (예: `myill_2.png`) 내보낼 것입니다.
3. 다음 (*Image Export Options*) 대화창에서:
   - 인쇄 조판의 *Export resolution* 과 (*Layout* 패널에서 설정하는) 내보내는 페이지의 크기 (dimensions) 를 무시할 수 있습니다.
   - For JPEG and JPG image formats, you can set the *JPEG Quality* of the output image.
   - *Enable antialiasing* 옵션으로 이미지 렌더링도 향상시킬 수 있습니다.
   - 사용자의 조판을 (예를 들면 다른 프로젝트와 공유하기 위해) **지리 참조된 (georeferenced) 이미지** 로 내보내고자 하는 경우 *Generate world file* 옵션을 체크하면, 내보내기 할 때 내보내는 이미지와 기반 이름은 같지만 확장자는 다른 (TIFF 는 `.tfw`, PNG 는 `.pnw`,JPEG 은 `jgw` 등등) *ESRI* 월드 파일 을 생성할 것입니다. 조판기 패널 에서 이 옵션을 기본값으로 체크할 수도 있습니다.

> **참고**
>
> 산출물이 여러 페이지인 경우, (*Generate world file* 옵션을 활성화했다면) 참조 맵 을 담고 있는 페이지만 월드 파일을 얻게 됩니다.

- *Crop to content* 옵션을 활성화하면, 조판기가 각 구성 페이지의 모든 항목들을 (맵, 범례, 축척 막대, 도형, 라벨, 이미지 등등) 감싸는 최소 영역만 이미지로 출력할 것입니다:
  - 조판이 단일 페이지를 담고 있는 경우, 조판에 있는 **모든 것** 을 포함하도록 산출물의 크기를 조정합니다. 페이지 크기도 모든 항목의 위치 (페이지 위 또는 페이지의 위쪽, 아래쪽, 왼쪽 또는 오른쪽) 에 따라 축소되거나 확대될 수 있습니다.
  - 조판이 여러 페이지인 경우, 각 페이지의 영역 (모든 페이지의 왼쪽 및 오른쪽, 그리고 첫 페이지의 위쪽과 마지막 페이지의 아래쪽) 에 있는 항목들을 포함하도록 각 페이지의 크기를 조정할 것입니다. 크기를 조정한 각 페이지는 개별 파일로 내보내집니다.

  *Crop to content* 대화창에서, 잘라낸 경계 주변에 여백을 추가할 수도 있습니다.
- *Open file after exporting* 옵션을 체크하면 인쇄 조판을 내보낸 뒤 내보낸 파일을 기본 이미지 뷰어로 열어볼 것입니다.

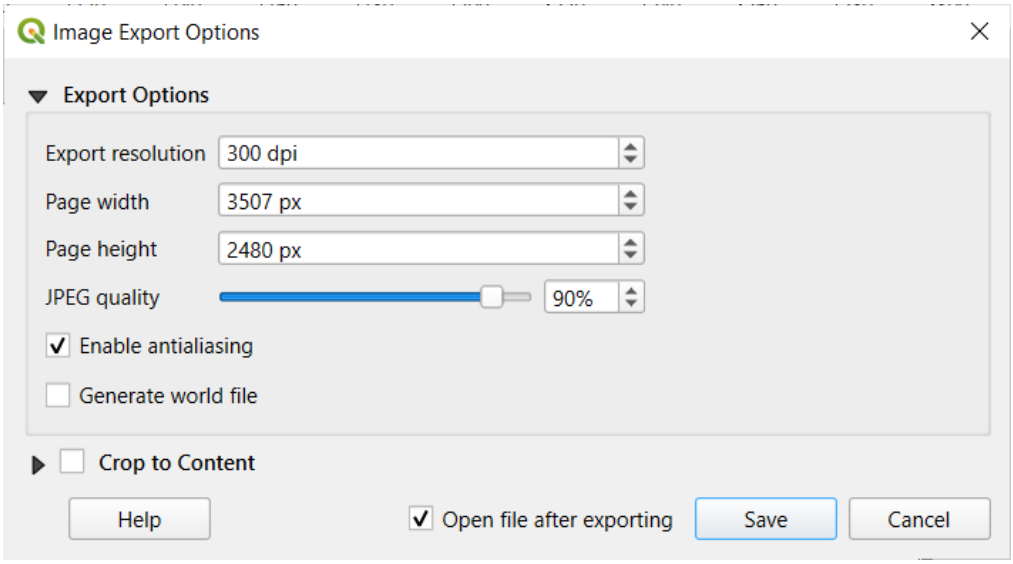



그림 18.65: 이미지 내보내기 옵션

> **팁**
>
> **항목이 용지 범위를 넘어가는 경우 투명도를 지원하는 이미지 포맷 사용**
>
> 조판기 항목이 용지 범위 바깥에 배치될 수도 있습니다. 그래서 *Crop to content* 옵션으로 내보내기할 때, 산출 이미지가 용지 범위를 벗어날 수도 있습니다. 용지 범위 바깥의 배경이 투명할 것이기 때문에, 투명도를 지원하지 않는 (예: `BMP` 및 `JPG`) 이미지 포맷의 경우 투명한 배경을 검은색으로만 렌더링합니다. 이미지를 "망치게"되죠. 이런 경우 투명도를 지원하는 (예: `TIFF` 및 `PNG`) 포맷을 사용하십시오.

> **참고**
>
> 포맷과 (예: `PNG`) 기저 Qt 라이브러리가 지원하는 경우, 내보내는 이미지가프로젝트 메타데이터 (작성자, 제목, 날짜, 설명 등등) 를 포함할 수도 있습니다.

### 18.3.3 SVG 로 내보내기

조판을 SVG 로 내보내려면:

1.  Export as SVG 아이콘을 클릭하십시오.
2. 경로 및 파일명을 입력하십시오. (이미지 내보내기와 마찬가지로, 여러 페이지로 구성돼 있는 경우 모든 파일의 기본명으로 사용됩니다.)
3. 그 다음 *SVG Export Options* 대화창에서 조판기의 기본내보내기 설정 을 무시하거나 또는 새로 환경 설정할 수 있습니다:
   -  *Export map layers as SVG groups*: 이 옵션을 활성화하면 QGIS 의 레이어명과 일치하는 이름을 가진 레이어 안에 항목들을 그룹화해서 내보내서, 문서의 내용을 더 쉽게 이해할 수 있도록 해줍니다.
   -  *Always export as vectors*: 일부 렌더링 옵션은 더 나은 렌더링을 위해 항목을 래스터화해야 합니다. 이 옵션을 체크하면, 산출물 파일의 모습이 인쇄 조판 미리보기와 달라질 위험을 무릅쓰고 객체를 벡터로 유지합니다. (더 자세한 내용은내보내기 설정 을 참조하세요.)
   -  *Export RDF metadata*: 문서의 제목, 작성자, 날짜, 설명 등등을 RDF 메타데이터로 내보냅니다.
   -  *Simplify geometries to reduce output file size*: 이 옵션은 도형의 꼭짓점을 **모두** 내보내지 않습니다. 그렇게 할 경우 내보내기 파일이 말도 안 될 정도로 복잡한 대용량이 되어 다른 응용 프로그램에서 열지 못 하게 될 수도 있기 때문입니다. 조판을 내보내는 도중, 내보내기 해상도에서 눈에 띄지 않는 불필요한 꼭짓점들을 제거해서 도형을 단순화시킬 것입니다. (예를 들어 내보내기 해상도가 `300 dpi` 인 경우 서로의 거리가 `1/600 inch` 미만인 꼭짓점을 제거할 것입니다.)
   - Set the *Text export*: controls whether text labels are always or preferably exported as *text or outline objects*.
   -  *Crop to content* 옵션 을 적용하십시오.
   -  *Disable tiled raster layer exports*: 파일을 내보낼 때, QGIS 는 메모리를 절약하기 위해 내장된 래스터 레이어 타일 렌더링을 사용합니다. 이 렌더링 방법은 종종 생성된 파일의 래스터에 눈에 보이는“이음매 (seam)”를 만들기도 합니다. 이 옵션을 체크하면 이런 문제를 해결할 수 있습니다. 내보내기 작업 도중 메모리를 더 많이 사용하긴 하지만 말이죠.
   -  *Open file after exporting* 옵션을 체크하면 인쇄 조판을 내보낸 뒤 내보낸 파일을 기본 SVG 뷰어로 열어볼 것입니다.

> **참고**
>
> 현재 SVG 출력물은 매우 기초적입니다. 이것은 QGIS 의 문제가 아니라, 기저 Qt 라이브러리의 문제입니다. Qt 향후 버전에서 해결되기를 바라고 있습니다.

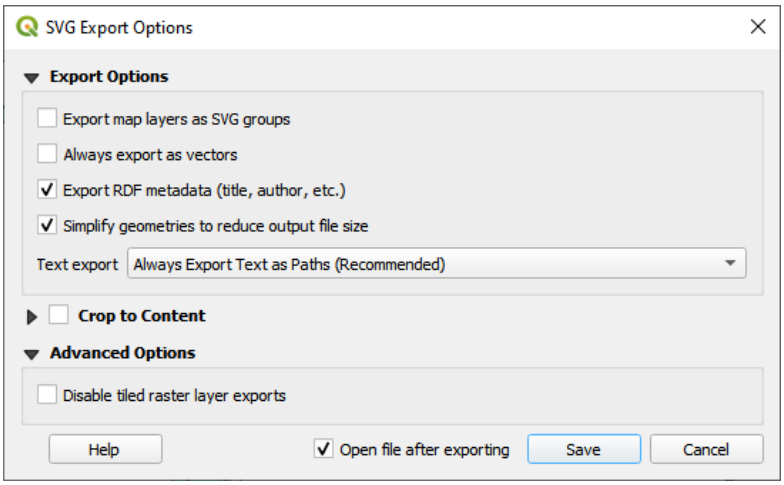



그림 18.66: SVG 내보내기 옵션

### 18.3.4 PDF 로 내보내기

조판을 PDF 로 내보내려면:

1. Export as PDF 아이콘을 클릭하십시오.
2. 경로 및 파일명을 입력하십시오: 이미지 및 SVG 내보내기와는 달리, 조판의 모든 페이지를 단일 PDF 파일로 내보냅니다.
3. 그 다음 *PDF Export Options* 대화창에서 조판기의 기본내보내기 설정 을 무시하거나 또는 새로 환경 설정할 수 있습니다:
   - *Always export as vectors*: 일부 렌더링 옵션은 더 나은 렌더링을 위해 항목을 래스터화해야 합니다. 이 옵션을 체크하면, 산출물 파일의 모습이 인쇄 조판 미리보기와 달라질 위험을 무릅쓰고 객체를 벡터로 유지합니다. (더 자세한 내용은내보내기 설정 을 참조하세요.)

     > **참고**
     >
     > **Mind opacity settings for vector layers**
     >
     > When you set a layer-wide opacity (or opacity for any vector symbology) to less than 100%, it forces a rasterized rendering of those objects during PDF exports. Depending on the size of the layer, this may considerably increase the PDF file size.

   - *Append georeference information*: 이 옵션은 정보를 가져오는참조 맵 이 첫 페이지에 있을 경우에만 사용할 수 있습니다.
   - *Export RDF metadata*: 문서의 제목, 작성자, 날짜, 설명 등등을 RDF 메타데이터로 내보냅니다.
   - Set the *Text export*: controls whether text labels are always or preferably exported as *text or outline objects*.
   - *Image compression*: PDF 의 이미지 압축 모드는:
     - *Lossy (JPEG)*: 기본 압축 모드입니다.
     - *Lossless*: 대부분의 경우 더 큰 용량의 파일을 생성하지만, 인쇄물로 출력 또는 외부 응용 프로그램에서 후처리를 하는 경우 훨씬 적합한 모드입니다. (Qt 5.13 이후 버전이 필요합니다.)

- *Create Geospatial PDF*: 이 옵션을 활성화하면 지리참조 PDF 파일 을 생성합니다.
- *Disable tiled raster layer exports*: 파일을 내보낼 때, QGIS 는 메모리를 절약하기 위해 내장된 타일 기반 렌더링을 사용합니다. 이 렌더링 방법은 종종 생성된 파일의 래스터에 눈에 보이는"이음매 (seam)"를 만들기도 합니다. 이 옵션을 체크하면 이런 문제를 해결할 수 있습니다. 내보내기 작업 도중 메모리를 더 많이 사용하긴 하지만 말이죠.
- *Simplify geometries to reduce output file size*: 조판을 내보내는 도중, 내보내기 해상도에서 눈에 띄지 않는 불필요한 꼭짓점들을 제거해서 도형을 단순화시킬 것입니다. (예를 들어 내보내기 해상도가 `300 dpi` 인 경우 서로의 거리가 `1/600 inch` 미만인 꼭짓점을 제거할 것입니다.) 이렇게 하면 내보내기 파일의 용량 및 복잡도를 줄일 수 있습니다. (다른 응용 프로그램에서 대용량 파일을 불러올 수 없을 수도 있기 때문입니다.)
- *Open file after exporting* 옵션을 체크하면 인쇄 조판을 내보낸 뒤 내보낸 파일을 기본 PDF 뷰어로 열어볼 것입니다.

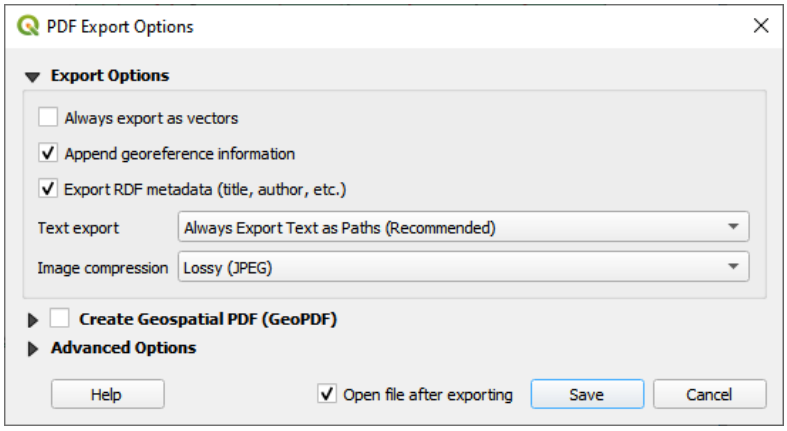



그림 18.67: PDF 내보내기 옵션

**참고**

지리공간 PDF 내보내기를 지원합니다. 다음과 같은 지리공간 PDF 특화 옵션을 사용할 수 있습니다:

- *Format*: 지리공간 PDF 포맷을 지정합니다 — 지리공간 PDF 포맷의 변종들이 몇 개 있습니다.
- *Include multiple map themes*: 포함시킬 맵 테마를 지정합니다.
- *Include vector feature information*: 레이어들을 선택하고 해당 레이어들을 논리적 PDF 그룹으로 그룹화합니다.

**참고**

인쇄 조판을 지리 참조를 지원하는 포맷으로 (예: `PDF` 및 `TIFF`) 내보내면 기본적으로 지리 참조된 산출물을 생성합니다.

**참고**

When exporting maps or layouts to vector formats such as PDF or SVG, line pattern fills are rendered using a line-by-

line vector-based approach instead of rasterized tiles. This change results in smaller and higher-quality output files, with individual pattern lines remaining fully editable in external vector graphics software.

### 18.3.5 지도책 생성

지도책 기능을 통해 지도책을 자동화된 방식으로 생성할 수 있습니다. 지도책은 테이블 또는 벡터 레이어 (*Coverage layer*) 의 피처를 이용해서 테이블/레이어에 각 피처 (**atlas feature**) 에 대한 산출물을 생성합니다. 가장 흔한 활용법은 맵 항목을 현재 지도책 피처로 확대/축소하는 것입니다. 더 심화된 활용 사례는 다음과 같은 것들이 있습니다:

- 지도책 피처와 동일한 속성을 공유하는, 또는 지도책 피처의 도형 내부에 있는, 또다른 레이어의 피처만 표시하는 맵 항목
- 피처들이 거듭 반복되면 텍스트가 대체되는 라벨 또는 HTML 항목
- 현재 지도책 피처의 관련상위 *(부모)* 또는 하위 *(자식)* 피처의 속성을 표시하는 테이블 항목

각 피처에 대해, 페이지 및 항목의 내보내기 설정에 따라 모든 페이지와 항목을 처리해서 산출물을 생성합니다.

**팁**

**유연성 향상을 위한 변수 사용**

QGIS 는 함수들과변수들 을 담고 있는 대형 패널을 제공하고 있습니다. 이 가운데 지도책 관련 함수 및 변수를 사용하면 조판기 항목은 물론, 지도책 상태에 따라 레이어의 심볼도 수정할 수 있습니다. 이 기능들을 결합하면 사용자가 높은 유연성을 발휘해서 고급 지도를 쉽게 생성할 수 있습니다.

지도책 생성 기능을 활성화하고 지도책 파라미터를 설정하려면, *Atlas* 패널을 사용하십시오. 이 패널은 다음과 같은 기능들을 담고 있습니다 (그림 18.68 참조):

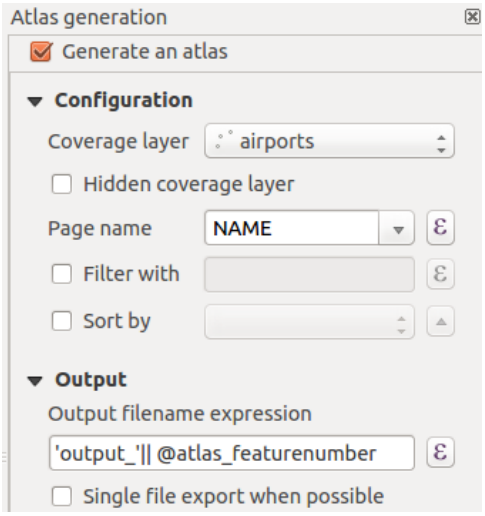



그림 18.68: 지도책 패널

- *Generate an atlas*: 지도책 생성 기능을 활성화/비활성화합니다.
- *Configuration*

  - *Coverage layer* : 이 콤보박스에서 생성 작업을 반복할 피처들을 담고 있는 테이블 또는 벡터 레이어를 선택할 수 있습니다.
  - *Hidden coverage layer*: 이 부가적인 체크박스를 활성화하면, 생성 도중 커버리지 레이어를 숨깁니다. (다른 레이어들은 숨기지 않습니다.)
  - *Page name*: 이 부가적인 콤보박스는 피처 페이지 (들) 를 위한 이름을 지정할 수 있습니다. 커버리지 레이어의 필드를 선택하거나, 표현식 을 설정할 수 있습니다. 이 옵션을 비워놓을 경우, QGIS 는 레이어에 적용된 필터 그리고/또는 정렬 순서에 따라 내부 ID 를 사용할 것입니다.
  - *Filter with*: 이 부가적인 텍스트란에 커버리지 레이어에 있는 피처를 필터링하기 위한 표현식을 작성할 수 있습니다. 표현식을 작성한 경우, `True` 로 평가된 피처만 처리할 것입니다.
  - *Sort by*: 이 부가적인 체크박스를 활성화하면, 커버리지 레이어의 필드 또는 표현식을 사용해서 커버리지 레이어 (및 산출물) 의 피처를 정렬할 수 있습니다. 클릭할 때마다 위를 가리키는 화살표 또는 아래를 가리키는 화살표로 바뀌는 정렬 방향 버튼으로 정렬 순서 (오름차순 또는 내림차순) 를 설정할 수 있습니다.

- *Output* 에서 지도책의 출력물을 환경 설정할 수 있습니다:
  - *Output filename expression*: 각 지도책 피처를 위한 파일명을 생성하는 데 쓰이는 표현식 기반 텍스트란입니다. 이 옵션은 여러 파일로 렌더링하는 경우에만 의미가 있습니다.
  - *Single file export when possible*: 선택한 출력물 포맷 (예: `PDF`) 으로 가능한 경우 단일 파일을 생성하도록 강제할 수 있습니다. 이 옵션을 활성화하면, *Output filename expression* 필드의 값이 의미없어집니다.
  - *Image export format*: Export atlas as Images… 버튼을 클릭한 경우 이 드롭다운 목록에서 산출물 포맷을 선택할 수 있습니다.

### 지도책으로 맵 제어

지도책의 가장 흔한 활용법은 커버리지 레이어를 반복해서 작업하는 동안 맵 항목을 현재 지도책 피처로 확대/축소하는 것입니다. 맵 항목의 *Controlled by atlas* 그룹 속성에서 이 습성을 설정할 수 있습니다. 맵 항목에 적용할 수 있는 서로 다른 설정에 대해서는지도책으로 제어 를 참조하세요.

### 표현식으로 라벨 사용자 지정

지도책이 반복해서 작업하는 피처의 라벨 내용을 조정하기 위해, 표현식을 포함시킬 수 있습니다. 사용자는 (함수, 필드, 또는 변수를 포함하는) 표현식 부분을 `[%` 와 `%]` 사이에 넣었는지 확인해야 합니다. (자세한 내용은라벨 항목 을 참조하세요.)

예를 들면, `CITY_NAME` 필드와 `ZIPCODE` 필드를 보유한 도시 레이어의 경우 다음과 같은 표현식을 삽입할 수 있습니다:

```
The area of [% concat( upper(CITY_NAME), ',', ZIPCODE, ' is ',
format_number($area/1000000, 2) ) %] km2
```

또는, 다음처럼 삽입할 수도 있죠:

```
The area of [% upper(CITY_NAME)%],[%ZIPCODE%] is
[%format_number($area/1000000,2) %] km2
```

`[% concat( upper(CITY_NAME), ',', ZIPCODE, ' is ', format_number($area/1000000, 2) ) %]` 라는 정보는 라벨 내부에 쓰이는 표현식입니다. 두 표현식 모두 생성된 지도책에서 다음 라벨 유형으로 출력될 겁니다:

```
The area of PARIS,75001 is 1.94 km2
```

### 지도책과 데이터 정의 무시 버튼 탐구

선택한 설정을 무시하기 위해 Data defined override 버튼을 사용할 수 있는 옵션이 여러 군데 있습니다. 이 옵션들은 지도책 생성 작업 시 특히 유용합니다. 이 위젯에 대한 자세한 내용은데이터 정의 무시 설정 을 참조하세요.

다음에 나올 예제들은 지도책 생성 작업에 QGIS 예시 데이터셋의 `Regions` 레이어를 선택해서 *Coverage layer* 로 사용하는, 맵 항목과 라벨 항목을 담고 있는 단일 페이지 조판이라고 가정합니다.

지역의 범위의 높이 (남-북) 가 너비 (동-서) 보다 더 긴 경우, 용지를 최적으로 활용하려면 랜드스케이프 *(Landscape)* 대신 포트레이트 *(Portrait)* 방향을 사용해야 합니다. Data Defined Override 버튼을 통해 용지 방향을 동적으로 설정할 수 있습니다.

페이지를 오른쪽 클릭하고 *Page Properties* 를 선택해서 패널을 여십시오. 표현식을 사용해서 지역 모양에 따라 용지 방향을 동적으로 설정하려 하기 때문에, *Orientation* 필드의 버튼을 클릭한 다음 *Edit…*를 선택하십시오. *Expression string builder* 대화창이 열리면 다음 표현식을 입력하십시오:

```
CASE WHEN bounds_width(@atlas_geometry) > bounds_height(@atlas_geometry)
THEN 'Landscape' ELSE 'Portrait' END
```

이제지도책을 미리보기 하면, 용지 방향 자체는 자동으로 변하지만 항목들의 배치가 이상하게 보일 수도 있습니다.

각 `Region` 마다 사용자가 조판기 항목의 위치도 다시 배치해야 합니다. 맵 항목의 경우, *Width* 속성의 버튼을 클릭해서 다음 표현식을 입력하면 맵 항목의 너비를 동적으로 설정할 수 있습니다:

```
@layout_pagewidth - 20
```

마찬가지로, *Height* 옵션의 버튼도 클릭해서 맵 항목의 크기를 제약하는 다음 표현식을 입력하십시오:

```
@layout_pageheight - 20
```

맵 항목이 페이지의 중심에 배치됐는지 확인하려면, 맵 항목의 *Reference point* 를 좌상단 라디오버튼으로 설정한 다음 *X* 및 *Y* 위치를 `10` 으로 입력하십시오.

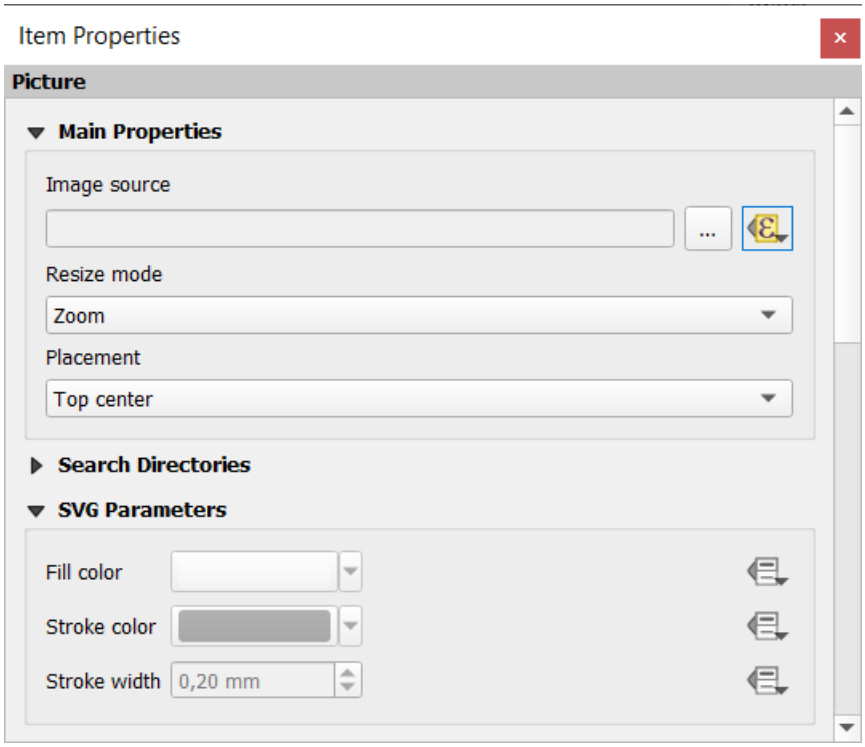
페이지 중심에 있는 맵 상단에 제목을 추가해봅시다. 라벨 항목을 선택하고 수평 정렬을 *Center* 로 설정하십시오. 그 다음 라벨을 올바른 위치로 이동시키고, *Reference point* 를 가운데 라디오 버튼으로 설정하십시오. 그리고 *X* 필드에 다음 표현식을 입력하십시오:

```
@layout_pagewidth / 2
```

다른 모든 조판기 항목에 대해 비슷한 방법으로 위치를 설정할 수 있습니다. 이렇게 동적으로 위치를 설정하면 용지 방향이 포트레이트든 랜드스케이프든 조판기 항목들이 정확히 배치됩니다. 또 제목을 피처 속성으로 사용자 지정하거나 (표현식으로 라벨 사용자 지정 예시 참조), 이미지를 변경하거나, 용지 방향에 따라 범례의 열 개수를 재조정하는 등 더 많은 미세 조정도 할 수 있습니다.

여기에 제공한 정보는 데이터 정의 무시 옵션에 대한 훌륭한 블로그 게시글 을 업데이트한 것입니다.

데이터 정의 무시 버튼을 사용하는 또다른 예는, 동적 그림의 활용법입니다. 다음 예시에서 바이너리 필드 유형인 `logo` 라는 이름의 BLOB(Binary Large Object) 필드를 담고 있는 GeoPackage 레이어를 사용합니다. (새 *GeoPackage* 레이어 생성하기 를 참조하세요.) 미리보기 및 지도책 생성 에서 설명한 대로 지도책이 반복되려면, 모든 객체에 대해 서로 다른 그림이 정의돼 있어야 합니다. 인쇄 조판기에 그림을 추가하고 지도책 맥락에서 그림의 *Item properties* 로 가십시오. *Main Properties* 의 *Image source* 부분에 데이터 정의 무시 버튼이 있을 것입니다.




다음 창에서 *Edit* 버튼을 누르면 *Expression String Builder* 대화창이 열립니다. *Fields and values* 부분에서 GeoPackage 레이어에 정의된 BLOB 필드를 찾을 수 있을 겁니다. `logo` 필드명을 더블클릭한 다음 *OK* 를 클릭하십시오.

사용자가 GeoPackage 레이어를 *Coverage layer* 로 선택해놨다면 지도책이 BLOB 필드의 항목 전체에 걸쳐 반복될 것입니다. (미리보기 및 지도책 생성 에서 더 자세한 지침을 찾아볼 수 있습니다)

이것은 지도책에 몇몇 고급 설정을 어떻게 활용할 수 있는지에 대한 2 개의 예시일 뿐입니다.

### 미리보기 및 지도책 생성

지도책 환경 설정을 마치고 지도책에 조판기 항목 (맵, 테이블, 이미지 등등) 을 연결하고 나면, *Atlas* → *Preview Atlas* 메뉴 옵션을 선택하거나 Preview Atlas 아이콘을 클릭해서 모든 페이지의 미리보기를 생성할 수 있습니다. 그 다음 다음 화살표를 클릭해서 모든 피처들을 탐색할 수 있습니다:

- First feature: 첫 피처로 돌아갑니다.
- Previous feature: 이전 피처로 돌아갑니다.
- Next feature: 다음 피처로 넘어갑니다.
- Last feature: 마지막 피처로 넘어갑니다.

콤보박스를 통해 특정 피처를 선택해서 미리보기할 수도 있습니다. 이 콤보박스는 지도책의 *Page name* 옵션에서 설정한 표현식에 따라 지도책 피처의 이름을 표시합니다.

단순한 조판의 경우, 지도책을 서로 다른 방법으로 생성할 수 있습니다. (자세한 정보는출력물 생성 작업 을 참조하세요.) *Layout* 메뉴 대신, 그냥 *Atlas* 메뉴 또는 지도책 툴바의 도구를 이용하십시오.

다시 말해 *Atlas* → *Print Atlas* 메뉴 옵션을 통해 사용자의 조판을 직접 인쇄할 수 있다는 뜻입니다. *Atlas* → *Export Atlas as PDF*…메뉴 옵션을 선택하면 PDF 도 생성할 수 있습니다. *Single file export when possible* 옵션을 활성화하지

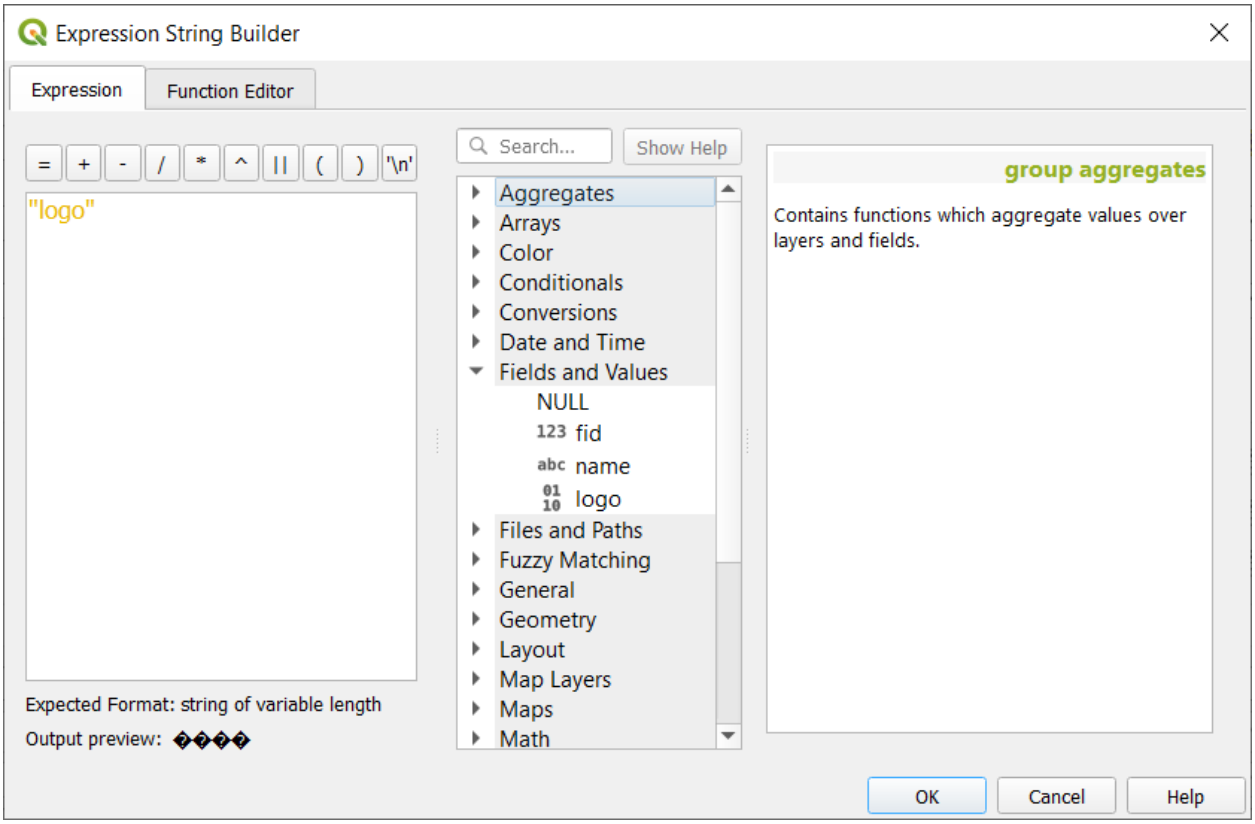






그림 18.69: 지도책 미리보기 툴바

않은 경우, 생성된 모든 PDF 파일을 저장할 디렉터리를 지정하도록 요청할 것입니다. 이 옵션을 활성화했다면, 파일명을 지정하도록 요청할 것입니다.

*Atlas → Export Atlas as Images…*또는 *Atlas → Export Atlas as SVG…*메뉴 옵션을 선택하면, 역시 폴더를 선택하도록 요청할 것입니다. 각 지도책 피처 조판의 각 페이지를 *Atlas* 패널에서 설정한 이미지 파일 포맷 또는 SVG 파일로 내보냅니다.

**참고**

산출물이 여러 페이지인 경우, 참조 맵 을 담고 있는 페이지만 월드 파일을 (각 피처 산출물마다) 얻게 된다는 점에서 지도책이 조판처럼 작동합니다.

**팁**

**특정 지도책 피처 인쇄**

지도책에서 오직 한 피처의 조판만을 인쇄하거나 내보내고자 하는 경우, 그냥 미리보기를 열고 드롭다운 목록에서 원하는 피처를 선택한 다음 *Layout → Print* 메뉴 옵션을 (또는 기타 지원 파일 포맷으로 내보내려면 *Export…*메뉴 옵션을) 선택하십시오.

### 지도책 생성에 프로젝트 정의 관계 사용

HTML 및 자바스크립트 지식을 가진 사용자라면 GeoJSON 객체를 작업할 수 있고 QGIS 프로젝트로부터 프로젝트 정의 관계도 사용할 수 있습니다. 이 접근 방식과 HTML 에 표현식을 직접 삽입하는 방식과의 차이는, 구조화되지 않은 완전한 GeoJSON 피처를 작업할 수 있다는 점입니다. 즉 GeoJSON 피처를 표현하는 기존 자바스크립트 라이브러리와 함수를 사용할 수 있다는 뜻이죠.

다음 코드는 정의된 관계에서 나온 모든 관련 하위 피처를 포함하고 있습니다. 자바스크립트의 `setFeature` 함수를 사용하면 사용자가 선호하는 어떤 (목록, 테이블 등) 유형으로든 관계를 표현하는 유연한 HTML 을 작성할 수 있습니다. 이 예시 코드는 관련 하위 피처의 동적인 구분점 (bullet) 목록을 생성합니다.

```
// Declare the two HTML div elements we will use for the parent feature id
// and information about the children
<div id="parent"></div>
<div id="my_children"></div>

<script type="text/javascript">
   function setFeature(feature)
   {
     // Show the parent feature's identifier (using its "ID" field)
     document.getElementById('parent').innerHTML = feature.properties.ID;
     //clear the existing relation contents
     document.getElementById('my_children').innerHTML = '';
     feature.properties.my_relation.forEach(function(child_feature) {
     // for each related child feature, create a list element
     // with the feature's name (using its "NAME" field)
       var node = document.createElement("li");
       node.appendChild(document.createTextNode(child_feature.NAME));
       document.getElementById('my_children').appendChild(node);
     });
   }
</script>
```

지도책 생성 도중 부모 피처를 담고 있는 커버리지 레이어를 반복해서 작업하게 될 것입니다. 각 페이지에 부모 피처의 식별자를 따르는 관련 하위 피처의 구분점 목록을 표시할 것입니다.

## 18.4 보고서 생성 작업

QGIS 에서 보고서를 설정하는 방법에 대해 설명합니다.

### 18.4.1 보고서란?

GIS 보고서란 당연히 서술적인 방식으로 정리된 정보를 맵, 텍스트, 그래픽, 표 등과 함께 담고 있는 문서를 말합니다. 즉석에서, 주기적으로, 반복적으로, 규칙적으로, 또는 필요할 때마다 보고서를 준비할 수 있습니다. 보고서는 특정한 기간, 사건, 출현, 주제 또는 위치를 참조할 수도 있습니다.

QGIS 에서 *Report* 는조판기 의 연장입니다.

보고서는 사용자가 자신의 GIS 프로젝트를 간결하고 구조적인 방식으로 출력할 수 있게 해줍니다.

*Project* → *New Report* 메뉴 또는 *Project* → *Layout Manager* 대화창에서 보고서를 생성할 수 있습니다.

> **참고**
>
> QGIS 보고서에 있는 맵도 인쇄 조판기와 지도책에 있는 맵과 동일한 방식으로 동작합니다. 여기에서는 QGIS 보고서의 세부 사항에 집중할 것입니다. 맵 처리에 대한 자세한 내용은인쇄 조판기 와지도책 을 참조하십시오.

### 18.4.2 시작하기

*Layout Manager* 대화창에서 *New from template* 드롭다운 목록의 *Empty Report* 를 선택한 다음 *Create…*버튼을 클릭하면 보고서를 생성할 수 있습니다.

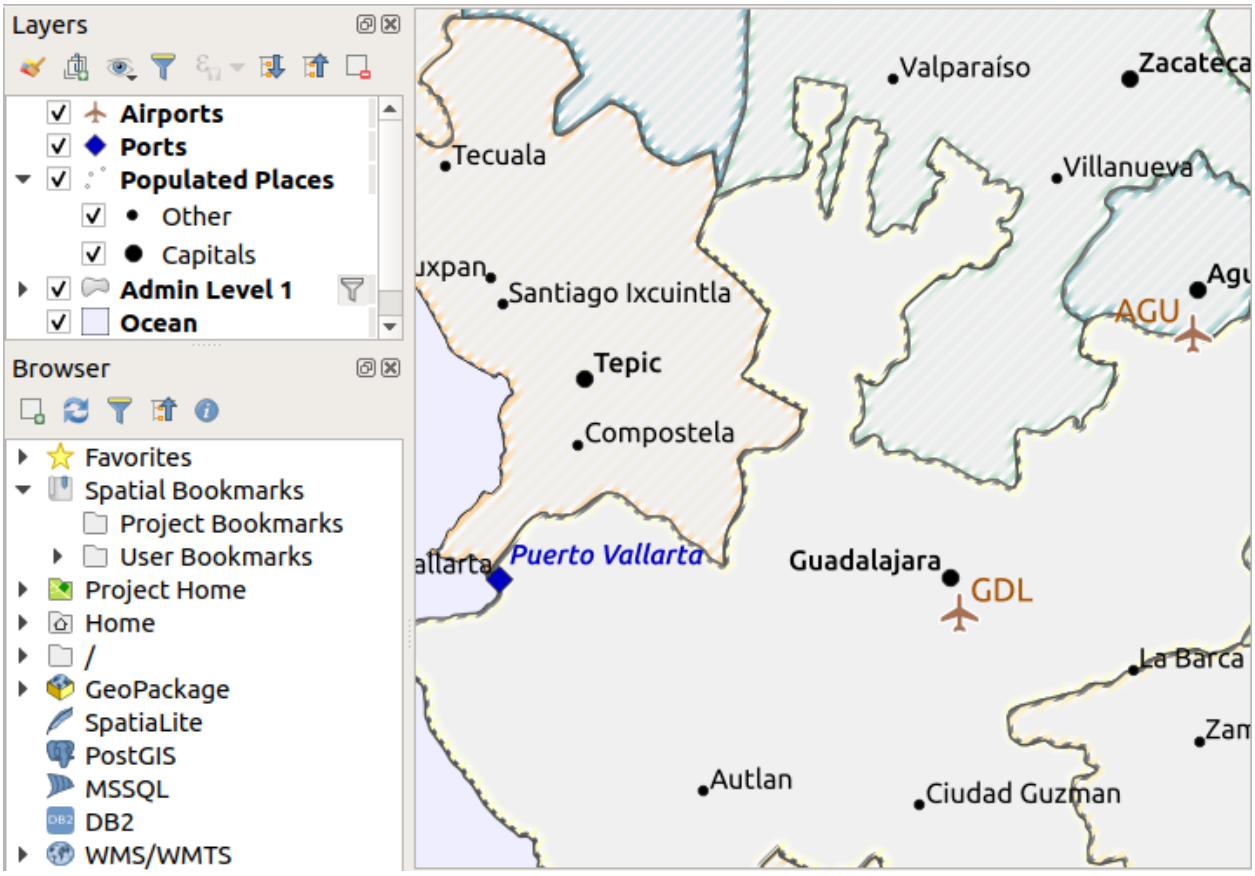
이 예시에서는 자연 지구 (Natural Earth) 데이터셋 (1:10,000,000) 의 몇몇 행정 경계, 인구 밀집 지역, 항구 및 공항을 사용합니다.

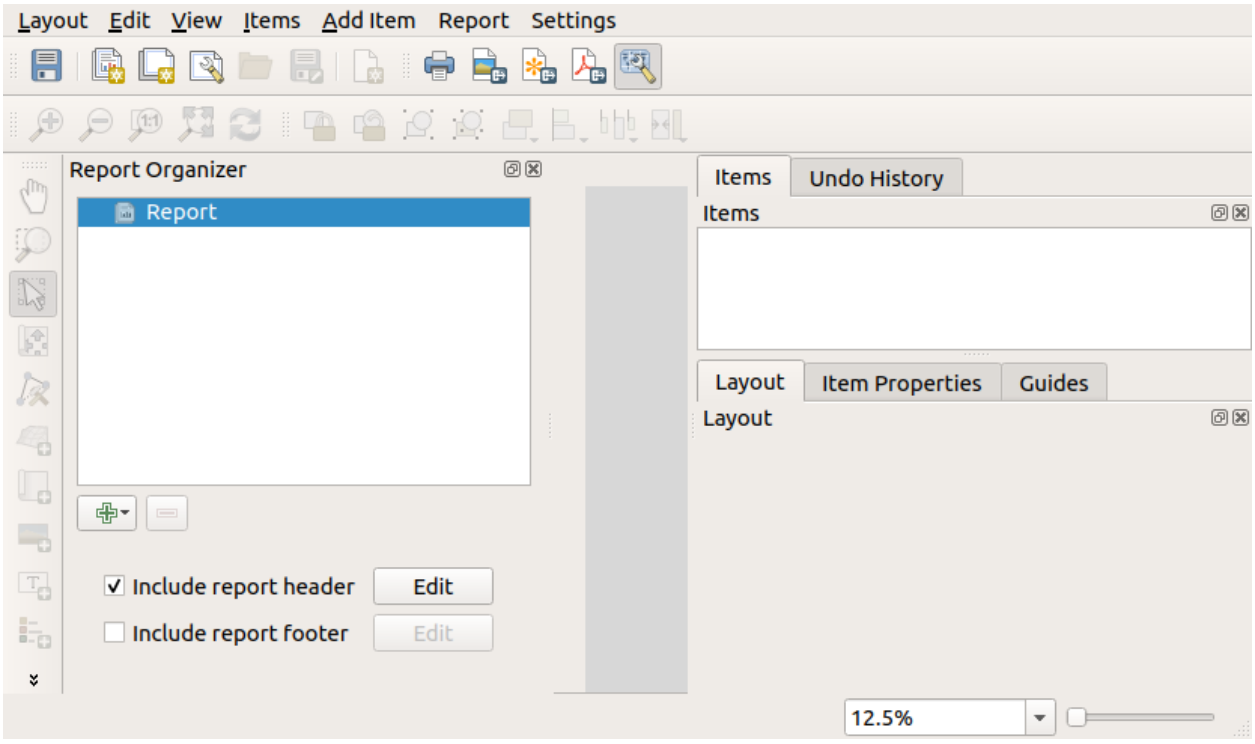
*Project* → *New Report* 메뉴 옵션을 선택하면, 비어 있는 보고서를 생성합니다. 처음에는 별로 볼 것도 없습니다 — 이 대화창은 왼쪽에 있는 *Report Organizer* 패널을 제외하면 인쇄 조판 설계자와 매우 비슷합니다:

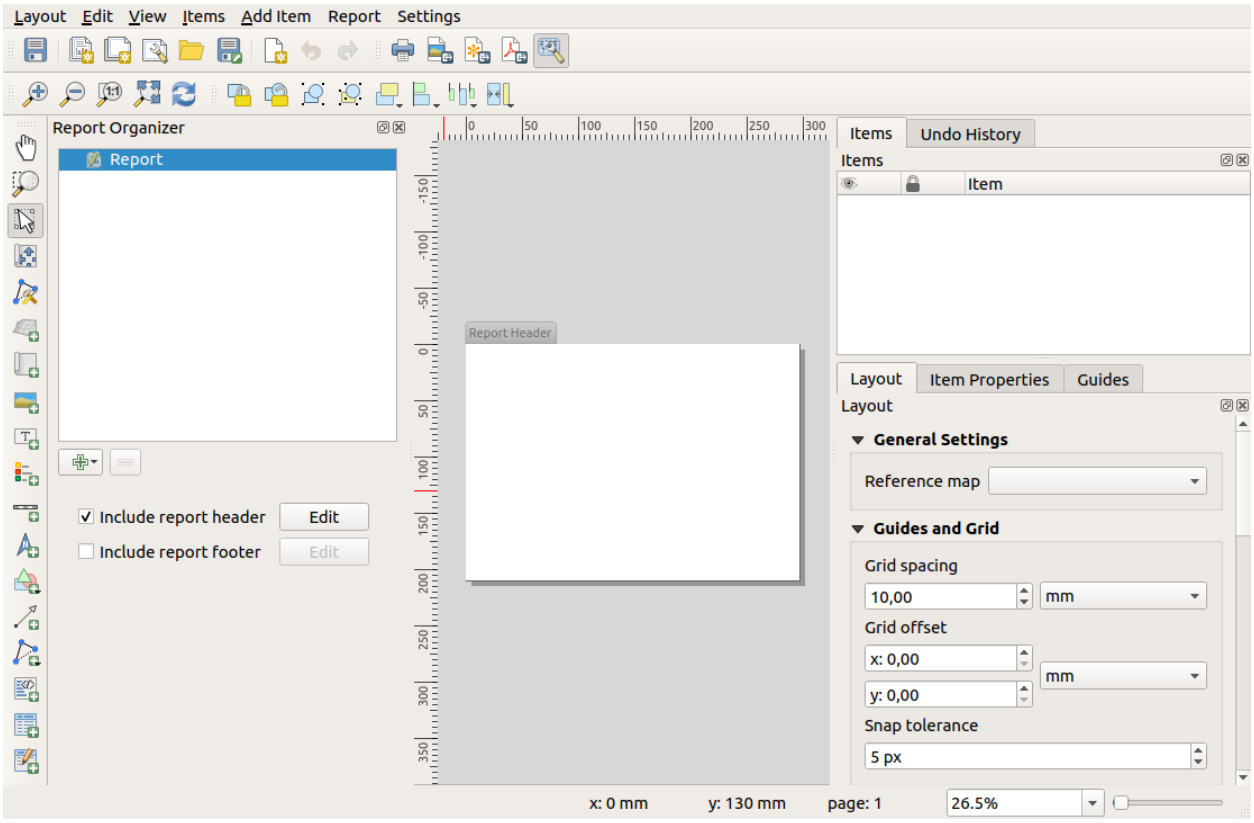
### 18.4.3 조판기 보고서 작업 공간

QGIS 보고서는 여러 개의 내포된 부분들로 이루어질 수 있습니다. 비어 있는 새 보고서는 처음에 주요 보고서 부분만 가지고 있습니다. 이 보고서 부분을 위한 옵션은 *Include report header* 와 *Include report footer* 뿐입니다. 이 옵션들을 활성화하면, 헤더가 보고서의 첫 번째 페이지 (들) 로 포함되고, 푸터가 마지막 페이지 (들) 를 구성할 것입니다. (필요한 경우 보고서의 개별 부분들을 여러 페이지로 만들 수 있습니다.) 헤더를 활성화하고 (*Include report header*) 옆에 있는 *Edit* 버튼을 클릭하십시오:

몇 가지 일이 일어납니다. 먼저, 설계자의 보고서 부분이 현재 편집중이라는 사실을 나타내는 편집 연필이 *Report Organizer* 의 *Report* 옆에 표시됩니다. 또 작은 *Report Header* 제목을 가진 새 페이지도 추가됩니다. 이 페이지는 기본적으로 랜드스케이프 방향이지만, 페이지를 오른쪽 클릭하고 *Page properties* 를 선택하면 이 속성을 (그리고 페이지의 다른 속성도) 변경할 수 있습니다. 이렇게 하면 페이지 *Item properties* 탭을 불러와서 페이지의 *Size*, *Width*, *Height* 는 물론 더 많은 속성을 설정할 수 있습니다.


Layers
Airports
Ports
Populated Places
Other
Capitals
Admin Level 1
Ocean
Browser
Favorites
Spatial Bookmarks
Project Bookmarks
User Bookmarks
Project Home
Home
/
GeoPackage
SpatiaLite
PostGIS
MSSQL
DB2
WMS/WMTS
Valparaíso
Zacatecas
Tecuala
Villanueva
Santiago Ixcuintla
AGU
Tepic
Compostela
Puerto Vallarta
Guadalajara
GDL
La Barca
Autlan
Ciudad Guzman


Layout
Edit
View
Items
Add Item
Report
Settings
Report Organizer
Report
Items
Undo History
Items
Layout
Item Properties
Guides
Layout
Include report header
Edit
Include report footer
Edit
12.5%




QGIS 보고서에서, 보고서의 모든 구성 요소는 개별 조판으로 만들어집니다. 표준 인쇄 조판과 동일한 도구를 사용해서 보고서의 모든 항목을 생성하고 수정할 수 있습니다. 즉 사용자가 원하는 모든 라벨, 그림, 맵, 표 등을 조합해서 사용할 수 있습니다. 예시를 위해 보고서 헤더에 몇몇 항목을 추가해봅시다:

*Include report footer* 옵션을 활성화하고 *Edit* 버튼을 클릭해서 보고서에 간단한 푸터도 생성하겠습니다.

다음 단계로 넘어가기 전에, 이 보고서를 내보내서 어떤 산출물이 나오는지 봅시다. *Report* 메뉴에서 내보내기를 할 수 있습니다 — 이 경우 *Export Report as PDF*…메뉴를 선택해서 전체 보고서를 PDF 파일로 렌더링하겠습니다. 다음은 헤더와 푸터로 이루어진, 그리 인상적이지도 않은 산출물인 두 페이지짜리 PDF 입니다:

좀 더 재미있는 보고서를 만들어볼까요. *Report Organizer* 에서 Add Section 버튼을 클릭하면, 보고서에 추가할 새로운 부분을 선택할 수 있습니다.

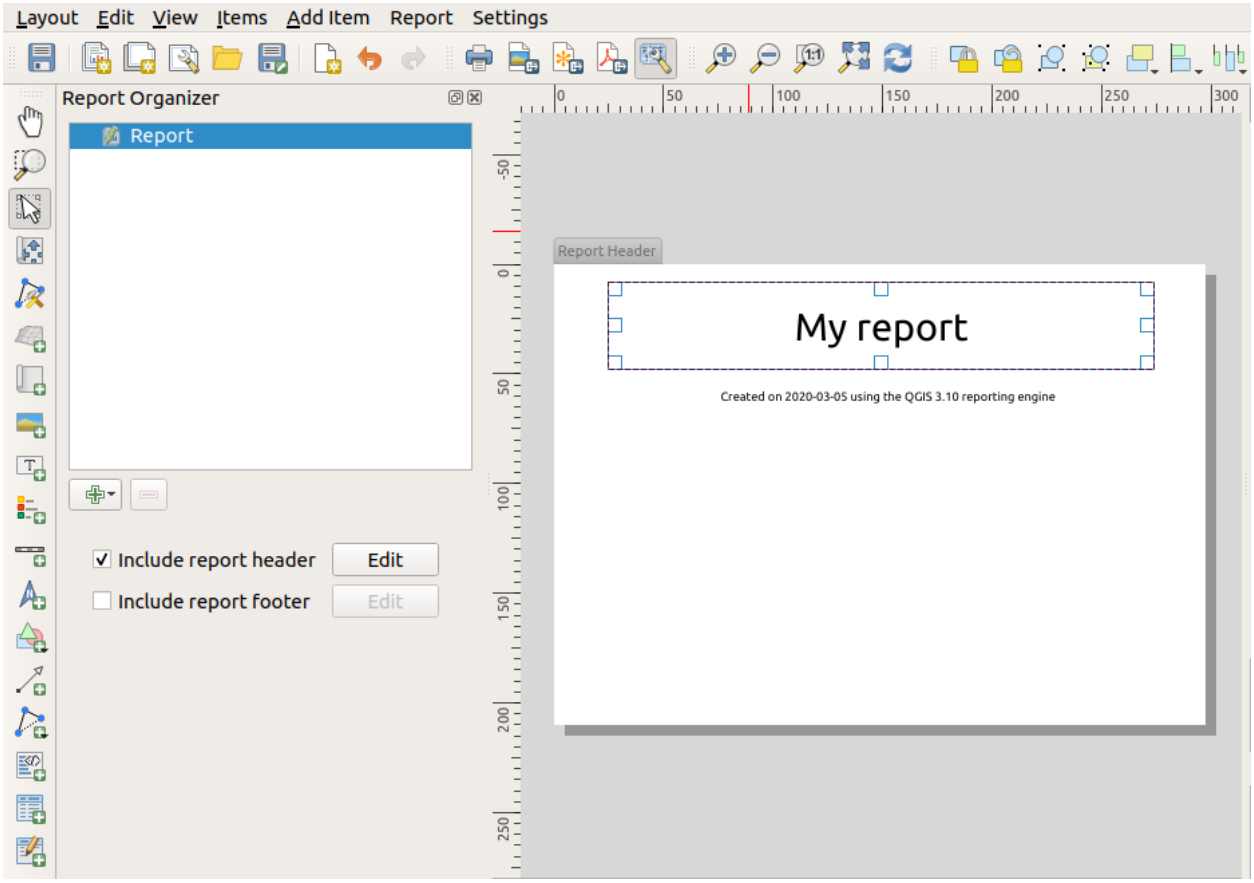
*Static Layout Section* 과 *Field Group Section* 옵션 가운데 하나를 선택할 수 있습니다.

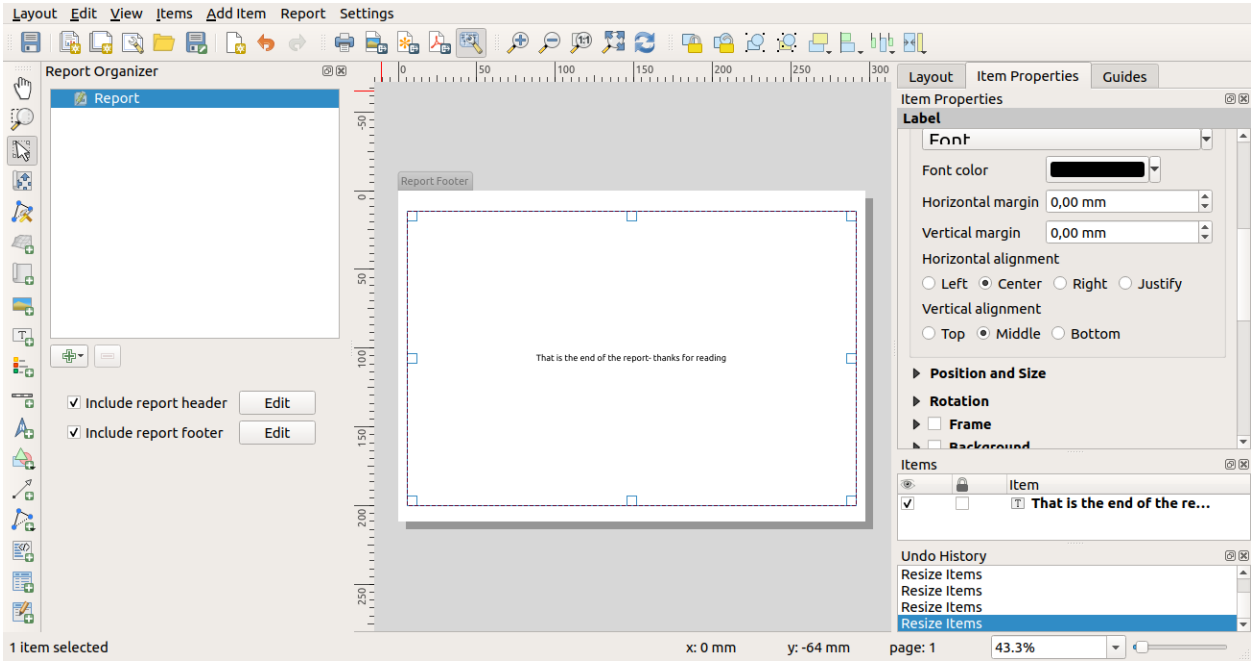
*Add Static Layout Section* 옵션은 정적인 단일 본문 (body) 조판을 추가합니다. 보고서 중간에 정적인 조판을 삽입하는 데 사용할 수 있습니다.

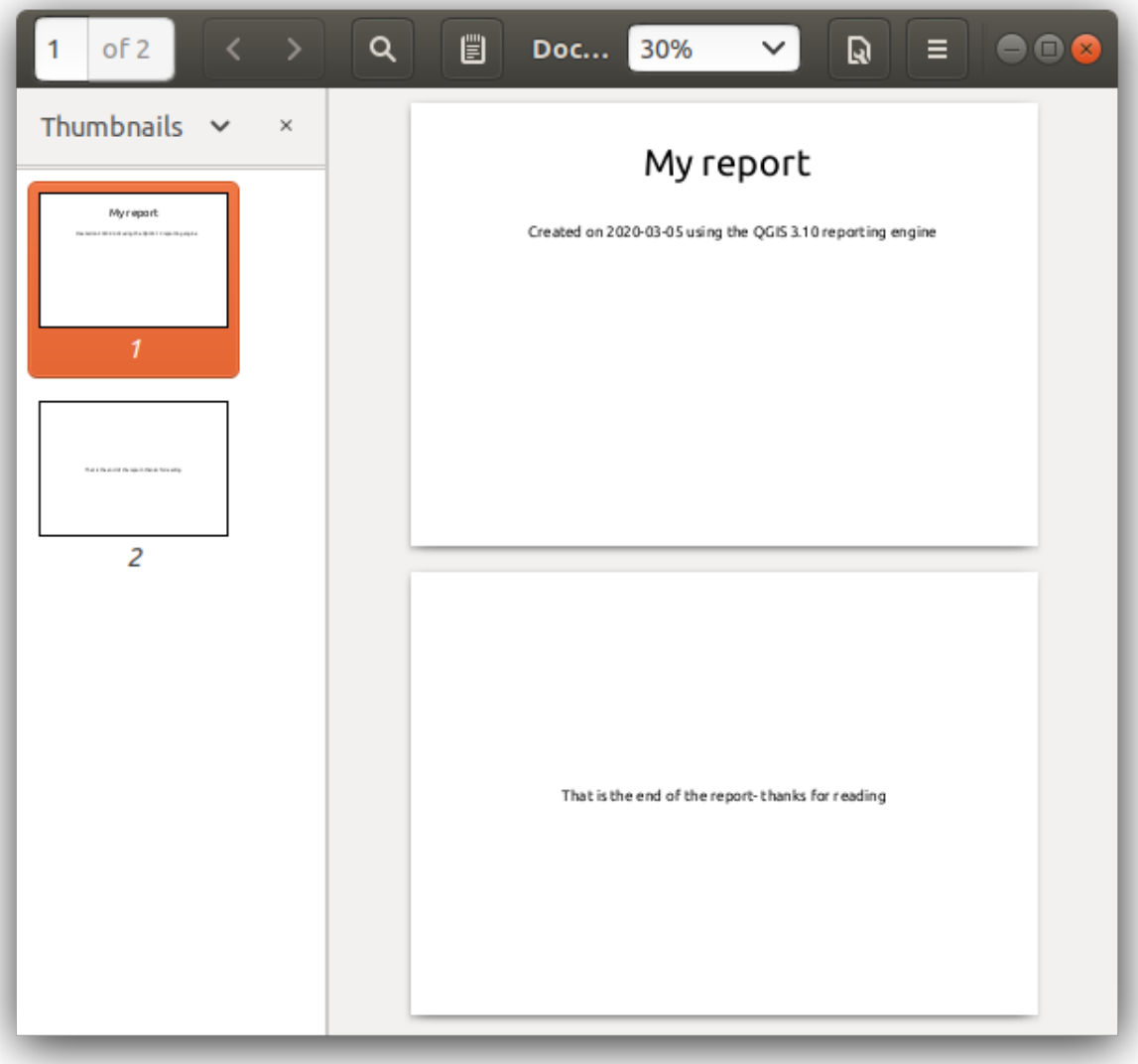
*Field Group Section* 옵션은 레이어의 모든 피처에 대해 필드 그룹의 본문 조판을 반복합니다. 선택한 그룹화 피처로 피처들을 정렬합니다. (오름차순/내림차순 정렬 옵션이 있습니다.) 필드 그룹 부분이 (예를 들어 다른 필드를 가진 또다른 필드 그룹 부분 같은) 자식 부분을 가지고 있는 경우, 그룹 피처에 대해 유일값을 가진 피처들만 반복 작업합니다. 이렇게 하면 내포된 보고서를 만들 수 있습니다.

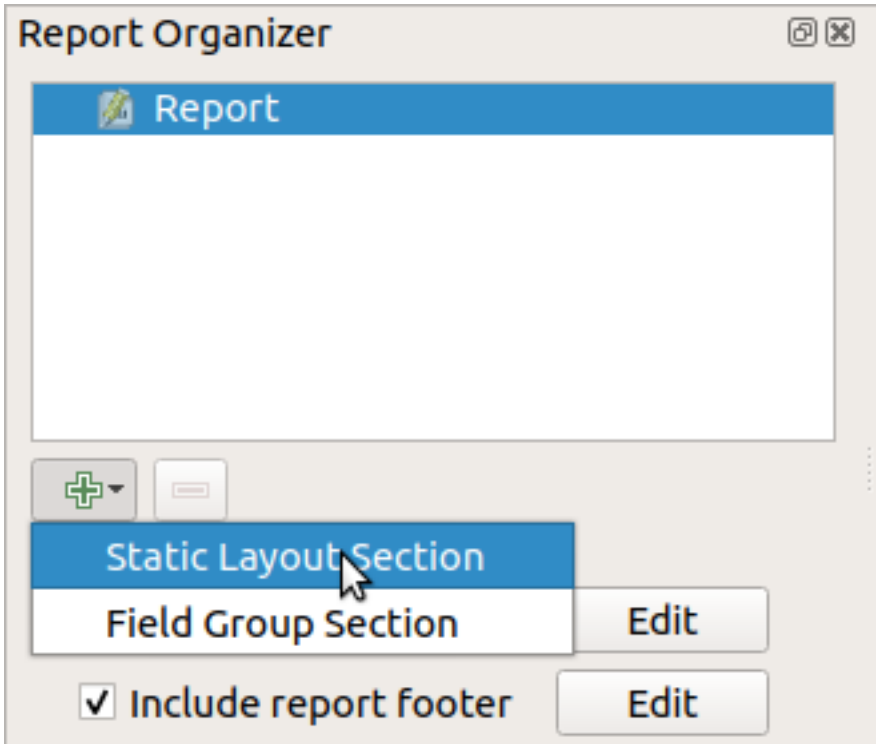
이제 보고서에 *Field Group Section* 을 추가하겠습니다. 가장 기초적인 수준에서, *Field Group Section* 을지도책 생성 과 동등하다고 생각할 수 있습니다. 반복 작업할 레이어를 선택하면, 발견된 피처 각각에 대해 보고서가 필드 그룹 부분을 삽입할 것입니다. 새 *Field Group Section* 을 선택하면 관련된 새로운 설정 여러 개를 볼 수 있습니다:

이 경우 *adm1name* 필드의 값을 사용해서'Admin Level 1'레이어의 모든 주 (☐) 를 반복 작업하도록 필드 그룹을 설정했습니다. 헤더 및 푸터를 포함시키는 기존 옵션은 물론 이 부분에 *body* 를 포함시키는 새 옵션도 있습니다. 이 옵션으로 본문을 추가하고 편집해보겠습니다:


Layout Edit View Items Add Item Report Settings
Report Organizer
Report
Include report header
Edit
Include report footer
Edit
Report Header
My report
Created on 2020-03-05 using the QGIS 3.10 reporting engine


Layout Edit View Items Add Item Report Settings
Report Organizer
Report
Include report header
Edit
Include report footer
Edit
Report Footer
That is the end of the report- thanks for reading
Layout
Item Properties
Guides
Item Properties
Label
Font
Font color
Horizontal margin
0,00 mm
Vertical margin
0,00 mm
Horizontal alignment
Left Center Right Justify
Vertical alignment
Top Middle Bottom
Position and Size
Rotation
Frame
Items
Item
That is the end of the re...
Undo History
Resize Items
Resize Items
Resize Items
Resize Items
1 item selected
x: 0 mm
y: -64 mm
page: 1
43.3%


1
of 2
Doc...
30%
Thumbnails
My report
Created on 2020-03-05 using the QGIS 3.10 reporting engine
1
2
That is the end of the report- thanks for reading




이제 맵과 주의 이름을 표시하는 라벨로 본문을 구성합니다. 주의 이름을 포함시키기 위해, *Add Item* ▸ *Add Label* 메뉴 옵션을 선택한 다음 *Insert or Edit an Expression*…을 통해 *Main Properties* 아래에서 텍스트를 데이터 정의했습니다.

다음 표현식을 사용했습니다 ('name'은 주의 이름을 담고 있는'Admin Level 1'레이어 속성의 이름입니다.):

```
[% "name" %]
```

맵이 현재 보고서 피처를 따르도록 설정합니다 (*Controlled by Report* 옵션을 체크해서 활성화합니다 — 지도책으로 제어 를 활성화하면 지도책이 현재 지도책 피처를 따르는 것과 동일합니다):

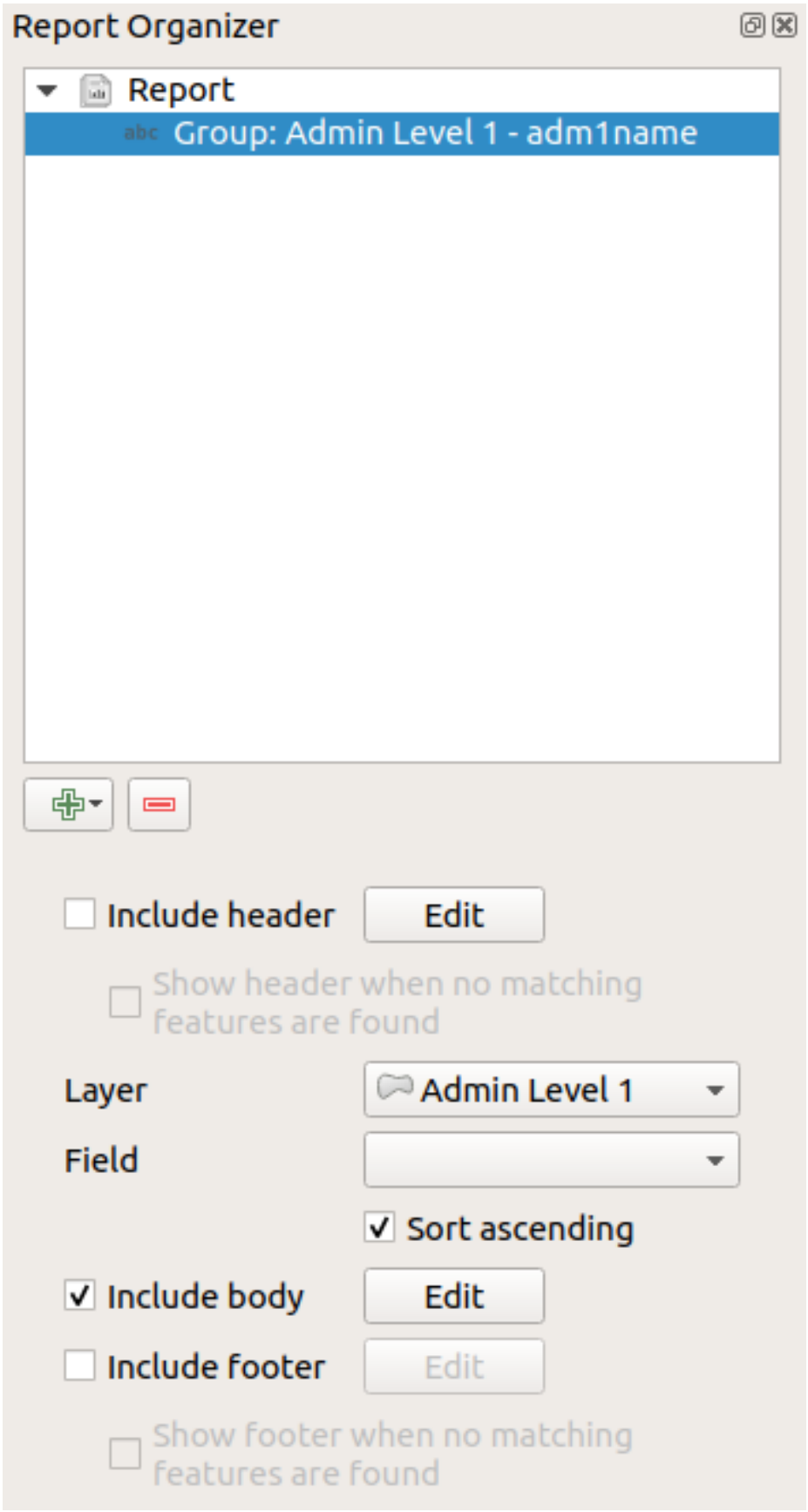
이제 보고서를 내보내면, 다음과 같은 산출물이 나옵니다:

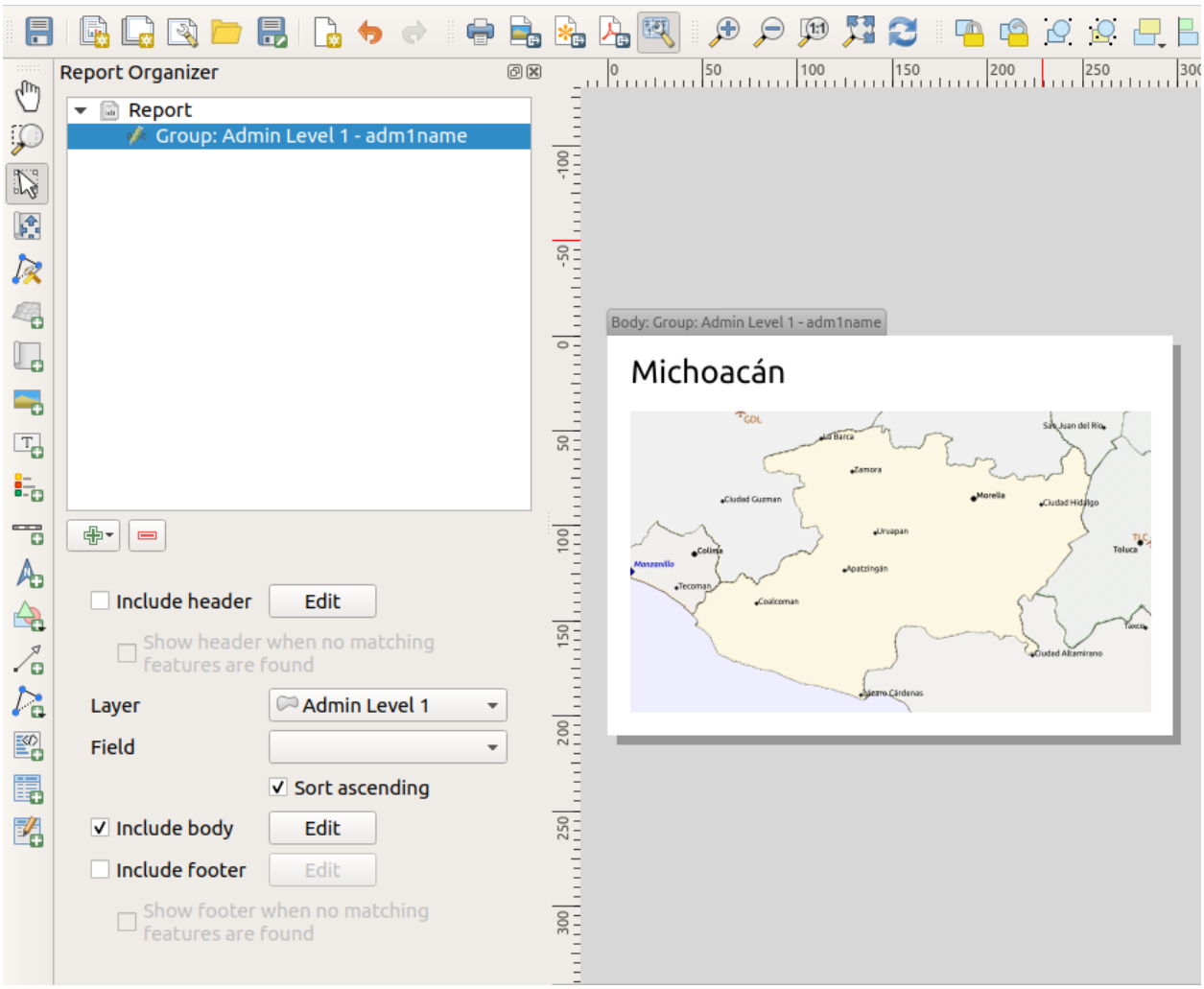
헤더와 푸터 페이지가 있긴 하지만 이제 조금 지도책다워졌군요.

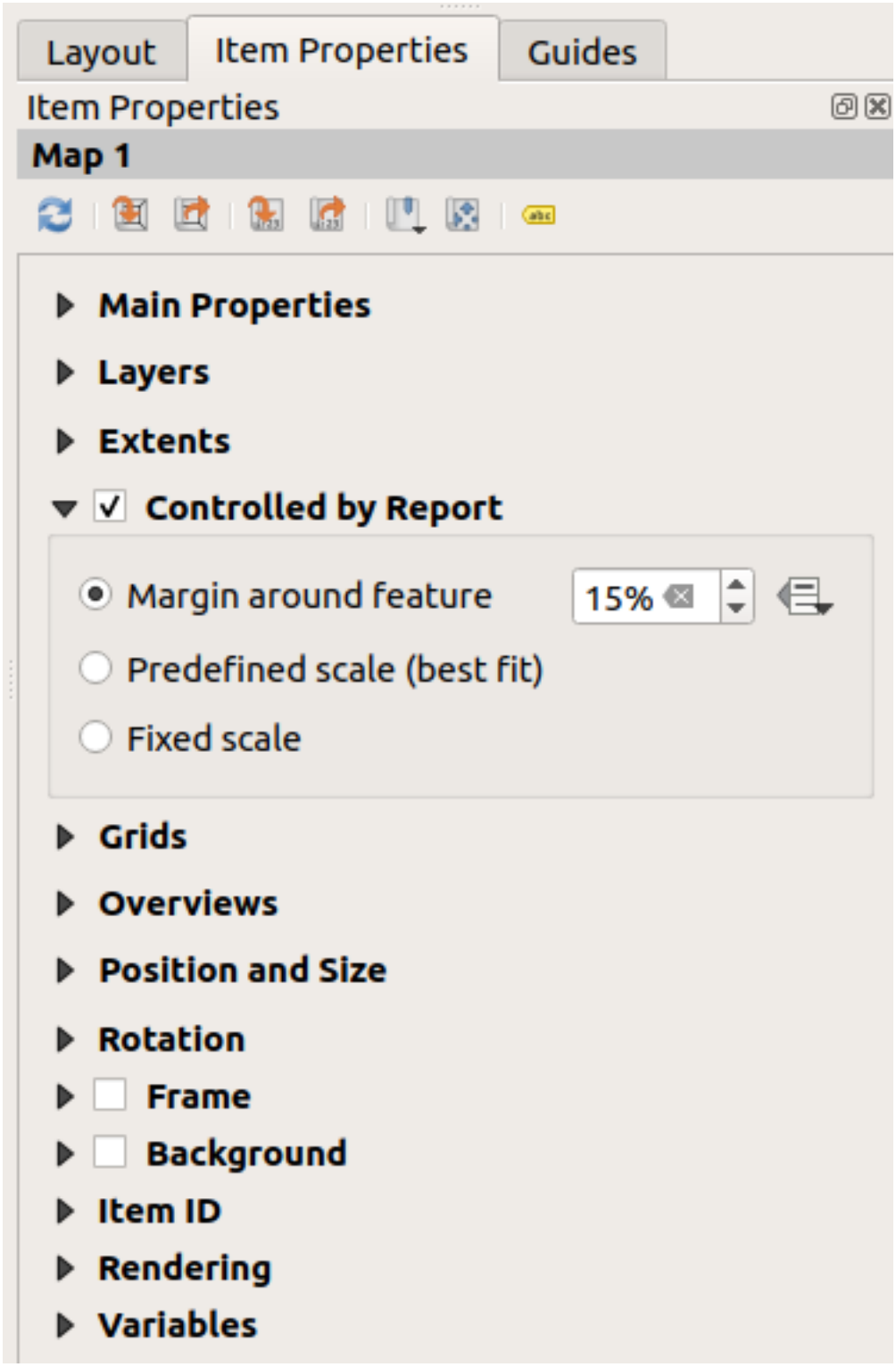
주 (☐) 그룹에 하위 부분을 추가해서 보고서를 좀 더 재미있게 만들어봅시다. 먼저 *Report Organizer* 에서'Admin Level 1'필드 그룹을 선택한 다음, Add Field 버튼을 클릭해서 새 *Field Group Section* 을 추가합니다:

*Field Group Section* 의 피처들을 반복 작업할 때, 상위 (부모) 그룹의 정의 필드 (이 경우 `adm1name`) 와 일치하도록 피처들을 필터링할 것입니다. 이때, 추가한 하위 부분이 *Populated Places* 레이어를 반복 작업하면서 일치하는 각 지역에 대한 본문 부분을 추가할 것입니다. 주목할 점은 *Populated Places* 레이어가 *adm1name* 상위 레이어의 정의 필드와 동일한 이름을 가진 속성을 보유하고 있다는 것입니다. 즉 각 지역을 해당 지역이 있는 주에 태그한다는 뜻입니다. (운이 좋다면 사용자 데이터가 이미 이런 구조로 되어 있을 겁니다. 그렇지 않다면, 위치로 속성 결합 공간 처리 알고리즘을 실행해서 사용자 고유의 필드를 생성하세요.) 이 보고서를 내보내면, QGIS 는 *Admin Level 1* 레이어의 첫 번째 주를 가져온 다음 모든 *Populated Places* 레이어를 일치하는 *adm1name* 값으로 반복 작업할 것입니다. 다음은 그 결과물입니다:

여기까지, 인구 밀집 지역 그룹에 대한 기초적인 본문을 지역의 맵과 일부 지역의 속성 테이블을 포함해서 생성했습니다. 이제 보고서는 보고서 헤더, 첫 번째 주에 대한 페이지, 해당 주에 있는 인구 밀집 지역 전체에 대한 페이지, 그 다음 주에 대한 페이지와 해당 주의 인구 밀집 지역 페이지, 그리고 마지막으로 보고서 푸터로 구성됐습니다. 인구 밀집 지역 그룹에 헤더를 추가하고자 했다면, 아래에 있는 그림과 같이 각 주의 인구 밀집 지역 페이지들 앞에 삽입되었을 것입니다.


Report Organizer
Report
Group: Admin Level 1 - adm1name
Include header
Edit
Show header when no matching features are found
Layer
Admin Level 1
Field
Sort ascending
Include body
Edit
Include footer
Edit
Show footer when no matching features are found


Report Organizer
Report
Group: Admin Level 1 - adm1name
Include header
Edit
Show header when no matching features are found
Layer
Admin Level 1
Field
Sort ascending
Include body
Edit
Include footer
Edit
Show footer when no matching features are found
Body: Group: Admin Level 1 - adm1name
Michoacán


Layout
Item Properties
Guides
Item Properties
Map 1
Main Properties
Layers
Extents
Controlled by Report
Margin around feature
15%
Predefined scale (best fit)
Fixed scale
Grids
Overviews
Position and Size
Rotation
Frame
Background
Item ID
Rendering
Variables

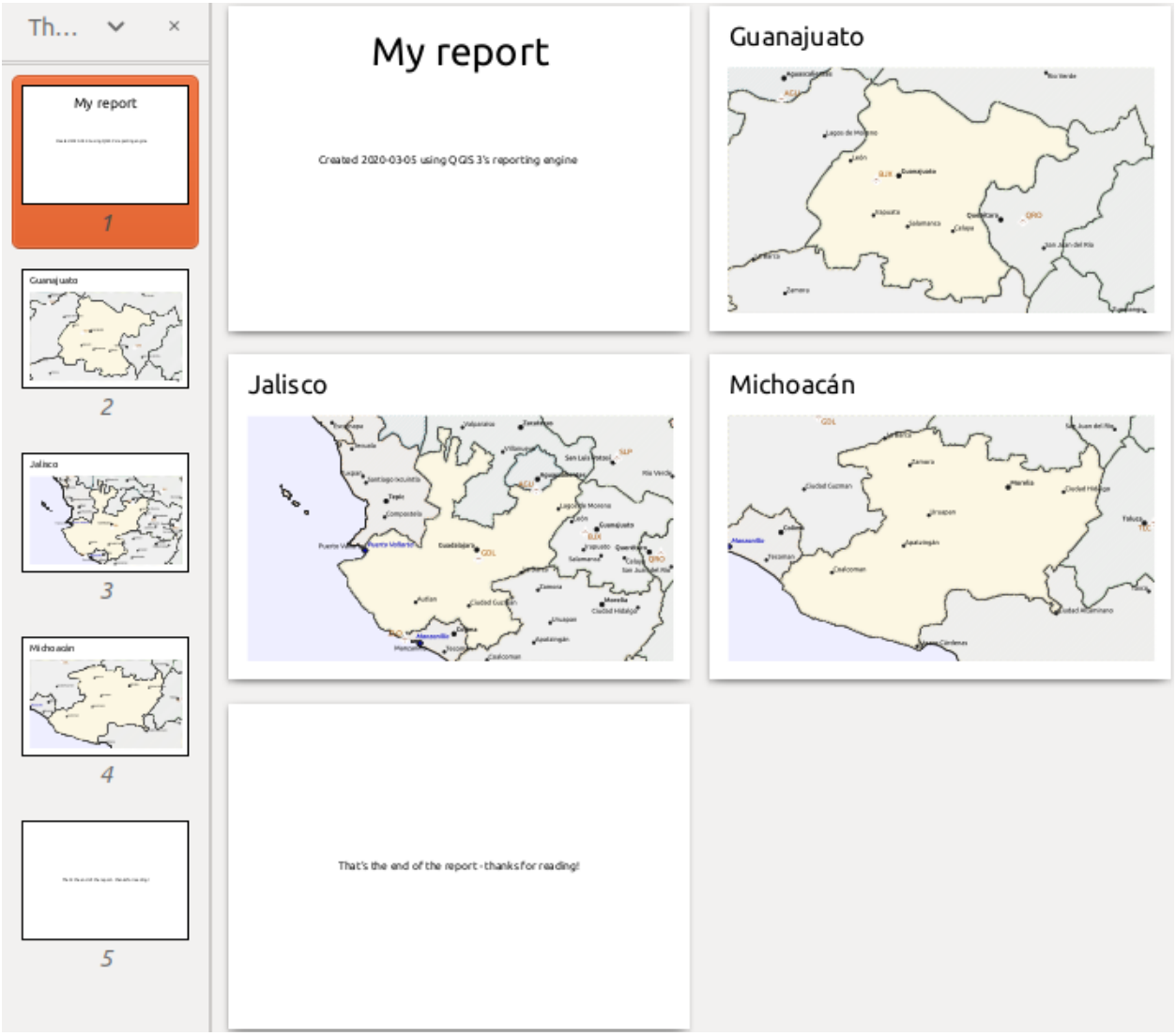
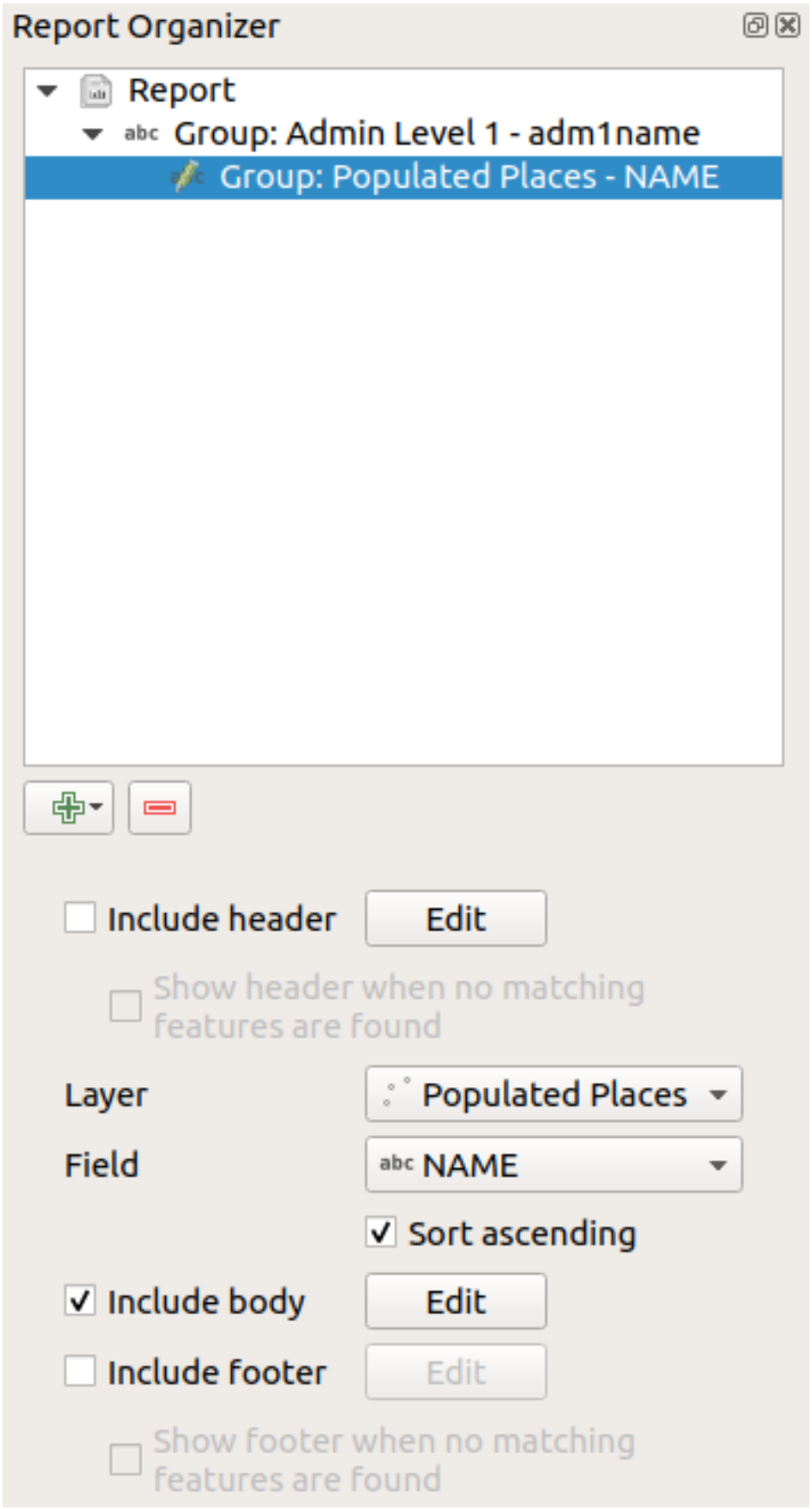
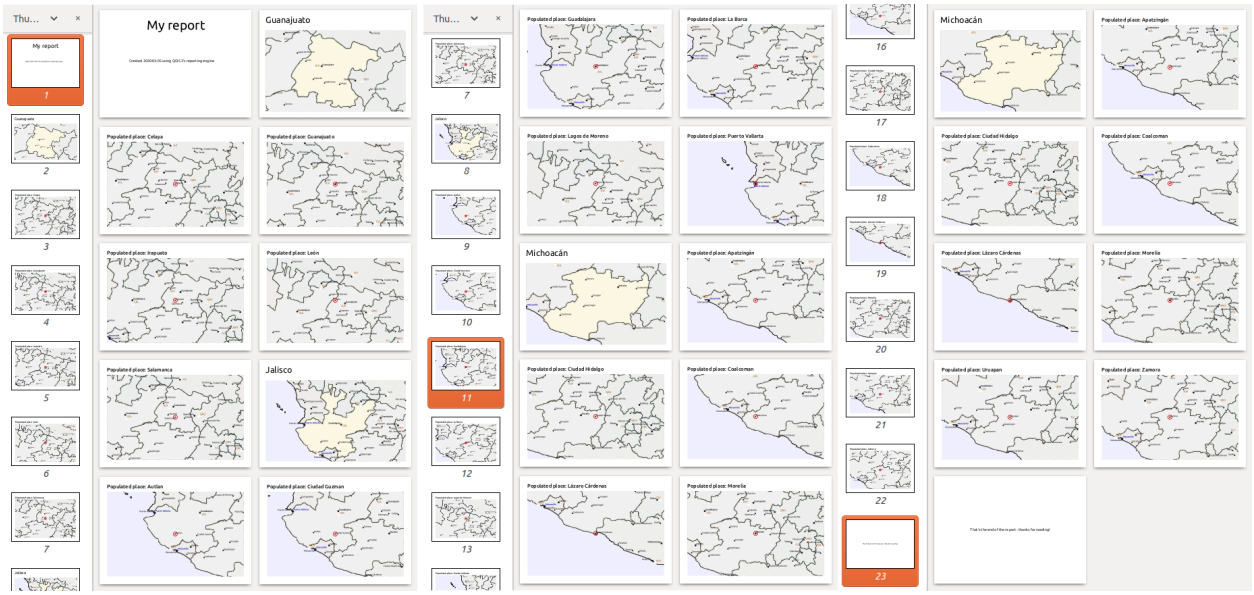



그림 18.70: 보고서 헤더, 각 주에 대한 페이지와 보고서 푸터


Report Organizer
Report
Group: Admin Level 1 - adm1name
Group: Populated Places - NAME
Include header
Edit
Show header when no matching features are found
Layer
Populated Places
Field
NAME
Sort ascending
Include body
Edit
Include footer
Edit
Show footer when no matching features are found




마찬가지로, 인구 밀집 지역 그룹에 푸터를 추가했다면 각 주의 마지막 지역 뒤에 삽입되었을 것입니다.

내포된 하위 부분 이외에도, 보고서에 하위 부분을 연속해서 포함시킬 수 있습니다. 'Admin Level 1'그룹에 *Airports* 를 위한 두 번째 하위 부분을 추가하는 경우, (*Airports* 레이어가 상위 그룹에 링크할 수 있는 *adm1name* 속성을 가지고 있는 경우) 보고서는 먼저 각 주별로 모든 인구 밀집 지역을 목록화한 다음, 다음 주로 넘어가기 전에 해당 주에 있는 모든 공항도 목록화할 것입니다.

여기서 중요한 점은 *Airports group* 이 *Populated Places group* 의 하위 부분이 아니라 *Admin Level 1 group* 의 하위 부분이라는 사실입니다.

이 경우 보고서는 다음과 같이 구성될 것입니다 (주기 (☐☐) 도 추가됐다는 사실을 기억하십시오. 이런 방식으로 피처 특화 그림을 추가하는 방법을 이 다음에 설명합니다):

### 보고서에 그림 포함시키기

그림은 보고서에 매우 유용할 수 있습니다. QGIS 는 그림을 보고서의 정적인 일부와 동적인 일부, 양쪽으로 사용할 수 있습니다. 표준적인 인쇄 조판의 경우와 동일한 방법으로 그림을 추가합니다. 정적인 보고서의 일부인 경우 (그리고 동적인 부분의 정적인 그림인 경우) 더 설명할 것은 없습니다.

그러나 보고서 피처에 맞게 다듬어진 그림을 원한다면, 사용자 레이어가 포함시킬 그림을 정의하는 데 사용할 수 있는 속성을 가지고 있어야만 합니다.

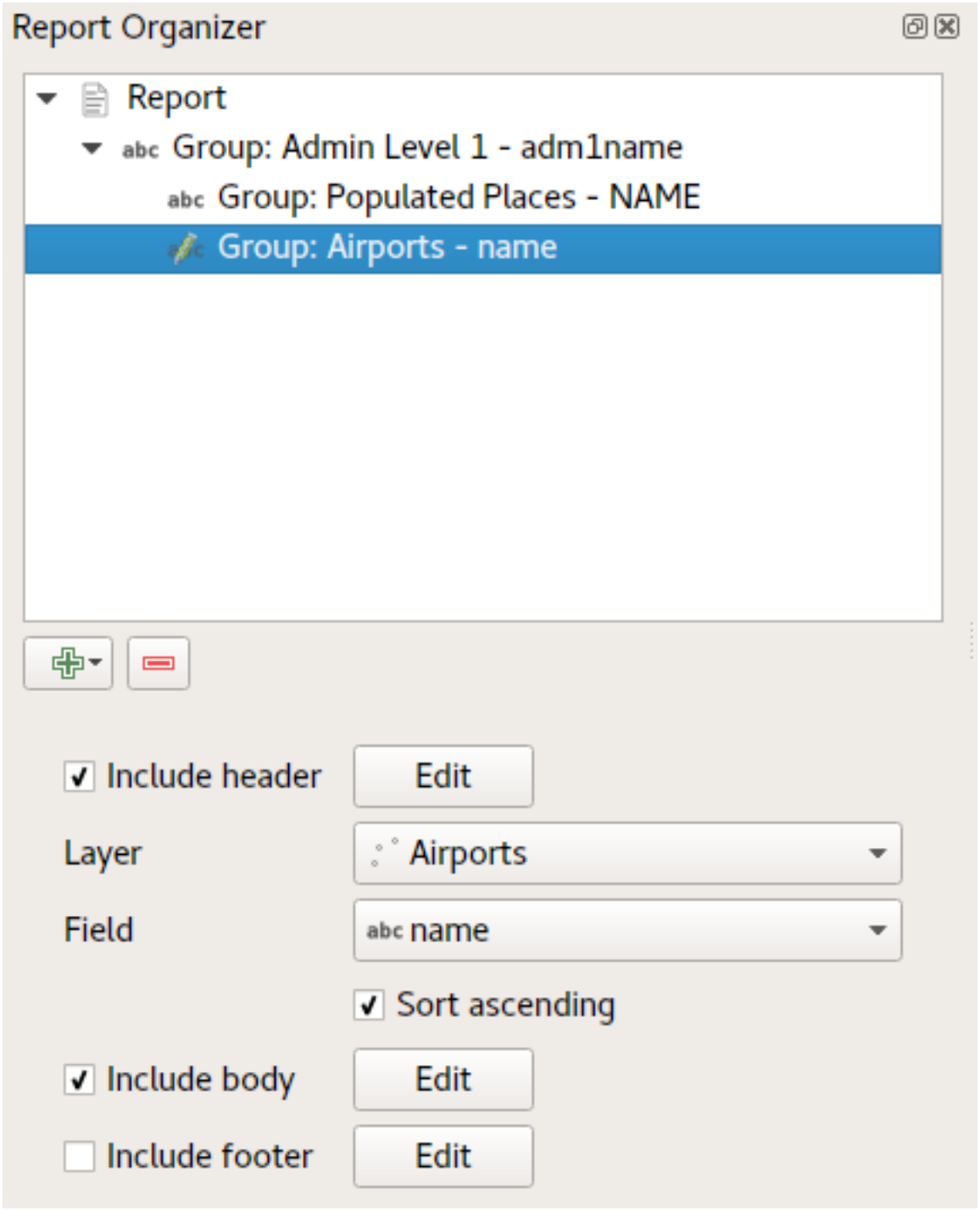
QGIS 는 보고서 이미지에 대해 절대 파일명에 의존합니다.

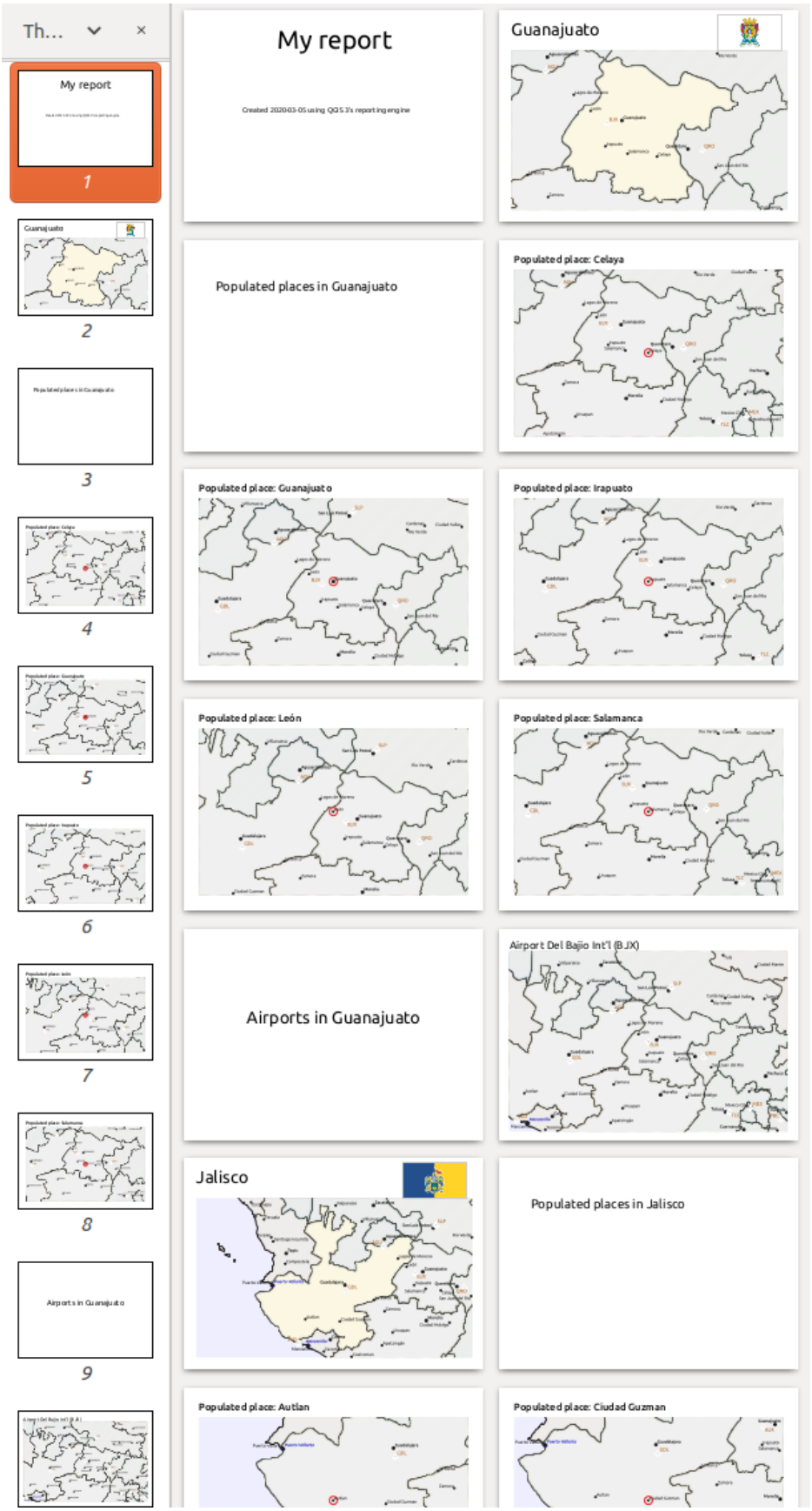
동적인 그림의 경우, 먼저 그룹의 본문 부분에 그림을 동일한 방법으로 추가하십시오. 그림 *Item properties* 패널에서 Data defined override 버튼으로 *Image Source* 를 설정한 다음 이미지의 절대 경로를 담고 있는 속성을 선택하거나, 또는 *Edit*⋯버튼을 클릭한 다음 이미지의 절대 경로를 생성하는 표현식을 입력하십시오.

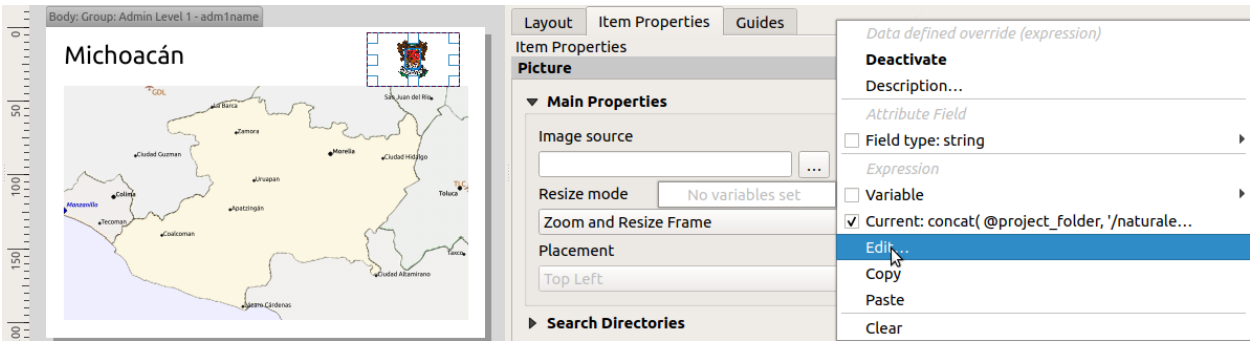
다음 예시 표현식은 프로젝트 파일이 위치한 디렉터리와 (`@project_path`) 파일명 생성 기반이 되는 속성을 (`adm1name`) 사용해서 그림을 가리키는 절대 경로를 지정하기 위해 문자열을 연결합니다. (이 경우 `adm1name` 속성의 문자열을 대문자로 변환한 다음 그 뒤에 `'_flag.png'` 를 붙여서 파일명을 생성합니다.)

```
concat(@project_folder, '/naturalearth/pictures/' ,
       upper("adm1name"), '_flag.png')
```

즉 그림이 프로젝트 파일 디렉터리의 `naturalearth/pictures` 하위 디렉터리에 위치한다는 뜻입니다.


Report Organizer
Report
Group: Admin Level 1 - adm1name
Group: Populated Places - NAME
Group: Airports - name
Include header
Edit
Layer
Airports
Field
name
Sort ascending
Include body
Edit
Include footer
Edit


Th...
My report
1
2
3
4
5
6
7
8
9
My report
Created 2020-03-05 using QGIS 3's reporting engine
Guanajuato
Populated places in Guanajuato
Populated place: Celaya
Populated place: Guanajuato
Populated place: Irapuato
Populated place: León
Populated place: Salamanca
Airports in Guanajuato
Airport Del Bajio Int'l (BJX)
Jalisco
Populated places in Jalisco
Populated place: Autlan
Populated place: Ciudad Guzman




### 맵에서 현재 보고서 피처를 강조하기

앞의 보고서에서, 맵에 있는 보고서 피처를 강조하는 데 (주의 경우) 강조 (highlight) 를, (인구 밀집 지역의 경우) 원을 사용했습니다. 맵에 있는 보고서 피처를 (맵 중앙에 배치하는 것을 제외하고) 강조하려면, 피처의 `@id` 와 지도책의 `@atlas_featureid` 를 비교해서 스타일을 데이터 정의해야만 합니다.

예를 들어, 보고서 피처에 다른 피처보다 더 두꺼운 라인/경계선을 사용하고자 하는 경우 라인 너비를 데이터 정의할 수 있습니다:

```
if($id=@atlas_featureid, 2.0, 0.1)
```

보고서 피처의 폴리곤 외곽선의 너비는 2 단위가 되는 반면, 다른 모든 피처의 외곽선 너비는 0.1 단위가 될 것입니다. 색상도 데이터 정의할 수 있습니다. 다음 예시는 보고서 피처에는 불투명한 진한 마젠타 색상을, 다른 피처에는 반투명한 연한 회색을 지정하는 표현식입니다:

```
if($id=@atlas_featureid, '#FF880088', '#88CCCCCC')
```

### 더 많은 1 레벨 그룹

부분 헤더 및 푸터와 함께 내포 부분과 연속 부분을 결합하면, 엄청난 유연성을 얻을 수 있습니다. 예를 들어, 다음 보고서에서 *Ports* 에 대한 또다른 필드 그룹을 주 보고서의 자식으로 추가하십시오. 이제 인구 밀집 지역과 공항과 함께 주들을 나열한 다음, 해당 지역에 있는 항구를 요약해서 나열할 것입니다:

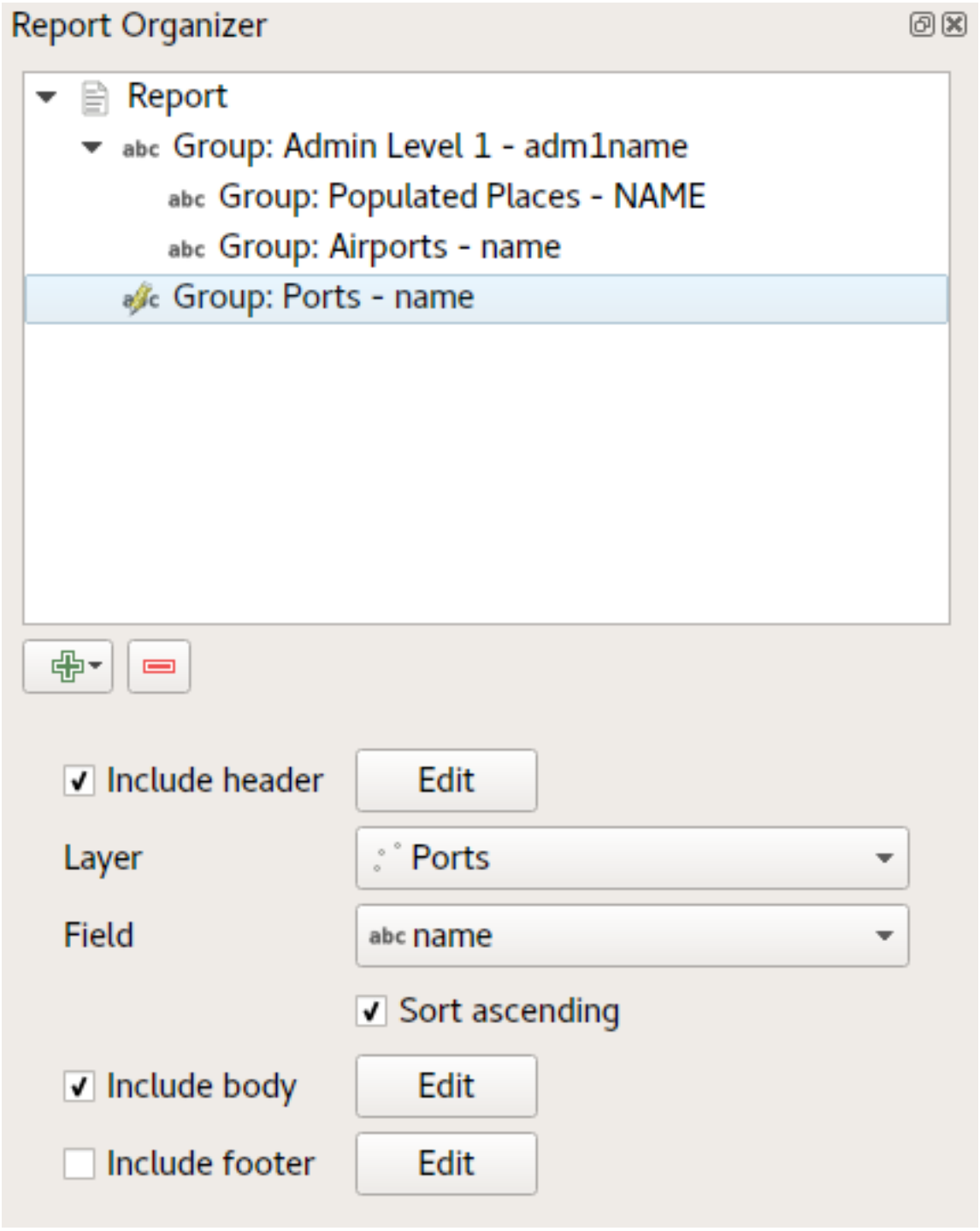
이렇게 내보내면 보고서의 마지막 부분에 다음과 같이 산출됩니다:

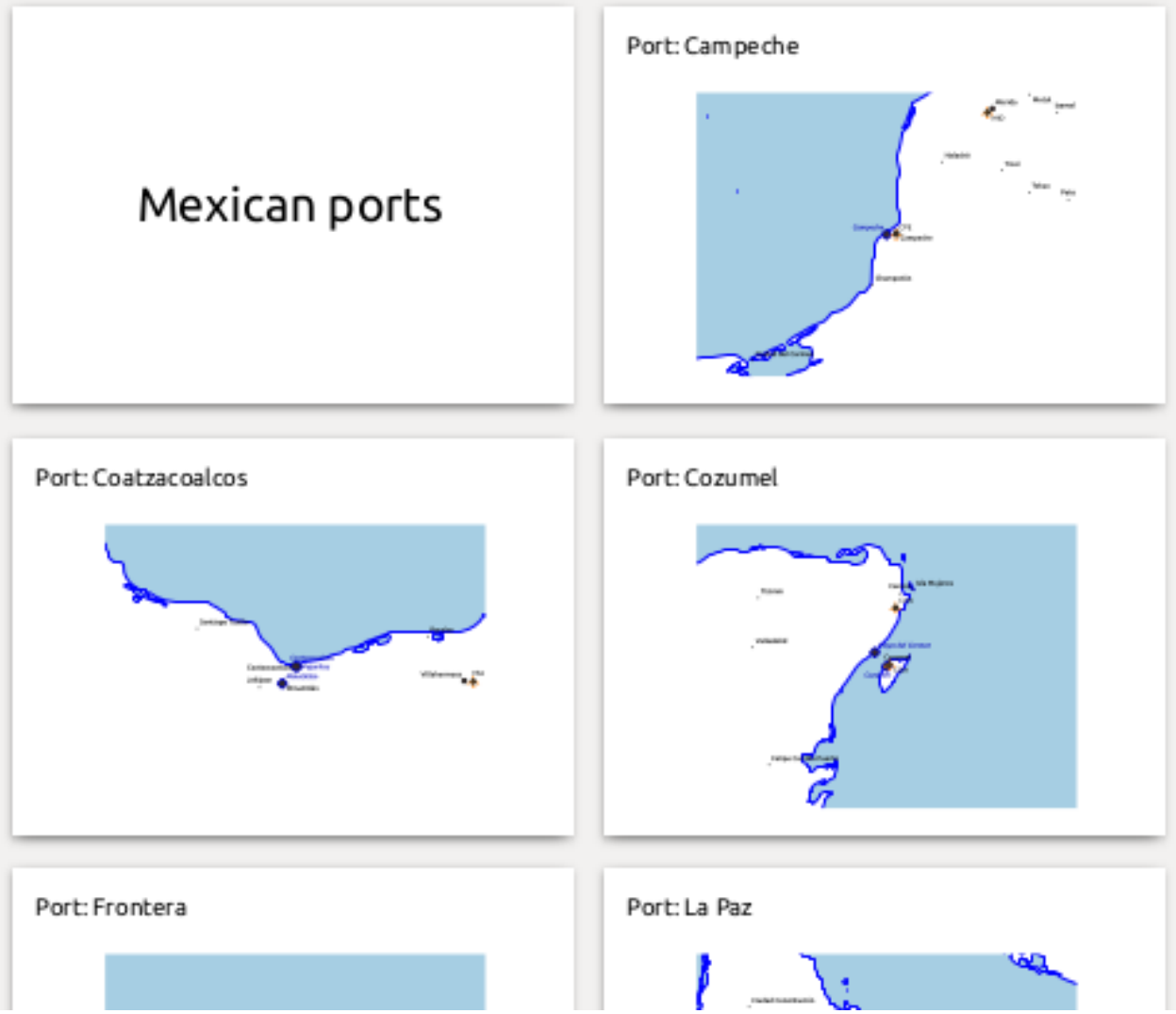
## 18.4.4 내보내기 설정

보고서를 내보낼 때 (*Report* → *Export Report as Images*⋯/*SVG*⋯/*PDF*⋯) 파일명을 입력하라는 요청을 받을 것입니다. 그 다음 내보내기 설정을 가장 적절한 산출물을 얻기 위해 미세 조정할 수 있습니다.

이제 알겠지만, QGIS 의 보고서는 대단히 강력하고 유연하답니다!

> **참고**
>
> 이 정보들은 GIS 소프트웨어 회사인 노스로드 (North Road) 의 블로그 게시글, Exploring Reports in QGIS 3.0 - the Ultimate Guide! 를 손본 것입니다.


Report Organizer
Report
Group: Admin Level 1 - adm1name
Group: Populated Places - NAME
Group: Airports - name
Group: Ports - name
Include header
Edit
Layer
Ports
Field
name
Sort ascending
Include body
Edit
Include footer
Edit


Mexican ports
Port: Campeche
Port: Coatzacoalcos
Port: Cozumel
Port: Frontera
Port: La Paz

CHAPTER 19

# OGC / ISO 프로토콜 작업

OGC(Open Geospatial Consortium) 는 전 세계 300 여 개 이상의 기업, 정부 기관, 비영리 단체 및 연구소가 회원으로 참여하고 있는 국제 기구입니다. OGC 회원들은 지리공간 콘텐츠 및 서비스, GIS 데이터 공간 처리 및 교환을 위한 표준을 개발하고 구현하고 있습니다.

지리적 객체에 대한 기본 데이터 모델을 형성하면서, OGC 는 상호 정보 교환이 가능한 위치 및 GIS 를 비롯한 지리공간 기술에 대한 특정한 필요를 만족시키기 위해 수많은 사양을 개발해왔고, 개발하고 있습니다. https://www.ogc.org/ 사이트에서 더 자세한 정보를 찾아볼 수 있습니다.

QGIS 가 지원하는 중요한 OGC 사양은 다음과 같습니다:

- **WMS** –웹 맵 서비스 (Web Map Service) (*WMS/WMTS 클라이언트*)
- **WMTS** –웹 맵 타일 서비스 (Web Map Tile Service) (*WMS/WMTS 클라이언트*)
- **WFS** –웹 피처 서비스 (Web Feature Service) (*WFS 및 WFS-T 클라이언트*)
- **WFS-T** –웹 피처 서비스 - 트랜잭션 (Web Feature Service - Transactional) (*WFS 및 WFS-T 클라이언트*)
- **WCS** –웹 커버리지 서비스 (Web Coverage Service) (*WCS 클라이언트*)
- **WPS** –웹 공간 처리 서비스 (Web Processing Service)
- **CSW** –웹 용 카탈로그 서비스 (Catalog Service for the Web)
- **SFS** –SQL 용 단순 피처 (Simple Features for SQL) (*PostgreSQL Database*)
- **GML** –지리 마크업 언어 (Geography Markup Language)
- **센서싱스 API** —센서싱스 (SensorThings) API (*SensorThings*)

OGC services are increasingly being used to exchange geospatial data between different GIS implementations and data stores. QGIS can deal with the above specifications as a client, being **SFS** (through support of the PostgreSQL data provider, see section *PostgreSQL Database*).

QGIS 서버, UMN MapServer 또는 GeoServer 를 설치하고 웹 서버를 사용해서 WMS, WMTS, WFS, WFS-T 및 WCS 프로토콜을 통해 사용자의 맵과 데이터를 공유할 수도 있습니다.

## 19.1 WMS/WMTS 클라이언트

### 19.1.1 WMS 지원 개요

QGIS 는 현재 WMS 1.1, 1.1.1 및 1.3 을 알아듣는 WMS 클라이언트 역할을 할 수 있습니다. 특히, DEMIS 처럼 공개된 서버를 대상으로 검증했습니다.

WMS 서버는 클라이언트 (예: QGIS) 가 범위, 레이어 집합, 심볼화 스타일, 그리고 투명도를 지정해서 래스터 맵을 요청하면 그에 따라 동작합니다. WMS 서버가 요청을 받으면 서버의 로컬 데이터소스를 찾아보고, 맵을 래스터화한 다음, 클라이언트에게 래스터 포맷으로 전송합니다. QGIS 의 경우, 이 포맷은 일반적으로 JPEG 또는 PNG 파일입니다.

일반적으로 WMS 는 완전한 웹 서비스라기보다는 REST(Representational State Transfer) 서비스에 더 가깝습니다. 따라서, QGIS 가 생성한 URL 을 가져다 웹 브라우저에 입력하면 QGIS 가 내부적으로 이용하는 이미지와 동일한 이미지를 받을 수 있습니다. 이 방법은 문제 해결에 유용합니다. 지금 시중에 여러 브랜드의 WMS 서버가 존재하는데, 모두 자신만의 방식으로 WMS 표준을 해석하고 있기 때문입니다.

사용자가 WMS 서버에 접근할 수 있는 URL 을 알고 있고, 해당 서버로 쓸 만한 인터넷 연결이 가능하며, 서버가 HTTP 를 데이터 전송 프로토콜이라고 이해하고 있는 경우, WMS 레이어를 매우 쉽게 추가할 수 있습니다.

또한, QGIS 는 GetCapabilities 요청이 전송되지 않는 한 사용자의 WMS 응답 (예: 이미지) 을 24 시간 동안 캐시에 담아둘 것입니다. GetCapabilities 요청은 WMS 서버 기능을 받아오기 위해 *WMS/WMTS* 대화창에 있는 *Connect* 버튼을 누를 때마다 전송됩니다. 프로젝트를 불러오는 시간을 최적화하기 위한 자동 기능입니다. 프로젝트를 WMS 레이어와 함께 저장하는 경우, 24 시간이 넘지 않은 한, 다음 번에 이 프로젝트를 열 때 캐시에서 대응하는 WMS 타일을 불러올 것입니다.

### 19.1.2 WMTS 지원 개요

QGIS 는 WMTS 클라이언트 역할도 할 수 있습니다. WMTS 란 지리공간 데이터의 타일셋을 배포하기 위한 OGC 표준입니다. WMTS 는 WMS 보다 더 빠르고 효율적으로 데이터를 배포할 수 있습니다. 왜냐하면 WMTS 가 사전에 타일셋을 생성해놓기 때문에, 클라이언트가 타일셋의 전송만을 요청하지 생성을 요청하는 것이 아니기 때문입니다. 이에 반해 WMS 요청은 일반적으로 데이터의 생성 및 전송 둘 다 요구합니다. OGC 표준이 아닌 서비스 가운데 타일로 구성된 지리공간 데이터를 살펴볼 수 있는 잘 알려진 예시라면 역시 구글맵이겠죠.

사용자가 원할 수도 있는 수준에 가깝도록 데이터를 다양한 축척으로 표시하기 위해, WMTS 타일셋은 서로 다른 여러 축척 수준으로 생성되며, GIS 클라이언트가 요청하는 경우 사용할 수 있도록 준비됩니다.

다음 도표는 타일셋의 개념을 설명하고 있습니다:

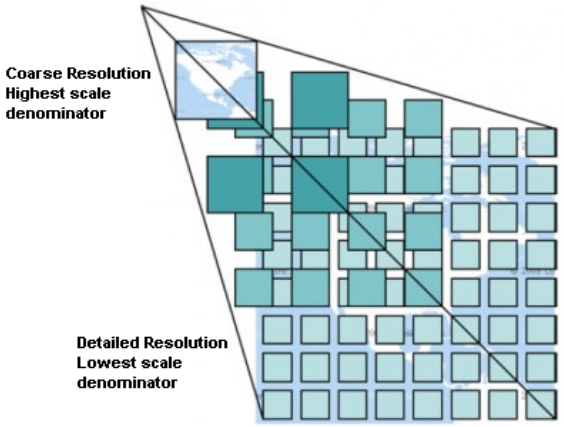



그림 19.1: WMTS 타일셋의 개념

QGIS 는 키-값 쌍 (Key-Value-Pairs; KVP) 및 RESTful 을 통해 두 가지 유형의 WMTS 인터페이스를 지원합니다. 이 두 인터페이스는 서로 다르기 때문에, QGIS 에서도 서로 다르게 설정해줘야 합니다.

1. **WMTS KVP** 서비스에 접근하기 위해, QGIS 사용자는 WMS/WMTS 인터페이스를 열고 WMTS 타일 서비스의 URL 에 다음과 같은 문자열을 추가해야만 합니다:

   ```
   "?SERVICE=WMTS&REQUEST=GetCapabilities"
   ```

   다음은 이런 주소 유형의 예시입니다:

   ```
   https://opencache.statkart.no/gatekeeper/gk/gk.open_wmts?service=WMTS&
   ↪request=GetCapabilities
   ```

   이 WMTS 에 있는 `topo2` 레이어를 검증하는 작업은 잘 진행됩니다. 이 문자열을 추가한다는 것은 WMS 웹 서비스 대신 WMTS 웹 서비스를 이용한다는 뜻입니다.

2. **RESTful WMTS** 서비스는 다른 형태의, 복잡하지 않은 URL 을 이용합니다. OGC 가 추천하는 서식은 다음과 같습니다:

   ```
   {WMTSBaseURL}/1.0.0/WMTSCapabilities.xml
   ```

   이 서식으로 해당 URL 이 RESTful 주소라는 사실을 알 수 있습니다. QGIS 의 WMS 설정 양식에 있는 URL 란에 RESTful WMTS 주소를 추가하는 것만으로도 RESTful WMTS 웹서비스에 접근할 수 있습니다. 오스트리아의 기본도 (basemap) 의 경우, 이런 주소 유형의 예시는 다음과 같습니다:

   ```
   https://maps.wien.gv.at/basemap/1.0.0/WMTSCapabilities.xml
   ```

> **참고**
>
> 지금도 WMS-C 라는 구식 서비스를 몇 개 찾을 수 있습니다. 이 서비스는 WMTS 와 매우 비슷합니다. (목적은 동일하지만 작동 방식이 약간 다릅니다.) WMTS 서비스를 관리하는 것과 동일한 방식으로 이 서비스를 관리할 수 있습니다. URL 끝에 `?tiled=true` 문자열만 추가하면 됩니다. 이 서비스의 사양에 대해 더 알고 싶다면 타일 맵 서비스 사양 을 참조하세요.
>
> 사용자 입장에서는 WMTS 와 WMS-C 를 종종 헷갈릴 수도 있습니다.

### 19.1.3 WMS/WMTS 서버 선택하기

처음으로 QGIS 에서 WMS/WMTS 기능을 사용하는 경우, 정의돼 있는 서버가 없을 겁니다.

그러면 사용자가 원하는 서버에 대해 연결을 생성해야 합니다:

1. 다음 가운데 한 방법으로 *Data Source Manager* 대화창의 *WMS/WMTS* 탭으로 가십시오:
   - Open Data Source Manager 버튼 (또는 `Ctrl+L` 키) 을 누른 다음 탭을 활성화합니다.
   - 레이어 관리 툴바에 있는 Add WMS layer 버튼을 클릭합니다.
   - 또는 *Layer* → *Add Layer* → *Add WMS/WMTS Layer…*메뉴를 선택합니다.
2. *Layers* 탭에서 *New…*버튼을 누르십시오. *Create a New WMS/WMTS Connection…*대화창이 열립니다.

> **팁**
>
> 탐색기 패널 에서 *WMS/WMTS* 항목을 오른쪽 클릭한 다음 *New Connection…*을 선택해도 *Create a New WMS/WMTS Connection…*대화창이 열립니다.

3. 이제 사용자가 원하는 WMS 서버에 연결하기 위한 파라미터를 다음에 나열한 순서대로 입력하십시오:

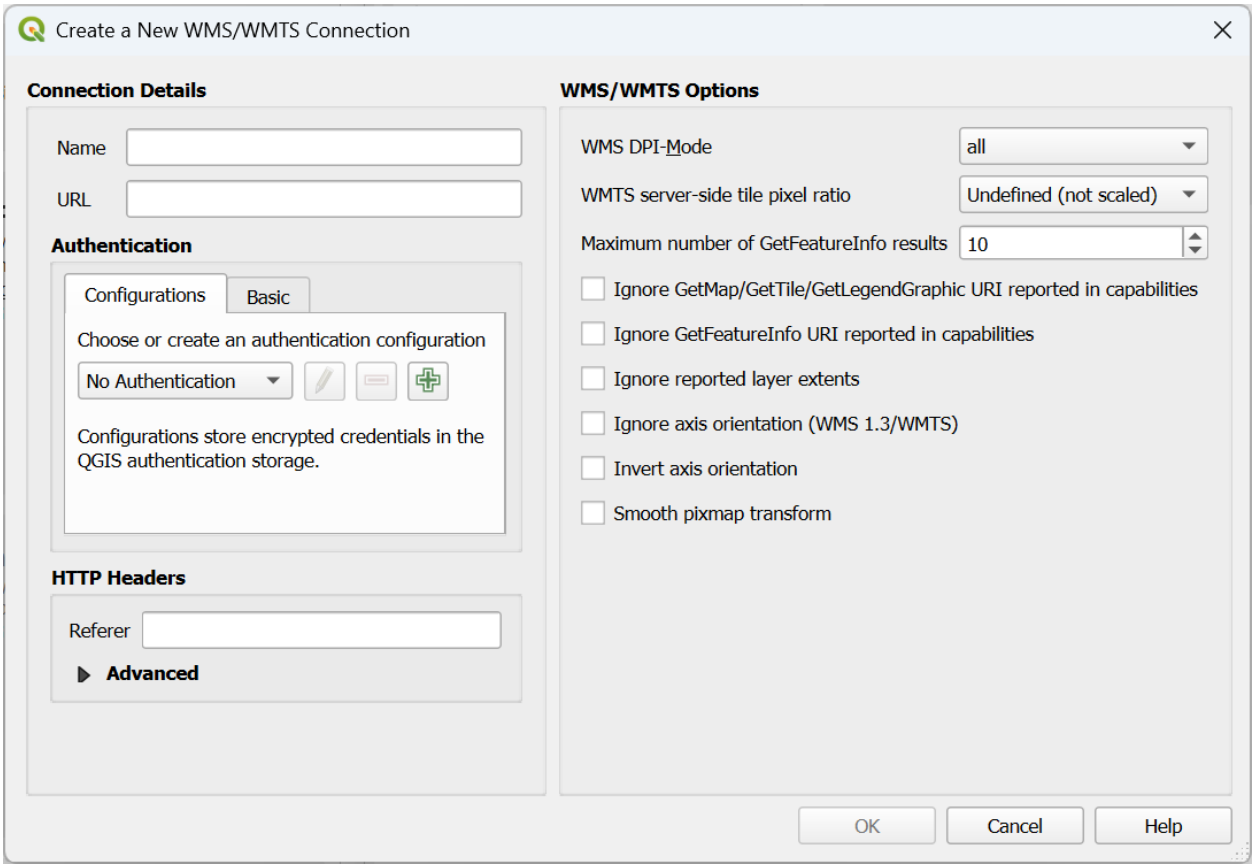



그림 19.2: WMS 서버에 연결 생성하기

- *Name*: 해당 연결을 위한 이름입니다. 사용자가 다른 WMS 서버와 구별할 수 있도록, *Server Connections* 드롭다운 목록에 이 이름을 추가할 것입니다.
- *URL*: 데이터를 제공하는 서버의 URL 입니다. 이 URL 은 분해할 수 있는 호스트명 – 사용자가 텔넷 연결을 열거나 또는 호스트를 핑하기 위해 사용할 URL 과 동일한, 예를 들면 기본 URL 만 있는 서식이어야만 합니다. 예를 들어, 사용자 URL 에 `request=GetCapabilities` 또는 `version=1.0.0` 같은 부가 문자열이 있어서는 안 됩니다.
- *Authentication* (선택적): 저장된 환경 설정 또는 *Username* 과 *Password* 를 가진 기본 인증을 사용합니다.

> **경고**
>
> *Authentication* 탭에서 **사용자명** 과 **비밀번호** 를 입력하면 연결 환경 설정에 보안되지 않은 인증 정보를 유지하게 됩니다. 예를 들면, 다른 사용자와 프로젝트 파일을 공유하는 경우, 이런 **인증 정보가 노출될 것입니다**. 따라서 인증 환경 설정 (*Configurations* 탭) 에 사용자 인증 정보를 대신

저장하는 편이 좋습니다. 더 자세한 내용은인증 시스템 을 참조하세요.

- HTTP *Referer*
- *WMS DPI-Mode*: **all**, **off**, **QGIS**, **UMN** 및 **GeoServer** 옵션 가운데 선택할 수 있습니다.
- *WMTS server-side tile pixel ratio*: WMTS 레이어 렌더링 시 기기의 스크린 DPI 를 기준으로 타일의 크기를 키우거나 줄일 수 있습니다. *Undefined (not scaled)*, *Standard (96 DPI)* 및 *High (192 DPI)* 옵션 가운데 선택할 수 있습니다.
- *Maximum number of GetFeatureInfo results*: GetFeatureInfo 요청이 이 연결을 사용해서 반환하는 레이어 당 결과물의 최대 개수에 대한 기본값을 지정합니다. (FEATURE_COUNT 파라미터를 참조하세요.) 기본값은 `10` 입니다. 서버 기본값을 사용하려면 (보통 `1` 입니다) `0` 으로 설정하십시오. 이 경우 요청에 FEATURE_COUNT 파라미터가 추가되지 않을 것입니다.
- ☐ *Ignore GetMap/GetTile/GetLegendGraphic URI reported in capabilities*: 이 옵션을 체크하면, 위에 있는 *URL* 필드에서 지정한 URI 를 사용합니다.
- ☐ *Ignore GetFeatureInfo URI reported in capabilities*: 이 옵션을 체크하면, 위에 있는 *URL* 필드에서 지정한 URI 를 사용합니다.
- ☐ *Ignore reported layer extents*: (특히 데이터 범위보다 공간을 더 차지하는 심볼을 가진 WMS 서버의 경우) 래스터 레이어가 보고한 영역이 렌더링할 수 있는 실제 면적보다 작을 수도 있기 때문에, 이 옵션을 체크하면 래스터 레이어를 보고된 영역대로 잘라 레이어 경계선에 잘린 심볼을 산출하는 일을 피할 수 있습니다.
- ☐ *Ignore axis orientation (WMS 1.3/WMTS)*
- ☐ *Invert axis orientation*
- ☐ *Smooth pixmap transformation*

4. *OK* 를 클릭하십시오.

새로운 WMS/WMTS 서버 연결을 생성하고 나면, 향후의 QGIS 세션을 위해 새 연결을 저장할 것입니다. `.XML` 파일로부터 연결을 불러오거나 *Load* 또는 연결을 `.XML` 파일로 저장 *Save* 할 수도 있다는 점을 기억하십시오.

인터넷에서 WMS 서비스를 받아올 수 있으려면 프록시 서버를 설정해야 하는 경우, 옵션에서 사용자 프록시 서버를 추가할 수 있습니다. *Settings* → *Options* 메뉴를 선택하고 *Network* 탭을 선택하십시오. 이 탭에서 사용자의 프록시 설정을 추가한 다음 ☑ *Use proxy for web access* 옵션을 체크하면 프록시 서버를 활성화시킬 수 있습니다. *Proxy type* … ▾ 드롭다운 목록에서 올바른 프록시 유형을 선택했는지 확인해야 합니다.

### 19.1.4 WMS/WMTS 레이어 불러오기

사용자 파라미터를 성공적으로 입력하고 나면, *Connect* 버튼을 클릭해서 선택한 서버의 기능 (capability) 을 받아올 수 있습니다. 이 기능은 이미지 인코딩, 레이어, 레이어 스타일 그리고 투영체를 담고 있습니다. 네트워크 작업이기 때문에, WMS 서버에 연결된 사용자 네트워크의 품질에 따라 응답 속도가 달라집니다. WMS 서버에서 데이터를 다운로드하는 동안, 메인 QGIS 대화창 좌하단에 다운로드 진행 상태를 표시합니다.

사용자의 스크린이 이제 그림 19.3 과 비슷하게 보일 것입니다. 이 그림은 WMS 서버의 응답을 보여주고 있습니다.

대화창의 *Layers* 탭 상단에는 서버가 서비스하는 관련 이미지 스타일 (들) 을 가진 레이어들을 내장하고 있는 레이어 그룹들을 포함할 수 있는 트리 구조가 표시됩니다. 각 항목은 다음으로 식별됩니다:

- *ID*
- *Name*: 서비스 이름입니다.

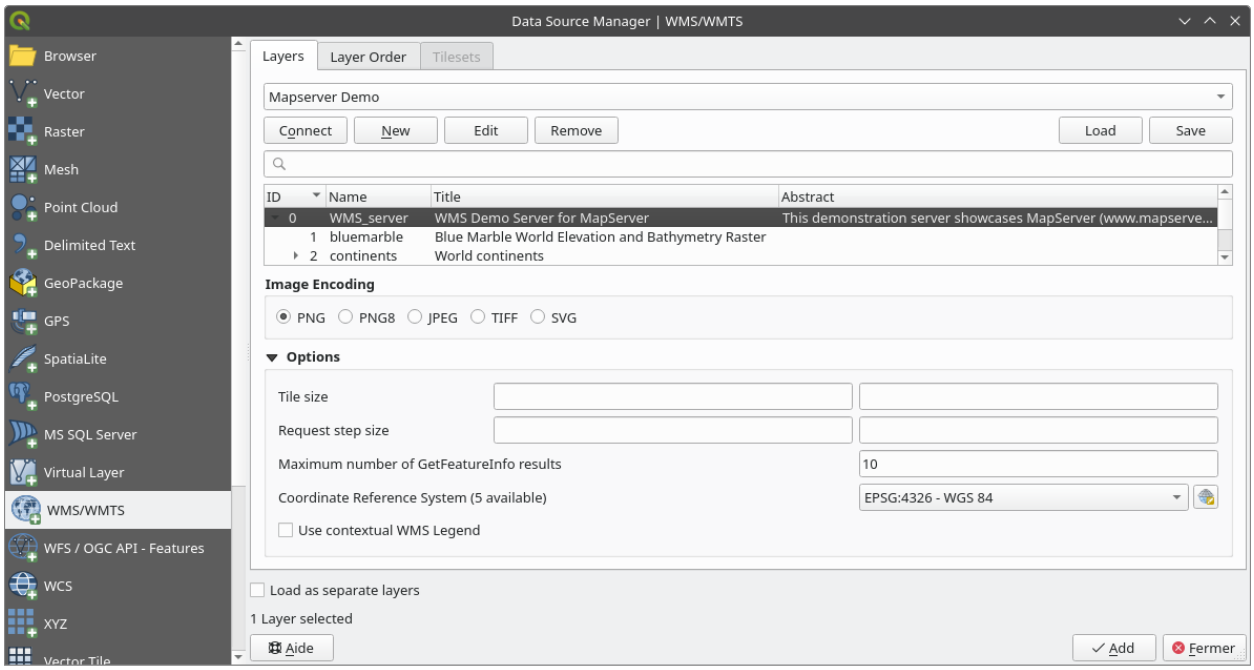



그림 19.3: 사용할 수 있는 레이어를 필터링해서 WMS 서버를 추가하는 대화창

- *Title*
- 그리고 *Abstract*

우상단에 있는  위젯을 사용해서 이 목록을 필터링할 수 있습니다.

### 이미지 인코딩

*Image encoding* 옵션은 클라이언트와 서버 둘 다 지원하는 이미지 포맷들을 표시합니다. 사용자의 이미지 정확도 요구 사항에 따라 하나를 선택하십시오.

> **팁**
>
> **이미지 인코딩**
>
> WMS 서버는 일반적으로 JPEG 또는 PNG 이미지 인코딩 가운데 하나를 선택할 수 있는 옵션을 제공하고 있습니다. JPEG 은 손실 압축 포맷인 반면, PNG 는 원시 (raw) 래스터 데이터를 충실히 재현합니다.
>
> WMS 데이터가 본질적으로 사진 같기를 기대하고, 그리고/또는 이미지 품질이 약간 떨어져도 괜찮다면 JPEG 을 선택하십시오. PNG 와 비교하면 데이터 전송 요구 사항이 일반적으로 1/5 로 줄어듭니다.
>
> 원본 데이터를 정확히 재현하기를 원하고 데이터 전송 요구 사항이 늘어나도 괜찮다면 PNG 를 선택하십시오.

### 옵션

대화창의 옵션 부분에서 WMS 요청을 환경 설정합니다. 다음을 정의할 수 있습니다:

- WMS 요청을 여러 요청으로 나누기 위해 타일 크기를 설정하려는 경우 (예: 256x256) *Tile size* 에서 타일 크기를 설정할 수 있습니다.
- *Request step size*: 타일 경계에서의 라벨 자르기 효과를 줄이기 원한다면, 이 단계 크기를 늘릴수록 요청은 커지고 타일 및 경계는 줄어들 것입니다. 기본값은 2,000 입니다.

- *Maximum number of GetFeatureInfo results*: GetFeatureInfo 요청이 불러온 레이어 (들) 에 대해 반환하는 결과물의 최대 개수를 지정합니다. 특정 레이어에 대해연결 기본값 을 무시할 수 있게 해줍니다.
- WMS 서버의 케이퍼빌리티에 따라 각 WMS 레이어를 여러 개의 좌표계로 표현할 수 있습니다. 목록에서 WMS 를 선택하면 웹 서버가 제공하는 기본 투영법을 가진 필드가 나타납니다. WMS 기본 투영법을 WMS 서버가 지원하는 또다른 좌표계로 대체하려면 Select CRS 위젯을 클릭하십시오.

  그림 6.4 과 비슷한 대화창이 뜰 것입니다. WMS 버전 대화창과 가장 다른 점은 WMS 서버가 지원하는 좌표계들만 표시될 것이라는 사실입니다.
- 마지막으로 WMS 서버가 해당 기능을 지원하는 경우 *Use contextual WMS Legend* 옵션을 활성화할 수 있습니다. 이 옵션을 활성화하면 사용자의 현재 맵 뷰 범위에 관련이 있는 범례만 표시할 것입니다. 즉 사용자가 현재 맵에서 볼 수 없는 항목에 대한 범례 항목은 담지 않는다는 뜻입니다.

You can select several layers at once, but only one image style per layer. When several layers are selected, they will be combined at the WMS server and transmitted to QGIS in one go, as a single layer. The default name is a slash (/) separated list of their original title. You can however opt to *Load as separate layers*, in which case each layer is added to the map canvas using its default name from the queried service.

**레이어 정렬**

*Layer Order* 탭은 현재 연결된 WMS 서버에서 사용할 수 있는 레이어들 가운데 선택한 레이어의 목록을 담고 있습니다.

WMS 서버는 레이어를 *Layers* 탭에 있는 목록의 순서대로, 목록의 맨 위에서 맨 아래 순서로 중첩시켜 렌더링합니다. 이 중첩 순서를 변경하고 싶다면, *Layer Order* 탭의 *Up* 과 *Down* 버튼을 사용해서 목록 순서를 변경할 수 있습니다.

**투명도**

*Layer Properties* 에 있는 *Global transparency* 설정은 적용할 수 있는 경우라면 언제 어디서든 활성화돼 있도록 하드코딩돼 있습니다.

### 19.1.5 타일셋

WMTS (캐시된 WMS) 서비스를 이용할 때 서버가 제공하는 *Tilesets* 탭을 통해 타일셋을 탐색할 수 있습니다. 이 테이블은 타일 크기, 포맷, 그리고 지원하는 좌표계 같은 추가적인 정보 목록을 담고 있습니다.

```
# example of WMTS service

https://opencache.statkart.no/gatekeeper/gk/gk.open_wmts?service=WMTS&
↪request=GetCapabilities
```

불러올 레이어를 선택하면, QGIS 의 일반적인래스터 렌더링 작업자 를 이용해서 스타일 작업을 할 수 있는 단일 밴드 부동소수점 유형의 래스터 레이어로 변환하는해석 방법 을 적용할 수도 있습니다.

이 기능과 함께, *View* → *Panels* (또는 *Settings* → *Panels*) 메뉴 옵션을 클릭한 다음 *Tile Scale Panel* 을 선택하면 타일 축척 슬라이드 바를 이용할 수 있습니다. 이렇게 하면 타일 서버에서 사용할 수 있는 축척들을 도킹된 멋진 슬라이드 바를 통해 알 수 있습니다.

### 19.1.6 식별 도구 이용하기

WMS 서버를 추가한 후에 그리고 WMS 서버에 쿼리 가능한 레이어가 있는 경우,  Identify 도구를 사용해서 맵 캔버스 상에 있는 픽셀을 선택할 수 있습니다. 각 픽셀을 선택할 때마다 WMS 서버로 쿼리를 전송합니다. 쿼리 결과는 평문 텍스트로 반환됩니다. 이 텍스트의 서식은 어떤 WMS 서버와 연결 중이냐에 따라 달라집니다.

**포맷 선택**

서버가 여러 출력 포맷을 지원하는 경우, 식별 결과 대화창에 지원 포맷을 담은 콤보박스가 자동적으로 추가되며, 프로젝트에 레이어에 대해 선택한 포맷을 저장할 수도 있습니다.

**GML 포맷 지원**

 Identify 도구는 WMS 서버 응답 (`GetFeatureInfo`) 을 GML 포맷으로 지원합니다. (현재 맥락의 QGIS GUI 내에서는 이 GML 을 객체라고 부르겠습니다.) 서버가"객체"포맷을 지원하고 선택한 경우, 식별 도구의 결과는 일반 벡터 레이어에서 나온 벡터 객체입니다. 트리에서 단일 객체를 선택한 경우, 맵에서 해당 객체가 강조되며 클립보드로 복사해서 다른 벡터 레이어로 붙여넣을 수 있습니다. UMN 맵서버가 `GetFeatureInfo` 를 GML 포맷으로 지원하도록 설정하는 다음 예시를 참조하세요.

```
# in layer METADATA add which fields should be included and define geometry (example):

"gml_include_items"   "all"
"ows_geometries"      "mygeom"
"ows_mygeom_type"     "polygon"

# Then there are two possibilities/formats available, see a) and b):

# a) basic (output is generated by Mapserver and does not contain XSD)
# in WEB METADATA define formats (example):
"wms_getfeatureinfo_formatlist" "application/vnd.ogc.gml,text/html"

# b) using OGR (output is generated by OGR, it is sent as multipart and contains XSD)
# in MAP define OUTPUTFORMAT (example):
OUTPUTFORMAT
    NAME "OGRGML"
    MIMETYPE "ogr/gml"
    DRIVER "OGR/GML"
    FORMATOPTION "FORM=multipart"
END

# in WEB METADATA define formats (example):
"wms_getfeatureinfo_formatlist" "OGRGML,text/html"
```

### 19.1.7 속성 살펴보기

WMS 서버를 추가하고 나면, 범례에서 서버를 오른쪽 클릭하고 *Properties* 메뉴 항목을 선택하면 서버 속성을 살펴볼 수 있습니다. WMS/WMTS 레이어 속성은 래스터 레이어 속성과 꽤 비슷하기 때문에래스터 속성 대화창 에서 자세한 설명을 찾아 볼 수 있습니다. 하지만 몇몇 차이점이 존재하는데, 이를 이제부터 설명할 것입니다.

정보 속성

#### 메타데이터 탭

*Metadata* 탭은 WMS 서버에 관한 풍부한 정보를 표시합니다. 일반적으로 해당 서버가 반환한 기능 선언문에서 수집한 정보입니다. WMS 표준 사양 (참고 문헌 및 웹사이트 에 있는 OPEN-GEOSPATIAL-CONSORTIUM 을 참조하세요) 을 읽어보면 많은 정의들을 얻을 수 있지만, 사용자의 편의를 위해 다음 정의들을 모아봤습니다:

- 서버 속성
  - **WMS Version** — 서버가 지원하는 WMS 버전입니다.
  - **Image Formats** — 맵 렌더링 시 서버가 응답할 수 있는 MIME 유형의 목록입니다. QGIS 는 기저 Qt 라이브러리와 함께 빌드된 어떤 포맷이든 지원하고 있습니다. 일반적으로, 최소한 `image/png` 및 `image/jpeg` 유형은 지원합니다.
  - **Identity Formats** — 식별 도구 사용 시 서버가 응답할 수 있는 MIME 유형의 목록입니다. QGIS 는 현재 `text-plain` 유형을 지원하고 있습니다.
- 레이어 속성
  - **Selected** — 현재 프로젝트에 서버를 추가했을 때 현재 레이어를 선택했는지 여부를 결정합니다.
  - **Visible** — 범례에 현재 레이어를 가시화하도록 선택했는지 여부를 결정합니다. (QGIS 현재 버전에서는 아직 사용되지 않습니다.)
  - **Can Identify** — 현재 레이어에 식별 도구를 사용했을 때 레이어가 어떤 결과든 반환할 것인지 여부를 결정합니다.
  - **Can be Transparent** — 현재 레이어를 투명도를 적용해서 렌더링할 수 있는지 여부를 결정합니다. 이 속성이 `Yes` 이고 이미지 인코딩이 투명도를 지원하는 경우 QGIS 현재 버전은 언제나 투명도를 적용할 것입니다.
  - **Can Zoom In** — 서버가 현재 레이어를 확대할 수 있는지 여부를 결정합니다. QGIS 현재 버전은 모든 WMS 레이어의 이 속성이 `Yes` 로 설정돼 있다고 가정합니다. 이 속성이 `No` 인 레이어는 이상하게 렌더링될 수도 있습니다.
  - **Cascade Count** — WMS 서버는 레이어를 위한 래스터 데이터를 얻기 위해 다른 WMS 서버에 대한 프록시 역할을 할 수 있습니다. 이 속성은 피어 WMS 서버가 반환하기 전에 현재 레이어에 대한 요청을 몇 번이나 포워딩했는지를 알려줍니다.
  - **Fixed Width, Fixed Height** — 현재 레이어가 고정된 소스 픽셀 크기를 가지고 있는지 여부를 결정합니다. QGIS 현재 버전은 모든 WMS 레이어의 이 속성이 설정돼 있지 않다고 가정합니다. 설정돼 있는 레이어는 이상하게 렌더링될 수도 있습니다.
  - **WGS 84 Bounding Box** — 레이어의 WGS 84 좌표가 적용된 경계 상자입니다. 일부 WMS 서버에서는 이 속성이 올바로 설정돼 있지 않습니다. (예를 들면 UTM 좌표를 대신 적용한 경우가 있습니다.) 만약 이런 경우라면, QGIS 가 해당 레이어의 초기 뷰를 굉장히'축소된'모습으로 렌더링할 수도 있습니다. 이런 오류는 WMS 웹마스터에게 반드시 알려야 합니다. WMS XML 항목 가운데 `LatLonBoundingBox`, `EX_GeographicBoundingBox` 또는 CRS:84 `BoundingBox` 가 문제일 수 있습니다.
  - **Available in CRS** — WMS 서버가 현재 레이어를 렌더링할 수 있는 투영체를 결정합니다. 이 투영체들은 WMS 고유 서식으로 정리돼 있습니다.
  - **Available in style** — WMS 서버가 현재 레이어를 렌더링할 수 있는 이미지 스타일을 결정합니다.

### 시계열 속성

WMS 및 WMTS 레이어에 래스터시계열 속성 을 (즉 *Dynamic Temporal Control* 을) 설정할 수 있습니다. 기본적으로 프로젝트에 시간 차원을 활성화한 WMS 또는 WMTS 레이어를 추가하는 경우, *Layers* 패널의 해당 레이어 옆에 Temporal Layer 아이콘을 표시해서 시간 차원이 활성화됐다는 사실을 나타냅니다. 이 레이어의 *Temporal* 속성의 기본값은 *Automatic* 시계열 모드로, 레이어가 기본적으로 시계열 제어기의 현재 시간 범위를 따를 것이라는 뜻입니다.

그리고 레이어의 특정 정적 시간값을 표시하도록 선택하려면 *Dynamic Temporal Control* 옵션을 체크 해제하고 *Static WMS-T Temporal Range* 프레임에서 다음 옵션 가운데 하나를 선택하면 됩니다:

- *Server default*: 서버 기본 시간 범위를 표시합니다.
- 서버가 알려주는 불연속 시계열 범위 데이터를 가진 *Predefined date* 또는 서버가 알려주는 사용할 수 있는 날짜들의 범위를 가진 *Predefined range* 를 선택할 수 있습니다. 후자의 경우 *Start date* 와 *End date* 가 필수입니다. 제공자가 연속하는 기간에 대한 데이터를 가지고 있는지 아닌지 여부에 따라 참조 시간 옵션에서 예상 날짜 서식을 추정할 수 있습니다. (다음 그림을 참조하세요.)
- *Follow project's temporal range*: 프로젝트 속성 대화창에서 정의된 프로젝트 수준 시계열 범위를 따릅니다.

어떤 시계열 데이터 제어를 사용하든, 정확한 시계열 데이터를 표시하게 해주는 몇몇 *WMS-T Settings* 가 있습니다:

- *Time slice mode*: 시간 슬라이스 모드를 다음 가운데 하나로 설정할 수 있습니다:
  - *Use whole temporal range*: 전체 시계열 범위를 사용
  - *Match to start of range*: 시계열 범위의 시작에 맞춤
  - *Match to end of range*: 시계열 범위의 종단에 맞춤
  - *Closest match to start of range*: 시계열 범위의 시작에 최대한 맞춤
  - *Closest match to end of range*: 시계열 범위의 종단에 최대한 맞춤
- *Ignore time components (use dates only)*: 이 옵션을 체크하면 서버 요청에서 시계열 쿼리의 시간 요소를 버리고 날짜 요소만 사용할 것입니다.

*Use Specific WMS-T Reference Time* 을 체크해서 레이어의 케이퍼빌리티에 보고된 시간으로부터 선택할 수도 있습니다. 서버가 (날짜 범위 대신) 날짜 & 시간 인스턴스의 불연속 집합을 알려주는 경우 유용합니다.

### QGIS 서버 속성

래스터 레이어속성 뿐만 아니라, QGIS 서버로 WMS/WMTS 레이어를 게시하는 경우 다음과 같은 옵션도 표시할 것입니다:

- *WMS Print layer*: (GetProjectSettings 응답에서) 인쇄에 쓰여야 할 대체 WMS 레이어를 설정할 수 있습니다. WMTS 레이어가 일반적인 상황에서 인쇄에 적합하지 않을 경우 유용합니다.
- *Publish WMS/WMTS data source uri*: 웹 클라이언트가 WMS/WMTS 데이터를 직접 가져올 수 있게 해줍니다.
- *Advertise as background layer*: WMS/WMTS 레이어를 배경 레이어로 알립니다.

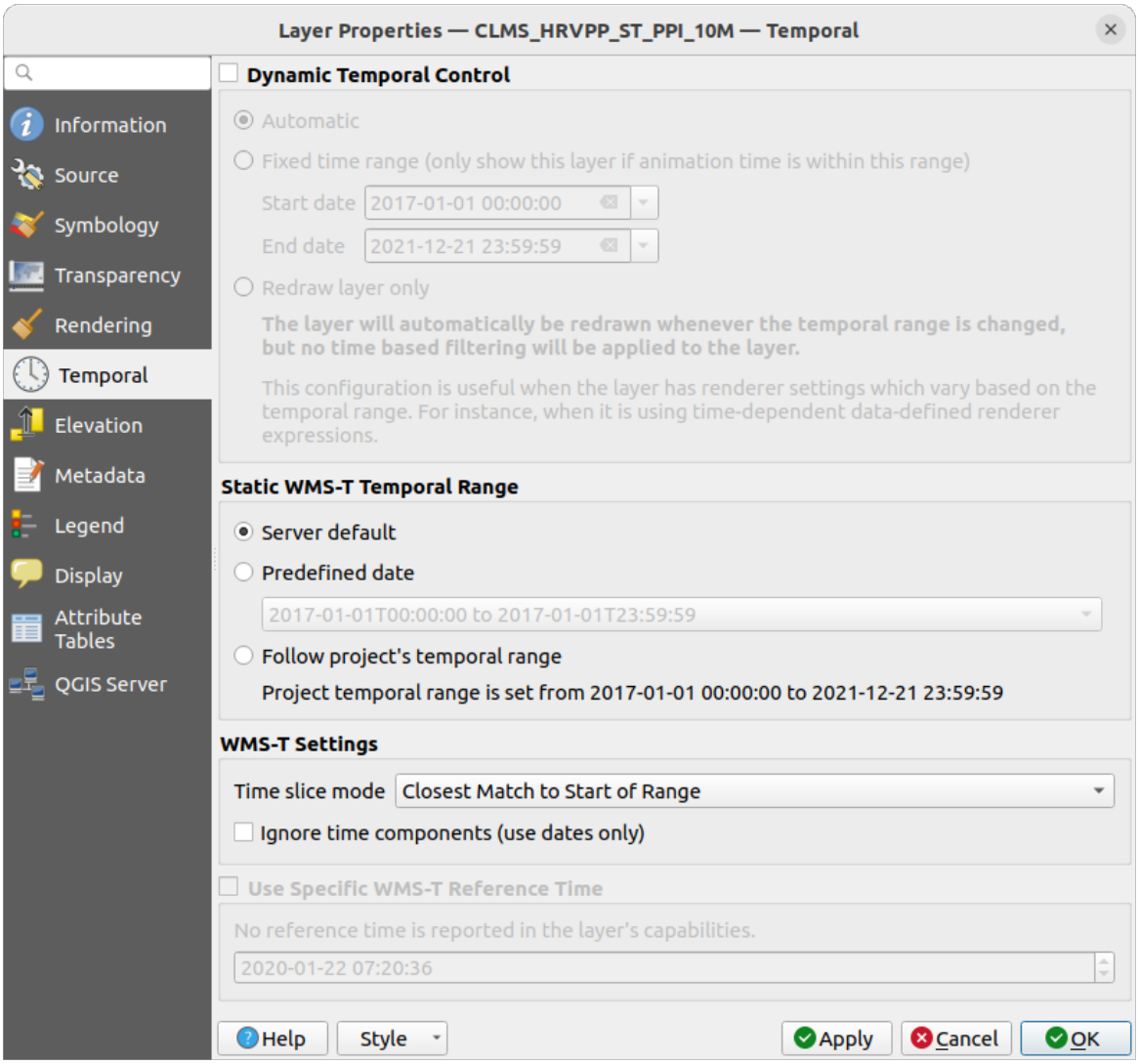



그림 19.4: WMTS 레이어의 시계열 속성

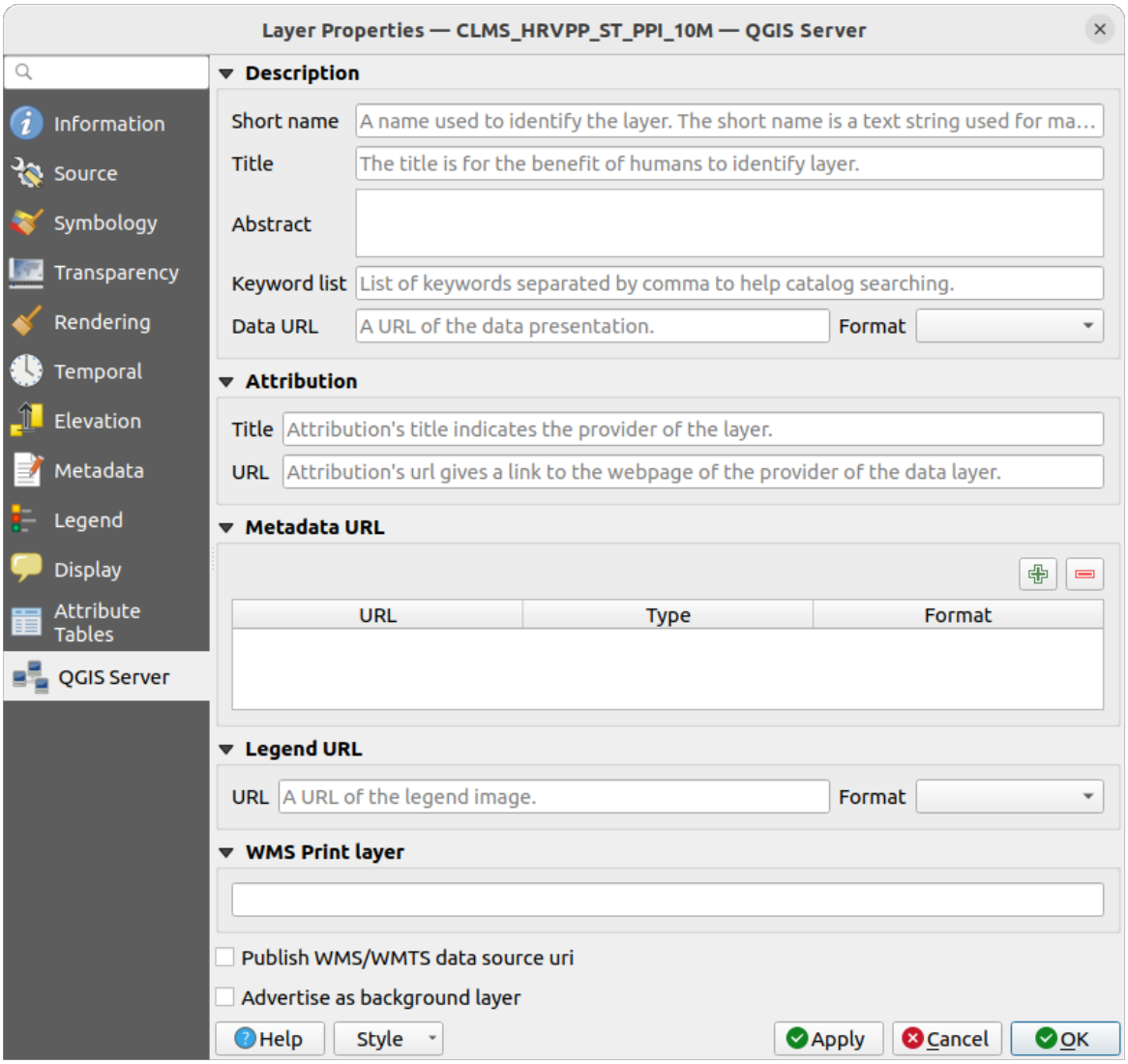



그림 19.5: WMS/WMTS 레이어의 QGIS 서버 속성

### 19.1.8 차례 및 조판기에 WMS 범례 그래픽 표시하기

QGIS WMS 데이터 제공자는 차례 (table of content) 의 레이어 목록 및 인쇄 조판기에 WMS 범례 그래픽을 표시할 수 있습니다. WMS 서버가 `GetLegendGraphic` 기능을 보유하고 있고 레이어에 `getCapability` URL 을 설정한 경우에만 WMS 범례를 표시할 것입니다. 따라서 해당 레이어를 위한 스타일을 따로 선택해야 합니다.

LegendGraphic 을 사용할 수 있는 경우, 레이어 아래에 표시합니다. LegendGraphic 이 매우 작기 때문에 (`QgsLegendInterface` 의 구조적 제약 때문에) 실제 크기로 열려면 LegendGraphic 을 클릭해야 합니다. 레이어의 범례를 클릭하면 전체 해상도의 범례를 담은 프레임이 열릴 것입니다.

인쇄 조판기에서는, 범례가 원본 (다운로드된) 크기로 통합될 것입니다. *Legend* ▸ *WMS LegendGraphic* 메뉴의 항목 속성 패널에서 범례 그래픽의 해상도를 사용자의 인쇄 작업 요구 사항에 맞게 설정할 수 있습니다.

이 범례는 사용자의 현재 축척을 기반으로 맥락에 맞는 정보를 표시할 것입니다. WMS 서버가 `GetLegendGraphic` 기능을 보유하고 있고 레이어에 `getCapability` URL 을 설정한 경우에만 WMS 범례를 표시할 것입니다. 따라서 해당 레이어를 위한 스타일을 따로 선택해야 합니다.

## 19.2 WCS 클라이언트

WCS(Web Coverage Service) 는 과학적 모델에 입력하기 위한 클라이언트 쪽 렌더링에 대해 그리고 기타 클라이언트에 대해 유용한 형태로 래스터 데이터에 접근할 수 있습니다. WCS 를 WFS 및 WMS 에 빗대어 설명할 수도 있습니다. WCS 는, WMS 및 WFS 서비스 인스턴스로서, 클라이언트가 공간 제약 조건 및 기타 쿼리 기준을 기반으로 서버가 보유하고 있는 정보의 일부분을 선택하게 할 수 있습니다.

QGIS 는 WCS 제공자를 내장하고 있으며, (서로 상당히 다른) 1.0 및 1.1 버전을 지원하고 있습니다. 그러나 현재 1.1 버전에 많은 문제점이 있기 때문에 1.0 버전을 선호합니다. (예를 들면 1.1 버전은 각 서버가 다양하고 독특한 설정들 덕분에 모두 서로 다른 방식으로 실행됩니다.)

내장 WCS 제공자는 모든 네트워크 요청을 처리하며 모든 표준 QGIS 네트워크 설정을 (특히 프록시 설정을) 사용합니다. 제공자가 캐시 모드 ('always cache', 'prefer cache', 'prefer network', 'always network') 를 사용할 수도 있고, 서버가 시간 영역 (temporal domain) 을 제공하는 경우 시점 (time position) 도 선택할 수 있습니다.

**Loading a WCS Layer**

To be able to load a WCS Layer, first create a connection to the WCS server:

1. Open Data Source Manager 버튼을 클릭해서 *Data Source Manager* 대화창을 여십시오.
2. Enable the *WCS* tab
3. Click on *New*⋯ to open the *Create a New WCS Connection* dialog
   - *Name*: A name for the connection. This name will be used in the Server Connections drop-down box so that you can distinguish it from other WCS servers.
   - *URL*: 데이터를 제공하는 서버의 URL 입니다. 이 URL 은 분해할 수 있는 호스트명 – 사용자가 텔넷 연결을 열거나 또는 호스트를 핑하기 위해 사용할 URL 과 동일한, 예를 들면 기본 URL 만 있는 서식이어야만 합니다. 예를 들어, 사용자 URL 에 `request=GetCapabilities` 또는 `version=1.0.0` 같은 부가 문자열이 있어서는 안 됩니다.
   - *Authentication* (선택적): 저장된 환경 설정 또는 *Username* 과 *Password* 를 가진 기본 인증을 사용합니다.

> **⚠ 경고**
>
> *Authentication* 탭에서 **사용자명** 과 **비밀번호** 를 입력하면 연결 환경 설정에 보안되지 않은 인증 정보를 유지하게 됩니다. 예를 들면, 다른 사용자와 프로젝트 파일을 공유하는 경우, 이런 **인증**

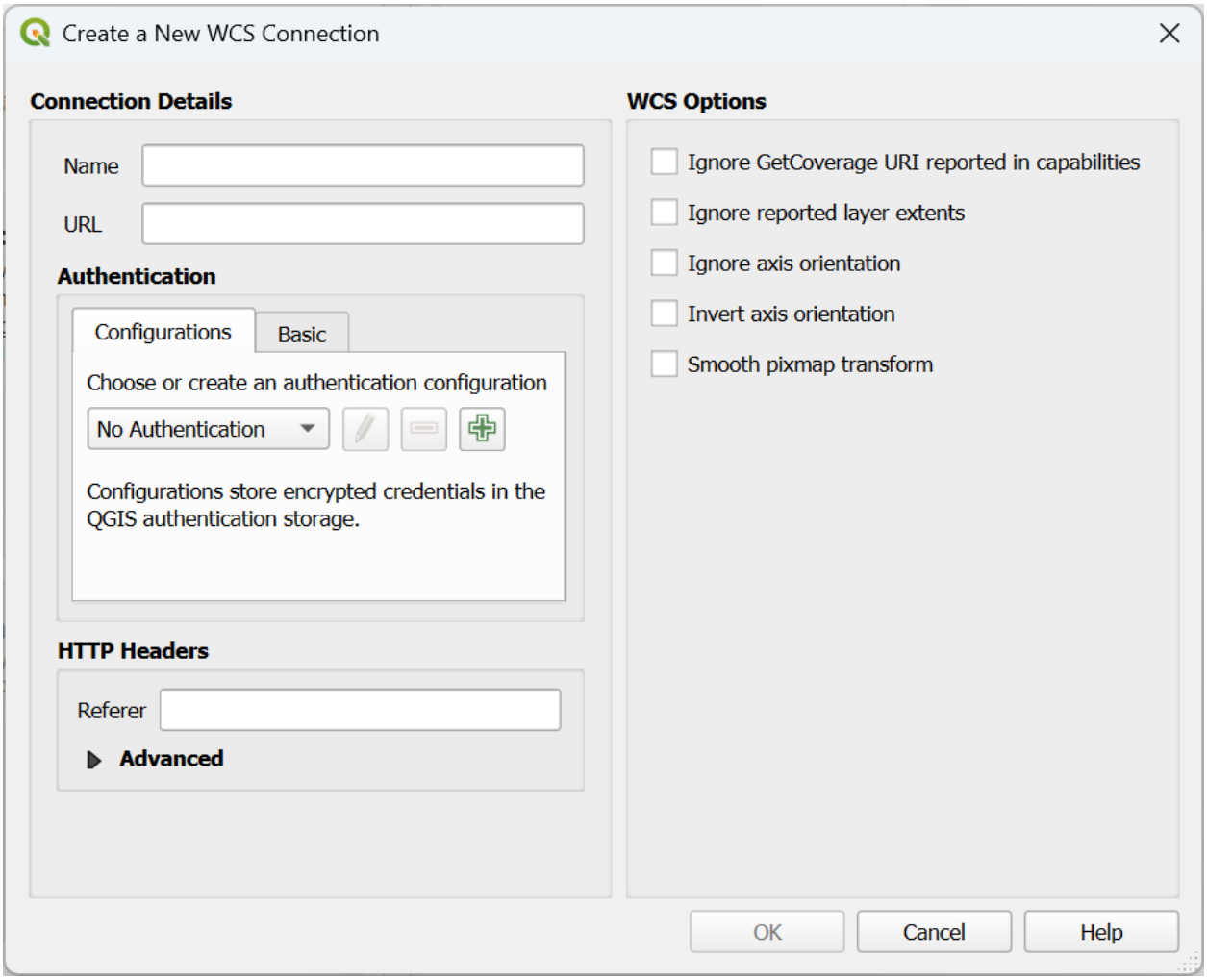



그림 19.6: Creating a connection to a WCS server

**정보가 노출될 것입니다.** 따라서 인증 환경 설정 (*Configurations* 탭) 에 사용자 인증 정보를 대신 저장하는 편이 좋습니다. 더 자세한 내용은인증 시스템 을 참조하세요.

- HTTP *Referer*
- ☐ *Ignore GetCoverage URI reported in capabilities*: if checked, use given URI from the *URL* field above.
- ☐ *Ignore reported layer extents*: because the extent reported by raster layers may be smaller than the actual area which can be rendered (notably for WCS servers with symbology which takes more space than the data extent), check this option to avoid cropping raster layers to their reported extents, resulting in truncated symbols on the borders of these layers.
- ☐ *Ignore axis orientation*
- ☐ *Invert axis orientation*
- ☐ *Smooth pixmap transformation*

4. *OK* 를 클릭해서 연결을 생성하십시오.

사용자가 설정했을 수도 있는 어떤 프록시 설정이라도 인식할 것이라는 사실을 기억하십시오. 또, `.XML` 파일로부터 연결을 불러오거나 *Load* 또는 연결을 `.XML` 파일로 저장 *Save* 할 수도 있다는 점도 기억하십시오.

Now we are ready to load WCS layers from the above connection.

## 19.3 WFS 및 WFS-T 클라이언트

QGIS 에서, WFS 레이어는 다른 모든 벡터 레이어와 매우 비슷하게 동작합니다. 피처를 식별하고 선택할 수 있고, 속성 테이블도 살펴볼 수 있습니다. QGIS 는 WFS 1.0.0, 1.1.0, 2.0 및 OAPIF(OGC API - Features) 를 편집 기능을 (WFS-T 를 통해) 포함해서 지원합니다. QGIS 는 백그라운드 다운로드, 점진적 (progressive) 렌더링, 다운로드한 피처의 디스크 상 (on-disk) 캐시 작업, 그리고 버전 자동 탐지도 지원합니다.

OGC API - Features - Part 4: Create, Replace, Update and Delete 를 구현한 서버의 레이어는 피처를 생성, 수정, 삭제할 수 있도록 편집 모드로 전환시킬 수 있습니다. 이때 생성/수정/삭제한 각 피처에 대해 전용 네트워크 요청이 필수적이기 때문에, 동시에 수백 개 이상의 피처를 한번에 수정하는 경우 처리 속도가 떨어질 수도 있다는 점을 기억하십시오.

일반적으로, WFS 레이어는 WMS 가 사용하는 방식과 매우 유사합니다. 기본 서버가 정의돼 있지 않기 때문에, 사용자가 직접 추가해야 합니다. 메타검색 플러그인 또는 사용자가 선호하는 웹 검색 엔진을 통해 WFS 서버를 찾을 수 있습니다. 공개 URL 목록이 여럿 있는데, 일부는 유지되고 있고 일부는 더 이상 서비스하고 있지 않습니다.

**WFS 레이어 불러오기**

이번 예시는 게이트웨이 지오매틱스 (Gateway Geomatics) WFS 서버를 이용해서 레이어를 표시합니다.

```
https://demo.gatewaygeomatics.com/cgi-bin/wfs_gateway?REQUEST=GetCapabilities&
↪VERSION=1.0.0&SERVICE=WFS
```

WFS 레이어를 불러오려면 먼저 WFS 서버와의 연결을 생성해야 합니다:

1. Open Data Source Manager 버튼을 클릭해서 *Data Source Manager* 대화창을 여십시오.
2. *WFS / OGC API - Features* 탭을 선택하십시오.
3. *New…*버튼을 클릭해서 *Create a New WFS Connection* 대화창을 여십시오.
4. In the *Connection Details* group, enter `Gateway Geomatics` as name

5. (앞에서 언급한) URL 을 입력하십시오.
6. Optionally, enter the *Authentication* settings

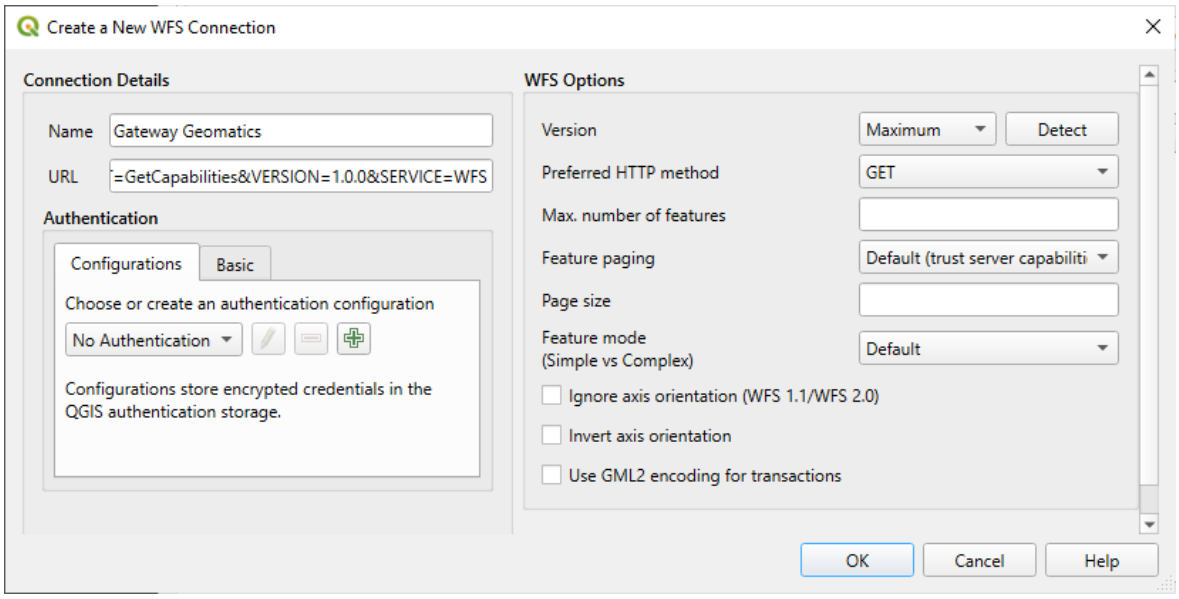



그림 19.7: WFS 서버와의 연결 생성하기

> **참고**
>
> In case of an OGC API - Features (OAPIF), the URL to provide should be the landing page, i.e., the main page from which it is possible to navigate to all the available service endpoints.

7. In the *WFS Options* group, you can:
   - 서버의 WFS 버전을 알 수 있습니다. 표시되지 않는 경우, *Detect* 버튼을 클릭하면 자동으로 받아옵니다.
   - Select the *Preferred HTTP method* to use for requests. The default is *GET*, but you can also select *POST*.
   - Define the *Maximum number of features* retrieved in a single GetFetFeature request. If empty, no limit is set.
   - *Feature mode (simple vs complex)*: helps determine how to handle servers serving data with simple or complex GML schemas. Available options are:
     - *Default*: if the server supports editing capabilities, this is the same as using *Simple Features*. If not, it fallbacks to using *Complex Features* mode.
     - *Simple Features*: returns data as simple features only
     - *Complex Features*: uses the *OGR GMLAS <https://gdal.org/en/stable/drivers/vector/gmlas.html>*, potentially returning complex features, depending on the data schema, where nested data structures are exposed by QGIS as JSON.
   - Depending on the WFS version, indicate whether to:
     - Enable *Feature paging* and specify the maximum number of features to retrieve with *Page size*. If no limit is defined, then the server default is applied.
     - *Ignore axis orientation (WFS 1.1/WFS 2.0)*: 버전 1.1, 2.0 인 경우 축의 방향을 무시하도록 강제할 수 있습니다.

  - *Invert axis orientation*: 축의 방향을 반전시킵니다.
  - *Use GML2 encoding for transactions*: 트랜잭션에 GML2 인코딩을 사용합니다.

> **경고**
>
> *Authentication* 탭에서 **사용자명** 과 **비밀번호** 를 입력하면 연결 환경 설정에 보안되지 않은 인증 정보를 유지하게 됩니다. 예를 들면, 다른 사용자와 프로젝트 파일을 공유하는 경우, 이런 **인증 정보가 노출될 것입니다**. 따라서 인증 환경 설정 (*Configurations* 탭) 에 사용자 인증 정보를 대신 저장하는 편이 좋습니다. 더 자세한 내용은인증 시스템 을 참조하세요.

8. *OK* 를 클릭해서 연결을 생성하십시오.

사용자가 설정했을 수도 있는 어떤 프록시 설정이라도 인식할 것이라는 사실을 기억하십시오. 또, `.XML` 파일로부터 연결을 불러오거나 *Load* 또는 연결을 `.XML` 파일로 저장 *Save* 할 수도 있다는 점도 기억하십시오.

이제 앞에서 설정한 연결로부터 WFS 레이어를 불러올 수 있습니다.

1. *Server Connections* 드롭다운 목록에서'Gateway Geomatics'를 선택하십시오.
2. *Connect* 를 클릭하십시오.
3. 목록에서 *Parks* 레이어를 선택하십시오.
4. 다음 옵션 가운데 하나를 선택할 수 있습니다:
   - *Only request features overlapping the view extent*: 이 옵션을 체크하면 뷰 범위와 겹치는 피처만 요청합니다.
   - *Change*···: 레이어 좌표계를 서비스가 지원하는 다른 좌표계로 변경합니다.
   - 또는 서비스에서 검색할 특정 피처를 지정하는 쿼리를 작성합니다: 레이어 행을 더블클릭하면 *SQL Query Composer* 대화창이 열립니다. 이 대화창은 서비스에서 사용할 수 있는 테이블과 열을 이용해서 고급 SQL 쿼리를 작성할 수 있는 위젯을 제공합니다. 정렬 함수, 필터 함수 같은 수많은 SQL 함수는 물론 공간 술어 (predicate) 및 연산자를 지원합니다.

     사용자가 작성한 쿼리는 검증 후 *WFS / OGC API - Features* 테이블 내부의 *SQL* 열에 나타날 것입니다.

     또한 필터링된 레이어는 *Layers* 패널에  아이콘과 함께 표시될 것입니다. 즉 언제라도 쿼리를 조정할 수 있다는 의미입니다.
5. Click *Add* to add the layer to the map, using its default name from the queried service.

QGIS 메인 창 좌하단에 다운로드 진행 상태가 가시화된다는 걸 볼 수 있을 것입니다. 레이어를 불러오고 나면, 피처를 한두 개 식별하고 선택해서 속성 테이블을 살펴볼 수 있습니다.

## 19.4 SensorThings

QGIS 는 웹 상에서 사물 인터넷 (Internet of Things) 센서 장치, 데이터, 그리고 앱과 상호 연결할 수 있는 개방형 통합 프레임워크를 제공하는 표준인 OGC 센서싱스 API 에 접속하는 기능을 제공합니다. 이 API 는 IoT 의 구문적·의미적 상호 운용성을 다루는 개방형 표준으로, 관측을 위해 그리고 관측 시 샘플링되는 피처들을 위해 표준화된 O&M(Observations and Measurements) 데이터 모델을 기반으로 개발되었습니다.

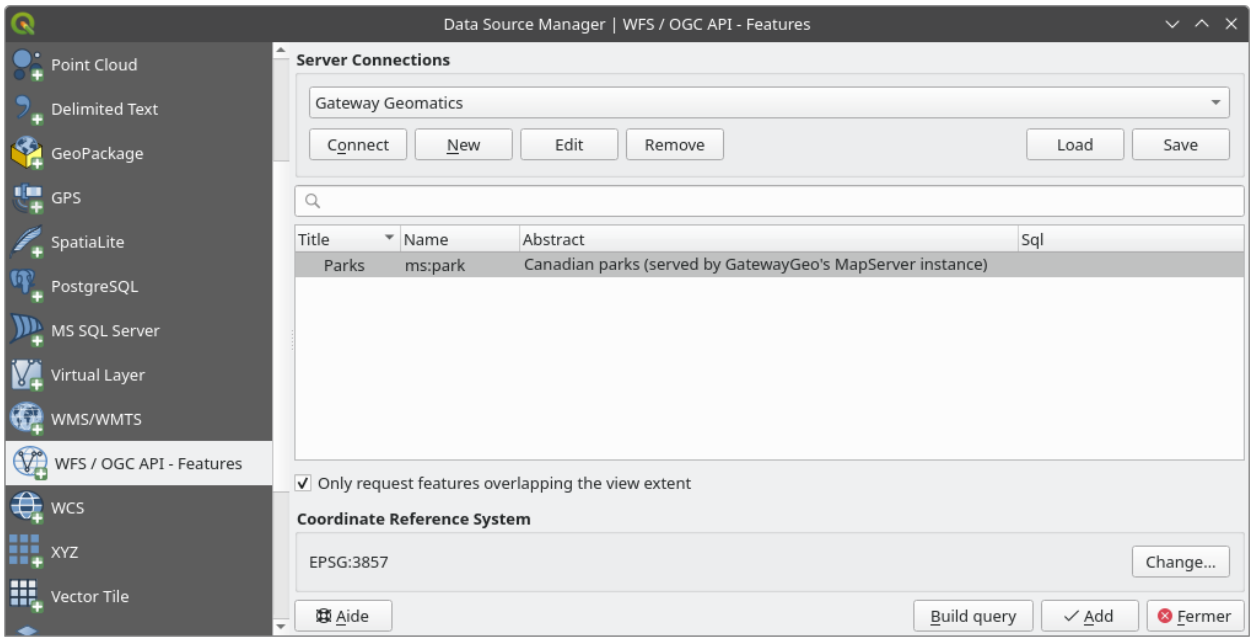



그림 19.8: WFS 레이어 추가하기

### 19.4.1 연결 설정하기

*Data Source Manager* 대화창에 있는 *SensorThings* 탭을 통해 QGIS 에 센서싱스 데이터를 추가할 수 있습니다.

새 연결을 생성하려면 *New* 버튼을 (또는 탐색기 패널에서 *New SensorThings Connection* 버튼을) 누른 다음 *Name* 과 *URL* 을 입력하십시오. 인증 및 *Referer* 같은 고급 옵션들도 환경 설정할 수 있습니다.

*OK* 버튼을 눌러 연결하십시오. 그러면 다음 작업을 할 수 있습니다:

- *Edit*: 센서싱스 연결 설정을 편집할 수 있습니다.
- *Remove*: 센서싱스 연결을 제거할 수 있습니다.

*Data Source Manager* 대화창의 *SensorThings* 항목 (*Save*) 또는 *Browser* 패널에 있는 *SensorThings* 의 컨텍스트 메뉴 (*Save Connections*) 를 통해 환경 설정을 `.XML` 파일로 저장할 수 있습니다. 마찬가지로 파일로부터 환경설정을 추가 (*Load*) 할 수도 있습니다.

### 19.4.2 센서싱스 데이터 불러오기

다음 다이어그램은 센서싱스 데이터셋에 저장된 레이어 (통칭 엔티티) 들 사이의 관계를 나타냅니다.

어떤 유형의 엔티티든 QGIS 로 불러올 수 있지만, 모든 유형이 공간 데이터는 아닙니다. 엔티티를 불러오려면 다음 *Layer Settings* 를 환경 설정해주면 됩니다:

- *Entity Type*: 데이터 모델에서 QGIS 에 레이어로 불러올 엔티티
- *Geometry Type*: 불러오기 위해 선택한 엔티티의 도형 유형. 목록을 실제로 지원되는 도형 유형들로 제한하려면 Check available types 아이콘을 누르십시오.
- *Page Size*: 용지 크기
- *Feature Limit*: 서비스에 요청할 피처의 최대 개수를 설정합니다.

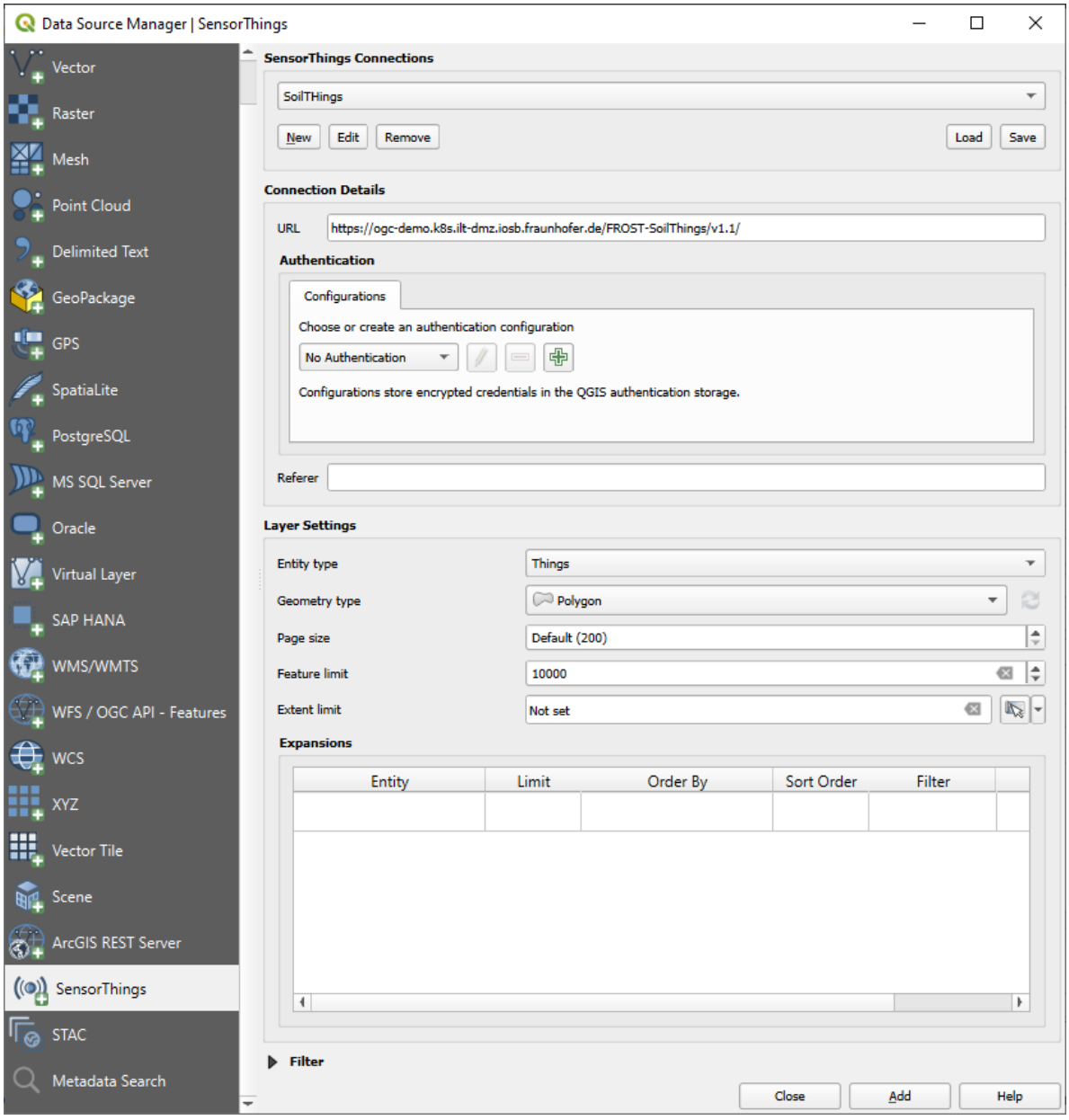



그림 19.9: 센서싱스 연결 대화창

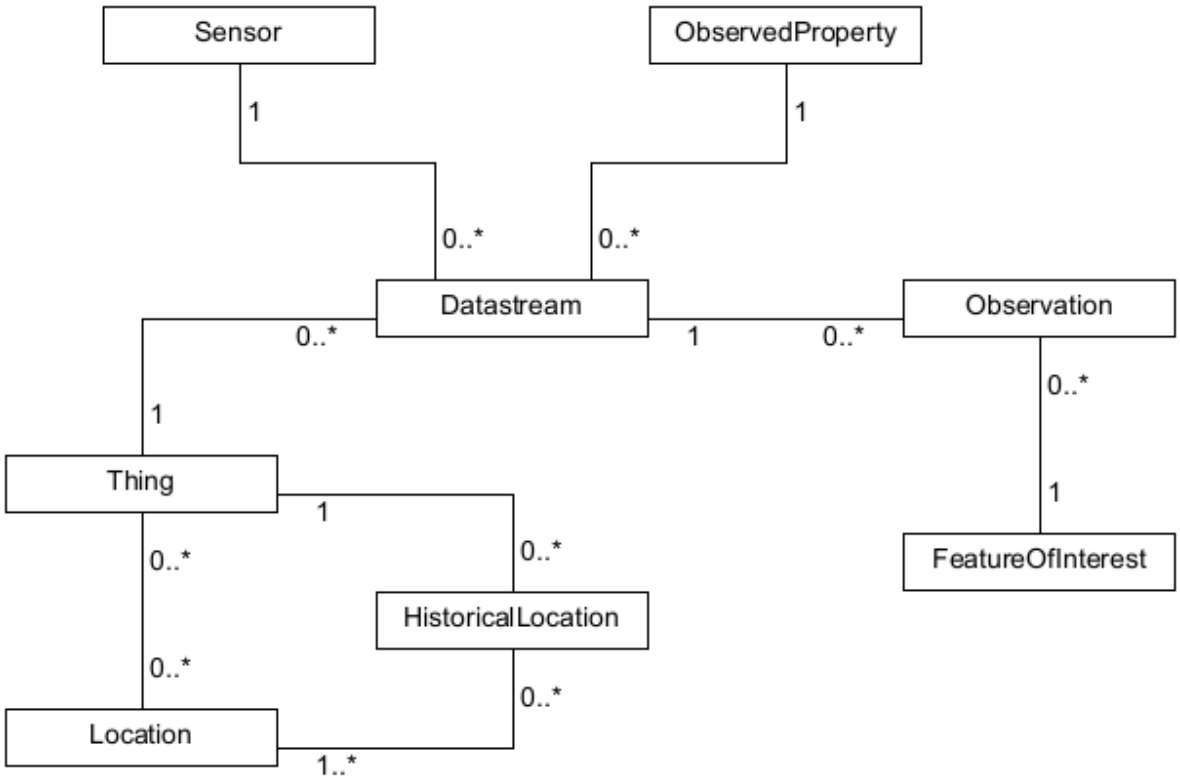



그림 19.10: O&M 데이터 모델 (출처: Sensor Web Tutorial by SIST network)

- *Extent Limit*: 범위 안에 있는 피처들만 요청되도록 레이어에 최대 범위 제한을 설정합니다.
- *Expansions*: 센서싱스 데이터 모델은 관계형 데이터베이스에서 테이블들이 결합되는 것과 유사한 방식으로 결과물을 관련 엔티티로 확장하는 메커니즘을 제공합니다. 이 접근 방식을 이용하면 선택한 레이어가 다른 항목에서 나온 데이터를 포함하도록 확장할 수 있습니다. 이때 하위 피처의 개수만큼 상위 피처를 생성하며 관계를 평탄화시키고, 속성 테이블에 추가 속성을 열로 추가합니다.

*Filter* 에서 데이터를 필터링하는 쿼리를 작성하십시오. 이때 센서싱스 필터 문법을 사용해야 합니다.

> **참고**
>
> QGIS 에 데이터를 불러온 다음에는, 레이어 속성 대화창의 *Source* 탭이 업데이트되어 이런 설정들과 필터링 옵션을 사용할 수 있습니다.

*Add* 를 누르면 QGIS 에 선택한 엔티티 유형을 레이어로 불러옵니다.

### 19.4.3 센서싱스 벡터 레이어 작업하기

QGIS 는 센서싱스 레이어를 벡터 레이어로 불러옵니다. 따라서 센서싱스 레이어의레이어 속성 대화창 은 벡터 레이어 속성 대화창과 동일한 탭들을 가지고 있어 선택 또는 식별 도구들을 통해 동일한 피처 상호 작용을 할 수 있습니다. 하지만 센서싱스 데이터를 작업할 때 고려해야 할 특이 사항들이 있습니다.

데이터 모델 때문에, 센서싱스 관측의 산출물 속성은 문자열 필드입니다. 이를테면 문자열에 있는 숫자 표현을 그라데이션 스타일로 사용하고자 할 때 그 값을 실수로 변환해주는 표현식을 사용해야 합니다. 실수 변환이 실패하는 경우 `try()` 를 사용해보십시오. (예: `try( to_real("Observation_result"), Null)`)

한 군데 이상의 위치에서 관측된 값들로 도표를 생성하고자 하는 경우, *Data Plotly* QGIS 플러그인을 설치하면 됩니다.

1. 이제 맵뷰에 있는 위치 포인트에서 관측값을 선택하십시오.
2. *Data Plotly* 패널을 열어서 *Use only selected features* 체크박스에 체크하십시오.
3. X 열에 날짜 & 시간 속성을 그리고 Y 열에 *Observation_result* 를 선택하십시오. 이렇게 하면 해당 위치의 관측값을 시계열로 볼 수 있는 도표를 그릴 것입니다.
4. 관측 속성이 하나만 나타나도록 필터링하십시오.
5. 맵 상에서 다른 위치를 선택하면 도표가 바뀝니다.

## 19.5 STAC (SpatioTemporal Asset Catalogs)

STAC (SpatioTemporal Asset Catalog) is a specification for describing geospatial data in a consistent and accessible way. It defines a standard structure for organizing and indexing spatial-temporal assets such as satellite imagery, drone photos, or sensor data so that you can search, preview, and use these datasets across different platforms and tools.

STAC enables QGIS users to connect to and browse geospatial datasets, either from static catalogs or dynamic APIs, and download or stream assets such as imagery or other spatial data.

More information at https://stacspec.org/en.

Key components of STAC in QGIS include:

- **STAC Items**: The basic unit of a STAC catalog. Each item represents one geospatial asset or a group of related assets (e.g. image bands) at a specific time and location. It includes metadata such as geometry, datetime, links to assets (e.g., TIFFs, JSON, COG), and properties.

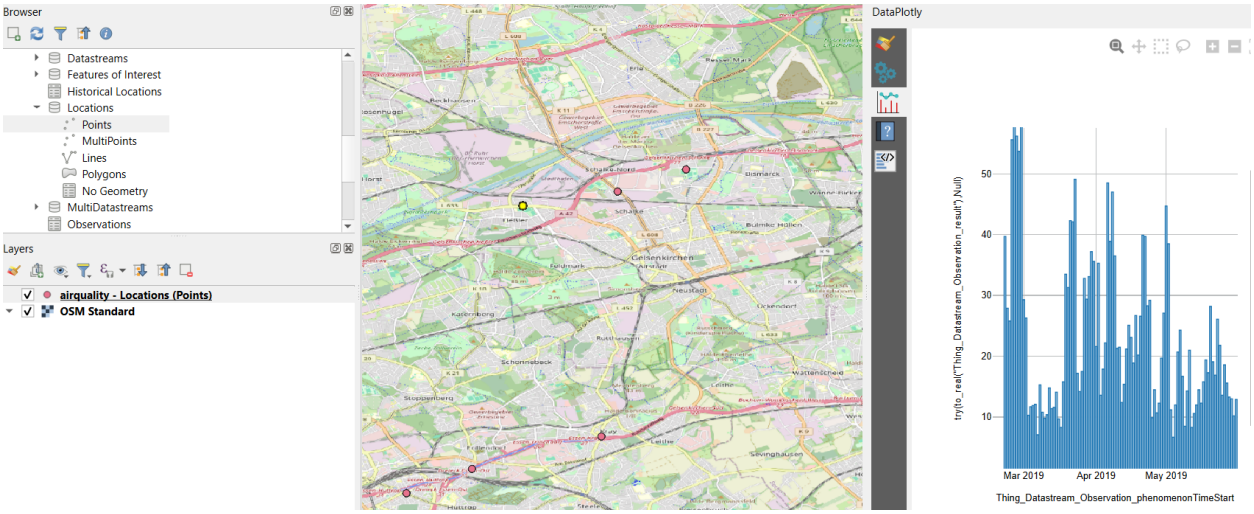



그림 19.11: Data Plotly 를 사용해서 어떤 위치의 공기질 관측값을 도표로 그리기

- **STAC Collections**: A grouping of items that share common characteristics and metadata (e.g. satellite mission). Collections may define spatial and temporal extents, licensing, and keywords.
- **STAC Catalogs**: A hierarchical container that organizes items and/or collections. Catalogs allow navigation of STAC datasets but do not necessarily include search capabilities.
- **STAC API**: An implementation of the STAC specification that allows querying and filtering of STAC items using spatial and temporal based filters, as well as searching within specific collections. STAC APIs follow the OGC API - Features pattern and support dynamic access to datasets.

There is an important distinction between static STAC catalogs and STAC API endpoints:

- **Static STAC Catalogs**: are collections of JSON files without search capabilities. They can be browsed via the Browser panel.
- **STAC API**: provide search capabilities and can be accessed in QGIS through both the Browser panel and the Data Source Manager.

### 19.5.1 연결 설정하기

STAC connections can be added in QGIS using either the *Browser panel* or the *Data Source Manager*:

- **Browser Panel:** In the *Browser panel*, right-click on the *STAC* entry and select *New STAC Connection…*. In the dialog that appears, enter a *Name* for the connection, the *URL* of the STAC catalog and optionally fill in *Authentication* credentials and a *Referer*. Then click *OK*.

  Use this method for browsing static STAC catalogs that do not support search or filtering.

- **Data Source Manager:** For STAC APIs you can use *Data Source Manager* dialog.

  Open the *Data Source Manager*, choose the *STAC* tab and click the *New* button. Fill in the *Name* and *URL* fields, and (optional) the *Authentication* credentials and a *Referer*. Press *OK* and then *Connect* to establish the connection, after that you will be able to:

  - *Edit* the STAC connection settings
  - *Remove* the STAC connection

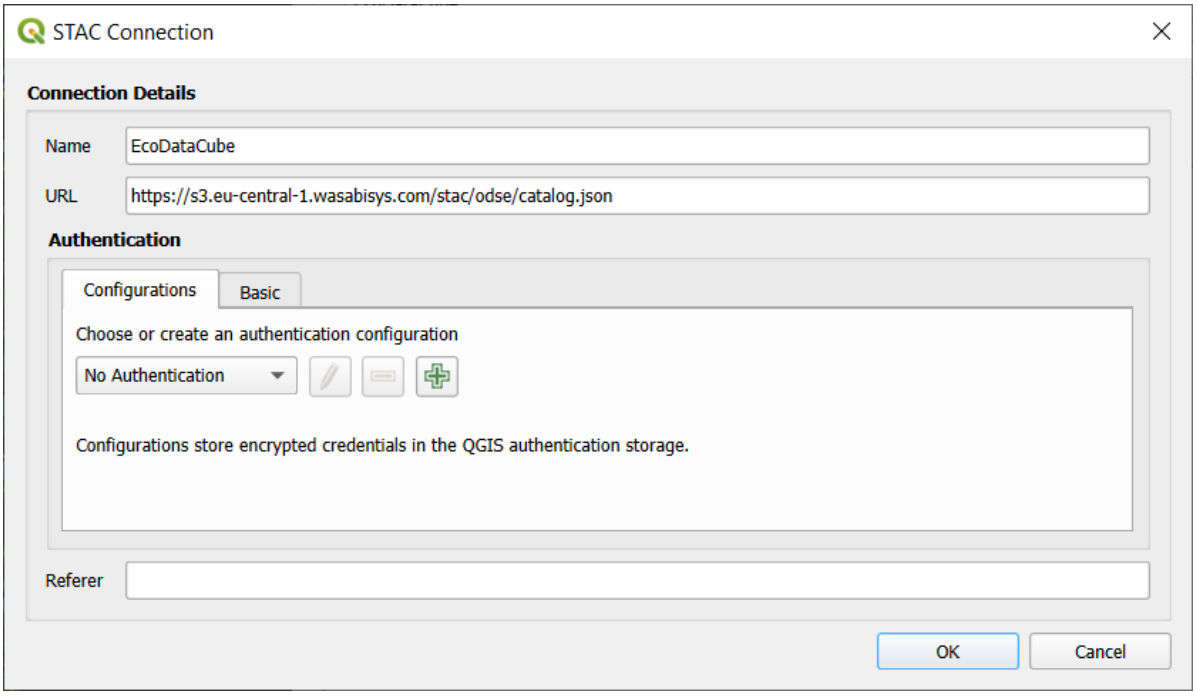



그림 19.12: Creating a connection to a STAC server

### 19.5.2 Browsing STAC Catalogs

Static STAC catalogs are displayed as hierarchical structures within the *Browser panel*. Once connected, the STAC catalog appears under *STAC* in the *Browser* panel. You can expand the catalog node to see its *Collections*. Expanding a collection reveals the individual *Items* it contains.

Right-click any STAC *Item* and choose *Details…* to view its metadata. The details panel shows the item's JSON content and a map of its coverage. If an item's asset is a cloud-optimized format (e.g. a `.COG`), you can add it directly to the map canvas. Otherwise, *Download Assets…* to save it locally before use.

### 19.5.3 Filtering and Searching STAC Items

For STAC API endpoints, QGIS supports spatial and temporal filtering, as well as limiting searches to specific collections.

To search for items:

1. Open *Data Source Manager* and go to the *STAC*
2. Select the STAC connection you created earlier and click *Filters…*
3. In the filter dialog, you can:
   - Enable *Spatial Extent* to restrict results to a bounding box or drawn area.
   - Enable *Temporal Extent* and define a date or range.
   - Limit to specific collections using *Only search within specific collections*
4. After applying filters, results will be listed in the main *Data Source Manager* panel.

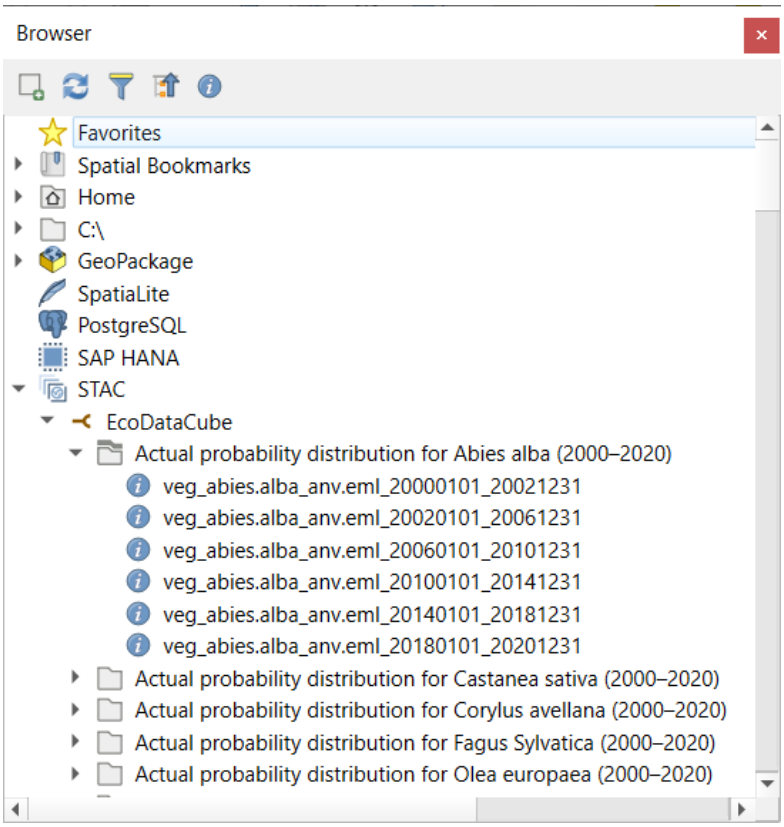



그림 19.13: STAC connection expanded in the Browser, showing Collections and Items

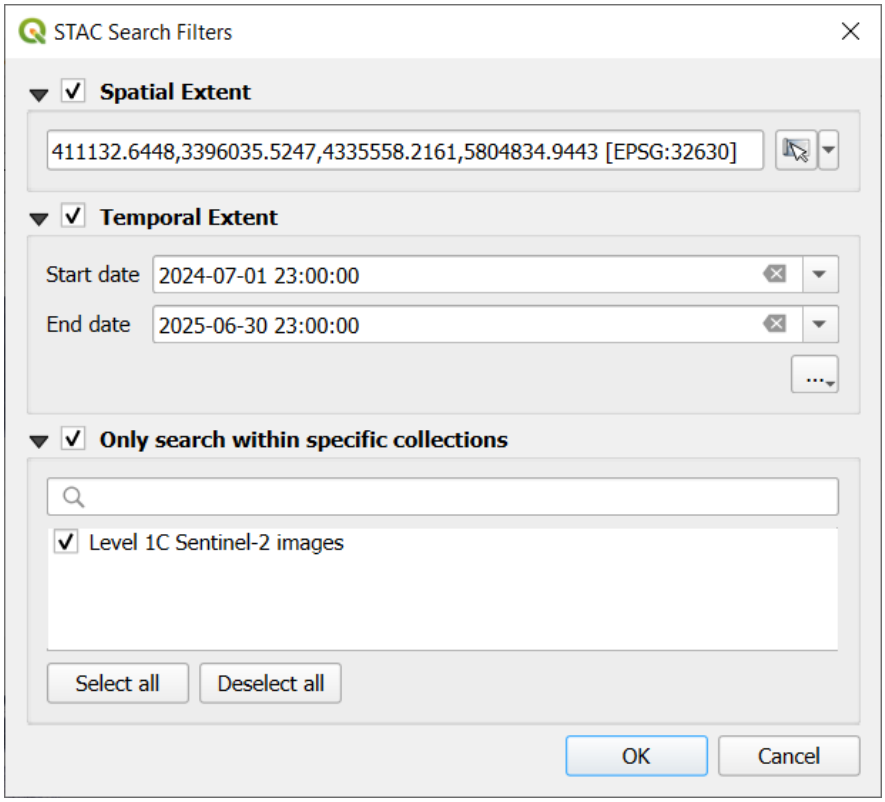



그림 19.14: Define spatial and temporal filters for a STAC catalog search

5. Optionally check *Show Footprints* to display the footprints of the results on the map.

Right-click a result item to access actions:

- *Zoom to Item*
- *Pan to Item*
- *Download Assets*
- *Details…*

If the item's asset requires download, use the *Download Assets* option.

Downloaded assets include the main dataset and any auxiliary files such as thumbnails or style files. After download, use standard QGIS tools (e.g., *Add Raster Layer…*) to load and display the data.

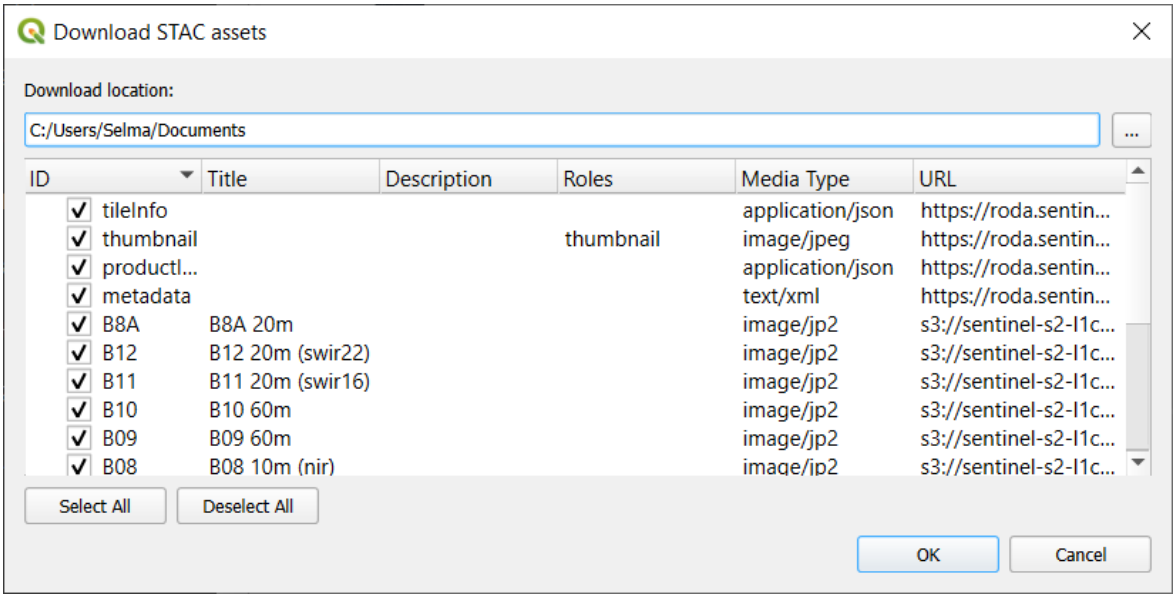



그림 19.15: Download STAC item assets

## 19.6 ArcGIS REST Server

An ArcGIS REST Server can host many different types of web services (feature service, map service, image service, ⋯ ). Read *ArcGIS REST 서버 이용하기* for instructions to connect to an ArcGIS REST server.

When loading layers form an ArcGIS Feature Service, symbols are automatically converted to QGIS-native symbology, allowing QGIS to visually reproduce the same appearance as ArcGIS web maps.

Layers from an ArcGIS Feature Service can be editable in QGIS if:

1. The ArcGIS service has editing enabled
2. Your credentials grant write access
3. And the remote layer supports editing operations (Create, Update, or Delete capabilities). If these options are disabled on the server, the layer will be read-only in QGIS.

Editing these layers follow the *same procedure* as for other vector layers. To update your map with the latest changes, use *View → Refresh* or press `F5`. If multiple users edit the same layer concurrently, it is recommended to refresh your data before performing edits to avoid conflicts.

CHAPTER 20

# GPS 데이터 작업

## 20.1 GNSS/GPS 데이터 소개

### 20.1.1 GPS 란?

**범지구 위치결정 시스템 (Global Positioning System)** 의 약어인 GPS 는 누구나 세계 어디에 있든 GPS 수신기를 가지고 있다면 자신의 정확한 위치를 찾을 수 있게 해주는 위성 기반 시스템입니다. GPS 는 예를 들어 비행기, 선박, 또는 도보 여행자의 항법 (navigation) 도우미로 사용됩니다. GPS 수신기는 위성이 발신하는 신호를 이용해서 자기 위치의 위도, 경도, 그리고 (때때로) 표고를 계산합니다. 수신기 대부분은 다음을 저장할 수 있는 기능도 가지고 있습니다:

- 위치 (또는 **웨이포인트 (waypoint)**)
- 계획된 **루트 (route)** 를 구성하는 일련의 위치들
- 시간에 걸쳐 수신기의 이동을 기록하는 **트랙 (track)** 로그

웨이포인트, 루트 및 트랙은 GPS 데이터의 세 가지 기본 피처 유형들입니다. QGIS 는 포인트 레이어에 웨이포인트를 표시하고, 라인스트링 레이어에 루트와 트랙을 표시합니다.

> **참고**
>
> QGIS 는 범지구 위성 항법 시스템 (Global Navigation Satellite System; GNSS) 수신기도 지원하고 있습니다. 그러나 이 문서에서는 계속 GPS 라는 용어를 사용할 것입니다.

## 20.1.2 GPS 데이터 전송 및 불러오기

### GPX 파일 불러오기

GPS 데이터를 저장하기 위한 수십 개의 서로 다른 파일 포맷이 있습니다. QGIS 가 이용하는 포맷은 GPX(GPS eXchange format) 라고 불리는데, 동일 파일 내에 웨이포인트, 루트 및 트랙을 얼마든지 담을 수 있는 표준 교환 (interchange) 포맷입니다.

GPX 파일을 불러오려면:

1. 다음과 같이 *Data Source Manager* 대화창에 있는 *GPS* 탭을 여십시오:
   - 툴바에 있는 Open Data Source Manager 버튼 (또는 Ctrl+L 키) 을 누른 다음 *GPS* 탭을 활성화합니다.
   - 또는 *Layer* → *Add Layer* → *Add GPX Layer*…메뉴를 선택합니다.
2. *GPX dataset* 옵션 옆에 있는 … Browse 버튼을 눌러 GPX 파일을 선택합니다.
3. *Feature types*: 체크 상자를 이용해서 파일에서 불러오고자 하는 피처 유형을 선택하십시오. 각 피처 유형을 (*Waypoints*, *Tracks* 또는 *Routes*) 개별 레이어로 불러올 것입니다.

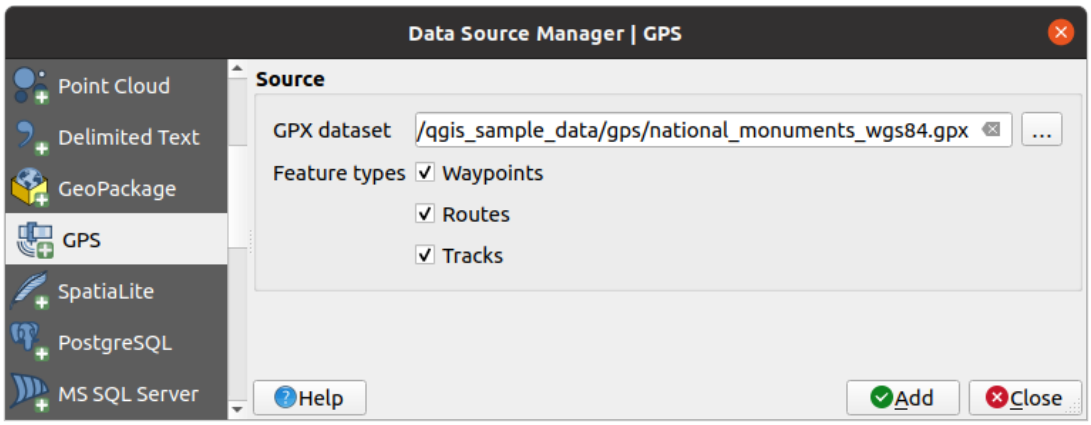



그림 20.1: GPS 데이터 불러오기 대화창

### 기기로 또는 기기에서 불러오기

서로 다른 GPS 기기 및 포맷 유형이 많습니다. QGIS 가 GPX 파일을 사용하기 때문에, 다른 GPS 파일 포맷들을 GPX 로 변환할 방법이 필요합니다. QGIS 는 자유 프로그램인 GPSBabel 을 사용합니다. GPSBabel 은 가민 (Garmin) 또는 마젤란 (Magellan) 같은 유명한 GPS 수신기와 구글 어스 또는 베이스캠프 (Basecamp) 같은 지도 작성 프로그램들 사이에서 웨이포인트, 트랙, 그리고 루트를 변환할 수 있습니다. 문자 그대로 수백 개의 GPS 수신기와 프로그램을 지원합니다. 또한 사용자 컴퓨터와 GPS 기기 사이에 GPS 데이터를 전송할 수도 있습니다.

QGIS 의 *Settings* → *Options* → *GPS* → *GPSBabel* 메뉴에서, 사용자는 자신의 기기 유형을 정의하고공간처리 *GPS* 알고리즘 이 이후 사용할 수 있는 변환 파라미터들을 설정할 수 있습니다.

1. *Path to GPSBabel*: 사용자는 먼저 GPSBabel 바이너리를 가리키는 경로를 정의해야 합니다.
2. 그 다음 사용자의 GPS 수신기를 추가할 수 있습니다. Add new device 또는 Remove device 버튼을 이용해서 기기 목록을 업데이트할 수 있습니다.
3. 각 기기에 대해:
   - *Device name*: 기기 이름을 입력합니다.

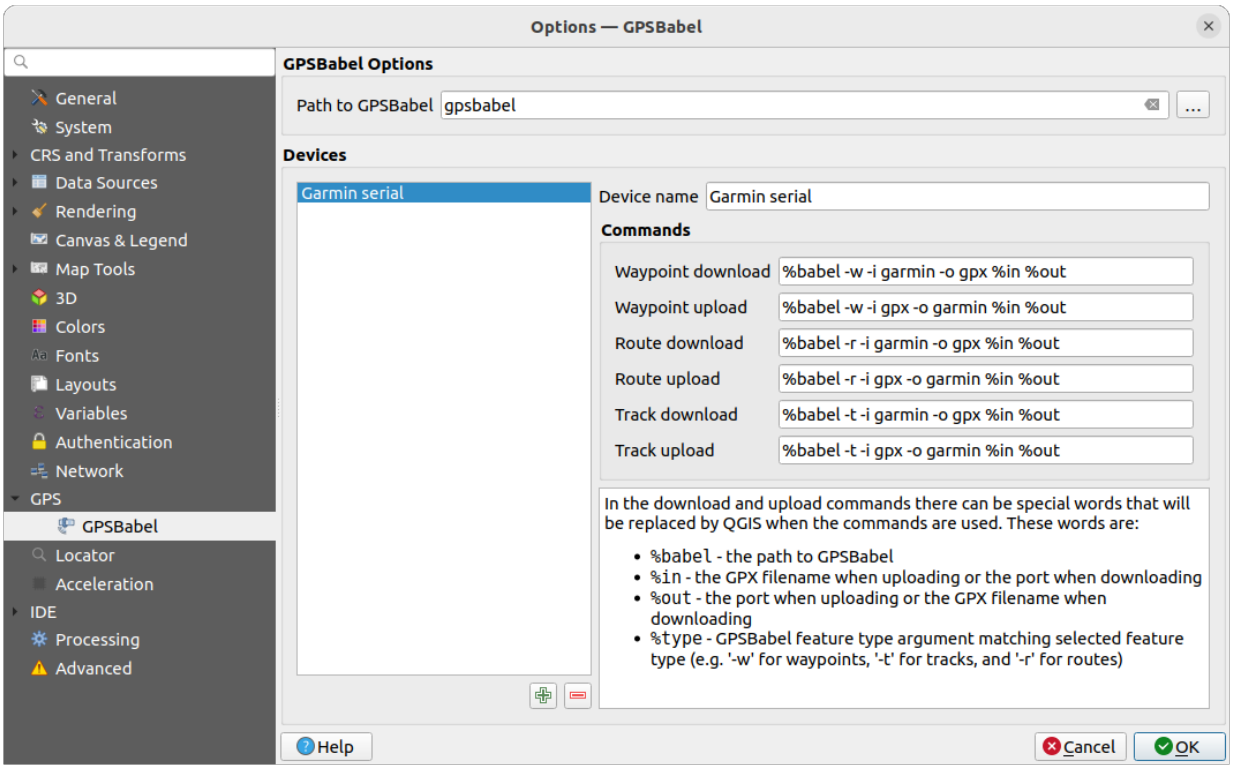



그림 20.2: GPSBabel 설정

- *Commands*: QGIS 가 해당 기기와 대화형 작업을 하는 동안 사용할 다양한 명령어들을 환경 설정합니다:
  - *Waypoint download*: 기기로부터 웨이포인트를 다운로드하는 명령어를 입력합니다.
  - *Waypoint upload*: 기기에 웨이포인트를 업로드하는 명령어를 입력합니다.
  - *Route download*: 기기로부터 루트를 다운로드하는 명령어를 입력합니다.
  - *Route upload*: 기기에 루트를 업로드하는 명령어를 입력합니다.
  - *Track download*: 기기로부터 트랙을 다운로드하는 명령어를 입력합니다.
  - *Track upload*: 기기에 트랙을 업로드하는 명령어를 입력합니다.

  이런 명령어들은 보통 GPSBabel 명령어이지만, GPX 파일을 생성할 수 있는 다른 어떤 명령 줄 프로그램이라도 사용할 수 있습니다. QGIS 는 이런 명령어를 실행할 때 `%type`, `%in`, 그리고 `%out` 키워드들을 대체할 것입니다.

  예를 들자면, 다운로드 명령어 `gpsbabel %type -i garmin -o gpx %in %out` 을 가진 기기 유형을 생성한 다음 `/dev/ttyS0` 포트로부터 웨이포인트를 다운로드해서 `output.gpx` 파일로 저장하는 경우 QGIS 가 키워드들을 대체해서 `gpsbabel -w -i garmin -o gpx /dev/ttyS0 output.gpx` 명령어를 실행할 것입니다.

  사용자의 활용 사례에 적합한 명령 줄 옵션을 알아보려면 GPSBabel 지침서를 읽어보십시오.

새 기기 유형을 생성하고 나면, GPS 다운로드 및 업로드 알고리즘에 대한 기기 목록에 추가될 것입니다.

> **참고**
>
> GPS 장치는 데이터를 서로 다른 좌표계로 저장할 수 있습니다. 사용자의 GPS 기기 또는 웹사이트에서 GPX

파일을 다운로드해서 QGIS 로 불러올 때, GPX 파일에 저장된 데이터가 WGS 84(위도/경도) 좌표계를 사용하고 있는지 확인해야 합니다. QGIS 가 이 좌표계를 기대하고 있고, 또 WGS 84 가 공식 GPX 사양이기도 합니다. GPX 1.1 스키마 문서 를 참조하세요.

## 20.2 실시간 GPS 추적

QGIS 는 사용자가 GPS 수신기를 가지고 현장 지도를 제작하는 일을 도울 수 있습니다. 이런 실시간 추적 작업은*GPS 툴바* 를 통해 이루어집니다. QGIS 와 GPS 수신기를 연결하기 전에기기 환경설정 을 해야 할 수도 있습니다.

### 20.2.1 GPS 툴바

*GPS Toolbar* 에는 실시간 추적 세션을 제어할 수 있는 주요 도구들이 있습니다. *View ▸ Toolbars ▸ GPS* 메뉴에서 이 툴바를 활성화시킬 수 있습니다. 이 툴바는 프로젝트의 상태, GPS, 그리고 현재 GPS 트랙을 따라가면서 합당한 경우에만 액션들을 활성화시킵니다. QGIS 가 기기를 탐지하고 나면, 해당 기기와 쌍방향 작업을 할 수 있을 것입니다:

- Connect to GPS: GPS 연결
- Disconnect from GPS: GPS 연결 해제
- Recenter map on GPS location: 자동 중심 정렬 파라미터 에 상관없이, 즉시 맵의 중심을 현재 GPS 위치로 이동시킬 것입니다.
- Set destination layer for GPS digitized features: QGIS 는 기본적으로 피처 디지타이즈 작업에 대해 *Follow active layer* 활성 레이어를 따를 것입니다. 다시 말해 GPS 디지타이즈 작업 도구가 *Layers* 패널에서 선택한 레이어에 맞춰 조정되며 그에 따라 해당 레이어에 생성한 피처를 저장할 것이라는 뜻입니다. 어떤 맥락에서는 이 습성이 유용할 수도 있는 반면, 원하지 않는 레이어에 피처를 실수로 저장하는 일을 피하려면 다른 레이어들과의 쌍방향 작업을 신중하게 해야 한다는 의미입니다. 이 옵션은 사용자가 실시간 추적 세션 동안 데이터를 저장하기 위한 레이어를 명확하게 지정할 수 있고, 필요에 따라 바꿀 수 있게 해줍니다. 보이는 도구들 또한 선택한 레이어 유형에 맞게 조정될 것입니다.

  GPS 대상 레이어는 피처 생성 시 자동으로 편집 가능 상태가 되며, 이를 사용자에게 알려줍니다.
- 피처 생성 관련 액션:
  - Add vertex to GPS track using GPS current location: GPS 트랙에 꼭짓점을 현재 GPS 위치를 사용해서 추가합니다.
  - Create a new point feature at current GPS location: 현재 GPS 위치에 새 포인트 피처를 생성합니다.
  - Create a new line feature using the current GPS track: 새 라인 피처를 현재 GPS 트랙을 사용해서 생성합니다.
  - Create a new polygon feature using the current GPS track: 새 폴리곤 피처를 현재 GPS 트랙을 사용해서 생성합니다.
  - Reset track: 트랙을 리셋합니다.
- Show GPS information: *GPS 정보* 패널을 엽니다.
- 사용자에게 필요할 수도 있는 몇몇 GPS 정보에 빠르게 접근하기 위한 표시 상자:
  - Show location: 위치 보이기

  - Show altitude (Geoid): 고도 보이기 (지오이드)
  - Show altitude (WGS 84 ellipsoid): 고도 보이기 (WGS 84 타원체)
  - Show ground speed: 지상 속도 보이기
  - Show bearing: 방위 보이기
  - Show total track length: 총 트랙 길이 보이기
  - Show distance from start of track: 트랙의 시작점으로부터의 거리 보이기
- Settings 버튼을 클릭하면 세션 도중 수정할 것이라고 예상되는 공통 설정을 담고 있는 드롭다운 메뉴를 엽니다:
  - *Show location marker*: 위치 마커 보이기
  - *Show bearing line*: 방위선 보이기
  - *Rotate map to match GPS direction*: 맵을 GPS 방향과 일치하도록 회전시키기
  - 맵 중심 재정렬을 제어하기 위한 옵션들:
    * *Always recenter map*: GPS 위치가 맵 캔버스의 중심으로부터 (맵 캔버스 범위의 비율로서의) 일정 거리만큼 오프셋되었을 때 항상 맵의 중심을 다시 정렬합니다.
    * *Recenter map when leaving extent*: GPS 위치가 맵 캔버스 범위를 벗어났을 때 맵의 중심을 다시 정렬합니다.
    * *Never recenter*: 맵의 중심을 다시 정렬하지 않습니다.
  - *Automatically add track vertices*: GPS 기기에서 새 위치를 수신받을 때마다 트랙 꼭짓점을 자동으로 추가합니다.
  - *Automatically save added features*: 피처가 GPS 위치로부터 생성되는 즉시 (일반적인 레이어 편집 버퍼를 건너뛰고) 대상 레이어에 커밋합니다.
  - *Time stamp destination*: 대상 레이어에 시간 수정 저장 용 필드를 조정하기 위한 타임스탬프를 기록합니다.
  - *Log to GeoPackage/SpatiaLite*…: 이 옵션을 체크하면, 사용자에게 기존 GeoPackage/SpatiaLite 파일을 선택하라는 또는 새 파일 이름을 입력하라는 메시지가 뜰 것입니다. 이 파일 안에 사전 정의 구조를 가진 `gps_points` 및 `gps_tracks` 테이블이 생성될 것입니다.

    수신되는 모든 GPS 메시지는 GPS 에서 나온 속도, 방위, 고도 및 정확도 정보와 함께 `gps_points` 레이어에 로그로 작성될 것입니다.

    GPS 를 연결 해제 했을 때 (또는 QGIS 를 종료했을 때) `gps_tracks` 테이블에 기록된 전체 GPS 트랙을 (트랙 길이, 시작 및 종료 시간 같은 몇몇 계산된 정보와 함께) 추가할 것입니다.
  - *Log NMEA sentences*…: 이 옵션을 체크하면, 텍스트 파일에 기기에서 나온 모든 원시 (raw) NMEA 문자열을 로그로 작성합니다.
  - *GPS settings*…: GPS 의전체 수준 옵션 대화창을 엽니다.

> **팁**
>
> **위치에 대한 실시간 상태 바 정보**

GPS 기기를 연결하고 사용자가 맵 캔버스 위로 커서를 이동시키면, 실시간 상태 바 메시지가 커서로부터 GPS 위치까지의 거리와 방위를 표시합니다. 이 표시는 프로젝트 수준 거리 및 방위 설정을 준수합니다. 터치 스크린 기기의 경우'길게 탭하기'이벤트로 실시간 상태 바 메시지를 촉발합니다.

### 20.2.2 GPS 정보 패널

QGIS 에서 실시간 GPS 추적을 완전하게 모니터링하려면, *GPS Information Panel* 을 (*View* → *Panels* → 메뉴 또는 `Ctrl+0` 키 조합으로) 활성화시켜야 할 수도 있습니다.

*GPS Information Panel* 의 우상단에 있는 *Connect* 버튼을 클릭하면 QGIS 와 플러그인된 GPS 수신기 사이에 연결을 시작하고, 그 후 *Disconnect* 로 바뀐 버튼을 클릭하면 연결을 해제합니다.

패널의 좌상단에서는 다음 버튼들을 사용할 수 있습니다:

- Position: GPS 위치와 센서에 대한 실시간 상세 정보를 보여줍니다.
- Signal: 연결된 위성의 GPS 신호 강도를 표시합니다.
- Settings: 세션 도중 수정해야 할 수도 있는실시간 추적 옵션 을 담고 있는 드롭다운 메뉴입니다.

#### 위치 및 추가 속성

GPS 가 위성으로부터 신호를 수신받고 있는 경우, Position 탭에서 사용자의 위치를 위도, 경도 그리고 고도로 — 추가 속성들과 함께 — 볼 수 있을 것입니다.

- *Latitude*: 위도
- *Longitude*: 경도
- *Altitude (Geoid)*: 평균 해수면 위 또는 아래 방향 고도/표고
- *Altitude (WGS 84 ellipsoid)*: WGS-84 지구 타원체 위 또는 아래 방향 고도/표고
- *Geoidal separation*: WGS-84 지구 타원체와 평균 해수면 (지오이드) 의 차이로, – 부호는 평균 해수면이 타원체 아래 있다는 의미입니다.
- *Time of fix*: 수정 시간
- *Speed*: 지상 속도
- *Direction*: 진북에서부터 여행 방향을 시계 방향으로 측정한 도 단위 방위
- *HDOP*: 수평 정밀도 희석 (Horizontal dilution of precision)
- *VDOP*: 수직 정밀도 희석 (Vertical dilution of precision)
- *PDOP*: 위치 정밀도 희석 (Dilution of precision)
- *H accuracy*: 미터 단위 수평 정확도
- *V accuracy*: 미터 단위 수직 정확도
- *3D accuracy*: 미터 단위 3 차원 제곱평균제곱근 (RMS; Root Mean Square)
- *Mode*: GPS 수신기의 2 차원/3 차원 모드 환경설정입니다. `automatic` 또는 `manual` 가운데 하나일 수 있습니다.
- *Dimensions*: 위치 수정 차원입니다. `2D`, `3D` 또는 `No fix` 가운데 하나일 수 있습니다.
- *Quality*: 위치정보 품질 지표

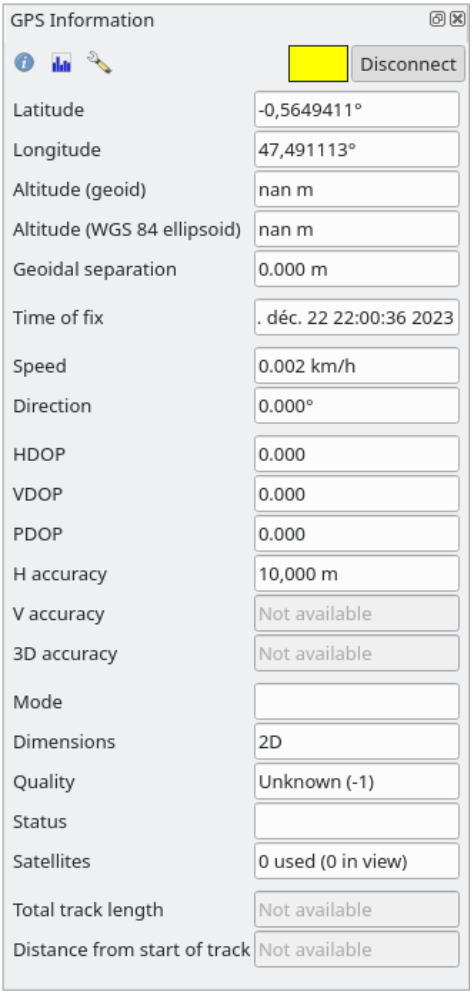



그림 20.3: GPS 위치 추적 및 추가 속성

- *Status*: 위치 수정 상태입니다. `Valid` 또는 `Invalid` 가운데 하나일 수 있습니다.
- *Satellites*: 수정 정보를 얻는 데 사용된 위성의 개수
- *Total track length*: 현재 GPS 트랙의 총 길이
- *Distance from start of track*: 현재 GPS 트랙의 첫 번째 꼭짓점에서 최신 꼭짓점까지의 직선 거리

#### 신호

 Signal 탭에서 신호를 수신받고 있는 위성의 신호 강도를 볼 수 있습니다.

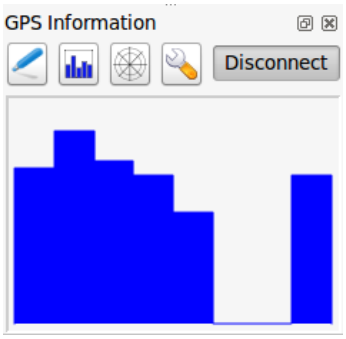



그림 20.4: GPS 추적 신호 강도

### 20.2.3 실시간 추적을 위해 블루투스 GPS 기기에 연결

현장 데이터 수집을 위해 QGIS 에 블루투스 GPS 기기를 연결할 수 있습니다. 이 작업을 하려면 블루투스 GPS 기기와 사용자 컴퓨터 상에 있는 블루투스 수신기가 필요합니다.

먼저 사용자 컴퓨터가 GPS 기기를 인식하고 페어링해야만 합니다. GPS 기기의 전원을 켠 다음, 사용자 컴퓨터의 알림 영역에 있는 블루투스 아이콘을 클릭하고 새 장치를 검색하십시오.

장치 선택 화면의 우측에서 모든 장치 유형을 선택해서 사용자의 GPS 기기가 연결할 수 있는 장치 목록에 나타나도록 하십시오. 다음 단계에서 시리얼 연결 서비스를 선택할 수 있을 겁니다. 해당 연결을 선택한 다음 *Configure* 버튼을 누르십시오.

블루투스 속성에서 GPS 연결에 할당된 COM 포트 번호를 확인해서 기억해두십시오.

GPS 기기를 인식한 후, 페어링해서 연결하십시오. 대부분의 경우 인증 코드는 `0000` 입니다.

이제 QGIS 에서 *GPS information* 패널을 열고 GPS 옵션 화면으로 전환하십시오. GPS 연결에 할당된 COM 포트 번호를 선택한 다음 *Connect* 버튼을 클릭하십시오. 잠시 후 사용자의 위치를 나타내는 커서가 나타날 것입니다.

QGIS 가 GPS 데이터를 받지 못 하는 경우, 사용자 GPS 기기를 재시작하고 5 ~ 10 초 동안 기다린 다음 다시 연결을 시도해야 합니다. 대부분의 경우 이렇게 하면 해결됩니다. 다시 연결 오류가 발생한다면 사용자 GPS 기기와 연결된 다른 블루투스 수신기가 근처에 있는지 확인해보십시오.

### 20.2.4 QGIS 에서의 GPS 기기 연결 예시

#### GPSMAP 60cs 사용법

##### MS 윈도우

이 GPS 기기를 가장 쉽게 사용하는 방법은 GPSGate 라는 (공개 소프트웨어는 아니지만 자유 소프트웨어인) 미들웨어를 사용하는 것입니다.

프로그램을 구동하고 GPS 기기를 스캔한 다음 (USB 와 블루투스 연결 둘 다 스캔합니다) QGIS 로 가서:

1. *Settings* ▸ *Options* ▸ *GPS* 메뉴 항목을 선택해서 플러그인된 기기를 탐지하십시오. *Autodetect* 를 선택하면 자동 탐지 모드를 사용할 수 있습니다.
2. *GPS Information* 패널에서 *Connect* 버튼을 클릭하십시오.

##### 우분투/민트 GNU/리눅스

윈도우와 마찬가지로 미들웨어를 사용하는 것이 가장 쉬운 방법입니다. 이 경우에는 `gpsd` 를 다음과 같이 설치하십시오:

1. 프로그램을 설치합니다.

```
sudo apt install gpsd
```

2. 그리고 `garmin_gps` 커널 모듈을 다음과 같이 불러오십시오:

```
sudo modprobe garmin_gps
```

3. 그 다음 기기를 연결하고, `dmesg` 명령어로 실제 기기가 연결됐는지, 예를 들어 `/dev/ttyUSB0` 처럼 마운트됐는지 확인하십시오.
4. 이제 다음과 같이 `gpsd` 를 구동할 준비가 끝났습니다:

```
gpsd /dev/ttyUSB0
```

5. 마지막으로 QGIS 의 실시간 GPS 추적 도구와 연결하십시오.

#### BTGP-38KM DataLogger (블루투스 전용)

`gpsd` (리눅스) 또는 GPSGate (윈도우) 프로그램을 사용하는 것이 가장 쉬운 방법입니다.

#### BlueMax GPS-4044 DataLogger (블루투스 및 USB 둘 다)

##### MS 윈도우

GPSGate 를 사용하든 사용하지 않든, *Autodetect* 모드를 선택하거나 또는 올바른 포트를 지정하는 것만으로도 USB 및 블루투스 연결 둘 다 실시간 추적이 가능합니다.

### 우분투/민트 GNU/리눅스

#### USB 연결

다음과 같이 `gpsd` 를 통해 실시간 추적을 할 수도 있고:

```
gpsd /dev/ttyACM3
```

또는 `gpsd` 를 사용하지 않고 QGIS 실시간 GPS 추적 도구에 기기를 직접 (예: `/dev/ttyACM3`) 연결해서 실시간 추적을 할 수도 있습니다.

#### 블루투스 연결

다음과 같이 `gpsd` 를 통해 실시간 추적을 할 수도 있고:

```
gpsd /dev/rfcomm0
```

또는 `gpsd` 를 사용하지 않고 QGIS 실시간 GPS 추적 도구에 기기를 직접 (예: `/dev/rfcomm0`) 연결해서 실시간 추적을 할 수도 있습니다.

CHAPTER 21

# 인증 시스템

## 21.1 인증 시스템 개요

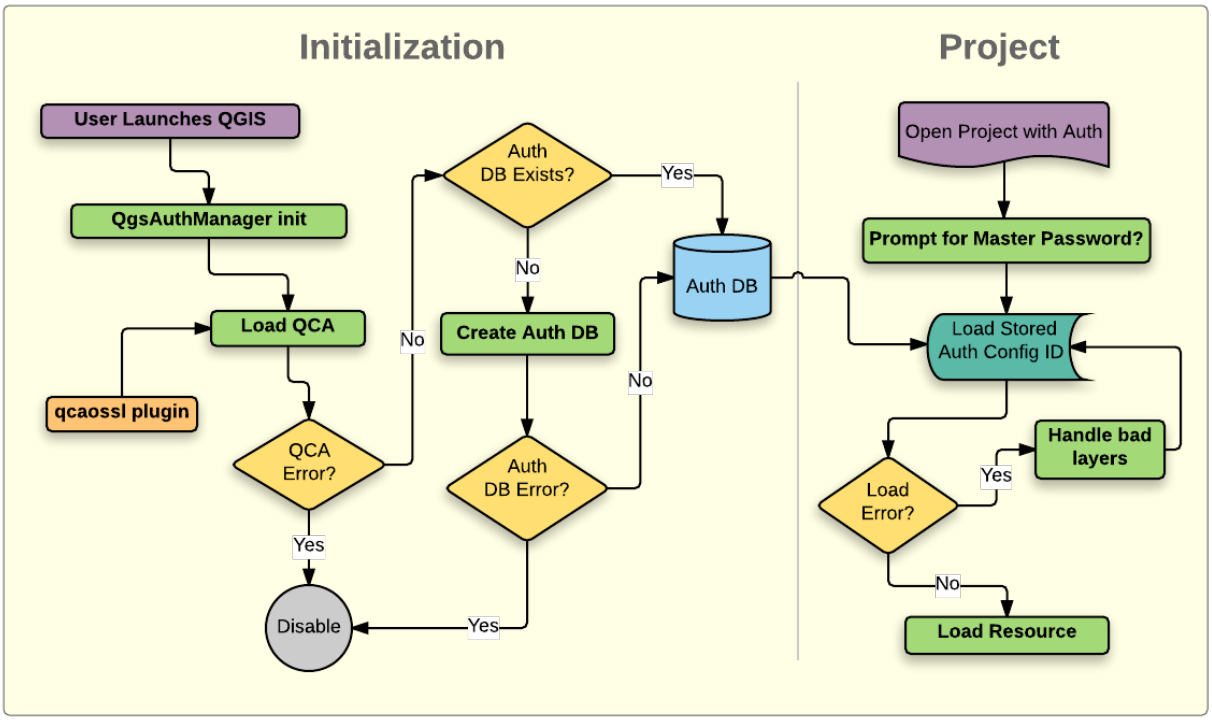



그림 21.1: 인증 시스템 개념도

### 21.1.1 인증 데이터베이스

인증 시스템은 기본적으로 `<profile directory>/qgis-auth.db` 위치에 있는 SQLite 데이터베이스 파일에 인증 환경 설정을 저장합니다.

현재의 다른 QGIS 사용자 환경 설정에 영향을 미치지 않고 서로 다른 QGIS 설치본 간에 이 인증 데이터베이스를 이동시킬 수 있습니다. 인증 데이터베이스가 일반적인 QGIS 설정과 완벽하게 분리되어 있기 때문입니다. 환경 설정을 데이터베이스에 처음으로 저장할 때 환경 설정 ID(랜덤한 문자 7 개 길이의 알파벳-숫자 문자열) 를 생성합니다. 이 ID 가 환경 설정을 가리키기 때문에, 관련 인증 정보를 노출시키는 일 없이 ID 를 (프로젝트, 플러그인, 또는 설정 파일과 같은) 평문 텍스트 응용 프로그램 구성 요소로 저장할 수 있습니다.

> **참고**
>
> 환경 변수 `QGIS_AUTH_DB_DIR_PATH` 를 사용하거나 또는 QGIS 구동 시 명령 줄에서 `--authdbdirectory` 옵션을 설정해서 `qgis-auth.db` 의 상위 (부모) 디렉터리를 설정할 수 있습니다.

### 21.1.2 사용자 정의 인증 데이터베이스

QGIS 가 앞에서 이야기한 기본 SQLite 데이터베이스 대신 사용자 정의 인증 데이터베이스를 사용하도록 환경설정할 수 있습니다. Qt SQL 모듈이 지원하는 모든 데이터베이스 (예: PostgreSQL, MySQL 등등) 를 사용할 수 있습니다. 물론 시스템 상에 해당 데이터베이스의 Qt SQL 드라이버가 설치되어 있어야 하겠죠.

사용자가 여러 QGIS 설치본들 사이에서 동일한 인증 데이터베이스를 공유하길 원한다거나, 기본 SQLite 데이터베이스가 아닌 다른 인증 데이터베이스를 사용하길 원한다거나, 또는 QGIS 서버가 중앙 집중형 (centralized) 인증 데이터베이스를 사용하고 있는 경우 이 기능이 유용할 수 있습니다.

사용자 정의 인증 데이터베이스를 환경설정할 수 있는 유일한 방법은 `driver://username:password@hostname:port/database?options` 형식의 연결 URI 에 환경 변수 `QGIS_AUTH_DB_URI` 를 설정하는 것입니다.

**이때:**

- `driver`: 사용할 Qt SQL 드라이버의 이름, 예: PostgreSQL 은 `QPSQL`, MySQL 은 `QMYSQL` 등등
- `username`: 데이터베이스에 연결하기 위해 쓰이는 사용자명
- `password`: 데이터베이스에 연결하기 위해 쓰이는 비밀번호
- `hostname`: 데이터베이스 서버의 호스트명
- `port`: 데이터베이스 서버의 포트 번호
- `database`: 사용할 데이터베이스의 이름
- `options`: 드라이버로 전송할 옵션들, 예: PostgreSQL 의 경우 `sslmode=require` 등등

> **참고**
>
> URI 옵션에서 `schema` 를 지정할 수 있습니다. 예: `QPSQL://username:password@hostname:port/database?schema=schema_name`
>
> QGIS 를 시작하기 전에 데이터베이스가 존재하고 있어야만 하며, 사용자는 데이터베이스에 연결하기 위해 필요한 권한과 필수 테이블이 존재하지 않을 경우 생성할 수 있는 권한을 가지고 있어야만 합니다.

> **경고**
>
> URI 에서 비밀번호는 환경 변수 안에 평문으로 저장되기 때문에, 비밀번호가 없는 사용자명 또는 데이터베이스에 제한적인 권한으로 연결되는 사용자명을 사용할 것을 권장합니다.

> **경고**
>
> SQLite 을 기반으로 하지 않는 모든 데이터베이스는 읽기 전용으로 간주될 것입니다. (필요한 경우 파이썬 플러그인을 통해 변경할 수 있습니다.)

이 고급 기능은 QGIS 가 하나 이상의 사용자 정의 인증 데이터베이스를 사용할 수 있도록, 또는 QGIS 가 사용할 자격 증명 저장소를 사용자가 직접 파이썬으로 구현할 수 있도록 설계되었습니다.

이 시스템은 여러 개의 인증 데이터베이스를 사용할 수 있도록 설계되었지만, 지금으로써는 여러 자격 증명 저장소를 관리할 수 있는 사용자향 인터페이스가 존재하지 않습니다. 즉 인증 데이터베이스를 여러 개 사용하고 싶다면, 대부분의 경우 파이썬 플러그인을 통해 수동으로 환경설정하고 관리해야 합니다.

### 21.1.3 마스터 비밀번호

데이터베이스 내에 민감한 정보를 저장하거나 접근하려면, 사용자가 마스터 비밀번호 를 정의해야만 합니다. 데이터베이스에 암호화된 데이터를 처음으로 저장할 때 새 마스터 비밀번호를 요청하고 확인합니다. 사용자가 민감한 정보에 접근할 때 마스터 비밀번호를 입력해야 합니다. 사용자가 캐시값을 수동으로 삭제하지만 않는다면 남은 세션 동안 (응용 프로그램을 종료할 때까지) 마스터 비밀번호를 캐시에 저장합니다. 기존 인증 환경 설정을 선택한다거나 또는 (WMS 레이어 추가 시처럼) 서버 환경 설정에 인증 환경 설정을 적용한다거나 하는 경우와 같은 인증 시스템을 이용하는 일부 인스턴스는 마스터 비밀번호를 요구하지 않습니다.

사용자 컴퓨터의 `Wallet/Keyring` 에 비밀번호를 저장하도록 선택할 수 있습니다.

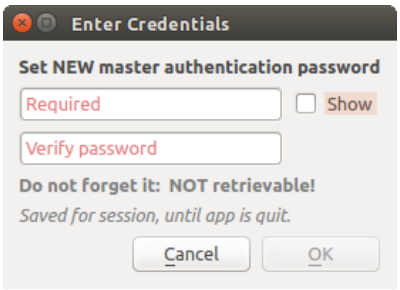



그림 21.2: 새 마스터 비밀번호 입력 대화창

> **참고**
>
> 환경 변수 `QGIS_AUTH_PASSWORD_FILE` 을 이용해서 마스터 비밀번호를 담고 있는 파일을 가리키는 경로를 설정할 수 있습니다.

#### 마스터 비밀번호 관리하기

마스터 비밀번호를 설정한 다음, 다시 리셋할 수 있습니다. 리셋하기 전에 현재 마스터 비밀번호를 입력해야 합니다. 이 과정 도중에 현재 데이터베이스의 완전한 백업 파일을 생성하는 옵션이 있습니다.




그림 21.3: 마스터 비밀번호 리셋하기

사용자가 마스터 비밀번호를 잊어버린 경우, 다시 찾거나 무시할 수 있는 방법은 없습니다. 또 마스터 비밀번호 없이 암호화된 정보를 받아올 수 있는 방법도 없습니다.

사용자가 기존 비밀번호를 세 번 부정확하게 입력한 경우, 대화창이 데이터베이스를 삭제하도록 제안할 것입니다.

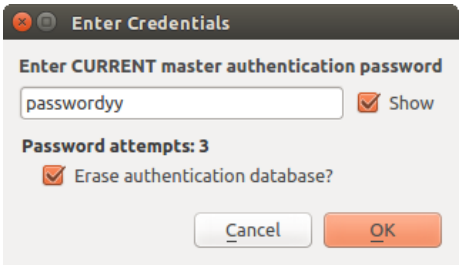



그림 21.4: 부정확한 시도를 세 번 한 후의 QGIS 비밀번호 메시지

### 21.1.4 인증 환경 설정

QGIS 옵션 대화창의 (*Settings* ▯ *Options*) *Authentication* 탭에 있는 **Configurations** 부분에서 인증 환경 설정을 관리할 수 있습니다.

To create a new authentication configuration:

1. Press the Add a new authentication configuration
2. Provide a *Name* and optional *Resource* URL
3. A seven-characters *ID* will be assigned to the configuration for identification purpose. It can be customized after you pressed the Unlock to edit ID button. Be aware that updating that ID may break things.
4. Select the *target method* and fill the corresponding connection details
5. Once done, click *Save* to save the configuration. The saved configuration appears in the list of authentication configurations and can be used with supported connections.

The saved configuration can later be edited using Edit selected configuration or removed using Delete selected configuration.

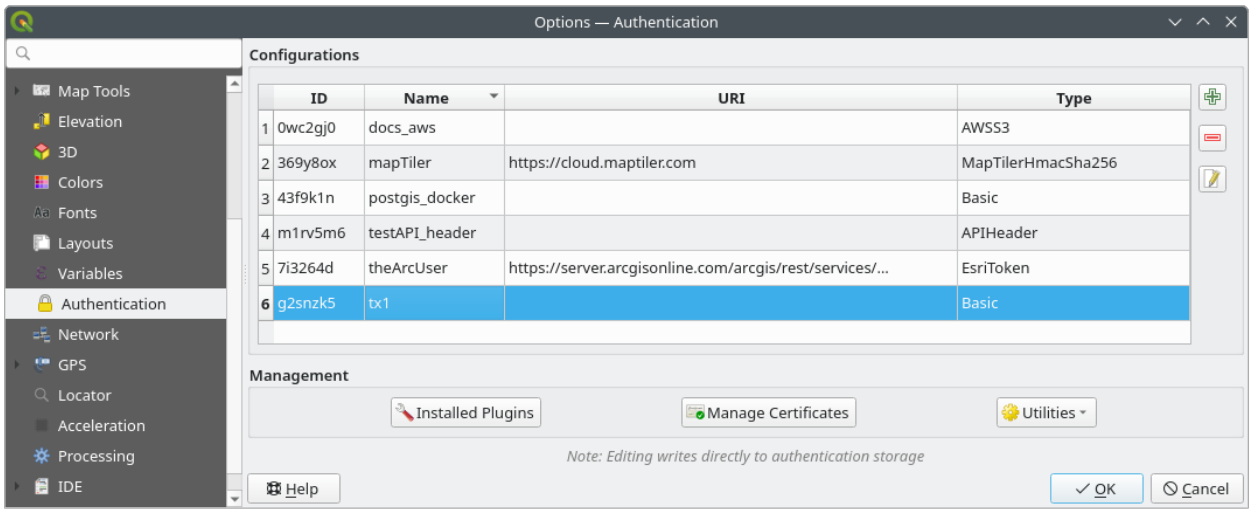



그림 21.5: 환경 설정 편집기

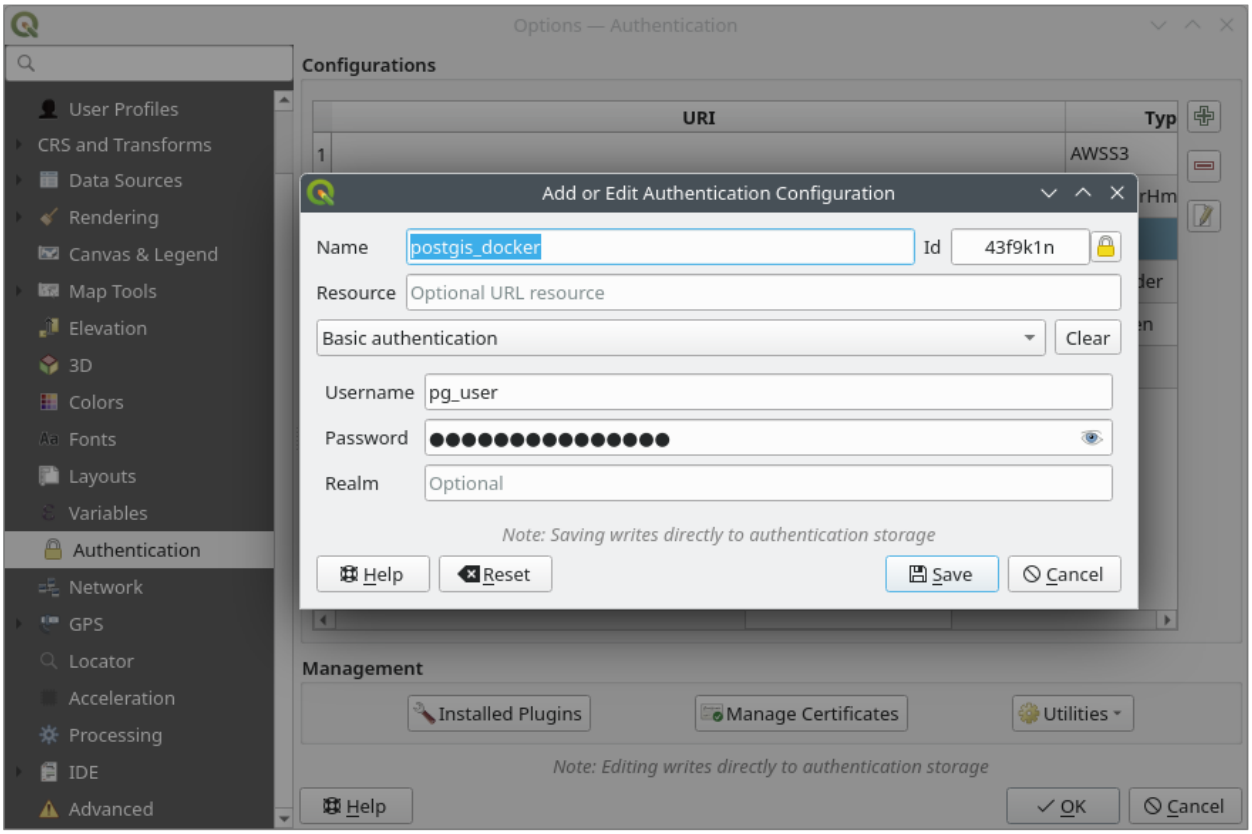



그림 21.6: 환경 설정 편집기 안에서 환경 설정 추가하기

The same type of operations for authentication configuration management (Add, Edit and Remove) can be done when configuring a given service connection, such as configuring an OWS service connection. For that, there are action buttons within the *configuration selector* for fully managing configurations found within the authentication database. In this case, there is no need to go to the *configurations* in *Authentication* tab of QGIS options unless you need to do more comprehensive configuration management.

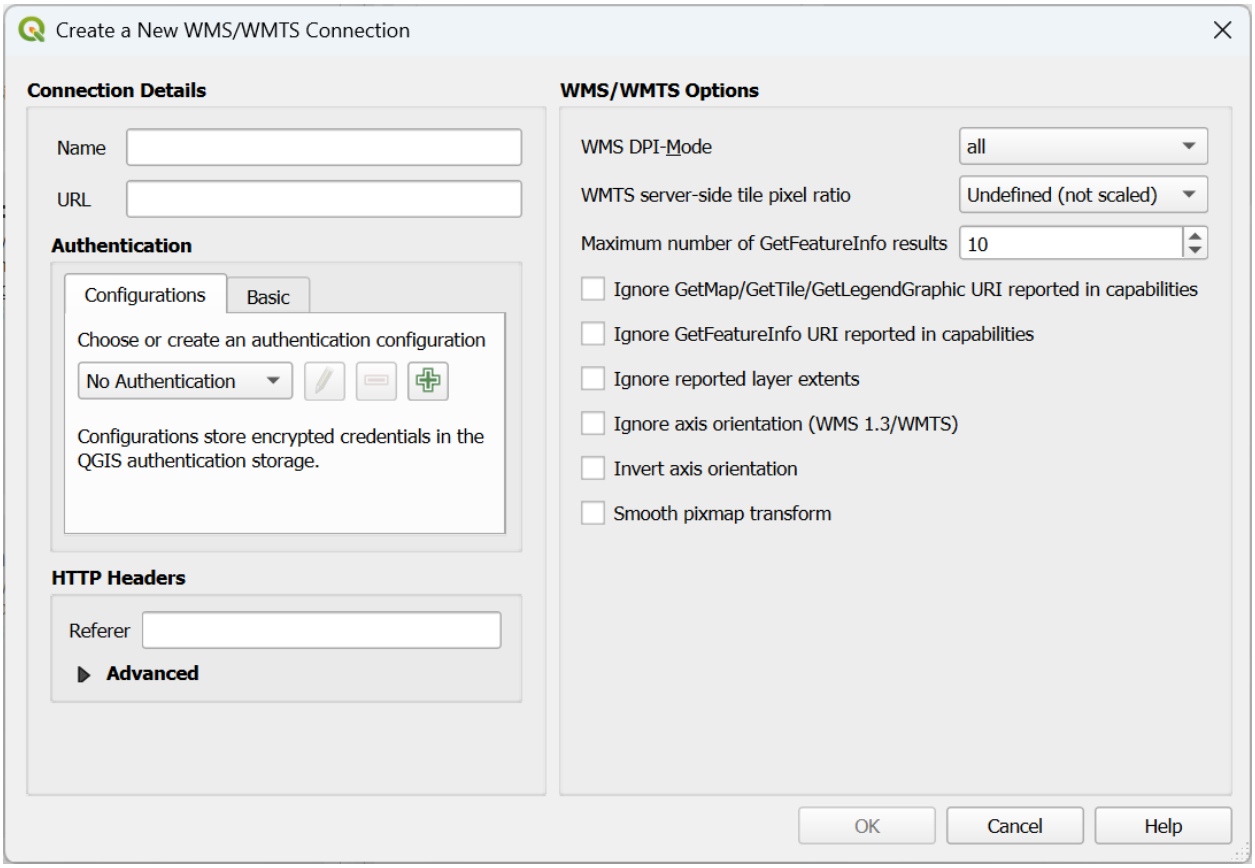



그림 21.7: WMS connection dialog showing authentication configuration widget

인증 환경 설정을 생성하거나 편집할 때, 요구되는 정보는 이름, 인증 방법, 그리고 인증 방법이 요구하는 기타 정보입니다. (사용할 수 있는 인증 유형에 관한 자세한 내용은인증 방법 을 참조하세요.)

### 21.1.5 인증 방법

QGIS 가 데이터 제공자 플러그인을 지원하는것과 거의 동일한 방식으로 C++ 플러그인이 사용 가능한 인증을 제공합니다. QGIS 코드와 플러그인을 둘 다 지원하는지 여부와 HTTP(S), 데이터베이스 같은 리소스/제공자에 필요한 접근 권한에 따라 인증 방법을 선택할 수 있습니다. 따라서 일부 인증 방법 플러그인은 인증 환경 설정 선택기가 표시된 모든 위치에 적용이 안 될 수도 있습니다. *Settings* ▯ *Options* 메뉴를 선택한 다음 *Authentication* 탭에 있는 *Installed Plugins* 버튼을 클릭하면 사용 가능한 인증 방법 플러그인과 호환 가능한 리소스/제공자 목록을 볼 수 있습니다.

QGIS 를 다시 컴파일할 필요가 없는 경우 새 인증 방법을 위한 플러그인을 생성할 수 있습니다. 현재 C++ 플러그인만을 지원하고 있기 때문에, 새로 추가한 플러그인을 사용하려면 QGIS 를 재시작해야 합니다. 기존 QGIS 설치본에 플러그인을 추가할 생각이라면, 사용자 플러그인이 기존 설치본과 동일한 버전을 대상으로 컴파일됐는지 확인하십시오.

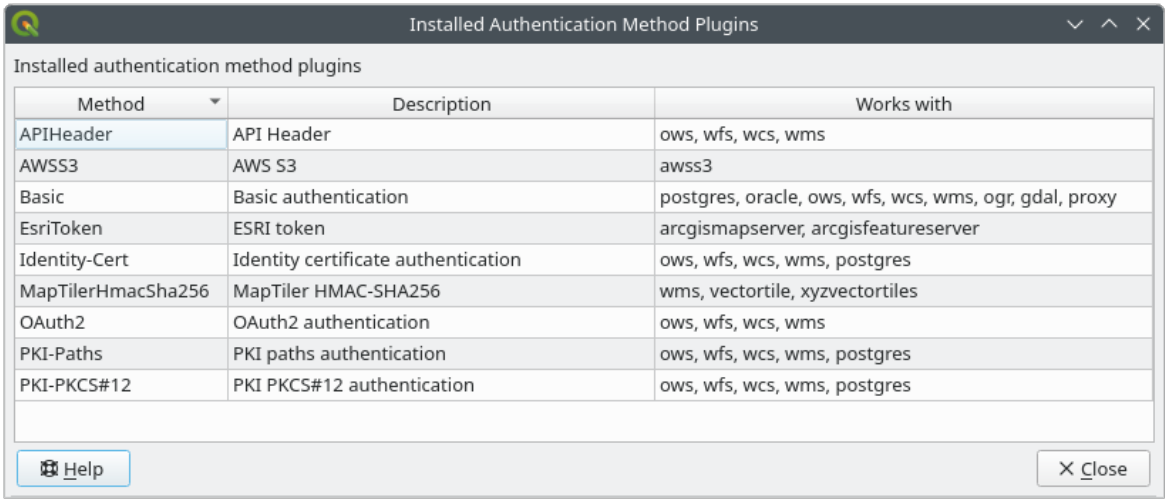



그림 21.8: 사용할 수 있는 인증 방법 플러그인 목록

### API Header Authentication

The API Header authentication method allows you to connect to services that require custom HTTP headers, such as API keys. For example, if a WMS service requires an API key sent as a header (e.g. `X-API-KEY`), you can use this method to provide it.

You can add one or more headers with the button; provide a *Header Name* and *Header Value* for each entry. Use to delete the selected header or the *Clear* button to remove all entries.

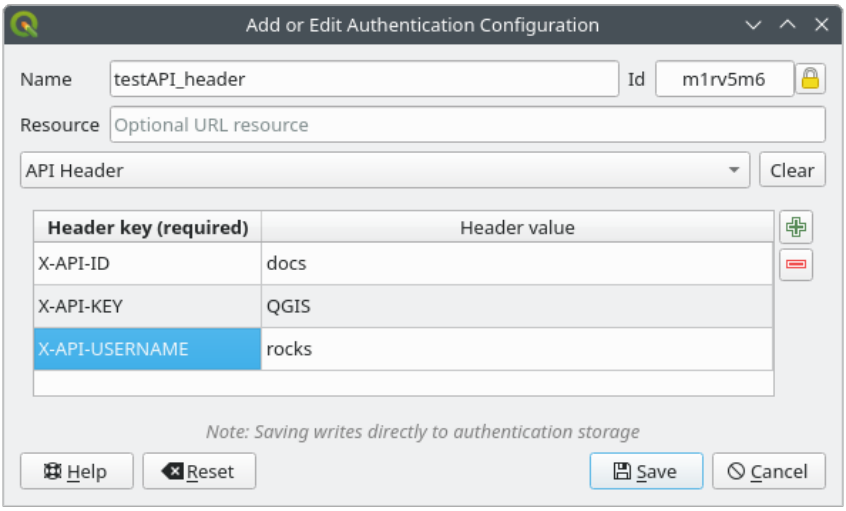



그림 21.9: API Header authentication configs

### AWS S3 Authentication

The AWS S3 authentication method allows you to connect to Amazon S3 resources by providing a *Name*, *Resource*, *Username*, *Password*, and *Region*.

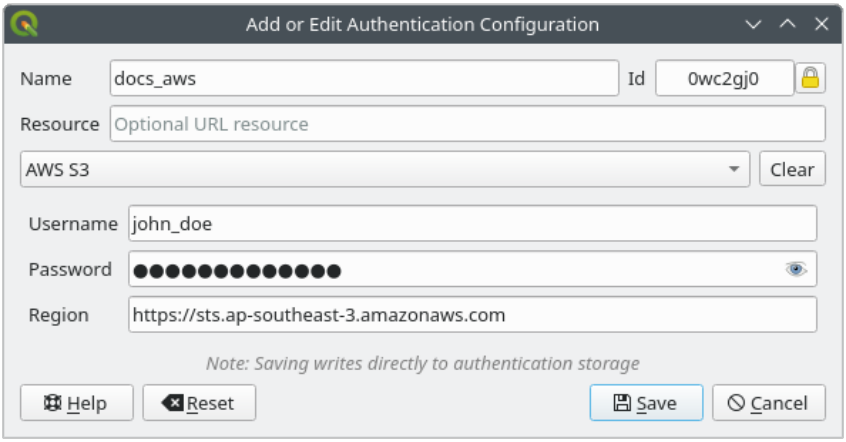



그림 21.10: AWS S3 authentication configs

### Basic Authentication

The Basic authentication method is used for services that require standard HTTP authentication with a *Username*, *Password* and *Realm*.

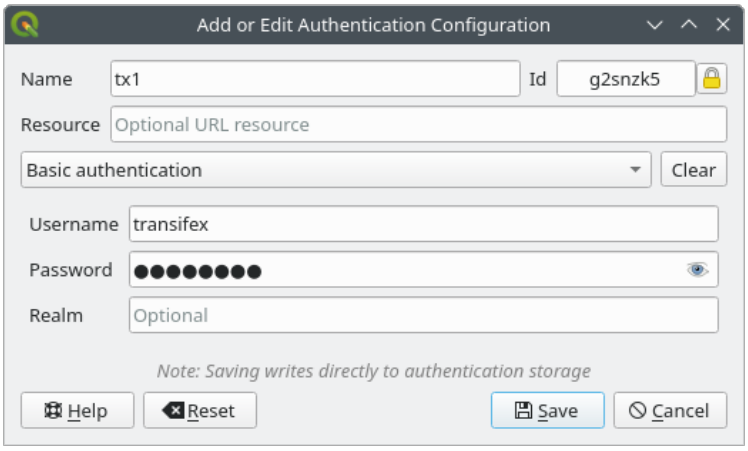



그림 21.11: HTTP 기본 인증 환경 설정

### ESRI Token Authentication

The ESRI token authentication method is used for *ArcGIS REST Servers* that require token-based authentication.

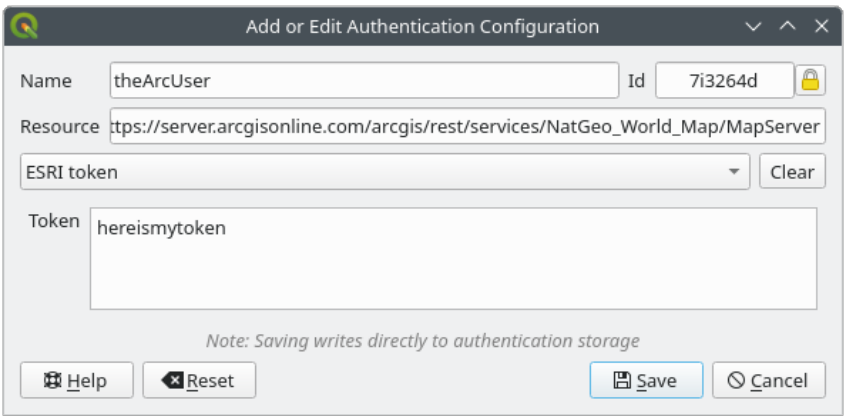



그림 21.12: ESRI 토큰 인증 환경 설정

### Identity certificate authentication

The Identity certificate authentication method allows you to connect using a client *identity certificate*.

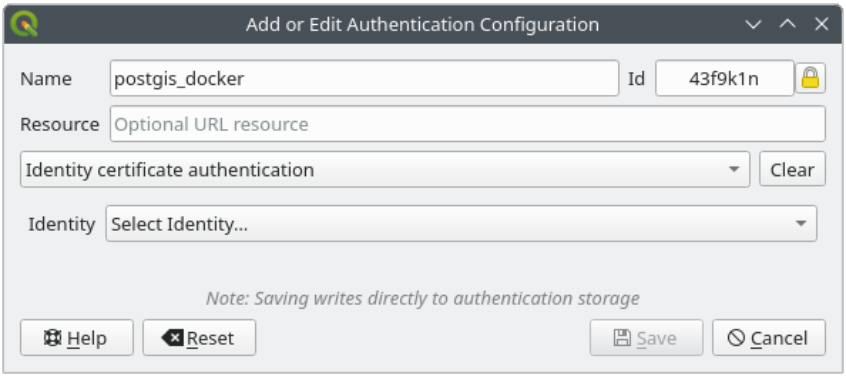



그림 21.13: Identity-cert authentication configs

### MapTiler HMAC-SHA256 Authentication

The MapTiler HMAC-SHA256 authentication method is used to connect to MapTiler services that require HMAC-SHA256 authorization.

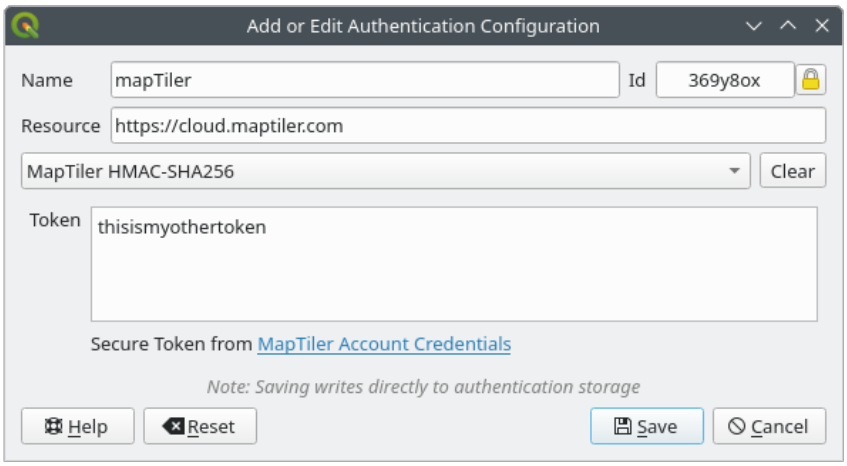



그림 21.14: MapTiler HmacSha256 authentication configs

### OAuth2 Authentication

The OAuth2 authentication method is used to connect to services that require OAuth2 2.0 authorization, allowing secure access using client credential and token-based authentication.

It supports using `localhost` as the redirect host, providing compatibility with services that do not accept `http://127.0.0.1` redirects (for example, Microsoft SharePoint).

An advanced OAuth2 configuration option, *Extra token header(s)*, allows attaching additional tokens returned by the OAuth2 token endpoint as HTTP(S) request headers.

### PKI-Paths Authentication

The PKI-Paths authentication method allows you to connect using separate certificate and key files stored on your system.

### PKI-PKCS#12 Authentication

The PKI-PKCS#12 authentication method allows you to connect using a single bundle file containing both the certificate and private key, with an optional passphrase for the key.

## 21.1.6 마스터 비밀번호 및 인증 환경 설정 유틸리티

*Authentication* 탭의 옵션 메뉴에서 (*Settings* → *Options*) 인증 데이터베이스 및 환경 설정을 관리할 수 있는 유틸리티 액션이 몇 개 있습니다:

- *Input master password*…: opens the master password input dialog, independent of performing any authentication database command
- *Clear cached master password*: unsets the master password if it has been set
- *Reset master password*…: opens a dialog to change the master password (the current password must be known) and optionally back up the current database
- *Clear network authentication access cache*: clears the authentication cache of all connections

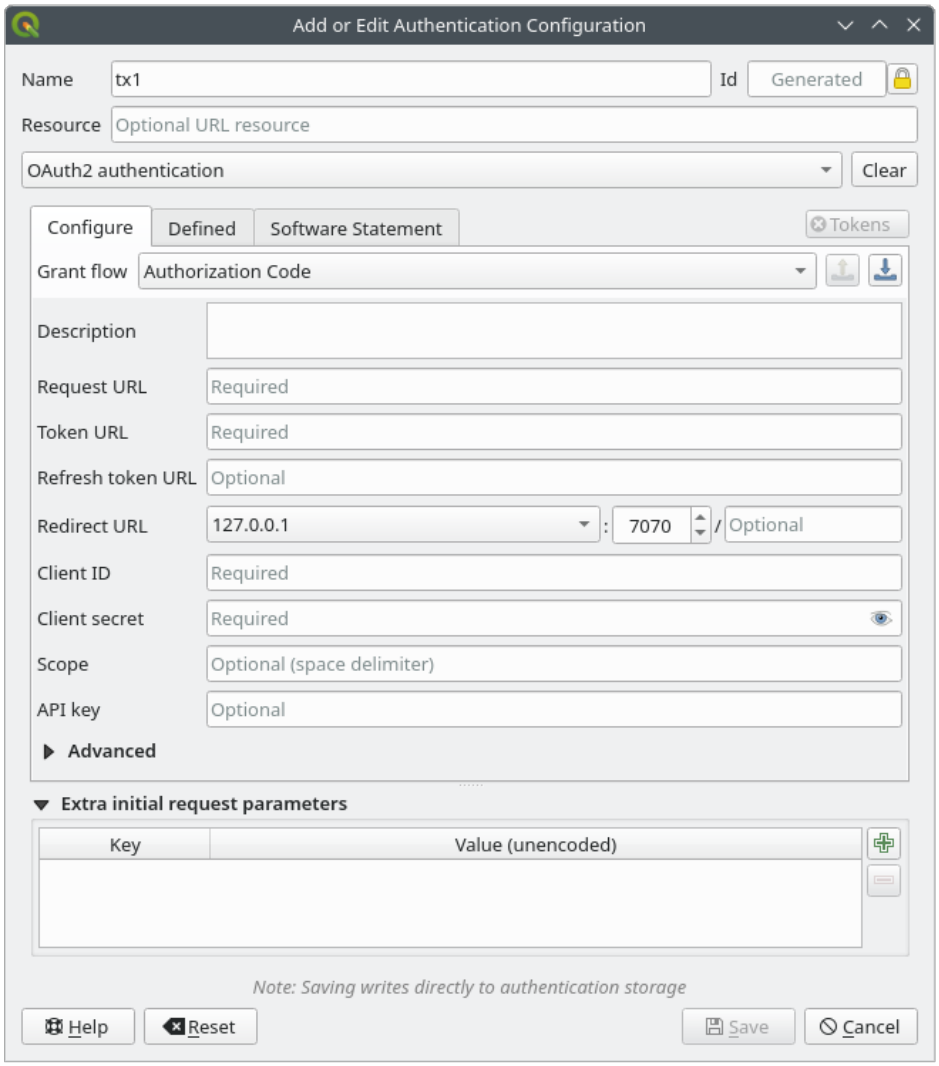



그림 21.15: OAuth2 인증 환경 설정

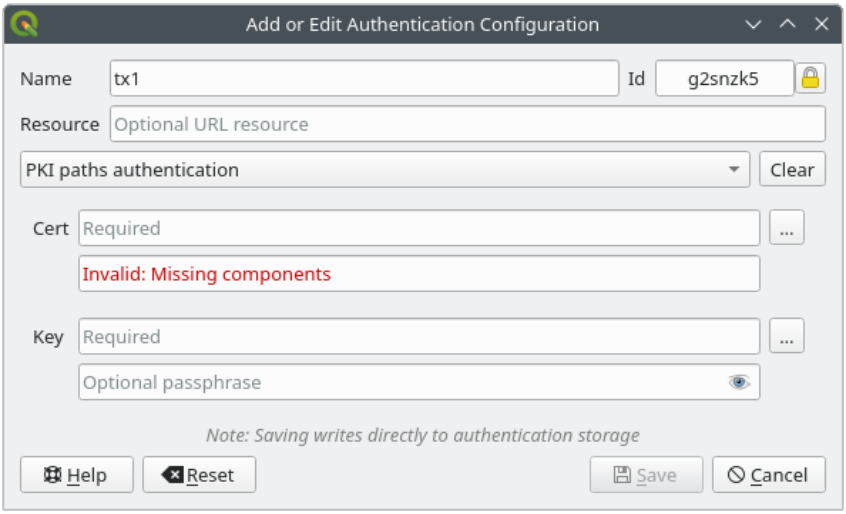



그림 21.16: PKI 경로 인증 환경 설정

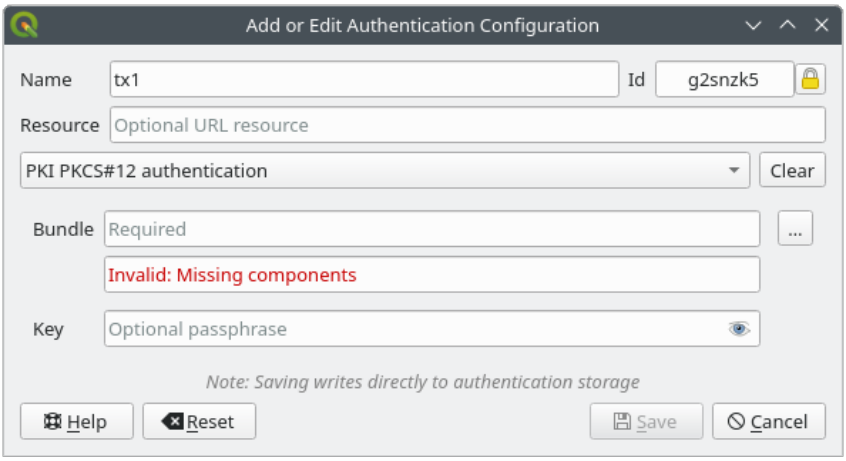



그림 21.17: PKI PKCS#12 파일 경로 인증 환경 설정

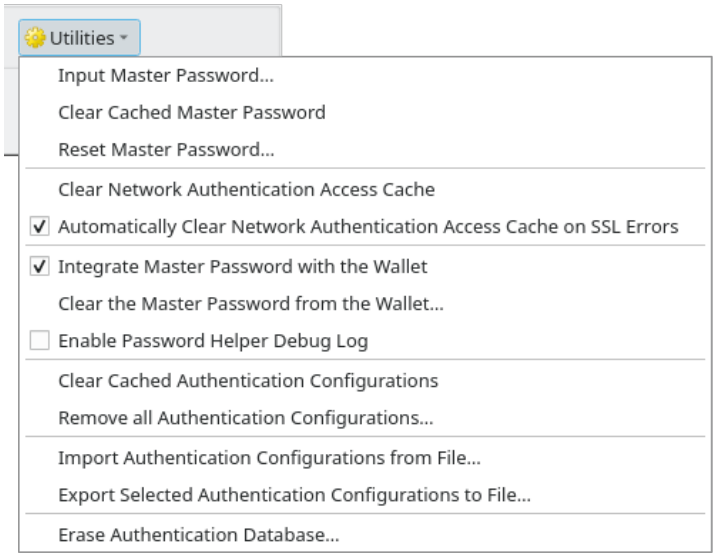



그림 21.18: 유틸리티 목록

- *Automatically clear network authentication access cache on SSL errors*: the connection cache stores all authentication data for connections, also when the connection fails. If you change authentication configurations or certification authorities, you should clear the authentication cache or restart QGIS. When this option is checked, the authentication cache will be automatically cleared every time an SSL error occurs and you choose to abort the connection
- *Integrate master password with the Wallet*: adds the master password to your personal Wallet/Keyring
- *Clear the master password from the Wallet*…: deletes the master password from your Wallet/Keyring
- *Enable password helper debug log*: enables a debug tool that will contain all the log information of the authentication methods
- *Clear cached authentication configurations*: clears the internal lookup cache for configurations, used to speed up network connections. This does not clear QGIS's core network access manager's cache, which requires a relaunch of QGIS.
- *Remove all authentication configurations*…: clears the database of all configuration records, without removing other stored records.
- *Import authentication configurations from file*…: imports from an `.XML` file details for creating custom authentication configurations.
- *Export selected authentication configurations to file*…: exports the selected items to a possibly encrypted `.XML` file.
- *Erase authentication database*…: schedules a backup of the current database and complete rebuild of the database table structure. The actions are scheduled for a later time, to ensure that other operations, like project loading, do not interrupt the operation or cause errors due to a temporarily missing database.

### 21.1.7 인증 환경 설정 사용하기

일반적으로, (WMS 같은) 네트워크 서비스 용 환경 설정 대화창에서 인증 환경 설정을 선택합니다. 하지만 인증이 필요한 곳이라면 어디든, 또는 제 3 자 PyQGIS 또는 C++ 플러그인 같은 비핵심 기능에 선택기 위젯을 내장시킬 수 있습니다.

When using the selector, *No authentication* is displayed in the pop-up menu control when nothing is selected, when there are no configurations to choose from, or when a previously assigned configuration can no longer be found in the database.

Use the drop-down menu to select an existing authentication configuration or press Create a new autentication configuration to create a configuration you could use. More details at 인증.

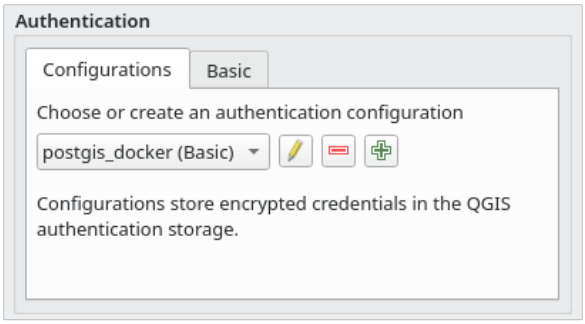



그림 21.19: 선택한 환경 설정을 표시한 인증 환경 설정 선택기

### 21.1.8 파이썬 바인딩

All classes and public functions have sip bindings, except QgsAuthCrypto, since management of the master password hashing and auth database encryption should be handled by the main app, and not via Python.

마스터 비밀번호를 입력하고 나면, 파이어폭스 작동 방식과 비슷하게 API 가 열려 인증 데이터베이스에 있는 인증 환경 설정에 접근할 수 있습니다. 하지만 PyQGIS 접근을 막는 어떤 방화벽도 정의되지 않은 상태입니다. 인증 정보에 접근할 수 있는 악성 PyQGIS 플러그인 또는 독립형 응용 프로그램을 다운로드해서 설치한 경우 문제가 발생할 수도 있습니다.

## 21.2 사용자 인증 작업 흐름

### 21.2.1 HTTP(S) 인증

가장 흔히 쓰이는 리소스 연결 가운데 하나는 예를 들어 웹 매핑 서비스처럼 HTTP(S) 프로토콜을 통한 연결로, 인증 방법 플러그인은 이런 유형의 연결에 대해 동작하는 경우가 많습니다. 인증 방법 플러그인은 HTTP 요청 객체에 접근해서 해당 요청은 물론 그 헤더까지도 조작할 수 있습니다. 이렇게 해서 수많은 형태의 인터넷 기반 인증을 할 수 있게 됩니다. 표준적인 사용자명/비밀번호 인증 방법을 이용해서 HTTP(S) 를 통해 연결하는 경우, 연결 시 HTTP BASIC 인증을 시도할 것입니다.

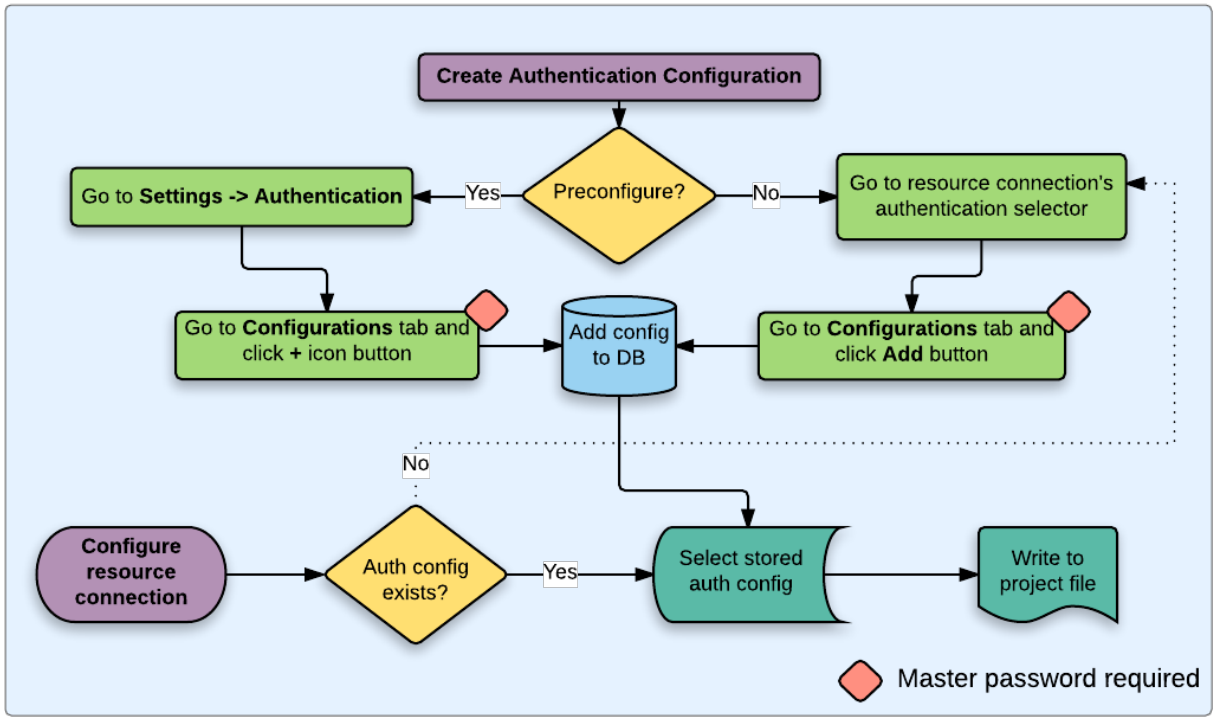



그림 21.20: 일반적인 사용자 작업 흐름

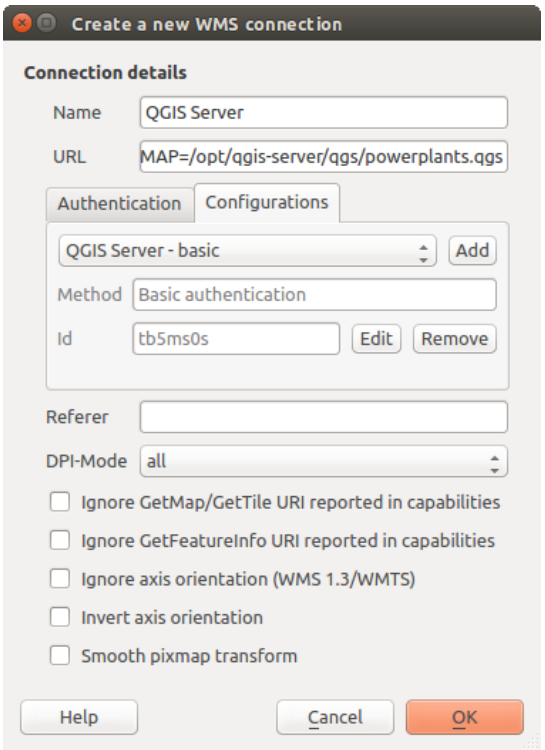



그림 21.21: HTTP BASIC 에 대한 WMS 연결 환경 설정하기

### 21.2.2 데이터베이스 인증

인증 환경 설정을 사용하지 않는다면, 데이터베이스 리소스에 대한 연결은 일반적으로 사용자 이름과 (선택적으로) 암호를 노출하게 되는 `key=value` 쌍으로 저장됩니다. 인증 시스템을 환경 설정할 때, `key=value` 는 `authfg=81t21b9` 처럼 인증 정보의 추상적인 표현이 될 것입니다.

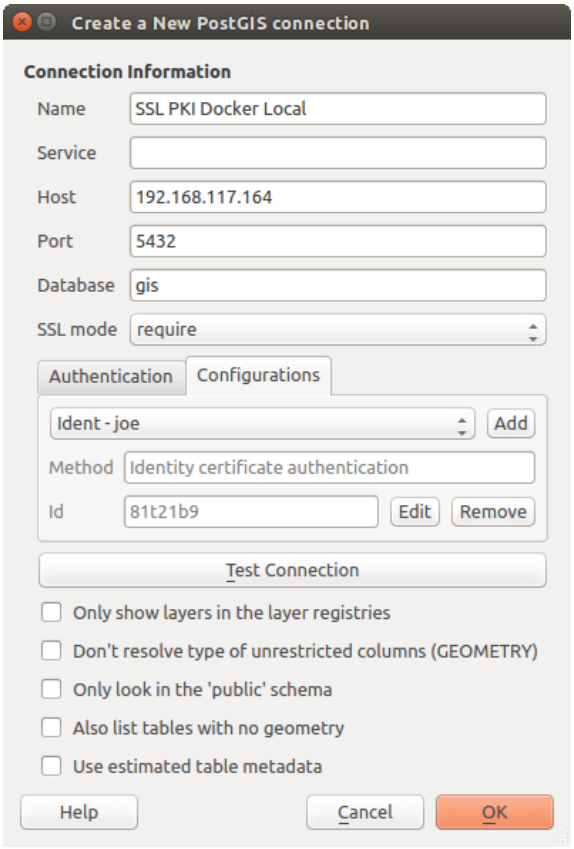



그림 21.22: PostgreSQL SSL-PKI 연결 환경 설정하기

### 21.2.3 PKI 인증

인증 시스템 내에서 PKI 구성 요소를 환경 설정하는 경우, 데이터베이스로 구성 요소를 가져오거나 또는 사용자 파일 시스템에 저장된 구성 요소 파일을 참조하는 옵션을 선택할 수 있습니다. 해당 요소가 자주 변경되거나, 또는 시스템 관리자가 해당 요소를 교체할 예정인 경우 후자가 유용할 수도 있습니다. 두 경우 모두, 데이터베이스 안에 있는 개인 키 (private key) 에 접근하는 데 필요한 암호 (passphrase) 를 모두 저장해야 할 것입니다.

**인증서 관리자** 안에 있는 개별 편집기에서 모든 PKI 구성 요소들을 관리할 수 있습니다. 인증서 관리자는 *Settings* ▸ *Options* 메뉴를 클릭하면 열리는 QGIS *Options* 대화창의 *Authentication* 탭에 있는 *Manage Certificates* 버튼을 클릭하면 열 수 있습니다.

*Certificate Manager* 에 **식별 정보 (Identities)**, **서버 (Servers)** 그리고 **권한 (Authorities)** 에 대한 편집기가 있습니다. 각각 각자의 탭이 있으며, 위의 작업 흐름도에서 발생하는 순서대로 아래에서 설명하고 있습니다. 탭 순서는 사용자가 작업 흐름에 익숙해지면 자주 사용하는 편집기 기준입니다.

**참고**

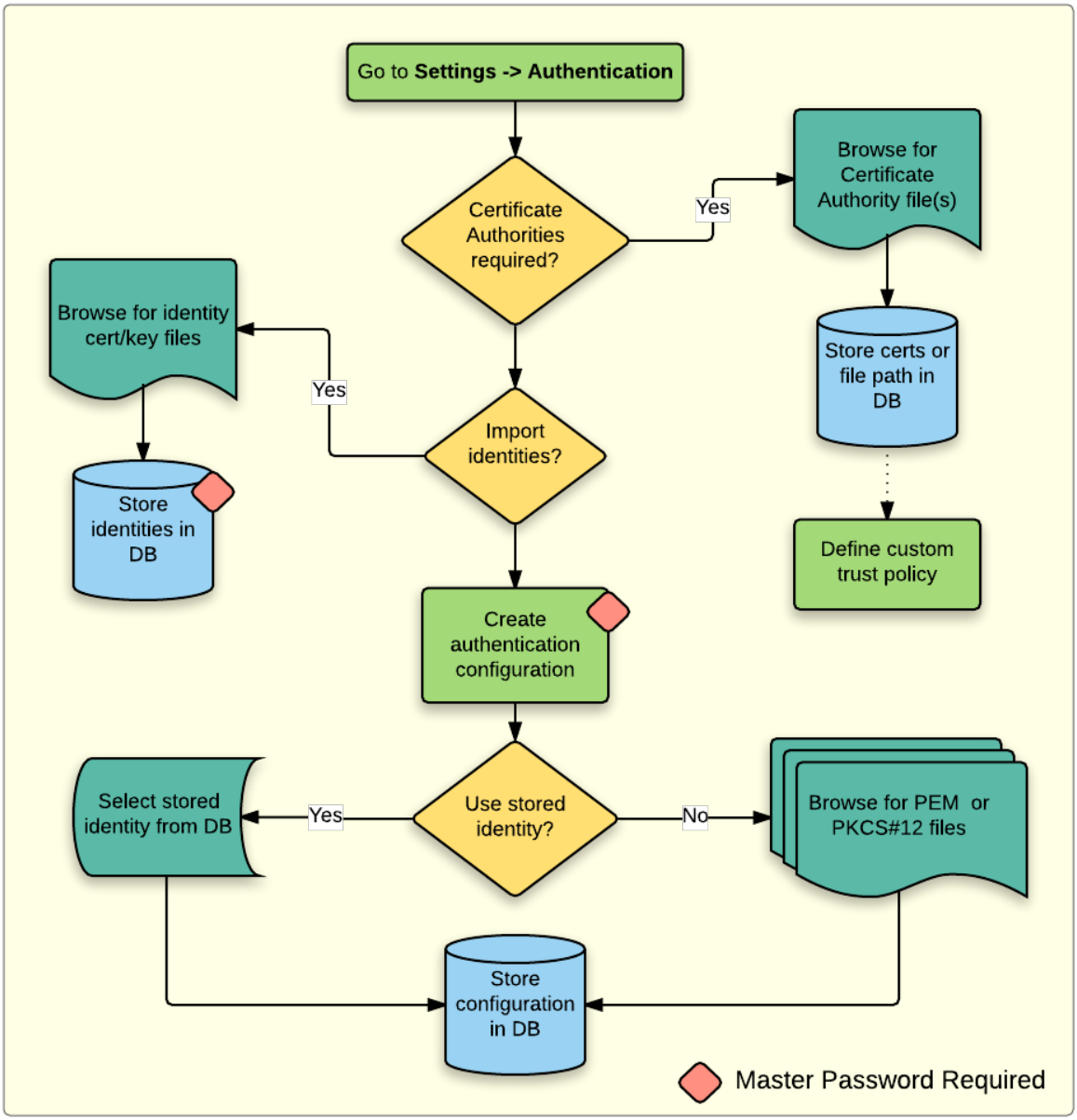



그림 21.23: PKI 환경 설정 작업 흐름

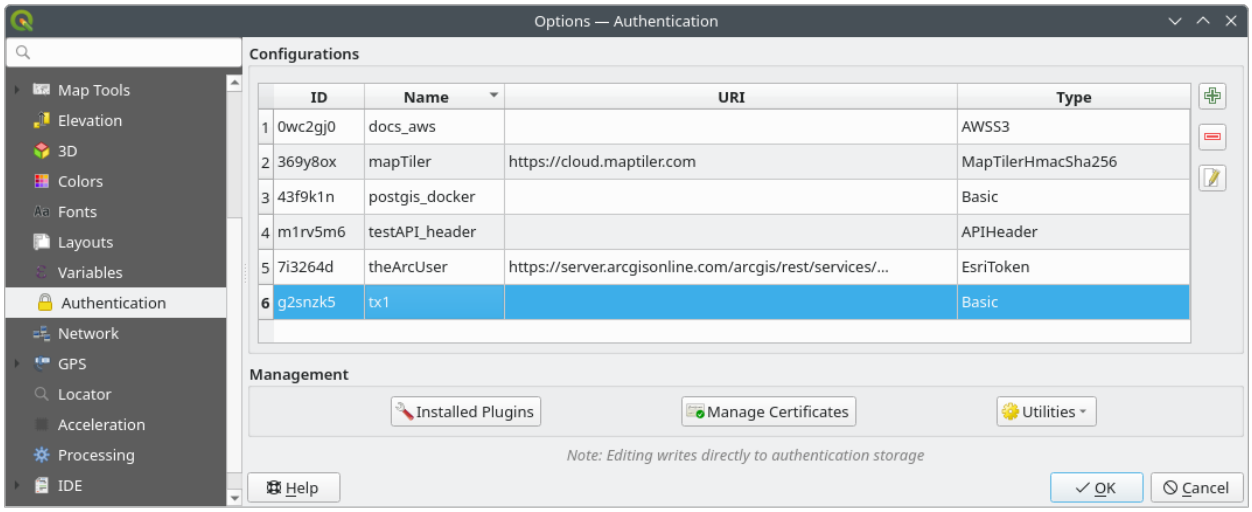



그림 21.24: 인증서 관리자 열기

모든 인증 시스템 편집 작업은 인증 데이터베이스에 바로 작성되기 때문에, 변경 사항을 저장하기 위해 *Options* 대화창에서 *OK* 버튼을 클릭해야 할 필요는 없습니다. *Options* 대화창에 있는 다른 설정들과 다른 점이죠.

### 권한

*Settings* → *Options* 대화창의 *Authentication* 탭에서 *Certificate Manager* 대화창을 열어 *Authorities* 탭을 선택하면 사용 가능한 인증서 권한 (Certificate Authorities; CAs) 을 관리할 수 있습니다.

앞의 작업 흐름도에서 볼 수 있듯이, 첫 단계로 인증서 권한의 참조 파일을 가져와야 합니다. 이 단계를 건너뛸 수도 있는데, 특히 사용자의 PKI 신뢰 연결고리 (trust chain) 가 사용자 OS 에 이미 설치돼 있는–예를 들어 상용 인증서 판매자의 인증서 같은–루트 인증서 권한에서 비롯되는 경우 이 단계가 필요없을 수도 있습니다. 사용자가 인증하는 루트 인증서 권한이 OS 가 신뢰하는 루트 인증서 권한이 아니라면 해당 인증서 권한 참조 파일을 가져오거나 파일 경로를 참조시켜야 합니다. (확실하지 않은 경우 시스템 관리자에게 연락하십시오.)

기본적으로, 사용자 OS 의 루트 인증서 권한을 사용할 수 있습니다. 하지만 OS 인증서 권한의 신뢰 설정은 상속되지 않습니다. 사용자가 인증서 신뢰 정책의 설정을 검토해야 합니다. 특히 사용자 OS 의 루트 인증서 권한의 정책이 수정됐을 경우엔 말이죠. 사용 기한이 지난 모든 인증서는, 사용자가 특별히 그 신뢰 정책을 무시하지 않는 한, 신뢰할 수 없다고 설정되어 보안된 서버 연결에 사용되지 않을 것입니다. 어느 인증서에 대해서든 QGIS 가 찾을 수 있는 신뢰 연결고리를 살펴보려면, 해당 인증서를 선택한 다음 Show information for certificate 아이콘을 클릭하십시오.

*Trust policy* 옵션을 사용하면 신뢰 연결고리 내에서 선택한 어떤 인증서든 신뢰 정책을 편집할 수 있습니다. 선택한 인증서에 대한 신뢰 정책을 어떻게 변경하든 간에, 선택한 인증서 별로 Save certificate trust policy change to database 아이콘을 클릭하지 않는 한 데이터베이스에 변경 사항을 저장하지 않을 것입니다. 대화창을 닫는다고 해도 신뢰 정책의 변경 사항은 적용되지 **않을** 것입니다.

*Options* 아이콘을 클릭하면 보안 연결에 사용할 만큼 신뢰할 수 있다고 필터링된 인증서 권한을–중간 인증서와 루트 인증서 모두–검토하거나, 또는 신뢰 정책의 기본값을 변경할 수도 있습니다.

**경고**

신뢰 정책의 기본값을 변경하면 보안 연결에 문제가 발생할 수도 있습니다.

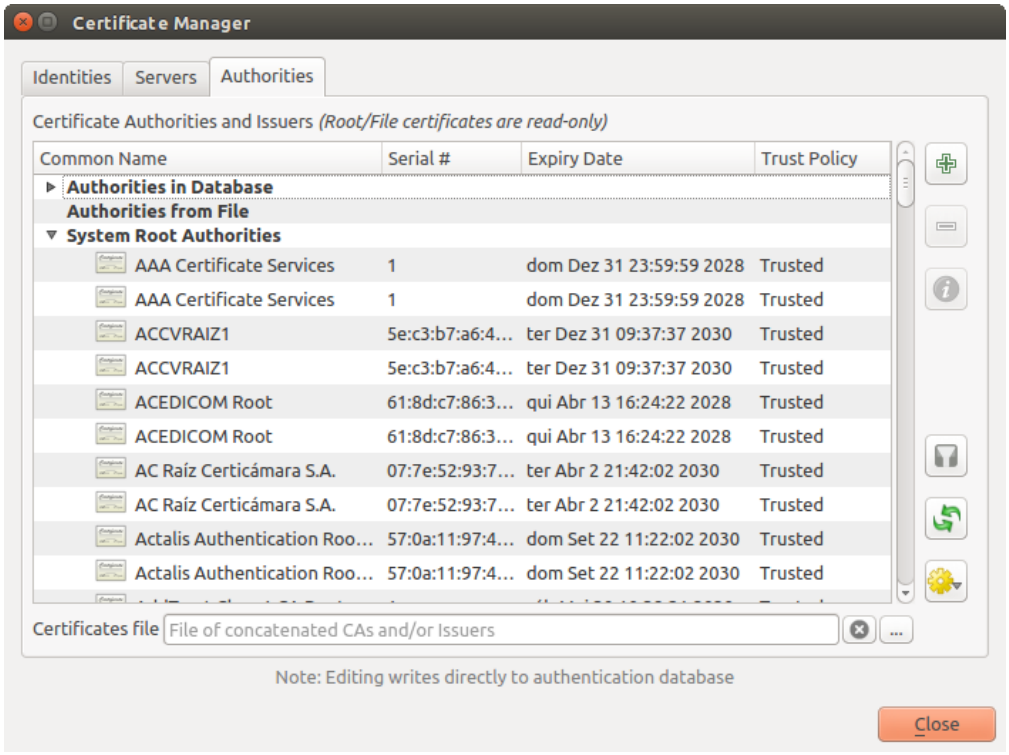



그림 21.25: 권한 편집기

인증서 권한을 개별적으로 가져오거나, 또는 여러 인증서 권한을 담고 있는 파일을 가리키는 파일 경로를 저장할 수 있습니다. 여러 CA 연결고리 인증서를 담는 표준 파일인 PEM 포맷은 파일의 제일 끝에 루트 인증서를 놓고, 그 위로 파일의 시작점을 향해 순서대로 서명된 자식 인증서들을 모두 저장합니다.

CA 인증서 가져오기 대화창은 이 파일 안에 있는 모든 CA 인증서를 순서에 상관없이 찾을 것입니다. 또 무결하지 않다고 판단되는 인증서를 (사용자가 그 신뢰 정책을 무시하려는 경우) 가져올 수 있는 옵션도 제공합니다. CA 인증서를 가져올 때 신뢰 정책을 무시할 수도 있고, 또는 나중에 **권한** 편집기에서 무시하도록 설정할 수도 있습니다.

> **참고**
>
> *PEM text* 옵션의 텍스트 란에 인증서 정보를 붙여넣는 경우, 암호화된 인증서는 지원하지 않는다는 사실을 기억하십시오.

### 신원

*Settings* ▸ *Options* 대화창의 *Authentication* 탭에서 *Certificate Manager* 대화창을 열어 *Identities* 탭을 선택하면 사용 가능한 클라이언트 신원 묶음을 관리할 수 있습니다. 신원이란 PKI 적용 서비스에 대해 사용자를 인증하는 것으로 보통 클라이언트 인증서와 개인 키로 이루어져 있는데, 두 개가 각각 개별 파일로 돼 있거나 또는 단일한"묶음 (bundled)"파일로 결합돼 있습니다. 이 묶음 또는 개인 키는 암호로 보호돼 있는 경우가 많습니다.

인증서 권한 (CAs) 을 가져오고 나면 인증 데이터베이스로 신원 묶음을 가져올 수도 있고 그러지 않을 수도 있습니다. 신원을 저장하고 싶지 않은 경우, 개별 인증 환경 설정 안에 신원 요소 파일 경로를 참조시킬 수 있습니다.

신원 묶음을 가져올 때, 신원 묶음은 암호로 보호돼 있을 수도 있고 아닐 수도 있습니다. 신원 묶음은 신뢰 연결고리를 형성하는 인증서 권한을 담고 있을 수도 있지만, 신뢰 연결고리 인증서를 가져오지는 않을 것입니다. 신뢰 연결고리 인증서는 *Authorities* 탭을 통해 개별적으로 추가할 수 있습니다.

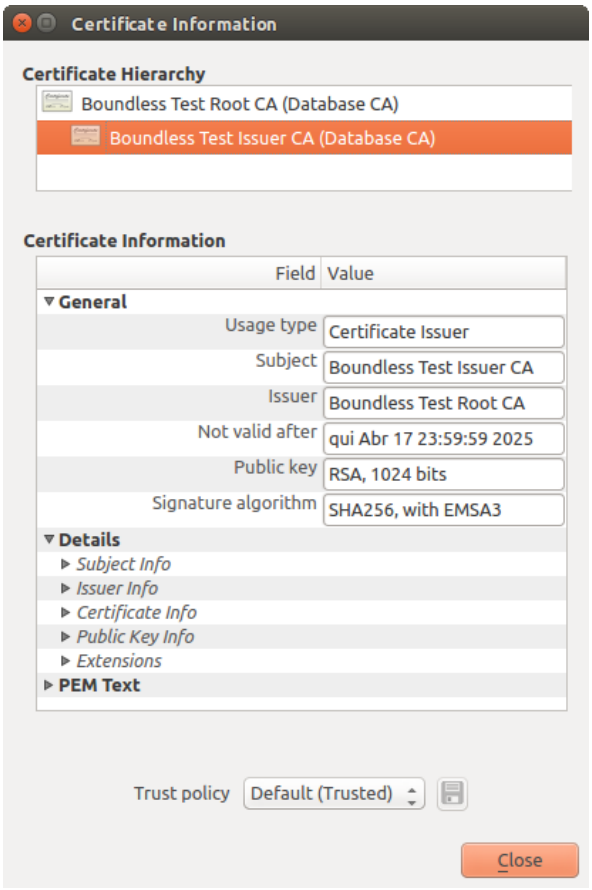



그림 21.26: 인증서 정보 대화창

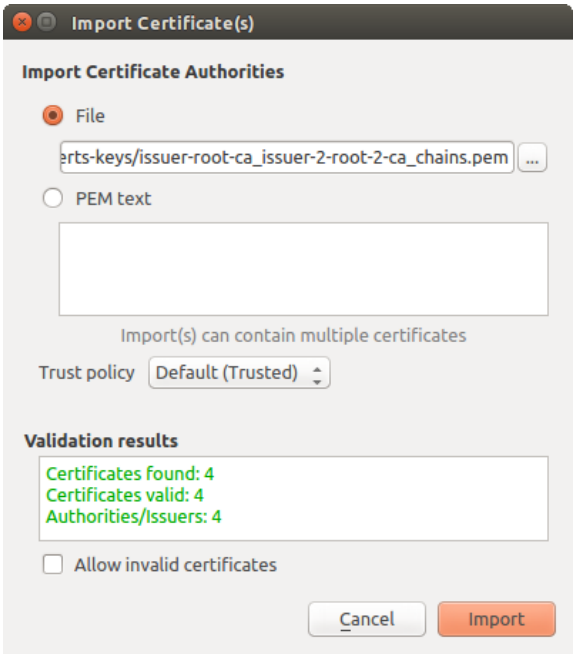



그림 21.27: 인증서 가져오기 대화창

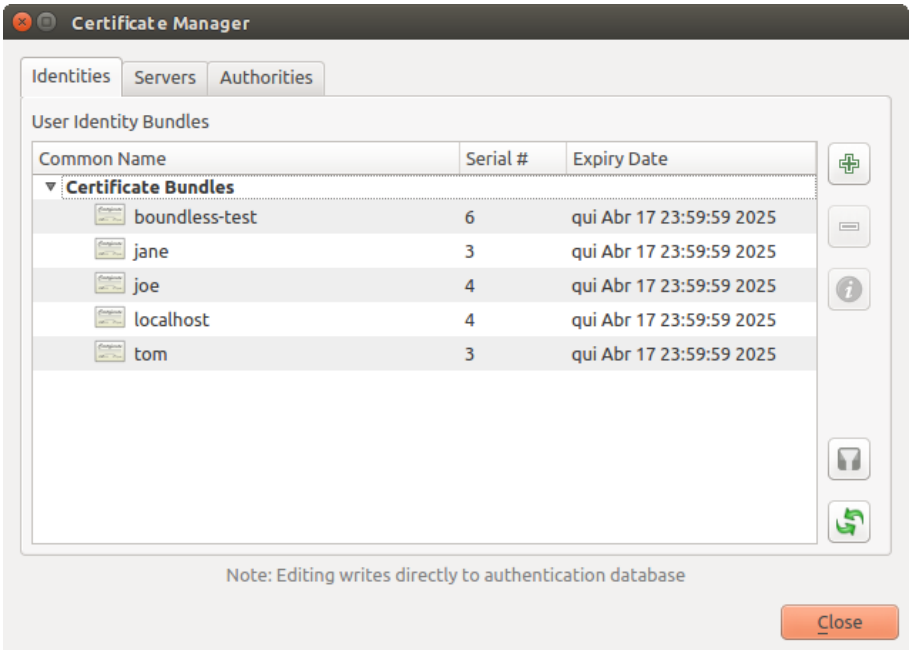



그림 21.28: 신원 편집기

신원 묶음의 인증서를 가져올 때, 개인 키는 데이터베이스에 QGIS 마스터 비밀번호를 이용해서 암호화되어 저장될 것입니다. 이후 데이터베이스에서 저장된 신원 묶음을 이용할 경우 마스터 비밀번호만 입력하면 됩니다.

PEM/DER(.pem/.der) 및 PKCS#12(.p12/.pfx) 구성 요소로 구성된 개인 신원 묶음을 지원합니다. 개인 키 또는 신원 묶음이 암호로 보호돼 있는 경우, 가져오기 전에 구성 요소를 검증하기 위해 비밀번호를 입력해야 합니다. 마찬가지로, 신원 묶음에 있는 클라이언트 인증서가 무결하지 않을 경우 (예를 들어 유효 기간 전 또는 후인 경우) 신원 묶음을 가져올 수 없습니다.

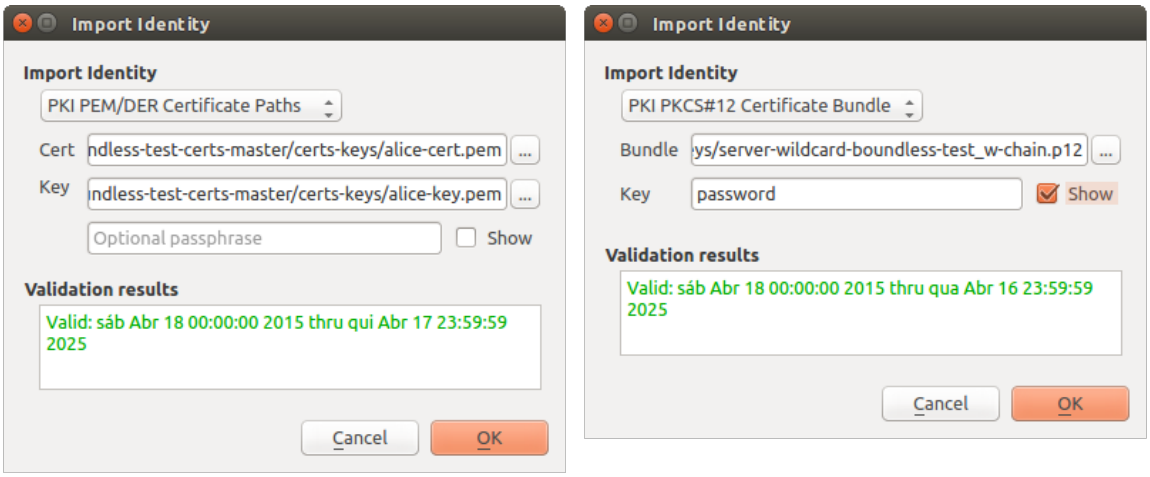



그림 21.29: PEM/DER and PKCS#12 identities import

### 21.2.4 잘못된 레이어 처리

때때로, 프로젝트 파일에 저장된 인증 환경 설정 ID 가 더 이상 무결하지 않은 경우가 있습니다. 현재 인증 데이터베이스가 프로젝트가 마지막으로 저장됐을 때와 달라졌을 수도 있고, 또는 인증 정보가 일치하지 않을 수도 있기 때문입니다. 이런 경우 QGIS 구동 시 *Handle bad layers* 대화창이 열릴 것입니다.

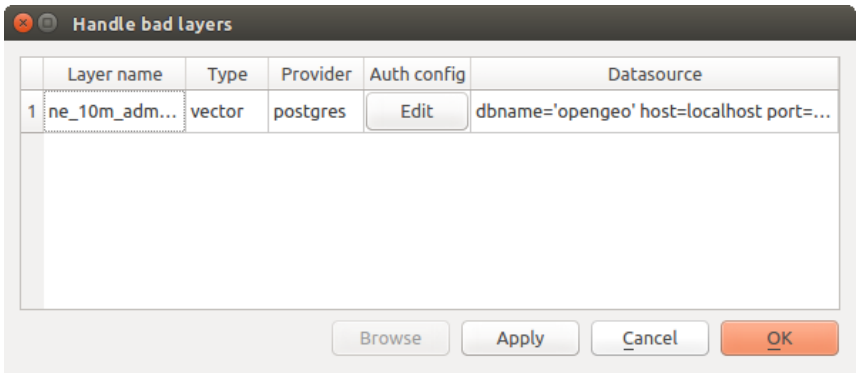



그림 21.30: 인증을 통해 잘못된 레이어 처리하기

데이터소스가 연관된 인증 환경 설정 ID 를 보유하고 있을 경우, 사용자가 편집할 수 있습니다. 이 경우 텍스트 편집기에서 프로젝트 파일을 열어 문자열을 편집하는 것과 거의 동일한 방식으로 자동적으로 데이터소스 문자열이 편집됩니다.

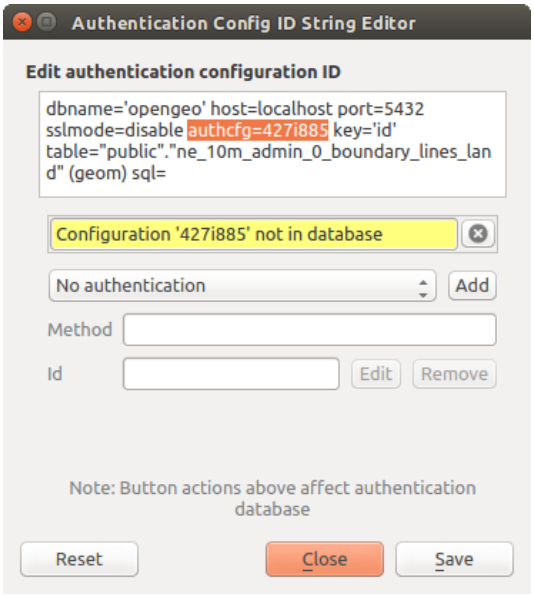



그림 21.31: 잘못된 레이어의 인증 환경 설정 ID 편집하기

### 21.2.5 인증 환경 설정 ID 변경

때때로, 리소스 접근에 관련된 인증 환경 설정 ID 를 변경해야 하는 경우가 있습니다. ID 변경이 유용한 다음과 같은 경우들이 있습니다:

- **리소스 인증 환경 설정 ID 가 더 이상 무결하지 않습니다**: 이는 이미 리소스와 연관돼 있는 ID 에 새로운 환경 설정을 할당 하기 위해 인증 데이터베이스를 전환했을 때 발생할 수 있습니다.
- **공유한 프로젝트 파일**: 사용자 사이에, 예를 들어 파일 공유 서버를 통해 프로젝트를 공유하려는 경우, 리소스와 연관된 (**a-z** 그리고/또는 **0-9** 를 포함할 수 있는) 문자 7 개 길이의 문자열을 사전에 정의할 수 있습니다. 그런 다음 개별 사용자는 리소스의 자격 증명과 관련된 인증 환경 설정 ID 를 변경합니다. 프로젝트가 공개됐을 때, ID 는 인증 데이터베이스에서 찾을 수 있지만 자격 증명은 사용자마다 다릅니다.

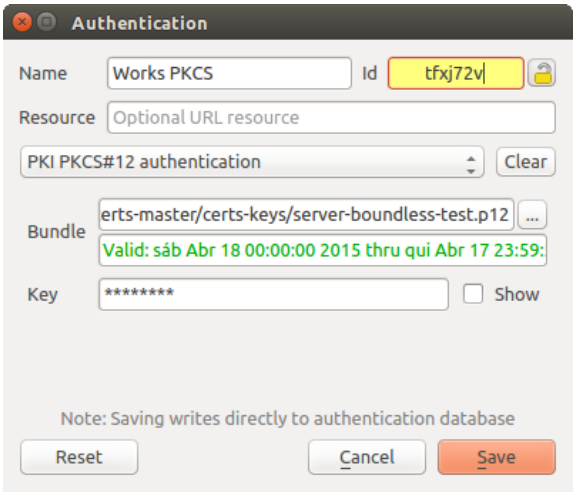



그림 21.32: 레이어의 인증 환경 설정 ID 변경하기 (고정 해제된 노란색 텍스트 란)

**경고**

인증 환경 설정 ID 를 변경하는 것은 고급 작업으로 간주되며, 변경해야 하는 이유를 완전히 이해하고 있는 경우에만 변경해야 합니다. ID 를 편집하기 전에 편집 가능 상태로 만들기 위해 ID 텍스트란 옆에 있는 자물쇠 아이콘을 클릭해야 하는 것은 이런 이유 때문입니다.

### 21.2.6 QGIS 서버 지원

QGIS 서버에서 인증 환경 설정이 적용된 레이어를 기본 맵으로 사용하는 프로젝트 파일을 사용하는 경우, QGIS 가 리소스를 불러오기 위해 필요한 다음과 같은 추가 설정 단계가 있습니다:

- 인증 데이터베이스를 사용할 수 있도록 설정해야 합니다.
- 인증 데이터베이스의 마스터 비밀번호를 사용할 수 있도록 설정해야 합니다.

인증 시스템을 처음 사용하는 경우, 서버가 활성화된사용자 프로파일 에 또는 `QGIS_AUTH_DB_DIR_PATH` 환경 변수가 정의한 디렉터리에 `qgis-auth.db` 파일을 생성하거나 사용할 것입니다. 서버 사용자에게 홈 디렉터리가 없을 수도 있는데, 이런 경우 환경 변수를 사용해서 서버 사용자가 읽기/쓰기 권한을 가지고 있으며 웹에서 접근할 수 있는 디렉터리 안에 위치하지 않은 디렉터리를 정의하십시오.

서버에 마스터 비밀번호를 넘겨주려면, 서버 프로세스 사용자가 읽을 수 있는 파일 시스템 상에 있으며 `QGIS_AUTH_PASSWORD_FILE` 환경 변수를 통해 정의한 경로에 있는 파일의 첫 행에 마스터 비밀번호를

작성하십시오. 서버 프로세스 사용자만 읽을 수 있도록 파일 권한을 제약했는지, 그리고 웹에서 접근할 수 있는 디렉터리에 파일을 저장하지는 않았는지 확인하십시오.

> **참고**
>
> 접근 후 바로 서버 환경에서 `QGIS_AUTH_PASSWORD_FILE` 환경 변수를 제거할 것입니다.

### 21.2.7 SSL 서버 예외

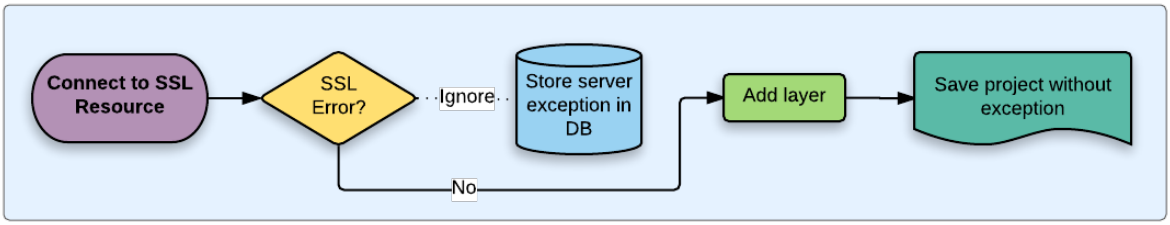



그림 21.33: SSL 서버 예외

*Settings* ▯ *Options* 대화창의 *Authentication* 탭에서 *Certificate Manager* 대화창을 열어 *Servers* 탭을 선택하면 SSL 서버 환경 설정 및 예외를 관리할 수 있습니다.

SSL 서버 연결 시 때때로 SSL 주고받기 (handshake) 또는 서버의 인증서에서 오류가 발생하는 경우가 있습니다. 이런 오류를 무시할 수도 있고 또는 이런 오류를 SSL 서버의 예외로 환경 설정할 수도 있습니다. 웹 브라우저가 SSL 오류를 무시할 수 있도록 해주는 방법과 비슷하지만, 좀 더 세밀하게 제어할 수 있습니다.

> **경고**
>
> 서버와 클라이언트 간의 SSL 전체 설정을 완전히 이해하고 있지 않다면 SSL 서버 환경 설정을 생성해서는 안 됩니다. 그 대신 서버 관리자에서 문제점을 보고하십시오.

> **참고**
>
> 일부 PKI 설정은 클라이언트 신원을 검증하기 위해 SSL 서버 인증서를 검증하는 데 사용하는 신뢰 연결고리와는 완전히 다른 인증서 권한 연결고리를 사용합니다. 이런 상황이라면, 연결 서버에 대해 어떤 환경 설정을 생성하더라도 사용자 클라이언트 신원 관련 문제점을 반드시 해결할 수 있는 것은 아닙니다. 사용자 클라이언트 신원 발행자 또는 서버 관리자만이 해당 문제점을 해결할 수 있습니다.

*Certificate Manager* 대화창에 있는  아이콘을 클릭하면 SSL 서버 환경 설정을 사전 설정할 수 있습니다. 아니면, 연결 중 SSL 오류가 발생하는 경우 환경 설정을 추가하면 (일시적으로 오류를 무시하거나 또는 데이터베이스에 환경 설정을 저장한 다음 무시할 수 있는) 다음 **SSL 오류** 대화창이 열립니다:

데이터베이스에 SSL 환경 설정을 저장하고 나면, 해당 설정을 편집하거나 삭제할 수 있습니다.

SSL 환경 설정을 사전 설정하고자 하는데 사용자의 서버 연결에 대해 가져오기 대화창이 작동하지 않는 경우, 다음 코드를 실행하면 **파이썬 콘솔** 을 통해 연결을 직접 촉발할 수 있습니다. (`https://bugreports.qt-project.org` 문자열을 사용자 서버의 URL 로 교체하십시오):

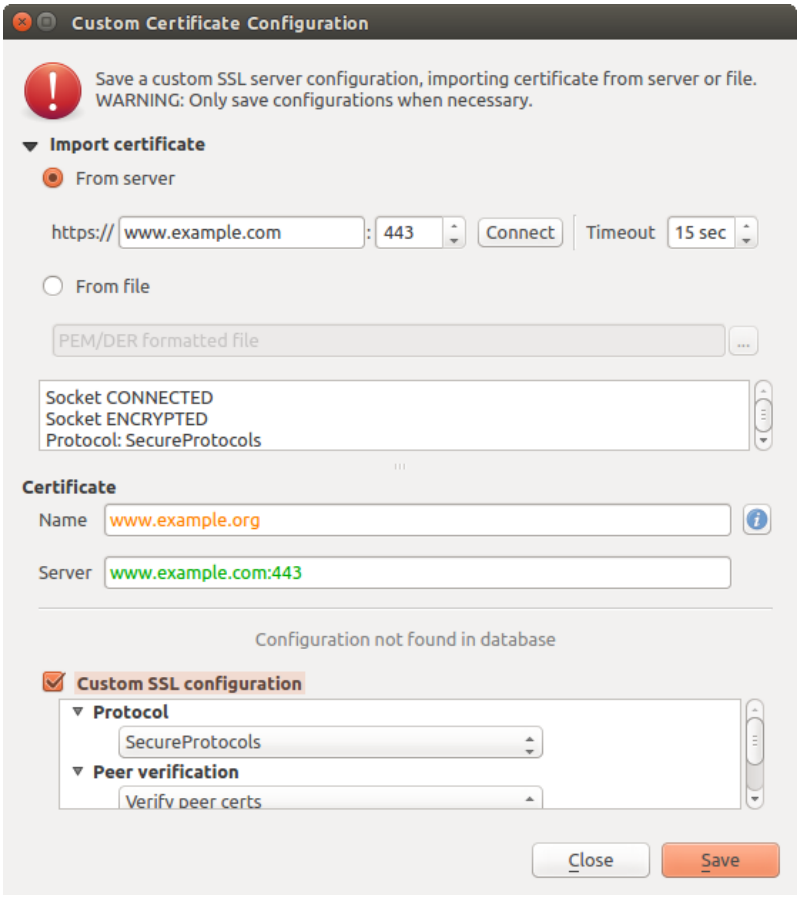



그림 21.34: 환경 설정을 직접 추가하기

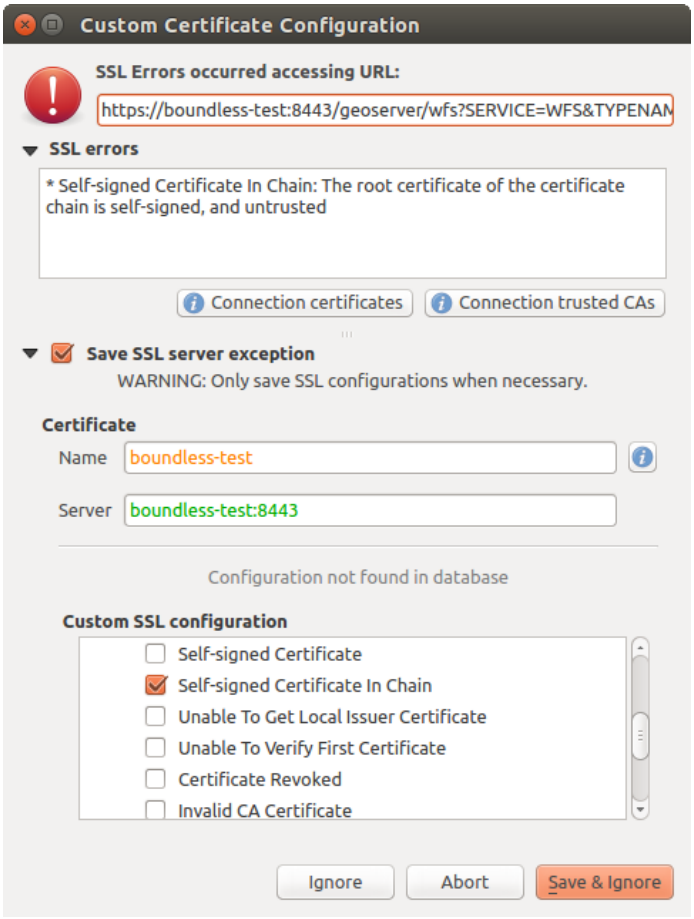



그림 21.35: SSL 오류 발생 도중 환경 설정을 추가하기

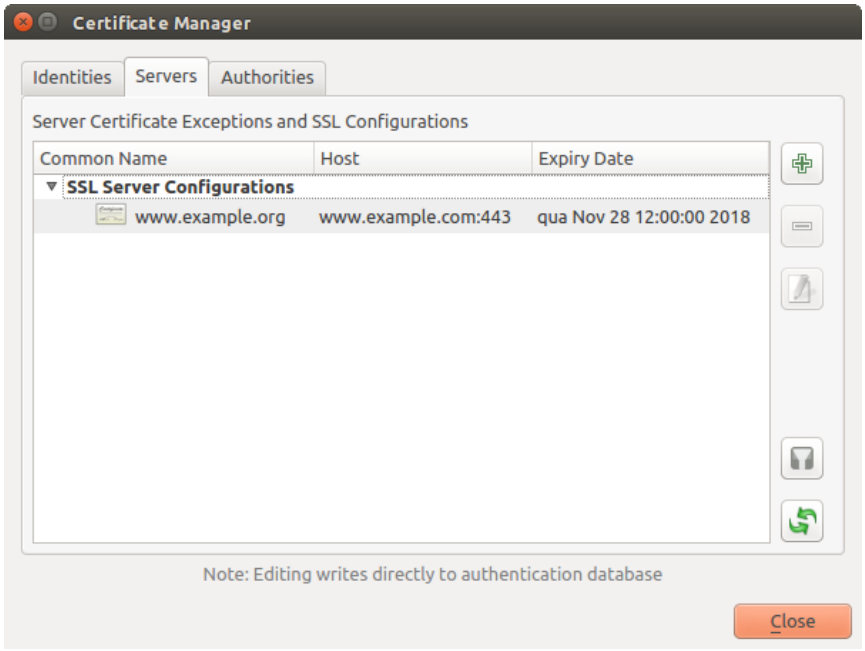



그림 21.36: 기존 SSL 환경 설정

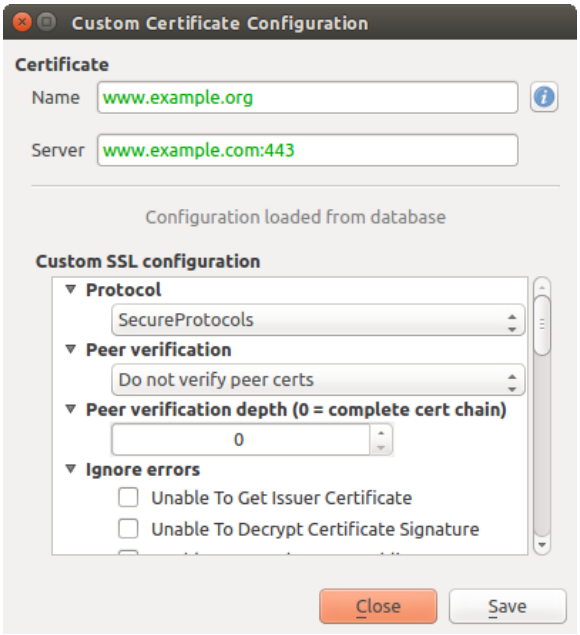



그림 21.37: 기존 SSL 환경 설정 편집하기

```
from qgis.PyQt.QtNetwork import QNetworkRequest
from qgis.PyQt.QtCore import QUrl
from qgis.core import QgsNetworkAccessManager

req = QNetworkRequest(QUrl('https://bugreports.qt-project.org'))
reply = QgsNetworkAccessManager.instance().get(req)
```

이렇게 하면 오류 발생 시 SSL 오류 대화창이 열립니다. 이 대화창에서 데이터베이스에 환경 설정을 저장하도록 선택할 수 있습니다.

CHAPTER 22

# GRASS GIS 통합

GRASS 를 통합하면 GRASS GIS 데이터베이스 및 기능에 접근할 수 있습니다. (참고 문헌 및 웹사이트 에서 GRASS-PROJECT 를 참조하세요.) 통합은 제공자와 플러그인 두 부분으로 이루어집니다. 제공자는 GRASS 래스터 및 벡터 레이어를 탐색하고 관리하고 가시화할 수 있게 해줍니다. 플러그인을 사용하면 새 GRASS 로케이션 및 맵셋을 생성하고, GRASS 영역을 변경하며, 벡터 레이어를 생성 및 편집하고, 400 개가 넘는 GRASS 모듈을 통해 GRASS 2 차원 및 3 차원 데이터를 분석할 수 있습니다. 이 장에서 제공자와 플러그인의 기능을 소개하고 GRASS 데이터를 관리하고 작업하는 몇몇 예시를 제공할 것입니다.

제공자는 GRASS 6 및 7 버전을 지원하며, 플러그인도 (QGIS 2.12 버전부터) GRASS 6 및 7 버전을 지원합니다. QGIS 배포판이 GRASS 6, GRASS 7 가운데 하나를 지원하는 제공자/플러그인을 담고 있을 수도 있고, 또는 6 및 7 버전을 지원하는 제공자/플러그인을 함께 담고 있을 수도 있습니다. (바이너리 파일들이 서로 다른 파일명입니다.) 하지만 실행 시간에는 둘 중 한 버전을 지원하는 제공자/플러그인만 불러올 수 있습니다.

## 22.1 예제 데이터셋

As an example, we will use the QGIS Alaska dataset (see section 예시 데이터 다운로드). It includes a small sample GRASS `LOCATION` with three vector layers and one raster elevation map. Create a new folder called `grassdata`, download the QGIS 'Alaska'dataset `qgis_sample_data.zip` from https://qgis.org/downloads-list/#data and unzip the file into `grassdata`.

More sample GRASS `LOCATIONs` are available at the GRASS website.

## 22.2 GRASS 래스터 및 벡터 레이어 불러오기

QGIS 에 제공자를 불러오면, 그 아래에 GRASS 로케이션을 담고 있는 각 폴더 항목을 가지고 있는 탐색기 트리에 로케이션 항목을 GRASS  아이콘과 함께 추가합니다. `grassdata` 폴더를 찾아 `alaska` 로케이션 및 `demo` 맵셋을 펼치십시오.

다른 모든 레이어와 마찬가지로, GRASS 레이어 항목을 더블클릭하거나 맵 캔버스 또는 범례로 드래그 & 드롭하면 GRASS 래스터 및 벡터 레이어를 불러올 수 있습니다.

> **팁**
>
> **GRASS 데이터 불러오기**
>
> GRASS 로케이션 항목을 찾을 수 없는 경우 *Help* → *About* → *Providers* 메뉴 옵션에서 GRASS 벡터 제공자를 불러온 상태인지 확인하십시오.

## 22.3 드래그 & 드롭으로 GRASS 로케이션에 데이터 가져오기

이 절에서는 GRASS 맵셋으로 래스터 및 벡터 데이터를 가져오는 방법을 예제를 통해 설명하겠습니다.

1. QGIS 탐색기에서 사용자가 데이터를 가져오고 싶은 맵셋을 찾으십시오.
2. QGIS 탐색기에서 GRASS 로 가져오고 싶은 레이어를 찾으십시오. 소스 데이터가 트리의 맵셋 위치에서 너무 멀리 떨어져 있는 경우 새 탐색기 인스턴스 (*Browser Panel (2)*) 를 열 수 있다는 사실을 기억하세요.
3. 대상 맵셋에 레이어를 드래그 & 드롭하십시오. 대용량 레이어인 경우 가져오기에 시간이 좀 걸릴 수도 있습니다. 가져오기를 완료할 때까지 새 레이어 항목 앞에 움직이는  아이콘이 나타날 겁니다.

래스터 데이터가 서로 다른 좌표계를 사용하는 경우, (빠른) *Approximate* 변형 또는 (정확한) *Exact* 변형을 사용해서 재투영할 수 있습니다. 소스 래스터를 가리키는 링크를 (`r.external` 을 이용해서) 생성한 상태이며, 소스 데이터가 동일한 좌표계를 사용하고 있고 GDAL 이 지원하는 포맷인 경우, 소스 데이터의 좌표계를 사용할 것입니다. *GRASS 옵션* 대화창에 있는 *Browser* 탭에서 이 옵션들을 설정할 수 있습니다.

소스 래스터가 더 많은 밴드를 가지고 있는 경우, 각 밴드 레이어 별로 **.< 밴드 번호 >** 접미어를 붙인 새 GRASS 맵을 생성한 다음 모든 맵들의 그룹을 래스터 그룹  아이콘과 함께 생성합니다. 외부 래스터인 경우 래스터 링크  아이콘을 사용합니다.

## 22.4 QGIS 탐색기에서 GRASS 데이터 관리하기

- 맵 복사하기: 동일한 로케이션 내에 있는 맵셋들 간에 GRASS 맵을 드래그 & 드롭해서 복사할 수도 있습니다.
- 맵 삭제하기: GRASS 맵을 오른쪽 클릭한 다음 컨텍스트 메뉴에서 *Delete* 액션을 선택하십시오.
- 맵 재명명하기: GRASS 맵을 오른쪽 클릭한 다음 컨텍스트 메뉴에서 *Rename* 액션을 선택하십시오.

## 22.5 GRASS 옵션

*GRASS Options* 대화창에서 GRASS 옵션을 설정할 수도 있습니다. 해당 대화창은 탐색기에서 로케이션 또는 맵셋을 오른쪽 클릭한 다음 컨텍스트 메뉴에서 *GRASS Options* 를 선택하면 열 수 있습니다.

## 22.6 GRASS 플러그인 시작하기

QGIS 에서 GRASS 기능을 사용하려면, 플러그인 관리자를 통해 GRASS 플러그인을 선택하고 불러와야만 합니다. *Plugins* → *Manage and Install Plugins*…메뉴를 선택하고 플러그인 관리자 대화창에서 *GRASS* 를 체크한 다음 *OK* 버튼을 클릭하십시오.

GRASS 플러그인을 시작하면 GRASS 메뉴 (*Plugins* → *GRASS*) 를 통해 다음 주 기능들을 제공합니다:

- Open Mapset
- New Mapset
- Close Mapset
- Open GRASS Tools
- Display Current GRASS Region
- GRASS Options

## 22.7 GRASS 맵셋 열기

플러그인의 GRASS 도구를 사용하려면 먼저 GRASS 맵셋을 열어야만 합니다. (열려 있는 맵셋이 없는 경우 도구는 비활성화 상태입니다.) 탐색기에서 맵셋 항목을 오른쪽 클릭한 다음 컨텍스트 메뉴에서 *Open mapset* 을 선택하면 맵셋을 열 수 있습니다.

## 22.8 GRASS 로케이션 및 맵셋

GRASS 데이터는 `GISDBASE` 라고 참조되는 디렉터리에 저장됩니다. `grassdata` 로 불리는 경우가 많은 이 디렉터리는 QGIS 에서 GRASS 플러그인 작업을 시작하기 전에 생성해야만 합니다. 이 디렉터리 안에서, GRASS GIS 데이터는 프로젝트 별로 정리되어 `LOCATION` 이라고 불리는 하위 디렉터리에 저장됩니다. 각 `LOCATION` 은 그 좌표계, 맵 투영체 그리고 지리적 경계로 정의됩니다. 각 `LOCATION` 은 (`LOCATION` 의 하위 디렉터리인) `MAPSET` 을 여러 개 보유할 수 있습니다. `MAPSET` 은 프로젝트를 서로 다른 주제 또는 하위 영역으로 세분하는 데, 또는 팀 멤버 개인 별 작업 공간 (workspace) 을 제공하는 데 사용됩니다. (참고 문헌 및 웹사이트 에서 소개하고 있는 마커스 네틀러와 헬레나 미타소바가 2008 년 출판한『Open Source GIS: A GRASS GIS Approach』를 참조하세요.) GRASS 모듈로 벡터 및 래스터 레이어를 분석하려면, 일반적으로 GRASS `LOCATION` 으로 레이어를 가져와야 합니다. (정확한 사실은 아닙니다 — GRASS 모듈 `r.external` 및 `v.external` 을 사용하면 레이어를 가져오지 않고도 외부 GDAL 지원 데이터셋을 가리키는 읽기 전용 링크를 생성할 수 있습니다. 아직 GRASS 작업에 익숙하지 않은 사용자가 쓰기에는 일반적이지 않은 방법이기 때문에, 이 문서에서는 이 기능을 설명하지 않을 것입니다.)

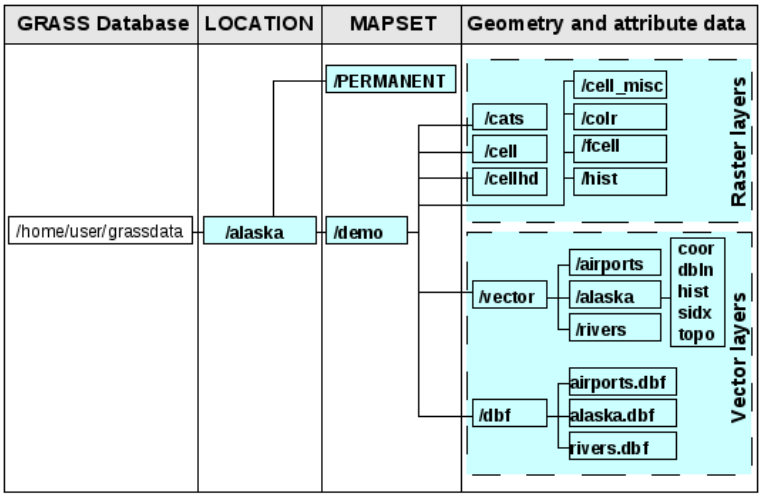



그림 22.1: 알래스카 로케이션에 있는 GRASS 데이터

## 22.9 GRASS 로케이션에 데이터 가져오기

탐색기에서 드래그 & 드롭으로 데이터를 쉽게 가져올 수 있는 방법은드래그 & 드롭으로 *GRASS* 로케이션에 데이터 가져오기 를 참조하세요.

이 절에서는 QGIS 알래스카 데이터셋이 제공한 GRASS `LOCATION` 'alaska'로 래스터 및 벡터 데이터를 전통적인 방식으로 가져오는 방법을 예제를 통해 설명하겠습니다. 따라서 QGIS 알래스카 데이터셋으로부터 (예시 데이터 다운로드 참조) 랜드커버 래스터 맵 `landcover.img` 와 벡터 GML 파일 `lakes.gml` 을 사용합니다.

1. QGIS 를 구동하고 GRASS 플러그인을 불러온 상태인지 확인하십시오.
2. GRASS 툴바에서, Open MAPSET 아이콘을 클릭해서 *MAPSET* 마법사를 여십시오.
3. QGIS 알래스카 데이터셋에서 `grassdata` 폴더를 — `LOCATION` 'alaska', `MAPSET` 'demo'를 — GRASS 데이터베이스로 선택한 다음 *OK* 버튼을 누르십시오.
4. 그리고 Open GRASS tools 아이콘을 클릭하십시오. *GRASS* 툴박스 대화창이 열립니다.
5. `landcover.img` 래스터 맵을 가져오려면, *Modules Tree* 탭에서 `r.in.gdal` 모듈을 클릭하십시오. 이 GRASS 모듈을 통해 GRASS `LOCATION` 으로 GDAL 지원 래스터 파일을 가져올 수 있습니다. `r.in.gdal` 모듈 대화창이 열릴 것입니다.
6. QGIS 알래스카 데이터셋에서 `raster` 폴더를 찾아 `landcover.img` 파일을 선택하십시오.
7. 래스터 산출물 이름을 `landcover_grass` 라고 지정한 다음 *Run* 버튼을 클릭하십시오. *Output* 탭에 현재 실행중인 GRASS 명령어 `r.in.gdal -o input=/path/to/landcover.img output=landcover_grass` 가 표시됩니다.
8. **Successfully finished** 메시지가 표시되면, *View Output* 버튼을 클릭하십시오. 이제 GRASS 로 `landcover_grass` 래스터 레이어를 가져와서 QGIS 캔버스에 가시화할 것입니다.
9. `lakes.gml` GML 파일을 가져오려면, *Modules Tree* 탭에서 `v.in.ogr` 모듈을 클릭하십시오. 이 GRASS 모듈을 통해 GRASS `LOCATION` 으로 OGR 지원 벡터 파일을 가져올 수 있습니다. `v.in.ogr` 모듈 대화창이 열릴 것입니다.
10. QGIS 알래스카 데이터셋에서 `gml` 폴더를 찾아 `lakes.gml` 파일을 선택하십시오.
11. 벡터 산출물 이름을 `lakes_grass` 라고 지정한 다음 *Run* 버튼을 클릭하십시오. 이 예제에서는 다른 옵션을

신경쓰지 않아도 됩니다. *Output* 탭에 현재 실행중인 GRASS 명령어 `v.in.ogr -o dsn=/path/to/lakes.gml output=lakes_grass` 가 표시됩니다.

12. **Successfully finished** 메시지가 표시되면, *View Output* 버튼을 클릭하십시오. 이제 GRASS 로 `lakes_grass` 벡터 레이어를 가져와서 QGIS 캔버스에 가시화할 것입니다.

### 22.9.1 새 GRASS 로케이션 생성하기

이 예제에서, 피트 단위를 사용하는 알베르스 정적도법 (Albers Equal Area projection) 으로 투영된 예시 GRASS `LOCATION` 'alaska'를 예시 데이터로 사용합니다. 이 다음에 나오는 GRASS 관련 절, 항에서 설명하는 모든 예시 및 예제에서 이 예시 GRASS `LOCATION` 'alaska'를 사용할 것입니다. 사용자 컴퓨터에 QGIS 예시 데이터셋을 다운로드하고 설치하는 편이 좋습니다. (예시 데이터 다운로드 를 참조하세요.)

1. QGIS 를 구동하고 GRASS 플러그인을 불러온 상태인지 확인하십시오.
2. QGIS 알래스카 데이터셋에서 (예시 데이터 다운로드 참조) `alaska.shp` 셰이프파일을 가시화하십시오. (파일에서 레이어 불러오기 를 참조하세요.)
3. GRASS 툴바에서,  New mapset 아이콘을 클릭해서 *MAPSET* 마법사를 여십시오.
4. 기존 GRASS 데이터베이스 (GISDBASE) 폴더 `grassdata` 를 선택하거나, 또는 사용자 컴퓨터에서 파일 탐색기를 사용해서 새 `LOCATION` 용 폴더를 생성한 다음 *Next* 버튼을 클릭하십시오.
5. 이 마법사를 통해 기존 `LOCATION` 안에 새 `MAPSET` 을 생성하거나 (새 맵셋 추가하기 참조) 또는 새 `LOCATION` 을 처음부터 생성할 수 있습니다.  *Create new location* 옵션을 체크하십시오. (그림 22.2 를 참조하세요.)
6. `LOCATION` 을 위한 이름을 — 이 예제에서는'alaska'를 — 입력한 다음 *Next* 버튼을 클릭하십시오.
7.  *Projection* 라디오 버튼을 체크해서 투영체 목록을 활성화한 다음 투영체를 선택하십시오.
8. 알베르스 정적도법 알래스카 (피트 단위) 투영체를 사용할 것입니다. 이 투영체가 `EPSG ID 2964` 라는 사실을 이미 알고 있으므로, 검색 란에 해당 문자열을 입력할 수 있습니다. (주의 사항: 또다른 `LOCATION` 과 그 투영체에 대해 이 과정을 반복하려 하는데 ESPG ID 를 기억하지 못 하는 경우, 상태 바의 우하단에 있는  CRS Status 아이콘을 클릭하십시오. (투영 작업 참조))
9. *Filter* 란에 `2964` 를 입력해서 투영체를 선택하십시오.
10. *Next* 를 클릭합니다.
11. 기본 영역 (region) 을 정의하려면, 동서남북 방향으로 `LOCATION` 의 경계를 입력해야 합니다. 이 예제에서는 그냥 *Set Current QGIS Extent* 버튼을 클릭해서, 불러온 `alaska.shp` 레이어의 범위를 GRASS 기본 영역 범위로 적용하겠습니다.
12. *Next* 를 클릭합니다.
13. 새 `LOCATION` 안에 `MAPSET` 도 정의해야 합니다. (새 `LOCATION` 을 생성하는 경우 꼭 필요한 단계입니다.) 사용자가 원하는 대로 명명할 수 있지만, 이 예제에서는'demo'라고 하겠습니다. GRASS 가 `PERMANENT` 라고 불리는 `MAPSET` 을 자동적으로 생성할 것입니다. `PERMANENT` 는 프로젝트 용 핵심 데이터, 데이터의 공간 범위 및 좌표계 정의를 저장하기 위해 고안됐습니다. (참고 문헌 및 웹사이트 에서 소개하는 Neteler, M., and Mitasova, H. (2008) 을 참조하세요.)
14. 작업 내용이 정확한지 작업 개요를 확인한 다음 *Finish* 버튼을 클릭하십시오.
15. 새 `LOCATION` 인'alaska'와 2 개의 `MAPSET` 'demo'및'PERMANENT'가 생성됐습니다. 현재 열려 있는 작업 맵셋은 사용자가 정의한 대로'demo'입니다.
16. GRASS 툴바에서 비활성 상태였던 일부 도구들이 이제 활성화됐다는 사실을 주목하십시오.

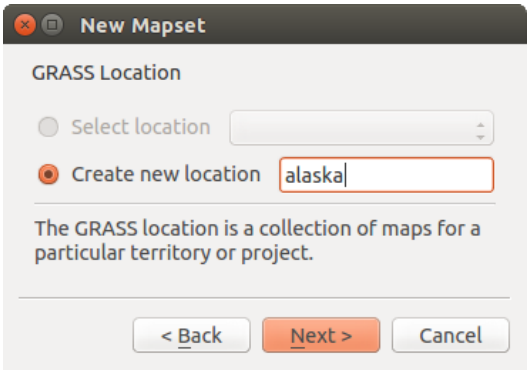



그림 22.2: QGIS 에서 새 GRASS 로케이션 또는 새 맵셋 생성하기

작업 단계가 너무 많다고 생각할 수도 있겠지만, 사실 LOCATION 을 생성할 수 있는 매우 빠른 방법으로 그렇게 나쁘지는 않습니다. 이제 LOCATION 'alaska'에 데이터를 가져올 준비가 끝났습니다. (*GRASS* 로케이션에 데이터 가져오기 를 참조하세요.) 아니면 QGIS 알래스카 데이터셋예시 데이터 다운로드 에 포함된 예시 GRASS LOCATION 'alaska'에 있는 기존 벡터 및 래스터 데이터를 사용해서 바로*GRASS* 벡터 데이터 모델 절로 넘어갈 수도 있습니다.

### 22.9.2 새 맵셋 추가하기

사용자는 본인이 생성한 GRASS MAPSET 에 대해서만 쓰기 권한을 가집니다. 즉 사용자 본인의 MAPSET 을 읽고 쓰는 것 이외에도 다른 사용자의 MAPSET 에 있는 맵을 읽을 수 있다는 뜻입니다. (다른 사용자도 사용자의 맵을 읽을 수 있습니다.) 그러나 사용자는 오직 본인이 생성한 MAPSET 에 있는 맵만을 수정하거나 제거할 수 있습니다.

모든 MAPSET 은 현재 경계 좌표값과 현재 선택한 래스터 해상도를 저장하는 WIND 파일을 포함하고 있습니다. (참고 문헌 및 웹사이트 에서 소개하는 Neteler, M., and Mitasova, H. (2008) 및*GRASS* 영역 도구 를 참조하세요.)

1. QGIS 를 구동하고 GRASS 플러그인을 불러온 상태인지 확인하십시오.
2. GRASS 툴바에서,  New mapset 아이콘을 클릭해서 *MAPSET* 마법사를 여십시오.
3. LOCATION 'alaska'가 있는 GRASS 데이터베이스 (GISDBASE) 폴더 grassdata 를 선택하십시오. LOCATION 'alaska'에'test'라는 심화 MAPSET 을 추가하려 합니다.
4. *Next* 를 클릭합니다.
5. 이 마법사를 통해 기존 LOCATION 안에 새 MAPSET 을 생성하거나 (새 맵셋 추가하기 참조) 또는 새 LOCATION 을 처음부터 생성할 수 있습니다.  *Select location* 라디오 버튼을 클릭한 다음 (그림 22.2 참조) *Next* 버튼을 클릭하십시오.
6. 새 MAPSET 에 대해 test 라는 이름을 입력하십시오. 마법사 대화창 하단에 기존 MAPSET 및 해당 소유자의 목록이 표시됩니다.
7. *Next* 버튼을 클릭해서 작업 내용이 정확한지 작업 개요를 확인한 다음 *Finish* 버튼을 클릭하십시오.

## 22.10 GRASS 벡터 데이터 모델

디지타이즈 작업 전에"GRASS 벡터 데이터 모델"을 이해하고 있어야 합니다. 일반적으로, GRASS 는 위상 벡터 모델을 이용합니다. 즉 면 (area) 을 닫힌 폴리곤으로 표현하는 것이 아니라 하나 이상의 경계로 표현한다는 뜻입니다. 인접한 두 면 사이의 경계를 한 번에 디지타이즈하고, 두 면이 해당 경계를 공유한다는 것입니다. 이 경계는 연결되어 빈자리 없이 닫힌 상태여야만 합니다. 면의 **중심 (centroid)** 이 면을 정의하고, 라벨의 위치가 됩니다.

경계 및 중심 이외에도, 벡터 맵은 포인트 및 라인을 담고 있을 수 있습니다. 이 모든 도형 항목들이 벡터 하나에 혼합되어 있을 수 있으며, 하나의 GRASS 벡터 맵 안에 서로 다른 소위'레이어'들로 표현될 것입니다. 따라서 GRASS 에서는, 레이어가 벡터 또는 래스터 맵이 아니라 벡터 레이어 내부의'층 (level)'이 됩니다. 이런 개념을 주의해서 구분해야만 합니다. (다만 도형 항목들을 혼합할 수 있다고는 해도, GRASS 에서조차 벡터 네트워크 분석과 같은 특별한 경우에만 사용되는 흔치 않은 상황입니다. 일반적으로, 서로 다른 도형 항목들을 서로 다른 레이어에 저장하는 편이 좋을 겁니다.)

하나의 벡터 데이터셋에 여러'레이어'를 저장할 수 있습니다. 예를 들면, 벡터 하나 안에 평야, 삼림, 호수 등을 저장할 수 있습니다. 인접한 삼림과 호수는 동일한 경계를 공유할 수 있지만, 각각 개별 속성 테이블을 보유합니다. 또 경계에 속성을 붙일 수도 있습니다. 삼림과 호수 사이의 경계가 도로이기 때문에 도로가 고유한 속성 테이블을 보유하는 경우를 예로 들 수 있겠죠.

GRASS 에서 객체의'레이어'는 숫자로 된'레이어'값으로 정의됩니다. 한 데이터셋 (예를 들어 삼림 또는 호수를 표현하는 도형) 안에 여러 레이어가 있는 경우'레이어'는 숫자로 지정됩니다. 현재로서는 숫자로만 지정할 수 있지만, 앞으로는 GRASS 사용자 인터페이스 상에서 이름으로 지정할 수 있게 될 예정입니다.

속성은 GRASS `LOCATION` 내부에 dBase, SQLite3 으로 저장하거나, 또는 외부 데이터베이스 테이블에, 예를 들어 PostgreSQL, MySQL, Oracle 등에 저장할 수 있습니다.

데이터베이스 테이블에 있는 속성은 `category` 값을 통해 도형 항목과 링크되어 있습니다.

`categoty` (키, ID) 는 도형 원시형 (primitive) 에 첨부된 정수로, 데이터베이스 테이블에 있는 유일 키 열을 가리키는 링크로 사용됩니다.

> **팁**
>
> **GRASS 벡터 모델 이해하기**
>
> GRASS 벡터 모델과 그 능력을 이해하는 데 있어 가장 좋은 방법은 벡터 모델을 좀 더 심도 있게 설명하고 있는 여러 GRASS 예제 가운데 하나를 다운로드하는 것입니다. 자세한 정보 및 문서 그리고 예제를 여러 언어로 제공하고 있는 https://grass.osgeo.org/learn/manuals/ 웹페이지를 참조하세요.

## 22.11 새 GRASS 벡터 레이어 생성하기

새 GRASS 벡터 레이어를 생성하려면, 탐색기에 있는 맵셋의 컨텍스트 메뉴에서 다음 액션 가운데 하나를 선택하십시오:

- *New Point Layer*: 새 포인트 레이어를 추가합니다.
- *New Line Layer*: 새 라인 레이어를 추가합니다.
- *New Polygon Layer*: 새 폴리곤 레이어를 추가합니다.

그리고 대화창에서 이름을 입력하십시오. 새 벡터 맵이 생성되어 맵 캔버스에 편집 모드 상태로 추가될 것입니다. 레이어 유형을 선택한다고 해도 벡터 맵 안에 디지타이즈할 수 있는 도형 유형이 제한되지는 않습니다. GRASS 에서는, 하나의 벡터 맵 안에 모든 도형 유형을 (포인트, 라인 및 폴리곤을) 구조화시킬 수 있습니다. 레이어 유형은

맵 캔버스에 레이어를 추가하기 위해서만 사용됩니다. 왜냐하면 QGIS 는 레이어가 특정 유형일 것을 요구하기 때문입니다.

기존 벡터 맵의 컨텍스트 메뉴에서 앞에서 설명한 액션 가운데 하나를 선택하면, 기존 벡터 맵에 레이어를 추가할 수도 있습니다.

GRASS 에서는, 하나의 레이어 안에 모든 도형 유형을 (점, 선 및 면을) 구조화시킬 수 있습니다. GRASS 가 위상 벡터 모델을 사용하므로, 새 GRASS 벡터 생성 시 도형 유형을 선택할 필요가 없기 때문입니다. QGIS 에서 셰이프파일을 생성하는 것과는 다릅니다. 왜냐하면 셰이프파일은 단순 객체 벡터 모델을 사용하기 때문입니다. (새 벡터 레이어 생성 을 참조하세요.)

## 22.12 GRASS 벡터 레이어 디지타이즈 및 편집하기

표준 QGIS 다지타이즈 작업 도구를 사용해서 GRASS 벡터 레이어를 디지타이즈할 수 있습니다. 하지만 다음 사실들 때문에 사용자가 알고 있어야 할 몇몇 특수성 또한 존재합니다:

- GRASS 위상 모델 대 QGIS 단순 객체
- GRASS 모델의 복잡성
  - 단일 맵 안의 다중 레이어
  - 단일 맵 안의 멀티 도형 유형
  - 다중 레이어에서 다중 객체가 도형을 공유

다음 문단부터 이 특수성들을 설명하겠습니다.

**저장, 변경 사항 무시, 실행 취소, 재실행**

> **경고**
>
> 편집 작업 동안 이루어진 모든 변경 사항은 즉시 벡터 맵과 관련 속성 테이블에 작성됩니다.

각 편집 작업마다 변경 사항을 작성하지만, 실행 취소/재실행할 수 있고 또는 편집 작업 종료 시 모든 변경 사항을 무시할 수도 있습니다. 실행 취소 또는 변경 사항 무시 기능을 이용하는 경우, 벡터 맵과 속성 테이블에 원본 상태를 다시 작성합니다.

이런 습성에는 다음과 같은 두 가지 중요한 이유가 있습니다:

- GRASS 벡터의 본질은 사용자가 본인의 작업을 원하고 있고 작업이 갑자기 중단되는 경우 (예를 들어 정전 시) 변경 데이터를 저장한 상태인 편이 낫다는 확신에서 나오기 때문입니다.
- 위상 데이터를 효과적으로 편집하려면 위상적 정확성에 관한 정보를 가시화해야 하는데, 변경 사항을 GRASS 벡터 맵에 작성하는 경우에만 맵에서 그런 정보를 얻을 수 있기 때문입니다.

**툴바**

GRASS 레이어를 편집하는 경우‘디지타이즈 작업 툴바’에서 다음과 같은 몇몇 특정 도구를 사용할 수 있습니다:

표 22.1: GRASS Digitizing Tools

| 아이콘 | 도구 | 목적 |
|---|---|---|
|  | New Point | 새 포인트를 디지타이즈합니다. |
|  | New Line | 새 라인을 디지타이즈합니다. |
|  | New Boundary | 새 경계를 디지타이즈합니다. |
|  | New Centroid | 새 중심 (라벨 위치) 을 디지타이즈합니다. |
|  | New Closed Boundary | 새 닫힌 경계를 디지타이즈합니다. |

**팁**

**GRASS 에서 폴리곤을 디지타이즈하기**

GRASS 에서 폴리곤을 생성하고 싶다면, 먼저 폴리곤의 경계를 디지타이즈합니다. 그리고 닫힌 경계 안에 중심 (라벨 포인트) 을 추가합니다. 그 이유는 위상 벡터 모델은 언제나 폴리곤의 경계가 아니라 중심에 폴리곤의 속성 정보를 링크시키기 때문입니다.

#### 카테고리

'cat'으로 줄여 부르는 일이 많은 카테고리는 일종의 ID 입니다. 이 이름은 GRASS 벡터가 `category` 라는 단일 속성만 보유하던 시절에 시작됐습니다. 카테고리는 도형과 속성 사이의 링크로 사용됩니다. 단일 도형이 여러 카테고리를 보유할 수도 있기 때문에, 서로 다른 레이어들 안에 있는 여러 객체들을 나타낼 수도 있습니다. 현재, QGIS 편집 도구를 사용해서 레이어 하나 당 오직 하나의 카테고리만 할당할 수 있습니다. 새 객체는 유일한 새 카테고리를 자동적으로 할당받게 됩니다. 경계를 제외하고 말이죠. 경계는 보통 면을 형성할 뿐 선형 객체를 타나내지는 않지만, 이후에–예를 들어 다른 레이어에서–경계에 대한 속성을 정의할 수 있습니다.

언제나 현재 편집 중인 레이어에서만 새 카테고리를 생성할 수 있습니다.

QGIS 편집 도구를 통해 카테고리에 더 많은 카테고리를 할당할 수는 없습니다. 그런 데이터는 아마도 다중 객체로 표현되어, 서로 다른 레이어들에 있는 개별 객체인 경우에도 삭제될 수도 있습니다.

#### 속성

현재 편집 중인 레이어의 속성만 수정할 수 있습니다. 벡터 맵이 레이어를 하나 이상 담고 있는 경우, 다른 레이어의 객체는 그 속성을 편집할 수 없다고 경고하기 위해 모든 속성이 `<not editable (layer #)>` 로 설정될 것입니다. 그 이유는 QGIS 가 레이어 하나 당 하나의 고정된 필드 집합만 지원하는 반면 다른 레이어는 다른 필드 집합을 보유하고 있을 수도 있고, 또 보통 보유하고 있기 때문입니다.

도형 원시형에 할당된 카테고리가 없는 경우, 유일한 새 카테고리가 자동적으로 할당되어 해당 도형의 속성을 변경할 때 속성 테이블 안에 새 레코드를 생성합니다.

**팁**

테이블에 있는 속성을 대규모로–예를 들어필드 계산기 이용하기 를 사용해서–업데이트하고 싶은데 사용자가 업데이트하고 싶지 않은, 카테고리가 없는 객체 (주로 경계) 가 존재할 경우, '고급 필터'를 `cat is not null` 로 설정해서 해당 객체를 필터링할 수 있습니다.

#### 편집 스타일

위상 심볼은 위상 데이터를 효과적으로 편집하는 데 필수 불가결합니다. 편집 작업 시작 시, 레이어 상에 특화된 'GRASS 편집'렌더링 작업자를 자동적으로 설정하며 편집 작업 종료 시 원래 렌더링 작업자로 되돌립니다. 레이어

속성 대화창의 *Style* 탭에서 스타일을 사용자 지정할 수도 있습니다. 다른 모든 스타일과 마찬가지로, 스타일을 프로젝트 파일에 저장할 수도 있고 개별 파일에 저장할 수도 있습니다. 스타일을 사용자 지정하는 경우, 이름은 변경하지 마십시오. 왜냐하면 편집 작업을 다시 시작할 때 이름을 사용해서 스타일을 리셋하기 때문입니다.

> **팁**
>
> 레이어 편집 중에 프로젝트 파일을 저장하지 마십시오. 레이어가 변경되지 않았을 경우 아무 의미도 가지지 못하는'편집 스타일'과 함께 레이어를 저장할 것이기 때문입니다.

스타일은 속성 테이블에 `topo_symbol` 이라는 필드로 일시적으로 추가된 위상 정보를 기반으로 합니다. 편집 작업 종료 시 이 필드는 자동적으로 제거됩니다.

> **팁**
>
> 속성 테이블에서 `topo_symbol` 필드를 제거하지 마십시오. 렌더링 작업자가 해당 열에 의존하기 때문에 객체가 보이지 않게 될 것입니다.

#### 스냅

면을 형성하려면, 연결된 경계의 꼭짓점들이 **정확히** 동일한 좌표를 보유하고 있어야 합니다. 맵 캔버스와 벡터 맵이 동일한 좌표계인 경우에만 스냅 도구를 사용해서 동일한 좌표를 부여할 수 있습니다. 그렇지 않은 경우, 벡터 맵 좌표를 맵 캔버스 좌표로 변환하고 다시 그 반대로 변환하는 과정에서 표현 오류 및 좌표계 변환 때문에 좌표가 조금씩 달라질 수도 있습니다.

> **팁**
>
> 편집 작업 시 레이어의 좌표계를 맵 캔버스에도 적용하십시오.

#### 제약 사항

한 번에 동일 벡터 안에 있는 여러 레이어들을 동시 편집할 수는 없습니다. 주된 이유는 단일 데이터 소스에 대해 여러 실행 취소 스택 (stack) 을 처리하는 것이 불가능하기 때문입니다.

리눅스 및 맥 상에서는 한 번에 하나의 GRASS 레이어만 편집할 수 있습니다. 데이터베이스 드라이버를 랜덤한 순서로 종료할 수 없는 GRASS 의 버그 때문입니다. 현재 GRASS 개발자들이 이 문제를 해결하고 있는 중입니다.

> **팁**
>
> **GRASS 편집 권한**
>
> 사용자가 편집하고자 하는 GRASS `MAPSET` 의 소유자여야만 합니다. 사용자의 소유가 아닌 `MAPSET` 에 있는 데이터 레이어를 편집하는 것은, 사용자가 쓰기 권한을 가지고 있더라도 불가능합니다.

## 22.13 GRASS 영역 도구

GRASS 에서 래스터 레이어를 작업하는 경우 (공간 작업창을 설정하는) 영역 (region) 정의가 중요합니다. 벡터 분석은 기본적으로 설정한 어느 영역 정의에도 제약을 받지 않습니다. 그러나 새로 생성한 모든 래스터는 원본의 범위 및 해상도에 상관없이 현재 정의된 GRASS 영역의 공간 범위 및 해상도를 적용받게 됩니다. 현재 GRASS 영역은 `$LOCATION/$MAPSET/WIND` 파일에 저장되는데, 이 파일은 동서남북 방향의 경계, 열 및 행의 개수, 수직 및 수평 공간 해상도를 정의합니다.

Display current GRASS region 버튼을 통해 QGIS 맵 캔버스에서 GRASS 영역의 가시성을 켜고 끌 수 있습니다.

'GRASS Tools' 도킹 위젯에 있는 'Region' 탭에서 GRASS 영역을 수정할 수 있습니다. 새 영역의 경계 및 해상도를 입력한 다음 *Apply* 버튼을 클릭하십시오. *Select the extent by dragging on canvas* 버튼을 클릭하면 QGIS 맵 캔버스 위에 마우스로 직사각형을 클릭 & 드래그해서 새 영역의 범위를 대화형으로 선택할 수 있습니다.

`g.region` GRASS 모듈은 사용자의 래스터 분석 작업을 위해 적절한 영역 범위 및 해상도를 정의할 수 있는 훨씬 많은 파라미터를 제공하고 있습니다. *GRASS 툴박스* 를 통해 이 파라미터들을 사용할 수 있습니다.

## 22.14 GRASS 툴박스

Open GRASS Tools GRASS 툴박스는 선택한 GRASS `LOCATION` 및 `MAPSET` 안에 있는 데이터를 작업할 수 있는 GRASS 모듈 기능을 제공하고 있습니다. GRASS 툴박스를 이용하려면, 먼저 사용자가 쓰기 권한을 보유하고 있는 (사용자가 해당 `MAPSET` 을 생성한 경우 보통 당연히 보유하고 있습니다) `LOCATION` 및 `MAPSET` 을 열어야 합니다. 분석 작업 도중에 새로 래스터 및 벡터 레이어를 생성하는 경우 현재 선택 중인 `LOCATION` 및 `MAPSET` 에 작성해야 하기 때문입니다.

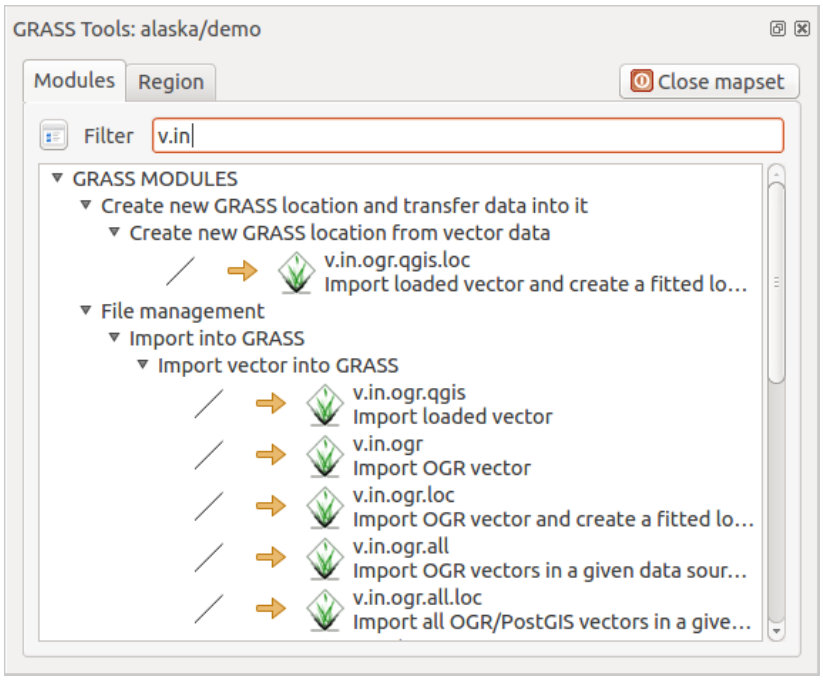



그림 22.3: GRASS 툴박스 및 모듈 트리

### 22.14.1 GRASS 모듈로 작업하기

The GRASS plugin provides a user-friendly toolbox with about 200 of the available GRASS modules and functionalities.

GRASS 위키 사이트 에서, QGIS 3.44 버전의 GUI 툴박스를 통해 사용할 수 있는 GRASS 모듈의 완전한 목록을 찾아볼 수 있습니다.

GRASS 툴박스의 내용을 사용자 지정할 수도 있습니다. 이 과정은*GRASS 툴박스 사용자 지정하기* 에서 설명하고 있습니다.

그림 22.3 에서 알 수 있는 바와 같이, 테마 별로 그룹화된 *Modules Tree* 또는 검색 가능한 *Modules List* 탭을 통해 알맞은 GRASS 모듈을 찾을 수 있습니다.

모듈의 아이콘을 클릭하면, 툴박스 대화창에 새 탭이 추가될 것입니다. 이 탭은 *Options*, *Output* 및 *Manual* 이라는 하위 탭 3 개를 보유하고 있습니다.

**옵션**

*Options* 탭은 보통 QGIS 맵 캔버스에 가시화된 래스터 및 벡터 레이어를 선택하고 해당 모듈에 특화된 파라미터를 입력해서 모듈을 실행할 수 있는 간소화된 대화창을 담고 있습니다.

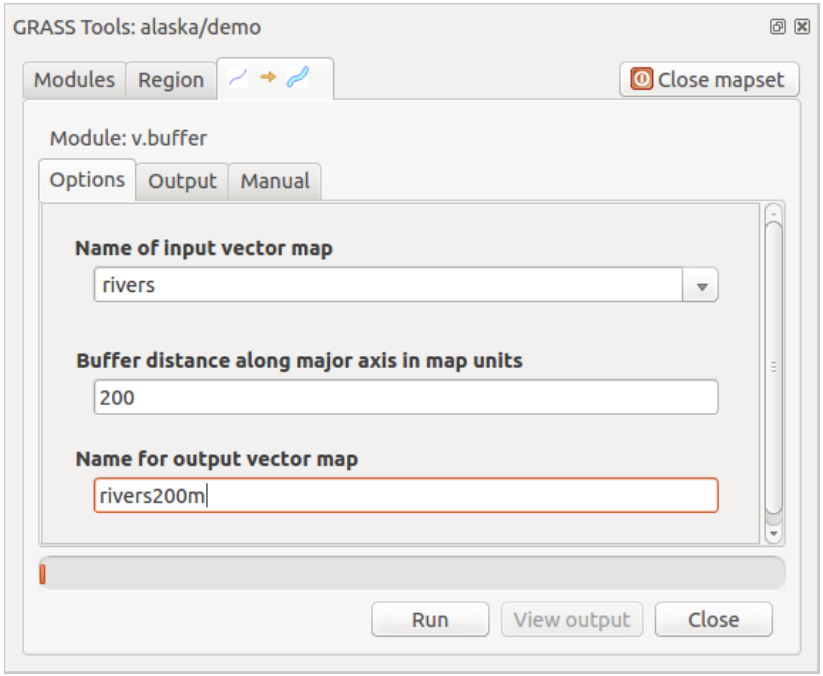



그림 22.4: GRASS 툴박스: 모듈 옵션

이 대화창을 간소하게 유지하기 위해, 대화창에 표시되는 모듈 파라미터들이 완전하지 않은 경우가 많습니다. 심화 모듈 파라미터 및 플래그를 사용하고 싶다면, GRASS 셸을 실행해서 명령 줄에서 모듈을 구동해야 합니다.

QGIS 1.8 버전부터 *Options* 탭의 간소화된 모듈 대화창 하단에 **[Show Advanced Options]** 버튼을 지원하고 있습니다. 현재에는 활용법을 보여주기 위한 목적으로 `v.in.ascii` 모듈에만 추가돼 있지만, QGIS 향후 버전에서는 GRASS 툴박스에 있는 좀 더 많은 또는 모든 모듈에 추가될 수도 있습니다. 이 버튼을 누르면 GRASS 셸로 전환할 필요없이 대화창에서 완전한 GRASS 모듈 옵션을 사용할 수 있습니다.

**Output**

*Output* 탭은 모듈의 산출물 상태에 관한 정보를 표시합니다. *Run* 버튼을 클릭했을 때, 모듈이 *Output* 탭으로 전환되어 사용자가 분석 과정에 관한 정보를 볼 수 있습니다. 모든 과정이 무사히 완료되면, 마지막으로 `Successfully finished` 메시지가 표시될 것입니다.

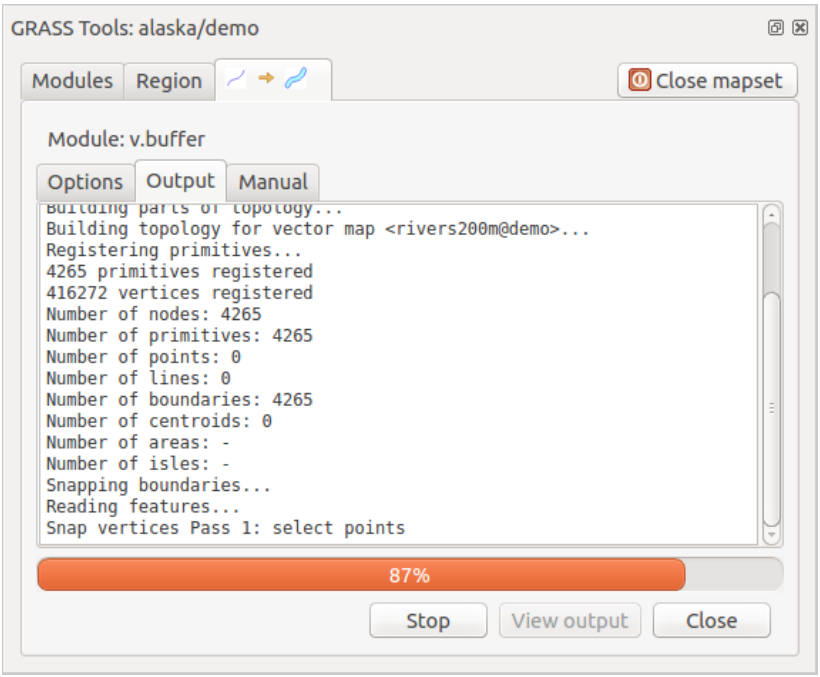



그림 22.5: GRASS 툴박스: 모듈 산출물

#### 사용자 설명서

*Manual* 탭은 GRASS 모듈의 HTML 도움말 페이지를 표시합니다. 심화 모듈 파라미터 및 플래그를 확인하는 데, 또는 해당 모듈의 목적에 관한 심도 있는 지식을 얻기 위해 이 탭을 사용할 수 있습니다. 각 모듈의 사용자 설명서 페이지 맨 끝에, *Main Help index*, *Thematic index* 및 *Full index* 를 가리키는 링크들이 있습니다. 이 링크들은 각각 `g.manual` 모듈이 제공하는 주요 도움말 색인, 테마 색인, 전체 색인으로 연결됩니다.

> **팁**
>
> **결과를 즉시 표시하기**
>
> 사용자의 맵 캔버스에 계산 결과를 즉시 표시하고 싶다면, 모듈 탭 하단에 있는 **[View Output]** 버튼을 누르면 됩니다.

### 22.14.2 GRASS 모듈 예제

다음 예제들이 일부 GRASS 모듈의 강력한 기능을 보여줄 것입니다.

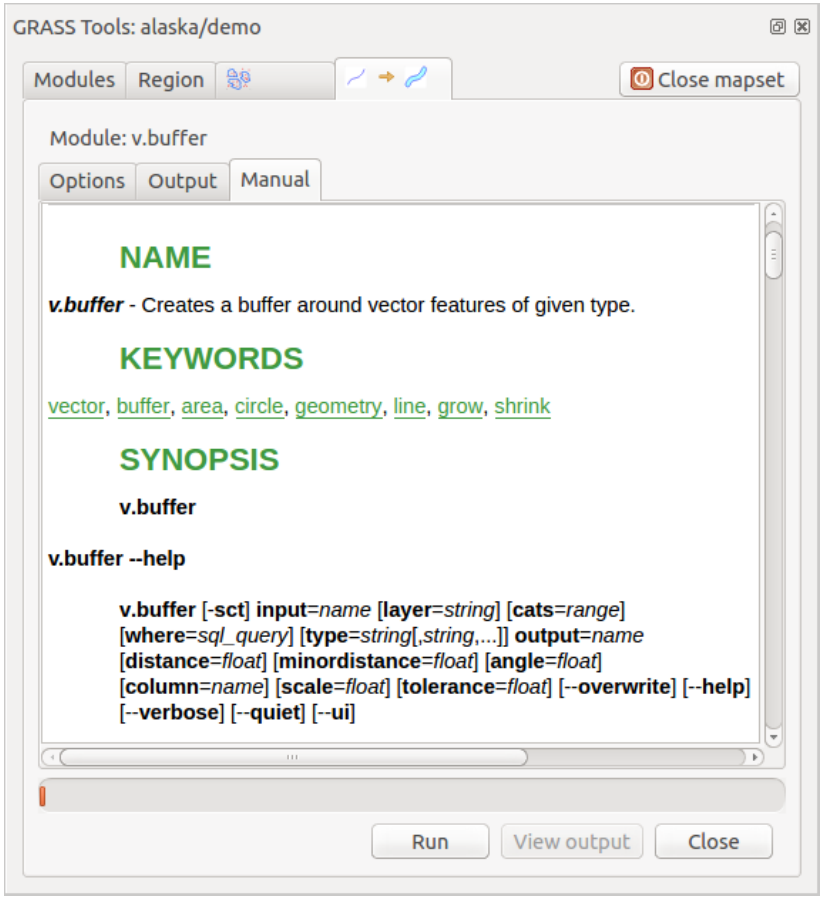



그림 22.6: GRASS 툴박스: 모듈 사용자 설명서

### 등고선 생성하기

첫 번째 예제는 표고 래스터 (DEM) 로부터 벡터 등고선 맵을 생성하는 것입니다. 이 예시에서는 사용자가*GRASS 로케이션에 데이터 가져오기* 에서 설명한 대로 `LOCATION` 'alaska'를 설정했다고 가정합니다.

- 먼저,  Open mapset 아이콘을 클릭하고 `LOCATION` 'alaska'를 선택해서 로케이션을 여십시오.
- 이제  Open GRASS tools 아이콘을 클릭해서 툴박스를 여십시오.
- 도구 범주 목록에서 *Raster* ▸ *Surface Management* ▸ *Generate vector contour lines* 를 더블클릭하십시오.
- 이제 `r.contour` 모듈을 한 번 클릭하면 앞에서 설명한 것처럼 (*GRASS 모듈로 작업하기* 참조) 모듈 대화창이 열릴 것입니다.
- *Name of input raster map* 옵션에 `gtopo30` 이라고 입력하십시오.
- *Increment between Contour levels*  옵션의 값을 100 으로 입력하십시오. (이렇게 하면 등고선을 100 미터 간격으로 생성할 것입니다.)
- *Name for output vector map* 옵션에서 이름을 `ctour_100` 으로 입력하십시오.
- *Run* 버튼을 눌러 공간 처리 과정을 시작하십시오. 산출물 창에 `Successfully finished` 메시지가 나타날 때까지 잠시 기다렸다가 *View Output* 버튼을 클릭한 다음 *Close* 버튼을 클릭하십시오.

알래스카가 넓은 영역이기 때문에, 전체를 표시하는 데 시간이 좀 걸릴 겁니다. 렌더링이 종료된 다음, 벡터 속성 대화창 에서 설명한 대로 레이어 속성 대화창을 열어서 라인 색상을 변경하면 표고 레스터 위에 중첩한 등고선이 명확하게 보이도록 할 수 있습니다.

그 다음, 알래스카 중심부에 있는 좁은 산지 지역으로 확대하십시오. 가까이 확대해보면 등고선이 급격히 꺾여 있다는 사실을 알 수 있을 겁니다. GRASS 는 벡터 맵의 전체 형태를 유지하면서도 살짝 수정할 수 있는 `v.generalize` 모듈을 제공하고 있습니다. 이 모듈은 서로 다른 목적에 따라 서로 다른 알고리즘을 여럿 사용합니다. 몇몇 (예를 들면 더글러스-포이커 및 꼭짓점 감축) 알고리즘은 꼭짓점 가운데 일부를 제거해서 라인을 단순화시킵니다. 산출되는 벡터는 더 빨리 불러올 수 있을 것입니다. 사용자가 너무나 상세한 벡터를 가지고 있지만, 매우 소축척의 맵을 생성하기 때문에 세밀한 사항이 필요없는 경우 이런 공간 처리를 하는 것이 유용합니다.

> **팁**
>
> **단순화 도구**
>
> QGIS 도 GRASS `v.generalize` 모듈의 더글러스-포이커 알고리즘과 동일하게 작동하는 *Vector* ▸ *Geometry Tools* ▸ *Simplify geometries* 도구를 보유하고 있다는 사실을 기억하십시오.

하지만, 이 예제의 목적은 다릅니다. `r.contour` 모듈이 생성한 등고선의 날카로운 각도를 매끄럽게 다듬어야 합니다. `v.generalize` 모듈의 알고리즘 가운데, 체이킨 (또는 에르미트 스플라인) 알고리즘이 바로 이런 처리를 할 수 있습니다. 다만 이런 알고리즘들이 벡터에 더 많은 꼭짓점들을 **추가** 할 수 있다는 사실을 잘 알고 있어야 합니다. 이런 경우 벡터를 불러오는 데 시간이 더 걸릴 수도 있습니다.

- GRASS 툴박스를 열고 *Vector* ▸ *Develop map* ▸ *Generalization* 범주를 더블클릭한 다음, `v.generalize` 모듈을 클릭해서 모듈 옵션 대화창을 여십시오.
- *Name of input vector* 옵션에'ctour_100'옵션이 표시돼 있는지 확인하십시오.
- 알고리즘 목록에서, `Chaiken's` 를 선택하십시오. 다른 모든 옵션은 기본값으로 내버려두고, 마지막 행으로 스크롤해서 *Name for output vector map* 란에'ctour_100_smooth'를 입력한 다음 *Run* 버튼을 클릭하십시오.
- 산출물 창에 `Successfully finished` 메시지가 나타날 때까지 잠시 기다렸다가 *View Output* 버튼을 클릭한 다음 *Close* 버튼을 클릭하십시오.

- 래스터 배경에서 벡터가 더 잘 보이도록 그리고 원본 등고선과 대조되도록 벡터의 색상을 변경할 수도 있습니다. 새 등고선이 원본의 전체 형태를 잘 보전하면서도 원본보다 더 매끄러워졌다는 사실을 알 수 있습니다.




그림 22.7: 벡터 맵을 매끄럽게 만드는 GRASS *v.generalize* 모듈

**팁**

**r.contour 의 다른 활용법**

동등한 다른 상황에서도 앞에서 설명한 처리 과정을 사용할 수 있습니다. 예를 들어 강수량 데이터 래스터 맵이 있을 경우, 동일한 방법으로 등강수량 (isohyetal) 선 벡터 맵을 생성할 수 있습니다.

### 음영기복도 3 차원 효과 생성하기

표고 레이어를 표시하고 맵에 3 차원 효과를 주는 데 쓰이는 몇 가지 방법들이 있습니다. 앞에서 보인 등고선을 활용하는 것도 지형도 생산에 자주 쓰이는 인기 있는 방법 가운데 하나입니다. 3 차원 효과를 줄 수 있는 또다른 방법이 바로 음영기복도입니다. 먼저 DEM(표고) 래스터에서 각 셀의 경사 (slope) 및 경사방향 (aspect) 을 계산한 다음, 하늘에 있는 태양의 위치를 시뮬레이션하고 각 셀에 그 반사율 (reflectance) 을 적용해서 음영기복도 효과를 생성합니다. 즉, 태양을 향해 있는 경사는 밝아지고 태양을 등지고 있는 경사는 (그림자가 져서) 어두워집니다.

- `gtopo30` 표고 래스터를 불러오는 것으로 이 예제를 시작하겠습니다. GRASS 툴박스를 연 다음, 래스터 범주에 있는 *Spatial analysis* ▷ *Terrain analysis* 를 더블클릭해서 지형 분석 대화창을 여십시오.
- 그리고 `r.shaded.relief` 를 클릭해서 모듈을 여십시오.
- *azimuth angle* 1,00 옵션에서 방위각을 270 에서 315 로 변경하십시오.
- 새 음영기복 래스터의 이름으로 `gtopo30_shade` 를 입력한 다음 *Run* 버튼을 누르십시오.
- 공간 처리 과정이 완료되면, 음영기복 래스터를 맵에 추가하십시오. 회색조로 표시되는 것을 볼 수 있을 것입니다.
- 음영기복도와 함께 `gtopo30` 레이어의 색상을 보려면, 범례에서 음영기복도를 `gtopo30` 맵 아래로 이동시킨 다음 `gtopo30` 레이어의 속성 대화창을 여십시오. *Transparency* 탭에서 투명도 수준을 약 25% 로 설정하십시오.

이제 `gtopo30` 표고 레이어가 색상표 및 투명도가 설정된 상태로 회색조인 음영기복도 **위에** 표시되었을 것입니다. 음영기복도가 어떤 시각 효과를 주는지 보려면, `gtopo30_shade` 레이어를 껐다가 다시 켜보십시오.

#### 벡터 맵에서의 래스터 통계

다음 예제에서는 GRASS 모듈이 어떻게 래스터 데이터를 합치고 벡터 맵에 있는 각 폴리곤에 대한 통계 열들을 추가하는지 보여줄 것입니다.

- 알래스카 데이터를 다시 사용해서, `shapefiles/trees.shp` 파일을 GRASS 로 가져오십시오. (*GRASS 로케이션에 데이터 가져오기* 를 참조하세요.)
- 이제 중간 단계를 처리해야 합니다–가져온 삼림 맵을 (경계 및 중심 둘 다 포함한) 완전한 GRASS 면 벡터로 만들려면 중심을 추가해야만 합니다.
- 툴박스에서 *Vector → Manage features* 를 선택한 다음, `v.centroids` 모듈을 여십시오.
- *Name for output vector map* 옵션에서 이름을 `forest_areas` 로 입력한 다음 모듈을 실행하십시오.
- 이제 `forest_areas` 벡터를 불러와서 삼림 유형을–낙엽수림, 상록수림, 혼효수림–서로 다른 색상으로 표시하십시오. 레이어 속성 대화창의 *Symbology* 탭에서 *Legend type* 옵션을'Unique value'로 선택하고 *Classification field* 옵션을'VEGDESC'으로 설정하십시오. (심볼 속성 절의 *Symbology* 탭에 관한 설명을 참조하세요.)
- 그리고 GRASS 툴박스 대화창을 다시 열고 다른 맵으로 *Vector → Vector update* 를 여십시오.
- `v.rast.stats` 모듈을 클릭한 다음 `gtopo30` 및 `forest_areas` 를 입력하십시오.
- 이제 추가 파라미터 하나만 더 설정하면 됩니다. *column prefix* 옵션에 `elev` 를 입력한 다음 *Run* 버튼을 클릭하십시오. 이 작업은 수많은 계산을 해야 하는 작업이기 때문에 아주 오래 (아마도 2 시간 정도) 걸릴 것입니다.
- 마지막으로, `forest_areas` 레이어의 속성 테이블을 열고 각 삼림 폴리곤에 대해 `elev_min`, `elev_max`, `elev_mean` 등을 포함한 새 열들 여럿이 추가된 것을 확인하십시오.

### 22.14.3 GRASS 툴박스 사용자 지정하기

GRASS 툴박스에 거의 모든 GRASS 모듈을 추가할 수 있습니다. 매우 단순한 XML 파일로 툴박스 내부에서 모듈의 모습 및 파라미터를 환경 설정할 수 있으며, 이 파일을 파싱하기 위한 XML 인터페이스도 있습니다.

`v.buffer` (v.buffer.qgm) 모듈을 생성하기 위한 예시 XML 파일의 내용은 다음과 같습니다:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE qgisgrassmodule SYSTEM "http://mrcc.com/qgisgrassmodule.dtd">

<qgisgrassmodule label="Vector buffer" module="v.buffer">
        <option key="input" typeoption="type" layeroption="layer" />
        <option key="buffer"/>
        <option key="output" />
</qgisgrassmodule>
```

The parser reads this definition and creates a new tab inside the Toolbox when you select the module.

CHAPTER 23

# QGIS 공간 처리 프레임워크

## 23.1 소개

이 장에서는 QGIS 공간 처리 프레임워크를 소개합니다. QGIS 공간 처리 프레임워크는 QGIS 에서 내장 알고리즘과 제 3 자 알고리즘을 호출하는 데 사용되는 지리 공간 처리 환경으로, 사용자가 공간 분석 업무를 더 생산적으로 더 쉽게 완수할 수 있도록 해줍니다.

공간 처리 프레임워크는핵심 플러그인 으로, 기본으로 설치되지만 사용자가 활성화시켜야 합니다:

1. *Plugins* → *Manage and install plugins*…메뉴 항목을 클릭하십시오.
2. 왼쪽에 있는 *Installed* 탭을 선택하십시오.
3.  *Processing* 항목 옆에 있는 체크박스에 체크하십시오.
4. 대화창을 닫으십시오.

   이제 최상위 메뉴에서 *Processing* 메뉴를 사용할 수 있습니다. 이 메뉴를 통해 공간 처리 프레임워크의 주요 요소에 접근할 수 있습니다.

다음 단락부터 이 프레임워크의 그래픽 요소들을 어떻게 사용하는지 그리고 각 요소들을 어떻게 최대한 사용할 수 있는지를 살펴볼 것입니다.

프레임워크 GUI 에는 4 개의 기본 항목들이 있습니다. 이 요소들은 서로 다른 목적을 위해 알고리즘을 실행하는 데 사용됩니다. 어떤 분석을 수행하는가 및 각 사용자와 프로젝트의 특성에 따라 어떤 도구를 사용할지를 결정합니다. *Processing* 메뉴에서 (곧 설명하겠지만, 툴박스 또는 알고리즘 실행 대화창에서 호출되는 배치 공간 처리 인터페이스를 제외한) 모든 도구들을 선택할 수 있습니다. (이 메뉴의 항목들이 4 개가 넘는다는 것을 볼 수 있을 겁니다. 여기에서 설명하는 항목 외 나머지는 알고리즘을 실행하기 위한 것이 아닙니다. 이 장의 다음 부분에서 설명하겠습니다.)

- *Toolbox*: GUI 의 메인 항목으로, 단일 알고리즘을 실행하거나 해당 알고리즘에 기반한 배치 프로세스 (batch process) 를 실행합니다.

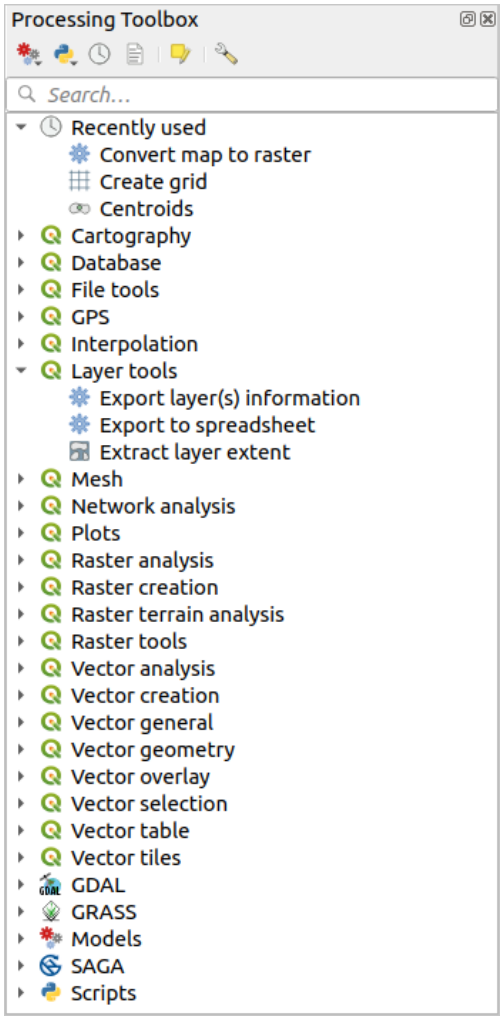



그림 23.1: 공간 처리 툴박스

- *Model Designer*: 모델 설계자 (model designer) 의 그래픽 인터페이스를 통해 여러 알고리즘을 결합시켜, 여러 하위 프로세스를 이용하는 단일 프로세스를 생성하는 작업 흐름을 정의합니다.

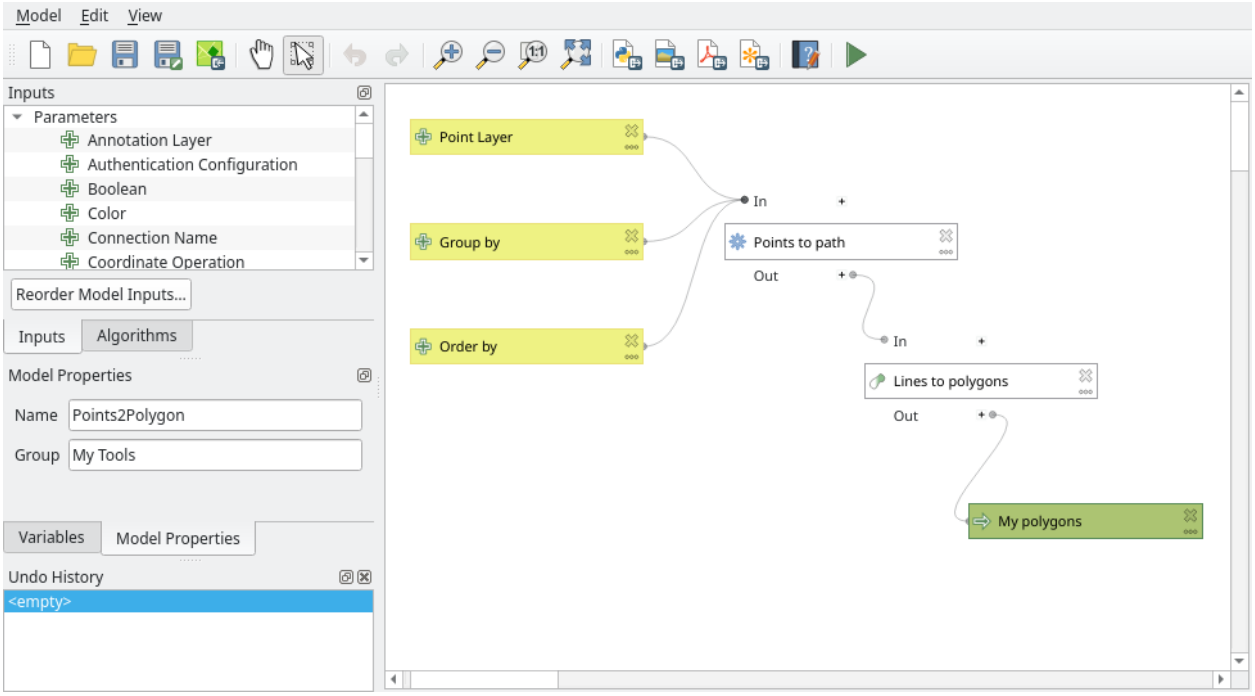



그림 23.2: 공간 처리 모델 설계자

- *History* 관리자: 이력 파일에 앞에서 언급한 항목을 사용해서 수행된 모든 액션을 저장하고, 나중에 이력 관리자를 통해 쉽게 재수행할 수 있습니다.
- *Batch Processing* 인터페이스: 이 인터페이스를 통해 배치 프로세스를 실행해서 여러 데이터셋에 대해 단일 알고리즘을 수행하는 과정을 자동화할 수 있습니다.

다음 절부터 각 항목을 자세히 살펴볼 것입니다.

## 23.2 공간 처리 프레임워크 환경 설정

공간 처리 옵션 메뉴 (*Settings* *Options* *Processing* tab) 에서 알고리즘 작동 방식을 환경 설정할 수 있습니다. 환경 설정 파라미터들은 대화창의 왼쪽에서 선택할 수 있는 개별 그룹으로 구분돼 있습니다.

### 23.2.1 일반 (General) 그룹

*General* 그룹은 알고리즘 대화창은 물론 입력 또는 출력 파라미터들이 어떤 습성을 가져야 할지를 제어하는 기본 설정들을 담고 있습니다. 하지만 이 설정들의 일부는알고리즘 실행 별로, 또는개별 파라미터 *(individual parameter)* 때문에 알고리즘 수준에서 무시될 수 있습니다.

- *Default output raster layer extension*: 기본 산출 래스터 확장자의 기본값은 `tif` 입니다.
- *Default output vector layer extension*: 기본 산출 벡터 레이어 확장자의 기본값은 `gpkg` 입니다.
- *Invalid features filtering*: 알고리즘 실행 시 무결하지 않은 피처를 필터링합니다:

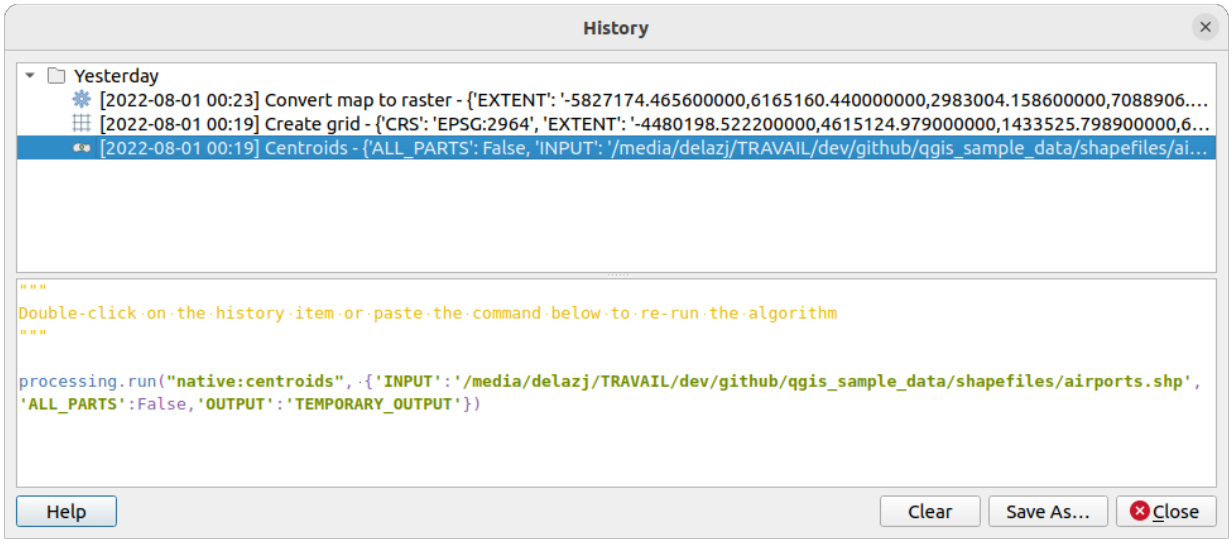



그림 23.3: 공간 처리 이력

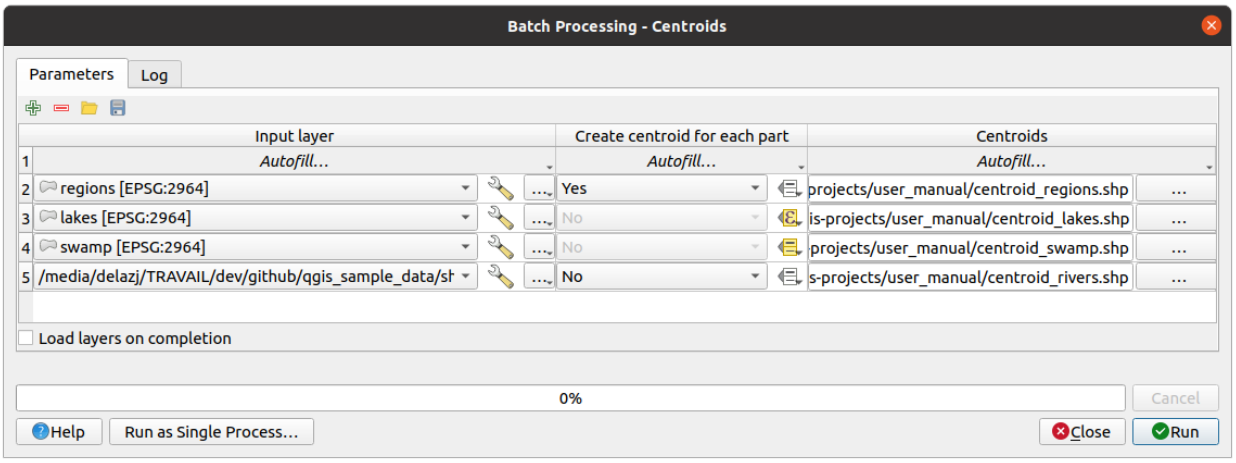



그림 23.4: 배치 공간 처리 인터페이스

  - *Do not filter (better performance)*: (무결한 그리고 무결하지 않은 도형을 가진) 모든 피처를 처리하지만, 도형의 무결성이 작업에 어떤 영향을 미치는지에 따라 결과에 오류가 있을 수도 있습니다.
  - *Skip (ignore) features with invalid geometries*: 사용자 데이터셋 가운데 (무결한 도형을 가진 피처의) 부분 집합만 처리할 것입니다.
  - *Stop algorithm execution when a geometry is invalid*: 알고리즘이 전체 레이어를 처리하게 하려면 무결하지 않은 도형을 추적하고 수정해야 합니다. 무결성 점검하기 또는도형 고치기 같은 알고리즘을 사용하면 무결하지 않은 도형을 추적하고 수정할 수 있습니다.

  알고리즘 런타임 시 입력물별로 *Invalid features filtering* 설정을 무시할 수 있습니다.

- *Keep dialog open after running algorithm*: 알고리즘 실행이 완료되어 그 산출 레이어를 QGIS 프로젝트로 불러오고 나면, 알고리즘 대화창이 닫힙니다. 대화창이 (알고리즘을 다른 파라미터 값으로 다시 실행하기 위해, 또는 로그 탭에 작성된 산출물을 좀 더 자세히 확인하기 위해) 계속 열려 있기를 바란다면, 이 옵션을 활성화하십시오.
- *Max Threads*: 스레드 최대 개수를 설정합니다.
- *Output folder* for non temporary outputs: If no folder path is provided for the Processing execution outputs, this is the folder in which they will be saved. Default is `processing` under the user home directory (e.g., `C:\Users\<YOURUSERNAME>\processing\` on Windows).
- *Override temporary output folder path*: 임시 산출물은 기본적으로 시스템 상에 있는 `tmp` 폴더에 저장됩니다. 이 옵션을 통해 저장소의 다른 위치를 설정할 수 있습니다.
- *Pre-execution script* 및 *Post-execution script*: 이 파라미터들은 공간 처리 스크립트 작업 기능을 통해 작성된 스크립트를 가리킵니다. 스크립트 작업 및 콘솔에 관한 부분에서 이 옵션들을 설명하고 있습니다.
- *Prefer output filename for layer names*: 알고리즘 자체가 해당 알고리즘이 생성하는 각 산출 레이어의 이름을 정의합니다. 몇몇 경우 고정된 이름을 사용할 수도 있는데, 어떤 입력 레이어를 사용하든 상관없이 동일한 산출 레이어 이름을 사용할 것이라는 뜻입니다. 다른 경우엔 입력 레이어의 이름 또는 알고리즘 실행에 사용되는 몇몇 파라미터에 따라 산출 레이어 이름이 달라지기도 합니다. 이 옵션을 활성화하면 대신 산출 파일명에서 레이어 이름을 가져올 것입니다. 산출물이 임시 파일로 저장되는 경우, 이 임시 파일의 이름은 다른 기존 임시 파일과의 충돌을 피하려는 목적으로 아주 길고 의미도 없다는 점을 기억해두십시오.
- *Results group name*: *Layers* 패널에서 모든 공간 처리 결과 레이어들을 한 그룹으로 모으고 싶다면, 이 파라미터에 그룹명을 설정하십시오. 기존 그룹을 사용해도 됩니다. 설정한 그룹에 QGIS 가 모든 산출 레이어를 추가할 것입니다. 이 파라미터는 기본적으로 비어 있기 때문에, 모든 산출 레이어가 *Layers* 패널에서 알고리즘 실행시 활성화돼 있는 항목에 따라 서로 다른 위치에 추가됩니다. 알고리즘 대화창에서 *Open output file after running algorithm* 옵션을 체크한 경우 *Layers* 패널에만 산출 레이어를 불러올 것이라는 점을 기억하십시오.
- *Show algorithms with known issues*: QGIS 는 기본적으로 (일반적으로 제 3 자 제공자의) 망가진 알고리즘을 출력하지 않습니다. 이 옵션을 체크하는 경우 공간처리 툴박스에서 경고 아이콘 및 해당 알고리즘에 문제가 있다는 사실을 설명하는 툴팁과 함께 표시되고, 해당 알고리즘을 사용할 수 있습니다. 이런 알고리즘은 사용자의 책임 하에 사용하십시오.
- *Show feature count for output vector layers*: 특정 데이터 포맷의 경우 피처 개수를 계산하는 데 시간이 걸릴 수도 있기 때문에 이 옵션을 기본적으로 비활성화되어 있습니다.
- *Show layer CRS definition in selection boxes*: 콤보박스에 레이어 좌표계 정의를 표시합니다.
- *Show tooltip when there are disabled providers*: 비활성화된 제공자가 있는 경우 툴팁을 표시합니다.
- *Style for line layers*, *Style for point layers*, *Style for polygons layers* 및 *Style for raster layers* 옵션은 산출 레이어의 (그러니까 공간 처리 알고리즘이 생성한 레이어의) 기본 렌더링 스타일을 설정하는 데 사용됩니다. QGIS 를 통해 사용자가 원하는 스타일을 생성하고 파일로 저장한 다음, 알고리즘이 사용할 수 있도록 설정에서 해당 파일을 가리키는 경로를 입력하기만 하면 됩니다. 공간 처리 프레임워크가 레이어를 불러와서 QGIS 맵 캔버스에 추가할 때마다 해당 스타일로 렌더링할 것입니다.

각 알고리즘별로 그리고 해당 알고리즘의 각 산출물별로 렌더링 스타일을 개별적으로 환경 설정할 수 있습니다. 툴박스에서 알고리즘 명을 오른쪽 클릭한 다음 *Edit rendering styles for outputs* 메뉴 옵션을 선택하십시오. 다음 그림과 같은 대화창이 열릴 겁니다.

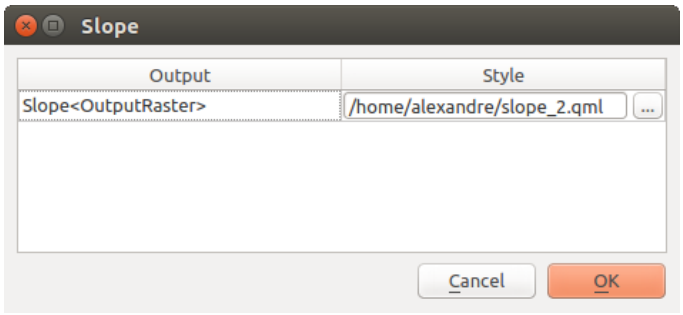



그림 23.5: 렌더링 스타일

각 산출물별로 사용자가 원하는 스타일 파일 (`.qml`) 을 선택한 다음 *OK* 버튼을 클릭하십시오.

- *Warn before executing if parameter CRS's do not match*: 기본적으로 QGIS 네이티브 알고리즘들은 (예를 들어 *Menus* → *QGIS (native C++)* 그룹에 목록화된 알고리즘들은) 실행 전에 입력 레이어들을 명백하게 첫 번째 레이어의 좌표계로 재투영합니다. 이 옵션을 체크하면, 입력 레이어들의 좌표계가 동일하지 않을 경우 재투영을 지원하지 않는 다른 도구들로부터 알림을 받을 수 있습니다. 제 3 자 제공자들은 상관없습니다.

### 23.2.2 메뉴 그룹

*Menus* 그룹에서 (내장된 또는 플러그인이 제공하는) 알고리즘, 스크립트 또는 모델을 (공간 처리 툴박스와 함께) 전용 메뉴 또는 툴바를 통해 사용할 수 있게 해야 할지를 제어합니다. 각 제공자의 각 항목에 대해 다음을 설정할 수 있습니다:

- *Add button in toolbar*: *Processing Algorithms* 툴바에서 사용할 수 있게 합니다.
- 알고리즘에 *Icon* 을 할당합니다.
- set a *Menu path*: the algorithm will then be available through an existing or a custom menu, e.g. `Vect&or/MyTopAlgorithms` will add a shortcut in a *MyTopAlgorithms* sub-menu of the default *Vector* menu.

이런 설정들을 적용하려면 QGIS 를 재시작하십시오. *Reset to defaults* 옵션을 사용해서 언제라도 사용자 변경 사항을 초기화할 수 있습니다.

### 23.2.3 모델 (Models) 및 스크립트 (Scripts) 그룹

In the *Models* and *Scripts* blocks, a default path, set under the active *User profile* folder, is provided for storing models and scripts respectively. As for the other options, you can modify the path, e.g., to lead to a remote or shared folder for all your users.

### 23.2.4 알고리즘 제공자 (Providers) 그룹

You will also find a block for algorithm  *Providers*. This is the place where installed providers expose their settings.

By default, QGIS is installed with two algorithm providers:

- *GDAL* whose algorithms you may *Activate* (or not) in this dialog. Read more about the *GDAL algorithms*.
- *GRASS* whose algorithms require the installation of the *GRASS GIS Processing Provider* from the *Plugins Manager*. You could then set some dedicated settings such as:
  -  *For raster layers, use r.external (faster) instead of r.in.gdal*
  -  *For vector layers, use v.external (faster) instead of v.in.ogr*
  - *Location of GRASS docs*: by default, and depending on your OS, QGIS utilizes a hardcoded list of paths for finding the local GRASS documentation and opening it whenever you hit the *Help* button of a GRASS algorithm dialog. Filling this option will override the default path, allowing you to use a custom location, or remote documentation.
  -  *Log console output*
  -  *Log execution commands*

Installed plugins that provide Processing algorithms may also have their provider listed in this group, with custom settings.

## 23.3 툴박스

*Processing Toolbox* 는 공간 처리 GUI 의 주요 항목으로, 사용자가 일상적으로 사용하게 될 가능성이 높습니다. 사용할 수 있는 모든 **알고리즘** 을 서로 다른 제공자, 도구 집합을 확장하기 위해 추가할 수 있는 **모델** 그리고 **스크립트** 로 그룹화한 목록을 표시합니다. 단일 프로세스이든 동일 알고리즘을 서로 다른 입력물에 대해 여러 번 실행하는 배치 프로세스이든 바로 이 툴박스에서 알고리즘을 실행하게 됩니다.

공간 처리 설정 대화창 에서 이 제공자들을 (비) 활성화할 수 있습니다. 기본적으로, 제 3 자 응용 프로그램에 의존하지 않는 (즉 실행하는 데 QGIS 요소만 요구하는) 제공자만 활성화된 상태입니다. 외부 응용 프로그램이 필요한 알고리즘은 추가 환경 설정이 필요할 수도 있습니다. 제공자 환경 설정에 대해서는 이 사용자 지침서의이후 부분 에서 설명하겠습니다.

툴박스 대화창 상단에는 다음과 같은 작업을 하기 위한 도구들이 있습니다:

-  Models 작업: *Create New Model…*, *Open Existing Model…*및 *Add Model to Toolbox…*
-  Scripts 작업: *Create New Script…*, *Create New Script from Template…*, *Open Existing Script…*및 *Add Script to Toolbox…*
-  History: 이력 패널 열기
-  Results Viewer: 결과물 뷰어 패널 열기
-  Edit Features In-Place: 툴박스의제자리 *(in-place)* 수정 모드 를 켜고 끕니다. 새 레이어를 산출하지 않고, 활성 레이어에 대해서 실행하는 데 적합한 알고리즘만 표시됩니다.
-  Options: 옵션 대화창 열기

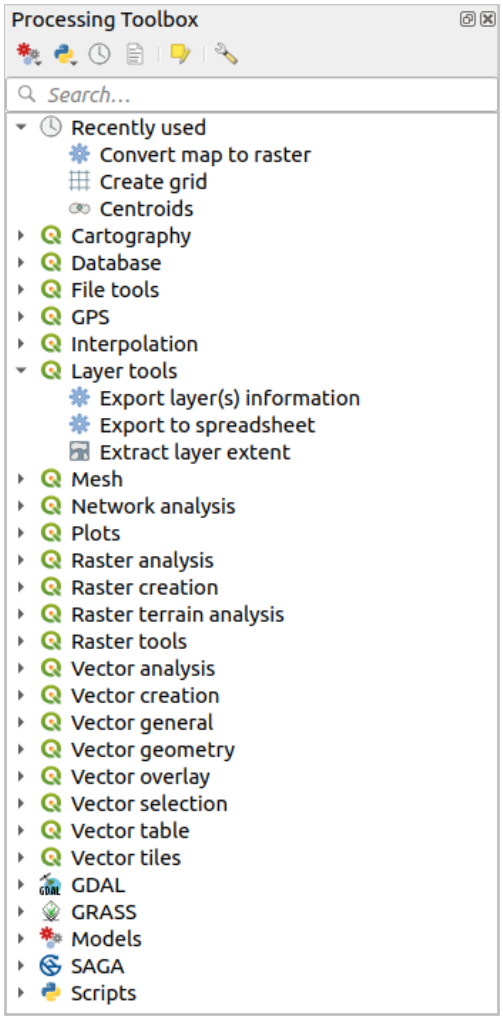



그림 23.6: 공간 처리 툴박스

이 툴바 아래에는 필요한 도구를 쉽게 찾을 수 있게 해주는 Search…란이 있습니다. 이 텍스트란에 어떤 단어 또는 문구라도 입력할 수 있습니다. 사용자가 입력하는 동안, 사용자가 입력한 텍스트 또는 키워드가 포함된 알고리즘, 모델 및 스크립트만 필터링해서 툴박스에 표시된다는 사실을 알 수 있을 겁니다.

The top of the algorithm list shows the most Recently Used tools and the Favorites tools, making it convenient to reexecute them. The Recently Used list is there by default, showing the algorithms you have recently executed. The Favorites list is created when you manually add an algorithm to it.

You can right-click on any algorithm in the toolbox and choose from the contextual menu to:

- *Execute*… to run the algorithm.
- *Execute as Batch Process*… to run the algorithm in batch mode.
- *Edit Rendering Styles for Outputs*… to customize the rendering styles for the algorithm's outputs.
- *Add to Favorites* to add the algorithm to the Favorites list.
- *Remove from Favorites* to remove the algorithm from the Favorites list.

Unlike the Recently Used list, the Favorites list remains static and is not affected by which algorithms were executed.

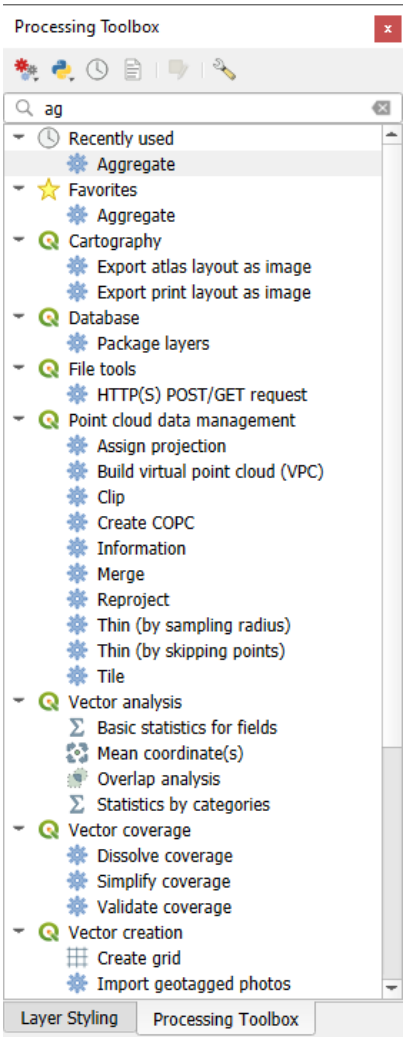



그림 23.7: 검색 결과를 표시하는 공간 처리 툴박스

어느 도구를 실행하려면, 툴박스에 있는 해당 이름을 더블클릭하기만 하면 됩니다.

### 23.3.1 알고리즘 대화창

사용자가 실행하고자 하는 알고리즘의 이름을 더블클릭하고 나면, 다음 그림 23.8 과 비슷한 대화창이 열립니다. (이 그림은 `Centroids` 알고리즘에 해당하는 대화창입니다.) 대화창 제목에는 해당 알고리즘이 속해 있는 그룹 이름이 포함될 것입니다.

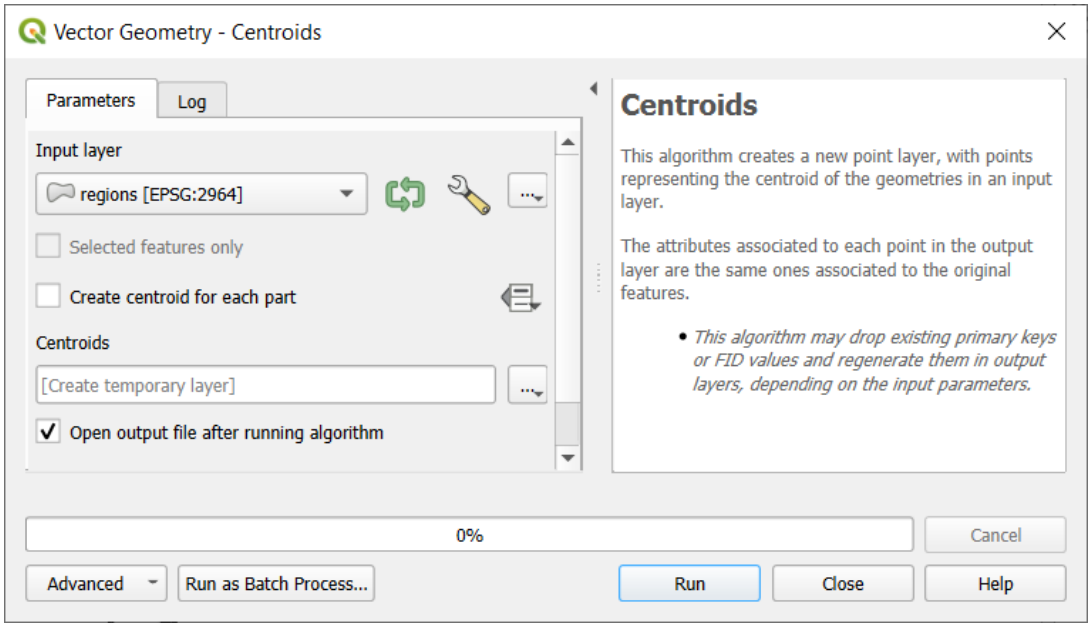



그림 23.8: 알고리즘 대화창 — 파라미터 탭

이 대화창 왼쪽에는 탭이 2 개 (*Parameters* 와 *Log*) 있으며, 오른쪽에는 알고리즘 설명이, 하단에는 몇몇 버튼들이 있습니다.

#### 파라미터 유형

*Parameters* 탭에서 알고리즘을 실행하는 데 필요한 입력값들을 설정할 수 있습니다. 이 탭은 설정해야 할 입력값 및 환경 설정 파라미터 목록을 표시합니다. 물론 실행할 알고리즘의 요구 사항에 따라 목록 내용이 달라지며, 해당 요구 사항을 기반으로 자동적으로 생성됩니다.

> **팁**
>
> **알고리즘 파라미터에 사용자의 고유 기본값을 설정하기**
>
> 일부 파라미터들을 QGIS 설치본에서 가져온 값으로 미리 채운 알고리즘 대화창이 열립니다. 하지만 특정 알고리즘 파라미터에사용자 고유 기본값 을 설정해서 알고리즘 구동 시 사용자 기본값을 사용하게 할 수도 있습니다.

알고리즘의 특성에 따라 파라미터의 개수 및 유형이 달라지긴 해도, 대화창의 구조는 모두 비슷합니다. 대화창에 있는 파라미터는 다음 유형들 가운데 하나일 것입니다:

- QGIS 에서 사용할 수 있는 (현재 열려 있는) 모든 벡터 레이어들의 목록에서 선택할 수 있는 **벡터 레이어**. 아직 불러오지 않은 레이어도 사용할 수 있습니다: 위젯의 오른쪽에 있는 …버튼을 누른 다음, 다음 가운데 하나를 선택하십시오:
  - *Select file*…: OS 의 파일 탐색기를 이용해서 디스크 상에 있는 파일을 선택합니다.
  - *Browse for layer*…: 데이터베이스 소스 (PostgreSQL, SQL Server, Oracle, …), 웹 서비스 (WFS, AFS, … ), 또는 디스크 상의 파일로부터 레이어를 직접 가져올 수 있게 해주는탐색기 패널 을 엽니다.




그림 23.9: 벡터 입력 위젯

> **팁**
>
> **Drag-and drop as convenient alternative for layer selection**
>
> Besides selecting from the drop-down menu or browsing sources, you can also drag and drop a layer from the *Layers* or *Browser* panel, or from the Operating System file explorer directly into the combobox.

> **참고**
>
> 레이어 위젯은 기본적으로 레이어 이름 옆에 레이어 좌표계를 표시합니다. 이런 추가 정보를 표시하고 싶지 않다면 공간 처리 설정 대화창에서 *General* → *Show layer CRS definition in selection boxes* 옵션의 체크박스를 해제해서 이 기능을 비활성화시킬 수 있습니다.

벡터 입력 위젯에는 다음과 같은 기능들도 있습니다:

- 반복 버튼: 이 버튼을 켜면, 알고리즘이 전체 레이어에 대해 한번만 실행되는 대신 레이어의 피처 하나마다 반복해서 실행되어 알고리즘이 실행되는 횟수만큼 산출물을 생성할 것입니다. 즉 레이어에 있는 모든 피처를 개별적으로 처리해야 하는 경우 그 과정을 자동화할 수 있습니다. 알고리즘이 반복해서 실행될 수 있는 입력 벡터를 여러 개 담고 있다면, 알고리즘 내에 선언된 파라미터들의 순서에서 가장 먼저 켜진 파라미터에 대해서만 반복될 것입니다.
- Advanced options 버튼은 해당 특정 파라미터에 사용할 설정들을 조정합니다. 이 설정들의 대상은:
  - *Invalid feature filtering*: 기본 메소드 를 이용해서 무시해야 할 무결하지 않은 도형을 가진 피처를 처리할 수 있게 합니다.
  - *Limit features processed*: 소스로부터 얼마 만큼의 개수의 피처들을 처리해야 할지를 결정하는 제한값을 지정할 수 있는 옵션입니다.
  - *Feature filter*: 도구 실행 시 레이어의 하위집합을 동적으로 생성하는 표현식을 입력할 수 있습니다. 이 표현식 필터는 레이어 필터를 따로 설정하기 위해 또는 레이어 하위집합을 생성하기 위해 별도의 단계를 거쳐야 하는 일을 피할 수 있게 해줍니다.
- *Selected features only*: 알고리즘을 벡터 레이어에서 선택된 피처들에 대해서만 실행되도록 제한할 수도 있습니다.

- QGIS 에서 사용할 수 있는 모든 테이블들의 목록에서 선택할 수 있는 **테이블**. 비공간 테이블은 QGIS 에 마치 벡터 레이어처럼 불려오며동일 위젯 을 사용합니다.

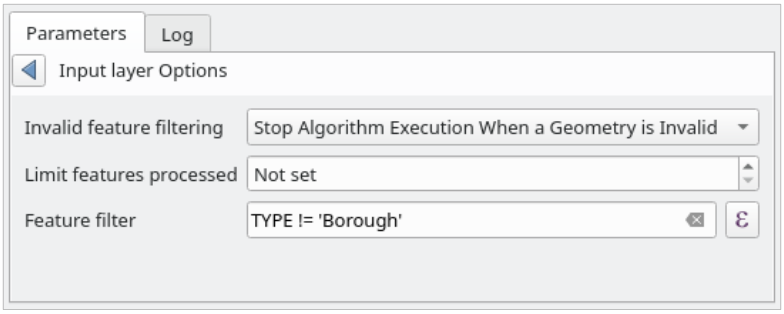



그림 23.10: 벡터 입력 위젯의 고급 옵션들

- QGIS 에서 사용할 수 있는 모든 래스터 레이어들의 목록에서 선택할 수 있는 **래스터 레이어**. 선택기 오른쪽에 …버튼이 있는데, 이 버튼을 통해 현재 QGIS 에 불려오지 않은 레이어를 나타내는 파일명을 선택할 수 있습니다.

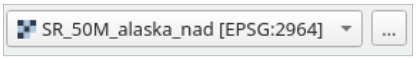



그림 23.11: 래스터 입력 위젯

- **옵션**: 사용 가능한 옵션들의 목록 가운데 선택할 수 있습니다.
- **숫자 값**: 스핀박스에서 선택할 수 있습니다. 어떤 맥락에서는 (파라미터가 피처 수준에서 적용되고 레이어 수준에서는 적용되지 않는 경우) 스핀박스 옆에 있는 Data-defined override 버튼을 누르면 수학 표현식을 입력해서 파라미터를 위한 여러 값을 생성할 수 있는표현식 작성기 가 열립니다. QGIS 가 불러온 데이터에 관련된 몇몇 유용한 변수들을 표현식에 추가할 수 있습니다. 즉 어떤 레이어의 셀 크기 또는 다른 레이어의 최북단 좌표 같은, 이런 변수들에서 파생된 어떤 값이라도 선택할 수 있다는 뜻입니다.

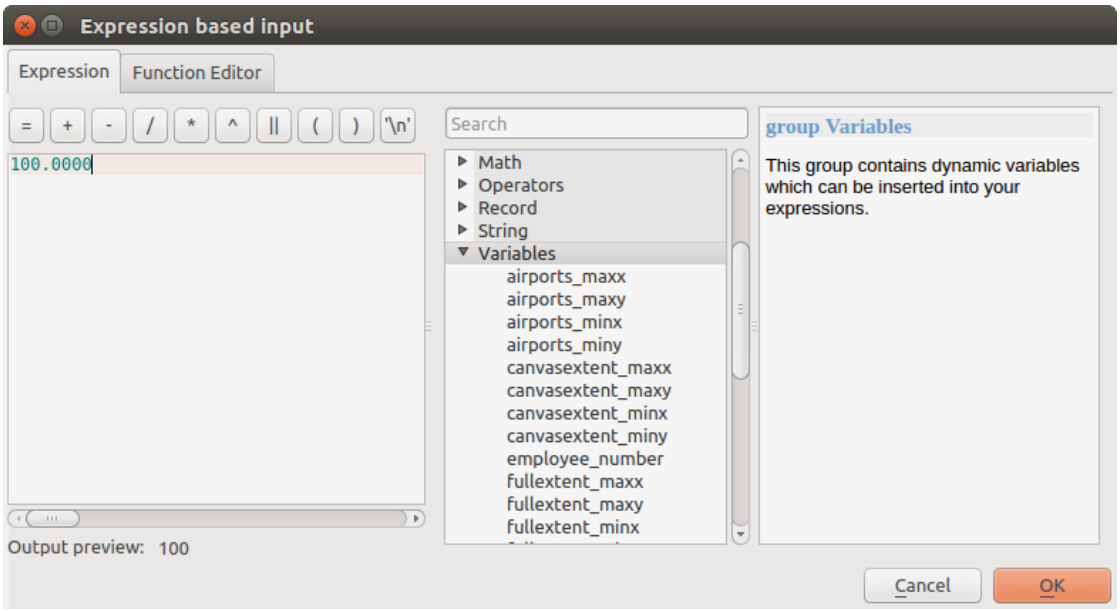



그림 23.12: 표현식 기반 입력

- **범위 (range)**: 텍스트란 2 개에서 최소 및 최대 값을 설정할 수 있습니다.
- **텍스트 문자열**: 텍스트란에 문자열을 입력할 수 있습니다.

- **필드**: 벡터 레이어의 속성 테이블에서 또는 다른 파라미터를 통해 선택한 단일 테이블에서 필드를 선택할 수 있습니다.
- **좌표계**: 드롭다운 목록에서 최근 사용한 좌표계 가운데 선택하거나 또는 목록 오른쪽에 있는 탐색 버튼을 누르면 열리는좌표계 선택기 에서 선택할 수 있습니다.

> **팁**
>
> **Drag-and-drop as convenient shortcut for CRS selection**
>
> You can also drag a layer from the Layers panel and drop it onto the CRS widget. The widget will automatically set its CRS to match the dropped layer's.

- **범위 (extent)**: `xmin, xmax, ymin, ymax` 서식으로 된 모서리 좌표를 통해 직사각형을 정의하는 텍스트란입니다. 맵 캔버스 범위를 사용하려면 Set to current map canvas extent 버튼을 누르십시오. 값 선택기 오른쪽에 있는 탐색 버튼을 클릭하면, 다음 세 가지 옵션을 제공하는 팝업 메뉴가 나타납니다:
  - *Calculate from layer*: 불러온 레이어들 가운데 선택한 레이어의 경계 상자 좌표로 텍스트란을 채웁니다.
  - *Calculate from layout map*: 현재 프로젝트에 있는 조판에서 선택한 맵 항목의 좌표로 텍스트란을 채웁니다.
  - *Calculate from bookmark*: 저장한 북마크의 좌표로 텍스트란을 채웁니다.
  - *Use current map canvas extent*: 현재 맵 캔버스 범위를 사용합니다.
  - *Draw on canvas*: 파라미터 대화창이 사라져서 캔버스 상에서 직사각형을 클릭 & 드래그할 수 있게 됩니다. 직사각형 영역을 정의하고 나면, 대화창이 범위 텍스트란에 값을 담은 채 다시 나타납니다.

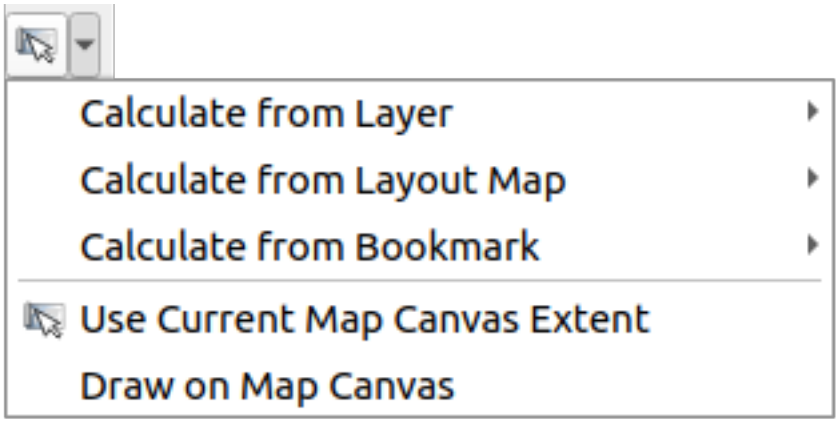



그림 23.13: 범위 선택기

- **항목 목록**: (래스터 레이어든 벡터 레이어든 또는 테이블이든 필드든) 해당 유형의 목록 가운데 선택할 수 있습니다. 옵션 왼쪽에 있는 탐색 버튼을 클릭하면 다음 그림과 같은 대화창이 열립니다. 다중 선택도 가능하며, 대화창이 닫혔을 때 선택한 항목의 개수가 파라미터 텍스트란 위젯에 표시됩니다.
- **간소 테이블**: 사용자가 테이블을 편집할 수 있습니다. 다른 목적도 있지만, 주로 색인 (lookup) 테이블 또는 합성곱 커널 (convolution kernel) 같은 파라미터를 정의하는 데 사용됩니다.

  테이블을 살펴보고 우측에 있는 버튼을 클릭해서 그 값을 편집하십시오.

  알고리즘에 따라, 대화창 우측에 있는 버튼들을 사용해서 여러 행들을 수정할 수도 있고 못 할 수도 있습니다.

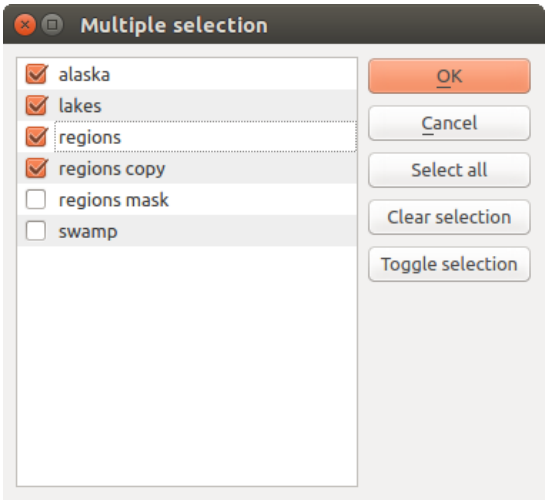



그림 23.14: 다중 선택

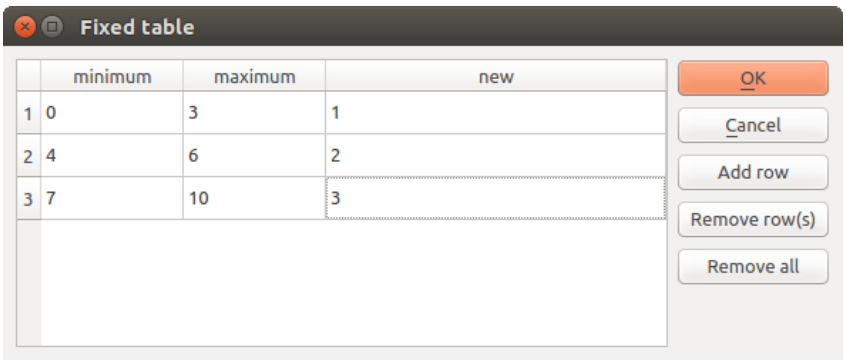



그림 23.15: 고정 테이블

> **참고**
>
> 일부 알고리즘은 실행하는 데 많은 파라미터를 필요로 합니다. 예를 들어래스터 계산기 에서는 사용자가 직접 셀 크기, 범위 및 좌표계를 지정해야 합니다. 알고리즘이 `Reference layers` 파라미터를 가지고 있을 경우 모든 파라미터를 직접 지정하는 일을 피할 수 있습니다. 이 파라미터를 사용하면 참조 레이어와 사용할 모든 레이어 속성 (셀 크기, 범위, 좌표계) 을 선택할 수 있습니다.

### 실행 로그 기록하기

Along with the *Parameters* tab, there is another tab named *Log* (see 그림 23.16 below). Information provided by the algorithm during its execution is written in this tab, allowing you to track the execution as well as being aware and having more details about the algorithm as it runs. You can directly click on the names of output files, folders, or HTML files listed in the log. Doing so will open the folder containing the generated file and automatically select it. Information on algorithm execution is also output in the *View → Panels → Log Messages Panel*.

하지만 모든 알고리즘이 *Log* 탭에 정보를 작성하지는 않습니다. 사실 최종 파일 이외에는 어떤 정보도 출력하지 않은 채 조용히 실행되는 알고리즘이 많습니다. 이 경우 *Log Messages Panel* 을 확인하십시오.

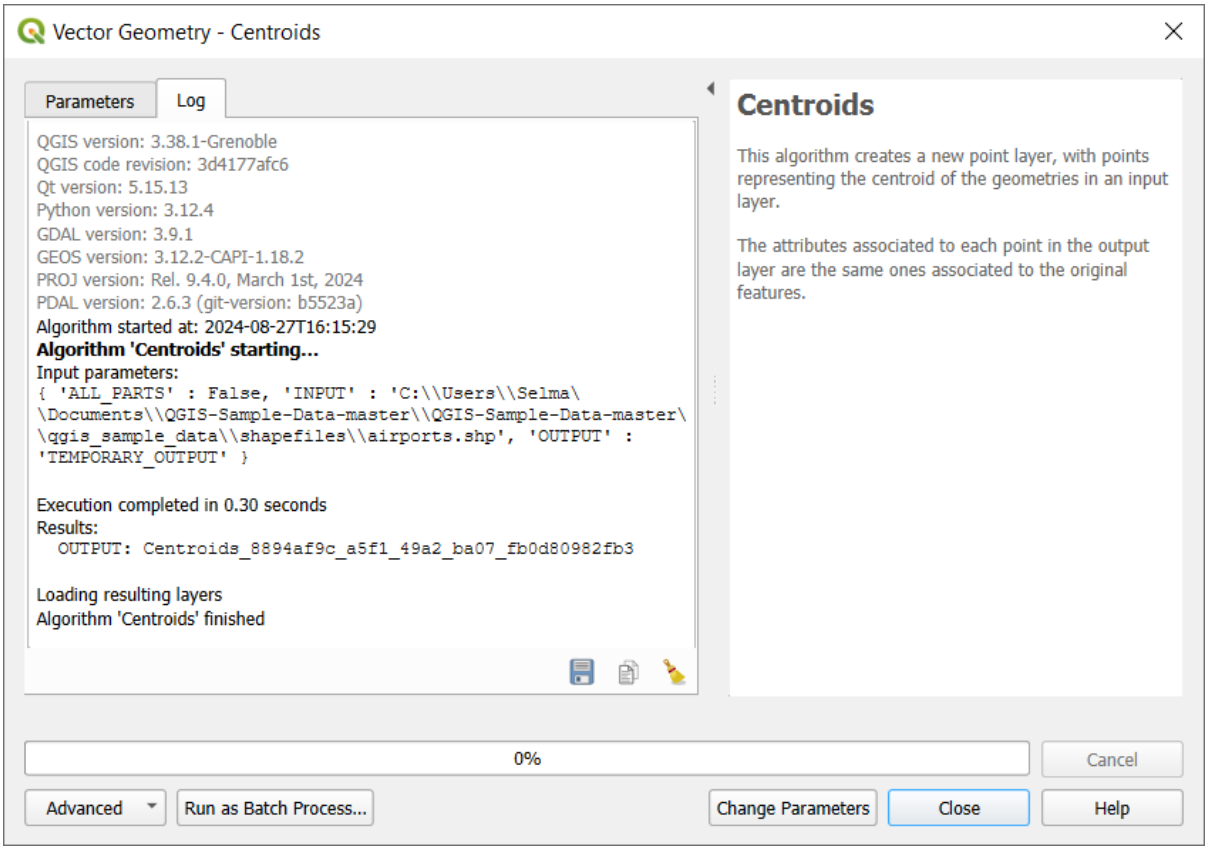



그림 23.16: 알고리즘 대화창 — 로그 탭

*Log* 탭 하단에 *Save Log to File*, *Copy Log to Clipboard* 및 *Clear Log* 버튼이 있습니다. 공간 처리 옵션 대화창의 *General* 그룹에서 *Keep dialog open after running algorithm* 옵션을 체크한 경우 이 도구들이 특히 유용합니다.

### 기타 도구

알고리즘 대화창 우측에 해당 알고리즘에 관한 간단한 설명을 표시해서 사용자가 알고리즘의 목적 및 기본 개념을 이해할 수 있도록 해주고 있습니다. 이런 설명이 존재하지 않는 경우, 설명 패널은 나타나지 않을 것입니다.

알고리즘이 사용하는 모든 파라미터에 대한 설명이나 예시를 포함할 수도 있는 더 자세한 도움말 파일이 필요한 경우, 대화창 하단에 있는 *Help* 버튼을 클릭하면공간 처리 알고리즘 문서 또는 (몇몇 제 3 자 제공자의 경우) 제공자 문서를 볼 수 있습니다.

*Advanced* [?] 메뉴는 알고리즘을 실행하지 않고서도 대화창에서 정의한 환경 설정을 재사용할 수 있는 기능을 제공합니다:

- *Algorithm Settings*···: 현재 알고리즘 실행에 대한 공간 처리 설정을 무시할 수 있게 해줍니다. 더 자세한 내용은알고리즘 설정 무시하기 를 참조하세요.
- *Copy as Python Command*: 대화창에서 정의한 파라미터들을 사용해서 도구를 실행할 수 있는 동등한*PyQGIS* 명령어 로 쉽게 복사할 수 있습니다.
- *Copy as qgis_process Command*: 거리 단위, 면적 단위, 타원체, 그리고 특정 레이어를 가진 GeoPackage 산출물처럼 까다로운 파라미터 값 같은 환경 설정을 포함하는*qgis_process* 명령어 를 쉽게 생성할 수 있습니다.
- *Copy as JSON*: 명령어의 모든 설정을 qgis_process 가 사용할 수 있도록 `JSON` 서식으로 복사합니다. (TIN 보간 파라미터 같은) 복잡 파라미터의 경우에도 명령어의 예상 서식을 볼 수 있는 편리한 방법입니다. `JSON` 서식으로 복사된 파라미터들을 쉽게 저장할 수 있고, 나중에 이 값들을 붙여넣기해서 복원할 수 있습니다.
- *Paste Settings*: 설정 파라미터들을 `JSON` 서식으로 붙여넣습니다.

The *Run as Batch Process*··· button triggers the *batch processing mode* allowing to configure and run multiple instances of the algorithm with a variety of parameters. A *Run as Single Process*··· button helps you switch back from the batch mode.

알고리즘 실행이 (성공적으로든 아니든) 완료되면, *Log* 탭이 활성화되어 있는 한 새로운 *Change Parameters* 버튼이 나타납니다.

### 알고리즘 설정 무시하기

알고리즘 대화창 하단에 있는 *Advanced* 드롭다운 메뉴에서 선택할 수 있는 *Algorithm Settings*···는 사용자가 해당 알고리즘의 실행에 대해서만 적용되는 일반적인 공간 처리 설정을 제어할 수 있는 패널을 엽니다. 이 패널의 목적은 사용자가 알고리즘의 일반적인 기본값을 변경해야 하는 일 없이전체 수준 공간 처리 설정 을 애드혹 (ad-hoc) 기반으로 무시할 수 있게 해주는 것입니다.

다음과 같은 설정을 무시할 수 있습니다:

- *Invalid feature filtering*: 각 파라미터 별로 설정을 무시하는 기존 방법과는 달리, 여기에서 처리 방법을 설정하면 알고리즘의 **모든 입력물** 에 적용될 것입니다.
- *Calculation settings*: *Distance units* 및 *Area units* 같은 거리/면적 측정에 사용할 단위를 무시할 수 있습니다.
- *Environment settings*: *Temporary folder* 및 *Number of threads to use* 같은 환경을 무시할 수 있습니다.

#### 투영체에 대한 소고 (□□)

공간 처리 알고리즘의 실행은 언제나 입력 레이어의 좌표계에서 수행됩니다. QGIS 의 실시간 (on-the-fly) 재투영 기능 때문에, 두 레이어가 중첩해서 일치하는 것처럼 보일 수도 있지만 공통 좌표계로 재투영하지 않고 원본 좌표를 사용하는 경우 일치하지 않을 수도 있다는 사실을 기억하십시오. *QGIS 내장 알고리즘* 에 하나 이상의 레이어를 입력할 때마다, 모든 입력 레이어를 — 벡터이든 래스터이든 간에 — 첫 번째 입력 레이어의 좌표계와 일치하도록 재투영할 것입니다.

하지만 공간 처리 프레임워크를 통해 실행되는 외부 응용 프로그램 알고리즘 대부분은 모든 입력 레이어가 이미 동일한 좌표계를 사용하고 분석 준비가 끝났다고 가정하기 때문에, 재투영 작업을 하지 않습니다.

기본적으로 알고리즘 대화창의 파라미터 탭은 각 레이어의 이름과 함께 좌표계 설명도 표시하기 때문에 동일한 좌표계를 공유하는 레이어들을 쉽게 입력 레이어로 선택할 수 있습니다. 이 추가 정보를 보기 싫다면, 공간 처리 설정 대화창에서 *General → Show layer CRS definition in selection boxes* 체크박스를 체크 해제해서 이 기능을 비활성화시킬 수 있습니다.

서로 일치하지 않는 좌표계가 적용된 2 개 이상의 레이어를 입력해서 알고리즘을 실행하려는 경우, *Warn before executing if layer CRS's do not match* 옵션을 활성화한 상태라면 경고 메시지가 나타날 것입니다.

그래도 알고리즘을 실행시킬 수는 있지만, 대부분의 경우 입력 레이어들이 서로 중첩하지 않기 때문에 텅 빈 레이어 같은 잘못된 산출물을 생성할 것이라는 점을 유념하십시오.

> **팁**
>
> **공간 처리 알고리즘을 사용해서 중간 단계 재투영하기**
>
> 좌표계가 일치하지 않아 여러 입력 레이어에 대해 알고리즘을 성공적으로 실행할 수 없는 경우, 레이어 재투영하기 같은 QGIS 내장 알고리즘을 사용해서 레이어들을 동일한 좌표계로 재투영한 다음 그 산출물을 사용해서 알고리즘을 실행하십시오.

### 23.3.2 알고리즘이 생성한 데이터 객체

Data objects generated by an algorithm can be of different types. They can be a layer (vector with or without geometry, raster, mesh, point cloud, ⋯), stored as a plain file on disk or in a database, text-based files such as `qml` style file or graphics `.html` file, a folder, ⋯

The parameters dialog will contain a widget corresponding to the needed output, where you can type the output channel to use for saving it. An output channel contains the information needed to save the resulting object somewhere. In the most usual case, you will save it to a file, but in the case of vector layers, and when they are generated by native algorithms (algorithms not using external applications) you can also save to a PostgreSQL, GeoPackage or SpatiaLite database, or a memory layer.

The **output** parameter for setting the result of the algorithm execution provides a text box for filling the path to the output result, and a drop-down ⋯ button for additional options (the availability depends on the algorithm you run):

- *Skip Output*: if you are not interested in a given output of the algorithm
- *Create Temporary Layer* (`TEMPORARY_OUTPUT`): the output is stored in a vector *temporary scratch layer*.
- *Save to File*⋯: you will be prompted with a save file dialog, where you can select the desired file path. Supported file extensions are shown in the file format selector of the dialog, depending on the kind of output and the algorithm.

  파일 확장자가 산출물의 포맷을 정의합니다. 지원 포맷은 알고리즘 자체가 어떤 포맷을 지원하느냐에 따라 달라집니다. 어떤 포맷을 선택하려면, 해당하는 파일 확장자를 선택하기만 하면 됩니다. (또는 파일 경로를 직접 입력하는 경우, 파일명에 추가하면 됩니다.) 사용자가 입력한 파일 경로의 확장자가 지원 포맷 가운데 어느 것과도 일치하지 않을 경우, 파일 경로에 기본 확장자가 붙어서 레이어 또는 테이블을 해당 확장자의 포맷으로 저장할 것입니다. 테이블의 경우 `.dbf`, 래스터 레이어의 경우 `.tif`, 벡터 레이어의 경우 `.gpkg`

가 기본 확장자입니다. 알고리즘 설정 대화창에서 QGIS 가 지원하는 다른 모든 포맷 가운데 하나를 선택해서 기본 확장자를 변경할 수 있습니다.

산출 데이터 객체를 위한 기본 폴더를 설정할 수 있습니다. 공간 처리 설정 대화창에서 (*Settings* → *Options* → *Processing*) *General* 그룹을 살펴보면 *Output folder* 라는 파라미터가 있을 겁니다. 알고리즘 실행시 사용자가 파일 경로 없이 파일명만 (예: `myfile.shp`) 입력하는 경우 이 산출물 폴더를 기본 경로로 사용합니다.

- *Save to a Temporary File*: if you do not enter any filename in the output text box (or select the corresponding option in the context menu), the result will be saved as a temporary file in the corresponding default file format, and it will be deleted once you exit QGIS (take care with that, in case you save your project and it contains temporary layers). The file is saved in the default `TEMP` folder on the machine, if *not overridden*.
- *Save to Geopackage…*: the output layer will be saved as a named table of a new or existing GeoPackage database
- *Save to Database Table…*: the output layer will be saved as a new table in a database such as GeoPackage, SpatiaLite, PostgreSQL or MS SQL Server. A connection should already exist. You'll need to provide the name of the new table and depending on the type of target database, select a schema,
- *Append to Layer…*: adds features output by the algorithm to an existing layer. Upon selecting the target layer, a field mapping panel opens, allowing to manually set how fields from the output layer are mapped to the target layer's fields.

- *Save to Directory*: Use a new or existing directory for storing the algorithm outputs
- *Save to a Temporary Directory*: creates a directory within the set `TEMP` folder for storing the algorithm outputs. The files will be deleted once you exit QGIS.

벡터 레이어를 반복 모드로 사용하는 알고리즘 실행 시, 입력한 파일 경로를 모든 생성 파일을 저장하는 기본 경로로 사용합니다. 즉 파일 경로의 파일명을 기본 이름으로 삼아 반복 작업 차례대로 번호를 붙일 것이라는 뜻입니다. 이런 모든 생성 파일의 포맷도 파일 경로의 파일 확장자 (포맷) 를 따릅니다.

Algorithms also generate graphics and text as HTML files. These results are shown at the end of the algorithm execution in the *Results Viewer* dialog. This dialog will keep the results produced by any algorithm during the current session, and can be shown at any time by selecting *Processing* → *Results Viewer* from the QGIS main menu.

For layer-based outputs, it is possible to  *Open output file after running algorithm* directly in QGIS, as a new entry in the *Layers* panel. By default, the files are opened.

## 23.4 이력 관리자

### 23.4.1 공간 처리 이력

어느 알고리즘을 실행할 때마다, 이력 관리자에 해당 공간 처리 관련 정보가 저장됩니다. 사용한 파라미터는 물론 실행 일시도 저장됩니다. 이력 관리자를 사용하면 공간 처리 프레임워크를 통해 처리했던 모든 작업을 쉽게 추적하고 제어할 수 있으며, 또 재실행할 수도 있습니다.

공간 처리 정보는 — 툴박스에서 알고리즘을 실행했다 하더라도 — 명령 줄 표현식으로 저장됩니다. 이 습성은 명령 줄 인터페이스 사용법을 배우는 사용자에게 유용합니다. 툴박스를 통해 알고리즘을 호출한 다음 이력 관리자를 확인하면 명령 줄에서 어떻게 동일 알고리즘을 호출할 수 있는지 알 수 있기 때문입니다.

화살표를 오른쪽 클릭하면 다음 작업을 할 수 있을 것입니다:

- *Copy as Python Command*: 대화창에서 동등한*PyQGIS 명령어* 를 쉽게 복사할 수 있습니다. 명령어 목록 아래 표시되는 코드도 마찬가지입니다.

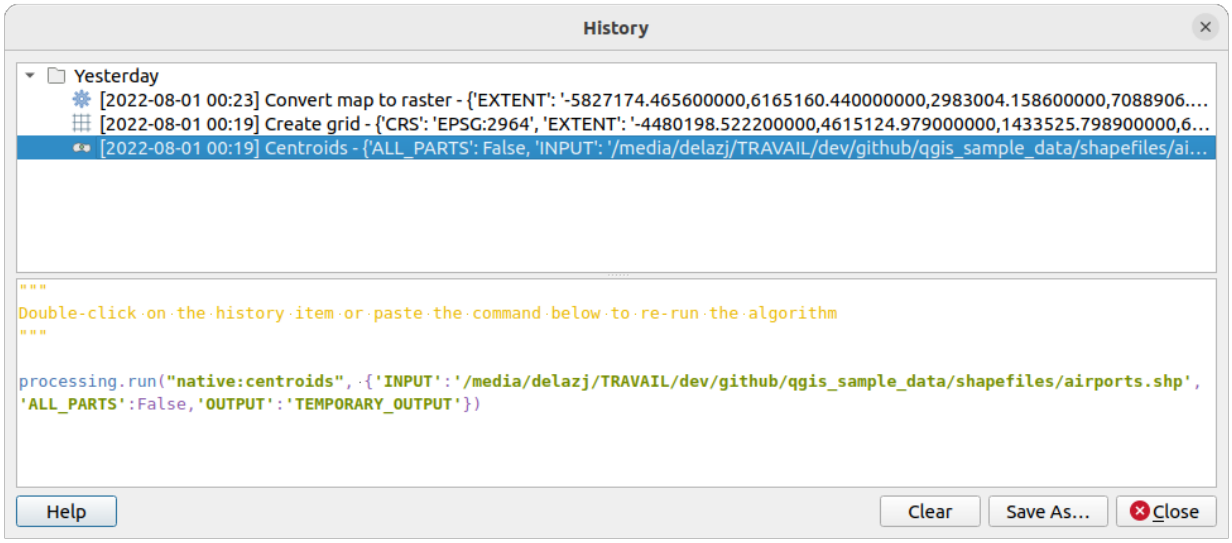



그림 23.17: 이력

- *Copy as qgis_process Command*: 거리 단위, 면적 단위, 타원체, 그리고 특정 레이어를 가진 GeoPackage 산출물처럼 까다로운 파라미터 값 같은 환경 설정을 포함하는*qgis_process* 명령어 를 쉽게 생성할 수 있습니다.
- *Copy as JSON*: 명령어의 모든 설정을 qgis_process 가 사용할 수 있도록 JSON 서식으로 복사합니다. (TIN 보간 파라미터 같은) 복잡 파라미터의 경우에도 명령어의 예상 서식을 볼 수 있는 편리한 방법입니다. JSON 서식으로 복사된 파라미터들을 쉽게 저장할 수 있고, 나중에 알고리즘 대화창에 이 값들을 붙여넣기해서 복원할 수 있습니다.
- *Create Test…*: 공간 처리 README 파일 에 있는 지침을 따라, 의심되는 알고리즘과 파라미터를 이용하는 테스트를 생성할 수 있습니다.

레지스트리 항목 탐색 이외에도, 어느 항목을 더블클릭하는 것만으로도 해당 공간 처리를 재실행할 수 있습니다. 그러면 이미 설정된 파라미터를 담은 알고리즘 대화창이 열립니다. 사용자 필요에 따라 어느 파라미터든 변경한 다음 알고리즘을 재실행할 수 있습니다.

*History* 대화창은 QGIS 공간 처리 알고리즘 및 스크립트의 하부 구조를 통합하는 데 기여할 수 있는 편리한 방법도 제공합니다.

### 23.4.2 공간 처리 로그

이력 관리자 대화창은 알고리즘 실행시 생산된 정보가 아니라 실행 호출만 담고 있습니다. 해당 정보는 QGIS 로그에 (*View → Panels → Log Messages Panel*) 작성됩니다.

제 3 자 알고리즘은 보통 명령 줄 인터페이스를 통해 실행됩니다. 명령 줄이란 콘솔을 통해 사용자와 소통하는 인터페이스입니다. 사용자가 해당 콘솔을 볼 수는 없어도, 사용자가 이런 알고리즘을 실행할 때마다 로그에 콘솔의 전체 덤프를 작성합니다. 로그가 이런 정보로 채워지는 일을 피하기 위해, 공간 처리 설정 대화창에서 각 제공자별로 이 습성을 비활성화시킬 수 있습니다.

몇몇 알고리즘은 — 지정한 입력 데이터로 결과물을 산출할 수 있다고 해도 — 데이터에 잠재적인 문제가 있다고 탐지한 경우 사용자에게 경고하기 위해 로그에 산출물 주석 또는 부가 정보를 추가할 수도 있습니다. 기대하지 않았던 산출물이 나온 경우 로그에 있는 해당 메시지를 반드시 확인하십시오.

## 23.5 모델 설계자

모델 설계자 는 사용하기 쉽고 간단한 인터페이스를 통해 복잡 모델을 생성할 수 있게 해줍니다. GIS 작업시, 대부분의 분석 작업은 각각 분리된 작업이라기 보다는 연속 작업들의 일부인 경우가 더 많습니다. 모델 설계자를 사용하면, 그런 연속 공간 처리 작업들을 단일 공간 처리 작업으로 묶을 수 있습니다. 서로 다른 입력 집합에 대해 여러 작업을 연속해서 처리하는 것보다 단일 공간 처리 작업을 실행하는 편이 더 편리하기 때문입니다. 몇 단계를 거치든, 몇 개의 서로 다른 알고리즘을 사용하든 상관없이 모델은 단일 알고리즘처럼 실행되기 때문에, 시간과 노력을 절약할 수 있습니다.

공간 처리 메뉴에서 (*Processing* ▯ *Model Designer*) 모델 설계자를 열 수 있습니다.

### 23.5.1 모델 설계자 인터페이스

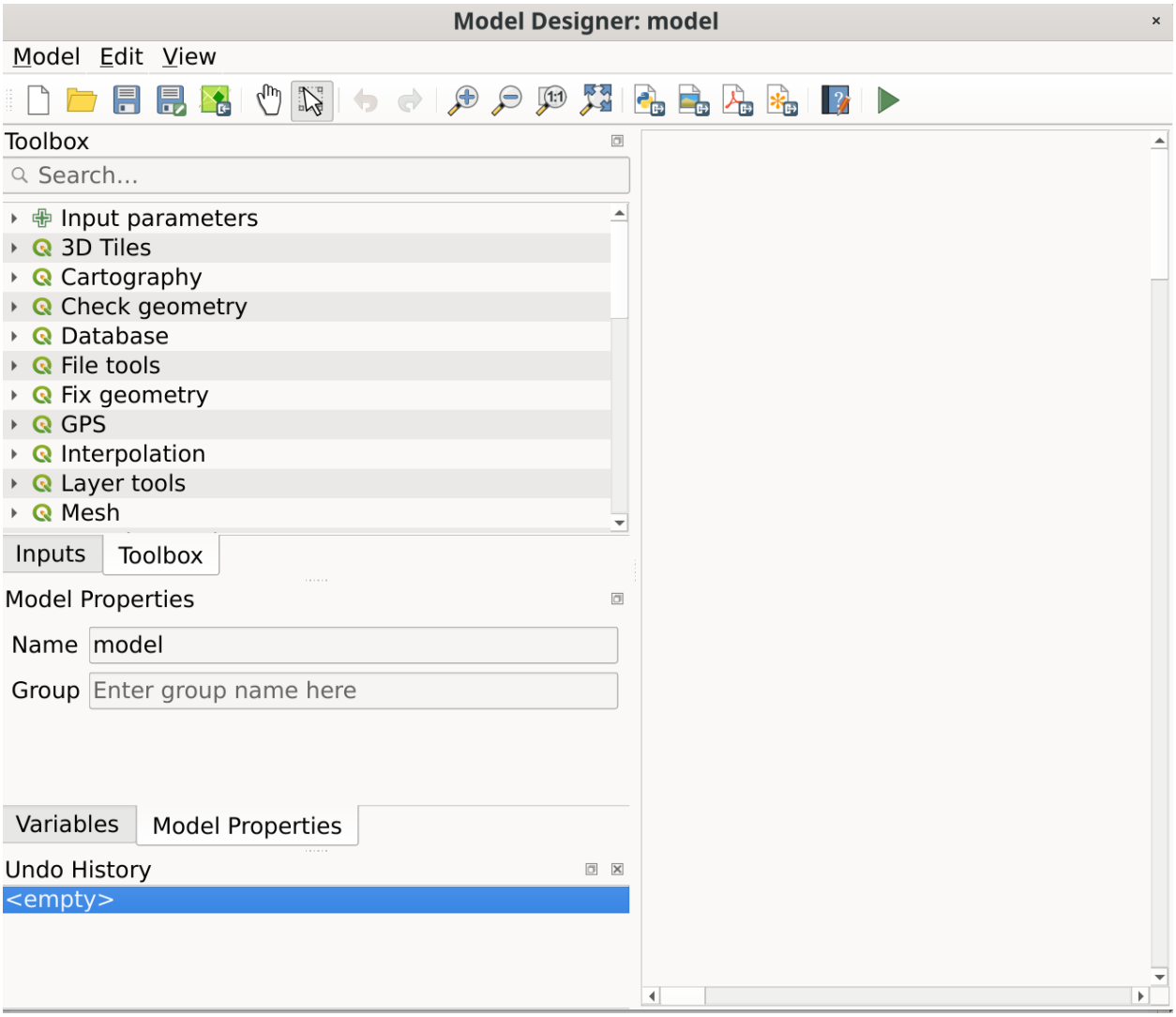



그림 23.18: 모델 설계자

모델 설계자의 주요 부분은 모델 구조 및 모델이 표현하는 워크플로 (workflow) 를 구성할 수 있는 작업 캔버스입니다.

대화창 최상단에 있는 다양한 메뉴와 *Navigation* 툴바를 통해 다양한 도구를 사용할 수 있습니다.

### 모델 메뉴

| 라벨 | 단축키 | 탐색 툴바 | 설명 |
|---|---|---|---|
| *Validate Model* | | | 모델에 쓰이는 알고리즘 및 입력물이 존재하는지 여부를 점검합니다. 모델을 배포하기 전에 사용하면 편한 도구입니다. |
| *Run Model…* | F5 | ☑ | 모델을 실행합니다. |
| *Run Selected Steps…* | Shift+F5 | | Runs only the selected steps in a model, allowing to run a subset of the model |
| *Reorder Model Inputs…* | | | 알고리즘 대화창에서 사용자에게 나타나는 입력물들의 순서를 설정합니다. |
| *Reorder Output Layers…* | | | 프로젝트에 결과물을 불러올 때 모델의 산출물이 반드시 따라야만 하는 특정한 순서를 설정합니다. |
| *Open Model…* | Ctrl+O | ☑ | 편집 또는 실행하기 위해 `.model3` 파일을 엽니다. |
| *Save Model* | Ctrl+S | ☑ | 디스크에 모델을 `.model3` 파일로 저장합니다. |
| *Save Model as…* | Ctrl+Shi | ☑ | 디스크에 모델을 새로운 `.model3` 파일로 저장합니다. |
| *Save Model in project* | | ☑ | 프로젝트 파일 공유 시 사용할 수 있도록 프로젝트 파일에 모델 파일을 내장시킵니다. |
| *Edit Model Help…* | | ☑ | 모델, 알고리즘, 파라미터 및 산출물은 물론 작성자와 버전 관리 내용을 문서화할 수 있는 인터페이스입니다. |
| *Export* | | | |
| *Export as Image…* | | ☑ | 모델의 그래픽 디자인을 (삽화 목적) 이미지 파일 포맷으로 저장합니다. |
| *Export as PDF…* | | | 모델의 그래픽 디자인을 (삽화 목적) `PDF` 파일 포맷으로 저장합니다. |
| *Export as SVG…* | | | 모델의 그래픽 디자인을 (삽화 목적) `SVG` 파일 포맷으로 저장합니다. |
| *Export as Script Algorithm…* | | ☑ | 모델의 지침을 포함하는 파이썬 스크립트 파일을 생성합니다. |

### 편집 메뉴

| 라벨 | 단축키 | 탐색 툴바 | 설명 |
|---|---|---|---|
| *Select All* | Ctrl+A | | 설계자에 있는 모델 구성 요소를 모두 선택합니다. |
| *Snap selected components to Grid* | | | 그리드에 요소들을 스냅해서 정렬시킵니다. |
| *Redo* | Ctrl+Y | ☑ | 마지막으로 취소시킨 액션을 다시 실행합니다. *Undo/Redo* 패널도 참조하세요. |
| *Undo* | Ctrl+Z | ☑ | 이전 변경 사항을 취소합니다. *Undo/Redo* 패널도 참조하세요. |
| *Cut* | Ctrl+X | | 모델에서 구성 요소 선택 집합을 잘라냅니다. |
| *Copy* | Ctrl+C | | 모델에서 구성 요소 선택 집합을 복사합니다. |
| *Paste* | Ctrl+V | | 또다른 또는 동일한 모델에 모델에서 잘라냈거나 복사한 구성 요소 선택 집합을 붙여넣습니다. 선택한 구성 요소는 원래 속성 및 주석을 유지합니다. |
| *Delete selected components* | Del | | 모델에서 구성 요소를 제거합니다. |
| *Add Group Box* | | | 관련된 구성 요소들을 시각적으로 묶기 위해 구성 요소들의 배경에 상자를 추가합니다. 큰 모델에서 워크플로를 명료하게 유지하는 데 특히 유용합니다. |

### 뷰 메뉴

| 라벨 | 단축키 | 탐색 툴바 | 설명 |
|---|---|---|---|
| *Zoom To* | | | 선택한 그룹 상자 영역으로 확대/축소합니다. |
| *Zoom In* | Ctrl++ | ☑ | |
| *Zoom Out* | Ctrl+- | ☑ | |
| *Zoom to 100%* | Ctrl+1 | ☑ | |
| *Zoom Full* | Ctrl+0 | ☑ | 설계자의 현재 캔버스에 있는 모든 구성 요소를 표시합니다. |
| *Show Comments* | | | 모델 설계자에 있는 모든 알고리즘 또는 입력물에 관련된 주석들을 표시합니다. |
| *Enable Snapping* | | | |
| *Toggle Panel Visibility* | Ctrl+Tab | | 설계자에패널들 을 켜고 끕니다. |

#### 패널

대화창의 왼쪽 부분은 모델에 새 요소를 추가하기 위해 사용할 수 있는 패널 5 개로 이루어져 있습니다:

1. *Inputs*: 사용자 모델을 형성할 수 있는 모든입력 파라미터 를 지정합니다.
2. *Toolbox*: here you can find available *Processing algorithms*. It also includes input parameters. Use *Search…* box to find algorithms or inputs.
3. *Model Properties*: 모델 이름 (필수) 및공간처리 툴박스 에서 모델 이름이 속하게 될 그룹을 지정합니다.
4. *Variables*: 모델은 유일하면서도 자체에서만 사용할 수 있는 전용변수 를 담고 있을 수 있습니다. 모델 안에서 쓰이는 어떤 표현식이든 이 변수들에 접근할 수 있습니다. 이런 변수는 모델 안에서 알고리즘을 제어하는 데 유용하며, 단일 변수를 변경해서 모델의 여러 측면을 제어할 수도 있습니다. *Variables* 패널에서 변수를 살펴보고 수정할 수 있습니다.
5. *Undo History*: 이 패널에는 모델 설계자에서 발생하는 모든 일들이 등록될 것입니다. 즉 사용자가 실수한 내용을 쉽게 취소할 수 있습니다.

#### 사용 가능한 알고리즘에 관해

Some algorithms that can be executed from the *Processing* ▷ *Toolbox* do not appear in the list of available algorithms when you are designing a model. To be included in a model, an algorithm must have the correct semantic. If an algorithm does not have such a well-defined semantic (for instance, if the number of output layers cannot be known in advance), then it is not possible to use it within a model, and it will not appear in the list of algorithms that you can find in the modeler dialog. On the other hand some algorithms are specific to the modeler. Those algorithms are located within the group 'Modeler Tools'.

### 23.5.2 모델 생성 작업

모델 생성은 기본 2 단계로 이루어집니다:

1. 필요한 입력물을 정의하기: 입력해야 할 항목들이 *Inputs* 패널의 **Parameters** 창에 추가되어 모델 실행 시 사용자가 그 값들을 설정할 수 있습니다. 모델은 그 자체로 알고리즘으로서, 공간 처리 프레임워크에서 사용할 수 있는 모든 알고리즘과 마찬가지로 파라미터 창을 자동적으로 생성합니다.
2. 작업 흐름을 정의하기: 모델의 입력 데이터를 사용해서, 알고리즘을 추가하고 해당 알고리즘이 정의된 입력물을 또는 모델 안에 있는 다른 알고리즘이 생성하는 산출물을 어떻게 사용할지 선택함으로써 작업 흐름을 정의합니다.

#### 입력물 정의

첫 단계는 모델에 대한 입력물을 정의하는 것입니다. 모델 설계자 창의 왼쪽에 있는 *Inputs* 패널에서 입력물을 찾아볼 수 있습니다. 입력물 위에 마우스 커서를 가져가면 추가 정보를 가진 툴팁이 뜰 것입니다. 모델 설계자에서 사용할 수 있는 파라미터들의 전체 목록과 스크립트 작업을 위한 파라미터 대응 문자열에 대해 알고 싶다면공간처리 알고리즘의 입력물 및 산출물 유형 을 읽어보십시오.

이 항목들 가운데 어느 항목을 더블클릭해도 그 특성을 정의할 수 있는 대화창이 열립니다. 파라미터에 따라, 해당 대화창이 적어도 하나 이상의 요소 (모델 실행시 사용자가 보게 될 파라미터 설명) 를 담고 있을 수도 있고 또는 하나 이상의 옵션을 담고 있을 수도 있습니다. 예를 들어 숫자 값을 추가하는 경우, 다음 그림에서 볼 수 있는 것처럼 파라미터 설명 이외에도 기본값을 설정하고 무결한 값의 범위도 설정해야 합니다.

☑ `Mandatory` 옵션을 체크하면 사용자 입력물을 사용자 모델을 위한 필수 입력물로 정의할 수 있고, ☐ `Advanced` 체크박스를 체크하면 `Advanced` 부분에서 입력물을 설정할 수 있습니다. 이 옵션은 모델에 파라미터가 많이 있는데 그 가운데 몇 개가 평범한 파라미터가 아니지만 그래도 선택하고자 하는 경우 특히 유용합니다.

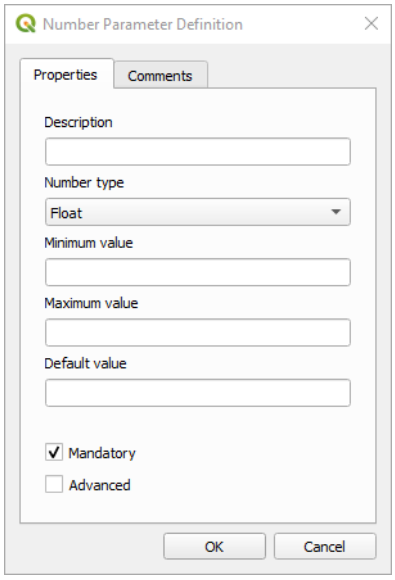



그림 23.19: 모델 파라미터 정의하기

입력물을 추가할 때마다, 모델 설계자 캔버스에 새 항목이 추가됩니다.




그림 23.20: 캔버스에 표시되는 모델 파라미터

파라미터 목록에서 입력물 유형을 모델 설계자 캔버스 상의 원하는 위치로 드래그해서 입력물을 추가할 수도 있습니다. 기존 입력물의 파라미터를 변경하고 싶다면, 그냥 입력물을 더블클릭하기만 하면 동일한 대화창이 뜰 것입니다.

또다른 모델 안에서 모델을 사용하는 경우, 필요한 입력물 및 산출물이 캔버스에 표시될 것입니다.

### 작업 흐름 정의

다음 예시에서 입력물 2 개와 알고리즘 2 개를 추가할 것입니다. 이 모델의 목적은 `Drape` 알고리즘을 사용해서 DEM 래스터 레이어에서 라인 레이어로 표고 값을 복사한 다음, `Climb Along Line` 알고리즘을 사용해서 라인 레이어의 총 오르막 (total ascent) 을 계산하는 것입니다.

*Inputs* 탭에서 라인을 `Vector Layer` 로, DEM 을 `Raster Layer` 로 두 입력물을 선택합니다. 이제 워크플로에 알고리즘을 추가할 준비가 끝났습니다.

Algorithms can be found in the *Toolbox* panel, grouped much in the same way as they are in the Processing toolbox.



그림 23.21: 모델 입력물

모델에 알고리즘을 추가하려면, 입력물을 추가할 때와 마찬가지로 알고리즘 이름을 더블 클릭하거나 모델 설계자 캔버스로 드래그 & 드롭하십시오. 입력물과 마찬가지로 알고리즘의 설명을 변경하고 코멘트를 추가할 수 있습니다. 알고리즘을 추가하면, 툴박스에서 알고리즘을 실행할 때 열리는 실행 패널에 있는 내용과 비슷한 내용을

담은 실행 대화창이 열릴 것입니다. 다음 그림은 `Drape (set Z value from raster)` 및 `Climb along line` 알고리즘 대화창입니다.

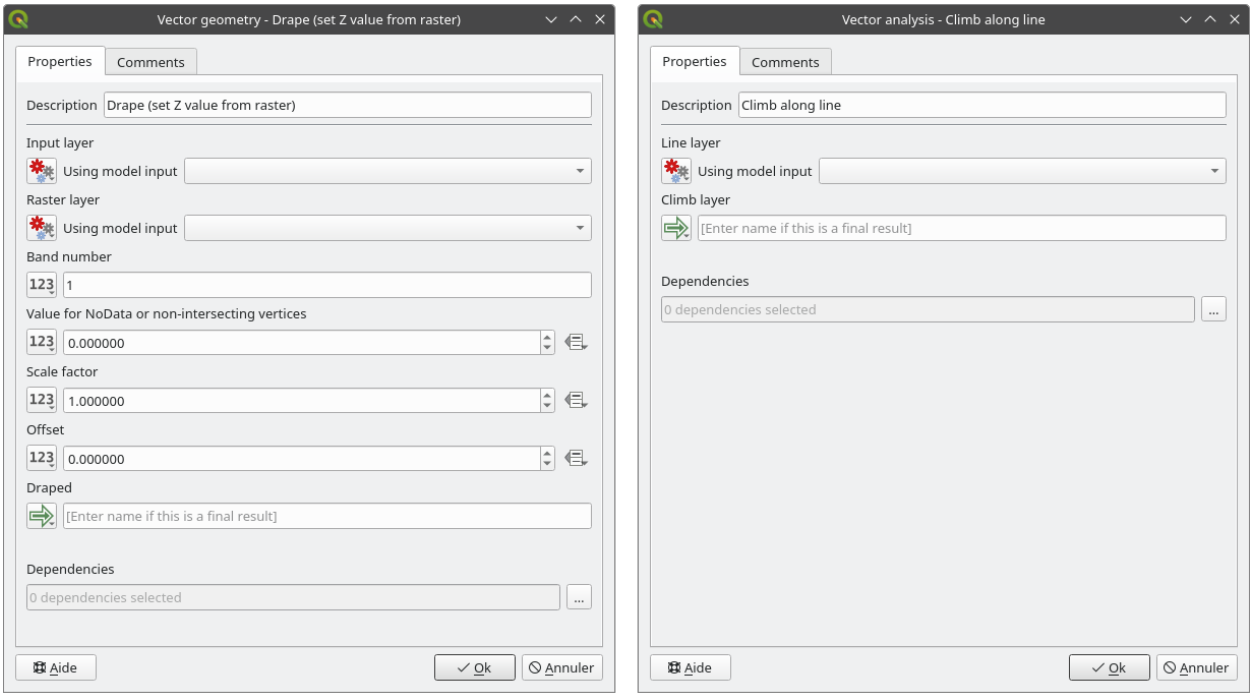



그림 23.22: 모델 알고리즘 파라미터

하지만 이 그림에서 볼 수 있듯이 몇몇 차이점이 존재합니다. 각 파라미터 옆에 워크플로 도중 해당 파라미터가 어떤 역할을 할지를 제어할 수 있는 드롭다운 메뉴가 있습니다:

- Value: 파라미터에 정적 값을 할당할 수 있습니다. 파라미터 유형에 따라 나타나는 위젯에서 숫자 (`5.0`) 또는 문자열 (`mytext`) 을 입력하거나, QGIS 프로젝트가 불러온 레이어 (들) 또는 폴더로부터 레이어 (들) 를 선택하거나, 목록에서 항목을 선택하거나 등등을 할 수 있습니다.
- Pre-calculated Value: 표현식 작성기 대화창이 열려 사용자가 파라미터를 채우기 위한 표현식을 정의할 수 있습니다. 모델 입력물은 물론 기타 레이어 통계 일부를 **변수** 로 사용할 수 있으며, 이들은 표현식 작성기의 검색 대화창 상단에 목록화되어 있습니다. 하위 (자식) 알고리즘이 실행되기 전에 이 표현식을 평가한 다음 해당 알고리즘이 실행되는 동안 사용합니다.
- Model Input: 모델에 추가된 입력물을 파라미터로 사용할 수 있습니다. 한번 클릭하기만 하면, 이 옵션이 해당 파라미터에 적합한 모든 입력물의 목록을 표시할 것입니다.
- Algorithm Output: 또다른 알고리즘의 산출물을 현재 알고리즘의 입력물로 사용할 수 있습니다. 모델 입력물과 마찬가지로, 이 옵션은 해당 파라미터에 적합한 모든 입력물의 목록을 표시할 것입니다.
- **산출 파라미터** 또한 드롭다운 메뉴에 앞의 옵션들을 가지고 있습니다:
  - 예를 들어 항상 하위 알고리즘의 산출물을 사전 정의된 GeoPackage 또는 PostgreSQL 레이어로 저장하는 등, 하위 알고리즘에 정적 산출물을 추가할 수 있습니다.
  - 예를 들어 오늘 날짜를 바탕으로 파일명을 자동 생성하고 산출물을 해당 파일로 저장하는 등, 하위 알고리즘에 표현식 기반 산출값을 사용할 수 있습니다.
  - 예를 들어 산출 파일 또는 폴더를 지정하는 파일/폴더 모델 입력물처럼 모델 입력물을 사용할 수 있습니다.

- 예를 들어 (모델 설계자 도구 의) 디렉터리 생성 알고리즘의 산출물 같은 또다른 알고리즘의 산출물을 사용할 수 있습니다.
- 추가적인 Model Output 옵션을 사용하면 모델에서 알고리즘 산출물을 사용할 수 있습니다. 알고리즘이 생성한 레이어가 또다른 알고리즘의 입력물로만 쓰이는 경우, 해당 텍스트 란을 편집하지 마십시오.

다음 그림에서 두 입력 파라미터가 `Model Input` 과 임시 산출 레이어로 정의된 것을 볼 수 있습니다:

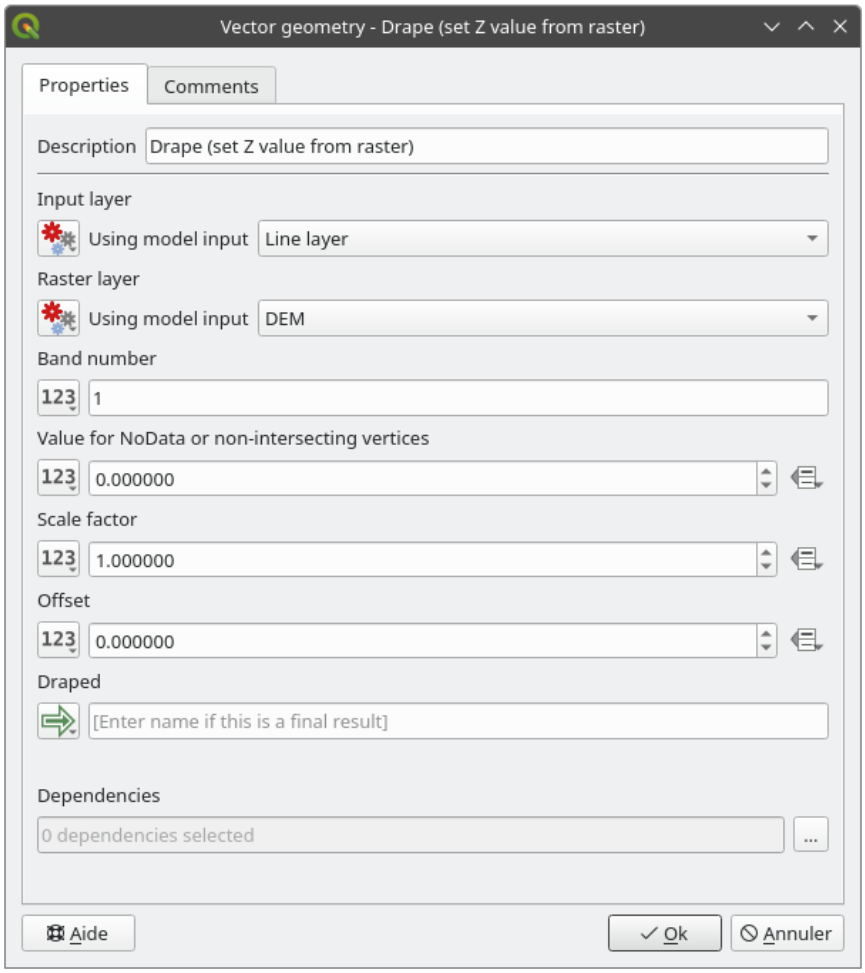



그림 23.23: 알고리즘 입력 및 출력 파라미터

또한 툴박스에서 알고리즘을 호출하는 경우 *Dependencies* 라는 추가적인 파라미터를 사용할 수 없을 것입니다. 이 파라미터는 알고리즘 하나를 현재 파라미터의 상위 (부모) 로 명확하게 정의해서 알고리즘들이 실행되는 순서를 정의할 수 있습니다. 이 파라미터를 사용하면 현재 알고리즘을 실행하기 전에 상위 알고리즘이 실행되도록 강제할 것입니다.

When you use the output of a previous algorithm as the input of your algorithm, that implicitly sets the previous algorithm as parent of the current one (and places the corresponding arrow in the modeler canvas). However, in some cases an algorithm might depend on another one even if it does not use any output object from it (for instance, an algorithm that executes a SQL sentence on a PostgreSQL database and another one that imports a layer into that same database). In that case, just select the previous algorithm in the *Dependencies* parameter and they will be executed in the correct order.

모든 파라미터에 무결한 값을 할당하고 나면, *OK* 버튼을 누르십시오. 알고리즘이 캔버스에 추가될 것입니다. 해당 알고리즘은 캔버스에 있는 다른 모든 항목들과 — 해당 알고리즘에 입력되는 객체를 제공하는 알고리즘이든 입력물이든 — 연결돼 있을 것입니다

Elements can be dragged to a different position on the canvas using the Select/Move/Link Item tool. This is useful to make the structure of the model clearer and more intuitive. You can also resize the elements, grasping their border. This is particularly useful if the description of the input or algorithm is long. With *View → Enable snapping* option checked, items resizing or displacement can be bound to a virtual grid, for a more visually structured algorithm design.

요소들 사이의 링크는 자동으로 업데이트되며 각 알고리즘의 상단과 하단에서 + 버튼을 볼 수 있습니다. 이 버튼을 클릭하면 해당 알고리즘의 입력물 및 산출물의 목록을 표시해서 사용자가 알고리즘의 개요를 빠르게 살펴볼 수 있습니다.

You can create or remove connections between model components using an interactive drag-and-drop interface. Each parameter and algorithm item on the model canvas includes small circular sockets, output sockets on the right and input sockets on the left. To connect components, click and drag from an output socket (either from a model input or an algorithm) to an input socket on another algorithm. A visual edge follows the cursor and completes the connection when released. To remove a connection, drag the edge away from the input socket. The algorithm dialogs update automatically to reflect changes made using the drag-and-drop interface.

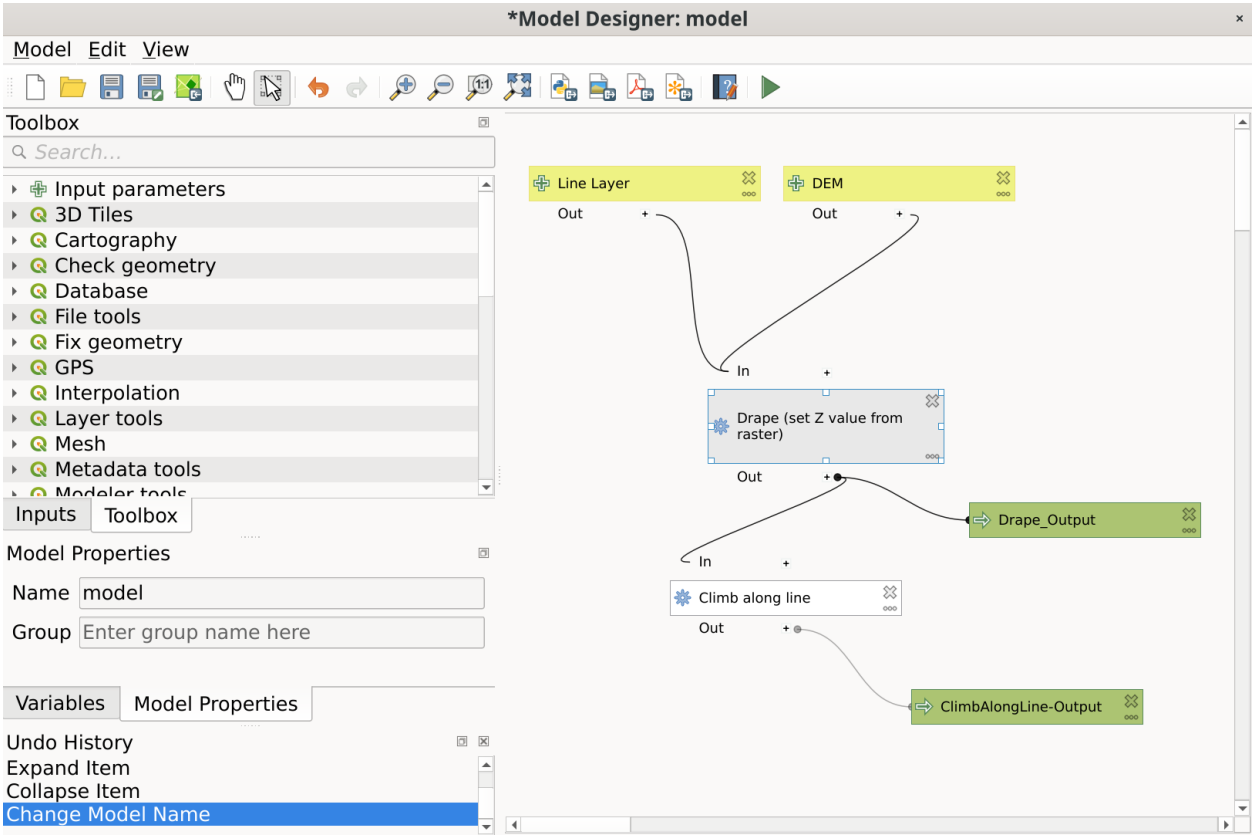



그림 23.24: 완성 모델

With the *Edit → Add Group Box* tool, you can add a draggable *box* to the canvas. This feature is very useful in big models to group related elements in the modeler canvas and to keep the workflow clean. For example we might group together all the algorithms of the example:

상자의 이름과 색상을 변경할 수 있습니다. 그룹 상자를 *View → Zoom To →* 도구와 함께 사용하는 경우, 모델의 특정 부분으로 확대/축소할 수 있기 때문에 매우 유용합니다. 물론 마우스 휠로 확대/축소할 수도 있습니다.

입력물의 순서 및 메인 모델 대화창에 목록화된 순서를 변경하고 싶을 수도 있습니다. `Input` 패널 하단에 `Re-order Model Inputs...` 버튼이 있는데, 이 버튼을 클릭하면 입력물의 순서를 변경할 수 있는 새 대화창이 뜹니다.

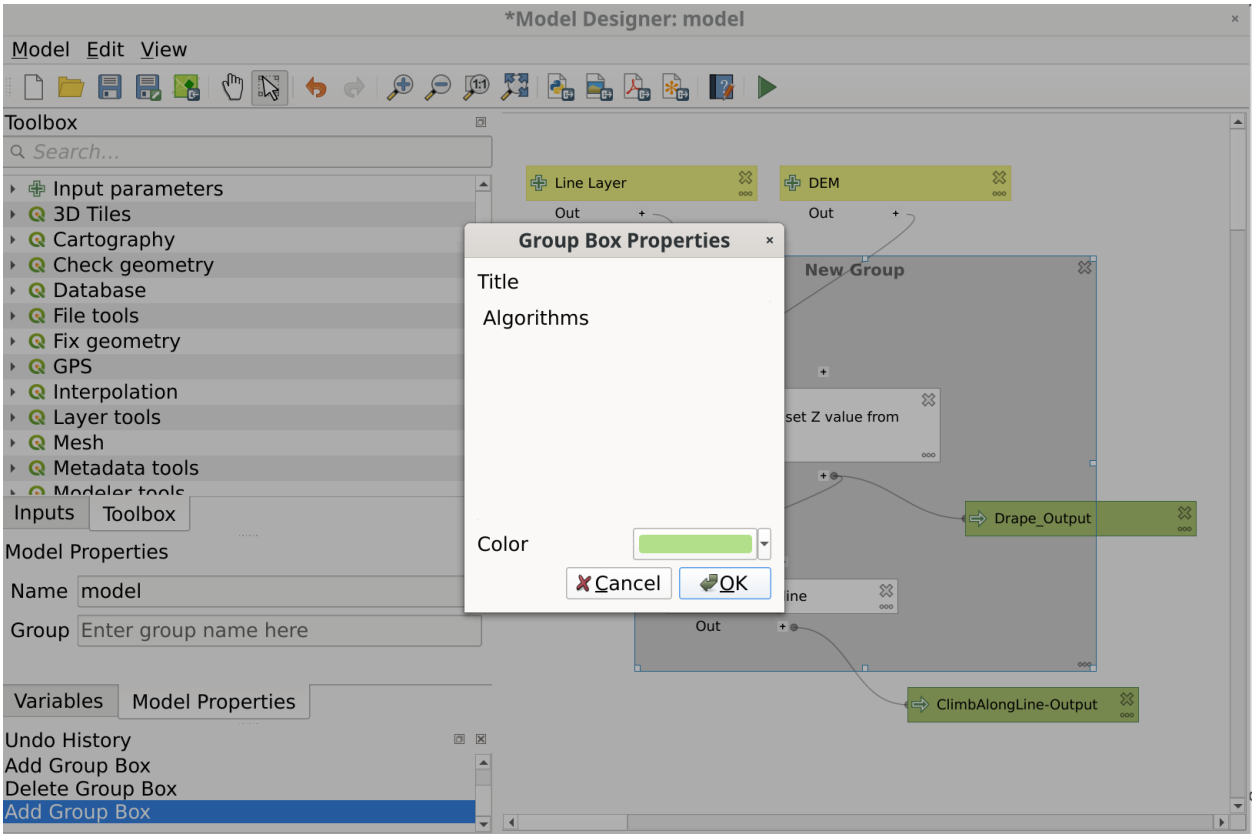



그림 23.25: 모델 그룹 상자

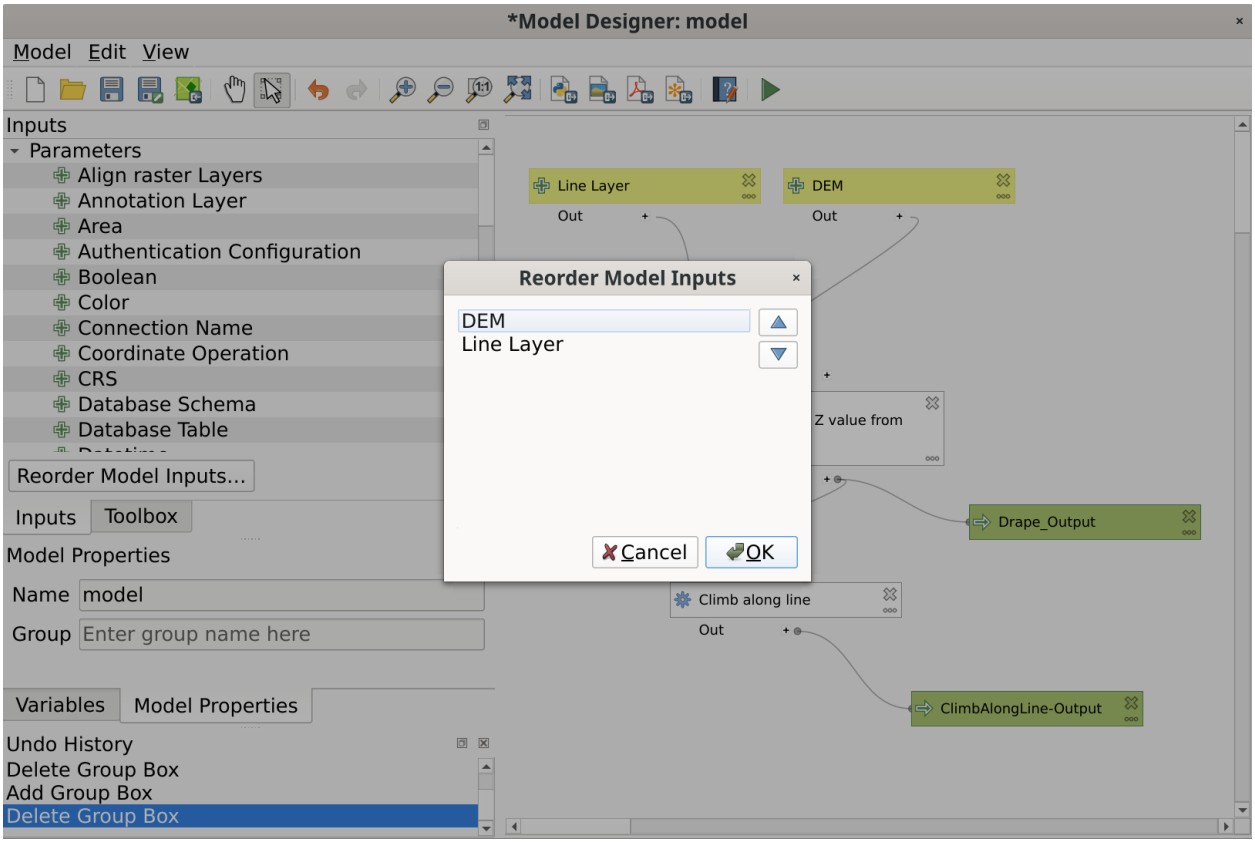



그림 23.26: 모델 입력물 재정렬

프로젝트에 결과물을 불러올 때 모델의 산출물이 반드시 따라야만 하는 특정한 순서를 설정해야 할 가능성도 존재합니다. 따라서 모델 설계자에서 모델을 실행할 때 캔버스 상에 레이어들을 논리적인 순서대로, 이를테면 벡터 레이어 산출물을 래스터 레이어 산출물 위로 또는 포인트 레이어 산출물을 폴리곤 레이어 산출물 위로 정렬할 수 있게 해주는 수단이 필요합니다. 모델 설계자에서는 레이어 트리 안에 산출물을 새 그룹 이름을 사용해서 또는 기존 그룹에 추가해서 자동으로 그룹화해주기 위한 선택적인"그룹 이름"도 설정할 수 있습니다. *Model* 메뉴에 있는 *Reorder Output Layers*…항목을 클릭하면 산출 레이어들의 순서를 변경할 수 있는 새 대화창이 열립니다:

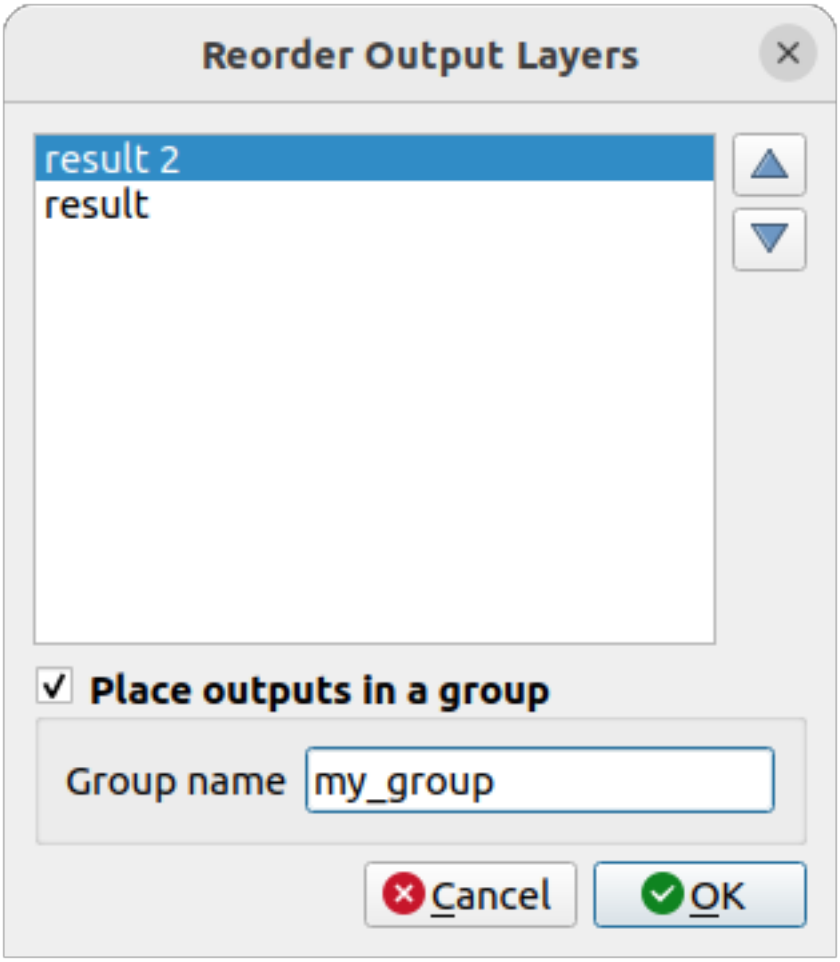



그림 23.27: 산출 레이어 재정렬

모델 설계자에 존재하는 입력물 또는 알고리즘에 주석도 추가할 수 있습니다. 항목의 *Comment* 탭으로 가거나, 항목을 오른쪽 클릭하면 됩니다. 이 탭에서 개별 모델 주석의 색상을 직접 설정할 수도 있습니다. 주석은 모델 설계자 캔버스에서만 가시화되며, 최종 알고리즘 대화창에는 표시되지 않습니다. *View* → *Show Comments* 옵션을 비활성화하면 주석을 숨길 수 있습니다.

Your model can be run in various ways:

- You can run the whole model by clicking on Run Model… from the toolbar, right-clicking on the model in the *Browser panel* or *Model* → *Run Model*…: when using the editor to execute a model, any non-default values will be saved in the inputs. This means that executing the model at a later time from the editor will have the dialog prefilled with those values on any subsequent run.
- You can select elements of the model and run only that subset of the model: press the *Run Selected Steps*…

option from the *Models* menu or from the contextual menu of a selected algorithm. The initial state will be taken from any previous executions of the model through the editor, so results from previous steps in the model are available for the selected steps. This makes it possible to fix parts of a large model, without having to constantly run the entire model to test. Especially useful when earlier steps in the model are time consuming.

- You can run a subset of the model, starting from a specific algorithm: right-click the algorithm and select *Run from Here*⋯. Likewise, values are taken from previous executions.
- In order to use the algorithm from the Processing Toolbox, it has to be saved and the modeler dialog closed, to allow the toolbox to refresh its contents.

### 사용자 모델 문서 작성하기

사용자는 사용자 모델을 문서화해야 하며, 모델 설계자 자체에서 문서를 작성할 수 있습니다. Edit model help 버튼을 누르면 다음 그림과 같은 대화창이 나타날 것입니다.

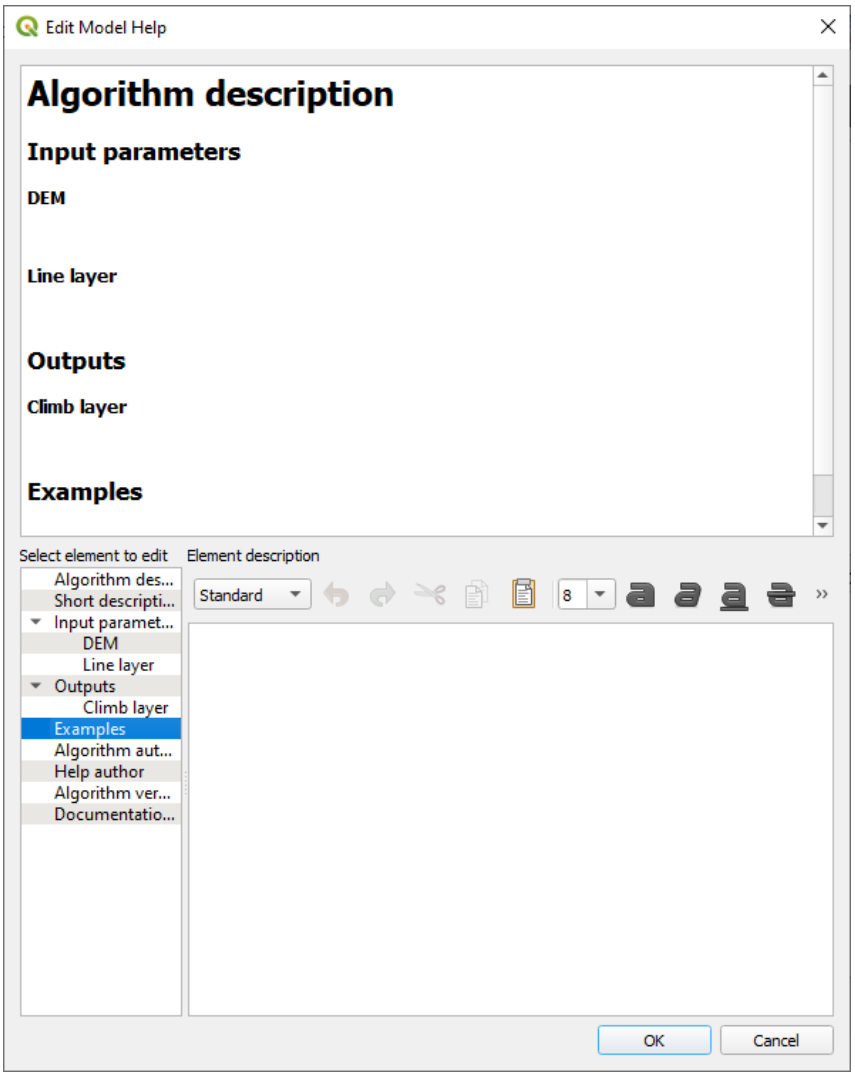



그림 23.28: 도움말 편집하기

대화창 오른쪽에는 간소한 HTML 페이지가 있습니다. 알고리즘의 입력 파라미터와 산출물에 대한 설명 그리고

모델 또는 모델의 작성자 같은 몇몇 추가 항목에 대한 일반적인 설명을 함께 사용해서 이 페이지의 내용을 생성합니다. 또한 모델의 활용법을 설명해주는 사용자 정의 예시를 입력할 수 있는 예시 부분도 있습니다. 처음 도움말 편집기를 열면, 이 모든 내용이 비어 있습니다. 그러나 대화창 왼쪽에 있는 항목들을 이용해서 이를 편집할 수 있습니다. 상단 부분에서 항목을 선택한 다음 하단 텍스트 란에 해당 항목에 대한 설명을 작성하십시오.

모델 도움말은 모델의 일부로 저장됩니다.

### 23.5.3 모델 저장 및 불러오기

#### 모델 저장하기


Save model 버튼을 클릭하면 현재 모델을 저장하고, 
Open Model 버튼을 누르면 이전에 저장한 모델을 불러올 수 있습니다. 모델은 `.model3` 확장자로 저장됩니다. 모델 설계자 대화창에서 이전에 저장한 모델인 경우, 저장시 파일명을 묻지 않을 것입니다. 이미 해당 모델과 연관된 파일이 있기 때문에, 이후 동일한 파일에 저장됩니다.

모델을 저장하기 전에, 대화창 상단에 있는 텍스트란에 모델의 이름 및 속하게 될 그룹을 입력해야 합니다.

`models` 폴더 (모델 저장시 파일명을 물어볼 때의 기본 폴더) 에 저장된 모델은 툴박스의 해당 그룹에 나타납니다. 툴박스를 열 때, `models` 폴더에서 `.model3` 확장자를 가진 파일을 찾아서 파일이 담고 있는 모델을 불러옵니다. 모델이 그 자체로 알고리즘이기 때문에, 다른 모든 알고리즘처럼 툴박스에 추가할 수 있습니다.


Save model in project 버튼을 클릭해서 프로젝트 파일 안에 모델을 저장할 수도 있습니다. 이 방법으로 저장한 모델은 디스크 상에 `.model3` 파일로 작성되지 않고, 프로젝트 파일 안에 내장될 것입니다.

공간 처리 툴박스의 
*Project models* 메뉴 및 *Project* → *Models* 메뉴 항목을 통해 프로젝트 모델을 사용할 수 있습니다.

*Processing Options* 환경 설정 대화창의 *Modeler* 그룹에서 *Model folder* 를 설정할 수 있습니다.

`models` 폴더에서 불러온 모델은 툴박스뿐만 아니라 모델 설계자 대화창의 *Algorithms* 탭에 있는 알고리즘 트리에도 나타납니다. 즉 모델에 다른 알고리즘을 추가하는 것과 마찬가지로 모델을 더 큰 모델의 일부분으로 포함시킬 수 있다는 뜻입니다.

모델은탐색기 패널에도 표시되며 해당 패널에서 실행할 수 있습니다.

#### 모델을 파이썬 스크립트로 내보내기

이 사용자 지침서의 이후 부분에서 설명하겠지만, QGIS 파이썬 콘솔에서 공간 처리 알고리즘을 호출할 수 있습니다. 파이썬을 이용해서 새 공간 처리 알고리즘을 생성할 수도 있습니다. 모델을 생성한 다음 이를 파이썬 파일로 내보내면 이런 파이썬 스크립트를 손쉽게 생성할 수 있습니다.

모델을 파이썬 스크립트 파일로 내보내려면, 모델 설계자 캔버스에서 
Export as Script Algorithm… 을 클릭하거나 공간 처리 툴박스에 있는 모델명을 오른쪽 클릭한 다음 
Export Model as Python Algorithm… 을 선택하십시오.

#### 모델을 이미지, PDF 또는 SVG 로 내보내기

Export as image, Export as PDF 또는 Export as SVG 를 클릭해서 모델을 SVG 또는 PDF 이미지로 (도해 (□□) 목적으로) 내보낼 수도 있습니다.

### 23.5.4 모델 편집

현재 생성 중인 모델을 편집해서 작업 흐름 및 모델 그 자체를 정의하는 알고리즘과 입력물 사이의 관계를 재정립할 수 있습니다.

You can also edit existing model from the *Browser panel* choosing *Edit Model…* option.

캔버스에 있는 알고리즘을 오른쪽 클릭하면, 다음 그림과 같은 컨텍스트 메뉴가 나타납니다:

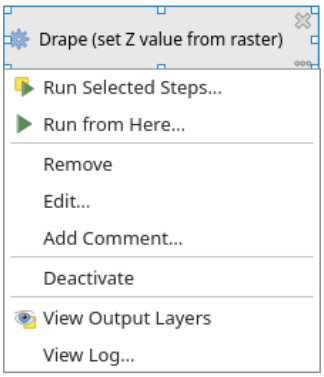



그림 23.29: 모델 설계자 오른쪽 클릭

- Options for *running part of the model* have been exposed earlier.
- Selecting the *Remove* option will cause the selected algorithm to be removed. An algorithm can be removed only if there are no other algorithms depending on it. That is, if no output from the algorithm is used as input in a different one.
- Selecting the *Edit…* option will show the parameter dialog so you can change the inputs and parameter values. Not all input elements available in the model will appear as available inputs. Layers or values generated at a more advanced step in the workflow defined by the model will not be available if they cause circular dependencies.

  새 값을 선택한 다음 이전과 같이 *OK* 버튼을 클릭하십시오. 모델 설계자 캔버스에서 모델 항목들 사이의 연결이 변경 사항에 맞춰 변경될 것입니다.
- *Add comment…*옵션을 선택하면 알고리즘의 습성을 자세히 설명하기 위한 코멘트를 추가할 수 있습니다.

> **참고**
>
> Right-clicking an input parameter will also allow you to *Remove*, *Edit…* and *Add comment…*

- 모델의 알고리즘 가운데 일부를 비활성화시키면 모델을 부분 실행시킬 수 있습니다. 알고리즘 항목을 오른쪽 클릭했을 때 나타나는 컨텍스트 메뉴에 있는 *Deactivate* 옵션을 선택하면 됩니다. 선택한 알고리즘 및 모델에서 해당 알고리즘에 종속된 모든 알고리즘이 회색조로 표시되고, 모델의 일부로서 실행되지 않을 것입니다.

  When right-clicking on an algorithm that is not active, you will see an *Activate* menu option that you can use to reactivate it.

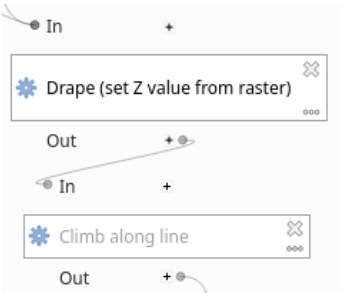



그림 23.30: Model with a deactivated algorithm

- When editing a model through the designer (and after having run that model), right-clicking any child step in the model allows to select *View Output Layers*. This will add the output layers from that step as new layers in the current QGIS project.

  This action is available for all child algorithms in the model, even if the model is not configured to use the outputs from those children as model outputs. This is designed as a helpful debugging action. If a model fails (or gives unexpected results), you can then trace through the model and view the outputs for suspected problematic steps. It avoids the need to add temporary outputs to a model and re-run to test.

  Additionally, this action is always available after running the model, even if the model itself failed (e.g., because of a misconfigured step later in the model).

- The *View Log* action helps you see the log for each child algorithm after you've closed down the algorithm dialog.

## 23.6 배치 공간 처리 인터페이스

### 23.6.1 소개

모든 (모델 포함) 알고리즘을 배치 (batch) 프로세스로 실행할 수 있습니다. 단일 입력물 집합이 아니라 여러 집합을 입력해서 필요한 만큼 여러 번 알고리즘을 실행할 수 있다는 뜻입니다. 대용량 데이터를 공간 처리하는 경우 이 배치 작업이 유리합니다. 툴박스에서 알고리즘을 여러 번 구동시킬 필요가 없기 때문입니다.

알고리즘을 배치 프로세스로 실행하려면, 툴박스에서 알고리즘명을 오른쪽 클릭한 다음 팝업 메뉴에서 *Execute as batch process* 옵션을 선택하십시오.

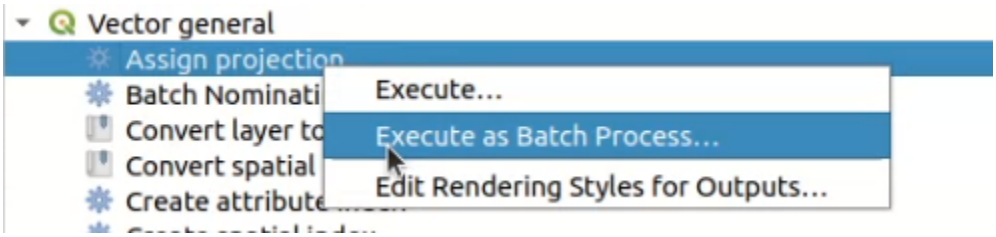



그림 23.31: 오른쪽 클릭으로 배치 프로세스 실행

선택한 알고리즘의 실행 대화창이 열리면, **[Run as batch process…]** 버튼을 눌러 배치 프로세스 작업 인터페이스도 열 수 있습니다.

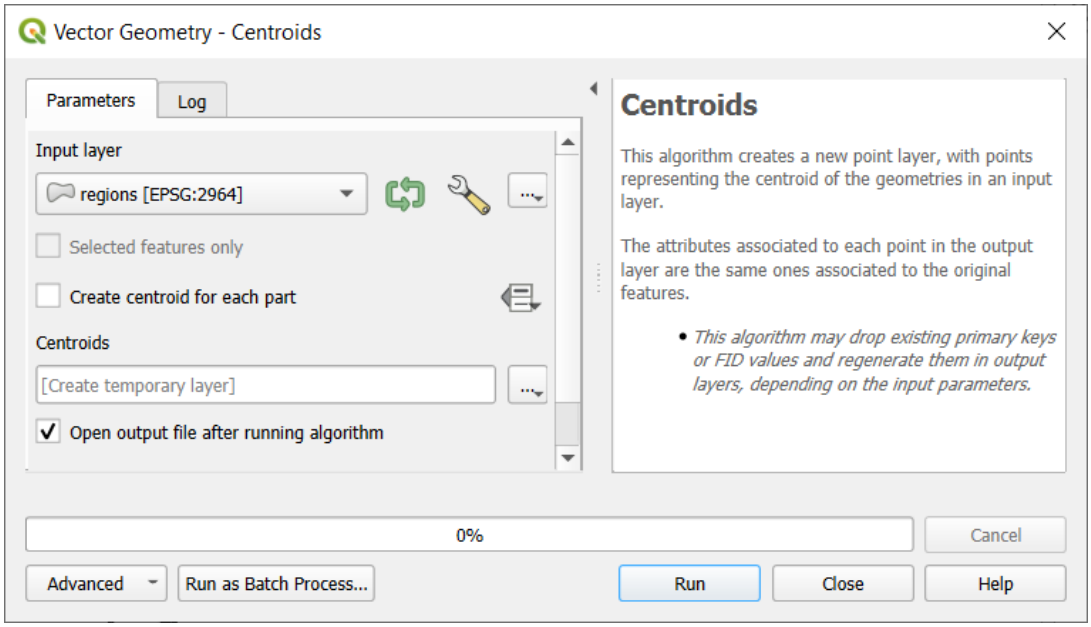



그림 23.32: 알고리즘 대화창에서 배치 프로세스 작업 시작

### 23.6.2 파라미터 표

배치 프로세스를 실행하는 작업은 단일 알고리즘을 실행하는 작업과 비슷합니다. 파라미터 값을 정의해야 하는데, 이 경우에는 각 파라미터 별로 단일 값이 아니라 알고리즘 반복 실행 횟수만큼 값을 하나씩 설정해줘야 합니다. 각 행이 반복 실행이고 각 열이 알고리즘의 파라미터인 다음 그림과 같은 표를 통해 이 값들을 설정합니다.

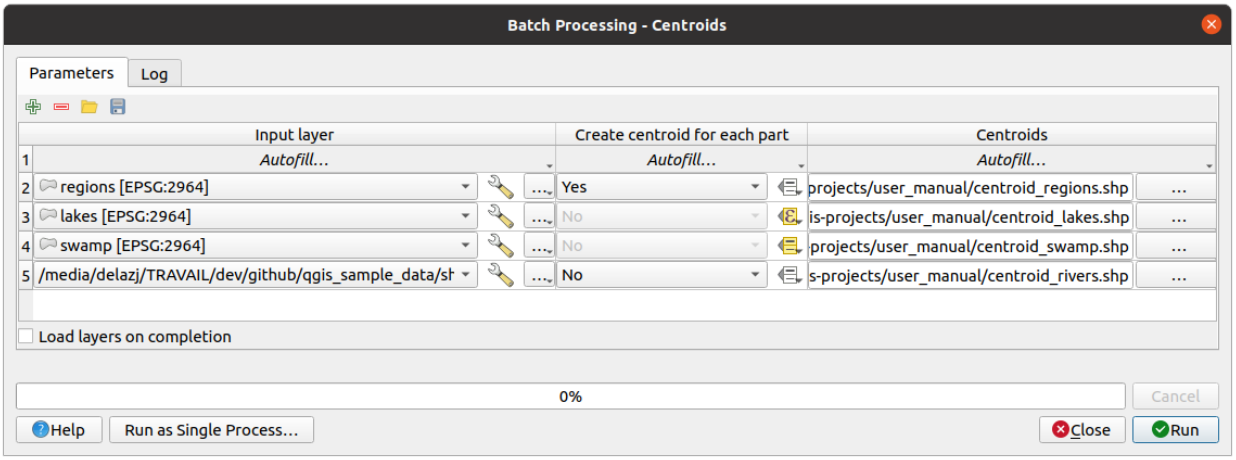



그림 23.33: 배치 프로세스 작업

최상단 툴바에서 다음과 같은 작업을 할 수 있습니다:

- Toggle advanced mode: 고급이라고 표시된 파라미터를 가진 알고리즘에만 사용할 수 있는 이 버튼은 배치 대화창에서 고급 파라미터를 보이거나 숨길 수 있게 해줍니다.
- Add row: 환경설정에 새로운 배치 프로세스 항목을 추가합니다.
- Remove row(s): 표로부터 선택한 행들을 제거합니다. 왼쪽에 있는 숫자를 클릭하면 행을 선택할 수 있으며,

다중 선택을 위한키보드 조합 을 사용할 수 있습니다.

- Open: 배치 프로세스 환경설정 파일을 엽니다.
- Save: 다음에 실행할 수 있도록 `.JSON` 파일에 배치 프로세스 환경설정을 저장합니다.

기본적으로, 파라미터 표는 2 개의 행만 담고 있습니다:

- 각 셀에 있는 첫 번째 행은 아래 셀들을 빠르게 채울 수 있는옵션 을 가진 *Autofill…* 드롭다운 메뉴를 표시합니다. 파라미터 유형에 따라 사용할 수 있는 옵션들이 달라집니다.
- 두 번째 행 (은 물론 이어지는 각 행) 은 알고리즘의 단일 실행을 나타내고, 각 셀은 파라미터 가운데 하나의 값을 담게 됩니다. 툴박스에서 알고리즘을 실행할 때 보게 되는 파라미터 대화창과 본질적으로 비슷하지만, 구조가 다릅니다.

표의 최하단에 있는 *Load layers on completion* 옵션을 이용해서 실행 완료 시 레이어를 불러올지 여부를 설정할 수 있습니다.

표의 행 수를 (알고리즘 실행 횟수만큼) 설정하고 나면, 원하는 값을 입력해야 합니다.

### 23.6.3 파라미터 표 채우기

대부분의 파라미터의 경우, 값을 설정하는 일은 사소한 작업입니다. 단일 프로세스 대화창 에서와 마찬가지로 적절한 위젯이 제공되어, 파라미터 유형에 따라 값을 입력하거나 또는 사용할 수 있는 값들의 목록에서 값을 선택할 수 있습니다. 사용할 수 있는 경우 데이터 정의 위젯도 포함됩니다.

셀별로 표를 채우는 작업을 피하기 위해 배치 프로세스 정의를 자동화하려면 파라미터의 *Autofill…*메뉴를 눌러서 열에 있는 값들을 대체하기 위한 다음 옵션들 가운데 하나를 선택할 수도 있습니다:

- *Fill Down*: 첫 번째 프로세스의 입력을 가져와서 다른 모든 프로세스에 입력할 것입니다.
- *Calculate by Expression…*: 해당 열 안에 있는 모든 기존 값들을 업데이트하는 데 사용할 수 있는 새로운 QGIS 표현식을 생성할 수 있습니다. 변수 를 통해 (다른 열의 기존 값들을 포함하는) 기존 파라미터 값들을 표현식 안에 사용할 수 있습니다. 예를 들면 각 레이어의 버퍼 거리를 기반으로 선분 개수를 설정할 수도 있습니다:

```
CASE WHEN @DISTANCE > 20 THEN 12 ELSE 8 END
```

- *Add Values by Expression…*: (기존 행에만 적용되는 *Calculate by Expression…*과는 반대로) 배열을 반환하는 표현식으로부터 나오는 값들을 사용해서 새 행들을 추가합니다. 의도한 사용례는 복소수열 (complex numeric series) 을 사용해서 배치 대화창을 채울 수 있도록 하는 것입니다. 예를 들어 거리 파라미터에 대해 표현식 `generate_series(100, 1000, 50)` 을 사용해서 배치 버퍼에 행을 추가하면 100, 150, 200, …. 1000 값을 가진 새 행들을 추가할 것입니다.
- 파일 또는 레이어 파라미터를 설정하는 경우, 더 많은 옵션이 제공됩니다:
  - *Add Files by Pattern…*: 검색 (*Look in*) 한 폴더에서 *File pattern* 과 일치하는 파일마다 표에 새로운 행을 추가합니다. 예를 들어 `*.shp` 패턴은 목록에 폴더의 모든 `SHP` 파일을 추가할 것입니다. 하위 폴더도 탐색하려면 *Search recursively* 옵션을 체크하십시오.
  - *Select Files…*: 디스크 상에서 파일을 개별적으로 선택합니다.
  - *Add All Files from a Directory…*: 선택한 디렉터리에 있는 모든 파일마다 표에 새로운 행을 추가합니다.
  - *Select from Open Layers…*: 활성화 프로젝트에 열려 있는 레이어 (들) 에서 선택합니다.

산출 데이터 파라미터는 알고리즘을 단일 프로세스로 실행할 때와 동일한 케이퍼빌리티를 노출시킵니다. 알고리즘에 따라 산출물을 다음과 같이 산출할 수 있습니다:

- 셀을 비워두는 경우 산출물을 건너뛸 수 있습니다.

- saved as a temporary layer: fill the cell with your chosen output name, select *Create Temporary Layer* from the ⋯ drop-down, and remember to tick the *Load layers on completion* checkbox. For temporary layers, the value you provide will be used as the layer name. You can use the *Autofill* options to construct that name.
- saved as a plain file (`.SHP`, `.GPKG`, `.XML`, `.PDF`, `.JPG`,⋯): choose *Select File/Folder*⋯ from the ⋯ drop-down. For plain files, you can set the path using the *Autofill* options exposed beforehand.

  E.g. use *Calculate by Expression*⋯ to set output file names to complex expressions like:

```
'/home/me/stuff/buffer_' || left(@INPUT, 30) || '_' || @DISTANCE || '.shp'
```

  You can also type the file path directly or use the file chooser dialog that appears when clicking on the accompanying *Select File/Folder*⋯ button. Once you select the file, a new dialog is shown to allow for auto-completion of other cells in the same column (same parameter).

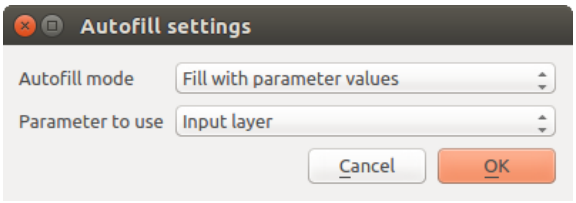



그림 23.34: 배치 프로세스 작업 - 저장

  기본값 (*Do not autofill*) 을 선택하는 경우, 그냥 파라미터 표에서 선택한 셀에 선택한 파일명을 입력할 것입니다. 다른 옵션을 선택하면, 선택한 셀 **아래** 에 있는 모든 셀에 정의한 기준에 따라 파일명을 입력할 것입니다.

  - *Fill with numbers*: 파일 이름에 증분 숫자를 추가합니다.
  - *Fill with parameter values*: 파일 이름에 동일한 행에 있는 파라미터 값을 추가할 파라미터를 선택할 수 있습니다. 산출 데이터 객체를 입력 객체에 따라 각각 명명하는 경우 특히 유용한 옵션입니다.

- 데이터베이스 컨테이너 내부에 레이어로 저장할 수 있습니다:

```
# Indicate a layer within a GeoPackage file
ogr:dbname='C:/Path/To/Geopackage.gpkg' table="New_Table" (geom)

# Use the "Calculate By Expression" to output to different layers in a GeoPackage
'ogr:dbname=\'' || @project_folder || '/Buffers.gpkg\' table="' || @INPUT || '_'␣
↪|| @DISTANCE || '" (geom)'
```

### 23.6.4 배치 프로세스 실행

필요한 모든 값을 설정한 다음 배치 프로세스를 실행하려면, *Run* 버튼만 누르면 됩니다. *Log* 패널이 활성화되어 실행 과정의 상세 정보 및 단계들을 출력합니다. 대화창 하단에 있는 진행 상태 막대에 전체 배치 작업의 진행도가 표시될 것입니다.

## 23.7 콘솔에서 공간 처리 알고리즘 사용

고급 사용자는 콘솔을 통해 생산성을 향상시키는 것은 물론, 공간 처리 프레임워크의 다른 모든 GUI 항목을 통해서도 수행할 수 없는 복잡한 작업을 수행할 수 있습니다. 명령 줄 인터페이스를 사용해서 여러 알고리즘을 연결한 모델을 정의하고, 더욱 유연하고 강력한 작업 흐름을 생성하기 위한 반복문 (`for`) 및 조건문 (`if`) 과 같은 부가적인 작업을 추가할 수 있습니다.

QGIS 는 공간 처리 콘솔을 내장하고 있지 않지만, 그 대신 QGIS 에 내장된파이썬 콘솔 에서 모든 공간 처리 명령어를 사용할 수 있습니다. 즉 사용자의 콘솔 작업에 공간 처리 명령어를 포함시켜 콘솔에서 사용할 수 있는 다른 (QGIS API 가 지원하는 메소드를 포함한) 모든 기능에 공간 처리 알고리즘을 연결할 수 있다는 의미입니다.

사용자가 파이썬 콘솔에서 실행하는 코드는, 비록 어느 특정한 공간 처리 메소드를 호출하지 않더라도, 다른 모든 알고리즘과 마찬가지로 향후 툴박스, 모델 설계자, 또는 기타 구성 요소에서 호출할 수 있는 새 알고리즘으로 변환할 수 있습니다. 사실, 툴박스에서 찾아볼 수 있는 알고리즘 가운데 일부는 단순한 스크립트일 뿐입니다.

이 절에서 QGIS 파이썬 콘솔에서 공간 처리 알고리즘을 사용하는 방법과 함께 파이썬으로 알고리즘을 작성하는 방법에 대해 설명하겠습니다.

### 23.7.1 파이썬 콘솔에서 알고리즘 호출하기

사용자는 가장 먼저 공간 처리 함수를 다음 명령어로 가져와야 합니다:

```
>>> from qgis import processing
```

이제 사용자가 이 콘솔을 통해 할 수 있는 (흥미로운) 일은 기본적으로 단 하나뿐입니다. 알고리즘을 실행하는 것이죠. `run()` 메소드를 사용하면 알고리즘을 실행할 수 있습니다. 첫 번째 파라미터로 실행할 알고리즘명을 그리고 해당 알고리즘의 요구 사항에 따라 다양한 추가 파라미터들을 받아 알고리즘을 실행하는 메소드입니다. 따라서 사용자가 가장 먼저 알아야 할 점은 실행할 알고리즘의 이름입니다. 이 이름은 툴박스에서 찾아볼 수 있는 이름이 아니며, 오히려 유일한 명령 줄 이름에 가깝습니다. 사용자 알고리즘의 정확한 이름을 찾으려면 `processingRegistry` 를 사용하면 됩니다. 사용자 콘솔에 다음 명령어를 입력하십시오:

```
>>> for alg in QgsApplication.processingRegistry().algorithms():
        print(alg.id(), "->", alg.displayName())
```

다음과 같은 화면을 보게 될 겁니다. (가독성을 향상시키기 위해 점선이 추가될 수도 있습니다.)

```
3d:tessellate --------------> Tessellate
gdal:aspect ----------------> Aspect
gdal:assignprojection ------> Assign projection
gdal:buffervectors ---------> Buffer vectors
gdal:buildvirtualraster ----> Build Virtual Raster
gdal:cliprasterbyextent ----> Clip raster by extent
gdal:cliprasterbymasklayer -> Clip raster by mask layer
gdal:clipvectorbyextent ----> Clip vector by extent
gdal:clipvectorbypolygon ---> Clip vector by mask layer
gdal:colorrelief -----------> Color relief
gdal:contour ---------------> Contour
gdal:convertformat ---------> Convert format
gdal:dissolve --------------> Dissolve
...
```

이 목록은 사용 가능한 모든 알고리즘 ID 를 제공자명과 알고리즘명으로 정렬하고 대응하는 명령 줄 이름과 함께 정리한 것입니다.

알고리즘의 명령 줄 이름을 찾았다면, 다음은 해당 알고리즘을 실행하기 위한 올바른 문법을 알아내야 합니다. 즉 `run()` 메소드 호출 시 어떤 파라미터가 필요한지 알고 있어야 한다는 뜻입니다.

알고리즘이 요구하는 파라미터 목록 및 알고리즘이 생성하는 산출물에 관한 정보를 얻을 수 있도록 알고리즘을 상세히 설명해주는 메소드가 존재합니다. 이런 정보를 얻으려면, `algorithmHelp(id_of_the_algorithm)` 메소드를 사용하면 됩니다. 알고리즘의 서술적인 전체 이름이 아니라 알고리즘 ID 를 사용하십시오.

이 메소드를 `native:buffer` 파라미터와 (`qgis:buffer` 는 `native:buffer` 의 별명으로, 똑같이 작동할 것입니다) 함께 호출하면, 다음과 같은 설명을 보게 됩니다:

```
>>> processing.algorithmHelp("native:buffer")
Buffer (native:buffer)

This algorithm computes a buffer area for all the features in an
input layer, using a fixed or dynamic distance.

The segments parameter controls the number of line segments to
use to approximate a quarter circle when creating rounded
offsets.

The end cap style parameter controls how line endings are handled
in the buffer.

The join style parameter specifies whether round, miter or
beveled joins should be used when offsetting corners in a line.

The miter limit parameter is only applicable for miter join
styles, and controls the maximum distance from the offset curve
to use when creating a mitered join.


----------------
Input parameters
----------------

INPUT: Input layer

   Parameter type: QgsProcessingParameterFeatureSource

   Accepted data types:
           - str: layer ID
           - str: layer name
           - str: layer source
           - QgsProcessingFeatureSourceDefinition
           - QgsProperty
           - QgsVectorLayer

DISTANCE: Distance

   Parameter type: QgsProcessingParameterDistance

   Accepted data types:
           - int
           - float
           - QgsProperty

SEGMENTS: Segments

   Parameter type: QgsProcessingParameterNumber

   Accepted data types:
```

(다음 페이지에 계속)

(이전 페이지에서 계속)

```
            - int
            - float
            - QgsProperty

END_CAP_STYLE: End cap style

   Parameter type: QgsProcessingParameterEnum

   Available values:
            - 0: Round
            - 1: Flat
            - 2: Square

   Accepted data types:
            - int
            - str: as string representation of int, e.g. '1'
            - QgsProperty

JOIN_STYLE: Join style

   Parameter type: QgsProcessingParameterEnum

   Available values:
            - 0: Round
            - 1: Miter
            - 2: Bevel

   Accepted data types:
            - int
            - str: as string representation of int, e.g. '1'
            - QgsProperty

MITER_LIMIT: Miter limit

   Parameter type: QgsProcessingParameterNumber

   Accepted data types:
            - int
            - float
            - QgsProperty

DISSOLVE: Dissolve result

   Parameter type: QgsProcessingParameterBoolean

   Accepted data types:
            - bool
            - int
            - str
            - QgsProperty

OUTPUT: Buffered

   Parameter type: QgsProcessingParameterFeatureSink

   Accepted data types:
```

(다음 페이지에 계속)

(이전 페이지에서 계속)

```
            - str: destination vector file, e.g. 'd:/test.shp'
            - str: 'memory:' to store result in temporary memory layer
            - str: using vector provider ID prefix and destination URI,
                   e.g. 'postgres:...' to store result in PostgreSQL table
            - QgsProcessingOutputLayerDefinition
            - QgsProperty

----------------
Outputs
----------------

OUTPUT:  <QgsProcessingOutputVectorLayer>
   Buffered
```

이제 알고리즘을 실행하기 위해 필요한 모든 내용을 알았습니다. 앞에서 언급했듯이, 알고리즘을 실행할 수 있는 명령어는 `run()` 입니다. 그 문법은 다음과 같습니다:

```
>>> processing.run(name_of_the_algorithm, parameters)
```

'parameters'는 사용자가 실행하려는 알고리즘에 따라 달라지는 파라미터 목록인데, `algorithmHelp()` 메소드가 제공하는 목록과 정확히 일치합니다.

```
 >>> processing.run("native:buffer", {'INPUT': '/data/lines.shp',
                'DISTANCE': 100.0,
                'SEGMENTS': 10,
                'DISSOLVE': True,
                'END_CAP_STYLE': 0,
                'JOIN_STYLE': 0,
                'MITER_LIMIT': 10,
                'OUTPUT': '/data/buffers.shp'})
```

써도 되고 안 써도 되는 파라미터인데 사용하고 싶지 않은 경우, 목록에 포함시키지 마십시오.

파라미터를 지정하지 않는 경우, 기본값을 사용할 것입니다.

파라미터의 유형에 따라, 값을 서로 다르게 입력해야 합니다. 다음은 각 입력 파라미터 유형 별로 값을 입력하는 방법을 간단하게 설명하는 목록입니다:

- 래스터 레이어, 벡터 레이어, 또는 테이블: 그냥 사용할 데이터 객체를 식별하는 이름 (QGIS 범례에 표시되는 이름) 또는 파일명 (해당 레이어가 열려 있지 않은 경우, 레이어를 열지만 맵 캔버스에는 추가하지 않습니다)을 나타내는 문자열을 입력하십시오. 레이어를 나타내는 QGIS 객체 인스턴스가 있을 경우, 해당 인스턴스도 파라미터로 넘겨줄 수 있습니다.
- 집합 (enumeration): 알고리즘이 집합 파라미터를 받을 경우, 해당 파라미터의 값을 정수 값으로 입력해야 합니다. 사용할 수 있는 옵션을 알고 싶다면, 앞의 예시에서처럼 `algorithmHelp()` 명령어를 사용하면 됩니다. 예를 들어 `native:buffer` 알고리즘은 JOIN_STYLE 이라는 집합을 받습니다:

```
JOIN_STYLE: Join style

   Parameter type: QgsProcessingParameterEnum

   Available values:
           - 0: Round
           - 1: Miter
           - 2: Bevel

   Accepted data types:
```

(다음 페이지에 계속)

(이전 페이지에서 계속)

```
            - int
            - str: as string representation of int, e.g. '1'
            - QgsProperty
```

이 경우, 옵션 3 개 가운데 하나를 이 파라미터로 입력할 수 있습니다. 순서가 0 부터 시작한다는 점을 기억하십시오.

- 불 (boolean): `True` 또는 `False` 를 사용하십시오.
- 다중 입력: 쌍반점 (`;`) 으로 구분된 입력 서술자 (descriptor) 를 나타내는 문자열을 입력하십시오. 단일 레이어 또는 테이블의 경우, 데이터 객체의 이름 또는 파일 경로를 각 입력 서술자로 사용할 수 있습니다.
- XXX 테이블의 필드: 사용할 필드의 이름을 나타내는 문자열을 입력하십시오. 이 파라미터는 대소문자를 구별합니다.
- 고정 테이블: 큰따옴표 (`"`) 로 감싸고 쉼표 (`,`) 로 구분한 모든 테이블 값의 목록을 입력하십시오. 이때 값은 상단 행부터 왼쪽에서 오른쪽 방향으로 입력합니다. 또는 테이블을 나타내는 값의 2 차원 배열을 사용할 수도 있습니다.
- CRS: 원하는 좌표계의 EPGS 코드 번호를 입력하십시오.
- 범위 (extent): `xmin`, `xmax`, `ymin` 및 `ymax` 값을 나타내는 쉼표 (`,`) 로 구분한 문자열을 입력해야 합니다.

불 (boolean), 파일, 문자열 및 숫자 파라미터에 대해서는 따로 설명이 필요없습니다.

문자열, 불, 또는 숫자 값과 같은 입력 파라미터에는 기본값이 있습니다. 해당 파라미터 항목이 빠져 있을 경우 기본값을 사용합니다.

산출 데이터 객체의 경우, 툴박스에서와 마찬가지로 산출물을 저장하기 위한 파일 경로를 입력하십시오. 산출물 객체를 지정하지 않을 경우 산출물을 임시 파일로 저장합니다. (또는 부가적인 산출물일 경우 건너뜁니다.) 파일의 확장자가 파일 포맷을 결정합니다. 알고리즘이 지원하지 않는 파일 확장자를 입력할 경우 해당 산출물 유형의 기본 파일 포맷을 사용하며, 입력한 파일 경로 끝에 해당 확장자를 덧붙일 것입니다.

툴박스에서 알고리즘을 실행할 때와는 달리, 파이썬 콘솔에서 `run()` 메소드를 사용해서 동일한 알고리즘을 실행해도 산출물이 QGIS 맵 캔버스에 추가되지 않습니다. 이 메소드는 (알고리즘 설명에 보이는) 하나 이상의 산출물 이름을 가진 딕셔너리를 키로 그리고 이 산출물들의 파일 경로를 값으로 반환합니다:

```
>>> myresult = processing.run("native:buffer", {'INPUT': '/data/lines.shp',
                'DISTANCE': 100.0,
                'SEGMENTS': 10,
                'DISSOLVE': True,
                'END_CAP_STYLE': 0,
                'JOIN_STYLE': 0,
                'MITER_LIMIT': 10,
                'OUTPUT': '/data/buffers.shp'})
>>> myresult['OUTPUT']
/data/buffers.shp
```

그러면 프로젝트에 산출물을 다음과 같이 다른 모든 레이어처럼 불러올 수 있습니다:

```
>>> buffered_layer = myresult['OUTPUT']
>>> QgsProject.instance().addMapLayer(buffered_layer)
```

프로젝트에 공간 처리 산출물을 즉시 불러오려면, `run()` 대신 `runAndLoadResults()` 메소드를 사용하면 됩니다.

```
>>> processing.runAndLoadResults("native:buffer", {parameters:values})
```

콘솔에서 알고리즘 대화창을 열고 싶다면 createAlgorithmDialog 메소드를 사용하면 됩니다. 필수적인 파라미터는 알고리즘명 하나뿐이지만, 파라미터 목록도 정의해서 대화창의 파라미터를 자동으로 채울 수도 있습니다:

```
>>> my_dialog = processing.createAlgorithmDialog("native:buffer", {
                'INPUT': '/data/lines.shp',
                'DISTANCE': 100.0,
                'SEGMENTS': 10,
                'DISSOLVE': True,
                'END_CAP_STYLE': 0,
                'JOIN_STYLE': 0,
                'MITER_LIMIT': 10,
                'OUTPUT': '/data/buffers.shp'})
>>> my_dialog.show()
```

execAlgorithmDialog 메소드는 대화창을 즉시 엽니다:

```
>>> processing.execAlgorithmDialog("native:buffer", {
                'INPUT': '/data/lines.shp',
                'DISTANCE': 100.0,
                'SEGMENTS': 10,
                'DISSOLVE': True,
                'END_CAP_STYLE': 0,
                'JOIN_STYLE': 0,
                'MITER_LIMIT': 10,
                'OUTPUT': '/data/buffers.shp'})
```

### 23.7.2 스크립트 생성 및 툴박스에서 실행

파이썬 코드를 작성하면 사용자 고유의 알고리즘을 생성할 수 있습니다. 공간 처리 스크립트는 QgsProcessingAlgorithm 클래스를 확장하기 때문에, 필수 함수를 구현하는 코드를 몇 줄 추가해야 합니다. 공간 처리 툴박스 상단의 *Scripts* 드롭다운 메뉴에 *Create new script* (백지 상태) 및 *Create New Script from Template* (QgsProcessingAlgorithm 클래스의 필수 함수를 위한 코드를 포함하는 템플릿) 메뉴가 있습니다. 공간 처리 스크립트 편집기가 열리면 사용자의 코드를 입력해야 합니다. 이 편집기에서 scripts 폴더 (파일 저장 대화창의 기본 폴더) 에 스크립트를 .py 확장자를 붙여 저장하면 대응하는 알고리즘을 생성할 것입니다.

사용자의 코드 안에서 알고리즘의 (툴박스에 표시될) 이름을 정의합니다.

다음 코드를 한번 살펴봅시다. 이 코드는 먼저 레이어를 평탄화한 다음, 사용자가 지정한 벡터 레이어 상에 사용자 정의 버퍼 거리로 버퍼 작업을 수행하는 공간 처리 알고리즘을 정의합니다:

```
from qgis.core import (QgsProcessingAlgorithm,
       QgsProcessingParameterNumber,
       QgsProcessingParameterFeatureSource,
       QgsProcessingParameterFeatureSink)

from qgis import processing

class algTest(QgsProcessingAlgorithm):
    INPUT_BUFFERDIST = 'BUFFERDIST'
    OUTPUT_BUFFER = 'OUTPUT_BUFFER'
    INPUT_VECTOR = 'INPUT_VECTOR'

    def __init__(self):
        super().__init__()

```

(다음 페이지에 계속)

(이전 페이지에서 계속)

```
    def name(self):
        return "algTest"

    def displayName(self):
        return "algTest script"

    def createInstance(self):
        return type(self)()

    def initAlgorithm(self, config=None):
        self.addParameter(QgsProcessingParameterFeatureSource(
            self.INPUT_VECTOR, "Input vector"))
        self.addParameter(QgsProcessingParameterNumber(
            self.INPUT_BUFFERDIST, "Buffer distance",
            QgsProcessingParameterNumber.Double,
            100.0))
        self.addParameter(QgsProcessingParameterFeatureSink(
            self.OUTPUT_BUFFER, "Output buffer"))

    def processAlgorithm(self, parameters, context, feedback):
        #DO SOMETHING
        algresult = processing.run("native:smoothgeometry",
            {'INPUT': parameters[self.INPUT_VECTOR],
             'ITERATIONS':2,
             'OFFSET':0.25,
             'MAX_ANGLE':180,
             'OUTPUT': 'memory:'},
            context=context, feedback=feedback, is_child_algorithm=True)
        smoothed = algresult['OUTPUT']
        algresult = processing.run('native:buffer',
            {'INPUT': smoothed,
            'DISTANCE': parameters[self.INPUT_BUFFERDIST],
            'SEGMENTS': 5,
            'END_CAP_STYLE': 0,
            'JOIN_STYLE': 0,
            'MITER_LIMIT': 10,
            'DISSOLVE': True,
            'OUTPUT': parameters[self.OUTPUT_BUFFER]},
            context=context, feedback=feedback, is_child_algorithm=True)
        buffered = algresult['OUTPUT']
        return {self.OUTPUT_BUFFER: buffered}
```

필요한 클래스들을 가져온 다음, 다음과 같은 `QgsProcessingAlgorithm` 함수들을 지정합니다:

- `name()`: 알고리즘 ID (소문자)
- `displayName()`: 사람이 읽을 수 있는 알고리즘 이름
- `createInstance()`: 알고리즘 클래스용 새 인스턴스 생성
- `initAlgorithm()`: 'parameterDefinitions'및'outputDefinitions'환경 설정

  여기에 알고리즘의 파라미터와 산출물을 설명합니다. 이 경우, 입력물용 피처 소스, 산출물용 피처 싱크(sink), 그리고 버퍼 거리용 숫자 값이 되겠군요.

- `processAlgorithm()`: 작업 실행

  먼저 도형을 평탄화시키기 위해 `smoothgeometry` 알고리즘을 실행한 다음, 평탄화된 산출물에 `buffer` 알고리즘을 실행합니다. 어떤 알고리즘 내부에서 다른 알고리즘을 실행시키려면 `is_child_algorithm`

인자를 `True` 로 설정해야 합니다. 입력 및 산출 파라미터를 어떻게 `smoothgeometry` 및 `buffer` 알고리즘에 파라미터로 사용하는지 볼 수 있을 겁니다.

입력물과 산출물로 사용할 수 있는 서로 다른 파라미터 유형이 많이 있습니다. 공간 처리 알고리즘의 입력물 및 산출물 유형 에서 전체 목록을 찾아볼 수 있습니다.

구성자 (constructor) 가 받는 첫 번째 파라미터는 파라미터의 이름이며, 두 번째는 파라미터의 (사용자 인터페이스용) 설명입니다. 나머지 구성자 파라미터들은 파라미터 유형에 특화되어 있습니다.

`QgsProcessingAlgorithm` 클래스의 `parameterAs` 함수를 사용하면 입력물을 QGIS 클래스로 변환할 수 있습니다. 예를 들어 버퍼 거리를 지정하는 숫자를 실수형 (double) 으로 받으려면:

```
self.parameterAsDouble(parameters, self.INPUT_BUFFERDIST, context)).
```

`processAlgorithm` 함수는 알고리즘이 정의한 모든 산출물에 대한 값을 담고 있는 목록을 반환해야 합니다. 이렇게 해서 동일 모델 안에 있는 다른 알고리즘을 포함한 다른 알고리즘이 이 산출물들에 접근할 수 있게 됩니다.

제대로 동작하는 알고리즘은 의미가 통하는만큼의 산출물을 정의하고 반환해야 합니다. 숫자 및 문자열 같은 피처가 아닌 산출물은 사용자 알고리즘을 더 큰 모델의 일부로 실행하는 경우 아주 유용합니다. 그 값을 모델 내부의 바로 다음 알고리즘의 입력 파라미터로 사용할 수 있기 때문입니다. 공간 처리된 피처의 개수, 실행중 찾아낸 무결하지 않은 피처의 개수, 피처 산출물의 개수 등과 같은 값을 위해 숫자형 산출물을 추가하는 것도 좋겠죠. 반환되는 산출물이 많을수록, 사용자 알고리즘도 더 유용해지니까요!

### 피드백

`processAlgorithm()` 메소드에 넘겨지는 `feedback()` 객체는 사용자 피드백/대화형 작업에 사용되어야 합니다. `feedback()` 객체의 `setProgress()` 함수를 사용하면 알고리즘의 진행도를 알려주는 진행 과정 막대를 (0 에서 100 사이의 값으로) 업데이트할 수 있습니다. 사용자 알고리즘이 완료까지 오래 걸리는 경우 매우 유용한 함수입니다.

`feedback()` 객체는 사용자가 알고리즘을 취소할 수 있게 하려면 감시해야 하는 `isCanceled()` 메소드를 제공합니다. `feedback()` 객체의 `pushInfo()` 메소드를 사용해서 사용자에게 정보를 전달할 수 있고, `reportError()` 메소드는 사용자에게 심각하지 않은 (non-fatal) 오류를 푸시하는 데 편리합니다.

알고리즘은 가급적 서술을 인쇄하거나 `QgsMessageLog` 에 로그를 작성하는 등의 다른 형태로 사용자에게 피드백을 제공해서는 안 되며, 대신 언제나 피드백 객체를 사용해야 합니다. 피드백 객체를 사용하면 알고리즘이 장황한 로그 작업을 할 수 있으며, 스레드 편에서도 안전합니다. (알고리즘이 일반적으로 배경 스레드에서 실행된다는 점을 감안하면 아주 중요한 이유죠.)

### 오류 처리하기

사용자 알고리즘의 실행을 방해하는 오류가 발생한 경우, 예를 들어 입력값이 무결하지 않거나 다른 조건이 맞지 않아 복구되서는 안 되고 될 수도 없을 때, `QgsProcessingException` 을 발생시켜야 합니다. 예를 들면:

```
if feature['value'] < 20:
  raise QgsProcessingException('Invalid input value {}, must be >= 20'.format(feature[
↪'value']))
```

심각하지 않은 오류인 경우 (예를 들어 피처가 NULL 도형을 가진 경우) `QgsProcessingException` 을 발생시키지 말고 대신 `feedback.reportError()` 를 통해 오류를 보고만 한 다음 해당 피처를 건너뛰는 편이 좋습니다. 이렇게 하면 사용자 알고리즘이"모델 친화적 (model-friendly)"이 됩니다. 심각하지 않은 오류인 경우 전체 알고리즘 실행을 정지하지 않기 때문이죠.

#### 사용자 스크립트 문서 작성하기

모델의 경우와 마찬가지로, 사용자 스크립트를 위해 그 목적 및 사용법을 설명하는 추가 문서를 생성할 수 있습니다.

`QgsProcessingAlgorithm()` 이 그 목적을 위해 `helpString()`, `shortHelpString()` 및 `helpUrl()` 함수를 제공합니다. 사용자에게 더 많은 도움말을 제공하려면 이 함수들을 지정하거나 무시하십시오.

툴박스에 있는 알고리즘 위로 커서를 가져다대면 나타나는 툴팁에는 `shortDescription()` 을 사용합니다.

### 23.7.3 실행 전 및 실행 후 스크립트 후크

스크립트를 알고리즘을 실행하기 전, 실행한 후에 실행되는 실행 전 (pre-execution) 및 실행 후 (post-execution) 후크 (hook) 로 사용할 수도 있습니다. 후크는 알고리즘을 실행할 때마다 수행해야 하는 작업을 자동화하기 위해 사용됩니다.

후크의 문법은 앞에서 설명한 문법과 동일하지만, `alg` 라는 또다른 전체 수준 변수를 사용할 수 있습니다. `alg` 는 방금 실행된 (또는 곧 실행될) 알고리즘을 나타내는 변수입니다.

공간 처리 옵션 대화창의 *General* 그룹에서 *Pre-execution script* 및 *Post-execution script* 라는 두 옵션을 볼 수 있는데, 이 두 옵션에 각각의 경우 실행될 스크립트의 파일명을 입력할 수 있습니다.

## 23.8 명령줄에서 공간 처리 알고리즘 사용

QGIS 는 사용자가 QGIS 데스크탑 자체를 실행하지 않고서도 명령줄에서 공간 처리 알고리즘과 (내장되어 있는 또는 플러그인이 제공하는) 모델을 직접 실행할 수 있는 `QGIS Processing Executor` 라는 도구를 제공합니다.

명령줄에서 `qgis_process` 를 실행하면 다음과 같은 화면을 보게 될 것입니다:

```
QGIS Processing Executor - 3.35.0-Master 'Master' (3.35.0-Master)
Usage: C:\OSGeo4W\apps\qgis-dev\bin\qgis_process.exe [--help] [--version] [--json] [--
↪verbose] [--no-python] [--skip-loading-plugins] [command] [algorithm id, path to␣
↪model file, or path to Python script] [parameters]

Options:

  --help or -h            Output the help
  --version or -v         Output all versions related to QGIS Process
  --json                  Output results as JSON objects
  --verbose               Output verbose logs
  --no-python             Disable Python support (results in faster startup)
  --skip-loading-plugins  Avoid loading enabled plugins (results in faster startup)

Available commands:

  plugins            list available and active plugins
  plugins enable     enables an installed plugin. The plugin name must be specified,␣
↪e.g. "plugins enable cartography_tools"
  plugins disable    disables an installed plugin. The plugin name must be specified,␣
↪e.g. "plugins disable cartography_tools"
  list               list all available processing algorithms
  help               show help for an algorithm. The algorithm id or a path to a␣
↪model file must be specified.
  run                runs an algorithm. The algorithm id or a path to a model file␣
```

(다음 페이지에 계속)

(이전 페이지에서 계속)

```
↪and parameter values must be specified.
                    Parameter values are specified after -- with PARAMETER=VALUE␣
↪syntax.
                    Ordered list values for a parameter can be created by specifying␣
↪the parameter multiple times,
                    e.g. --LAYERS=layer1.shp --LAYERS=layer2.shp
                    Alternatively, a '-' character in place of the parameters␣
↪argument indicates that the parameters should be read from STDIN as a JSON object.
                    The JSON should be structured as a map containing at least the
↪"inputs" key specifying a map of input parameter values.
                    This implies the --json option for output as a JSON object.
                    If required, the ellipsoid to use for distance and area␣
↪calculations can be specified via the "--ELLIPSOID=name" argument.
                    If required, an existing QGIS project to use during the␣
↪algorithm execution can be specified via the "--PROJECT_PATH=path" argument.
                    When passing parameters as a JSON object from STDIN, these extra␣
↪arguments can be provided as an "ellipsoid" and a "project_path" key respectively.
```

**참고**

qgis_process 도구는 설치된 플러그인 가운데 metadata.txt 파일에 hasProcessingProvider=yes 를 명시한 플러그인만 식별하고, 활성화시키거나 불러올 수 있습니다.

**힌트**

시스템 상에서 창 관리자 (window manager) 를 통하지 않고 (예: 헤드리스 (headless) 서버) qgis_process 를 호출하기 전에 다음을 설정해줘야 합니다:

```
export QT_QPA_PLATFORM=offscreen
```

list 명령어를 사용하면 사용할 수 있는 모든 제공자 및 알고리즘의 목록을 얻을 수 있습니다.

```
qgis_process list
```

help 명령어를 사용하면 명령어 또는 알고리즘에 대한 자세한 정보를 얻을 수 있습니다.

```
qgis_process help qgis:regularpoints
```

run 명령어를 사용하면 알고리즘 또는 모델을 실행할 수 있습니다. 첫 번째 파라미터로 알고리즘의 이름 또는 모델을 가리키는 경로를 지정하십시오.

```
qgis_process run native:buffer -- INPUT=source.shp DISTANCE=2 OUTPUT=buffered.shp
```

파라미터가 값 목록을 입력받는 경우, 동일 변수를 여러번 설정하십시오.

```
qgis_process run native:mergevectorlayers -- LAYERS=input1.shp LAYERS=input2.shp␣
↪OUTPUT=merged.shp
```

알고리즘을 실행하는 도중에 텍스트 기반 피드백 막대가 표시되며, CTRL+C 단축키로 실행을 취소할 수 있습니다.

run 명령어는 다음과 같은 심화 파라미터들도 지원합니다.

- --json: 표준출력 (STDOUT) 산출물 서식을 JSON 구성 방식으로

- `--ellipsoid`: 타원체를 지정한 타원체로 설정
- `--distance_units`: 지정한 거리 단위를 사용
- `--area_units`: 지정한 면적 단위를 사용
- `--project_path`: 알고리즘을 실행하기 위해 지정한 프로젝트를 불러옵니다.

`qgis_process` 는 예를 들어 스스로를 알고리즘 용 파라미터 목록 (dictionary) 유형 객체로 지정하는 파라미터 유형 같은 복잡 입력 파라미터를 지원합니다. 복잡 파라미터가 표준입력 (STDIN) 을 통해 지정될 것이라는 사실을 나타내기 위해, `qgis_process` 명령어는 (일반적인 인자 목록 자리에 후행 `-` 가 있는) 다음 서식을 따라야만 합니다:

```
qgis_process run algorithmId -
```

JSON 객체는 입력 파라미터 값을 매핑하는"inputs"키를 담고 있어야만 합니다. 다음은 그 예시입니다:

```
echo "{'inputs': {'INPUT': 'my_shape.shp', 'DISTANCE': 5}}" | qgis_process run␣
↪native:buffer -
```

또한 이 JSON 객체는 거리 단위, 면적 단위, 타원체 및 프로젝트 경로 같은 추가 설정도 포함할 수 있습니다:

```
{
 'ellipsoid': 'EPSG:7019',
 'distance_units': 'feet',
 'area_units': 'ha',
 'project_path': 'C:/temp/my_project.qgs'
 'inputs': {'DISTANCE': 5, 'SEGMENTS': 8 ... }
}
```

표준입력 (STDIN) 을 통해 입력 파라미터를 지정한다는 것은 자동적으로 산출물을 `JSON` 포맷으로 산출한다는 것을 의미합니다.

## 23.9 새 공간 처리 알고리즘을 파이썬 스크립트로 작성하기

파이썬을 사용해서 공간 처리 알고리즘을 작성하기 위한 옵션이 2 개 있습니다.

- `QgsProcessingAlgorithm` 확장
- *@alg 장식자 (decorator)* 사용

QGIS 내부에서, *Processing Toolbox* 상단에 있는 *Scripts* 메뉴에서 *Create new script* 를 선택하면 사용자 코드를 작성할 수 있는 *Processing Script Editor* 가 열립니다. 이 작업을 단순화하려면, 동일한 메뉴에서 *Create new script from template* 을 선택해서 스크립트 템플릿을 바탕으로 사용자 코드를 작성할 수 있습니다. 이 메뉴는 `QgsProcessingAlgorithm` 을 확장하는 템플릿을 엽니다.

`scripts` 폴더 (기본 저장 위치) 에 스크립트를 `.py` 확장자를 붙여 저장하면, *Processing Toolbox* 에서 해당 알고리즘을 사용할 수 있습니다.

### 23.9.1 QgsProcessingAlgorithm 확장

다음 코드는:

1. 벡터 데이터를 입력물로 받습니다.
2. 피처의 개수를 셉니다.
3. 버퍼 작업을 수행합니다.
4. 버퍼 작업의 결과물로부터 래스터 레이어를 생성합니다.
5. 버퍼 레이어, 래스터 레이어, 그리고 피처의 개수를 반환합니다.

```
from qgis.PyQt.QtCore import QCoreApplication
from qgis.core import (QgsProcessing,
                       QgsProcessingAlgorithm,
                       QgsProcessingException,
                       QgsProcessingOutputNumber,
                       QgsProcessingParameterDistance,
                       QgsProcessingParameterFeatureSource,
                       QgsProcessingParameterVectorDestination,
                       QgsProcessingParameterRasterDestination)
from qgis import processing


class ExampleProcessingAlgorithm(QgsProcessingAlgorithm):
    """
    This is an example algorithm that takes a vector layer,
    creates some new layers and returns some results.
    """

    def tr(self, string):
        """
        Returns a translatable string with the self.tr() function.
        """
        return QCoreApplication.translate('Processing', string)

    def createInstance(self):
        # Must return a new copy of your algorithm.
        return ExampleProcessingAlgorithm()

    def name(self):
        """
        Returns the unique algorithm name.
        """
        return 'bufferrasterextend'

    def displayName(self):
        """
        Returns the translated algorithm name.
        """
        return self.tr('Buffer and export to raster (extend)')

    def group(self):
        """
        Returns the name of the group this algorithm belongs to.
        """
        return self.tr('Example scripts')
```

(다음 페이지에 계속)

(이전 페이지에서 계속)

```
    def groupId(self):
        """
        Returns the unique ID of the group this algorithm belongs
        to.
        """
        return 'examplescripts'

    def shortHelpString(self):
        """
        Returns a localised short help string for the algorithm.
        """
        return self.tr('Example algorithm short description')

    def initAlgorithm(self, config=None):
        """
        Here we define the inputs and outputs of the algorithm.
        """
        # 'INPUT' is the recommended name for the main input
        # parameter.
        self.addParameter(
            QgsProcessingParameterFeatureSource(
                'INPUT',
                self.tr('Input vector layer'),
                types=[QgsProcessing.TypeVectorAnyGeometry]
            )
        )
        self.addParameter(
            QgsProcessingParameterVectorDestination(
                'BUFFER_OUTPUT',
                self.tr('Buffer output'),
            )
        )
        # 'OUTPUT' is the recommended name for the main output
        # parameter.
        self.addParameter(
            QgsProcessingParameterRasterDestination(
                'OUTPUT',
                self.tr('Raster output')
            )
        )
        self.addParameter(
            QgsProcessingParameterDistance(
                'BUFFERDIST',
                self.tr('BUFFERDIST'),
                defaultValue = 1.0,
                # Make distance units match the INPUT layer units:
                parentParameterName='INPUT'
            )
        )
        self.addParameter(
            QgsProcessingParameterDistance(
                'CELLSIZE',
                self.tr('CELLSIZE'),
                defaultValue = 10.0,
                parentParameterName='INPUT'
```

(다음 페이지에 계속)

(이전 페이지에서 계속)

```
            )
        )
        self.addOutput(
            QgsProcessingOutputNumber(
                'NUMBEROFFEATURES',
                self.tr('Number of features processed')
            )
        )

    def processAlgorithm(self, parameters, context, feedback):
        """
        Here is where the processing itself takes place.
        """
        # First, we get the count of features from the INPUT layer.
        # This layer is defined as a QgsProcessingParameterFeatureSource
        # parameter, so it is retrieved by calling
        # self.parameterAsSource.
        input_featuresource = self.parameterAsSource(parameters,
                                                     'INPUT',
                                                     context)
        numfeatures = input_featuresource.featureCount()

        # Retrieve the buffer distance and raster cell size numeric
        # values. Since these are numeric values, they are retrieved
        # using self.parameterAsDouble.
        bufferdist = self.parameterAsDouble(parameters, 'BUFFERDIST',
                                            context)
        rastercellsize = self.parameterAsDouble(parameters, 'CELLSIZE',
                                                context)
        if feedback.isCanceled():
            return {}
        buffer_result = processing.run(
            'native:buffer',
            {
                # Here we pass on the original parameter values of INPUT
                # and BUFFER_OUTPUT to the buffer algorithm.
                'INPUT': parameters['INPUT'],
                'OUTPUT': parameters['BUFFER_OUTPUT'],
                'DISTANCE': bufferdist,
                'SEGMENTS': 10,
                'DISSOLVE': True,
                'END_CAP_STYLE': 0,
                'JOIN_STYLE': 0,
                'MITER_LIMIT': 10
            },
            # Because the buffer algorithm is being run as a step in
            # another larger algorithm, the is_child_algorithm option
            # should be set to True
            is_child_algorithm=True,
            #
            # It's important to pass on the context and feedback objects to
            # child algorithms, so that they can properly give feedback to
            # users and handle cancelation requests.
            context=context,
            feedback=feedback)

```

(다음 페이지에 계속)

(이전 페이지에서 계속)

```
        # Check for cancelation
        if feedback.isCanceled():
            return {}

        # Run the separate rasterization algorithm using the buffer result
        # as an input.
        rasterized_result = processing.run(
            'qgis:rasterize',
            {
                # Here we pass the 'OUTPUT' value from the buffer's result
                # dictionary off to the rasterize child algorithm.
                'LAYER': buffer_result['OUTPUT'],
                'EXTENT': buffer_result['OUTPUT'],
                'MAP_UNITS_PER_PIXEL': rastercellsize,
                # Use the original parameter value.
                'OUTPUT': parameters['OUTPUT']
            },
            is_child_algorithm=True,
            context=context,
            feedback=feedback)

        if feedback.isCanceled():
            return {}

        # Return the results
        return {'OUTPUT': rasterized_result['OUTPUT'],
                'BUFFER_OUTPUT': buffer_result['OUTPUT'],
                'NUMBEROFFEATURES': numfeatures}
```

공간 처리 알고리즘 표준 함수:

- **createInstance (필수)**
  사용자 알고리즘의 새 복사본을 반환해야만 합니다. 클래스명을 변경할 경우, 이 함수가 반환하는 값도 일치하도록 업데이트하는 것을 잊지 마세요!
- **name (필수)**
  알고리즘을 식별하는 데 사용되는 유일한 알고리즘명을 반환합니다.
- **displayName (필수)**
  변형된 (translated) 알고리즘명을 반환합니다.
- **group**
  이 알고리즘이 종속된 그룹의 이름을 반환합니다.
- **groupId**
  이 알고리즘이 종속된 그룹의 유일한 ID 를 반환합니다.
- **shortHelpString**
  알고리즘을 위한, 번역된 (localized) 짧은 도움말 문자열을 반환합니다.
- **initAlgorithm (필수)**
  이 함수에서 알고리즘의 입력물과 산출물을 정의합니다.

  주요 입력물 및 주요 산출물 파라미터의 이름으로 각각 `INPUT` 및 `OUTPUT` 을 추천합니다.

  파라미터가 다른 파라미터를 의존하는 경우, 해당 관계를 지정하기 위해 `parentParameterName` 을 사용합니다. (레이어의 필드/밴드 또는 레이어의 거리 단위일 수 있습니다.)
- **processAlgorithm (필수)**
  공간 처리를 실행하는 함수입니다.

`parameterAsSource` 및 `parameterAsDouble` 과 같은 특별한 목적을 위한 함수를 사용해서 파라미터를 받아옵니다.

공간 처리 알고리즘에서 다른 공간 처리 알고리즘을 실행하는 데 `processing.run` 을 사용할 수 있습니다. 첫 번째 파라미터는 알고리즘의 이름이고, 두 번째 파라미터는 알고리즘이 받는 파라미터의 목록입니다. 알고리즘 내부에서 또다른 알고리즘을 실행하는 경우 보통 `is_child_algorithm` 을 `True` 로 설정합니다. `context` 및 `feedback` 은 알고리즘에 실행 환경 및 사용자와의 (취소 요청을 받거나, 진행 과정을 보고하거나, 텍스트 피드백을 제공하는 등) 통신 채널에 대해 알려줍니다. "자식"알고리즘이 받는 파라미터로 (부모) 알고리즘의 파라미터를 사용하는 경우, 원본 파라미터 값을 사용해야 합니다. (예: `parameters['OUTPUT']`)

취소에 대해 논리적으로 가능한만큼 자주 피드백 객체를 확인하는 편이 좋습니다! 이렇게 하면 사용자가 원하지 않는 공간 처리 과정을 기다리는 대신 즉각 취소할 수 있으니까요.

알고리즘은 목록 (dictionary) 으로 정의한 모든 산출물 파라미터에 대해 값을 반환해야 합니다. 이 경우, 해당 값은 공간 처리한 버퍼, 래스터화 산출 레이어 그리고 피처 개수입니다. 목록 키는 원본 파라미터/산출물 이름과 일치해야만 합니다.

### 23.9.2 @alg 장식자

@alg 장식자 (decorator) 를 이용하면, 파이썬 코드를 작성한 다음 작성한 코드를 제대로 된 공간 처리 알고리즘으로 만드는 데 필요한 추가 정보를 제공하는 몇 줄을 더 추가해서 사용자 고유의 알고리즘을 생성할 수 있습니다. 이렇게 하면 알고리즘 생성 및 입력물/산출물의 지정을 간단히 완료할 수 있습니다.

장식자 접근법의 중요 제약 가운데 하나는, 이 방법으로 생성된 알고리즘이 언제나 사용자의 공간 처리 스크립트 제공자에 추가될 것이라는 점입니다. 이런 알고리즘을, 예를 들어 플러그인에서 사용하기 위해 사용자 지정 제공자에 추가할 수는 없습니다.

다음 코드는 @alg 장식자를 사용해서:

1. 벡터 데이터를 입력물로 받습니다.
2. 피처의 개수를 셉니다.
3. 버퍼 작업을 수행합니다.
4. 버퍼 작업의 결과물로부터 래스터 레이어를 생성합니다.
5. 버퍼 레이어, 래스터 레이어, 그리고 피처의 개수를 반환합니다.

```
from qgis import processing
from qgis.processing import alg
from qgis.core import QgsProject

@alg(name='bufferrasteralg', label='Buffer and export to raster (alg)',
     group='examplescripts', group_label='Example scripts')
# 'INPUT' is the recommended name for the main input parameter
@alg.input(type=alg.SOURCE, name='INPUT', label='Input vector layer')
# 'OUTPUT' is the recommended name for the main output parameter
@alg.input(type=alg.RASTER_LAYER_DEST, name='OUTPUT',
           label='Raster output')
@alg.input(type=alg.VECTOR_LAYER_DEST, name='BUFFER_OUTPUT',
           label='Buffer output')
@alg.input(type=alg.DISTANCE, name='BUFFERDIST', label='BUFFER DISTANCE',
           default=1.0)
@alg.input(type=alg.DISTANCE, name='CELLSIZE', label='RASTER CELL SIZE',
           default=10.0)
@alg.output(type=alg.NUMBER, name='NUMBEROFFEATURES',
            label='Number of features processed')
```

(다음 페이지에 계속)

(이전 페이지에서 계속)

```
20
21  def bufferrasteralg(instance, parameters, context, feedback, inputs):
22      """
23      Description of the algorithm.
24      (If there is no comment here, you will get an error)
25      """
26      input_featuresource = instance.parameterAsSource(parameters,
27                                                       'INPUT', context)
28      numfeatures = input_featuresource.featureCount()
29      bufferdist = instance.parameterAsDouble(parameters, 'BUFFERDIST',
30                                              context)
31      rastercellsize = instance.parameterAsDouble(parameters, 'CELLSIZE',
32                                                  context)
33      if feedback.isCanceled():
34          return {}
35      buffer_result = processing.run('native:buffer',
36                                     {'INPUT': parameters['INPUT'],
37                                      'OUTPUT': parameters['BUFFER_OUTPUT'],
38                                      'DISTANCE': bufferdist,
39                                      'SEGMENTS': 10,
40                                      'DISSOLVE': True,
41                                      'END_CAP_STYLE': 0,
42                                      'JOIN_STYLE': 0,
43                                      'MITER_LIMIT': 10
44                                      },
45                                     is_child_algorithm=True,
46                                     context=context,
47                                     feedback=feedback)
48      if feedback.isCanceled():
49          return {}
50      rasterized_result = processing.run('qgis:rasterize',
51                                         {'LAYER': buffer_result['OUTPUT'],
52                                          'EXTENT': buffer_result['OUTPUT'],
53                                          'MAP_UNITS_PER_PIXEL': rastercellsize,
54                                          'OUTPUT': parameters['OUTPUT']
55                                         },
56                                         is_child_algorithm=True, context=context,
57                                         feedback=feedback)
58      if feedback.isCanceled():
59          return {}
60      return {'OUTPUT': rasterized_result['OUTPUT'],
61              'BUFFER_OUTPUT': buffer_result['OUTPUT'],
62              'NUMBEROFFEATURES': numfeatures}
```

코드를 보면 알 수 있듯이 2 개의 알고리즘 (‘native:buffer’및‘qgis:rasterize’) 을 사용합니다. 후자 (‘qgis:rasterize’) 는 전자 (‘native:buffer’) 가 생성한 버퍼 레이어로부터 래스터 레이어를 생성합니다.

이전 장을 읽어보았다면 이 공간 처리 과정이 일어나는 코드 부분을 이해하기 어렵지는 않을 겁니다. 하지만 처음 몇 줄의 경우 추가 설명이 필요합니다. 이 줄들은 사용자의 코드를 툴박스 또는 모델 설계자 같은 어떤 GUI 구성 요소에서도 실행할 수 있는 알고리즘으로 변환시키는 데 필요한 정보를 제공하고 있습니다.

이 줄들은 모두 알고리즘 코딩 작업을 단순화시켜주는 `@alg` 장식자 함수를 호출하고 있습니다.

- @alg 장식자를 사용해서 툴박스에 있는 알고리즘의 이름 및 위치를 정의합니다.
- @alg.input 장식자를 사용해서 알고리즘의 입력물을 정의합니다.
- @alg.output 장식자를 사용해서 알고리즘의 산출물을 정의합니다.

기존 파라미터와 그에 대응하는 문자열을 알고 싶다면공간 처리 알고리즘의 입력물 및 산출물 유형 을 읽어보십시오.

### 23.9.3 알고리즘 산출물 건네주기

레이어 (래스터 또는 벡터) 를 표현하는 산출물을 선언하는 경우, 알고리즘 실행 종료시 알고리즘이 QGIS 에 산출물을 추가하려 할 것입니다.

- 래스터 레이어 산출물: QgsProcessingParameterRasterDestination / alg.RASTER_LAYER_DEST
- 벡터 레이어 산출물: QgsProcessingParameterVectorDestination / alg.VECTOR_LAYER_DEST

따라서 `processing.run()` 메소드가 사용자의 현재 프로젝트에 자신이 생성한 레이어를 추가하지 않았는데도 불구하고, 산출물 레이어 2 개 (버퍼 및 래스터 버퍼) 를 불러올 것입니다. 두 레이어가 사용자가 입력한 대상 위치에 (또는 사용자가 대상 위치를 지정하지 않은 경우 임시 위치에) 저장되기 때문입니다.

레이어가 알고리즘의 산출물로 생성되는 경우, 산출물로 선언되어야 합니다. 그렇지 않으면 모델 설계자에서 알고리즘을 제대로 사용할 수 없습니다. 선언한 내용과 알고리즘이 실제로 생성한 산출물이 일치하지 않을 것이기 때문입니다.

산출물 목록에 문자열, 숫자 등을 지정해서 ("NUMBEROFFEATURES"예시에서 보였듯이) 문자열, 숫자 등을 반환시킬 수 있습니다. 그러나 항상 사용자 알고리즘에서 나온 산출물로 명확하게 정의해야 합니다. 알고리즘이 유용한 값을 가능한 한 많이 산출하는 것을 권장합니다. 사용자 알고리즘이 모델의 일부로 사용되는 경우 다음 알고리즘에 매우 유용하게 사용할 수 있기 때문입니다.

### 23.9.4 사용자에게 정보 전달하기

사용자 알고리즘의 처리 시간이 오래 걸릴 경우, 사용자에게 진행 상황을 알려주는 편이 좋습니다. `feedback` (`QgsProcessingFeedback`) 을 사용하면 됩니다.

`setProgressText(text)` 및 `setProgress(percent)` 메소드 2 개를 사용하면 각각 진행 상황 텍스트 및 진행 상황 막대를 업데이트할 수 있습니다.

`pushCommandInfo(text)`, `pushDebugInfo(text)`, `pushInfo(text)` 그리고 `reportError(text)` 메소드를 사용하면 더 많은 정보를 제공할 수 있습니다.

사용자 스크립트에 문제가 있을 경우, 해당 문제를 처리하는 올바른 방법은 `QgsProcessingException` 예외를 발생시키는 것입니다. 예외 작성자에게 메시지를 인자로 넘겨줄 수 있습니다. 공간 처리 프레임워크는 알고리즘이 어디서 (툴박스, 모델 설계자, 파이썬 콘솔 등등) 실행되느냐에 따라 예외를 처리하고 사용자에게 알려줄 것입니다.

### 23.9.5 사용자 스크립트 문서 작성하기

`QgsProcessingAlgorithm` 클래스의 `helpString()` 및 `helpUrl()` 메소드를 과부하 (overload) 시키면 사용자 스크립트를 문서화할 수 있습니다.

### 23.9.6 플래그

`QgsProcessingAlgorithm` 클래스의 `flags()` 메소드를 무시하면 QGIS 에 사용자 알고리즘에 대해 더 많이 알려줄 수 있습니다. 예를 들어 QGIS 에 스크립트를 모델 설계자로부터 숨겨야 한다고, 취소될 수 있다고, 스레드 편에서 안전하지 않다고, 그 외의 많은 것을 알려줄 수 있습니다.

> **팁**
>
> 기본적으로 공간 처리 프레임워크는 공간 처리 작업을 실행하는 동안 QGIS 가 즉시 반응할 수 있도록 유지하기 위해 알고리즘을 개별 스레드로 실행합니다. 사용자 알고리즘이 정기적으로 충돌을 일으킨다면, 배경 스레드에서 작업하기엔 안전하지 않은 API 호출을 사용하고 있을 가능성이 높습니다. 공간 처리 프레임워크가 그 대신 사용자 알고리즘을 주요 스레드에서 실행시키도록 강제하려면, 사용자 알고리즘의 `flags()` 메소드에서 `QgsProcessingAlgorithm.FlagNoThreading` 플래그를 반환하게 해보십시오.

### 23.9.7 스크립트 알고리즘 작성을 위한 모범적인 예제

사용자가 자신의 스크립트 알고리즘을 생성하는 경우, 그 중에서도 특히 생성한 스크립트를 다른 QGIS 사용자와 공유하고 싶은 경우 고려해야 할 사항들을 정리해봤습니다. 이 간단한 규칙을 따른다면 툴박스, 모델 설계자, 배치 프로세스 인터페이스와 같은 서로 다른 공간 처리 구성 요소를 아우르는 일관성을 유지할 수 있을 겁니다.

- 산출 레이어를 불러오지 마십시오. 공간 처리 프레임워크가 사용자의 산출물을 처리하게 하고, 필요한 경우 불러오십시오.
- 항상 사용자 알고리즘이 생성한 산출물을 선언하십시오.
- 스크립트에서 메시지 상자를 표시한다는가 하는, 어떤 GUI 항목도 사용하지 마십시오. 사용자에게 정보를 제공하고 싶은 경우, 피드백 객체 (`QgsProcessingFeedback`) 메소드를 사용하거나 또는 `QgsProcessingException` 예외를 날리십시오.

QGIS 는 이미 수많은 공간 처리 알고리즘을 제공하고 있습니다. QGIS 저장소에서 코드를 찾아볼 수 있습니다.

## 23.10 공간 처리 알고리즘의 입력물 및 산출물 유형

다음은 공간 처리 프레임워크가 지원하는 입력물 및 산출물 유형을 대응하는 alg 장식자 상수와 함께 정리한 목록입니다. 알려져 있는 경우 모델 작성자에서의 이름도 포함됩니다.

### 23.10.1 입력물 유형

| 입력물 이름 | 클래스 | alg 상수 | 설명 |
|---|---|---|---|
| 주석 레이어 | `QgsProcessingParameterAnnotationLayer` | `alg.ANNOTATION_LAYEI` | 주석 레이어 |
| 면적 | `QgsProcessingParameterArea` | | 면적 측정값을 표현하는 숫자형 파라미터 |

다음 페이지에 계속

표 23.1 – 이전 페이지에서 계속

| 입력물 이름 | 클래스 | alg 상수 | 설명 |
|---|---|---|---|
| 인증 환경설정 (Authentication Configuration) | QgsProcessingParameterAuthConfig | alg.AUTH_CFG | 사용자가 사용할 수 있는 인증 환경 설정에서 선택하거나 또는 새 인증 환경 설정을 생성할 수 있게 해줍니다. |
| 래스터 밴드 | QgsProcessingParameterBand | alg.BAND | 래스터 레이어의 밴드 |
| 불 (Boolean) | QgsProcessingParameterBoolean | alg.BOOL | 불 (boolean) 값 |
| 연결 이름 | QgsProcessingParameterProviderConnection | | 특정 데이터 제공자에 사용할 수 있는 등록된 연결들에서 선택한 집합 |
| 색상 | QgsProcessingParameterColor | alg.COLOR | 색상 |
| 좌표 연산 | QgsProcessingParameterCoordinateOperation | alg.COORDINATE_OPERA | 좌표 연산 (좌표계 변환 용) |
| CRS | QgsProcessingParameterCrs | alg.CRS | 좌표계 |
| 데이터베이스 스키마 | QgsProcessingParameterDatabaseSchema | alg.DATABASE_SCHEMA | 데이터베이스 스키마 |
| 데이터베이스 테이블 | QgsProcessingParameterDatabaseTable | alg.DATABASE_TABLE | 데이터베이스 테이블 |
| 날짜 | ProcessingDateTimeParameterDataType.Date | alg.DATE | 단순한 날짜 |
| 날짜 & 시간 | ProcessingDateTimeParameterDataType.DateTime | alg.DATETIME | 날짜와 시간 |
| 거리 | QgsProcessingParameterDistance | alg.DISTANCE | 거리 값을 위한 실수형 (double) 숫자 파라미터 |
| DXF 레이어 | QgsProcessingParameterDxfLayers | | DXF 파일로 내보낼 벡터 레이어 목록 |
| 열거 목록 (enum) | QgsProcessingParameterEnum | alg.ENUM | 사전 정의된 값들의 집합에서 선택할 수 있는 집합 (enumeration) |
| 표현식 | QgsProcessingParameterExpression | alg.EXPRESSION | 표현식 |
| 범위 (extent) | QgsProcessingParameterExtent | alg.EXTENT | xmin, xmax, ymin, ymax 로 정의된 공간 범위 |
| | QgsProcessingParameterField | alg.FIELD | 벡터 레이어의 속성 테이블에 있는 필드 |
| 필드 집계 | QgsProcessingParameterAggregate | | 원하는 산출 필드의 정의, 유형, 그리고 필드를 채우는 데 쓰이는 집계로 이루어진, “집계 (aggregate)”환경설정을 위한 파라미터 |

다음 페이지에 계속

표 23.1 – 이전 페이지에서 계속

| 입력물 이름 | 클래스 | alg 상수 | 설명 |
|---|---|---|---|
| 필드 매핑 작업자 | QgsProcessingParameterFieldMapping | | 원하는 산출 필드의 정의, 유형, 그리고 필드를 채우는 데 쓰이는 표현식으로 이루어진, "필드 매핑(field mapping)"환경설정을 위한 파라미터 |
| 파일/폴더 | QgsProcessingParameterFile | alg.FILE | 기존 파일의 파일명 |
| | QgsProcessingParameterFileDestination | alg.FILE_DEST | 새로 생성된 산출물 파일의 파일명 |
| | QgsProcessingParameterFolderDestination | alg.FOLDER_DEST | 폴더 (대상 폴더) |
| 도형 | QgsProcessingParameterGeometry | alg.GEOMETRY | 도형 |
| | ProcessingNumberParameterType.Integer | alg.INT | 정수 |
| 인쇄 조판기 | QgsProcessingParameterLayout | alg.LAYOUT | 조판 |
| 인쇄 조판 항목 | QgsProcessingParameterLayoutItem | alg.LAYOUT_ITEM | 조판 항목 |
| 맵 레이어 | QgsProcessingParameterMapLayer | alg.MAPLAYER | 맵 레이어 |
| 맵 테마 | QgsProcessingParameterMapTheme | alg.MAP_THEME | 프로젝트 맵 테마 |
| 매트릭스 | QgsProcessingParameterMatrix | alg.MATRIX | 매트릭스 |
| 메시 데이터셋 그룹 | QgsProcessingParameterMeshDatasetGroups | | 메시 레이어의 데이터셋 그룹 |
| 메시 데이터셋 시간 | QgsProcessingParameterMeshDatasetTime | | 메시 레이어의 시간 설정에서 나온 데이터셋 인덱스 |
| 메시 레이어 | QgsProcessingParameterMeshLayer | alg.MESH_LAYER | 메시 레이어 |
| 다중 입력 | QgsProcessingParameterMultipleLayers | alg.MULTILAYER | 레이어 집합 |
| 숫자 | ProcessingNumberParameterType.Double | alg.NUMBER | 숫자 값 |
| 포인트 | QgsProcessingParameterPoint | alg.POINT | 포인트 |
| 점구름 속성 | QgsProcessingParameterPointCloudAttribute | | 점구름 레이어의 속성 |
| | QgsProcessingParameterPointCloudDestination | alg.POINTCLOUD_LAYEI | 알고리즘이 생성한 점구름 레이어의 대상 경로를 지정하는 점구름 레이어 대상 파라미터 |

다음 페이지에 계속

표 23.1 – 이전 페이지에서 계속

| 입력물 이름 | 클래스 | alg 상수 | 설명 |
|---|---|---|---|
| 점구름 레이어 | QgsProcessingParameterPointCloudLayer | alg.POINTCLOUD_LAYEI | 점구름 (point cloud) 레이어 |
| | QgsProcessingParameterProviderConnection | alg.PROVIDER_CONNECT | 데이터베이스 제공자에 대해 사용할 수 있는 연결 |
| 범위 | QgsProcessingParameterRange | alg.RANGE | 숫자의 범위 |
| 래스터 레이어 | QgsProcessingParameterRasterLayer | alg.RASTER_LAYER | 래스터 레이어 |
| | QgsProcessingParameterRasterDestination | alg.RASTER_LAYER_DES | 알고리즘이 생성한 래스터 레이어의 대상 경로를 지정하는 래스터 레이어 대상 파라미터 |
| 축척 | QgsProcessingParameterScale | alg.SCALE | 맵 축척 |
| | QgsProcessingParameterFeatureSink | alg.SINK | 피처 싱크 (sink) |
| 문자열 | QgsProcessingParameterString | alg.STRING | 텍스트 문자열 |
| TIN 생성 레이어 | QgsProcessingParameterTinInputLayers | | 꼭짓점들 그리고/또는 단절선 (breakline) 들로 TIN 을 생성할 여러 레이어들의 집합 |
| 시간 | ProcessingDateTimeParameterDataType.Time | alg.TIME | 단순한 시간 |
| 벡터 피처 | QgsProcessingParameterFeatureSource | alg.SOURCE | 피처 소스 |
| 벡터 레이어 | QgsProcessingParameterVectorLayer | alg.VECTOR_LAYER | 벡터 레이어 |
| | QgsProcessingParameterVectorDestination | alg.VECTOR_LAYER_DES | 알고리즘이 생성한 벡터 레이어의 대상 경로를 지정하는 벡터 레이어 대상 파라미터 |
| 벡터 타일 작성자 레이어 | QgsProcessingParameterVectorTileDestination | | 알고리즘이 생성한 벡터 타일 레이어의 대상 경로를 지정하는 벡터 타일 레이어 대상 파라미터 |
| 부피 | QgsProcessingParameterVolume | | 부피 측정값을 표현하는 숫자형 파라미터 |

### 23.10.2 산출물 유형

| 클래스 | alg 상수 | 설명 |
|---|---|---|
| QgsProcessingOutputBoolean | alg.BOOL | 불 (boolean) 값 |
| QgsProcessingOutputNumber | alg.DISTANCE | 거리 값을 위한 실수형 (double) 숫자 파라미터 |
| QgsProcessingOutputFile | alg.FILE | 기존 파일의 파일명 |
| QgsProcessingOutputFolder | alg.FOLDER | 폴더 |
| QgsProcessingOutputHtml | alg.HTML | HTML |

다음 페이지에 계속

표 23.2 – 이전 페이지에서 계속

| 클래스 | alg 상수 | 설명 |
|---|---|---|
| QgsProcessingOutputNumber | alg.INT | 정수 |
| QgsProcessingOutputLayerDefinition | alg.LAYERDEF | 레이어 정의 |
| QgsProcessingOutputMapLayer | alg.MAPLAYER | 맵 레이어 |
| QgsProcessingOutputMultipleLayers | alg.MULTILAYER | 레이어 집합 |
| QgsProcessingOutputNumber | alg.NUMBER | 숫자 값 |
| QgsProcessingOutputPointCloudLayer | alg.POINTCLOUD_LAYER | 점구름 (point cloud) 레이어 |
| QgsProcessingOutputRasterLayer | alg.RASTER_LAYER | 래스터 레이어 |
| QgsProcessingOutputString | alg.STRING | 텍스트 문자열 |
| QgsProcessingOutputVectorLayer | alg.VECTOR_LAYER | 벡터 레이어 |
| QgsProcessingOutputVectorTileLayer | | 벡터 타일 레이어 |

## 23.11 외부 응용 프로그램 환경 설정

The Processing framework can be extended using additional applications. Algorithms that rely on external applications are managed by their own algorithm providers. Additional providers can be found as separate plugins, and installed using the QGIS Plugin Manager.

여기에서는 이런 추가적인 응용 프로그램을 포함시키려면 공간 처리 프레임워크를 어떻게 환경 설정해야 하는지, 그리고 추가 응용 프로그램을 기반으로 한 알고리즘의 몇몇 특정 기능에 관해 설명할 것입니다. 시스템을 정확하게 환경 설정하고 나면, 다른 모든 알고리즘과 마찬가지로 툴박스 또는 모델 설계자 같은 어떤 구성 요소에서도 외부 알고리즘을 실행할 수 있을 것입니다.

기본적으로, QGIS 가 내장하지 않은 외부 응용 프로그램에 의존하는 모든 알고리즘은 활성화된 상태가 아닙니다. 공간 처리 설정 대화창에서 이런 알고리즘을 활성화시킬 수 있지만, 사용자 시스템에 해당 응용 프로그램이 이미 설치돼 있어야 합니다.

### 23.11.1 윈도우 사용자를 위한 소고

If you are not an advanced user and you are running QGIS on Windows, you might not be interested in reading the rest of this chapter. Make sure you install QGIS in your system using the standalone installer. That will automatically install GRASS in your system and configure it so it can be run from QGIS. All the algorithms from this provider will be ready to be run without needing any further configuration. If installing with the OSGeo4W application, make sure that you also select GRASS for installation.

### 23.11.2 파일 포맷에 대한 소고

외부 소프트웨어 사용시, QGIS 가 파일을 열 수 있다는 것이 다른 소프트웨어도 해당 파일을 열고 처리할 수 있다는 의미는 아닙니다. 대부분의 경우 다른 소프트웨어도 QGIS 가 연 파일을 읽을 수 있지만, 어떤 경우 읽을 수 없을 수도 있습니다. 데이터베이스 또는 흔하지 않은 파일 포맷을 사용하는 경우, 그것이 래스터 레이어든 벡터 레이어든, 문제가 발생할 수도 있습니다. 이런 경우 QGIS 와 외부 소프트웨어 둘 다 지원하고 있는 것이 확실한, 잘 알려진 파일 포맷으로 바꿔서 해본 다음, (로그 패널에서) 콘솔 산출물을 확인해서 뭐가 잘못된 건지 자세히 알아보십시오.

예를 들면 GRASS 래스터 레이어를 입력받는 외부 알고리즘을 호출했는데 문제가 발생해서 작업을 완료할 수 없는 경우가 있을 수도 있습니다. 이런 이유로, 이런 레이어는 알고리즘에 사용할 수 있다고 나타나지 않을 것입니다.

하지만 벡터 레이어 사용 시엔 이런 문제가 전혀 없을 것입니다. 외부 응용 프로그램에 레이어를 넘겨주기 전에 QGIS 가 원본 파일 포맷을 외부 응용 프로그램이 지원하는 파일 포맷으로 자동 변환하기 때문입니다. 이

습성 때문에 공간 처리 작업 시간이 늘어나게 되는데, 대용량 레이어의 경우 특히 시간이 오래 걸릴 것입니다. 그러므로 셰이프파일 포맷 데이터셋의 레이어를 처리하는 데 드는 시간보다 데이터베이스 연결로부터 비슷한 용량의 레이어를 처리하는 데 드는 시간이 더 오래 걸려도 너무 놀라지 마십시오.

외부 응용 프로그램을 사용하지 않는 제공자는, QGIS 를 통해서 레이어를 열어 분석하기 때문에 QGIS 가 열 수 있는 모든 레이어를 처리할 수 있습니다.

QGIS 가 생성한 모든 래스터 및 벡터 레이어 산출물 포맷을 입력 레이어로 사용할 수 있습니다. 일부 제공자가 특정 포맷을 지원하지 않는 경우에도, 모든 제공자가 QGIS 가 나중에 자동 변환할 수 있는 공통 포맷으로 내보낼 수 있기 때문입니다. 입력 레이어에 이런 변환 작업이 필요한 경우 공간 처리 작업 시간이 늘어날 수도 있습니다.

### 23.11.3 벡터 레이어 선택에 대한 소고

외부 응용 프로그램이 QGIS 안에 존재하는 벡터 레이어 선택 집합을 인지하게 만들 수도 있습니다. 하지만 원본 벡터 레이어들이 외부 응용 프로그램이 지원하지 않는 포맷인 경우, 이렇게 하려면 모든 입력 벡터 레이어를 재작성해야만 합니다. 어떤 선택 집합도 없을 때만, 또는 공간 처리 일반 환경 설정에서 *Use only selected features* 옵션이 비활성화된 경우에만 외부 응용 프로그램으로 레이어를 직접 넘겨줄 수 있습니다.

그 외의 경우, 선택한 객체만 내보내야 하는데 이렇게 되면 실행 시간이 늘어나게 됩니다.

### 23.11.4 제 3 자 제공자 사용하기

#### R 스크립트와 라이브러리

공간 처리 프레임워크에서 R 을 활성화하려면 **공간 처리 R 제공자** 플러그인을 설치하고 QGIS 에 맞게 R 을 환경 설정해야 합니다.

QGIS 옵션 대화창의 (*Settings* → *Options*) *Processing* 탭에 있는 *Provider* → *R* 에서 환경 설정할 수 있습니다.

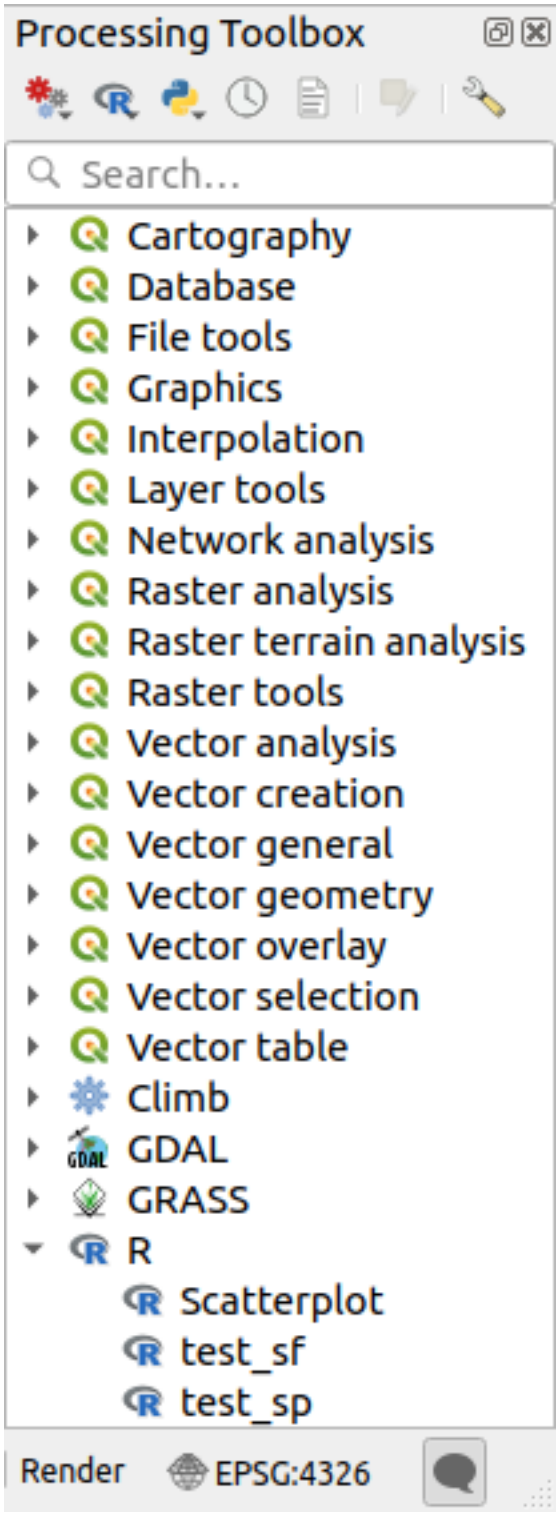
사용자의 운영 체제에 따라, *R folder* 를 통해 사용자의 R 바이너리가 있는 위치를 지정해야 할 수도 있습니다.

> **참고**
>
> **윈도우** 상에서는 R 실행 파일이 보통 `C:\Program Files\R\` 아래 있는 폴더 (`R-<version>`) 에 있습니다. 바이너리 파일이 **아니라** 폴더를 지정하세요!
>
> **리눅스** 상에서는 R 폴더가 PATH 환경 변수에 포함돼 있는지 확인만 해주면 됩니다. 터미널 창에서 `R` 명령어로 R 을 실행할 수 있다면, 모든 준비가 끝난 겁니다.

**공간 처리 R 제공자** 플러그인을 설치했다면, *Processing Toolbox* 에서 다음과 같은 몇몇 예시 스크립트를 볼 수 있을 것입니다:

- *Scatterplot*: 지정한 벡터 레이어의 숫자 필드 2 개로부터 산점도 (scatter plot) 를 생성하는 R 함수를 실행합니다.
- *test_sf*: `sf` 패키지에 의존하는 몇몇 작업을 수행합니다. R 패키지 `sf` 가 설치되었는지 확인하는 데 사용할 수 있습니다. 해당 패키지가 설치되지 않았다면, R 이 공간 처리 옵션 대화창의 *Provider* → *R* 에서 지정한 *Package repository* 를 통해 (이 패키지가 의존하는 모든 패키지와 함께) 해당 패키지를 설치하려 할 것입니다. 기본 저장소는 https://cran.r-project.org/ 입니다. 설치에 시간이 걸릴 수도 있습니다.
- *test_sp*: R 패키지 `sp` 가 설치되었는지 확인하는 데 사용할 수 있습니다. 해당 패키지가 설치되지 않았다면, R 이 설치하려 할 것입니다.

QGIS 에 맞게 R 을 정확히 환경 설정했다면, 이 스크립트들을 실행할 수 있을 것입니다.


Processing Toolbox
Search...
Cartography
Database
File tools
Graphics
Interpolation
Layer tools
Network analysis
Raster analysis
Raster terrain analysis
Raster tools
Vector analysis
Vector creation
Vector general
Vector geometry
Vector overlay
Vector selection
Vector table
Climb
GDAL
GRASS
R
Scatterplot
test_sf
test_sp
Render
EPSG:4326

### QGIS 저장소에서 R 스크립트 추가하기

R integration in QGIS is different from that of some other third party providers like SAGA in that there is not a predefined set of algorithms you can run (except for some example script that come with the *Processing R Provider* plugin).

QGIS 저장소에서 예시 R 스크립트 집합을 사용할 수 있습니다. *QGIS* 리소스 공유 플러그인을 사용해 이 예시 스크립트를 불러와서 활성화하려면 다음 단계들을 수행하십시오.

1. *QGIS* 리소스 공유 플러그인을 추가하십시오. (플러그인 관리자의 *Settings* 에서 *Show also experimental plugins* 옵션을 활성화해야 할 수도 있습니다.)
2. QGIS 리소스 공유 플러그인을 여십시오. (*Plugins → Resource Sharing → Resource Sharing*)
3. *Settings* 탭을 선택하십시오.
4. *Reload repositories* 를 클릭하십시오.
5. *All* 탭을 선택하십시오.
6. 목록에서 *QGIS R script collection* 을 선택한 다음 *Install* 버튼을 클릭하십시오.
7. 이제 *Installed* 탭의 목록에 R 스크립트가 추가되었을 것입니다.
8. 플러그인을 종료하십시오.
9. *Processing Toolbox* 를 여십시오. 모든 것이 제대로 되었다면, 예시 스크립트가 R 아래 여러 그룹으로 나타나 있을 것입니다. (다음 스크린샷에서는 몇몇 그룹만 펼쳐져 있습니다.)

   가장 위에 있는 스크립트들은 공간 처리 *R* 제공자 플러그인의 예시 스크립트들입니다.
10. 만약 어떤 이유로든 *Processing Toolbox* 에 스크립트가 추가되지 않았다면, 다음 작업을 해볼 수 있습니다:
    1. 공간 처리 설정 대화창을 엽니다. (*Settings → Options → Processing* 탭)
    2. *Providers → R → R scripts folder* 를 선택합니다.
       - 우분투의 경우, 다음을 가리키는 경로를 설정하십시오 (또는, 더 나은 방법으로, PATH 에 포함시키십시오):

         /home/<user>/.local/share/QGIS/QGIS3/profiles/default/resource_sharing/repositories/github.com/qgis/QGIS-Resources/collections/rscripts
       - 윈도우의 경우, 다음을 가리키는 경로를 설정하십시오 (또는, 더 나은 방법으로, PATH 에 포함시키십시오):

         C:\Users\<user>\AppData\Roaming\QGIS\QGIS3\profiles\default\resource_sharing\repositories\github.com\qgis\QG Resources\collections\rscripts

       경로를 편집하려면 더블 클릭하십시오. 이제 경로를 그냥 붙여넣기/입력하거나, 또는 …버튼을 클릭하면 열리는 대화창에서 디렉터리를 찾은 다음 *Add* 버튼을 클릭하면 됩니다. 여러 디렉터리를 지정할 수도 있습니다. 각 디렉터리는 쌍반점 (`;`) 으로 구분됩니다.

QGIS 2 버전 온라인 저장소에서 모든 R 스크립트를 가져오고 싶다면, *QGIS R script collection* 대신 *QGIS R script collection (from QGIS 2)* 를 선택하면 됩니다. 벡터 데이터 입력물 또는 산출물에 의존하는 스크립트는 아마도 작동하지 않을 것입니다.

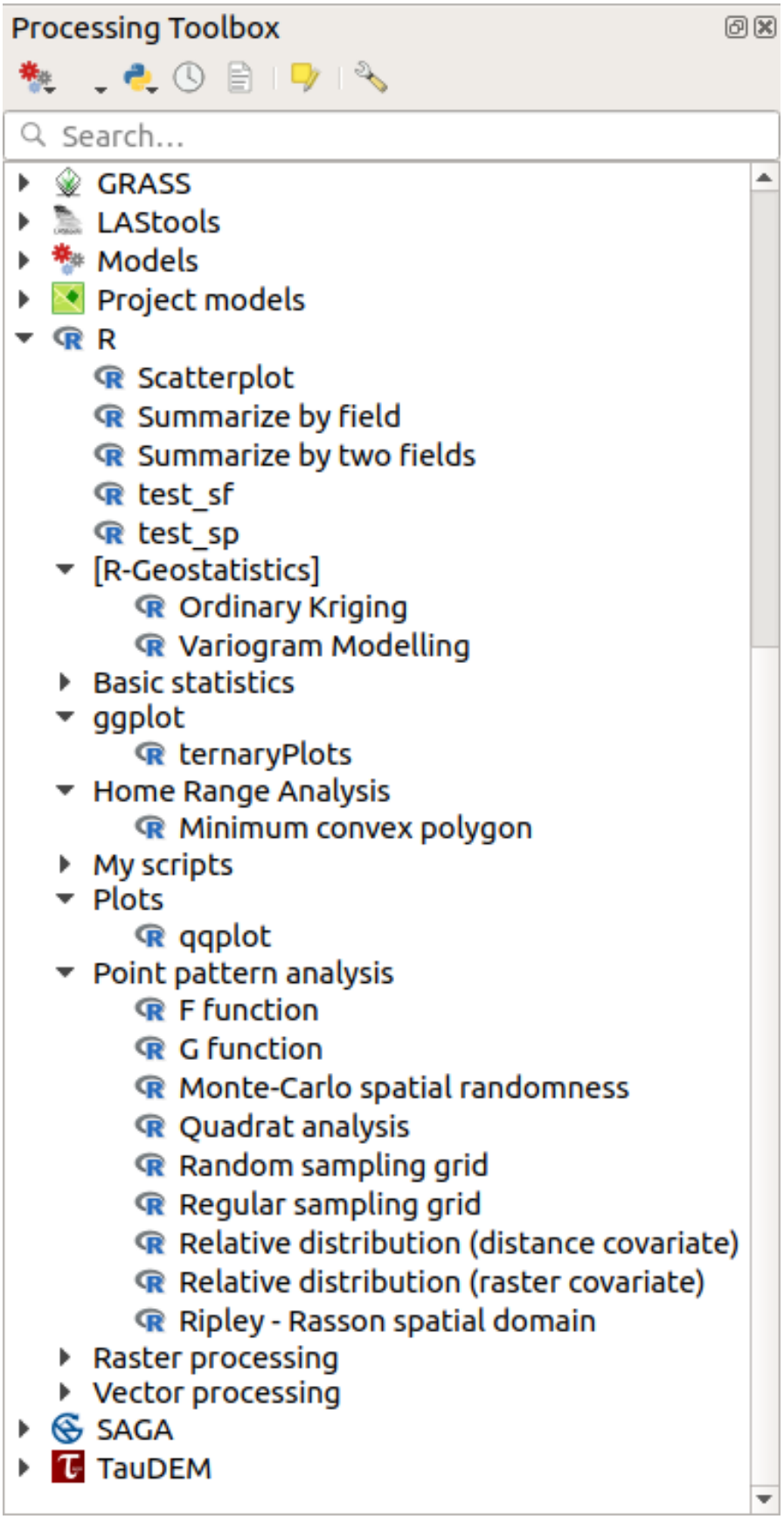
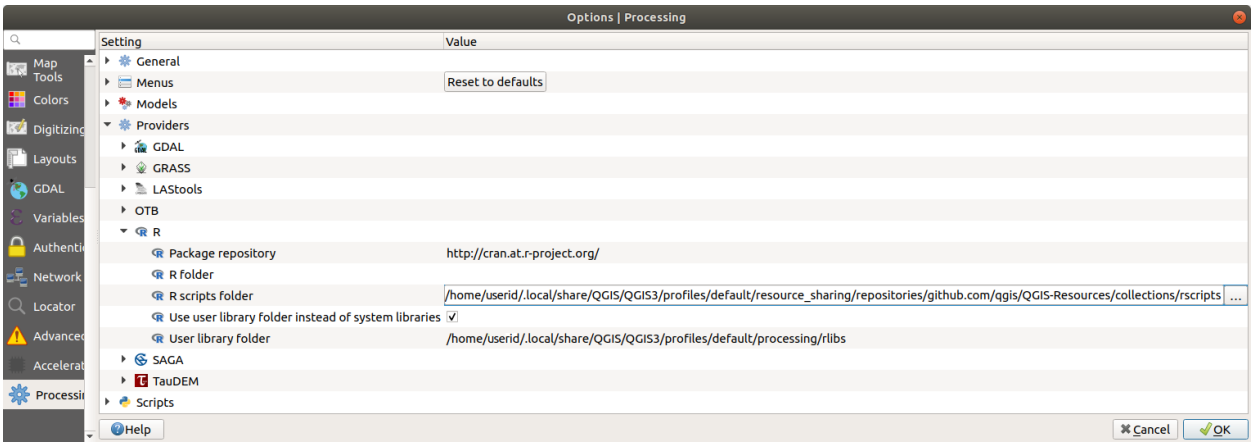
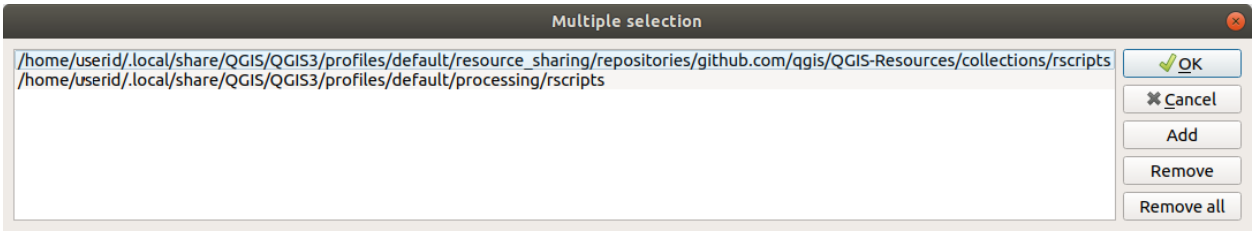



그림 23.35: 몇몇 R 스크립트가 표시된 *Processing Toolbox*




## R 스크립트 생성하기

R 을 실행하는 것과 동일한 방법으로 스크립트를 작성하고 R 명령어를 호출할 수 있습니다. 이 절에서는 QGIS 에서 R 명령어를 사용하기 위한 문법과, R 명령어에 QGIS 객체 (레이어, 테이블) 를 사용하는 방법을 설명합니다.

R 함수를 (또는 사용자가 개발해서 QGIS 에서 사용하고자 하는 좀 더 복잡한 R 스크립트를) 호출하는 알고리즘을 추가하려면, R 명령어를 수행하는 스크립트 파일을 생성해야 합니다.

R 스크립트 파일의 확장자는 `.rsx` 이며, R 문법 및 R 스크립트 작성에 대한 기초 지식만 있다면 R 스크립트를 꽤 간단하게 생성할 수 있습니다. R 스크립트 파일은 R 스크립트 폴더에 저장해야 합니다. 공간 처리 설정 대화창에 있는 *R* 설정 그룹에서 해당 폴더를 (*R scripts folder*) 지정할 수 있습니다.

입력 폴리곤 레이어에 있는 폴리곤의 경계 내부에 랜덤한 그리드를 생성하는 R 메소드 `spsample` 을 호출하는 아주 간단한 스크립트 파일을 살펴보겠습니다. 이 메소드는 `maptools` 패키지에 속해 있습니다. 사용자가 QGIS 에 통합하고 싶을 거의 모든 알고리즘이 공간 데이터를 사용하거나 생성하기 때문에, 사용자는 `maptools` 그리고 `sp`/`sf` 같은 공간 패키지에 대해 잘 알고 있어야 합니다.

```
##Random points within layer extent=name
##Point pattern analysis=group
##Vector_layer=vector
##Number_of_points=number 10
##Output=output vector
library(sp)
spatpoly = as(Vector_layer, "Spatial")
pts=spsample(spatpoly,Number_of_points,type="random")
spdf=SpatialPointsDataFrame(pts, as.data.frame(pts))
Output=st_as_sf(spdf)
```

2 중 파이썬 주석 기호 (`##`) 로 시작하는 첫 몇 행이 스크립트의 표시 이름 및 그룹을 정의하고, QGIS 에 입력물 및 산출물을 알려줍니다.

> **참고**
>
> 사용자 고유의 R 스크립트를 어떻게 작성하는지에 대한 자세한 정보를 알고 싶다면 QGIS 교육 교재의 R 소개에서*R 문법* 을 살펴보시기 바랍니다.

입력물 파라미터를 선언하는 경우, QGIS 는 해당 정보를 사용자에게 해당 파라미터의 값을 요청하기 위한 사용자 인터페이스를 생성하고, 나중에 R 함수에 입력할 수 있는 대응하는 R 변수를 생성한다는 두 가지 목적을 위해 사용합니다.

앞의 예시에서, `Vector_layer` 라는 `vector` 입력물 유형을 선언하고 있습니다. 이 알고리즘을 실행하면, QGIS 가 사용자가 선택한 레이어를 열어서 역시 `Vector_layer` 라는 이름을 가진 변수에 저장할 것입니다. 즉, 파라미터명이 R 에서 해당 파라미터의 값에 접근하기 위해 사용하는 변수명이기도 합니다. (따라서 예비된 R 단어를 파라미터명으로 사용해서는 안 됩니다.)

벡터 및 래스터 레이어 같은 공간 파라미터는 `st_read()` (또는 `readOGR()`) 및 `brick()` (또는 `readGDAL`) 명령어를 사용해서 (사용자 설명 파일에 이런 명령어를 추가하지 않아도 됩니다 — QGIS 가 처리할 테니까요) 읽어들여 `sf` (또는 `Spatial*DataFrame`) 객체로 저장합니다.

테이블 필드 유형은 선택한 필드의 이름을 담고 있는 문자열로 저장됩니다.

`##load_vector_using_rgdal` 을 지정하면 `st_read()` 대신 `readOGR()` 명령어를 사용해서 벡터 파일을 읽어들일 수 있습니다. 이렇게 하면 `sf` 객체 대신 `Spatial*DataFrame` 객체를 생성할 것입니다.

`##load_raster_using_rgdal` 을 지정하면 `brick()` 대신 `readGDAL()` 명령어를 사용해서 래스터 파일을 읽어들일 수 있습니다.

사용자가 고급 사용자이며 QGIS 가 레이어용 객체를 생성하기를 원하지 않는다면, 객체 대신 파일명을 담은 문자열을 선호한다고 알려주는 `##pass_filenames` 태그를 사용하면 됩니다. 이 태그를 사용할 경우, 파일이 담고 있는 데이터에 대해 어느 작업도 하기 전에 사용자가 해당 파일을 열어야 합니다.

이런 정보를 바탕으로, 이제 R 스크립트의 (파이썬 주석 기호로 시작하지 않는) 첫 줄들을 이해할 수 있습니다.

```
library(sp)
spatpoly = as(Vector_layer, "Spatial")
pts=spsample(polyg,numpoints,type="random")
```

`sp` 라이브러리가 `spsample` 함수를 제공하기 때문에, 가장 먼저 해당 라이브러리를 불러와야 합니다. `Vector_layer` 변수가 `sf` 객체를 담고 있습니다. `sp` 라이브러리에서 함수 (`spsample`) 를 사용할 것이기 때문에, `as` 함수를 이용해서 `sf` 객체를 `SpatialPolygonsDataFrame` 객체로 변환해야만 합니다.

그 다음 이 객체 및 (생성할 포인트의 개수를 지정하는) `numpoints` 입력 파라미터와 함께 `spsample` 함수를 호출합니다.

이미 `Output` 이라는 벡터 산출물을 선언했기 때문에, `sf` 객체를 담고 있는 `Output` 이라는 변수를 생성해야 합니다.

이 변수를 생성하기 위해서는 두 단계를 거쳐야 합니다. 먼저 *SpatialPointsDataFrame* 함수를 통해 함수의 결과물에서 `SpatialPolygonsDataFrame` 객체를 생성합니다. 그 다음, (`sf` 라이브러리의) `st_as_sf` 함수를 사용해서 해당 객체를 `sf` 객체로 변환합니다.

중간 단계 변수는 사용자 마음대로 명명해도 됩니다. 단 최종 산출물을 저장하는 변수가 정의된 이름과 (이 경우 `Output`) 동일한 이름을 가지고 있는지, 그리고 해당 변수가 적합한 값을 (벡터 레이어 산출물에 대한 `sf` 객체를) 담고 있는지 확인하십시오.

이 경우, `spsample` 메소드의 산출물을 `SpatialPointsDataFrame` 객체를 통해 명백하게 `sf` 객체로 변환해야 합니다. 산출물 자체가 QGIS 에 그대로 반환하기에 적합하지 않은 `ppp` 클래스의 객체이기 때문입니다.

사용자 알고리즘이 래스터 레이어를 생성하는 경우, 사용자가 `##dontuserasterpackage` 태그를 사용했는지 여부에 따라 래스터를 저장하는 방식이 달라질 것입니다. 해당 태그를 사용했다면, `writeGDAL()` 메소드를

사용해서 레이어를 저장합니다. 사용하지 않았다면, `raster` 패키지의 `writeRaster()` 메소드를 사용할 것입니다.

`##pass_filenames` 옵션을 사용한 경우, `raster` 패키지를 (`writeRaster()` 메소드를 통해) 사용해서 산출물을 생성합니다.

사용자 알고리즘이 어떤 레이어도 생성하지 않지만, 그 대신 콘솔에 텍스트 산출물을 생성하는 경우 알고리즘 실행이 완료된 후 콘솔을 열고 싶다는 표현을 해야 합니다. 그렇게 하려면 사용자가 출력하고자 하는 산출물을 생성하는 명령어 줄 앞에 > ('초과') 기호를 삽입하기만 하면 됩니다. 다른 명령어 줄의 산출물은 표시되지 않을 것입니다. 예를 들면, 다음은 벡터 레이어의 속성에서 선택한 필드 (열) 에 대해 정규성 검정 (normality test) 을 수행하는 알고리즘의 설명 파일입니다:

```
##layer=vector
##field=field layer
##nortest=group
library(nortest)
>lillie.test(layer[[field]])
```

마지막 행의 산출물을 출력하지만, 그 위 행의 산출물은 출력되지 않습니다. (그리고 각 행의 산출물 가운데 어느 쪽도 QGIS 에 자동적으로 추가되지 않습니다.)

사용자 알고리즘이 (`plot()` 메소드를 사용해서) 어느 종류의 그래픽이라도 생성하는 경우, 다음 줄을 추가하십시오 (`output_plots_to_html` 이 예전 `showplots` 입니다):

```
##output_plots_to_html
```

이 태그를 사용하면, QGIS 가 모든 R 그래픽 산출물을 임시 파일로 돌려서, R 실행이 종료된 다음 불러올 것입니다.

콘솔 및 그래픽 산출물 둘 다 공간 처리 산출물 관리자를 통해 사용할 수 있습니다.

더 자세한 정보를 원한다면, 공식 QGIS 저장소에 있는 R 스크립트 파일을 확인해보십시오. (*QGIS 저장소에서 R 스크립트 추가하기* 에서 설명한대로 *QGIS* 리소스 공유 플러그인을 사용해서 다운로드하고 설치하십시오.) 대부분 단순한 편이라 사용자 고유의 스크립트를 어떻게 생성하는지 이해하는 데 큰 도움을 줄 것입니다.

> **참고**
>
> 기본적으로 `sf`, `rgdal` 및 `raster` 라이브러리를 불러오기 때문에, 사용자가 대응하는 `library()` 명령어를 추가하지 않아도 됩니다. 하지만, 필요할 수도 있는 기타 추가 라이브러리는, 예를 들어 (`ggplot2` 라이브러리를 불러오기 위한) `library(ggplot2)` 와 같이 직접 입력해서 명시적으로 불러와야 합니다. 해당 패키지가 사용자 시스템 상에 설치돼 있지 않다면, 공간 처리 프레임워크가 다운로드해서 설치하려 할 것입니다. 이렇게 하면 R 단독형 소프트웨어도 해당 패키지를 사용할 수 있습니다. 패키지를 다운로드해야 할 경우, 스크립트를 처음 실행할 때 시간이 오래 걸릴 수도 있다는 사실을 **명심하십시오**.

#### sf_test 실행시 설치되는 R 라이브러리

R 스크립트 `sp_test` 는 `sp` 및 `raster` R 패키지를 불러오려 합니다.

R 스크립트 `sf_test` 는 `sf` 및 `raster` R 패키지를 불러오려 합니다. 이 두 패키지가 설치되지 않은 경우, R 이 이들을 (그리고 이들이 의존하는 모든 라이브러리를) 불러와서 설치하려 할 수도 있습니다.

우분투 상에 *R* 3.4.4 버전을 처음 설치한 다음 공간 처리 툴박스에서 공간 처리 *R* 제공자 플러그인 2.0 버전으로 `sf_test` 를 실행했다면, 다음 R 라이브러리들이 `~/.local/share/QGIS/QGIS3/profiles/default/processing/rscripts` 폴더에 설치되었을 겁니다.

```
abind, askpass, assertthat, backports, base64enc, BH, bit, bit64, blob, brew, callr,␣
↪classInt, cli, colorspace, covr, crayon, crosstalk, curl, DBI, deldir,
desc, dichromat, digest, dplyr, e1071, ellipsis, evaluate, fansi, farver, fastmap,␣
↪gdtools, ggplot2, glue, goftest, gridExtra, gtable, highr, hms,
htmltools, htmlwidgets, httpuv, httr, jsonlite, knitr, labeling, later, lazyeval,␣
↪leafem, leaflet, leaflet.providers, leafpop, leafsync, lifecycle, lwgeom,
magrittr, maps, mapview, markdown, memoise, microbenchmark, mime, munsell, odbc,␣
↪openssl, pillar, pkgbuild, pkgconfig, pkgload, plogr, plyr, png, polyclip,
praise, prettyunits, processx, promises, ps, purrr, R6, raster, RColorBrewer, Rcpp,␣
↪reshape2, rex, rgeos, rlang, rmarkdown, RPostgres, RPostgreSQL,
rprojroot, RSQLite, rstudioapi, satellite, scales, sf, shiny, sourcetools, sp,␣
↪spatstat, spatstat.data, spatstat.utils, stars, stringi, stringr, svglite,
sys, systemfonts, tensor, testthat, tibble, tidyselect, tinytex, units, utf8, uuid,␣
↪vctrs, viridis, viridisLite, webshot, withr, xfun, XML, xtable
```

### GRASS

Configuring GRASS is very easy. First, the path to the GRASS folder has to be defined, but only if you are running Windows.

기본적으로, 공간 처리 프레임워크는 QGIS 와 함께 빌드된 GRASS 배포판을 사용하기 위해 GRASS 연결기 (connector) 를 환경 설정하려 합니다. 대부분의 시스템에서 문제없이 설정되지만, 문제가 발생한다면 사용자가 GRASS 연결기를 직접 환경 설정해야 할 수도 있습니다. 또 서로 다른 버전의 GRASS 들을 사용하고자 하는 경우, 해당 설정을 변경해서 다른 버전이 설치돼 있는 폴더를 가리키게 할 수도 있습니다. 알고리즘이 정확하게 작동하려면 GRASS 7 버전이 필요합니다.

리눅스 환경인 경우, GRASS 가 정확하게 설치됐는지 그리고 터미널 창에서 문제없이 실행할 수 있는지만 확인해주면 됩니다.

GRASS algorithms use a region for calculations. This region can be defined manually or automatically, taking the minimum extent that covers all the input layers used to execute the algorithm each time.

### LAStools

To use LAStools in QGIS, you need to download and install LAStools on your computer and install the LAStools plugin (available from the official repository) in QGIS.

On Linux platforms, you will need Wine to be able to run some of the tools.

공간 처리 옵션 대화창 (*Settings* ▸ *Options*, *Processing* 탭, *Providers* ▸ *LAStools*) 에서 LAStools 를 활성화하고 환경 설정합니다. 여기에서 LAStools 의 위치 (*LAStools folder*) 및 Wine 의 위치 (*Wine folder*) 를 지정할 수 있습니다. 우분투의 경우, 기본 Wine 폴더는 `/usr/bin` 입니다.

### OTB 응용 프로그램

QGIS 공간 처리 프레임워크는 OTB 응용 프로그램을 완전하게 지원합니다.

OTB (Orfeo ToolBox) is an image processing library for remote sensing data. It also provides applications that provide image processing functionalities. The list of applications and their documentation are available in OTB CookBook

> **참고**
>
> OTB 는 QGIS 와 함께 배포되지 않으며 개별적으로 설치해야 한다는 사실을 기억하십시오. OTB 바이너리 패키지는 다운로드 페이지 에서 찾을 수 있습니다.

QGIS 공간 처리 프레임워크가 OTB 라이브러리를 찾을 수 있도록 환경 설정하려면:

1. 공간 처리 설정 대화창을 엽니다: *Settings* ▸ *Options* ▸ *Processing*
2. *Providers* 메뉴에서 *OTB* 항목을 볼 수 있습니다:
    1. *OTB* 항목을 펼치십시오.
    2. *OTB folder* 를 사용자가 OTB 를 설치한 위치로 설정하십시오.
    3. *OTB application folder* 를 사용자가 OTB 응용 프로그램을 설치한 위치 (`<PATH_TO_OTB_INSTALLATION>/lib/otb/applications`) 로 설정하십시오.
    4. *OK* 를 클릭해서 설정을 저장하고 대화창을 닫으십시오.

정확하게 설정했다면, *Processing Toolbox* 에서 OTB 알고리즘을 사용할 수 있습니다.

### QGIS 공간 처리 프레임워크에서 사용할 수 있는 OTB 설정 문서

- **OTB folder**: OTB 를 사용할 수 있는 디렉터리입니다.
- **OTB application folder**: OTB 응용 프로그램의 위치 (들) 입니다.

  경로를 여러 개 지정할 수 있습니다.
- **Logger level** (부가적): OTB 응용 프로그램이 사용할 로그 작업자 (logger) 의 수준을 설정합니다.

  로그 작업의 수준은 알고리즘 실행 도중 출력되는 상세 내용의 양을 제어합니다. 로그 작업자 수준을 위한 값으로 `INFO`, `WARNING`, `CRITICAL`, `DEBUG` 가운데 하나를 사용할 수 있습니다. 기본값은 `INFO` 입니다. 이 옵션은 고급 사용자용 환경 설정입니다.
- **Maximum RAM to use** (부가적): OTB 응용 프로그램은 기본적으로 모든 가용 시스템 RAM 을 사용합니다.

  하지만 이 옵션을 통해 OTB 가 RAM 을 (MB 단위의) 특정 용량만 사용하도록 할 수 있습니다. 256 으로 설정하면 OTB 처리 제공자가 무시할 것입니다. 이 옵션은 고급 사용자용 환경 설정입니다.
- **Geoid file** (부가적): 지오이드 (geoid) 파일을 가리키는 경로를 지정합니다.

  이 옵션은 OTB 응용 프로그램에서 사용하는 `elev.dem.geoid` 와 `elev.geoid` 파라미터의 값을 설정합니다. 이 값을 전체 수준으로 설정하면 여러 처리 알고리즘이 공통으로 사용하게 할 수 있습니다. 기본값은 비어 있습니다.
- **SRTM tiles folder** (부가적): SRTM 타일을 사용할 수 있는 디렉터리입니다.

  처리 과정 동안 파일 다운로드를 막기 위해 SRTM 데이터를 사용자 시스템에 저장할 수 있습니다. 이 옵션은 OTB 응용 프로그램에서 사용하는 `elev.dem.path` 와 `elev.dem` 파라미터의 값을 설정합니다. 이 값을 전체 수준으로 설정하면 여러 처리 알고리즘이 공통으로 사용하게 할 수 있습니다. 기본값은 비어 있습니다.

### 호환성 및 문제 해결

OTB 6.6.1 버전부터, OTB 의 새 배포판은 적어도 배포 시기에 사용할 수 있는 최신 QGIS 버전과 호환되도록 만들어지고 있습니다.

QGIS 공간 처리 프레임워크에서 OTB 응용 프로그램 사용에 문제가 발생한 경우, OTB 버그 추적기 에 `qgis` 라벨을 사용해서 스레드를 여십시오.

OTB 및 QGIS 관련 추가 정보는 OTB 쿡북 에서 찾아볼 수 있습니다.

CHAPTER 24

# 공간 처리 제공자 및 알고리즘

이 문서는 공간 처리 알고리즘 및 파라미터를 (사용자 인터페이스에 나타난 대로) 설명합니다.

## 24.1 QGIS 알고리즘 제공자

QGIS 알고리즘 제공자는 거의 QGIS API 만 사용해서 다양한 분석 및 지리 공간 처리 작업을 구현합니다. 따라서 QGIS 제공자의 거의 모든 알고리즘은 어떤 추가적인 환경설정 없이도"설치 후 바로"작동할 것입니다.

QGIS 제공자는 플러그인의 알고리즘을 일부 포함하며, QGIS 고유 알고리즘도 추가합니다.

### 24.1.1 3D 타일

#### B3DM 을 GLTF 로 변환하기

파일을 레거시 `.B3DM` 포맷에서 `.GLTF` 또는 `.GLB` 포맷으로 변환합니다.

##### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input B3DM** | `INPUT` | [3D Tile] | 변환할 입력 파일입니다. |
| **Output file** | `OUTPUT` | [3D Tile]<br>기본값: `[Save to temporary file]` | Specify the output 3D tile file. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

#### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output file** | `OUTPUT` | [3D Tile] | `.GLTF` 포맷의 산출 레이어입니다. |

#### 파이썬 코드

**알고리즘 ID**: `native:b3dmtogltf`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### GLTF 를 벡터 피처로 변환하기

`.GLTF` 또는 `.GLB` 파일의 내용을 표준 벡터 레이어 포맷으로 변환합니다.

#### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input GLTF** | `INPUT` | [3D Tile] | 변환할 입력 파일입니다. |
| **Output polygons**<br>선택적 | `OUTPUT_POLYGONS` | [vector: polygon]<br>기본값: `[Create temporary layer]` | Specify the output vector layer to store polygon features. *One of*:<br>• 산출물 건너뛰기<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |
| **Output lines**<br>선택적 | `OUTPUT_LINES` | [vector: line]<br>기본값: `[Create temporary layer]` | Specify the output vector layer to store line features. *One of*:<br>• 산출물 건너뛰기<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

#### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output polygons** | `OUTPUT_POLYGONS` | [vector: polygon] | 입력 레이어의 폴리곤 피처를 저장하고 있는 벡터 레이어입니다. |
| **Output lines** | `OUTPUT_LINES` | [vector: line] | 입력 레이어의 라인 피처를 저장하고 있는 벡터 레이어입니다. |

#### 파이썬 코드

**알고리즘 ID**: `native:gltftovector`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 24.1.2 지도 제작

#### 피처에 포인트 정렬

이 알고리즘은 또다른 참조 레이어에 있는 최근접 피처를 사용해서 입력 레이어의 포인트 피처를 정렬하는 데 필요한 기울기를 계산합니다. 산출 레이어에 최근접 참조 피처에 대한 각도 (시계 방향 도 단위) 로 채워진 새 필드를 추가합니다.

마커 심볼을 회전시키기 위해 계산된 회전 필드를 자동적으로 사용하도록 산출 레이어의 심볼을 설정할 수 있는 옵션도 있습니다. 원하는 경우, 포인트 정렬시 사용할 최대 거리를 설정해서 고립된 포인트가 멀리 있는 피처에 정렬되지 않도록 할 수 있습니다.

**힌트**

이 알고리즘은 건물 포인트 심볼이 가장 가까이 있는 도로 방향을 따르도록 하는 등의 활용 사례들을 위해 설계되었습니다.

포인트 피처를제자리 피처 수정 할 수 있습니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: point] | 기울기를 계산할 포인트 피처 |
| **Reference layer** | `REFERENCE_LAYER` | [vector: geometry] | 기울기 계산을 위해 가장 가까이 있는 피처를 찾아야 할 레이어 |
| **Maximum distance to consider**<br>선택적 | `MAX_DISTANCE` | [numeric: double]<br>기본값: 설정하지 않음 | 이 거리 내에서 참조 피처를 찾지 못할 경우, 포인트 피처에 기울기를 할당하지 않습니다. |
| **Angle field name** | `FIELD_NAME` | [string]<br>기본값: 'rotation' | 기울기 값을 저장할 필드 |
| **Automatically apply symbology** | `APPLY_SYMBOLOGY` | [boolean]<br>기본값: True | 각도 필드 값을 사용해서 피처의 심볼 마커를 기울입니다. |
| **Aligned layer** | `OUTPUT` | [vector: point]<br>기본값: `[Save to temporary file]` | Specify the rotated output vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Aligned layer** | `OUTPUT` | [vector: point] | 기울기 필드를 추가한 포인트 레이어입니다. QGIS 에 불러오면, 기본적으로 입력 레이어 심볼에 레이어 마커 심볼의 데이터 정의 기울기를 적용합니다. |

### 파이썬 코드

**알고리즘 ID**: `native:angletonearest`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 스타일 데이터베이스 결합

여러 QGIS 스타일 데이터베이스를 단일 스타일 데이터베이스로 결합합니다. 서로 다른 소스 데이터베이스에 동일한 이름을 가진 동일한 유형의 항목이 있을 경우 산출된 결합 데이터베이스에서 유일한 이름을 가지도록 재명명할 것입니다.

**더 보기**

프로젝트에서 스타일 데이터베이스 생성하기

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input databases** | `INPUT` | [file] [list] | QGIS 스타일 항목을 담고 있는 파일 |
| **Objects to combine** | `OBJECTS` | [enumeration] [list] | 입력 데이터베이스에 있는 스타일 항목 가운데 새 데이터베이스에 넣고자 하는 항목의 유형. 다음 가운데 하나일 수 있습니다:<br>• 0 —심볼<br>• 1 —색상표<br>• 2 —텍스트 서식<br>• 3 —라벨 설정 |
| **Output style database** | `OUTPUT` | [file]<br>기본값: `[Save to temporary file]` | Output `.XML` file combining the selected style items. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Color ramp count** | `COLORRAMPS` | [numeric: integer] | 색상표 개수 |
| **Label settings count** | `LABELSETTINGS` | [numeric: integer] | 라벨 설정 개수 |
| **Output style database** | `OUTPUT` | [file] | 선택한 스타일 항목을 결합한 산출 `.XML` 파일 |
| **Symbol count** | `SYMBOLS` | [numeric: integer] | 심볼 개수 |
| **Text format count** | `TEXTFORMATS` | [numeric: integer] | 텍스트 서식 개수 |

### 파이썬 코드

**알고리즘 ID**: `native:combinestyles`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 스타일에서 범주 렌더링 작업자 생성하기

벡터 레이어의 렌더링 작업자를 스타일 데이터베이스에서 일치하는 심볼을 사용해서 범주 렌더링 작업자로 설정합니다. 아무 스타일 파일도 지정하지 않은 경우, 사용자의 현재심볼 라이브러리 에서 나온 심볼을 대신 사용합니다.

표현식 또는 필드를 지정해서 렌더링 작업자를 위한 범주를 생성합니다. 각 범주는 지정한 QGIS XML 스타일 데이터베이스 내부에 존재하는 심볼과 개별적으로 맞춰집니다. 일치하는 심볼명이 나올 때마다, 범주의 심볼을 이 일치하는 심볼로 설정할 것입니다.

원한다면 심볼과 일치시킬 수 없는 범주 그리고 범주와 일치되지 않는 심볼 목록을 담은 테이블을 산출물로 지정할 수도 있습니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: any] | 범주화된 스타일을 적용할 벡터 레이어 |
| **Categorize using expression** | `FIELD` | [expression] | 피처를 범주화하기 위한 필드 또는 표현식 |
| **Style database (leave blank to use saved symbols)** | `STYLE` | [file] | 입력 레이어 범주에 적용할 심볼을 담고 있는 (`.XML`) 파일. 스타일 관리자의공유 심볼 도구에서 이 파일을 얻을 수 있습니다. 아무 파일도 지정하지 않은 경우, QGIS 로컬 심볼 라이브러리를 사용합니다. |
| **Use case-sensitive match to symbol names** | `CASE_SENSITIVE` | [boolean] 기본값: False | True 인 (체크한) 경우, 범주와 심볼의 이름 비교시 대소문자를 구분합니다. |
| **Ignore non-alphanumeric characters while matching** | `TOLERANT` | [boolean] 기본값: False | True 인 (체크한) 경우, 범주 및 심볼의 이름에 있는 알파벳 또는 숫자가 아닌 문자를 무시할 것입니다. 이름이 일치하는 지 비교하는 동안 허용도가 크게 올라갑니다. |

다음 페이지에 계속

표 24.9 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Non-matching categories**<br>선택적 | NON_MATCHING_CA | [vector: table]<br>기본값: [Skip output] | Output table for categories which do not match any symbol in the database. *One of*:<br>• 산출물 건너뛰기<br>• 임시 레이어 생성 (TEMPORARY_OUTPUT)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |
| **Non-matching symbol names**<br>선택적 | NON_MATCHING_SY | [vector: table]<br>기본값: [Skip output] | Output table for symbols from the provided style database which do not match any category. *One of*:<br>• 산출물 건너뛰기<br>• 임시 레이어 생성 (TEMPORARY_OUTPUT)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Non-matching categories** | NON_MATCHING_CA | [vector: table] | 지정한 스타일 데이터베이스에 있는 어떤 심볼과도 일치되지 않는 범주를 목록화 |
| **Non-matching symbol names** | NON_MATCHING_SY | [vector: table] | 입력 스타일 데이터베이스의 심볼 가운데 어떤 범주와도 일치하지 않는 심볼을 목록화 |
| **Categorized layer** | OUTPUT | [same as input] | 범주 스타일을 적용한 입력 벡터 레이어. 새 레이어를 산출하지 않습니다. |

### 파이썬 코드

**알고리즘 ID**: `native:categorizeusingstyle`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 프로젝트에서 스타일 데이터베이스 생성하기

QGIS 프로젝트에서 모든 (심볼, 색상표, 텍스트 서식 및 라벨 설정) 스타일 객체를 추출합니다.

추출한 심볼은 QGIS 스타일 데이터베이스 (XML 포맷) 에 저장되는데, 스타일 관리자 대화창을 통해 이 데이터베이스를 관리하고 가져올 수 있습니다.

**더 보기**

스타일 데이터베이스 결합

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input project (leave blank to use current)**<br>선택적 | INPUT | [file] | 스타일 항목을 추출할 QGIS 프로젝트 파일 |
| **Objects to extract** | OBJECTS | [enumeration] [list] | 입력 프로젝트에 있는 스타일 항목 가운데 새 데이터베이스에 넣고자 하는 항목의 유형. 다음 가운데 하나일 수 있습니다:<br>• 0 — 심볼<br>• 1 — 색상표<br>• 2 — 텍스트 서식<br>• 3 — 라벨 설정 |
| **Output style database** | OUTPUT | [file]<br>기본값: [Save to temporary file] | Specify the output .XML file for the selected style items. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Color ramp count** | COLORRAMPS | [numeric: integer] | 색상표 개수 |
| **Label settings count** | LABELSETTINGS | [numeric: integer] | 라벨 설정 개수 |
| **Output style database** | OUTPUT | [file] | 선택한 스타일 항목을 저장한 산출 .XML 파일 |
| **Symbol count** | SYMBOLS | [numeric: integer] | 심볼 개수 |
| **Text format count** | TEXTFORMATS | [numeric: integer] | 텍스트 서식 개수 |

### 파이썬 코드

**알고리즘 ID**: `native:stylefromproject`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 지도책 조판을 이미지로 내보내기

인쇄 조판 지도책을 이미지 파일 (예: PNG 또는 JPEG 이미지) 로 내보냅니다.

커버리지 레이어를 설정한 경우, 이 알고리즘에 노출된 선택한 조판의 지도책 설정을 덮어쓸 것입니다. 이 경우, 비어 있는 필터 또는 표현식에 의한 정렬이 해당 설정을 끌 것입니다.

### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Atlas layout** | `LAYOUT` | [layout] | 내보낼 조판 |
| **Coverage layer** 선택적 | `COVERAGE_LAYER` | [vector: geometry] | 지도책을 생성하는 데 사용할 레이어 |
| **Filter expression** | `FILTER_EXPRESSION` | [expression] | 지도책 객체들을 필터링하는 데 사용할 표현식 |
| **Sort expression** 선택적 | `SORTBY_EXPRESSI` | [expression] | 지도책 객체들을 정렬하는 데 사용할 표현식 |
| **Reverse sort order** | `SORTBY_REVERSE` | [boolean] 기본값: False | 정렬 순서를 반전할지 결정합니다. 정렬 표현식을 지정한 경우 쓰입니다. |
| **Output filename expression** | `FILENAME_EXPRESSION` | [expression] 기본값: 'output_' \|\| @atlas_featurenumber | 파일명을 생성하는 데 쓰일 표현식 |
| **Output folder** | `FOLDER` | [folder] | 이미지들이 생성될 대상 폴더 |

#### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Map layers to assign to unlocked map item(s)** 선택적 | `LAYERS` | [enumeration] [layer] | 내용물이 잠기지 않은 맵 항목 (들) 에 표시될 레이어들 |

다음 페이지에 계속

표 24.14 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Image format** | EXTENSION | [enumeration] 기본값: png | 생성된 산출물 (들) 의 파일 포맷. 사용할 수 있는 포맷 목록은 OS 와 설치된 드라이버에 따라 달라집니다. |
| **DPI** 선택적 | DPI 기본값: 설정하지 않음 | [numeric: double] | 산출 파일 (들) 의 DPI. 설정하지 않을 경우 인쇄 조판 설정값을 사용할 것입니다. |
| **Generate world file** | GEOREFERENCE | [boolean] 기본값: True | 월드 파일을 생성할지 여부를 결정합니다. |
| **Export RDF metadata** | INCLUDE_METADATA | [boolean] 기본값: True | RDF 메타데이터 (제목, 저자 등등) 를 생성할지 여부를 결정합니다. |
| **Enable antialiasing** | ANTIALIAS | [boolean] 기본값: True | 위신호 제거 (antialiasing) 를 활성화할지 여부를 결정합니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Image file** | OUTPUT | [file] | 지도책 조판이 생성한 이미지 파일 |

### 파이썬 코드

**알고리즘 ID**: `native:atlaslayouttoimage`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 지도책 조판을 PDF 로 내보내기 (다중 파일)

인쇄 조판 지도책을 여러 개의 PDF 파일들로 내보냅니다.

커버리지 레이어를 설정한 경우, 이 알고리즘에 노출된 선택한 조판의 지도책 설정을 덮어쓸 것입니다. 이 경우, 비어 있는 필터 또는 표현식에 의한 정렬이 해당 설정을 끌 것입니다.

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Atlas layout** | LAYOUT | [layout] | 내보낼 조판 |

다음 페이지에 계속

표 24.16 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Coverage layer**<br>선택적 | COVERAGE_LAYER | [vector: geometry] | 지도책을 생성하는 데 사용할 레이어 |
| **Filter expression** | FILTER_EXPRESSION | [expression] | 지도책 객체들을 필터링하는 데 사용할 표현식 |
| **Sort expression**<br>선택적 | SORTBY_EXPRESSI | [expression] | 지도책 객체들을 정렬하는 데 사용할 표현식 |
| **Reverse sort order** | SORTBY_REVERSE | [boolean]<br>기본값: False | 정렬 순서를 반전할지 결정합니다. 정렬 표현식을 지정한 경우 쓰입니다. |
| **Output filename**<br>선택적 | OUTPUT_FILENAME | [expression] | PDF 산출 파일의 이름 패턴 |
| **Output folder** | OUTPUT_FOLDER | [folder] | 산출 PDF 파일을 저장할 대상 폴더 |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Map layers to assign to unlocked map item(s)**<br>선택적 | LAYERS | [enumeration]<br>[layer] | 내용물이 잠기지 않은 맵 항목 (들) 에 표시될 레이어들 |
| **DPI**<br>선택적 | DPI<br>기본값: 설정하지 않음 | [numeric: double] | 산출 파일 (들) 의 DPI. 설정하지 않을 경우 인쇄 조판 설정값을 사용할 것입니다. |
| **Always export as vectors** | FORCE_VECTOR | [boolean]<br>기본값: False | 벡터 데이터를 벡터로 유지할지 여부를 결정합니다. |
| **Always export as raster** | FORCE_RASTER | [boolean]<br>기본값: False | 맵에 있는 모든 항목을 강제로 래스터화합니다. 이 파라미터는 FORCE_VECTOR 파라미터를 우선합니다. |
| **Append georeference information** | GEOREFERENCE | [boolean]<br>기본값: True | 월드 파일을 생성할지 여부를 결정합니다. |
| **Export RDF metadata** | INCLUDE_METADATA | [boolean]<br>기본값: True | RDF 메타데이터 (제목, 저자 등등) 를 생성할지 여부를 결정합니다. |
| **Disable tiled raster layer exports** | DISABLE_TILED | [boolean]<br>기본값: False | 래스터를 타일화할지 여부를 결정합니다. |
| **Simplify geometries to reduce output file size** | SIMPLIFY | [boolean]<br>기본값: True | 산출 파일 용량을 줄이기 위해 도형들을 단순화할지 여부를 결정합니다. |
| **Text export** | TEXT_FORMAT | [enumeration]<br>기본값: 0 | Determines if text should be exported as path or text objects. *Possible options* are:<br>• 0 - 항상 텍스트를 경로로 내보내기 (추천)<br>• 1 - 항상 텍스트를 텍스트 객체로 내보내기<br>• 2 - Prefer exporting texts as text objects |

다음 페이지에 계속

표 24.17 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Image compression** | IMAGE_COMPRESSION | [enumeration] 기본값: 0 | 이미지의 압축 수준 및 이미지 파일이 산출물을 인쇄하거나 외부 응용 프로그램에서 후처리하는 데 얼마나 적합할 수 있을지를 결정합니다. 다음 옵션 가운데 하나를 선택할 수 있습니다:<br>• 0 - 손실 압축 (JPEG)<br>• 1 - 비손실 압축 |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **PDF file** | OUTPUT | [file] | 내보낸 지도책 조판에 대응하는 PDF 파일 |

### 파이썬 코드

**알고리즘 ID**: `native:atlaslayouttomultiplepdf`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 지도책 조판을 PDF 로 내보내기 (단일 파일)

인쇄 조판 지도책을 PDF 파일 1 개로 내보냅니다.

커버리지 레이어를 설정한 경우, 이 알고리즘에 노출된 선택한 조판의 지도책 설정을 덮어쓸 것입니다. 이 경우, 비어 있는 필터 또는 표현식에 의한 정렬이 해당 설정을 끌 것입니다.

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Atlas layout** | LAYOUT | [layout] | 내보낼 조판 |
| **Coverage layer** 선택적 | COVERAGE_LAYER | [vector: geometry] | 지도책을 생성하는 데 사용할 레이어 |
| **Filter expression** | FILTER_EXPRESSION | [expression] | 지도책 객체들을 필터링하는 데 사용할 표현식 |

다음 페이지에 계속

표 24.19 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Sort expression**<br>선택적 | SORTBY_EXPRESSI | [expression] | 지도책 객체들을 정렬하는 데 사용할 표현식 |
| **Reverse sort order** | SORTBY_REVERSE | [boolean]<br>기본값: False | 정렬 순서를 반전할지 결정합니다. 정렬 표현식을 지정한 경우 쓰입니다. |
| **PDF file** | OUTPUT | [file]<br>기본값: [Save to temporary file] | Name (including path) of the output file. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Map layers to assign to unlocked map item(s)**<br>선택적 | LAYERS | [enumeration]<br>[layer] | 내용물이 잠기지 않은 맵 항목 (들) 에 표시될 레이어들 |
| **DPI**<br>선택적 | DPI<br>기본값: 설정하지 않음 | [numeric: double] | 산출 파일 (들) 의 DPI. 설정하지 않을 경우 인쇄 조판 설정값을 사용할 것입니다. |
| **Always export as vectors** | FORCE_VECTOR | [boolean]<br>기본값: False | 벡터 데이터를 벡터로 유지할지 여부를 결정합니다. |
| **Always export as raster** | FORCE_RASTER | [boolean]<br>기본값: False | 맵에 있는 모든 항목을 강제로 래스터화합니다. 이 파라미터는 FORCE_VECTOR 파라미터를 우선합니다. |
| **Append georeference information** | GEOREFERENCE | [boolean]<br>기본값: True | 월드 파일을 생성할지 여부를 결정합니다. |
| **Export RDF metadata** | INCLUDE_METADATA | [boolean]<br>기본값: True | RDF 메타데이터 (제목, 저자 등등) 를 생성할지 여부를 결정합니다. |
| **Disable tiled raster layer exports** | DISABLE_TILED | [boolean]<br>기본값: False | 래스터를 타일화할지 여부를 결정합니다. |
| **Simplify geometries to reduce output file size** | SIMPLIFY | [boolean]<br>기본값: True | 산출 파일 용량을 줄이기 위해 도형들을 단순화할지 여부를 결정합니다. |
| **Text export** | TEXT_FORMAT | [enumeration]<br>기본값: 0 | Determines if text should be exported as path or text objects. *Possible options* are:<br>• 0 - 항상 텍스트를 경로로 내보내기 (추천)<br>• 1 - 항상 텍스트를 텍스트 객체로 내보내기<br>• 2 - Prefer exporting texts as text objects |

다음 페이지에 계속

표 24.20 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Image compression** | IMAGE_COMPRESSION | [enumeration] 기본값: 0 | 이미지의 압축 수준 및 이미지 파일이 산출물을 인쇄하거나 외부 응용 프로그램에서 후처리하는 데 얼마나 적합할 수 있을지를 결정합니다. 다음 옵션 가운데 하나를 선택할 수 있습니다:<br>• 0 - 손실 압축 (JPEG)<br>• 1 - 비손실 압축 |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **PDF file** | OUTPUT | [file] | 내보낸 지도책 조판에 대응하는 PDF 파일 |

### 파이썬 코드

**알고리즘 ID**: `native:atlaslayouttopdf`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 인쇄 조판을 이미지로 내보내기

인쇄 조판을 이미지 파일 (예: PNG 또는 JPEG 이미지) 로 내보냅니다.

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Print layout** | LAYOUT | [layout] | 내보낼 조판 |
| **Image file** | OUTPUT | [file] 기본값: [Save to temporary file] | Name (including path) of the output file. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Map layers to assign to unlocked map item(s)**<br>선택적 | LAYERS | [enumeration] [layer] | 내용물이 잠기지 않은 맵 항목 (들) 에 표시될 레이어들 |
| **DPI**<br>선택적 | DPI<br>기본값: 설정하지 않음 | [numeric: double] | 산출 파일 (들) 의 DPI. 설정하지 않을 경우 인쇄 조판 설정값을 사용할 것입니다. |
| **Generate world file** | GEOREFERENCE | [boolean]<br>기본값: True | 월드 파일을 생성할지 여부를 결정합니다. |
| **Export RDF metadata** | INCLUDE_METADATA | [boolean]<br>기본값: True | RDF 메타데이터 (제목, 저자 등등) 를 생성할지 여부를 결정합니다. |
| **Enable antialiasing** | ANTIALIAS | [boolean]<br>기본값: True | 위신호 제거 (antialiasing) 를 활성화할지 여부를 결정합니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Image file** | OUTPUT | [file] | 내보낸 인쇄 조판에 대응하는 이미지 파일 |

### 파이썬 코드

**알고리즘 ID**: `native:printlayouttoimage`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 인쇄 조판을 PDF 로 내보내기

인쇄 조판을 PDF 파일로 내보냅니다.

#### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Print Layout** | LAYOUT | [layout] | 내보낼 조판 |
| **PDF file** | OUTPUT | [file]<br>기본값: [Save to temporary file] | Name (including path) of the output file. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

#### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Map layers to assign to unlocked map item(s)**<br>선택적 | LAYERS | [enumeration] [layer] | 내용물이 잠기지 않은 맵 항목 (들) 에 표시될 레이어들 |
| **DPI**<br>선택적 | DPI<br>기본값: 설정하지 않음 | [numeric: double] | 산출 파일 (들) 의 DPI. 설정하지 않을 경우 인쇄 조판 설정값을 사용할 것입니다. |
| **Always export as vectors** | FORCE_VECTOR | [boolean]<br>기본값: False | 벡터 데이터를 벡터로 유지할지 여부를 결정합니다. |
| **Always export as raster** | FORCE_RASTER | [boolean]<br>기본값: False | 맵에 있는 모든 항목을 강제로 래스터화합니다. 이 파라미터는 FORCE_VECTOR 파라미터를 우선합니다. |
| **Append georeference information** | GEOREFERENCE | [boolean]<br>기본값: True | 월드 파일을 생성할지 여부를 결정합니다. |
| **Export RDF metadata** | INCLUDE_METADATA | [boolean]<br>기본값: True | RDF 메타데이터 (제목, 저자 등등) 를 생성할지 여부를 결정합니다. |
| **Disable tiled raster layer exports** | DISABLE_TILED | [boolean]<br>기본값: False | 래스터를 타일화할지 여부를 결정합니다. |
| **Simplify geometries to reduce output file size** | SIMPLIFY | [boolean]<br>기본값: True | 산출 파일 용량을 줄이기 위해 도형들을 단순화할지 여부를 결정합니다. |

다음 페이지에 계속

표 24.26 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Text export** | `TEXT_FORMAT` | [enumeration] 기본값: 0 | Determines if text should be exported as path or text objects. *Possible options* are:<br>• 0 - 항상 텍스트를 경로로 내보내기 (추천)<br>• 1 - 항상 텍스트를 텍스트 객체로 내보내기<br>• 2 - Prefer exporting texts as text objects |
| **Image compression** | `IMAGE_COMPRESSION` | [enumeration] 기본값: 0 | 이미지의 압축 수준 및 이미지 파일이 산출물을 인쇄하거나 외부 응용 프로그램에서 후처리하는 데 얼마나 적합할 수 있을지를 결정합니다. 다음 옵션 가운데 하나를 선택할 수 있습니다:<br>• 0 - 손실 압축 (JPEG)<br>• 1 - 비손실 압축 |
| **Export layers as separate PDF files** | `SEPARATE_LAYERS` | [boolean] 기본값: False | 참으로 설정할 경우, 조판에 있는 맵 항목마다 레이어별로 개별 PDF 파일을 생성할 것입니다. 이 경우 기타 복잡한 조판 항목별로 개별 PDF 파일을 생성할 수도 있기 때문에 논리적으로 잘게 쪼개진 조판 구성 요소들을 담은 PDF 파일 집합이 산출될 수도 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **PDF file** | `OUTPUT` | [file] | 내보낸 인쇄 조판에 대응하는 PDF 파일 (들) |

### 파이썬 코드

**알고리즘 ID**: `native:printlayouttopdf`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 라벨 추출하기

렌더링한 맵에서 라벨 정보를 지정한 범위와 축척으로 추출합니다.

맵 테마를 지정한 경우, 렌더링한 맵의 가시성 및 심볼이 해당 테마와 일치할 것입니다. 맵 테마가 없다면, 프로젝트에서 가시화된 모든 레이어를 사용할 것입니다. 여기서 라벨 정보란 (포인트 도형 역할을 하는) 위치, 관련 레이어 이름 및 피처 ID, 라벨 텍스트, (도 단위, 시계 방향) 기울기, 여러 줄 정렬, 그리고 글꼴 세부 정보를 포함합니다.

**경고**

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Map extent** | `EXTENT` | [extent] | 맵에서 라벨을 추출할 범위<br>다음과 같은 메소드를 사용할 수 있습니다:<br>• 레이어로부터 계산…: 현재 프로젝트에 불러온 레이어의 범위를 사용합니다.<br>• 레이어 맵으로부터 계산…: 활성 프로젝트의조판 맵 항목 범위를 사용합니다.<br>• 북마크로부터 계산…: 저장되어 있는북마크 의 범위를 사용합니다.<br>• 맵 캔버스 범위 사용<br>• 캔버스에 그리기: 연산에 넣을 범위를 정하는 직사각형을 클릭 & 드래그합니다.<br>• 좌표를 `xmin`, `xmax`, `ymin`, `ymax` 로 입력 |
| **Map scale** | `SCALE` | [scale] | 추출한 라벨을 라벨 속성을 이용해서 이 축척에서 렌더링할 것입니다. |
| **Map theme**<br>선택적 | `MAP_THEME` | [maptheme] | 라벨을 추출할 레이어를 표시하는 맵 테마입니다. 맵 테마를 설정하지 않는 경우, 현재 가시화된 레이어의 라벨을 추출합니다. |
| **Include unplaced labels** | `INCLUDE_UNPLACED` | [boolean]<br>기본값: True | 충돌하는 (즉 제자리에 배치되지 않은) 라벨을 포함해서, 중첩하는 라벨들을 모두 추출해야 할지 여부를 지정합니다. |

다음 페이지에 계속

표 24.28 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Extracted labels** | `OUTPUT` | [vector: point]<br>기본값: `[Create temporary layer]` | Specify the output vector layer for the extent(s). *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Map resolution (in DPI)**<br>선택적 | `DPI`<br>기본값: 96.0 | [numeric: double] | |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Extracted labels** | `OUTPUT` | [vector: point] | 추출한 라벨을 표현하는 포인트 벡터 레이어입니다. 각 피처는 소스 (레이어, 피처 ID) 를 식별하는 속성 및 할당된 라벨 속성 (텍스트, 글꼴, 크기, 기울기, …) 을 가지고 있습니다. 레이어에는 라벨 작업 기본 스타일과 NULL 심볼도 적용됩니다.<br>**경고**<br>생성된 필드 가운데 일부가 10 글자를 초과하는 이름을 가질 수도 있기 때문에, 산출물을 ESRI 셰이프파일 포맷 (`.SHP`) 으로 저장하는 경우 QGIS 에서 레이어를 불러올 때 예상치 못 한 렌더링으로 이어질 수도 있습니다. |

### 파이썬 코드

**알고리즘 ID**: `native:extractlabels`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 인쇄 조판 맵 범위 레이어

인쇄 조판 맵의 (조판기 단위, 예를 들어참조 맵 단위의) 크기, 축척, 기울기를 지정하는 속성과 함께 인쇄 조판 맵 항목 (들) 의 범위를 담고 있는 폴리곤 레이어를 생성합니다.

맵 항목 파라미터를 지정한 경우, 일치하는 맵의 범위만 내보낼 것입니다. 지정하지 않은 경우, 조판에 있는 모든 맵의 범위를 내보낼 것입니다.

부가적으로, 특정 산출 좌표계를 지정할 수 있습니다. 지정하지 않으면, 원본 맵 항목의 좌표계를 사용할 것입니다.

### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Print layout** | `LAYOUT` | [enumeration] | 현재 프로젝트의 인쇄 조판 |
| **Map item**<br>선택적 | `MAP` | [enumeration]<br>기본값: *All the map items* | 사용자가 추출하려는 정보를 가진 맵 항목 (들). 아무 항목도 지정하지 않으면 모든 맵 항목을 처리합니다. |
| **Extent** | `OUTPUT` | [vector: polygon]<br>기본값: `[Create temporary layer]` | Specify the output vector layer for the extent(s). *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

#### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Overrride CRS**<br>선택적 | `CRS` | [crs]<br>기본값: *The layout CRS* | 정보를 보고할 레이어용 좌표계를 선택합니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Map height** | HEIGHT | [numeric: double] | |
| **Extent** | OUTPUT | [vector: polygon] | 입력 조판 맵 항목 (들) 의 범위를 모두 담고 있는 산출 폴리곤 벡터 레이어 |
| **Map rotation** | ROTATION | [numeric: double] | |
| **Map scale** | SCALE | [numeric: double] | |
| **Map width** | WIDTH | [numeric: double] | |

### 파이썬 코드

**알고리즘 ID**: `native:printlayoutmapextenttolayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 레이어 스타일 설정

입력한 스타일을 레이어에 적용합니다. 이 스타일은 QML 파일로 정의되어 있어야만 합니다.

어떤 새 산출물도 생성하지 않습니다. 스타일은 즉시 레이어에 할당됩니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input Layer** | INPUT | [layer] | 스타일을 적용하려는 입력 레이어 |
| **Style file** | STYLE | [file] | `.qml` 스타일 파일을 가리키는 경로 |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| | OUTPUT | [same as input] | 새 스타일을 할당한 입력 레이어입니다. 새 레이어를 산출하지 않습니다. |

### 파이썬 코드

**알고리즘 ID**: `native:setlayerstyle`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 위상 색상 작업

필요한 색상의 개수를 최소화하면서, 인접한 폴리곤이 동일한 색상 인덱스를 공유하지 않도록 폴리곤 피처에 색상 인덱스를 할당합니다.

이 알고리즘은 색상 할당시 사용할 방법을 선택할 수 있습니다.

원한다면 최소 색상 개수를 지정할 수 있습니다. 색상 인덱스는 **color_id** 라는 새 속성으로 저장됩니다.

다음 예시는 서로 다른 색상 4 개를 선택한 알고리즘을 보여줍니다. 그림에서 볼 수 있듯이 각 색상 클래스가 동일한 개수의 피처에 적용됩니다.

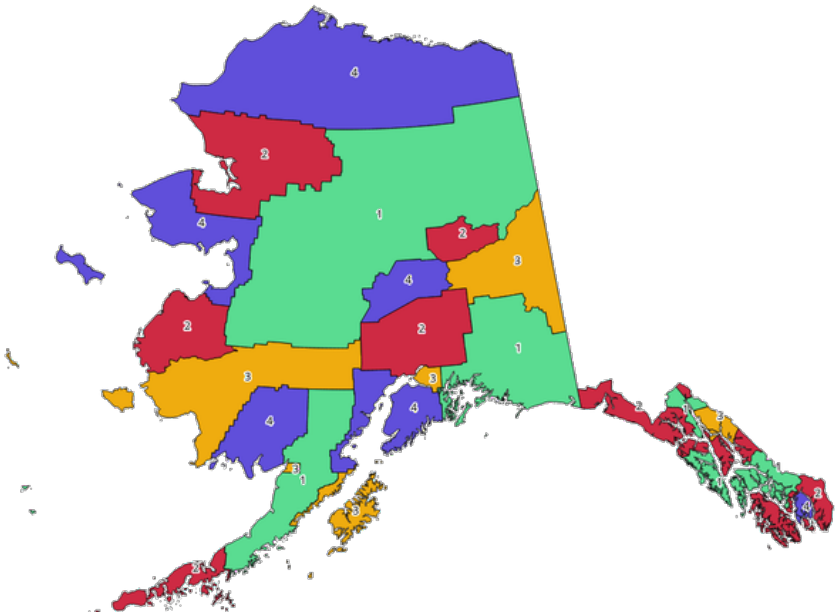


그림 24.1: 위상 색상 예시

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: polygon] | 입력 폴리곤 레이어 |
| **Minimum number of colors** | `MIN_COLORS` | [numeric: integer]<br>기본값: 4 | 할당할 색상의 최소 개수. 최소값은 1, 최대값은 1000 입니다. |
| **Minimum distance between features** | `MIN_DISTANCE` | [numeric: double]<br>기본값: 0.0 | 가까이 있는 (그러나 접하지는 않는) 피처에 동일한 색상을 할당하는 일을 방지합니다. 최소값은 0.0 입니다. |
| **Balance color assignment** | `BALANCE` | [enumeration]<br>기본값: 0 | 옵션은 다음과 같습니다:<br>• 0 —피처 개수로<br>각 개별 색상 인덱스가 할당된 피처의 개수가 균형을 이루도록 시도합니다.<br>• 1 —할당 면적으로<br>각 색상이 할당된 피처의 총 면적이 균형을 이루도록 색상을 할당합니다. 이 모드는 색상을 적용한 맵에서 어느 한 색상이 큰 피처에 집중되어 우세해 보이지 않도록 하는 데 유용합니다.<br>• 2 —색상들 사이의 거리로<br>동일한 색상의 피처 사이의 거리를 최대화하는 방향으로 색상을 할당합니다. 이 모드는 맵 전체에서 색상이 보다 균등하게 배포되도록 해줍니다. |
| **Colored** | `OUTPUT` | [vector: polygon]<br>기본값: `[Create temporary layer]` | Specify the output layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Colored** | `OUTPUT` | [vector: polygon] | `color_id` 열이 추가된 폴리곤 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `qgis:topologicalcoloring`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 주 레이어로부터 주석 이전

프로젝트에 있는 주요 주석 레이어로부터 새로운 주석 레이어로 모든주석 을 이전합니다. 그 다음 레이어 스택에서 항목 배치를 조정할 수 있습니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **New layer name** | `LAYER_NAME` | [string] 기본값: 'Annotations' | 생성할 주석 레이어의 이름 |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **New layer name** | `OUTPUT` | [layer] | 주요 주석 레이어로부터 나온 항목들을 가진 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:transferannotationsfrommain`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 24.1.3 Check Geometry

#### Dangle-end lines

Detects dangle-end lines in line geometries and reports them as errors. A dangle-end line is a line feature that terminates at a vertex connected to only one segment, resulting in an endpoint without a proper connection.

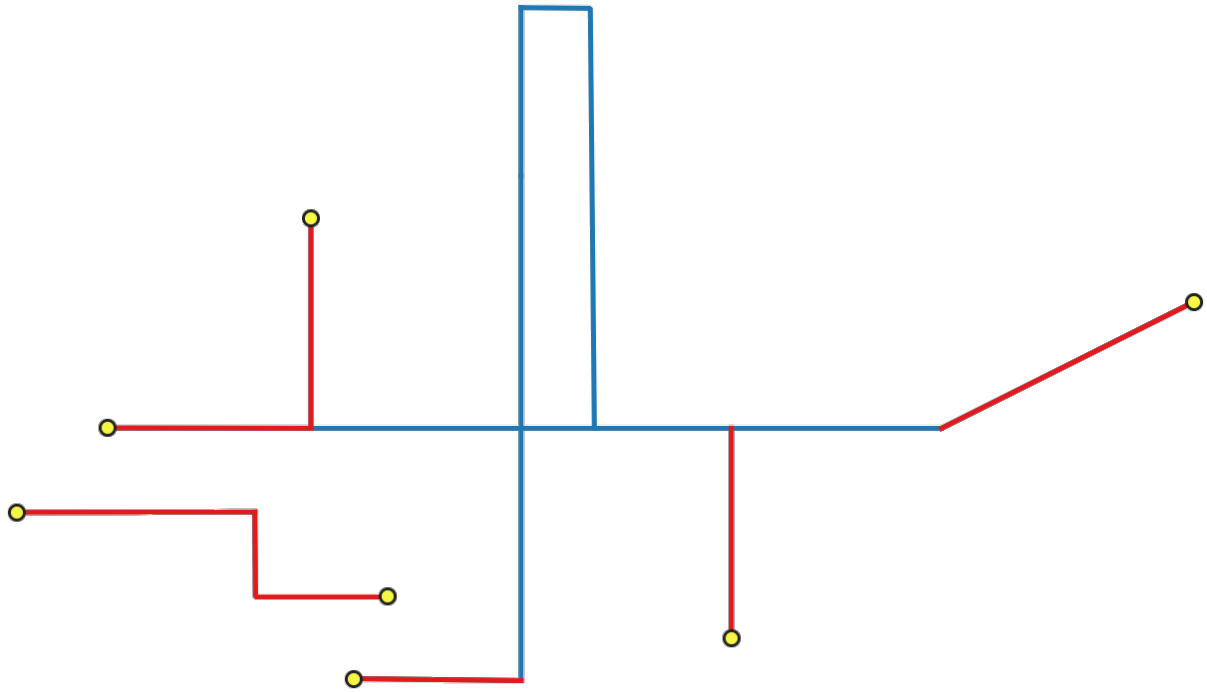


그림 24.2: Identifying dangle-end features (in red) and vertices (yellow).

#### Parameters

#### Basic parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: line] | Layer with the geometries to check. |
| **Unique feature identifier** | `UNIQUE_ID` | [tablefield: any] | Field storing unique values for feature identification. |
| **Dangle-end errors** | `ERRORS` | [vector: point]<br>Default: `[Create temporary layer]` | Specification of the output layer containing the errors location. *One of*:<br>• Create Temporary Layer (`TEMPORARY_OUTPUT`)<br>• Save to File…<br>• Save to Geopackage…<br>• Save to Database Table…<br>The file encoding can also be changed here. |

다음 페이지에 계속

표 24.40 – 이전 페이지에서 계속

| Label | Name | Type | Description |
|---|---|---|---|
| **Dangle-end features** Optional | `OUTPUT` | [vector: line] Default: `[Skip output]` | Line layer containing the identified dangle-end line features. *One of*:<br>• Skip Output<br>• Create Temporary Layer (`TEMPORARY_OUTPUT`)<br>• Save to File…<br>• Save to Geopackage…<br>• Save to Database Table…<br>The file encoding can also be changed here. |

### Advanced parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Tolerance** | `TOLERANCE` | [numeric: integer] Default: 8 | Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than $10^{-n}$ apart (in map units) are considered to be merged. |

### Outputs

| Label | Name | Type | Description |
|---|---|---|---|
| **Dangle-end errors** | ERRORS | [vector: point] | Output point layer representing the error locations and information. The output layer contains the following fields:<br>• gc_layerid: the ID of the input layer.<br>• gc_layername: the name of the input layer.<br>• gc_partidx: the index of the feature's geometry part containing the dangle-end.<br>• gc_ringidx: the index of the feature's geometry ring containing the dangle-end.<br>• gc_vertidx: the vertex index of the dangle-end in the feature's geometry ring.<br>• gc_errorx: the x coordinate of the dangle-end.<br>• gc_errory: the y coordinate of the dangle-end.<br>• gc_error<br>• UNIQUE_ID field: the unique ID of the input feature with the dangle-end. |
| **Dangle-end features** | OUTPUT | [vector: line] | Output line layer containing the input dangle-end features. If no dangle-end features are found, the output layer will be empty. Available fields are the same as in the ERRORS output. |

### Python code

**Algorithm ID**: native:checkgeometrydangle

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 콘솔에서 공간 처리 알고리즘 사용 for details on how to run processing algorithms from the Python console.

### Degenerate polygons

Checks the polygons with less than 3 points, which are degenerate polygons. Degenerate polygons are errors.

> **참고**
>
> This algorithm detects invalid polygon geometries (degenerate polygons). To ensure these invalid features are not filtered out before processing, open Advanced options next to the *Input layer* and set *Invalid feature filtering* to `Do not Filter (Better Performance)`.

### Parameters

### Basic parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: polygon] | Layer with the geometries to check. |
| **Unique feature identifier** | `UNIQUE_ID` | [tablefield: any] | Field storing unique values for feature identification. |
| **Degenerate polygons errors** | `ERRORS` | [vector: point]<br>Default: `[Create temporary layer]` | Specification of the output layer containing the centroid of the degenerate polygons. *One of*:<br>• Create Temporary Layer (`TEMPORARY_OUTPUT`)<br>• Save to File…<br>• Save to Geopackage…<br>• Save to Database Table…<br>The file encoding can also be changed here. |
| **Degenerate polygons features**<br>Optional | `OUTPUT` | [vector: polygon]<br>Default: `[Skip output]` | Specification of the output layer containing the degenerate polygons. *One of*:<br>• Skip Output<br>• Create Temporary Layer (`TEMPORARY_OUTPUT`)<br>• Save to File…<br>• Save to Geopackage…<br>• Save to Database Table…<br>The file encoding can also be changed here. |

### Advanced parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Tolerance** | `TOLERANCE` | [numeric: integer]<br>Default: 8 | Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than $10^{-n}$ apart (in map units) are considered to be merged. |

### Outputs

| Label | Name | Type | Description |
|---|---|---|---|
| **Degenerate polygons errors** | ERRORS | [vector: point] | Output point layer representing the error locations and information. The output layer contains the following fields:<br>• gc_layerid: the ID of the input layer.<br>• gc_layername: the name of the input layer.<br>• gc_partidx: the index of the geometry part of the degenerate polygon.<br>• gc_ringidx: the index of the ring of the degenerate polygon.<br>• gc_vertidx<br>• gc_errorx: the x coordinate of the centroid of the degenerate polygon<br>• gc_errory: the y coordinate of the centroid of the degenerate polygon<br>• gc_error<br>• UNIQUE_ID field: the unique ID of the input feature that is degenerate. |
| **Degenerate polygons features** | OUTPUT | [vector: polygon] | Output polygon layer with features containing the degenerate polygons. If no degenerate polygon features are found, the output layer will be empty. Available fields are the same as in the ERRORS output. |

### Python code

**Algorithm ID**: native:checkgeometrydegeneratepolygon

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 콘솔에서 공간 처리 알고리즘 사용 for details on how to run processing algorithms from the Python console.

### Duplicated geometries

Checks for duplicated geometries in a vector layer, and reports them as errors.

### Parameters

#### Basic parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | Layer with the geometries to check |
| **Unique feature identifier** | `UNIQUE_ID` | [tablefield: any] | Field storing unique values for feature identification |
| **Duplicate geometries errors** | `ERRORS` | [vector: point]<br>Default: `[Create temporary layer]` | Specification of the output layer containing the errors location. *One of*:<br>• Create Temporary Layer (`TEMPORARY_OUTPUT`)<br>• Save to File…<br>• Save to Geopackage…<br>• Save to Database Table…<br>The file encoding can also be changed here. |
| **Duplicate geometries**<br>Optional | `OUTPUT` | [vector: same as input]<br>Default: `[Skip output]` | Specification of the output layer containing the geometries that were found to be duplicated. *One of*:<br>• Skip Output<br>• Create Temporary Layer (`TEMPORARY_OUTPUT`)<br>• Save to File…<br>• Save to Geopackage…<br>• Save to Database Table…<br>The file encoding can also be changed here. |

#### Advanced parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Tolerance** | `TOLERANCE` | [numeric: integer]<br>Default: 8 | Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than $10^{-n}$ apart (in map units) are considered to be merged. |

### Outputs

| Label | Name | Type | Description |
|---|---|---|---|
| **Duplicate geometries errors** | ERRORS | [vector: point] | Output point layer representing the error locations and information. The output layer contains the following fields:<br>• gc_layerid: the ID of the input layer.<br>• gc_layername: the name of the input layer.<br>• gc_partidx<br>• gc_ringidx<br>• gc_vertidx<br>• gc_errorx: the x coordinate of the centroid of the duplicate geometry.<br>• gc_errory: the y coordinate of the centroid of the duplicate geometry.<br>• gc_error: the indices of the duplicate geometry (all except the highest index).<br>• UNIQUE_ID field: the unique ID of the input feature that is a duplicate. |
| **Duplicate geometries** | OUTPUT | [vector: same as input] | Output layer containing the geometries that were found to be duplicated. Available fields are the same as in the ERRORS output. |

### Python code

**Algorithm ID**: native:checkgeometryduplicate

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 콘솔에서 공간 처리 알고리즘 사용 for details on how to run processing algorithms from the Python console.

### Duplicated vertices

Checks for duplicated vertices in line or polygon geometries, and reports them as errors.

**더 보기**

*Delete duplicated vertices*, 복제 꼭짓점 제거하기

### Parameters

### Basic parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: line, polygon] | Layer with the geometries to check |
| **Unique feature identifier** | `UNIQUE_ID` | [tablefield: any] | Field storing unique values for feature identification |
| **Duplicated vertices errors** | `ERRORS` | [vector: point]<br>Default: `[Create temporary layer]` | Specification of the output layer containing the errors location. *One of*:<br>• Create Temporary Layer (`TEMPORARY_OUTPUT`)<br>• Save to File…<br>• Save to Geopackage…<br>• Save to Database Table…<br>The file encoding can also be changed here. |
| **Duplicated vertices features**<br>Optional | `OUTPUT` | [vector: same as input]<br>Default: `[Skip output]` | Specification of the output layer containing the features with duplicated vertices. *One of*:<br>• Skip Output<br>• Create Temporary Layer (`TEMPORARY_OUTPUT`)<br>• Save to File…<br>• Save to Geopackage…<br>• Save to Database Table…<br>The file encoding can also be changed here. |

### Advanced parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Tolerance** | `TOLERANCE` | [numeric: integer]<br>Default: 8 | Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than $10^{-n}$ apart (in map units) are considered to be merged. |

### Outputs

| Label | Name | Type | Description |
|---|---|---|---|
| **Duplicated vertices errors** | `ERRORS` | [vector: point] | Output point layer representing the error locations and information. The output layer contains the following fields:<br>• `gc_layerid`: the ID of the input layer.<br>• `gc_layername`: the name of the input layer.<br>• `gc_partidx`: the index of the feature's geometry part containing the duplicate vertex.<br>• `gc_ringidx`: the index of the feature's geometry ring containing the duplicate vertex.<br>• `gc_vertidx`: the index of the duplicate vertex in the feature's geometry ring.<br>• `gc_errorx`: the x coordinate of the duplicate vertex.<br>• `gc_errory`: the y coordinate of the duplicate vertex.<br>• `gc_error`<br>• `UNIQUE_ID` field: the unique ID of the input feature that has duplicate vertices. |
| **Duplicated vertices features** | `OUTPUT` | [vector: same as input] | Output layer containing the features with duplicated vertices. Available fields are the same as in the `ERRORS` output. |

### Python code

**Algorithm ID**: `native:checkgeometryduplicatenodes`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 콘솔에서 공간 처리 알고리즘 사용 for details on how to run processing algorithms from the Python console.

### Features inside polygon

Checks the input geometries contained in the polygons from the polygon layers list. A polygon layer can be checked against itself. Any contained features are reported as errors.

**더 보기**

*Delete features*

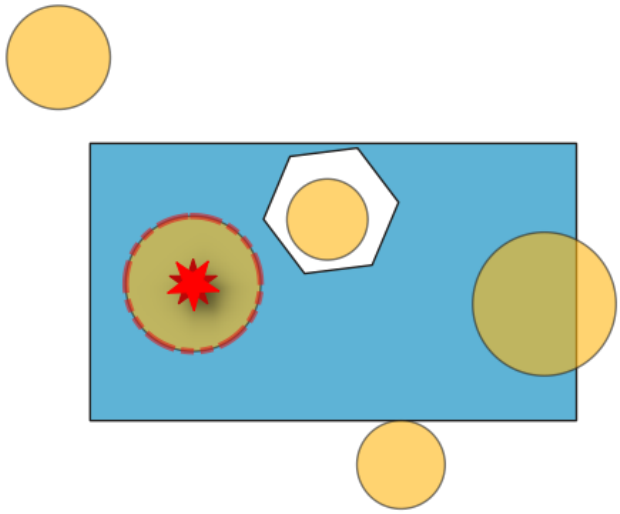


그림 24.3: Reporting errors on features inside a polygon.

### Parameters

### Basic parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | Layer with the geometries to check. |
| **Unique feature identifier** | `UNIQUE_ID` | [tablefield: any] | Field storing unique values for feature identification. |
| **Polygon layers** | `POLYGONS` | [vector: polygon][list] | List of polygon layers to check against. |
| **Errors from contained features** | `ERRORS` | [vector: point]<br>Default: `[Create temporary layer]` | Point layer representing the contained features. *One of*:<br>• Create Temporary Layer (`TEMPORARY_OUTPUT`)<br>• Save to File…<br>• Save to Geopackage…<br>• Save to Database Table…<br>The file encoding can also be changed here. |
| **Contained features**<br>Optional | `OUTPUT` | [vector: same as input]<br>Default: `[Skip output]` | Specification of the output layer containing features within the checked polygons. *One of*:<br>• Skip Output<br>• Create Temporary Layer (`TEMPORARY_OUTPUT`)<br>• Save to File…<br>• Save to Geopackage…<br>• Save to Database Table…<br>The file encoding can also be changed here. |

### Advanced parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Tolerance** | `TOLERANCE` | [numeric: integer]<br>Default: 8 | Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than $10^{-n}$ apart (in map units) are |

| Label | Name | Type | Description |
|---|---|---|---|
| **Errors from contained features** | ERRORS | [vector: point] | Output point layer representing the error locations and information. The output layer contains the following fields:<br>• gc_layerid: the ID of the input layer.<br>• gc_layername: the name of the input layer.<br>• gc_partidx<br>• gc_ringidx<br>• gc_vertidx<br>• gc_errorx: the x coordinate of the centroid of the contained feature.<br>• gc_errory: the y coordinate of the centroid of the contained feature.<br>• gc_error: the layer name and feature ID of a polygon that contains the feature.<br>• UNIQUE_ID field: the unique ID of the input feature that is contained within another. |
| **Contained features** | OUTPUT | [vector: same as input] | Output layer containing features within the checked polygons. Available fields are the same as in the ERRORS output. |

### Python code

**Algorithm ID**: native:checkgeometrycontained

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 콘솔에서 공간 처리 알고리즘 사용 for details on how to run processing algorithms from the Python console.

### Holes

Detects holes in polygon geometries and reports them as errors.

더 보기

*Fill holes*

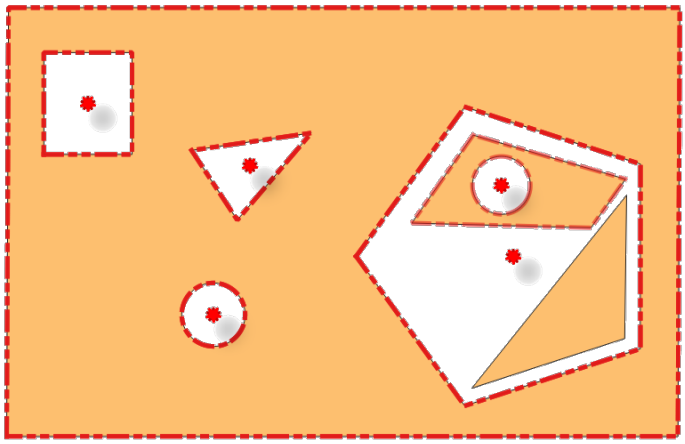


그림 24.4: Reporting errors for holes on a polygon feature.

### Parameters

#### Basic parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: polygon] | Layer with the geometries to check |
| **Unique feature identifier** | `UNIQUE_ID` | [tablefield: any] | Field storing unique values for feature identification |
| **Holes errors** | `ERRORS` | [vector: point]<br>Default: `[Create temporary layer]` | Specification of the output layer containing the errors location. *One of*:<br>• Create Temporary Layer (`TEMPORARY_OUTPUT`)<br>• Save to File…<br>• Save to Geopackage…<br>• Save to Database Table…<br>The file encoding can also be changed here. |
| **Polygons with holes**<br>Optional | `OUTPUT` | [vector: polygon]<br>Default: `[Skip output]` | Polygon layer with the features containing holes. *One of*:<br>• Skip Output<br>• Create Temporary Layer (`TEMPORARY_OUTPUT`)<br>• Save to File…<br>• Save to Geopackage…<br>• Save to Database Table…<br>The file encoding can also be changed here. |

#### Advanced parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Tolerance** | `TOLERANCE` | [numeric: integer]<br>Default: 8 | Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than $10^{-n}$ apart (in map units) are considered to be merged. |

| Label | Name | Type | Description |
|---|---|---|---|
| **Holes errors** | ERRORS | [vector: point] | Output point layer representing the error locations and information. The output layer contains the following fields:<br>• gc_layerid: the ID of the input layer.<br>• gc_layername: the name of the input layer.<br>• gc_partidx: the index of the feature's geometry part containing the hole.<br>• gc_ringidx: the index of the feature's geometry ring containing the hole.<br>• gc_vertidx<br>• gc_errorx: the x coordinate of the centroid of the hole.<br>• gc_errory: the y coordinate of the centroid of the hole.<br>• gc_error<br>• UNIQUE_ID field: the unique ID of the input feature that has a hole. |
| **Polygon with holes** | OUTPUT | [vector: polygon] | Output polygon layer with features containing holes. If no holes are found, the output layer will be empty. Available fields are the same as in the ERRORS output. |

### Python code

**Algorithm ID**: native:checkgeometryhole

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 콘솔에서 공간 처리 알고리즘 사용 for details on how to run processing algorithms from the Python console.

### Lines intersecting each other

Checks intersections between line geometries within a layer. Intersections between two different lines are errors.

더 보기

라인 교차, 교차 *(Intersection)*, *Self-intersections*

### Parameters

### Basic parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: line] | Layer with the geometries to check. |
| **Unique feature identifier** | `UNIQUE_ID` | [tablefield: any] | Field storing unique values for feature identification. |
| **Intersection errors** | `ERRORS` | [vector: point]<br>Default: `[Create temporary layer]` | Specification of the output layer containing the intersections location. *One of*:<br>• Create Temporary Layer (`TEMPORARY_OUTPUT`)<br>• Save to File…<br>• Save to Geopackage…<br>• Save to Database Table…<br>The file encoding can also be changed here. |
| **Intersecting feature**<br>Optional | `OUTPUT` | [vector: line]<br>Default: `[Skip output]` | Specification of the output layer for features intersecting each other. *One of*:<br>• Skip Output<br>• Create Temporary Layer (`TEMPORARY_OUTPUT`)<br>• Save to File…<br>• Save to Geopackage…<br>• Save to Database Table…<br>The file encoding can also be changed here. |

### Advanced parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Tolerance** | `TOLERANCE` | [numeric: integer]<br>Default: 8 | Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than $10^{-n}$ apart (in map units) are considered to be merged. |

### Outputs

| Label | Name | Type | Description |
|---|---|---|---|
| **Intersection errors** | `ERRORS` | [vector: point] | Output point layer representing the error locations and information. The output layer contains the following fields:<br>• `gc_layerid`: the ID of the input layer.<br>• `gc_layername`: the name of the input layer.<br>• `gc_partidx`: the index of the feature's geometry part where the intersection occurs.<br>• `gc_ringidx`<br>• `gc_vertidx`<br>• `gc_errorx`: the x coordinate of the intersection point.<br>• `gc_errory`: the y coordinate of the intersection point.<br>• `UNIQUE_ID` field: the unique ID of an intersecting input feature.<br>• `gc_error`: the layer name and index of the other intersecting feature. |
| **Intersecting feature** | `OUTPUT` | [vector: line] | Output layer containing, for each identified intersection, the feature (with same `UNIQUE_ID` value) it belongs to. If no intersections are found, the output layer will be empty. Available fields are the same as in the `ERRORS` output. |

### Python code

**Algorithm ID**: `native:checkgeometrylineintersection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 콘솔에서 공간 처리 알고리즘 사용 for details on how to run processing algorithms from the Python console.

### Lines intersecting other layer

Checks if the input line layer features intersect with the check layer features. An input feature that intersects with a check layer feature is an error.

**더 보기**

라인 교차, 교차 *(Intersection)*, *Self-intersections*

### Parameters

#### Basic parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: line] | Layer with the geometries to check. |
| **Unique feature identifier** | `UNIQUE_ID` | [tablefield: any] | Field storing unique values for feature identification. |
| **Check layer** | `CHECK_LAYER` | [vector: line, polygon] | Layer with the geometries to check against. |
| **Line intersecting other layer errors** | `ERRORS` | [vector: point]<br>Default: `[Create temporary layer]` | Specification of the output point layer containing the locations of intersection errors. *One of*:<br>• Create Temporary Layer (`TEMPORARY_OUTPUT`)<br>• Save to File…<br>• Save to Geopackage…<br>• Save to Database Table…<br>The file encoding can also be changed here. |
| **Line intersecting other layer features**<br>Optional | `OUTPUT` | [line]<br>Default: `[Skip output]` | Line layer with the input features that intersect features in the check layer. *One of*:<br>• Skip Output<br>• Create Temporary Layer (`TEMPORARY_OUTPUT`)<br>• Save to File…<br>• Save to Geopackage…<br>• Save to Database Table…<br>The file encoding can also be changed here. |

#### Advanced parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Tolerance** | `TOLERANCE` | [numeric: integer]<br>Default: 8 | Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than $10^{-n}$ apart (in map units) are considered to be merged. |

#### Outputs

| Label | Name | Type | Description |
|---|---|---|---|
| **Line intersecting other layer errors** | `ERRORS` | [vector: point] | Output point layer representing the error location and information. The output layer contains the following fields:<br>• `gc_layerid`: the ID of the input layer.<br>• `gc_layername`: the name of the input layer.<br>• `gc_partidx`: the index of the feature's geometry part where the intersection occurs.<br>• `gc_ringidx`<br>• `gc_vertidx`<br>• `gc_errorx`: the x coordinate of the intersection point.<br>• `gc_errory`: the y coordinate of the intersection point.<br>• `UNIQUE_ID` field: the unique ID of the intersecting input feature.<br>• `gc_error`: the index of the check layer where the intersection occurs. |
| **Line intersecting other layer features** | `OUTPUT` | [vector: line] | Output layer containing, for each identified intersection, the input feature it belongs to. If no intersections are found, the output layer will be empty. Available fields are the same as in the `ERRORS` output. |

### Python code

**Algorithm ID**: `native:checkgeometrylinelayerintersection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 콘솔에서 공간 처리 알고리즘 사용 for details on how to run processing algorithms from the Python console.

## Missing vertices along borders

Checks for missing vertices along polygon borders. To be topologically correct, a vertex at the junction of two polygons must be present on both polygons. Missing vertices are errors.

더 보기

*Add missing vertices along borders*

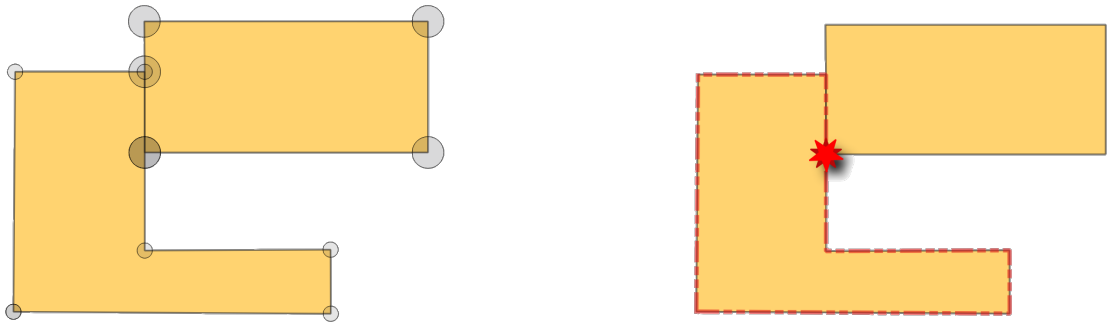


그림 24.5: Reporting errors for missing vertices on polygon features.

### Parameters

### Basic parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: polygon] | Layer with the geometries to check. |
| **Unique feature identifier** | `UNIQUE_ID` | [tablefield: any] | Field storing unique values for feature identification. |
| **Missing vertices errors** | `ERRORS` | [vector: point]<br>Default: `[Create temporary layer]` | Specification of the output layer containing the errors location. *One of*:<br>• Create Temporary Layer (`TEMPORARY_OUTPUT`)<br>• Save to File…<br>• Save to Geopackage…<br>• Save to Database Table…<br>The file encoding can also be changed here. |
| **Missing vertices features**<br>Optional | `OUTPUT` | [vector: polygon]<br>Default: `[Skip output]` | Polygon layer with the features whose vertices are missing. *One of*:<br>• Skip Output<br>• Create Temporary Layer (`TEMPORARY_OUTPUT`)<br>• Save to File…<br>• Save to Geopackage…<br>• Save to Database Table…<br>The file encoding can also be changed here. |

### Advanced parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Tolerance** | `TOLERANCE` | [numeric: integer]<br>Default: 8 | Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than $10^{-n}$ apart (in map units) are considered to be merged. |

### Outputs

| Label | Name | Type | Description |
|---|---|---|---|
| **Missing vertices errors** | `ERRORS` | [vector: point] | Output point layer representing the error locations and information. The output layer contains the following fields:<br>• `gc_layerid`: the ID of the input layer.<br>• `gc_layername`: the name of the input layer.<br>• `gc_partidx`<br>• `gc_ringidx`<br>• `gc_vertidx`<br>• `gc_errorx`: the x coordinate of the missing vertex.<br>• `gc_errory`: the y coordinate of the missing vertex.<br>• `gc_error`<br>• `UNIQUE_ID` field: the unique ID of the input feature that has a missing vertex. |
| **Missing vertices features** | `OUTPUT` | [vector: polygon] | Output polygon layer with features whose vertices are missing. There will be as many (duplicate) features as there are missing vertices in each geometry. If no vertices are missing, the output layer will be empty. Available fields are the same as in the `ERRORS` output. |

### Python code

**Algorithm ID**: `native:checkgeometrymissingvertex`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 콘솔에서 공간 처리 알고리즘 사용 for details on how to run processing algorithms from the Python console.

### Overlaps

Calculates overlapping areas in polygon geometries, and reports areas smaller than a given minimum as errors.

더 보기

*Delete overlaps*, *Validate coverage*

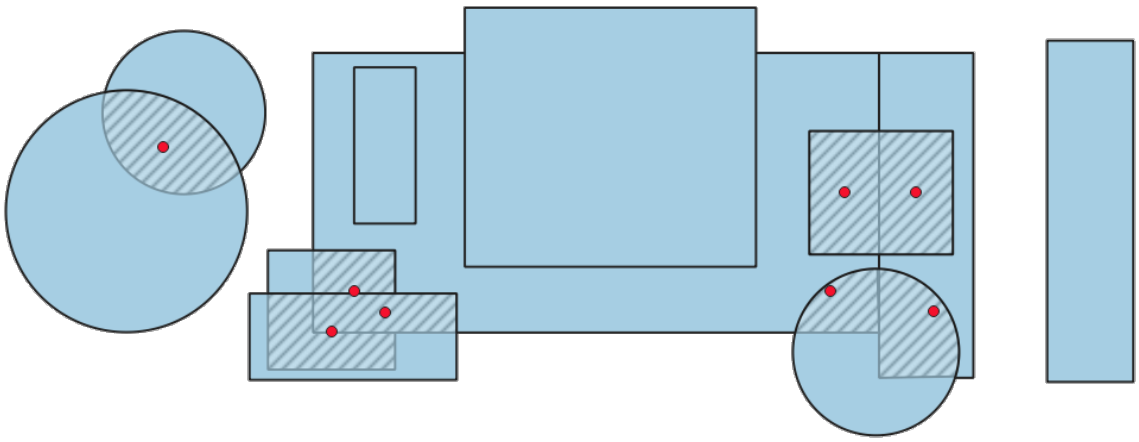


그림 24.6: Hashed polygons indicate overlapping areas smaller than the specified minimum.

### Parameters

### Basic parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: polygon] | Layer with the geometries to check. |
| **Unique feature identifier** | `UNIQUE_ID` | [tablefield: any] | Field storing unique values for feature identification. |
| **Minimum overlap area** | `MIN_OVERLAP_ARE` | [numeric: double] | Minimum area of the overlap to be reported as an error. If set to 0, all overlaps are reported. |
| **Overlap errors** | `ERRORS` | [vector: point] Default: `[Create temporary layer]` | Specification of the output layer containing centroid points of the overlapping areas. *One of*:<br>• Create Temporary Layer (`TEMPORARY_OUTPUT`)<br>• Save to File…<br>• Save to Geopackage…<br>• Save to Database Table…<br>The file encoding can also be changed here. |
| **Overlap features** Optional | `OUTPUT` | [vector: polygon] Default: `[Skip output]` | Specification of the output layer for features containing overlaps. *One of*:<br>• Skip Output<br>• Create Temporary Layer (`TEMPORARY_OUTPUT`)<br>• Save to File…<br>• Save to Geopackage…<br>• Save to Database Table…<br>The file encoding can also be changed here. |

### Advanced parameters



| Label | Name | Type | Description |
|---|---|---|---|
| **Tolerance** | `TOLERANCE` | [numeric: integer] | Numerical precision of geometric operations, |

### Outputs

| Label | Name | Type | Description |
|---|---|---|---|
| **Overlap errors** | ERRORS | [vector: point] | Output point layer representing the error locations and information. The output layer contains the following fields:<br>• gc_layerid: the ID of the input layer.<br>• gc_layername: the name of the input layer.<br>• gc_errorx: the x coordinate of the centroid of the overlapping area.<br>• gc_errory: the y coordinate of the centroid of the overlapping area.<br>• gc_error: the area of the overlapping geometry.<br>• UNIQUE_ID field: the unique ID of an overlapped input feature.<br>• gc_overlap_feature_{unique_id}: the UNIQUE_ID field value for the other overlapping feature. |
| **Overlap features** | OUTPUT | [vector: polygon] | Output layer containing the overlapping areas. Available fields are the same as in the ERRORS output. |

### Python code

**Algorithm ID**: native:checkgeometryoverlap

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 콘솔에서 공간 처리 알고리즘 사용 for details on how to run processing algorithms from the Python console.

### Points outside lines

Checks if the points in the input layer are covered by a line in the selected line layers. A point not covered by a line is an error.

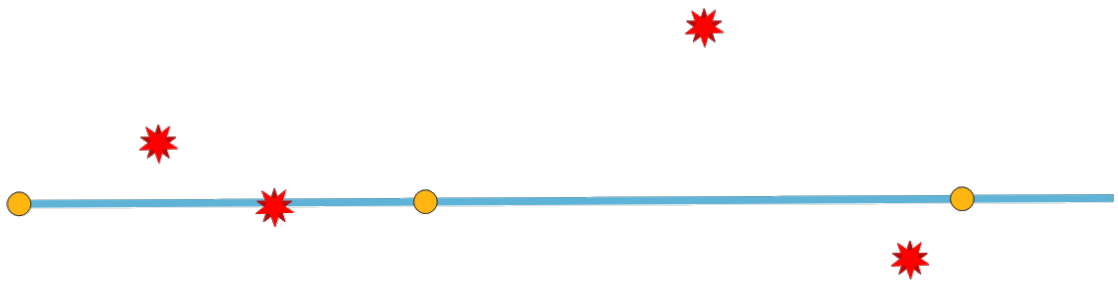


그림 24.7: Reporting errors on points not covered by a line.

### Parameters

### Basic parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: point] | Layer with the geometries to check. |
| **Unique feature identifier** | `UNIQUE_ID` | [tablefield: any] | Field storing unique values for feature identification. |
| **Line layers** | `LINES` | [vector: line][list] | Layer(s) with the lines to check against. |
| **Points not covered by a line** | `ERRORS` | [vector: point]<br>Default: `[Create temporary layer]` | Specification of the output layer containing points not covered by a line. *One of*:<br>• Create Temporary Layer (`TEMPORARY_OUTPUT`)<br>• Save to File…<br>• Save to Geopackage…<br>• Save to Database Table…<br>The file encoding can also be changed here. |

### Advanced parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Tolerance** | `TOLERANCE` | [numeric: integer]<br>Default: 8 | Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than $10^{-n}$ apart (in map units) are considered to be merged. |

### Outputs

| Label | Name | Type | Description |
|---|---|---|---|
| **Points not covered by a line** | ERRORS | [vector: point] | Output point layer representing the error locations and information. The output layer contains the following fields:<br>• gc_layerid: the ID of the input layer.<br>• gc_layername: the name of the input layer.<br>• gc_partidx<br>• gc_ringidx<br>• gc_vertidx<br>• gc_errorx: the x coordinate of point outside the lines.<br>• gc_errory: the y coordinate of point outside the lines.<br>• gc_error<br>• UNIQUE_ID field: the unique ID of the input feature that is outside the lines. |

### Python code

**Algorithm ID**: native:checkgeometrypointcoveredbyline

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 콘솔에서 공간 처리 알고리즘 사용 for details on how to run processing algorithms from the Python console.

## Points outside polygons

Checks if points from the input layer are in polygons from the selected polygon layers. Points that are not fully inside polygons are errors.

### Parameters

### Basic parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Input layer** | INPUT | [vector: point] | Layer with the geometries to check. |
| **Unique feature identifier** | UNIQUE_ID | [tablefield: any] | Field storing unique values for feature identification. |
| **Polygon layers** | POLYGONS | [vector: polygon][list] | Layer(s) with the polygons to check against. |

다음 페이지에 계속

표 24.73 – 이전 페이지에서 계속

| Label | Name | Type | Description |
|---|---|---|---|
| **Points outside polygons errors** | `ERRORS` | [vector: point] Default: `[Create temporary layer]` | Specification of the output layer containing points outside the polygons. *One of*:<br>• Create Temporary Layer (`TEMPORARY_OUTPUT`)<br>• Save to File…<br>• Save to Geopackage…<br>• Save to Database Table…<br>The file encoding can also be changed here. |

### Advanced parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Tolerance** | `TOLERANCE` | [numeric: integer] Default: 8 | Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than $10^{-n}$ apart (in map units) are considered to be merged. |

### Outputs

| Label | Name | Type | Description |
|---|---|---|---|
| **Points outside polygons errors** | `ERRORS` | [vector: point] | Output point layer representing the error locations and information. The output layer contains the following fields:<br>• `gc_layerid`: the ID of the input layer.<br>• `gc_layername`: the name of the input layer.<br>• `gc_partidx`<br>• `gc_ringidx`<br>• `gc_vertidx`<br>• `gc_errorx`: the x coordinate of point outside the polygons.<br>• `gc_errory`: the y coordinate of point outside the polygons.<br>• `gc_error`<br>• `UNIQUE_ID` field: the unique ID of the input feature that is outside the polygons. |

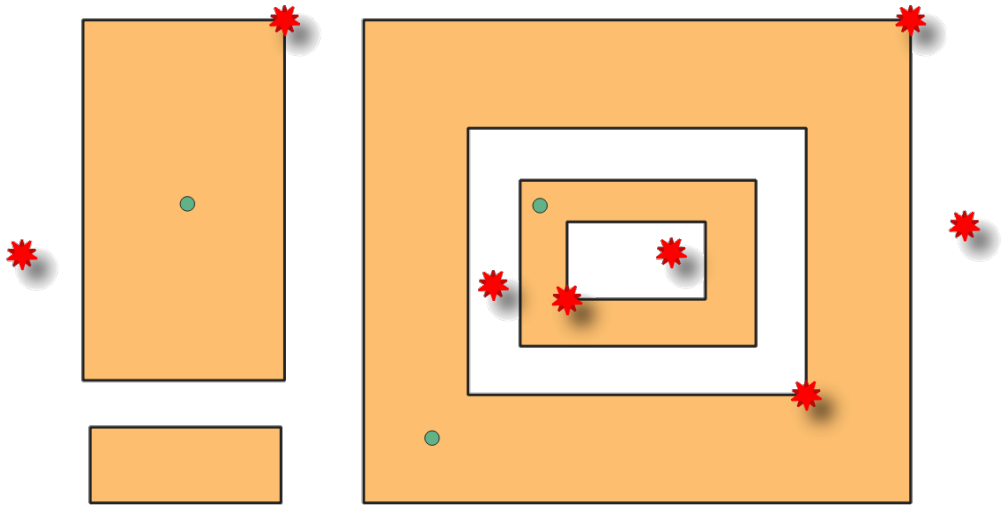


그림 24.8: Reporting errors on points outside polygons.

### Python code

**Algorithm ID**: `native:checkgeometrypointinpolygon`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 콘솔에서 공간 처리 알고리즘 사용 for details on how to run processing algorithms from the Python console.

### Self-contacts

Checks if the geometry has self contact points (in line or polygon), i.e., a vertex that touches more than two segments of the same ring. Self contacts are errors.

참고

This algorithm detects invalid polygon geometries (self-contacts). To ensure these invalid features are not filtered out before processing, open Advanced options next to the *Input layer* and set *Invalid feature filtering* to `Do not Filter (Better Performance)`.

더 보기

*Self-intersections*

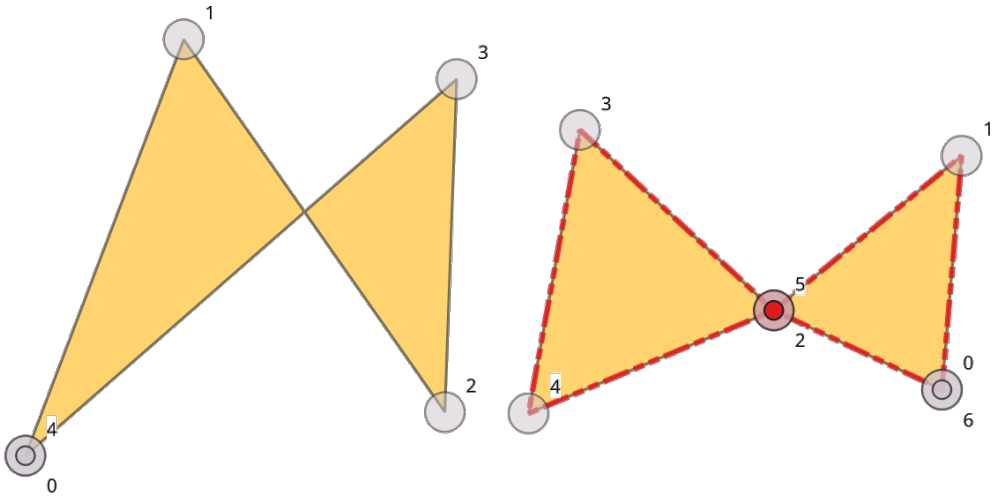



그림 24.9: Self-intersection vs self-contact.

### Parameters

### Basic parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: line, polygon] | Layer with the geometries to check. |
| **Unique feature identifier** | `UNIQUE_ID` | [tablefield: any] | Field storing unique values for feature identification. |
| **Self contact error points** | `ERRORS` | [vector: point]<br>Default: `[Create temporary layer]` | Specification of the output layer containing the errors location. *One of*:<br>• Create Temporary Layer (`TEMPORARY_OUTPUT`)<br>• Save to File…<br>• Save to Geopackage…<br>• Save to Database Table…<br>The file encoding can also be changed here. |
| **Self contact features**<br>Optional | `OUTPUT` | [same as input]<br>Default: `[Skip output]` | Line or polygon layer containing self-contact features. *One of*:<br>• Skip Output<br>• Create Temporary Layer (`TEMPORARY_OUTPUT`)<br>• Save to File…<br>• Save to Geopackage…<br>• Save to Database Table…<br>The file encoding can also be changed here. |

### Advanced parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Tolerance** | `TOLERANCE` | [numeric: integer]<br>Default: 8 | Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than $10^{-n}$ apart (in map units) are |

| Label | Name | Type | Description |
|---|---|---|---|
| **Self contact error points** | `ERRORS` | [vector: point] | Output point layer representing the error locations and information. The output layer contains the following fields:<br>• `gc_layerid`: the ID of the input layer.<br>• `gc_layername`: the name of the input layer.<br>• `gc_partidx`: the index of the feature's geometry part where the self-contact occurs.<br>• `gc_ringidx`: the index of the ring where the self-contact occurs.<br>• `gc_vertidx`: the index of the vertex where the self-contact occurs.<br>• `gc_errorx`: the x coordinate of the self-contact.<br>• `gc_errory`: the y coordinate of the self-contact.<br>• `gc_error`<br>• `UNIQUE_ID` field: the unique ID of the self-contacting input feature. |
| **Self contact features** | `OUTPUT` | [vector: same as input] | Output polygon or line layer with features containing the self-contact features. If no self-contact features are found, the output layer will be empty. Available fields are the same as in the `ERRORS` output. |

### Python code

**Algorithm ID**: `native:checkgeometryselfcontact`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 콘솔에서 공간 처리 알고리즘 사용 for details on how to run processing algorithms from the Python console.

### Self-intersections

Detects self-intersections in line or polygon geometries, and reports them as errors. Self-intersections occur when the segments of a geometry cross over each other without having a common vertex.

> **참고**
>
> This algorithm detects invalid polygon geometries (self-intersections). To ensure these invalid features are not filtered out before processing, open Advanced options next to the *Input layer* and set *Invalid feature filtering* to `Do not Filter (Better Performance)`.

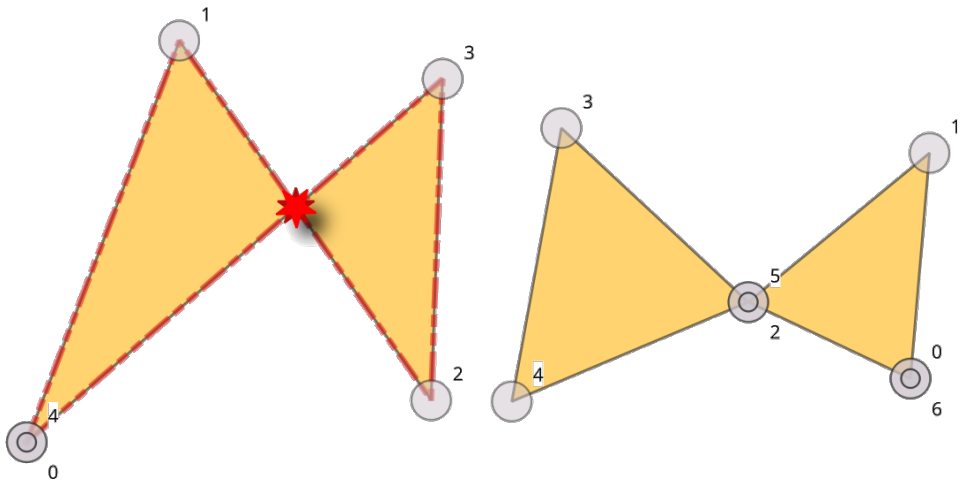



그림 24.10: Self-intersection vs self-contact.

더 보기

*Split self-intersecting geometries*, *Self-contacts*

### Parameters

### Basic parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: line, polygon] | Layer with the geometries to check |
| **Unique feature identifier** | `UNIQUE_ID` | [tablefield: any] | Field storing unique values for feature identification |
| **Self-intersecting errors** | `ERRORS` | [vector: point] Default: `[Create temporary layer]` | Specification of the output layer containing the errors location. *One of*:<br>• Create Temporary Layer (`TEMPORARY_OUTPUT`)<br>• Save to File…<br>• Save to Geopackage…<br>• Save to Database Table…<br>The file encoding can also be changed here. |

다음 페이지에 계속

표 24.79 – 이전 페이지에서 계속

| Label | Name | Type | Description |
|---|---|---|---|
| **Self-intersecting features**<br>Optional | `OUTPUT` | [same as input]<br>Default: `[Skip output]` | Line or polygon layer containing self-intersected features. *One of*:<br>• Skip Output<br>• Create Temporary Layer (`TEMPORARY_OUTPUT`)<br>• Save to File…<br>• Save to Geopackage…<br>• Save to Database Table…<br>The file encoding can also be changed here. |

### Advanced parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Tolerance** | `TOLERANCE` | [numeric: integer]<br>Default: 8 | Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than $10^{-n}$ apart (in map units) are considered to be merged. |

### Outputs

| Label | Name | Type | Description |
|---|---|---|---|
| **Self-intersecting errors** | ERRORS | [vector: point] | Output point layer representing the error locations and information. The output layer contains the following fields:<br>• gc_layerid: the ID of the input layer.<br>• gc_layername: the name of the input layer.<br>• gc_partidx: the index of the feature's geometry part where the self-intersection occurs.<br>• gc_ringidx: the index of the feature's geometry ring where the self-intersection occurs.<br>• gc_vertidx<br>• gc_errorx: the x coordinate of the self-intersection.<br>• gc_errory: the y coordinate of the self-intersection.<br>• gc_error<br>• gc_segment_1: the index of the first segment involved in the intersection.<br>• gc_segment_2: the index of the second segment involved in the intersection.<br>• UNIQUE_ID field: the unique ID of the self-intersecting input feature. |
| **Self-intersecting features** | OUTPUT | [vector: line, polygon] | Output polygon or line layer with features containing the self-intersecting features. If no self-intersecting features are found, the output layer will be empty. Available fields are the same as in the ERRORS output. |

### Python code

**Algorithm ID**: native:checkgeometryselfintersection

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 콘솔에서 공간 처리 알고리즘 사용 for details on how to run processing algorithms from the Python console.

### Sliver polygons

Detects sliver polygons in a polygon layer, i.e., polygons with a thinness greater than a specified value.

The thinness is the ratio between the area of the minimum square containing the polygon and the area of the polygon itself (a square has a thinness value of 1). The thinness value is between 1 and +infinity. A maximum area can be set for limiting the checks to polygons of a lower area. Polygons having a thinness higher than the maximum thinness are errors. To fix sliver polygons, use the *Fix small polygons* algorithm.

**참고**

This algorithm detects invalid polygon geometries (sliver polygons). To ensure these invalid features are not filtered out before processing, open Advanced options next to the *Input layer* and set *Invalid feature filtering* to `Do not Filter (Better Performance)`.

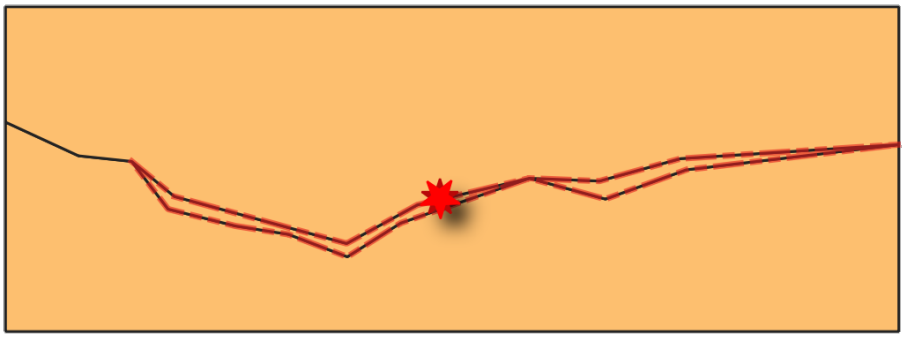


그림 24.11: Reporting errors on features with sliver polygons.

**더 보기**

*Fix small polygons*

### Parameters

### Basic parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: polygon] | Layer with the geometries to check. |
| **Unique feature identifier** | `UNIQUE_ID` | [tablefield: any] | Field storing unique values for feature identification. |
| **Maximum thinness** | `MAX_THINNESS` | [numeric: double] Default: 20.0 | Maximum thinness value of a polygon to be considered valid. Any polygon with a thinness value above this threshold is reported as error. |

다음 페이지에 계속

표 24.82 – 이전 페이지에서 계속

| Label | Name | Type | Description |
|---|---|---|---|
| **Maximum area (map units squared)** | `MAX_AREA` | [numeric: double]<br>Default: 0.0 | Maximum area of a polygon to be checked (in map units). Any polygon with an area above this threshold is skipped. A value of 0 means all polygons are checked. |
| **Sliver polygon errors** | `ERRORS` | [vector: point]<br>Default: `[Create temporary layer]` | Specification of the output layer containing the centroid points of sliver polygon features. *One of*:<br>• Create Temporary Layer (`TEMPORARY_OUTPUT`)<br>• Save to File…<br>• Save to Geopackage…<br>• Save to Database Table…<br>The file encoding can also be changed here. |
| **Sliver polygon features**<br>Optional | `OUTPUT` | [vector: polygon]<br>Default: `[Skip output]` | Polygon layer containing sliver polygon features. *One of*:<br>• Skip Output<br>• Create Temporary Layer (`TEMPORARY_OUTPUT`)<br>• Save to File…<br>• Save to Geopackage…<br>• Save to Database Table…<br>The file encoding can also be changed here. |

### Advanced parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Tolerance** | `TOLERANCE` | [numeric: integer]<br>Default: 8 | Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than $10^{-n}$ apart (in map units) are considered to be merged. |

### Outputs

| Label | Name | Type | Description |
|---|---|---|---|
| **Sliver polygon errors** | ERRORS | [vector: point] | Output point layer representing the error locations and information. The output layer contains the following fields:<br>• gc_layerid: the ID of the input layer.<br>• gc_layername: the name of the input layer.<br>• gc_partidx<br>• gc_ringidx<br>• gc_vertidx<br>• gc_errorx: the x coordinate of the centroid of the sliver polygon feature.<br>• gc_errory: the y coordinate of the centroid of the sliver polygon feature.<br>• gc_error: the thinness value of the sliver polygon feature.<br>• UNIQUE_ID field: the unique ID of the input feature that is a sliver polygon. |
| **Sliver polygon features** | OUTPUT | [vector: polygon] | Output layer with features containing the sliver polygons. If no sliver polygon features are found, the output layer will be empty. Available fields are the same as in the ERRORS output. |

### Python code

**Algorithm ID**: native:checkgeometrysliverpolygon

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 콘솔에서 공간 처리 알고리즘 사용 for details on how to run processing algorithms from the Python console.

### Small angles

Compares the angles within line or polygon geometries to a specified threshold, and reports as error any angle below that value.

더 보기

*Delete small angles*

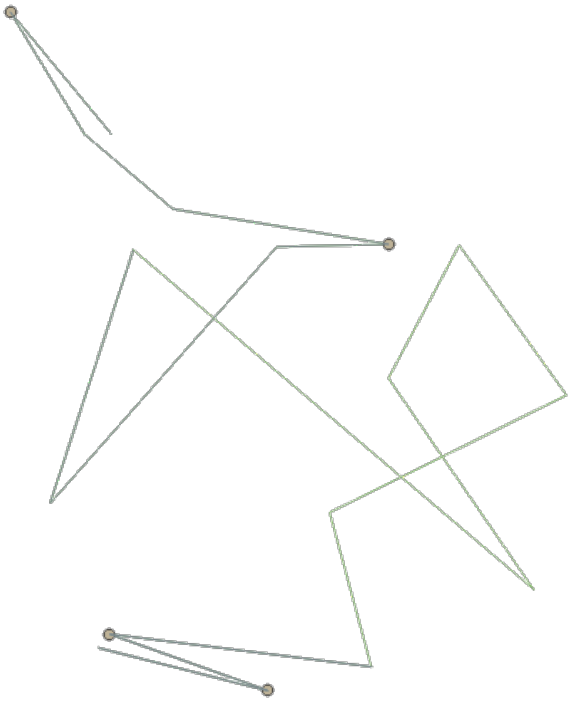


그림 24.12: Reporting errors on a line feature for angles lower than 15°.

### Parameters

#### Basic parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: line, polygon] | Layer with the geometries to check. |
| **Unique feature identifier** | `UNIQUE_ID` | [tablefield: any] | Field storing unique values for feature identification. |
| **Minimum angle** | `MIN_ANGLE` | [numeric: double] | Allowed minimum angle between adjacent segments in a feature's geometry. Any angle below this threshold is reported as error. |
| **Small angle errors** | `ERRORS` | [vector: point] Default: `[Create temporary layer]` | Specification of the output layer containing the errors location. *One of*:<br>• Create Temporary Layer (`TEMPORARY_OUTPUT`)<br>• Save to File…<br>• Save to Geopackage…<br>• Save to Database Table…<br>The file encoding can also be changed here. |

#### Advanced parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Tolerance** | `TOLERANCE` | [numeric: integer] Default: 8 | Numerical precision of geometric operations, given as an integer $n$, meaning that vertices less than $10^{-n}$ apart (in map units) are considered to be merged. |

| Label | Name | Type | Description |
|---|---|---|---|
| **Small angle errors** | `ERRORS` | [vector: point] | Output point layer representing the error locations and information. The output layer contains the following fields:<br>• `gc_layerid`: the ID of the input layer.<br>• `gc_layername`: the name of the input layer.<br>• `gc_partidx`: the index of the feature's geometry part containing the small angle.<br>• `gc_ringidx`: the index of the feature's geometry ring containing the small angle.<br>• `gc_vertidx`: the index of the vertex with the small angle in the feature's geometry ring.<br>• `gc_errorx`: the x coordinate of the vertex with the small angle.<br>• `gc_errory`: the y coordinate of the vertex with the small angle.<br>• `gc_error`: the error angle value.<br>• `UNIQUE_ID` field: the unique ID of the input feature with the small angle. |

### Python code

**Algorithm ID**: `native:checkgeometryangle`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 콘솔에서 공간 처리 알고리즘 사용 for details on how to run processing algorithms from the Python console.

### Small polygons

Detects polygon features whose area is below a specified value as errors.

**참고**

This algorithm detects invalid polygon geometries (small polygons). To ensure these invalid features are not filtered out before processing, open Advanced options next to the *Input layer* and set *Invalid feature filtering* to `Do not Filter (Better Performance)`.

**더 보기**

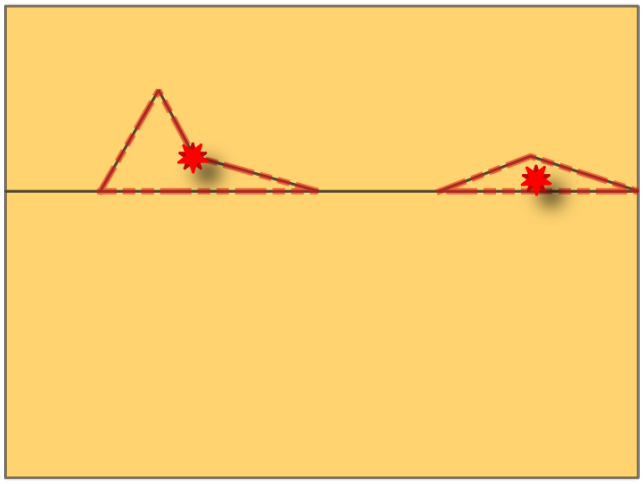


그림 24.13: Reporting errors on features with small polygons (below given area threshold).

*Fix small polygons*

### Parameters

### Basic parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: polygon] | Layer with the geometries to check. |
| **Unique feature identifier** | `UNIQUE_ID` | [tablefield: any] | Field storing unique values for feature identification. |
| **Area threshold** | `AREATHRESHOLD` | [numeric: double] Default: 0 | Minimum area of a polygon to be considered valid, in selected units. Any polygon with an area below this threshold is reported as error. |
| **Small polygons errors** | `ERRORS` | [vector: point] Default: `[Create temporary layer]` | Specification of the output layer containing centroid points of small polygons. *One of*:<br>• Create Temporary Layer (`TEMPORARY_OUTPUT`)<br>• Save to File…<br>• Save to Geopackage…<br>• Save to Database Table…<br>The file encoding can also be changed here. |

다음 페이지에 계속

표 24.88 – 이전 페이지에서 계속

| Label | Name | Type | Description |
|---|---|---|---|
| **Small polygons features**<br>Optional | `OUTPUT` | [vector: polygon]<br>Default: `[Skip output]` | Specification of the output layer for features containing small polygon features. *One of*:<br>• Skip Output<br>• Create Temporary Layer (`TEMPORARY_OUTPUT`)<br>• Save to File…<br>• Save to Geopackage…<br>• Save to Database Table…<br>The file encoding can also be changed here. |

### Advanced parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Tolerance** | `TOLERANCE` | [numeric: integer]<br>Default: 8 | Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than $10^{-n}$ apart (in map units) are considered to be merged. |

### Outputs

| Label | Name | Type | Description |
|---|---|---|---|
| **Small polygons errors** | `ERRORS` | [vector: point] | Output point layer representing the error locations and information. The output layer contains the following fields:<br>• `gc_layerid`: the ID of the input layer.<br>• `gc_layername`: the name of the input layer.<br>• `gc_partidx`<br>• `gc_ringidx`<br>• `gc_vertidx`<br>• `gc_errorx`: the x coordinate of the centroid of the small polygon.<br>• `gc_errory`: the y coordinate of the centroid of the small polygon.<br>• `gc_error`: the area of the small polygon.<br>• `UNIQUE_ID` field: the unique ID of the input feature that is a small polygon. |

다음 페이지에 계속

표 24.90 – 이전 페이지에서 계속

| Label | Name | Type | Description |
|---|---|---|---|
| **Small polygons features** | `OUTPUT` | [vector: polygon] | Output polygon layer with features containing the small polygon features. If no small polygon features are found, the output layer will be empty. Available fields are the same as in the `ERRORS` output. |

### Python code

**Algorithm ID**: `native:checkgeometryarea`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 콘솔에서 공간 처리 알고리즘 사용 for details on how to run processing algorithms from the Python console.

### Small gaps

Checks for gaps between polygons in the input layer. Gaps with an area smaller than the gap threshold are reported as errors. If an allowed gaps layer is provided, the algorithm ignores gaps that fall entirely within polygons from this layer. An optional buffer can be applied to the allowed gaps.

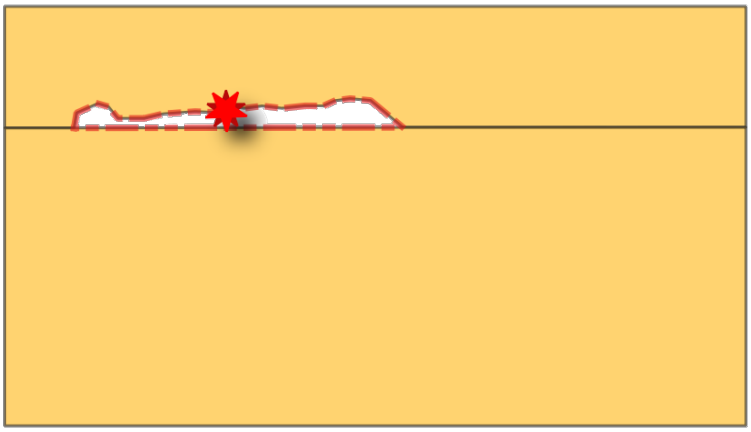


그림 24.14: Reporting errors on polygon features for gaps smaller than the specified threshold.

**더 보기**

*Fill gaps*, *Validate coverage*

### Parameters

#### Basic parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: polygon] | Layer with the geometries to check. |
| **Unique feature identifier** | `UNIQUE_ID` | [tablefield: any] | Field storing unique values for feature identification. |
| **Gap threshold** | `GAP_THRESHOLD` | [numeric: double] Default: 0.0 | Maximum area of gaps to be reported as errors, in map units. If set to 0, all the gaps are reported. |
| **Allowed gaps layer** Optional | `ALLOWED_GAPS_LAYE` | [vector: polygon] | Optional layer specifying polygons whose areas should be ignored during the gap check. Gaps that fall entirely within these polygons will not be reported as errors. |
| **Allowed gaps buffer** Optional | `ALLOWED_GAPS_BUFF` | [numeric: double] Default: 0.0 | Buffer distance to apply to the allowed gaps layer, in selected units. Gaps located within this buffered area are ignored. Allows adding a spatial tolerance around allowed gaps to avoid reporting near-boundary gaps as errors. |
| **Neighbors layer** | `NEIGHBORS` | [vector: table] Default: `[Create temporary layer]` | Specification of the output table representing the gap ID and the unique ID of its neighbor features. *One of*:<br>• Create Temporary Layer (`TEMPORARY_OUTPUT`)<br>• Save to File…<br>• Save to Geopackage…<br>• Save to Database Table…<br>The file encoding can also be changed here. |
| **Gap errors** Optional | `ERRORS` | [vector: point] Default: `[Skip output]` | Specification of the output layer containing the centroid points of the gaps. *One of*:<br>• Skip Output<br>• Create Temporary Layer (`TEMPORARY_OUTPUT`)<br>• Save to File…<br>• Save to Geopackage…<br>• Save to Database Table…<br>The file encoding can also be changed here. |
| **Gap features** | `OUTPUT` | [vector: polygon] Default: `[Create temporary layer]` | Specification of the output layer containing the gap geometries. *One of*:<br>• Create Temporary Layer (`TEMPORARY_OUTPUT`)<br>• Save to File…<br>• Save to Geopackage…<br>• Save to Database Table…<br>The file encoding can also be changed here. |

### Advanced parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Tolerance** | TOLERANCE | [numeric: integer] Default: 8 | Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than $10^{-n}$ apart (in map units) are considered to be merged. |

### Outputs

| Label | Name | Type | Description |
|---|---|---|---|
| **Neighbors layer** | NEIGHBORS | [vector: table] | A 1−N relational table, meaning that one gap can be associated with multiple neighboring polygons. The output table contains the following fields:<br>• gc_errorid: the ID of the gap.<br>• UNIQUE_ID field: the unique ID of an input feature that is a neighbor of the gap. |
| **Gap errors** | ERRORS | [vector: point] | Output point layer representing the error locations and information. The output layer contains the following fields:<br>• gc_layerid: the ID of the input layer.<br>• gc_layername: the name of the input layer.<br>• gc_partidx<br>• gc_ringidx<br>• gc_vertidx<br>• gc_errorx: the x coordinate of the centroid of the gap.<br>• gc_errory: the y coordinate of the centroid of the gap.<br>• gc_error: the area of the gap.<br>• gc_errorid: the ID of the gap. |
| **Gap features** | OUTPUT | [vector: polygon] | Output layer containing the gap geometries. Available fields are the same as in the ERRORS output. |

### Python code

**Algorithm ID**: `native:checkgeometrygap`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 콘솔에서 공간 처리 알고리즘 사용 for details on how to run processing algorithms from the Python console.

### Small segments

Calculates length of individual segments in line or polygon geometries, and reports segments shorter than a minimum length as errors.

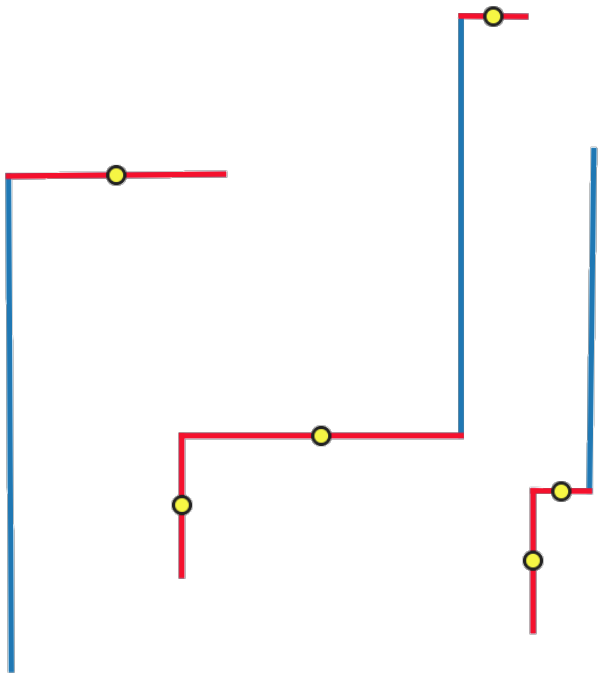


그림 24.15: Reporting errors (in red) on single line segments shorter than the given threshold.

**더 보기**

*Delete features*

### Parameters

#### Basic parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: line, polygon] | Layer with the geometries to check. |
| **Unique feature identifier** | `UNIQUE_ID` | [tablefield: any] | Field storing unique values for feature identification. |
| **Minimum segment length** | `MIN_SEGMENT_LEN` | [numeric: double]<br>Default: 0.0 | Minimum length of segments to be considered valid, in map units. |
| **Short segments errors** | `ERRORS` | [vector: point]<br>Default: `[Create temporary layer]` | Specification of the output layer containing centroid of short segments. *One of*:<br>• Create Temporary Layer (`TEMPORARY_OUTPUT`)<br>• Save to File…<br>• Save to Geopackage…<br>• Save to Database Table…<br>The file encoding can also be changed here. |
| **Short segments features**<br>Optional | `OUTPUT` | [vector: same as input]<br>Default: `[Skip output]` | Specification of the output layer for features containing short segments. *One of*:<br>• Skip Output<br>• Create Temporary Layer (`TEMPORARY_OUTPUT`)<br>• Save to File…<br>• Save to Geopackage…<br>• Save to Database Table…<br>The file encoding can also be changed here. |

#### Advanced parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Tolerance** | `TOLERANCE` | [numeric: integer]<br>Default: 8 | Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than $10^{-n}$ apart (in map units) are considered to be merged. |

#### Outputs

| Label | Name | Type | Description |
|---|---|---|---|
| **Short segments errors** | ERRORS | [vector: point] | Output point layer representing the error locations and information. The output layer contains the following fields:<br>• gc_layerid: the ID of the input layer.<br>• gc_layername: the name of the input layer.<br>• gc_partidx: the index of the feature's geometry part containing the small segment.<br>• gc_ringidx: the index of the feature's geometry ring containing the small segment.<br>• gc_vertidx: the index of the end vertex of the small segment in the feature's geometry ring.<br>• gc_errorx: the x coordinate of the centroid of the small segment.<br>• gc_errory: the y coordinate of the centroid of the small segment.<br>• gc_error: the error segment length.<br>• UNIQUE_ID field: the unique ID of the input feature with the small segment. |
| **Short segments features** | OUTPUT | [vector: same as input] | Output layer containing, for each identified short segment, the feature it belongs to. Available fields are the same as in the ERRORS output. |

### Python code

**Algorithm ID**: native:checkgeometrysegmentlength

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 콘솔에서 공간 처리 알고리즘 사용 for details on how to run processing algorithms from the Python console.

### Strictly multipart

Checks if multipart geometries have more than one part. Multipart geometries with only one part are errors.

더 보기

*Convert to strictly multipart*

### Parameters

### Basic parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | Layer with the geometries to check. |
| **Unique feature identifier** | `UNIQUE_ID` | [tablefield: any] | Field storing unique values for feature identification. |
| **One-part geometry errors** | `ERRORS` | [vector: point]<br>Default: `[Create temporary layer]` | Specification of the output layer containing centroid of the multipart geometries that have only one part. *One of*:<br>• Create Temporary Layer (`TEMPORARY_OUTPUT`)<br>• Save to File…<br>• Save to Geopackage…<br>• Save to Database Table…<br>The file encoding can also be changed here. |
| **One-part geometry features**<br>Optional | `OUTPUT` | [vector: same as input]<br>Default: `[Skip output]` | Specification of the output layer for features containing multipart geometries that have only one part. *One of*:<br>• Skip Output<br>• Create Temporary Layer (`TEMPORARY_OUTPUT`)<br>• Save to File…<br>• Save to Geopackage…<br>• Save to Database Table…<br>The file encoding can also be changed here. |

### Advanced parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Tolerance** | `TOLERANCE` | [numeric: integer]<br>Default: 8 | Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than $10^{-n}$ apart (in map units) are considered to be merged. |

#### Outputs

| Label | Name | Type | Description |
|---|---|---|---|
| **One-part geometry errors** | ERRORS | [vector: point] | Output point layer representing the error locations and information. The output layer contains the following fields:<br>• gc_layerid: the ID of the input layer.<br>• gc_layername: the name of the input layer.<br>• gc_partidx<br>• gc_ringidx<br>• gc_vertidx<br>• gc_errorx: the x coordinate of the centroid of the multipart geometry that has only one part.<br>• gc_errory: the y coordinate of the centroid of the multipart geometry that has only one part.<br>• gc_error<br>• UNIQUE_ID field: the unique ID of the input feature that is multipart but has only one part. |
| **One-part geometry features** | OUTPUT | [vector: same as input] | Output layer containing features that are multipart but have only one part. Available fields are the same as in the ERRORS output. |

#### Python code

**Algorithm ID**: native:checkgeometrymultipart

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 콘솔에서 공간 처리 알고리즘 사용 for details on how to run processing algorithms from the Python console.

### 24.1.4 데이터베이스

#### PostgreSQL 로 내보내기

새 관계를 생성하면서 벡터 레이어를 PostgreSQL 데이터베이스로 내보냅니다. 동일한 이름을 가진 관계가 존재하는 경우, 새 관계를 생성하기 전에 기존 관계를 제거할 수 있습니다. 이 알고리즘을 실행하기 전에 QGIS 와 PostgreSQL 데이터베이스 사이의 연결을 생성해야 합니다. (저장된 연결 생성하기 참조)

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Layer to import** | INPUT | [vector: any] | 데이터베이스에 추가할 벡터 레이어 |
| **Database (connection name)** | DATABASE | [string] | 데이터베이스 연결의 이름. (데이터베이스명이 아닙니다) 기존 연결은 콤보박스에 표시될 것입니다. |
| **Schema (schema name)**<br>선택적 | SCHEMA | [string]<br>기본값: 'public' | 데이터를 저장할 스키마의 이름. 새 스키마일 수도 있고 기존 스키마일 수도 있습니다. |
| **Table to import to (leave blank to use layer name)**<br>선택적 | TABLENAME | [string]<br>기본값: '' | 가져온 벡터 파일을 위한 테이블 이름을 정의합니다. 아무것도 추가하지 않으면 레이어명을 사용할 것입니다. |
| **Primary key field**<br>선택적 | PRIMARY_KEY | [tablefield: any] | 벡터 레이어에 있는 기존 필드로부터 기본 키 (primary key) 를 설정합니다. **유일한 (unique)** 값을 가진 열을 데이터베이스용 기본 키로 사용할 수 있습니다. |
| **Geometry column** | GEOMETRY_COLUMN | [string]<br>기본값: 'geom' | Defines the name of the geometry column in the new PostgreSQL table. Geometry information for the features is stored in this column. |
| **Encoding**<br>선택적 | ENCODING | [string]<br>기본값: 'UTF-8' | 산출 레이어의 인코딩을 정의합니다. |
| **Overwrite** | OVERWRITE | [boolean]<br>기본값: True | 지정한 테이블이 존재하는 경우, 이 옵션을 True 로 설정하면 피처를 추가하기 전에 기존 테이블을 삭제하고 새 테이블을 생성할 것입니다. 이 옵션의 값이 False 인데 기존 테이블이 존재하는 경우, 알고리즘이 예외를 던질 것입니다. ("relation already exists") |
| **Create spatial index** | CREATEINDEX | [boolean]<br>기본값: True | 공간 인덱스를 생성할지 여부를 지정합니다. |
| **Convert field names to lowercase** | LOWERCASE_NAMES | [boolean]<br>기본값: True | 입력 벡터 레이어의 필드명을 소문자로 변환합니다. |
| **Drop length constraint on character fields** | DROP_STRING_LEN | [boolean]<br>기본값: False | 문자 필드의 길이 제약 조건을 제거할지 여부를 지정합니다. |
| **Create single-part geometries instead of multi-part** | FORCE_SINGLEPAR | [boolean]<br>기본값: False | 산출 레이어의 피처가 다중 부분 대신 단일 부분이어야 하는지 여부를 지정합니다. 기존 도형 정보를 유지하는 것이 기본값입니다. |

#### 산출물

이 알고리즘은 아무것도 산출하지 않습니다.

#### 파이썬 코드

**Algorithm ID**: `native:importintopostgis`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### SpatiaLite 로 내보내기

벡터 레이어를 SpatiaLite 데이터베이스로 내보냅니다. 이 알고리즘을 실행하기 전에 QGIS 와 SpatiaLite 데이터베이스 사이의 연결을 생성해야 합니다. (*SpatiaLite* 레이어 참조)

#### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Layer to import** | `INPUT` | [vector: any] | 데이터베이스에 추가할 벡터 레이어 |
| **File database** | `DATABASE` | [vector: any] | 연결할 SQLite/SpatiaLite 데이터베이스 파일 |
| **Table to import to (leave blank to use layer name)**<br>선택적 | `TABLENAME` | [string]<br>기본값: ‘’ | 가져온 벡터 파일을 위한 테이블 이름을 정의합니다. 아무것도 지정하지 않으면 레이어명을 사용할 것입니다. |
| **Primary key field**<br>선택적 | `PRIMARY_KEY` | [tablefield: any] | 입력 벡터 레이어의 필드를 기본 키로 사용합니다. |
| **Geometry column** | `GEOMETRY_COLUMN` | [string]<br>기본값: ‘geom’ | 새 SpatiaLite 테이블의 도형 열의 이름을 정의합니다. 이 열에 객체에 대한 도형 정보를 저장합니다. |
| **Encoding**<br>선택적 | `ENCODING` | [string]<br>기본값: ‘UTF-8’ | 산출 레이어의 인코딩을 정의합니다. |
| **Overwrite** | `OVERWRITE` | [boolean]<br>기본값: True | 지정한 테이블이 존재하는 경우, 이 옵션을 `True` 로 설정하면 레이어의 피처를 추가하기 전에 기존 테이블을 삭제하고 새 테이블을 생성할 것입니다. 이 옵션의 값이 `False` 인데 기존 테이블이 존재하는 경우, 알고리즘이 예외를 던질 것입니다. (“table already exists”) |
| **Create spatial index** | `CREATEINDEX` | [boolean]<br>기본값: True | 공간 인덱스를 생성할지 여부를 지정합니다. |
| **Convert field names to lowercase** | `LOWERCASE_NAMES` | [boolean]<br>기본값: True | 입력 벡터 레이어의 필드명을 소문자로 변환합니다. |

다음 페이지에 계속

표 24.101 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Drop length constraint on character fields** | `DROP_STRING_LEN` | [boolean]<br>기본값: False | 문자 필드의 길이 제약 조건을 제거할지 여부를 지정합니다. |
| **Create single-part geometries instead of multi-part** | `FORCE_SINGLEPAR` | [boolean]<br>기본값: False | 산출 레이어의 피처가 다중 부분 대신 단일 부분이어야 하는지 여부를 지정합니다. 기존 도형 정보를 유지하는 것이 기본값입니다. |

### 산출물

이 알고리즘은 아무것도 산출하지 않습니다.

### 파이썬 코드

**Algorithm ID**: `native:importintospatialite`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 패키지 레이어

GeoPackage 에 레이어를 추가합니다.

GeoPackage 가 존재하고 `Overwrite existing GeoPackage` 옵션을 활성화했다면, 기존 GeoPackage 를 재작성할 (제거한 다음 새로 생성할) 만약 GeoPackage 가 존재하는데 `Overwrite existing GeoPackage` 옵션이 비활성 상태라면, 레이어를 추가할 것입니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layers** | `LAYERS` | [vector: any] [list] | GeoPackage 로 가져올 (벡터) 레이어입니다. 래스터 레이어는 지원하지 않습니다. 래스터 레이어를 추가하면, `QgsProcessingException` 클래스를 던질 것입니다. |
| **Overwrite existing GeoPackage** | `OVERWRITE` | [boolean]<br>기본값: False | 지정한 GeoPackage 가 존재하는 경우, 이 옵션을 `True` 로 설정하면 레이어를 추가하기 전에 기존 GeoPackage 를 삭제하고 새 GeoPackage 를 생성할 것입니다. 이 옵션의 값을 `False` 로 설정하면, 레이어를 추가할 것입니다. |

다음 페이지에 계속

표 24.102 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Save layer styles into GeoPackage** | SAVE_STYLES | [boolean] 기본값: True | Save the layer styles. |
| **Save only selected features** | SELECTED_FEATURES | [boolean] 기본값: False | 레이어에 선택 집합이 있는 경우, 이 옵션을 True 로 설정하면 선택한 피처들만 저장하게 될 것입니다. 레이어에 선택 집합이 없다면 모든 피처를 저장할 것입니다. |
| **Export related layers following relations defined in the project** | EXPORT_RELATED_LA | [boolean] 기본값: False | 입력 레이어가 프로젝트에 설정된관계 를 가지고 있는 경우 이 옵션을 True 로 설정하면 관계된 레이어 (들) 도 내보내게 될 것입니다. 레이어에 선택한 피처가 있다면, 관계된 레이어가 동시에 입력 레이어가 아닌 이상 선택한 피처와 관계된 피처만 내보낼 것입니다. |
| **Destination GeoPackage** | OUTPUT | [file] 기본값: [Save to temporary file] | GeoPackage 파일을 저장할 위치를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Layers within new package** | OUTPUT_LAYERS | [string] [list] | GeoPackage 에 추가된 레이어 목록입니다. |

### 파이썬 코드

**알고리즘 ID**: native:package

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### PostgreSQL 에서 SQL 실행 및 불러오기

QGIS 와 연결된 PostgreSQL 데이터베이스 상에서 SQL 데이터베이스 쿼리 작업을 허용하고 그 결과를 불러올 수 있습니다. 이 알고리즘은 새 레이어를 생성하지 **않을 것입니다**. 이 알고리즘은 레이어 자체에 대해 쿼리를 실행하도록 설계되었습니다.

예시

1. 기존 필드의 모든 값을 고정 값으로 설정하십시오. SQL 쿼리 문자열은 다음과 같습니다:

   ```
   UPDATE your_table SET field_to_update=20;
   ```

   이 예시는 `your_table` 테이블의 `field_to_update` 필드의 값을 모두 `20` 으로 설정할 것입니다.

2. 새 `area` 열을 생성하고 `ST_AREA` PostGIS 함수를 통해 각 피처의 면적을 계산하십시오.

   ```
   -- Create the new column "area" on the table your_table"
   ALTER TABLE your_table ADD COLUMN area double precision;
   -- Update the "area" column and calculate the area of each feature:
   UPDATE your_table SET area=ST_AREA(geom);
   ```

**더 보기**

*PostgreSQL 에서 SQL 실행*, *SQL 실행하기*, *SpatiaLite 에서 SQL 실행*

#### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Database (connection name)** | `DATABASE` | [string] | 데이터베이스 연결의 이름. (데이터베이스명이 아닙니다) 기존 연결은 콤보박스에 표시될 것입니다. |
| **SQL query** | `SQL` | [string] | SQL 쿼리를 정의합니다. 예: `'UPDATE my_table SET field=10'` |
| **Unique ID field name** | `ID_FIELD` | [string]<br>기본값: id | 기본 키 필드를 (산출 테이블의 열) 설정합니다. |
| **Geometry field name**<br>선택적 | `GEOMETRY_FIELD` | [string]<br>기본값: 'geom' | 도형 열의 이름을 (산출 테이블의 열) 설정합니다. |

#### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **SQL layer** | `OUTPUT` | [vector: any] | QGIS 로 불러올 산출 벡터 레이어입니다. |

#### 파이썬 코드

**알고리즘 ID**: `qgis:postgisexecuteandloadsql`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### PostgreSQL 에서 SQL 실행

QGIS 와 연결된 PostgreSQL 데이터베이스 상에서 SQL 데이터베이스 쿼리 작업을 허용합니다. 이 알고리즘은 새 레이어를 생성하지 **않을 것입니다**. 이 알고리즘은 레이어 자체에 대해 쿼리를 실행하도록 설계되었습니다.

**예시**

1. 기존 필드의 모든 값을 고정 값으로 설정하십시오. SQL 쿼리 문자열은 다음과 같습니다:

   ```
   UPDATE your_table SET field_to_update=20;
   ```

   이 예시는 `your_table` 테이블의 `field_to_update` 필드의 값을 모두 `20` 으로 설정할 것입니다.

2. 새 `area` 열을 생성하고 `ST_AREA` PostGIS 함수를 통해 각 피처의 면적을 계산하십시오.

   ```
   -- Create the new column "area" on the table your_table"
   ALTER TABLE your_table ADD COLUMN area double precision;
   -- Update the "area" column and calculate the area of each feature:
   UPDATE your_table SET area=ST_AREA(geom);
   ```

**더 보기**

*PostgreSQL* 에서 *SQL* 실행 및 불러오기, *SQL* 실행하기, *SpatiaLite* 에서 *SQL* 실행

#### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Database (connection name)** | `DATABASE` | [string] | 데이터베이스 연결의 이름. (데이터베이스명이 아닙니다) 기존 연결은 콤보박스에 표시될 것입니다. |
| **SQL query** | `SQL` | [string] | SQL 쿼리를 정의합니다. 예: `'UPDATE my_table SET field=10'` |

#### 산출물

아무것도 산출하지 않습니다. SQL 쿼리는 제자리에서 (in-place) 실행됩니다.

#### 파이썬 코드

**알고리즘 ID**: `native:postgisexecutesql`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### SpatiaLite 에서 SQL 실행

SpatiaLite 데이터베이스 상에서 SQL 데이터베이스 쿼리 작업을 허용합니다. 어떤 데이터베이스인지는 입력 레이어 또는 파일이 결정합니다. 이 알고리즘은 새 레이어를 생성하지 **않을 것입니다**. 이 알고리즘은 레이어 자체에 대해 쿼리를 실행하도록 설계되었습니다.

**더 보기**

*PostgreSQL 에서 SQL 실행*, *SQL 실행하기*

For some SQL query examples see *PostgreSQL Query Examples*.

#### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Database layer (or file)** | `DATABASE` | [file] | 연결할 SQLite/SpatiaLite 데이터베이스 파일 |
| **SQL query** | `SQL` | [string]<br>기본값: '' | SQL 쿼리를 정의합니다. 예: `'UPDATE my_table SET field=10'` |

#### 산출물

아무것도 산출하지 않습니다. SQL 쿼리는 제자리에서 (in-place) 실행됩니다.

#### 파이썬 코드

**알고리즘 ID**: `native:spatialiteexecutesql`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### SpatialLite 에서 SQL 실행 (등록된 DB)

QGIS 와 연결된 SpatiaLite 데이터베이스 상에서 SQL 데이터베이스 쿼리 작업을 허용합니다. 이 알고리즘은 새 레이어를 생성하지 **않을 것입니다**. 이 알고리즘은 레이어 자체에 대해 쿼리를 실행하도록 설계되었습니다.

**더 보기**

*PostgreSQL 에서 SQL 실행*, *SQL 실행하기*

For some SQL query examples see *PostgreSQL Query Examples*.

#### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Database** | `DATABASE` | [enumeration] 기본값: 설정하지 않음 | 현재 세션에 연결된 SQLite/SpatiaLite 데이터베이스를 선택하십시오. |
| **SQL query** | `SQL` | [string] 기본값: '' | SQL 쿼리를 정의합니다. 예: `'UPDATE my_table SET field=10'` |

#### 산출물

아무것도 산출하지 않습니다. SQL 쿼리는 제자리에서 (in-place) 실행됩니다.

#### 파이썬 코드

**알고리즘 ID**: `native:spatialiteexecutesqlregistered`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 24.1.5 파일 도구

#### Download file via HTTP(S)

Downloads a URL to the file system with an HTTP(S) GET or POST request.

##### 파라미터

##### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **URL** | `URL` | [string] | 다운로드할 파일의 URL |
| **File destination** | `OUTPUT` | [string]<br>기본값: `[Save to temporary file]` | Specification of the file destination. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

##### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Method** | `METHOD` | [enumeration]<br>기본값: 0 | 요청을 위해 사용할 HTTP 메소드입니다. 다음 옵션을 사용할 수 있습니다:<br>• 0 —GET<br>• 1 —POST |
| **Data**<br>선택적 | `DATA` | [string] | 요청이 POST 인 경우 본문 (body) 에 추가할 데이터입니다. |

##### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **File destination** | `OUTPUT` | [string] | 다운로드한 파일을 저장한 위치 |

##### 파이썬 코드

**Algorithm ID**: `native:filedownloader`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### HTTP(S) POST/GET request

Performs a HTTP(S) POST/GET request and returns the HTTP status code and the reply data. If an error occurs then the error code and the message will be returned. Optionally, the result can be written to a file on the disk. By default the algorithm will warn on errors. Optionally, the algorithm can be set to treat HTTP errors as failures.

#### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **URL or file path** | `URL` | [string] | The URL or path of the file to open. |
| **Method** | `METHOD` | [enumeration]<br>기본값: 0 | 요청을 위해 사용할 HTTP 메소드입니다. 다음 옵션을 사용할 수 있습니다:<br>• 0 —GET<br>• 1 —POST |
| **POST Data**<br>선택적 | `DATA` | [string] | 요청이 POST 인 경우 본문 (body) 에 추가할 데이터입니다. |
| **Authentication**<br>선택적 | `AUTH_CONFIG` | [authconfig]<br>Default: No authentication | An authentication configuration to pass |
| **Consider HTTP errors as failures** | `FAIL_ON_ERROR` | [boolean]<br>기본값: False | If set, the algorithm will fail on encountering an HTTP error. |
| **File destination**<br>선택적 | `OUTPUT` | [string]<br>Default: `[Skip Output]` | The result can be written to a file instead of being returned as a string. Specification of the file destination. *One of*:<br>• 산출물 건너뛰기<br>• 임시 파일로 저장<br>• 파일로 저장… |

#### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **File destination** | `OUTPUT` | [file] | The downloaded file with the returned result |
| **HTTP Status** | `STATUS_CODE` | [string] | The HTTP Status |
| **Network error code** | `ERROR_CODE` | [string] | The error code when it fails (like as well on 404 HTTP Status etc.) |
| **Network error message** | `ERROR_MESSAGE` | [string] | A string containing the error message in case of failure |
| **Result data** | `RESULT_DATA` | [string] | A string containing the result data in case of success |

#### 파이썬 코드

**Algorithm ID**: `native:httprequest`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### Open file or URL

Opens files in their default associated application, or URLs in the user's default web browser.

#### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **URL or file path** | `URL` | [string] | The URL or path of the file to open. |

#### 산출물

이 알고리즘은 아무것도 산출하지 않습니다.

#### 파이썬 코드

**Algorithm ID**: `native:openurl`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 24.1.6 Fix Geometry

### Add missing vertices along borders

Adds the missing vertices along polygons borders, based on an error layer from the *Missing vertices along borders* algorithm.

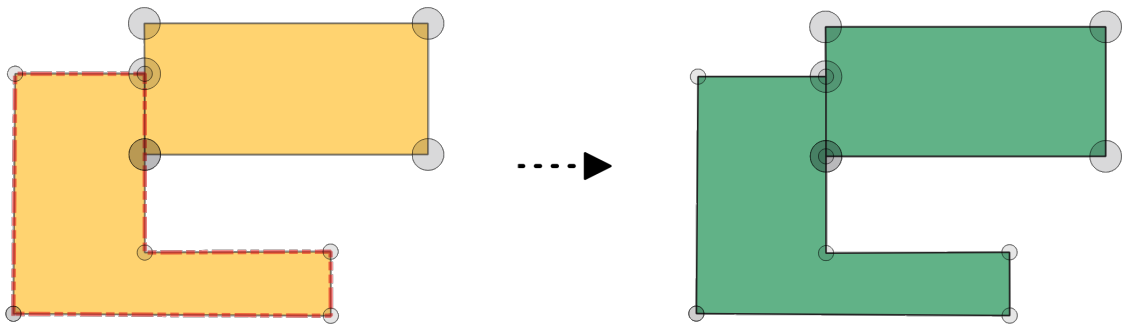


그림 24.16: Fixing missing vertices along polygon borders.

### Parameters

#### Basic parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: polygon] | Layer with the geometries to fix, same as input in the check algorithm. |
| **Error layer** | `ERRORS` | [vector: point] | Point layer with the errors location, generated by the check algorithm. |
| **Field of original feature unique identifier** | `UNIQUE_ID` | [tablefield: any] | Field storing unique values for feature identification in input layer, as set in the check algorithm. |
| **Field of part index** | `PART_IDX` | [tablefield: integer] Default: `gc_partidx` | Field storing the erroneous feature's geometry part number. |
| **Field of ring index** | `RING_IDX` | [tablefield: integer] Default: `gc_ringidx` | Field storing the erroneous feature's geometry ring number. |
| **Field of vertex index** | `VERTEX_IDX` | [tablefield: integer] Default: `gc_vertidx` | Field storing the erroneous feature's geometry vertex number. |
| **Border vertices fixed layer** | `OUTPUT` | [vector: polygon] Default: `[Create temporary layer]` | Specification of the output layer containing fixed features. *One of*:<br>• Create Temporary Layer (`TEMPORARY_OUTPUT`)<br>• Save to File…<br>• Save to Geopackage…<br>• Save to Database Table…<br>The file encoding can also be changed here. |

다음 페이지에 계속

표 24.115 – 이전 페이지에서 계속

| Label | Name | Type | Description |
|---|---|---|---|
| **Report layer from fixing border vertices** | `REPORT` | [vector: point] Default: `[Create temporary layer]` | Specification of the output layer containing the fixes location. *One of*:<br>• Create Temporary Layer (`TEMPORARY_OUTPUT`)<br>• Save to File…<br>• Save to Geopackage…<br>• Save to Database Table…<br>The file encoding can also be changed here. |

### Advanced parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Tolerance** | `TOLERANCE` | [numeric: integer] Default: 8 | Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than $10^{-n}$ apart (in map units) are considered to be merged. |

### Outputs

| Label | Name | Type | Description |
|---|---|---|---|
| **Report layer from fixing border vertices** | `REPORT` | [vector: point] | Output point layer representing the error locations and fix applied. The output layer contains the following fields:<br>• `gc_layerid`: the ID of the input layer.<br>• `gc_layername`: the name of the input layer.<br>• `gc_partidx`<br>• `gc_ringidx`<br>• `gc_vertidx`<br>• `gc_errorx`: the x coordinate of the missing vertex.<br>• `gc_errory`: the y coordinate of the missing vertex.<br>• `gc_error`<br>• `report`: a text field describing the fix applied.<br>• `error_fixed`: a boolean field indicating whether the error was fixed.<br>• `UNIQUE_ID` field: the unique ID of the input feature that has a missing vertex. |

다음 페이지에 계속

표 24.117 – 이전 페이지에서 계속

| Label | Name | Type | Description |
|---|---|---|---|
| **Border vertices fixed layer** | `OUTPUT` | [vector: polygon] | Output layer with the geometry fix applied to the input features. The layer contains the same fields as in the input layer. |

### Python code

**Algorithm ID**: `native:fixgeometrymissingvertex`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 콘솔에서 공간 처리 알고리즘 사용 for details on how to run processing algorithms from the Python console.

## Convert to strictly multipart

Converts multipart geometries that consists of only one geometry into singlepart geometries, based on an error layer from the *Strictly multipart* algorithm. Layer geometry type will not be changed and will remain multipart.

**더 보기**

다중 부분으로 조성하기

### Parameters

### Basic parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: line, polygon] | Layer with the geometries to fix, same as input in the check algorithm. |
| **Error layer** | `ERRORS` | [vector: point] | Point layer with the errors location, generated by the check algorithm. |
| **Field of original feature unique identifier** | `UNIQUE_ID` | [tablefield: any] | Field storing unique values for feature identification in input layer, as set in the check algorithm. |
| **Strictly-multipart layer** | `OUTPUT` | [vector: same as input]<br>Default: `[Create temporary layer]` | Specification of the output layer containing fixed features. *One of*:<br>• Create Temporary Layer (`TEMPORARY_OUTPUT`)<br>• Save to File…<br>• Save to Geopackage…<br>• Save to Database Table…<br>The file encoding can also be changed here. |

다음 페이지에 계속

표 24.118 – 이전 페이지에서 계속

| Label | Name | Type | Description |
|---|---|---|---|
| **Report layer from fixing multipart** | `REPORT` | [vector: point]<br>Default: `[Create temporary layer]` | Specification of the output layer containing the centroids of the fixed geometries. *One of*:<br>• Create Temporary Layer (`TEMPORARY_OUTPUT`)<br>• Save to File…<br>• Save to Geopackage…<br>• Save to Database Table…<br>The file encoding can also be changed here. |

### Advanced parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Tolerance** | `TOLERANCE` | [numeric: integer]<br>Default: 8 | Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than $10^{-n}$ apart (in map units) are considered to be merged. |

### Outputs

| Label | Name | Type | Description |
|---|---|---|---|
| **Report layer from fixing multipart** | `REPORT` | [vector: point] | Output point layer representing the error locations and fix applied. The output layer contains the following fields:<br>• `gc_layerid`: the ID of the input layer.<br>• `gc_layername`: the name of the input layer.<br>• `gc_partidx`<br>• `gc_ringidx`<br>• `gc_vertidx`<br>• `gc_errorx`: the x coordinate of the centroid of the multipart geometry that has only one part.<br>• `gc_errory`: the y coordinate of the centroid of the multipart geometry that has only one part.<br>• `gc_error`<br>• `report`: a text field describing the fix applied.<br>• `error_fixed`: a boolean field indicating whether the error was fixed.<br>• `UNIQUE_ID` field: the unique ID of the input feature that is multipart but has only one part. |
| **Strictly-multipart layer** | `OUTPUT` | [vector: same as input] | Output layer with the geometry fix applied to the input features. The layer contains the same fields as in the input layer. |

### Python code

**Algorithm ID**: `native:fixgeometrymultipart`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 콘솔에서 공간 처리 알고리즘 사용 for details on how to run processing algorithms from the Python console.

### Delete duplicated vertices

Deletes duplicated vertices from the input geometries, based on errors reported by the *Duplicated vertices* algorithm.

**더 보기**

복제 꼭짓점 제거하기

### Parameters

### Basic parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: line, polygon] | Layer with the geometries to fix, same as input in the check algorithm. |
| **Error layer** | `ERRORS` | [vector: point] | Point layer with the errors location, generated by the check algorithm. |
| **Field of original feature unique identifier** | `UNIQUE_ID` | [tablefield: any] | Field storing unique values for feature identification in input layer, as set in the check algorithm. |
| **Field of part index** | `PART_IDX` | [tablefield: integer]<br>Default: `gc_partidx` | Field storing the erroneous feature's geometry part number. |
| **Field of ring index** | `RING_IDX` | [tablefield: integer]<br>Default: `gc_ringidx` | Field storing the erroneous feature's geometry ring number. |
| **Field of vertex index** | `VERTEX_IDX` | [tablefield: integer]<br>Default: `gc_vertidx` | Field storing the erroneous feature's geometry vertex number. |
| **Fixed duplicate vertices layer** | `OUTPUT` | [vector: same as input]<br>Default: `[Create temporary layer]` | Specification of the output layer containing fixed features. *One of*:<br>• Create Temporary Layer (`TEMPORARY_OUTPUT`)<br>• Save to File…<br>• Save to Geopackage…<br>• Save to Database Table…<br>The file encoding can also be changed here. |
| **Report layer from fixing duplicate vertices** | `REPORT` | [vector: point]<br>Default: `[Create temporary layer]` | Specification of the output layer containing the fixes location. *One of*:<br>• Create Temporary Layer (`TEMPORARY_OUTPUT`)<br>• Save to File…<br>• Save to Geopackage…<br>• Save to Database Table…<br>The file encoding can also be changed here. |

### Advanced parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Tolerance** | `TOLERANCE` | [numeric: integer]<br>Default: 8 | Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than $10^{-n}$ apart (in map units) are considered to be merged. |

### Outputs

| Label | Name | Type | Description |
|---|---|---|---|
| **Report layer from fixing duplicate vertices** | `REPORT` | [vector: point] | Output point layer representing the error locations and fix applied. The output layer contains the following fields:<br>• `gc_layerid`: the ID of the input layer.<br>• `gc_layername`: the name of the input layer.<br>• `gc_partidx`: the index of the feature's geometry part containing the duplicate vertex.<br>• `gc_ringidx`: the index of the feature's geometry ring containing the duplicate vertex.<br>• `gc_vertidx`: the index of the duplicate vertex in the feature's geometry ring.<br>• `gc_errorx`: the x coordinate of the duplicate vertex.<br>• `gc_errory`: the y coordinate of the duplicate vertex.<br>• `gc_error`<br>• `report`: a text field describing the fix applied.<br>• `error_fixed`: a boolean field indicating whether the error was fixed.<br>• `UNIQUE_ID` field: the unique ID of the input feature that has duplicate vertices. |
| **Fixed duplicate vertices layer** | `OUTPUT` | [vector: same as input] | Output layer with the geometry fix applied to the input features. The layer contains the same fields as in the input layer. |

### Python code

**Algorithm ID**: `native:fixgeometryduplicatenodes`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 콘솔에서 공간 처리 알고리즘 사용 for details on how to run processing algorithms from the Python console.

### Delete features

Deletes error features based on an error layer from some check geometry algorithms, such as:

- *Features inside polygon*
- *Degenerate polygons*
- *Small segments*
- *Duplicated geometries*

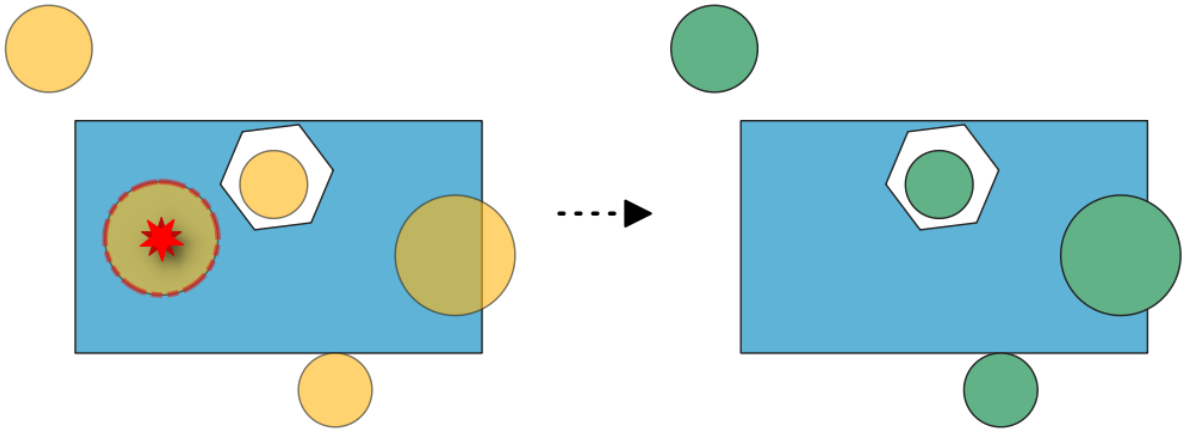


그림 24.17: Fixing errors from the "*Features inside polygon*" algorithm check.

**더 보기**

복제 도형 삭제하기, 도형 삭제

### Parameters

### Basic parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | Layer with the geometries to fix, same as input in the check algorithm. |
| **Error layer** | `ERRORS` | [vector: point] | Point layer with the errors location, generated by the check algorithm. |
| **Field of original feature unique identifier** | `UNIQUE_ID` | [tablefield: any] | Field storing unique values for feature identification in input layer, as set in the check algorithm. |

다음 페이지에 계속

표 24.124 – 이전 페이지에서 계속

| Label | Name | Type | Description |
|---|---|---|---|
| **Cleaned layer** | `OUTPUT` | [vector: same as input]<br>Default: `[Create temporary layer]` | Specification of the output layer containing fixed features. *One of*:<br>• Create Temporary Layer (`TEMPORARY_OUTPUT`)<br>• Save to File…<br>• Save to Geopackage…<br>• Save to Database Table…<br>The file encoding can also be changed here. |
| **Report layer from deleting features** | `REPORT` | [vector: point]<br>Default: `[Create temporary layer]` | Specification of the output layer containing the fixes location. *One of*:<br>• Create Temporary Layer (`TEMPORARY_OUTPUT`)<br>• Save to File…<br>• Save to Geopackage…<br>• Save to Database Table…<br>The file encoding can also be changed here. |

### Outputs

| Label | Name | Type | Description |
|---|---|---|---|
| **Report layer from deleting features** | `REPORT` | [vector: point] | Output point layer representing the error locations and fix applied. The output layer contains the following fields:<br>• `gc_layerid`: the ID of the input layer.<br>• `gc_layername`: the name of the input layer.<br>• `gc_partidx`<br>• `gc_ringidx`<br>• `gc_vertidx`<br>• `gc_errorx`: the x coordinate of the centroid of the erroneous geometry.<br>• `gc_errory`: the y coordinate of the centroid of the erroneous geometry.<br>• `gc_error`: the index of the feature where the error belongs.<br>• `report`: a text field describing the fix applied.<br>• `error_fixed`: a boolean field indicating whether the error was fixed.<br>• `UNIQUE_ID` field: the unique ID of the input feature with the error. |
| **Cleaned layer** | `OUTPUT` | [vector: same as input] | Output layer with features removed based on detected errors. The layer contains the same fields as in the input layer. |

### Python code

**Algorithm ID**: `native:fixgeometrydeletefeatures`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 콘솔에서 공간 처리 알고리즘 사용 for details on how to run processing algorithms from the Python console.

### Delete overlaps

Deletes overlapping areas based on an error layer from the *Overlaps* algorithm.

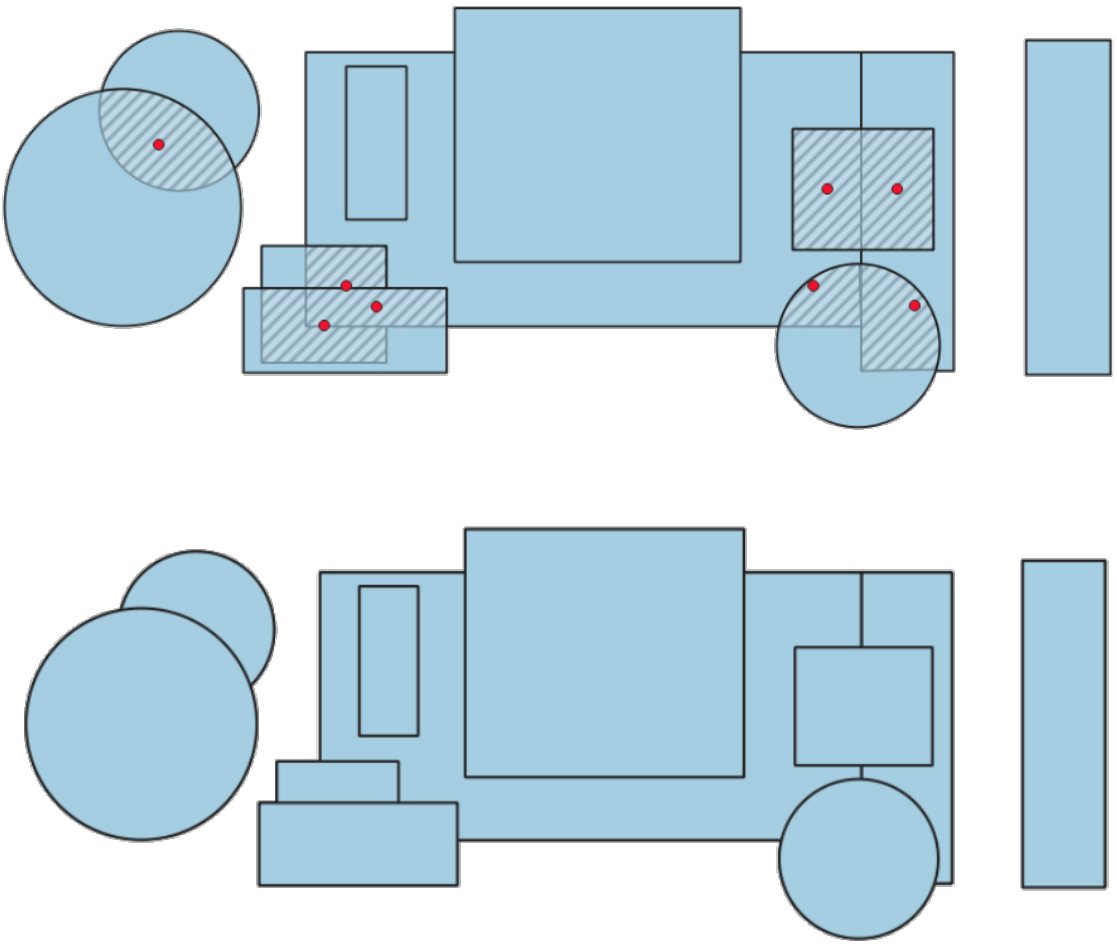


그림 24.18: Deleting overlapping areas.

### Parameters

### Basic parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: polygon] | Layer with the geometries to fix, same as input in the check algorithm. |
| **Error layer** | `ERRORS` | [vector: point] | Point layer with the errors location, generated by the check algorithm. |
| **Field of original feature unique identifier** | `UNIQUE_ID` | [tablefield: any] | Field storing unique values for feature identification in input layer, as set in the check algorithm. |
| **Field of overlap feature unique identifier** | `OVERLAP_FEATURE_UNI` | [tablefield: any] | Field storing unique values for the overlapping feature identification in input layer, as set in the check algorithm. |
| **Field of error value** | `ERROR_VALUE_ID` | [tablefield: any] | Field storing the error value for each feature, as set in the check algorithm. |
| **No-overlap layer** | `OUTPUT` | [vector: polygon]<br>Default: `[Create temporary layer]` | Specification of the output layer containing fixed features. *One of*:<br>• Create Temporary Layer (`TEMPORARY_OUTPUT`)<br>• Save to File…<br>• Save to Geopackage…<br>• Save to Database Table…<br>The file encoding can also be changed here. |
| **Report layer from fixing overlaps** | `REPORT` | [vector: point]<br>Default: `[Create temporary layer]` | Specification of the output layer containing the fixes location. *One of*:<br>• Create Temporary Layer (`TEMPORARY_OUTPUT`)<br>• Save to File…<br>• Save to Geopackage…<br>• Save to Database Table…<br>The file encoding can also be changed here. |

### Advanced parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Tolerance** | `TOLERANCE` | [numeric: integer]<br>Default: 8 | Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than $10^{-n}$ apart (in map units) are considered to be merged. |

### Outputs

| Label | Name | Type | Description |
|---|---|---|---|
| **Report layer from fixing overlaps** | `REPORT` | [vector: point] | Output point layer representing the error locations and fix applied. The output layer contains the following fields:<br>• `gc_layerid`: the ID of the input layer.<br>• `gc_layername`: the name of the input layer.<br>• `gc_errorx`: the x coordinate of the centroid of the overlapping area.<br>• `gc_errory`: the y coordinate of the centroid of the overlapping area.<br>• `gc_error`: the area of the overlapping geometry.<br>• `UNIQUE_ID` field: the unique ID of the unique ID of an overlapped input feature.<br>• `gc_overlap_feature_{unique_id}`: the `UNIQUE_ID` field value for the other overlapping feature.<br>• `report`: a text field describing the fix applied or justifying the failure. Possible values are:<br>– Remove overlapping area from neighboring polygon with shortest shared edge<br>– Error is obsolete<br>• `error_fixed`: a boolean field indicating whether the error was fixed. |
| **No-overlap layer** | `OUTPUT` | [vector: polygon] | Output layer with input features edited. Overlapping areas reported as errors are removed. The layer contains the same fields as in the input layer. |

### Python code

**Algorithm ID**: `native:fixgeometryoverlap`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 콘솔에서 공간 처리 알고리즘 사용 for details on how to run processing algorithms from the Python console.

### Delete small angles

Deletes vertices based on an error layer from the *Small angles* algorithm. When deletion of a vertex results in a duplicate vertex (when a spike vertex is deleted), the duplicate vertex is deleted to keep a single vertex and preserve topology.

> **주의**
>
> This algorithm removes the vertex at the reported small angles, generating new segments that may form a new small angle.

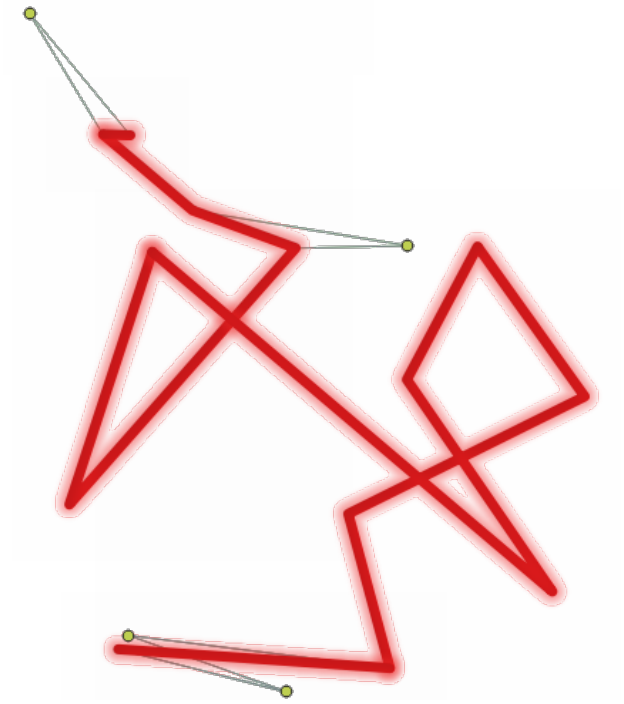


그림 24.19: Fixing reported errors on a line feature for angles lower than 15°.

### Parameters

### Basic parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: line, polygon] | Layer with the geometries to fix, same as input in the check algorithm |
| **Error layer** | `ERRORS` | [vector: point] | Point layer with the errors location, generated by the check algorithm |
| **Field of original feature unique identifier** | `UNIQUE_ID` | [tablefield: any] | Field storing unique values for feature identification in input layer, as set in the check algorithm |

다음 페이지에 계속

표 24.129 – 이전 페이지에서 계속

| Label | Name | Type | Description |
|---|---|---|---|
| **Field of part index** | `PART_IDX` | [tablefield: integer]<br>Default: `gc_partidx` | Field storing the erroneous feature's geometry part number. |
| **Field of ring index** | `RING_IDX` | [tablefield: integer]<br>Default: `gc_ringidx` | Field storing the erroneous feature's geometry ring number. |
| **Field of vertex index** | `VERTEX_IDX` | [tablefield: integer]<br>Default: `gc_vertidx` | Field storing the erroneous feature's geometry vertex number. |
| **Small angle fixed layer** | `OUTPUT` | [vector: same as input]<br>Default: `[Create temporary layer]` | Specification of the output layer containing fixed features. *One of*:<br>• Create Temporary Layer (`TEMPORARY_OUTPUT`)<br>• Save to File…<br>• Save to Geopackage…<br>• Save to Database Table…<br>The file encoding can also be changed here. |
| **Report layer from fixing small angles** | `REPORT` | [vector: point]<br>Default: `[Create temporary layer]` | Specification of the output layer containing the fixes location. *One of*:<br>• Create Temporary Layer (`TEMPORARY_OUTPUT`)<br>• Save to File…<br>• Save to Geopackage…<br>• Save to Database Table…<br>The file encoding can also be changed here. |

### Advanced parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Tolerance** | `TOLERANCE` | [numeric: integer]<br>Default: 8 | Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than $10^{-n}$ apart (in map units) are considered to be merged. |

### Outputs

| Label | Name | Type | Description |
|---|---|---|---|
| **Report layer from fixing small angles** | `REPORT` | [vector: point] | Output point layer representing the error locations and fix applied. The output layer contains the following fields:<br>• `gc_layerid`: the ID of the input layer.<br>• `gc_layername`: the name of the input layer.<br>• `gc_partidx`: the index of the feature's geometry part containing the small angle.<br>• `gc_ringidx`: the index of the feature's geometry ring containing the small angle.<br>• `gc_vertidx`: the index of the vertex with the small angle in the feature's geometry ring.<br>• `gc_errorx`: the x coordinate of the vertex with the small angle.<br>• `gc_errory`: the y coordinate of the vertex with the small angle.<br>• `gc_error`: the error angle value.<br>• `UNIQUE_ID` field: the unique ID of the input feature with the small angle.<br>• `report`: a text field describing the fix applied or justifying the failure.<br>• `error_fixed`: a boolean field indicating whether the error was fixed. |
| **Small angle fixed layer** | `OUTPUT` | [same as input] | Output layer with the geometry fix applied to the input features The layer contains the same fields as in the input layer. |

### Python code

**Algorithm ID**: `native:fixgeometryangle`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 콘솔에서 공간 처리 알고리즘 사용 for details on how to run processing algorithms from the Python console.

### Fill gaps

Fills the gaps based on a gap and neighbors layer from the *Small gaps* algorithm. Three different fixing methods are available.

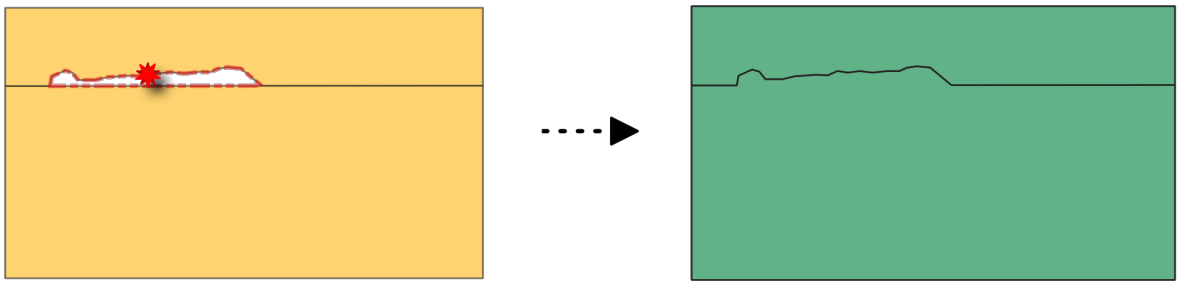


그림 24.20: Fixing gap between polygon features using the `Add to largest neighbouring area` method.

### Parameters

### Basic parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: polygon] | Layer with the geometries to fix, same as input in the check algorithm. |
| **Neighbors layer** | `NEIGHBORS` | [vector: table] | Relational table with the unique ID of the gap and its neighbors in input layer, generated by the check algorithm. |
| **Gaps layer** | `GAPS` | [vector: polygon] | Polygon layer with the gaps geometries, generated by the check algorithm. |
| **Method** | `METHOD` | [enumeration] Default: 0 | Method to apply to the gap fixing. Possible values are:<br>• 0 —`Add to longest shared edge`: adds the gap to the neighboring feature with the longest shared edge.<br>• 1 —`Create new feature`: creates a new feature for each gap.<br>• 2 —`Add to largest neighbouring area`: adds the gap to the neighboring feature with the largest area. |
| **Field of original feature unique identifier** | `UNIQUE_ID` | [tablefield: any] Default: gc_errorid | Field storing unique values for feature identification in input layer, as set in the check algorithm. |
| **Field of error id** | `ERROR_ID_IDX` | [tablefield: any] | Field storing the error ID for each gap, as output by the check algorithm. |

다음 페이지에 계속

표 24.132 – 이전 페이지에서 계속

| Label | Name | Type | Description |
|---|---|---|---|
| **Gaps-filled layer** | `OUTPUT` | [vector: polygon]<br>Default: `[Create temporary layer]` | Specification of the output layer containing fixed features. *One of*:<br>• Create Temporary Layer (`TEMPORARY_OUTPUT`)<br>• Save to File…<br>• Save to Geopackage…<br>• Save to Database Table…<br>The file encoding can also be changed here. |
| **Report layer from fixing gaps** | `REPORT` | [vector: point]<br>Default: `[Create temporary layer]` | Specification of the output layer containing the fixes location. *One of*:<br>• Create Temporary Layer (`TEMPORARY_OUTPUT`)<br>• Save to File…<br>• Save to Geopackage…<br>• Save to Database Table…<br>The file encoding can also be changed here. |

### Advanced parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Tolerance** | `TOLERANCE` | [numeric: integer]<br>Default: 8 | Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than $10^{-n}$ apart (in map units) are considered to be merged. |

### Outputs

| Label | Name | Type | Description |
|---|---|---|---|
| **Report layer from fixing gaps** | `REPORT` | [vector: point] | Output point layer representing the error locations and fix applied. The output layer contains the following fields:<br>• `gc_layerid`: the ID of the input layer.<br>• `gc_layername`: the name of the input layer.<br>• `gc_partidx`<br>• `gc_ringidx`<br>• `gc_vertidx`<br>• `gc_errorx`: the x coordinate of the centroid of the gap.<br>• `gc_errory`: the y coordinate of the centroid of the gap.<br>• `gc_error`: the area of the gap.<br>• `gc_errorid`: the ID of the gap.<br>• `report`: a text field mentioning the method used to fix the gap.<br>• `error_fixed`: a boolean field indicating whether the error was fixed. |
| **Gaps-filled layer** | `OUTPUT` | [vector: polygon] | Output layer of polygons without gaps. The layer contains the same fields as in the input layer. |

### Python code

**Algorithm ID**: `native:fixgeometrygap`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 콘솔에서 공간 처리 알고리즘 사용 for details on how to run processing algorithms from the Python console.

### Fill holes

Deletes holes in polygon geometries based on an error layer from the *Holes* algorithm.

**더 보기**

구멍 삭제하기

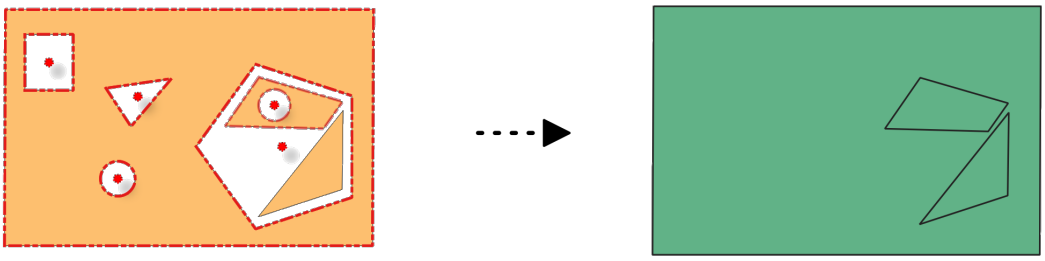


그림 24.21: Deleting holes in a polygon feature.

### Parameters

### Basic parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: line, polygon] | Layer with the geometries to fix, same as input in the check algorithm |
| **Error layer** | `ERRORS` | [vector: point] | Point layer with the errors location, generated by the check algorithm |
| **Field of original feature unique identifier** | `UNIQUE_ID` | [tablefield: any] | Field storing unique values for feature identification in input layer, as set in the check algorithm |
| **Field of part index** | `PART_IDX` | [tablefield: integer] Default: `gc_partidx` | Field storing the erroneous feature's geometry part number. |
| **Field of ring index** | `RING_IDX` | [tablefield: integer] Default: `gc_ringidx` | Field storing the erroneous feature's geometry ring number. |
| **Field of vertex index** | `VERTEX_IDX` | [tablefield: integer] Default: `gc_vertidx` | Field storing the erroneous feature's geometry vertex number. |
| **Holes-filled layer** | `OUTPUT` | [vector: polygon] Default: `[Create temporary layer]` | Specification of the output layer containing fixed features. *One of*:<br>• Create Temporary Layer (`TEMPORARY_OUTPUT`)<br>• Save to File…<br>• Save to Geopackage…<br>• Save to Database Table…<br>The file encoding can also be changed here. |

다음 페이지에 계속

표 24.135 – 이전 페이지에서 계속

| Label | Name | Type | Description |
|---|---|---|---|
| **Report layer from fixing holes** | `REPORT` | [vector: point] Default: `[Create temporary layer]` | Specification of the output layer containing the fixes location. *One of*:<br>• Create Temporary Layer (`TEMPORARY_OUTPUT`)<br>• Save to File…<br>• Save to Geopackage…<br>• Save to Database Table…<br>The file encoding can also be changed here. |

### Advanced parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Tolerance** | `TOLERANCE` | [numeric: integer] Default: 8 | Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than $10^{-n}$ apart (in map units) are considered to be merged. |

### Outputs

| Label | Name | Type | Description |
|---|---|---|---|
| **Report layer from fixing holes** | `REPORT` | [vector: point] | Output point layer representing the error locations and fix applied. The output layer contains the following fields:<br>• `gc_layerid`: the ID of the input layer.<br>• `gc_layername`: the name of the input layer.<br>• `gc_partidx`: the index of the feature's geometry part containing the hole.<br>• `gc_ringidx`: the index of the feature's geometry ring containing the hole.<br>• `gc_vertidx`<br>• `gc_errorx`: the x coordinate of the centroid of the hole.<br>• `gc_errory`: the y coordinate of the centroid of the hole.<br>• `gc_error`<br>• `UNIQUE_ID` field: the unique ID of the input feature that has a hole.<br>• `report`: a text field describing the fix applied or justifying the failure.<br>• `error_fixed`: a boolean field indicating whether the error was fixed. |
| **Holes-filled layer** | `OUTPUT` | [vector: polygon] | Output layer of polygons without holes. The layer contains the same fields as in the input layer. |

### Python code

**Algorithm ID**: `native:fixgeometryhole`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 콘솔에서 공간 처리 알고리즘 사용 for details on how to run processing algorithms from the Python console.

### Fix small polygons

Merges neighboring polygons according to the chosen method, based on an error layer from the *Small polygons* or *Sliver polygons* algorithm.

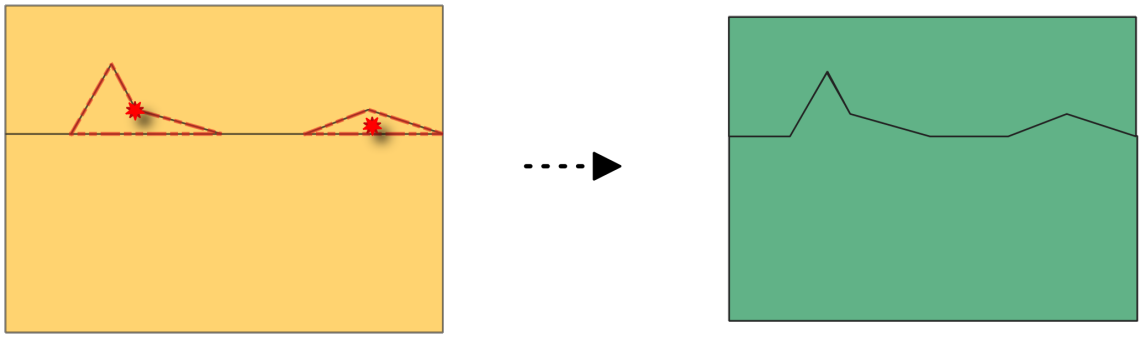


그림 24.22: Before and after fixing small polygons (method: merge with neighboring polygon with largest area).

### Parameters

### Basic parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: polygon] | Layer with the geometries to fix, same as input in the check algorithm. |
| **Error layer** | `ERRORS` | [vector: point] | Point layer with the errors location, generated by the check algorithm. |

다음 페이지에 계속

표 24.138 – 이전 페이지에서 계속

| Label | Name | Type | Description |
|---|---|---|---|
| **Method** | METHOD | [enumeration] Default: 0 | Method to apply to the small polygon geometries:<br>• 0 —Merge with neighboring polygon with longest shared edge: merges the small polygon geometry with the neighboring polygon that has the longest shared edge.<br>• 1 —Merge with neighboring polygon with largest area: merges the small polygon geometry with the neighboring polygon that has the largest area.<br>• 2 —Merge with neighboring polygon with identical attribute value, if any, or leave as is: merges the small polygon geometry with a neighboring polygon that has the same attribute value, if such a polygon exists. |
| **Field to consider when merging polygons with the identical attribute method**<br>Optional | MERGE_ATTRIBUTE | [tablefield: any] | Field in the input layer to consider when merging polygons with the identical attribute method. |
| **Field of original feature unique identifier** | UNIQUE_ID | [tablefield: any] | Field storing unique values for feature identification in input layer, as set in the check algorithm. |
| **Field of part index** | PART_IDX | [tablefield: integer] Default: gc_partidx | Field storing the erroneous feature's geometry part number. |
| **Field of ring index** | RING_IDX | [tablefield: integer] Default: gc_ringidx | Field storing the erroneous feature's geometry ring number. |
| **Field of vertex index** | VERTEX_IDX | [tablefield: integer] Default: gc_vertidx | Field storing the erroneous feature's geometry vertex number. |
| **Small polygons merged layer** | OUTPUT | [vector: polygon] Default: [Create temporary layer] | Specification of the output layer containing fixed features. *One of*:<br>• Create Temporary Layer (TEMPORARY_OUTPUT)<br>• Save to File…<br>• Save to Geopackage…<br>• Save to Database Table…<br>The file encoding can also be changed here. |

다음 페이지에 계속

표 24.138 – 이전 페이지에서 계속

| Label | Name | Type | Description |
|---|---|---|---|
| **Report layer from merging small polygons** | `REPORT` | [vector: point] Default: `[Create temporary layer]` | Specification of the output layer containing the fixes location. *One of*:<br>• Create Temporary Layer (`TEMPORARY_OUTPUT`)<br>• Save to File…<br>• Save to Geopackage…<br>• Save to Database Table…<br>The file encoding can also be changed here. |

### Advanced parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Tolerance** | `TOLERANCE` | [numeric: integer] Default: 8 | Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than $10^{-n}$ apart (in map units) are considered to be merged. |

### Outputs

| Label | Name | Type | Description |
|---|---|---|---|
| **Report layer from merging small polygons** | `REPORT` | [vector: point] | Output point layer representing the error locations and fix applied. The output layer contains the following fields:<br>• `gc_layerid`: the ID of the input layer.<br>• `gc_layername`: the name of the input layer.<br>• `gc_partidx`<br>• `gc_ringidx`<br>• `gc_vertidx`<br>• `gc_errorx`: the x coordinate of the centroid of the small polygon.<br>• `gc_errory`: the y coordinate of the centroid of the small polygon.<br>• `gc_error`: the area of the small polygon.<br>• `UNIQUE_ID` field: the unique ID of the input feature that is a small polygon.<br>• `report`: a text field describing the fix applied or justifying the failure.<br>• `error_fixed`: a boolean field indicating whether the error was fixed. |

다음 페이지에 계속

표 24.140 – 이전 페이지에서 계속

| Label | Name | Type | Description |
|---|---|---|---|
| **Small polygons merged layer** | `OUTPUT` | [vector: polygon] | Output layer of polygons with small polygons merged according to the chosen method. The layer contains the same fields as in the input layer. |

### Python code

**Algorithm ID**: `native:fixgeometryarea`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 콘솔에서 공간 처리 알고리즘 사용 for details on how to run processing algorithms from the Python console.

## Split self-intersecting geometries

Splits self-intersecting geometries based on an error layer from the *Self-intersections* algorithm.

참고

This algorithm detects invalid polygon geometries (self-intersections). To ensure these invalid features are not filtered out before processing, open Advanced options next to the *Input layer* and set *Invalid feature filtering* to `Do not Filter (Better Performance)`.

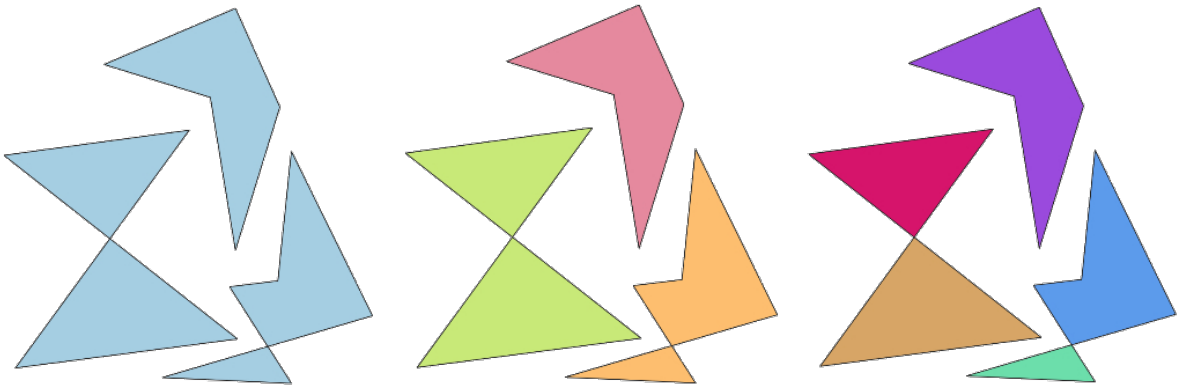


그림 24.23: Fixing self-intersection using various methods: as multi-part (middle), and multiple single-part geometries (right).

### Parameters

### Basic parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: line, polygon] | Layer with the geometries to fix, same as input in the check algorithm |
| **Error layer** | `ERRORS` | [vector: point] | Point layer with the errors location, generated by the check algorithm |
| **Method** | `METHOD` | [enumeration] | Method to apply to the self-intersecting geometries:<br>• 0 – `Split feature into a multi-part feature`: splits the self-intersecting geometry into multiple parts stored in a single multipart feature.<br>• 1 – `Split feature into multiple single-part features`: splits the self-intersecting geometry into multiple parts, each part stored in a separate feature. |
| **Field of original feature unique identifier** | `UNIQUE_ID` | [tablefield: any] | Field storing unique values for feature identification in input layer, as set in the check algorithm |
| **Field of part index** | `PART_IDX` | [tablefield: integer]<br>Default: `gc_partidx` | Field storing the erroneous feature's geometry part number. |
| **Field of ring index** | `RING_IDX` | [tablefield: integer]<br>Default: `gc_ringidx` | Field storing the erroneous feature's geometry ring number. |
| **Field of vertex index** | `VERTEX_IDX` | [tablefield: integer]<br>Default: `gc_vertidx` | Field storing the erroneous feature's geometry vertex number. |
| **Field of segment 1** | `SEGMENT_1` | [tablefield: integer]<br>Default: `gc_segment_1` | Field storing the number of the first segment involved in the erroneous intersection. |
| **Field of segment 2** | `SEGMENT_2` | [tablefield: integer]<br>Default: `gc_segment_2` | Field storing the number of the second segment involved in the erroneous intersection. |
| **Self-intersections fixed layer** | `OUTPUT` | [vector: same as input]<br>Default: `[Create temporary layer]` | Specification of the output layer containing fixed features. *One of*:<br>• Create Temporary Layer (`TEMPORARY_OUTPUT`)<br>• Save to File…<br>• Save to Geopackage…<br>• Save to Database Table…<br>The file encoding can also be changed here. |

다음 페이지에 계속

표 24.141 – 이전 페이지에서 계속

| Label | Name | Type | Description |
|---|---|---|---|
| **Report layer from fixing self-intersections** | `REPORT` | [vector: point]<br>Default: `[Create temporary layer]` | Specification of the output layer containing the fixes location. *One of*:<br>• Create Temporary Layer (`TEMPORARY_OUTPUT`)<br>• Save to File…<br>• Save to Geopackage…<br>• Save to Database Table…<br>The file encoding can also be changed here. |

### Advanced parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Tolerance** | `TOLERANCE` | [numeric: integer]<br>Default: 8 | Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than $10^{-n}$ apart (in map units) are considered to be merged. |

### Outputs

| Label | Name | Type | Description |
|---|---|---|---|
| **Report layer from fixing self-intersections** | `REPORT` | [vector: point] | Output point layer representing the error locations and fix applied. The output layer contains the following fields:<br>• `gc_layerid`: the ID of the input layer.<br>• `gc_layername`: the name of the input layer.<br>• `gc_partidx`: the index of the feature's geometry part where the self-intersection occurs.<br>• `gc_ringidx`: the index of the feature's geometry ring where the self-intersection occurs.<br>• `gc_vertidx`<br>• `gc_errorx`: the x coordinate of the self-intersection.<br>• `gc_errory`: the y coordinate of the self-intersection.<br>• `gc_error`<br>• `gc_segment_1`: the index of the first segment involved in the intersection.<br>• `gc_segment_2`: the index of the second segment involved in the intersection.<br>• `UNIQUE_ID` field: the unique ID of the self-intersecting input feature.<br>• `report`: a text field describing the fix applied.<br>• `error_fixed`: a boolean field indicating whether the error was fixed. |
| **Self-intersections fixed layer** | `OUTPUT` | [vector: same as input] | Output layer with the geometry fix applied to the input features. The layer contains the same fields as in the input layer. |

## Python code

**Algorithm ID**: `native:fixgeometryselfintersection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 콘솔에서 공간 처리 알고리즘 사용 for details on how to run processing algorithms from the Python console.

### 24.1.7 GPS

#### GPS 데이터 변환하기

GPSBabel 도구 를 이용해서 GPS 데이터 파일을 여러 포맷들로부터 GPX 표준 포맷으로 변환합니다.

#### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input file** | `INPUT` | [file] | 변환할 데이터를 담고 있는 파일 |
| **Format** | `FORMAT` | [enumeration] | 변환할 파일의, 이 목록 에 있는 포맷 |
| **Feature type** | `FEATURE_TYPE` | [enumeration]<br>기본값: 0 | 변환할 데이터의 유형<br>• 0 —웨이포인트 (waypoint)<br>• 1 —경로 (route)<br>• 2 —트랙 (track) |
| **Output** | `OUTPUT` | [vector: any]<br>기본값: `[Save to temporary file]` | Specification of the output GPX file. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

#### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output layer** | `OUTPUT_LAYER` | [vector: any] | GPX 표준 포맷으로 된 데이터를 가진 산출 레이어 |

#### 파이썬 코드

**알고리즘 ID**: `native:convertgpsdata`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### GPX 피처 유형 변환하기

GPSBabel 도구 를 이용해서 GPX 피처를 어떤 유형으로부터 다른 유형으로 (예를 들면 모든 웨이포인트 피처를 경로로) 변환합니다.

#### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input file** | `INPUT` | [file] | 변환할 데이터를 담고 있는 파일 |
| **Conversion** | `CONVERSION` | [enumeration]<br>기본값: 0 | 적용할 변환 유형<br>• 0 — 경로를 웨이포인트로<br>• 1 — 트랙을 웨이포인트로<br>• 2 — 웨이포인트를 경로로<br>• 3 — 웨이포인트를 트랙으로 |
| **Output** | `OUTPUT` | [vector: point or line]<br>기본값: `[Save to temporary file]` | Specification of the output file. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

#### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output** | `OUTPUT` | [vector: any] | 변환된 GPX 피처를 가진 산출 레이어 |

#### 파이썬 코드

**알고리즘 ID**: `native:convertgpxfeaturetype`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 기기에서 GPS 데이터를 다운로드하기

GPSBabel 도구 를 이용해서 GPS 기기로부터 데이터를 GPX 표준 포맷으로 다운로드합니다.

#### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Device** | `DEVICE` | [enumeration] 기본값: `Garmin serial` | 데이터를 생성하기 위해 쓰이는 GPS 기기입니다. *GPS 설정* 대화창에서 선언해야만 합니다. |
| **Port** | `PORT` | [enumeration] | 기기가 연결된 포트입니다. OS 에 따라 사용할 수 있는 포트가 달라집니다. |
| **Feature type** | `FEATURE_TYPE` | [enumeration] 기본값: 0 | 변환할 데이터의 유형<br>• 0 —웨이포인트 (waypoint)<br>• 1 —경로 (route)<br>• 2 —트랙 (track) |
| **Output** | `OUTPUT` | [vector: any] 기본값: `[Save to temporary file]` | Specification of the output file. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

#### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output layer** | `OUTPUT_LAYER` | [vector: any] | GPX 표준 포맷으로 된 데이터를 가진 산출 레이어 |

#### 파이썬 코드

**알고리즘 ID**: `native:downloadgpsdata`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

#### 기기에 GPS 데이터를 업로드하기

GPSBabel 도구 를 이용해서 GPX 표준 포맷으로부터 GPS 기기로 데이터를 업로드합니다.

#### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input file** | `INPUT` | [file] | 업로드할 데이터를 담고 있는 `.GPX` 파일 |
| **Device** | `DEVICE` | [enumeration] 기본값: `Garmin serial` | 데이터를 업로드하고자 하는 GPS 기기입니다. *GPS 설정* 대화창에서 선언해야만 합니다. |
| **Port** | `PORT` | [enumeration] | 기기가 연결된 포트입니다. OS 에 따라 사용할 수 있는 포트가 달라집니다. |
| **Feature type** | `FEATURE_TYPE` | [enumeration] 기본값: 0 | 업로드할 데이터의 유형<br>• 0 — 웨이포인트 (waypoint)<br>• 1 — 경로 (route)<br>• 2 — 트랙 (track) |

#### 산출물

산출물이 없습니다. 성공하는 경우 데이터가 기기에 불러와져 있습니다.

#### 파이썬 코드

**알고리즘 ID**: `native:uploadgpsdata`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 24.1.8 보간법

#### 히트 맵 (커널 밀도 추정)

커널 밀도 추정 (kernel density estimation) 을 사용해서 입력 포인트 벡터 레이어의 밀도 (히트 맵 (heat map)) 래스터 레이어를 생성합니다.

어떤 위치에 있는 포인트의 개수를 기반으로 밀도를 계산합니다. 군집 포인트의 개수가 많을수록 더 큰 값이 산출됩니다. 히트 맵을 사용하면 포인트의 핫 스폿 *(hot spot)* 및 군집을 쉽게 식별할 수 있습니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Point layer** | `INPUT` | [vector: point] | 히트 맵에 사용할 포인트 벡터 레이어 |
| **Radius** | `RADIUS` | [numeric: double]<br>기본값: 100.0 | 히트 맵의 맵 단위 검색 반경 (또는 커널 대역폭). 이 반경은 포인트가 주위에 영향을 주는 거리를 지정합니다. 값이 클수록 더 평탄화되지만, 값이 작을수록 포인트 밀도의 상세한 변동을 자세히 볼 수 있을지도 모릅니다. |
| **Output raster size** | `PIXEL_SIZE` | [numeric: double]<br>기본값: 0.1 | 산출 래스터 레이어의 레이어 단위 픽셀 크기<br>GUI 에서 크기는 행/열의 개수 (`Number of rows` / `Number of columns`) **또는** 픽셀 크기 (`Pixel Size X` / `Pixel Size Y`) 로 지정할 수 있습니다. 행 또는 열의 개수를 늘리면 산출 래스터의 셀 크기가 작아지고 파일 용량은 커집니다. `Rows`, `Columns`, `Pixel Size X` 및 `Pixel Size Y` 의 값은 동시에 업데이트될 것입니다. 행의 개수를 2 배 늘리면 열의 개수도 2 배로 늘어나고, 셀 크기는 반으로 줄 것입니다. 산출 래스터의 범위는 (거의) 동일하게 유지됩니다. |
| **Radius from field** 선택적 | `RADIUS_FIELD` | [tablefield: numeric] | 입력 레이어의 속성 필드로부터 각 피처에 대한 검색 반경을 설정합니다. |
| **Weight from field** 선택적 | `WEIGHT_FIELD` | [tablefield: numeric] | 입력 피처에 속성 필드로 가중치를 줄 수 있습니다. 산출 히트 맵 상에서 특정 피처의 영향력을 높이는 데 사용할 수 있습니다. |
| **Kernel shape** | `KERNEL` | [enumeration]<br>기본값: *0* | 포인트로부터의 거리가 늘어날수록 포인트의 영향력이 낮아지는 비율 (rate) 을 제어합니다. 서로 다른 커널은 서로 다른 비율로 영향력이 낮아집니다. 예를 들어 삼중가중 (triweight) 커널은 예파네치니코프 (Епанечников; Epanechnikov) 커널보다 포인트에 더 가까운 거리에 더 높은 가중치를 줍니다. 즉 삼중가중 커널은“더 선명한 (sharper)”핫 스폿을 산출하고, 예파네치니코프 커널은“더 평탄한 (smoother)”핫 스폿을 산출합니다.<br>여러 형태를 사용할 수 있습니다. (더 자세한 내용은 위키백과 페이지 를 참조하세요.)<br>• 0 —4 차 다항 (Quartic)<br>• 1 —삼각 (Triangular)<br>• 2 —균일 (Uniform)<br>• 3 —삼중가중 (Triweight)<br>• 4 —예파네치니코프 (Epanechnikov) |

다음 페이지에 계속

표 24.151 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Decay ratio (Triangular kernels only)** 선택적 | `DECAY` | [numeric: double] 기본값: *0.0* | 삼각 커널 전용으로, 피처로부터의 거리가 멀어질수록 피처로부터의 열 (heat) 이 어떻게 낮아지는지 심도 있게 제어할 수 있습니다.<br>• 값이 (최소값인) 0 인 경우 열이 지정한 반경의 중심에 집중되고 경계에서는 완전히 꺼질 것입니다.<br>• 값이 0.5 인 경우 검색 반경의 경계에 있는 픽셀의 열이 중심에 있는 픽셀의 열의 반이 될 것입니다.<br>• 값이 1 인 경우 열이 검색 반경의 원 전체에 걸쳐 균등하게 퍼질 것입니다. (이는'균일'커널과 동일합니다.)<br>• 값이 1 보다 큰 경우 열이 경계로 갈수록 중심보다 높아질 것입니다. |
| **Output value scaling** | `OUTPUT_VALUE` | [enumeration] 기본값: *0* | 산출 히트 맵 래스터의 값을 변경할 수 있습니다. 다음 가운데 하나를 선택할 수 있습니다:<br>• 0 — 원본 (Raw)<br>• 1 — 척도 변경 (Scaled) |
| **Heatmap** | `OUTPUT` | [raster] 기본값: `[Save to temporary file]` | Specify the output raster layer with kernel density values. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Heatmap** | `OUTPUT` | [raster] | 커널 밀도값을 가진 래스터 레이어 |

### 예제: 히트 맵 생성하기

이 예제를 따라가려면, QGIS 예시 데이터셋 (예시 데이터 다운로드 참조) 에 있는 `airports` 벡터 포인트 레이어가 필요합니다. 히트 맵 생성에 대한 또다른 탁월한 QGIS 지침을 http://qgistutorials.com 에서 찾을 수 있습니다.

그림 24.24 에서 알래스카 주의 공항을 볼 수 있습니다.

1. QGIS *Interpolation* 그룹에서 *Heatmap (Kernel Density Estimation)* 알고리즘을 여십시오.
2. *Point layer*  필드에서 현재 프로젝트에 불러온 포인트 레이어의 목록 가운데 `airports` 를 선택하십시오.
3. *Radius* 를 `1000000` 미터로 변경하십시오.

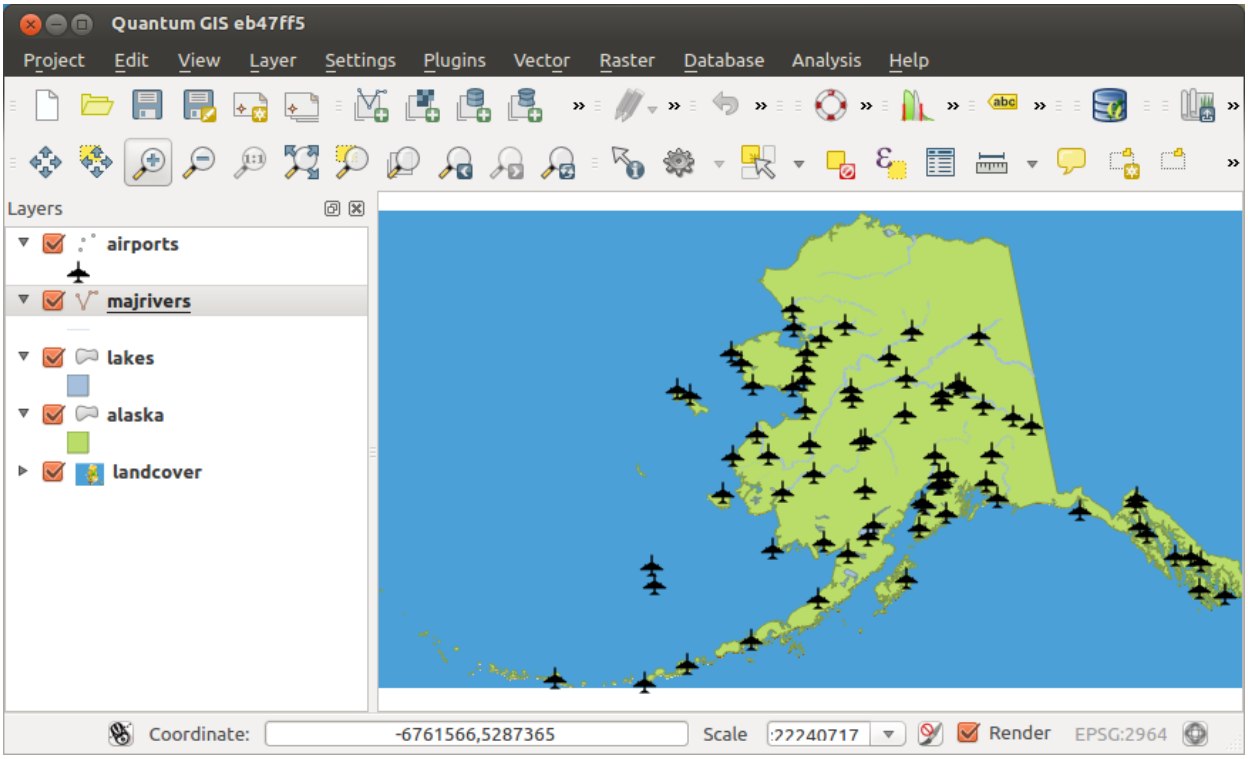



그림 24.24: 알래스카 주의 공항들

4. *Pixel size X* 를 `1000` 으로 변경하십시오. *Pixel size Y*, *Rows* 그리고 *Columns* 는 자동적으로 업데이트될 것입니다.
5. *Run* 버튼을 클릭해서 공항 히트 맵을 생성하고 불러오십시오. (그림 24.26 참조)

QGIS 가 히트 맵을 생성해서 사용자 맵 창에 추가할 것입니다. 히트 맵은 기본적으로 회색조 음영으로 렌더링됩니다. 색이 밝을수록 공항이 밀집되어 있다는 의미입니다. 이제 QGIS 에서 히트 맵에 스타일을 적용해서 모양을 향상시킬 수 있습니다.

1. `heatmap_airports` 레이어의 속성 대화창을 여십시오. (`heatmap_airports` 레이어를 선택한 다음, 오른쪽 클릭으로 컨텍스트 메뉴를 열고 *Properties* 메뉴를 선택하십시오.)
2. *Symbology* 탭을 선택하십시오.
3. *Render type*  옵션을'Singleband pseudocolor'로 변경하십시오.
4. *Color ramp*  에서 알맞은 색상표를, 예를 들어 `YlOrRd` 를 선택하십시오.
5. *Classify* 버튼을 클릭하십시오.
6. *OK* 를 클릭해서 레이어를 업데이트하십시오.

마지막 결과물은 그림 24.27 와 같습니다.

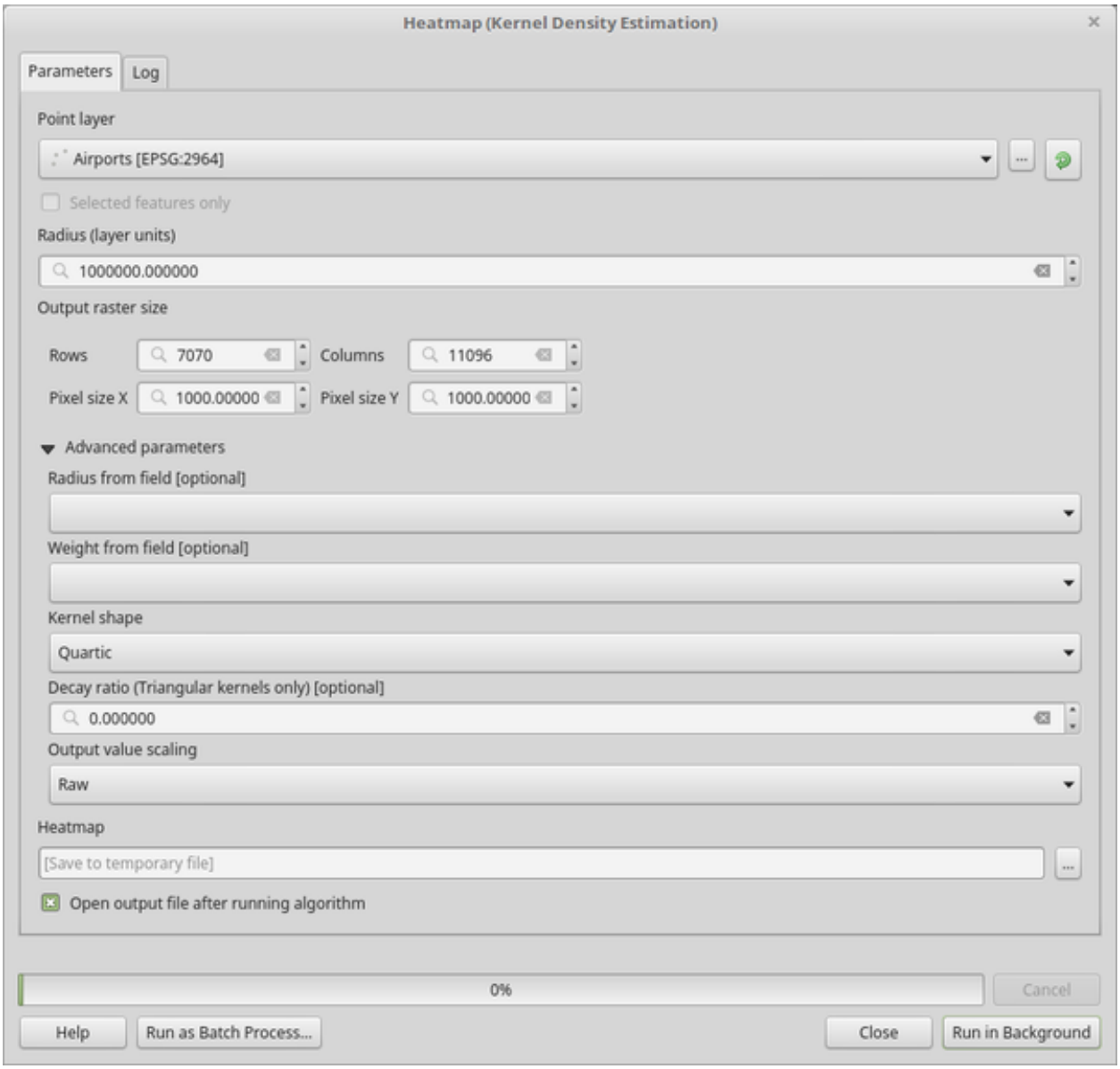



그림 24.25: 히트 맵 대화창

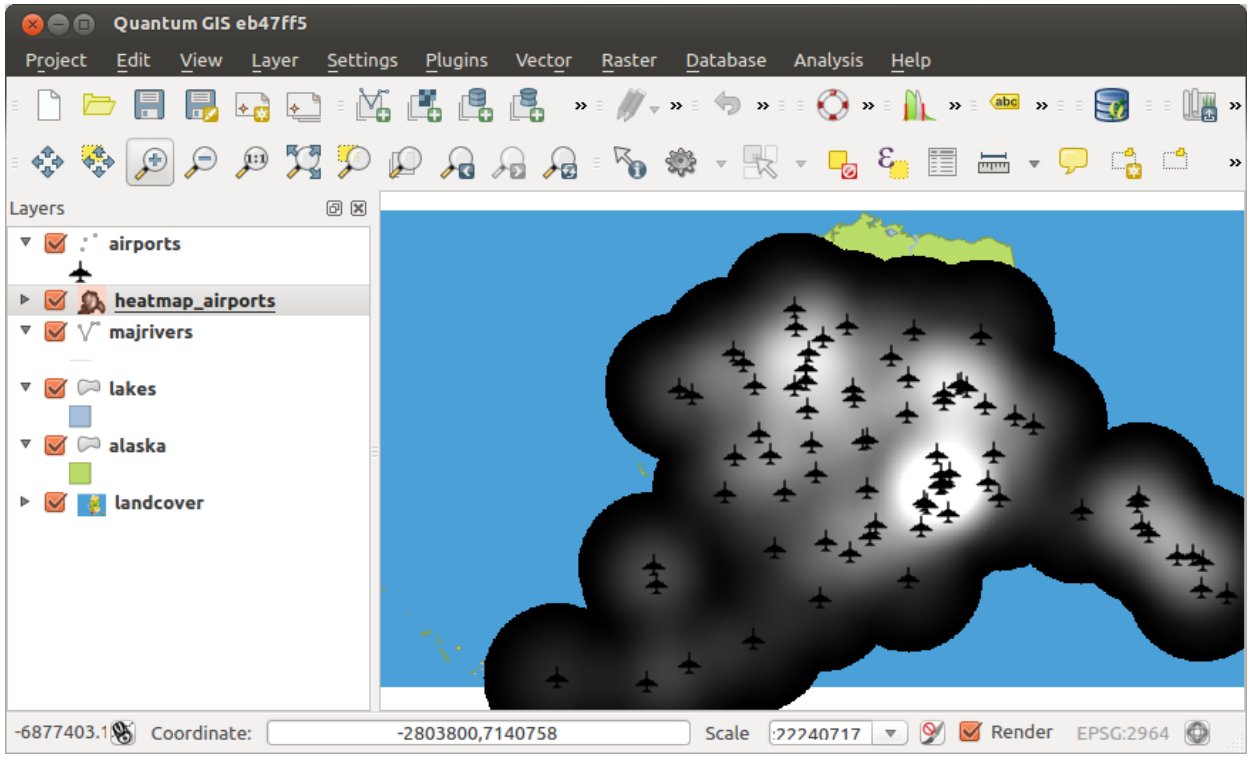



그림 24.26: 불러온 직후의 히트 맵은 회색 면처럼 보입니다

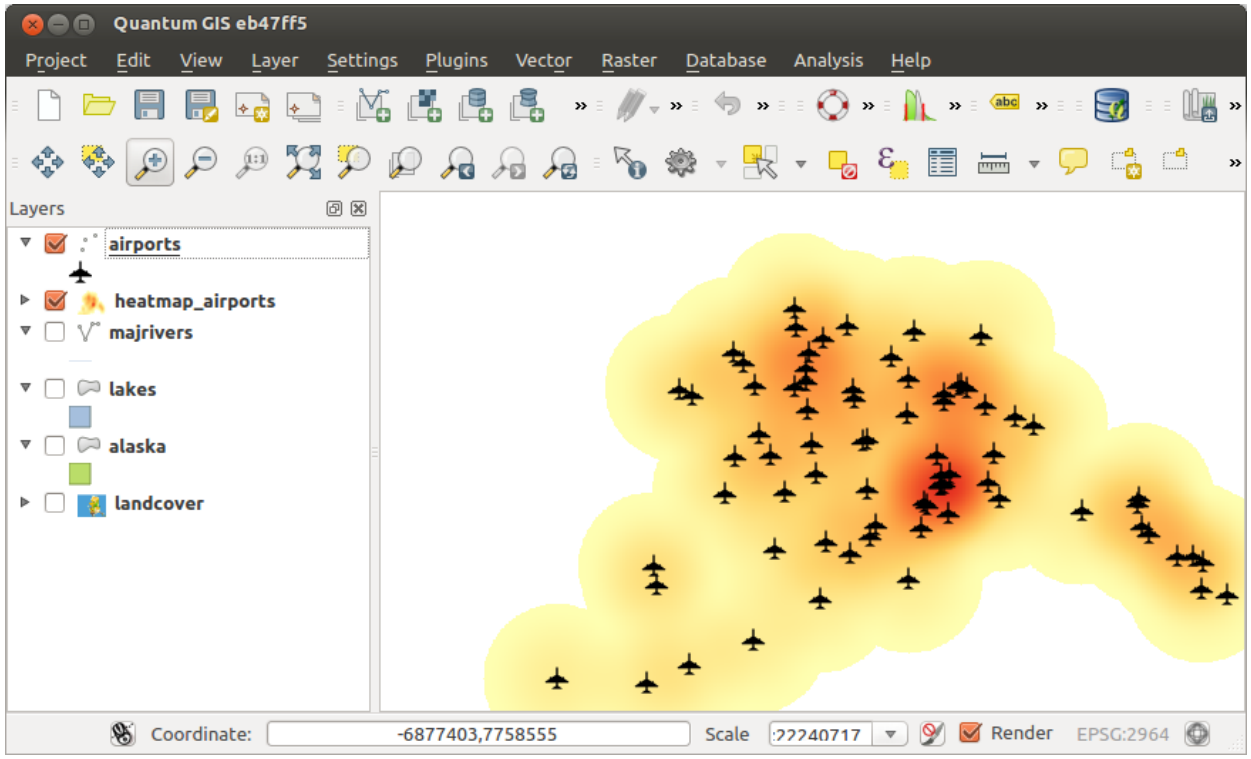



그림 24.27: 스타일을 적용한 알래스카 주 공항의 히트 맵

### 파이썬 코드

**알고리즘 ID**: `qgis:heatmapkerneldensityestimation`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 역거리 가중 보간법

포인트 벡터 레이어의 역거리 가중 (IDW; Inverse Distance Weighted) 보간을 생성합니다.

보간 처리 과정에서, 어떤 포인트가 사용자가 생성하고자 하는 알려지지 않은 포인트로부터 멀어질수록 해당 포인트의 영향력이 다른 포인트에 상대적으로 낮아지도록 예시 포인트에 가중치를 부여합니다.

역거리 가중 보간법도 약점이 있습니다. 예시 데이터 포인트가 균등하지 않게 분포한 경우 보간 셜과의 질이 떨어질 수 있습니다.

게다가, 보간된 면의 최대 및 최소값은 예시 데이터 포인트 위치에만 나타날 수 있습니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer(s)** | `INTERPOLATION_DATA` | [string] | 보간을 위해 사용할, 문자열로 코딩된 벡터 레이어 (들) 및 필드 (들) (더 자세한 내용은 InterpolationWidgets 에 있는 `ParameterInterpolationData` 클래스를 참조하세요.)<br>보간 데이터 문자열을 작성하기 위해 다음과 같은 GUI 항목이 제공됩니다:<br>• **Vector layer** [vector: any]<br>• **Interpolation attribute** [tablefield: numeric]: 보간 작업에 사용할 속성<br>• **Use Z-coordinate for interpolation** [boolean]: 레이어에 저장된 Z 값 (기본값: False)<br>추가된 각 레이어-필드 조합별로, 다음과 같은 유형을 선택할 수 있습니다:<br>• *Points*: Elevation sample locations containing the values to be interpolated (e.g., terrain height, Z). These points are the primary input for calculating the output surface.<br>• *Structured lines*: Lines representing known continuous surface features that guide the terrain shape or structure (e.g., ridges, road centerlines, levees). They describe terrain direction while assuming continuity along the line.<br>• *Break lines*: Lines representing known surface discontinuities or hard boundaries (e.g., cliffs, river banks, excavation edges, lake/pond outlines). They mark where the surface may change sharply. It is required if there are significant changes in elevation along the line.<br>문자열에서 레이어-필드 항목은 `'::|::'` 로 구분됩니다. 레이어-필드 항목의 하위 항목은 `'::~::'` 로 구분됩니다. |
| **Distance coefficient P** | `DISTANCE_COEFFICIE` | [numeric: double]<br>기본값: 2.0 | 보간 작업을 위한 거리 계수 (coefficient) 를 설정합니다. 최소값은 0.0, 최대값은 100.0 입니다. |

다음 페이지에 계속

표 24.153 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Extent (xmin, xmax, ymin, ymax)** | `EXTENT` | [extent] | 산출 래스터 레이어의 범위입니다.<br>다음과 같은 메소드를 사용할 수 있습니다:<br>• 레이어로부터 계산…: 현재 프로젝트에 불러온 레이어의 범위를 사용합니다.<br>• 레이어 맵으로부터 계산…: 활성 프로젝트의조판 맵 항목 범위를 사용합니다.<br>• 북마크로부터 계산…: 저장되어 있는북마크 의 범위를 사용합니다.<br>• 맵 캔버스 범위 사용<br>• 캔버스에 그리기: 연산에 넣을 범위를 정하는 직사각형을 클릭 & 드래그합니다.<br>• 좌표를 `xmin`, `xmax`, `ymin`, `ymax` 로 입력 |
| **Output raster size** | `PIXEL_SIZE` | [numeric: double]<br>기본값: 0.1 | 산출 래스터 레이어의 레이어 단위 픽셀 크기<br>GUI 에서 크기는 행/열의 개수 (`Number of rows` / `Number of columns`) **또는** 픽셀 크기 (`Pixel Size X` / `Pixel Size Y`) 로 지정할 수 있습니다. 행 또는 열의 개수를 늘리면 산출 래스터의 셀 크기가 작아지고 파일 용량은 커집니다. `Rows`, `Columns`, `Pixel Size X` 및 `Pixel Size Y` 의 값은 동시에 업데이트될 것입니다. 행의 개수를 2 배 늘리면 열의 개수도 2 배로 늘어나고, 셀 크기는 반으로 줄 것입니다. 산출 래스터의 범위는 (거의) 동일하게 유지됩니다. |
| **Interpolated** | `OUTPUT` | [raster]<br>기본값: `[Save to temporary file]` | Raster layer of interpolated values. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Interpolated** | `OUTPUT` | [raster] | 보간된 값을 가진 래스터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `qgis:idwinterpolation`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 라인 밀도

이 알고리즘은 각 래스터 셀에 대해 각 셀의 원형 구역 (circular neighbourhood) 내에 있는 선형 객체들의 밀도 측정값을 계산합니다. 이 측정값은 각 셀에 대해 원형 구역과 교차하는 모든 라인 선분을 합한 값을 해당 원형 구역의 면적으로 나누어 얻어집니다. 라인 선분에 가중치를 적용할 수 있습니다.

**참고**

This algorithm uses ellipsoid based measurements and respects the current *ellipsoid settings*.

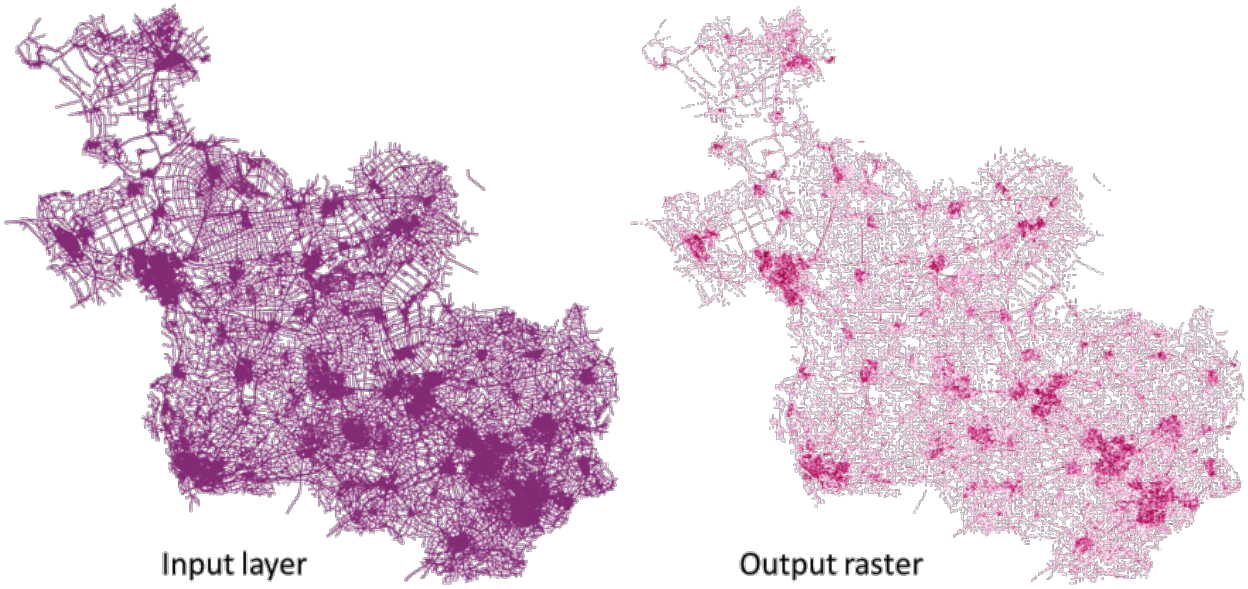



그림 24.28: 라인 밀도의 예. 입력 레이어 소스: 네덜란드 오퍼레이셀 주의 도로 (OSM)

### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input line layer** | `INPUT` | [vector: line] | 라인 객체를 담고 있는 입력 벡터 레이어 |
| **Weight field** | `WEIGHT` | [tablefield: numeric] | 계산 도중 사용할 가중치 인자를 담고 있는 레이어 필드 |
| **Search Radius** | `RADIUS` | [numeric: double]<br>기본값: 10.0 | 원형 구역의 반경. 여기에 단위를 지정할 수 있습니다. |
| **Pixel size** | `PIXEL_SIZE` | [numeric: double]<br>기본값: 10.0 | 산출 래스터 레이어의 레이어 단위 픽셀 크기. 래스터 픽셀은 정사각형입니다. |
| **Line density raster** | `OUTPUT` | [raster]<br>기본값: `[Save to temporary file]` | The output as a raster layer. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

#### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Creation options**<br>선택적 | `CREATION_OPTIONS`<br>(for QGIS <= 3.42, this was `CREATE_OPTIONS`) | [string]<br>기본값: '' | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL* 드라이버 옵션 참조) 배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (`\|`) 로 복수의 옵션들을 분리하십시오. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Line density raster** | `OUTPUT` | [raster] | 산출 라인 밀도 래스터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:linedensity`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 삼각불규칙망 보간법

포인트 벡터 레이어의 삼각불규칙망 (TIN; Triangulated Irregular Network) 보간을 생성합니다.

삼각불규칙망 보간법을 사용하면 최근접 이웃한 포인트로 이루어진 삼각형으로 형성된 면을 생성할 수 있습니다. 삼각불규칙망을 생성하려면 선택한 예시 포인트들 주위에 외접원 (circumcircle) 을 생성하고, 외접원들의 교차점을 중첩하지 않는 망으로 연결해서 가능한 한 촘촘한 삼각형들을 생성해야 합니다. 산출되는 면은 평탄하지 않습니다.

이 알고리즘은 보간된 값을 가진 래스터 레이어와 삼각망 경계를 가진 벡터 라인 레이어를 둘 다 생성합니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer(s)** | `INTERPOLATION_DATA` | [string] | 보간을 위해 사용할, 문자열로 코딩된 벡터 레이어 (들) 및 필드 (들) (더 자세한 내용은 InterpolationWidgets 에 있는 `ParameterInterpolationData` 클래스를 참조하세요.)<br>보간 데이터 문자열을 작성하기 위해 다음과 같은 GUI 항목이 제공됩니다:<br>• **Vector layer** [vector: any]<br>• **Interpolation attribute** [tablefield: numeric]: 보간 작업에 사용할 속성<br>• **Use Z-coordinate for interpolation** [boolean]: 레이어에 저장된 Z 값 (기본값: False)<br>추가된 각 레이어-필드 조합별로, 다음과 같은 유형을 선택할 수 있습니다:<br>• *Points*: Elevation sample locations containing the values to be interpolated (e.g., terrain height, Z). These points are the primary input for calculating the output surface.<br>• *Structured lines*: Lines representing known continuous surface features that guide the terrain shape or structure (e.g., ridges, road centerlines, levees). They describe terrain direction while assuming continuity along the line.<br>• *Break lines*: Lines representing known surface discontinuities or hard boundaries (e.g., cliffs, river banks, excavation edges, lake/pond outlines). They mark where the surface may change sharply. It is required if there are significant changes in elevation along the line.<br>문자열에서 레이어-필드 항목은 `'::|::'` 로 구분됩니다. 레이어-필드 항목의 하위 항목은 `'::~::'` 로 구분됩니다. |

다음 페이지에 계속

표 24.158 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Interpolation method** | METHOD | [enumeration] 기본값: 0 | 사용할 보간법을 설정합니다. 다음 가운데 하나를 선택할 수 있습니다:<br>• *Linear*<br>• *Clough-Toucher (cubic)* |
| **Extent (xmin, xmax, ymin, ymax)** | EXTENT | [extent] | 산출 래스터 레이어의 범위입니다.<br>다음과 같은 메소드를 사용할 수 있습니다:<br>• 레이어로부터 계산…: 현재 프로젝트에 불러온 레이어의 범위를 사용합니다.<br>• 레이어 맵으로부터 계산…: 활성 프로젝트의조판 맵 항목 범위를 사용합니다.<br>• 북마크로부터 계산…: 저장되어 있는북마크 의 범위를 사용합니다.<br>• 맵 캔버스 범위 사용<br>• 캔버스에 그리기: 연산에 넣을 범위를 정하는 직사각형을 클릭 & 드래그합니다.<br>• 좌표를 xmin, xmax, ymin, ymax 로 입력 |
| **Output raster size** | PIXEL_SIZE | [numeric: double] 기본값: 0.1 | 산출 래스터 레이어의 레이어 단위 픽셀 크기<br>GUI 에서 크기는 행/열의 개수 (Number of rows / Number of columns) **또는** 픽셀 크기 (Pixel Size X / Pixel Size Y) 로 지정할 수 있습니다. 행 또는 열의 개수를 늘리면 산출 래스터의 셀 크기가 작아지고 파일 용량은 커집니다. Rows, Columns, Pixel Size X 및 Pixel Size Y 의 값은 동시에 업데이트될 것입니다. 행의 개수를 2 배 늘리면 열의 개수도 2 배로 늘어나고, 셀 크기는 반으로 줄 것입니다. 산출 래스터의 범위는 (거의) 동일하게 유지됩니다. |
| **Interpolated** | OUTPUT | [raster] 기본값: [Save to temporary file] | The output TIN interpolation as a raster layer. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |
| **Triangulation** | TRIANGULATION | [vector: line] 기본값: [Skip output] | The output TIN as a vector layer. *One of*:<br>• 산출물 건너뛰기<br>• 임시 레이어 생성 (TEMPORARY_OUTPUT)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

#### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Interpolated** | `OUTPUT` | [raster] | 삼각불규칙망 보간의 산출물을 저장한 래스터 레이어 |
| **Triangulation** | `TRIANGULATION` | [vector: line] | 삼각불규칙망 보간의 산출물을 저장한 벡터 레이어 |

#### 파이썬 코드

**알고리즘 ID**: `qgis:tininterpolation`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 24.1.9 레이어 도구

#### 레이어 정보 내보내기

선택한 레이어 (들) 의 범위에 대응하는 피처를 가진 폴리곤 레이어를 생성합니다.

각 피처에 추가적인 레이어 상세 정보 (좌표계, 제공자 이름, 파일 경로, 레이어 이름, 하위 집합 필터, 요약 및 속성) 을 속성 (attribute) 으로 첨부합니다.

#### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layers** | `LAYERS` | [layer] [list] | Input layers to get information on. |
| **Output** | `OUTPUT` | [vector: polygon] 기본값: `[Create temporary layer]` | Specification of the output layer with information. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output** | `OUTPUT` | [vector: polygon] | 입력 레이어의 범위 및 관련 정보를 속성으로 가진 폴리곤 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:exportlayersinformation`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 스프레드시트로 내보내기

벡터 레이어 선택 집합의 속성을 스프레드시트 문서로 내보내거나, 선택적으로 기존 스프레드시트에 추가적인 시트로 첨부합니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layers** | `LAYERS` | [vector: any] [list] | 입력 벡터 레이어입니다. 산출 스프레드시트는 각 레이어에 대한 시트 하나씩으로 이루어지며, 각 시트는 해당 레이어의 속성을 담고 있습니다. |
| **Use field aliases as column headings** | `USE_ALIAS` | [boolean]<br>기본값: False | 스프레드시트에 속성 테이블의 필드 별명을 사용합니다. |
| **Export formatted values instead of raw values** | `FORMAT-TED_VALUES` | [boolean]<br>기본값: False | `True` 로 설정하면 스프레드시트에 (예를 들면값들의 맵 또는 값들의 관계 에서 나온) 서식을 가진, 사람이 읽을 수 있는 값으로 내보냅니다. |
| **Overwrite existing spreadsheet** | `OVERWRITE` | [boolean]<br>기본값: True | 지정한 스프레드시트가 존재하는 경우, 이 옵션을 `True` 로 설정하면 기존 스프레드시트를 덮어쓸 것입니다. 이 옵션을 `False` 로 설정했는데 기존 스프레드시트가 존재한다면, 레이어를 추가적인 시트로 첨부할 것입니다. |
| **Destination spreadsheet** | `OUTPUT` | [file]<br>기본값: `[Save to temporary file]` | Output spreadsheet with a sheet for every layer. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Destination spreadsheet** | `OUTPUT` | [file] | 모든 레이어에 대해 각각 시트 1 개를 가진 스프레드시트 |
| **Layers within spreadsheet** | `OUTPUT_LAYERS` | [list] | 스프레드시트에 추가된 시트 목록입니다. |

### 파이썬 코드

**알고리즘 ID**: `native:exporttospreadsheet`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 레이어 범위 추출하기

모든 입력 레이어를 커버하는 최소 경계 상자 (북-남 방향의 직사각형) 를 가진 벡터 레이어를 생성합니다.

산출 레이어는 입력 레이어 전체의 경계 상자 하나를 담고 있습니다.

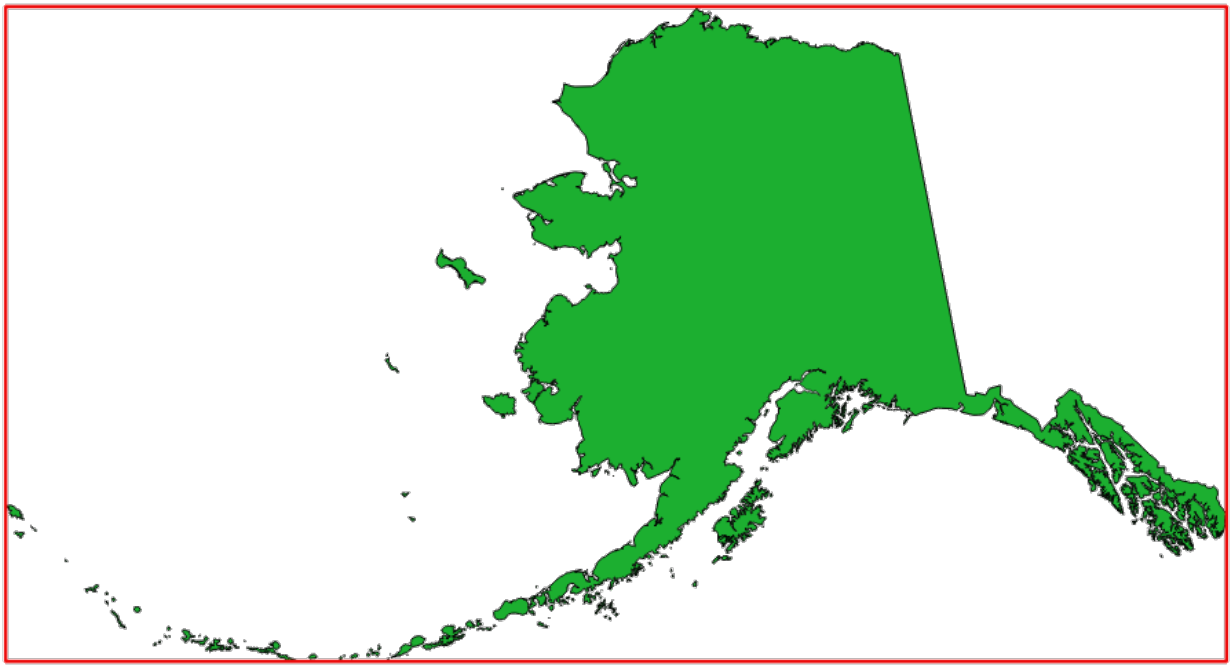


그림 24.29: 소스 레이어의 빨간색 경계 상자

**기본 메뉴**: *Vector → Research Tools*

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Layer** | `INPUT` | [layer] | 입력 레이어 |
| **Extent** | `OUTPUT` | [vector: polygon] 기본값: `[Create temporary layer]` | Specify the polygon vector layer for the output extent. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Extent** | `OUTPUT` | [vector: polygon] | 범위를 (최소 경계 상자를) 가진 산출물 (폴리곤) 벡터 레이어 |

### 파이썬 코드

**Algorithm ID**: `native:polygonfromlayerextent`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 24.1.10 메시 (mesh)

#### 등고선 내보내기

메시 스칼라 데이터셋으로부터 등고선을 벡터 레이어로 생성합니다.

#### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input mesh layer** | INPUT | [mesh] | 데이터를 내보낼 메시 레이어 |
| **Dataset groups** | DATASET_GROUPS | [layer] [list] | 데이터셋 그룹 |
| **Dataset time** | DATASET_TIME | [datetime] | 연산에 넣을 시간 범위<br>• 0 — 현재 캔버스 시간<br>• 1 — 정의된 날짜 & 시간<br>• 2 — 데이터셋 그룹의 시간 지점 |
| **Increment between contour levels**<br>선택적 | INCREMENT | [numeric: double]<br>기본값: 설정하지 않음 | 생성된 수준들 사이의 간격 |
| **Minimum contour level**<br>선택적 | MINIMUM | [numeric: double]<br>기본값: 설정하지 않음 | Starting level values of contours. No generated levels will be lower than this value. |
| **Maximum contour level**<br>선택적 | MAXIMUM | [numeric: double]<br>기본값: 설정하지 않음 | 등고선의 최대값. 예를 들어 생성된 수준 가운데 이 값을 초과하는 수준은 없을 것입니다. |
| **List of contours level**<br>선택적 | CONTOUR_LEVEL_LIST | [string]<br>기본값: 설정하지 않음 | List of desired levels of contours (separated by commas). If filled, the increment, minimum, and maximum fields will not be considered. |
| **Output coordinate system**<br>선택적 | CRS_OUTPUT | [crs] | 산출물에 할당할 좌표계 |
| **Exported contour lines** | OUTPUT_LINES | [vector: line]<br>기본값: [Create temporary layer] | Specify the output line layer representing the contours of the mesh layer. *One of*:<br>• 임시 레이어 생성 (TEMPORARY_OUTPUT)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |
| **Exported contour polygons** | OUTPUT_POLYGONS | [vector: polygon]<br>기본값: [Create temporary layer] | Specify the output polygon layer representing the contours of the mesh layer. *One of*:<br>• 임시 레이어 생성 (TEMPORARY_OUTPUT)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Exported contour lines** | `OUTPUT_LINES` | [vector: line] | 메시 레이어의 등고선을 표현하는 라인 레이어 |
| **Exported contour polygons** | `OUTPUT_POLYGONS` | [vector: polygon] | 메시 레이어의 등고선을 표현하는 폴리곤 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:meshcontours`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 메시에서 라인 상의 단면 (cross section) 데이터셋 값 내보내기

벡터 레이어가 담고 있는 라인에서 메시 데이터셋의 값을 추출합니다.

각 라인을 해상도 거리 파라미터를 이용해서 라인의 꼭짓점에서 값을 추출하기 위해 이산화 시킵니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input mesh layer** | `INPUT` | [mesh] | 데이터를 내보낼 메시 레이어 |
| **Dataset groups** | `DATASET_GROUPS` | [layer] [list] | 데이터셋 그룹 |
| **Dataset time** | `DATASET_TIME` | [datetime] | 연산에 넣을 시간 범위<br>• 0 —현재 캔버스 시간<br>• 1 —정의된 날짜 & 시간<br>• 2 —데이터셋 그룹의 시간 지점 |
| **Lines for data export** | `INPUT_LINES` | [vector: line] | 데이터셋 메시에서 데이터를 추출하게 될 라인 |
| **Line segmentation resolution** | `RESOLUTION` | [numeric: double]<br>기본값: 10.0 | The distance between points on the lines where the data will be extracted from the dataset mesh. Units can be selected. |
| **Digits count for coordinates** | `COORDINATES_DIGITS` | [numeric: integer]<br>기본값: 2 | Number of digits for rounding the coordinate values |
| **Digits count for dataset value** | `DATASET_DIGITS` | [numeric: integer]<br>기본값: 2 | Number of digits for rounding the dataset values |

다음 페이지에 계속

표 24.168 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Exported data CSV file** | `OUTPUT` | [file]<br>기본값: `[Save to temporary file]` | Specification of the output file. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Exported data CSV file** | `OUTPUT` | [file] | |

### 파이썬 코드

**알고리즘 ID**: `native:meshexportcrosssection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 메시 변 내보내기

메시 레이어의 변 (edge) 을 변의 데이터셋 값을 속성값으로 가진 라인 벡터 레이어로 내보냅니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input mesh layer** | `INPUT` | [mesh] | 데이터를 내보낼 메시 레이어 |
| **Dataset groups** | `DATASET_GROUPS` | [layer] [list] | 데이터셋 그룹 |
| **Dataset time** | `DATASET_TIME` | [datetime] | 연산에 넣을 시간 범위<br>• 0 —현재 캔버스 시간<br>• 1 —정의된 날짜 & 시간<br>• 2 —데이터셋 그룹의 시간 지점 |
| **Output coordinate system**<br>선택적 | `CRS_OUTPUT` | [crs] | 산출물에 할당할 좌표계 |

다음 페이지에 계속

표 24.170 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Export vector option** | `VECTOR_OPTION` | [enumeration] | 내보내는 벡터 값의 좌표 유형<br>• 0 — 데카르트 평면 좌표 (x,y)<br>• 1 — 극좌표 (크기 (magnitude), 도 단위)<br>• 2 — 데카르트 평면 좌표 및 극좌표 |
| **Output vector layer** | `OUTPUT` | [vector: line]<br>기본값: `[Create temporary layer]` | Specification of the output file. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output vector layer** | `OUTPUT` | [vector: line] | 입력 메시 레이어의 변을 관련 데이터셋 값과 함께 담고 있는 산출 벡터 라인 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:exportmeshedges`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 메시 면 내보내기

메시 레이어의 면 (face) 을 면의 데이터셋 값을 속성값으로 가진 폴리곤 벡터 레이어로 내보냅니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input mesh layer** | `INPUT` | [mesh] | 데이터를 내보낼 메시 레이어 |
| **Dataset groups** | `DATASET_GROUPS` | [layer] [list] | 데이터셋 그룹 |
| **Dataset time** | `DATASET_TIME` | [datetime] | 연산에 넣을 시간 범위<br>• 0 — 현재 캔버스 시간<br>• 1 — 정의된 날짜 & 시간<br>• 2 — 데이터셋 그룹의 시간 지점 |
| **Output coordinate system**<br>선택적 | `CRS_OUTPUT` | [crs] | 산출물에 할당할 좌표계 |
| **Export vector option** | `VECTOR_OPTION` | [enumeration] | 내보내는 벡터 값의 좌표 유형<br>• 0 — 데카르트 평면 좌표 (x,y)<br>• 1 — 극좌표 (크기 (magnitude), 도 단위)<br>• 2 — 데카르트 평면 좌표 및 극좌표 |
| **Output vector layer** | `OUTPUT` | [vector: polygon]<br>기본값: `[Create temporary layer]` | Specification of the output file. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output vector layer** | `OUTPUT` | [vector: polygon] | 입력 메시 레이어의 면을 관련 데이터셋 값과 함께 담고 있는 산출 벡터 폴리곤 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:exportmeshfaces`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 메시를 그리드 상에 내보내기

메시 레이어의 데이터셋 값을 각 포인트의 데이터셋 값을 속성값으로 가진 그리드화된 포인트 벡터 레이어로 내보냅니다.

부피 (3 차원으로 쌓아 올린 데이터셋 값) 관련 데이터의 경우, 면 위에서 내보낸 데이터셋 값들을메시 레이어 속성 에 정의된 메소드를 (기본값은 다중 수준 평균화 메소드입니다) 이용해서 평균을 냅니다. 1 차원 메시는 지원하지 않습니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input mesh layer** | `INPUT` | [mesh] | 데이터를 내보낼 메시 레이어 |
| **Dataset groups** | `DATASET_GROUPS` | [layer] [list] | 데이터셋 그룹 |
| **Dataset time** | `DATASET_TIME` | [datetime] | 연산에 넣을 시간 범위<br>• 0 —현재 캔버스 시간<br>• 1 —정의된 날짜 & 시간<br>• 2 —데이터셋 그룹의 시간 지점 |
| **Extent**<br>선택적 | `EXTENT` | [extent] | 데이터를 처리할 공간 범위를 지정합니다.<br>다음과 같은 메소드를 사용할 수 있습니다:<br>• 레이어로부터 계산…: 현재 프로젝트에 불러온 레이어의 범위를 사용합니다.<br>• 레이어 맵으로부터 계산…: 활성 프로젝트의조판 맵 항목 범위를 사용합니다.<br>• 북마크로부터 계산…: 저장되어 있는북마크 의 범위를 사용합니다.<br>• 맵 캔버스 범위 사용<br>• 캔버스에 그리기: 연산에 넣을 범위를 정하는 직사각형을 클릭 & 드래그합니다.<br>• 좌표를 `xmin, xmax, ymin, ymax` 로 입력 |
| **Grid spacing**<br>선택적 | `GRID_SPACING` | [numeric: double]<br>기본값: 10.0 | 사용할 샘플 포인트들 사이의 간격 |

다음 페이지에 계속

표 24.174 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output coordinate system**<br>선택적 | `CRS_OUTPUT` | [crs] | 산출물에 할당할 좌표계 |
| **Export vector option** | `VECTOR_OPTION` | [enumeration] | 내보내는 벡터 값의 좌표 유형<br>• 0 —데카르트 평면 좌표 (x,y)<br>• 1 —극좌표 (크기 (magnitude), 도 단위)<br>• 2 —데카르트 평면 좌표 및 극좌표 |
| **Output vector layer** | `OUTPUT` | [vector: point]<br>기본값: `[Create temporary layer]` | Specification of the output file. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output vector layer** | `OUTPUT` | [vector: point] | 겹친 면으로부터 계산한 데이터셋 값을 가진 산출 벡터 포인트 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:exportmeshongrid`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 메시 꼭짓점 내보내기

메시 레이어의 꼭짓점을 꼭짓점의 데이터셋 값을 속성값으로 가진 포인트 벡터 레이어로 내보냅니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input mesh layer** | `INPUT` | [mesh] | 데이터를 내보낼 메시 레이어 |
| **Dataset groups** | `DATASET_GROUPS` | [layer] [list] | 데이터셋 그룹 |
| **Dataset time** | `DATASET_TIME` | [datetime] | 연산에 넣을 시간 범위<br>• 0 — 현재 캔버스 시간<br>• 1 — 정의된 날짜 & 시간<br>• 2 — 데이터셋 그룹의 시간 지점 |
| **Output coordinate system**<br>선택적 | `CRS_OUTPUT` | [crs] | 산출물에 할당할 좌표계 |
| **Export vector option** | `VECTOR_OPTION` | [enumeration] | 내보내는 벡터 값의 좌표 유형<br>• 0 — 데카르트 평면 좌표 (x,y)<br>• 1 — 극좌표 (크기 (magnitude), 도 단위)<br>• 2 — 데카르트 평면 좌표 및 극좌표 |
| **Output vector layer** | `OUTPUT` | [vector: point]<br>기본값: `[Create temporary layer]` | Specification of the output file. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output vector layer** | `OUTPUT` | [vector: point] | 입력 메시 레이어의 꼭짓점을 관련 데이터셋 값과 함께 담고 있는 산출 벡터 포인트 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:exportmeshvertices`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 메시 데이터셋의 포인트에서 시계열 값 내보내기

벡터 레이어가 담고 있는 포인트에서 메시 데이터셋의 시계열 값을 추출합니다.

시간 지점 (time step) 이 기본값 (0 시간) 을 유지하고 있는 경우, 맨 처음 선택한 데이터셋 그룹의 처음 두 데이터셋의 시간 지점을 사용합니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input mesh layer** | `INPUT` | [mesh] | 데이터를 추출할 메시 레이어 |
| **Dataset groups** | `DATASET_GROUPS` | [layer] [list] | 데이터셋 그룹 |
| **Starting time** | `STARTING_TIME` | [datetime] | 연산에 넣을 시간 범위의 시작 시간<br>• 0 —현재 캔버스 시간<br>• 1 —정의된 날짜 & 시간<br>• 2 —데이터셋 그룹의 시간 지점 |
| **Finishing time** | `FINISHING_TIME` | [datetime] | 연산에 넣을 시간 범위의 종단 시간<br>• 0 —현재 캔버스 시간<br>• 1 —정의된 날짜 & 시간<br>• 2 —데이터셋 그룹의 시간 지점 |
| **Time step (hours)** 선택적 | `TIME_STEP` | [numeric: double]<br>기본값: 0 | 추출할 연속 시간 지점 2 개 사이의 시간입니다. `0` 으로 유지하면 맨 처음 선택한 데이터셋 그룹의 시간 지점을 사용합니다. |
| **Points for data export** | `INPUT_POINTS` | [vector: point] | 데이터셋 메시에서 데이터를 추출하게 될 포인트를 담고 있는 벡터 레이어 |
| **Digits count for coordinates** | `COORDINATES_DIGITS` | [numeric: integer] | Number of digits for rounding coordinate values<br>기본값: 2 |
| **Digits count for dataset value** | `DATASET_DIGITS` | [numeric: integer]<br>기본값: 2 | Number of digits for rounding dataset values |
| **Exported data CSV file** | `OUTPUT` | [file]<br>기본값: `[Save to temporary file]` | Specification of the output file. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Exported data CSV file** | `OUTPUT` | [file] | 중첩하는 포인트 피처 위치의 메시 데이터셋 시계열 값을 담고 있는 `.CSV` 파일 |

### 파이썬 코드

**알고리즘 ID**: `native:meshexporttimeseries`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 메시 데이터셋 래스터화하기

메시 데이터셋으로부터 래스터 레이어를 생성합니다.

부피 (3 차원으로 쌓아 올린 데이터셋 값) 관련 데이터의 경우, 면 위에서 내보낸 데이터셋 값들을메시 레이어 속성 에 정의된 메소드를 (기본값은 다중 수준 평균화 메소드입니다) 이용해서 평균을 냅니다. 1 차원 메시는 지원하지 않습니다.

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input mesh layer** | `INPUT` | [mesh] | 데이터를 내보낼 메시 레이어 |
| **Dataset groups** | `DATASET_GROUPS` | [layer] [list] | 데이터셋 그룹 |
| **Dataset time** | `DATASET_TIME` | [datetime] | 연산에 넣을 시간 범위<br>• 0 —현재 캔버스 시간<br>• 1 —정의된 날짜 & 시간<br>• 2 —데이터셋 그룹의 시간 지점 |

다음 페이지에 계속

표 24.180 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Extent**<br>선택적 | EXTENT | [extent] | 데이터를 처리할 공간 범위를 지정합니다. 다음과 같은 메소드를 사용할 수 있습니다:<br>• 레이어로부터 계산…: 현재 프로젝트에 불러온 레이어의 범위를 사용합니다.<br>• 레이어 맵으로부터 계산…: 활성 프로젝트의조판 맵 항목 범위를 사용합니다.<br>• 북마크로부터 계산…: 저장되어 있는북마크 의 범위를 사용합니다.<br>• 맵 캔버스 범위 사용<br>• 캔버스에 그리기: 연산에 넣을 범위를 정하는 직사각형을 클릭 & 드래그합니다.<br>• 좌표를 xmin, xmax, ymin, ymax 로 입력 |
| **Pixel size** | PIXEL_SIZE | [numeric: double]<br>기본값: 1.0 | 산출 래스터 레이어의 픽셀 크기 |
| **Output coordinate system**<br>선택적 | CRS_OUTPUT | [crs] | 산출물에 할당할 좌표계 |
| **Output raster layer** | OUTPUT | [raster]<br>기본값: [Save to temporary file] | Specification of the output file. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Creation options**<br>선택적 | CREATION_OPTIONS (for QGIS <= 3.42, this was CREATE_OPTIONS) | [string]<br>기본값: ' ' | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL* 드라이버 옵션 참조) 배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (\|) 로 복수의 옵션들을 분리하십시오. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output raster layer** | `OUTPUT` | [raster] | 메시 레이어로부터 계산한 데이터셋 값을 가진 산출 래스터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:meshrasterize`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## Surface to Polygon

Exports a polygon file containing the boundary of a mesh layer. The resulting polygon may contain holes and may be a multi-part polygon.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input mesh layer** | `INPUT` | [mesh] | The input mesh layer whose boundary will be exported as a polygon |
| **Output coordinate system**<br>선택적 | `OUTPUT_CRS` | [crs] | The coordinate reference system (CRS) for the output polygon layer. If not specified, the CRS of the input mesh layer will be used. |
| **Output vector layer** | `OUTPUT` | [vector: polygon]<br>기본값: `[Create temporary layer]` | Specification of the output file. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output vector layer** | `OUTPUT` | [vector: polygon] | The resulting polygon layer containing the boundary of the mesh layer. The polygon may contain holes and may be multi-part. |

### 파이썬 코드

**Algorithm ID**: `native:surfacetopolygon`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## TIN 메시 생성하기

벡터 레이어로부터 TIN 메시 레이어를 생성합니다. TIN 메시는 델로네 삼각분할 (Delaunay triangulation) 을 이용해서 생성됩니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layers** | `SOURCE_DATA` | [vector: any] [list] | 메시 레이어를 생성하기 위해 결합될 벡터 레이어들 |
| **Vector layer** | GUI 만을 통해서 지정 | [vector: any] [list] | 메시 레이어를 생성하기 위해 결합될 벡터 레이어를 선택하기 위한 선택기 |
| **Value on vertex** | GUI 만을 통해서 지정 | [tablefield: any] | 선택한 레이어에서 사용할 필드를 선택하기 위한 선택기입니다. 각 꼭짓점에 각 꼭짓점의 원본 피처의 대응하는 값을 할당합니다. |
| **Use Z-coordinate for value on vertex** | GUI 만을 통해서 지정 | [boolean]<br>기본값: False | 이 옵션을 체크하면, 꼭짓점 메시 레이어의 Z 값에 벡터 레이어 포인트 또는 폴리곤/라인의 꼭짓점의 Z 값을 할당합니다. 이 옵션은 입력 레이어가 3 차원일 경우에만 사용할 수 있습니다. |
| **Output format** | `MESH_FORMAT` | [enumeration]<br>기본값: 2DM | 생성된 레이어의 산출물 포맷<br>• 0 —2DM<br>• 1 —SELAFIN<br>• 2 —PLY<br>• 3 —Ugrid<br>• 4 —Mike21 |

다음 페이지에 계속

표 24.185 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output coordinate system**<br>선택적 | `CRS_OUTPUT` | [crs] | 산출물에 할당할 좌표계 |
| **Output file** | `OUTPUT_MESH` | [mesh]<br>기본값: `[Save to temporary file]` | Specification of the output file. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output file** | `OUTPUT_MESH` | [mesh] | 벡터 레이어로부터 계산한 데이터셋 값을 가진 산출 메시 레이어 |

파이썬 코드

**알고리즘 ID**: `native:tinmeshcreation`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 24.1.11 Metadata tools

### Add history metadata

Adds a new history entry to the layer's metadata.

### Parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Layer** | `INPUT` | [layer] | The input layer (vector, raster, etc.) to which the history entry will be added. |
| **History entry** | `HISTORY` | [string] | The text to be appended as a new entry in the layer's history metadata. This will be added to any existing history entries. |

### Outputs

| Label | Name | Type | Description |
|---|---|---|---|
| **Output** | `OUTPUT` | [same as input] | The resulting layer with the updated history in its **Metadata properties**. |

### Python code

**Algorithm ID**: `native:addhistorymetadata`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 콘솔에서 공간 처리 알고리즘 사용 for details on how to run processing algorithms from the Python console.

## Copy Layer Metadata

Copies metadata from a source layer to a target layer. Any existing metadata in the target layer will be replaced with the metadata from the source layer. This includes all metadata fields, such as history, abstract, and other properties.

### Parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Source layer** | `SOURCE` | [layer] | The layer from which metadata will be copied. |
| **Target layer** | `TARGET` | [layer] | The layer to which metadata will be pasted. Any existing metadata in this layer will be replaced. |
| **Save metadata as default** | `DEFAULT` | [boolean]<br>Default: False | If checked, the metadata information will be saved with the layer, hence available by default in subsequent projects. |

### Outputs

| Label | Name | Type | Description |
|---|---|---|---|
| **Output layer** | `OUTPUT` | [layer] | The target layer with the metadata replaced by the metadata from the source layer. This includes all metadata fields, such as history, abstract, and other properties. |

### Python code

**Algorithm ID**: `native:copylayermetadata`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 콘솔에서 공간 처리 알고리즘 사용 for details on how to run processing algorithms from the Python console.

## Export Layer Metadata

Exports the metadata of a layer to a QMD file.

### Parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Layer** | `INPUT` | [layer] | The layer whose metadata will be exported. |
| **Output** | `OUTPUT` | [file]<br>Default: `[Save to temporary file]` | Specification of the output file. *One of*:<br>• Save to a Temporary File<br>• Save to File… |

### Outputs

| Label | Name | Type | Description |
|---|---|---|---|
| **Output file** | `OUTPUT` | [file] | The `.qmd` file containing the exported metadata. This file can be used to import metadata into another layer. |

### Python code

**Algorithm ID**: `native:exportlayermetadata`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 콘솔에서 공간 처리 알고리즘 사용 for details on how to run processing algorithms from the Python console.

### Set Layer Metadata

Applies metadata to a layer from a `.qmd` file.

### Parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Layer** | `INPUT` | [layer] | The layer to which the metadata will be applied. |
| **Metadata file** | `METADATA` | [file] | The `.qmd` file containing the metadata to be applied. |
| **Save metadata as default** | `DEFAULT` | [boolean] Default: False | If checked, the metadata information will be saved with the layer, hence available by default in subsequent projects. |

### Outputs

| Label | Name | Type | Description |
|---|---|---|---|
| **Output layer** | `OUTPUT` | [same as input] | The input layer with the metadata replaced by the metadata from the `.qmd` file. |

### Python code

**Algorithm ID**: `native:setlayermetadata`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 콘솔에서 공간 처리 알고리즘 사용 for details on how to run processing algorithms from the Python console.

### Set Metadata Fields

Sets various metadata fields for a layer.

### Parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Layer** | `INPUT` | [layer] | The layer whose metadata fields will be updated |
| **Identifier** Optional | `IDENTIFIER` | [string] | A unique identifier for the layer |
| **Parent identifier** Optional | `PARENT_IDENTIFIER` | [string] | The identifier of the parent resource, if applicable |
| **Title** Optional | `TITLE` | [string] | The title of the layer |
| **Type** Optional | `TYPE` | [string] | The type of data stored in the layer |
| **Language** Optional | `LANGUAGE` | [string] | The language of the metadata |
| **Encoding** Optional | `ENCODING` | [string] | The character encoding used in the metadata |
| **Abstract** Optional | `ABSTRACT` | [string] | A brief description or abstract of the layer |
| **Coordinate reference system** Optional | `CRS` | [crs] | The coordinate reference system of the layer |
| **Fees** Optional | `FEES` | [string] | Information about any fees associated with accessing the layer |
| **Ignore empty fields** | `IGNORE_EMPTY` | [boolean] Default: False | If checked, no update will be done to metadata fields that are not filled |

### Outputs

| Label | Name | Type | Description |
|---|---|---|---|
| **Output layer** | `OUTPUT` | [same as input] | The input layer with the specified metadata fields updated. |

### Python code

**Algorithm ID**: `native:setmetadatafields`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 콘솔에서 공간 처리 알고리즘 사용 for details on how to run processing algorithms from the Python console.

### Update Layer Metadata

Copies all non-empty metadata fields from a source layer to a target layer. Leaves empty input fields unchanged in the target.

### Parameters

| Label | Name | Type | Description |
|---|---|---|---|
| **Source layer** | `SOURCE` | [layer] | The layer from which non-empty metadata fields will be copied |
| **Target layer** | `TARGET` | [layer] | The layer to which non-empty metadata fields will be pasted. Empty fields in the source layer will not overwrite existing metadata in the target layer. |

### Outputs

| Label | Name | Type | Description |
|---|---|---|---|
| **Target layer** | `OUTPUT` | [layer] | The target layer with updated metadata. |

### Python code

**Algorithm ID**: `native:updatelayermetadata`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 콘솔에서 공간 처리 알고리즘 사용 for details on how to run processing algorithms from the Python console.

## 24.1.12 모델 설계자 도구

> **경고**
>
> 이 도구들은 모델 설계자에서만 사용할 수 있습니다. 공간 처리 툴박스에서 사용할 수는 없습니다.

#### 표현식 계산

QGIS 표현식의 결과를 계산해서, 동일한 결과를 한 번 이상 사용해야 할 때 모델 전반에서 동일한 표현식을 여러 번 써야 할 필요를 없애줍니다. 또한 다른 방법으로는 불가능한 활용을 가능하게 해줍니다. 예를 들면, 타임스탬프 값을 한번 생성해서 모델 안에서 여러 번 사용할 수 있습니다. 이때 타임스탬프를 매번 다시 계산한다면, 모델의 런타임 동안 값이 바뀌게 될 것입니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input** | INPUT | [expression] | 계산할 표현식 |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Value** | OUTPUT | [Result Value] | 계산한 결과값으로, 산출물의 데이터 유형은 이 알고리즘에 사용된 특정 표현식에 따라 달라질 것입니다. |

### 파이썬 코드

**알고리즘 ID**: `native:calculateexpression`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 조건부 분기

표현식 평가를 기반으로 모델의 일부가 실행될 수 있도록 모델에 조건부 분기 (conditional branch) 를 추가합니다. 대부분의 경우 모델의 흐름을 제어하기 위해 도구 의존성을 이용합니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Field** | BRANCH | [string] | 조건의 이름 |
| **Field** | CONDITION | [expression] | 평가할 표현식 |

#### 산출물

없음

#### 파이썬 코드

**알고리즘 ID**: `native:condition`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 디렉터리 생성

파일 시스템 상에 새 디렉터리를 생성합니다. 디렉터리는 재귀적으로 생성될 것입니다. 즉, 지정된 디렉터리 경로 전체를 구성하기 위해 필요한 모든 상위 디렉터리도 생성할 것입니다. 지정한 디렉터리가 이미 존재하는 경우 오류를 발생시키지 않습니다.

#### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Directory path** | `PATH` | [string] | 생성할 폴더 경로 |

#### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output** | `OUTPUT` | [folder] | 생성한 폴더 |

#### 파이썬 코드

**알고리즘 ID**: `native:createdirectory`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 피처 필터

입력 레이어에서 피처를 필터링해서 하나 또는 몇 개의 산출물로 전송 (□□) 합니다. 가능한 모든 입력 레이어에 공통된 속성 이름을 하나도 모르는 경우, `$id` 및 `uuid` 같은 피처 도형 및 일반 레코드 메커니즘에 대해서만 필터링할 수 있습니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: any] | 입력 레이어 |
| **Outputs and filters** (하나 이상) | `OUTPUT_<name of the filter>` | [same as input] | 필터링된 산출 레이어 (필터의 개수만큼) |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output** (하나 이상) | `native:filter_1:<name of filter>` | [same as input] | 필터링된 피처를 가진 산출 레이어 (필터의 개수만큼) |

### 파이썬 코드

**알고리즘 ID**: `native:filter`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 레이어 유형으로 필터링

레이어를 레이어 유형으로 필터링합니다. 입력 레이어가 벡터 레이어냐 또는 래스터 레이어냐에 따라, 입력 레이어를 서로 다른 산출물로 전송 (□□) 할 것입니다.

#### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [layer] | 일반 맵 레이어 |

#### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Vector features** 선택적 | `VECTOR` | [vector] | 호환 가능한 경우, 입력물의 벡터 레이어 |
| **Raster layer** 선택적 | `RASTER` | [raster] | 호환 가능한 경우, 입력물의 래스터 레이어 |

#### 파이썬 코드

**알고리즘 ID**: `native:filterlayersbytype`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 프로젝트로 레이어 불러오기

레이어를 현재 프로젝트로 불러옵니다.

#### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Layer** | `INPUT` | [layer] | 범례에 불러올 레이어 |
| **Loaded layer name** | `NAME` | [string] | 불러온 레이어의 이름 |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Layer** | OUTPUT | [same as input] | (재명명된) 불러온 레이어 |

### 파이썬 코드

**알고리즘 ID**: native:loadlayer

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 예외 발생

예외를 발생시켜 모델의 실행을 취소합니다. 예외 메시지를 사용자 정의할 수 있고, 표현식 기반 조건을 지정할 수 있는 옵션도 있습니다. 표현식 조건을 사용한다면, 표현식 결과가 참인 경우에만 예외를 발생시킬 것입니다. 결과가 거짓인 경우 예외를 발생시키지 않고, 모델은 방해받지 않고 계속 실행됩니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Message** | MESSAGE | [string] | 표시할 메시지 |
| **Condition**<br>선택적 | CONDITION | [expression] | 참인지 평가할 표현식 |

### 산출물

로그 패널에 메시지를 표시합니다.

### 파이썬 코드

**알고리즘 ID**: native:raiseexception

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 메시지 발생

로그에 정보 메시지를 발생시킵니다. 메시지를 사용자 정의할 수 있고, 표현식 기반 조건을 지정할 수 있는 옵션도 있습니다. 표현식 조건을 사용한다면, 표현식 결과가 참인 경우에만 메시지를 로그에 작성할 것입니다. 결과가 거짓인 경우 메시지를 발생시키지 않을 것입니다.

#### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Information message** | `MESSAGE` | [string] | 표시할 메시지 |
| **Condition** 선택적 | `CONDITION` | [expression] | 참인지 평가할 표현식 |

#### 산출물

로그 패널에 메시지를 표시합니다.

#### 파이썬 코드

**알고리즘 ID**: `native:raisemessage`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 경고 발생

로그에 경고 메시지를 발생시킵니다. 경고 메시지를 사용자 정의할 수 있고, 표현식 기반 조건을 지정할 수 있는 옵션도 있습니다. 표현식 조건을 사용한다면, 표현식 결과가 참인 경우에만 경고를 로그에 작성할 것입니다. 결과가 거짓인 경우 경고를 발생시키지 않을 것입니다.

#### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Message** | `MESSAGE` | [string] | 표시할 메시지 |
| **Condition** 선택적 | `CONDITION` | [expression] | 참인지 평가할 표현식 |

### 산출물

로그 패널에 메시지를 표시합니다.

### 파이썬 코드

**알고리즘 ID**: `native:raisewarning`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 레이어 재명명하기

레이어를 재명명합니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Layer** | `INPUT` | [layer] | 재명명할 레이어 |
| **New name** | `NAME` | [string] | 레이어의 새 이름 |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Layer** | `OUTPUT` | [same as input] | (재명명된) 산출물 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:renamelayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 로그를 파일로 저장

모델의 실행 로그를 파일로 저장합니다. 로그를 HTML 서식 버전으로 저장할 수 있는 옵션이 있습니다.

#### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Use HTML** | `USE_HTML` | [Boolean]<br>기본값: False | HTML 서식 사용 여부 |

#### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **File** | `OUTPUT` | [string] | 로그 저장 위치 |

#### 파이썬 코드

**알고리즘 ID**: `native:savelog`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 프로젝트 변수 설정

현재 프로젝트에 대해 표현식 변수를 설정합니다.

#### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Variable name** | `NAME` | [string] | 변수의 이름 |
| **Variable value** | `VALUE` | [string] | 저장할 값 |

### 산출물

없음

### 파이썬 코드

**알고리즘 ID**: `native:setprojectvariable`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 문자열 연결

공간 처리 프레임워크 모델 설계자에서 문자열 2 개를 하나로 연결 (concatenation) 합니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input 1** | `INPUT_1` | [string] | 첫 번째 문자열 |
| **Input 2** | `INPUT_2` | [string] | 두 번째 문자열 |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Concatenation** | `CONCATENATION` | [string] | 연결된 문자열 |

### 파이썬 코드

**알고리즘 ID**: `native:stringconcatenation`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 가변 거리 버퍼

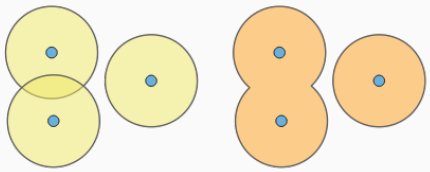
> **경고**
>
> 이 알고리즘은 중요도가 떨어져 더 이상 사용되지 않기 때문에 언제든 사라지게 될 수 있습니다. 대신버퍼 알고리즘을 사용하는 편이 좋습니다.

입력 레이어의 모든 피처에 대한 버퍼 영역을 계산합니다.

지정한 피처에 대한 버퍼 크기는 속성에 따라 정의됩니다. 따라서 서로 다른 피처의 버퍼 크기가 서로 다를 수 있습니다.

> **더 보기**
>
> 버퍼

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: any] | 입력 벡터 레이어 |
| **Distance field** | `DISTANCE` | [tablefield: numeric] | 버퍼의 거리 반경을 정의할 속성 |
| **Segments** | `SEGMENTS` | [numeric: integer]<br>기본값: *5* | 둥글린 (rounded) 오프셋을 생성하는 경우 사분원에 가깝게 만드는 데 사용할 라인 선분의 개수를 제어합니다. |
| **Dissolve result** | `DISSOLVE` | [boolean]<br>기본값: *False* | 최종 버퍼를 융해시킬지 여부를 선택할 수 있습니다. 이 옵션을 참으로 설정하면 모든 입력 피처를 커버하는 단일 피처를 산출합니다.<br>  <br>그림 24.30: 일반 버퍼와 융해된 버퍼 |
| **End cap style** | `END_CAP_STYLE` | [enumeration]<br>기본값: Round | 버퍼에서 라인의 끝을 어떻게 처리할지 제어합니다.<br>  <br>그림 24.31: 둥글림, 편평 및 정사각형 캡 스타일 |

다음 페이지에 계속

표 24.220 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Join style** | `JOIN_STYLE` | [enumeration] 기본값: Round | 라인에 있는 모서리를 오프셋시킬 때 둥글림, 마이터, 또는 베벨 결합을 사용할지 여부를 지정합니다. |
| **Miter limit** | `MITER_LIMIT` | [numeric: double] 기본값: 2.0 | 마이터 결합 스타일에만 적용할 수 있으며, 마이터 결합 생성시 사용할 오프셋 곡선으로부터의 최장 거리를 제어합니다. |

#### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Buffer** | `OUTPUT` | [vector: polygon] | 버퍼 폴리곤 벡터 레이어 |

#### 파이썬 코드

**알고리즘 ID**: `qgis:variabledistancebuffer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 24.1.13 망 분석

#### 서비스 영역 (레이어로부터)

Returns all the edges or parts of edges of a network that can be reached within a distance or a time, starting from a point layer. This allows evaluation of accessibility within a network, e.g., what are the places I can navigate to on a road network without spending cost greater than a given value (the cost can be distance or time).

> **참고**
>
> This algorithm uses ellipsoid based measurements and respects the current *ellipsoid settings*.

### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Vector layer representing network** | INPUT | [vector: line] | 커버할 망을 표현하는 라인 벡터 레이어 |
| **Vector layer with start points** | START_POINTS | [vector: point] | 서비스 영역을 생성하는 시작점으로 사용되는 피처를 가진 포인트 벡터 레이어 |
| **Path type to calculate** | STRATEGY | [enumeration] 기본값: 0 | 계산할 경로의 유형을 다음 가운데 하나로 선택할 수 있습니다:<br>• 0 — 최단 (Shortest)<br>• 1 — 최속 (Fastest) |
| **Travel cost (distance for "Shortest", time for "Fastest")** | TRAVEL_COST | [numeric: double] 기본값: 0.0 | 최단 경로를 탐색하는 경우 (망 레이어 단위의) 거리로 추정한 값, 최속 경로의 경우 (시 단위의) 시간으로 추청한 값 |
| **Service area (lines)** | OUTPUT_LINES | [vector: line] 기본값: [Create temporary layer] | Specify the output line layer for the service area. *One of*:<br>• 산출물 건너뛰기<br>• 임시 레이어 생성 (TEMPORARY_OUTPUT)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |
| **Service area (boundary nodes)** | OUTPUT | [vector: point] 기본값: [Skip output] | Specify the output point layer for the service area boundary nodes. *One of*:<br>• 산출물 건너뛰기<br>• 임시 레이어 생성 (TEMPORARY_OUTPUT)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |
| **Non-routable features** | OUTPUT_NON_ROUTABI | [vector: point] 기본값: [Skip output] | Specify the output which will be used to store any input features which could not be routed (e.g., those which are too far from the network layer). *One of*:<br>• 산출물 건너뛰기<br>• 임시 레이어 생성 (TEMPORARY_OUTPUT)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Direction field**<br>선택적 | `DIRECTION_FIELD` | [tablefield: any] | 망 경계에 대해 방향을 지정하는 데 사용하는 필드입니다.<br>The values used in this field are specified with the three parameters `Value for forward direction`, `Value for backward direction` and `Value for both directions`. Forward and reverse directions correspond to a one-way edge, "both directions" indicates a two-way edge. If a feature does not have a value in this field, or no field is set then the default direction setting (provided with the `Default direction` parameter) is used. |
| **Value for forward direction**<br>선택적 | `VALUE_FORWARD` | [string]<br>기본값: `''` (empty string) | 순방향 경계를 식별하기 위해 방향 필드에서 설정한 값 |
| **Value for backward direction**<br>선택적 | `VALUE_BACKWARD` | [string]<br>기본값: `''` (empty string) | 역방향 경계를 식별하기 위해 방향 필드에서 설정한 값 |
| **Value for both directions**<br>선택적 | `VALUE_BOTH` | [string]<br>기본값: `''` (empty string) | 양방향 경계를 식별하기 위해 방향 필드에서 설정한 값 |
| **Default direction** | `DEFAULT_DIRECTION` | [enumeration]<br>기본값: 2 | 피처가 방향 필드에 설정된 값을 가지고 있지 않거나 또는 방향 필드를 설정하지 않은 경우, 이 방향 값을 사용합니다. 다음 가운데 하나를 선택할 수 있습니다:<br>• 0 —순방향<br>• 1 —역방향<br>• 2 —양방향 |
| **Speed field**<br>선택적 | `SPEED_FIELD` | [tablefield: numeric] | 최속 경로를 탐색하는 경우 망의 경계의 (`km/h` 단위의) 속도 값을 지정하는 필드<br>피처가 이 필드에 값을 가지고 있지 않거나 또는 아무 필드도 설정하지 않은 경우, (`Default speed` 파라미터로 지정한) 기본 속도 값을 사용합니다. |
| **Default speed (km/h)** | `DEFAULT_SPEED` | [numeric: double]<br>Default: 50.0 | Value to use to calculate the travel time if no speed value is provided for an edge in the specified field |
| **Topology tolerance** | `TOLERANCE` | [numeric: double]<br>기본값: 0.0 | 지정한 허용 오차보다 가까이 있는 노드를 가진 라인 2 개를 연결된 것으로 간주합니다. |

| | | | |
|---|---|---|---|
| **Include upper/lower bound points** | INCLUDE_BOUNDS | [boolean]<br>기본값: False | 서비스 영역의 경계에 있는 망 경계를 위한 포인트 2 개를 가진 산출 포인트 레이어를 생성합니다. 한 포인트는 해당 망 경계의 시작점, 다른 포인트는 종단점입니다. |
| **Maximum point distance from network**<br>선택적 | POINT_TOLERANCE | [numeric: double]<br>기본값: 설정하지 않음 | Specifies an optional limit on the distance from the points to the network layer. If a point is further from the network than this distance it will be treated as non-routable. If not set, endpoints will be snapped to the nearest point on the network layer, regardless of how far away from the network they actually are. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Service area (boundary nodes)** | OUTPUT | [vector: point] | 서비스 영역의 경계 노드를 담은 산출 포인트 레이어 |
| **Service area (lines)** | OUTPUT_LINES | [vector: line] | 시작점에서 지정한 비용으로 서비스할 수 있는 망의 부분을 표현하는 라인 레이어 |
| **Non routable features** | OUTPUT_NON_ROUTABI | [vector: point] | An optional output which will be used to store any input features which could not be routed (e.g., those which are too far from the network layer). |

### 파이썬 코드

**Algorithm ID**: `native:serviceareafromlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 서비스 영역 (포인트로부터)

Returns all the edges or parts of edges of a network that can be reached within a given distance or time, starting from a point feature. This allows the evaluation of accessibility within a network, e.g., what are the places I can navigate to on a road network without spending a cost greater than a given value (the cost can be distance or time).

> **참고**
>
> This algorithm uses ellipsoid based measurements and respects the current *ellipsoid settings*.

### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Vector layer representing the network** | `INPUT` | [vector: line] | 커버할 망을 표현하는 라인 벡터 레이어 |
| **Path type to calculate** | `STRATEGY` | [enumeration]<br>기본값: 0 | 계산할 경로의 유형을 다음 가운데 하나로 선택할 수 있습니다:<br>• 0 — 최단 (Shortest)<br>• 1 — 최속 (Fastest) |
| **Start point** | `START_POINT` | [coordinates] | 그 주위에 서비스 영역을 계산할 포인트의 좌표<br>Press the ⋯ button next to the option and click on the canvas to fill the parameter with the clicked point coordinate. |
| **Travel cost (distance for "Shortest", time for "Fastest")** | `TRAVEL_COST` | [numeric: double]<br>기본값: 0.0 | 최단 경로를 탐색하는 경우 (망 레이어 단위의) 거리로 추정한 값, 최속 경로의 경우 (시 단위의) 시간으로 추청한 값 |
| **Service area (lines)**<br>선택적 | `OUTPUT_LINES` | [vector: line]<br>기본값: `[Create temporary layer]` | Specify the output line layer for the service area. *One of*:<br>• 산출물 건너뛰기<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |
| **Service area (boundary nodes)**<br>선택적 | `OUTPUT` | [vector: point]<br>기본값: `[Skip output]` | Specify the output point layer for the service area boundary nodes. *One of*:<br>• 산출물 건너뛰기<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Direction field**<br>선택적 | `DIRECTION_FIELD` | [tablefield: any] | 망 경계에 대해 방향을 지정하는 데 사용하는 필드입니다.<br>The values used in this field are specified with the three parameters `Value for forward direction`, `Value for backward direction` and `Value for both directions`. Forward and reverse directions correspond to a one-way edge, "both directions" indicates a two-way edge. If a feature does not have a value in this field, or no field is set then the default direction setting (provided with the `Default direction` parameter) is used. |
| **Value for forward direction**<br>선택적 | `VALUE_FORWARD` | [string]<br>기본값: `''` (empty string) | 순방향 경계를 식별하기 위해 방향 필드에서 설정한 값 |
| **Value for backward direction**<br>선택적 | `VALUE_BACKWARD` | [string]<br>기본값: `''` (empty string) | 역방향 경계를 식별하기 위해 방향 필드에서 설정한 값 |
| **Value for both directions**<br>선택적 | `VALUE_BOTH` | [string]<br>기본값: `''` (empty string) | 양방향 경계를 식별하기 위해 방향 필드에서 설정한 값 |
| **Default direction** | `DEFAULT_DIRECTION` | [enumeration]<br>기본값: 2 | 피처가 방향 필드에 설정된 값을 가지고 있지 않거나 또는 방향 필드를 설정하지 않은 경우, 이 방향 값을 사용합니다. 다음 가운데 하나를 선택할 수 있습니다:<br>• 0 —순방향<br>• 1 —역방향<br>• 2 —양방향 |
| **Speed field**<br>선택적 | `SPEED_FIELD` | [tablefield: numeric] | 최속 경로를 탐색하는 경우 망의 경계의 (`km/h` 단위의) 속도 값을 지정하는 필드<br>피처가 이 필드에 값을 가지고 있지 않거나 또는 아무 필드도 설정하지 않은 경우, (`Default speed` 파라미터로 지정한) 기본 속도 값을 사용합니다. |
| **Default speed (km/h)** | `DEFAULT_SPEED` | [numeric: double]<br>Default: 50.0 | Value to use to calculate the travel time if no speed value is provided for an edge in the specified field |
| **Topology tolerance** | `TOLERANCE` | [numeric: double]<br>기본값: 0.0 | 지정한 허용 오차보다 가까이 있는 노드를 가진 라인 2 개를 연결된 것으로 간주합니다. |

| | | | |
|---|---|---|---|
| **Maximum point distance from network**<br>선택적 | POINT_TOLERANCE | [numeric: double]<br>기본값: 0.0 | Specifies an optional limit on the distance from the start point to the network layer. If the point is further from the network than this distance an error will be raised. If not set, the point will be snapped to the nearest point on the network layer, regardless of how far away from the network it actually is. |
| **Include upper/lower bound points** | INCLUDE_BOUNDS | [boolean]<br>기본값: False | 서비스 영역의 경계에 있는 망 경계를 위한 포인트 2 개를 가진 산출 포인트 레이어를 생성합니다. 한 포인트는 해당 망 경계의 시작점, 다른 포인트는 종단점입니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Service area (boundary nodes)** | OUTPUT | [vector: point] | 서비스 영역의 경계 노드를 담은 산출 포인트 레이어 |
| **Service area (lines)** | OUTPUT_LINES | [vector: line] | 시작점에서 지정한 비용으로 서비스할 수 있는 망의 부분을 표현하는 라인 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:serviceareafrompoint`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 최단 경로 (레이어에서 포인트로)

벡터 레이어와 지정한 종단점으로 정의된 여러 시작점들로부터 최적 (최단 또는 최속) 경로를 계산합니다.

**참고**

This algorithm uses ellipsoid based measurements and respects the current *ellipsoid settings*.

### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Vector layer representing network** | INPUT | [vector: line] | 커버할 망을 표현하는 라인 벡터 레이어 |
| **Path type to calculate** | STRATEGY | [enumeration] 기본값: 0 | 계산할 경로의 유형을 다음 가운데 하나로 선택할 수 있습니다:<br>• 0 — 최단 (Shortest)<br>• 1 — 최속 (Fastest) |
| **Vector layer with start points** | START_POINTS | [vector: point] | 경로의 시작점으로 사용되는 피처를 가진 포인트 벡터 레이어 |
| **End point (x, y)** | END_POINT | [coordinates] | 경로의 종단점을 표현하는 포인트 피처<br>Press the ⋯ button next to the option and click on the canvas to fill the parameter with the clicked point coordinate. |
| **Shortest path** | OUTPUT | [vector: line] 기본값: [Create temporary layer] | Specify the output line layer for the shortest paths. *One of*:<br>• 임시 레이어 생성 (TEMPORARY_OUTPUT)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |
| **Non-routable features** 선택적 | OUTPUT_NON_ROUTABI | [vector: point] 기본값: [Skip output] | Specify the output which will be used to store any input features which could not be routed (e.g., those which are too far from the network layer). *One of*:<br>• 산출물 건너뛰기<br>• 임시 레이어 생성 (TEMPORARY_OUTPUT)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

#### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Direction field**<br>선택적 | DIRECTION_FIELD | [tablefield: any] | 망 경계에 대해 방향을 지정하는 데 사용하는 필드입니다.<br>The values used in this field are specified with the three parameters Value for forward direction, Value for backward direction and Value for both directions. Forward and reverse directions correspond to a one-way edge, "both directions" indicates a two-way edge. If a feature does not have a value in this field, or no field is set then the default direction setting (provided with the Default direction parameter) is used. |
| **Value for forward direction**<br>선택적 | VALUE_FORWARD | [string]<br>기본값: '' (empty string) | 순방향 경계를 식별하기 위해 방향 필드에서 설정한 값 |
| **Value for backward direction**<br>선택적 | VALUE_BACKWARD | [string]<br>기본값: '' (empty string) | 역방향 경계를 식별하기 위해 방향 필드에서 설정한 값 |
| **Value for both directions**<br>선택적 | VALUE_BOTH | [string]<br>기본값: '' (empty string) | 양방향 경계를 식별하기 위해 방향 필드에서 설정한 값 |
| **Default direction** | DEFAULT_DIRECTION | [enumeration]<br>기본값: 2 | 피처가 방향 필드에 설정된 값을 가지고 있지 않거나 또는 방향 필드를 설정하지 않은 경우, 이 방향 값을 사용합니다. 다음 가운데 하나를 선택할 수 있습니다:<br>• 0 —순방향<br>• 1 —역방향<br>• 2 —양방향 |
| **Speed field**<br>선택적 | SPEED_FIELD | [tablefield: numeric] | 최속 경로를 탐색하는 경우 망의 경계의 (km/h 단위의) 속도 값을 지정하는 필드<br>피처가 이 필드에 값을 가지고 있지 않거나 또는 아무 필드도 설정하지 않은 경우, (Default speed 파라미터로 지정한) 기본 속도 값을 사용합니다. |
| **Default speed (km/h)** | DEFAULT_SPEED | [numeric: double]<br>Default: 50.0 | Value to use to calculate the travel time if no speed value is provided for an edge in the specified field |
| **Topology tolerance** | TOLERANCE | [numeric: double]<br>기본값: 0.0 | 지정한 허용 오차보다 가까이 있는 노드를 가진 라인 2 개를 연결된 것으로 간주합니다. |

| | | | |
|---|---|---|---|
| **Maximum point distance from network**<br>선택적 | POINT_TOLERANCE | [numeric: double]<br>기본값: 설정하지 않음 | Specifies an optional limit on the distance from the start and end points to the network layer. If a start feature is further from the network than this distance it will be treated as non-routable. If the end point is further from the network than this distance an error will be raised. If not set, points will be snapped to the nearest point on the network layer, regardless of how far away from the network they actually are. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Shortest path** | OUTPUT | [vector: line] | 각 시작점부터 종단점까지의 최단 또는 최속 경로를 담은 라인 레이어 |
| **Non routable features** | OUTPUT_NON_ROUTABI | [vector: point] | An optional output layer which will be used to store any input features which could not be routed (e.g., those which are too far from the network layer). |

### 파이썬 코드

**알고리즘 ID**: `native:shortestpathlayertopoint`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 최단 경로 (포인트에서 레이어로)

지정한 시작점과 포인트 벡터 레이어로 정의한 여러 종단점 사이의 최적 (최단 또는 최속) 경로를 계산합니다.

**참고**

This algorithm uses ellipsoid based measurements and respects the current *ellipsoid settings*.

**경고**

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Vector layer representing network** | INPUT | [vector: line] | 커버할 망을 표현하는 라인 벡터 레이어 |
| **Path type to calculate** | STRATEGY | [enumeration] 기본값: 0 | 계산할 경로의 유형을 다음 가운데 하나로 선택할 수 있습니다:<br>• 0 — 최단 (Shortest)<br>• 1 — 최속 (Fastest) |
| **Start point** | START_POINT | [coordinates] | 경로의 시작점을 표현하는 포인트 피처<br>Press the ⋯ button next to the option and click on the canvas to fill the parameter with the clicked point coordinate. |
| **Vector layer with end points** | END_POINTS | [vector: point] | 경로의 종단점으로 사용되는 피처를 가진 포인트 벡터 레이어 |
| **Shortest path** | OUTPUT | [vector: line] 기본값: [Create temporary layer] | Specify the output line layer for the shortest paths. *One of*:<br>• 임시 레이어 생성 (TEMPORARY_OUTPUT)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |
| **Non-routable features** | OUTPUT_NON_ROUTABI 선택적 | [vector: point] 기본값: [Skip output] | Specify the output which will be used to store any input features which could not be routed (e.g., those which are too far from the network layer). *One of*:<br>• 산출물 건너뛰기<br>• 임시 레이어 생성 (TEMPORARY_OUTPUT)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

#### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Direction field**<br>선택적 | DIREC-<br>TION_FIELD | [tablefield: any] | 망 경계에 대해 방향을 지정하는 데 사용하는 필드입니다.<br>The values used in this field are specified with the three parameters `Value for forward direction`, `Value for backward direction` and `Value for both directions`. Forward and reverse directions correspond to a one-way edge, "both directions" indicates a two-way edge. If a feature does not have a value in this field, or no field is set then the default direction setting (provided with the `Default direction` parameter) is used. |
| **Value for forward direction**<br>선택적 | VALUE_FORWARD | [string]<br>기본값: '' (empty string) | 순방향 경계를 식별하기 위해 방향 필드에서 설정한 값 |
| **Value for backward direction**<br>선택적 | VALUE_BACKWARD | [string]<br>기본값: '' (empty string) | 역방향 경계를 식별하기 위해 방향 필드에서 설정한 값 |
| **Value for both directions**<br>선택적 | VALUE_BOTH | [string]<br>기본값: '' (empty string) | 양방향 경계를 식별하기 위해 방향 필드에서 설정한 값 |
| **Default direction** | DE-<br>FAULT_DIRECTION | [enumeration]<br>기본값: 2 | 피처가 방향 필드에 설정된 값을 가지고 있지 않거나 또는 방향 필드를 설정하지 않은 경우, 이 방향 값을 사용합니다. 다음 가운데 하나를 선택할 수 있습니다:<br>• 0 —순방향<br>• 1 —역방향<br>• 2 —양방향 |
| **Speed field**<br>선택적 | SPEED_FIELD | [tablefield: numeric] | 최속 경로를 탐색하는 경우 망의 경계의 (`km/h` 단위의) 속도 값을 지정하는 필드<br>피처가 이 필드에 값을 가지고 있지 않거나 또는 아무 필드도 설정하지 않은 경우, (`Default speed` 파라미터로 지정한) 기본 속도 값을 사용합니다. |
| **Default speed (km/h)** | DEFAULT_SPEED | [numeric: double]<br>Default: 50.0 | Value to use to calculate the travel time if no speed value is provided for an edge in the specified field |
| **Topology tolerance** | TOLERANCE | [numeric: double]<br>기본값: 0.0 | 지정한 허용 오차보다 가까이 있는 노드를 가진 라인 2 개를 연결된 것으로 간주합니다. |

| | | | |
|---|---|---|---|
| **Maximum point distance from network**<br>선택적 | POINT_TOLERANCE | [numeric: double]<br>기본값: 설정하지 않음 | Specifies an optional limit on the distance from the start and end points to the network layer. If the start point is further from the network than this distance an error will be raised. If an end feature is further from the network than this distance it will be treated as non-routable. If not set, points will be snapped to the nearest point on the network layer, regardless of how far away from the network they actually are. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Shortest path** | OUTPUT | [vector: line] | 각 시작점부터 종단점까지의 최단 또는 최속 경로를 담은 라인 레이어 |
| **Non routable features** | OUTPUT_NON_ROUTABI | [vector: point] | An optional output layer which will be used to store any input features which could not be routed (e.g., those which are too far from the network layer). |

### 파이썬 코드

**알고리즘 ID**: `native:shortestpathpointtolayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 최단 경로 (포인트에서 포인트로)

지정한 시작점 및 종단점 사이의 최적 (최단 또는 최속) 경로를 계산합니다.

> **참고**
>
> This algorithm uses ellipsoid based measurements and respects the current *ellipsoid settings*.

### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Vector layer representing network** | INPUT | [vector: line] | 커버할 망을 표현하는 라인 벡터 레이어 |
| **Path type to calculate** | STRATEGY | [enumeration] 기본값: 0 | 계산할 경로의 유형을 다음 가운데 하나로 선택할 수 있습니다:<br>• 0 — 최단 (Shortest)<br>• 1 — 최속 (Fastest) |
| **Start point (x, y)** | START_POINT | [coordinates] | Point feature representing the start point of the routes.<br>Press the ⋯ button next to the option and click on the canvas to fill the parameter with the clicked point coordinate. |
| **End point (x, y)** | END_POINT | [coordinates] | Point feature representing the end point of the routes.<br>Press the ⋯ button next to the option and click on the canvas to fill the parameter with the clicked point coordinate. |
| **Shortest path** | OUTPUT | [vector: line] | Specify the output line layer for the shortest paths. *One of*:<br>• 임시 레이어 생성 (TEMPORARY_OUTPUT)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

#### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Direction field**<br>선택적 | DIRECTION_FIELD | [tablefield: any] | 망 경계에 대해 방향을 지정하는 데 사용하는 필드입니다.<br>The values used in this field are specified with the three parameters Value for forward direction, Value for backward direction and Value for both directions. Forward and reverse directions correspond to a one-way edge, "both directions" indicates a two-way edge. If a feature does not have a value in this field, or no field is set then the default direction setting (provided with the Default direction parameter) is used. |

다음 페이지에 계속

표 24.235 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Value for forward direction**<br>선택적 | VALUE_FORWARD | [string]<br>기본값: '' (empty string) | 순방향 경계를 식별하기 위해 방향 필드에서 설정한 값 |
| **Value for backward direction**<br>선택적 | VALUE_BACKWARD | [string]<br>기본값: '' (empty string) | 역방향 경계를 식별하기 위해 방향 필드에서 설정한 값 |
| **Value for both directions**<br>선택적 | VALUE_BOTH | [string]<br>기본값: '' (empty string) | 양방향 경계를 식별하기 위해 방향 필드에서 설정한 값 |
| **Default direction** | DEFAULT_DIRECTION | [enumeration]<br>기본값: 2 | 피처가 방향 필드에 설정된 값을 가지고 있지 않거나 또는 방향 필드를 설정하지 않은 경우, 이 방향 값을 사용합니다. 다음 가운데 하나를 선택할 수 있습니다:<br>• 0 —순방향<br>• 1 —역방향<br>• 2 —양방향 |
| **Speed field**<br>선택적 | SPEED_FIELD | [tablefield: numeric] | 최속 경로를 탐색하는 경우 망의 경계의 (km/h 단위의) 속도 값을 지정하는 필드<br>피처가 이 필드에 값을 가지고 있지 않거나 또는 아무 필드도 설정하지 않은 경우, (Default speed 파라미터로 지정한) 기본 속도 값을 사용합니다. |
| **Default speed (km/h)** | DEFAULT_SPEED | [numeric: double]<br>Default: 50.0 | Value to use to calculate the travel time if no speed value is provided for an edge in the specified field |
| **Topology tolerance** | TOLERANCE | [numeric: double]<br>기본값: 0.0 | 지정한 허용 오차보다 가까이 있는 노드를 가진 라인 2 개를 연결된 것으로 간주합니다. |
| **Maximum point distance from network**<br>선택적 | POINT_TOLERANCE | [numeric: double]<br>기본값: 설정하지 않음 | Specifies an optional limit on the distance from the start and end points to the network layer. If either point is further from the network than this distance an error will be raised. If not set, points will be snapped to the nearest point on the network layer, regardless of how far away from the network they actually are. |

#### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Shortest path** | `OUTPUT` | [vector: line] | 각 시작점부터 종단점까지의 최단 또는 최속 경로를 담은 라인 레이어 |

#### 파이썬 코드

**알고리즘 ID**: `native:shortestpathpointtopoint`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 24.1.14 도표

#### 막대 그래프

범주 및 레이어 필드에서 막대 그래프 (bar plot) 를 색성합니다.

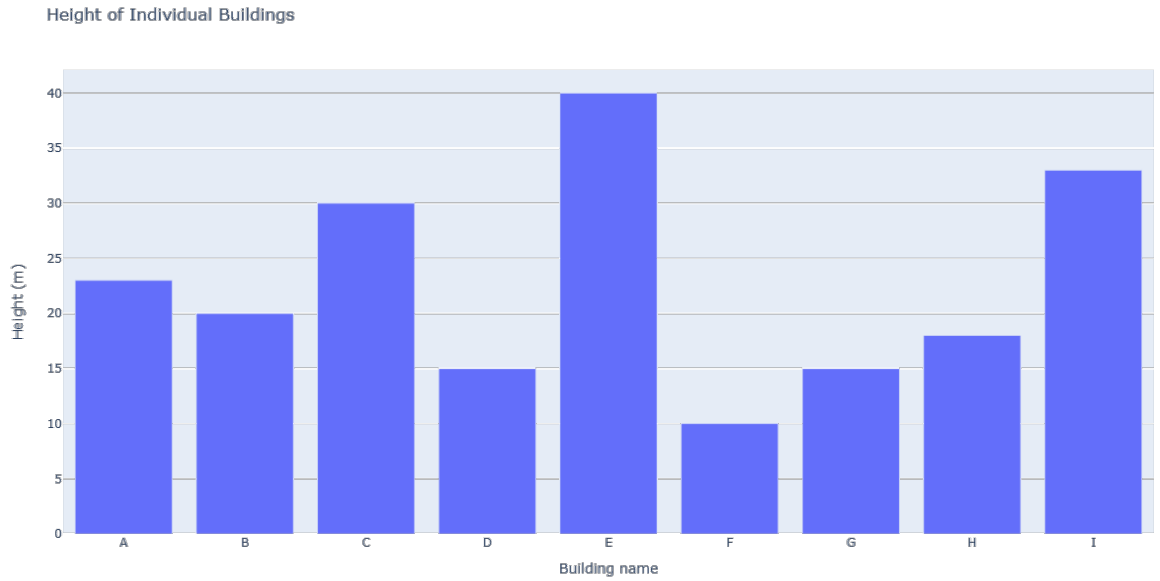



그림 24.32: Bar plot example

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | INPUT | [vector: geometry] | 입력 벡터 레이어 |
| **Category field name** | NAME_FIELD | [tablefield: any] | (X 축) 막대를 그룹화하는 데 사용할 범주 필드 |
| **Value field** | VALUE_FIELD | [tablefield: numeric] | (Y 축) 그래프에 사용할 값 |
| **Title** Optional | TITLE | [string] Default: “” | Title of the plot |
| **X-axis Title** Optional | XAXIS_TITLE | [string] Default: “” | If empty, the name of the category field is used. With a single space, the axis title is hidden. |
| **Y-axis Title** Optional | YAXIS_TITLE | [string] Default: “” | If empty, the name of the value field is used. With a single space, the axis title is hidden. |
| **Bar plot** | OUTPUT | [html] 기본값: [Save to temporary file] | Specify the HTML file for the plot. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Bar plot** | OUTPUT | [html] | 그래프 HTML 파일. *Processing* → *Result Viewer* 메뉴로 볼 수 있습니다. |

### 파이썬 코드

**알고리즘 ID**: `qgis:barplot`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 상자 수염 그림

범주 필드 및 숫자형 레이어 필드에서 상자 수염 그림 (box plot) 을 생성합니다.

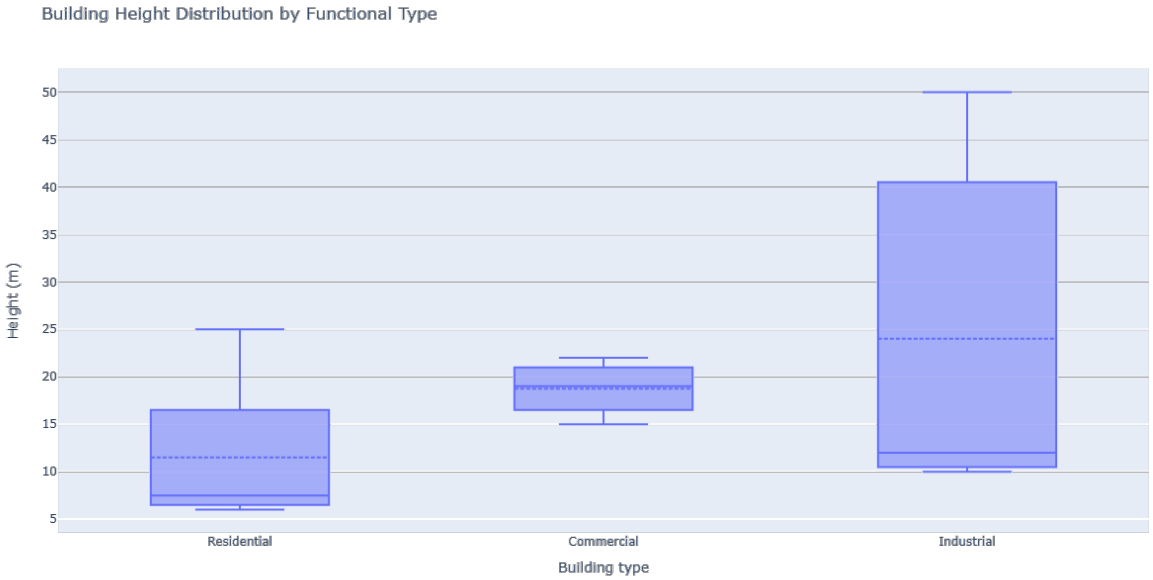



그림 24.33: Example of box plot

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | 입력 벡터 레이어 |
| **Category name field** | `NAME_FIELD` | [tablefield: any] | (X 축) 상자를 그룹화하는 데 사용할 범주 필드 |
| **Value field** | `VALUE_FIELD` | [tablefield: numeric] | (Y 축) 그래프에 사용할 값 |
| **Additional statistic lines** | `MSD` | [enumeration]<br>기본값: 0 | 그래프에 추가할 부가적인 통계 정보입니다. 다음 가운데 하나를 선택할 수 있습니다:<br>• 0 —평균 표시<br>• 1 —표준 편차 표시<br>• 2 —평균 및 표준 편차를 표시하지 않음 |
| **Title**<br>Optional | `TITLE` | [string]<br>Default: “” | Title of the plot |
| **X-axis Title**<br>Optional | `XAXIS_TITLE` | [string]<br>Default: “” | If empty, the name of the category field is used. With a single space, the axis title is not shown. |
| **Y-axis Title**<br>Optional | `YAXIS_TITLE` | [string]<br>Default: “” | If empty, the name of the value field is used. With a single space, the axis title is not shown. |
| **Box plot** | `OUTPUT` | [html]<br>기본값: `[Save to temporary file]` | Specify the HTML file for the plot. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Box plot** | `OUTPUT` | [html] | 그래프 HTML 파일. *Processing* → *Result Viewer* 메뉴로 볼 수 있습니다. |

### Generate elevation profile image

Creates an elevation profile image from a list of map layers and an optional terrain layer.

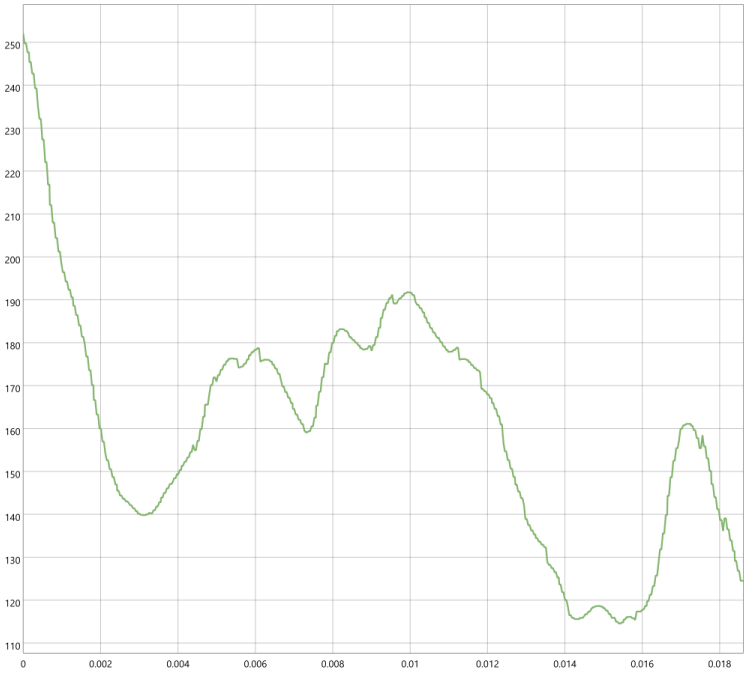



그림 24.34: Example of elevation profile

#### 파라미터

### Basic parameters

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Profile curve** | `CURVE` | [geometry: line] | The LineString geometry (WKT or GeoJSON) representing the profile curve along which the elevation profile will be generated |
| **Map layers** | `MAP_LAYERS` | [layer] [list] | The list of map layers to include in the elevation profile |
| **Chart width (in pixels)** | `WIDTH` | [numeric: integer]<br>Default: 400<br>Minimum value: 0 | The width of the output chart in pixels. |
| **Chart height (in pixels)** | `HEIGHT` | [numeric: integer]<br>Default: 300<br>Minimum value: 0 | The height of the output chart in pixels. |
| **Terrain layer**<br>Optional | `TERRAIN_LAYER` | [raster] | A terrain layer (e.g., DEM) to use for elevation data. If not provided, elevation data will be derived from the map layers. |

### Advanced parameters

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Chart minimum distance (X axis)** Optional | `MINIMUM_DISTANCE` | [numeric: double] | The minimum distance (X axis) to display on the chart. If not specified, the chart will auto-scale. |
| **Chart maximum distance (X axis)** Optional | `MAXIMUM_DISTANCE` | [numeric: double] | The maximum distance (X axis) to display on the chart. If not specified, the chart will auto-scale. |
| **Chart minimum elevation (Y axis)** Optional | `MINIMUM_ELEVATION` | [numeric: double] | The minimum elevation (Y axis) to display on the chart. If not specified, the chart will auto-scale. |
| **Chart maximum elevation (Y axis)** Optional | `MAXIMUM_ELEVATION` | [numeric: double] | The maximum elevation (Y axis) to display on the chart. If not specified, the chart will auto-scale. |
| **Chart text color** Optional | `TEXT_COLOR` | [color] | The color of the text in the chart (e.g., axis labels, titles). |
| **Chart background color** Optional | `BACKGROUND_COLOR` | [color] | The background color of the chart. |
| **Chart border color** Optional | `BORDER_COLOR` | [color] | The color of the chart border. |
| **Profile tolerance** | `TOLERANCE` | [numeric: double] Default: 5.0 Minimum value: 0 | Defines how far a feature (vector point, line, polygon, or point cloud) can be from the profile line to be included in the results. It uses map units and does not affect other layer types. |
| **Chart DPI** | `DPI` | [numeric: integer] Default: 96 Minimum value: 0 | The resolution of the output image in dots per inch (DPI). |
| **Output image** | `OUTPUT` | [file] 기본값: `[Save to temporary file]` | Specify the image file for the plot. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output image** | `OUTPUT` | [file] | The generated elevation profile image in the specified format. |

### 파이썬 코드

**Algorithm ID**: `native:generateelevationprofileimage`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 평균 및 표준 편차 그래프

평균 및 표준 편차를 표시하는 상자 수염 그림을 생성합니다.

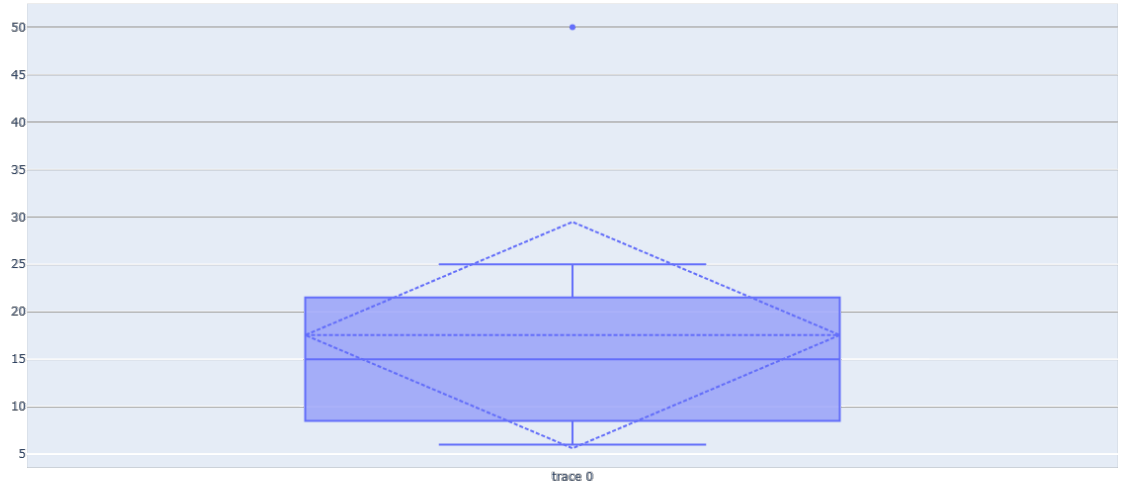



그림 24.35: Example of mean and standard deviation plot

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input table** | `INPUT` | [vector: geometry] | 입력 벡터 레이어 |
| **Category name field** | `NAME_FIELD` | [tablefield: any] | (X 축) 상자를 그룹화하는 데 사용할 범주 필드 |
| **Value field** | `VALUE_FIELD` | [tablefield: numeric] | (Y 축) 그래프에 사용할 값 |
| **Plot** | `OUTPUT` | [html]<br>기본값: `[Save to temporary file]` | Specify the HTML file for the plot. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Plot** | `OUTPUT` | [html] | 그래프 HTML 파일. *Processing* → *Result Viewer* 메뉴로 볼 수 있습니다. |

### 파이썬 코드

**알고리즘 ID**: `qgis:meanandstandarddeviationplot`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 극좌표 그래프

입력 벡터 레이어의 값을 기반으로 극좌표 그래프 (polar plot) 를 생성합니다.

필드 2 개를 파라미터로 입력해야만 합니다. 각 피처에 대해 (피처를 그룹화하기 위한) 범주를 정의하는 필드와 그래프를 그리기 위한 (숫자형이어야 하는) 변수를 가진 필드입니다.

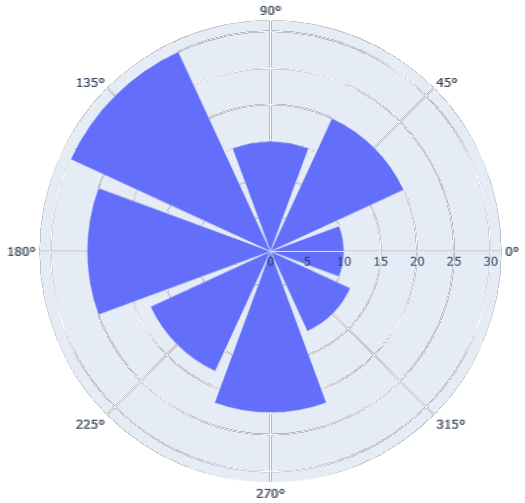



그림 24.36: Example of polar plot

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | INPUT | [vector: geometry] | 입력 벡터 레이어 |
| **Category name field** | NAME_FIELD | [tablefield: any] | (X 축) 피처를 그룹화하는 데 사용할 범주 필드 |
| **Value field** | VALUE_FIELD | [tablefield: numeric] | (Y 축) 그래프에 사용할 값 |
| **Polar plot** | OUTPUT | [html]<br>기본값: [Save to temporary file] | Specify the HTML file for the plot. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Polar plot** | OUTPUT | [html] | 그래프 HTML 파일. *Processing* → *Result Viewer* 메뉴로 볼 수 있습니다. |

### 파이썬 코드

**알고리즘 ID**: `qgis:polarplot`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 래스터 레이어 히스토그램

래스터 레이어의 값으로 히스토그램을 생성합니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | INPUT | [raster] | 입력 래스터 레이어 |
| **Band number** | BAND | [raster band] | 히스토그램에 사용할 래스터 밴드 |
| **number of bins** | BINS | [numeric: integer]<br>기본값: 10 | 히스토그램에 사용할 (X 축) 계급 간격 (bin) 의 개수. 최소값은 2 입니다. |
| **Histogram** | OUTPUT | [html]<br>기본값: [Save to temporary file] | Specify the HTML file for the plot. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

다음 페이지에 계속

표 24.248 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Histogram** | `OUTPUT` | [html] | 그래프 HTML 파일. *Processing → Result Viewer* 메뉴로 볼 수 있습니다. |

### 파이썬 코드

**알고리즘 ID**: `qgis:rasterlayerhistogram`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 벡터 레이어 히스토그램

벡터 레이어의 속성값으로 히스토그램을 생성합니다.

히스토그램을 계산하는 데 사용할 속성은 숫자형이어야만 합니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: any] | 입력 벡터 레이어 |
| **Attribute** | `FIELD` | [tablefield: numeric] | (Y 축) 그래프에 사용할 값 |
| **number of bins** | `BINS` | [numeric: integer]<br>기본값: 10 | 히스토그램에 사용할 (X 축) 계급 간격 (bin) 의 개수. 최소값은 2 입니다. |
| **Histogram** | `OUTPUT` | [html]<br>기본값: `[Save to temporary file]` | Specify the HTML file for the plot. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

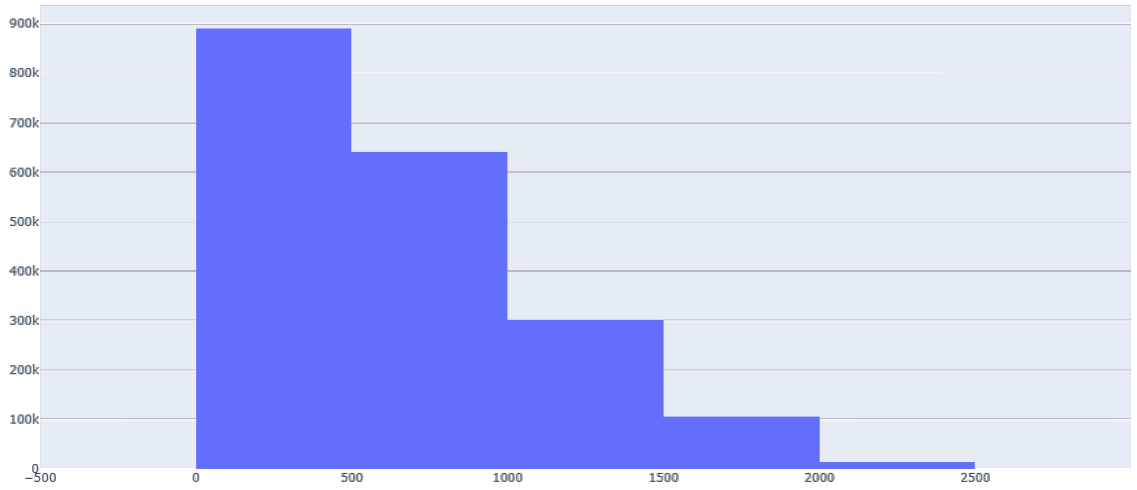



그림 24.37: Raster layer histogram example

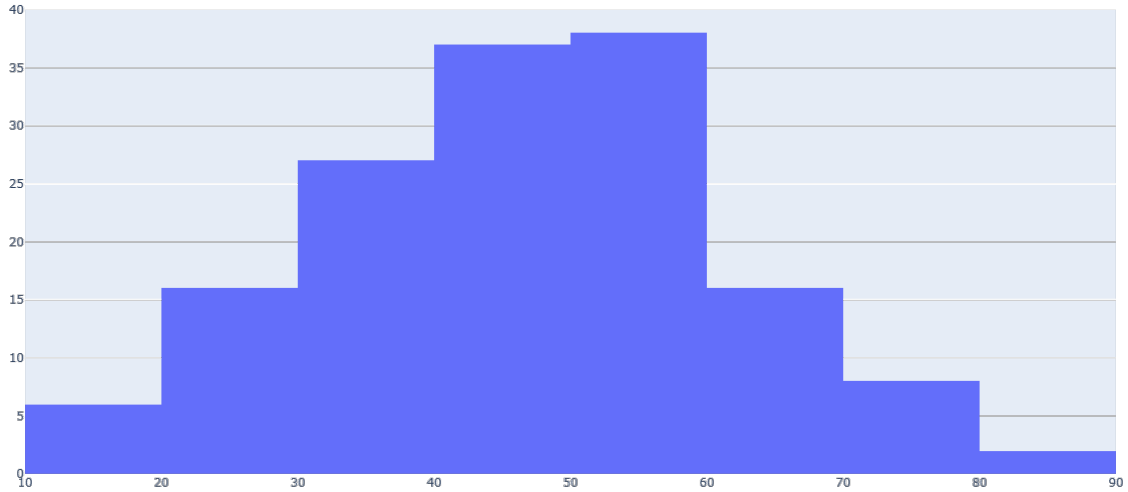



그림 24.38: Example of vector layer histogram

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Histogram** | `OUTPUT` | [html] | 그래프 HTML 파일. *Processing → Result Viewer* 메뉴로 볼 수 있습니다. |

### 파이썬 코드

**알고리즘 ID**: `qgis:vectorlayerhistogram`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 벡터 레이어 산점도

벡터 레이어에 대해 단순 `X` - `Y` 산점도 (scatter plot) 를 생성합니다.

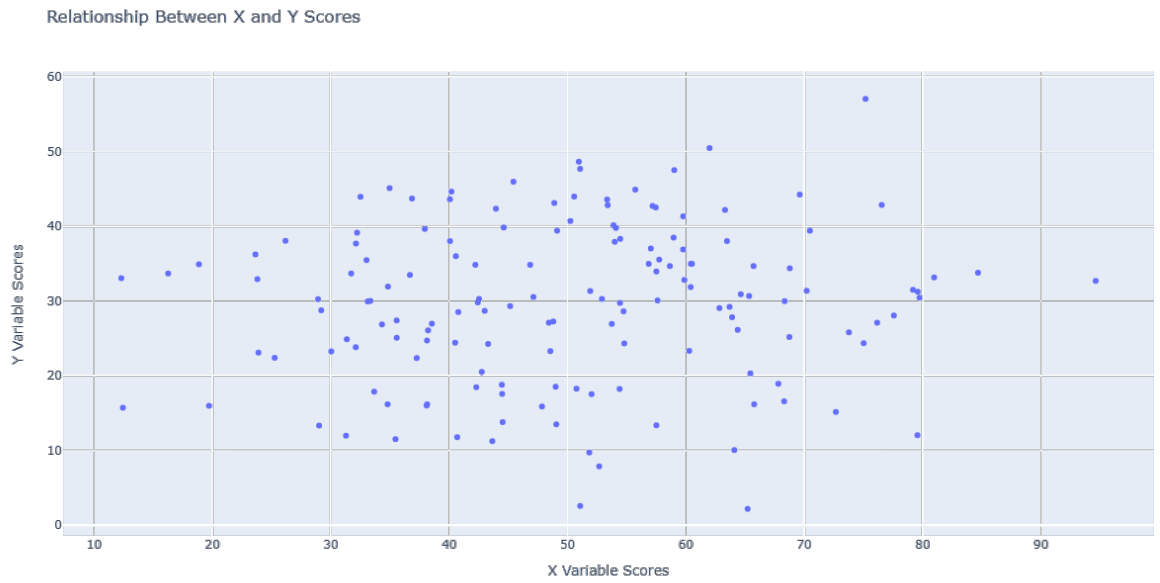



그림 24.39: Example of vector layer scatterplot

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | 입력 벡터 레이어 |
| **X attribute** | `XFIELD` | [tablefield: numeric] | X 축에 사용할 필드 |
| **Y attribute** | `YFIELD` | [tablefield: numeric] | Y 축에 사용할 필드 |
| **Hover text** Optional | `HOVERTEXT` | [expression] Default: “” | Text to be shown when hovering with the mouse over data points. It can be picked from a field or based on an expression. |
| **Title** Optional | `TITLE` | [string] Default: “” | Title of the plot |
| **X-axis Title** Optional | `XAXIS_TITLE` | [string] Default: “” | If empty, the field name of the x attribute is used. With a single space, the axis title is not shown. |
| **Y-axis Title** Optional | `YAXIS_TITLE` | [string] Default: “” | If empty, the field name of the y attribute is used. With a single space, the axis title is not shown. |
| **Use logarithmic scale for x-axis** | `XAXIS_LOG` | [boolean] Default: False | When enabled, uses logarithmic scale for the x-axis |
| **Use logarithmic scale for y-axis** | `YAXIS_LOG` | [boolean] Default: False | When enabled, uses logarithmic scale for the y-axis |
| **Scatterplot** | `OUTPUT` | [html] 기본값: `[Save to temporary file]` | Specify the HTML file for the plot. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Scatterplot** | `OUTPUT` | [html] | 그래프 HTML 파일. *Processing → Result Viewer* 메뉴로 볼 수 있습니다. |

### 파이썬 코드

**알고리즘 ID**: `qgis:vectorlayerscatterplot`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 벡터 레이어 3 차원 산점도

벡터 레이어에 대해 3 차원 산점도를 생성합니다.

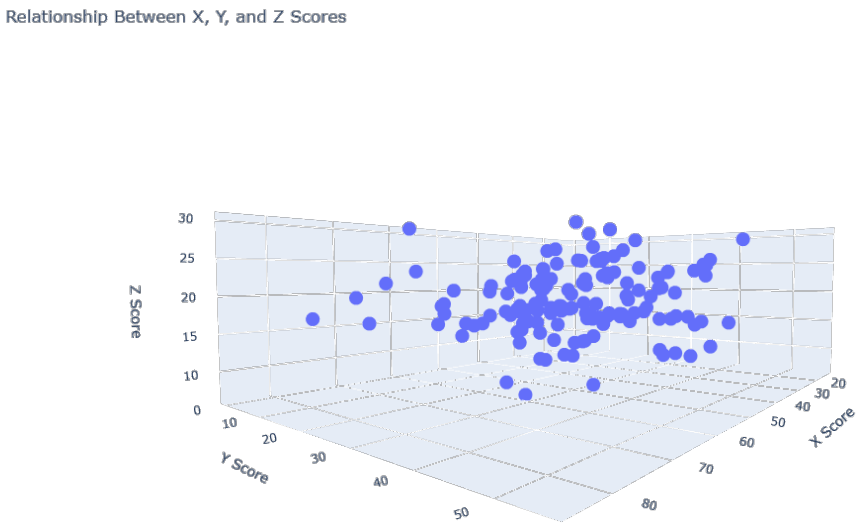



그림 24.40: Example of 3D scatter plot

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | 입력 벡터 레이어 |
| **X attribute** | `XFIELD` | [tablefield: numeric] | X 축에 사용할 필드 |
| **Y attribute** | `YFIELD` | [tablefield: numeric] | Y 축에 사용할 필드 |
| **Z attribute** | `ZFIELD` | [tablefield: numeric] | Z 축에 사용할 필드 |
| **Title**<br>Optional | `TITLE` | [string]<br>Default: “” | Title of the plot |
| **X-axis Title**<br>Optional | `XAXIS_TITLE` | [string]<br>Default: “” | If empty, the field name of the X attribute is used. |
| **Y-axis Title**<br>Optional | `YAXIS_TITLE` | [string]<br>Default: “” | If empty, the field name of the Y attribute is used. |
| **Z-axis Title**<br>Optional | `ZAXIS_TITLE` | [string]<br>Default: “” | If empty, the field name of the Z attribute is used. |
| **Scatterplot 3D** | `OUTPUT` | [html]<br>기본값: `[Save to temporary file]` | Specify the HTML file for the plot. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

#### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Scatterplot 3D** | `OUTPUT` | [html] | 그래프 HTML 파일. *Processing → Result Viewer* 메뉴로 볼 수 있습니다. |

#### 파이썬 코드

**알고리즘 ID**: `qgis:scatter3dplot`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 24.1.15 점구름 변환

> **주의**
>
> Running these algorithms requires QGIS installed with PDAL >= 2.5.0 (see *Help → About* menu).

#### 포맷 변환하기

점구름을 다른 파일 포맷으로, 예를 들면 압축 `.LAZ` 파일을 생성하는 방식으로 변환합니다.

#### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [point cloud] | 변환할 입력 점구름 레이어 |
| **Converted** | `OUTPUT` | [point cloud]<br>기본값: `[Save to temporary file]` | Specify the point cloud file to use as output. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Converted** | `OUTPUT` | [point cloud] | 수정된 파일 포맷의 산출 점구름 레이어입니다. 현재 `.LAS`, `.LAZ`, `.COPC.LAZ`, 그리고 `.VPC` 포맷을 지원합니다. |

### 파이썬 코드

**알고리즘 ID**: `pdal:convertformat`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 래스터로 내보내기

점구름 데이터를 2 차원 래스터 그리드로 내보냅니다. 지정한 해상도가 그리드 셀 크기이며, 지정한 속성의 값을 작성합니다.

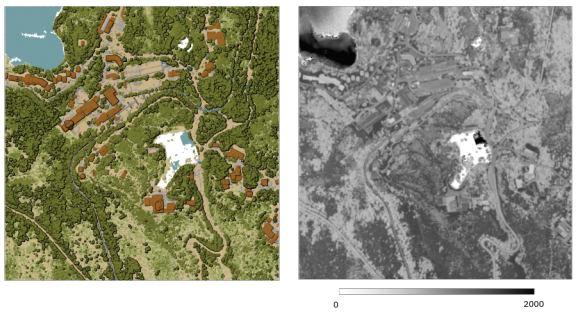



그림 24.41: 포인트의 강도 (intensity) 속성을 사용한 래스터 산출물

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [point cloud] | 내보낼 입력 점구름 레이어 |
| **Attribute** | `ATTRIBUTE` | [field] [enumeration] | 값을 추출할 점구름 레이어의 필드 |
| **Resolution of the density raster** | `RESOLUTION` | [numeric: double] 기본값: 1.0 | 산출 래스터의 셀 크기 |

다음 페이지에 계속

표 24.258 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Tile size for parallel runs** | `TILE_SIZE` | [numeric: integer]<br>기본값: 1000 | |
| **Exported** | `OUTPUT` | [raster]<br>기본값: `[Save to temporary file]` | Specify the raster file to export the data to. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Filter expression**<br>선택적 | `FILTER_EXPRESSION` | [expression] | 점구름 데이터에 있는 피처의 하위 집합을 선택하기 위한*PDAL 표현식* |
| **Cropping extent**<br>선택적 | `FILTER_EXTENT` | [extent] | 점구름 데이터에 있는 피처의 하위 집합을 선택하기 위한 맵 범위입니다.<br>다음과 같은 메소드를 사용할 수 있습니다:<br>• 레이어로부터 계산…: 현재 프로젝트에 불러온 레이어의 범위를 사용합니다.<br>• 레이어 맵으로부터 계산…: 활성 프로젝트의조판 맵 항목 범위를 사용합니다.<br>• 북마크로부터 계산…: 저장되어 있는북마크 의 범위를 사용합니다.<br>• 맵 캔버스 범위 사용<br>• 캔버스에 그리기: 연산에 넣을 범위를 정하는 직사각형을 클릭 & 드래그합니다.<br>• 좌표를 `xmin`, `xmax`, `ymin`, `ymax` 로 입력 |
| **X origin of a tile for parallel runs**<br>선택적 | `ORIGIN_X` | [numeric: double] | |
| **Y origin of a tile for parallel runs**<br>선택적 | `ORIGIN_Y` | [numeric: double] | |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Exported** | `OUTPUT` | [raster] | 점구름 레이어의 피처를 내보낸 산출 래스터 레이어입니다. 현재 `.TIF` 포맷을 지원합니다. |

### 파이썬 코드

**알고리즘 ID**: `pdal:exportraster`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 래스터 내보내기 (삼각분할 사용)

점구름 데이터를 포인트들을 삼각분할한 다음 삼각형들에서 나온 셀 값들을 보간해서 2 차원 래스터 그리드로 내보냅니다.

> **참고**
>
> 이 알고리즘은 대용량 데이터셋을 처리하는 경우 느릴 수 있습니다. 사용자의 점구름 밀도가 높다면, 지상 포인트들을래스터로 내보내기 알고리즘을 사용해서 래스터로 내보내면 됩니다.

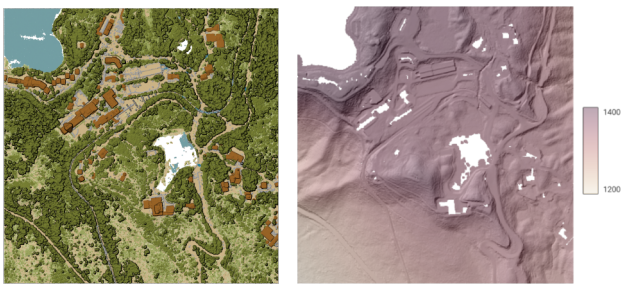



그림 24.42: 점구름 삼각분할로 생성된 지형 래스터 산출물

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [point cloud] | 내보낼 입력 점구름 레이어 |
| **Resolution of the density raster** | `RESOLUTION` | [numeric: double]<br>기본값: 1.0 | 산출 래스터의 셀 크기 |
| **Tile size for parallel runs** | `TILE_SIZE` | [numeric: integer]<br>기본값: 1000 | |
| **Exported** | `OUTPUT` | [raster]<br>기본값: `[Save to temporary file]` | Specify the raster file to export the data to. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Filter expression**<br>선택적 | `FILTER_EXPRESSION` | [expression] | 점구름 데이터에 있는 피처의 하위 집합을 선택하기 위한*PDAL* 표현식 |
| **Cropping extent**<br>선택적 | `FILTER_EXTENT` | [extent] | 점구름 데이터에 있는 피처의 하위 집합을 선택하기 위한 맵 범위입니다.<br>다음과 같은 메소드를 사용할 수 있습니다:<br>• 레이어로부터 계산…: 현재 프로젝트에 불러온 레이어의 범위를 사용합니다.<br>• 레이어 맵으로부터 계산…: 활성 프로젝트의조판 맵 항목 범위를 사용합니다.<br>• 북마크로부터 계산…: 저장되어 있는북마크 의 범위를 사용합니다.<br>• 맵 캔버스 범위 사용<br>• 캔버스에 그리기: 연산에 넣을 범위를 정하는 직사각형을 클릭 & 드래그합니다.<br>• 좌표를 `xmin`, `xmax`, `ymin`, `ymax` 로 입력 |
| **X origin of a tile for parallel runs**<br>선택적 | `ORIGIN_X` | [numeric: double] | |
| **Y origin of a tile for parallel runs**<br>선택적 | `ORIGIN_Y` | [numeric: double] | |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Exported (using triangulation)** | `OUTPUT` | [raster] | 점구름 레이어의 피처를 내보낸 산출 래스터 레이어입니다. 현재 `.TIF` 포맷을 지원합니다. |

### 파이썬 코드

**알고리즘 ID**: `pdal:exportrastertin`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 벡터로 내보내기

점구름 데이터를 3 차원 포인트를 가진 벡터 레이어 (GeoPackage) 로 내보냅니다. 선택적인 추가 속성을 가질 수도 있습니다.

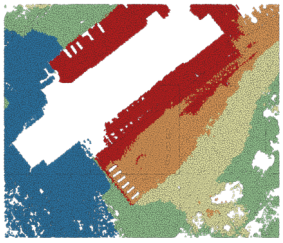


그림 24.43: 점구름 (지상 포인트) 를 표고 기반 스타일을 적용한 벡터 레이어로 내보내기

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [point cloud] | 내보낼 입력 점구름 레이어 |
| **Attribute** 선택적 | `ATTRIBUTE` | [field] [list] | 포인트와 함께 내보낼 점구름 레이어의 하나 이상의 필드 |

다음 페이지에 계속

표 24.264 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Exported** | `OUTPUT` | [vector]<br>기본값: `[Save to temporary file]` | Specify the vector file to export the data to.<br>*One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Filter expression**<br>선택적 | `FILTER_EXPRESSION` | [expression] | 점구름 데이터에 있는 피처의 하위 집합을 선택하기 위한*PDAL* 표현식 |
| **Cropping extent**<br>선택적 | `FILTER_EXTENT` | [extent] | 점구름 데이터에 있는 피처의 하위 집합을 선택하기 위한 맵 범위입니다.<br>다음과 같은 메소드를 사용할 수 있습니다:<br>• 레이어로부터 계산…: 현재 프로젝트에 불러온 레이어의 범위를 사용합니다.<br>• 레이어 맵으로부터 계산…: 활성 프로젝트의조판 맵 항목 범위를 사용합니다.<br>• 북마크로부터 계산…: 저장되어 있는북마크 의 범위를 사용합니다.<br>• 맵 캔버스 범위 사용<br>• 캔버스에 그리기: 연산에 넣을 범위를 정하는 직사각형을 클릭 & 드래그합니다.<br>• 좌표를 `xmin`, `xmax`, `ymin`, `ymax` 로 입력 |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Exported** | `OUTPUT` | [vector] | 점구름 레이어의 피처를 내보낸 산출 벡터 레이어입니다. 현재 `.GPKG` 포맷을 지원합니다. |

#### 파이썬 코드

**알고리즘 ID**: `pdal:exportvector`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 24.1.16 점구름 데이터 관리

**주의**

Running these algorithms requires QGIS installed with PDAL >= 2.5.0 (see *Help → About* menu).

#### 투영체 할당하기

점구름 레이어의 좌표계가 누락되었거나 잘못된 경우, 새 레이어를 생성하는 방식으로 점구름 레이어에 좌표계를 할당합니다.

**더 보기**

재투영하기

#### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [point cloud] | 좌표계를 할당할 입력 점구름 레이어 |
| **Desired CRS** | `CRS` | [crs] | 레이어에 적용할 좌표계 |
| **Output layer** | `OUTPUT` | [point cloud]<br>기본값: `[Save to temporary file]` | Specify the point cloud file to use as output.<br>*One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Converted** | `OUTPUT` | [point cloud] | 새 좌표계를 가진 산출 점구름 레이어입니다. 현재 `.LAS`, `.LAZ`, `.COPC.LAZ`, 그리고 `.VPC` 포맷을 지원합니다. |

### 파이썬 코드

**알고리즘 ID**: `pdal:assignprojection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 가상 점구름 (VPC) 작성하기

입력 점구름 데이터로부터가상 점구름 *(Virtual Point Cloud)* 을 생성합니다.

선택적인 파라미터들을 체크하지 않은 채 알고리즘을 실행하면, 알고리즘이 입력 파일의 메타데이터만 읽어올 것이기 때문에 VPC 를 매우 빠르게 작성할 것입니다. 선택적인 파라미터를 하나라도 설정한 경우, 알고리즘이 모든 포인트를 읽어올 것이기 때문에 시간이 걸릴 수 있습니다.

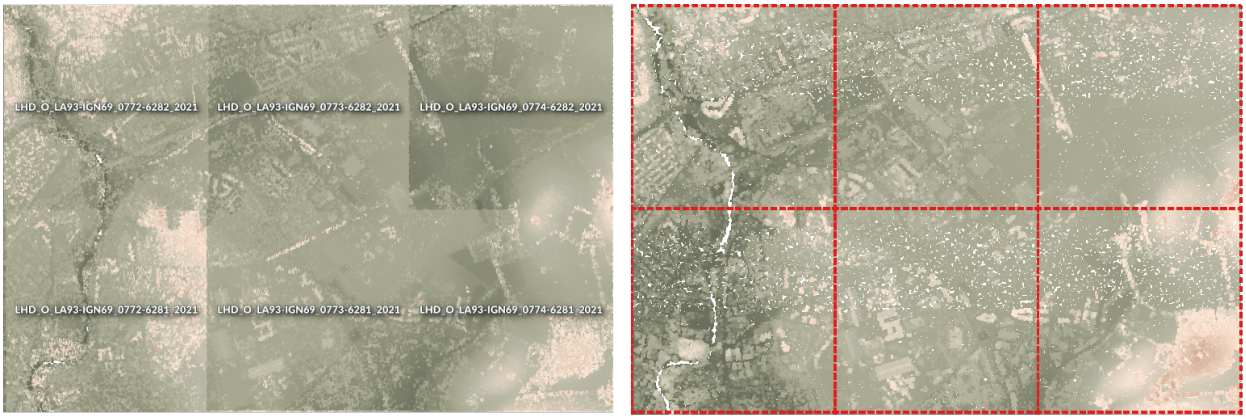



그림 24.44: 점구름 타일 집합으로부터 오버뷰를 가진 가상 점구름 생성하기

**더 보기**

병합하기

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layers** | `LAYERS` | [point cloud] [list] | 가상 점구름 레이어 내부에 결합할 입력 점구름 레이어들 |
| **Calculate boundary polygon** | `BOUNDARY` | [boolean]<br>기본값: False | 참으로 설정하면 데이터의 (직사각형 범위만 표시하기보다) 정확한 경계를 표시합니다. |
| **Calculate statistics** | `STATISTICS` | [boolean]<br>기본값: False | 참으로 설정하면 여러 속성들의 값의 범위를 이해할 수 있습니다. |
| **Build overview point cloud** | `OVERVIEW` | [boolean]<br>기본값: False | 모든 입력 데이터의 (원본 데이터에서 1,000 번째마다의 포인트만 사용해서) "솎아낸 (thinned)"단일 점구름을 생성합니다.<br>VPC 파일 옆에 오버뷰 점구름을 생성할 것입니다. 예를 들어 `mydata.vpc` 의 경우 오버뷰 점구름의 이름은 `mydata-overview.copc.laz` 가 될 것입니다. |
| **Virtual point cloud** | `OUTPUT` | [point cloud]<br>기본값: `[Save to temporary file]` | Specify the point cloud file to build the data into. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Virtual point cloud** | `OUTPUT` | [raster] | 모든 입력 데이터를 결합한, 가상 파일로써의 산출 점구름 레이어입니다. |

### 파이썬 코드

**알고리즘 ID**: `pdal:virtualpointcloud`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 잘라내기 (Clip)

산출되는 점구름이 폴리곤 내부에 들어오는 포인트만 담게 되도록 점구름 레이어를 폴리곤 레이어로 잘라냅니다.

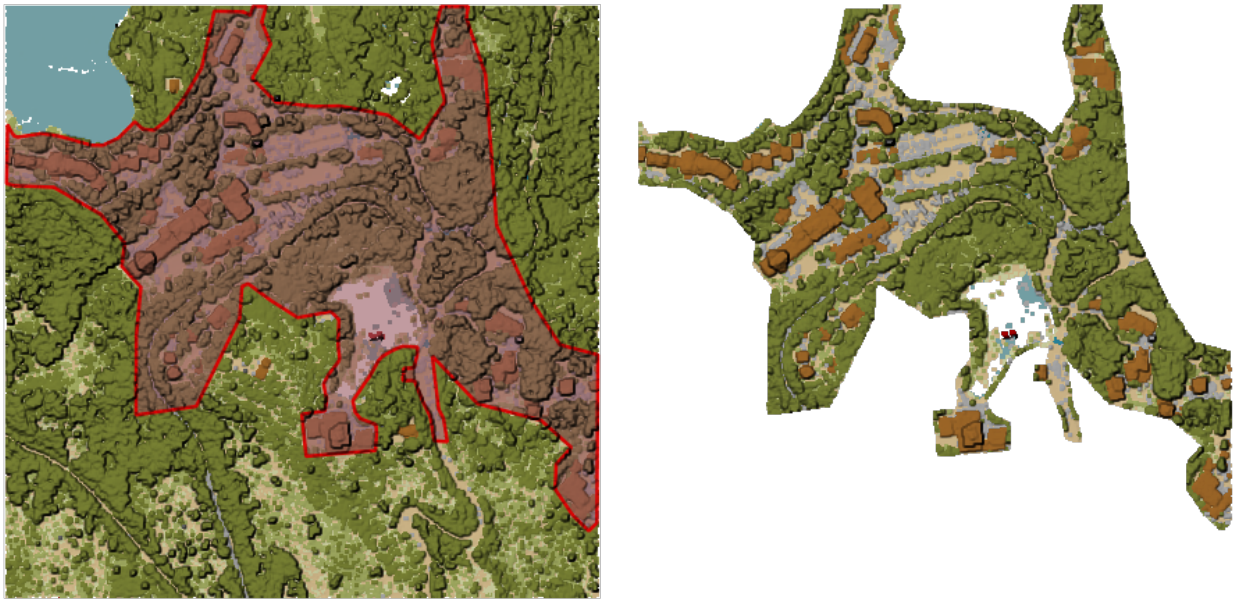


그림 24.45: 입력 점구름 레이어를 폴리곤 커버리지로 잘라내기

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [point cloud] | 잘라낼 입력 점구름 레이어 |
| **Clipping polygons** | `OVERLAY` | [vector: polygon] | 포인트들을 잘라내기 위한 커버리지로 사용할 폴리곤 벡터 레이어 |
| **Clipped** | `OUTPUT` | [point cloud]<br>기본값: `[Save to temporary file]` | Specify the point cloud file to export the clipped points to. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Filter expression** 선택적 | FILTER_EXPRESSION | [expression] | 점구름 데이터에 있는 피처의 하위 집합을 선택하기 위한*PDAL* 표현식 |
| **Cropping extent** 선택적 | FILTER_EXTENT | [extent] | 점구름 데이터에 있는 피처의 하위 집합을 선택하기 위한 맵 범위<br>다음과 같은 메소드를 사용할 수 있습니다:<br>• 레이어로부터 계산…: 현재 프로젝트에 불러온 레이어의 범위를 사용합니다.<br>• 레이어 맵으로부터 계산…: 활성 프로젝트의조판 맵 항목 범위를 사용합니다.<br>• 북마크로부터 계산…: 저장되어 있는북마크 의 범위를 사용합니다.<br>• 맵 캔버스 범위 사용<br>• 캔버스에 그리기: 연산에 넣을 범위를 정하는 직사각형을 클릭 & 드래그합니다.<br>• 좌표를 xmin, xmax, ymin, ymax 로 입력 |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Clipped** | OUTPUT | [point cloud] | 커버리지 폴리곤 레이어 내부에 들어오는 포인트들을 피처로 가진 산출 점구름입니다. |

### 파이썬 코드

**알고리즘 ID**: pdal:clip

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### COPC 생성하기

모든 입력 점구름 파일에 대한 인덱스를 배치 (batch) 모드로 생성합니다.

#### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layers** | `LAYERS` | [point cloud] [list] | 인덱스를 생성할 입력 점구름 레이어들 |
| **Output directory** 선택적 | `OUTPUT` | [folder] 기본값: `[Skip output]` | Specify the folder to create the new files in. *One of*:<br>• 산출물 건너뛰기<br>• 임시 디렉터리에 저장<br>• 디렉터리에 저장 |

#### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output directory** | `OUTPUT` | [folder] | 동반하는 COPC 인덱스 파일을 가진 점구름 레이어들을 담고 있는 산출 폴더입니다. |

#### 파이썬 코드

**알고리즘 ID**: `pdal:createcopc`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 정보

입력 점구름 파일로부터 기본 메타데이터를 산출합니다.

산출 정보의 예시:

```
LAS           1.4
point format  6
count         56736130
scale         0.001 0.001 0.001
offset        431749.999 5440919.999 968.898
extent        431250 5440420 424.266
              432249.999 5441419.999 1513.531
crs           ETRS89 / UTM zone 34N (N-E) (EPSG:3046)  (vertical CRS missing!)
```

(다음 페이지에 계속)

(이전 페이지에서 계속)

```
units         horizontal=metre  vertical=unknown

Attributes:
- X floating 8
- Y floating 8
- Z floating 8
- Intensity unsigned 2
- ReturnNumber unsigned 1
- NumberOfReturns unsigned 1
- ScanDirectionFlag unsigned 1
- EdgeOfFlightLine unsigned 1
- Classification unsigned 1
- ScanAngleRank floating 4
- UserData unsigned 1
- PointSourceId unsigned 2
- GpsTime floating 8
- ScanChannel unsigned 1
- ClassFlags unsigned 1
```

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [point cloud] | 메타데이터 정보를 추출할 입력 점구름 레이어 |
| **Layer information** | `OUTPUT` | [file]<br>기본값: `[Save to temporary file]` | Specify the file to store the metadata information. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Layer information** | `OUTPUT` | [html] | 메타데이터 정보를 저장할 `HTML` 파일입니다. |

### 파이썬 코드

**알고리즘 ID**: `pdal:info`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 병합하기

점구름 파일 여러 개를 단일 파일로 병합합니다.

**더 보기**

가상 점구름 *(VPC)* 작성하기

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layers** | `LAYERS` | [point cloud] [list] | 단일 레이어로 병합시킬 입력 점구름 레이어들 |
| **Merged** | `OUTPUT` | [point cloud]<br>기본값: `[Save to temporary file]` | Specify the output point cloud merging input files. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Filter expression**<br>선택적 | `FILTER_EXPRESSION` | [expression] | 점구름 데이터에 있는 피처의 하위 집합을 선택하기 위한*PDAL* 표현식 |
| **Cropping extent**<br>선택적 | `FILTER_EXTENT` | [extent] | 점구름 데이터에 있는 피처의 하위 집합을 선택하기 위한 맵 범위<br>다음과 같은 메소드를 사용할 수 있습니다:<br>• 레이어로부터 계산…: 현재 프로젝트에 불러온 레이어의 범위를 사용합니다.<br>• 레이어 맵으로부터 계산…: 활성 프로젝트의조판 맵 항목 범위를 사용합니다.<br>• 북마크로부터 계산…: 저장되어 있는북마크 의 범위를 사용합니다.<br>• 맵 캔버스 범위 사용<br>• 캔버스에 그리기: 연산에 넣을 범위를 정하는 직사각형을 클릭 & 드래그합니다.<br>• 좌표를 `xmin, xmax, ymin, ymax` 로 입력 |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Merged** | `OUTPUT` | [point cloud] | 모든 입력 데이터를 병합한 산출 점구름 레이어입니다. |

### 파이썬 코드

**알고리즘 ID**: `pdal:merge`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 재투영하기

점구름을 다른 좌표계로 재투영합니다.

**더 보기**

투영체 할당하기

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [point cloud] | 다른 좌표계로 재투영할 입력 점구름 레이어 |
| **Target CRS** | `CRS` | [crs] | 레이어에 적용할 좌표계 |
| **Reprojected** | `OUTPUT` | [point cloud]<br>기본값: `[Save to temporary file]` | Specify the reprojected point cloud file. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Coordinate operation**<br>선택적 | `OPERATION` | [datum] | 원본 좌표계와 대상 좌표계 사이에 데이터를 재투영하기 위해 사용할기준점 변환 입니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **REPROJECTED** | `OUTPUT` | [point cloud] | 대상 좌표계를 적용한 산출 점구름 레이어입니다. |

### 파이썬 코드

**알고리즘 ID**: `pdal:reproject`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 솎아내기 (샘플링 반경)

거리 포인트로 샘플링을 수행하는 방식으로 점구름의 솎아낸 버전을 생성합니다. (특정 반경 안에 들어오는 포인트의 개수를 감소시킵니다.)

**더 보기**

솎아내기 (포인트 건너뛰기)

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [point cloud] | 솎아낸 버전을 생성할 입력 점구름 레이어 |
| **Sampling radius (in map units)** | `SAMPLING_RADIUS` | [numeric: double]<br>기본값: 1.0 | 이 안에 들어오는 포인트들을 유일 포인트로 샘플링할 거리 |

다음 페이지에 계속

표 24.284 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Thinned (by radius)** | `OUTPUT` | [point cloud]<br>기본값: `[Save to temporary file]` | Specify the output point cloud with reduced points. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Filter expression** 선택적 | `FILTER_EXPRESSION` | [expression] | 점구름 데이터에 있는 피처의 하위 집합을 선택하기 위한*PDAL* 표현식 |
| **Cropping extent** 선택적 | `FILTER_EXTENT` | [extent] | 점구름 데이터에 있는 피처의 하위 집합을 선택하기 위한 맵 범위<br>다음과 같은 메소드를 사용할 수 있습니다:<br>• 레이어로부터 계산…: 현재 프로젝트에 불러온 레이어의 범위를 사용합니다.<br>• 레이어 맵으로부터 계산…: 활성 프로젝트의조판 맵 항목 범위를 사용합니다.<br>• 북마크로부터 계산…: 저장되어 있는북마크 의 범위를 사용합니다.<br>• 맵 캔버스 범위 사용<br>• 캔버스에 그리기: 연산에 넣을 범위를 정하는 직사각형을 클릭 & 드래그합니다.<br>• 좌표를 `xmin`, `xmax`, `ymin`, `ymax` 로 입력 |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Thinned (by radius)** | `OUTPUT` | [point cloud] | 감소된 포인트들을 가진 산출 점구름 레이어입니다. |

### 파이썬 코드

**알고리즘 ID**: `pdal:thinbyradius`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 솎아내기 (포인트 건너뛰기)

N 번째마다의 포인트만 남기는 방식으로 점구름의 솎아낸 버전을 생성합니다. (가까이 있는 포인트들을 건너뛰어 포인트의 개수를 감소시킵니다.)

**더 보기**

솎아내기 *(샘플링 반경)*

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [point cloud] | 솎아낸 버전을 생성할 입력 점구름 레이어 |
| **Number of points to skip** | `POINTS_NUMBER` | [numeric: integer]<br>기본값: 1 | 입력 레이어에서 N 번째마다의 포인트만 남깁니다. |
| **Thinned (by decimation)** | `OUTPUT` | [point cloud]<br>기본값: `[Save to temporary file]` | Specify the output point cloud with reduced points. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Filter expression**<br>선택적 | `FILTER_EXPRESSION` | [expression] | 점구름 데이터에 있는 피처의 하위 집합을 선택하기 위한*PDAL* 표현식 |

다음 페이지에 계속

표 24.288 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Cropping extent** 선택적 | `FILTER_EXTENT` | [extent] | 점구름 데이터에 있는 피처의 하위 집합을 선택하기 위한 맵 범위<br>다음과 같은 메소드를 사용할 수 있습니다:<br>• 레이어로부터 계산…: 현재 프로젝트에 불러온 레이어의 범위를 사용합니다.<br>• 레이어 맵으로부터 계산…: 활성 프로젝트의조판 맵 항목 범위를 사용합니다.<br>• 북마크로부터 계산…: 저장되어 있는북마크 의 범위를 사용합니다.<br>• 맵 캔버스 범위 사용<br>• 캔버스에 그리기: 연산에 넣을 범위를 정하는 직사각형을 클릭 & 드래그합니다.<br>• 좌표를 `xmin`, `xmax`, `ymin`, `ymax` 로 입력 |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Thinned (by decimation)** | `OUTPUT` | [point cloud] | 감소된 포인트들을 가진 산출 점구름 레이어입니다. |

### 파이썬 코드

**알고리즘 ID**: `pdal:thinbydecimate`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 타일

입력 점구름 파일로부터 타일을 생성합니다. QGIS 에서 점구름 데이터셋을 (보기 또는 분석) 작업할 때 최고의 성능을 보여주기 위해 권장됩니다.

**더 보기**

가상 점구름 *(VPC)* 작성하기, *COPC* 생성하기

### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layers** | `LAYERS` | [point cloud] [list] | 타일을 생성할 입력 점구름 레이어들 |
| **Tile length** | `LENGTH` | [numeric: double] Default: 1000.0 | 생성된 각 타일의 변의 길이 |
| **Output directory** | `OUTPUT` | [folder] 기본값: `[Save to temporary folder]` | Specify the folder to store the generated tiles. *One of*:<br>• 임시 디렉터리에 저장<br>• 디렉터리에 저장 |

#### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Assign CRS** 선택적 | `CRS` | [crs] | 레이어에 적용할 좌표계 |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output directory** | `OUTPUT` | [folder] | 입력 파일로부터 생성된 타일들을 담고 있는 산출 폴더입니다. |

### 파이썬 코드

**알고리즘 ID**: `pdal:tile`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 24.1.17 점구름 추출

> **주의**
>
> Running these algorithms requires QGIS installed with PDAL >= 2.5.0 (see *Help* → *About* menu).

#### 경계

점구름 레이어의 경계를 담고 있는 폴리곤 파일을 추출합니다. 구멍을 담고 있을 수도 있으며 다중 부분 폴리곤일 수도 있습니다.

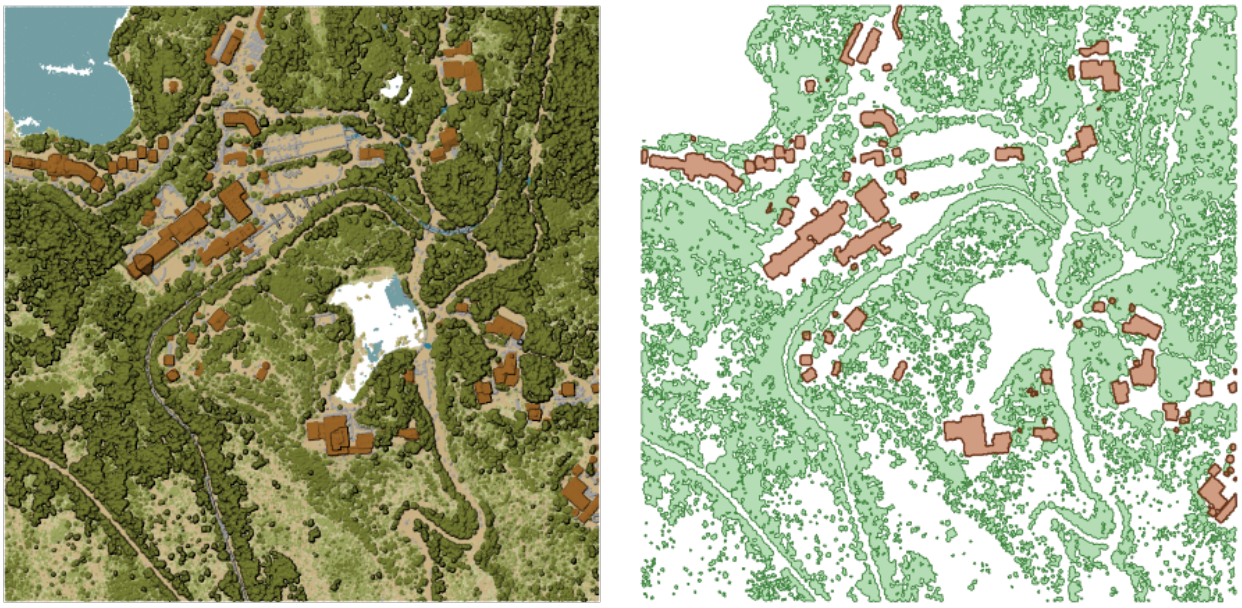


그림 24.46: 입력 점구름 레이어에서 상층 식생 (high vegetation) 및 건물 폴리곤 추출하기

#### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [point cloud] | 경계를 계산할 입력 점구름 레이어 |
| **Resolution of cells used to calculate boundary**<br>선택적 | `RESOLUTION` | [numeric: double] | 경계를 계산하는 데 사용한 셀의 해상도 |
| **Minimal number of points in a cell to consider cell occupied**<br>선택적 | `THRESHOLD` | [numeric: integer] | 셀을 차 있다고 간주하려면 필요한 셀 안 포인트의 최소 개수 |

다음 페이지에 계속

표 24.293 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Boundary** | `OUTPUT` | [vector: polygon]<br>기본값: `[Save to temporary file]` | Specify the point cloud file to use as output.<br>*One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Filter expression**<br>선택적 | `FILTER_EXPRESSION` | [expression] | 점구름 데이터에 있는 피처의 하위 집합을 선택하기 위한*PDAL* 표현식 |
| **Cropping extent**<br>선택적 | `FILTER_EXTENT` | [extent] | 점구름 데이터에 있는 피처의 하위 집합을 선택하기 위한 맵 범위<br>다음과 같은 메소드를 사용할 수 있습니다:<br>• 레이어로부터 계산…: 현재 프로젝트에 불러온 레이어의 범위를 사용합니다.<br>• 레이어 맵으로부터 계산…: 활성 프로젝트의조판 맵 항목 범위를 사용합니다.<br>• 북마크로부터 계산…: 저장되어 있는북마크 의 범위를 사용합니다.<br>• 맵 캔버스 범위 사용<br>• 캔버스에 그리기: 연산에 넣을 범위를 정하는 직사각형을 클릭 & 드래그합니다.<br>• 좌표를 `xmin`, `xmax`, `ymin`, `ymax` 로 입력 |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Boundary** | `OUTPUT` | [vector: polygon] | 점구름 경계를 가진 산출 폴리곤 벡터 레이어입니다. 현재 `.GPKG` 포맷을 지원합니다. |

### 파이썬 코드

**알고리즘 ID**: `pdal:boundary`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 밀도

각 래스터 셀 내부의 포인트 개수를 바탕으로 한 래스터 파일을 추출합니다. 점구름 데이터셋의 품질을 확인하는 데 유용합니다.

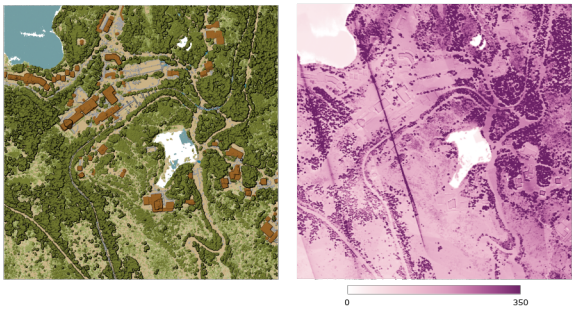



그림 24.47: 래스터로써의 포인트 밀도 (2x2 미터 당 포인트 개수)

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [point cloud] | 내보낼 입력 점구름 레이어 |
| **Resolution of the density raster** | `RESOLUTION` | [numeric: double]<br>기본값: 1.0 | 산출 래스터의 셀 크기 |
| **Tile size for parallel runs** | `TILE_SIZE` | [numeric: integer]<br>기본값: 1000 | 병렬 실행을 위해 데이터를 분할시킬 타일의 크기 |
| **Density** | `OUTPUT` | [raster]<br>기본값: `[Save to temporary file]` | Specify the raster file to export the data to. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Filter expression**<br>선택적 | `FILTER_EXPRESSION` | [expression] | 점구름 데이터에 있는 피처의 하위 집합을 선택하기 위한*PDAL 표현식* |
| **Cropping extent**<br>선택적 | `FILTER_EXTENT` | [extent] | 점구름 데이터에 있는 피처의 하위 집합을 선택하기 위한 맵 범위<br>다음과 같은 메소드를 사용할 수 있습니다:<br>• 레이어로부터 계산…: 현재 프로젝트에 불러온 레이어의 범위를 사용합니다.<br>• 레이어 맵으로부터 계산…: 활성 프로젝트의조판 맵 항목 범위를 사용합니다.<br>• 북마크로부터 계산…: 저장되어 있는북마크 의 범위를 사용합니다.<br>• 맵 캔버스 범위 사용<br>• 캔버스에 그리기: 연산에 넣을 범위를 정하는 직사각형을 클릭 & 드래그합니다.<br>• 좌표를 `xmin`, `xmax`, `ymin`, `ymax` 로 입력 |
| **X origin of a tile for parallel runs**<br>선택적 | `ORIGIN_X` | [numeric: double] | |
| **Y origin of a tile for parallel runs**<br>선택적 | `ORIGIN_Y` | [numeric: double] | |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Density** | `OUTPUT` | [raster] | 각 셀 내부의 포인트 개수를 가진 산출 래스터 레이어입니다. 현재 `.TIF` 포맷을 지원합니다. |

### 파이썬 코드

**알고리즘 ID**: `pdal:density`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 필터

입력 점구름으로부터 PDAL 표현식과 일치하는 그리고/또는 잘라내기 (cropping) 직사각형 안에 들어오는 포인트를 추출합니다.

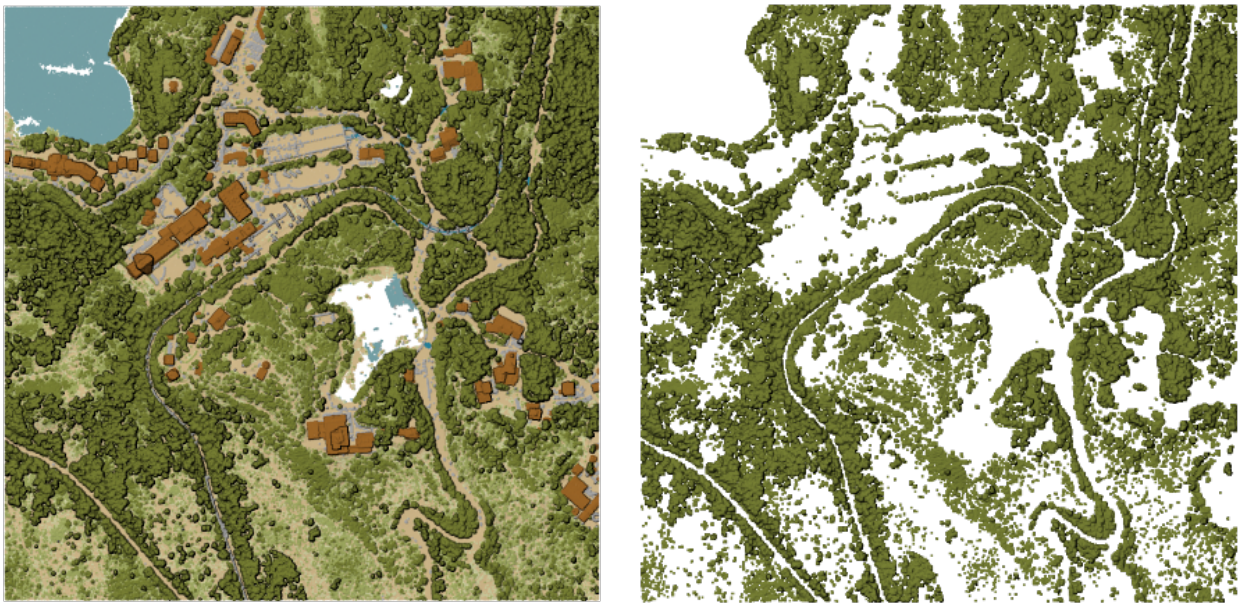


그림 24.48: 입력 점구름 레이어에서 상층 식생 (high vegetation) 범주로 필터링하기

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [point cloud] | 내보낼 입력 점구름 레이어 |
| **Filter expression** 선택적 | `FILTER_EXPRESSION` | [expression] | 점구름 데이터에 있는 피처의 하위 집합을 선택하기 위한*PDAL* 표현식 |

다음 페이지에 계속

표 24.299 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Cropping extent**<br>선택적 | `FILTER_EXTENT` | [extent] | 점구름 데이터에 있는 피처의 하위 집합을 선택하기 위한 맵 범위<br>다음과 같은 메소드를 사용할 수 있습니다:<br>• 레이어로부터 계산…: 현재 프로젝트에 불러온 레이어의 범위를 사용합니다.<br>• 레이어 맵으로부터 계산…: 활성 프로젝트의조판 맵 항목 범위를 사용합니다.<br>• 북마크로부터 계산…: 저장되어 있는북마크 의 범위를 사용합니다.<br>• 맵 캔버스 범위 사용<br>• 캔버스에 그리기: 연산에 넣을 범위를 정하는 직사각형을 클릭 & 드래그합니다.<br>• 좌표를 `xmin`, `xmax`, `ymin`, `ymax` 로 입력 |
| **Filtered** | `OUTPUT` | [point cloud]<br>기본값: `[Save to temporary file]` | Specify the point cloud file to export the data to. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Filtered** | `OUTPUT` | [point cloud] | 입력 점구름 레이어에서 필터링한 피처를 가진 산출 점구름 레이어입니다. 현재 `.LAS`, `.LAZ`, `.COPC.LAZ`, 그리고 `.VPC` 포맷을 지원합니다. |

### 파이썬 코드

**알고리즘 ID**: `pdal:filter`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 24.1.18 래스터 분석

#### 값으로부터 셀 스택 백분율 순위 구하기

단일 입력 값을 바탕으로 래스터 스택의 셀 별 백분율 순위 값을 계산해서 산출 래스터에 작성합니다.

각 셀의 위치에서, 입력 래스터의 중첩되고 정렬된 모든 셀 값들의 스택에 있는 해당 위치의 값들 가운데 지정한 값의 순위를 매깁니다. 지정한 값이 스택 값 분포를 벗어나는 경우, 이 알고리즘은 셀 값들 가운데 해당 값의 순위를 매길 수가 없기 때문에 NODATA 를 반환합니다.

백분위수 (percentile) 를 계산하기 위한 방법이 두 가지 있습니다:

- 포괄적 (inclusive) 선형 보간법 (PERCENTRANK.INC)
- 배타적 (exclusive) 선형 보간법 (PERCENTRANK.EXC)

선형 보간법은 서로 다른 값에 대해 유일한 백분율 순위를 반환합니다. 이 두 가지 보간법 모두 리브레오피스 또는 마이크로소프트 엑셀이 구현하는 해당 보간법을 따릅니다.

참조 래스터가 산출 래스터의 범위와 해상도를 정의합니다. 입력 래스터 레이어의 셀 크기가 참조 래스터의 셀 크기와 일치하지 않을 경우, 최근접 이웃 리샘플링 메소드를 이용해서 입력 래스터 레이어를 리샘플링할 것입니다. “NODATA 값 무시”파라미터를 설정하지 않았다면 입력 레이어의 모든 NODATA 값을 NODATA 셀 산출물로 산출할 것입니다. 산출 래스터의 데이터 유형은 항상 `Float32` 일 것입니다.

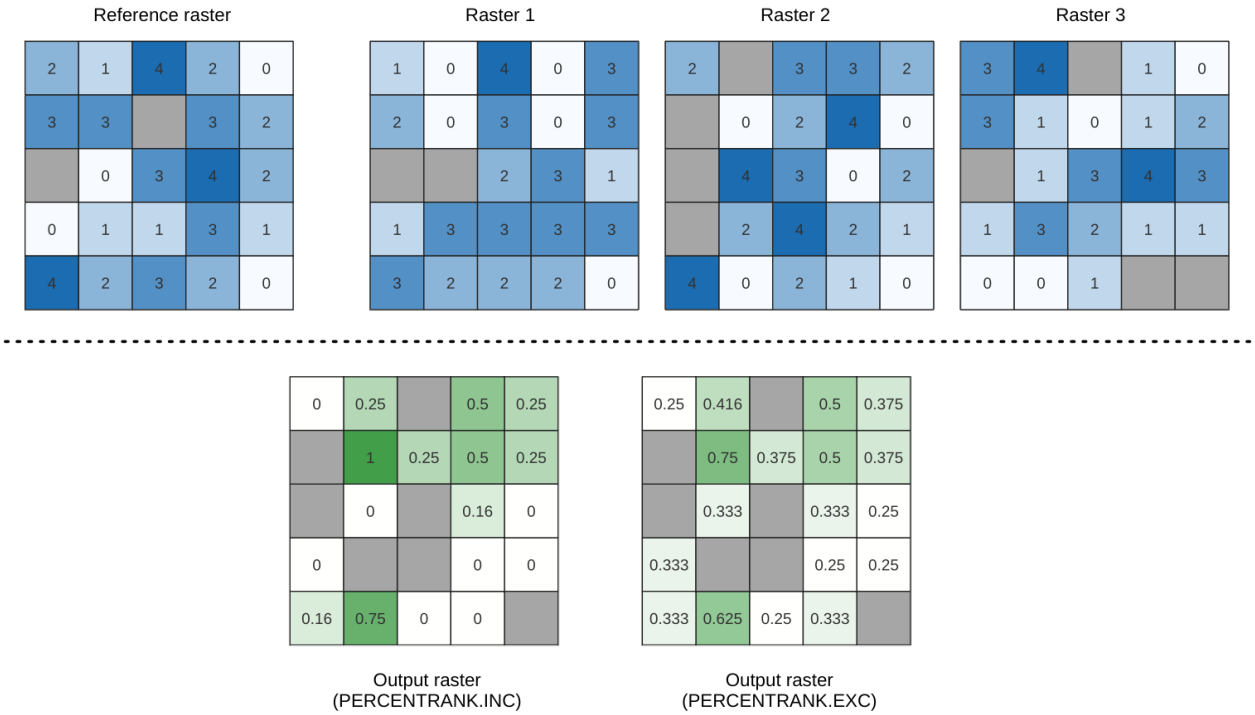



그림 24.49: 백분율 순위 값 = 1. `NoData` 셀 (회색) 은 무시합니다.

**더 보기**

셀 스택 백분위수 구하기, 래스터 레이어로부터 셀 스택 백분율 순위 구하기

### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layers** | `INPUT` | [raster] [list] | 평가할 래스터 레이어들입니다. 데이터 래스터 스택에 다중밴드 래스터가 쓰인 경우, 이 알고리즘은 항상 래스터의 첫 번째 밴드에 분석을 수행할 것입니다. |
| **Method** | `METHOD` | [enumeration] 기본값: 0 | 백분위수 (percentile) 를 계산하기 위한 방법:<br>• 0 —포괄적 선형 보간법 (PERCENTRANK.INC)<br>• 1 —배타적 선형 보간법 (PERCENTRANK.EXC) |
| **Value** | `VALUE` | [numeric: double] 기본값: 10.0 | 입력 래스터의 중첩되고 정렬된 모든 셀 값들의 스택에 있는 각각의 값들 가운데 순위를 매길 값 |
| **Ignore NoData values** | `IGNORE_NODATA` | [boolean] 기본값: True | 이 옵션을 체크하지 않을 경우, 입력 레이어에 있는 모든 NODATA 셀이 산출 래스터에서 NODATA 셀로 산출될 것입니다. |
| **Reference layer** | `REFERENCE_LAYER` | [raster] | 산출 래스터를 생성하기 위한 (범위, 좌표계, 픽셀 크기) 기준 레이어 |
| **Output layer** | `OUTPUT` | [same as input] 기본값: `[Save to temporary file]` | Specification of the output raster. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

#### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output NoData value** | `OUTPUT_NODATA_VALU` | [numeric: double] 기본값: -9999.0 | 산출 레이어에 있는 NODATA 에 사용할 값 |
| **Creation options** 선택적 | `CREATION_OPTIONS` (for QGIS <= 3.42, this was `CREATE_OPTIONS`) | [string] 기본값: '' | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL* 드라이버 옵션 참조) 배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (\|) 로 복수의 옵션들을 분리하십시오. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output layer** | `OUTPUT` | [raster] | 결과물을 담고 있는 산출 래스터 레이어 |
| **CRS authority identifier** | `CRS_AUTHID` | [string] | 산출 래스터 레이어의 좌표계 |
| **Extent** | `EXTENT` | [string] | 산출 래스터 레이어의 공간 영역 |
| **Width in pixels** | `WIDTH_IN_PIXELS` | [numeric: integer] | 산출 래스터 레이어의 열 개수 |
| **Height in pixels** | `HEIGHT_IN_PIXEI` | [numeric: integer] | 산출 래스터 레이어의 행 개수 |
| **Total pixel count** | `TOTAL_PIXEL_COUNT` | [numeric: integer] | 산출 래스터 레이어의 픽셀 개수 |

### 파이썬 코드

**알고리즘 ID**: `native:cellstackpercentrankfromvalue`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 셀 스택 백분위수 구하기

래스터 스택의 셀 별 백분위수 값을 계산해서 산출 래스터에 결과값을 작성합니다. 반환할 백분위수는 (0 에서 1 사이의) 백분위수 입력값에 따라 결정됩니다. 각 셀의 위치에서, 입력 래스터의 중첩되고 정렬된 모든 셀 값들의 스택의 각각의 값들을 이용해서 지정한 백분위수를 구합니다.

백분위수를 계산하기 위한 방법이 세 가지 있습니다:

- 최근접 순위: 지정한 백분위수에 가장 가까운 값을 반환합니다.
- 포괄적 (inclusive) 선형 보간법 (PERCENTRANK.INC)
- 배타적 (exclusive) 선형 보간법 (PERCENTRANK.EXC)

선형 보간법은 서로 다른 백분위수에 대해 유일한 값을 반환합니다. 이 두 가지 보간법 모두 리브레오피스 또는 마이크로소프트 엑셀이 구현하는 해당 보간법을 따릅니다.

참조 래스터가 산출 래스터의 범위와 해상도를 정의합니다. 입력 래스터 레이어의 셀 크기가 참조 래스터의 셀 크기와 일치하지 않을 경우, 최근접 이웃 리샘플링 메소드를 이용해서 입력 래스터 레이어를 리샘플링할 것입니다. "NODATA 값 무시"파라미터를 설정하지 않았다면 입력 레이어의 모든 NODATA 값을 NODATA 셀 산출물로 산출할 것입니다. 산출 래스터의 데이터 유형은 항상 `Float32` 일 것입니다.

**더 보기**

셀 스택 백분위수 구하기, 래스터 레이어로부터 셀 스택 백분율 순위 구하기

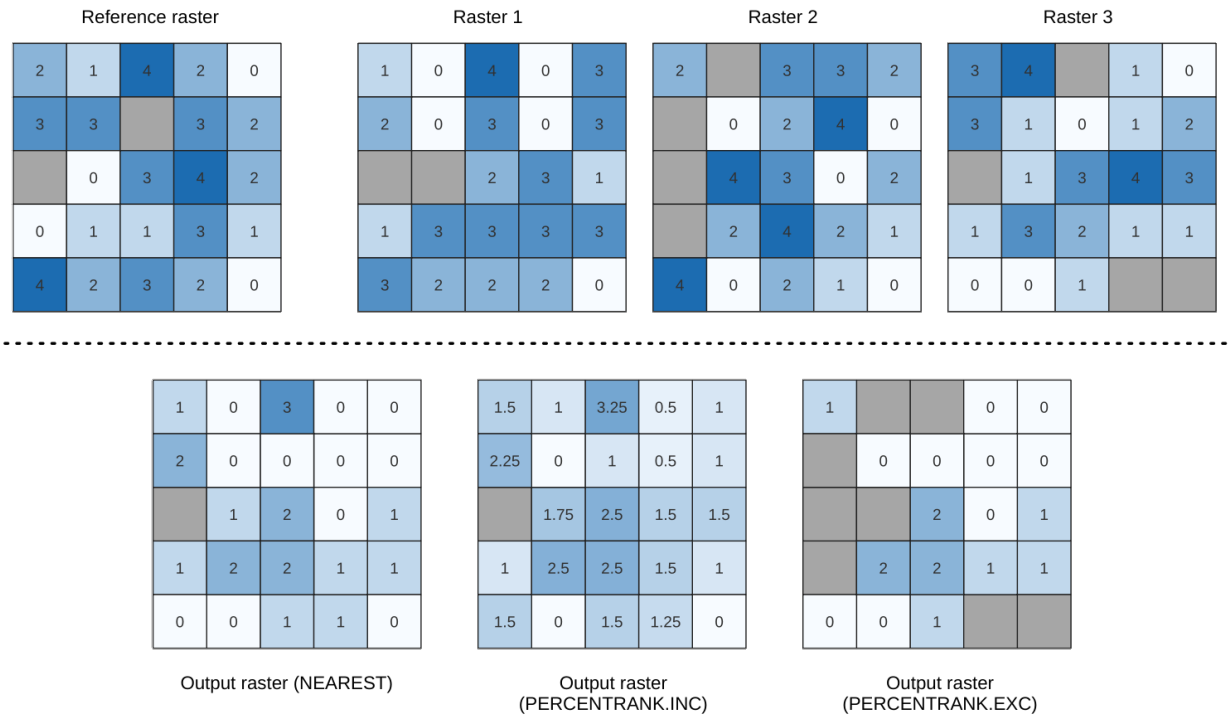



그림 24.50: 백분위수 = 0.25. NoData 셀 (회색) 은 무시합니다.

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layers** | INPUT | [raster] [list] | 평가할 래스터 레이어들입니다. 데이터 래스터 스택에 다중밴드 래스터가 쓰인 경우, 이 알고리즘은 항상 래스터의 첫 번째 밴드에 분석을 수행할 것입니다. |
| **Method** | METHOD | [enumeration] 기본값: 0 | 백분위수 (percentile) 를 계산하기 위한 방법:<br>• 0 —최근접 순위: 지정한 백분위수에 가장 가까운 값을 반환합니다.<br>• 1 —포괄적 선형 보간법 (PERCENTRANK.INC)<br>• 2 —배타적 선형 보간법 (PERCENTRANK.EXC) |
| **Percentile** | VALUE | [numeric: double] 기본값: 0.25 | 입력 래스터의 중첩되고 정렬된 모든 셀 값들의 스택에 있는 각각의 값들 가운데 순위를 매길 0 에서 1 사이의 값 |
| **Ignore NoData values** | IGNORE_NODATA | [boolean] 기본값: True | 이 옵션을 체크하지 않을 경우, 입력 레이어에 있는 모든 NODATA 셀이 산출 래스터에서 NODATA 셀로 산출될 것입니다. |

다음 페이지에 계속

표 24.304 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Reference layer** | REFERENCE_LAYER | [raster] | 산출 래스터를 생성하기 위한 (범위, 좌표계, 픽셀 크기) 기준 레이어 |
| **Output layer** | OUTPUT | [same as input] 기본값: [Save to temporary file] | Specification of the output raster. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output NoData value** | OUTPUT_NODATA_VALU | [numeric: double] 기본값: -9999.0 | 산출 레이어에 있는 NODATA 에 사용할 값 |
| **Creation options** 선택적 | CREATION_OPTIONS (for QGIS <= 3.42, this was CREATE_OPTIONS) | [string] 기본값: '' | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL* 드라이버 옵션 참조) 배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (\|) 로 복수의 옵션들을 분리하십시오. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output layer** | OUTPUT | [raster] | 결과물을 담고 있는 산출 래스터 레이어 |
| **CRS authority identifier** | CRS_AUTHID | [string] | 산출 래스터 레이어의 좌표계 |
| **Extent** | EXTENT | [string] | 산출 래스터 레이어의 공간 영역 |
| **Width in pixels** | WIDTH_IN_PIXELS | [numeric: integer] | 산출 래스터 레이어의 열 개수 |
| **Height in pixels** | HEIGHT_IN_PIXEI | [numeric: integer] | 산출 래스터 레이어의 행 개수 |
| **Total pixel count** | TOTAL_PIXEL_COUNT | [numeric: integer] | 산출 래스터 레이어의 픽셀 개수 |

### 파이썬 코드

**알고리즘 ID**: `native:cellstackpercentile`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 래스터 레이어로부터 셀 스택 백분율 순위 구하기

입력 값 레이어를 바탕으로 래스터 스택의 셀 별 백분율 순위 값을 계산해서 산출 래스터에 작성합니다.

각 셀의 위치에서, 입력 래스터의 중첩되고 정렬된 모든 셀 값들의 스택에 있는 해당 위치의 값들 가운데 값 레이어의 현재 값의 순위를 매깁니다. 지정한 값이 스택 값 분포를 벗어나는 경우, 이 알고리즘은 셀 값들 가운데 해당 값의 순위를 매길 수가 없기 때문에 NODATA 를 반환합니다.

백분위수 (percentile) 를 계산하기 위한 방법이 두 가지 있습니다:

- 포괄적 (inclusive) 선형 보간법 (PERCENTRANK.INC)
- 배타적 (exclusive) 선형 보간법 (PERCENTRANK.EXC)

선형 보간법은 서로 다른 백분위수에 대해 유일한 값을 반환합니다. 이 두 가지 보간법 모두 리브레오피스 또는 마이크로소프트 엑셀이 구현하는 해당 보간법을 따릅니다.

참조 래스터가 산출 래스터의 범위와 해상도를 정의합니다. 입력 래스터 레이어의 셀 크기가 참조 래스터의 셀 크기와 일치하지 않을 경우, 최근접 이웃 리샘플링 메소드를 이용해서 입력 래스터 레이어를 리샘플링할 것입니다. "NODATA 값 무시"파라미터를 설정하지 않았다면 입력 레이어의 모든 NODATA 값을 NODATA 셀 산출물로 산출할 것입니다. 산출 래스터의 데이터 유형은 항상 `Float32` 일 것입니다.

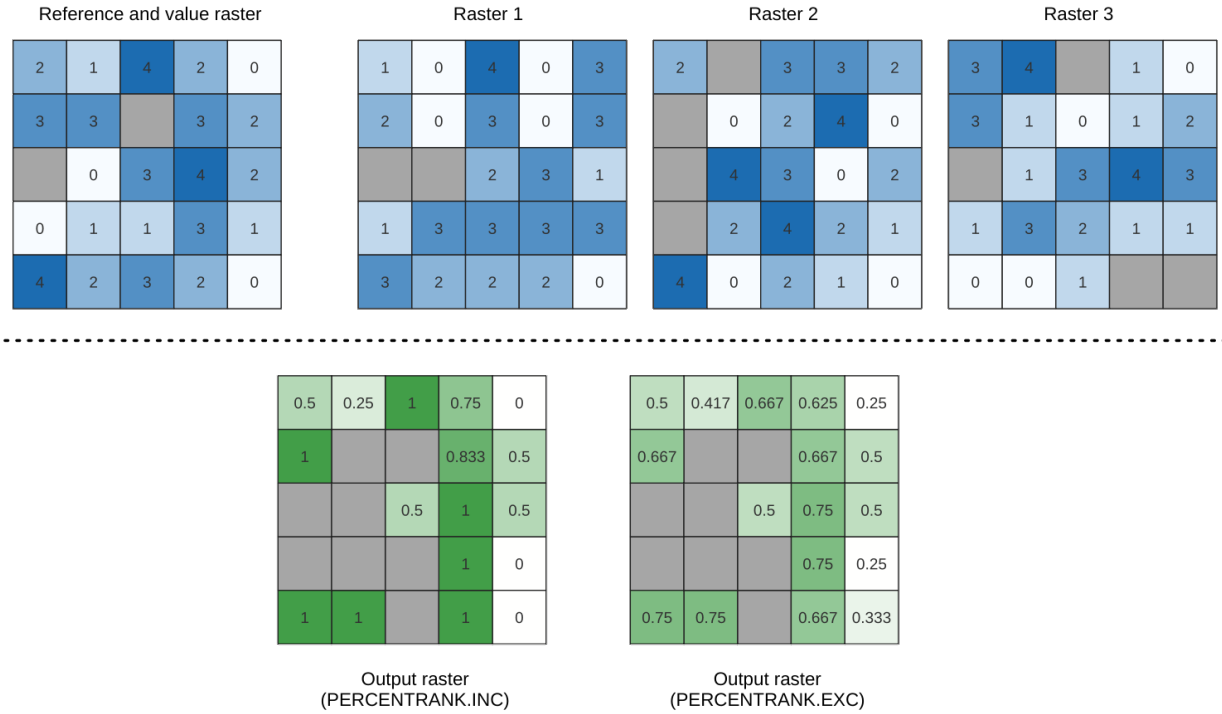



그림 24.51: 값 래스터 레이어 셀의 순위 매기기. `NoData` 셀 (회색) 은 무시합니다.

**더 보기**

셀 스택 백분위수 구하기, 값으로부터 셀 스택 백분율 순위 구하기

### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layers** | INPUT | [raster] [list] | 평가할 래스터 레이어들입니다. 데이터 래스터 스택에 다중밴드 래스터가 쓰인 경우, 이 알고리즘은 항상 래스터의 첫 번째 밴드에 분석을 수행할 것입니다. |
| **Value raster layer** | INPUT_VALUE_RASTE | [raster] | 중첩된 모든 레이어 스택 가운데 값의 순위를 매길 레이어 |
| **Value raster band** | VALUE_RASTER_BA | [numeric: integer]<br>기본값: 1 | 비교할"값 래스터 레이어"의 밴드 |
| **Method** | METHOD | [enumeration]<br>기본값: 0 | 백분위수 (percentile) 를 계산하기 위한 방법:<br>• 0 —포괄적 선형 보간법 (PERCENTRANK.INC)<br>• 1 —배타적 선형 보간법 (PERCENTRANK.EXC) |
| **Ignore NoData values** | IGNORE_NODATA | [boolean]<br>기본값: True | 이 옵션을 체크하지 않을 경우, 입력 레이어에 있는 모든 NODATA 셀이 산출 래스터에서 NODATA 셀로 산출될 것입니다. |
| **Reference layer** | REFERENCE_LAYER | [raster] | 산출 래스터를 생성하기 위한 (범위, 좌표계, 픽셀 크기) 기준 레이어 |
| **Output layer** | OUTPUT | [same as input]<br>기본값: [Save to temporary file] | Specification of the output raster. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

#### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output NoData value** | OUTPUT_NODATA_VALU | [numeric: double]<br>기본값: -9999.0 | 산출 레이어에 있는 NODATA 에 사용할 값 |
| **Creation options**<br>선택적 | CREATION_OPTIONS<br>(for QGIS <= 3.42, this was CREATE_OPTIONS) | [string]<br>기본값: '' | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL* 드라이버 옵션 참조)<br>배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (\|) 로 복수의 옵션들을 분리하십시오. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output layer** | OUTPUT | [raster] | 결과물을 담고 있는 산출 래스터 레이어 |
| **CRS authority identifier** | CRS_AUTHID | [string] | 산출 래스터 레이어의 좌표계 |
| **Extent** | EXTENT | [string] | 산출 래스터 레이어의 공간 영역 |
| **Width in pixels** | WIDTH_IN_PIXELS | [numeric: integer] | 산출 래스터 레이어의 열 개수 |
| **Height in pixels** | HEIGHT_IN_PIXEI | [numeric: integer] | 산출 래스터 레이어의 행 개수 |
| **Total pixel count** | TOTAL_PIXEL_COUNT | [numeric: integer] | 산출 래스터 레이어의 픽셀 개수 |

### 파이썬 코드

**알고리즘 ID**: `native:cellstackpercentrankfromrasterlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 셀 통계

이 알고리즘은 입력 래스터 레이어를 기반으로 셀 당 통계를 계산한 다음 산출 래스터의 각 셀에 산출된 통계를 씁니다. 각 셀의 위치에서, 산출값은 모든 입력 래스터의 중첩된 셀 값들의 함수로 정의됩니다.

입력 레이어 가운데 **하나에만** NODATA 셀이 있어도, 기본적으로 산출 래스터에서도 NODATA 셀로 산출될 것입니다. *Ignore NoData values* 옵션을 활성화한 경우, 통계 계산에서 NODATA 입력을 무시할 것입니다. 이때 해당 위치의 모든 셀이 NODATA 인 경우 NODATA 를 산출할 수도 있습니다.

*Reference layer* 파라미터는 산출 래스터 생성 시 참조물로 사용할 기존 래스터 레이어를 지정합니다. 산출 래스터는 참조 레이어와 동일한 범위, 좌표계, 그리고 픽셀 크기를 가질 것입니다.

**계산 방법**: 입력 레이어의 셀 크기가 참조 래스터 레이어의 셀 크기와 일치하지 않는 경우 입력 레이어를 `nearest neighbor resampling` 방법으로 리샘플링할 것입니다. `Mean`, `Standard deviation` 및 `Variance` 함수 (입력 부동소수점 유형에 따라 항상 `Float32` 또는 `Float64` 유형입니다) 또는 `Count` 및 `Variety` 함수 (항상 `Int32` 유형입니다) 를 이용하는 경우를 제외하면, 산출 래스터 데이터 유형은 입력 데이터셋 가운데 가장 복잡도가 높은 데이터 유형으로 설정될 것입니다.

- `Count`: 개수 통계는 언제나 현재 셀 위치에서 NODATA 값이 아닌 셀의 개수를 산출할 것입니다.
- `Median`: 입력 레이어의 숫자가 짝수일 경우, 정렬된 셀 입력 값의 두 가운데 값의 산술 평균으로 중간값을 계산할 것입니다.
- `Minority/Majority`: 유일한 희귀값이나 최빈값을 찾을 수 없는 경우, 모든 입력 셀의 값이 동등할 경우를 제외하고, NODATA 를 산출합니다.

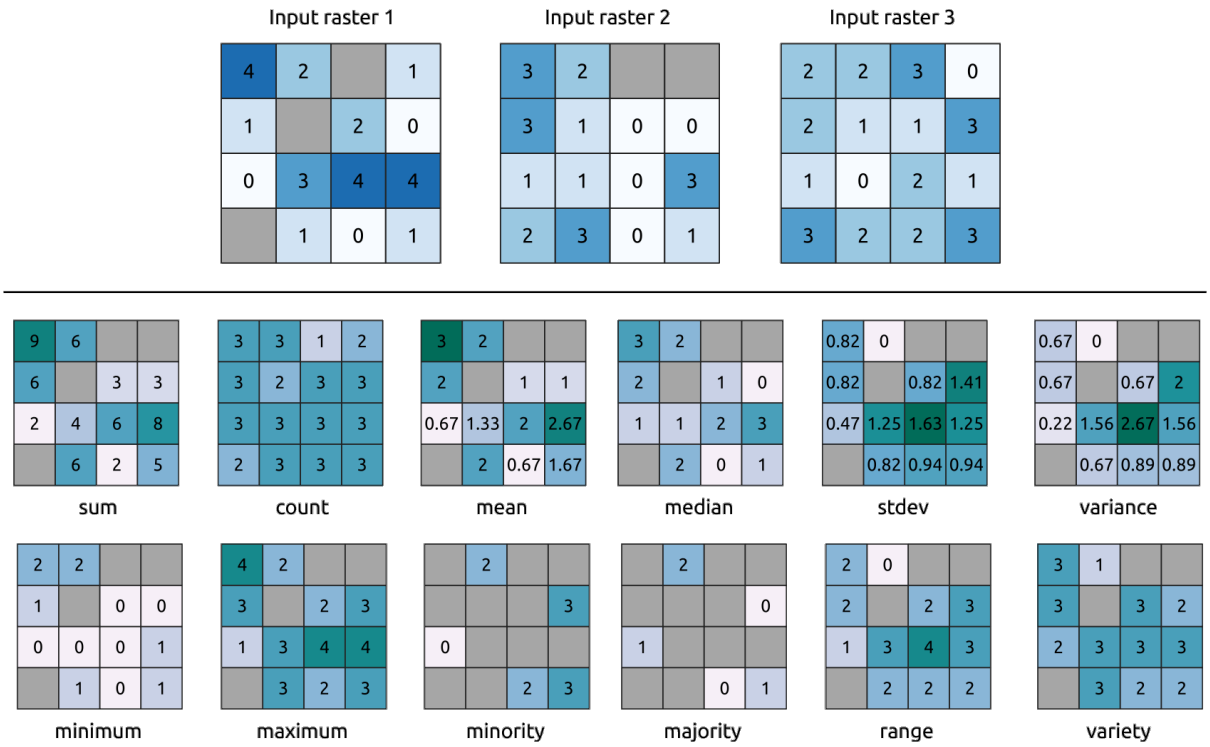



그림 24.52: 모든 통계 함수의 예. NoData (회색) 셀도 계산에 넣습니다.

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layers** | INPUT | [raster] [list] | 입력 래스터 레이어 |
| **Statistic** | STATISTIC | [enumeration] 기본값: 0 | 다음과 같은 통계를 낼 수 있습니다:<br>• 0 —합계 (Sum)<br>• 1 —개수 (Count)<br>• 2 —평균 (Mean)<br>• 3 —중앙값 (Median)<br>• 4 —표준 편차 (Standard deviation)<br>• 5 —분산 (Variance)<br>• 6 —최소값 (Minimum)<br>• 7 —최대값<br>• 8 —희귀값 (Minority, 가장 드문 값)<br>• 9 —최빈값 (Majority, 가장 흔한 값)<br>• 10 —범위 (Range, 최대 ~ 최소)<br>• 11 —다양도 (Variety, 유일값 개수) |
| **Ignore NoData values** | IGNORE_NODATA | [boolean] 기본값: True | NODATA 셀을 무시하고 중첩한 모든 셀에 대해 통계를 계산합니다. |

다음 페이지에 계속

표 24.310 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Reference layer** | `REF_LAYER` | [raster] | 산출 레이어 생성 시 (범위, 좌표계, 픽셀 크기를) 참조할 참조 레이어 |
| **Output layer** | `OUTPUT` | [same as input]<br>기본값: `[Save to temporary file]` | Specification of the output raster. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output NoData value**<br>선택적 | `OUTPUT_NO_DATA_VAI` | [numeric: double]<br>기본값: -9999.0 | 산출 레이어에 있는 NODATA 에 사용할 값 |
| **Creation options**<br>선택적 | `CREATION_OPTIONS` (for QGIS <= 3.42, this was `CREATE_OPTIONS`) | [string]<br>기본값: '' | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL* 드라이버 옵션 참조)<br>배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (\|) 로 복수의 옵션들을 분리하십시오. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **CRS authority identifier** | `CRS_AUTHID` | [crs] | 산출 래스터 레이어의 좌표계 |
| **Extent** | `EXTENT` | [string] | 산출 래스터 레이어의 공간 영역 |
| **Height in pixels** | `HEIGHT_IN_PIXEI` | [numeric: integer] | 산출 래스터 레이어의 행 개수 |
| **Output raster** | `OUTPUT` | [raster] | 결과물을 담고 있는 산출 래스터 레이어 |
| **Total pixel count** | `TOTAL_PIXEL_COUNT` | [numeric: integer] | 산출 래스터 레이어의 픽셀 개수 |
| **Width in pixels** | `WIDTH_IN_PIXELS` | [numeric: integer] | 산출 래스터 레이어의 열 개수 |

### 파이썬 코드

**알고리즘 ID**: `native:cellstatistics`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 빈도와 동일

이 알고리즘은 셀바이셀 (cell-by-cell) 기준으로 입력 래스터 스택의 값들의 빈도 (횟수) 가 값 레이어의 값과 동일한지 평가합니다. 입력 래스터 레이어가 산출 래스터의 범위 및 해상도를 정의하고, 산출 래스터는 언제나 `Int32` 유형입니다.

데이터 래스터 스택에 다중밴드 래스터가 쓰인 경우, 이 알고리즘은 항상 래스터의 첫 번째 밴드에 분석을 수행할 것입니다–다른 밴드를 분석하려면 GDAL 을 이용하십시오. 산출 NODATA 값은 직접 설정할 수 있습니다.

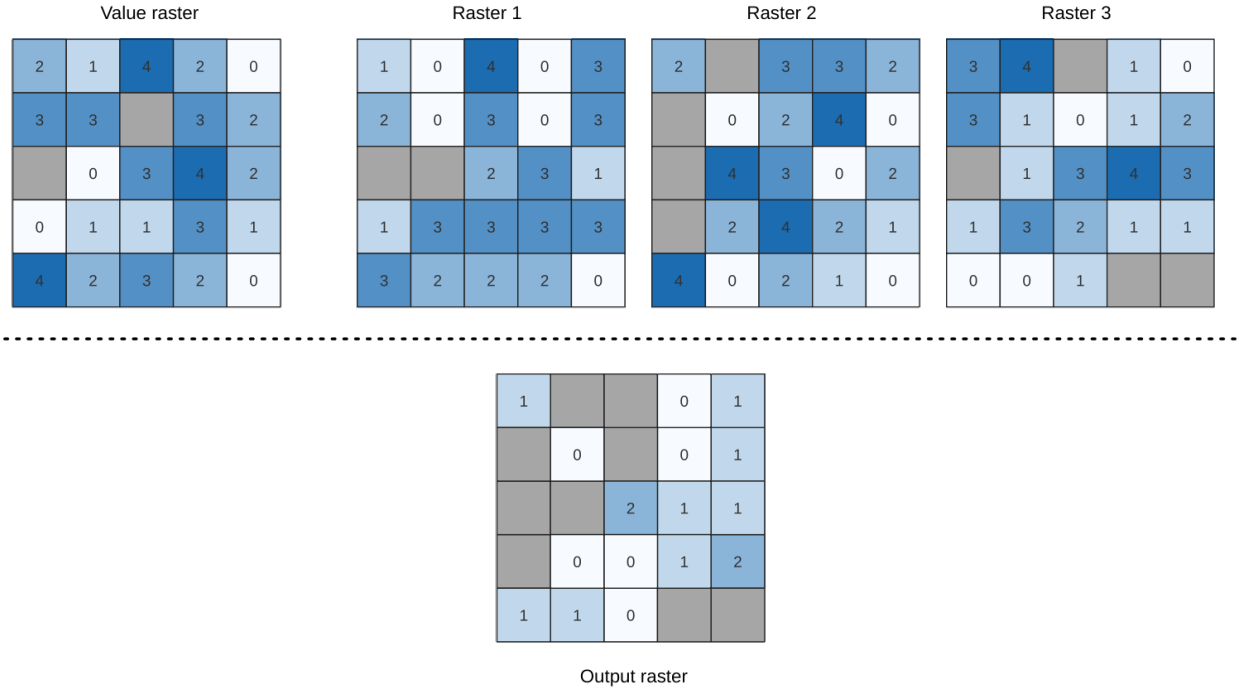



그림 24.53: 산출 래스터에 있는 각 셀의 값은 래스터 목록에서 대응하는 셀의 값이 값 래스터의 해당 셀의 값과 동일한 횟수를 나타냅니다. `NoData` (회색) 셀도 고려합니다.

**더 보기**

빈도 초과, 빈도 미만

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input value raster** | `INPUT_VALUE_RASTE` | [raster] | 입력 값 레이어는 샘플 레이어들을 위한 기준 레이어 역할을 합니다. |
| **Value raster band** | `INPUT_VALUE_RASTE` | [raster band] 기본값: 래스터 레이어의 첫 번째 밴드 | 샘플로 사용하려는 밴드를 선택하십시오. |

다음 페이지에 계속

표 24.313 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input raster layers** | INPUT_RASTERS | [raster] [list] | 평가할 래스터 레이어들입니다. 데이터 래스터 스택에 다중밴드 래스터가 쓰인 경우, 이 알고리즘은 항상 래스터의 첫 번째 밴드에 분석을 수행할 것입니다. |
| **Ignore NoData values** | IGNORE_NODATA | [boolean]<br>기본값: False | 이 옵션을 체크하지 않을 경우, 값 래스터 또는 데이터 레이어 스택에 있는 모든 NODATA 셀이 산출 래스터에서 NODATA 셀로 산출될 것입니다. |
| **Output layer** | OUTPUT | [same as input]<br>기본값: [Save to temporary file] | Specification of the output raster. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output NoData value**<br>선택적 | OUTPUT_NO_DATA_VAI | [numeric: double]<br>기본값: -9999.0 | 산출 레이어에 있는 NODATA 에 사용할 값 |
| **Creation options**<br>선택적 | CREATION_OPTIONS (for QGIS <= 3.42, this was CREATE_OPTIONS) | [string]<br>기본값: '' | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL* 드라이버 옵션 참조) 배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (\|) 로 복수의 옵션들을 분리하십시오. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output layer** | OUTPUT | [raster] | 결과물을 담고 있는 산출 래스터 레이어 |
| **CRS authority identifier** | CRS_AUTHID | [string] | 산출 래스터 레이어의 좌표계 |
| **Extent** | EXTENT | [string] | 산출 래스터 레이어의 공간 영역 |
| **Count of cells with equal value occurrences** | FOUND_LOCATIONS | [numeric: integer] | |
| **Height in pixels** | HEIGHT_IN_PIXEI | [numeric: integer] | 산출 래스터 레이어의 행 개수 |
| **Total pixel count** | TOTAL_PIXEL_COUNT | [numeric: integer] | 산출 래스터 레이어의 픽셀 개수 |
| **Mean frequency at valid cell locations** | MEAN_FREQUENCY_ | [numeric: double] | |

다음 페이지에 계속

표 24.315 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Count of value occurrences** | `OCCURRENCE_COUNT` | [numeric: integer] | |
| **Width in pixels** | `WIDTH_IN_PIXELS` | [numeric: integer] | 산출 래스터 레이어의 열 개수 |

### 파이썬 코드

**알고리즘 ID**: `native:equaltofrequency`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 래스터 퍼지화 (가우스 멤버십)

이 알고리즘은 가우스 멤버십 (gaussian membership) 함수를 통해 입력 래스터의 각 셀에 멤버십 값을 할당해서 입력 래스터를 퍼지화한 래스터로 변형시킵니다. 멤버십 값의 범위는 0 에서 1 까지입니다. 퍼지화된 래스터에서 0 값은 정의된 퍼지 집합 (fuzzy set) 에 속하지 않는다는 의미이며, 1 은 완전히 속한다는 뜻입니다. 가우스 멤버십 함수는 $\mu(x) = e^{-f1*(x-f2)^2}$ 으로 정의되는데, *f1* 이 확산 (spread) 이고 *f2* 는 중점 (midpoint) 입니다.

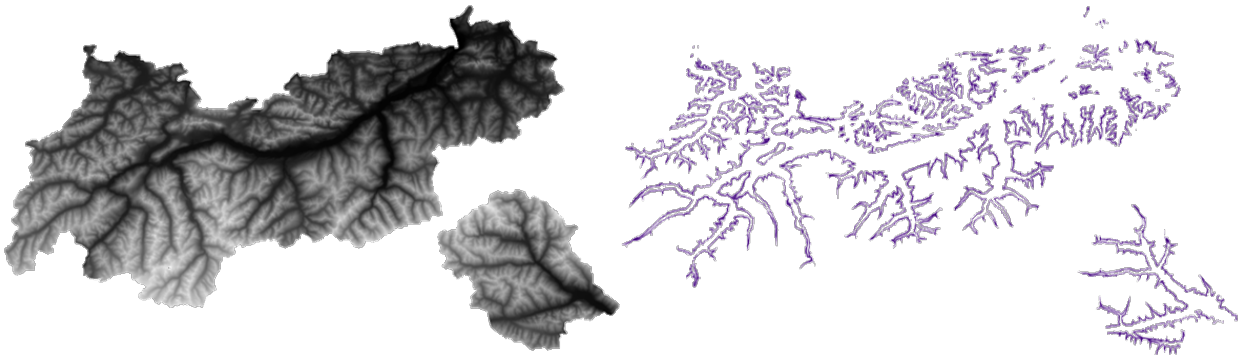


그림 24.54: 래스터 퍼지화의 예. 입력 래스터 소스: 오스트리아 티롤 지역 - data.tirol.gv.at

**더 보기**

래스터 퍼지화 *(큰 값 멤버십)*, 래스터 퍼지화 *(선형 멤버십)*, 래스터 퍼지화 *(근접 멤버십)*, 래스터 퍼지화 *(거듭제곱 멤버십)*, 래스터 퍼지화 *(작은 값 멤버십)*

### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input Raster** | `INPUT` | [raster] | 입력 래스터 레이어 |
| **Band Number** | `BAND` | [raster band]<br>기본값: 래스터 레이어의 첫 번째 밴드 | 래스터가 다중 밴드인 경우, 퍼지화하려는 밴드를 선택하십시오. |
| **Function midpoint** | `FUZZYMIDPOINT` | [numeric: double]<br>기본값: 10.0 | 가우스 함수의 중점 |
| **Function spread** | `FUZZYSPREAD` | [numeric: double]<br>기본값: 0.01 | 가우스 함수의 확산 |
| **Fuzzified raster** | `OUTPUT` | [same as input]<br>기본값: `[Save to temporary file]` | Specification of the output raster. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

#### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Creation options**<br>선택적 | `CREATION_OPTIONS`<br>(for QGIS <= 3.42, this was `CREATE_OPTIONS`) | [string]<br>기본값: ‘’ | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL 드라이버 옵션* 참조)<br>배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (`|`) 로 복수의 옵션들을 분리하십시오. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Fuzzified raster** | `OUTPUT` | [same as input] | 결과물을 담고 있는 산출 래스터 레이어 |
| **CRS authority identifier** | `CRS_AUTHID` | [crs] | 산출 래스터 레이어의 좌표계 |
| **Extent** | `EXTENT` | [string] | 산출 래스터 레이어의 공간 영역 |
| **Width in pixels** | `WIDTH_IN_PIXELS` | [numeric: integer] | 산출 래스터 레이어의 열 개수 |
| **Height in pixels** | `HEIGHT_IN_PIXEL` | [numeric: integer] | 산출 래스터 레이어의 행 개수 |
| **Total pixel count** | `TOTAL_PIXEL_COUNT` | [numeric: integer] | 산출 래스터 레이어의 픽셀 개수 |

### 파이썬 코드

**알고리즘 ID**: `native:fuzzifyrastergaussianmembership`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 래스터 퍼지화 (큰 값 멤버십)

이 알고리즘은 큰 값 멤버십 (large membership) 함수를 통해 입력 래스터의 각 셀에 멤버십 값을 할당해서 입력 래스터를 퍼지화한 래스터로 변형시킵니다. 멤버십 값의 범위는 0 에서 1 까지입니다. 퍼지화된 래스터에서 0 값은 정의된 퍼지 집합 (fuzzy set) 에 속하지 않는다는 의미이며, 1 은 완전히 속한다는 뜻입니다. 큰 값 멤버십 함수는 $\mu(x) = \frac{1}{1+\left(\frac{x}{f2}\right)^{-f1}}$ 로 정의되는데, *f1* 이 확산 (spread) 이고 *f2* 는 중점 (midpoint) 입니다.

**더 보기**

래스터 퍼지화 *(가우스 멤버십)*, 래스터 퍼지화 *(선형 멤버십)*, 래스터 퍼지화 *(근접 멤버십)*, 래스터 퍼지화 *(거듭제곱 멤버십)*, 래스터 퍼지화 *(작은 값 멤버십)*

### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input Raster** | `INPUT` | [raster] | 입력 래스터 레이어 |
| **Band Number** | `BAND` | [raster band]<br>기본값: 래스터 레이어의 첫 번째 밴드 | 래스터가 다중 밴드인 경우, 퍼지화하려는 밴드를 선택하십시오. |
| **Function midpoint** | `FUZZYMIDPOINT` | [numeric: double]<br>Default: 50.0 | 큰 값 함수의 중점 |
| **Function spread** | `FUZZYSPREAD` | [numeric: double]<br>기본값: 5.0 | 큰 값 함수의 확산 |
| **Fuzzified raster** | `OUTPUT` | [same as input]<br>기본값: `[Save to temporary file]` | Specification of the output raster. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Creation options**<br>선택적 | CRE-<br>ATION_OPTIONS<br>(for QGIS <= 3.42, this was CRE-<br>ATE_OPTIONS) | [string]<br>기본값: ' | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL* 드라이버 옵션 참조)<br>배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (\|) 로 복수의 옵션들을 분리하십시오. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Fuzzified raster** | OUTPUT | [same as input] | 결과물을 담고 있는 산출 래스터 레이어 |
| **CRS authority identifier** | CRS_AUTHID | [crs] | 산출 래스터 레이어의 좌표계 |
| **Extent** | EXTENT | [string] | 산출 래스터 레이어의 공간 영역 |
| **Width in pixels** | WIDTH_IN_PIXELS | [numeric: integer] | 산출 래스터 레이어의 열 개수 |
| **Height in pixels** | HEIGHT_IN_PIXEI | [numeric: integer] | 산출 래스터 레이어의 행 개수 |
| **Total pixel count** | TO-<br>TAL_PIXEL_COUNT | [numeric: integer] | 산출 래스터 레이어의 픽셀 개수 |

### 파이썬 코드

**알고리즘 ID**: `native:fuzzifyrasterlargemembership`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 래스터 퍼지화 (선형 멤버십)

이 알고리즘은 선형 멤버십 (linear membership) 함수를 통해 입력 래스터의 각 셀에 멤버십 값을 할당해서 입력 래스터를 퍼지화한 래스터로 변형시킵니다. 멤버십 값의 범위는 0 에서 1 까지입니다. 퍼지화된 래스터에서 0 값은 정의된 퍼지 집합 (fuzzy set) 에 속하지 않는다는 의미이며, 1 은 완전히 속한다는 뜻입니다. 선형 멤버십 함수는

$$\mu(X)\begin{cases}0 & x \le a\\ \dfrac{x-a}{b-a} & a < x < b\\ 1 & x \ge b\end{cases}$$

로 정의되는데, $a$ 가 하한 (low bound) 이고 $b$ 는 상한 (high bound) 입니다. 이 방정식은 하한과 상한 사이에 있는 픽셀 값에 선형 변형을 통해 멤버십 값을 할당합니다. 하한보다 작은 픽셀 값은 0 을 할당받으며, 상한보다 큰 픽셀 값은 1 을 할당받습니다.

**더 보기**

래스터 퍼지화 *(가우스 멤버십)*, 래스터 퍼지화 *(큰 값 멤버십)*, 래스터 퍼지화 *(근접 멤버십)*, 래스터 퍼지화 *(거듭제곱 멤버십)*, 래스터 퍼지화 *(작은 값 멤버십)*

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input Raster** | `INPUT` | [raster] | 입력 래스터 레이어 |
| **Band Number** | `BAND` | [raster band]<br>기본값: 래스터 레이어의 첫 번째 밴드 | 래스터가 다중 밴드인 경우, 퍼지화하려는 밴드를 선택하십시오. |
| **Low fuzzy membership bound** | `FUZZYLOWBOUND` | [numeric: double]<br>기본값: 0.0 | 선형 함수의 하한 |
| **High fuzzy membership bound** | `FUZZYHIGHBOUND` | [numeric: double]<br>기본값: 1.0 | 선형 함수의 상한 |
| **Fuzzified raster** | `OUTPUT` | [same as input]<br>기본값: `[Save to temporary file]` | Specification of the output raster. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Creation options**<br>선택적 | `CREATION_OPTIONS`<br>(for QGIS <= 3.42, this was `CREATE_OPTIONS`) | [string]<br>기본값: ' ' | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL* 드라이버 옵션 참조)<br>배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (`|`) 로 복수의 옵션들을 분리하십시오. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Fuzzified raster** | OUTPUT | [same as input] | 결과물을 담고 있는 산출 래스터 레이어 |
| **CRS authority identifier** | CRS_AUTHID | [crs] | 산출 래스터 레이어의 좌표계 |
| **Extent** | EXTENT | [string] | 산출 래스터 레이어의 공간 영역 |
| **Width in pixels** | WIDTH_IN_PIXELS | [numeric: integer] | 산출 래스터 레이어의 열 개수 |
| **Height in pixels** | HEIGHT_IN_PIXEI | [numeric: integer] | 산출 래스터 레이어의 행 개수 |
| **Total pixel count** | TOTAL_PIXEL_COUNT | [numeric: integer] | 산출 래스터 레이어의 픽셀 개수 |

### 파이썬 코드

**알고리즘 ID**: `native:fuzzifyrasterlinearmembership`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 래스터 퍼지화 (근접 멤버십)

이 알고리즘은 근접 멤버십 (near membership) 함수를 통해 입력 래스터의 각 셀에 멤버십 값을 할당해서 입력 래스터를 퍼지화한 래스터로 변형시킵니다. 멤버십 값의 범위는 0 에서 1 까지입니다. 퍼지화된 래스터에서 0 값은 정의된 퍼지 집합 (fuzzy set) 에 속하지 않는다는 의미이며, 1 은 완전히 속한다는 뜻입니다. 근접 멤버십 함수는 $\mu(x) = \frac{1}{1 + f1 * (x - f2)^2}$ 로 정의되는데, *f1* 이 확산 (spread) 이고 *f2* 는 중점 (midpoint) 입니다.

**더 보기**

래스터 퍼지화 *(가우스 멤버십)*, 래스터 퍼지화 *(큰 값 멤버십)*, 래스터 퍼지화 *(선형 멤버십)*, 래스터 퍼지화 *(거듭제곱 멤버십)*, 래스터 퍼지화 *(작은 값 멤버십)*

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input Raster** | INPUT | [raster] | 입력 래스터 레이어 |

다음 페이지에 계속

표 24.325 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Band Number** | BAND | [raster band]<br>기본값: 래스터 레이어의 첫 번째 밴드 | 래스터가 다중 밴드인 경우, 퍼지화하려는 밴드를 선택하십시오. |
| **Function midpoint** | FUZZYMIDPOINT | [numeric: double]<br>Default: 50.0 | 근접 함수의 중점 |
| **Function spread** | FUZZYSPREAD | [numeric: double]<br>기본값: 0.01 | 근접 함수의 확산 |
| **Fuzzified raster** | OUTPUT | [same as input]<br>기본값: [Save to temporary file] | Specification of the output raster. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Creation options**<br>선택적 | CREATION_OPTIONS (for QGIS <= 3.42, this was CREATE_OPTIONS) | [string]<br>기본값: | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL* 드라이버 옵션 참조) 배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (\|) 로 복수의 옵션들을 분리하십시오. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Fuzzified raster** | OUTPUT | [same as input] | 결과물을 담고 있는 산출 래스터 레이어 |
| **CRS authority identifier** | CRS_AUTHID | [crs] | 산출 래스터 레이어의 좌표계 |
| **Extent** | EXTENT | [string] | 산출 래스터 레이어의 공간 영역 |
| **Width in pixels** | WIDTH_IN_PIXELS | [numeric: integer] | 산출 래스터 레이어의 열 개수 |
| **Height in pixels** | HEIGHT_IN_PIXEI | [numeric: integer] | 산출 래스터 레이어의 행 개수 |
| **Total pixel count** | TOTAL_PIXEL_COUNT | [numeric: integer] | 산출 래스터 레이어의 픽셀 개수 |

### 파이썬 코드

**알고리즘 ID**: `native:fuzzifyrasternearmembership`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 래스터 퍼지화 (거듭제곱 멤버십)

이 알고리즘은 거듭제곱 멤버십 (power membership) 함수를 통해 입력 래스터의 각 셀에 멤버십 값을 할당해서 입력 래스터를 퍼지화한 래스터로 변형시킵니다. 멤버십 값의 범위는 0 에서 1 까지입니다. 퍼지화된 래스터에서 0 값은 정의된 퍼지 집합 (fuzzy set) 에 속하지 않는다는 의미이며, 1 은 완전히 속한다는 뜻입니다. 거듭제곱 멤버십 함수는 $\mu(x)\begin{cases}0 & x \le a\\ \left(\frac{x-a}{b-a}\right)^{f1} & a < x < b\\ 1 & x \ge b\end{cases}$ 로 정의되는데, $a$ 가 하한 (low bound) 이고 $b$ 는 상한 (high bound) 이며, $f1$ 은 지수 (exponent) 입니다. 이 방정식은 하한과 상한 사이에 있는 픽셀 값에 거듭제곱 변형을 통해 멤버십 값을 할당합니다. 하한보다 작은 픽셀 값은 0 을 할당받으며, 상한보다 큰 픽셀 값은 1 을 할당받습니다.

**더 보기**

래스터 퍼지화 *(가우스 멤버십)*, 래스터 퍼지화 *(큰 값 멤버십)*, 래스터 퍼지화 *(선형 멤버십)*, 래스터 퍼지화 *(근접 멤버십)*, 래스터 퍼지화 *(작은 값 멤버십)*

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input Raster** | `INPUT` | [raster] | 입력 래스터 레이어 |
| **Band Number** | `BAND` | [raster band]<br>기본값: 래스터 레이어의 첫 번째 밴드 | 래스터가 다중 밴드인 경우, 퍼지화하려는 밴드를 선택하십시오. |
| **Low fuzzy membership bound** | `FUZZYLOWBOUND` | [numeric: double]<br>기본값: 0.0 | 거듭제곱 함수의 하한 |
| **High fuzzy membership bound** | `FUZZYHIGHBOUND` | [numeric: double]<br>기본값: 1.0 | 거듭제곱 함수의 상한 |
| **High fuzzy membership bound** | `FUZZYEXPONENT` | [numeric: double]<br>기본값: 2.0 | 거듭제곱 함수의 지수 |
| **Fuzzified raster** | `OUTPUT` | [same as input]<br>기본값: `[Save to temporary file]` | Specification of the output raster. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Creation options**<br>선택적 | CREATION_OPTIONS<br>(for QGIS <= 3.42, this was CREATE_OPTIONS) | [string]<br>기본값: ' ' | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL* 드라이버 옵션 참조)<br>배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (\|) 로 복수의 옵션들을 분리하십시오. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Fuzzified raster** | OUTPUT | [same as input] | 결과물을 담고 있는 산출 래스터 레이어 |
| **CRS authority identifier** | CRS_AUTHID | [crs] | 산출 래스터 레이어의 좌표계 |
| **Extent** | EXTENT | [string] | 산출 래스터 레이어의 공간 영역 |
| **Width in pixels** | WIDTH_IN_PIXELS | [numeric: integer] | 산출 래스터 레이어의 열 개수 |
| **Height in pixels** | HEIGHT_IN_PIXEI | [numeric: integer] | 산출 래스터 레이어의 행 개수 |
| **Total pixel count** | TOTAL_PIXEL_COUNT | [numeric: integer] | 산출 래스터 레이어의 픽셀 개수 |

### 파이썬 코드

**알고리즘 ID**: `native:fuzzifyrasterpowermembership`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 래스터 퍼지화 (작은 값 멤버십)

이 알고리즘은 작은 값 멤버십 (small membership) 함수를 통해 입력 래스터의 각 셀에 멤버십 값을 할당해서 입력 래스터를 퍼지화한 래스터로 변형시킵니다. 멤버십 값의 범위는 0 에서 1 까지입니다. 퍼지화된 래스터에서 0 값은 정의된 퍼지 집합 (fuzzy set) 에 속하지 않는다는 의미이며, 1 은 완전히 속한다는 뜻입니다. 작은 값 멤버십 함수는

$$\mu(x) = \frac{1}{1 + \left(\frac{x}{f2}\right)^{f1}}$$

로 정의되는데, *f1* 이 확산 (spread) 이고 *f2* 는 중점 (midpoint) 입니다.

**더 보기**

래스터 퍼지화 (가우스 멤버십), 래스터 퍼지화 (큰 값 멤버십) 래스터 퍼지화 (선형 멤버십), 래스터 퍼지화 (근접 멤버십), 래스터 퍼지화 (거듭제곱 멤버십)

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input Raster** | `INPUT` | [raster] | 입력 래스터 레이어 |
| **Band Number** | `BAND` | [raster band]<br>기본값: 래스터 레이어의 첫 번째 밴드 | 래스터가 다중 밴드인 경우, 퍼지화하려는 밴드를 선택하십시오. |
| **Function midpoint** | `FUZZYMIDPOINT` | [numeric: double]<br>Default: 50.0 | 작은 값 함수의 중점 |
| **Function spread** | `FUZZYSPREAD` | [numeric: double]<br>기본값: 5.0 | 작은 값 함수의 확산 |
| **Fuzzified raster** | `OUTPUT` | [same as input]<br>기본값: `[Save to temporary file]` | Specification of the output raster. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Creation options**<br>선택적 | `CRE-ATION_OPTIONS`<br>(for QGIS <= 3.42, this was `CRE-ATE_OPTIONS`) | [string]<br>기본값: ‘’ | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL* 드라이버 옵션 참조)<br>배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (`|`) 로 복수의 옵션들을 분리하십시오. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Fuzzified raster** | `OUTPUT` | [same as input] | 결과물을 담고 있는 산출 래스터 레이어 |
| **CRS authority identifier** | `CRS_AUTHID` | [crs] | 산출 래스터 레이어의 좌표계 |
| **Extent** | `EXTENT` | [string] | 산출 래스터 레이어의 공간 영역 |
| **Width in pixels** | `WIDTH_IN_PIXELS` | [numeric: integer] | 산출 래스터 레이어의 열 개수 |

다음 페이지에 계속

표 24.333 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Height in pixels** | `HEIGHT_IN_PIXEL` | [numeric: integer] | 산출 래스터 레이어의 행 개수 |
| **Total pixel count** | `TOTAL_PIXEL_COUNT` | [numeric: integer] | 산출 래스터 레이어의 픽셀 개수 |

### 파이썬 코드

**알고리즘 ID**: `native:fuzzifyrastersmallmembership`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 빈도 초과

이 알고리즘은 셀바이셀 (cell-by-cell) 기준으로 입력 래스터 스택의 값들의 빈도 (횟수) 가 값 래스터의 값을 초과하는지 평가합니다. 입력 래스터 레이어가 산출 래스터의 범위 및 해상도를 정의하고, 산출 래스터는 언제나 `Int32` 유형입니다.

데이터 래스터 스택에 다중밴드 래스터가 쓰인 경우, 이 알고리즘은 항상 래스터의 첫 번째 밴드에 분석을 수행할 것입니다–다른 밴드를 분석하려면 GDAL 을 이용하십시오. 산출 NODATA 값은 직접 설정할 수 있습니다.

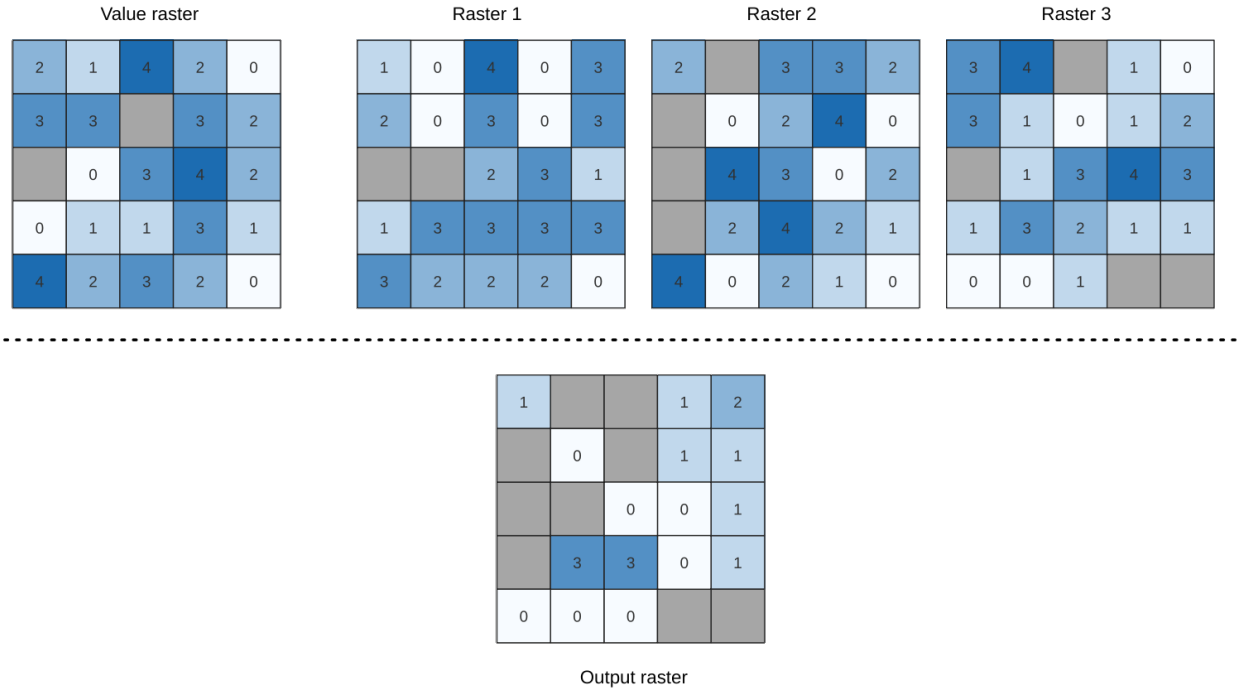



그림 24.55: 산출 래스터에 있는 각 셀의 값은 래스터 목록에서 대응하는 셀의 값이 값 래스터의 해당 셀의 값을 초과하는 횟수를 나타냅니다. `NoData` (회색) 셀도 고려합니다.

더 보기

빈도와 동일, 빈도 미만

### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input value raster** | INPUT_VALUE_RASTE | [raster] | 입력 값 레이어는 샘플 레이어들을 위한 기준 레이어 역할을 합니다. |
| **Value raster band** | INPUT_VALUE_RASTE | [raster band]<br>기본값: 래스터 레이어의 첫 번째 밴드 | 샘플로 사용하려는 밴드를 선택하십시오. |
| **Input raster layers** | INPUT_RASTERS | [raster] [list] | 평가할 래스터 레이어들입니다. 데이터 래스터 스택에 다중밴드 래스터가 쓰인 경우, 이 알고리즘은 항상 래스터의 첫 번째 밴드에 분석을 수행할 것입니다. |
| **Ignore NoData values** | IGNORE_NODATA | [boolean]<br>기본값: False | 이 옵션을 체크하지 않을 경우, 값 래스터 또는 데이터 레이어 스택에 있는 모든 NODATA 셀이 산출 래스터에서 NODATA 셀로 산출될 것입니다. |
| **Output layer** | OUTPUT | [same as input]<br>기본값: [Save to temporary file] | Specification of the output raster. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

#### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output NoData value**<br>선택적 | OUTPUT_NO_DATA_VAI | [numeric: double]<br>기본값: -9999.0 | 산출 레이어에 있는 NODATA 에 사용할 값 |
| **Creation options**<br>선택적 | CREATION_OPTIONS (for QGIS <= 3.42, this was CREATE_OPTIONS) | [string]<br>기본값: '' | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL* 드라이버 옵션 참조)<br>배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (\|) 로 복수의 옵션들을 분리하십시오. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output layer** | OUTPUT | [raster] | 결과물을 담고 있는 산출 래스터 레이어 |
| **CRS authority identifier** | CRS_AUTHID | [string] | 산출 래스터 레이어의 좌표계 |
| **Extent** | EXTENT | [string] | 산출 래스터 레이어의 공간 영역 |
| **Count of cells with equal value occurrences** | FOUND_LOCATIONS | [numeric: integer] | |
| **Height in pixels** | HEIGHT_IN_PIXEI | [numeric: integer] | 산출 래스터 레이어의 행 개수 |
| **Total pixel count** | TOTAL_PIXEL_COUNT | [numeric: integer] | 산출 래스터 레이어의 픽셀 개수 |
| **Mean frequency at valid cell locations** | MEAN_FREQUENCY_ | [numeric: double] | |
| **Count of value occurrences** | OCCURRENCE_COUNT | [numeric: integer] | |
| **Width in pixels** | WIDTH_IN_PIXELS | [numeric: integer] | 산출 래스터 레이어의 열 개수 |

### 파이썬 코드

**알고리즘 ID**: `native:greaterthanfrequency`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

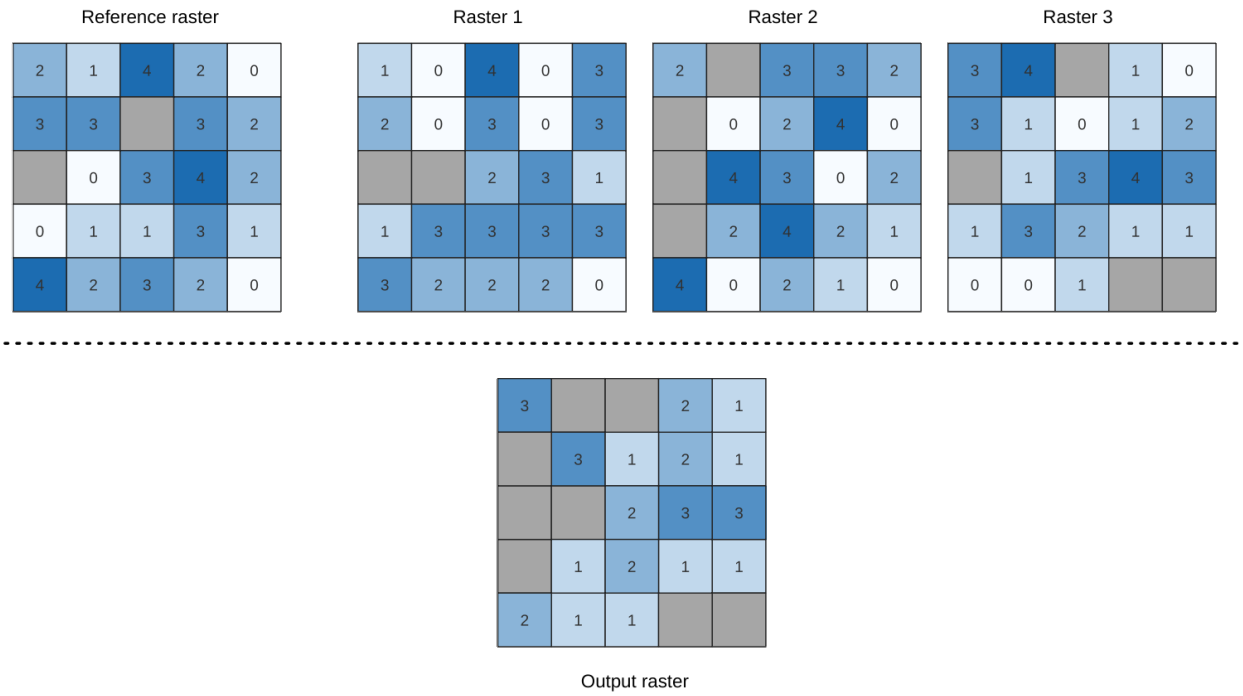
## 래스터 스택에서 가장 높은 위치

이 알고리즘은 셀바이셀 (cell-by-cell) 기준으로 래스터 스택에서 가장 높은 값을 가진 래스터의 위치를 평가합니다. 위치는 1 로 시작해서 입력 래스터의 총 개수까지 셉니다. 입력 래스터들의 순서와 알고리즘은 관계가 있습니다. 가장 높은 값을 가진 래스터가 여러 개일 경우, 첫 번째 래스터를 위치 값으로 사용할 것입니다.

데이터 래스터 스택에 다중밴드 래스터가 쓰인 경우, 이 알고리즘은 항상 래스터의 첫 번째 밴드에 분석을 수행할 것입니다 – 다른 밴드를 분석하려면 GDAL 을 이용하십시오. “ignore NoData”파라미터를 체크하지 않을 경우, 래스터 레이어 스택에 있는 모든 NODATA 셀이 산출 래스터에서 NODATA 셀로 산출될 것입니다. 산출 NODATA 값은 직접 설정할 수 있습니다. 기준 래스터 레이어가 산출 래스터의 범위 및 해상도를 정의하고, 산출 래스터는 언제나 `Int32` 유형입니다.

**더 보기**

래스터 스택에서 가장 낮은 위치




### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input raster layers** | `INPUT_RASTERS` | [raster] [list] | 비교할 래스터 레이어들의 목록 |
| **Reference layer** | `REFERENCE_LAYER` | [raster] | 산출 래스터를 생성하기 위한 (범위, 좌표계, 픽셀 크기) 기준 레이어 |
| **Ignore NoData values** | `IGNORE_NODATA` | [boolean]<br>기본값: False | 이 옵션을 체크하지 않을 경우, 데이터 레이어 스택에 있는 모든 NODATA 셀이 산출 래스터에서 NODATA 셀로 산출될 것입니다. |
| **Output layer** | `OUTPUT` | [raster]<br>기본값: `[Save to temporary file]` | Specification of the output raster containing the result. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output NoData value** | `OUTPUT_NODATA_VALU` | [numeric: double]<br>기본값: -9999.0 | 산출 레이어에 있는 NODATA 에 사용할 값 |
| **Creation options**<br>선택적 | `CREATION_OPTIONS`<br>(for QGIS <= 3.42, this was `CREATE_OPTIONS`) | [string]<br>기본값: ' | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL 드라이버 옵션* 참조)<br>배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (\|) 로 복수의 옵션들을 분리하십시오. |

### 파이썬 코드

**알고리즘 ID**: `native:highestpositioninrasterstack`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 빈도 미만

이 알고리즘은 셀바이셀 (cell-by-cell) 기준으로 입력 래스터 스택의 값들의 빈도 (횟수) 가 값 래스터의 값 미만인지 평가합니다. 입력 래스터 레이어가 산출 래스터의 범위 및 해상도를 정의하고, 산출 래스터는 언제나 `Int32` 유형입니다.

데이터 래스터 스택에 다중밴드 래스터가 쓰인 경우, 이 알고리즘은 항상 래스터의 첫 번째 밴드에 분석을 수행할 것입니다–다른 밴드를 분석하려면 GDAL 을 이용하십시오. 산출 NODATA 값은 직접 설정할 수 있습니다.

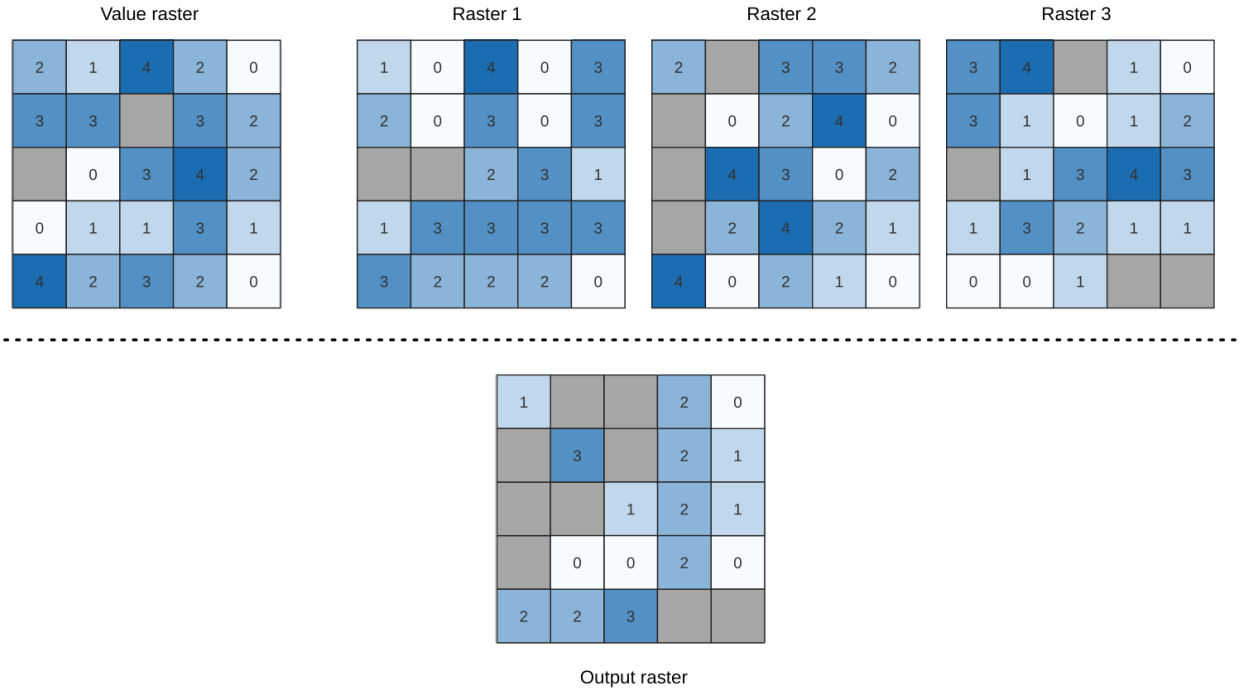



그림 24.56: 산출 래스터에 있는 각 셀의 값은 래스터 목록에서 대응하는 셀의 값이 값 래스터의 해당 셀의 값에 미만하는 횟수를 나타냅니다. `NoData` (회색) 셀도 고려합니다.

**더 보기**

빈도와 동일, 빈도 초과

### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input value raster** | INPUT_VALUE_RASTE | [raster] | 입력 값 레이어는 샘플 레이어들을 위한 기준 레이어 역할을 합니다. |
| **Value raster band** | INPUT_VALUE_RASTE | [raster band]<br>기본값: 래스터 레이어의 첫 번째 밴드 | 샘플로 사용하려는 밴드를 선택하십시오. |
| **Input raster layers** | INPUT_RASTERS | [raster] [list] | 평가할 래스터 레이어들입니다. 데이터 래스터 스택에 다중밴드 래스터가 쓰인 경우, 이 알고리즘은 항상 래스터의 첫 번째 밴드에 분석을 수행할 것입니다. |
| **Ignore NoData values** | IGNORE_NODATA | [boolean]<br>기본값: False | 이 옵션을 체크하지 않을 경우, 값 래스터 또는 데이터 레이어 스택에 있는 모든 NODATA 셀이 산출 래스터에서 NODATA 셀로 산출될 것입니다. |
| **Output layer** | OUTPUT | [same as input]<br>기본값: [Save to temporary file] | Specification of the output raster. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

#### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output NoData value**<br>선택적 | OUTPUT_NO_DATA_VAI | [numeric: double]<br>기본값: -9999.0 | 산출 레이어에 있는 NODATA 에 사용할 값 |
| **Creation options**<br>선택적 | CREATION_OPTIONS<br>(for QGIS <= 3.42, this was CREATE_OPTIONS) | [string]<br>기본값: '' | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL* 드라이버 옵션 참조)<br>배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (\|) 로 복수의 옵션들을 분리하십시오. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output layer** | OUTPUT | [raster] | 결과물을 담고 있는 산출 래스터 레이어 |
| **CRS authority identifier** | CRS_AUTHID | [string] | 산출 래스터 레이어의 좌표계 |
| **Extent** | EXTENT | [string] | 산출 래스터 레이어의 공간 영역 |
| **Count of cells with equal value occurrences** | FOUND_LOCATIONS | [numeric: integer] | |
| **Height in pixels** | HEIGHT_IN_PIXEI | [numeric: integer] | 산출 래스터 레이어의 행 개수 |
| **Total pixel count** | TOTAL_PIXEL_COUNT | [numeric: integer] | 산출 래스터 레이어의 픽셀 개수 |
| **Mean frequency at valid cell locations** | MEAN_FREQUENCY_ | [numeric: double] | |
| **Count of value occurrences** | OCCURRENCE_COUNT | [numeric: integer] | |
| **Width in pixels** | WIDTH_IN_PIXELS | [numeric: integer] | 산출 래스터 레이어의 열 개수 |

### 파이썬 코드

**알고리즘 ID**: `native:lessthanfrequency`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

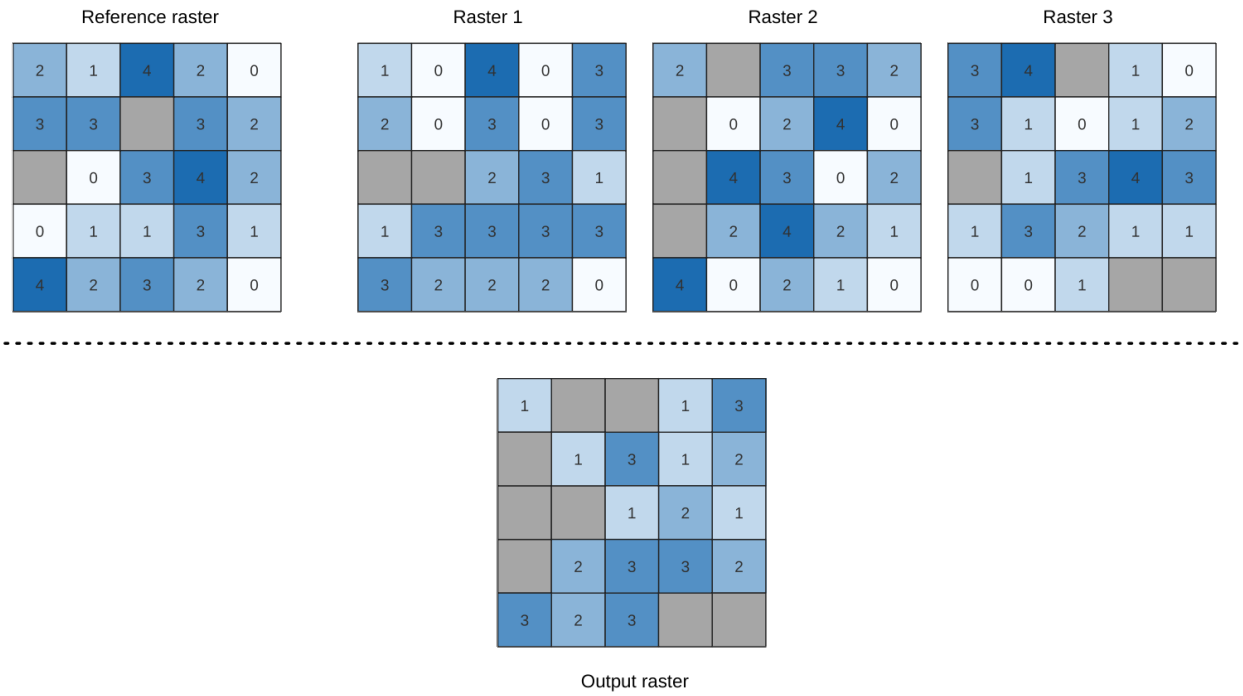
## 래스터 스택에서 가장 낮은 위치

이 알고리즘은 셀바이셀 (cell-by-cell) 기준으로 래스터 스택에서 가장 낮은 값을 가진 래스터의 위치를 평가합니다. 위치는 1 로 시작해서 입력 래스터의 총 개수까지 셉니다. 입력 래스터들의 순서와 알고리즘은 관계가 있습니다. 가장 낮은 값을 가진 래스터가 여러 개일 경우, 첫 번째 래스터를 위치 값으로 사용할 것입니다.

데이터 래스터 스택에 다중밴드 래스터가 쓰인 경우, 이 알고리즘은 항상 래스터의 첫 번째 밴드에 분석을 수행할 것입니다 – 다른 밴드를 분석하려면 GDAL 을 이용하십시오. “ignore NoData”파라미터를 체크하지 않을 경우, 래스터 레이어 스택에 있는 모든 NODATA 셀이 산출 래스터에서 NODATA 셀로 산출될 것입니다. 산출 NODATA 값은 직접 설정할 수 있습니다. 기준 래스터 레이어가 산출 래스터의 범위 및 해상도를 정의하고, 산출 래스터는 언제나 `Int32` 유형입니다.

**더 보기**

래스터 스택에서 가장 높은 위치




### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input raster layers** | `INPUT_RASTERS` | [raster] [list] | 비교할 래스터 레이어들의 목록 |
| **Reference layer** | `REFERENCE_LAYER` | [raster] | 산출 래스터를 생성하기 위한 (범위, 좌표계, 픽셀 크기) 기준 레이어 |
| **Ignore NoData values** | `IGNORE_NODATA` | [boolean]<br>기본값: False | 이 옵션을 체크하지 않을 경우, 데이터 레이어 스택에 있는 모든 NODATA 셀이 산출 래스터에서 NODATA 셀로 산출될 것입니다. |
| **Output layer** | `OUTPUT` | [raster]<br>기본값: `[Save to temporary file]` | Specification of the output raster containing the result. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output NoData value** | `OUTPUT_NODATA_VALU` | [numeric: double]<br>기본값: -9999.0 | 산출 레이어에 있는 NODATA 에 사용할 값 |
| **Creation options**<br>선택적 | `CREATION_OPTIONS`<br>(for QGIS <= 3.42, this was `CREATE_OPTIONS`) | [string]<br>기본값: ' | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL 드라이버 옵션* 참조)<br>배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (\|) 로 복수의 옵션들을 분리하십시오. |

### 파이썬 코드

**알고리즘 ID**: `native:lowestpositioninrasterstack`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 래스터 불 (boolean) AND

입력 래스터 집합에 대해 불 (boolean) `AND` 를 계산합니다. 모든 입력 래스터들이 어떤 픽셀에 대해 0 이 아닌 값을 가지고 있다면, 산출 래스터에서 해당 픽셀의 값을 `1` 로 설정할 것입니다. 입력 래스터 가운데 하나라도 어떤 픽셀에 대해 `0` 값을 가지고 있다면, 산출 래스터에서 해당 픽셀의 값을 `0` 으로 설정할 것입니다.

참조 레이어 파라미터는 산출 래스터 생성 시 참조물로 사용할 기존 래스터 레이어를 지정합니다. 산출 래스터는 참조 레이어와 동일한 범위, 좌표계, 그리고 픽셀 크기를 가질 것입니다.

입력 레이어 가운데 **하나에만** NODATA 픽셀이 있어도, 기본적으로 산출 래스터에서도 NODATA 픽셀로 산출될 것입니다. *Treat nodata values as false* 옵션을 활성화한 경우, NODATA 입력을 `0` 입력값과 동일하게 취급할 것입니다.

**더 보기**

래스터 불 *(boolean) OR*

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layers** | `INPUT` | [raster] [list] | 입력 래스터 레이어의 목록 |
| **Reference layer** | `REF_LAYER` | [raster] | 산출 레이어 생성 시 (범위, 좌표계, 픽셀 크기를) 참조할 참조 레이어 |
| **Treat NoData values as false** | `NODATA_AS_FALSE` | [boolean]<br>기본값: False | 작업 수행 시 입력 파일에 있는 NODATA 값을 0 으로 취급 |
| **Output layer** | `OUTPUT` | [raster]<br>기본값: `[Save to temporary file]` | Specification of the output raster containing the result. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output NoData value** | NO_DATA | [numeric: double]<br>기본값: -9999.0 | 산출 레이어에 있는 NODATA 에 사용할 값 |
| **Output data type** | DATA_TYPE | [enumeration]<br>기본값: 5 | 산출 래스터 데이터 유형입니다. 다음 가운데 하나를 선택할 수 있습니다:<br>• 0 —Byte (8 비트 부호 없는 정수 (quint8))<br>• 1 —Int16 (16 비트 부호 있는 정수 (qint16))<br>• 2 —UInt16 (16 비트 부호 없는 정수 (quint16))<br>• 3 —Int32 (32 비트 부호 있는 정수 (qint32))<br>• 4 —UInt32 (32 비트 부호 없는 정수 (quint32))<br>• 5 —Float32 (32 비트 부동소수점형 (float))<br>• 6 —Float64 (64 비트 부동소수점형 (double))<br>• 7 —CInt16 (Complex Int16)<br>• 8 —CInt32 (Complex Int32)<br>• 9 —CFloat32 (Complex Float32)<br>• 10 —CFloat64 (Complex Float64)<br>• 11 —Int8 (8 비트 부호 있는 정수 (qint8))<br>QGIS 와 함께 빌드된 GDAL 버전에 따라 사용할 수 있는 옵션이 달라집니다. (*Help* ▯ *About* 메뉴 참조) |
| **Creation options**<br>선택적 | CREATION_OPTIONS<br>(for QGIS <= 3.42, this was CREATE_OPTIONS) | [string]<br>기본값: ‘’ | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL 드라이버 옵션* 참조)<br>배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (\|) 로 복수의 옵션들을 분리하십시오. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Extent** | EXTENT | [string] | 산출 래스터 레이어의 공간 영역 |
| **CRS authority identifier** | CRS_AUTHID | [crs] | 산출 래스터 레이어의 좌표계 |
| **Width in pixels** | WIDTH_IN_PIXELS | [numeric: integer] | 산출 래스터 레이어의 열 개수 |
| **Height in pixels** | HEIGHT_IN_PIXEL | [numeric: integer] | 산출 래스터 레이어의 행 개수 |

다음 페이지에 계속

표 24.348 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Total pixel count** | TOTAL_PIXEL_COUNT | [numeric: integer] | 산출 래스터 레이어의 픽셀 개수 |
| **NoData pixel count** | NODATA_PIXEL_COUN | [numeric: integer] | 산출 래스터 레이어에 있는 NODATA 픽셀 개수 |
| **True pixel count** | TRUE_PIXEL_COUN | [numeric: integer] | 산출 래스터 레이어에 있는 (값이 1 인) 참 픽셀의 개수 |
| **False pixel count** | FALSE_PIXEL_COU | [numeric: integer] | 산출 래스터 레이어에 있는 (값이 0 인) 거짓 픽셀의 개수 |
| **Output layer** | OUTPUT | [raster] | 결과물을 담고 있는 산출 래스터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:rasterbooleanand`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 래스터 불 (boolean) OR

입력 래스터 집합에 대해 불 (boolean) `OR` 를 계산합니다. 모든 입력 래스터들이 어떤 픽셀에 대해 0 값을 가지고 있다면, 산출 래스터에서 해당 픽셀의 값을 `0` 으로 설정할 것입니다. 입력 래스터 가운데 하나라도 어떤 픽셀에 대해 `1` 값을 가지고 있다면, 산출 래스터에서 해당 픽셀의 값을 `1` 로 설정할 것입니다.

참조 레이어 파라미터는 산출 래스터 생성 시 참조물로 사용할 기존 래스터 레이어를 지정합니다. 산출 래스터는 참조 레이어와 동일한 범위, 좌표계, 그리고 픽셀 크기를 가질 것입니다.

입력 레이어 가운데 **하나에만** NODATA 픽셀이 있어도, 기본적으로 산출 래스터에서도 NODATA 픽셀로 산출될 것입니다. *Treat nodata values as false* 옵션을 활성화한 경우, NODATA 입력을 `0` 입력값과 동일하게 취급할 것입니다.

**더 보기**

래스터 불 *(boolean) AND*

### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layers** | INPUT | [raster] [list] | 입력 래스터 레이어의 목록 |
| **Reference layer** | REF_LAYER | [raster] | 산출 레이어 생성 시 (범위, 좌표계, 픽셀 크기를) 참조할 참조 레이어 |

다음 페이지에 계속

표 24.349 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Treat NoData values as false** | `NODATA_AS_FALSE` | [boolean]<br>기본값: False | 작업 수행 시 입력 파일에 있는 NODATA 값을 0 으로 취급 |
| **Output layer** | `OUTPUT` | [raster]<br>기본값: `[Save to temporary file]` | Specification of the output raster containing the result. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output NoData value** | `NO_DATA` | [numeric: double]<br>기본값: -9999.0 | 산출 레이어에 있는 NODATA 에 사용할 값 |
| **Output data type** | `DATA_TYPE` | [enumeration]<br>기본값: 5 | 산출 래스터 데이터 유형입니다. 다음 가운데 하나를 선택할 수 있습니다:<br>• 0 —Byte (8 비트 부호 없는 정수 (quint8))<br>• 1 —Int16 (16 비트 부호 있는 정수 (qint16))<br>• 2 —UInt16 (16 비트 부호 없는 정수 (quint16))<br>• 3 —Int32 (32 비트 부호 있는 정수 (qint32))<br>• 4 —UInt32 (32 비트 부호 없는 정수 (quint32))<br>• 5 —Float32 (32 비트 부동소수점형 (float))<br>• 6 —Float64 (64 비트 부동소수점형 (double))<br>• 7 —CInt16 (Complex Int16)<br>• 8 —CInt32 (Complex Int32)<br>• 9 —CFloat32 (Complex Float32)<br>• 10 —CFloat64 (Complex Float64)<br>• 11 —Int8 (8 비트 부호 있는 정수 (qint8))<br>QGIS 와 함께 빌드된 GDAL 버전에 따라 사용할 수 있는 옵션이 달라집니다. (*Help* → *About* 메뉴 참조) |
| **Creation options** 선택적 | `CREATION_OPTIONS` (for QGIS <= 3.42, this was `CREATE_OPTIONS`) | [string]<br>기본값: '' | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL 드라이버 옵션* 참조)<br>배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (\|) 로 복수의 옵션들을 분리하십시오. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Extent** | `EXTENT` | [string] | 산출 래스터 레이어의 공간 영역 |
| **CRS authority identifier** | `CRS_AUTHID` | [crs] | 산출 래스터 레이어의 좌표계 |
| **Width in pixels** | `WIDTH_IN_PIXELS` | [numeric: integer] | 산출 래스터 레이어의 열 개수 |
| **Height in pixels** | `HEIGHT_IN_PIXEL` | [numeric: integer] | 산출 래스터 레이어의 행 개수 |
| **Total pixel count** | `TOTAL_PIXEL_COUNT` | [numeric: integer] | 산출 래스터 레이어의 픽셀 개수 |
| **NoData pixel count** | `NODATA_PIXEL_COUN` | [numeric: integer] | 산출 래스터 레이어에 있는 NODATA 픽셀 개수 |
| **True pixel count** | `TRUE_PIXEL_COUN` | [numeric: integer] | 산출 래스터 레이어에 있는 (값이 1 인) 참 픽셀의 개수 |
| **False pixel count** | `FALSE_PIXEL_COU` | [numeric: integer] | 산출 래스터 레이어에 있는 (값이 0 인) 거짓 픽셀의 개수 |
| **Output layer** | `OUTPUT` | [raster] | 결과물을 담고 있는 산출 래스터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:rasterbooleanor`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 래스터 계산기

래스터 레이어를 사용해 대수 (algebra) 작업을 수행합니다.

산출되는 레이어는 표현식에 따라 계산된 값을 가질 것입니다. 표현식은 숫자값, 연산자를 담을 수 있고, 현재 프로젝트에 있는 모든 레이어를 참조할 수 있습니다.

**더 보기**

래스터 계산기 *(가상)*, 래스터 계산기, 래스터 계산기

### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layers** | `INPUT` | [raster] [list] | 입력 래스터 레이어의 목록 |
| **Expression** | `EXPRESSION` | [expression] | 산출 래스터 레이어를 계산하는 데 사용할래스터 기반 표현식 입니다. |
| **Output extent**<br>선택적 | `EXTENT` | [extent] | 산출 래스터 레이어의 공간 범위를 지정합니다. 범위를 지정하지 않은 경우, 선택한 참조 레이어를 모두 커버하는 최소 범위를 사용할 것입니다.<br>다음과 같은 메소드를 사용할 수 있습니다:<br>• 레이어로부터 계산…: 현재 프로젝트에 불러온 레이어의 범위를 사용합니다.<br>• 레이어 맵으로부터 계산…: 활성 프로젝트의조판 맵 항목 범위를 사용합니다.<br>• 북마크로부터 계산…: 저장되어 있는북마크 의 범위를 사용합니다.<br>• 맵 캔버스 범위 사용<br>• 캔버스에 그리기: 연산에 넣을 범위를 정하는 직사각형을 클릭 & 드래그합니다.<br>• 좌표를 `xmin`, `xmax`, `ymin`, `ymax` 로 입력 |
| **Output cell size (leave empty to set automatically)**<br>선택적 | `CELLSIZE` | [numeric: double] | 산출 래스터 레이어의 셀 크기입니다. 셀 크기를 지정하지 않은 경우, 선택한 참조 레이어 (들) 의 최소 셀 크기를 사용할 것입니다. 셀 크기는 X 축 및 Y 축에 대해 동일할 것입니다. |
| **Output CRS**<br>선택적 | `CRS` | [crs] | 산출 래스터 레이어의 좌표계입니다. 좌표계를 지정하지 않은 경우, 첫 번째 참조 레이어의 좌표계를 사용할 것입니다. |
| **Calculated** | `OUTPUT` | [raster]<br>기본값: `[Save to temporary file]` | Specification of the output raster. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Creation options**<br>선택적 | CRE-<br>ATION_OPTIONS | [string]<br>기본값: ˚ | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL* 드라이버 옵션 참조)<br>배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (\|) 로 복수의 옵션들을 분리하십시오. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Calculated** | OUTPUT | [raster] | 계산된 값을 가진 산출 래스터 파일 |

### 파이썬 코드

**알고리즘 ID**: `native:rastercalc`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 래스터 계산기 (가상)

래스터 레이어를 사용해서 산술 작업을 수행하고 인메모리 (in-memory) 결과물을 생성합니다.

산출되는 레이어는 표현식에 따라 계산된 값을 가질 것입니다. 표현식은 숫자값, 연산자를 담을 수 있고, 현재 프로젝트에 있는 모든 레이어를 참조할 수 있습니다.

URI 로 정의되는 래스터 레이어의 레이어 픽셀을 실시간 (on-the-fly) 으로 계산해서 가상 래스터 레이어를 정의합니다. 디스크 상에 새 파일로 저장되는 것이 아니라, 이 가상 레이어는 계산에 사용된 래스터에 계속 연결되어 있기 때문에 계산에 사용된 래스터를 삭제하거나 이동시키면 가상 레이어에도 오류가 발생합니다. *Layer name* 을 지정할 수 있습니다. 지정하지 않으면 계산 표현식을 그대로 사용합니다. 프로젝트로부터 가상 레이어를 제거하면 삭제되며, *Export → Save as…*레이어 컨텍스트 메뉴를 이용하면 디스크 상에 파일로 저장할 수 있습니다.

**더 보기**

래스터 계산기, 래스터 계산기, 래스터 계산기

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layers** | INPUT | [raster] [list] | 입력 래스터 레이어의 목록 |
| **Expression** | EXPRESSION | [expression] | 산출 래스터 레이어를 계산하는 데 사용할래스터 기반 표현식 입니다. |
| **Output extent**<br>선택적 | EXTENT | [extent] | 산출 래스터 레이어의 공간 범위를 지정합니다. 범위를 지정하지 않은 경우, 선택한 참조 레이어를 모두 커버하는 최소 범위를 사용할 것입니다.<br>다음과 같은 메소드를 사용할 수 있습니다:<br>• 레이어로부터 계산…: 현재 프로젝트에 불러온 레이어의 범위를 사용합니다.<br>• 레이어 맵으로부터 계산…: 활성 프로젝트의조판 맵 항목 범위를 사용합니다.<br>• 북마크로부터 계산…: 저장되어 있는북마크 의 범위를 사용합니다.<br>• 맵 캔버스 범위 사용<br>• 캔버스에 그리기: 연산에 넣을 범위를 정하는 직사각형을 클릭 & 드래그합니다.<br>• 좌표를 xmin, xmax, ymin, ymax 로 입력 |
| **Output cell size (leave empty to set automatically)**<br>선택적 | CELLSIZE | [numeric: double] | 산출 래스터 레이어의 셀 크기입니다. 셀 크기를 지정하지 않은 경우, 선택한 참조 레이어 (들) 의 최소 셀 크기를 사용할 것입니다. 셀 크기는 X 축 및 Y 축에 대해 동일할 것입니다. |
| **Output CRS**<br>선택적 | CRS | [crs] | 산출 래스터 레이어의 좌표계입니다. 좌표계를 지정하지 않은 경우, 첫 번째 참조 레이어의 좌표계를 사용할 것입니다. |
| **Output layer name**<br>선택적 | LAYER_NAME | [string] | 생성된 레이어에 할당할 이름입니다. 설정하지 않는 경우, 계산 표현식의 텍스트를 사용합니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Calculated** | OUTPUT | [raster] | 계산된 값을 가진 산출 가상 래스터 파일입니다. |

### 파이썬 코드

**알고리즘 ID**: `native:virtualrastercalc`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 래스터 레이어 속성

지정한 래스터 레이어의 기본 속성을 반환합니다. 반환되는 속성에는 범위, (맵 단위) 픽셀 크기 및 픽셀 차원, 밴드 개수, 그리고 NODATA 값이 포함됩니다.

이 알고리즘은 모델에 있는 다른 알고리즘에 입력 값으로 사용하기 위해 이런 유용한 속성을 추출하는 수단으로 설계되었습니다. 예를 들면 GDAL 래스터 알고리즘에 기존 래스터의 픽셀 크기를 전송할 수 있게 말이죠.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [raster] | 입력 래스터 레이어 |
| **Band number**<br>선택적 | `BAND` | [raster band]<br>기본값: 설정하지 않음 | 특정 밴드의 속성도 반환할지 여부를 선택합니다. 밴드를 지정한 경우, 선택한 밴드의 NODATA 값도 반환합니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Number of bands in raster** | `BAND_COUNT` | [numeric: integer] | 래스터 밴드의 개수 |
| **CRS authority identifier** | `CRS_AUTHID` | [string] | 산출 래스터 레이어의 좌표계 |
| **Extent** | `EXTENT` | [string] | 래스터 레이어의 좌표계 단위 범위 |
| **Band has a NoData value set** | `HAS_NODATA_VALU` | [Boolean] | 선택한 밴드에 있는 NODATA 픽셀에 대한 값이 래스터 레이어에 설정되어 있는지 여부를 나타냅니다. |
| **Height in pixels** | `HEIGHT_IN_PIXEI` | [numeric: integer] | 래스터 레이어의 열 개수 |
| **Band NoData value** | `NODATA_VALUE` | [numeric: double] | (설정되어 있는 경우) 선택한 밴드에 있는 NODATA 픽셀의 값 |
| **Pixel size (height) in map units** | `PIXEL_HEIGHT` | [numeric: integer] | 픽셀의 맵 단위 수직 크기 |
| **Pixel size (width) in map units** | `PIXEL_WIDTH` | [numeric: integer] | 픽셀의 맵 단위 수평 크기 |
| **Width in pixels** | `WIDTH_IN_PIXELS` | [numeric: integer] | 래스터 레이어의 행 개수 |

다음 페이지에 계속

표 24.358 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Maximum x-coordinate** | X_MAX | [numeric: double] | |
| **Minimum x-coordinate** | X_MIN | [numeric: double] | |
| **Maximum y-coordinate** | Y_MAX | [numeric: double] | |
| **Minimum y-coordinate** | Y_MIN | [numeric: double] | |

### 파이썬 코드

**알고리즘 ID**: `native:rasterlayerproperties`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 래스터 레이어 통계

래스터 레이어의 지정한 밴드의 값으로부터 기본 통계를 계산합니다. *Processing → Results viewer* 메뉴로 산출물을 불러옵니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | INPUT | [raster] | 입력 래스터 레이어 |
| **Band number** | BAND | [raster band]<br>기본값: 입력 레이어의 첫 번째 밴드 | 래스터가 다중 밴드인 경우, 통계를 얻고자 하는 밴드를 선택하십시오. |
| **Statistics** | OUTPUT_HTML_FILE | [html]<br>기본값: [Save to temporary file] | 산출 레이어를 지정합니다:<br>• 산출물 건너뛰기<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Maximum value** | MAX | [numeric: double] | |
| **Mean value** | MEAN | [numeric: double] | |
| **Minimum value** | MIN | [numeric: double] | |
| **Statistics** | OUTPUT_HTML_FILE | [html] | 산출물 파일은 다음 정보를 담고 있습니다:<br>• 파일 분석: 래스터 레이어의 경로<br>• 최소값: 래스터의 최소값<br>• 최대값: 래스터의 최대값<br>• 범위: 최대값과 최소값의 차<br>• 합계: 값들의 총합<br>• 평균값: 값들의 평균<br>• 표준 편차: 값들의 표준 편차<br>• 제곱 합계: 전체 평균과 각 관측값의 차를 제곱한 값의 합계 |
| **Range** | RANGE | [numeric: double] | |
| **Standard deviation** | STD_DEV | [numeric: double] | |
| **Sum** | SUM | [numeric: double] | |
| **Sum of the squares** | SUM_OF_SQUARES | [numeric: double] | |

### 파이썬 코드

**알고리즘 ID**: `native:rasterlayerstatistics`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 래스터 레이어 유일값 보고

지정한 래스터 레이어 하나에 있는 각 유일값의 개수와 면적을 반환합니다. 면적 계산은 해당 레이어 좌표계의 면적 단위로 이루어집니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | INPUT | [raster] | 입력 래스터 레이어 |
| **Band number** | BAND | [raster band]<br>기본값: 입력 레이어의 첫 번째 밴드 | 래스터가 다중 밴드인 경우, 통계를 얻고자 하는 밴드를 선택하십시오. |

다음 페이지에 계속

표 24.361 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Unique values report** | OUTPUT_HTML_FILE | [file]<br>기본값: [Save to temporary file] | 산출 레이어를 지정합니다:<br>• 산출물 건너뛰기<br>• 임시 파일로 저장<br>• 파일로 저장… |
| **Unique values table** | OUTPUT_TABLE | [vector: table]<br>기본값: [Skip output] | 유일값을 위한 테이블을 지정합니다:<br>• 산출물 건너뛰기<br>• 임시 레이어 생성 (TEMPORARY_OUTPUT)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **CRS authority identifier** | CRS_AUTHID | [string] | 산출 래스터 레이어의 좌표계 |
| **Extent** | EXTENT | [string] | 산출 래스터 레이어의 공간 영역 |
| **Height in pixels** | HEIGHT_IN_PIXEL | [numeric: integer] | 산출 래스터 레이어의 행 개수 |
| **NoData pixel count** | NODATA_PIXEL_COUN | [numeric: integer] | 산출 래스터 레이어에 있는 NODATA 픽셀 개수 |
| **Total pixel count** | TOTAL_PIXEL_COUNT | [numeric: integer] | 산출 래스터 레이어의 픽셀 개수 |
| **Unique values report** | OUTPUT_HTML_FILE | [html] | 산출 HTML 파일은 다음 정보를 담고 있습니다:<br>• 파일 분석: 래스터 레이어의 경로<br>• 범위: 범위의 xmin, ymin, xmax, ymax 좌표<br>• 투영체: 레이어의 투영체<br>• 픽셀 단위 너비: 열의 개수와 픽셀 너비 크기<br>• 픽셀 단위 높이: 행의 개수와 픽셀 너비 크기<br>• 총 픽셀 개수: 모든 픽셀의 개수<br>• NODATA 픽셀 개수: NODATA 값을 가진 픽셀의 개수 |

다음 페이지에 계속

표 24.362 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Unique values table** | `OUTPUT_TABLE` | [vector: table] | 다음 열 3 개를 가진 테이블:<br>• *value*: 픽셀 값<br>• *count*: 해당 값을 가진 픽셀의 개수<br>• *m2* 또는 *deg2* 또는 *ft2* 또는…: 이 값을 가진 픽셀들의 총 면적입니다. 열 이름은 레이어 좌표계의 면적 단위에 따라 달라지며 면적 계산은 해당 단위로 이루어집니다. |
| **Width in pixels** | `WIDTH_IN_PIXELS` | [numeric: integer] | 산출 래스터 레이어의 열 개수 |

### 파이썬 코드

**알고리즘 ID**: `native:rasterlayeruniquevaluesreport`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 래스터 레이어 구역 통계

래스터 레이어의 값을 또다른 레이어에서 정의한 구역 (zone) 으로 범주화한 통계를 계산합니다.

**더 보기**

구역 통계

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input Layer** | `INPUT` | [raster] | 입력 래스터 레이어 |
| **Band number** | `BAND` | [raster band]<br>기본값: 래스터 레이어의 첫 번째 밴드 | 래스터가 다중 밴드인 경우, 통계를 계산하고자 하는 밴드를 선택하십시오. |
| **Zones layer** | `ZONES` | [raster] | 구역을 정의하는 래스터 레이어. 동일한 픽셀 값을 가진 인접한 픽셀들로 구역을 지정합니다. |

다음 페이지에 계속

표 24.363 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Zones band number** | ZONES_BAND | [raster band]<br>기본값: 래스터 레이어의 첫 번째 밴드 | 래스터가 다중 밴드인 경우, 구역을 정의하는 밴드를 선택하십시오. |
| **Statistics** | OUTPUT_TABLE | [vector: table]<br>기본값: [Create temporary layer] | Specification of the output report. *One of*:<br>• 임시 레이어 생성 (TEMPORARY_OUTPUT)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Reference layer**<br>선택적 | REF_LAYER | [enumeration]<br>기본값: 0 | 산출 레이어에 구역을 결정할 때 참조물로 사용될 중심 (centroid) 을 계산하기 위해 사용되는 래스터 레이어입니다. 다음 가운데 하나를 선택할 수 있습니다:<br>• 0 —입력 레이어: 소스 래스터 레이어에서 각 픽셀의 중심 위치의 구역 래스터 레이어 값을 샘플링해서 구역을 결정합니다.<br>• 1 —구역 레이어: 구역 래스터 레이어의 각 픽셀의 중심 위치에서 입력 래스터 레이어를 샘플링할 것입니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **CRS authority identifier** | CRS_AUTHID | [string] | 산출 래스터 레이어의 좌표계 |
| **Extent** | EXTENT | [string] | 산출 래스터 레이어의 공간 영역 |
| **Height in pixels** | HEIGHT_IN_PIXEI | [numeric: integer] | 산출 래스터 레이어의 행 개수 |
| **NoData pixel count** | NODATA_PIXEL_COUN | [numeric: integer] | 산출 래스터 레이어에 있는 NODATA 픽셀 개수 |

다음 페이지에 계속

표 24.365 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Statistics** | OUTPUT_TABLE | [vector: table] | 산출 레이어는 **각 구역별로** 다음 정보를 담고 있습니다:<br>• 면적: 구역의 제곱 래스터 단위 면적<br>• 합계: 구역에 있는 픽셀 값의 총합<br>• 개수: 구역에 있는 픽셀들의 개수<br>• 최소값: 구역에 있는 최소 픽셀 값<br>• 최대값: 구역에 있는 최대 픽셀 값<br>• 평균: 구역에 있는 픽셀 값들의 평균 |
| **Total pixel count** | TOTAL_PIXEL_COUNT | [numeric: integer] | 산출 래스터 레이어의 픽셀 개수 |
| **Width in pixels** | WIDTH_IN_PIXELS | [numeric: integer] | 산출 래스터 레이어의 열 개수 |

### 파이썬 코드

**알고리즘 ID**: `native:rasterlayerzonalstats`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## Raster minimum/maximum

Extracts the minimum and maximum pixel values from a specified band of a raster layer. If multiple pixels share the same minimum or maximum value, only one of them will be included in the output.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | INPUT | [raster] | Raster layer from which to extract the minimum and maximum pixel values. |
| **Band number** | BAND | [raster band]<br>기본값: 1 | The band of the raster to analyze. If the raster is multiband, specify the band number (starting from 1). |
| **Extract** | EXTREMA | [enumeration]<br>Default: 0 (Minimum and Maximum) | Choose which extrema to extract:<br>• 0: Minimum and Maximum<br>• 1: Minimum only<br>• 2: Maximum only |

다음 페이지에 계속

표 24.366 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output layer** | `OUTPUT` | [vector: point]<br>기본값: `[Create temporary layer]` | Specification of the output layer. *One of*:<br>• 산출물 건너뛰기<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output layer** | `OUTPUT` | [vector: point] | Vector layer with point features at the locations of the minimum and/or maximum pixel values. |

### 파이썬 코드

**Algorithm ID**: `native:rasterminmax`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## Raster rank

`NEW in 3.44`

Performs a cell-by-cell analysis in which output values match the rank of a sorted list of overlapping cell values from input layers. The output raster will be multi-band if multiple ranks are provided. If multiband rasters are used in the data raster stack, the algorithm will always perform the analysis on the first band of the rasters.

**더 보기**

값으로부터 셀 스택 백분율 순위 구하기, 셀 스택 백분위수 구하기, 래스터 레이어로부터 셀 스택 백분율 순위 구하기, 셀 통계, 래스터 계산기

### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT_RASTERS` | [raster] [list] | Input raster layer(s) to calculate the rank from. |
| **Rank** | `RANKS` | [numeric: integer] [list]<br>Default: [1] | List of ranks to calculate for each pixel in the sorted list of cells values from the input rasters. Cells whose values match the given rank(s) are kept in the output layer. The ranks are 1-based, meaning that the first rank is 1. Negative values are also supported. |
| **NoData value handling** | `NODATA_HANDLING` | [enumeration]<br>기본값: 0 | Choose how to handle NoData values in the input raster:<br>• 0 —Exclude NoData from values lists: will not include NoData values in the list of pixel values when calculating ranks.<br>• 1 —Presence of NoData in a values list results in NoData output cells: the output raster will have NoData values in the corresponding locations. |
| **Ranked** | `OUTPUT` | [raster]<br>기본값: `[Save to temporary file]` | Specification of the output raster containing the rank values. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

#### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output extent**<br>선택적 | `EXTENT` | [extent] | 산출 래스터 레이어의 공간 범위를 지정합니다. 범위를 지정하지 않은 경우, 선택한 참조 레이어를 모두 커버하는 최소 범위를 사용할 것입니다.<br>다음과 같은 메소드를 사용할 수 있습니다:<br>• 레이어로부터 계산…: 현재 프로젝트에 불러온 레이어의 범위를 사용합니다.<br>• 레이어 맵으로부터 계산…: 활성 프로젝트의조판 맵 항목 범위를 사용합니다.<br>• 북마크로부터 계산…: 저장되어 있는북마크 의 범위를 사용합니다.<br>• 맵 캔버스 범위 사용<br>• 캔버스에 그리기: 연산에 넣을 범위를 정하는 직사각형을 클릭 & 드래그합니다.<br>• 좌표를 `xmin`, `xmax`, `ymin`, `ymax` 로 입력 |
| **Output cell size**<br>선택적 | `CELL_SIZE` | [numeric: double] | 산출 래스터 레이어의 셀 크기입니다. 셀 크기를 지정하지 않은 경우, 선택한 참조 레이어 (들) 의 최소 셀 크기를 사용할 것입니다. 셀 크기는 X 축 및 Y 축에 대해 동일할 것입니다. |
| **Output CRS**<br>선택적 | `CRS` | [crs] | 산출 래스터 레이어의 좌표계입니다. 좌표계를 지정하지 않은 경우, 첫 번째 참조 레이어의 좌표계를 사용할 것입니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Ranked** | `OUTPUT` | [raster] | Output raster layer containing the result. |

### 파이썬 코드

**Algorithm ID**: `native:rasterrank`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 래스터 표면 부피

지정한 기준면 (base level) 에 따라 래스터 표면 아래의 부피를 계산합니다. 주로 수치 표고 모델 (DEM; Digital Elevation Model) 에 유용한 알고리즘입니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **INPUT layer** | `INPUT` | [raster] | 표면을 표현하는 입력 래스터 |
| **Band number** | `BAND` | [raster band]<br>기본값: 래스터 레이어의 첫 번째 밴드 | 래스터가 다중 밴드인 경우, 표면을 정의할 밴드를 선택하십시오. |
| **Base level** | `LEVEL` | [numeric: double]<br>기본값: 0.0 | 기준 또는 참조 값을 정의합니다. 부피를 `Method` 파라미터에 따라 (아래 참조) 계산하는 데 이 기준을 사용합니다. |
| **Method** | `METHOD` | [enumeration]<br>기본값: 0 | 래스터 픽셀 값과 `Base level` 의 차로 지정하는 부피 계산용 방법을 정의합니다. 다음 옵션 가운데 하나를 선택할 수 있습니다:<br>• 0 —기준면 위만 셉니다: 기준면 위에 있는 픽셀만 부피에 추가할 것입니다.<br>• 1 —기준면 아래만 셉니다: 기준면 아래 있는 픽셀만 부피에 추가할 것입니다.<br>• 2 —기준면 아래의 부피를 뺍니다: 기준면 위에 있는 픽셀을 부피에 추가하고, 기준면 아래 있는 픽셀을 부피에서 뺄 것입니다.<br>• 3 —기준면 아래의 부피를 추가합니다: 픽셀이 기준면 위에 있든 아래 있든 상관없이 부피에 추가합니다. 이 옵션은 픽셀 값과 기준면의 차의 절대값을 합한 것과 동일합니다. |
| **Surface volume report** | `OUTPUT_HTML_FILE` | [html]<br>기본값: `[Save to temporary file]` | Specification of the output HTML report. *One of*:<br>• 산출물 건너뛰기<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

다음 페이지에 계속

표 24.371 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Surface volume table** | OUTPUT_TABLE | [vector: table]<br>기본값: [Skip output] | Specification of the output table. *One of*:<br>• 산출물 건너뛰기<br>• 임시 레이어 생성 (TEMPORARY_OUTPUT)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Volume** | VOLUME | [numeric: double] | 계산된 부피 |
| **Area** | AREA | [numeric: double] | 제곱 맵 단위의 면적 |
| **Pixel_count** | PIXEL_COUNT | [numeric: integer] | 분석된 픽셀의 총 개수 |
| **Surface volume report** | OUTPUT_HTML_FILE | [html] | HTML 서식으로 된 (부피, 면적 및 픽셀 개수를 담고 있는) 산출 보고서 |
| **Surface volume table** | OUTPUT_TABLE | [vector: table] | (부피, 면적 및 픽셀 개수를 담고 있는) 산출 테이블 |

### 파이썬 코드

**알고리즘 ID**: native:rastersurfacevolume

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 래스터로 재범주화

래스터를 벡터 테이블에서 지정된 범위를 기반으로 하는 새 범주 값을 할당해서 재범주화합니다.

### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Raster layer** | `INPUT_RASTER` | [raster] | 재범주화할 래스터 레이어 |
| **Band number** | `RASTER_BAND` | [raster band]<br>기본값: 래스터 레이어의 첫 번째 밴드 | 래스터가 다중 밴드인 경우, 재범주화하려는 밴드를 선택하십시오. |
| **Layer containing class breaks** | `INPUT_TABLE` | [vector: any] | 범주화에 사용할 값을 담고 있는 벡터 레이어 |
| **Minimum class value field** | `MIN_FIELD` | [tablefield: numeric] | 범주 범위의 최소값을 가진 필드입니다. 가능한 가장 낮은 값을 획득하려면 `-inf` 를 사용하십시오. |
| **Maximum class value field** | `MAX_FIELD` | [tablefield: numeric] | 범주 범위의 최대값을 가진 필드입니다. 가능한 가장 높은 값을 획득하려면 `inf` 를 사용하십시오. |
| **Output value field** | `VALUE_FIELD` | [tablefield: numeric] | (대응하는 최소 및 최대값 사이의) 범주에 들어오는 픽셀에 할당될 값을 가진 필드입니다. 이 범위의 값을 NODATA 로 설정하려면 `nan` 을 사용하십시오. |
| **Reclassified raster** | `OUTPUT` | [raster]<br>기본값: `[Save to temporary file]` | Specification of the output raster. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

#### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output NoData value** | `NO_DATA` | [numeric: double]<br>기본값: -9999.0 | NODATA 값에 적용할 값입니다. |
| **Range boundaries** | `RANGE_BOUNDARIE` | [enumeration]<br>기본값: 0 | 범주화를 위한 비교 규칙을 정의합니다. 다음 옵션 가운데 선택할 수 있습니다:<br>• 0 — 최소값 < 값 <= 최대값<br>• 1 — 최소값 <= 값 < 최대값<br>• 2 — 최소값 <= 값 <= 최대값<br>• 3 — 최소값 < 값 < 최대값 |
| **Use NoData when no range matches value** | `NO-DATA_FOR_MISSIN` | [boolean]<br>기본값: False | 어떤 범주에도 들어오지 않는 밴드 값에 NODATA 값을 적용합니다. False 로 설정한 경우, 원본 값을 유지합니다. |

다음 페이지에 계속

표 24.374 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output data type** | DATA_TYPE | [enumeration] 기본값: 5 | 산출 래스터 파일의 포맷을 정의합니다. 다음 가운데 하나로 지정할 수 있습니다:<br>• 0 —Byte (8 비트 부호 없는 정수 (quint8))<br>• 1 —Int16 (16 비트 부호 있는 정수 (qint16))<br>• 2 —UInt16 (16 비트 부호 없는 정수 (quint16))<br>• 3 —Int32 (32 비트 부호 있는 정수 (qint32))<br>• 4 —UInt32 (32 비트 부호 없는 정수 (quint32))<br>• 5 —Float32 (32 비트 부동소수점형 (float))<br>• 6 —Float64 (64 비트 부동소수점형 (double))<br>• 7 —CInt16 (Complex Int16)<br>• 8 —CInt32 (Complex Int32)<br>• 9 —CFloat32 (Complex Float32)<br>• 10 —CFloat64 (Complex Float64)<br>• 11 —Int8 (8 비트 부호 있는 정수 (qint8))<br>QGIS 와 함께 빌드된 GDAL 버전에 따라 사용할 수 있는 옵션이 달라집니다. (*Help* ⍰ *About* 메뉴 참조) |
| **Creation options** 선택적 | CREATION_OPTIONS (for QGIS <= 3.42, this was CREATE_OPTIONS) | [string] 기본값: ' ' | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL* 드라이버 옵션 참조) 배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (\|) 로 복수의 옵션들을 분리하십시오. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Reclassified raster** | OUTPUT | [raster] | 재범주화된 밴드 값을 가진 산출 래스터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:reclassifybylayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 테이블로 재범주화

래스터를 고정 테이블에서 지정된 범위를 기반으로 하는 새 범주 값을 할당해서 재범주화합니다.

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Raster layer** | `INPUT_RASTER` | [raster] | 재범주화할 래스터 레이어 |
| **Band number** | `RASTER_BAND` | [raster band]<br>기본값: 1 | 값을 재계산하고자 하는 래스터 밴드 |
| **Reclassification table** | `TABLE` | [vector: table] | 각 범주의 경계 (`Minimum` 및 `Maximum`)를 설정할 값들과 범주에 들어오는 밴드 값들에 할당할 새로운 `Value` 로 채워질 3 열 테이블입니다. `-inf` 값을 최소값으로, `inf` 값을 최대값으로 사용할 수 있고, `nan` 을 사용하면 산출값을 NODATA 로 설정할 수 있습니다. |
| **Reclassified raster** | `OUTPUT` | [raster]<br>기본값: `[Save to temporary file]` | Specification of the output raster layer. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output NoData value** | `NO_DATA` | [numeric: double]<br>기본값: -9999.0 | NODATA 값에 적용할 값입니다. |

다음 페이지에 계속

표 24.377 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Range boundaries** | RANGE_BOUNDARIE | [enumeration] 기본값: 0 | 범주화를 위한 비교 규칙을 정의합니다. 다음 옵션 가운데 선택할 수 있습니다:<br>• 0 —최소값 < 값 <= 최대값<br>• 1 —최소값 <= 값 < 최대값<br>• 2 —최소값 <= 값 <= 최대값<br>• 3 —최소값 < 값 < 최대값 |
| **Use NoData when no range matches value** | NO-DATA_FOR_MISSIN | [boolean] 기본값: False | 어떤 범주에도 들어오지 않는 밴드 값에 NODATA 값을 적용합니다. False 로 설정한 경우, 원본 값을 유지합니다. |
| **Output data type** | DATA_TYPE | [enumeration] 기본값: 5 | 산출 래스터 파일의 포맷을 정의합니다. 다음 가운데 하나로 지정할 수 있습니다:<br>• 0 —Byte (8 비트 부호 없는 정수 (quint8))<br>• 1 —Int16 (16 비트 부호 있는 정수 (qint16))<br>• 2 —UInt16 (16 비트 부호 없는 정수 (quint16))<br>• 3 —Int32 (32 비트 부호 있는 정수 (qint32))<br>• 4 —UInt32 (32 비트 부호 없는 정수 (quint32))<br>• 5 —Float32 (32 비트 부동소수점형 (float))<br>• 6 —Float64 (64 비트 부동소수점형 (double))<br>• 7 —CInt16 (Complex Int16)<br>• 8 —CInt32 (Complex Int32)<br>• 9 —CFloat32 (Complex Float32)<br>• 10 —CFloat64 (Complex Float64)<br>• 11 —Int8 (8 비트 부호 있는 정수 (qint8))<br>QGIS 와 함께 빌드된 GDAL 버전에 따라 사용할 수 있는 옵션이 달라집니다. (*Help* → *About* 메뉴 참조) |
| **Creation options** 선택적 | CRE-ATION_OPTIONS (for QGIS <= 3.42, this was CRE-ATE_OPTIONS) | [string] 기본값: ‘’ | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL 드라이버 옵션* 참조)<br>배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (\|) 로 복수의 옵션들을 분리하십시오. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Reclassified raster** | `OUTPUT` | [raster] | 재범주화된 밴드 값을 가진 산출 래스터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:reclassifybytable`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 래스터 크기 조정

래스터 레이어의 히스토그램 (픽셀 값) 의 형태 (분포) 를 유지하면서, 래스터 레이어를 새 값의 범위로 크기 조정 (rescale) 합니다. 소스 래스터의 최소 및 최대 픽셀값에서 대상 최소 및 최대 픽셀 범위로 선형 보간을 사용해서 입력값을 매핑합니다.

이 알고리즘은 기본적으로 원본 NODATA 값을 유지하지만, 이를 무시할 수 있는 옵션이 있습니다.

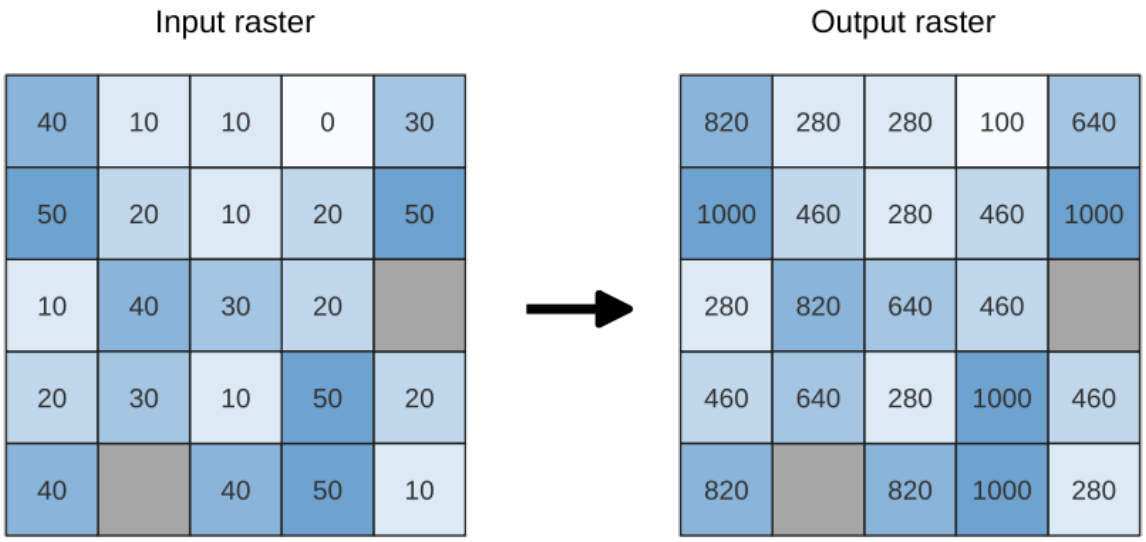



그림 24.57: 래스터 레이어의 값을 [0 - 50] 에서 [100 - 1000] 으로 크기 조정하기

### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input Raster** | `INPUT` | [raster] | 크기 조정에 사용할 래스터 레이어 |
| **Band number** | `BAND` | [raster band]<br>기본값: 입력 레이어의 첫 번째 밴드 | 래스터가 다중 밴드인 경우, 밴드를 선택하십시오. |
| **New minimum value** | `MINIMUM` | [numeric: double]<br>기본값: 0.0 | 크기 조정된 레이어에 사용할 최소 픽셀값 |
| **New maximum value** | `MAXIMUM` | [numeric: double]<br>기본값: 255.0 | 크기 조정된 레이어에 사용할 최대 픽셀값 |
| **New NoData value**<br>선택적 | `NODATA` | [numeric: double]<br>기본값: 설정하지 않음 | NODATA 픽셀에 할당할 값. 이 값을 설정하지 않으면, 원본 NODATA 값을 유지합니다. |
| **Rescaled** | `OUTPUT` | [raster]<br>기본값: `[Save to temporary file]` | Specification of the output raster layer. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

#### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Creation options**<br>선택적 | `CREATION_OPTIONS` (for QGIS <= 3.42, this was `CREATE_OPTIONS`) | [string]<br>기본값: '' | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL 드라이버 옵션* 참조) 배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (\|) 로 복수의 옵션들을 분리하십시오. |

#### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Rescaled** | `OUTPUT` | [raster] | 크기 조정된 밴드 값을 가진 산출 래스터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:rescaleraster`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 래스터 반올림

래스터 데이터셋의 셀 값을 지정한 소수점 이하의 자릿수에 따라 반올림합니다.

그 반대로, 음의 소수점 이하 자릿수를 이용해서 값을 밑 n 의 거듭제곱으로 반올림할 수도 있습니다. 예를 들어 밑 n 의 값이 10 이고 소수점 이하 자릿수가 -1 인 경우 이 알고리즘은 셀 값을 10 의 배수로, -2 인 경우 100 의 배수로, … 로 반올림합니다. 임의의 밑 값을 선택하더라도, 이 알고리즘은 동일한 곱셈 원칙을 적용합니다. 래스터 레이어를 일반화 (generalize) 하는 데 셀 값을 밑 n 의 배수로 반올림하는 방법을 쓸 수도 있습니다.

이 알고리즘은 입력 래스터의 데이터 유형을 유지합니다. 즉 바이트/정수형 래스터는 밑 n 의 배수로만 반올림될 수 있습니다. 그렇지 않을 경우 경고가 뜨면서 해당 래스터를 바이트/정수형 래스터로서 복사할 것입니다.

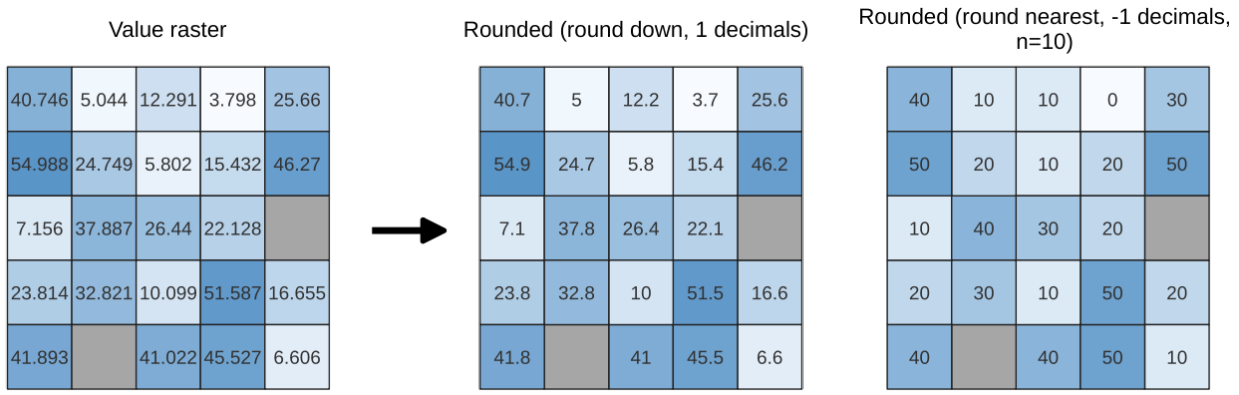



그림 24.58: 래스터의 값을 반올림하기

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input raster** | `INPUT` | [raster] | 공간 처리할 래스터 |
| **Band number** | `BAND` | [raster band]<br>기본값: 1 | 래스터의 밴드 |

다음 페이지에 계속

표 24.381 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Rounding direction** | ROUNDING_DIRECTION | [list]<br>기본값: 1 | 대상의 값을 반올림할 방법을 선택하는 방식입니다. 다음 옵션 가운데 하나를 선택할 수 있습니다:<br>• 0 — 올림 (Round up)<br>• 1 — 반올림 (Round to nearest)<br>• 2 — 내림 (Round down) |
| **Number of decimals places** | DECIMAL_PLACES | [numeric: integer]<br>기본값: 2 | 반올림할 소수점 이하 자릿수. 셀 값을 밑 n 의 배수로 반올림하려면 음의 값을 사용하십시오. |
| **Output raster** | OUTPUT | [raster]<br>기본값: [Save to temporary file] | Specification of the output file. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Base n for rounding to multiples of n** | BASE_N | [numeric: integer]<br>기본값: 10 | DECIMAL_PLACES 파라미터가 음수일 경우, 래스터 값을 밑 n 값의 배수로 반올림합니다. |
| **Creation options**<br>선택적 | CREATION_OPTIONS<br>(for QGIS <= 3.42, this was CREATE_OPTIONS) | [string]<br>기본값: '' | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL 드라이버 옵션* 참조)<br>배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (\|) 로 복수의 옵션들을 분리하십시오. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output raster** | OUTPUT | [raster] | 선택한 밴드의 값을 반올림한 산출 래스터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:roundrastervalues`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 래스터 값 샘플링

포인트 위치에 있는 래스터 값을 추출합니다. 래스터 레이어가 다중 밴드인 경우, 각 밴드를 샘플링합니다.

산출 레이어의 속성 테이블은 래스터 레이어의 밴드 개수만큼의 새 열을 가질 것입니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input Layer** | `INPUT` | [vector: point] | Point vector layer to use for sampling |
| **Raster Layer** | `RASTERCOPY` | [raster] | 지정한 포인트 위치에서 샘플링할 래스터 레이어 |
| **Output column prefix** | `COLUMN_PREFIX` | [string]<br>기본값: 'SAMPLE_' | 추가될 열의 이름 앞에 붙일 접두어 |
| **Sampled**<br>선택적 | `OUTPUT` | [vector: point]<br>기본값: `[Create temporary layer]` | Specify the output layer containing the sampled values. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Sampled** | `OUTPUT` | [vector: point] | 샘플링된 값을 담고 있는 산출 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:rastersampling`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 구역 히스토그램

폴리곤 피처 내부에 담겨진 래스터 레이어에서 나온 각 유일값의 개수를 표현하는 필드를 추가합니다.

산출 레이어의 속성 테이블은 폴리곤 (들) 과 교차하는 래스터 레이어의 유일값 개수만큼의 필드를 가질 것입니다.

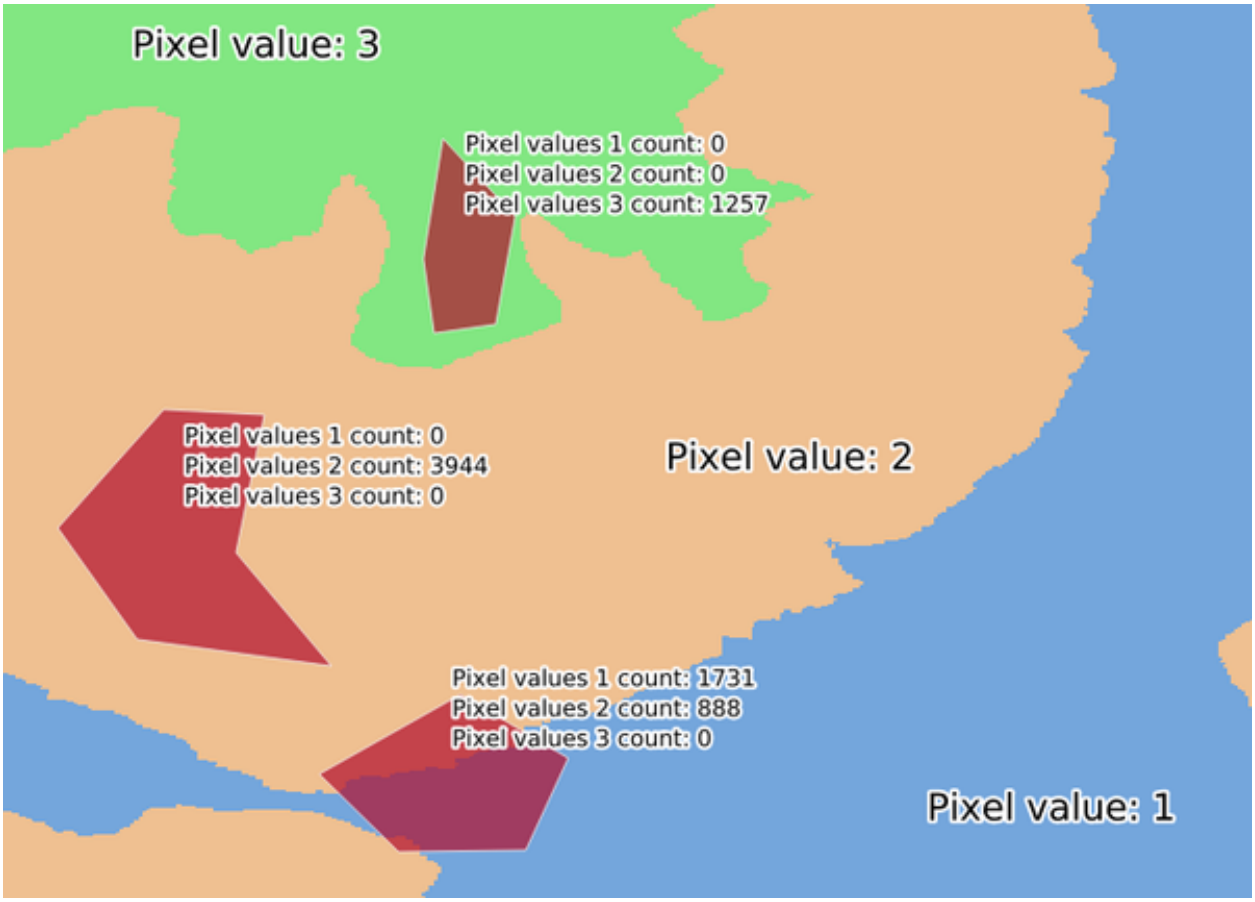



그림 24.59: 래스터 레이어 히스토그램 예시

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Raster layer** | `INPUT_RASTER` | [raster] | 입력 래스터 레이어 |
| **Band number** | `RASTER_BAND` | [raster band]<br>기본값: 입력 레이어의 첫 번째 밴드 | 래스터가 다중 밴드인 경우, 밴드를 선택하십시오. |
| **Vector layer containing zones** | `INPUT_VECTOR` | [vector: polygon] | 구역을 정의하는 벡터 폴리곤 레이어 |
| **Output column prefix** | `COLUMN_PREFIX`<br>선택적 | [string]<br>기본값: 'HISTO_' | 산출 열의 이름 앞에 붙일 접두어 |
| **Output zones** | `OUTPUT` | [vector: polygon]<br>기본값: `[Create temporary layer]` | Specify the output vector polygon layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output zones** | `OUTPUT` | [vector: polygon] | 산출 벡터 폴리곤 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:zonalhistogram`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### Zonal Minimum/Maximum Point

Extracts point features corresponding to the minimum and maximum pixel values within polygon zones.

The output will contain one point feature for the minimum and one for the maximum raster value for every individual zonal feature from a polygon layer. The created point layer will be in the same spatial reference system as the selected raster layer.

> **경고**
>
> This algorithm drops existing primary keys or FID values and regenerates them in output layers.

#### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input Layer** | `INPUT` | [vector: polygon] | Polygon layer defining the zones. |
| **Raster Layer** | `RASTER` | [raster] | Raster layer to extract the minimum and maximum values from. |
| **Raster Band** | `BAND` | [raster band]<br>기본값: 래스터 레이어의 첫 번째 밴드 | If the raster has multiple bands, select the band to process. |
| **Zonal extrema** | `OUTPUT` | [vector: point]<br>기본값: `[Create temporary layer]` | Specification of the output layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

#### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Zonal extrema** | `OUTPUT` | [vector: point] | Output layer containing the minimum and maximum points for each zone. |

### 파이썬 코드

**Algorithm ID**: `native:zonalminmaxpoint`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 구역 통계

중첩하는 폴리곤 벡터 레이어의 각 피처에 해당하는 래스터 레이어 영역의 통계를 계산합니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: polygon] | 구역을 담고 있는 벡터 폴리곤 레이어 |
| **Raster layer** | `INPUT_RASTER` | [raster] | 입력 래스터 레이어 |
| **Raster band** | `RASTER_BAND` | [raster band]<br>기본값: 입력 레이어의 첫 번째 밴드 | 래스터가 다중 밴드인 경우, 통계를 얻고자 하는 밴드를 선택하십시오. |
| **Output column prefix** | `COLUMN_PREFIX` | [string]<br>기본값: '_' | 산출 열의 이름 앞에 붙일 접두어 |
| **Statistics to calculate** | `STATISTICS` | [enumeration] [list]<br>기본값: [0,1,2] | 산출물을 위한 통계 연산자의 목록. 다음 옵션 가운데 선택할 수 있습니다:<br>• 0 —개수 (Count)<br>• 1 —합계 (Sum)<br>• 2 —평균 (Mean)<br>• 3 —중앙값 (Median)<br>• 4 —표준 편차 (St. dev.)<br>• 5 —최소값 (Minimum)<br>• 6 —최대값 (Maximum)<br>• 7 —범위 (Range)<br>• 8 —희귀값 (Minority)<br>• 9 —최빈값 (Majority)<br>• 10 —다양도 (Variety)<br>• 11 —변동 (Variance) |

다음 페이지에 계속

표 24.387 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Zonal Statistics** | `OUTPUT` | [vector: polygon] 기본값: `[Create temporary layer]` | Specify the output vector polygon layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

#### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Zonal Statistics** | `OUTPUT` | [vector: polygon] | 통계를 추가한 입력 구역 벡터 레이어 |

#### 파이썬 코드

**알고리즘 ID**: `native:zonalstatisticsfb`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 24.1.19 래스터 생성

#### 상수형 래스터 레이어 생성

지정한 범위 및 셀 크기를 지정한 값으로 채운 래스터 레이어를 생성합니다.

또한 산출 데이터 유형도 지정할 수 있습니다. 이 알고리즘은 선택한 산출 래스터 데이터 유형으로 표현할 수 없는 값이 입력된 경우 실행을 중단할 것입니다.

### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Desired extent** | `EXTENT` | [extent] | 산출 래스터 레이어의 공간 범위를 지정합니다. 내부적으로 타일 크기의 배수로 확장될 것입니다.<br>다음과 같은 메소드를 사용할 수 있습니다:<br>• 레이어로부터 계산…: 현재 프로젝트에 불러온 레이어의 범위를 사용합니다.<br>• 레이어 맵으로부터 계산…: 활성 프로젝트의조판 맵 항목 범위를 사용합니다.<br>• 북마크로부터 계산…: 저장되어 있는북마크 의 범위를 사용합니다.<br>• 맵 캔버스 범위 사용<br>• 캔버스에 그리기: 연산에 넣을 범위를 정하는 직사각형을 클릭 & 드래그합니다.<br>• 좌표를 `xmin, xmax, ymin, ymax` 로 입력 |
| **Target CRS** | `TARGET_CRS` | [crs]<br>기본값: Project CRS | 산출 래스터 레이어 용 좌표계 |
| **Pixel size** | `PIXEL_SIZE` | [numeric: double]<br>기본값: 1.0 | 맵 단위의 픽셀 크기 (X=Y) 입니다. |
| **Constant value** | `NUMBER` | [numeric: double]<br>기본값: 1.0 | 산출 래스터 레이어 용 상수 (constant) 픽셀 값입니다. |
| **Constant** | `OUTPUT` | [raster]<br>기본값: `[Save to temporary file]` | Specification of the output raster. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

#### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output raster data type** | OUTPUT_TYPE<br>기본값: 5 | [enumeration] | 산출 래스터 파일의 데이터 유형을 정의합니다. 다음 가운데 하나로 지정할 수 있습니다:<br>• 0 —바이트 (Byte)<br>• 1 —부호 있는 16 비트 정수형 (Integer16)<br>• 2 —부호 없는 16 비트 정수형 (Unsigned Integer16)<br>• 3 —부호 있는 32 비트 정수형 (Integer32)<br>• 4 —부호 없는 32 비트 정수형 (Unsigned Integer32)<br>• 5 —부동소수점 32 비트 실수형 (Float32)<br>• 6 —부동소수점 64 비트 실수형 (Float64) |
| **Creation options**<br>선택적 | CREATION_OPTIONS<br>(for QGIS <= 3.42, this was CREATE_OPTIONS) | [string]<br>기본값: '' | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL* 드라이버 옵션 참조)<br>배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (\|) 로 복수의 옵션들을 분리하십시오. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Constant** | OUTPUT | [raster] | 원하는 범위를 지정한 값으로 채운 지정한 픽셀 크기로 커버하는 래스터 |

### 파이썬 코드

**알고리즘 ID**: `native:createconstantrasterlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 랜덤 래스터 레이어 생성 (이항 분포)

지정한 범위 및 셀 크기를 이항식으로 (binomially) 분포된 랜덤한 값으로 채운 래스터 레이어를 생성합니다.

기본적으로, 지정한 10 배수 샘플 크기 (N of 10) 와 0.5 확률 (probability) 로 값을 선택할 것입니다. 샘플 크기 (N) 및 확률에 대한 고급 파라미터를 사용하면 이를 무시할 수 있습니다. 래스터 데이터 유형은 정수형으로 설정됩니다. (기본값은 부호가 있는 16 비트 정수형 (Integer16) 입니다.) 이항 분포 랜덤 값은 양의 정수로 정의됩니다. 부동소수점형 래스터는 정수를 부동소수점형으로 표현할 것입니다.

#### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Desired extent** | `EXTENT` | [extent] | 산출 래스터 레이어의 공간 범위를 지정합니다. 내부적으로 타일 크기의 배수로 확장될 것입니다.<br>다음과 같은 메소드를 사용할 수 있습니다:<br>• 레이어로부터 계산…: 현재 프로젝트에 불러온 레이어의 범위를 사용합니다.<br>• 레이어 맵으로부터 계산…: 활성 프로젝트의조판 맵 항목 범위를 사용합니다.<br>• 북마크로부터 계산…: 저장되어 있는북마크 의 범위를 사용합니다.<br>• 맵 캔버스 범위 사용<br>• 캔버스에 그리기: 연산에 넣을 범위를 정하는 직사각형을 클릭 & 드래그합니다.<br>• 좌표를 `xmin`, `xmax`, `ymin`, `ymax` 로 입력 |
| **Target CRS** | `TARGET_CRS` | [crs]<br>기본값: Project CRS | 산출 래스터 레이어 용 좌표계 |
| **Pixel size** | `PIXEL_SIZE` | [numeric: double]<br>기본값: 1.0 | 맵 단위의 픽셀 크기 (X=Y) 입니다. |
| **Output raster** | `OUTPUT` | [raster]<br>기본값: `[Save to temporary file]` | Specification of the output raster. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output raster data type** | `OUTPUT_TYPE`<br>기본값: 0 | [enumeration] | 산출 래스터 파일의 데이터 유형을 정의합니다. 다음 가운데 하나로 지정할 수 있습니다:<br>• 0 —부호 있는 16 비트 정수형 (Integer16)<br>• 1 —부호 없는 16 비트 정수형 (Unsigned Integer16)<br>• 2 —부호 있는 32 비트 정수형 (Integer32)<br>• 3 —부호 없는 32 비트 정수형 (Unsigned Integer32)<br>• 4 —부동소수점 32 비트 실수형 (Float32)<br>• 5 —부동소수점 64 비트 실수형 (Float64) |
| **N** | `N` | [numeric: integer]<br>기본값: 10 | |
| **Probability** | `PROBABILITY` | [numeric: double]<br>기본값: 0.5 | |
| **Creation options**<br>선택적 | `CREATION_OPTIONS`<br>(for QGIS <= 3.42, this was `CREATE_OPTIONS`) | [string]<br>기본값: '' | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL* 드라이버 옵션 참조)<br>배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (\|) 로 복수의 옵션들을 분리하십시오. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output raster** | `OUTPUT` | [raster] | 원하는 범위를 랜덤한 값으로 채운 픽셀 크기로 커버하는 래스터 |

### 파이썬 코드

**알고리즘 ID**: `native:createrandombinomialrasterlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 랜덤 래스터 레이어 생성 (지수 분포)

지정한 범위 및 셀 크기를 지수적으로 (exponentially) 분포된 랜덤한 값으로 채운 래스터 레이어를 생성합니다.

기본적으로, 지정한 1.0 λ(람다) 로 값을 선택합니다. λ 에 대한 고급 파라미터를 사용하면 이를 무시할 수 있습니다. 래스터 데이터 유형은 기본적으로 부동소수점 32 비트 실수형 (Float32) 으로 설정됩니다. 지수 분포 랜덤 값은 부동소수점형 숫자이기 때문입니다.

#### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Desired extent** | `EXTENT` | [extent] | 산출 래스터 레이어의 공간 범위를 지정합니다. 내부적으로 타일 크기의 배수로 확장될 것입니다.<br>다음과 같은 메소드를 사용할 수 있습니다:<br>• 레이어로부터 계산…: 현재 프로젝트에 불러온 레이어의 범위를 사용합니다.<br>• 레이어 맵으로부터 계산…: 활성 프로젝트의조판 맵 항목 범위를 사용합니다.<br>• 북마크로부터 계산…: 저장되어 있는북마크 의 범위를 사용합니다.<br>• 맵 캔버스 범위 사용<br>• 캔버스에 그리기: 연산에 넣을 범위를 정하는 직사각형을 클릭 & 드래그합니다.<br>• 좌표를 `xmin`, `xmax`, `ymin`, `ymax` 로 입력 |
| **Target CRS** | `TARGET_CRS` | [crs]<br>기본값: Project CRS | 산출 래스터 레이어 용 좌표계 |
| **Pixel size** | `PIXEL_SIZE` | [numeric: double]<br>기본값: 1.0 | 맵 단위의 픽셀 크기 (X=Y) 입니다. |
| **Output raster** | `OUTPUT` | [raster]<br>기본값: `[Save to temporary file]` | Specification of the output raster. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output raster data type** | `OUTPUT_TYPE`<br>기본값: 0 | [enumeration] | 산출 래스터 파일의 데이터 유형을 정의합니다. 다음 가운데 하나로 지정할 수 있습니다:<br>• 0 —부동소수점 32 비트 실수형 (Float32)<br>• 1 —부동소수점 64 비트 실수형 (Float64) |
| **Lambda** | `LAMBDA` | [numeric: double]<br>기본값: 1.0 | |
| **Creation options**<br>선택적 | `CREATION_OPTIONS`<br>(for QGIS <= 3.42, this was `CREATE_OPTIONS`) | [string]<br>기본값: ' ' | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL* 드라이버 옵션 참조)<br>배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (\|) 로 복수의 옵션들을 분리하십시오. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output raster** | `OUTPUT` | [raster] | 원하는 범위를 랜덤한 값으로 채운 픽셀 크기로 커버하는 래스터 |

### 파이썬 코드

**알고리즘 ID**: `native:createrandomexponentialrasterlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 랜덤 래스터 레이어 생성 (감마 분포)

지정한 범위 및 셀 크기를 감마 분포된 랜덤한 값으로 채운 래스터 레이어를 생성합니다.

기본적으로, 지정한 알파 및 베타 1.0 값으로 값을 선택합니다. 알파 및 베타에 대한 고급 파라미터를 사용하면 이를 무시할 수 있습니다. 래스터 데이터 유형은 기본적으로 부동소수점 32 비트 실수형 (Float32) 으로 설정됩니다. 감마 분포 랜덤 값은 부동소수점형 숫자이기 때문입니다.

#### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Desired extent** | `EXTENT` | [extent] | 산출 래스터 레이어의 공간 범위를 지정합니다. 내부적으로 타일 크기의 배수로 확장될 것입니다.<br>다음과 같은 메소드를 사용할 수 있습니다:<br>• 레이어로부터 계산…: 현재 프로젝트에 불러온 레이어의 범위를 사용합니다.<br>• 레이어 맵으로부터 계산…: 활성 프로젝트의조판 맵 항목 범위를 사용합니다.<br>• 북마크로부터 계산…: 저장되어 있는북마크 의 범위를 사용합니다.<br>• 맵 캔버스 범위 사용<br>• 캔버스에 그리기: 연산에 넣을 범위를 정하는 직사각형을 클릭 & 드래그합니다.<br>• 좌표를 `xmin`, `xmax`, `ymin`, `ymax` 로 입력 |
| **Target CRS** | `TARGET_CRS` | [crs]<br>기본값: Project CRS | 산출 래스터 레이어 용 좌표계 |
| **Pixel size** | `PIXEL_SIZE` | [numeric: double]<br>기본값: 1.0 | 맵 단위의 픽셀 크기 (X=Y) 입니다. |
| **Output raster** | `OUTPUT` | [raster]<br>기본값: `[Save to temporary file]` | Specification of the output raster. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output raster data type** | `OUTPUT_TYPE`<br>기본값: 0 | [enumeration] | 산출 래스터 파일의 데이터 유형을 정의합니다. 다음 가운데 하나로 지정할 수 있습니다:<br>• 0 —부동소수점 32 비트 실수형 (Float32)<br>• 1 —부동소수점 64 비트 실수형 (Float64) |
| **Alpha** | `ALPHA` | [numeric: double]<br>기본값: 1.0 | |
| **Beta** | `BETA` | [numeric: double]<br>기본값: 1.0 | |
| **Creation options**<br>선택적 | `CREATION_OPTIONS`<br>(for QGIS <= 3.42, this was `CREATE_OPTIONS`) | [string]<br>기본값: '' | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL* 드라이버 옵션 참조)<br>배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (\|) 로 복수의 옵션들을 분리하십시오. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output raster** | `OUTPUT` | [raster] | 원하는 범위를 랜덤하게 분포된 값으로 채운 픽셀 크기로 커버하는 래스터 |

### 파이썬 코드

**알고리즘 ID**: `native:createrandomgammarasterlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 랜덤 래스터 레이어 생성 (기하 분포)

지정한 범위 및 셀 크기를 기하학적으로 (geometrically) 분포된 랜덤한 값으로 채운 래스터 레이어를 생성합니다.

기본적으로, 지정한 0.5 확률 (probability) 로 값을 선택할 것입니다. 평균값에 대한 고급 파라미터를 사용하면 이를 무시할 수 있습니다. 래스터 데이터 유형은 정수형으로 설정됩니다. (기본값은 부호가 있는 16 비트 정수형 (Integer16) 입니다.) 기하 분포 랜덤 값은 양의 정수로 정의됩니다. 부동소수점형 래스터는 정수를 부동소수점형으로 표현할 것입니다.

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Desired extent** | `EXTENT` | [extent] | 산출 래스터 레이어의 공간 범위를 지정합니다. 내부적으로 타일 크기의 배수로 확장될 것입니다.<br>다음과 같은 메소드를 사용할 수 있습니다:<br>• 레이어로부터 계산…: 현재 프로젝트에 불러온 레이어의 범위를 사용합니다.<br>• 레이어 맵으로부터 계산…: 활성 프로젝트의조판 맵 항목 범위를 사용합니다.<br>• 북마크로부터 계산…: 저장되어 있는북마크 의 범위를 사용합니다.<br>• 맵 캔버스 범위 사용<br>• 캔버스에 그리기: 연산에 넣을 범위를 정하는 직사각형을 클릭 & 드래그합니다.<br>• 좌표를 `xmin`, `xmax`, `ymin`, `ymax` 로 입력 |
| **Target CRS** | `TARGET_CRS` | [crs]<br>기본값: Project CRS | 산출 래스터 레이어 용 좌표계 |
| **Pixel size** | `PIXEL_SIZE` | [numeric: double]<br>기본값: 1.0 | 맵 단위의 픽셀 크기 (X=Y) 입니다. |
| **Output raster** | `OUTPUT` | [raster]<br>기본값: `[Save to temporary file]` | Specification of the output raster. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output raster data type** | `OUTPUT_TYPE`<br>기본값: 0 | [enumeration] | 산출 래스터 파일의 데이터 유형을 정의합니다. 다음 가운데 하나로 지정할 수 있습니다:<br>• 0 —부호 있는 16 비트 정수형 (Integer16)<br>• 1 —부호 없는 16 비트 정수형 (Unsigned Integer16)<br>• 2 —부호 있는 32 비트 정수형 (Integer32)<br>• 3 —부호 없는 32 비트 정수형 (Unsigned Integer32)<br>• 4 —부동소수점 32 비트 실수형 (Float32)<br>• 5 —부동소수점 64 비트 실수형 (Float64) |
| **Probability** | `PROBABILITY` | [numeric: double]<br>기본값: 0.5 | |
| **Creation options**<br>선택적 | `CREATION_OPTIONS`<br>(for QGIS <= 3.42, this was `CREATE_OPTIONS`) | [string]<br>기본값: '' | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL* 드라이버 옵션 참조)<br>배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (\|) 로 복수의 옵션들을 분리하십시오. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output raster** | `OUTPUT` | [raster] | 원하는 범위를 랜덤하게 분포된 값으로 채운 픽셀 크기로 커버하는 래스터 |

### 파이썬 코드

**알고리즘 ID**: `native:createrandomgeometricrasterlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 랜덤 래스터 레이어 생성 (음이항 분포)

지정한 범위 및 셀 크기를 음이항식으로 (negative binomially) 분포된 랜덤한 값으로 채운 래스터 레이어를 생성합니다.

기본적으로, 지정한 10.0 배수 샘플 크기 (k of 10) 와 0.5 확률 (probability) 로 값을 선택할 것입니다. 샘플 크기 (k) 및 확률에 대한 고급 파라미터를 사용하면 이를 무시할 수 있습니다. 래스터 데이터 유형은 정수형으로 설정됩니다. (기본값은 부호가 있는 16 비트 정수형 (Integer16) 입니다.) 음의 이항 분포 랜덤 값은 양의 정수로 정의됩니다. 부동소수점형 래스터는 정수를 부동소수점형으로 표현할 것입니다.

#### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Desired extent** | `EXTENT` | [extent] | 산출 래스터 레이어의 공간 범위를 지정합니다. 내부적으로 타일 크기의 배수로 확장될 것입니다.<br>다음과 같은 메소드를 사용할 수 있습니다:<br>• 레이어로부터 계산…: 현재 프로젝트에 불러온 레이어의 범위를 사용합니다.<br>• 레이어 맵으로부터 계산…: 활성 프로젝트의조판 맵 항목 범위를 사용합니다.<br>• 북마크로부터 계산…: 저장되어 있는북마크 의 범위를 사용합니다.<br>• 맵 캔버스 범위 사용<br>• 캔버스에 그리기: 연산에 넣을 범위를 정하는 직사각형을 클릭 & 드래그합니다.<br>• 좌표를 `xmin`, `xmax`, `ymin`, `ymax` 로 입력 |
| **Target CRS** | `TARGET_CRS` | [crs]<br>기본값: Project CRS | 산출 래스터 레이어 용 좌표계 |
| **Pixel size** | `PIXEL_SIZE` | [numeric: double]<br>기본값: 1.0 | 맵 단위의 픽셀 크기 (X=Y) 입니다. |
| **Output raster** | `OUTPUT` | [raster]<br>기본값: `[Save to temporary file]` | Specification of the output raster. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output raster data type** | OUTPUT_TYPE<br>기본값: 0 | [enumeration] | 산출 래스터 파일의 데이터 유형을 정의합니다. 다음 가운데 하나로 지정할 수 있습니다:<br>• 0 —부호 있는 16 비트 정수형 (Integer16)<br>• 1 —부호 없는 16 비트 정수형 (Unsigned Integer16)<br>• 2 —부호 있는 32 비트 정수형 (Integer32)<br>• 3 —부호 없는 32 비트 정수형 (Unsigned Integer32)<br>• 4 —부동소수점 32 비트 실수형 (Float32)<br>• 5 —부동소수점 64 비트 실수형 (Float64) |
| **Distribution parameter k** | K_PARAMETER | [numeric: integer]<br>기본값: 10 | |
| **Probability** | PROBABILITY | [numeric: double]<br>기본값: 0.5 | |
| **Creation options**<br>선택적 | CREATION_OPTIONS<br>(for QGIS <= 3.42, this was CREATE_OPTIONS) | [string]<br>기본값: '' | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL* 드라이버 옵션 참조)<br>배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (\|) 로 복수의 옵션들을 분리하십시오. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output raster** | OUTPUT | [raster] | 원하는 범위를 랜덤하게 분포된 값으로 채운 픽셀 크기로 커버하는 래스터 |

### 파이썬 코드

**알고리즘 ID**: `native:createrandomnegativebinomialrasterlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 랜덤 래스터 레이어 생성 (정규 분포)

지정한 범위 및 셀 크기를 정규 (normal) 분포된 랜덤한 값으로 채운 래스터 레이어를 생성합니다.

기본적으로, 지정한 0.0 평균 및 1.0 표준 편차로 값을 선택합니다. 평균 및 표준 편차 값에 대한 고급 파라미터를 사용하면 이를 무시할 수 있습니다. 래스터 데이터 유형은 기본적으로 부동소수점 32 비트 실수형 (Float32) 으로 설정됩니다. 정규 분포 랜덤 값은 부동소수점형 숫자이기 때문입니다.

#### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Desired extent** | `EXTENT` | [extent] | 산출 래스터 레이어의 공간 범위를 지정합니다. 내부적으로 타일 크기의 배수로 확장될 것입니다.<br>다음과 같은 메소드를 사용할 수 있습니다:<br>• 레이어로부터 계산…: 현재 프로젝트에 불러온 레이어의 범위를 사용합니다.<br>• 레이어 맵으로부터 계산…: 활성 프로젝트의조판 맵 항목 범위를 사용합니다.<br>• 북마크로부터 계산…: 저장되어 있는북마크 의 범위를 사용합니다.<br>• 맵 캔버스 범위 사용<br>• 캔버스에 그리기: 연산에 넣을 범위를 정하는 직사각형을 클릭 & 드래그합니다.<br>• 좌표를 `xmin`, `xmax`, `ymin`, `ymax` 로 입력 |
| **Target CRS** | `TARGET_CRS` | [crs]<br>기본값: Project CRS | 산출 래스터 레이어 용 좌표계 |
| **Pixel size** | `PIXEL_SIZE` | [numeric: double]<br>기본값: 1.0 | 맵 단위의 픽셀 크기 (X=Y) 입니다. |
| **Output raster** | `OUTPUT` | [raster]<br>기본값: `[Save to temporary file]` | Specification of the output raster. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output raster data type** | `OUTPUT_TYPE`<br>기본값: 0 | [enumeration] | 산출 래스터 파일의 데이터 유형을 정의합니다. 다음 가운데 하나로 지정할 수 있습니다:<br>• 0 —부동소수점 32 비트 실수형 (Float32)<br>• 1 —부동소수점 64 비트 실수형 (Float64) |
| **Mean of normal distribution** | `MEAN` | [numeric: double]<br>기본값: 0.0 | |
| **Standard deviation of normal distribution** | `STDDEV` | [numeric: double]<br>기본값: 1.0 | |
| **Creation options**<br>선택적 | `CREATION_OPTIONS`<br>(for QGIS <= 3.42, this was `CREATE_OPTIONS`) | [string]<br>기본값: ' | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL* 드라이버 옵션 참조)<br>배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (\|) 로 복수의 옵션들을 분리하십시오. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output raster** | `OUTPUT` | [raster] | 원하는 범위를 랜덤하게 분포된 값으로 채운 픽셀 크기로 커버하는 래스터 |

### 파이썬 코드

**알고리즘 ID**: `native:createrandomnormalrasterlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 랜덤 래스터 레이어 생성 (푸아송 분포)

지정한 범위 및 셀 크기를 푸아송 (poisson) 분포된 랜덤한 값으로 채운 래스터 레이어를 생성합니다.

기본적으로, 지정한 1.0 평균으로 값을 선택할 것입니다. 평균값에 대한 고급 파라미터를 사용하면 이를 무시할 수 있습니다. 래스터 데이터 유형은 정수형으로 설정됩니다. (기본값은 부호가 있는 16 비트 정수형 (Integer16) 입니다.) 푸아송 분포 랜덤 값은 양의 정수로 정의됩니다. 부동소수점형 래스터는 정수를 부동소수점형으로 표현할 것입니다.

#### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Desired extent** | `EXTENT` | [extent] | 산출 래스터 레이어의 공간 범위를 지정합니다. 내부적으로 타일 크기의 배수로 확장될 것입니다.<br>다음과 같은 메소드를 사용할 수 있습니다:<br>• 레이어로부터 계산…: 현재 프로젝트에 불러온 레이어의 범위를 사용합니다.<br>• 레이어 맵으로부터 계산…: 활성 프로젝트의조판 맵 항목 범위를 사용합니다.<br>• 북마크로부터 계산…: 저장되어 있는북마크 의 범위를 사용합니다.<br>• 맵 캔버스 범위 사용<br>• 캔버스에 그리기: 연산에 넣을 범위를 정하는 직사각형을 클릭 & 드래그합니다.<br>• 좌표를 `xmin`, `xmax`, `ymin`, `ymax` 로 입력 |
| **Target CRS** | `TARGET_CRS` | [crs]<br>기본값: Project CRS | 산출 래스터 레이어 용 좌표계 |
| **Pixel size** | `PIXEL_SIZE` | [numeric: double]<br>기본값: 1.0 | 맵 단위의 픽셀 크기 (X=Y) 입니다. |
| **Output raster** | `OUTPUT` | [raster]<br>기본값: `[Save to temporary file]` | Specification of the output raster. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output raster data type** | `OUTPUT_TYPE`<br>기본값: 0 | [enumeration] | 산출 래스터 파일의 데이터 유형을 정의합니다. 다음 가운데 하나로 지정할 수 있습니다:<br>• 0 —부호 있는 16 비트 정수형 (Integer16)<br>• 1 —부호 없는 16 비트 정수형 (Unsigned Integer16)<br>• 2 —부호 있는 32 비트 정수형 (Integer32)<br>• 3 —부호 없는 32 비트 정수형 (Unsigned Integer32)<br>• 4 —부동소수점 32 비트 실수형 (Float32)<br>• 5 —부동소수점 64 비트 실수형 (Float64) |
| **Mean** | `MEAN` | [numeric: double]<br>기본값: 1.0 | |
| **Creation options**<br>선택적 | `CREATION_OPTIONS`<br>(for QGIS <= 3.42, this was `CREATE_OPTIONS`) | [string]<br>기본값: '' | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL* 드라이버 옵션 참조)<br>배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (\|) 로 복수의 옵션들을 분리하십시오. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output raster** | `OUTPUT` | [raster] | 원하는 범위를 랜덤하게 분포된 값으로 채운 픽셀 크기로 커버하는 래스터 |

### 파이썬 코드

**알고리즘 ID**: `native:createrandompoissonrasterlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 랜덤 래스터 레이어 생성 (균등 분포)

지정한 범위 및 셀 크기를 균등하게 (uniformly) 분포된 랜덤한 값으로 채운 래스터 레이어를 생성합니다.

기본적으로, 값들은 지정한 산출 래스터 유형의 최소값과 최대값 사이의 범위일 것입니다. 하한 및 상한 경계 값에 대한 고급 파라미터를 사용하면 이를 무시할 수 있습니다. 이 경계들이 동일한 값을 가지거나 둘 다 (기본값) 0 인 경우, 이 알고리즘은 선택한 래스터 데이터 유형의 값 범위 전체에서 랜덤한 값을 생성할 것입니다. 산출 래스터 유형이 받아들일 수 있는 범위를 벗어난 경계들을 선택하면, 알고리즘이 중단될 것입니다.

#### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Desired extent** | `EXTENT` | [extent] | 산출 래스터 레이어의 공간 범위를 지정합니다. 내부적으로 타일 크기의 배수로 확장될 것입니다.<br>다음과 같은 메소드를 사용할 수 있습니다:<br>• 레이어로부터 계산…: 현재 프로젝트에 불러온 레이어의 범위를 사용합니다.<br>• 레이어 맵으로부터 계산…: 활성 프로젝트의조판 맵 항목 범위를 사용합니다.<br>• 북마크로부터 계산…: 저장되어 있는북마크 의 범위를 사용합니다.<br>• 맵 캔버스 범위 사용<br>• 캔버스에 그리기: 연산에 넣을 범위를 정하는 직사각형을 클릭 & 드래그합니다.<br>• 좌표를 `xmin`, `xmax`, `ymin`, `ymax` 로 입력 |
| **Target CRS** | `TARGET_CRS` | [crs]<br>기본값: Project CRS | 산출 래스터 레이어 용 좌표계 |
| **Pixel size** | `PIXEL_SIZE` | [numeric: double]<br>기본값: 1.0 | 맵 단위의 픽셀 크기 (X=Y) 입니다. |
| **Output raster** | `OUTPUT` | [raster]<br>기본값: `[Save to temporary file]` | Specification of the output raster. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output raster data type** | `OUTPUT_TYPE`<br>기본값: 5 | [enumeration] | 산출 래스터 파일의 데이터 유형을 정의합니다. 다음 가운데 하나로 지정할 수 있습니다:<br>• 0 —바이트 (Byte)<br>• 1 —부호 있는 16 비트 정수형 (Integer16)<br>• 2 —부호 없는 16 비트 정수형 (Unsigned Integer16)<br>• 3 —부호 있는 32 비트 정수형 (Integer32)<br>• 4 —부호 없는 32 비트 정수형 (Unsigned Integer32)<br>• 5 —부동소수점 32 비트 실수형 (Float32)<br>• 6 —부동소수점 64 비트 실수형 (Float64) |
| **Lower bound for random number range** | `LOWER_BOUND` | [numeric: double]<br>기본값: 0.0 | |
| **Upper bound for random number range** | `UPPER_BOUND` | [numeric: double]<br>기본값: 0.0 | |
| **Creation options**<br>선택적 | `CREATION_OPTIONS`<br>(for QGIS <= 3.42, this was `CREATE_OPTIONS`) | [string]<br>기본값: ' ' | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL* 드라이버 옵션 참조)<br>배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (\|) 로 복수의 옵션들을 분리하십시오. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output raster** | `OUTPUT` | [raster] | 원하는 범위를 랜덤하게 분포된 값으로 채운 픽셀 크기로 커버하는 래스터 |

파이썬 코드

**알고리즘 ID**: `native:createrandomuniformrasterlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 24.1.20 래스터 지형 분석

향

입력 수치 지형 모델 (DTM; Digital Terrain Model) 의 향 (aspect) 을 계산합니다. 산출되는 향 래스터 레이어는 경사 방향을 표현하는 0 에서 360 까지의 값을 담고 있습니다. 이 값은 북 (0°) 에서 시작해 시계방향으로 늘어납니다.

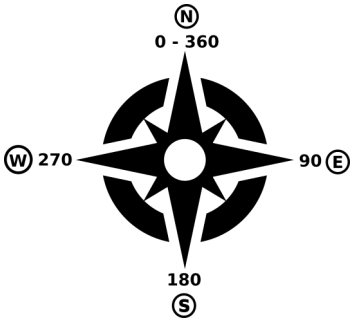



그림 24.60: 향 값

다음 그림은 색상표로 재범주화된 향 레이어입니다:

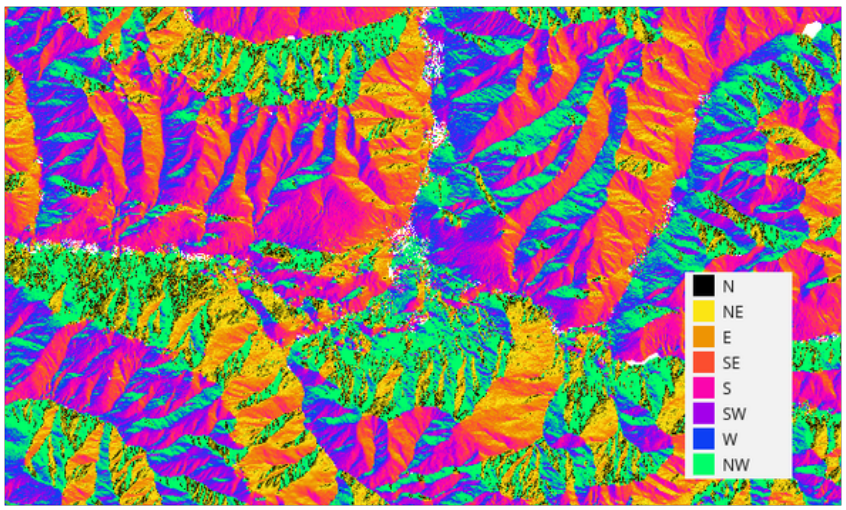



그림 24.61: 재범주화된 향 레이어

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Elevation layer** | `INPUT` | [raster] | DTM 래스터 레이어 |
| **Z factor** | `Z_FACTOR` | [numeric: double]<br>기본값: 1.0 | 수직 과장 (exaggeration). 이 파라미터는 Z 단위가 X 및 Y 단위와, 예를 들어 피트와 미터처럼 다를 때 유용합니다. 이 파라미터를 사용해서 차이를 조정할 수 있습니다. 기본값은 1(과장 없음) 입니다. |
| **Aspect** | `OUTPUT` | [raster]<br>기본값: `[Save to temporary file]` | Specify the output aspect raster layer. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Aspect** | `OUTPUT` | [raster] | 산출 향 래스터 레이어 |

### 파이썬 코드

**Algorithm ID**: `native:aspect`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## DTM 필터 (경사 기반)

DEM 의 셀들을 지표 셀과 (지표가 아닌) 객체 셀로 분류하기 위해 DEM 을 필터링하는 데 사용할 수 있습니다.

이 도구는 포셀만 (G. Vosselman, 2000) 이 설명한 개념을 사용해서, 가까이 있는 2 개의 셀 사이의 높이 차이가 큰 경우 이는 지형의 급경사 때문이 아닐 가능성이 크다는 가정에 바탕을 두고 있습니다. 더 높은 셀이 지표가 아닐 수도 있는 확률은 두 셀 사이의 거리가 줄어들수록 증가합니다. 따라서 이 필터는 두 셀 사이의 최대 높이 차이 (`dz_max`) 를 셀들 사이의 거리 (`d`) 의 함수 (`dz_max( d ) = d`) 로 정의합니다. 어떤 셀의 커널 반경 안에, 두 셀 사이의 거리에서 허용된 최대 높이 차이를 초과하는 셀이 하나도 없는 경우 해당 셀을 지형으로 분류합니다.

지형 경사 근사치 (`s`) 파라미터를 사용해서 연구 지역의 전체적인 경사와 일치하도록 필터 함수를 수정 (`dz_max( d ) = d * s`) 합니다. 5% 신뢰 구간 (`ci = 1.65 * sqrt( 2 * stddev )`) 을 이용해서 필터 기준을 완화 (`dz_max( d ) = d * s + ci`) 시키거나 증폭 (`dz_max( d ) = d * s - ci`) 시켜 이 필터 함수를 더욱 수정할 수도 있습니다.

참고 문헌*: Vosselman, G. (2000): Slope based filtering of laser altimetry data. IAPRS, Vol. XXXIII, Part B3, Amsterdam, The Netherlands, 935-942*

더 보기

This tool is a port of the SAGA DTM Filter (slope-based)

### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [raster] | DTM 래스터 레이어 |
| **Band number** | `BAND` | [raster band] | 연산에 넣을 DEM 의 밴드 |
| **Kernel radius (pixels)** | `RADIUS` | [numeric: integer]<br>기본값: 5 | 필터 커널의 (픽셀 단위) 반경입니다. 지표가 아닌 객체 옆에 있는 지표 셀까지 도달할 정도로 충분히 큰 값이어야만 합니다. |
| **Terrain slope (%, pixel size/vertical units)** | `TERRAIN_SLOPE` | [numeric: double]<br>Default: 30.0 | % 단위 지형 경사 근사치입니다. 지형 경사는'높이 단위: 래스터 픽셀 차원' 비율을 연산에 넣도록 조정해야만 합니다. 경사가 좀 더 급한 지형에서 필터 기준을 완화시키는 데 쓰입니다. |
| **Filter modification** | `FIL-TER_MODIFICATIC` | [list]<br>기본값: 0 | 필터 커널을 수정하지 않고 적용시킬지, 또는 신뢰 구간을 이용해서 높이 기준을 완화 또는 증폭시킬지 여부를 선택합니다.<br>• 0 - 없음<br>• 1 - 필터 완화<br>• 2 - 필터 증폭 |
| **Standard deviation** | `STAN-DARD_DEVIATION` | [numeric: double]<br>기본값: 0.1 | 높이 한계값에 적용되는 5% 신뢰 구간을 계산하는 데 쓰이는 표준 편차입니다. |
| **Output layer (ground)**<br>선택적 | `OUTPUT_GROUND` | [raster]<br>기본값: `[Save to temporary file]` | Specify the filtered DEM containing only cells classified as ground. *One of*:<br>• 산출물 건너뛰기<br>• 임시 파일로 저장<br>• 파일로 저장… |
| **Output layer (non-ground objects)**<br>선택적 | `OUT-PUT_NONGROUND` | [raster]<br>기본값: `[Skip output]` | Specify the non-ground objects removed by the filter. *One of*:<br>• 산출물 건너뛰기<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Creation options**<br>선택적 | `CREATION_OPTIONS`<br>(for QGIS <= 3.42, this was `CREATE_OPTIONS`) | [string]<br>기본값: ' ' | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL* 드라이버 옵션 참조)<br>배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (\|) 로 복수의 옵션들을 분리하십시오. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output layer (ground)** | `OUTPUT_GROUND` | [raster] | 지표로 분류된 셀들만 담고 있는 필터링된 DEM 입니다. |
| **Output layer (non-ground objects)** | `OUTPUT_NONGROUND` | [raster] | 필터가 제거한 지표가 아닌 객체들입니다. |

### 파이썬 코드

**알고리즘 ID**: `native:dtmslopebasedfilter`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## Fill sinks (Wang & Liu)

`NEW in 3.44`

Uses a method proposed by Wang & Liu to identify and fill surface depressions in digital elevation models.

The method was enhanced to allow the creation of hydrologically sound elevation models, i.e. not only to fill the depression(s) but also to preserve a downward slope along the flow path. If desired, this is accomplished by preserving a minimum slope gradient (and thus elevation difference) between cells.

*References: Wang, L. & H. Liu (2006): An efficient method for identifying and filling surface depressions in digital elevation models for hydrologic analysis and modelling. International Journal of Geographical Information Science, Vol. 20, No. 2: 193-213.*

**더 보기**

This tool is a port of the SAGA Fill Sinks (Wang & Liu) tool.

### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | INPUT | [raster] | DTM 래스터 레이어 |
| **Band number** | BAND | [raster band]<br>Default: 1 | 연산에 넣을 DEM 의 밴드 |
| **Minimum slope (degrees)** | MIN_SLOPE | [numeric: double]<br>Default: 0.10 | Minimum slope gradient to preserve from cell to cell; with a value of zero, sinks are filled up to the spill elevation (which results in flat areas). |
| **Output layer (filled DEM)**<br>선택적 | OUTPUT_FILLED_DEM | [raster]<br>기본값: [Save to temporary file] | Specify the ouput raster corresponding to the depression-free DEM. *One of*:<br>• 산출물 건너뛰기<br>• 임시 파일로 저장<br>• 파일로 저장… |
| **Output layer (flow directions)**<br>선택적 | OUTPUT_FLOW_DIRECT | [raster]<br>기본값: [Skip output] | Specify the output raster with computed flow directions. *One of*:<br>• 산출물 건너뛰기<br>• 임시 파일로 저장<br>• 파일로 저장… |
| **Output layer (watershed basins)**<br>선택적 | OUTPUT_WATERSHED_B | [raster]<br>기본값: [Skip output] | Specify the output raster corresponding to the delineated watershed basins. *One of*:<br>• 산출물 건너뛰기<br>• 임시 파일로 저장<br>• 파일로 저장… |

#### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Creation options**<br>선택적 | CREATION_OPTIONS | [string]<br>기본값: '' | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL* 드라이버 옵션 참조)<br>배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (\|) 로 복수의 옵션들을 분리하십시오. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output layer (filled DEM)** | `OUTPUT_FILLED_DEM` | [raster] | Ouput raster corresponding to the depression-free digital elevation model. |
| **Output layer (flow directions)** | `OUTPUT_FLOW_DIRECT` | [raster] | Output raster with computed flow directions; 0=N, 1=NE, 2=E, ⋯7=NW. |
| **Output layer (watershed basins)** | `OUTPUT_WATERSHED_B` | [raster] | Output raster corresponding to the delineated watershed basins. |

### 파이썬 코드

**Algorithm ID**: `native:fillsinkswangliu`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 음영기복

입력 DTM 으로부터 음영기복 (hillshade) 래스터 레이어를 계산합니다.

태양의 위치에 따라 레이어의 음영을 계산합니다. 태양의 수평 각도 (방위각, azimuth) 와 수직 각도 (태양 고도) 둘 다 변경할 수 있는 옵션이 있습니다.

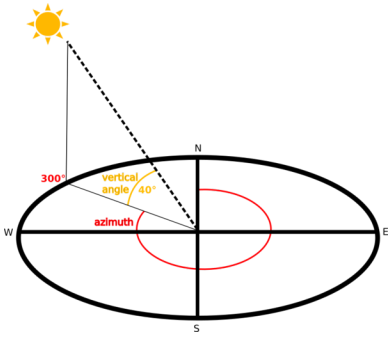



그림 24.62: 방위각 및 수직 각도

음영기복 레이어는 0(완전한 그림자) 에서 255(완전한 태양) 까지의 값을 담고 있습니다. 음영기복은 보통 해당 지역의 기복 (relief) 을 더 잘 이해하기 위해 사용됩니다.

음영기복 레이어에 투명도를 적용한 다음 표고 래스터와 중첩시키면 아주 흥미로운 맵을 볼 수 있습니다:

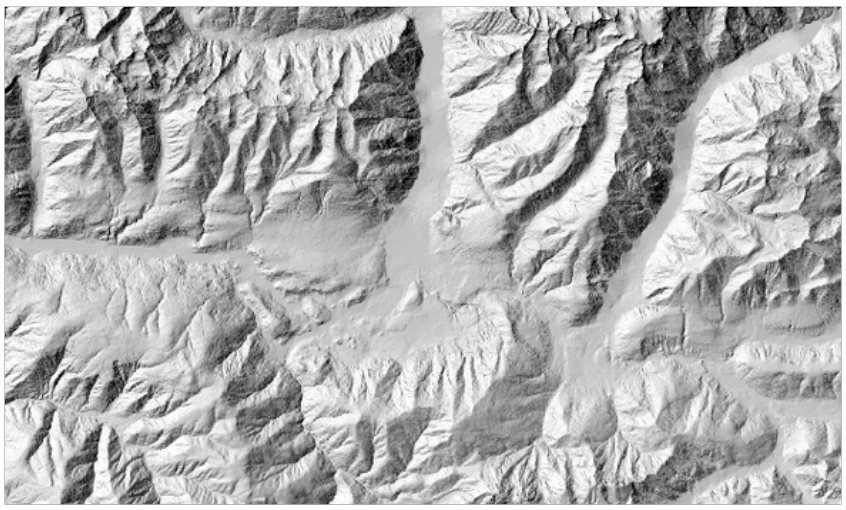


그림 24.63: 방위각이 300 이고 수직 각도가 45 인 음영기복 레이어

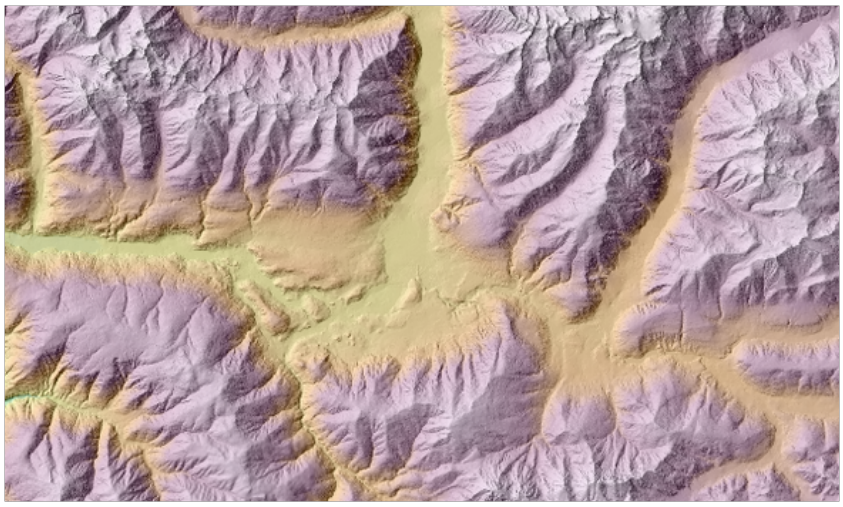


그림 24.64: 음영기복과 표고 레이어의 중첩

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Elevation layer** | `INPUT` | [raster] | DTM 래스터 레이어 |
| **Z factor** | `Z_FACTOR` | [numeric: double]<br>기본값: 1.0 | 수직 과장 (exaggeration). 이 파라미터는 Z 단위가 X 및 Y 단위와, 예를 들어 피트와 미터처럼 다를 때 유용합니다. 이 파라미터를 사용해서 차이를 조정할 수 있습니다. 이 파라미터의 값을 증가시키면 마지막 결과물을 과장할 것입니다. (경사가 더 심하게 보이게 만듭니다.) 기본값은 1(과장 없음) 입니다. |
| **Azimuth (horizontal angle)** | `AZIMUTH` | [numeric: double]<br>기본값: 300.0 | 태양의 수평 각도를 (도 단위 시계 방향으로) 설정합니다. 0 에서 360 까지의 범위로, 0 이 북쪽입니다. |
| **Vertical angle** | `V_ANGLE` | [numeric: double]<br>기본값: 40.0 | 태양의 수직 각도를 (도 단위로) 설정합니다. 태양의 고도를 말합니다. 0(최소 고도) 에서 90(최고 고도) 사이에서 값을 설정할 수 있습니다. |
| **Hillshade** | `OUTPUT` | [raster]<br>기본값: `Save to temporary file` | Specify the output hillshade raster layer. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Hillshade** | `OUTPUT` | [raster] | 산출 음영기복 래스터 레이어 |

### 파이썬 코드

**Algorithm ID**: `native:hillshade`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 고도분포곡선

입력 DEM 으로부터 고도분포곡선 (hypsometric curve) 을 계산합니다. 사용자가 지정한 산출 폴더에 계산한 곡선을 CSV 파일로 생성합니다.

고도분포곡선이란 지리적 영역의 표고값의 누적 히스토그램을 말합니다.

고도분폭고선을 지역의 지형학에 의한 풍경의 변화를 감지하는 데 사용할 수 있습니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **DEM to analyze** | `INPUT_DEM` | [raster] | 고도를 계산하는 데 사용하기 위한 DEM 래스터 레이어 |
| **Boundary layer** | `BOUNDARY_LAYER` | [vector: polygon] | 고도분포곡선을 계산하는 데 사용되는 영역의 경계를 담은 폴리곤 벡터 레이어 |
| **Step** | `STEP` | [numeric: double]<br>기본값: 100.0 | 곡선들 사이의 수직 거리 |
| **Use % of area instead of absolute value** | `USE_PERCENTAGE` | [boolean]<br>기본값: False | CSV 파일의 `Area` 필드에 절대 면적 대신 면적의 백분율을 작성합니다. |
| **Hypsometric curves** | `OUTPUT_DIRECTORY` | [folder] | Specify the output folder for the hypsometric curves. *One of*:<br>• 임시 디렉터리에 저장<br>• 디렉터리에 저장 |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Hypsometric curves** | `OUTPUT_DIRECTORY` | [folder] | 고도분포곡선을 담은 파일을 담고 있는 디렉터리입니다. 입력 벡터 레이어의 각 피처별로, 면적 및 고도 값을 가진 CSV 파일을 생성할 것입니다.<br>파일명은 `histogram_` 으로 시작해서, 그 뒤에 레이어명과 피처 ID 를 붙입니다. |

### 파이썬 코드

**알고리즘 ID**: `qgis:hypsometriccurves`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

| | A | B |
|---|---|---|
| 1 | Area | Elevation |
| 2 | 177475194.383 | 307 |
| 3 | 233206029.24 | 407 |
| 4 | 295553735.793 | 507 |
| 5 | 394718815.615 | 607 |
| 6 | 501801102.615 | 707 |
| 7 | 624399019.792 | 807 |
| 8 | 828877274.39 | 907 |
| 9 | 1042693465.68 | 1007 |
| 10 | 1277373021.81 | 1107 |
| 11 | 1556443975.41 | 1207 |
| 12 | 1888617494.27 | 1307 |
| 13 | 2248520437.31 | 1407 |
| 14 | 2627916813.17 | 1507 |
| 15 | 3010880212.04 | 1607 |
| 16 | 3411087555.34 | 1707 |

### 기복

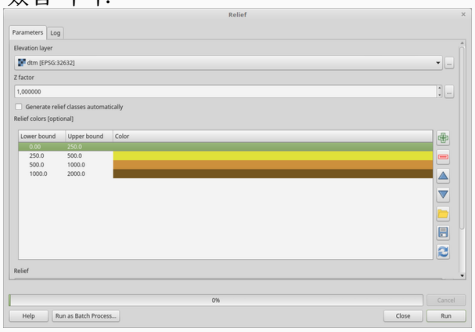
수치 표고 데이터로부터 음영 기복 (relief) 레이어를 생성합니다. 기복 색상을 직접 지정할 수도 있고, 알고리즘이 모든 기복 색상 범주를 자동으로 선택하도록 할 수도 있습니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Elevation layer** | `INPUT` | [raster] | DTM 래스터 레이어 |
| **Z factor** | `Z_FACTOR` | [numeric: double]<br>기본값: 1.0 | 수직 과장 (exaggeration). 이 파라미터는 Z 단위가 X 및 Y 단위와, 예를 들어 피트와 미터처럼 다를 때 유용합니다. 이 파라미터를 사용해서 차이를 조정할 수 있습니다. 이 파라미터의 값을 증가시키면 마지막 결과물을 과장할 것입니다. (경사가 더 심하게 보이게 만듭니다.) 기본값은 1(과장 없음) 입니다. |
| **Generate relief classes automatically** | `AUTO_COLORS` | [boolean]<br>기본값: False | 이 옵션을 활성화하면 알고리즘이 기복 색상 범주를 자동으로 생성할 것입니다. |

다음 페이지에 계속

표 24.428 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Relief colors** 선택적 | COLORS | [table widget] | 기복 색상을 직접 선택하고 싶은 경우 테이블 위젯을 사용합니다. 사용자가 원하는만큼 많은 색상 범주를 추가할 수 있습니다. 각 범주 마다 최소 및 최대 범위를 선택한 다음 마지막으로 색상 행을 클릭하면 색상 위젯에서 색상을 선택할 수 있습니다.<br>  <br>그림 24.66: 기복 색상 범주를 직접 설정하기<br>패널 오른쪽에 있는 버튼들로 다음 작업을 할 수 있습니다: 색상 범주를 추가하거나 제거하고, 이미 정의된 색상 범주의 순서를 변경하고, 색상 범주를 가진 기존 파일을 열고, 현재 색상 범주를 파일로 저장할 수 있습니다. |
| **Relief** | OUTPUT | [raster]<br>기본값: [Save to temporary file] | Specify the output relief raster layer. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |
| **Frequency distribution** 선택적 | FREQUENCY_DISTRIBU | [vector: table]<br>기본값: [Skip output] | Specify the CSV table for the output frequency distribution. *One of*:<br>• 산출물 건너뛰기<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Relief** | OUTPUT | [raster] | 산출 기복 래스터 레이어 |
| **Frequency distribution** | OUTPUT | [vector: table] | 산출 도수 분포 |



그림 24.65: 기복 레이어

### 파이썬 코드

**알고리즘 ID**: `qgis:relief`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 험상 지수

라일리 (Shawn J. Riley) 와 디글로리아 (Stephen D. Degloria) 의 1999 년 논문 에서 설명하는 지형 이질성 (heterogeneity) 의 정량적 측정을 계산합니다. 모든 위치에서 3x3 픽셀 그리드 내부의 표고 변화를 요약해서 험상 지수 (Ruggedness Index) 를 계산합니다.

각 픽셀은 중심 셀과 중심 셀을 둘러싼 셀 8 개의 표고 차를 담고 있습니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Elevation layer** | `INPUT` | [raster] | DTM 래스터 레이어 |
| **Z factor** | `Z_FACTOR` | [numeric: double]<br>기본값: 1.0 | 수직 과장 (exaggeration). 이 파라미터는 Z 단위가 X 및 Y 단위와, 예를 들어 피트와 미터처럼 다를 때 유용합니다. 이 파라미터를 사용해서 차이를 조정할 수 있습니다. 이 파라미터의 값을 증가시키면 마지막 결과물을 과장할 것입니다. (험상이 더 심하게 보이게 만듭니다.) 기본값은 1(과장 없음) 입니다. |

다음 페이지에 계속

표 24.430 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Ruggedness** | OUTPUT | [raster]<br>기본값: [Save to temporary file] | Specify the output ruggedness raster layer. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Ruggedness** | OUTPUT | [raster] | 산출 험상 래스터 레이어 |

### 파이썬 코드

**Algorithm ID**: `native:ruggednessindex`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 경사

입력 래스터 레이어로부터 경사도 (slope) 를 계산합니다. 경사도란 지형의 기울어진 각도를 말하며, **도 단위** 로 표현됩니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Elevation layer** | INPUT | [raster] | DTM 래스터 레이어 |
| **Z factor** | Z_FACTOR | [numeric: double]<br>기본값: 1.0 | 수직 과장 (exaggeration). 이 파라미터는 Z 단위가 X 및 Y 단위와, 예를 들어 피트와 미터처럼 다를 때 유용합니다. 이 파라미터를 사용해서 차이를 조정할 수 있습니다. 이 파라미터의 값을 증가시키면 마지막 결과물을 과장할 것입니다. (경사가 더 심하게 보이게 만듭니다.) 기본값은 1(과장 없음) 입니다. |
| **Slope** | OUTPUT | [raster]<br>기본값: [Save to temporary file] | Specify the output slope raster layer. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

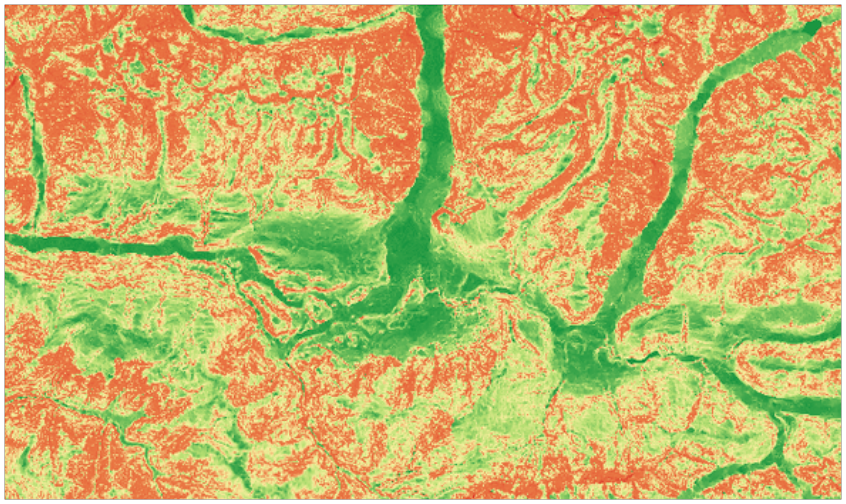


그림 24.67: 낮은 값 (빨간색) 에서 높은 값 (녹색) 까지의 변화를 보여주는 험상 레이어

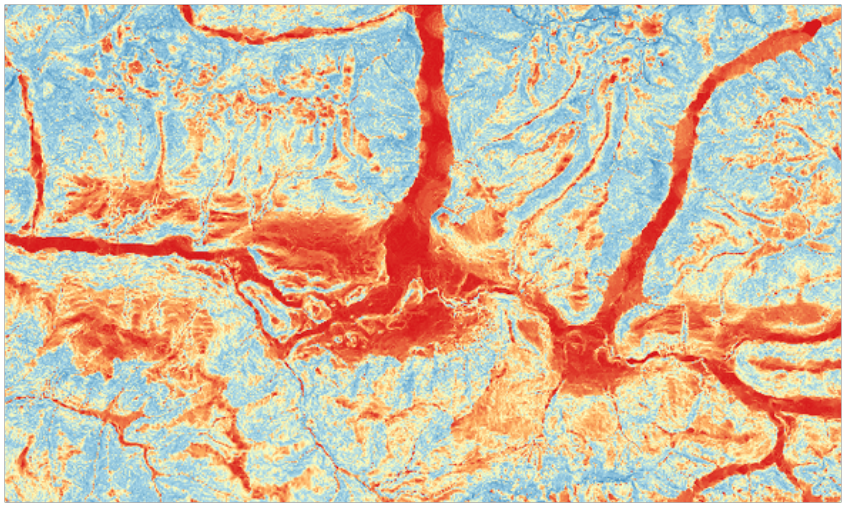


그림 24.68: 평평한 영역은 빨간색, 경사진 영역은 파란색

#### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Slope** | OUTPUT | [raster] | 산출 경사도 래스터 레이어 |

#### 파이썬 코드

**Algorithm ID**: native:slope

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 24.1.21 래스터 도구

#### 래스터 정렬

래스터가 동일한 셀 크기를 가지도록 리샘플링하고 참조 래스터와 동일한 좌표계로 재투영해서 래스터를 정렬시킵니다.

> **경고**
>
> **이 알고리즘은**모델 설계자 **맥락에서만 사용할 수 있습니다.** 다른 맥락의 경우래스터들 정렬 을 대신 사용하십시오.

#### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | INPUT | [raster] | 정렬시킬 입력 래스터 레이어 |

다음 페이지에 계속

표 24.434 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Resampling method** | RESAMPLING_METHOD | [enumeration] 기본값: 0 | Method to use for input layer resampling. Available options are:<br>• 0 — 최근접 이웃<br>• 1 — 쌍선형 (2x2 커널)<br>• 2 — 3 차 (4x4 커널)<br>• 3 — 3 차 B-스플라인 (4x4 커널)<br>• 4 — 란초시 (6x6 커널)<br>• 5 — 평균<br>• 6 — 모드<br>• 7 — 최대값<br>• 8 — 최소값<br>• 9 — 중앙값<br>• 10 — 제 1 사분위수 (Q1)<br>• 11 — 제 3 사분위수 (Q3) |
| **Rescale values according to the cell size** | RESCALE | [boolean] 기본값: No | |
| **Reference layer** | REFERENCE_LAYER | [raster] | 입력 레이어에 적용될 범위, 셀 크기, 좌표계를 가져오는 데 사용할 래스터 레이어입니다. |
| **Override reference CRS** 선택적 | CRS | [crs] | 참조 레이어 좌표계 대신 사용할 좌표계 |
| **Override reference cell size X** 선택적 | CELL_SIZE_X | [numeric: double] | 참조 레이어의 X 방향 셀 크기 대신 사용할 X 방향 셀 크기 |
| **Override reference cell size Y** 선택적 | CELL_SIZE_Y | [numeric: double] | 참조 레이어의 Y 방향 셀 크기 대신 사용할 Y 방향 셀 크기 |
| **Override reference grid offset X** 선택적 | GRID_OFFSET_X | [numeric: double] | 셀 그리드에 적용할 X 방향 오프셋 |
| **Override reference grid offset Y** 선택적 | GRID_OFFSET_Y | [numeric: double] | 셀 그리드에 적용할 Y 방향 오프셋 |

다음 페이지에 계속

표 24.434 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Clip to extent**<br>선택적 | `EXTENT` | [extent] | 산출 래스터 레이어의 범위를 지정합니다. 내부적으로 타일 크기의 배수로 확장될 것입니다.<br>다음과 같은 메소드를 사용할 수 있습니다:<br>• 레이어로부터 계산…: 현재 프로젝트에 불러온 레이어의 범위를 사용합니다.<br>• 레이어 맵으로부터 계산…: 활성 프로젝트의조판 맵 항목 범위를 사용합니다.<br>• 북마크로부터 계산…: 저장되어 있는북마크 의 범위를 사용합니다.<br>• 맵 캔버스 범위 사용<br>• 캔버스에 그리기: 연산에 넣을 범위를 정하는 직사각형을 클릭 & 드래그합니다.<br>• 좌표를 `xmin, xmax, ymin, ymax` 로 입력 |
| **Aligned raster** | `OUTPUT` | [raster] | 픽셀을 리샘플링한 산출 래스터 |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Aligned raster** | `OUTPUT` | [raster] | 픽셀을 리샘플링한 산출 래스터 |

### 파이썬 코드

**알고리즘 ID**: `native:alignsingleraster`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 래스터들 정렬

래스터들이 동일한 셀 크기를 가지도록 리샘플링하고 참조 래스터와 동일한 좌표계로 재투영해서 래스터들을 정렬시킵니다.

> **⚠ 경고**
>
> **이 알고리즘은**모델 설계자 **맥락에서는 사용할 수 없습니다.** 래스터 정렬 을 대신 사용하십시오.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layers** | `INPUT` | [raster] [list] | 입력 래스터 레이어들의, 리샘플링 옵션 (`QgsProcessingParameterAlignRasterLayers` 로써 채워집니다—GUI 에서 선택한 각 레이어에 대해 *Configure Raster*…버튼을 클릭하면 됩니다) 을 합친 목록:<br>**입력 레이어 [문자열] (`inputFile`)**<br>정렬시킬 입력 레이어의 전체 경로<br>**산출 파일 [문자열] (`outputFile`)**<br>그에 대응하는 정렬된 산출 레이어의 전체 경로<br>**리샘플링 메소드 [목록 (enumeration)] (`resampleMethod`)**<br>Method to use for input layer resampling. `Available options` are:<br>• 0 —최근접 이웃<br>• 1 —쌍선형 (2x2 커널)<br>• 2 —3 차 (4x4 커널)<br>• 3 —3 차 B-스플라인 (4x4 커널)<br>• 4 —란초시 (6x6 커널)<br>• 5 —평균<br>• 6 —모드<br>• 7 —최대값<br>• 8 —최소값<br>• 9 —중앙값<br>• 10 —제 1 사분위수 (Q1)<br>• 11 —제 3 사분위수 (Q3)<br>**셀 크기에 따른 크기 조정 값** [불 (boolean)] (`rescale`) |
| **Reference layer** | `REFERENCE_LAYER` | [raster] | 입력 레이어에 적용될 범위, 셀 크기, 좌표계를 가져오는 데 사용할 래스터 레이어입니다. |
| **Override reference CRS**<br>선택적 | `CRS` | [crs] | 참조 레이어 좌표계 대신 사용할 좌표계 |
| **Override reference cell size X**<br>선택적 | `CELL_SIZE_X` | [numeric: double] | 참조 레이어의 X 방향 셀 크기 대신 사용할 X 방향 셀 크기 |

다음 페이지에 계속

표 24.436 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Override reference cell size Y**<br>선택적 | `CELL_SIZE_Y` | [numeric: double] | 참조 레이어의 Y 방향 셀 크기 대신 사용할 Y 방향 셀 크기 |
| **Override reference grid offset X**<br>선택적 | `GRID_OFFSET_X` | [numeric: double] | 셀 그리드에 적용할 X 방향 오프셋 |
| **Override reference grid offset Y**<br>선택적 | `GRID_OFFSET_Y` | [numeric: double] | 셀 그리드에 적용할 Y 방향 오프셋 |
| **Clip to extent**<br>선택적 | `EXTENT` | [extent] | 산출 래스터 레이어의 범위를 지정합니다. 내부적으로 타일 크기의 배수로 확장될 것입니다.<br>다음과 같은 메소드를 사용할 수 있습니다:<br>• 레이어로부터 계산…: 현재 프로젝트에 불러온 레이어의 범위를 사용합니다.<br>• 레이어 맵으로부터 계산…: 활성 프로젝트의조판 맵 항목 범위를 사용합니다.<br>• 북마크로부터 계산…: 저장되어 있는북마크 의 범위를 사용합니다.<br>• 맵 캔버스 범위 사용<br>• 캔버스에 그리기: 연산에 넣을 범위를 정하는 직사각형을 클릭 & 드래그합니다.<br>• 좌표를 `xmin, xmax, ymin, ymax` 로 입력 |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Aligned rasters** | `OUTPUT_LAYERS` | [raster] [list] | 픽셀을 리샘플링한 산출 래스터들 |

### 파이썬 코드

**알고리즘 ID**: `native:alignrasters`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 맵을 래스터로 변환

맵 캔버스 내용의 래스터 이미지를 생성합니다.

맵 테마 를 선택하면 사전 선택한 레이어 집합을 각 레이어를 위해 정의된 스타일로 렌더링할 수 있습니다.

아무 맵 테마도 설정하지 않은 경우 대신 단일 레이어를 선택할 수 있습니다.

맵 테마나 레이어 둘 다 설정하지 않은 경우, 현재 맵 내용을 렌더링할 것입니다. 입력한 최소 범위를 타일 크기의 배수가 되도록 내부적으로 확장할 것입니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Minimum extent to render (xmin, xmax, ymin, ymax)** | `EXTENT` | [extent] | 산출 래스터 레이어의 범위를 지정합니다. 내부적으로 타일 크기의 배수로 확장될 것입니다.<br>다음과 같은 메소드를 사용할 수 있습니다:<br>• 레이어로부터 계산…: 현재 프로젝트에 불러온 레이어의 범위를 사용합니다.<br>• 레이어 맵으로부터 계산…: 활성 프로젝트의조판 맵 항목 범위를 사용합니다.<br>• 북마크로부터 계산…: 저장되어 있는북마크 의 범위를 사용합니다.<br>• 맵 캔버스 범위 사용<br>• 캔버스에 그리기: 연산에 넣을 범위를 정하는 직사각형을 클릭 & 드래그합니다.<br>• 좌표를 `xmin, xmax, ymin, ymax` 로 입력 |
| **Tile size** | `TILE_SIZE` | [numeric: integer]<br>기본값: 1024 | 산출 래스터 레이어의 타일 크기를 지정합니다. 최소값은 64 입니다. |
| **Map units per pixel** | `MAP_UNITS_PER_P` | [numeric: double]<br>기본값: 100.0 | (맵 단위) 픽셀 크기입니다. 최소값은 0.0 입니다. |
| **Make background transparent** | `MAKE_BACKGROUNI` | [boolean]<br>기본값: False | 투명한 배경을 가진 맵을 내보낼 수 있습니다. `True` 로 설정한 경우 (RGB 대신) RGBA 산출물을 생성할 것입니다. |
| **Map theme to render**<br>선택적 | `MAP_THEME` | [enumeration] | 렌더링을 위해 기존맵 테마 를 사용합니다. |
| **Single layer to render**<br>선택적 | `LAYER` | [enumeration] | 렌더링할 단일 레이어를 선택합니다. |
| **Output layer** | `OUTPUT` | [raster]<br>기본값: `[Save to temporary file]` | Specification of the output raster. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output layer** | `OUTPUT` | [raster] | 산출 래스터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:rasterize`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## NODATA 셀 채우기

입력 래스터에 있는 NODATA 값을 선택한 값으로 재설정해서 NODATA 픽셀이 없는 래스터 데이터셋을 산출합니다.

이 알고리즘은 입력 래스터의 데이터 유형을 유지합니다. 예를 들어 정수형 래스터에 부동소수점형 채우기 값을 적용할 경우 소수점 이하 값을 잘라낼 것입니다.

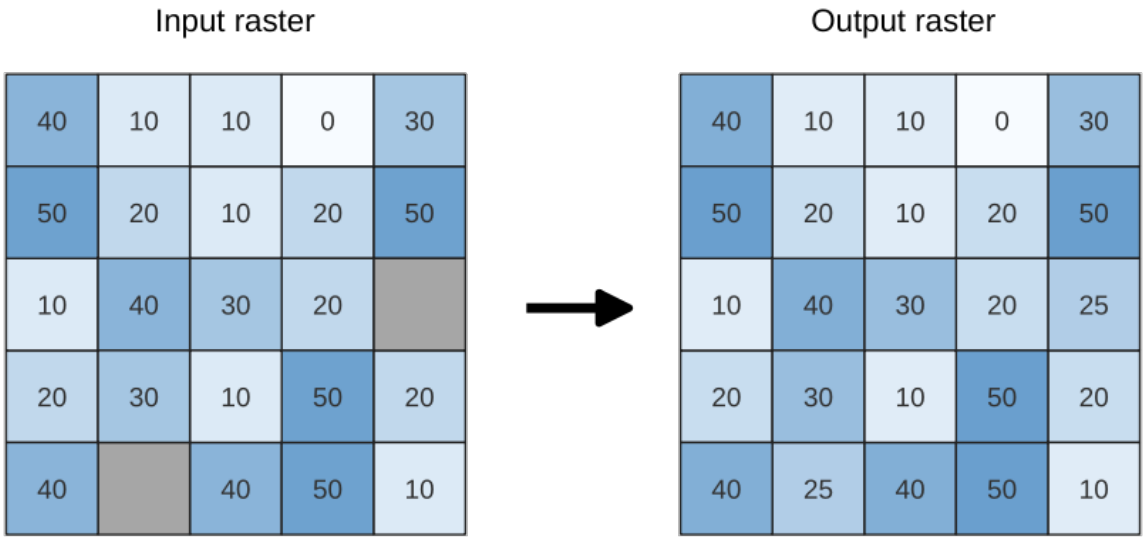



그림 24.69: 래스터의 NODATA 값 (회색) 채우기

### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input raster** | `INPUT` | [raster] | 공간 처리할 래스터 |
| **Band number** | `BAND` | [raster band]<br>기본값: 1 | 래스터의 밴드 |
| **Fill value** | `FILL_VALUE` | [numeric: double]<br>기본값: 1.0 | NODATA 픽셀 용으로 사용할 값 설정 |
| **Output raster** | `OUTPUT` | [raster]<br>기본값: `[Save to temporary file]` | Specification of the output raster. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

#### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Creation options**<br>선택적 | `CREATION_OPTIONS`<br>(for QGIS <= 3.42, this was `CREATE_OPTIONS`) | [string]<br>기본값: ‘’ | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL* 드라이버 옵션 참조)<br>배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (`|`) 로 복수의 옵션들을 분리하십시오. |

#### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output raster** | `OUTPUT` | [raster] | 채워진 데이터 셀을 가진 산출 래스터 레이어 |

#### 파이썬 코드

**알고리즘 ID**: `native:fillnodata`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### XYZ 타일 생성하기 (목록)

현재 QGIS 프로젝트를 개별 이미지로 목록 (directory) 구조에 사용해서 래스터"XYZ"타일을 생성합니다.
생성된 타일을 맵 레이어로 사용해서 리플렛 HTML 산출 파일을 생성할지를 선택할 수 있습니다.

#### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Extent (xmin, xmax, ymin, ymax)** | `EXTENT` | [extent] | 타일의 범위를 지정합니다. 내부적으로 타일 크기의 배수로 확장될 것입니다.<br>다음과 같은 메소드를 사용할 수 있습니다:<br>• 레이어로부터 계산…: 현재 프로젝트에 불러온 레이어의 범위를 사용합니다.<br>• 레이어 맵으로부터 계산…: 활성 프로젝트의조판 맵 항목 범위를 사용합니다.<br>• 북마크로부터 계산…: 저장되어 있는북마크 의 범위를 사용합니다.<br>• 맵 캔버스 범위 사용<br>• 캔버스에 그리기: 연산에 넣을 범위를 정하는 직사각형을 클릭 & 드래그합니다.<br>• 좌표를 `xmin, xmax, ymin, ymax` 로 입력 |
| **Minimum zoom** | `ZOOM_MIN` | [numeric: integer]<br>기본값: 12 | 최소값은 0, 최대값은 25 입니다. |
| **Maximum zoom** | `ZOOM_MAX` | [numeric: integer]<br>기본값: 12 | 최소값은 0, 최대값은 25 입니다. |
| **DPI** | `DPI` | [numeric: integer]<br>기본값: 96 | 최소값은 48, 최대값은 600 입니다. |
| **Background color** 선택적 | `BACK-GROUND_COLOR` | [color]<br>기본값: QColor(0, 0, 0, 0) | 타일용 배경 색상을 선택합니다. |
| **Enable antialiasing** | `ANTIALIAS` | [boolean]<br>기본값: True | 위신호 제거 (antialiasing) 를 활성화할지 여부를 결정합니다. |
| **Tile format** | `TILE_FORMAT` | [enumeration]<br>기본값: 0 | 다음 가운데 하나로 지정할 수 있습니다:<br>• 0 —PNG<br>• 1 —JPG |
| **Quality (JPG only)** 선택적 | `QUALITY` | [numeric: integer]<br>기본값: 75 | 최소값은 1, 최대값은 100 입니다. |

다음 페이지에 계속

표 24.443 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Metatile size**<br>선택적 | METATILESIZE | [numeric: integer]<br>기본값: 4 | XYZ 타일 생성시 사용자 지정 메타타일 크기를 지정합니다. 값이 클수록 타일 렌더링 속도와 라벨 작업이 향상되지만 (라벨이 없는 틈 (gap) 이 적어집니다), 그 대신 더 많은 메모리를 사용할 수도 있습니다. 최소값은 1, 최대값은 20 입니다. |
| **Tile width**<br>선택적 | TILE_WIDTH | [numeric: integer]<br>기본값: 256 | 최소값은 1, 최대값은 4096 입니다. |
| **Tile height**<br>선택적 | TILE_HEIGHT | [numeric: integer]<br>기본값: 256 | 최소값은 1, 최대값은 4096 입니다. |
| **Use inverted tile Y axis (TMS conventions)** | TMS_CONVENTION | [boolean]<br>기본값: False | |
| **Output directory** | OUTPUT_DIRECTORY | [folder]<br>기본값: [Save to temporary folder] | Specification of the output directory (for the tiles). *One of*:<br>• 산출물 건너뛰기<br>• 임시 디렉터리에 저장<br>• 디렉터리에 저장 |
| **Output html (Leaflet)**<br>선택적 | OUTPUT_HTML | [html]<br>기본값: [Save to temporary file] | Specification of the output HTML file. *One of*:<br>• 산출물 건너뛰기<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Leaflet HTML output title**<br>선택적 | HTML_TITLE | [string]<br>기본값: 설정하지 않음 | 리플렛 HTML 산출 파일에 쓰이는 HTML <title> 태그입니다. |
| **Leaflet HTML output attribution**<br>선택적 | HTML_ATTRIBUTIC | [string]<br>기본값: 설정하지 않음 | 리플렛 HTML 산출 파일에 쓰이는 사용자 정의 맵 속성입니다. HTML 링크를 사용할 수 있습니다. |
| **Include OpenStreetMap basemap in Leaflet HTML output** | HTML_OSM | [boolean]<br>기본값: False | 리플렛 HTML 산출 파일에 오픈스트리트맵의 기저 맵 (basemap) 레이어 (소스: https://tile.openstreetmap.org) 를 포함시킵니다. 알맞은 맵 속성을 자동으로 추가합니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output directory** | `OUTPUT_DIRECTORY` | [folder] | 산출물 디렉터리 (타일 용) |
| **Output html (Leaflet)** | `OUTPUT_HTML` | [html] | 산출 HTML (리플렛) 파일 |

### 파이썬 코드

**알고리즘 ID**: `native:tilesxyzdirectory`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## XYZ 타일 생성하기 (MBTiles)

현재 QGIS 프로젝트를 개별 파일로“MBTiles”포맷에 사용해서 래스터“XYZ”타일을 생성합니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Extent (xmin, xmax, ymin, ymax)** | `EXTENT` | [extent] | 타일의 범위를 지정합니다. 내부적으로 타일 크기의 배수로 확장될 것입니다.<br>다음과 같은 메소드를 사용할 수 있습니다:<br>• 레이어로부터 계산…: 현재 프로젝트에 불러온 레이어의 범위를 사용합니다.<br>• 레이어 맵으로부터 계산…: 활성 프로젝트의조판 맵 항목 범위를 사용합니다.<br>• 북마크로부터 계산…: 저장되어 있는북마크 의 범위를 사용합니다.<br>• 맵 캔버스 범위 사용<br>• 캔버스에 그리기: 연산에 넣을 범위를 정하는 직사각형을 클릭 & 드래그합니다.<br>• 좌표를 `xmin, xmax, ymin, ymax` 로 입력 |
| **Minimum zoom** | `ZOOM_MIN` | [numeric: integer]<br>기본값: 12 | 최소값은 0, 최대값은 25 입니다. |

다음 페이지에 계속

표 24.446 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Maximum zoom** | `ZOOM_MAX` | [numeric: integer]<br>기본값: 12 | 최소값은 0, 최대값은 25 입니다. |
| **DPI** | `DPI` | [numeric: integer]<br>기본값: 96 | 최소값은 48, 최대값은 600 입니다. |
| **Background color**<br>선택적 | `BACK-GROUND_COLOR` | [color]<br>기본값: QColor(0, 0, 0, 0) | 타일용 배경 색상을 선택합니다. |
| **Enable antialiasing** | `ANTIALIAS` | [boolean]<br>기본값: True | 위신호 제거 (antialiasing) 를 활성화할지 여부를 결정합니다. |
| **Tile format** | `TILE_FORMAT` | [enumeration]<br>기본값: 0 | 다음 가운데 하나로 지정할 수 있습니다:<br>• 0 —PNG<br>• 1 —JPG |
| **Quality (JPG only)**<br>선택적 | `QUALITY` | [numeric: integer]<br>기본값: 75 | 최소값은 1, 최대값은 100 입니다. |
| **Metatile size**<br>선택적 | `METATILESIZE` | [numeric: integer]<br>기본값: 4 | XYZ 타일 생성시 사용자 지정 메타타일 크기를 지정합니다. 값이 클수록 타일 렌더링 속도와 라벨 작업이 향상되지만 (라벨이 없는 틈 (gap) 이 적어집니다), 그 대신 더 많은 메모리를 사용할 수도 있습니다. 최소값은 1, 최대값은 20 입니다. |
| **Output file (for MBTiles)** | `OUTPUT_FILE` | [file]<br>기본값: `[Save to temporary file]` | Specification of the output file. *One of*:<br>• 산출물 건너뛰기<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output file (for MBTiles)** | `OUTPUT_FILE` | [file] | 산출 파일 |

### 파이썬 코드

**알고리즘 ID**: `native:tilesxyzmbtiles`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 24.1.22 벡터 분석

#### 필드에 대한 기본 통계

Generates basic statistics for a field of the attribute table of a vector layer. Numeric, date, time and string fields are supported. The statistics returned will depend on the field type.

Statistics can be generated as a table or an HTML file and are available in the *Processing → Results viewer*.

**기본 메뉴**: *Vector → Analysis Tools*

#### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input vector** | `INPUT_LAYER` | [vector: any] | 통계를 계산할 벡터 레이어 |
| **Field to calculate statistics on** | `FIELD_NAME` | [tablefield: any] | 통계를 계산할 지원하는 테이블 필드 |
| **Statistics**<br>선택적 | `OUTPUT` | [vector: table]<br>기본값: `[Create temporary layer]` | Specify the output table for the generated statistics. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |
| **Statistics report**<br>선택적 | `OUTPUT_HTML_FILE` | [html]<br>기본값: `[Save to temporary file]` | Specification of the file for the calculated statistics. *One of*:<br>• 산출물 건너뛰기<br>• 임시 파일로 저장<br>• 파일로 저장… |

#### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Statistics** | `OUTPUT` | [vector: table] | Table containing the calculated statistics |
| **Statistics report** | `OUTPUT_HTML_FILE` | [html] | 계산된 통계를 가진 HTML 파일 |
| **Count** | `COUNT` | [numeric: integer] | |
| **Number of unique values** | `UNIQUE` | [numeric: integer] | |
| **Number of empty (null) values** | `EMPTY` | [numeric: integer] | |
| **Number of non-empty values** | `FILLED` | [numeric: integer] | |
| **Minimum value** | `MIN` | [same as input] | |
| **Maximum value** | `MAX` | [same as input] | |
| **Minimum length** | `MIN_LENGTH` | [numeric: integer] | |
| **Maximum length** | `MAX_LENGTH` | [numeric: integer] | |

다음 페이지에 계속

표 24.449 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Mean length** | MEAN_LENGTH | [numeric: double] | |
| **Coefficient of Variation** | CV | [numeric: double] | |
| **Sum** | SUM | [numeric: double] | |
| **Mean value** | MEAN | [numeric: double] | |
| **Standard deviation** | STD_DEV | [numeric: double] | |
| **Range** | RANGE | [numeric: double] | |
| **Median** | MEDIAN | [numeric: double] | |
| **Minority (rarest occurring value)** | MINORITY | [same as input] | |
| **Majority (most frequently occurring value)** | MAJORITY | [same as input] | |
| **First quartile** | FIRSTQUARTILE | [numeric: double] | |
| **Third quartile** | THIRDQUARTILE | [numeric: double] | |
| **Interquartile Range (IQR)** | IQR | [numeric: double] | |

### 파이썬 코드

**Algorithm ID**: native:basicstatisticsforfields

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 라인을 따라 오르막

라인 도형을 따라 총 오르막 (climb) 과 내리막 (descent) 을 계산합니다. 입력 레이어가 Z 값을 가지고 있어야만 합니다. Z 값을 사용할 수 없는 경우, 늘어뜨리기 *(래스터로부터 Z 값 설정하기)* 알고리즘을 사용해서 DEM 레이어로부터 Z 값을 추가할 수도 있습니다.

산출 레이어는 입력 레이어에 각 라인 도형의 총 오르막 (climb), 총 내리막 (descent), 최저 표고 (minelev) 및 최고 표고 (maxelev) 를 담고 있는 필드를 추가한 복사본입니다. 입력 레이어가 이렇게 추가되는 필드와 동일한 이름을 가진 필드를 가지고 있는 경우, 기존 필드를 재명명할 것입니다. ("name_2", "name_3"등으로 첫 번째로 일치하지 않는 이름을 찾아 필드명을 수정할 것입니다.)

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Line layer** | INPUT | [vector: line] | 오르막을 계산할 라인 레이어입니다. Z 값을 가지고 있어야만 합니다. |

다음 페이지에 계속

표 24.450 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Climb layer** | `OUTPUT` | [vector: line]<br>기본값: `[Create temporary layer]` | Specification of the output (line) layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Climb layer** | `OUTPUT` | [vector: line] | 오르막 계산에서 나온 결과를 가진 새 속성을 담고 있는 라인 레이어 |
| **Total climb** | `TOTALCLIMB` | [numeric: double] | 입력 레이어에 있는 모든 라인 도형에 대한 오르막 합계 |
| **Total descent** | `TOTALDESCENT` | [numeric: double] | 입력 레이어에 있는 모든 라인 도형에 대한 내리막 합계 |
| **Minimum elevation** | `MINELEVATION` | [numeric: double] | 레이어에 있는 도형 가운데 최저 표고 |
| **Maximum elevation** | `MAXELEVATION` | [numeric: double] | 레이어에 있는 도형 가운데 최고 표고 |

### 파이썬 코드

**Algorithm ID**: `native:climbalongline`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 폴리곤에서 포인트 개수 세기

포인트 레이어와 폴리곤 레이어를 받아 폴리곤 레이어의 각 폴리곤 내부에 들어오는 포인트 레이어의 포인트 개수를 셉니다.

입력 폴리곤 레이어와 정확히 동일한 내용을 가지면서, 각 폴리곤에 대응하는 포인트 개수를 가진 추가 필드를 담고 있는 새 폴리곤 레이어를 생성합니다.

각 포인트에 가중치를 할당하기 위해 부가적인 가중치 필드를 사용할 수 있습니다. 또는, 유일 범주 필드를 지정할 수 있습니다. 이 두 옵션을 모두 사용하는 경우, 가중치 필드를 우선하고 유일 범주 필드는 무시할 것입니다.

☑ 폴리곤 피처를제자리 피처 수정 할 수 있습니다.

기본 메뉴: *Vector → Analysis Tools*

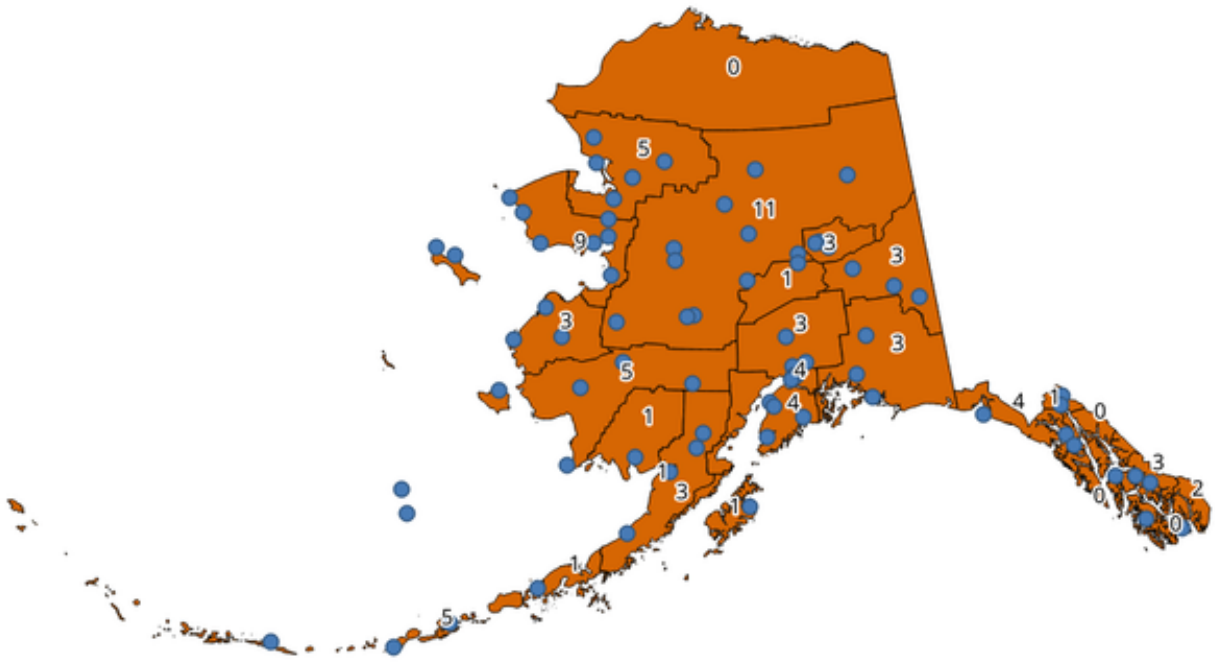



그림 24.70: 포인트 개수를 표시하는 폴리곤 라벨

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Polygons** | `POLYGONS` | [vector: polygon] | 포인트의 개수를 셀 피처를 가진 폴리곤 레이어 |
| **Points** | `POINTS` | [vector: point] | 개수를 셀 포인트 레이어 |
| **Weight field**<br>선택적 | `WEIGHT` | [tablefield: any] | 포인트 레이어의 필드. 생성된 개수는 폴리곤이 담고 있는 포인트의 가중치 필드의 합계가 될 것입니다. 가중치 필드가 숫자가 아닌 경우, 개수는 `0` 이 됩니다. |
| **Class field**<br>선택적 | `CLASSFIELD` | [tablefield: any] | 선택한 속성을 기반으로 포인트를 범주화해서 폴리곤 내부에 동일한 속성값을 가진 포인트가 여러 개 있는 경우, 하나로 셉니다. 따라서 폴리곤에 들어오는 포인트의 최종 개수는 폴리곤에서 발견된 서로 다른 범주의 개수입니다. |
| **Count field name** | `FIELD` | [string]<br>기본값: 'NUMPOINTS' | 포인트 개수를 저장할 필드의 이름 |

다음 페이지에 계속

표 24.452 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Count** | `OUTPUT` | [vector: polygon]<br>기본값: `[Create temporary layer]` | Specification of the output layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

#### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Count** | `OUTPUT` | [vector: polygon] | 포인트 개수를 가진 새 열을 담고 있는 속성 테이블을 가진 산출 레이어 |

#### 파이썬 코드

**알고리즘 ID**: `native:countpointsinpolygon`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### DBSCAN 군집 형성

이상값 (noise) (DBSCAN) 알고리즘을 가진 응용 프로그램의 밀도 기반 공간 군집 형성의 2 차원 구현을 기반으로 포인트 피처를 군집시킵니다.

이 알고리즘은 최소 군집 크기 파라미터 및 군집 포인트 사이에 허용된 최장 거리 파라미터, 2 개를 필요로 합니다.

**더 보기**

*ST-DBSCAN* 군집 형성, *k*-평균 군집 형성

### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | INPUT | [vector: point] | 분석할 레이어 |
| **Minimum cluster size** | MIN_SIZE | [numeric: integer]<br>기본값: 5 | 군집을 형성할 피처의 최소 개수 |
| **Maximum distance between clustered points** | EPS | [numeric: double]<br>기본값: 1.0 | 이 거리 (eps) 를 초과하면 동일 군집에 속할 수 없습니다. |
| **Clusters** | OUTPUT | [vector: point]<br>기본값: [Create temporary layer] | Specify the vector layer for the result of the clustering. *One of*:<br>• 임시 레이어 생성 (TEMPORARY_OUTPUT)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

#### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Treat border points as noise (DBSCAN*)** | DBSCAN* | [boolean]<br>기본값: False | True 로 설정하면, 군집의 경계 상에 있는 포인트를 군집되지 않은 포인트로 취급하고, 군집 내부에 있는 포인트만 군집된 것으로 태그합니다. |
| **Cluster field name** | FIELD_NAME | [string]<br>기본값: 'CLUSTER_ID' | 연관 군집 번호를 저장할 필드의 이름 |
| **Cluster size field name** | SIZE_FIELD_NAME | [string]<br>기본값: 'CLUSTER_SIZE' | 동일 군집에 있는 피처들의 개수를 가진 필드의 이름 |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Clusters** | OUTPUT | [vector: point] | 원본 피처와 함께 원본 피처가 속한 군집을 설정하는 필드를 가진 벡터 레이어 |
| **Number of clusters** | NUM_CLUSTERS | [numeric: integer] | 발견된 군집의 개수 |

### 파이썬 코드

**알고리즘 ID**: `native:dbscanclustering`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 거리 매트릭스

어떤 포인트 피처와 동일 레이어 또는 다른 레이어에 있는 최근접 피처 사이의 거리를 계산합니다.

**기본 메뉴**: *Vector ▸ Analysis Tools*

**더 보기**

최근접으로 속성 결합하기

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input point layer** | `INPUT` | [vector: point] | (포인트 **로부터**) 거리 매트릭스를 계산할 포인트 레이어 |
| **Input unique ID field** | `INPUT_FIELD` | [tablefield: any] | 입력 레이어의 피처를 유일하게 식별하기 위해 사용하는 필드입니다. 산출 속성 테이블에 사용됩니다. |
| **Target point layer** | `TARGET` | [vector: point] | (포인트 **로**) 검색할 최근접 포인트 (들) 를 담고 있는 포인트 레이어 |
| **Target unique ID field** | `TARGET_FIELD` | [tablefield: any] | 대상 레이어의 피처를 유일하게 식별하기 위해 사용하는 필드입니다. 산출 속성 테이블에 사용됩니다. |
| **Output matrix type** | `MATRIX_TYPE` | [enumeration] 기본값: 0 | 서로 다른 계산 유형을 사용할 수 있습니다:<br>• 0 —선형 (N * *k* x 3) 거리 매트릭스: 각 입력 포인트에 대해 각 *k* 개의 최근접 대상 포인트까지의 거리를 보고합니다. 산출 매트릭스는 입력 포인트마다 최대 *k* 개의 행으로 이루어지며, 각 행은 *InputID*, *TargetID* 및 *Distance* 열 3 개를 가지고 있습니다.<br>• 1 —표준 (N x T) 거리 매트릭스<br>• 2 —거리 매트릭스 요약 (평균, 표준 편차, 최소값, 최대값): 각 입력 포인트마다, 대상 포인트까지의 거리에 대한 통계를 보고합니다. |

다음 페이지에 계속

표 24.457 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Use only the nearest (k) target points** | `NEAREST_POINTS` | [numeric: integer]<br>기본값: 0 | 대상 레이어에 있는 모든 포인트 (*0*) 까지의 거리를 계산하거나, 또는 최근접 피처의 개수 (*k*) 로 제한할 수 있습니다. |
| **Distance matrix** | `OUTPUT` | [vector: point]<br>기본값: `[Create temporary layer]` | Specification of the output vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Distance matrix** | `OUTPUT` | [vector: point] | 각 입력 피처별로 계산한 거리를 담고 있는 포인트 (또는"Linear (N * *k* x 3)" 의 경우 멀티포인트) 벡터 레이어입니다. 이 레이어의 피처와 속성 테이블은 선택한 산출 매트릭스 유형에 따라 달라집니다. |

### 파이썬 코드

**알고리즘 ID**: `qgis:distancematrix`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 최근접 허브까지의 거리 (라인에서 허브로)

입력 벡터 레이어의 각 피처를 대상 레이어에 있는 최근접 피처에 결합 (join) 시키는 라인을 생성합니다. 각 피처의중심 *(center)* 을 기반으로 거리를 계산합니다.

**더 보기**

최근접 허브까지의 거리 (포인트), 최근접으로 속성 결합하기

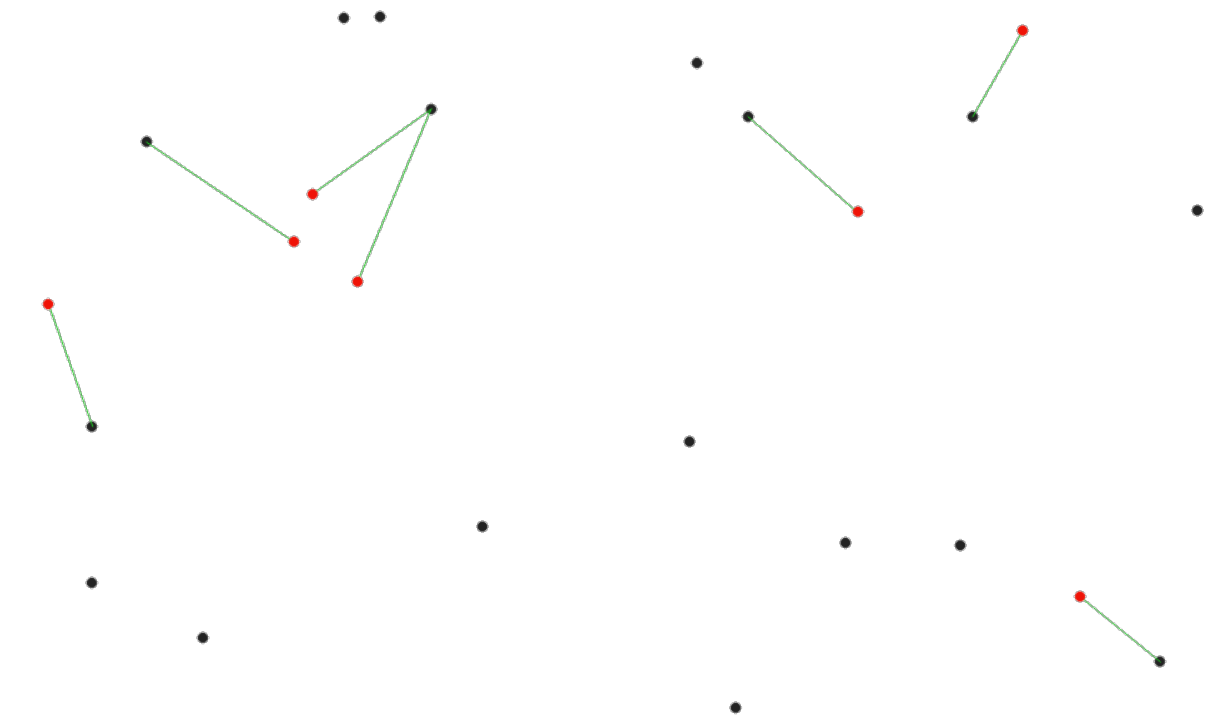


그림 24.71: 빨간색 입력 피처의 최근접 허브를 표시

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Source points layer** | `INPUT` | [vector: geometry] | 최근접 피처를 검색할 벡터 레이어 |
| **Destination hubs layer** | `HUBS` | [vector: geometry] | 검색할 피처를 담고 있는 벡터 레이어 |
| **Hub layer name attribute** | `FIELD` | [tablefield: any] | 대상 레이어의 피처를 유일하게 식별하기 위해 사용하는 필드입니다. 산출 속성 테이블에 사용됩니다. |
| **Measurement unit** | `UNIT` | [enumeration] 기본값: 0 | 최근접 피처까지의 거리를 보고하기 위한 단위<br>• 0 —미터<br>• 1 —피트<br>• 2 —마일<br>• 3 —킬로미터<br>• 4 —레이어 단위 |

다음 페이지에 계속

표 24.459 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Hub distance** | `OUTPUT` | [vector: line]<br>기본값: `[Create temporary layer]` | Specify the output line vector layer connecting the matching points. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Hub distance** | `OUTPUT` | [vector: line] | 입력 피처, 최근접 피처의 식별자, 계산된 거리의 속성을 가진 라인 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `qgis:distancetonearesthublinetohub`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 최근접 허브까지의 거리 (포인트)

입력 피처의중심 *(center)* 을 표현하는 포인트 레이어를 최근접 피처의 (중심 포인트를 기반으로 하는) 식별자 및 포인트들 사이의 거리를 담고 있는 필드 2 개를 추가해서 생성합니다.

**더 보기**

최근접 허브까지의 거리 (라인에서 허브로), 최근접으로 속성 결합하기

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Source points layer** | `INPUT` | [vector: any] | 최근접 피처를 검색할 벡터 레이어 |
| **Destination hubs layer** | `HUBS` | [vector: any] | 검색할 피처를 담고 있는 벡터 레이어 |
| **Hub layer name attribute** | `FIELD` | [tablefield: any] | 대상 레이어의 피처를 유일하게 식별하기 위해 사용하는 필드입니다. 산출 속성 테이블에 사용됩니다. |
| **Measurement unit** | `UNIT` | [enumeration] 기본값: 0 | 최근접 피처까지의 거리를 보고하기 위한 단위<br>• 0 —미터<br>• 1 —피트<br>• 2 —마일<br>• 3 —킬로미터<br>• 4 —레이어 단위 |
| **Hub distance** | `OUTPUT` | [vector: point] 기본값: `[Create temporary layer]` | Specify the output point vector layer with the nearest hub. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Hub distance** | `OUTPUT` | [vector: point] | 소스 피처의 속성, 최근접 피처의 식별자 및 계산된 거리를 가진, 소스 피처의 중심을 표현하는 포인트 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `qgis:distancetonearesthubpoints`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 라인으로 결합 (허브 라인)

스포크 레이어의 포인트에서 허브 레이어의 일치하는 포인트로 라인을 연결해서 허브 & 스포크 (hub and spoke) 다이어그램을 생성합니다.

허브 포인트의 허브 ID 필드와 스포크 포인트의 스포크 ID 필드 사이의 일치 여부를 바탕으로 어떤 허브가 각 포인트와 연결될지 결정합니다.

입력 레이어가 포인트 레이어가 아닌 경우, 도형의 표면 상에 있는 포인트를 연결 위치로 삼을 것입니다.

측지선 (□□□; geodesic line) 을 부가적으로 생성할 수 있습니다. 측지선이란 타원체의 표면 상에서 두 점을 잇는 최단 경로를 말합니다. 측지 모드를 사용하는 경우, 반대 자오선 (antimeridian; ±180° 경도) 에서 생성된 라인을 분할할 수 있습니다. 이렇게 하면 라인 렌더링을 향상시킬 수 있습니다. 또, 꼭짓점 사이의 거리를 지정할 수 있습니다. 거리가 짧을수록 더 밀집한, 더 정확한 라인을 산출합니다.

**경고**

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

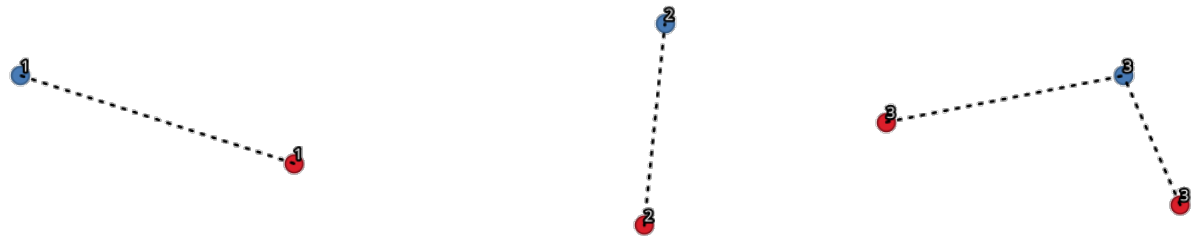



그림 24.72: 공통 필드/속성을 바탕으로 포인트 결합

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Hub layer** | `HUBS` | [vector: geometry] | 입력 레이어 |
| **Hub ID field** | `HUB_FIELD` | [tablefield: any] | 결합을 위한 ID 를 가진 허브 레이어의 필드 |
| **Hub layer fields to copy (leave empty to copy all fields)** 선택적 | `HUB_FIELDS` | [tablefield: any] [list] | 복사할 허브 레이어의 필드 (들). 아무 필드도 선택하지 않을 경우 모든 필드를 복사합니다. |
| **Spoke layer** | `SPOKES` | [vector: geometry] | 부가적인 스포크 포인트 레이어 |
| **Spoke ID field** | `SPOKE_FIELD` | [tablefield: any] | 결합을 위한 ID 를 가진 스포크 레이어의 필드 |

다음 페이지에 계속

표 24.463 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Spoke layer fields to copy (leave empty to copy all fields)** 선택적 | SPOKE_FIELDS | [tablefield: any] [list] | 복사할 스포크 레이어의 필드 (들). 아무 필드도 선택하지 않을 경우 모든 필드를 복사합니다. |
| **Create geodesic lines** | GEODESIC | [boolean] 기본값: False | 측지선 (타원체 표면 상에서 두 점을 잇는 최단 경로) 을 생성합니다. |
| **Hub lines** | OUTPUT | [vector: line] 기본값: [Create temporary layer] | Specify the output hub line vector layer. *One of*:<br>• 임시 레이어 생성 (TEMPORARY_OUTPUT)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Distance between vertices (geodesic lines only)** | GEODESIC_DISTAN | [numeric: double] 기본값: 1000.0 (킬로미터) | 연속하는 꼭짓점 사이의 (킬로미터 단위) 거리. 거리가 짧을수록 더 밀집한, 더 정확한 라인을 산출합니다. |
| **Split lines at antimeridian (±180 degrees longitude)** | ANTIMERIDIAN_SPLIT | [boolean] 기본값: False | ±180° 경도에서 라인을 (라인 렌더링을 향상시키기 위해) 분할합니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Hub lines** | OUTPUT | [vector: line] | 입력 레이어에서 일치하는 포인트들을 연결한 산출 라인 레이어 |

### 파이썬 코드

**알고리즘 ID**: native:hublines

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### k-평균 군집 형성

각 입력 피처에 대해 k-평균 군집 개수를 바탕으로 2 차원 거리를 계산합니다.

k-평균 군집 형성은 피처들을 각 피처가 최근접 평균을 가진 군집에 속하는 k 개의 군집으로 나누는 것을 목적으로 합니다. 평균 포인트는 군집된 피처들의 무게중심 (barycenter) 으로 표현됩니다.

입력 도형이 라인 또는 폴리곤인 경우, 피처의 중심 (centroid) 을 기반으로 군집을 형성합니다.

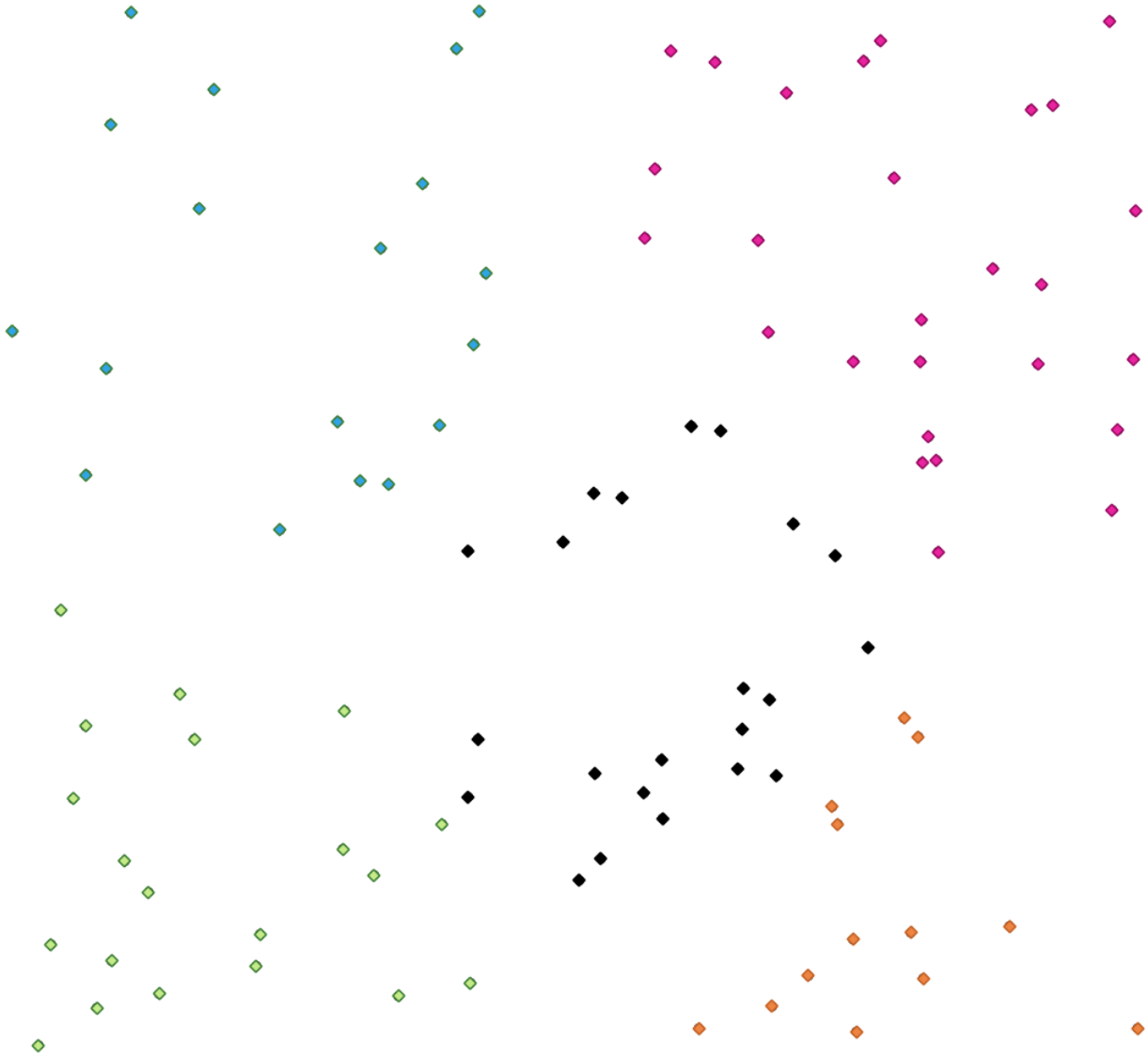


그림 24.73: 범주가 5 개인 포인트 군집들

**더 보기**

*DBSCAN 군집 형성*, *ST-DBSCAN 군집 형성*

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | 분석할 레이어 |
| **Number of clusters** | `CLUSTERS` | [numeric: integer]<br>기본값: 5 | 피처를 통해 생성할 군집의 개수 |
| **Method** | `METHOD` | [enumeration]<br>기본값: 0 | Initial centers selection method. Possible values are:<br>• 0 —Farthest points: the initial centers are selected as the farthest points from each other.<br>• 1 —K-means++: The first center is selected uniformly at random from the set of input features. Subsequent centers are chosen based on a weighted probability distribution that favors features farther from existing centers. This ensures that the initial centers are well-dispersed, reducing the likelihood of poor clustering and improving the algorithm's performance.<br>*References: Arthur, David & Vassilvitskii, Sergei. (2007). K-Means++: The Advantages of Careful Seeding. Proc. of the Annu. ACM-SIAM Symp. on Discrete Algorithms. 8.* |
| **Clusters** | `OUTPUT` | [vector: same as input]<br>기본값: `[Create temporary layer]` | Specify the output vector layer for the generated clusters. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Cluster field name** | `FIELD_NAME` | [string]<br>기본값: 'CLUSTER_ID' | 연관 군집 번호를 저장할 필드의 이름 |
| **Cluster size field name** | `SIZE_FIELD_NAME` | [string]<br>기본값: 'CLUSTER_SIZE' | 동일 군집에 있는 피처들의 개수를 가진 필드의 이름 |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Clusters** | `OUTPUT` | [vector: same as input] | 원본 피처가 속한 군집 및 군집에서의 번호를 지정하는 필드와 함께 원본 피처를 담고 있는 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:kmeansclustering`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 유일 값 목록

속성 테이블 필드에서 유일 값 (unique value) 을 목록화하고 그 개수를 셉니다.

**기본 메뉴**: *Vector → Analysis Tools*

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: any] | 분석할 레이어 |
| **Target field(s)** | `FIELDS` | [tablefield: any] [list] | Field(s) to analyze |
| **Unique values**<br>선택적 | `OUTPUT` | [vector: table]<br>기본값:`[Create temporary layer]` | Specify the summary table layer with unique values. *One of*:<br>• 산출물 건너뛰기<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |
| **HTML report**<br>선택적 | `OUTPUT_HTML_FILE` | [html]<br>기본값:`[Save to temporary file]` | HTML report of unique values in the *Processing → Results viewer*. *One of*:<br>• 산출물 건너뛰기<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Unique values** | `OUTPUT` | [vector: table] | 유일 값을 가진 요약 테이블 레이어 |
| **HTML report** | `OUTPUT_HTML_FILE` | [html] | 유일 값의 HTML 보고서. *Processing → Results viewer* 메뉴로 볼 수 있습니다. |
| **Total unique values** | `TOTAL_VALUES` | [numeric: integer] | 입력 필드에 있는 유일 값들의 개수 |
| **Unique values concatenated** | `UNIQUE_VALUES` | [string] | 입력 필드에서 발견된 유일 값을 쉼표로 구분한 목록의 문자열 |

### 파이썬 코드

**Algorithm ID**: `native:listuniquevalues`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 평균 좌표 (들)

입력 좌표에 있는 도형 무리 (mass) 의 중심 (center) 을 가진 포인트 레이어를 계산합니다.

무리의 중심을 계산할 때 각 피처에 적용할 가중치를 담고 있는 속성을 지정할 수 있습니다.

파라미터에서 속성을 선택한 경우, 해당 필드에 있는 값에 따라 피처를 그룹화할 것입니다. 전체 레이어의 무리의 중심으로 단일 포인트를 생성하는 대신, 산출 레이어가 각 범주에 있는 피처 무리의 중심을 담게 될 것입니다.

**기본 메뉴**: *Vector → Analysis Tools*

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | 입력 벡터 레이어 |
| **Weight field** 선택적 | `WEIGHT` | [tablefield: numeric] | 가중치를 적용한 평균을 작업하려는 경우 사용할 필드 |
| **Unique ID field** | `UID` | [tablefield: any] | 평균을 계산할 유일 필드 |
| **Mean coordinates** | `OUTPUT` | [vector: point] 기본값: `[Create temporary layer]` | Specify the (point vector) layer for the result. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Mean coordinates** | `OUTPUT` | [vector: point] | 산출되는 포인트 (들) 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:meancoordinates`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 최근접 이웃 분석

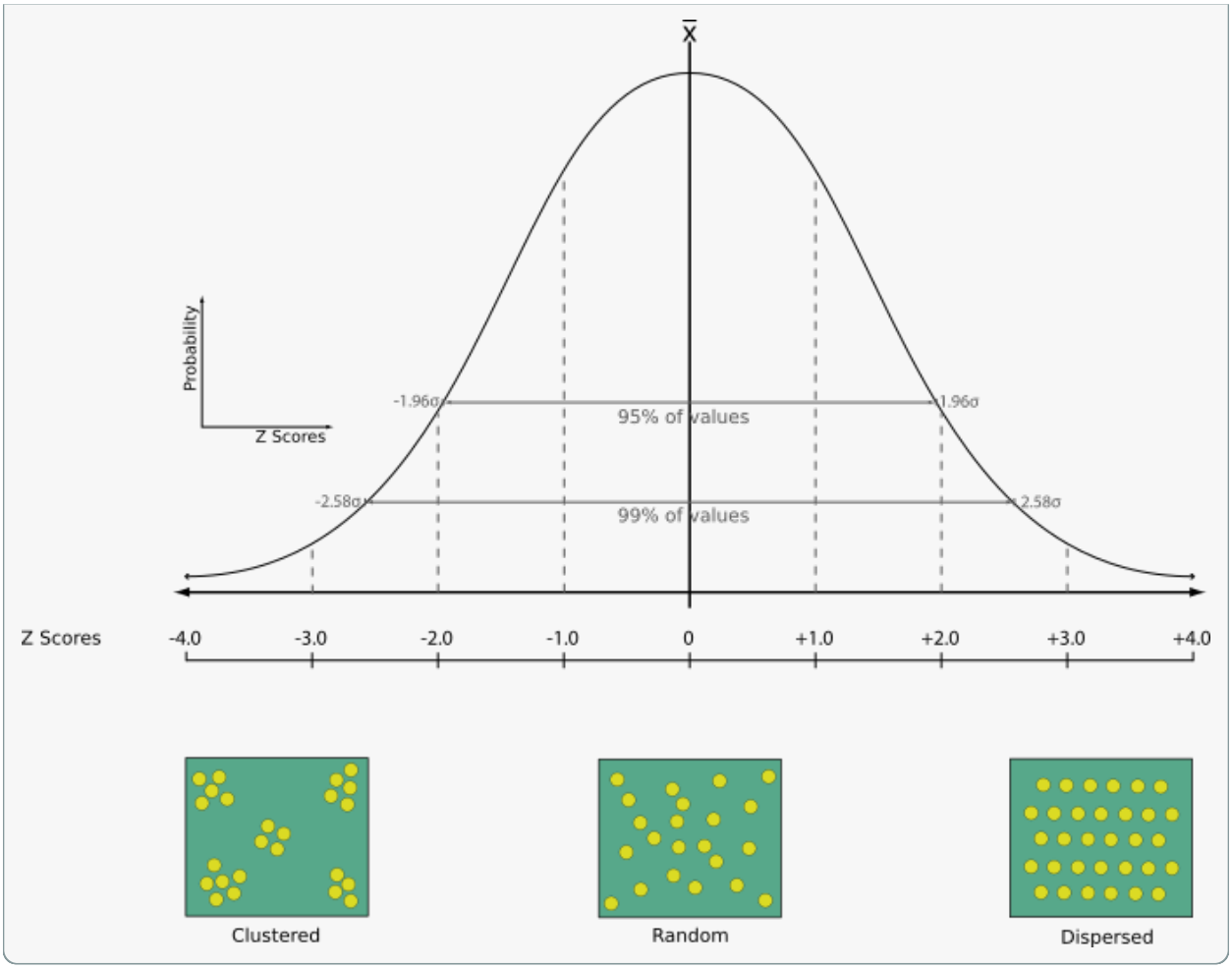
포인트 레이어에 최근접 이웃 분석을 수행합니다. 산출물은 사용자 데이터가 어떻게 분포되었는지 (군집되었는지, 랜덤한지, 또는 분포되었는지) 알려줍니다.

다음과 같은 계산된 통계 값을 가진 산출물을 HTML 파일로 생성합니다:

- 관측 평균 거리
- 기대 평균 거리
- 최근접 이웃 지수
- 포인트의 개수
- Z-점수 (Z-Score): 정규 분포와 Z-점수를 비교하면 사용자 데이터가 어떻게 분포되었는지 알 수 있습니다. 낮은 Z-점수는 데이터가 공간적으로 랜덤한 처리 과정의 결과일 가능성이 낮다는 의미이고, 높은 Z-점수는 사용자 데이터가 공간적으로 랜덤한 처리 과정의 결과일 가능성이 높다는 의미입니다.

**참고**

This algorithm uses ellipsoid based measurements and respects the current *ellipsoid settings*.




기본 메뉴: *Vector → Analysis Tools*

**더 보기**

최근접으로 속성 결합하기

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: point] | 통계를 계산할 포인트 벡터 레이어 |
| **Nearest neighbour** 선택적 | `OUTPUT_HTML_FILE` | [html] 기본값: `[Save to temporary file]` | Specification of the HTML file for the computed statistics. *One of*:<br>• 산출물 건너뛰기<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Nearest neighbour** | `OUTPUT_HTML_FILE` | [html] | 계산된 통계를 가진 HTML 파일 |
| **Observed mean distance** | `OBSERVED_MD` | [numeric: double] | 관측 평균 거리 |
| **Expected mean distance** | `EXPECTED_MD` | [numeric: double] | 기대 평균 거리 |
| **Nearest neighbour index** | `NN_INDEX` | [numeric: integer] | 최근접 이웃 지수 |
| **Number of points** | `POINT_COUNT` | [numeric: integer] | 포인트의 개수 |
| **Z-Score** | `Z_SCORE` | [numeric: double] | Z 점수 (Z-Score) |

### 파이썬 코드

**알고리즘 ID**: `native:nearestneighbouranalysis`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 중첩 분석

중첩 레이어에서 선택한 피처가 입력 레이어의 피처를 중첩하는 영역의 면적 및 커버 백분율을 계산합니다.

산출물 레이어에 중첩 레이어에서 선택한 각 피처가 입력 피처를 중첩하는 총 면적 및 커버 백분율을 보고하는 새 속성을 추가합니다.

> **참고**
>
> This algorithm uses ellipsoid based measurements and respects the current *ellipsoid settings*.

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: polygon] | 입력 레이어 |
| **Overlay layers** | `LAYERS` | [vector: polygon] [list] | 중첩 레이어 |

다음 페이지에 계속

표 24.475 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Overlap** | OUTPUT | [same as input]<br>기본값: [Create temporary layer] | Specify the output vector layer. *One of*:<br>• 임시 레이어 생성 (TEMPORARY_OUTPUT)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Grid size**<br>선택적 | GRID_SIZE | [numeric: double]<br>기본값: 설정하지 않음 | If provided, the input geometries are snapped to a grid of the given size, and the result vertices are computed on that same grid. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Overlap** | OUTPUT | [same as input] | 중첩 레이어에서 선택한 각 피처가 입력 피처를 중첩하는 (맵 단위) 총 면적 및 커버 백분율을 보고하는 새 속성을 추가한 산출 레이어 |

### 파이썬 코드

**알고리즘 ID**: native:calculatevectoroverlaps

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 피처 사이의 최단 라인

소스 레이어와 대상 레이어 사이의 최단 라인을 가진 라인 레이어를 생성합니다. 기본적으로 대상 레이어의 첫 번째 최근접 피처만 연산에 넣습니다. N 번째 최근접 이웃 피처의 번호를 지정할 수 있습니다. 최장 거리를 지정한 경우, 최장 거리 미만에 들어오는 피처만 연산에 넣을 것입니다.

산출 피처는 소스 레이어의 모든 속성, N 번째 최근접 피처의 모든 속성, 그리고 추가적인 거리 필드를 담게 될 것입니다.

**중요**

이 알고리즘은 거리에 대해 순전히 데카르트 평면 계산만을 사용합니다. 피처의 근접성을 결정할 때 측지(geodetic) 또는 타원체 속성을 연산에 넣지 않습니다. 소스 레이어의 좌표계를 기반으로 측정값 및 산출 좌표계를 계산합니다.

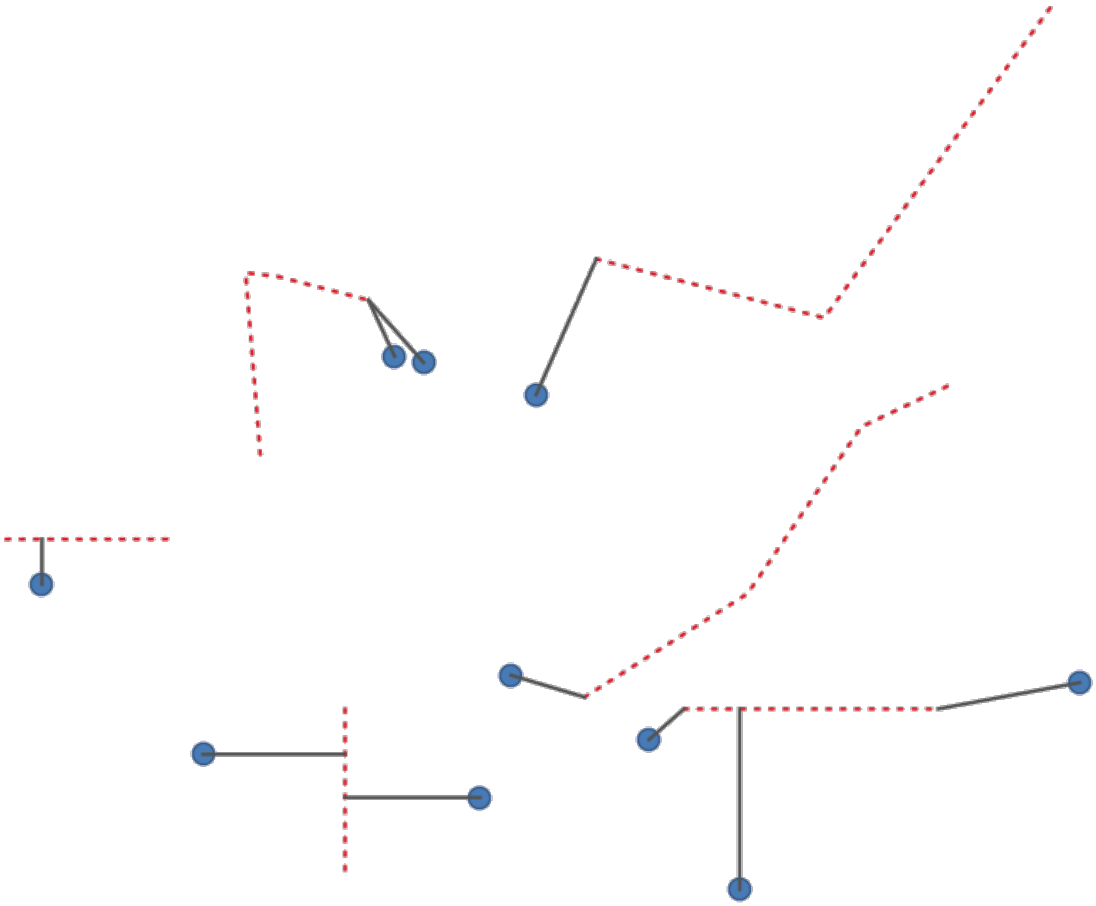


그림 24.74: 포인트 피처에서 라인까지의 최단 라인

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Source layer** | `SOURCE` | [vector: geometry] | 최근접 이웃을 검색할 기원 (origin) 레이어 |
| **Destination layer** | `DESTINATION` | [vector: geometry] | 최근접 이웃을 검색할 대상 (target) 레이어 |
| **Method** | `METHOD` | [enumeration]<br>기본값: 0 | 최단 거리 계산법을 다음 가운데 하나로 지정할 수 있습니다:<br>• 0 — 피처 상에 있는 최근접 포인트까지의 거리<br>• 1 — 피처 중심 (centroid) 까지의 거리 |
| **Maximum number of neighbors** | `NEIGHBORS` | [numeric: integer]<br>기본값: 1 | 검색할 이웃의 최대 개수 |
| **Maximum distance** 선택적 | `DISTANCE` | [numeric: double] | 이 거리보다 가까이 있는 대상 피처만 연산에 넣을 것입니다. |
| **Shortest lines** | `OUTPUT` | [vector: line]<br>기본값: `[Create temporary layer]` | Specify the output vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output layer** | `OUTPUT` | [vector: line] | 대상 레이어의 대응하는 최근접 이웃 (들) 에 소스 피처를 결합시킨 라인 벡터 레이어입니다. 소스 피처와 대상 피처의 모든 속성과 계산한 거리를 담고 있습니다. |

### 파이썬 코드

**알고리즘 ID**: `native:shortestline`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### ST-DBSCAN 군집 형성

이상값 (noise) (ST-DBSCAN) 알고리즘을 가진 응용 프로그램의 시공간 밀도 기반 군집 형성의 2 차원 구현을 기반으로 포인트 피처를 군집시킵니다.

**더 보기**

*DBSCAN 군집 형성*, *k-평균 군집 형성*

#### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: point] | 분석할 레이어 |
| **Date/time field** | `DATE-TIME_FIELD` | [tablefield: date] | 시계열 정보를 담고 있는 필드 |
| **Minimum cluster size** | `MIN_SIZE` | [numeric: integer]<br>기본값: 5 | 군집을 형성할 피처의 최소 개수 |
| **Maximum distance between clustered points** | `EPS` | [numeric: double]<br>기본값: 1.0 | 이 거리 (eps) 를 초과하면 동일 군집에 속할 수 없습니다. |
| **Maximum time duration between clustered points** | `EPS2` | [numeric: double]<br>기본값: 0.0 (days) | 이 기간을 초과하면 피처 2 개가 동일 군집 (eps2) 에 속할 수 없습니다. 밀리초, 초, 분, 시, 일 및 주 (☐) 단위를 사용할 수 있습니다. |
| **Clusters** | `OUTPUT` | [vector: point]<br>기본값: `[Create temporary layer]` | Specify the vector layer for the result of the clustering. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

#### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Treat border points as noise (DBSCAN*)** | `DBSCAN*` | [boolean]<br>기본값: False | `True` 로 설정하면, 군집의 경계 상에 있는 포인트를 군집되지 않은 포인트로 취급하고, 군집 내부에 있는 포인트만 군집된 것으로 태그합니다. |
| **Cluster field name** | `FIELD_NAME` | [string]<br>기본값: 'CLUSTER_ID' | 연관 군집 번호를 저장할 필드의 이름 |

다음 페이지에 계속

표 24.481 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Cluster size field name** | SIZE_FIELD_NAME | [string] 기본값: 'CLUSTER_SIZE' | 동일 군집에 있는 피처들의 개수를 가진 필드의 이름 |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Clusters** | OUTPUT | [vector: point] | 원본 피처와 함께 원본 피처가 속한 군집을 설정하는 필드를 가진 벡터 레이어 |
| **Number of clusters** | NUM_CLUSTERS | [numeric: integer] | 발견된 군집의 개수 |

### 파이썬 코드

**알고리즘 ID**: `native:stdbscanclustering`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 범주 통계

필드의 통계를 부모 범주 (parent class) 에 따라 계산합니다. 부모 범주란 다른 필드들에서 나온 값을 조합한 것입니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input vector layer** | INPUT | [vector: any] | 유일 범주 및 값을 가진 입력 벡터 레이어 |
| **Field to calculate statistics on (if empty, only count is calculated)** 선택적 | VALUES_FIELD_NAME | [tablefield: any] | 이 파라미터가 비어 있는 경우 개수만 계산할 것입니다. |
| **Field(s) with categories** | CATEGORIES_FIELD_NA | [tablefield: any] [list] | (결합되어) 범주를 정의하는 필드들 |

다음 페이지에 계속

표 24.483 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Statistics by category** | OUTPUT | [vector: table]<br>기본값: [Create temporary layer] | Specify the output table for the generated statistics. *One of*:<br>• 임시 레이어 생성 (TEMPORARY_OUTPUT)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Statistics by category** | OUTPUT | [vector: table] | 통계를 담고 있는 테이블 |

분석되는 필드의 유형에 따라, 각 그룹화된 값에 대해 다음 통계를 반환합니다:

| 통계 정보 | 문자열 | Numeric | 날짜 |
|---|---|---|---|
| 개수 (COUNT) | ✔ | ✔ | ✔ |
| 유일 값 (UNIQUE) | ✔ | | ✔ |
| 비어 있는 (NULL) 값 (EMPTY) | ✔ | | ✔ |
| 비지 않은 값 (FILLED) | ✔ | | ✔ |
| 최소값 (MIN) | ✔ | ✔ | ✔ |
| 최대값 (MAX) | ✔ | ✔ | ✔ |
| 범위 (RANGE) | | ✔ | |
| 합계 (SUM) | | ✔ | |
| 평균값 (MEAN) | | ✔ | |
| 중간값 (MEDIAN) | | ✔ | |
| 표준 편차 (STD_DEV) | | ✔ | |
| 변동 계수 (coefficient of variation — CV) | | ✔ | |
| 소수 값 (가장 드물게 나오는 값 — MINORITY) | | ✔ | |
| 다수 값 (가장 자주 나오는 값 — MAJORITY) | | ✔ | |
| 제 1 사분위수 (FIRSTQUARTILE) | | ✔ | |
| 제 3 사분위수 (THIRDQUARTILE) | | ✔ | |
| 사분위수 범위 (제 1 사분위수와 제 3 사분위수 사이의 범위 — IQR) | | ✔ | |
| 최단 길이 (MIN_LENGTH) | ✔ | | |
| 평균 길이 (MEAN_LENGTH) | ✔ | | |

다음 페이지에 계속

표 24.485 – 이전 페이지에서 계속

| 통계 정보 | 문자열 | Numeric | 날짜 |
|---|---|---|---|
| 최장 길이 (MAX_LENGTH) | ☑ | | |

### 파이썬 코드

**알고리즘 ID**: `qgis:statisticsbycategories`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 라인 길이 합계

폴리곤 레이어와 라인 레이어를 받아 각 폴리곤과 교차하는 라인 길이의 총합과 총 개수를 측정합니다.

입력 폴리곤 레이어와 동일한 피처를 가지면서, 각 폴리곤과 교차하는 라인의 길이와 개수를 담고 있는 새로운 두 속성을 추가한 레이어를 산출합니다.

**참고**

This algorithm uses ellipsoid based measurements and respects the current *ellipsoid settings*.

☑ 폴리곤 피처를제자리 피처 수정 할 수 있습니다.

**기본 메뉴**: *Vector → Analysis Tools*

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Lines** | LINES | [vector: line] | 입력 라인 벡터 레이어 |
| **Polygons** | POLYGONS | [vector: polygon] | 폴리곤 벡터 레이어 |
| **Lines length field name** | LEN_FIELD | [string]<br>기본값: 'LENGTH' | 라인 길이를 저장할 필드의 이름 |
| **Lines count field name** | COUNT_FIELD | [string]<br>기본값: 'COUNT' | 라인 개수를 저장할 필드의 이름 |

다음 페이지에 계속

표 24.486 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Line length** | `OUTPUT` | [vector: polygon] 기본값: `[Create temporary layer]` | Specify the output polygon layer with generated statistics. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Line length** | `OUTPUT` | [vector: polygon] | 라인 길이 및 개수 필드를 가진 산출 폴리곤 레이어 |

파이썬 코드

**알고리즘 ID**: `native:sumlinelengths`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 24.1.23 Vector coverage

#### Dissolve coverage

Operates on a coverage (represented as a set of polygon features with exactly matching edge geometry) to dissolve (union) the geometries. It provides a heavily optimized approach for unioning these features compared with the standard Dissolve tools.

**더 보기**

융해

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: polygon] | 입력 폴리곤 벡터 레이어 |
| **Dissolved** | `OUTPUT` | [vector: polygon]<br>기본값: `[Create temporary layer]` | Specify the output vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Dissolved** | `OUTPUT` | [vector: polygon] | The output polygon vector layer with dissolved geometries. |

### 파이썬 코드

**Algorithm ID**: `native:coverageunion`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## Simplify coverage

Operates on a coverage (represented as a set of polygon features with exactly matching edge geometry) to apply a Visvalingam−Whyatt simplification to the edges, reducing complexity in proportion with the provided tolerance, while retaining a valid coverage (i.e. no edges will cross or touch after the simplification). Geometries will never be removed, but they may be simplified down to just a triangle. Also, some geometries (such as polygons which have too few non-repeated points) will be returned unchanged.

If the input dataset is not a valid coverage due to overlaps, it will still be simplified, but invalid topology such as crossing edges will still be invalid.

> ⚠ 주의
>
> Running this algorithm requires QGIS installed with GEOS >= 3.12 (see *Help* → *About* menu).

더 보기

단순화, 평탄화

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: polygon] | 입력 폴리곤 벡터 레이어 |
| **Tolerance** | `TOLERANCE` | [numeric: double]<br>기본값: 0.0 | The maximum distance (in unit of choice) between two consecutive vertices to get merged. |
| **Preserve boundary** | `PRESERVE_BOUNDARY` | [boolean]<br>기본값: False | When enabled, the outside edges of the coverage will be preserved without simplification |
| **Simplified** | `OUTPUT` | [vector: polygon]<br>기본값: `[Create temporary layer]` | Specify the output vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Simplified** | `OUTPUT` | [vector: polygon] | The output polygon vector layer, with a lower number of vertices. |

### 파이썬 코드

**Algorithm ID**: `native:coveragesimplify`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### Validate coverage

Analyzes a coverage (represented as a set of polygon features with exactly matching edge geometry) to find places where the assumption of exactly matching edges is not met. Invalidity includes polygons that overlap or that have gaps smaller than the specified gap width.

**주의**

Running this algorithm requires QGIS installed with GEOS >= 3.12 (see *Help* → *About* menu).

**더 보기**

무결성 점검하기, *Small gaps*, *Overlaps*

#### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: polygon] | 입력 폴리곤 벡터 레이어 |
| **Gap width** | `GAP_WIDTH` | [numeric: double]<br>기본값: 0.0 | The maximum width of gap to detect |
| **Invalid edges** | `INVALID_EDGES` | [vector: line]<br>기본값: `[Create temporary layer]` | Specify the output vector layer. *One of*:<br>• 산출물 건너뛰기<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

#### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Invalid edges** | `OUTPUT` | [vector: line] | The output line vector layer showing error edges for each invalid polygon. |
| **Validity result** | `IS_VALID` | [boolean] | Returns whether the coverage is valid or not. |

파이썬 코드

**Algorithm ID**: `native:coveragevalidate`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 24.1.24 벡터 생성

#### 오프셋 (평행) 라인 배열

레이어에 있는 라인 피처의 오프셋 버전을 여러 개 생성해서 각 피처의 복사본을 생성합니다. 새 복사본마다 지정한 거리만큼 오프셋이 증가합니다.

양의 거리는 라인을 왼쪽으로 오프셋시키고, 음의 거리는 오른쪽으로 오프셋시킬 것입니다.

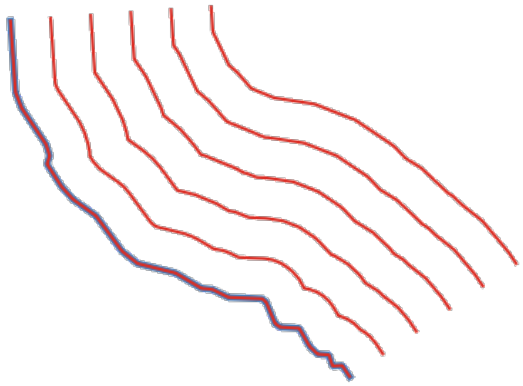


그림 24.75: 소스 피처는 파란색, 오프셋 피처는 빨간색

라인 피처를제자리 피처 수정 할 수 있습니다.

**경고**

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

**더 보기**

라인 오프셋시키기, 이동시킨 피처의 배열

### 파라미터

#### 기본 파라미터

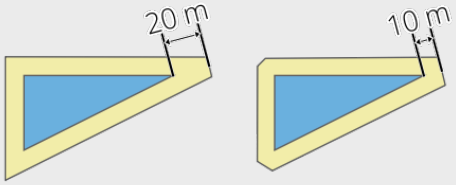
| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: line] | 오프셋에 사용할 입력 라인 벡터 레이어 |
| **Number of features to create** | `COUNT` | [numeric: integer] 기본값: 10 | 각 피처에 대해 생성할 오프셋 복사본의 개수 |
| **Offset step distance** | `OFFSET` | [numeric: double] 기본값: 1.0 | 연속된 두 오프셋 복사본 사이의 거리 |
| **Offset lines** | `OUTPUT` | [vector: line] 기본값: `[Create temporary layer]` | Specify the output line layer with offset features. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

#### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Segments** | `SEGMENTS` | [numeric: double] 기본값: 8 | 둥글린 (rounded) 오프셋을 생성하는 경우 사분원에 가깝게 만드는 데 사용할 라인 선분의 개수 |
| **Join style** | `JOIN_STYLE` | [enumeration] 기본값: 0 | 라인에 있는 모서리를 오프셋시킬 때 둥글림, 마이터, 또는 베벨 결합을 사용할지 여부를 지정합니다. 다음 가운데 하나를 선택할 수 있습니다:<br>• 0 —둥글림 (Round)<br>• 1 —마이터 (Miter)<br>• 2 —베벨 (Bevel)<br>  <br>그림 24.76: 둥글림, 마이터, 베벨 결합 스타일 |

다음 페이지에 계속

표 24.495 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Miter limit** | `MITER_LIMIT` | [numeric: double]<br>기본값: 2.0 | 마이터 결합 스타일에만 적용할 수 있으며, 마이터 결합 생성시 사용할 오프셋 도형으로부터의 최장 거리를 오프셋 거리의 인자로 설정합니다. (최소값은 1.0 입니다.)<br>  <br>그림 24.77: 제한값이 2 인 10m 버퍼와 제한값이 1 인 10m 버퍼 |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Offset lines** | `OUTPUT` | [vector: line] | Output line layer with the original and offset features. Attributes will be copied to the corresponding outputs. The following attributes related to the input parameters will be added: `instance` indexing the `COUNT` offset lines and `offset` designating the offset distance from the original line. |

### 파이썬 코드

**알고리즘 ID**: `native:arrayoffsetlines`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 이동시킨 피처의 배열

레이어에 있는 피처의 이동 (translated) 버전을 여러 개 생성해서 각 피처의 복사본을 생성합니다. 새 복사본마다 X, Y, 그리고/또는 Z 축에 사전 설정한 거리만큼 변위가 증가합니다.

도형에 있는 M 값도 이동시킬 수 있습니다.

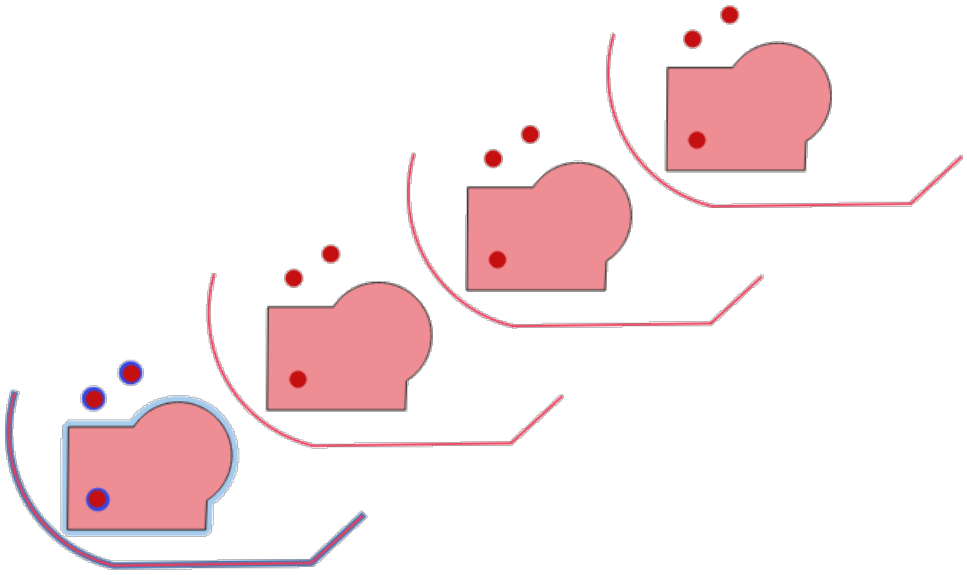


그림 24.78: 입력 레이어는 파란색, 이동된 피처를 담은 산출 레이어는 빨간색

포인트, 라인 및 폴리곤 피처를제자리 피처 수정 할 수 있습니다.

**경고**

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

**더 보기**

이동시키기 *(Translate)*, 오프셋 *(*평행*)* 라인 배열

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | INPUT | [vector: geometry] | 이동시킬 입력 벡터 레이어 |
| **Number of features to create** | COUNT | [numeric: integer]<br>기본값: 10 | 각 피처에 대해 생성할 복사본의 개수 |

다음 페이지에 계속

표 24.497 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Step distance (x-axis)** | `DELTA_X` | [numeric: double]<br>기본값: 0.0 | X 축에 적용할 변위 (displacement) |
| **Step distance (y-axis)** | `DELTA_Y` | [numeric: double]<br>기본값: 0.0 | Y 축에 적용할 변위 (displacement) |
| **Step distance (z-axis)** | `DELTA_Z` | [numeric: double]<br>기본값: 0.0 | Z 축에 적용할 변위 (displacement) |
| **Step distance (m values)** | `DELTA_M` | [numeric: double]<br>기본값: 0.0 | M 축에 적용할 변위 (displacement) |
| **Translated** | `OUTPUT` | [same as input]<br>기본값: `[Create temporary layer]` | Output vector layer with translated (moved) copies of the features. The original features are also copied. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Translated** | `OUTPUT` | [same as input] | 이동된 피처 복사본을 담은 산출 벡터 레이어입니다. 원본 피처도 복사됩니다. |

### 파이썬 코드

**알고리즘 ID**: `native:arraytranslatedfeatures`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 그리드 생성하기

지정한 범위를 커버하는 그리드를 가진 벡터 레이어를 생성합니다. 그리드 셀이 서로 다른 형태일 수 있습니다:

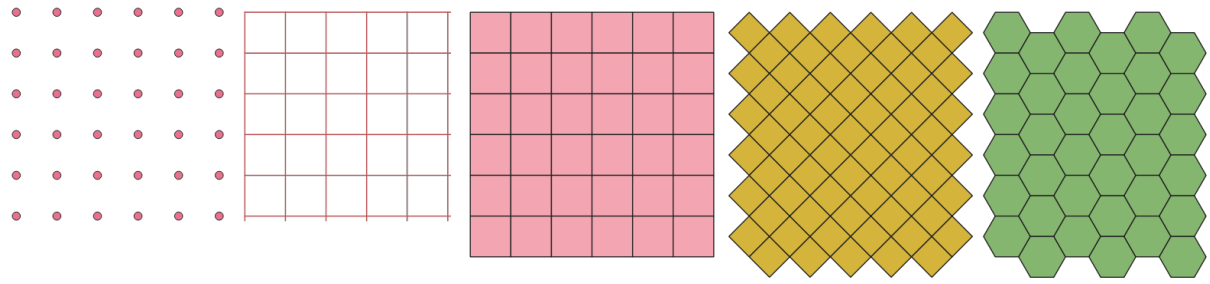


그림 24.79: 동일한 범위에 중첩 없이 적용된 서로 다른 그리드 셀 형태들

그리드에 있는 각 요소의 크기 그리고/또는 배치를 수평 및 수직 간격을 사용해서 정의합니다. 산출 레이어의 좌표계는 반드시 정의해야만 합니다. 그리드 범위와 간격 값은 반드시 이 좌표계의 좌표와 단위를 통해 표현해야만 합니다.

**기본 메뉴**: *Vector → Research Tools*

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Grid type** | `TYPE` | [enumeration]<br>기본값: 0 | 그리드의 형태를 지정합니다. 다음 가운데 하나를 선택할 수 있습니다:<br>• 0 — 포인트<br>• 1 — 라인<br>• 2 — 직사각형 (폴리곤)<br>• 3 — 마름모꼴 (폴리곤)<br>• 4 — 육각형 (폴리곤) |
| **Grid extent** | `EXTENT` | [extent] | 그리드의 범위<br>다음과 같은 메소드를 사용할 수 있습니다:<br>• 레이어로부터 계산…: 현재 프로젝트에 불러온 레이어의 범위를 사용합니다.<br>• 레이어 맵으로부터 계산…: 활성 프로젝트의조판 맵 항목 범위를 사용합니다.<br>• 북마크로부터 계산…: 저장되어 있는북마크 의 범위를 사용합니다.<br>• 맵 캔버스 범위 사용<br>• 캔버스에 그리기: 연산에 넣을 범위를 정하는 직사각형을 클릭 & 드래그합니다.<br>• 좌표를 `xmin, xmax, ymin, ymax` 로 입력 |
| **Horizontal spacing** | `HSPACING` | [numeric: double]<br>기본값: 1.0 | 그리드 셀의 X 축 크기 |

다음 페이지에 계속

표 24.499 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Vertical spacing** | `VSPACING` | [numeric: double]<br>기본값: 1.0 | 그리드 셀의 Y 축 크기 |
| **Horizontal overlay** | `HOVERLAY` | [numeric: double]<br>기본값: 0.0 | 연속된 두 그리드 셀 사이의 X 축 중첩 거리 |
| **Vertical overlay** | `VOVERLAY` | [numeric: double]<br>기본값: 0.0 | 연속된 두 그리드 셀 사이의 Y 축 중첩 거리 |
| **Grid CRS** | `CRS` | [crs]<br>기본값: *Project CRS* | 그리드에 적용할 좌표계 |
| **Grid** | `OUTPUT` | [vector: geometry]<br>기본값: `[Create temporary layer]` | Resulting vector grid layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Grid** | `OUTPUT` | [vector: geometry] | 산출되는 벡터 그리드 레이어입니다. 산출 도형 유형 (포인트, 라인, 또는 폴리곤)은 *Grid type* 에 따라 달라집니다. 피처는 위에서 아래로, 왼쪽에서 오른쪽으로 생성됩니다. 속성 테이블은 다음으로 채워집니다:<br>• `id`<br>• `left`, `right`, `top` 및 `bottom` 가장자리의 좌표<br>• 그리드에서의 배치: `row_index` 및 `column_index` (포인트, 직사각형 및 육각형 그리드 유형에 사용 가능) |

### 파이썬 코드

**알고리즘 ID**: `native:creategrid`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### Create layer from extent

Creates a new vector layer that contains a single feature with geometry matching the extent of the input layer.

It can be used in models to convert a literal extent (`xmin`, `xmax`, `ymin`, `ymax` format) into a layer which can be used for other algorithms which require a layer based input.

**더 보기**

*Create layer from point*

#### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Extent (xmin, xmax, ymin, ymax)** | `INPUT` | [extent] | Input extent<br>다음과 같은 메소드를 사용할 수 있습니다:<br>• 레이어로부터 계산…: 현재 프로젝트에 불러온 레이어의 범위를 사용합니다.<br>• 레이어 맵으로부터 계산…: 활성 프로젝트의조판 맵 항목 범위를 사용합니다.<br>• 북마크로부터 계산…: 저장되어 있는북마크 의 범위를 사용합니다.<br>• 맵 캔버스 범위 사용<br>• 캔버스에 그리기: 연산에 넣을 범위를 정하는 직사각형을 클릭 & 드래그합니다.<br>• 좌표를 `xmin`, `xmax`, `ymin`, `ymax` 로 입력 |
| **Extent** | `OUTPUT` | [vector: polygon]<br>기본값: `[Create temporary layer]` | Specify the output vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Extent** | `OUTPUT` | [vector: polygon] | The output (extent) vector layer |

### 파이썬 코드

**Algorithm ID**: `native:extenttolayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## Create layer from point

Creates a new vector layer that contains a single feature with geometry matching a point parameter. It can be used in models to convert a point into a point layer for algorithms which require a layer based input.

**더 보기**

*Create layer from extent*

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Point** | `INPUT` | [coordinates] | Input point, including CRS info (example: `397254,6214446 [EPSG:32632]`). If the CRS is not provided, the Project CRS will be used.<br>The point can be specified by clicking on the map canvas. |
| **Point** | `OUTPUT` | [vector: point]<br>기본값: `[Create temporary layer]` | Specify the output layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Point** | `OUTPUT` | [vector: point] | The output point vector layer containing the input point. |

### 파이썬 코드

**Algorithm ID**: `native:pointtolayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 테이블에서 포인트 레이어 생성하기

좌표 필드를 담고 있는 테이블로부터 포인트 레이어를 생성합니다.

X 및 Y 좌표 이외에 Z 및 M 필드도 지정할 수 있습니다.

**경고**

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: any] | 입력 벡터 레이어 또는 테이블 |
| **X field** | `XFIELD` | [tablefield: any] | X 좌표를 담고 있는 필드 |
| **Y field** | `YFIELD` | [tablefield: any] | Y 좌표를 담고 있는 필드 |
| **Z field**<br>선택적 | `ZFIELD` | [tablefield: any] | Z 좌표를 담고 있는 필드 |
| **M field**<br>선택적 | `MFIELD` | [tablefield: any] | M 값을 담고 있는 필드 |
| **Target CRS** | `TARGET_CRS` | [crs]<br>기본값:<br>`EPSG:4326` | 레이어에 사용할 좌표계입니다. 제공되는 좌표계가 규칙을 따른다고 가정합니다. |

다음 페이지에 계속

표 24.505 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Points from table** | `OUTPUT` | [vector: point]<br>기본값: `[Create temporary layer]` | Specify the resulting point layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Points from table** | `OUTPUT` | [vector: point] | 산출되는 포인트 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:createpointslayerfromtable`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 라인을 따라 포인트 (픽셀의 중심) 생성하기

입력 래스터 및 라인 레이어로부터 포인트 벡터 레이어를 생성합니다.

여기서 포인트는 라인 레이어와 교차 (intersect) 하는 픽셀의 중심 (centroid) 에 대응합니다.

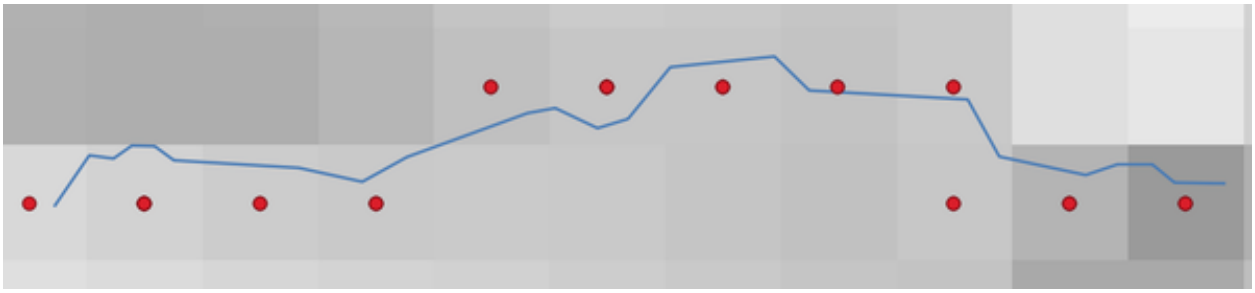


그림 24.80: 픽셀 중심 포인트

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Raster layer** | `INPUT_RASTER` | [raster] | 입력 래스터 레이어 |
| **Vector layer** | `INPUT_VECTOR` | [vector: line] | 입력 라인 벡터 레이어 |
| **Points along line** | `OUTPUT` | [vector: point]<br>기본값: `[Create temporary layer]` | Resulting point layer with pixel centroids. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Points along line** | `OUTPUT` | [vector: point] | 픽셀 중심을 담은 산출 포인트 레이어 |

### 파이썬 코드

**알고리즘 ID**: `qgis:generatepointspixelcentroidsalongline`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 폴리곤 내부에 포인트 (픽셀의 중심) 생성하기

입력 래스터 및 폴리곤 레이어로부터 포인트 벡터 레이어를 생성합니다.

여기서 포인트는 폴리곤 레이어와 교차 (intersect) 하는 픽셀의 중심 (centroid) 에 대응합니다.

**경고**

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

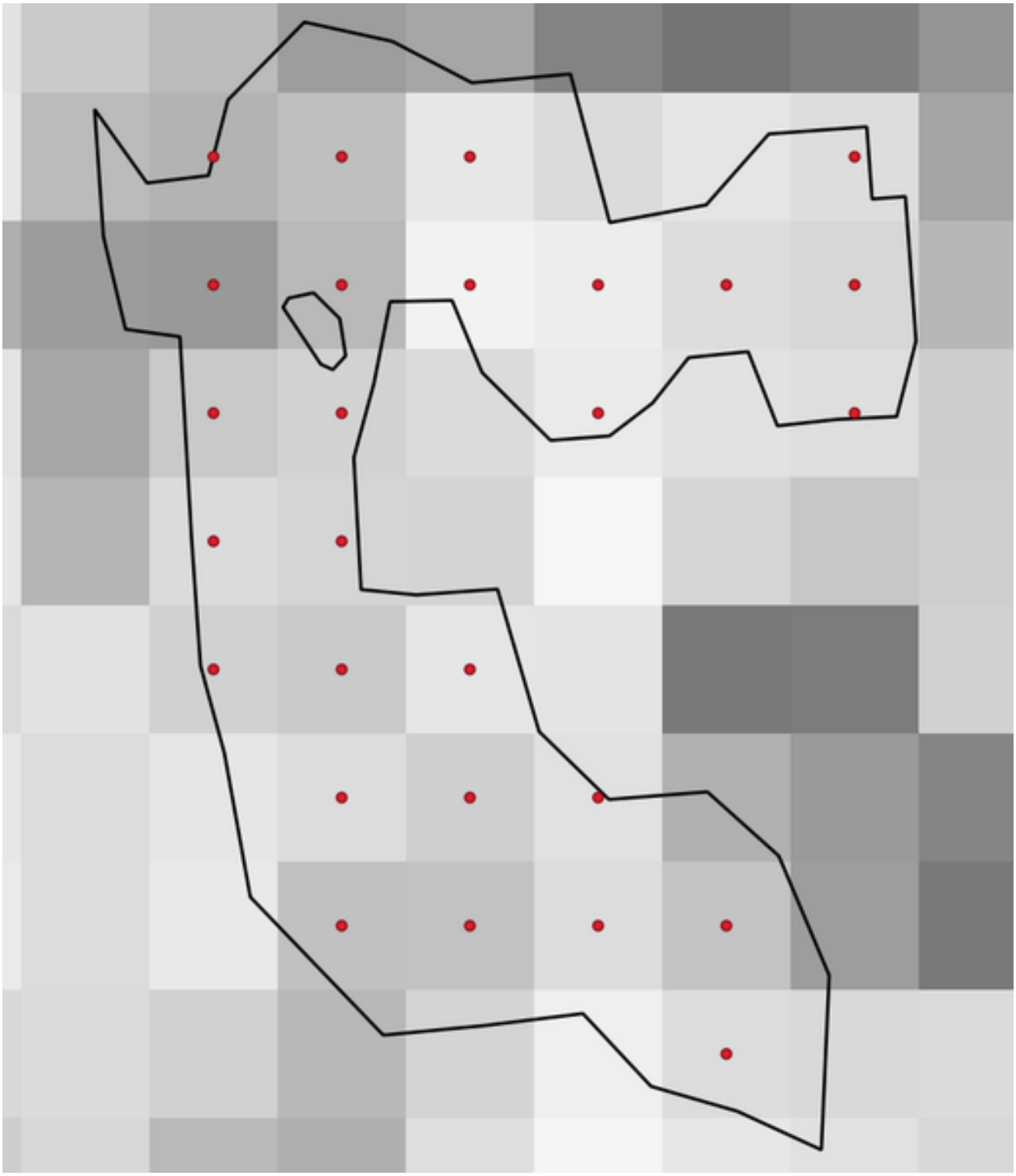


그림 24.81: 픽셀 중심 포인트

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Raster layer** | `INPUT_RASTER` | [raster] | 입력 래스터 레이어 |
| **Vector layer** | `INPUT_VECTOR` | [vector: polygon] | 입력 폴리곤 벡터 레이어 |
| **Points inside polygons** | `OUTPUT` | [vector: point]<br>기본값: `[Create temporary layer]` | Resulting point layer of pixel centroids. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Points inside polygons** | `OUTPUT` | [vector: point] | 픽셀 중심을 담은 산출 포인트 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:generatepointspixelcentroidsinsidepolygons`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 지오태그된 사진 가져오기

소스 폴더에 있는 JPEG 또는 HEIC/HEIF 이미지에서 나온 지오태그 (geotag) 된 위치에 대응하는 포인트 레이어를 생성합니다.

이 포인트 레이어는 입력 파일별로 지오태그를 읽어올 수 있는 단일 PointZ 피처를 담게 될 것입니다. 지오태그에서 읽어온 모든 고도 정보는 포인트의 Z 값을 설정하는 데 쓰일 것입니다.

경도 및 위도 이외에, 사진에 해당 정보가 있다면 고도, 방향 및 타임스탬프 정보도 포인트 속성으로 추가될 것입니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input folder** | FOLDER | [folder] | 지오태그된 사진을 담고 있는 소스 폴더를 가리키는 경로 |
| **Scan recursively** | RECURSIVE | [boolean]<br>기본값: False | 이 옵션을 활성화하면, 폴더는 물론 그 하위 폴더도 스캔할 것입니다. |
| **Photos**<br>선택적 | OUTPUT | [vector: point]<br>기본값: [Create temporary layer] | Specify the point vector layer for the geotagged photos. *One of*:<br>• 산출물 건너뛰기<br>• 임시 레이어 생성 (TEMPORARY_OUTPUT)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |
| **Invalid photos table**<br>선택적 | INVALID | [vector: table]<br>기본값: [Skip output] | Specify the table of unreadable or non-geotagged photos. *One of*:<br>• 산출물 건너뛰기<br>• 임시 레이어 생성 (TEMPORARY_OUTPUT)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Photos** | OUTPUT | [vector: point] | 지오태그된 사진에 대한 포인트 벡터 레이어입니다. 이 레이어는 경로 및 사진 미리보기 설정이 자동으로 지정된 상태입니다. |
| **Invalid photos table**<br>선택적 | INVALID | [vector: table] | 읽을 수 없거나 지오태그되지 않은 사진의 테이블도 생성할 수 있습니다. |

### 파이썬 코드

**알고리즘 ID**: `native:importphotos`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 포인트를 경로로

입력 포인트 레이어의 표현식 또는 필드가 정의하는 순서에 따라 포인트를 결합해서 포인트 레이어를 라인 레이어로 변환합니다.

필드 또는 표현식이 라인 피처를 구분하기 위해 포인트를 그룹화할 수 있습니다.

라인 벡터 레이어는 물론, 산출 라인을 시작점과 일련의 (방위각에 상대적인) 방위/방향 및 거리로 서술하는 텍스트 파일도 산출할 수 있습니다.

**참고**

This algorithm uses ellipsoid based measurements and respects the current *ellipsoid settings*.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input point layer** | `INPUT` | [vector: point] | 입력 포인트 벡터 레이어 |
| **Create closed paths** | `CLOSE_PATH` | [boolean]<br>기본값: False | 이 옵션을 체크하면, 라인의 첫번째 및 마지막 포인트를 연결시켜 생성된 경로를 닫을 것입니다. |
| **Order expression**<br>선택적 | `ORDER_EXPRESSION` | [expression] | 경로에 있는 포인트를 연결할 순서를 제공하는 필드 또는 표현식입니다. 설정하지 않으면 피처 ID(`$id`) 를 사용합니다. |
| **Sort text containing numbers naturally** | `NATURAL_SORT` | [boolean]<br>기본값: False | 체크박스를 체크하면, 지정한 표현식을 기반으로 피처들을 자연스럽게 정렬합니다. (예: 'a9' < 'a10') |
| **Path group expression**<br>선택적 | `GROUP_EXPRESSIC` | [expression] | 필드 또는 표현식에서 같은 값을 가진 포인트 피처들을 동일한 라인에 그룹화할 것입니다. 설정하지 않는 경우 모든 입력 포인트로 단일 경로를 그립니다. |

다음 페이지에 계속

표 24.513 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Paths** | OUTPUT | [vector: line]<br>기본값: [Create temporary layer] | Specify the line vector layer of the path. *One of*:<br>• 임시 레이어 생성 (TEMPORARY_OUTPUT)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |
| **Directory for text output**<br>선택적 | OUTPUT_TEXT_DIR | [folder]<br>기본값: [Skip output] | Specify the directory that will contain the description files of points and paths. *One of*:<br>• 산출물 건너뛰기<br>• 임시 디렉터리에 저장<br>• 디렉터리에 저장 |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Paths** | OUTPUT | [vector: line] | 경로를 담은 라인 벡터 레이어 |
| **Directory for text output** | OUTPUT_TEXT_DIR | [folder] | 포인트와 경로를 서술하는 파일을 담고 있는 디렉터리 |

### 파이썬 코드

**알고리즘 ID**: `native:pointstopath`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 라인을 따라 랜덤 포인트 생성하기

포인트를 또다른 레이어의 라인 위에 배치한 새 포인트 레이어를 생성합니다.

입력 레이어의 각 라인마다, 산출 레이어에 지정한 개수의 포인트를 추가합니다. 포인트를 추가하는 과정은 다음과 같습니다:

1. 입력 레이어에서 라인 피처 하나를 랜덤하게 선택합니다.
2. 다중 부분 피처인 경우, 부분 하나를 랜덤하게 선택합니다.
3. 해당 라인의 선분 하나를 랜덤하게 선택합니다.
4. 해당 선분의 한 위치를 랜덤하게 선택합니다.

이 과정은 즉 라인에서 (상대적으로 짧은 선분으로 이루어진) 만곡된 부분이 (상대적으로 긴 선분으로 이루어진) 직선 부분보다 더 포인트를 받게 된다는 뜻입니다. 다음 그림에서 라인을 따라 랜덤 포인트 생성하기 알고리즘의 산출물이 (라인을 따라 평균적으로 균등한 간격으로 포인트를 생성하는) 라인에 랜덤 포인트 생성하기 알고리즘의 산출물에 비해 더 많은 포인트를 생성한 것을 볼 수 있습니다.

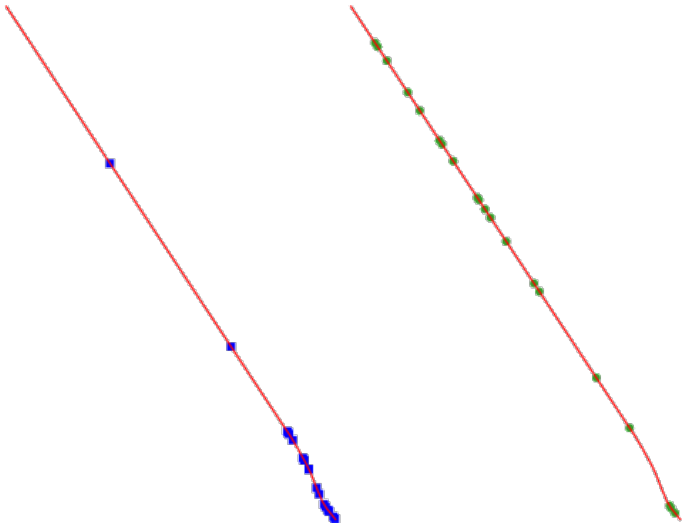


그림 24.82: 알고리즘 산출물의 예. 왼쪽: 라인을 따라 랜덤 포인트 생성하기, 오른쪽: 라인에 랜덤 포인트 생성하기

포인트끼리 너무 가까워지는 일을 피하기 위해 최소 거리를 지정할 수 있습니다.

**경고**

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

**더 보기**

라인에 랜덤 포인트 생성하기

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input point layer** | `INPUT` | [vector: line] | 입력 라인 벡터 레이어 |
| **Number of points** | `POINTS_NUMBER` | [numeric: integer]<br>기본값: 1 | 생성할 포인트의 개수 |
| **Minimum distance between points** | `MIN_DISTANCE` | [numeric: double]<br>기본값: 0.0 | 포인트들 사이의 최소 거리 |

다음 페이지에 계속

표 24.515 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Random points** | `OUTPUT` | [vector: point]<br>기본값: `[Create temporary layer]` | The output random points. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Random points** | `OUTPUT` | [vector: point] | 산출 랜덤 포인트 레이어 |

### 파이썬 코드

**알고리즘 ID**: `qgis:qgisrandompointsalongline`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 범위 안에 랜덤 포인트 생성하기

지정한 범위 내부에 지정한 개수의 랜덤 포인트를 담은 새 포인트 레이어를 생성합니다.

포인트들이 서로 너무 가까워지지 않게 하기 위한 거리 인자를 지정할 수 있습니다. 포인트 사이의 최소 거리가 새 포인트를 생성할 수 없을 정도인 경우, 거리를 줄이거나 최대 시도 횟수를 늘리면 됩니다.

**기본 메뉴**: *Vector ▸ Research Tools*

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input extent** | `EXTENT` | [extent] | 랜덤 포인트용 맵 범위<br>다음과 같은 메소드를 사용할 수 있습니다:<br>• 레이어로부터 계산…: 현재 프로젝트에 불러온 레이어의 범위를 사용합니다.<br>• 레이어 맵으로부터 계산…: 활성 프로젝트의조판 맵 항목 범위를 사용합니다.<br>• 북마크로부터 계산…: 저장되어 있는북마크 의 범위를 사용합니다.<br>• 맵 캔버스 범위 사용<br>• 캔버스에 그리기: 연산에 넣을 범위를 정하는 직사각형을 클릭 & 드래그합니다.<br>• 좌표를 `xmin, xmax, ymin, ymax` 로 입력 |
| **Number of points** | `POINTS_NUMBER` | [numeric: integer]<br>기본값: 1 | 생성할 포인트의 개수 |
| **Minimum distance between points** | `MIN_DISTANCE` | [numeric: double]<br>기본값: 0.0 | 포인트들 사이의 최소 거리 |
| **Target CRS** | `TARGET_CRS` | [crs]<br>기본값: *Project CRS* | 랜덤 포인트 레이어의 좌표계 |
| **Random points** | `OUTPUT` | [vector: point]<br>기본값: `[Create temporary layer]` | The output random points. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Maximum number of search attempts given the minimum distance** | `MAX_ATTEMPTS` | [numeric: integer]<br>기본값: 200 | 포인트를 배치하기 위한 최대 시도 횟수 |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Random points** | `OUTPUT` | [vector: point] | 산출 랜덤 포인트 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:randompointsinextent`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 레이어 경계 안에 랜덤 포인트 생성하기

지정한 레이어의 범위 내부에 지정한 개수의 랜덤 포인트를 담은 새 포인트 레이어를 생성합니다.

포인트끼리 너무 가까워지는 일을 피하기 위해 최소 거리를 지정할 수 있습니다.

**기본 메뉴**: *Vector ▸ Research Tools*

> **⚠ 경고**
>
> This algorithm drops existing primary keys or FID values and regenerates them in output layers.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: polygon] | 영역을 정의하는 입력 폴리곤 레이어 |
| **Number of points** | `POINTS_NUMBER` | [numeric: integer]<br>기본값: 1 | 생성할 포인트의 개수 |
| **Minimum distance between points** | `MIN_DISTANCE` | [numeric: double]<br>기본값: 0.0 | 포인트들 사이의 최소 거리 |
| **Random points** | `OUTPUT` | [vector: point]<br>기본값: `[Create temporary layer]` | The output random points. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Random points** | `OUTPUT` | [vector: point] | 산출 랜덤 포인트 레이어 |

### 파이썬 코드

**알고리즘 ID**: `qgis:randompointsinlayerbounds`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 폴리곤에 랜덤 포인트 생성하기

포인트를 또다른 레이어의 폴리곤 안에 배치한 포인트 레이어를 생성합니다.

입력 레이어의 각 (폴리곤/멀티폴리곤) 피처 도형마다, 산출 레이어에 지정한 개수의 포인트를 추가합니다.

산출 포인트 레이어에서 포인트들이 너무 가까이 배치되는 일을 피하기 위해 피처 당 그리고 전체 수준의 최소 거리를 지정할 수 있습니다. 최소 거리를 지정한 경우, 각 피처마다 지정한 개수의 포인트를 생성하지 못 할 수도 있습니다. 생성된 포인트 및 누락된 포인트의 개수는 알고리즘에서 나온 산출물에서 알 수 있습니다.

다음 그림은 피처 당 그리고 전체 수준의 거리 및 0 또는 0 이 아닌 (zero/non-zero) 최소 거리를 보여줍니다. (동일한 시드로 생성되었기 때문에 적어도 처음 생성된 포인트는 동일할 것입니다.)

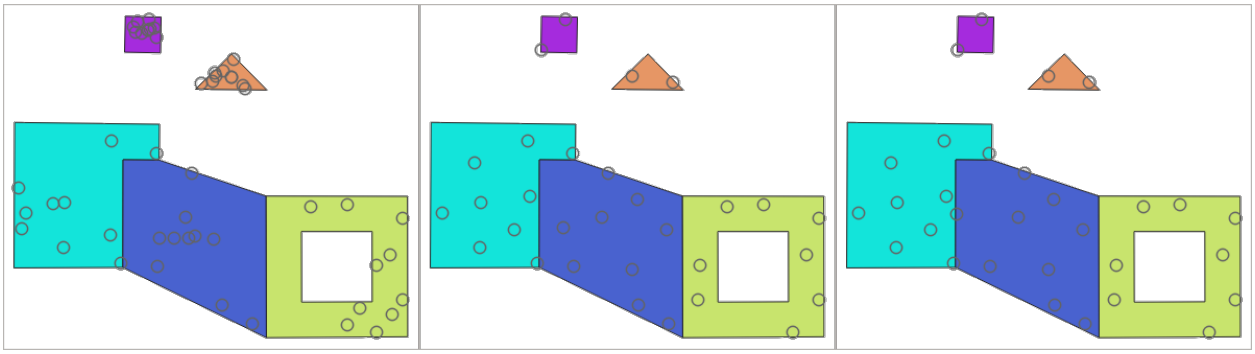


그림 24.83: 폴리곤 피처 당 포인트 10 개, 왼쪽: 최소 거리 = 0, 가운데: 최소 거리 = 1, 오른쪽: 최소 거리 = 1, 전체 수준 최소 거리 = 0

포인트 당 최대 시도 횟수를 지정할 수 있습니다. 이 옵션은 0 이 아닌 최소 거리에 대해서만 지정할 수 있습니다.

난수 발생기 용 시드를 지정할 수 있습니다. 이렇게 하면 알고리즘이 서로 다른 설정으로 실행되어도 동일한 난수 배열을 사용할 수 있습니다.

포인트가 생성된 폴리곤 피처의 속성을 포함시킬 수 있습니다. (*Include polygon attributes*)

모든 피처에 대해 대략 동일한 포인트 밀도를 얻고자 하는 경우, 폴리곤 피처 도형의 면적을 이용해서 포인트 개수를 데이터 정의할 수 있습니다.

더 보기

폴리곤 내부에 랜덤 포인트 생성하기

### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input polygon layer** | INPUT | [vector: line] | 입력 폴리곤 벡터 레이어 |
| **Number of points for each feature** | POINTS_NUMBER | [numeric: integer]<br>기본값: 1 | 생성할 포인트의 개수 |
| **Minimum distance between points**<br>선택적 | MIN_DISTANCE | [numeric: double]<br>기본값: 0.0 | 한 폴리곤 피처 내부의 포인트들 사이의 최소 거리 |
| **Random points in polygons** | OUTPUT | [vector: point]<br>기본값: [Create temporary layer] | The output random points. *One of*:<br>• 임시 레이어 생성 (TEMPORARY_OUTPUT)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

#### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Global minimum distance between points**<br>선택적 | MIN_DISTANCE_GI | [numeric: double]<br>기본값: 0.0 | 포인트들 사이의 전체 수준 최소 거리입니다. *Minimum distance between points (per feature)* 파라미터가 의미가 있으려면 해당 파라미터보다 짧아야 합니다. |
| **Maximum number of search attempts (for Min. dist. > 0)**<br>선택적 | MAX_TRIES_PER_I | [numeric: integer]<br>기본값: 10 | 포인트 당 최대 시도 횟수입니다. 포인트들 사이의 최소 거리가 (0 보다 크게) 설정된 경우에만 의미가 있습니다. |
| **Random seed**<br>선택적 | SEED | [numeric: integer]<br>기본값: 설정하지 않음 | 난수 생성기에 사용할 시드 |
| **Include polygon attributes** | INCLUDE_POLYGON_A | [boolean]<br>기본값: True | 설정한 경우, 포인트가 배치된 폴리곤에서 속성을 가져올 것입니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Random points in polygons** | `OUTPUT` | [vector: point] | 산출 랜덤 포인트 레이어 |
| **Number of features with empty or no geometry** | `FEATURES_WITH_EMPT` | [numeric: integer] | |
| **Total number of points generated** | `OUTPUT_POINTS` | [numeric: integer] | |
| **Number of missed points** | `POINTS_MISSED` | [numeric: integer] | 최소 거리 제약조건 때문에 생성하지 못 한 포인트의 개수 |
| **Number of features with missed points** | `POLYGONS_WITH_MISSE` | [numeric: integer] | 비어 있거나 도형이 없는 피처는 포함하지 않음 |

### 파이썬 코드

**알고리즘 ID**: `native:randompointsinpolygons`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 폴리곤 내부에 랜덤 포인트 생성하기

입력 폴리곤 레이어의 각 폴리곤 내부에 지정한 개수의 랜덤 포인트를 담은 새 포인트 레이어를 생성합니다.

샘플링 전략 2 개를 사용할 수 있습니다:

- 포인트 개수: 각 피처별 포인트의 개수
- 포인트 밀도: 각 피처별 포인트의 밀도

포인트끼리 너무 가까워지는 일을 피하기 위해 최소 거리를 지정할 수 있습니다.

**기본 메뉴**: *Vector → Research Tools*

**경고**

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

**더 보기**

폴리곤에 랜덤 포인트 생성하기

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: polygon] | 입력 폴리곤 벡터 레이어 |
| **Sampling strategy** | `STRATEGY` | [enumeration]<br>기본값: 0 | 사용할 샘플링 전략을 지정합니다. 다음 가운데 하나를 선택할 수 있습니다:<br>• 0 — 포인트 개수: 각 피처별 포인트의 개수<br>• 1 — 포인트 밀도: 각 피처별 포인트의 밀도 |
| **Point count or density** | `VALUE` | [numeric: double]<br>기본값: 1.0 | 선택한 *Sampling strategy* 에 따라 포인트의 개수 또는 밀도를 지정합니다. |
| **Minimum distance between points** | `MIN_DISTANCE` | [numeric: double]<br>기본값: 0.0 | 포인트들 사이의 최소 거리 |
| **Random points** | `OUTPUT` | [vector: point]<br>기본값: `[Create temporary layer]` | The output random points. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Random points** | `OUTPUT` | [vector: point] | 산출 랜덤 포인트 레이어 |

### 파이썬 코드

**알고리즘 ID**: `qgis:randompointsinsidepolygons`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 라인에 랜덤 포인트 생성하기

포인트를 또다른 레이어의 라인 위에 배치한 포인트 레이어를 생성합니다.

입력 레이어의 각 (라인/멀티라인) 피처 도형마다, 산출 레이어에 지정한 개수의 포인트를 추가합니다.

산출 포인트 레이어에서 포인트들이 너무 가까이 배치되는 일을 피하기 위해 피처 당 그리고 전체 수준의 최소 거리를 지정할 수 있습니다. 최소 거리를 지정한 경우, 각 피처마다 지정한 개수의 포인트를 생성하지 못 할 수도 있습니다. 생성된 포인트 및 누락된 포인트의 개수는 알고리즘에서 나온 산출물에서 알 수 있습니다.

다음 그림은 피처 당 그리고 전체 수준의 거리 및 0 또는 0 이 아닌 (zero/non-zero) 최소 거리를 보여줍니다. (동일한 시드로 생성되었기 때문에 적어도 처음 생성된 포인트는 동일할 것입니다.)



그림 24.84: 라인 피처 당 포인트 5 개, 왼쪽: 최소 거리 = 0, 가운데: 최소 거리 ≠ 0, 오른쪽: 최소 거리 ≠ 0, 전체 수준 최소 거리 = 0

포인트 당 최대 시도 횟수를 지정할 수 있습니다. 이 옵션은 0 이 아닌 최소 거리에 대해서만 지정할 수 있습니다.

난수 발생기 용 시드를 지정할 수 있습니다. 이렇게 하면 알고리즘이 서로 다른 설정으로 실행되어도 동일한 난수 배열을 사용할 수 있습니다.

포인트가 생성된 라인 피처의 속성을 포함시킬 수 있습니다. (*Include line attributes*)

모든 피처에 대해 대략 동일한 포인트 밀도를 얻고자 하는 경우, 라인 피처 도형의 길이를 이용해서 포인트 개수를 데이터 정의할 수 있습니다.

**더 보기**

라인을 따라 랜덤 포인트 생성하기

### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input line layer** | INPUT | [vector: line] | 입력 라인 벡터 레이어 |
| **Number of points for each feature** | POINTS_NUMBER | [numeric: integer]<br>기본값: 1 | 생성할 포인트의 개수 |
| **Minimum distance between points (per feature)**<br>선택적 | MIN_DISTANCE | [numeric: double]<br>기본값: 0.0 | 한 라인 피처 상의 포인트들 사이의 최소 거리 |
| **Random points on lines** | OUTPUT | [vector: point]<br>기본값: [Create temporary layer] | The output random points. *One of*:<br>• 임시 레이어 생성 (TEMPORARY_OUTPUT)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

#### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Global minimum distance between points**<br>선택적 | MIN_DISTANCE_GI | [numeric: double]<br>기본값: 0.0 | 포인트들 사이의 전체 수준 최소 거리입니다. *Minimum distance between points (per feature)* 파라미터가 의미가 있으려면 해당 파라미터보다 짧아야 합니다. |
| **Maximum number of search attempts (for Min. dist. > 0)**<br>선택적 | MAX_TRIES_PER_I | [numeric: integer]<br>기본값: 10 | 포인트 당 최대 시도 횟수입니다. 포인트들 사이의 최소 거리가 (0 보다 크게) 설정된 경우에만 의미가 있습니다. |
| **Random seed**<br>선택적 | SEED | [numeric: integer]<br>기본값: 설정하지 않음 | 난수 생성기에 사용할 시드 |
| **Include line attributes** | INCLUDE_LINE_ATTI | [boolean]<br>기본값: True | 설정한 경우, 포인트가 배치된 폴리곤에서 속성을 가져올 것입니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Random points on lines** | OUTPUT | [vector: point] | 산출 랜덤 포인트 레이어 |
| **Number of features with empty or no geometry** | FEATURES_WITH_EMPT | [numeric: integer] | |
| **Number of features with missed points** | LINES_WITH_MISS | [numeric: integer] | 비어 있거나 도형이 없는 피처는 포함하지 않음 |
| **Total number of points generated** | POINTS_GENERATE | [numeric: integer] | |
| **Number of missed points** | POINTS_MISSED | [numeric: integer] | 최소 거리 제약조건 때문에 생성하지 못 한 포인트의 개수 |

### 파이썬 코드

**알고리즘 ID**: `native:randompointsonlines`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 래스터 픽셀을 포인트로

래스터 레이어에 있는 각 픽셀에 대응하는 포인트를 담은 벡터 레이어를 생성합니다.

래스터 레이어에 있는 각 개별 픽셀의 중심에 대해 포인트 피처를 생성해서 래스터 레이어를 벡터 레이어로 변환합니다. 산출물에서 NODATA 픽셀은 건너뜁니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Raster layer** | INPUT_RASTER | [raster] | 입력 래스터 레이어 |
| **Band number** | RASTER_BAND | [raster band] | 데이터를 추출할 래스터 밴드 |
| **Field name** | FIELD_NAME | [string]<br>기본값: 'VALUE' | 래스터 밴드 값을 저장할 필드의 이름 |

다음 페이지에 계속

표 24.530 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Vector points** | `OUTPUT` | [vector: point]<br>기본값: `[Create temporary layer]` | Specify the resulting point layer of pixels centroids. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Vector points** | `OUTPUT` | [vector: point] | 픽셀 중심을 담은 산출 포인트 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:pixelstopoints`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 래스터 픽셀을 폴리곤으로

래스터 레이어에 있는 각 픽셀에 대응하는 폴리곤을 담은 벡터 레이어를 생성합니다.

래스터 레이어에 있는 각 개별 픽셀의 범위에 대해 폴리곤 피처를 생성해서 래스터 레이어를 벡터 레이어로 변환합니다. 산출물에서 NODATA 픽셀은 건너뜁니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Raster layer** | `INPUT_RASTER` | [raster] | 입력 래스터 레이어 |
| **Band number** | `RASTER_BAND` | [raster band] | 데이터를 추출할 래스터 밴드 |
| **Field name** | `FIELD_NAME` | [string]<br>기본값: 'VALUE' | 래스터 밴드 값을 저장할 필드의 이름 |

다음 페이지에 계속

표 24.532 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Vector polygons** | `OUTPUT` | [vector: polygon]<br>기본값: `[Create temporary layer]` | Specify the resulting polygon layer of pixel extents. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Vector polygons** | `OUTPUT` | [vector: polygon] | 픽셀 범위를 담은 산출 폴리곤 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:pixelstopolygons`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 정규 포인트 생성하기

지정한 범위 내부에 있는 정규 그리드에 배치된 포인트를 담은 새 포인트 레이어를 생성합니다.

포인트들 사이의 (모든 방향으로 동일한) 간격 또는 생성할 포인트의 개수로 그리드를 지정합니다. 후자의 경우, 범위로부터 간격을 결정할 것입니다. 완전한 직교 그리드를 생성하기 위해, 후자의 경우 최소한 사용자가 지정한 개수의 포인트를 생성합니다.

포인트 간격에 랜덤한 오프셋을 적용할 수 있습니다. 이렇게 하면 비정규 포인트 패턴을 산출합니다.

**기본 메뉴**: *Vector → Research Tools*

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input extent (xmin, xmax, ymin, ymax)** | `EXTENT` | [extent] | 랜덤 포인트용 맵 범위<br>다음과 같은 메소드를 사용할 수 있습니다:<br>• 레이어로부터 계산…: 현재 프로젝트에 불러온 레이어의 범위를 사용합니다.<br>• 레이어 맵으로부터 계산…: 활성 프로젝트의조판 맵 항목 범위를 사용합니다.<br>• 북마크로부터 계산…: 저장되어 있는북마크 의 범위를 사용합니다.<br>• 맵 캔버스 범위 사용<br>• 캔버스에 그리기: 연산에 넣을 범위를 정하는 직사각형을 클릭 & 드래그합니다.<br>• 좌표를 `xmin, xmax, ymin, ymax` 로 입력 |
| **Point spacing/count** | `SPACING` | [numeric: integer]<br>기본값: 100 | `Use point spacing` 파라미터의 활성화 여부에 따라 포인트들 사이의 간격 또는 포인트의 개수를 지정합니다. |
| **Initial inset from corner (LH side)** | `INSET` | [numeric: double]<br>기본값: 0.0 | 포인트를 좌상단에 상대적으로 오프셋시킵니다. 이 값은 X 및 Y 축 모두에 사용됩니다. |
| **Apply random offset to point spacing** | `RANDOMIZE` | [boolean]<br>기본값: False | 이 파라미터를 활성화하면 포인트에 랜덤한 간격을 적용할 것입니다. |
| **Use point spacing** | `IS_SPACING` | [boolean]<br>기본값: True | 이 파라미터를 비활성화하면 포인트 간격을 사용하지 않습니다. |
| **Output layer CRS** | `CRS` | [crs]<br>기본값: *Project CRS* | 랜덤 포인트 레이어의 좌표계 |
| **Regular points** | `OUTPUT` | [vector: point]<br>기본값: `[Create temporary layer]` | Specify the output regular point layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

#### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Regular points** | `OUTPUT` | [vector: point] | 산출 정규 포인트 레이어 |

#### 파이썬 코드

**알고리즘 ID**: `qgis:regularpoints`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 24.1.25 벡터 일반

#### 투영체 할당하기

벡터 레이어에 새 투영체를 할당합니다.

입력 레이어와 정확하게 동일한 피처 및 도형을 가지지만, 새 좌표계를 할당한 새 레이어를 생성합니다. 도형은 재투영되지 **않습니다**. 그냥 다른 좌표계로 할당될 뿐입니다.

이 알고리즘은 부정확한 투영체를 할당받은 레이어를 복구하는 데 쓰일 수 있습니다.

이 알고리즘은 속성을 수정하지 않습니다.

**더 보기**

*Define projection*, 투영체 찾기, 레이어 재투영하기

#### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | 잘못된 좌표계를 가진 또는 좌표계가 없는 벡터 레이어 |
| **Assigned CRS** | `CRS` | [crs]<br>기본값:<br>`EPSG:4326 - WGS84` | 벡터 레이어에 할당할 새 좌표계를 선택합니다. |

다음 페이지에 계속

표 24.536 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Assigned CRS**<br>선택적 | `OUTPUT` | [same as input]<br>기본값: `[Create temporary layer]` | Specify the output vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Assigned CRS** | `OUTPUT` | [same as input] | 할당된 투영체를 가진 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:assignprojection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 배치 (batch) 노미나팀 지오코더

입력 레이어 문자열 필드를 대상으로 노미나팀 (Nominatim) 서비스를 이용해서 배치 지오코딩 작업을 수행합니다. 산출 레이어는 지오코딩된 위치는 물론 지오코딩된 위치와 관련된 속성 여러 개를 반영한 포인트 객체를 가질 것입니다.

☑ 포인트 피처를제자리 피처 수정 할 수 있습니다.

**참고**

이 알고리즘은 오픈스트리트맵 재단이 제공하는 노미나팀 지오코딩 서비스의 활용 정책 을 준수합니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: any] | 피처를 지오코딩할 벡터 레이어 |
| **Address field** | `FIELD` | [tablefield: string] | 지오코딩할 주소를 담고 있는 필드 |
| **Geocoded** | `OUTPUT` | [vector: point]<br>기본값: `[Create temporary layer]` | Specify the output layer containing only the geocoded addresses. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Geocoded** | `OUTPUT` | [vector: point] | 지오코딩된 주소에 대응하는 포인트 피처를 가진 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:batchnominatimgeocoder`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 레이어를 공간 북마크로 변환하기

레이어에 있는 피처의 범위에 대응하는 공간 북마크를 생성합니다.

#### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input Layer** | `INPUT` | [vector: line, polygon] | 입력 벡터 레이어 |
| **Bookmark destination** | `DESTINATION` | [enumeration] 기본값: 0 | 북마크의 위치를 선택합니다. 다음 가운데 하나를 선택할 수 있습니다:<br>• 0 —프로젝트 북마크<br>• 1 —사용자 북마크 |
| **Name field** | `NAME_EXPRESSION` | [expression] | 생성된 북마크에 이름을 부여할 필드 또는 표현식 |
| **Group field** | `GROUP_EXPRESSIC` | [expression] | 생성된 북마크를 위한 그룹을 제공할 필드 또는 표현식 |

#### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Count of bookmarks added** | `COUNT` | [numeric: integer] | |

#### 파이썬 코드

**알고리즘 ID**: `native:layertobookmarks`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 공간 북마크를 레이어로 변환하기

저장된 공간 북마크에 대한 폴리곤 피처를 담고 있는 새 레이어를 생성합니다. 현재 프로젝트에 속한 북마크만 내보내도록 필터링할 수도 있고, 모든 사용자 북마크를 내보내도록 할 수도 있고, 또는 이 두 옵션을 조합할 수도 있습니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Bookmark source** | `SOURCE` | [enumeration] [list]<br>기본값: [0,1] | 북마크의 소스 (들) 을 선택합니다. 다음 가운데 하나 이상을 선택할 수 있습니다:<br>• 0 — 프로젝트 북마크<br>• 1 — 사용자 북마크 |
| **Output CRS** | `CRS` | [crs]<br>기본값: `EPSG:4326 - WGS84` | 산출 레이어의 좌표계 |
| **Output** | `OUTPUT` | [vector: polygon]<br>기본값: `[Create temporary layer]` | Specify the output layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output** | `OUTPUT` | [vector: polygon] | 산출 (북마크) 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:bookmarkstolayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 속성 인덱스 생성하기

쿼리 속도를 높이기 위해 속성 테이블의 필드에 대해 인덱스를 생성합니다. 레이어의 데이터 제공자와 필드 유형에 따라 인덱스 생성 지원 여부가 달라집니다.

어떤 새 산출물도 생성하지 않습니다. 레이어 자체에 인덱스를 저장합니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input Layer** | `INPUT` | [vector: any] | 속성 인덱스를 생성하려는 벡터 레이어를 선택합니다. |
| **Attribute to index** | `FIELD` | [tablefield: any] | 벡터 레이어의 필드 |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Indexed layer** | `OUTPUT` | [same as input] | 입력 벡터 레이어에 지정한 필드에 대한 인덱스를 추가한 복사본 |

### 파이썬 코드

**알고리즘 ID**: `native:createattributeindex`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 공간 인덱스 생성하기

레이어에 있는 피처로의 접근 속도를 높이기 위해 피처의 공간적 위치를 바탕으로 인덱스를 생성합니다. 레이어의 데이터 제공자에 따라 공간 인덱스 생성 지원 여부가 달라집니다.

어떤 새 산출물도 생성하지 않습니다.

**기본 메뉴**: *Vector → Data Management Tools*

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input Layer** | `INPUT` | [vector: geometry] | 입력 벡터 레이어 |

#### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| 인덱스를 생성한 레이어 | `OUTPUT` | [same as input] | 입력 벡터 레이어에 공간 인덱스를 추가한 복사본 |

#### 파이썬 코드

**알고리즘 ID**: `native:createspatialindex`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### Define projection

Sets an existing layer's projection to the provided CRS without reprojecting features.

투영체 할당하기 알고리즘과는 반대로 이 알고리즘은 현재 레이어를 수정하지, 새 레이어를 산출하지는 않을 것입니다.

**참고**

셰이프파일 포맷 데이터셋의 경우, 지정한 좌표계와 일치하도록 `.prj` 및 `.qpj` 파일을 덮어쓸 것입니다. 해당 파일이 없는 경우 생성할 것입니다.

**기본 메뉴**: *Vector → Data Management Tools*

**더 보기**

투영체 할당하기, 투영체 찾기, 레이어 재투영하기

#### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | 투영체 정보가 빠진 벡터 레이어 |
| **CRS** | `CRS` | [crs] | 벡터 레이어에 할당할 좌표계를 선택합니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| | INPUT | [same as input] | 정의된 투영체를 가진 입력 벡터 레이어 |

### 파이썬 코드

**Algorithm ID**: `native:definecurrentprojection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 복제 도형 삭제하기

복제된 도형을 찾아서 제거합니다.

속성까지 확인하지는 않기 때문에, 피처 2 개가 동일한 도형을 가지고 있지만 속성은 다른 경우 둘 중 하나만 산출 레이어에 추가될 것입니다.

**참고**

This algorithm does not require valid geometries as input.

**더 보기**

도형 삭제, *NULL* 도형 제거하기, 속성으로 사본 삭제하기

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | INPUT | [vector: geometry] | 사용자가 삭제하고 싶은 복제 도형이 담긴 레이어 |

다음 페이지에 계속

표 24.550 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Cleaned** | `OUTPUT` | [same as input]<br>기본값: `[Create temporary layer]` | Specify the output layer containing the unique features. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Count of discarded duplicate records** | `DUPLICATE_COUNT` | [numeric: integer] | 삭제한 복제 레코드의 개수 |
| **Cleaned** | `OUTPUT` | [same as input] | 복제 도형이 없는 산출 레이어 |
| **Count of retained records** | `RETAINED_COUNT` | [numeric: integer] | 유일 레코드의 개수 |

### 파이썬 코드

**알고리즘 ID**: `native:deleteduplicategeometries`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 속성으로 사본 삭제하기

Deletes duplicate rows by only considering the specified field(s). The first matching row will be retained, and duplicates will be discarded.

이 복제 레코드를 분석을 위한 개별 산출물로 저장할 수 있는 옵션이 있습니다.

**더 보기**

복제 도형 삭제하기

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: any] | 입력 레이어 |
| **Fields to match duplicates by** | `FIELDS` | [tablefield: any] [list] | 사본을 정의하는 필드입니다. 이 모든 필드에서 일치하는 값을 가진 피처를 사본으로 식별합니다. |
| **Filtered (no duplicates)** | `OUTPUT` | [same as input]<br>기본값: `[Create temporary layer]` | Specify the output layer containing the unique features. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |
| **Filtered (duplicates)**<br>선택적 | `DUPLICATES` | [same as input]<br>기본값: `[Skip output]` | Specify the output layer containing only the duplicates. *One of*:<br>• 산출물 건너뛰기<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Filtered (duplicates)** | `DUPLICATES` | [same as input] | Vector layer containing the duplicate features. |
| **Count of discarded duplicate records** | `DUPLICATE_COUNT` | [numeric: integer] | 삭제한 복제 레코드의 개수 |
| **Filtered (no duplicates)** | `OUTPUT` | [same as input] | 유일 피처를 담고 있는 벡터 레이어 |
| **Count of retained records** | `RETAINED_COUNT` | [numeric: integer] | 유일 레코드의 개수 |

### 파이썬 코드

**알고리즘 ID**: `native:removeduplicatesbyattribute`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 데이터셋 변경 사항 탐지

두 벡터 레이어를 비교해서 두 레이어 사이에 어느 피처가 변경되지 않았는지, 추가되었는지, 또는 삭제되었는지 판별합니다. 이 알고리즘은 동일한 데이터셋의 서로 다른 두 버전을 비교하기 위해 설계되었습니다.

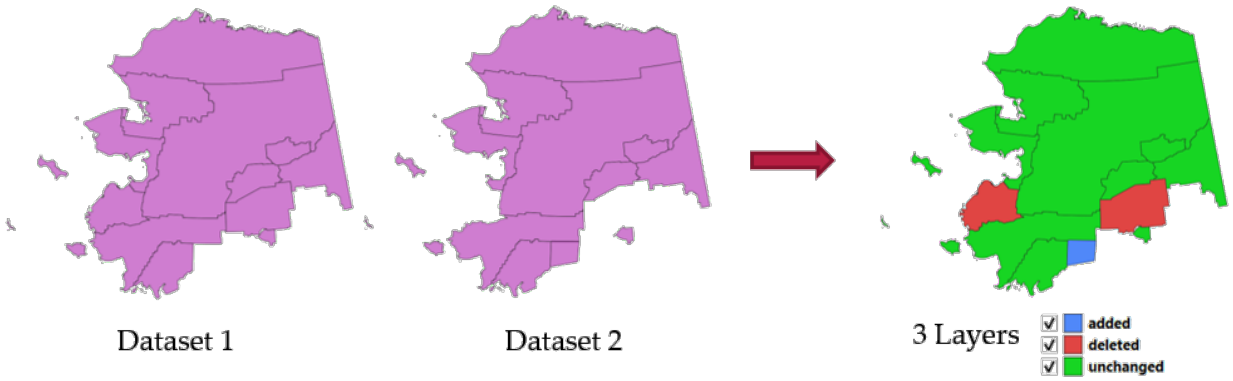



그림 24.85: 데이터셋 변경 사항 탐지의 예

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Original layer** | `ORIGINAL` | [vector: geometry] | 원본 버전으로 간주하는 벡터 레이어 |
| **Revised layer** | `REVISED` | [vector: geometry] | 개정된 또는 수정된 벡터 레이어 |
| **Attributes to consider for match** 선택적 | `COMPARE_ATTRIBUTES` | [tablefield: any] [list] | 일치하는지 고려할 속성입니다. 기본적으로 모든 속성을 비교합니다. |
| **Geometry comparison behavior** 선택적 | `MATCH_TYPE` | [enumeration] 기본값: 1 | 비교 규칙을 정의합니다. 다음 옵션 가운데 선택할 수 있습니다:<br>• 0 — 완전 일치 (Exact Match): 도형의 순서 및 꼭짓점 개수도 포함<br>• 1 — 오차 허용 일치 (Tolerant Match, 위상적 동등성): 도형을 동등한 것으로 간주합니다. |

다음 페이지에 계속

표 24.554 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Unchanged features**<br>선택적 | UNCHANGED | [vector: same as Original layer] | Specify the output vector layer containing the unchanged features. *One of*:<br>• 산출물 건너뛰기<br>• 임시 레이어 생성 (TEMPORARY_OUTPUT)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |
| **Added features**<br>선택적 | ADDED | [vector: same as Original layer] | Specify the output vector layer containing the added features. *One of*:<br>• 산출물 건너뛰기<br>• 임시 레이어 생성 (TEMPORARY_OUTPUT)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |
| **Deleted features**<br>선택적 | DELETED | [vector: same as Original layer] | Specify the output vector layer containing the deleted features. *One of*:<br>• 산출물 건너뛰기<br>• 임시 레이어 생성 (TEMPORARY_OUTPUT)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Unchanged features** | UNCHANGED | [vector: same as Original layer] | 변경되지 않은 피처를 담고 있는 벡터 레이어 |
| **Added features** | ADDED | [vector: same as Original layer] | 추가된 피처를 담고 있는 벡터 레이어 |
| **Deleted features** | DELETED | [vector: same as Original layer] | 삭제된 피처를 담고 있는 벡터 레이어 |
| **Count of unchanged features** | UNCHANGED_COUNT | [numeric: integer] | 변경되지 않은 피처의 개수 |
| **Count of features added in revised layer** | ADDED_COUNT | [numeric: integer] | 개정 레이어에 추가된 피처의 개수 |

다음 페이지에 계속

표 24.555 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Count of features deleted from original layer** | DELETED_COUNT | [numeric: integer] | 원본 레이어에서 삭제된 피처의 개수 |

### 파이썬 코드

**알고리즘 ID**: native:detectvectorchanges

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 도형 삭제

입력 레이어 속성 테이블의 단순한 무 *(☐)* 도형 복사본을 생성합니다. 소스 레이어의 속성 테이블은 그대로 유지합니다.

파일을 로컬 폴더에 저장하는 경우, 수많은 포맷 가운데 선택할 수 있습니다.

☑ 포인트, 라인 및 폴리곤 피처를제자리 피처 수정 할 수 있습니다.

**더 보기**

복제 도형 삭제하기, *NULL* 도형 제거하기

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | INPUT | [vector: geometry] | 입력 벡터 레이어 |
| **Dropped geometries** | OUTPUT | [vector: table] | Specify the output geometryless layer. *One of*:<br>• 임시 레이어 생성 (TEMPORARY_OUTPUT)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

#### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Dropped geometries** | `OUTPUT` | [vector: table] | 산출 무 (□) 도형 레이어. 원본 속성 테이블의 복사본입니다. |

#### 파이썬 코드

**알고리즘 ID**: `native:dropgeometries`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### SQL 실행하기

소스 레이어에 `SQL` 문법으로 된 SELECT 절만을 기반으로 하는 단순 또는 복잡 쿼리를 실행합니다.

입력 데이터소스를 `input1`, `input2`, ⋯, `inputN` 으로 식별하므로, 단순 쿼리는 `SELECT * FROM input1` 과 같이 보일 것입니다.

단순 쿼리 이외에, `SQL query` 파라미터 자체에 표현식 또는 변수를 추가할 수 있습니다. 이 알고리즘을 공간 처리 모델 내에서 실행하는데 모델 입력물을 쿼리의 파라미터로 받고자 할 때 매우 유용한 방법입니다. 이때 쿼리의 예를 들자면 `SELECT * FROM [% @table %]` 정도가 되는데, 여기서 `@table` 은 모델 입력물로 식별되는 변수입니다.

쿼리의 결과를 새 레이어로 추가할 것입니다.

**더 보기**

*SpatiaLite* 에서 *SQL* 실행, *PostgreSQL* 에서 *SQL* 실행

#### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Additional input datasources (called input1, .., inputN in the query)** | `INPUT_DATASOURCES` | [vector: any] [list] | 쿼리할 레이어의 목록입니다. SQL 편집기에서, 얼마나 많은 레이어를 선택했느냐에 따라 이 레이어들을 **실제** 이름으로 참조할 수도 있고 또는 **input1**, **input2**, ⋯, **inputN** 으로 참조할 수도 있습니다. |
| **SQL query** | `INPUT_QUERY` | [string] | 사용자 SQL 쿼리 문자열을, 예를 들어 `SELECT * FROM input1` 처럼 입력합니다. |

다음 페이지에 계속

표 24.558 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Unique identifier field**<br>선택적 | `INPUT_UID_FIELD` | [string] | 유일 ID 를 가진 열을 지정합니다. |
| **Geometry field**<br>선택적 | `INPUT_GEOMETRY_FI` | [string] | 도형 필드를 지정합니다. |
| **Geometry type**<br>선택적 | `INPUT_GEOMETRY_TY` | [enumeration]<br>기본값: 0 | 산출되는 도형의 유형을 선택합니다. 기본적으로 알고리즘이 자동 탐지할 것입니다. 다음 가운데 하나로 선택할 수 있습니다:<br>• 0 —자동 탐지 (Autodetect)<br>• 1 —도형 없음<br>• 2 —포인트<br>• 3 —라인스트링<br>• 4 —폴리곤<br>• 5 —멀티포인트<br>• 6 —멀티라인스트링<br>• 7 —멀티폴리곤 |
| **CRS**<br>선택적 | `INPUT_GEOMETRY_CR` | [crs] | 산출 레이어에 할당할 좌표계 |
| **SQL Output** | `OUTPUT` | [vector: geometry]<br>기본값: `[Create temporary layer]` | Specify the output layer created by the query. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **SQL Output** | `OUTPUT` | [vector: geometry] | 쿼리가 생성한 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `qgis:executesql`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 레이어를 DXF 로 내보내기

레이어를 DXF 파일로 내보냅니다. 각 레이어에 대해, DXF 파일에 생성되는 대상 레이어에 있는 피처를 분할하는 데 사용되는 값을 가지고 있는 필드를 선택할 수 있습니다.

**더 보기**

새 *DXF* 파일 생성하기

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layers** | LAYERS | [vector: geometry] [list] | List of input vector layers with options associated (filled as a QgsProcessingParameterDxfLayers item —done in GUI by pressing *Configure Layer*⋯ button for each selected layer):<br>**Layer [string] (layer)**<br>Full path of the input layer to export<br>**Output layer attribute [tablefield: any] (attributeIndex)**<br>Attribute index to split the input layer using unique values<br>**Output layer name [string] (overriddenLayerName)**<br>Overridden layer name to be used in the exported DXF file (the original project layer remains untouched)<br>**Allow data defined symbol blocks** [boolean] (buildDataDefinedBlocks)<br>**Maximum number of symbol blocks [numeric: integer] (dataDefinedBlocksMaximumNumberOfClasses)**<br>−1 means no limitation. |
| **Symbology mode** | SYMBOLOGY_MODE | [enumeration] 기본값: 0 | 산출 레이어에 적용할 심볼 유형입니다. 다음 가운데 하나를 선택할 수 있습니다:<br>• 0 —심볼 없음<br>• 1 —피처 심볼<br>• 2 —심볼 레이어 심볼 |
| **Symbology scale** | SYMBOLOGY_SCALE | [scale] 기본값: 1:1,000,000 | 데이터 내보내기의 기본 축척입니다. |
| **Map theme** 선택적 | MAP_THEME | [map theme] | Match layer styling to the provided map theme. |
| **Encoding** | ENCODING | [enumeration] | 레이어에 적용할 인코딩입니다. |
| **CRS** | CRS | [crs] | 산출 레이어의 좌표계를 선택합니다. |
| **Extent** 선택적 | EXTENT | [extent] | Limit exported features to those with geometries intersecting the provided extent. |

다음 페이지에 계속

표 24.560 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Use layer title as name** | `USE_LAYER_TITLE` | [boolean]<br>기본값: False | Name the output layer with the layer title (as set in layer metadata or QGIS Server properties) instead of the layer name. |
| **Force 2D** | `FORCE_2D` | [boolean]<br>기본값: False | |
| **Export labels as MTEXT elements** | `MTEXT` | [boolean]<br>기본값: True | 라벨을 MTEXT 또는 TEXT 로 내보냅니다. |
| **Use only selected features** | `SELECTED_FEATURES` | [boolean]<br>기본값: False | Exports only the selected features. |
| **DXF** | `OUTPUT` | [file]<br>기본값: `[Save to temporary file]` | Specification of the output DXF file. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **DXF** | `OUTPUT` | [file] | 입력 레이어를 담고 있는 `.DXF` 파일 |

### 파이썬 코드

**알고리즘 ID**: `native:dxfexport`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 선택한 피처 추출하기

선택한 피처를 새 레이어로 저장합니다.

> **참고**
>
> 지정한 레이어에서 선택한 피처가 없는 경우, 빈 레이어를 생성할 것입니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input Layer** | `INPUT` | [vector: any] | 저장할 피처를 선택한 레이어 |
| **Selected features** | `OUTPUT` | [same as input]<br>기본값: `[Create temporary layer]` | Specify the vector layer for the selected features. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Selected features** | `OUTPUT` | [same as input] | 선택한 피처만 담은 벡터 레이어, 또는 아무 피처도 선택하지 않은 경우 피처가 없는 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:saveselectedfeatures`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 셰이프파일 인코딩 정보 추출

이 알고리즘은 셰이프파일에 내장된 속성 인코딩 정보를 추출합니다. 부가적인 `.cpg` 파일로 지정된 인코딩과 `.dbf` LDID 헤더 블록에 있는 모든 인코딩 상세 사항을 모두 고려합니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input Layer** | `INPUT` | [vector: geometry] | 인코딩 정보를 추출할 ESRI 셰이프파일 (`.SHP`) 레이어 |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Shapefile encoding** | `ENCODING` | [string] | 입력 파일에 지정된 인코딩 정보 |
| **CPG encoding** | `CPG_ENCODING` | [string] | 부가적인 `.CPG` 파일에 지정된 인코딩 정보 |
| **LDID encoding** | `LDID_ENCODING` | [string] | `.dbf` LDID 헤더 블록에 지정된 인코딩 정보 |

### 파이썬 코드

**알고리즘 ID**: `native:shpencodinginfo`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 투영체 찾기

최종 후보 좌표계 목록을, 예를 들어 알 수 없는 투영체를 가진 레이어를 위해 생성합니다.

대상 영역 파라미터를 통해 레이어가 커버하리라 기대되는 영역을 지정해야만 합니다. QGIS 는 이 대상 영역의 좌표계를 알고 있어야만 합니다.

이 알고리즘은 알려진 모든 좌표계에서 레이어의 범위를 테스트해보고, 레이어의 경계가 대상 영역에 근접하는 모든 투영체를 목록화합니다.

**더 보기**

투영체 할당하기, *Define projection*, 레이어 재투영하기

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input Layer** | `INPUT` | [vector: geometry] | 알 수 없는 투영체를 가진 레이어 |
| **Target area for layer (xmin, xmax, ymin, ymax)** | `TARGET_AREA` | [extent] | 레이어가 커버하는 영역입니다.<br>다음과 같은 메소드를 사용할 수 있습니다:<br>• 레이어로부터 계산…: 현재 프로젝트에 불러온 레이어의 범위를 사용합니다.<br>• 레이어 맵으로부터 계산…: 활성 프로젝트의조판 맵 항목 범위를 사용합니다.<br>• 북마크로부터 계산…: 저장되어 있는북마크 의 범위를 사용합니다.<br>• 맵 캔버스 범위 사용<br>• 캔버스에 그리기: 연산에 넣을 범위를 정하는 직사각형을 클릭 & 드래그합니다.<br>• 좌표를 `xmin, xmax, ymin, ymax` 로 입력 |
| **CRS candidates** | `OUTPUT` | [vector: table]<br>기본값: `[Create temporary layer]` | Specify the table (geometryless layer) for the CRS suggestions (EPSG codes). *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **CRS candidates** | `OUTPUT` | [vector: table] | 기준을 만족시키는 모든 좌표계 (EPSG 코드) 를 담은 테이블 |

### 파이썬 코드

**Algorithm ID**: `native:findprojection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 관계 평탄화

관련 자식 피처 하나 당 부모 피처 하나를 담고 있는 단일 레이어를 내보내서 벡터 레이어에 대한관계 를 평탄화 (flatten) 합니다. 이 마스터 피처는 관련 피처들에 대한 모든 속성을 담고 있습니다. 이 알고리즘을 사용하면 관계를, 예를 들어 CSV 와 같은 일반적인 테이블로 내보낼 수 있게 됩니다.

> **경고**
>
> This algorithm drops existing primary keys or FID values and regenerates them in output layers.

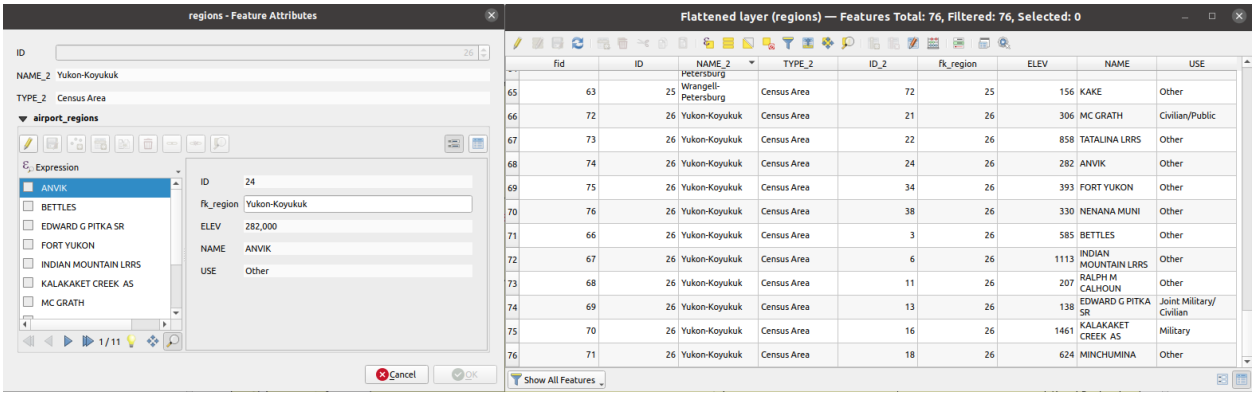



그림 24.86: 관련 자식을 가진 지역의 양식 (왼쪽) - 각 관련 자식마다 결합된 속성을 가진 복제 지역 피처 (오른쪽)

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input Layer** | `INPUT` | [vector: any] | 비정규화 (de-normalize) 해야 할 관계를 가진 레이어 |
| **Flattened Layer** 선택적 | `OUTPUT` | [same as input] 기본값: `[Create temporary layer]` | Specify the output (flattened) layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Flattened layer** | `OUTPUT` | [same as input] | 관련 피처들의 모든 속성을 가진 마스터 피처를 담고 있는 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:flattenrelationships`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 필드값으로 속성 결합하기

입력 벡터 레이어를 받아 입력 레이어의 속성 테이블에 속성을 추가한 확장 버전인 새 벡터 레이어를 생성합니다.

두 번째 벡터 레이어에서 추가 속성 및 그 값을 가져옵니다. 결합 기준을 정의하기 위해 각 레이어에서 속성을 선택합니다.

**경고**

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

**더 보기**

최근접으로 속성 결합하기, 위치로 속성 결합하기

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input Layer** | `INPUT` | [vector: any] | 입력 벡터 레이어. 산출 레이어는 이 레이어의 피처와 두 번째 레이어에 있는 일치하는 피처의 속성으로 이루어질 것입니다. |
| **Table field** | `FIELD` | [tablefield: any] | 결합을 위해 사용할 소스 레이어의 필드 |
| **Input layer 2** | `INPUT_2` | [vector: any] | 결합할 속성 테이블을 가진 레이어 |

다음 페이지에 계속

표 24.570 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Table field 2** | FIELD_2 | [tablefield: any] | 결합을 위해 사용할 두 번째 (결합) 레이어의 필드. 필드 유형은 입력 테이블의 필드 유형과 동일해야만 (또는 호환되어야만) 합니다. |
| **Layer 2 fields to copy**<br>선택적 | FIELDS_TO_COPY | [tablefield: any] [list] | 사용자가 추가하고자 하는 특정 필드를 선택합니다. 기본적으로 모든 필드를 추가합니다. |
| **Join type** | METHOD | [enumeration]<br>기본값: 1 | 최종 결합 레이어의 유형을 지정합니다. 다음 가운데 하나를 선택할 수 있습니다:<br>• 0 —일치하는 각 피처마다 개별 피처를 생성합니다. (1 대다)<br>• 1 —처음으로 일치하는 피처의 속성만 가져옵니다. (1 대 1) |
| **Discard records which could not be joined** | DISCARD_NONMATCHIN | [boolean]<br>기본값: True | 결합할 수 없는 피처를 유지하지 않으려면 이 파라미터를 활성화하십시오. |
| **Joined field prefix**<br>선택적 | PREFIX | [string] | 결합된 필드명에 접두어를 추가합니다. 결합된 필드를 더 쉽게 식별하고 이름의 충돌을 방지하기 위해서입니다. |
| **Joined layer**<br>선택적 | OUTPUT | [same as input]<br>기본값: [Create temporary layer] | Specify the output vector layer for the join. *One of*:<br>• 산출물 건너뛰기<br>• 임시 레이어 생성 (TEMPORARY_OUTPUT)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |
| **Unjoinable features from first layer**<br>선택적 | NON_MATCHING | [same as input]<br>기본값: [Skip output] | Specify the output vector layer for unjoinable features from first layer. *One of*:<br>• 산출물 건너뛰기<br>• 임시 레이어 생성 (TEMPORARY_OUTPUT)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Number of joined features from input table** | `JOINED_COUNT` | [numeric: integer] | |
| **Unjoinable features from first layer**<br>선택적 | `NON_MATCHING` | [same as input] | 일치하지 않는 피처를 담은 벡터 레이어 |
| **Joined layer**<br>선택적 | `OUTPUT` | [same as input] | 결합으로 추가된 속성을 가진 산출 벡터 레이어 |
| **Number of unjoinable features from input table**<br>선택적 | `UNJOINABLE_COUNT` | [numeric: integer] | |

### 파이썬 코드

**알고리즘 ID**: `native:joinattributestable`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 위치로 속성 결합하기

입력 벡터 레이어를 받아 입력 레이어의 속성 테이블에 속성을 추가한 확장 버전인 새 벡터 레이어를 생성합니다.

두 번째 벡터 레이어에서 추가 속성 및 그 값을 가져옵니다. 첫 번째 레이어의 각 피처에 추가되는 두 번째 레이어의 값을 선택하기 위한 공간 기준을 적용합니다.

**기본 메뉴**: *Vector → Data Management Tools*

**경고**

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

**더 보기**

최근접으로 속성 결합하기, 필드값으로 속성 결합하기, 위치로 속성 결합하기 *(요약)*

### 공간 관계성 탐색하기

기하 서술 (geometric predicates) 은 피처들의 도형들이 공간 일부분을 공유하는지, 공유한다면 어떻게 공유하는지를 비교해서, 어떤 피처가 또다른 피처와 어떤 공간 관계성을 가지는지를 판단하는 데 사용되는 불 (boolean) 함수입니다.

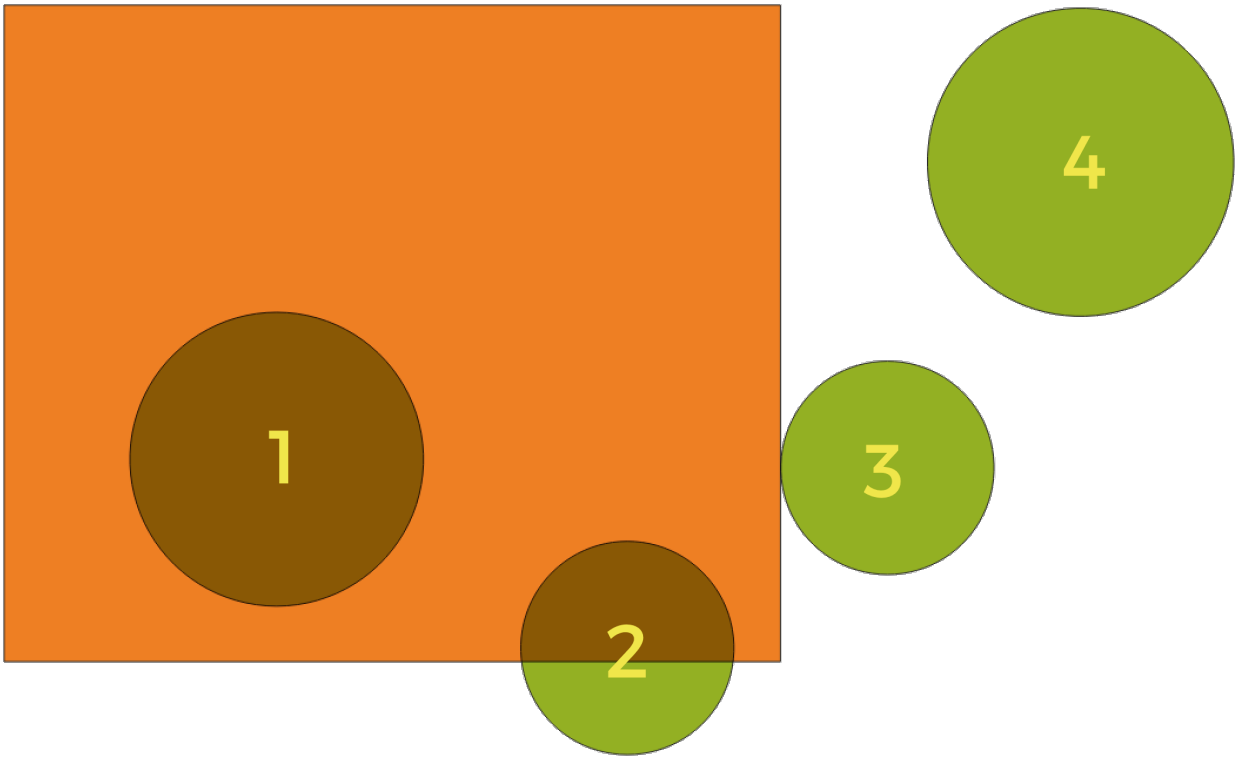



그림 24.87: 레이어들 사이의 공간 관계성 검색

앞의 그림을 이용해서, 녹색 원 피처를 주황색 직사각형 피처와 공간적으로 비교해서 녹색 원 피처를 검색할 것입니다. 다음과 같은 기하 서술을 사용할 수 있습니다:

**교차** ***(Intersection)***
: 도형들이 서로 교차하는지 검증합니다. 도형들이 공간적으로 교차하는 경우 (공간의 어떤 부분이라도 공유–중첩 또는 접하는 경우) 1(참) 을 반환하고 교차하지 않는 경우 0(거짓) 을 반환합니다. 앞의 그림에서, 1 번, 2 번 및 3 번 원을 반환할 것입니다.

**포함** ***(Contain)***
: B 의 포인트 가운데 A 외부에 있는 포인트가 하나도 없고 B 내부의 포인트 가운데 최소한 하나가 A 의 내부에 있는 경우에만 1(참) 을 반환합니다. 앞의 그림에서 어떤 원도 반환하지 않지만, 반대로 검색할 경우 직사각형이 1 번 원을 완전히 담고 있기 때문에 직사각형을 반환할 것입니다. 내부 *(are within)* 의 역 (□) 입니다.

**분절** ***(Disjoint)***
: 도형들이 공간의 어떤 부분도 공유하지 않는–중첩하지도 접하지도 않는 경우 1(참) 을 반환합니다. 4 번 원만 반환합니다.

**동등** ***(Equal)***
: 도형들이 정확하게 동일한 경우에만 1(참) 을 반환합니다. 아무 원도 반환하지 않을 것입니다.

**접촉** ***(Touch)***
: 도형이 다른 도형과 접하는지 검증합니다. 도형들이 최소한 포인트 1 개를 공유하지만 각 도형의 내부가

교차하지 않는 경우 1(참) 을 반환합니다. 3 번 원만 반환합니다.

**중첩** ***(Overlap)***
도형이 다른 도형과 중첩하는지 검증합니다. 도형들이 동일한 차원이며 공간을 공유하지만 서로를 완전히 담고 있지 않은 경우 1(참) 을 반환합니다. 2 번 원만 반환합니다.

**내부** ***(Are within)***
도형이 다른 도형 내부에 있는지 검증합니다. A 가 B 내부에 완전히 들어가 있는 경우 1(참) 을 반환합니다. 1 번 원만 반환합니다.

**공간교차** ***(Cross)***
입력 도형들이 전부는 아니지만 일부 내부 포인트를 공유하지만 입력 도형의 최고 차원보다 낮은 차원에서 실제로 가로지르는 경우 1(참) 을 반환합니다. 예를 들어, 폴리곤을 가로지르는 라인은 라인으로서 공간교차할 것입니다 (참). 라인을 가로지르는 라인은 포인트로서 공간교차할 것입니다 (참). 폴리곤을 가로지르는 폴리곤은 폴리곤으로서 가로지릅니다 (거짓). 앞의 그림에서, 아무 원도 반환하지 않을 것입니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Join to features in** | `INPUT` | [vector: geometry] | 입력 벡터 레이어. 산출 레이어는 이 레이어의 피처와 두 번째 레이어에 있는 일치하는 피처의 속성으로 이루어질 것입니다. |
| **Where the features** | `PREDICATE` | [enumeration] [list] 기본값: [0] | 소스 피처와 대상 피처를 결합시킬 수 있으려면 두 피처가 가져야 할 공간 관계의 유형입니다. 다음 가운데 하나 이상을 선택할 수 있습니다:<br>• 0 — 교차 (intersect)<br>• 1 — 포함 (contain)<br>• 2 — 동등 (equal)<br>• 3 — 접촉 (touch)<br>• 4 — 중첩 (overlap)<br>• 5 — 내부 (are within)<br>• 6 — 공간교차 (cross)<br>조건을 하나 이상 선택한 경우, 적어도 하나는 (OR 연산자) 추출할 피처를 만족시켜야 합니다. |
| **By comparing to** | `JOIN` | [vector: geometry] | 결합 레이어입니다. 이 벡터 레이어의 피처가 공간 관계를 만족하는 경우 소스 레이어의 속성 테이블에 자신의 속성을 **추가** 할 것입니다. |
| **Fields to add (leave empty to use all fields)**<br>선택적 | `JOIN_FIELDS` | [tablefield: any] [list] | 결합 레이어에서 사용자가 추가하고자 하는 특정 필드를 선택합니다. 기본적으로 모든 필드를 추가합니다. |

다음 페이지에 계속

표 24.572 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Join type** | METHOD | [enumeration] | 최종 결합 레이어의 유형을 지정합니다. 다음 가운데 하나를 선택할 수 있습니다:<br>• 0 —일치하는 각 피처마다 개별 피처를 생성합니다. (1 대다)<br>• 1 —처음으로 일치하는 피처의 속성만 가져옵니다. (1 대 1)<br>• 2 —최대 중첩 영역을 가진 피처의 속성만 가져오기 (1 대 1) |
| **Discard records which could not be joined** | DISCARD_NONMATCHIN | [boolean]<br>기본값: False | 산출물에서 결합시킬 수 없는 입력 레이어 피처를 제거합니다. |
| **Joined field prefix**<br>선택적 | PREFIX | [string] | 결합된 필드명에 접두어를 추가합니다. 결합된 필드를 더 쉽게 식별하고 이름의 충돌을 방지하기 위해서입니다. |
| **Joined layer**<br>선택적 | OUTPUT | [same as input]<br>기본값: [Create temporary layer] | Specify the output vector layer for the join. *One of*:<br>• 산출물 건너뛰기<br>• 임시 레이어 생성 (TEMPORARY_OUTPUT)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |
| **Unjoinable features from first layer**<br>선택적 | NON_MATCHING | [same as input]<br>기본값: [Skip output] | Specify the output vector layer for unjoinable features from first layer. *One of*:<br>• 산출물 건너뛰기<br>• 임시 레이어 생성 (TEMPORARY_OUTPUT)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Number of joined features from input table** | JOINED_COUNT | [numeric: integer] | |
| **Unjoinable features from first layer**<br>선택적 | NON_MATCHING | [same as input] | 일치하지 않는 피처를 담은 벡터 레이어 |

다음 페이지에 계속

표 24.573 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Joined layer** | OUTPUT | [same as input] | 결합으로 추가된 속성을 가진 산출 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:joinattributesbylocation`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 위치로 속성 결합하기 (요약)

입력 벡터 레이어를 받아 입력 레이어의 속성 테이블에 속성을 추가한 확장 버전인 새 벡터 레이어를 생성합니다.

두 번째 벡터 레이어에서 추가 속성 및 그 값을 가져옵니다. 첫 번째 레이어의 각 피처에 추가되는 두 번째 레이어의 값을 선택하기 위한 공간 기준을 적용합니다.

이 알고리즘은 두 번째 레이어에서 일치하는 피처의 값에 대한 통계 요약을 계산합니다. (예: 최대값, 평균값 등등)

**더 보기**

위치로 속성 결합하기

## 공간 관계성 탐색하기

기하 서술 (geometric predicates) 은 피처들의 도형들이 공간 일부분을 공유하는지, 공유한다면 어떻게 공유하는지를 비교해서, 어떤 피처가 또다른 피처와 어떤 공간 관계성을 가지는지를 판단하는 데 사용되는 불 (boolean) 함수입니다.

앞의 그림을 이용해서, 녹색 원 피처를 주황색 직사각형 피처와 공간적으로 비교해서 녹색 원 피처를 검색할 것입니다. 다음과 같은 기하 서술을 사용할 수 있습니다:

**교차** ***(Intersection)***
: 도형들이 서로 교차하는지 검증합니다. 도형들이 공간적으로 교차하는 경우 (공간의 어떤 부분이라도 공유–중첩 또는 접하는 경우) 1(참) 을 반환하고 교차하지 않는 경우 0(거짓) 을 반환합니다. 앞의 그림에서, 1 번, 2 번 및 3 번 원을 반환할 것입니다.

**포함** ***(Contain)***
: B 의 포인트 가운데 A 외부에 있는 포인트가 하나도 없고 B 내부의 포인트 가운데 최소한 하나가 A 의 내부에 있는 경우에만 1(참) 을 반환합니다. 앞의 그림에서 어떤 원도 반환하지 않지만, 반대로 검색할 경우 직사각형이 1 번 원을 완전히 담고 있기 때문에 직사각형을 반환할 것입니다. 내부 *(are within)* 의 역 (□) 입니다.

**분절** ***(Disjoint)***
: 도형들이 공간의 어떤 부분도 공유하지 않는–중첩하지도 접하지도 않는 경우 1(참) 을 반환합니다. 4 번 원만 반환합니다.

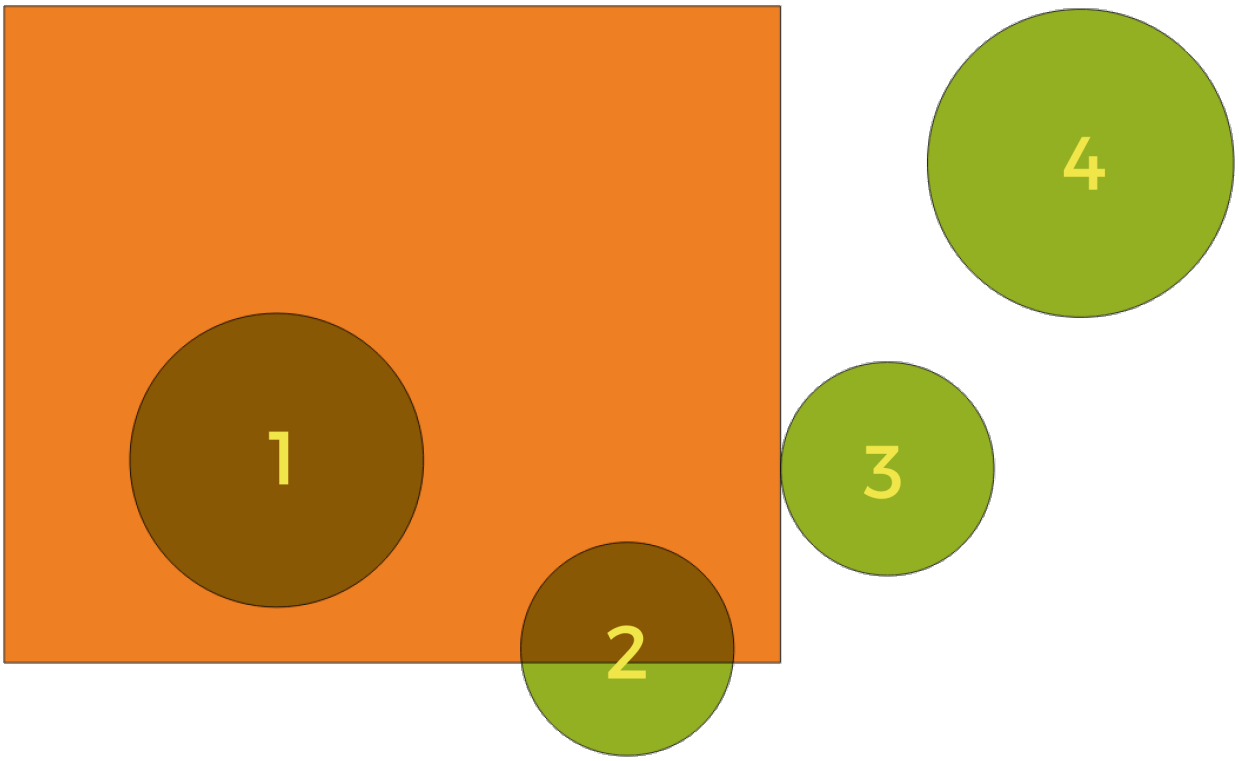



그림 24.88: 레이어들 사이의 공간 관계성 검색

**동등** ***(Equal)***
도형들이 정확하게 동일한 경우에만 1(참) 을 반환합니다. 아무 원도 반환하지 않을 것입니다.

**접촉** ***(Touch)***
도형이 다른 도형과 접하는지 검증합니다. 도형들이 최소한 포인트 1 개를 공유하지만 각 도형의 내부가 교차하지 않는 경우 1(참) 을 반환합니다. 3 번 원만 반환합니다.

**중첩** ***(Overlap)***
도형이 다른 도형과 중첩하는지 검증합니다. 도형들이 동일한 차원이며 공간을 공유하지만 서로를 완전히 담고 있지 않은 경우 1(참) 을 반환합니다. 2 번 원만 반환합니다.

**내부** ***(Are within)***
도형이 다른 도형 내부에 있는지 검증합니다. A 가 B 내부에 완전히 들어가 있는 경우 1(참) 을 반환합니다. 1 번 원만 반환합니다.

**공간교차** ***(Cross)***
입력 도형들이 전부는 아니지만 일부 내부 포인트를 공유하지만 입력 도형의 최고 차원보다 낮은 차원에서 실제로 가로지르는 경우 1(참) 을 반환합니다. 예를 들어, 폴리곤을 가로지르는 라인은 라인으로서 공간교차할 것입니다 (참). 라인을 가로지르는 라인은 포인트로서 공간교차할 것입니다 (참). 폴리곤을 가로지르는 폴리곤은 폴리곤으로서 가로지릅니다 (거짓). 앞의 그림에서, 아무 원도 반환하지 않을 것입니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Join to features in** | INPUT | [vector: geometry] | 입력 벡터 레이어. 산출 레이어는 이 레이어의 피처와 두 번째 레이어에 있는 일치하는 피처의 속성으로 이루어질 것입니다. |
| **Where the features** | PREDICATE | [enumeration] [list]<br>기본값: [0] | 소스 피처와 대상 피처를 결합시킬 수 있으려면 두 피처가 가져야 할 공간 관계의 유형입니다. 다음 가운데 하나 이상을 선택할 수 있습니다:<br>• 0 —교차 (intersect)<br>• 1 —포함 (contain)<br>• 2 —동등 (equal)<br>• 3 —접촉 (touch)<br>• 4 —중첩 (overlap)<br>• 5 —내부 (are within)<br>• 6 —공간교차 (cross)<br>조건을 하나 이상 선택한 경우, 적어도 하나는 (OR 연산자) 추출할 피처를 만족시켜야 합니다. |
| **By comparing to** | JOIN | [vector: geometry] | 결합 레이어입니다. 이 벡터 레이어의 피처가 공간 관계를 만족하는 경우 소스 레이어의 속성 테이블에 자신의 속성의 요약을 **추가** 할 것입니다. |
| **Fields to summarize (leave empty to use all fields)**<br>선택적 | JOIN_FIELDS | [tablefield: any] [list] | 결합 레이어에서 사용자가 추가하고자 하는 특정 필드를 선택합니다. 기본적으로 모든 필드를 추가합니다. |

다음 페이지에 계속

표 24.574 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Summaries to calculate (leave empty to use all fields)**<br>선택적 | SUMMARIES | [enumeration] [list]<br>기본값: [] | 각 입력 피처마다, 일치하는 피처의 결합된 필드 상에서 통계를 계산합니다. 다음 가운데 하나 이상을 선택할 수 있습니다:<br>• 0 —개수<br>• 1 —유일값<br>• 2 —최소값<br>• 3 —최대값<br>• 4 —범위 (range)<br>• 5 —합계<br>• 6 —평균<br>• 7 —중앙값<br>• 8 —표준 편차<br>• 9 —희소값 (minority)<br>• 10 —최빈값 (majority)<br>• 11 —제 1 사분위수<br>• 12 —제 3 사분위수<br>• 13 —사분위수의 범위<br>• 14 —비어 있는 값<br>• 15 —존재하는 값<br>• 16 —최단 거리<br>• 17 —최장 거리<br>• 18 —평균 거리 |
| **Discard records which could not be joined** | DISCARD_NONMATCHIN | [boolean]<br>기본값: False | 산출물에서 결합시킬 수 없는 입력 레이어 피처를 제거합니다. |
| **Joined layer** | OUTPUT | [same as input]<br>기본값: [Create temporary layer] | Specify the output vector layer for the join. *One of*:<br>• 임시 레이어 생성 (TEMPORARY_OUTPUT)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Joined layer** | OUTPUT | [same as input] | 결합으로부터 나온 요약된 속성을 가진 산출 벡터 레이어 |

### 파이썬 코드

**Algorithm ID**: `native:joinbylocationsummary`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 최근접으로 속성 결합하기

입력 벡터 레이어를 받아 속성 테이블에 필드를 추가한 새 벡터 레이어를 생성합니다. 두 번째 벡터 레이어에서 추가 속성 및 그 값을 가져옵니다. 각 레이어에서 최근접 피처를 찾아 피처를 결합합니다.

기본적으로 최근접 피처들만 결합하지만, k-최근접 이웃한 피처도 결합할 수 있습니다.

최대 거리를 지정한 경우, 그 거리보다 가까운 피처만 결합할 것입니다.

**경고**

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

**더 보기**

최근접 이웃 분석, 필드값으로 속성 결합하기, 위치로 속성 결합하기, 거리 매트릭스

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | 입력 레이어 |
| **Input layer 2** | `INPUT_2` | [vector: geometry] | 결합 레이어 |
| **Layer 2 fields to copy (leave empty to copy all fields)** | `FIELDS_TO_COPY` | [fields] | 복사할 결합 레이어 필드 (비어 있을 경우, 모든 필드를 복사할 것입니다.) |
| **Discard records which could not be joined** | `DIS-CARD_NONMATCHIN` | [boolean] 기본값: False | 산출물에서 결합시킬 수 없는 입력 레이어 레코드를 제거합니다. |
| **Joined field prefix** | `PREFIX` | [string] | 결합된 필드의 접두어 |
| **Maximum nearest neighbors** | `NEIGHBORS` | [numeric: integer] 기본값: 1 | 최근접 이웃의 최대 개수 |
| **Maximum distance** | `MAX_DISTANCE` | [numeric: double] | 최대 검색 거리 |

다음 페이지에 계속

표 24.576 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Joined layer**<br>선택적 | OUTPUT | [same as input]<br>기본값: [Create temporary layer] | Specify the vector layer containing the joined features. *One of*:<br>• 산출물 건너뛰기<br>• 임시 레이어 생성 (TEMPORARY_OUTPUT)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |
| **Unjoinable features from first layer** | NON_MATCHING | [same as input]<br>기본값: [Skip output] | Specify the vector layer containing the features that could not be joined. *One of*:<br>• 산출물 건너뛰기<br>• 임시 레이어 생성 (TEMPORARY_OUTPUT)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Joined layer** | OUTPUT | [same as input] | 산출 결합 레이어 |
| **Unjoinable features from first layer** | NON_MATCHING | [same as input] | 첫 번째 레이어의 피처 가운데 결합 레이어의 어느 피처와도 결합할 수 없는 피처를 담고 있는 레이어 |
| **Number of joined features from input table** | JOINED_COUNT | [numeric: integer] | 입력 테이블에서 나온 결합된 피처의 개수 |
| **Number of unjoinable features from input table** | UNJOINABLE_COUNT | [numeric: integer] | 입력 테이블에서 나온 결합할 수 없는 피처의 개수 |

### 파이썬 코드

**알고리즘 ID**: `native:joinbynearest`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

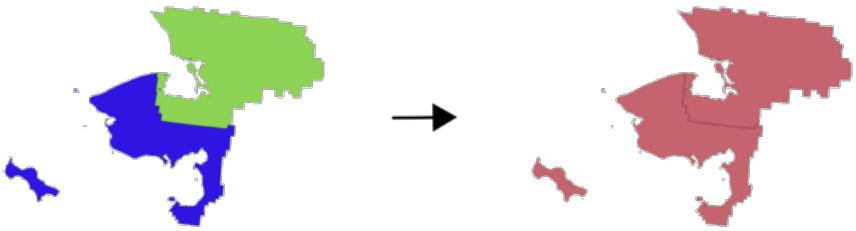
### 벡터 레이어 병합하기

**동일한 도형** 유형의 벡터 레이어 여러 개를 단일 레이어로 병합합니다.

The attribute table of the resulting layer will contain the fields from all input layers. If fields with the same name but different types are found then the exported field will be automatically converted into a string type field. Optionally, new fields storing the original layer name and source can be added.

입력 레이어 가운데 Z 또는 M 값을 가진 레이어가 하나라도 있을 경우, 산출 레이어도 그 값을 가질 것입니다. 이와 비슷하게, 입력 레이어 가운데 하나라도 다중 부분 (multi-part) 유형인 경우 산출 레이어도 다중 부분 레이어일 것입니다.

병합된 레이어의 대상 좌표계도 부가적으로 설정할 수 있습니다. 좌표계를 설정하지 않은 경우, 첫 번째 입력 레이어의 좌표계를 사용할 것입니다. 모든 레이어를 이 좌표계와 일치하도록 재투영할 것입니다.



**기본 메뉴**: *Vector → Data Management Tools*

**경고**

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

**더 보기**

벡터 레이어 분할하기

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input Layers** | `LAYERS` | [vector: any] [list] | 단일 레이어로 병합할 레이어들. 동일한 도형 유형 레이어이어야 합니다. |
| **Destination CRS** 선택적 | `CRS` | [crs] | 산출 레이어의 좌표계를 선택합니다. 좌표계를 지정하지 않은 경우, 첫 번째 입력 레이어의 좌표계를 사용할 것입니다. |
| **Add source layer information (layer name and path)** | `ADD_SOURCE_FIEI` | [boolean] 기본값: True | Add fields storing the original layer name and path |

다음 페이지에 계속

표 24.578 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Merged** | `OUTPUT` | [same as input]<br>기본값: `[Create temporary layer]` | Specify the output vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Merged** | `OUTPUT` | [same as input] | 입력 레이어들의 모든 피처와 속성을 담고 있는 산출 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:mergevectorlayers`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 표현식으로 정렬하기

표현식에 따라 벡터 레이어를 정렬합니다. 표현식에 따라 피처 인덱스를 변경한다는 뜻입니다.

주의하십시오: 일부 제공자의 경우 기대와는 다르게 작동할 수도 있습니다. 매번 순서가 유지되지 않을 수도 있습니다.

**경고**

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input Layer** | `INPUT` | [vector: any] | 정렬할 입력 벡터 레이어 |
| **Expression** | `EXPRESSION` | [expression] | 정렬 작업에 사용할 표현식 |
| **Sort ascending** | `ASCENDING` | [boolean]<br>기본값: True | 이 파라미터를 활성화하면 벡터 레이어를 작은 값에서 큰 값으로 정렬할 것입니다. |
| **Sort nulls first** | `NULLS_FIRST` | [boolean]<br>기본값: False | 이 파라미터를 활성화하면 NULL 값을 처음에 배치합니다. |
| **Ordered** | `OUTPUT` | [same as input]<br>기본값: `[Create temporary layer]` | Specify the output vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Ordered** | `OUTPUT` | [same as input] | 산출 (정렬된) 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:orderbyexpression`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 셰이프파일 수리하기

SHX 파일을 (재) 생성해서 망가진 셰이프파일 데이터셋을 수리합니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input Shapefile** | `INPUT` | [file] | 없거나 깨진 SHX 파일을 가진 ESRI 셰이프파일 데이터셋을 가리키는 전체 경로 |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Repaired layer** | `OUTPUT` | [vector: geometry] | SHX 파일을 수리한 입력 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:repairshapefile`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 레이어 재투영하기

벡터 레이어를 다른 좌표계로 재투영합니다. 재투영된 레이어는 입력 레이어와 동일한 피처 및 속성을 가질 것입니다.

포인트, 라인 및 폴리곤 피처를제자리 피처 수정 할 수 있습니다.

**더 보기**

투영체 할당하기, *Define projection*, 투영체 찾기

### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input Layer** | INPUT | [vector: geometry] | 재투영할 입력 벡터 레이어 |
| **Target CRS** | TARGET_CRS | [crs]<br>기본값: EPSG:4326 - WGS84 | 대상 좌표계 |
| **만곡 도형을 직선 선분으로 변환** | CONVERT_CURVED_GEO | [boolean]<br>기본값: False | 이 옵션을 체크하면, 처리 과정에서 잠재적인 왜곡 문제를 피하기 위해 만곡 도형을 직선 선분들로 변환시킬 것입니다. |
| **Reprojected** | OUTPUT | [same as input]<br>기본값: [Create temporary layer] | Specify the output vector layer. *One of*:<br>• 임시 레이어 생성 (TEMPORARY_OUTPUT)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

#### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Coordinate Operation**<br>선택적 | OPERATION | [string] | 언제나 현재 프로젝트의 변형 설정을 사용하도록 강제하는 대신, 특정 재투영 작업을 위해 사용할 방법을 지정합니다. 특정 레이어를 재투영하는 데 정확한 변형 경로 (pipeline) 를 제어해야만 하는 경우 유용한 옵션입니다. PROJ 6 이상 버전이 필요합니다.<br>기준점 *(datum)* 변환 을 읽어보십시오. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Reprojected** | OUTPUT | [same as input] | 산출 (재투영된) 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:reprojectlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 벡터 피처를 파일로 저장

이 알고리즘은 벡터 피처를 지정한 파일 데이터셋으로 저장합니다.

For dataset formats supporting layers, an optional layer name parameter can be used to specify a custom string. Optional GDAL-defined dataset and layer options can be specified. For more information on this, read the online GDAL documentation on the format.

### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Vector features** | `INPUT` | [vector: any] | 입력 벡터 레이어 |
| **Saved features** | `OUTPUT` | [same as input]<br>기본값: `[Save to temporary file]` | Specify the file to save the features to. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

#### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Layer name**<br>선택적 | `LAYER_NAME` | [string] | 산출 레이어에 사용할 이름 |
| **GDAL dataset options**<br>선택적 | `DATASOURCE_OPTIONS` | [string] | 산출 포맷의 GDAL 데이터셋 생성 옵션입니다. 개별 옵션을 쌍반점으로 구분하십시오. |
| **GDAL layer options**<br>선택적 | `LAYER_OPTIONS` | [string] | 산출 포맷의 GDAL 레이어 생성 옵션입니다. 개별 옵션을 쌍반점으로 구분하십시오. |

다음 페이지에 계속

표 24.588 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **기존 파일 처리 시 취할 액션** | ACTION_ON_EXISTIN | [enumeration] 기본값: 0 | 기존 피처들을 관리할 방법입니다. 다음 메소드들 가운데 하나를 선택할 수 있습니다:<br>• 0 —Create or overwrite file<br>• 1 —Create or overwrite layer<br>• 2 —Append features to existing layer, but do not create new fields<br>• 3 —Append features to existing layer, and create new fields if needed |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Saved features** | OUTPUT | [same as input] | 저장된 피처를 담은 벡터 레이어 |
| **File name and path** | FILE_PATH | [string] | 산출 파일명과 경로 |
| **Layer name** | LAYER_NAME | [string] | (지정한 경우) 레이어명 |

### 파이썬 코드

**알고리즘 ID**: `native:savefeatures`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 레이어 인코딩 설정

이 알고리즘은 레이어의 속성을 읽어오는 데 쓰이는 인코딩을 설정합니다. 레이어에 항구적인 변경 사항을 남기지는 않고, 현재 세션 동안 어떻게 레이어를 읽어올지에 대해서만 영향을 줍니다.

**참고**

인코딩 변경을 지원하는 것은 일부 벡터 레이어 데이터소스뿐입니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Saved features** | `INPUT` | [vector: geometry] | 인코딩을 설정할 벡터 레이어 |
| **Encoding** | `ENCODING` | [string] | 현재 QGIS 세션에서 레이어에 할당할 텍스트 인코딩 |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output layer** | `OUTPUT` | [same as input] | 설정한 인코딩을 가진 입력 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:setlayerencoding`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 문자로 피처 분할하기

지정한 문자 위치에서 필드 값을 분할해서 피처를 여러 산출 피처로 분할합니다. 예를 들면, 레이어가 단일 필드에 쉼표로 구분된 여러 값을 가지고 있는 피처를 담고 있는 경우 이 알고리즘을 사용해서 해당 값들을 분할시켜 산출 피처 여러 개를 생성할 수 있습니다. 산출물에서 도형 및 기타 속성은 그대로 유지됩니다. 유연성을 위해 정규 표현식을 구분자 문자열로 사용할 수도 있습니다.

☑ 포인트, 라인 및 폴리곤 피처를제자리 피처 수정 할 수 있습니다.

**⚠ 경고**

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input Layer** | `INPUT` | [vector: any] | 입력 벡터 레이어 |
| **Split using values in the field** | `FIELD` | [tablefield: any] | 분할 작업에 사용할 필드 |
| **Split value using character** | `CHAR` | [string] | 분할 작업에 사용할 문자 |
| **Use regular expression separator** | `REGEX` | [boolean]<br>기본값: False | |
| **Split** | `OUTPUT` | [same as input]<br>기본값: `Create temporary layer` | Specify the output vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Split** | `OUTPUT` | [same as input] | 산출 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:splitfeaturesbycharacter`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 벡터 레이어 분할하기

입력 레이어와 속성을 바탕으로 산출물 폴더에 벡터 집합을 생성합니다. 산출물 폴더는 원하는 필드에서 발견된 유일 값 개수만큼 많은 레이어를 담을 것입니다.

생성되는 파일의 개수는 지정한 속성에서 발견된 서로 다른 값들의 개수와 동일합니다.

병합 과는 정반대 작업입니다.

**기본 메뉴**: *Vector → Data Management Tools*

**더 보기**

벡터 레이어 병합하기

### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input Layer** | `INPUT` | [vector: any] | 입력 벡터 레이어 |
| **Unique ID field** | `FIELD` | [tablefield: any] | 분할 작업에 사용할 필드 |
| **Output directory** | `OUTPUT` | [folder]<br>기본값: `[Save to temporary folder]` | Specify the directory for the output layers. *One of*:<br>• 임시 디렉터리에 저장<br>• 디렉터리에 저장 |

#### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output file type**<br>선택적 | `FILE_TYPE` | [enumeration]<br>기본값: 대화창의 `gpkg` | 산출 파일의 확장자를 선택합니다. 지정하지 않거나 무결하지 않은 경우, 공간 처리 프레임워크 설정의"기본 산출 벡터 레이어 확장자"가 산출 파일 포맷이 될 것입니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output directory** | `OUTPUT` | [folder] | 산출 레이어를 저장할 디렉터리 |
| **Output layers** | `OUTPUT_LAYERS` | [same as input] [list] | 분할로 산출된 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:splitvectorlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

#### 테이블 절단하기

레이어 내부에 있는 모든 피처를 삭제해서 레이어를 절단 (truncate) 합니다.

> **⚠ 경고**
>
> 이 알고리즘은 제자리에서 레이어를 수정하기 때문에, 삭제한 피처를 복구할 수 없습니다!

#### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input Layer** | `INPUT` | [vector: any] | 입력 벡터 레이어 |

#### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Truncated layer** | `OUTPUT` | [folder] | The input layer, all features deleted |

#### 파이썬 코드

**알고리즘 ID**: `native:truncatetable`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 24.1.26 벡터 도형

#### 도형 속성 추가하기

벡터 레이어에 있는 피처의 도형 속성을 계산해서 산출 레이어에 포함시킵니다.

입력 레이어와 동일한 내용에 선택한 좌표계를 바탕으로 측정한 기하학적 정보를 담고 있는 속성을 추가한 새 벡터 레이어를 생성합니다.

테이블에 추가되는 속성은 도형 유형과 입력 레이어의 차원에 따라 달라집니다:

- **포인트** 레이어: X (`xcoord`), Y (`ycoord`), Z (`zcoord`) 좌표 그리고/또는 M 값 (`mvalue`)
- **라인** 레이어: `length`, 그리고 라인스트링과 복합곡선 (CompoundCurve) 도형 유형의 경우 `sinuosity` 피처와 직선 거리 (`straightdis`)
- **폴리곤** 레이어: `perimeter` 및 `area`

참고

This algorithm uses ellipsoid based measurements and respects the current *ellipsoid settings*.

기본 메뉴: *Vector → Geometry Tools*

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | 입력 벡터 레이어 |
| **Calculate using** | `CALC_METHOD` | [enumeration] 기본값: 0 | 기하학적 속성에 사용할 계산용 파라미터입니다. 다음 가운데 하나를 선택할 수 있습니다:<br>• 0 —레이어 좌표계<br>• 1 —프로젝트 좌표계<br>• 2 —타원체 |
| **Added geom info** | `OUTPUT` | [same as input] 기본값: `[Create temporary layer]` | Specify the output (input copy with geometry) layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Added geom info** | `OUTPUT` | [same as input] | 입력 벡터 레이어에 도형 필드를 추가한 복사본 |

### 파이썬 코드

**Algorithm ID**: `native:exportaddgeometrycolumns`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 아핀 (affine) 변형

레이어 도형에 아핀 변형을 적용합니다. 아핀 변형은 좌표 이동 (translation), 축척 작업 및 기울이기를 포함할 수 있습니다. 이 작업은 축척, 기울이기, 그리고 좌표 이동 순서대로 이루어집니다.

Z 및 M 값도 (존재할 경우) 이동 및 축척 조정할 수 있습니다.

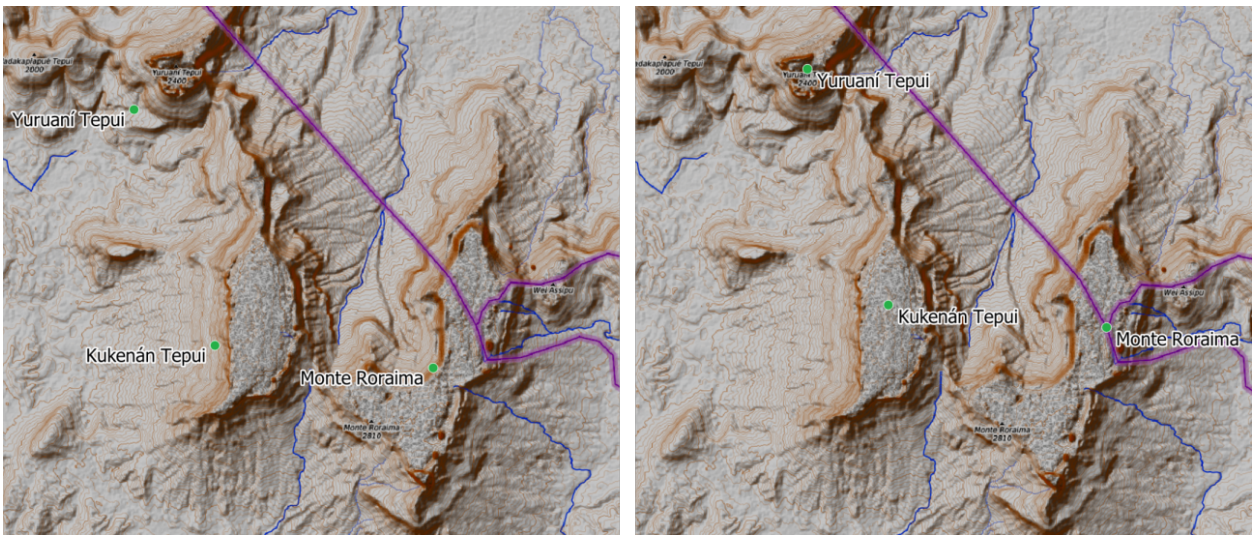



그림 24.89: 벡터 포인트 레이어 (녹색 점) 의 아핀 변형 (좌표 이동) 전 (왼쪽) 과 후 (오른쪽).

포인트, 라인 및 폴리곤 피처를제자리 피처 수정 할 수 있습니다.

**더 보기**

이동시키기 *(Translate)*

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | 입력 벡터 레이어 |
| **Translation (x-axis)** | `DELTA_X` | [numeric: double]<br>기본값: 0.0 | X 축에 적용할 변위 (displacement) |
| **Translation (y-axis)** | `DELTA_Y` | [numeric: double]<br>기본값: 0.0 | Y 축에 적용할 변위 (displacement) |
| **Translation (z-axis)** | `DELTA_Z` | [numeric: double]<br>기본값: 0.0 | Z 축에 적용할 변위 (displacement) |
| **Translation (m-values)** | `DELTA_M` | [numeric: double]<br>기본값: 0.0 | M 값에 적용할 오프셋 |

다음 페이지에 계속

표 24.601 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Scale factor (x-axis)** | SCALE_X | [numeric: double]<br>기본값: 1.0 | X 축에 적용할 척도값 (확장 또는 수축) |
| **Scale factor (y-axis)** | SCALE_Y | [numeric: double]<br>기본값: 1.0 | Y 축에 적용할 척도값 (확장 또는 수축) |
| **Scale factor (z-axis)** | SCALE_Z | [numeric: double]<br>기본값: 1.0 | Z 축에 적용할 척도값 (확장 또는 수축) |
| **Scale factor (m-values)** | SCALE_M | [numeric: double]<br>기본값: 1.0 | M 값에 적용할 척도값 (확장 또는 수축) |
| **Rotation around z-axis (degrees counter-clockwise)** | ROTATION_Z | [numeric: double]<br>기본값: 0.0 | 기울기의 도 단위 각도 |
| **Transformed** | OUTPUT | [same as input]<br>기본값: [Create temporary layer] | Specify the output vector layer. *One of*:<br>• 임시 레이어 생성 (TEMPORARY_OUTPUT)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Transformed** | OUTPUT | [same as input] | 산출 (변형된) 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: native:affinetransform

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 집계

벡터 또는 테이블 레이어를 받아 `group by` 표현식을 바탕으로 피처를 집계 (aggregate) 해서 새 레이어를 생성합니다.

`group by` 표현식이 동일한 값을 반환하는 피처들을 함께 그룹화합니다.

`group by` 파라미터에 상수값을, 예를 들어 NULL 값을 사용해서 모든 소스 피처를 함께 그룹화할 수 있습니다.

Array("Field1", "Field2") 같은 배열 함수를 사용해서 피처를 여러 필드를 통해 그룹화할 수도 있습니다.

각 그룹마다 (도형이 있는 경우) 도형을 다중 부분 도형 하나로 결합시킵니다. 지정한 각 집계 정의에 따라 계산되는 산출 속성이 달라집니다.

이 알고리즘은 QGIS 표현식 엔진의 기본집계 함수 를 사용할 수 있습니다.

**참고**

This algorithm uses ellipsoid based measurements and respects the current *ellipsoid settings*.

**더 보기**

도형 모으기, 융해

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: any] | 입력 벡터 레이어 |
| **Group by expression** | `GROUP_BY` | [tablefield: any]<br>기본값: 'NULL' | 그룹화 필드를 선택합니다. *NULL* 값을 설정하면 모든 피처를 하나의 그룹으로 묶을 것입니다. |

다음 페이지에 계속

표 24.603 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Aggregates** | AGGREGATES | [list] | 산출 레이어 필드 정의의 목록입니다. 다음은 필드 정의의 예시입니다: *{'aggregate': 'sum', 'delimiter': ',', 'input': '$area', 'length': 10, 'name': 'totarea', 'precision': 0, 'type': 6}* 이 목록은 기본적으로 입력 레이어의 모든 필드를 담고 있습니다. GUI 환경에서, 이 필드들과 그 정의들을 편집할 수 있으며, 다음 작업도 할 수 있습니다:<br>•  버튼을 클릭해서 새 필드를 추가할 수 있습니다.<br>•  버튼을 클릭해서 선택한 필드를 삭제할 수 있습니다.<br>•  및  버튼을 클릭해서 필드 순서를 변경할 수 있습니다.<br>•  버튼을 클릭해서 (입력 레이어 필드의) 기본값으로 리셋할 수 있습니다.<br>사용자가 정보를 검색하고 싶은 각 필드마다 다음을 정의해줘야 합니다:<br>***Input expression* (`input`) [expression]**<br>입력 레이어에서 나온 필드 또는 표현식<br>***Aggregate function* (`aggregate`) [enumeration]**<br>Function to use on the input expression to return the aggregated value. These are mainly the 집계 함수, plus dedicated "first_value"and "last_value" functions for ordered list items.<br>기본값: *concatenate* (문자열 데이터 유형), *sum* (숫자 데이터 유형)<br>***Delimiter* (`delimiter`) [string]**<br>예를 들어 `concatenate` 함수로 집계된 값들을 구분하기 위한 텍스트 문자열<br>기본값: ,<br>***Output field name* (`name`) [string]**<br>산출 레이어의 집계된 필드의 이름입니다. 기본적으로 입력 필드명을 그대로 유지합니다.<br>***Type* (`type`) [enumeration]**<br>Data type of the output field. Available types may not be compatible with the output layer provider. One of:<br>⚠ 주의<br>For certain field types, e.g. lists, an extra `sub_type` parameter helps refine the specific type of the data. It is automatically set in the GUI but may be needed if you're running the algorithm in Python or from the command line.<br>• 1 — 불 (Boolean) |

표 24.603 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Load fields from layer** | GUI 전용 | [vector: any] | 또다른 레이어에서 필드를 불러와 집계에 사용할 수 있습니다. |
| **Aggregated** | `OUTPUT` | [same as input]<br>기본값: `[Create temporary layer]` | Specify the output (aggregate) layer *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Aggregated** | `OUTPUT` | [same as input] | 집계값을 가진 다중 도형 (multigeometry) 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:aggregate`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 경계

입력 도형들의 닫힌 조합 경계선 (combinatorial boundary) 을 반환합니다. (예: 도형의 위상 경계)

폴리곤 및 라인 레이어 전용 알고리즘입니다.

**폴리곤 도형** 의 경우, 경계는 폴리곤의 고리 (ring) 를 형성하는 모든 라인으로 이루어져 있습니다.

**라인 도형** 의 경우, 경계는 라인의 종단점입니다.

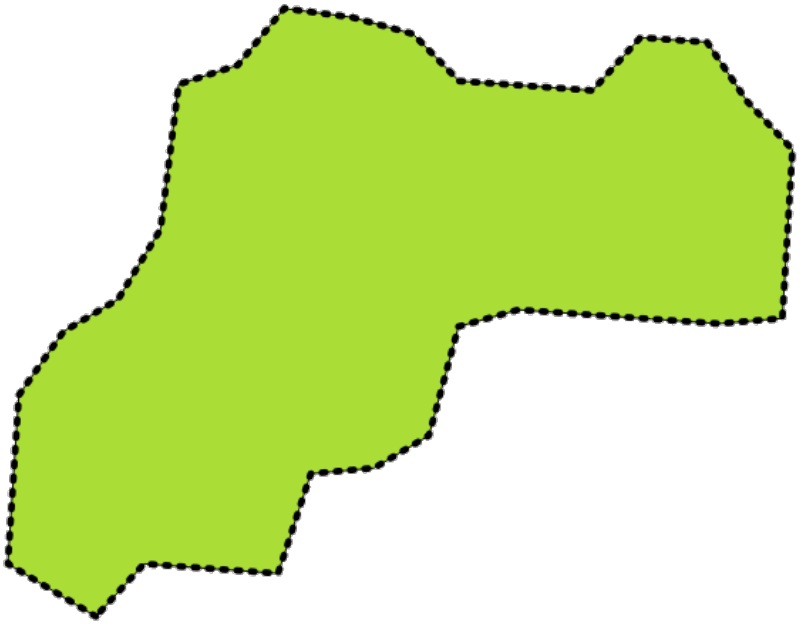


그림 24.90: 소스 폴리곤 레이어의 경계 (검은 점선)

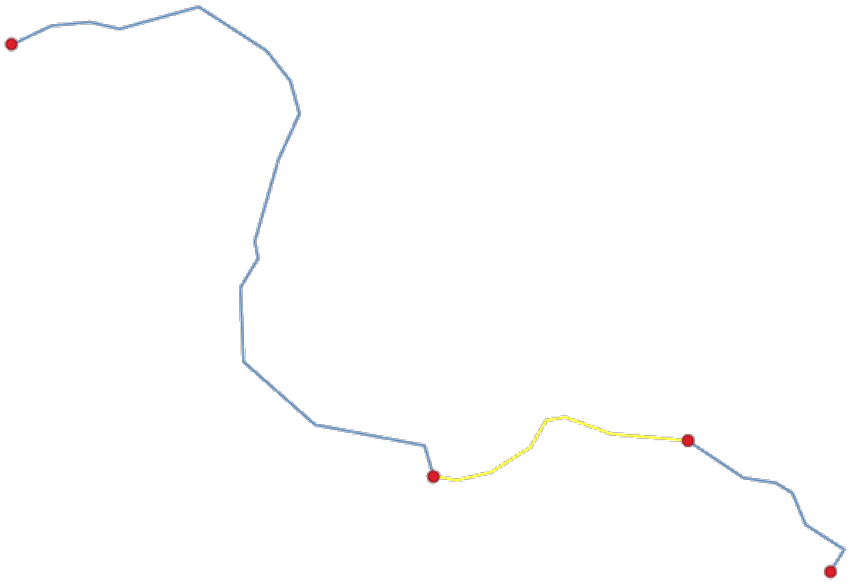


그림 24.91: 라인 레이어의 경계 (빨간 포인트), 노란색은 선택한 피처

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: line, polygon] | 입력 라인 또는 폴리곤 벡터 레이어 |
| **Boundary** | `OUTPUT` | [vector: point, line] 기본값: `[Create temporary layer]` | Specify the output (boundary) layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Boundary** | `OUTPUT` | [vector: point, line] | 입력 레이어에서 나온 경계 (라인의 경우 포인트, 폴리곤의 경우 라인) |

### 파이썬 코드

**알고리즘 ID**: `native:boundary`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 경계 상자

입력 레이어에 있는 각 피처의 경계 상자 (envelope) 를 계산합니다. 폴리곤 및 라인 도형을 지원합니다.

☑ 폴리곤 피처를제자리 피처 수정 할 수 있습니다.

**더 보기**

최소 경계 도형

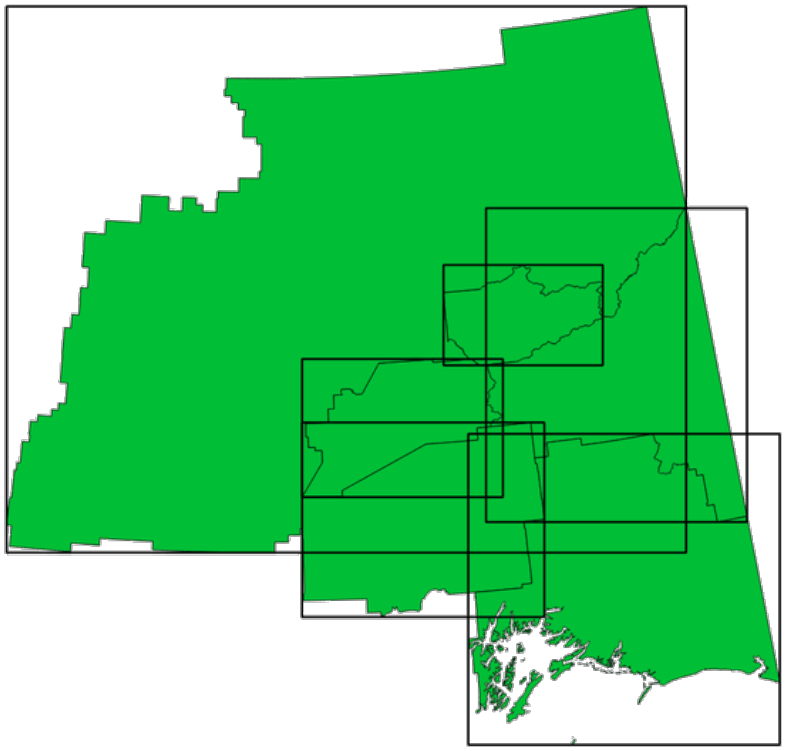


그림 24.92: 각 폴리곤 피처의 경계 상자를 표현하는 검은 선

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: line, polygon] | 입력 라인 또는 폴리곤 벡터 레이어 |
| **Bounds** | `OUTPUT` | [vector: polygon] 기본값: `[Create temporary layer]` | Specify the output (bounding box) layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Bounds** | `OUTPUT` | [vector: polygon] | 입력 레이어의 경계 상자입니다. 산출 레이어는 입력물의 속성 이외에도, 생성된 폴리곤의 `width`, `height`, `area` 및 `perimeter` 필드들을 담고 있습니다. |



### 파이썬 코드

**알고리즘 ID:** `native:boundingboxes`

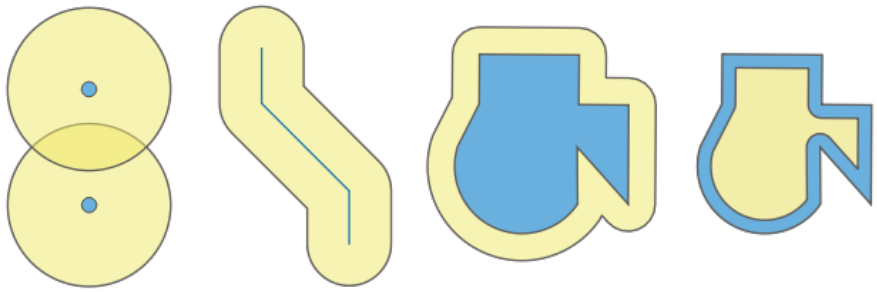


그림 24.93: 버퍼 (노란색) –양의 버퍼를 가진 포인트, 라인, 폴리곤 | 음의 버퍼를 가진 폴리곤

**경고**

This algorithm may drop existing primary keys or FID values and regenerate them in output layers, depending on the input parameters.

**더 보기**

가변 거리 버퍼, 다중 고리 버퍼 *(상수 거리)*, 변동 너비 버퍼 *(M 값으로)*

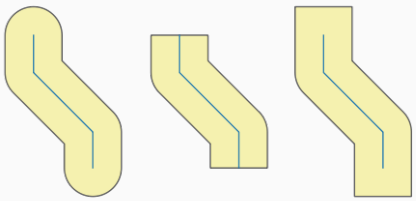
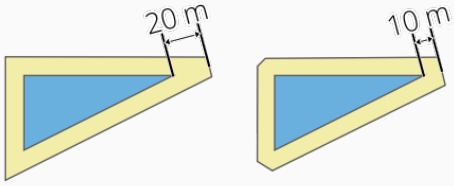
### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | 입력 벡터 레이어 |
| **Distance** | `DISTANCE` | [numeric: double]<br>기본값: 10.0 | Buffer distance (from the boundary of each feature). You can use the Data Defined button on the right to choose a field from which the radius will be calculated. This way you can have different radius for each feature. |
| **Segments** | `SEGMENTS` | [numeric: integer]<br>기본값: 5 | 둥글린 (rounded) 오프셋을 생성하는 경우 사분원에 가깝게 만드는 데 사용할 라인 선분의 개수를 제어합니다. |

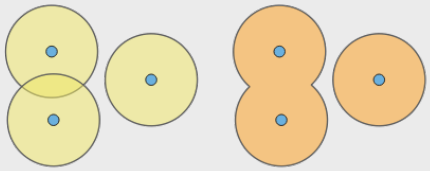
다음 페이지에 계속

표 24.609 – 이전 페이지에서 계속

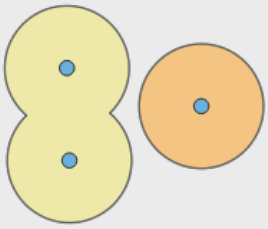
| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **End cap style** | `END_CAP_STYLE` | [enumeration]<br>기본값: 0 | 버퍼에서 라인의 끝을 어떻게 처리할지 제어합니다. 다음 가운데 하나로 지정할 수 있습니다:<br>• 0 —둥글림 (Round)<br>• 1 —편평 (Flat)<br>• 2 —정사각형 (Square)<br>  <br>그림 24.94: 둥글림, 편평 및 정사각형 캡 스타일 |
| **Join style** | `JOIN_STYLE` | [enumeration]<br>기본값: 0 | 라인에 있는 모서리를 오프셋시킬 때 둥글림, 마이터, 또는 베벨 결합을 사용할지 여부를 지정합니다. 다음 옵션 가운데 하나를 선택할 수 있습니다:<br>• 0 —둥글림 (Round)<br>• 1 —마이터 (Miter)<br>• 2 —베벨 (Bevel)<br>  <br>그림 24.95: 둥글림, 마이터, 베벨 결합 스타일 |
| **Miter limit** | `MITER_LIMIT` | [numeric: double]<br>기본값: 2.0 | 마이터 결합 스타일에만 적용할 수 있으며, 마이터 결합 생성시 사용할 오프셋 도형으로부터의 최장 거리를 오프셋 거리의 인자로 설정합니다. (최소값은 1.0 입니다.)<br>  <br>그림 24.96: 제한값이 2 인 10m 버퍼와 제한값이 1 인 10m 버퍼 |

다음 페이지에 계속

표 24.609 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Dissolve result** | `DISSOLVE` | [boolean]<br>기본값: False | 최종 버퍼를 융해 (dissolve) 합니다. `True` 로 설정하면 (활성화하면) 중첩하는 버퍼들을 단일 다중부분 피처로 융해 (결합) 할 것입니다.<br>  <br>그림 24.97: 표준 버퍼 (단일 부분 피처 3 개 - 왼쪽) | 융해된 버퍼 (2 개의 부분을 가진 다중부분 피처 1 개 - 오른쪽) |
| **Buffered** | `OUTPUT` | [vector: polygon]<br>기본값: `[Create temporary layer]` | Specify the output (buffer) layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Keep disjoint features separate** | `SEPARATE_DISJOINT` | [boolean]<br>기본값: False | `True` (체크) 이며 dissolved 파라미터를 체크한 경우, 중첩하거나 접하지 않는 피처들을 (다중 부분 피처의 부분들로가 아닌) 개별 피처들로 내보낼 것입니다.<br>  <br>그림 24.98: 단일 부분 피처 2 개를 산출 |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Buffered** | `OUTPUT` | [vector: polygon] | 산출 (버퍼) 폴리곤 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:buffer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 중심

입력 레이어의 도형의 중심 (centroid) 을 표현하는 포인트를 담은 새 포인트 레이어를 생성합니다.

중심이란 피처의 (모든 부분의) 무게중심을 표현하는 단일 포인트를 말합니다. 따라서 피처 경계선 바깥에 위치할 수 있지만, 피처의 각 부분에 포인트가 위치할 수도 있습니다.

산출 레이어에 있는 포인트의 속성은 원본 피처의 속성과 동일합니다.

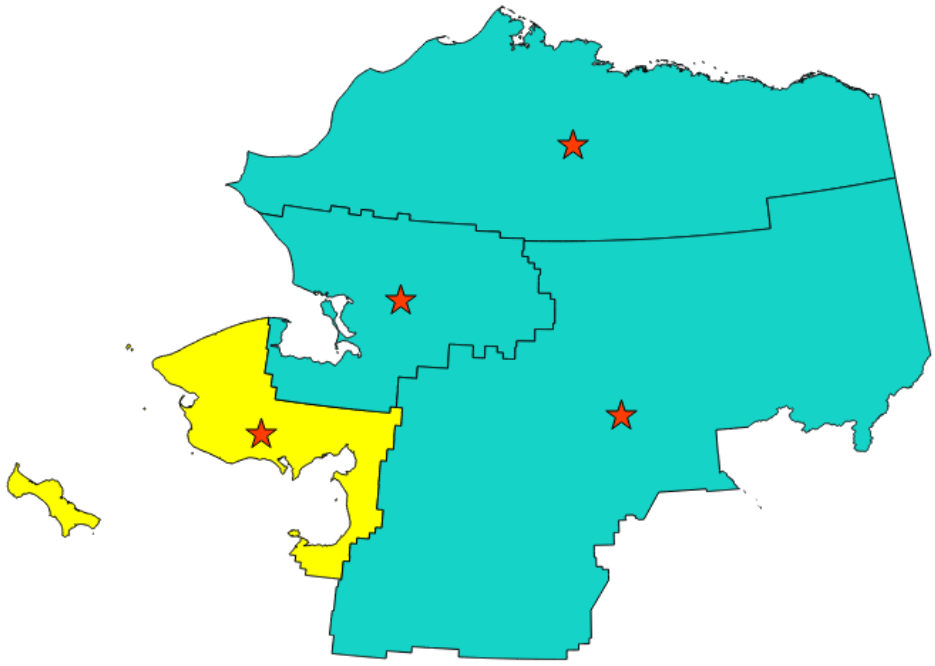


그림 24.99: 입력 레이어의 피처의 중심을 표현하는 빨간색 별

포인트 피처를제자리 피처 수정 할 수 있습니다.

기본 메뉴: *Vector → Geometry Tools*

**경고**

This algorithm may drop existing primary keys or FID values and regenerate them in output layers, depending on the input parameters.

**더 보기**

표면에 포인트 생성하기

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | 입력 벡터 레이어 |
| **Create centroid for each part** | `ALL_PARTS` | [boolean]<br>기본값: False | `True` 로 설정하면 (활성화하면) 도형의 각 부분별로 중심을 생성할 것입니다. |
| **Centroids** | `OUTPUT` | [vector: point]<br>기본값: `[Create temporary layer]` | Specify the output (centroid) layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Centroids** | `OUTPUT` | [vector: point] | 산출 포인트 벡터 레이어 (중심) |

### 파이썬 코드

**알고리즘 ID**: `native:centroids`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 무결성 점검하기

벡터 레이어의 도형에 대해 무결성 (validity) 점검을 수행합니다.

도형을 세 그룹 (무결함, 무결하지 않음 및 오류) 으로 분류해서 각 그룹별 피처를 담은 벡터 레이어를 생성합니다:

- **무결한 산출물** 레이어는 (위상 오류가 없는) 무결한 피처만 담고 있습니다.
- **무결하지 않은 산출물** 레이어는 알고리즘이 발견한 무결하지 않은 피처를 모두 담고 있습니다.
- **오류 산출물** 레이어는 무결하지 않은 피처를 발견한 위치를 가리키는 포인트 레이어입니다.

생성된 레이어들의 속성 테이블은 몇몇 (**오류** 레이어의 경우"message", **무결하지 않은** 레이어의 경우"FID"및"_errors", 그리고 **무결한** 레이어의 경우"FID"만) 추가 정보를 담을 것입니다:

생성된 각 레이어의 속성 테이블은 몇몇 (발견된 오류 개수 및 오류 유형) 추가 정보를 담을 것입니다:

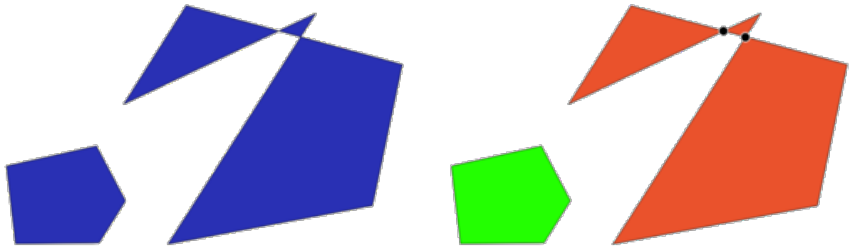


그림 24.100: 왼쪽: 입력 레이어, 오른쪽: 무결한 레이어 (녹색), 무결하지 않은 레이어 (주황색)

**기본 메뉴**: *Vector → Geometry Tools*

**더 보기**

*Check Geometry*, 도형 고치기, *Validate coverage*

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT_LAYER` | [vector: geometry] | 입력 벡터 레이어 |
| **Method** | `METHOD` | [enumeration] 기본값: 2 | 무결성을 검증하기 위해 사용하는 메소드입니다. 다음 옵션 가운데 선택할 수 있습니다:<br>• 0: 디지타이즈 작업 설정에서 선택한 메소드<br>• 1: QGIS<br>• 2: GEOS |
| **Ignore ring self intersection** | `IGNORE_RING_SELF_` | [boolean] 기본값: False | 무결성 점검 작업시 자체 교차 (self-intersect) 하는 고리를 무시합니다. |

다음 페이지에 계속

표 24.614 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Valid output** | VALID_OUTPUT | [same as input]<br>기본값: [Create temporary layer] | Specify the vector layer to contain a copy of the valid features of the source layer. *One of*:<br>• 산출물 건너뛰기<br>• 임시 레이어 생성 (TEMPORARY_OUTPUT)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |
| **Invalid output** | INVALID_OUTPUT | [same as input]<br>기본값: [Create temporary layer] | Vector layer containing copy of the invalid features of the source layer with the field _errors listing the summary of the error(s) found. *One of*:<br>• 산출물 건너뛰기<br>• 임시 레이어 생성 (TEMPORARY_OUTPUT)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |
| **Error output** | ERROR_OUTPUT | [vector: point]<br>기본값: [Create temporary layer] | Point layer of the exact position of the validity problems detected with the message field describing the error(s) found. *One of*:<br>• 산출물 건너뛰기<br>• 임시 레이어 생성 (TEMPORARY_OUTPUT)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Count of errors** | ERROR_COUNT | [numeric: integer] | 오류를 초래한 도형의 개수 |
| **Error output** | ERROR_OUTPUT | [vector: point] | 발견한 오류 (들) 를 서술하는 message 필드와 함께 감지된 무결성 문제점의 정확한 위치를 담고 있는 포인트 레이어 |
| **Count of invalid features** | INVALID_COUNT | [numeric: integer] | 무결하지 않은 도형의 개수 |
| **Invalid output** | INVALID_OUTPUT | [same as input] | 발견한 오류 (들) 의 요약 목록을 담은 _errors 필드와 함께 소스 레이어의 무결하지 않은 피처의 복사본을 담고 있는 벡터 레이어 |

다음 페이지에 계속

표 24.615 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Count of valid features** | `VALID_COUNT` | [numeric: integer] | 무결한 도형의 개수 |
| **Valid output** | `VALID_OUTPUT` | [same as input] | 소스 레이어의 무결한 피처의 복사본을 담고 있는 벡터 레이어 |

### 파이썬 코드

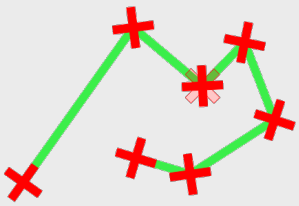
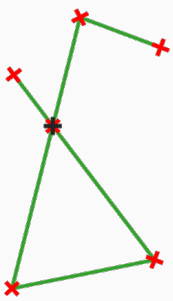
**Algorithm ID**: `native:checkvalidity`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

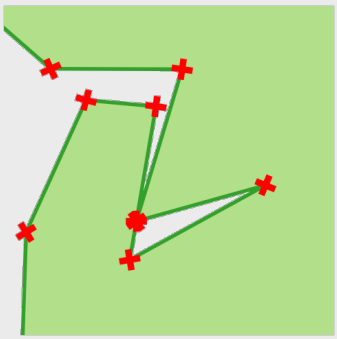
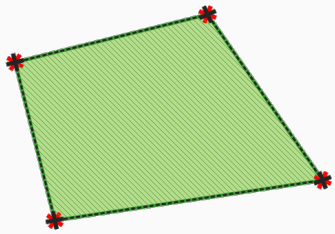
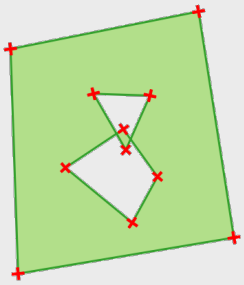
### 오류 메시지의 유형 및 그 의미

표 24.616: Some examples of geometry check failures in QGIS

| Geometry context | GEOS validation and error message | QGIS validation and error message |
|---|---|---|
| **Consecutive points on a line have the same coordinates**  | *Repeated point* | *Line a contains x duplicate node(s) at b* |
| **Segments of a line intersect each other**  | | *Segments a and b of line c intersect at d* |

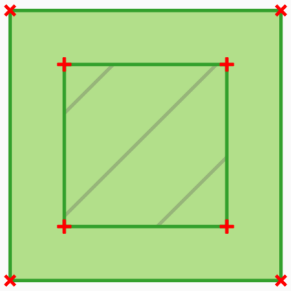
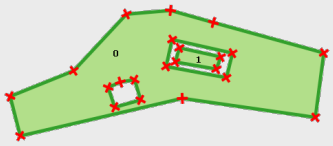
다음 페이지에 계속

표 24.616 – 이전 페이지에서 계속

| Geometry context | GEOS validation and error message | QGIS validation and error message |
|---|---|---|
| **Polygon geometry touches itself and generates a ring**  | ✗ *Ring self-intersection* | ✗ *Ring self-intersection* |
| **Two rings (exterior or interior) of a polygon geometry are identical**  | ✗ *Duplicate rings* | |
| **Geometry touches itself**  | ✗ *Self-intersection* | |

다음 페이지에 계속

표 24.616 – 이전 페이지에서 계속

| Geometry context | GEOS validation and error message | QGIS validation and error message |
|---|---|---|
| **A polygon geometry is on top of another polygon geometry**  | ✗ *Nested shell* | |
| **Part of a MultiPolygon geometry is within a hole of a MultiPolygon geometry**  | | ✗ *Polygon a lies inside polygon b* |
| Point geometry does not have a proper coordinate pair. The coordinate pair does not contain a latitude value and a longitude value in that order. | ✗ *Invalid coordinate* | |

### 도형 모으기

벡터 레이어를 받아 그 도형들을 새 다중 부분 도형으로 모읍니다.

하나 이상의 속성을 지정해서 동일한 (지정한 속성에 대해 동일한 값을 가지는) 범주에 속하는 도형만 모을 수도 있고, 아니면 모든 도형을 모을 수도 있습니다.

산출 도형은 — 단일 부분 도형이라도 — 모두 다중 도형으로 변환될 것입니다. 이 알고리즘은 중첩하는 도형들을 융해하지 않습니다. 각 도형 부분의 형상을 수정하지 않은 채 함께 모을 것입니다.

대체 옵션을 알고 싶다면다중 부분으로 조성하기 또는집계 알고리즘을 참조하세요.

**기본 메뉴**: *Vector ▸ Geometry Tools*

⚠ 경고

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

**더 보기**

집계, 다중 부분으로 조성하기, 융해

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | 입력 벡터 레이어 |
| **Unique ID fields** | `FIELD` | [tablefield: any] [list] | 도형을 모으기 위한 하나 이상의 속성을 선택합니다. |
| **Collected** | `OUTPUT` | [same as input] | 모아진 도형을 담은 벡터 레이어 |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Collected** | `OUTPUT` | [same as input]<br>기본값: `[Create temporary layer]` | Specify the output vector layer for the collected geometries. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 파이썬 코드

**알고리즘 ID**: `native:collect`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 오목 껍질

입력 포인트 레이어에 있는 피처의 오목 껍질 (concave hull) 을 계산합니다.

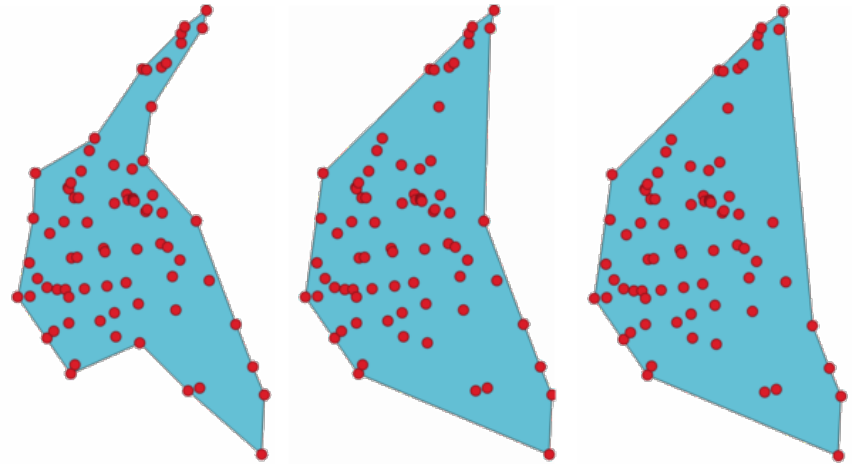


그림 24.101: 서로 다른 한계값 (0.3, 0.6, 0.9) 을 가진 오목 껍질들

**더 보기**

볼록 껍질

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input point layer** | `INPUT` | [vector: point] | 입력 포인트 벡터 레이어 |
| **Threshold** | `ALPHA` | [numeric: double]<br>기본값: 0.3 | 0(최대 오목 껍질) 에서 1(볼록 껍질) 까지의 숫자 |
| **Allow holes** | `HOLES` | [boolean]<br>기본값: True | 최종 오목 껍질에 구멍을 허용할지 여부를 선택합니다. |
| **Split multipart geometry into singlepart geometries** | `NO_MULTIGEOMETF` | [boolean]<br>기본값: False | 다중 부분 도형 대신 단일 부분 도형을 원하는 경우 활성화하십시오. |
| **Concave hull** | `OUTPUT` | [vector: polygon]<br>기본값: `[Create temporary layer]` | Specify the output vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Concave hull** | `OUTPUT` | [vector: polygon] | 산출 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:concavehull`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 도형 유형 변환하기

기존 도형을 바탕으로 다른 도형 유형의 새 레이어를 생성합니다.

산출 레이어의 속성 테이블은 입력 레이어의 속성 테이블과 동일합니다.

모든 변환이 가능하지는 않습니다. 예를 들어 라인 레이어를 포인트 레이어로 변환할 수는 있지만, 포인트 레이어를 라인 레이어로 변환할 수는 없습니다.

**더 보기**

폴리곤화, 라인을 폴리곤으로, 폴리곤을 라인으로, 포인트를 경로로

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | 입력 벡터 레이어 |
| **New geometry type** | `TYPE` | [enumeration]<br>기본값: 0 | 산출 피처에 적용할 도형 유형입니다. 다음 가운데 하나로 지정할 수 있습니다:<br>• 0 —중심 (centroid)<br>• 1 —노드<br>• 2 —라인스트링<br>• 3 —멀티라인스트링<br>• 4 —폴리곤 |

다음 페이지에 계속

표 24.621 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Converted** | `OUTPUT` | [vector: geometry]<br>기본값: `[Create temporary layer]` | Specify the output vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Converted** | `OUTPUT` | [vector: geometry] | 산출 벡터 레이어입니다. 파라미터에 따라 유형이 달라집니다. |

### 파이썬 코드

**Algorithm ID**: `native:convertgeometrytype`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 만곡 도형으로 변환

도형을 그에 대응하는 만곡 도형으로 변환합니다.

기존 만곡 도형은 그대로 유지될 것입니다.

☑ 라인 및 폴리곤 피처를제자리 피처 수정 할 수 있습니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: line or polygon] | 입력 벡터 레이어 |
| **Maximum distance tolerance** | `DISTANCE` | [numeric: double]<br>기본값: 0.000001 | 꼭짓점의 원 위치와 변환된 만곡 도형에서의 위치 간에 허용되는 최장 거리 |

다음 페이지에 계속

표 24.623 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Maximum angle tolerance** | ANGLE | [numeric: double]<br>기본값: 0.000001 | 포인트들이 후보 원호 위에 이미 균등하게 퍼져 있다면 원호로 대체해도 되는 선분이라고 간주합니다. 이 파라미터는 포인트 균등 간격 작업을 위한 테스트 시 허용되는 (도 단위) 최대 각도 편차를 지정합니다. |
| **Curves** | OUTPUT | [vector: compoundcurve or curvepolygon]<br>기본값: [Create temporary layer] | Specify the output vector layer. *One of*:<br>• 임시 레이어 생성 (TEMPORARY_OUTPUT)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Curves** | OUTPUT | [vector: compoundcurve or curvepolygon] | 만곡 도형을 가진 산출 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:converttocurves`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 볼록 껍질

입력 포인트 레이어에 있는 각 피처의 볼록 껍질 (convex hull) 을 계산합니다.

레이어 전체 또는 피처의 하위 집합 그룹을 커버하는 볼록 껍질 계산에 대해서는'minimum-bounding-geometry' 알고리즘을 참조하세요.

☑ 폴리곤 피처를제자리 피처 수정 할 수 있습니다.

**기본 메뉴**: *Vector → Geoprocessing Tools*

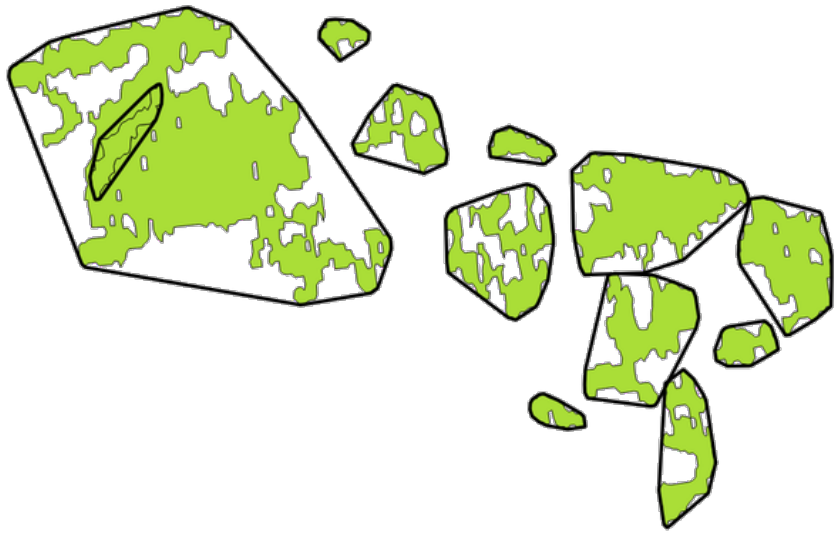


그림 24.102: 각 피처의 볼록 껍질을 표현하는 검은 선

**더 보기**

최소 경계 도형, 오목 껍질

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | 입력 벡터 레이어 |
| **Convex hull** | `OUTPUT` | [vector: polygon]<br>기본값: `[Create temporary layer]` | Specify the output vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Convex hull** | `OUTPUT` | [vector: polygon] | The output (convex hull) vector layer. Other than the input attributes, the output layer also contains the following fields: `area` and `perimeter` of the generated polygon. |

### 파이썬 코드



**알고리즘 ID**: `native:convexhull`

```
import processing
```

### 쐐기 버퍼 생성하기

입력 포인트로부터 쐐기 (wedge) 모양 버퍼를 생성합니다.

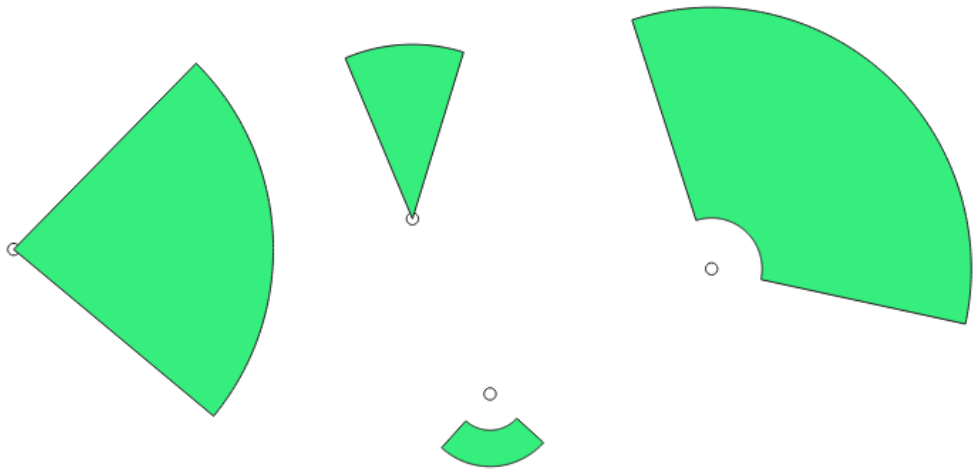


그림 24.103: 쐐기 버퍼

이 알고리즘의 본래 산출물은 만곡폴리곤 (CurvePolygon) 도형이지만, 산출물 포맷에 따라 자동으로 폴리곤으로 선분화될 수도 있습니다.

**더 보기**

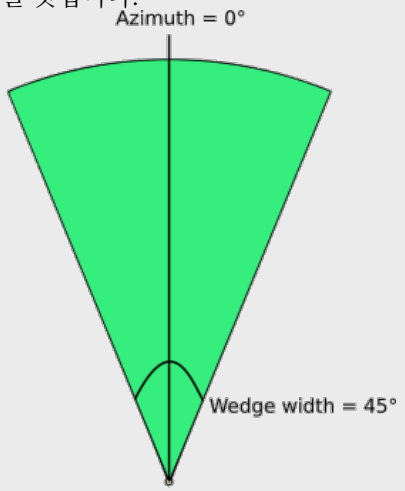
버퍼, 변동 너비 버퍼 *(M* 값으로*)*, 줄어드는 버퍼 생성하기

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: point] | 입력 포인트 벡터 레이어 |
| **Azimuth (degrees from North)** | `AZIMUTH` | [numeric: double]<br>기본값: 0.0 | 쐐기의 중간값 (도 단위) 각도 |

다음 페이지에 계속

표 24.627 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Wedge width (in degrees)** | `WIDTH` | [numeric: double]<br>기본값: 45.0 | 버퍼의 (도 단위) 너비입니다. 쐐기는 방위각 방향의 양쪽으로 각도 너비의 반씩 확장될 것입니다.<br>  <br><br>그림 24.104: 쐐기 버퍼의 방위각 및 너비 값 |
| **Outer radius** | `OUTER_RADIUS` | [numeric: double]<br>기본값: 1.0 | 쐐기의 외곽 크기 (길이) 입니다. 이 크기는 소스 포인트에서 쐐기 모양의 경계까지의 거리를 의미합니다. |
| **Inner radius**<br>선택적 | `INNER_RADIUS` | [numeric: double]<br>기본값: 0.0 | 내곽 반경 값입니다. 0 으로 설정하면 쐐기가 소스 포인트에서 시작할 것입니다. |
| **Buffers** | `OUTPUT` | [vector: polygon]<br>기본값: `[Create temporary layer]` | Specify the output vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Buffers** | `OUTPUT` | [vector: polygon] | 산출 (쐐기 버퍼) 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:wedgebuffers`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 델로네 삼각분할

입력 포인트 레이어에 대응하는 델로네 삼각분할 (Delaunay triangulation) 을 통해 폴리곤 레이어를 생성합니다.

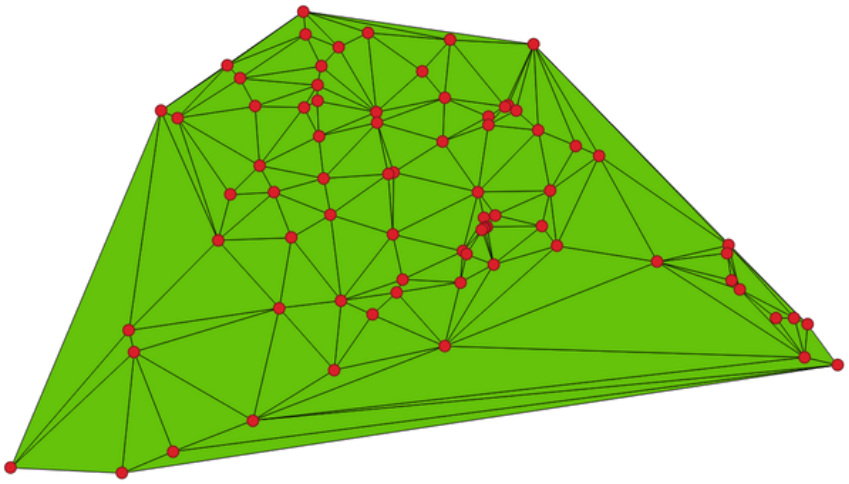


그림 24.105: 포인트에 대한 델로네 삼각분할

**기본 메뉴**: *Vector → Geometry Tools*

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: point] | 입력 포인트 벡터 레이어 |
| **Tolerance**<br>선택적 | `TOLERANCE` | [numeric: double]<br>기본값: 0.0 | 삼각분할의 견고성을 향상시키는 데 사용할 수 있는 선택적인 스냅 허용 오차를 지정합니다. |
| **Add point IDs to output** | `ADD_ATTRIBUTES` | [boolean]<br>기본값: True | 산출물에 관련 포인트 피처 ID 를 저장하고 있는 필드들을 추가해야 할지 여부를 지정합니다. 거짓으로 설정하는 경우 `id` 필드를 사용해서 폴리곤을 식별합니다. |

다음 페이지에 계속

표 24.629 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Delaunay triangulation** | `OUTPUT` | [vector: polygon] 기본값: `[Create temporary layer]` | Specify the output vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Delaunay triangulation** | `OUTPUT` | [vector: polygon] | 산출 (델로네 삼각분할) 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:delaunaytriangulation`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 구멍 삭제하기

폴리곤 레이어를 받아 폴리곤에 있는 구멍을 제거합니다. 구멍을 가진 폴리곤을 외곽 고리만 가진 폴리곤으로 대체한 새 벡터 레이어를 생성합니다. 속성은 그대로 유지됩니다.

최적 최소 면적 파라미터를 사용하면 지정한 면적 한계값보다 좁은 구멍만 제거할 수 있습니다. 이 파라미터를 기본값 `0.0` 으로 내버려두면 모든 구멍을 제거한 산출물을 생성합니다.

☑ 폴리곤 피처를제자리 피처 수정 할 수 있습니다.

**더 보기**

qgisfixgeometryhole

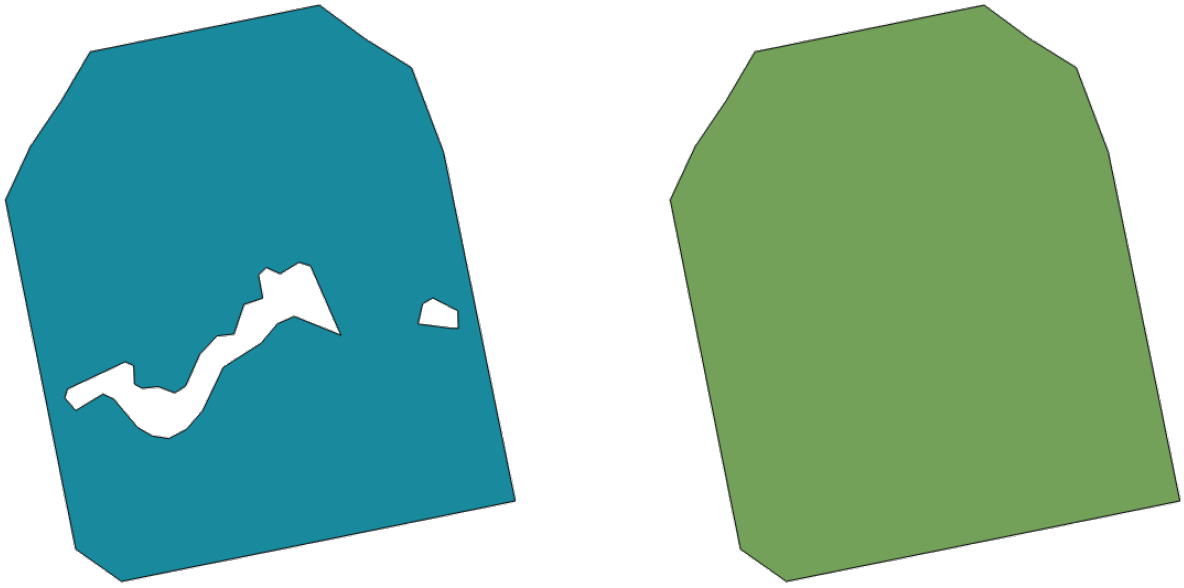


그림 24.106: 청소 이전과 이후

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: polygon] | 입력 폴리곤 벡터 레이어 |
| **Remove holes with area less than**<br>선택적 | `MIN_AREA` | [numeric: double]<br>기본값: 0.0 | 이 한계값보다 좁은 면적을 가진 구멍만 제거할 것입니다. 값이 `0.0` 인 경우, 모든 구멍을 제거한 산출물을 생성할 것입니다. |
| **Cleaned** | `OUTPUT` | [same as input]<br>기본값: `[Create temporary layer]` | Specify the output vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Cleaned** | `OUTPUT` | [same as input] | 산출 (청소된) 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:deleteholes`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

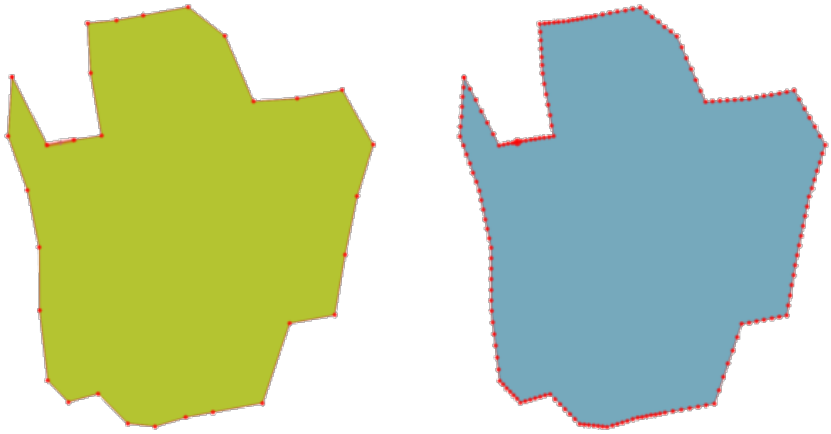


그림 24.107: 치밀화 이전과 이후의 꼭짓점을 표시하는 빨간 포인트

간격으로 치밀화하기

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: line, polygon] | 입력 라인 또는 폴리곤 벡터 레이어 |
| **Vertices to add** | `VERTICES` | [numeric: integer] 기본값: 1 | 각 선분에 추가할 새 꼭짓점의 개수 |
| **Densified** | `OUTPUT` | [same as input] 기본값: `[Create temporary layer]` | Specify the output vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Densified** | `OUTPUT` | [same as input] | 산출 (치밀화된) 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:densifygeometries`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

거리를 3 으로 지정하면 선분 `[0 0] -> [10 0]` 을 `[0 0] -> [2.5 0] -> [5 0] -> [7.5 0] -> [10 0]` 으로 변환할 것입니다. 이 선분에 추가 꼭짓점 3 개가 필요하고, 간격을 2.5 씩 증가시켜야 선분 전체에 추가 꼭짓점을 고른 간격으로 배치할 수 있기 때문입니다.

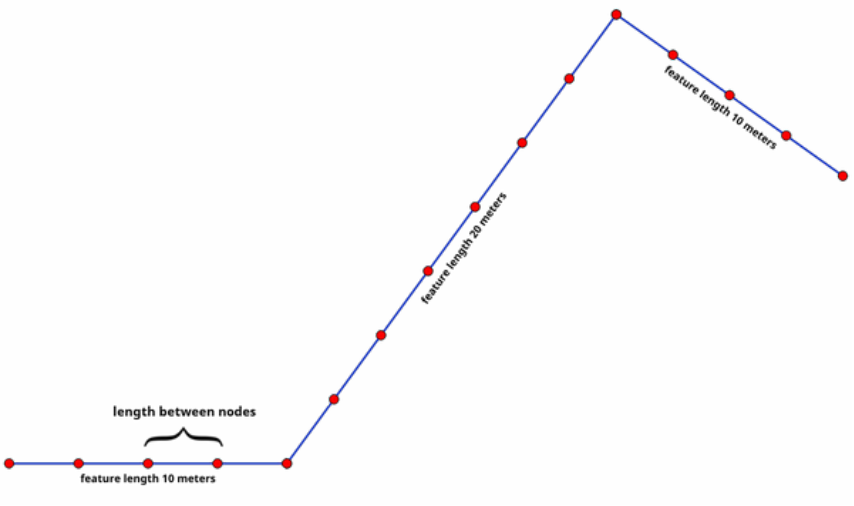



그림 24.108: 지정한 간격으로 도형 치밀화

라인 및 폴리곤 피처를제자리 피처 수정 할 수 있습니다.

**더 보기**

개수로 치밀화하기

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: line, polygon] | 입력 라인 또는 폴리곤 벡터 레이어 |
| **Interval between vertices to add** | `INTERVAL` | [numeric: double]<br>기본값: 1.0 | 연속된 두 꼭짓점 사이의 최대 거리 |
| **Densified** | `OUTPUT` | [same as input]<br>기본값: `[Create temporary layer]` | Specify the output vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Densified** | `OUTPUT` | [same as input] | 산출 (치밀화된) 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:densifygeometriesgivenaninterval`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 융해

벡터 레이어를 받아 그 피처들을 새 피처로 결합합니다. 하나 이상의 속성을 지정해서 동일한 (지정한 속성에 대해 동일한 값을 가지는) 범주에 속하는 피처들만 융해 (dissolve) 시킬 수도 있고, 아니면 모든 피처를 단일 피처로 융해시킬 수도 있습니다.

모든 산출 도형을 다중 도형으로 변환시킬 것입니다. 입력물이 폴리곤 레이어일 경우, 인접 폴리곤들의 융해된 공통 경계선을 지울 것입니다. "해체된 피처를 개별 피처로 유지"설정을 활성화하면 중첩하거나 접하지 않는 피처 및 부분을 (단일 다중부분 피처의 부분이 아니라) 개별 피처로 내보낼 것입니다.

산출 속성 테이블은 입력 레이어와 동일한 필드를 가질 것입니다. 산출 레이어의 필드에 있는 값들은 제일 먼저 처리될 입력 피처의 값들입니다.

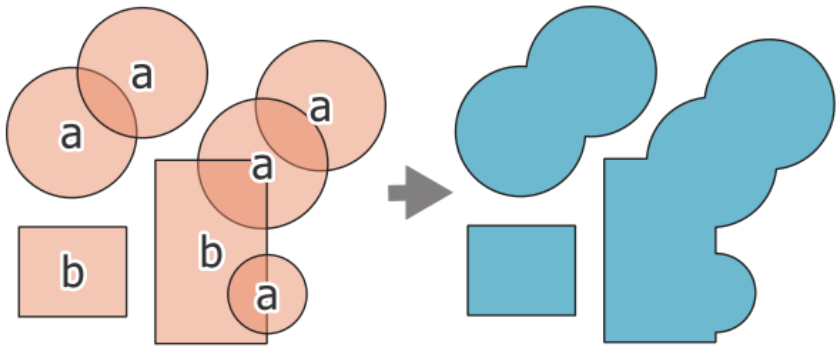



그림 24.109: 레이어를 단일 다중부분 피처로 융해시키기

**기본 메뉴**: *Vector → Geoprocessing Tools*

**경고**

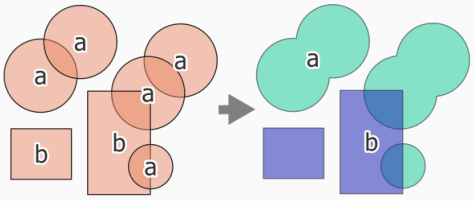
This algorithm drops existing primary keys or FID values and regenerates them in output layers.

더 보기

*Dissolve coverage*, 집계, 도형 모으기

### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | 입력 벡터 레이어 |
| **Dissolve field(s)**<br>선택적 | `FIELD` | [tablefield: any] [list]<br>기본값: [] | 선택한 필드 (들) 에서 동일한 값을 가지고 있는 피처들을 단일 피처로 대체하고 도형들을 병합할 것입니다.<br>아무 필드도 선택하지 않은 경우 모든 피처를 융해시켜 단일 (다중 부분) 피처를 산출합니다.<br>  <br><br>그림 24.110: 공통 속성을 바탕으로 폴리곤 레이어 융해 (다중부분 피처 2 개) |
| **Dissolved** | `OUTPUT` | [same as input]<br>기본값: `[Create temporary layer]` | Specify the output vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

#### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Keep disjoint features separate** | SEPARATE_DISJOINT | [boolean]<br>기본값: False | 융해된 피처의 부분들을 (다중부분 피처의 부분들로가 아닌) 개별 피처로 내보내기 |

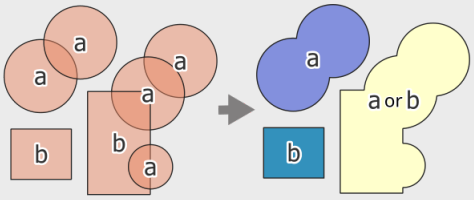



그림 24.111: 소스 (왼쪽) | 모두 융해시키기 (개별 피처 3 개 - 오른쪽)

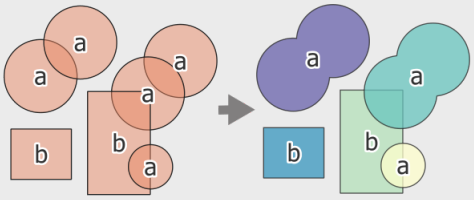



그림 24.112: 소스 (왼쪽) | 필드를 바탕으로 융해시키기 (개별 피처 5 개 - 오른쪽)

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Dissolved** | OUTPUT | [same as input] | 융해된 도형들을 가진 산출 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:dissolve`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 늘어뜨리기 (래스터로부터 Z 값 설정하기)

래스터 레이어의 밴드에서 샘플링한 값을 사용해서 피처 도형의 중첩하는 모든 꼭짓점에 Z 값을 설정합니다. 래스터 값을 사전 설정한 척도로 조정할 수 있는 옵션도 있습니다.

Z 값이 레이어에 이미 존재하는 경우, 새 값으로 덮어쓸 것입니다. Z 값이 없는 경우, Z 값을 가지도록 도형을 업그레이드할 것입니다.

☑ Z 좌표 활성화된 포인트, 라인 및 폴리곤 피처를제자리 피처 수정 할 수 있습니다.

**더 보기**

래스터로부터 *M* 값 설정하기, *Z* 값 설정하기

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | 입력 벡터 레이어 |
| **Raster layer** | `RASTER` | [raster] | Z 값을 가진 래스터 레이어 |
| **Band number** | `BAND` | [raster band]<br>기본값: 1 | Z 값을 가져올 래스터 밴드 |
| **Value for NoData or non-intersecting vertices** | `NODATA` | [numeric: double]<br>기본값: 0.0 | 꼭짓점이 래스터 (의 무결한 픽셀) 와 교차하지 않는 경우 사용할 값 |
| **Scale factor** | `SCALE` | [numeric: double]<br>기본값: 1.0 | 척도 값: 밴드 값을 이 값으로 곱합니다. |
| **Offset** | `OFFSET` | [numeric: double]<br>기본값: 0.0 | 오프셋 값: “척도 (scale factor)”를 적용한 다음 밴드 값에 대수 (□□) 적으로 추가됩니다. |
| **Updated** | `OUTPUT` | [same as input]<br>기본값: `[Create temporary layer]` | Specify the output vector layer (with Z values from the raster layer). *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Updated** | `OUTPUT` | [same as input] | 래스터 레이어에서 나온 Z 값을 가진 산출 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:setzfromraster`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## M/Z 값 제거

입력 도형에서 M(측정) 또는 Z(고도) 값을 제거합니다.

**더 보기**

*M* 값 설정하기, *Z* 값 설정하기

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | Z 또는 M 값을 가진 입력 벡터 레이어 |
| **Drop M Values** | `DROP_M_VALUES` | [boolean]<br>기본값: False | 도형에서 M 값을 제거합니다. |
| **Drop Z Values** | `DROP_Z_VALUES` | [boolean]<br>기본값: False | 도형에서 Z 값을 제거합니다. |
| **Z/M Dropped** | `OUTPUT` | [same as input]<br>기본값: `[Create temporary layer]` | Specify the output vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Z/M Dropped** | `OUTPUT` | [same as input] | (입력 레이어와 동일하지만, 도형에서 M 그리고/또는 Z 차원을 제거한) 산출 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:dropmzvalues`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 선택한 폴리곤 소거하기

입력 레이어에서 선택한 폴리곤을 공통 경계를 지워서 특정 인접 폴리곤과 결합합니다. 인접 폴리곤은 최대 또는 최소 면적을 가진 폴리곤일 수도 있고, 또는 소거 (eliminate) 될 폴리곤과 공통 경계를 가장 많이 공유하는 폴리곤일 수도 있습니다.

소거 알고리즘은 보통 조각 (sliver) 폴리곤, 예를 들어 폴리곤 교차 공간 처리 작업 도중 입력물의 경계와 유사하지만 동일하지는 않은 경우 산출되는 아주 작은 폴리곤을 제거하는 데 사용됩니다.

**기본 메뉴**: *Vector → Geoprocessing Tools*

**더 보기**

도형 고치기

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: polygon] | 입력 폴리곤 벡터 레이어 |
| **Merge selection with the neighboring polygon with the** | `MODE` | [enumeration] 기본값: 설정하지 않음 | 선택한 폴리곤을 제거하기 위해 사용할 파라미터를 선택합니다:<br>• 0 —최대 면적 (Largest Area)<br>• 1 —최소 면적 (Smallest Area)<br>• 2 —최대 공유 경계 (Largest Common Boundary) |

다음 페이지에 계속

표 24.644 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Eliminated** | `OUTPUT` | [vector: polygon]<br>기본값: `[Create temporary layer]` | Specify the output vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Eliminated** | `OUTPUT` | [vector: polygon] | 산출 폴리곤 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `qgis:eliminateselectedpolygons`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 라인 조각내기

라인 레이어를 받아 각 라인을 원본에 있는 선분들을 표현하는 라인들의 집합으로 대체한 새 레이어를 생성합니다.

산출되는 레이어의 각 라인은 시작점과 종단점만 담고 있습니다. 시작점과 종단점 사이에 중간 꼭짓점이 없습니다.

If the input layer consists of CircularStrings or CompoundCurves, the output layer will be of the same type and contain only single curve segments.

**참고**

- This algorithm drops existing primary keys or FID values and regenerates them in output layers.
- This algorithm does not require valid geometries as input.

라인 피처를제자리 피처 수정 할 수 있습니다.

**더 보기**

세분화, 라인 부스트링 생성하기

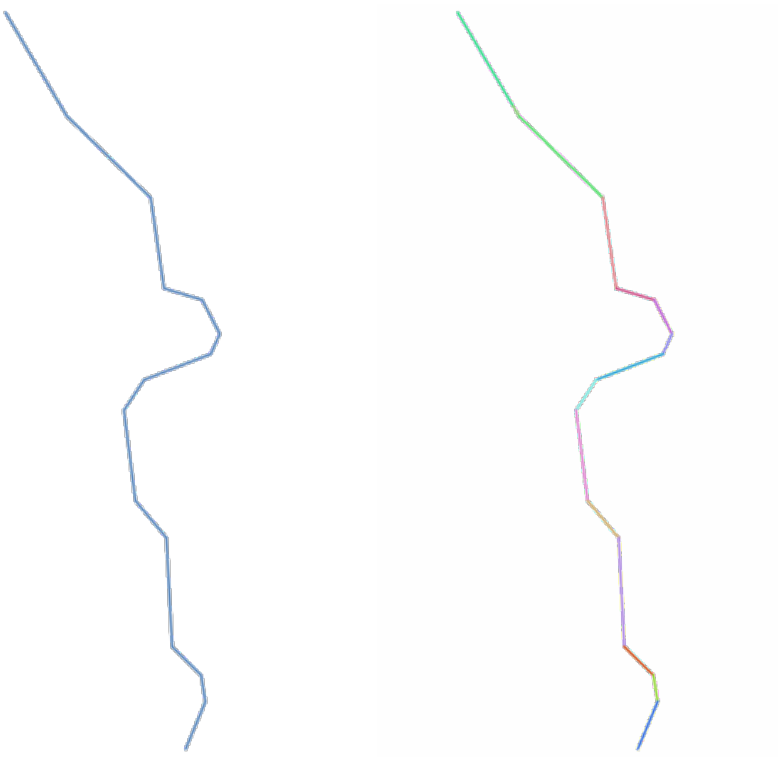


그림 24.113: 원본 라인 레이어와 조각낸 레이어

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: line] | 입력 라인 벡터 레이어 |
| **Exploded** | `OUTPUT` | [vector: line]<br>기본값: `[Create temporary layer]` | Specify the output vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Exploded** | `OUTPUT` | [vector: line] | 입력 레이어의 각 선분을 표현하는 피처를 가진 산출 라인 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:explodelines`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

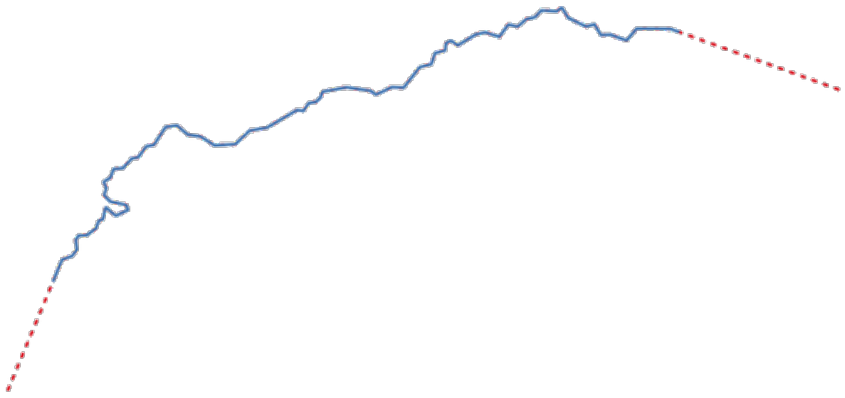


그림 24.114: 원본 레이어의 처음과 마지막 연장을 표현하는 빨간 점선

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: line] | 입력 라인 벡터 레이어 |
| **Start distance** | `START_DISTANCE` | [numeric: double]<br>기본값: 0.0 | 라인의 첫 번째 선분을 연장할 거리 (시작점) |
| **End distance** | `END_DISTANCE` | [numeric: double]<br>기본값: 0.0 | 라인의 마지막 선분을 연장할 거리 (시작점) |
| **Extended** | `OUTPUT` | [vector: line]<br>기본값: `[Create temporary layer]` | Specify the output vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Extended** | `OUTPUT` | [vector: line] | 산출 (연장된) 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:extendlines`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 는 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | 입력 벡터 레이어 |
| **Summaries to calculate** | `SUMMARIES` | [enumeration]<br>기본값: [0] | 도형의 M 값에 대한 통계를 계산합니다. 다음 가운데 하나 이상을 지정할 수 있습니다:<br>• 0 —첫 번째<br>• 1 —마지막<br>• 2 —개수<br>• 3 —합계<br>• 4 —평균<br>• 5 —중앙값<br>• 6 —표준 편차 (population)<br>• 7 —최소값<br>• 8 —최대값<br>• 9 —범위 (range)<br>• 10 —소수 기준 (Minority)<br>• 11 —다수 기준 (Majority)<br>• 12 —다양도 (Variety)<br>• 13 —제 1 사분위수<br>• 14 —제 3 사분위수<br>• 15 —사분위수의 범위 |
| **Output column prefix** | `COLUMN_PREFIX` | [string]<br>기본값: 'm_' | 산출 (M) 열의 이름 앞에 붙일 접두어 |
| **Extracted** | `OUTPUT` | [same as input]<br>기본값: `[Create temporary layer]` | Specify the output layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Extracted** | `OUTPUT` | [same as input] | M 값을 가진 산출 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:extractmvalues`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 특정 꼭짓점 추출하기

벡터 레이어를 받아 입력 도형에 있는 특정 꼭짓점을 표현하는 포인트를 담은 새 포인트 레이어를 생성합니다.

예를 들어, 이 알고리즘을 도형의 첫 번째 또는 마지막 꼭짓점을 추출하기 위해 사용할 수 있습니다. 각 포인트와 관련된 속성은 꼭짓점이 속한 피처의 속성과 동일합니다.

꼭짓점 인덱스 파라미터는 추출할 꼭짓점의 인덱스를 지정하는 쉼표로 구분된 문자열을 입력받습니다. 첫 번째 꼭짓점은 0 번 인덱스에 대응하고, 두 번째 꼭짓점은 1 번 인덱스에, 세 번째 꼭짓점은 등등. 음의 인덱스를 사용해서 도형의 마지막 꼭짓점을 찾을 수 있습니다. 예를 들어 -1 번 인덱스는 마지막 꼭짓점에 대응하고, -2 번 인덱스는 마지막에서 두 번째 꼭짓점을 등등.

특정 꼭짓점의 위치 (예: 0, -1 등등), 원본 꼭짓점 인덱스, 꼭짓점이 속한 부분 및 그 부분 (은 물론 폴리곤의 경우 꼭짓점이 속한 고리) 안에서의 꼭짓점 인덱스, 원본 도형 상에서 첫 번째 꼭짓점부터의 거리, 꼭짓점의 원본 도형에 대한 이등분선 (bisector) 각도를 포함하는 추가 필드들을 꼭짓점에 추가합니다.

포인트 피처를제자리 피처 수정 할 수 있습니다.

**경고**

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

**더 보기**

꼭짓점 추출하기, *M* 값으로 꼭짓점 필터링하기, *Z* 값으로 꼭짓점 필터링하기

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | 입력 벡터 레이어 |
| **Vertex indices** | `VERTICES` | [string]<br>기본값: '0' | 추출할 꼭짓점의 인덱스를 가리키는 쉼표로 구분된 문자열 |

다음 페이지에 계속

표 24.652 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Vertices** | `OUTPUT` | [vector: point]<br>기본값: `[Create temporary layer]` | Specify the output vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Vertices** | `OUTPUT` | [vector: point] | The output (point) vector layer containing the specified vertices from the input layer geometries. Other than the input attributes, the output layer also contains the following fields:<br>• `vertex_pos`: the index of the vertex in the geometry, as requested by the user.<br>• `vertex_index`: the index of the vertex within the geometry. The first vertex is indexed as 0.<br>• `vertex_part`: the part number of the geometry that the vertex belongs to. For single-part geometries, this is always 0. For multi-part geometries (e.g. MultiPolygon or MultiLine), this indicates which part the vertex comes from.<br>• `vertex_part_ring`: for polygon geometries, this indicates whether the vertex belongs to the outer ring (0) or one of the inner rings (1, 2, ⋯).<br>• `vertex_part_index`: the index of the vertex in its specific ring (outer or inner).<br>• `distance`: the cumulative length of line segments from the first vertex of the geometry.<br>• `angle`: the angle (in degrees) between the previous and the current vertex. |

### 파이썬 코드

**알고리즘 ID**: `native:extractspecificvertices`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 꼭짓점 추출하기

벡터 레이어를 받아 입력 도형에 있는 꼭짓점들을 표현하는 포인트들을 담은 새 포인트 레이어를 생성합니다.

각 포인트와 관련된 속성은 꼭짓점이 속한 피처의 속성과 동일합니다.

꼭짓점의 인덱스 (0 부터 시작), 꼭짓점이 속한 피처의 부분 및 그 부분 (은 물론 폴리곤의 경우 꼭짓점이 속한 고리) 안에서의 꼭짓점 인덱스, 원본 도형 상에서 첫 번째 꼭짓점부터의 거리, 꼭짓점의 원본 도형에 대한 이등분선 (bisector) 각도를 포함하는 추가 필드들을 꼭짓점에 추가합니다.

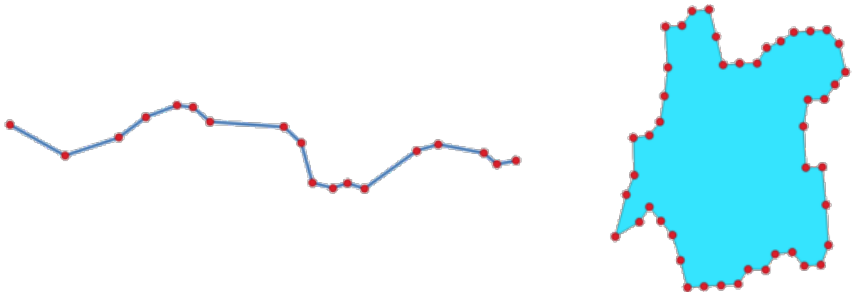


그림 24.115: 라인 및 폴리곤 레이어에서 추출한 꼭짓점들

포인트 피처를제자리 피처 수정 할 수 있습니다.

**기본 메뉴**: *Vector → Geometry Tools*

**경고**

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

**더 보기**

특정 꼭짓점 추출하기, *M* 값으로 꼭짓점 필터링하기, *Z* 값으로 꼭짓점 필터링하기

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | 입력 벡터 레이어 |
| **Vertices** | `OUTPUT` | [vector: point]<br>기본값: `[Create temporary layer]` | Specify the output vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Vertices** | `OUTPUT` | [vector: point] | The output (point) vector layer containing the vertices from the input layer geometries. Other than the input attributes, the output layer also contains the following fields:<br>• `vertex_index`: the index of the vertex within the geometry. The first vertex is indexed as 0.<br>• `vertex_part`: the part number of the geometry that the vertex belongs to. For single-part geometries, this is always 0. For multi-part geometries (e.g. MultiPolygon or MultiLine), this indicates which part the vertex comes from.<br>• `vertex_part_ring`: for polygon geometries, this indicates whether the vertex belongs to the outer ring (0) or one of the inner rings (1, 2, ⋯).<br>• `vertex_part_index`: the index of the vertex in its specific ring (outer or inner).<br>• `distance`: the distance from the previous vertex.<br>• `angle`: returns approximate angle at a vertex. Usually the average angle between adjacent segments, and can be pictured as the orientation of a line following the curvature of the geometry at the specified vertex. |

### 파이썬 코드

**알고리즘 ID**: `native:extractvertices`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## Z 값 추출하기

도형에서 피처 속성으로 Z 값을 추출합니다.

기본적으로 각 피처의 첫 꼭짓점의 Z 값만 추출하지만, 도형의 모든 Z 값에 대해 합계, 평균, 최소값 및 최대값을 포함하는 통계를 계산할 수 있는 옵션이 있습니다.

**더 보기**

*M* 값 추출하기, *Z* 값 설정하기, *M/Z* 값 제거

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | 입력 벡터 레이어 |
| **Summaries to calculate** | `SUMMARIES` | [enumeration]<br>기본값: [0] | 도형의 Z 값에 대한 통계를 계산합니다. 다음 가운데 하나 이상을 지정할 수 있습니다:<br>• 0 —첫 번째<br>• 1 —마지막<br>• 2 —개수<br>• 3 —합계<br>• 4 —평균<br>• 5 —중앙값<br>• 6 —표준 편차 (population)<br>• 7 —최소값<br>• 8 —최대값<br>• 9 —범위 (range)<br>• 10 —소수 기준 (Minority)<br>• 11 —다수 기준 (Majority)<br>• 12 —다양도 (Variety)<br>• 13 —제 1 사분위수<br>• 14 —제 3 사분위수<br>• 15 —사분위수의 범위 |
| **Output column prefix** | `COLUMN_PREFIX` | [string]<br>기본값: 'z_' | 산출 (Z) 열의 이름 앞에 붙일 접두어 |

다음 페이지에 계속

표 24.656 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Extracted** | OUTPUT | [same as input]<br>기본값: [Create temporary layer] | Specify the output layer. *One of*:<br>• 임시 레이어 생성 (TEMPORARY_OUTPUT)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Extracted** | OUTPUT | [same as input] | Z 값을 가진 산출 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:extractzvalues`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## M 값으로 꼭짓점 필터링하기

지정한 최소값 이상 그리고/또는 최대값 이하인 M 값을 가지는 꼭짓점 포인트를 가진 도형만 반환해서, 꼭짓점의 M 값으로 꼭짓점을 필터링합니다.

최소값을 지정하지 않은 경우 최대값에 대해서만 점검하고, 마찬가지로 최대값을 지정하지 않은 경우 최소값에 대해서만 점검합니다.

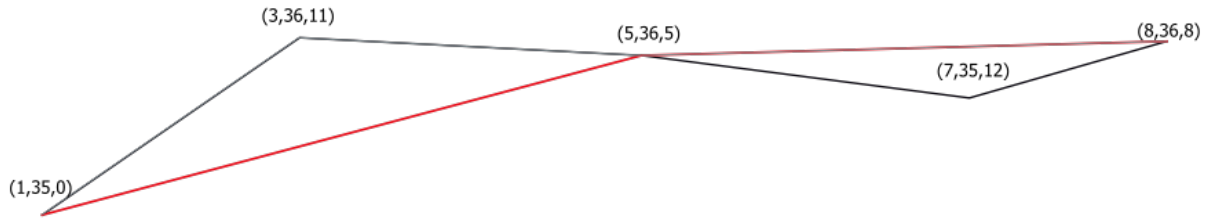



그림 24.116: 검은 라인에서 M 값이 10 이하인 꼭짓점만 잇는 빨간 라인

Allows *features in-place modification* of point, line and polygon features with M enabled

**경고**

Depending on the input geometry attributes and the filters used, the geometries created by this algorithm may no longer be valid.

**더 보기**

Z 값으로 꼭짓점 필터링하기, 꼭짓점 추출하기, 특정 꼭짓점 추출하기

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | Input vector layer to remove vertices from |
| **Minimum**<br>선택적 | `MIN` | [numeric: double]<br>기본값: 설정하지 않음 | 허용되는 최소 M 값 |
| **Maximum**<br>선택적 | `MAX` | [numeric: double]<br>기본값: 설정하지 않음 | 허용되는 최대 M 값 |
| **Filtered** | `OUTPUT` | [same as input]<br>기본값: `[Create temporary layer]` | Specify the output vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Filtered** | `OUTPUT` | [same as input] | 필터링된 피처만 가진 산출 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:filterverticesbym`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## Z 값으로 꼭짓점 필터링하기

지정한 최소값 이상 그리고/또는 최대값 이하인 Z 값을 가지는 꼭짓점 포인트를 가진 도형만 반환해서, 꼭짓점의 Z 값으로 꼭짓점을 필터링합니다.

최소값을 지정하지 않은 경우 최대값에 대해서만 점검하고, 마찬가지로 최대값을 지정하지 않은 경우 최소값에 대해서만 점검합니다.

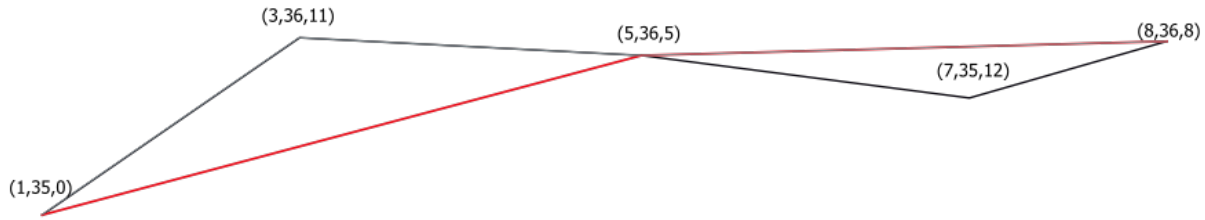



그림 24.117: 검은 라인에서 Z 값이 10 이하인 꼭짓점만 잇는 빨간 라인

Allows *features in-place modification* of point, line and polygon features with Z enabled

**경고**

Depending on the input geometry attributes and the filters used, the geometries created by this algorithm may no longer be valid. You may need to run the 도형 고치기 algorithm to ensure their validity.

**더 보기**

*M* 값으로 꼭짓점 필터링하기, 꼭짓점 추출하기, 특정 꼭짓점 추출하기

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | Input vector layer to remove vertices from |
| **Minimum**<br>선택적 | `MIN` | [numeric: double]<br>  <br>기본값: 설정하지 않음 | 허용되는 최소 Z 값 |
| **Maximum**<br>선택적 | `MAX` | [numeric: double]<br>  <br>기본값: 설정하지 않음 | 허용되는 최대 Z 값 |
| **Filtered** | `OUTPUT` | [same as input]<br>기본값: `[Create temporary layer]` | Specify the output vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Filtered** | `OUTPUT` | [same as input] | 필터링된 피처만 가진 산출 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:filterverticesbyz`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 도형 고치기

입력 꼭짓점을 하나도 잃지 않은 채 지정한 무결하지 않은 도형의 무결한 표현을 생성하려 합니다. 이미 무결한 도형은 다른 처리 없이 그대로 반환됩니다. 이 알고리즘은 항상 다중 도형 레이어를 산출합니다.

☑ M 좌표 활성화되지 않은 포인트, 라인 및 폴리곤 피처를제자리 피처 수정 할 수 있습니다.

**경고**

산출물에서 M 값은 제거될 것입니다.

**더 보기**

무결성 점검하기, *Fix Geometry*

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | 입력 벡터 레이어 |
| **Repair method** | `METHOD` | [enumeration]<br>기본값: 1 | 도형을 고치는 데 쓰이는 방법입니다. 다음 가운데 하나로 선택할 수 있습니다:<br>• 0 —`Linework`: 모든 고리를 노드화된 라인 집합으로 결합한 다음 해당 라인 집합에서 무결한 폴리곤을 추출합니다.<br>• 1 —`Structure`: first makes all rings valid and then merges shells and subtracts holes from shells to generate valid result. Assumes that holes and shells are correctly categorized.<br>**주의**<br>Using this method requires QGIS installed with GEOS >= 3.10 (see *Help* ▯ *About* menu). |
| **Fixed geometries** | `OUTPUT` | [same as input]<br>기본값: `[Create temporary layer]` | Specify the output vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Fixed geometries** | `OUTPUT` | [same as input] | 고쳐진 도형을 가진 산출 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:fixgeometries`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 오른손 법칙 강제하기

폴리곤 도형이 오른손 법칙 (right-hand rule) 을 따르도록 강제합니다. 즉 폴리곤의 경계가 닫고 있는 영역은 경계 방향의 오른쪽에 있다는 의미입니다. 정확히 말하자면, 폴리곤의 외곽 고리는 시계 방향이며 모든 내곽 고리는 시계 반대 방향입니다.

폴리곤 피처를제자리 피처 수정 할 수 있습니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: polygon] | 입력 벡터 레이어 |
| **Reoriented** | `OUTPUT` | [vector: polygon]<br>기본값: `[Create temporary layer]` | Specify the output vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Reoriented** | `OUTPUT` | [vector: polygon] | 방향을 바꾼 도형을 가진 산출 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:forcerhr`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 반대 자오선에서 분할된 측지선

라인이 반대 자오선 (±180° 경도) 을 지나갈 때마다 라인을 여러 측지 (geodesic) 선분으로 분할합니다.

반대 자오선에서 분할하면 일부 투영체에서 라인을 가시적으로 표시하는 데 도움이 될 수 있습니다. 반환되는 도형은 언제나 다중 부분 도형일 것입니다.

입력 도형의 라인 선분이 반대 자오선을 지날 때마다, 이 선분의 반대 자오선 양쪽에 있는 포인트를 연결하는 측지선을 이용해서 분할 위치의 위도를 결정해서 라인 선분을 선분 2 개로 분할할 것입니다.

입력 도형이 Z 또는 M 값을 가지고 있을 경우 반대 자오선에서 생성된 새 꼭짓점에 이 값을 선형적으로 보간할 것입니다.

> **참고**
>
> This algorithm uses ellipsoid based measurements and respects the current *ellipsoid settings*.

☑ 라인 피처를제자리 피처 수정 할 수 있습니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: line] | 입력 라인 벡터 레이어 |

다음 페이지에 계속

표 24.666 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Split** | `OUTPUT` | [vector: line]<br>기본값: `[Create temporary layer]` | Specify the output line vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Split** | `OUTPUT` | [vector: line] | 반대 자오선에서 분할된 산출 라인 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:antimeridiansplit`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 표현식으로 도형 수정/생성하기

QGIS 표현식을 사용해서 입력 피처의 기존 도형을 업데이트 (하거나 새 도형을 생성) 합니다.

산출 피처의 도형을 수정하고 생성하는 데 QGIS 표현식 엔진의 모든 유연성을 활용해서 복잡한 도형을 수정할 수 있습니다.

QGIS 표현식 함수에 대한 도움이 필요하다면, 표현식 작성기 의 내장 도움말을 참조하세요.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: any] | 입력 벡터 레이어 |

다음 페이지에 계속

표 24.668 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output geometry type** | `OUTPUT_GEOMETRY` | [enumeration] 기본값: 0 | 산출 도형은 표현식에 크게 의존합니다. 예를 들어, 사용자가 버퍼를 생성한 경우 도형 유형이 폴리곤이어야 합니다. 다음 가운데 하나로 지정할 수 있습니다:<br>• 0 — 폴리곤<br>• 1 — 라인<br>• 2 — 포인트 |
| **Output geometry has z values** | `WITH_Z` | [boolean] 기본값: False | 산출 도형에 Z 차원을 포함시켜야 하는지 여부를 선택합니다. |
| **Output geometry has m values** | `WITH_M` | [boolean] 기본값: False | 산출 도형에 M 차원을 포함시켜야 하는지 여부를 선택합니다. |
| **Geometry expression** | `EXPRESSION` | [expression] 기본값: '$geometry' | 사용자가 사용하고자 하는 도형 표현식을 추가하십시오. 표현식 대화창을 여는 버튼을 클릭해도 됩니다. 이 대화창은 관련 있는 모든 표현식 목록과 함께 도움말과 지침을 가지고 있습니다. |
| **Modified geometry** | `OUTPUT` | [vector: geometry] 기본값: `[Create temporary layer]` | Specify the output vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Modified geometry** | `OUTPUT` | [vector: geometry] | 산출 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:geometrybyexpression`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 라인에 포인트를 보간하기

라인 또는 만곡 도형을 따라 지정한 거리에 보간된 포인트 도형을 생성합니다.

Z 또는 M 값은 기존 Z 또는 M 값에서 선형적으로 보간됩니다.

도형이 다중 부분인 경우, 부스트링 (substring) 계산시 첫 번째 부분만 고려합니다.

지정한 거리가 피처의 길이를 초과하는 경우, 산출 피처가 NULL 도형을 가질 것입니다.

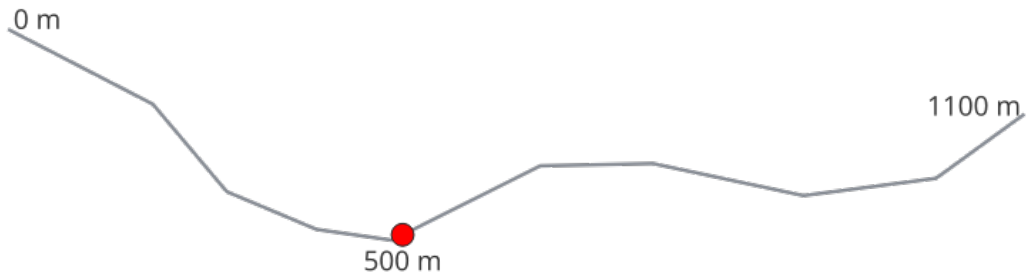



그림 24.118: 라인의 시작에서 500m 거리에 보간된 포인트

**더 보기**

도형을 따라 포인트 생성하기

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: line, polygon] | 입력 라인 또는 폴리곤 벡터 레이어 |
| **Distance** | `DISTANCE` | [numeric: double]<br>기본값: 0.0 | 라인의 시작으로부터의 거리 |
| **Interpolated points** | `OUTPUT` | [vector: point]<br>기본값: `[Create temporary layer]` | Specify the output vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Interpolated points** | `OUTPUT` | [vector: point] | 라인 또는 폴리곤 도형을 따라 지정한 거리에 있는 피처를 담은 산출 포인트 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:interpolatepoint`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 가장 큰 N 개의 부분을 남기기

폴리곤 또는 멀티폴리곤을 가진 레이어를 받아 각 멀티폴리곤 피처에서 가장 큰 *n* 개의 폴리곤만 남긴 새 레이어를 반환합니다. 해당 피처가 *n* 개 이하의 부분만 가지고 있는 경우, 그냥 피처를 복사할 것입니다.

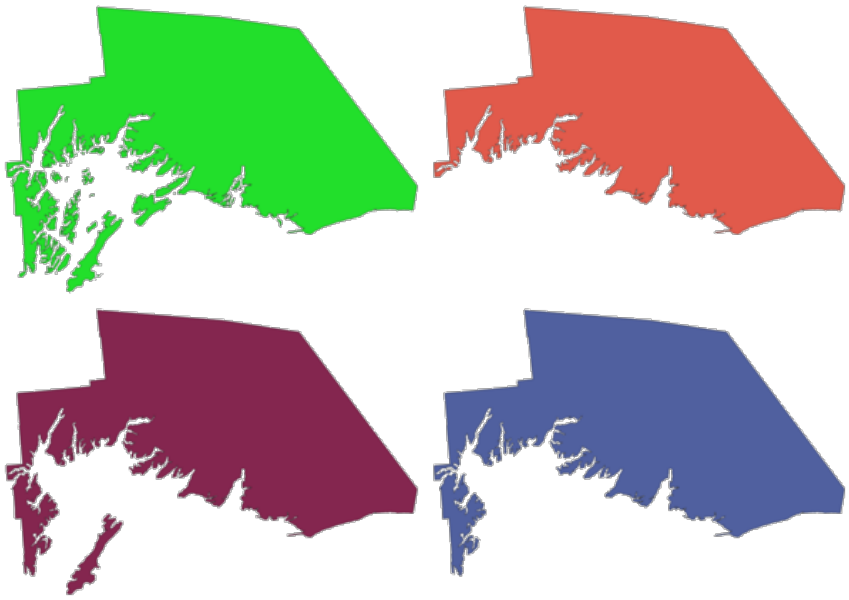


그림 24.119: 좌상단부터 시계 방향으로: 원본 다중 부분 피처, 가장 큰 1, 2, 3 개의 부분을 남긴 피처들

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Polygons** | `INPUT` | [vector: polygon] | 입력 폴리곤 벡터 레이어 |
| **Parts to keep** | `PARTS` | [numeric: integer]<br>기본값: 1 | 남길 부분의 개수. 1 로 설정하면 피처의 가장 큰 부분만 남길 것입니다. |
| **Parts** | `OUTPUT` | [vector: polygon]<br>기본값: `[Create temporary layer]` | Specify the output polygon vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Parts** | `OUTPUT` | [vector: polygon] | 각 피처의 가장 큰 N 개의 부분을 담은 산출 폴리곤 벡터 레이어 |

### 파이썬 코드

**Algorithm ID**: `native:keepnbiggestparts`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 라인 부스트링 생성하기

지정한 (라인의 시작에서 측정한) 시작 거리와 종단 거리 사이에 떨어지는 라인 (또는 곡선) 의 부분을 반환합니다.

Z 또는 M 값은 기존 Z 또는 M 값에서 선형적으로 보간됩니다.

도형이 다중 부분인 경우, 부스트링 (substring) 계산시 첫 번째 부분만 고려합니다.

라인 피처를제자리 피처 수정 할 수 있습니다.

**더 보기**

라인 연장하기




그림 24.120: 시작 거리를 0m 로, 종단 거리를 250m 로 설정한 부스트링 라인

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: line] | 입력 라인 벡터 레이어 |
| **Start distance** | `START_DISTANCE` | [numeric: double]<br>기본값: 0.0 | 입력 라인을 따라 산출 피처의 시작점까지의 거리 |
| **End distance** | `END_DISTANCE` | [numeric: double]<br>기본값: 1.0 | 입력 라인을 따라 산출 피처의 종단점까지의 거리 |
| **Substring** | `OUTPUT` | [vector: line]<br>기본값: `[Create temporary layer]` | Specify the output line vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Substring** | `OUTPUT` | [vector: line] | 산출 라인 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:linesubstring`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 라인을 폴리곤으로



입력 라인 레이어의 라인을 폴리곤 고리로 사용해서 폴리곤 레이어를 생성합니다.

산출 레이어의 속성 테이블은 입력 레이어의 속성 테이블과 동일합니다.

폴리곤을 라인으로, 폴리곤화, 도형 유형 변환하기

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | INPUT | [vector: line] | 입력 라인 벡터 레이어 |
| **Polygons** | OUTPUT | [vector: polygon]<br>기본값: [Create temporary layer] | Specify the output polygon vector layer. *One of*:<br>• 임시 레이어 생성 (TEMPORARY_OUTPUT)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Polygons** | OUTPUT | [vector: polygon] | 산출 폴리곤 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `qgis:linestopolygons`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 라인 병합하기

멀티라인스트링 도형의 모든 연결된 부분들을 단일 라인스트링 도형으로 결합합니다.

입력 멀티라인스트링 도형의 한 부분이라도 연결되지 않은 경우, 병합할 수 있는 모든 라인을 담은 멀티라인스트링과 연결되지 않은 모든 라인 부분을 가진 도형을 산출할 것입니다.

라인 피처를제자리 피처 수정 할 수 있습니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: line] | 입력 라인 벡터 레이어 |
| **Merged** | `OUTPUT` | [vector: line]<br>기본값: `[Create temporary layer]` | Specify the output line vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Merged** | `OUTPUT` | [vector: line] | 산출 (병합된) 라인 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:mergelines`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 최소 경계 도형

입력 레이어의 피처들을 외함 (▢▢) 하는 도형을 생성합니다. 피처를 필드로 그룹화할 수 있습니다. 그러면 그룹화 값별로 일치하는 값을 가진 피처의 도형을 커버하는 (최소 경계 상자) 도형을 가진 피처 1 개씩 담은 레이어를 산출할 것입니다.

다음과 같은 외함 (enclose) 하는 도형 유형을 지원합니다:

- 경계 상자 (엔벨로프)
- 기울어진 직사각형 (oriented rectangle)
- 원
- 볼록 껍질

**더 보기**

최소 외함 원

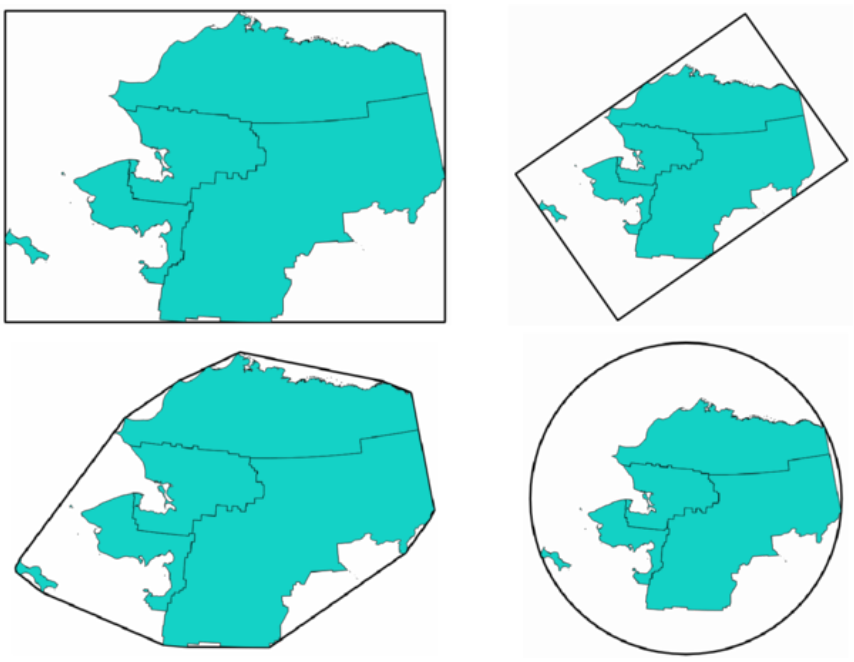


그림 24.121: 좌상단부터 시계 방향으로: 엔벨로프, 기울어진 직사각형, 원, 볼록 껍질

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | 입력 벡터 레이어 |
| **Field**<br>선택적 | `FIELD` | [tablefield: any] | 피처를 필드로 그룹화할 수 있습니다. 이 파라미터를 지정하면, 그룹화 값별로 일치하는 값을 가진 피처만 커버하는 최소 도형을 가진 피처를 1 개씩 담은 레이어를 산출할 것입니다. |
| **Geometry type** | `TYPE` | [enumeration]<br>기본값: 0 | 외함 도형 유형을 지정합니다. 다음 가운데 하나를 선택할 수 있습니다:<br>• 0 — 엔벨로프 (경계 상자)<br>• 1 — 최소 기울어진 직사각형<br>• 2 — 최소 외함 원<br>• 3 — 볼록 껍질 |
| **Bounding geometry** | `OUTPUT` | [vector: polygon]<br>기본값: `[Create temporary layer]` | Specify the output polygon vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Bounding geometry** | `OUTPUT` | [vector: polygon] | The output (bounding) polygon vector layer. It has the fields from the input layer and, depending on the selected parameter `TYPE`, the |

### 파이썬 코드

**알고리즘 ID**: `qgis:minimumboundinggeometry`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 최소 외함 원

입력 레이어에 있는 피처의 최소 외함 원을 계산합니다.

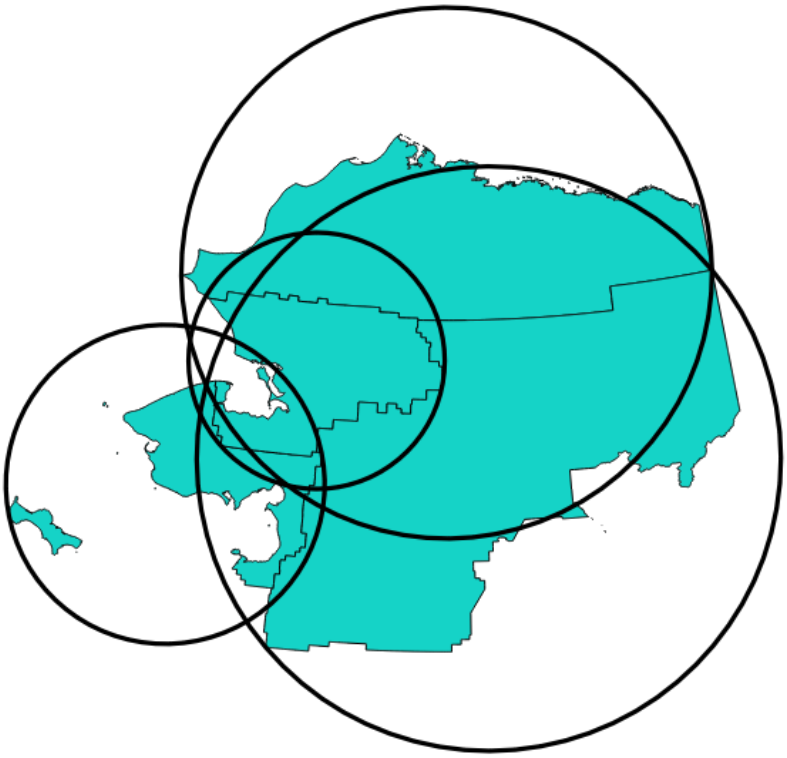


그림 24.122: 각 피처를 외함하는 원들

폴리곤 피처를제자리 피처 수정 할 수 있습니다.

**더 보기**

최소 경계 도형

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | 입력 벡터 레이어 |
| **Number of segments in circles** | `SEGMENTS` | [numeric: integer] 기본값: 72 | 원에 가깝게 만드는 데 사용할 선분의 개수입니다. 최소값은 8, 최대값은 100,000 입니다. |
| **Minimum enclosing circles** | `OUTPUT` | [vector: polygon] 기본값: `[Create temporary layer]` | Specify the output polygon vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Minimum enclosing circles** | `OUTPUT` | [vector: polygon] | The output polygon vector layer. Other than the input attributes, the output layer also contains the following fields: `area` and `radius` of the generated polygon. |

### 파이썬 코드

**알고리즘 ID**: `native:minimumenclosingcircle`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 다중 고리 버퍼 (상수 거리)

고정 또는 동적인 서리와 고리의 개수를 사용해서 입력 레이어의 피처에 대해 다중 고리 (도넛) 버퍼를 계산합니다.

☑ 폴리곤 피처를제자리 피처 수정 할 수 있습니다.

> **⚠ 경고**
>
> This algorithm drops existing primary keys or FID values and regenerates them in output layers.

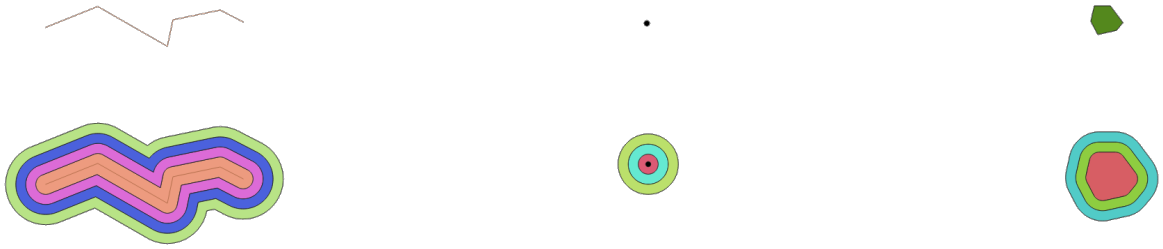


그림 24.123: 라인, 포인트 및 폴리곤 레이어의 다중 고리 버퍼

**더 보기**

버퍼, 가변 거리 버퍼, 직사각형, 타원형, 마름모, 한쪽 버퍼 생성하기

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | 입력 벡터 레이어 |
| **Number of rings** | `RINGS` | [numeric: integer]<br>기본값: 1 | 고리의 개수. 유일값일 수도 있고 (모든 피처에 대해 동일한 개수의 고리) 또는 피처 데이터에서 추출한 값일 수도 있습니다 (피처 값에 따라 고리의 개수가 달라집니다). |
| **Distance between rings** | `DISTANCE` | [numeric: double]<br>기본값: 1.0 | 고리들 사이의 개수. 유일값일 수도 있고 (모든 피처에 대해 동일한 거리) 또는 피처 데이터에서 추출한 값일 수도 있습니다 (피처 값에 따라 거리가 달라집니다). |
| **Multi-ring buffer (constant distance)** | `OUTPUT` | [vector: polygon]<br>기본값: `[Create temporary layer]` | Specify the output polygon vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Multi-ring buffer (constant distance)** | `OUTPUT` | [vector: polygon] | The output polygon vector layer. Other than the input attributes, the output layer also contains the following fields: `ringId` and `distance` of the generated buffer polygon. |

### 파이썬 코드

### 다중 부분에서 단일 부분으로

입력 레이어의 다중 부분 피처를 단일 부분 피처로 분할합니다.
산출 레이어의 속성은 입력 레이어와 동일하지만 개별 피처별로 나뉩니다.



그림 24.124: 왼쪽: 다중 부분 소스 레이어, 오른쪽: 단일 부분 산출 레이어

☑ 포인트, 라인 및 폴리곤 피처를제자리 피처 수정 할 수 있습니다.

**기본 메뉴**: *Vector ▸ Geometry Tools*

**경고**

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

**더 보기**

도형 모으기, 다중 부분으로 조성하기

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | 입력 벡터 레이어 |
| **Single parts** | `OUTPUT` | [same as input]<br>기본값: `[Create temporary layer]` | Specify the output polygon vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Single parts** | `OUTPUT` | [same as input] | 산출 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:multiparttosingleparts`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 라인 오프셋시키기

라인을 지정한 거리로 오프셋시킵니다. 양의 거리는 라인을 왼쪽으로 오프셋시키고, 음의 거리는 오른쪽으로 오프셋기킬 것입니다.

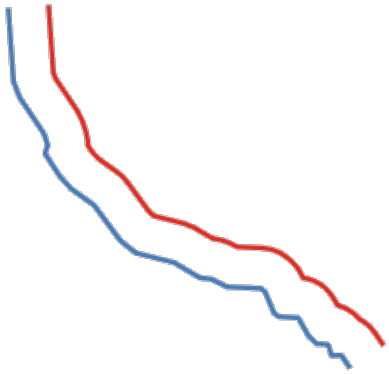


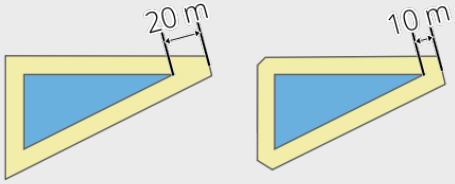
그림 24.125: 소스 피처는 파란색, 오프셋 피처는 빨간색

라인 피처를제자리 피처 수정 할 수 있습니다.

**더 보기**

오프셋 *(평행)* 라인 배열, 이동시키기 *(Translate)*

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | INPUT | [vector: line] | 입력 라인 벡터 레이어 |
| **Distance** | DISTANCE | [numeric: double]<br>기본값: 10.0 | 오프셋 거리입니다. 오른쪽에 있는 데이터 정의 버튼을 클릭해서 반경을 계산할 필드를 선택할 수 있습니다. 이렇게 하면 각 피처별로 서로 다른 반경을 지정할 수 있습니다. (가변 거리 버퍼 참조) |
| **Segments** | SEGMENTS | [numeric: integer]<br>기본값: 8 | 둥글린 (rounded) 오프셋을 생성하는 경우 사분원에 가깝게 만드는 데 사용할 라인 선분의 개수를 제어합니다. |
| **Join style** | JOIN_STYLE | [enumeration]<br>기본값: 0 | 라인에 있는 모서리를 오프셋시킬 때 둥글림, 마이터, 또는 베벨 결합을 사용할지 여부를 지정합니다. 다음 옵션 가운데 하나를 선택할 수 있습니다:<br>• 0 —둥글림 (Round)<br>• 1 —마이터 (Miter)<br>• 2 —베벨 (Bevel)<br>  <br>그림 24.126: 둥글림, 마이터, 베벨 결합 스타일 |
| **Miter limit** | MITER_LIMIT | [numeric: double]<br>기본값: 2.0 | 마이터 결합 스타일에만 적용할 수 있으며, 마이터 결합 생성시 사용할 오프셋 도형으로부터의 최장 거리를 오프셋 거리의 인자로 설정합니다. (최소값은 1.0 입니다.)<br>  <br>그림 24.127: 제한값이 2 인 10m 버퍼와 제한값이 1 인 10m 버퍼 |
| **Offset** | OUTPUT | [vector: line]<br>기본값: [Create temporary layer] | Specify the output (offset) layer. *One of*:<br>• 임시 레이어 생성 (TEMPORARY_OUTPUT)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Offset** | `OUTPUT` | [vector: line] | 산출 (오프셋) 라인 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:offsetline`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 기울어진 최소 경계 상자

입력 레이어의 각 피처에 대해 최소 면적의 기울어진 직사각형을 계산합니다.

> **참고**
>
> For singlepart point features, the output corresponds to the bounding box of the geometry.

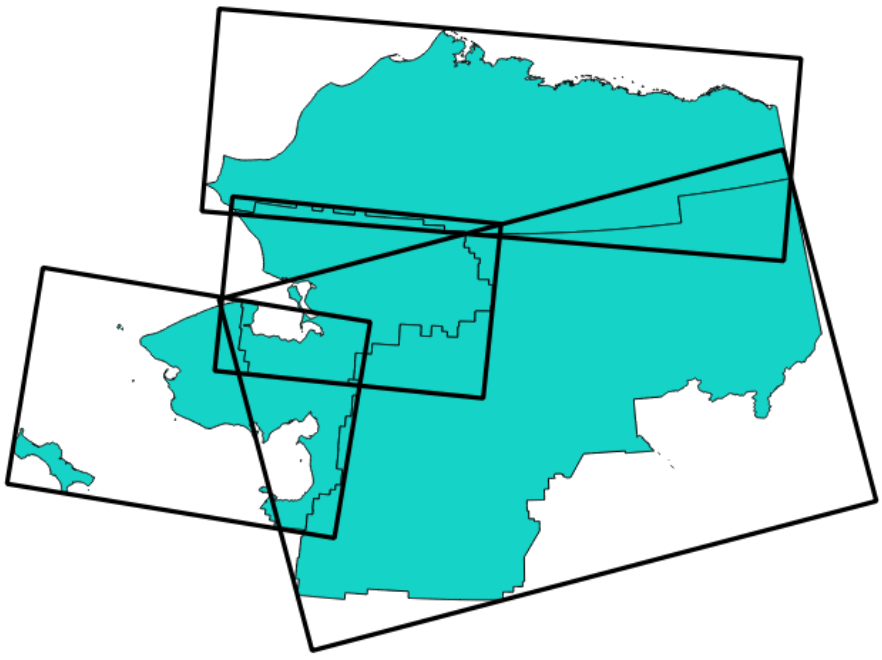


그림 24.128: 기울어진 최소 경계 상자

폴리곤 피처를제자리 피처 수정 할 수 있습니다.

**더 보기**

최소 경계 도형

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | 입력 벡터 레이어 |
| **Bounding boxes** | `OUTPUT` | [vector: polygon]<br>기본값: `[Create temporary layer]` | Specify the output polygon vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Bounding boxes** | `OUTPUT` | [vector: polygon] | The output polygon vector layer. Other than the input attributes, the output layer also contains the following fields: `width`, `height`, `angle`, `area` and `perimeter` of the generated polygon. |

### 파이썬 코드

**알고리즘 ID**: `native:orientedminimumboundingbox`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 직교화

입력 라인 또는 폴리곤 레이어를 직교 (orthogonal) 시키려 합니다. 이 공간 처리 알고리즘은 도형의 꼭짓점들을 이동시켜 도형 안의 모든 각도를 직각 아니면 직선으로 만들려 합니다.

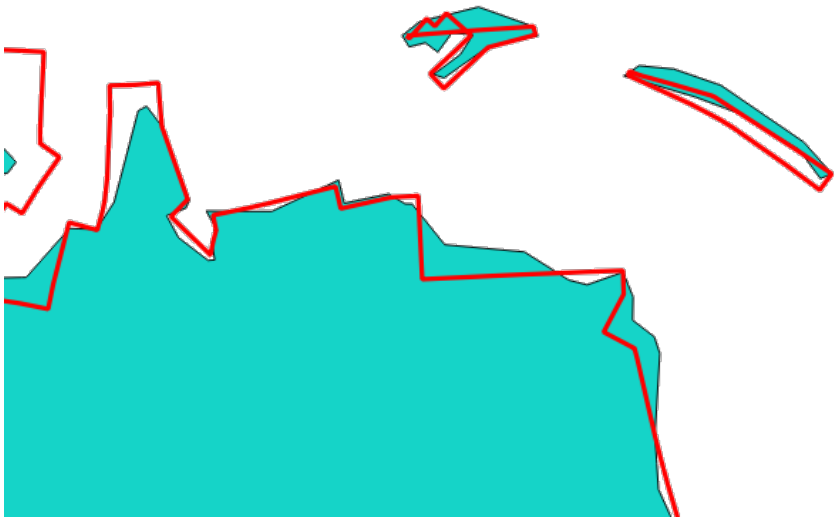


그림 24.129: 파란색: 소스 레이어, 빨간색: 직교화된 산출물

라인 및 폴리곤 피처를제자리 피처 수정 할 수 있습니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: line, polygon] | 입력 라인 또는 폴리곤 벡터 레이어 |
| **Maximum angle tolerance (degrees)** | `ANGLE_TOLERANCE` | [numeric: double] Default: 15.0 | 꼭짓점이 직각 또는 직선으로부터 조정될 수 있는 최대 편차를 지정합니다. 허용 오차가 작을수록 이미 직각에 가까운 꼭짓점만 조정되고, 허용 오차가 클수록 직각에서 더 벗어난 꼭짓점도 조정될 것입니다. |
| **Maximum algorithm iterations** | `MAX_ITERATIONS` | [numeric: integer] 기본값: 1000 | 최대 반복 횟수를 크게 설정할수록 처리 시간은 더 걸리지만 더 직교 도형에 가까운 도형을 산출할 것입니다. |

다음 페이지에 계속

표 24.692 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Orthogonalized** | `OUTPUT` | [same as input]<br>기본값: `[Create temporary layer]` | Specify the output polygon vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Orthogonalized** | `OUTPUT` | [same as input] | 조정된 각도를 가진 산출 폴리곤 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:orthogonalize`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 표면에 포인트 생성하기

입력 레이어의 각 피처에 대해 피처 도형의 표면에 놓여지게 될 포인트를 반환합니다.

포인트 피처를제자리 피처 수정 할 수 있습니다.

**경고**

This algorithm may drop existing primary keys or FID values and regenerate them in output layers, depending on the input parameters.

**더 보기**

중심

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | 입력 벡터 레이어 |
| **Create point on surface for each part** | `ANGLE_TOLERANCE` | [boolean] | 이 파라미터를 활성화하면 도형의 각 부분에 대해 포인트를 생성할 것입니다. |
| **Point** | `OUTPUT` | [vector: point] 기본값: `[Create temporary layer]` | Specify the output point vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Point** | `OUTPUT` | [vector: point] | 산출 포인트 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:pointonsurface`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 도형을 따라 포인트 생성하기

라인 또는 폴리곤 도형을 따라 규칙적인 간격으로 포인트를 생성합니다. 생성된 포인트는 도형을 따라 측정한 거리 및 해당 포인트에서 라인의 각도를 담고 있는 새 속성들을 가지게 될 것입니다.

시작 및 종단 오프셋을 지정할 수 있는 옵션도 있습니다. 이 옵션은 포인트를 도형의 시작 및 종단에서 얼마나 떨어져 생성해야 할지를 제어합니다.

**⚠ 경고**

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

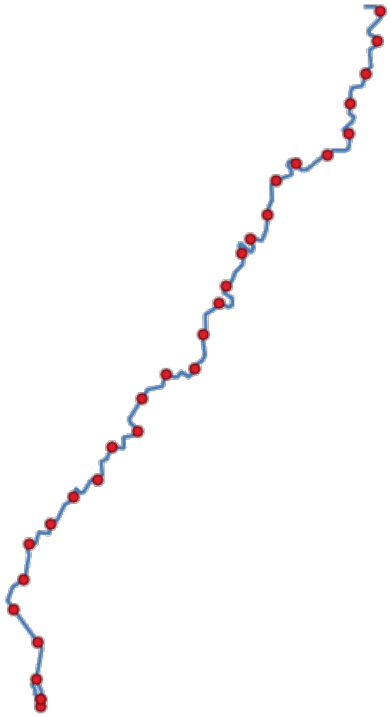


그림 24.130: 소스 라인 레이어를 따라 생성된 포인트

**더 보기**

라인에 포인트를 보간하기

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: line, polygon] | 입력 라인 또는 폴리곤 벡터 레이어 |
| **Distance** | `DISTANCE` | [numeric: double]<br>기본값: 1.0 | 라인을 따라 연속된 두 포인트 사이의 거리 |
| **Start offset** | `START_OFFSET` | [numeric: double]<br>기본값: 0.0 | 첫 번째 포인트의 위치를 나타내는 입력 라인 시작점에서의 거리 |
| **End offset** | `END_OFFSET` | [numeric: double]<br>기본값: 0.0 | 더 이상 포인트 피처를 생성해서는 안 되는 위치를 나타내는 입력 라인 종단점에서의 거리 |

다음 페이지에 계속

표 24.696 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Interpolated points** | `OUTPUT` | [vector: point]<br>기본값: `[Create temporary layer]` | Specify the output vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Interpolated points** | `OUTPUT` | [vector: point] | Point vector layer with features placed along lines or polygon boundaries of the input layer. Other than the input attributes on every related point, the output layer also contains the following fields: `distance` and `angle` of the generated point. |

### 파이썬 코드

**알고리즘 ID**: `native:pointsalonglines`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 포인트 변위시키기

근접 (proximity) 거리를 지정, 근접한 포인트 피처들을 식별해서 피처들의 무게중심을 중심으로 하는 원 위에 방사상으로 분포시킵니다. 중첩된 피처들을 흩트릴 수 있는 편리한 도구입니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: point] | 입력 포인트 벡터 레이어 |
| **Minimum distance to other points** | `PROXIMITY` | [numeric: double]<br>기본값: 1.0 | 이 거리 미만의 포인트 피처들을 근접하다고 판단합니다. 근접한 피처들을 함께 분포시킵니다. |
| **Displacement distance** | `DISTANCE` | [numeric: double]<br>기본값: 1.0 | 근접한 피처들을 배치할 원의 반경 |
| **Horizontal distribution for two point case** | `HORIZONTAL` | [boolean]<br>기본값: False | 단 2 개의 포인트만 근접해 있다고 식별된 경우, 원 위에 수직이 아닌 수평으로 정렬합니다. |
| **Displaced** | `OUTPUT` | [vector: point]<br>기본값: `[Create temporary layer]` | Specify the output vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Displaced** | `OUTPUT` | [vector: point] | 입력 포인트 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `qgis:pointsdisplacement`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 도달불능극

폴리곤 레이어에 대해 도달불능극 (pole of inaccessibility) 을 계산합니다. 도달불능극이란 표면의 경계선에서 가장 멀리 있는 내부 포인트를 말합니다.

이 알고리즘은 블라디미르 아카폰킨 (Vladimir Agafonkin) 이 2016 년 개발한, 지정한 허용 오차 내에서 진정한 도달불능극 발견을 보장하는 반복적인 접근법인'polylabel'알고리즘을 사용합니다. 허용 오차가 더 정밀할수록 (값이 적을수록) 반복을 더 하게 되기 때문에 계산 시간이 더 걸릴 것입니다.



그림 24.131: 도달불능극

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: polygon] | 입력 벡터 레이어 |
| **Tolerance** | `TOLERANCE` | [numeric: double]<br>기본값: 1.0 | 계산을 위한 허용 오차를 설정합니다. |
| **Point** | `OUTPUT` | [vector: point]<br>기본값: `[Create temporary layer]` | Specify the output polygon vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Point** | `OUTPUT` | [vector: point] | The output point vector layer. It will contain a new attribute with the distance `dist_pole` from the calculated pole to the polygon boundary. |

### 파이썬 코드

**알고리즘 ID**: `native:poleofinaccessibility`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 폴리곤화

**닫힌 (closed)** 피처를 가진 라인 레이어로부터 피처 경계선을 생성한 폴리곤 레이어를 생성합니다.

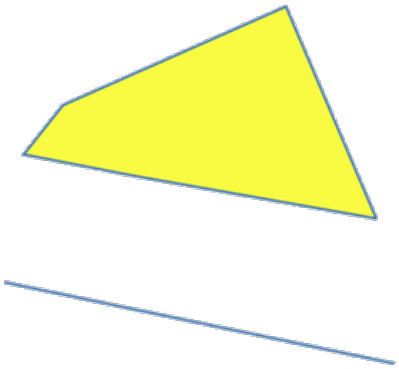


그림 24.132: 닫힌 라인으로부터 생성된 노란색 폴리곤

**참고**

폴리곤으로 변형되기 위해서는 라인 레이어가 닫힌 도형을 가지고 있어야만 합니다.

**더 보기**

폴리곤을 라인으로, 라인을 폴리곤으로, 도형 유형 변환하기

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: line] | 입력 라인 벡터 레이어 |
| **Keep fields from the input layer** | `KEEP_FIELDS` | [boolean]<br>기본값: False | 입력 레이어의 필드를 (값이 아닌 테이블 구조만) 유지하려면 체크하십시오. |
| **Polygons from lines** | `OUTPUT` | [vector: polygon]<br>기본값: `[Create temporary layer]` | Specify the output polygon vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Polygons from lines** | `OUTPUT` | [vector: polygon] | 라인으로부터 생성한 산출 폴리곤 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:polygonize`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 폴리곤을 라인으로

폴리곤 레이어를 받아 입력 레이어의 폴리곤의 경계선을 나타내는 라인으로 라인 레이어를 생성합니다.

산출 레이어의 속성 테이블은 입력 레이어의 속성 테이블과 동일합니다.

**기본 메뉴**: *Vector → Geometry Tools*

**더 보기**

라인을 폴리곤으로, 폴리곤화, 도형 유형 변환하기



그림 24.133: 알고리즘이 산출한 검은색 라인

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: polygon] | 입력 폴리곤 벡터 레이어 |
| **Lines** | `OUTPUT` | [vector: line]<br>기본값: `[Create temporary layer]` | Specify the output line vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Lines** | `OUTPUT` | [vector: line] | 폴리곤으로부터 생성한 산출 라인 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:polygonstolines`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 포인트 투영하기 (데카르트)

지정한 거리 및 방향 (방위각) 으로 포인트 도형을 투영합니다.

✔ 포인트 피처를제자리 피처 수정 할 수 있습니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: point] | 입력 포인트 벡터 레이어 |
| **Bearing (degrees from North)** | `BEARING` | [numeric: double] 기본값: 0.0 | 북쪽에 시작하는 시계 방향 도 (°) 단위 각도 |
| **Distance** | `DISTANCE` | [numeric: double] 기본값: 1.0 | 도형을 오프셋시킬 레이어 단위 거리 |
| **Projected** | `OUTPUT` | [vector: point] 기본값: `[Create temporary layer]` | Specify the output point vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Projected** | `OUTPUT` | [vector: point] | 산출 (투영된) 포인트 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:projectpointcartesian`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 다중 부분으로 조성하기

단일 부분 도형을 가진 벡터 레이어를 받아 모든 도형이 다중 부분인 새 벡터 레이어를 생성합니다.

이미 다중 부분인 입력 피처는 변경되지 않을 것입니다.

이 알고리즘은 다중 부분 피처를 요구하는 데이터 제공자와 호환시키기 위해 도형을 다중 부분 유형으로 강제 변환시키는 데 사용할 수 있습니다.

포인트, 라인 및 폴리곤 피처를제자리 피처 수정 할 수 있습니다.

더 보기

집계, 도형 모으기, *Convert to strictly multipart*

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | 입력 벡터 레이어 |
| **Multiparts** | `OUTPUT` | [same as input]<br>기본값: `[Create temporary layer]` | Specify the output multipart vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Multiparts** | `OUTPUT` | [same as input] | 산출 다중 부분 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:promotetomulti`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 직사각형, 타원형, 마름모

입력 포인트 레이어의 각 피처에 대해 직사각형, 타원형, 마름모 형태의 버퍼 영역을 생성합니다.

형태 파라미터를 모든 피처에 대해 고정할 수도 있고, 아니면 필드 또는 표현식을 사용해서 동적으로 설정할 수도 있습니다.

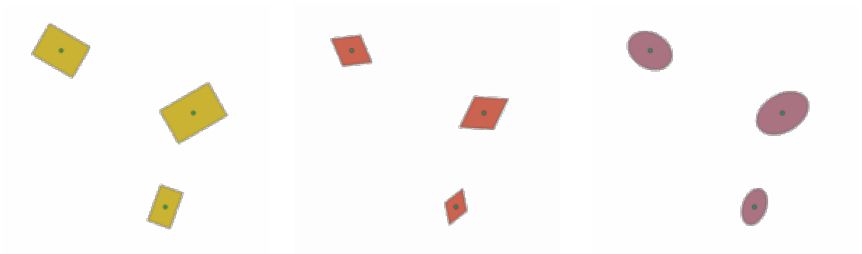


그림 24.134: 동적 파라미터를 가진 서로 다른 버퍼 형태

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: point] | 입력 포인트 벡터 레이어 |
| **Buffer shape** | `SHAPE` | [enumeration] | 사용할 형태를 지정합니다. 다음 가운데 하나를 선택할 수 있습니다:<br>• 0 —직사각형<br>• 1 —타원형 (oval)<br>• 2 —마름모꼴 |
| **Width** | `WIDTH` | [numeric: double]<br>기본값: 1.0 | 버퍼 형태의 너비 |
| **Height** | `HEIGHT` | [numeric: double]<br>기본값: 1.0 | 버퍼 형태의 높이 |
| **Rotation**<br>선택적 | `ROTATION` | [numeric: double]<br>기본값: 0.0 | 버퍼 형태의 기울기 |
| **Number of segments** | `SEGMENTS` | [numeric: integer]<br>기본값: 36 | 완전한 원을 위한 선분의 개수 (타원형 형태) |
| **Output** | `OUTPUT` | [vector: polygon]<br>기본값: `[Create temporary layer]` | Specify the output vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output** | `OUTPUT` | [vector: polygon] | 버퍼 형태를 가진 산출 벡터 레이어 |

### 파이썬 코드



알고리즘 **ID**: `native:rectanglesovalsdiamonds`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

### 복제 꼭짓점 제거하기

꼭짓점 제거가 도형의 무결성을 무너뜨리지 않는 선에서, 도형에서 복제 꼭짓점을 제거합니다.

허용 오차 파라미터는 꼭짓점들이 동일한지 여부를 결정할 때 사용할 좌표에 대한 허용 오차를 지정합니다.

기본적으로 복제 꼭짓점 탐지 과정에서 Z 값을 고려하지는 않습니다. 예를 들어 X 및 Y 좌표는 동일하지만 Z 값이 다른 두 꼭짓점은 그대로 복제 꼭짓점으로 식별되어 하나를 제거할 것입니다. *Use Z Value* 파라미터를 참으로 설정한 경우에만 Z 값을 검증해서 X 및 Y 좌표가 동일하지만 Z 값은 다른 꼭짓점들을 남길 것입니다.

포인트, 라인 및 폴리곤 피처를제자리 피처 수정 할 수 있습니다.

**참고**

다중 부분 도형의 서로 다른 부분에 있는 복제 꼭짓점은 검사하지 않습니다. 예를 들면 이 알고리즘은 중첩되는 포인트를 가진 다중 부분 도형을 변경하지 않을 것입니다.

**더 보기**

꼭짓점 추출하기, 특정 꼭짓점 추출하기, 복제 도형 삭제하기, *Delete duplicated vertices*

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | 입력 벡터 레이어 |
| **Tolerance** | `TOLERANCE` | [numeric: double]<br>기본값: 0.000001 | 지정한 거리보다 가까이 있는 꼭짓점들을 복제 꼭짓점으로 간주합니다. |
| **Use Z value** | `USE_Z_VALUE` | [boolean]<br>기본값: False | *Use Z Value* 파라미터를 참으로 설정한 경우, Z 값도 검사해서 X 및 Y 좌표가 동일하지만 Z 값은 다른 꼭짓점들을 그대로 남길 것입니다. |
| **Cleaned** | `OUTPUT` | [same as input]<br>기본값: `[Create temporary layer]` | Specify the output vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Cleaned** | `OUTPUT` | [same as input] | 복제 꼭짓점이 없는 산출 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:removeduplicatevertices`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## NULL 도형 제거하기

벡터 레이어에서 도형을 보유하지 않는 객체를 모두 제거합니다. 다른 모든 객체는 그대로 복제될 것입니다.

NULL 도형을 가진 피처를 개별 레이어로 저장할 수 있습니다.

*Also remove empty geometries* 옵션을 체크한 경우, 이 알고리즘은 비어 있는 도형, 즉 좌표가 없는 도형을 가진 객체를 제거합니다. 이런 경우, NULL 산출물도 이 옵션을 반영해서 NULL 및 비어 있는 도형을 둘 다 담게 될 것입니다.

**더 보기**

복제 도형 삭제하기

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | NULL 이 아닌 도형을 가진 입력 벡터 레이어 |
| **Also remove empty geometries** | `REMOVE_EMPTY` | [boolean] | |
| **Non null geometries** | `OUTPUT`<br>선택적 | [same as input]<br>기본값: `[Create temporary layer]` | Specify the output vector layer for the non-NULL (and non-empty) geometries. *One of*:<br>• 산출물 건너뛰기<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

다음 페이지에 계속

표 24.714 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Null geometries**<br>선택적 | NULL_OUTPUT | [same as input]<br>기본값: [Skip output] | Specify the output vector layer for the NULL (and empty) geometries. *One of*:<br>• 산출물 건너뛰기<br>• 임시 레이어 생성 (TEMPORARY_OUTPUT)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Null geometries** | NULL_OUTPUT | [same as input] | 산출 벡터 레이어 (NULL 도형 및 선택한 경우 비어 있는 도형도) |
| **Non null geometries** | OUTPUT | [same as input] | 산출 벡터 레이어 (NULL 도형 및 선택한 경우 비어 있는 도형도 없는) |

### 파이썬 코드

**알고리즘 ID**: native:removenullgeometries

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 라인 방향 반전시키기

라인 레이어의 방향을 반전시킵니다.

☑ 라인 피처를제자리 피처 수정 할 수 있습니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | INPUT | [vector: line] | 입력 라인 벡터 레이어 |

다음 페이지에 계속

표 24.716 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Reversed** | `OUTPUT` | [vector: line]<br>기본값: `[Create temporary layer]` | Specify the output line vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Reversed** | `OUTPUT` | [vector: line] | 반전된 라인을 담은 산출 라인 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:reverselinedirection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 기울이기

피처 도형을 시계 방향의 지정한 각도로 기울입니다. 각 피처의 중심 (centroid) 을 기준으로, 또는 사전 설정한 유일 포인트를 기준으로 기울입니다.

포인트, 라인 및 폴리곤 피처를제자리 피처 수정 할 수 있습니다.

**더 보기**

이동시키기 *(Translate)*, *X* 좌표와 *Y* 좌표 바꾸기

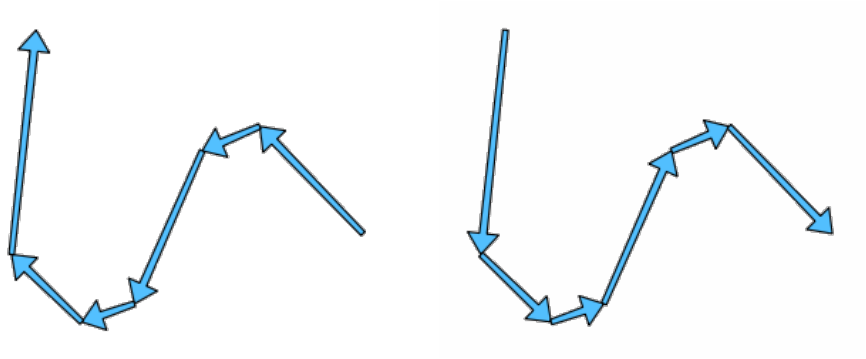


그림 24.135: 방향 반전 이전과 이후

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | 입력 벡터 레이어 |
| **Rotation (degrees clockwise)** | `ANGLE` | [numeric: double]<br>기본값: 0.0 | 기울기의 도 단위 각도 |
| **Rotation anchor point (x, y)** 선택적 | `ANCHOR` | [coordinates]<br>기본값: 설정하지 않음 | 피처 기울이기의 기준이 될 포인트의 X, Y 좌표. 이 파라미터를 설정하지 않는 경우 각 피처의 중심을 기준으로 기울입니다. |
| **Rotated** | `OUTPUT` | [same as input]<br>기본값: `[Create temporary layer]` | Specify the output vector layer (with rotated geometries). *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Rotated** | `OUTPUT` | [same as input] | 기울인 도형을 담은 산출 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:rotatefeatures`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면 공간 처리 툴박스 알고리즘 사용 을 참조하세요.

### 둥글기

Calculates the roundness of each feature and stores it as a new field `roundness`. The input vector layer must contain polygons.

폴리곤의 둥글기는"4π × 폴리곤의 면적/둘레 ²"로 정의됩니다. 둥글기는 0 에서 1 사이의 값으로, 완벽한 원의 둥글기는 1 이고 완전히 평평한 폴리곤의 둥글기는 0 입니다.

> **참고**
>
> 다중부분 폴리곤 피처의 경우 이 알도리즘은 NULL 을 반환합니다.

☑ 폴리곤 피처를제자리 피처 수정 할 수 있습니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: polygon] | 입력 벡터 레이어 |
| **Roundness** | `OUTPUT` | [vector: polygon]<br>기본값: `[Create temporary layer]` | Specify the output vector layer (with roundness field). *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Roundness** | `OUTPUT` | [same as input] | The output vector layer with `roundness` value in a field |

### 파이썬 코드

**알고리즘 ID**: `native:roundness`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 최대 각도로 선분화

도형의 만곡 구간을 선형 구간으로 변환시켜 도형을 선분화시킵니다.

직선화된 도형에 있는 꼭짓점들 사이의 최대 허용 반경 각도를 지정해서 선분화를 수행합니다. (예를 들면 원본 원호로부터 생성한 원호의 각도를 선형화된 도형에 있는 연속된 두 산출 꼭짓점의 중선 (□□) 으로 삼는 식입니다.) 곡선 구간이 없는 도형은 변경하지 않을 것입니다.

**더 보기**

최대 거리로 선분화, 단순화, 평탄화

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: line, polygon] | 입력 라인 또는 폴리곤 벡터 레이어 |
| **Maximum angle between vertices (degrees)** | `ANGLE` | [numeric: double]<br>기본값: 5.0 | 직선화된 도형에 있는 연속된 두 꼭짓점 사이의 최대 허용 반경 각도 |
| **Segmentized** | `OUTPUT` | [same as input]<br>기본값: `[Create temporary layer]` | Specify the output vector layer (with segmentized geometries). *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Segmentized** | `OUTPUT` | [same as input] | 선분화된 도형을 담은 산출 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:segmentizebymaxangle`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 최대 거리로 선분화

도형의 만곡 구간을 선형 구간으로 변환시켜 도형을 선분화시킵니다.

원본 곡선과 선분화된 표현 사이의 최대 허용 오프셋 거리를 지정해서 선분화를 수행합니다. 만곡 도형이 아닌 도형은 변경하지 않을 것입니다.

**더 보기**

최대 각도로 선분화, 단순화, 평탄화

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: line, polygon] | 입력 라인 또는 폴리곤 벡터 레이어 |
| **Maximum offset distance** | `DISTANCE` | [numeric: double]<br>기본값: 1.0 | 원본 곡선과 선분화된 표현 사이의 최대 허용 오프셋 거리를 레이어 단위로 지정합니다. |
| **Segmentized** | `OUTPUT` | [same as input]<br>기본값: `[Create temporary layer]` | Specify the output vector layer (with segmentized geometries). *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Segmentized** | `OUTPUT` | [same as input] | 선분화된 도형을 담은 산출 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:segmentizebymaxdistance`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### M 값 설정하기

레이어에 있는 도형에 M 값을 설정합니다.

M 값이 레이어에 이미 존재하는 경우, 새 값으로 덮어쓸 것입니다. M 값이 없는 경우, 모든 도형에 대해 지정한 값을 초기 M 값으로 사용해서 M 값을 가지도록 도형을 업그레이드할 것입니다.

M 좌표 활성화된 포인트, 라인 및 폴리곤 피처를제자리 피처 수정 할 수 있습니다.

**팁**

Identify Features 버튼을 사용해서 M 값을 확인하고 추가하십시오. *Identify Results* 대화창에서 결과를 볼 수 있습니다.

**더 보기**

래스터로부터 *M* 값 설정하기, *Z* 값 설정하기, *M/Z* 값 제거

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | 입력 벡터 레이어 |
| **M Value** | `M_VALUE` | [numeric: double]<br>기본값: 0.0 | 피처 도형에 할당할 M 값 |
| **M Added** | `OUTPUT` | [same as input]<br>기본값: `[Create temporary layer]` | Specify the output vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **M Added** | `OUTPUT` | [same as input] | 도형에 M 값을 할당한 산출 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:setmvalue`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 래스터로부터 M 값 설정하기

래스터 레이어의 밴드에서 샘플링한 값을 사용해서 피처 도형의 중첩하는 모든 꼭짓점에 M 값을 설정합니다. 래스터 값을 사전 설정한 척도로 조정할 수 있는 옵션도 있습니다.

M 값이 레이어에 이미 존재하는 경우, 새 값으로 덮어쓸 것입니다. M 값이 없는 경우, M 값을 가지도록 도형을 업그레이드할 것입니다.

M 좌표 활성화된 포인트, 라인 및 폴리곤 피처를제자리 피처 수정 할 수 있습니다.

**더 보기**

늘어뜨리기 *(래스터로부터 Z 값 설정하기)*, *M 값 설정하기*

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | 입력 벡터 레이어 |
| **Raster layer** | `RASTER` | [raster] | M 값을 가진 래스터 레이어 |
| **Band number** | `BAND` | [raster band]<br>기본값: 1 | M 값을 가져올 래스터 밴드 |
| **Value for NoData or non-intersecting vertices** | `NODATA` | [numeric: double]<br>기본값: 0.0 | 꼭짓점이 래스터 (의 무결한 픽셀) 와 교차하지 않는 경우 사용할 값 |
| **Scale factor** | `SCALE` | [numeric: double]<br>기본값: 1.0 | 척도 값: 밴드 값을 이 값으로 곱합니다. |
| **Offset** | `OFFSET` | [numeric: double]<br>기본값: 0.0 | 오프셋 값: “척도 (scale factor)”를 적용한 다음 밴드 값에 대수 (□□) 적으로 추가됩니다. |

다음 페이지에 계속

표 24.728 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Updated** | `OUTPUT` | [same as input]<br>기본값: `[Create temporary layer]` | Specify the output vector layer (with updated M values). *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Updated** | `OUTPUT` | [same as input] | 업데이트된 M 값을 가진 산출 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:setmfromraster`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## Z 값 설정하기

레이어에 있는 도형에 Z 값을 설정합니다.

Z 값이 레이어에 이미 존재하는 경우, 새 값으로 덮어쓸 것입니다. Z 값이 없는 경우, 모든 도형에 대해 지정한 값을 초기 Z 값으로 사용해서 Z 값을 가지도록 도형을 업그레이드할 것입니다.

☑ Z 좌표 활성화된 포인트, 라인 및 폴리곤 피처를제자리 피처 수정 할 수 있습니다.

**팁**

Identify Features 버튼을 사용해서 Z 값을 확인하고 추가하십시오. *Identify Results* 대화창에서 결과를 볼 수 있습니다.

**더 보기**

늘어뜨리기 *(래스터로부터 Z 값 설정하기)*, *M 값 설정하기*, *M/Z 값 제거*

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | 입력 벡터 레이어 |
| **Z Value** | `Z_VALUE` | [numeric: double] 기본값: 0.0 | 피처 도형에 할당할 Z 값 |
| **Z Added** | `OUTPUT` | [same as input] 기본값: `[Create temporary layer]` | Specify the output vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Z Added** | `OUTPUT` | [same as input] | Z 값을 할당한 산출 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:setzvalue`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 단순화

라인 또는 폴리곤 레이어에 있는 도형을 단순화합니다. 입력 레이어에 있는 피처와 동일한 피처를 담고 있지만, 도형이 더 적은 꼭짓점을 가지고 있는 새 레이어를 생성합니다.

이 알고리즘은 거리 기반 (“더글러스-포이커 (Douglas-Peucker)”알고리즘), 면적 기반 (“비쉬왈링감 (Visvalingam)” 알고리즘) 및 도형을 그리드에 스냅하기를 포함하는 여러 단순화 메소드를 선택할 수 있습니다.

라인 및 폴리곤 피처를제자리 피처 수정 할 수 있습니다.

**기본 메뉴**: *Vector → Geometry Tools*



그림 24.136: 왼쪽에서 오른쪽으로: 소스 레이어와 증가하는 단순화 허용 오차

**더 보기**

*Simplify coverage*, 평탄화, 개수로 치밀화하기, 간격으로 치밀화하기

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: line, polygon] | 입력 라인 또는 폴리곤 벡터 레이어 |
| **Simplification method** | `METHOD` | [enumeration] 기본값: 0 | 단순화 메소드를 설정합니다. 다음 가운데 하나를 선택할 수 있습니다:<br>• 0 —거리 (더글라스-포이커)<br>• 1 —그리드에 스냅<br>• 2 —면적 (비쉬왈링감) |
| **Tolerance** | `TOLERANCE` | [numeric: double] 기본값: 1.0 | 한계값 허용 오차 (레이어 단위): 노드 2 개 사이의 거리가 허용 오차 값보다 짧을 경우, 해당 선분을 단순화하고 꼭짓점을 제거할 것입니다. |
| **Simplified** | `OUTPUT` | [same as input] 기본값: `[Create temporary layer]` | Specify the output (simplified) vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Simplified** | `OUTPUT` | [same as input] | 산출 (단순화된) 벡터 레이어 |

### 한쪽 버퍼 생성하기

라인의 한쪽에만 지정된 거리로 버퍼를 계산합니다.
버퍼는 언제나 폴리곤 레이어로 산출됩니다.



그림 24.137: 동일 라인 벡터 레이어의 왼쪽 대 오른쪽 버퍼

**더 보기**

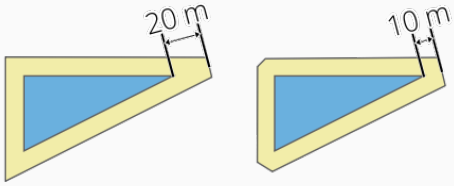
버퍼

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: line] | 입력 라인 벡터 레이어 |
| **Distance** | `DISTANCE` | [numeric: double]<br>기본값: 10.0 | 버퍼 거리 |
| **Side** | `SIDE` | [enumeration]<br>기본값: 0 | 버퍼를 생성할 쪽을 지정합니다. 다음 가운데 하나를 선택할 수 있습니다:<br>• 0 – 왼쪽<br>• 1 – 오른쪽 |
| **Segments** | `SEGMENTS` | [numeric: integer]<br>기본값: 8 | 둥글린 (rounded) 오프셋을 생성하는 경우 사분원에 가깝게 만드는 데 사용할 라인 선분의 개수를 제어합니다. |

다음 페이지에 계속

표 24.734 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Join style** | `JOIN_STYLE` | [enumeration] 기본값: 0 | 라인에 있는 모서리를 오프셋시킬 때 둥글림, 마이터, 또는 베벨 결합을 사용할지 여부를 지정합니다. 다음 옵션 가운데 하나를 선택할 수 있습니다:<br>• 0 —둥글림 (Round)<br>• 1 —마이터 (Miter)<br>• 2 —베벨 (Bevel)<br>  <br>그림 24.138: 둥글림, 마이터, 베벨 결합 스타일 |
| **Miter limit** | `MITER_LIMIT` | [numeric: double] 기본값: 2.0 | 마이터 결합 스타일에만 적용할 수 있으며, 마이터 결합 생성시 사용할 오프셋 도형으로부터의 최장 거리를 오프셋 거리의 인자로 설정합니다. (최소값은 1.0 입니다.)<br>  <br>그림 24.139: 제한값이 2 인 10m 버퍼와 제한값이 1 인 10m 버퍼 |
| **Buffer** | `OUTPUT` | [vector: polygon] 기본값: `[Create temporary layer]` | Specify the output (buffer) layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Buffer** | `OUTPUT` | [vector: polygon] | 산출 (버퍼) 폴리곤 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:singlesidedbuffer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 평탄화

라인 또는 폴리곤 레이어의 피처 도형에 **꼭짓점과 모서리** 를 추가해서 도형을 평탄화 (smooth) 합니다.

반복 파라미터가 각 도형에 평탄화 작업을 몇 번 반복할지 결정합니다. 반복 횟수가 늘어날수록 도형은 더 매끄러워지지만 도형의 노드 개수가 기하급수적으로 늘어날 것입니다.



그림 24.140: 반복 횟수를 늘릴수록 더 매끄러워지는 도형

오프셋 파라미터는 평탄화된 도형이 얼마나 원본 도형을"엄중하게"따르는지 제어합니다. 값이 작을수록 더 엄중하게 따르고, 값이 클수록 더 엉성하게 따를 것입니다.

최대 각도 파라미터를 사용하면 노드를 큰 각도로 평탄화하는 일을 막을 수 있습니다. 노드 양쪽의 선분이 이루는 각도가 이 값보다 큰 경우, 해당 노드를 평탄화하지 않을 것입니다. 예를 들어 최대 각도를 90 도 이하로 설정하면 도형에 있는 직각을 유지할 것입니다.

☑ 라인 및 폴리곤 피처를제자리 피처 수정 할 수 있습니다.

**더 보기**

단순화, *Simplify coverage*, 개수로 치밀화하기, 간격으로 치밀화하기

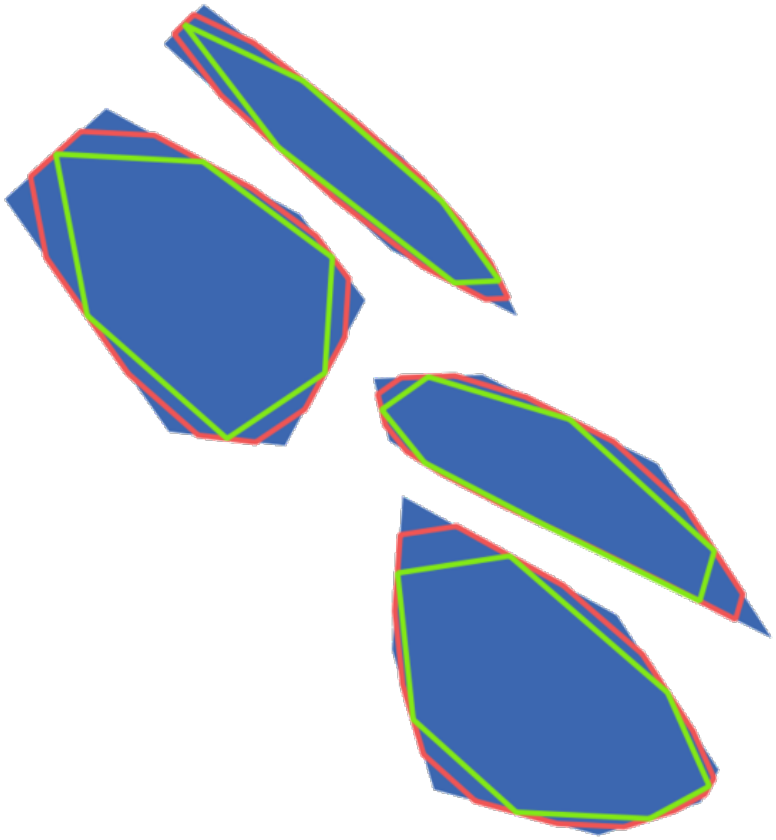


그림 24.141: 파란색은 입력 레이어, 빨간 라인은 오프셋 값 0.25, 녹색 라인은 오프셋 값 0.50

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: line, polygon] | 입력 라인 또는 폴리곤 벡터 레이어 |
| **Iterations** | `ITERATIONS` | [numeric: integer]<br>기본값: 1 | 반복 횟수를 늘리면 더 매끄러운 도형을 (더 많은 꼭짓점과 함께) 산출합니다. |
| **Offset** | `OFFSET` | [numeric: double]<br>기본값: 0.25 | 이 값을 늘리면 입력 라인/경계선으로부터 평탄화된 라인/경계선을 더 멀리 이동시킬 것입니다. |
| **Maximum node angle to smooth** | `MAX_ANGLE` | [numeric: double]<br>기본값: 180.0 | 이 각도 값 미만의 노드를 전부 평탄화할 것입니다. |
| **Smoothed** | `OUTPUT` | [same as input]<br>기본값: `[Create temporary layer]` | Specify the output (smoothed) layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Smoothed** | `OUTPUT` | [same as input] | 산출 (평탄화된) 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:smoothgeometry`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 레이어에 도형을 스냅시키기

레이어의 도형을 또다른 레이어의 도형에, 아니면 동일 레이어의 다른 도형에 스냅시킵니다.

허용 오차 거리를 바탕으로 연결 작업을 수행하고, 도형을 참조 도형과 일치시키기 위해 필요한만큼 꼭짓점을 삽입하거나 제거할 것입니다.

포인트, 라인 및 폴리곤 피처를제자리 피처 수정 할 수 있습니다.

**더 보기**

그리드에 포인트를 스냅시키기

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | 입력 벡터 레이어 |
| **Reference layer** | `REFERENCE_LAYER` | [vector: geometry] | 스냅시킬 벡터 레이어 |
| **Tolerance** | `TOLERANCE` | [numeric: double]<br>기본값: 10.0 | 입력 꼭짓점을 스냅시키려면 참조 레이어의 도형에 얼마나 가까워야 하는지 제어합니다. |

다음 페이지에 계속

표 24.738 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Behavior** | BEHAVIOR | [enumeration]<br>기본값: 0 | 기존 노드 또는 (이동시킬 꼭짓점에 가장 근접한 포인트를 가진) 선분에 스냅시킬 수 있습니다. 다음 스냅 옵션 가운데 선택할 수 있습니다:<br>• 0 —노드를 정렬시키는 편을 선호, 필요한 위치에 추가 꼭짓점을 삽입<br>선분이 노드보다 가까이 있어도 노드에 스냅하는 편을 선호합니다. 허용 가능한 오차 범위 내에 있을 경우 도형들이 서로를 정확히 따르게 만들기 위해 새 노드를 삽입할 것입니다.<br>• 1 —최근접 포인트를 선호, 필요한 위치에 추가 꼭짓점을 삽입<br>노드인지 선분인지 상관없이 가장 가까운 포인트에 스냅합니다. 허용 가능한 오차 범위 내에 있을 경우 도형들이 서로를 정확히 따르게 만들기 위해 새 노드를 삽입할 것입니다.<br>• 2 —노드를 정렬시키는 편을 선호, 새 꼭짓점 삽입 안 함<br>선분이 노드보다 가까이 있어도 노드에 스냅하는 편을 선호합니다. 새 노드를 삽입하지 않을 것입니다.<br>• 3 —최근접 포인트를 선호, 새 꼭짓점 삽입 안 함<br>노드인지 선분인지 상관없이 가장 가까운 포인트에 스냅합니다. 새 노드를 삽입하지 않을 것입니다.<br>• 4 —종단 포인트만 이동, 노드 정렬 선호<br>라인의 시작/종단 포인트에만 스냅합니다. (포인트 피처에도 스냅할 것이고, 폴리곤 피처는 수정되지 않을 것입니다.) 노드에 스냅하는 편을 선호합니다.<br>• 5 —종단 포인트만 이동, 최근접 포인트 선호<br>라인의 시작/종단 포인트에만 스냅합니다. (포인트 피처에도 스냅할 것이고, 폴리곤 피처는 수정되지 않을 것입니다.) 가장 가까운 포인트에 스냅합니다.<br>• 6 —종단 포인트와 종단 포인트만 스냅<br>라인의 시작/종단 포인트를 다른 라인의 시작/종단 포인트에만 스냅합니다.<br>• 7 —고정 노드 (anchor node) 에 스냅 (단일 레이어에서만) |

다음 페이지에 계속

표 24.738 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Snapped geometry** | `OUTPUT` | [same as input]<br>기본값: `[Create temporary layer]` | Specify the output (snapped) layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Snapped geometry** | `OUTPUT` | [same as input] | 산출 (스냅된) 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:snapgeometries`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 그리드에 포인트를 스냅시키기

벡터 레이어에 있는 도형의 좌표를 수정해서 그리드의 최근접 포인트에 모든 포인트 또는 꼭짓점을 스냅시킵니다.

스냅된 도형을 계산할 수 없는 (또는 도형이 완전히 무너진) 경우 피처의 도형을 제거할 것입니다.

X, Y, Z 또는 M 축 상에서 스냅 작업을 수행할 수 있습니다. 어떤 축에 대해 그리드 간격이 0 인 경우, 해당 축에서 스냅 작업이 비활성화될 것입니다.

포인트, 라인 및 폴리곤 피처를제자리 피처 수정 할 수 있습니다.

**경고**

그리드에 스냅시키면 일부 모서리에서 무결하지 않은 도형을 생성할 수도 있습니다.

**더 보기**

레이어에 도형을 스냅시키기

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | 입력 벡터 레이어 |
| **X Grid Spacing** | `HSPACING` | [numeric: double] 기본값: 1.0 | X 축 상의 그리드 간격 |
| **Y Grid Spacing** | `VSPACING` | [numeric: double] 기본값: 1.0 | Y 축 상의 그리드 간격 |
| **Z Grid Spacing** | `ZSPACING` | [numeric: double] 기본값: 0.0 | Z 축 상의 그리드 간격 |
| **M Grid Spacing** | `MSPACING` | [numeric: double] 기본값: 0.0 | M 축 상의 그리드 간격 |
| **Snapped** | `OUTPUT` | [same as input] 기본값: `[Create temporary layer]` | Specify the output (snapped) layer. *One of*: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장… • GeoPackage 로 저장… • 데이터베이스 테이블에 저장… • 레이어에 추가… 이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Snapped** | `OUTPUT` | [same as input] | 산출 (스냅된) 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:snappointstogrid`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 최대 거리로 라인 분할하기

라인 (또는 곡선) 레이어를 받아 각 피처를 각 부분이 지정한 최대 길이인 다중 부분으로 분할합니다. 새 라인 부스트링의 시작점 및 종단점에서의 Z 및 M 값은 기존 값에서 선형적으로 보간됩니다.

라인 피처를제자리 피처 수정 할 수 있습니다.

**경고**

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: line] | 입력 라인 벡터 레이어 |
| **Maximum line length** | `LENGTH` | [numeric: double]<br>기본값: 10.0 | 산출 라인의 최장 길이 |
| **Split** | `OUTPUT` | [vector: line]<br>기본값: `[Create temporary layer]` | Specify the output line vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Split** | `OUTPUT` | [vector: line] | The new line vector layer. The length of the feature geometries is less than or equal to the length specified in the LENGTH parameter. An additional field containing the `order` will be added. |

### 파이썬 코드

**알고리즘 ID**: `native:splitlinesbylength`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 세분화

도형을 세분화 (subdivide) 합니다. 반환되는 도형은 원본 도형에서 세분화한 부분들을 담고 있는 집합일 것입니다. 이때 어떤 부분도 지정한 최대 노드 개수를 넘지 않습니다.

이 알고리즘은 공간 인덱스를 더 쉽게 작성하고 공간 처리 작업을 더 빨리 수행할 수 있도록 복잡 도형을 덜 복잡한 부분들로 나누는 데 유용합니다. 만곡 도형은 세분화 전에 선분화될 것입니다.

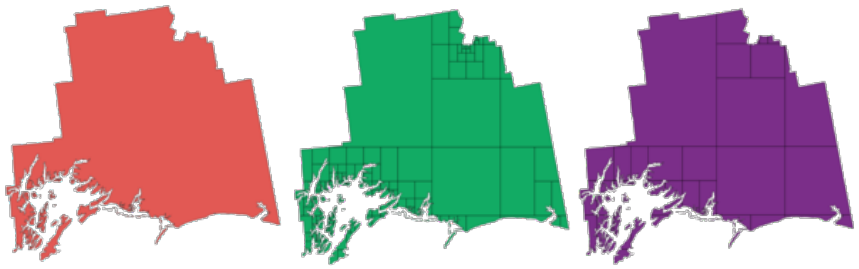


그림 24.142: 왼쪽: 입력 레이어, 중앙: 최대 노드 값 100, 오른쪽: 최대 노드 값 200

포인트, 라인 및 폴리곤 피처를제자리 피처 수정 할 수 있습니다.

**참고**

도형을 세분화하면 무결하지 않거나 자체 교차하는 도형 부분들을 생성할 수도 있습니다.

**더 보기**

라인 조각내기, 라인 부스트링 생성하기

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | 입력 벡터 레이어 |
| **Maximum nodes in parts** | `MAX_NODES` | [numeric: integer]<br>기본값: 256 | 각 새 도형 부분이 가질 수 있는 꼭짓점의 최대 개수입니다. 값이 높을수록 하위 부분의 개수가 적어집니다. |

다음 페이지에 계속

표 24.744 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Subdivided** | `OUTPUT` | [same as input]<br>기본값: `[Create temporary layer]` | Specify the output (subdivided) vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Subdivided** | `OUTPUT` | [same as input] | 산출 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:subdivide`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## X 좌표와 Y 좌표 바꾸기

입력 도형의 X 좌표와 Y 좌표를 바꿉니다.

이 알고리즘은 위도와 경도 값이 잘못 뒤바뀐 도형을 수리하는 데 유용합니다.

포인트, 라인 및 폴리곤 피처를제자리 피처 수정 할 수 있습니다.

**더 보기**

이동시키기 *(Translate)*, 기울이기

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | 입력 벡터 레이어 |
| **Swapped** | `OUTPUT` | [same as input]<br>기본값: `[Create temporary layer]` | Specify the output vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Swapped** | `OUTPUT` | [same as input] | 산출 (바뀐) 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:swapxy`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 줄어드는 버퍼 생성하기

라인 도형을 따라 시작점 및 종단점의 버퍼 지름을 지정해서 줄어드는 버퍼를 생성합니다.

**더 보기**

변동 너비 버퍼 *(M* 값으로*)*, 버퍼, 쐐기 버퍼 생성하기

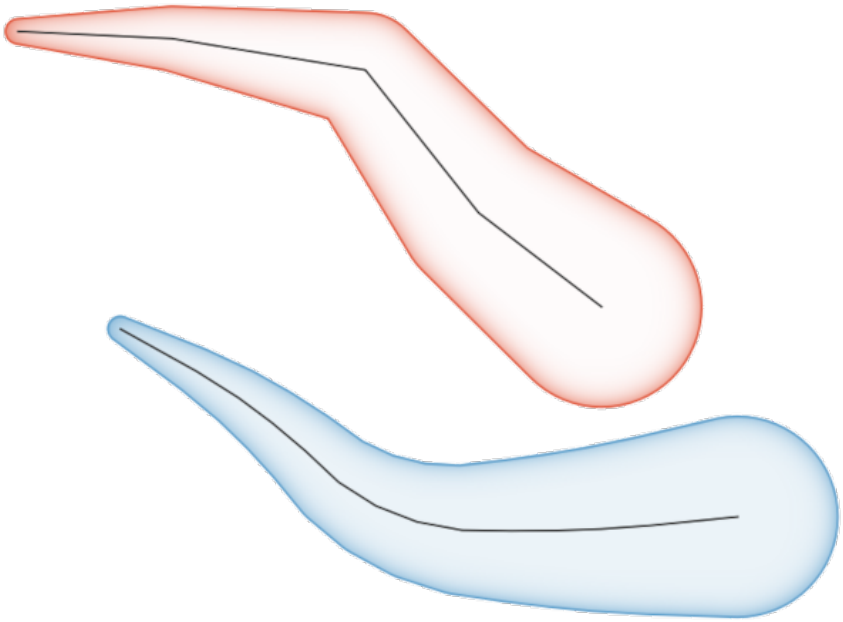


그림 24.143: 줄어드는 버퍼 예시

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: line] | 입력 라인 벡터 레이어 |
| **Start width** | `START_WIDTH` | [numeric: double]<br>기본값: 0.0 | 라인 피처의 시작 포인트에 적용되는 버퍼 반경을 설정합니다. |
| **End width** | `END_WIDTH` | [numeric: double]<br>기본값: 0.0 | 라인 피처의 종단 포인트에 적용되는 버퍼 반경을 설정합니다. |
| **Segments** | `SEGMENTS` | [numeric: integer]<br>기본값: 16 | 둥글린 (rounded) 오프셋을 생성하는 경우 사분원에 가깝게 만드는 데 사용할 라인 선분의 개수를 제어합니다. |
| **Buffered** | `OUTPUT` | [vector: polygon]<br>기본값: `[Create temporary layer]` | Specify the output (buffer) layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Buffered** | `OUTPUT` | [vector: polygon] | 산출 (버퍼) 폴리곤 레이어 |

### 모자이크화

폴리곤 도형 레이어에 있는 도형을 삼각형 부분들로 나누어 모자이크화 (tessellate) 합니다.

산출 레이어는 각 입력 피처에 대한 멀티폴리곤 도형들로 이루어집니다. 각 멀티폴리곤 도형은 삼각형 부분 폴리곤 여러 개로 이루어집니다.

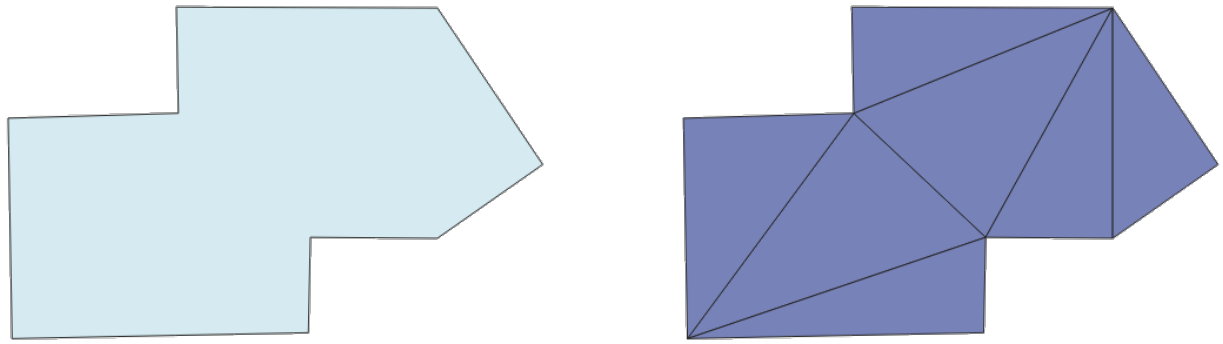


그림 24.144: 모자이크화된 폴리곤 (오른쪽)

☑ 폴리곤 피처를제자리 피처 수정 할 수 있습니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: polygon] | 입력 폴리곤 벡터 레이어 |
| **Tesselated** | `OUTPUT` | [vector: polygon]<br>기본값: `[Create temporary layer]` | Specify the output layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Tesselated** | `OUTPUT` | [vector: polygon] | 산출 멀티폴리곤 Z 레이어 |

### 파이썬 코드

**알고리즘 ID**: `3d:tessellate`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 절개선 생성하기

[멀티] 라인스트링의 포인트에 대한 절개선 (transect) 을 생성합니다.

절개선이란 입력 폴리라인 (의 꼭짓점) 에 대한 (기본값이 수직인) 지정 각도 방향의 라인을 말합니다.

절개선은 다음과 같은 새 필드와 함께 피처 (들) 의 필드 (들) 를 반환합니다:

- TR_FID: 원본 피처의 ID
- TR_ID: 절개선의 ID. 각 절개선은 유일 ID 를 가집니다.
- TR_SEGMENT: 라인스트링 선분의 ID
- TR_ANGLE: 원본 라인의 꼭짓점 위치에서의 도 단위 각도
- TR_LENGTH: 반환된 절개선의 총 길이
- TR_ORIENT: 절개선의 쪽. (라인의 왼쪽만, 오른쪽만, 또는 양쪽)

**경고**

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

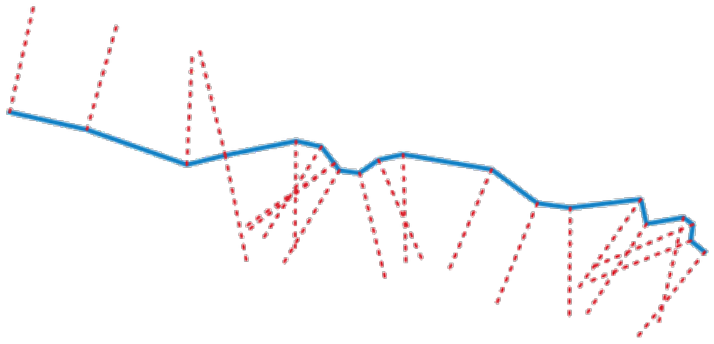


그림 24.145: 입력 라인 레이어의 절개선을 나타내는 빨간 점선

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: line] | 입력 라인 벡터 레이어 |
| **Length of the transect** | `LENGTH` | [numeric: double]<br>기본값: 5.0 | 절개선의 맵 단위 길이 |
| **Angle in degrees from the original line at the vertices** | `ANGLE` | [numeric: double]<br>기본값: 90.0 | 절개선의 각도를 변경합니다. |
| **Side to create the transect** | `SIDE` | [enumeration] | 절개선의 쪽을 설정합니다. 다음 옵션 가운데 선택할 수 있습니다:<br>• 0 — 왼쪽<br>• 1 — 오른쪽<br>• 2 — 양쪽 |
| **Transect** | `OUTPUT` | [vector: line]<br>기본값: `[Create temporary layer]` | Specify the output line layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Transect** | `OUTPUT` | [vector: line] | 산출 라인 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:transect`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 이동시키기 (Translate)

레이어 내에서 사전 정의한 X 및 Y 변위로 오프셋시켜 도형을 이동시킵니다.
도형에 있는 Z 및 M 값도 이동시킬 수 있습니다.

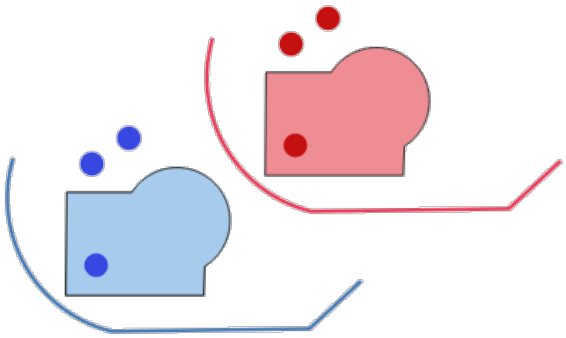


그림 24.146: Features in red represent the translated geometry of the input (blue) layer

포인트, 라인 및 폴리곤 피처를제자리 피처 수정 할 수 있습니다.

**더 보기**

이동시킨 피처의 배열, 라인 오프셋시키기, 기울이기, *X* 좌표와 *Y* 좌표 바꾸기

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | 입력 벡터 레이어 |
| **Offset distance (x-axis)** | `DELTA_X` | [numeric: double]<br>기본값: 0.0 | X 축에 적용할 변위 (displacement) |
| **Offset distance (y-axis)** | `DELTA_Y` | [numeric: double]<br>기본값: 0.0 | Y 축에 적용할 변위 (displacement) |
| **Offset distance (z-axis)** | `DELTA_Z` | [numeric: double]<br>기본값: 0.0 | Z 축에 적용할 변위 (displacement) |
| **Offset distance (m values)** | `DELTA_M` | [numeric: double]<br>기본값: 0.0 | M 축에 적용할 변위 (displacement) |

다음 페이지에 계속

표 24.754 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Translated** | `OUTPUT` | [same as input]<br>기본값: `[Create temporary layer]` | Specify the output vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Translated** | `OUTPUT` | [same as input] | 산출 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:translategeometry`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 변동 너비 버퍼 (M 값으로)

라인을 따라 변동 너비 버퍼를 생성합니다. 각 꼭짓점에서의 버퍼 직경은 라인 도형의 M 값을 따릅니다.

**더 보기**

줄어드는 버퍼 생성하기, 버퍼, *M* 값 설정하기, 가변 거리 버퍼

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: line] | 입력 라인 벡터 레이어 |

다음 페이지에 계속

표 24.756 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Segments** | `SEGMENTS` | [numeric: integer] 기본값: 16 | 사분원별 버퍼 선분의 개수. 유일값일 수도 있고 (모든 피처에 대해 동일한 개수) 또는 피처 데이터에서 추출한 값일 수도 있습니다 (피처 값에 따라 개수가 달라집니다). |
| **Buffered** | `OUTPUT` | [vector: polygon] 기본값: `[Create temporary layer]` | Specify the output (buffer) layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Buffered** | `OUTPUT` | [vector: polygon] | 변동 버퍼 폴리곤 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:bufferbym`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 보로노이 폴리곤

포인트 레이어를 받아, 입력 포인트에 대응하는 (티센 (Thiessen) 폴리곤이라고도 하는) 보로노이 (Voronoi) 폴리곤을 담고 있는 폴리곤 레이어를 생성합니다.

보로노이 폴리곤 내부의 모든 위치가 다른 어떤 포인트보다 관련 포인트에 더 가깝습니다.

**기본 메뉴**: *Vector → Geometry Tools*

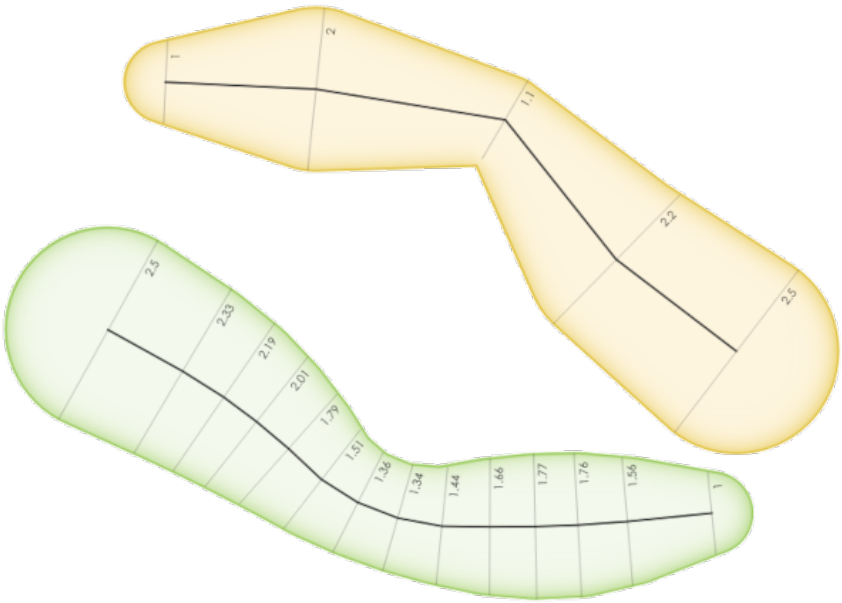



그림 24.147: 변동 버퍼 예시

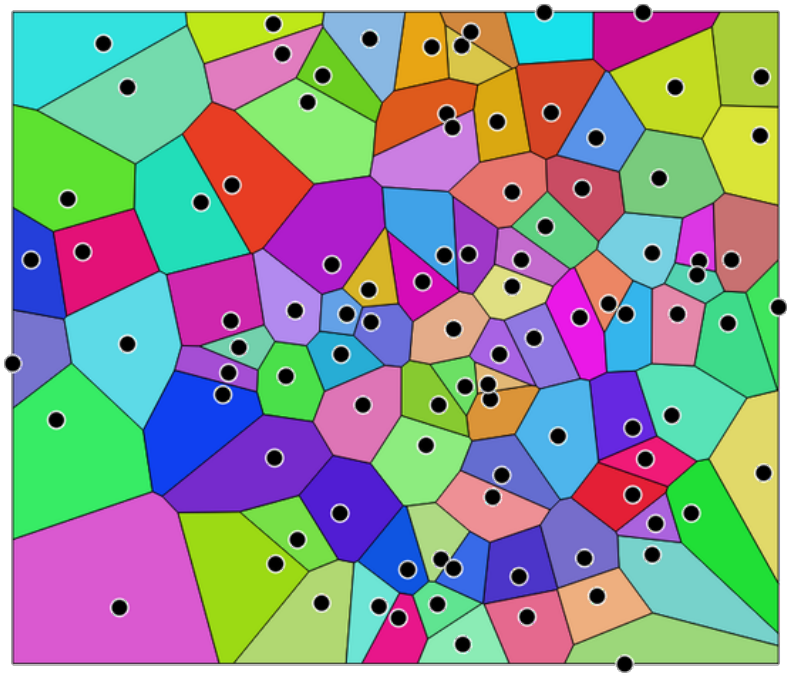


그림 24.148: 보로노이 폴리곤

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: point] | 입력 포인트 벡터 레이어 |
| **Buffer region (% of extent)** | `BUFFER` | [numeric: double]<br>기본값: 0.0 | 산출 레이어의 범위는 입력 레이어의 범위보다 이만큼 더 커질 것입니다. |
| **Tolerance**<br>선택적 | `TOLERANCE` | [numeric: double]<br>기본값: 0.0 | 보로노이 폴리곤 생성의 경고성을 향상시키는 데 사용할 수 있는 선택적인 스냅 허용 오차를 지정합니다. |
| **Copy attributes from input features** | `COPY_ATTRIBUTES` | [boolean]<br>기본값: True | 산출물에 관련 포인트 피처 ID 를 저장하고 있는 필드들을 추가해야 할지 여부를 지정합니다. 거짓으로 설정하는 경우 `id` 필드를 생성해서 폴리곤을 식별합니다. |
| **Voronoi polygons** | `OUTPUT` | [vector: polygon]<br>기본값: `[Create temporary layer]` | Specify the output layer (with the Voronoi polygons). *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Voronoi polygons** | `OUTPUT` | [vector: polygon] | 입력 포인트 벡터 레이어에 대한 보로노이 폴리곤 |

### 파이썬 코드

**알고리즘 ID**: `native:voronoipolygons`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 24.1.27 벡터 중첩

#### 잘라내기 (Clip)

추가 폴리곤 레이어의 피처를 사용해서 벡터 레이어를 잘라냅니다.

산출 레이어는 입력 레이어의 피처 가운데 중첩 레이어의 폴리곤 내부에 들어오는 부분만 담을 것입니다.

**경고**

**피처 수정 전용**

이 작업은 피처 도형만 수정합니다. 중첩 작업으로 피처의 면적 또는 길이와 같은 속성 (property) 이 변경되긴 하지만, 피처의 속성 (attribute) 값은 **수정되지 않습니다**. 이런 속성 (property) 을 속성 (attribute) 으로 저장하고 있다면, 속성 (attribute) 을 직접 업데이트해야 할 것입니다.

마스크 도형이 도형 전체를 담고 있지 않은 경우, 이 알고리즘은 제공자 편에서 공간 인덱스를 사용해서 도형을 준비하고 잘라내기 작업을 수행합니다.

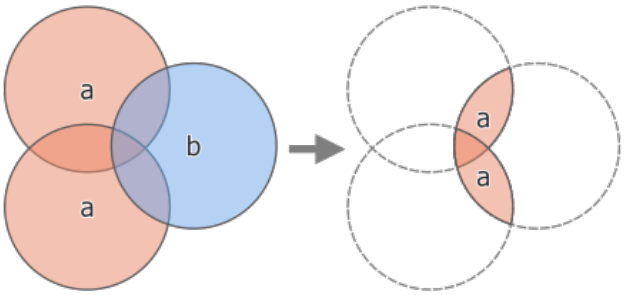



그림 24.149: 잘라내기 작업: 피처 2 개를 가진 입력 레이어'a'와 피처 1 개를 가진 중첩 레이어'b'(왼쪽) - 수정된'a' 피처를 가진 새로운 산출 레이어 (오른쪽)

포인트, 라인 및 폴리곤 피처를제자리 피처 수정 할 수 있습니다.

**기본 메뉴**: *Vector ▸ Geoprocessing Tools*

**더 보기**

교차 *(Intersection)*, 차감하기 *(Difference)*

#### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | 잘라낼 피처를 담고 있는 레이어 |
| **Overlay layer** | `OVERLAY` | [vector: polygon] | 잘라내기 피처를 담고 있는 레이어 |

다음 페이지에 계속

표 24.760 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Clipped** | `OUTPUT` | [same as input]<br>기본값: `[Create temporary layer]` | Specify the layer to contain the features from the input layer that are inside the overlay (clipping) layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Clipped** | `OUTPUT` | [same as input] | 중첩 레이어가 잘라낸 입력 레이어의 피처를 담고 있는 레이어 |

### 파이썬 코드

**Algorithm ID**: `native:clip`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 차감하기 (Difference)

입력 레이어에서 중첩 레이어의 경계선 내부에 들어오지 않는 피처를 추출합니다.

중첩 레이어의 피처 (들) 와 일부분 중첩하는 입력 레이어의 피처를 중첩 레이어 피처 (들) 의 경계선을 따라 분할한 다음, 중첩 레이어 피처 (들) 외부에 있는 부분을 남깁니다.

**경고**

**피처 수정 전용**

이 작업은 피처 도형만 수정합니다. 중첩 작업으로 피처의 면적 또는 길이와 같은 속성 (property) 이 변경되긴 하지만, 피처의 속성 (attribute) 값은 **수정되지 않습니다**. 이런 속성 (property) 을 속성 (attribute) 으로 저장하고 있다면, 속성 (attribute) 을 직접 업데이트해야 할 것입니다.

포인트, 라인 및 폴리곤 피처를제자리 피처 수정 할 수 있습니다.

**기본 메뉴**: *Vector → Geoprocessing Tools*

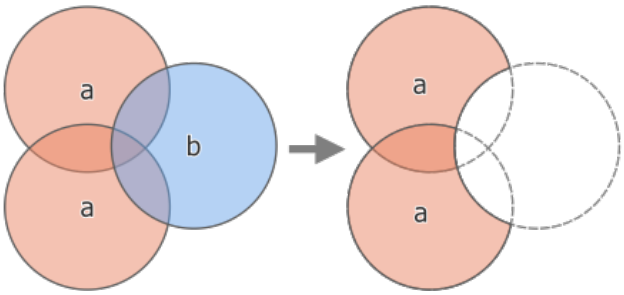



그림 24.150: 차감 작업: 피처 2 개를 가진 입력 레이어'a'와 피처 1 개를 가진 중첩 레이어'b'(왼쪽) - 수정된'a'피처를 가진 새로운 산출 레이어 (오른쪽)

**더 보기**

차감하기 *(다중)*, 대칭 차감, 잘라내기 *(Clip)*

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | 피처 (의 일부분) 를 추출할 레이어 |
| **Overlay layer** | `OVERLAY` | [vector: geometry] | 입력 레이어 도형에서 차감할 도형을 담고 있는 레이어입니다. 이 중첩 레이어는 최소한 입력 레이어 도형만큼의 차원 (포인트: 0 차원, 라인: 1 차원, 폴리곤: 2 차원, 부피: 3 차원) 을 가질 것을 요구합니다. |
| **Difference** | `OUTPUT` | [same as input]<br>기본값: `[Create temporary layer]` | Specify the layer to contain the (parts of) features from the input layer that are not inside the overlay layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Grid size**<br>선택적 | `GRID_SIZE` | [numeric: double]<br>기본값: 설정하지 않음 | If provided, the input geometries are snapped to a grid of the given size, and the result vertices are computed on that same grid. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|

### 파이썬 코드

**Algorithm ID**: `native:difference`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 차감하기 (다중)

입력 레이어의 피처 가운데 모든 중첩 레이어 (들) 의 피처와 전혀 중첩하거나 접하지 않는, 또는 일부분만 중첩하는 피처를 추출합니다.

각 중첩 레이어에 대해 이전 차감 작업의 결과 전부와 해당 중첩 레이어 간의 차를 계산합니다. 중첩 레이어의 피처 (들) 와 일부분 중첩하는 입력 레이어의 피처를 중첩 레이어의 피처 경계선을 따라 분할하고, 중첩 레이어의 피처 바깥 부분만 유지합니다.

**경고**

**피처 수정 전용**

이 작업은 피처 도형만 수정합니다. 중첩 작업으로 피처의 면적 또는 길이와 같은 속성 (property) 이 변경되긴 하지만, 피처의 속성 (attribute) 값은 **수정되지 않습니다**. 이런 속성 (property) 을 속성 (attribute) 으로 저장하고 있다면, 속성 (attribute) 을 직접 업데이트해야 할 것입니다.

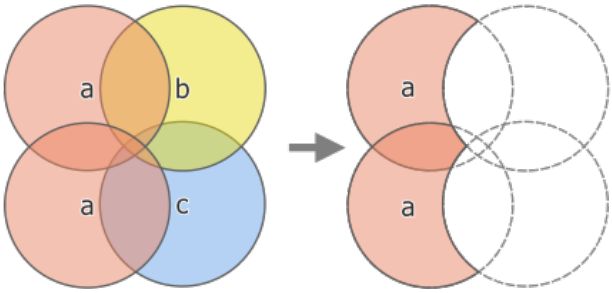



그림 24.151: 차감 작업: 피처 2 개를 가진 입력 레이어‘a’와 피처 1 개를 가진 중첩 레이어‘b’및‘c’(왼쪽) - 수정된‘a’ 피처를 가진 새로운 산출 레이어 (오른쪽)

**더 보기**

차감하기 *(Difference)*, 대칭 차감, 잘라내기 *(Clip)*

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | 피처 (의 일부분) 를 추출할 레이어 |
| **Overlay layers** | `OVERLAYS` | [vector: geometry] [list] | 입력 레이어 도형에서 차감할 도형을 담고 있는 레이어 목록입니다. 이 중첩 레이어들은 최소한 입력 레이어 도형만큼의 차원 (포인트: 0 차원, 라인: 1 차원, 폴리곤: 2 차원, 부피: 3 차원) 을 가질 것을 요구합니다. |
| **Difference** | `OUTPUT` | [same as input] 기본값: `[Create temporary layer]` | Specify the layer to contain the (parts of) features from the input layer that do not overlap features of the overlay layers. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Difference** | `OUTPUT` | [same as input] | 중첩 레이어의 피처와 중첩하지 않는 입력 레이어의 피처 (의 일부분) 를 담은 레이어 |

### 파이썬 코드

**Algorithm ID**: `native:multidifference`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 범위로 추출/잘라내기

Creates a new vector layer that only contains features which fall within a specified extent. Any features which intersect the extent will be included by default.

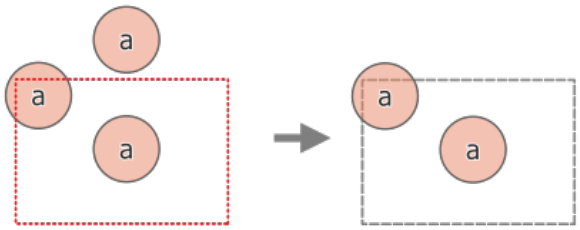



그림 24.152: 피처 3 개를 가진 입력 레이어'a'와 점선으로 표시된 범위 사이의 추출 작업 (왼쪽) - 참조를 위해 범위를 점선으로 표시한 산출 피처들 (오른쪽)

Optionally, feature geometries can also be clipped to the extent.

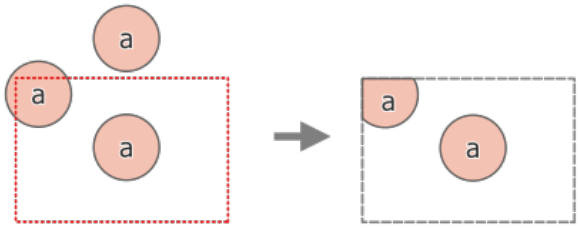



그림 24.153: 피처 3 개를 가진 입력 레이어'a'와 점선으로 표시된 범위 사이의 추출 작업 (왼쪽) - 참조를 위해 범위를 점선으로 표시한 산출 피처들 (오른쪽)

**더 보기**

잘라내기 *(Clip)*

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | 피처 (의 일부분) 를 추출할 레이어 |

다음 페이지에 계속

표 24.767 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Extent (xmin, xmax, ymin, ymax)** | `EXTENT` | [extent] | 잘라내기용 범위<br>다음과 같은 메소드를 사용할 수 있습니다:<br>• 레이어로부터 계산…: 현재 프로젝트에 불러온 레이어의 범위를 사용합니다.<br>• 레이어 맵으로부터 계산…: 활성 프로젝트의조판 맵 항목 범위를 사용합니다.<br>• 북마크로부터 계산…: 저장되어 있는북마크 의 범위를 사용합니다.<br>• 맵 캔버스 범위 사용<br>• 캔버스에 그리기: 연산에 넣을 범위를 정하는 직사각형을 클릭 & 드래그합니다.<br>• 좌표를 `xmin, xmax, ymin, ymax` 로 입력 |
| **Clip features to extent** | `CLIP` | [boolean]<br>기본값: False | If checked, the geometries will be clipped to the extent chosen instead of taking the whole geometry as output. Moreover, output geometries will be automatically converted to multi geometries to ensure uniform output types. |
| **Extracted** | `OUTPUT` | [same as input]<br>기본값: `[Create temporary layer]` | Specify the layer to contain the features from the input layer that are inside the clip extent. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Extracted** | `OUTPUT` | [same as input] | 잘라낸 피처를 담고 있는 레이어 |

### 파이썬 코드

**Algorithm ID**: `native:extractbyextent`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 교차 (Intersection)

입력 레이어의 피처에서 중첩 레이어에 있는 피처와 중첩하는 부분을 추출합니다.

(산출) 교차 레이어에 있는 피처는 입력 및 중첩 레이어에 있는 중첩하는 피처들의 속성을 모두 가질 것입니다.

**경고**

**피처 수정 전용**

이 작업은 피처 도형만 수정합니다. 중첩 작업으로 피처의 면적 또는 길이와 같은 속성 (property) 이 변경되긴 하지만, 피처의 속성 (attribute) 값은 **수정되지 않습니다**. 이런 속성 (property) 을 속성 (attribute) 으로 저장하고 있다면, 속성 (attribute) 을 직접 업데이트해야 할 것입니다.

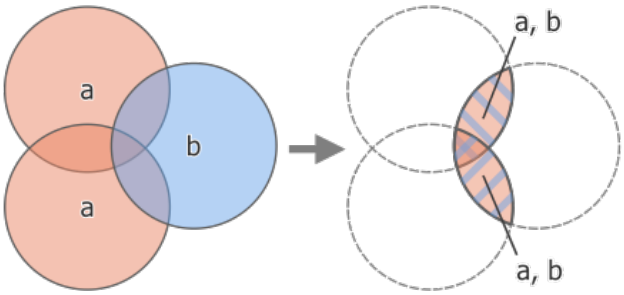



그림 24.154: 교차 작업: 피처 2 개를 가진 입력 레이어'a'와 피처 1 개를 가진 중첩 레이어'b'(왼쪽) - 중첩하는 영역이 피처 2 개와 두 레이어의 속성을 가진 새로운 레이어가 됩니다 (오른쪽)

**기본 메뉴**: *Vector → Geoprocessing Tools*

**경고**

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

**더 보기**

교차 *(다중)*, 잘라내기 *(Clip)*, 차감하기 *(Difference)*

### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | INPUT | [vector: geometry] | 피처 (의 일부분) 를 추출할 레이어 |
| **Overlay layer** | OVERLAY | [vector: geometry] | 중첩을 확인할 피처를 담고 있는 레이어입니다. 이 중첩 레이어의 도형은 최소한 입력 레이어 도형만큼의 차원 (포인트: 0 차원, 라인: 1 차원, 폴리곤: 2 차원, 부피: 3 차원) 을 가질 것을 요구합니다. |
| **Input fields to keep (leave empty to keep all fields)** 선택적 | INPUT_FIELDS | [tablefield: any] [list] 기본값: 설정하지 않음 | 산출물에 남길 입력 레이어의 필드 (들). 아무 필드도 선택하지 않을 경우 모든 필드를 복사합니다. |
| **Overlay fields to keep (leave empty to keep all fields)** 선택적 | OVERLAY_FIELDS | [tablefield: any] [list] 기본값: 설정하지 않음 | 산출물에 남길 중첩 레이어의 필드 (들). 아무 필드도 선택하지 않을 경우 모든 필드를 복사합니다. 충돌을 피하기 위해 중복되는 필드명 뒤에는 번호를 붙일 것입니다. |
| **Intersection** | OUTPUT | [same as input] 기본값: [Create temporary layer] | Specify the layer to contain (the parts of) the features from the input layer that overlap one or more features from the overlay layer. *One of*:<br>• 임시 레이어 생성 (TEMPORARY_OUTPUT)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

#### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Overlay fields prefix** 선택적 | OVERLAY_FIELDS_PREF | [string] | 중첩 레이어의 필드를 식별하기 위한 접두어를 추가합니다. 충돌을 피하기 위해 중복되는 필드명 뒤에는 번호를 붙일 것입니다. |
| **Grid size** 선택적 | GRID_SIZE | [numeric: double] 기본값: 설정하지 않음 | If provided, the input geometries are snapped to a grid of the given size, and the result vertices are computed on that same grid. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Intersection** | `OUTPUT` | [same as input] | 중첩 레이어와 중첩하는 입력 레이어의 피처 (의 일부분) 를 담은 레이어 |

### 파이썬 코드

**Algorithm ID**: `native:intersection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 교차 (다중)

입력 레이어와 모든 중첩 레이어들에 있는 피처들의 중첩하는 부분들을 추출합니다.

산출 레이어에 있는 피처는 입력 및 중첩 레이어에 있는 중첩하는 피처들의 속성을 모두 가질 것입니다.

> **경고**
>
> **피처 수정 전용**
>
> 이 작업은 피처 도형만 수정합니다. 중첩 작업으로 피처의 면적 또는 길이와 같은 속성 (property) 이 변경되긴 하지만, 피처의 속성 (attribute) 값은 **수정되지 않습니다**. 이런 속성 (property) 을 속성 (attribute) 으로 저장하고 있다면, 속성 (attribute) 을 직접 업데이트해야 할 것입니다.

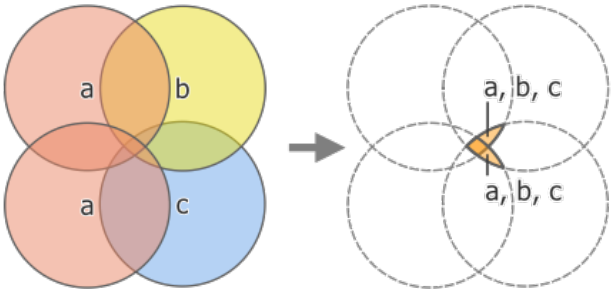



그림 24.155: 차감 작업: 피처 2 개를 가진 입력 레이어'a'와 피처 1 개를 가진 중첩 레이어'b'및'c'(왼쪽) - 중첩하는 영역이 피처 2 개와 모든 레이어의 속성을 가진 새로운 레이어가 됩니다 (오른쪽)

> **경고**
>
> This algorithm drops existing primary keys or FID values and regenerate them in output layers.

더 보기

교차 *(Intersection)*, 잘라내기 *(Clip)*, 차감하기 *(Difference)*

### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | INPUT | [vector: geometry] | 피처 (의 일부분) 를 추출할 레이어 |
| **Overlay layers** | OVERLAYS | [vector: geometry] [list] | 중첩을 확인할 피처를 담고 있는 레이어들입니다. 이 중첩 레이어들의 도형은 최소한 입력 레이어 도형만큼의 차원 (포인트: 0 차원, 라인: 1 차원, 폴리곤: 2 차원, 부피: 3 차원) 을 가질 것을 요구합니다. |

#### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Overlay fields prefix**<br>선택적 | OVERLAY_FIELDS_PREF | [string] | 중첩 레이어들의 필드를 식별하기 위한 접두어를 추가합니다. 충돌을 피하기 위해 중복되는 필드명 뒤에는 번호를 붙일 것입니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Intersection** | OUTPUT | [same as input] | 중첩 레이어들과 중첩하는 입력 레이어의 피처 (의 일부분) 를 담은 레이어 |

### 파이썬 코드

**Algorithm ID**: `native:multiintersection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 라인 교차

두 레이어의 라인들이 교차하는 위치의 포인트 피처를 생성합니다.

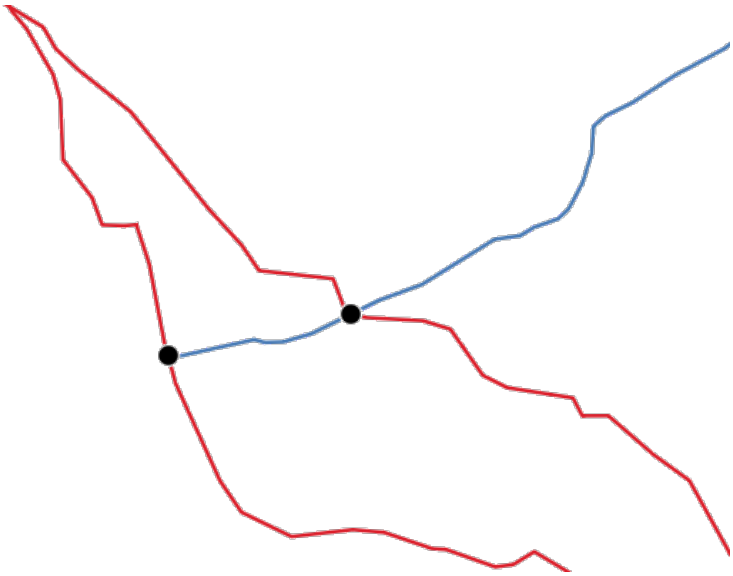


그림 24.156: 교차점 포인트

**기본 메뉴**: *Vector → Analysis Tools*

> **경고**
>
> This algorithm drops existing primary keys or FID values and regenerate them in output layers.

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: line] | 입력 라인 벡터 레이어 |
| **Intersect layer** | `INTERSECT` | [vector: line] | 라인 교차점을 찾는 데 사용할 레이어 |
| **Input fields to keep (leave empty to keep all fields)** 선택적 | `INPUT_FIELDS` | [tablefield: any] [list] 기본값: 설정하지 않음 | 산출물에 남길 입력 레이어의 필드 (들). 아무 필드도 선택하지 않을 경우 모든 필드를 복사합니다. |
| **Intersect fields to keep (leave empty to keep all fields)** 선택적 | `INTERSECT_FIELDS` | [tablefield: any] [list] 기본값: 설정하지 않음 | 산출물에 남길 교차 레이어의 필드 (들). 아무 필드도 선택하지 않을 경우 모든 필드를 복사합니다. 충돌을 피하기 위해 중복되는 필드명 뒤에는 번호를 붙일 것입니다. |

다음 페이지에 계속

표 24.775 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Intersection** | `OUTPUT` | [vector: point]<br>기본값: `[Create temporary layer]` | Specify the layer to contain the intersection points of the lines from the input and overlay layers. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Intersect fields prefix**<br>선택적 | `INTERSECT_FIELDS_PRE` | [string] | 교차 레이어의 필드를 식별하기 위한 접두어를 추가합니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Intersections** | `OUTPUT` | [vector: point] | 두 레이어의 속성을 모두 가진, 라인 교차점의 포인트 벡터 레이어 |

### 파이썬 코드

**Algorithm ID**: `native:lineintersections`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 라인으로 분할하기

레이어에 있는 라인 또는 폴리곤을 분절점 (breaking point) 을 정의하기 위한 또다른 레이어의 라인 또는 폴리곤 고리를 이용해서 분할합니다. 두 레이어에 있는 도형들 사이의 교차점을 분할점으로 삼습니다.

산출물은 분할된 피처를 위한 다중 도형을 담을 것입니다.

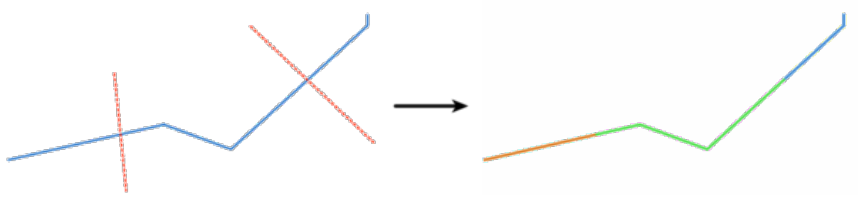


그림 24.157: 라인 분할하기

☑ 라인 및 폴리곤 피처를제자리 피처 수정 할 수 있습니다.

**⚠ 경고**

This algorithm drops existing primary keys or FID values and regenerate them in output layers.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: line, polygon] | 분할할 라인 또는 폴리곤을 담고 있는 레이어 |
| **Split layer** | `LINES` | [vector: line, polygon] | 분절점을 정의하는 데 사용되는 라인 또는 고리를 가진 레이어 |
| **Split** | `OUTPUT` | [same as input]<br>기본값: `[Create temporary layer]` | Specify the layer to contain the splitted (in case they are intersected by a line in the split layer) line/polygon features from the input layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Split** | `OUTPUT` | [same as input] | 입력 레이어에서 분할된 라인 또는 폴리곤을 가진 산출 벡터 레이어 |

### 파이썬 코드

**Algorithm ID**: `native:splitwithlines`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 대칭 차감

입력 및 중첩 레이어 양쪽의 피처들 가운데 두 레이어가 중첩하는 영역을 제거한 피처를 담고 있는 레이어를 생성합니다.

대칭 차감 레이어의 속성 테이블은 입력 및 중첩 레이어 양쪽의 속성과 필드를 모두 담고 있습니다.

> **경고**
>
> **피처 수정 전용**
>
> 이 작업은 피처 도형만 수정합니다. 중첩 작업으로 피처의 면적 또는 길이와 같은 속성 (property) 이 변경되긴 하지만, 피처의 속성 (attribute) 값은 **수정되지 않습니다**. 이런 속성 (property) 을 속성 (attribute) 으로 저장하고 있다면, 속성 (attribute) 을 직접 업데이트해야 할 것입니다.

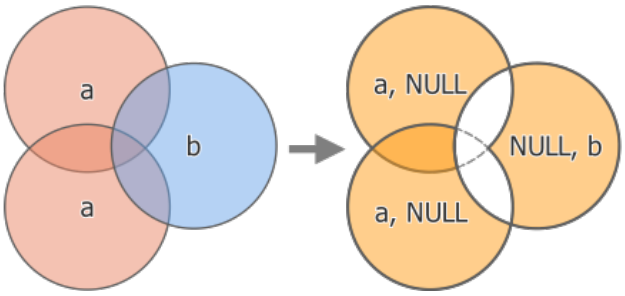



그림 24.158: 대칭 차감 작업: 피처 2 개를 가진 입력 레이어'a'와 피처 1 개를 가진 중첩 레이어'b'(왼쪽) - 피처 3 개와 두 레이어의 속성을 모두 가진 산출 레이어 (오른쪽)

**기본 메뉴**: *Vector ▯ Geoprocessing Tools*

**경고**

This algorithm drops existing primary keys or FID values and regenerate them in output layers.

**더 보기**

차감하기 *(Difference)*, 잘라내기 *(Clip)*, 교차 *(Intersection)*

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | 피처 (의 일부분) 를 추출할 첫 번째 레이어 |
| **Overlay layer** | `OVERLAY` | [vector: geometry] | 피처 (의 일부분) 를 추출할 두 번째 레이어. 도형 유형이 입력 레이어와 동일한 편이 좋습니다. |
| **Symmetrical difference** | `OUTPUT` | [same as input]<br>기본값: `[Create temporary layer]` | Specify the layer to contain (the parts of) the features from the input and overlay layers that do not overlap features from the other layer.<br>*One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Overlay fields prefix**<br>선택적 | `OVERLAY_FIELDS_PREF` | [string] | 중첩 레이어의 필드를 식별하기 위한 접두어를 추가합니다. 충돌을 피하기 위해 중복되는 필드명 뒤에는 번호를 붙일 것입니다. |
| **Grid size**<br>선택적 | `GRID_SIZE` | [numeric: double]<br>기본값: 설정하지 않음 | If provided, the input geometries are snapped to a grid of the given size, and the result vertices are computed on that same grid. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Symmetrical difference** | `OUTPUT` | [same as input] | 각 레이어에서 다른 레이어와 중첩하지 않은 피처 (의 일부분) 와 두 레이어의 속성을 모두 담은 레이어 |

### 파이썬 코드

**Algorithm ID**: `native:symmetricaldifference`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 통합 (Union)

입력 레이어 내부에서 피처들 사이의 중첩을 검사하고, 중첩하는 그리고 중첩하지 않는 부분들을 개별 피처로 생성합니다. 중첩하는 영역은 동일한 중첩하는 피처를 해당 중첩 영역에 걸친 피처들의 개수만큼 생성할 것입니다.

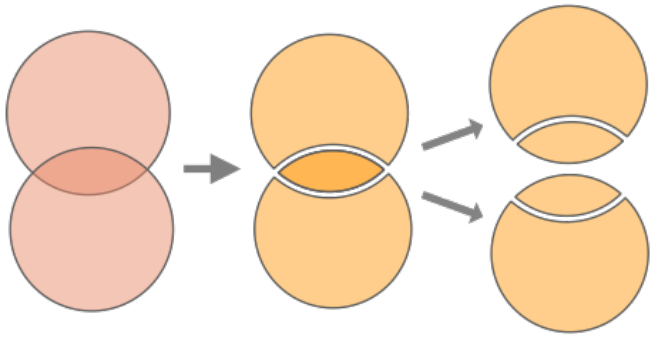


그림 24.159: 통합 작업: 중첩하는 피처 2 개를 가진 단일 입력 레이어 (왼쪽) - 피처 4 개를 가진 산출 레이어 (중앙) - 명확하게 보여주기 위해 이동시킨 산출 피처들 (오른쪽)

각 레이어의 피처를 다른 레이어의 피처와 중첩하는 부분에서 분할한 다음 입력 및 중첩 레이어의 모든 부분들을 담은 레이어를 생성하는 경우, 중첩 레이어도 사용할 수 있습니다. 동일한 레이어 상에 있는 피처는 서로를 분할시키지 않을 것입니다. 통합 레이어의 속성 테이블은 중첩하지 않는 피처의 경우 각 원본 레이어의 속성값으로, 중첩하는 피처의 경우 두 레이어의 속성값을 담게 됩니다.

> **참고**
>
> 중첩 레이어를 사용하면, 동일한 레이어 상에 있는 피처들은 서로를 분할시키지 않을 것입니다. 다른 레이어와 마찬가지로 동일한 레이어 상에서 중첩하는 피처들을 분할시키려면, 먼저 여러 레이어와 함께 알고리즘을 실행한 다음 해당 산출물에 대해서만 알고리즘을 다시 실행하십시오.

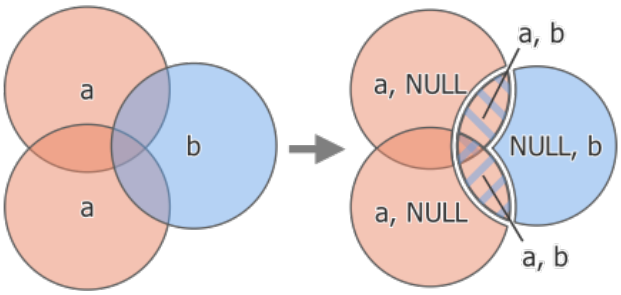



그림 24.160: 통합 작업: 피처 2 개를 가진 입력 레이어'a'와 피처 1 개를 가진 중첩 레이어'b'(왼쪽) - 피처 5 개와 두 레이어의 속성을 모두 가진 산출 레이어 (오른쪽)

**기본 메뉴**: *Vector → Geoprocessing Tools*

**경고**

This algorithm drops existing primary keys or FID values and regenerate them in output layers.

**더 보기**

통합 *(다중)*, 잘라내기 *(Clip)*, 차감하기 *(Difference)*, 교차 *(Intersection)*

### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | 모든 교차점에서 분할할 입력 벡터 레이어 |
| **Overlay layer**<br>선택적 | `OVERLAY` | [vector: geometry] | 첫 번째 레이어로 결합될 레이어. 도형 유형이 입력 레이어와 동일한 편이 좋습니다. |
| **Union** | `OUTPUT` | [same as input]<br>기본값: `[Create temporary layer]` | Specify the layer to contain the (split and duplicated) features from the input layer and the overlay layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

#### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Overlay fields prefix**<br>선택적 | `OVERLAY_FIELDS_PREFIX` | [string] | 중첩 레이어의 필드를 식별하기 위한 접두어를 추가합니다. 충돌을 피하기 위해 중복되는 필드명 뒤에는 번호를 붙일 것입니다. |
| **Grid size** | `GRID_SIZE` | [numeric: double] | If provided, the input geometries are snapped |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Union** | `OUTPUT` | [same as input] | 공간 처리된 레이어 (들) 에서 나온 중첩하는 그리고 중첩하지 않는 부분들을 모두 담고 있는 레이어 |

### 파이썬 코드

**Algorithm ID**: `native:union`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 통합 (다중)

입력 레이어 내부에서 피처들 사이의 중첩을 검사하고, 중첩하는 그리고 중첩하지 않는 부분들을 개별 피처로 생성합니다. 중첩하는 영역은 동일한 중첩하는 피처를 해당 중첩 영역에 걸친 피처들의 개수만큼 생성할 것입니다.

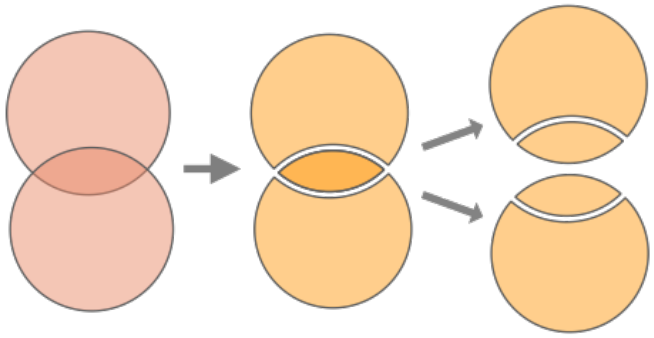


그림 24.161: 통합 작업: 중첩하는 피처 2 개를 가진 단일 입력 레이어 (왼쪽) - 피처 4 개를 가진 산출 레이어 (중앙) - 명확하게 보여주기 위해 이동시킨 산출 피처들 (오른쪽)

각 레이어의 피처를 다른 모든 레이어의 피처와 중첩하는 부분에서 분할한 다음 입력 및 중첩 레이어의 모든 부분들을 담은 레이어를 생성하는 경우, 여러 개의 중첩 레이어도 사용할 수 있습니다. 동일한 레이어 상에 있는 피처는 서로를 분할시키지 않을 것입니다. 통합 레이어의 속성 테이블은 중첩하지 않는 피처의 경우 각 원본 레이어의 속성값으로, 중첩하는 피처의 경우 중첩 레이어들의 속성값을 담게 됩니다.

> **참고**
>
> 중첩 레이어를 사용하면, 동일한 레이어 상에 있는 피처들은 서로를 분할시키지 않을 것입니다. 다른 레이어와 마찬가지로 동일한 레이어 상에서 중첩하는 피처들을 분할시키려면, 먼저 여러 레이어와 함께 알고리즘을 실행한 다음 해당 산출물에 대해서만 알고리즘을 다시 실행하십시오.

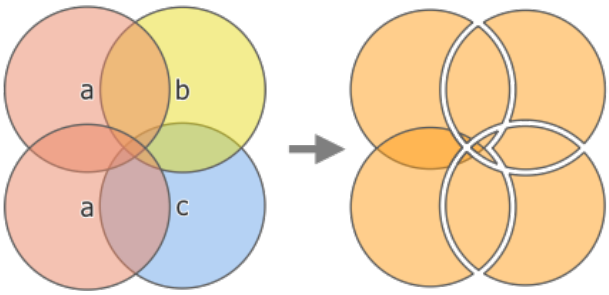



그림 24.162: 통합 작업: 피처 2 개를 가진 입력 레이어'a'와 피처 1 개를 가진 중첩 레이어'b'및'c'(왼쪽) - 피처 11 개와 모든 레이어의 속성을 가진 산출 레이어 (오른쪽)

**경고**

This algorithm drops existing primary keys or FID values and regenerate them in output layers.

**더 보기**

통합 *(Union)*, 잘라내기 *(Clip)*, 차감하기 *(Difference)*, 교차 *(Intersection)*

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | 모든 교차점에서 분할할 입력 벡터 레이어 |
| **Overlay layers**<br>선택적 | `OVERLAYS` | [vector: geometry]<br>[list] | 첫 번째 레이어로 결합될 레이어들. 도형 유형이 입력 레이어와 동일한 편이 좋습니다. |
| **Union** | `OUTPUT` | [same as input]<br>기본값: `[Create temporary layer]` | Specify the layer to contain the (split and duplicated) features from the input layer and the overlay layers. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Overlay fields prefix**<br>선택적 | `OVERLAY_FIELDS_PREF` | [string] | 중첩 레이어들의 필드를 식별하기 위한 접두어를 추가합니다. 충돌을 피하기 위해 중복되는 필드명 뒤에는 번호를 붙일 것입니다. |

#### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Union** | `OUTPUT` | [same as input] | 공간 처리된 레이어 (들) 에서 나온 중첩하는 그리고 중첩하지 않는 부분들과 모든 레이어의 속성을 담고 있는 레이어 |

#### 파이썬 코드

**Algorithm ID**: `native:multiunion`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 24.1.28 벡터 선택

#### 속성으로 추출하기

입력 레이어에서 벡터 레이어 2 개를 생성합니다. 첫 번째 레이어는 일치하는 피처만 담고, 두 번째 레이어는 일치하지 않는 피처를 모두 담을 것입니다.

산출 레이어에 피처를 추가하는 기준은 입력 레이어의 속성값을 바탕으로 합니다.

**더 보기**

속성으로 선택하기

#### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: any] | 피처를 추출할 레이어 |
| **Selection attribute** | `FIELD` | [tablefield: any] | 레이어의 필터링 필드 |

다음 페이지에 계속

표 24.789 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Operator** | OPERATOR | [enumeration] 기본값: 0 | 다음과 같은 많은 연산자를 사용할 수 있습니다:<br>• 0 —=<br>• 1 —≠<br>• 2 —><br>• 3 —>=<br>• 4 —<<br>• 5 —<=<br>• 6 —begins with (…으로 시작)<br>• 7 —contains (포함)<br>• 8 —is null (NULL 값임)<br>• 9 —is not null (NULL 값 아님)<br>• 10 —does not contain (포함하지 않음) |
| **Value** 선택적 | VALUE | [string] | 평가할 값 |
| **Extracted (attribute)** | OUTPUT | [same as input] 기본값: [Create Temporary Layer] | Specify the output vector layer for matching features. *One of*:<br>• 임시 레이어 생성 (TEMPORARY_OUTPUT)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |
| **Extracted (non-matching)** | FAIL_OUTPUT | [same as input] 기본값: [Skip output] | Specify the output vector layer for non-matching features. *One of*:<br>• 산출물 건너뛰기<br>• 임시 레이어 생성 (TEMPORARY_OUTPUT)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Extracted (attribute)** | OUTPUT | [same as input] | 입력 레이어에서 일치하는 피처를 담은 벡터 레이어 |
| **Extracted (non-matching)** | FAIL_OUTPUT | [same as input] | 입력 레이어에서 일치하지 않는 피처를 담은 벡터 레이어 |

### 파이썬 코드

**Algorithm ID**: `native:extractbyattribute`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 표현식으로 추출하기

입력 레이어에서 벡터 레이어 2 개를 생성합니다. 첫 번째 레이어는 일치하는 피처만 담고, 두 번째 레이어는 일치하지 않는 피처를 모두 담을 것입니다.

산출 레이어에 피처를 추가하는 기준은 QGIS 표현식을 바탕으로 합니다. 표현식에 대해 더 자세히 알고 싶다면표현식 을 참조하세요.

**더 보기**

표현식으로 선택하기

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: any] | 입력 벡터 레이어 |
| **Expression** | `EXPRESSION` | [expression] | 벡터 레이어를 필터링할 표현식 |
| **Matching features** | `OUTPUT` | [same as input]<br>기본값: `[Create Temporary Layer]` | Specify the output vector layer for matching features. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |
| **Non-matching** | `FAIL_OUTPUT` | [same as input]<br>기본값: `[Skip output]` | Specify the output vector layer for non-matching features. *One of*:<br>• 산출물 건너뛰기<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Matching features** | `OUTPUT` | [same as input] | 입력 레이어에서 일치하는 피처를 담은 벡터 레이어 |
| **Non-matching** | `FAIL_OUTPUT` | [same as input] | 입력 레이어에서 일치하지 않는 피처를 담은 벡터 레이어 |

### 파이썬 코드

**Algorithm ID**: `native:extractbyexpression`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 위치로 추출하기

입력 레이어에서 일치하는 피처만 담고 있는 새 벡터 레이어를 생성합니다.

산출 레이어에 피처를 추가하는 기준은 입력 레이어의 각 피처와 추가 레이어의 피처들 사이의 공간 관계를 바탕으로 합니다.

**더 보기**

위치로 선택하기, 거리 안에서 추출하기

## 공간 관계성 탐색하기

기하 서술 (geometric predicates) 은 피처들의 도형들이 공간 일부분을 공유하는지, 공유한다면 어떻게 공유하는지를 비교해서, 어떤 피처가 또다른 피처와 어떤 공간 관계성을 가지는지를 판단하는 데 사용되는 불 (boolean) 함수입니다.

앞의 그림을 이용해서, 녹색 원 피처를 주황색 직사각형 피처와 공간적으로 비교해서 녹색 원 피처를 검색할 것입니다. 다음과 같은 기하 서술을 사용할 수 있습니다:

**교차** ***(Intersection)***
: 도형들이 서로 교차하는지 검증합니다. 도형들이 공간적으로 교차하는 경우 (공간의 어떤 부분이라도 공유–중첩 또는 접하는 경우) 1(참) 을 반환하고 교차하지 않는 경우 0(거짓) 을 반환합니다. 앞의 그림에서, 1 번, 2 번 및 3 번 원을 반환할 것입니다.

**포함** ***(Contain)***
: B 의 포인트 가운데 A 외부에 있는 포인트가 하나도 없고 B 내부의 포인트 가운데 최소한 하나가 A 의 내부에 있는 경우에만 1(참) 을 반환합니다. 앞의 그림에서 어떤 원도 반환하지 않지만, 반대로 검색할 경우 직사각형이 1 번 원을 완전히 담고 있기 때문에 직사각형을 반환할 것입니다. 내부 *(are within)* 의 역 (□) 입니다.

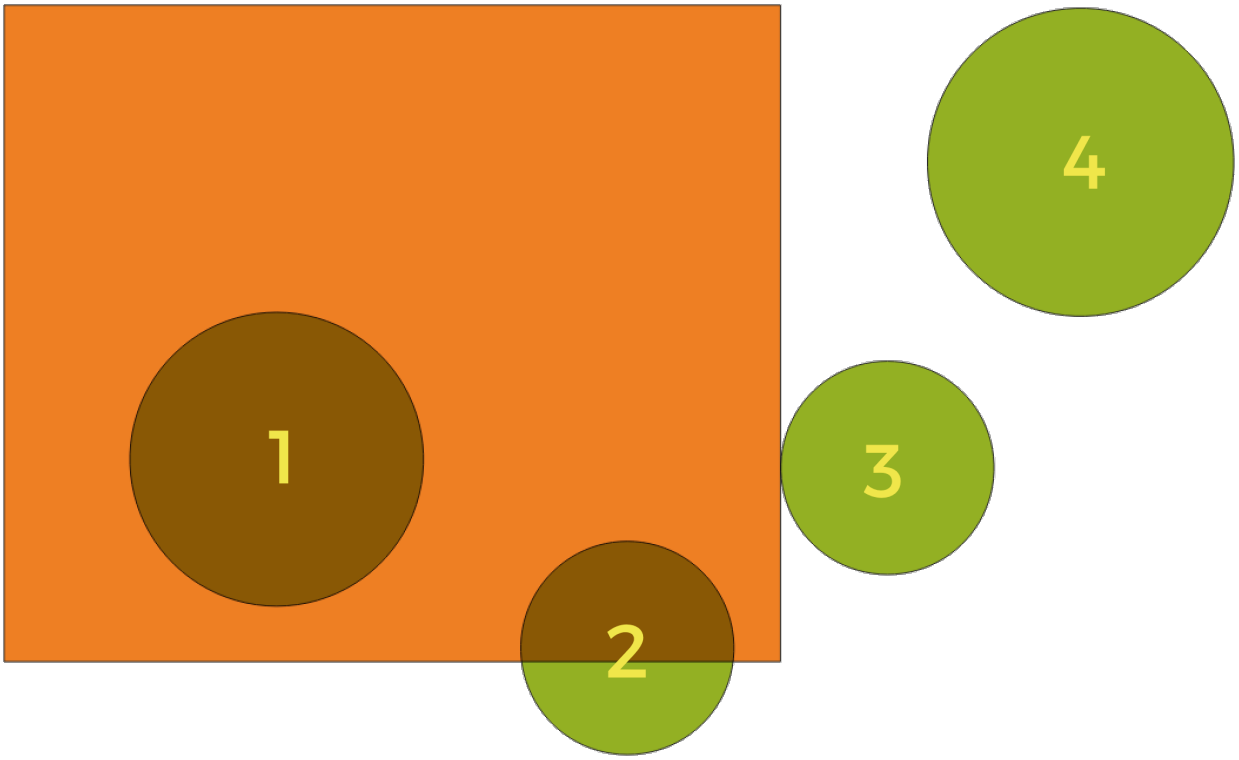



그림 24.163: 레이어들 사이의 공간 관계성 검색

**분절** ***(Disjoint)***
도형들이 공간의 어떤 부분도 공유하지 않는–중첩하지도 접하지도 않는 경우 1(참) 을 반환합니다. 4 번 원만 반환합니다.

**동등** ***(Equal)***
도형들이 정확하게 동일한 경우에만 1(참) 을 반환합니다. 아무 원도 반환하지 않을 것입니다.

**접촉** ***(Touch)***
도형이 다른 도형과 접하는지 검증합니다. 도형들이 최소한 포인트 1 개를 공유하지만 각 도형의 내부가 교차하지 않는 경우 1(참) 을 반환합니다. 3 번 원만 반환합니다.

**중첩** ***(Overlap)***
도형이 다른 도형과 중첩하는지 검증합니다. 도형들이 동일한 차원이며 공간을 공유하지만 서로를 완전히 담고 있지 않은 경우 1(참) 을 반환합니다. 2 번 원만 반환합니다.

**내부** ***(Are within)***
도형이 다른 도형 내부에 있는지 검증합니다. A 가 B 내부에 완전히 들어가 있는 경우 1(참) 을 반환합니다. 1 번 원만 반환합니다.

**공간교차** ***(Cross)***
입력 도형들이 전부는 아니지만 일부 내부 포인트를 공유하지만 입력 도형의 최고 차원보다 낮은 차원에서 실제로 가로지르는 경우 1(참) 을 반환합니다. 예를 들어, 폴리곤을 가로지르는 라인은 라인으로서 공간교차할 것입니다 (참). 라인을 가로지르는 라인은 포인트로서 공간교차할 것입니다 (참). 폴리곤을 가로지르는 폴리곤은 폴리곤으로서 가로지릅니다 (거짓). 앞의 그림에서, 아무 원도 반환하지 않을 것입니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Extract features from** | `INPUT` | [vector: geometry] | 입력 벡터 레이어 |
| **Where the features (geometric predicate)** | `PREDICATE` | [enumeration] [list] 기본값: [0] | 입력 피처가 선택될 수 있으려면 입력 피처와 교차 피처가 가져야 할 공간 관계의 유형입니다. 다음 가운데 하나 이상을 선택할 수 있습니다:<br>• 0 —교차 (intersect)<br>• 1 —포함 (contain)<br>• 2 —격리 (disjoint)<br>• 3 —동등 (equal)<br>• 4 —접촉 (touch)<br>• 5 —중첩 (overlap)<br>• 6 —내부 (are within)<br>• 7 —공간 교차 (cross)<br>조건을 하나 이상 선택한 경우, 적어도 하나는 (OR 연산자) 추출할 피처를 만족시켜야 합니다. |
| **By comparing to the features from** | `INTERSECT` | [vector: geometry] | 교차 벡터 레이어 |
| **Extracted (location)** | `OUTPUT` | [same as input] 기본값: `[Create temporary layer]` | Specify the output vector layer for the features that have the chosen spatial relationship(s) with one or more features in the comparison layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Extracted (location)** | `OUTPUT` | [same as input] | 비교 레이어의 하나 이상의 피처와 선택한 공간 관계 (들) 을 가지는 입력 레이어의 피처를 저장한 산출 벡터 레이어 |

### 파이썬 코드

**Algorithm ID**: `native:extractbylocation`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 거리 안에서 추출하기

입력 레이어에서 조건을 만족하는 피처만 담고 있는 새 벡터 레이어를 생성합니다. 추가적인 참조 레이어에 있는 피처로부터 지정한 최장 거리 안에 있는 피처를 복사합니다.

**더 보기**

거리 안에서 선택하기, 위치로 추출하기

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Extract features from** | `INPUT` | [vector: geometry] | 피처를 복사해올 입력 벡터 레이어 |
| **By comparing to the features from** | `REFERENCE` | [vector: geometry] | 피처의 근접성을 이용할 벡터 레이어 |
| **Where the features are within** | `DISTANCE` | [numeric: double]<br>기본값: 100.0 | 참조 피처 주변에서 이 거리 안에 들어오는 입력 피처를 선택하기 위한 최장 거리 |
| **Modify current selection by** | `METHOD` | [enumeration]<br>기본값: 0 | 알고리즘의 선택 작업을 어떻게 관리해야 할지 지정합니다. 다음 가운데 하나를 선택할 수 있습니다:<br>• 0 — 새 선택 집합 생성<br>• 1 — 현재 선택 집합에 추가<br>• 2 — 현재 선택 집합 내에서 선택<br>• 3 — 현재 선택 집합 내에서 제거 |
| **Extracted (location)** | `OUTPUT` | [same as input]<br>기본값: `[Create temporary layer]` | Specify the output vector layer for the features that are within the set distance from reference features. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Extracted (location)** | `OUTPUT` | [same as input] | 참조 피처로부터의 거리 조건을 만족하는 입력 레이어의 피처를 가진 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:extractwithindistance`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 도형 유형으로 필터링하기

피처를 도형 유형으로 필터링합니다. 입력 피처가 포인트, 라인, 또는 폴리곤 도형을 가지고 있느냐에 따라, 입력 피처를 서로 다른 산출물로 전송 (□□) 할 것입니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: any] | 평가할 레이어 |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Point features**<br>선택적 | `POINTS` | [vector: point] | 포인트를 가진 레이어 |
| **Line features**<br>선택적 | `LINES` | [vector: line] | 라인을 가진 레이어 |
| **Polygon features**<br>선택적 | `POLYGONS` | [vector: polygon] | 폴리곤을 가진 레이어 |
| **Features with no geometry**<br>선택적 | `NO_GEOMETRY` | [vector: table] | 무 (□) 도형 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:filterbygeometry`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 랜덤하게 추출하기

벡터 레이어를 받아 입력 레이어에 있는 피처의 하위 집합만을 담고 있는 새 벡터 레이어를 생성합니다.

하위 집합은 피처 ID 를 사용해서 랜덤하게 정의됩니다. 백분율 또는 개수 값을 사용해서 하위 집합에 들어갈 피처의 총 개수를 정의합니다.

**경고**

This algorithm drops existing primary keys or FID values and regenerate them in output layers.

**더 보기**

랜덤하게 선택하기

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: any] | 피처를 선택할 소스 벡터 레이어 |
| **Method** | `METHOD` | [enumeration] 기본값: 0 | 랜덤 선택 메소드를 설정합니다. 다음 가운데 하나를 선택할 수 있습니다:<br>• 0 — 선택한 피처의 개수<br>• 1 — 선택한 피처의 백분율 |
| **Number/percentage of selected features** | `NUMBER` | [numeric: integer] 기본값: 10 | 선택할 피처들의 개수 또는 백분율 |
| **Extracted (random)** | `OUTPUT` | [same as input] 기본값: `[Create temporary layer]` | Specify the output vector layer for the randomly selected features. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Extracted (random)** | `OUTPUT` | [same as input] | 입력 레이어에서 랜덤하게 선택한 피처를 담고 있는 벡터 레이어 |

### 파이썬 코드

**Algorithm ID**: `native:randomextract`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 하위 집합 내에서 랜덤하게 추출하기

벡터 레이어를 받아 입력 레이어에 있는 피처의 하위 집합만을 담고 있는 새 벡터 레이어를 생성합니다.

하위 집합은 피처 ID 를 사용해서 랜덤하게 정의됩니다. 백분율 또는 개수 값을 사용해서 하위 집합에 들어갈 피처의 총 개수를 정의합니다. 백분율/개수 값은 전체 레이어가 아니라 각 범주에 적용됩니다. 범주는 지정한 속성에 따라 정의됩니다.

**더 보기**

하위 집합 내에서 랜덤하게 선택하기

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: any] | 피처를 선택할 벡터 레이어 |
| **ID field** | `FIELD` | [tablefield: any] | 피처를 선택할 소스 벡터 레이어의 범주 |
| **Method** | `METHOD` | [enumeration]<br>기본값: 0 | 랜덤 선택 메소드를 설정합니다. 다음 가운데 하나를 선택할 수 있습니다:<br>• 0 —선택한 피처의 개수<br>• 1 —선택한 피처의 백분율 |
| **Number/percentage of selected features** | `NUMBER` | [numeric: integer]<br>기본값: 10 | 선택할 피처들의 개수 또는 백분율 |

다음 페이지에 계속

표 24.801 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Extracted (random stratified)** | `OUTPUT` | [same as input]<br>기본값: `[Create temporary layer]` | Specify the output vector layer for the randomly selected features. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Extracted (random stratified)** | `OUTPUT` | [same as input] | 입력 레이어에서 랜덤하게 선택한 피처를 담고 있는 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `qgis:randomextractwithinsubsets`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 랜덤하게 선택하기

벡터 레이어를 받아 입력 레이어에 있는 피처의 하위 집합을 선택합니다. 이 알고리즘은 새 레이어를 생성하지 않습니다.

하위 집합은 피처 ID 를 사용해서 랜덤하게 정의됩니다. 백분율 또는 개수 값을 사용해서 하위 집합에 들어갈 피처의 총 개수를 정의합니다.

**기본 메뉴**: *Vector → Research Tools*

**더 보기**

랜덤하게 추출하기

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | INPUT | [vector: any] | 피처를 선택할 벡터 레이어 |
| **Method** | METHOD | [enumeration] 기본값: 0 | 랜덤 선택 메소드를 설정합니다. 다음 가운데 하나를 선택할 수 있습니다:<br>• 0 —선택한 피처의 개수<br>• 1 —선택한 피처의 백분율 |
| **Number/percentage of selected features** | NUMBER | [numeric: integer] 기본값: 10 | 선택할 피처들의 개수 또는 백분율 |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | INPUT | [same as input] | 선택한 피처를 가진 입력 레이어 |

### 파이썬 코드

**알고리즘 ID**: `qgis:randomselection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 하위 집합 내에서 랜덤하게 선택하기

벡터 레이어를 받아 입력 레이어에 있는 피처의 하위 집합을 선택합니다. 이 알고리즘은 새 레이어를 생성하지 않습니다.

하위 집합은 피처 ID 를 사용해서 랜덤하게 정의됩니다. 백분율 또는 개수 값을 사용해서 하위 집합에 들어갈 피처의 총 개수를 정의합니다.

백분율/개수 값은 전체 레이어가 아니라 각 범주에 적용됩니다.

범주는 지정한 속성에 따라 정의됩니다. 알고리즘을 위한 입력 파라미터로 속성을 지정합니다.

어떤 새 산출물도 생성하지 않습니다.

**기본 메뉴**: *Vector ▸ Research Tools*

**더 보기**

하위 집합 내에서 랜덤하게 추출하기

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: any] | 피처를 선택할 벡터 레이어 |
| **ID field** | `FIELD` | [tablefield: any] | 피처를 선택할 입력 레이어의 범주 |
| **Method** | `METHOD` | [enumeration] 기본값: 0 | 랜덤 선택 메소드를 설정합니다. 다음 가운데 하나를 선택할 수 있습니다:<br>• 0 —선택한 피처의 개수<br>• 1 —선택한 피처의 백분율 |
| **Number/percentage of selected features** | `NUMBER` | [numeric: integer] 기본값: 10 | 선택할 피처들의 개수 또는 백분율 |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [same as input] | 선택한 피처를 가진 입력 레이어 |

### 파이썬 코드

**알고리즘 ID**: `qgis:randomselectionwithinsubsets`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 속성으로 선택하기

벡터 레이어에서 선택 집합을 생성합니다.

피처를 선택하는 기준은 입력 레이어의 속성값을 바탕으로 합니다.

**더 보기**

속성으로 추출하기

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | INPUT | [vector: any] | 피처를 선택할 벡터 레이어 |
| **Selection attribute** | FIELD | [tablefield: any] | 레이어의 필터링 필드 |
| **Operator** | OPERATOR | [enumeration]<br>기본값: 0 | 다음과 같은 많은 연산자를 사용할 수 있습니다:<br>• 0 —=<br>• 1 —≠<br>• 2 —><br>• 3 —>=<br>• 4 —<<br>• 5 —<=<br>• 6 —begins with (…으로 시작)<br>• 7 —contains (포함)<br>• 8 —is null (NULL 값임)<br>• 9 —is not null (NULL 값 아님)<br>• 10 —does not contain (포함하지 않음) |
| **Value**<br>선택적 | VALUE | [string] | 평가할 값 |
| **Modify current selection by** | METHOD | [enumeration]<br>기본값: 0 | 알고리즘의 선택 작업을 어떻게 관리해야 할지 지정합니다. 다음 가운데 하나를 선택할 수 있습니다:<br>• 0 —새 선택 집합 생성<br>• 1 —현재 선택 집합에 추가<br>• 2 —현재 선택 집합에서 제거<br>• 3 –현재 선택 집합 내에서 선택 |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | INPUT | [same as input] | 선택한 피처를 가진 입력 레이어 |

### 파이썬 코드

**알고리즘 ID**: `qgis:selectbyattribute`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 표현식으로 선택하기

벡터 레이어에서 선택 집합을 생성합니다.

피처를 선택하는 기준은 QGIS 표현식을 바탕으로 합니다. 표현식에 대해 더 자세히 알고 싶다면표현식 을 참조하세요.

**더 보기**

표현식으로 추출하기

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: any] | 입력 벡터 레이어 |
| **Expression** | `EXPRESSION` | [expression] | 입력 레이어를 필터링할 표현식 |
| **Modify current selection by** | `METHOD` | [enumeration]<br>기본값: 0 | 알고리즘의 선택 작업을 어떻게 관리해야 할지 지정합니다. 다음 가운데 하나를 선택할 수 있습니다:<br>• 0 —새 선택 집합 생성<br>• 1 —현재 선택 집합에 추가<br>• 2 —현재 선택 집합에서 제거<br>• 3 –현재 선택 집합 내에서 선택 |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [same as input] | 선택한 피처를 가진 입력 레이어 |

### 파이썬 코드

**알고리즘 ID**: `qgis:selectbyexpression`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 위치로 선택하기

벡터 레이어에서 선택 집합을 생성합니다.

피처를 선택하는 기준은 입력 레이어의 각 피처와 추가 레이어의 피처들 사이의 공간 관계를 바탕으로 합니다.

**기본 메뉴**: *Vector → Research Tools*

**더 보기**

위치로 추출하기, 거리 안에서 선택하기

### 공간 관계성 탐색하기

기하 서술 (geometric predicates) 은 피처들의 도형들이 공간 일부분을 공유하는지, 공유한다면 어떻게 공유하는지를 비교해서, 어떤 피처가 또다른 피처와 어떤 공간 관계성을 가지는지를 판단하는 데 사용되는 불 (boolean) 함수입니다.

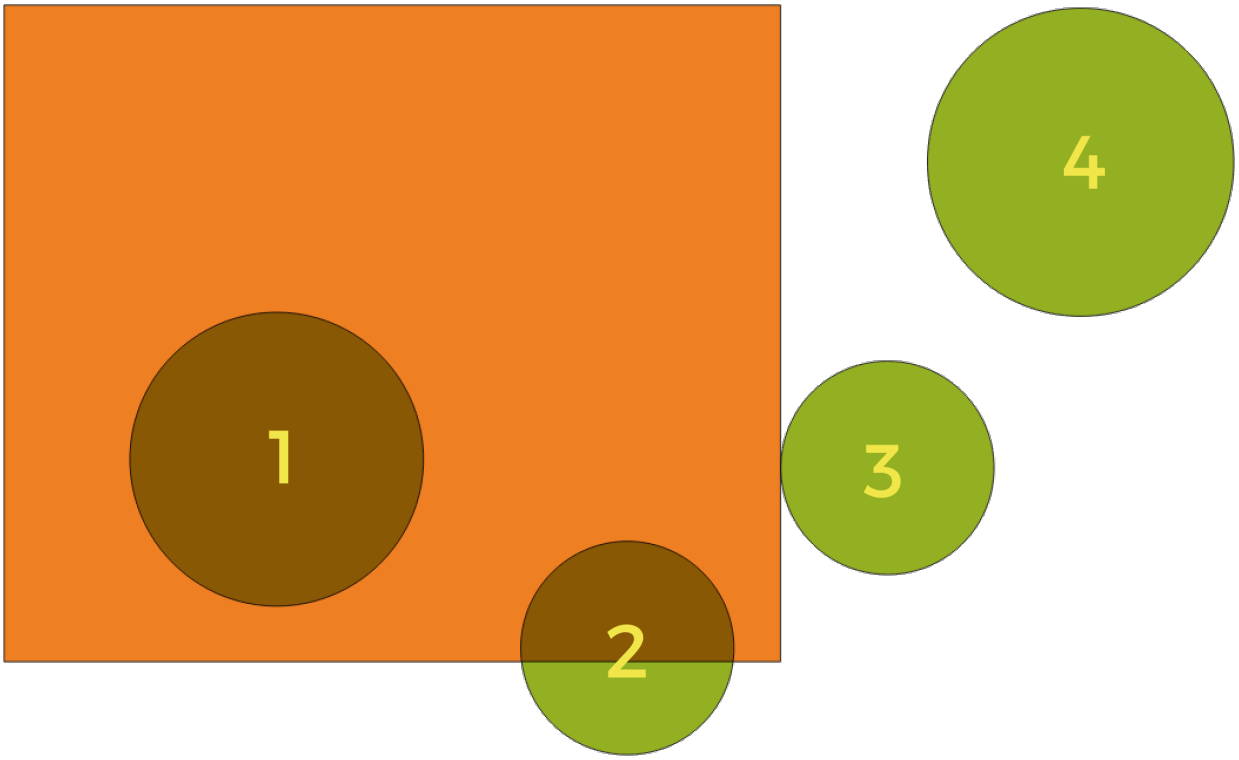



그림 24.164: 레이어들 사이의 공간 관계성 검색

앞의 그림을 이용해서, 녹색 원 피처를 주황색 직사각형 피처와 공간적으로 비교해서 녹색 원 피처를 검색할 것입니다. 다음과 같은 기하 서술을 사용할 수 있습니다:

**교차** ***(Intersection)***
: 도형들이 서로 교차하는지 검증합니다. 도형들이 공간적으로 교차하는 경우 (공간의 어떤 부분이라도 공유–중첩 또는 접하는 경우) 1(참) 을 반환하고 교차하지 않는 경우 0(거짓) 을 반환합니다. 앞의 그림에서, 1 번, 2 번 및 3 번 원을 반환할 것입니다.

**포함** ***(Contain)***
B 의 포인트 가운데 A 외부에 있는 포인트가 하나도 없고 B 내부의 포인트 가운데 최소한 하나가 A 의 내부에 있는 경우에만 1(참) 을 반환합니다. 앞의 그림에서 어떤 원도 반환하지 않지만, 반대로 검색할 경우 직사각형이 1 번 원을 완전히 담고 있기 때문에 직사각형을 반환할 것입니다. 내부 *(are within)* 의 역 (□) 입니다.

**분절** ***(Disjoint)***
도형들이 공간의 어떤 부분도 공유하지 않는–중첩하지도 접하지도 않는 경우 1(참) 을 반환합니다. 4 번 원만 반환합니다.

**동등** ***(Equal)***
도형들이 정확하게 동일한 경우에만 1(참) 을 반환합니다. 아무 원도 반환하지 않을 것입니다.

**접촉** ***(Touch)***
도형이 다른 도형과 접하는지 검증합니다. 도형들이 최소한 포인트 1 개를 공유하지만 각 도형의 내부가 교차하지 않는 경우 1(참) 을 반환합니다. 3 번 원만 반환합니다.

**중첩** ***(Overlap)***
도형이 다른 도형과 중첩하는지 검증합니다. 도형들이 동일한 차원이며 공간을 공유하지만 서로를 완전히 담고 있지 않은 경우 1(참) 을 반환합니다. 2 번 원만 반환합니다.

**내부** ***(Are within)***
도형이 다른 도형 내부에 있는지 검증합니다. A 가 B 내부에 완전히 들어가 있는 경우 1(참) 을 반환합니다. 1 번 원만 반환합니다.

**공간교차** ***(Cross)***
입력 도형들이 전부는 아니지만 일부 내부 포인트를 공유하지만 입력 도형의 최고 차원보다 낮은 차원에서 실제로 가로지르는 경우 1(참) 을 반환합니다. 예를 들어, 폴리곤을 가로지르는 라인은 라인으로서 공간교차할 것입니다 (참). 라인을 가로지르는 라인은 포인트로서 공간교차할 것입니다 (참). 폴리곤을 가로지르는 폴리곤은 폴리곤으로서 가로지릅니다 (거짓). 앞의 그림에서, 아무 원도 반환하지 않을 것입니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Select features from** | `INPUT` | [vector: geometry] | 입력 벡터 레이어 |
| **Where the features (geometric predicate)** | `PREDICATE` | [enumeration] [list] 기본값: [0] | 입력 피처가 선택될 수 있으려면 입력 피처와 교차 피처가 가져야 할 공간 관계의 유형입니다. 다음 가운데 하나 이상을 선택할 수 있습니다:<br>• 0 —교차 (intersect)<br>• 1 —포함 (contain)<br>• 2 —격리 (disjoint)<br>• 3 —동등 (equal)<br>• 4 —접촉 (touch)<br>• 5 —중첩 (overlap)<br>• 6 —내부 (are within)<br>• 7 —공간 교차 (cross)<br>조건을 하나 이상 선택한 경우, 적어도 하나는 (OR 연산자) 추출할 피처를 만족시켜야 합니다. |
| **By comparing to the features from** | `INTERSECT` | [vector: geometry] | 교차 벡터 레이어 |

다음 페이지에 계속

표 24.811 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Modify current selection by** | METHOD | [enumeration] 기본값: 0 | 알고리즘의 선택 작업을 어떻게 관리해야 할지 지정합니다. 다음 가운데 하나를 선택할 수 있습니다:<br>• 0 —새 선택 집합 생성<br>• 1 —현재 선택 집합에 추가<br>• 2 —현재 선택 집합 내에서 선택<br>• 3 —현재 선택 집합 내에서 제거 |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | INPUT | [same as input] | 선택한 피처를 가진 입력 레이어 |

### 파이썬 코드

**Algorithm ID**: `native:selectbylocation`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 거리 안에서 선택하기

벡터 레이어에서 선택 집합을 생성합니다. 추가적인 참조 레이어에 있는 피처로부터 지정한 최장 거리 안에 있는 피처를 선택합니다.

**더 보기**

거리 안에서 추출하기, 위치로 선택하기

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Select features from** | INPUT | [vector: geometry] | 피처를 선택할 입력 벡터 레이어 |
| **By comparing to the features from** | REFERENCE | [vector: geometry] | 피처의 근접성을 이용할 벡터 레이어 |

다음 페이지에 계속

표 24.813 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Where the features are within** | DISTANCE | [numeric: double]<br>기본값: 100.0 | 참조 피처 주변에서 이 거리 안에 들어오는 입력 피처를 선택하기 위한 최장 거리 |
| **Modify current selection by** | METHOD | [enumeration]<br>기본값: 0 | 알고리즘의 선택 작업을 어떻게 관리해야 할지 지정합니다. 다음 가운데 하나를 선택할 수 있습니다:<br>• 0 —새 선택 집합 생성<br>• 1 —현재 선택 집합에 추가<br>• 2 —현재 선택 집합 내에서 선택<br>• 3 —현재 선택 집합 내에서 제거 |

#### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | INPUT | [same as input] | 선택한 피처를 가진 입력 레이어 |

#### 파이썬 코드

**알고리즘 ID**: `native:selectwithindistance`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 24.1.29 벡터 테이블

#### 자동 증가 필드 추가하기

벡터 레이어에 각 피처에 대해 연속 값을 가지는 새 정수형 필드를 추가합니다.

이 필드는 레이어에 있는 피처를 위한 유일 ID 로 사용할 수 있습니다. 입력 레이어에는 새 속성을 추가하지 않지만 그 대신 새 레이어를 생성할 것입니다.

증가하는 일련의 숫자의 맨 처음 시작값을 지정할 수 있습니다. 증가하는 일련의 숫자는 그룹화 필드를 바탕으로 할 수 있고, 피처의 정렬 순서도 지정할 수 있습니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | INPUT | [vector: any] | 입력 벡터 레이어 |
| **Field name** | FIELD_NAME | [string]<br>기본값: 'AUTO' | 자동 증가하는 값을 가진 필드의 이름 |
| **Start values at**<br>선택적 | START | [numeric: integer]<br>기본값: 0 | 증가하는 번호의 맨 처음 숫자를 선택합니다. |
| **Modulus value**<br>선택적 | MODULUS | [numeric: integer]<br>기본값: 0 | 선택적인 계수 값 (modulus value) 을 지정하면 필드 값이 계수 값에 도달할 때마다 개수를 START 로 재시작할 것입니다. 0 으로 설정하면 재시작하지 않습니다. |
| **Group values by**<br>선택적 | GROUP_FIELDS | [tablefield: any] [list] | 그룹화할 필드 (들) 를 선택합니다. 전체 레이어에 대해 연속되는 번호 한 무리를 적용하는 대신, 이 필드 (들) 의 결합으로 반환되는 각 값에 대해 개별 번호 무리 (들) 을 적용합니다. |
| **Sort expression**<br>선택적 | SORT_EXPRESSION | [expression] | 표현식을 사용해서 레이어 전체 수준에서 피처들을 정렬하거나, 필드 그룹을 설정한 경우 필드 그룹을 기반으로 피처를 정렬합니다. |
| **Sort ascending** | SORT_ASCENDING | [boolean]<br>기본값: True | sort expression 을 설정한 경우, 이 옵션을 사용해서 피처에 값을 할당하는 순서를 제어할 수 있습니다. |
| **Sort nulls first** | SORT_NULLS_FIRS | [boolean]<br>기본값: False | sort expression 을 설정한 경우, 이 옵션을 사용해서 *NULL* 값을 맨 처음 정렬시킬지 또는 마지막으로 정렬시킬지 설정할 수 있습니다. |
| **Incremented** | OUTPUT | [same as input]<br>기본값: [Create temporary layer] | Specify the output vector layer with the auto increment field. *One of*:<br>• 임시 레이어 생성 (TEMPORARY_OUTPUT)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Incremented** | OUTPUT | [same as input] | 자동 증가 필드를 가진 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:addautoincrementalfield`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 속성 테이블에 필드 추가하기

벡터 레이어에 새 필드를 추가합니다.

속성의 이름 및 특성은 파라미터로 정의합니다.

입력 레이어에는 새 속성을 추가하지 않지만 그 대신 새 레이어를 생성할 것입니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: any] | 입력 레이어 |
| **Field name** | `FIELD_NAME` | [string] | 새 필드의 이름 |
| **Field type** | `FIELD_TYPE` | [enumeration]<br>기본값: 0 | 새 필드의 유형을 다음 가운데 선택할 수 있습니다:<br>• 0 —정수형 (32 비트)<br>• 1 —배정밀도 부동소수점 실수형 (double)<br>• 2 —텍스트 (문자열)<br>• 3 —불 (Boolean)<br>• 4 —날짜<br>• 5 —시간<br>• 6 —날짜 & 시간<br>• 7 —바이너리 객체 (BLOB)<br>• 8 —문자열 목록<br>• 9 —정수형 목록<br>• 10 —배정밀도 부동소수점 실수형 목록 |
| **Field length** | `FIELD_LENGTH` | [numeric: integer]<br>기본값: 10 | 필드의 길이 |
| **Field precision** | `FIELD_PRECISION` | [numeric: integer]<br>기본값: 0 | 필드의 정밀도. 부동소수점 필드 유형에 유용합니다. |
| **Field alias**<br>선택적 | `FIELD_ALIAS` | [string] | 필드의 별명으로 사용할 이름을 설정합니다. 모든 서식 유형이 별명을 지원하진 않습니다. |
| **Field comment**<br>선택적 | `FIELD_COMMENT` | [string] | 필드를 설명하는 주석을 저장합니다. 모든 서식 유형이 주석을 지원하진 않습니다. |

다음 페이지에 계속

표 24.817 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Added** | `OUTPUT` | [same as input]<br>기본값: `[Create temporary layer]` | Specify the output vector layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Added** | `OUTPUT` | [same as input] | 새 필드를 추가한 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:addfieldtoattributestable`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 유일값 인덱스 필드 추가하기

벡터 레이어를 받아 그 속성에 새 숫자 필드를 추가합니다.

이 필드의 값은 지정한 속성의 값에 대응합니다. 따라서 동일한 속성값을 가진 피처는 새 필드에 동일한 값을 가지게 될 것입니다.

이 알고리즘은 지정한 속성과 동등한, 동일한 범주를 정의하는 숫자 필드를 생성합니다.

입력 레이어에는 새 속성을 추가하지 않지만 그 대신 새 레이어를 생성할 것입니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: any] | 입력 레이어 |
| **Class field** | `FIELD` | [tablefield: any] | 이 필드에 동일한 값을 가진 피처는 동일한 인덱스를 얻게 될 것입니다. |

다음 페이지에 계속

표 24.819 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output field name** | `FIELD_NAME` | [string]<br>기본값: 'NUM_FIELD' | 인덱스를 담고 있는 새 필드의 이름 |
| **Layer with index field** | `OUTPUT` | [same as input]<br>기본값: `[Create temporary layer]` | Vector layer with the numeric field containing indexes. *One of*:<br>• 산출물 건너뛰기<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |
| **Class summary** | `SUMMARY_OUTPUT` | [vector: table]<br>기본값: `[Skip output]` | Specify the table to contain the summary of the class field mapped to the corresponding unique value. *One of*:<br>• 산출물 건너뛰기<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Layer with index field** | `OUTPUT` | [same as input] | 인덱스를 담고 있는 숫자 필드를 가진 벡터 레이어 |
| **Class summary** | `SUMMARY_OUTPUT` | [vector: table] | 대응하는 유일값에 맵핑된 범주 필드의 요약을 담은 테이블 |

### 파이썬 코드

**알고리즘 ID**: `native:adduniquevalueindexfield`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 레이어에 X/Y 필드 추가하기

포인트 레이어에 X 및 Y(또는 위도 및 경도) 필드를 추가합니다. X/Y 필드는 레이어와 다른 좌표계로 계산할 수 있습니다. (예를 들면 투영된 좌표계에서 레이어를 위한 위도/경도 필드를 생산할 수 있습니다.)

☑ 포인트 피처를제자리 피처 수정 할 수 있습니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: point] | 입력 레이어 |
| **Coordinate system** | `CRS` | [crs]<br>기본값:<br>"EPSG:4326" | 생성된 X 및 Y 필드에 사용할 좌표계 |
| **Field prefix**<br>선택적 | `PREFIX` | [string] | 입력 레이어 필드와 이름 충돌을 피하기 위해 새 필드명 앞에 붙일 접두어 |
| **Added fields** | `OUTPUT` | [vector: point]<br>기본값: `[Create temporary layer]` | Specify the output layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Added fields** | `OUTPUT` | [vector: point] | 입력 레이어와 동일하지만 새 `x` 및 `y` 필드 2 개를 가진 산출 레이어 |

### 파이썬 코드

**Algorithm ID**: `native:addxyfields`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 고급 파이썬 필드 계산기

벡터 레이어에 각 피처에 표현식을 적용해서 나온 값을 새 속성으로 추가합니다.
이 표현식은 파이썬 함수로 정의됩니다.

**경고**

This algorithm is a potential security risk if executed with unchecked inputs, and may result in system damage or data leaks.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: any] | 입력 벡터 레이어 |
| **Result field name** | `FIELD_NAME` | [string]<br>기본값: 'NewField' | 새 필드의 이름 |
| **Field type** | `FIELD_TYPE` | [enumeration]<br>기본값: 0 | 새 필드의 데이터 유형입니다. 다음 가운데 하나로 지정할 수 있습니다:<br>• 0 —정수형 (32 비트)<br>• 1 —배정밀도 부동소수점 실수형 (double)<br>• 2 —텍스트 (문자열)<br>• 3 —불 (Boolean)<br>• 4 —날짜<br>• 5 —시간<br>• 6 —날짜 & 시간<br>• 7 —바이너리 객체 (BLOB)<br>• 8 —문자열 목록<br>• 9 —정수형 목록<br>• 10 —배정밀도 부동소수점 실수형 목록 |
| **Field length** | `FIELD_LENGTH` | [numeric: integer]<br>기본값: 10 | 필드의 길이 |
| **Field precision** | `FIELD_PRECISION` | [numeric: integer]<br>기본값: 3 | 필드의 정밀도. 부동소수점 필드 유형에 유용합니다. |
| **Global expression**<br>선택적 | `GLOBAL` | [string] | 계산기가 입력 레이어의 모든 피처에 대해 반복 작업을 시작하기 전에 전체 수준 (global) 표현식 부분의 코드를 한번만 실행할 것입니다. 따라서, 필요한 모듈을 가져오거나 차후의 계산에 사용될 변수를 계산하는 표현식을 여기에 작성해야 합니다. |

다음 페이지에 계속

표 24.823 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Formula** | FORMULA | [string] | 값을 구할 파이썬 공식입니다. 예를 들어 입력 폴리곤 레이어의 면적을 계산하려면 다음 공식을 추가하면 됩니다:<br>`value = $geom.area()`<br>To access an existing field use its name enclosed in the angle brackets. For example, to uppercase values of the "address"field:<br>`value = <address>.upper()` |
| **Calculated** | OUTPUT | [same as input]<br>기본값: [Create temporary layer] | 새로 계산된 필드를 가질 산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다:<br>• 임시 레이어 생성 (TEMPORARY_OUTPUT)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Calculated** | OUTPUT | [same as input] | 새로 계산된 필드를 가진 산출 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `qgis:advancedpythonfieldcalculator`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 필드 (들) 삭제하기

벡터 레이어를 받아 입력 레이어와 동일하지만 선택한 열을 삭제 (drop) 한 새 벡터 레이어를 생성합니다.

**더 보기**

필드 유지

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: any] | 필드 (들) 를 삭제할 입력 벡터 레이어 |
| **Fields to drop** | `COLUMN` | [tablefield: any] [list] | 삭제할 필드 (들) |
| **Remaining fields** | `OUTPUT` | [same as input]<br>기본값: `[Create temporary layer]` | Specify the output vector layer with the remaining fields. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Remaining fields** | `OUTPUT` | [same as input] | 남아 있는 필드를 담은 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:deletecolumn`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### HStore 필드 조각내기

HStore 필드에 있는 모든 유일 키마다 새 필드를 추가한 입력 레이어의 복사본을 생성합니다.

예상 필드 목록은 쉼표로 구분된 부가적인 목록입니다. 이 목록을 설정한 경우, 이 목록에 있는 필드만 추가하고 HStore 필드를 업데이트합니다. 유일 키는 기본적으로 모두 추가됩니다.

The PostgreSQL HStore is a simple key-value store used in PostgreSQL and GDAL (when reading an OSM file with the `other_tags` field.

#### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | 입력 벡터 레이어 |
| **HStore field** | `FIELD` | [tablefield: any] | 삭제할 필드 (들) |
| **Expected list of fields separated by a comma**<br>선택적 | `EXPECTED_FIELDS` | [string]<br>기본값: ' ' | 추출할 필드의 쉼표 구분 목록입니다. 이 키들을 제거해서 HStore 필드를 업데이트할 것입니다. |
| **Exploded** | `OUTPUT` | [same as input]<br>기본값: `[Create temporary layer]` | 산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

#### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Exploded** | `OUTPUT` | [same as input] | 산출 벡터 레이어 |

#### 파이썬 코드

**알고리즘 ID**: `native:explodehstorefield`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 바이너리 필드 추출하기

바이너리 필드의 내용을 개별 파일로 저장해서 추출합니다. 소스 테이블에 있는 속성에서, 또는 좀 더 복잡한 표현식을 바탕으로 나온 값을 사용해서 파일명을 생성할 수 있습니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: any] | 바이너리 데이터를 담고 있는 입력 벡터 레이어 |
| **Binary field** | `FIELD` | [tablefield: any] | 바이너리 데이터를 담고 있는 필드 |
| **File name** | `FILENAME` | [expression] | 각 산출 파일을 명명하기 위한 필드 또는 표현식 기반 텍스트 |
| **Destination folder** | `FOLDER` | [folder]<br>기본값: `[Save to temporary folder]` | Folder in which to store the output files. *One of*:<br>• 임시 디렉터리에 저장<br>• 디렉터리에 저장 |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Folder** | `FOLDER` | [folder] | 산출 파일을 담고 있는 폴더 |

### 파이썬 코드

**알고리즘 ID**: `native:extractbinary`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 필드 계산기

필드 계산기를 엽니다. (표현식 을 참조하세요.) 지원되는 표현식과 함수를 모두 사용할 수 있습니다.

표현식의 결과물로 새 레이어를 생성합니다.

필드 계산기는모델 설계자 에서 매우 유용하게 사용됩니다.

**참고**

This algorithm uses ellipsoid based measurements and respects the current *ellipsoid settings*.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | INPUT | [vector: any] | 계산할 레이어 |
| **Output field name** | FIELD_NAME | [string] | 계산 결과를 저장할 필드의 이름 |
| **Output field type** | FIELD_TYPE | [enumeration]<br>기본값: 0 | 필드의 데이터 유형입니다. 다음 가운데 하나로 지정할 수 있습니다:<br>• 0 —배정밀도 부동소수점 실수형 (double)<br>• 1 —정수형 (32 비트)<br>• 2 —텍스트 (문자열)<br>• 3 —날짜<br>• 4 —시간<br>• 5 —날짜 & 시간<br>• 6 —불 (Boolean)<br>• 7 —바이너리 객체 (BLOB)<br>• 8 —문자열 목록<br>• 9 —정수형 목록<br>• 10 —배정밀도 부동소수점 실수형 목록 |
| **Output field width** | FIELD_LENGTH | [numeric: integer]<br>기본값: 0 | 산출 필드의 길이 (최소값 0) |
| **Field precision** | FIELD_PRECISION | [numeric: integer]<br>기본값: 0 | 산출 필드의 정밀도 (최소 0, 최대 15) |
| **Create new field** | NEW_FIELD | [boolean]<br>기본값: True | 산출 필드를 새 필드로 생성할지 여부 |
| **Formula** | FORMULA | [expression] | 결과를 계산하는 데 사용할 공식 |
| **Calculated** | OUTPUT | [same as input]<br>기본값: [Create temporary layer] | 산출 레이어를 지정합니다.<br>• 임시 레이어 생성 (TEMPORARY_OUTPUT)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Calculated** | OUTPUT | [same as input] | 계산된 필드 값을 가진 산출 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:fieldcalculator`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 필드 재구성하기

벡터 레이어의 속성 테이블 구조를 편집할 수 있습니다.

필드 맵핑을 사용해서 필드의 유형 및 이름을 수정할 수 있습니다.

원본 레이어는 수정하지 않습니다. 지정한 필드 맵핑에 따라 수정된 속성 테이블을 담고 있는 새 레이어를 생성합니다.

> **참고**
>
> 필드에 대한제약조건 을 가진 템플릿 레이어를 사용하는 경우, 해당 정보가 배경 색상과 툴팁을 가진 위젯에 표시됩니다. 환경 설정 도중 이 정보를 힌트로 취급하십시오. 이 알고리즘은 산출 레이어에 어떤 제약조건도 추가하지 않으며, 또한 제약조건을 확인하거나 강제하지도 않을 것입니다.

필드 재구성 (refactor field) 알고리즘은:

- 필드 이름 및 유형을 변경할 수 있습니다.
- 필드를 추가하거나 제거할 수 있습니다.
- 필드를 재정렬시킬 수 있습니다.
- 표현식을 기반으로 새 필드를 계산할 수 있습니다.
- 다른 레이어에서 필드 목록을 불러올 수 있습니다.

> **참고**
>
> This algorithm uses ellipsoid based measurements and respects the current *ellipsoid settings*.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: any] | 수정할 레이어 |

다음 페이지에 계속

표 24.833 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Fields mapping** | `FIELDS_MAPPING` | [list] | 산출 필드와 그 정의를 담은 목록입니다. 내장된 테이블은 소스 레이어의 모든 필드의 목록을 담고 있으며, 이 필드들을 편집할 수 있습니다:<br>•  버튼을 클릭해서 새 필드를 추가할 수 있습니다.<br>•  버튼을 클릭해서 선택한 필드를 삭제할 수 있습니다.<br>•  및  버튼을 클릭해서 선택한 필드의 순서를 변경할 수 있습니다.<br>•  버튼을 클릭해서 기본 뷰로 리셋할 수 있습니다.<br>•  버튼을 클릭해서 필드 목록에서 선택한 집합을 반전시킬 수 있습니다.<br>재사용하고 싶은 각 필드마다, 다음 옵션을 설정해줘야 합니다:<br>***Source expression*** **(`expression`) [expression]**<br>입력 레이어에서 나온 필드 또는 표현식<br>***Name*** **(`name`) [string]**<br>산출 레이어의 필드명입니다. 기본적으로 입력 필드명을 그대로 유지합니다.<br>***Type*** **(`type`) [enumeration]**<br>Data type of the output field. Available types may not be compatible with the output layer provider. One of:<br>**주의**<br>For certain field types, e.g. lists, an extra `sub_type` parameter helps refine the specific type of the data. It is automatically set in the GUI but may be needed if you're running the algorithm in Python or from the command line.<br>• 1 —불 (Boolean)<br>• 2 —정수형 (32 비트)<br>• 4 —정수형 (64 비트)<br>• 6 —배정밀도 부동소수점 실수형 (double)<br>• 9 —정수형 목록<br>• 9 —정수형 (64 비트) 목록<br>• 9 —배정밀도 부동소수점 실수형 (double) 목록<br>• 110 —텍스트 (문자열)<br>• 11 —문자열 목록<br>• 12 —바이너리 객체 (BLOB)<br>• 14 —날짜<br>• 15 —시간<br>• 16 —날짜 & 시간<br>***Sub-type*** **(`sub_type`) [enumeration]** |

표 24.833 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Refactored** | `OUTPUT` | [same as input]<br>기본값: `[Create temporary layer]` | Specification of the output layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Refactored** | `OUTPUT` | [same as input] | 재구성된 필드를 가진 산출 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:refactorfields`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 필드 재명명

벡터 레이어의 기존 필드를 재명명합니다.

원본 레이어는 수정하지 않습니다. 재명명된 필드를 담고 있는 속성 테이블을 가진 새 레이어를 생성합니다.

**더 보기**

필드 재구성하기

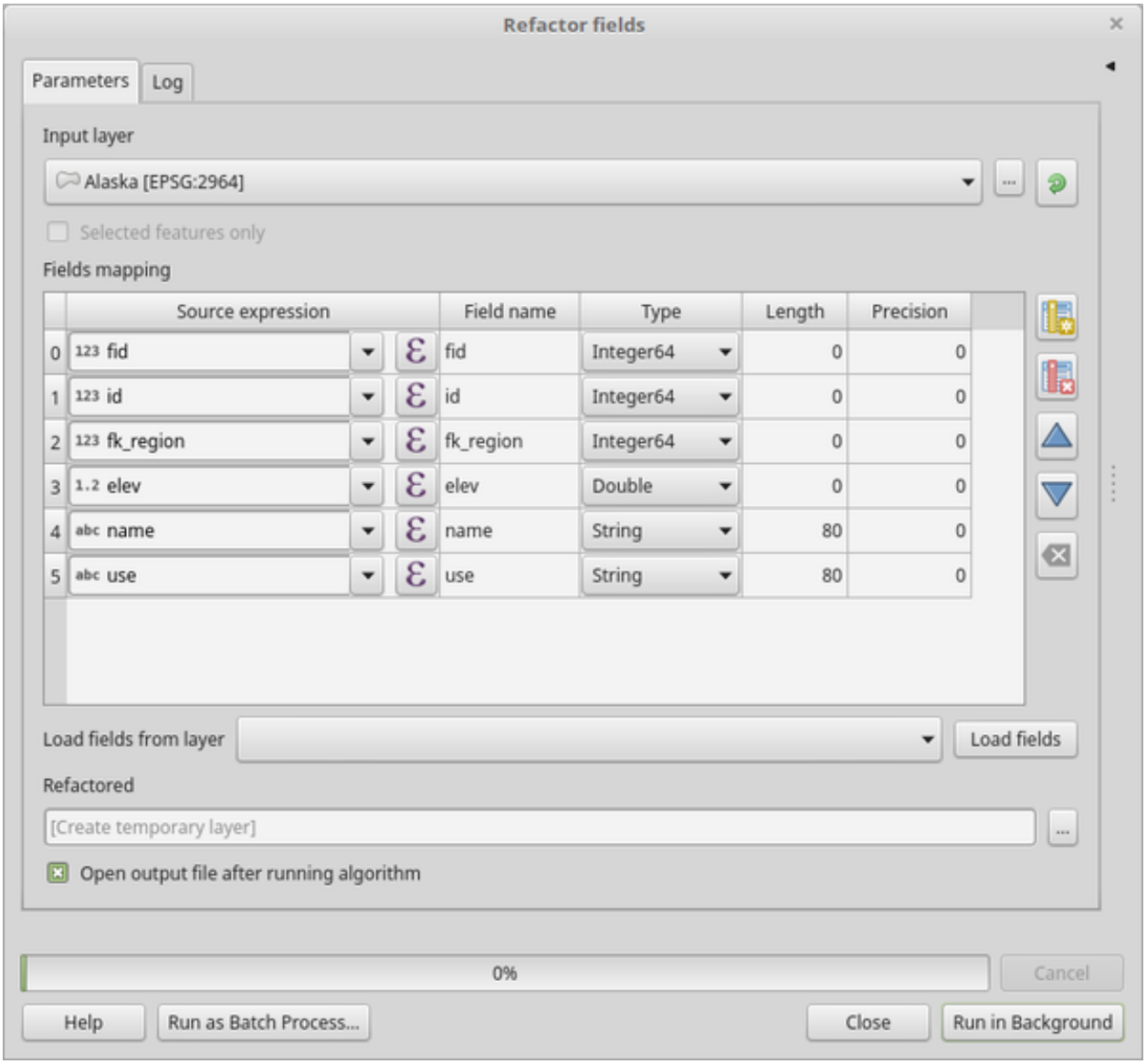



그림 24.165: 필드 재구성 대화창

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: any] | 입력 벡터 레이어 |
| **Field to rename** | `FIELD` | [tablefield: any] | 변경할 필드 |
| **New field name** | `NEW_NAME` | [string] | 새 필드명 |
| **Renamed** | `OUTPUT` | [same as input]<br>기본값: `[Create temporary layer]` | Specification of the output layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Renamed** | `OUTPUT` | [same as input] | 재명명된 필드를 가진 산출 레이어 |

### 파이썬 코드

**Algorithm ID**: `native:renametablefield`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 필드 유지

벡터 레이어를 받아 선택한 필드만을 유지한 새 벡터 레이어를 생성합니다. 다른 모든 필드는 삭제될 것입니다.

**더 보기**

필드 (들) 삭제하기

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: any] | 입력 벡터 레이어 |
| **Fields to retain** | `FIELDS` | [tablefield: any] [list] | 레이어에 유지할 필드들의 목록 |
| **Retained fields** | `OUTPUT` | [same as input]<br>기본값: `[Create temporary layer]` | Specification of the output layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Retained fields** | `OUTPUT` | [same as input] | 유지된 필드를 가진 산출 레이어 |

### 파이썬 코드

**알고리즘 ID**: `native:retainfields`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 텍스트를 부정소수점 실수형으로

벡터 레이어의 지정한 속성 유형을 수정해서, 숫자열을 담고 있는 텍스트 속성을 숫자 속성으로 (예: '1'을 `1.0` 으로) 변환합니다.

이 알고리즘은 소스 레이어를 수정하지 않고 새 벡터 레이어를 생성합니다.

변환 작업이 불가능한 경우 선택한 열이 `NULL` 값을 가질 것입니다.

#### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: any] | 입력 벡터 레이어 |
| **Text attribute to convert to float** | `FIELD` | [tablefield: string] | 부동소수점 필드로 변환될 입력 레이어의 문자열 필드 |
| **Float from text** | `OUTPUT` | [same as input]<br>기본값: `[Create Temporary Layer]` | Specify the output layer. *One of*:<br>• 임시 레이어 생성 (`TEMPORARY_OUTPUT`)<br>• 파일로 저장…<br>• GeoPackage 로 저장…<br>• 데이터베이스 테이블에 저장…<br>• 레이어에 추가…<br>이 파라미터에서 파일 인코딩도 변경할 수 있습니다. |

#### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Float from text** | `OUTPUT` | [same as input] | 문자열 필드를 부동소수점 필드로 변환한 산출 벡터 레이어 |

#### 파이썬 코드

**알고리즘 ID**: `qgis:texttofloat`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 24.1.30 벡터 타일

#### 벡터 타일 다운로드

입력 벡터 타일 레이어의 벡터 타일을 다운로드해서 로컬 벡터 타일 파일로 저장합니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector tiles] | 일부 타일을 추출할 벡터 타일 레이어 |
| **Extent** | `EXTENT` | [extent] | 다운로드한 영역의 공간 범위를 정의합니다. 내부적으로 타일 크기의 배수로 확장될 것입니다.<br>다음과 같은 메소드를 사용할 수 있습니다:<br>• 레이어로부터 계산…: 현재 프로젝트에 불러온 레이어의 범위를 사용합니다.<br>• 레이어 맵으로부터 계산…: 활성 프로젝트의조판 맵 항목 범위를 사용합니다.<br>• 북마크로부터 계산…: 저장되어 있는북마크 의 범위를 사용합니다.<br>• 맵 캔버스 범위 사용<br>• 캔버스에 그리기: 연산에 넣을 범위를 정하는 직사각형을 클릭 & 드래그합니다.<br>• 좌표를 `xmin, xmax, ymin, ymax` 로 입력 |
| **Maximum zoom level to download** | `MAX_ZOOM` | [numeric: integer]<br>기본값: 10 | 얼마나 확장해서 타일로부터 데이터를 가져올지 정의합니다. |
| **Tile limit** | `TILE_LIMIT` | [numeric: integer]<br>기본값: 100 | 확대/축소 수준 및 범위를 고려한 다운로드할 최대 타일 개수입니다. |
| **Output** | `OUTPUT` | [vector tiles]<br>기본값: [Save to temporary file] | Specification of the output vector tile file. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output** | `OUTPUT` | [vector tiles] | 다운로드한 타일들을 저장하고 있는 로컬 벡터 타일 파일입니다. |

### 파이썬 코드

**알고리즘 ID**: `native:downloadvectortiles`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 벡터 타일 (MBTiles) 작성

하나 이상의 벡터 파일을 빠른 맵 렌더링과 소용량 데이터에 최적화된 데이터 포맷인 벡터 타일로 내보냅니다.

MBTiles 는 즉각적인 활용 및 전송을 위해 SQLite 데이터베이스에 타일화된 맵 데이터를 저장하기 위한 사양입니다. MBTiles 파일은 타일셋 (tileset) 으로 알려져 있습니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layers** | `INPUT` | [vector: any] [list] | 벡터 타일을 생성하기 위해 결합될 레이어들의 목록 |
| **Minimum zoom level** | `MIN_ZOOM` | [numeric: integer] 기본값: 0 | 타일셋이 데이터를 제공하기 위한 최저 확대/축소 수준. 0 에서 24 사이로 설정합니다. |
| **Maximum zoom level** | `MAX_ZOOM` | [numeric: integer] 기본값: 3 | 타일셋이 데이터를 제공하기 위한 최고 확대/축소 수준. 0 에서 24 사이로 설정합니다. |
| **Extent** 선택적 | `EXTENT` | [extent] 기본값: 설정하지 않음 | 렌더링된 맵 영역의 최대 범위. 경계는 모든 확대/축소 수준이 커버하는 영역을 정의해야만 합니다. |
| **Metadata: Name** 선택적 | `META_NAME` | [string] | 타일셋의 이름 |
| **Metadata: Description** 선택적 | `META_DESCRIPTIC` | [string] | 타일셋 내용의 설명 |
| **Metadata: Attribution** 선택적 | `META_ATTRIBUTIC` | [string] | 데이터 소스 그리고/또는 맵 용 스타일을 설명하는 속성 문자열 |
| **Metadata: Version** 선택적 | `META_VERSION` | [string] | 타일셋의 버전. MBTiles 사양이 아니라 타일셋 자체의 리비전 (revision) 을 참조하는 버전입니다. |
| **Metadata: Type** 선택적 | `META_TYPE` | [string] | 타일셋의 유형. `overlay` 또는 `baselayer` 가운데 하나를 값으로 쓸 수 있습니다. |
| **Metadata: Center** 선택적 | `META_CENTER` | [string] | 맵의 기본 뷰의 중심. (쉼표로 분리된 숫자 문자열: 경도, 위도, 확대/축소 수준) 예: `-122.1906,37.7599,11` |

다음 페이지에 계속

표 24.843 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Destination MBTiles** | `OUTPUT` | [vector tiles] 기본값: [Save to temporary file] | Specification of the output MBTiles file. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Destination MBTiles** | `OUTPUT` | [file] | 산출 벡터 타일인 `.mbtiles` 파일입니다. |

### 파이썬 코드

**알고리즘 ID**: `native:writevectortiles_mbtiles`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 벡터 타일 (XYZ) 작성

하나 이상의 벡터 파일을 빠른 맵 렌더링과 소용량 데이터에 최적화된 데이터 포맷인 벡터 타일로 내보냅니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **File template** | `XYZ_TEMPLATE` | [string] 기본값: '{z}/{x}/{y}.pbf' | 벡터 타일 URL 을 생성하기 위한 템플릿 |
| **Input layers** | `INPUT` | [vector: any] [list] | 벡터 타일을 생성하기 위해 결합될 레이어들의 목록 |
| **Minimum zoom level** | `MIN_ZOOM` | [numeric: integer] 기본값: 0 | 타일셋이 데이터를 제공하기 위한 최저 확대/축소 수준. 0 에서 24 사이로 설정합니다. |
| **Maximum zoom level** | `MAX_ZOOM` | [numeric: integer] 기본값: 3 | 타일셋이 데이터를 제공하기 위한 최고 확대/축소 수준. 0 에서 24 사이로 설정합니다. |

다음 페이지에 계속

표 24.845 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Extent**<br>선택적 | `EXTENT` | [extent]<br>기본값: 설정하지 않음 | 렌더링된 맵 영역의 최대 범위. 경계는 모든 확대/축소 수준이 커버하는 영역을 정의해야만 합니다. |
| **Output directory** | `OUTPUT_DIRECTORY` | [folder]<br>기본값: [Save to temporary folder] | Specification of the output vector tiles folder.<br>*One of*:<br>• 임시 디렉터리에 저장<br>• 디렉터리에 저장 |

산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output directory** | `OUTPUT_DIRECTORY` | [folder] | 확대/축소 수준에 대응하는 하위 폴더들에 저장된 벡터 타일 (`.pbf`) 파일들의 서로 다른 하위 집합들을 담고 있는 폴더입니다. |

파이썬 코드

**알고리즘 ID**: `native:writevectortiles_xyz`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 24.2 GDAL 알고리즘 제공자

GDAL (Geospatial Data Abstraction Library) 은 래스터 및 벡터 지리공간 데이터 포맷을 위한 번역 (translator) 라이브러리입니다. 공간 처리 프레임워크의 알고리즘들은 GDAL 래스터 프로그램 및 GDAL 벡터 프로그램 에서 파생되었습니다.

### 24.2.1 래스터 분석

#### 향

GDAL 이 지원하는 모든 표고 래스터로부터 경사 방향 (aspect) 맵을 생성합니다. 경사 방향이란 경사면이 마주보는 나침반 방향을 말합니다. 각 픽셀은 북쪽으로부터 측정한, 방위각을 나타내는 도 단위 0°-360° 값을 가지게 됩니다. 북반구에서 경사면의 북측은 주로 음영이 지고 (0°-90° 사이의 작은 방위각), 남측은 태양 복사를 더 많이 받습니다 (180°-270° 사이의 큰 방위각).

This algorithm is derived from the GDAL DEM utility.

**기본 메뉴**: *Raster → Analysis*

#### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [raster] | 입력 표고 래스터 레이어 |
| **Band number** | `BAND` | [raster band]<br>기본값: 1 | 표고로 사용할 밴드의 번호 |
| **Return trigonometric angle instead of azimuth** | `TRIG_ANGLE` | [boolean]<br>기본값: False | 서로 다른 범주를 산출하는 삼각법에 의한 각도를 활성화합니다: 0° (동), 90° (북), 180° (서), 270° (남) |
| **Return 0 for flat instead of -9999** | `ZERO_FLAT` | [boolean]<br>기본값: False | 이 옵션을 활성화하면 평야 지대에 -9999 값 대신 0 값을 삽입할 것입니다. |
| **Compute edges** | `COMPUTE_EDGES` | [boolean]<br>기본값: False | 표고 래스터에서 경계를 생성합니다. |
| **Use Zevenbergen&Thorne formula instead of the Horn's one** | `ZEVENBERGEN` | [boolean]<br>기본값: False | 매끄러운 풍경을 위해 체벤베르겐 & 손 (Zevenbergen&Thorne) 공식을 활성화합니다. |
| **Aspect** | `OUTPUT` | [raster]<br>기본값: `[Save to temporary file]` | Output raster layer. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

#### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Additional creation options**<br>선택적 | `CREATION_OPTIONS`<br>(for QGIS <= 3.42, this was `OPTIONS`) | [string]<br>기본값: '' | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL 드라이버 옵션* 참조) 배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (\|) 로 복수의 옵션들을 분리하십시오. |

다음 페이지에 계속

표 24.848 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Additional command-line parameters** 선택적 | EXTRA | [string] 기본값: 설정하지 않음 | 부가 GDAL 명령 줄 옵션을 추가합니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Aspect** | OUTPUT | [raster] | 도 단위 각도 값을 가진 산출 래스터 |

### 파이썬 코드

**알고리즘 ID**: `gdal:aspect`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 색상 기복

GDAL 이 지원하는 모든 표고 레이어로부터 색상 기복 (color relief) 맵을 생성합니다. 색상 기복은 특히 표고를 묘사하기 위해 사용될 수 있습니다. 이 알고리즘은 표고 및 텍스트 기반 색상 환경 설정 파일로부터 계산한 값을 가진 밴드 4 개의 래스터를 산출합니다. 지정한 표고 값 사이의 색상은 기본적으로 부드럽게 혼합되어, 보기 좋게 색상화된 표고 래스터를 산출합니다.

This algorithm is derived from the GDAL DEM utility.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | INPUT | [raster] | 입력 표고 래스터 레이어 |
| **Band number** | BAND | [raster band] 기본값: 1 | 표고로 사용할 밴드의 번호 |
| **Compute edges** | COMPUTE_EDGES | [boolean] 기본값: False | 표고 래스터에서 경계를 생성합니다. |
| **Color configuration file** | COLOR_TABLE | [file] | 텍스트 기반 색상 환경 설정 파일 |

다음 페이지에 계속

표 24.850 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Matching mode** | MATCH_MODE | [enumeration]<br>기본값: 2 | 다음 가운데 하나로 지정할 수 있습니다:<br>• 0 —색상을 엄격하게 맞춤<br>• 1 —최근접 RGBA 4 배수 (quadruple) 사용<br>• 2 —부드럽게 혼합된 색상 사용 |
| **Additional creation options**<br>선택적 | CREATION_OPTIONS (for QGIS <= 3.42, this was OPTIONS) | [string]<br>기본값: '' | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL 드라이버 옵션* 참조) 배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (\|) 로 복수의 옵션들을 분리하십시오. |
| **Additional command-line parameters**<br>선택적 | EXTRA | [string]<br>기본값: 설정하지 않음 | 부가 GDAL 명령 줄 옵션을 추가합니다. |
| **Color relief** | OUTPUT | [raster]<br>기본값: [Save to temporary file] | Output raster layer. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Color relief** | OUTPUT | [raster] | 밴드 4 개의 산출 래스터 |

### 파이썬 코드

**알고리즘 ID**: gdal:colorrelief

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### NODATA 채우기

데이터 값이 없는 래스터 영역을 경계로부터 보간해서 채웁니다. NODATA 영역을 둘러싸고 있는 픽셀들의 값을 역거리 가중치법 (inverse distance weighting) 으로 보간해서 NODATA 영역의 값을 계산합니다. 보간 작업 완료 후 그 결과를 평탄화합니다. GDAL 이 지원하는 모든 래스터 레이어를 입력할 수 있습니다. 이 알고리즘은 일반적으로 상당히 연속적으로 변화하는 (예를 들어 표고 모델 같은) 래스터에서 빠진 영역을 보간하는 데 적합합니다. 좀 더 불규칙적으로 변화하는 (항공사진 같은) 이미지의 작은 구멍이나 금이 간 부분을 채우는 데에도 적합합니다. 다만 희박하게 분포한 포인트 데이터로부터 래스터를 보간하는 데에는 이 알고리즘을 사용하지 않는 편이 좋습니다.

This algorithm is derived from the GDAL fillnodata utility.

**기본 메뉴**: *Raster → Analysis*

#### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| 라벨 | 이름 | 유형 | 설명 |
| **Input layer** | `INPUT` | [raster] | 입력 래스터 레이어 |
| **Band number** | `BAND` | [raster band]<br>기본값: 1 | 작업할 밴드. NODATA 값은 0 값으로 표현되어야만 합니다. |
| **Maximum distance (in pixels) to search out for values to interpolate** | `DISTANCE` | [numeric: integer]<br>기본값: 10 | 모든 방향에서 보간할 값을 찾기 위해 검색할 픽셀들의 개수 |
| **Number of smoothing iterations to run after the interpolation** | `ITERATIONS` | [numeric: integer]<br>기본값: 0 | 보간 작업의 결과를 평탄화하기 위해 (0 번 이상) 실행할 3x3 필터링의 횟수 |
| **Validity mask** | `MASK_LAYER` | [raster] | 채워야 할 영역을 정의하는 래스터 레이어 |
| **Filled** | `OUTPUT` | [raster]<br>기본값: `[Save to temporary file]` | Specification of the output raster layer. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

#### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Additional creation options**<br>선택적 | `CREATION_OPTIONS` (for QGIS <= 3.42, this was `OPTIONS`) | [string]<br>기본값: ' ' | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL 드라이버 옵션* 참조)<br>배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (\|) 로 복수의 옵션들을 분리하십시오. |

다음 페이지에 계속

표 24.853 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Additional command-line parameters** 선택적 | EXTRA | [string] 기본값: 설정하지 않음 | 부가 GDAL 명령 줄 옵션을 추가합니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Filled** | OUTPUT | [raster] | 산출 래스터 |

### 파이썬 코드

**알고리즘 ID**: `gdal:fillnodata`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 그리드 (데이터 메트릭)

지정한 창과 산출 그리드 도형을 사용해서 몇몇 데이터 메트릭 (data metrics) 을 계산합니다.

This algorithm is derived from the GDAL grid utility.

**기본 메뉴**: *Raster ▸ Analysis*

**더 보기**

GDAL grid tutorial

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Point layer** | INPUT | [vector: point] | 입력 포인트 벡터 레이어 |

다음 페이지에 계속

표 24.855 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Data metric to use** | METRIC | [enumeration] 기본값: 0 | 다음 가운데 하나로 지정할 수 있습니다:<br>• 0 —최소값: 그리드 노드 검색 타원에서 발견된 최소값<br>• 1 —최대값: 그리드 노드 검색 타원에서 발견된 최대값<br>• 2 —범위: 그리드 노드 검색 타원에서 발견된 최소값 및 최대값의 차<br>• 3 —개수: 그리드 노드 검색 타원에서 발견된 데이터 포인트의 개수<br>• 4 —평균 거리: 그리드 노드 (검색 타원의 중심) 와 그리드 노드 검색 타원에서 발견된 모든 데이터 포인트 사이의 평균 거리<br>• 5 —포인트 사이의 평균 거리: 그리드 노드 검색 타원에서 발견된 데이터 포인트들 사이의 평균 거리. 타원 내부에 있는 각 포인트쌍 사이의 거리를 계산하고 모든 거리의 평균을 그리드 노드 값으로 설정합니다. |
| **The first radius of search ellipse** | RADIUS_1 | [numeric: double] 기본값: 0.0 | 검색 타원의 첫 번째 (기울기 각도가 0 인 경우 X 축) 반경 |
| **The second radius of search ellipse** | RADIUS_2 | [numeric: double] 기본값: 0.0 | 검색 타원의 두 번째 (기울기 각도가 0 인 경우 Y 축) 반경 |
| **Angle of search ellipse rotation in degrees (counter clockwise)** | ANGLE | [numeric: double] 기본값: 0.0 | 타원 기울기의 도 단위 각도. 타원은 반시계 방향으로 기울어집니다. |
| **Minimum number of data points to use** | MIN_POINTS | [numeric: double] 기본값: 0.0 | 평균을 낼 데이터 포인트의 최소 개수. 이 값보다 적은 포인트가 발견된 그리드 노드는 비어 있다고 간주되어 NODATA 마커로 채워질 것입니다. |
| **NoData** | NODATA | [numeric: double] 기본값: 0.0 | 비어 있는 포인트를 채울 NODATA 마커 |
| **Interpolated (data metrics)** | OUTPUT | [raster] 기본값: [Save to temporary file] | Specify the output raster layer with interpolated values. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Z value from field** 선택적 | Z_FIELD | [tablefield: numeric] | 보간 작업용 필드 |

다음 페이지에 계속

표 24.856 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Additional creation options**<br>선택적 | CRE-ATION_OPTIONS (for QGIS <= 3.42, this was OPTIONS) | [string]<br>기본값: ' | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL 드라이버 옵션* 참조)<br>배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (\|) 로 복수의 옵션들을 분리하십시오. |
| **Additional command-line parameters**<br>선택적 | EXTRA | [string]<br>기본값: 설정하지 않음 | 부가 GDAL 명령 줄 옵션을 추가합니다. |
| **Output data type** | DATA_TYPE | [enumeration]<br>기본값: 5 | 산출 래스터 파일의 데이터 유형을 정의합니다. 다음 가운데 하나로 지정할 수 있습니다:<br>• 0 —Byte (8 비트 부호 없는 정수 (quint8))<br>• 1 —Int16 (16 비트 부호 있는 정수 (qint16))<br>• 2 —UInt16 (16 비트 부호 없는 정수 (quint16))<br>• 3 —UInt32 (32 비트 부호 없는 정수 (quint32))<br>• 4 —Int32 (32 비트 부호 있는 정수 (qint32))<br>• 5 —Float32 (32 비트 부동소수점형 (float))<br>• 6 —Float64 (64 비트 부동소수점형 (double))<br>• 7 —CInt16 (Complex Int16)<br>• 8 —CInt32 (Complex Int32)<br>• 9 —CFloat32 (Complex Float32)<br>• 10 —CFloat64 (Complex Float64)<br>• 11 —Int8 (8 비트 부호 있는 정수 (qint8))<br>QGIS 와 함께 빌드된 GDAL 버전에 따라 사용할 수 있는 옵션이 달라집니다. (*Help* *About* 메뉴 참조) |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Interpolated (data metrics)** | OUTPUT | [raster] | 보간된 값을 가진 산출 래스터 |

### 파이썬 코드

**알고리즘 ID**: `gdal:griddatametrics`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 그리드 (최근접 이웃을 검색하는 역거리 가중치법)

최근접 이웃 메소드와 결합된 역거리 제곱 (Inverse Distance to a Power) 그리드 작업을 계산합니다. 최대 개수의 데이터 포인트를 사용해야 하는 경우 이상적입니다.

This algorithm is derived from the GDAL grid utility.

**더 보기**

GDAL grid tutorial

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Point layer** | `INPUT` | [vector: point] | 입력 포인트 벡터 레이어 |
| **Weighting power** | `POWER` | [numeric: double]<br>기본값: 2.0 | 가중치 제곱지수 |
| **Smoothing** | `SMOOTHING` | [numeric: double]<br>기본값: 0.0 | 평탄화 파라미터 |
| **The radius of the search circle** | `RADIUS` | [numeric: double]<br>기본값: 1.0 | 검색 원의 반경 |
| **Maximum number of data points to use** | `MAX_POINTS` | [numeric: integer]<br>기본값: 12 | 이 값을 초과하는 개수의 포인트를 검색하지 않습니다. |
| **Minimum number of data points to use** | `MIN_POINTS` | [numeric: integer]<br>기본값: 0 | 평균을 낼 데이터 포인트의 최소 개수. 이 값보다 적은 포인트가 발견된 그리드 노드는 비어 있다고 간주되어 NODATA 마커로 채워질 것입니다. |
| **NoData** | `NODATA` | [numeric: double]<br>기본값: 0.0 | 비어 있는 포인트를 채울 NODATA 마커 |
| **Interpolated (IDW with NN search)** | `OUTPUT` | [raster]<br>기본값: `[Save to temporary file]` | Specify the output raster layer with interpolated values. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Z value from field**<br>선택적 | `Z_FIELD` | [tablefield: numeric] | 보간 작업용 필드 |
| **Additional creation options**<br>선택적 | `CRE-ATION_OPTIONS` (for QGIS <= 3.42, this was `OPTIONS`) | [string]<br>기본값: ' | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL 드라이버 옵션* 참조)<br>배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (\|) 로 복수의 옵션들을 분리하십시오. |
| **Additional command-line parameters**<br>선택적 | `EXTRA` | [string]<br>기본값: 설정하지 않음 | 부가 GDAL 명령 줄 옵션을 추가합니다. |
| **Output data type** | `DATA_TYPE` | [enumeration]<br>기본값: 5 | 산출 래스터 파일의 데이터 유형을 정의합니다. 다음 가운데 하나로 지정할 수 있습니다:<br>• 0 —Byte (8 비트 부호 없는 정수 (quint8))<br>• 1 —Int16 (16 비트 부호 있는 정수 (qint16))<br>• 2 —UInt16 (16 비트 부호 없는 정수 (quint16))<br>• 3 —UInt32 (32 비트 부호 없는 정수 (quint32))<br>• 4 —Int32 (32 비트 부호 있는 정수 (qint32))<br>• 5 —Float32 (32 비트 부동소수점형 (float))<br>• 6 —Float64 (64 비트 부동소수점형 (double))<br>• 7 —CInt16 (Complex Int16)<br>• 8 —CInt32 (Complex Int32)<br>• 9 —CFloat32 (Complex Float32)<br>• 10 —CFloat64 (Complex Float64)<br>• 11 —Int8 (8 비트 부호 있는 정수 (qint8))<br>QGIS 와 함께 빌드된 GDAL 버전에 따라 사용할 수 있는 옵션이 달라집니다. (*Help → About* 메뉴 참조) |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Interpolated (IDW with NN search)** | `OUTPUT` | [raster] | 보간된 값을 가진 산출 래스터 |

### 파이썬 코드

**알고리즘 ID**: `gdal:gridinversedistancenearestneighbor`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 그리드 (역거리 제곱)

역거리 제곱 그리드 작업 방법이란 가중치를 부여한 평균 보간기 (interpolator) 를 말합니다.

모든 데이터 포인트 및 산출 그리드 도형의 좌표를 포함하는 산포 (scatter) 된 데이터 값을 담은 입력 배열을 지정해야 합니다. 이 알고리즘 함수는 산출 그리드에 지정한 위치에 대한 보간값을 계산할 것입니다.

This algorithm is derived from the GDAL grid utility.

**기본 메뉴**: *Raster → Analysis*

**더 보기**

GDAL grid tutorial

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Point layer** | `INPUT` | [vector: point] | 입력 포인트 벡터 레이어 |
| **Weighting power** | `POWER` | [numeric: double]<br>기본값: 2.0 | 가중치 제곱지수 |
| **Smothing** | `SMOOTHING` | [numeric: double]<br>기본값: 0.0 | 평탄화 파라미터 |
| **The first radius of search ellipse** | `RADIUS_1` | [numeric: double]<br>기본값: 0.0 | 검색 타원의 첫 번째 (기울기 각도가 0 인 경우 X 축) 반경 |
| **The second radius of search ellipse** | `RADIUS_2` | [numeric: double]<br>기본값: 0.0 | 검색 타원의 두 번째 (기울기 각도가 0 인 경우 Y 축) 반경 |

다음 페이지에 계속

표 24.861 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Angle of search ellipse rotation in degrees (counter clockwise)** | ANGLE | [numeric: double]<br>기본값: 0.0 | 타원 기울기의 도 단위 각도. 타원은 반시계 방향으로 기울어집니다. |
| **Maximum number of data points to use** | MAX_POINTS | [numeric: integer]<br>기본값: 0 | 이 값을 초과하는 개수의 포인트를 검색하지 않습니다. |
| **Minimum number of data points to use** | MIN_POINTS | [numeric: integer]<br>기본값: 0 | 평균을 낼 데이터 포인트의 최소 개수. 이 값보다 적은 포인트가 발견된 그리드 노드는 비어 있다고 간주되어 NODATA 마커로 채워질 것입니다. |
| **NoData** | NODATA | [numeric: double]<br>기본값: 0.0 | 비어 있는 포인트를 채울 NODATA 마커 |
| **Interpolated (IDW)** | OUTPUT | [raster]<br>기본값: [Save to temporary file] | Specify the output raster layer with interpolated values. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Z value from field**<br>선택적 | Z_FIELD | [tablefield: numeric] | 보간 작업용 필드 |
| **Additional creation options**<br>선택적 | CREATION_OPTIONS (for QGIS <= 3.42, this was OPTIONS) | [string]<br>기본값: ‘’ | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL* 드라이버 옵션 참조)<br>배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (\|) 로 복수의 옵션들을 분리하십시오. |
| **Additional command-line parameters**<br>선택적 | EXTRA | [string]<br>기본값: 설정하지 않음 | 부가 GDAL 명령 줄 옵션을 추가합니다. |

다음 페이지에 계속

표 24.862 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output data type** | `DATA_TYPE` | [enumeration] 기본값: 5 | 산출 래스터 파일의 데이터 유형을 정의합니다. 다음 가운데 하나로 지정할 수 있습니다:<br>• 0 —Byte (8 비트 부호 없는 정수 (quint8))<br>• 1 —Int16 (16 비트 부호 있는 정수 (qint16))<br>• 2 —UInt16 (16 비트 부호 없는 정수 (quint16))<br>• 3 —UInt32 (32 비트 부호 없는 정수 (quint32))<br>• 4 —Int32 (32 비트 부호 있는 정수 (qint32))<br>• 5 —Float32 (32 비트 부동소수점형 (float))<br>• 6 —Float64 (64 비트 부동소수점형 (double))<br>• 7 —CInt16 (Complex Int16)<br>• 8 —CInt32 (Complex Int32)<br>• 9 —CFloat32 (Complex Float32)<br>• 10 —CFloat64 (Complex Float64)<br>• 11 —Int8 (8 비트 부호 있는 정수 (qint8))<br>QGIS 와 함께 빌드된 GDAL 버전에 따라 사용할 수 있는 옵션이 달라집니다. (*Help* → *About* 메뉴 참조) |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Interpolated (IDW)** | `OUTPUT` | [raster] | 보간된 값을 가진 산출 래스터 |

### 파이썬 코드

**알고리즘 ID**: `gdal:gridinversedistance`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 그리드 (선형)

선형 (linear) 방법은 점구름 (point cloud) 의 델로네 삼각분할 (Delaunay triangulation) 을 계산해서 선형 보간을 수행합니다. 포인트가 삼각분할의 어느 삼각형 안에 있는지 찾은 다음, 해당 삼각형 내부의 무게중심 좌표로부터 선형 보간을 수행합니다. 포인트가 어떤 삼각형에도 들어가지 않는 경우, 이 알고리즘은 검색 반경에 따라 최근접 포인트의 값 또는 NODATA 값을 사용할 것입니다.

This algorithm is derived from the GDAL grid utility.

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Point layer** | `INPUT` | [vector: point] | 입력 포인트 벡터 레이어 |
| **Search distance** | `RADIUS` | [numeric: double]<br>기본값: -1.0 | 보간할 포인트가 델로네 삼각분할의 삼각형 안에 들어가지 않는 경우, 이 최대 거리를 사용해서 최근접 이웃을 찾거나 또는 최대 거리 안에 최근접 이웃이 없다면 NODATA 를 사용합니다. 이 파라미터를 `-1` 로 설정하면, 검색 반경이 무한대가 됩니다. `0` 으로 설정하면 NODATA 값을 사용할 것입니다. |
| **NoData** | `NODATA` | [numeric: double]<br>기본값: 0.0 | 비어 있는 포인트를 채울 NODATA 마커 |
| **Interpolated (Linear)** | `OUTPUT` | [raster]<br>기본값: `[Save to temporary file]` | Specify the output raster layer with interpolated values. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Z value from field**<br>선택적 | `Z_FIELD` | [tablefield: numeric] | 보간 작업용 필드 |
| **Additional creation options**<br>선택적 | `CREATION_OPTIONS`<br>(for QGIS <= 3.42, this was `OPTIONS`) | [string]<br>기본값: '' | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL 드라이버 옵션* 참조)<br>배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (`\|`) 로 복수의 옵션들을 분리하십시오. |

다음 페이지에 계속

표 24.865 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Additional command-line parameters** 선택적 | EXTRA | [string] 기본값: 설정하지 않음 | 부가 GDAL 명령 줄 옵션을 추가합니다. |
| **Output data type** | DATA_TYPE | [enumeration] 기본값: 5 | 산출 래스터 파일의 데이터 유형을 정의합니다. 다음 가운데 하나로 지정할 수 있습니다:<br>• 0 —Byte (8 비트 부호 없는 정수 (quint8))<br>• 1 —Int16 (16 비트 부호 있는 정수 (qint16))<br>• 2 —UInt16 (16 비트 부호 없는 정수 (quint16))<br>• 3 —UInt32 (32 비트 부호 없는 정수 (quint32))<br>• 4 —Int32 (32 비트 부호 있는 정수 (qint32))<br>• 5 —Float32 (32 비트 부동소수점형 (float))<br>• 6 —Float64 (64 비트 부동소수점형 (double))<br>• 7 —CInt16 (Complex Int16)<br>• 8 —CInt32 (Complex Int32)<br>• 9 —CFloat32 (Complex Float32)<br>• 10 —CFloat64 (Complex Float64)<br>• 11 —Int8 (8 비트 부호 있는 정수 (qint8))<br>QGIS 와 함께 빌드된 GDAL 버전에 따라 사용할 수 있는 옵션이 달라집니다. (*Help* ⍰ *About* 메뉴 참조) |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Interpolated (Linear)** | OUTPUT | [raster] | 보간된 값을 가진 산출 래스터 |

### 파이썬 코드

**알고리즘 ID**: `gdal:gridlinear`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 그리드 (이동 평균)

이동 평균 (Moving Average) 은 간단한 데이터 평균 알고리즘입니다. 이 알고리즘은 타원 형태의 이동 창을 사용해서 값을 검색하고 창 안에 들어오는 모든 데이터 포인트의 평균을 계산합니다. 지정한 각도로 검색 타원을 기울일 수 있고, 타원의 중심은 그리드 노드에 위치합니다. 또 평균을 계산하기 위한 데이터 포인트의 최소 개수도 설정할 수 있어서, 창 안에 충분한 포인트가 없을 경우 해당 그리드 노드가 비어 있다고 간주하고 지정한 NODATA 값으로 채울 것입니다.

This algorithm is derived from the GDAL grid utility.

**기본 메뉴**: *Raster ▸ Analysis*

**더 보기**

GDAL grid tutorial

#### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Point layer** | `INPUT` | [vector: point] | 입력 포인트 벡터 레이어 |
| **The first radius of search ellipse** | `RADIUS_1` | [numeric: double]<br>기본값: 0.0 | 검색 타원의 첫 번째 (기울기 각도가 0 인 경우 X 축) 반경 |
| **The second radius of search ellipse** | `RADIUS_2` | [numeric: double]<br>기본값: 0.0 | 검색 타원의 두 번째 (기울기 각도가 0 인 경우 Y 축) 반경 |
| **Angle of search ellipse rotation in degrees (counter clockwise)** | `ANGLE` | [numeric: double]<br>기본값: 0.0 | 타원 기울기의 도 단위 각도. 타원은 반시계 방향으로 기울어집니다. |
| **Minimum number of data points to use** | `MIN_POINTS` | [numeric: integer]<br>기본값: 0 | 평균을 낼 데이터 포인트의 최소 개수. 이 값보다 적은 포인트가 발견된 그리드 노드는 비어 있다고 간주되어 NODATA 마커로 채워질 것입니다. |
| **NoData** | `NODATA` | [numeric: double]<br>기본값: 0.0 | 비어 있는 포인트를 채울 NODATA 마커 |
| **Interpolated (moving average)** | `OUTPUT` | [raster]<br>기본값: `[Save to temporary file]` | Specify the output raster layer. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Z value from field**<br>선택적 | `Z_FIELD` | [tablefield: numeric] | 보간 작업용 필드 |
| **Additional creation options**<br>선택적 | `CRE-ATION_OPTIONS` (for QGIS <= 3.42, this was `OPTIONS`) | [string]<br>기본값: ' | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL 드라이버 옵션* 참조)<br>배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (\|) 로 복수의 옵션들을 분리하십시오. |
| **Additional command-line parameters**<br>선택적 | `EXTRA` | [string]<br>기본값: 설정하지 않음 | 부가 GDAL 명령 줄 옵션을 추가합니다. |
| **Output data type** | `DATA_TYPE` | [enumeration]<br>기본값: 5 | 산출 래스터 파일의 데이터 유형을 정의합니다. 다음 가운데 하나로 지정할 수 있습니다:<br>• 0 —Byte (8 비트 부호 없는 정수 (quint8))<br>• 1 —Int16 (16 비트 부호 있는 정수 (qint16))<br>• 2 —UInt16 (16 비트 부호 없는 정수 (quint16))<br>• 3 —UInt32 (32 비트 부호 없는 정수 (quint32))<br>• 4 —Int32 (32 비트 부호 있는 정수 (qint32))<br>• 5 —Float32 (32 비트 부동소수점형 (float))<br>• 6 —Float64 (64 비트 부동소수점형 (double))<br>• 7 —CInt16 (Complex Int16)<br>• 8 —CInt32 (Complex Int32)<br>• 9 —CFloat32 (Complex Float32)<br>• 10 —CFloat64 (Complex Float64)<br>• 11 —Int8 (8 비트 부호 있는 정수 (qint8))<br>QGIS 와 함께 빌드된 GDAL 버전에 따라 사용할 수 있는 옵션이 달라집니다. (*Help → About* 메뉴 참조) |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Interpolated (moving average)** | `OUTPUT` | [raster] | 보간된 값을 가진 산출 래스터 |

### 파이썬 코드

**알고리즘 ID**: `gdal:gridaverage`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 그리드 (최근접 이웃)

최근접 이웃 (Nearest Neighbor) 방법은 어떤 보간이나 평탄화도 수행하지 않습니다. 그냥 그리드 노드 검색 타원에서 발견된 최근접 포인트의 값을 받아서 이를 결과물로 반환할 뿐입니다. 어떤 포인트도 발견하지 못 한 경우, 지정한 NODATA 값을 반환할 것입니다.

This algorithm is derived from the GDAL grid utility.

**기본 메뉴**: *Raster → Analysis*

**더 보기**

GDAL grid tutorial

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Point layer** | `INPUT` | [vector: point] | 입력 포인트 벡터 레이어 |
| **The first radius of search ellipse** | `RADIUS_1` | [numeric: double]<br>기본값: 0.0 | 검색 타원의 첫 번째 (기울기 각도가 0 인 경우 X 축) 반경 |
| **The second radius of search ellipse** | `RADIUS_2` | [numeric: double]<br>기본값: 0.0 | 검색 타원의 두 번째 (기울기 각도가 0 인 경우 Y 축) 반경 |
| **Angle of search ellipse rotation in degrees (counter clockwise)** | `ANGLE` | [numeric: double]<br>기본값: 0.0 | 타원 기울기의 도 단위 각도. 타원은 반시계 방향으로 기울어집니다. |

다음 페이지에 계속

표 24.870 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **NoData** | `NODATA` | [numeric: double]<br>기본값: 0.0 | 비어 있는 포인트를 채울 NODATA 마커 |
| **Interpolated (Nearest neighbour)** | `OUTPUT` | [raster]<br>기본값: `[Save to temporary file]` | Specify the output raster layer with interpolated values. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Z value from field**<br>선택적 | `Z_FIELD` | [tablefield: numeric] | 보간 작업용 필드 |
| **Additional creation options**<br>선택적 | `CREATION_OPTIONS` (for QGIS <= 3.42, this was `OPTIONS`) | [string]<br>기본값: ' | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL 드라이버 옵션* 참조) 배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (\|) 로 복수의 옵션들을 분리하십시오. |
| **Additional command-line parameters**<br>선택적 | `EXTRA` | [string]<br>기본값: 설정하지 않음 | 부가 GDAL 명령 줄 옵션을 추가합니다. |

다음 페이지에 계속

표 24.871 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output data type** | DATA_TYPE | [enumeration] 기본값: 5 | 산출 래스터 파일의 데이터 유형을 정의합니다. 다음 가운데 하나로 지정할 수 있습니다:<br>• 0 —Byte (8 비트 부호 없는 정수 (quint8))<br>• 1 —Int16 (16 비트 부호 있는 정수 (qint16))<br>• 2 —UInt16 (16 비트 부호 없는 정수 (quint16))<br>• 3 —UInt32 (32 비트 부호 없는 정수 (quint32))<br>• 4 —Int32 (32 비트 부호 있는 정수 (qint32))<br>• 5 —Float32 (32 비트 부동소수점형 (float))<br>• 6 —Float64 (64 비트 부동소수점형 (double))<br>• 7 —CInt16 (Complex Int16)<br>• 8 —CInt32 (Complex Int32)<br>• 9 —CFloat32 (Complex Float32)<br>• 10 —CFloat64 (Complex Float64)<br>• 11 —Int8 (8 비트 부호 있는 정수 (qint8))<br>QGIS 와 함께 빌드된 GDAL 버전에 따라 사용할 수 있는 옵션이 달라집니다. (*Help* *About* 메뉴 참조) |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Interpolated (Nearest neighbour)** | OUTPUT | [raster] | 보간된 값을 가진 산출 래스터 |

### 파이썬 코드

**알고리즘 ID**: `gdal:gridnearestneighbor`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 음영기복

보기 좋게 그림자가 진 기복 (shaded relief) 효과를 준 래스터를 산출합니다. 이 알고리즘은 지형을 가시화하는 데 매우 유용합니다. 광원의 방위각 및 고도, 수직 강조 인자, 수직 및 수평 단위 사이의 차이를 처리하기 위한 척도 인자를 지정할 수 있는 옵션이 있습니다.

This algorithm is derived from the GDAL DEM utility .

기본 메뉴: *Raster → Analysis*

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [raster] | 입력 표고 래스터 레이어 |
| **Band number** | `BAND` | [raster band]<br>기본값: 1 | 표고 정보를 담고 있는 밴드 |
| **Z factor (vertical exaggeration)** | `Z_FACTOR` | [numeric: double]<br>기본값: 1.0 | 산출 표고 래스터의 높이를 강조하는 인자 |
| **Scale (ratio of vertical units to horizontal)** | `SCALE` | [numeric: double]<br>기본값: 1.0 | 수직 단위와 수평 단위의 비율 |
| **Azimuth of the light** | `AZIMUTH` | [numeric: double]<br>기본값: 315.0 | 표고 래스터를 비추는 광원의 방위각을 도 단위로 정의합니다. 래스터의 상단 (북)에서 비추는 경우 값은 0 이고, 동쪽에서 비추는 경우 값은 90 이고 등등. |
| **Altitude of the light** | `ALTITUDE` | [numeric: double]<br>기본값: 45.0 | 광원의 고도를 도 단위로 정의합니다. 표고 래스터 바로 위에서 비추는 경우 값은 90 이고, 표면에 평행하게 비추는 경우 값은 0 입니다. |
| **Compute edges** | `COMPUTE_EDGES` | [boolean]<br>기본값: False | 표고 래스터에서 경계를 생성합니다. |
| **Use Zevenbergen&Thorne formula (instead of the Horn's one)** | `ZEVENBERGEN` | [boolean]<br>기본값: False | 매끄러운 풍경을 위해 체벤베르겐 & 손 (Zevenbergen&Thorne) 공식을 활성화합니다. |
| **Combined shading** | `COMBINED` | [boolean]<br>기본값: False | |
| **Multidirectional shading** | `MULTIDIRECTIONAL` | [boolean]<br>기본값: False | |
| **Hillshade** | `OUTPUT` | [raster]<br>기본값: `[Save to temporary file]` | Specify the output raster layer with interpolated values. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Additional creation options**<br>선택적 | `CREATION_OPTIONS` (for QGIS <= 3.42, this was `OPTIONS`) | [string]<br>기본값: ' | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL 드라이버 옵션* 참조)<br>배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (\|) 로 복수의 옵션들을 분리하십시오. |
| **Additional command-line parameters**<br>선택적 | `EXTRA` | [string]<br>기본값: 설정하지 않음 | 부가 GDAL 명령 줄 옵션을 추가합니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Hillshade** | `OUTPUT` | [raster] | 보간된 값을 가진 산출 래스터 |

### 파이썬 코드

**알고리즘 ID**: `gdal:hillshade`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 근사 검은색

검은색/하얀색에 가까운 경계를 검은색으로 변환합니다.

이 알고리즘은 이미지를 스캔해서, 가장자리 주변의 검은색/하얀색/하나 이상의 사용자 지정 색상에 근사하거나 검은색/하얀색/하나 이상의 사용자 지정 색상인 모든 픽셀을 검은색 또는 하얀색으로 설정합니다. 이 알고리즘은 손실 있는 압축 항공사진을"고쳐서"모자이크 작업시 색상 픽셀을 투명하게 처리할 수 있도록 하는 데 자주 사용됩니다.

This algorithm is derived from the GDAL nearblack utility.

**기본 메뉴**: *Raster → Analysis*

### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [raster] | 입력 표고 래스터 레이어 |
| **How far from black (white)** | `NEAR` | [numeric: integer]<br>기본값: 15 | 픽셀값이 검은색/하얀색/사용자 지정 색상으로 간주되려면 검은색/하얀색/사용자 지정 색상과 얼마나 다를 수 있는지 선택합니다. |
| **Search for nearly white pixels instead of nearly black** | `WHITE` | [boolean]<br>기본값: False | 근사 검정색 픽셀 대신 근사 하얀색 (255) 픽셀을 검색합니다. |
| **Nearblack** | `OUTPUT` | [raster]<br>기본값: `[Save to temporary file]` | Specify the output raster layer. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

#### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Additional creation options**<br>선택적 | `CREATION_OPTIONS` (for QGIS <= 3.42, this was `OPTIONS`) | [string]<br>기본값: ‘’ | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL 드라이버 옵션* 참조) 배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (\|) 로 복수의 옵션들을 분리하십시오. |
| **Additional command-line parameters**<br>선택적 | `EXTRA` | [string]<br>기본값: 설정하지 않음 | 부가 GDAL 명령 줄 옵션을 추가합니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Nearblack** | `OUTPUT` | [raster] | 산출 래스터 |

### 파이썬 코드

**알고리즘 ID**: `gdal:nearblack`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 근접성 (래스터 거리)

각 픽셀의 중심 (center) 에서 대상 픽셀로 식별된 최근접 픽셀의 중심까지의 거리를 나타내는 래스터 근접성 (proximity) 맵을 생성합니다. 대상 픽셀은 래스터 픽셀값이 대상 픽셀값 집합에 있는 소스 래스터의 픽셀입니다.

This algorithm is derived from the GDAL proximity utility.

**기본 메뉴**: *Raster → Analysis*

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [raster] | 입력 표고 래스터 레이어 |
| **Band number** | `BAND` | [raster band]<br>기본값: 1 | 표고 정보를 담고 있는 밴드 |
| **List of target pixels**<br>선택적 | `VALUES` | [string]<br>기본값: '' | Comma-separated list of pixel values in the source image to consider as target pixels. If not specified, all non-zero pixels will be considered target pixels. |
| **Distance units** | `UNITS` | [enumeration]<br>기본값: 1 | 생성된 거리를 픽셀 단위로 나타낼지 또는 지리참조 좌표로 나타낼지 설정합니다. 다음 가운데 하나를 선택할 수 있습니다:<br>• 0 — 지리참조 좌표<br>• 1 — 픽셀 좌표 |
| **The maximum distance to generate**<br>선택적 | `MAX_DISTANCE` | [numeric: double]<br>기본값: 0.0 | The maximum distance to be generated. The NoData value will be used for pixels beyond this distance. Distance is interpreted according to the value of the *Distance units* parameter. |
| **Value to apply to pixels within the maximum distance**<br>선택적 | `REPLACE` | [numeric: double]<br>기본값: 0.0 | Specify a value to apply to all pixels that are within the maximum distance from target pixels (including the target pixels) instead of a distance value. |

다음 페이지에 계속

표 24.879 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **NoData value to use for the destination proximity raster**<br>선택적 | `NODATA` | [numeric: double]<br>기본값: 0.0 | Specify the NoData value to use for the pixels beyond the maximum distance. If a NoData value is not provided, the output band will be queried for its NoData value. If the output band does not have a NoData value, then the value `65535` will be used. |
| **Proximity map** | `OUTPUT` | [raster]<br>기본값: `[Save to temporary file]` | Specify the output raster layer. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Additional creation options**<br>선택적 | `CREATION_OPTIONS` (for QGIS <= 3.42, this was `OPTIONS`) | [string]<br>기본값: ‘’ | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL* 드라이버 옵션 참조)<br>배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (`|`) 로 복수의 옵션들을 분리하십시오. |
| **Additional command-line parameters**<br>선택적 | `EXTRA` | [string]<br>기본값: 설정하지 않음 | 부가 GDAL 명령 줄 옵션을 추가합니다. |

다음 페이지에 계속

표 24.880 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output data type** | DATA_TYPE | [enumeration] 기본값: 5 | 산출 래스터 파일의 데이터 유형을 정의합니다. 다음 가운데 하나로 지정할 수 있습니다:<br>• 0 —Byte (8 비트 부호 없는 정수 (quint8))<br>• 1 —Int16 (16 비트 부호 있는 정수 (qint16))<br>• 2 —UInt16 (16 비트 부호 없는 정수 (quint16))<br>• 3 —UInt32 (32 비트 부호 없는 정수 (quint32))<br>• 4 —Int32 (32 비트 부호 있는 정수 (qint32))<br>• 5 —Float32 (32 비트 부동소수점형 (float))<br>• 6 —Float64 (64 비트 부동소수점형 (double))<br>• 7 —CInt16 (Complex Int16)<br>• 8 —CInt32 (Complex Int32)<br>• 9 —CFloat32 (Complex Float32)<br>• 10 —CFloat64 (Complex Float64)<br>• 11 —Int8 (8 비트 부호 있는 정수 (qint8))<br>QGIS 와 함께 빌드된 GDAL 버전에 따라 사용할 수 있는 옵션이 달라집니다. (*Help* ⍰ *About* 메뉴 참조) |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Proximity map** | OUTPUT | [raster] | 산출 래스터 |

### 파이썬 코드

**알고리즘 ID**: `gdal:proximity`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 거칠기

표고로부터 계산한 값을 가진 단일 밴드 래스터를 산출합니다. 거칠기 (roughness) 란 표면의 불규칙한 정도를 말합니다. 거칠기는 중앙 픽셀과 그를 둘러싼 셀들 사이의 가장 큰 차이로 계산됩니다. 거칠기를 결정하는 작업은 지형 표고 데이터 분석에서 일정 역할을 합니다. 이 알고리즘은 하천 형태학 (river morphology), 기후학 (climatology), 자연 지리학 (physical geography) 전반의 계산에 유용합니다.

This algorithm is derived from the GDAL DEM utility.

**기본 메뉴**: *Raster → Analysis*

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [raster] | 입력 표고 래스터 레이어 |
| **Band number** | `BAND` | [raster band]<br>기본값: 1 | 표고로 사용할 밴드의 번호 |
| **Compute edges** | `COMPUTE_EDGES` | [boolean]<br>기본값: False | 표고 래스터에서 경계를 생성합니다. |
| **Roughness** | `OUTPUT` | [raster]<br>기본값: `[Save to temporary file]` | Specify the output raster layer. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Additional creation options**<br>선택적 | `CREATION_OPTIONS` (for QGIS <= 3.42, this was `OPTIONS`) | [string]<br>기본값: '' | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL* 드라이버 옵션 참조)<br>배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (\|) 로 복수의 옵션들을 분리하십시오. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Roughness** | `OUTPUT` | [raster] | 단일 밴드 산출 거칠기 래스터. -9999 값을 NODATA 값으로 사용합니다. |

### 파이썬 코드

**알고리즘 ID**: `gdal:roughness`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 거르기

지정한 (픽셀 단위) 한계 크기보다 작은 래스터 폴리곤을 제거하고 최대 이웃 폴리곤의 픽셀값으로 대체합니다. 이 알고리즘은 사용자 래스터 맵에 수많은 작은 영역이 있을 경우 유용합니다.

This algorithm is derived from the GDAL sieve utility.

**기본 메뉴**: *Raster → Analysis*

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [raster] | 입력 표고 래스터 레이어 |
| **Threshold** | `THRESHOLD` | [numeric: integer]<br>기본값: 10 | 이 크기보다 작은 래스터 폴리곤만 제거할 것입니다. |
| **Use 8-connectedness** | `EIGHT_CONNECTED` | [boolean]<br>기본값: False | If False, consider as neighbours only the (4) pixels that connect the edges horizontally or vertically. If True, every pixel that touches an edge or a corner is considered as neighbour. |
| **Do not use the default validity mask for the input band** | `NO_MASK` | [boolean]<br>기본값: False | |
| **Validity mask**<br>선택적 | `MASK_LAYER` | [raster] | 기본값 대신 사용할 무결성 마스크 |
| **Sieved** | `OUTPUT` | [raster]<br>기본값: `[Save to temporary file]` | Specify the output raster layer. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Additional command-line parameters**<br>선택적 | EXTRA | [string]<br>기본값: 설정하지 않음 | 부가 GDAL 명령 줄 옵션을 추가합니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Sieved** | OUTPUT | [raster] | 산출 래스터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `gdal:sieve`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 경사

GDAL 이 지원하는 모든 표고 래스터로부터 경사도 (slope) 맵을 생성합니다. 경사도란 수평에 대한 경사각 (angle of inclination) 을 말합니다. 경사도 값의 유형을 도 단위 또는 백분율로 지정할 수 있는 옵션이 있습니다.

This algorithm is derived from the GDAL DEM utility.

**기본 메뉴**: *Raster → Analysis*

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | INPUT | [raster] | 입력 표고 래스터 레이어 |
| **Band number** | BAND | [raster band]<br>기본값: 1 | 표고 정보를 담고 있는 밴드 |
| **Ratio of vertical units to horizontal** | SCALE | [numeric: double]<br>기본값: 1.0 | 수직 단위와 수평 단위의 비율 |

다음 페이지에 계속

표 24.888 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Express slope as percent (instead of degrees)** | `AS_PERCENT` | [boolean]<br>기본값: False | 경사도를 도 단위 대신 백분율로 표현합니다. |
| **Compute edges** | `COMPUTE_EDGES` | [boolean]<br>기본값: False | 표고 래스터에서 경계를 생성합니다. |
| **Use Zevenbergen&Thorne formula (instead of the Horn's one)** | `ZEVENBERGEN` | [boolean]<br>기본값: False | 매끄러운 풍경을 위해 체벤베르겐 & 손 (Zevenbergen&Thorne) 공식을 활성화합니다. |
| **Slope** | `OUTPUT` | [raster]<br>기본값: `[Save to temporary file]` | Specify the output raster layer. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Additional creation options**<br>선택적 | `CREATION_OPTIONS` (for QGIS <= 3.42, this was `OPTIONS`) | [string]<br>기본값: '' | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL 드라이버 옵션* 참조) 배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (\|) 로 복수의 옵션들을 분리하십시오. |
| **Additional command-line parameters**<br>선택적 | `EXTRA` | [string]<br>기본값: 설정하지 않음 | 부가 GDAL 명령 줄 옵션을 추가합니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Slope** | `OUTPUT` | [raster] | 산출 래스터 |

### 파이썬 코드

**알고리즘 ID**: `gdal:slope`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 지형 험상 지수 (TRI)

표고로부터 계산한 값을 가진 단일 밴드 래스터를 산출합니다. TRI 란 지형 험상 지수 (Terrain Ruggedness Index) 의 약자로, 중앙 픽셀과 그를 둘러싼 셀 사이의 평균 차 (mean difference) 로 정의됩니다.

This algorithm is derived from the GDAL DEM utility.

**기본 메뉴**: *Raster → Analysis*

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [raster] | 입력 표고 래스터 레이어 |
| **Band number** | `BAND` | [raster band]<br>기본값: 1 | 표고로 사용할 밴드의 번호 |
| **Compute edges** | `COMPUTE_EDGES` | [boolean]<br>기본값: False | 표고 래스터에서 경계를 생성합니다. |
| **Terrain Ruggedness Index** | `OUTPUT` | [raster]<br>기본값: `[Save to temporary file]` | Specify the output raster layer. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Additional creation options**<br>선택적 | `CREATION_OPTIONS`<br>(for QGIS <= 3.42, this was `OPTIONS`) | [string]<br>기본값: '' | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL* 드라이버 옵션 참조) 배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (\|) 로 복수의 옵션들을 분리하십시오. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Terrain Ruggedness Index** | `OUTPUT` | [raster] | 산출 험상 래스터. -9999 값을 NODATA 값으로 사용합니다. |

### 파이썬 코드

**알고리즘 ID**: `gdal:triterrainruggednessindex`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 위상 위치 지수 (TPI)

표고로부터 계산한 값을 가진 단일 밴드 래스터를 산출합니다. TPI 란 위상 위치 지수 (Topographic Position Index) 의 약자로, 중앙 픽셀과 그를 둘러싼 셀 사이의 차 (difference) 로 정의됩니다.

This algorithm is derived from the GDAL DEM utility.

**기본 메뉴**: *Raster → Analysis*

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [raster] | 입력 표고 래스터 레이어 |
| **Band number** | `BAND` | [raster band]<br>기본값: 1 | 표고 값으로 사용할 밴드의 번호 |
| **Compute edges** | `COMPUTE_EDGES` | [boolean]<br>기본값: False | 표고 래스터에서 경계를 생성합니다. |
| **Terrain Ruggedness Index** | `OUTPUT` | [raster]<br>기본값: `[Save to temporary file]` | Specify the output raster layer. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

#### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Additional creation options**<br>선택적 | `CREATION_OPTIONS`<br>(for QGIS <= 3.42, this was `OPTIONS`) | [string]<br>기본값: ' ' | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL* 드라이버 옵션 참조)<br>배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (\|) 로 복수의 옵션들을 분리하십시오. |

#### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Terrain Ruggedness Index** | `OUTPUT` | [raster] | 산출 래스터 |

#### 파이썬 코드

**알고리즘 ID**: `gdal:tpitopographicpositionindex`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 24.2.2 래스터 변환

#### gdal2xyz

래스터 데이터를 XYZ ASCII 파일 포맷으로 변환합니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [raster] | 변환할 래스터 레이어 |
| **Band number** | `BAND` | [raster band]<br>기본값: 입력 레이어의 첫 번째 밴드 | 래스터가 다중 밴드인 경우, 변환하려는 밴드를 선택하십시오. |
| **Input pixel value to treat as NoData**<br>선택적 | `NODATA_INPUT` | [numeric: integer]<br>기본값: 설정하지 않음 | Input pixel value to treat as "NoData"<br>**주의**<br>Using this parameter requires QGIS installed with GDAL >= 3.7 (see *Help* *About* menu). |
| **Destination NoData**<br>선택적 | `NODATA_OUTPUT` | [numeric: integer]<br>기본값: 설정하지 않음 | Assign specified "NoData"value to output.<br>**주의**<br>Using this parameter requires QGIS installed with GDAL >= 3.7 (see *Help* *About* menu). |
| **Do not output NoData values** | `SKIP_NODATA` | [boolean]<br>기본값: False | Do not output "NoData"values.<br>**주의**<br>Using this parameter requires QGIS installed with GDAL >= 3.3 (see *Help* *About* menu). |
| **Output comma-separated values** | `CSV` | [boolean]<br>기본값: False | 산출 파일이 쉼표 구분 값 (CSV) 유형이어야 하는지를 설정합니다. |
| **XYZ ASCII file** | `OUTPUT` | [file]<br>기본값: `[Save to temporary file]` | Specification of the output file. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **XYZ ASCII file** | `INPUT` | [table] | 래스터 밴드에서 추출한 값을 담고 있는 테이블 파일 |

#### 파이썬 코드

**알고리즘 ID**: `gdal:gdal2xyz`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### PCT 에서 RGB 로

8 비트 색상표 이미지를 24 비트 RBG 이미지로 변환합니다. 입력 파일의 의사색상 (pseudocolor) 밴드를 원하는 포맷의 RGB 파일로 변환할 것입니다.

This algorithm is derived from the GDAL pct2rgb utility.

**기본 메뉴**: *Raster → Conversion*

#### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [raster] | 입력 8 비트 래스터 이미지 |
| **Band number** | `BAND` | [raster band]<br>기본값: 입력 레이어의 첫 번째 밴드 | 래스터가 다중 밴드인 경우, 변환하려는 밴드를 선택하십시오. |
| **Generate a RGBA file** | `RGBA` | [boolean]<br>기본값: False | 산출 파일이 RGBA 유형이어야 하는지를 설정합니다. |
| **PCT to RGB** | `OUTPUT` | [file]<br>기본값: `[Save to temporary file]` | Specification of the output file. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

#### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **PCT to RGB** | `OUTPUT` | [raster] | 24 비트 RGB 래스터 이미지 |

### 파이썬 코드

**알고리즘 ID**: `gdal:pcttorgb`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 폴리곤화 (래스터를 벡터로)

래스터에서 공통 픽셀값을 공유하는 픽셀들이 연속된 모든 영역에 대한 벡터 폴리곤을 생성합니다. 각 폴리곤의 픽셀값을 나타내는 속성과 함께 폴리곤을 생성합니다.

This algorithm is derived from the GDAL polygonize utility.

**기본 메뉴**: *Raster → Conversion*

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [raster] | 입력 래스터 레이어 |
| **Band number** | `BAND` | [raster band]<br>기본값: 입력 레이어의 첫 번째 밴드 | 래스터가 다중 밴드인 경우, 사용하려는 밴드를 선택하십시오. |
| **Name of the field to create** | `FIELD` | [string]<br>기본값: 'DN' | 연속 영역의 속성을 가질 필드의 이름을 지정합니다. |
| **Use 8-connectedness** | `EIGHT_CONNECTED` | [boolean]<br>기본값: False | 이 파라미터를 활성화하지 않으면, 래스터 셀이 연결된 것으로 간주되려면 공통 경계를 가지고 있어야만 합니다 (*4*-연결성). 활성화한 경우, 접촉하는 래스터 셀도 연결된 것으로 간주합니다 (*8*-연결성). |
| **Vectorized** | `OUTPUT` | [vector: polygon]<br>기본값: `[Save to temporary file]` | Specification of the output (polygon) vector layer. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Additional command-line parameters**<br>선택적 | EXTRA | [string]<br>기본값: 설정하지 않음 | 부가 GDAL 명령 줄 옵션을 추가합니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Vectorized** | OUTPUT | [vector: polygon] | 산출 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `gdal:polygonize`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 밴드 재배열하기

입력 래스터 레이어에서 선택한 밴드 (들) 를 사용해서 새 래스터를 생성합니다. 이 알고리즘은 새로 성생된 래스터의 밴드들을 재정렬할 수도 있습니다.

This algorithm is derived from the GDAL translate utility.

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | INPUT | [raster] | 입력 래스터 레이어 |
| **Selected band(s)** | BANDS | [raster band] [list]<br>기본값: 설정하지 않음 | 새 래스터를 생성하기 위해 사용할 밴드의 정렬된 목록 |
| **Converted** | OUTPUT | [raster]<br>기본값: [Save to temporary file] | Specification of the output raster. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Additional creation options**<br>선택적 | `CRE-ATION_OPTIONS`<br>(for QGIS <= 3.42, this was `OPTIONS`) | [string]<br>기본값: ' | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL 드라이버 옵션* 참조)<br>배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (\|) 로 복수의 옵션들을 분리하십시오. |
| **Output data type** | `DATA_TYPE` | [enumeration]<br>기본값: 0 | 산출 래스터 파일의 데이터 유형을 정의합니다. 다음 가운데 하나로 지정할 수 있습니다:<br>• 0 —입력 레이어의 데이터 유형 사용<br>• 1 —Byte (8 비트 부호 없는 정수 (quint8))<br>• 2 —Int16 (16 비트 부호 있는 정수 (qint16))<br>• 3 —UInt16 (16 비트 부호 없는 정수 (quint16))<br>• 4 —UInt32 (32 비트 부호 없는 정수 (quint32))<br>• 5 —Int32 (32 비트 부호 있는 정수 (qint32))<br>• 6 —Float32 (32 비트 부동소수점형 (float))<br>• 7 —Float64 (64 비트 부동소수점형 (double))<br>• 8 —CInt16 (Complex Int16)<br>• 9 —CInt32 (Complex Int32)<br>• 10 —CFloat32 (Complex Float32)<br>• 11 —CFloat64 (Complex Float64)<br>• 12 —Int8 (8 비트 부호 있는 정수 (qint8))<br>QGIS 와 함께 빌드된 GDAL 버전에 따라 사용할 수 있는 옵션이 달라집니다. (*Help → About* 메뉴 참조) |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Converted** | `OUTPUT` | [raster] | 재배열한 밴드를 가진 산출 래스터 레이어 |

#### 파이썬 코드

**알고리즘 ID**: `gdal:rearrange_bands`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### RGB 를 PCT 로

24 비트 RGB 이미지를 8 비트 색상표 이미지로 변환합니다. 다운샘플링된 RGB 히스토그램 상에서 중간값 자르기 (median cut) 알고리즘을 사용해서 입력 RGB 이미지에 대한 최적의 의사색상을 계산합니다. 이 변환 알고리즘은 산출 이미지의 시각적 질을 최대화하기 위해 플로이드-스타인버그 디더링 (오차확산법) 을 활용합니다.

래스터 맵을 범주화하는데 범주의 개수를 줄이고자 하는 경우 그 전에 사용자 이미지를 이 알고리즘으로 다운샘플링하는 것이 유용할 수 있습니다.

This algorithm is derived from the GDAL rgb2pct utility.

**기본 메뉴**: *Raster → Conversion*

#### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [raster] | 입력 (RGB) 래스터 레이어 |
| **Number of colors** | `NCOLORS` | [numeric: integer]<br>기본값: 2 | 산출 이미지가 담게 될 색상의 개수. 2 에서 256 까지의 값을 사용할 수 있습니다. |
| **RGB to PCT** | `OUTPUT` | [raster]<br>기본값: `[Save to temporary file]` | Specification of the output raster. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

#### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **RGB to PCT** | `OUTPUT` | [raster] | 산출 래스터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `gdal:rgbtopct`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 번역하기 (포맷 변환)

래스터 데이터를 서로 다른 포맷으로 변환합니다.

This algorithm is derived from the GDAL translate utility.

**기본 메뉴**: *Raster → Conversion*

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [raster] | 입력 래스터 레이어 |
| **Override the projection of the output file**<br>선택적 | `TARGET_CRS` | [crs] | 산출 파일용 투영체를 지정합니다. |
| **Assign a specified NoData value to output bands**<br>선택적 | `NODATA` | [numeric: double]<br>기본값: 설정하지 않음 | 산출 래스터에 있는 NODATA 에 사용할 값을 정의합니다 |
| **Copy all subdatasets of this file to individual output files** | `COPY_SUBDATASET` | [boolean]<br>기본값: False | 하위 데이터셋 (subdataset) 용 개별 파일을 생성합니다. |
| **Converted** | `OUTPUT` | [raster]<br>기본값: `[Save to temporary file]` | Specification of the output (translated) raster layer. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Additional creation options**<br>선택적 | `CREATION_OPTIONS` (for QGIS <= 3.42, this was `OPTIONS`) | [string]<br>기본값: ' ' | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL 드라이버 옵션* 참조)<br>배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (\|) 로 복수의 옵션들을 분리하십시오. |
| **Additional command-line parameters**<br>선택적 | `EXTRA` | [string]<br>기본값: 설정하지 않음 | 부가 GDAL 명령 줄 옵션을 추가합니다. |
| **Output data type** | `DATA_TYPE` | [enumeration]<br>기본값: 0 | 산출 래스터 파일의 데이터 유형을 정의합니다. 다음 가운데 하나로 지정할 수 있습니다:<br>• 0 —입력 레이어의 데이터 유형 사용<br>• 1 —Byte (8 비트 부호 없는 정수 (quint8))<br>• 2 —Int16 (16 비트 부호 있는 정수 (qint16))<br>• 3 —UInt16 (16 비트 부호 없는 정수 (quint16))<br>• 4 —UInt32 (32 비트 부호 없는 정수 (quint32))<br>• 5 —Int32 (32 비트 부호 있는 정수 (qint32))<br>• 6 —Float32 (32 비트 부동소수점형 (float))<br>• 7 —Float64 (64 비트 부동소수점형 (double))<br>• 8 —CInt16 (Complex Int16)<br>• 9 —CInt32 (Complex Int32)<br>• 10 —CFloat32 (Complex Float32)<br>• 11 —CFloat64 (Complex Float64)<br>• 12 —Int8 (8 비트 부호 있는 정수 (qint8))<br>QGIS 와 함께 빌드된 GDAL 버전에 따라 사용할 수 있는 옵션이 달라집니다. (*Help* ▸ *About* 메뉴 참조) |

#### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Converted** | `OUTPUT` | [raster] | 산출 (번역된) 래스터 레이어 |

#### 파이썬 코드

**알고리즘 ID**: `gdal:translate`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 24.2.3 래스터 추출

#### 래스터를 범위로 잘라내기

GDAL 이 지원하는 모든 래스터 파일을 지정한 범위로 잘라냅니다.

This algorithm is derived from the GDAL translate utility.

**기본 메뉴**: *Raster → Extraction*

#### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [raster] | 입력 래스터 |

다음 페이지에 계속

표 24.912 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Clipping extent** | EXTENT | [extent] | 산출 래스터에 사용될 범위. 지정한 경계 상자 안에 들어오는 픽셀만 산출물에 포함될 것입니다.<br>다음과 같은 메소드를 사용할 수 있습니다:<br>• 레이어로부터 계산…: 현재 프로젝트에 불러온 레이어의 범위를 사용합니다.<br>• 레이어 맵으로부터 계산…: 활성 프로젝트의조판 맵 항목 범위를 사용합니다.<br>• 북마크로부터 계산…: 저장되어 있는북마크 의 범위를 사용합니다.<br>• 맵 캔버스 범위 사용<br>• 캔버스에 그리기: 연산에 넣을 범위를 정하는 직사각형을 클릭 & 드래그합니다.<br>• 좌표를 xmin, xmax, ymin, ymax 로 입력 |
| **Override the projection for the output file** | OVERCRS | [boolean]<br>기본값: False | 체크박스를 체크하면, 산출 파일에 입력 레이어의 좌표계를 할당합니다. |
| **Assign a specified NoData value to output bands**<br>선택적 | NODATA | [numeric: double]<br>기본값: 설정하지 않음 | 산출 래스터에 있는 NODATA 에 사용할 값을 정의합니다 |
| **Clipped (extent)** | OUTPUT | [raster]<br>기본값: [Save to temporary file] | Specification of the output raster layer. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Additional creation options**<br>선택적 | CREATION_OPTIONS (for QGIS <= 3.42, this was OPTIONS) | [string]<br>기본값: '' | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL 드라이버 옵션* 참조)<br>배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (\|) 로 복수의 옵션들을 분리하십시오. |

다음 페이지에 계속

표 24.913 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output data type** | DATA_TYPE | [enumeration]<br>기본값: 0 | 산출 래스터 파일의 포맷을 정의합니다.<br>옵션:<br>• 0 —입력 레이어의 데이터 유형 사용<br>• 1 —Byte (8 비트 부호 없는 정수 (quint8))<br>• 2 —Int16 (16 비트 부호 있는 정수 (qint16))<br>• 3 —UInt16 (16 비트 부호 없는 정수 (quint16))<br>• 4 —UInt32 (32 비트 부호 없는 정수 (quint32))<br>• 5 —Int32 (32 비트 부호 있는 정수 (qint32))<br>• 6 —Float32 (32 비트 부동소수점형 (float))<br>• 7 —Float64 (64 비트 부동소수점형 (double))<br>• 8 —CInt16 (Complex Int16)<br>• 9 —CInt32 (Complex Int32)<br>• 10 —CFloat32 (Complex Float32)<br>• 11 —CFloat64 (Complex Float64)<br>• 12 —Int8 (8 비트 부호 있는 정수 (qint8))<br>QGIS 와 함께 빌드된 GDAL 버전에 따라 사용할 수 있는 옵션이 달라집니다. (*Help* *About* 메뉴 참조) |
| **Additional command-line parameters**<br>선택적 | EXTRA | [string]<br>기본값: 설정하지 않음 | 부가 GDAL 명령 줄 옵션을 추가합니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Clipped (extent)** | OUTPUT | [raster] | 지정한 범위로 잘라낸 산출 래스터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `gdal:cliprasterbyextent`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 래스터를 마스크 레이어로 잘라내기

GDAL 이 지원하는 모든 래스터 파일을 벡터 마스크 레이어로 잘라냅니다.

This algorithm is derived from the GDAL warp utility.

**기본 메뉴**: *Raster → Extraction*

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [raster] | 입력 래스터 |
| **Mask layer** | `MASK` | [vector: polygon] | 래스터를 잘라내기 위한 벡터 마스크 |
| **Source CRS** | `SOURCE_CRS` | [crs] | 좌표 참조에 입력 래스터를 사용하도록 설정 |
| **Target CRS** | `TARGET_CRS` | [crs] | 좌표 참조에 마스크 레이어를 사용하도록 설정 |
| **Target extent**<br>선택적 | `TARGET_EXTENT` | [extent] | 생성할 산출 파일의 범위<br>다음과 같은 메소드를 사용할 수 있습니다:<br>• 레이어로부터 계산…: 현재 프로젝트에 불러온 레이어의 범위를 사용합니다.<br>• 레이어 맵으로부터 계산…: 활성 프로젝트의조판 맵 항목 범위를 사용합니다.<br>• 북마크로부터 계산…: 저장되어 있는북마크 의 범위를 사용합니다.<br>• 맵 캔버스 범위 사용<br>• 캔버스에 그리기: 연산에 넣을 범위를 정하는 직사각형을 클릭 & 드래그합니다.<br>• 좌표를 `xmin, xmax, ymin, ymax` 로 입력 |
| **Assign a specified NoData value to output bands**<br>선택적 | `NODATA` | [numeric: double]<br>기본값: 설정하지 않음 | 산출 래스터에 있는 NODATA 에 사용할 값을 정의합니다 |
| **Create an output alpha band** | `ALPHA_BAND` | [boolean]<br>기본값: False | 산출물용 알파 밴드를 생성합니다. 그러면 알파 밴드가 픽셀에 대한 투명도 값을 담습니다. |

다음 페이지에 계속

표 24.915 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Match the extent of the clipped raster to the extent of the mask layer** | `CROP_TO_CUTLINE` | [boolean]<br>기본값: True | 이 파라미터를 활성화하면 산출 래스터에 벡터 레이어 범위를 적용합니다. |
| **Keep resolution of input raster** | `KEEP_RESOLUTION` | [boolean]<br>기본값: False | 이 파라미터를 활성화하면 산출 래스터의 해상도를 변경하지 않을 것입니다. |
| **Set output file resolution** | `SET_RESOLUTION` | [boolean]<br>기본값: False | 이 파라미터를 활성화하면 산출물 해상도 (셀 크기) 를 지정합니다. |
| **X Resolution to output bands**<br>선택적 | `X_RESOLUTION` | [numeric: double]<br>기본값: 설정하지 않음 | 산출 래스터의 셀의 너비 |
| **Y Resolution to output band**<br>선택적 | `Y_RESOLUTION` | [numeric: double]<br>기본값: 설정하지 않음 | 산출 래스터의 셀의 높이 |
| **Use multithreaded warping implementation** | `MULTITHREADING` | [boolean]<br>기본값: False | 이미지 무리를 처리하고 입력/출력 작업을 동시에 수행하기 위해 스레드 2 개를 사용할 것입니다. 계산 자체는 다중 스레드 작업이 아니라는 사실을 기억하십시오. |
| **Clipped (mask)** | `OUTPUT` | [raster]<br>기본값: `[Save to temporary file]` | Specification of the output raster layer. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Additional creation options**<br>선택적 | `CREATION_OPTIONS` (for QGIS <= 3.42, this was `OPTIONS`) | [string]<br>기본값: '' | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL 드라이버 옵션* 참조) 배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (\|) 로 복수의 옵션들을 분리하십시오. |

다음 페이지에 계속

표 24.916 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output data type** | DATA_TYPE | [enumeration]<br>기본값: 0 | 산출 래스터 파일의 포맷을 정의합니다.<br>옵션:<br>• 0 —입력 레이어의 데이터 유형 사용<br>• 1 —Byte (8 비트 부호 없는 정수 (quint8))<br>• 2 —Int16 (16 비트 부호 있는 정수 (qint16))<br>• 3 —UInt16 (16 비트 부호 없는 정수 (quint16))<br>• 4 —UInt32 (32 비트 부호 없는 정수 (quint32))<br>• 5 —Int32 (32 비트 부호 있는 정수 (qint32))<br>• 6 —Float32 (32 비트 부동소수점형 (float))<br>• 7 —Float64 (64 비트 부동소수점형 (double))<br>• 8 —CInt16 (Complex Int16)<br>• 9 —CInt32 (Complex Int32)<br>• 10 —CFloat32 (Complex Float32)<br>• 11 —CFloat64 (Complex Float64)<br>• 12 —Int8 (8 비트 부호 있는 정수 (qint8))<br>QGIS 와 함께 빌드된 GDAL 버전에 따라 사용할 수 있는 옵션이 달라집니다. (*Help* *About* 메뉴 참조) |
| **Additional command-line parameters**<br>선택적 | EXTRA | [string]<br>기본값: 설정하지 않음 | 부가 GDAL 명령 줄 옵션을 추가합니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Clipped (mask)** | OUTPUT | [raster] | 벡터 레이어로 잘라낸 산출 래스터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `gdal:cliprasterbymasklayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 등고선

GDAL 이 지원하는 모든 표고 래스터에서 등고선 (contour line) 을 추출합니다.

This algorithm is derived from the GDAL contour utility.

**기본 메뉴**: *Raster → Extraction*

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [raster] | 입력 래스터 |
| **Band number** | `BAND` | [raster band]<br>기본값: 1 | 등고선을 생성할 래스터 밴드 |
| **Interval between contour lines** | `INTERVAL` | [numeric: double]<br>기본값: 10.0 | 등고선 사이의 간격을 지정한 표고 래스터의 단위로 정의합니다. (최소값: 0) |
| **Attribute name (if not set, no elevation attribute is attached)**<br>선택적 | `FIELD_NAME` | [string]<br>기본값: 'ELEV' | 표고를 넣을 속성을 위한 이름을 지정합니다. |
| **Offset from zero relative to which to interpret intervals**<br>선택적 | `OFFSET` | [numeric: double]<br>기본값: 0.0 | |
| **Contours** | `OUTPUT` | [vector: line]<br>기본값: `[Save to temporary file]` | Specification of the output vector layer. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Produce 3D vector** | `CREATE_3D` | [boolean]<br>기본값: False | 2 차원 대신 3 차원 벡터를 생성하도록 강제합니다. 모든 꼭짓점에 표고를 포함시킵니다. |
| **Treat all raster values as valid** | `IGNORE_NODATA` | [boolean]<br>기본값: False | 데이터셋에 있는 모든 NODATA 값을 무시합니다. |
| **Input pixel value to treat as "NoData"**<br>선택적 | `NODATA` | [numeric: double]<br>기본값: 설정하지 않음 | 산출 래스터에 있는 NODATA 에 사용할 값을 정의합니다 |
| **Additional command-line parameters**<br>선택적 | `EXTRA` | [string]<br>기본값: 설정하지 않음 | 추가적인 GDAL 명령 줄 옵션을 추가합니다. 대응하는 GDAL 유틸리티 문서를 참조합니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Contours** | `OUTPUT` | [vector: line] | 등고선을 담은 산출 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `gdal:contour`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 등고선 폴리곤

GDAL 이 지원하는 모든 표고 래스터에서 등고선 폴리곤을 추출합니다.

This algorithm is derived from the GDAL contour utility.

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [raster] | 입력 래스터 |
| **Band number** | `BAND` | [raster band]<br>기본값: 1 | 등고선을 생성할 래스터 밴드 |
| **Interval between contour lines** | `INTERVAL` | [numeric: double]<br>기본값: 10.0 | 등고선 사이의 간격을 지정한 표고 래스터의 단위로 정의합니다. (최소값: 0) |
| **Offset from zero relative to which to interpret intervals**<br>선택적 | `OFFSET` | [numeric: double]<br>기본값: 0.0 | |
| **Attribute name for minimum elevation of contour polygon**<br>선택적 | `FIELD_NAME_MIN` | [string]<br>기본값: 'ELEV_MIN' | 등고선 폴리곤의 최저 표고를 넣을 속성을 위한 이름을 지정합니다. 아무 이름도 지정하지 않으면 최저 표고 속성을 추가하지 않습니다. |
| **Attribute name for maximum elevation of contour polygon**<br>선택적 | `FIELD_NAME_MAX` | [string]<br>기본값: 'ELEV_MAX' | 등고선 폴리곤의 최고 표고를 넣을 속성을 위한 이름을 지정합니다. 아무 이름도 지정하지 않으면 최고 표고 속성을 추가하지 않습니다. |

다음 페이지에 계속

표 24.921 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Contours** | `OUTPUT` | [vector: polygon]<br>기본값: `[Save to temporary file]` | Specification of the output vector layer. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Produce 3D vector** | `CREATE_3D` | [boolean]<br>기본값: False | 2 차원 대신 3 차원 벡터를 생성하도록 강제합니다. 모든 꼭짓점에 표고를 포함시킵니다. |
| **Treat all raster values as valid** | `IGNORE_NODATA` | [boolean]<br>기본값: False | 데이터셋에 있는 모든 NODATA 값을 무시합니다. |
| **Input pixel value to treat as "NoData"**<br>선택적 | `NODATA` | [numeric: double]<br>기본값: 설정하지 않음 | 산출 래스터에 있는 NODATA 에 사용할 값을 정의합니다 |
| **Additional command-line parameters**<br>선택적 | `EXTRA` | [string]<br>기본값: 설정하지 않음 | 추가적인 GDAL 명령 줄 옵션을 추가합니다. 대응하는 GDAL 유틸리티 문서를 참조합니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Contours** | `OUTPUT` | [vector: polygon] | 등고선 폴리곤을 담은 산출 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `gdal:contour_polygon`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 24.2.4 래스터 기타 알고리즘

#### 오버뷰 작성하기 (피라미드)

래스터 레이어의 렌더링 시간을 단축하기 위해 오버뷰 (피라미드) 를 생성할 수 있습니다. 오버뷰란 QGIS 가 확대/축소 수준에 따라 사용하는 데이터의 저해상도 복사본을 말합니다.

This algorithm is derived from the GDAL addo utility.

**기본 메뉴**: *Raster → Miscellaneous*

#### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [raster] | 입력 래스터 레이어 |
| **Remove all existing overviews** | `CLEAN` | [boolean]<br>기본값: False | 래스터에서 기존 오버뷰를 제거합니다. 기본값은 제거하지 않는다입니다. |

#### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Overview levels** | `LEVELS` | [string]<br>기본값: '2 4 8 16' | 입력 래스터 레이어의 원본 해상도에서 계산할 오버뷰 수준의 개수를 정의합니다. 기본적으로 4 개의 수준을 계산합니다. |
| **Resampling method**<br>선택적 | `RESAMPLING` | [enumeration]<br>기본값: 0 | 정의된 리샘플링 방법으로 오버뷰를 계산합니다. 다음과 같은 리샘플링 방법을 사용할 수 있습니다:<br>• 0 – 최근접 이웃 (`nearest`)<br>• 1 – 평균 (`average`)<br>• 2 – 가우스 (`gauss`)<br>• 3 — 3 차 (4x4 커널) (`cubic`)<br>• 4 — 3 차 B-스플라인 (4x4 커널) (`cubicspline`)<br>• 5 — 란초시 (6x6 커널) (`lanczos`)<br>• 6 – 평균 MP (`average_mp`)<br>• 7 – 스펙트럼/페이즈 공간에서의 평균 (Average in Mag/Phase Space) (`average_magphase`)<br>• 8 – 모드 (`mode`) |

다음 페이지에 계속

표 24.925 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Overviews format** 선택적 | FORMAT | [enumeration] 기본값: 0 | 오버뷰를 내부적으로 저장할 수도 있고, GTiff 또는 ERDAS Imagine 파일로 외부적으로 저장할 수도 있습니다. 오버뷰는 기본적으로 산출 래스터로 저장됩니다. 다음과 같은 포맷을 사용할 수 있습니다:<br>• 0 –내부 저장 (가능한 경우)<br>• 1 –외부 저장 (GTiff .ovr)<br>• 2 –외부 저장 (ERDAS Imagine .aux) |
| **Additional command-line parameters** 선택적 | EXTRA | [string] 기본값: 설정하지 않음 | 부가 GDAL 명령 줄 옵션을 추가합니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Pyramidized** | OUTPUT | [raster] | 오버뷰를 담은 산출 래스터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `gdal:overviews`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 가상 래스터 작성하기

GDAL 이 지원하는 입력 래스터 목록의 모자이크인 가상 데이터셋 (VRT) 을 작성합니다. 모자이크를 사용해서 여러 래스터 파일을 병합할 수 있습니다.

This algorithm is derived from the GDAL buildvrt utility.

**기본 메뉴**: *Raster → Miscellaneous*

### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layers** | INPUT | [raster] [list] | GDAL 이 지원하는 래스터 레이어 |
| **Resolution** | RESOLUTION | [enumeration]<br>기본값: 0 | 모자이크의 산출 해상도. 기본적으로 래스터 파일들의 평균 해상도를 선택할 것입니다.<br>옵션:<br>• 0 —평균 (average)<br>• 1 —최고 (highest)<br>• 2 —최저 (lowest) |
| **Place each input file into a separate band** | SEPARATE | [boolean]<br>기본값: False | '참'으로 설정하면 VRT 의 개별 적층 (stacked) 밴드에 각 래스터 파일을 넣도록 정의할 수 있습니다. |
| **Allow projection difference** | PROJ_DIFFERENCE | [boolean]<br>기본값: False | 산출 밴드들이 입력 래스터 레이어들의 투영체에서 추출한 서로 다른 투영체를 가지도록 허용합니다. |
| **Virtual** | OUTPUT | [raster]<br>기본값: [Save to temporary file] | Specification of the output raster layer. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

#### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Add alpha mask band to VRT when source raster has none** | ADD_ALPHA | [boolean]<br>기본값: False | 소스 래스터에 알파 마스크 밴드가 없을 경우, VRT 에 알파 마스크 밴드를 추가합니다. |
| **Override projection for the output file**<br>선택적 | ASSIGN_CRS | [crs]<br>기본값: 설정하지 않음 | 산출 파일용 투영체를 무시합니다. 재투영하지 않습니다. |
| **Resampling algorithm** | RESAMPLING | [enumeration]<br>기본값: 0 | The resampling algorithm to use. Options:<br>• 0 —최근접 이웃 (nearest)<br>• 1 —쌍선형 (2x2 커널) (bilinear)<br>• 2 —3 차 (4x4 커널) (cubic)<br>• 3 —3 차 B-스플라인 (4x4 커널) (cubicspline)<br>• 4 —란초시 (6x6 커널) (lanczos)<br>• 5 —평균 (average)<br>• 6 —모드 (mode) |

다음 페이지에 계속

표 24.928 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **NoData value(s) for input bands (space separated)** 선택적 | SRC_NODATA | [string] 기본값: 설정하지 않음 | 입력 밴드 (들) 를 위한 공백으로 구분된 NODATA 값 (들) |
| **Additional command-line parameters** | EXTRA | [string] 기본값: 설정하지 않음 | 부가 GDAL 명령 줄 옵션을 추가합니다. |

산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Virtual** | OUTPUT | [raster] | 산출 래스터 레이어 |

파이썬 코드

**알고리즘 ID**: `gdal:buildvirtualraster`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### gdal2tiles

Generates a directory with small tiles and metadata, following the OSGeo Tile Map Service Specification. See also the OpenGIS Web Map Tile Service Implementation Standard. Simple web pages with viewers based on Google Maps, OpenLayers and Leaflet are generated as well. To explore your maps on-line in the web browser, you only need to upload the generated directory onto a web server.

이 알고리즘은 제공 맵이 `EPSG:4326` 투영체를 사용하는 경우 구글 어스용 필수 메타데이터 (KML SuperOverlay) 도 생성합니다.

타일 생성 도중 ESRI 월드 파일과 내장 지리참조를 사용하지만, 제대로 된 지리참조 없이도 그림을 발행할 수 있습니다.

This algorithm is derived from the GDAL gdal2tiles utility.

### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [raster] | GDAL 이 지원하는 래스터 레이어 |
| **Tile cutting profile** | `PROFILE` | [enumeration]<br>기본값: 0 | 다음 가운데 하나로 지정할 수 있습니다:<br>• 0 —메르카토르 (`mercator`)<br>• 1 —측지선 (`geodetic`)<br>• 2 —래스터 (`raster`) |
| **Zoom levels to render**<br>선택적 | `ZOOM` | [string]<br>기본값: | |
| **Web viewer to generate** | `VIEWER` | [enumeration]<br>기본값: 0 | 다음 가운데 하나로 지정할 수 있습니다:<br>• 0 —모두 (`all`)<br>• 1 —구글 지도 (`google`)<br>• 2 —오픈레이어 (`openlayers`)<br>• 3 —리플렛 (`leaflet`)<br>• 4 —없음 (`none`) |
| **Title of the map**<br>선택적 | `TITLE` | [string]<br>기본값: | |
| **Copyright of the map** | `COPYRIGHT` | [string]<br>기본값: | |
| **Output directory** | `OUTPUT` | [folder]<br>기본값: `[Save to temporary folder]` | Specify the output folder for the tiles. *One of*:<br>• 임시 디렉터리에 저장<br>• 디렉터리에 저장 |

#### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Resampling method** | `RESAMPLING` | [enumeration]<br>기본값: 0 | 사용할 리샘플링 알고리즘입니다. 다음 가운데 하나를 선택할 수 있습니다:<br>• 0 —평균 (`average`)<br>• 1 —최근접 이웃 (`nearest`)<br>• 2 —쌍선형 (2x2 커널) (`bilinear`)<br>• 3 —3 차 (4x4 커널) (`cubic`)<br>• 4 —3 차 B-스플라인 (4x4 커널) (`cubicspline`)<br>• 5 —란초시 (6x6 커널) (`lanczos`)<br>• 6 —에일리어싱 제거 (`antialias`) |

다음 페이지에 계속

표 24.931 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **The spatial reference system used for the source input data**<br>선택적 | SOURCE_CRS | [crs]<br>기본값: 설정하지 않음 | |
| **Transparency value to assign to the input data**<br>선택적 | NODATA | [numeric: double]<br>기본값: 0.0 | |
| **URL address where the generated tiles are going to be published**<br>선택적 | URL | [string]<br>기본값: ' ' | |
| **Google Maps API key (http://code.google.cc**<br>선택적 | GOOGLE_KEY | [string]<br>기본값: ' ' | 사용자의 구글 지도 API 키 |
| **Bing Maps API key (https://www.bingma**<br>선택적 | BING_KEY | [string]<br>기본값: ' ' | 사용자의 빙 맵 API 키 |
| **Generate only missing files** | RESUME | [boolean]<br>기본값: False | |
| **Generate KML for Google Earth** | KML | [boolean]<br>기본값: False | |
| **Avoid automatic generation of KML files for EPSG:4326** | NO_KML | [boolean]<br>기본값: False | |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output directory** | OUTPUT | [folder] | 타일을 저장한 산출물 폴더 |

### 파이썬 코드

**알고리즘 ID**: `gdal:gdal2tiles`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 병합하기

래스터 파일들을 단순한 방식으로 병합합니다. 이 알고리즘은 입력 래스터의 의사색상 테이블을 사용해서 산출 래스터 유형을 정의할 수 있습니다. 모든 이미지가 동일한 좌표계를 사용해야만 합니다.

This algorithm is derived from the GDAL merge utility.

**기본 메뉴**: *Raster → Miscellaneous*

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layers** | `INPUT` | [raster] [list] | 입력 래스터 레이어 |
| **Grab pseudocolor table from first layer** | `PCT` | [boolean]<br>기본값: False | 색상 작업에 첫 번째 레이어의 의사색상 테이블을 사용할 것입니다. |
| **Place each input file into a separate band** | `SEPARATE` | [boolean]<br>기본값: False | 개별 밴드에 각 입력 파일을 배치합니다. |
| **Output data type** | `DATA_TYPE` | [enumeration]<br>기본값: 5 | 산출 래스터 파일의 포맷을 정의합니다. 다음 가운데 하나로 지정할 수 있습니다:<br>• 0 —Byte (8 비트 부호 없는 정수 (quint8))<br>• 1 —Int16 (16 비트 부호 있는 정수 (qint16))<br>• 2 —UInt16 (16 비트 부호 없는 정수 (quint16))<br>• 3 —UInt32 (32 비트 부호 없는 정수 (quint32))<br>• 4 —Int32 (32 비트 부호 있는 정수 (qint32))<br>• 5 —Float32 (32 비트 부동소수점형 (float))<br>• 6 —Float64 (64 비트 부동소수점형 (double))<br>• 7 —CInt16 (Complex Int16)<br>• 8 —CInt32 (Complex Int32)<br>• 9 —CFloat32 (Complex Float32)<br>• 10 —CFloat64 (Complex Float64)<br>• 11 —Int8 (8 비트 부호 있는 정수 (qint8))<br>QGIS 와 함께 빌드된 GDAL 버전에 따라 사용할 수 있는 옵션이 달라집니다. (*Help → About* 메뉴 참조) |
| **Merged** | `OUTPUT` | [raster]<br>기본값: `[Save to temporary file]` | Specification of the output raster layer. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input pixel value to treat as "NoData"**<br>선택적 | `NODATA_INPUT` | [numeric: double]<br>기본값: 설정하지 않음 | 병합되는 파일들에서 이 픽셀값을 가진 픽셀을 무시합니다. |
| **Assign specified "NoData" value to output**<br>선택적 | `NODATA_OUTPUT` | [numeric: double]<br>기본값: 설정하지 않음 | 산출 밴드에 지정한 NODATA 값을 할당합니다. |
| **Additional creation options**<br>선택적 | `CREATION_OPTIONS` (for QGIS <= 3.42, this was `OPTIONS`) | [string]<br>기본값: '' | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL 드라이버 옵션* 참조)<br>배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (`\|`) 로 복수의 옵션들을 분리하십시오. |
| **Additional command-line parameters** | `EXTRA` | [string]<br>기본값: 설정하지 않음 | 부가 GDAL 명령 줄 옵션을 추가합니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Merged** | `OUTPUT` | [raster] | 산출 래스터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `gdal:merge`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 영상융합

영상융합 (pan-sharpening) 작업을 수행합니다. 이 알고리즘은 (GeoTIFF 같은) "대표적인"산출 데이터셋을 생성할 수도 있고, 또는 영상융합 작업을 서술하는 VRT 데이터셋을 생성할 수도 있습니다.

See GDAL Pansharpen.

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Spectral dataset** | `SPECTRAL` | [raster] | 입력 (스펙트럼) 래스터 레이어 |
| **Panchromatic dataset** | `PANCHROMATIC` | [raster] | 입력 (전정색 (□□□)) 래스터 레이어 |
| **Output** | `OUTPUT` | [raster]<br>기본값: `[Save to temporary file]` | Specify the output (sharpened) raster layer. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Resampling algorithm** | `RESAMPLING` | [enumeration]<br>기본값: 2 | The resampling algorithm to use. Options:<br>• 0 —최근접 이웃 (`nearest`)<br>• 1 —쌍선형 (2x2 커널) (`bilinear`)<br>• 2 —3 차 (4x4 커널) (`cubic`)<br>• 3 —3 차 B-스플라인 (4x4 커널) (`cubicspline`)<br>• 4 —란초시 (6x6 커널) (`lanczos`)<br>• 5 —평균 (`average`) |
| **Additional creation options**<br>선택적 | `CREATION_OPTIONS` (for QGIS <= 3.42, this was `OPTIONS`) | [string]<br>기본값: ' ' | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL 드라이버 옵션* 참조) 배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (\|) 로 복수의 옵션들을 분리하십시오. |
| **Additional command-line parameters**<br>선택적 | `EXTRA` | [string]<br>기본값: 설정하지 않음 | 부가 GDAL 명령 줄 옵션을 추가합니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output** | `OUTPUT` | [raster] | 산출 (선명해진) 래스터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `gdal:pansharp`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 래스터 계산기

넘파이 (NumPy) 문법을 가진 명령 줄 래스터 계산기입니다. 넘파이 배열이 지원하는 +, -, * 및 / 같은 모든 기본 산술 연산자는 물론 > 같은 논리 연산자를 사용할 수 있습니다. 모든 입력 래스터는 동일한 차원을 가지고 있어야만 하지만 어떤 투영체도 확인하지 않는다는 점을 기억하십시오.

See the GDAL Raster Calculator utility docs.

**더 보기**

래스터 계산기, 래스터 계산기 *(가상)*

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer A** | `INPUT_A` | [raster] | 첫 번째 입력 래스터 레이어 (필수) |
| **Number of raster band for A** | `BAND_A` | [raster band] | 입력 레이어 A 의 밴드 (필수) |
| **Input layer B** 선택적 | `INPUT_B` | [raster] 기본값: 설정하지 않음 | 두 번째 입력 래스터 레이어 |
| **Number of raster band for B** 선택적 | `BAND_B` | [raster band] | 입력 레이어 B 의 밴드 |
| **Input layer C** 선택적 | `INPUT_C` | [raster] 기본값: 설정하지 않음 | 세 번째 입력 래스터 레이어 |

다음 페이지에 계속

표 24.939 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Number of raster band for C**<br>선택적 | BAND_C | [raster band] | 입력 레이어 C 의 밴드 |
| **Input layer D**<br>선택적 | INPUT_D | [raster]<br>기본값: 설정하지 않음 | 네 번째 입력 래스터 레이어 |
| **Number of raster band for D**<br>선택적 | BAND_D | [raster band] | 입력 레이어 D 의 밴드 |
| **Input layer E**<br>선택적 | INPUT_E | [raster]<br>기본값: 설정하지 않음 | 다섯 번째 입력 래스터 레이어 |
| **Number of raster band for E**<br>선택적 | BAND_E | [raster band] | 입력 레이어 E 의 밴드 |
| **Input layer F**<br>선택적 | INPUT_F | [raster] | 여섯 번째 입력 래스터 레이어 |
| **Number of raster band for F**<br>선택적 | BAND_F | [raster band]<br>기본값: 설정하지 않음 | 입력 레이어 F 의 밴드 |
| **Calculation in gdalnumeric syntax using +-/* or any numpy array functions (i.e. logical_and())** | FORMULA | [string]<br>기본값: ' | 계산 공식입니다. 다음은 몇몇 예시입니다:<br>• `A*(A>0)` —래스터 A 의 값이 0 을 초과하는 경우 래스터 A 의 값을 산출합니다. 0 을 초과하지 않으면, 0 을 산출합니다.<br>• `A*(A>0 and A>B)` —래스터 A 의 값이 0 을 초과하고 래스터 B 의 값을 초과하는 경우 래스터 A 의 값을 산출합니다. 그렇지 않으면, 0 을 산출합니다.<br>• `A*logical_or(A<=177, A>=185)` —래스터 A 의 값이 177 이하 또는 185 이상인 경우 래스터 A 의 값을 산출합니다. 그렇지 않으면, 0 을 산출합니다.<br>• `sqrt(A*A+B*B)` —래스터 A 의 값의 제곱과 래스터 B 의 값의 제곱의 합의 제곱근을 산출합니다. |
| **Set output NoData value**<br>선택적 | NO_DATA | [numeric: double]<br>기본값: 설정하지 않음 | NODATA 용으로 사용할 값 |

다음 페이지에 계속

표 24.939 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Handling of extent differences** | `EXTENT_OPT` | [enumeration] 기본값: 0 | Determines how to handle rasters with different extents. Supported options are:<br>• 0 —무시<br>• 1 —실패<br>• 2 —통합 (union)<br>• 3 —교차 영역 (intersect)<br>**주의**<br>Using this parameter requires QGIS installed with GDAL >= 3.3 (see *Help* *About* menu). |
| **Output extent** 선택적 | `INPUT` | [extent] | Custom extent of the output raster.<br>다음과 같은 메소드를 사용할 수 있습니다:<br>• 레이어로부터 계산…: 현재 프로젝트에 불러온 레이어의 범위를 사용합니다.<br>• 레이어 맵으로부터 계산…: 활성 프로젝트의조판 맵 항목 범위를 사용합니다.<br>• 북마크로부터 계산…: 저장되어 있는북마크 의 범위를 사용합니다.<br>• 맵 캔버스 범위 사용<br>• 캔버스에 그리기: 연산에 넣을 범위를 정하는 직사각형을 클릭 & 드래그합니다.<br>• 좌표를 `xmin, xmax, ymin, ymax` 로 입력<br>**주의**<br>Using this parameter requires QGIS installed with GDAL >= 3.3 (see *Help* *About* menu). |

다음 페이지에 계속

표 24.939 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output raster type** | RTYPE | [enumeration]<br>기본값: 5 | 산출 래스터 파일의 데이터 유형을 정의합니다. 다음 가운데 하나로 지정할 수 있습니다:<br>• 0 —Byte (8 비트 부호 없는 정수 (quint8))<br>• 1 —Int16 (16 비트 부호 있는 정수 (qint16))<br>• 2 —UInt16 (16 비트 부호 없는 정수 (quint16))<br>• 3 —UInt32 (32 비트 부호 없는 정수 (quint32))<br>• 4 —Int32 (32 비트 부호 있는 정수 (qint32))<br>• 5 —Float32 (32 비트 부동소수점형 (float))<br>• 6 —Float64 (64 비트 부동소수점형 (double))<br>• 7 —CInt16 (Complex Int16)<br>• 8 —CInt32 (Complex Int32)<br>• 9 —CFloat32 (Complex Float32)<br>• 10 —CFloat64 (Complex Float64)<br>• 11 —Int8 (8 비트 부호 있는 정수 (qint8))<br>QGIS 와 함께 빌드된 GDAL 버전에 따라 사용할 수 있는 옵션이 달라집니다. (*Help* ⍰ *About* 메뉴 참조) |
| **Calculated** | OUTPUT | [raster]<br>기본값: [Save to temporary file] | Specify the output (calculated) raster layer. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Additional creation options**<br>선택적 | CREATION_OPTIONS (for QGIS <= 3.42, this was OPTIONS) | [string]<br>기본값: ‘’ | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL 드라이버 옵션* 참조)<br>배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (\|) 로 복수의 옵션들을 분리하십시오. |
| **Additional command-line parameters**<br>선택적 | EXTRA | [string]<br>기본값: ‘’ | 부가 GDAL 명령 줄 옵션을 추가합니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Calculated** | `OUTPUT` | [raster] | 산출 (계산된) 래스터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `gdal:rastercalculator`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 래스터 정보

gdalinfo 프로그램이 GDAL 이 지원하는 래스터 데이터셋에 대한 여러 가지 정보를 목록화합니다.

This algorithm is derived from the GDAL info utility.

**기본 메뉴**: *Raster ▸ Miscellaneous*

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [raster] | 입력 래스터 레이어 |
| **Force computation of the actual min/max values for each band** | `MIN_MAX` | [boolean]<br>기본값: False | 데이터셋에 있는 각 밴드에 대해 실제 최소/최대 값을 계산하도록 강제합니다. |
| **Read and display image statistics (force computation if necessary)** | `STATS` | [boolean]<br>기본값: False | 이미지 통계를 읽고 표시합니다. 이미지에 저장된 통계가 없는 경우 강제로 계산합니다. |
| **Suppress GCP info** | `NO_GCP` | [boolean]<br>기본값: False | 지상기준점 (Ground Control Point) 목록을 인쇄하지 않습니다. 이 파라미터는 수천 개의 GCP 를 담고 있는 L1B AVHRR 또는 HDF4 MODIS 처럼 대용량 GCP 를 가지고 있는 데이터셋에 유용할 수도 있습니다. |
| **Suppress metadata info** | `NO_METADATA` | [boolean]<br>기본값: False | 메타데이터를 인쇄하지 않습니다. 일부 데이터셋은 수많은 메타데이터 문자열을 담고 있을 수도 있습니다. |

다음 페이지에 계속

표 24.942 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Layer information** | `OUTPUT` | [html]<br>기본값: `[Save to temporary file]` | Specify the HTML file for output. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Additional command-line parameters**<br>선택적 | `EXTRA` | [string]<br>기본값: 설정하지 않음 | 부가 GDAL 명령 줄 옵션을 추가합니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Layer information** | `OUTPUT` | [html] | 입력 래스터 레이어에 대한 정보를 담고 있는 HTML 파일 |

### 파이썬 코드

**알고리즘 ID**: `gdal:gdalinfo`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 타일 재작업

입력 타일 집합을 재작업 (retile) 합니다. 모든 입력 타일들은 동일한 좌표계를 지리참조해야만 하며, 동일한 개수의 밴드를 가져야만 합니다. 피라미드 수준을 생성할 수 있는 옵션도 있습니다.

This algorithm is derived from the GDAL Retile utility.

### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input files** | `INPUT` | [raster] [list] | 입력 래스터 파일 |
| **Tile width** | `TILE_SIZE_X` | [numeric: integer]<br>기본값: 256 | 타일의 픽셀 단위 너비 (최소값 0) |
| **Tile height** | `TILE_SIZE_Y` | [numeric: integer]<br>기본값: 256 | 타일의 픽셀 단위 높이 (최소값 0) |
| **Overlap in pixels between consecutive tiles** | `OVERLAP` | [numeric: integer]<br>기본값: 0 | |
| **Number of pyramid levels to build** | `LEVELS` | [numeric: integer]<br>기본값: 1 | 최소값: 0 |
| **Output directory** | `OUTPUT` | [folder]<br>기본값: `[Save to temporary folder]` | Specify the output folder for the tiles. *One of*:<br>• 임시 디렉터리에 저장<br>• 디렉터리에 저장 |
| **CSV file containing the tile(s) georeferencing information** | `OUTPUT_CSV` | [file]<br>기본값: `[Skip output]` | Specify the output file for the tiles. *One of*:<br>• 산출물 건너뛰기<br>• 임시 파일로 저장<br>• 파일로 저장… |

#### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Source coordinate reference system**<br>선택적 | `SOURCE_CRS` | [crs]<br>기본값: 설정하지 않음 | |
| **Resampling method** | `RESAMPLING` | [enumeration]<br>기본값: 0 | The resampling algorithm to use. Options:<br>• 0 —최근접 이웃 (`nearest`)<br>• 1 —쌍선형 (2x2 커널) (`bilinear`)<br>• 2 —3 차 (4x4 커널) (`cubic`)<br>• 3 —3 차 B-스플라인 (4x4 커널) (`cubicspline`)<br>• 4 —란초시 (6x6 커널) (`lanczos`) |
| **Column delimiter used in the CSV file**<br>선택적 | `DELIMITER` | [string]<br>기본값: ';' | 타일 (들) 의 지리참조 정보를 담고 있는 CSV 파일에 사용할 구분자 |

다음 페이지에 계속

표 24.946 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Additional creation options**<br>선택적 | CREATION_OPTIONS (for QGIS <= 3.42, this was OPTIONS) | [string]<br>기본값: '' | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL 드라이버 옵션* 참조)<br>배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (\|) 로 복수의 옵션들을 분리하십시오. |
| **Additional command-line parameters**<br>선택적 | EXTRA | [string]<br>기본값: '' | 부가 GDAL 명령 줄 옵션을 추가합니다. |
| **Output data type** | DATA_TYPE | [enumeration]<br>기본값: 5 | 산출 래스터 파일의 포맷을 정의합니다. 다음 가운데 하나로 지정할 수 있습니다:<br>• 0 —Byte (8 비트 부호 없는 정수 (quint8))<br>• 1 —Int16 (16 비트 부호 있는 정수 (qint16))<br>• 2 —UInt16 (16 비트 부호 없는 정수 (quint16))<br>• 3 —UInt32 (32 비트 부호 없는 정수 (quint32))<br>• 4 —Int32 (32 비트 부호 있는 정수 (qint32))<br>• 5 —Float32 (32 비트 부동소수점형 (float))<br>• 6 —Float64 (64 비트 부동소수점형 (double))<br>• 7 —CInt16 (Complex Int16)<br>• 8 —CInt32 (Complex Int32)<br>• 9 —CFloat32 (Complex Float32)<br>• 10 —CFloat64 (Complex Float64)<br>• 11 —Int8 (8 비트 부호 있는 정수 (qint8))<br>QGIS 와 함께 빌드된 GDAL 버전에 따라 사용할 수 있는 옵션이 달라집니다. (*Help → About* 메뉴 참조) |
| **Build only the pyramids** | ONLY_PYRAMIDS | [boolean]<br>기본값: False | |
| **Use separate directory for each tile row** | DIR_FOR_ROW | [boolean]<br>기본값: False | |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output directory** | `OUTPUT` | [folder] | 타일을 저장한 산출물 폴더 |
| **CSV file containing the tile(s) georeferencing information** | `OUTPUT_CSV` | [file] | 타일의 지리참조 정보를 담고 있는 CSV 파일 |

### 파이썬 코드

**알고리즘 ID**: `gdal:retile`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 타일 인덱스

각 입력 래스터 파일의 레코드, 파일명을 담고 있는 속성, 그리고 래스터의 윤곽선을 그리는 폴리곤 도형을 가진 벡터 레이어를 작성합니다. 이 산출물은 MapServer 에서 래스터 타일 인덱스로 사용하기에 적합합니다.

This algorithm is derived from the GDAL Tile Index utility.

**기본 메뉴**: *Raster → Miscellaneous*

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input files** | `LAYERS` | [raster] [list] | 입력 래스터 파일입니다. 파일 여러 개를 지정할 수 있습니다. |
| **Field name to hold the file path to the indexed rasters** | `PATH_FIELD_NAME` Optional | [string] 기본값: 'location' | 인덱스가 추가된 래스터를 가리키는 파일 경로/위치를 담을 산출 필드의 이름입니다. |
| **Store absolute path to the indexed rasters** | `ABSOLUTE_PATH` | [boolean] 기본값: False | 타일 인덱스 파일에 래스터 파일을 가리키는 절대 경로를 저장할지 여부를 설정합니다. 기본적으로 명령 줄에서 지정한 그대로 파일에 래스터 파일명을 넣을 것입니다. |
| **Skip files with different projection reference** | `PROJ_DIFFERENCE` | [boolean] 기본값: False | 타일 인덱스에 이미 삽입된 파일과 동일한 투영체를 가진 파일만 삽입할 것입니다. 기본적으로 투영체를 확인하지 않고 모든 입력물을 받아들입니다. |

다음 페이지에 계속

표 24.948 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Tile index** | `OUTPUT` | [vector: polygon]<br>기본값: `[Save to temporary file]` | Specify the polygon vector layer to write the index to. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Transform geometries to the given CRS**<br>선택적 | `TARGET_CRS` | [crs] | 입력 파일의 도형을 지정한 대상 좌표계로 변형할 것입니다. 기본적으로 입력 래스터와 동일한 좌표계에서 단순한 직사각형 폴리곤을 생성합니다. |
| **The name of the field to store the SRS of each tile**<br>선택적 | `CRS_FIELD_NAME` | [string] | 각 파일의 좌표계를 저장할 필드의 이름입니다. |
| **The format in which the CRS of each tile must be written** | `CRS_FORMAT` | [enumeration]<br>기본값: 0 | 좌표계의 서식을 지정합니다. 다음 가운데 하나를 선택할 수 있습니다:<br>• 0 –자동 (`AUTO`)<br>• 1 –Well-known text (`WKT`)<br>• 2 –EPSG (`EPSG`)<br>• 3 –Proj.4 (`PROJ`) |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Tile index** | `OUTPUT` | [vector: polygon] | 타일 인덱스를 가진 폴리곤 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `gdal:tileindex`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### Viewshed

Calculates a viewshed raster from an input raster DEM using method defined in Wang2000 for a user defined point.

#### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [raster] | 입력 표고 래스터 레이어 |
| **Band number** | `BAND` | [raster band]<br>기본값: 1 | 표고로 사용할 밴드의 번호 |
| **Observer location** | `OBSERVER` | [coordinates] | 관찰자의 위치 |
| **Observer height** | `OBSERVER_HEIGHT` | [numeric: double]<br>기본값: 1.0 | 관찰자의 DEM 단위로 된 고도 |
| **Target height** | `TARGET_HEIGHT` | [numeric: double]<br>기본값: 1.0 | 대상 (target) 요소의 DEM 단위로 된 고도 |
| **Maximum distance from observer to compute visibility** | `MAX_DISTANCE` | [numeric: double]<br>기본값: 100.0 | 관찰자로부터 가시권을 계산할 DEM 단위로 된 최장 거리 |
| **Output** | `OUTPUT` | [raster]<br>기본값: `[Save to temporary file]` | Output raster layer. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

#### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Additional creation options**<br>선택적 | `CREATION_OPTIONS` (for QGIS <= 3.42, this was `OPTIONS`) | [string]<br>기본값: '' | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL* 드라이버 옵션 참조)<br>배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (\|) 로 복수의 옵션들을 분리하십시오. |
| **Additional command-line parameters** | `EXTRA` | [string]<br>기본값: 설정하지 않음 | 부가 GDAL 명령 줄 옵션을 추가합니다. |

#### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output** | OUTPUT | [raster] | 가시권을 표시하는 래스터 레이어입니다 |

#### 파이썬 코드

**알고리즘 ID**: `gdal:viewshed`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 24.2.5 래스터 투영체

#### 투영체 할당하기

래스터 데이터셋에 좌표계를 적용합니다.

This algorithm is derived from the GDAL edit utility.

**기본 메뉴**: *Raster → Projections*

#### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | INPUT_LAYER | [raster] | 입력 래스터 레이어 |
| **Desired CRS** | CRS | [crs] | 산출 레이어의 투영체 (좌표계) |

#### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Layer with projection** | OUTPUT | [raster] | 새 투영체 정보를 가진 산출 래스터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `gdal:assignprojection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 투영체 추출하기

래스터 파일의 투영체를 추출해서 `.wld` 확장자를 가진 월드 파일에 작성합니다.

This algorithm is derived from the GDAL srsinfo utility.

**기본 메뉴**: *Raster → Projections*

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input file** | `INPUT_LAYER` | [raster] | 입력 래스터: 래스터 레이어가 파일 기반이어야 합니다. 이 알고리즘이 래스터 파일을 가리키는 경로를 생성된 `.wld` 파일의 위치로 사용하기 때문입니다. 파일이 아닌 래스터 레이어를 사용하면 오류가 발생할 것입니다. |
| **Create also .prj file** | `PRJ_FILE_CREATE` | [boolean]<br>기본값: False | 이 파라미터를 활성화하면 투영체 정보를 담은 `.prj` 파일도 생성합니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **World file** | `WORLD_FILE` | [file] | 래스터 파일에 대한 변형 (transformation) 파라미터를 담고 있는, `.wld` 확장자를 가진 텍스트 파일입니다. |
| **ESRI Shapefile prj file** | `PRJ_FILE` | [file] | 좌표계를 서술하는, `.prj` 확장자를 가진 텍스트 파일입니다. *Create also .prj file* 파라미터를 거짓으로 설정한 경우 `None` 이 될 것입니다. |

### 파이썬 코드

**알고리즘 ID**: `gdal:extractprojection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 왜곡하기 (재투영)

래스터 레이어를 또다른 좌표계로 재투영합니다. 산출 파일의 해상도 및 리샘플링 메소드를 선택할 수 있습니다.

This algorithm is derived from the GDAL warp utility.

**기본 메뉴**: *Raster → Projections*

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [raster] | 재투영할 입력 래스터 레이어 |
| **Source CRS**<br>선택적 | `SOURCE_CRS` | [crs] | 입력 래스터 레이어의 좌표계를 정의합니다. |
| **Target CRS**<br>선택적 | `TARGET_CRS` | [crs]<br>기본값:<br>`EPSG:4326` | 산출 레이어의 좌표계 |
| **Resampling method to use** | `RESAMPLING` | [enumeration]<br>기본값: 0 | 사용할 픽셀값 리샘플링 메소드입니다. 다음 옵션들 가운데 하나를 선택할 수 있습니다:<br>• 0 — 최근접 이웃<br>• 1 — 쌍선형 (2x2 커널)<br>• 2 — 3 차 (4x4 커널)<br>• 3 — 3 차 B-스플라인 (4x4 커널)<br>• 4 — 란초시 (6x6 커널)<br>• 5 — 평균<br>• 6 — 모드<br>• 7 — 최대값<br>• 8 — 최소값<br>• 9 — 중앙값<br>• 10 — 제 1 사분위수 (Q1)<br>• 11 — 제 3 사분위수 (Q3) |
| **NoData value for output bands**<br>선택적 | `NODATA` | [numeric: double]<br>기본값: 설정하지 않음 | 산출 밴드에 사용할 NODATA 값을 설정합니다. 설정하지 않을 경우, 소스 데이터셋에서 NODATA 값을 복사할 것입니다. |

다음 페이지에 계속

표 24.958 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Output file resolution in target georeferenced units**<br>선택적 | TARGET_RESOLUTION | [numeric: double]<br>기본값: 설정하지 않음 | 재투영한 결과를 저장할 산출 파일의 해상도를 정의합니다. |
| **Reprojected** | OUTPUT | [raster]<br>기본값: [Save to temporary file] | Specification of the output raster layer. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Additional creation options**<br>선택적 | CREATION_OPTIONS (for QGIS <= 3.42, this was OPTIONS) | [string]<br>기본값: '' | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL 드라이버 옵션* 참조)<br>배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (\|) 로 복수의 옵션들을 분리하십시오. |
| **Output data type** | DATA_TYPE | [enumeration]<br>기본값: 0 | 산출 래스터 파일의 포맷을 정의합니다. 다음 가운데 하나로 지정할 수 있습니다:<br>• 0 —입력 레이어의 데이터 유형 사용<br>• 1 —Byte (8 비트 부호 없는 정수 (quint8))<br>• 2 —Int16 (16 비트 부호 있는 정수 (qint16))<br>• 3 —UInt16 (16 비트 부호 없는 정수 (quint16))<br>• 4 —UInt32 (32 비트 부호 없는 정수 (quint32))<br>• 5 —Int32 (32 비트 부호 있는 정수 (qint32))<br>• 6 —Float32 (32 비트 부동소수점형 (float))<br>• 7 —Float64 (64 비트 부동소수점형 (double))<br>• 8 —CInt16 (Complex Int16)<br>• 9 —CInt32 (Complex Int32)<br>• 10 —CFloat32 (Complex Float32)<br>• 11 —CFloat64 (Complex Float64)<br>• 12 —Int8 (8 비트 부호 있는 정수 (qint8))<br>QGIS 와 함께 빌드된 GDAL 버전에 따라 사용할 수 있는 옵션이 달라집니다. (*Help → About* 메뉴 참조) |

다음 페이지에 계속

표 24.959 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Georeferenced extents of output file to be created**<br>선택적 | TARGET_EXTENT | [extent] | 생성할 산출 파일의 지리참조된 범위를 설정합니다. (기본적으로 *Target CRS* 파라미터에서 설정합니다. 이 범위를 설정한 경우 *CRS of the target raster extent* 파라미터에서 설정합니다.)<br>다음과 같은 메소드를 사용할 수 있습니다:<br>• 레이어로부터 계산…: 현재 프로젝트에 불러온 레이어의 범위를 사용합니다.<br>• 레이어 맵으로부터 계산…: 활성 프로젝트의조판 맵 항목 범위를 사용합니다.<br>• 북마크로부터 계산…: 저장되어 있는북마크 의 범위를 사용합니다.<br>• 맵 캔버스 범위 사용<br>• 캔버스에 그리기: 연산에 넣을 범위를 정하는 직사각형을 클릭 & 드래그합니다.<br>• 좌표를 xmin, xmax, ymin, ymax 로 입력 |
| **CRS of the target raster extent**<br>선택적 | TARGET_EXTENT_CRS | [crs] | 산출 파일의 범위로 지정한 좌표를 해석할 좌표계를 지정합니다. 이 파라미터를 산출 데이터셋의 대상 좌표계와 혼동해서는 안 됩니다. 그 대신, 예를 들어 산출물 좌표가 측지 경위도 좌표계라는 것을 알고 있지만, 결과물을 투영된 좌표계로 산출하려는 경우 이 파라미터가 유용할 것입니다. |
| **Use multithreaded warping implementation** | MULTITHREADING | [boolean]<br>기본값: False | 이미지 무리를 처리하고 입력/출력 작업을 동시에 수행하기 위해 스레드 2 개를 사용할 것입니다. 계산 자체는 다중 스레드 작업이 아니라는 사실을 기억하십시오. |
| **Additional command-line parameters**<br>선택적 | EXTRA | [string]<br>기본값: 설정하지 않음 | 부가 GDAL 명령 줄 옵션을 추가합니다. |

## 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Reprojected** | OUTPUT | [raster]<br>기본값: [Save to temporary file] | 재투영된 산출 래스터 레이어 |

#### 파이썬 코드

**알고리즘 ID**: `gdal:warpreproject`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 24.2.6 벡터 변환

#### 포맷 변환하기

OGR 이 지원하는 모든 벡터 레이어를 OGR 이 지원하는 또다른 포맷으로 변환합니다.

This algorithm is derived from the ogr2ogr utility.

#### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: any] | 입력 벡터 레이어 |
| **Convert all layers from dataset** | `CONVERT_ALL_LAYERS` | [boolean] 기본값: False | 데이터셋 전체를 변환합니다. 이 옵션이 지원하는 산출물 포맷은 `GPKG` 와 `GML` 입니다. |
| **Converted** | `OUTPUT` | [same as input] | Specification of the output vector layer. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장…<br>`Save to File` 의 경우, 산출물 포맷을 지정해야 합니다. 모든 GDAL 벡터 포맷을 지원합니다. `Save to a Temporary File` 의 경우, QGIS 기본 벡터 포맷을 사용할 것입니다. |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Additional creation options**<br>선택적 | `CREATION_OPTIONS`<br>(for QGIS <= 3.42, this was `OPTIONS`) | [string]<br>기본값: ''(추가 옵션 없음) | 부가적인 GDAL 생성 옵션 |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Converted** | `OUTPUT` | [same as input] | 산출 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `gdal:convertformat`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 래스터화 (속성으로 덮어쓰기)

래스터 레이어를 벡터 레이어에서 나온 값으로 덮어씁니다. 새 값들은 중첩하는 벡터 피처의 속성값을 바탕으로 할당됩니다.

This algorithm is derived from the GDAL rasterize utility.

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | 입력 벡터 레이어 |
| **Input raster layer** | `INPUT_RASTER` | [raster] | 입력 래스터 레이어 |
| **Field to use for a burn-in value**<br>선택적 | `FIELD` | [tablefield: numeric] | 픽셀 값을 설정하기 위해 사용할 속성 필드를 정의합니다. |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Add burn in values to existing raster values** | ADD | [boolean]<br>기본값: False | 거짓으로 설정하면, 픽셀에 선택한 필드의 값을 할당합니다. 참으로 설정하면, 입력 래스터 레이어의 값에 선택한 필드의 값을 추가합니다. |
| **Additional command-line parameters**<br>선택적 | EXTRA | [string]<br>기본값: '' | 부가 GDAL 명령 줄 옵션을 추가합니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Rasterized** | OUTPUT | [raster] | 덮어쓰기한 입력 래스터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `gdal:rasterize_over`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 래스터화 (고정값으로 덮어쓰기)

래스터 레이어의 일부분을 고정값으로 덮어씁니다. 입력 (중첩하는) 벡터 레이어를 바탕으로 덮어쓸 픽셀을 선택합니다.

This algorithm is derived from the GDAL rasterize utility.

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | INPUT | [vector: geometry] | 입력 벡터 레이어 |
| **Input raster layer** | INPUT_RASTER | [raster] | 입력 래스터 레이어 |
| **A fixed value to burn** | BURN | [numeric: double]<br>기본값: 0.0 | 덮어쓸 값 |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Add burn in values to existing raster values** | ADD | [boolean]<br>기본값: False | 거짓으로 설정하면, 픽셀에 고정값을 할당합니다. 참으로 설정하면, 입력 래스터 레이어의 값에 고정값을 추가합니다. |
| **Additional command-line parameters**<br>선택적 | EXTRA | [string]<br>기본값: '' | 부가 GDAL 명령 줄 옵션을 추가합니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Rasterized** | OUTPUT | [raster] | 덮어쓰기한 입력 래스터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `gdal:rasterize_over_fixed_value`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 래스터화 (벡터를 래스터로)

벡터 도형 (포인트, 라인 및 폴리곤) 을 래스터 이미지로 변환합니다.

This algorithm is derived from the GDAL rasterize utility.

**기본 메뉴**: *Raster → Conversion*

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | INPUT | [vector: geometry] | 입력 벡터 레이어 |
| **Field to use for a burn-in value**<br>선택적 | FIELD | [tablefield: numeric] | 픽셀용 속성을 선택해야 할 속성 필드를 정의합니다. |

다음 페이지에 계속

표 24.970 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **A fixed value to burn**<br>선택적 | BURN | [numeric: double]<br>기본값: 0.0 | 모든 피처의 밴드에 덮어쓸 고정값 |
| **Burn value extracted from the "Z" values of the feature** | USE_Z | [boolean]<br>기본값: False | 피처의"Z"값으로부터 덮어쓸 값을 추출해야 한다는 사실을 나타냅니다. 포인트 및 라인과 (각 선분을 따라 선형 보간) 작동합니다. 폴리곤의 경우, 평면인 경우에만 (모든 꼭짓점의 Z 값이 동일한 경우에만) 제대로 작동합니다. |
| **Output raster size units** | UNITS | [enumeration]<br>기본값: 0 | 산출 래스터의 크기/해상도를 정의할 때 사용할 단위입니다. 다음 가운데 하나를 선택할 수 있습니다:<br>• 0 — 픽셀<br>• 1 — 지리참조된 단위 |
| **Width/Horizontal resolution** | WIDTH | [numeric: double]<br>기본값: 0.0 | 산출 래스터의 (크기 단위가"픽셀"인 경우) 너비를, 또는 (크기 단위가"지리참조된 단위"인 경우) 수평 해상도를 설정합니다. 최소값은 0.0 입니다. |
| **Height/Vertical resolution** | HEIGHT | [numeric: double]<br>기본값: 0.0 | 산출 래스터의 (크기 단위가"픽셀"인 경우) 높이를, 또는 (크기 단위가"지리참조된 단위"인 경우) 수직 해상도를 설정합니다. |
| **Output extent**<br>선택적 | EXTENT | [extent] | 산출 래스터 레이어의 범위입니다. 범위를 지정하지 않은 경우, 선택한 참조 레이어 (들) 을 커버하는 최소 범위를 사용할 것입니다.<br>다음과 같은 메소드를 사용할 수 있습니다:<br>• 레이어로부터 계산…: 현재 프로젝트에 불러온 레이어의 범위를 사용합니다.<br>• 레이어 맵으로부터 계산…: 활성 프로젝트의조판 맵 항목 범위를 사용합니다.<br>• 북마크로부터 계산…: 저장되어 있는북마크 의 범위를 사용합니다.<br>• 맵 캔버스 범위 사용<br>• 캔버스에 그리기: 연산에 넣을 범위를 정하는 직사각형을 클릭 & 드래그합니다.<br>• 좌표를 xmin, xmax, ymin, ymax 로 입력 |
| **Assign a specified NoData value to output bands**<br>선택적 | NODATA | [numeric: double]<br>기본값: 0.0 | 산출 밴드에 지정한 NODATA 값을 할당합니다 |

다음 페이지에 계속

표 24.970 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Rasterized** | `OUTPUT` | [raster]<br>기본값: `[Save to temporary file]` | Specification of the output raster layer. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장…<br>`Save to File` 의 경우, 산출물 포맷을 지정해야 합니다. 모든 GDAL 래스터 포맷을 지원합니다. `Save to a Temporary File` 의 경우, QGIS 기본 래스터 포맷을 사용할 것입니다. |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Additional creation options**<br>선택적 | `CREATION_OPTIONS`<br>(for QGIS <= 3.42, this was `OPTIONS`) | [string]<br>기본값: ' | 래스터를 생성하기 위한 (색상, 블록 크기, 파일 압축 등등) 하나 이상의 생성 옵션을 추가하기 위한 파라미터입니다. 사용자 편의를 위해, 사전 정의 프로파일을 사용할 수 있습니다. (*GDAL 드라이버 옵션* 참조)<br>배치 (batch) 프로세스 및 모델 설계자의 경우: 파이프 문자 (\|) 로 복수의 옵션들을 분리하십시오. |
| **Output data type** | `DATA_TYPE` | [enumeration]<br>기본값: 5 | 산출 래스터 파일의 포맷을 정의합니다.<br>옵션:<br>• 0 —Byte (8 비트 부호 없는 정수 (quint8))<br>• 1 —Int16 (16 비트 부호 있는 정수 (qint16))<br>• 2 —UInt16 (16 비트 부호 없는 정수 (quint16))<br>• 3 —UInt32 (32 비트 부호 없는 정수 (quint32))<br>• 4 —Int32 (32 비트 부호 있는 정수 (qint32))<br>• 5 —Float32 (32 비트 부동소수점형 (float))<br>• 6 —Float64 (64 비트 부동소수점형 (double))<br>• 7 —CInt16 (Complex Int16)<br>• 8 —CInt32 (Complex Int32)<br>• 9 —CFloat32 (Complex Float32)<br>• 10 —CFloat64 (Complex Float64)<br>• 11 —Int8 (8 비트 부호 있는 정수 (qint8))<br>QGIS 와 함께 빌드된 GDAL 버전에 따라 사용할 수 있는 옵션이 달라집니다. (*Help → About* 메뉴 참조) |

다음 페이지에 계속

표 24.971 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Pre-initialize the output image with value**<br>선택적 | INIT | [numeric: double] | 산출 이미지의 밴드를 이 값으로 사전 초기화합니다. 산출물 파일에서 NODATA 값으로 마크하지 않습니다. 모든 밴드에 동일한 값을 사용합니다. |
| **Invert rasterization** | INVERT | [boolean]<br>기본값: False | 고정값을 덮어쓰거나, 또는 이미지에서 지정한 폴리곤 안에 들어가지 않는 모든 부분에 첫 번째 피처 관련 값을 덮어씁니다. |
| **Additional command-line parameters**<br>선택적 | EXTRA | [string]<br>기본값: ' ' | 부가 GDAL 명령 줄 옵션을 추가합니다. |

#### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Rasterized** | OUTPUT | [raster] | 산출 래스터 레이어 |

#### 파이썬 코드

**알고리즘 ID**: `gdal:rasterize`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 24.2.7 벡터 공간 처리

#### 벡터에 버퍼 생성하기

벡터 레이어의 피처 주변에 버퍼를 생성합니다.

#### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | INPUT | [vector: geometry] | 입력 벡터 레이어 |

다음 페이지에 계속

표 24.973 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Geometry column name** | `GEOMETRY` | [string]<br>기본값: 'geometry' | 사용할 입력 레이어 도형 열의 이름 |
| **Buffer distance** | `DISTANCE` | [numeric: double]<br>기본값: 10.0 | 최소값: 0.0 |
| **Dissolve by attribute**<br>선택적 | `FIELD` | [tablefield: any]<br>기본값: 설정하지 않음 | 융해 작업에 사용할 필드 |
| **Dissolve results** | `DISSOLVE` | [boolean]<br>기본값: False | If set, the result is dissolved. If no field is set for dissolving, all the buffers are dissolved into one feature. |
| **Produce one feature for each geometry in any kind of geometry collection in the source file** | `EXPLODE_COLLECTIC` | [boolean]<br>기본값: False | |
| **Buffer** | `OUTPUT` | [vector: polygon]<br>기본값: `[Save to temporary file]` | Specify the output buffer layer. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Additional creation options**<br>선택적 | `CREATION_OPTIONS` (for QGIS <= 3.42, this was `OPTIONS`) | [string]<br>기본값: '(추가 옵션 없음) | 부가적인 GDAL 생성 옵션 |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Buffer** | `OUTPUT` | [vector: polygon] | 산출 버퍼 레이어 |

### 파이썬 코드

**알고리즘 ID**: `gdal:buffervectors`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 벡터를 범위로 잘라내기

OGR 이 지원하는 모든 벡터 파일을 지정한 범위로 잘라냅니다.

This algorithm is derived from the GDAL ogr2ogr utility.

### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | 입력 벡터 레이어 |
| **Clip extent** | `EXTENT` | [extent] | 산출 벡터 파일에 사용되어야 할 경계 상자를 정의합니다. 대상 좌표계의 좌표로 이 경계 상자를 정의해야 합니다.<br>다음과 같은 메소드를 사용할 수 있습니다:<br>• 레이어로부터 계산…: 현재 프로젝트에 불러온 레이어의 범위를 사용합니다.<br>• 레이어 맵으로부터 계산…: 활성 프로젝트의조판 맵 항목 범위를 사용합니다.<br>• 북마크로부터 계산…: 저장되어 있는북마크 의 범위를 사용합니다.<br>• 맵 캔버스 범위 사용<br>• 캔버스에 그리기: 연산에 넣을 범위를 정하는 직사각형을 클릭 & 드래그합니다.<br>• 좌표를 `xmin, xmax, ymin, ymax` 로 입력 |
| **Clipped (extent)** | `OUTPUT` | [same as input]<br>기본값: `[Save to temporary file]` | Specify the output (clipped) layer. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

#### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Additional creation options**<br>선택적 | `CREATION_OPTIONS` (for QGIS <= 3.42, this was `OPTIONS`) | [string]<br>기본값: ˚(추가 옵션 없음) | 부가적인 GDAL 생성 옵션 |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Clipped (extent)** | `OUTPUT` | [same as input] | 산출 (잘라낸) 레이어입니다. 기본 포맷은 "ESRI 셰이프파일"입니다. |

### 파이썬 코드

**알고리즘 ID**: `gdal:clipvectorbyextent`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 벡터를 마스크 레이어로 잘라내기

OGR 이 지원하는 모든 벡터 레이어를 마스크 폴리곤 레이어로 잘라냅니다.

This algorithm is derived from the GDAL ogr2ogr utility.

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | 입력 벡터 레이어 |
| **Mask layer** | `MASK` | [vector: polygon] | 입력 벡터 레이어에 잘라내기 범위로 사용할 레이어 |
| **Clipped (mask)** | `OUTPUT` | [same as input]<br>기본값: `[Save to temporary file]` | The output (masked) layer. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Additional creation options**<br>선택적 | `CREATION_OPTIONS` (for QGIS <= 3.42, this was `OPTIONS`) | [string]<br>기본값: ''(추가 옵션 없음) | 부가적인 GDAL 생성 옵션 |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Clipped (mask)** | `OUTPUT` | [same as input] | 산출 (마스크로 잘라낸) 레이어입니다. 기본 포맷은"ESRI 셰이프파일"입니다. |

### 파이썬 코드

**알고리즘 ID**: `gdal:clipvectorbypolygon`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 융해

지정한 속성/필드에서 동일한 값을 가진 도형을 융해 (결합) 시킵니다. 다중 부분 도형을 산출합니다.

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | 융해시킬 입력 레이어 |
| **Dissolve field** 선택적 | `FIELD` | [tablefield: any] | 융해 작업에 사용할 입력 레이어의 필드 |
| **Geometry column name** | `GEOMETRY` | [string] 기본값: 'geometry' | 융해 작업에 사용할 입력 레이어 도형 열의 이름 |
| **Dissolved** | `OUTPUT` | [same as input] 기본값: `[Save to temporary file]` | Specify the output layer. *One of*: • 임시 파일로 저장 • 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Produce one feature for each geometry in any kind of geometry collection in the source file** | EXPLODE_COLLECTIC | [boolean]<br>기본값: False | 소스 파일에 있는 모든 유형의 도형 집합의 각 도형에 대해 피처 1 개를 생성합니다. |
| **Keep input attributes** | KEEP_ATTRIBUTES | [boolean]<br>기본값: False | 입력 레이어의 모든 속성을 유지합니다. |
| **Count dissolved features** | COUNT_FEATURES | [boolean]<br>기본값: False | 산출 레이어에 융해된 피처의 개수를 포함시킵니다. |
| **Compute area and perimeter of dissolved features** | COMPUTE_AREA | [boolean]<br>기본값: False | Compute the area and perimeter of dissolved features and include them in the output layer |
| **Compute min/max/sum/mean for attribute** | COMPUTE_STATISTICS | [boolean]<br>기본값: False | 지정한 숫자 속성에 대한 통계 (최소값, 최대값, 합계 및 평균) 를 계산해서 산출 레이어에 포함시킵니다. |
| **Numeric attribute to calculate statistics on**<br>선택적 | STATISTICS_ATTRIBUTE | [tablefield: numeric] | 통계를 계산할 숫자 속성 |
| **Additional creation options**<br>선택적 | CREATION_OPTIONS<br>(for QGIS <= 3.42, this was OPTIONS) | [string]<br>기본값: ''(추가 옵션 없음) | 부가적인 GDAL 생성 옵션 |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Dissolved** | OUTPUT | [same as input] | The output multipart geometry layer (with dissolved geometries) |

### 파이썬 코드

**알고리즘 ID**: `gdal:dissolve`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 곡선 오프셋시키기

라인을 지정한 거리로 오프셋시킵니다. 양의 거리는 라인을 왼쪽으로 오프셋시키고, 음의 거리는 오른쪽으로 오프셋기킬 것입니다.

### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: line] | 입력 라인 레이어 |
| **Geometry column name** | `GEOMETRY` | [string]<br>기본값: 'geometry' | 사용할 입력 레이어 도형 열의 이름 |
| **Offset distance (left-sided: positive, right-sided: negative)** | `DISTANCE` | [numeric: double]<br>기본값: 10.0 | |
| **Offset curve** | `OUTPUT` | [vector: line]<br>기본값: `[Save to temporary file]` | Specify the output line layer. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

#### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Additional creation options**<br>선택적 | `CREATION_OPTIONS` (for QGIS <= 3.42, this was `OPTIONS`) | [string]<br>기본값: ''(추가 옵션 없음) | 부가적인 GDAL 생성 옵션 |

#### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Offset curve** | `OUTPUT` | [vector: line] | 산출 오프셋 곡선 레이어 |

### 파이썬 코드

**알고리즘 ID**: `gdal:offsetcurve`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 한쪽 버퍼 생성하기

라인 벡터 레이어에 있는 라인의 한쪽 (오른쪽 또는 왼쪽) 에 버퍼를 생성합니다.

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: line] | 입력 라인 레이어 |
| **Geometry column name** | `GEOMETRY` | [string]<br>기본값: 'geometry' | 사용할 입력 레이어 도형 열의 이름 |
| **Buffer distance** | `DISTANCE` | [numeric: double]<br>기본값: 10.0 | |
| **Buffer side** | `BUFFER_SIDE` | [enumeration]<br>기본값: 0 | 다음 가운데 하나로 지정할 수 있습니다:<br>• 0 — 오른쪽<br>• 1 — 왼쪽 |
| **Dissolve by attribute**<br>선택적 | `FIELD` | [tablefield: any]<br>기본값: 설정하지 않음 | 융해 작업에 사용할 필드 |
| **Dissolve all results** | `DISSOLVE` | [boolean]<br>기본값: False | **이 파라미터를 설정하면 결과물을 융해합니다.**<br>융해 작업을 위한 필드를 설정하지 않는 경우, 모든 버퍼를 하나의 피처로 융해합니다. |
| **Produce one feature for each geometry in any kind of geometry collection in the source file** | `EXPLODE_COLLECTIC` | [boolean]<br>기본값: False | |
| **One-sided buffer** | `OUTPUT` | [vector: polygon]<br>기본값: `[Save to temporary file]` | Specify the output buffer layer. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Additional creation options**<br>선택적 | `CREATION_OPTIONS`<br>(for QGIS <= 3.42, this was `OPTIONS`) | [string]<br>기본값: ''(추가 옵션 없음) | 부가적인 GDAL 생성 옵션 |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **One-sided buffer** | `OUTPUT` | [vector: polygon] | 산출 버퍼 레이어 |

### 파이썬 코드

**알고리즘 ID**: `gdal:onesidebuffer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## 라인을 따라 포인트 생성하기

라인 벡터 레이어의 각 라인 상에서 시작점으로부터의 거리에 포인트를 생성합니다. 이 길이는 라인 길이의 비율 (fraction) 로 정의됩니다.

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: line] | 입력 라인 레이어 |
| **Geometry column name** | `GEOMETRY` | [string]<br>기본값: 'geometry' | 사용할 입력 레이어 도형 열의 이름 |
| **Distance from line start represented as a fraction of line length** | `DISTANCE` | [numeric: double]<br>기본값: 0.5 | |

다음 페이지에 계속

표 24.991 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Points along lines** | `OUTPUT` | [vector: point]<br>기본값: `[Save to temporary file]` | Specify the output point layer. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

#### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Additional creation options**<br>선택적 | `CREATION_OPTIONS` (for QGIS <= 3.42, this was `OPTIONS`) | [string]<br>기본값: ''(추가 옵션 없음) | 부가적인 GDAL 생성 옵션 |

#### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Points along line** | `OUTPUT` | [vector: point] | 산출 포인트 레이어 |

#### 파이썬 코드

**알고리즘 ID**: `gdal:pointsalonglines`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 24.2.8 벡터 기타 알고리즘

#### 가상 벡터 작성하기

Creates a virtual vector layer that contains a set of vector layers.

이 알고리즘은 또다른 알고리즘이 여러 레이어가 필요하지만 레이어를 지정하는 `vrt` 파라미터가 단 하나의 인자만 받는 경우에 특히 유용합니다.

### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input datasources** | `INPUT` | [vector: any] [list] | 가상 벡터를 작성하는 데 사용하려는 벡터 레이어들을 선택합니다. |
| **Create "unioned" VRT** | `UNIONED` | [boolean]<br>기본값: False | 모든 벡터들을 단일 `vrt` 파일로 통합하고자 할 경우 활성화하십시오. |
| **Virtual vector** | `OUTPUT` | [same as input]<br>기본값: `[Save to temporary file]` | Specify the output layer containing only the duplicates. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장… |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Virtual vector** | `OUTPUT` | [vector: any] | 선택한 소스로부터 만든 산출 가상 벡터 |

### 파이썬 코드

**알고리즘 ID**: `gdal:buildvirtualvector`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

## SQL 실행하기

소스 레이어에 `SQL` 문법으로 된 단순 또는 복잡 쿼리를 실행합니다. 쿼리의 결과를 새 레이어로 추가할 것입니다.

This algorithm is derived from the GDAL ogr2ogr utility.

### 파라미터

### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: any] | OGR 이 지원하는 입력 벡터 레이어 |
| **SQL expression** | `SQL` | [string] | SQL 쿼리를 정의합니다. 예: `SELECT * FROM my_table WHERE name is not null` |

다음 페이지에 계속

표 24.996 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **SQL dialect** | `DIALECT` | [enumeration]<br>기본값: 0 | 사용할 SQL 방언을 지정합니다. 다음 가운데 하나를 선택할 수 있습니다:<br>• 0 —없음<br>• 1 —OGR SQL<br>• 2 —SQLite |
| **SQL result** | `OUTPUT` | [vector: any] | Specification of the output layer. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장…<br>`Save to File` 의 경우, 산출물 포맷을 지정해야 합니다. 모든 GDAL 벡터 포맷을 지원합니다. `Save to a Temporary File` 의 경우, 기본 산출 벡터 레이어 포맷을 사용할 것입니다. |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Additional creation options**<br>선택적 | `CREATION_OPTIONS`<br>(for QGIS <= 3.42, this was `OPTIONS`) | [string]<br>기본값: ''(추가 옵션 없음) | 부가적인 GDAL 생성 옵션 |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **SQL result** | `OUTPUT` | [vector: any] | 쿼리가 생성한 벡터 레이어 |

### 파이썬 코드

**알고리즘 ID**: `gdal:executesql`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### PostgreSQL 로 내보내기 (사용할 수 있는 연결)

사용할 수 있는 연결을 기반으로 PostgreSQL 데이터베이스로 벡터 레이어를 가져옵니다. 이 연결은 사전에제대로 정의되어 있어야 합니다. *Save Username* 과 *Save Password* 체크박스가 체크되어 있는지 확인하십시오. 그런 다음에야 이 알고리즘을 사용할 수 있습니다.

This algorithm is derived from the GDAL ogr2ogr utility.

#### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Database (connection name)** | `DATABASE` | [string] | 연결할 PostgreSQL 데이터베이스 |
| **Input layer** | `INPUT` | [vector: any] | 데이터베이스로 내보낼 OGR 이 지원하는 벡터 레이어 |
| **Shape encoding**<br>선택적 | `SHAPE_ENCODING` | [string]<br>기본값: ' ' | 데이터에 적용할 인코딩을 설정합니다. |
| **Output geometry type** | `GTYPE` | [enumeration]<br>기본값: 0 | 산출 도형 유형을 정의합니다. 다음 가운데 하나를 선택할 수 있습니다:<br>• 0 —<br>• 1 —없음<br>• 2 —도형<br>• 3 —포인트<br>• 4 —라인스트링<br>• 5 —폴리곤<br>• 6 —도형집합<br>• 7 —멀티포인트<br>• 8 —멀티폴리곤<br>• 9 —멀티라인스트링<br>• 10 —CIRCULARSTRING<br>• 11 —COUMPOUNDCURVE<br>• 12 —CURVEPOLYGON<br>• 13 —MULTICURVE<br>• 14 —MULTISURFACE<br>• 15 —CONVERT_TO_LINEAR<br>• 16 —CONVERT_TO_CURVE |
| **Assign an output CRS**<br>선택적 | `A_SRS` | [crs]<br>기본값: 설정하지 않음 | 데이터베이스 테이블의 산출 좌표계를 정의합니다. |
| **Reproject to this CRS on output**<br>선택적 | `T_SRS` | [crs]<br>기본값: 설정하지 않음 | 산출물에 이 좌표계로 재투영/변형합니다. |
| **Override source CRS**<br>선택적 | `S_SRS` | [crs]<br>기본값: 설정하지 않음 | 입력 레이어의 좌표계를 무시합니다. |
| **Schema (schema name)**<br>선택적 | `SCHEMA` | [string]<br>기본값: 'public' | 데이터베이스 테이블의 스키마를 정의합니다. |

다음 페이지에 계속

표 24.999 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Table to export to (leave blank to use layer name)**<br>선택적 | TABLE | [string]<br>기본값: '' | 데이터베이스로 가져올 테이블의 이름을 정의합니다. 테이블명은 기본적으로 입력 벡터 파일의 이름입니다. |
| **Primary Key (new field)**<br>선택적 | PK | [string]<br>기본값: 'id' | 데이터베이스 테이블의 기본 키가 될 속성 필드를 정의합니다. |
| **Primary Key (existing field, used if the above option is left empty)**<br>선택적 | PRIMARY_KEY | [tablefield: any]<br>기본값: 설정하지 않음 | 데이터베이스 테이블의 기본 키가 될 내보낸 레이어의 속성 필드를 정의합니다. |
| **Geometry column name**<br>선택적 | GEOCOLUMN | [string]<br>기본값: 'geom' | 도형 정보를 담게 될 데이터베이스의 속성 필드를 정의합니다. |
| **Vector dimensions**<br>선택적 | DIM | [enumeration]<br>기본값: 0 (2 차원) | Defines the coordinate dimensions of the imported vector data. One of:<br>• 0 — 2 (XY)<br>• 1 — 3 (XYZ)<br>• 2 — 4 (XYZM) |
| **Distance tolerance for simplification**<br>선택적 | SIMPLIFY | [string]<br>기본값: '' | 가져올 벡터 도형을 단순화하기 위한 거리 허용 오차를 정의합니다. 기본값은 단순화하지 않는 것입니다. |
| **Maximum distance between 2 nodes (densification)**<br>선택적 | SEGMENTIZE | [string]<br>기본값: '' | 노드 2 개 사이의 최대 거리를 정의합니다. 이 파라미터는 중간 포인트를 생성하기 위해 사용합니다. 기본값은 치밀화하지 않는 것입니다. |
| **Select features by extent (defined in input layer CRS)**<br>선택적 | SPAT | [extent]<br>기본값: 설정하지 않음 | 산출 테이블에 들어갈 지정 범위에서 피처를 선택할 수 있습니다.<br>다음과 같은 메소드를 사용할 수 있습니다:<br>• 레이어로부터 계산…: 현재 프로젝트에 불러온 레이어의 범위를 사용합니다.<br>• 레이어 맵으로부터 계산…: 활성 프로젝트의조판 맵 항목 범위를 사용합니다.<br>• 북마크로부터 계산…: 저장되어 있는북마크 의 범위를 사용합니다.<br>• 맵 캔버스 범위 사용<br>• 캔버스에 그리기: 연산에 넣을 범위를 정하는 직사각형을 클릭 & 드래그합니다.<br>• 좌표를 xmin, xmax, ymin, ymax 로 입력 |
| **Clip the input layer using the above (rectangle) extent** | CLIP | [boolean]<br>기본값: False | 입력 레이어를 사용자가 사전에 정의한 범위로 잘라낼 것입니다. |

다음 페이지에 계속

표 24.999 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Select features using a SQL "WHERE" statement (Ex: column="value")** 선택적 | WHERE | [string] 기본값: '' | 입력 레이어에서 어떤 피처를 선택해야 할지를 SQL "WHERE" 선언문으로 정의합니다. |
| **Group N features per transaction (Default: 2000)** 선택적 | GT | [string] 기본값: '' | N 이 크기를 정의하는 트랜잭션 (transaction) 에서 입력 피처를 그룹화할 수 있습니다. 기본적으로 N 은 트랜잭션 크기를 피처 2 만 개로 제한합니다. |
| **Overwrite existing table** | OVERWRITE | [boolean] 기본값: True | 데이터베이스에 동일한 이름을 가진 테이블이 있는 경우, 그리고 이 옵션을 참으로 설정한 경우, 테이블을 덮어쓸 것입니다. |
| **Append to existing table** | APPEND | [boolean] 기본값: False | 이 파라미터를 참으로 설정하면 기존 테이블에 벡터 데이터를 추가할 것입니다. 입력 레이어에서 검색된 새 필드는 무시합니다. 기본값은 새 테이블을 생성할 것입니다. |
| **Append and add new fields to existing table** | ADDFIELDS | [boolean] 기본값: False | 이 파라미터를 참으로 설정하면 기존 테이블에 벡터 데이터를 추가하고, 새 테이블은 생성하지 않을 것입니다. 입력 레이어에서 검색된 새 필드는 테이블에 추가됩니다. 기본값은 새 테이블을 생성할 것입니다. |
| **Do not launder columns/table names** | LAUNDER | [boolean] 기본값: False | 이 옵션을 참으로 설정하면 (열의 이름을 소문자로 변환한다거나, 공백 및 기타 무결하지 않은 문자를 제거하는 등의) 기본 습성을 막을 수 있습니다. |
| **Do not create Spatial Index** | INDEX | [boolean] 기본값: False | 산출 테이블에 공간 인덱스를 생성하는 일을 막습니다. 기본값은 공간 인덱스를 추가하는 것입니다. |
| **Continue after a failure, skipping the failed feature** | SKIPFAILURES | [boolean] 기본값: False | Continue after a failure, skipping the failed feature. |
| **Validate geometries based on Simple Features specification** | MAKEVALID | [boolean] 기본값: False | Applies cleanup operations on geometries to ensure they are valid regarding the rules of the Simple Features specification. |
| **Promote to Multipart** | PROMOTETOMULTI | [boolean] 기본값: True | 산출 테이블에서 피처의 도형 유형을 다중 부분으로 변환합니다. |
| **Keep width and precision of input attributes** | PRECISION | [boolean] 기본값: True | 열 속성을 입력 데이터를 따르도록 수정하는 일을 막습니다. |
| **Additional creation options** 선택적 | CREATION_OPTIONS (for QGIS <= 3.42, this was OPTIONS) | [string] 기본값: ''(추가 옵션 없음) | 부가적인 GDAL 생성 옵션 |

#### 산출물

이 알고리즘은 아무것도 산출하지 않습니다.

#### 파이썬 코드

**알고리즘 ID**: `gdal:importvectorintopostgisdatabaseavailableconnections`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### PostgreSQL 로 내보내기 (새 연결)

Imports vector layers inside a PostgreSQL database. A new connection to the PostgreSQL database must be created.

This algorithm is derived from the GDAL ogr2ogr utility.

#### 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: any] | 데이터베이스로 내보낼 OGR 이 지원하는 벡터 레이어 |
| **Shape encoding** 선택적 | `SHAPE_ENCODING` | [string] 기본값: | 데이터에 적용할 인코딩을 설정합니다. |
| **Output geometry type** | `GTYPE` | [enumeration] 기본값: 0 | 산출 도형 유형을 정의합니다. 다음 가운데 하나를 선택할 수 있습니다:<br>• 0 — 사용 안 함<br>• 1 — 없음<br>• 2 — 도형<br>• 3 — 포인트<br>• 4 — 라인스트링<br>• 5 — 폴리곤<br>• 6 — 도형집합<br>• 7 — 멀티포인트<br>• 8 — 멀티폴리곤<br>• 9 — 멀티라인스트링<br>• 10 — CIRCULARSTRING<br>• 11 — COUMPOUNDCURVE<br>• 12 — CURVEPOLYGON<br>• 13 — MULTICURVE<br>• 14 — MULTISURFACE<br>• 15 — CONVERT_TO_LINEAR<br>• 16 — CONVERT_TO_CURVE |

다음 페이지에 계속

표 24.1000 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Assign an output CRS**<br>선택적 | A_SRS | [crs]<br>기본값: 설정하지 않음 | 데이터베이스 테이블의 산출 좌표계를 정의합니다. |
| **Reproject to this CRS on output**<br>선택적 | T_SRS | [crs]<br>기본값: 설정하지 않음 | 산출물에 이 좌표계로 재투영/변형합니다. |
| **Override source CRS**<br>선택적 | S_SRS | [crs]<br>기본값: 설정하지 않음 | 입력 레이어의 좌표계를 무시합니다. |
| **Host**<br>선택적 | HOST | [string]<br>기본값: 'localhost' | 데이터베이스의 호스트명 |
| **Port**<br>선택적 | PORT | [string]<br>기본값: '5432' | PostgreSQL 데이터베이스 서버가 열어놓은 포트 번호 |
| **Username**<br>선택적 | USER | [string]<br>기본값: '' | 데이터베이스에 로그인하는 데 쓰이는 사용자명 |
| **Database name**<br>선택적 | DBNAME | [string]<br>기본값: '' | 데이터베이스의 이름 |
| **Password**<br>선택적 | PASSWORD | [string]<br>기본값: '' | 데이터베이스에 연결하기 위해 사용자명과 함께 쓰이는 비밀번호 |
| **Schema (schema name)**<br>선택적 | SCHEMA | [string]<br>기본값: 'public' | 데이터베이스 테이블의 스키마를 정의합니다. |
| **Table name, leave blank to use input name**<br>선택적 | TABLE | [string]<br>기본값: '' | 데이터베이스로 가져올 테이블의 이름을 정의합니다. 테이블명은 기본적으로 입력 벡터 파일의 이름입니다. |
| **Primary Key (new field)**<br>선택적 | PK | [string]<br>기본값: 'id' | 데이터베이스 테이블의 기본 키가 될 속성 필드를 정의합니다. |
| **Primary Key (existing field, used if the above option is left empty)**<br>선택적 | PRIMARY_KEY | [tablefield: any]<br>기본값: 설정하지 않음 | 데이터베이스 테이블의 기본 키가 될 내보낸 레이어의 속성 필드를 정의합니다. |
| **Geometry column name**<br>선택적 | GEOCOLUMN | [string]<br>기본값: 'geom' | 도형 정보를 담게 될 데이터베이스의 속성 필드를 정의합니다. |
| **Vector dimensions**<br>선택적 | DIM | [enumeration]<br>기본값: 0 (2 차원) | Defines the coordinate dimensions of the imported vector data. One of:<br>• 0 —2 (XY)<br>• 1 —3 (XYZ)<br>• 2 —4 (XYZM) |
| **Distance tolerance for simplification**<br>선택적 | SIMPLIFY | [string]<br>기본값: '' | 가져올 벡터 도형을 단순화하기 위한 거리 허용 오차를 정의합니다. 기본값은 단순화하지 않는 것입니다. |
| **Maximum distance between 2 nodes (densification)**<br>선택적 | SEGMENTIZE | [string]<br>기본값: '' | 노드 2 개 사이의 최대 거리를 정의합니다. 이 파라미터는 중간 포인트를 생성하기 위해 사용합니다. 기본값은 치밀화하지 않는 것입니다. |

다음 페이지에 계속

표 24.1000 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Select features by extent (defined in input layer CRS)**<br>선택적 | SPAT | [extent]<br>기본값: 설정하지 않음 | 산출 테이블에 들어갈 지정 범위에서 피처를 선택할 수 있습니다.<br>다음과 같은 메소드를 사용할 수 있습니다:<br>• 레이어로부터 계산…: 현재 프로젝트에 불러온 레이어의 범위를 사용합니다.<br>• 레이어 맵으로부터 계산…: 활성 프로젝트의조판 맵 항목 범위를 사용합니다.<br>• 북마크로부터 계산…: 저장되어 있는북마크 의 범위를 사용합니다.<br>• 맵 캔버스 범위 사용<br>• 캔버스에 그리기: 연산에 넣을 범위를 정하는 직사각형을 클릭 & 드래그합니다.<br>• 좌표를 xmin, xmax, ymin, ymax 로 입력 |
| **Clip the input layer using the above (rectangle) extent** | CLIP | [boolean]<br>기본값: False | 입력 레이어를 사용자가 사전에 정의한 범위로 잘라낼 것입니다. |
| **Fields to include (leave empty to use all fields)**<br>선택적 | FIELDS | [string] [list]<br>기본값: [] | 가져온 벡터 파일에서 유지할 필드를 정의합니다. 아무것도 선택하지 않으면 모든 필드를 가져옵니다. |
| **Select features using a SQL "WHERE" statement (Ex: column="value")**<br>선택적 | WHERE | [string]<br>기본값: '' | 산출 레이어를 위해 어떤 피처를 선택해야 할지를 SQL "WHERE"선언문으로 정의합니다. |
| **Group N features per transaction (Default: 2000)**<br>선택적 | GT | [string]<br>기본값: '' | N 이 크기를 정의하는 트랜잭션 (transaction) 에서 입력 피처를 그룹화할 수 있습니다. 기본적으로 N 은 트랜잭션 크기를 피처 2 만 개로 제한합니다. |
| **Overwrite existing table** | OVERWRITE | [boolean]<br>기본값: True | 데이터베이스에 동일한 이름을 가진 테이블이 있는 경우, 그리고 이 옵션을 참으로 설정한 경우, 테이블을 덮어쓸 것입니다. |
| **Append to existing table** | APPEND | [boolean]<br>기본값: False | 이 파라미터를 참으로 설정하면 기존 테이블에 벡터 데이터를 추가할 것입니다. 입력 레이어에서 검색된 새 필드는 무시합니다. 기본값은 새 테이블을 생성할 것입니다. |

다음 페이지에 계속

표 24.1000 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Append and add new fields to existing table** | ADDFIELDS | [boolean]<br>기본값: False | 이 파라미터를 참으로 설정하면 기존 테이블에 벡터 데이터를 추가하고, 새 테이블은 생성하지 않을 것입니다. 입력 레이어에서 검색된 새 필드는 테이블에 추가됩니다. 기본값은 새 테이블을 생성할 것입니다. |
| **Do not launder columns/table names** | LAUNDER | [boolean]<br>기본값: False | 이 옵션을 참으로 설정하면 (열의 이름을 소문자로 변환한다거나, 공백 및 기타 무결하지 않은 문자를 제거하는 등의) 기본 습성을 막을 수 있습니다. |
| **Do not create Spatial Index** | INDEX | [boolean]<br>기본값: False | 산출 테이블에 공간 인덱스를 생성하는 일을 막습니다. 기본값은 공간 인덱스를 추가하는 것입니다. |
| **Continue after a failure, skipping the failed feature** | SKIPFAILURES | [boolean]<br>기본값: False | Continue after a failure, skipping the failed feature. |
| **Validate geometries based on Simple Features specification** | MAKEVALID | [boolean]<br>기본값: False | Applies cleanup operations on geometries to ensure they are valid regarding the rules of the Simple Features specification. |
| **Promote to Multipart** | PROMOTETOMULTI | [boolean]<br>기본값: True | 산출 테이블에서 피처의 도형 유형을 다중 부분으로 변환합니다. |
| **Keep width and precision of input attributes** | PRECISION | [boolean]<br>기본값: True | 열 속성을 입력 데이터를 따르도록 수정하는 일을 막습니다. |
| **Additional creation options**<br>선택적 | CREATION_OPTIONS<br>(for QGIS <= 3.42, this was OPTIONS) | [string]<br>기본값: ''(추가 옵션 없음) | 부가적인 GDAL 생성 옵션 |

### 산출물

이 알고리즘은 아무것도 산출하지 않습니다.

### 파이썬 코드

**알고리즘 ID**: `gdal:importvectorintopostgisdatabasenewconnection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### 벡터 정보

Creates an information file that lists information about an OGR-supported data source. The output will be shown in a 'Result' window and can be written into a HTML-file. The information includes the geometry type, feature count, the spatial extent, the projection information, the list of attributes and related type, list of relations and field domains, and many more.

This algorithm is derived from the GDAL ogrinfo utility.

### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | 입력 벡터 레이어 |
| **Enable listing of all layers in the dataset** | `ALL_LAYERS` | [boolean]<br>기본값: False | If checked, QGIS will output the information of all the layers in the input dataset instead of information of the first layer only. Convenient when a multi layer format (GeoPackage, GML, KML, SpatiaLite, SQLite, ⋯) is used as input. |
| **Summary output only** | `SUMMARY_ONLY` | [boolean]<br>기본값: True | Suppress listing of individual features and show only summary information like projection, schema, feature count and extents. |
| **Suppress metadata info** | `NO_METADATA` | [boolean]<br>기본값: False | Suppress metadata printing. Some datasets may contain a lot of metadata strings. |
| **Layer information** | `OUTPUT` | [html]<br>기본값: `[Save to temporary file]` | Specify the output HTML file that includes the file information. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장⋯<br>HTML 파일을 지정하지 않으면 산출물을 임시 파일에 작성할 것입니다. |

#### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Additional command line parameters**<br>선택적 | `EXTRA` | [string]<br>기본값: '(추가 옵션 없음) | 부가 GDAL 명령 줄 옵션을 추가합니다. |

#### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Layer information** | `OUTPUT` | [html] | 파일 정보를 담고 있는 산출 HTML 파일 |

#### 파이썬 코드

**알고리즘 ID**: `gdal:ogrinfo`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

### Vector Information (JSON)

Creates an information file that lists information about an OGR-supported data source. The output will be shown in a 'Result' window and can be written into a `.JSON` file. The information includes the geometry type, feature count, the spatial extent, the projection information, the list of attributes and related type, list of relations and field domains, and many more.

This algorithm is derived from the GDAL ogrinfo utility.

> **주의**
>
> Running this algorithm requires QGIS installed with GDAL >= 3.7 (see *Help → About* menu).

#### 파라미터

#### 기본 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Input layer** | `INPUT` | [vector: geometry] | 입력 벡터 레이어 |
| **Enable listing of all layers in the dataset** | `ALL_LAYERS` | [boolean]<br>기본값: False | If checked, QGIS will output the information of all the layers in the input dataset instead of information of the first layer only. Convenient when a multi layer format (GeoPackage, GML, KML, SpatiaLite, SQLite, ⋯) is used as input. |
| **Enable listing of features** | `FEATURES` | [boolean]<br>기본값: False | |
| **Suppress metadata info** | `NO_METADATA` | [boolean]<br>기본값: False | Suppress metadata printing. Some datasets may contain a lot of metadata strings. |

다음 페이지에 계속

표 24.1004 – 이전 페이지에서 계속

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Layer information** | `OUTPUT` | [file]<br>기본값: `[Save to temporary file]` | Specify the output JSON file that includes the file information. *One of*:<br>• 임시 파일로 저장<br>• 파일로 저장…<br>If no JSON file is defined the output will be written to a temporary file |

### 고급 파라미터

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Additional command line parameters**<br>선택적 | `EXTRA` | [string]<br>기본값: ''(추가 옵션 없음) | 부가 GDAL 명령 줄 옵션을 추가합니다. |

### 산출물

| 라벨 | 이름 | 유형 | 설명 |
|---|---|---|---|
| **Layer information** | `OUTPUT` | [file] | The output *.JSON* file that includes the file information. |

### 파이썬 코드

**Algorithm ID**: `gdal:ogrinfojson`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **이름** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

CHAPTER 25

# 플러그인

## 25.1 QGIS 플러그인

QGIS 는 플러그인 아키텍처로 설계됐습니다. 즉 응용 프로그램에 새로운 여러 기능 및 함수들을 쉽게 추가할 수 있다는 의미입니다. QGIS 에서 일부 기능들은 실제로 플러그인으로 실행되고 있습니다.

### 25.1.1 핵심 및 외부 플러그인

QGIS 플러그인은 **핵심 플러그인** 또는 **외부 플러그인** 둘 중 하나로 실행됩니다.

핵심 플러그인 은 QGIS 개발팀이 관리하며, 모든 QGIS 배포판에 자동적으로 포함됩니다. 핵심 플러그인은 **C++** 또는 **파이썬** 두 언어 가운데 하나로 작성됩니다.

외부 플러그인 대부분은 현재 파이썬으로 작성됩니다. 외부 플러그인은'공식'QGIS 저장소 https://plugins.qgis.org/plugins/ 에 저장되거나 또는 개인 작성자들이 관리하는 외부 저장소에 저장됩니다. 공식 저장소는 플러그인 사용법, 최저 QGIS 버전, 홈페이지, 작성자 및 기타 중요 정보에 대한 상세 문서를 제공합니다. 기타 외부 저장소의 경우, 외부 플러그인 자체에 문서가 포함돼 있을 수도 있습니다. QGIS 사용자 지침서는 외부 플러그인 문서를 포함하고 있지 않습니다.

플러그인을 설치하거나 활성화하려면, *Plugins* 메뉴에서 *Manage and install plugins*…항목을 클릭해서 플러그인 관리자 대화창을 여십시오. 외부 파이썬 플러그인은 활성화된사용자 프로파일 경로의 `python/plugins` 폴더 아래 설치됩니다.

*Settings* ▸ *Options* ▸ *System* 메뉴를 통해 사용자 지정 C++ 플러그인 라이브러리를 가리키는 경로를 추가할 수 있습니다.

### 25.1.2 플러그인 대화창

#### 설정 탭

사용자는 왼쪽 패널 하단에 있는  *Settings* 탭에서 사용자 응용 프로그램에 어떤 플러그인을 표시할 수 있는지 환경 설정할 수 있습니다. 다음 옵션들을 사용할 수 있습니다:

-  *Check for Updates on Startup*. QGIS will automatically check for plugin updates at startup, at most once every 3 days. Whenever an installed plugin has an update available, QGIS will inform you.
-  *Show also Experimental Plugins*: QGIS 가 일반적으로 작업 활용에 부적합한, 개발 초반에 있는 플러그인을 표시할 것입니다. 이런 플러그인의 경우 안정 버전 또는 실험적 버전 가운데 하나를 설치할 수 있고, 어느 때든 두 버전 간에 전환할 수 있습니다.
-  *Show also Deprecated Plugins*: QGIS 에 대체 기능이 있기 때문에, 유지/관리가 안 되고 있기 때문에, 더 이상 QGIS 에서 사용할 수 없는 기능에 의존하고 있기 때문에 등등의 이유로 보통 유지/관리되고 있지 않은 플러그인을 표시합니다. 이런 플러그인은 일반적으로 작업 활용에 부적합하므로 플러그인 목록에 회색으로 나타납니다.

QGIS 는 기본적으로 *Plugin Repositories* 부분에 `https://plugins.qgis.org/plugins/plugins.xml?qgis=version` 이라는 URL 을 가진 공식 플러그인 저장소를 제공합니다. (이때 `<version>` 은 사용자가 실행 중인 QGIS 의 정확한 버전을 나타냅니다.) 외부 작성자 저장소를 추가하려면,  *Add…*버튼을 클릭하고 *Repository Details* 양식에 이름과 URL 을 입력하십시오. 이때 URL 은 `http://` 또는 `file://` 프로토콜 유형일 수 있습니다.

기본 QGIS 저장소는 공개 저장소로 접근하는 데 어떤 인증도 필요하지 않습니다. 하지만 사용자가 자신의 플러그인 저장소를 열어서 인증을 (기본 인증, PKI 등을) 요구할 수는 있습니다. 인증 장에서 QGIS 가 지원하는 인증 방법에 대한 자세한 정보를 얻을 수 있습니다.

추가한 저장소 가운데 하나 이상을 더 이상 원하지 않을 경우, 설정 탭에서  *Edit…*버튼으로 비활성화시키거나 또는  *Delete* 버튼으로 완전히 삭제할 수 있습니다.

#### 플러그인 탐색하기

#### 탭

*Plugins* 대화창 위쪽에 있는 탭들은 플러그인의 설치, 생성, 또는 업데이트 상태를 바탕으로 하는 플러그인 목록을 제공합니다. 플러그인의 설정에 따라 다음 탭들을 사용할 수 있습니다:

-  *All*: 활성화된 저장소에서 사용할 수 있는 플러그인을 모두 표시합니다.
-  *Installed*: 사용자가 설치한 플러그인과 사용자가 설치 해제할 수 없는, 기본적으로 설치된 핵심 플러그인을 둘 다 표시합니다.
-  *Not installed*: 활성화된 저장소에 설치 해제된 또는 아직 설치되지 않은 플러그인을 표시합니다.
-  *New*: 마지막으로 확인한 (*Check for Updates on Startup*) 이후 배포된 플러그인을 표시합니다.
-  *Upgradeable*: 설치된 플러그인 가운데 저장소에 좀 더 최신 버전이 게시된 플러그인을 표시합니다.
-  *Invalid*: 설치된 플러그인 가운데 (의존성 누락, 불러오는 도중 오류 발생, QGIS 버전과 호환되지 않는 기능 등등) 어떤 이유로든 현재 작동하지 않는 플러그인을 모두 표시합니다.

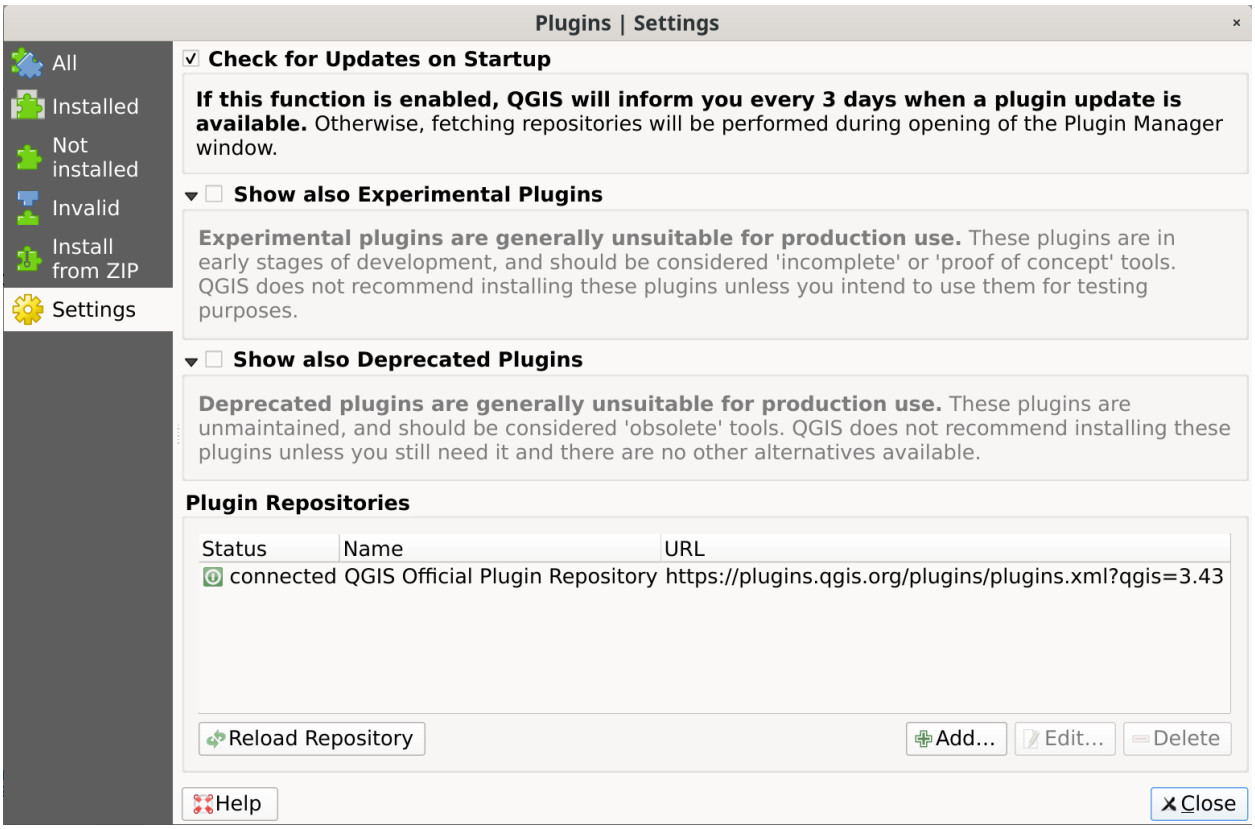



그림 25.1: *Settings* 탭

이 탭들의 최상단에 있는 *Search* 기능을 통해 (작성자, 이름, 설명, 태그 등등의) 메타데이터 정보를 이용해서 어떤 플러그인이든 찾을 수 있습니다.

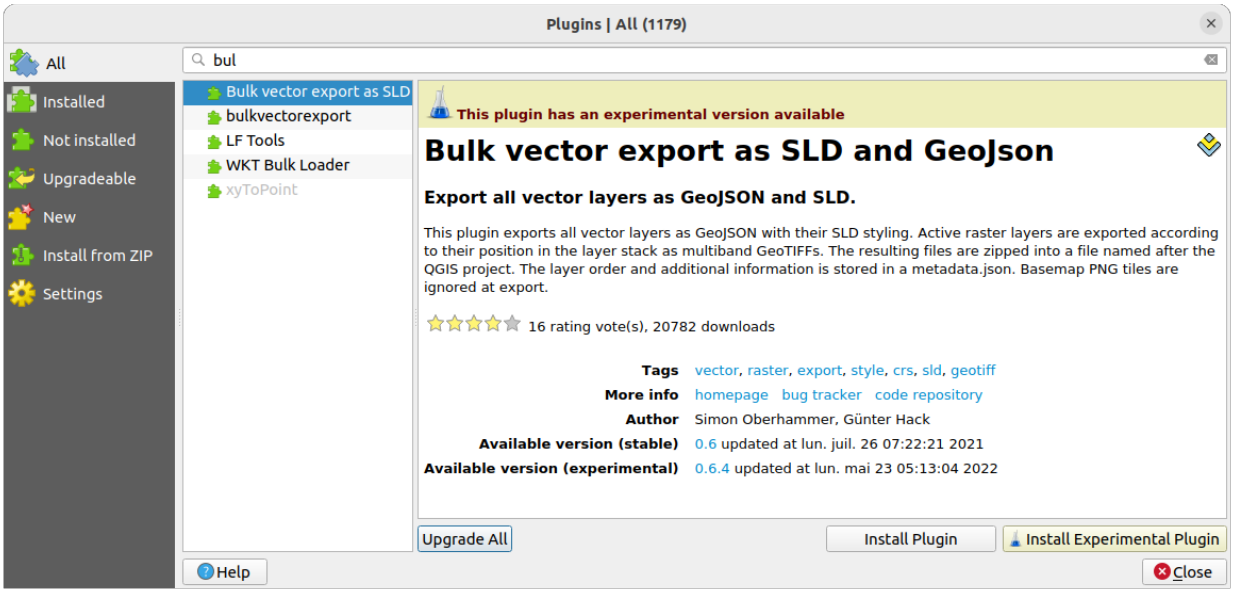



그림 25.2: *All* 탭에서 플러그인 검색하기

### 플러그인

플러그인을 선택하면 오른쪽 패널에 몇몇 메타데이터가 표시될 것입니다:

- (*Show also Experimental Plugins* 옵션을 체크한 경우) 플러그인이 실험적 버전인지 또는 실험적 버전을 사용할 수 있는지에 대한 정보
- 요약 및 설명
- 평점 (사용자가 선호하는 플러그인을 평가할 수 있습니다!)
- tags
- 홈페이지, 트래커 및 코드 저장소를 가리키는 유용한 몇몇 링크
- 작성자 (들)
- 저장소의 다운로드 페이지를 가리키는 링크를 가진, 사용할 수 있는 버전 (들) 또는 설치된 플러그인의 로컬 폴더를 가리키는 경로

*Plugin Manager* 대화창에서 플러그인의 최신 버전을 쌍방향으로 작업할 수 있습니다. 실험적 버전 사용을 활성화한 경우, 실험적 버전이 최신 안정 버전보다 더 최신인 경우에만 표시될 것입니다. 어떤 탭이 활성화되어 있는지 그리고 선택한 플러그인이 설치되어 있는지에 따라, 다음 옵션들 가운데 일부를 선택할 수 있게 될 것입니다:

- *Install*: 선택한 플러그인의 최신 안정 버전을 설치합니다.
- *Install Experimental Plugin*: 선택한 플러그인의 실험적 버전을 설치합니다.
- *Reinstall Plugin*: 예를 들어 어떤 플러그인의 불러오기가 실패한 경우, 해당 플러그인과 동일한 안정 버전을 설치합니다.
- *Reinstall Experimental Plugin*: installs the same experimental version of the plugin e.g. after it has failed to load
- *Upgrade Plugin*: 선택한 플러그인을 최신 안정 버전으로 업그레이드합니다.

- *Upgrade Experimental Plugin*: 선택한 플러그인을 최신 실험적 버전으로 업그레이드합니다.
- *Upgrade All*: 설치된 모든 플러그인을 (이전에 설치된 플러그인이 안정 버전인지 실험적 버전인지에 따라) 최신 안정 버전 또는 실험적 버전으로 업그레이드합니다.
- *Downgrade Plugin*: 실험적 버전의 플러그인을 이전 안정 버전으로 다운그레이드합니다.
- *Downgrade Experimental Plugin*: 실험적 버전의 플러그인을 가장 마지막으로 게시된 실험적 버전으로 다운그레이드합니다. 아직 게시되지 않은 버전을 테스트하는 경우 이 옵션이 필요할 수 있습니다.
- *Uninstall Plugin*: 사용자 프로파일에서 설치된 플러그인을 제거합니다.

설치된 플러그인 왼쪽에는 체크박스가 표시됩니다. 이 체크박스를 체크 해제하면 플러그인이 일시적으로 비활성화됩니다.

목록에 있는 플러그인을 오른쪽 클릭하면 플러그인 목록을 다양한 메타데이터를 기준으로 정렬시킬 수 있습니다. 새로 정렬된 순서는 모든 탭에 적용됩니다. 다음과 같은 정렬 옵션을 선택할 수 있습니다:

- *Sort by Name*: 이름으로 정렬
- *Sort by Downloads*: 다운로드로 정렬
- *Sort by Vote*: 평점으로 정렬
- *Sort by Status*: 설치 및 업그레이드 상태로 정렬
- *Sort by Date Created*: 생성 날짜로 정렬
- *Sort by Date Updated*: 업데이트 날짜로 정렬

### ZIP 탭에서 설치하기

*Install from ZIP* 탭은 ZIP 으로 압축된 포맷에서 플러그인을 가져올 수 있는, 예를 들어 저장소에서 플러그인을 직접 다운로드할 수 있는 파일 선택기 위젯을 제공합니다. 암호화 파일을 지원합니다.

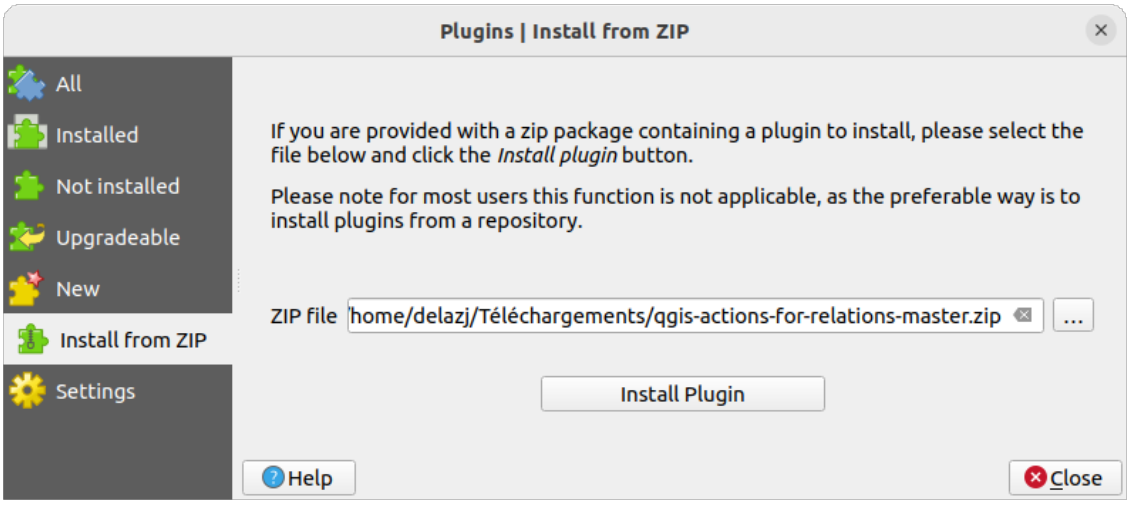



그림 25.3: *Install from zip* 탭

## 25.2 QGIS 핵심 플러그인 사용하기

### 25.2.1 데이터베이스 관리자 플러그인

The DB Manager Plugin is intended to be the main tool to integrate and manage spatial database formats supported by QGIS (PostgreSQL, SpatiaLite, GeoPackage, Oracle Spatial, Virtual layers) in one user interface. The DB Manager Plugin provides several features. You can drag layers from the QGIS Browser into the DB Manager, and it will import your layer into your spatial database. You can drag and drop tables between spatial databases and they will get imported.

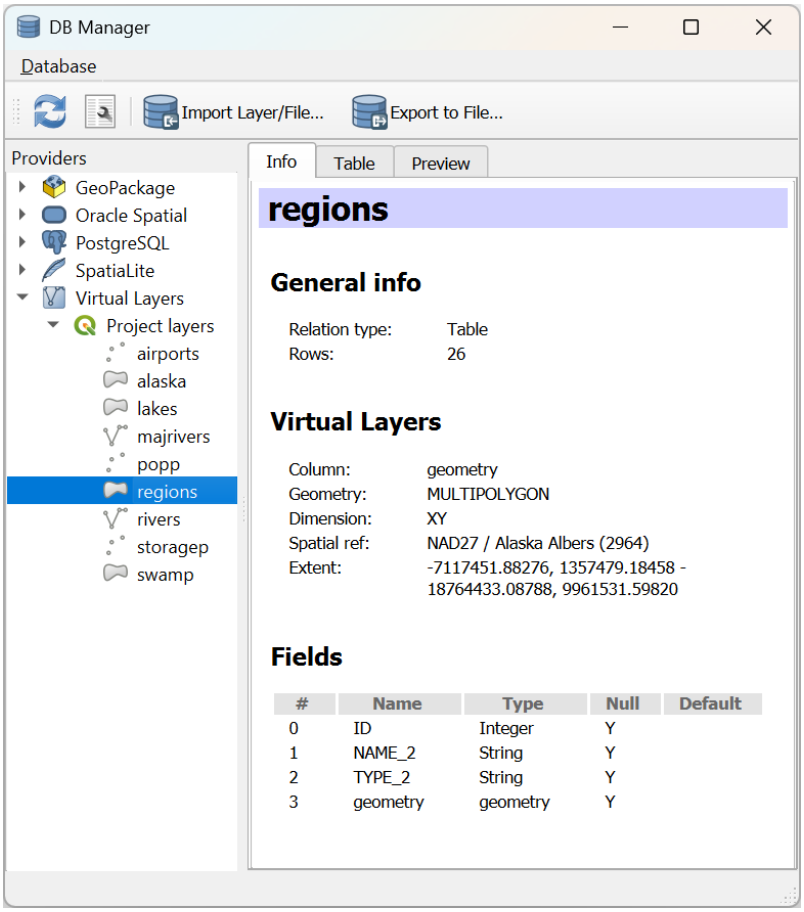



그림 25.4: 데이터베이스 관리자 대화창

The *Database* menu allows you to connect to an existing database, to start the SQL window and to exit the DB Manager Plugin. Once you are connected to an existing database, the menus *Schema* (relevant for DBMSs, such as PostgreSQL) and *Table* will appear.

*Schema* 메뉴는 스키마를 생성하고 (비어 있는 경우에만) 삭제할 수 있는 도구를 포함하고 있습니다. 그리고 (예를 들어 PostGIS 토폴로지 확장 기능처럼) 데이터베이스가 위상을 지원하는 경우, *TopoViewer* 를 구동할 수 있습니다.

*Table* 메뉴를 통해 테이블을 생성/편집할 수 있고 테이블 및 뷰를 삭제할 수 있습니다. 테이블의 내용을 완전히 지우거나 어떤 스키마에서 다른 스키마로 테이블을 옮길 수도 있습니다. *Run Vacuum Analyze* 로 선택한 테이블에 빈공간 분석을 수행할 수 있습니다. 빈공간 *(vacuum)* 은 공간을 확보하고 재사용할 수 있도록 해주며, 분석 *(analyze)* 은 쿼리를 가장 효율적으로 실행하는 방식을 결정하기 위해 사용되는 통계를 업데이트합니다. *Change Log-*

*ging*…으로는 테이블에 변경 사항 로그 작업 지원을 추가할 수 있습니다. 마지막으로, *Import Layer/File*…메뉴로 레이어/파일을 가져올 수 있고 *Export to File*…메뉴로 파일로 내보낼 수 있습니다.

> **참고**
>
> DB 관리자를 이용해서 PostgreSQL 데이터베이스의 테이블 및 열에 주석을 추가할 수 있습니다.

*Providers* 창은 QGIS 가 지원하는 모든 기존 데이터베이스의 목록을 담고 있습니다. 어느 항목을 더블클릭하면 해당 데이터베이스에 연결할 수 있습니다. 어느 항목을 오른쪽 클릭하면 기존 스키마 및 테이블을 재명명하거나 삭제할 수 있습니다. 컨텍스트 메뉴를 통해 QGIS 캔버스에 테이블도 추가할 수 있습니다.

데이터베이스에 연결돼 있는 경우, DB 관리자의 **메인** 창에서 탭 4 개를 사용할 수 있습니다. *Info* 탭은 테이블 및 그 도형은 물론 기존 필드, 제약 조건 및 인덱스에 대한 정보를 제공합니다. 이 탭에서 선택한 테이블에 공간 인덱스를 생성할 수 있습니다. *Table* 탭은 테이블을 표시하고, *Preview* 탭은 도형을 미리보기로 렌더링합니다. *SQL Window* 를 열면, SQL 창을 새 탭에 배치할 것입니다.

### SQL 창에서 작업하기

DB 관리자를 통해 사용자의 공간 데이터베이스에 SQL 쿼리를 실행할 수 있습니다. 쿼리를 저장하고 불러올 수 있으며, 사용자 쿼리를 공식으로 표현하게 도와주는 *SQL Query Builder* 도 있습니다. *Load as new layer* 옵션을 활성화하고 *Column(s) with unique values* (IDs), *Geometry column* 및 *Layer name (prefix)* 옵션을 지정해주면 공간 산출물도 살펴볼 수 있습니다. SQL 의 일부분을 강조시킨 다음 `Ctrl+R` 조합 키 또는 *Execute* 버튼을 클릭하면 해당 부분만 실행하도록 할 수 있습니다.

QGIS also adds support for the `REGEXP` function in some providers. This allows users to use regular expressions in SQL filters or expressions, for example:

```
SELECT * FROM places WHERE name REGEXP '^A';
```

This returns all features where the `name` field starts with the letter A.

After executing your query, you can select specific cells in the result set. Use the `Ctrl+C` shortcut to copy the selected cells to the clipboard. The copied data is available as a formatted table. This allows you to paste the data into other applications, such as spreadsheet where it will show up as a table.

*Query History* 버튼을 클릭하면 각 데이터베이스 및 제공자의 최신 쿼리 20 개를 저장합니다.

항목을 더블 클릭하면 SQL 창에 문자열을 추가할 것입니다.

> **참고**
>
> 가상 레이어를 생성하기 위해 SQL 창을 사용할 수도 있습니다. 이런 경우 SQL 창을 열기 전에 데이터베이스를 선택하는 대신, **Virtual Layers** 아래 있는 **QGIS Layers** 를 선택하십시오. 사용할 SQL 문법은가상 레이어 생성 의 지침을 참조하십시오.

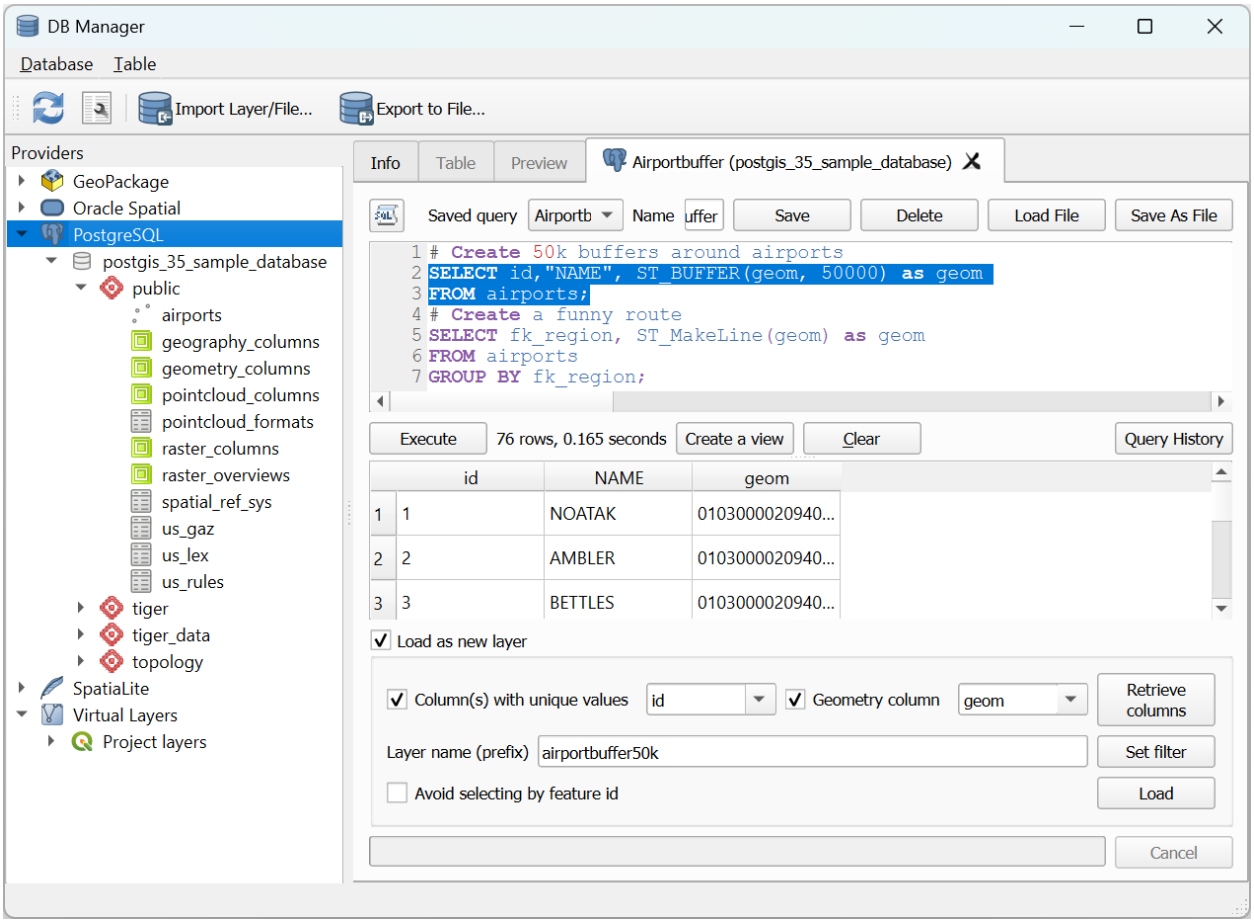



그림 25.5: 데이터베이스 관리자 SQL 창에서 SQL 쿼리 실행하기

### Import Vector Layer

You can import layer or file into your database. Here are the parameters you can set for the import process:

- *Input*: Select the layer or file to import. Using the dropdown menu, select from the list of loaded layers in QGIS or click on the ⋯ button to select a file from disk. Check the *Import only selected features* to import only the selected features of the layer.
- *Output table*: Choose the *Schema* and provide a name for the new table.
- *Options*: Here are some options for the import process:
  - *Primary key*: Provide naming for the primary key field. By default, it is named `id`.
  - *Geometry column*: Provide naming for the geometry column. By default, it is named `geom`.
  - *Source SRID*: Define the SRID for the geometry column. By default, it uses the layer's CRS.
  - *Target SRID*: Define the target SRID to reproject the geometries during the import process. By default, it uses the layer's CRS.
  - *Encoding*: Define the encoding of the source data. By default, it uses `Automatic`. It is QGIS's automatic detection mode that attempts to guess the file's character encoding based on available metadata or system locale.
  - *Replace destination table (if exists)*: If a table with the same name already exists in the selected schema, it will be replaced.
  - *Do not promote to multi-part*: Geometries will be imported as single-part geometries.
  - *Convert field names to lower case*: All field names will be converted to lower case.
  - *Create spatial index*: A spatial index will be created on the geometry column after import.
  - *Comment*: Add comments to table. Only available for PostgreSQL databases.

### Export to Vector File

To export a table from your database to a vector file, select the desired *Format* and *Save as* location. Under *Options*, you can set the following parameters:

- *Source SRID*: Define the source SRID of the geometry column. By default, it uses the layer's CRS.
- *Target SRID*: Define the target SRID to reproject the geometries during the export process. By default, it uses the layer's CRS.
- *Encoding*: Define the encoding of the output data.
- *Replace destination table (if exists)*: If a table with the same name already exists at the selected location, it will be replaced.

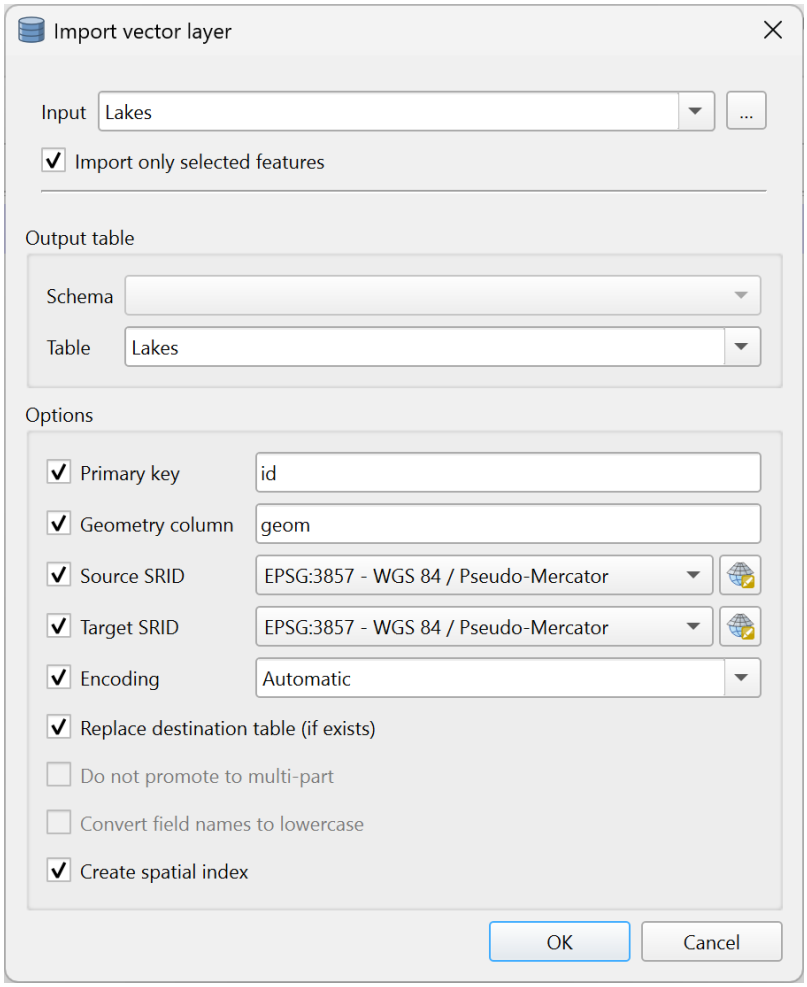



그림 25.6: Importing a vector layer into a spatial database using DB Manager

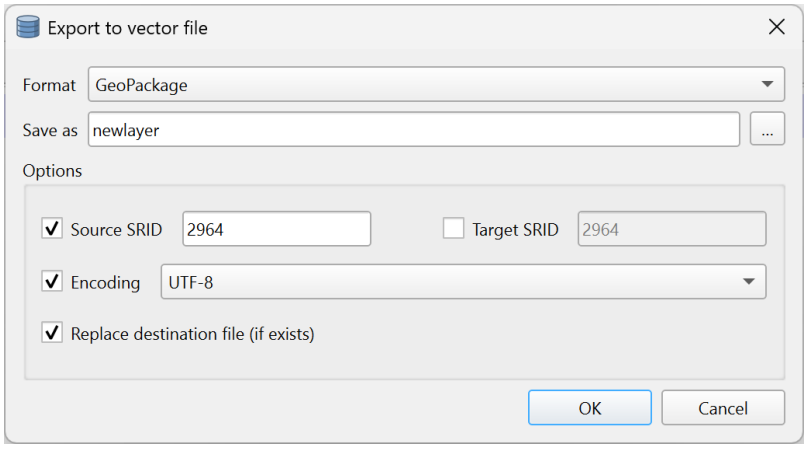



그림 25.7: Exporting a table to a vector file using DB Manager

### 25.2.2 도형 점검기 플러그인

도형 점검기는 레이어의 도형 무결성을 점검하고 수정할 수 있는 강력한 핵심 플러그인입니다. *Vector* *Check Geometries*…메뉴에서 이 플러그인을 사용할 수 있습니다.

#### 점검 환경설정

*Check Geometries* 대화창의 첫 번째 *Setup* 탭은 서로 다른 설정 그룹을 표시합니다:

- *Input vector layers*: 점검할 레이어를 선택합니다. *Only selected features* 체크박스를 체크하면 선택한 피처의 도형만 점검하도록 제약할 수 있습니다.
- *Allowed geometry types*: 입력 레이어 (들) 의 도형 유형을 다음으로 제한할 수 있습니다:
  - Point
  - Multipoint
  - Line
  - Multiline
  - Polygon
  - Multipolygon
- *Geometry validity*: 도형 유형에 따라, 도형의 무결성 기준을 다음 가운데 선택할 수 있습니다:
  - *Self intersections*
  - *Duplicate nodes*
  - *Self contacts*
  - *Polygon with less than 3 nodes.*
- *Geometry properties*: 도형 유형에 따라, 다음 옵션들을 선택할 수 있습니다:
  - *Polygons and multipolygons may not contain any holes*
  - *Multipart objects must consist of more than one part*
  - *Lines must not have dangles*
- *Geometry conditions*: 사용자가 도형의 무결성을 점검하기 위한 기준을 추가할 수 있습니다:
  - *Minimal segment length (map units)* 1,00
  - *Minimum angle between segment (deg)* 1,00
  - *Minimal polygon area (map units sqr.)* 1,00
  - *Maximum thinness* 1,00 최대 얇기 및 *Max. area (map units sqr.)* 1,00 최대 면적 (맵 단위) 으로 *No sliver polygons* 결함이 있는 조각 폴리곤을 지정
- *Topology checks*: 도형 유형에 따라, 다음 여러 옵션들을 선택할 수 있습니다:
  - *Checks for duplicates*

  - *Checks for features within other features*
  - *Checks for overlaps smaller than* 1,00
  - *Checks for gaps smaller than* 1,00
  - *Points must be covered by lines*
  - *Points must properly lie inside a polygon*
  - *Lines must not intersect any other lines*
  - *Lines must not intersect with features of layer*
  - *Polygons must follow boundaries of layer*
- *Tolerance*: 점검을 위한 허용 오차를 맵 레이어 단위로 정의할 수 있습니다.
- *Output vector layer*: 산출 벡터 레이어의 유형을 다음 가운데 선택할 수 있습니다:
  - *Modify input layer*
  - *Create new layers*

환경 설정에 만족하는 경우, *Run* 버튼을 클릭하면 됩니다.

도형 점검기 플러그인 은 다음과 같은 오류를 찾아낼 수 있습니다:

- 자체 교차: 스스로 교차하는 폴리곤
- 복제 노드: 한 선분에 2 개의 복제 노드
- 구멍: 폴리곤 내부의 구멍
- 선분 길이: 한계값 미만의 선분 길이
- 최소 각도: 한계값 미만의 각도를 이루는 2 개의 선분
- 최소 면적: 한계값 미만의 폴리곤 면적
- 조각 폴리곤: 긴 둘레를 가진 (아주 작은 면적의) 아주 작은 폴리곤에서 발생하는 오류
- 복제 피처
- 피처 내부의 피처
- 중첩: 중첩하는 폴리곤
- 틈: 폴리곤 사이의 틈

다음 그림은 이 플러그인의 서로 다른 점검 결과를 보여줍니다.

### 점검 결과 분석하기

점검 결과는 두 번째 *Result* 탭에 표시되며, 캔버스에 오류의 오버뷰 레이어로서 표시됩니다. (레이어명은 기본적으로 앞에 접두어 `checked_` 가 붙습니다.) *Geometry check result* 표는 한 행에 오류 하나를 담고, 그 행의 열은 각각 레이어명, 레이어 ID, 오류 유형, 오류 위치의 좌표, (오류 유형에 따라 달라지는) 값, 그리고 마지막으로 오류의 해결 방법을 담게 됩니다. 이 표의 하단에 있는 *Export* 버튼을 클릭하면 오류를 서로 다른 파일 포맷으로 내보낼 수 있습니다. 오류의 총 개수 및 수정된 오류의 개수도 볼 수 있습니다.

행을 선택하면 오류의 위치를 볼 수 있습니다. *Error* (기본값), *Feature*, *Don't move*, 그리고 *Highlight selected features* 가운데 다른 액션을 선택해서 이 습성을 변경할 수 있습니다.

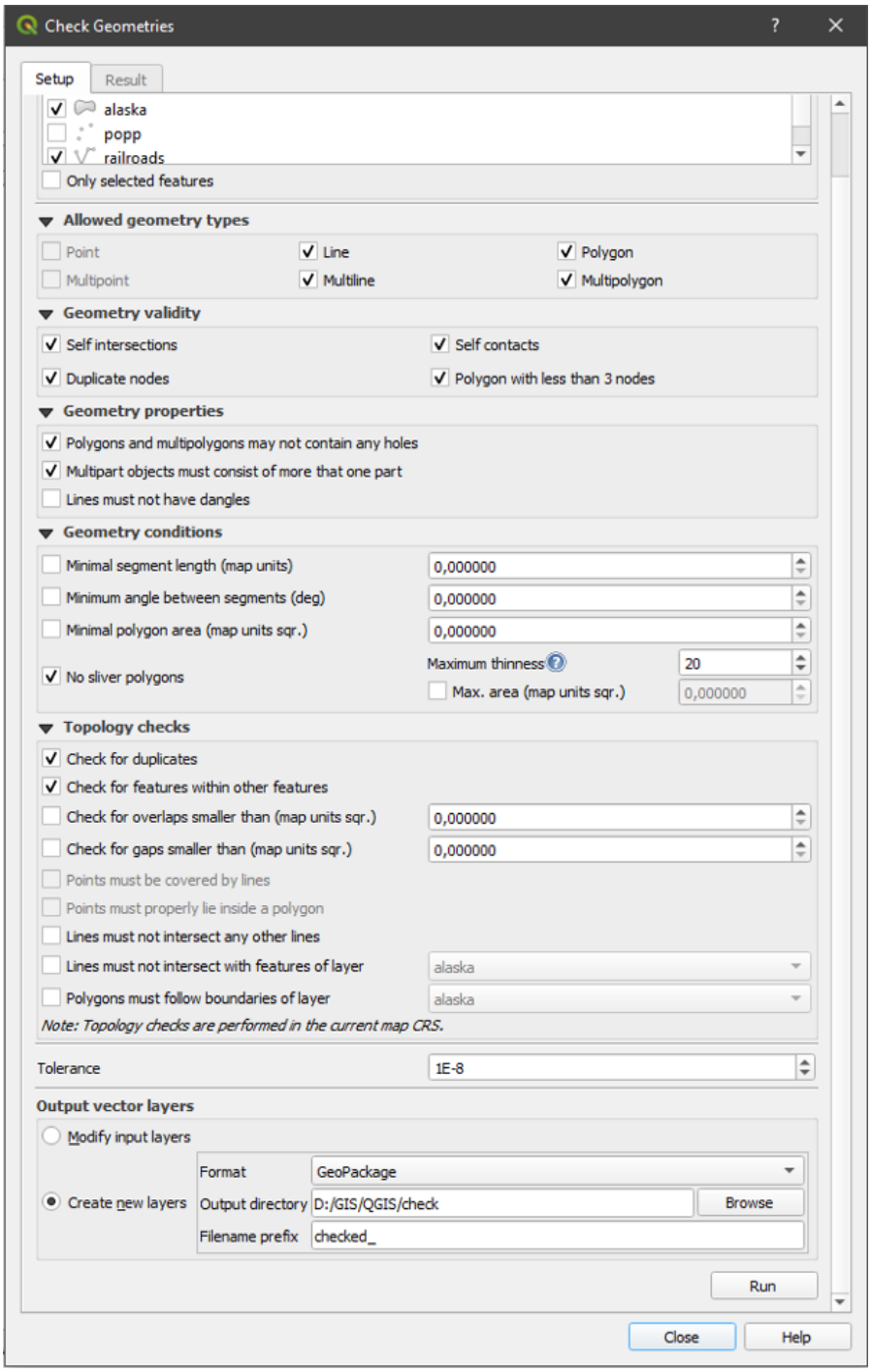



그림 25.8: 도형 점검기 플러그인

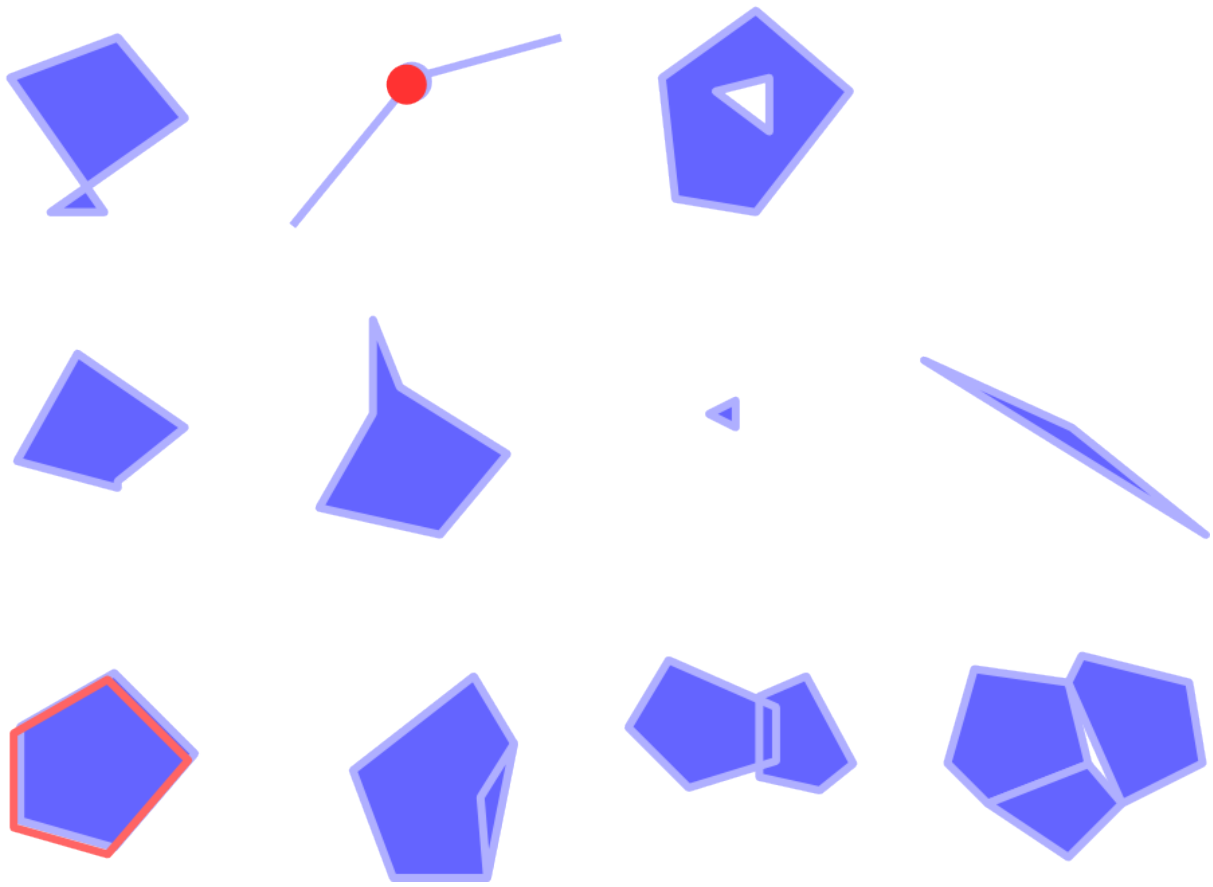


그림 25.9: 도형 점검기 플러그인이 지원하는 서로 다른 점검들

확대/축소 액션 아래에서 테이블의 행을 클릭하면:

- *Show selected features in attribute table*: 속성 테이블에 선택한 피처를 표시할 수 있습니다.
- *Fix selected errors using default resolution*: 기본 해결 방법을 사용해서 선택한 오류를 수정할 수 있습니다.
- *Fix selected errors, prompt for resolution method*: 해결 방법을 선택해서 선택한 오류를 수정할 수 있습니다. 다음 해결 방법 가운데 하나를 선택할 수 있는 창이 뜰 것입니다:
  - 최장 공유 경계를 가진 인접 폴리곤과 병합
  - 최대 면적을 가진 인접 폴리곤과 병합
  - 인접 폴리곤이 동일한 속성값을 가지고 있을 경우 인접 폴리곤과 병합, 없을 경우 내버려두기
  - 피처 삭제하기
  - 어떤 액션도 취하지 않기
- *Error resolution settings*: 오류 유형에 따라 기본 해결 방법을 변경할 수 있습니다.

> **팁**
>
> 한번에 오류 수정하기
>
> 표에서 `Ctrl` & 클릭을 사용해서 하나 이상의 행을 선택하면 여러 오류들을 한 번에 수정할 수 있습니다.

마지막으로, *Attribute to use when merging features by attribute value* 옵션을 통해 어떤 속성을 사용해서 피처들을 병합할지 선택할 수 있습니다.

### 25.2.3 메타검색 카탈로그 클라이언트 (MetaSearch Catalog Client)

#### 소개

메타검색 (MetaSearch) 은 메타데이터 카탈로그 서비스와 대화형으로 작업하는 QGIS 플러그인으로, OGC 레코드 API 와 OGC 웹 용 카탈로그 서비스 (CSW) 표준 을 둘 다 지원합니다.

메타검색은 QGIS 내부에서 메타데이터 카탈로그를 검색할 수 있는 쉽고 직관적인 접근법과 사용자 친화적인 인터페이스를 제공하고 있습니다.

#### QGIS 에서 메타데이터 카탈로그 작업하기

QGIS 는 기본적으로 메타검색을 모든 의존성과 함께 내장하고 있으며, QGIS 플러그인 관리자에서 활성화할 수 있습니다.

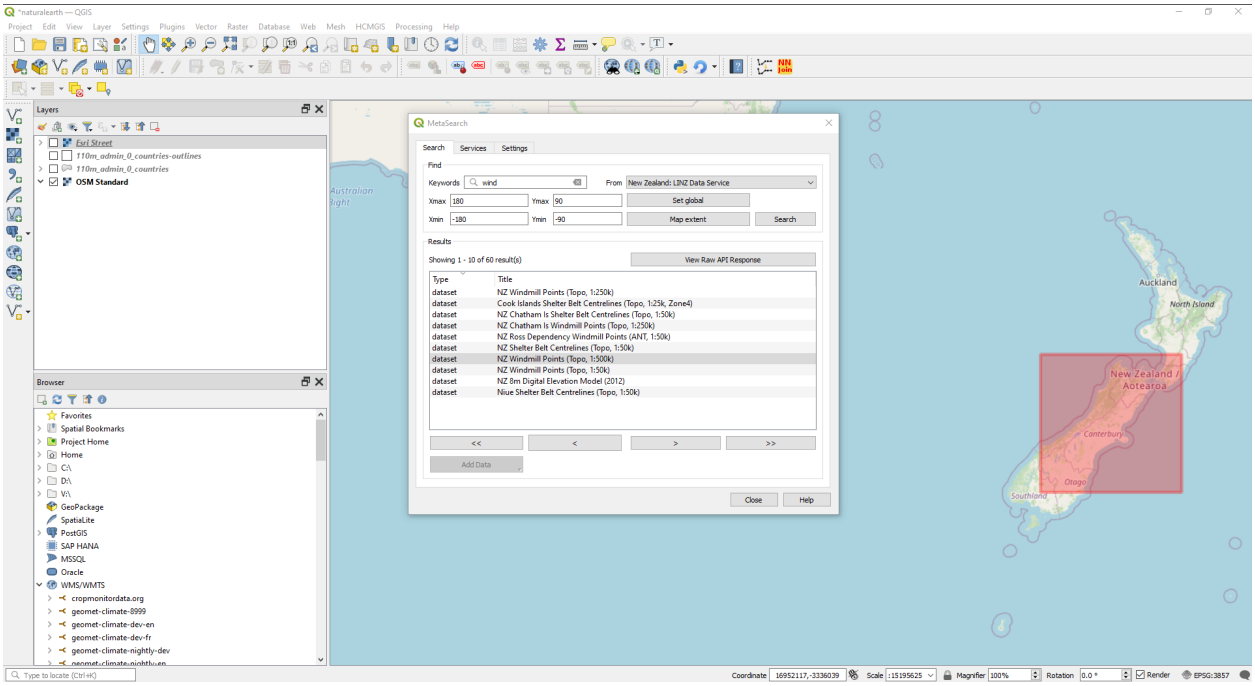



그림 25.10: 메타검색으로 서비스를 검색한 결과

## OGC 레코드 API

OGC API - Records 는 웹 상에서 지리공간 리소스를 발견하기 위한 OGC (Open Geospatial Consortium) 표준입니다.

### 웹 용 카탈로그 서비스 (CSW)

CSW (Catalog Service for the Web) 는 OGC (Open Geospatial Consortium) 사양으로, 데이터, 서비스 및 기타 잠재적인 리소스 관련 메타데이터를 찾고, 탐색하고, 쿼리하기 위한 공통 인터페이스를 정의하고 있습니다.

### 구동

메타검색을 구동하려면,  아이콘을 클릭하거나 QGIS 주 메뉴에서 *Web* → *MetaSearch* → *MetaSearch* 메뉴 옵션을 선택하십시오. 메타검색 대화창이 열릴 것입니다. 대화창의 주 GUI 는 *Services*, *Search* 및 *Settings* 탭 3 개로 이루어져 있습니다.

### 카탈로그 서비스 관리하기

*Services* 탭에서 사용할 수 있는 모든 카탈로그 서비스를 관리할 수 있습니다. 메타검색은 카탈로그 서비스의 기본 목록을 제공하는데, *Add Default Services* 버튼을 클릭해서 목록에 카탈로그 서비스를 추가할 수 있습니다.

목록에 있는 모든 카탈로그 서비스 항목을 보려면, 드롭다운 메뉴를 클릭하십시오.

카탈로그 서비스 항목을 추가하려면:

1. *New* 버튼을 클릭하십시오.

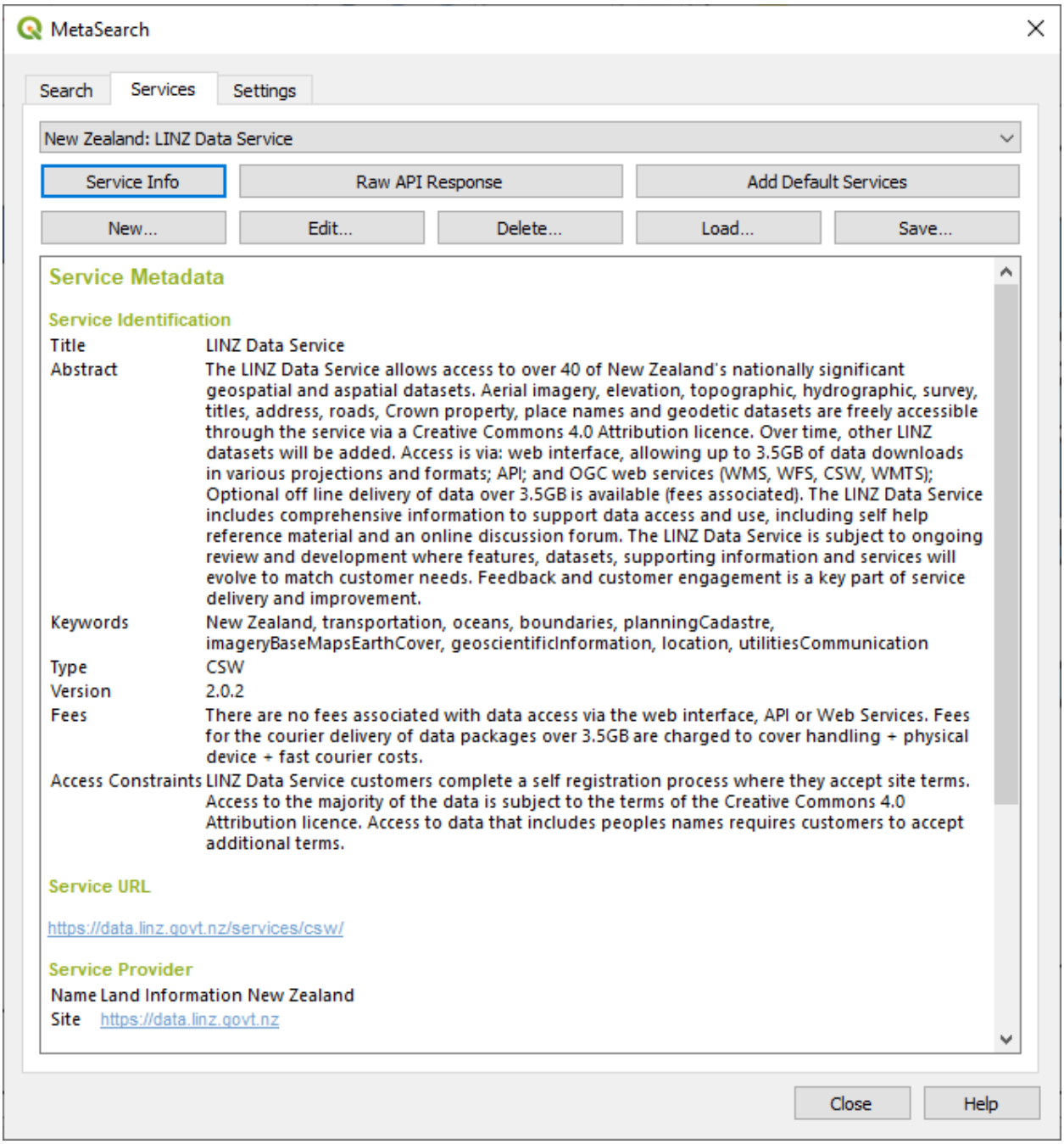



그림 25.11: 카탈로그 서비스 관리하기

2. 서비스 이름 (*Name*) 은 물론, 종단점 (*URL*) 도 입력하십시오. OGC CSW 2.0.2 카탈로그의 경우 (전체 GetCapabilities URL 이 아니라) 기본 URL 만 필요하다는 사실을 기억하십시오. OGC 레코드 API 카탈로그의 경우, URL 이 선택 집합 목록 (collection) 의 종단점을 가리키는 경로여야 합니다.
3. 카탈로그가 인증을 요구하는 경우, 적절한 *User name* 및 *Password* 인증 정보를 입력하십시오.
4. *OK* 를 클릭하면 서비스가 목록에 추가될 것입니다.

기존 카탈로그 서비스 항목을 편집하려면:

1. 편집하고자 하는 항목을 선택한 다음
2. *Edit* 버튼을 클릭하고
3. *Name* 또는 *URL* 값을 수정하십시오.
4. :guilabel:`OK`를 클릭하십시오.

카탈로그 서비스 항목을 삭제하려면, 삭제하고자 하는 항목을 선택한 다음 *Delete* 버튼을 클릭하십시오. 항목 삭제를 확인하는 메시지 창이 뜰 것입니다.

메타검색은 서비스 연결을 XML 파일로 저장하고 불러올 수 있습니다. 서로 다른 응용 프로그램 사이에서 설정을 공유해야 하는 경우 이 기능이 유용합니다. 다음은 이 XML 파일 포맷의 예시입니다.

```
<?xml version="1.0" encoding="UTF-8"?>
<qgsCSWConnections version="1.0">
    <csw type="OGC CSW 2.0.2" name="Data.gov CSW" url="https://catalog.data.gov/csw-
↪all"/>
    <csw type="OGC CSW 2.0.2" name="Geonorge - National CSW service for Norway" url=
↪"https://www.geonorge.no/geonetwork/srv/eng/csw"/>
    <csw type="OGC CSW 2.0.2" name="Geoportale Nazionale - Servizio di ricerca␣
↪Italiano" url="http://www.pcn.minambiente.it/geoportal/csw"/>
    <csw type="OGC CSW 2.0.2" name="LINZ Data Service" url="http://data.linz.govt.nz/
↪feeds/csw"/>
    <csw type="OGC CSW 2.0.2" name="Nationaal Georegister (Nederland)" url="http://
↪www.nationaalgeoregister.nl/geonetwork/srv/eng/csw"/>
    <csw type="OGC CSW 2.0.2" name="RNDT - Repertorio Nazionale dei Dati Territoriali␣
↪- Servizio di ricerca" url="http://www.rndt.gov.it/RNDT/CSW"/>
    <csw type="OGC CSW 2.0.2" name="UK Location Catalogue Publishing Service" url=
↪"http://csw.data.gov.uk/geonetwork/srv/en/csw"/>
    <csw type="OGC CSW 2.0.2" name="UNEP/GRID-Geneva Metadata Catalog" url="http://
↪metadata.grid.unep.ch:8080/geonetwork/srv/eng/csw"/>
</qgsCSWConnections>
```

항목 목록을 불러오려면:

1. *Load* 버튼을 클릭하십시오. 새 창이 열릴 것입니다.
2. *Browse* 버튼을 클릭한 다음 불러오고 싶은 항목의 XML 파일을 찾아 선택하십시오.
3. *Open* 버튼을 클릭하십시오. 항목 목록이 표시될 것입니다.
4. 목록에서 원하는 항목을 선택한 다음 *Load* 버튼을 누르십시오.

카탈로그 서비스를 선택하고 *Service Info* 버튼을 클릭하면 서비스 식별 정보, 서비스 제공자 및 연락 정보와 같은 서비스 관련 정보를 표시합니다. 원시 (raw) API 응답을 살펴보고 싶다면, *Raw API Response* 버튼을 누르십시오. 서버 정보를 원시 JSON 또는 XML 서식으로 표시한 개별 창이 열릴 것입니다.

카탈로그 서비스 검색하기

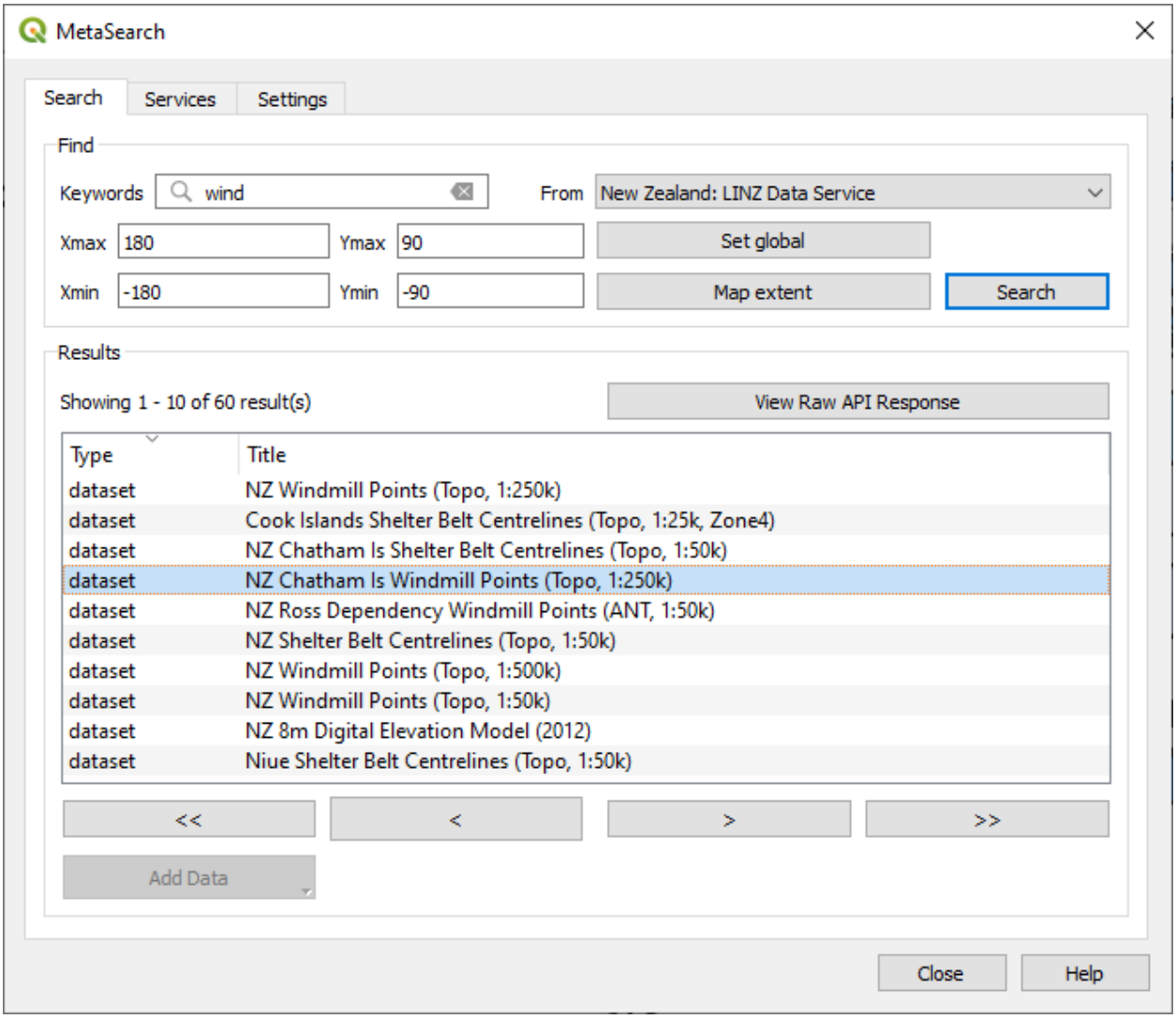



그림 25.12: 카탈로그 서비스 검색하기

*Search* 탭에서 데이터 및 서비스에 대해 카탈로그 서비스를 쿼리할 수 있으며, 다양한 검색 파라미터를 설정하고 검색 결과를 살펴볼 수 있습니다.

다음과 같은 검색 파라미터를 사용할 수 있습니다:

- *Keywords*: 텍스트 검색 키워드를 마음대로 지정할 수 있습니다.
- *From*: 쿼리를 수행할 카탈로그 서비스를 선택할 수 있습니다.
- **경계 상자**: *Xmax*, *Xmin*, *Ymax*, 및 *Ymin* 을 정의해서 필터링할 관심 공간 영역을 지정합니다. *Set Global* 버튼을 클릭하면 전체 검색을, *Map Extent* 버튼을 클릭하면 가시화된 영역만 또는 값을 직접 입력해서 정의한 영역만 검색할 수 있습니다.

*Search* 버튼을 누르면 선택한 메타데이터 카탈로그를 검색할 것입니다. 검색 결과는 목록으로 표시되어, 열 제목을 클릭하면 정렬시킬 수 있습니다. 검색 결과 표 아래 있는 방향 버튼을 통해 검색 결과를 탐색할 수 있습니다.

검색 결과를 선택한 다음:

- *View Raw API Response* 버튼을 클릭하면 서비스 응답을 원시 JSON 또는 XML 서식으로 담고 있는 창이 열립니다.
- 메타데이터 레코드가 관련 경계 상자를 보유하고 있는 경우, 경계 상자의 범위가 맵 상에 표시될 것입니다.
- 레코드를 더블클릭하면 모든 관련 접근 링크를 담은 레코드 메타데이터를 표시합니다. 이 링크를 클릭하면 사용자 웹브라우저로 링크를 엽니다.
- 레코드를 지원하는 웹 서비스 (WMS/WMTS, WFS, WCS, ArcGIS REST 서비스 등등) 또는 GIS 파일일 경우, *Add Data* 버튼이 활성화될 것입니다. 이 버튼을 클릭하면, 메타검색이 이 서비스가 무결한 OWS 인지 검증할 것입니다. 검증을 통과한 경우, 적절한 QGIS 연결 목록에 서비스를 추가하고 알맞은 연결 대화창을 열 것입니다. *Add GIS File* 을 선택한다면 현재 프로젝트에 링크된 GIS 파일이 추가될 것입니다.

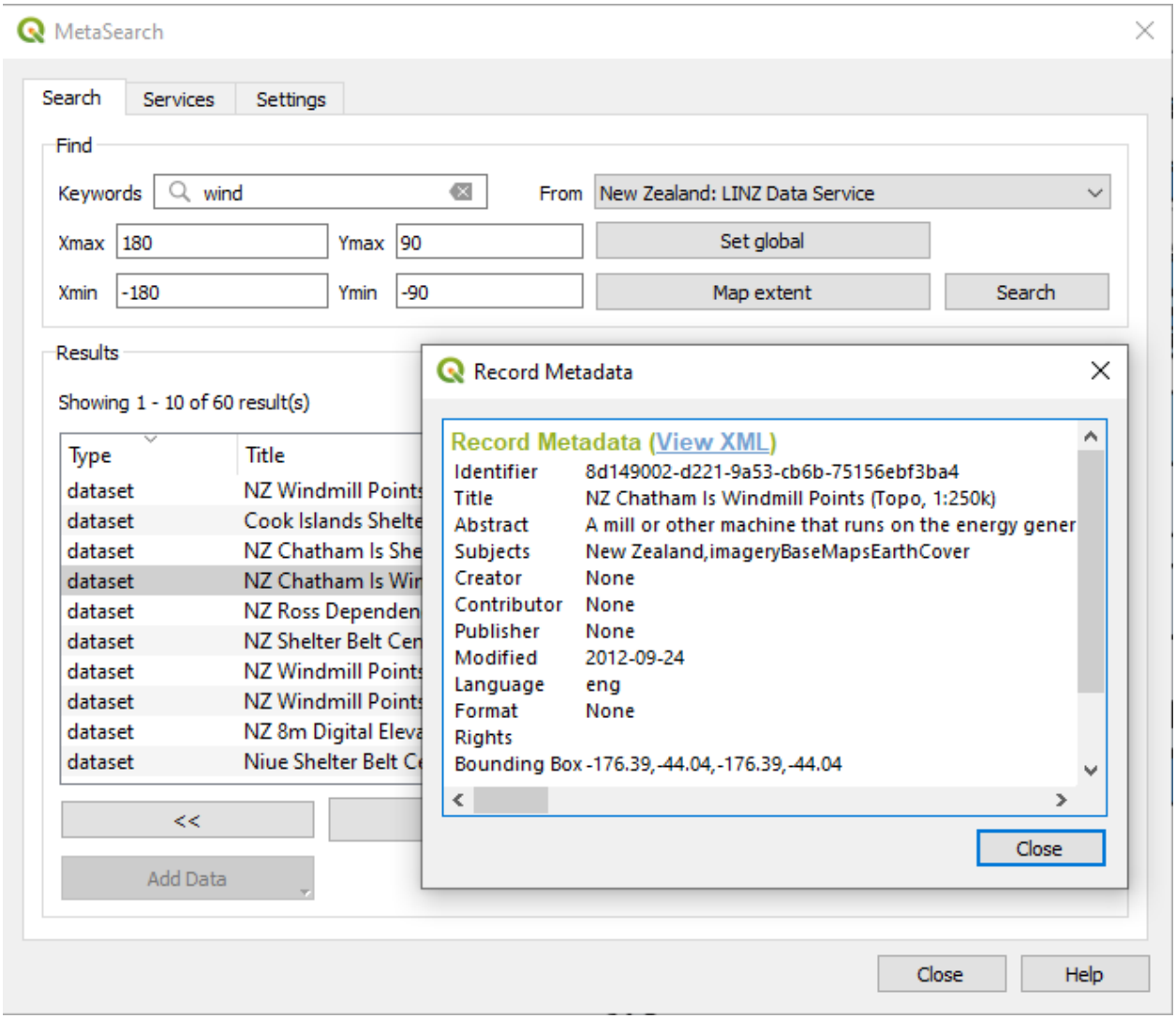



그림 25.13: 메타데이터 레코드 표시

### 설정

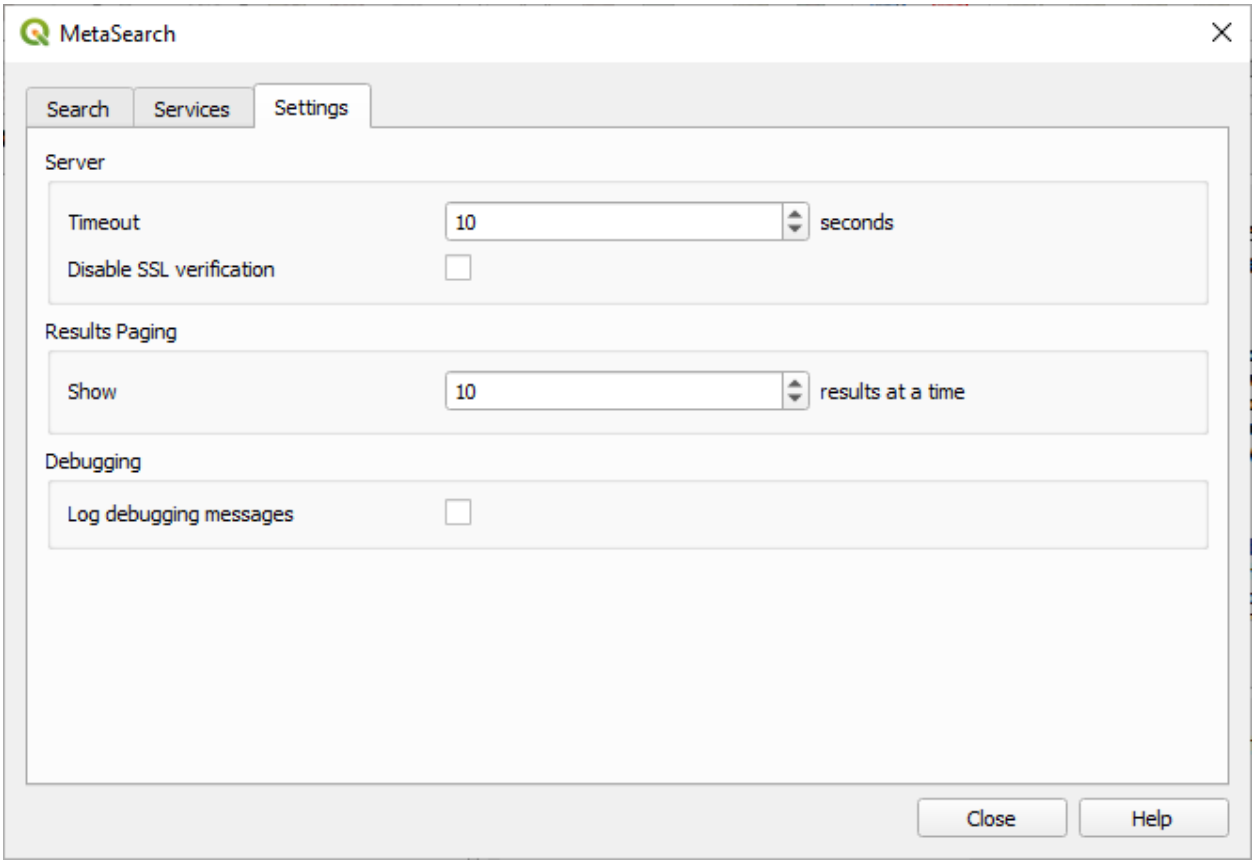



그림 25.14: 메타검색 설정하기

*Settings* 탭에서 메타검색 설정을 다음과 같이 미세 조정할 수 있습니다:

- *Server Timeout*: 메타데이터 카탈로그를 검색할 때, 몇 초 후에야 연결 시도 차단으로 판단할지 설정합니다. 기본값은 10 입니다.
- *Disable SSL verification*: SSL 검증을 끄는 옵션입니다.
- *Results paging*: 메타데이터 카탈로그를 검색할 때, 페이지 당 결과를 몇 개나 표시할지 설정합니다. 기본값은 10 입니다.
- *Log debugging messages*: 로그 메시지 패널 에서 디버그 작업 메시지 로그를 활성화시킬 수 있는 옵션입니다.

### 카탈로그 서버 오류

카탈로그가 웹 브라우저에서는 작동하지만, 메타검색에서는 작동하지 않는 경우가 있습니다. 그 원인은 카탈로그 서버의 환경 설정 때문일 수도 있습니다. 카탈로그 서버 제공자는 자신의 환경 설정에서 URL 들이 일관되고 최신 상태인 것을 보장해야 합니다. (HTTP -> HTTPS 전환 시나리오에서 흔히 일어나는 일입니다.) 해당 문제에 대한 심화 설명 및 해결책을 알고 싶다면 pycsw FAQ item 을 읽어보십시오. FAQ 항목들이 pycsw 에 특화되어 있기는 하지만, 다른 카탈로그 서버에도 일반적으로 적용할 수 있습니다.

### 25.2.4 오프라인 편집 작업 플러그인

For data collection, it is a common situation to work with a laptop or a cell phone offline in the field. Upon returning to the network, the changes need to be synchronized with the master datasource (e.g., a PostgreSQL database). If several persons are working simultaneously on the same datasets, it is difficult to merge the edits by hand, even if people don't change the same features.

 Offline Editing 플러그인은 데이터소스의 콘텐츠를 SpatiaLite 또는 GeoPackage 데이터베이스로 복사해서 전용 테이블에 오프라인 편집 내용을 저장함으로써 동기화를 자동화합니다. 다시 네트워크에 연결한 다음 마스터 데이터셋에 오프라인 편집 내용을 적용할 수도 있습니다.

플러그인을 사용하려면:

1. Open a project with some vector layers (e.g., from an Esri Shapefile, PostgreSQL or WFS-T datasource).
2. 사용자가 이미 플러그인을 활성화했다고 가정합니다. (핵심 및 외부 플러그인 참조) *Database* → *Offline Editing* →  *Convert to offline project* 메뉴 옵션을 선택하십시오. 오프라인 프로젝트 생성 대화창이 열립니다.
3. *Storage type* 옵션에서 저장소 유형을 선택하십시오. *GeoPackage* 또는 *SpatiaLite* 데이터베이스 유형을 선택할 수 있습니다.
4. *Browse* 버튼을 통해 *Offline data* 를 저장할 데이터베이스의 위치를 지정하십시오. 기존 파일일 수도 있고 새로 생성할 수도 있습니다.
5. *Select remote layers* 부분에서 저장하려는 레이어를 체크하십시오. 해당 레이어의 내용이 데이터베이스 테이블에 저장됩니다.

   **참고**

   대상 데이터베이스 포맷이 목록을 네이티브하게 지원하지 못 하기 때문에, 오프라인 편집 작업 플러그인이 {string, number} 목록 필드를 문자열 필드로 변환합니다. 이때 값을 쉼표로 구분합니다. 이렇게 하면 오프라인 시 이런 필드의 내용을 읽고 편집할 수 있게 됩니다.

   원본 레이어와 오프라인 레이어로부터 나온 필드들을 둘 다 처리하고자 하는 경우, *try()* and *array* 표현식 함수를 사용하면 됩니다. 다음은 그 예시입니다:

   ```
   try(array_contains("field",1),array_contains(string_to_array("field"),1))
   ```

6.  *Only synchronize selected features if a selection is present* 옵션을 체크해서 하위 집합 (subset) 에 대해서만 오프라인 편집 작업 및 저장을 할 수 있도록 설정할 수 있습니다. 대용량 레이어의 경우 이 옵션을 항상 활성화하는 편이 좋습니다.

   끝났습니다!
7. 사용자 프로젝트를 저장한 다음 현장에 가져가십시오.
8. 오프라인에서 레이어를 편집하십시오.
9. 다시 온라인에 접속하면, *Database* → *Offline Editing* →  *Synchronize* 메뉴 옵션을 선택해서 변경 사항을 업로드하십시오.

**참고**

*Layers* 패널에서 오프라인 시 사용된 레이어를  아이콘으로 표시합니다.

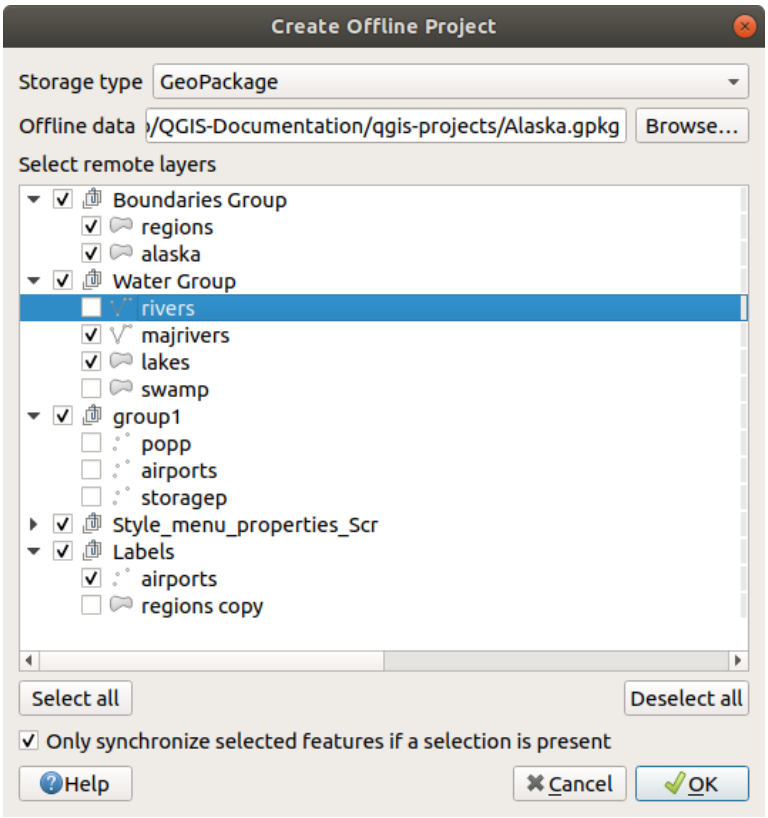



그림 25.15: 오프라인 프로젝트 생성하기

### 25.2.5 위상 점검기 플러그인

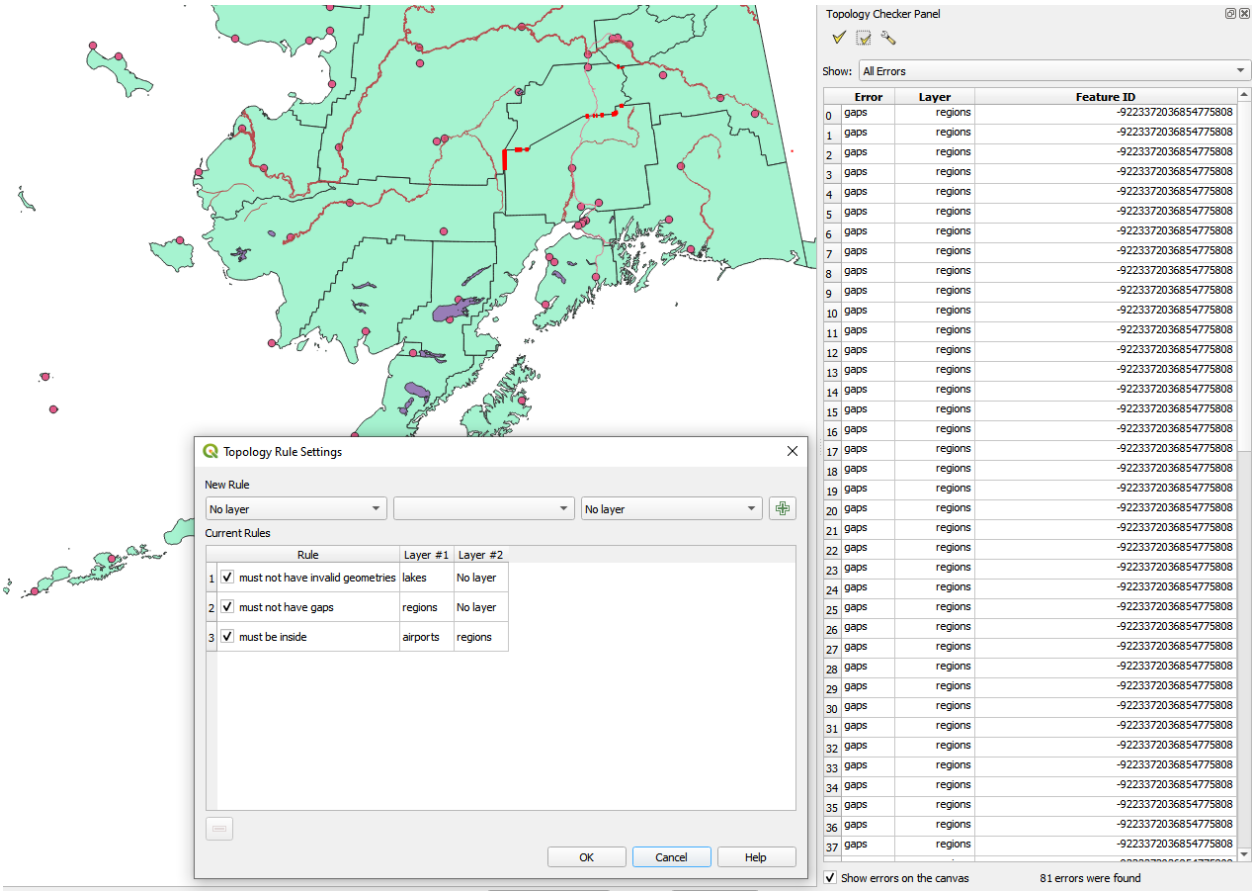



그림 25.16: 위상 점검기 플러그인

위상 (topology) 이란 지리적 지역의 객체를 표현하는 포인트, 라인 및 폴리곤 간의 관계를 말합니다. 위상 점검기 플러그인을 사용해서 사용자 벡터 파일을 살펴보고 여러 위상 규칙을 기준으로 위상을 점검할 수 있습니다. 이런 규칙들은 사용자 객체가 서로'동등 (equal)'한지, '포함 (contain)'하는지, '커버 (cover)'하는지, '커버되는 (CoveredBy)'지, '공간교차 (cross)'하는지, '분절 (disjoint)'되는지, '교차 (intersect)'하는지, '중첩 (overlap)'하는지, '접촉 (touch)'하는지, 또는 서로의'내부 (within)'에 있는지 그 공간 관계를 점검합니다. 이 관계는 사용자 벡터 데이터에 개별적으로 어느 위상 규칙을 적용하느냐에 따라 달라집니다. (예를 들면, 일반적으로 라인 레이어에서는 오버슛 (overshoot) 을 허용하지 않지만, 오버슛이 막다른 길을 표현하는 경우 사용자 벡터 레이어에서 제거하지 않을 것입니다.)

QGIS 는 위상 편집 기능을 내장하고 있는데, 오류가 없는 새 객체를 생성하는 데 매우 유용합니다. 그러나 기존 데이터의 오류 및 사용자가 발생시킨 오류를 찾기란 어렵습니다. 이 플러그인을 사용하면 일련의 규칙을 통해 그런 오류들을 찾을 수 있습니다.

*Topology checker* 플러그인을 활성화하려면:

1. *Plugins* 메뉴로 가십시오.
2. *Manage and Install plugins* 플러그인 관리자를 연 다음 *Installed* 설치된 플러그인을 선택하십시오.
3. *Topology checker* 위상 점검기 옵션을 체크하십시오.
4. *Plugin manager* 대화창을 닫으십시오. *Vector* 메뉴에 *Topology checker* 항목이 추가되었습니다.

 *Topology checker* 를 활성화시킨 다음, 위상 점검기를 열고 *Configure* 를 선택해서 사용자의 위상 규칙을 생성하십시오.

**포인트 레이어** 상에서 다음 규칙들을 사용할 수 있습니다:

- *Must be covered by*: 사용자 프로젝트에서 벡터 레이어를 선택할 수 있습니다. 지정한 벡터 레이어로 커버되지 않는 포인트를'오류'항목으로 분류합니다.
- *Must be covered by endpoints of*: 사용자 프로젝트에서 라인 레이어를 선택할 수 있습니다.
- *Must be inside*: 사용자 프로젝트에서 폴리곤 레이어를 선택할 수 있습니다. 포인트가 폴리곤 내부에 들어가야만 합니다. QGIS 는 들어가지 않는 포인트에'오류'를 작성합니다.
- *Must not have duplicates*: 어느 포인트가 두 번 이상 표현되는 경우, 해당 포인트를'오류'항목으로 분류합니다.
- *Must not have invalid geometries*: 도형이 무결한지 여부를 점검합니다.
- *Must not have multi-part-geometries*: 모든 다중 부분 포인트를'오류'로 분류합니다.

**라인 레이어** 상에서 다음 규칙들을 사용할 수 있습니다:

- *End points must be covered by*: 사용자 프로젝트에서 포인트 레이어를 선택할 수 있습니다.
- *Must not have dangles*: 라인 레이어에 있는 오버슛을 표시할 것입니다.
- *Must not have duplicates*: 어느 라인 피처가 두 번 이상 표현되는 경우, 해당 라인을'오류'항목으로 분류합니다.
- *Must not have invalid geometries*: 도형이 무결한지 여부를 점검합니다.
- *Must not have multi-part geometries*: 도형 한 개가 실제로는 단순 (단일 부분, single-part) 도형의 집합인 경우가 종종 있습니다. 이런 도형을 다중 부분 (multi-part) 도형이라고 합니다. 다중 부분 도형이 한 가지 유형의 단순 도형으로만 이루어져 있을 경우 멀티포인트, 멀티라인스트링, 멀티폴리곤이라 부릅니다. 모든 다중 부분 라인을'오류'로 분류합니다.
- *Must not have pseudos*: 라인 도형의 종단점이 다른 두 도형의 종단점과 연결돼 있어야 합니다. 종단점이 다른 한 도형의 종단점과만 연결돼 있는 경우, 이런 종단점을 유사 (pseudo) 노드라고 합니다.

**폴리곤 레이어** 상에서 다음 규칙들을 사용할 수 있습니다:

- *Must contain*: 폴리곤 레이어가 두 번째 레이어의 포인트 도형을 적어도 하나 포함해야만 합니다.
- *Must not have duplicates*: 어떤 폴리곤과 동일한 폴리곤이 동일 레이어에 있어서는 안 됩니다. 어떤 폴리곤이 두 번 이상 표현되는 경우, 해당 폴리곤을'오류'로 분류합니다.
- *Must not have gaps*: 인접한 폴리곤 사이에 틈이 있어서는 안 됩니다. 행정구역 경계를 그 예로 들 수 있습니다. (미국의 주 (☐) 폴리곤들 사이엔 아무 틈도 존재하지 않습니다.)
- *Must not have invalid geometries*: 도형이 무결한지 여부를 점검합니다. 다음은 무결한 도형을 정의하는 규칙 가운데 일부입니다:
  - 폴리곤 고리는 닫혀 있어야만 합니다.
  - 구멍을 정의하는 고리는 외부 경계선을 정의하는 고리 내부에 있어야만 합니다.
  - 고리들이 자체 교차해서는 안 됩니다. (서로 접하거나 공간교차해서는 안 됩니다.)
  - 고리는 한 포인트에서 접촉하는 경우 외엔 서로 접해서는 안 됩니다.
- *Must not have multi-part geometries*: 도형 한 개가 실제로는 단순 (단일 부분, single-part) 도형의 집합인 경우가 종종 있습니다. 이런 도형을 다중 부분 (multi-part) 도형이라고 합니다. 다중 부분 도형이 한 가지 유형의 단순 도형으로만 이루어져 있을 경우 멀티포인트, 멀티라인스트링, 멀티폴리곤이라 부릅니다. 예를 들어 여러 개의 섬으로 이루어진 국가라면 멀티폴리곤으로 표현할 수 있습니다.
- *Must not overlap*: 인접한 폴리곤들이 공통 영역을 공유해서는 안 됩니다.
- *Must not overlap with*: 한 레이어에서 인접한 폴리곤들이 또다른 레이어의 폴리곤들과 공통 영역을 공유해서는 안 됩니다.

*New rule* 을 생성할 때, *Current rules* 에 새 규칙을 추가하려면 Add rule 아이콘을 클릭하십시오. 체크박스를 클릭하면 규칙을 하나하나 활성화시키거나 비활성화시킬 수 있습니다. 규칙을 오른쪽 클릭하면 다음과 같은 옵션들을 설정할 수 있습니다:

- *Select All*: 규칙들을 모두 선택합니다.
- *Activate* 또는 *Deactivate*: 선택한 규칙을 활성화 또는 비활성화시킵니다.
- *Toggle activation*: 선택한 규칙의 활성화 상태를 켜고끕니다.
- *Delete*: 선택한 규칙을 삭제합니다. Delete selected rules 버튼으로도 삭제할 수 있습니다.

*OK* 를 클릭한 다음 *Topology checker* 패널에서 다음을 선택하십시오:

- Validate All: 해당 레이어 (들) 의 모든 피처에 활성화된 규칙들을 적용합니다.
- 또는 Validate Extent: 현재 맵 캔버스 안에 있는 해당 레이어 (들) 의 피처에 활성화된 규칙들을 적용합니다. 이 버튼은 눌린 상태로 유지되어 맵 캔버스 범위가 변할 때마다 그 결과를 업데이트할 것입니다.

오류는 오류 유형, 레이어 ID 및 피처 ID 를 담고 있는 결과 테이블에 나타날 것입니다. 오류들을 특정 오류 유형으로 필터링하려면 *Filter errors by rule* 메뉴를 사용하십시오.

*Show errors on the canvas* 옵션을 체크하면 캔버스 상에 오류 위치를 표시합니다. 테이블에서 행을 하나 클릭하면 맵 캔버스를 해당 피처로 확대/축소할 것입니다. 이때*QGIS* 디지타이즈 작업 도구 를 사용해서 해당 피처의 오류를 고칠 수 있습니다.

다음은 QGIS 가 제공하는 핵심 플러그인 목록입니다. 기본적으로 활성화돼 있지 않을 수도 있습니다.

| 아이콘 | 플러그인 | 설명 | 지침서 참조 |
|---|---|---|---|
| | DB 관리자 | QGIS 에서 사용자 데이터베이스를 관리합니다. | 데이터베이스 관리자 플러그인 |
| | 도형 점검기 | 벡터 도형에서 오류를 점검하고 고칩니다. | 도형 점검기 플러그인 |
| | GRASS 7 | GRASS 기능 | *GRASS GIS* 통합 |
| | GRASS GIS 제공자 | GRASS GIS 공간 처리 기능 | *GRASS GIS* 통합 |
| | 메타검색 카탈로그 클라이언트 (MetaSearch Catalog Client) | 메타데이터 카탈로그 서비스 (CSW) 와 대화형 작업을 합니다. | 메타검색 카탈로그 클라이언트 *(MetaSearch Catalog Client)* |
| | 오프라인 편집 작업 | 오프라인에서 편집하고 데이터베이스와 동기화합니다. | 오프라인 편집 작업 플러그인 |
| | 공간 처리 | 공간 데이터 처리 프레임워크 | *QGIS* 공간 처리 프레임워크 |
| | 위상 점검기 | 벡터 레이어에서 위상 오류를 찾습니다. | 위상 점검기 플러그인 |

**참고**

GRASS 7 및 GRASS GIS 제공자 핵심 플러그인을 사용하려면 환경설정을 해줘야 합니다. 여기 에서 관련 정보를 찾아볼 수 있습니다.

## 25.3 QGIS 파이썬 콘솔

이 장을 보다 보면 알게 되겠지만, QGIS 는 플러그인 아키텍처로 설계되어 있습니다. 플러그인은 지리공간 업계에서는 매우 유명한 언어인 파이썬으로 작성할 수 있습니다.

QGIS 는 사용자가 객체 (레이어, 피처, 인터페이스) 와 대화형 작업을 할 수 있도록 파이썬 API(예시 코드를 보고 싶다면 PyQGIS 개발자 쿡북 을 참조) 를 제공합니다. 물론 QGIS 에는 파이썬 콘솔도 있습니다.

The QGIS Python Console is an interactive shell for Python command executions. It also has a Python file editor that allows you to edit and save your Python scripts. Both console and editor are based on PyQScintilla2 package. To open the console go to *Plugins* → *Python Console* (`Ctrl+Alt+P`) or click on the Python Console icon in the *Plugins* toolbar.

### 25.3.1 대화형 콘솔

이 콘솔은 사용자가 파이썬 명령어를 실행할 수 있게 해주는 파이썬 해석기입니다. QGIS(analysis, core, gui, server, processing, 3d) 와 Qt(QtCore, QtGui, QtNetwork, QtWidgets, QtXml) 의 모듈은 물론 파이썬의 math, os, re 및 sys 모듈이 이미 구현되어 있기 때문에 직접 사용할 수 있습니다.

대화형 콘솔은 툴바, 입력 영역 그리고 출력 영역으로 이루어져 있습니다.

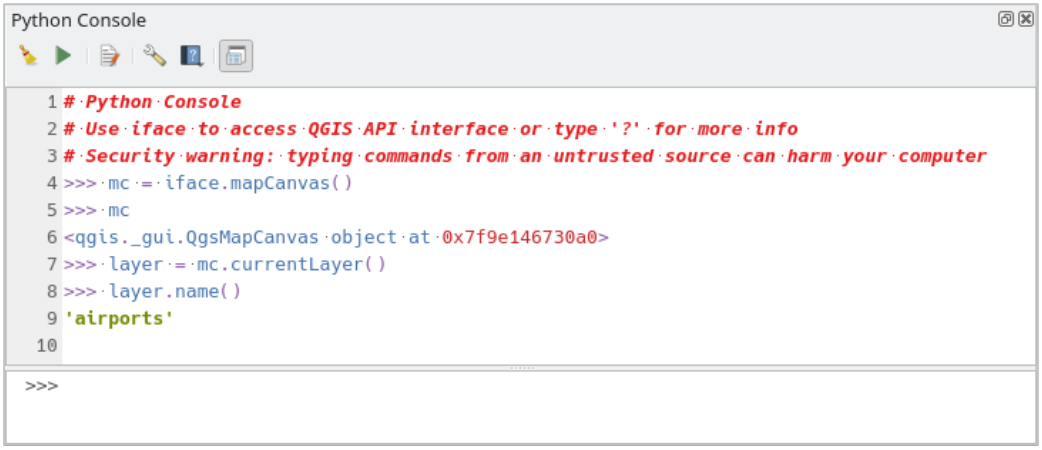



그림 25.17: 파이썬 콘솔

#### 툴바

툴바는 다음 도구들을 제공합니다:

- Clear Console: 출력 영역을 지웁니다.
- Run Command: 입력 영역에서 사용할 수 있습니다. `Enter` 키를 누르는 것과 동일한 기능입니다.
- Show Editor: 코드 편집기 를 켜고 끕니다.
- Options…: 콘솔 속성 을 환경설정할 수 있는 대화창을 엽니다.
- Help…: 여러 문서에 접근할 수 있는 메뉴를 제공합니다:
  - 파이썬 콘솔 도움말 (현재 페이지)
  - PyQGIS API documentation

  - PyQGIS 쿡북
-  Dock Code Editor: QGIS 인터페이스에 패널을 도킹시키거나 도킹을 해제합니다.

### 입력 영역

콘솔 입력 영역의 주요 기능은 다음과 같습니다:

- 다음 API 들을 위한 코드 완성, 문법 구문 강조, 그리고 팝업 도움말:
  - Python
  - PyQGIS
  - PyQt5
  - QScintilla2
  - osgeo-gdal-ogr
  - osgeo-geos
- `Ctrl+Alt+Space`: *Python settings* 에서 활성화하면 사용할 수 있는 자동 완성 목록 보기
- 입력 영역에서 `Enter` 키를 누르거나  아이콘을 클릭해서 코드 조각 (snippet) 실행하기
- 출력 영역에서 컨텍스트 메뉴의 *Run Selected* 를 선택하거나 `Ctrl+E` 조합키를 눌러 코드 조각 실행하기
- 입력 영역에서 `Up` 및 `Down` 방향키를 눌러 명령어 이력을 탐색하고 원하는 명령어 실행하기
- `Ctrl+Shift+Space` 조합키로 명령어 이력 살펴보기: 어느 행을 더블클릭하면 해당 명령어를 실행합니다. 입력 영역에서 컨텍스트 메뉴를 통해 *Command History* 대화창을 열 수도 있습니다.
- 명령어 이력 저장 및 삭제: 명령어 이력은 활성화된사용자 프로파일 폴더 아래 `console_history.txt` 파일로 저장됩니다.
- 다음 특수 명령어들을 입력하십시오:
  - `?`: 파이썬 콘솔 도움말을 출력합니다.
  - `_api` 또는 `_api(object)`: 전자는 QGIS C++ API 문서를 열고 후자는 (QGIS C++ API 또는 Qt API 문서에 있는) 특정 객체 문서를 엽니다.
  - `_pyqgis` 또는 `_pyqgis(object)`: 전자는 QGIS Python API 문서를 열고 후자는 (QGIS Python API 또는 Qt API 문서에 있는) 특정 객체 문서를 엽니다.
  - `_cookbook`: PyQGIS 쿡북 을 엽니다.
  - `!`: 명령어 앞에 느낌표를 붙이면 파이썬 콘솔에서 셸 (Shell) 명령어를 실행할 수 있습니다. 콘솔이 하위 프로세스를 시작해서 그 산출물을 파이썬 콘솔 산출물로 포워딩할 것입니다. 하위 프로세스가 실행되는 동안, 파이썬 콘솔 입력은 표준 입력 (STDIN) 모드로 전환되어 입력되는 문자들을 하위 프로세스로 포워딩합니다. 즉 하위 프로세스가 확인을 요청할 때 이를 전송해줄 수 있습니다. 콘솔이 STDIN 모드일 때 `Ctrl+C` 조합키를 누르면 하위 프로세스를 중단시킵니다. `var = !cmd` 구문을 사용하면 변수에 대한 명령어 결과에 영향을 줄 수도 있습니다.

```
>>> !echo QGIS Rocks!
QGIS Rocks

>>> !gdalinfo --version
GDAL 3.10.3, released 2025/04/01

>>> !pip install black
```

(다음 페이지에 계속)

(이전 페이지에서 계속)

```
# Install black python formatter using pip (if available)

>>> sql_formats = !ogrinfo --formats | grep SQL
>>> sql_formats
['SQLite -vector- (rw+v): SQLite / Spatialite', '  MSSQLSpatial -vector-␣
↪(rw+): Microsoft SQL Server Spatial Database', '  PostgreSQL -vector-␣
↪(rw+): PostgreSQL/PostGIS', '  MySQL -vector- (rw+): MySQL', '  PGDUMP -
↪vector- (w+v): PostgreSQL SQL dump']
```

**팁**

**출력 패널에서 실행한 명령어 재사용**

출력 패널에서 일부 텍스트를 선택한 다음 `Ctrl+E` 조합키를 누르면 코드 조각을 실행할 수 있습니다. 선택한 텍스트가 인터프리터 프롬프트 (>>>, . . .) 를 담고 있어도 상관없습니다.

### 25.3.2 코드 편집기

Use the Show Editor button in the Interactive Console to enable the editor widget. It allows editing and saving Python scripts and offers advanced functionalities to manage your code. Depending on the *enabled settings*, it provides various capabilities for easier code writing, such as code completion, highlighting syntax and calltips for supported APIs. Automatic indentation, parenthesis insertion, code commenting and syntax checking are also available.

The code editor area allows to simultaneously work on different scripts, each in a specific tab. Press New editor to add a new tab. You can run partially or totally a script from the *Code Editor* and output the result in the Interactive Console output area.

**팁**

Press `Ctrl+Space` to view the auto-completion list.

At the top of the dialog, a toolbar provides access to a few commands. Right-clicking the editor area provides some more options. All available tools are described in the following table.

표 25.1: Python Console Editor Tools

| Tool name | Function | Location |
|---|---|---|
| *Open script…* | Loads a Python file in the code editor dialog, as a new tab | 툴바 |
| *Open in external editor…* | Opens a saved Python script in the default external program set for Python file editing | |
| *Save* (`Ctrl+S`) | Saves the current script | |
| *Save as…* (`Ctrl+Shift+S`) | Saves the current script as a new file | |
| *Run script* (`Ctrl+Shift+E`) | Executes the whole script in the Interactive console (this creates a byte-compiled file with the extension `.pyc`) | Toolbar & Contextual menu |

다음 페이지에 계속

표 25.1 – 이전 페이지에서 계속

| Tool name | Function | Location |
|---|---|---|
| *Context help* (F1) | Attempts to display help on the selected string (class, method, object, ⋯) in its corresponding API documentation | Contextual menu |
| *Run selected* (Ctrl+E) | Executes in the Interactive console the lines selected in the script | Toolbar & Contextual menu |
| *Cut* (Ctrl+X) | Cuts selected text to the clipboard | |
| *Copy* (Ctrl+C) | Copies selected text to the clipboard | |
| *Paste* (Ctrl+V) | Pastes a cut or copied text | |
| *Find text* | Allows to search and replace a text in the script.<br>• 데스크탑 환경 기본 단축키 (Ctrl+G 및 Shift+Ctrl+G)를 사용해서 다음/이전 찾기<br>• 검색창 입력 시 첫 번째 일치 항목을 자동으로 찾기<br>• 검색창을 열 때 처음 찾은 문자열을 선택하도록 설정하기<br>• Enable the *Use Regular Expressions* option in the search bar to use regular expressions instead of plain text matching;<br>• Replace the matching text with the text in the replace box. When using a regular expression, you can reference captured groups (e.g. via \1) in the replacement text. Use *Replace All* to replace all matches at once.<br>• Esc 키를 눌러 검색창 닫기 | |
| *Toggle comment* (Ctrl+:) | Comments out or uncomment selected lines, by adding or removing # character (potentially followed by a space) at the start of the lines. If the selection contains both comments and actual commands, adds # at the start of each non empty line. Pressing again the button reverts to the initial state. | |
| *Reformat code* (Ctrl+Alt+F) | Allows to manually apply various formatting rules (sort imports, indentation, line length,⋯) to the code, following *user-defined settings*. This may require installation of additional Python modules. | |
| *Object inspector* ⋯ | Shows and hides a dedicated browser with a tree structure for classes and functions available in the script. Click an item fo a quick access to its definition. The tool requires an activation from the *Python settings - Run and Debug*. | |
| *Hide editor* | Hides the Python code editor block. To make it visible again, press Show editor button from the interactive console toolbar. | Contextual menu |
| *Check syntax* (Ctrl+4) | Browses the code and reports syntax errors, such as missing parenthesis, colons, wrong indentation,⋯ | |
| *Undo* (Ctrl+Z) | Undoes the latest action | |
| *Redo* (Ctrl+Shift+Z) | Reverts undone actions to a more recent | |
| *Select all* (Ctrl+A) | Selects the whole script | |

다음 페이지에 계속

표 25.1 – 이전 페이지에서 계속

| Tool name | Function | Location |
|---|---|---|
| *Share on GitHub* | Shares the script as a *Secret Gist* or *Public Gist* on GitHub, provided a *GitHub access token*. | |
| *Options…* | Opens the *Python settings* dialog. | |

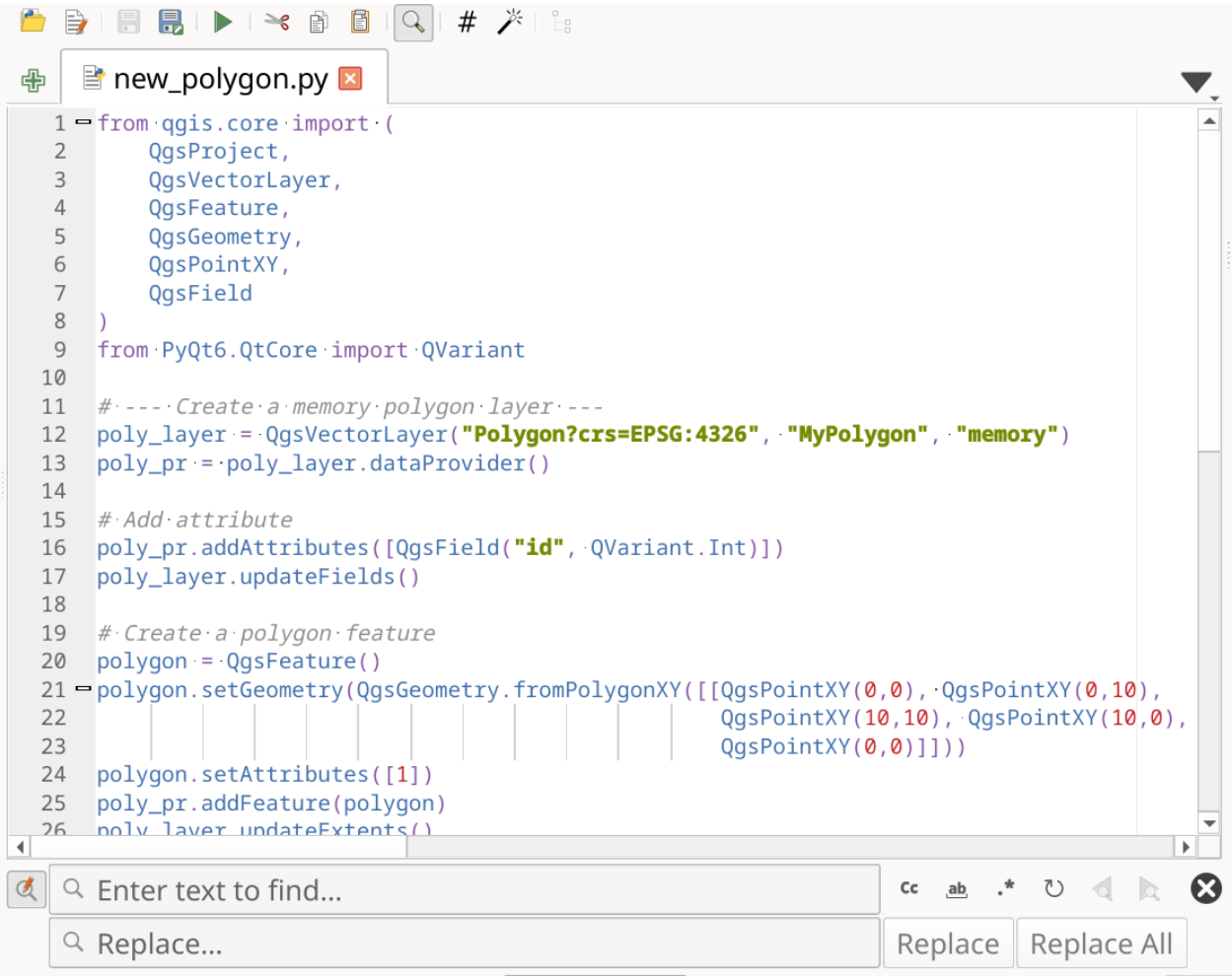



그림 25.18: 파이썬 콘솔 편집기

CHAPTER 26

# 부록

## 26.1 부록 C: QGIS 파일 포맷

### 26.1.1 QGS/QGZ ? QGIS 프로젝트 파일 포맷

**QGS** 포맷은 QGIS 프로젝트를 저장하기 위한 XML 포맷입니다. **QGZ** 포맷은 QGS 파일 및 QGD 파일을 담고 있는 압축 (zip) 보관 파일입니다. **QGD** 파일은 프로젝트를 위한 보조 (auxiliary) 데이터를 담고 있는 QGIS 프로젝트 관련 SQLite 데이터베이스입니다. 아무 보조 데이터도 없는 경우, QGD 파일도 비어 있을 것입니다.

QGS 파일은 QGIS 프로젝트를 저장하기 위해 필요한, 다음을 포함하는 모든 것을 담고 있습니다:

- 프로젝트 제목
- 프로젝트 좌표계
- 레이어 트리
- 스냅 작업 설정
- 관계
- 맵 캔버스 범위
- 프로젝트 모델
- legend
- 맵뷰 도크 (dock) (2 차원 및 3 차원)
- 기저 데이터셋 (데이터소스) 와의 링크를 보유한 레이어 및 범위, 좌표계, 결합 (join), 스타일, 렌더링 작업자, 혼한 모드, 투명도 등을 포함하는 기타 레이어 속성
- 프로젝트 속성

다음 그림들은 QGS 파일에 있는 최상위 수준의 태그와, `ProjectLayers` 태그를 펼친 모습을 보여주고 있습니다.




그림 26.1: QGS 파일의 최상위 수준 태그들

```
−<projectlayers>
  −<maplayer styleCategories="AllStyleCategories" readOnly="0" autoRefreshTime="0" autoRefreshEnabled="0" refreshOnNotifyEnabled="0" maxScale="0"
   geometry="Polygon" labelsEnabled="0" type="vector" simplifyDrawingHints="1" hasScaleBasedVisibilityFlag="0" simplifyDrawingTol="1"
   simplifyMaxScale="1" minScale="1e+8" simplifyAlgorithm="0" simplifyLocal="1" refreshOnNotifyMessage="">
    +<extent></extent>
     <id>watersheds_b62efa19_8809_4406_b6ec_2951ac4c94c5</id>
    −<datasource>
       ./QGIS-Training-Data-2.0/exercise_data/processing/generalize/watersheds.shp
     </datasource>
    +<keywordList></keywordList>
     <layername>watersheds</layername>
    +<srs></srs>
    +<resourceMetadata></resourceMetadata>
     <provider encoding="UTF-8">ogr</provider>
     <vectorjoins/>
     <layerDependencies/>
     <dataDependencies/>
     <legend type="default-vector"/>
     <expressionfields/>
    +<map-layer-style-manager current="default"></map-layer-style-manager>
     <auxiliaryLayer/>
    +<flags></flags>
    +<renderer-v2 symbollevels="0" enableorderby="0" type="singleSymbol" forceraster="0"></renderer-v2>
    +<customproperties></customproperties>
     <blendMode>0</blendMode>
     <featureBlendMode>0</featureBlendMode>
     <layerOpacity>1</layerOpacity>
    +<SingleCategoryDiagramRenderer diagramType="Histogram" attributeLegend="1"></SingleCategoryDiagramRenderer>
    +<DiagramLayerSettings priority="0" linePlacementFlags="18" dist="0" showAll="1" placement="1" obstacle="0" zIndex="0"></DiagramLayerSettings>
    +<geometryOptions removeDuplicateNodes="0" geometryPrecision="0"></geometryOptions>
    +<fieldConfiguration></fieldConfiguration>
    +<aliases></aliases>
     <excludeAttributesWMS/>
     <excludeAttributesWFS/>
    +<defaults></defaults>
    +<constraints></constraints>
    +<constraintExpressions></constraintExpressions>
     <expressionfields/>
    +<attributeactions></attributeactions>
    +<attributetableconfig actionWidgetStyle="dropDown" sortExpression="" sortOrder="0"></attributetableconfig>
    +<conditionalstyles></conditionalstyles>
     <editform tolerant="1"/>
     <editforminit/>
     <editforminitcodesource>0</editforminitcodesource>
     <editforminitfilepath/>
    +<editforminitcode></editforminitcode>
     <featformsuppress>0</featformsuppress>
     <editorlayout>generatedlayout</editorlayout>
    +<editable></editable>
    +<labelOnTop></labelOnTop>
     <widgets/>
     <previewExpression>ID</previewExpression>
     <mapTip/>
   </maplayer>
 </projectlayers>
```

그림 26.2: QGS 파일의 펼쳐진 최상위 수준 `ProjectLayers` 태그

### 26.1.2 QLR 파일 QGIS 레이어 정의 파일

레이어 정의 파일 (**QLR**) 은 레이어 데이터소스를 가리키는 포인터는 물론 레이어에 대한 QGIS 스타일 정보도 담고 있는 XML 파일입니다.

이 파일의 활용 사례는 간단합니다. 데이터소스를 열고 관련 스타일 정보를 모두 가져오기 위한 단일 파일을 가지게 되는 것이죠. QLR 파일을 사용하면, 파일을 연다는 간단한 방법으로 기저 데이터소스도 감출 수 있습니다.

QLR 활용의 예시로 MS SQL 레이어를 열어보겠습니다. MS SQL 연결 대화창을 열고, 연결하고, 선택하고, 불러와서 마지막으로 스타일을 적용하는 대신, 그냥 필요한 모든 스타일과 함께 올바른 MS SQL 레이어를 가리키는 `.qlr` 파일만 추가하면 됩니다.

향후 `.qlr` 파일이 하나 이상의 레이어를 참조하게 될 수도 있습니다.

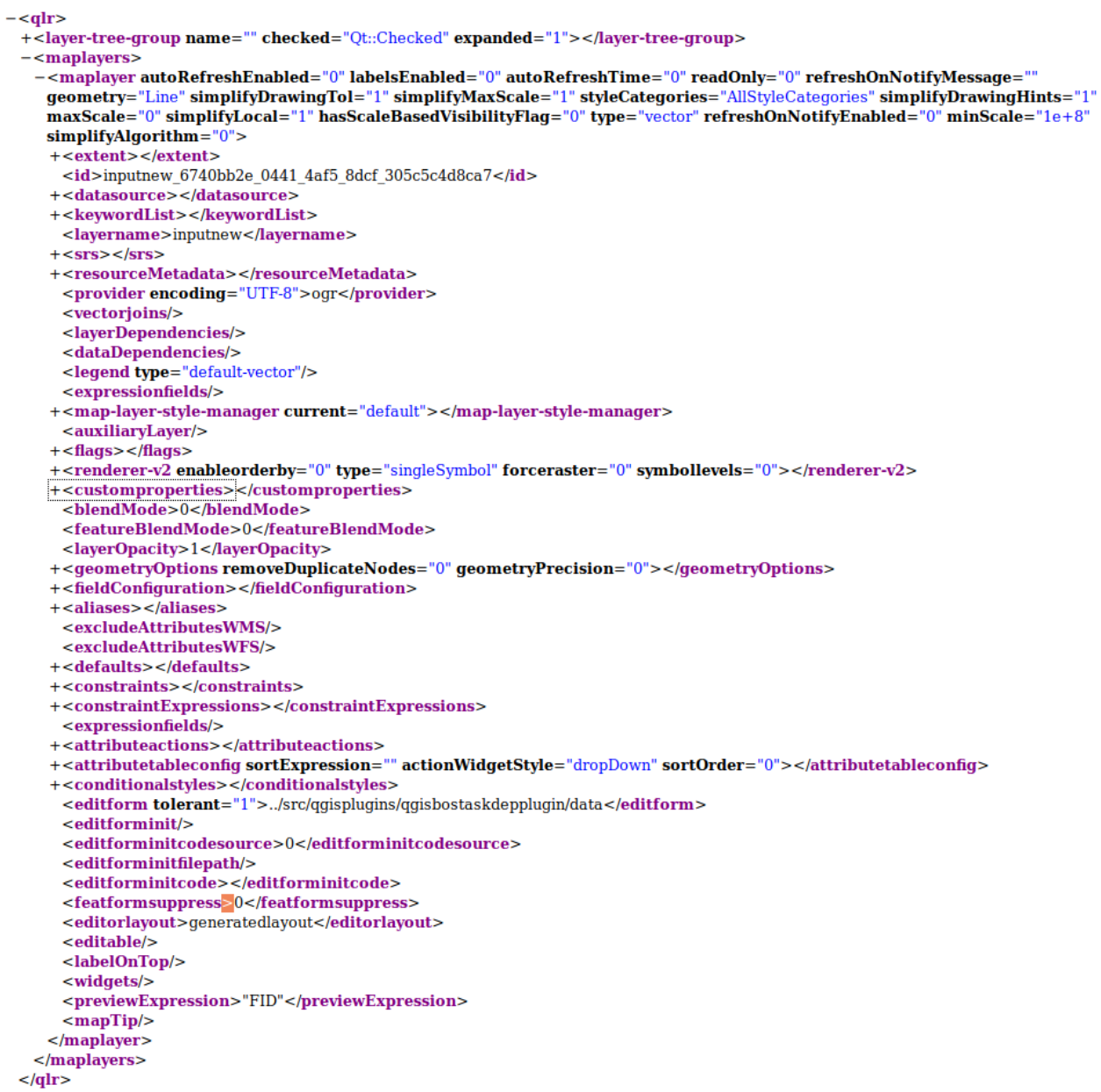



그림 26.3: QLR 파일의 최상위 수준 태그들

### 26.1.3 QML 와 QGIS 스타일 파일 포맷

**QML** 은 레이어 스타일을 저장하기 위한 XML 포맷입니다.

QML 파일은 피처 도형의 렌더링을 위해 QGIS 가 처리할 수 있는 — 심볼의 정의, 크기 및 기울기, 라벨, 투명도 그리고 혼합 모드 등을 포함하는 — 모든 정보를 담고 있습니다.

다음 그림은 QML 파일에 있는 최상위 수준의 태그를 보여주고 있습니다. (`renderer_v2` 및 그 하위 `symbol` 태그만 펼쳐져 있습니다.)

## 26.2 부록 D: QGIS R 스크립트 문법

Contributed by Matteo Ghetta - funded by Scuola Superiore Sant'Anna

공간 처리 프레임워크에서 R 스크립트를 작성하는 일은 그 특수 문법 때문에 조금 까다로울 수도 있습니다.

공간 처리 R 스크립트는 그 **입력물** 과 **산출물** 을 정의하는 것으로 시작합니다. 각각 그 앞에 2 중 해시 문자 (`##`) 를 삽입해야 합니다.

입력물 앞에, 알고리즘을 배치할 그룹을 지정할 수 있습니다. 이미 존재하는 그룹이라면 알고리즘이 해당 그룹에 추가될 것입니다. 존재하지 않는 그룹이라면 해당 그룹을 생성할 것입니다. 다음 예시에서 그룹의 명칭은 *My group* 입니다:

```
##My Group=group
```

### 26.2.1 입력물

모든 입력 데이터 및 파라미터를 지정해야 합니다. 입력물의 유형은 여럿 있습니다:

- 벡터: `##Layer = vector`
- 벡터 필드: `##F = Field Layer` (여기서'Layer'는 필드가 속한 입력 벡터 레이어의 명칭입니다)
- 래스터: `##r = raster`
- 테이블: `##t = table`
- 숫자: `##Num = number`
- 문자열: `##Str = string`
- 불 (boolean): `##Bol = boolean`
- 드롭다운 메뉴의 항목들은 쌍반점 `;` 으로 구분되어야만 합니다: `##type=selection point;lines; point+lines`

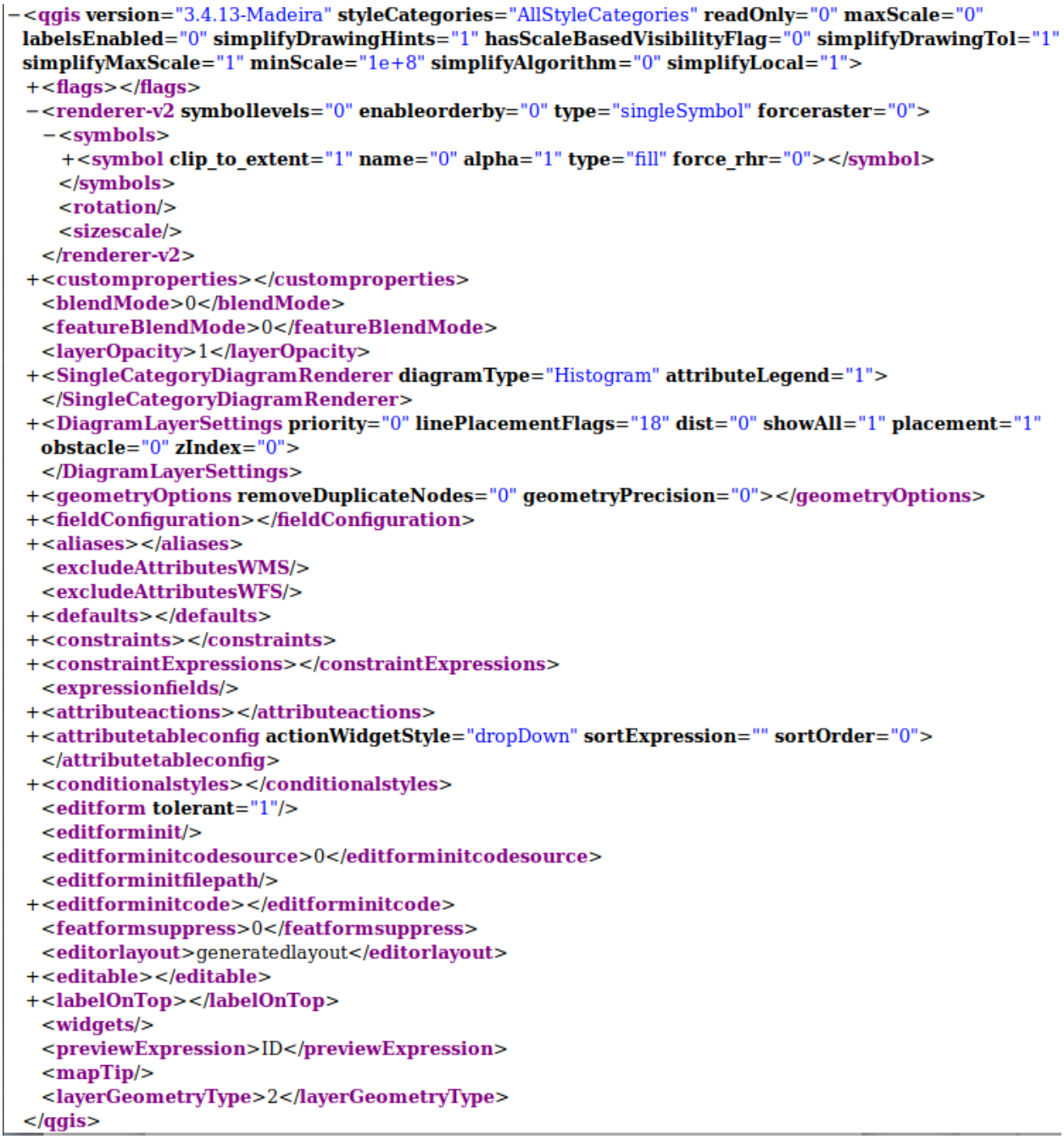



그림 26.4: QML 파일의 최상위 수준 태그들 (`renderer_v2` 및 그 하위 `symbol` 태그만 펼쳐져 있습니다)

### 26.2.2 산출물

입력물의 경우와 마찬가지로, 스크립트 시작 부분에서 각 산출물을 정의해야 합니다.

- 벡터: `##output= output vector`
- 래스터: `##output= output raster`
- 테이블: `##output= output table`
- 플롯 (plot): `##output_plots_to_html` (이전 버전에서의 ##showplots)
- 결과물 뷰어 에 R 산출물을 표시하려면, 사용자가 표시하고자 하는 산출물을 생산하는 명령어 앞에 > 문자를 삽입하십시오.

### 26.2.3 QGIS R 스크립트용 문법 요약

수많은 입력 및 산출 파라미터 유형을 지원합니다.

#### 입력 파라미터 유형

| 파라미터 | 문법 예시 | 반환하는 객체 |
|---|---|---|
| vector | Layer = vector | sf 객체 (또는 ##load_vector_using_rgdal 을 지정한 경우 SpatialDataFrame 객체) |
| vector point | Layer = vector point | sf 객체 (또는 ##load_vector_using_rgdal 을 지정한 경우 SpatialDataFrame 객체) |
| vector line | Layer = vector line | sf 객체 (또는 ##load_vector_using_rgdal 을 지정한 경우 SpatialDataFrame 객체) |
| vector polygon | Layer = vector polygon | sf 객체 (또는 ##load_vector_using_rgdal 을 사용하는 경우 SpatialPolygonsDataFrame 객체) |
| multiple vector | Layer = multiple vector | sf 객체 (또는 ##load_vector_using_rgdal 을 지정한 경우 SpatialDataFrame 객체) |
| table | Layer = table | CVS 에서 변환한 데이터프레임으로 `read.csv` 함수의 기본 객체 |
| field | Field = Field Layer | 선택한 필드의 명칭, 예: `"Area"` |
| raster | Layer = raster | RasterBrick 객체로 `raster` 패키지의 기본 객체 |
| multiple raster | Layer = multiple raster | RasterBrick 객체로 `raster` 패키지의 기본 객체 |
| number | N = number | 정수 또는 부동소수점 가운데 선택 |
| string | S = string | 텍스트란에 입력한 문자열 |
| longstring | LS = longstring | 텍스트란에 입력한 문자열로 일반 문자열보다 길 수 있음 |
| selection | S = selection first;second;third | 드롭다운 메뉴에서 선택한 항목의 문자열 |
| crs | C = crs | 선택한 좌표계의 문자열, 서식: `"EPSG:4326"` |
| extent | E = extent | `raster` 패키지의 범위 객체로, 값을 `E@xmin` 으로 추출 가능 |
| point | P = point | 맵을 클릭한 경우, 클릭한 포인트의 좌표 |
| file | F = file | 선택한 파일을 가리키는 경로, 예: “/home/matteo/file.txt” |
| folder | F = folder | 선택한 폴더를 가리키는 경로, 예: “/home/matteo/Downloads” |

파라미터는 **부가적 (optional)** 일 수 있습니다. 무시할 수 있다는 뜻입니다.

입력물을 부가적인 것으로 설정하려면, 다음과 같이 입력물 **앞에** `optional` 문자열을 추가하십시오:

```
##Layer = vector
##Field1 = Field Layer
##Field2 = optional Field Layer
```

#### 산출 파라미터 유형

| 파라미터 | 문법 예시 |
|---|---|
| vector | Output = output vector |
| raster | Output = output raster |
| table | Output = output table |
| file | Output = output file |

> **참고**
>
> 공간 처리 결과물 뷰어 에서 플롯을 `.png` 포맷으로 저장할 수도 있고, 또는 알고리즘 인터페이스에서 직접 저장할 수도 있습니다.

#### 스크립트 본문

스크립트 본문은 R 문법을 따르며, 사용자 스크립트에 잘못된 점이 있을 경우 **로그** 패널에서 그 원인을 찾아볼 수 있습니다.

스크립트에서 모든 추가 라이브러리를 불러와야 한다는 점을 **기억하십시오**:

```
library(sp)
```

### 26.2.4 예제

#### 벡터 산출물 예시

온라인 저장소에서 입력 레이어의 범위 안에서 랜덤한 포인트들을 생성하는 알고리즘을 선택해봅시다:

```
##Point pattern analysis=group
##Layer=vector polygon
##Size=number 10
##Output=output vector
library(sp)
spatpoly = as(Layer, "Spatial")
pts=spsample(spatpoly,Size,type="random")
spdf=SpatialPointsDataFrame(pts, as.data.frame(pts))
Output=st_as_sf(spdf)
```

설명 (스크립트의 각 줄별로):

1. `Point pattern analysis` 는 알고리즘이 속한 그룹입니다.
2. `Layer` 는 입력 **벡터** 레이어입니다.
3. `Size` 는 기본값이 10 인 **숫자** 파라미터입니다.

4. `Output` 은 알고리즘이 생성할 **벡터** 레이어입니다.
5. `library(sp)` 는 **sp** 라이브러리를 불러옵니다.
6. `spatpoly = as(Layer, "Spatial")` 은 입력 레이어를 **sp** 객체로 변형합니다.
7. `sp` 라이브러리의 `spsample` 함수를 호출, 앞에서 정의한 입력물을 (`Layer` 및 `Size` 를) 사용해서 실행합니다.
8. `SpatialPointsDataFrame` 함수를 사용해서 *SpatialPointsDataFrame* 객체를 생성합니다.
9. `st_as_sf` 함수를 사용해서 산출 벡터 레이어를 생성합니다.

끝입니다! QGIS 범례에 있는 벡터 레이어와 함께 알고리즘을 실행하고, 랜덤 포인트의 개수를 선택하십시오. 산출되는 레이어가 사용자 맵에 추가될 것입니다.

### 래스터 산출물 예시

다음 스크립트는 평범한 기본 크리그 격자법 (kriging) 을 수행합니다. `automap` R 패키지의 `autoKrige` 함수를 사용해서 입력 포인트 벡터 레이어의 지정 필드로부터 보간한 값들의 래스터 맵을 생성할 것입니다. 먼저 크리그 격자법 모델을 계산한 다음 래스터를 생성할 것입니다. 이 래스터는 `raster` R 패키지의 `raster` 함수를 통해 생성됩니다.

```
##Basic statistics=group
##Layer=vector point
##Field=Field Layer
##Output=output raster
##load_vector_using_rgdal
require("automap")
require("sp")
require("raster")
table=as.data.frame(Layer)
coordinates(table)= ~coords.x1+coords.x2
c = Layer[[Field]]
kriging_result = autoKrige(c~1, table)
prediction = raster(kriging_result$krige_output)
Output<-prediction
```

`##load_vector_using_rgdal` 을 사용하면, 입력 벡터 레이어를 `SpatialDataFrame` 객체로 사용할 수 있게 됩니다. `sf` 객체에서 변형해야 할 필요가 없어집니다.

### 테이블 산출물 예시

`Summary Statistics` 알고리즘이 테이블 파일 (`.csv`) 을 산출하도록 편집해봅시다.

스크립트 본문은 다음과 같습니다:

```
##Basic statistics=group
##Layer=vector
##Field=Field Layer
##Stat=Output table
Summary_statistics<-data.frame(rbind(
    sum(Layer[[Field]]),
    length(Layer[[Field]]),
    length(unique(Layer[[Field]])),
    min(Layer[[Field]]),
    max(Layer[[Field]]),
```

(다음 페이지에 계속)

(이전 페이지에서 계속)

```
    max(Layer[[Field]])-min(Layer[[Field]]),
    mean(Layer[[Field]]),
    median(Layer[[Field]]),
    sd(Layer[[Field]]),
  row.names=c("Sum:","Count:","Unique values:","Minimum value:","Maximum value:",
↪"Range:","Mean value:","Median value:","Standard deviation:"))
colnames(Summary_statistics)<-c(Field)
Stat<-Summary_statistics
```

세 번째 줄에서 입력물의 **벡터 필드** 를 지정하고, 네 번째 줄에서 알고리즘에 테이블을 산출해야 한다고 알려줍니다. 마지막 줄에서 스크립트가 생성한 `Stat` 객체를 받아 `.csv` 테이블로 변환할 것입니다.

### 콘솔 산출물 예시

앞의 예시를 사용해서 테이블을 생성하는 대신 **결과물 뷰어** 에 결과물을 출력해보겠습니다:

```
##Basic statistics=group
##Layer=vector
##Field=Field Layer
Summary_statistics<-data.frame(rbind(
sum(Layer[[Field]]),
length(Layer[[Field]]),
length(unique(Layer[[Field]])),
min(Layer[[Field]]),
max(Layer[[Field]]),
max(Layer[[Field]])-min(Layer[[Field]]),
mean(Layer[[Field]]),
median(Layer[[Field]]),
sd(Layer[[Field]])),row.names=c("Sum:","Count:","Unique values:","Minimum value:",
↪"Maximum value:","Range:","Mean value:","Median value:","Standard deviation:"))
colnames(Summary_statistics)<-c(Field)
>Summary_statistics
```

이 스크립트는 두 군데 수정 사항을 제외하면 앞의 것과 완전히 동일합니다:

1. 산출물을 지정하지 않았습니다. (네 번째 줄을 제거했습니다.)
2. 마지막 줄 맨 앞에 > 기호를 삽입했습니다. 결과물 뷰어를 통해 사용할 수 있는 객체를 생성해야 한다고 공간처리 프레임워크에 알려주는 것입니다.

### 플롯 예시

플롯을 생성하려면, 다음 스크립트에서와 같이 `##output_plots_to_html` 파라미터를 사용해야 합니다:

```
##Basic statistics=group
##Layer=vector
##Field=Field Layer
##output_plots_to_html
####output_plots_to_html
qqnorm(Layer[[Field]])
qqline(Layer[[Field]])
```

이 스크립트는 벡터 레이어 (`Layer`) 의 필드 (`Field`) 를 입력물로 사용해서 (분포의 정규성을 점검하기 위한) *QQ Plot* 을 생성합니다.

이 플롯은 공간 처리 결과물 뷰어 에 자동적으로 추가됩니다.

## 26.3 부록 E: QGIS 응용 프로그램 네트워크 연결

이 문서는 QGIS 가 만드는 사전 환경설정된/자동화된 네트워크 연결 목록입니다. 일부 연결은 연결되기 전 사용자의 액션을 요구하기 때문에 사용자가 시작해야 하고, 나머지는 자동으로 연결됩니다.

| 이름 | 목적 | UX 모드 | 서버 | 서버로 전송되는 정보 | 서버에 저장되는 정보 |
|---|---|---|---|---|---|
| **qgis.org** | | | | | |
| 파이썬 API 도움말 | PyQGIS 문서 탐색기 | 사용자 시작 | https://qgis.org/pyqgis/3.44/search | IP, QGIS 버전, OS | 서버 로그에 IP 저장 |
| **version.qgis.o** | | | | | |
| 새 버전 확인 | 사용자에게 새 QGIS 버전을 사용할 수 있다고 알림 | 자동 | https://version.qgis.org | IP, QGIS 버전, OS | 서버 로그에 IP 저장 |
| **feed.qgis.o** | | | | | |
| QGIS 피드 | 피드로부터 QGIS 뉴스 가져오기 | 자동 | https://feed.qgis.org | IP, QGIS 버전, 언어 코드, 마지막 다운로드의 타임스탬프, OS | 서버 로그에 있는 IP; QGIS 버전, OS 및 IP를 집계해서 몇몇 통계 수집에 사용 |
| **plugins.qgis.or** | | | | | |
| 플러그인 업데이트 확인 | 사용자에게 플러그인 업데이트 알림 | 사용자 시작/자동 (환경설정 가능) | https://plugins.qgis.org | IP, QGIS 버전, OS | 서버 로그에 IP 저장 |
| 플러그인 목록 | 플러그인 목록 가져오기 | 사용자 시작 | https://plugins.qgis.org | IP, QGIS 버전, OS | 서버 로그에 IP 저장 |
| 플러그인 설치 | 플러그인 패키지 다운로드 및 설치 | 사용자 시작 | https://plugins.qgis.org | IP, QGIS 버전, OS | 플러그인 다운로드 계수기를 1 올림 |
| 스타일 | 사용자 공헌 스타일 목록 | 사용자 시작 | https://hub.qgis.org/styles | IP, QGIS 버전, OS | 다운로드 계수기를 1 올림 |
| **제 3 자** | | | | | |
| 지형 데이터 | 3 차원 뷰를 위한 DEM 생성 | 사용자 시작 | https://s3.amazonaws.com/elevation-tiles-prod/terrarium/{z}/{x}/{y}. | IP, QGIS 버전, OS | 아마존 TOS 참조 |
| 구글 지도 지오코더 | 지오코딩 서비스 | 사용자 시작 | https://maps.googleapis.com/maps/api/geocode/json | IP, QGIS 버전, OS | 구글 지도 API TOS 참조 |
| 노미나팀 (Nominatim) 지오코더 | 지오코딩 서비스 | 사용자 시작 | https://nominatim.qgis.org/ui/search.html | IP, QGIS 버전, OS | |
| 측지 그리드 | 새 PROJ 그리드 추가 | 사용자 시작 | https://cdn.proj.org | IP, PROJ 버전 | 하루가 지나면 접근 로그를 영구히 삭제 |

CHAPTER 27

# 참고 문헌 및 웹사이트


GDAL-SOFTWARE-SUITE. Geospatial data abstraction library. https://gdal.org/en/latest, 2013.

GRASS-PROJECT. Geographic resource analysis support system. https://grass.osgeo.org, 2013.

NETELER, M., AND MITASOVA, H. Open source gis: A grass gis approach, 2008.

OGR-SOFTWARE-SUITE. Geospatial data abstraction library. https://gdal.org/en/latest, 2013.

OPEN-GEOSPATIAL-CONSORTIUM. Web map service (1.1.1) implementation specification. https://portal.ogc.org/files/?artifact_id=1081&version=1&format=pdf, 2002.

OPEN-GEOSPATIAL-CONSORTIUM. Web map service (1.3.0) implementation specification. https://portal.ogc.org/files/?artifact_id=14416&format=pdf, 2004.

POSTGIS-PROJECT. Spatial support for postgresql. http://www.refractions.net/products/postgis/, 2013.