

QGIS Desktop 3.16 User Guide

QGIS Project

2022 년 04 월 02 일

Contents

1 들어가며 **1**
- 1.1 QGIS 3.16 에서 달라진 점들 . . . 2

2 머리말 **3**

3 표기 방법 **5**
- 3.1 GUI 표기 방법 . . . 5
- 3.2 텍스트 또는 키보드 표기 방법 . . . 6
- 3.3 OS 별 지침 . . . 6

4 주요 기능 **7**
- 4.1 데이터 보기 . . . 7
- 4.2 데이터 탐색 및 지도 제작 . . . 8
- 4.3 데이터 생성, 편집, 관리 및 내보내기 . . . 8
- 4.4 데이터 분석 . . . 9
- 4.5 온라인 맵 발행 . . . 9
- 4.6 플러그인을 통한 확장 QGIS 기능 . . . 9
- 4.7 파이썬 콘솔 . . . 10
- 4.8 알려진 문제점들 . . . 10

5 시작하기 **11**
- 5.1 QGIS 설치 . . . 11
- 5.2 QGIS 실행 및 종료 . . . 12
- 5.3 예시 세션: 래스터 및 벡터 레이어 불러오기 . . . 13

6 프로젝트 파일 작업 **19**
- 6.1 QGIS 프로젝트 소개 . . . 19
- 6.2 엉터리 파일 경로 처리 . . . 22
- 6.3 산출물 생성 . . . 22

7 QGIS GUI **23**
- 7.1 메뉴 바 . . . 24
- 7.2 패널 및 툴바 . . . 39
- 7.3 맵 뷰 . . . 42
- 7.4 3D 맵 뷰 . . . 46
- 7.5 상태 바 . . . 53

8 탐색기 패널 **55**
8.1 탐색기에서 열/실행할 수 있는 리소스들 . . . 58
8.2 탐색기 패널 최상위 수준 항목들 . . . 58
8.3 리소스 . . . 62

9 QGIS 환경 설정 **65**
9.1 옵션 . . . 65
9.2 사용자 프로파일 작업 . . . 93
9.3 프로젝트 속성 . . . 95
9.4 사용자 지정 . . . 103
9.5 키보드 단축키 . . . 106
9.6 고급 설정으로 QGIS 실행 . . . 107

10 투영 작업 **113**
10.1 투영 지원 개요 . . . 113
10.2 레이어 좌표계 . . . 113
10.3 프로젝트 좌표계 . . . 115
10.4 좌표계 선택기 . . . 117
10.5 사용자 정의 좌표계 . . . 117
10.6 기준 (datum) 변환 . . . 119

11 일반 도구 **123**
11.1 컨텍스트 도움말 . . . 123
11.2 패널 . . . 123
11.3 프로젝트 내포 작업 . . . 133
11.4 맵 캔버스 작업 . . . 134
11.5 대화형 피처 작업 . . . 151
11.6 레이어 속성 저장 및 공유 . . . 159
11.7 값을 변수로 저장 . . . 162
11.8 인증 . . . 164
11.9 일반 위젯 . . . 164

12 스타일 라이브러리 **175**
12.1 스타일 관리자 . . . 175
12.2 심볼 선택기 . . . 183
12.3 라벨 설정 . . . 193
12.4 3D 심볼 생성 . . . 210

13 데이터소스 관리 **215**
13.1 데이터 열기 . . . 215
13.2 레이어 생성 . . . 242
13.3 데이터 포맷 및 필드 탐구 . . . 257

14 벡터 데이터 작업 **267**
14.1 벡터 속성 대화창 . . . 267
14.2 표현식 . . . 343
14.3 함수 목록 . . . 349
14.4 속성 테이블 작업 . . . 495
14.5 편집 작업 . . . 519

15 래스터 데이터 작업 **551**
15.1 래스터 속성 대화창 . . . 551
15.2 래스터 분석 . . . 570
15.3 지리참조자 . . . 572

16 메시 데이터 작업 **579**
16.1 메시란 무엇일까요? . . . 579
16.2 지원 포맷 . . . 581
16.3 메시 데이터셋 속성 . . . 582

17 벡터 타일 작업 **589**
17.1 벡터 타일이란? . . . 589
17.2 지원 포맷 . . . 589

18 맵 조판 작업 **591**
18.1 인쇄 조판기 개요 . . . 591
18.2 조판기 항목 . . . 608
18.3 출력물 생성 작업 . . . 653
18.4 보고서 생성 작업 . . . 664

19 OGC / ISO 프로토콜 작업 **683**
19.1 WMS/WMTS 클라이언트 . . . 684
19.2 WCS 클라이언트 . . . 692
19.3 WFS 및 WFS-T 클라이언트 . . . 693

20 GPS 데이터 작업 **697**
20.1 GPS 플러그인 . . . 697
20.2 실시간 GPS 추적 . . . 702

21 인증 시스템 **707**
21.1 인증 시스템 개요 . . . 707
21.2 사용자 인증 작업 흐름 . . . 717
21.3 보안 고려 사항 . . . 731

22 GRASS GIS 통합 **733**
22.1 예제 데이터셋 . . . 733
22.2 GRASS 래스터 및 벡터 레이어 불러오기 . . . 734
22.3 드래그 & 드롭으로 GRASS 로케이션에 데이터 가져오기 . . . 734
22.4 QGIS 탐색기에서 GRASS 데이터 관리하기 . . . 734
22.5 GRASS 옵션 . . . 735
22.6 GRASS 플러그인 시작하기 . . . 735
22.7 GRASS 맵셋 열기 . . . 735
22.8 GRASS 로케이션 및 맵셋 . . . 735
22.9 GRASS 로케이션에 데이터 가져오기 . . . 736
22.10 GRASS 벡터 데이터 모델 . . . 739
22.11 새 GRASS 벡터 레이어 생성하기 . . . 739
22.12 GRASS 벡터 레이어 디지타이즈 및 편집하기 . . . 740
22.13 GRASS 영역 도구 . . . 742
22.14 GRASS 툴박스 . . . 742

23 QGIS 공간 처리 프레임워크 **751**
23.1 소개 . . . 751
23.2 공간 처리 프레임워크 환경 설정 . . . 753
23.3 툴박스 . . . 756
23.4 이력 관리자 . . . 764
23.5 그래픽 모델 생성기 . . . 765
23.6 배치 프로세스 인터페이스 . . . 780
23.7 콘솔에서 공간 처리 알고리즘 사용 . . . 782
23.8 명령 줄에서 공간 처리 알고리즘 사용 . . . 792
23.9 새 공간 처리 알고리즘을 파이썬 스크립트로 작성 . . . 793

23.10 외부 응용 프로그램 환경 설정 . . . 803

24 공간 처리 제공자 및 알고리즘 815
24.1 QGIS 알고리즘 제공자 . . . 815
24.2 GDAL 알고리즘 제공자 . . . 1198
24.3 LAStools 알고리즘 제공자 . . . 1285
24.4 TauDEM 알고리즘 제공자 . . . 1323
24.5 OTB 응용 프로그램 제공자 . . . 1380

25 플러그인 1381
25.1 QGIS 플러그인 . . . 1381
25.2 QGIS 핵심 플러그인 사용하기 . . . 1388
25.3 QGIS 파이썬 콘솔 . . . 1406

26 도움 및 지원 1411
26.1 메일링 리스트 . . . 1411
26.2 IRC . . . 1412
26.3 상업적 지원 . . . 1412
26.4 버그 추적기 . . . 1412
26.5 블로그 . . . 1413
26.6 플러그인 . . . 1413
26.7 위키 . . . 1413

27 공헌자 1415
27.1 저자 . . . 1415
27.2 번역자 . . . 1417
27.3 번역 통계 . . . 1419

28 부록 1421
28.1 부록 A: GNU 일반 공중 사용 허가서 . . . 1421
28.2 부록 B: GNU 자유 문서 사용 허가서 . . . 1425
28.3 부록 C: QGIS 파일 포맷 . . . 1430
28.4 부록 D: QGIS R 스크립트 문법 . . . 1434

29 참고 문헌 및 웹사이트 1441

CHAPTER 1

들어가며

이 문서는 지리 정보 시스템 (GIS; Geographical Information System) 소프트웨어인 QGIS 의 사용자 지침서입니다. QGIS 는 GNU 일반 공중 사용 허가 (GNU General Public License) 를 따릅니다. QGIS 홈페이지 https://www.qgis.org 에서 더 자세한 정보를 찾아볼 수 있습니다.

이 문서의 내용은 해당 분야에 가장 해박한 저자와 편집자가 작성하고 검증한 것입니다. 그럼에도 불구하고, 잘못된 내용이 있을 수도 있습니다.

그렇기에 저자, 편집자, 출판자는 이 문서의 오류와 그에 따른 결과에 대해 어떤 책임이나 의무를 지지 않습니다. 틀렸다고 생각되는 내용이 있다면 언제든지 지적해주십시오.

QGIS 사용자 지침서는 reST(reStructuredText) 를 이용해 조판되었습니다. github 에서 이 문서의 reST 소스코드를 구할 수 있으며, 온라인으로는 https://www.qgis.org/ko/docs/ 웹페이지에서 HTML 문서를 보거나 PDF 문서를 다운로드할 수 있습니다. QGIS 프로젝트의 문서 영역에서 이 문서의 번역본들도 검색하고 다운로드할 수 있습니다.

이 문서의 작성 및 번역에 참여하고 싶다면, https://qgis.org/ko/site/getinvolved/index.html 웹페이지를 방문해주십시오.

이 문서 내의 링크

이 문서는 내부 및 외부 링크를 포함하고 있습니다. 내부 링크를 클릭하면 문서 내의 해당 위치로 이동합니다. 외부 링크를 클릭하면 인터넷 주소가 열립니다.

이 문서의 저자와 편집자

공헌자 에 이 사용자 지침서를 작성하고, 검토하고 번역하는 데 공헌한 사람들의 목록이 있습니다.

Copyright (c) 2004 - 2020 QGIS 개발팀

인터넷: https://www.qgis.org

이 문서의 라이선스

자유 소프트웨어 재단이 발행한 버전 1.3 이상의 GNU 자유 문서 사용 허가서의 조건에 따라 이 문서를 장절항목 변경 없이, 앞표지 및 뒤표지 텍스트 추가 없이 복사, 배포, 그리고/또는 수정할 수 있습니다. 이 라이선스의 복사본은 부록부록 *B: GNU* 자유 문서 사용 허가서 에 포함되어 있습니다.

1.1 QGIS 3.16 에서 달라진 점들

이번 QGIS 배포판에서 QGIS 3.10 에 비해 버그 수백여 개를 수정했으며, 많은 새 기능 및 개선 사항을 포함하고 있습니다. 이전 버전보다 이번 버전을 사용하는 것을 추천합니다. 새 기능 목록을 알고 싶다면, https://qgis.org/ko/site/forusers/visualchangelogs.html 에서 시각적 변경 로그를 살펴보십시오.

CHAPTER 2

머리말

경이로운 지리 정보 시스템 (GIS) 의 세계로 오신 것을 환영합니다!

QGIS 는 오픈소스 지리 정보 시스템입니다. 이 프로젝트는 2002 년 5 월에 시작되어 같은 해 6 월 SourceForge 상의 프로젝트로 만들어졌습니다. 우리는 (전통적으로 값비싼 독점 소프트웨어인) GIS 소프트웨어를 개인용 컴퓨터를 쓸 수 있는 사람이라면 누구나 이용할 수 있도록 하기 위해 최선을 다해왔습니다. 현재 QGIS 는 대부분의 유닉스 플랫폼, 윈도우 및 맥 OS 상에서 실행됩니다. QGIS 는 Qt 도구 모음 (https://www.qt.io/) 과 C++ 을 이용해서 개발됐습니다. 즉 QGIS 가 스마트하게 동작하며 기분 좋고 사용하기 쉬운 GUI(Graphical User Interface) 를 가지고 있다는 뜻입니다.

QGIS 는 일반적인 함수와 기능을 제공하는 사용자 친화적인 GIS 를 지향합니다. 이 프로젝트의 1 차 목표는 GIS 데이터 뷰어를 제공하는 것이었습니다. QGIS 가 전환기를 맞이한 것은 QGIS 가 일상적인 GIS 데이터 뷰어, 데이터 캡처, 고급 GIS 분석, 수준 높은 맵, 지도책 및 보고서 형태의 발표를 위해 이용되기 시작했을 때였습니다. 플러그인 아키텍처를 통해 새 포맷 지원기능을 손쉽게 추가할 수 있는 QGIS 는 수많은 래스터 및 벡터 데이터 포맷을 지원합니다.

QGIS 는 GNU 일반 공중 사용 허가 (GPL: General Public License) 에 따라 배포됩니다. 이 라이선스 하에서 QGIS 를 개발한다는 것은 사용자가 소스코드를 살펴보고 수정할 수 있고 GIS 프로그램에 언제나 접근할 수 있어서, 무료로 사용하고 마음대로 수정할 수 있게 함을 보장한다는 것을 의미합니다. 사용자가 QGIS 를 다운로드했을 때 이 라이선스의 전체 사본을 함께 받았을 겁니다. 이 문서의 부록부록 *A: GNU* 일반 공중 사용 허가서 에서도 이 라이선스를 찾아볼 수 있습니다.

팁: 최신 문서

언제나 QGIS 웹사이트의 문서 영역 https://www.qgis.org/ko/docs/ 에서 이 문서의 최신 버전을 찾아볼 수 있습니다.

CHAPTER 3

표기 방법

이 장에서는 이 설명서 전체에 사용된 일관된 스타일에 대해 설명합니다.

3.1 GUI 표기 방법

GUI 를 표기하는 스타일은 GUI 가 표현되는 모양과 비슷하도록 했습니다. 일반적으로 마우스를 가져가지 않았을 때의 모양을 반영했기 때문에, 사용자는 GUI 에서 이 설명서에서 설명하는 것과 비슷한 무언가를 찾을 수 있습니다.

- 메뉴 옵션: *Layer → Add a Raster Layer* 또는 *Settings → Toolbars → Digitizing*
- 도구:  Add a Raster Layer
- 버튼: *Save as Default*
- 대화창 제목: *Layer Properties*
- 탭: *General*
- 체크박스:  *Render*
- 라디오 버튼:  *Postgis SRID*  *EPSG ID*
- 숫자 선택: 
- 문자열 선택: 
- 파일 탐색: …
- 색상 선택: 
- 슬라이더: 
- 텍스트 입력: 

그림자는 클릭할 수 있는 GUI 라는 사실을 나타냅니다.

3.2 텍스트 또는 키보드 표기 방법

이 설명서는 클래스 또는 메소드와 같은 서로 다른 개체를 나타내기 위해 텍스트, 키보드 명령어 및 코딩과 관련된 스타일도 포함하고 있습니다. 이 스타일은 QGIS 에서 실제로 볼 수 있는 텍스트 또는 코딩의 모양과 일치하지는 않습니다.

- 하이퍼링크: https://qgis.org
- 키 조합: `Ctrl+B` 를 눌러보십시오. Ctrl 키를 누른 상태에서 B 를 누르라는 뜻입니다.
- 파일명: `lakes.shp`
- 클래스명: **NewLayer**
- 메소드: *classFactory*
- 서버: *myhost.de*
- 사용자 텍스트: `qgis --help`

코드는 다음 예시처럼 고정폭 글꼴로 표시합니다.

```
PROJCS["NAD_1927_Albers",
  GEOGCS["GCS_North_American_1927",
```

3.3 OS 별 지침

GUI 순서 및 짧은 텍스트는 문장 안의 서식으로 다음 예처럼 표현될 수 있습니다. *File* *QGIS* *Quit to close QGIS* 를 클릭하세요. 리눅스, 유닉스 및 윈도우 OS 에서는 먼저 파일 메뉴를 클릭한 다음 종료를 클릭해야 하고, 맥 OS 에선 먼저 QGIS 메뉴를 클릭한 다음 종료를 클릭해야 한다는 뜻입니다.

많은 양의 텍스트는 목록으로 표현될 수도 있습니다:

- 이렇게 하세요
- 저렇게 하세요
- 또는 저렇게 하세요

또는 문단으로 표현될 수도 있습니다:

이렇게 하시고 이렇게 하신 후 이렇게 하시고 이렇게 이렇게 하신 다음 이러이러하게 이렇게 하십시오.

저렇게 하시고 저렇게 하신 후 저렇게 하시고 저렇게 저렇게 하신 다음 저러저러하게 저렇게 하십시오.

사용자 지침서 전체에 걸쳐 나오는 스크린샷은 서로 다른 플랫폼에서 생성되었습니다.

CHAPTER 4

주요 기능

QGIS 는 핵심 기능과 플러그인의 형태로 제공되는 수많은 GIS 공통 기능들을 제공합니다. 위치 탐색란에서 함수, 데이터셋과 그 이상을 쉽게 검색할 수 있습니다.

주요 기능 및 플러그인의 일반 카테고리 6 개를 간단히 소개한 다음, 통합 파이썬 콘솔에 대해 처음으로 통찰합니다.

4.1 데이터 보기

서로 다른 포맷 및 투영체를 가진 (2D 또는 3D) 벡터와 래스터 데이터의 조합을 내부 또는 공통 포맷으로 변환하지 않은 채로 볼 수 있습니다. 다음과 같은 포맷들을 지원합니다:

- PostGIS, SpatiaLite 및 MS SQL Spatial, Oracle Spatial, 그리고 설치된 OGR 라이브러리가 지원하는 GeoPackage, ESRI Shapefile, MapInfo, SDTS, GML 을 포함하는 수많은 벡터 포맷들을 이용하는 공간 대응 (spatially-enabled) 테이블과 뷰들. 벡터 데이터 작업 을 참조하십시오.
- GeoTIFF, ERDAS IMG, ArcInfo ASCII GRID, JPEG, PNG 등 설치된 GDAL (Geospatial Data Abstraction Library) 라이브러리가 지원하는 래스터 및 이미지 포맷들. 래스터 데이터 작업 을 참조하십시오.
- 메시 데이터 (TIN 및 정규 그리드를 지원합니다). 메시 데이터 작업 을 참조하십시오.
- 벡터 타일
- GRASS 데이터베이스 (location/mapset) 가 지원하는 GRASS 래스터 및 벡터 데이터. *GRASS GIS* 통합 을 참조하십시오.
- WMS, WMTS, WCS, WFS 및 WFS-T 를 포함하는, 온라인 공간 데이터로 서비스되는 OGC 웹 서비스들. *OGC / ISO* 프로토콜 작업 을 참조하십시오.

  QGIS 인증 인프라를 통해 웹 서비스 및 기타 리소스들을 위한 사용자/비밀번호, 인증서 및 키를 관리할 수 있습니다.
- 스프레드시트 (ODS/XLSX)

시계열 데이터를 지원합니다.

4.2 데이터 탐색 및 지도 제작

사용자는 맵을 작성하고 친숙한 GUI 를 통해 공간 데이터를 양방향으로 탐색할 수 있습니다. GUI 로도 많은 유용한 도구들이 있습니다.

- QGIS 탐색기
- 실시간 (on-the-fly) 투영 변환
- DB 관리자
- 인쇄 레이아웃
- 보고서
- 오버뷰 패널
- 공간 북마크
- 주석 도구
- 객체 식별/선택
- 속성 편집/보기/검색
- 데이터 정의 피처 라벨 작업
- 데이터 정의 벡터 및 래스터 심볼 도구
- 경위선망 레이어가 지원되는 지도 조판
- 지도의 방위표, 축척 및 저작권 라벨
- 프로젝트 저장 및 불러오기 지원

4.3 데이터 생성, 편집, 관리 및 내보내기

벡터 및 래스터 레이어를 여러 가지 포맷으로 생성, 편집, 관리, 변환할 수 있습니다. QGIS 는 다음 기능들을 제공합니다:

- 벡터 디지타이즈 작업 도구
- 다수의 파일 포맷 및 GRASS 벡터 레이어를 생성하고 편집할 수 있는 능력
- 이미지에 위치 정보를 부여 (geocode) 하기 위한 지리참조자 (georeferencer) 플러그인
- GPX 포맷 가져오기/내보내기, 그리고 다른 GPS 포맷을 GPX 로 변환하거나 GPS 기기로 직접 다운로드/업로드할 수 있는 GPS 도구 (리눅스의 경우, GPS 기기 목록에 usb: 가 추가됐습니다)
- 오픈스트리트맵 데이터 가시화 및 편집 지원
- DB 관리자 플러그인을 사용해서 파일로부터 공간 데이터베이스 테이블을 생성할 수 있는 능력
- 공간 데이터베이스 테이블 처리 개선
- 벡터 속성 테이블 관리 도구
- 스크린샷을 지리좌표를 가진 이미지로 저장하는 옵션
- 스타일을 내보낼 수 있게 개선된 DXF 내보내기 도구 및 CAD 와 유사한 기능들을 수행하기 위한 플러그인

4.4 데이터 분석

공간 데이터베이스 및 다른 OGR 지원 포맷에 대해 공간 데이터 분석을 수행할 수 있습니다. 현재 QGIS 는 벡터 분석, 래스터 분석, 샘플링, 공간 처리 (geoprocessing), 도형 및 데이터베이스 관리 도구를 제공하고 있습니다. 통합된 GRASS 도구도 사용할 수 있는데, 이 도구는 400 개가 넘는 모듈로 이루어진 완전한 GRASS 기능을 포함합니다. (*GRASS GIS* 통합 을 참조하십시오.) 아니면, 공간 처리 플러그인을 사용할 수도 있습니다. 이 플러그인은 강력한 지리공간 분석 프레임워크를 제공해서 QGIS 에서 내장 알고리즘뿐 아니라 GDAL, SAGA, GRASS, R 등과 같은 외부 알고리즘도 호출할 수 있습니다. (소개 를 참조하십시오.) 분석 함수는 모두 배경에서 실행되기 때문에, 공간 처리가 완료되기 전에도 사용자 작업을 계속할 수 있습니다.

그래픽 모델 작성자는 직관적인 그래픽 환경에서 함수들을 완전한 워크플로로 결합/연결시킬 수 있습니다.

4.5 온라인 맵 발행

QGIS 를 WMS, WMTS, WMS-C 또는 WFS 와 WFS-T 클라이언트로 사용할 수 있으며 (*OGC / ISO* 프로토콜 작업 참조), QGIS 서버의 WMS, WCS 및 WFS 프로토콜을 통해 웹서버를 사용해서 인터넷에 사용자 데이터를 공개할 수도 있습니다 (QGIS-Server-manual 참조).

4.6 플러그인을 통한 확장 QGIS 기능

QGIS 는 사용자의 특수한 목적에 맞게 변화될 수 있습니다. 확정성이 있는 플러그인 아키텍처와 플러그인을 만드는데 사용할 수 있는 라이브러리들이 이를 가능하게 합니다. 심지어 C++ 또는 파이썬을 이용해 새로운 응용 프로그램을 만드는 것도 가능합니다!

4.6.1 핵심 플러그인

핵심 플러그인은 다음과 같습니다.

1. DB 관리자 (데이터베이스에서 레이어와 테이블을 교환, 편집, 살펴보기 및 SQL 쿼리 실행)
2. 도형 검사기 (도형 오류를 확인)
3. GDAL 지리참조자 (GDAL 을 이용해 래스터에 투영 정보를 추가)
4. GPS 도구 (GPS 데이터 불러오기 및 가져오기)
5. GRASS 7 (GRASS GIS 통합)
6. 메타 검색 카탈로그 클라이언트 (웹 (CSW) 표준을 따르기 위해 OGC 카탈로그 서비스를 지원하는 메타데이터 카탈로그 서비스와 상호작용)
7. 오프라인 편집 작업 (데이터베이스 오프라인 편집 및 동기화 가능)
8. 공간 처리 (QGIS 공간 데이터 처리 프레임워크)
9. 위상 검사기 (벡터 레이어에서 위상기하학적 오류를 확인)

4.6.2 외부 파이썬 플러그인

QGIS 는 커뮤니티가 개발한 외부 파이썬 플러그인을 점점 더 많이 제공하고 있습니다. 이 플러그인들은 공식 플러그인 저장소에 있으며 파이썬 플러그인 설치기를 통해 쉽게 설치할 수 있습니다. 플러그인 대화창 을 참조하십시오.

4.7 파이썬 콘솔

스크립트 작업 시 통합된 파이썬 콘솔을 이용하면 편리합니다. *Plugins* → *Python Console* 메뉴에서 파이썬 콘솔을 실행할 수 있습니다. 콘솔은 모달리스 (non-modal) 유틸리티 창으로 열립니다. QGIS 환경과 상호작용하기 위해, `QgisInterface` 클래스의 인스턴스인 `qgis.utils.iface` 변수를 제공합니다. 이 인터페이스를 통해 맵 캔버스, 메뉴, 툴바 및 QGIS 응용 프로그램의 다른 부분들에 접근할 수 있습니다. 사용자가 스크립트를 작성한 다음 QGIS 창에 스크립트를 드래그 & 드롭하면 자동으로 실행될 것입니다.

파이썬 콘솔을 통한 작업 및 QGIS 플러그인과 응용 프로그램 개발에 대해 더 많은 정보를 알고 싶다면, *QGIS 파이썬 콘솔* 과 PyQGIS-Developer-Cookbook 을 참조하시기 바랍니다.

4.8 알려진 문제점들

4.8.1 열린 파일 개수 제한

대용량 QGIS 프로젝트를 열었을 때 모든 레이어에 아무 문제가 없음에도 불구하고 일부 레이어가 불량으로 표시된다면, 이 문제점을 맞닥뜨린 것일 수도 있습니다. 리눅스의 경우 (다른 OS 들도 마찬가지입니다만) 프로세스가 열 수 있는 파일 개수를 제한합니다. 이 리소스 제한은 프로세스 별로 걸리며 상속됩니다. 셸 (shell) 에 내장된 `ulimit` 명령어는 오직 현재 셸 프로세스의 제한만을 변경할 수 있으며, 모든 하위 프로세스가 새 제한값을 상속할 것입니다.

다음과 같이 입력하면 현재의 모든 `ulimit` 정보를 볼 수 있습니다:

```
$ ulimit -aS
```

콘솔에 다음 명령어를 입력하면 프로세스 당 현재 허용된 열린 파일 개수를 볼 수 있습니다:

```
$ ulimit -Sn
```

기존 세션 의 제한을 변경하려면, 다음과 비슷한 명령어를 입력해야 할 수도 있습니다:

```
$ ulimit -Sn #number_of_allowed_open_files
$ ulimit -Sn
$ qgis
```

영구적으로 수정하려면:

대부분의 리눅스 시스템에서, 로그인 시 `/etc/security/limits.conf` 또는 `/etc/security/limits.d/*.conf` 파일에 담겨 있는 설정에 따라 `pam_limits` 모듈이 리소스 제한을 설정합니다. 루트 권한을 가지고 있다면 (`sudo` 를 통해서도 가능합니다) 이 파일들을 편집할 수 있습니다. 그러나 변경한 내용을 적용하려면 로그아웃 후 다시 로그인해야 합니다.

추가 정보:

https://www.cyberciti.biz/faq/linux-increase-the-maximum-number-of-open-files/ https://linuxaria.com/article/open-files-in-linux

CHAPTER 5

시작하기

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

5.1 QGIS 설치

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

5.1.1 바이너리로부터 설치

MS 윈도우 와 맥 OS 의 경우 표준 인스톨러를 사용할 수 있습니다. 수많은 GNU/Linux 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://download.qgis.org 를 방문하십시오.

5.1.2 소스로부터 설치

소스로부터 QGIS 를 빌드해야 한다면, 설치 지침을 참조해주십시오. QGIS 소스코드와 함께 `INSTALL` 이라는 파일로 배포하고 있습니다. 온라인 상의 https://github.com/qgis/QGIS/blob/master/INSTALL.md 에서도 설치 지침을 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.)

5.1.3 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.

5.1.4 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 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/sample-data/ 에서 (사우스다코타 주의 스피어피시 또는 노스캐롤라이나 주 같은) 샘플 위치들을 모아 놓은 것을 찾아보십시오.

5.2 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 다음과 같은 방법을 사용하십시오:

- *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

5.3 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. QGIS 에 파일들을 불러오려면:
    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 더블클릭해서 여십시오.
    5. *Coordinate Reference System Selector* 대화창이 열립니다. *Filter* 메뉴에서 `2964` 를 입력해서 아래쪽에 있는 좌표계 (Coordinate Reference System) 를 필터링하십시오.
    6. *NAD27 / Alaska Albers* 항목을 선택합니다.
    7. *OK* 를 클릭합니다.
    8. 데이터소스 관리자 창을 닫으십시오.

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

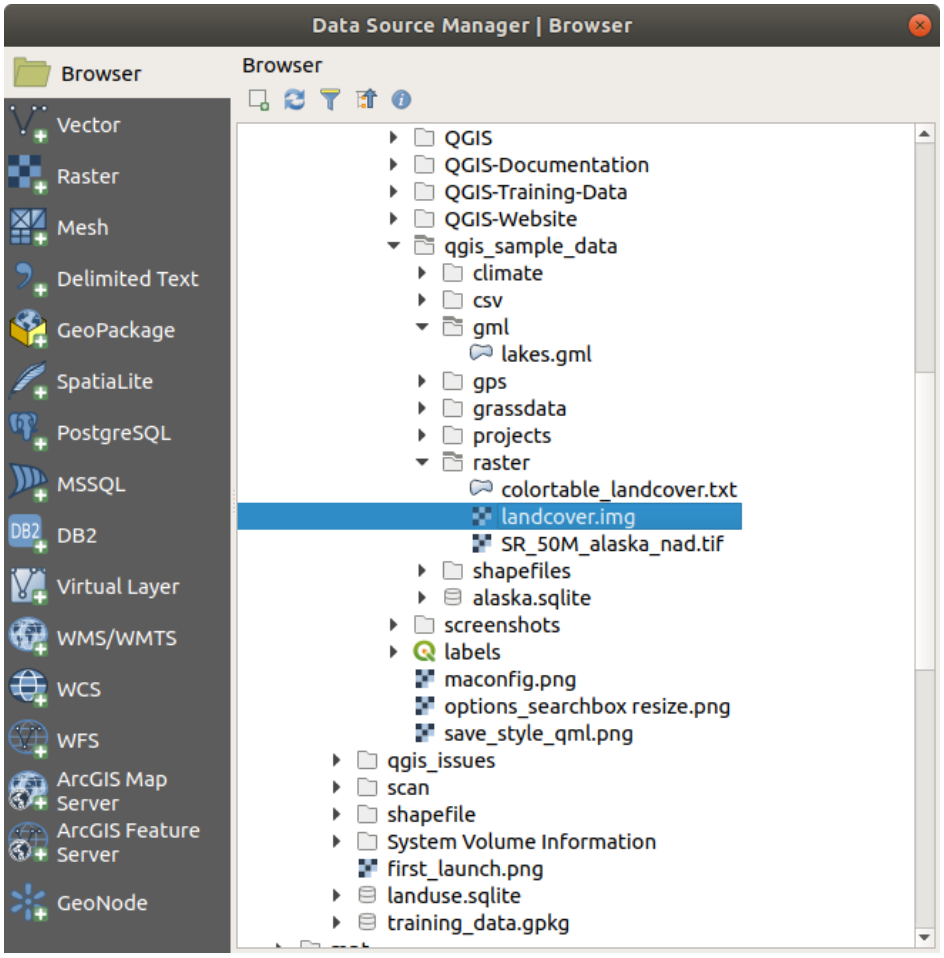


그림 5.1: QGIS 의 새 프로젝트에 데이터 추가하기

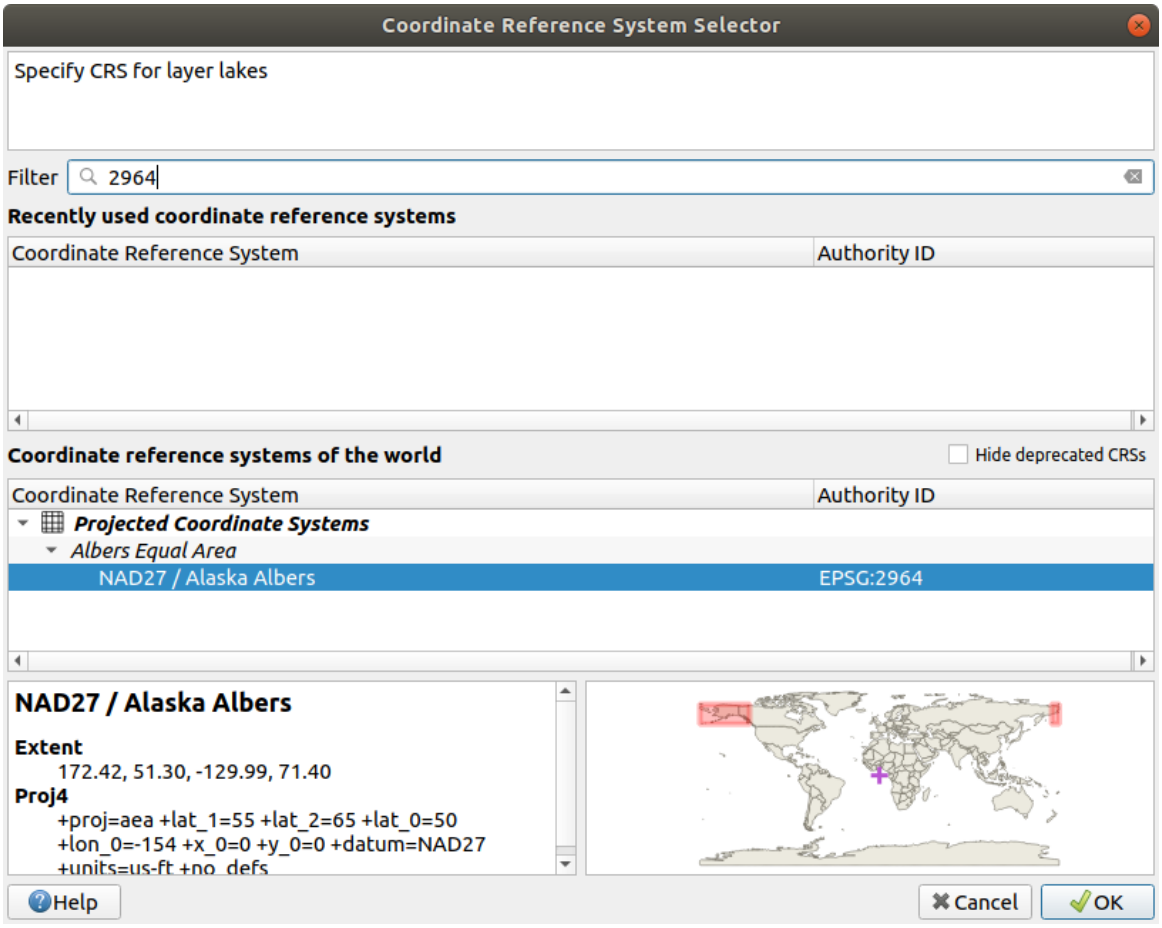


그림 5.2: 해당 데이터의 좌표계를 선택

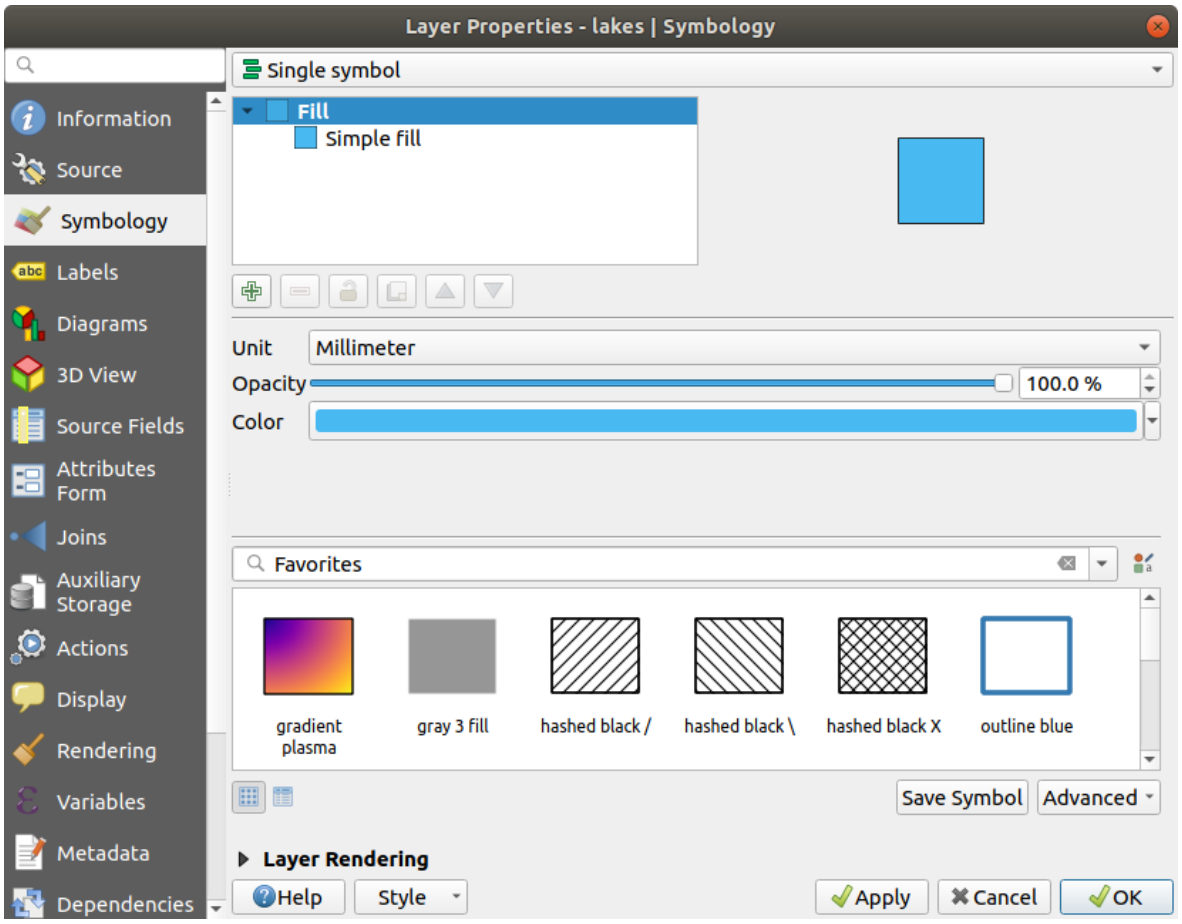


그림 5.3: 호수 색상을 선택하기

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

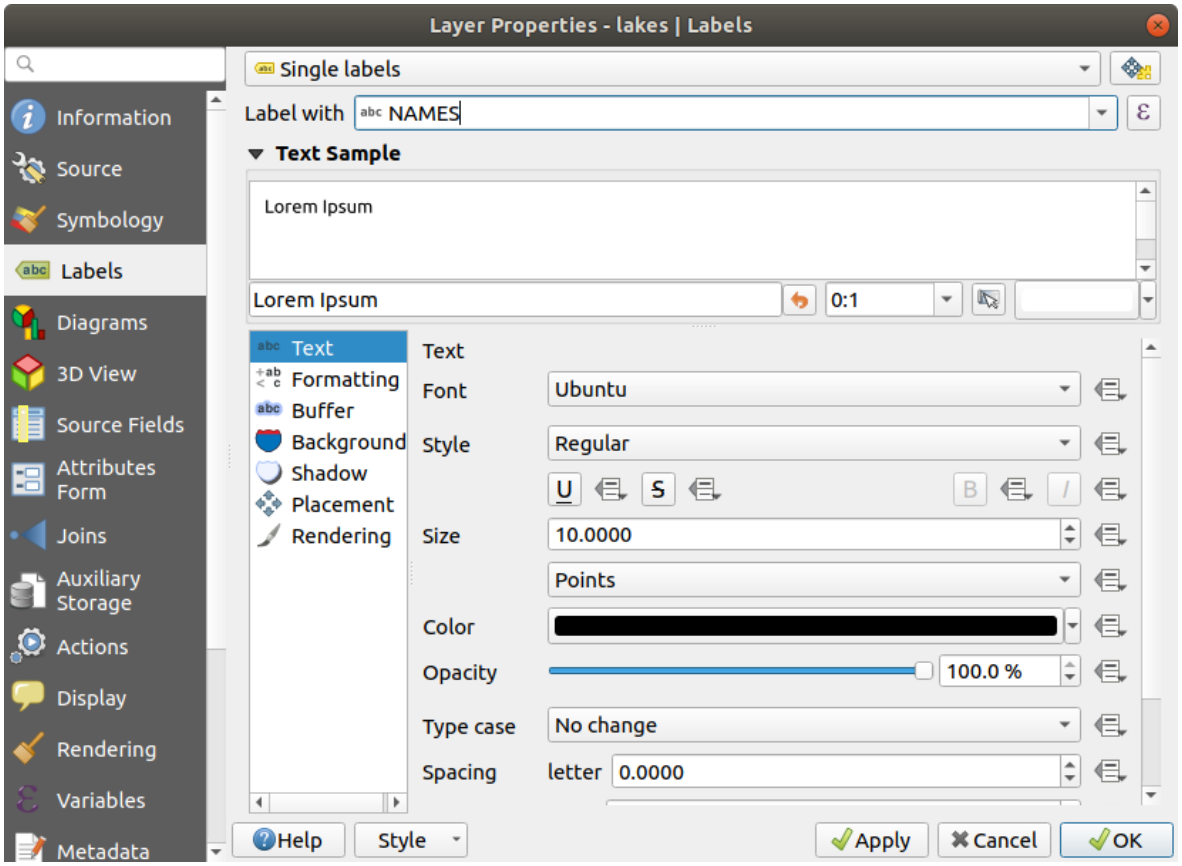


그림 5.4: 호수의 이름을 표시

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 클릭합니다.
   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.
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.

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 6

프로젝트 파일 작업

6.1 QGIS 프로젝트 소개

프로젝트란 사용자의 QGIS 세션 상태를 말합니다. QGIS 는 한 번에 프로젝트 하나에서만 작업 가능합니다. 설정값은 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 체크해서 이 옵션을 활성화 할 수 있습니다.

기존 프로젝트를 QGIS 로 불러오려면 탐색기 패널에서 프로젝트 파일을 선택하거나, *Project* →  *Open*…, *Project* → *New from template* or *Project* → *Open Recent* → 메뉴를 이용하면 됩니다.

QGIS 를 실행하면 스크린샷, 명칭, 파일 경로를 포함하는 *Project Templates* 와 *Recent Projects* 목록을 (최대 10 개까지) 표시합니다. *Recent Projects* 목록을 통해 최근 작업했던 프로젝트를 간편하게 불러올 수 있습니다. 프로젝트 또는 프로젝트 템플릿을 열려면 해당 항목을 더블클릭하십시오. 자동으로 새 프로젝트를 생성하려면 레이어를 추가하면 됩니다. 그러면 목록이 사라지며 맵 캔버스가 나타날 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.

프로젝트 파일이 저장하는 정보는 다음을 포함합니다:

- 추가된 레이어
- 레이어 별 쿼리 가능 여부

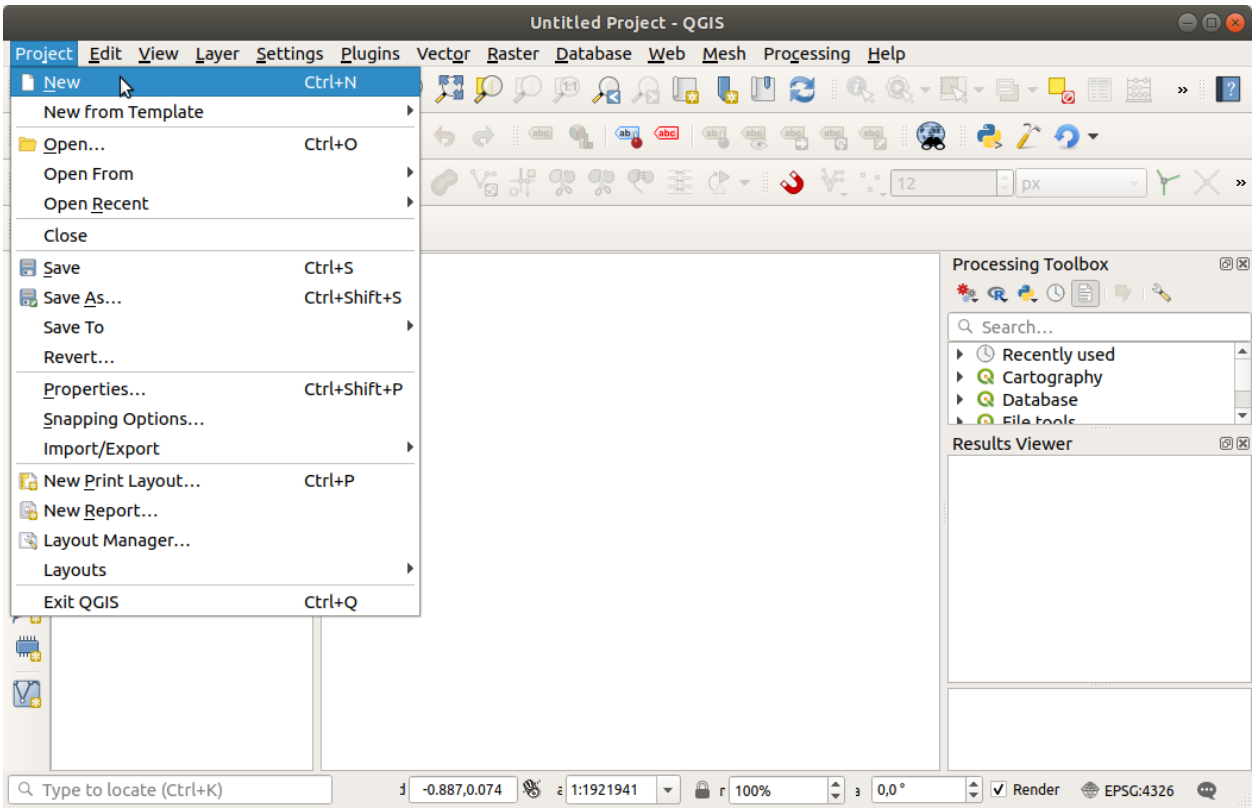


그림 6.1: QGIS 에서 새 프로젝트 시작

- 심볼 및 스타일을 포함한 레이어 속성
- 맵 뷰의 투영법
- 마지막 뷰의 영역
- 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 업데이트됐습니다. QGIS 과거 버전에서 저장된 프로젝트 파일은 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 `.qgs~` 확장자를 가진 프로젝트 파일의 백업을 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 데이터베이스에 프로젝트를 저장하거나 불러올 수 있습니다:

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

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

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

6.2 엉터리 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 수도 있습니다. 이 경우 탐색하는 데 시간이 걸릴 수도 있습니다.
- *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 

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.

6.3 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. 이미지를 지리참조시킬 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 7

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.)

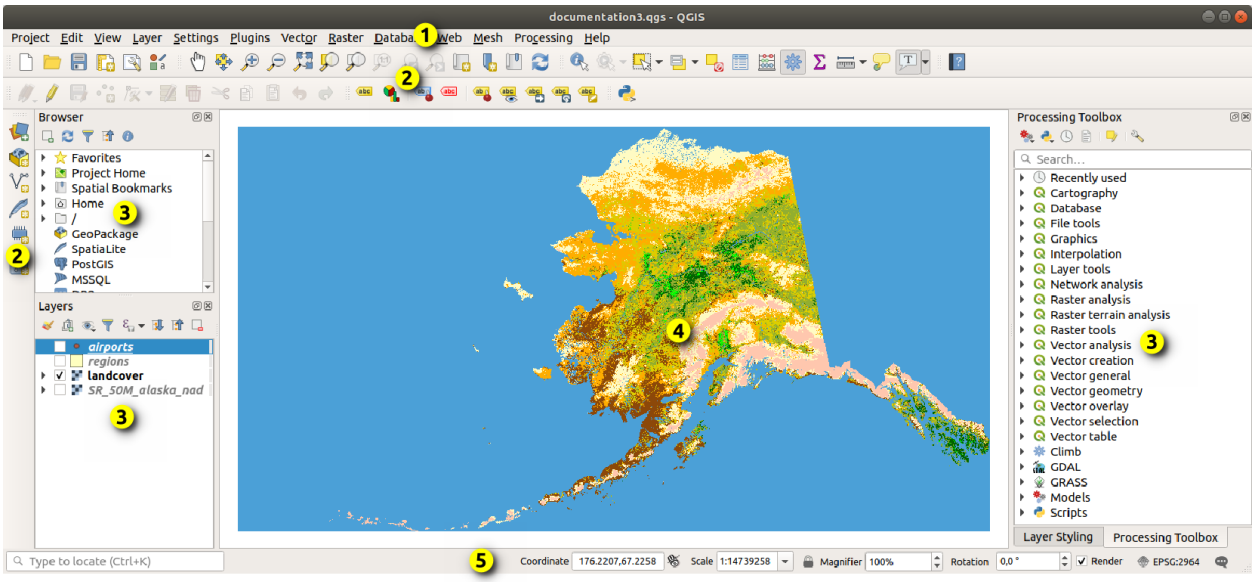


그림 7.1: 알래스카 예시 데이터를 불러온 QGIS GUI

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

주요 QGIS GUI(그림 7.1) 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.

7.1 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.

7.1.1 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 또는 PostgreSQL 데이터베이스로부터 프로젝트를 불러옵니다.
- *Close* : 프로젝트를 닫거나 가장 최근 저장한 상태로 되돌립니다.
- *Save* : 프로젝트를 `.qgs` 또는 `.qgz` 파일 포맷으로 저장하거나, GeoPackage 나 PostgreSQL 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.

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
PostgreSQL…			*QGIS* 프로젝트 소개
Open Recent	Alt+J + R		*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			
Templates…			*QGIS* 프로젝트 소개
GeoPackage…			*QGIS* 프로젝트 소개
PostgreSQL…			*QGIS* 프로젝트 소개
Revert…			
Properties…	Ctrl+Shift+P		프로젝트 속성
Snapping Options…			스냅 허용 오차 및 검색 반경 설정
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
Exit QGIS	Ctrl+Q		

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

7.1.2 편집

Edit 메뉴는 레이어 속성이나 도형을 편집하는 데 필요한 내장 도구 대부분을 제공합니다. (자세한 내용은편집 작업을 참조하세요.)

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 7.1 – 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
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 Circular String		*Shape Digitizing*	원호 스트링 추가
Add Circular String by Radius		*Shape Digitizing*	원호 스트링 추가
Add Circle		*Shape Digitizing*	원 그리기
Add Circle from 2 Points		*Shape Digitizing*	원 그리기
Add Circle from 3 Points		*Shape Digitizing*	원 그리기
Add Circle from 3 Tangents		*Shape Digitizing*	원 그리기
Add Circle from 2 Tangents and a Point		*Shape Digitizing*	원 그리기

다음 페이지에 계속

표 7.1 – 이전 페이지에서 계속

메뉴 옵션	단축키	툴바	참조
Add Circle by a Center Point and Another Point		*Shape Digitizing*	원 그리기
Add Rectangle		*Shape Digitizing*	직사각형 그리기
Add Rectangle from Extent		*Shape Digitizing*	직사각형 그리기
Add Rectangle from Center and a Point		*Shape Digitizing*	직사각형 그리기
Add Rectangle from 3 Points (Distance from 2nd and 3rd point)		*Shape Digitizing*	직사각형 그리기
Add Rectangle from 3 Points (Distance from projected point on segment p1 and p2)		*Shape Digitizing*	직사각형 그리기
Add Regular Polygon		*Shape Digitizing*	정다각형 그리기
Add Regular Polygon from Center and a Point		*Shape Digitizing*	정다각형 그리기
Add Regular Polygon from Center and a Corner		*Shape Digitizing*	정다각형 그리기
Add Regular Polygon from 2 Points		*Shape Digitizing*	정다각형 그리기
Add Ellipse		*Shape Digitizing*	타원 그리기
Add Ellipse from Center and 2 Points		*Shape Digitizing*	타원 그리기
Add Ellipse from Center and a Point		*Shape Digitizing*	타원 그리기
Add Ellipse from Extent		*Shape Digitizing*	타원 그리기
Add Ellipse from Foci		*Shape Digitizing*	타원 그리기
Add Annotation			주석 도구
Text Annotation		*Attributes*	주석 도구
Form Annotation		*Attributes*	주석 도구
HTML Annotation		*Attributes*	주석 도구
SVG Annotation		*Attributes*	주석 도구
Move Feature(s)		*Advanced Digitizing*	피처 (들) 이동
Copy and Move Feature(s)		*Advanced Digitizing*	피처 (들) 이동
Delete Selected		*Digitizing*	선택한 피처 삭제하기

다음 페이지에 계속

표 7.1 – 이전 페이지에서 계속

메뉴 옵션	단축키	툴바	참조
Modify Attributes of Selected Features		*Digitizing*	속성값 편집
Rotate Feature(s)		*Advanced Digitizing*	피처 (들) 회전
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
Merge Attributes of Selected Features		*Advanced Digitizing*	선택한 피처의 속성 병합
Vertex Tool (All Layers)		*Digitizing*	꼭짓점 도구
Vertex Tool (Current Layer)		*Digitizing*	꼭짓점 도구
Rotate Point Symbols		*Advanced Digitizing*	포인트 심볼 회전
Offset Point Symbols		*Advanced Digitizing*	포인트 심볼 오프셋
Reverse Line		*Advanced Digitizing*	라인 반전
Trim/extend Feature		*Advanced Digitizing*	피처 다듬기/연장하기

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

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

7.1.3 뷰

맵은 맵 뷰로 렌더링됩니다. *View* 도구를 이용해서 이런 뷰들과 상호작용할 수 있습니다. (더 자세한 정보는맵 캔버스 작업 을 참조하세요.) 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*: shows or hides 활성화된패널 을 켜고 끕니다 — 피처 디지타이즈 작업 시 (캔버스 최대 가시화를 위해) 유용합니다. 또 QGIS 의 주 캔버스를 사용한 프리젠테이션 (프로젝터 용/녹화 용) 용도로도 유용합니다.
- *Toggle Map Only*: 패널, 툴바, 메뉴, 상태 바를 모두 끄고 맵 캔버스만 보이게 합니다. 전체 화면 옵션과 함께 사용하면, 사용자의 화면에 맵만 표시됩니다.

메뉴 옵션	단축키	툴바	참조
New Map View	`Ctrl+M`		맵 뷰
New 3D Map View	`Ctrl+Alt+M`		*3D* 맵 뷰
Pan Map		*Map Navigation*	확대/축소 및 이동
Pan Map to Selection		*Map Navigation*	
Zoom In	`Ctrl+Alt++`	*Map Navigation*	확대/축소 및 이동
Zoom Out	`Ctrl+Alt+-`	*Map Navigation*	확대/축소 및 이동
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 Angle		*Attributes*	측정
Statistical Summary		*Attributes*	통계 요약 패널

다음 페이지에 계속

표 7.2 – 이전 페이지에서 계속

메뉴 옵션	단축키	툴바	참조
Zoom Full	Ctrl+Shift+F	*Map Navigation*	확대/축소 및 이동
Zoom To Selection	Ctrl+J	*Map Navigation*	확대/축소 및 이동
Zoom To Layer		*Map Navigation*	확대/축소 및 이동
Zoom To Native Resolution (100%)		*Map Navigation*	확대/축소 및 이동
Zoom Last		*Map Navigation*	확대/축소 및 이동
Zoom Next		*Map Navigation*	확대/축소 및 이동
Decorations	Alt+V + D		지도 장식
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 Photocopy (Grayscale)			
Simulate Fax (Mono)			
Simulate Color Blindness (Protanope)			
Simulate Color Blindness (Deuteronope)			
Show Map Tips		*Attributes*	표시 속성
New Spatial Bookmark…	Ctrl+B	*Map Navigation*	공간 북마크
Show Spatial Bookmarks	Ctrl+Shift+B	*Map Navigation*	공간 북마크
Show Spatial Bookmark Manager			공간 북마크
Refresh	F5	*Map Navigation*	
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

다음 페이지에 계속

표 7.2 – 이전 페이지에서 계속

메뉴 옵션	단축키	툴바	참조
Toggle Selected Layers			레이어 패널
Toogle Selected Layers Independently			레이어 패널
Hide Deselected Layers			레이어 패널
Panels			패널 및 툴바
Advanced Digitizing			고급 디지타이즈 작업 패널
Browser			탐색기 패널
Browser (2)			탐색기 패널
GPS Information			실시간 *GPS* 추적
GRASS Tools			*GRASS GIS* 통합
Layer Order			레이어 순서 패널
Layer Styling			레이어 스타일 작업 패널
Layers			레이어 패널
Log Messages			로그 메시지 패널
Overview			오버뷰 패널
Processing Toolbox			툴박스
Results Viewer			툴박스
Snapping and Digitizing Options			스냅 허용 오차 및 검색 반경 설정
Spatial Bookmark Manager			공간 북마크
Statistics			통계 요약 패널
Tile Scale			타일셋
Undo/Redo			실행 취소/재실행 패널
Toolbars			패널 및 툴바
Advanced Digitizing Toolbar			고급 디지타이즈 작업
Attributes Toolbar			
Data Source Manager Toolbar			데이터소스 관리
Database Toolbar			
Digitizing Toolbar			기존 레이어 디지타이징
Help Toolbar			
Label Toolbar			라벨 툴바
Manage Layers Toolbar			데이터소스 관리
Map Navigation Toolbar			
Plugins Toolbar			플러그인
Project Toolbar			
Raster Toolbar			
Selection Toolbar			피처 선택
Shape Digitizing Toolbar			형태 디지타이즈 작업
Snapping Toolbar			스냅 허용 오차 및 검색 반경 설정
Vector Toolbar			
Web Toolbar			
GRASS			*GRASS GIS* 통합
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.

7.1.4 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. (프로젝트 내포 작업 참조)

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

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 *shapefile* 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 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 PostGIS Layer…	`Ctrl+Shift+D`	*Manage Layers*	데이터베이스 관련 도구
Add SpatiaLite Layer…	`Ctrl+Shift+L`	*Manage Layers*	*SpatiaLite* 레이어
Add MSSQL Spatial Layer…		*Manage Layers*	데이터베이스 관련 도구
Add Oracle Spatial Layer…		*Manage Layers*	데이터베이스 관련 도구
Add DB2 Spatial Layer…	`Ctrl+Shift+2`	*Manage Layers*	데이터베이스 관련 도구
Add/Edit Virtual Layer…		*Manage Layers*	가상 레이어 생성

다음 페이지에 계속

표 7.3 – 이전 페이지에서 계속

메뉴 옵션	단축키	툴바	참조
Add WMS/WMTS Layer…	Ctrl+Shift+W	*Manage Layers*	*WMS/WMTS* 레이어 불러오기
Add XYZ Layer…			*XYZ* 타일 서비스 사용하기
Add ArcGIS Map Service Layer…		*Manage Layers*	
Add WCS Layer…		*Manage Layers*	*WCS* 클라이언트
Add WFS Layer…		*Manage Layers*	*WFS* 및 *WFS-T* 클라이언트
Add ArcGIS Feature Service Layer …		*Manage Layers*	
Add Vector Tile Layer…			
Embed Layers and Groups…			프로젝트 내포 작업
Add from Layer Definition File…			레이어 정의 파일
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
Toggle Editing		*Digitizing*	기존 레이어 디지타이징
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)			
Set CRS of Layer(s)	Ctrl+Shift+C		레이어 좌표계
Set Project CRS from Layer			프로젝트 좌표계
Layer Properties…			벡터 속성 대화창, 래스터 속성 대화창, 메시 데이터셋 속성
Filter…	Ctrl+F		쿼리 작성기
Labeling			라벨 속성
Show in Overview			오버뷰 패널

다음 페이지에 계속

표 7.3 – 이전 페이지에서 계속

메뉴 옵션	단축키	툴바	참조
Show All in Overview			오버뷰 패널
Hide All from Overview			오버뷰 패널

7.1.5 설정

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
Options…	옵션

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

7.1.6 플러그인

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.

7.1.7 벡터

다음은 모든 핵심 플러그인을 활성화했을 때 볼 수 있는 *Vector* 메뉴의 모습입니다.

메뉴 옵션	단축키	툴바	참조
Check Geometries…			도형 점검기 플러그인
GPS Tools	`Alt+O + G`	*Vector*	*GPS* 플러그인
Topology Checker		*Vector*	위상 점검기 플러그인
Geoprocessing Tools	`Alt+O + G`		
Buffer…			버퍼
Clip…			잘라내기 *(Clip)*

다음 페이지에 계속

표 7.4 – 이전 페이지에서 계속

메뉴 옵션	단축키	툴바	참조
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…			유효성 점검하기
Delaunay Triangulation…			들로네 삼각분할
Densify by Count…			개수로 치밀화하기
Add Geometry Attributes…			도형 속성 추가하기
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		
Select by Location…			위치로 선택하기
Extract Layer Extent…			레이어 범위 추출하기
Random Points in Extent…			범위 안에 랜덤 포인트 생성하기
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

QGIS 는 기본적으로 *Vector* 메뉴에 하위 그룹으로 그룹화한공간 처리 알고리즘들을 추가합니다. 이렇게 하면 서로 다른 제공자들의 수많은 일반적인 벡터 기반 GIS 작업을 단축키로 사용할 수 있기 때문입니다. 해당 하위 메뉴를 전부 사용할 수 없을 경우, *Plugins* *Manage and Install Plugins…*메뉴에서 공간 처리 플러그인을 활성화하십시오.

Vector 메뉴 도구 목록은 공간 처리 알고리즘들 또는 몇몇 외부플러그인 들을 추가하면 확장될 수 있다는 사실을

기억하십시오.

7.1.8 래스터

다음은 모든 핵심 플러그인을 활성화했을 때 볼 수 있는 *Raster* 메뉴의 모습입니다.

메뉴 옵션	단축키	툴바	참조
 *Raster calculator…*			래스터 계산기
Align Raster…			래스터 정렬
 *Georeferencer*	`Alt+R + G`	*Raster*	지리참조자
Analysis			
Aspect…			경사 방향
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*)*
Hillshade…			음영기복
Proximity (Raster Distance)…			근접성 *(*래스터 거리*)*
Roughness…			거칠기
Sieve…			거르기
Slope…			경사도
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
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*)*

QGIS 는 기본적으로 *Raster* 메뉴에 하위 그룹으로 그룹화한공간 처리 알고리즘들을 추가합니다. 이렇게 하면 서로 다른 제공자들의 수많은 일반적인 래스터 기반 GIS 작업을 단축키로 사용할 수 있기 때문입니다. 해당 하위 메뉴를 전부 사용할 수 없을 경우, *Plugins* *Manage and Install Plugins…*메뉴에서 공간 처리 플러그인을 활성화하십시오.

Raster 메뉴 도구 목록은 공간 처리 알고리즘들 또는 몇몇 외부플러그인 들을 추가하면 확장될 수 있다는 사실을 기억하십시오.

7.1.9 데이터베이스

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

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
DB Manager⋯		*Database*	데이터베이스 관리자 플러그인

처음으로 QGIS 를 실행했을 경우 핵심 플러그인을 모두 불러오지는 않습니다.

7.1.10 웹

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

메뉴 옵션	단축키	툴바	참조
MetaSearch	Alt+W + M		메타검색 카탈로그 클라이언트
Metasearch		*Web*	메타검색 카탈로그 클라이언트
Help			메타검색 카탈로그 클라이언트

처음으로 QGIS 를 실행했을 경우 핵심 플러그인을 모두 불러오지는 않습니다.

7.1.11 메시 (mesh)

Mesh 메뉴는메시 레이어 를 취급하는 데 필요한 도구를 제공합니다.

메뉴 옵션	단축키	툴바	참조
Mesh Calculator⋯			

7.1.12 공간 처리

메뉴 옵션	단축키	툴바	참조
Toolbox	Ctrl+Alt+T		툴박스
Graphical Modeler…	Ctrl+Alt+G		그래픽 모델 생성기
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.

7.1.13 도움말

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			

7.1.14 QGIS

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

메뉴 옵션	단축키
Preferences	
About QGIS	
Hide QGIS	
Show All	
Hide Others	
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.

7.2 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.

7.2.1 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.

사용자의 필요에 따라 모든 툴바를 이동할 수 있습니다. 또는, 오른쪽 클릭하면 나오는 컨텍스트 메뉴를 통해, 아니면 툴바 위에 마우스를 가져 가서 툴바를 꺼버릴 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.

7.2.2 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:

- 고급 디지타이즈 작업 패널
- 탐색기 패널
- *GPS* 정보 패널
- 식별 패널
- 레이어 정렬 패널
- 레이어 스타일 작업 패널
- 레이어 패널
- 로그 메시지 패널
- 오버뷰 패널
- 공간처리 툴박스
- *Result Viewer Panel*
- 공간 북마크 관리자 패널
- 통계 패널
- 타일 축척 패널
- 실행 취소/재실행 패널

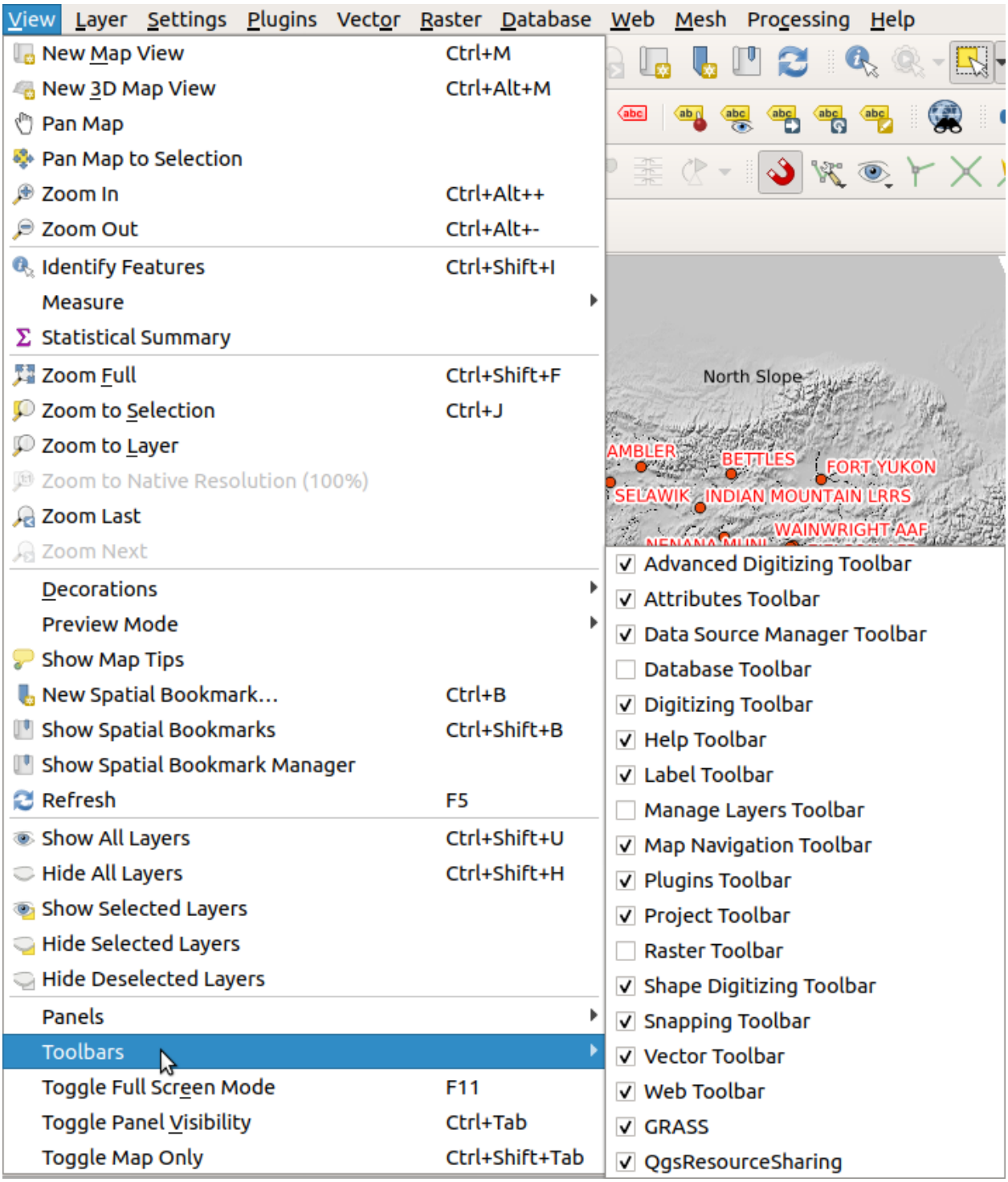

View
Layer
Settings
Plugins
Vector
Raster
Database
Web
Mesh
Processing
Help
New Map View Ctrl+M
New 3D Map View Ctrl+Alt+M
Pan Map
Pan Map to Selection
Zoom In Ctrl+Alt++
Zoom Out Ctrl+Alt+-
Identify Features Ctrl+Shift+I
Measure
Statistical Summary
Zoom Full Ctrl+Shift+F
Zoom to Selection Ctrl+J
Zoom to Layer
Zoom to Native Resolution (100%)
Zoom Last
Zoom Next
Decorations
Preview Mode
Show Map Tips
New Spatial Bookmark... Ctrl+B
Show Spatial Bookmarks Ctrl+Shift+B
Show Spatial Bookmark Manager
Refresh F5
Show All Layers Ctrl+Shift+U
Hide All Layers Ctrl+Shift+H
Show Selected Layers
Hide Selected Layers
Hide Deselected Layers
Panels
Toolbars
Toggle Full Screen Mode F11
Toggle Panel Visibility Ctrl+Tab
Toggle Map Only Ctrl+Shift+Tab
North Slope
AMBLER
BETTLES
FORT YUKON
SELAWIK
INDIAN MOUNTAIN LRRS
WAINWRIGHT AAF
Advanced Digitizing Toolbar
Attributes Toolbar
Data Source Manager Toolbar
Database Toolbar
Digitizing Toolbar
Help Toolbar
Label Toolbar
Manage Layers Toolbar
Map Navigation Toolbar
Plugins Toolbar
Project Toolbar
Raster Toolbar
Shape Digitizing Toolbar
Snapping Toolbar
Vector Toolbar
Web Toolbar
GRASS
QgsResourceSharing

그림 7.2: 툴바 메뉴

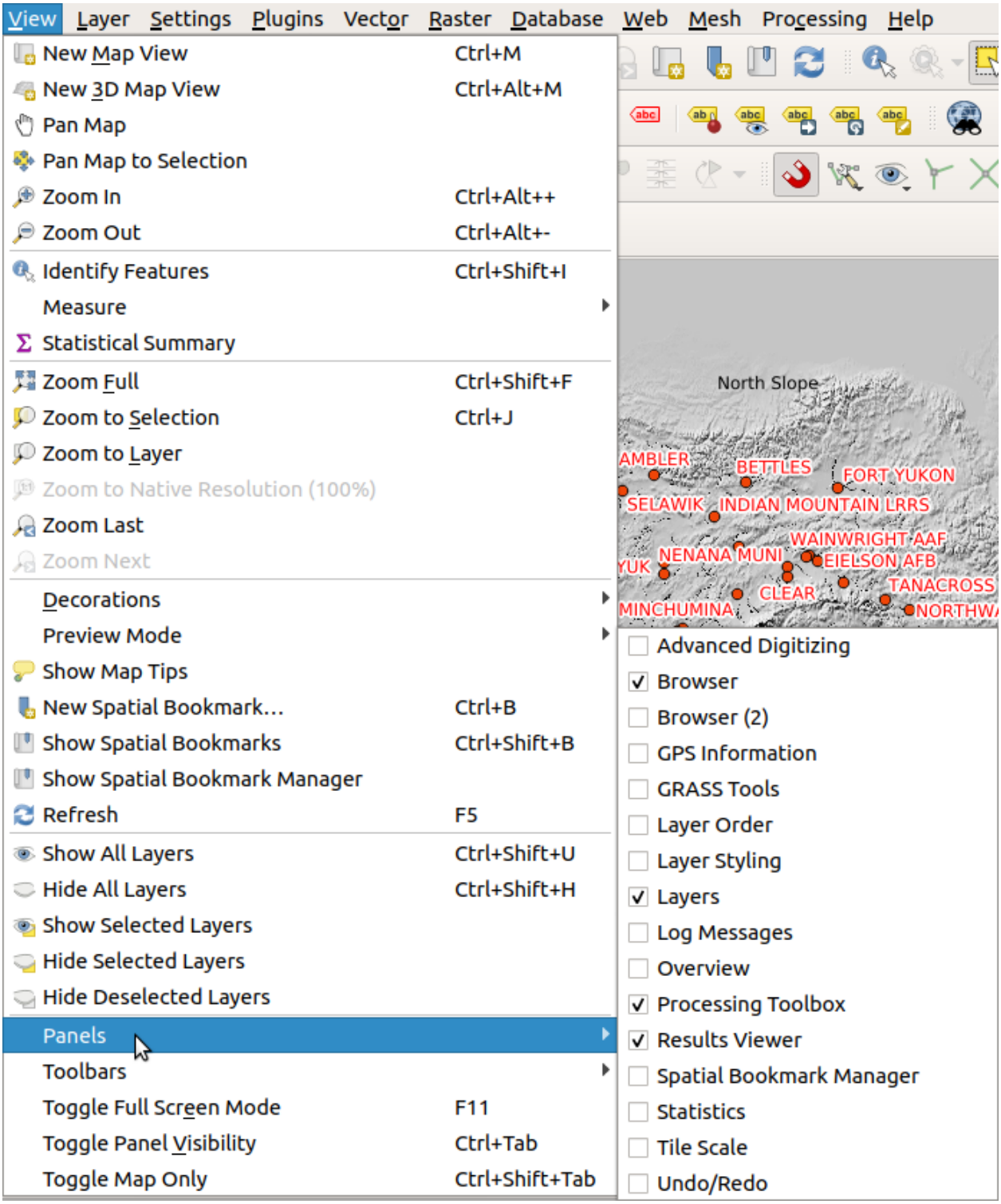

View
Layer
Settings
Plugins
Vector
Raster
Database
Web
Mesh
Processing
Help
New Map View Ctrl+M
New 3D Map View Ctrl+Alt+M
Pan Map
Pan Map to Selection
Zoom In Ctrl+Alt++
Zoom Out Ctrl+Alt+-
Identify Features Ctrl+Shift+I
Measure
Statistical Summary
Zoom Full Ctrl+Shift+F
Zoom to Selection Ctrl+J
Zoom to Layer
Zoom to Native Resolution (100%)
Zoom Last
Zoom Next
Decorations
Preview Mode
Show Map Tips
New Spatial Bookmark... Ctrl+B
Show Spatial Bookmarks Ctrl+Shift+B
Show Spatial Bookmark Manager
Refresh F5
Show All Layers Ctrl+Shift+U
Hide All Layers Ctrl+Shift+H
Show Selected Layers
Hide Selected Layers
Hide Deselected Layers
Panels
Toolbars
Toggle Full Screen Mode F11
Toggle Panel Visibility Ctrl+Tab
Toggle Map Only Ctrl+Shift+Tab
North Slope
AMBLER
BETTLES
FORT YUKON
SELAWIK
INDIAN MOUNTAIN LRRS
WAINWRIGHT AAF
NENANA MUNI
EIELSON AFB
TANACROSS
CLEAR
MINCHUMINA
Advanced Digitizing
Browser
Browser (2)
GPS Information
GRASS Tools
Layer Order
Layer Styling
Layers
Log Messages
Overview
Processing Toolbox
Results Viewer
Spatial Bookmark Manager
Statistics
Tile Scale
Undo/Redo

그림 7.3: 패널 메뉴

7.3 맵 뷰

7.3.1 맵 뷰 탐구하기

맵 캔버스 라고도 불리는 맵 뷰는 QGIS 의"총화"입니다. 이 영역에 맵을 2 차원으로 표시하기 때문이죠. 이 창에 표시되는 맵은 사용자가 불러온 레이어에 적용한 (심볼, 라벨, 가시화 여부 등등) 렌더링을 반영할 것입니다. 또한 레이어 및 해당 프로젝트의 좌표계 (Coordinate Reference System) 에 따라 달라집니다.

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, 프로젝트 좌표계로 레이어 범위를"실시간 (on-the-fly)"으로 변환한 다음 레이어 범위로 맵 뷰를 확대하거나 축소합니다. 프로젝트에 이미 레이어가 존재하는 경우라면 맵 캔버스 크기 조정은 일어나지 않으므로 현재 맵 캔버스의 범위 안에 들어오는 피처들만 표시될 것입니다.

맵 뷰를 클릭하면 다음과 같은 대화형 작업을 할 수 있을 겁니다:

- Pan Map 도구, 화살표 키, 또는 `Space` 키나 마우스의 가운데 버튼 또는 마우스 휠을 누른 채 마우스를 움직이면 디스플레이 영역을 맵의 다른 지역으로 옮기며 이동할 수 있습니다.
- Zoom In 과 Zoom Out 전용 도구를 사용해서 맵을 확대/축소할 수 있습니다. 확대/축소 도구를 전환하려면 `Alt` 키를 누르십시오. 또 마우스 휠을 앞쪽으로 돌리면 확대, 뒤쪽으로 돌리면 축소됩니다. 확대/축소는 마우스 커서 위치를 중심으로 이루어집니다.

  Settings → *Options* → *Map tools* 메뉴에서 *Zoom factor* 를 사용자 지정할 수 있습니다.
- 불러온 모든 레이어들의 전체 영역으로 (Zoom Full) 확대/축소할 수 있고, 어느 레이어 영역으로 (Zoom to Layer) 확대/축소할 수도 있으며, 선택한 피처들의 영역으로 (Zoom to Selection) 확대/축소할 수도 있습니다.
- Zoom Last 및 Zoom Next 버튼이나 마우스의 전/후 버튼을 이용하면 캔버스 뷰 이력을 통해 전/후를 찾아갈 수 있습니다.

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

QGIS 는 기본적으로 *Layers* 패널에 단단히 연결된 ("메인 맵"이라고도 하는) 단일 맵 뷰를 엽니다. 메인 맵은 사용자가 *Layers* 패널 영역에서 작업한 변경 사항을 자동적으로 반영합니다. 그러나 *Layers* 패널의 현재 상태로부터 분기될 수 있는 내용을 가진 추가적인 맵 뷰를 열 수도 있습니다. 이 추가 맵 뷰는 2D 또는*3D* 유형일 수 있으며, 서로 다른 축척 또는 범위를 표시할 수도 있고, 또는 불러온 레이어들을맵 테마 를 통해 서로 다른 집합으로 표시할 수도 있습니다.

7.3.2 추가 맵 뷰 설정하기

새 맵 뷰를 추가하려면 *View* → *New Map View* 메뉴 항목을 선택하십시오. 메인 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.

추가된 맵 캔버스의 상단에는 다음 기능을 가진 툴바가 존재합니다:

- Zoom Full, Zoom to Selection 과 Zoom to Layer : 뷰 내부를 이동할 수 있습니다.

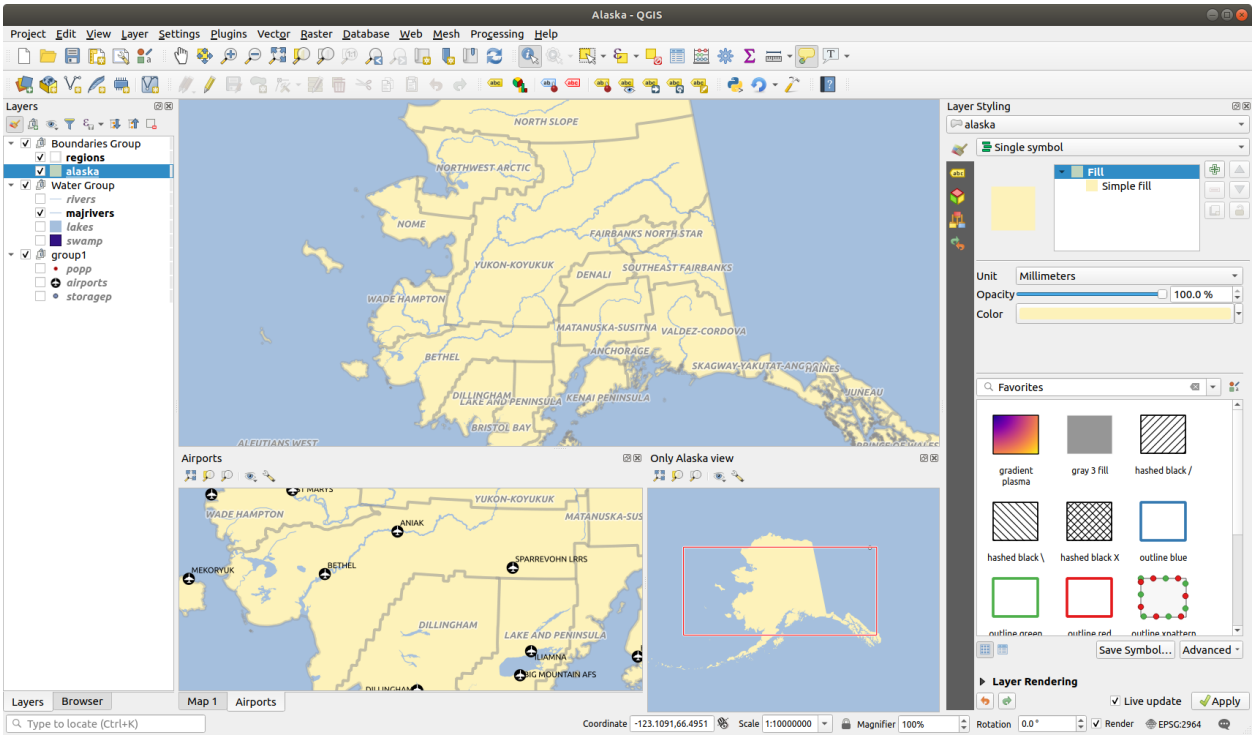


그림 7.4: 서로 다른 설정을 가진 다중 맵 뷰들

- 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 뷰를 설정합니다.
  - *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* : 축척을 설정합니다.
  - *Rotation* : 맵의 기울기를 설정합니다.
  - *Magnification* : 맵의 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* : 주석을 표시합니다.
  - *Show cursor position* : 커서 위치를 보여줍니다.
  - *Show main canvas extent* : 메인 캔버스의 범위를 표시합니다.
  - *Show labels*: 이 옵션을 해제하면 표시된 레이어의 속성에 라벨이 설정되어 있더라도 라벨을 보이지 않게 할 수 있습니다.
  - *Change map CRS*… : 좌표계를 변경합니다.
  - *Rename view*… : 뷰의 명칭을 다시 설정합니다.

7.3.3 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*…
   - or *Export Map to PDF*…

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

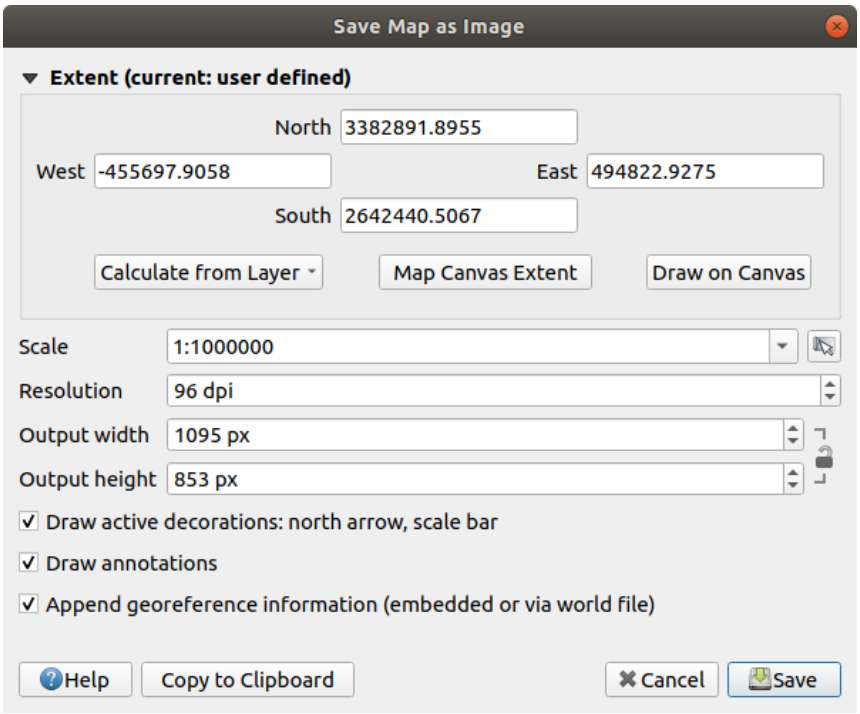


그림 7.5: 맵을 이미지로 저장 대화창

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.

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 명칭의 월드 파일 (PNG 이미지의 경우 확장자 PNGW, JPG 이미지의 경우 확장자 JPGW, …) 을 저장합니다. PDF 포맷의 경우 PDF 파일에 정보를 내장합니다.
8. PDF 로 내보낼 경우, *Save map as PDF*…대화창에서 더 많은 옵션을 설정할 수 있습니다:

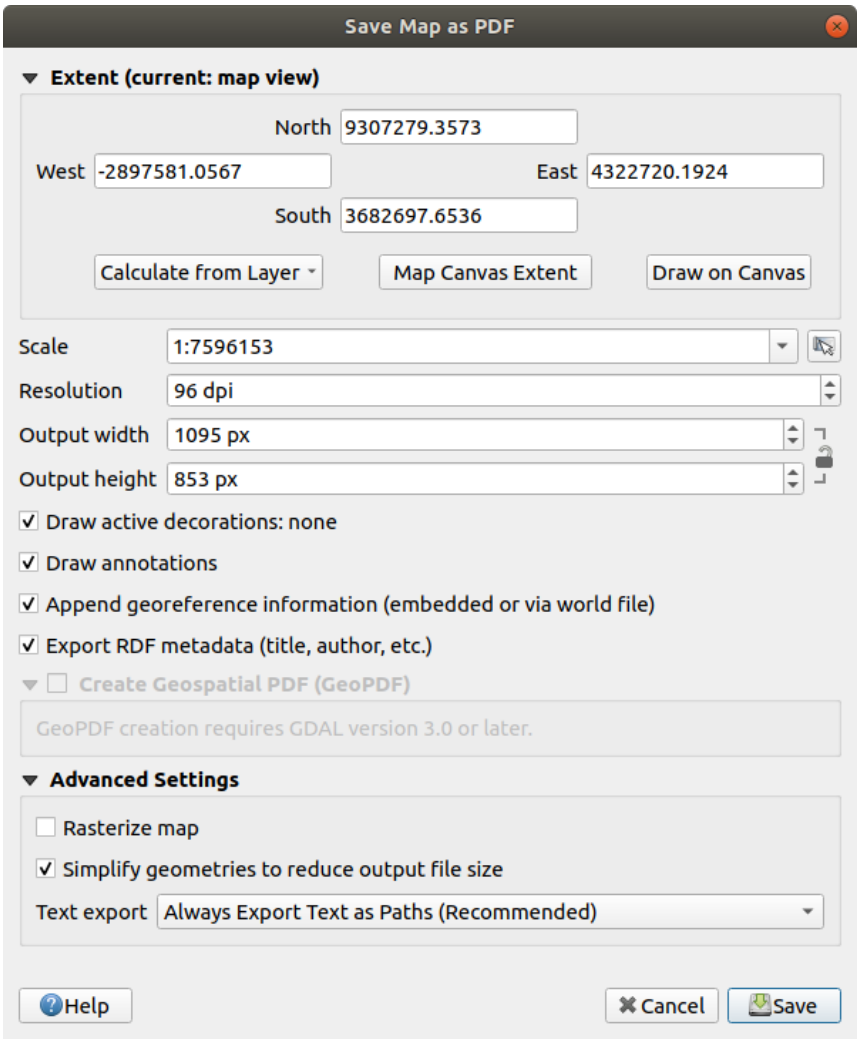


그림 7.6: 맵을 PDF 로 저장 대화창

- *Export RDF metadata* : 문서의 제목, 저자, 날짜, 설명 등등을 RDF 메타데이터로 내보냅니다.
- *Create Geospatial PDF (GeoPDF)*: 이 옵션을 활성화하면 지리참조 PDF 파일 을 생성합니다. (GDAL 버전 3 이상이 필요합니다.) 다음과 같은 옵션을 설정할 수 있습니다:
  - *Format* : GeoPDF 포맷을 선택합니다.
  - *Include vector feature information* : 산출물 GeoPDF 파일에 있는 맵 안에 가시화된 피처들의 모든 도형 및 속성 정보를 GeoPDF 파일에 내장시킬 것입니다.

참고: QGIS 3.10 버전부터, GDAL 3 를 설치한 경우 GeoPDF 파일을 데이터소스로 이용할 수도

있습니다. QGIS 의 GeoPDF 지원에 대해 더 자세한 정보를 알고 싶다면 https://north-road.com/2019/09/03/qgis-3-10-loves-geopdf/ 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 용량 및 복잡도를 줄일 수 있습니다. (다른 응용 프로그램에서 대용량 파일을 불러올 수 없을 수도 있기 때문입니다.)
- *Text export*: 텍스트 라벨을 제대로 된 텍스트 오브젝트로 내보낼지 (*Always export texts as text objects*) 경로로만 내보낼지 (*Always export texts as paths*) 설정합니다. 텍스트 라벨을 텍스트 객체로 내보낼 경우 (잉크스케이프 같은) 외부 응용 프로그램에서 일반 텍스트처럼 편집할 수 있습니다. **하지만** 렌더링의 질이 하락하는 부작용이 있습니다. **게다가** 버퍼 같은 특정 텍스트 설정이 적용된 경우 렌더링 오류가 생길 수도 있습니다. 경로로 내보내기를 추천하는 이유입니다.

9. *Save* : 파일의 위치, 명칭, 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.

7.4 3D 맵 뷰

3D 맵 뷰를 통해 3D 가시화를 지원합니다. *View* → *New 3D Map View* 메뉴를 선택하면 3D 맵 뷰를 생성해서 열 수 있습니다. 해당 메뉴를 클릭하면 QGIS 의 플로팅 패널이 나타날 것입니다. 이 패널을 다른 창에 도킹시킬 수 있습니다.

먼저, 3D 맵 뷰는 2D 메인 맵 캔버스와 동일한 범위와 뷰를 표시합니다. 이 뷰를 3D 로 변환시키는 데 사용할 수 있는 탐색 (navigation) 도구 모음이 있습니다.

3D 맵 뷰 최상단에 다음과 같은 도구들이 있습니다:

- Camera control: 카메라의 각도와 방향을 동일하게 유지하면서 뷰를 이동합니다.
- Zoom Full: 뷰의 크기를 전체 레이어 범위로 다시 조정합니다.
- Toggle on-screen notification: (맵 뷰를 더 쉽게 조정하기 위한) 탐색 위젯을 켜고 끕니다.
- Identify: 지형에서 클릭한 포인트 또는 클릭한 3D 피처 (들) 에 대한 정보를 반환합니다. 더 자세한 정보는피처 식별 를 참조하세요.
- Measurement line: 두 포인트 사이의 수평 거리를 측정합니다.
- Animations: 애니메이션 플레이어 위젯을 켜고 끕니다.
- Save as image…: 현재 뷰를 이미지 파일 포맷으로 내보냅니다.
- Export 3D Scene…: 현재 뷰를 (`.obj` 파일 형식의) 3D 신 (scene) 으로 내보내, 블렌더 등의 응용 프로그램에서 후처리할 수 있게 해줍니다. 지형 및 벡터 피처를 3D 객체로 내보냅니다. 레이어속성 및 맵 뷰환경 설정 을 무시하는 내보내기 설정에는 다음이 포함됩니다:
  - *Scene name* 및 *Folder*: 신 명칭 및 대상 폴더
  - *Terrain resolution*: 지형 해상도

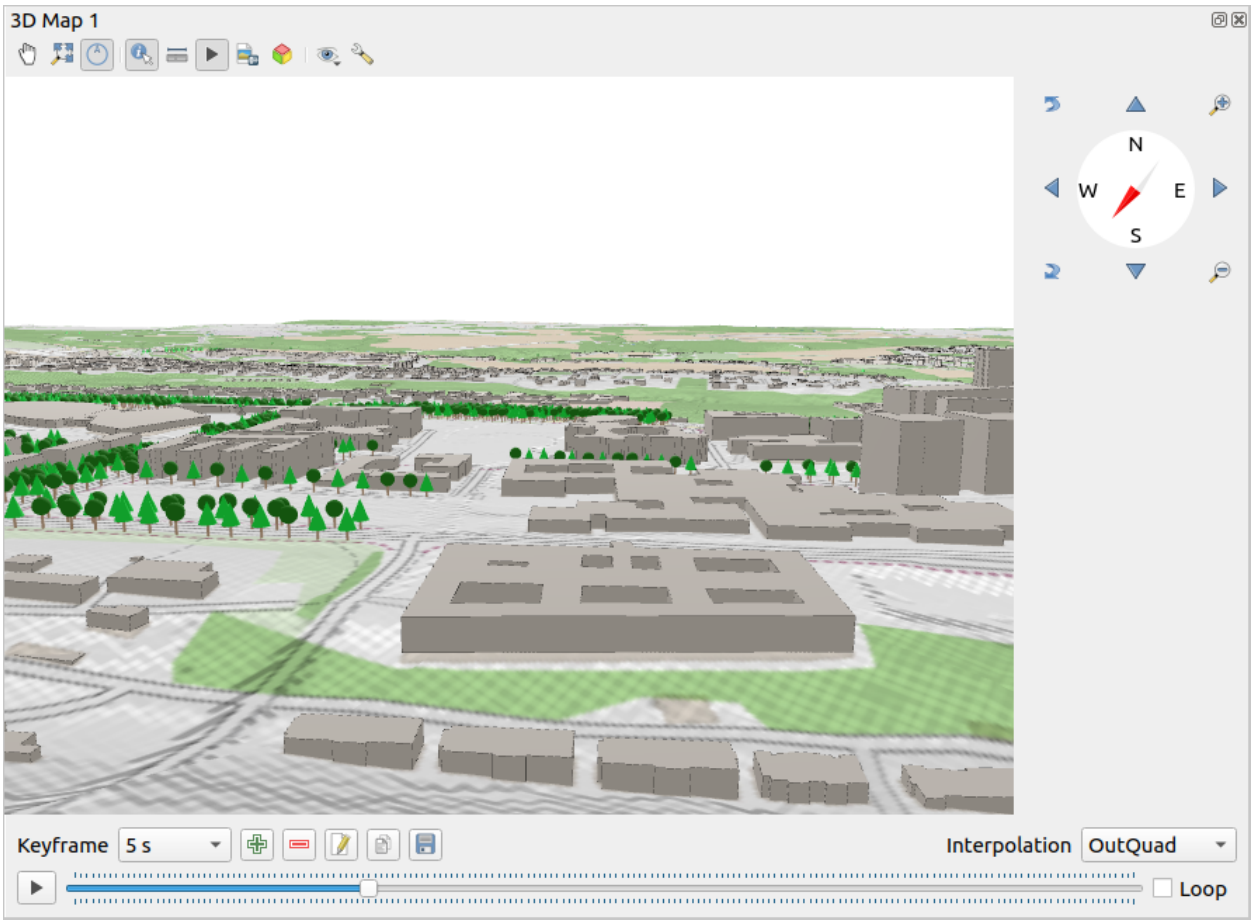


그림 7.7: 3D 맵 뷰 대화창

  - *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 모음을 선택할 수 있도록 해줍니다.
-  Configure: 맵 뷰를설정 합니다.

7.4.1 탐색 옵션

맵 뷰를 3D 로 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 과  Zoom Out 도구를 클릭합니다.
  - 마우스 휠을 돌립니다. (`Ctrl` 키를 누른 채 돌리면 더 섬세하게 확대/축소할 수 있습니다.)
  - 마우스 오른쪽 버튼을 누른 채 드래그하면 확대 (아래쪽) 하거나 축소 (위쪽) 할 수 있습니다.

카메라 뷰를 초기화하려면 3D 캔버스 패널 최상단에 있는  Zoom Full 버튼을 클릭하십시오.

7.4.2 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/qt-5/qeasingcurve.html#EasingFunction-typedef 를 참조하세요.)

   타임 슬라이드 막대를 움직여서도 애니메이션을 미리보기할 수 있습니다. Repeat 버튼을 켜면 애니메이션을 반복해서 실행할 것입니다. 이때 애니메이션을 멈추려면 버튼을 클릭하십시오.

Keyframe 목록을 이용해서 카메라의 서로 다른 뷰들을 탐색할 수 있습니다. 타임셋이 활성화된 상태에서 맵 뷰를 변경하면 자동적으로 관련된 위치를 업데이트할 것입니다. Edit keyframe 버튼으로 키프레임의 시간을 편집하거나 Remove keyframe 버튼으로 키프레임을 삭제할 수도 있습니다.

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.

7.4.3 신 (scene) 설정

3D 맵 뷰는 기본 설정값으로 열리지만, 이를 사용자 지정할 수 있습니다. 3D 캔버스 패널 최상단의 Configure… 버튼을 클릭하면 *3D configuration* 창이 열립니다.

3D 설정 대화창에는 3D 신을 미세 조정할 수 있는 다양한 옵션들이 있습니다:

지형

- *Terrain*: 자세한 내용으로 들어가기 전에, 3D 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 2D 맵 레이어에서 추출한 텍스처로 이루어져 있습니다.
  - 표고 지형의 *Type* 으로 다음 가운데 하나를 선택할 수 있습니다:
    - *Flat terrain*
    - 불러온 *DEM (Raster Layer)*
    - Mapzen 도구로 생성한 표고 타일 을 불러오는 *Online* 서비스 — 자세한 내용은 https://registry.opendata.aws/terrain-tiles/ 를 참조하세요.
    - 불러온 *Mesh* 데이터셋
  - *Elevation*: 지형을 생성하기 위해 사용되는 래스터 또는 메시 레이어입니다. 래스터 레이어는 표고를 나타내는 밴드를 담고 있어야 합니다. 메시 레이어의 경우, 꼭짓점의 Z 값을 사용합니다.

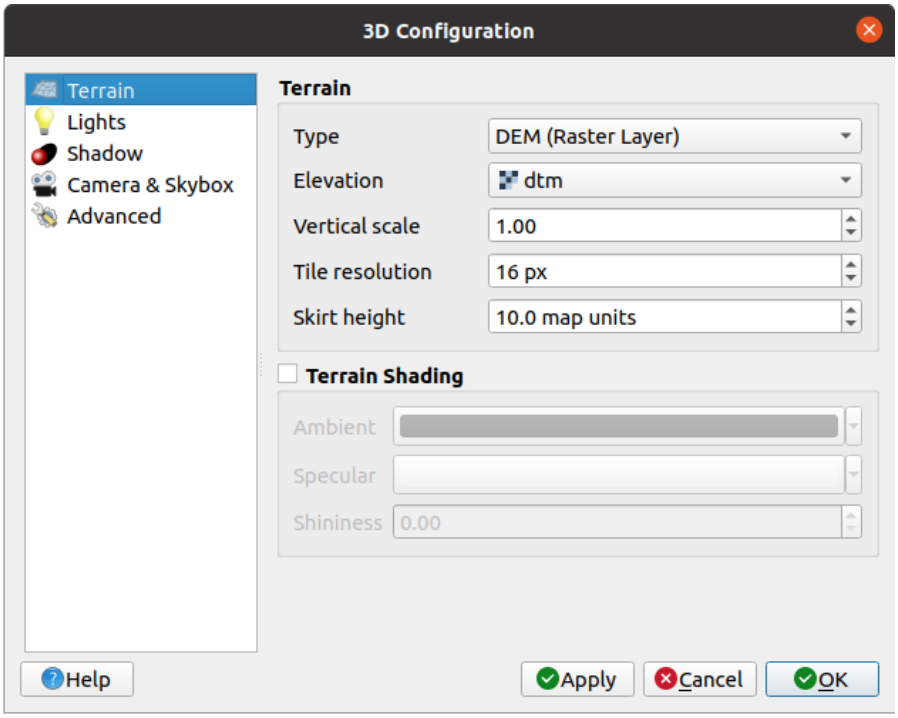


그림 7.8: 3D 맵 설정 대화창

  - *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.
- 메시 레이어를 지형으로 사용하는 경우, (와이어프레임 표시, 삼각형 평활화 등을 설정하는) *Triangles settings* 및 (렌더링 색상을 균일하게 할지 또는 지형 수준에 따라 달리 할지를 설정하는) *Rendering colors settings* 를 환경 설정할 수 있습니다. 메시 레이어 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.

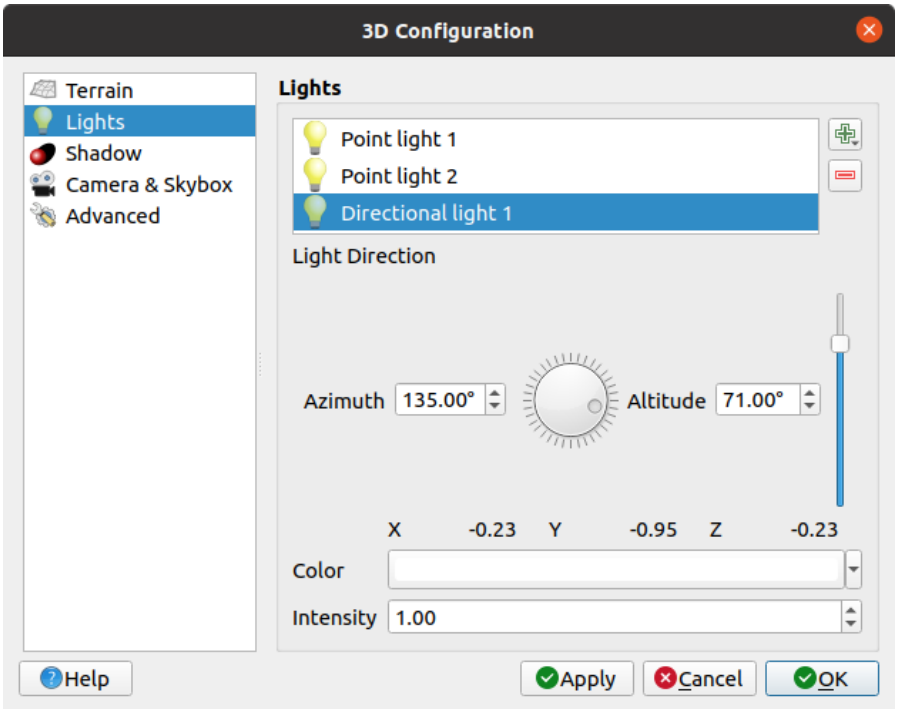


그림 7.9: 3D 맵 조명 환경 설정 대화창

음영

Show shadow 옵션을 체크하면 사용자 신에 음영을 표시하는데, 다음을 지정해야 합니다:

- *Directional light*
- *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.

카메라와 스카이박스

- 카메라의 *Field of view*: 파노라마 신을 생성할 수 있습니다. 기본값은 45° 입니다.
- ☐ *Show skybox* 옵션을 체크하면 신에 스카이박스를 렌더링할 수 있습니다. 다음과 같은 스카이박스 유형 가운데 선택할 수 있습니다:
  - *Panoramic texture*: 360° 하늘 풍경을 단일 파일로 제공합니다.
  - *Distinct faces*: 신을 담고 있는 육면체의 각 면을 각각의 텍스처 파일로 제공합니다.

  텍스처 파일은 디스크 상에 있는 파일이나 원격 URL 이 될 수도 있고, 또는 프로젝트에 내장되어 있을 수도 있습니다. (자세한 내용은여기를 참조)

고급

- *Map tile resolution*: 2D 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 3D 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 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 light sources*: 광원 위치에 구체를 표시해서 신 내용에 따라 광원을 더 쉽게 배치/재배치할 수 있습니다.

7.4.4 3D 벡터 레이어

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

7.5 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.

7.5.1 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: 활성화 레이어의 필드 가운데 단일 필드만 탐색하도록 제한하기

"활성화 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.

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.

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.

7.5.2 액션 보고하기

위치 탐색란 옆의 영역에는 필요한 경우 (예를 들어 레이어에 있는 피처를 선택했다거나, 레이어를 제거했다거나 하는) 사용자가 수행한 작업의 요약이 표시되거나, 사용자가 마우스를 가져다 댄 도구에 대한 자세한 설명이 표시될 것입니다. (모든 도구가 자세한 설명을 지원하지는 않습니다.)

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.

7.5.3 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 클릭하면 확대경으로 확대/축소할 수 있도록 축척을 고정시킵니다. 맵 축척을 변경하지 않고 맵을 확대할 수 있어, 라벨 및 심볼의 위치를 정밀하게 조정하는 데 편리합니다. 확대 수준은 백분율로 표시됩니다. 만약 *Magnifier* 가 100% 수준이라면 현재 맵은 확대되지 않은 상태입니다. 또한 *Settings → Options → Rendering → Rendering behavior* 메뉴에서 확대 기본값을 정의할 수 있습니다. 이 옵션은 고해상도 모니터에서 심볼이 너무 작아지지 않게 하는 데 매우 유용합니다.

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

상태 바의 우측에 레이어가 맵 뷰에 렌더링되는 것을 일시적으로 막는 데 쓰이는 작은 체크박스가 있습니다. (렌더링 작업 을 참조하십시오.)

렌더링 기능 오른쪽에 현재 투영 좌표계를 보여주는 *EPSG:code* 버튼이 있습니다. 이 아이콘을 클릭하면 *Project Properties* 대화창이 열려, 맵 뷰에 또다른 좌표계를 적용할 수 있습니다.

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 메뉴 옵션에서 실행 시 기본적으로 적용할 좌표계를 설정할 수 있다는 사실을 알아두십시오.

7.5.4 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.

플러그인 관리자 설정 에 따라, 상태 바 우측에 새로 설치할 수 있는 ( ) 플러그인 또는 업데이트할 수 있는 ( ) 플러그인을 알려주는 아이콘이 나타날 때가 있습니다. 해당 아이콘을 클릭하면 플러그인 관리자 대화창이 열립니다.

CHAPTER 8

탐색기 패널

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 그래픽 모델 생성기 (graphical modeller) 에서도 열고 편집할 수 있습니다.

Layers 패널에서 *Browser* 패널로, 예를 들어 GeoPackage 또는 PostGIS 데이터베이스로 레이어를 드래그 & 드롭할 수 있습니다.

탐색기 패널 (그림 8.1) 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. ▼ 아이콘을 클릭하면 해당 줄기 (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 잎 (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*

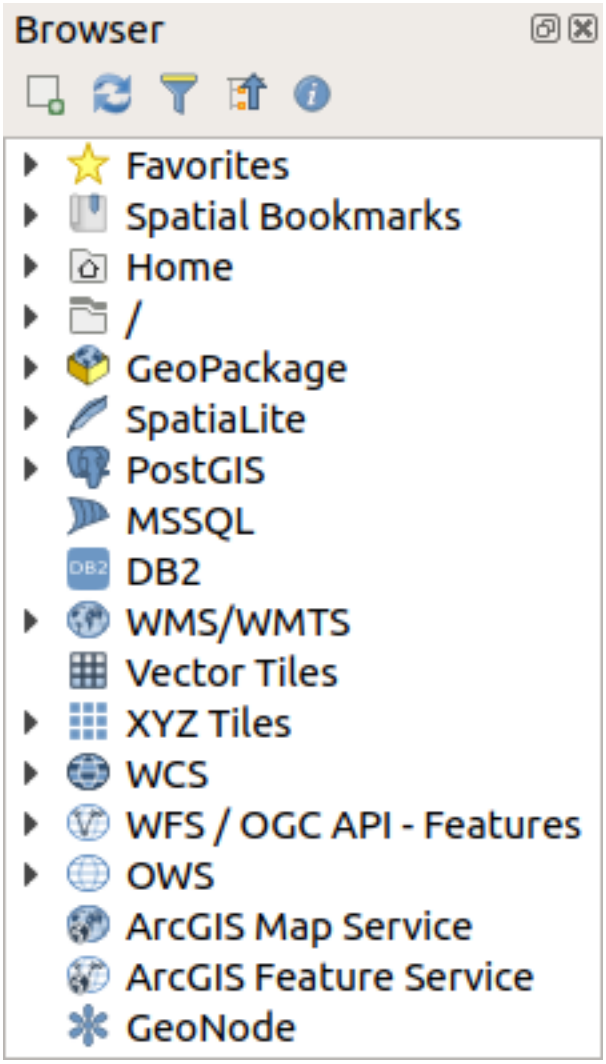

Browser
Favorites
Spatial Bookmarks
Home
/
GeoPackage
SpatiaLite
PostGIS
MSSQL
DB2
WMS/WMTS
Vector Tiles
XYZ Tiles
WCS
WFS / OGC API - Features
OWS
ArcGIS Map Service
ArcGIS Feature Service
GeoNode

그림 8.1: 탐색기 패널

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 하단에속성 위젯 그림처럼 표시됩니다.

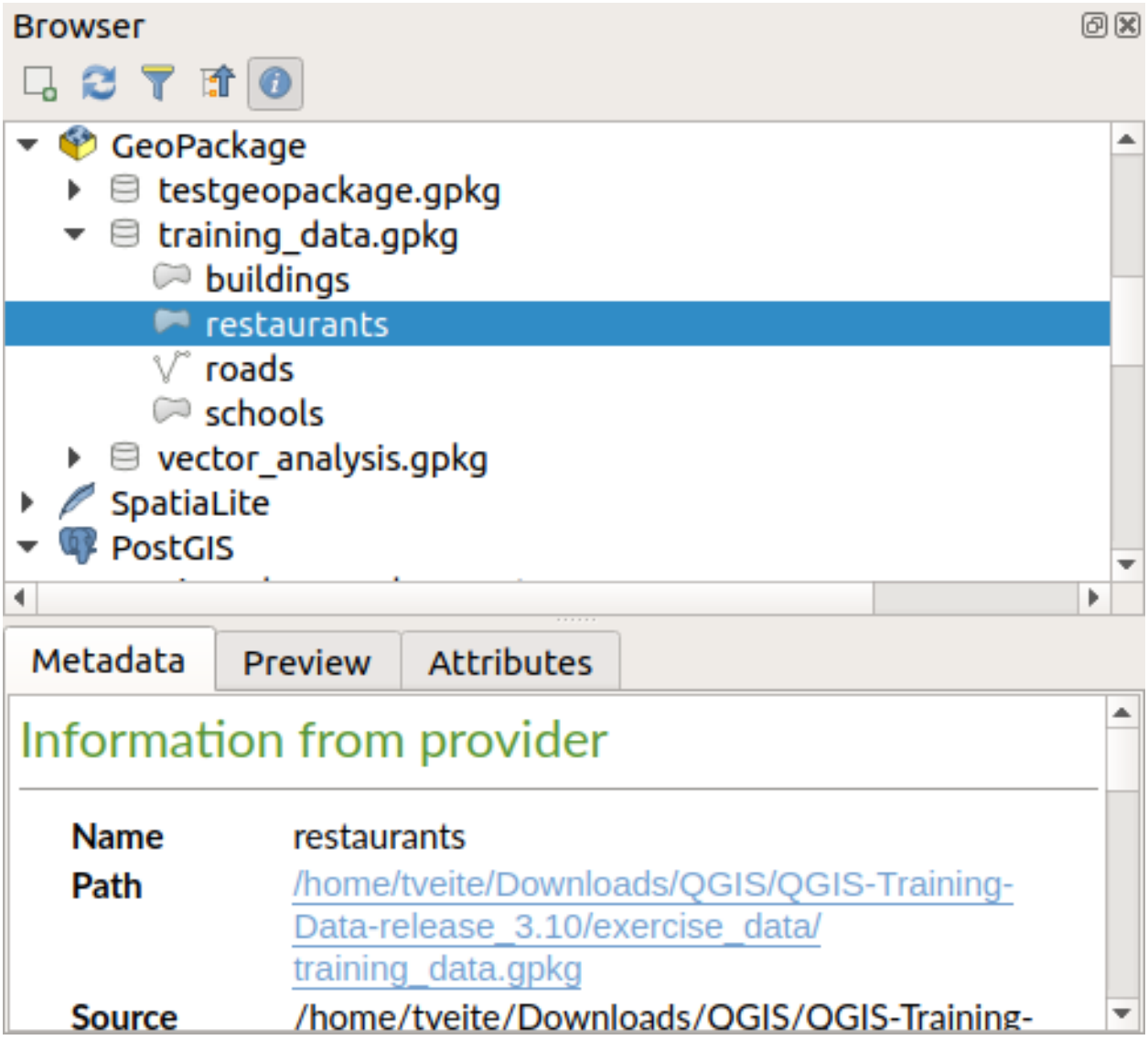


그림 8.2: 속성 위젯

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

8.1 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 1 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.

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

8.2 탐색기 패널 최상위 수준 항목들

8.2.1 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.

8.2.2 공간 북마크

여기에서 *Project Bookmarks* 와 *User Bookmarks* 로 정리된 사용자 공간 북마크를 찾을 수 있을 것입니다.

최상위 수준 컨텍스트 메뉴에서, 북마크를 생성하고 (*New Spatial Bookmark*…), 공간 북마크 관리자 대화창을 열고 (*Show the Spatial Bookmark Manager*), 공간 북마크를 가져오거나 (*Import Spatial Bookmarks*…) 공간 북마크를 내보낼 수 있습니다 (*Export Spatial Bookmarks*…).

북마크 항목을 통해 공간 북마크가 가리키는 영역으로 확대/축소하고 (*Zoom to Bookmark*), 공간 북마크를 편집하고 (*Edit Spatial Bookmark*…), 공간 북마크를 삭제할 수 있습니다 (*Delete Spatial Bookmark*).

8.2.3 홈

사용자 파일 시스템의 홈 디렉터리/폴더입니다. 이 항목을 오른쪽 클릭하고 *Add as a Favorite* 을 선택하면 *Favorites* 에 이 위치를 추가할 것입니다. 이 항목의 컨텍스트 메뉴에서:

- *Add*: 디렉터리, Geopackage 또는 ESRI Shapefile 데이터셋을 추가할 수 있습니다.
- *Hide from Browser*: 홈 디렉터리를 숨길 수 있습니다.
- *Fast Scan this Directory*: 빠른 스캔 기능을 켜고 끌 수 있습니다.
- *Open Directory*: 사용자의 파일 관리자에서 홈 디렉터리를 열 수 있습니다.
- *Open in Terminal*: 터미널창에서 홈 디렉터리를 열 수 있습니다.

- *Properties…*, *Directory Properties…*: 홈 디렉터리의 속성을 검사할 수 있습니다.

8.2.4 /

사용자 파일 시스템의 루트 디렉터리/폴더입니다.

8.2.5 Geopackage

Geopackage 파일/데이터베이스입니다. 최상위 수준 컨텍스트 메뉴에서 Geopackage 파일/데이터베이스를 생성하거나 (*Create Database…*) 기존 Geopackage 파일/데이터베이스를 추가할 수 있습니다 (*New Connection…*).

각 Geopackage 의 컨텍스트 메뉴를 통해 목록에서 해당 Geopackage 를 제거하고 (*Remove connection…*), 해당 Geopackage 에 새 레이어 또는 테이블을 추가하고 (*Create new Layer or Table…*), 해당 Geopackage 를 제거하고 (*Delete <name of geopackage>),* 데이터베이스 용량을 확보할 수 있습니다 *(:guilabel:`Compact Database (VACUUM)*).

레이어/테이블 항목의 경우:

- *Rename Layer <layer name>…*: 재명명할 수 있습니다.
- *Export Layer → To file*: 파일로 내보낼 수 있습니다.
- *Add Layer to Project*: 프로젝트에 추가할 수 있습니다.
- *Delete Layer*: 삭제할 수 있습니다.
- *Layer Properties…*, *File Properties…*: 속성을 검사할 수 있습니다.

8.2.6 SpatiaLite

SpatiaLite 데이터베이스 연결입니다.

최상위 수준 컨텍스트 메뉴에서 SpatiaLite 파일/데이터베이스를 생성하거나 (*Create Database…*) 기존 SpatiaLite 파일/데이터베이스를 추가할 수 있습니다 (*New Connection…*).

각 SpatiaLite 파일의 컨텍스트 메뉴를 통해 SpatiaLite 연결을 삭제할 수 있습니다 (*Delete*).

레이어/테이블 항목의 경우:

- *Export Layer → To file*: 파일로 내보낼 수 있습니다.
- *Add Layer to Project*: 프로젝트에 추가할 수 있습니다.
- *Delete Layer*: 삭제할 수 있습니다.
- *Layer Properties…*: 속성을 검사할 수 있습니다.

8.2.7 PostGIS

PostGIS 데이터베이스 연결입니다.

최상위 수준 컨텍스트 메뉴에서 새 PostGIS 연결을 추가할 수 있습니다 (*New Connection…*).

각 연결의 컨텍스트 메뉴에서 연결을 새로고침하거나 (*Refresh*), 편집하거나 (*Edit connection…*), 삭제하거나 (*Delete connection*), 스키마를 생성할 수 있습니다 (*Create Schema…*).

각 스키마의 컨텍스트 메뉴에서 새로고침하거나 (*Refresh*), 재명명하거나 (*Rename Schema…*), 삭제할 수 있습니다 (*Delete Schema*).

레이어/테이블 항목의 경우:

- *Rename Table*…: 재명명할 수 있습니다.
- *Truncate Table*…: 내용을 제거할 수 있습니다.
- *Export Layer → To file*: 파일로 내보낼 수 있습니다.
- *Add Layer to Project*: 프로젝트에 추가할 수 있습니다.
- *Delete Layer*: 삭제할 수 있습니다.
- *Layer Properties*…: 속성을 검사할 수 있습니다.

8.2.8 MSSQL

Microsoft SQL 서버 연결입니다.

최상위 수준 컨텍스트 메뉴에서 새 PostGIS 연결을 추가할 수 있습니다 (*New Connection*…).

각 연결의 컨텍스트 메뉴에서 연결을 새로고침하거나 (*Refresh*), 편집하거나 (*Edit connection*…), 삭제하거나 (*Delete connection*), 스키마를 생성할 수 있습니다 (*Create Schema*…).

각 스키마의 컨텍스트 메뉴에서 새로고침하거나 (*Refresh*), 재명명하거나 (*Rename Schema*…), 삭제할 수 있습니다 (*Delete Schema*).

레이어/테이블 항목의 경우:

- *Rename Table*…: 재명명할 수 있습니다.
- *Truncate Table*…: 내용을 제거할 수 있습니다.
- *Export Layer → To file*: 파일로 내보낼 수 있습니다.
- *Add Layer to Project*: 프로젝트에 추가할 수 있습니다.
- *Delete Layer*: 삭제할 수 있습니다.
- *Layer Properties*…: 속성을 검사할 수 있습니다.

8.2.9 DB2

IBM DB2 데이터베이스 연결입니다.

최상위 수준 컨텍스트 메뉴에서 새 PostGIS 연결을 추가할 수 있습니다 (*New Connection*…).

각 연결의 컨텍스트 메뉴에서 연결을 새로고침하거나 (*Refresh*), 편집하거나 (*Edit connection*…), 삭제하거나 (*Delete connection*), 스키마를 생성할 수 있습니다 (*Create Schema*…).

각 스키마의 컨텍스트 메뉴에서 새로고침하거나 (*Refresh*), 재명명하거나 (*Rename Schema*…), 삭제할 수 있습니다 (*Delete Schema*).

레이어/테이블 항목의 경우:

- *Rename Table*…: 재명명할 수 있습니다.
- *Truncate Table*…: 내용을 제거할 수 있습니다.
- *Export Layer → To file*: 파일로 내보낼 수 있습니다.
- *Add Layer to Project*: 프로젝트에 추가할 수 있습니다.
- *Delete Layer*: 삭제할 수 있습니다.
- *Layer Properties*…: 속성을 검사할 수 있습니다.

8.2.10 WMS/WMTS

웹 맵 서비스 (WMS; Web Map Service) 와 웹 맵 타일 서비스 (WMTS; Web Map Tile Service) 입니다.

최상위 수준 컨텍스트 메뉴에서 새 PostGIS 연결을 추가할 수 있습니다 (*New Connection…*).

각 WSM/WMTS 서비스의 컨텍스트 메뉴에서 새로고침하거나 (*Refresh*), 편집하거나 (*Edit…*), 삭제할 수 있습니다 (*Delete*).

맵 캔버스 상에 WMS/WMTS 서비스를 드래그하면 그룹 레이어를 추가할 수 있습니다.

WMS/WMTS 레이어 항목의 경우:

- *Export Layer → To file*: 파일로 내보낼 수 있습니다.
- *Add Layer to Project*: 프로젝트에 추가할 수 있습니다.
- *Layer Properties…*: 속성을 검사할 수 있습니다.

8.2.11 벡터 타일

벡터 타일 서비스입니다.

최상위 수준 컨텍스트 메뉴에서 기존 서비스를 추가하고 (*New Connection…*), XML 파일로 연결을 저장하거나 (*Save Connections…*) XML 파일에서 불러올 수 있습니다 (*Load Connections…*).

8.2.12 XYZ 타일

XYZ 타일 서비스입니다.

최상위 수준 컨텍스트 메뉴에서 기존 서비스를 추가하고 (*New Connection…*), XML 파일로 연결을 저장하거나 (*Save Connections…*) XML 파일에서 불러올 수 있습니다 (*Load Connections…*).

XYZ 타일 서비스 항목의 경우:

- *Edit…*: 편집할 수 있습니다.
- *Delete*: 삭제할 수 있습니다.
- *Export Layer → To file*: 파일로 내보낼 수 있습니다.
- *Add Layer to Project*: 프로젝트에 추가할 수 있습니다.
- *Layer Properties…*: 속성을 검사할 수 있습니다.

8.2.13 WCS

웹 커버리지 서비스 (WCS; Web Coverage Service) 입니다.

최상위 수준 컨텍스트 메뉴에서 새 PostGIS 연결을 추가할 수 있습니다 (*New Connection…*).

각 WCS 서비스의 컨텍스트 메뉴에서 새로고침하거나 (*Refresh*), 편집하거나 (*Edit…*), 삭제할 수 있습니다 (*Delete*).

WCS 레이어 항목의 경우:

- *Export Layer → To file*: 파일로 내보낼 수 있습니다.
- *Add Layer to Project*: 프로젝트에 추가할 수 있습니다.
- *Layer Properties…*: 속성을 검사할 수 있습니다.

8.2.14 WFS/OGC API 피처

웹 피처 서비스 (WFS; Web Feature Service) 와 (WFS3 이라고도 하는) *OGC API 피처* 서비스 입니다.

최상위 수준 컨텍스트 메뉴에서 새 PostGIS 연결을 추가할 수 있습니다 (*New Connection…*).

각 WFS 서비스의 컨텍스트 메뉴에서 새로고침하거나 (*Refresh*), 편집하거나 (*Edit…*), 삭제할 수 있습니다 (*Delete*).

WFS 레이어 항목의 경우:

- *Export Layer To file*: 파일로 내보낼 수 있습니다.
- *Add Layer to Project*: 프로젝트에 추가할 수 있습니다.
- *Layer Properties…*: 속성을 검사할 수 있습니다.

8.2.15 OWS

여기에서 사용자의 모든 오픈 웹 서비스 (OWS; Open Web Service) 의 — WMS, WCS, WFS 등등 — 읽기 전용 목록을 찾을 수 있을 것입니다.

8.2.16 ArcGIS 맵 서비스

8.2.17 ArcGIS 피처 서비스

8.2.18 GeoNode

최상위 수준 컨텍스트 메뉴에서 새 PostGIS 연결을 추가할 수 있습니다 (*New Connection…*).

각 서비스의 컨텍스트 메뉴에서 새로고침하거나 (*Refresh*), 편집하거나 (*Edit…*), 삭제할 수 있습니다 (*Delete*).

서비스 레이어 항목의 경우:

- *Export Layer To file*: 파일로 내보낼 수 있습니다.
- *Add Layer to Project*: 프로젝트에 추가할 수 있습니다.
- *Layer Properties…*: 속성을 검사할 수 있습니다.

8.3 리소스

- 프로젝트 파일. QGIS 프로젝트 파일의 컨텍스트 메뉴에서:
  - *Open Project*: 열 수 있습니다.
  - *Extract Symbols…*: 심볼을 추출할 수 있습니다. 스타일 관리자 대화창이 열려 XML 파일로 심볼을 내보내거나, 기본 스타일에 추가하거나, PNG 또는 SVG 로 내보낼 수 있습니다.
  - *File Properties…*: 속성을 검사할 수 있습니다.

  프로젝트 파일을 펼쳐서 프로젝트 레이어를 볼 수 있습니다. 레이어의 컨텍스트 메뉴를 통해 탐색기가 공통적으로 제공하는 액션을 사용할 수 있습니다.
- QGIS 레이어 정의 파일 (QLR). 컨텍스트 메뉴에서 다음 액션을 사용할 수 있습니다:
  - *Export Layer To file*: 파일로 내보낼 수 있습니다.
  - *Add Layer to Project*: 프로젝트에 추가할 수 있습니다.

  - *Layer Properties*…: 속성을 검사할 수 있습니다.
- 공간 처리 모델 (.model3). 컨텍스트 메뉴에서 다음 액션을 사용할 수 있습니다:
  - *Run Model*…: 모델을 실행합니다.
  - *Edit Model*…: 모델을 편집합니다.
- QGIS 인쇄 조판기 템플릿 (QPT). 컨텍스트 메뉴에서 다음 액션을 사용할 수 있습니다:
  - *New Layout from Template*: 새 조판을 템플릿으로 생성합니다.
- 파이썬 스크립트 (.py). 컨텍스트 메뉴에서 다음 액션을 사용할 수 있습니다:
  - *Run script*…: 스크립트를 실행합니다.
  - *Open in External Editor*: 스크립트를 외부 편집기로 엽니다.
- 잘 알려진 래스터 포맷들. 컨텍스트 메뉴에서 다음 액션을 사용할 수 있습니다:
  - *Delete File <dataset name>*: 파일을 삭제합니다.
  - *Export Layer → To file*: 파일로 내보낼 수 있습니다.
  - *Add Layer to Project*: 프로젝트에 추가할 수 있습니다.
  - *Layer Properties*…, *File Properties*…: 속성을 검사할 수 있습니다.

  몇몇 포맷의 경우 *Open <file type> Externally*…메뉴도 사용할 수 있습니다.
- 잘 알려진 벡터 포맷들. 컨텍스트 메뉴에서 다음 액션을 사용할 수 있습니다:
  - *Delete File <dataset name>*: 파일을 삭제합니다.
  - *Export Layer → To file*: 파일로 내보낼 수 있습니다.
  - *Add Layer to Project*: 프로젝트에 추가할 수 있습니다.
  - *Layer Properties*…, *File Properties*…: 속성을 검사할 수 있습니다.

  몇몇 포맷의 경우 *Open <file type> Externally*…메뉴도 사용할 수 있습니다.

CHAPTER 9

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 으로 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.

9.1 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.

9.1.1 일반 설정

시스템 로케일 무시

QGIS 는 기본적으로 사용자의 운영체제 설정에 의존해서 언어를 설정하고 수치 (numerical value) 를 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 정해진 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* 를 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)
- ☐ *Modeless data source manager dialog* : 프로젝트에 레이어를 추가하는 동안데이터소스 관리자 대화창을 열어둔 채 QGIS 인터페이스와 상호작용시킬지 여부를 설정합니다.

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 macros* 옵션은 프로젝트 이벤트에 대한 액션을 수행하기 위해 작성된 매크로를 처리하기 위해 만들어졌습니다. 'Never', 'Ask', 'For this session only'그리고'Always (not recommended)'가운데 선택할 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.

9.1.2 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.

심볼 또는 라벨에 SVG 파일을 사용할 경우, QGIS 는 다음과 같은 작업을 할 수 있습니다:

- 파일 시스템에서 파일 불러오기: 파일 경로를 통해 파일을 식별하는데, QGIS 는 대응하는 이미지를 표시하기 위해 경로를 분해해야 합니다.
- 원격 URL 에서 파일 불러오기: 마찬가지로, 원격 리소스에서 성공적으로 검색되어야 이미지를 불러올 것입니다.
- 항목 내에 SVG 파일 내장시키기: 현재 프로젝트, 스타일 데이터베이스, 또는 인쇄 조판 템플릿 안에 파일을 내장시킵니다. 그러면 언제나 SVG 파일을 항목의 일부로 렌더링할 것입니다. 이 방법으로 사용자 지정 SVG 심볼을 내장한 프로젝트를 서로 다른 사용자와 QGIS 설치판 사이에서 쉽게 공유할 수 있도록 간편하게 생성할 수 있습니다.

  심볼 또는 라벨에서 내장 SVG 파일을 추출해서 디스크에 저장할 수도 있습니다.

참고: 앞에서 언급한 프로젝트에 SVG 파일을 불러오거나 저장하는 옵션을, 사용자 지정려고 하는 심볼, 라벨, 장식에 사용할 래스터 이미지에도 적용할 수 있습니다.

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 링크가 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 시험용 문서로 연결합니다.

설정

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

환경

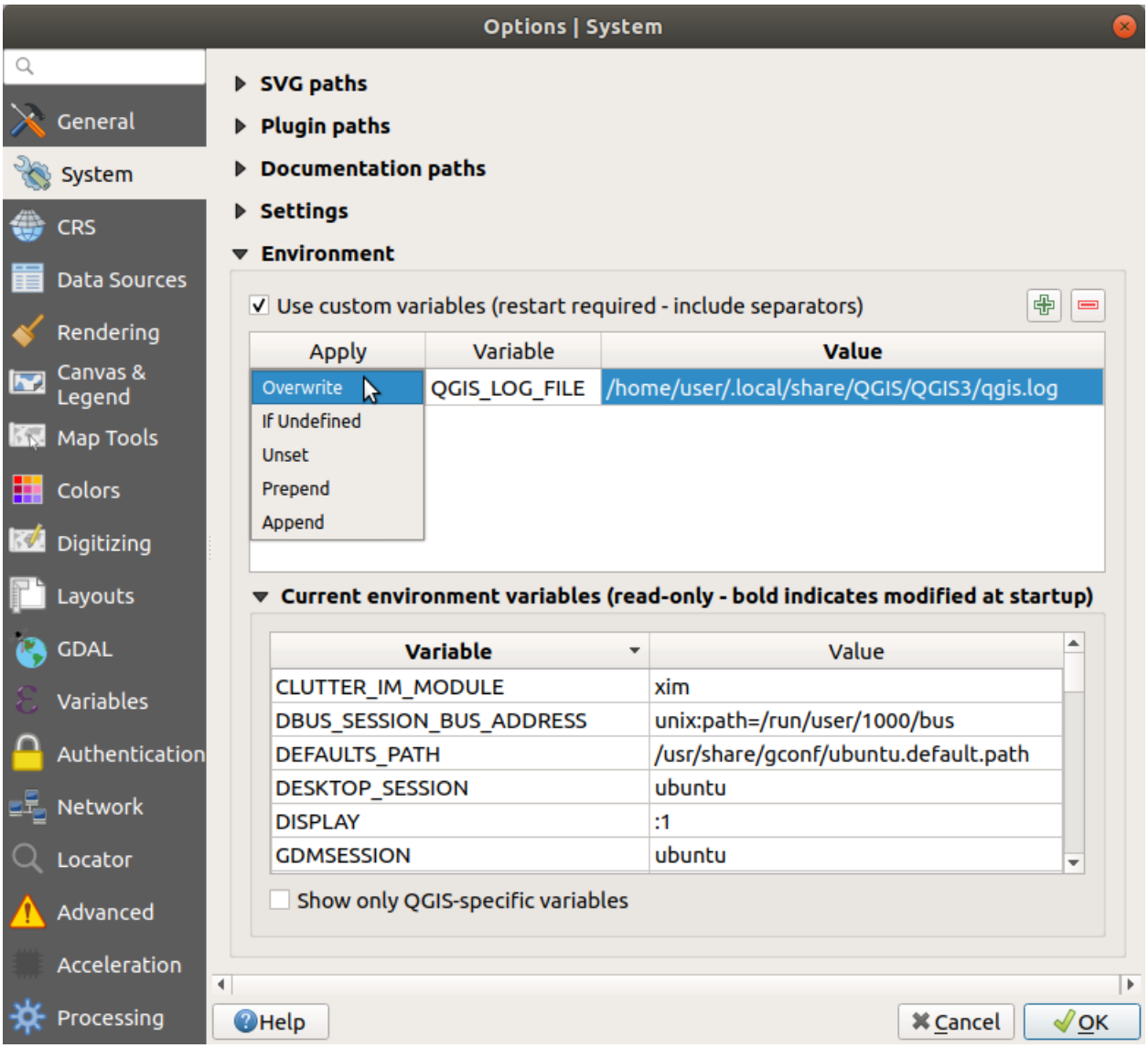


그림 9.1: QGIS 의 시스템 환경 변수

Environment 그룹에서 시스템 환경 변수를 살펴볼 수 있고, 많은 변수를 설정할 수 있습니다. 이 옵션은 맥 OS 처럼 GUI 응용 프로그램이 사용자의 셸 환경을 반드시 상속하지는 않는 플랫폼에서 유용합니다. SAGA, GRASS 같은 공간 처리 툴박스가 제어하는 외부 도구 모음을 위한 환경 변수들을 설정하고 살펴보는 데에도, 그리고 소스 코드의 특정 부분에 대해 산출물을 디버그하는 작업을 활성화시키는 데에도 유용합니다.

☑ *Use custom variables (restart required - include separators)* 를 활성화하면 변수들을 *Add* 및 *Remove* 할 수 있습니다. *Current environment variables* 에 이미 정의된 환경 변수를 표시하며, ☑ *Show only QGIS-specific variables* 를 활성화해서 환경 변수를 필터링할 수도 있습니다.

9.1.3 좌표계 설정

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

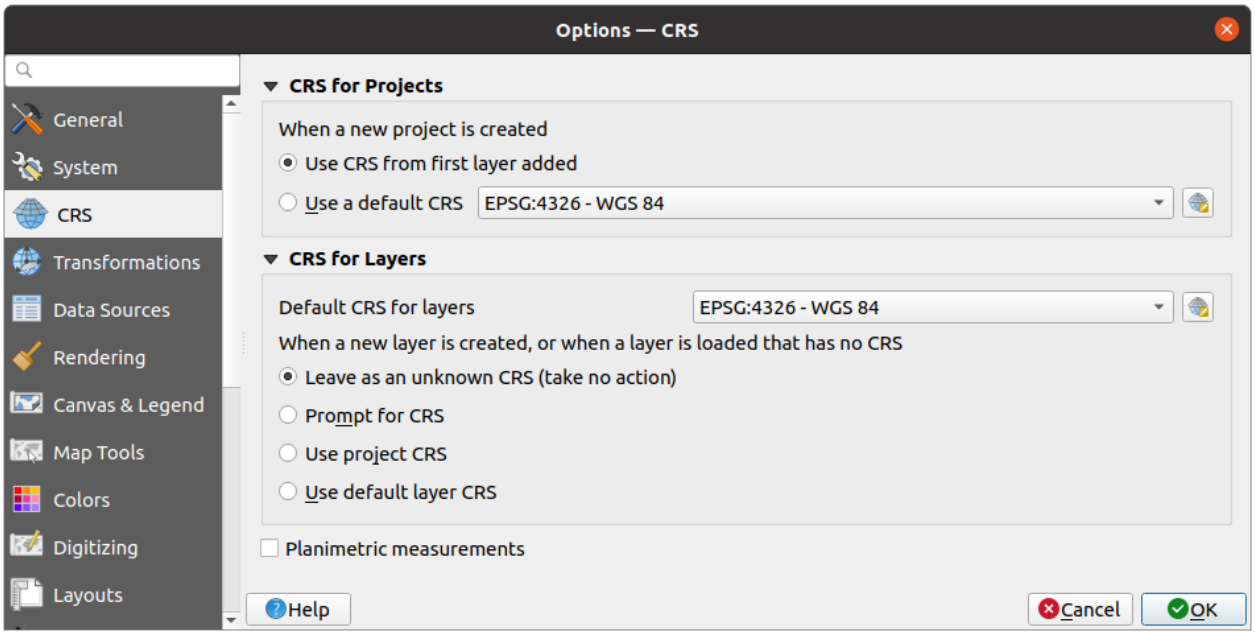


그림 9.2: QGIS 에서 좌표계 설정하기

새 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.

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

- *Leave as unknown CRS (take no action)*
- *Prompt for CRS*
- *Use project CRS*
- *Use a default CRS*

Planimetric measurements: 새로 생성한 프로젝트 용"평면 측정 (planimetric measurements)"속성 기본값을 설정합니다.

9.1.4 변환 설정

Transformations 탭에서, 프로젝트에 레이어를 불러오거나 레이어를 재투영할 때 적용할 좌표 변환 방법을 설정할 수 있습니다.

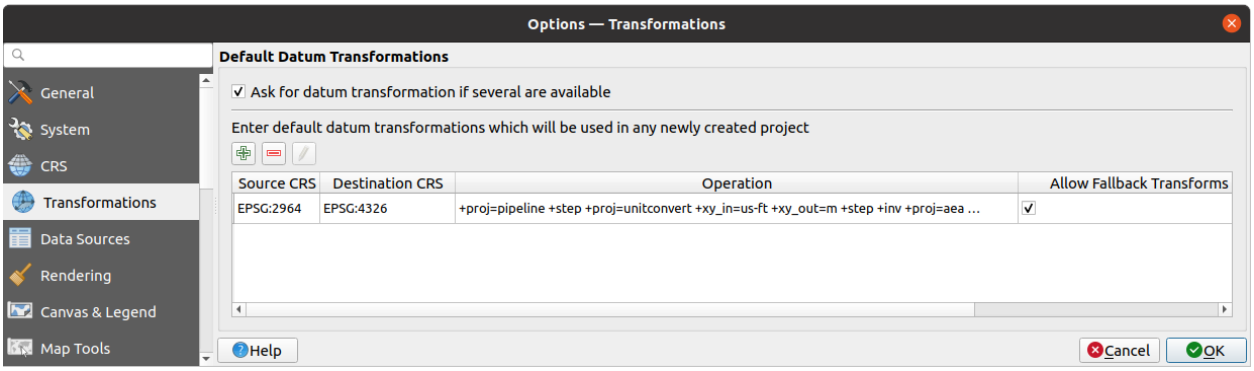


그림 9.3: 변환 설정

기본 기준 (datum) 변환

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

- QGIS 기본 변환 설정을 사용해서 자동적으로 공간 처리
- 그리고/또는 사용자 지정 선호도를 통해 다음과 같이 더 자세히 조정:
  - *Ask for datum transformation if several are available*: 여러 변환 방법을 사용할 수 있을 때 기준 변환 방법을 사용자가 설정할 수 있게 할지 여부를 설정.
  - 기본으로 적용할 미리 정의된 데이터 변환 목록. 자세한 내용은기준 *(datum)* 변환 을 참조하세요.

9.1.5 데이터소스 설정

피처 속성 및 테이블

- *Open new attribute tables as docked windows*
- 다른 응용 프로그램의 피처를 붙여넣을 경우 *Copy features as* 옵션으로'Plain text, no geometry', 'Plain text, WKT geometry', 또는'GeoJSON'가운데 하나를 선택합니다.
- *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.

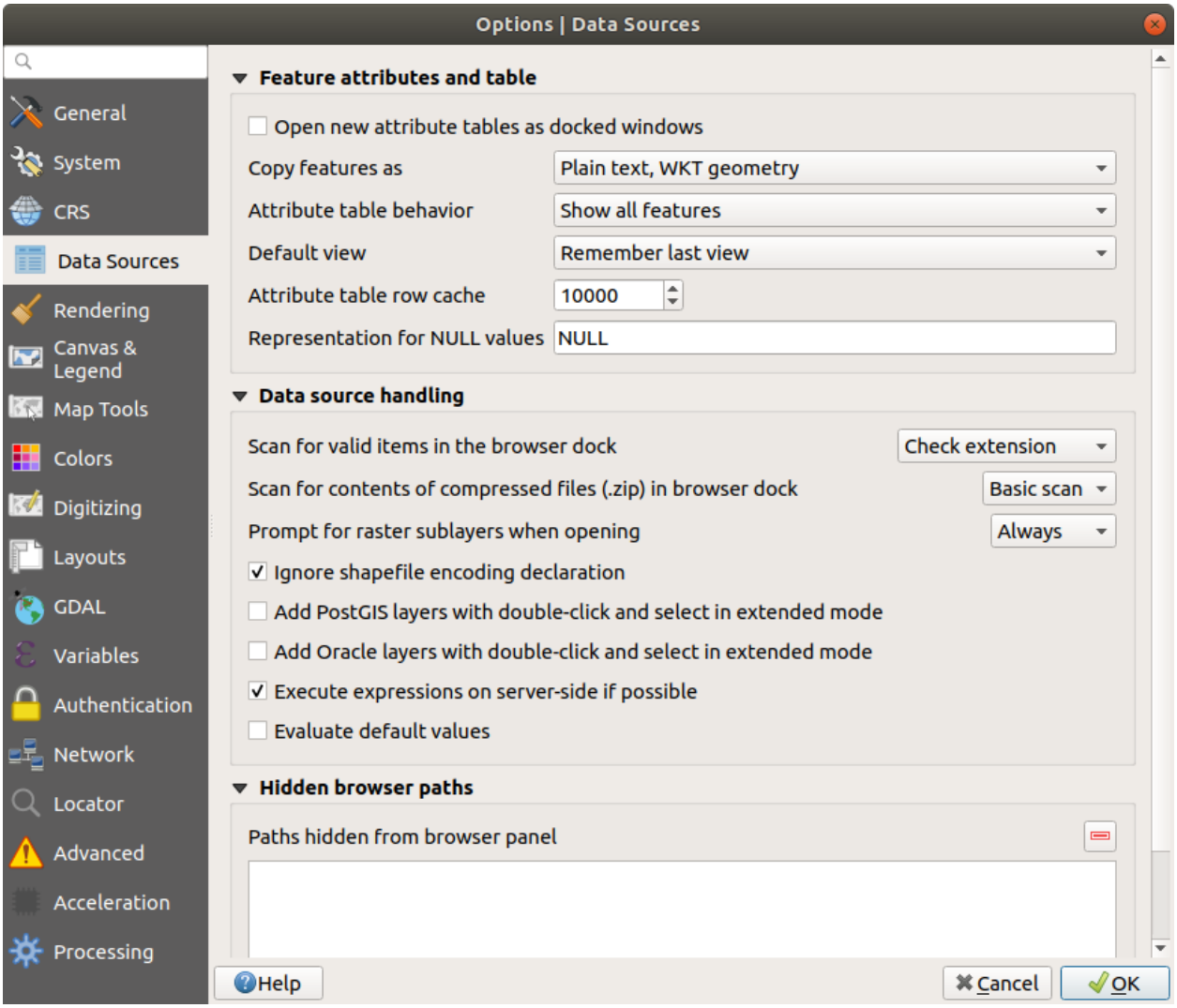


그림 9.4: QGIS 에서 데이터소스 설정하기

- *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 raster 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.
-  *Ignore shapefile encoding declaration* 을 체크하면 shapefile 에 인코딩 정보가 있더라도 QGIS 가 무시할 것입니다.
-  *Execute expressions on server-side if possible*: 데이터소스에서 피처를 요청할 경우, QGIS 는 서버에 직접 필터링 기준을 전송해서 기준에 맞는 피처만 다운로드하는 방법으로 요청을 최적화하려 할 것입니다. 예를 들어 사용자 인터페이스 상의 목록에 베른에 사는 농부들만 표시돼야 할 경우, QGIS 는 데이터베이스에 `WHERE "hometown" = 'Bern'` 을 전송할 것입니다. 어떤 경우에는 필터 기준이 너무 복잡해서 QGIS 표현식을 데이터베이스 호환 SQL 로 번역할 수 없을 수도 있습니다. 이런 경우 QGIS 는, 작업 효율은 많이 떨어지겠지만, 안전한 작업을 위해 전체 데이터를 다운로드해서 사용자 시스템 상에서 필터링할 것입니다.

  이 옵션을 비활성화하면 성능 저하를 감수하더라도 언제나 전체 데이터를 다운로드해서 사용자 시스템 상에서 필터링 작업을 하도록 QGIS 를 강제할 수 있습니다. 이 옵션은 안전차단기 (safety break) 라는 의미를 가지며, 사용자가 QGIS 표현식 번역 엔진의 오류를 식별한 경우에만 비활성화해야 합니다.

숨겨진 탐색기 경로

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

단축 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 번째로…넘어갈 것입니다.

9.1.6 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.
- ☑ *Use render caching where possible to speed up redraws*
- ☑ *Render layers in parallel using many CPU cores*
- ☑ *Max cores to use*
- *Map update interval (default to 250 ms)*
- ☑ *Enable feature simplification by default for newly added layers*
- *Simplification threshold*
- *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 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*: PostGIS, Oracle 등의 제공자 단에서 도형을 단순화하며, 로컬 단의 단순화와는 달리 도형 기반 계산에 영향을 줄 수도 있습니다.
- *Maximum scale at which the layer should be simplified*
- 1.2 *Magnification level* (확대경 참조)

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

렌더링 품질

- ☑ *Make lines appear less jagged at the expense of some drawing performance*

곡선 선분

- *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 사이에서 최대 각도 *(Maximum angle)* 또는 최대 거리 *(Maximum difference)* 가운데 선택할 수 있습니다.

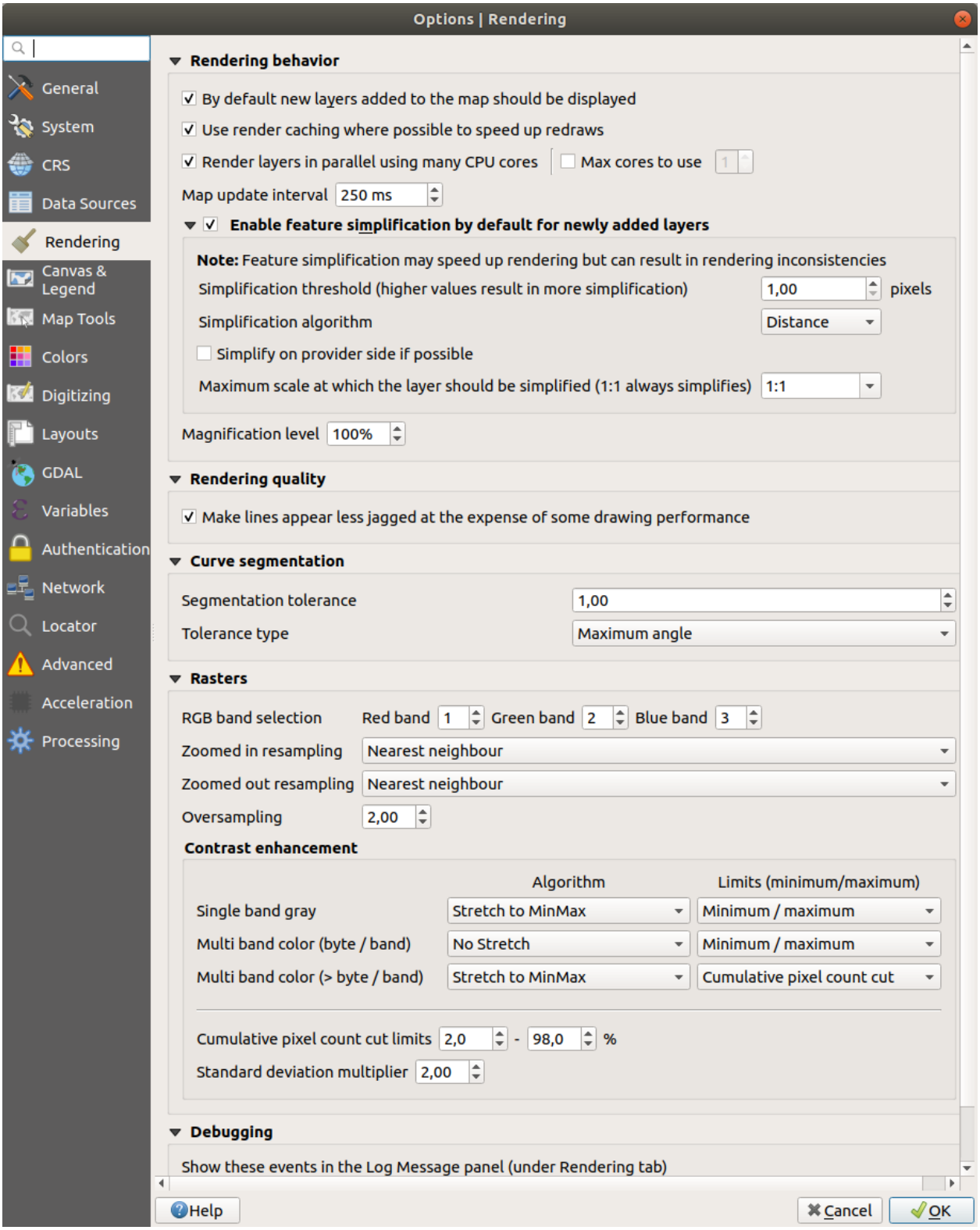


그림 9.5: 프로젝트 속성 대화창의 렌더링 탭

래스터

- *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'와'Cubic'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.)

대비 향상 (Contrast enhancement)

Single band gray, *Multi band color (byte/band)* 또는 *Multi band color (>byte/band)* 에 대비 향상 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.

래스터 렌더링의 경우, 다음 옵션도 정의할 수 있습니다:

- *Cumulative pixel count cut limits*
- *Standard deviation multiplier*

디버그

- *Map canvas refresh*: 렌더링에 걸리는 시간을 *Log Messages* 패널에서 디버그합니다.

9.1.7 캔버스 및 범례 설정

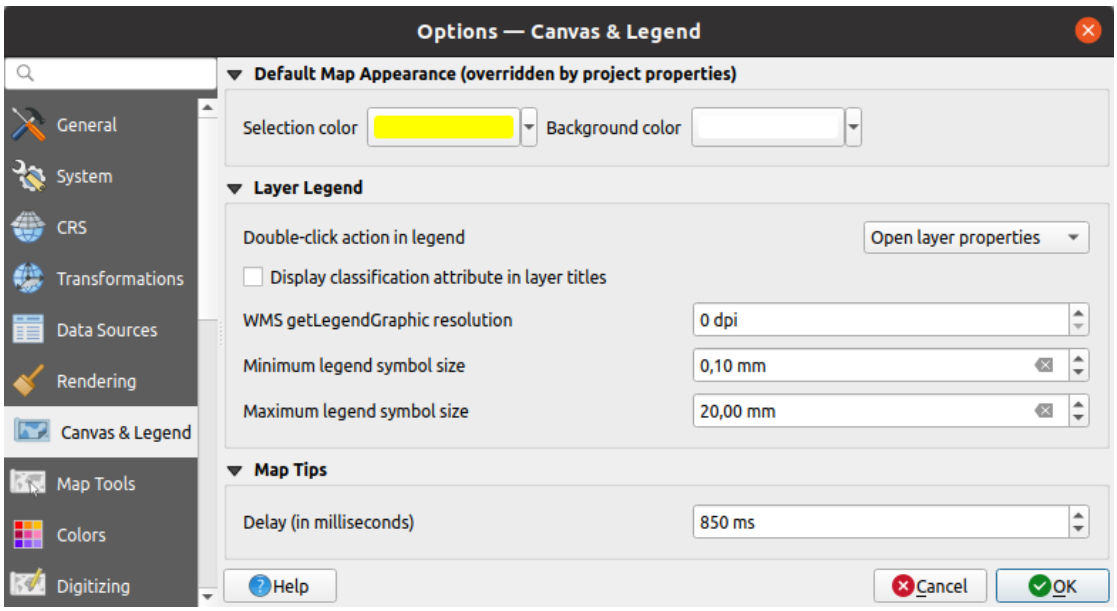


그림 9.6: 캔버스 및 범례 설정

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*  : 더블 클릭을 하면'Open layer properties'를 할지'Open attribute table'을 할지 또는'Open layer styling dock'을 할지 선택할 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.

9.1.8 맵 도구 설정

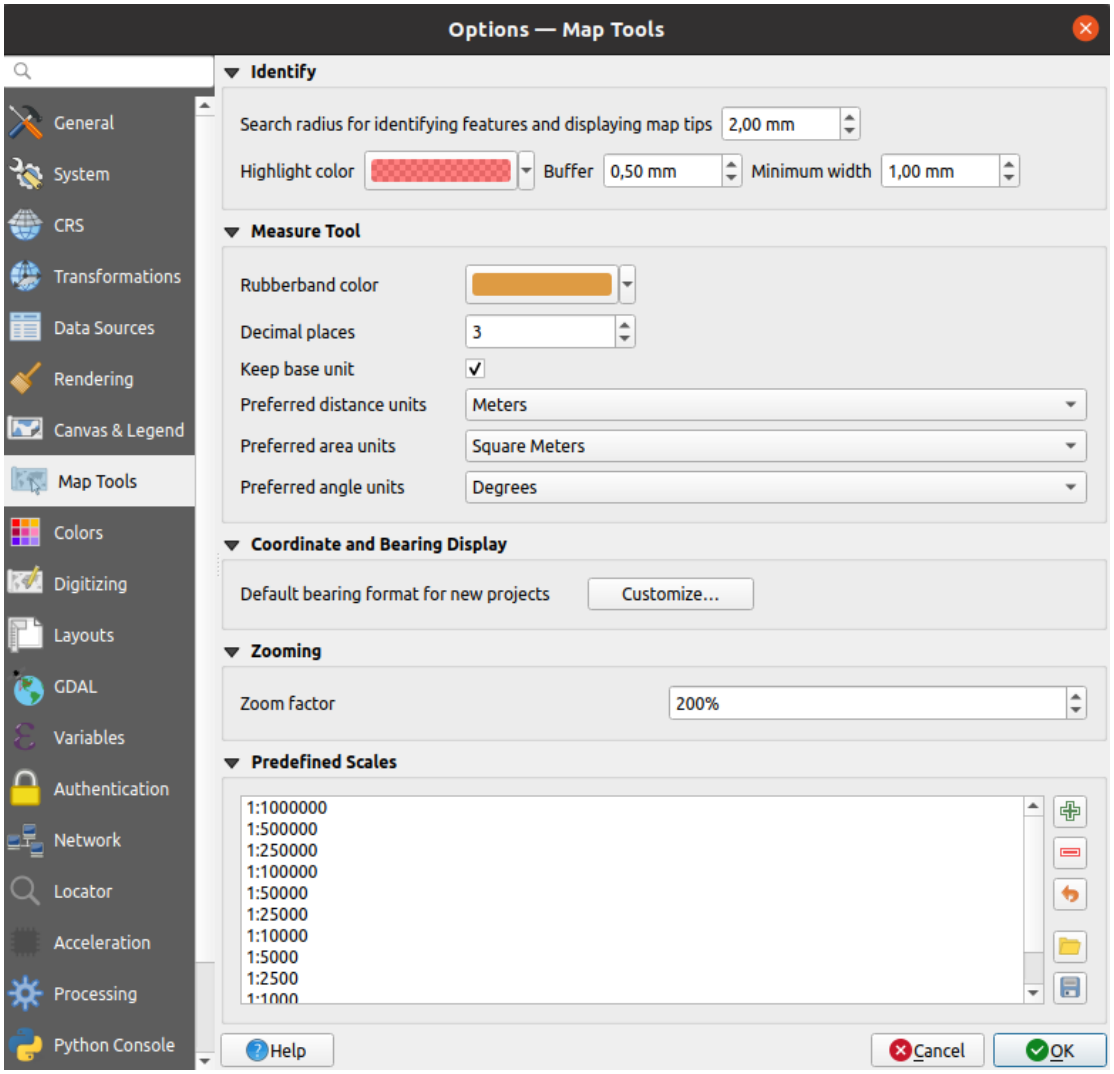


그림 9.7: QGIS 에서 맵 도구 설정하기

이 탭은식별 도구 의 습성과 관련된 여러 옵션을 제공합니다.

- *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.
- *Decimal places* 를 정의합니다.
- *Keep base unit* 을 체크하면 자동적으로 큰 숫자로 (예를 들어 미터를 킬로미터로) 변환하는 걸 막아줍니다.
- *Preferred distance units*: '미터', '킬로미터', '피트', '야드', '마일', '해리 (Nautical Miles)', '센티미터', '밀리미터', '도 (Degrees)'또는'맵 단위'가운데 하나를 선택합니다.
- *Preferred area units*: '제곱미터', '제곱킬로미터', '제곱피트', '제곱야드', '제곱마일', '헥타르', '에이커 (Acres)', '제곱해리', '제곱센티미터', '제곱밀리미터', '제곱도'또는'맵 단위'가운데 하나를 선택합니다.
- *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.

좌표 및 방향 표시

- *Default bearing format for new projects* 를 정의합니다: 맵 캔버스 위에서 마우스를 이동시킬 때 상태 바에 표시되는 마우스 좌표를 표시하는 데 사용되는 기본 좌표/방향 형식입니다. 프로젝트 속성 대화창에서 이 옵션을 무시할 수 있습니다.

이동 및 확대/축소

- 확대/축소 도구 또는 마우스 휠 용 *Zoom factor* 를 정의합니다.

사전 정의 축척

여기에서 사전 정의된 축척 목록을 볼 수 있습니다. 및 버튼으로 사용자 지정 축척을 추가하거나 제거할 수 있습니다. 또 축척을 `.XML` 파일 포맷으로 내보내거나 불러올 수 있습니다. 사용자 변경 사항을 제거하고 사전 정의 목록으로 초기화할 수 있다는 사실을 기억하십시오.

9.1.9 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 내용은기본 스타일 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.

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

- *Add* 또는 *Remove*: 색상을 추가하거나 삭제합니다.

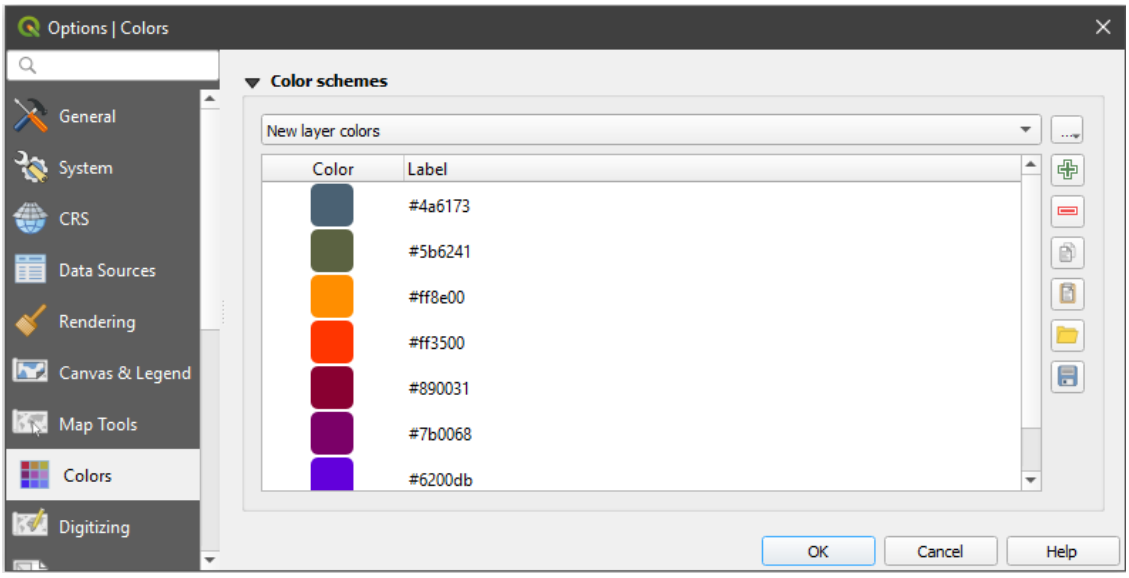


그림 9.8: 색상 설정

- *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.

9.1.10 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*: 모든 속성들의 마지막으로 사용된 값을 기억해서 다음 피처를 디지타이즈할 때 기본값으로 사용할지 여부를 설정합니다. 레이어 별로 작동합니다.
- *Validate geometries*. 많은 노드를 보유한 복잡한 라인 및 폴리곤을 편집하는 작업 시 렌더링이 아주 느려질 수 있습니다. QGIS 의 기본 유효성 확인 과정에 시간이 오래 걸릴 수 있기 때문입니다. GEOS 도형 유효성 확인을 (GEOS 3.3 버전부터 사용 가능) 활성화하거나, 도형 유효성 확인 과정을 비활성화해서 렌더링 속도를 높일 수 있습니다. GEOS 도형 유효성 확인은 훨씬 빠르지만, 첫 도형 문제점만 보고한다는 단점이 있습니다.

  선택 집합에 따라 도형 오류에 대한 보고도 달라질 수 있다는 점을 기억하십시오. (오류 메시지의 유형 및 그 의미 참조)
- *Default Z value*: 새 3D 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

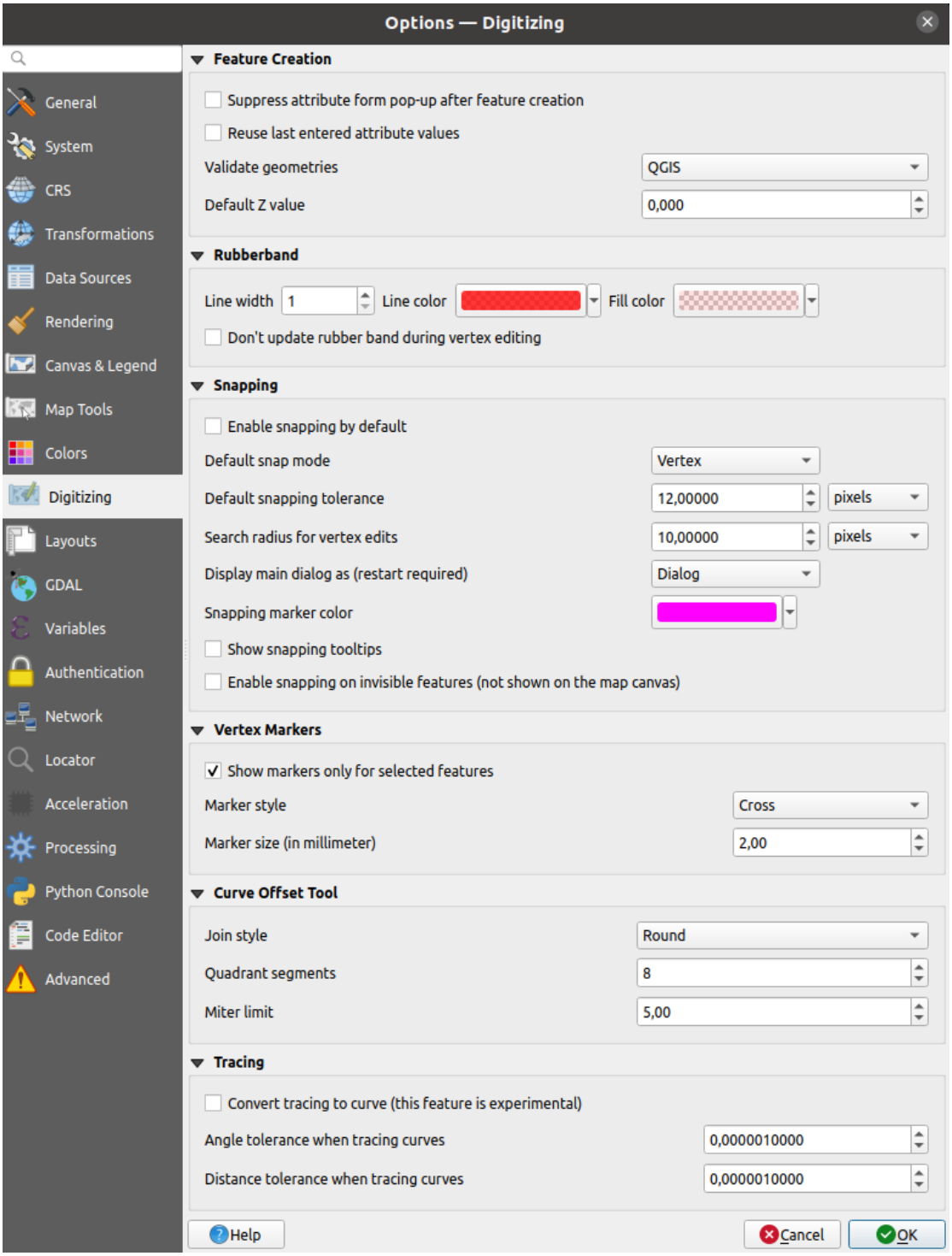


그림 9.9: QGIS 에서 디지타이즈 작업 설정하기

- *Enable snapping by default*: 프로젝트를 열 때부터 스냅 작업을 활성화합니다.
- *Default snap mode* 를 정의합니다. 'Vertex', 'Vertex and segment', 'Segment'가운데 하나를 선택할 수 있습니다.
- *Default snapping tolerance* 를 정의합니다. 맵 단위 또는 픽셀 개수로 설정할 수 있습니다.
- *Search radius for vertex edits* 을 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 세 가지 옵션은고급 디지타이즈 작업 의 Offset Curve 에서 쓰입니다. 다양한 설정을 통해, 라인 오프셋의 형태를 변경할 수 있습니다. GEOS 3.3 버전부터 이 옵션들을 사용할 수 있습니다.

- *Join style* 로'Round', 'Mitre'또는'Bevel'가운데 하나를 선택할 수 있습니다.
- *Quadrant segments*
- *Miter limit*

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.

9.1.11 조판기 설정

조판 기본값

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

그리드 모양

- *Grid style* 에서'Solid', 'Dots', 'Crosses'가운데 하나를 선택합니다.
- *Grid color* 를 정의합니다.

그리드 및 가이드 기본값

- *Grid spacing* 1,00 을 정의합니다.
- *Grid offset* 1,00 : X 축 및 Y 축에 대한 그리드 오프셋을 정의합니다.

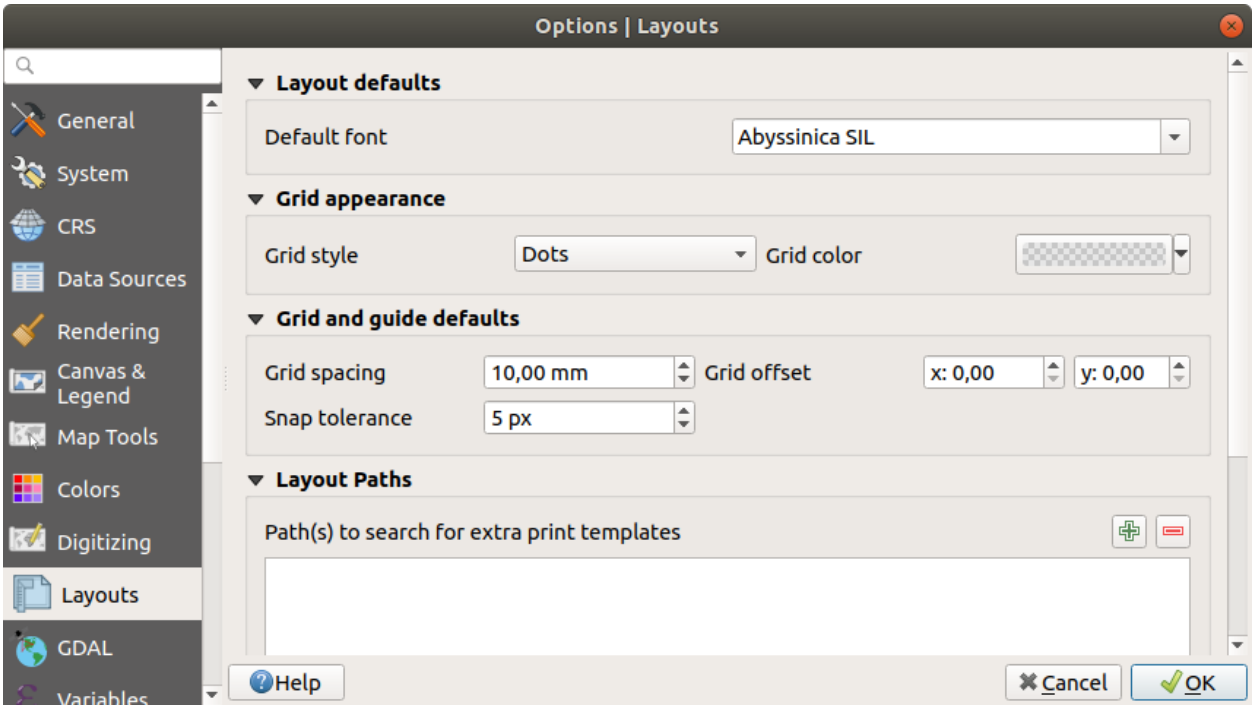


그림 9.10: QGIS 에서 조판기 설정하기

- *Snap tolerance* 1,00 를 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.

9.1.12 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.

래스터 드라이버 옵션

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

- *Edit create options*: 사용자가 래스터 파일을 산출할 때 사용할 서로 다른 파일 변환 프로파일 — 예를 들면 미리 정의된 파라미터들의 조합 (압축 유형 및 수준, 블록 크기, 오버뷰, 색채분석 (colorimetry), 투명도 등등) — 을 편집하거나 추가할 수 있습니다. 정의할 수 있는 파라미터는 드라이버에 따라 달라집니다.

  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:

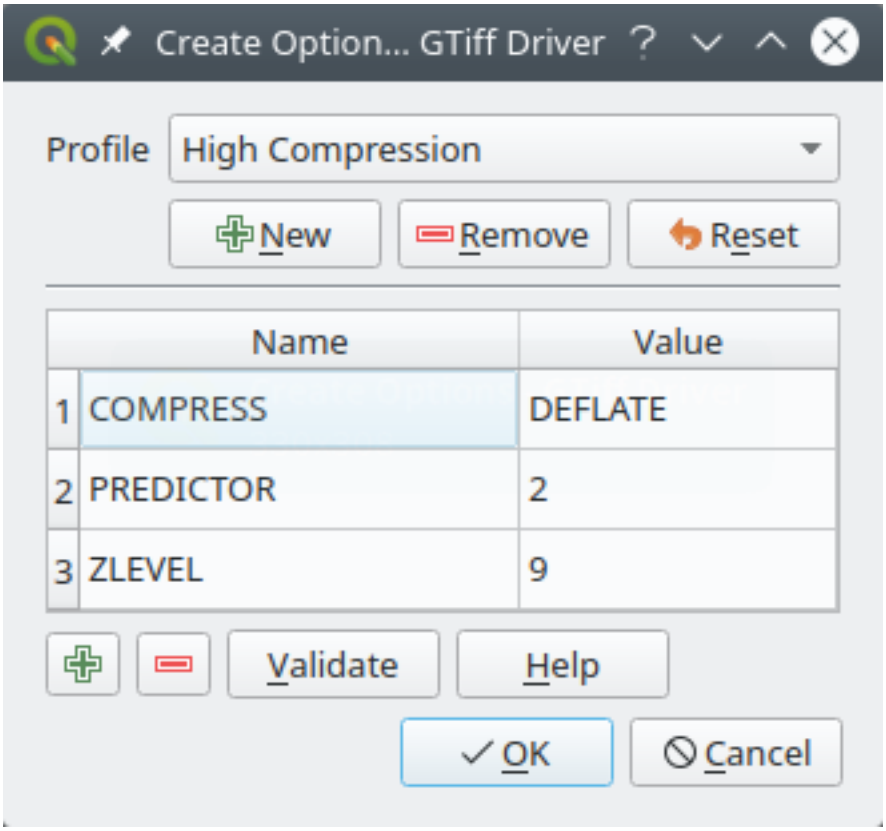


그림 9.11: 옵션 프로파일 생성 (GeoTiff 용) 예시

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

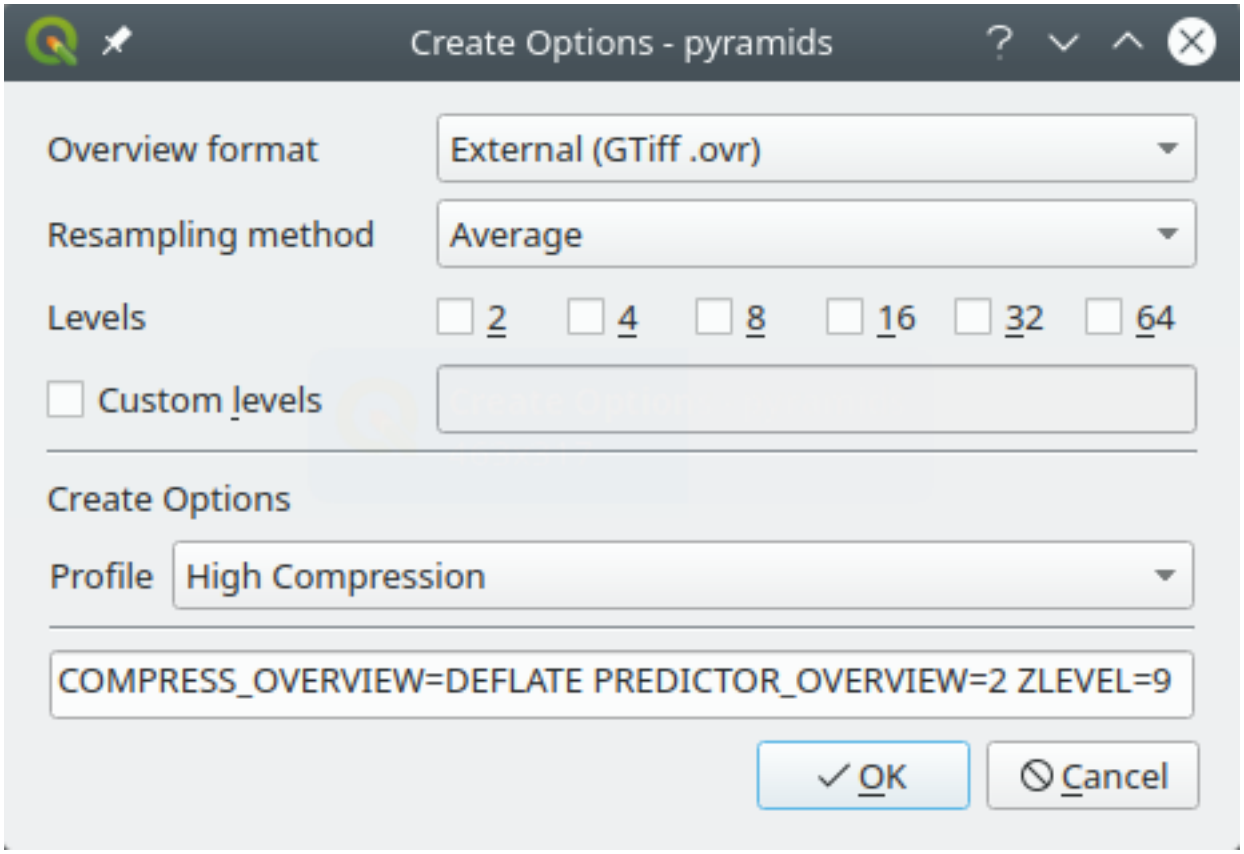


그림 9.12: 피라미드 프로파일 예시

GDAL 래스터 및 벡터 드라이버

Raster Drivers 및 *Vector Drivers* (개별) 탭에서, 파일을 읽기 그리고/또는 쓰기 위해 어떤 GDAL 드라이버를 활성화할지를 정의할 수 있습니다. 몇몇 경우 GDAL 드라이버 하나 이상을 사용할 수 있기 때문입니다.

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

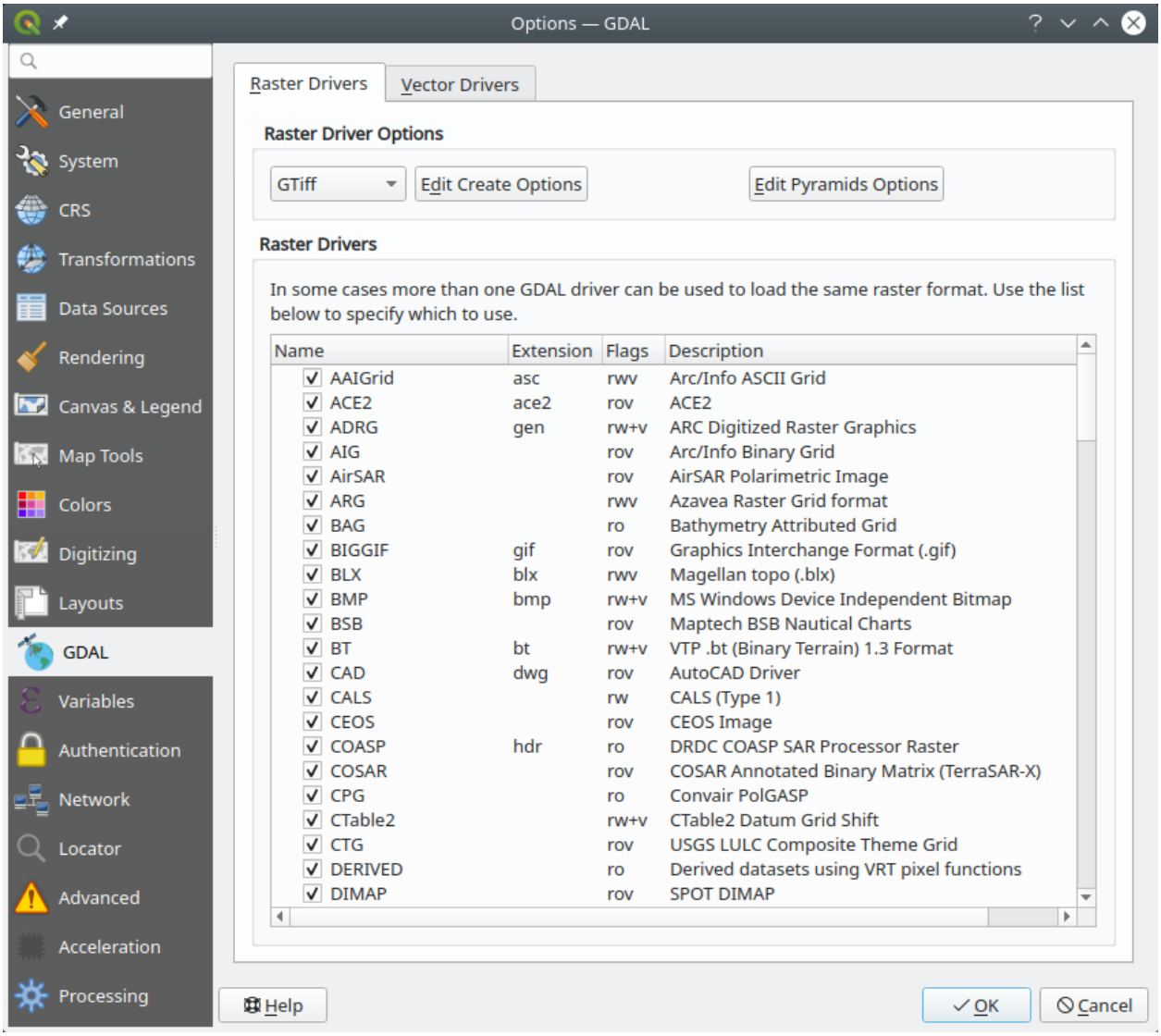


그림 9.13: QGIS 에서 GDAL 설정하기 - 래스터 드라이버

9.1.13 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.

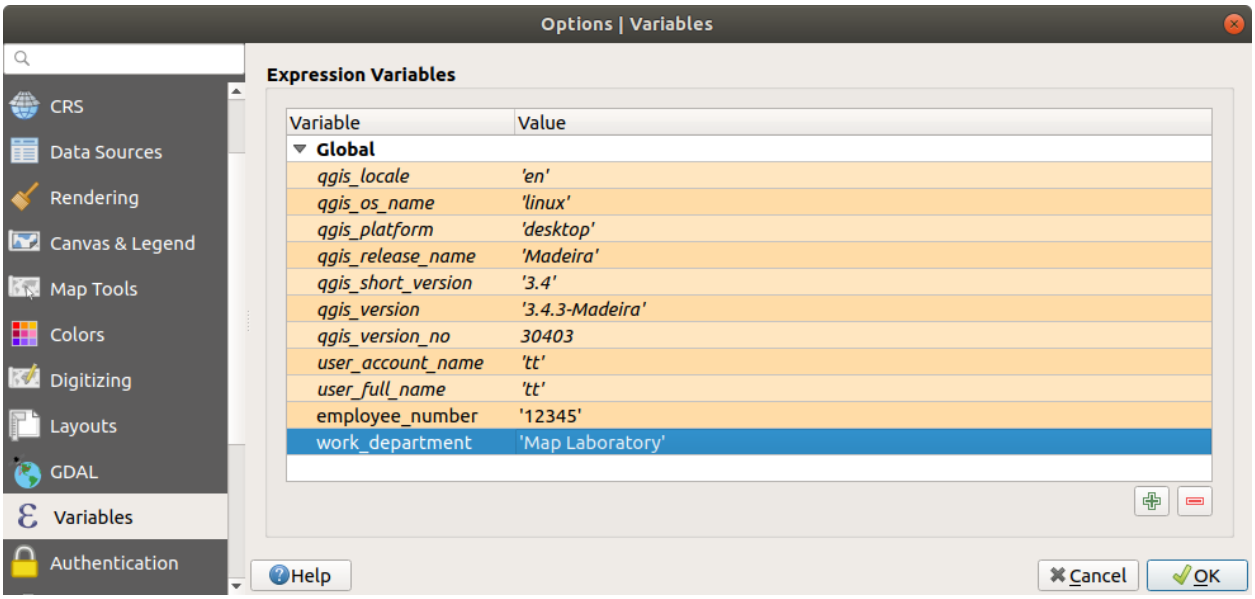


그림 9.14: QGIS 에서 변수 설정하기

9.1.14 인증 설정

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

9.1.15 네트워크 설정

일반

- *Timeout for network requests (ms)* 를 정의합니다. 기본값은 60,000 입니다.
- *Default expiration period for WMS Capabilities (hours)* 를 정의합니다. 기본값은 24 입니다.
- *Default expiration period for WMS-C/WMTS tiles (hours)* 를 정의합니다. 기본값은 24 입니다.
- *Max retry in case of tile or feature request errors* 를 정의합니다.
- *User-Agent* 를 정의합니다.

캐시 설정

캐시 용 *Directory* 와 *Size* 를 정의합니다. 또 *automatically clear the connection authentication cache on SSL errors (recommended)* 옵션은 SLL 오류 시 연결 인증 캐시를 자동으로 삭제하는 도구를 제공합니다.

웹 접근 용 프록시

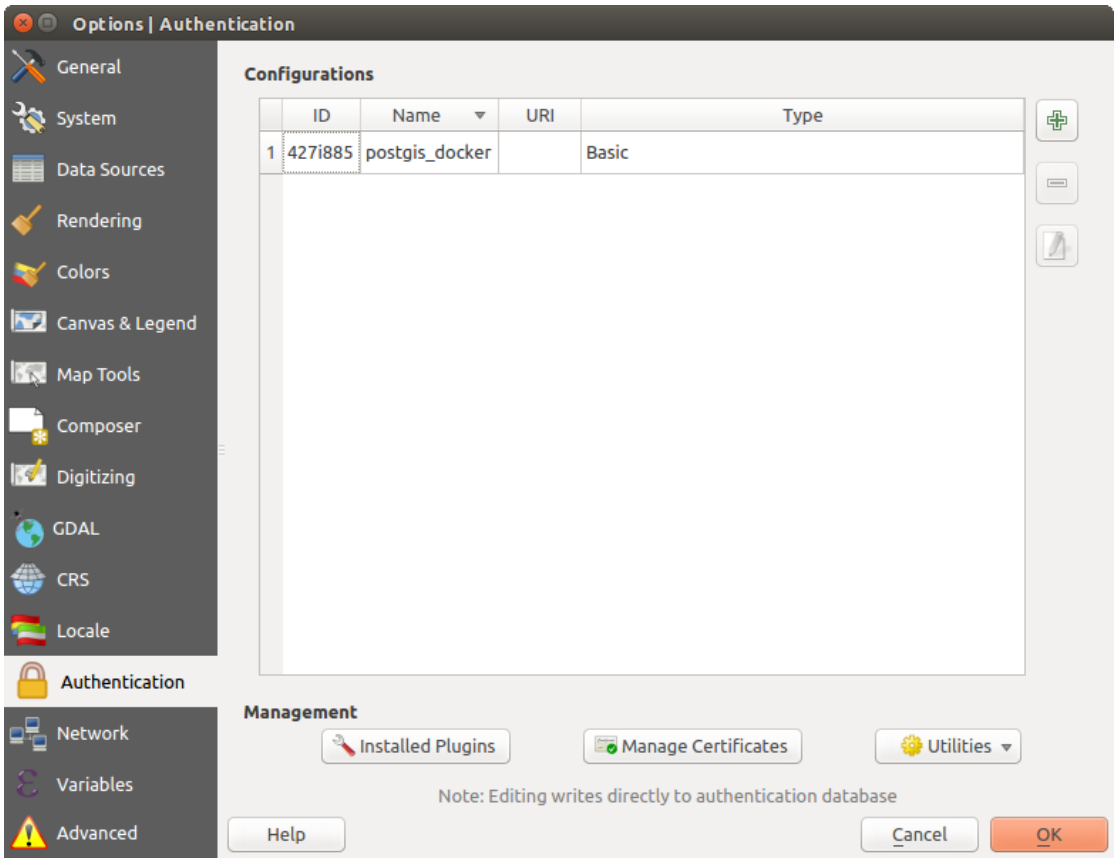


그림 9.15: QGIS 에서 인증 설정하기

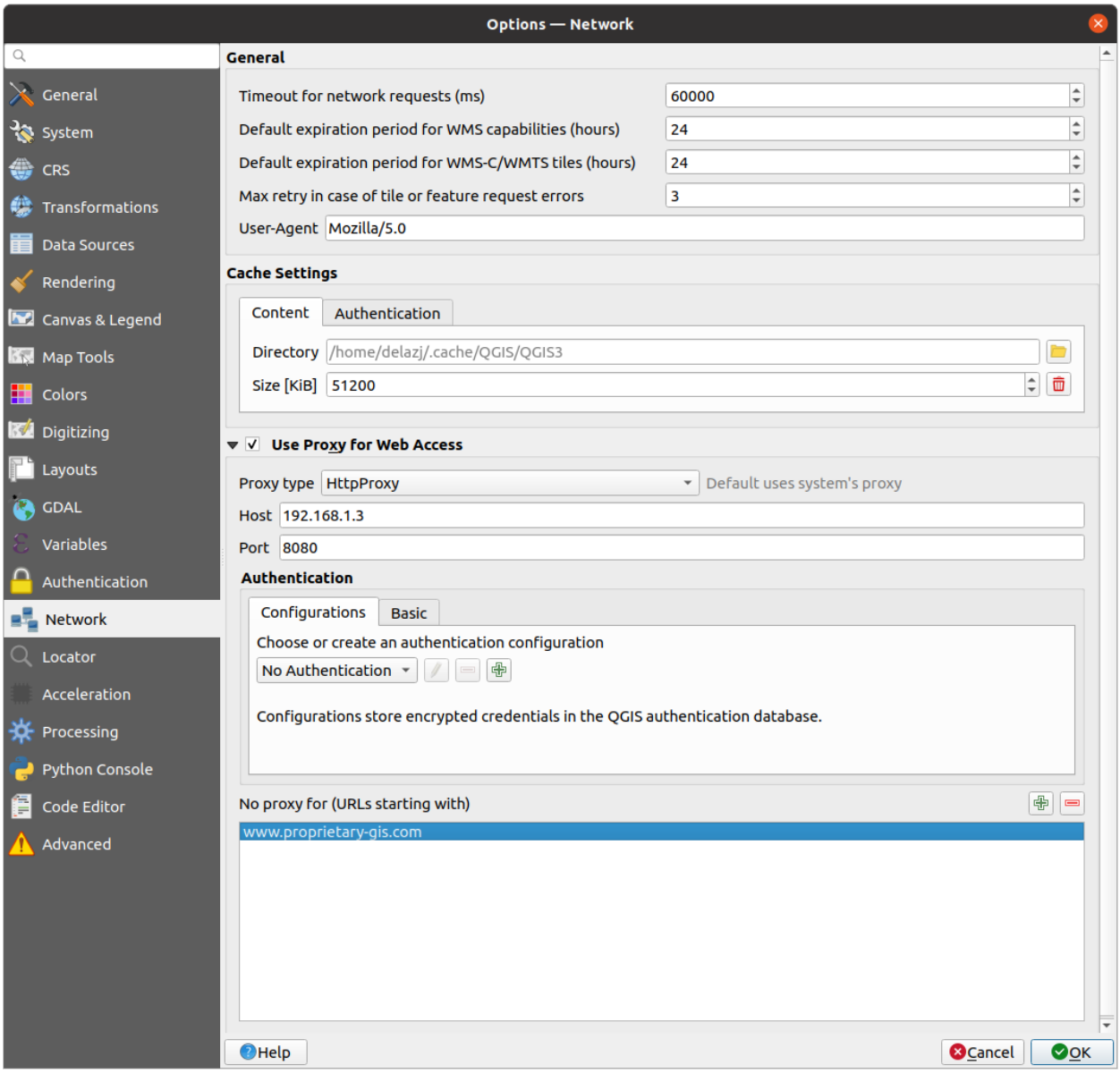


그림 9.16: QGIS 에서 프록시 설정하기

- *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.
  - *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 9.16 그림 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/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.

9.1.16 위치 탐색 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:

- 프로젝트 레이어 (`l`): *Layers* 패널에 있는 레이어를 찾아서 선택합니다.
- 프로젝트 조판기 (`pl`): 인쇄 조판을 찾아서 엽니다.
- 액션 (`.`): QGIS 액션을 찾아서 실행합니다. 액션이란 QGIS 에 있는 어떤 도구나 메뉴도 될 수 있고, 패널을 열 수도 있고, …
- 활성 레이어 피처 (`f`): 현재 활성화된 레이어의 어느 항목에서건 일치하는 속성을 검색해서 선택된 피처로 확대/축소합니다. 검색 결과의 최대 개수를 환경설정하려면 아이콘을 클릭하십시오.
- 모든 레이어의 피처 (`af`): 각검색 가능한 레이어 의표시 명칭 에서 일치하는 속성을 검색해서 선택된 피처로 확대/축소합니다. 검색 결과의 최대 개수 및 레이어별 검색 결과의 최대 개수를 환경설정하려면 아이콘을 클릭하십시오.
- 계산기 (`=`): 모든 QGIS 표현식을 검증할 수 있고, 표현식이 유효한 경우 결과물을 클립보드로 복사할 수 있습니다.
- 공간 북마크 (`b`): 북마크를 찾아서 북마크의 범위로 확대/축소합니다.
- 설정 (`set`): 프로젝트를 탐색해서 연 다음, 응용 프로그램 수준의 속성 대화창을 엽니다.
- 좌표로 가기 (`go`): 쉼표 또는 공백으로 분리된 x 및 y 좌표 쌍 또는 서식화된 URL(예: 오픈스트리트맵, Leaflet, 오픈레이어스, 구글 지도 등등) 이 정의하는 위치로 맵 캔버스를 이동시킵니다. 좌표는 WGS 84 (`epsg:4326`) 그리고/또는 맵 캔버스 좌표계여야 합니다.

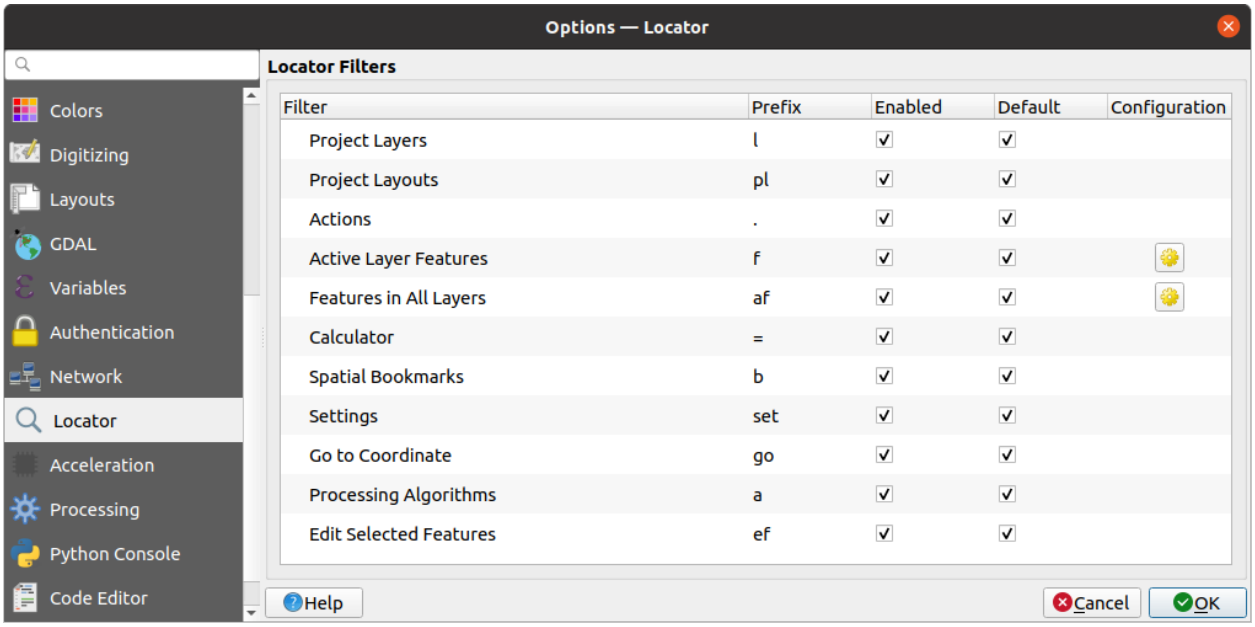


그림 9.17: QGIS 에서 위치 탐색 설정하기

- 공간 처리 알고리즘 (`a`): 공간 처리 알고리즘을 검색해서 공간 처리 알고리즘 대화창을 엽니다.
- 선택한 피처 편집 (`ef`): 빠른 접근을 허용해서 활성 레이어에 호환되는제자리 수정 공간 처리 알고리즘을 실행합니다.

이 대화창에서 다음과 같은 작업을 할 수 있습니다:

- 필터 *Prefix* 를 사용자 정의할 수 있습니다. 예를 들어 필터를 촉발하기 위해 사용할 키워드를 정의할 수 있습니다.
- 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 Nominatim 검색, 데이터베이스 직접 검색, 레이어 카탈로그 검색 플러그인 등등을 사용할 수 있습니다.

9.1.17 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.

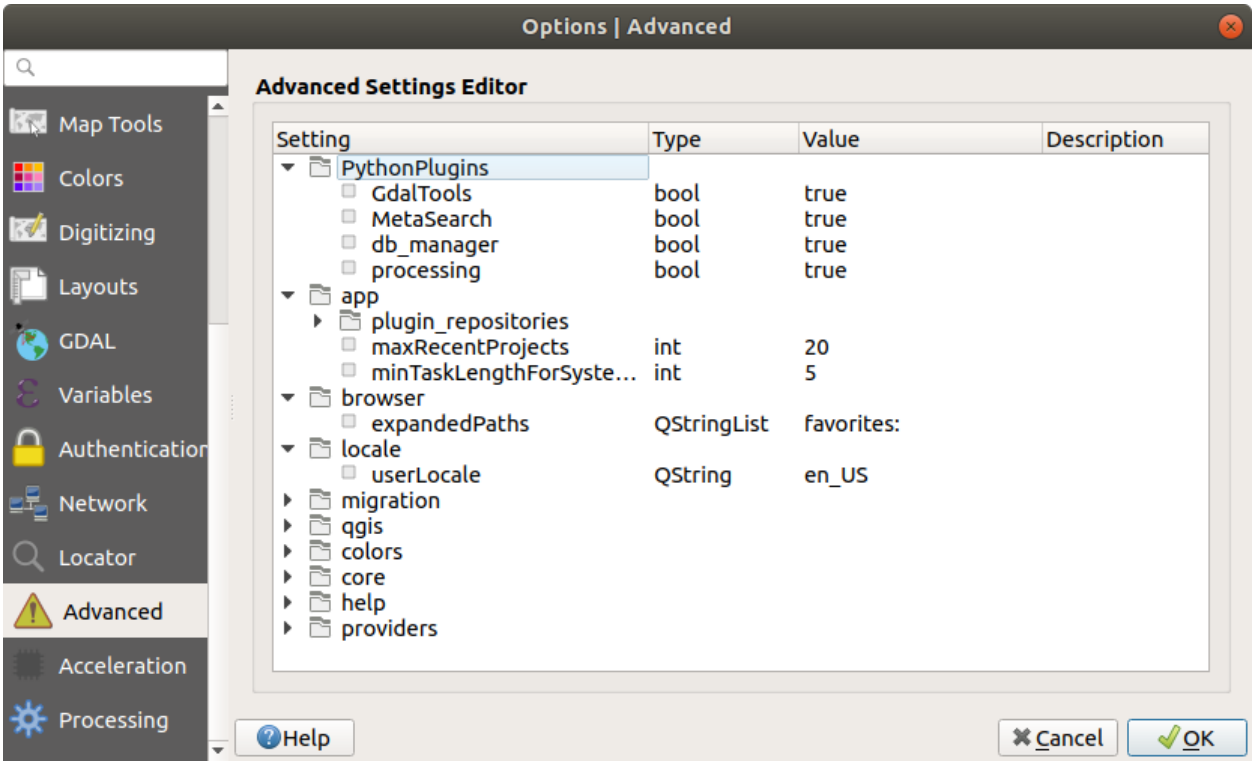


그림 9.18: QGIS 의 고급 설정 탭

경고: 고급 설정 탭을 무턱대고 이용하지 말 것

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.

9.1.18 가속 설정

OpenCL 가속을 설정합니다.

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

9.1.19 공간 처리 설정

Processing 탭은 QGIS 공간 처리 프레임워크가 사용하는 도구와 데이터 제공자의 일반 설정을 제공합니다. 자세한 내용은*QGIS* 공간 처리 프레임워크 를 참조하세요.

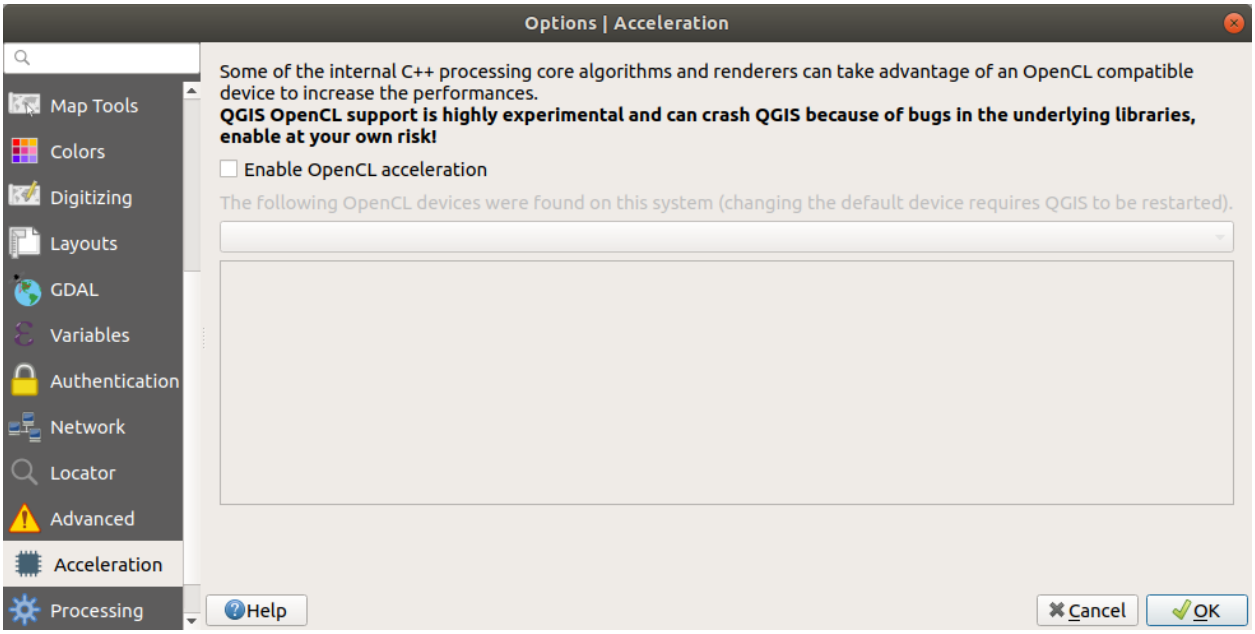


그림 9.19: 가속 탭

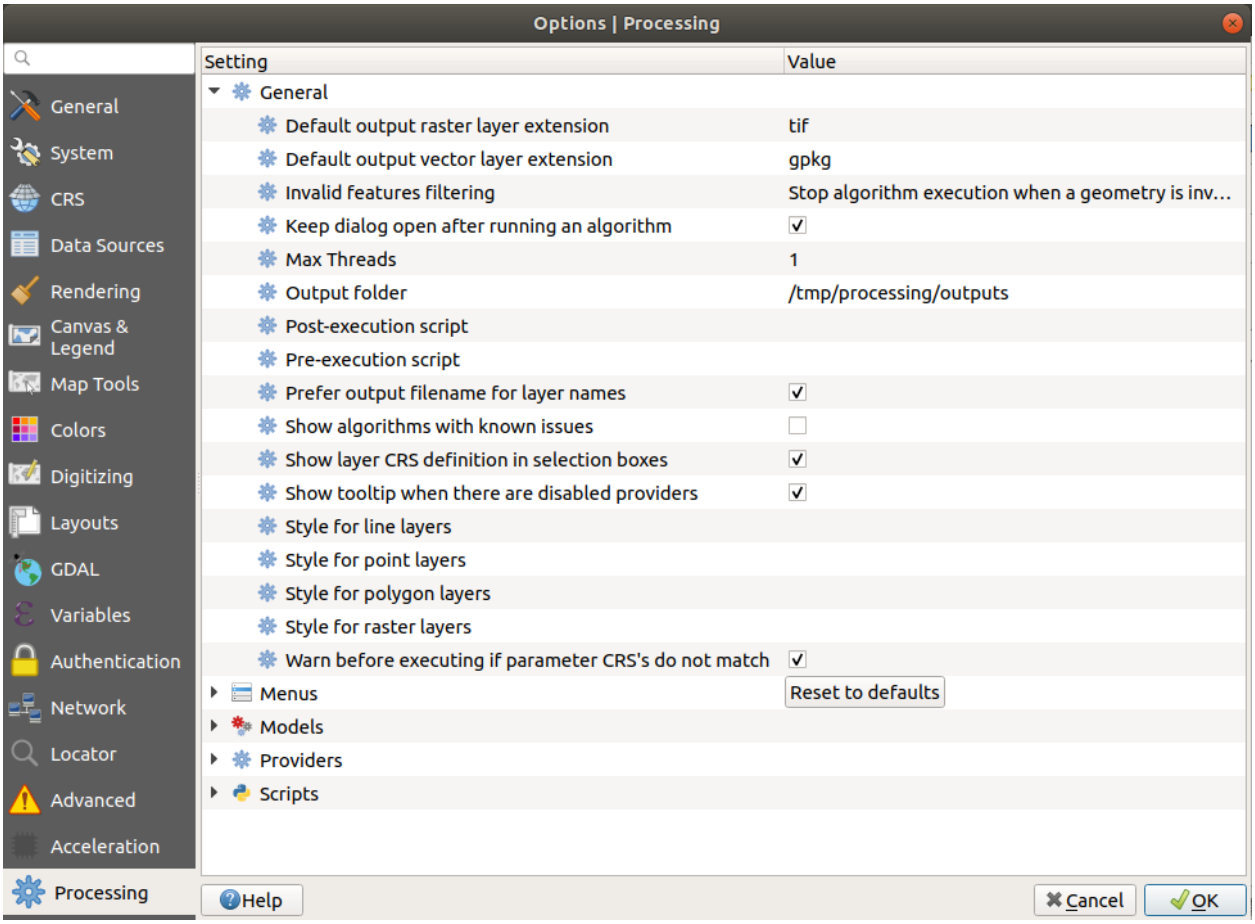


그림 9.20: QGIS 의 공간 처리 설정 탭

9.1.20 파이썬 콘솔 설정

Python Console 설정을 통해 파이썬 편집기 (대화형 콘솔, 코드 편집기, 프로젝트 매크로, 사용자 지정 표현식 등등) 의 습성을 관리하고 제어할 수 있습니다. 다음에 있는 Options··· 아이콘을 클릭해도 파이썬 콘솔 설정에 접근할 수 있습니다:

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

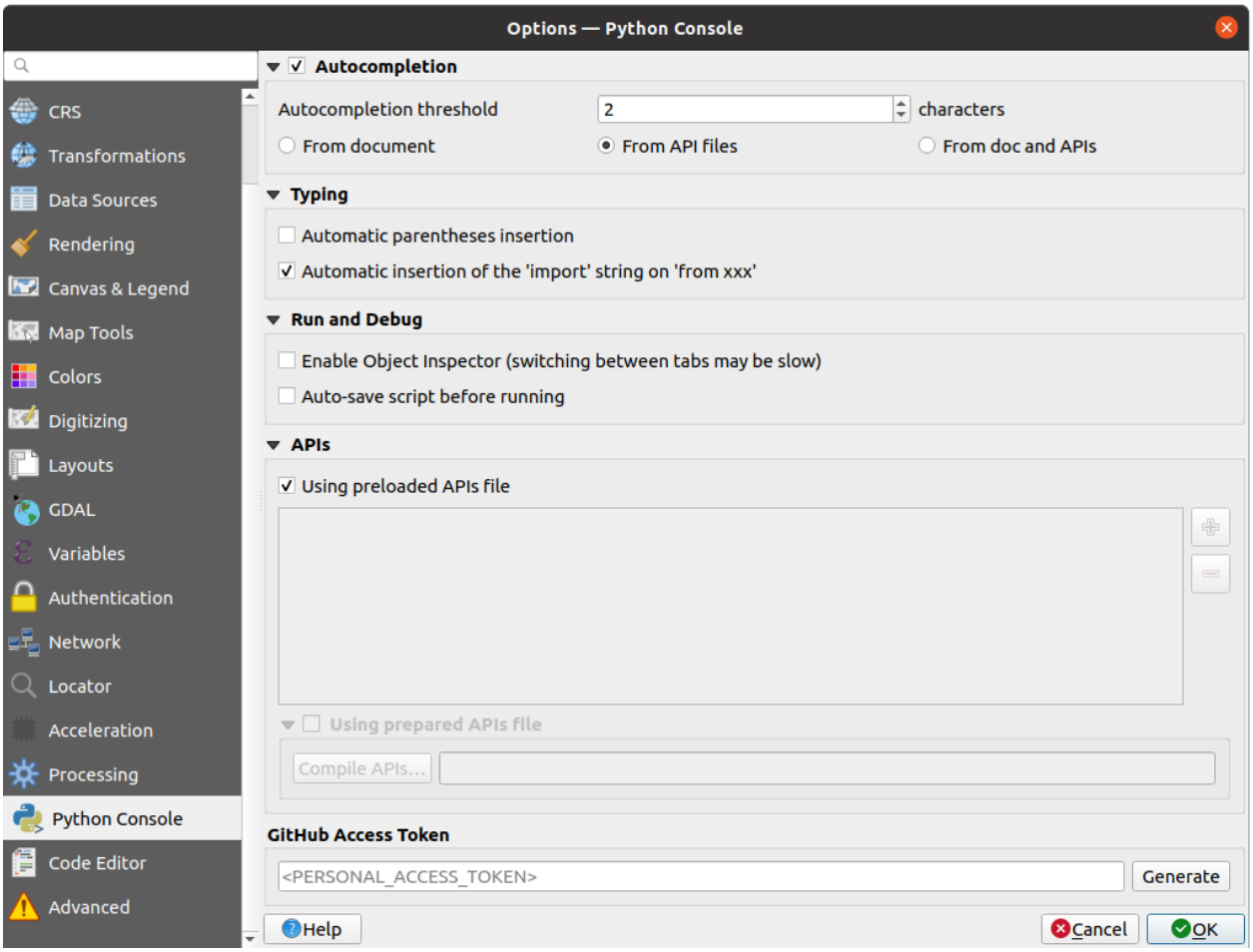


그림 9.21: 파이썬 콘솔 설정 탭

사용자는 다음을 지정할 수 있습니다:

- *Autocompletion*: 이 옵션을 체크하면 코드 자동완성이 활성화됩니다. 현재 문서, 설치된 API 파일 또는 둘 다에서 자동완성을 가져올 수 있습니다.
  - *Autocompletion threshold*: 자동완성 목록을 표시할 한계값 (입력한 문자 개수) 을 설정합니다.
- *Typing* 아래 있는
  - *Automatic parentheses insertion*: 이 옵션을 활성화하면 괄호를 자동으로 닫습니다.

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

- *Run and Debug* 아래 있는
  - *Enable Object Inspector (switching between tabs may be slow)*: 객체 점검기를 활성화합니다.
  - *Auto-save script before running*: 실행 시 스크립트를 자동으로 저장하는 옵션입니다. 시스템의 임시 디렉터리에 임시 파일을 저장하고, 실행 후에 자동적으로 삭제할 것입니다.

APIs 에서는 다음을 지정할 수 있습니다:

- *Using preloaded APIs file*: 미리 불러온 API 파일을 사용할지 선택할 수 있습니다. 이 옵션을 활성화하지 않는 경우 API 파일을 추가할 수 있고, 또 준비된 API 파일을 사용할지도 선택할 수 있습니다. (다음 옵션 참조)
- *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.
- *GitHub access token* 란에서 파이썬 코드 편집기 내부에서부터 코드 조각들을 공유할 수 있게 해주는 개인 토큰 (personal token) 을 생성할 수 있습니다. 더 자세한 내용은 GitHub 인증 을 참조하세요.

9.1.21 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 *Font* 글꼴 집합과 *Size* 글꼴 크기를 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, 미리 정의된 스키마를 선택하면 되돌릴 수 있습니다.
  - 주석, 인용, 함수, 배경 등등에 사용할 색상과 같은 코드 작성의 각 요소의색상 을 변경할 수 있습니다.

9.2 사용자 프로파일 작업

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

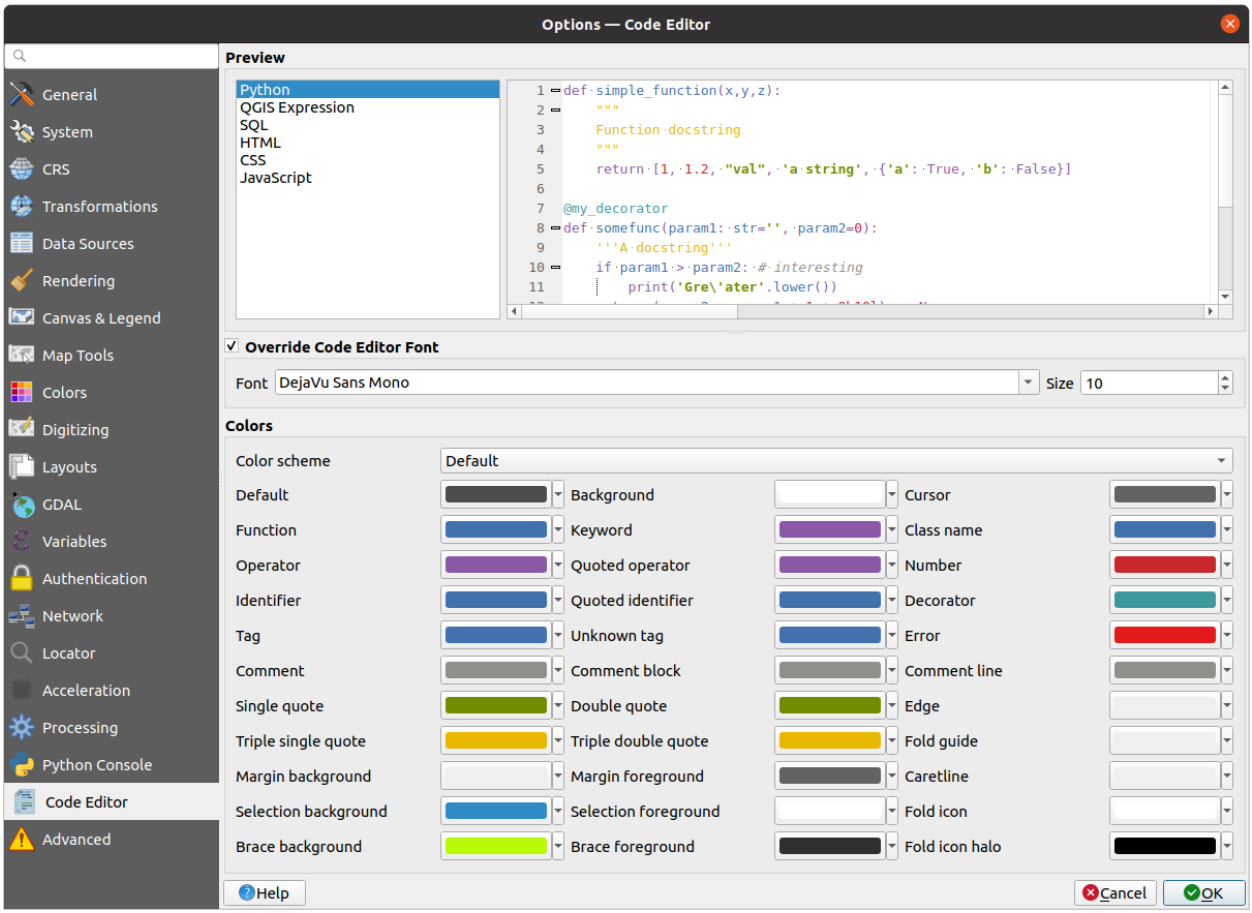


그림 9.22: 코드 편집기 설정 탭

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

1. *New profile*…항목을 클릭하십시오.
2. 프로파일 명칭을 입력해야 하는데, 명칭을 입력하면 `~/<UserProfiles>/` 아래 동일 명칭의 폴더를 생성합니다. 이때:
   - `~` 는 **HOME** 디렉터리를 나타냅니다. 윈도우에서는 보통 `C:\Users\(user)` 같은 디렉터리입니다.
   - `<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.

일부러 변경하지 않는 한, 마지막으로 닫은 QGIS 세션의 사용자 프로파일을 다음 QGIS 세션에서 사용할 것입니다.

팁: 버그 지속성 검사를 위해 새 사용자 프로파일로 QGIS 실행하기

QGIS 의 어떤 기능에서 이상한 습성을 발견한 경우, 새 사용자 프로파일을 생성한 다음 해당 명령을 다시 실행해보십시오. 버그는 때때로 현재 사용자 프로파일에 있는 어떤 잔존물 때문에 발생하기도 합니다. 사용자 프로파일을 새로 생성하면 새 (초기화된) 프로파일로 QGIS 를 재시작하기 때문에 버그가 사라지는 경우도 있습니다.

9.3 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).

9.3.1 일반 속성

General 탭에 있는 *General settings* 부분에서:

- 프로젝트 파일의 위치를 볼 수 있고
- 프로젝트 홈 용 폴더를 (탐색기에 있는 `Project Home` 항목에서) 설정할 수 있습니다. 프로젝트 파일이 있는 폴더에 대한 상대 경로일 수도 있고 (직접 입력) 또는 절대 경로일 수도 있습니다. 프로젝트 홈은 프로젝트에 유용한 데이터와 기타 콘텐츠를 저장하는 데 쓰일 수 있습니다.
- 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 위치에 상대적인 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.

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:

- *Ellipsoid*: 거리 및 면적 계산이 완전히 의존하는 타원체입니다. 다음 가운데 하나를 선택할 수 있습니다:
  - **None/Planimetric**: 이 타원체를 선택하면 데카르트 도량법으로 된 값을 반환합니다.
  - **Custom**: 이 타원체를 선택하면 긴반지름 (semi-major) 과 짧은반지름 (semi-minor) 의 값을 설정해야 합니다.
  - 또는 미리 정의된 목록에서 기존 타원체 가운데 하나를 선택할 수도 있습니다. (Clarke 1866, Clarke 1880 IGN, New International 1967, WGS 84, ⋯)
- 길이 및 둘레에 대한 *units for distance measurements* 및 면적에 대한 *units for area measurements* 를 설정하면 현재 프로젝트에서 QGIS 옵션에 설정된 기본 단위를 무시하게 만들며, 다음에 쓰입니다:
  - 속성 테이블 필드 업데이트 바
  - 필드 계산기 계산
  - 식별 도구에서 파생되는 길이, 둘레 및 면적 값
  - 측정 대화창에 표시되는 기본 단위

Coordinate and Bearing display 에서 상태 바에 표시되는 마우스 좌표 및 식별 도구에 표시되는 파생 좌표에 쓰이는 좌표/방향 서식 및 단위의 유형을 선택하고 사용자 지정할 수 있습니다.

마지막으로, 전체 수준에서 사전 정의된 축척을 무시하는 *Project predefined scales* 목록을 설정할 수 있습니다.

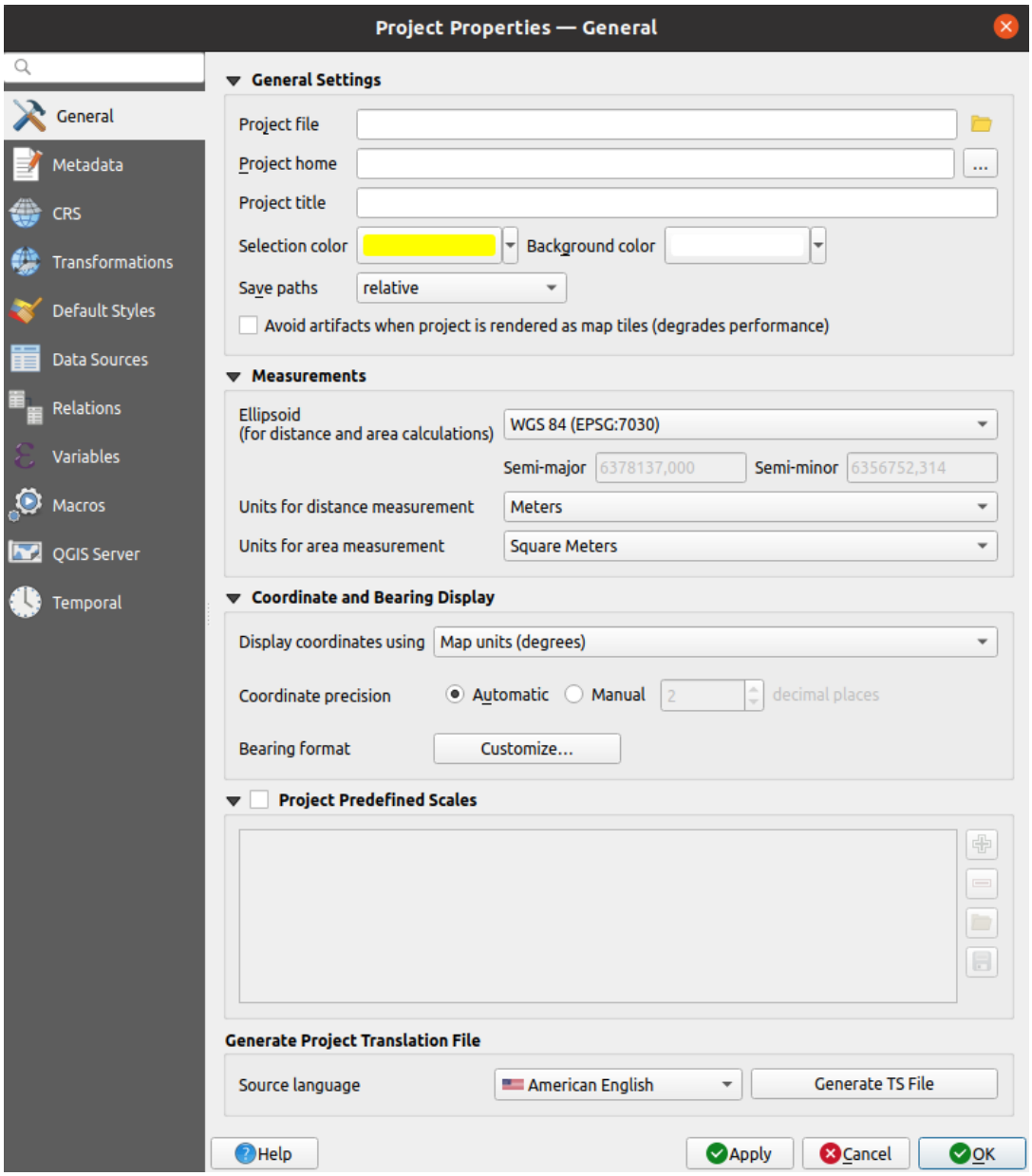


그림 9.23: 프로젝트 속성 대화창의 일반 탭

9.3.2 메타데이터 속성

Metadata 탭에서 다음과 같은 자세한 메타데이터를 (이외에도 있지만) 정의할 수 있습니다: 작성자 (author), 생성일, 언어, 개요 (abstract), 카테고리, 키워드, 연락처 정보, 링크, 이력. 특정 항목이 입력됐는지 확인하는 유효성 확인 기능도 있지만, 강제 사항은 아닙니다. 더 자세한 내용은벡터 레이어 메타데이터 속성 을 참조하세요.

9.3.3 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.

9.3.4 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 내용은기준 *(datum)* 변환 을 참조하세요.

9.3.5 기본 스타일 속성

Default Styles 탭에서 기존에 정의된 `.qml` 스타일이 없는 새 레이어를 어떻게 렌더링할지 제어할 수 있습니다. 다음과 같은 작업을 할 수 있습니다:

- 레이어 도형 유형은 물론 기본 *Color Ramp* 에 따라 적용할 기본 심볼 (*Marker*, *Line*, *Fill*) 을 설정할 수 있습니다.
- *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.

 *Style Manager* 버튼을 클릭하면스타일 관리자 대화창에 바로 접근해서 심볼 및 색상표의 환경을 설정할 수 있습니다.

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.

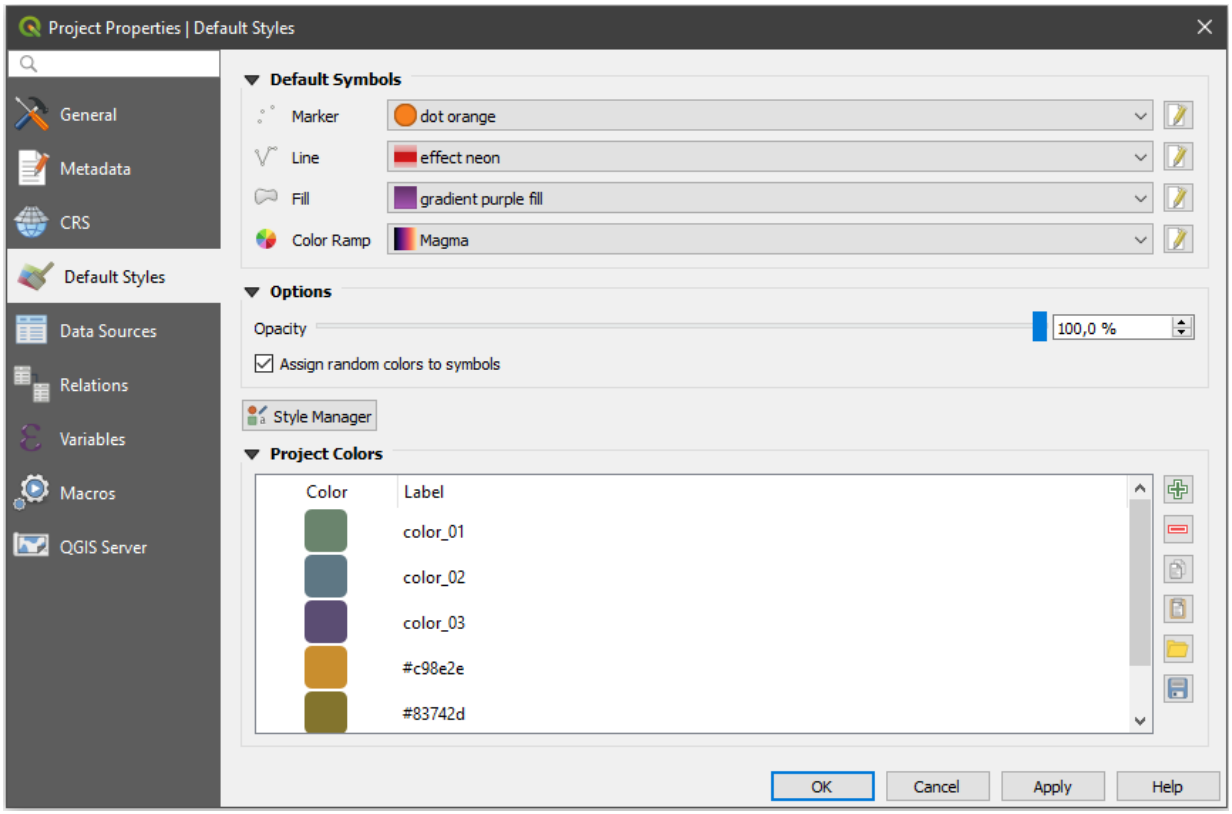


그림 9.24: 기본 스타일 탭

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.

9.3.6 데이터소스 속성

Data Sources 탭에서 다음 설정을 할 수 있습니다:

- ☐ *Automatically create transaction groups where possible*: 이 모드를 활성화한 경우, 동일 데이터베이스로부터 나온 모든 레이어의 편집 상태를 동기화합니다. 예를 들어, 레이어 하나를 편집 상태로 만들면 모든 레이어가 편집 상태로 바뀝니다. 레이어 하나를 커밋하거나 롤백하면, 다른 레이어들도 그렇게 됩니다. 또한 편집 변경 사항을 로컬에서 버퍼링하는 대신, 사용자가 레이어 저장을 클릭할 때 커밋되는 데이터베이스의 트랜잭션 (transaction) 으로 직접 전송합니다. 프로젝트에 있는 어떤 레이어도 편집 중이 아닐 경우에만 이 옵션을 (비) 활성화할 수 있다는 것을 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.
- ☐ *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.
- *Layers Capabilities* 를 다음과 같이 환경 설정할 수 있습니다:
  - 어떤 레이어가 `identifiable` 한지, 즉식별 도구 에 반응할지 설정합니다. 기본적으로 모든 레이어는 쿼리 가능하도록 설정돼 있습니다.
  - 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.
  - 어떤 레이어가 `searchable` 한지, 즉위치 탐색 위젯 을 통해 쿼리할 수 있을지 설정합니다. 기본적으로 모든 레이어는 검색 가능하도록 설정돼 있습니다.
  - 어떤 레이어가 `required` 상태로 정의돼 있는지 설정합니다. 목록에서 체크된 레이어들은 프로젝트에서 우연히 제거되는 일이 없도록 보호됩니다.

  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.

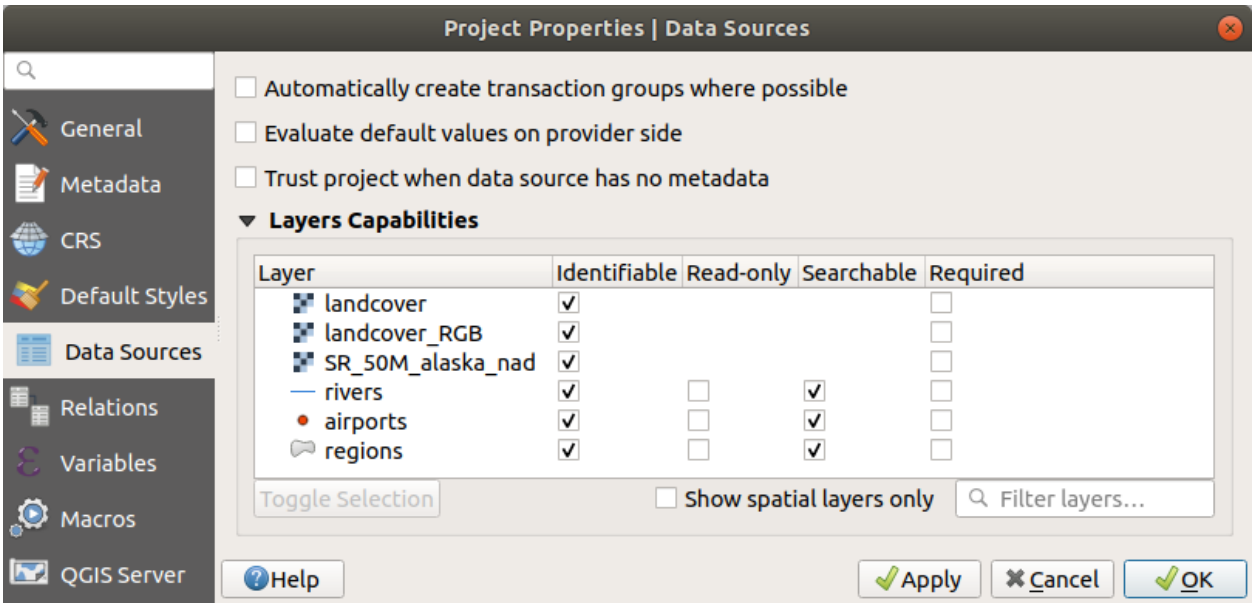


그림 9.25: 데이터소스 탭

9.3.7 관계 속성

Relations 탭은 일대다 (1:n) 관계를 정의하는 데 쓰입니다. 프로젝트 속성 대화창에서 이 관계를 정의합니다. 어떤 레이어에 관계가 존재하는 경우, 양식 뷰에 있는 새 사용자 인터페이스 요소가 (예를 들어 피처를 식별하고 그 양식을 열 때) 관련 요소들의 목록을 표시할 것입니다. 이렇게 되면, 예를 들어 어떤 파이프라인 또는 도로 구간의 점검 이력을 보여줄 수 있는 강력한 방법이 생기는 겁니다. 일대다 또는 다대다 관계 생성 에서 일대다 관계에 대한 더 자세한 내용을 찾아볼 수 있습니다.

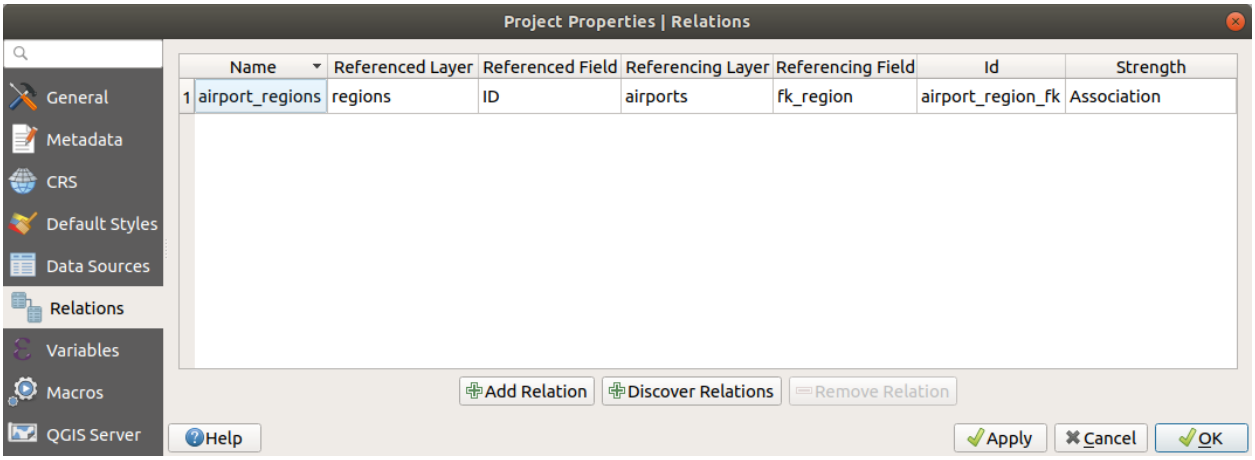


그림 9.26: 관계 탭

9.3.8 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.

9.3.9 매크로 속성

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

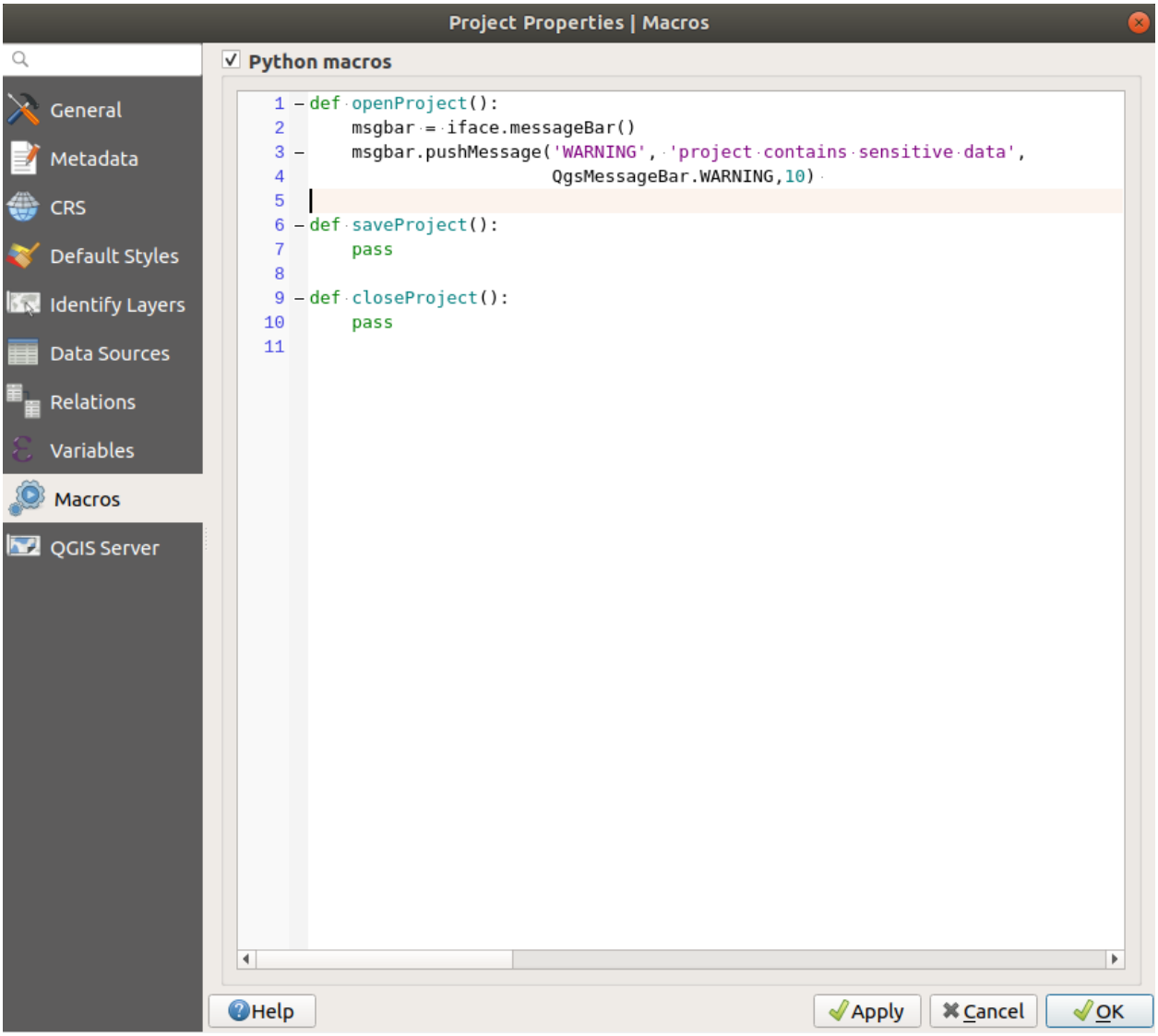


그림 9.27: QGIS 의 매크로 설정

9.3.10 QGIS 서버 속성

QGIS Server 탭은 사용자 프로젝트를 온라인 상에 퍼블리싱하기 위해 환경 설정할 수 있습니다. 이 탭에서 QGIS 서버의 WMS 및 WFS 성능, 범위, 그리고 좌표계 제약 조건에 관한 정보를 정의할 수 있습니다. Creatingwms-fromproject 와 이어지는 부분에서 더 자세한 내용을 다루고 있습니다.

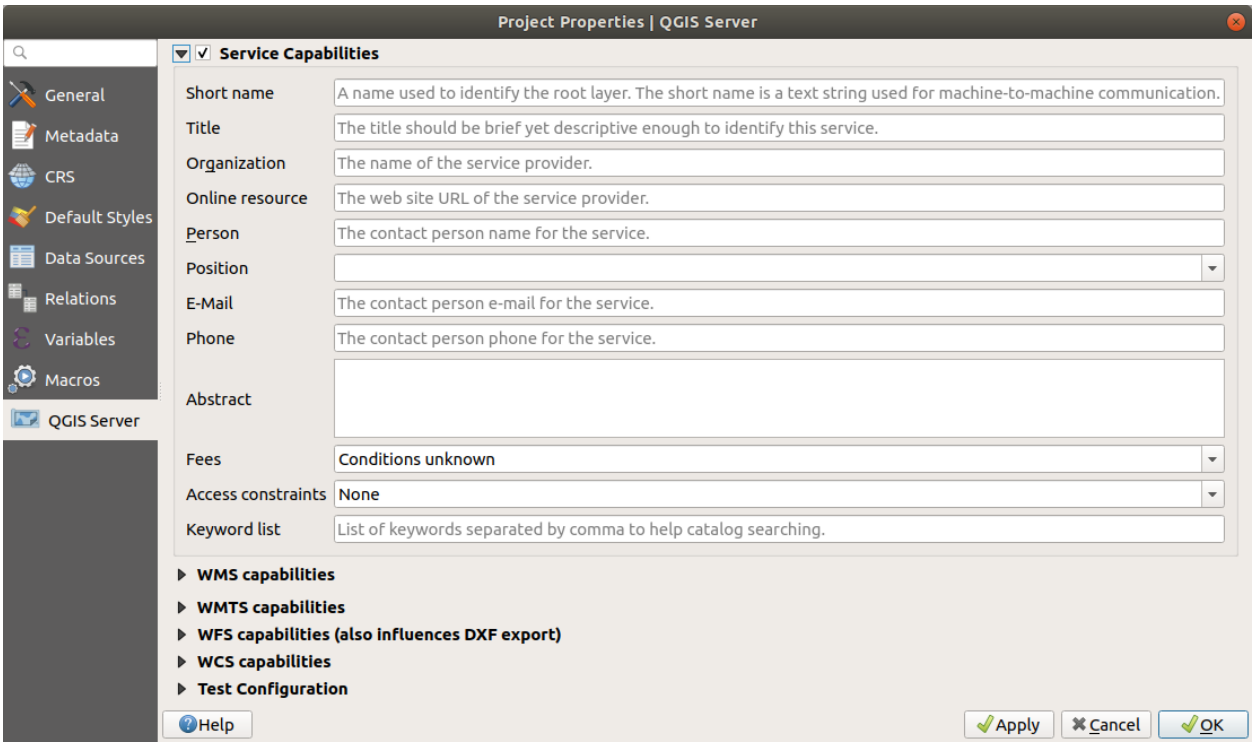


그림 9.28: QGIS 서버 설정 탭

9.3.11 시계열 속성

Temporal 탭에서 사용자 프로젝트의 시계열 범위를 설정할 수 있습니다. 사용자가 직접 범위를 입력할 수도 있고, 현재 프로젝트의 시계열 레이어로부터 범위를 계산할 수도 있습니다.

9.4 사용자 지정

사용자 지정 작업 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:

- **메뉴** 또는메뉴 바 에 있는 몇몇 하위 메뉴

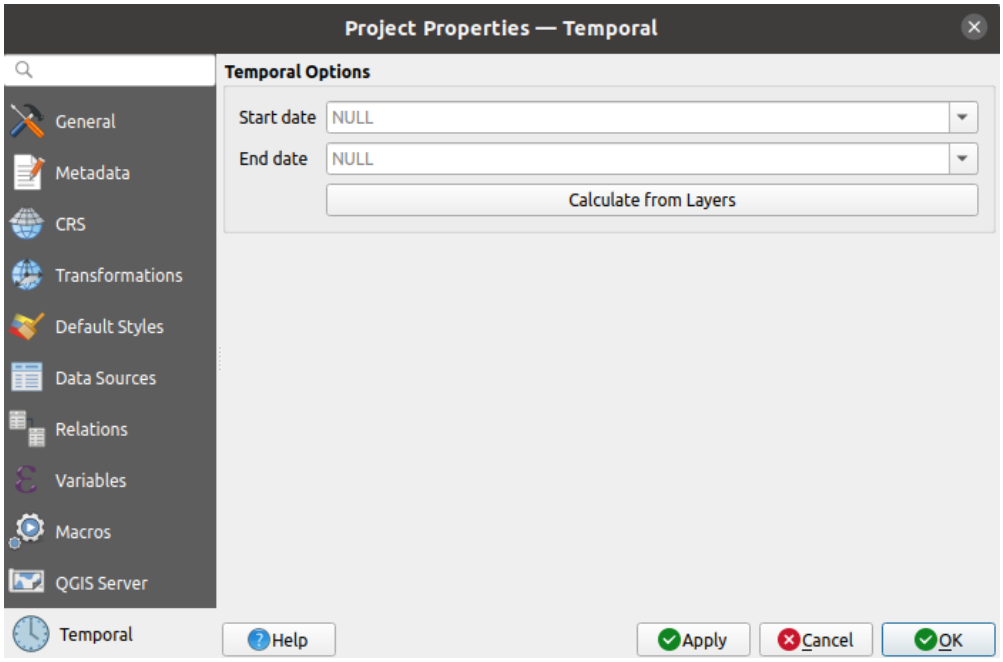


그림 9.29: QGIS 시계열 탭

- **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)

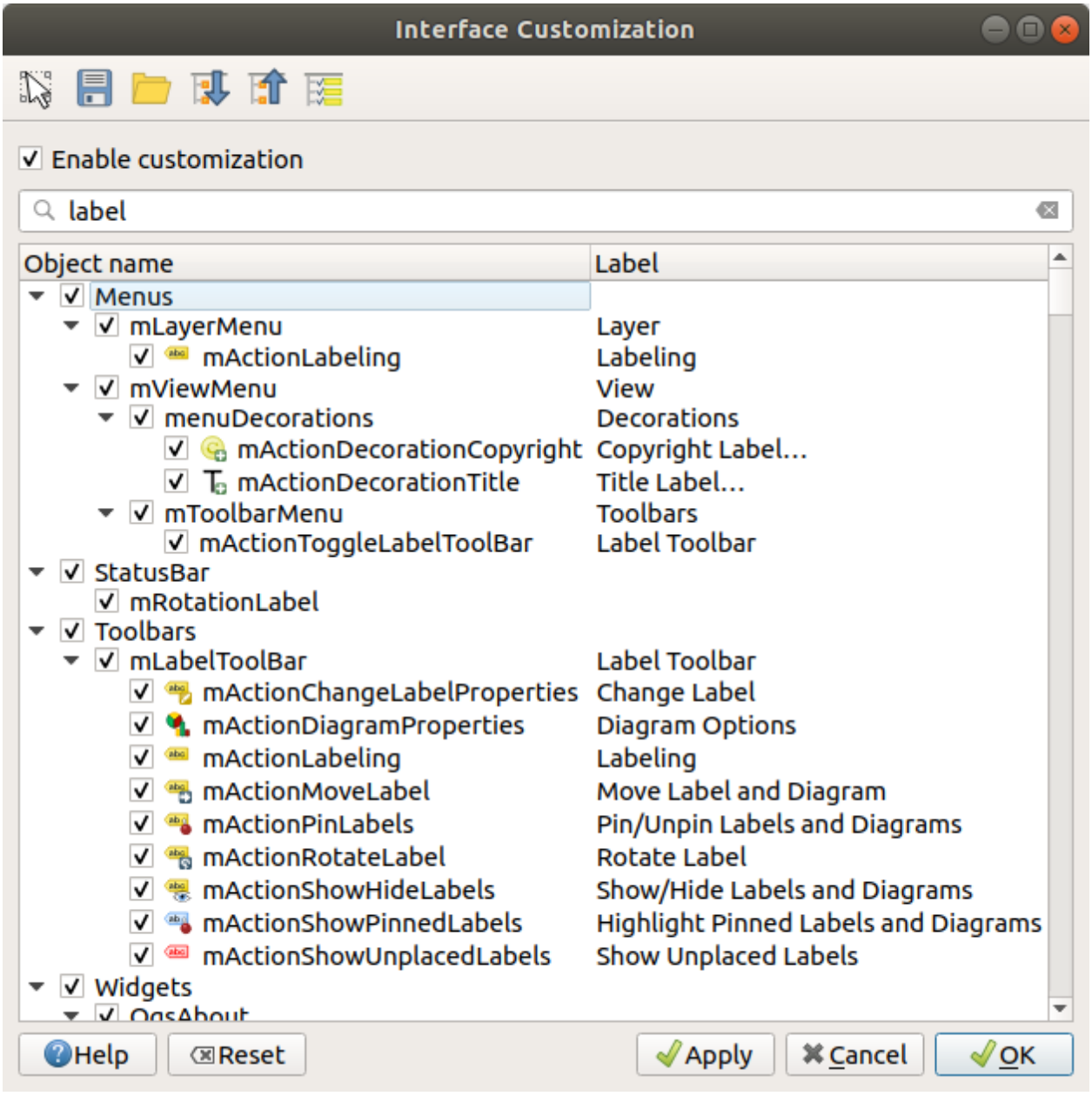


그림 9.30: 사용자 지정 작업 대화창

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

9.5 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.

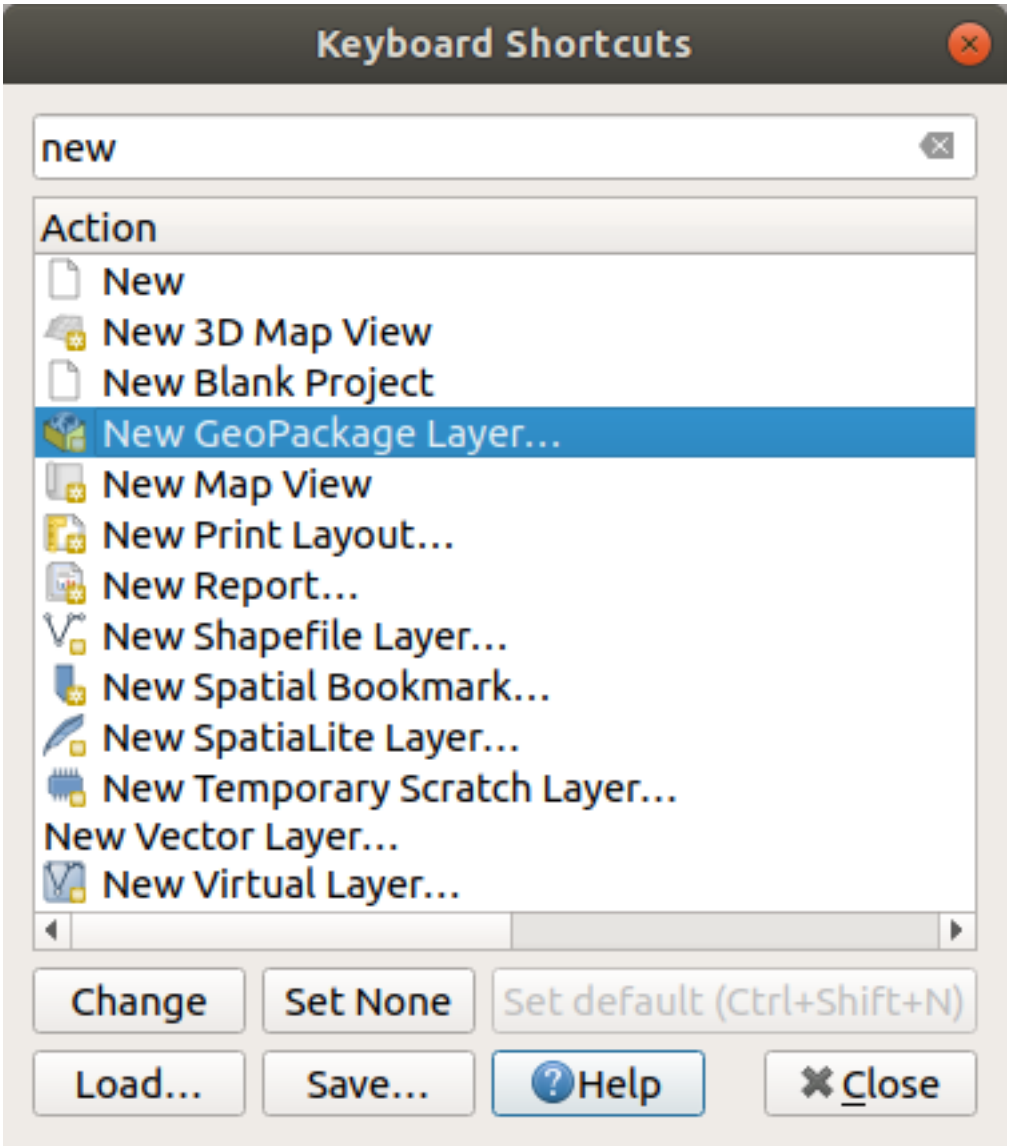


그림 9.31: 단축키 옵션 정의

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.

사용자가 지정하려는 다른 모든 도구에 대해서도 이와 같이 진행하십시오. 사용자 환경 설정을 마치고 나서, 대화창의 *Close* 버튼을 누르기만 하면 변경 사항이 적용됩니다. 또 *Save* 버튼을 눌러 변경 사항을 `.XML` 파일로 저장한 다음 또다른 QGIS 설치본에서 *Load* 버튼을 눌러 불러올 수도 있습니다.

9.6 고급 설정으로 QGIS 실행

9.6.1 명령 줄과 환경 변수

사용자의 OS 상에서 다른 응용 프로그램과 마찬가지로*QGIS 를 실행* 해보았습니다. QGIS 는 좀 더 고급의 사용 환경 설정을 위한 명령 줄 옵션을 제공합니다. (몇몇 경우 명령 줄 옵션 대신 환경 변수를 사용할 수도 있습니다.) 옵션 목록을 보려면 명령 줄에 `qgis --help` 라고 입력하십시오. 다음 목록을 반환합니다:

```
QGIS is a user friendly Open Source Geographic Information System.
Usage: /usr/bin/qgis.bin [OPTION] [FILE]
  OPTION:
        [--version]     display version information and exit
        [--snapshot filename]   emit snapshot of loaded datasets to given file
        [--width width] width of snapshot to emit
        [--height height]       height of snapshot to emit
        [--lang language]       use language for interface text (changes existing␣
↪override)
        [--project projectfile] load the given QGIS project
        [--extent xmin,ymin,xmax,ymax]  set initial map extent
        [--nologo]      hide splash screen
        [--noversioncheck]      don't check for new version of QGIS at startup
        [--noplugins]   don't restore plugins on startup
        [--nocustomization]     don't apply GUI customization
        [--customizationfile path]      use the given ini file as GUI customization
        [--globalsettingsfile path]     use the given ini file as Global Settings␣
↪(defaults)
        [--authdbdirectory path] use the given directory for authentication database
        [--code path]   run the given python file on load
        [--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.
        [--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.
```

(다음 페이지에 계속)

(이전 페이지에서 계속)

```
        [--help]                this text
        [--]            treat all following arguments as FILEs

  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, 다음 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

This option hides the splash screen when you start QGIS.

--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.

--customizationfile

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

--globalsettingsfile

이 옵션을 사용하면 기본 설정이라고도 하는 전체 수준 설정 파일 (`.ini`) 을 가리키는 경로를 지정할 수 있습니다. 지정한 파일에 있는 설정이 원래 기본 설정을 대체하지만, 사용자 프로파일의 설정이 최우선 순위를 가질 것입니다. 기본 전체 수준 설정의 위치는 `your_QGIS_PKG_path/resources/qgis_global_settings.ini` 파일입니다.

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.

이에 대응하는 환경 변수는 `QGIS_GLOBAL_SETTINGS_FILE` 입니다.

--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)
```

사용자의 현재 위치가 `load_alaska.py` 파일이 있는 디렉터리라고 가정하면, 다음 명령어를 사용해서 QGIS 를 실행한 다음 래스터 파일 `landcover.img` 를 불러와서 해당 레이어를'Alaska'로 명명할 수 있습니다:

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

--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

사용자의 프로파일 폴더에서 특정 프로파일을 사용해서 QGIS 를 실행합니다. 일부러 변경하지 않는 한, 선택한 사용자 프로파일을 다음 QGIS 세션에서 사용할 것입니다.

--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.

9.6.2 조직 내에서 QGIS 활용하기

사용자가 조직 내에서 사용자 지정 환경 설정 파일과 함께 QGIS 를 활용해야 하는 경우, 먼저 `your_QGIS_PKG_path/resources/qgis_global_settings.ini` 위치에 있는 기본 설정 파일을 복사/붙여넣기 해야 합니다. 이 파일은 이미 `[]` 로 시작하는 블록으로 식별되는 몇몇 기본 단락을 담고 있습니다. 이런 기본값들을 유지하면서 파일 하단에 사용자 고유의 단락을 추가하는 방법을 추천합니다. 파일 안에서 단락을 복사한 경우, QGIS 는 위에서 아래 방향으로 마지막 단락을 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"
```

설정 `INI` 파일의 가능성을 살펴보려면, QGIS 데스크탑에서 사용자가 원하는 환경 설정을 변경한 다음 텍스트 편집기를 통해 사용자 프로파일 폴더에 위치한 `INI` 파일에서 변경 사항을 찾아보는 것이 좋습니다. WMS/WMTS, PostGIS 연결, 프록시 설정, 맵 팁 등과 같은 수많은 설정을 `INI` 파일을 사용해서 설정할 수 있습니다.

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 10

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.

10.1 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.

10.2 레이어 좌표계

데이터를 특정 목표 좌표계로 정확히 투영하려면, 사용자 데이터가 좌표계 정보를 담고 있어야 하거나 사용자가 레이어에 정확한 좌표계를 직접 할당해야 합니다. PostGIS 레이어의 경우, QGIS 는 해당 PostGIS 레이어가 생성됐던 당시 지정된 공간 참조 식별자를 사용합니다. OGR 또는 GDAL 이 지원하는 데이터의 경우, QGIS 는 좌표계를 지정하는 공인된 방법 (recognized means) 이 있는지 여부에 의존합니다. 예를 들어 shapefile 포맷의 경우 레이어의 좌표계를 나타내는 WKT(ESRI Well-Known Text) 를 담고 있는 파일을 말합니다. 이 투영체 파일은 `.shp` 파일과 동일한 기본 명칭 (base name) 과 `.prj` 확장자로 이루어져 있습니다. 예를 들어 `alaska.shp` 파일은 이에 대응하는 `alaska.prj` 라는 투영체 파일을 가지고 있을 것입니다.

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.

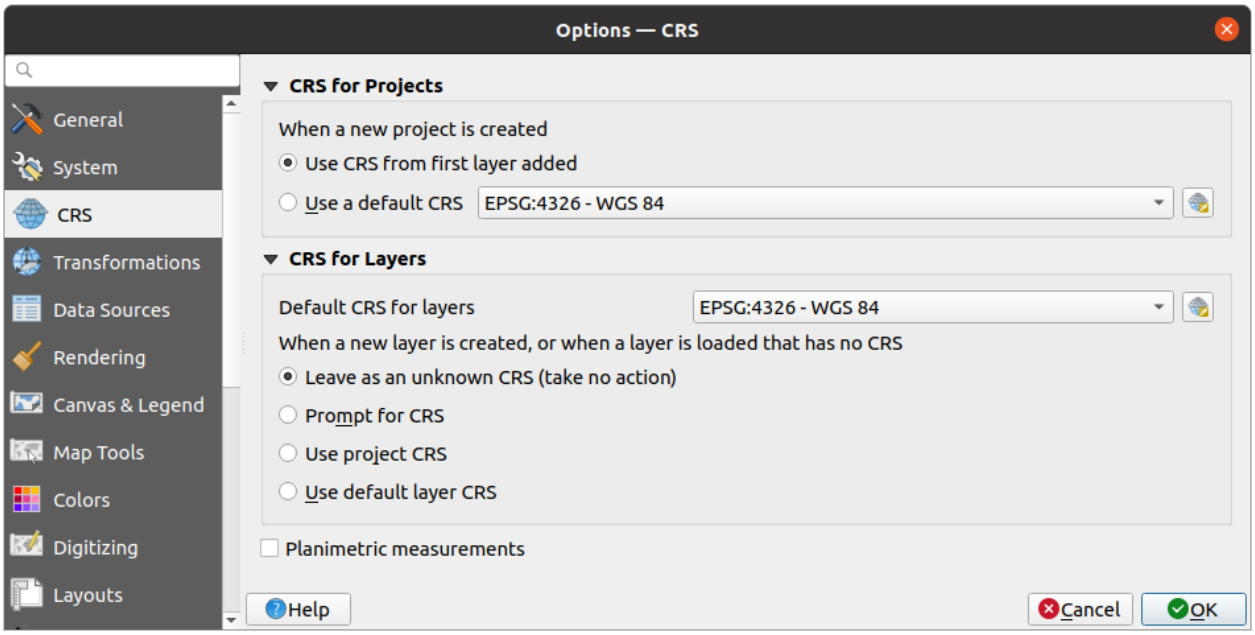


그림 10.1: QGIS 옵션 대화창의 CRS 탭

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*: 레이어에 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 원 (raw) 좌표를 해석하는 방법을 변경할 뿐입니다.

10.3 프로젝트 좌표계

QGIS 의 모든 프로젝트도 좌표계와 관련돼 있습니다. 프로젝트 좌표계는 사용자의 QGIS 맵 캔버스 내에서 데이터의 기저 원좌표를 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 경도/위도 기반 좌표계입니다. 이 기본 좌표계는 *Settings* ▸ *Options*⋯메뉴를 클릭하면 열리는 대화창의 *CRS* 탭에 있는 *CRS for New Projects* 설정을 통해 변경할 수 있습니다. (그림 10.1 그림 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 레이어와 맵 좌표를 지표면 상의 위치와는 아무 상관 없는 단순 2 차원 데카르트 좌표로 취급할 것입니다. 레이어 좌표계를 (원좌표를 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 원좌표를 이용해서 그릴 뿐입니다.
  - 타원체를 무시하고 `None/Planimetric` 으로 강제합니다.
  - 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)* 이나 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.

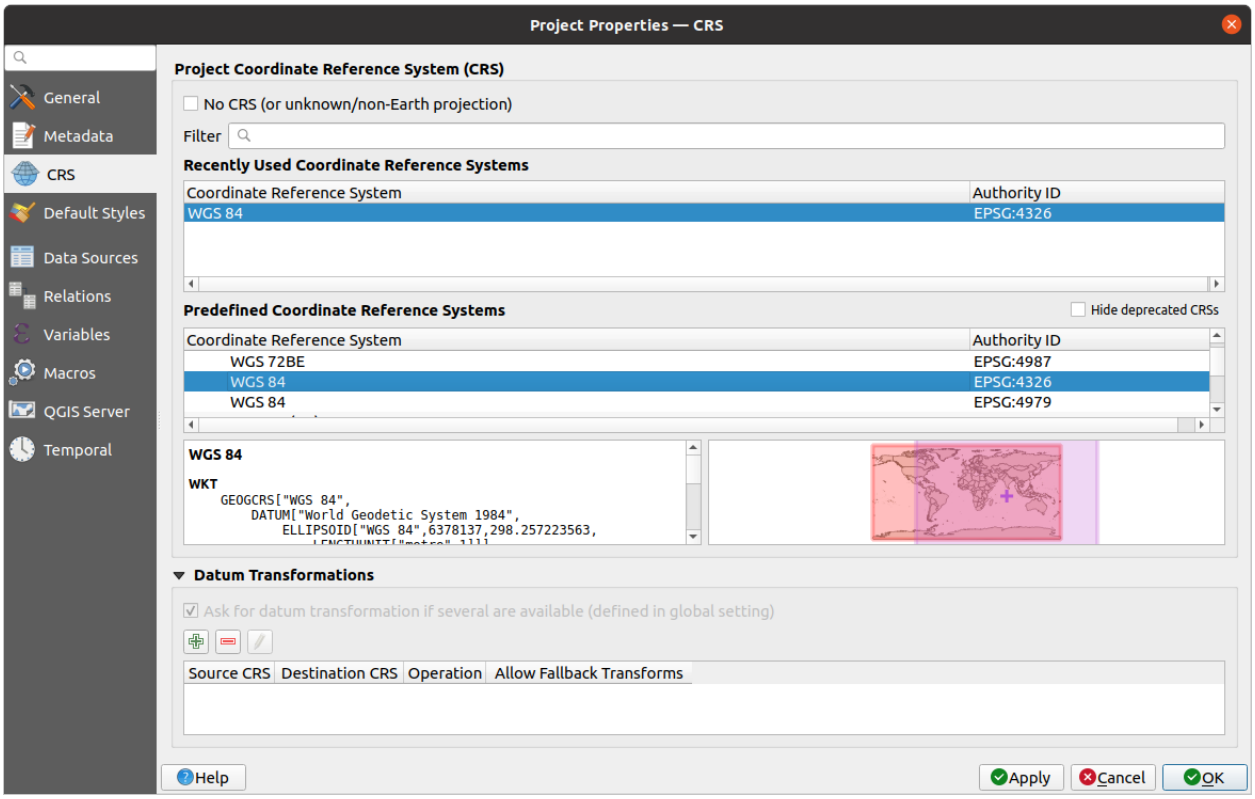


그림 10.2: 프로젝트 속성 대화창

사용자가 자신의 QGIS 프로젝트를 위한 새 좌표계를 선택할 때마다, 선택한 좌표계와 일치시키기 위해 *Project properties* 대화창 (*Project → Properties…*) 의 *General* 탭에서 측정 단위를 자동적으로 변경할 것입니다. 예를 들어, 몇몇 좌표계는 자신의 좌표를 미터가 아닌 피트 단위로 정의하기 때문에 사용자의 QGIS 프로젝트를 그런 좌표계 가운데 하나로 설정하면 사용자 프로젝트가 피트 단위를 기본값으로 사용해서 측정하도록 설정할 것입니다.

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.

10.4 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 명칭을 알고 있을 경우, 검색 기능을 이용해서 좌표계를 찾을 수 있습니다. EPSG 코드, 식별자, 또는 명칭을 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.

좌표계 선택기는 선택한 좌표계를 유효하게 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.

10.5 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 명칭
2. PROJ 또는 WKT 문법으로 된 지도 제작 파라미터

새 좌표계를 생성하려면:

1. Add new CRS 아이콘을 클릭합니다.
2. 좌표계를 설명하는 명칭을 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 유효한 투영체 정의인지 테스트해보려면 *Validate* 를 클릭하십시오.

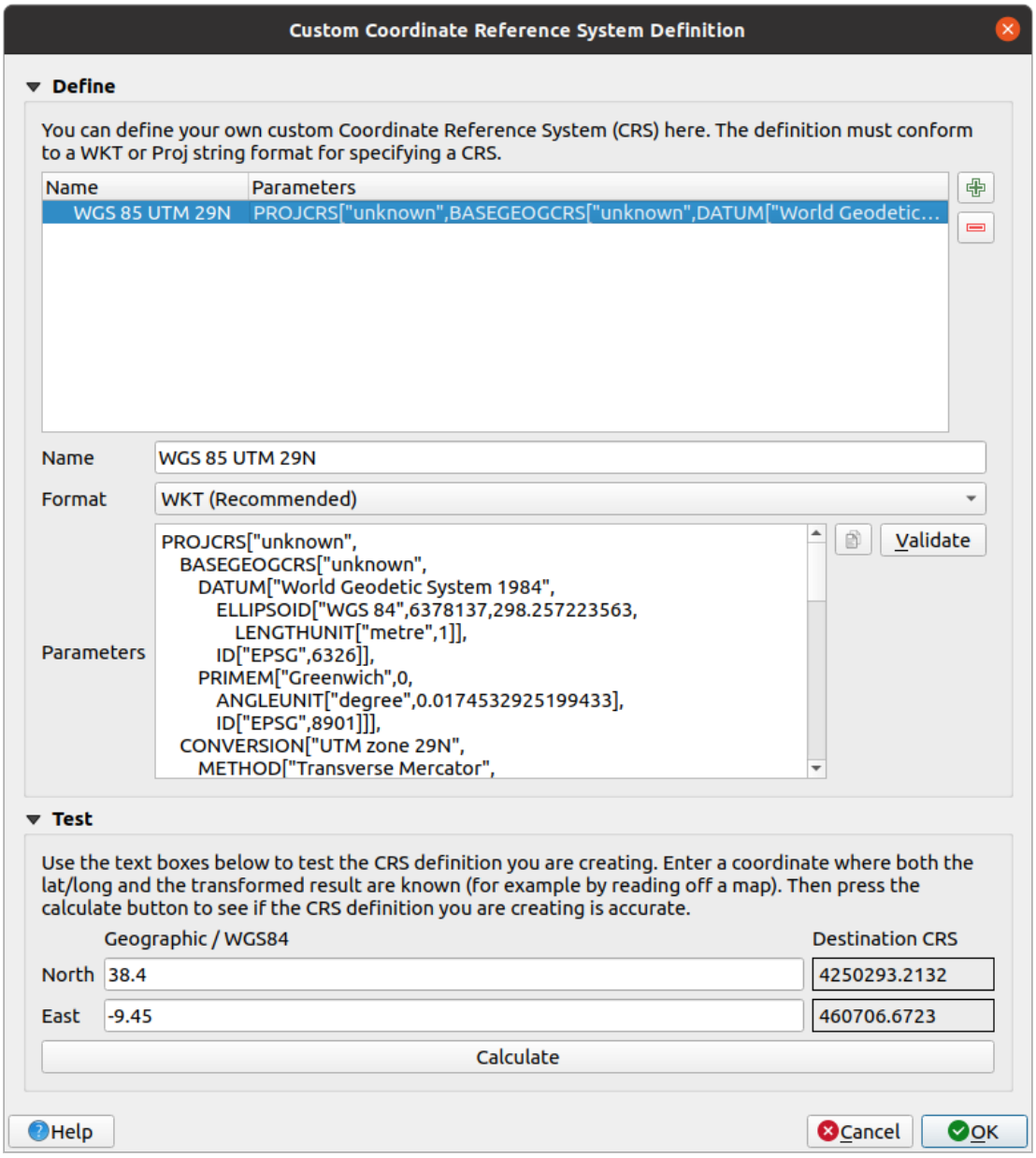


그림 10.3: 사용자 지정 좌표계 대화창

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.

10.5.1 QGIS 에서 NTv2 변환 통합

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

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

그림 10.4: NTv2 변환 설정하기

10.6 기준 (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 기준 변환 방법이 하나 이상 있을 경우, 사용자에게 프로젝트에서 어떤 기준 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 기준 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.

     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.

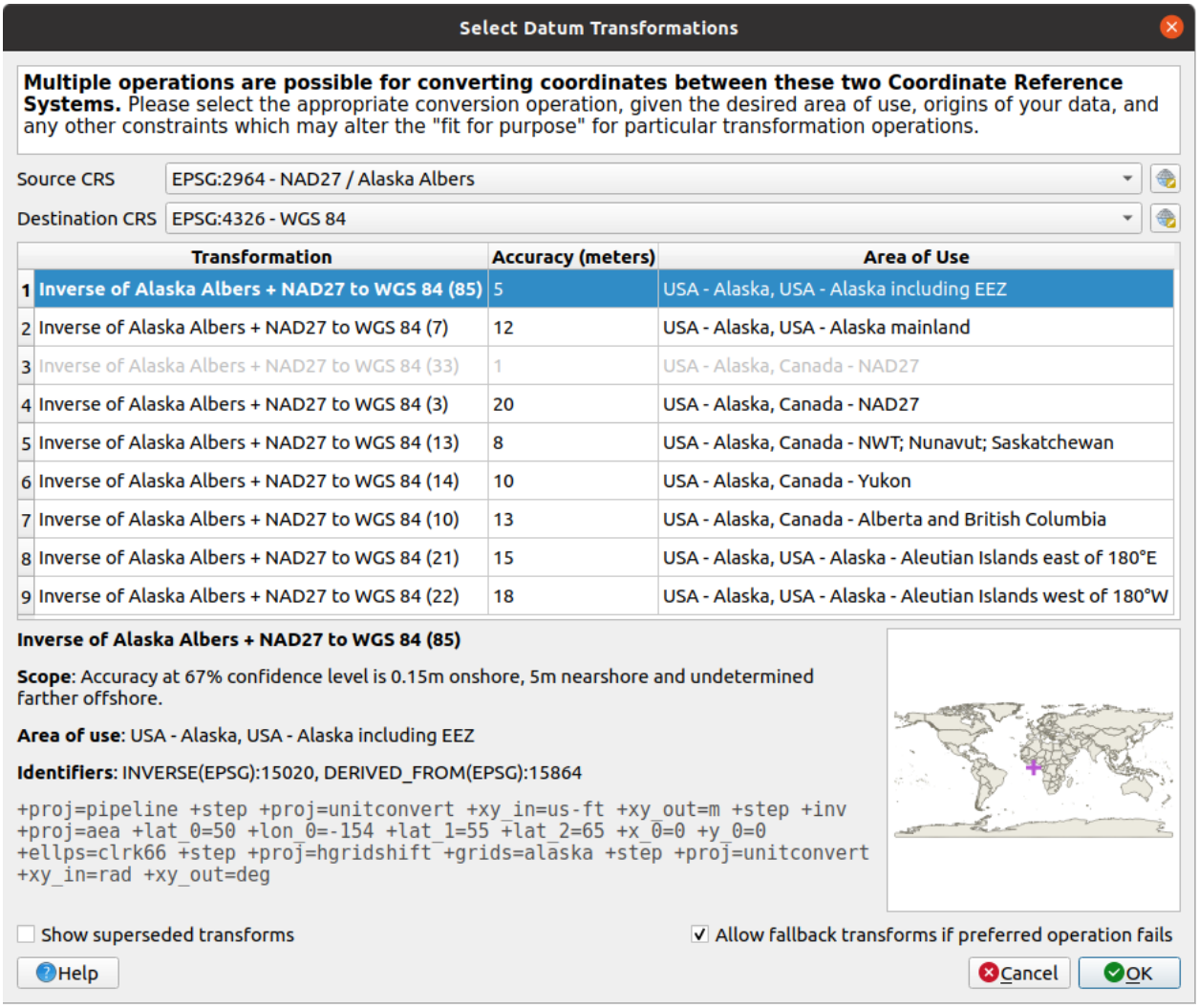


그림 10.5: 선호하는 기본 기준 변환을 선택하기

5. *Allow fallback transforms if preferred operation fails*: 사용자가 선호하는 변환 방법이 실패할 경우 변환을 되돌릴지 여부를 설정하십시오.

6. *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 자동적으로 이제부터 선택한 기준 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 기준 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 11

일반 도구

11.1 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.

11.2 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.

11.2.1 레이어 패널

Layers 패널 (통칭"맵 범례") 에서 프로젝트가 보유한 모든 레이어 목록을 볼 수 있으며, 레이어의 가시성을 관리할 수 있습니다. `Ctrl+1` 키 조합을 눌러 레이어를 보이게 하거나 숨길 수 있습니다. Z 순서를 변경하기 위해 범례 안에서 레이어를 선택해, 위아래로 드래그할 수 있습니다. Z 순서란 범례 목록 상위에 위치한 레이어가 범례 목록 하위에 있는 레이어를 덮는다는 뜻입니다.

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

레이어 패널 상단에 있는 툴바의 기능은 다음과 같습니다:

-  Open the layer styling dock (F7): 레이어 스타일 작업 패널을 켜고 끕니다.
-  Add new group: 새 레이어 그룹을 만듭니다.
-  Manage Map Themes: 레이어의 가시성을 제어하고 서로 다른 맵 테마에 레이어를 배치합니다.

- Filter Legend by Map Content: 맵에 표시되며 피처가 현재 맵 캔버스 안에 들어오는 레이어만, 레이어 패널에 레이어 스타일을 렌더링합니다. 해당 사항이 없을 경우, 일반적인 NULL 심볼로 표시합니다. 레이어 심볼을 바탕으로 하는 이 방식으로 어떤 레이어의 어떤 피처 유형이 사용자의 관심 지역에 들어오는지 손쉽게 알아볼 수 있습니다.
- 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.

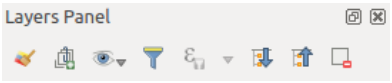


그림 11.1: 레이어 패널의 레이어 툴바

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

맵 테마 환경 설정하기

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 환경 설정할 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 환경 설정하십시오.
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 명칭 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 명칭을 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 명칭을 변경할 수도 있고, *Remove Current Theme* 버튼을 클릭하면 활성 테마를 삭제할 수 있습니다.

맵 테마는 미리 환경 설정해놓은 서로 다른 조합들을 빠르게 바꾸는 데 유용합니다. 목록에서 맵 테마를 선택하면 해당 테마의 조합을 복구합니다. 인쇄 조판기에서도 환경 설정해놓은 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 주 요소는 프로젝트에 추가된 벡터 또는 래스터 레이어 목록을 담은 프레임으로, 이 레이어들을 그룹화할 수도 있습니다. 패널에서 어떤 항목을 선택하느냐에 따라 오른쪽 클릭으로 나타나는 컨텍스트 메뉴의 내용이 달라집니다. 컨텍스트 메뉴의 옵션들은 다음과 같습니다.

옵션	벡터 레이어	래스터 레이어	그룹
Zoom to Layer/Group	✓	✓	✓
Zoom to Selection	✓		
Show in Overview	✓	✓	
Show Feature Count	✓		
Copy Layer/Group	✓	✓	✓
Rename Layer/Group	✓	✓	✓
Zoom to Native Resolution (100%)		✓	
Stretch Using Current Extent		✓	
Update SQL Layer…	✓		

다음 페이지에 계속

표 11.1 – 이전 페이지에서 계속

옵션	벡터 레이어	래스터 레이어	그룹
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	✓	✓	
Set CRS	✓	✓	
Set Layer/Group CRS…	✓	✓	✓
Set Project CRS from Layer	✓	✓	
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…	✓		
Save Selected Features As…	✓		
Save As Layer Definition File…	✓	✓	✓
Save As QGIS Layer Style File…	✓	✓	
Styles	✓	✓	
Copy Style	✓	✓	

다음 페이지에 계속

표 11.1 – 이전 페이지에서 계속

옵션	벡터 레이어	래스터 레이어	그룹
Paste Style	☑	☑	☑
Add…	☑	☑	
Rename Current…	☑	☑	
Edit symbol…	☑		
Copy Symbol	☑		
Paste Symbol	☑		
Properties…	☑	☑	

표: 레이어 패널의 컨텍스트 메뉴 항목

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 다음 두 가지 방법이 있습니다:

1. 아이콘을 클릭해서 새 그룹을 추가합니다. 그룹의 명칭을 입력한 다음 `Enter` 를 누르십시오. 이제 기존 레이어를 클릭한 채로 그룹으로 드래그하십시오.
2. 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

레이어/그룹에 대한 상세 정보를 제공하기 위한 아이콘이 *Layers* 패널에 있는 레이어 또는 그룹 옆에 나타나는 경우가 종종 있습니다. 다음과 같은 심볼로 나타납니다:

-  : 레이어가 편집 상태여서 사용자가 데이터를 수정할 수 있다는 것을 나타냅니다.
-  : 편집 중인 레이어에 일부 저장되지 않은 변경 사항이 있다는 것을 나타냅니다.
-  : 레이어에필터 가 적용됐다는 것을 나타냅니다. 이 아이콘 위에 마우스를 가져가면 필터 표현식을 볼 수 있고, 더블클릭하면 설정을 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. (엉터리 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 OGR 벡터 유형 가운데 하나로 저장하십시오.
-  : 좌표계가 없거나, 알려지지 않은 좌표계를 가진 레이어를 식별합니다.
-  : 캔버스 애니메이션이 제어하는 시계열 레이어를 식별합니다.

벡터 레이어 스타일 편집 작업

레이어 패널에서 레이어 렌더링을 쉽고 빠르게 변경할 수 있는 빠른 방법이 있습니다. 벡터 레이어를 오른쪽 클릭한 다음 목록에서 *Styles* 를 선택하면 다음과 같은 작업을 할 수 있습니다:

- 레이어에 현재 적용된스타일 을 확인합니다. 해당 레이어에 여러 스타일을 정의했을 경우, 한 스타일에서 다른 스타일로 바꿔가면 맵 캔버스에서 해당 레이어의 렌더링이 자동적으로 업데이트되는 걸 볼 수 있습니다.
- 현재 스타일의 일부분 또는 전부를 복사해서, 이 스타일을 적용할 수 있는 다른 레이어에 붙여넣을 수 있습니다.

  팁: 손쉽게 레이어 스타일 공유

  컨텍스트 메뉴에서, 레이어의 스타일을 복사해 그룹 또는 레이어 집합에 붙여 넣을 수 있습니다. 원본 레이어와 동일한 유형 (벡터 또는 래스터) 인 모든 레이어에, 그리고 벡터 레이어의 경우 동일한 도형 유형 (포인트, 라인, 폴리곤) 을 가진 모든 레이어에 스타일을 적용합니다.

- 현재 스타일의 명칭을 변경하거나, (사실 현재 스타일의 복사본인) 새 스타일을 추가하거나, 또는 (다중 스타일을 사용할 수 있는 경우) 현재 스타일을 삭제할 수 있습니다.

참고: 이 작업들은 래스터 또는 메시 레이어에 대해서도 똑같이 쓸 수 있습니다.

- update the *symbol color* using a **Color Wheel**. For convenience, the recently used colors are also available at the bottom of the color wheel.
- *Edit symbol*…: 심볼 선택기 대화창을 열고 피처 심볼을 변경합니다. (심볼, 크기, 색상 등등)

(범주, 등급 또는규칙 기반 을 기반으로 하는) 범주 심볼 유형을 사용하는 경우, 범주 단계 항목의 컨텍스트 메뉴에서 앞에서 언급한 심볼 수준 옵션을 사용할 수 있습니다. 또 *Toggle Items*, *Show All Items* 및 *Hide All Items* 항목을 통해 피처의 모든 범주의 가시성을 전환할 수도 있습니다. 이렇게 하면 항목들을 하나하나 체크 또는 체크 해제하지 않아도 됩니다.

팁: 범주의 잎 (leaf) 항목을 더블 클릭하면 *Symbol Selector* 대화창을 엽니다.

11.2.2 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:

- 레이어 유형에 따라:
  - 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.
  - 메시 레이어의 경우 레이어의 *Symbology* 및 *3D View* 속성을 설정합니다. 이 옵션들은메시 데이터셋 속성 과 동일합니다.
- *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.

벡터 타일 레이어의 경우, 현재 화면에 보이는 규칙만 표시하는 *Visible rules only* 옵션이 있습니다. 사용자가 현재 맵 캔버스 확대/축소 수준 안에 들어오는 규칙만 작업하기를 바라는 경우 매우 유용한 옵션입니다.

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.

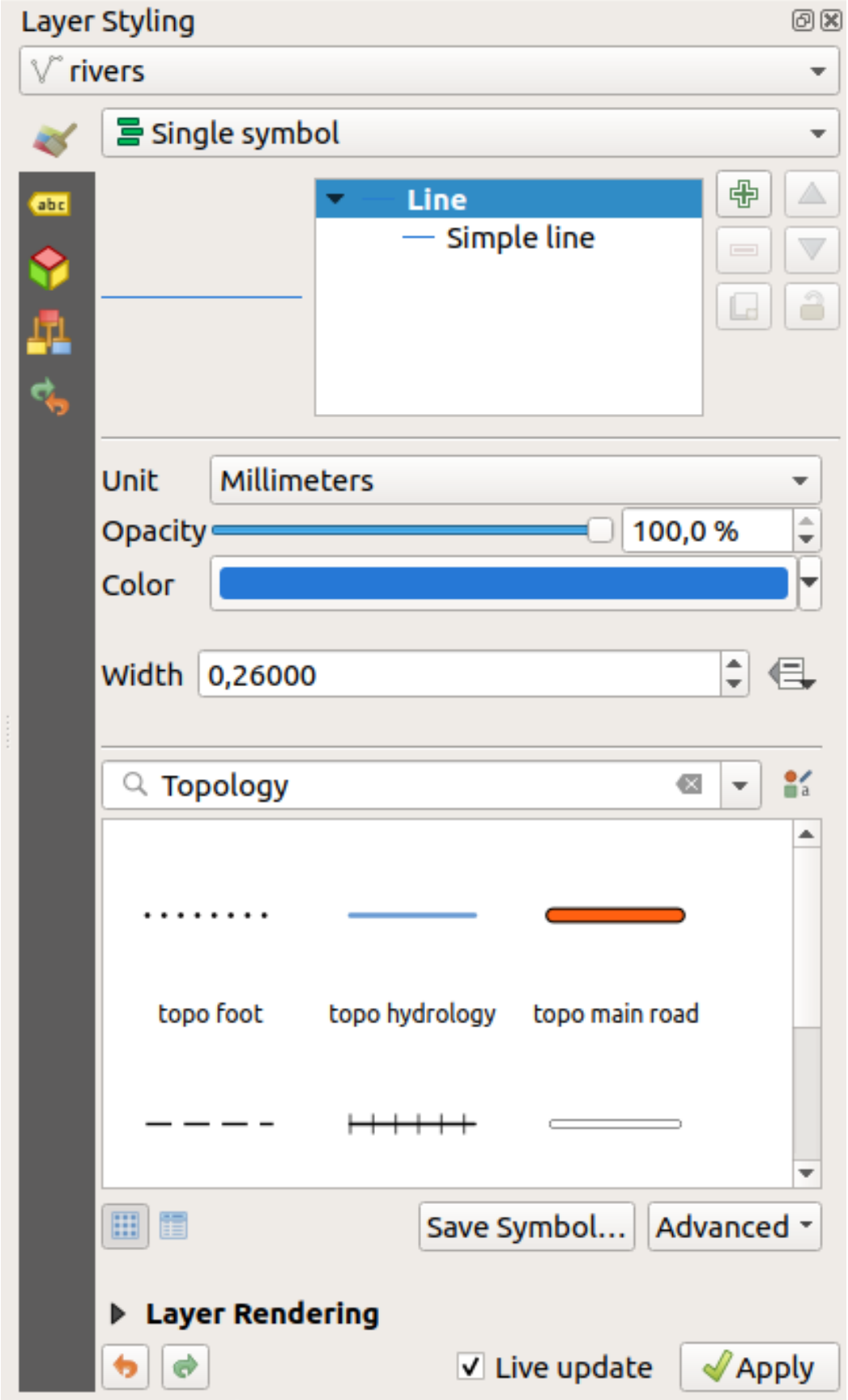


그림 11.2: 레이어 스타일 작업 패털에서 레이어 심볼 정의하기

11.2.3 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 레이어 패널에 있는 순서에 독립적으로 레이어를 렌더링하는 순서를 정의할 수 있습니다. 레이어 목록 아래 있는 ☑ *Control rendering order* 옵션을 체크하고 사용자가 원하는 대로 패널에 있는 레이어들을 재정렬하십시오. 예를 들어 그림 11.3 그림을 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.

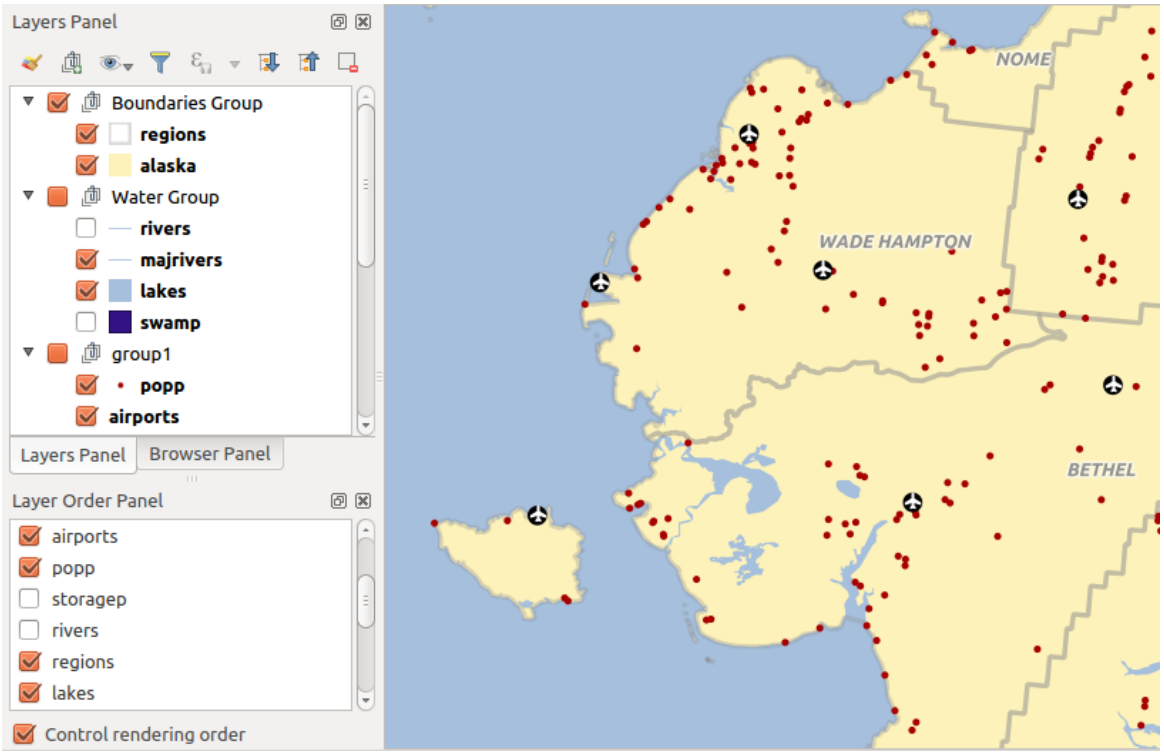


그림 11.3: 범례에 독립적인 레이어 순서 정의하기

11.2.4 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.

11.2.5 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.

11.2.6 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.

11.2.7 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
- 사용할 열 (column) 또는 표현식
- 대화창 우하단의 드롭다운 버튼을 통해 반환할 통계 정보. 레이어 필드 유형에 따라 (또는 표현식의 값 유형에 따라) 사용할 수 있는 통계 정보는 다음과 같습니다:

통계 정보	문자열	정수	실수	날짜
개수	☑	☑	☑	☑
고유 값 (Distinct Value) 개수	☑			☑
누락 값 (Missing value) 개수	☑	☑	☑	☑
합계		☑	☑	
평균		☑	☑	☑
표준 편차		☑	☑	
표본 표준 편차		☑	☑	
최소값	☑	☑	☑	☑
최대값	☑	☑	☑	☑
범위		☑	☑	☑
소수 기준 (Minority)	☑	☑	☑	
다수 기준 (Majority)	☑	☑	☑	
다양도 (Variety)		☑	☑	
제 1 사분위수 (First Quartile)		☑	☑	
제 3 사분위수 (Third Quartile)		☑	☑	
사분위간 범위 (Inter Quartile Range)		☑	☑	
최소 길이	☑			
최대 길이	☑			
평균 길이	☑			

표: 필드 유형에 따라 사용할 수 있는 통계

통계 요약 정보는:

- 전체 레이어의 통계를 반환하거나, *selected features only* 옵션으로 선택한 피처에 대한 통계만 반환할 수도 있습니다.
- 기저 데이터소스가 바뀔 경우 (예를 들어 피처/필드를 새로 추가하거나, 피처/필드를 제거하거나, 속성을 수정하는 경우) 버튼을 통해 통계를 다시 계산할 수 있습니다.
- 클립보드로 복사한 다음 다른 응용 프로그램에 테이블로 붙여넣을 수 있습니다.

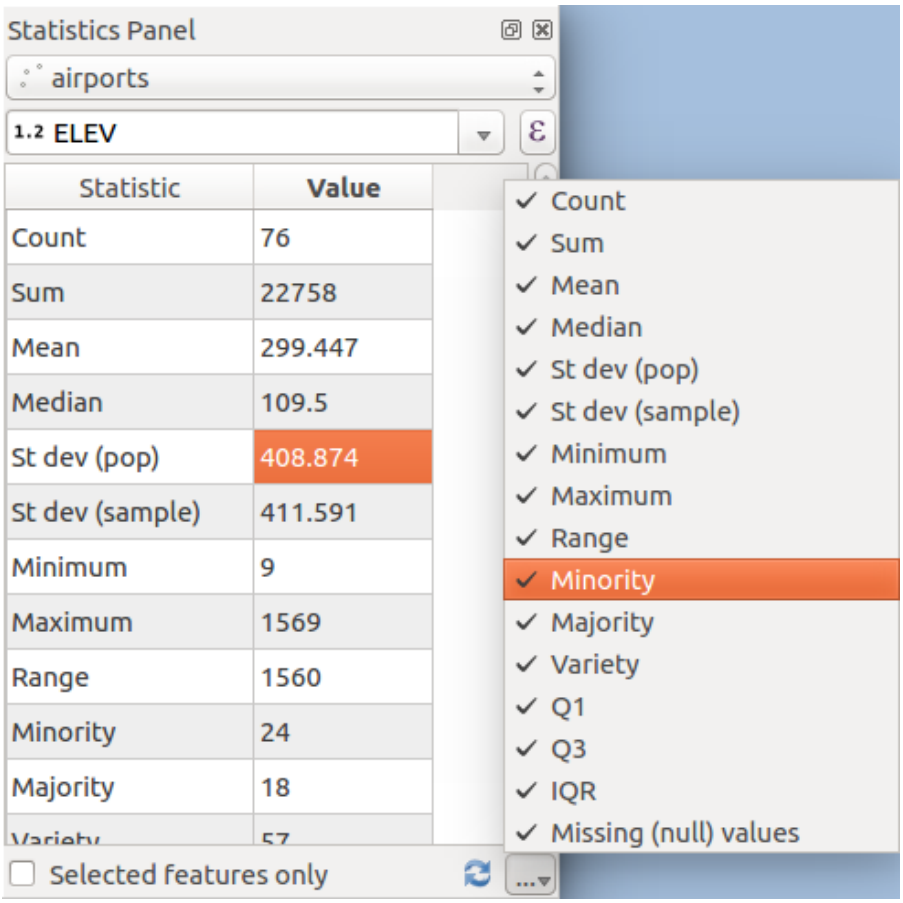


그림 11.4: 필드에 대한 통계 정보 종류 표시

11.3 프로젝트 내포 작업

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 11.5 그림 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 명칭 옆에 아이콘이 추가되고, 그 위에 마우스를 가져가면 원본 프로젝트 파일 경로를 보여주는 툴팁을 표시합니다.

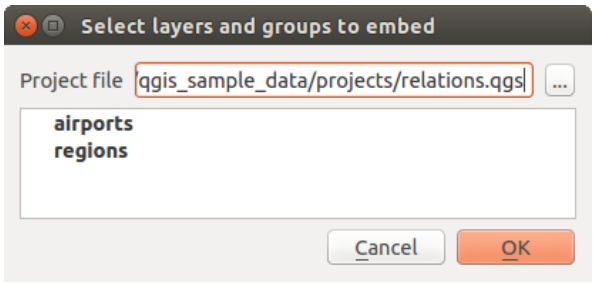


그림 11.5: 삽입할 레이어 및 그룹 선택

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.

11.4 맵 캔버스 작업

11.4.1 렌더링 작업

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

- 레이어 추가
- 맵 이동 또는 확대/축소
- QGIS 창 크기 변경
- 레이어 (들) 의 가시성 변경

사용자는 QGIS 상에서 다양한 방법으로 렌더링 과정을 제어할 수 있습니다.

축척에 따른 렌더링

축척에 의존적인 렌더링을 통해 어떤 (래스터 또는 벡터) 레이어를 가시화하는 최소 및 최대 축척을 설정할 수 있습니다. 축척 의존 렌더링을 설정하려면, 범례에 있는 레이어를 더블 클릭해서 *Properties* 대화창을 여십시오. *Rendering* 탭에서 *Scale dependent visibility* 를 체크한 다음, *Minimum (exclusive)* 과 *Maximum (inclusive)* 축척값을 입력합니다.

레이어 패널에서 레이어에 대한 축척 의존 가시성을 활성화시킬 수도 있습니다. 레이어를 오른쪽 클릭한다음 컨텍스트 메뉴에서 *Set Layer Scale Visibility* 를 선택하십시오.

Set to current canvas scale 버튼을 클릭하면 가시성 범위의 한계값으로 현재 맵 캔버스의 축척을 쓸 수 있습니다.

참고: 맵 축척이 가시성 축척 범위를 벗어나서 맵 캔버스에 레이어를 렌더링하지 않는 경우, 레이어 패널에서 해당 레이어를 회색조로 표시하며 해당 레이어의 컨텍스트 메뉴에 새로운 옵션 *Zoom to Visible Scale* 이 추가됩니다. 이 옵션을 선택하면 해당 레이어가 가시화될 가장 가까운 축척값으로 맵을 확대/축소할 것입니다.

맵 렌더링 제어

다음에서 설명하는 대로, 다양한 방법으로 맵 렌더링을 제어할 수 있습니다.

렌더링 보류

렌더링 작업을 보류하려면, 상태 바 우하단에 있는 *Render* 체크박스를 클릭하십시오. *Render* 체크박스를 해제하면, QGIS 는렌더링 작업 에서 설명하는 어떤 이벤트에도 대응하지 않고 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.

레이어 추가 옵션 설정

새 레이어를 불러올 때 항상 그리지 않도록 하는 옵션을 설정할 수 있습니다. 즉 맵에 레이어가 추가되어도, 기본적으로 범례의 가시성 체크박스는 해제된 상태일 것이라는 뜻입니다. 이 옵션을 설정하려면, *Settings* ▸ *Options* 메뉴 옵션을 선택한 다음 *Rendering* 탭을 클릭하십시오. *By default new layers added to the map should be displayed* 체크박스를 해제하십시오. 이후 맵에 추가되는 모든 레이어가 기본적으로 꺼져 있게 (보이지 않게) 될 것입니다.

렌더링 중단

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

렌더링 질 변경

QGIS 는 맵의 렌더링 질을 변경할 수 있는 옵션을 보유하고 있습니다. *Settings* → *Options* 메뉴 옵션을 선택하고, *Rendering* 탭을 클릭한 다음 *Make lines appear less jagged at the expense of some drawing performance* 체크박스를 체크하거나 해제해보십시오.

렌더링 속도 향상

몇 가지 설정으로 렌더링 속도를 향상시킬 수 있습니다. *Settings* → *Options* 메뉴 옵션으로 QGIS 설정 대화창을 열고, *Rendering* 탭으로 이동해서 다음 체크박스들을 체크하거나 해제해보십시오:

- *Use render caching where possible to speed up redraws* 를 체크
- *Render layers in parallel using many CPU cores* 를 체크한 다음 *Max cores to use* 를 설정
- 백그라운드에서 맵을 별개의 이미지로 렌더링해서 각 *Map Update interval* 시간마다 이 (오프스크린) 이미지의 내용을 가져다가 맵 캔버스를 업데이트하게 됩니다. 하지만 렌더링 작업이 이 간격보다 빨리 끝나는 경우, 그 즉시 화면에 업데이트됩니다.
- *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.

11.4.2 확대/축소 및 이동

여러 가지 방법으로 관심 지역으로 확대/축소하고 이동할 수 있습니다. *Map Navigation* 툴바를 사용할 수도 있고, 맵 상에서 마우스와 키보드를 사용할 수도 있습니다. 또 *View* 메뉴에 있는 메뉴 액션을 사용하거나 *Layers* 패널에서 해당 레이어의 컨텍스트 메뉴를 사용할 수도 있습니다.

아이콘	라벨	사용법	뷰 메뉴	맵 탐색 툴바	레이어 컨텍스트 메뉴
	맵 이동	이 도구를 활성화하면, 맵 캔버스 상 어디를 왼쪽 클릭하건 커서 위치로 맵을 이동시킵니다. 또 왼쪽 마우스 버튼을 누른 채 맵 캔버스를 드래그해서 맵을 이동시킬 수도 있습니다.	☑	☑	
	확대	이 도구를 활성화하면, 맵 캔버스 상 어디를 왼쪽 클릭하건 한 단계 확대합니다. 마우스 커서 위치를 중심으로 관심 지역을 확대할 것입니다. 또 왼쪽 마우스 버튼을 누른 채 맵 캔버스 상에 직사각형을 드래그해서 확대할 수도 있습니다.	☑	☑	
	축소	이 도구를 활성화하면, 맵 캔버스 상 어디를 왼쪽 클릭하건 한 단계 축소합니다. 마우스 커서 위치를 중심으로 관심 지역을 축소할 것입니다. 또 왼쪽 마우스 버튼을 누른 채 맵 캔버스 상에 직사각형을 드래그해서 해당 지역에서 맵을 축소할 수도 있습니다.	☑	☑	

다음 페이지에 계속

표 11.2 – 이전 페이지에서 계속

아이콘	라벨	사용법	뷰 메뉴	맵 탐색 툴바	레이어 컨텍스트 메뉴
	선택한 집합으로 맵 이동	맵을 활성화 레이어의 선택한 피처로 이동합니다.	☑	☑	
	선택한 집합으로 확대/축소	활성화 레이어의 선택한 피처로 확대/축소합니다.	☑	☑	☑
	레이어로 확대/축소	활성화 레이어의 전체 영역으로 확대/축소합니다.	☑	☑	☑
	전체 영역으로 확대/축소	프로젝트에 있는 모든 레이어의 전체 영역으로 확대/축소합니다.	☑	☑	
	이전 영역으로 확대/축소	액션 이력에 있는 이전 영역으로 맵을 확대/축소합니다.	☑	☑	
	다음 영역으로 확대/축소	액션 이력에 있는 다음 영역으로 맵을 확대/축소합니다.	☑	☑	
	원본 해상도로 확대/축소	활성화된 래스터 레이어의 픽셀 1 개가 모니터 스크린의 픽셀 1 개와 동일한 크기가 되는 수준으로 맵을 확대/축소합니다.	☑	☑	☑

Settings ▸ 🔧 *Options* ▸ *Map tools* 메뉴 아래에 있는 *Zoom factor* 를 설정하면 확대/축소 도중의 축척 조정 습성을 정의할 수 있습니다. 동일 메뉴에서, 맵 캔버스 하단에서 사용할 수 있게 되는 *Predefined Scales* 목록도 설정할 수 있습니다.

맵 캔버스 상에서 마우스 사용

앞에서 설명한  Pan,  Zoom In,  Zoom Out 도구를 사용하는 방법 외에, 맵 캔버스 안에서 마우스 휠을 누르고 드래그하면 (맥 OS 에서는 `cmd` 키를 눌러야 할 수도 있습니다) 맵을 이동할 수 있습니다. 마우스 휠을 돌려서 맵을 확대하거나 축소할 수 있습니다. 마우스커서의 위치를 중심으로 관심 지역을 확대/축소할 것입니다. `Ctrl` 키를 누른 상태에서 마우스 휠을 돌리면 더 정밀하게 확대/축소할 수 있습니다.

맵 캔버스 상에서 키보드 사용

키보드의 `spacebar` 를 누른 채 마우스 커서를 움직이면  Pan 도구로 맵 캔버스를 드래그하는 것과 동일한 방식으로 맵을 이동시킬 것입니다.

방향 키를 이용해서 맵을 이동할 수 있습니다. 맵 영역 안쪽에 마우스 커서를 가져간 다음 방향 키를 누르면 위쪽, 아래쪽, 왼쪽, 오른쪽으로 이동합니다.

키보드의 `PgUp` 과 `PgDown` 키를 누르면 설정한 확대/축소 인자 (zoom factor) 에 따라 맵을 확대하거나 축소할 수 있습니다. 맵 캔버스 상에서 `Ctrl++` 또는 `Ctrl+-` 키 조합을 누르면 즉각적으로 맵을 확대/축소합니다.

특정한 맵 도구 (식별, 측정 등등) 를 활성화한 상태라면, `Shift` 키를 누른 상태로 맵 상에 사각형을 드래그해서 해당 지역으로 확대할 수 있습니다. 선택 도구 (선택 도구는 추가 선택에 `Shift` 키를 이용합니다) 나 편집 도구가 아닌 맵 도구에 한해 이 기능이 활성화됩니다.

11.4.3 공간 북마크

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 명칭으로) 저장할 수도 있습니다. 프로젝트 북마크는 프로젝트 파일 안에 저장되는데, 다른 사용자와 공유해야 하는 프로젝트인 경우 유용할 것입니다.

북마크 생성

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.

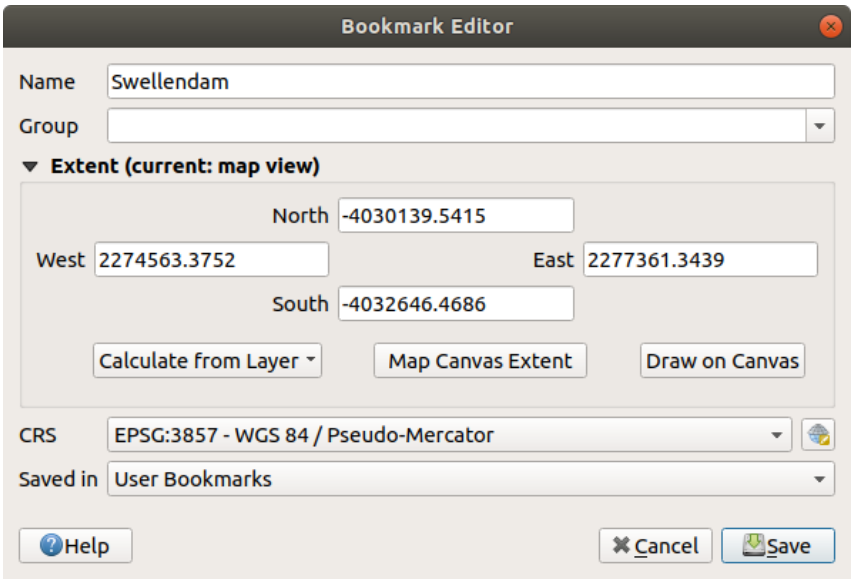


그림 11.6: 북마크 편집기 대화창

3. 북마크를 잘 설명하는 명칭을 입력하십시오.
4. 관련 북마크를 저장할 그룹명을 입력하거나 선택하십시오.
5. 범위 선택기를 사용해서 저장하고자 하는 영역의 범위를 선택하십시오. 불러온 레이어의 범위, 현재 맵 캔버스의 범위, 또는 현재 맵 캔버스 위에 마우스로 드래그한 사각형을 통해 범위를 계산할 수 있습니다.
6. 선택한 범위에 사용할 *CRS* 를 지정하십시오.
7. *Saved in* 옵션에서 북마크를 *User Bookmarks* 로 또는 *Project Bookmarks* 로 저장할지 선택하십시오.
8. *Save* 를 클릭하면 북마크가 목록에 추가됩니다.

북마크 여러 개를 동일한 명칭으로 만들 수 있다는 사실을 기억하십시오.

북마크 관련 작업

북마크를 사용하거나 관리하려면, *Spatial Bookmarks* 패널이나 *Browser* 를 사용하면 됩니다.

Spatial Bookmarks Manager 패널을 열려면 *View* → *Show Spatial Bookmark Manager* 메뉴 옵션을 선택하거나 `Ctrl+7` 키 조합을 누르십시오. *Browser* 패널에 있는 *Spatial Bookmarks* 항목을 보려면 *View* → *Show Bookmarks* 메뉴를 선택하거나 `Ctrl+Shift+B` 키 조합을 누르십시오.

다음과 같은 작업들을 수행할 수 있습니다:

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 명칭, 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. 폴더 (사용자 또는 프로젝트) 또는 하위 폴더 (그룹) 로 북마크를 드래그 & 드롭할 수도 있습니다.

위치 탐색란 에 북마크 명칭을 입력해서 북마크로 확대/축소할 수도 있습니다.

11.4.4 지도 장식

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.

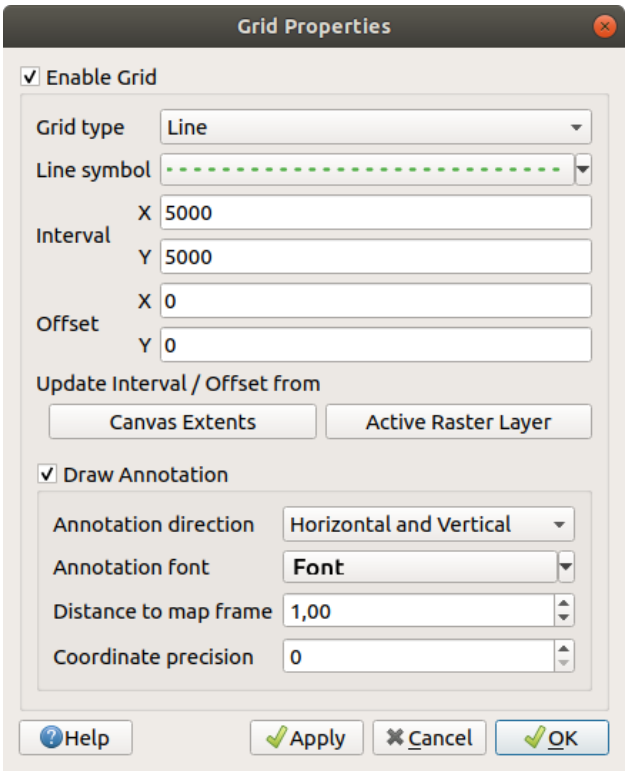


그림 11.7: 그리드 대화창

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.

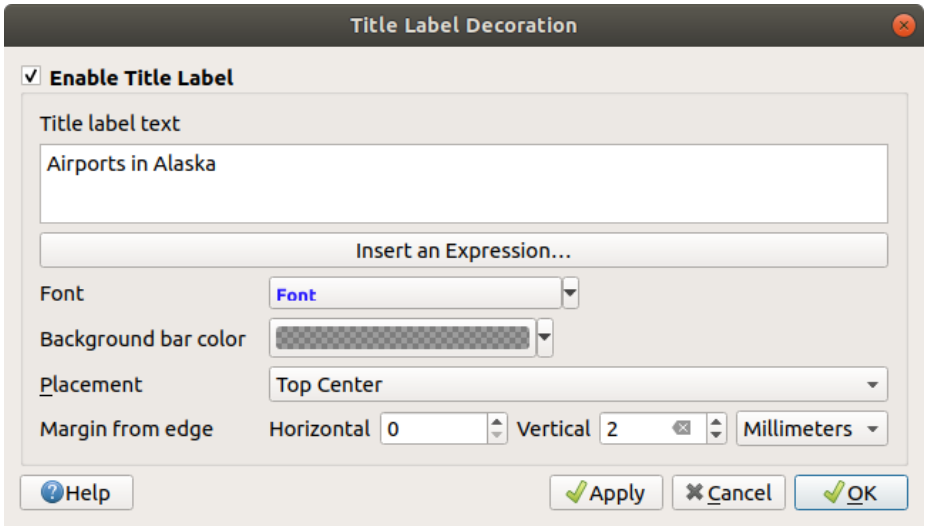


그림 11.8: 제목 장식 대화창

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.

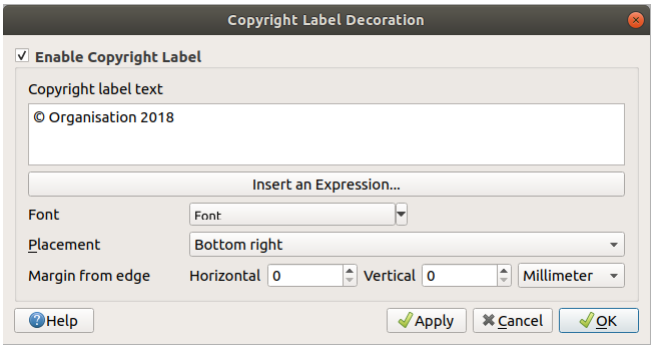


그림 11.9: 저작권 장식 대화창

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.

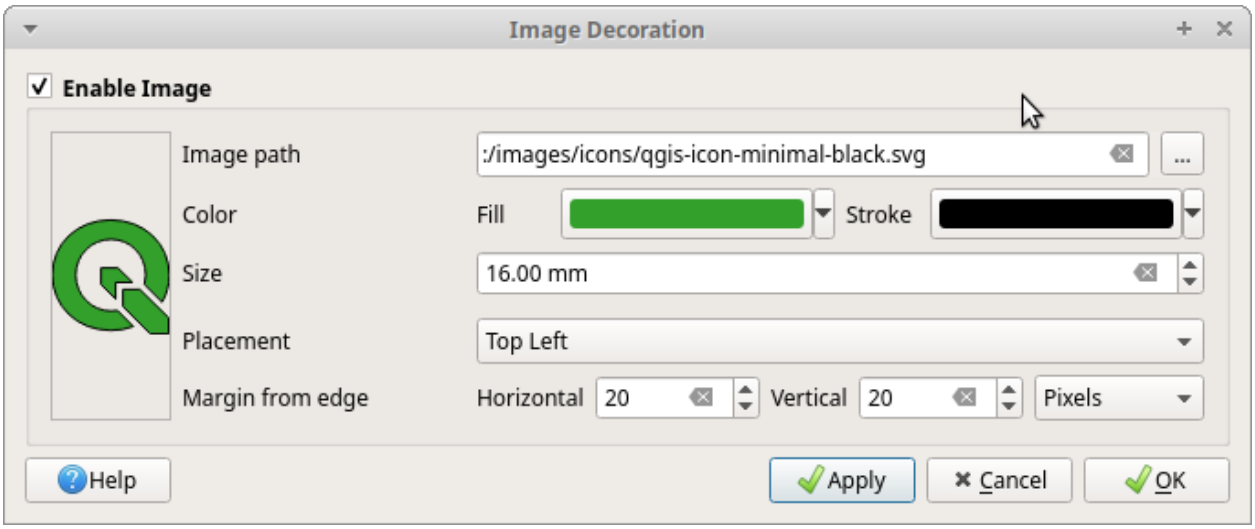


그림 11.10: 이미지 장식 대화창

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.

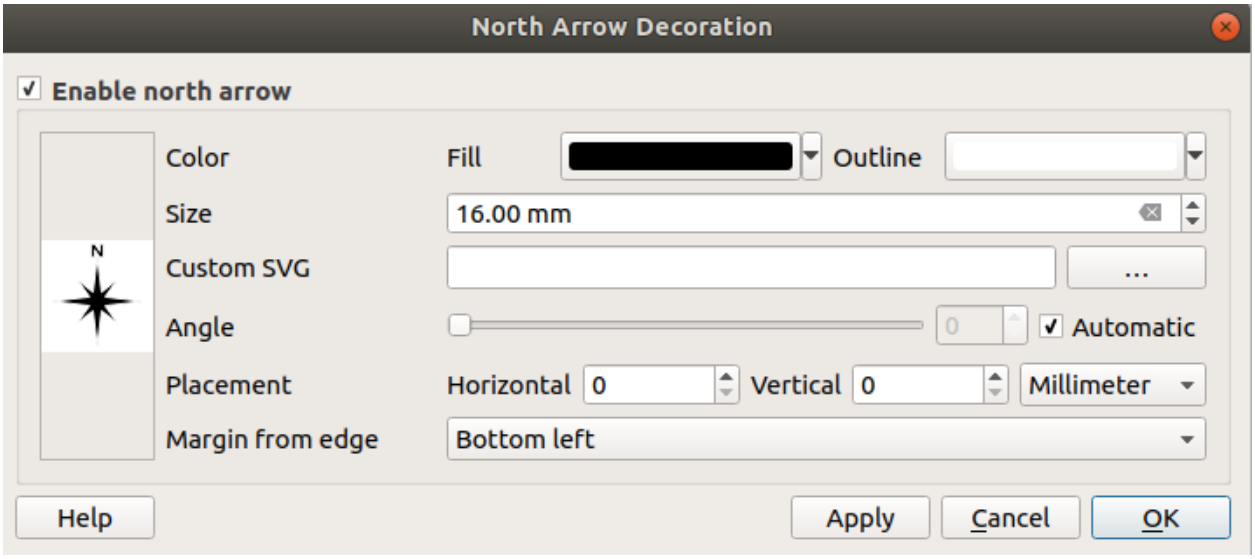


그림 11.11: 방위표 대화창

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 도구는 맵 캔버스에 단순한 축척 막대를 추가합니다. 사용자가 스타일과 위치는 물론 막대의 라벨도 제어할 수 있습니다.

QGIS 에서는 축척을 사용자 맵 프레임과 동일한 단위로만 표시할 수 있습니다. 따라서 사용자 프로젝트의 좌표계가 미터 단위라면, 축척 막대를 피트 단위로 생성할 수 없습니다. 마찬가지로, 사용자가 십진수도 (decimal degrees) 를 사용하고 있다면, 미터 단위로 거리를 표시하는 축척 막대를 생성할 수 없습니다.

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 설정하십시오.
7. *Automatically snap to round number on resize* 옵션을 체크하면 읽기 쉬운 값을 표시하도록 할 수 있습니다.

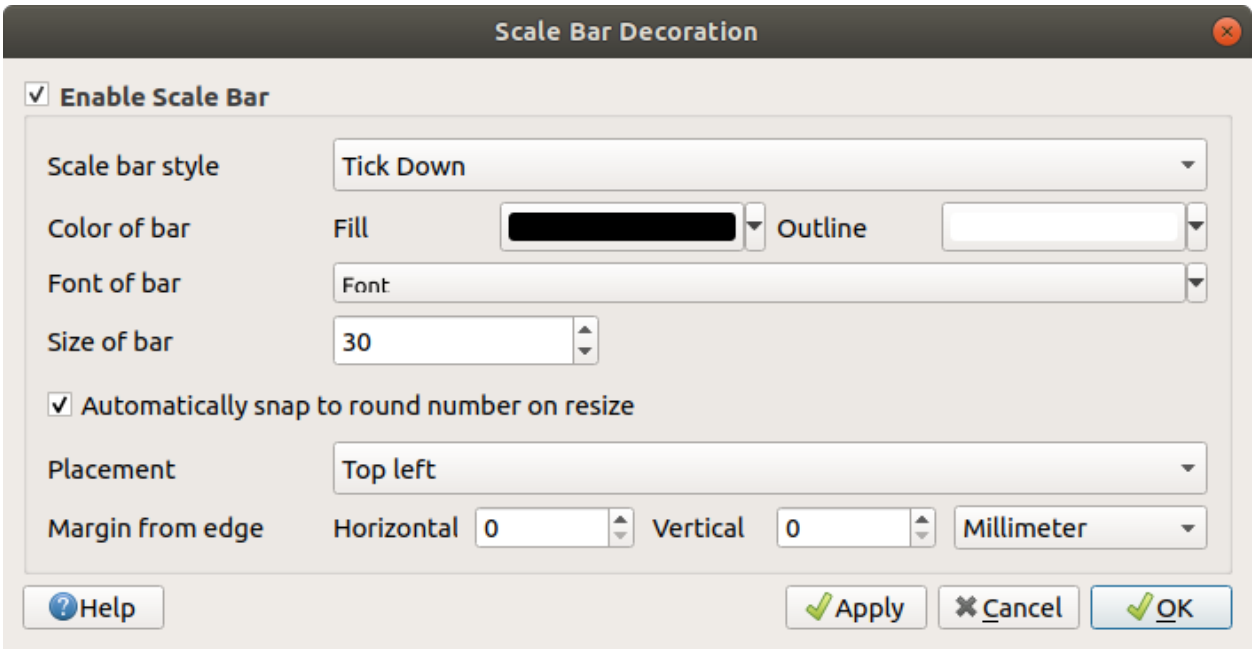


그림 11.12: 축척 막대 대화창

8. *Placement* 콤보박스에서 축척 막대의 배치를 선택하십시오.
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 명칭 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 11.13: 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.

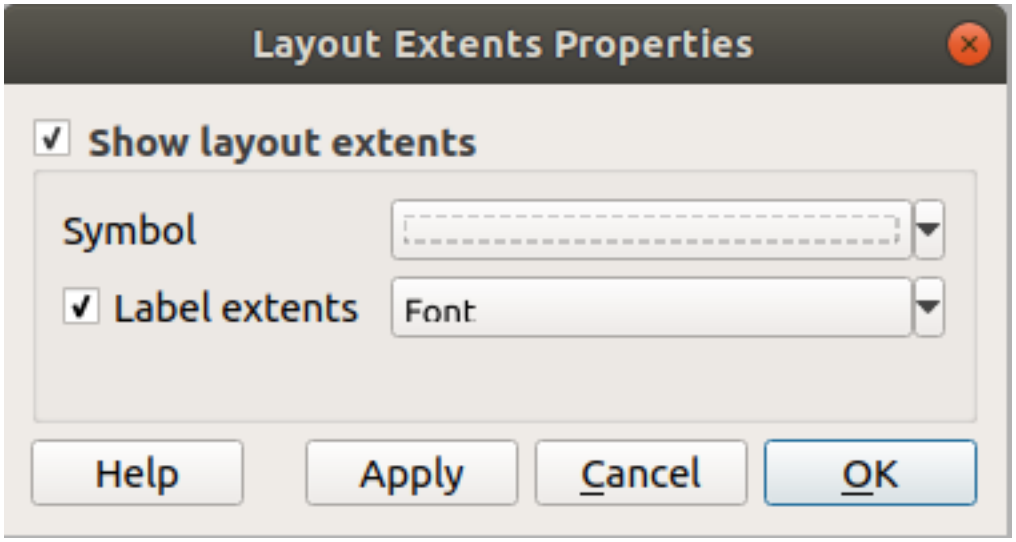


그림 11.14: 조판 범위 대화창

11.4.5 주석 도구

주석이란 맵 캔버스에 추가된 정보로 말풍선 안에 표시됩니다. 서로 다른 유형의 정보를 표시할 수 있으며, *Attributes Toolbar* 에 있는 서로 다른 유형에 대응하는 도구를 사용하면 해당 유형의 주석을 추가합니다:

- Text Annotation: 사용자 지정 서식 텍스트
- HTML Annotation: `html` 파일의 내용을 배치
- SVG Annotation: `SVG` 심볼을 추가
- Form Annotation: 사용자 지정 `ui` 파일에 있는 벡터 레이어의 속성을 표시하는 데 유용합니다. (그림 11.15 그림을 참조하세요.) 이는사용자 지정 속성 양식 과 비슷하지만, 주석 항목으로 표시된다는 점이 다릅니다. 더 자세한 정보를 알고 싶다면 팀 서튼이 게시한 https://www.youtube.com/watch?v=0pDBuSbQ02o&feature=youtu.be&t=2m25s 영상도 참조하세요.

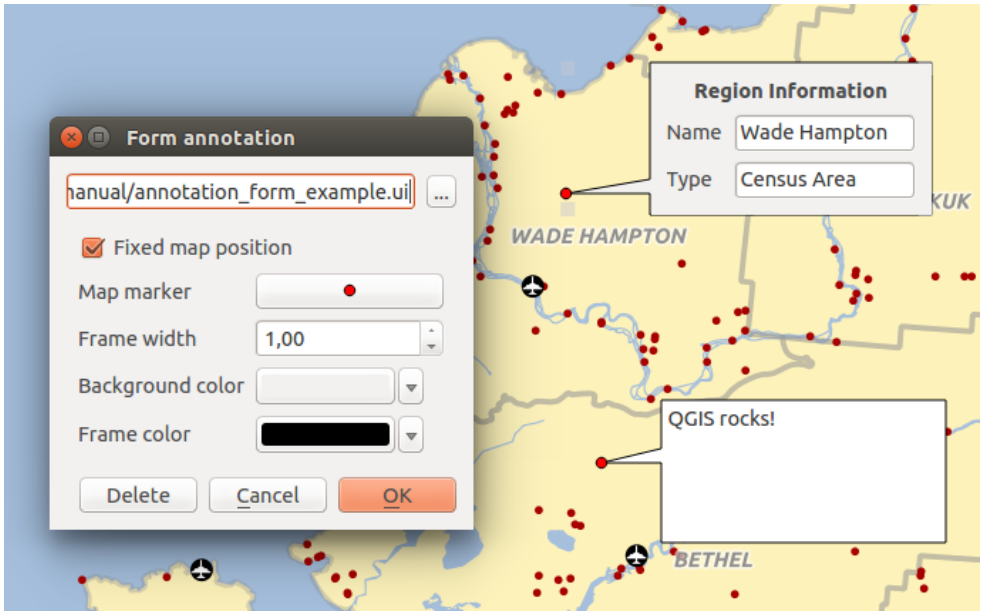


그림 11.15: 사용자 지정 QT 디자이너 주석 양식

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

- 주석 유형에 따라 `html`, `svg` 또는 `ui` 파일을 가리키는 경로를 입력하는 파일 선택기가 맨 위에 있습니다. 텍스트 주석의 경우, 사용자가 텍스트 란에 직접 메시지를 입력하고 일반 글꼴 도구로 텍스트의 렌더링 방법을 설정할 수 있습니다.
- *Fixed map position*: 이 옵션을 체크 해제하면, 말풍선을 (맵 위치 대신) 스크린 위치를 기반으로 배치합니다. 즉 맵 캔버스 범위와 상관없이 언제나 말풍선을 표시합니다.
- *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.

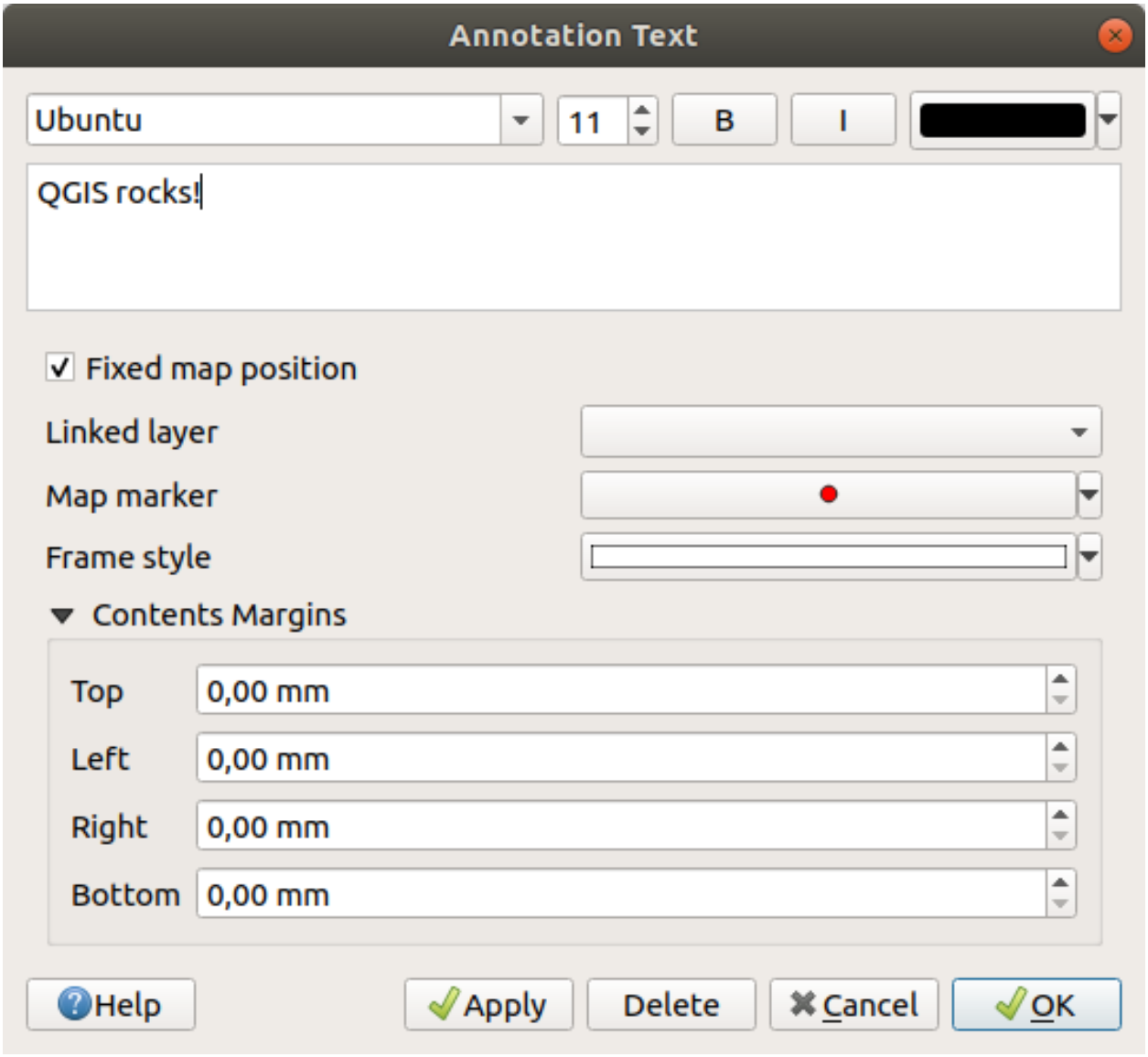


그림 11.16: 주석 텍스트 대화창

주석 도구를 활성화하면 주석을 선택할 수 있습니다. 선택한 주석의 맵 위치를 (맵 마커를 드래그해서) 옮기거나, 말풍선만 이동시킬 수도 있습니다.  Move Annotation 도구를 사용해서도 맵 캔버스 상에서 말풍선을 이동할 수 있습니다.

주석을 삭제하려면, 주석을 선택한 다음 `Del` 또는 `Backspace` 키를 누르거나 또는 주석을 더블 클릭한 다음 속성 대화창에 있는 *Delete* 버튼을 클릭하십시오.

참고: *Annotation* 도구가 활성화됐을 때 (주석 이동, 텍스트 주석, 양식 주석 등) `Ctrl+T` 를 누르면 말풍선의 가시성 상태가 반전됩니다.

팁: 주석으로 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.

11.4.6 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.

대화형 길이, 면적 및 각도 측정

측정을 시작하려면 *Attributes toolbar* 에 있는 아이콘을 클릭하십시오. 아이콘 옆에 있는 아래쪽 화살표를 누르면 길이, 면적, 또는 각도 측정 도구로 번갈아 변경합니다. 측정 대화창은 *Project* → *Properties…* → *General* 메뉴에서 설정한 기본 단위를 사용합니다.

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.

기본값인 Measure Line 은 지정한 포인트들 사이의 실제 거리를 정의한 타원체에 따라 측정합니다. 이 도구를 선택하면 사용자가 맵 상에 포인트를 클릭할 수 있습니다. 각 선분 길이는 물론 총합 길이도 측정 창에 표시됩니다. 측정을 중단하려면 오른쪽 클릭하십시오. 이제 *Copy All* 버튼을 사용하면 사용자의 모든 선분 측정값을 한번에 클립보드로 복사할 수 있습니다.

측정 도구 사용 중에 총합 길이 옆에 있는 드롭다운 목록에서 대화형으로 측정 단위를 ('미터', '킬로미터', '피트', '야드', '마일', '해리', '센티미터', '밀리미터', '도', '맵 단위') 변경할 수 있다는 사실을 기억하십시오. 위젯은 새로운 또는 다른 프로젝트를 열 때까지 이렇게 선택한 단위를 유지합니다.

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

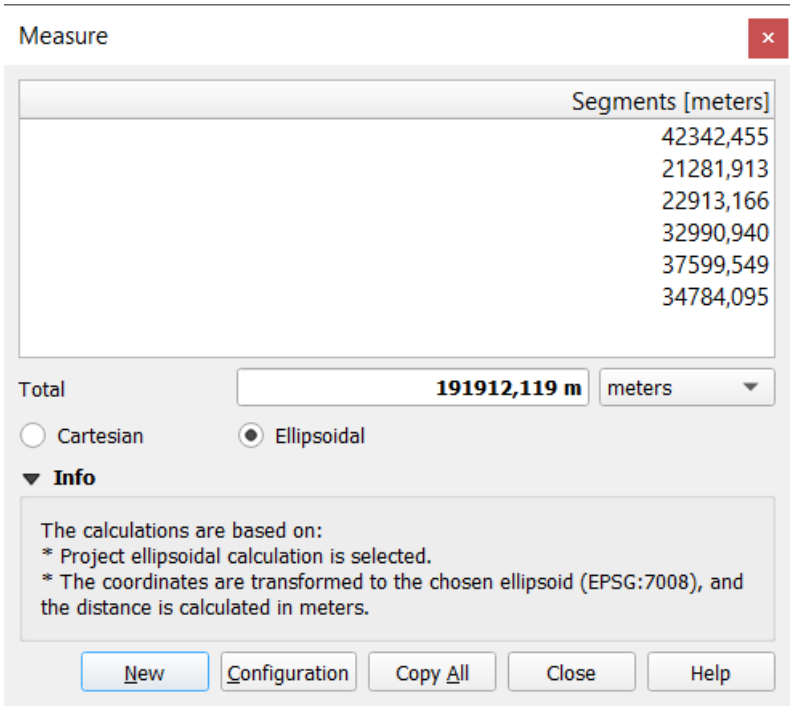


그림 11.17: 거리 측정

Measure Area 도구로 면적도 측정할 수 있습니다. 측정 창에 누적 면적 크기를 표시합니다. 오른쪽 클릭으로 그리기를 중단합니다. 서로 다른 면적 단위로 ('제곱미터', '제곱킬로미터', '제곱피트', '제곱야드', '제곱마일', '헥타르', '에이커', '제곱센티미터', '제곱밀리미터', '제곱해리', '제곱도', '맵 단위') 변경할 수 있는 것은 물론 *Info* 부분도 볼 수 있습니다.

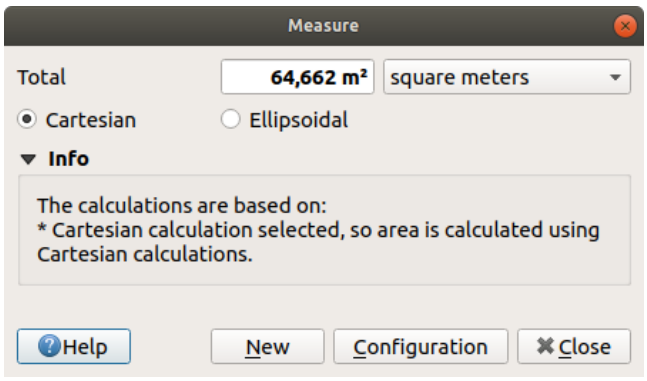


그림 11.18: 면적 측정

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

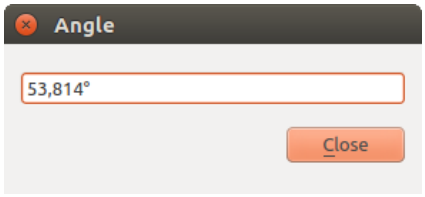


그림 11.19: 각도 측정

11.5 대화형 피처 작업

11.5.1 피처 선택

QGIS 는 맵 캔버스에서 피처를 선택할 수 있는 여러 도구를 제공하고 있습니다. *Edit* → *Select* 메뉴 또는:guilabel:*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 것들을 실수로 다른 곳을 클릭해버리는 바람에 선택 해제됐을 경우 매우 유용합니다.

 *Select Feature(s)* 도구 사용 시, `Shift` 또는 `Ctrl` 키를 누르면 피처를 선택할지 말지를 (예를 들면 현재 선택 집합에 추가할지 아니면 제거할지) 토글할 수 있습니다.

다른 도구들의 경우, 다음 키를 누르면 서로 다른 습성이 발현됩니다:

- `Shift`: 현재 선택 집합에 피처를 추가
- `Ctrl`: 현재 선택 집합에서 피처를 제거
- `Ctrl+Shift`: 현재 선택 집합과 교차, 예를 들어 현재 선택 집합에서 중첩하는 피처들만 유지
- `Alt`: 선택 형태에 완전히 들어가는 피처들만 선택. `Shift` 또는 `Ctrl` 키와 조합하면, 현재 선택 집합에 피처를 추가하거나 제거할 수 있습니다.

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 By Expression… 도구로 표현식 대화창을 이용해서 피처를 선택할 수 있습니다.
-  Select Features By Value… 도구 또는 `F3` 키
-  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)

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 필드에 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.

각 필드 옆에, 검색 습성을 제어할 수 있는 옵션을 나열한 드롭다운 목록이 있습니다:

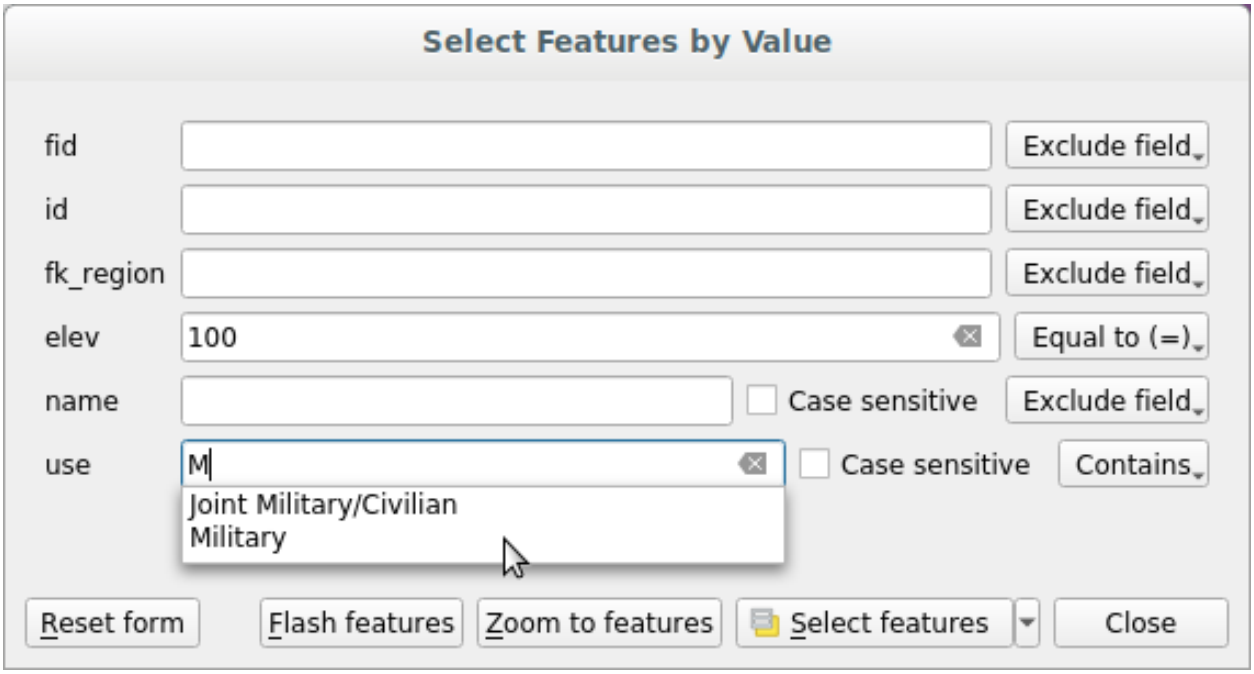


그림 11.20: 양식 대화창을 이용해 피처 필터링/선택

필드 검색 옵션	문자열	수치	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.

11.5.2 피처 식별

식별 도구를 통해 맵 캔버스와 상호작용해서 팝업창으로 피처에 대한 정보를 얻을 수 있습니다. 피처를 식별하려면, 다음 도구를 이용하십시오:

- *View* → *Identify Features* 메뉴
- `Ctrl+Shift+I` (또는 X `Cmd+Shift+I`) 키 조합
- 속성 툴바에 있는  Identify Features 아이콘

피처 식별 도구 이용하기

QGIS 는  Identify Features 도구로 피처를 식별할 수 있는 몇 가지 방법을 제공합니다:

- **왼쪽 클릭** 하면 *Identify Results* 패널에서 설정한선택 모드 및선택 마스크 에 따라 피처를 식별합니다.
- *Identify Results* 패널에서 설정한선택 모드 를 따르는 *Identify Feature(s)* 로 **오른쪽 클릭** 하면 모든 가시화된 레이어에서 모든 스냅된 피처를 가져옵니다. 이 방법은 컨텍스트 메뉴를 열어, 사용자가 식별할 피처를 또는 피처에 실행할 작업을 더 정밀하게 선택할 수 있도록 해줍니다.
- *Identify Results* 패널 에서 설정한선택 모드 를 따르는 *Identify Features by Polygon* 으로 **오른쪽 클릭** 하면 선택한 기존 폴리곤과 중첩하는 피처를, *Identify Results* 패널에서 설정한선택 마스크 에 따라 식별합니다.

팁: 쿼리할 레이어를 피처 식별 도구로 필터링하기

Project → *Properties…* → *Data Sources* 메뉴의 *Layer Capabilities* 에서 레이어 옆에 있는 *Identifiable* 열을 체크 해제하면, **Current Layer** 가 아닌 모드로  Identify Features 도구를 사용할 경우 해당 레이어를 쿼리 대상에서 제외합니다. 사용자가 관심을 가진 레이어에서만 피처를 반환받을 수 있는 간편한 방법입니다.

피처 (들) 를 클릭하면, 클릭한 피처 (들) 에 대한 정보 목록을 담은 *Identify Results* 대화창이 열립니다. 기본 뷰는 첫 번째 항목이 레이어명이고 그 하위 항목이 식별된 피처 (들) 인 트리 뷰입니다. 각 피처는 필드명을 그 값과 함께 표시합니다. *Layer Properties* → *Display* 메뉴에서 이 필드를 설정합니다. 그 다음 해당 피처에 대한 다른 모든 정보를 나열합니다.

피처 정보

사용자 지정 필드를 표출시키기 위해 식별 결과 대화창을 사용자 지정할 수도 있지만, 기본적으로 다음과 같은 유형의 정보를 표시할 것입니다:

- 피처의표시 명칭
- **Actions**: 피처 식별 창에 액션을 추가할 수 있습니다. 액션 라벨을 클릭해서 액션을 수행합니다. 기본적으로, 편집 작업을 위한 `View feature form` 이라는 액션 하나만 추가돼 있습니다. 레이어의 속성 대화창에서 더 많은 액션을 정의할 수 있습니다. (액션 속성 참조)
- **Derived**: 다른 정보로부터 계산된 또는 파생된 정보로, 다음을 포함합니다:
  - 피처의 도형과 관련된 일반 정보:
    - 도형 유형에 따라, 데카르트 평면에서 레이어 좌표계의 단위로 측정된 피처의 길이, 둘레 또는 면적. 3D 라인 벡터의 경우 데카르트 평면 상의 라인 길이를 구할 수 있습니다.
    - 프로젝트 속성 대화창에서 *Measurements* 용으로 설정된 타원체가 있는 경우, 도형 유형에 따라 타원체 상에서 지정한 단위로 측정된 피처의 길이, 둘레 또는 면적
    - 피처가 보유한 도형 부분의 개수와 클릭한 부분의 개수
    - 피처가 보유한 꼭짓점의 개수
  - 프로젝트 속성의 *Coordinates display* 설정을 사용한 좌표 정보:
    - 클릭한 포인트의 `X` 및 `Y` 좌표값
    - 클릭한 포인트에 가장 가까운 꼭짓점의 번호
    - 가장 가까운 꼭짓점의 `X` 및 `Y` 좌표값 (가능한 경우 `Z`/`M` 좌표값)
    - 곡선을 클릭한 경우, 해당 구간의 반경도 표시합니다.
- **Data attributes**: 사용자가 클릭한 피처의 속성 필드 및 값들의 목록입니다.
- 사용자가관계성 을 정의한 경우, 관계가 있는 파생 피처에 대한 정보:
  - 관계의 명칭
  - 참조 필드 항목, 예를 들어 관계가 있는 파생 피처의 명칭 등
  - **Actions**: 레이어의 속성 대화창에서 정의된 액션들의 목록 (액션 속성) 이며, 기본 액션은 `View feature form` 입니다.
  - **Data attributes**: 관계가 있는 파생 피처의 속성 필드 및 값들의 목록입니다.

참고: *Identify Results* 패널에서 피처 속성 내부에 있는 링크를 클릭할 수 있으며, 사용자 기본 웹브라우저로 열릴 것입니다.

식별 결과 대화창

창 상단에 여러 도구가 있습니다:

- Open Form: 현재 피처의 양식을 엽니다.
- Expand tree: 트리를 펼칩니다.
- Collapse tree: 트리를 접습니다.
- Expand New Results by Default: 다음 식별할 피처의 정보를 접을지 펼칠지 여부를 정의합니다.

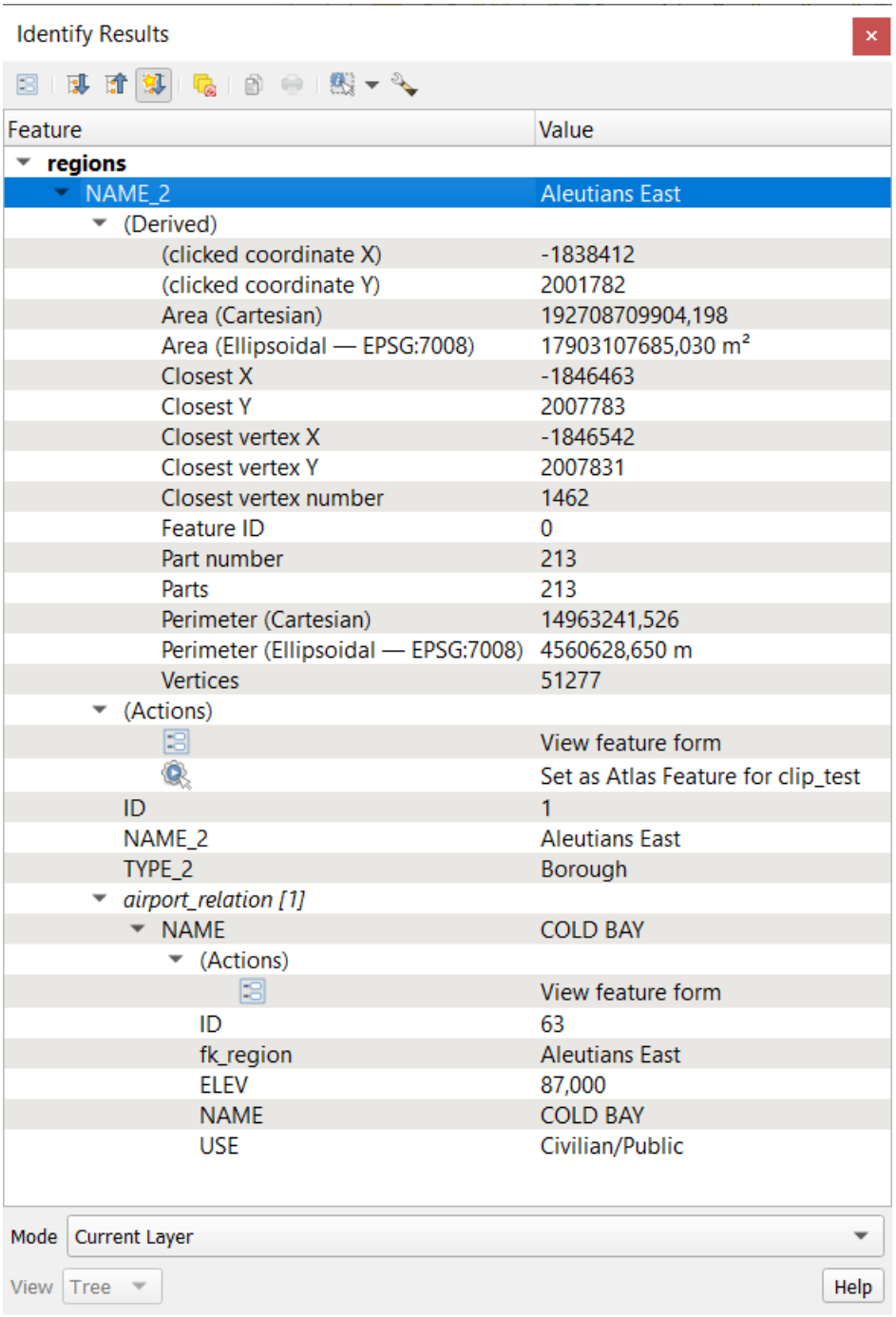


그림 11.21: 식별 결과 대화창

- Clear Results: 결과를 초기화합니다.
- Copy selected feature to clipboard
- Print selected HTML response
- 식별할 피처를 가져오기 위해 사용할 선택 모드:
  - Identify Features by area or single click
  - Identify Features by Polygon
  - Identify Features by Freehand
  - Identify Features by Radius

참고: Identify Features by Polygon 도구 사용 시, 모든 기존 폴리곤을 오른쪽 클릭하면 또다른 레이어의 중첩하는 피처를 식별하는 데 사용할 수 있습니다.

창 하단에는 *Mode* 와 *View* 콤보박스가 있습니다. *Mode* 콤보박스를 통해 어떤 레이어에 있는 피처를 식별해야 하는지 정의할 수 있습니다:

- **Current layer**: 선택한 레이어에 있는 피처만 식별합니다. 레이어가 맵 캔버스에 보이지 않아도 됩니다.
- **Top down, stop at first**: 가시화된 레이어 가운데 가장 위에 있는 레이어의 피처만 식별합니다.
- **Top down**: 가시화된 레이어들의 모든 피처를 식별합니다. 식별 결과가 패널에 표시됩니다.
- **Layer selection**: 컨텍스트 메뉴를 열어서 사용자가 피처를 식별할 레이어를 선택할 수 있습니다. 오른쪽 클릭과 동일하게 동작합니다. 선택한 피처만 식별 결과 패널에 표시될 것입니다.

View 를 **트리**, **테이블** 또는 **그래프** 로 설정할 수 있습니다. 래스터 레이어에 대해서만'테이블'과'그래프'를 설정할 수 있습니다.

식별 도구에서 Identify Settings 아래 있는 *Auto open form for single feature results* 를 이용할 수 있습니다. 이 체크박스를 체크하면, 피처를 하나 식별할 때마다 QGIS 가 해당 피처의 속성을 보여주는 양식을 엽니다. 피처 속성을 빠르게 편집할 수 있는 편리한 방법입니다.

식별된 항목의 컨텍스트 메뉴에서 다른 기능도 찾아볼 수 있습니다. 예를 들면, 컨텍스트 메뉴를 통해 다음 작업을 할 수 있습니다:

- 피처 양식 보기
- 피처로 확대/축소
- 피처 복사: 모든 피처 도형 및 속성 복사
- 피처 집합 켜고끄기: 식별한 피처를 선택 집합에 추가
- 속성값 복사: 사용자가 클릭한 속성값만 복사
- 피처 속성 복사: 피처의 속성을 복사
- 결과 삭제: 창에서 식별 결과를 제거
- 강조 삭제: 맵 상에서 강조된 피처를 제거
- 모든 피처 강조
- 레이어 강조
- 레이어 활성화: 활성화할 레이어를 선택

- 레이어 속성: 레이어 속성 창 열기
- 모두 확장
- 모두 접기

11.6 레이어 속성 저장 및 공유

11.6.1 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.

스타일은 벡터 레이어의 경우 피처 (심볼, 라벨, 필드 및 양식 정의, 액션, 다이어그램 등등의 설정), 래스터 레이어의 경우 픽셀 (밴드 또는 색상 렌더링, 투명도, 피라미드, 히스토그램 등등) 을 렌더링하거나 상호작용하기 위해 레이어 속성 대화창에서 설정한 모든 정보를 저장합니다.

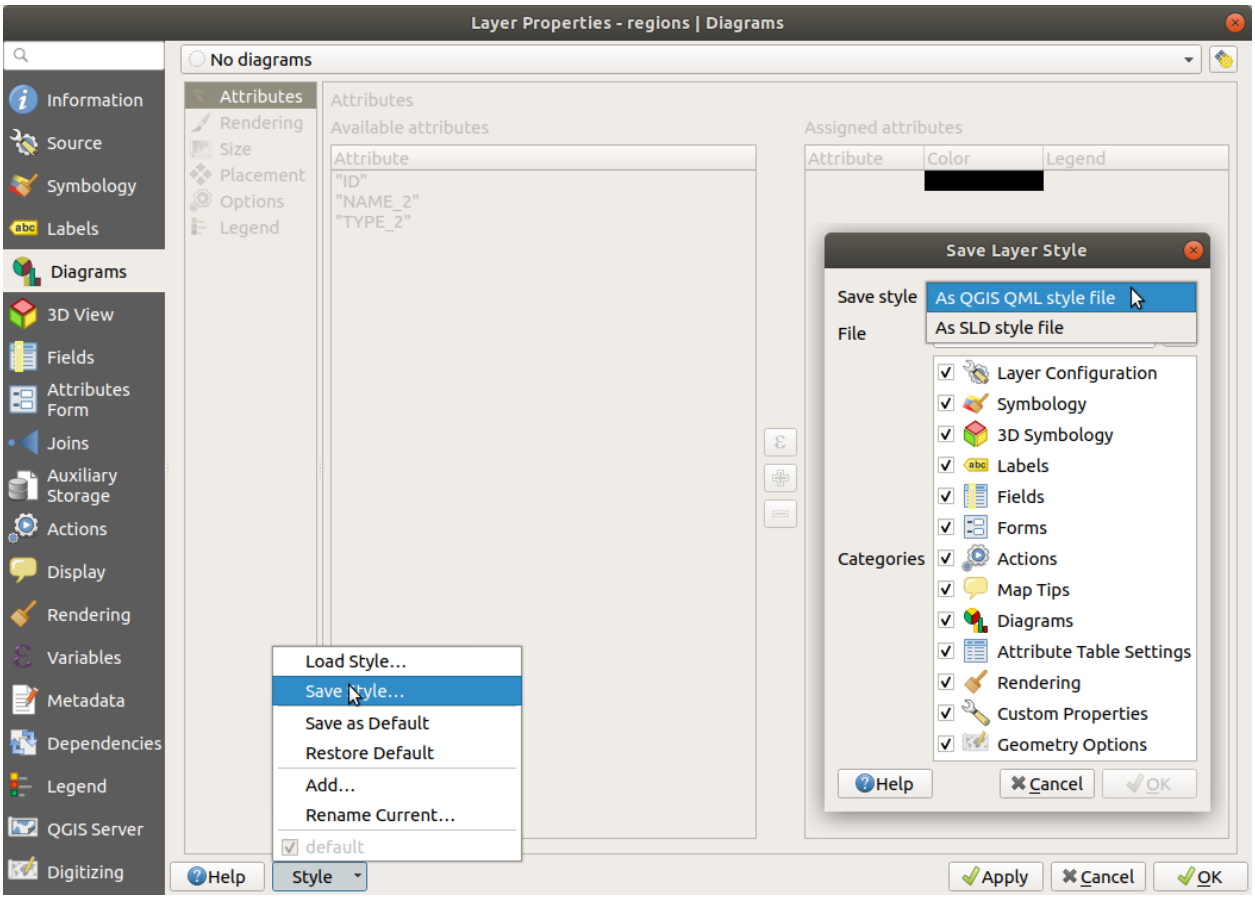


그림 11.22: 벡터 레이어 스타일 콤보박스 옵션들

불러온 레이어에 적용되는 스타일의 기본 명칭은 `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.
- **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

Layers 패널에 있는 레이어를 오른쪽 클릭하면 레이어 스타일을 복사, 붙여넣기, 추가 또는 재명명할 수 있습니다.

11.6.2 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 명칭으로) `.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.

데이터베이스에 저장

레이어 데이터소스가 데이터베이스 제공자인 경우 벡터 레이어 스타일도 데이터베이스에 저장할 수 있습니다. PostGIS, GeoPackage, SpatiaLite, MS SQL 및 Oracle 포맷을 지원합니다. 데이터베이스의 (테이블명 `layer_styles`) 테이블에 레이어 스타일을 저장합니다. *Save Style…* ▸ *Save in database* 메뉴를 클릭한 다음 대화창에서 스타일 명칭을 정의하고, 설명을 추가하고, 가능한 경우 `.ui` 파일을 추가하고 해당 스타일을 기본 스타일로 설정할지 체크하십시오.

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.

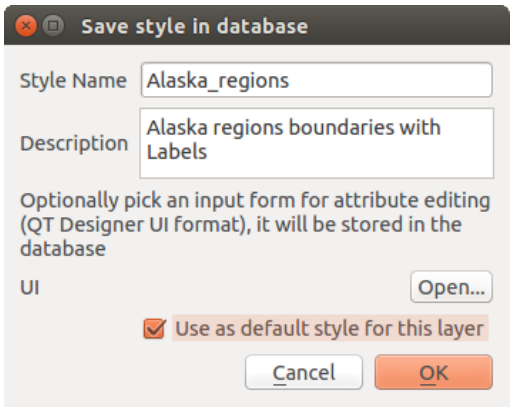


그림 11.23: 데이터베이스 대화창에서 스타일 저장

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 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 명칭 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.

11.6.3 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.

11.7 값을 변수로 저장

QGIS 는 표현식에 쓰일 수 있는 자주 사용되는 유용한 값들 (예를 들어 프로젝트 제목 또는 사용자의 이름) 을 변수를 이용해서 저장할 수 있습니다. 응용 프로그램 전체 수준, 프로젝트 수준, 레이어 수준, 조판 수준그리고 조판기 항목 수준에서 변수를 정의할 수 있습니다. CSS 의 순차적 (cascading) 규칙처럼, 변수도 덮어쓸 수 있습니다. 예를 들어, 프로젝트 수준의 변수는 동일한 명칭을 가진 모든 응용 프로그램 수준의 변수를 덮어쓸 것입니다. 변수명 앞에 `@` 문자를 붙여서 텍스트 문자열 또는 다른 사용자 지정 표현식을 작성하는 데 쓸 수 있습니다. 다음은 인쇄 조판기에서 라벨을 생성하는 예시입니다:

```
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)
- 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 명칭 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.

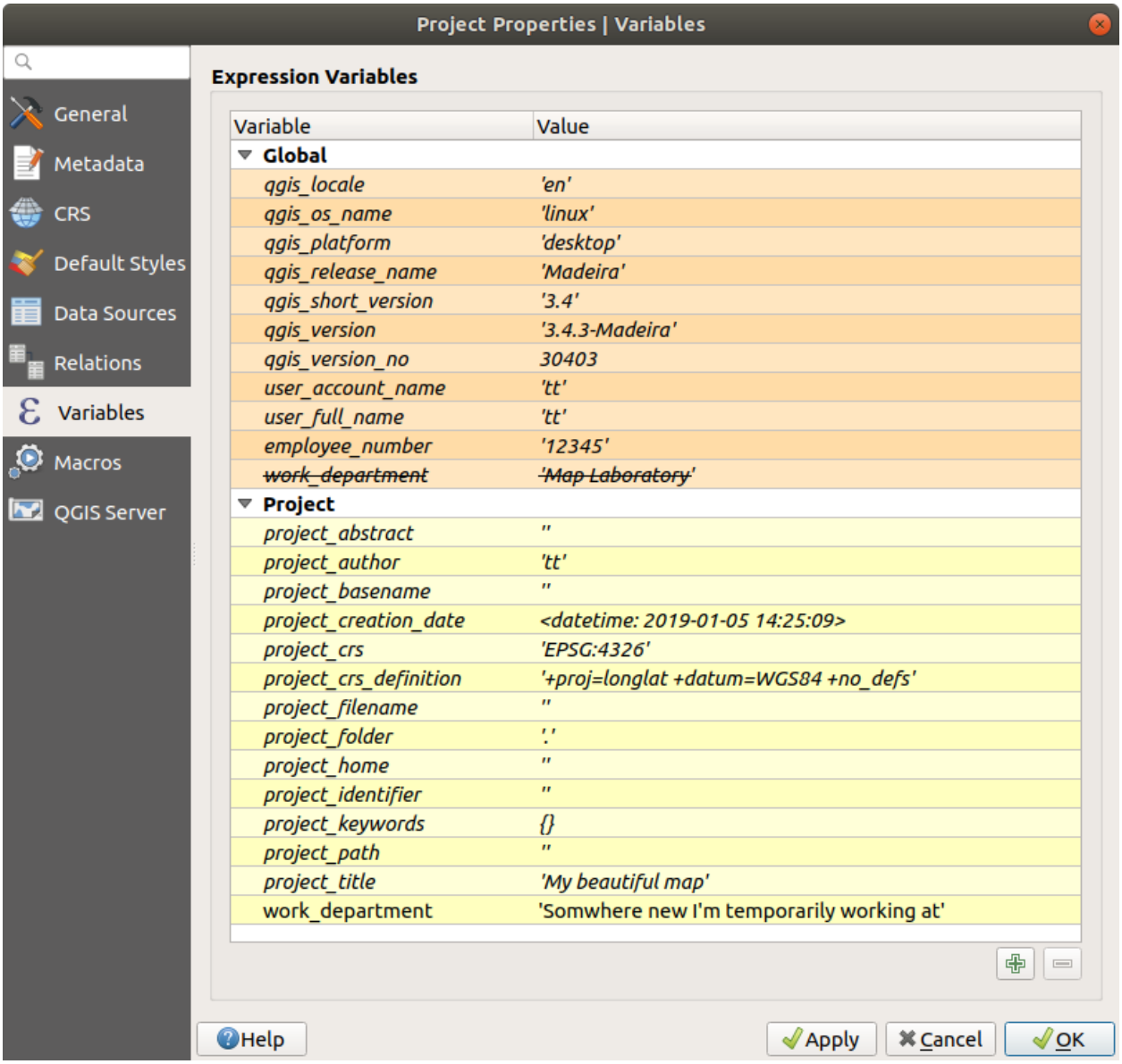


그림 11.24: 프로젝트 수준 변수 편집기

11.8 인증

QGIS 에는 안전한 방식으로 인증서 (authentication credentials) 를 저장/검색할 수 있는 기능이 있습니다. 사용자는 인증 환경 설정 안에 인증서를 안전하게 저장할 수 있는데, 인증 환경 설정은 자체 데이터베이스에 저장되어 서버 또는 데이터베이스 연결에 적용할 수 있으며 프로젝트 또는 설정 파일에서 ID 토큰으로 안전하게 참조할 수 있습니다. 더 자세한 정보를 원한다면인증 시스템 을 참조하세요.

인증 시스템 및 그 자체 데이터베이스를 처음 실행할 때 마스터 비밀번호를 설정해줘야 합니다.

11.9 일반 위젯

사용자가 QGIS 에서 자주 사용해야 할 옵션이 몇 개 있습니다. QGIS 는 사용자 편의를 위해 다음에 소개하는 특별 위젯들을 제공합니다.

11.9.1 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.

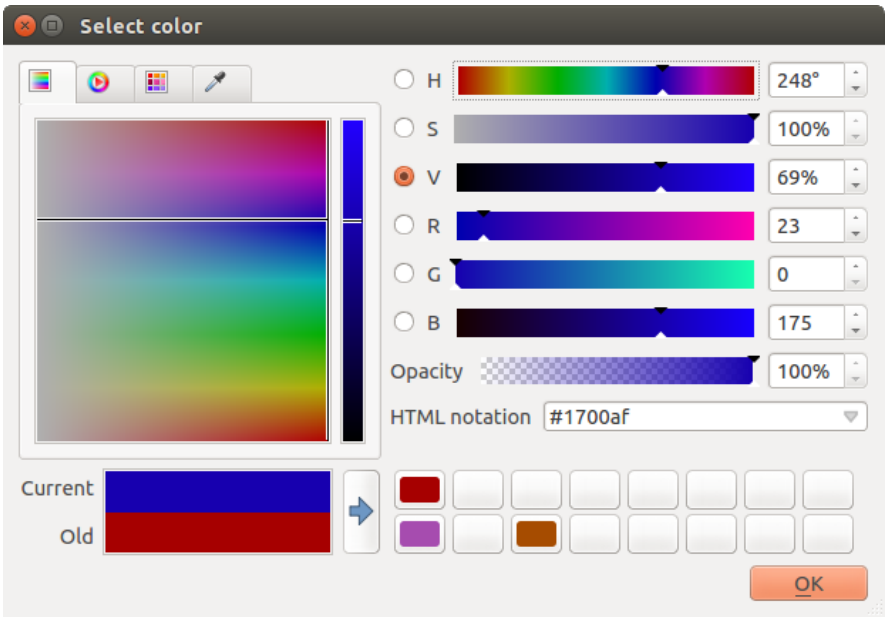


그림 11.25: 색상 선택기의 색상표 탭

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 11.27 그림 참조) 에 사용자 지정 색상표를 추가합니다.

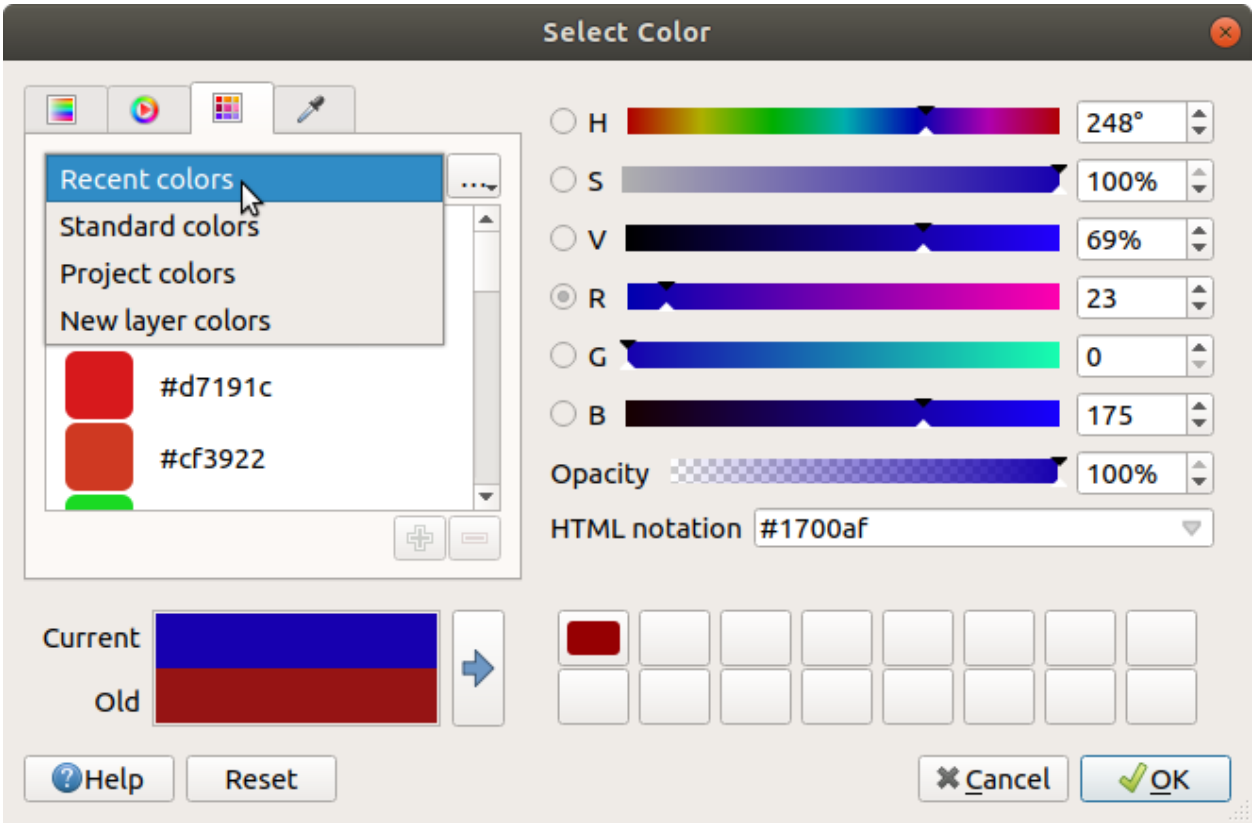


그림 11.26: 색상 선택기의 색상 견본표 탭

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.

어떤 방법을 이용하든, 항상 HSV (색상 hue, 채도 saturation, 명도 value) 와 RGB (적 red, 녹 green, 청 blue) 값에 대한 색상 슬라이드 바를 통해 선택한 색상을 설명합니다. *HTML notation* 으로도 색상을 식별할 수 있습니다.

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

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
- *Style Manager* 대화창에서 **Favorites** 로 표시된 `gradient` 또는 `catalog: cpt-city` 색상표의 미리보기
- 호환 가능한 색상표 데이터베이스에 접근할 수 있는 *All Color Ramps*
- 현재 위젯에서 사용할 수 있는 모든 지원 색상표 유형을 생성하는 *Create New Color Ramp*… (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

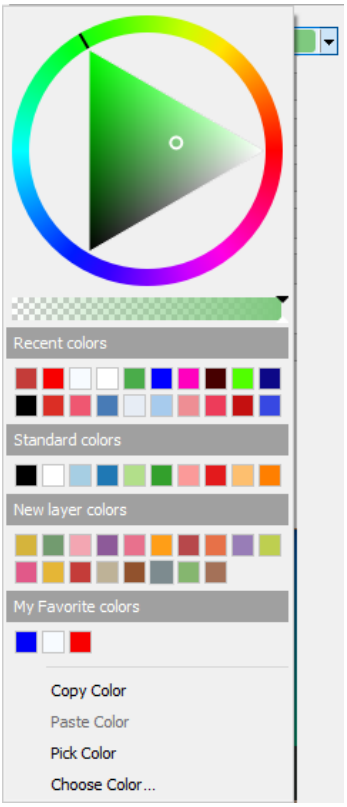


그림 11.27: 빠른 색상 선택기 메뉴

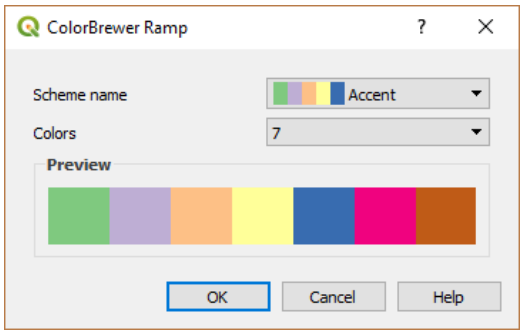


그림 11.28: 컬러브루어 (colorbrewer) 색상표 사용자 지정하기

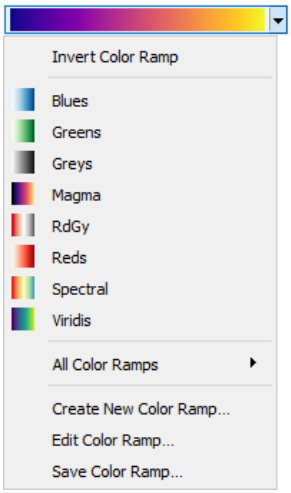


그림 11.29: 빠른 색상표 선택 위젯

11.9.2 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 환경 설정할 수 있습니다.

11.9.3 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:

- *Font Size*: 관련 단위로 된 글꼴 크기
- *Recent Fonts* : (맨 위에) 활성화 글꼴이 체크되어 있는 최근 사용 글꼴 목록
- *Configure Format*…: 글꼴 선택기 위젯 버튼을 클릭하는 것과 동일합니다. 텍스트 서식 파라미터를 설정하는 대화창을 엽니다. 맥락에 따라, OS 의 기본 *Text format* 대화창이 될 수도 있고 또는라벨 텍스트 서식 설정 절에서 설명하는 고급 서식 작업 옵션 (투명도, 기울기, 버퍼, 배경, 그림자 등등) 을 가진 QGIS 의 사용자 지정 대화창이 될 수도 있습니다.
- *Copy Format*: 텍스트의 서식을 복사합니다.
- *Paste Format*: 텍스트에 클립보드의 서식을 붙여넣습니다. 텍스트 서식을 빠르게 환경 설정할 수 있습니다.
- 색상 위젯: 색상을 쉽게 설정할 수 있습니다.



그림 11.30: 글꼴 선택기 드롭다운 메뉴

11.9.4 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*
- *Meters at Scale*: 이 옵션을 선택하면 기저 맵 단위가 (인치, 피트, 지리도 (geographic degree) 등등) 무엇이든, 항상 크기를 미터 단위로 설정할 수 있습니다. 현재 프로젝트 타원체 설정을 기반으로 크기를 미터 단위로 계산하며, 현재 맵 범위의 중심에서 거리의 투영을 미터 단위로 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.

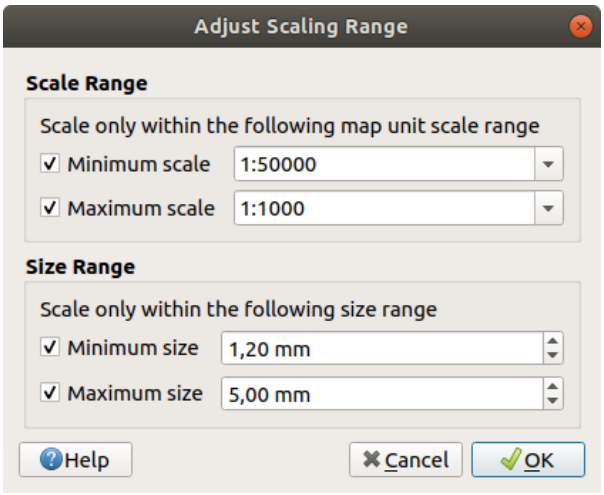


그림 11.31: 축척 범위 조정 대화창

11.9.5 숫자 서식 작업

숫자 서식기를 사용하면, 서로 다른 다양한 서식 작업 기술을 사용해서 (예를 들면 과학적 표기법, 통화 가치, 백분율 값 등등) 표시할 숫자값의 서식을 설정할 수 있습니다. 이를 사용해서, 한 예로, 조판의 축척 막대 또는 고정 표에 들어갈 텍스트의 서식을 설정할 수 있습니다.

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.
- *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.

11.9.6 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.

11.9.7 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, 유효한 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*: 유효한 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 환경 설정된 data-defined override 버튼을 활성화시키거나 비활성화시킬 수 있습니다.

데이터 정의 어시스턴트 인터페이스 사용하기

Data-defined override 버튼이 수치 또는 색상 파라미터와 연결되어 있는 경우, *Assistant*…옵션을 사용해서 각 피처의 파라미터에 데이터를 어떻게 적용할지를 변경할 수 있습니다. 어시스턴트는 다음과 같은 작업을 할 수 있습니다:

- *Input* 데이터를 정의할 수 있습니다. 예를 들어:
  - 필드 목록 상자 또는 Set column expression 기능을 사용해서 표현할 속성 (표현식 참조)
  - 표현할 값의 범위: 사용자가 직접 입력하거나 Fetch value range from layer 버튼을 눌러 선택한 속성 또는 사용자 데이터에 적용된 표현식이 반환한 최소최대값으로 필드를 자동적으로 채울 수 있습니다.
- *Apply transform curve*: 기본적으로 산출값은 선형 척도를 따라 입력 피처에 적용되지만, 이 로직을 다음과 같이 무시할 수 있습니다: 변환 옵션을 활성화하고, 그래픽을 클릭해서 중단점 (break point)(들) 을 추가한 다음 해당 중단점 (들) 을 드래그해서 사용자 지정 분포를 적용하십시오.
- *Output* 값을 정의할 수 있습니다: 이 옵션은 정의할 파라미터에 따라 달라집니다. 다음 전체 수준 값을 설정할 수 있습니다:
  - 선택한 속성에 적용할 최소최대값 (색상 설정 작업 시색상표 를 제공해야 합니다.)
  - *Scale method*: 표현에 사용할 척도 방법을 **Flannery**, **Exponential**, **Surface** 또는 **Radius** 가운데 선택할 수 있습니다.
  - *Exponent*: 데이터 척도에 사용할 지수를 정의할 수 있습니다.
  - NULL 값을 가진 피처를 표현할 산출값 또는색상 을 정의할 수 있습니다.

적용이 가능한 속성인 경우, 대화창 오른쪽에 실시간으로 업데이트되는 미리보기를 표시해서 사용자가 값 조정 작업을 제어할 수 있도록 해줍니다.

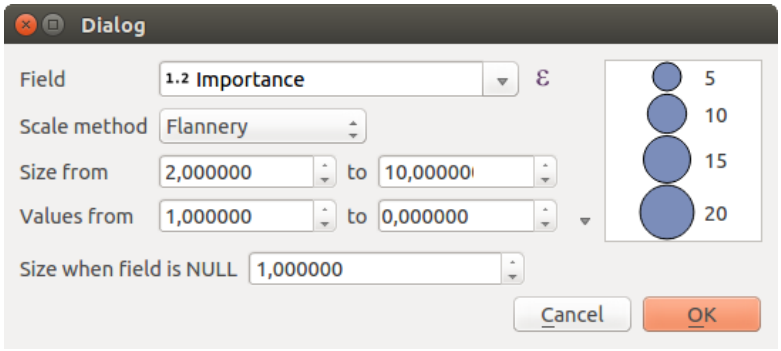


그림 11.32: 데이터 정의 크기 어시스턴트

위의 변화하는 크기 어시스턴트에 나온 값들은 다음 표현식으로'데이터 정의 무시'크기를 설정할 것입니다:

```
coalesce(scale_exp(Importance, 1, 20, 2, 10, 0.57), 1)
```

CHAPTER 12

스타일 라이브러리

12.1 스타일 관리자

12.1.1 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

스타일 항목 정리

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.

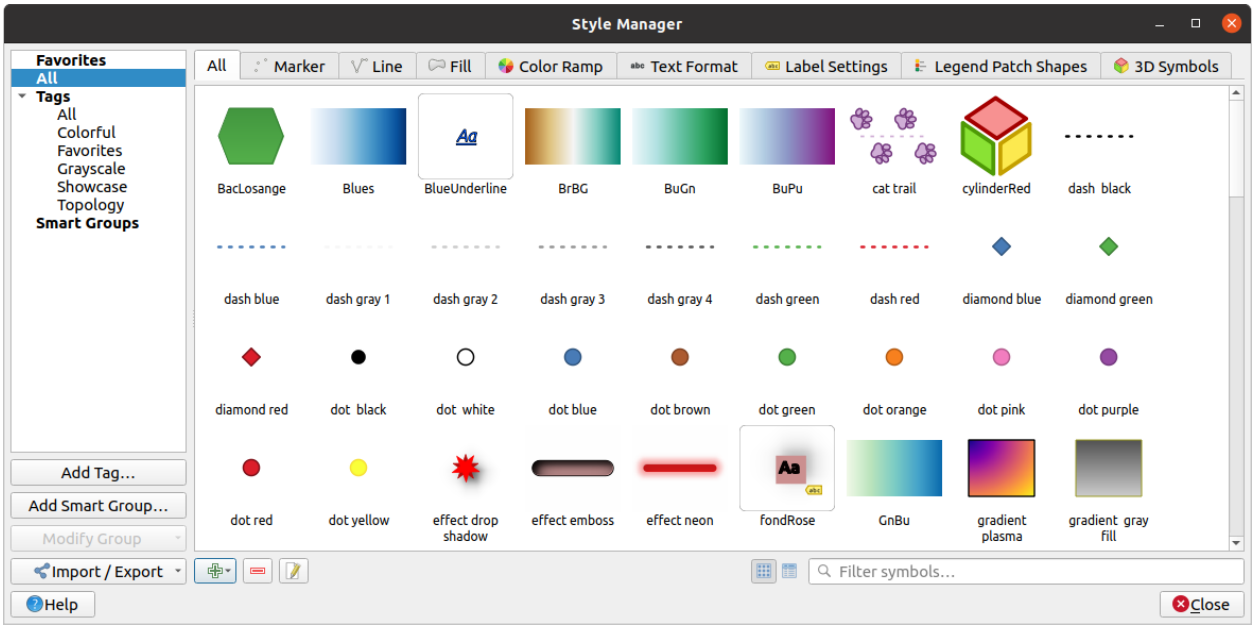


그림 12.1: 스타일 관리자

- *Color ramp* 탭은 색상표를 표시합니다.
- abc *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.
- *3D Symbols* 탭은*3D* 맵 뷰 에서 렌더링될 피처를 위해*3D* 속성 (돌출 (extrusion), 음영, 표고 등등) 으로 심볼을 환경 설정합니다.

각 항목 분류 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 12.2 그림을 참조하세요.) 스마트 그룹을 생성하려면 *Add Smart Group*…버튼을 클릭하십시오. 대화창에서 선택할 항목을 필터링할 (특정 태그가 붙어 있거나, 명칭에 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 명칭을 선택하거나 새 태그를

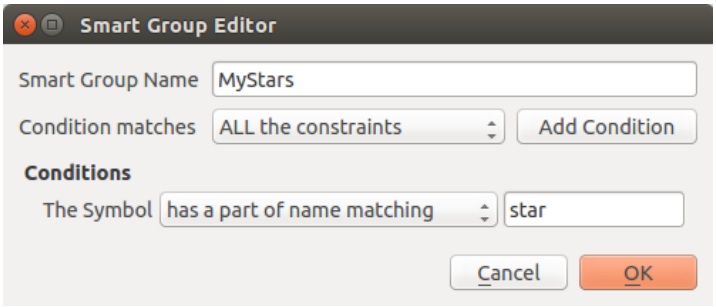


그림 12.2: 스마트 그룹 생성하기

선택하십시오.

- 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.

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.

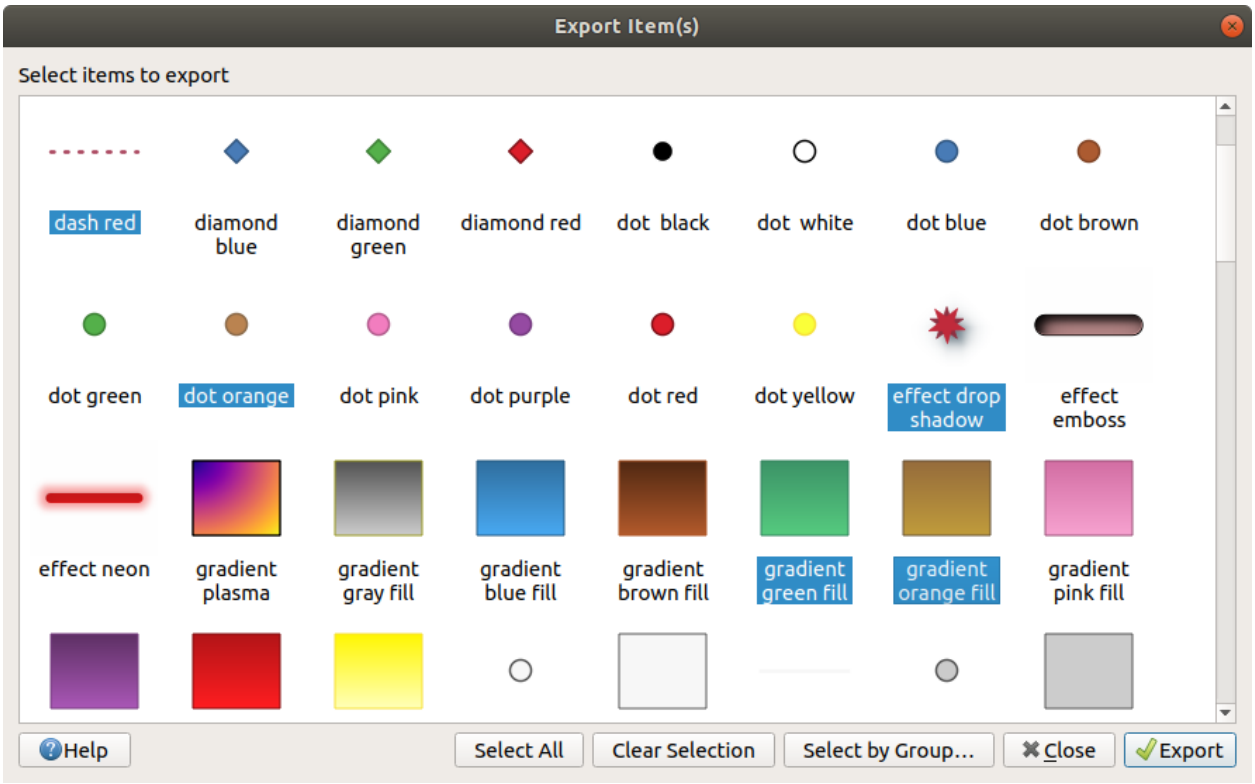


그림 12.3: 스타일 항목 내보내기

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 명칭을 정의합니다.
6. 미리보기에서 사용자 라이브러리에 추가하고자 하는 심볼을 선택하십시오.
7. 마지막으로 *Import* 를 클릭하십시오.

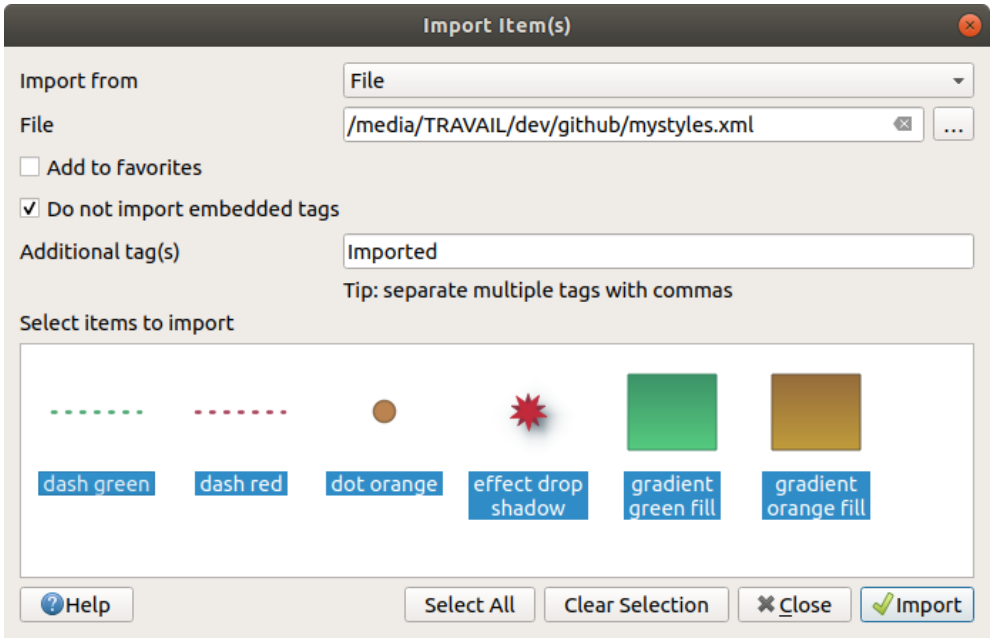


그림 12.4: 스타일 항목 가져오기

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.

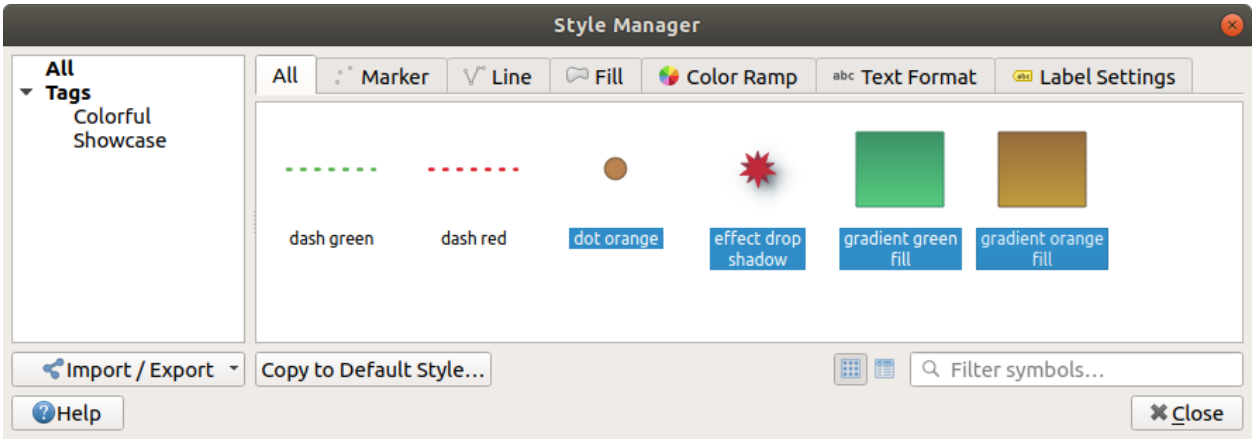


그림 12.5: 스타일 항목 파일 열기

이 대화창에서 단일 심볼을 `.PNG` 또는 `.SVG` 파일로 내보낼 수도 있습니다.

12.1.2 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 및 끝 색상을 설정해서 **연속적 (continuous)** 일수도 있고 **비연속적 (discrete)** 일수도 있는 색상표를 생성합니다. 색상표 미리보기를 더블 클릭하면 중간 색상 지점 (intermediatecolor stop) 을 사용자가 원하는 만큼 추가할 수 있습니다.
- *Color presets*: 사용자가 선택한 색상 목록으로 이루어진 색상표를 생성할 수 있습니다.
- *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.

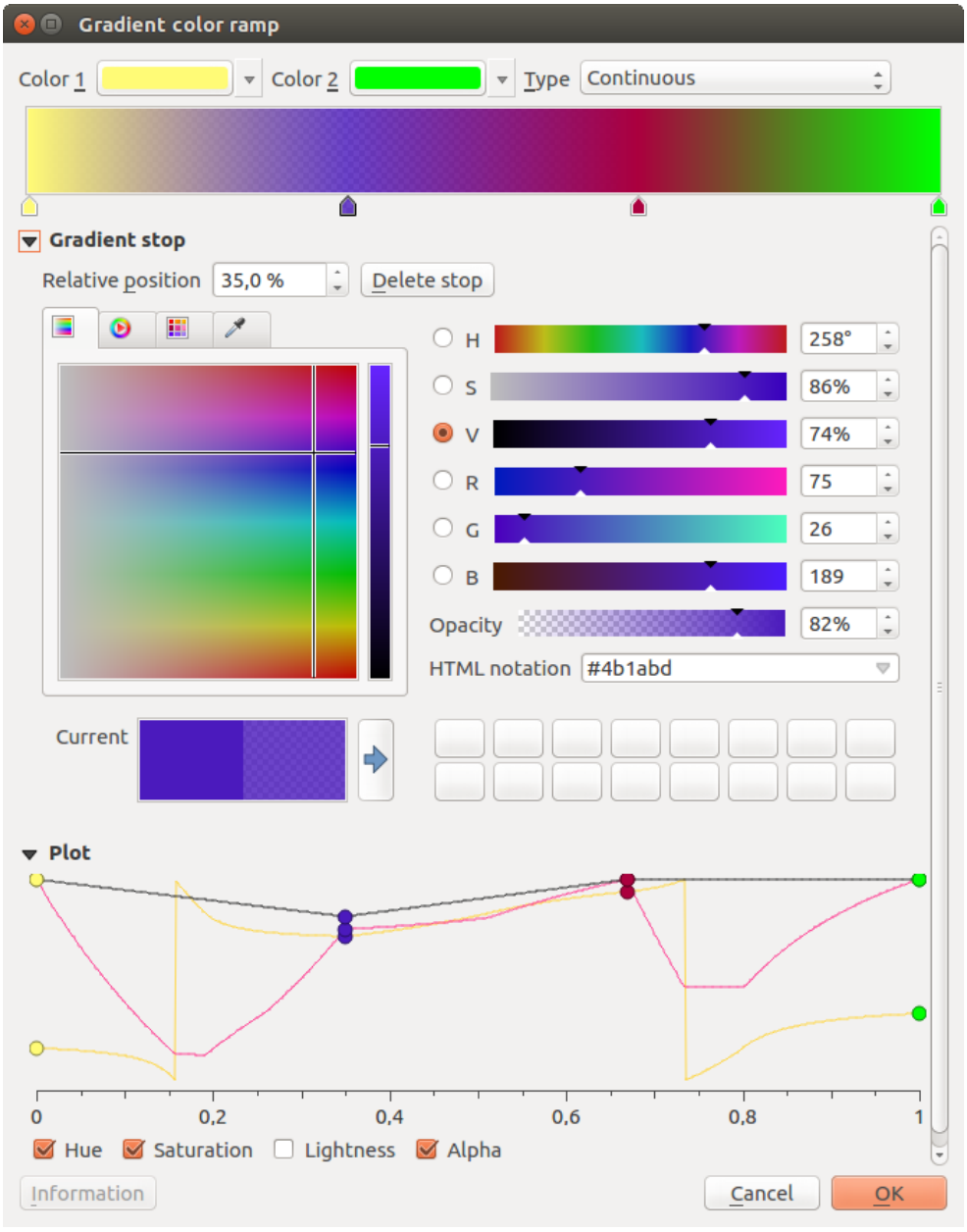


그림 12.6: 복수의 색상 지점을 지정한 사용자 지정 그레이디언트 색상표의 예시

- *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.

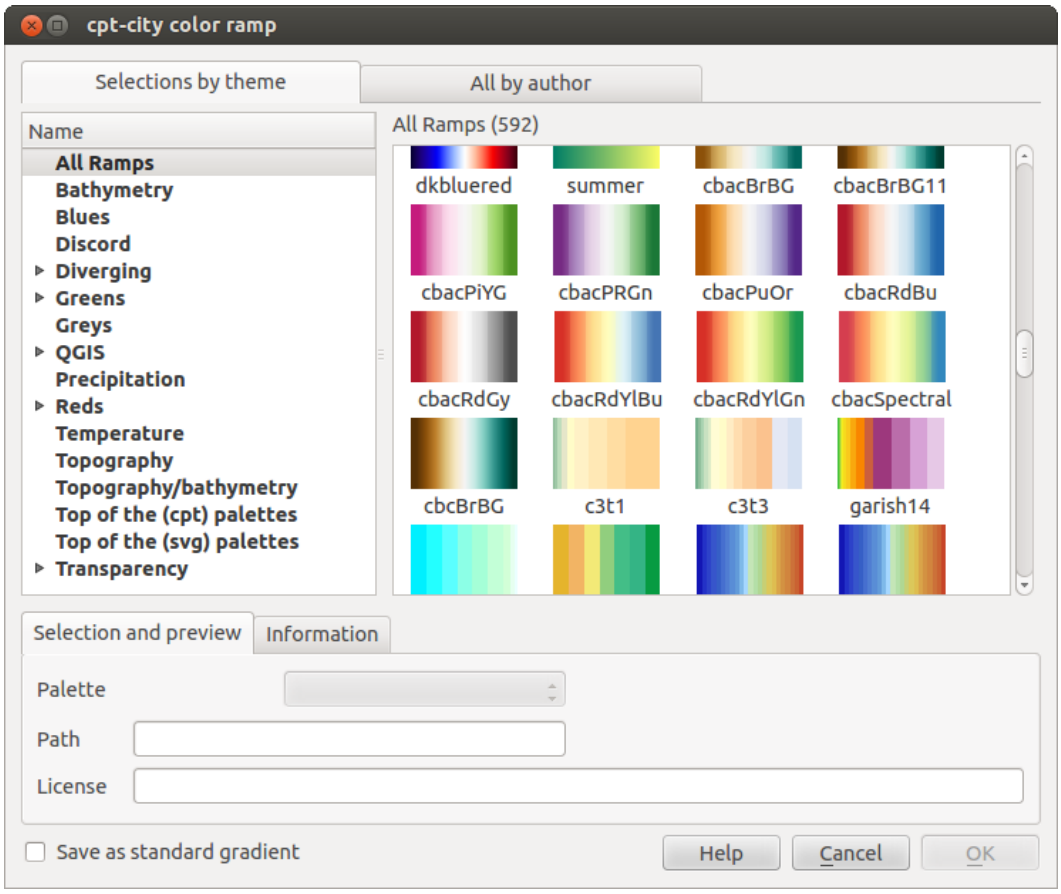


그림 12.7: 색상표 수백 개를 가진 cpt-city 대화창

팁: 그레이디언트 색상표의 색상 지점을 쉽게 조정

색상표 미리보기를 더블 클릭하거나, 색상 스폿 (colorspot) 에서 색상표 미리보기로 색상을 드래그 & 드롭하면 새 색상 지점을 추가합니다. 색상 선택기 위젯을 이용하거나, 색상 지점의 각 파라미터를 조정하면 각 색상 지점을 조정할 수 있습니다. 마우스, 방향키 (`Shift` 키와 함께 누르면 더 길게 움직입니다) 또는 *Relative position* 스핀박스 (spinbox) 를 이용해서 색상 지점의 위치를 이동시킬 수도 있습니다. `DEL` 키는 물론 *Delete stop* 버튼을 누르면 선택한 색상 지점을 삭제합니다.

12.2 심볼 선택기

심볼 선택기는 심볼을 디자인하기 위한 주 대화창입니다. 마커 (marker), 라인 (line) 또는 채우기 (fill) 심볼을 생성하거나 편집할 수 있습니다.

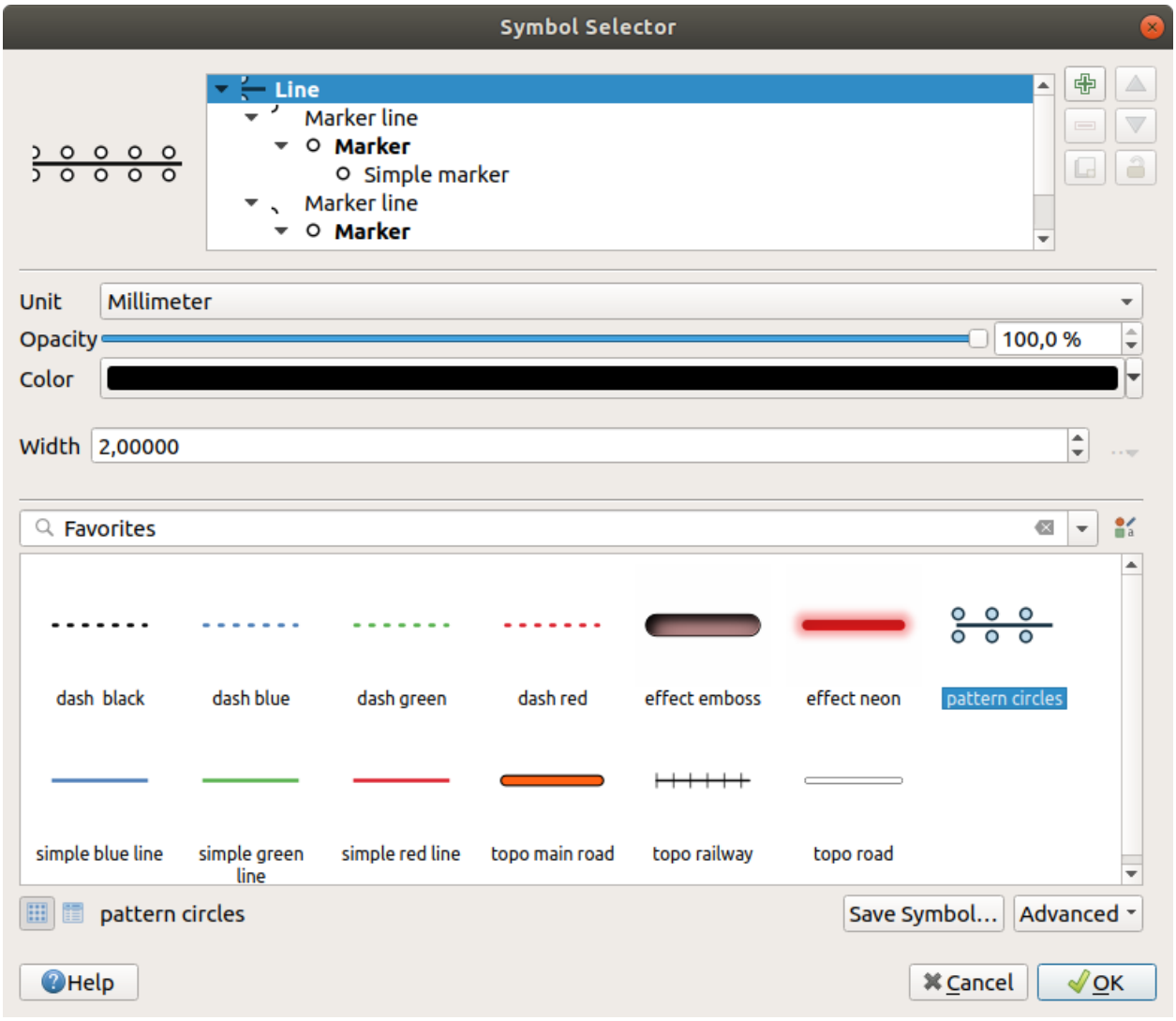


그림 12.8: 라인 심볼 디자인하기

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.

12.2.1 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.
- **|Up| |Down|**: 심볼 레이어를 위 또는 아래로 이동합니다.

12.2.2 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**, **축척에 따른 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, 명칭 및 관련 태그를 함께 표시하는) 아이콘 목록
  - 또는  Icon View 버튼을 통한 아이콘 미리보기
- 심볼 라이브러리에 편집 중인 심볼을 추가하려면 *Save Symbol* 버튼을 클릭하십시오.
- *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.
  - 심볼이 적용되는 레이어의심볼 에 따라, *Advanced* 메뉴에서 다음과 같은 추가 설정을 할 수 있습니다:
    * 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**, **축척에 따른 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:

- **단순 마커** (기본값)

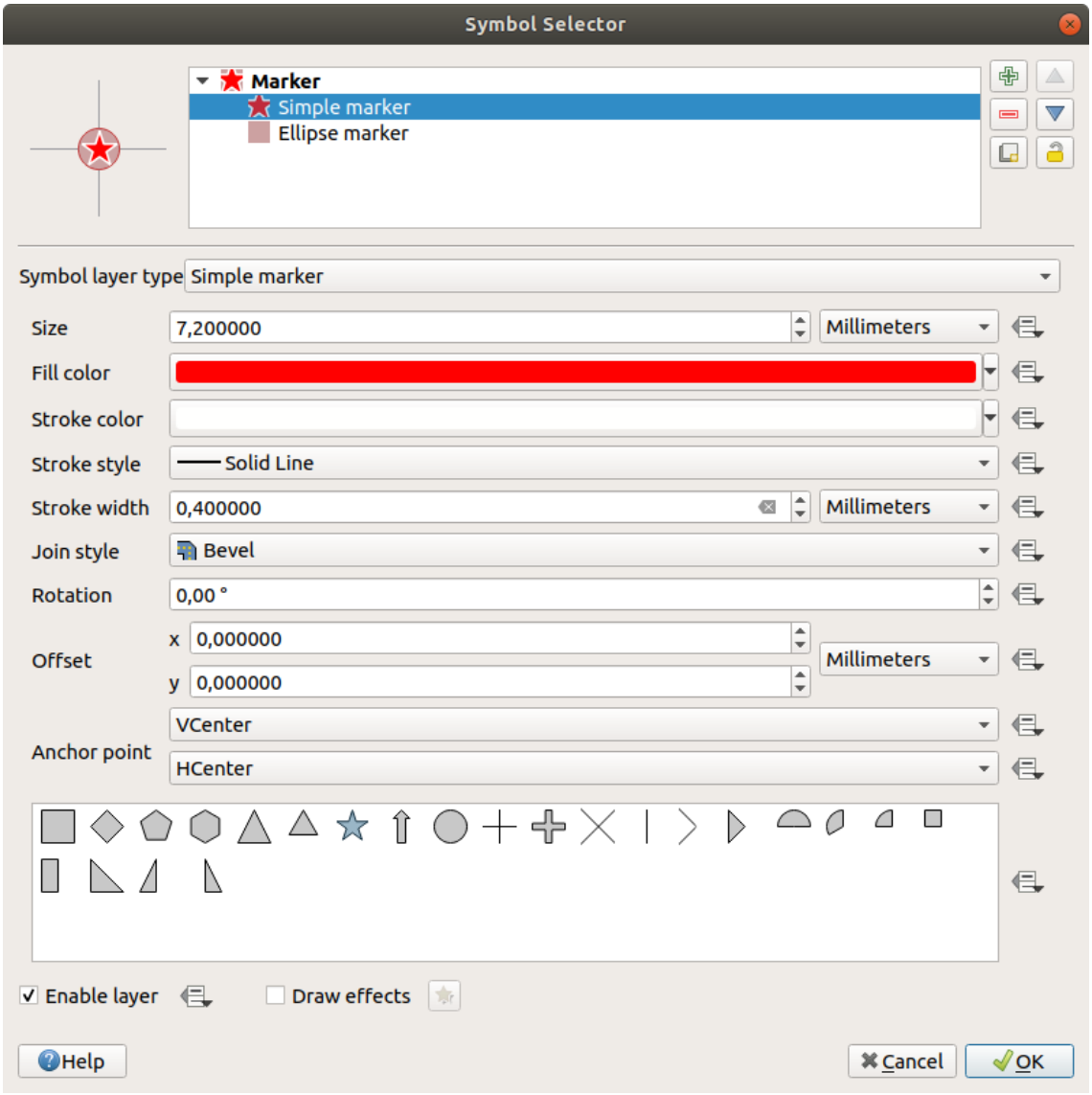


그림 12.9: 단순 마커 심볼 디자인하기

- **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**: 이미지 (`PNG`, `JPG`, `BMP` 등등) 를 마커 심볼로 사용합니다. 이때 이미지는 디스크 상의 파일일 수도 있고, 원격 URL 일 수도 있으며, 스타일 데이터베이스에 (자세한 내용은여기를 참조하세요) 내장되어 있을 수도 있습니다. 이미지의 너비와 높이를 독립적으로 설정할 수도 있고, Lock aspect ratio 버튼으로 고정할 수도 있습니다. 공통 단위 가운데 하나를 사용하거나, 이미지의 원본 크기의 (너비의) 백분율을 사용해서 이미지 크기를 설정할 수 있습니다.
- **벡터장 마커 (vector field marker)** (벡터장 마커 참조)
- **SVG 마커**: 사용자의 (*Settings* → *Options*··· → *System* 메뉴에서 설정한) SVG 경로에서 마커 심볼로 렌더링할 이미지를 불러옵니다. 이미지의 너비와 높이를 독립적으로 설정할 수도 있고, Lock aspect ratio 버튼으로 고정할 수도 있습니다. 각 SVG 파일의 색상 및 외곽선도 조정할 수 있습니다. 이때 이미지는 디스크 상의 파일일 수도 있고, 원격 URL 일 수도 있으며, 스타일 데이터베이스에 (자세한 내용은여기를 참조하세요) 내장되어 있을 수도 있습니다.

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.

팁: SVG 마커 심볼 사용자 지정 활성화

SVG marker 의 색상을 변경할 수 있게 만들려면, 채우기 색상 용 `param(fill)`, 외곽선 용 `param(outline)` 및 획 (stroke) 의 너비 용 `param(outline-width)` 자리표시자 (placeholder) 를 추가해야 합니다. 이 자리표시자들은 다음과 같이 기본값 앞에 선택적으로 삽입할 수 있습니다:

```
<svg width="100%" height="100%">
<rect fill="param(fill) #ff0000" stroke="param(outline) #00ff00" stroke-width=
↪"param(outline-width) 10" width="100" height="100">
</rect>
</svg>
```

라인 심볼

라인 도형 피처에 어울리는 라인 심볼은 다음과 같은 여러 *Symbol layer types* 를 가지고 있습니다:

- **단순 라인** (기본값): 다음 옵션을 설정할 수 있습니다:

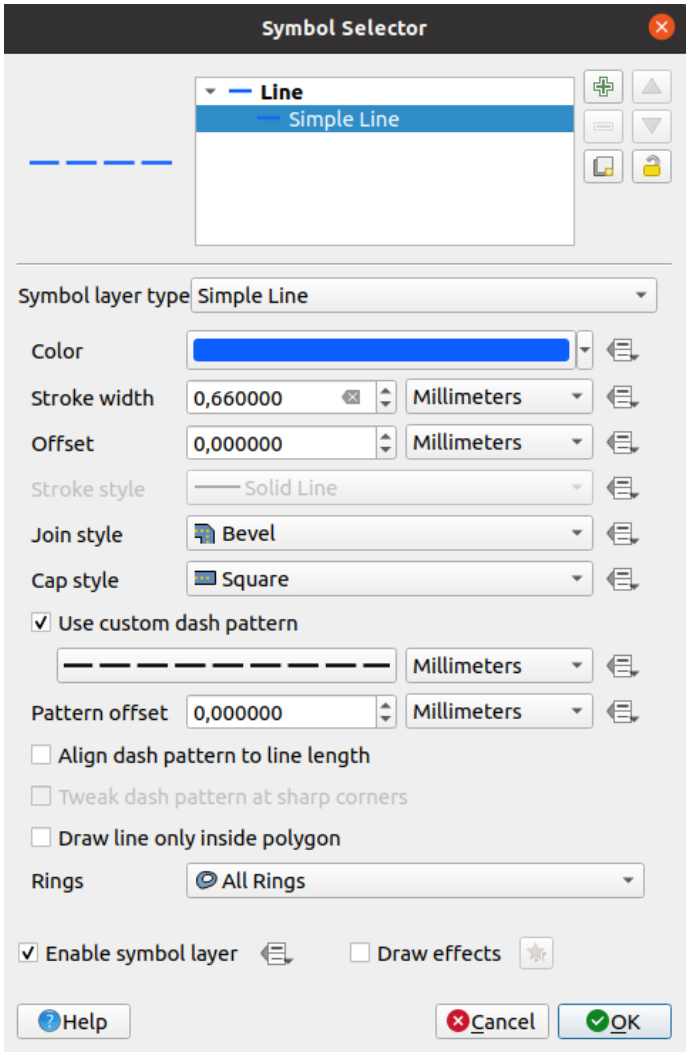


그림 12.10: 단순 라인 심볼 디자인하기

단순 라인 심볼 레이어 유형의 속성은 많은 부분단순 마커 심볼 과 동일하며, 추가로:

- *Cap style*: 라인의 끝을 어떻게 그릴지 설정합니다.
- *Use custom dash pattern*: *Stroke style* 설정을 무시하고 사용자 지정 점선 패턴을 사용합니다.
- *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.
- *Draw line only inside polygon*: 이 옵션을 활성화하면 폴리곤 내부에만 라인을 그립니다.

- **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*
  - *Arrow type*
  - *Arrow width*
  - *Arrow width at start*
  - *Head length*
  - *Head thickness*
  - *Offset*

  (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)
- **마커 라인**: 라인의 길이만큼마커 심볼 을 반복합니다.
  - 마커를 일정한 간격으로 배치할 수도 있고, 도형에 따라 처음, 마지막 또는 각 꼭짓점에, 라인의 중심점 아니면 각 선분의 중심점마다, 혹은 만곡 포인트마다 배치할 수도 있습니다.
  - 마커를 라인을 따라 지정한 오프셋으로 배치할 수도 있습니다.
  - *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.
  - 마커 라인을 라인 자체로부터 오프셋만큼 떨어지게 할 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
  - *Rotate hash to follow line direction*: 라인 방향을 따라 해시 기울이기

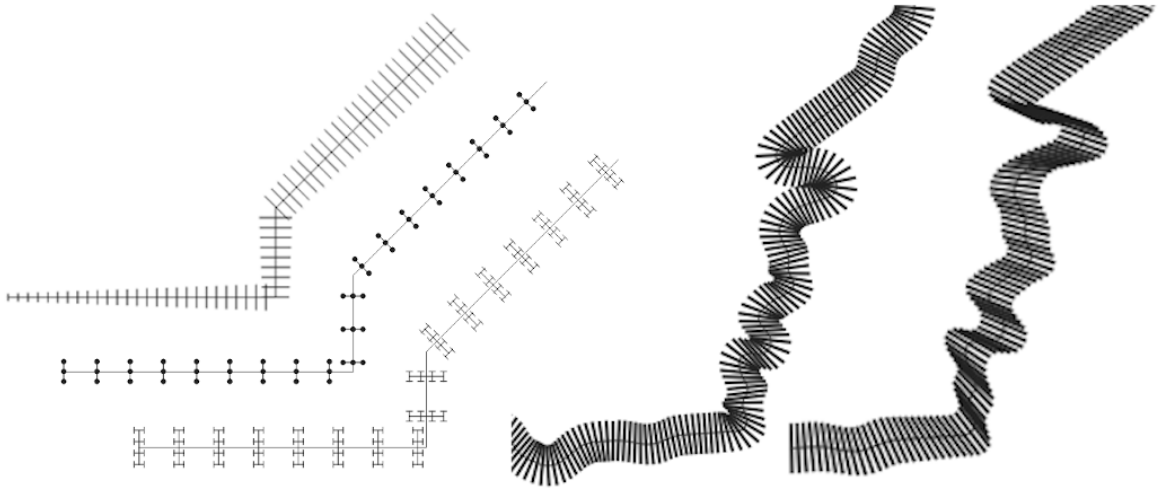


그림 12.11: 해시 라인의 예시

채우기 심볼

폴리곤 도형 피처에 어울리는 채우기 심볼은 다음과 같은 여러 *Symbol layer types* 를 가지고 있습니다:

- **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 point inside polygon*: 폴리곤 내부에 포인트를 그리도록 강제할 수 있습니다.
  - *Draw point on every part of multi-part feature*: 여러 부분으로 이루어진 피처의 모든 부분에 포인트를 그리거나, 또는 가장 큰 부분에만 그릴 수도 있습니다.
  - 현재 피처 도형과 겹치는 부분들을 유지하면서 (*Clip markers to polygon boundary*) 또는 심볼이 속해 있는 도형 부분을 유지하면서 (*Clip markers to current part boundary only*) 마커 심볼 전체 또는 일부를 표시할 수 있습니다.
- **도형 생성기 (geometry generator)** (도형 생성기 참조)
- **그레이디언트 채우기**: 단순한 2 색상 그레이디언트 또는 사전 정의된그레이디언트 색상표 를 바탕으로 방사형, 선형 또는 원추형 그레이디언트를 사용해서 폴리곤을 채웁니다. 그레이디언트를 회전시킬 수도 있고, 단일 피처에 한해 또는 전체 맵 범위에 걸쳐 적용할 수도 있습니다. 또 좌표를 통해, 또는 (피처 또는 맵의) 중심을 이용해서 시작점 및 종단점을 설정할 수도 있습니다. 데이터 정의 오프셋도 정의할 수 있습니다.
- **라인 패턴 채우기**: 라인 심볼 레이어 의 해치 (hatch) 패턴으로 폴리곤을 채웁니다. 라인의 기울기, 라인 사이의 간격 및 피처 경계로부터의 오프셋을 설정할 수 있습니다.
- **포인트 패턴 채우기**: 마커 심볼 레이어 의 해치 (hatch) 패턴으로 폴리곤을 채웁니다. 마커들의 행 사이의 거리, 전치 (□□; displacement) 및 피처 경계로부터의 오프셋을 설정할 수 있습니다.
- **랜덤 마커 채우기**: 폴리곤 경계선 안의 랜덤한 위치에마커 심볼 을 배치해서 폴리곤을 채웁니다. 다음을 설정할 수 있습니다:
  - 렌더링할 마커 심볼의 개수를 확실한 개수 또는 밀도 기반으로 (채우기 밀도는 서로 다른 축척 또는 확대/축소 수준에서도 동일할 것입니다) 설정할 수 있습니다.

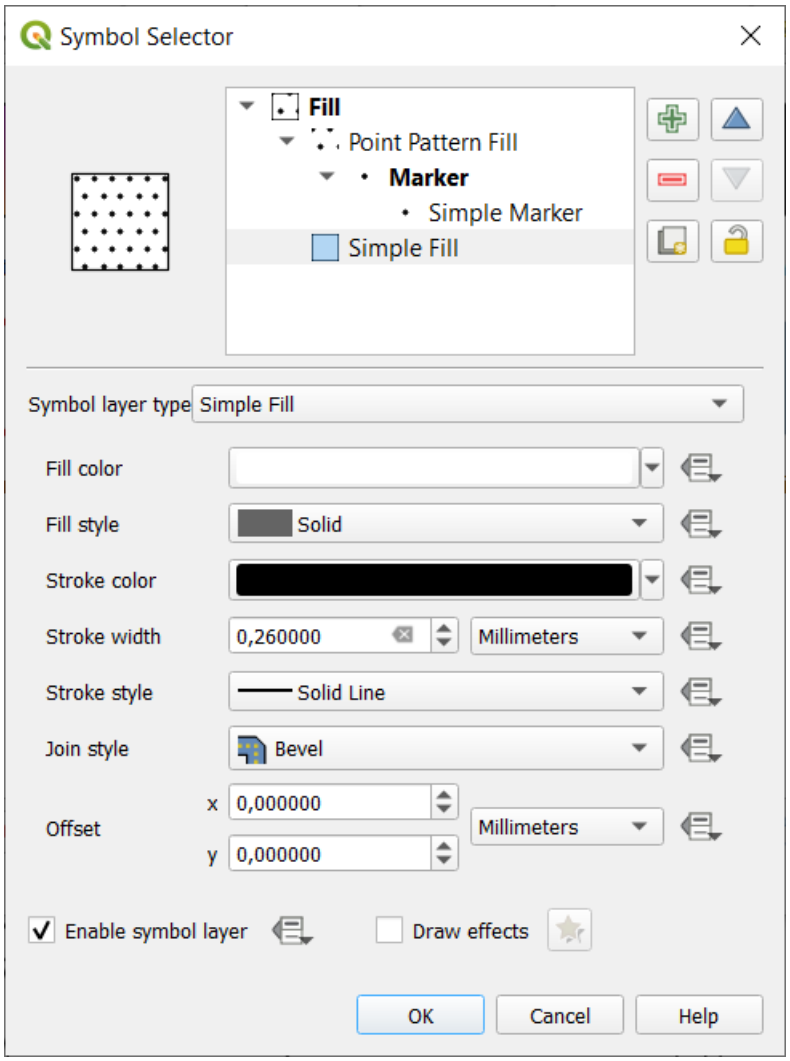


그림 12.12: 단순 채우기 심볼 디자인하기

  - 맵이 새로고침될 때마다 변함없는 마커 배치를 보장하도록 시드 (seed) 개수를 랜덤하게 설정할 수 있습니다. (이 옵션은 QGIS 서버 및 타일 기반 렌더링에서도 보기 좋도록 하는 랜덤한 배치도 허용합니다.)
  - 폴리곤 경계 가까이 렌더링된 마커를 폴리곤 경계선에 맞춰 잘라낼지 말지를 설정할 수 있습니다.
- **래스터 이미지 채우기**: 래스터 이미지 (PNG JPG, BMP 등등) 의 타일로 폴리곤을 채웁니다. 이때 이미지는 디스크 상의 파일일 수도 있고, 원격 URL 일 수도 있으며, 문자열로 인코딩된 파일로 (자세한 내용은여기를 참조하세요) 내장되어 있을 수도 있습니다. (데이터 정의된) 투명도, 이미지 너비, 좌표 모드 (객체 또는 뷰), 기울기 및 오프셋 옵션을 설정할 수 있습니다. 공통 단위 가운데 하나를 사용하거나, 이미지의 원본 크기의 백분율을 사용해서 이미지 너비를 설정할 수 있습니다.
- **SVG 채우기**: *SVG* 마커 로 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.
- **외곽선: 화살표**: 라인화살표 심볼 레이어를 이용해서 폴리곤 경계선을 표현합니다. 외곽선 화살표의 설정은 라인 심볼의 설정과 동일합니다.
- **외곽선: 해시 라인**: 해시 라인 심볼 레이어를 이용해서 폴리곤 경계선 (내부 고리, 외부 고리 또는 모든 고리) 을 표현합니다. 외곽선 해시 라인의 설정은 라인 심볼의 설정과 동일합니다.
- **외곽선: 마커 라인**: 마커 라인 심볼 레이어를 이용해서 폴리곤 경계선 (내부 고리, 외부 고리 또는 모든 고리) 을 표현합니다. 외곽선 마커 라인의 설정은 라인 심볼의 설정과 동일합니다.
- **외곽선: 단순 라인**: 단순 라인 심볼 레이어를 이용해서 폴리곤 경계선 (내부 고리, 외부 고리 또는 모든 고리) 을 표현합니다. 외곽선 단순 라인의 설정은 라인 심볼의 설정과 동일합니다. *Draw line only inside polygon* 옵션을 선택하면 폴리곤 안쪽에 폴리곤 경계선을 그리는데, 서로 인접한 폴리곤들의 경계선을 명확하게 표현하는 데 유용합니다.

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.)

도형 생성기

모든 심볼 유형에 대해 사용할 수 있는 *Geometry Generator* 심볼 레이어는 렌더링 과정 중에 실시간으로 도형을 생성하는표현식 문법 을 사용할 수 있습니다. 산출 도형은 원본 도형 유형과 일치할 필요가 없고, 서로 다르게 수정된 여러 심볼 레이어들을 중첩해서 추가할 수 있습니다.

다음은 몇몇 예시입니다:

```
-- render the centroid of a feature
centroid( $geometry )

-- visually overlap features within a 100 map units distance from a point
-- feature, i.e generate a 100m buffer around the point
buffer( $geometry, 100 )

-- Given polygon layer1( id1, layer2_id, ...) and layer2( id2, fieldn...)
-- render layer1 with a line joining centroids of both where layer2_id = id2
make_line( centroid( $geometry ),
           centroid( geometry( get_feature( 'layer2', 'id2', attribute(
               $currentfeature, 'layer2_id') ) )
```

(다음 페이지에 계속)

(이전 페이지에서 계속)

```
        )

-- Create a nice radial effect of points surrounding the central feature
-- point when used as a MultiPoint geometry generator
collect_geometries(
  array_foreach(
    generate_series( 0, 330, 30 ),
      project( $geometry, .2, radians( @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.

12.3 라벨 설정

라벨은 벡터 피처 또는 맵 위에 표시할 수 있는 텍스트 정보를 말합니다. 심볼로는 표현할 수 없는 세부 정보를 추가합니다. QGIS 는 두 가지의 텍스트 관련 항목을 사용할 수 있습니다.

- *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*: 텍스트 서식 설정을 위치 또는 다른 텍스트 또는 피처와의 상호 작용과 관련된 속성 (호출, 배치, 중첩, 축척에 따른 가시성, 마스크 등등) 으로 확장합니다.

  이 속성들은 벡터 레이어 *Layer Properties* 대화창의 *Labels* 탭이나 *Layer Styling* 패널, 또는라벨 툴바 의 Layer Labeling Options 버튼을 통해 벡터 레이어를 위한 스마트 라벨 작업을 환경 설정하는 데 사용됩니다.

  Label Settings 항목을 생성하려면:

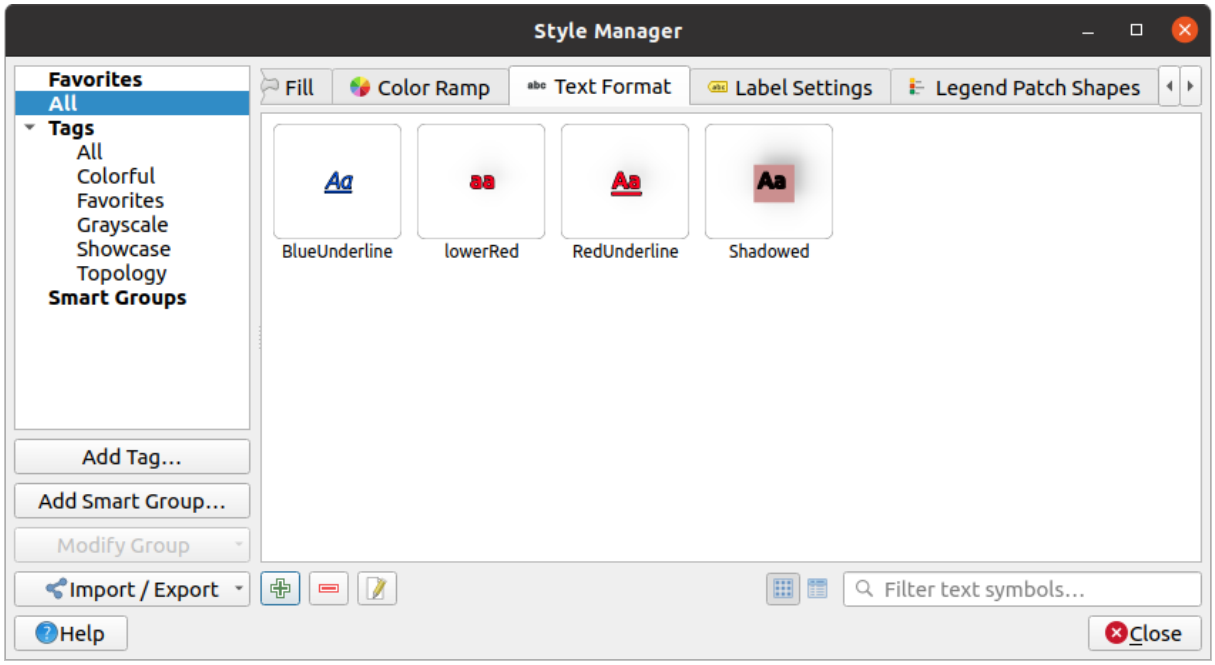


그림 12.13: 스타일 관리자 대화창의 텍스트 서식

1. *Style Manager* 대화창을 열고
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.

12.3.1 라벨 텍스트 서식 설정

다음 속성들 가운데 대부분이 *Text Format* 및 *Label Settings* 항목들과 동일합니다.

텍스트 탭

abc *Text* 탭에서 다음 항목을 설정할 수 있습니다:

- *Font*: 사용자 컴퓨터에서 사용할 수 있는 글꼴 가운데 선택할 수 있습니다.
- *Style*: 일반적인 글꼴 스타일은 물론, 밑줄 또는 취소선도 설정할 수 있습니다.
- *Size*: 텍스트의 크기를 모든지원 단위 로 설정할 수 있습니다.
- *Color*: 텍스트의 색상을 설정할 수 있습니다.
- *Opacity*: 텍스트의 투명도를 선택할 수 있습니다.

탭 하단에 있는 위젯은 사용자의스타일 관리자 데이터베이스 에 저장된 호환 가능한 항목들을 필터링 가능한 목록으로 보여줍니다. 이를 통해 기존 서식 또는 설정을 바탕으로 현재 텍스트 서식 또는 라벨 설정을 쉽게 환경설정할 수 있고, 스타일 데이터베이스에 새 항목을 추가할 수도 있습니다. *Save format*…또는 *Save settings*…버튼을 누르고 명칭 및 태그 (들) 를 입력하면 됩니다.

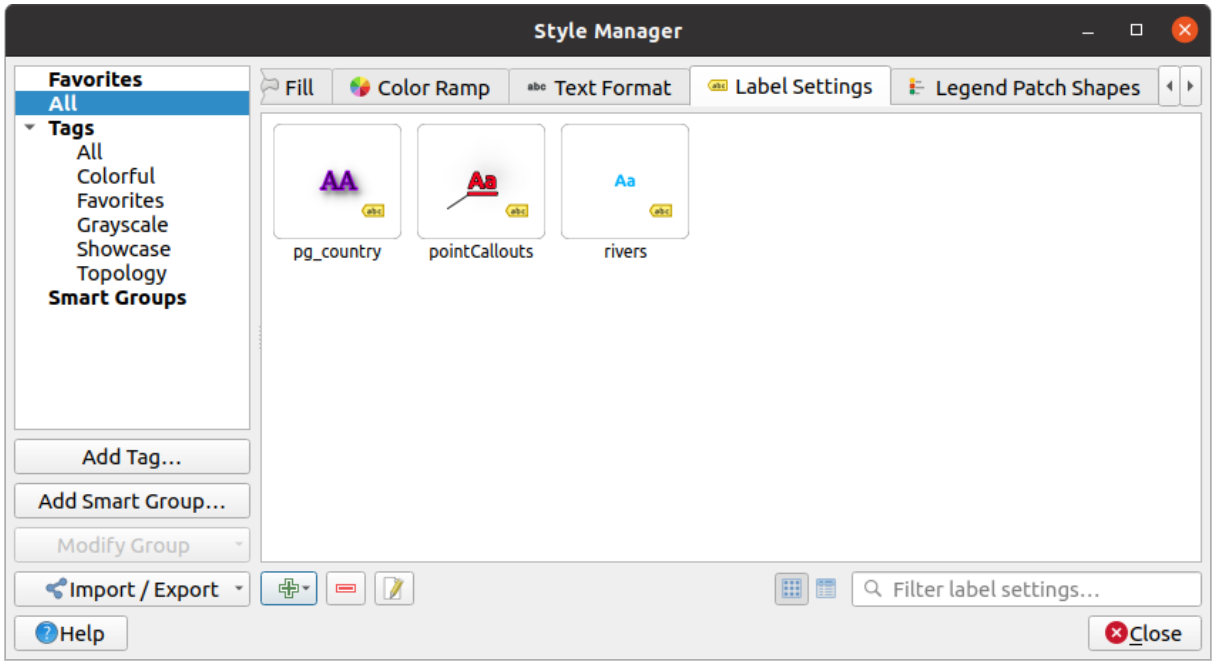


그림 12.14: 스타일 관리자 대화창에서 라벨 설정

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.

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
  - *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.
- 단어들 사이의 여백, 개별 문자 사이의 여백을 변경하려면 *Spacing* 옵션을 사용하십시오.
- *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.

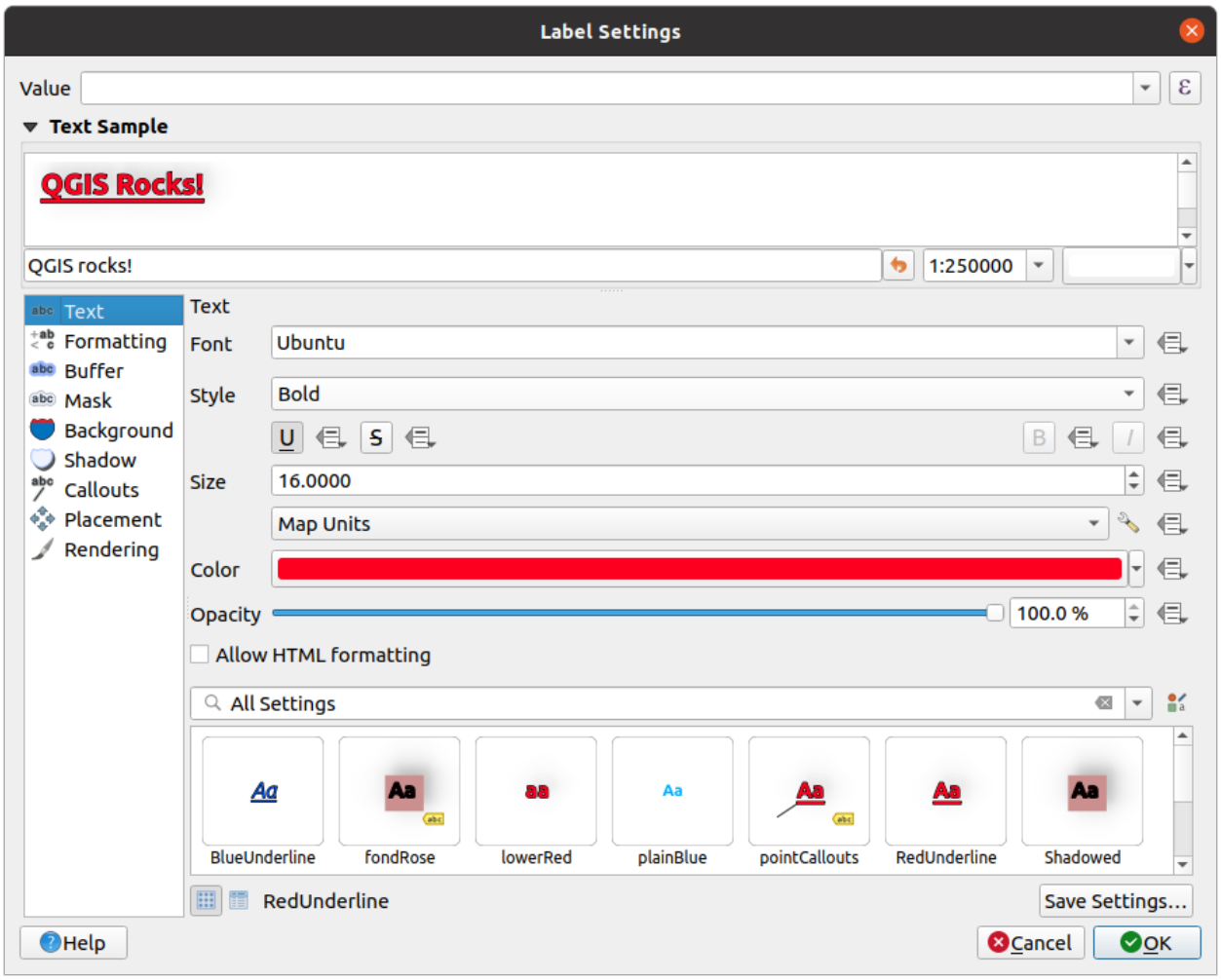


그림 12.15: 라벨 설정 - 텍스트 탭

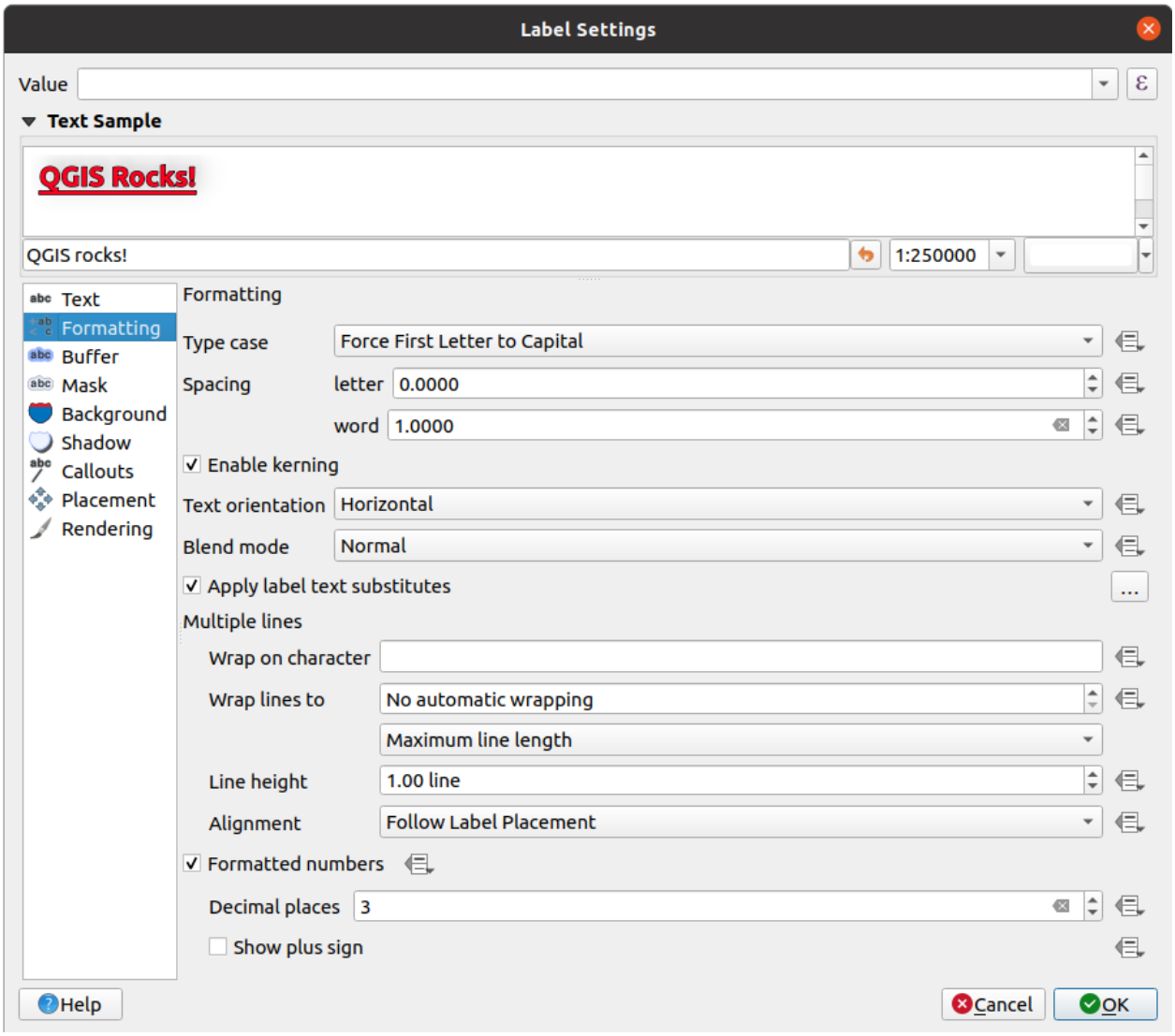


그림 12.16: 라벨 설정 - 서식 탭

- 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*: 행 높이를 설정할 수 있습니다.
  - *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

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.)

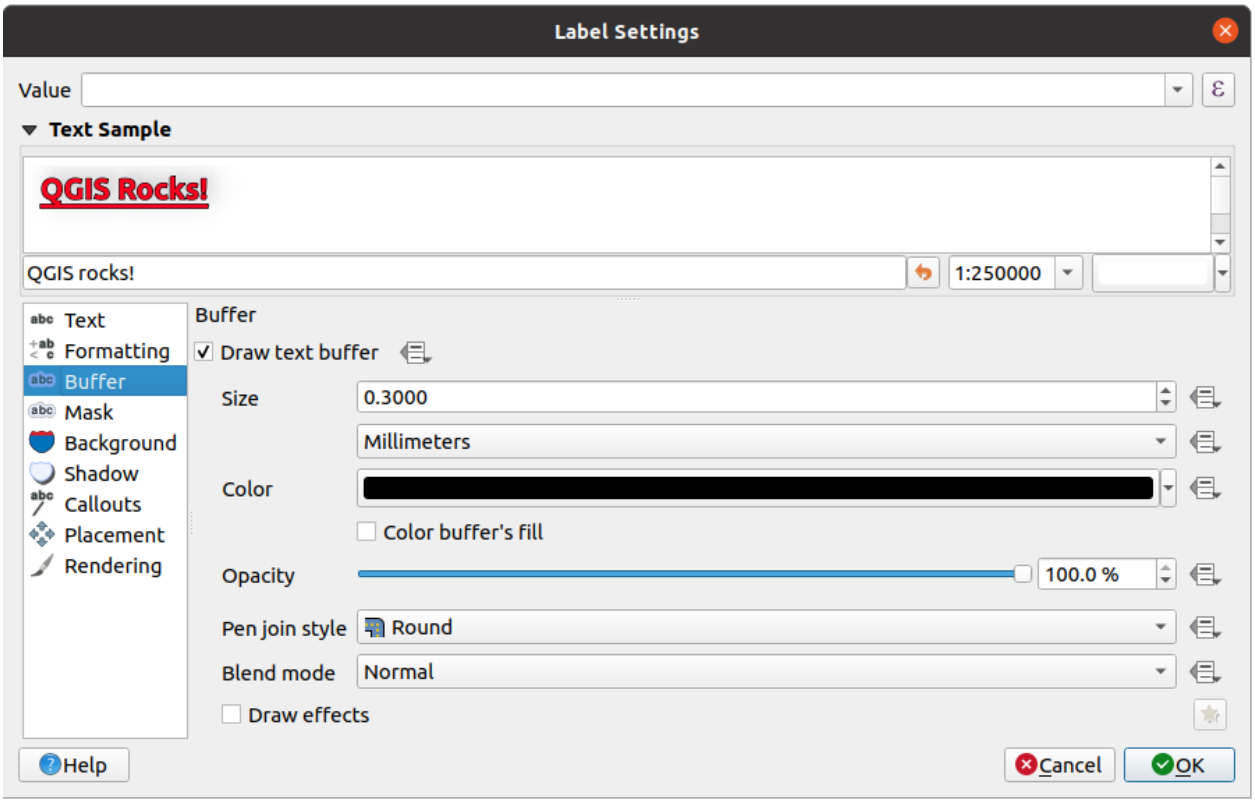


그림 12.17: 라벨 설정 - 버퍼 탭

- 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* 직사각형, *Square* 정사각형, *Circle* 원, 또는 *Ellipse* 타원형 같은 정규 형태
- 파일, URL 이나 프로젝트 또는 스타일 데이터베이스에 내장된 *SVG* 심볼 (자세한 내용은여기를 참조)
- 심볼 라이브러리 에서 생성하거나 선택할 수 있는 *Marker Symbol*

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.

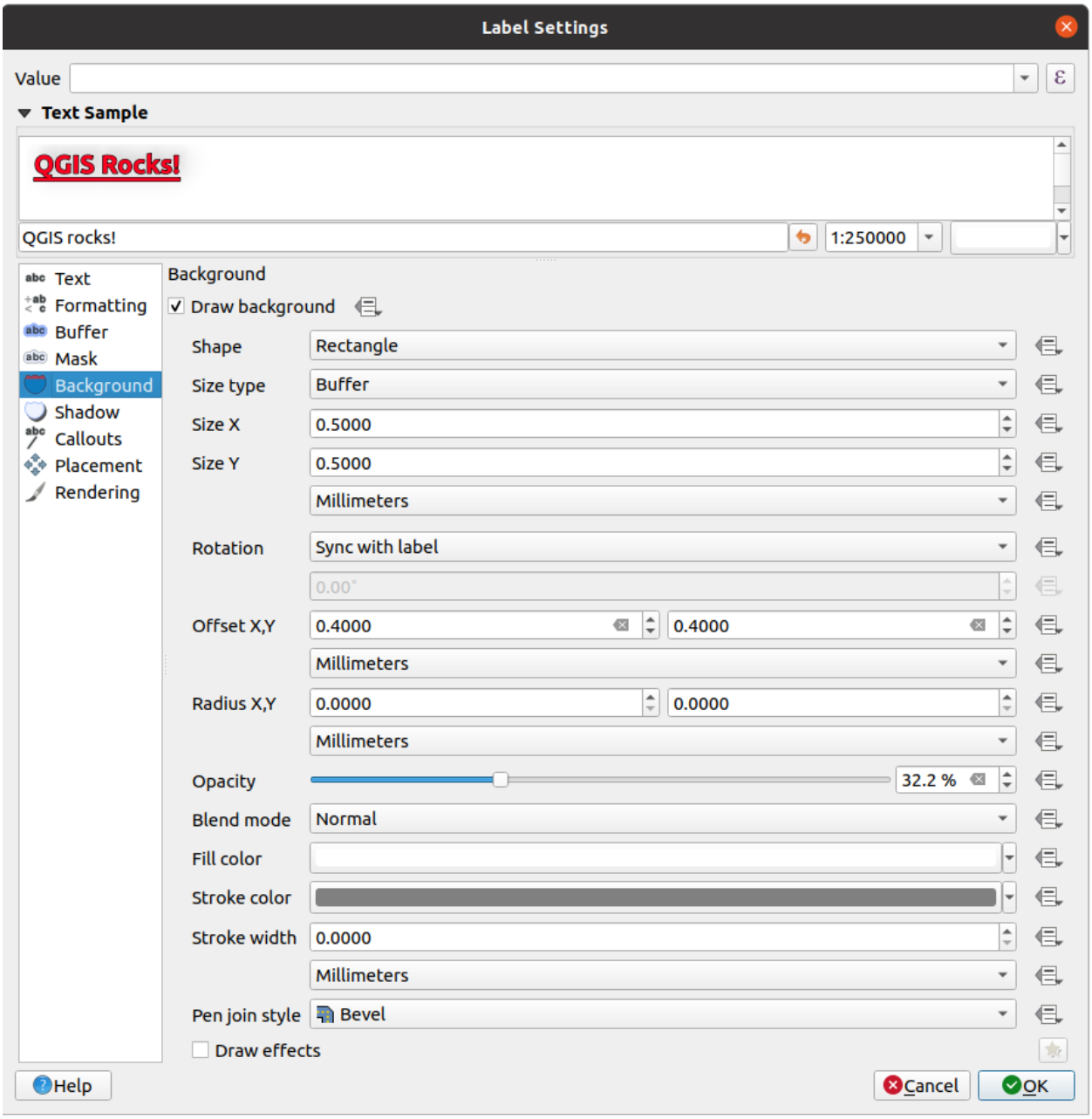


그림 12.18: 라벨 설정 - 배경 탭

- *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)
- *Fill color*, *Stroke color* 및 *Stroke width*: 마커 심볼이 아닌 다른 형태 유형의 채우기 색상, 획 색상 및 획 너비를 설정할 수 있습니다. SVG 심볼에 대한 변경 사항을 기본 설정으로 되돌리려면 *Load symbol parameters* 를 사용하십시오.
- *Pen join style*: *Round* 원호, *Miter* 직각 또는 *Bevel* 경사 가운데 하나를 선택할 수 있습니다. (직사각형 및 정사각형 형태에만 적용할 수 있습니다.)
- *Draw effects*: 텍스트 가독성을 향상시키기 위한 고급 그리기 효과 를 추가할 수 있습니다.

그림자 탭

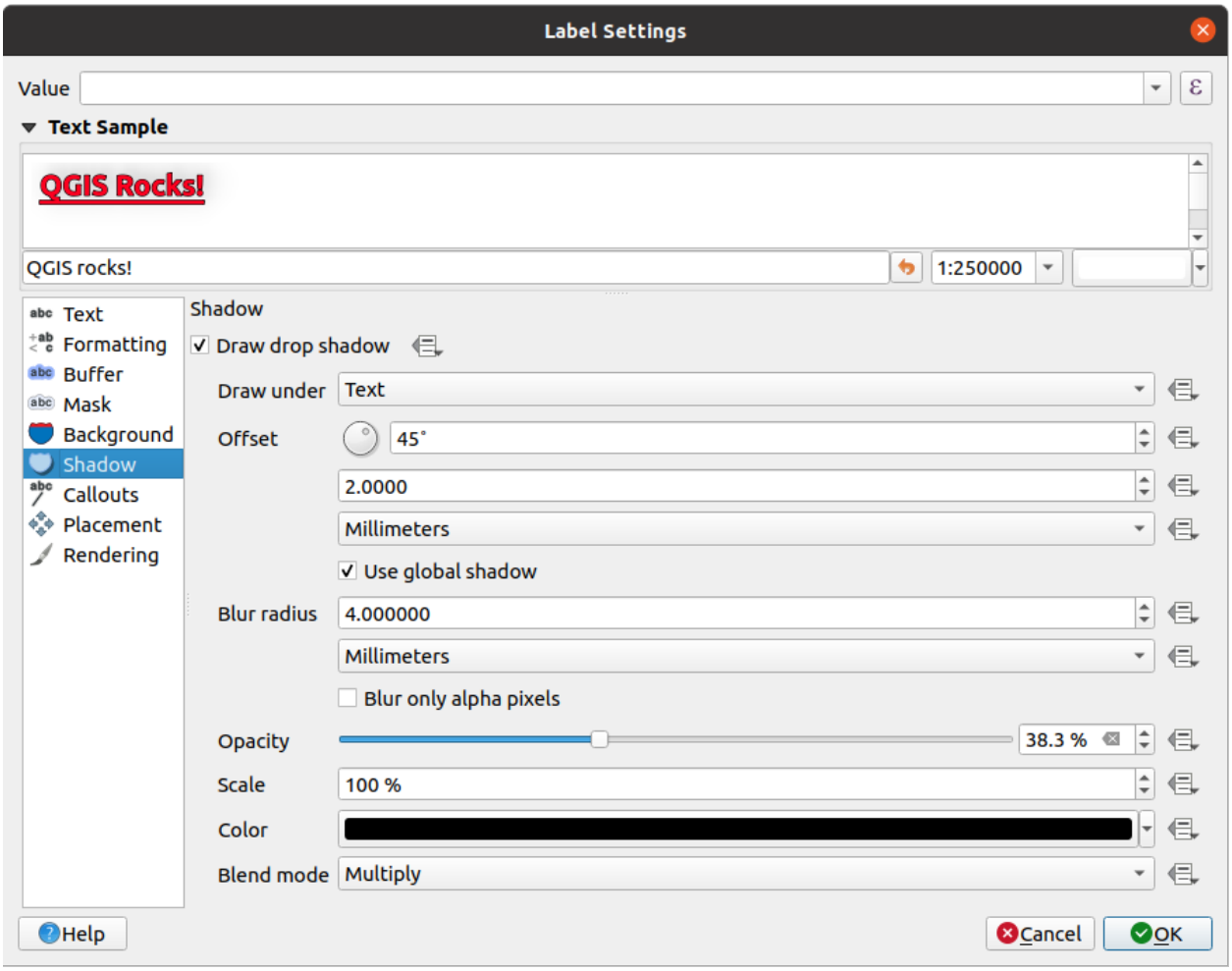


그림 12.19: 라벨 설정 - 음영 탭

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.
- 그림자가 질 항목으로부터의 *Offset* 을 다음 가운데 설정하십시오:
  - 각도: 그림자가 질 각도를 기저 항목의 방향에 따라 시계 방향으로 설정합니다.
  - 그림자가 질 항목으로부터의 그림자 오프셋 거리를 설정합니다.
  - 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.)

12.3.2 라벨의 상호 작용을 환경 설정하기

앞에서 설명한 텍스트 서식 설정 이외에도, 라벨이 다른 라벨 또는 피처와 어떻게 상호 작용할지 설정할 수 있습니다.

마스크 탭

Mask 탭에서 라벨 주위에 마스크 영역을 정의할 수 있습니다. 사용자가 심볼 및 라벨을 비슷한 색상으로 덮으면서 라벨이 잘 보이도록 하려 하는 경우 이 기능이 아주 유용합니다.

라벨에 마스크 효과를 생성하려면:

1. 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 *Mask* 탭에서 이 마스크 형태를 마스크 소스로 선택하십시오. (마스크 속성 참조)

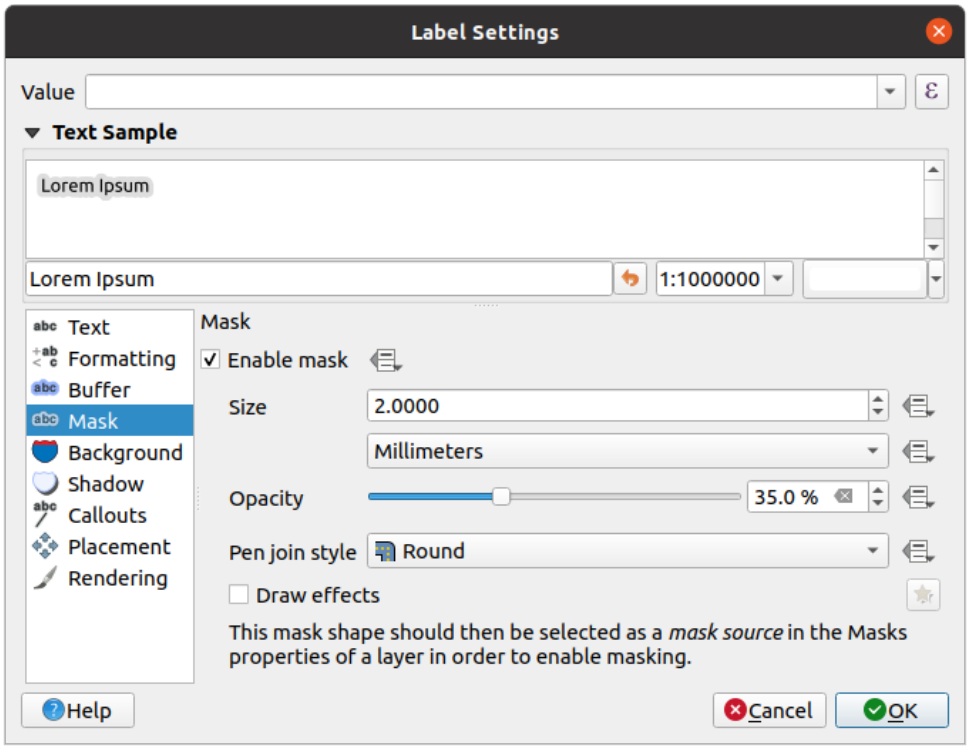


그림 12.20: 라벨 설정 - 마스크 탭

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
2. 레이어 효과 및 데이터 정의 설정을 포함한, 라인 심볼 의 완전한 성능을 가진 *Line style* 을 선택하십시오.
3. *Minimum length*: 호출 하인의 최단 길이를 설정합니다.
4. *Offset from feature*: 이 옵션은 호출 라인이 끝나는 지점에서 피처 (또는 폴리곤인 경우 고정점 (anchor point)) 로부터의 거리를 제어할 수 있습니다. 예를 들어 피처의 경계에 밀착하는 호출 라인을 그리는 일을 피할 수 있습니다.
5. *Offset from label area*: 이 옵션은 (호출 라인이 끝나는) 라벨 고정점으로부터의 거리를 제어할 수 있습니다. 텍스트에 밀착하는 호출 라인을 그리는 일을 피할 수 있습니다.
6. *Draw lines to all feature parts*: 이 옵션을 활성화하면 피처 라벨에서 피처의 모든 부분으로 호출 라인을 그립니다.
7. *Anchor point*: 연결자 라인의 종단점인 (폴리곤) 피처의 고정점을 설정합니다. 다음 옵션 가운데 선택할 수 있습니다:



그림 12.21: 다양한 호출 설정을 가진 라벨

- *Pole of inaccessibility*: 도달불능극
- *Point on exterior*: 외곽 포인트
- *Point on surface*: 표면 포인트
- *Centroid*: 중심점

8. *Label anchor point*: 연결자 라인이 라벨 텍스트의 어디에 결합할지 제어합니다. 다음 옵션 가운데 선택할 수 있습니다:
   - *Closest point*: 가장 가까운 포인트에 연결합니다.
   - *Centroid*: 중심점
   - 가장자리에 있는 고정 위치 (*Top left*, *Top center*, *Top right*, *Left middle*, *Right middle*, *Bottom left*, *Bottom center* and *Bottom right*) 에 연결합니다.

배치 탭

라벨 배치 및 라벨 작업 순서를 환경 설정하려면 *Placement* 탭을 선택하십시오. 포인트, 라인 또는 폴리곤과 같은 벡터 레이어 유형에 따라 배치 옵션이 달라지며, 이는 전체 수준*PAL 설정* 의 영향을 받는다는 점을 기억하십시오.

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.
  - 우선 순위 위치의 데이터 정의 목록을 통해 개별 피처마다 사용자 정의 또는 설정할 수 있는 *Position priority* 옵션으로 배치할 수 있습니다. 이런 방법으로 특정 위치만 이용하도록 할 수도 있는데, 예를 들면 해안 피처의 경우 라벨이 육지 위로 올라오지 않도록 예방할 수 있습니다.

    기본적으로, 제도 제작법 모드 배치의 우선 순위는 (guidelines from Krygier and Wood (2011) 및 다른 지도 제작법 교과서들을 따르는) 다음 순서대로 입니다:

    1. 우상단
    2. 좌상단
    3. 우하단
    4. 좌하단
    5. 우중단
    6. 좌중단
    7. 약간 우측으로 쏠린 상단
    8. 약간 좌측으로 쏠린 하단
- *Around Point*: 라벨을 피처 주위의 (*Distance* 에서 설정하는) 동일 반경 원 안에 배치할 수 있습니다. 배치 우선 순위는"우상단"에서 시계 방향으로 내려갑니다. 데이터 정의 *Quadrant* 옵션을 이용하면 라벨의 배치를 더 제약할 수도 있습니다.
- *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:

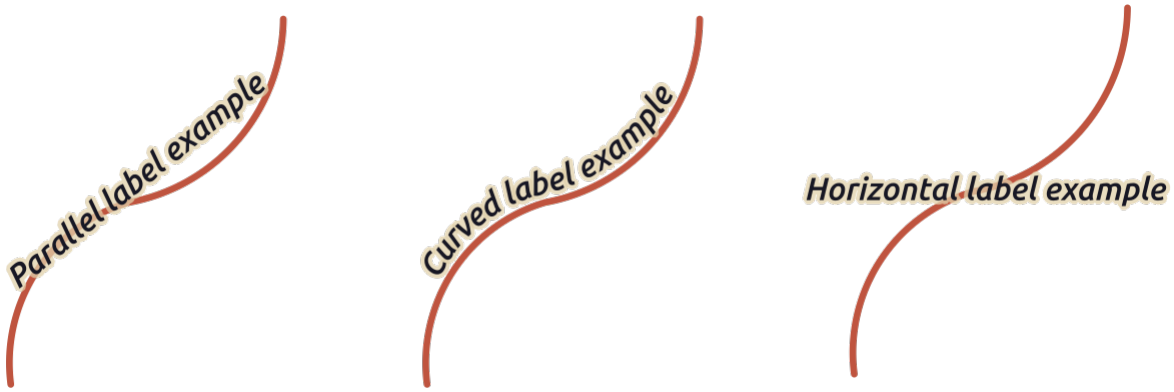


그림 12.22: 라인 피처의 라벨 배치 예시

- *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:
  - 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*….
  - *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
  - 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.

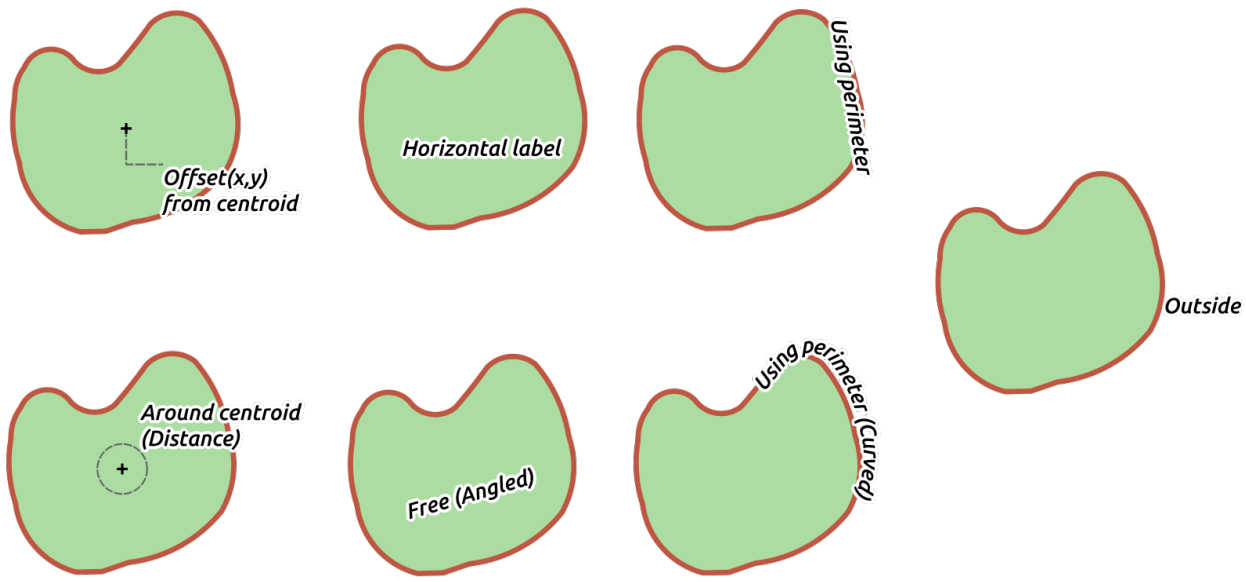


그림 12.23: 폴리곤 피처의 라벨 배치 예시

- *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:

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.
- 텍스트의 *Rotation*. 라벨이 고정되어 있는지 아닌지에 상관없이 관련 필드에 있는 기울기 값을 유지해서 라벨에 적용하고자 하는 경우, *Preserve data rotation values* 항목을 체크하십시오. 이 옵션을 체크하지 않으면, 라벨 기울기를 고정 해제하고 속성 테이블에서 그 값을 삭제합니다.

  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 우선순위를 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.

- QGIS 는 라벨 렌더링 작업 도중 라벨을 읽기 쉽게 표시하기 위해 라벨들의 위치를 자동적으로 평가하고 서로 충돌할 경우 일부 라벨을 숨길 수 있습니다. 하지만 *Show all labels for this layer (including colliding labels)* 체크박스를 활성화시켜 라벨들의 위치를 직접 수정할 수 있습니다. (라벨 툴바 참조)
- *Show label* 및 *Always Show* 에서 데이터 정의 표현식을 이용하면, 어떤 라벨을 렌더링해야 하는지를 미세 조정할 수 있습니다.
- *Show upside-down labels*: 라벨을 위아래로 뒤집어 표시할 수 있습니다. **Never**, **When rotation defined** 또는 **Always** 가운데 선택할 수 있습니다.

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.

12.4 3D 심볼 생성

Style Manager 를 통해*3D* 맵 뷰 에서 렌더링될 모든 도형 유형의 3D 심볼을 생성하고 저장할 수 있습니다.

다른 항목들과 마찬가지로,  *3D Symbols* 탭을 선택하고  버튼 메뉴를 확장해서 다음을 생성할 수 있습니다:

- *3D* 포인트 심볼
- *3D* 라인 심볼
- *3D* 폴리곤 심볼

12.4.1 포인트 레이어

- *Radius*, *Size* 또는 *Length* 로 정의된 *Sphere*, *Cylinder*, *Cube*, *Cone*, *Plane* 및 *Torus* 같은 서로 다른 단순 3 차원 형태를 정의할 수 있습니다. 3 차원 형태의 크기 단위는 프로젝트 좌표계를 참조합니다.
- *Diffuse*, *Ambient*, *Specular* 및 *Shininess* 메뉴를 통해 3 차원 형태의 그림자를 정의할 수 있습니다. (https://en.wikipedia.org/wiki/Phong_reflection_model#Description 참조)
- *3D Model* 을 선택했다면, 단순 포인트 좌표가 그 위치를 결정할 것입니다.
- 3 차원 점구름 (point cloud) 을 가시화하기 위해 *Billboard Height*, *Billboard symbol* 그리고 *Altitude clamping* 으로 정의된 *Billboard* 형태를 사용할 수 있습니다. 이 심볼은 안정적인 크기를 가질 것입니다.
- *Altitude clamping* 은 *Absolute*, *Relative* 또는 *Terrain* 가운데 하나로 설정할 수 있습니다. 3 차원 벡터의 높이값이 0 부터 측정된 절대값인 경우 *Absolute* 설정을 선택할 수 있습니다. *Relative* 및 *Terrain* 설정을 선택하면 기저 지형 표고에 지정한 표고값을 추가합니다.
- X, Y, Z 축을 따라 객체를 이동시키려면 *Translation* 옵션을 사용하면 됩니다.
- 3 차원 형태를 위한 *Scale factor* 는 물론 X, Y, Z 축을 기준으로 하는 *Rotation* 도 정의할 수 있습니다.

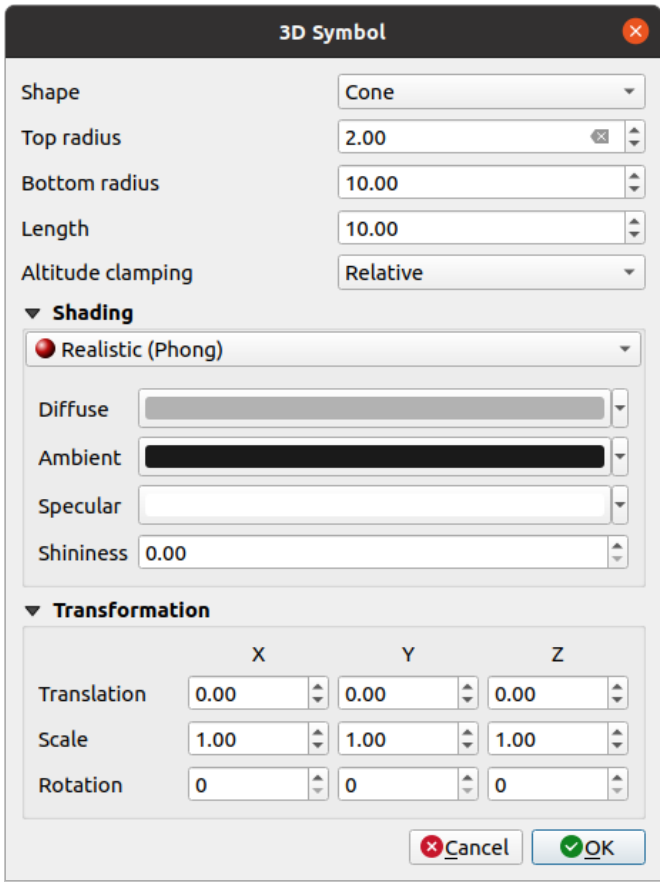


그림 12.24: 3 차원 포인트 심볼의 속성

12.4.2 라인 레이어

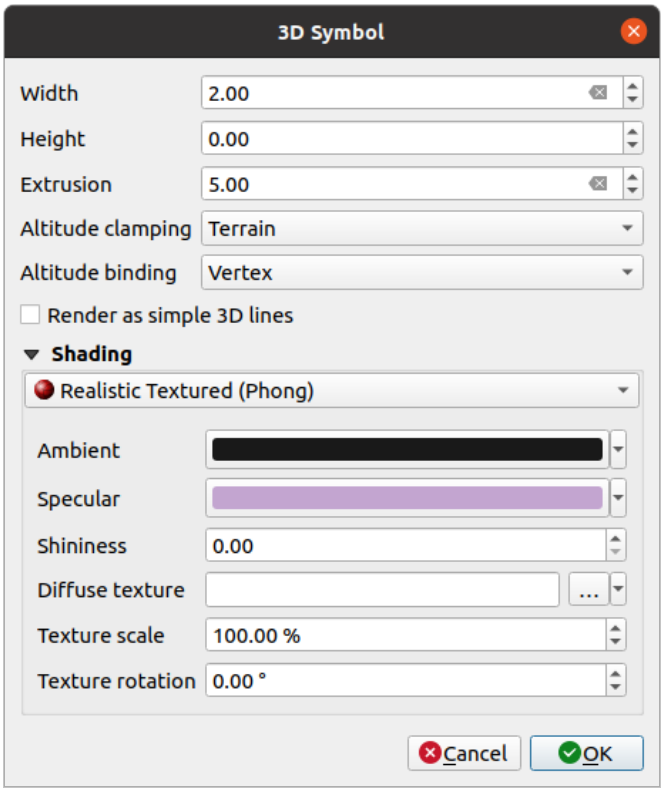


그림 12.25: 3 차원 라인 심볼의 속성

- *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.
- *Diffuse*, *Ambient*, *Specular* 및 *Shininess* 메뉴를 통해 그림자를 정의할 수 있습니다.

12.4.3 폴리곤 레이어

- 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.
- 부가적인 *Add back faces* 옵션으로 배면 (☐☐) 을 생성할 수 있고, *Invert normals* 옵션으로 법선 (☐☐) 을 뒤집을 수 있습니다.

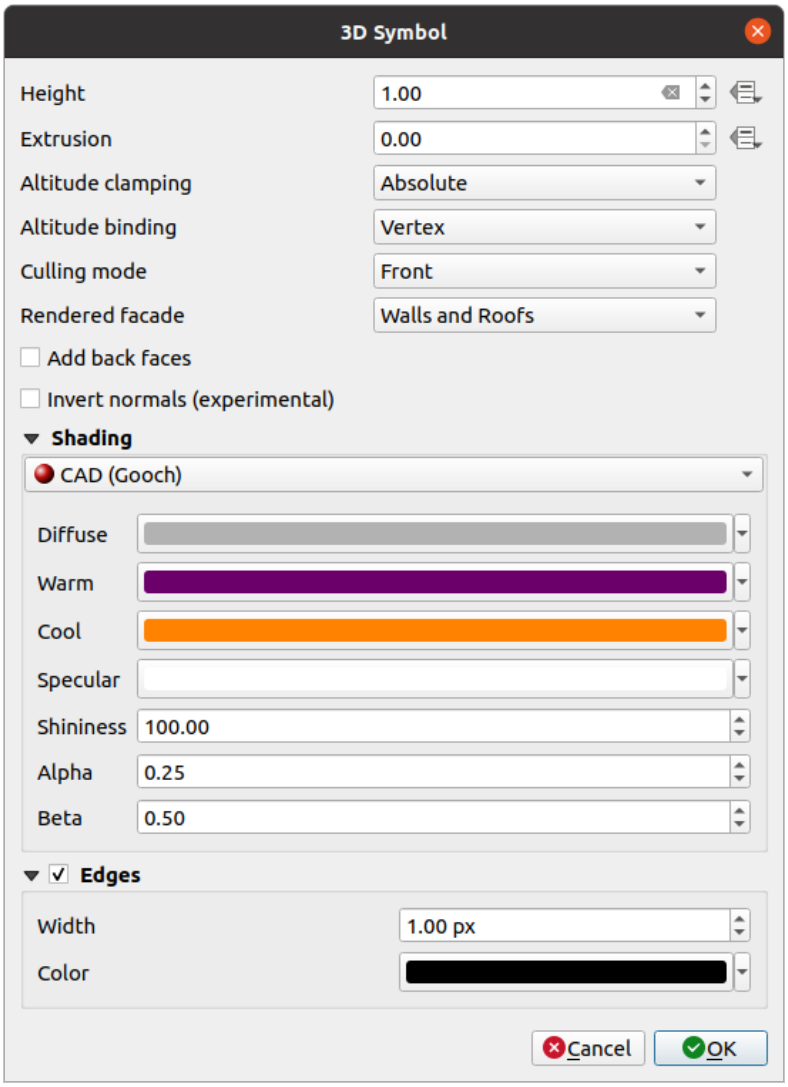


그림 12.26: 3 차원 폴리곤 심볼의 속성

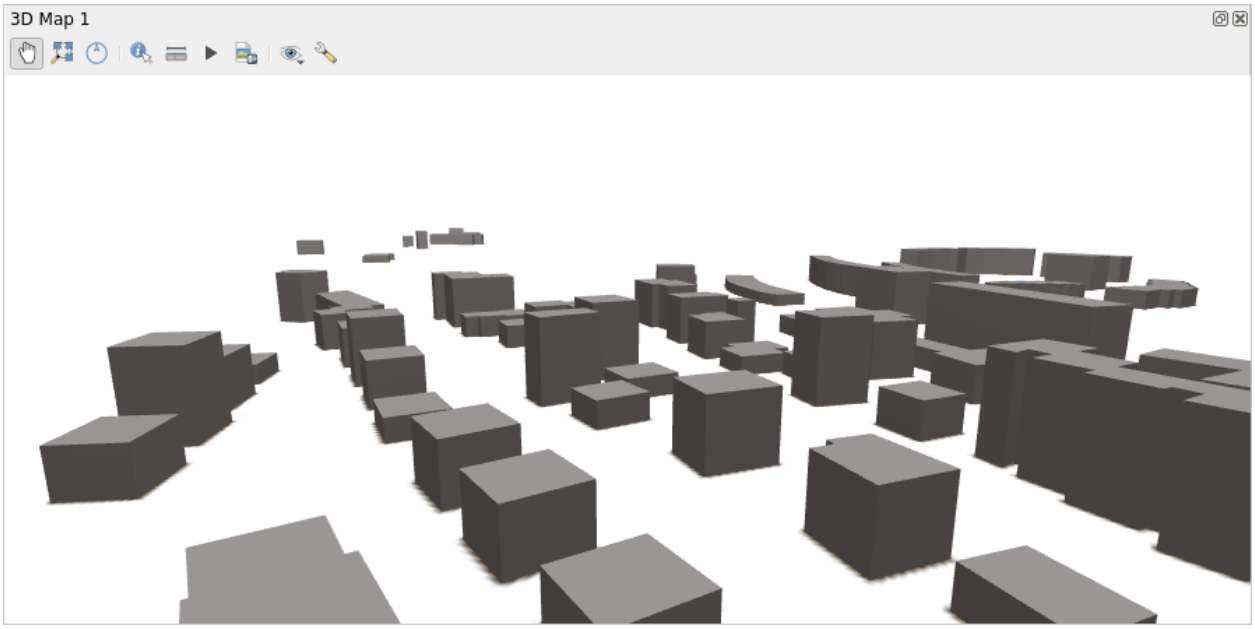


그림 12.27: 데이터 정의 압출

- *Width* 및 *Color* 로 *Edges* 를 정의할 수 있습니다.

12.4.4 응용 프로그램 예시

앞에서 설명한 설정들을 자세히 알아보려면, https://public.cloudmergin.com/projects/saber/luxembourg/tree 페이지에 있는 프로젝트 파일을 다운받아보십시오.

CHAPTER 13

데이터소스 관리

13.1 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:

- GeoPackage, GML, GeoJSON, GPX, KML, 쉼표 구분 값, ESRI 포맷 (shapefile, Geodatabase 등등), MapInfo 및 MicroStation 파일 포맷, AutoCAD DWG/DXF, GRASS, 그 외에도 많은 포맷들을 포함하는 벡터 데이터 유형을 지원합니다. 지원 벡터 유형 의 전체 목록을 읽어보세요.
- GeoTIFF, JPEG, ASCII Gridded XYZ, MBTiles, R 또는 Idrisi 래스터, GDAL Virtual, SRTM, Sentinel Data, ERDAS IMAGINE, ArcInfo Binary Grid, ArcInfo ASCII Grid, 그 외에도 많은 포맷들을 포함하는 래스터 데이터 유형을 지원합니다. 지원 래스터 유형 의 전체 목록을 읽어보세요.
- PostgreSQL/PostGIS, SQLite/SpatiaLite, Oracle, DB2 또는 MSSQL Spatial, MySQL, 그 외에도 많은 데이터베이스를 포함하는 데이터베이스 유형을 지원합니다.
- 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 를 참조하세요.
- 보존된 (archived) 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.

GDAL 과 QGIS 자체 제공자는 벡터 포맷 80 개 이상, 래스터 포맷 140 개 이상을 지원하고 있습니다.

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 벡터 또는 래스터 파일 기반 데이터는 물론 QGIS 가 지원하는 데이터베이스나 웹 서비스도 열 수 있는 통합 인터페이스를 제공합니다. (그림 13.1 참조) *Settings* → *Options* → *General* 메뉴의 *Modeless data source manager dialog* 옵션을 켜면 모달리스 대화창으로, 끄면 모달 대화창으로 설정할 수 있습니다.

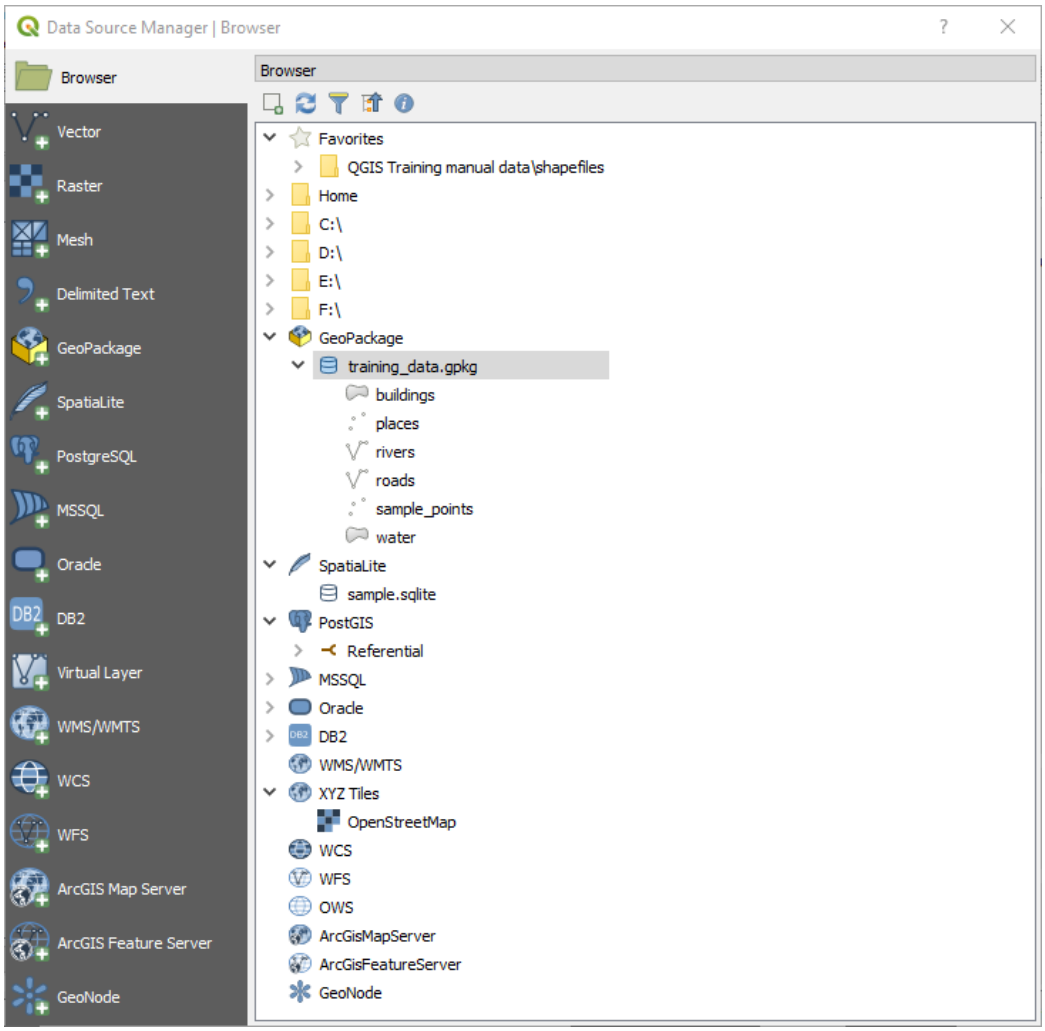


그림 13.1: QGIS 데이터소스 관리자 대화창

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.

13.1.1 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 13.2 그림의 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.
-  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. (공간 북마크 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*

- *PostGIS*
- *MSSQL*
- *Oracle*
- *DB2*
- *WMS/WMTS*
- *Vector Tiles*
- *XYZ Tiles*
- *WCS*
- *WFS/OGC API-Features*
- *OWS*
- *ArcGIS Map Service*
- *ArcGIS Feature Service*
- *GeoNode*

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 리프 (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:

- *Delete File "<name of file>"*…
- *Export Layer -> To File*…
- *Add Layer to Project*
- *Layer Properties*
- *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 그룹은 *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

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 명칭을 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 명칭을 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 명칭을 더블 클릭하거나, 범례에서 레이어 명칭을 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, 추출할 수도 (truncate) 있습니다.
- 폴더인 경우, 사용자의 즐겨찾기에 북마크해두고 탐색 트리에서 숨길 수 있습니다. *Settings → Options → Data Sources* 탭에서 이렇게 숨긴 풀더들을 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.

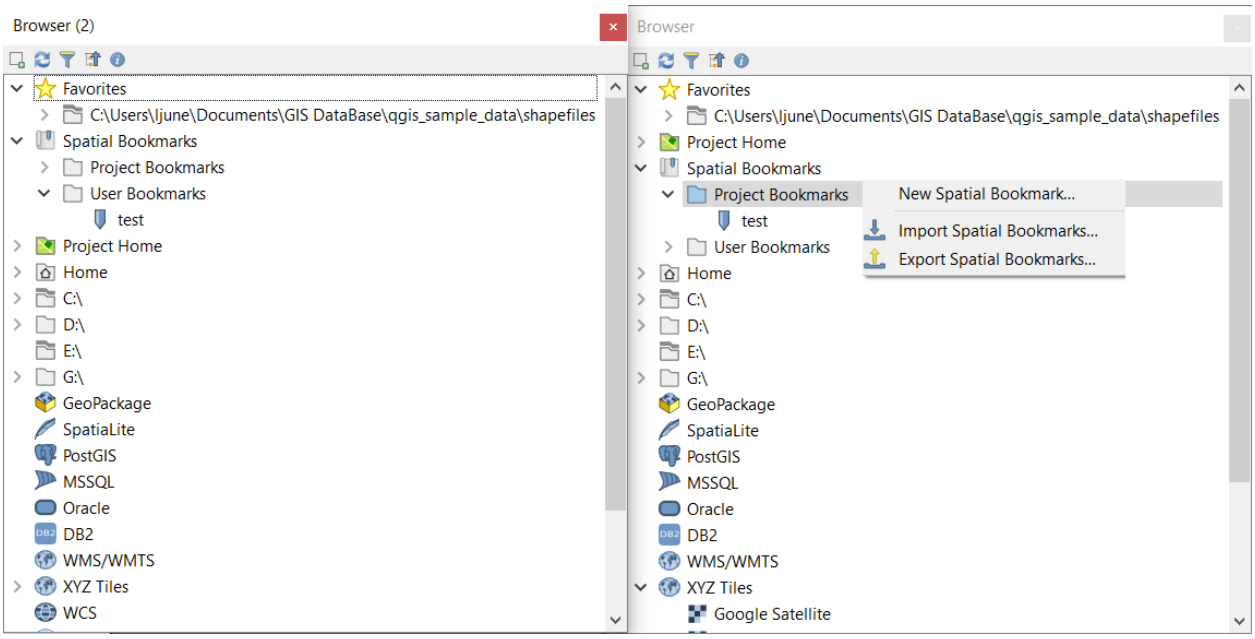


그림 13.2: 나란히 있는 QGIS 탐색기 패널

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.

13.1.2 데이터베이스 관리자

DB Manager 플러그인은 QGIS 가 지원하는 공간 데이터베이스 유형 (PostGIS, SpatiaLite, GeoPackage, Oracle Spatial, MSSQL, DB2, 가상 레이어) 을 통합하고 관리하기 위한 또다른 도구입니다. *Plugins → Manage and Install Plugins …*메뉴에서 이 플러그인을 활성화시킬 수 있습니다.

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.

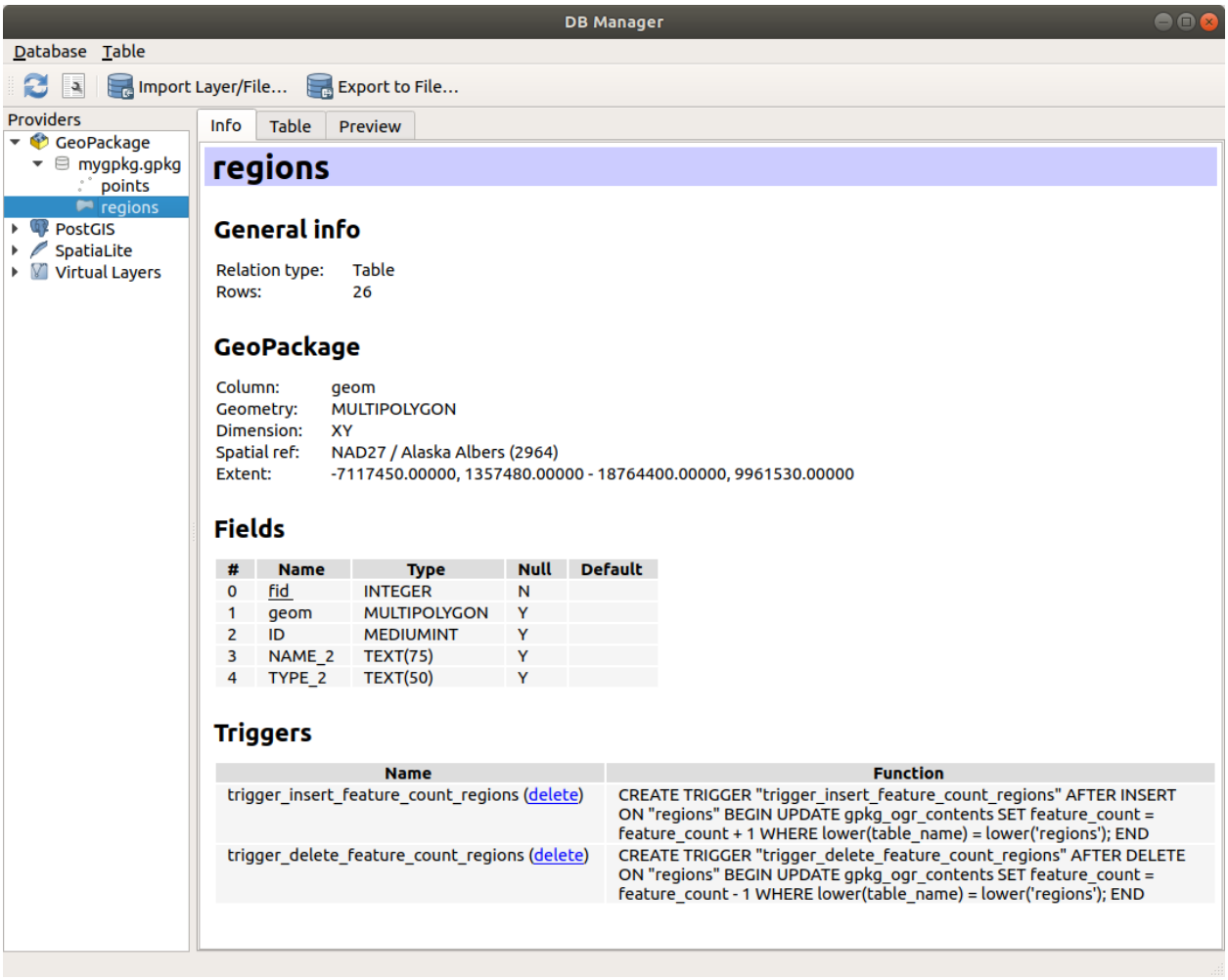


그림 13.3: 데이터베이스 관리자 대화창

13.1.3 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 Shapefile, 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.

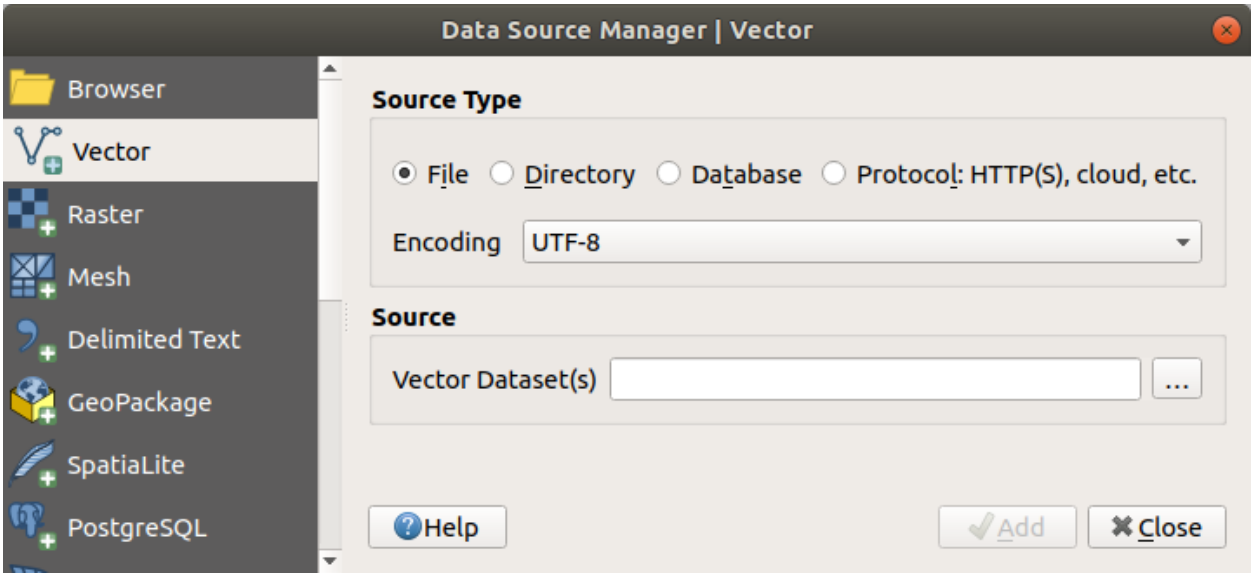


그림 13.4: 벡터 레이어 추가 대화창

   - 래스터 데이터 (GeoTiff, MBTiles, 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.
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. *Open* 을 클릭하면 *Data Source Manager* 대화창으로 선택한 파일을 불러들입니다.

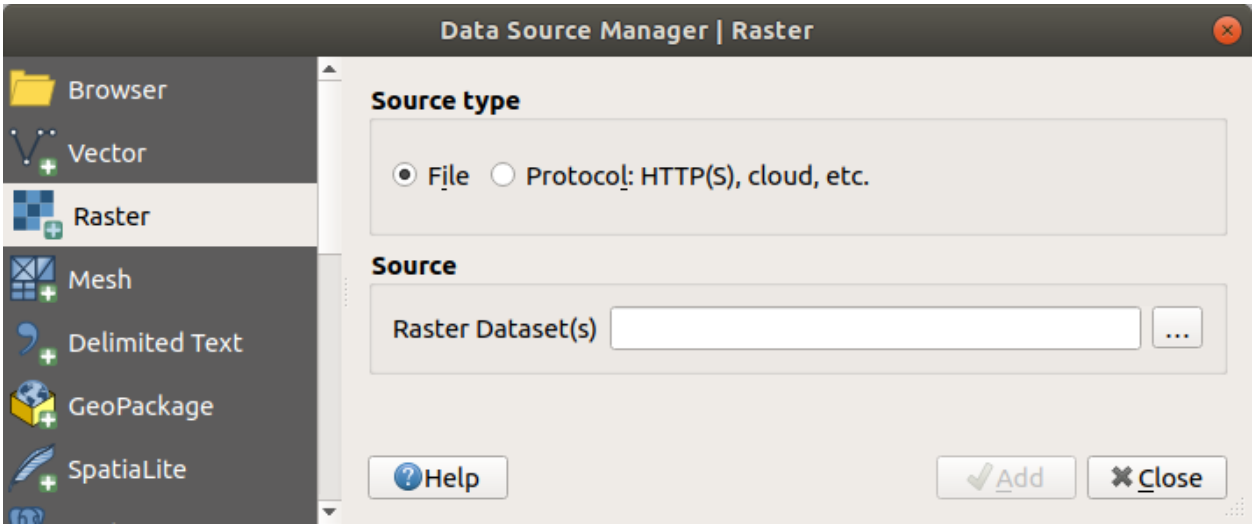


그림 13.5: 래스터 레이어 추가 대화창

6. *Add* 를 클릭하면 QGIS 로 파일을 불러와서 맵 뷰에 표시합니다. 그림 13.7 그림은 QGIS 가 `alaska.shp` 파일을 불러들인 모습입니다.

참고: 벡터 파일을 불러오는 경우 GDAL 드라이버가 열기 액션을 정의하는 옵션을 제공합니다. 이 옵션들은 벡터 파일을 선택했을 때 나타납니다. https://gdal.org/drivers/vector/ 에서 이 옵션들을 자세히 설명하고 있습니다.

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.

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 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`, `MSSQL` 은 물론 `PostgreSQL` 또는 `MySQL` 등 많은 데이터베이스 유형을 선택할 수 있습니다.

  New 버튼을 누르면 *Create a New OGR Database Connection* 대화창이 열립니다. 저장된 연결 생성하기 에서 이 대화창의 파라미터에 대한 내용을 찾아 볼 수 있습니다. *Open* 을 클릭하면, 예를 들어 PostGIS 활성 데이터베이스에서 사용 가능한 데이터베이스 테이블을 고를 수 있습니다.
- *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.
  - `AWS S3`, `Google Cloud Storage`, `Microsoft Azure Blob`, `Alibaba OSS Cloud`, `Open Stack Swift Storage` 와 같은 클라우드 저장소: *Bucket or container* 와 *Object key* 를 입력해야

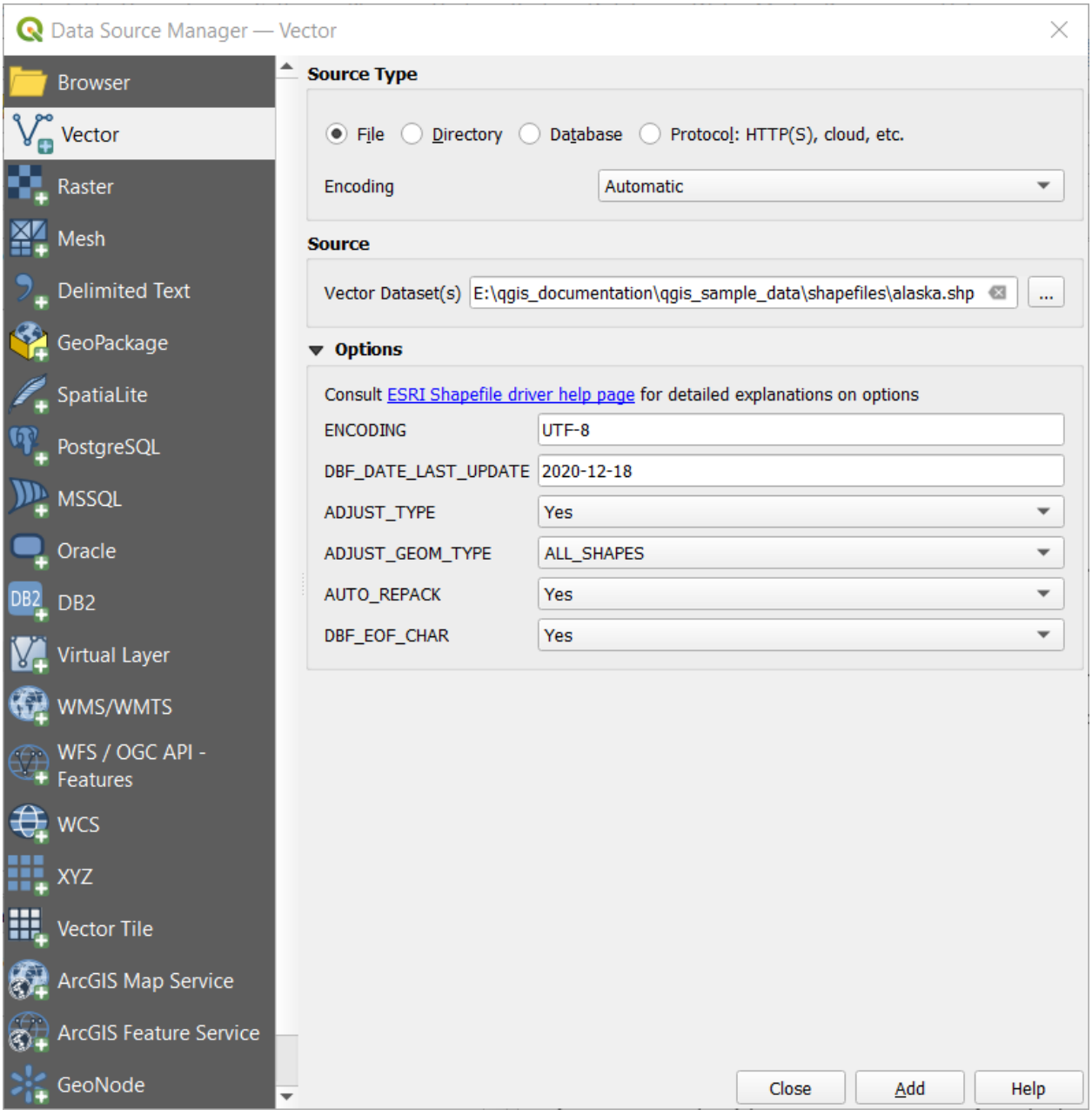


그림 13.6: 열기 옵션으로 shapefile 불러오기

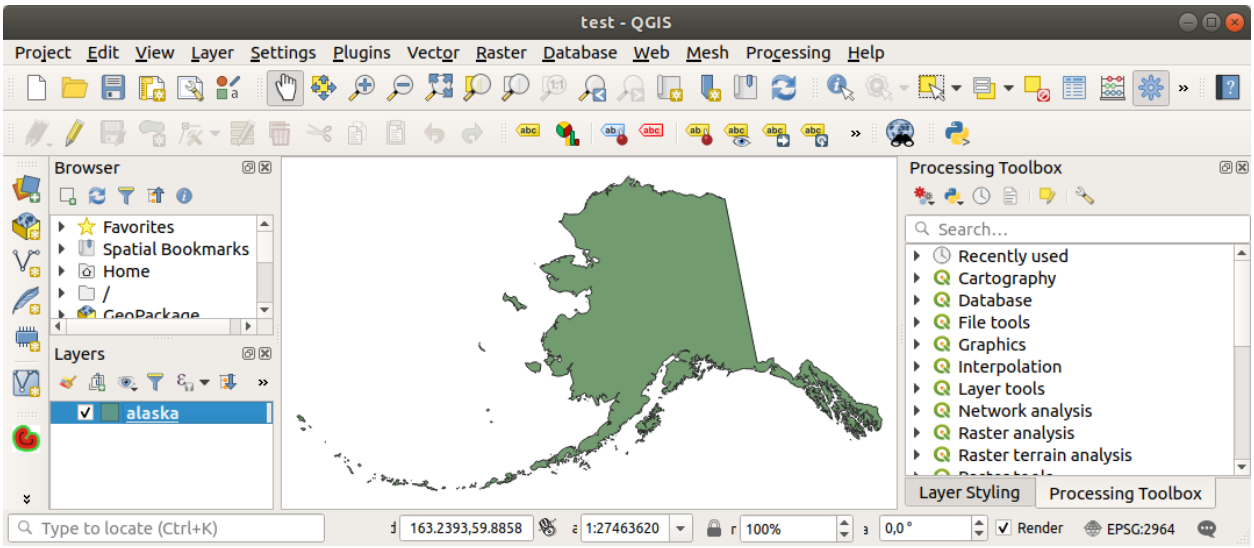


그림 13.7: 알래스카 shapefile 을 불러들인 QGIS

합니다.

- `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.

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, 경계 (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. 레이어를 선택한 다음 *Add* 를 클릭하십시오. 자체 메시 렌더링을 사용해서 레이어를 추가할 것입니다.

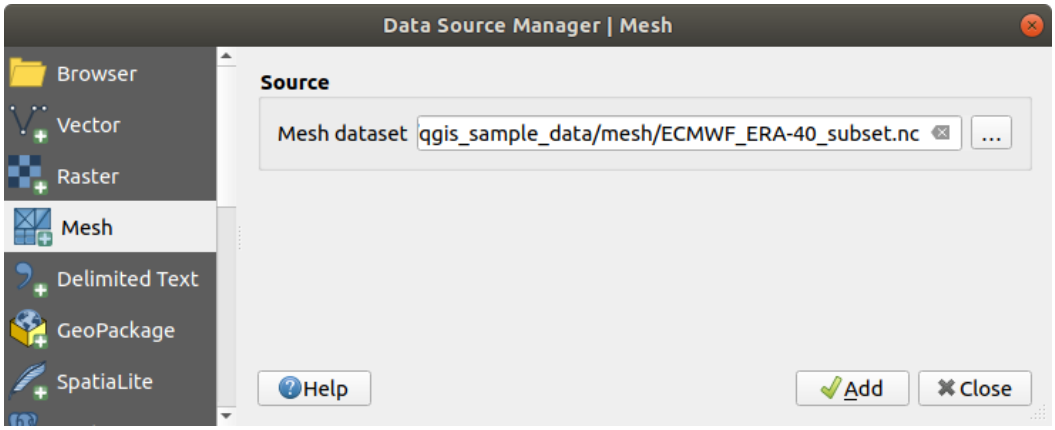


그림 13.8: 데이터소스 관리자의 메시 탭

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. *Layer name* 필드에 프로젝트에서 레이어용으로 사용할 명칭을 (예를 들어 `Elevation` 을) 입력하십시오.
5. 다음에 설명하는 내용처럼 사용자 데이터셋 및 필요를 만족시키는 환경을 설정하십시오.

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.

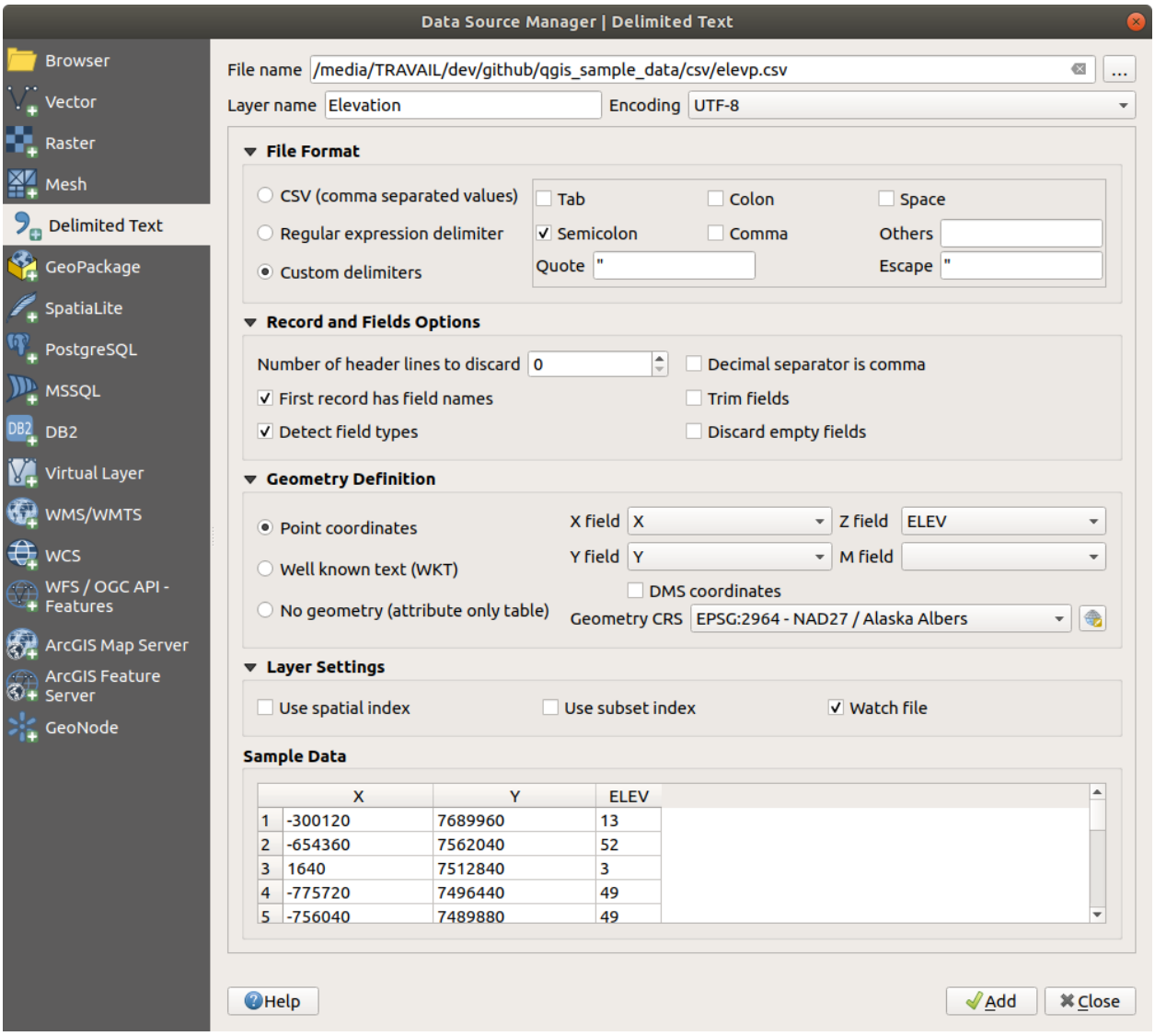


그림 13.9: 구분 텍스트 대화창

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.

파싱 작업자 속성을 설정하는 대로, 대화창 하단에 예시 데이터 미리보기가 업데이트됩니다.

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 명칭의 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.

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. 데이터를 저장할 새 GeoPackage 파일 같은, *Target package* 의 위치를 입력하십시오. 기존 파일을 입력하면 해당 파일을 덮어 쓸 것입니다.
2. 그리기 파일에 있는 데이터의 좌표계를 지정하십시오.
3. 그리기 파일의 블록 (block) 을 일반적인 요소로 가져오려면 ☑ *Expand block references* 를 체크하십시오.
4. 가져온 레이어를 curved 만곡 도형 유형으로 승격시키려면 ☑ *Use curves* 를 체크하십시오.
5. *Import* 버튼을 클릭해서 사용할 DWG/DXF 파일을 (GeoPackage 당 하나씩) 선택하십시오. 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. 맨 위 *Group name* 에서, 프로젝트에 그리기 파일을 그룹화할 그룹명을 설정하십시오.
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.

OpenStreetMap 벡터 가져오기

OpenStreetMap 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.

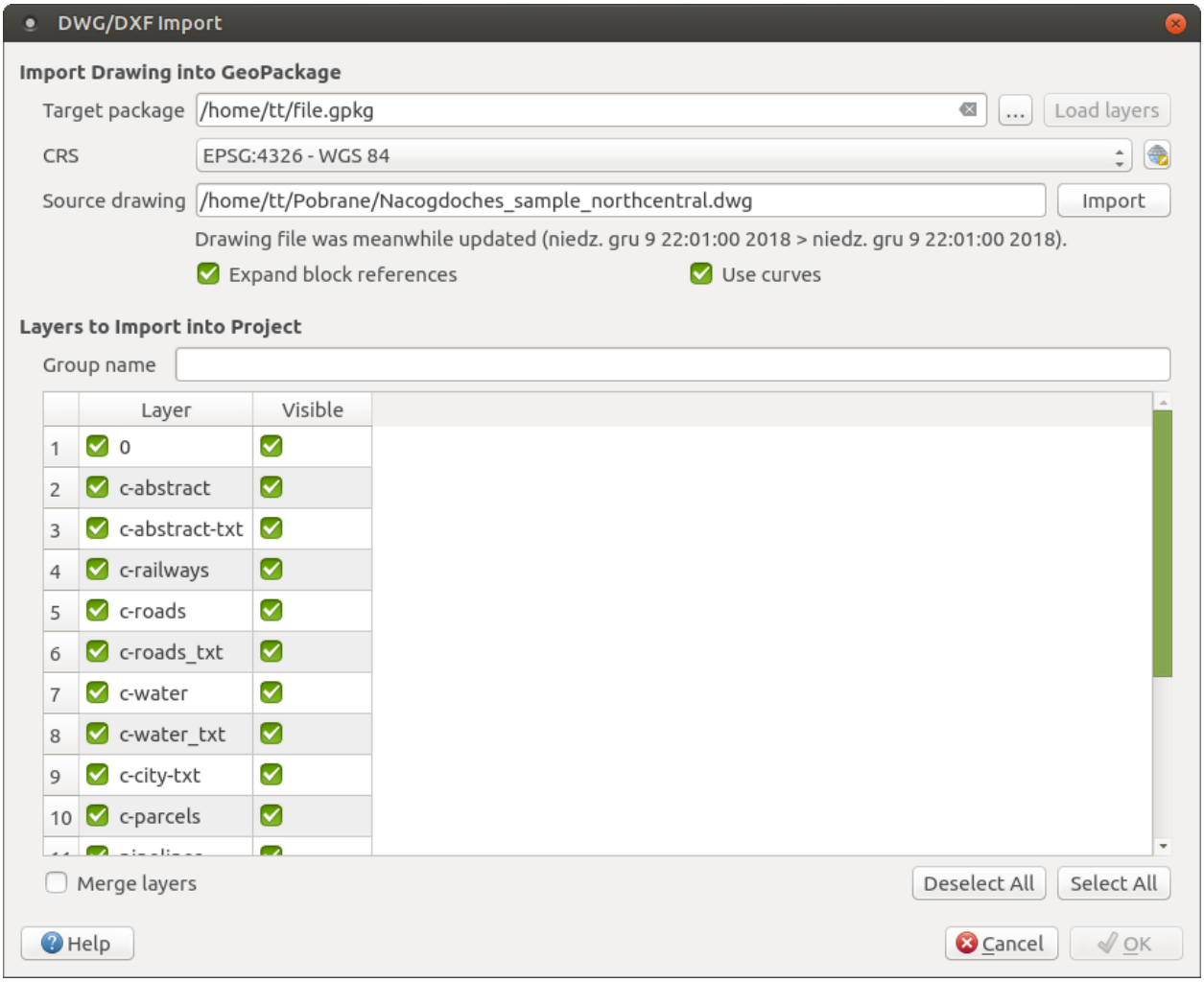


그림 13.10: DWG/DXF 파일 가져오기 대화창

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

핵심 플러그인인 `GPS Tools` 를 이용하면 QGIS 에 GPS 데이터를 불러올 수 있습니다. *GPS 플러그인* 에서 해당 지침을 설명하고 있습니다.

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 PostGIS Layer*…또는 `Ctrl+Shift+D`
-  *Add MSSQL Spatial Layer*
-  *Add Oracle Spatial Layer*…또는 `Ctrl+Shift+O`
-  *Add DB2 Spatial Layer*…또는 `Ctrl+Shift+2`

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.

연결 대화창 대부분은 PostGreSQL 데이터베이스 도구를 예시 삼아 다음에 설명할 공통 기초를 따릅니다. 다른 제공자들에 특화된 추가 설정에 대한 설명은 각각 다음 절에서 찾아볼 수 있습니다:

- *MSSQL Spatial* 을 연결하기
- *Oracle* 데이터베이스를 연결하기
- *DB2* 데이터베이스를 연결하기

처음으로 PostGIS 데이터소스를 이용하는 경우, 데이터를 담고 있는 데이터베이스로의 연결을 생성해야만 합니다. 먼저 앞에서 설명한 대로 올바른 버튼을 클릭해서 *Add PostGIS Table(s)* 대화창을 여십시오. (그림 13.12 참조) 연결 관리자에 접근하려면, *New* 버튼을 클릭해서 *Create a New PostGIS Connection* 대화창을 여십시오.

PostGIS 연결에 필요한 파라미터를 설명하겠습니다. 다른 데이터베이스 유형의 경우, 특정 연결 요구사항 에서 차이점들을 참조하세요.

- *Name*: 해당 연결을 위한 명칭입니다. *Database* 파라미터와 동일해도 됩니다.
- *Service*: 호스트명/포트 (및 어쩌면 데이터베이스) 대신 쓸 수 있는 서비스 파라미터입니다. `pg_service.conf` 파일 안에 정의할 수도 있습니다. 더 자세한 내용은*PostgreSQL* 서비스 연결 파일 을 참조하세요.
- *Host*: 데이터베이스 호스트의 명칭입니다. 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*: PostgreSQL 데이터베이스 서버가 열어 놓은 포트 번호입니다. PostGIS 용 기본 포트는 `5432` 입니다.
- *Database*: 데이터베이스의 명칭입니다.
- *SSL mode*: SSL 암호화 설정입니다. 다음 옵션 가운데 하나를 선택할 수 있습니다:
  - *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.
- *Authentication*: 기본적인 인증 정보
  - *User name*: 데이터베이스에 로그인하는 데 쓰이는 사용자명
  - *Password*: 데이터베이스에 연결하기 위해 사용자명 과 함께 쓰이는 비밀번호

  `User name` 과 `Password` 파라미터 가운데 하나를, 또는 둘 다 저장할 수 있습니다. 이 내용은 해당 데이터베이스에 연결해야 할 때마다 기본적으로 사용됩니다. 이 내용을 저장하지 않는다면 다음 QGIS 세션에서 데이터베이스에 연결할 때 인증 정보를 제공하라는 알림을 띄울 것입니다. 현재 세션을 종료할 때까지, QGIS 는 사용자가 입력한 연결 파라미터를 임시 내부 캐시에 저장해서 동일한 데이터베이스가 이를 요구할때마다 사용자명/비밀번호를 반환합니다.

> **경고: QGIS 사용자 설정 및 보안**
>
> *Authentication* 탭에서 **사용자명** 과 **비밀번호** 를 저장하면 연결 환경 설정에 보안되지 않은 인증 정보를 저장하게 됩니다. 예를 들면, 다른 사용자와 프로젝트 파일을 공유하는 경우, 이런 **인증 정보가 노출될 것입니다**. 따라서 인증 환경 설정 (*Configurations* 탭 – 더 자세한 내용은인증 시스템 참조) 또는 서비스

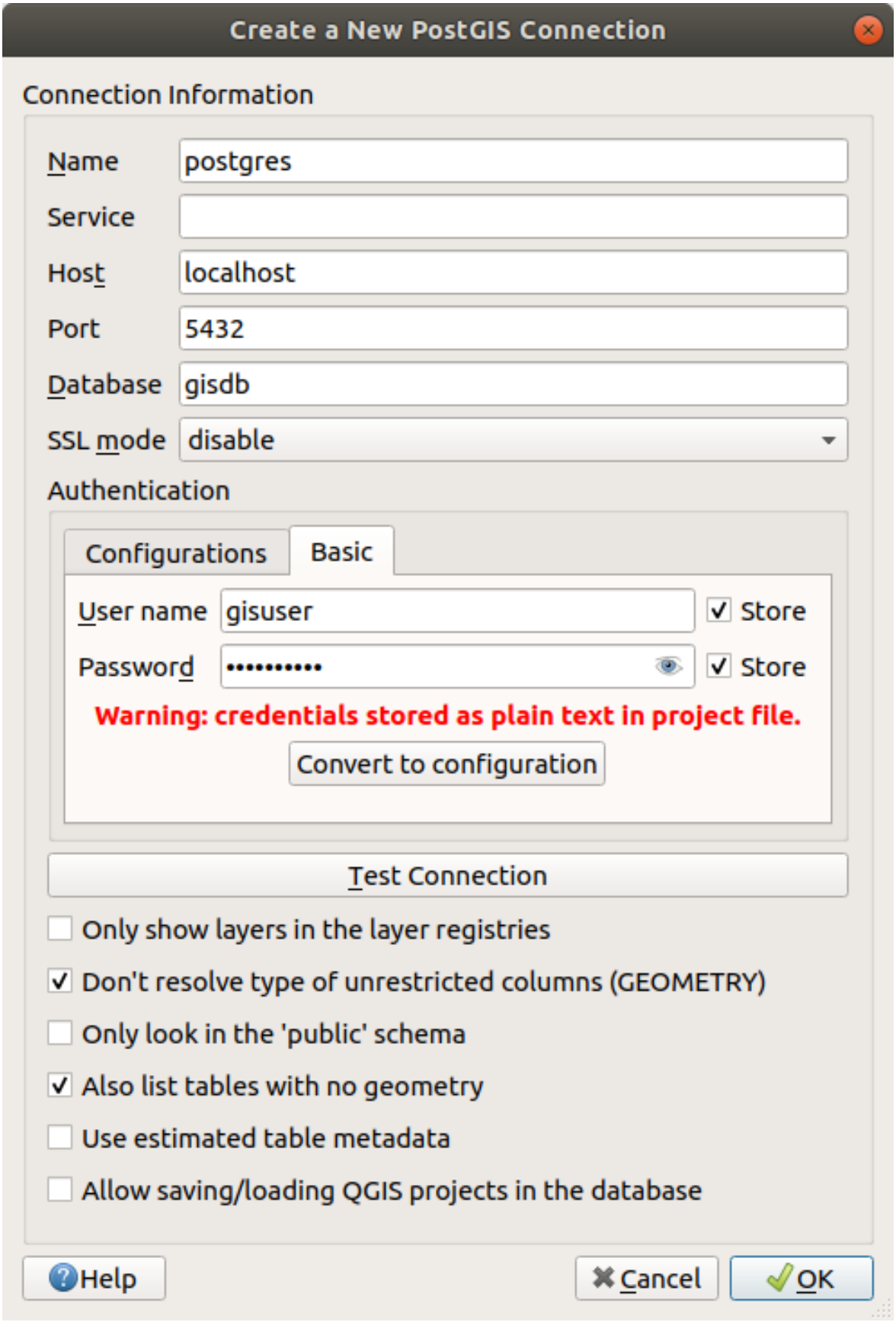


그림 13.11: 새 PostGIS 연결 생성 대화창

> 연결 파일 (예시를 보려면*PostgreSQL 서비스 연결 파일* 참조) 에 사용자 인증 정보를 대신 저장하는 편이 좋습니다.

- *Authentication*: 인증 환경을 설정합니다. 인증 환경 설정을 선택하십시오.  버튼으로 환경 설정을 추가할 수 있습니다. 다음 가운데 하나를 선택할 수 있습니다:
  - 기본 인증
  - PKI PKCS#12 인증
  - PKI 경로 인증
  - 개인식별인증서 (identity certificate) 를 저장한 PKI

데이터베이스 유형에 따라 다음 체크박스들을 선택적으로 활성화할 수 있습니다:

-  *Only show layers in the layer registries*
-  *Don't resolve type of unrestricted columns (GEOMETRY)*
-  *Only look in the 'public' schema*
-  *Also list tables with no geometry*
-  *Use estimated table metadata*
-  *Allow saving/loading QGIS projects in the database* - 자세한 내용은여기를 참조

팁: 수행 속도 향상을 위해 예측된 테이블 메타데이터 이용

레이어 초기화 시, 데이터베이스 테이블에 저장된 도형의 특성을 밝히기 위해 다양한 쿼리가 필요할 수도 있습니다. *Use estimated table metadata* 옵션을 체크하면, 전체 테이블을 쿼리하는 대신 일부 행만 검사하고 테이블 통계를 이용하는 쿼리를 실행합니다. 대용량 데이터셋의 경우 작업 속도를 현저하게 높일 수 있지만, 레이어 특성을 부정확하게 표시할 수도 있고 (예를 들어 필터링된 레이어의 피처 개수가 부정확할 것입니다) 유일 (unique) 하다고 간주한 열이 사실은 그렇지 않은 경우 이상하게 동작할 가능성조차 있습니다.

모든 파라미터와 옵션을 설정하고 나면, *Test Connection* 버튼을 클릭해서 연결을 검증하거나 *OK* 를 눌러 적용시킬 수 있습니다. 이제 *Add PostGIS Table(s)* 에서 *Connect* 를 클릭하면 대화창에 (그림 13.12 처럼) 선택한 데이터베이스에 있는 테이블이 표시됩니다.

특정 연결 요구사항

데이터베이스 유형의 특성이 각각 다르기 때문에, 모든 데이터베이스 연결 옵션이 동일하지는 않습니다. 다음은 데이터베이스별 특화 옵션에 대한 설명입니다.

PostgreSQL 서비스 연결 파일

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 명령어를 실행하기 전에 `export PGSERVICE=water_service` 같은 명령어를 입력해야 합니다.

여기 에서 모든 PostgreSQL 파라미터들을 찾아볼 수 있습니다.

참고: 서비스 파일에 비밀번호를 저장하고 싶지 않다면 .pg_pass 옵션을 이용할 수 있습니다.

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.

`PGSERVICEFILE` 환경 변수를 생성해서 서비스 파일의 위치를 지정할 수 있습니다. (예를 들어 * 닉스 운영체제에서 일시적으로 `PGSERVICEFILE` 변수를 설정하려면 `export PGSERVICEFILE=/home/web/.pg_service.conf` 명령어를 실행하십시오.)

`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 명칭의 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 데이터베이스를 연결하기

Oracle Spatial 의 공간 기능들을 이용하면 Oracle 데이터베이스 내부에서 지리 및 위치 데이터를 자체 유형으로 관리할 수 있습니다. 저장된 연결 생성하기 에 있는 몇몇 옵션 이외에도, 연결 대화창은 다음을 제안합니다:

- **Database**: Oracle 인스턴스의 SID 또는 SERVICE_NAME 입니다.
- **Port**: Oracle 데이터베이스 서버가 열어 놓은 포트 번호입니다. 기본 포트는 `1521` 입니다.
- **Options**: Oracle 연결 특화 옵션입니다. (예: OCI_ATTR_PREFETCH_ROWS, OCI_ATTR_PREFETCH_MEMORY) 이 옵션 문자열의 서식은 옵션 명칭을 쌍반점으로 구분한 목록이거나, 옵션 = 값 쌍의 목록입니다.
- **Workspace**: 전환할 작업 공간입니다.
- **Schema**: 데이터를 저장할 스키마입니다.

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*

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.

DB2 데이터베이스를 연결하기

저장된 연결 생성하기 에서 설명하는 몇몇 옵션 이외에도, ODBC 에 정의된 *Service/DSN* 명칭을 이용하거나 또는 *Driver*, *Host* 및 *Port* 정보를 이용해서 DB2 데이터베이스 (자세한 정보는*DB2 Spatial* 레이어 참조) 로의 연결을 지정할 수 있습니다.

ODBC **Service/DSN** 연결은 ODBC 에 정의된 서비스명을 필요로 합니다.

드라이버/호스트/포트 연결은 다음을 필요로 합니다:

- **Driver**: DB2 드라이버의 명칭입니다. 일반적으로 `IBM DB2 ODBC DRIVER` 일 겁니다.
- **DB2 Host**: 데이터베이스 호스트의 명칭입니다. 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.
- **DB2 Port**: DB2 데이터베이스 서버가 열어 놓은 포트 번호입니다. 기본 DB2 LUW 포트는 `50000` 입니다. 기본 DB2 z/OS 포트는 `446` 입니다.

팁: DB2 공간 레이어

DB2GSE.ST_GEOMETRY_COLUMNS 뷰에 있는 행이 DB2 공간 레이어를 정의합니다.

참고: QGIS 에서 DB2 공간 테이블을 효과적으로 작업하려면, 테이블이 *PRIMARY KEY* 로 정의된 *INTEGER* 또는 *BIGINT* 열을 가지고 있어야만 합니다. 새 피처를 추가하려 할 경우, 이 열은 *GENERATED* 특성도 가지고 있어야 합니다.

공간 열이 특정 공간 참조 식별자 (대부분의 경우 WGS84 좌표를 뜻하는 `4326`) 로 등록돼 있으면 편리합니다. `ST_Register_Spatial_Column` 저장 프로시저 를 호출하면 공간 열을 등록할 수 있습니다.

MSSQL Spatial 을 연결하기

저장된 연결 생성하기 에 있는 몇몇 옵션 이외에도, 새 MSSQL 연결 생성 대화창에서 **Provider/DSN** 명칭을 입력할 수 있습니다. 사용할 수 있는 데이터베이스도 표시할 수 있습니다.

데이터베이스 레이어 불러오기

사용자가 데이터베이스 연결을 하나 이상 정의하고 나면 (저장된 연결 생성하기 참조) 데이터베이스에서 레이어를 불러올 수 있습니다. 물론 불러올 데이터가 있어야만 합니다. 예를 들어 PostGIS 데이터베이스로 데이터를 가져오는 작업에 대한 내용은*PostgreSQL* 로 데이터 가져오기 를 참조하십시오.

데이터베이스에서 레이어를 불러오려면 다음 단계를 거쳐야 합니다:

1. "Add <database> table(s)"대화창을 엽니다. (저장된 연결 생성하기 참조)
2. 드롭다운 목록에서 연결을 선택한 다음 *Connect* 를 클릭합니다.
3. ☑ *Also list tables with no geometry* 체크박스를 활성화하거나 비활성화합니다.
4. ☑ *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.
5. 이용할 수 있는 레이어 목록에서 사용자가 추가하려 하는 레이어 (들) 를 찾습니다.

6. 그 레이어를 클릭해서 선택하십시오. `Shift` 또는 `Ctrl` 키를 누른 채 클릭하면 레이어 여러 개를 선택할 수 있습니다.
7. 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.
8. `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.
9. *Add* 버튼을 클릭해서 맵에 레이어를 추가합니다.

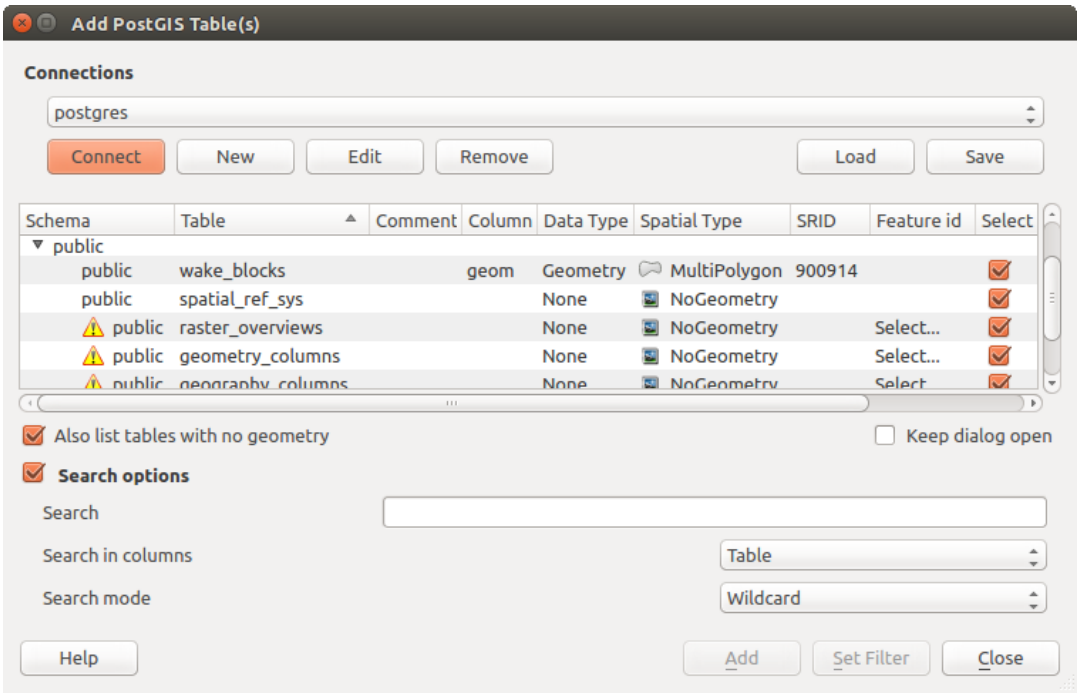


그림 13.12: PostGIS 테이블 추가 대화창

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.

13.1.4 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)

13.1.5 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.

13.1.6 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

Browser 의 *Vector Tiles* 최상위 수준 항목에서 벡터 타일 서비스를 찾을 수 있습니다. 오른쪽 클릭으로 컨텍스트 메뉴를 열어 *New Generic Connection* …을 선택하면 벡터 타일 서비스를 추가할 수 있습니다. *Name* 및 *URL* 을 추가해서 서비스를 설정하십시오. 벡터 타일 서비스는 `.pbf` 포맷의 타일을 제공해야만 합니다. 이 대화창은 ☑ *Min. Zoom Level* 과 ☑ *Max. Zoom Level* 을 정의하기 위한 메뉴 2 개를 제공합니다. 벡터 타일이 피라미드 구조로 되어 있기 때문입니다. 이 옵션들을 사용해서 타일 피라미드로부터 레이어를 개별적으로 생성할 수 있습니다. QGIS 는 이 레이어들을 사용해서 벡터 타일을 렌더링하게 됩니다. (OpenStreetMap 벡터 타일이 사용하는) 메르카토르 투영법의 경우, 확대/축소 0 수준은 1:500,000,000 축척의 전세계를 의미합니다. 확대/축소 14 수준은 1:35,000 축척을 의미합니다. 그림 13.13 은 MapTiler Planet 벡터 타일 서비스를 환경 설정하는 대화창을 보여주고 있습니다.

New ArcGIS Vector Tile Service Connection …메뉴를 선택하면 ArcGIS 벡터 타일 서비스에 연결할 수 있습니다.

XYZ 타일 서비스 사용하기

Browser 의 *XYZ Tiles* 최상위 수준 항목에서 XYZ 타일 서비스를 찾을 수 있습니다. OpenStreetMap XYZ 타일 서비스가 기본적으로 환경 설정돼 있습니다. XYZ 타일의 (오른쪽 클릭으로 열리는) 컨텍스트 메뉴에서 *New Connection* 을 선택하면 XYZ 타일 프로토콜을 사용하는 다른 서비스를 추가할 수 있습니다. 그림 13.14 는 OpenStreetMap XYZ 타일 서비스 환경 설정을 담고 있는 대화창입니다.

이 환경 설정을 XML 로 (*Save Connections*) 저장할 수 있고, 이를 컨텍스트 메뉴를 통해 (*Load Connections*) 불러올 수 있습니다. 인증 환경 설정도 지원합니다. OpenStreetMap 용 XML 파일은 다음과 비슷한 내용을 담고 있습니다:

```
<!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>
```

XYZ 타일 서비스 연결을 설정한 다음, 해당 항목을 오른쪽 클릭하면:

- *Edit*…: XYZ 연결 설정을 편집합니다.
- *Delete*: 연결을 삭제합니다.
- *Export layer…→ To File*: XYZ 타일 레이어를래스터로 저장 합니다.

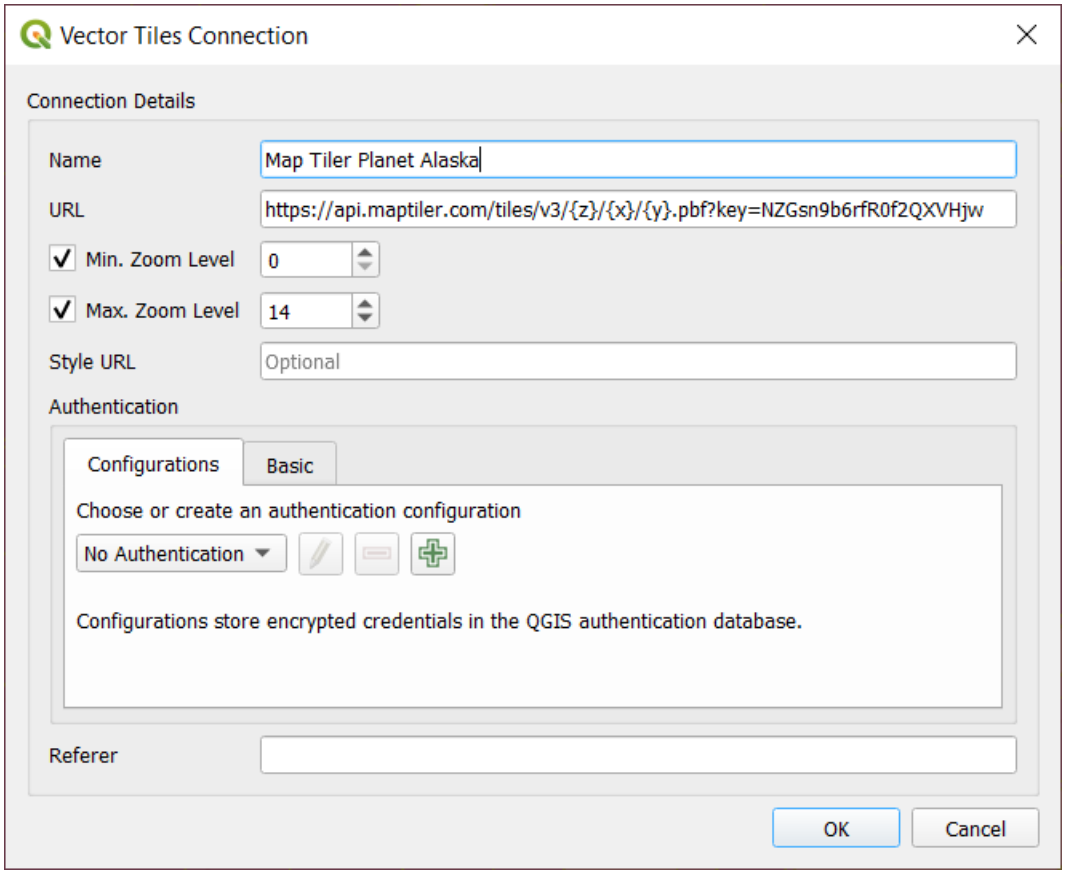


그림 13.13: 벡터 타일 - Maptiler Planet 환경 설정

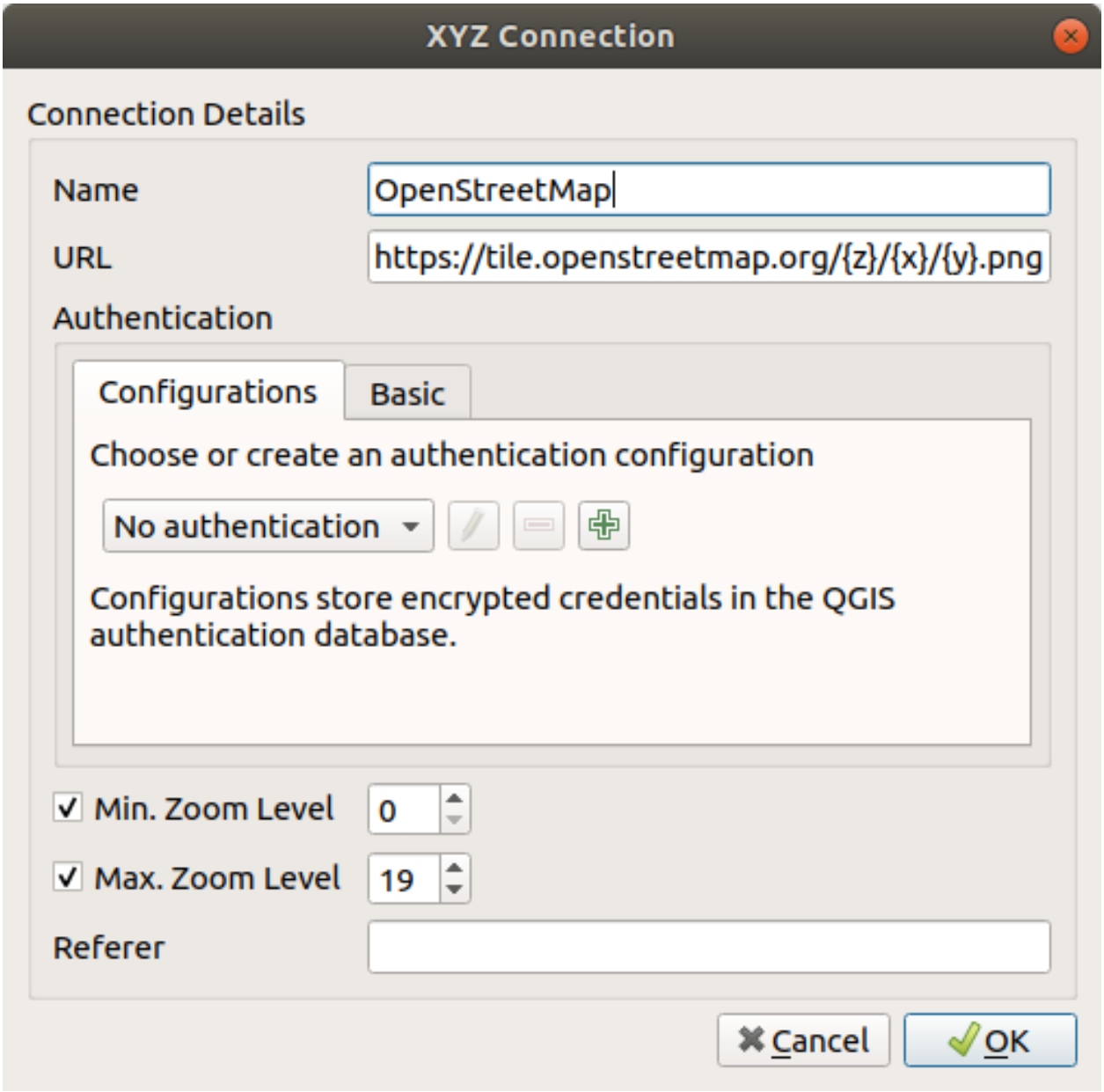


그림 13.14: XYZ 타일 - OpenStreetMap 환경 설정

- *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 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

13.2 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.

13.2.1 새 벡터 레이어 생성

QGIS 는 서로 다른 포맷들로 새 레이어를 생성할 수 있습니다. QGIS 는 GeoPackage, Shapefile, 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

새 GeoPackage 레이어를 생성하려면 *Layer* → *Create Layer* → 메뉴 또는 *Data Source Manager* 툴바에서  *New GeoPackage Layer*…를 선택하십시오. 그림 13.15 와 같은 *New GeoPackage Layer* 대화창이 열립니다.

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 명칭에 적합한 확장자를 자동적으로 추가할 것입니다.
2. 새 레이어/테이블의 명칭을 (*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.

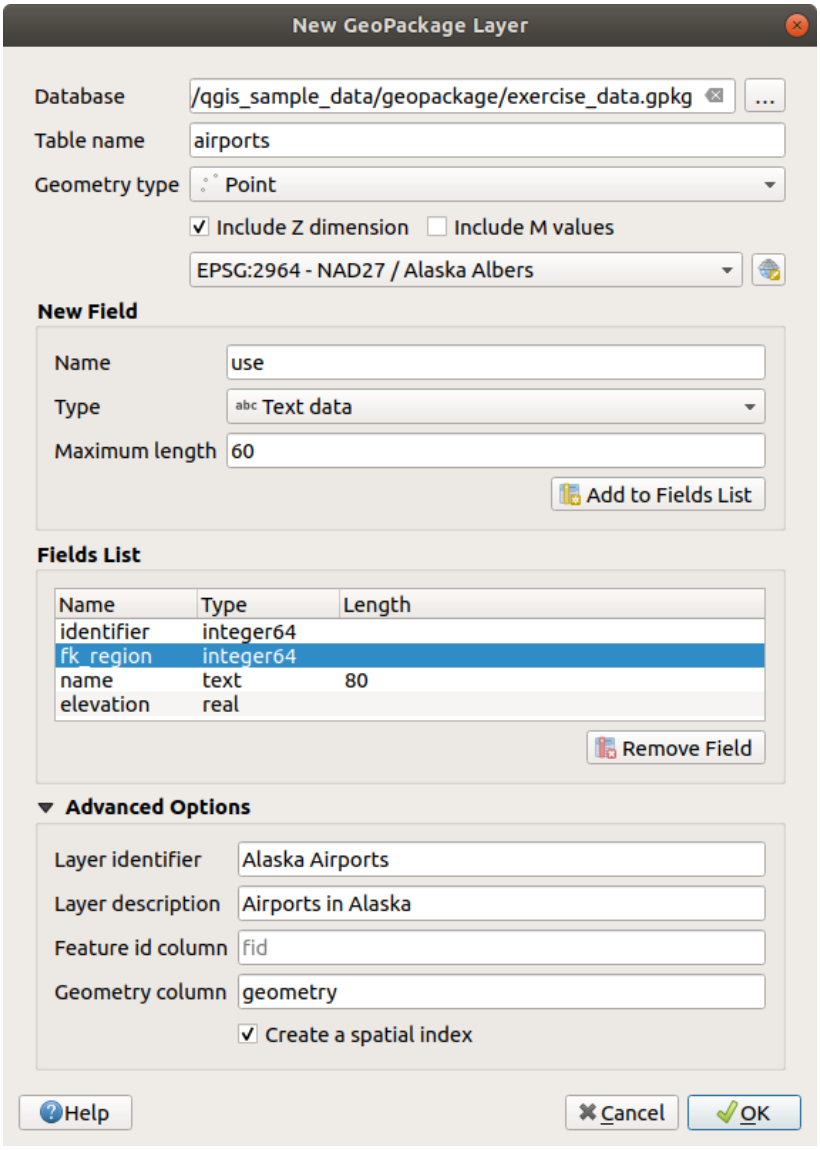


그림 13.15: 새 GeoPackage 레이어 생성 대화창

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. 필요한 모든 필드를 정확하게 추가했다면 *OK* 를 클릭하십시오. QGIS 가 범례에 새 레이어를 추가할 것입니다. 기존 레이어 디지타이징 절에서 설명하는 대로 새 레이어를 편집할 수 있습니다.

GeoPackage 레이어 생성 시 QGIS 는 레이어의 기본 키로 동작하는 `fid` 라는 명칭의 *Feature id column* 을 기본적으로 생성합니다. 이 명칭은 수정할 수 있습니다. 도형 필드가 있는 경우, 그 명칭은 `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 shapefile 레이어 생성하기

새 ESRI Shapefile 포맷 레이어를 생성하려면 *Layer* → *Create Layer* → 메뉴 또는 *Data Source Manager* 툴바에서 *New Shapefile Layer…*를 선택하십시오. 그림 13.16 과 같은 *New Shapefile Layer* 대화창이 열립니다.

1. *File name* 옆에 있는 …버튼을 사용해서 경로 및 파일명을 지정하십시오. QGIS 가 사용자가 지정한 명칭에 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. *Type*: 데이터 유형을 선택하십시오. *Decimal number*, *Whole number*, *Text data* 그리고 *Date* 속성만 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. 필요한 모든 필드를 정확하게 추가했다면 *OK* 를 클릭하십시오. QGIS 가 범례에 새 레이어를 추가할 것입니다. 기존 레이어 디지타이징 절에서 설명하는 대로 새 레이어를 편집할 수 있습니다.

기본적으로 `id` 라는 첫번째 정수열을 추가하지만, 이를 제거할 수 있습니다.

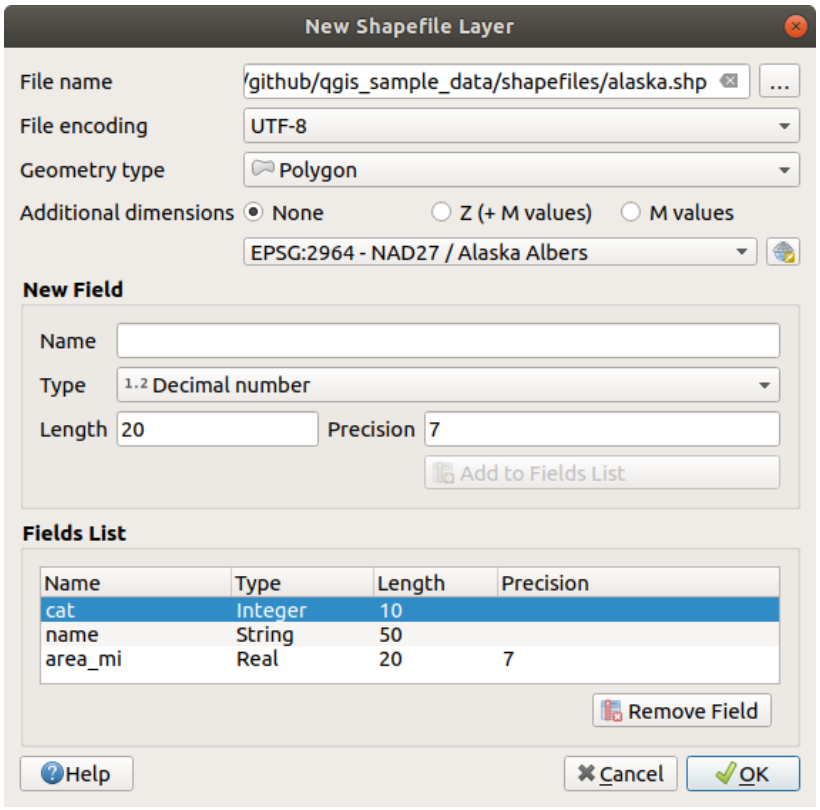


그림 13.16: 새 shapefile 레이어 생성 대화창

새 SpatiaLite 레이어 생성하기

새 SpatiaLite 레이어를 생성하려면 *Layer* → *Create Layer* → 메뉴 또는 *Data Source Manager* 툴바에서 *New SpatiaLite Layer*…를 선택하십시오. 그림 13.17 과 같은 *New SpatiaLite Layer* 대화창이 열립니다.

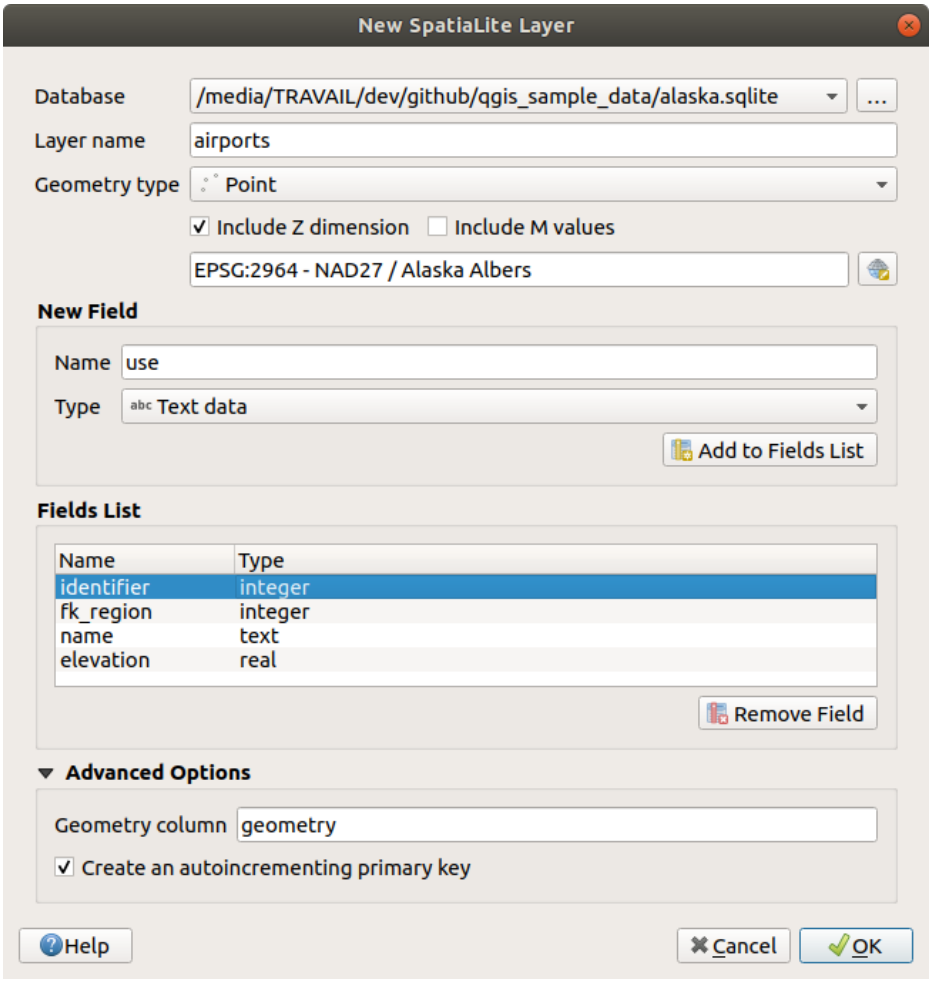


그림 13.17: 새 SpatiaLite 레이어 생성 대화창

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 명칭에 적합한 확장자를 자동적으로 추가할 것입니다.
2. 새 레이어용 명칭을 (*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.

생성 중인 레이어에 필드를 추가하려면:

1. *Name*: 필드명을 입력하십시오.
2. *Type*: 데이터 유형을 선택하십시오. *Text data*, *Whole number* 및 *Decimal number* 유형을 지원합니다.

3. *Add to Fields List* 버튼을 클릭하십시오.
4. 사용자가 추가해야 할 각 필드마다 앞의 단계를 반복합니다.
5. 필요한 모든 필드를 정확하게 추가했다면 *OK* 를 클릭하십시오. QGIS 가 범례에 새 레이어를 추가할 것입니다. 기존 레이어 디지타이징 절에서 설명하는 대로 새 레이어를 편집할 수 있습니다.

원한다면 guilabel:*Advanced Options* 부분에 있는 *Create an autoincrementing primary key* 를 체크해서 오름차순 기본 키를 생성할 수 있습니다. (기본 명칭이 `geometry` 인) *Geometry column* 을 재명명할 수도 있습니다.

SpatiaLite 레이어는데이터베이스 관리자 를 통해 더 심도 있게 관리할 수 있습니다.

새 GPX 레이어 생성하기

새 GPX 파일을 생성하려면, 먼저 GPS 플러그인을 불러와야 합니다. *Plugins* *Plugin Manager*…메뉴 옵션을 선택하면 플러그인 관리자 대화창이 열립니다. *GPS Tools* 체크박스를 활성화하십시오.

이 플러그인을 불러오고 나면, *Layer* 메뉴에서 *Create Layer* *Create new GPX Layer*…를 선택하십시오. 대화창에서 새 파일을 저장할 위치를 선택한 다음 *Save* 버튼을 누르십시오. *Layers Panel* 에 새 레이어 `waypoints`, `routes` 그리고 `tracks` 3 개가 추가됩니다.

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 사용하는 데 편리합니다.

새 임시 스크래치 레이어를 생성하려면 *Layer* *Create Layer* 메뉴 또는 *Data Source Manager* 툴바에서 *New Temporary Scratch Layer*…를 선택하십시오. 그림 13.18 과 같은 *New Temporary Scratch Layer* 대화창이 열립니다.

1. *Layer name*: 레이어 명칭을 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. 임시 스크래치 레이어 설정에 만족한다면 *OK* 를 클릭하십시오. QGIS 가 *Layers* 패널에 새 레이어를 추가할 것입니다. 기존 레이어 디지타이징 절에서 설명하는 대로 새 레이어를 편집할 수 있습니다.

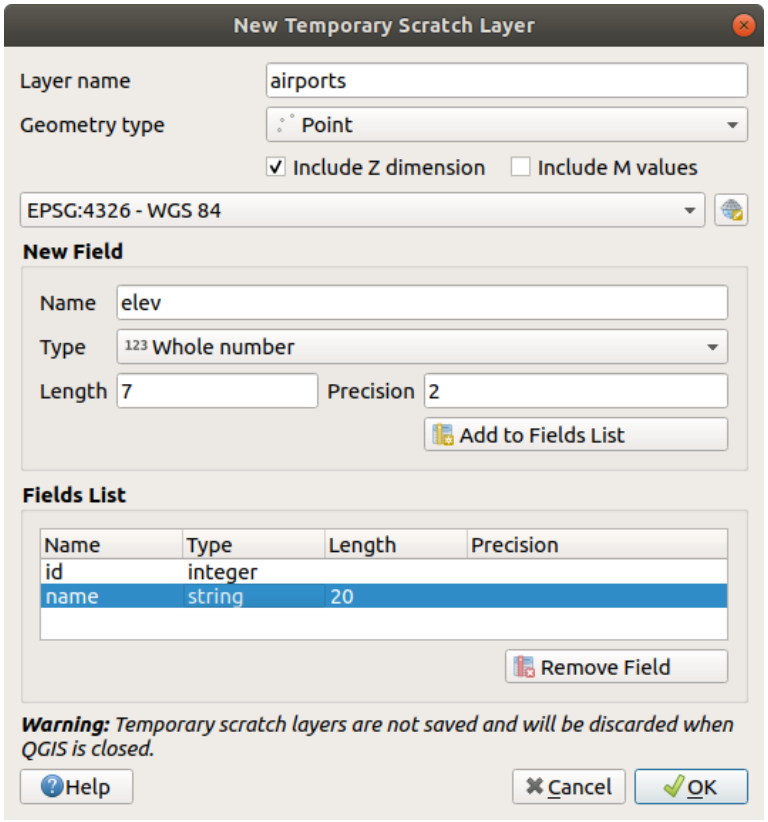


그림 13.18: 새 임시 스크래치 레이어 생성 대화창

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

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.

13.2.2 기존 레이어로부터 레이어 생성

Layer ▸ *Save As*…메뉴를 선택하거나 *Layers panel* 에서 레이어를 오른쪽 클릭하고 다음 가운데 하나를 선택하면, 래스터 및 벡터 레이어 둘 다 서로 다른 포맷으로 저장 그리고/또는 서로 다른 좌표계로 재투영할 수 있습니다:

- *Export* ▸ *Save As*…: 래스터 레이어용
- *Export* ▸ *Save Features As*…또는 *Export* ▸ *Save Selected Features As*…: 벡터 레이어용
- 레이어 트리에서 *Browser Panel* 에 있는 PostGIS 항목으로 레이어를 드래그 & 드롭하십시오. 이때 *Browser Panel* 에 PostGIS 연결이 있어야만 한다는 사실을 기억하십시오.

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*: **layer**, **Map view** 또는 **user-defined** 범위 가운데 하나를 선택할 수 있습니다.
- *Add saved file to map*: 캔버스에 저장한 파일을 새 레이어로 추가합니다.

그런데 다음 몇몇 파라미터는 래스터 및 벡터 유형에 특화돼 있습니다:

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*

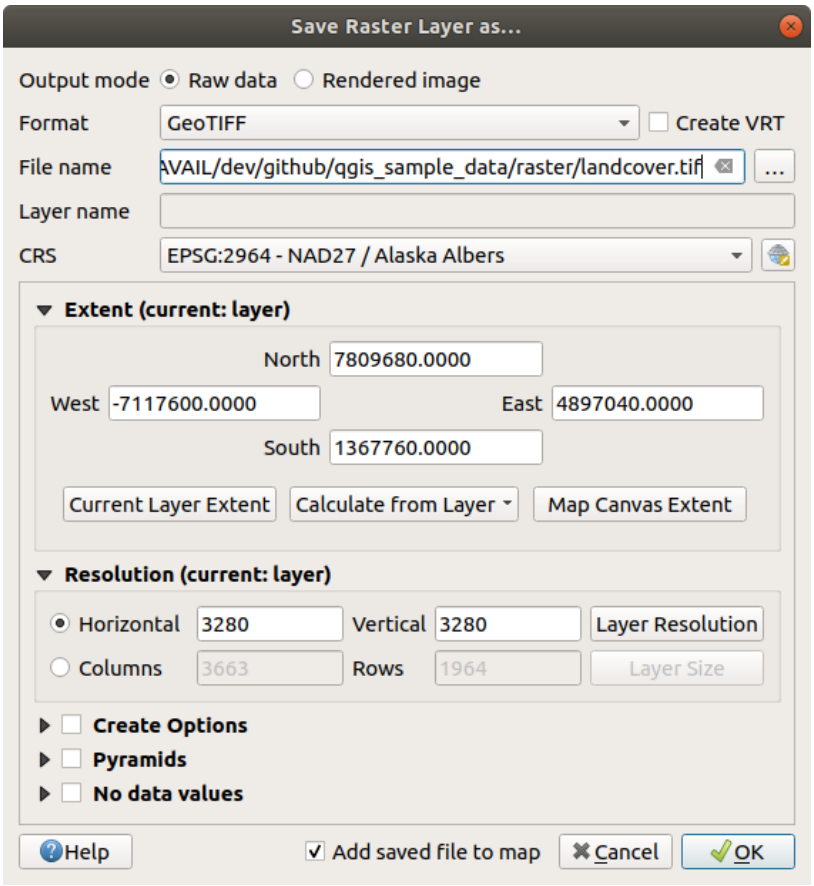


그림 13.19: 새 래스터 레이어로 저장하기

벡터 특화 파라미터

내보내기 포맷에 따라, 다음 가운데 일부 옵션을 사용할 수 있습니다:

- *Format*: GeoPackage, GML, ESRI Shapefile, 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*: 사용자가 `value map` 같은편집 위젯 으로 필드 습성을 설정한 경우, *Replace all selected raw fields values by displayed values* 를 체크하면 레이어 안에 표시된 값을 유지할 수 있습니다.
- *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 shapefile 로 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:

- 파일 전체를 덮어 쓰기
- 대상 레이어만 덮어 쓰기 (레이어명 설정 가능)

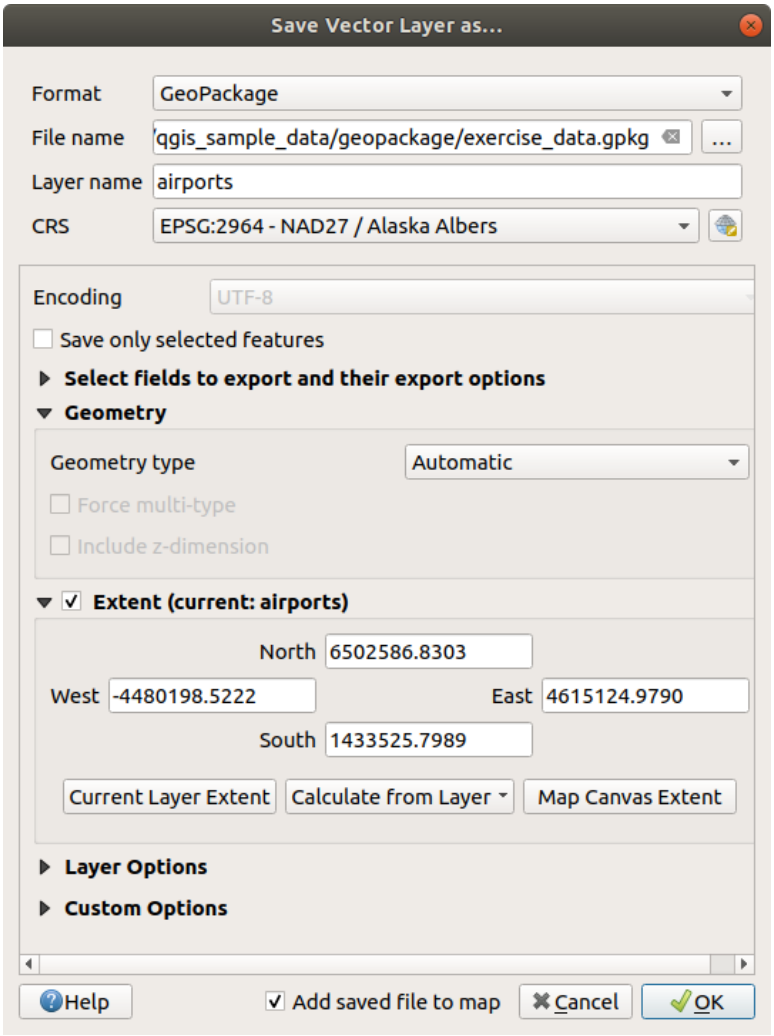


그림 13.20: 새 벡터 레이어로 저장하기

- 기존 대상 레이어에 피처 추가
- 새 필드가 있을 경우 필드와 함께 피처 추가

ESRI Shapefile, MapInfo .tab 같은 포맷의 경우도 피처를 추가할 수 있습니다.

13.2.3 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.

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.

   각 레이어에 대해, 모든 피처를 단일 DXF 레이어로 내보낼지 또는 피처들을 DXF 산출물에 생성된 레이어들로 분할하는 데 쓰이는 값을 보유한 필드를 의존할지 선택할 수 있습니다.

다음 체크박스들도 선택적으로 사용할 수 있습니다:

- ☑ *Use the layer title as name if set*: 이 옵션을 활성화하면 레이어명 대신 레이어 제목을 사용합니다.
- ☑ *Export features intersecting the current map extent*: 이 옵션을 활성화하면 현재 맵 캔버스 범위를 가로지르는 피처도 내보냅니다.
- ☐ *Force 2d output (eg. to support polyline width)*: 이 옵션을 활성화하면 산출물을 2 차원으로 강제합니다.
- ☑ *Export label as MTEXT elements*: 이 옵션을 활성화하면 라벨을 MTEXT(다중행 텍스트) 로 내보냅니다. 체크를 해제할 경우 일반 텍스트로 내보냅니다.

13.2.4 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 명칭을 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.

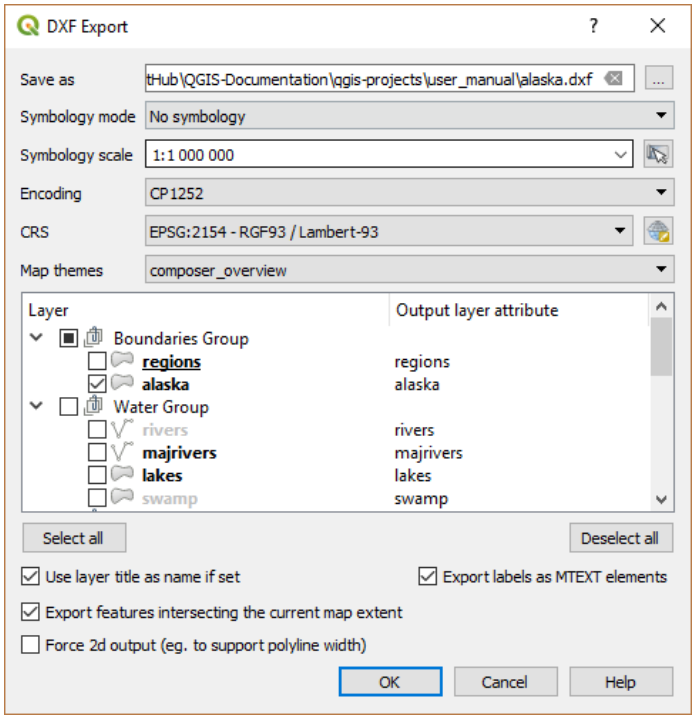


그림 13.21: 프로젝트를 DXF 로 내보내기 대화창

13.2.5 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 명칭 (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.

예를 들어 공항과 국가 정보를 결합시키려면, 다음과 같이 결합 (join) 및 복잡 쿼리를 생성할 수도 있습니다:

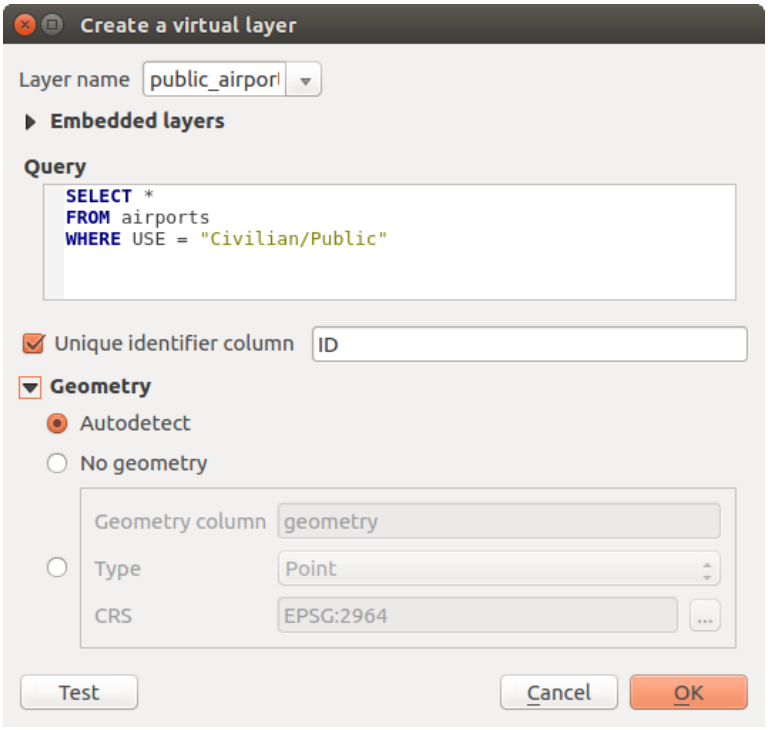


그림 13.22: 가상 레이어 생성 대화창

```
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` 명칭을 이용하십시오.

순수한 SQL 쿼리와는 반대로, 가상 레이어 쿼리의 모든 필드는 명칭을 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 명칭을 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 명칭의 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.

13.3 데이터 포맷 및 필드 탐구

13.3.1 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.

13.3.2 벡터 데이터

QGIS 에서 사용할 수 있는 기능 및 도구들 가운데 다수가, 벡터 데이터소스에 상관없이 동일하게 동작합니다. 하지만 포맷 사양의 차이점들 때문에 (GeoPackage, ESRI Shapefile, MapInfo 와 MicroStation 파일 포맷, AutoCAD DXF, PostGIS, SpatiaLite, DB2, Oracle Spatial, MSSQL 공간 데이터베이스 등등) QGIS 가 일부 속성을 서로 다르게 처리할 수도 있습니다. OGR 단순 피처 라이브러리 를 통해 이를 지원합니다. 이 절에서 이런 사양들을 어떻게 작업하는지 설명하겠습니다.

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 Shapefile 포맷

ESRI Shapefile 포맷은, 예를 들어 GeoPackage 및 SpatiaLite 와 비교하면 몇몇 제한이 존재하더라도, 지금도 QGIS 에서 가장 자주 이용되는 벡터 파일 포맷 가운데 하나입니다.

ESRI Shapefile 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 Shapefile 포맷 데이터셋은 `.prj` 확장자를 쓰는 파일도 포함할 수 있는데, 이 파일은 투영체 정보를 담고 있습니다. 투영체 파일이 있다면 매우 유용하지만, 필수적인 파일은 아닙니다. Shapefile 포맷 데이터셋은 추가 파일들을 포함할 수도 있습니다. 더 자세한 내용은 ESRI 기술 사양 문서 를 참조하세요.

GDAL 3.1 버전은 압축 ESRI Shapefile 포맷 (`shz` 및 `shp.zip`) 읽고 쓰기를 지원합니다.

ESRI Shapefile 포맷 데이터셋 성능 향상

ESRI Shapefile 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 Shapefile 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 Shapefile 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 Shapefile 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 명칭을 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 명칭은 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 저장하려면, CSVT 파일을 이용해야만 합니다. (필드 서식 제어에 *CSVT* 파일 사용하기 을 참조하세요.)

유효한 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 OGR 드라이버는 모든 필드가 (텍스트 등의) 문자열이라고 가정합니다. OGR 드라이버에 (그리고 QGIS 에) 서로 다른 열들이 어떤 데이터 유형인지 알려주는 다음 CSVT 파일을 생성할 수 있습니다:

유형	명칭	예시
범자연수 (Whole number)	Integer	4
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

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.

더 자세한 내용은 GDAL CSV 드라이버 에서 찾아볼 수 있습니다.

PostGIS 레이어

PostGIS 레이어는 PostgreSQL 데이터베이스에 저장돼 있습니다. PostGIS 의 장점은 공간 인덱스, 필터링 및 쿼리 작업 능력입니다. PostGIS 를 이용하면, 선택 및 식별 같은 벡터 기능들을 QGIS 에서 OGR 레이어로 작업하는 것보다 더 정확하게 작업할 수 있습니다.

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.

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.

QGIS layer_style 테이블 및 데이터베이스 백업

`pg_dump` 와 `pg_restore` 명령어를 이용해서 사용자 PostGIS 데이터베이스를 백업하려 하는데 기본 레이어 스타일을 QGIS 에서 저장한 대로 복구되지 않는다면, 복구 명령 전에 `DOCUMENT` 에 XML 옵션을 설정해줘야 합니다.

```
SET XML OPTION DOCUMENT;
```

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, 부동소수점형, 불린 (boolean), 2 진 객체 (binary object), 문자형 (varchar), 도형형, 타임스탬프 (timestamp), 배열 (array), HStore, JSON 등의 데이터 유형을 지원합니다.

PostgreSQL 로 데이터 가져오기

DB 관리자 플러그인과 shp2pgsql 및 ogr2ogr 명령 줄 도구를 포함하는 도구 몇 가지를 이용해서 PostgreSQL/PostGIS 로 데이터를 가져올 수 있습니다.

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 Shapefile 포맷 데이터셋을 불러올 수 있는 **shp2pgsql** 이라는 기능을 포함하고 있습니다. 예를 들어, `gis_data` 라는 PostgreSQL 데이터베이스로 `lakes.shp` 이라는 Shapefile 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: PostGIS 에서 데이터셋 내보내기

PostGIS 데이터베이스를 Shapefile 포맷 데이터셋으로 내보내는 도구인 **pgsql2shp** 도 있습니다. 이 도구도 PostGIS 배포판에 내장돼 있습니다.

---

ogr2ogr

shp2pgsql 및 DB 관리자 이외에, PostGIS 로 지리 데이터를 불러올 수 있는 또다른 도구인 **ogr2ogr** 도 있습니다. 이 도구는 사용자가 GDAL 을 설치할 때 함께 설치됩니다.

PostGIS 로 Shapefile 포맷 데이터셋을 가져오려면, 다음 명령어를 입력하십시오:

```
ogr2ogr -f "PostgreSQL" PG:"dbname=postgis host=myhost.de user=postgres
password=topsecret" alaska.shp
```

이 명령어는 호스트 서버 *myhost.de* 에 있는 PostGIS 데이터베이스 *postgis* 로 Shapefile 포맷 데이터셋 `alaska.shp` 를 사용자 *postgre* 와 비밀번호 *topsecret* 을 이용해서 불러올 것입니다.

PostGIS 를 지원하려면 OGR 을 PostgreSQL 과 함께 빌드해야만 한다는 사실을 기억하십시오. 함께 빌드했는지 확인하려면 (에서) 다음과 같이 입력해보세요:

```
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. Shapefile 포맷 데이터셋을 불러온 후에 (다음 성능 향상시키기 에서 설명하는 대로) 추가 단계로 일반적인 SQL 명령어 **CREATE INDEX** 를 이용해서 직접 생성해야 합니다.

성능 향상시키기

특히 네트워크를 통할 경우, PostgreSQL 데이터베이스에서 피처를 받아오는 작업에 시간이 많이 걸릴 수 있습니다. 데이터베이스에 있는 각 레이어가 PostGIS 공간 인덱스를 보유하도록 보장해서 PostgreSQL 레이어의 렌더링 속도를 향상시킬 수 있습니다. PostGIS 는 공간 검색 속도를 높이기 위해 GiST(Generalized Search Tree) 인덱스를 생성할 수 있습니다. (GiST 인덱스에 대한 정보는 PostGIS 문서 를 참조하세요.)

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
```

(다음 페이지에 계속)

(이전 페이지에서 계속)

```
       \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 (the_geom GIST_GEOMETRY_OPS);
CREATE INDEX
gis_data=# VACUUM ANALYZE alaska_lakes;
VACUUM
gis_data=# \q
gsherman@madison:~/current$
```

경도 180° 선을 가로지르는 벡터 레이어

많은 GIS 패키지들이 경도 180° 선을 가로지르는 지리 참조 시스템 (위도/경도) 으로 벡터 맵을 감싸고 있지 않습니다. (http://postgis.refractions.net/documentation/manual-2.0/ST_Shift_Longitude.html) 그 결과, QGIS 에서 이런 맵을 열었을 때, 서로 가까이 있어야 하지만 멀리 떨어져 있는 두 위치를 보게 됩니다. 그림 13.23 에서, 맵 캔버스 왼쪽 구석에 있는 작은 점들 (채팅 제도) 은 그리드 내부, 뉴질랜드 본도의 오른쪽에 있어야 합니다.

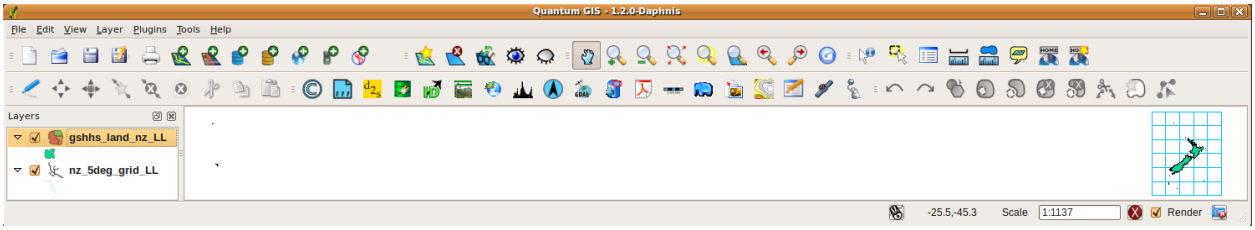


그림 13.23: 경도 180° 선을 가로지르는 위도/경도 맵

PostGIS 와 **ST_Shift_Longitude** 함수를 이용해서 경도를 변환하면 이 문제를 해결할 수 있습니다. 이 함수는 도형이 보유한 모든 객체의 모든 요소의 모든 포인트/꼭짓점을 읽은 다음 경도 좌표가 0° 미만인 경우 360° 를 더합니다. 그 결과, 경도 180° 선을 중심으로 하는 맵에 0° 에서 360° 범위로 변환된 데이터를 그리게 됩니다.

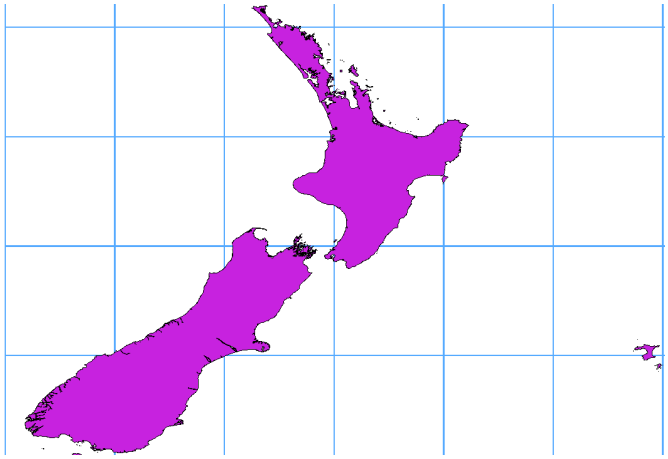


그림 13.24: 경도 180° 선을 가로지르는 데이터에 **ST_Shift_Longitude** 함수를 적용

활용

- DB 관리자 플러그인 등을 이용해서 PostGIS 로 데이터를 불러옵니다. (*PostgreSQL* 로 데이터 가져오기 참조)
- PostGIS 명령 줄 인터페이스를 통해 다음 명령어를 입력하고 실행합니다. (이 예시에서, "TABLE"은 사용자 PostGIS 테이블의 실제 명칭입니다): `gis_data=# update TABLE set the_geom=ST_Shift_Longitude(the_geom);`
- 모든 작업이 제대로 됐다면, 업데이트한 피처의 개수를 확인하는 메시지를 볼 수 있을 겁니다. 그리고 나서야 맵을 불러와서 바뀐 모습 (그림 13.24) 을 볼 수 있습니다.

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.

또는 포맷을 `SQLite` 으로 선택한 다음 *Custom Options* ▸ *Data source* 란에 `SPATIALITE=YES` 를 추가해도 됩니다. 이렇게 하면 GDAL 이 SpatiaLite 데이터베이스를 추가할 겁니다. https://gdal.org/drivers/vector/sqlite.html 도 참조하세요.

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*: 기본값 (NO) 은 GeoJSON 2008 입니다. YES 로 설정하면 업데이트된 RFC 7946 표준을 사용할 것입니다. 주요 차이점은 https://gdal.org/drivers/vector/geojson.html#rfc-7946-write-support 를 참조하세요. 간단히 요약하면: EPSG:4326 만 사용할 수 있고, 다른 좌표계는 변환될 것이며, 방향의 오른손 법칙을 따르도록 폴리곤을 작성할 것이고, "경계 상자 (bbox)"배열의 값은 [minx, miny, maxx, maxy] 가 아닌 [west, south, east, north] 입니다. 피처 집합 (FeatureCollection), 피처 및 도형 객체에서 일부 확장자 명칭들을 사용할 수 없으며, 기본 좌표 정밀도는 십진수 7 자리입니다.
- *WRITE_BBOX*: YES 로 설정하면 피처 및 피처 집합 수준에서 도형의 경계 상자를 포함합니다.

GeoJSON 이외에, "GeoJSON - 새줄 문자 구분 (Newline Delimited)"으로 내보낼 수 있는 옵션도 있습니다. (https://gdal.org/drivers/vector/geojsonseq.html 참조) 피처를 가진 피처 집합 (FeatureCollection with Features) 대신 새줄 문자 (newline) 로 연속적으로 구분된 (아마도 피처만의) 한 유형을 줄줄이 이어지게 할 수 있습니다.

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.

DB2 Spatial 레이어

리눅스, 유닉스 및 윈도우 용 IBM DB2(DB2 LUW), z/OS 용 IBM DB2(메인프레임) 그리고 IBM DashDB 제품은 관계형 테이블 열에 공간 데이터를 저장하고 분석할 수 있습니다. QGIS 용 DB2 제공자는 이 데이터베이스에 있는 공간 데이터를 가시화하고 분석 및 수정할 수 있는 완전한 기능을 제공합니다.

DB2 z/OS KnowledgeCenter, DB2 LUW KnowledgeCenter 그리고 DB2 DashDB KnowledgeCenter 에서 이런 능력에 대한 사용자 문서를 볼 수 있습니다.

DB2 공간 능력을 작업하는 데 관한 더 자세한 정보를 알고 싶다면, IBM DeveloperWorks 에 있는 DB2 Spatial Tutorial 을 확인해보세요.

DB2 제공자는 현재 윈도우 ODBC 드라이버를 통해 윈도우 환경만 지원하고 있습니다.

QGIS 를 실행하는 클라이언트는 다음 중 하나를 설치해야 합니다:

- DB2 LUW
- IBM 데이터 서버 드라이버 패키지
- IBM 데이터 서버 클라이언트

QGIS 에서 DB2 데이터를 열려면, 탐색기 패널 또는데이터베이스 레이어 불러오기 를 참조하세요.

동일한 머신 상에 있는 DB2 LUW 데이터베이스에 접근하거나 또는 DB2 LUW 를 클라이언트로 이용하는 경우, DB2 실행 파일 및 지원 파일들을 윈도우 경로에 포함시켜야 합니다. 다음과 같은 배치 파일을 **db2.bat** 이라는 명칭으로 생성한 다음 **%OSGEO4W_ROOT%/etc/ini** 디렉터리로 복사/이동해 넣으면 됩니다.

```
@echo off
REM Point the following to where DB2 is installed
SET db2path=C:\Program Files (x86)\sqllib
REM This should usually be ok - modify if necessary
SET gskpath=C:\Program Files (x86)\ibm\gsk8
SET Path=%db2path%\BIN;%db2path%\FUNCTION;%gskpath%\lib64;%gskpath%\lib;%path%
```

CHAPTER 14

벡터 데이터 작업

14.1 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	심볼[1]
라벨[1]	마스크[1]	*3D* 뷰[1]
도표	필드	속성 양식
결합	보조 저장소	액션
표시	렌더링	시계열
변수	메타데이터	의존성
범례	*QGIS* 서버	디지타이즈 작업
외부 플러그인[2] 탭		

[1] 레이어 스타일 작업 패널 에서도 사용할 수 있습니다.

[2] 사용자가 설치한외부 플러그인 이 이 대화창에 탭을 추가할 수도 있습니다. 이 문서에서는 이에 대해 설명하지 않습니다. 외부 플러그인 문서를 참조하십시오.

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 (프로젝트 내포 작업 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.

14.1.1 정보 속성

Information 탭은 읽기 전용으로, 현재 레이어에 대한 요약 정보 및 메타데이터를 한 눈에 볼 수 있는 흥미로운 장소입니다. 다음과 같은 정보를 제공합니다:

- 레이어 제공자 기반 정보 (저장소 유형, 경로, 도형 유형, 데이터소스 인코딩, 범위 등등)
- 채워진 메타데이터 에서 고른 정보 (접근 정보, 링크, 연락처, 이력 등등)
- 레이어 도형 관련 정보 (공간 범위, 좌표계 등등) 또는 도형 속성 (필드 개수, 각 도형의 특성 등등)

14.1.2 소스 속성

Source 탭에서 벡터 레이어에 대한 일반 설정을 정의할 수 있습니다.

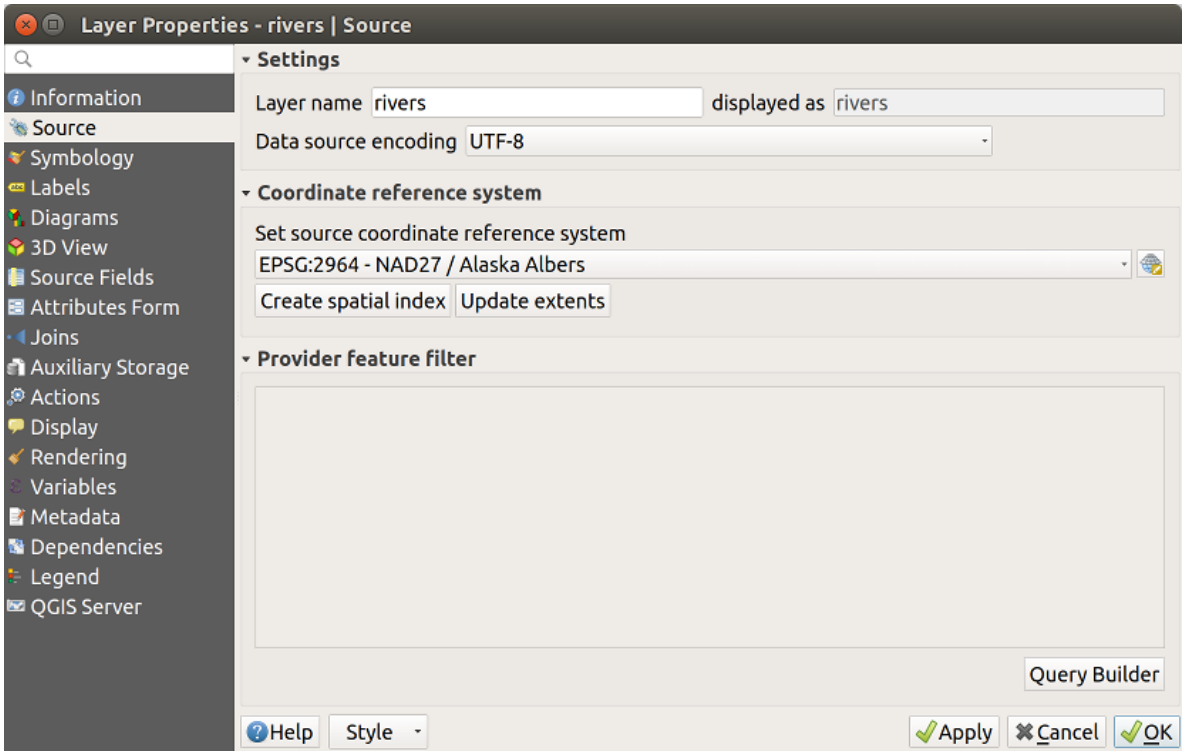


그림 14.1: 벡터 레이어 속성 대화창의 소스 탭

Layers Panel 에 표시할 *Layer name* 설정 이외에도, 다음과 같은 옵션을 설정할 수 있습니다:

좌표계

- 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 또다른 좌표계로 재투영하고 싶은 경우, 공간 처리 레이어 재투영 알고리즘을 이용하거나다른 레이어로 레이어를 저장 하는 편이 좋습니다.
- *Create spatial index*: 공간 인덱스를 생성합니다. (OGR 지원 유형 전용입니다.)
- *Update extents*: 레이어의 범위 정보를 업데이트합니다.

쿼리 작성기

레이어 속성 대화창의 *Source* 탭의 *Provider Feature Filter* 그룹 하단에 있는 쿼리 작성기 버튼을 누르면 *Query Builder* 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 14.2 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 (OGR, PostgreSQL, MSSQL 등등) 수준에서 필터를 생성하기 때문에, 문법은 데이터 제공자를 따릅니다. (예를 들어 ESRI Shapefile 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.

필드 목록은 레이어의 모든 필드를 담고 있습니다. SQL WHERE 절 란에 속성 열을 추가하려면, 필드 목록에 있는 명칭을 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 명칭을 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.

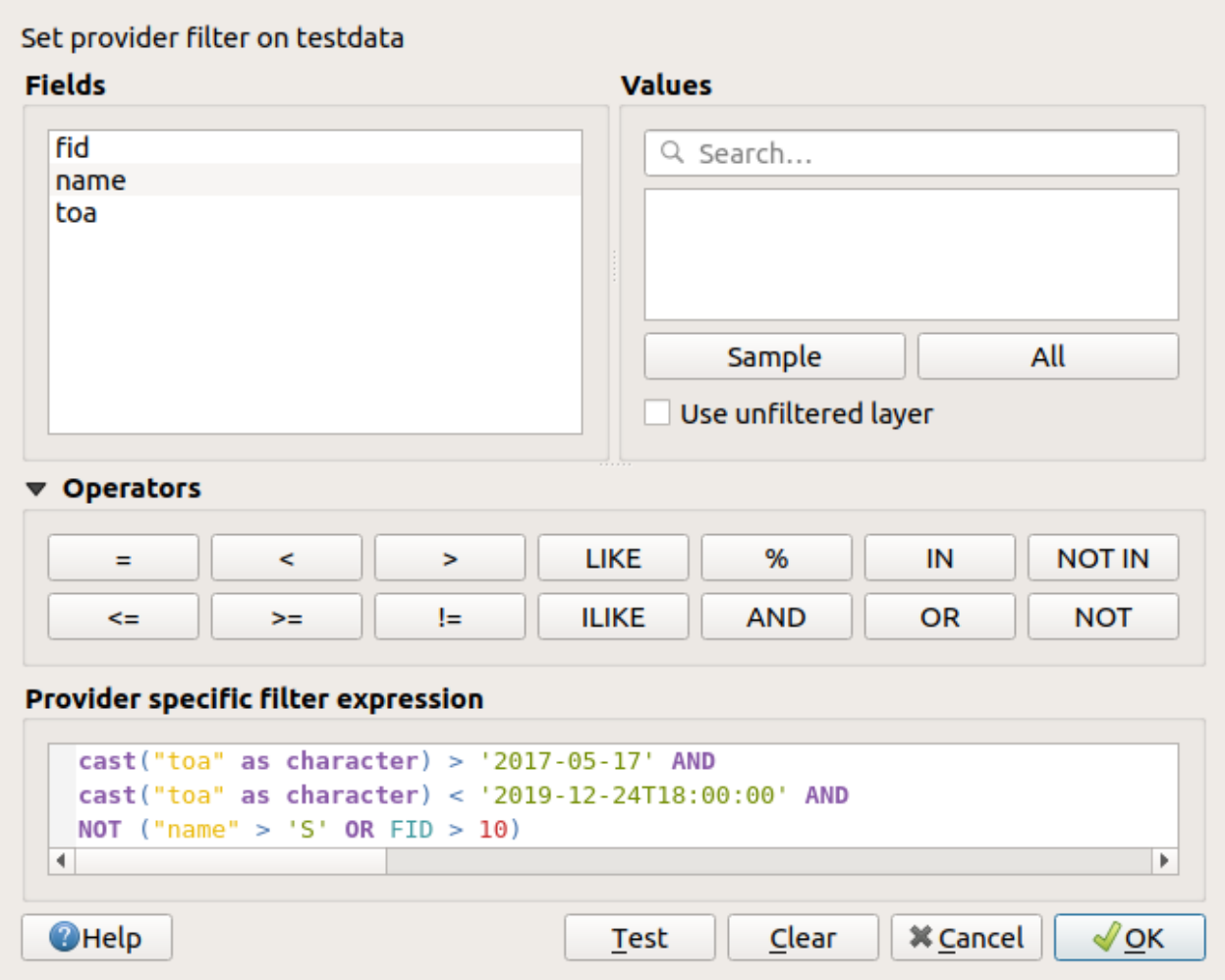

Set provider filter on testdata
Fields
fid
name
toa
Values
Search...
Sample
All
Use unfiltered layer
Operators
=
<
>
LIKE
%
IN
NOT IN
<=
>=
!=
ILIKE
AND
OR
NOT
Provider specific filter expression
cast("toa" as character) > '2017-05-17' AND
cast("toa" as character) < '2019-12-24T18:00:00' AND
NOT ("name" > 'S' OR FID > 10)
Help
Test
Clear
Cancel
OK

그림 14.2: 쿼리 작성기

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.

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.

14.1.3 심볼 속성

 심볼 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) 및 열지도 렌더링 작업자를 이용할 수 있는 반면 폴리곤 레이어의 경우 그 외에도 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.

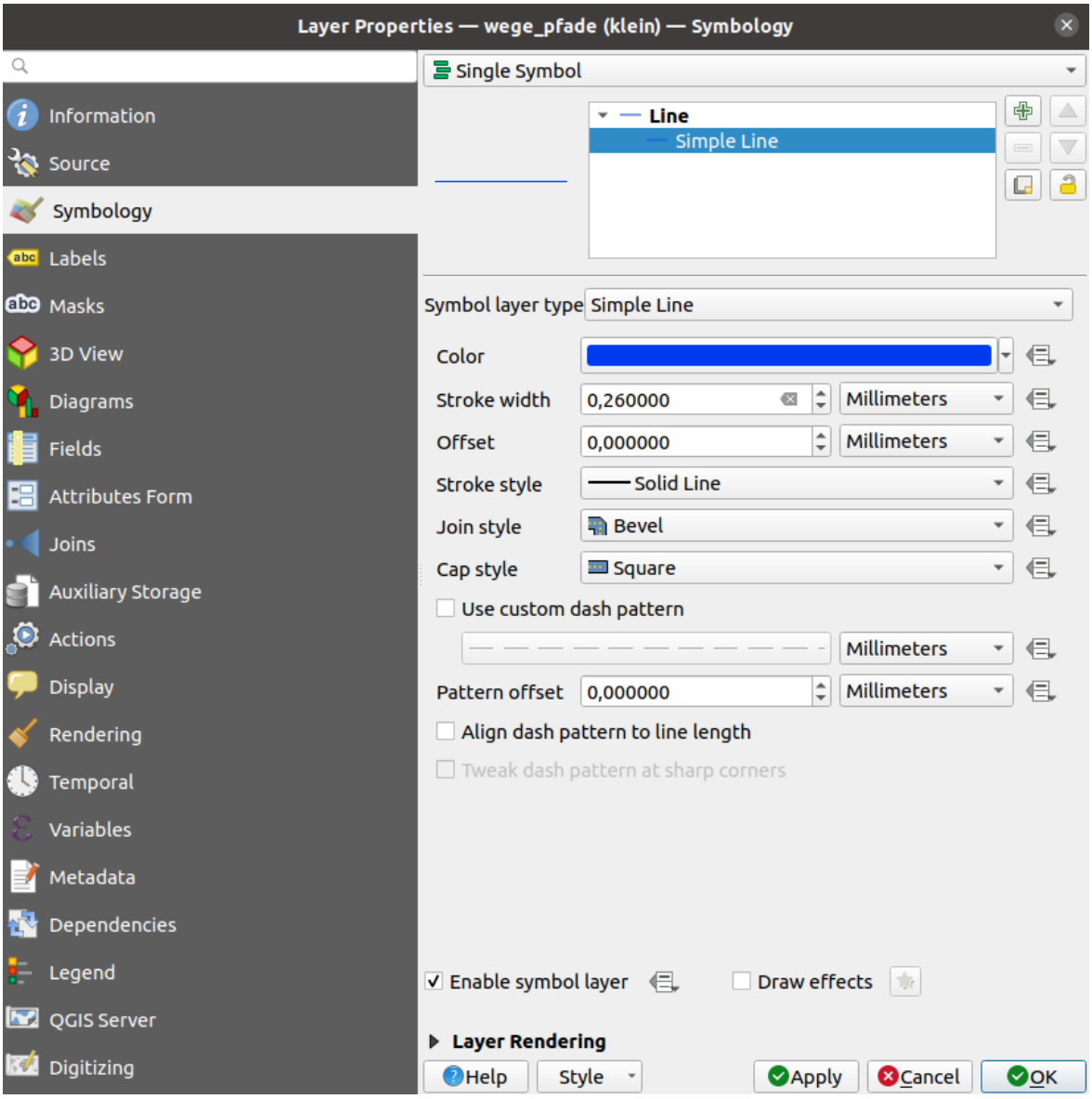


그림 14.3: 단일 심볼 라인 속성

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.

범주 렌더링 작업자

Categorized 렌더링 작업자는 필드 또는 표현식의 개별 값을 속성에 반영하는 사용자 지정 심볼을 이용해서 레이어의 피처를 렌더링합니다.

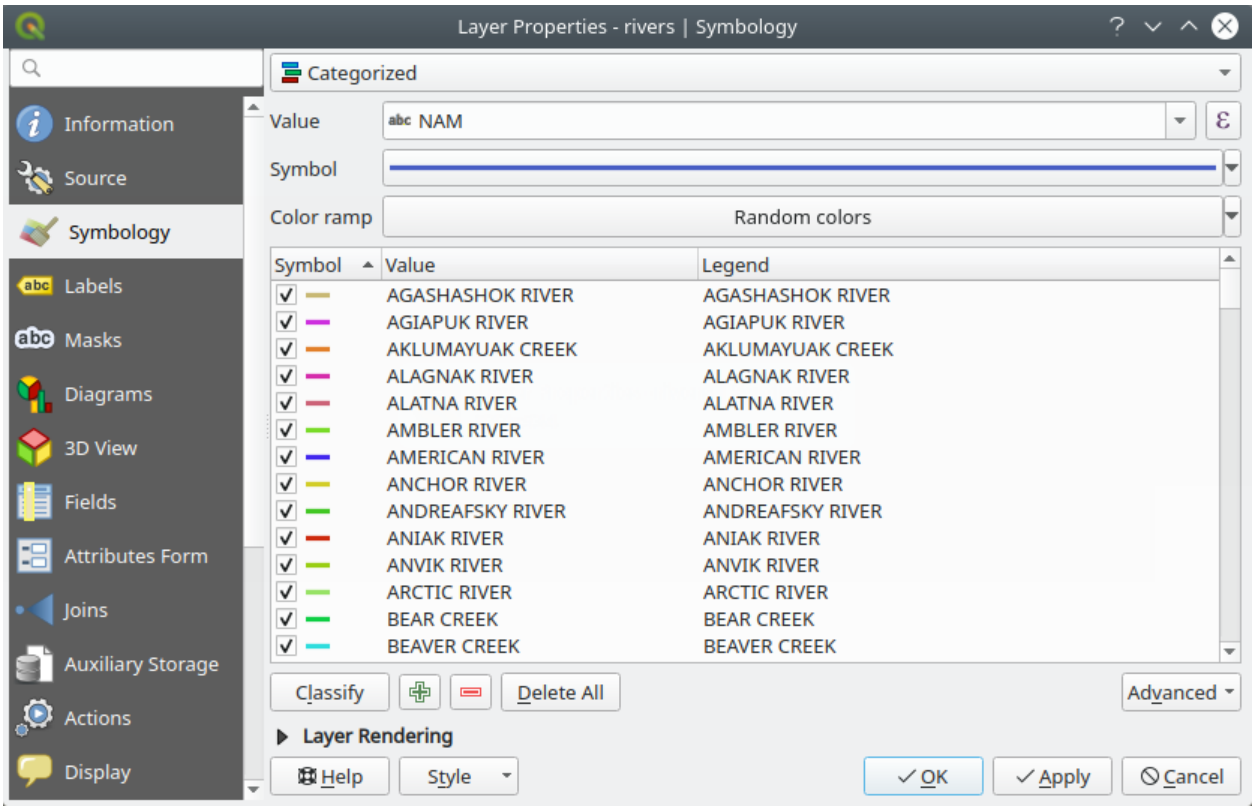


그림 14.4: 범주 심볼 작업 옵션들

레이어에 대해 범주 심볼을 사용하려면:

1. 범주의 *Value* 를 선택하십시오: 입력란에 입력하거나 관련 ε 버튼으로 작성할 수 있는표현식, 또는 기존 필드일 수 있습니다. 범주 작업에 표현식을 사용하면 심볼의 목적을 위해 즉석 필드를 생성할 필요가 없습니다. (예를 들어 하나 이상의 속성으로부터 사용자의 범주 기준을 파생시킬 수 있습니다.)

   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:

```
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. 각 심볼에 적용되는 색상을 선택할 색상 범위 같은색상표 를 지정하십시오.

   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. 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:

- Add 새 범주를 추가할 수도 있고, Remove 선택한 범주를 제거할 수도 있으며, 또는 *Delete All* 모든 범주를 삭제할 수도 있습니다.
- 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.

Advanced 메뉴는 범주화 작업의 속도를 향상시키거나 심볼 렌더링을 미세 조정할 수 있는 옵션을 제공합니다:

- *Match to saved symbols*: 각 범주에 범주의 범주화 값을 나타내는 명칭을 가진 심볼을심볼 라이브러리 를 사용해서 할당합니다.
- *Match to symbols from file*…: 심볼을 보유한 파일을 지정해서, 각 범주에 범주의 범주화 값을 나타내는 명칭을 가진 심볼을 할당합니다.
- 심볼 수준…: 심볼 렌더링 순서를 정의합니다.

팁: *Layers* **패널에서 직접 범주 편집**

레이어 심볼이범주, 등급 또는규칙 기반 심볼 모드를 기반으로 하는 경우, *Layers* 패널에서 각 범주를 편집할 수 있습니다. 레이어의 하위 항목을 오른쪽 클릭하면:

-  *Toggle items*: 해당 항목의 가시성을 켜고 끕니다.
-  *Show all items*: 모든 항목을 표시합니다.
-  *Hide all items*: 모든 항목을 숨깁니다.
- 색상 선택기 휠을 통해 심볼 색상을 수정합니다.
- *Edit symbol*…: 심볼 선택기 대화창에서 심볼을 편집합니다.
- *Copy symbol*: 심볼을 복사합니다.
- *Paste symbol*: 심볼을 붙여넣습니다.

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:

- 값 (필드 목록 상자 또는  Set value expression 기능을 이용합니다.)
- 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.)
- 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 14.5 는 QGIS 예시 데이터셋의 `major_rivers` 레이어에 대한 등급 렌더링 작업자 대화창의 예시입니다.

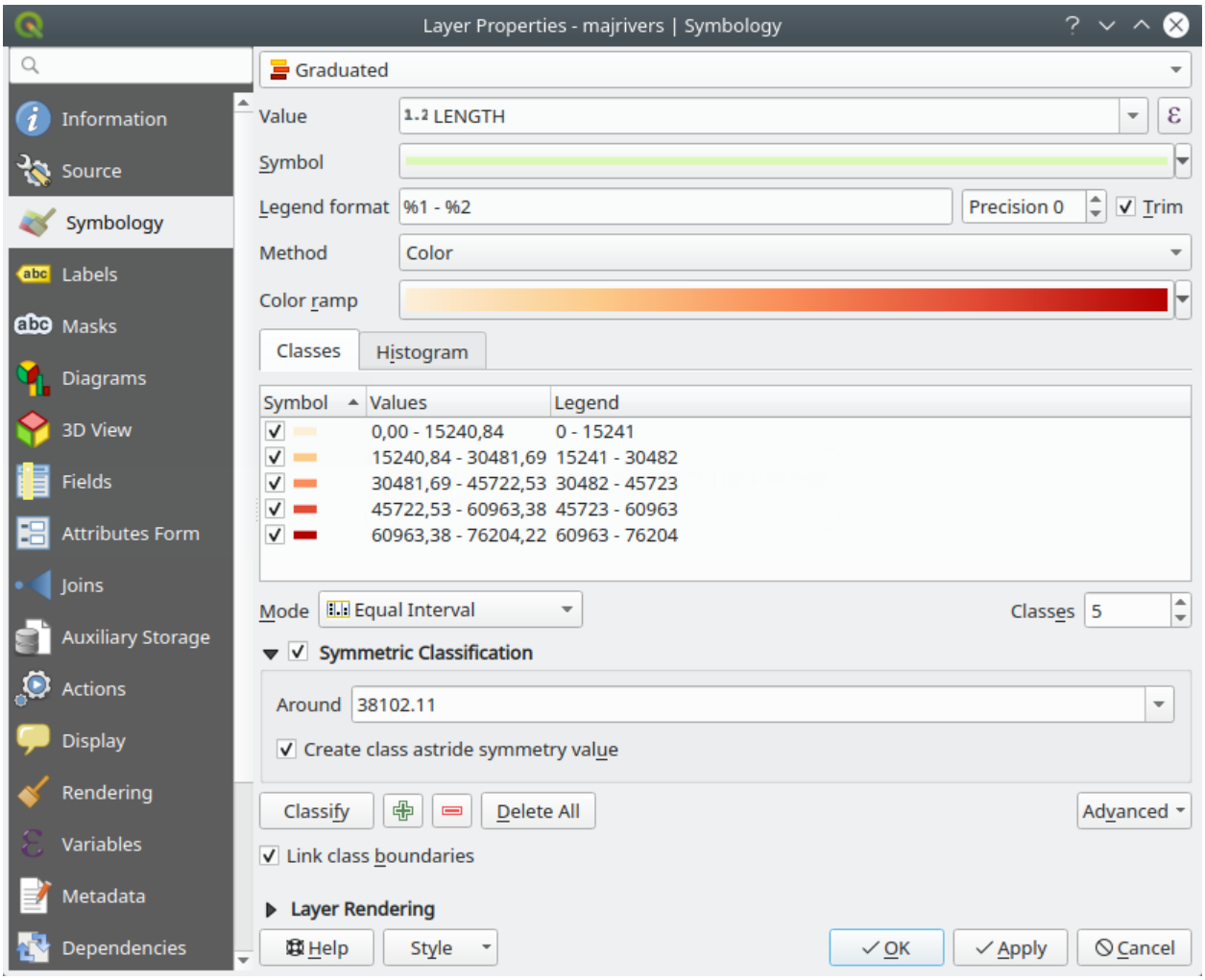


그림 14.5: 등급 심볼 작업 옵션들

팁: 표현식을 이용한 테마 지도

표현식의 결과를 이용해서 범주 및 등급 테마 맵을 생성할 수 있습니다. 벡터 레이어의 속성 대화창에서, Set column expression 기능을 통해 속성 선택기를 확장합니다. 즉 범주화 속성을 여러 필드의 합성으로 또는 일종의 틀로 만들고자 할 경우, 사용자 속성 테이블에 새 열을 만들어 범주화 속성을 입력할 필요가 없다는 뜻입니다.

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.)

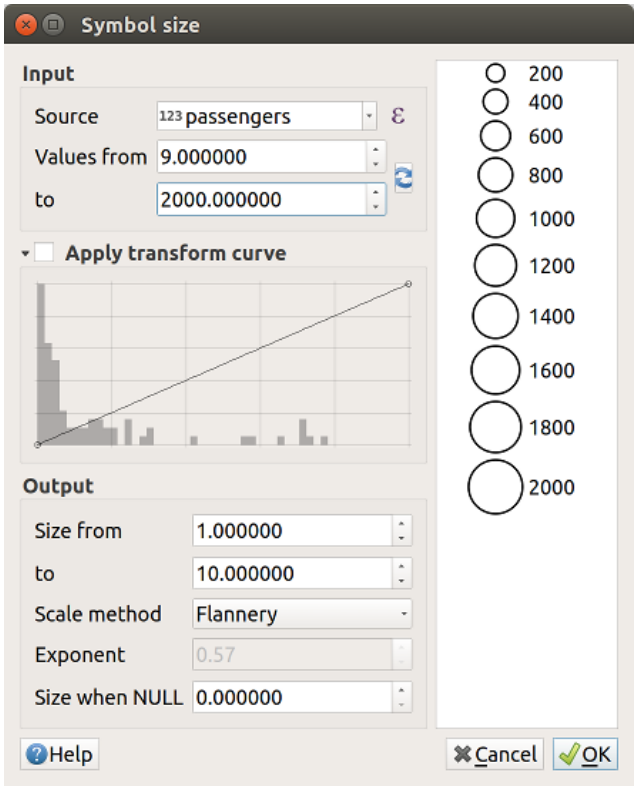


그림 14.6: 공항의 표고를 기반으로 공항 크기 조정하기

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.

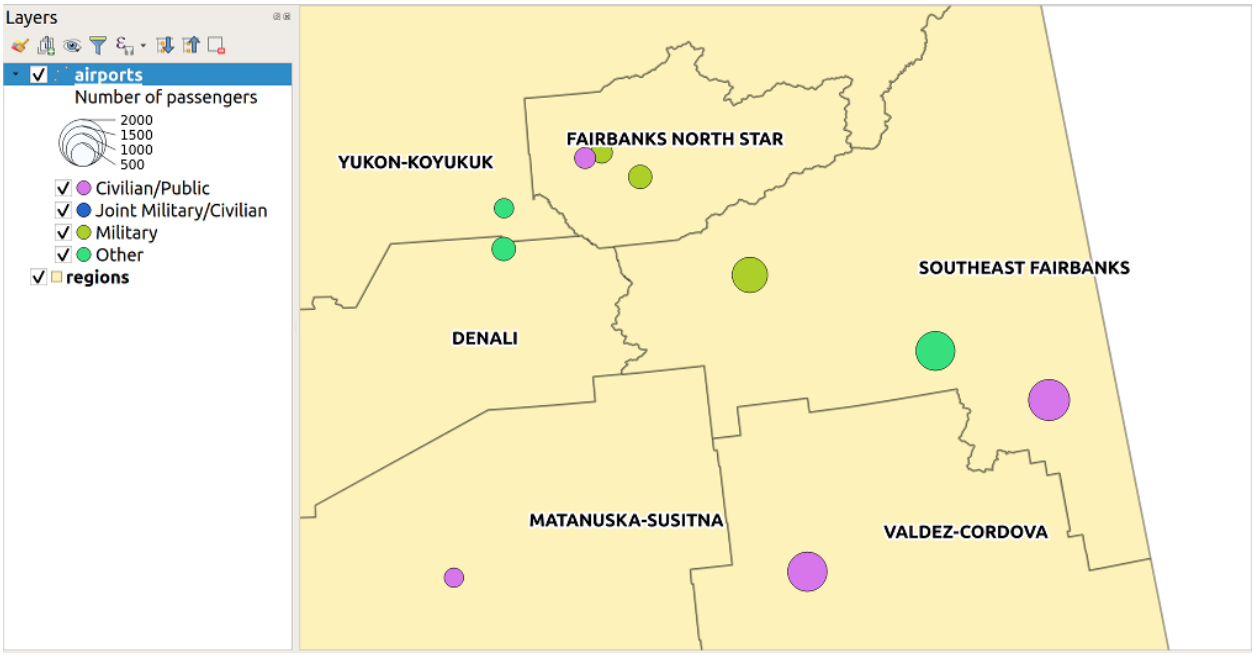


그림 14.7: 크기 척도를 조정한 범례를 가진 다변량 분석의 예시

규칙 기반 렌더링 작업자

Rule-based 렌더링 작업자는 선택한 피처의 속성이 어떤 범주에 할당되는지를 반영하는 규칙 기반 심볼을 이용해서 레이어에 있는 모든 피처를 렌더링합니다. 이 규칙은 SQL 선언문을 기반으로 하며 내포될 수 있습니다. 대화창에서 필터 또는 척도로 규칙을 그룹화할 수 있으며, 사용자가 심볼 수준을 활성화할지 또는 가장 먼저 일치한 (first-matched) 규칙만 이용할지 결정할 수 있습니다.

규칙을 생성하려면:

1. 기존 행을 더블 클릭하거나 (렌더링 모드가 활성화된 경우, QGIS 는 기본적으로 심볼을 규칙 없이 추가합니다) Edit rule 또는 Add rule 버튼을 클릭해서 활성화하십시오.
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 ε 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.

규칙을 선택하면, *Refine selected rules* 드롭다운 메뉴를 사용해서 하위 범주에 있는 규칙 피처들을 정리할 수도 있습니다. 다음을 기반으로 규칙 개선 작업을 자동화시킬 수 있습니다:

- **축척 (scale)**
- **범주 (category)**: 범주 렌더링 작업자 적용
- **범위 (range)**: 등급 렌더링 작업자 적용

개선된 범주들은 트리 위계에서 규칙의 하위 항목처럼 나타나며, 앞에서와 마찬가지로 각 범주의 심볼을 설정할 수 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** (범주 렌더링 작업자) 또는 **범위** (등급 렌더링 작업자) 를 통해 개선할 수 있는 하위 메뉴를 엽니다.

이렇게 생성한 규칙도 맵 범례의 트리 위계에 나타납니다. 맵 범례에 있는 규칙을 더블 클릭하면 레이어 속성 대화창의 심볼 탭에 트리에 있는 심볼을 위한 배경인 규칙을 표시합니다.

그림 14.8 은 QGIS 예시 데이터셋의 `rivers` 레이어에 대한 규칙 기반 렌더링 작업자 대화창의 예시입니다.

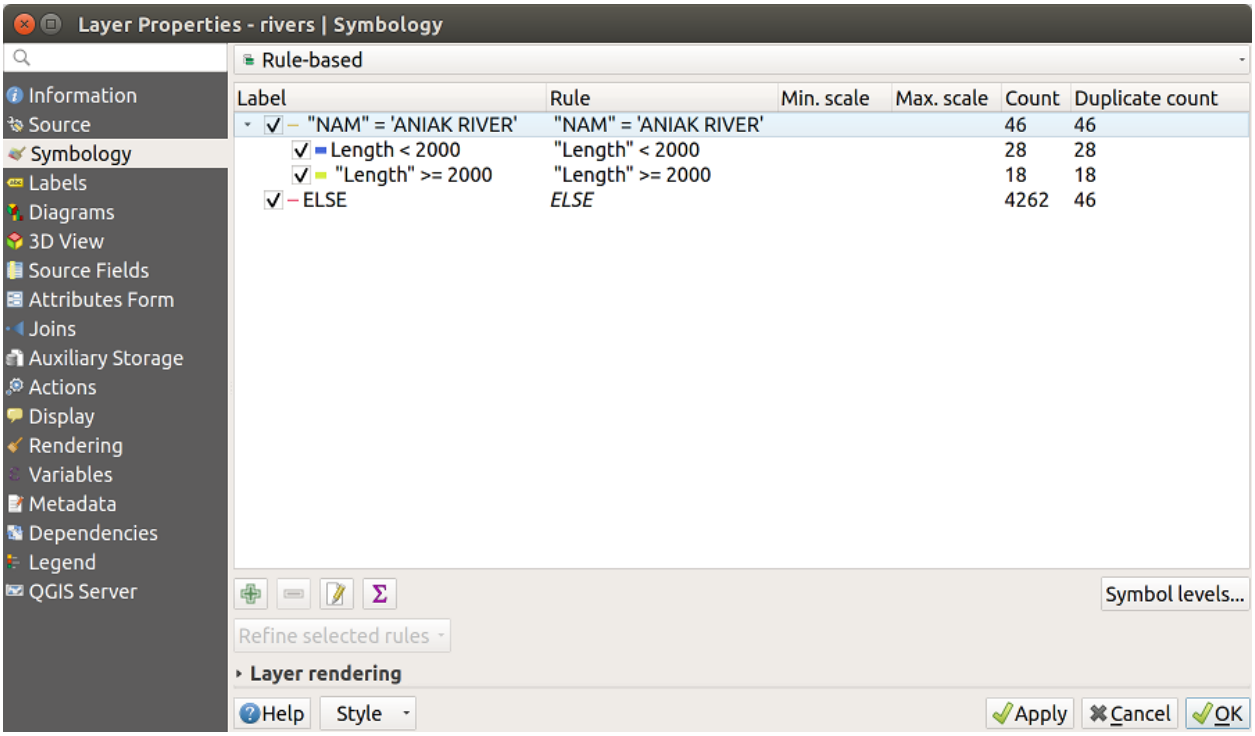


그림 14.8: 규칙 기반 심볼 작업 옵션들

포인트 변위 렌더링 작업자

Point Displacement 렌더링 작업자는 포인트 레이어의 모든 피처를, 모든 피처가 동일한 위치에 있다 하더라도, 시각화합니다. 이를 위해 서로의 *Distance* 허용 오차 안에 들어오는 포인트들을 선택해서 서로 다른 *Placement methods* 를 따라 포인트 무게중심 (barycenter) 주위로 배치합니다:

- **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.

Center symbol 위젯은 심볼 및 중점 (middle point) 의 색상을 사용자 지정할 수 있습니다. 분산된 포인트 심볼들에 대해, 무 (☐) 심볼, 단일 심볼, 범주, 등급 또는 규칙 기반 렌더링 작업자 가운데 하나를 *Renderer* 드롭다운 목록에서 선택하고 *Renderer Settings*…버튼으로 사용자 지정해서 적용할 수 있습니다.

Displacement lines 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 사이에 간격을 추가할 수 있습니다.)

Labels 그룹 옵션을 사용해서 포인트 라벨 작업을 수행하십시오: 라벨이 피처의 실제 위치가 아니라 심볼의 변위된 위치에 배치됩니다. *Label attribute*, *Label font* 및 *Label color* 는 물론, 라벨을 표시할 최소 맵 축척을 *Minimum map scale* 을 통해 설정할 수 있습니다.

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.

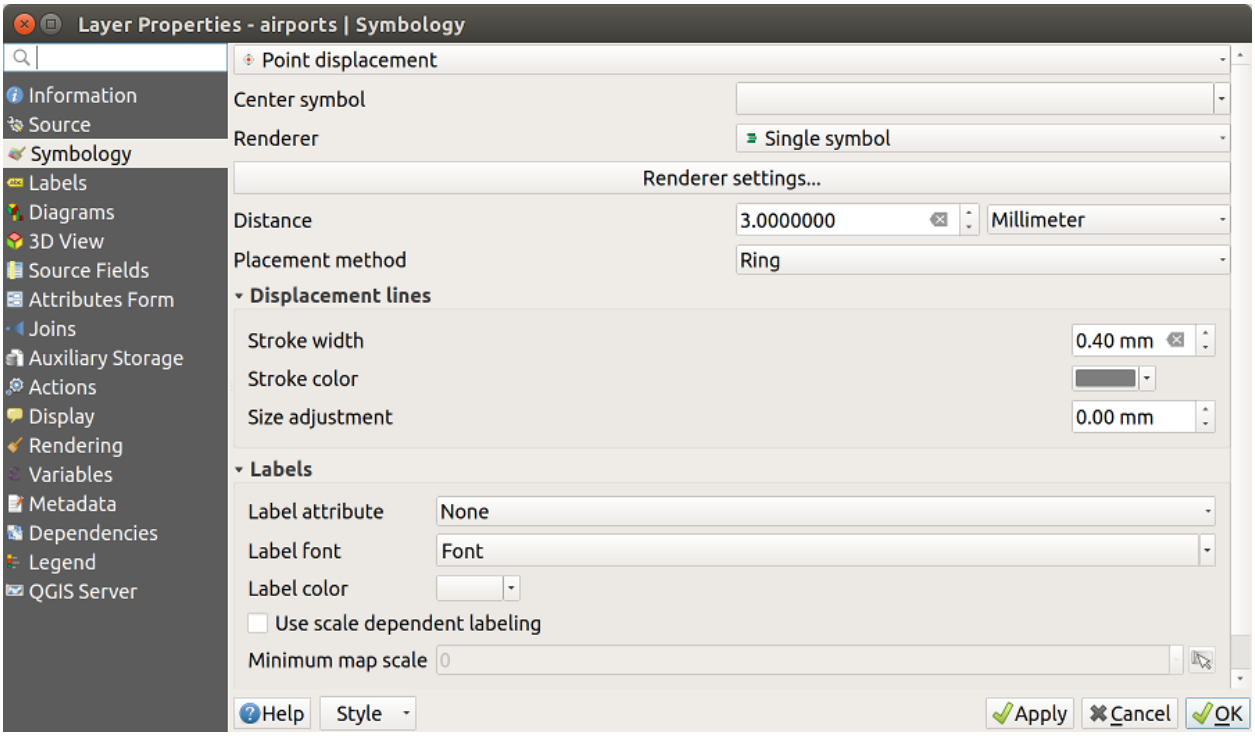


그림 14.9: 포인트 변위 대화창

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. 서로가 지정한 *Distance* 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:

- *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.
- *Renderer* 드롭다운 목록을 사용해서 레이어에 다른 어떤 피처 렌더링 유형 (단일, 범주, 규칙 기반 등등) 도 적용할 수 있습니다. 그 다음 *Renderer Settings*…버튼을 클릭하면 피처의 심볼을 평소와 다름없이 환경 설정할 수 있습니다. 이 렌더링 작업자는 군집되지 않은 피처 상에서만 가시화된다는 사실을 기억하십시오. 또 군집 안에 있는 모든 포인트 피처가 동일한 심볼 색상을 가지고 있을 경우, 해당 색상이 군집의 `@cluster_color` 변수를 설정합니다.

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.

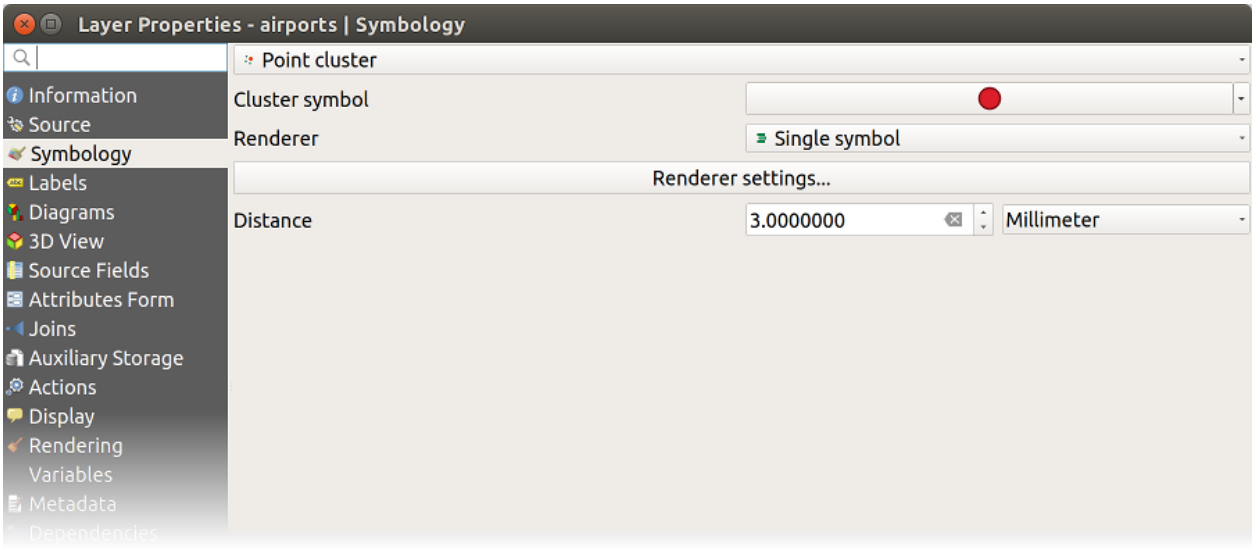


그림 14.10: 포인트 군집 대화창

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.

열지도 렌더링 작업자

Heatmap 렌더링 작업자는 [멀티] 포인트 레이어에 대해 실시간 동적 열지도를 생성할 수 있습니다. 사용자가 열지도 반경을 밀리미터, 포인트, 픽셀, 맵 단위 또는 인치로 지정할 수 있고, 열지도 스타일을 위한 색상표를 선택하고 편집할 수 있으며, 렌더링 속도와 품질 사이의 균형을 선택하기 위한 슬라이드 바를 이용할 수도 있습니다. 필드 또는 표현식을 통해 최대값 제한을 정의하고 포인트에 가중치를 줄 수도 있습니다. 피처를 추가 또는 제거하면 열지도 렌더링 작업자가 열지도 스타일을 자동적으로 업데이트합니다.

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.

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*

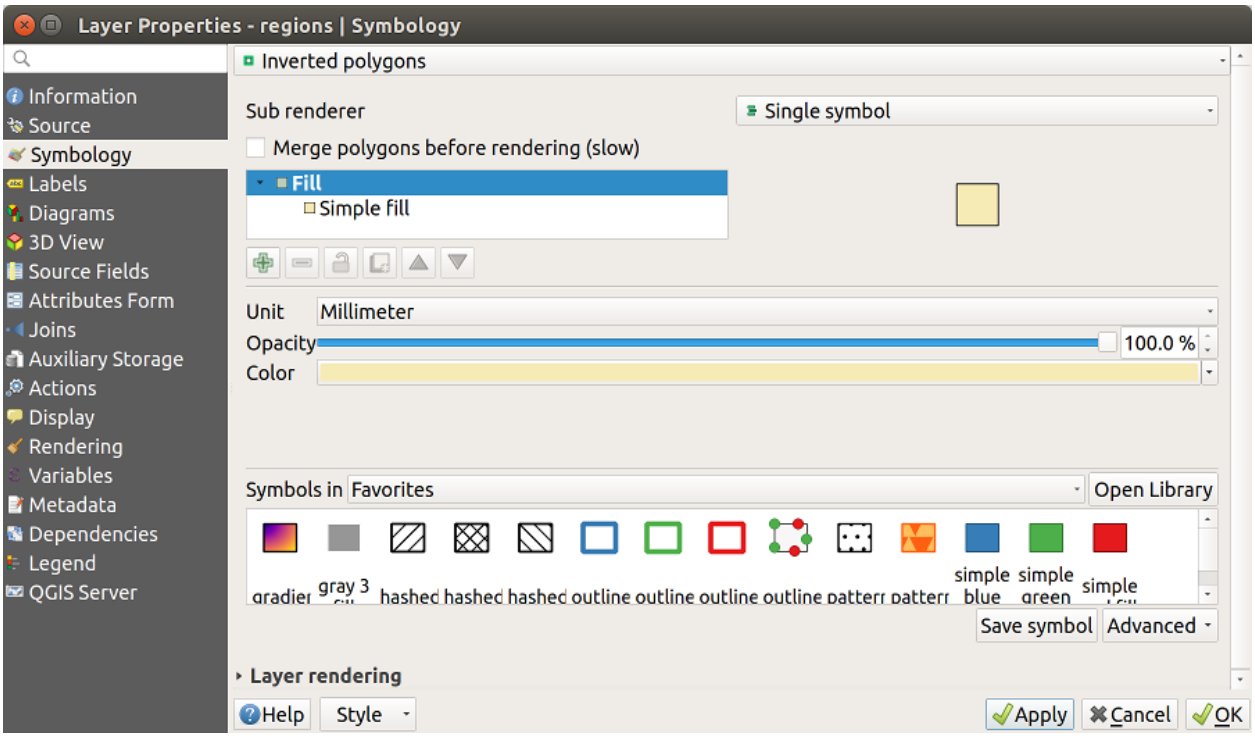


그림 14.11: 반전 폴리곤 대화창

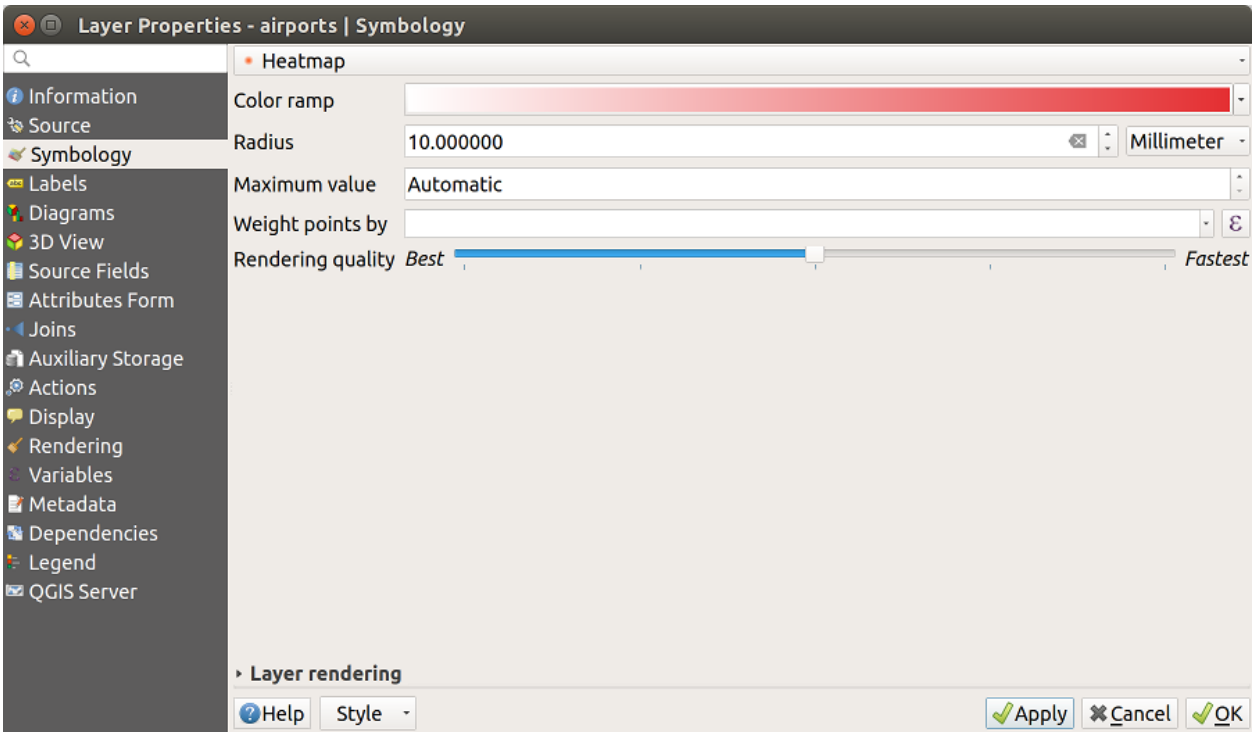


그림 14.12: 열지도 대화창

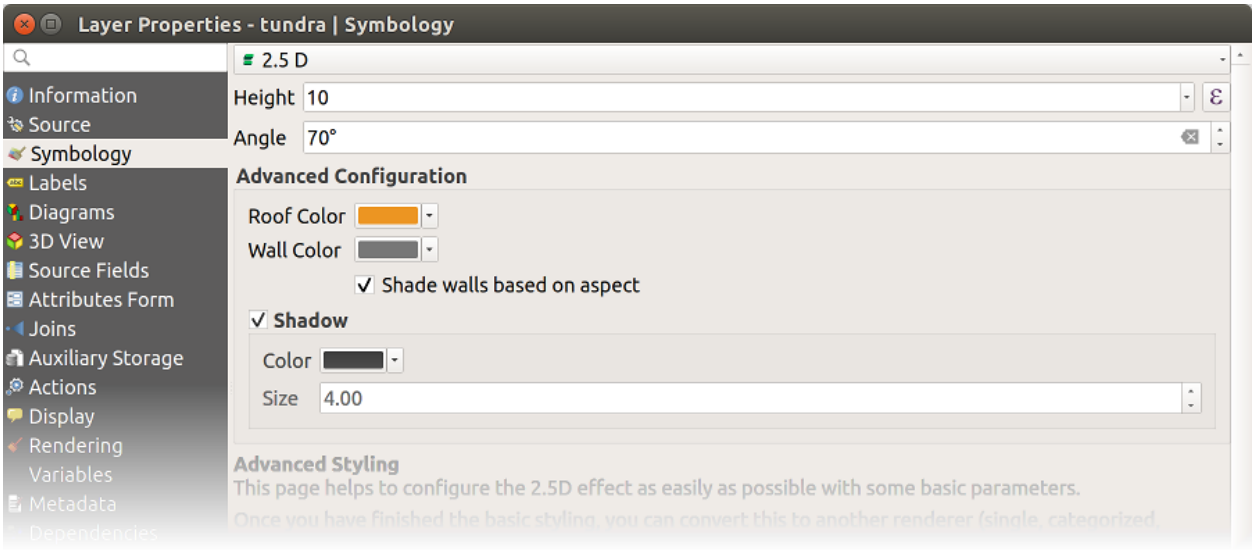


그림 14.13: 2.5D 대화창

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.

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…

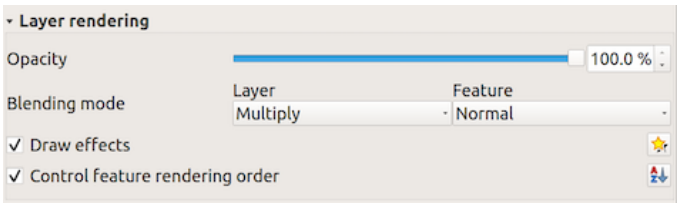


그림 14.14: 레이어 렌더링 옵션들

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.

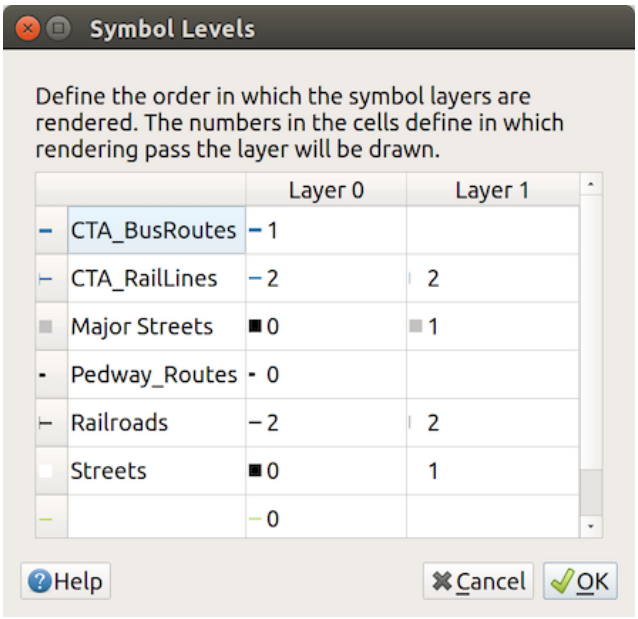


그림 14.15: 심볼 수준 대화창

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.

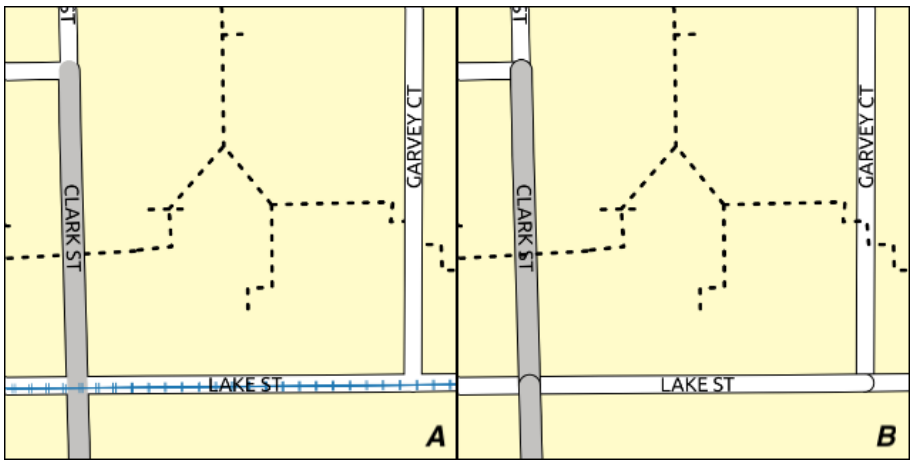


그림 14.16: 심볼 수준 (A) 활성화 (B) 비활성화 비교

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.
- 범례를 표현하는 데사용할 심볼 을 설정합니다.
- 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.

사용자가 파라미터를 설정할 때마다 대화창의 오른쪽 패널에 범례 미리보기가 표시됩니다. 접힌 범례의 경우, 심볼의 수평 중앙에서 대응하는 범례 텍스트를 잇는 인출선 (leader line) 을 그립니다.

참고: 현재, 레이어 심볼을 위한 데이터 정의 크기 범례는 단일 심볼, 범주 또는 등급 심볼을 사용하는 포인트 레이어에만 적용할 수 있습니다.

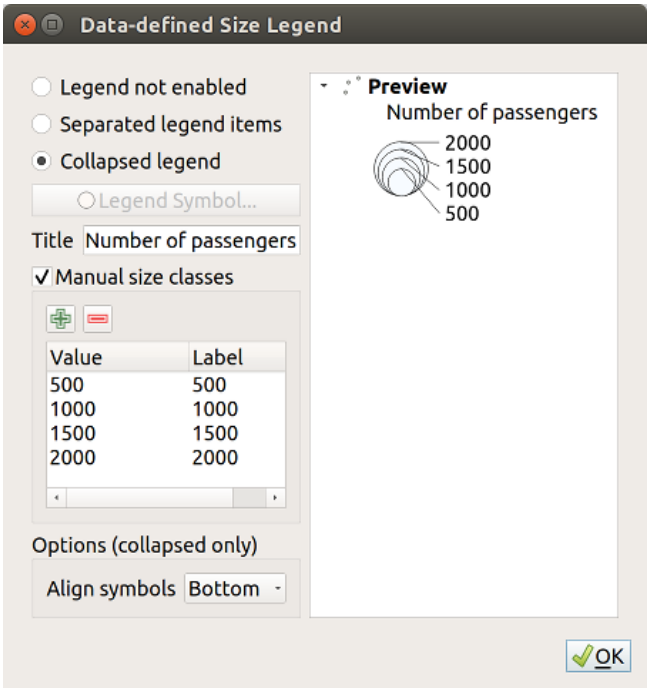


그림 14.17: 크기 척도 범례 설정하기

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 14.18 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 모드와그리기 모드 는 물론, *Opacity* 도 조정할 수 있습니다. 이 세 가지는 모든 효과 유형의 공통 속성입니다.
- **흐리기 (Blur)**: 벡터 레이어에 흐리기 효과를 줍니다. 사용자가 *Blur type* (*Stack blur (fast)* 또는 *Gaussian blur (quality)*) 및 *Blur strength* 를 변경할 수 있습니다.
- **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

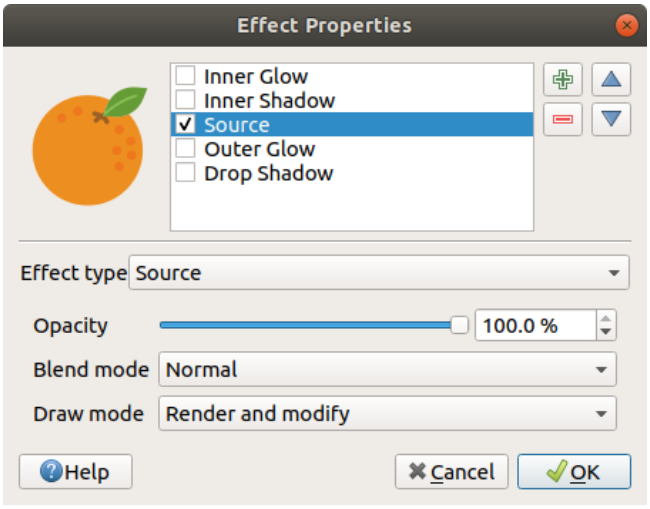


그림 14.18: 그리기 효과: 소스 대화창

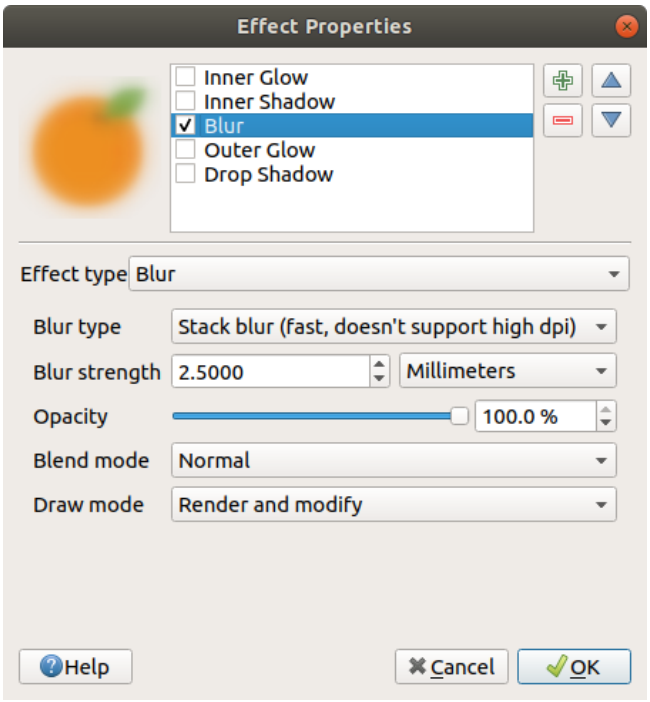


그림 14.19: 그리기 효과: 흐리기 대화창

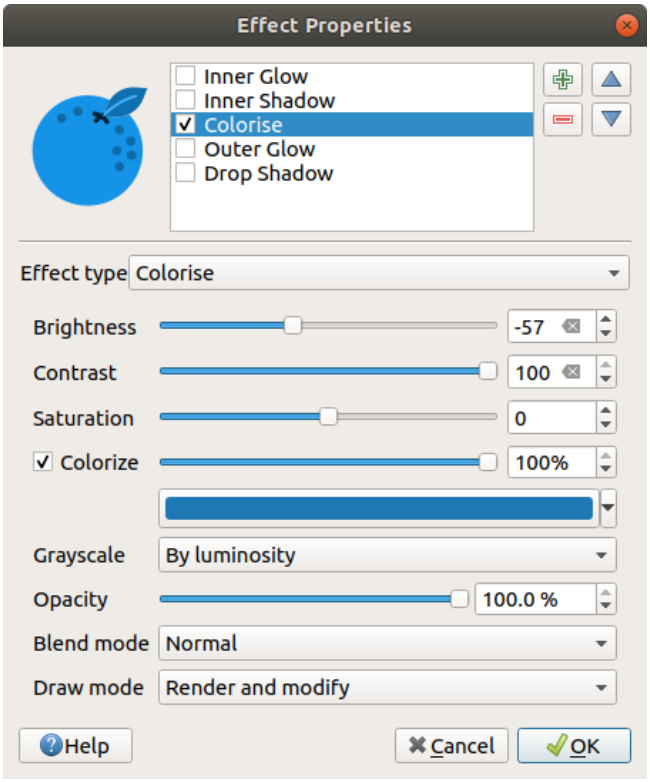


그림 14.20: 그리기 효과: 색상화 대화창

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:
  - *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.

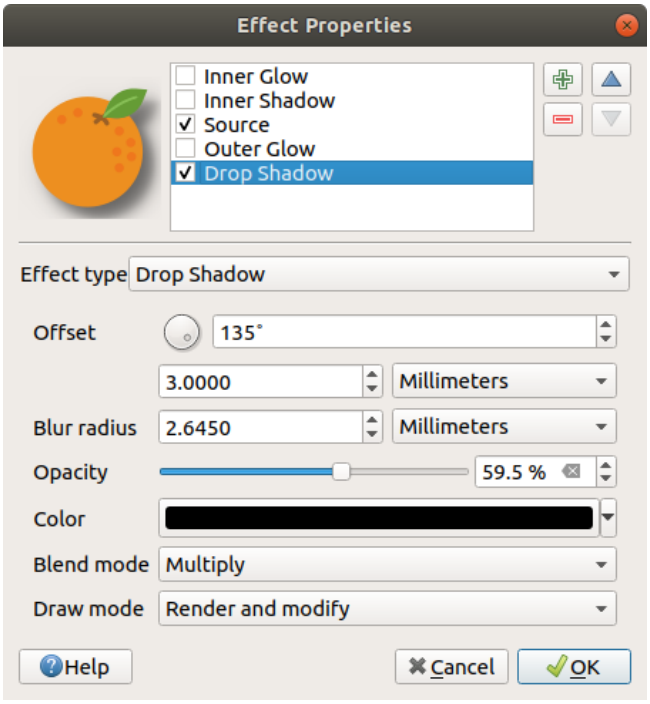


그림 14.21: 그리기 효과: 그림자 생성 효과 대화창

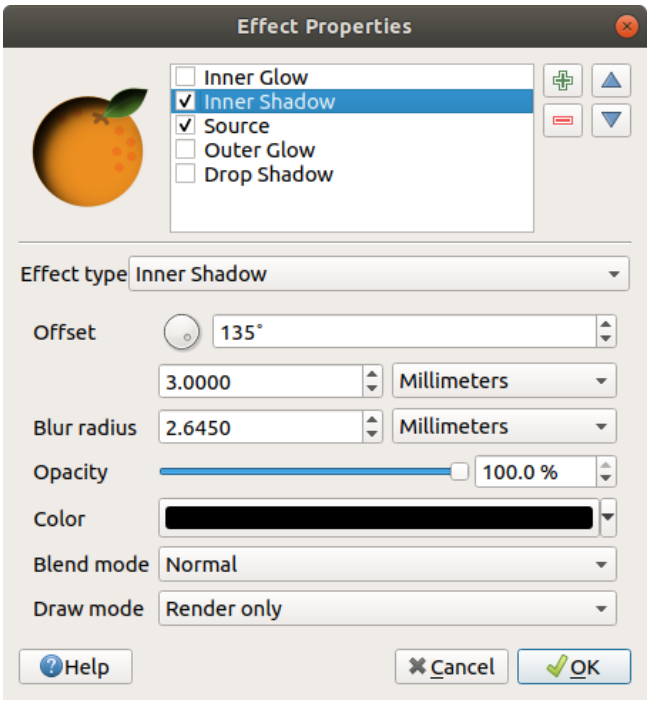


그림 14.22: 그리기 효과: 내부 그림자 대화창

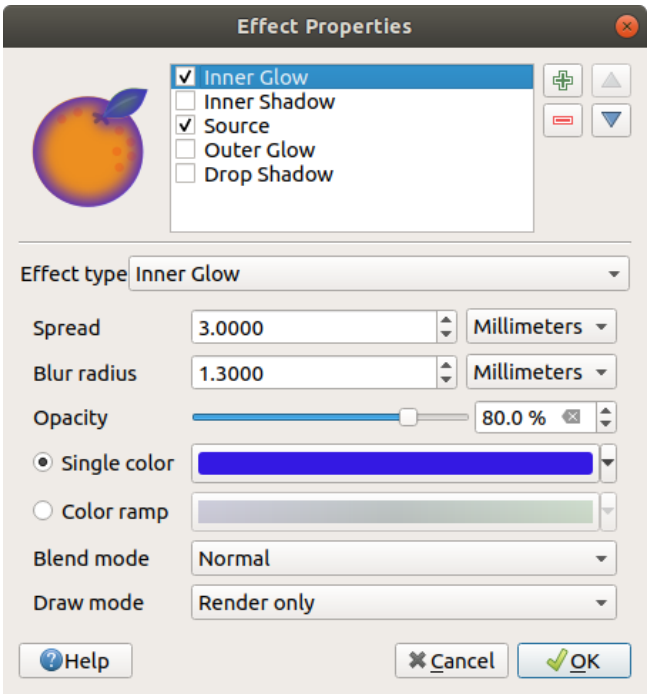


그림 14.23: 그리기 효과: 내부 광원 대화창

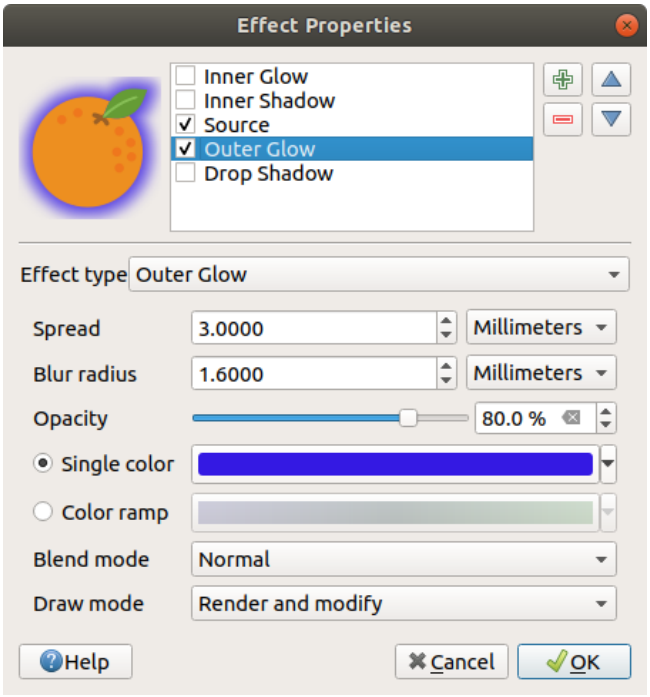


그림 14.24: 그리기 효과: 외부 광원 대화창

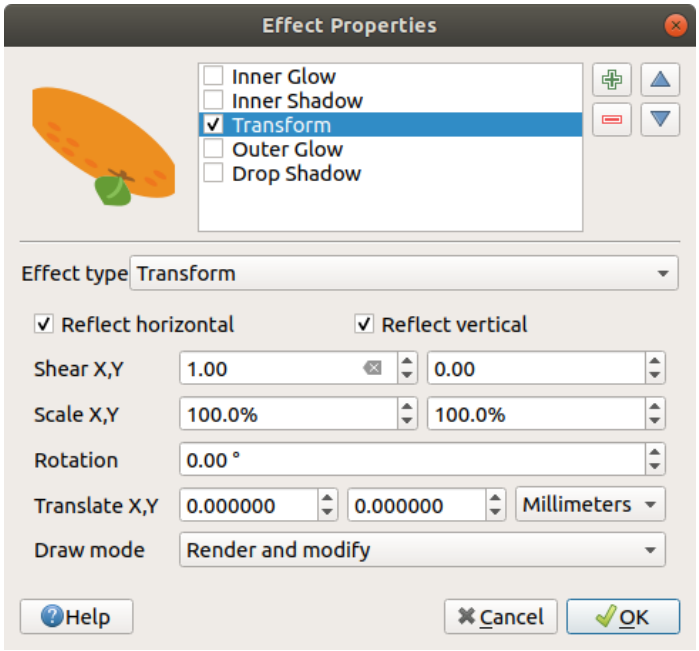


그림 14.25: 그리기 효과: 변형 대화창

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.

14.1.4 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.

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:

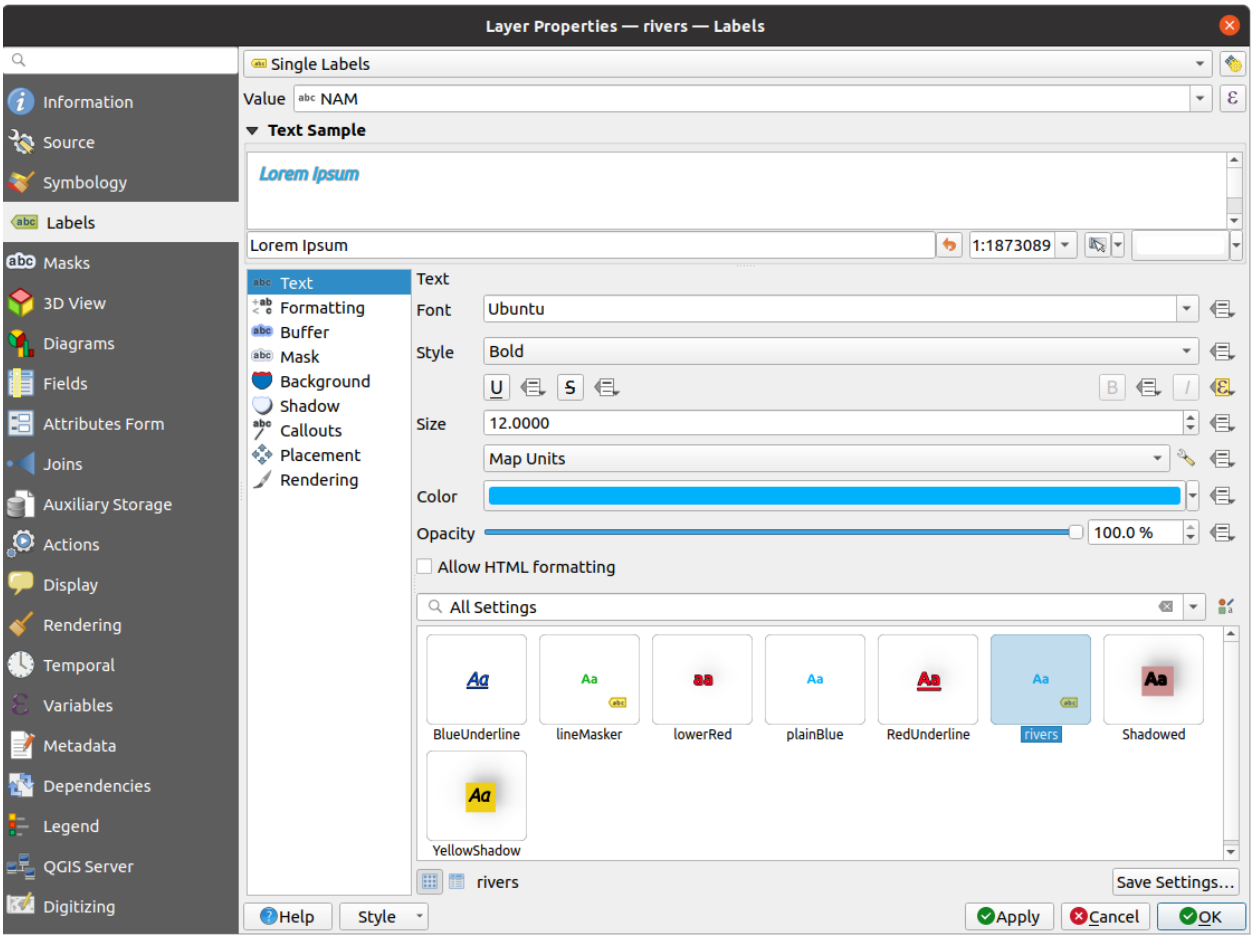


그림 14.26: 레이어 라벨 작업 설정 - 단일 라벨

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.

여러 탭에 라벨을 사용자 지정할 수 있는 다음 옵션들이 표시됩니다:

- abc 텍스트
- 서식 작업
- abc 버퍼
- abc 마스크
- 배경
- 그림자
- 호출
- 배치

- 렌더링

라벨 설정 에서 각 속성을 설정하는 방법을 설명하고 있습니다.

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:

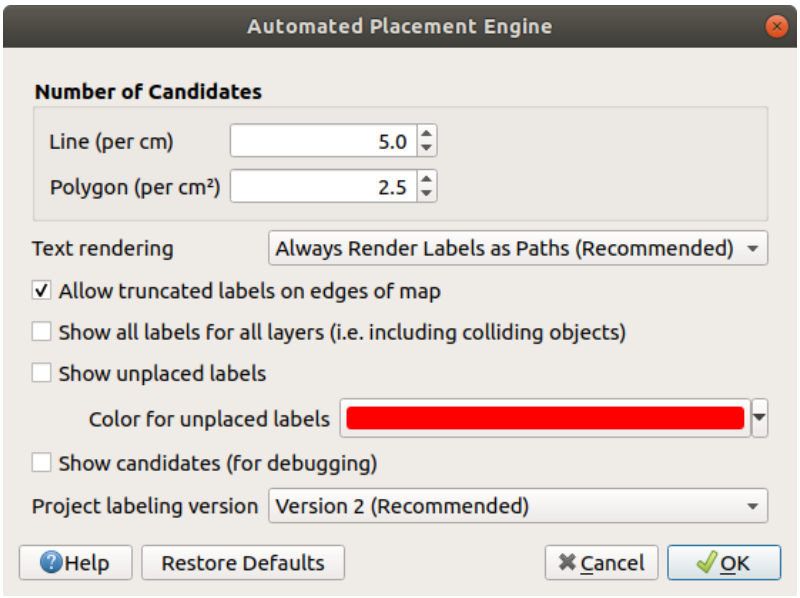


그림 14.27: 라벨 배치 자동화 엔진

- *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*: 맵 캔버스 또는조판 을 PDF 또는 SVG 로 내보낼 때 라벨 렌더링 위젯의 기본값을 설정합니다. *Always render labels as text* 를 선택한 경우 (잉크스케이프 같은) 외부 응용 프로그램에서 라벨을 일반 텍스트처럼 편집할 수 있습니다. **하지만** 렌더링의 질이 하락하는 부작용이 있습니다. **게다가** 버퍼 같은 특정 텍스트 설정이 적용된 경우 렌더링 오류가 생길 수도 있습니다. 라벨을 외곽선으로 내보내는 *Always render labels as paths (recommended)* 옵션을 추천하는 이유입니다.
- *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 명칭대로, 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.

- *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, *Labels* 탭에 있는 주 드롭다운 목록에서 **규칙 기반 라벨 작업** 옵션을 선택한 다음 대화창 하단에 있는 버튼을 클릭하십시오. 새 대화창이 열리면 피처를 필터링할 표현식 및 표현식에 대한 설명을 입력하십시오. 라벨 규칙이 적용돼야 할축척 범위 를 설정할 수도 있습니다. 이 대화창에서 이용할 수 있는 다른 옵션들은 앞에서 설명했던공통 설정 입니다.

주 대화창에 기존 규칙들의 요약을 표시합니다. (그림 14.29 참조) 규칙을 여러 개 추가할 수 있고, 드래그 & 드롭으로 규칙들을 재배열하거나 연결시킬 수도 있습니다. 버튼으로 규칙을 제거할 수도 있고, 규칙을 더블 클릭하거나 버튼을 클릭하면 규칙을 편집할 수도 있습니다.

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 14.30 은 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.

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:

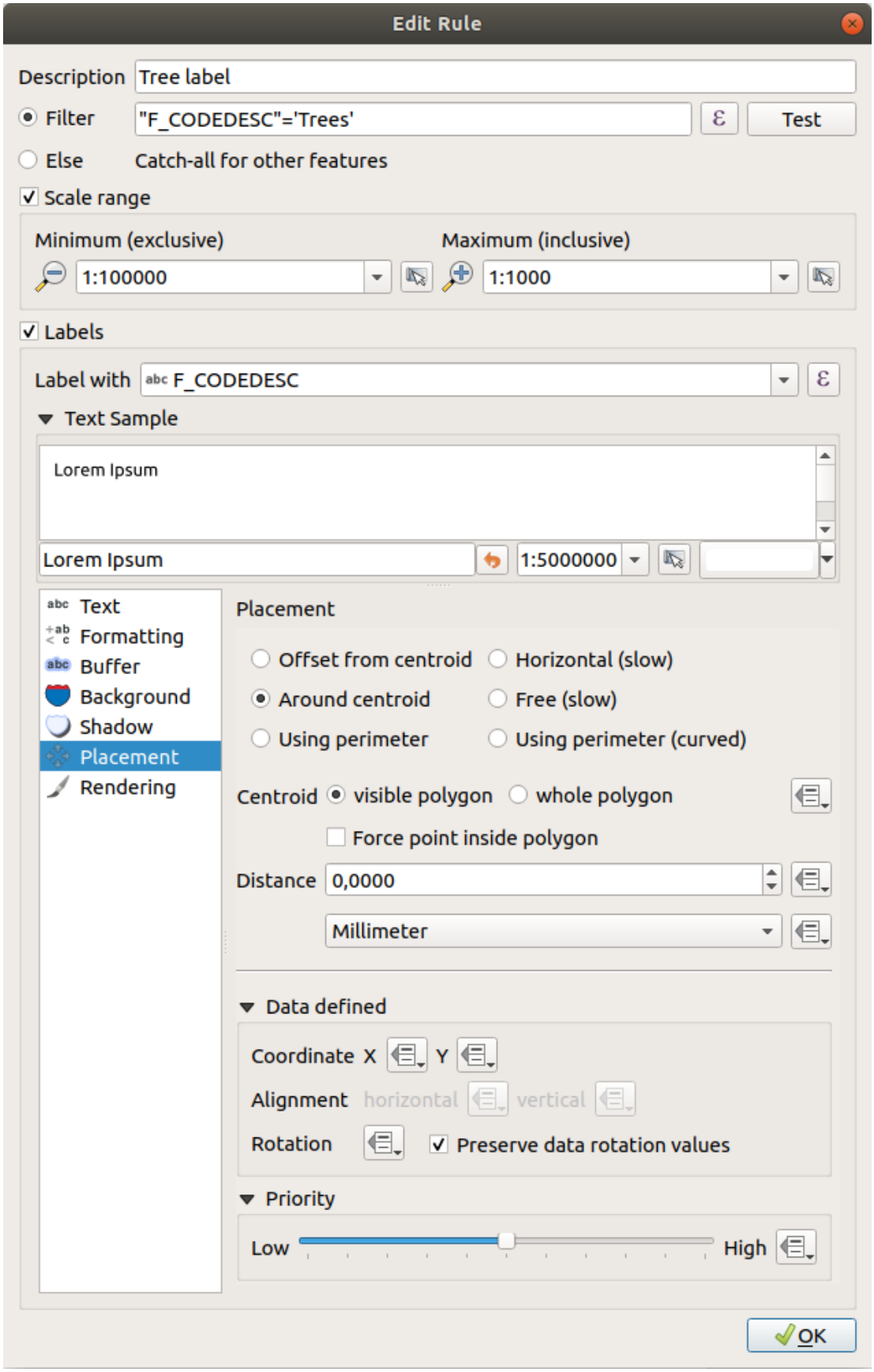

Edit Rule
Description
Tree label
Filter
"F_CODEDESC"='Trees'
Test
Else
Catch-all for other features
Scale range
Minimum (exclusive)
1:100000
Maximum (inclusive)
1:1000
Labels
Label with
F_CODEDESC
Text Sample
Lorem Ipsum
Lorem Ipsum
1:5000000
Text
Formatting
Buffer
Background
Shadow
Placement
Rendering
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 14.28: 규칙 설정

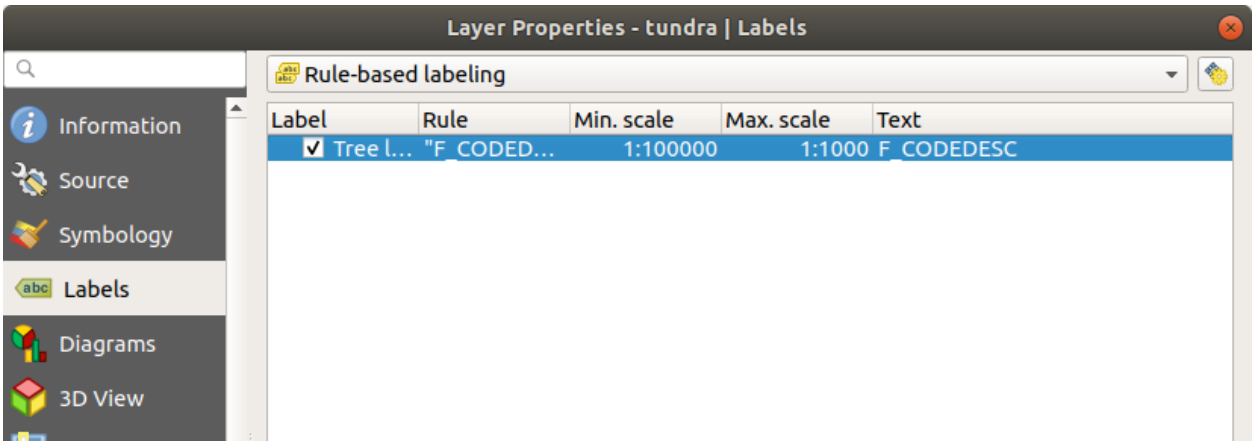


그림 14.29: 규칙 기반 라벨 작업 패널

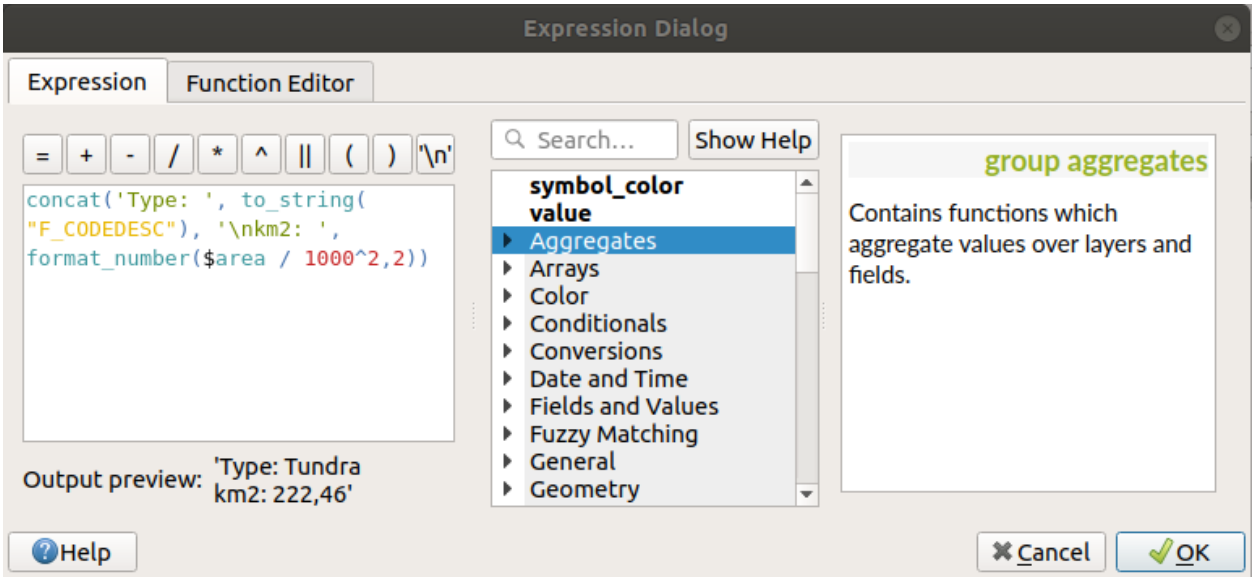


그림 14.30: 라벨 작업에 표현식 이용하기

```
"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 명칭만 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 명칭 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.

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.

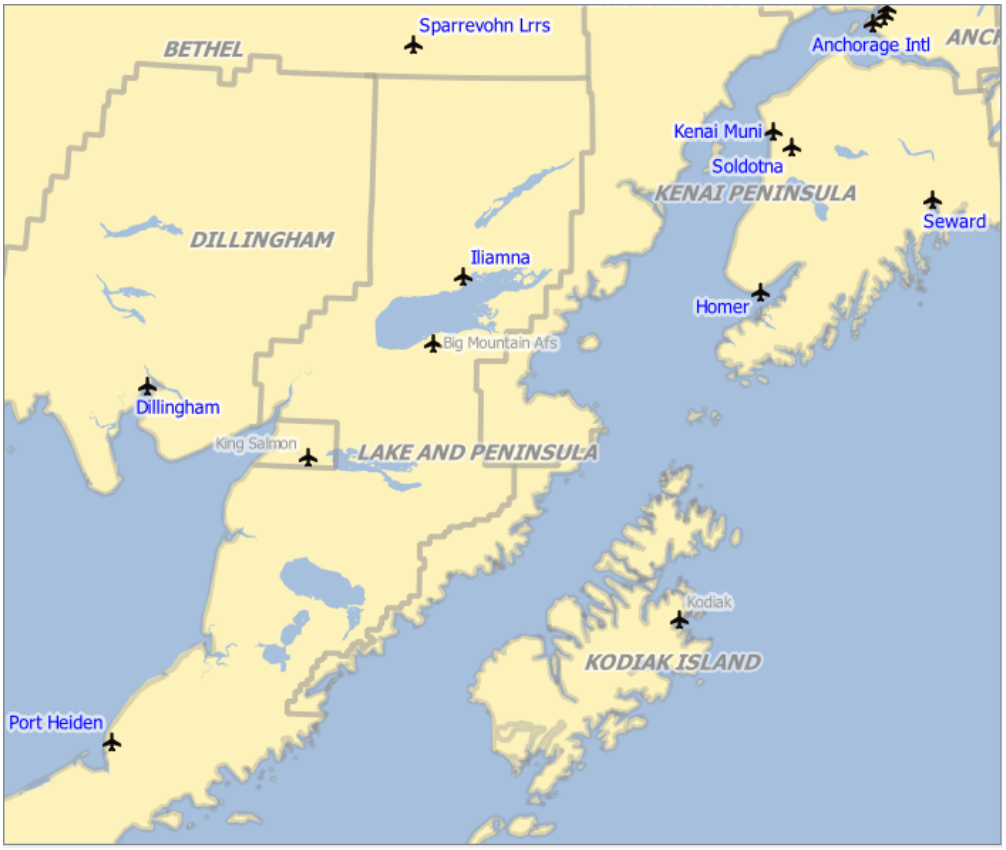


그림 14.31: 공항의 속성을 기반으로 서식을 작업한 공항 라벨

라벨 툴바

Label Toolbar 는 라벨 또는 도표 속성을 조작할 수 있는 몇몇 도구를 제공합니다.



그림 14.32: 라벨 툴바

가독성을 위해 라벨 툴바를 설명하는 데"라벨"이란 단어를 쓰고 있기는 해도, 도구 명칭으로 언급되는 경우 도구들이 도표와 거의 동일한 방식으로 동작한다는 점을 기억하십시오:

- Highlight Pinned Labels and Diagrams: 라벨이 있는 벡터 레이어가 편집 가능한 상태인 경우 라벨의 핀을 초록색으로 강조합니다. 아닌 경우 핀은 파란색입니다.
- Toggles Display of Unplaced Labels: 어떤 중요한 라벨이 (예를 들면 중첩 또는 다른 제약조건 때문에) 맵에서 없어졌는지 여부를 확인할 수 있습니다. 이런 라벨은 사용자 지정 색상으로 표시됩니다. (자동화된 배치 엔진 설정하기 참조)
- Pin/Unpin Labels and Diagrams: 영역을 클릭 또는 드래그해서 라벨 (들) 에 핀을 꽂습니다. `Shift` 키를 누른 채 영역을 클릭하거나 드래그하면 라벨 (들) 의 핀을 뺍니다. 마지막으로 `Ctrl` 키를 누른 채 영역을 클릭하거나 드래그하면 라벨 (들) 의 핀 상태를 반전시킬 수 있습니다.
- Show/Hide Labels and Diagrams: `Shift` 키를 누른 채 라벨을 클릭하거나 또는 영역을 클릭하거나 드래그하면 라벨을 숨깁니다. 라벨이 숨겨진 상태일 경우, 그 가시성을 복구하려면 피처를 클릭하기만 하면 됩니다. 영역을 드래그하면 해당 영역의 모든 라벨의 가시성이 복구될 것입니다.
- Moves a Label or Diagram: 원하는 위치로 라벨을 드래그하기만 하면 됩니다.
- Rotate Label: 라벨을 클릭한 채 마우스를 움직이면 텍스트가 기울어집니다.
- 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

데이터 정의 무시 설정과 함께 *Label Toolbar* 를 이용하면, 맵 캔버스에 있는 라벨을 조작 (이동, 편집, 회전) 할 수 있습니다. 이제 Move label 기능에 데이터 정의 무시 기능을 사용하는 예시를 설명하겠습니다. (그림 14.33 을 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.

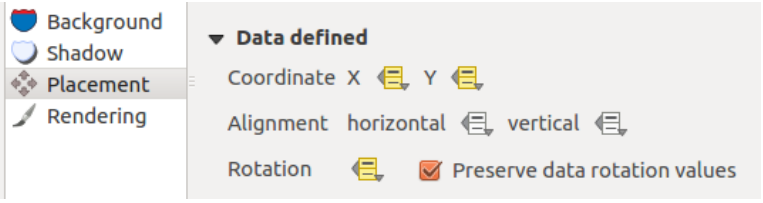


그림 14.33: 데이터 정의 무시를 통한 벡터 폴리곤 레이어의 라벨 작업

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 14.34 를 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.

14.1.5 도표 속성

Diagrams 탭에서 벡터 레이어에 그래픽을 중첩시킬 수 있습니다. (그림 14.35 참조)

현재 도표의 핵심 구현은 다음을 지원하고 있습니다:

- *No diagrams*: 기본값입니다. 피처 위로 어떤 도표도 표시하지 않습니다.
- *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*: 텍스트 도표는 원을 수평으로 나누어 그 조각들 안에 통계값을 표시합니다.

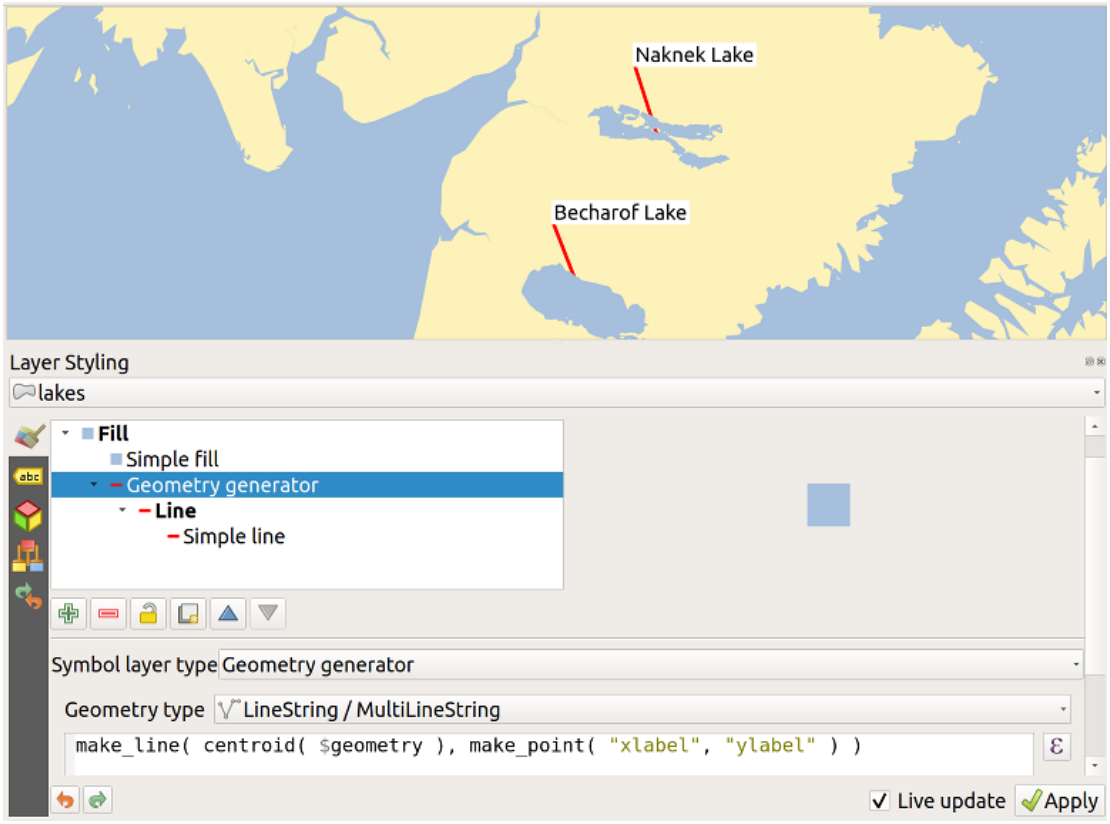


그림 14.34: 이동시킨 라벨

-  *Histogram*: 히스토그램은 각 속성을 표현하는 여러 색상의 막대들을 연달아 정렬한 도표입니다.
-  *Stacked bars*: 누적 막대 그래프로, 각 속성을 표현하는 여러 색상의 막대들을 수직 또는 수평으로 연달아 쌓습니다.

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.

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:

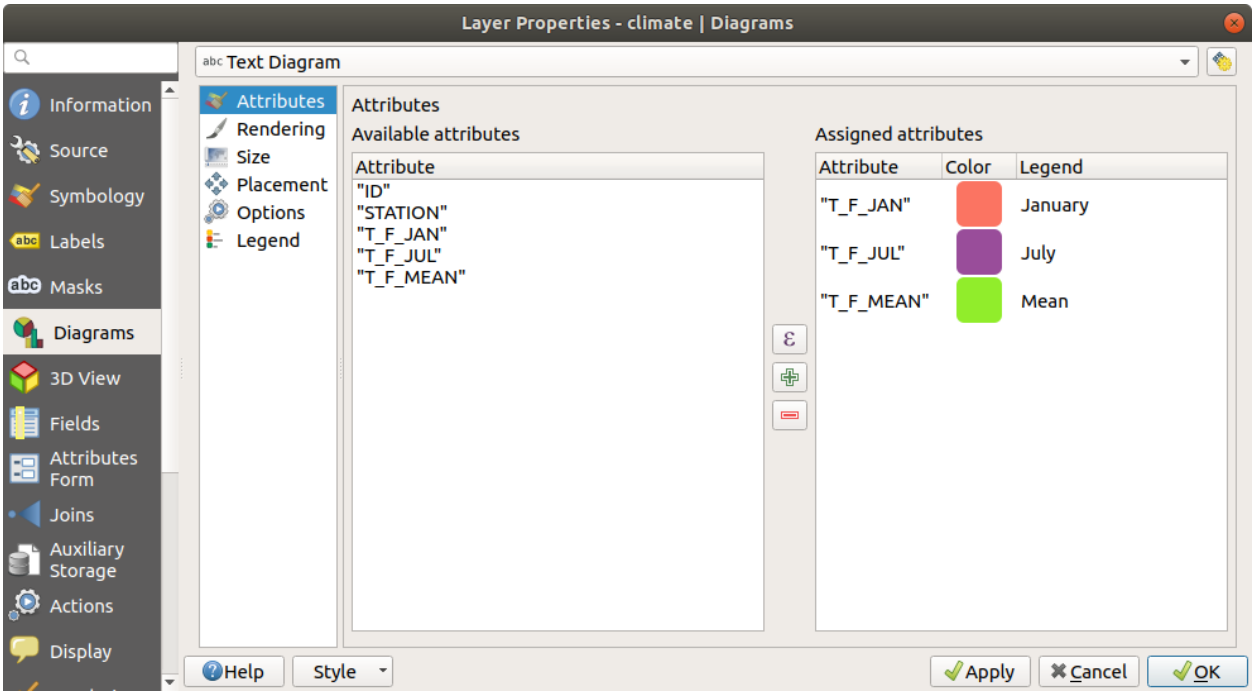


그림 14.35: 도표 속성 - 속성 탭

- *Diagram z-index*: 도표들이 어떤 순서로 그려지는지, 그리고 라벨 위에 그려질지 여부를 제어합니다. 높은 인덱스를 가진 도표가 다른 도표 및 라벨 위에 그려집니다.
- *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.

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.

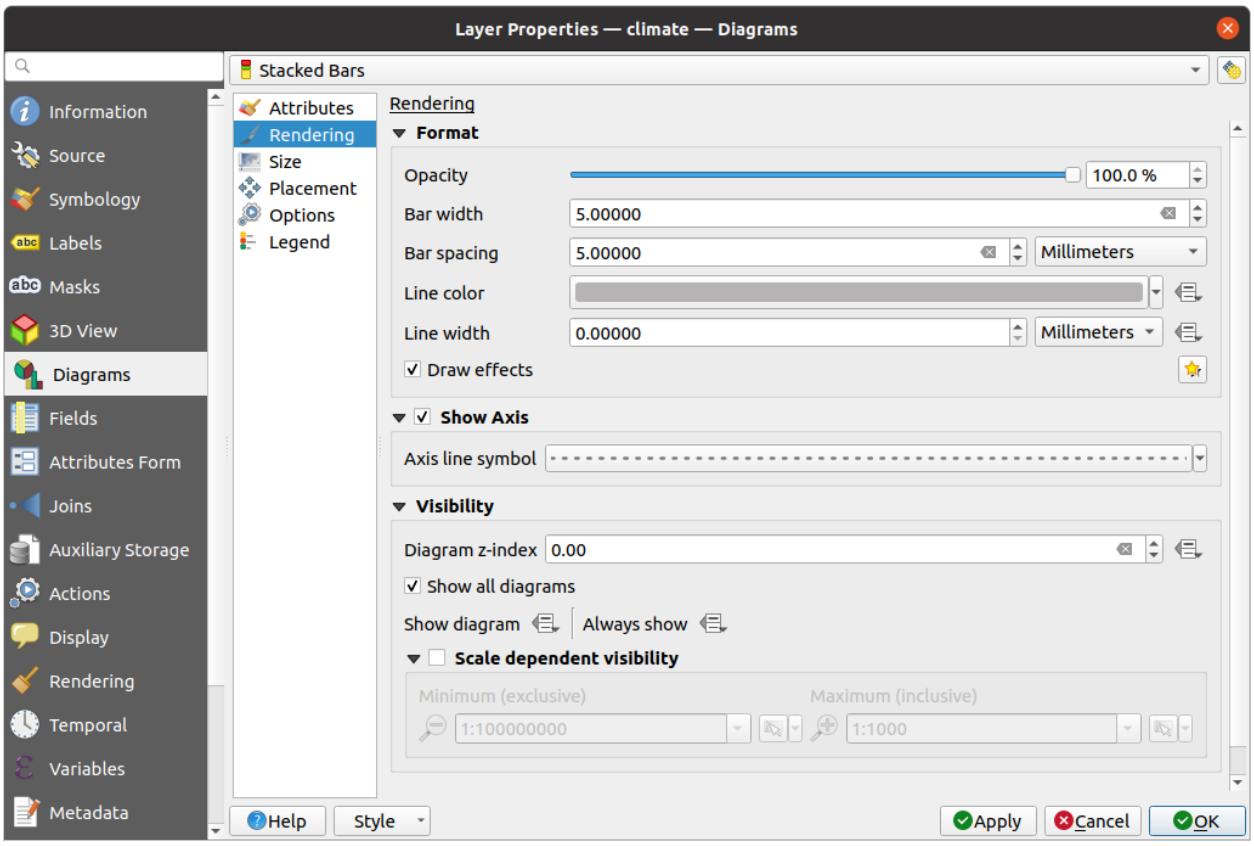


그림 14.36: 도표 속성 - 렌더링 탭

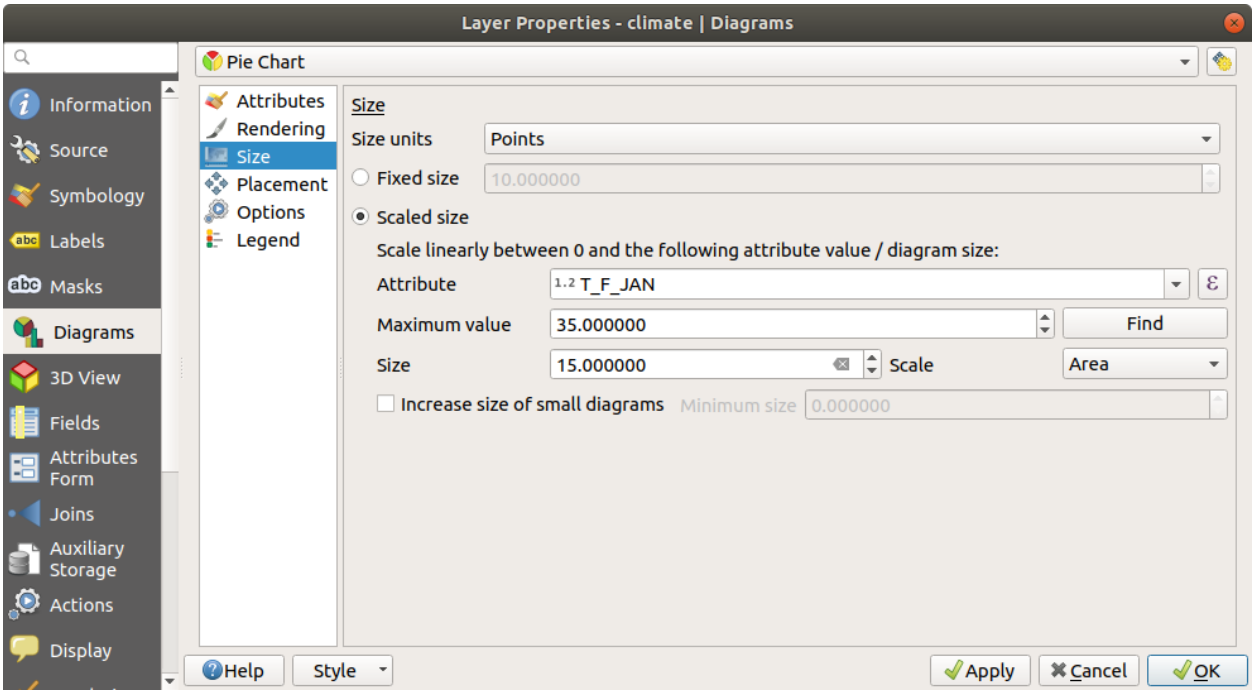


그림 14.37: 도표 속성 - 크기 탭

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.

Priority 부분에서 각 도표의 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.

Discourage diagrams and labels from covering features: 방해물 로 사용할 피처를 정의합니다. 예를 들어 QGIS 는 이런 피처 위로 도표든 라벨이든 배치하지 않으려 할 것입니다. 그 다음 우선 순위를 사용해서 더 높은 가중치를 가진 방해물로 인해 도표를 누락해도 될지를 평가합니다.

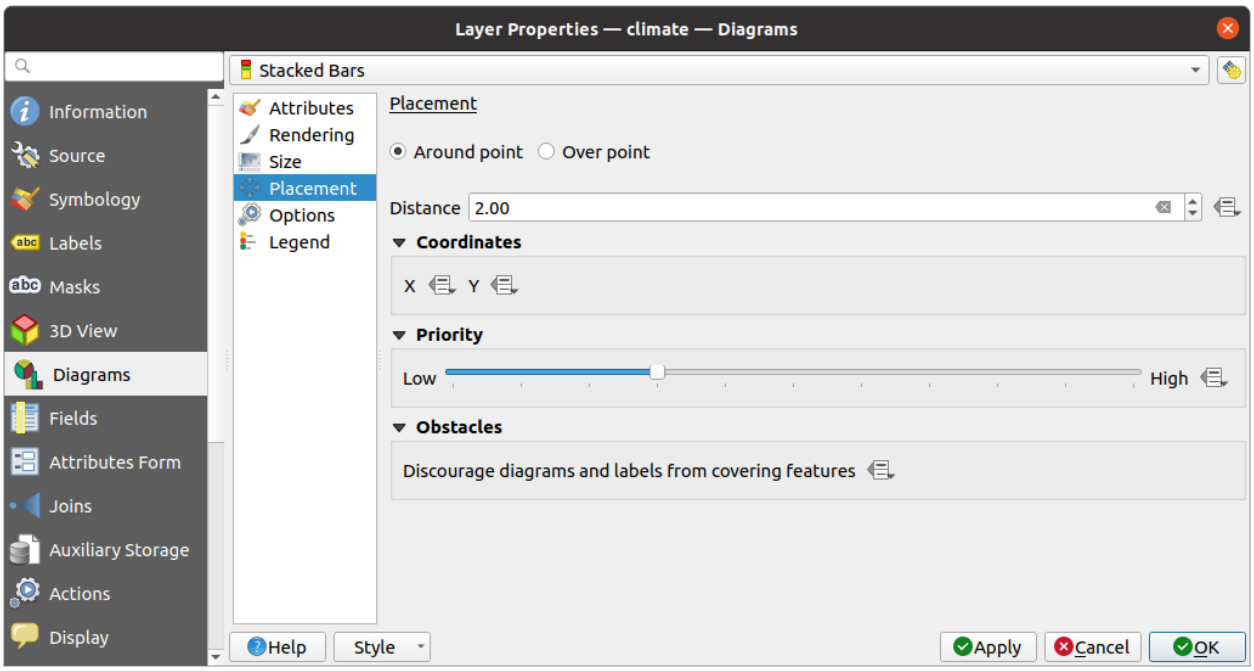


그림 14.38: 도표 속성 - 배치 탭

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.

14.1.6 마스크 속성

The *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.

14.1.7 3D 뷰 속성

3D View 탭은*3D* 맵 뷰 도구에서 묘사되어야 할 벡터 레이어를 위한 설정을 제공합니다.

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 잎새 (leaf) 수준에서 타일이 16 개가 됩니다. (모든 추가적인 확대/축소 수준은 타일 개수가 4 배수가 됩니다.) 기본값은 `3` 이며, 최대값은 `8` 입니다.
- ☑ *Show bounding boxes of tiles*: 표시되어야 할 타일이 표시되지 않는 문제가 있을 경우 특히 유용한 옵션입니다.

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.

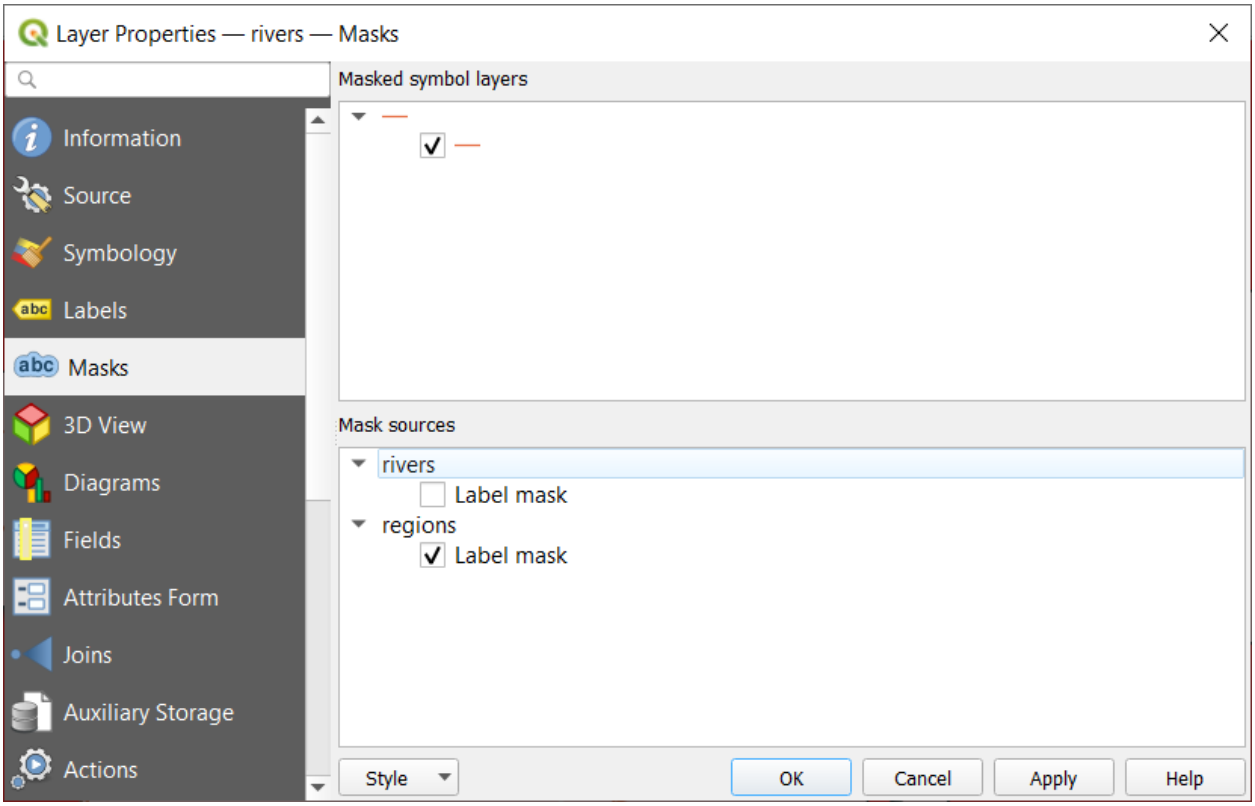


그림 14.39: 레이어 속성 - 마스크 탭

14.1.8 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.

필드명을 더블클릭하면 필드를 재명명할 수도 있습니다. PostgreSQL, Oracle, 메모리 레이어 및 (OGR 데이터 유형 및 버전에 따라 달라지는) 일부 OGR 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.

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.

가상 필드와보조 저장소 를 포함하는 데이터셋에 담겨진 필드 외에도, *Fields* 탭은 모든결합 레이어 *(joined layer)* 의 필드들의 목록도 가지고 있습니다. 필드의 출처에 따라 서로 다른 배경 색상을 적용합니다.

이 대화창은 목록의 각 필드에 대해 필드의 `type`, `type name`, `length` 및 `precision` 같은 읽기 전용 특성도 목록화합니다. 레이어를 `WMS` 또는 `WFS` 로 서비스하는 경우, 여기에서 어떤 필드를 가져올 수 있는지도 확인할 수 있습니다.

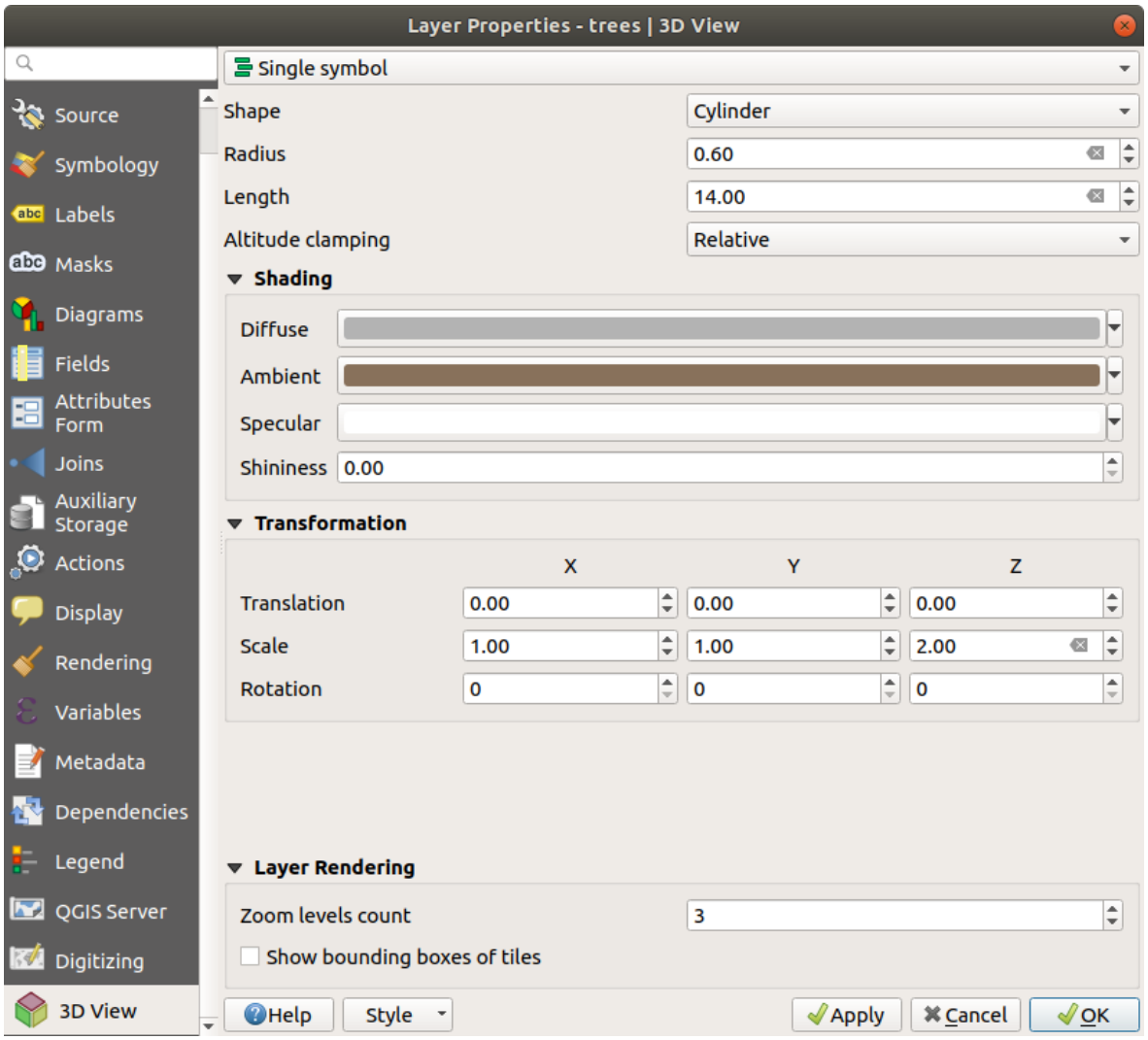


그림 14.40: 포인트 레이어의 3 차원 속성

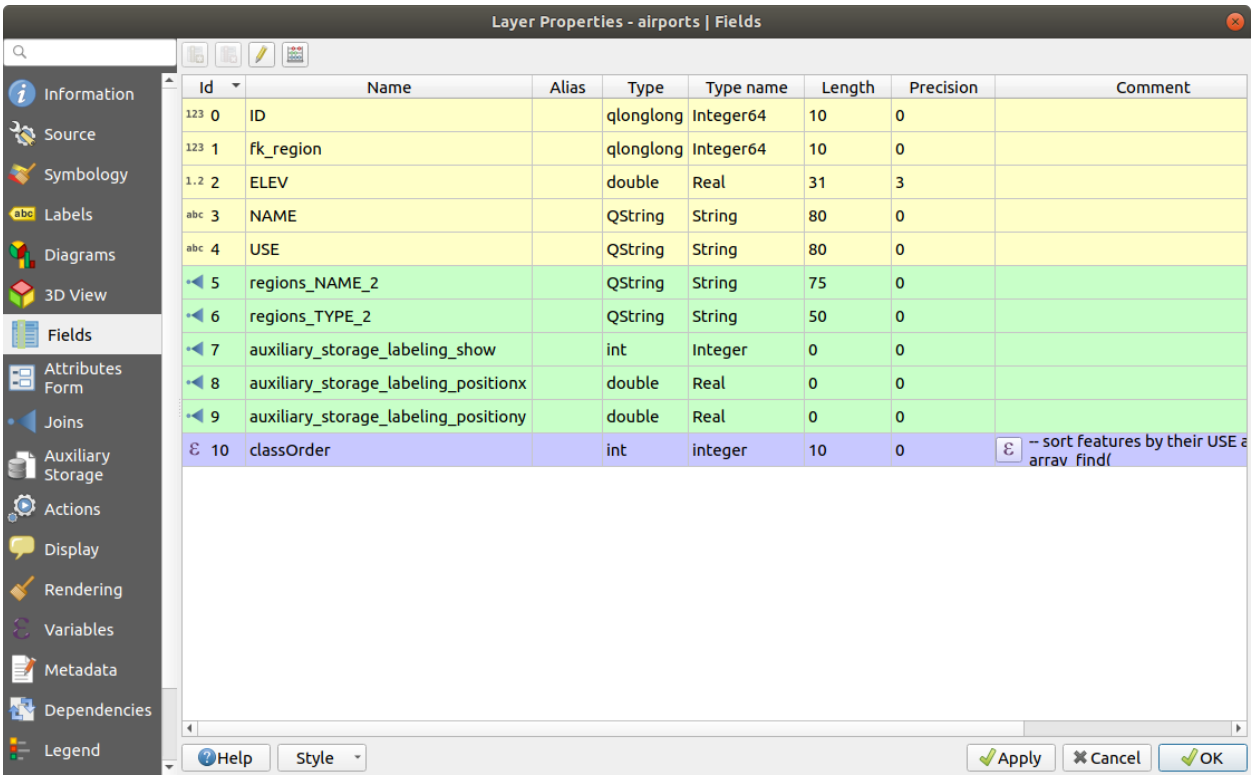


그림 14.41: 필드 속성 탭

14.1.9 속성 양식 속성

Attributes Form 탭에서 새 피처를 생성하거나 기존 피처를 쿼리할 때 표시할 양식을 설정할 수 있습니다. 다음을 정의할 수 있습니다:

- 피처 양식 또는 (라벨, 위젯, 제약조건 등등의) 속성 테이블에 있는 각 필드의 모양 및 습성
- 양식의 구조 (사용자 지정 또는 자동 생성)
- 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 이외에도, 더 복잡한 양식 구조를 만들 수 있습니다. 예를 들면 위젯을 그룹 및 탭에 내장시킬 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.

자동 생성 양식

`Autogenerate` 옵션이 활성화된 경우, *Available widgets* 패널이 양식에 표시될 (레이어와 레이어 관계에서 나온) 필드 목록을 표시합니다. 필드를 선택하면 오른쪽 패널에서 필드의 모양 및 습성의 환경을 설정할 수 있습니다:

- 필드에사용자 지정 라벨과 자동화 점검 을 추가할 수 있습니다.
- 사용할특정 위젯 을 설정할 수 있습니다.

드래그 & 드롭 설계자

사용자는 드래그 & 드롭 설계자를 통해 속성 필드를 나타내기 위한 여러 컨테이너를 (탭 또는 그룹을) 가진 양식을 생성할 수 있습니다. 그림 14.42 의 예시를 참조하세요.

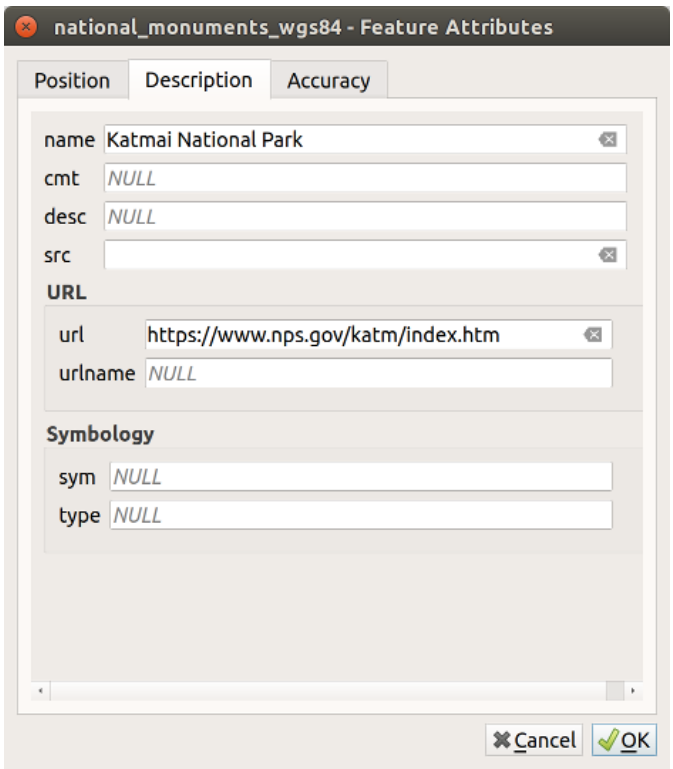


그림 14.42: 탭과 명명된 그룹을 가진 내장 양식 생성의 결과물

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 않을 필드를 선택하고 버튼을 클릭하면 선택한 필드를 제거합니다. 다른 패널에서 제거한 필드를 드래그 & 드롭하면 다시 추가할 수 있습니다. 동일한 필드를 몇 번이고 추가할 수도 있습니다.

3. *Form Layout* 패널에 있는 필드를 드래그 & 드롭하면 필드 위치를 재배열할 수 있습니다.
4. 동일 카테고리에 속한 관련 필드에 컨테이너를 (탭 또는 그룹 프레임을) 추가하면 더 나은 양식 구조를 만들 수 있습니다.

   1. 먼저 아이콘을 클릭해서 필드 및 그룹을 표시할 탭을 생성하십시오.
   2. 그리고 컨테이너의 속성을 다음과 같이 설정하십시오:
      - 명칭
      - 유형: 예를 들어 *tab* 또는 *group in container* (탭 또는 또다른 그룹에 내포된 그룹)
      - *Number of columns*: 내장된 필드가 배포되어야 할 열의 개수

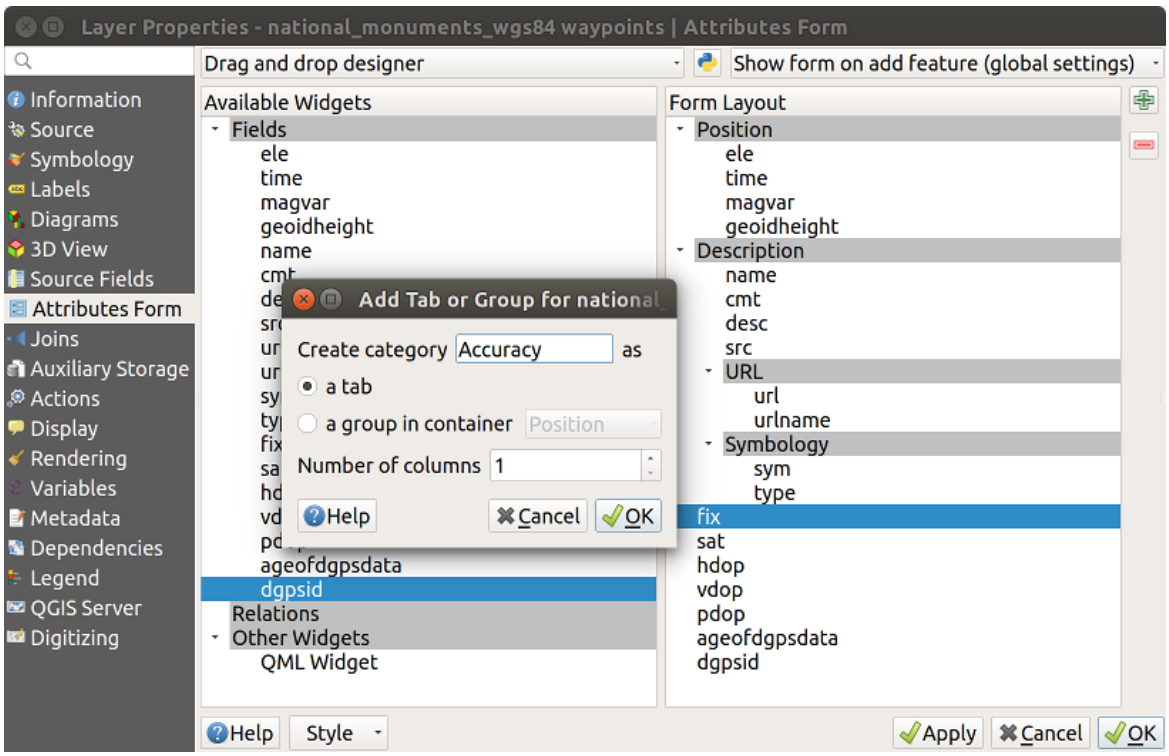


그림 14.43: **속성 편집기 조판** 으로 컨테이너를 생성하는 대화창

이후에 항목을 선택한 다음 세 번째 패널에서 다음 설정을 하면 이 속성들과 기타 속성들을 업데이트할 수 있습니다:

- 컨테이너의 라벨을 숨기기 또는 표시하기
- 컨테이너를 그룹 상자로 표시하기 (탭인 경우에만 사용할 수 있습니다)
- 컨테이너 명칭을 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.
- 배경 색상을 추가하기

3. 사용자가 원하는 만큼 많은 컨테이너를 생성할 수 있습니다. 기존 탭 아래 또다른 탭 또는 그룹 프레임을 생성하려면 아이콘을 클릭하십시오.

5. 다음 단계는 각 컨테이너에 관련 필드를 할당하는 것입니다. 그냥 드래그 & 드롭하면 됩니다. 그룹 및 탭도 같은 방법으로 옮길 수 있습니다.
6. 사용 중인 필드의위젯을 사용자 지정 하십시오.
7. 레이어가일대다 또는 다대다 관계 에 속해 있는 경우, *Available widgets* 패널에서 *Form Layout* 패널로 관계명을 드래그 & 드롭하면 됩니다. 현재 레이어의 양식에서 지정한 위치에 관련 레이어 속성 양식을 내포시킬 것입니다. 다른 항목의 경우, 관계 라벨을 선택하면 다음 몇몇 속성을 환경 설정할 수 있습니다:
   - 관계 라벨을 숨기기 또는 표시하기
   - 링크 버튼 표시하기
   - 링크 해제 버튼 표시하기
8. 레이어 속성 대화창의 *Apply* 버튼을 클릭합니다.
9. 피처 속성 양식을 (예를 들어  Identify features 도구를 사용해서) 열면, 새 양식을 표시할 것입니다.

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 명칭으로 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.

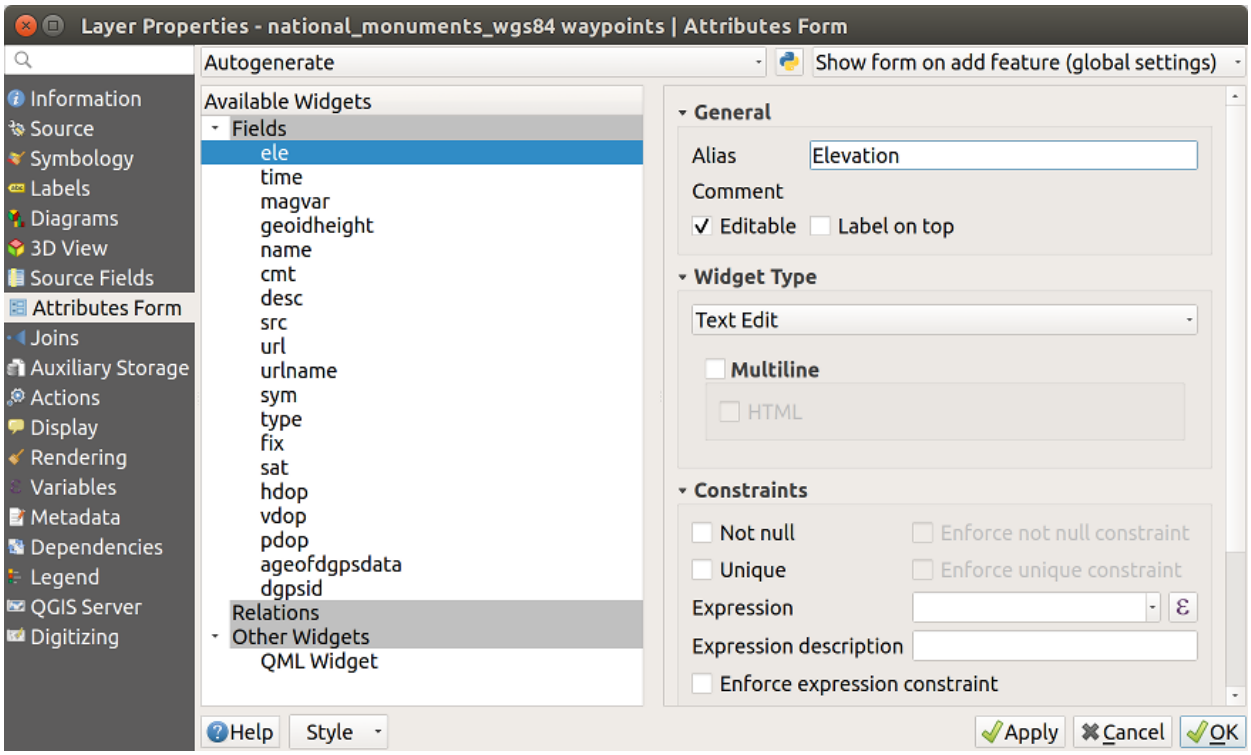


그림 14.44: 속성 열을 위한 편집 위젯을 선택할 수 있는 대화창

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

Show label: 양식에 필드명을 표시해야 하는지 여부를 나타냅니다. (드래그 & 드롭 설계자 모드에서만)

일반 옵션

- *Alias*: 필드용으로 쓰이는, 사람이 읽을 수 있는 명칭입니다. 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 제한도 무시하지 않는다는 사실을 기억하십시오.
- ☑ *Label on top*: 피처 양식에 있는 위젯의 위 또는 옆에 필드명을 배치합니다.

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.

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"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 주황색 가위표가 나타납니다.

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:

- **체크박스 (Checkbox)**: 그 상태가 삽입하는 값을 정의하는 체크박스를 표시합니다.
- **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.
- **첨부 (Attachment)**: 파일 경로를 상대 또는 절대 경로로 저장하기 위한"Open file"대화창을 사용합니다. (문서 경로를 가리키는) 하이퍼링크, 이미지 또는 웹페이지를 표시하는 데에도 사용할 수 있습니다.
- **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.
- **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.

- **관계성 참조 (Relation Reference)**: 관계 가 설정된 경우 참조하는 필드 (예: 자식 레이어의 외래 키) 에 할당되는 기본 위젯입니다. 자식 피처의 목록과 양식을 내장한 부모 피처의 양식에 직접 접근할 수 있습니다.
- **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.
- **값 지도 (Value Map)**: 사전 정의된 항목을 가지고 있는 콤보박스를 표시합니다. 값은 속성에 저장돼 있으며, 설명은 콤보박스에 표시됩니다. 값을 직접 정의할 수도 있고, 레이어 또는 CSV 파일에서 불러올 수도 있습니다.
- **값 관계 (Value Relation)**: 콤보박스에 관련 테이블에서 가져온 값을 제공합니다. 레이어, 키 열, 값 열 가운데 하나를 선택할 수 있습니다. 표준 습성을 변경할 수 있는 몇몇 옵션들이 – NULL 값 허용, 값으로 정렬, 다중 선택 허용 및 자동완성 사용–있습니다. 이 양식은 드롭다운 목록을 표시하거나, 자동완성 체크박스를 활성화한 경우 한 줄짜리 텍스트 편집란을 표시할 것입니다.

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.

14.1.10 결합 속성



Joins 탭에서, 현재 레이어의 (`Target layer` 의) 피처와 또다른 불러온 벡터 레이어의 (또는 테이블의) 피처를 결합시킬 수 있습니다. 결합은 레이어가 공유하는 속성을 기반으로 합니다. 레이어가 (테이블이) 무 (☐) 도형일 수도 아닐 수도 있지만 두 레이어의 결합 속성만큼은 동일한 유형이어야 합니다.

결합을 생성하려면:

1.  Add new join 아이콘을 클릭하면 *Add vector join* 대화창이 열립니다.
2. 대상 벡터 레이어와 연결하려는 *Join layer* 를 선택하십시오.
3. 결합 레이어와 대상 레이어 양쪽에 공통으로 존재하는 *Join field* 와 *Target field* 를 지정하십시오.
4. *OK* 를 누르면 *Join* 패널에 선택한 파라미터들의 요약이 추가됩니다.

앞의 단계들을 통해 생성된 결합은, 결합 레이어에서 처음으로 일치하는 피처의 **모든** 속성을 대상 레이어의 피처에 추가시킵니다. 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 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.

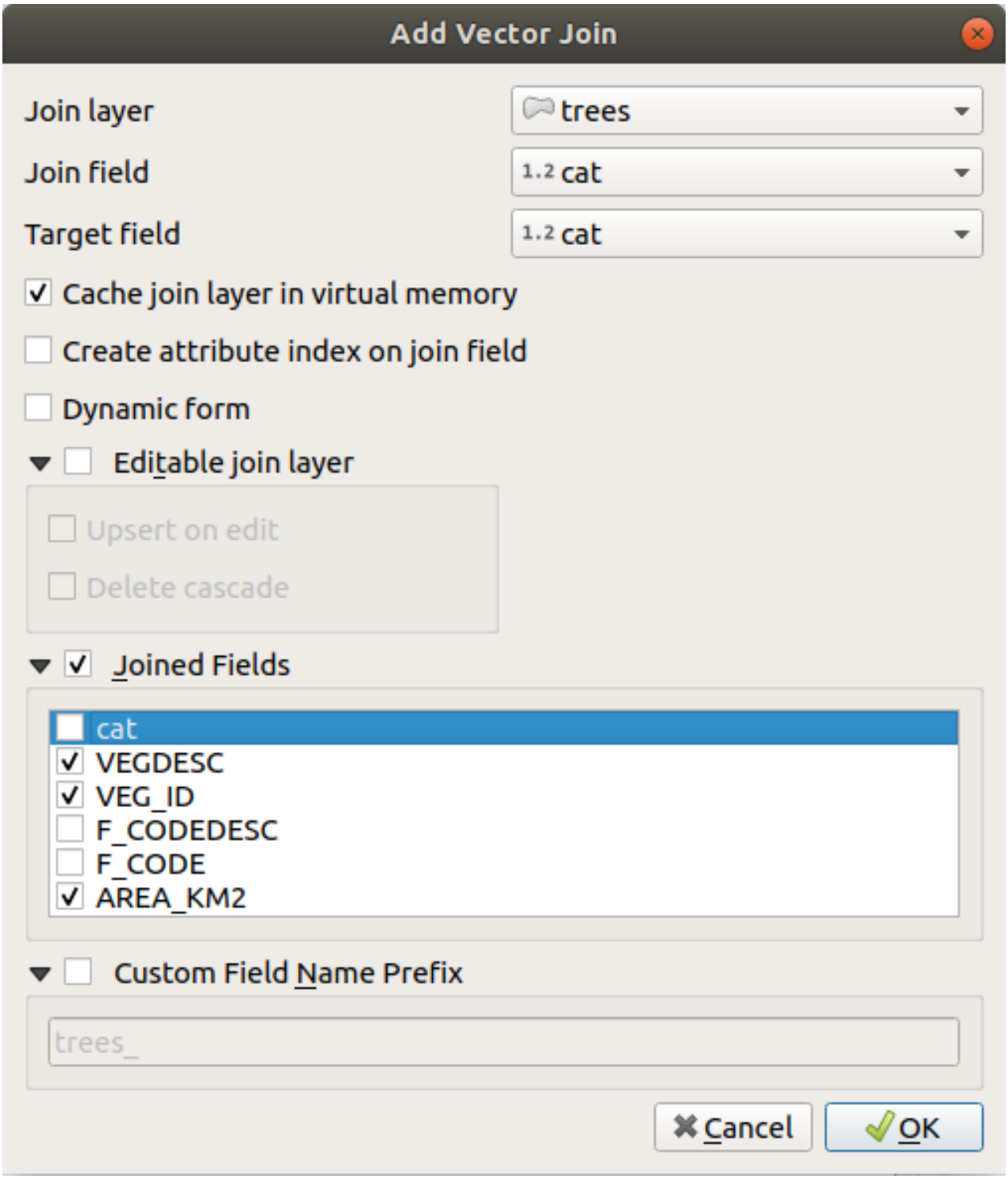


그림 14.45: 기존 벡터 레이어에 속성 테이블 결합

- 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*: 명칭의 충돌을 방지하기 위해 결합된 필드명에 사용자 지정 접두어를 추가합니다.

QGIS 는 현재 OGR(예: CSV, DBF 및 엑셀), 구분 텍스트, 그리고 PostgreSQL 제공자가 지원하는 비공간 테이블 유형들의 결합도 지원합니다.

14.1.11 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:

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:

현재 데이터소스에서 보조 레이어의 환경을 설정하자마자, 탭에서 보조 레이어의 정보를 검색할 수 있습니다:

이 보조 레이어는 다음과 같은 특성을 가집니다:

- 기본 키는 `ID` 이고,
- 보조 필드를 사용하는 피처는 `0` 개이며,
- 보조 필드도 `0` 개입니다.

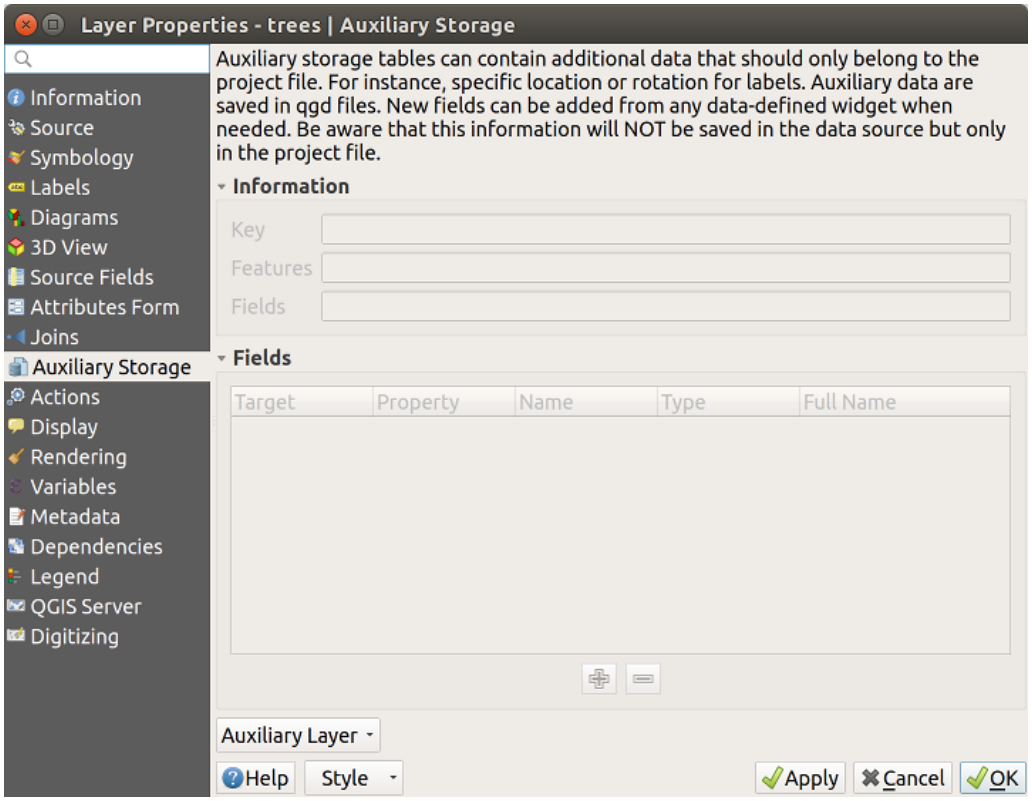


그림 14.46: 보조 저장소 탭

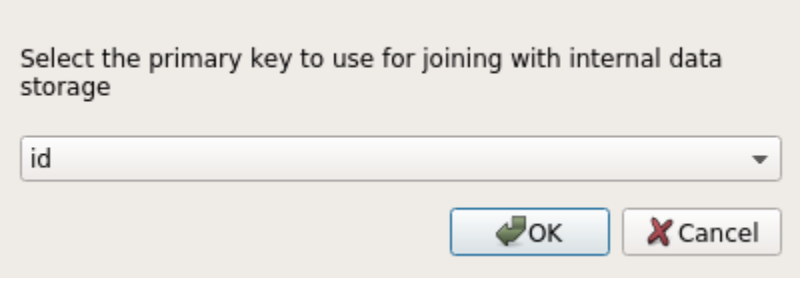


그림 14.47: 보조 레이어 생성 대화창

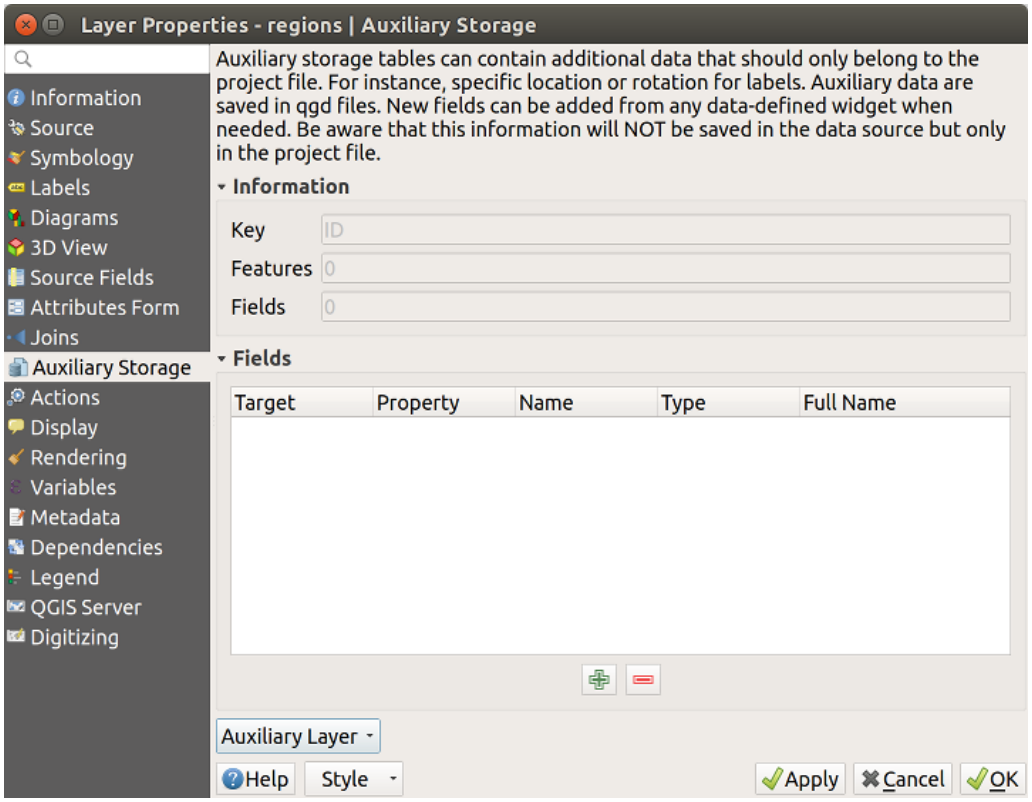


그림 14.48: 보조 레이어 키

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:

앞 그림에서 볼 수 있듯이, 라벨용으로 필드 `21` 개가 자동적으로 생성되고 환경 설정됐습니다. 예를 들면, 기저 SQLite 데이터베이스에서 `FontStyle` 보조 필드의 유형은 `String` 이며 명칭은 `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.

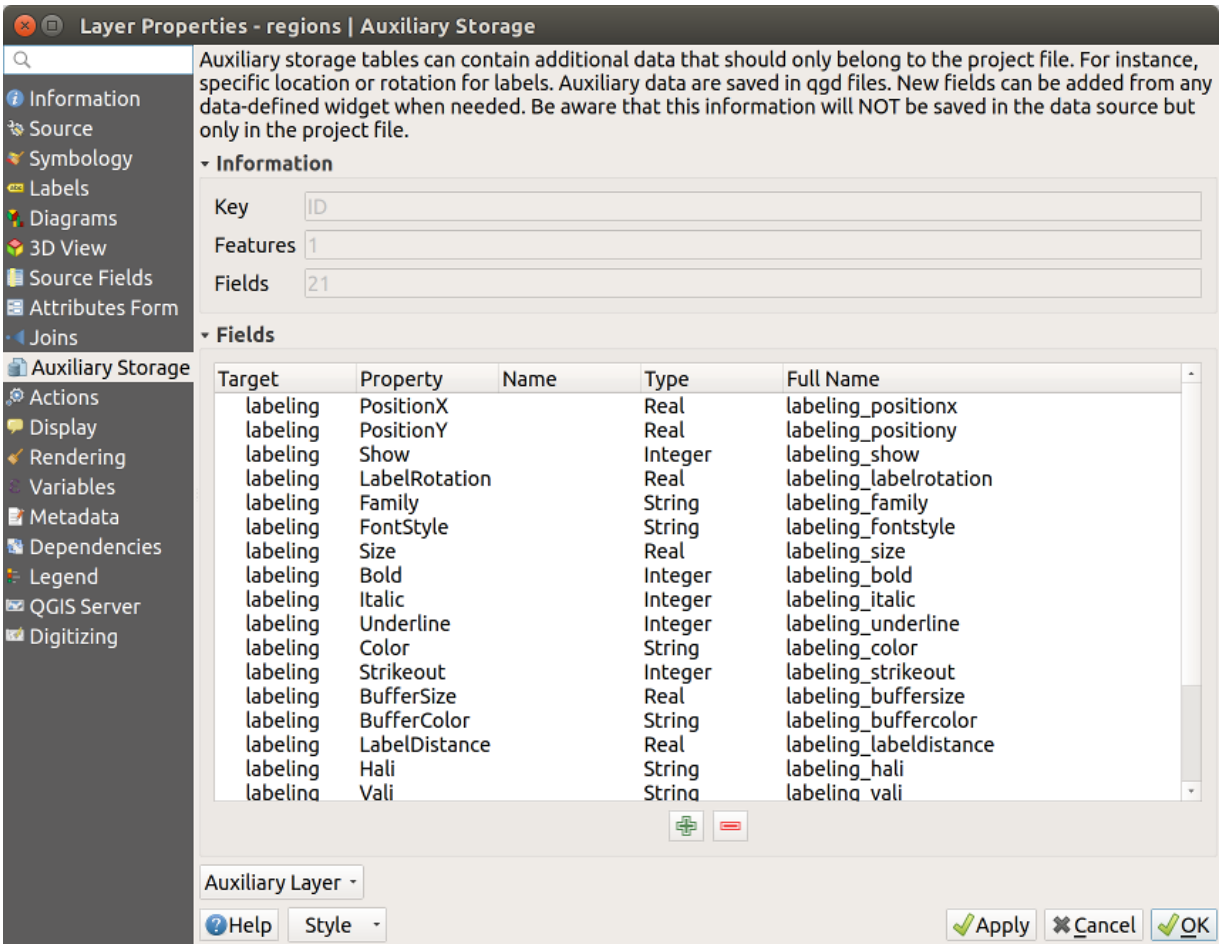

Layer Properties - regions | Auxiliary Storage
Information
Source
Symbology
Labels
Diagrams
3D View
Source Fields
Attributes Form
Joins
Auxiliary Storage
Actions
Display
Rendering
Variables
Metadata
Dependencies
Legend
QGIS Server
Digitizing
Auxiliary storage tables can contain additional data that should only belong to the project file. For instance, specific location or rotation for labels. Auxiliary data are saved in qgd files. New fields can be added from any data-defined widget when needed. Be aware that this information will NOT be saved in the data source but only in the project file.
Information
Key
ID
Features
1
Fields
21
Fields
Target Property Name Type Full Name
labeling PositionX Real labeling_positionx
labeling PositionY Real labeling_positiony
labeling Show Integer labeling_show
labeling LabelRotation Real labeling_labelrotation
labeling Family String labeling_family
labeling FontStyle String labeling_fontstyle
labeling Size Real labeling_size
labeling Bold Integer labeling_bold
labeling Italic Integer labeling_italic
labeling Underline Integer labeling_underline
labeling Color String labeling_color
labeling Strikeout Integer labeling_strikeout
labeling BufferSize Real labeling_buffersize
labeling BufferColor String labeling_buffercolor
labeling LabelDistance Real labeling_labeldistance
labeling Hali String labeling_hali
labeling Vali String labeling_vali
Auxiliary Layer
Help
Style
Apply
Cancel
OK

그림 14.49: 보조 필드

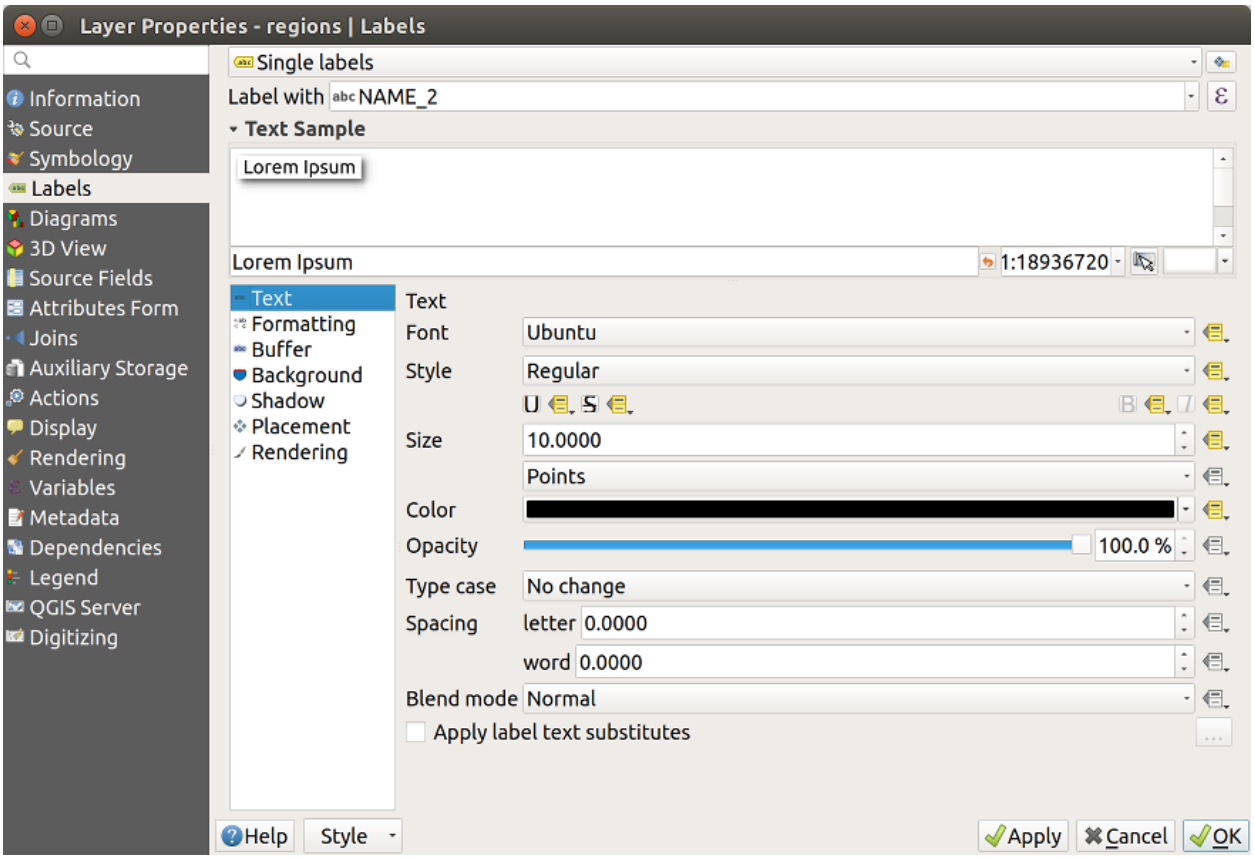


그림 14.50: 자동적으로 생성된 데이터 정의 속성들

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:

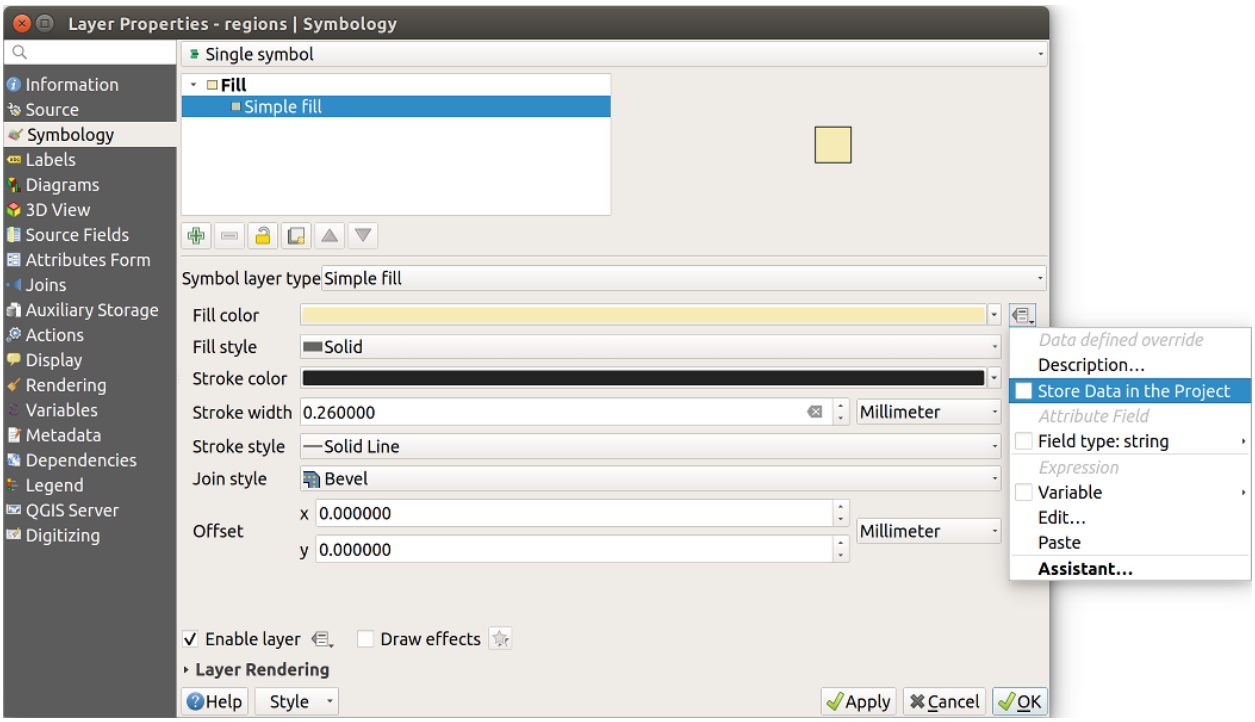


그림 14.51: 심볼을 위한 데이터 정의 속성 메뉴

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 명칭이 필요합니다. *Store data in the project* 를 선택하면 필드의 *Type* 을 표시하고 보조 필드용 유일 명칭을 입력하도록 하는 창이 열립니다. 예를 들면, *Fill color* 보조 필드를 생성할 때 다음과 같은 창이 열립니다:

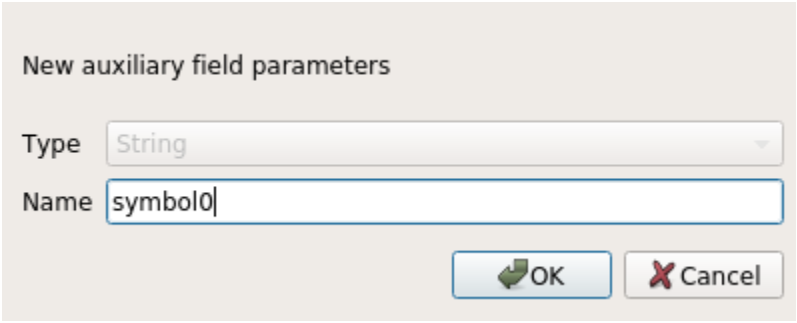


그림 14.52: 심볼을 위한 보조 필드의 명칭

보조 필드를 생성하고 나면, 보조 저장소 탭에서 해당 필드를 검색할 수 있습니다:

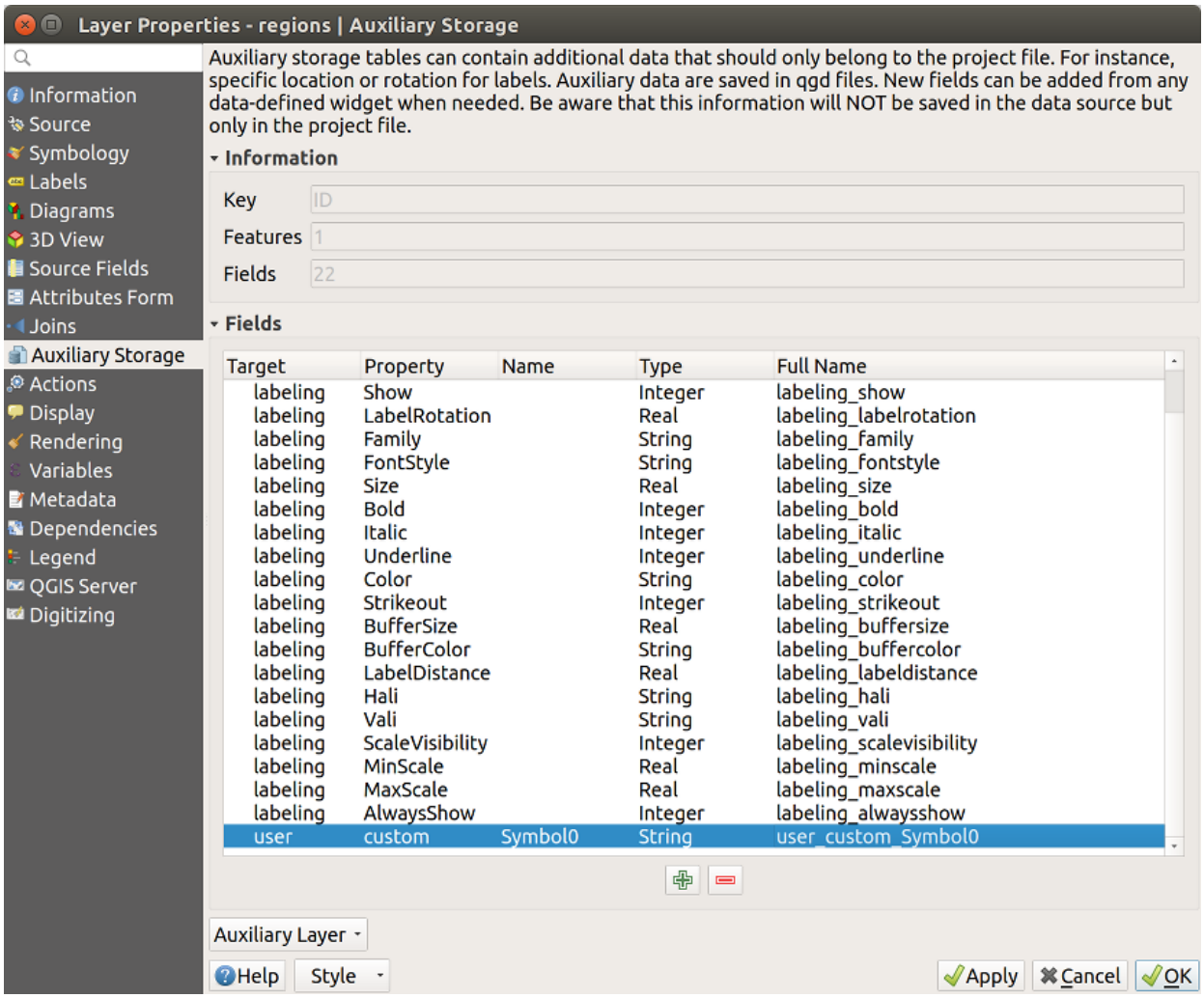


그림 14.53: 보조 필드 심볼

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:

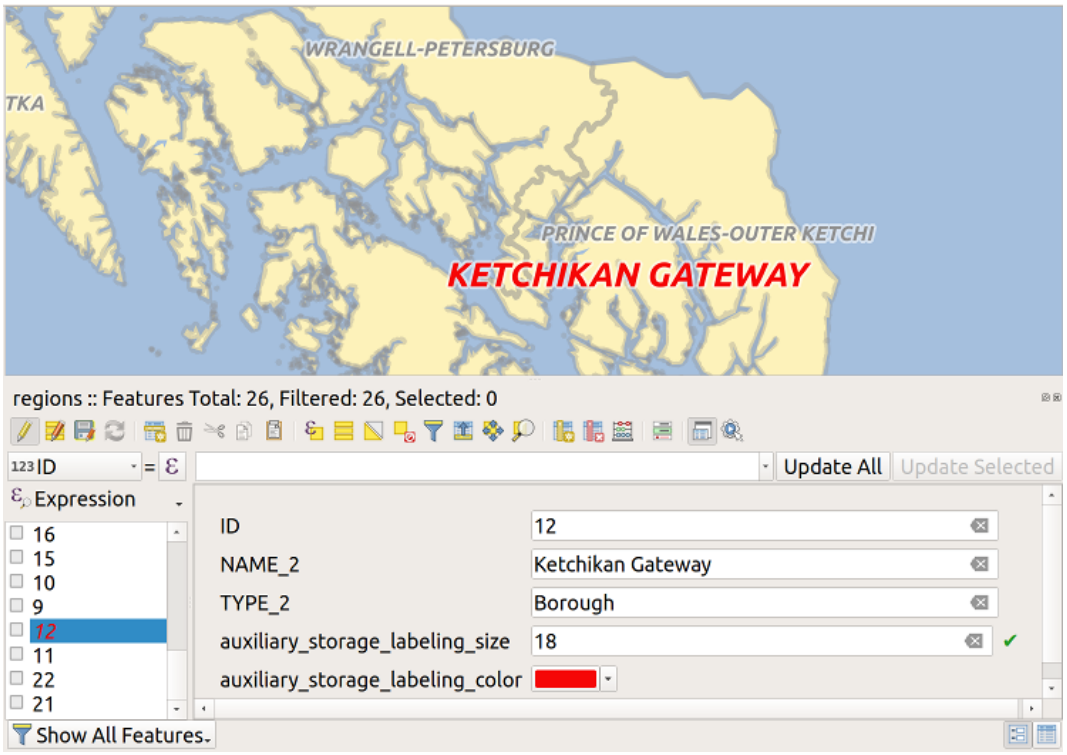


그림 14.54: 보조 필드를 보유한 양식

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.

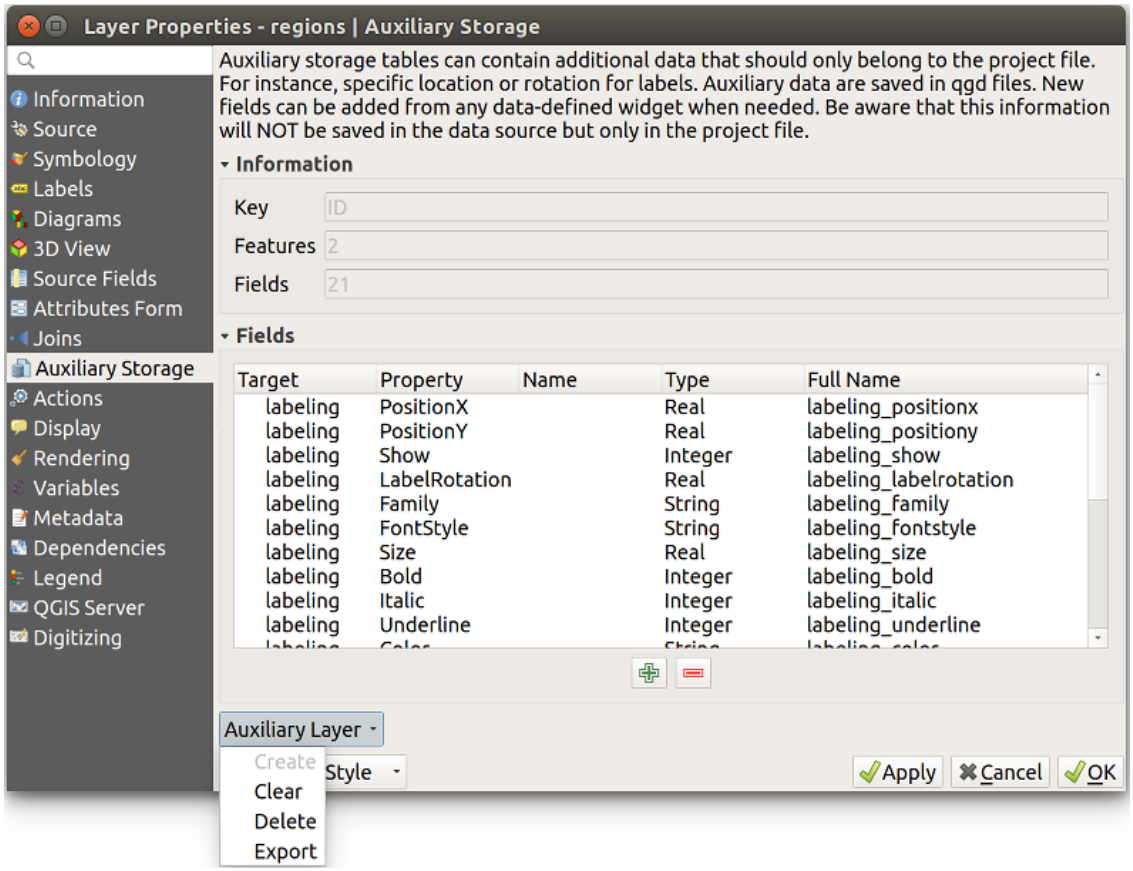


그림 14.55: 보조 레이어 관리

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.

14.1.12 액션 속성



QGIS 는 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.

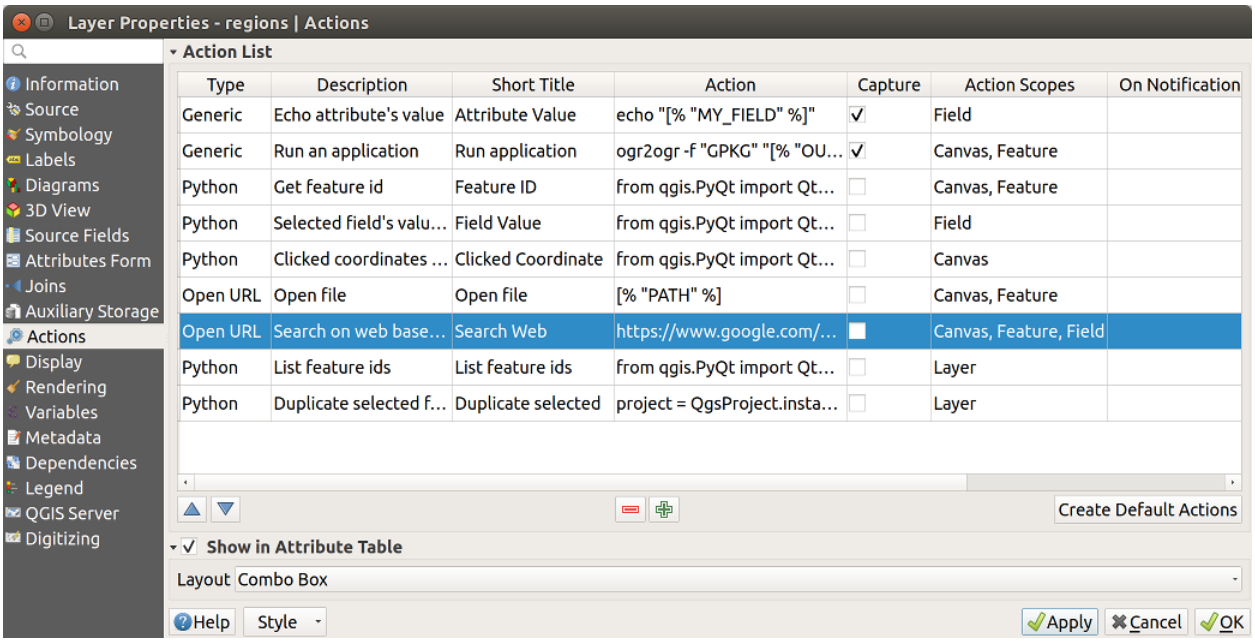


그림 14.56: 몇몇 예시 액션을 담고 있는 오버뷰 액션 대화창

외부 응용 프로그램을 자주 실행하거나 사용자 벡터 레이어에 있는 하나 이상의 값들을 기반으로 웹 페이지를 보려 하는 경우 액션이 유용합니다. 액션은 6 개의 유형으로 나뉘며, 다음과 같이 사용할 수 있습니다:

- 외부 프로세스를 구동하는 일반, 맥, 윈도우 및 유닉스 액션
- 파이썬 표현식을 실행하는 파이썬 액션
- 어디에서나 볼 수 있는 일반 및 파이썬 액션
- 각각 대응하는 플랫폼 상에서만 볼 수 있는 맥, 윈도우 및 유닉스 액션 (예를 들어, 편집기를 여는'편집'액션을 3 개 정의해도 사용자는 편집기를 실행하는 플랫폼의 전용'편집'액션만을 보고 실행할 수 있습니다.)

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 환경 설정하기 참조)

액션 정의하기

속성 액션을 정의하려면, 벡터 *Layer Properties* 대화창을 열고 *Actions* 탭을 선택하십시오. *Actions* 탭에서 Add a new action 을 클릭하면 *Edit Action* 대화창이 열립니다.

Type 옵션으로 액션 유형을 선택하고 액션을 설명하는 명칭을 부여하십시오. 액션 자체의 명칭이 액션을 작동시켰을 때 실행될 응용 프로그램의 명칭을 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. (다음액션 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.

Action Scopes 는 **어디에서** 액션을 사용할 수 있어야 하는지를 정의할 수 있습니다. 다음 4 개 가운데 선택할 수 있습니다:

1. *Feature Scope*: 속성 테이블에 있는 셀을 오른쪽 클릭하면 액션을 사용할 수 있습니다.
2. *Field Scope*: 속성 테이블에 있는 셀, 피처 양식, 그리고 메인 툴바의 기본 액션 버튼을 오른쪽 클릭하면 액션을 사용할 수 있습니다.
3. *Layer Scope*: 속성 테이블 툴바에 있는 액션 버튼으로 액션을 사용할 수 있습니다. 이 액션 유형은 단일 피처가 아니라 레이어 전체를 대상으로 한다는 점을 유의하십시오.
4. *Canvas*: 툴바에 있는 메인 액션 버튼으로 액션을 사용할 수 있습니다.

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 명칭을 `(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.

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* 액션 유형을 선택합니다.
6. 액션명을, 예를 들어 `Google Search` 를 입력합니다.
7. *Short Name*, 또는 *Icon* 도 따로 추가할 수 있습니다.
8. 액션의 *Scope* 를 선택하십시오. 더 자세한 내용은액션 정의하기 를 참조하세요. 이 예시의 경우 기본 설정을 유지합니다.
9. 액션이 실행할 외부 프로그램의 명칭을 지정해야 합니다. 이 경우, 파이어폭스를 이용하겠습니다. 이 프로그램이 사용자 경로에 없을 경우, 전체 경로를 입력해야 합니다.
10. 외부 응용 프로그램 명칭 뒤에, 구글 검색 용 URL `https://www.google.com/search?q=` 를 추가합니다. 검색어는 포함시키지 않습니다.
11. *Action* 란에 있는 텍스트가 이제 `https://www.google.com/search?q=` 여야 합니다.
12. `lakes` 레이어의 필드명을 담고 있는 드롭다운 박스를 클릭합니다. *Insert* 버튼 바로 왼쪽에 있습니다.

13. 드롭다운 박스에서'NAMES'를 선택한 다음 *Insert* 를 클릭합니다.
14. 이제 사용자 액션 텍스트가 다음과 같이 보일 겁니다:

    ```
    https://www.google.com/search?q=[%NAMES%]
    ```

15. 액션 생성을 완료하고 추가하려면, *OK* 버튼을 클릭합니다.

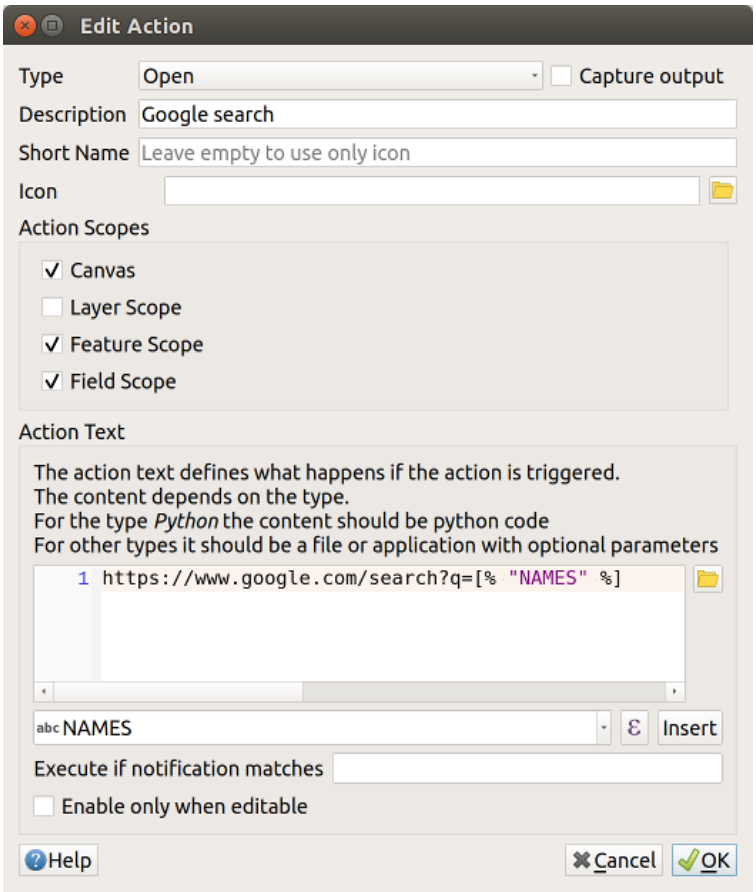


그림 14.57: 예시 액션 환경을 설정한 액션 편집 대화창

이렇게 액션을 완성했습니다. 이제 사용할 준비가 끝났습니다. 최종 액션 텍스트는 다음과 같아야 합니다:

```
https://www.google.com/search?q=[%NAMES%]
```

이제 액션을 사용할 수 있습니다. *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, 파이어폭스를 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

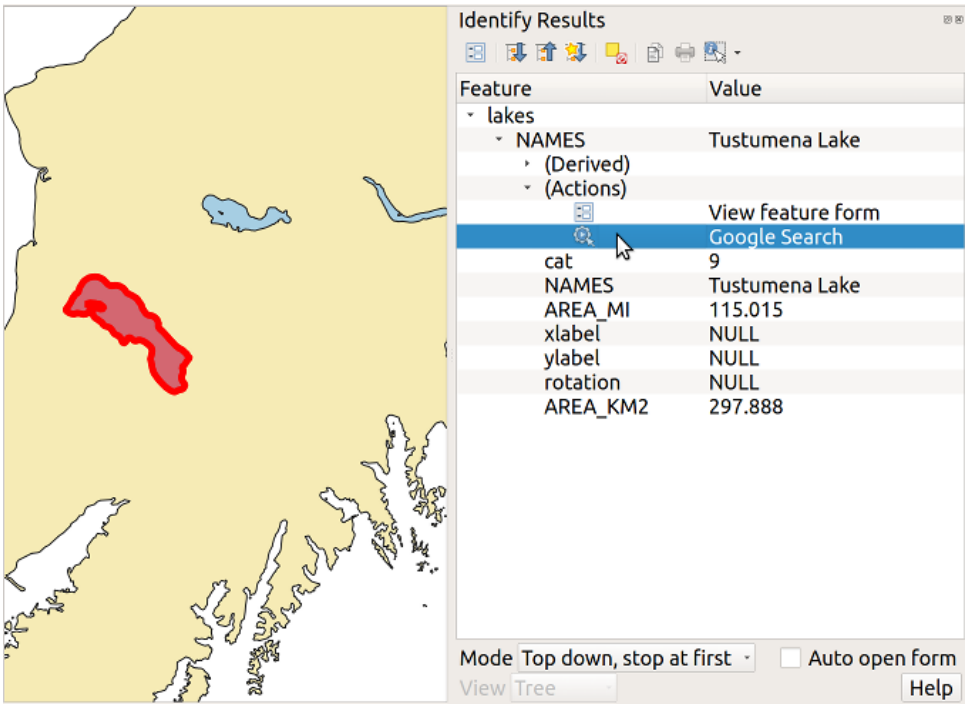


그림 14.58: 객체 선택 및 액션 선택

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 (Shapefile 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'
  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
```

(다음 페이지에 계속)

(이전 페이지에서 계속)

```
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 명칭은 데이터를 기반으로 합니다. (*filename* 및 *layername* 은 액션이 작동한 벡터의 속성 테이블에 있는 열들의 명칭입니다.)

```
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" %]')
```

14.1.13 표시 속성

Display 탭에서 피처 식별에 사용되는 필드의 환경을 설정할 수 있습니다:

- *Display name*: 필드 또는표현식 을 기반으로 하는 표시명입니다:
  - 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 GeoPDF 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
- *HTML Map Tip* 은 맵 도움말을 위해 특별히 생성됩니다. 이 도움말은 필드, 표현식, 그리고 HTML 태그 (여러 줄, 글꼴, 이미지, 하이퍼링크 등등) 을 섞은 더 복잡하고 풍성한 HTML 텍스트입니다.

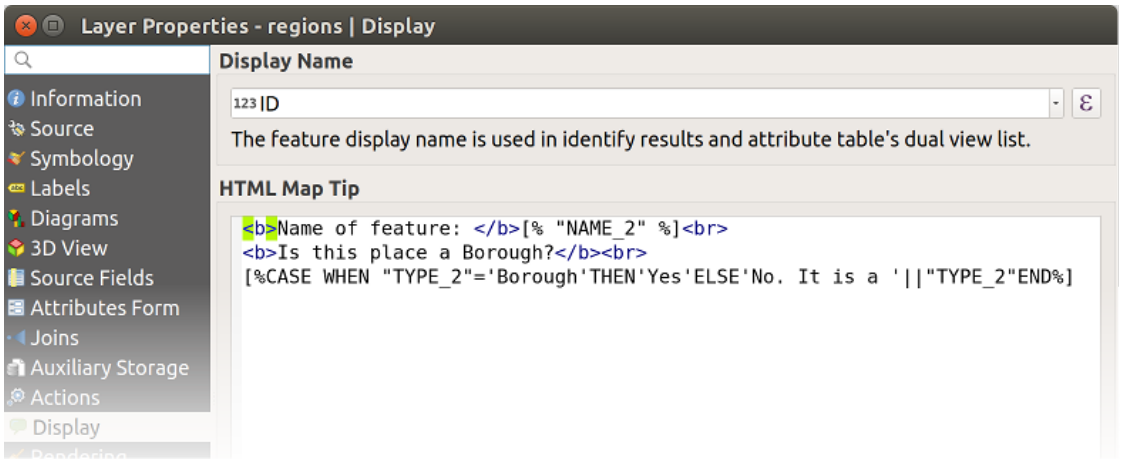


그림 14.59: 맵 도움말 용 HTML 코드

맵 도움말을 활성화하려면, *View* ▷ *Show Map Tips* 메뉴 옵션을 선택하거나 *Attributes Toolbar* 에 있는 Show Map Tips 아이콘을 클릭하십시오. 맵 도움말은 교차 세션 (cross-session) 기능으로, 한 번 활성화시키면 다시 비활성화시키기 전까지는 계속 활성화 상태로 — 심지어 QGIS 를 종료하고 다시 시작 (새 세션) 해도 — 모든 프로젝트의 모든 레이어에 적용된다는 뜻입니다.

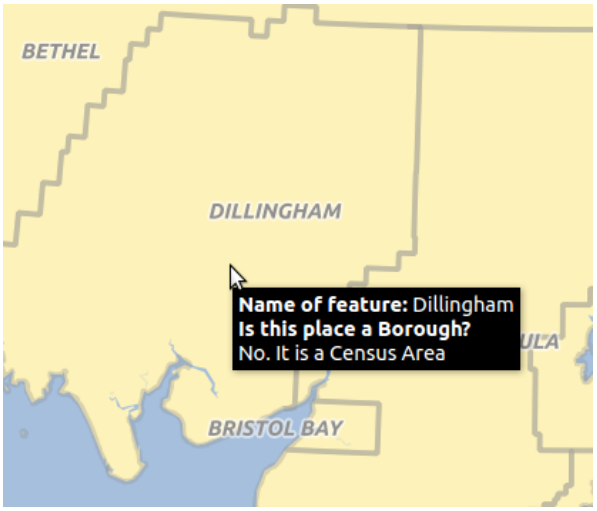


그림 14.60: HTML 코드로 생성된 맵 도움말

14.1.14 렌더링 속성

축척에 따른 가시성

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 내용은축척에 따른 렌더링 을 참조하세요.

도형 단순화

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.

극도로 세밀한 레이어 (예: 수많은 노드를 지닌 폴리곤 레이어) 를 렌더링하는 경우, 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.

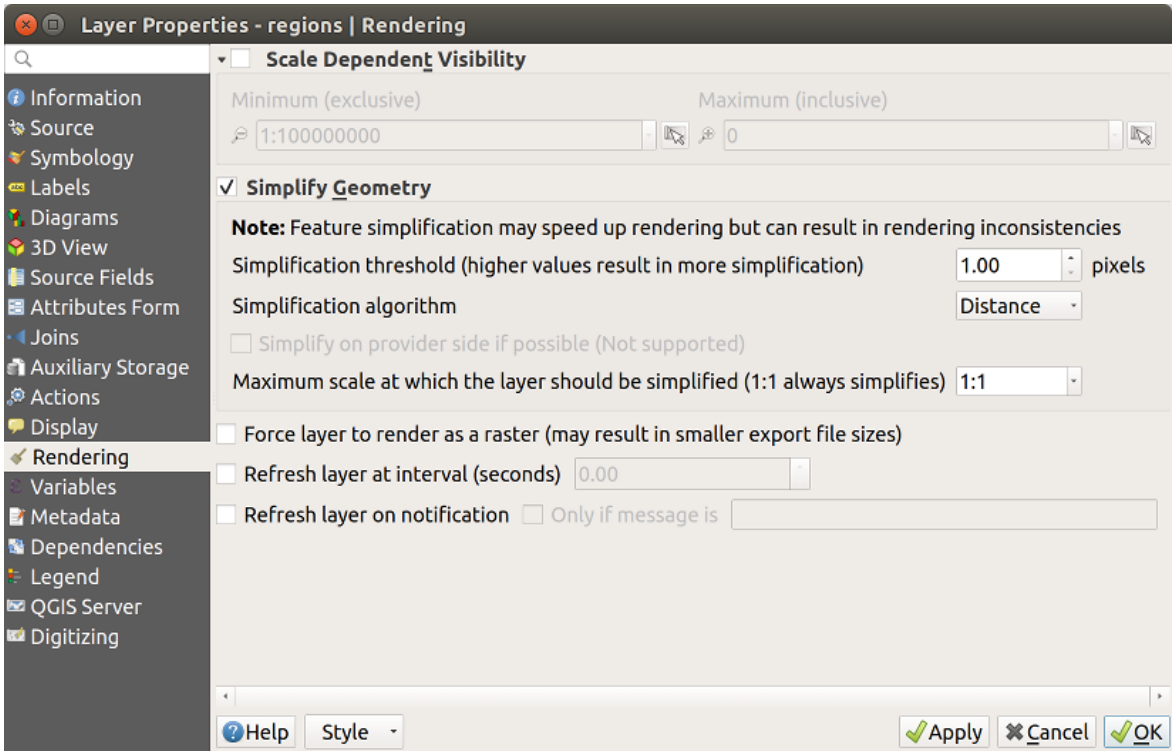


그림 14.61: 레이어 도형 단순화 대화창

Refresh layer at interval (seconds): 개별 레이어들을 자동으로, 일치하는 시간 간격으로 새로고침하도록 설정합니다. 하나 이상의 레이어에 자동 업데이트 간격을 설정한 경우 여러번 새로고침하지 않도록 캔버스 업데이트를 연기합니다.

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.

14.1.15 변수 속성

ε *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.

14.1.16 메타데이터 속성

 *Metadata* 탭은 사용자 레이어에 대한 메타데이터 보고서를 생성하고 편집할 수 있는 옵션들을 제공합니다. 채워야 할 정보는 다음과 같습니다:

- 데이터 *Identification*: 데이터의 기본 속성 (페어런트, 식별자, 타이틀, 요약, 언어 등등)
- 데이터가 속한 *Categories*: **ISO** 범주들과 함께 사용자 지정 범주도 추가할 수 있습니다.
- *Keywords*: 데이터 및 관련 개념을 표준 어휘를 따라 검색하기 위한 키워드
- *Access*: 데이터셋에 접근하기 위한 정보 (라이선스, 권한, 비용 및 제약조건)
- *Extent*: 데이터셋의 공간 (좌표계, 맵 범위, 고도) 또는 시계열 범위
- *Contact*: 데이터셋 소유자 (들) 의 정보
- *Links*: 보조 리소스 및 관련 정보를 가리키는 링크
- *History*: 데이터셋의 이력

Validation 탭에서 이렇게 채워진 정보를 요약해서 제공하며 양식과 관련된 잠재적인 문제점들을 식별할 수 있습니다. 그 다음 문제점을 해결하거나 무시할 수 있습니다.

메타데이터는 현재 프로젝트 파일에 저장됩니다. 메타데이터를 파일 기반 레이어와 함께 `.qmd` 파일로 저장할 수도 있고, 또는 원격 레이어 (예: PostGIS) 의 경우 로컬 `.sqlite` 데이터베이스에 저장할 수도 있습니다.

14.1.17 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.

14.1.18 범례 속성

 *Legend* 속성 탭은레이어 패널 그리고/또는인쇄 조판기 범례 를 위한, 다음과 같은 옵션을 포함하는 고급 설정을 제공합니다:

-  *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.
- 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.

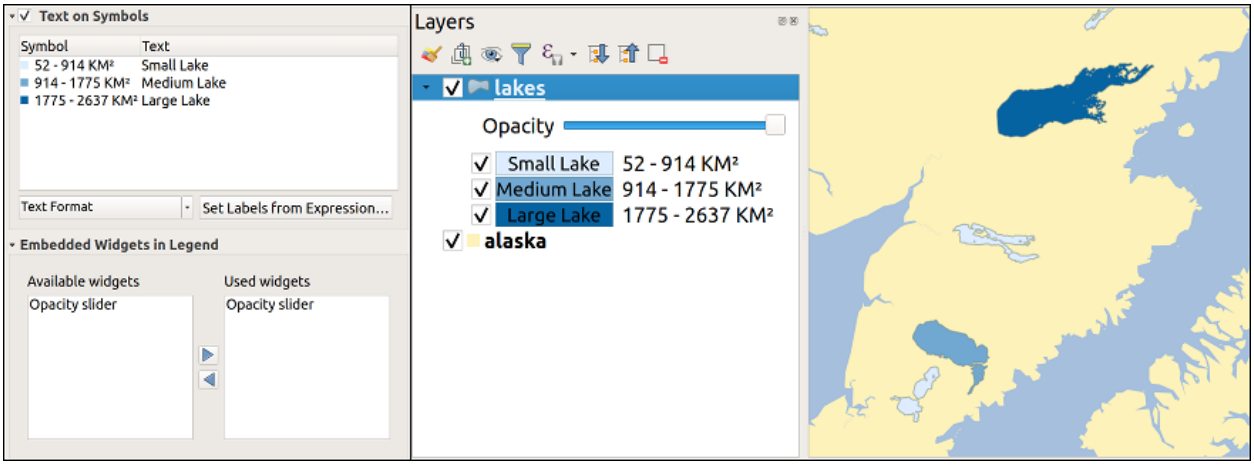


그림 14.62: 심볼 위에 텍스트 설정 (좌), *Layers* 패널에서 텍스트 렌더링 설정 (우)

14.1.19 QGIS 서버 속성

QGIS Server 탭은 *Description*, *Attribution*, *MetadataURL*, 그리고 *LegendUrl* 부분으로 이루어져 있습니다.

Description 부분에서 요청에 있는 레이어를 참조하는 데 사용된 *Short name* 을 변경할 수 있습니다. (단축명에 대해 더 알고 싶다면, server_short_name 을 읽어보십시오.) 여기에서 레이어에 대한 *Title* 및 *Abstract* 를 추가하고 편집할 수 있고, *Keyword list* 를 정의할 수도 있습니다. 이 키워드 목록은 메타데이터 카탈로그에서 이용할 수 있습니다. XML 메타데이터 파일에서 제목을 사용하려 하는 경우, *DataUrl* 란에 링크를 입력해야 합니다.

XML 메타데이터 카탈로그에서 속성 데이터를 얻으려면 *Attribution* 부분을 이용하십시오.

MetadataUrl 부분에서 XML 메타데이터 카탈로그를 가리키는 일반 경로를 정의할 수 있습니다. 이 정보는 향후 세션을 위해 QGIS 프로젝트 파일에 저장되며, QGIS 서버에서 사용될 것입니다.

LegendUrl 부분에서는 *URL* 란에 범례 이미지의 URL 을 입력할 수 있습니다. *Format* 드롭다운 옵션을 이용해서 해당 이미지의 적절한 포맷을 적용시키십시오. 현재 PNG, JPG 및 JPEG 이미지 포맷을 지원하고 있습니다.

QGIS 서버에 대해 더 알고 싶다면, QGIS-Server-manual 을 읽어보십시오.

14.1.20 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.

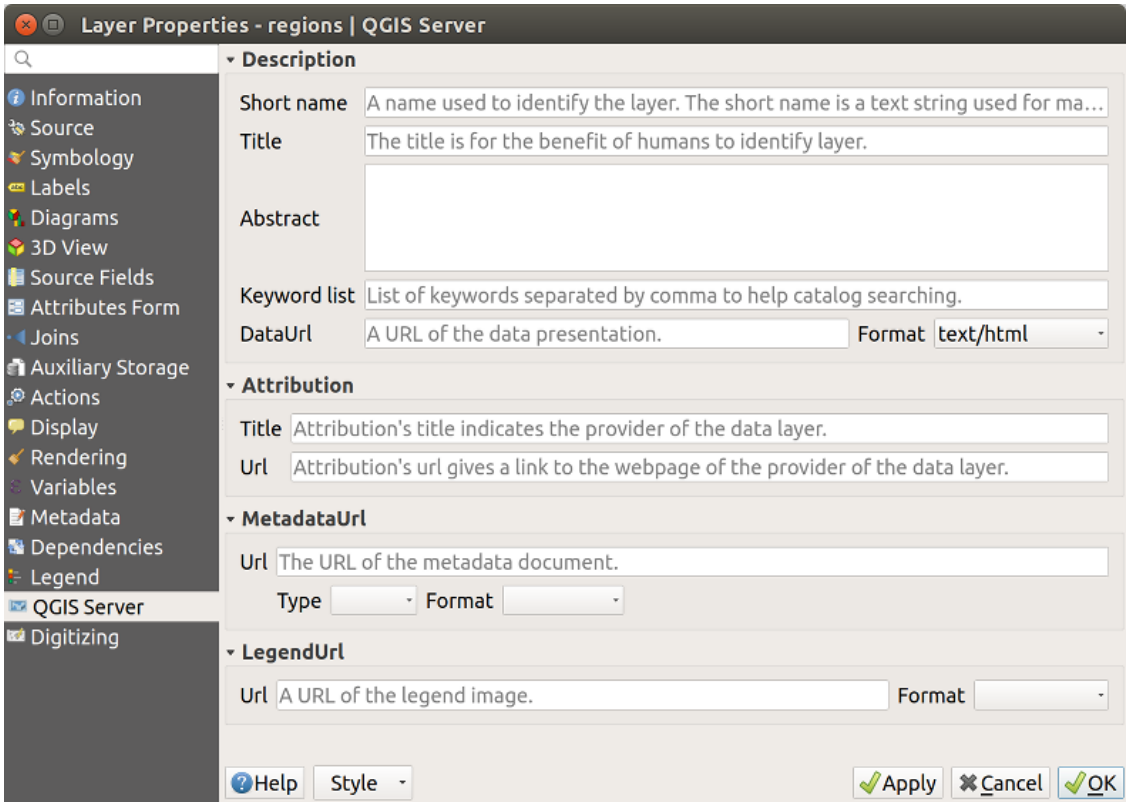


그림 14.63: 벡터 레이어 속성 대화창의 QGIS 서버 탭

도형 점검

Geometry checks 부분에서 도형별 기반으로 추가적인 유효성 확인 작업을 활성화시킬 수 있습니다. 어떤 도형이든 수정 직후 이 점검을 통과하지 못하는 경우 도형 유효성 확인 패널에 보고 메시지를 표시합니다. 점검을 통과하지 못하는 한, 레이어를 저장할 수는 없습니다. *Is valid* 옵션을 체크하면 도형에 대해 자체 교차 같은 기본 유효성 점검을 실행할 것입니다.

위상 점검

Topology checks 부분에서 추가적인 위상 유효성 점검을 활성화시킬 수 있습니다. 사용자가 레이어를 저장할 때 위상 점검을 실행할 것입니다. 도형 유효성 확인 패널에 점검 오류 보고 메시지를 표시할 것입니다. 유효성 오류가 존재하는 한, 레이어를 저장할 수는 없습니다. 위상 점검은 수정한 피처의 경계상자 영역 안에서 실행됩니다. 동일한 영역에 다른 피처가 존재할 수도 있기 때문에, 현재 편집 세션에서 발생한 오류는 물론 이런 피처와 관련된 위상적 오류도 보고 메시지에 포함됩니다.

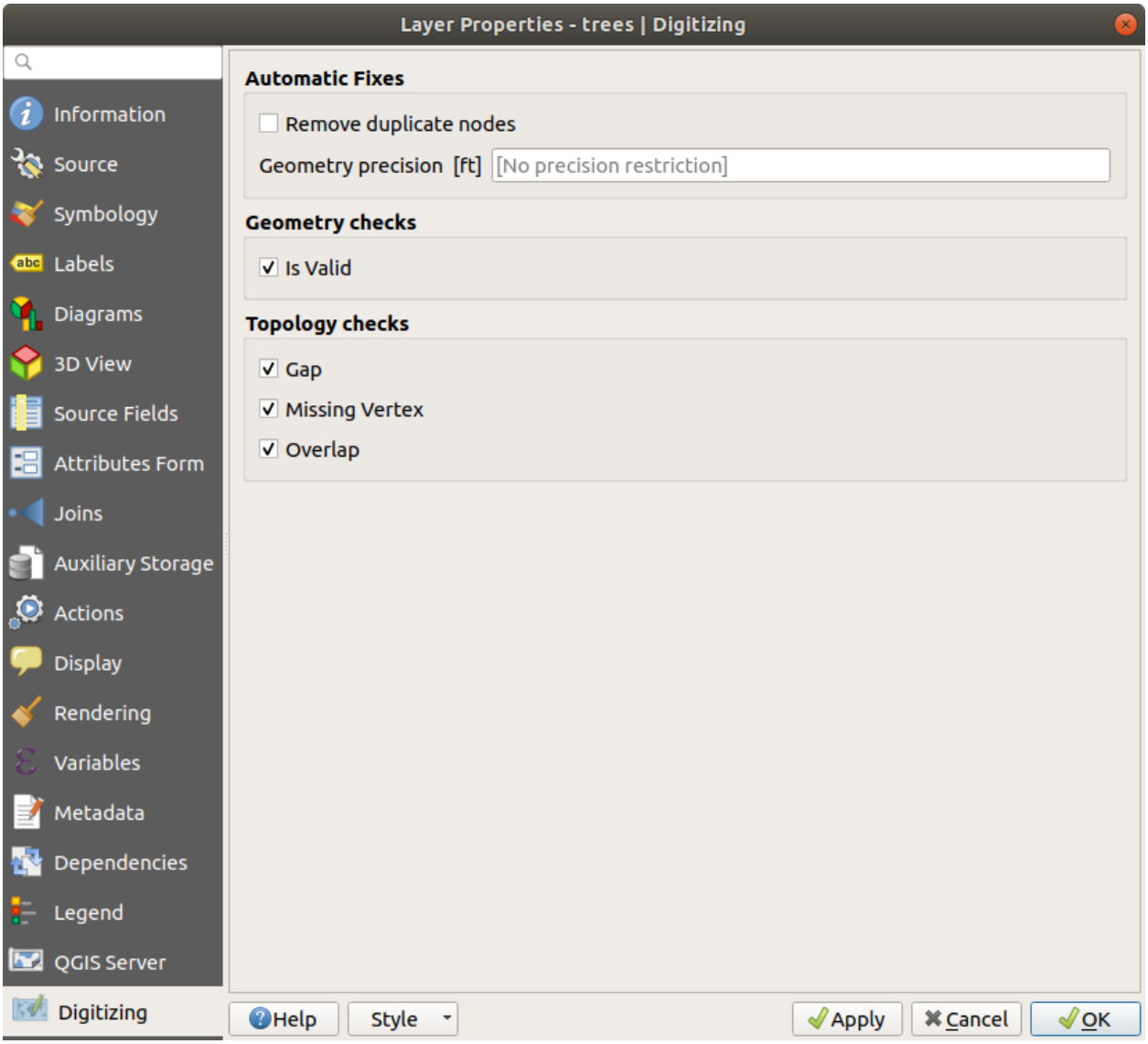
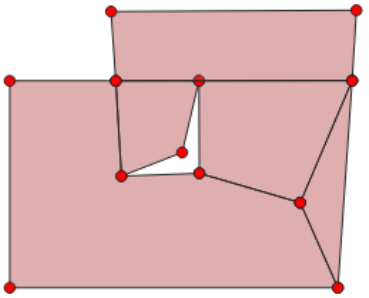
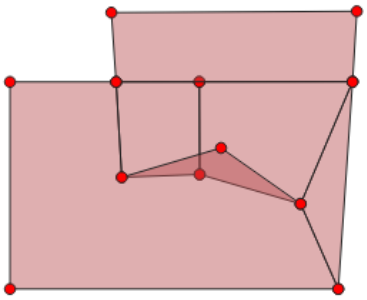
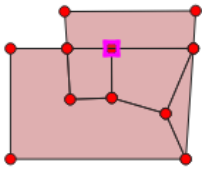


그림 14.64: 벡터 레이어 속성 대화창의 QGIS 디지타이즈 작업 탭

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 유효성 확인 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.

14.2 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.

14.2.1 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:

- 표현식 탭: 사전 정의 함수 목록을 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:

- 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 유효하지 않은 인수 (argument) 를 표시합니다.
  - 마커 *(Marker)*: 단일 위치에 있는 다른 모든 오류를 (예를 들어 빠진 괄호, 예상 밖의 문자 등을) 표시합니다.

  팁: 사용자 표현식을 코멘트로 문서화

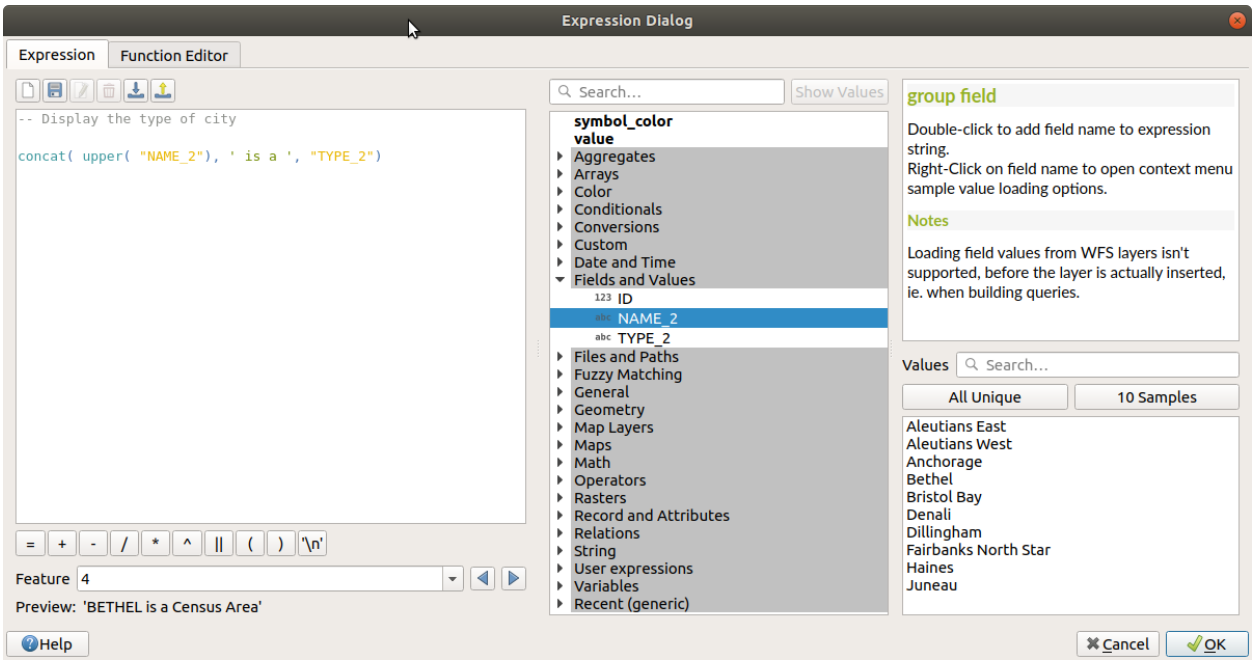


그림 14.65: 표현식 탭

복잡 표현식을 사용할 때, 사용자가 기억하기 쉽도록 텍스트를 여러 줄 또는 인라인 (inline) 코멘트 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.
  - 표현식의 실시간 *Output preview* 를 볼 수 있습니다. 기본적으로 레이어의 첫 번째 피처를 대상으로 미리보기를 구합니다. *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 아마도 *select* 피처 *from* 레이어 *where* 조건 또는 *update* 레이어 *set* 필드 = 새 _ 값 *where* 조건 같은 유형의 쿼리들을 알고 있을 것입니다. QGIS 표현식도 이런 모든 정보를 필요로 하지만, 사용자가 표현식 작성기 대화창을 열기 위해 사용하는 도구가 일부 정보를 제공합니다. 예를 들어 (높이) 필드를 가진 (건물) 레이어가 있다면:
   - 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 피처를 선택하고, Field calculator 버튼을 누른 다음 *Update existing field* 로"height"를 선택하면 이미"update buildings set height = ??? where height > 20"이라는 명령어를 입력한 것입니다. 이 경우

이제 사용자가 입력해야 하는 것은 **새 값** 뿐입니다. 예를 들어 이미 선택한 건물들의 높이를 50 으로 설정하려면 `50` 을 입력하기만 하면 됩니다.

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.

- 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 명칭을 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 누르면, 사용자가 손쉽게 접근할 수 있도록 하려는 중요한 표현식을 저장할 수 있습니다. 이렇게 저장한 표현식들은 가운데 패널에 있는 **User expressions** 그룹에서 사용할 수 있습니다. 사용자 표현식은 사용자 프로파일 (`<userprofile>/QGIS/QGIS3.ini` 파일) 에 저장되며 현재 사용자 프로파일의 모든 프로젝트에서의 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.

14.2.2 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.

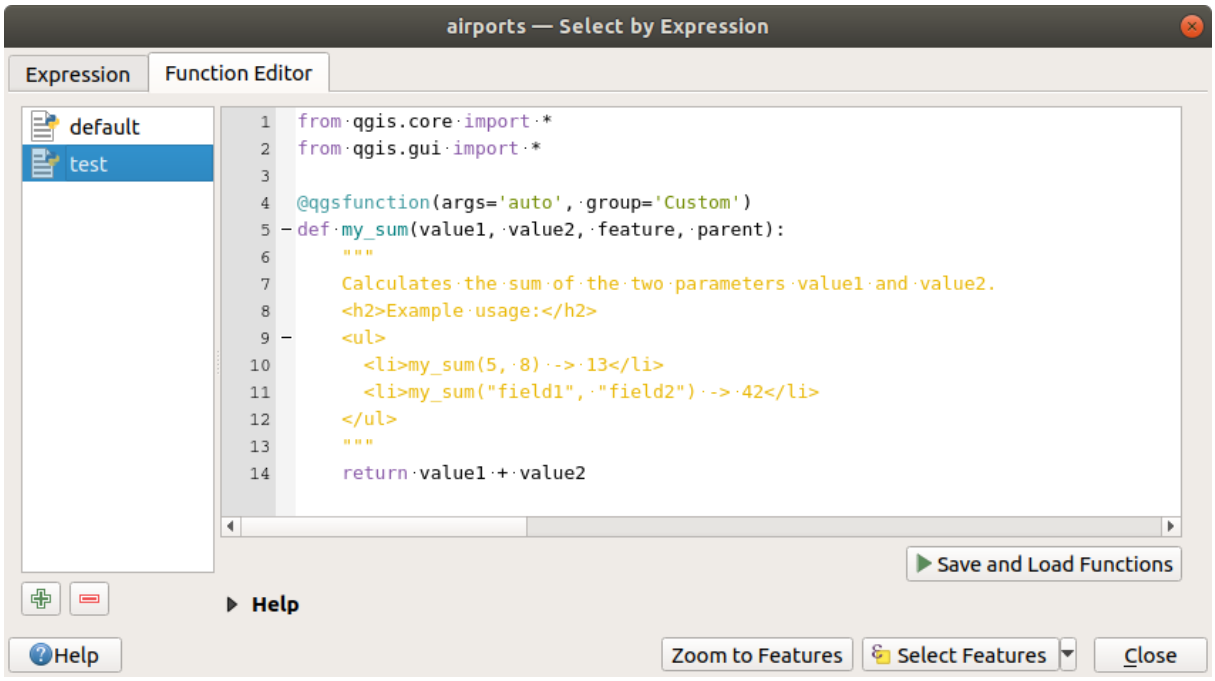


그림 14.66: 함수 편집기 탭

새 함수를 생성하려면:

1. New File 버튼을 누르십시오.
2. 팝업 양식에 사용할 명칭을 입력하고 *OK* 를 누르십시오.

   사용자가 지정한 명칭을 가진 새 항목이 *Function Editor* 탭의 왼쪽 패널에 추가됩니다. 이 항목은 QGIS 템플릿 파일 기반 파이썬 `.py` 파일로, 활성화된사용자 프로파일 디렉터리 아래 있는 `/python/expressions` 폴더에 저장됩니다.
3. 파일의 내용 — 파이썬 스크립트 템플릿 — 은 오른쪽 패널에 표시됩니다. 사용자의 필요에 따라 코드 및 도움말을 업데이트하십시오.
4. *Save and Load Functions* 버튼을 누르십시오. *Expression* 탭의 함수 트리에 사용자가 작성한 함수를 추가합니다. 기본적으로 `Custom` 그룹 아래 추가될 것입니다.
5. 사용자의 새 함수를 즐겨보세요.
6. 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 파이썬 함수는 사용자 프로파일 디렉터리 아래 저장됩니다. 즉 QGIS 를 시작할 때마다, 현재 사용자 프로파일에서 정의된 모든 함수를 자동적으로 불러올 것입니다. `/python/expressions` 폴더에만 새 함수를 저장하고 프로젝트 파일에는 저장하지 않는다는 사실을 알아두십시오. 사용자 지정 함수를 이용하는 프로젝트를 공유하는 경우, `/python/expressions` 폴더에 있는 `.py` 파일도 함께 공유해야 합니다.

사용자 지정 함수를 삭제하려면:

1. *Function Editor* 탭을 활성화하고

2. 목록에서 함수를 선택한 다음

3. Remove selected function 버튼을 누르십시오. 목록에서 선택한 함수를 제거하고, 사용자 프로파일 폴더에서 해당 함수에 대응하는 `.py` 파일도 삭제합니다.

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(args='auto', group='Custom')
def my_sum(value1, value2, feature, parent):
    """
    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
```

`args='auto'` 함수 인자를 사용하는 경우 필수 함수 인자의 개수는 파이썬으로 정의된 함수의 인자 개수 (에서 `feature` 와 `parent` 2 를 뺀 개수) 로 계산될 것입니다. `group='Custom'` 인자는 표현식 대화창에서 이 함수가 포함될 목록을 나타냅니다.

다음과 같은 키워드 인자를 추가할 수도 있습니다:

- 표현식이 피처 도형에 접근해야 하는 경우 `usesgeometry=True` 인자를 추가하면 됩니다. 기본값은 `False` 입니다.
- 표현식이 NULL 값에 대해 사용자가 정의한 처리를 하는 경우 `handlesnull=True` 인자를 추가하면 됩니다. 기본값인 `False` 를 사용한다면, 파라미터 가운데 하나라도 NULL 인 경우 항상 NULL 을 산출할 것입니다.
- `referenced_columns=[list]`: 함수가 요구하는 속성명 배열입니다. 기본값은 `[QgsFeatureRequest.ALL_ATTRIBUTES]` 입니다.

이제 이렇게 정의한 예시 함수를 표현식 탭에서 사용할 수 있습니다:

PyQGIS-Developer-Cookbook 에서 파이썬 코드 생성 관련 심화 정보를 찾아볼 수 있습니다.

14.3 함수 목록

QGIS 에서 사용할 수 있는 함수, 연산자 및 변수의 목록을 다음에 범주별로 정리했습니다.

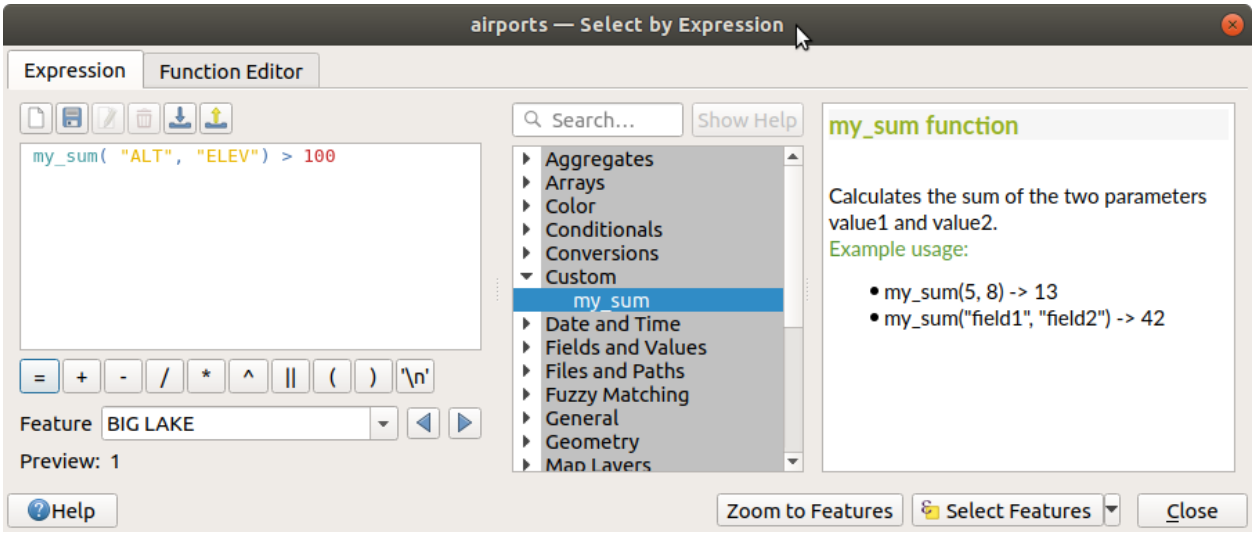


그림 14.67: 표현식 탭에 추가된 사용자 지정 함수

14.3.1 집계 함수

이 그룹은 레이어 및 필드에 있는 값들을 집계하는 함수를 담고 있습니다.

- *aggregate*
- *array_agg*
- *collect*
- *concatenate*
- *concatenate_unique*
- *count*
- *count_distinct*
- *count_missing*
- *iqr*
- *majority*
- *max_length*
- *maximum*
- *mean*
- *median*
- *min_length*
- *minimum*
- *minority*
- *q1*
- *q3*

- *range*
- *relation_aggregate*
- *stdev*
- *sum*

aggregate

다른 레이어의 피처를 이용해서 계산한 집계값을 반환합니다.

문법	aggregate(layer, aggregate, expression, [filter], [concatenator=”], [order_by]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **layer** - 레이어명 또는 레이어 ID 가운데 하나를 나타내는 문자열 • **aggregate** - 계산할 집계에 대응하는 문자열. 유효한 옵션은 다음과 같습니다: – count – count_distinct – count_missing – min – max – sum – mean – median – stdev – stdevsample – range – minority – majority – q1: 제 1 사분위수 (First Quartile) – q3: 제 3 사분위수 (Third Quartile) – iqr: 사분위간 범위 (Inter Quartile Range) – min_length: 최단 문자열 길이 – max_length: 최장 문자열 길이 – concatenate: 연결자 (concatenator) 로 문자열을 결합 – concatenate_unique: 연결자로 유일 문자열을 결합 – collect: 집계된 다중 부분 도형을 생성 – array_agg: 집계된 값의 배열을 생성 • **expression** - 집계할 하위 표현식 또는 필드명 • **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. • **concatenator** - ‘연쇄 (concatenate)’집계를 위한 값을 결합하는 데 사용할 부가적인 문자열 • **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 • `aggregate('rail_stations','sum', "passengers"/7)` → 값들의 합을 구하기 전에“passengers”필드를 7 로 나누어“passengers”의 일일 평균을 계산 • `aggregate(layer:='rail_stations',aggregate:='sum', expression:="passengers",filter:="class">3)` → “class”속성이 3 보다 큰 객체에서만“passengers”필드의 모든 값에 대한 합계 • `aggregate(layer:='rail_stations',aggregate:='concatenate', expression:="name", concatenator:=',')` → rail_stations 레이어에서 모든 객체에 대한 쉼표로 구분된 name 필드의 값 목록 • `aggregate(layer:='countries', aggregate:='max', expression:="code", filter:=intersects( $geometry, geometry(@parent) ) )` → ‘countries’레이어에서 교차하는 country 의 country 코드 • `aggregate(layer:='rail_stations',aggregate:='sum', expression:="passengers",filter:=contains( @atlas_geometry, $geometry ) )` → 현재 지도책 객체 안에 있는 rail_stations 레이어의 passengers 필드에서 나온 모든 값의 합계 • `aggregate(layer:='rail_stations', aggregate:='collect', expression:=centroid($geometry), filter:=region_name = attribute(@parent,'name') )` → 현재 객체와 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]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **expression** - 집계할 필드의 하위 표현식 • **group_by** - 그룹 집계 계산에 사용할 부가적인 표현식 • **filter** - 집계하는 데 사용되는 객체를 필터링하는 데 사용할 부가적인 표현식 • **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]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **expression** - 집계할 도형 표현식 • **group_by** - 그룹 집계 계산에 사용할 부가적인 표현식 • **filter** - 집계하는 데 사용되는 객체를 필터링하는 데 사용할 부가적인 표현식
예제	• `collect( $geometry )` → 집계된 도형들로 이루어진 다중 부분 도형 • `collect( centroid($geometry), group_by:="region", filter:="use" = 'civilian' )` → 피처의 지역 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]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **expression** - 집계할 필드의 하위 표현식 • **group_by** - 그룹 집계 계산에 사용할 부가적인 표현식 • **filter** - 집계하는 데 사용되는 객체를 필터링하는 데 사용할 부가적인 표현식 • **concatenator** - 값을 결합하는 데 사용할 부가적인 문자열. 기본적으로 비어 있습니다. • **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]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **expression** - 집계할 필드의 하위 표현식 • **group_by** - 그룹 집계 계산에 사용할 부가적인 표현식 • **filter** - 집계하는 데 사용되는 객체를 필터링하는 데 사용할 부가적인 표현식 • **concatenator** - 값을 결합하는 데 사용할 부가적인 문자열. 기본적으로 비어 있습니다. • **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]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **expression** - 집계할 필드의 하위 표현식 • **group_by** - 그룹 집계 계산에 사용할 부가적인 표현식 • **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]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **expression** - 집계할 필드의 하위 표현식 • **group_by** - 그룹 집계 계산에 사용할 부가적인 표현식 • **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]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **expression** - 집계할 필드의 하위 표현식 • **group_by** - 그룹 집계 계산에 사용할 부가적인 표현식 • **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]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **expression** - 집계할 필드의 하위 표현식 • **group_by** - 그룹 집계 계산에 사용할 부가적인 표현식 • **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]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **expression** - 집계할 필드의 하위 표현식 • **group_by** - 그룹 집계 계산에 사용할 부가적인 표현식 • **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]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **expression** - 집계할 필드의 하위 표현식 • **group_by** - 그룹 집계 계산에 사용할 부가적인 표현식 • **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]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **expression** - 집계할 필드의 하위 표현식 • **group_by** - 그룹 집계 계산에 사용할 부가적인 표현식 • **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]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **expression** - 집계할 필드의 하위 표현식 • **group_by** - 그룹 집계 계산에 사용할 부가적인 표현식 • **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]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **expression** - 집계할 필드의 하위 표현식 • **group_by** - 그룹 집계 계산에 사용할 부가적인 표현식 • **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]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **expression** - 집계할 필드의 하위 표현식 • **group_by** - 그룹 집계 계산에 사용할 부가적인 표현식 • **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]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **expression** - 집계할 필드의 하위 표현식 • **group_by** - 그룹 집계 계산에 사용할 부가적인 표현식 • **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]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **expression** - 집계할 필드의 하위 표현식 • **group_by** - 그룹 집계 계산에 사용할 부가적인 표현식 • **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]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **expression** - 집계할 필드의 하위 표현식 • **group_by** - 그룹 집계 계산에 사용할 부가적인 표현식 • **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]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **expression** - 집계할 필드의 하위 표현식 • **group_by** - 그룹 집계 계산에 사용할 부가적인 표현식 • **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]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **expression** - 집계할 필드의 하위 표현식 • **group_by** - 그룹 집계 계산에 사용할 부가적인 표현식 • **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]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **relation** - 관계 ID 를 표현하는 문자열 • **aggregate** - 계산할 집계에 대응하는 문자열. 유효한 옵션은 다음과 같습니다: – count – count_distinct – count_missing – min – max – sum – mean – median – stdev – stdevsample – range – minority – majority – q1: 제 1 사분위수 (First Quartile) – q3: 제 3 사분위수 (Third Quartile) – iqr: 사분위간 범위 (Inter Quartile Range) – min_length: 최단 문자열 길이 – max_length: 최장 문자열 길이 – concatenate: 연결자 (concatenator) 로 문자열을 결합 – concatenate_unique: 연결자로 유일 문자열을 결합 – collect: 집계된 다중 부분 도형을 생성 – array_agg: 집계된 값의 배열을 생성 • **expression** - 집계할 하위 표현식 또는 필드명 • **concatenator** - ‘연쇄 (concatenate)’집계를 위한 값을 결합하는 데 사용할 부가적인 문자열 • **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 • `relation_aggregate('my_relation','sum', "passengers"/7)` → ‘my_relation’관계를 사용해서 매칭한 모든 파생 객체에 대해‘passengers’필드의 합을 7 로 나눈 값 • `relation_aggregate('my_relation','concatenate', "towns", concatenator:=',')` → ‘my_relation’관계를 사용해서 매칭한 모든 파생 객체에 대한‘towns’필드의 쉼표 구분 목록 • `relation_aggregate('my_relation','array_agg', "id")` → ‘my_relation’관계를 사용해서 매칭한 모든 파생 객체에서 나온 ID 필드의 목록

더 읽어볼 거리: 일대다 또는 다대다 관계 생성

stdev

필드 또는 표현식에서 집계 표준 편차값을 반환합니다.

문법	stdev(expression, [group_by], [filter]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **expression** - 집계할 필드의 하위 표현식 • **group_by** - 그룹 집계 계산에 사용할 부가적인 표현식 • **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]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **expression** - 집계할 필드의 하위 표현식 • **group_by** - 그룹 집계 계산에 사용할 부가적인 표현식 • **filter** - 집계하는 데 사용되는 객체를 필터링하는 데 사용할 부가적인 표현식
예제	• `sum("population",group_by:="state")` → 'state'필드로 그룹화된'population'값의 합계

14.3.2 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*
- *array_all*
- *array_append*
- *array_cat*
- *array_contains*
- *array_distinct*
- *array_filter*
- *array_find*
- *array_first*

- *array_foreach*
- *array_get*
- *array_insert*
- *array_intersect*
- *array_last*
- *array_length*
- *array_prepend*
- *array_remove_all*
- *array_remove_at*
- *array_reverse*
- *array_slice*
- *array_sort*
- *array_to_string*
- *generate_series*
- *regexp_matches*
- *string_to_array*

array

파라미터로써 전달된 모든 값들을 담고 있는 배열을 반환합니다.

문법	array(value1, value2, ⋯)
인자	• **value** - 값
예제	• `array(2,10)` → [2, 10]

array_all

배열이 지정한 배열의 모든 값을 담고 있는 경우 참을 반환합니다.

문법	array_all(array_a, array_b)
인자	• **array_a** - 배열 • **array_b** - 검색할 값들의 배열
예제	• `array_all(array(1,2,3),array(2,3))` → 참 • `array_all(array(1,2,3),array(1,2,4))` → 거짓

array_append

지정한 값을 맨 끝에 추가한 배열을 반환합니다.

문법	array_append(array, value)
인자	• **array** - 배열 • **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 • **value** - 검색할 값
예제	• `array_contains(array(1,2,3),2)` → 참

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)
인자	• **array** - 배열 • **expression** - 각 항목을 평가할 표현식입니다. 변수 *@element* 는 현재 값으로 대체될 것입니다.
예제	• `array_filter(array(1,2,3),@element < 3)` → [1, 2]

array_find

배열 내부의 값의 (0 으로 시작하는) 인덱스를 반환합니다. 값을 찾을 수 없는 경우 -1 을 반환합니다.

문법	array_find(array, value)
인자	• **array** - 배열 • **value** - 검색할 값
예제	• `array_find(array(1,2,3),2)` → 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 • **expression** - 각 항목을 평가할 표현식입니다. 변수 *@element* 는 현재 값으로 대체될 것입니다.
예제	• `array_foreach(array('a','b','c'),upper(@element))` → ['A', 'B', 'C'] • `array_foreach(array(1,2,3),@element + 10)` → [11, 12, 13]

array_get

배열의 (첫 번째가 0) N 번째 값을 반환합니다.

문법	array_get(array, index)
인자	• **array** - 배열 • **index** - 가져올 인덱스 (0 기반)
예제	• `array_get(array('a','b','c'),1)` → 'b'

array_insert

지정한 위치에 지정한 값을 삽입한 배열을 반환합니다.

문법	array_insert(array, pos, value)
인자	• **array** - 배열 • **pos** - 추가할 위치 (0 기반) • **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 • **array2** - 또다른 배열
예제	• `array_intersect(array(1,2,3,4),array(4,0,2,5))` → 참

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_prepend

지정한 값을 맨 앞에 추가한 배열을 반환합니다.

문법	array_prepend(array, value)
인자	• **array** - 배열 • **value** - 추가할 값
예제	• `array_prepend(array(1,2,3),0)` → [0, 1, 2, 3]

array_remove_all

모든 항목에서 지정한 값을 제거한 배열을 반환합니다.

문법	array_remove_all(array, value)
인자	• **array** - 배열 • **value** - 제거할 값
예제	• `array_remove_all(array('a','b','c','b'),'b')` → ['a', 'c']

array_remove_at

지정한 인덱스를 제거한 배열을 반환합니다.

문법	array_remove_at(array, pos)
인자	• **array** - 배열 • **pos** - 제거할 위치 (0 기반)
예제	• `array_remove_at(array(1,2,3),1)` → [1, 3]

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 • **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) • **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] • `array_slice(array(1,2,3,4,5),0,-1)` → [1, 2, 3, 4, 5] • `array_slice(array(1,2,3,4,5),-5,-1)` → [1, 2, 3, 4, 5] • `array_slice(array(1,2,3,4,5),0,0)` → [1] • `array_slice(array(1,2,3,4,5),-2,-1)` → [4, 5] • `array_slice(array(1,2,3,4,5),-1,-1)` → [5] • `array_slice(array('Dufour','Valmiera','Chugiak','Brighton'),1,2)` → ['Valmiera', 'Chugiak'] • `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]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **array** - 배열 • **ascending** - 배열을 내림차순으로 정렬하려면 이 파라미터를 false 로 설정하십시오.
예제	• `array_sort(array(3,2,1))` → [1, 2, 3]

array_to_string

배열 요소들을 구분자로 구분하고 빈 값은 선택적인 문자열로 대체한 문자열로 연결합니다.

문법	array_to_string(array, [delimiter=','], [empty_value='']) [] 괄호는 부가적인 인자를 표시합니다
인자	• **array** - 입력 배열 • **delimiter** - 연결된 배열 요소들을 분리하는 데 사용하는 문자열 구분자 • **empty_value** - 빈 (길이가 0 인) 값을 대체하는 데 사용할 선택적인 문자열
예제	• `array_to_string(array('1','2','3'))` → '1,2,3' • `array_to_string(array(1,2,3),'-')` → '1-2-3' • `array_to_string(array('1','','3'),',','0')` → '1,0,3'

generate_series

이어지는 일련의 숫자를 담고 있는 배열을 생성합니다.

문법	generate_series(start, stop, [step=1]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **start** - 시퀀스 (일련의 숫자) 의 첫 번째 값 • **stop** - 시퀀스가 도달하면 중지하는 값 • **step** - 값 사이의 증분으로 사용되는 값
예제	• `generate_series(1,5)` → [1, 2, 3, 4, 5] • `generate_series(5,1,-1)` → [5, 4, 3, 2, 1]

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='']) [] 괄호는 부가적인 인자를 표시합니다
인자	• **string** - 정규 표현식에 대해 그룹을 캡처해올 문자열 • **regex** - 그룹을 캡처하는 데 사용된 정규 표현식 • **empty_value** - 빈 (길이가 0 인) 값을 대체하는 데 사용할 선택적인 문자열
예제	• `regexp_matches('QGIS=>rocks','(.*)=>(.*)')` → ['QGIS', 'rocks'] • `regexp_matches('key=>','(.*)=>(.*)','empty value')` → ['key', 'empty value']

string_to_array

문자열을 지정한 구분자와 빈 값을 위한 선택적인 문자열을 사용해서 배열로 분해합니다.

문법	string_to_array(string, [delimiter=','], [empty_value='']) [] 괄호는 부가적인 인자를 표시합니다
인자	• **string** - 입력 문자열 • **delimiter** - 입력 문자열을 분할하는 데 사용한 문자열 구분자 • **empty_value** - 빈 (길이가 0 인) 값을 대체하는 데 사용할 선택적인 문자열
예제	• `string_to_array('1,2,3',',')` → ['1', '2', '3'] • `string_to_array('1,,3',',','0')` → ['1', '0', '3']

14.3.3 색상 함수

이 그룹은 색상을 처리하기 위한 함수를 담고 있습니다.

- *color_cmyk*
- *color_cmyka*
- *color_grayscale_average*
- *color_hsl*
- *color_hsla*
- *color_hsv*
- *color_hsva*
- *color_mix_rgb*
- *color_part*

- *color_rgb*
- *color_rgba*
- *create_ramp*
- *darker*
- *lighter*
- *project_color*
- *ramp_color*
- *set_color_part*

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 • **magenta** - 색상의 자홍색 (magenta) 요소를 0 에서 100 사이의 백분율 정수값으로 입력 • **yellow** - 색상의 황색 (yellow) 요소를 0 에서 100 사이의 백분율 정수값으로 입력 • **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 • **magenta** - 색상의 자홍색 (magenta) 요소를 0 에서 100 사이의 백분율 정수값으로 입력 • **yellow** - 색상의 황색 (yellow) 요소를 0 에서 100 사이의 백분율 정수값으로 입력 • **black** - 색상의 흑색 (black) 요소를 0 에서 100 사이의 백분율 정수값으로 입력 • **alpha** - 투명도를 0(완전히 투명) 에서 255(완전히 불투명) 사이의 정수값으로 입력
예제	• `color_cmyk(100,50,0,10,200)` → '0,115,230,200'

color_grayscale_average

회색조 필터를 적용하고 지정한 색상에서 나온 문자열 표현을 반환합니다.

문법	color_grayscale_average(color)
인자	• **color** - 색상 문자열
예제	• `color_grayscale_average('255,100,50')` → '135,135,135,255'

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 • **saturation** - 색상의 채도 백분율을 0 에서 100 사이의 정수값으로 입력 • **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 • **saturation** - 색상의 채도 백분율을 0 에서 100 사이의 정수값으로 입력 • **lightness** - 색상의 명도 백분율을 0 에서 100 사이의 정수값으로 입력 • **alpha** - 투명도를 0(완전히 투명) 에서 255(완전히 불투명) 사이의 정수값으로 입력
예제	• `color_hsla(100,50,70,200)` → '166,217,140,200'

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 • **saturation** - 색상의 채도 백분율을 0 에서 100 사이의 정수값으로 입력 • **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 • **saturation** - 색상의 채도 백분율을 0 에서 100 사이의 정수값으로 입력 • **value** - 색상의 명도 백분율을 0 에서 100 사이의 정수값으로 입력 • **alpha** - 투명도를 0(완전히 투명) 에서 255(완전히 불투명) 사이의 정수값으로 입력
예제	• `color_hsva(40,100,100,200)` → '255,170,0,200'

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 • **color2** - 색상 문자열 • **ratio** - 비율
예제	• `color_mix_rgb('0,0,0','255,255,255',0.5)` → '127,127,127,255'

color_part

색상 문자열에서 예를 들어 적색 요소 또는 알파 요소 같은 특정 요소를 반환합니다.

문법	color_part(color, component)
인자	• **color** - 색상 문자열 • **component** - 반환할 색상 요소에 대응하는 문자열입니다. 사용할 수 있는 옵션은 다음과 같습니다: – red: RGB 적색 요소 (0-255) – green: RGB 녹색 요소 (0-255) – blue: RGB 청색 요소 (0-255) – alpha: 알파 (투명도) 값 (0-255) – hue: HSV 색조 (0-360) – saturation: HSV 채도 (0-100) – value: HSV 명도 (0-100) – hsl_hue: HSL 색조 (0-360) – hsl_saturation: HSL 채도 (0-100) – lightness: HSL 명도 (0-100) – cyan: CMYK 청록색 요소 (0-100) – magenta: CMYK 자홍색 요소 (0-100) – yellow: CMYK 황색 요소 (0-100) – black: CMYK 흑색 요소 (0-100)
예제	• `color_part('200,10,30','green')` → 10

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 • **green** - 녹색 요소를 0 에서 255 사이의 정수값으로 입력 • **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 • **green** - 녹색 요소를 0 에서 255 사이의 정수값으로 입력 • **blue** - 청색 요소를 0 에서 255 사이의 정수값으로 입력 • **alpha** - 투명도를 0(완전히 투명) 에서 255(완전히 불투명) 사이의 정수값으로 입력
예제	• `color_rgba(255,127,0,200)` → '255,127,0,200'

create_ramp

색상 문자열 및 단계의 맵에서 나온 그레이디언트 색상표를 반환합니다.

문법	create_ramp(map, [discrete=false]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **map** - 색상 문자열과 간격의 맵 • **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

더 어두운 (또는 더 밝은) 색상 문자열을 반환합니다.

문법	darker(color, factor)
인자	• **color** - 색상 문자열 • **factor** - 감광 (darkening) 인자에 대응하는 정수: – 이 인자가 100 을 초과하는 경우, 함수가 더 어두운 색상을 반환합니다. (예를 들어 이 인자를 200 으로 설정하면 휘도가 절반인 색상을 반환합니다.) – 이 인자가 100 미만인 경우 더 밝은 색상을 반환하지만, 이런 목적을 위해서는 lighter() 함수를 사용할 것을 권장합니다. – 이 인자가 0 또는 음수인 경우, 불특정 값을 반환합니다.
예제	• `darker('200,10,30', 200)` → '100,5,15,255'

더 읽어볼 거리: *lighter*

lighter

더 밝은 (또는 더 어두운) 색상 문자열을 반환합니다.

문법	lighter(color, factor)
인자	• **color** - 색상 문자열 • **factor** - 증광 (lightening) 인자에 대응하는 정수: – 이 인자가 100 을 초과하는 경우, 함수가 더 밝은 색상을 반환합니다. (예를 들어 이 인자를 150 으로 설정하면 휘도가 50% 더 높은 색상을 반환합니다.) – 이 인자가 100 미만인 경우 더 어두운 색상을 반환하지만, 이런 목적을 위해서는 darker() 함수를 사용할 것을 권장합니다. – 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

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' • **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 • **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

set_color_part

색상 문자열에 대해 예를 들어 적색 요소 또는 알파 요소 같은 특정 색상 요소를 설정합니다.

문법	set_color_part(color, component, value)
인자	• **color** - 색상 문자열 • **component** - 설정할 색상 요소에 대응하는 문자열입니다. 사용할 수 있는 옵션은 다음과 같습니다: – red: RGB 적색 요소 (0-255) – green: RGB 녹색 요소 (0-255) – blue: RGB 청색 요소 (0-255) – alpha: 알파 (투명도) 값 (0-255) – hue: HSV 색조 (0-360) – saturation: HSV 채도 (0-100) – value: HSV 명도 (0-100) – hsl_hue: HSL 색조 (0-360) – hsl_saturation: HSL 채도 (0-100) – lightness: HSL 명도 (0-100) – cyan: CMYK 청록색 요소 (0-100) – magenta: CMYK 자홍색 요소 (0-100) – yellow: CMYK 황색 요소 (0-100) – black: CMYK 흑색 요소 (0-100) • **value** - 앞에 나열된 범위를 따르는, 색상 요소의 새 값
예제	• `set_color_part('200,10,30','green',50)` → '200,50,30,255'

14.3.4 조건 함수

이 그룹은 표현식에서 조건 검사를 처리하는 함수를 담고 있습니다.

- *CASE*
- *coalesce*
- *if*
- *nullif*
- *regexp_match*
- *try*

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 • **THEN result** - *condition* 이 참으로 평가되는 경우 *result* 를 평가하고 반환합니다. • **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. • `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 유효한 모든 표현식 또는 값입니다.
예제	• `coalesce(NULL, 2)` → 2 • `coalesce(NULL, 2, 3)` → 2 • `coalesce(7, NULL, 3*2)` → 7 • `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 • **result_when_true** - 조건이 참이거나, 거짓으로 변환되지 않는 또다른 값일 경우 반환될 결과입니다. • **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' • `if( 1+1=3, 'Yes', 'No' )` → 'No' • `if( 5 > 3, 1, 0)` → 1 • `if( '', 'It is true (not empty)', 'It is false (empty)' )` → 'It is false (empty)' • `if( ' ', 'It is true (not empty)', 'It is false (empty)' )` → 'It is true (not empty)' • `if( 0, 'One', 'Zero' )` → 'Zero' • `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. • **value2** - NULL 대체를 촉발할 제어 값입니다.
예제	• `nullif('(none)', '(none)')` → NULL • `nullif('text', '(none)')` → 'text' • `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 • **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 • `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]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **expression** - 실행해야 할 표현식 • **alternative** - 표현식이 오류를 반환하는 경우 반환될 결과입니다.
예제	• `try( to_int( '1' ), 0 )` → 1 • `try( to_int( 'a' ), 0 )` → 0 • `try( to_date( 'invalid_date' ) )` → NULL

14.3.5 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*
- *hash*
- *md5*
- *sha256*
- *to_base64*
- *to_date*
- *to_datetime*
- *to_decimal*
- *to_dm*
- *to_dms*
- *to_int*

- *to_interval*
- *to_real*
- *to_string*
- *to_time*

from_base64

Decodes a string in the Base64 encoding into a binary value.

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 • **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' • `hash('QGIS', 'md5')` → '57470aaa9e22adaefac7f5f342f1c6da' • `hash('QGIS', 'sha1')` → 'f87cfb2b74cdd5867db913237024e7001e62b114' • `hash('QGIS', 'sha224')` → '4093a619ada631c770f44bc643ead18fb393b93d6a6af1861fcfece0' • `hash('QGIS', 'sha256')` → 'eb045cba7a797aaa06ac58830846e40c8e8c780bc0676d3393605fae50c053 • `hash('QGIS', 'sha384')` → '91c1de038cc3d09fdd512e99f9dd9922efadc39ed21d3922e69a4305cc2550 • `hash('QGIS', 'sha512')` → 'c2c092f2ab743bf8edbeb6d028a745f30fc720408465ed369421f0a4e20fa5 • `hash('QGIS', 'sha3_224')` → '467f49a5039e7280d5d42fd433e80d203439e338eaabd701f0d6c17d' • `hash('QGIS', 'sha3_256')` → '540f7354b6b8a6e735f2845250f15f4f3ba4f666c55574d9e9354575de0 • `hash('QGIS', 'sha3_384')` → '96052da1e77679e9a65f60d7ead961b28797782314478638 6eb43647b0 • `hash('QGIS', 'sha3_512')` → '900d079dc69761da113980253aa8ac0414a8bd6d09879a916228f87437 • `hash('QGIS', 'keccak_224')` → '5b0ce6acef8b0a121d4ac4f3eaa8503c799ad4e26a3392d1fb201478' • `hash('QGIS', 'keccak_256')` → '991c520aa6815392de24087f61b2ae0fd56abbfeee4a8ca019c1011d3 • `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

Encodes a binary value into a string, using the Base64 encoding.

문법	to_base64(value)
인자	• **value** - 인코딩할 바이너리 값
예제	• `to_base64('QGIS')` → 'UUdJUw=='

to_date

문자열을 날짜 객체로 변환합니다. 문자열을 파싱하기 위해 부가적인 서식 문자열을 지정할 수 있습니다; 해당 서식에 대한 추가 문서를 보고 싶다면 QDate::fromString 을 참조하세요.

문법	to_date(string, [format], [language]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **string** - 날짜 값을 나타내는 문자열 • **format** - 문자열을 날짜 유형으로 변환하는 데 쓰이는 서식 • **language** - 문자열을 날짜 유형으로 변환하는 데 사용된 언어 (소문자, 2-3 문자, ISO 639 언어 코드)
예제	• `to_date('2012-05-04')` → 2012-05-04 • `to_date('June 29, 2019','MMMM d, yyyy')` → 2019-06-29 • `to_date('29 juin, 2019','d MMMM, yyyy','fr')` → 2019-06-29

to_datetime

문자열을 날짜 & 시간 객체로 변환합니다. 문자열을 파싱하기 위해 부가적인 서식 문자열을 지정할 수 있습니다; 해당 서식에 대한 추가 문서를 보고 싶다면 QDate::fromString 과 QTime::fromString 을 참조하세요.

문법	to_datetime(string, [format], [language]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **string** - 날짜 & 시간 값을 나타내는 문자열 • **format** - 문자열을 날짜 & 시간 유형으로 변환하는 데 쓰이는 서식 • **language** - 문자열을 날짜 & 시간 유형으로 변환하는 데 사용된 언어 (소문자, 2-3 문자, ISO 639 언어 코드)
예제	• `to_datetime('2012-05-04 12:50:00')` → 2012-05-04T12:50:00 • `to_datetime('June 29, 2019 @ 12:34','MMMM d, yyyy @ HH:mm')` → 2019-06-29T12:34 • `to_datetime('29 juin, 2019 @ 12:34','d MMMM, yyyy @ HH:mm','fr')` → 2019-06-29T12:34

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=]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **coordinate** - 경도 또는 위도 값 • **axis** - 좌표의 축. 'x'또는'y'입니다. • **precision** - 소수점 이하 자릿수 • **formatting** - 서식 유형을 지정합니다. 받을 수 있는 값은 NULL(기본값), 'aligned'또는 'suffix'입니다.
예제	• `to_dm(6.1545681, 'x', 3)` → 6°9.274′ • `to_dm(6.1545681, 'y', 4, 'aligned')` → 6°09.2741′N • `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=]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **coordinate** - 경도 또는 위도 값 • **axis** - 좌표의 축. 'x'또는'y'입니다. • **precision** - 소수점 이하 자릿수 • **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″ • `to_dms(6.1545681, 'y', 4, 'aligned')` → 6°09′16.4452″N • `to_dms(6.1545681, 'y', 4, 'suffix')` → 6°9′16.4452″N

to_int

문자열을 정수형 숫자로 변환합니다. 값을 정수형으로 변환할 수 없는 경우 (예를 들어'123asd'는 유효하지 않습니다) 아무것도 반환하지 않습니다.

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 • `to_interval( '0.5 hours' )` → interval: 30 minutes • `to_datetime('2012-05-05 12:00:00') - to_interval('1 day 2 hours')` → 2012-05-04T10:00:00

to_real

문자열을 실수형 숫자로 변환합니다. 값을 실수형으로 변환할 수 없는 경우 (예를 들어'123.56asd'는 유효하지 않습니다) 아무것도 반환하지 않습니다. 정밀도가 변환 결과보다 작은 경우 변경 사항을 저장한 다음 숫자를 반올림합니다.

문법	to_real(string)
인자	• **string** - 실수형 숫자로 변환할 문자열
예제	• `to_real('123.45')` → 123.45

to_string

숫자를 문자열로 변환합니다.

문법	to_string(number)
인자	• **number** - 정수형 또는 실수형 값. 문자열로 변환할 숫자입니다.
예제	• `to_string(123)` → '123'

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]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **string** - 시간 값을 나타내는 문자열 • **format** - 문자열을 시간 유형으로 변환하는 데 쓰이는 서식 • **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 • `to_time('12:34','HH:mm')` → 12:34:00 • `to_time('12:34','HH:mm','fr')` → 12:34:00

14.3.6 사용자 지정 함수

이 그룹은 사용자가 생성한 함수를 담고 있습니다. 자세한 내용은함수 편집기 를 참조하세요.

14.3.7 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*
- *datetime_from_epoch*
- *day*
- *day_of_week*
- *epoch*
- *format_date*
- *hour*
- *make_date*
- *make_datetime*
- *make_interval*

- *make_time*
- *minute*
- *month*
- *now*
- *second*
- *to_date*
- *to_datetime*
- *to_interval*
- *to_time*
- *week*
- *year*

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 • **datetime2** - 앞 날짜를 나타내는 문자열, 날짜 또는 날짜 & 시간
예제	• `day(age('2012-05-12','2012-05-02'))` → 10 • `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 • `day(to_interval('3 weeks 2 days'))` → 23 • `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.

<table>
<tr><td>문법</td><td>format_date(datetime, format, [language])
[] 괄호는 부가적인 인자를 표시합니다</td></tr>
<tr><td>인자</td><td>

datetime - 날짜, 시간 또는 날짜 & 시간 값
format - 문자열을 서식화하는 데 쓰이는 문자열 템플릿

<table>
<tr><th>표현식</th><th>산출물</th></tr>
<tr><td>d</td><td>앞에 0 이 없는 숫자 일 (1 에서 31)</td></tr>
<tr><td>dd</td><td>앞에 0 이 있는 숫자 일 (01 에서 31)</td></tr>
<tr><td>ddd</td><td>축약된 현지 요일 이름 (예: 'Mon'에서'Sun')</td></tr>
<tr><td>dddd</td><td>축약하지 않은 현지 요일 이름 (예: 'Monday'에서'Sunday')</td></tr>
<tr><td>M</td><td>앞에 0 이 없는 숫자 월 (1-12)</td></tr>
<tr><td>MM</td><td>앞에 0 이 있는 숫자 월 (01-12)</td></tr>
<tr><td>MMM</td><td>축약된 현지 월 이름 (예: 'Jan'에서'Dec')</td></tr>
<tr><td>MMMM</td><td>축약하지 않은 현지 월 이름 (예: 'January'에서'December')</td></tr>
<tr><td>yy</td><td>두 자리 숫자 연도 (00-99)</td></tr>
<tr><td>yyyy</td><td>네 자리 숫자 연도</td></tr>
</table>
다음 표현식은 서식 문자열의 시간 부분에 쓰일 수도 있습니다:
<table>
<tr><th>표현식</th><th>산출물</th></tr>
<tr><td>h</td><td>앞에 0 이 없는 시간 (0 에서 23 또는 AM/PM 표시가 있는 경우 1 에서 12)</td></tr>
<tr><td>hh</td><td>앞에 0 이 있는 시간 (00 에서 23 또는 AM/PM 표시가 있는 경우 01 에서 12)</td></tr>
<tr><td>H</td><td>앞에 0 이 없는 시간 (AM/PM 표시가 있는 경우에도 0 에서 23)</td></tr>
<tr><td>HH</td><td>앞에 0 이 있는 시간 (AM/PM 표시가 있는 경우에도 00 에서 23)</td></tr>
<tr><td>m</td><td>앞에 0 이 없는 분 (0 에서 59)</td></tr>
<tr><td>mm</td><td>앞에 0 이 있는 분 (00 에서 59)</td></tr>
<tr><td>s</td><td>앞에 0 이 없는 초 (0 에서 59)</td></tr>
<tr><td>ss</td><td>앞에 0 이 있는 초 (00 에서 59)</td></tr>
<tr><td>z</td><td>뒤에 0 이 없는 밀리초 (0 에서 999)</td></tr>
<tr><td>zzz</td><td>뒤에 0 이 있는 밀리초 (000 에서 999)</td></tr>
<tr><td>AP or A</td><td>AM/PM 시간으로 해석합니다. AP 는'AM'또는'PM'이어야만 합니다.</td></tr>
<tr><td>ap or a</td><td>AM/PM 시간으로 해석합니다. ap 는'am'또는'pm'이어야만 합니다.</td></tr>
</table>

날짜 유형을 사용자 지정 문자열로 서식화하는 데 사용된 언어 (소문자, 2-3 문자, ISO 639 언어 코드)

</td></tr>
<tr><td>예제</td><td>

<code>format_date('2012-05-15','dd.MM.yyyy')</code> → '15.05.2012'
<code>format_date('2012-05-15','d MMMM yyyy','fr')</code> → '15 mai 2012'
<code>format_date('2012-05-15','dddd')</code> → 'Tuesday'
<code>format_date('2012-05-15 13:54:20','dd.MM.yy')</code> → '15.05.12'
<code>format_date('13:54:20','hh:mm AP')</code> → '01:54 PM'

</td></tr>
</table>

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 • `hour(age('2012-07-22T13:00:00','2012-07-22T10:00:00'))` → 3 • `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 유효하지 않습니다. • **month** - 월 번호, 1=January • **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 유효하지 않습니다. • **month** - 월 번호, 1=January • **day** - 일 번호, 월의 첫날을 1 로 시작 • **hour** - 시간 번호 • **minute** - 분 • **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]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **years** - 연도 개수 (1 년 길이를 365.25 일로 가정) • **months** - 월 개수 (1 개월 길이를 30 일로 가정) • **weeks** - 주 개수 • **days** - 일 개수 • **hours** - 시간 개수 • **minutes** - 분 개수 • **seconds** - 초 개수
예제	• `make_interval(hours:=3)` → interval: 3 hours • `make_interval(days:=2, hours:=3)` → interval: 2.125 days • `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 • **minute** - 분 • **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 • `minute(age('2012-07-22T00:20:00','2012-07-22T00:00:00'))` → 20 • `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 • `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 • `second(age('2012-07-22T00:20:00','2012-07-22T00:00:00'))` → 1200 • `second(age('2012-01-01','2010-01-01'))` → 63072000

to_date

문자열을 날짜 객체로 변환합니다. 문자열을 파싱하기 위해 부가적인 서식 문자열을 지정할 수 있습니다; 해당 서식에 대한 추가 문서를 보고 싶다면 QDate::fromString 을 참조하세요.

문법	to_date(string, [format], [language]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **string** - 날짜 값을 나타내는 문자열 • **format** - 문자열을 날짜 유형으로 변환하는 데 쓰이는 서식 • **language** - 문자열을 날짜 유형으로 변환하는 데 사용된 언어 (소문자, 2-3 문자, ISO 639 언어 코드)
예제	• `to_date('2012-05-04')` → 2012-05-04 • `to_date('June 29, 2019','MMMM d, yyyy')` → 2019-06-29 • `to_date('29 juin, 2019','d MMMM, yyyy','fr')` → 2019-06-29

to_datetime

문자열을 날짜 & 시간 객체로 변환합니다. 문자열을 파싱하기 위해 부가적인 서식 문자열을 지정할 수 있습니다; 해당 서식에 대한 추가 문서를 보고 싶다면 QDate::fromString 과 QTime::fromString 을 참조하세요.

문법	to_datetime(string, [format], [language]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **string** - 날짜 & 시간 값을 나타내는 문자열 • **format** - 문자열을 날짜 & 시간 유형으로 변환하는 데 쓰이는 서식 • **language** - 문자열을 날짜 & 시간 유형으로 변환하는 데 사용된 언어 (소문자, 2-3 문자, ISO 639 언어 코드)
예제	• `to_datetime('2012-05-04 12:50:00')` → 2012-05-04T12:50:00 • `to_datetime('June 29, 2019 @ 12:34','MMMM d, yyyy @ HH:mm')` → 2019-06-29T12:34 • `to_datetime('29 juin, 2019 @ 12:34','d MMMM, yyyy @ HH:mm', 'fr')` → 2019-06-29T12:34

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 • `to_interval( '0.5 hours' )` → interval: 30 minutes • `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]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **string** - 시간 값을 나타내는 문자열 • **format** - 문자열을 시간 유형으로 변환하는 데 쓰이는 서식 • **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 • `to_time('12:34','HH:mm')` → 12:34:00 • `to_time('12:34','HH:mm','fr')` → 12:34:00

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 • `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 • `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>
```

14.3.8 필드 및 값

이 그룹은 레이어에 있는 필드의 목록을 담고 있습니다.

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.

14.3.9 파일 및 경로 함수

이 그룹은 파일 및 경로 명칭을 처리하는 함수를 담고 있습니다.

- *base_file_name*
- *file_exists*
- *file_name*
- *file_path*
- *file_size*
- *file_suffix*
- *is_directory*
- *is_file*

base_file_name

디렉터리 또는 파일 확장자 없이 파일의 기본명 (base name) 을 반환합니다.

문법	base_file_name(path)
인자	• **path** - 파일 경로
예제	• `base_file_name('/home/qgis/data/country_boundaries.shp')` → 'country_boundaries'

file_exists

파일 경로가 실재하는 경우 참을 반환합니다.

문법	file_exists(path)
인자	• **path** - 파일 경로
예제	• `file_exists('/home/qgis/data/country_boundaries.shp')` → 참

file_name

디렉터리를 제외한 (파일 확장자 포함) 파일명을 반환합니다.

문법	file_name(path)
인자	• **path** - 파일 경로
예제	• `file_name('/home/qgis/data/country_boundaries.shp')` → 'country_boundaries.shp'

file_path

파일 경로의 디렉터리 요소를 반환합니다. 파일명을 포함하지 않습니다.

문법	file_path(path)
인자	• **path** - 파일 경로
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
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
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
예제	• `is_directory('/home/qgis/data/country_boundaries.shp')` → 거짓 • `is_directory('/home/qgis/data/')` → 참

is_file

경로가 파일에 대응하는 경우 참을 반환합니다.

문법	is_file(path)
인자	• **path** - 파일 경로
예제	• `is_file('/home/qgis/data/country_boundaries.shp')` → 참 • `is_file('/home/qgis/data/')` → 거짓

14.3.10 양식 함수

이 함수 그룹은 속성 양식 맥락에서만, 예를 들면필드 위젯 설정에서 실행되는 함수들을 담고 있습니다.

- *current_parent_value*
- *current_value*

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.

14.3.11 퍼지 매칭 함수

이 그룹은 값들을 퍼지 비교하기 위한 함수를 담고 있습니다.

- *hamming_distance*
- *levenshtein*
- *longest_common_substring*
- *soundex*

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 • **string2** - 문자열
예제	• `hamming_distance('abc','xec')` → 2 • `hamming_distance('abc','ABc')` → 2 • `hamming_distance(upper('abc'),upper('ABC'))` → 0

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 • **string2** - 문자열
예제	• `levenshtein('kittens','mitten')` → 2 • `levenshtein('Kitten','kitten')` → 1 • `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 • **string2** - 문자열
예제	• `longest_common_substring('ABABC','BABCA')` → 'BABC' • `longest_common_substring('abcDeF','abcdef')` → 'abc' • `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' • `soundex('rupert')` → 'R163' • `soundex('rubin')` → 'R150'

14.3.12 일반 함수

이 그룹은 일반으로 분류된 함수를 담고 있습니다.

- *env*
- *eval*
- *eval_template*
- *is_layer_visible*
- *layer_property*
- *var*
- *with_variable*

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' • `env( 'MY_OWN_PREFIX_VAR' )` → 'Z:' • `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' • `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** - 레이어명 또는 레이어 ID 가운데 하나를 나타내는 문자열
예제	• `is_layer_visible('baseraster')` → 참

layer_property

일치하는 레이어 속성 또는 메타데이터 값을 반환합니다.

문법	layer_property(layer, property)
인자	• **layer** - 레이어명 또는 레이어 ID 가운데 하나를 나타내는 문자열 • **property** - 반환할 속성에 대응하는 문자열입니다. 사용할 수 있는 옵션은 다음과 같습니다: – name: 레이어명 – id: 레이어 ID – title: 메타데이터 제목 문자열 – abstract: 메타데이터 요약 문자열 – keywords: 메타데이터 키워드 – data_url: 메타데이터 URL – attribution: 메타데이터 권한 문자열 – attribution_url: 메타데이터 권한 URL – source: 레이어 소스 – min_scale: 레이어 가시화 최저 축척 – max_scale: 레이어 가시화 최고 축척 – is_editable: 레이어 편집 모드가 켜져 있는지 여부 – crs: 레이어 좌표계 – crs_definition: 레이어 좌표계 전체 정의 – crs_description: 레이어 좌표계 설명 – extent: 레이어 범위 (도형 객체로서) – distance_units: 레이어 거리 단위 – type: 레이어 유형, 예: 벡터 또는 래스터 – storage_type: 저장소 포맷 (벡터 레이어 전용) – geometry_type: 도형 유형, 예: 포인트 (벡터 레이어 전용) – feature_count: 레이어에 있는 피처의 대략적인 개수 (벡터 레이어 전용) – path: 레이어 데이터소스를 가리키는 파일 경로. 파일 기반 레이어에만 사용할 수 있습니다.
예제	• `layer_property('streets','title')` → 'Basemap Streets' • `layer_property('airports','feature_count')` → 120 • `layer_property('landsat','crs')` → 'EPSG:4326'

더 읽어볼 거리: 벡터, 래스터 및메시 레이어 속성

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 • **value** - 설정할 값 • **expression** - 변수를 사용할 수 있는 표현식
예제	• `with_variable('my_sum', 1 + 2 + 3, @my_sum * 2 + @my_sum * 5)` → 42

14.3.13 도형 함수

이 그룹은 도형 객체를 대상으로 하는 (예: buffer, transform, $area 등) 함수를 담고 있습니다.

- *angle_at_vertex*
- *$area*
- *area*
- *azimuth*
- *boundary*
- *bounds*
- *bounds_height*
- *bounds_width*
- *buffer*
- *buffer_by_m*
- *centroid*
- *close_line*
- *closest_point*
- *collect_geometries*
- *combine*
- *contains*
- *convex_hull*
- *crosses*
- *difference*
- *disjoint*
- *distance*

- *distance_to_vertex*
- *end_point*
- *extend*
- *exterior_ring*
- *extrude*
- *flip_coordinates*
- *force_rhr*
- *geom_from_gml*
- *geom_from_wkb*
- *geom_from_wkt*
- *geom_to_wkb*
- *geom_to_wkt*
- *$geometry*
- *geometry*
- *geometry_n*
- *hausdorff_distance*
- *inclination*
- *interior_ring_n*
- *intersection*
- *intersects*
- *intersects_bbox*
- *is_closed*
- *is_empty*
- *is_empty_or_null*
- *is_multipart*
- *is_valid*
- *$length*
- *length*
- *line_interpolate_angle*
- *line_interpolate_point*
- *line_locate_point*
- *line_merge*
- *line_substring*
- *m*
- *m_max*

- *m_min*
- *main_angle*
- *make_circle*
- *make_ellipse*
- *make_line*
- *make_point*
- *make_point_m*
- *make_polygon*
- *make_rectangle_3points*
- *make_regular_polygon*
- *make_square*
- *make_triangle*
- *minimal_circle*
- *nodes_to_points*
- *num_geometries*
- *num_interior_rings*
- *num_points*
- *num_rings*
- *offset_curve*
- *order_parts*
- *oriented_bbox*
- *overlaps*
- *overlay_contains*
- *overlay_crosses*
- *overlay_disjoint*
- *overlay_equals*
- *overlay_intersects*
- *overlay_nearest*
- *overlay_touches*
- *overlay_within*
- *$perimeter*
- *perimeter*
- *point_n*
- *point_on_surface*
- *pole_of_inaccessibility*

- *project*
- *relate*
- *reverse*
- *rotate*
- *segments_to_lines*
- *shortest_line*
- *simplify*
- *simplify_vw*
- *single_sided_buffer*
- *smooth*
- *start_point*
- *sym_difference*
- *tapered_buffer*
- *touches*
- *transform*
- *translate*
- *union*
- *wedge_buffer*
- *within*
- *$x*
- *x*
- *$x_at*
- *x_max*
- *x_min*
- *$y*
- *y*
- *$y_at*
- *y_max*
- *y_min*
- *z*
- *z_max*
- *z_min*

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 • **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

$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 참조 시스템 (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

포인트 a 의 수직선에서 포인트 b 로 시계 방향으로 측정한 진북 기준 방위각을 라디안 단위 각도로 반환합니다.

문법	azimuth(point_a, point_b)
인자	• **point_a** - 포인트 도형 • **point_b** - 포인트 도형
예제	• `degrees( azimuth( make_point(25, 45), make_point(75, 100) ) )` → 42.273689 • `degrees( azimuth( make_point(75, 100), make_point(25,45) ) )` → 222.273689

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)' • `geom_to_wkt(boundary(geom_from_wkt('LineString(1 1,0 0,-1 1)')))` → 'MultiPoint ((1 1),(-1 1))'

더 읽어볼 거리: 경계 알고리즘

bounds

입력 도형의 경계 상자 (bounding box) 를 표현하는 도형을 반환합니다. 해당 도형의 공간 참조 시스템 상에서 경계 상자를 계산합니다.

문법	bounds(geometry)
인자	• **geometry** - 도형
예제	• `bounds($geometry)` → 현재 피처 도형의 경계 상자 • `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))'

더 읽어볼 거리: 경계 상자 알고리즘

bounds_height

도형의 경계 상자 (bounding box) 의 높이를 표현하는 도형을 반환합니다. 해당 도형의 공간 참조 시스템 상에서 경계 상자를 계산합니다.

문법	bounds_height(geometry)
인자	• **geometry** - 도형
예제	• `bounds_height($geometry)` → 현재 피처 도형의 경계 상자의 높이 • `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 참조 시스템 상에서 경계 상자를 계산합니다.

문법	bounds_width(geometry)
인자	• **geometry** - 도형
예제	• `bounds_width($geometry)` → 현재 피처 도형의 경계 상자의 너비 • `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 참조 시스템 상에서 거리를 계산합니다.

문법	buffer(geometry, distance, [segments=8]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **geometry** - 도형 • **distance** - 레이어 단위의 버퍼 거리 • **segments** - 둥근 결합 스타일을 사용한 경우 사분원을 나타내는 데 사용할 선분의 개수입니다. 숫자가 클수록 더 많은 노드를 가진 매끈한 버퍼를 생성합니다.
예제	• `buffer($geometry, 10.5)` → 현재 피처 도형에 버퍼를 10.5 단위만큼 적용한 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]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **geometry** - 입력 도형. M 값을 가진 (멀티) 라인 도형이어야만 합니다. • **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

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

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)' • `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 • **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 • **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))' • `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)'

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 • **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 )' ) )` → 참 • `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)' ) )` → 거짓

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: 볼록 껍질 알고리즘

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 • **geometry2** - 도형
예제	• `crosses( geom_from_wkt( 'LINESTRING(3 5, 4 4, 5 3)' ), geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ) )` → 참 • `crosses( geom_from_wkt( 'POINT(4 5)' ), geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ) )` → 거짓

더 읽어볼 거리: *overlay_crosses*

difference

geometry2 와 교차하지 않는 geometry1 의 부분을 표현하는 도형을 반환합니다.

문법	difference(geometry1, geometry2)
인자	• **geometry1** - 도형 • **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)'

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 • **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)' ) )` → 참 • `disjoint( geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ), geom_from_wkt( 'POINT(4 4)' ))` → 거짓

더 읽어볼 거리: *overlay_disjoint*

distance

두 도형 사이의 (공간 참조 시스템 기반) 최단 거리를 투영체 단위로 반환합니다.

문법	distance(geometry1, geometry2)
인자	• **geometry1** - 도형 • **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 • **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: 특정 꼭짓점 추출하기 알고리즘

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 참조 시스템의 단위입니다.

문법	extend(geometry, start_distance, end_distance)
인자	• **geometry** - (멀티) 라인스트링 도형 • **start_distance** - 라인 시작점을 연장할 거리 • **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)' • `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

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)'

extrude

입력 [멀티] 곡선 또는 [멀티] 라인스트링을 지정한 X 및 Y 좌표로 연장한 압출 (extruded) 버전을 반환합니다.

문법	extrude(geometry, x, y)
인자	• **geometry** - 폴리곤 도형 • **x** - x 확장, 숫자 값 • **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))' • `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)))'

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)'

더 읽어볼 거리: *X* 좌표와 *Y* 좌표 바꾸기 알고리즘

force_rhr

도형이 오른손 규칙을 따르도록 강제합니다. 폴리곤 경계 안의 영역이 경계의 오른쪽에 있다는 의미입니다. 자세히 말하자면, 외곽 고리는 시계 방향이며 내곽 고리는 반시계 방향입니다.

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

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]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **geometry** - 도형 • **precision** - 숫자 정밀도
예제	• `geom_to_wkt( make_point(6, 50) )` → 'POINT(6 50)' • `geom_to_wkt(centroid(geom_from_wkt('Polygon((1 1, 0 0, -1 1, 1 1))')))` → 'POINT(0 0.66666667)' • `geom_to_wkt(centroid(geom_from_wkt('Polygon((1 1, 0 0, -1 1, 1 1))')), 2)` → 'POINT(0 0.67)'

$geometry

현재 객체의 도형을 반환합니다. 다른 함수들과 함께 공간 처리하하기 위해 사용할 수 있습니다.

문법	$geometry
예제	• `geom_to_wkt( $geometry )` → 'POINT(6 50)'

geometry

객체의 도형을 반환합니다.

문법	geometry(feature)
인자	• **feature** - 피처 객체
예제	• `geom_to_wkt( geometry( get_feature( layer, attributeField, value ) ) )` → 'POINT(6 50)' • `intersects( $geometry, geometry( get_feature( layer, attributeField, value ) ) )` → 참

geometry_n

도형 집합에서 특정 도형을 반환하거나, 입력 도형이 집합이 아닌 경우 NULL 을 반환합니다.

문법	geometry_n(geometry, index)
인자	• **geometry** - 도형 집합 • **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)'

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 매칭 선형 네트워크에서 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]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **geometry1** - 도형 • **geometry2** - 도형 • **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 • `hausdorff_distance( geom_from_wkt('LINESTRING (130 0, 0 0, 0 150)'),geom_from_wkt('LINESTRING (10 10, 10 150, 130 10)'))` → 14.142135623 • `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

포인트 a 에서 포인트 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 • **point_b** - 포인트 도형
예제	• `inclination( make_point( 5, 10, 0 ), make_point( 5, 10, 5 ) )` → 0.0 • `inclination( make_point( 5, 10, 0 ), make_point( 5, 10, 0 ) )` → 90.0 • `inclination( make_point( 5, 10, 0 ), make_point( 50, 100, 0 ) )` → 90.0 • `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 • **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))'

intersection

두 도형이 공유하는 부분을 표현하는 도형을 반환합니다.

문법	intersection(geometry1, geometry2)
인자	• **geometry1** - 도형 • **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)' • `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 • **geometry2** - 도형
예제	• `intersects( geom_from_wkt( 'POINT(4 4)' ), geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ) )` → 참 • `intersects( geom_from_wkt( 'POINT(4 5)' ), geom_from_wkt( 'POINT(5 5)' ) )` → 거짓

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 • **geometry2** - 도형
예제	• `intersects_bbox( geom_from_wkt( 'POINT(4 5)' ), geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ) )` → 참 • `intersects_bbox( geom_from_wkt( 'POINT(6 5)' ), geom_from_wkt( 'POLYGON((3 3, 4 4, 5 5, 3 3))' ) )` → 거짓

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)'))` → 거짓 • `is_closed(geom_from_wkt('LINESTRING(0 0, 1 1, 2 2, 0 0)'))` → 참

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)'))` → 거짓 • `is_empty(geom_from_wkt('LINESTRING EMPTY'))` → 참 • `is_empty(geom_from_wkt('POINT(7 4)'))` → 거짓 • `is_empty(geom_from_wkt('POINT EMPTY'))` → 참

is_empty_or_null

도형이 NULL 이거나 비어 있는 (좌표가 없는) 경우 참을 반환하고, 그렇지 않은 경우 거짓을 반환합니다. 이 함수는 '$geometry IS NULL or is_empty($geometry)'표현식과 비슷합니다.

문법	is_empty_or_null(geometry)
인자	• **geometry** - 도형
예제	• `is_empty_or_null(NULL)` → 참 • `is_empty_or_null(geom_from_wkt('LINESTRING(0 0, 1 1, 2 2)'))` → 거짓 • `is_empty_or_null(geom_from_wkt('LINESTRING EMPTY'))` → 참 • `is_empty_or_null(geom_from_wkt('POINT(7 4)'))` → 거짓 • `is_empty_or_null(geom_from_wkt('POINT EMPTY'))` → 참

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))'))` → 참 • `is_multipart(geom_from_wkt('POINT (0 0)'))` → 거짓

is_valid

도형이 유효한 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)'))` → 참 • `is_valid(geom_from_wkt('LINESTRING(0 0)'))` → 거짓

$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 참조 시스템 (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

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 • **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 • **distance** - 라인을 따라 보간할 거리
예제	• `geom_to_wkt(line_interpolate_point(geometry:=geom_from_wkt('LineString(0 0, 10 0)'),distance:=5))` → 'Point (5 0)'

더 읽어볼 거리: 라인에 포인트를 보간하기 알고리즘

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 • **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)' • `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 • **start_distance** - 하위 스트링의 시작점까지의 거리 • **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

m

포인트 도형의 M 값을 반환합니다.

문법	m(geometry)
인자	• **geometry** - 포인트 도형
예제	• `m( geom_from_wkt( 'POINTM(2 5 4)' ) )` → 4

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 • `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 • `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

도형의 (북쪽에서 도 단위로 시계 방향) 주 각도 (main angle) 를 반환합니다. 도형을 완전히 커버하는 기울어진 최소 경계 직사각형의 각도를 나타냅니다.

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]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **center** - 원의 중심 포인트 • **radius** - 원의 반지름 • **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))' • `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))' • `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]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **center** - 타원의 중심 포인트 • **semi_major_axis** - 타원의 긴반지름 • **semi_minor_axis** - 타원의 짧은반지름 • **azimuth** - 타원의 방향 • **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))' • `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))' • `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)' • `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

Creates a point geometry from an x and y (and optional z and m) value.

문법	make_point(x, y, [z], [m]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **x** - 포인트의 X 좌표 • **y** - 포인트의 Y 좌표 • **z** - 포인트의 부가적인 Z 좌표 • **m** - 포인트의 부가적인 M 값
예제	• `geom_to_wkt(make_point(2,4))` → 'Point (2 4)' • `geom_to_wkt(make_point(2,4,6))` → 'PointZ (2 4 6)' • `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 • **y** - 포인트의 Y 좌표 • **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], ⋯) [] 괄호는 부가적인 인자를 표시합니다
인자	• **outerRing** - 폴리곤의 외곽 고리가 될 닫힌 라인 도형 • **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))' • `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]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **point1** - 첫 번째 포인트 • **point2** - 두 번째 포인트 • **point3** - 세 번째 포인트 • **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(make_point(0, 0), make_point(0, 5), make_point(5, 5), 0)))` → 'Polygon ((0 0, 0 5, 5 5, 5 0, 0 0))' • `geom_to_wkt(make_rectangle(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]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **center** - 정다각형 폴리곤의 중심 • **radius** - 두 번째 포인트. 정다각형 폴리곤이 내접되는 경우 첫 번째가 됩니다. 정다각형 폴리곤이 외접되는 경우 첫 번째 변의 중간점이 됩니다. • **number_sides** - 정다각형 폴리곤의 변/경계선의 개수 • **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))' • `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 • **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))' • `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 • **point2** - 삼각형의 두 번째 포인트 • **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))' • `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)'

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]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **geometry** - 도형 • **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))' • `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

nodes_to_points

입력 도형에 있는 모든 노드로 이루어진 멀티포인트 도형을 반환합니다.

문법	nodes_to_points(geometry, [ignore_closing_nodes=false]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **geometry** - 도형 객체 • **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))' • `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

num_geometries

도형 집합에 있는 도형의 개수를 반환하거나, 입력 도형이 집합이 아닌 경우 NULL 을 반환합니다.

문법	num_geometries(geometry)
인자	• **geometry** - 도형 집합
예제	• `num_geometries(geom_from_wkt('GEOMETRYCOLLECTION(POINT(0 1), POINT(0 0), POINT(1 0), POINT(1 1))'))` → 4

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 참조 시스템의 단위입니다.

문법	offset_curve(geometry, distance, [segments=8], [join=1], [miter_limit=2.0]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **geometry** - (멀티) 라인스트링 도형 • **distance** - 오프셋 거리입니다. 양의 값이면 라인 좌측으로 버퍼를 적용하고, 음의 값이면 라인 우측으로 적용할 것입니다 • **segments** - 둥근 결합 스타일을 사용한 경우 사분원을 나타내는 데 사용할 선분의 개수입니다. 숫자가 클수록 더 많은 노드를 가진 매끈한 라인을 생성합니다. • **join** - 모서리 부분의 결합 스타일. 1 = 둥글게, 2 = 마이터, 그리고 3 = 비스듬하게 • **miter_limit** - 매우 뾰족한 모서리에 사용된 마이터 비율에 대한 제한 (마이터 결합만 사용하는 경우)
예제	• `offset_curve($geometry, 10.5)` → 라인을 왼쪽으로 10.5 단위만큼 오프셋 • `offset_curve($geometry, -10.5)` → 라인을 오른쪽으로 10.5 단위만큼 오프셋 • `offset_curve($geometry, 10.5, segments=16, join=1)` → 라인을 왼쪽으로 10.5 단위만큼 오프셋. 선분을 더 많이 사용할수록 더 매끈한 곡선을 생성합니다. • `offset_curve($geometry, 10.5, join=3)` → 비스듬한 결합 스타일을 사용해서 라인을 왼쪽으로 10.5 단위만큼 오프셋

더 읽어볼 거리: 라인 오프셋시키기 알고리즘

order_parts

지정한 기준으로 멀티 도형의 부분들을 정렬합니다.

문법	order_parts(geometry, orderby, ascending)
인자	• **geometry** - 멀티 유형 도형 • **orderby** - 정렬 기준을 정의하는 표현식 문자열 • **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)))' • `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

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 • **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)' ) )` → 참 • `overlaps( geom_from_wkt( 'LINESTRING(0 0, 1 1)' ), geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ) )` → 거짓

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]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **layer** - 중첩을 검증할 레이어 • **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. • **filter** - 검증할 대상 피처를 필터링할 선택적인 표현식입니다. 이 표현식을 설정하지 않은 경우, 모든 피처를 검증할 것입니다. • **limit** - 매칭하는 피처의 개수를 제한할 선택적인 정수입니다. 이 파라미터를 설정하지 않은 경우, 매칭하는 모든 피처를 반환할 것입니다. • **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')` → 현재 피처가'region'을 공간적으로 담고 있는 경우 참을 반환합니다. • `overlay_contains('regions', filter:= population > 10000)` → 현재 피처가 인구 10,000 명을 초과하는'region'을 공간적으로 담고 있는 경우 참을 반환합니다. • `overlay_contains('regions', name)` → 현재 피처에 담겨 있는'region'의 이름들의 배열 • `array_to_string(overlay_contains('regions', name))` → 현재 피처에 담겨 있는'region'의 이름들을 쉼표로 구분한 목록인 문자열 • `array_length(overlay_contains('regions', name))` → 현재 피처에 담겨 있는'region'의 개수 • `array_sort(overlay_contains(layer:='regions', expression:="name", filter:= population > 10000))` → 현재 피처에 담겨 있으며 인구 10,000 명을 초과하는'region'의 이름들을 정렬한 배열 • `overlay_contains(layer:='regions', expression:=geom_to_wkt($geometry), limit:=2)` → 현재 피처에 담겨 있는 최대 2 개까지의'region'에 대한, (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]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **layer** - 중첩을 검증할 레이어 • **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. • **filter** - 검증할 대상 피처를 필터링할 선택적인 표현식입니다. 이 표현식을 설정하지 않은 경우, 모든 피처를 검증할 것입니다. • **limit** - 매칭하는 피처의 개수를 제한할 선택적인 정수입니다. 이 파라미터를 설정하지 않은 경우, 매칭하는 모든 피처를 반환할 것입니다. • **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')` → 현재 피처가'region'을 공간 교차하는 경우 참을 반환합니다. • `overlay_crosses('regions', filter:= population > 10000)` → 현재 피처가 인구 10,000 명을 초과하는'region'을 공간 교차하는 경우 참을 반환합니다. • `overlay_crosses('regions', name)` → 현재 피처를 공간 교차하는'region'의 이름들의 배열 • `array_to_string(overlay_crosses('regions', name))` → 현재 피처를 공간 교차하는'region'의 이름들을 쉼표로 구분한 목록인 문자열 • `array_sort(overlay_crosses(layer:='regions', expression:="name", filter:= population > 10000))` → 현재 피처를 공간 교차하며 인구 10,000 명을 초과하는'region'의 이름들을 정렬한 배열 • `overlay_crosses(layer:='regions', expression:= geom_to_wkt($geometry), limit:=2)` → 현재 피처를 공간 교차하는 최대 2 개까지의'region'에 대한, (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]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **layer** - 중첩을 검증할 레이어 • **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. • **filter** - 검증할 대상 피처를 필터링할 선택적인 표현식입니다. 이 표현식을 설정하지 않은 경우, 모든 피처를 검증할 것입니다. • **limit** - 매칭하는 피처의 개수를 제한할 선택적인 정수입니다. 이 파라미터를 설정하지 않은 경우, 매칭하는 모든 피처를 반환할 것입니다. • **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')` → 현재 피처가 모든'region'과 공간적으로 분리되어 있는 경우 참을 반환합니다. • `overlay_disjoint('regions', filter:= population > 10000)` → 현재 피처가 인구 10,000 명을 초과하는 모든'region'과 공간적으로 분리되어 있는 경우 참을 반환합니다. • `overlay_disjoint('regions', name)` → 현재 피처와 공간적으로 분리되어 있는'region'의 이름들의 배열 • `array_to_string(overlay_disjoint('regions', name))` → 현재 피처와 공간적으로 분리되어 있는'region'의 이름들을 쉼표로 구분한 목록인 문자열 • `array_sort(overlay_disjoint(layer:='regions', expression:="name", filter:= population > 10000))` → 현재 피처와 공간적으로 분리되어 있으며 인구 10,000 명을 초과하는'region'의 이름들을 정렬한 배열 • `overlay_disjoint(layer:='regions', expression:= geom_to_wkt($geometry), limit:=2)` → 현재 피처와 공간적으로 분리되어 있는 최대 2 개까지의'region'에 대한, (WKT 서식) 도형들의 배열

더 읽어볼 거리: *disjoint*, 배열 조작, 위치로 선택하기 알고리즘

overlay_equals

현재 피처가 대상 레이어의 피처와 적어도 하나 이상 공간적으로 동등한지 여부, 또는 현재 피처와 공간적으로 동등한 대상 레이어의 피처에 대한 표현식 기반 결과물의 배열을 반환합니다.

기저 GEOS "Equals"서술부에 대해 자세히 알고 싶다면, PostGIS ST_Equals 함수의 설명을 읽어보세요.

문법	overlay_equals(layer, [expression], [filter], [limit], [cache=false]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **layer** - 중첩을 검증할 레이어 • **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. • **filter** - 검증할 대상 피처를 필터링할 선택적인 표현식입니다. 이 표현식을 설정하지 않은 경우, 모든 피처를 검증할 것입니다. • **limit** - 매칭하는 피처의 개수를 제한할 선택적인 정수입니다. 이 파라미터를 설정하지 않은 경우, 매칭하는 모든 피처를 반환할 것입니다. • **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')` → 현재 피처가'region'과 공간적으로 동등한 경우 참을 반환합니다. • `overlay_equals('regions', filter:= population > 10000)` → 현재 피처가 인구 10,000 명을 초과하는'region'과 공간적으로 동등한 경우 참을 반환합니다. • `overlay_equals('regions', name)` → 현재 피처와 공간적으로 동등한'region'의 이름들의 배열 • `array_to_string(overlay_equals('regions', name))` → 현재 피처와 공간적으로 동등한'region'의 이름들을 쉼표로 구분한 목록인 문자열 • `array_sort(overlay_equals(layer:='regions', expression:="name", filter:= population > 10000))` → 현재 피처와 공간적으로 동등하며 인구 10,000 명을 초과하는'region'의 이름들을 정렬한 배열 • `overlay_equals(layer:='regions', expression:= geom_to_wkt($geometry), limit:=2)` → 현재 피처와 공간적으로 동등한 최대 2 개까지의'region'에 대한, (WKT 서식) 도형들의 배열

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]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **layer** - 중첩을 검증할 레이어 • **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. • **filter** - 검증할 대상 피처를 필터링할 선택적인 표현식입니다. 이 표현식을 설정하지 않은 경우, 모든 피처를 검증할 것입니다. • **limit** - 매칭하는 피처의 개수를 제한할 선택적인 정수입니다. 이 파라미터를 설정하지 않은 경우, 매칭하는 모든 피처를 반환할 것입니다. • **cache** - 이 파라미터를 참으로 설정하면 로컬 공간 인덱스를 작성합니다. (대부분의 경우 바람직하지 않습니다. 사용자가 현저하게 느린 데이터 제공자와 작업하는 게 아니라면 말이죠.)
예제	• `overlay_intersects('regions')` → 현재 피처가'region'을 공간적으로 교차하는 경우 참을 반환합니다. • `overlay_intersects('regions', filter:= population > 10000)` → 현재 피처가 인구 10,000 명을 초과하는'region'을 공간적으로 교차하는 경우 참을 반환합니다. • `overlay_intersects('regions', name)` → 현재 피처를 교차하는'region'의 이름들의 배열 • `array_to_string(overlay_intersects('regions', name))` → 현재 피처를 교차하는'region'의 이름들을 쉼표로 구분한 목록인 문자열 • `array_sort(overlay_intersects(layer:='regions', expression:="name", filter:= population > 10000))` → 현재 피처를 교차하며 인구 10,000 명을 초과하는'region'의 이름들을 정렬한 배열 • `overlay_intersects(layer:='regions', expression:=geom_to_wkt($geometry), limit:=2)` → 현재 피처를 교차하는 최대 2 개까지의'region'에 대한, (WKT 서식) 도형들의 배열

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]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **layer** - 대상 레이어 • **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. • **filter** - 검증할 대상 피처를 필터링할 선택적인 표현식입니다. 이 표현식을 설정하지 않은 경우, 대상 레이어에 있는 모든 피처를 검증할 것입니다. • **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. • **max_distance** - 매칭하는 피처를 검색할 거리를 제한하는 선택적인 파라미터입니다. 설정하지 않은 경우, 대상 레이어에 있는 모든 피처를 사용할 것입니다. • **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')` → "airports"레이어에 매칭하는 피처가 최소한 하나 이상 있는 경우 참을 반환합니다. • `overlay_nearest('airports', max_distance:= 5000)` → 현재 피처에서 5,000 맵 단위 거리 안에'airport'레이어의 피처가 있는 경우 참을 반환합니다. • `overlay_nearest('airports', name)` → 현재 피처에 가장 가까이 있는'airport' 레이어의 피처 이름을 배열로 반환합니다. • `array_to_string(overlay_nearest('airports', name))` → 현재 피처에 가장 가까이 있는'airport'레이어의 피처 이름을 문자열로 반환합니다. • `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. • `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. • `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]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **layer** - 중첩을 검증할 레이어 • **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. • **filter** - 검증할 대상 피처를 필터링할 선택적인 표현식입니다. 이 표현식을 설정하지 않은 경우, 모든 피처를 검증할 것입니다. • **limit** - 매칭하는 피처의 개수를 제한할 선택적인 정수입니다. 이 파라미터를 설정하지 않은 경우, 매칭하는 모든 피처를 반환할 것입니다. • **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')` → 현재 피처가'region'을 공간적으로 접하는 경우 참을 반환합니다. • `overlay_touches('regions', filter:= population > 10000)` → 현재 피처가 인구 10,000 명을 초과하는'region'을 공간적으로 접하는 경우 참을 반환합니다. • `overlay_touches('regions', name)` → 현재 피처를 접하고 있는'region'의 이름들의 배열 • `string_to_array(overlay_touches('regions', name))` → 현재 피처를 접하고 있는'region'의 이름들을 쉼표로 구분한 목록인 문자열 • `array_sort(overlay_touches(layer:='regions', expression:="name", filter:= population > 10000))` → 현재 피처를 접하며 인구 10,000 명을 초과하는'region'의 이름들을 정렬한 배열 • `overlay_touches(layer:='regions', expression:=geom_to_wkt($geometry), limit:=2)` → 현재 피처를 접하는 최대 2 개까지의 'region'에 대한, (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]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **layer** - 중첩을 검증할 레이어 • **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. • **filter** - 검증할 대상 피처를 필터링할 선택적인 표현식입니다. 이 표현식을 설정하지 않은 경우, 모든 피처를 검증할 것입니다. • **limit** - 매칭하는 피처의 개수를 제한할 선택적인 정수입니다. 이 파라미터를 설정하지 않은 경우, 매칭하는 모든 피처를 반환할 것입니다. • **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')` → 현재 피처가 공간적으로'region'의 피처 내부에 있는 경우 참을 반환합니다. • `overlay_within('regions', filter:= population > 10000)` → 현재 피처가 공간적으로 인구 10,000 명을 초과하는'region'내부에 있는 경우 참을 반환합니다. • `overlay_within('regions', name)` → 현재 피처를 담고 있는'region'의 이름들의 배열 • `array_to_string(overlay_within('regions', name))` → 현재 피처를 담고 있는'region'의 이름들을 쉼표로 구분한 목록인 문자열 • `array_sort(overlay_within(layer:='regions', expression:="name", filter:= population > 10000))` → 현재 피처를 담고 있으며 인구 10,000 명을 초과하는'region'의 이름들을 정렬한 배열 • `overlay_within(layer:='regions', expression:=geom_to_wkt($geometry), limit:=2)` → 현재 피처를 담고 있는 최대 2 개까지의'region'에 대한, (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 참조 시스템 (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 • **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 • **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)'

더 읽어볼 거리: 도달불능극 알고리즘

project

라디안 단위 거리, 방향 (방위각), 그리고 표고를 이용해서 시작점에서부터 투영된 포인트를 반환합니다.

문법	project(point, distance, azimuth, [elevation]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **point** - 시작점 • **distance** - 투영할 거리 • **azimuth** - 0 이 진북에 해당하는 시계 방향 라디안 단위 방위각 • **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 • **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 • **geometry** - 도형 • **pattern** - 매칭시킬 DE-9IM 패턴
예제	• `relate( geom_from_wkt( 'LINESTRING(40 40,120 120)' ), geom_from_wkt( 'LINESTRING(40 40,60 120)' ), '**1F001**' )` → 참

reverse

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)'

더 읽어볼 거리: 라인 방향 반전시키기 알고리즘

rotate

도형의 기울인 (rotated) 버전을 반환합니다. 해당 도형의 공간 참조 시스템에서 계산합니다.

문법	rotate(geometry, rotation, [center]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **geometry** - 도형 • **rotation** - 시계 방향 도 단위 회전 • **center** - 회전 중심 포인트. 이를 지정하지 않으면, 도형 경계 상자의 중심을 사용합니다.
예제	• `rotate($geometry, 45, make_point(4, 5))` → 도형을 (4, 5) 포인트 기준으로 시계 방향 45 도 회전시킵니다. • `rotate($geometry, 45)` → 도형을 도형 경계 상자의 중심을 기준으로 시계 방향 45 도 회전시킵니다.

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

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 • **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 • **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)'

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 • **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 참조 시스템의 단위입니다.

문법	single_sided_buffer(geometry, distance, [segments=8], [join=1], [miter_limit=2.0]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **geometry** - (멀티) 라인스트링 도형 • **distance** - 버퍼 거리. 양의 값이면 라인 좌측으로 버퍼를 적용하고, 음의 값이면 라인 우측으로 적용할 것입니다. • **segments** - 둥근 결합 스타일을 사용한 경우 사분원을 나타내는 데 사용할 선분의 개수입니다. 숫자가 클수록 더 많은 노드를 가진 매끈한 버퍼를 생성합니다. • **join** - 모서리 부분의 결합 스타일. 1 = 둥글게, 2 = 마이터, 그리고 3 = 비스듬하게 • **miter_limit** - 매우 뾰족한 모서리에 사용된 마이터 비율에 대한 제한 (마이터 결합만 사용하는 경우)
예제	• `single_sided_buffer($geometry, 10.5)` → 라인 왼쪽으로 10.5 단위만큼 버퍼 생성 • `single_sided_buffer($geometry, -10.5)` → 라인 오른쪽으로 10.5 단위만큼 버퍼 생성 • `single_sided_buffer($geometry, 10.5, segments=16, join=1)` → 라인 왼쪽으로 10.5 단위만큼 버퍼 생성. 선분을 더 많이 사용할수록 더 매끈한 곡선을 생성합니다. • `single_sided_buffer($geometry, 10.5, join=3)` → 비스듬한 결합 스타일을 사용해서 라인 왼쪽으로 10.5 단위만큼 버퍼 생성

더 읽어볼 거리: 한쪽 버퍼 생성하기 알고리즘

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]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **geometry** - 도형 • **iterations** - 반복해서 적용할 평활화 횟수입니다. 숫자가 클수록 더 매끄럽지만 더 복잡한 도형을 생성합니다. • **offset** - 평활화된 도형이 얼마나 엄격하게 원본 도형을 따를지 제어하는 0 에서 0.5 사이의 값입니다. 값이 적을수록 더 평활화되고, 값이 클수록 덜 평활화됩니다. • **min_length** - 평활화 작업에 적용할 최소 선분 길이입니다. 도형의 짧은 선분에 추가 노드를 과도하게 배치하는 일을 피하려면 이 파라미터를 사용하면 됩니다. • **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)'

더 읽어볼 거리: 평탄화 알고리즘

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: 특정 꼭짓점 추출하기 알고리즘

sym_difference

두 도형이 교차하지 않는 부분들을 표현하는 도형을 반환합니다.

문법	sym_difference(geometry1, geometry2)
인자	• **geometry1** - 도형 • **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]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **geometry** - 입력 도형. 반드시 (멀티) 라인 도형이어야만 합니다. • **start_width** - 라인 시작점에서의 버퍼 너비 • **end_width** - 라인 종단점에서의 버퍼 너비 • **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

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 • **geometry2** - 도형
예제	• `touches( geom_from_wkt( 'LINESTRING(5 3, 4 4)' ), geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ) )` → 참 • `touches( geom_from_wkt( 'POINT(4 4)' ), geom_from_wkt( 'POINT(5 5)' ) )` → 거짓

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 • **source_auth_id** - 원본 인증 좌표계 ID • **dest_auth_id** - 대상 인증 좌표계 ID
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 변환 (translated) 버전을 반환합니다. 해당 도형의 공간 참조 시스템 상에서 계산합니다.

문법	translate(geometry, dx, dy)
인자	• **geometry** - 도형 • **dx** - Δx • **dy** - Δy
예제	• `translate($geometry, 5, 10)` → 원래 도형과 동일한 유형의 도형

더 읽어볼 거리: 이동시키기 *(Translate)* 알고리즘

union

도형들의 모든 포인트를 통합한 집합을 표현하는 도형을 반환합니다.

문법	union(geometry1, geometry2)
인자	• **geometry1** - 도형 • **geometry2** - 도형
예제	• `geom_to_wkt( union( make_point(4, 4), make_point(5, 5) ) )` → 'MULTIPOINT(4 4, 5 5)'

wedge_buffer

포인트 도형에서 나온 쐐기 모양의 버퍼를 반환합니다.

문법	wedge_buffer(center, azimuth, width, outer_radius, [inner_radius=0.0]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **center** - 버퍼의 중심 포인트 (원점). 포인트 도형이어야만 합니다. • **azimuth** - 쐐기의 중선에서 포인트까지의 (도 단위) 각도 • **width** - 버퍼의 (도 단위) 너비. 쐐기는 방위각 방향의 양쪽으로 각도 너비의 반씩 확장될 것입니다. • **outer_radius** - 버퍼 외부 반경 • **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

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 • **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))' ) )` → 참 • `within( geom_from_wkt( 'POINT( 5 5 )' ), geom_from_wkt( 'POLYGON((0 0, 0 1, 1 1, 1 0, 0 0 ))' ) )` → 거짓

더 읽어볼 거리: *overlay_within*

$x

현재 포인트 피처의 X 좌표를 반환합니다. 피처가 다중 부분 피처인 경우, 첫 번째 포인트의 X 좌표를 반환할 것입니다.

문법	$x
예제	• `$x` → 42

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 • `x( $geometry )` → 현재 객체 중심점의 X 좌표

$x_at

현재 객체 도형의 X 좌표를 추출합니다.

문법	$x_at(i)
인자	• **i** - 라인의 포인트의 인덱스 (0 부터 인덱스 시작; 음의 값은 -1 부터 시작하며 마지막 인덱스부터 적용)
예제	• `$x_at(1)` → 5

x_max

도형의 최대 X 좌표를 반환합니다. 해당 도형의 공간 참조 시스템 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 참조 시스템 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 다중 부분 피처인 경우, 첫 번째 포인트의 Y 좌표를 반환할 것입니다.

문법	$y
예제	• `$y` → 42

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 • `y( $geometry )` → 현재 객체 중심점의 Y 좌표

$y_at

현재 객체 도형의 Y 좌표를 추출합니다.

문법	$y_at(i)
인자	• **i** - 라인의 포인트의 인덱스 (0 부터 인덱스 시작; 음의 값은 -1 부터 시작하며 마지막 인덱스부터 적용)
예제	• `$y_at(1)` → 2

y_max

도형의 최대 Y 좌표를 반환합니다. 해당 도형의 공간 참조 시스템 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 참조 시스템 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

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_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 • `z_max( geom_from_wkt( 'MULTIPOINT ( 0 0 1 , 1 1 3 )' ) )` → 3 • `z_max( make_line( make_point( 0,0,0 ), make_point( -1,-1,-2 ) ) )` → 0 • `z_max( geom_from_wkt( 'LINESTRING( 0 0 0, 1 0 2, 1 1 -1 )' ) )` → 2 • `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 • `z_min( geom_from_wkt( 'MULTIPOINT ( 0 0 1 , 1 1 3 )' ) )` → 1 • `z_min( make_line( make_point( 0,0,0 ), make_point( -1,-1,-2 ) ) )` → -2 • `z_min( geom_from_wkt( 'LINESTRING( 0 0 0, 1 0 2, 1 1 -1 )' ) )` → -1 • `z_min( geom_from_wkt( 'POINT ( 0 0 )' ) )` → NULL

14.3.14 조판 함수

이 그룹은 인쇄 조판기 항목 속성을 처리하는 함수를 담고 있습니다.

- *item_variables*

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

14.3.15 맵 레이어

이 그룹은 현재 프로젝트에서 사용할 수 있는 레이어 목록을 담고 있습니다. 이 함수 그룹을 사용하면 여러 레이어를 참조하는, 예를 들면집계, 속성 또는공간 쿼리를 수행하는 표현식을 쉽게 작성할 수 있습니다.

또 레이어를 처리하기 위한 몇몇 편리한 함수들도 제공합니다.

- *decode_uri*

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]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **layer** - URI 를 디코딩해야 할 레이어입니다. • **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'} • `decode_uri(@layer)` → {'layerId': NULL, 'layerName': 'layer', 'path': '/home/qgis/geopackage.gpkg'} • `decode_uri(@layer, 'path')` → 'C:\my_data\qgis\shape.shp'

14.3.16 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*
- *hstore_to_map*
- *json_to_map*
- *map*
- *map_akeys*
- *map_avals*
- *map_concat*
- *map_delete*
- *map_exist*
- *map_get*
- *map_insert*
- *map_to_hstore*
- *map_to_json*
- *to_json*

from_json

JSON 서식 문자열을 불러옵니다.

문법	from_json(string)
인자	• **string** - JSON 문자열
예제	• `from_json('{"qgis":"rocks"}')` → { 'qgis': 'rocks'} • `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'}

json_to_map

JSON 서식 문자열로부터 맵을 생성합니다.

문법	json_to_map(string)
인자	• **string** - 입력 문자열
예제	• `json_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) • **value** - 값
예제	• `map('1','one','2', 'two')` → { '1': 'one', '2': 'two'}

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 • **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 • **key** - 검색할 키
예제	• `map_exist(map('1','one','2','two'),'3')` → 거짓

map_get

지정한 키에 대응하는 맵의 값을 반환합니다. 키가 실재하지 않는 경우 NULL 을 반환합니다.

문법	map_get(map, key)
인자	• **map** - 맵 • **key** - 검색할 키
예제	• `map_get(map('1','one','2','two'),'2')` → 'two' • `map_get( item_variables('Map 0'), 'map_scale')` → 현재 인쇄 조판의 ('Map 0'항목이 있는 경우) 'Map 0'항목의 축척

map_insert

추가한 키/값을 가진 맵을 반환합니다. 키가 이미 존재하는 경우, 그 값을 무시합니다.

문법	map_insert(map, key, value)
인자	• **map** - 맵 • **key** - 추가할 키 • **value** - 추가할 값
예제	• `map_insert(map('1','one'),'3','three')` → { '1': 'one', '3': 'three'} • `map_insert(map('1','one','2','overridden'),'2','two')` → { '1': 'one', '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('qgis','rocks'))` → '"qgis"=>"rocks"'

map_to_json

맵 요소들을 JSON 서식 문자열로 병합합니다.

문법	map_to_json(map)
인자	• **map** - 입력 맵
예제	• `map_to_json(map('qgis','rocks'))` → {"qgis":"rocks"}

to_json

맵, 배열, 또는 다른 값으로부터 JSON 서식 문자열을 생성합니다.

문법	to_json(value)
인자	• **value** - 입력 값
예제	• `to_json(map('qgis','rocks'))` → {“qgis”:”rocks”} • `to_json(array(1,2,3))` → [1,2,3]

14.3.17 수학 함수

이 그룹은 (제곱근, 삼각함수 등의) 수학 함수를 담고 있습니다.

- *abs*
- *acos*
- *asin*
- *atan*
- *atan2*
- *azimuth*
- *ceil*
- *clamp*
- *cos*
- *degrees*
- *exp*
- *floor*
- *inclination*
- *ln*
- *log*
- *log10*
- *max*
- *min*
- *pi*
- *radians*
- *rand*
- *randf*
- *round*

- *scale_exp*
- *scale_linear*
- *sin*
- *sqrt*
- *tan*

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) • **dx** - X 좌표 차 (difference)
예제	• `atan2(1.0, 1.732)` → 0.523611477769969

azimuth

포인트 a 의 수직선에서 포인트 b 로 시계 방향으로 측정한 진북 기준 방위각을 라디안 단위 각도로 반환합니다.

문법	azimuth(point_a, point_b)
인자	• **point_a** - 포인트 도형 • **point_b** - 포인트 도형
예제	• `degrees( azimuth( make_point(25, 45), make_point(75, 100) ) )` → 42.273689 • `degrees( azimuth( make_point(75, 100), make_point(25,45) ) )` → 222.273689

ceil

숫자를 올림합니다.

문법	ceil(value)
인자	• **value** - 숫자
예제	• `ceil(4.9)` → 5 • `ceil(-4.9)` → -4

clamp

입력값을 지정한 범위로 제한합니다.

문법	clamp(minimum, input, maximum)
인자	• **minimum** - 입력 받을 수 있는 최소값 • **input** - 최소 및 최대 로 지정된 범위로 제한될 값 • **maximum** - 입력 받을 수 있는 최대값
예제	• `clamp(1,5,10)` → 5 *input* 이 1 과 10 사이이기 때문에, 그대로 반환됩니다. • `clamp(1,0,10)` → 1 *input* 이 최소값 1 미만이기 때문에, 함수가 1 을 반환합니다 • `clamp(1,11,10)` → 10 *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 • `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 • `floor(-4.9)` → -5

inclination

포인트 a 에서 포인트 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 • **point_b** - 포인트 도형
예제	• `inclination( make_point( 5, 10, 0 ), make_point( 5, 10, 5 ) )` → 0.0 • `inclination( make_point( 5, 10, 0 ), make_point( 5, 10, 0 ) )` → 90.0 • `inclination( make_point( 5, 10, 0 ), make_point( 50, 100, 0 ) )` → 90.0 • `inclination( make_point( 5, 10, 0 ), make_point( 5, 10, -5 ) )` → 180.0

ln

값의 자연로그를 반환합니다.

문법	ln(value)
인자	• **value** - 숫자 값
예제	• `ln(1)` → 0 • `ln(2.7182818284590452354)` → 1

log

전달된 값 및 밑 (base) 의 로그 값을 반환합니다.

문법	log(base, value)
인자	• **base** - 양수 • **value** - 양수
예제	• `log(2, 32)` → 5 • `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 • `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 • `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 • `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 • `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]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **min** - 원하는 가장 작은 난수를 나타내는 정수 • **max** - 원하는 가장 큰 난수를 나타내는 정수 • **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]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **min** - 원하는 가장 작은 난수를 나타내는 실수 • **max** - 원하는 가장 큰 난수를 나타내는 실수 • **seed** - 시드로 사용할 값
예제	• `randf(1, 10)` → 4.59258286403147

round

숫자를 소수점 이하 자릿수로 반올림합니다.

문법	round(value, [places=0]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **value** - 반올림할 실수 • **places** - 소수점 이하 자리를 반올림할 자릿수를 나타내는 선택적인 정수입니다. 음수값도 가능합니다.
예제	• `round(1234.567, 2)` → 1234.57 • `round(1234.567)` → 1235

scale_exp

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_exp(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. • **domain_min** - 입력 값이 취해야 할 가장 작은 값인 입력 범위의 최소값을 지정합니다. • **domain_max** - 입력 값이 취해야 할 가장 큰 값인 입력 범위의 최대값을 지정합니다. • **range_min** - 함수가 산출해야 할 가장 작은 값인 산출 범위의 최소값을 지정합니다. • **range_max** - 함수가 산출해야 할 가장 큰 값인 산출 범위의 최대값을 지정합니다. • **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_exp(5,0,10,0,100,2)` → 25 지수 2 를 사용해서 부드럽게 조정 (ease in) • `scale_exp(3,0,10,0,100,0.5)` → 54.772 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. • **domain_min** - 입력 값이 취해야 할 가장 작은 값인 입력 범위의 최소값을 지정합니다. • **domain_max** - 입력 값이 취해야 할 가장 큰 값인 입력 범위의 최대값을 지정합니다. • **range_min** - 함수가 산출해야 할 가장 작은 값인 산출 범위의 최소값을 지정합니다. • **range_max** - 함수가 산출해야 할 가장 큰 값인 산출 범위의 최대값을 지정합니다.
예제	• `scale_linear(5,0,10,0,100)` → 50 • `scale_linear(0.2,0,1,0,360)` → 72 0 에서 1 사이의 값을 0 에서 360 사이의 각도로 조정 • `scale_linear(1500,1000,10000,9,20)` → 9.6111111 1,000 명에서 10,000 명 사이에서 변화하는 인구를 글꼴 크기 9 에서 20 사이로 조정

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

14.3.18 연산자

이 그룹은 +, -, * 같은 연산자를 담고 있습니다. 다음에 설명하는 수학 함수들 대부분이, 입력 값 가운데 하나가 NULL 인 경우 NULL 을 반환한다는 점을 유념하십시오.

함수	설명
a + b	두 값의 덧셈 (a 더하기 b)
a - b	두 값의 뺄셈 (a 빼기 b)
a * b	두 값의 곱셈 (a 곱하기 b)
a / b	두 값의 나눗셈 (a 를 b 로 나누기)
a % b	a 를 b 로 나눈 나머지 (예: 7%2=1, 또는 7 을 2 로 나누면 3 과 나머지 1)
a ^ b	두 값의 거듭제곱 (예: 2^2=4 또는 2^3=8)
a < b	두 값을 비교해서 왼쪽 값이 오른쪽 값보다 작은 경우 1 로 평가 (a 가 b 미만)
a <= b	두 값을 비교해서 왼쪽 값이 오른쪽 값보다 작거나 같은 경우 1 로 평가 (a 가 b 이하)
a <> b	두 값을 비교해서 두 값이 같지 않은 경우 1 로 평가
a = b	두 값을 비교해서 두 값이 같은 경우 1 로 평가
a != b	a 와 b 는 같지 않음
a > b	두 값을 비교해서 왼쪽 값이 오른쪽 값보다 큰 경우 1 로 평가 (a 가 b 초과)
a >= b	두 값을 비교해서 왼쪽 값이 오른쪽 값보다 크거나 같은 경우 1 로 평가 (a 가 b 이상)
a ~ b	a 와 정규 표현식 b 가 일치
||	두 값을 문자열로 결합합니다. 한 쪽 값이 NULL 인 경우 NULL 을 반환합니다.
'\n'	문자열에 줄바꿈 삽입
LIKE	첫 번째 파라미터가 지정한 패턴과 일치하는 경우 1 을 반환

다음 페이지에 계속

표 14.1 – 이전 페이지에서 계속

함수	설명
ILIKE	첫 번째 파라미터가 대소문자를 구분하는 지정 패턴과 일치하는 경우 1 을 반환 (대소문자 상태까지 일치하는지 확인하려 할 때 LIKE 대신 ILIKE 를 쓸 수 있습니다)
a IS b	두 값이 동일한지 검증해서 a 와 b 가 동일한 경우 1 을 반환
a OR b	조건 a 또는 조건 b 가 참인 경우 1 을 반환
a AND b	조건 a 와 조건 b 가 참인 경우 1 을 반환
NOT	조건 무효화
"Column_name"	*Column_name* 필드의 값으로, 다음 작은따옴표와 혼동하지 않도록 조심하십시오.
'string'	문자열 값으로, 이전 큰따옴표와 혼동하지 않도록 조심하십시오.
NULL	NULL 값
a IS NULL	a 가 값을 가지고 있지 않음
a IS NOT NULL	a 가 값을 가지고 있음
a IN (value[,value])	a 가 값 목록에 있음
a NOT IN (value[,value])	a 가 값 목록에 있지 않음

다음은 몇몇 예시입니다:

- 문자열과 열 명칭의 값을 연결합니다:

```
'My feature''s id is: ' || "gid"
```

- "description"속성 필드의 값이'Hello'문자열로 시작하는지 검증합니다 (% 문자의 위치를 주목하세요):

```
"description" LIKE 'Hello%'
```

14.3.19 공간 처리 함수

이 그룹은 공간 처리 알고리즘을 대상으로 실행되는 함수를 담고 있습니다.

- 파라미터

파라미터

공간 처리 알고리즘의 입력 파라미터의 값을 반환합니다.

문법	parameter(name)
인자	• **name** - 대응하는 입력 파라미터의 이름
예제	• `parameter('BUFFER_SIZE')` → 5.6

14.3.20 래스터 함수

이 그룹은 래스터 레이어를 대상으로 실행되는 함수를 담고 있습니다.

- *raster_statistic*
- *raster_value*

raster_statistic

래스터 레이어의 통계를 반환합니다.

문법	raster_statistic(layer, band, property)
인자	• **layer** - 래스터 레이어 이름 또는 레이어 ID 가운데 하나를 나타내는 문자열 • **band** - 래스터 레이어의 1 로 시작하는 밴드 번호를 나타내는 정수 • **property** - 반환할 속성에 대응하는 문자열입니다. 사용할 수 있는 옵션은 다음과 같습니다: – min: 최소값 – max: 최대값 – avg: 평균값 – stdev: 값들의 표준 편차 – range: 값의 범위 (최대 - 최소) – sum: 래스터에서 나온 모든 값들의 합계
예제	• `raster_statistic('lc',1,'avg')` → 'lc'래스터 레이어의 밴드 1 의 평균값 • `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 • **band** - 값을 가져올 밴드 번호 • **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

14.3.21 레코드 및 속성 함수

이 그룹은 레코드 식별자를 대상으로 하는 함수를 담고 있습니다.

- *attribute*
- *attributes*
- *$currentfeature*
- *display_expression*
- *get_feature*
- *get_feature_by_id*
- *$id*
- *is_selected*
- *maptip*
- *num_selected*
- *represent_value*
- *sqlite_fetch_and_increment*
- *uuid*

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 • **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

평가 중인 현재 피처를 반환합니다. 현재 피처의 속성값을 평가하기 위해 `attribute` 함수와 함께 사용할 수 있습니다.

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]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **layer** - 레이어 (또는 레이어 ID 또는 이름) • **feature** - 평가해야 할 피처 • **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 • `display_expression('a_layer_id', $currentfeature, 'False')` → 지정한 피처의 평가하지 않은 표시 표현식

get_feature

지정한 속성값과 일치하는 레이어의 첫 번째 피처를 반환합니다.

문법	get_feature(layer, attribute, value)
인자	• **layer** - 레이어 이름 또는 ID • **attribute** - 속성 이름 • **value** - 일치시킬 속성값
예제	• `get_feature('streets','name','main st')` → "name"필드에"main st"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 • **feature_id** - 반환해야 할 피처의 ID
예제	• `get_feature_by_id('streets', 1)` → "streets"레이어에서 ID 가 1 인 피처

더 읽어볼 거리: *$id*

$id

현재 행의 객체 ID 를 반환합니다.

문법	$id
예제	• `$id` → 42

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()` → 현재 레이어에서 현재 피처를 선택한 경우 참

'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)` → 현재 레이어에서 선택한 피처가 활성화된 지도책의 피처인 경우 참 • `is_selected(get_feature(@layer, 'name', 'Main St.'))` → 현재 레이어에서"Main St."라는 유일한 이름을 가진 피처를 선택한 경우 참 • `is_selected(get_feature_by_id(@layer, 1))` → 현재 레이어에서 ID 가 1 인 피처를 선택한 경우 참

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) • **feature** - 선택했는지 검증해야 할 피처
예제	• `is_selected( 'streets', get_feature('streets', 'name', "street_name"))` → 현재 건물의 거리를 선택했다면 참입니다. (건물 레이어가 'street_name'이라는 필드명을 가지고 있고'streets'레이어가 유일값들을 담고 있는'name'이라는 필드를 가고 있다고 가정합니다.) • `is_selected( 'streets', get_feature_by_id('streets', 1))` → "streets"레이어에서 ID 가 1 인 피처를 선택한 경우 참

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]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **layer** - 레이어 (또는 레이어 ID 또는 이름) • **feature** - 평가해야 할 피처 • **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 • `maptip('a_layer_id', $currentfeature, 'False')` → 지정한 피처의 평가하지 않은 맵 도움말

num_selected

지정한 레이어에서 선택한 객체들의 개수를 반환합니다. 기본적으로 표현식을 평가한 레이어 상에서 작동합니다.

문법	num_selected([layer=current layer]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **layer** - 선택됐는지 검증할 레이어 (또는 레이어 ID 또는 이름)
예제	• `num_selected()` → 현재 레이어에서 선택한 피처 개수 • `num_selected('streets')` → 'streets'레이어에서 선택한 피처 개수

represent_value

필드값에 대해 구성된 표현값을 반환합니다. 구성된 위젯 유형에 따라 다릅니다. 흔히'Value Map'위젯에 유용합니다.

문법	represent_value(value, fieldName)
인자	• **value** - 해석되어야 할 값입니다. 대부분 필드일 겁니다. • **fieldName** - 위젯 환경 설정을 불러와야 할 필드 이름입니다. (선택적)
예제	• `represent_value("field_with_value_map")` → 값에 대한 설명 • `represent_value('static value', 'field_name')` → 통계값에 대한 설명

더 읽어볼 거리: 위젯 유형

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]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **database** - SQLite 파일 또는 GeoPackage 레이어를 가리키는 경로 • **table** - 시퀀스를 관리하는 테이블의 이름 • **id_field** - 현재 값을 담고 있는 필드의 이름 • **filter_attribute** - 이 시퀀스의 고유 식별자를 담고 있는 필드의 이름입니다. UNIQUE 인덱스를 가지고 있어야만 합니다. • **filter_value** - 사용할 시퀀스의 이름 • **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 • `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, 일대다 또는 다대다 관계 생성

uuid

Generates a Universally Unique Identifier (UUID) for each row using the Qt QUuid::createUuid method. Each UUID is 38 characters long.

문법	uuid()
예제	• `uuid()` → '{0bd2f60f-f157-4a6d-96af-d4ba4cb366a1}'

14.3.22 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.

14.3.23 문자열 함수

이 그룹은 문자열을 대상으로 하는 (예: 치환, 대문자 변환 등) 함수를 담고 있습니다.

- *ascii*
- *char*
- *concat*
- *format*
- *format_date*
- *format_number*
- *left*
- *length*
- *lower*
- *lpad*
- *regexp_match*
- *regexp_replace*
- *regexp_substr*
- *replace*
- *right*
- *rpad*
- *strpos*
- *substr*
- *title*
- *to_string*
- *trim*

- *upper*
- *wordwrap*

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' • `concat('a','b','c','d','e')` → 'abcde' • `concat('Anno ', 1984)` → 'Anno 1984' • `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
```

- NULL 값인 인수가 하나라도 있을 경우, `||` 또는 `+` 둘 다 NULL 값을 반환할 것입니다. NULL 값에 상관없이 다른 인수를 반환하게 하려면, `concat` 함수를 사용해야 할 수도 있습니다:

```
'My feature id is: ' + NULL ==> NULL
'My feature id is: ' || NULL => NULL
concat('My feature id is: ', NULL) => 'My feature id is: '
```

format

지정한 인자를 이용해서 문자열 서식을 변경합니다.

문법	format(string, arg1, arg2, ⋯)
인자	• **string** - 인자를 위한 대체문자 (placeholder) 를 가진 문자열입니다. 대체문자에%1, %2 등을 사용하십시오. 대체문자는 반복할 수 있습니다. • **arg** - 모든 유형, 제한 없는 개수의 인자
예제	• `format('This %1 a %2','is', 'test')` → 'This is a test'

format_date

날짜 유형 또는 문자열을 사용자 정의 문자열 서식으로 변환합니다. Qt 날짜 & 시간 서식 문자열을 사용하십시오. QDateTime::toString 을 참조하세요.

<table>
<tr><td>문법</td><td>format_date(datetime, format, [language])
[] 괄호는 부가적인 인자를 표시합니다</td></tr>
<tr><td>인자</td><td>

datetime - 날짜, 시간 또는 날짜 & 시간 값
format - 문자열을 서식화하는 데 쓰이는 문자열 템플릿

<table>
<tr><th>표현식</th><th>산출물</th></tr>
<tr><td>d</td><td>앞에 0 이 없는 숫자 일 (1 에서 31)</td></tr>
<tr><td>dd</td><td>앞에 0 이 있는 숫자 일 (01 에서 31)</td></tr>
<tr><td>ddd</td><td>축약된 현지 요일 이름 (예: 'Mon'에서'Sun')</td></tr>
<tr><td>dddd</td><td>축약하지 않은 현지 요일 이름 (예: 'Monday'에서'Sunday')</td></tr>
<tr><td>M</td><td>앞에 0 이 없는 숫자 월 (1-12)</td></tr>
<tr><td>MM</td><td>앞에 0 이 있는 숫자 월 (01-12)</td></tr>
<tr><td>MMM</td><td>축약된 현지 월 이름 (예: 'Jan'에서'Dec')</td></tr>
<tr><td>MMMM</td><td>축약하지 않은 현지 월 이름 (예: 'January'에서'December')</td></tr>
<tr><td>yy</td><td>두 자리 숫자 연도 (00-99)</td></tr>
<tr><td>yyyy</td><td>네 자리 숫자 연도</td></tr>
</table>
다음 표현식은 서식 문자열의 시간 부분에 쓰일 수도 있습니다:
<table>
<tr><th>표현식</th><th>산출물</th></tr>
<tr><td>h</td><td>앞에 0 이 없는 시간 (0 에서 23 또는 AM/PM 표시가 있는 경우 1 에서 12)</td></tr>
<tr><td>hh</td><td>앞에 0 이 있는 시간 (00 에서 23 또는 AM/PM 표시가 있는 경우 01 에서 12)</td></tr>
<tr><td>H</td><td>앞에 0 이 없는 시간 (AM/PM 표시가 있는 경우에도 0 에서 23)</td></tr>
<tr><td>HH</td><td>앞에 0 이 있는 시간 (AM/PM 표시가 있는 경우에도 00 에서 23)</td></tr>
<tr><td>m</td><td>앞에 0 이 없는 분 (0 에서 59)</td></tr>
<tr><td>mm</td><td>앞에 0 이 있는 분 (00 에서 59)</td></tr>
<tr><td>s</td><td>앞에 0 이 없는 초 (0 에서 59)</td></tr>
<tr><td>ss</td><td>앞에 0 이 있는 초 (00 에서 59)</td></tr>
<tr><td>z</td><td>뒤에 0 이 없는 밀리초 (0 에서 999)</td></tr>
<tr><td>zzz</td><td>뒤에 0 이 있는 밀리초 (000 에서 999)</td></tr>
<tr><td>AP or A</td><td>AM/PM 시간으로 해석합니다. AP 는'AM'또는'PM'이어야만 합니다.</td></tr>
<tr><td>ap or a</td><td>AM/PM 시간으로 해석합니다. ap 는'am'또는'pm'이어야만 합니다.</td></tr>
</table>

날짜 유형을 사용자 지정 문자열로 서식화하는 데 사용된 언어 (소문자, 2-3 문자, ISO 639 언어 코드)

</td></tr>
<tr><td>예제</td><td>

<code>format_date('2012-05-15','dd.MM.yyyy')</code> → '15.05.2012'
<code>format_date('2012-05-15','d MMMM yyyy','fr')</code> → '15 mai 2012'
<code>format_date('2012-05-15','dddd')</code> → 'Tuesday'
<code>format_date('2012-05-15 13:54:20','dd.MM.yy')</code> → '15.05.12'
<code>format_date('13:54:20','hh:mm AP')</code> → '01:54 PM'

</td></tr>
</table>

format_number

천 단위를 로케일 구분자로 서식화한 숫자를 반환합니다. 지정한 소수점 이하 자릿수로 숫자를 잘라 맞추기도 합니다.

문법	format_number(number, places, [language]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **number** - 서식화할 숫자 • **places** - 문자열을 잘라 맞출 십진 자릿수를 나타내는 정수입니다. • **language** - 숫자를 문자열로 서식화하는 데 사용된 언어 (소문자, 2-3 문자, ISO 639 언어 코드)
예제	• `format_number(10000000.332,2)` → '10,000,000.33' • `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 • **length** - 정수. 문자열의 왼쪽을 기준으로 반환할 문자의 개수입니다.
예제	• `left('Hello World',5)` → 'Hello'

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 참조 시스템 (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'

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 • **width** - 새 문자열의 길이 • **fill** - 남은 공백을 채울 문자
예제	• `lpad('Hello', 10, 'x')` → 'xxxxxHello' • `lpad('Hello', 3, 'x')` → 'Hel'

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 • **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 • `regexp_match('Budač','udač\\b')` → 2

regexp_replace

지정한 정규 표현식을 치환한 문자열을 반환합니다.

문법	regexp_replace(input_string, regex, replacement)
인자	• **input_string** - 일치하는 경우 치환할 문자열 • **regex** - 치환할 정규 표현식입니다. 백슬래시 문자는 이중으로 이스케이프시켜야만 합니다. (예: 공백 문자와 매칭하려면"\\s") • **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' • `regexp_replace('ABC123','\\d+','')` → 'ABC' • `regexp_replace('my name is John','(.*) is (.*)','\\2 is \\1')` → 'John is my name'

regexp_substr

문자열에서 지정한 정규 표현식과 일치하는 부분을 반환합니다.

문법	regexp_substr(input_string, regex)
인자	• **input_string** - 일치하는 경우 검색할 문자열 • **regex** - 일치시킬 정규 표현식. 백슬래시 문자는 이중으로 이스케이프시켜야만 합니다. (예: 공백 문자와 일치시키려면"\\s")
예제	• `regexp_substr('abc123','(\\d+)')` → '123'

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 • **before** - 치환할 문자열 또는 문자열 배열 • **after** - 치환에 사용할 문자열 또는 문자열 배열
예제	• `replace('QGIS SHOULD ROCK','SHOULD','DOES')` → 'QGIS DOES ROCK' • `replace('QGIS ABC',array('A','B','C'),array('X','Y','Z'))` → 'QGIS XYZ' • `replace('QGIS',array('Q','S'),'')` → 'GI'

맵 변이형

지정한 맵 키를 값의 쌍으로 치환한 문자열을 반환합니다.

문법	replace(string, map)
인자	• **string** - 입력 문자열 • **map** - 키와 값을 담고 있는 맵
예제	• `replace('APP SHOULD ROCK',map('APP','QGIS','SHOULD','DOES'))` → 'QGIS DOES ROCK'

right

문자열 가장 오른쪽에 있는 *n* 개의 문자를 담은 하위 문자열을 반환합니다.

문법	right(string, length)
인자	• **string** - 문자열 • **length** - 정수. 문자열의 오른쪽을 기준으로 반환할 문자의 개수입니다.
예제	• `right('Hello World',5)` → 'World'

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 • **width** - 새 문자열의 길이 • **fill** - 남은 공백을 채울 문자
예제	• `rpad('Hello', 10, 'x')` → 'Helloxxxxx' • `rpad('Hello', 3, 'x')` → 'Hel'

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 • **needle** - 검색할 문자열
예제	• `strpos('HELLO WORLD','WORLD')` → 7 • `strpos('HELLO WORLD','GOODBYE')` → 0

substr

문자열의 부분을 반환합니다.

문법	substr(string, start, [length]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **string** - 전체 입력 문자열 • **start** - 추출할 시작 위치를 나타내는 1 부터 시작하는 정수; 시작 위치가 음수이면 반환되는 문자열이 문자열 마지막에서 시작 값을 뺀 위치에서 시작합니다. • **length** - 추출할 문자열의 길이를 나타내는 정수. 길이가 음수이면 반환 문자열은 문자열의 끝에서 지정된 길이의 문자를 생략합니다.
예제	• `substr('HELLO WORLD',3,5)` → 'LLO W' • `substr('HELLO WORLD',6)` → 'WORLD' • `substr('HELLO WORLD',-5)` → 'WORLD' • `substr('HELLO',3,-1)` → 'LL' • `substr('HELLO WORLD',-5,2)` → 'WO' • `substr('HELLO WORLD',-5,-1)` → 'WORL'

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'

to_string

숫자를 문자열로 변환합니다.

문법	to_string(number)
인자	• **number** - 정수형 또는 실수형 값. 문자열로 변환할 숫자입니다.
예제	• `to_string(123)` → '123'

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'

upper

문자열을 대문자로 변환합니다.

문법	upper(string)
인자	• **string** - 대문자로 변환할 문자열
예제	• `upper('hello WOrld')` → 'HELLO WORLD'

wordwrap

최대/최소 문자 개수로 행갈이한 문자열을 반환합니다.

문법	wordwrap(string, wrap_length, [delimiter_string]) [] 괄호는 부가적인 인자를 표시합니다
인자	• **string** - 행갈이할 문자열 • **wrap_length** - 정수. wrap_length 가 양수인 경우 행갈이할 문자의 이상적인 최대 개수를 나타내는 숫자이고, 음수인 경우 행갈이할 문자의 최소 개수를 나타냅니다. • **delimiter_string** - 새 줄로 행갈이할 선택적인 구분자 문자열
예제	• `wordwrap('UNIVERSITY OF QGIS',13)` → 'UNIVERSITY OF QGIS' • `wordwrap('UNIVERSITY OF QGIS',-3)` → 'UNIVERSITY OF QGIS'

14.3.24 사용자 표현식

이 그룹은사용자 표현식 으로 저장된 표현식들을 담고 있습니다.

14.3.25 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 지도 피처 (피처 객체로써)
atlas_featureid	현재 지도 피처 ID
atlas_featurenumber	현재 조판기에 있는 지도 피처의 번호
atlas_filename	현재 지도 파일명
atlas_geometry	현재 지도 피처 도형
atlas_layerid	현재 지도 커버리지 레이어 ID
atlas_layername	현재 지도 커버리지 레이어명
atlas_pagename	현재 지도 페이지 명칭
atlas_totalfeatures	지도에 있는 피처의 총 개수
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.
frame_duration	각 애니메이션 프레임의 시계열 기간 (간격 값으로)
frame_number	애니메이션 재생 시 현재 프레임 번호
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_part_count	렌더링된 피처의 도형에 있는 부분의 개수

다음 페이지에 계속

표 14.2 – 이전 페이지에서 계속

변수	설명
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
grid_axis	현재 그리드 주석 축 (예: 경도는'x', 위도는'y')
grid_number	현재 그리드 주석 값
item_id	조판기 항목 사용자 ID (유일해야 하는 것은 아닙니다)
item_uuid	조판기 항목 유일 ID
layer	현재 레이어
layer_id	현재 레이어의 ID
layer_ids	현재 프로젝트에 있는 모든 맵 레이어의 ID 목록
layer_name	현재 레이어의 명칭
layers	현재 프로젝트에 있는 모든 맵 레이어의 목록
layout_dpi	조판 해상도 (DPI)
layout_name	조판 명칭
layout_numpages	조판한 페이지 수
layout_page	조판기에 있는 현재 항목의 페이지 번호
layout_pageheight	조판기에서 활성화된 페이지의 높이 (밀리미터 단위)
layout_pagewidth	조판기에서 활성화된 페이지의 너비 (밀리미터 단위)
legend_column_count	범례에 있는 열의 개수
legend_filter_by_map	맵이 범례 내용을 필터링했는지 여부를 표시
legend_filter_out_atlas	범례에서 지도를 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
map_crs_definition	현재 맵 좌표계의 완전한 정의
map_crs_description	현재 맵 좌표계의 명칭
map_crs_ellipsoid	현재 맵 좌표계 타원체의 약어
map_crs_proj4	현재 맵 좌표계의 PROJ4 정의
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
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

다음 페이지에 계속

표 14.2 – 이전 페이지에서 계속

변수	설명
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.
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 파일명의 (경로 및 확장자를 제외한) 기본 명칭
project_creation_date	프로젝트 생성 날짜. 프로젝트 메타데이터에서 가져옵니다.
project_crs	프로젝트 좌표계
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.
project_ellipsoid	현재 프로젝트의 타원체 명칭. 측지 (geodetic) 면적 또는 도형 길이 계산 시 사용됩니다.
project_filename	현재 프로젝트의 파일명
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 (파일명을 포함한) 전체 경로
project_title	현재 프로젝트의 제목
project_units	프로젝트 좌표계의 단위
qgis_locale	QGIS 의 현재 언어
qgis_os_name	현재 운영체제의 명칭을 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
symbol_angle	피처를 렌더링하는 데 쓰이는 심볼의 각도 (마커 심볼의 경우에만 유효합니다.)
symbol_color	피처를 렌더링하는 데 쓰이는 심볼의 색상
symbol_count	(조판기 범례에 표시되는) 심볼이 표현하는 피처의 개수
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 유효합니다)

다음 페이지에 계속

표 14.2 – 이전 페이지에서 계속

변수	설명
symbol_marker_row	마커에 대한 행 번호 (포인트 패턴 채우기에 대해서만 유효합니다)
user_account_name	현재 사용자의 운영체제 계정명
user_full_name	현재 사용자의 운영체제 사용자명
value	현재 값
with_variable	표현식 내부에서 활용하기 위한 변수를 설정할 수 있고, 동일한 값을 반복해서 재계산하는 일을 피할 수 있습니다.
zoom_level	렌더링 중인 타일의 (현재 맵 축척에서 파생된) 확대/축소 수준입니다. 일반적으로 [0, 20] 간격 단위입니다.

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'
  )
)
```

14.3.26 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.

14.4 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.

14.4.1 서문: 공간 및 비공간 테이블

QGIS 는 공간 및 비공간 레이어를 불러올 수 있습니다. 현재 PostgreSQL, MSSQL, SpatiaLite, DB2 및 Oracle 제공자는 물론, 구분 텍스트와 OGR 이 지원하는 테이블도 포함합니다. 레이어를 불러오면, *Layers* 패널에 있는 목록에 레이어가 추가됩니다. 레이어가 공간적으로 활성화됐는지 아닌지 여부가 사용자가 맵 상에서 레이어와 대화형 작업을 할 수 있는지 없는지 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.

14.4.2 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 14.68 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.

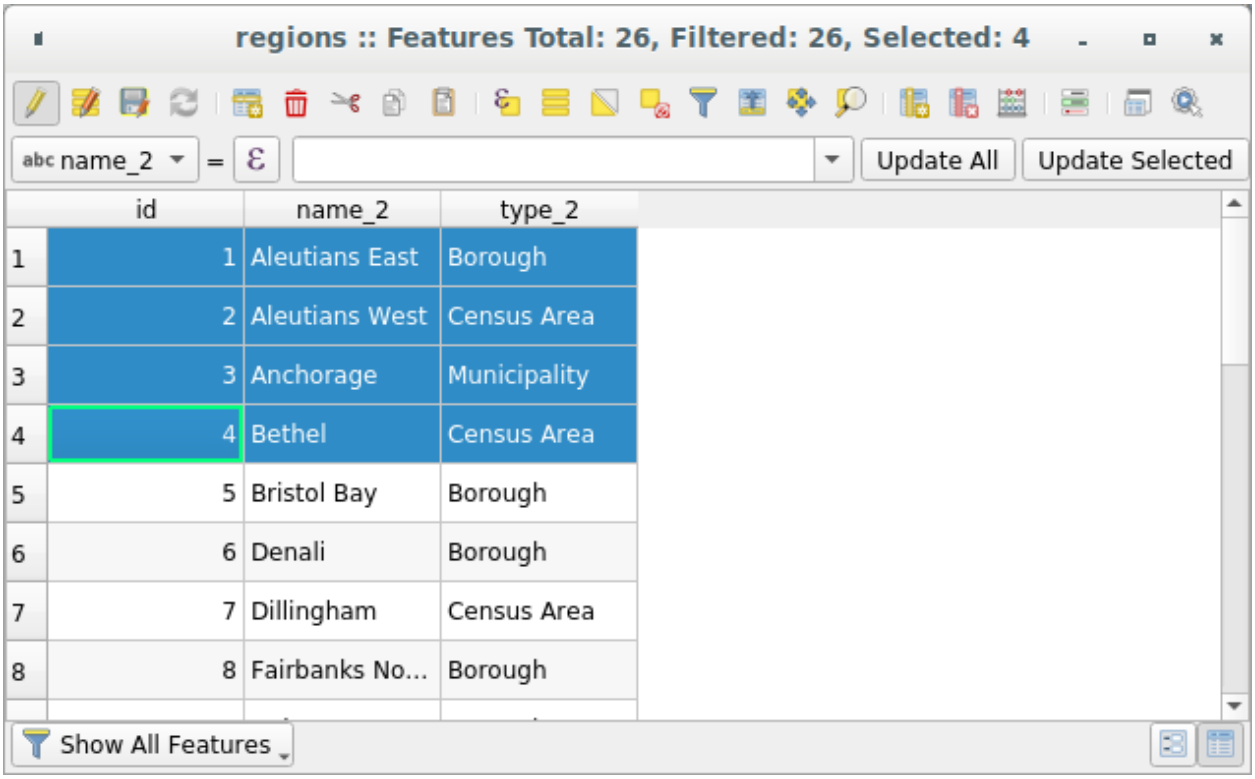


그림 14.68: "regions"레이어의 속성 테이블

속성 테이블 창 상단에 있는 아이콘들은 다음과 같은 기능을 제공합니다:

표 14.3: 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
	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	
	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	

참고: 데이터 유형 및 사용자의 QGIS 버전과 함께 빌드한 OGR 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: 각 행이 피처를 나타내고 각 열이 필드를 나타내는 표 서식으로 여러 피처의 값을 표시합니다.
-  Form view: 첫 번째 패널에피처 식별자 를 표시하고 두 번째 패널에는 클릭한 식별자의 속성만 표시합니다. 첫 번째 패널 상단에 있는 풀다운 메뉴에서 속성 (*Column preview*) 또는 표현식 (*Expression*) 을 사용해서 "식별자 (identifier)"를 지정할 수 있습니다. 이 풀다운 메뉴는 최근 사용한 표현식 10 개도 재사용하기 위해 포함하고 있습니다. 양식 뷰는 레이어 필드 환경 설정 (속성 양식 속성 참조) 을 이용합니다. 첫 번째 패널 하단에 있는 화살표를 가지고 피처 식별자를 탐색할 수 있습니다. 목록에서 피처를 노란색 마커로 선택하면, 캔버스 상에서도 해당 피처가 노란색으로 선택됩니다. 속성 테이블 상단에 있는  버튼을 클릭하면 해당 피처로 확대/축소합니다. (체크 사각형을 사용하지 않고) 목록에 있는 항목을 클릭하면 해당 피처가 어디에 있는지 알 수 있도록 빨간색으로 한번 번쩍입니다.

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.

열 환경 설정하기

테이블 뷰의 경우, 속성 테이블에 무엇을 어떻게 표시할지에 대한 환경을 설정할 수 있는 도구에 접근하려면 열 헤더를 오른쪽 클릭하십시오.

열 숨기기와 정리하기 및 액션 활성화하기

열 헤더를 오른쪽 클릭하면, 속성 테이블에서 열을 숨기는 옵션을 선택할 수 있습니다. 열을 숨김 해제하거나 열의 순서를 변경하는 등 한 번에 여러 열들의 습성을 변경하려면, *Organize columns* …메뉴 옵션을 선택하십시오. 새 대화창에서 다음 작업을 할 수 있습니다:

- 표시하거나 숨기려하는 열을 체크/체크 해제할 수 있습니다.
- 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 액션의 드롭다운 상자 또는 버튼 목록을 표시하는 새로운 가상 *Actions* 열을 활성화할 수 있습니다. 액션에 관한 자세한 정보는액션 속성 을 참조하세요.

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.
- *Autosize*: 열에 가장 알맞은 너비로 조정할 수 있습니다.

열 헤더의 오른쪽 경계선을 드래그해도 너비를 변경할 수 있습니다. 레이어는 열의 새로운 크기를 유지하기 때문에, 다음에 속성 테이블을 열어도 변경한 너비가 복구됩니다.

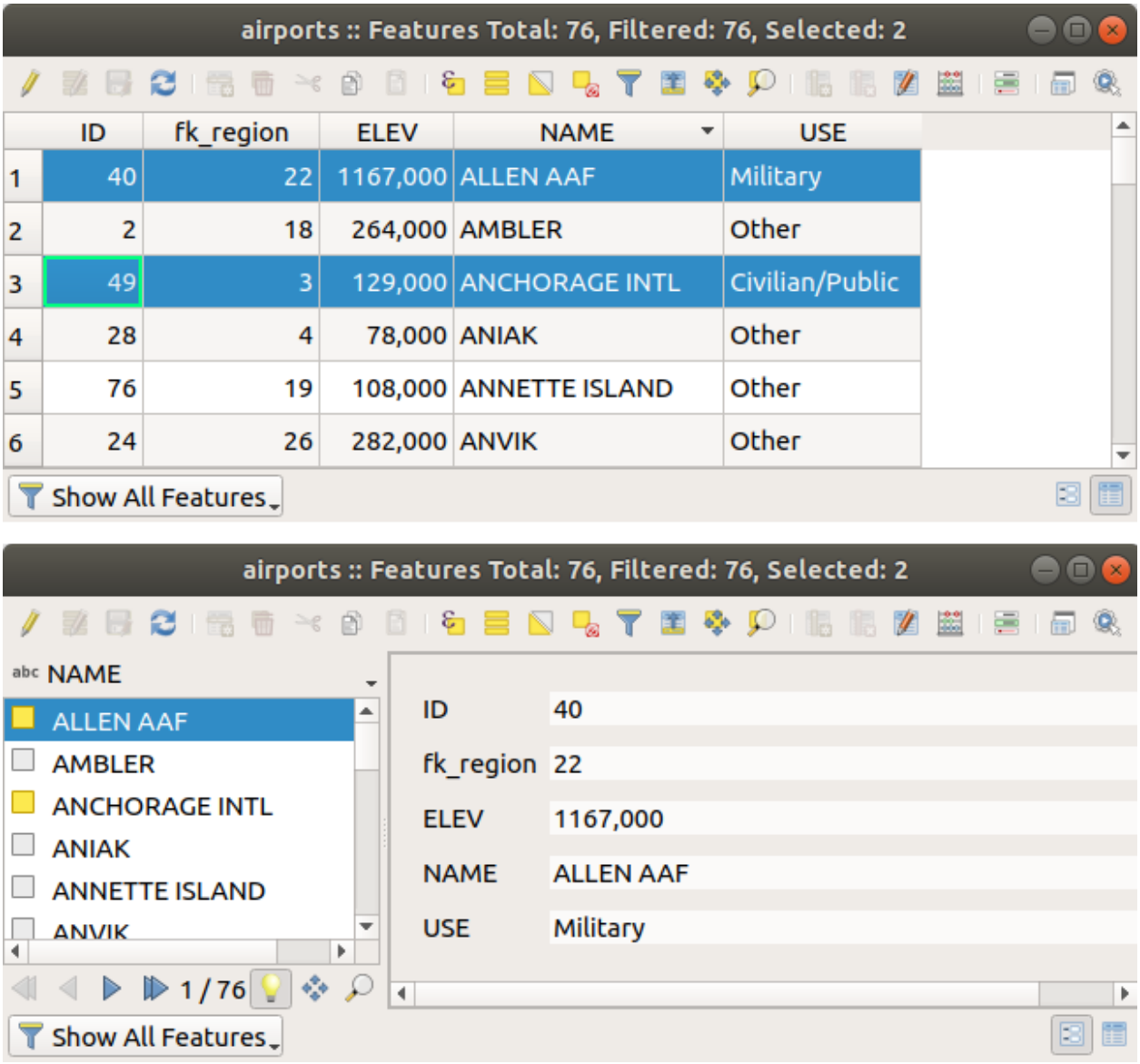


그림 14.69: 속성 테이블의 테이블 뷰 (위) 대 양식 뷰 (아래)

열 정렬하기

열 헤더를 클릭하면 어느 열 기준으로도 테이블을 정렬할 수 있습니다. 작은 화살표가 정렬 순서를 나타냅니다. (아래쪽 화살표는 맨 위 행에서 아래 행 방향으로 내림차순, 위쪽 화살표는 맨 위 행에서 아래 행 방향으로 오름차순을 의미합니다.) 열 헤더 컨텍스트 메뉴의 *Sort* 옵션을 선택한 다음 표현식을 입력해서 행들을 정렬하도록 선택할 수도 있습니다. 예를 들어 여러 열을 이용해서 행을 정렬하려면 `concat(col0, col1)` 같은 표현식을 작성하면 됩니다.

양식 뷰에서는, *Sort by preview expression* 옵션을 통해 피처 식별자를 정렬할 수 있습니다.

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 등등)

테이블 뷰의 (양식 뷰에서는 사용할 수 없습니다) 속성 창 우상단에 있는 아이콘을 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 명칭
- 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

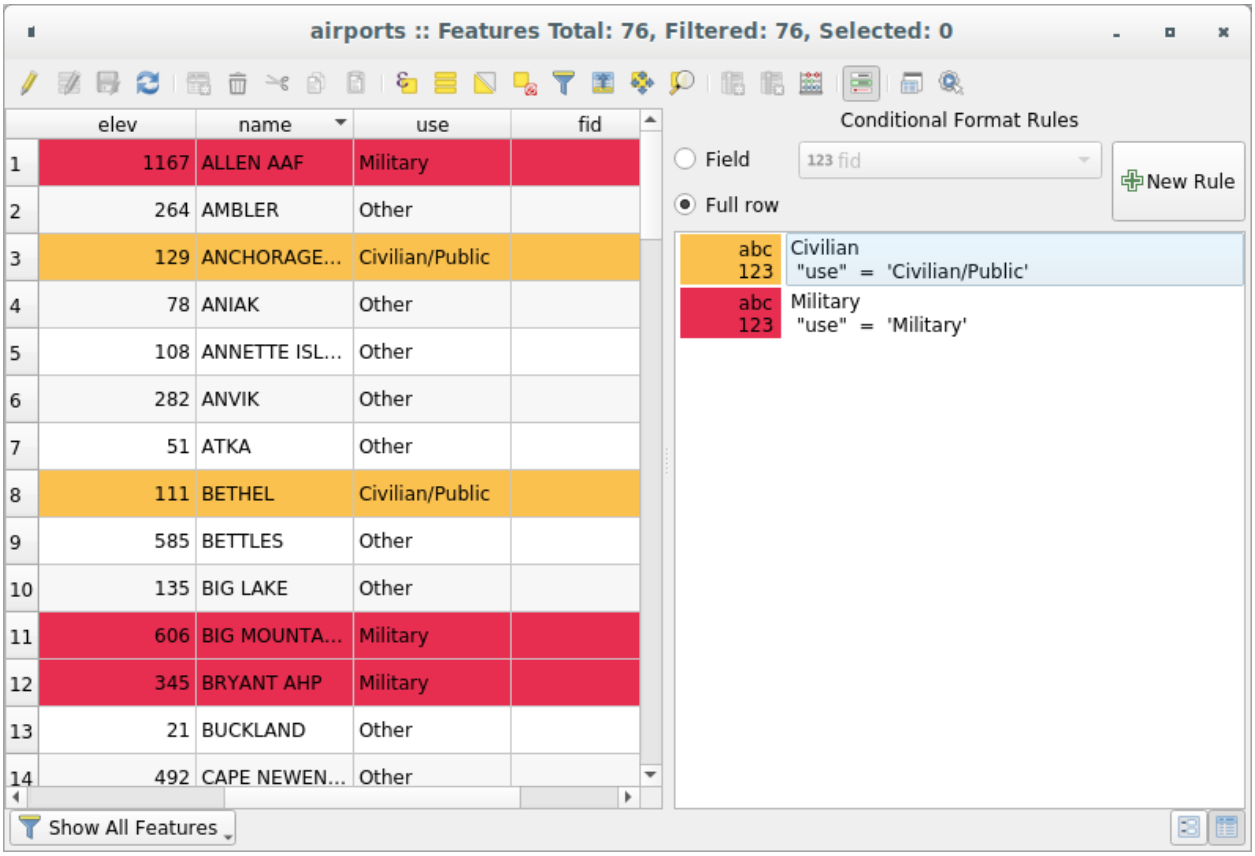


그림 14.70: 속성 테이블의 조건부 서식

14.4.3 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 이용해서 피처 필터링하기/선택하기 를 이용해서 피처를 선택할 수도 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*
- *Show Features visible on map*
- *Show Edited and New Features*
- *Field Filter*: 필드의 값을 기반으로 필터링할 수 있습니다. 목록에서 열을 선택해서 값을 입력한 다음 `Enter` 키를 눌러 필터링합니다. 그러면 속성 테이블에 일치하는 피처만 표시될 것입니다.
- *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.

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.

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.

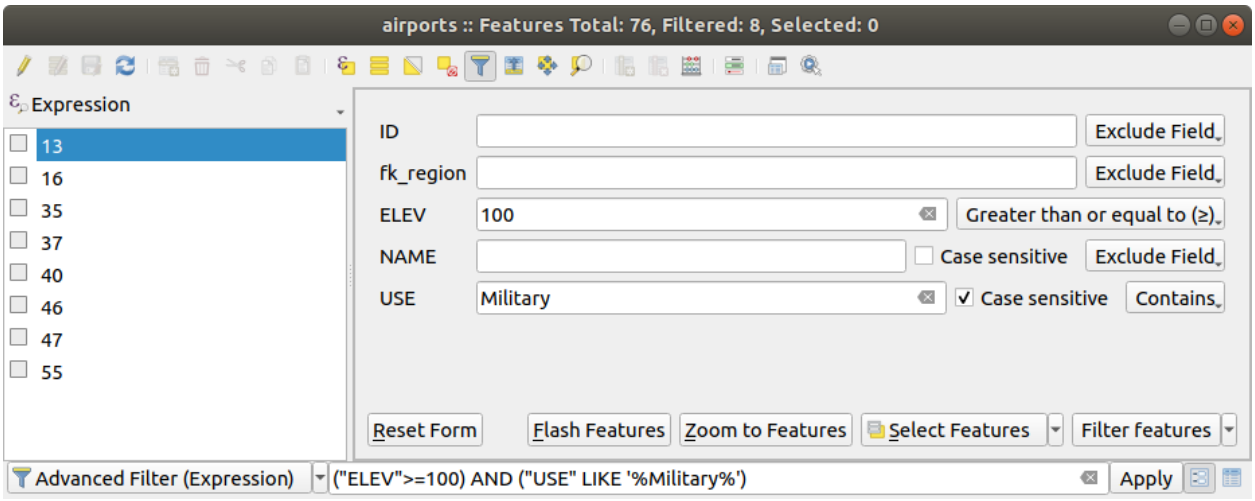


그림 14.71: 필터 양식으로 필터링한 속성 테이블

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.

14.4.4 피처에 대한 액션 사용

컨텍스트 메뉴를 통해 피처를 다음과 같이 처리할 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.

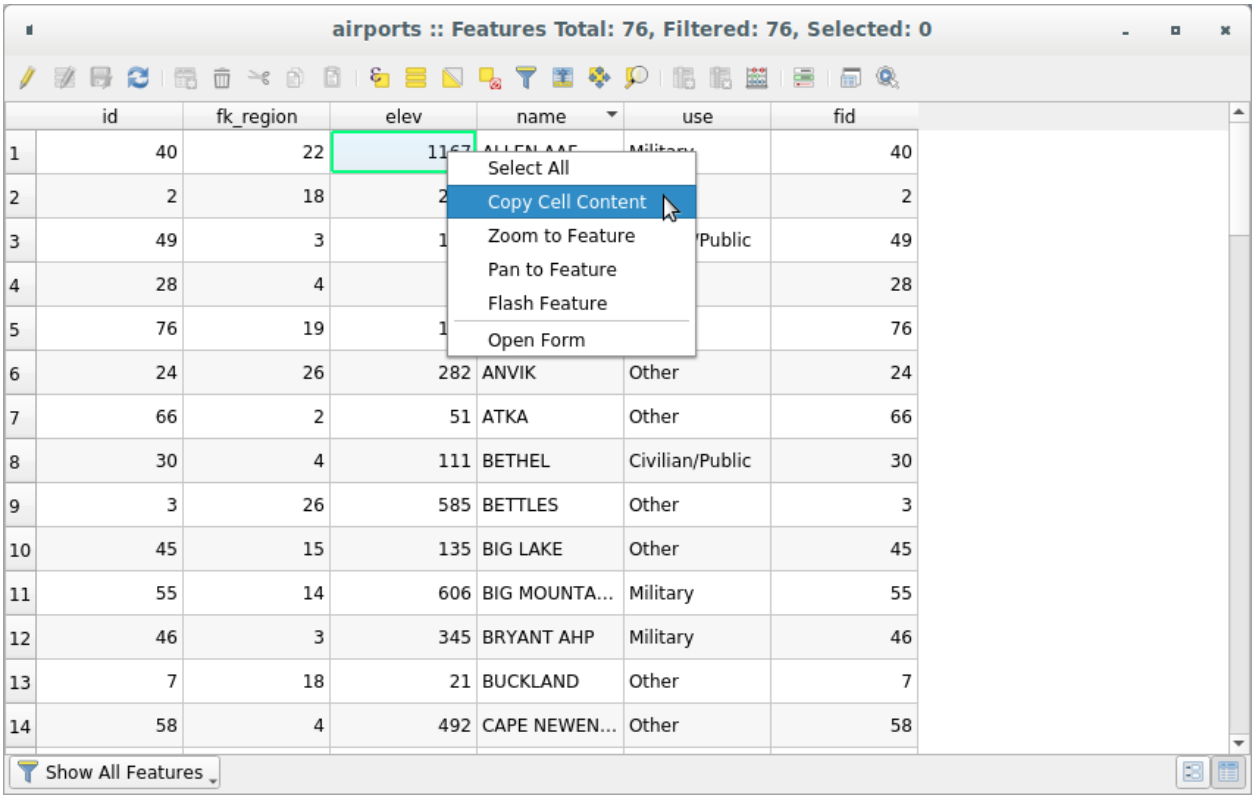


그림 14.72: 셀 내용 복사 버튼

외부 프로그램에서 (예: 엑셀, 리브레오피스, QGIS 또는 사용자 지정 웹 응용 프로그램 등) 속성 데이터를 이용하고 싶은 경우, 하나 또는 그 이상의 행 (들) 을 선택한 다음 Copy selected rows to clipboard 아이콘을 클릭하거나 `Ctrl+C` 조합키를 누르십시오.

Settings ▸ *Options* ▸ *Data Sources* 메뉴에서 *Copy features as* 드롭다운 목록을 통해 붙여넣을 포맷을 다음 가운데 정의할 수 있습니다:

- 평문 텍스트, 도형 없음
- 평문 텍스트, WKT 도형
- GeoJSON

이 컨텍스트 메뉴에 액션 목록을 표시할 수도 있습니다. *Layer properties* ▸ *Actions* 탭에서 활성화할 수 있습니다. 액션에 관한 자세한 정보는액션 속성 을 참조하세요.

선택한 피처를 새 레이어로 저장하기

선택한 피처를 모든 OGR 지원 벡터 포맷으로 저장할 수 있고, 또다른 좌표계로 변형시킬 수도 있습니다. *Layers* 패널에 있는 레이어의 컨텍스트 메뉴에서, *Export* → *Save selected features as…*를 클릭해서 산출물 데이터셋의 명칭, 포맷 및 좌표계를 정의할 수 있습니다. (기존 레이어로부터 레이어 생성 을 참조하세요.) 선택한 피처만 저장하는 *Save only selected features* 옵션이 체크돼 있을 것입니다. 이 대화창에서 OGR 생성 옵션도 설정할 수 있습니다.

14.4.5 속성값 편집

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.

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 14.73 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

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` shapefile 을 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.

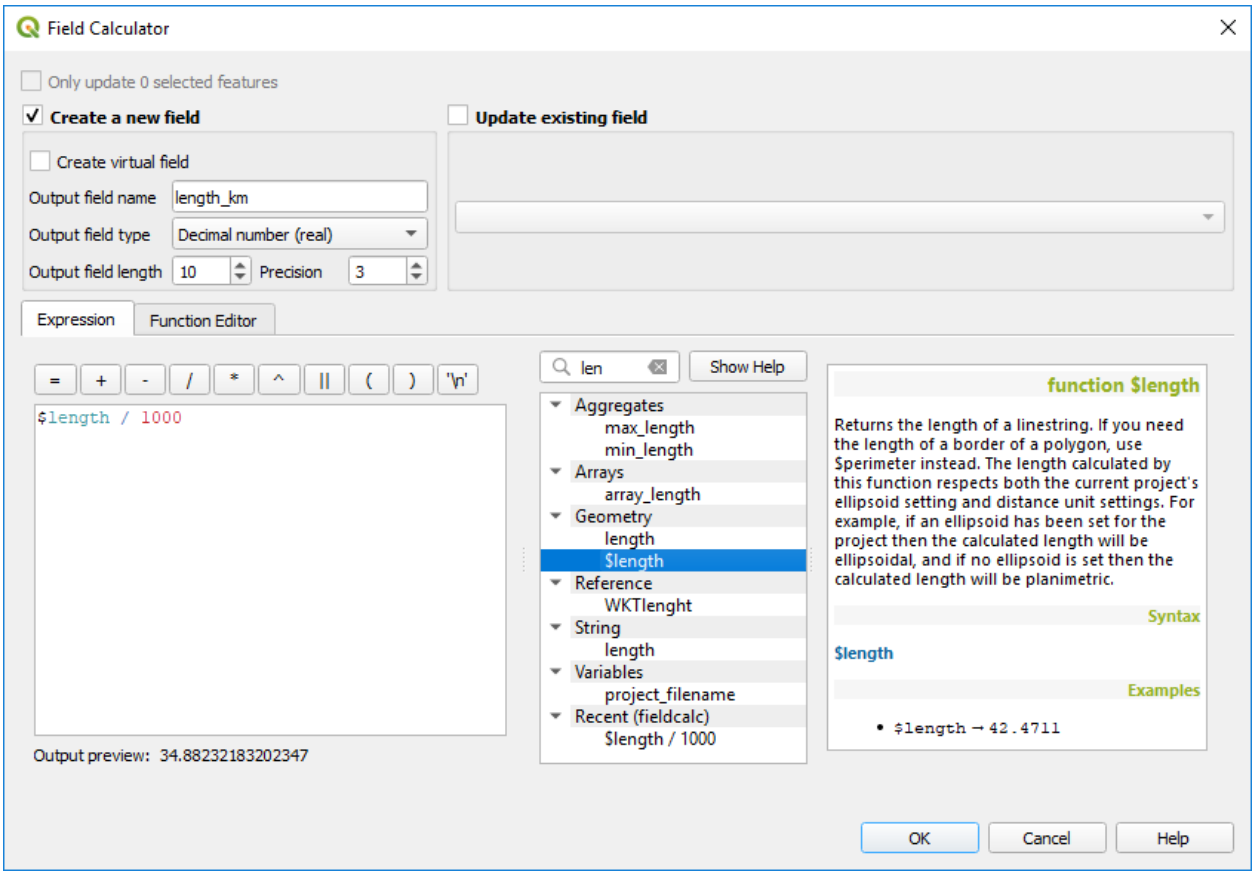


그림 14.73: 필드 계산기

7. *Geometry* 그룹에 있는 `$length` 함수를 더블클릭해서 필드 계산기의 표현식 란에 도형의 길이를 추가하십시오.
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 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.

참고: 가상 필드 사용법

- 가상 필드는 레이어 속성에 영구히 유지되지 않습니다. 가상 필드가 생성된 프로젝트의 프로젝트 파일에만 저장되고 사용할 수 있다는 뜻입니다.
- 필드 생성 시에만 가상 필드로 설정할 수 있습니다. 가상 필드는 레이어 속성 대화창의 필드 탭에서 정규 물리 또는 결합 필드와 구분하기 위해 보라색 배경으로 표시됩니다. 가상 필드를 생성한 다음에도 코멘트 열에 있는 표현식 버튼을 누르면 가상 필드의 표현식을 편집할 수 있습니다. 가상 필드의 표현식을 수정할 수 있는 표현식 편집기 창이 열릴 것입니다.

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.

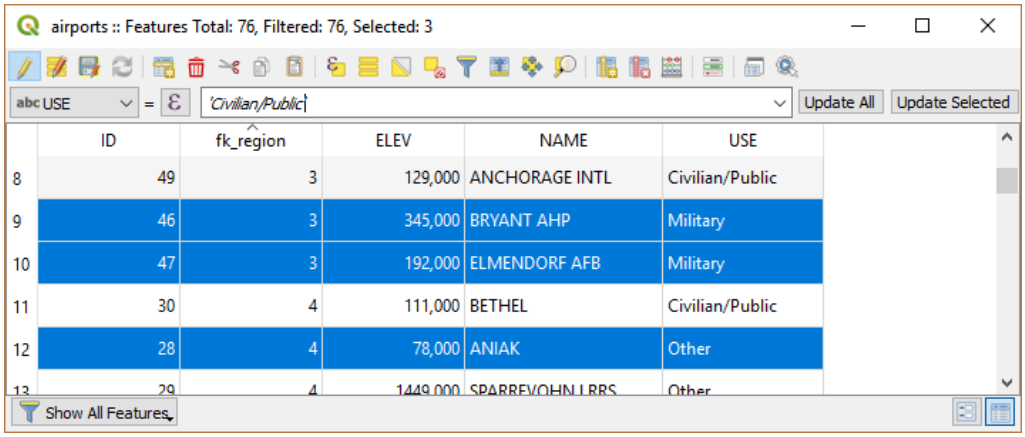


그림 14.74: 간편 필드 계산 막대

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.

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.

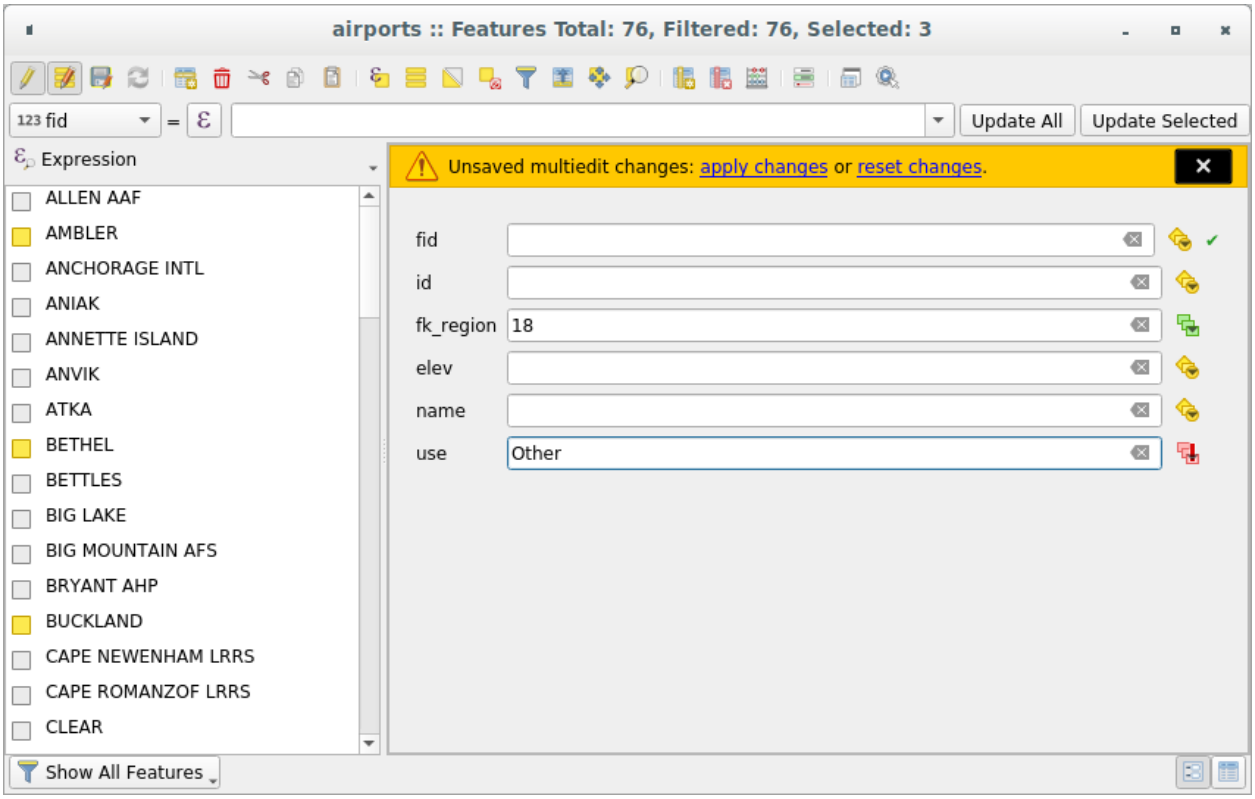


그림 14.75: 여러 피처의 필드들을 편집하기

14.4.6 일대다 또는 다대다 관계 생성

관계는 데이터베이스에서 자주 이용되는 기술입니다. 관계란 서로 다른 레이어 (테이블) 의 피처 (행) 가 서로에게 속할 수 있다는 개념입니다.

1 대다 (1-N) 관계 소개

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.

1 대다 (1-N) 관계의 레이어들

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, `regions` shapefile 과 `regions` 레이어에 대한 외래 키 필드 (`fk_region`) 를 가지고 있는 `airports` shapefile 을 불러오겠습니다. 이 말은 각 공항이 딱 1 개 지역에만 속해 있는 반면 각 지역은 공항을 몇 개든 가지고 있을 수 있다는 뜻입니다. (전형적인 일대다 관계죠.)

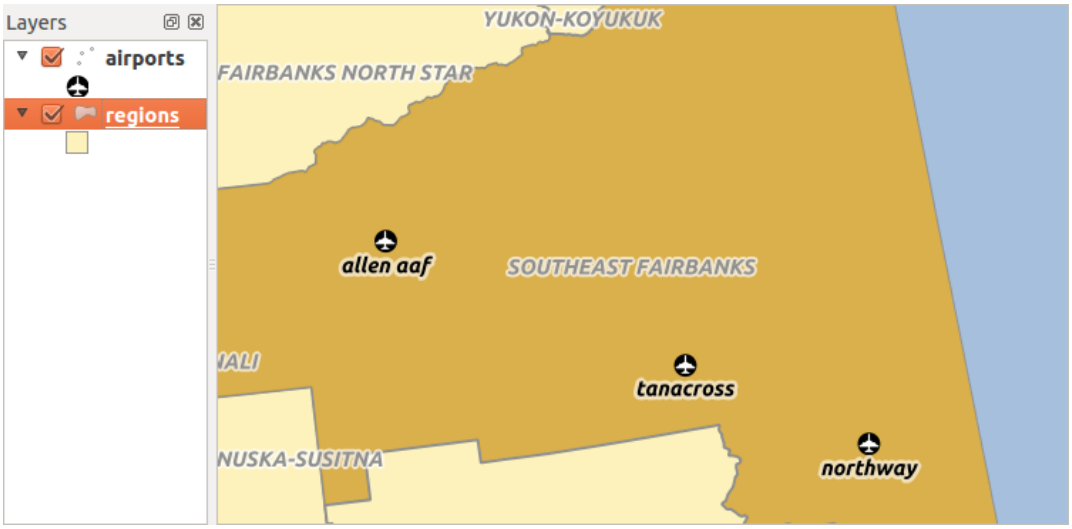


그림 14.76: 알래스카의 지역과 공항

1 대다 (1-N) 관계의 외래 키

`airports` 레이어의 속성 테이블에 있는 기존 속성 이외에, 외래 키로 동작하는 또다른 `fk_region` 필드가 필요할 겁니다. (데이터베이스를 사용하고 있다면, 외래 키에 제약 조건을 정의해두는 편이 좋습니다.)

이 `fk_region` 필드는 항상 지역의 ID 를 담고 있습니다. ID 가 속한 지역을 가리키는 포인터라고 생각할 수도 있습니다. 그리고 편집 작업을 위한 사용자 지정 편집기를 디자인할 수 있으며, QGIS 가 설정을 처리할 겁니다. 서로 다른 제공자들과 작동하며 (따라서 shapefile 및 CSV 파일도 사용할 수 있습니다) 사용자가 해야 할 일은 QGIS 에 사용자의 두 테이블 간의 관계를 알려주는 것뿐입니다.

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.

- **명칭 (Name)**: 제목으로 사용됩니다. 사람이 읽을 수 있는 문자열이어야 하며, 관계의 목적을 설명합니다. 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)**: 부모와 자식 레이어 관계의 강도를 설정합니다. 기본 *Association* 유형은 부모 레이어가 자식 레이어에 단순히 링크돼 있다는 의미인 반면, *Composition* 유형은 부모 레이어의 피처를 복제하는 경우 자식 피처도 복제되도록 할 수 있습니다.

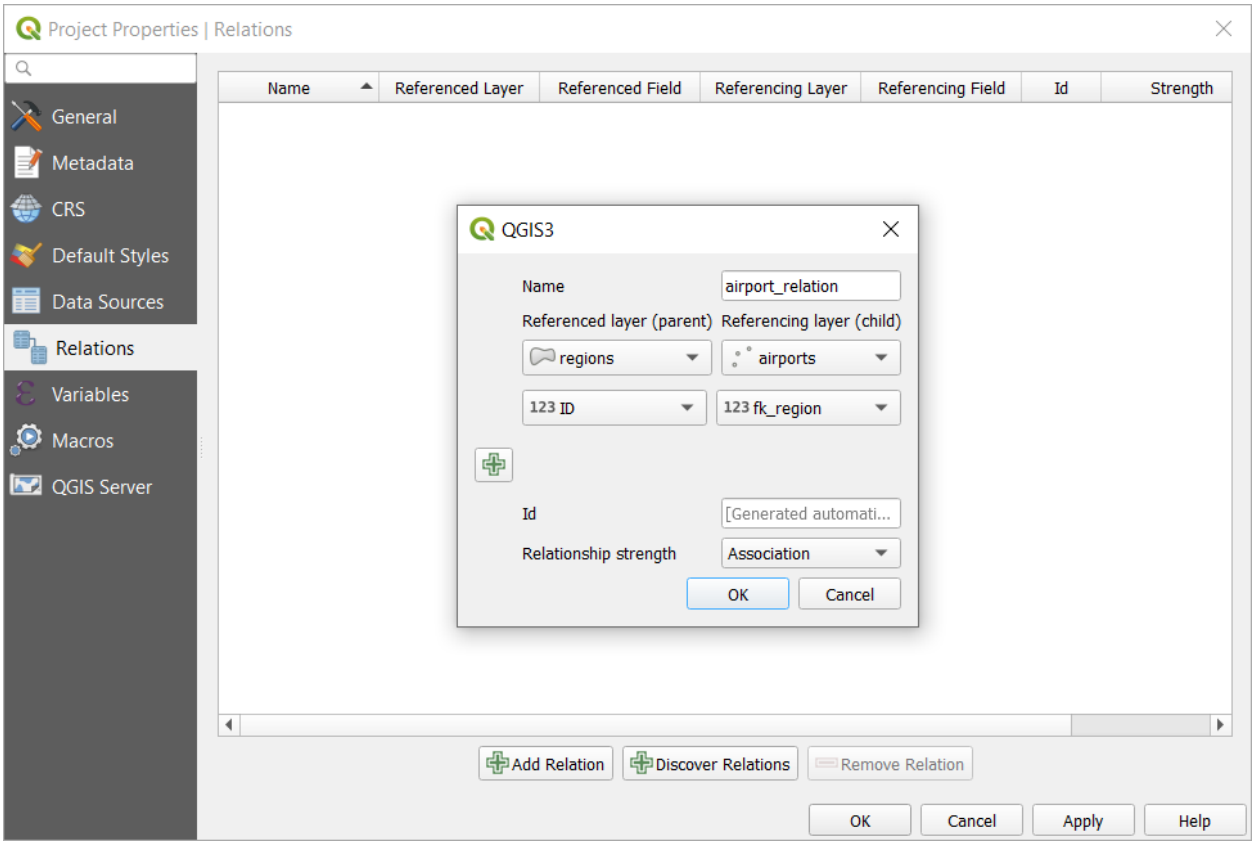


그림 14.77: 지역과 공항 레이어 사이에 관계 추가하기

Relations 탭에서, *Discover Relation* 버튼을 눌러 불러온 레이어들의 제공자로부터 사용할 수 있는 관계를 가져올 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 모든 편집 내용을 저장합니다.

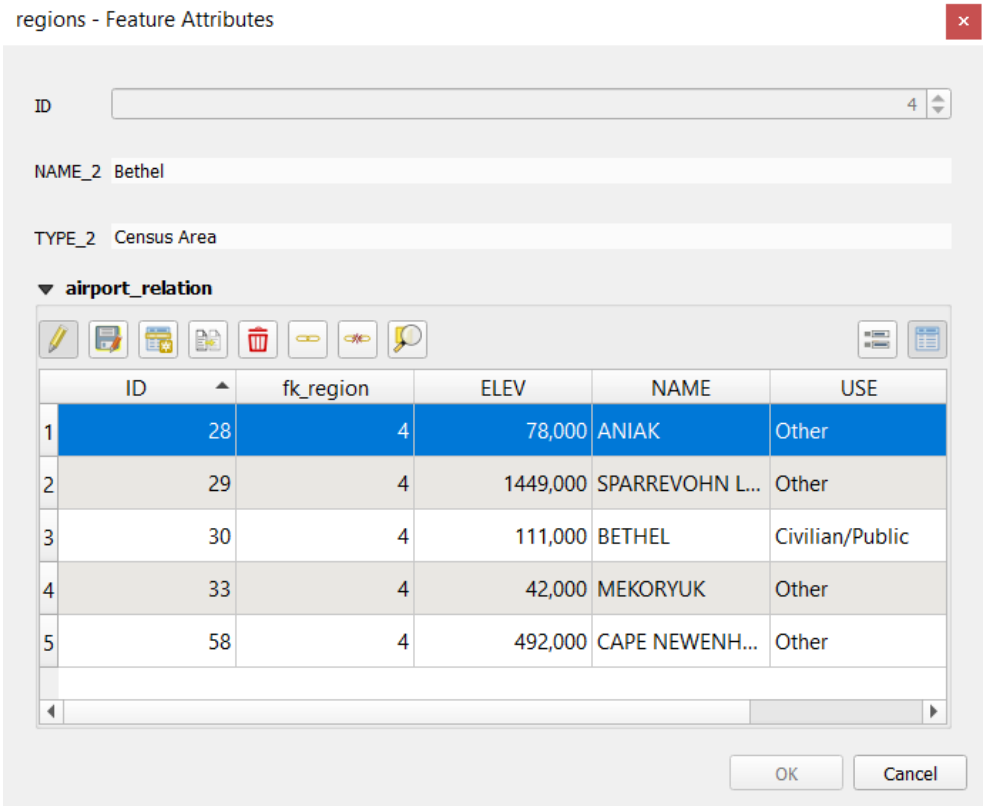


그림 14.78: “airports”와 관계가 생성된“regions”의 식별 대화창

- 버튼은 공항 레이어 속성 테이블에 새 레코드를 추가할 것입니다. 또 기본적으로 현재 지역에 새 공항을 할당할 것입니다.
- 버튼은 버튼과 동일하지만, 속성 테이블에 새 레코드를 추가하기 전에 맵 캔버스에서 공항 도형을 디지타이즈할 수 있게 해줍니다. 아이콘은 도형 유형에 따라 변한다는 점을 기억하십시오.
- 버튼은 하나 이상의 자식 피처를 복사할 수 있습니다.
- 버튼은 선택한 공항을 영구적으로 삭제할 것입니다.
- 버튼은 기존의 어떤 공항이라도 선택해서 현재 지역에 할당시킬 수 있는 새 대화창을 엽니다. 사용자가 잘못된 지역에 공항을 실수로 생성한 경우 이 기능이 유용할 겁니다.
- 버튼은 현재 지역에서 선택한 공항을 할당 해제시켜, 사실상 비할당 상태로 (외래 키가 NULL 로 설정된 상태로) 남겨둡니다.
- 버튼을 사용하면 선택한 자식 피처로 맵을 확대/축소할 수 있습니다.
- 오른쪽에 있는 와 두 버튼은 테이블 뷰와 양식 뷰를 서로 전환하는데, 양식 뷰에서는 모든 공항을 각각의 양식대로 살펴볼 수 있습니다.

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.

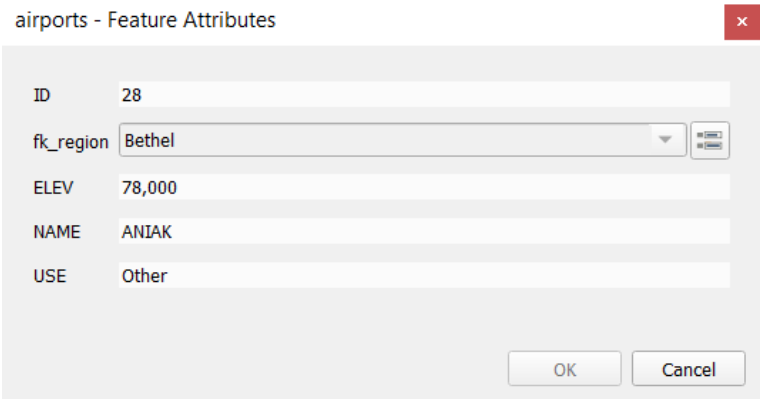
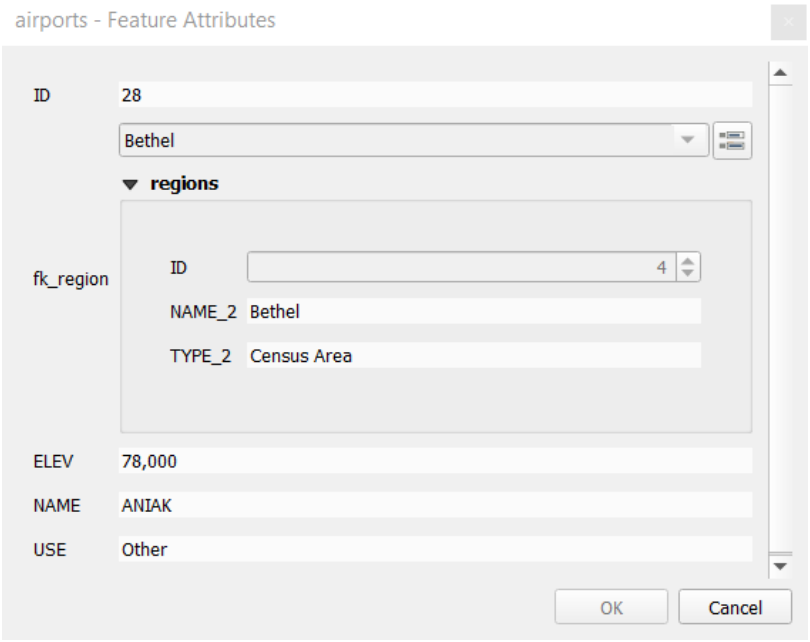


그림 14.79: “regions”와 관계가 생성된“airports”의 식별 대화창

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 14.80 에서, 공항의 평균 해발고도가 500m 를 초과하는 모든 지역을 선택합니다.

이 양식에서 여러 서로 다른 집계 함수들을 사용할 수 있을 것입니다.

다대다 (N-M) 관계 소개

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.

다음 SQL 코드는 PostgreSQL/PostGIS 스키마에 *locations* 라는 명칭의 N-M 관계를 위해 필요한 테이블 3 개를 생성합니다. PostGIS 의 경우 *Database* ▸ *DB Manager*…메뉴를 통해, 또는 pgAdmin 같은 외부 도구들을 통해 코드를 실행할 수 있습니다. 공항 테이블은 `airports` 레이어를, 항공사 테이블은 `airlines` 레이어를 저장하고 있습니다. 이 두 테이블에서 관계를 분명히 하기 위해 사용되는 필드는 몇 개 되지 않습니다. 까다로운 부분은 `airports_airlines` 테이블입니다. 이 테이블에 모든 공항에 대해 모든 항공사를 (그리고 그 반대도) 전부 목록화해야 합니다. 이런 유형의 테이블을 피벗 *(pivot)* 테이블 이라고 합니다. 공항과 항공사 둘 다 이미 각각의 레이어에 존재하는 경우에만 공항을 항공사와 관련지을 수 있도록 강제하는 것이 이 테이블의 제약조건 입니다.

```
CREATE SCHEMA locations;

CREATE TABLE locations.airports
(
   id serial NOT NULL,
```

(다음 페이지에 계속)

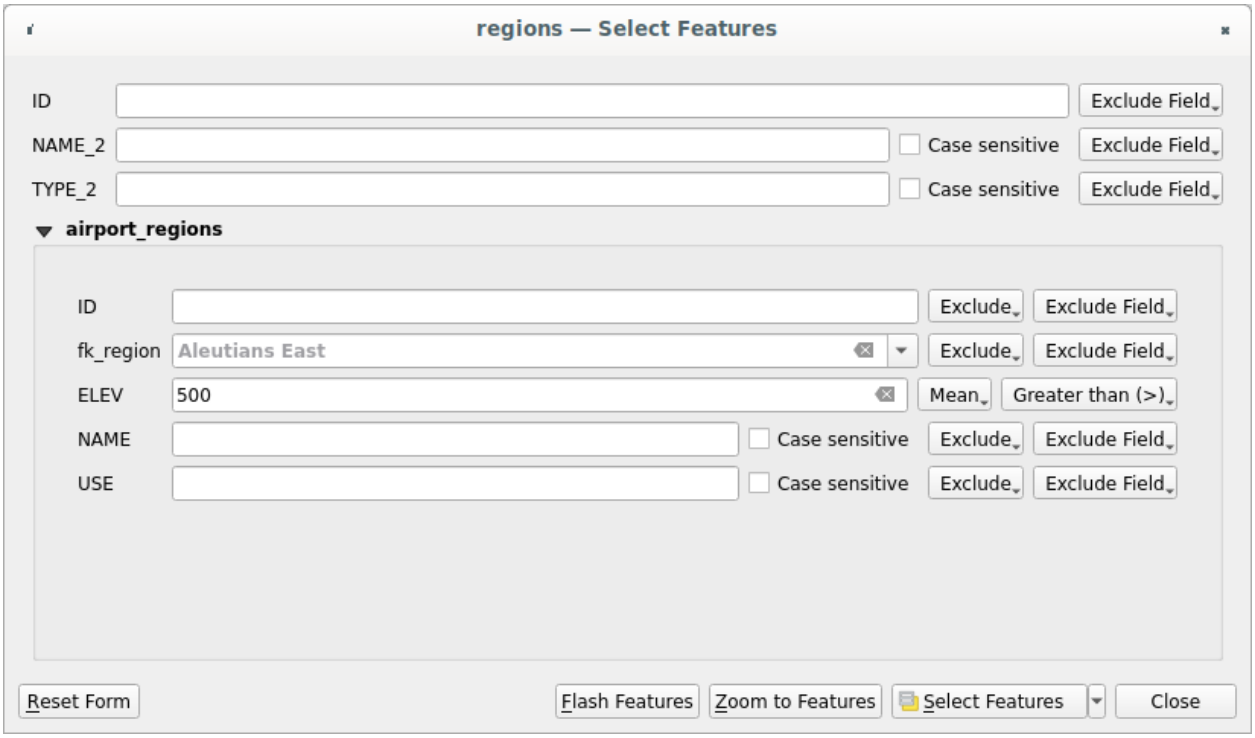


그림 14.80: 자식 값으로 부모 피처 선택

(이전 페이지에서 계속)

```
   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);

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,
```

(다음 페이지에 계속)

(이전 페이지에서 계속)

```
   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.

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` 레이어를 위한 올바른 기수성 (cardinality) 을 선택해야 합니다. 전자의 경우 **airlines (id)** 옵션을, 후자의 경우 **airports (id)** 옵션을 선택해야 합니다.

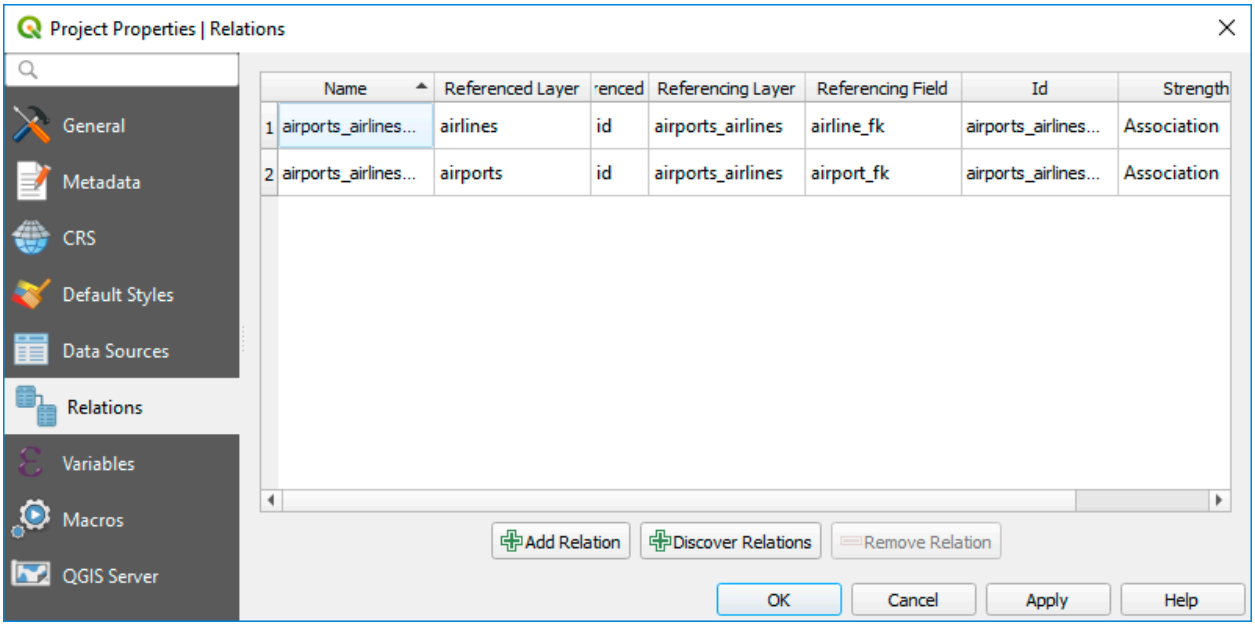


그림 14.81: 관계와 자동검색

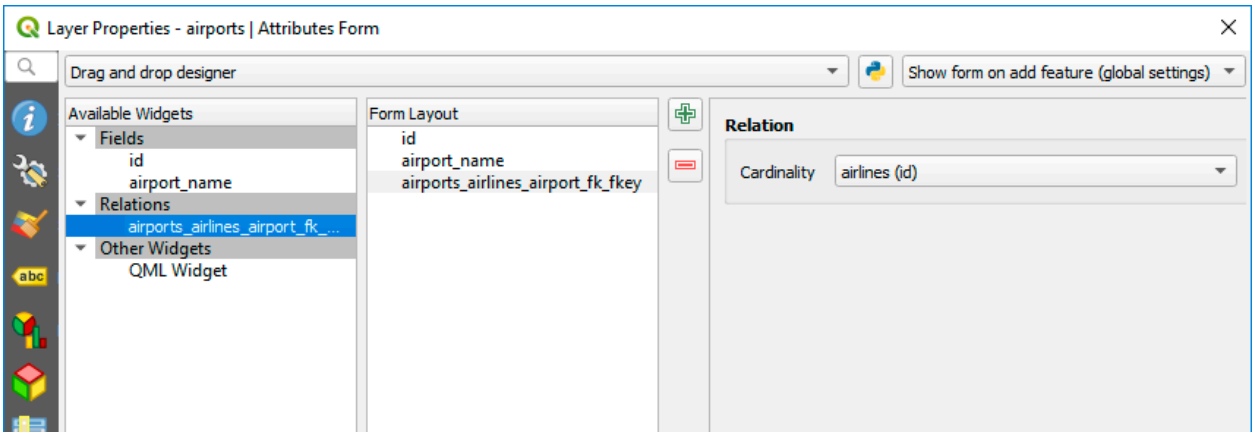


그림 14.82: 관계 기수성 설정

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.

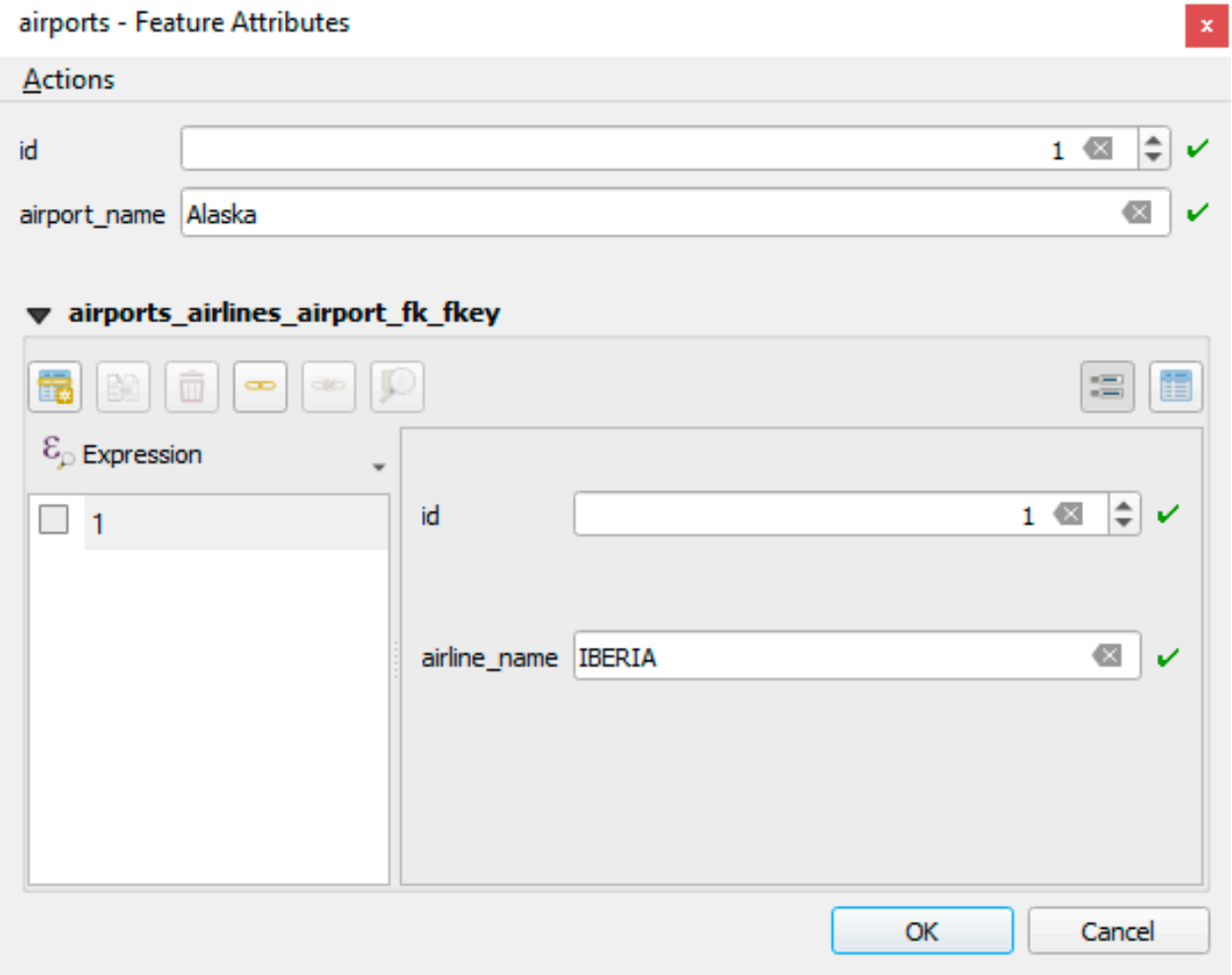


그림 14.83: 공항과 항공사 사이의 N-M 관계

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 레이어인 (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.

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.

14.5 편집 작업

QGIS 는 OGR, SpatiaLite, PostGIS, MSSQL Spatial 및 Oracle Spatial 벡터 레이어와 테이블을 편집할 수 있는 다양한 기능을 보유하고 있습니다.

참고: GRASS 레이어 편집 과정은 조금 다릅니다. 자세한 내용은*GRASS* 벡터 레이어 디지타이즈 및 편집하기 를 참조하세요.

팁: 병행 편집

QGIS 이번 버전은 사용자가 피처를 편집하는 도중에 다른 사용자가 동일한 피처를 편집하고 있지는 않은지 추적하지 않습니다. 마지막에 저장하는 변경 사항이 적용될 것입니다.

14.5.1 스냅 허용 오차 및 검색 반경 설정

벡터 레이어 도형을 정확하게, 최적으로 편집하기 위해 피처 꼭짓점에 대한 스냅 허용오차 및 검색 반경을 적절한 값으로 설정해야 합니다.

스냅 허용오차

스냅 허용 오차란 사용자가 새 꼭짓점을 설정하거나 또는 기존 꼭짓점을 이동시킬 때 연결하려는 가장 가까운 꼭짓점 또는 선분을 검색하기 위해 QGIS 가 사용하는 거리를 말합니다. 스냅 허용 오차 안이 아닌 경우, QGIS 는 기존 꼭짓점 또는 선분에 스냅시키는 대신 사용자가 마우스 버튼을 놓은 위치에 꼭짓점을 둘 것입니다.

스냅 허용 오차 설정은 허용 오차를 사용하는 모든 도구에 적용됩니다.

Snapping Toolbar 에 있는 Enable snapping 버튼을 클릭하거나 `s` 키를 누르면 스냅 모드를 활성화/비활성화할 수 있습니다. 이 툴바에서 스냅 모드, 허용 오차 값, 단위도 환경 설정할 수 있습니다.

Project → *Snapping Options*…에서도 스냅 모드 환경을 설정할 수 있습니다.

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*: 스냅 모드 및 허용 오차를 레이어 기반으로 활성화하고 조정할 수 있습니다. (그림 14.84 를 참조하세요.) 어떤 레이어를 편집하고 있는데 다른 레이어에 꼭짓점을 스냅시켜야 하는 경우, 대상 레이어가 체크된 상태인지 확인하고 스냅 허용 오차를 훨씬 큰 값으로 올리십시오. 스냅 옵션 대화창에서 체크되지 않은 레이어에서는 스냅이 결코 일어나지 않을 것입니다.

스냅 모드를 `To vertex`, `To segment`, 그리고 `To vertex and segment` 가운데 하나로 선택할 수 있습니다.

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.

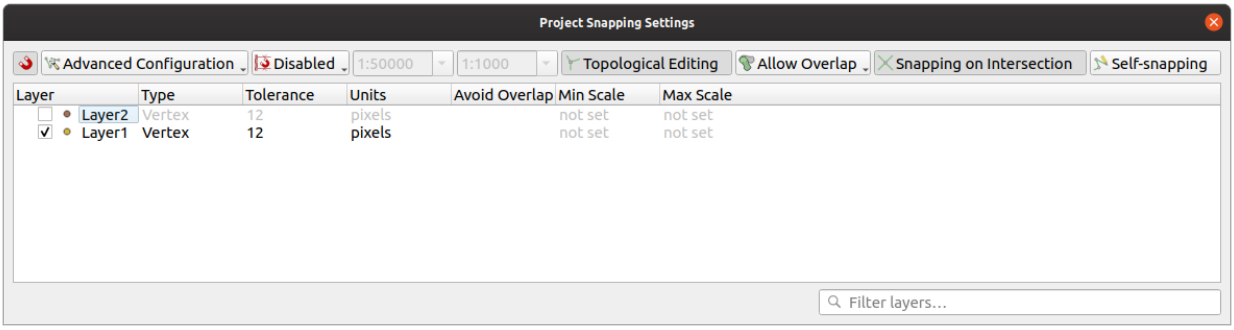


그림 14.84: 스냅 옵션 (고급설정 모드)

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

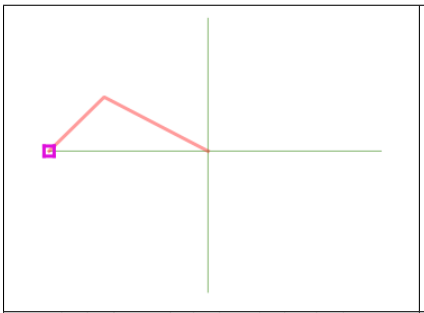
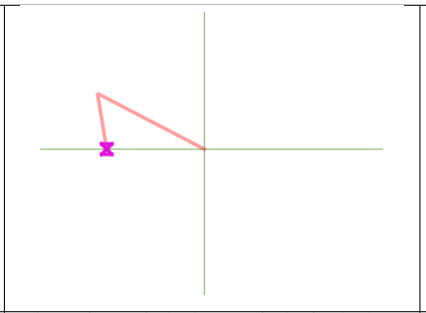
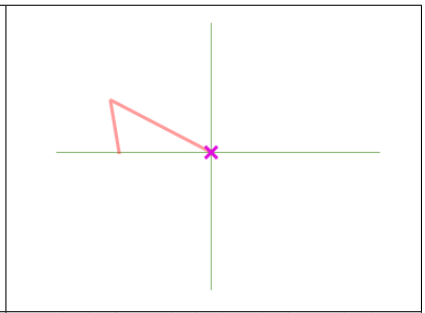
Settings ▸ *Options* ▸ *Digitizing* 탭에서 모든 새 프로젝트가 기본적으로 스냅을 활성화시키도록 설정할 수 있습니다. 기본 스냅 모드, 허용 오차 값, 그리고 단위도 설정할 수 있습니다. 여기서 설정한 값들이 *Snapping Options* 대화창에 표시될 것입니다.

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.

스냅 아이콘

QGIS 는 스냅 의 유형에 따라 서로 다른 스냅 아이콘을 표시해드립니다:

		
꼭짓점에 스냅하기: 상자 아이콘	선분에 스냅하기: 모래시계 아이콘	교차점에 스냅하기: 가위표 아이콘

사용자 설정의 *Digitizing* 부분에서 이 아이콘들의 색상을 변경할 수 있다는 것을 기억하십시오.

검색 반경

Search radius for vertex edits 는 맵을 클릭했을 때 사용자가 선택하려는 가장 가까운 꼭짓점을"검색"하기 위해 QGIS 가 사용하는 거리를 말합니다. 검색 반경 안이 아닌 경우, QGIS 가 아무것도 찾지 못 하고 편집 작업을 위한 어떤 꼭짓점도 선택하지 못 할 것입니다. *Settings* ▸ *Options* ▸ *Digitizing* 탭에서 꼭짓점 편집 용 검색 반경을 정의할 수 있습니다. (이 탭이 스냅 기본 값을 정의하는 곳입니다.)

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.

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.

14.5.2 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.

새 폴리곤의 중첩을 피하기

스냅 모드를 `Advanced configuration` 으로 설정했을 때, 폴리곤 레이어의 경우 *Avoid overlap* 이라는 옵션이 존재합니다. 이 옵션은 선택한 레이어에서 기존 피처에 중첩하는 새 피처를 그리지 못 하도록 막습니다. 다른 폴리곤과 인접한 폴리곤을 디지타이징할 때 속도를 향상시키는 효과가 있습니다.

이미 폴리곤이 하나 있는 경우, 이 옵션을 활성화한 상태로 두 번째 폴리곤을 기존 폴리곤과 서로 교차하도록 디지타이즈할 수 있는데, 이때 QGIS 가 두 번째 폴리곤을 기존 폴리곤의 경계선에 맞춰 자를 것입니다. 공통 경계선의 꼭짓점을 모두 디지타이즈하지 않아도 된다는 장점이 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.

도형 점검기

이 핵심 플러그인으로 도형의 유효성을 확인할 수 있습니다. 도형 점검기 플러그인 에서 이 플러그인에 관한 자세한 정보를 찾아볼 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.

14.5.3 기존 레이어 디지타이징

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: Shapefile 을) 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:

아이콘	목적	아이콘	목적
	현재 편집 내용		편집 변경
	레이어 편집 내용 저장		
	신규레코드 추가		피처 추가: 포인트 캡처
	피처 추가: 라인 캡처		피처 추가: 폴리곤 캡처
	꼭짓점 도구 (모든 레이어)		꼭짓점 도구 (현재 레이어)
	선택한 모든 피처의 속성을 동시에 수정		
	선택 항목 삭제		피쳐 자르기
	피쳐 복사		피쳐 붙여넣기
	실행 취소		재실행

표: 편집 작업 - 벡터 레이어 기본 편집 툴바

이 모든 디지타이즈 작업 도구를 쓰는 도중에 도구를 비활성화시키지 않고서도 맵 캔버스에서확대/축소 및 이동 할 수 있다는 사실을 기억하십시오.

모든 편집 세션은 선택한 레이어의 컨텍스트 메뉴, 속성 테이블 대화창, 디지타이즈 작업 툴바, 또는 *Edit* 메뉴에 있는  Toggle editing 옵션을 선택해서 시작됩니다.

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.

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.

비도형 피처를 추가하려면,  Add Record 버튼을 클릭하면 열리는 피처 양식에 속성을 입력할 수 있습니다. 공간 활성화된 도구로 피처를 생성하려면, 먼저 도형을 디지타이즈한 다음 그 속성을 입력해야 합니다. 도형을 디지타이즈하려면:

1. 맵 영역을 왼쪽 클릭해서 사용자의 새 피처의 첫 번째 포인트를 생성하십시오. 포인트 피처의 경우 이것만으로 충분하며, 필요한 경우 피처 속성을 입력할 수 있는 피처 양식이 열릴 것입니다. 레이어 속성에서도형 정밀도 를 설정했다면 정규 거리를 기반으로 피처를 생성하는그리드에 스냅 기능을 사용할 수 있습니다.
2. 라인 또는 폴리곤 도형의 경우, 사용자가 캡처하려는 각 추가 포인트를 왼쪽 클릭하거나 또는자동 투사 기능을 이용해서 디지타이즈 작업 속도를 올릴 수 있습니다. 이렇게 하면 사용자가 배치한 꼭짓점들 사이에 연속적인 직선을 생성할 것입니다.

참고: 최근 추가한 노드를 되돌리려면 `Delete` 또는 `Backspace` 키를 누르십시오.

3. 포인트 추가 작업을 완료했다면, 맵 영역 어디든 오른쪽 클릭해서 해당 피처의 도형 입력 작업이 끝났다고 확인하십시오.

참고: 라인 또는 폴리곤 도형을 디지타이즈하는 도중 선형 *Add feature* 도구와원호 스트링 도구 사이를 전환해가며 복합 만곡 도형을 생성할 수 있습니다.

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.

4. 디지타이즈 작업을 완료하면 새 피처를 위한 정보를 입력할 수 있는 속성 대화창이 열립니다. 그림 14.85 은 알래스카에 있는 가상의 새 강을 위한 속성을 설정하는 예시입니다. 하지만, *Settings* ▸ *Options* 메뉴의 *Digitizing* 대화창에서 다음 옵션들을 활성화시킬 수도 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.

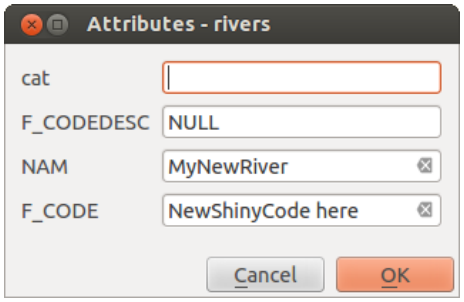


그림 14.85: 새 벡터 피처를 디지타이즈한 후의 속성 입력 양식 대화창

꼭짓점 도구

참고: QGIS 3 버전의 주요 변경사항

QGIS 3 버전에서, 노드 도구를 완전히 새로 설계하고 꼭짓점 도구 로 명칭을 변경했습니다. 이전에는"클릭 & 드래그"인간공학 (ergonomy) 으로 작동했지만, 지금은"클릭-클릭"워크플로 (workflow) 를 사용합니다. 이렇게 해서 여러 레이어의 객체들을 동시에 디지타이즈하거나 편집하는 동안 꼭짓점 도구가 있는 고급 디지타이즈 작업 패널의 장점을 취하는 것과 같은 중요한 개선을 이룰 수 있었습니다.

편집 가능 상태의 모든 벡터 레이어에 대해, Vertex tool (Current Layer) 은 CAD 소프트웨어와 유사한 피처 꼭짓점 수정 기능을 제공합니다. 그냥 한 번에 여러 꼭짓점을 선택해서 한꺼번에 이동, 추가 또는 삭제할 수 있습니다. 꼭짓점

도구는 피처 위상 편집 작업도 지원합니다. 이 도구는 선택 지속적이기 때문에 일부 작업이 완료된 후에도 해당 피처 및 도구에 대한 선택 집합의 활성화 상태가 유지됩니다.

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 이번 버전은'Semi-transparent circle', 'Cross'및'None'세 종류의 꼭짓점 마커를 지원하고 있습니다. 마커 스타일을 변경하려면, *Settings* 메뉴에서 *Options* 를 선택한 다음, *Digitizing* 탭에서 적절한 항목을 선택하십시오.

기본 작업

먼저 Vertex Tool (Current Layer) 를 활성화하십시오. 꼭짓점에 마우스를 가져가면 빨간색 원이 표시될 것입니다.

- **꼭짓점 선택하기**: `Shift` 키를 누른 채 한 번에 하나씩 클릭하거나, 여러 꼭짓점을 감싸는 직사각형을 클릭 & 드래그해서 꼭짓점을 선택할 수 있습니다. 꼭짓점을 선택하면, 그 색상이 파란색으로 변합니다. 현재 선택 집합에 더 많은 꼭짓점을 추가하려면, `Shift` 키를 누른 채 클릭하십시오. `Ctrl` 키를 누른 채 클릭하면 선택 집합에서 꼭짓점을 제거할 수 있습니다.
- **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.

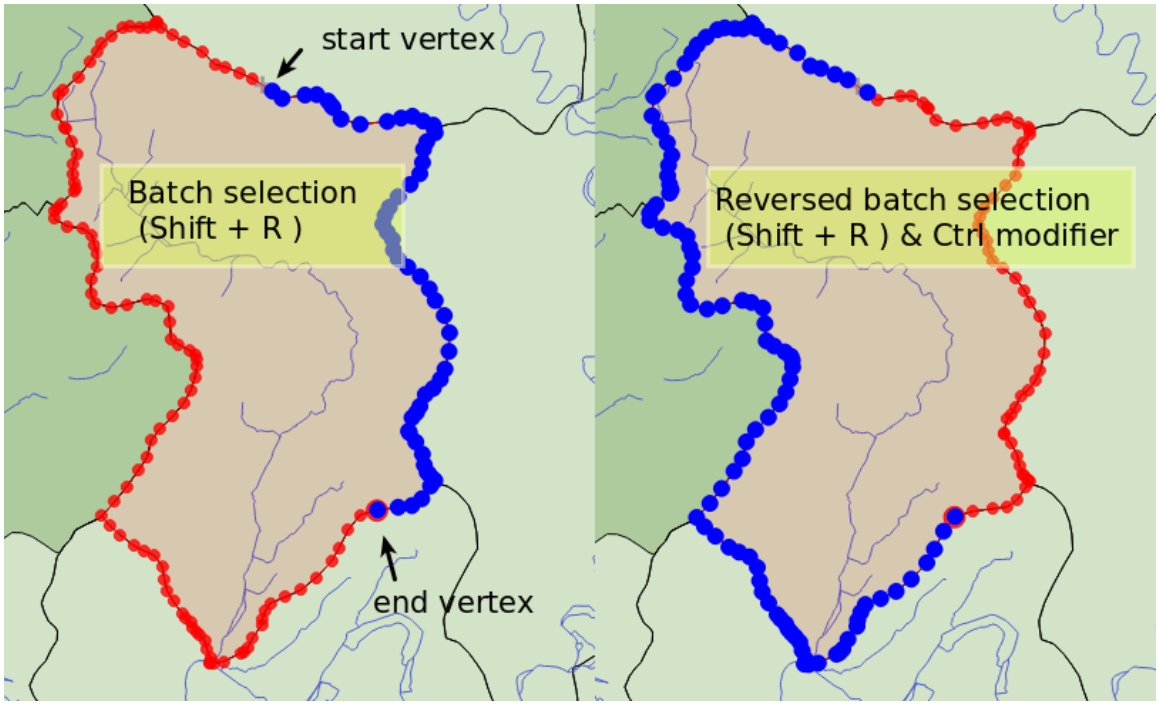


그림 14.86: `Shift+R` 로 배치 꼭짓점 선택하기

`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 추가하기**: 꼭짓점을 추가하려 할 때, 선분 중앙에 새 가상 노드가 나타납니다. 그냥 이 노드를 잡아 새 꼭짓점을 추가하십시오. 경계선 상의 어느 위치를 더블클릭해도 새 노드를 생성합니다. 라인 도형의 경우, 양극단에 라인을 연장할 수 있는 가상 노드가 나타납니다.

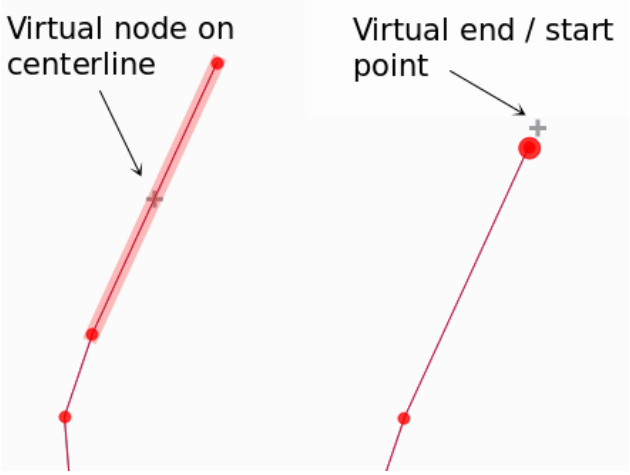


그림 14.87: 꼭짓점을 추가하기 위한 가상 노드

- **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 이동하기**: 이동시키고자 하는 모든 꼭짓점을 선택한 다음, 선택한 꼭짓점 또는 경계를 클릭하고 다시 이동하려는 위치를 클릭하십시오. 선택한 모든 꼭짓점이 함께 움직일 것입니다. 스냅이 활성화돼 있는 경우, 전체 집합을 가장 가까운 꼭짓점 또는 라인으로 점프시킬 수 있습니다. 두 번째로 클릭하기 전에 거리, 각도, 정확한 X, Y 위치를 위한 고급 디지타이즈 작업 패널의 제약조건을 이용할 수 있습니다.

  이때 그리드에 스냅 기능을 사용할 수 있습니다. 레이어 속성에서도형 정밀도 용 값을 설정했다면, 도형 정밀도를 따르는 확대/축소 수준에서 그리드가 나타납니다.

꼭짓점의 각 변경 사항은 *Undo* 대화창에 개별 항목으로 각각 저장됩니다. 이 설정이 켜져 있는 동안 모든 작업이 위상 편집을 지원한다는 사실을 기억하십시오. 실시간 재투영도 지원합니다. 포인터를 꼭짓점 위에 가져가면 꼭짓점 도구가 꼭짓점을 식별할 수 있는 도움말 말풍선 (tooltip) 을 표시합니다.

꼭짓점 편집기 패널

피처에 *Vertex tool* 을 사용할 때, 꼭짓점을 오른쪽 클릭하면 QGIS 가 피처의 모든 꼭짓점과 각 꼭짓점의 *x*, *y* (존재하는 경우 *z*, *m*) 좌표 및 (원호 도형의 경우 반경을 나타내는) *r* 목록을 담은 *Vertex Editor* 패널을 엽니다. 이 테이블에서 어느 행을 선택하기만 하면 맵 캔버스에 있는 해당 꼭짓점이 선택되며, 그 반대의 경우도 가능합니다. 테이블에서 좌표를 변경하기만 하면 해당 꼭짓점의 위치가 업데이트됩니다. 여러 행을 선택해서 한꺼번에 삭제할 수도 있습니다.

참고: QGIS 3.4 에서 변경된 내용

피처를 오른쪽 클릭하는 즉시 꼭짓점 편집기를 열고 해당 피처를 잠급니다. 즉 다른 모든 피처의 편집 모드를 비활성화시킵니다. 이렇게 잠겨 있는 동안, 해당 피처만 편집할 수 있습니다: 이 피처의 꼭짓점 및 선분만 클릭 또는 드래그해서 선택하거나 이동시킬 수 있습니다. 이 잠긴 피처에만 새 꼭짓점을 추가할 수 있습니다. 또한 이제 꼭짓점 도구를 활성화시킬 때 자동적으로 꼭짓점 편집기 패널이 스스로 열리고, 꼭짓점 편집기를 사용할 때마다 그 위치/도킹 상태를 기억합니다.

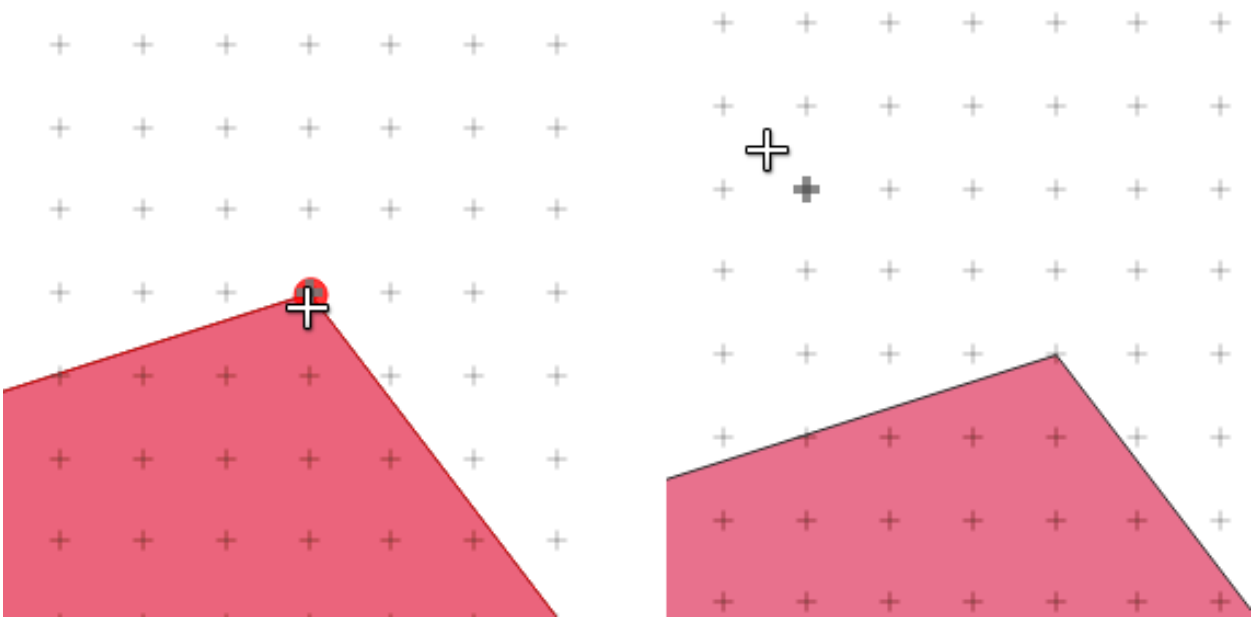


그림 14.88: 꼭짓점을 선택해서 그리드로 이동시키기

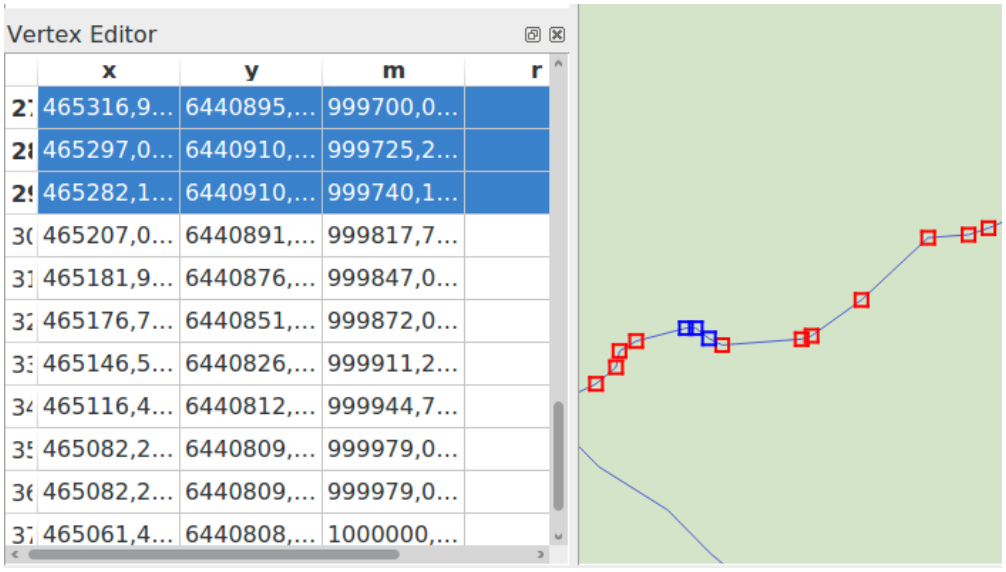


그림 14.89: 선택한 노드를 보여주는 꼭짓점 편집기 패널

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.

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/OGR 1.10 과 함께 구동되는 GDAL Shapefile

드라이버가 삽입될 데이터의 길이를 동적으로 수용하기 위해 문자열 및 정수형 필드를 자동 확장하기 때문입니다.

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 14.90 을 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.

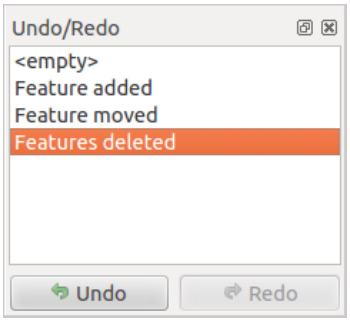


그림 14.90: 실행 취소/재실행 디지타이즈 작업 이력

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.

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.

14.5.4 고급 디지타이즈 작업

아이콘	목적	아이콘	목적
	고급 디지타이즈 작업 도구 활성화		투사 활성화
	피처 (들) 이동		피처 (들) 복사 및 이동
	피처 (들) 회전		피처 단순화
	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		

표: 고급 편집 작업 - 벡터 레이어 고급 편집 작업 툴바

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 단순화

 Simplify Feature 도구는 도형의 유효성이 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.

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.

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.

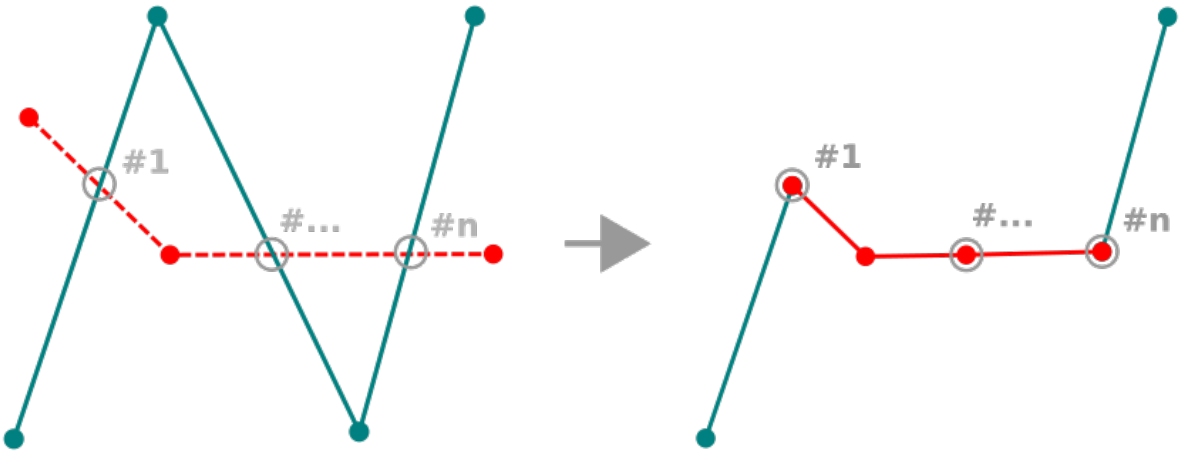


그림 14.91: 라인 재형성하기

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 유효하지 않은 폴리곤이 생성되기 때문입니다.

참고: 재형성 도구는 폴리곤 고리 또는 닫힌 라인의 시작점을 변경시킬 수도 있습니다. 즉, '두 번'나타나던 포인트가 더 이상 그러지 않을 수도 있다는 뜻입니다. 대부분의 응용 프로그램에서 문제가 없을 수도 있지만, 고려해야 할 사항임에는 틀림없습니다.

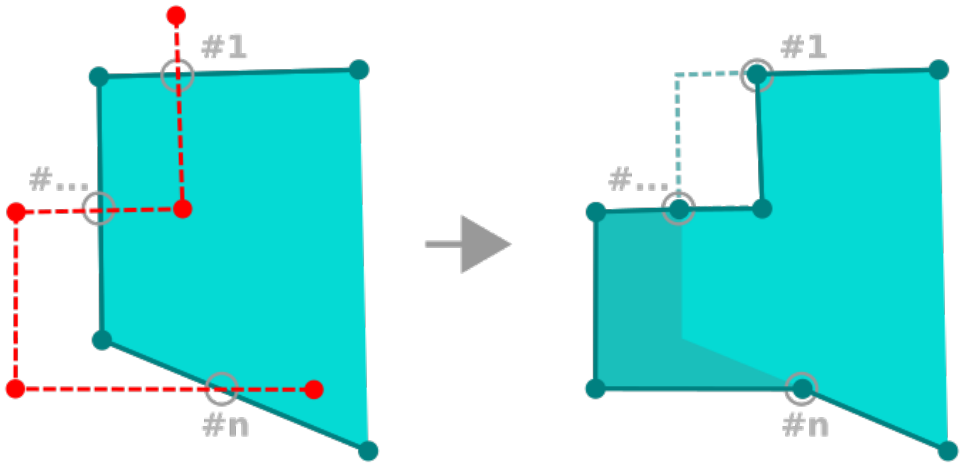


그림 14.92: 폴리곤 재형성하기

곡선 오프셋

Offset Curve 도구는 라인 레이어를 평행하게 이동시킨 객체를 생성합니다. 이 도구를 편집 중인 레이어에 적용하면 도형을 수정하며, 배경 레이어에 적용하면 라인/고리를 복사해서 편집 중인 레이어에 추가합니다. 따라서 이 도구는 거리 라인 레이어를 생성하는 데 적합합니다. 이 도구는 변위 거리를 보여주는 *User Input* 대화창을 엽니다.

라인 레이어를 이동시킨 객체를 생성하려면, 먼저 편집 모드를 켠 다음 Offset Curve 도구를 활성화시켜야 합니다. 그 다음 피처를 클릭해서 이동시킵니다. 마우스를 움직여 원하는 곳을 클릭하거나 또는 사용자 입력 위젯에 원하는 거리를 입력하십시오. 이동 후에 Save Layer Edits 도구로 사용자 변경 사항을 저장할 수도 있습니다.

QGIS 옵션 대화창 (의 디지타이즈 탭에서 **Curve offset tools** 부분) 에서 **Join style**, **Quadrant segments**, **Miter limit** 과 같은 몇몇 파라미터를 환경 설정할 수 있습니다.

라인 반전

라인 도형의 방향을 변경하는 것은 지도제작 목적 또는 네트워크 분석 준비에 유용할 수 있습니다.

라인 방향을 변경하려면:

1. Reverse line 아이콘을 클릭해서 라인 반전 도구를 활성화하십시오.
2. 라인을 클릭하십시오. 라인의 방향이 반전됩니다.

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. 분할하고자 하는 피처 (들) 를 가로지르는 라인을 그리십시오. 선택 집합이 활성화돼 있다면 선택한 피처들만 분할됩니다. 분할이 되면, 기본 값 또는 절 *(clause)* 이 대응하는 필드에 적용되며 기본적으로 부모 피처의 다른 속성들도 새 피처에 복사됩니다.
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. 새 대화창의 테이블 하단에 있는 *Merge* 행이 산출되는 피처의 속성들을 표시합니다. 이 값들을 다음과 같이 수정할 수 있습니다:
   - 대응하는 셀에서 값을 직접 대체합니다.
   - 테이블에서 행을 선택하고 *Take attributes from selected feature* 를 클릭하면 선택한 최초 피처의 값을 사용합니다.
   - *Skip all fields* 를 클릭하면 빈 속성을 사용합니다.
   - 또는, 테이블 상단에 있는 드롭다운 메뉴를 펼쳐 앞의 옵션들 가운데 하나를 선택하면 대응하는 필드에만 적용됩니다. 또 최초 피처들의 속성을 집계하도록 (필드 유형에 따라 최소, 최대, 평균, 합계, 개수, 연결 등등) 선택할 수도 있습니다. (함수들의 완전한 목록은통계 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.

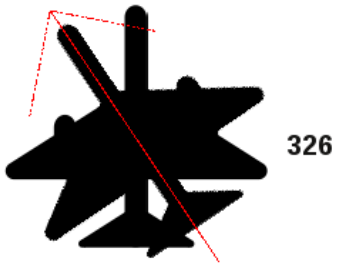


그림 14.93: 포인트 심볼 기울이기

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 14.93 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

디지타이즈한 라인이 또다른 라인에 스냅하기엔 너무 짧거나 너무 긴 경우 (라인이 빠졌거나 라인과 교차하는 경우) 선분을 연장하거나 짧게 만들 수 있어야 합니다.

Trim/Extend 도구는 [멀티] 라인은 물론 [멀티] 폴리곤도 수정할 수 있습니다. 게다가 라인의 종단만 대상으로 삼을 필요도 없습니다. 도형의 어떤 선분도 수정할 수 있습니다.

참고: 도형의 유효성이 망가질 수도 있습니다.

참고: 이 도구가 작동하려면 선분 스냅 모드를 활성화시켜야만 합니다.

이 도구는 사용자에게 또다른 선분을 연장할지 또는 다듬을지에 대한 제한 (선분) 을 선택할 것을 요청합니다. 꼭짓점 도구와는 달리, 편집 중인 레이어만 수정하는지 검사합니다.

두 선분 모두 3 차원인 경우, 제한 선분이 Z 값을 가지도록 보간합니다.

다듬기의 경우, 클릭해서 짧게 만들 부분을 선택해야만 합니다.

14.5.5 형태 디지타이즈 작업

Shape Digitizing 툴바의 도구들로 정규 형태 및 만곡 도형을 그릴 수 있습니다.

원호 스트링 추가

 Add circular string 또는  Add circular string by radius 버튼으로 원호 도형을 가진 라인 또는 폴리곤 피처를 추가할 수 있습니다.

이 도구들로 피처를 생성하는 작업은 다른 디지타이즈 작업 도구들과 동일한 규칙을 따릅니다: 꼭짓점을 배치하려면 왼쪽 클릭, 도형을 완료하려면 오른쪽 클릭하십시오. 도형을 그리는 동안, 두 원호 스트링 도구 사이를 전환하는 것은 물론선형 도형 도구 로도 전환해서 복합 도형을 생성할 수 있습니다.

참고: 만곡 도형은 호환되는 데이터 제공자에서만 만곡 도형으로 저장됩니다

QGIS 가 모든 편집 가능한 데이터 유형 내에서 만곡 도형을 디지타이즈하는 것을 허용하고 있기는 하지만, 피처를 만곡 도형으로 저장하려면 곡선을 지원하는 (예를 들어 PostGIS, 메모리 레이어, GML 또는 WFS 같은) 데이터 제공자를 사용해야 합니다. 그렇지 않을 경우 QGIS 가 원호를 선분화시킬 것입니다.

원 그리기

원을 그릴 수 있는 도구들의 집합이 있습니다. 다음은 그 도구들의 설명입니다.

원은 원호 스트링으로 변환됩니다. 따라서원호 스트링 추가 에서 설명한대로 데이터 제공자가 허용하는 경우 만곡 도형으로 저장될 것이고, 아닌 경우 QGIS 가 원호를 선분화할 것입니다.

-  Add circle from 2 points: 포인트 2 개로 지름 및 원의 방향을 정의합니다. (왼쪽 클릭, 오른쪽 클릭)
-  Add circle from 3 points: 포인트 3 개가 원의 경계선 상에 있다고 가정하고 원을 그립니다. (왼쪽 클릭, 왼쪽 클릭, 오른쪽 클릭)
-  Add circle from center and a 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.
-  Add circle from 3 tangents: 선분 3 개를 접선으로 하는 원을 그립니다. **선분 스냅 모드를 활성화해야만 한다는 사실을 기억하십시오.** (스냅 허용 오차 및 검색 반경 설정 을 참조하세요.) 접선으로 추가할 선분을 클릭하십시오. 접선 2 개가 평행하는 경우, 오류 메시지를 표시하고 입력을 초기화합니다. (왼쪽 클릭, 왼쪽 클릭, 오른쪽 클릭)
-  Add 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.

-  Add 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)
-  Add Ellipse from center and a point: 중심과 모서리를 지정한 경계 상자에 내접하도록 타원을 그립니다. (왼쪽 클릭, 오른쪽 클릭)
-  Add Ellipse from extent: 서로 바라보는 두 모서리를 지정한 경계 상자에 내접하도록 타원을 그립니다. (왼쪽 클릭, 오른쪽 클릭)
-  Add 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)

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.

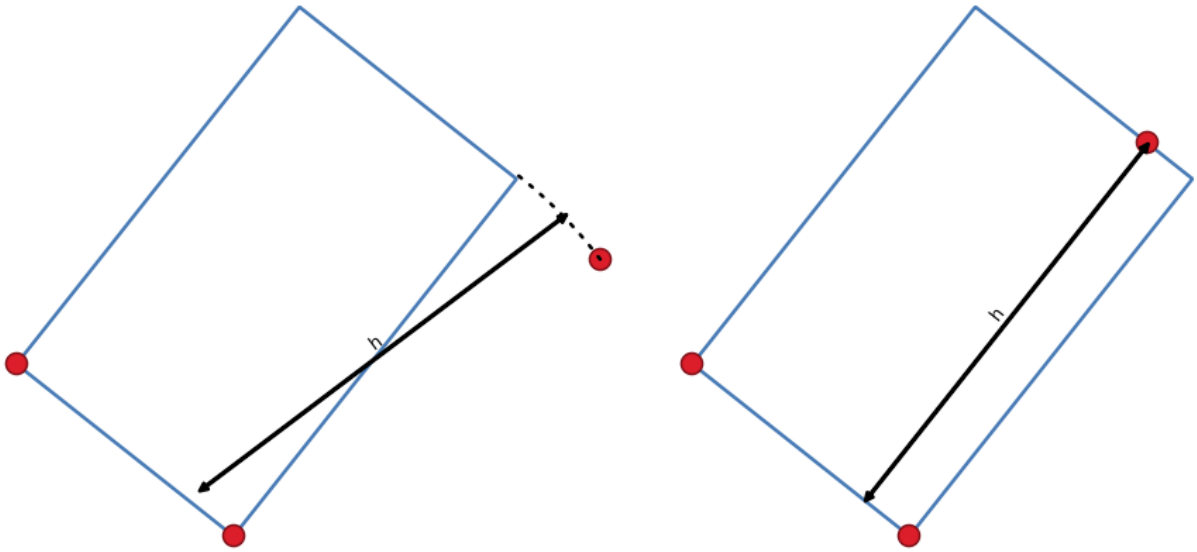


그림 14.94: 포인트 3 개로 직사각형 그리기: 거리 (오른쪽) 와 투영 (왼쪽)

14.5.6 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, 좌표를 직접 입력해서 사용자의 새 도형을 정밀하게 정의할 수도 있습니다.

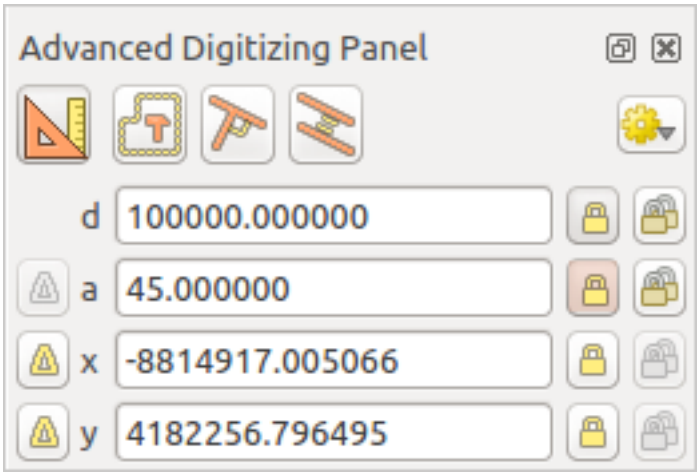


그림 14.95: 고급 디지타이즈 작업 패널

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.

개념

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.

스냅 설정

고급 디지타이즈 작업 도구의 스냅 환경을 설정하려면  아이콘을 클릭하십시오. 이 도구가 공통 각도로 스냅하도록 할 수 있습니다. 옵션은 다음과 같습니다:

- *Do not snap to common angles*
- *Snap to 30º angles*
- *Snap to 45º angles*
- *Snap to 90º angles*

피처에 어떻게 스냅할지도 설정할 수 있습니다. 옵션은 다음과 같습니다:

- *Do not snap to vertices or segments*
- *Snap according to project configuration*
- *Snap to all layers*

키보드 단축키

고급 디지타이즈 작업 패널의 활용 속도를 높이려면, 다음과 같은 키보드 단축키를 사용할 수 있습니다:

키	간단한 설명	`Ctrl+` 또는 `Alt+` 키조합	`Shift+` 키조합
`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	최근 꼭짓점에 상대적인 Y 변경
`C`	작성 모드 변경		
`P`	수직 및 평행 모드 변경		

절대 참조 디지타이즈 작업

아무것도 없는 상태에서 새 도형을 그리는 경우, 지정한 좌표에서 꼭짓점 디지타이즈를 시작할 수 있다면 매우 편리합니다.

예를 들면 폴리곤 레이어에 새 피처를 추가하려면,  아이콘을 클릭하십시오. 피처 편집을 시작하길 바라는 X 및 Y 좌표를 다음과 같이 선택할 수 있습니다:

- *x* 텍스트 란을 클릭합니다. (또는 `X` 단축키를 누릅니다.)
- 원하는 X 좌표를 입력한 다음 `Enter` 키를 누르거나 오른쪽에 있는  아이콘을 클릭하면 맵 캔버스 상의 X 좌표의 축에 마우스가 고정됩니다.
- *y* 텍스트 란을 클릭합니다. (또는 `Y` 단축키를 누릅니다.)

- 원하는 Y 좌표를 입력한 다음 Enter 키를 누르거나 오른쪽에 있는  아이콘을 클릭하면 맵 캔버스 상의 Y 좌표의 축에 마우스가 고정됩니다.

파랑색 점선 2 개와 초록색 가위표가 사용자가 입력한 좌표를 정확히 가리키고 있습니다. 마우스 위치는 초록색 가위표에 고정돼 있습니다. 맵 캔버스를 클릭해서 디지타이즈 작업을 시작하십시오.

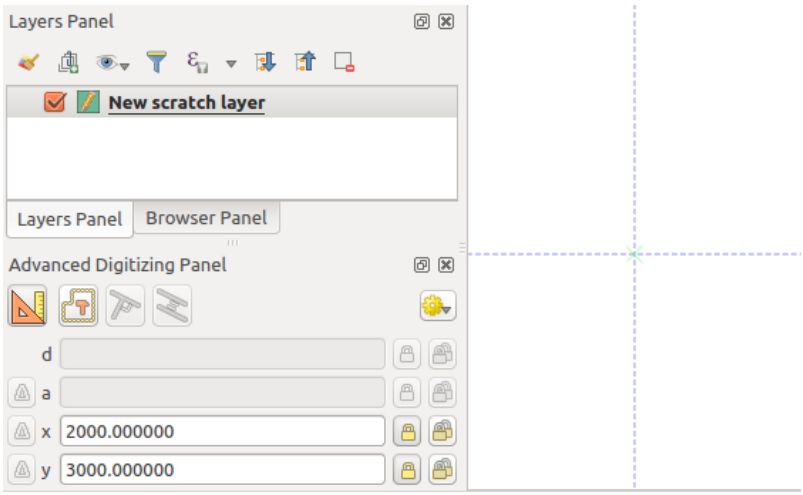


그림 14.96: 지정 좌표에서 그리기 시작

사용자의 손으로 직접 디지타이즈를 계속해서 좌표 몇 쌍을 추가할 수도 있고, 또는 선분의 **길이** (거리) 및 **각도** 를 입력할 수도 있습니다.

지정 길이로 선분을 그리고 싶다면, *d (distance)* 텍스트란을 클릭하고 (또는 D 단축키를 누르고) 거리 값을 (맵 단위로) 입력한 다음 Enter 키를 누르거나 오른쪽에 있는  아이콘을 클릭하면 맵 캔버스에 있는 마우스를 선분의 길이로 고정시킵니다. 맵 캔버스에서는, 거리 텍스트란에 입력한 값이 반경인 원이 클릭한 포인트를 둘러싸고 있습니다.

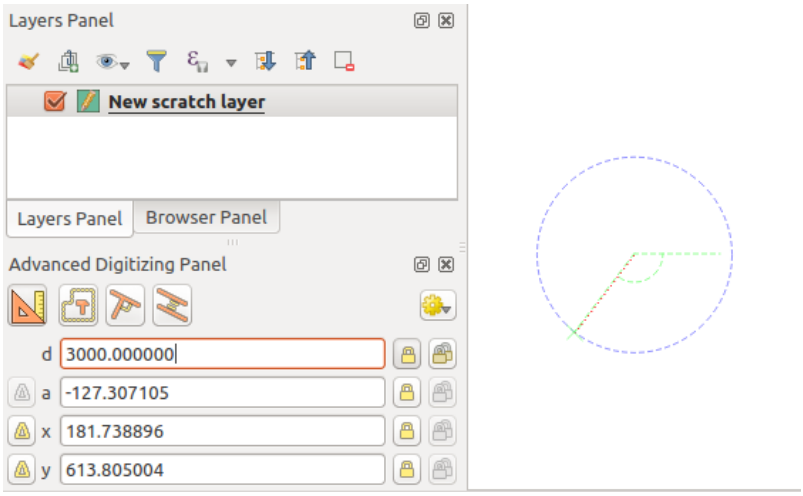


그림 14.97: 고정 길이 선분

마지막으로, 선분의 각도도 설정할 수 있습니다. 앞에서 설명한 바와 마찬가지로, *a (angle)* 텍스트란을 클릭하고 (또는 A 단축키를 누르고) 각도 값을 (도 단위로) 입력한 다음 Enter 키를 누르거나 오른쪽에 있는  아이콘을 클릭하면 각도가 고정됩니다. 이렇게 하면 선분이 원하는 각도대로 꺾일 것입니다.

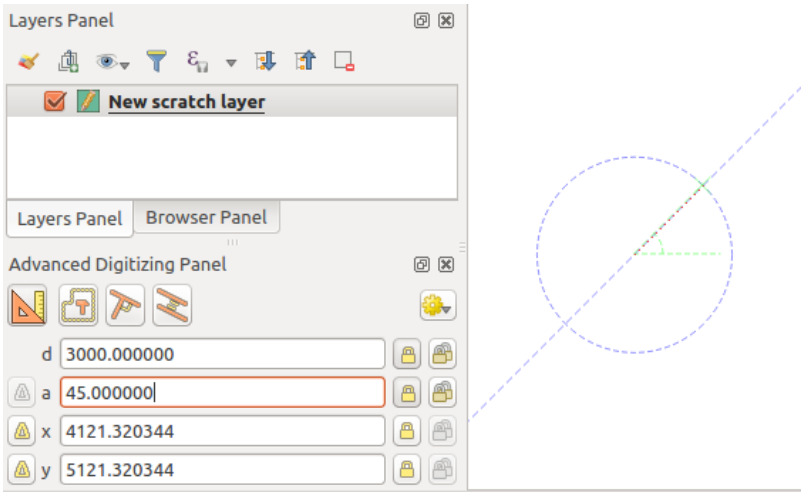


그림 14.98: 고정 각도 선분

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* 또는 *y* 텍스트란 왼쪽에 있는 아이콘을 클릭해서 (혹은 `Shift+X` 또는 `Shift+Y` 조합키를 눌러서) 이전 꼭짓점에 상대적인 좌표를 켜고 끌 수 있습니다. 이 옵션을 켜면, 직전 꼭짓점을 원점으로 삼아 X 및 Y 좌표를 측정합니다.

지속 고정

Continuous lock 버튼을 클릭하면 디지타이즈 작업, 각도, 거리, X 및 Y 좌표에 대한 절대 또는 상대 참조 제약 조건 모두 지속적으로 고정시킬 수 있습니다. 지속 고정을 사용하면 여러 포인트 또는 꼭짓점을 동일한 제약 조건으로 디지타이즈할 수 있습니다.

평행 및 수직 라인

앞에서 설명한 모든 도구들은 Perpendicular 및 Parallel 도구와 결합해서 사용할 수 있습니다. 이 두 도구는 선분을 또다른 선분에 완벽히 직각으로 또는 평행하게 그릴 수 있습니다.

수직 선분을 그리려면, 편집 도중에 Perpendicular 아이콘을 클릭해서 (또는 `P` 단축키를 눌러서) 활성화시키십시오. 수직 라인을 그리기 전에, 직각의 기준이 될 기존 피처의 선분을 클릭하십시오. (기존 피처의 라인이 밝은 주황색으로 강조될 것입니다.) 사용자의 라인이 스냅될 위치에 다음과 같이 파랑색 점선이 보일 겁니다:

평행 라인을 그리는 방법도 동일합니다. Parallel 아이콘을 클릭해서 (또는 `P` 단축키를 두 번 눌러서) 활성화한 다음, 참조 라인으로 사용하려는 선분을 클릭하고 사용자 피처를 다음과 같이 그리십시오:

이 두 도구는 수직 및 평행 각도에 딱 맞는 각도를 찾아서 편집 작업 중에 해당 파라미터를 고정시킵니다.

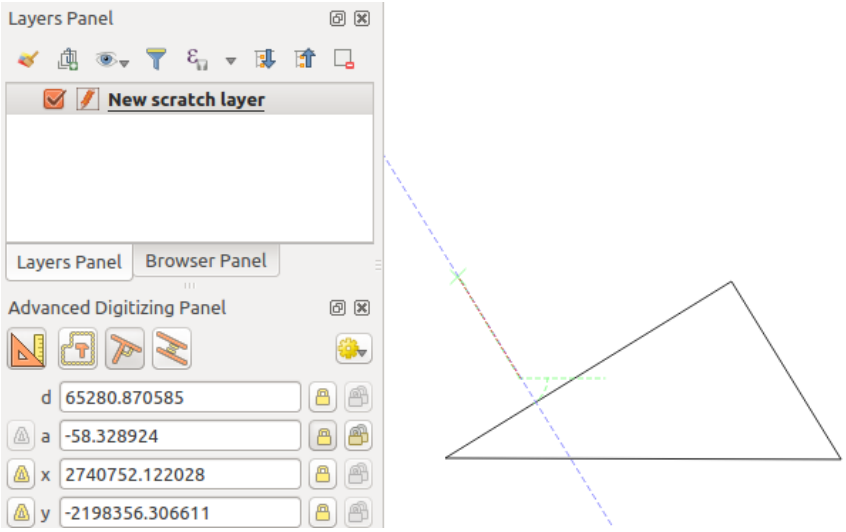


그림 14.99: 수직 디지타이즈 작업

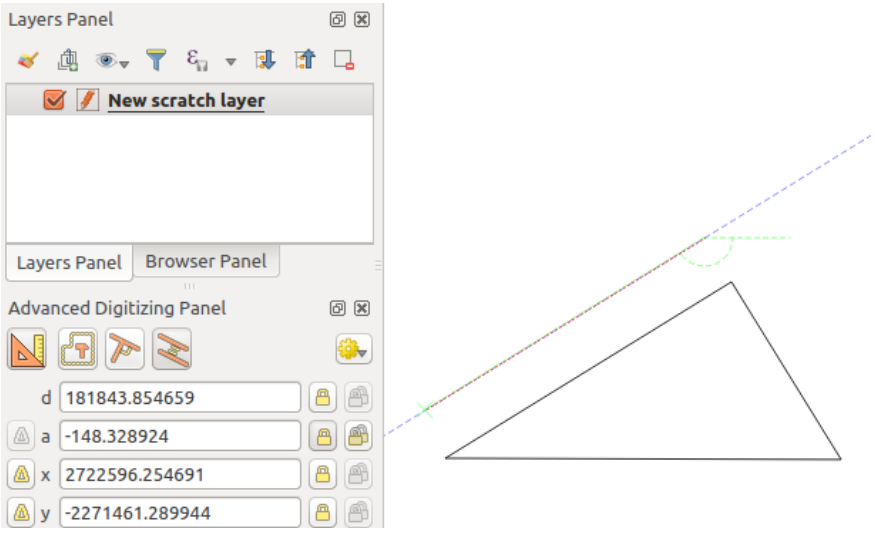


그림 14.100: 평행 디지타이즈 작업

작성 모드

Construction 아이콘을 클릭하거나 또는 C 단축키를 누르면 작성 모드를 활성화/비활성화할 수 있습니다. 작성 모드 상태에서 맵 캔버스를 클릭하면 새 꼭짓점을 추가하지는 않지만, 클릭한 위치를 캡처해서 이후 거리, 각도, 또는 X 및 Y 좌표의 상대값을 고정시킬 참조 포인트로 이용할 수 있습니다.

예를 들어, 기존 포인트에서 정확히 동일한 거리에 있는 포인트를 몇 개 그리는 데 작성 모드를 사용할 수 있습니다.

맵 캔버스에 있는 기존 포인트와 올바르게 활성화된 스냅 모드를 통해, 기존 포인트에서 지정 거리 및 각도에 있는 다른 포인트를 쉽게 그릴 수 있습니다. 버튼을 클릭한 다음, Construction 아이콘을 클릭하거나 C 단축키를 눌러서 작성 모드도 활성화시켜야 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:

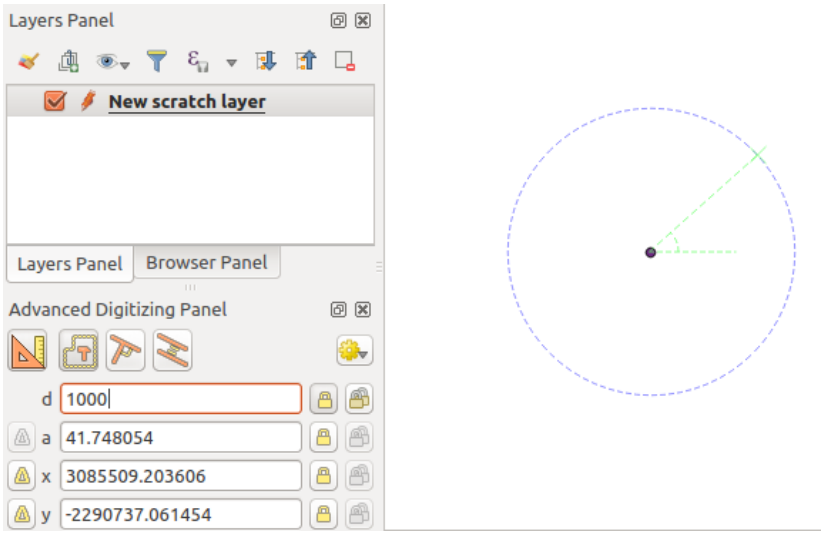


그림 14.101: 점거리

새 포인트를 추가하기 전에, C 단축키를 눌러 작성 모드를 끄십시오. 이제 맵 캔버스를 클릭하면, 입력한 거리에 포인트가 배치될 것입니다.

예를 들자면 원점 포인트로부터 동일한 거리에 있지만, 새로 추가한 포인트에서 특정 각도에 있는 다른 포인트를 생성하는 데 각도 제약조건을 사용할 수도 있습니다. Construction 아이콘을 클릭하거나 C 단축키를 눌러서 작성 모드를 켜십시오. 새로 추가한 포인트를 클릭한 다음 다른 포인트도 클릭해서 방향 선분을 설정하십시오. 다음으로 *d* 텍스트란을 클릭하고 (또는 D 단축키를 누르고) 원하는 거리를 입력한 다음 Enter 키를 누르십시오. *a* 텍스트란을 클릭하고 (또는 A 단축키를 누르고) 원하는 각도를 입력한 다음 Enter 키를 누르십시오. 마우스 위치의 거리 및 각도 둘 다 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.

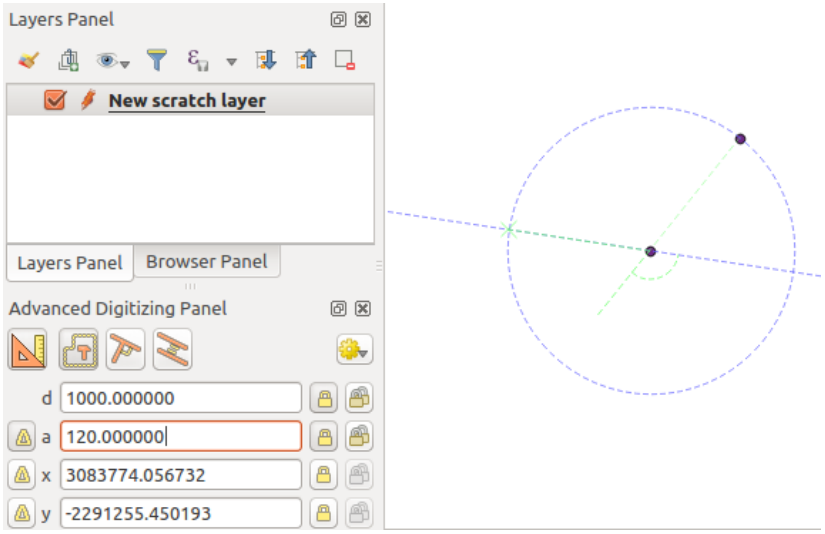


그림 14.102: 포인트들로부터의 거리 및 각도

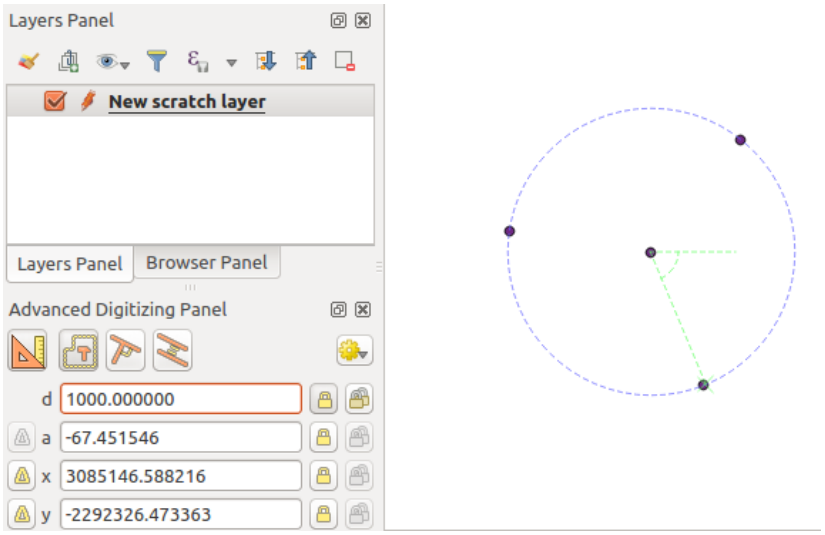


그림 14.103: 지정 거리 및 각도에 있는 포인트들

14.5.7 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.

   ---

   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.

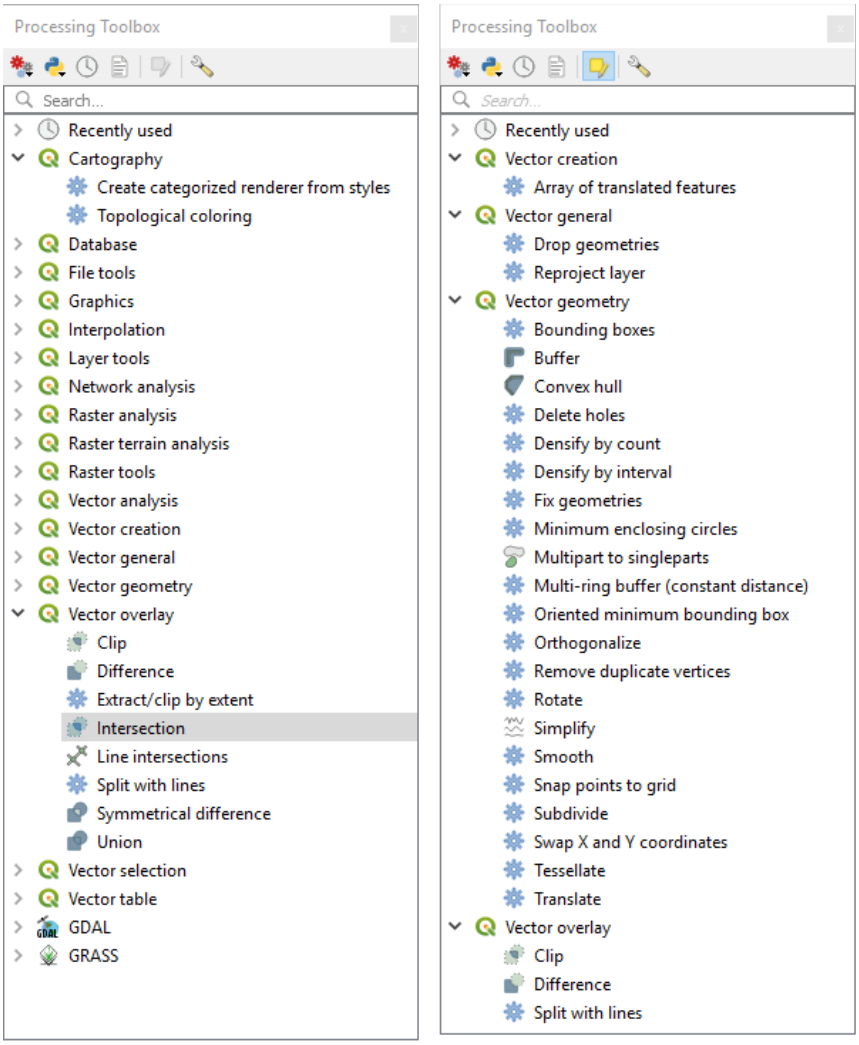


그림 14.104: 공간 처리 알고리즘: 전체 (왼쪽) 대 제자리 폴리곤 편집기 (오른쪽)

CHAPTER 15

래스터 데이터 작업

15.1 래스터 속성 대화창

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
-  심볼
-  투명도
-  히스토그램
-  렌더링
-  피라미드
-  메타데이터
-  범례
-  *QGIS* 서버

팁: 실시간 렌더링 업데이트

레이어 스타일 작업 패널 은 레이어 속성 대화창의 공통 기능 가운데 일부를 제공하며, 레이어 스타일 환경 설정을 빠르게 할 수 있고 맵 캔버스에서 사용자 살펴볼 수 있는 훌륭한 모달리스 위젯입니다.

참고: 원본 프로젝트 파일에서 삽입 레이어 (프로젝트 내포 작업 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.

15.1.1 정보 속성

Information 탭은 읽기 전용으로, 현재 레이어에 대한 요약 정보 및 메타데이터를 한 눈에 볼 수 있는 흥미로운 장소입니다. 다음과 같은 정보를 제공합니다:

- 레이어 제공자 기반 정보 (저장소 유형, 경로, 데이터 유형, 범위, 너비/높이, 압축 여부, 픽셀 크기, 밴드 통계, 열/행 개수, 래스터의 NODATA 값 등등)
- 제공된 메타데이터 에서 추출한 정보: 접근 정보, 링크, 연락처, 이력 등은 물론 데이터셋 정보 (좌표계, 범위, 밴드 등등) 까지

15.1.2 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.

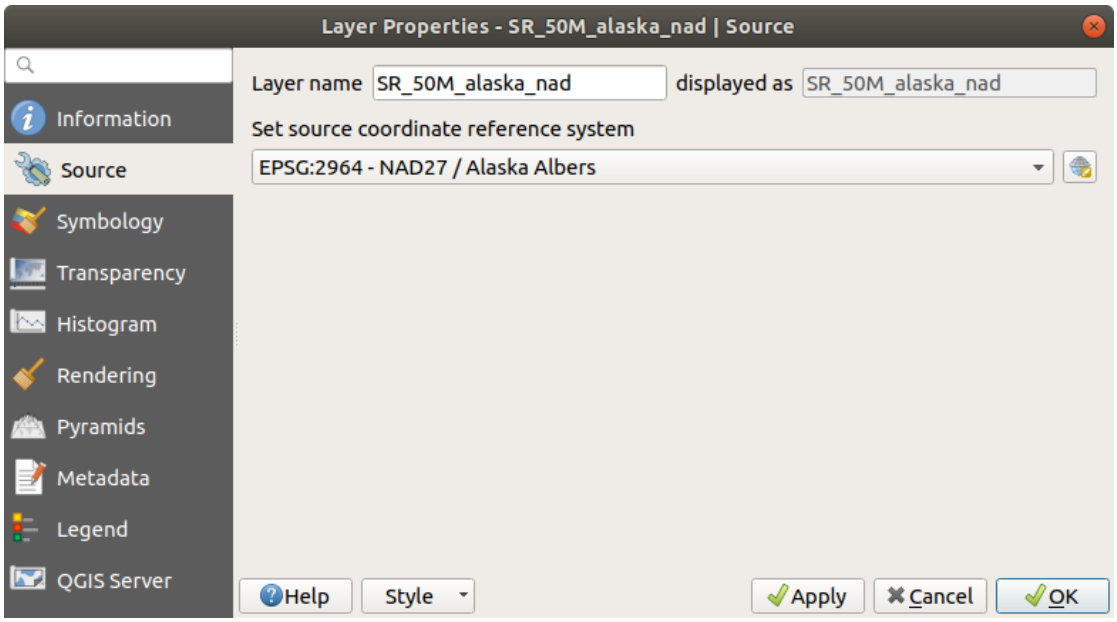


그림 15.1: 래스터 레이어 속성 — 소스 대화창

15.1.3 심볼 속성

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: 여러 밴드를 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. 음영기복: 밴드로부터 음영기복을 생성합니다.
6. 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.

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.

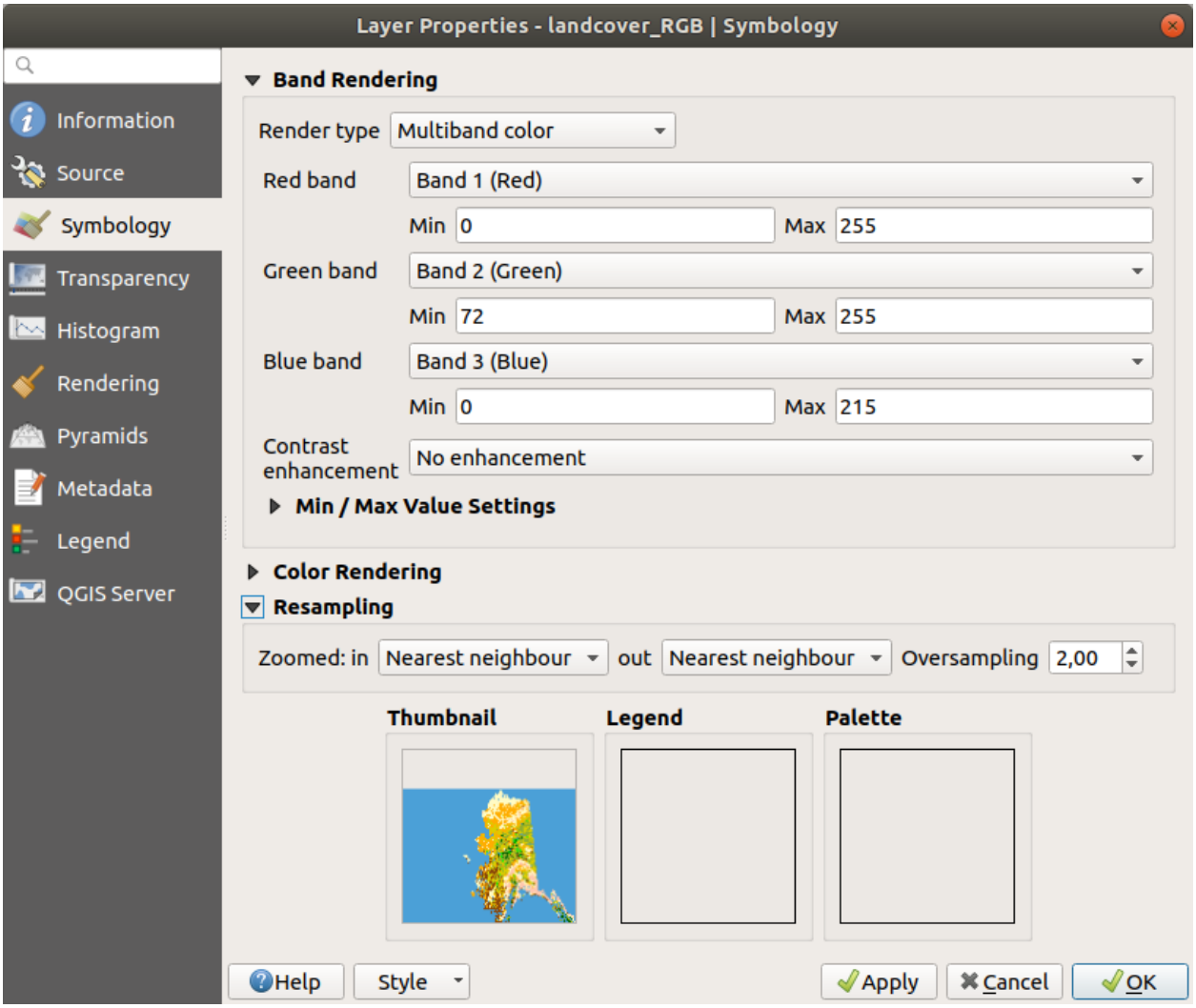


그림 15.2: 래스터 심볼 — 다중 밴드 색상 렌더링

- *Change Opacity*…선택 집합의 투명도를 변경합니다.
- *Change Label*…: 선택 집합의 라벨을 변경합니다.

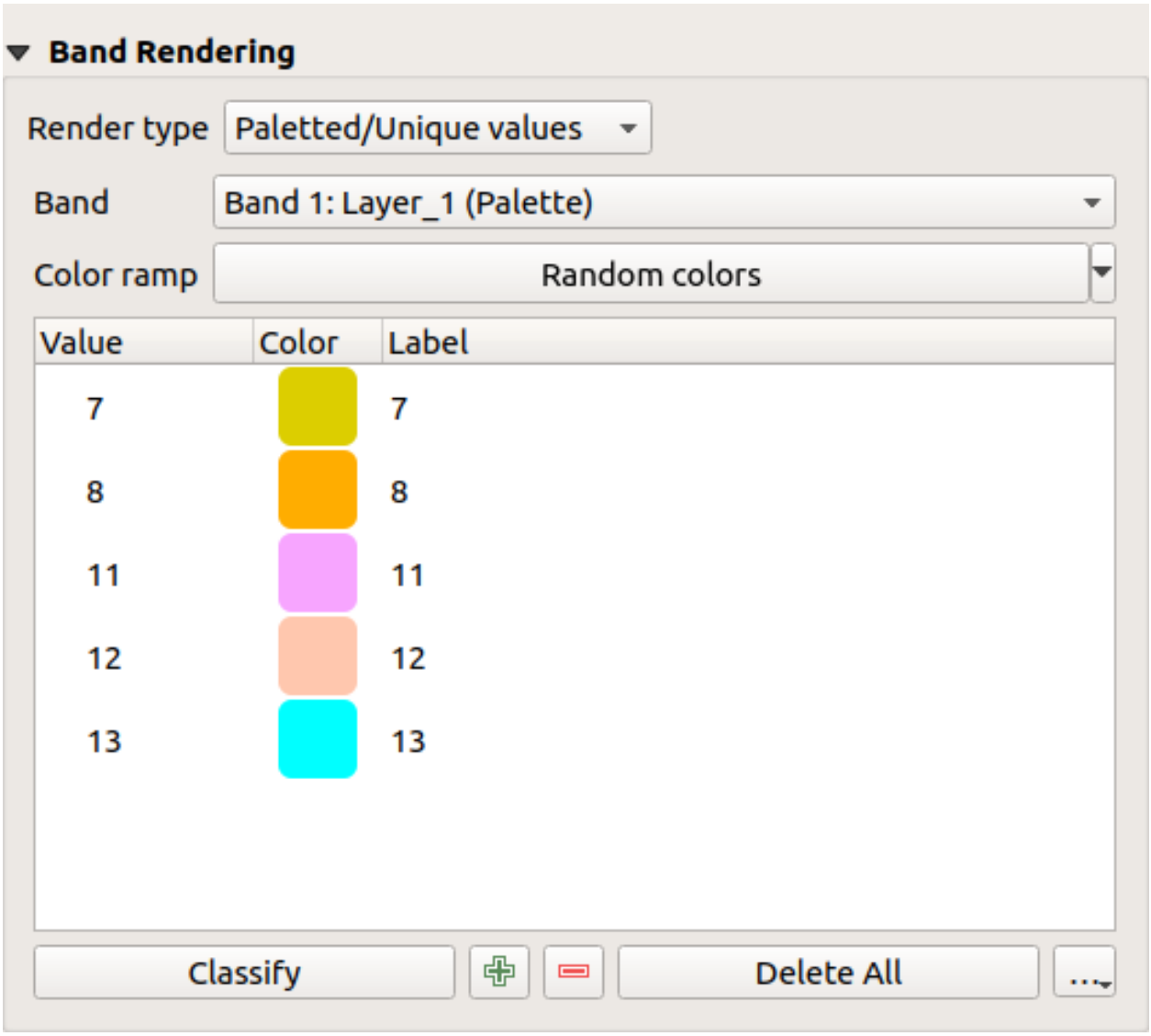


그림 15.3: 래스터 심볼 — 팔레트/유일값 렌더링

색상표 아래 오른쪽에 있는 … (*Advanced options*) 버튼을 클릭하면 열리는 풀다운 메뉴는 색상표를 불러오고 (*Load Color Map from File*…) 내보낼 수 있으며 (*Export Color Map to File*…), 또 범주를 불러올 수도 (*Load Classes from Layer*) 있습니다.

단일 밴드 회색조

이 렌더링 작업자는 *Color gradient* 옵션을'Black to white'또는'White to black'가운데 선택해서 단일 밴드 레이어를 렌더링할 수 있습니다. 최소/최대 값 설정 에서 색상 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.

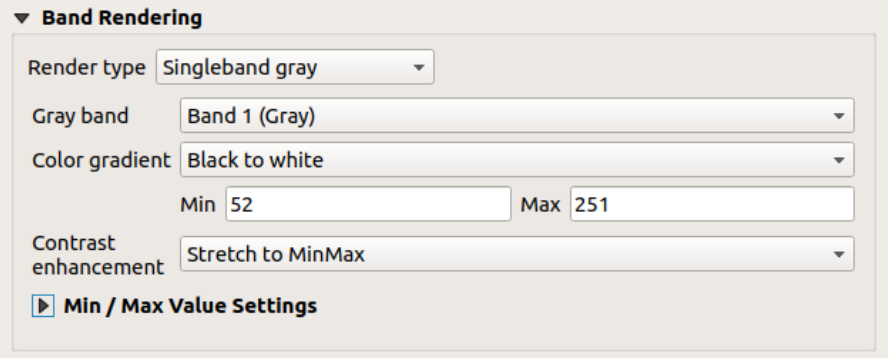


그림 15.4: 래스터 심볼 — 단일 밴드 회색조 렌더링

단일 밴드 의사색상

연속적인 팔레트를 포함하는 단일 밴드 파일 용 렌더링 옵션입니다. 다중 밴드 래스터의 밴드들을 위한 색상 표도 생성할 수 있습니다.

레이어의 *Band* 와값의 범위 를 사용해서, 세 가지 유형의 색상 *Interpolation* 방법을 사용할 수 있습니다:

- 불연속 (Discrete; *Value* 열의 헤더에 <= 심볼이 나타납니다.)
- 선형 (Linear)
- 엄밀 (Exact; *Value* 열의 헤더에 = 심볼이 나타납니다.)

Color ramp 드롭다운 목록은 사용할 수 있는 색상표를 담고 있습니다. 새 색상표를 생성하고 현재 선택한 색상표를 편집하거나 저장할 수 있습니다. 색상표 명칭은 환경 설정 및 QML 파일에 저장될 것입니다.

Label unit suffix 는 범례에 있는 값 뒤에 추가되는 라벨입니다.

범주화 *Mode* 를  'Equal interval'로 선택하면, *number of classes*  에서 단계 개수를 선택한 다음 *Classify* 버튼을 누르기만 하면 됩니다. *Mode* 를  'Continuous'로 선택한 경우, QGIS 는 *Min* 과 *Max* 값에 따라 단계를 자동적으로 생성합니다.

 Add values manually 아이콘은 색상표에 값을 추가합니다.  Remove selected row 아이콘은 색상표에서 값을 삭제합니다. *Value* 열을 더블클릭하면 특정 값을 삽입할 수 있고, *Color* 열을 더블클릭하면 해당 값에 적용할 색상을 선택할 수 있는 *Change color* 대화창이 열립니다. 또 각 색상마다 라벨도 추가할 수 있지만, 피처 식별 도구 사용 시 이 값은 표시되지 않습니다.

색상표에서 선택한 행들을 오른쪽 클릭하면 다음 옵션을 가진 컨텍스트 메뉴가 나타납니다:

- *Change Color*…: 선택 집합의 색상을 변경합니다.
- *Change Opacity*…선택 집합의 투명도를 변경합니다.

 Load color map from file 또는  Export color map to file 버튼을 이용해서 기존 색상표를 불러오거나 또는 다른 세션을 위해 색상표를 저장할 수도 있습니다.

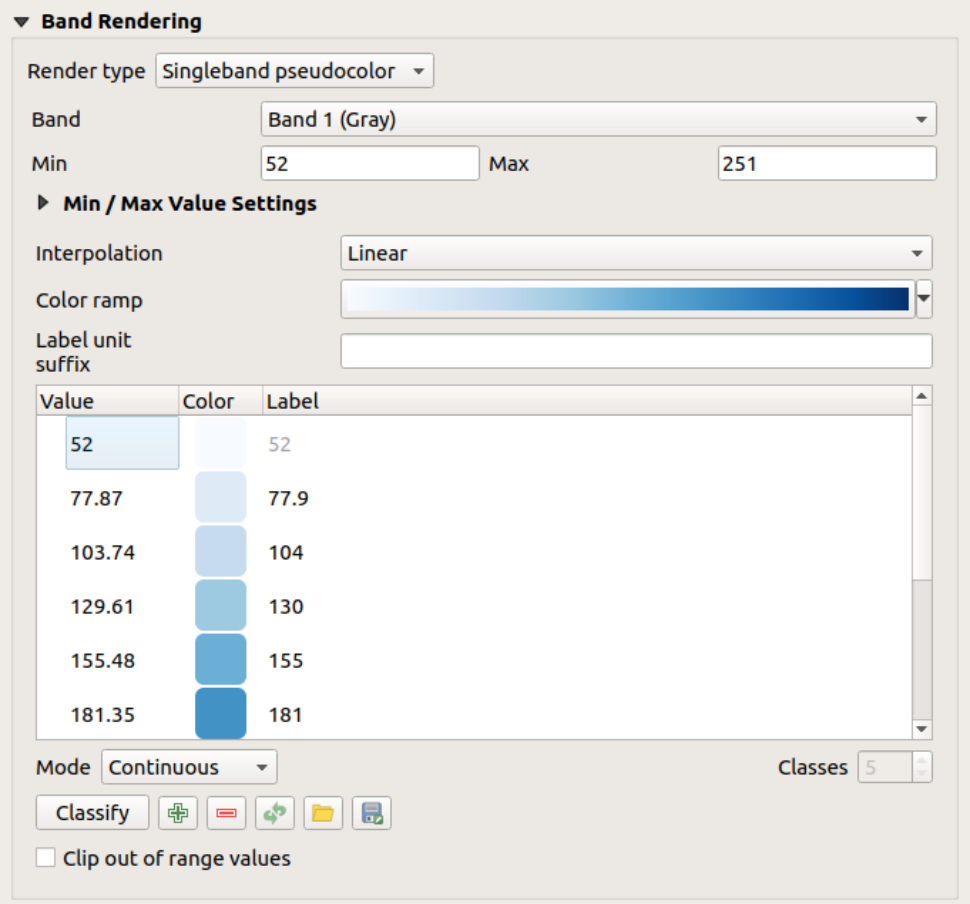


그림 15.5: 래스터 심볼 — 단일 밴드 의사색상 렌더링

☑ *Clip out of range values* 옵션은 QGIS 가 *Max* 값을 초과하는 픽셀을 렌더링하지 않도록 합니다.

음영기복

음영기복을 사용해서 래스터 레이어의 밴드를 렌더링합니다.

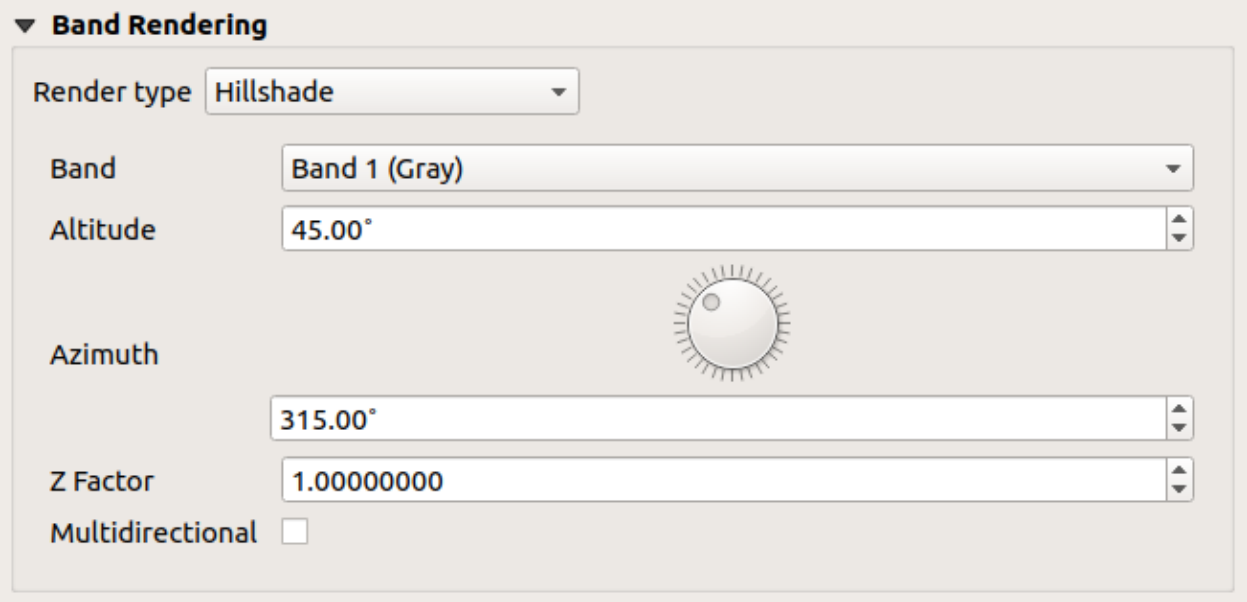


그림 15.6: 래스터 심볼 — 음영기복 렌더링

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.

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

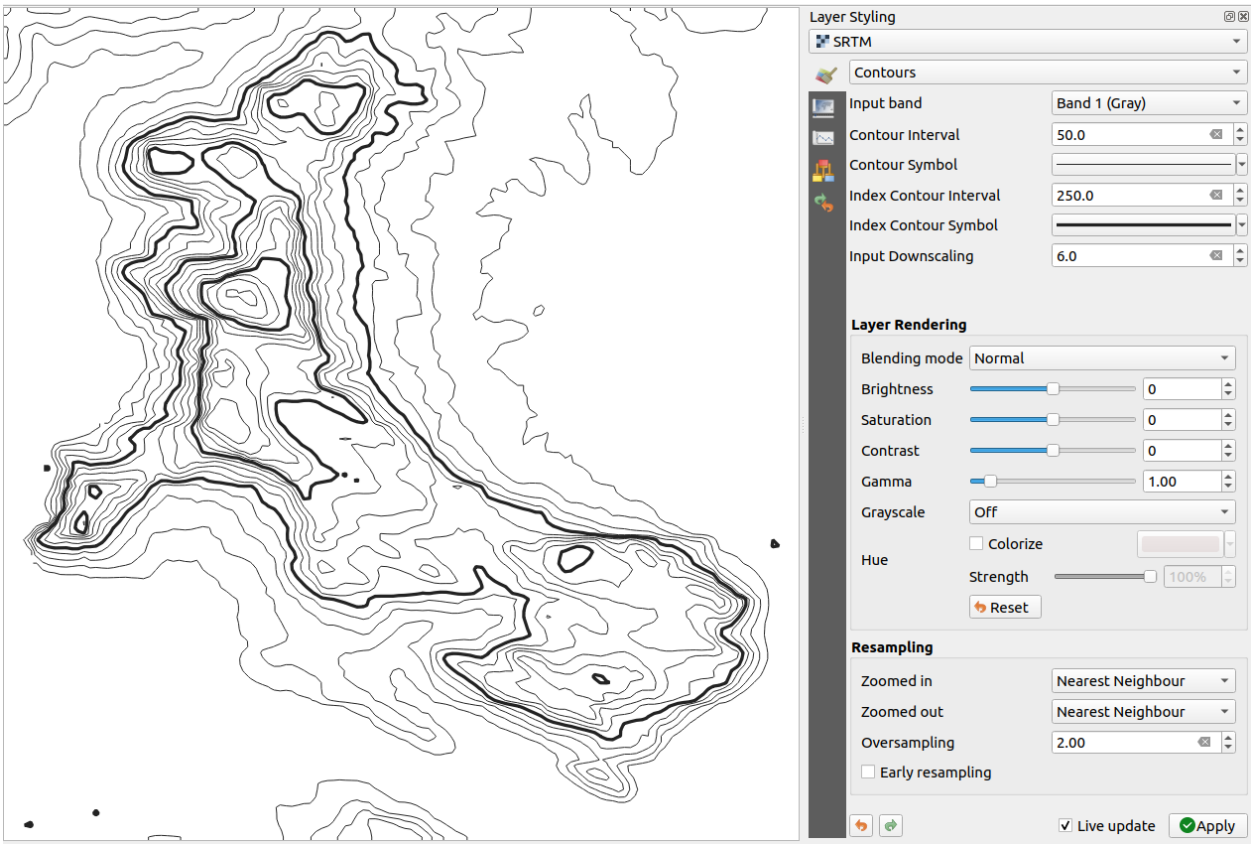


그림 15.7: 래스터 심볼 - 등고선 렌더링

- *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.

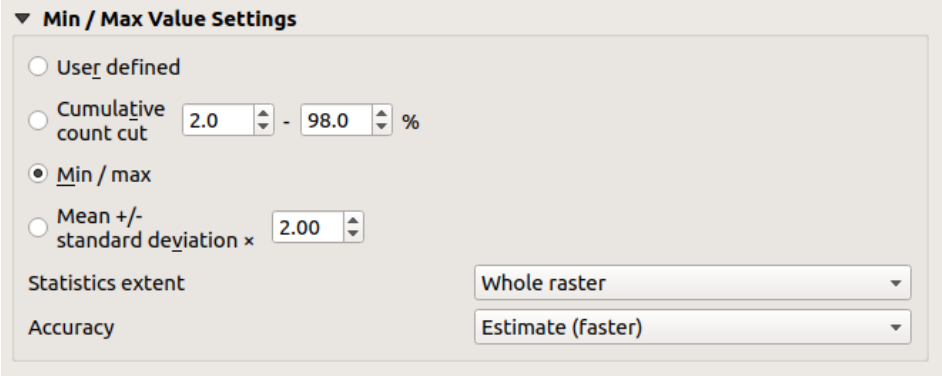


그림 15.8: 래스터 심볼 — 최소/최대 값 설정하기

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.

색상 렌더링

모든 *Band rendering* 에 대해, *Color rendering* 도 설정할 수 있습니다.

혼합 모드 (혼합 모드 참조) 가운데 하나를 이용해서 사용자 래스터 파일 (들) 을 위한 특수 렌더링 효과를 줄 수 있습니다.

Brightness 옵션으로 명도를, *Saturation* 옵션으로 채도를, *Gamma* 옵션으로 감마값을, *Contrast* 옵션으로 대조를 조정해서 색상을 심화 설정할 수 있습니다. 또 *Grayscale* 옵션도 사용할 수 있는데, 'Off', 'By lightness', 'By luminosity'및'By average'가운데 하나를 선택할 수 있습니다. 색상표에 있는 *Hue* (색채) 하나를 선택한 다음 해당 색채의'강도 (strength)'를 조정할 수 있습니다.

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.

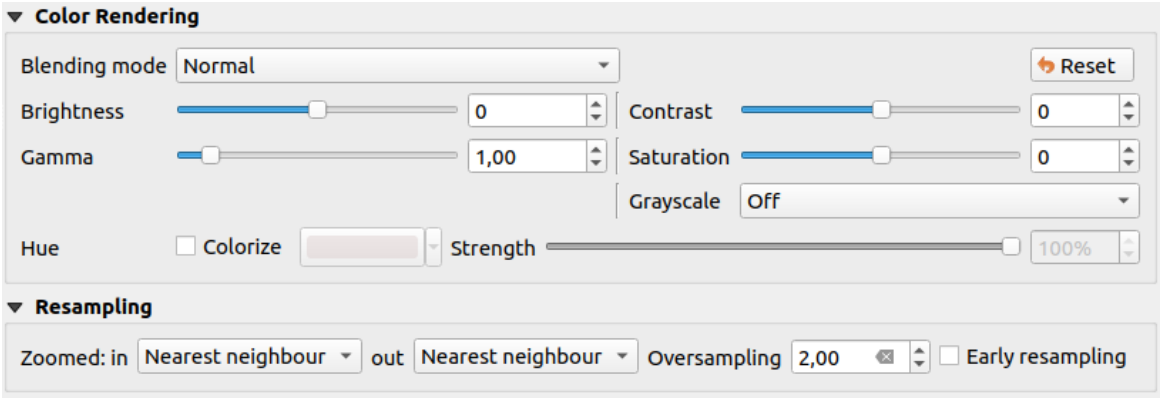


그림 15.9: 래스터 심볼 — 색상 렌더링 및 리샘플링 설정들

'최근접 이웃 (Nearest neighbour)'방법을 적용하면, 확대 시 맵의 픽셀이 두드러져 보일 수 있습니다. '이중선형 (Bilinear)'또는'3 차 (Cubic)'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.

Symbology 탭 하단에서 레이어의 섬네일, 범례 심볼 그리고 팔레트를 볼 수 있습니다.

15.1.4 투명도 속성

QGIS 는 래스터 레이어의 투명도 수준을 설정할 수 있습니다. 아래 있는 레이어가 (있을 경우) 현재 래스터 레이어를 뚫고 비쳐 보이는 정도를 정하려면 투명도 슬라이드 바 를 이용하십시오. 래스터 레이어들을 중첩하는 하는 (예를 들어 음영기복도 위에 범주 래스터 맵을 중첩하는) 경우 매우 유용한 기능입니다. 이렇게 하면 맵이 좀 더 3 차원적으로 보이게 됩니다.

또한 *Additional no data value* 옵션에서, NODATA 로 처리돼야 할 추가적인 래스터 값을 입력할 수 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.

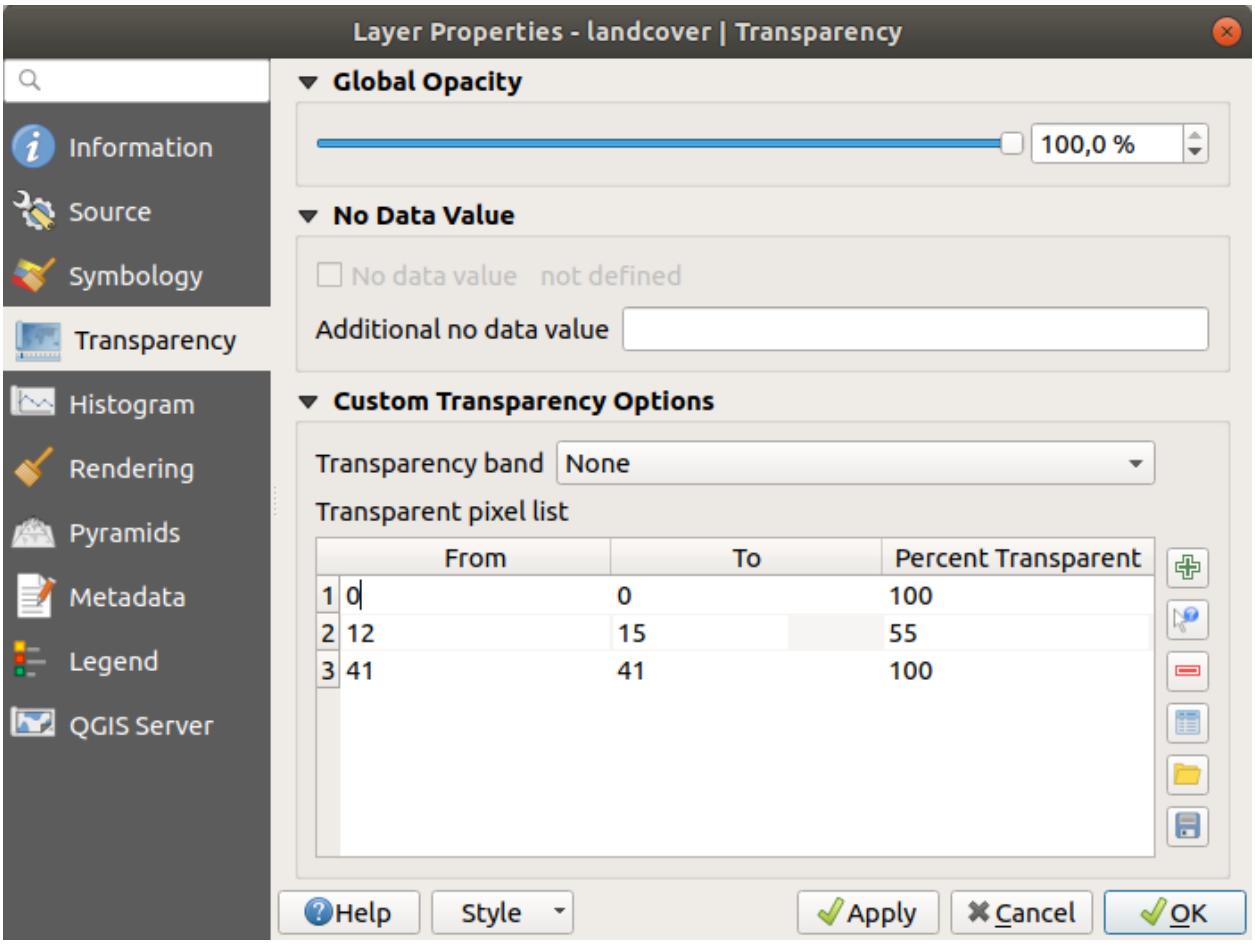


그림 15.10: 래스터 투명도

2. 픽셀의 **빨간색**, **초록색**, **파란색** 값을 입력한 다음 적용할 **투명도 백분율** 을 조정하십시오.
3. 그 대신에,  Add values from display 버튼으로 래스터에서 픽셀 값을 직접 가져올 수도 있습니다. 그 다음 투명도 값을 입력하십시오.
4. 더 많은 값을 사용자 지정 투명도로 조정하려면 앞 단계를 반복하십시오.
5. *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.

15.1.5 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.

15.1.6 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 한계값으로 설정할 수 있습니다. 자세한 내용은축척에 따른 렌더링 을 참조하세요.
- *Refresh layer at interval (seconds)*: 개별 레이어들을 자동으로 새로고침하는 타이머를 설정합니다. 하나 이상의 레이어에 자동 업데이트 간격을 설정한 경우 여러번 새로고침하지 않도록 캔버스 업데이트를 연기합니다.

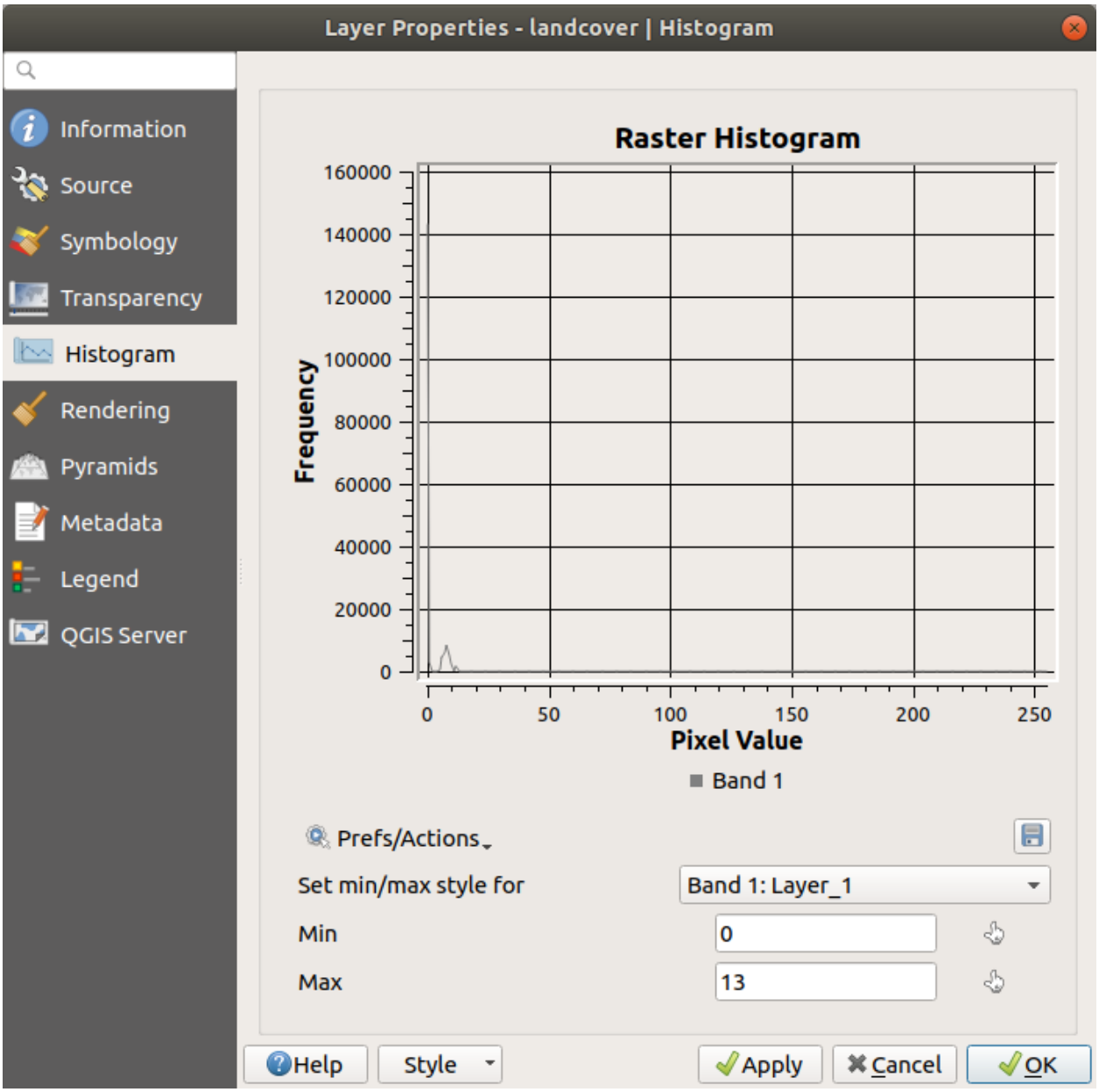


그림 15.11: 래스터 히스토그램

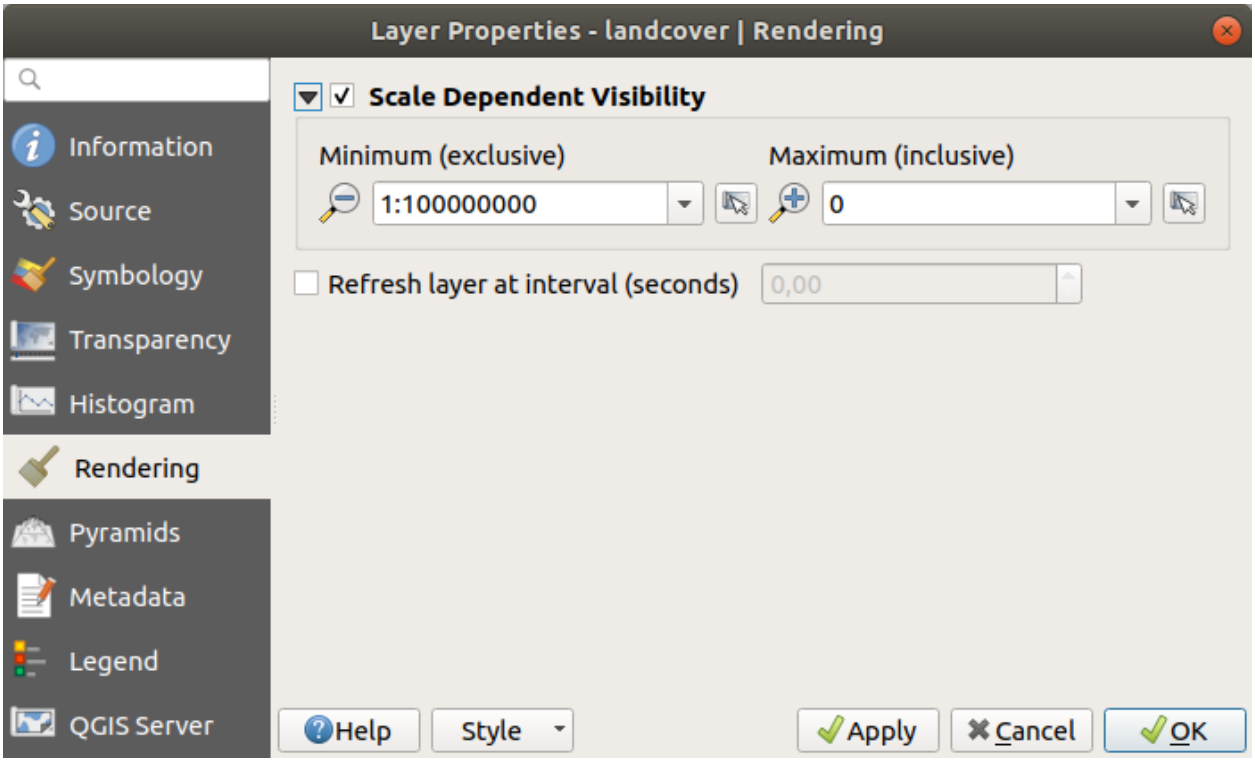


그림 15.12: 래스터 렌더링

15.1.7 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 할 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)
- 가우스 (Gauss)
- 3 차 (Cubic)

- 3 차 스플라인 (Cubic Spline)
- 란초시 (Laczos)
- 모드 (Mode)
- 없음 (None)

마지막으로, *Build Pyramids* 버튼을 클릭해서 생성 과정을 시작하십시오.

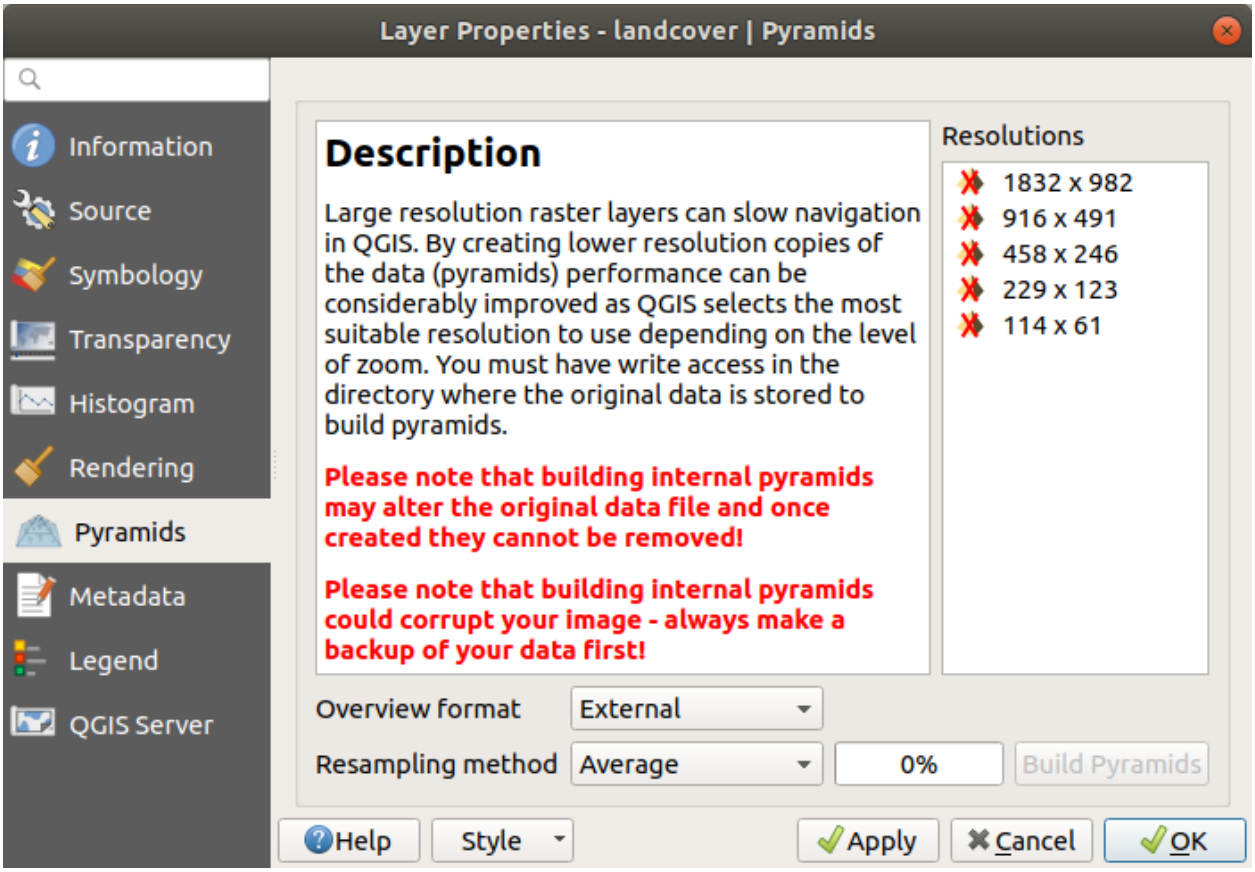


그림 15.13: 래스터 피라미드

15.1.8 메타데이터 속성

Metadata 탭은 사용자 레이어에 대한 메타데이터 보고서를 생성하고 편집할 수 있는 옵션을 제공합니다. 더 자세한 내용은벡터 레이어 메타데이터 속성 을 참조하세요.

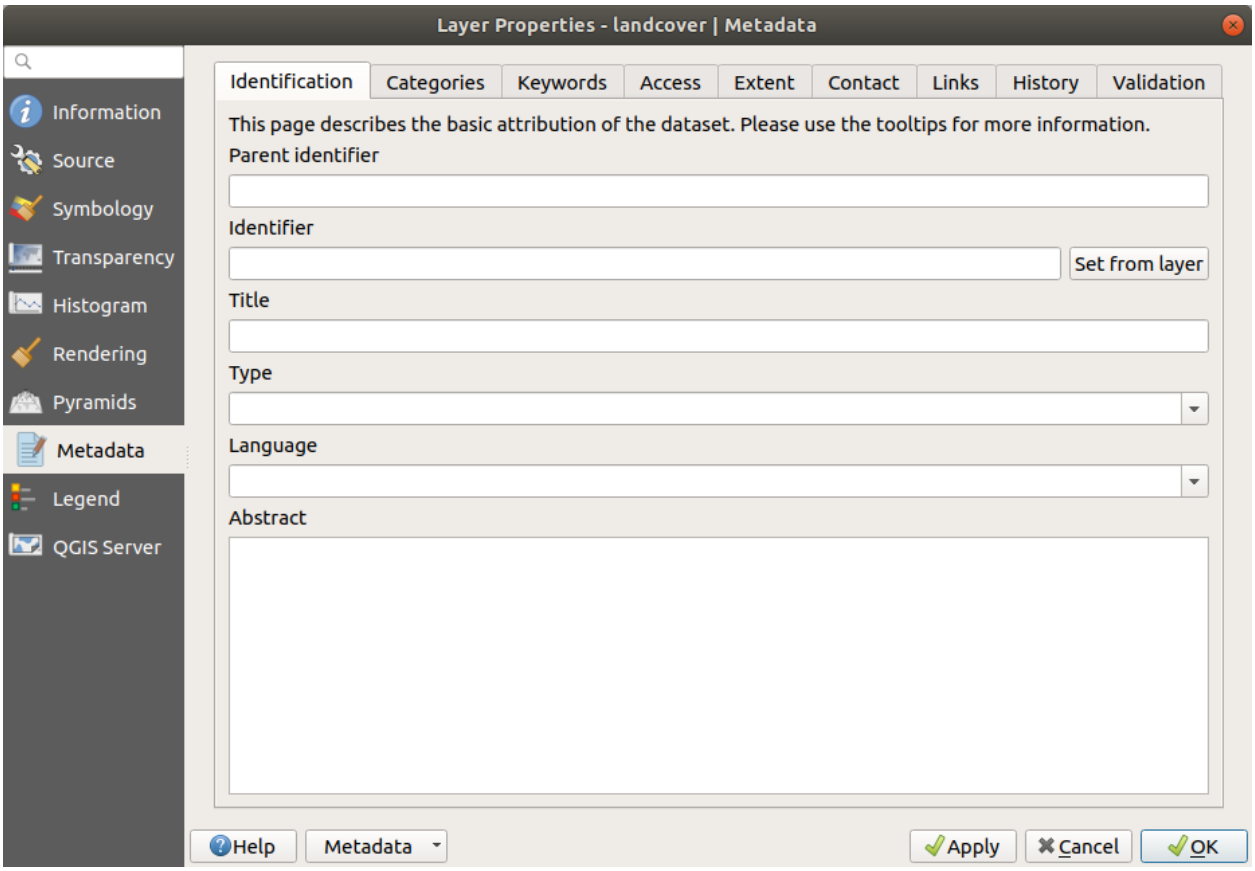


그림 15.14: 래스터 메타데이터

15.1.9 범례 속성

Legend 탭은 레이어 패널에 있는 레이어 트리 안에 삽입할 수 있는 위젯 목록을 표시합니다. 레이어 작업 시 자주 사용되는 몇몇 액션 (투명도, 필터링, 선택, 스타일, 기타 등등의 설정) 에 빨리 접근하기 위한 방법입니다.

QGIS 가 기본적으로 투명도 위젯을 제공하고는 있지만, 자체 위젯을 등록하고 플러그인이 관리하는 레이어에 사용자 지정 액션을 할당하는 플러그인으로 위젯 목록을 확장할 수 있습니다.

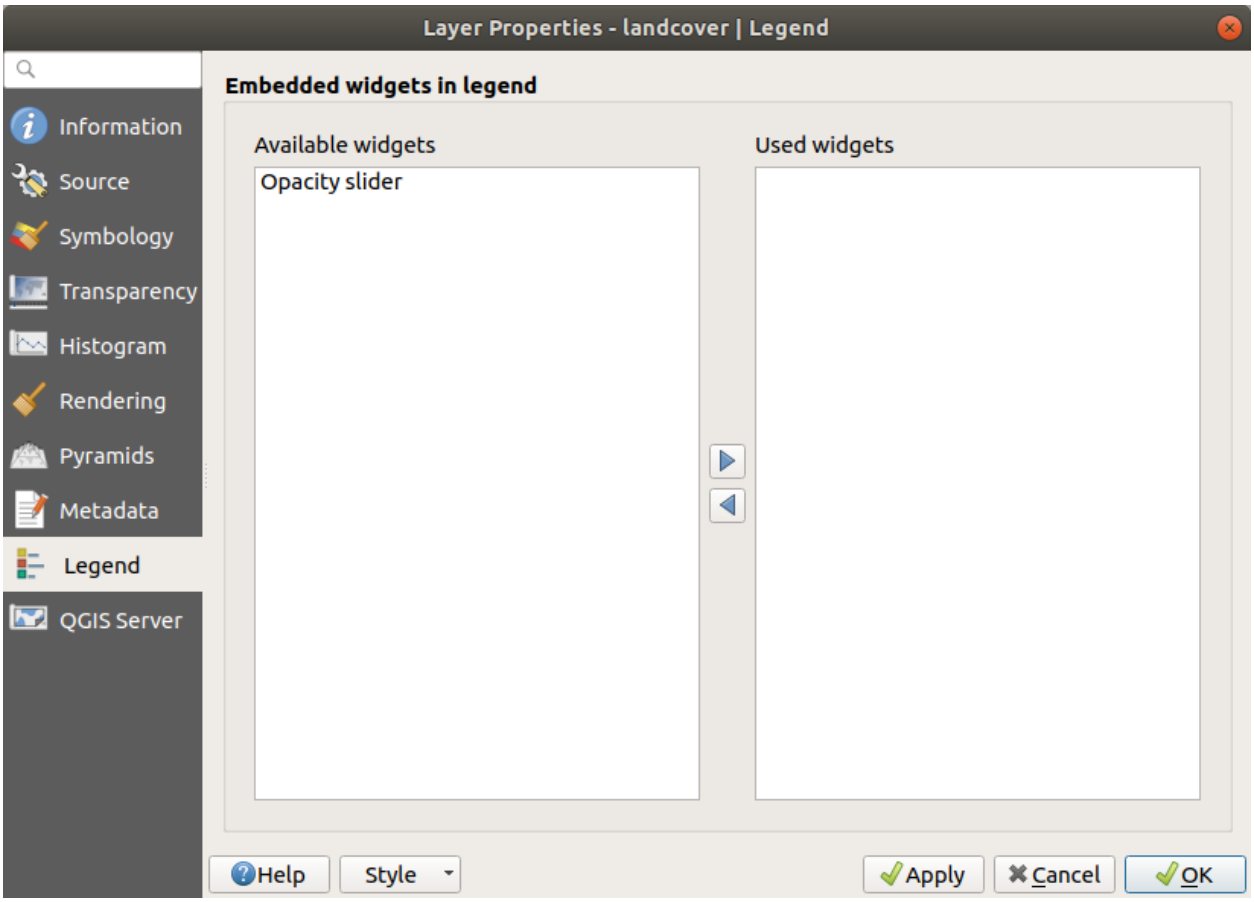


그림 15.15: 래스터 범례

15.1.10 QGIS 서버 속성

QGIS Server 탭에서 *Description*, *Attribution*, *MetadataUrl* 그리고 *LegendUrl* 에 대한 정보를 제공할 수 있습니다.

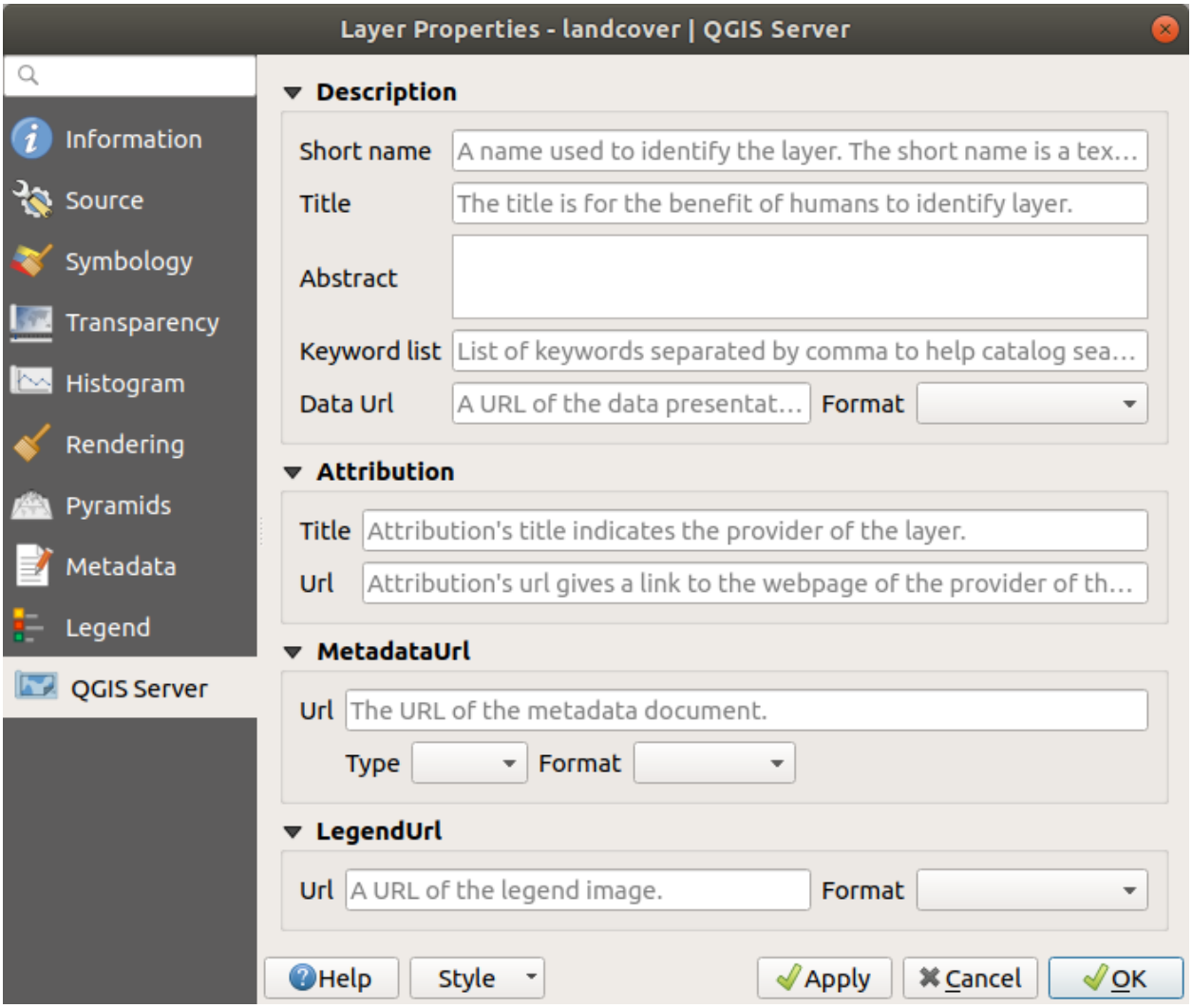


그림 15.16: 래스터 속성에서의 QGIS 서버

15.2 래스터 분석

15.2.1 래스터 계산기

Raster 메뉴에 있는 *Raster Calculator* 는 기존 래스터 픽셀값을 기반으로 계산을 수행할 수 있습니다. (그림 15.17 참조) 계산 결과는 GDAL 지원 포맷의 새 래스터 레이어로 작성됩니다.

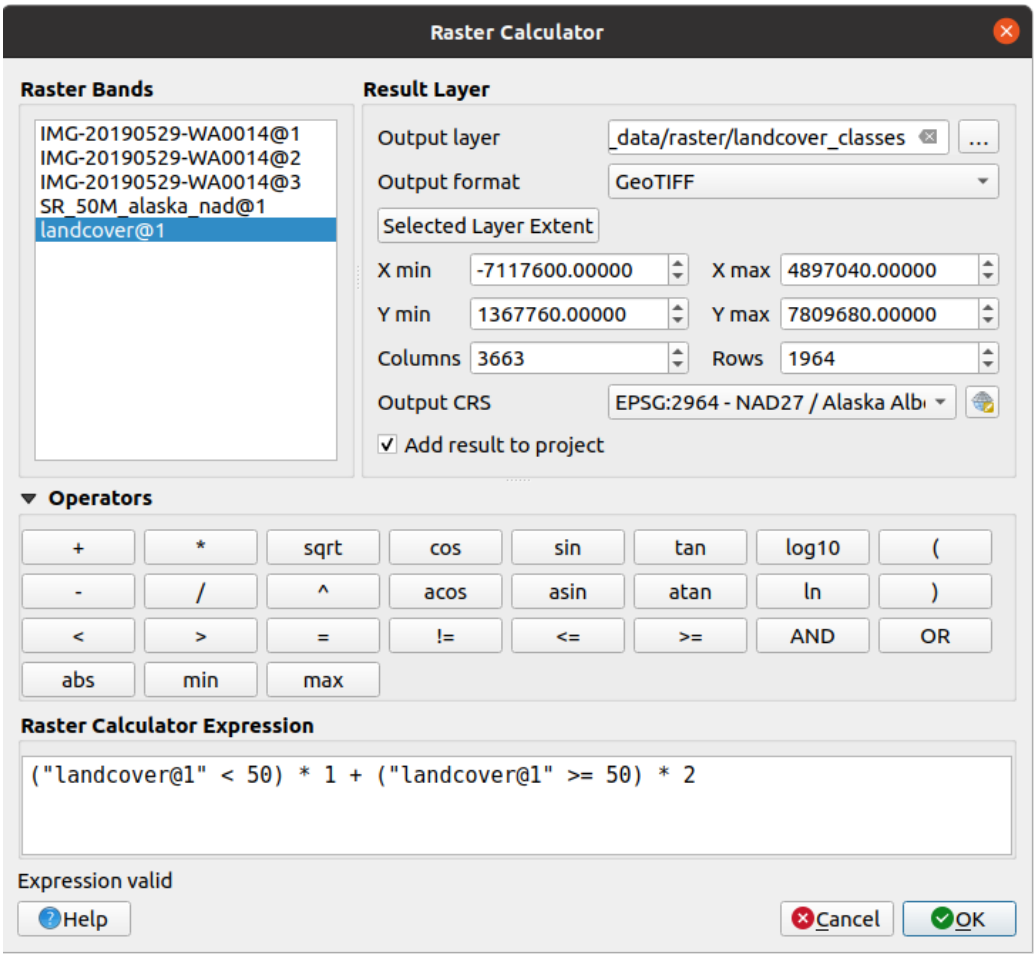


그림 15.17: 래스터 계산기

Raster bands 목록은 사용할 수 있는 모든 불러온 래스터 레이어들을 담고 있습니다. 래스터 계산기 표현식 란에 래스터를 추가하려면, 필드 목록에 있는 명칭을 더블클릭하십시오. 그 다음 연산자를 이용해서 계산 표현식을 작성하거나, 또는 그냥 표현식 란에 계산 표현식을 직접 입력할 수도 있습니다.

먼저 **Result layer** 부분에서 산출 레이어를 정의해야 합니다. 그 다음 입력 레이어를 기반으로, 또는 X, Y 좌표 및 열과 행을 기반으로 계산 영역의 범위를 정의해서 산출 레이어의 해상도를 설정할 수 있습니다. 입력 레이어가 서로 다른 해상도인 경우, 'Nearest Neighbor'알고리즘으로 값들을 리샘플링할 것입니다.

Operators 부분은 사용할 수 있는 모든 연산자를 담고 있습니다. 래스터 계산기 표현식 란에 연산자를 추가하려면, 알맞은 버튼을 클릭하십시오. 수학 계산 (+, –, * 등등) 및 삼각함수 (`sin`, `cos`, `tan` 등등) 를 사용할 수 있습니다. 조건 표현식 (=, !=, <, >= 등등) 은 거짓인 경우 0, 참인 경우 1 을 반환하기 때문에 다른 연산자 및 함수와 함께 사용할 수 있습니다.

Add result to project 체크박스를 활성화하면, 범례 영역에 산출 레이어를 자동적으로 추가하므로 사용자가 가시화할 수 있습니다.

힌트: 래스터 계산기 알고리즘도 참조하세요.

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.

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.

15.2.2 래스터 정렬

이 도구는 여러 래스터를 입력받아 완벽하게 정렬할 수 있습니다. 무슨 뜻이냐 하면:

- 동일한 좌표계로 재투영하고,
- 그리드에 있는 셀 크기 및 오프셋을 동일하게 리샘플링하며,
- 관심 지역으로 잘라내고,
- 필요한 경우 값을 재조정할 수 있습니다.

이때 모든 래스터는 다른 파일로 저장될 것입니다.

먼저, *Raster* → *Align Raster*…에서 이 도구를 열고  Add new raster 아이콘을 클릭해서 QGIS 에 있는 기존 래스터 가운데 하나를 선택하십시오. 정렬 작업 뒤에 래스터를 저장할 산출 파일, 리샘플링 방법, 그리고 *Rescale values according to the cell size* 옵션으로 셀 크기에 따라 값을 재조정할지 여부를 선택하십시오. 리샘플링 방법은 다음 가운데 하나를 선택할 수 있습니다. (그림 15.18 을 참조하세요.):

- **Nearest Neighbor**
- **Bilinear (2x2 kernel)**
- **Cubic (4x4 kernel)**: 3 차 회선 근사값 (Cubic Convolution Approximation)
- **Cubic B-Spline (4x4 kernel)**: 3 차 B 스플라인 근사값 (Cubic B-Spline Approximation)
- **Lanczos (6x6 kernel)**: 란초시 창함수 싱크 보간법 (Lanczos windowed sinc interpolation)

- **Average**: NODATA 가 아닌 모든 기여 픽셀 (contributing pixel) 의 평균값을 계산
- **Mode**: 표본 추출 (sampling) 한 모든 포인트 가운데 가장 흔히 나타나는 값을 선택
- NODATA 가 아닌 모든 기여 픽셀의 **Maximum**, **Minimum**, **Mediane**, **First Quartile (Q1)** 또는 **Third Quartile (Q3)**

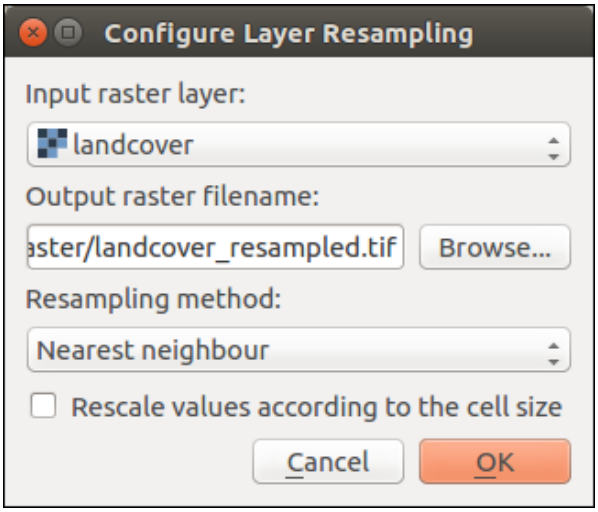


그림 15.18: 래스터 리샘플링 옵션 선택하기

Align raster 대화창에서, 래스터 레이어 목록으로부터 Edit file settings 아이콘으로 파일 설정을 편집하거나 Remove an existing file 아이콘으로 기존 파일을 제거할 수 있습니다. 다음 기타 옵션도 하나 이상 선택할 수 있습니다. (그림 15.19 를 참조하세요.):

- *Reference Layer*: 참조 레이어 선택
- *CRS*: 새 좌표계로 변형
- *Cell size*: 다른 셀 크기를 설정
- *Grid Offset*: 다른 그리드 오프셋을 설정
- *Clip to Extent*: 사용자 정의, 레이어 기반, 또는 맵 캔버스 기반 가운데 하나를 선택
- *Output Size*: 산출물 크기를 설정
- *Add aligned raster to the map canvas*: 맵 캔버스에 정렬한 래스터를 추가할지 여부를 선택

15.3 지리참조자

지리참조자는 래스터용 월드 파일을 생성하기 위한 도구입니다. 이 플러그인은 새로운 GeoTiff 를 생성하거나, 기존 이미지에 월드 파일을 추가해서 래스터에 지리 또는 투영 좌표계를 참조시킬 수 있습니다. 래스터를 지리참조하는 기본적인 접근법은 사용자가 좌표를 정확히 결정할 수 있는 래스터 상의 포인트의 위치를 찾는 것입니다.

피쳐

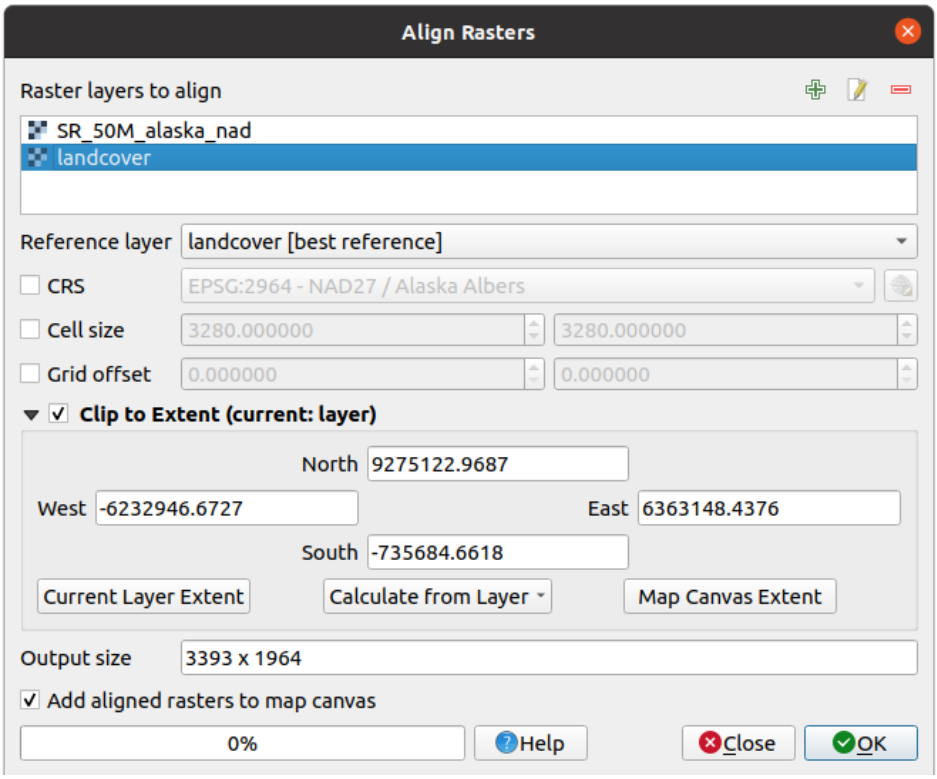


그림 15.19: 래스터 정렬

아이콘	목적	아이콘	목적
	래스터 열기		지리참조 작업 시작
	GDAL 스크립트 생성		GCP 점 로드
	다른 이름으로 GCP 점 저장		투영설정
	점추가		점삭제
	GCP 점이동		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		

표 - 지리참조자: 지리참조자 도구

15.3.1 일반 절차

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 *Raster* → *Georeferencer* 메뉴 옵션을 선택하는 것입니다. 그림 15.20 과 같은 지리참조자 플러그인 대화창이 열릴 것입니다.

이 예시를 위해 SDGS(South Dakota Geological Survey) 가 제공하는 사우스다코타 주의 지형 데이터 (topo sheet) 를 사용할 것입니다. 이 지형 데이터는 나중에 GRASS `spearfish60` 위치 데이터와 함께 가시화시킬 수 있습니다. https://grass.osgeo.org/sampledata/spearfish_toposheet.tar.gz 에서 해당 지형 데이터를 다운로드할 수 있습니다.

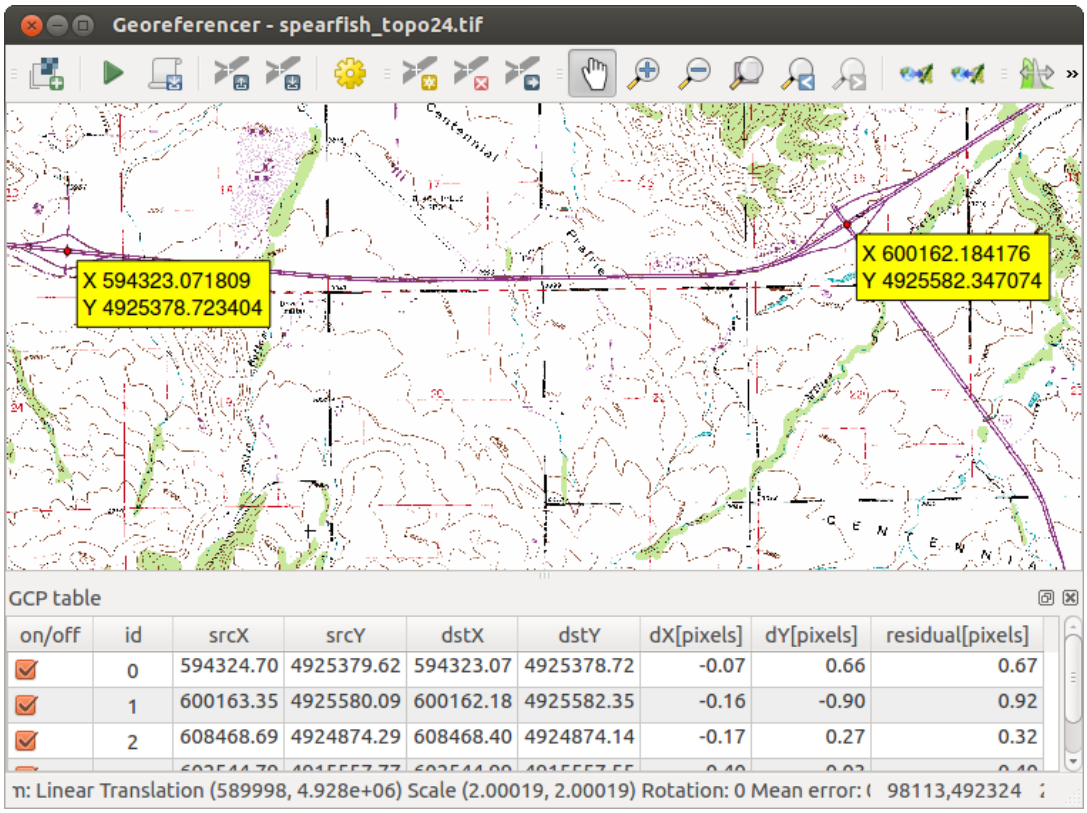


그림 15.20: 지리참조자 대화창

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.  Add Point 버튼을 사용해서 주 작업 영역에 포인트를 추가하고 그 좌표를 입력하십시오. (그림 15.21 을 참조하세요.) 이 절차에서 다음 세 가지 옵션을 선택할 수 있습니다:
   - 래스터 이미지에 포인트를 클릭한 다음 X 및 Y 좌표를 직접 입력합니다.
   - 래스터 이미지에 포인트를 클릭한 다음  From map canvas 버튼을 클릭해서 QGIS 맵 캔버스에 이미 불러온 참조 맵을 통해 X 및 Y 좌표를 추가합니다.
   - GCP 가 잘못된 곳에 있을 경우,  버튼으로 두 창 모두에서 GCP 를 이동시킬 수 있습니다.
3. 점입력을 계속하려면 최소 4 점이 필요하며 점이 많을수록 결과가 좋아집니다. GCP 점 세트를 작업위치에서 확대/축소 및 이동하응 추가도구가 있습니다.

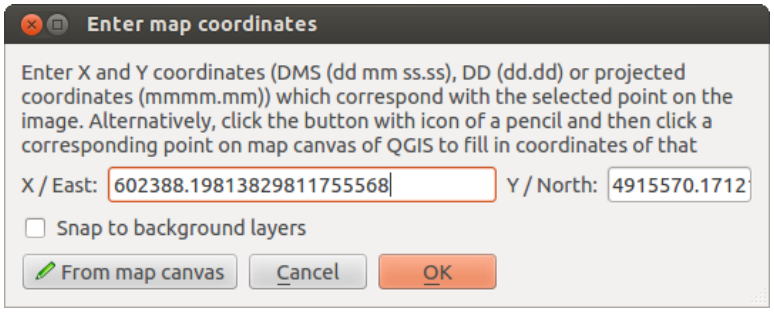


그림 15.21: 래스터 이미지에 포인트 추가

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 좌표계를 쓰고 있지만, 지리참조 정보만 누락돼 있는 경우 가장 적합한 알고리즘입니다. 적어도 2 개 이상의 GCP 가 필요합니다.

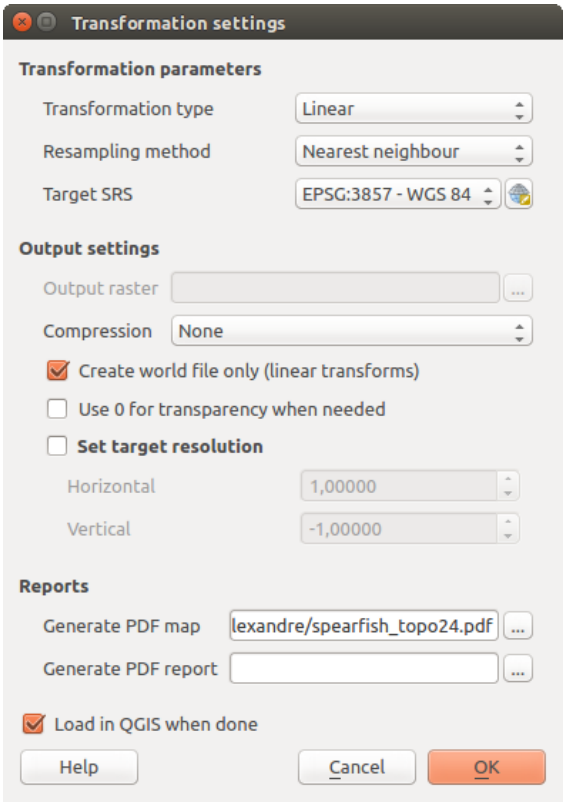


그림 15.22: 지리참조자 변형 설정 정의하기

- **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.
- **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.
- **Polynomial 2**, **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.
- **Projective** 알고리즘은 Polynomial 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, **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 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. 선형 (Linear)
3. 3 차 (Cubic)
4. 3 차 스플라인 (Cubic Spline)
5. 란초시 (Lánczos)

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 SRS* (공간 참조 시스템) 을 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 지형 데이터의 모든 하얀색 영역이 투명하게 변할 것입니다.
- 마지막으로, *Load in QGIS when done* 옵션을 활성화하면 변형 작업 종료시 QGIS 맵 캔버스에 산출 래스터를 자동적으로 불러올 것입니다.

래스터 속성 표시 및 조정

Settings -> *Raster properties* 메뉴 옵션을 선택하면 사용자가 지리참조시키고자 하는 래스터 파일의레이어 속성 대화창이 열립니다.

지리참조자 환경 설정

- GCP 좌표 그리고/또는 그 ID 를 표시할 것인지 여부를 정의할 수 있습니다.
- 잔차 (residual) 단위로 픽셀과 지도 단위를 선택할 수 있습니다.
- PDF 보고서는 좌우 여백과 PDF 지도의 용지 크기를 설정할 수 있습니다.
- 마지막으로  *Show Georeferencer window docked* 옵션을 활성화해서 지리참조자 대화창을 고정할 수 있습니다.

변형 작업 실행

모든 GCP 를 수집하고 모든 변형 설정을 정의했다면, 그냥  Start georeferencing 버튼을 눌러 새 지리참조 래스터를 생성하십시오.

CHAPTER 16

메시 데이터 작업

16.1 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, 경계 (edge) 및 면 (face) 의 집합을 담고 있습니다.

- **꼭짓점**: XY(Z) 포인트 (레이어 좌표계)
- **경계**: 연결된 꼭짓점 쌍
- **면**: 면이란 닫힌 도형 (closed shape) 을 형성하는 경계의 집합을 말합니다. 일반적으로 삼각형 또는 사변형 (quadrilateral) 이며, 드물게 더 많은 꼭짓점을 가진 폴리곤도 면이 될 수 있습니다.

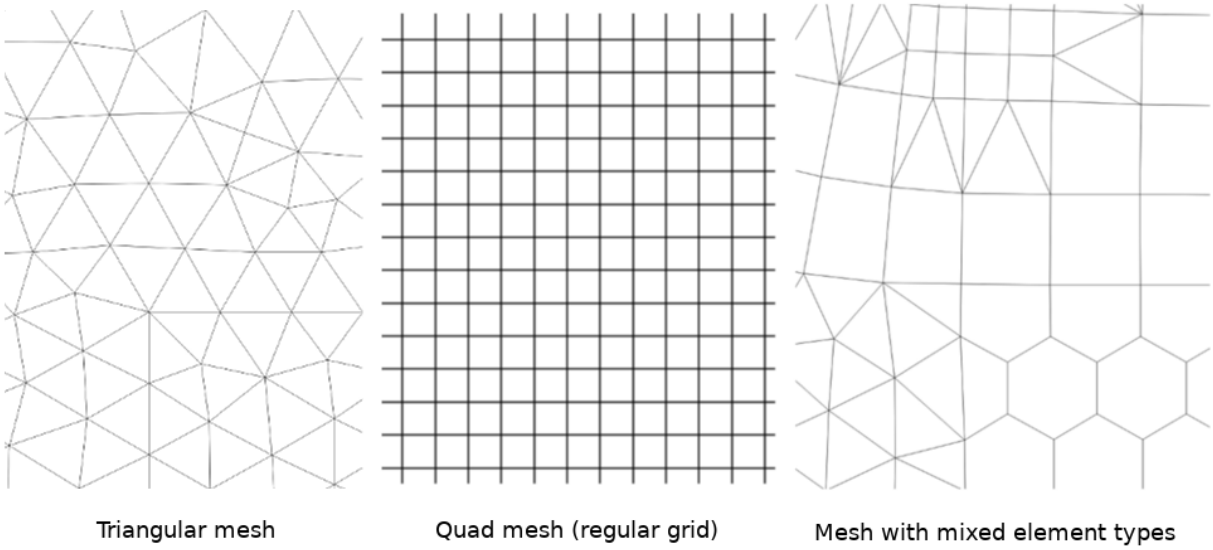


그림 16.1: 서로 다른 메시 유형

현재 QGIS 는 삼각형 또는 정사각형을 사용하는 메시 데이터를 렌더링할 수 있습니다.

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:

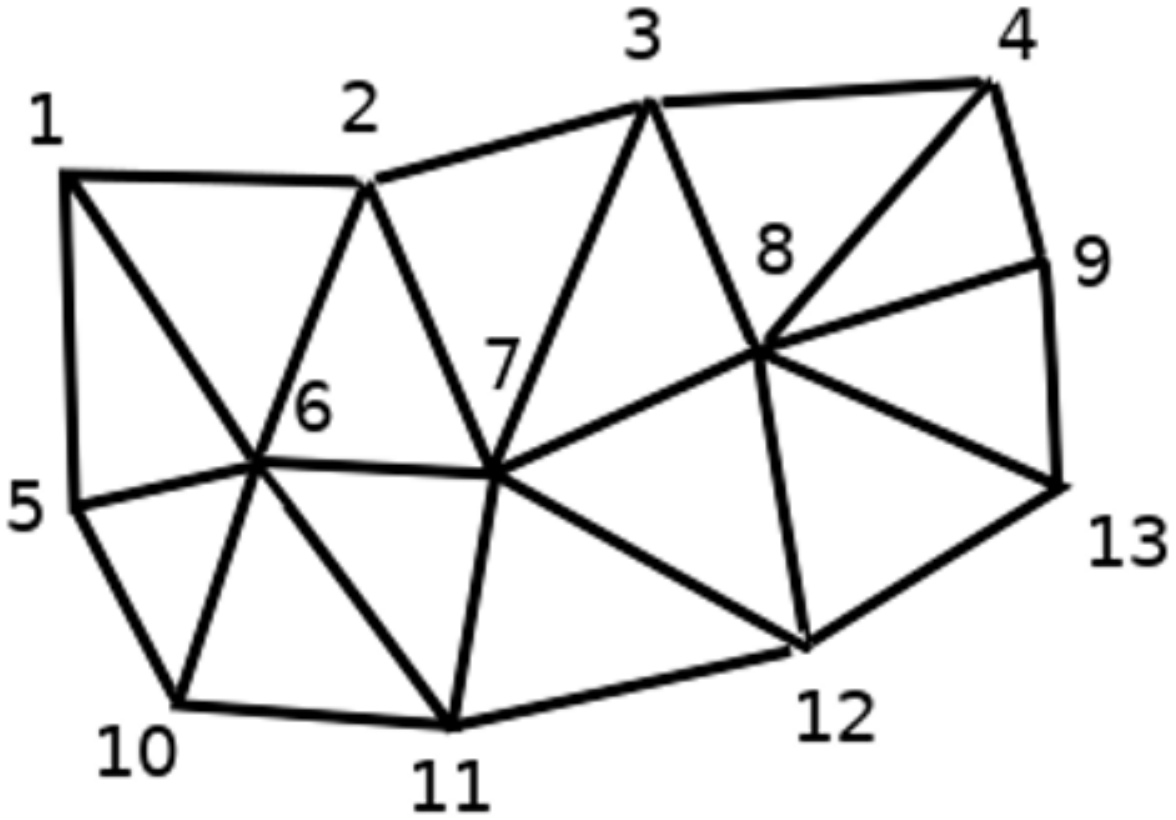


그림 16.2: 번호가 붙은 꼭짓점을 가진 삼각형 그리드

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.

비슷한 방식으로, 메시 데이터셋이 각 꼭짓점에 벡터값도 저장할 수 있습니다. 예를 들어 지정한 타임 스탬프 순간의 풍향처럼 말이죠:

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.

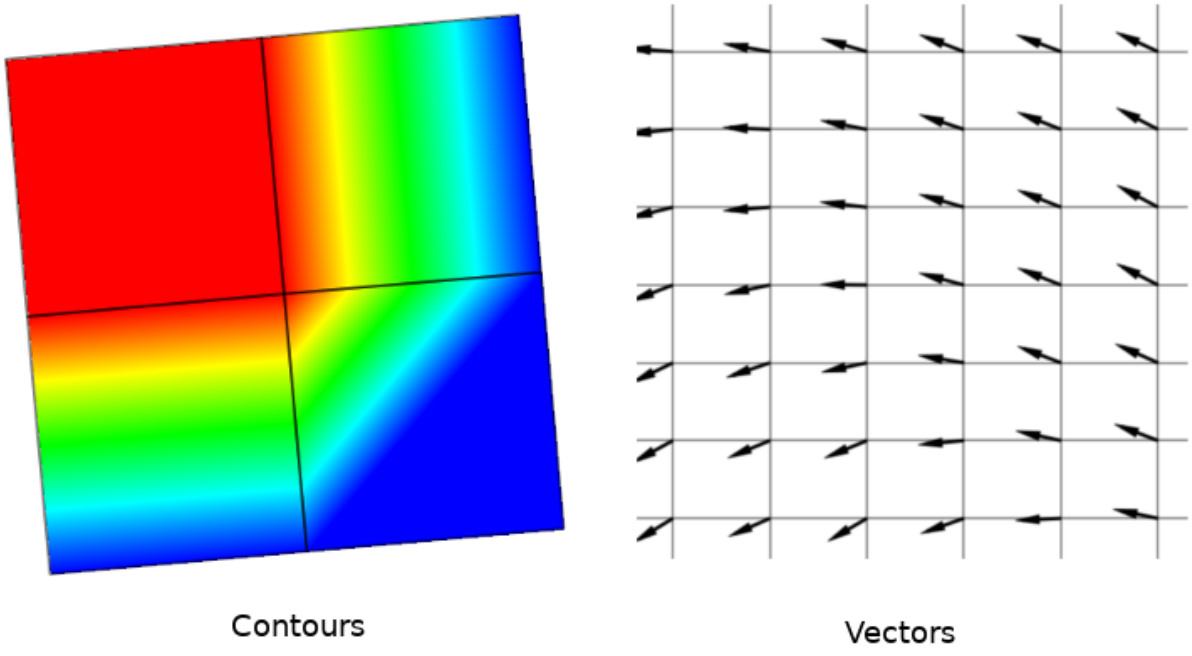


그림 16.3: 메시 데이터를 가시화할 수 있는 방법들

16.2 지원 포맷

QGIS 는 MDAL 드라이버 를 사용해서 메시 데이터에 접근합니다. 따라서 태생적으로 다음과 같은 포맷을 지원합니다:

- `NetCDF`: 과학 데이터용 일반 포맷
- `GRIB`: 기상학에서 일반적으로 사용되는 포맷
- `XMDF`: 예를 들자면 TUFLOW 모델링 패키지에서 나온 수력학 (hydraulic) 산출물의 포맷
- `DAT`: 다양한 유체역학 모델링 패키지의 산출물 포맷 (예: BASEMENT, HYDRO_AS-2D, TUFLOW)
- `3Di`: 기상 `3Di`: 기상예보협약 (http://cfconventions.org/) 에 기반한 3Di 모델링 패키지 포맷
- 다음 웹페이지에서 메시 데이터셋의 몇몇 예시를 찾아볼 수 있습니다: https://apps.ecmwf.int/datasets/data/interim-full-daily/levtype=sfc/

Data Source Manager 대화창에 있는  *Mesh* 탭을 통해 QGIS 로 메시 데이터셋을 불러올 수 있습니다. 더 자세한 내용은메시 레이어 불러오기 를 읽어보세요.

16.3 메시 데이터셋 속성

16.3.1 정보 속성

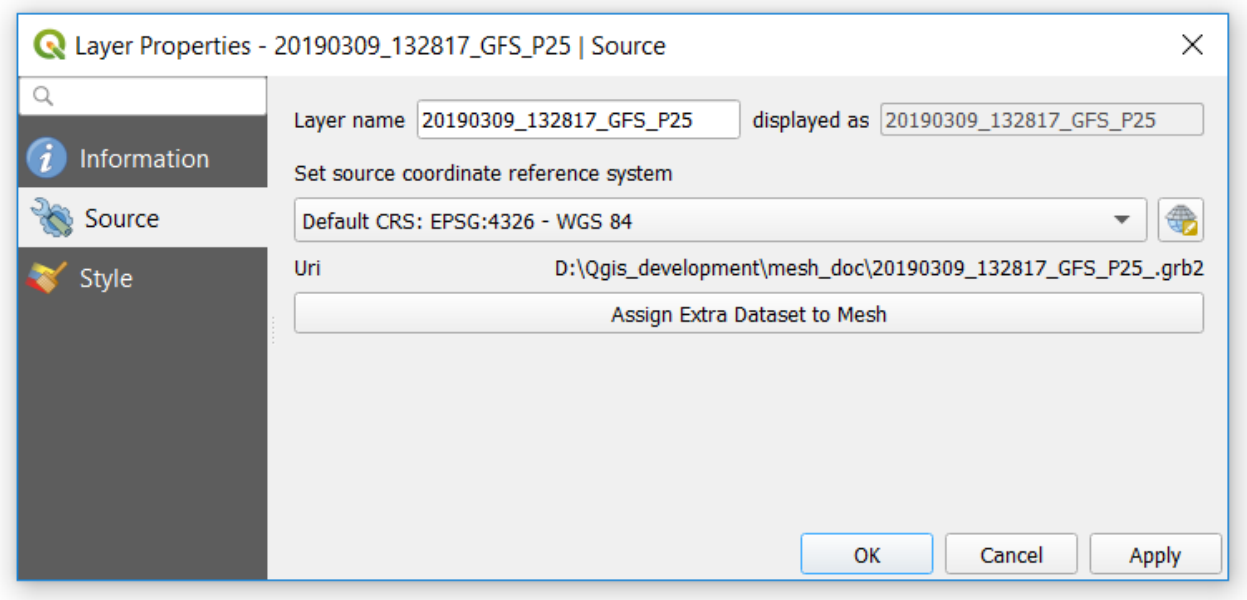


그림 16.4: 메시 레이어 속성

Information 탭은 읽기 전용으로, 현재 레이어에 대한 요약 정보 및 메타데이터를 한 눈에 볼 수 있는 흥미로운 장소입니다. URI, 꼭짓점 개수, 면 개수 및 데이터셋 그룹 개수 정보를 (레이어 제공자를 기반으로) 제공합니다.

16.3.2 소스 속성

Source 탭은 선택한 메시에 대한 다음과 같은 기본 정보를 표시합니다:

- *Layers* 패널에 표시될 레이어명
- 좌표계 설정: 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.

Assign Extra Dataset to Mesh 버튼을 사용해서 현재 메시 레이어에 더 많은 그룹을 추가할 수 있습니다.

16.3.3 심볼 속성

Symbology 탭을 선택하면 다음 그림과 같은 대화창을 활성화합니다:

심볼 속성은 다음 여러 탭으로 나누어집니다:

- 일반 *(General)*
- 등고선 심볼 *(Contours Symbology)*
- 벡터 심볼 *(Vectors Symbology)*

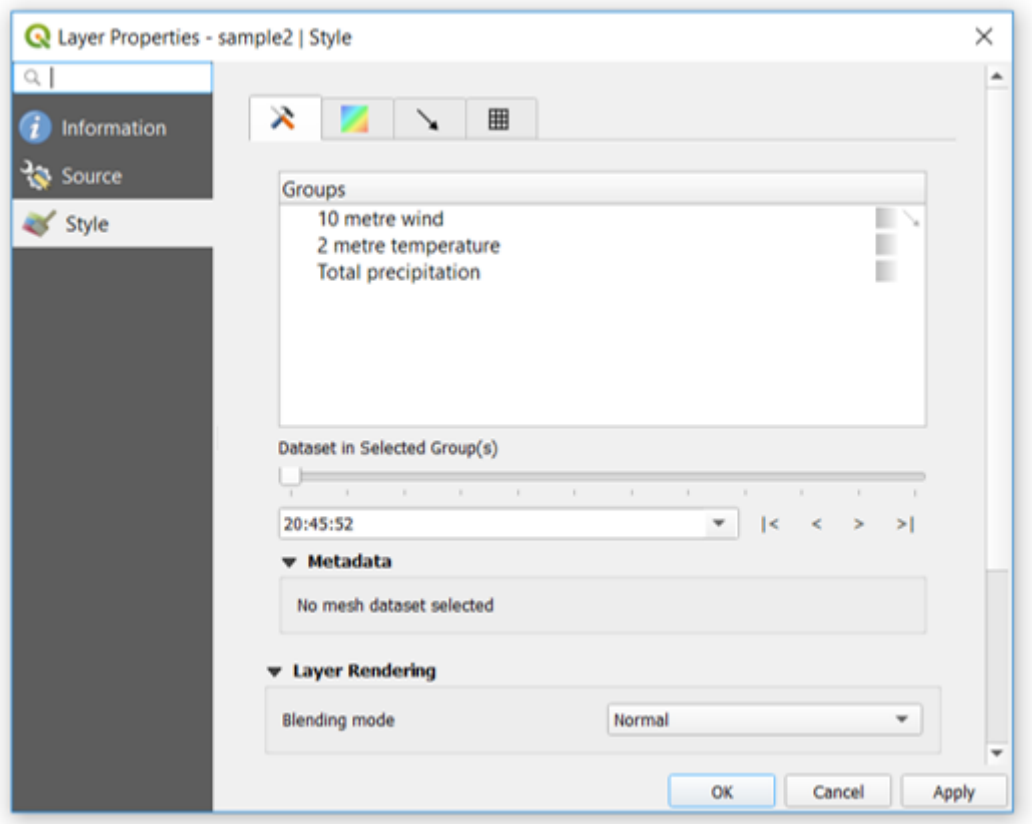


그림 16.5: 메시 레이어 심볼

- 렌더링 *(Rendering)*

일반


일반 탭은 다음 항목들을 표시합니다:

- 메시 데이터셋에서 사용할 수 있는 그룹들
- 예를 들어 레이어가 시계열 차원을 가지고 있는 경우, 선택한 그룹 (들) 의 데이터셋
- 사용할 수 있는 경우, 메타데이터
- 선택한 데이터셋에 사용할 수 있는혼합 모드

슬라이드 바  , 콤보박스  그리고 |<, <, >, >| 버튼으로 데이터의 또다른 차원을 — 또다른 차원이 있다면 — 탐색할 수 있습니다. 슬라이드 바를 움직이면 그에 따라 메타데이터를 나타냅니다. 다음메시 그룹 그림을 예시로 참조하세요. 맵 캔버스도 물론 선택한 데이터셋 그룹을 표시할 것입니다.

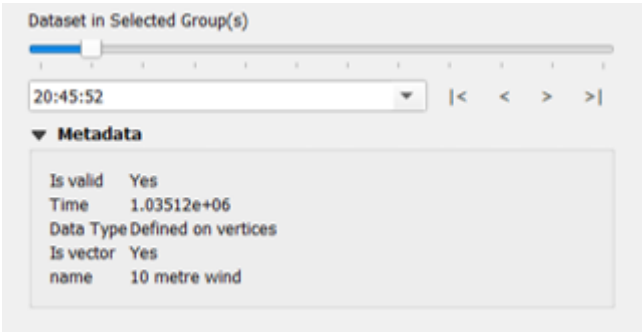


그림 16.6: 선택한 그룹 (들) 의 데이터셋

다음 탭들을 통해 각 그룹에 심볼을 적용할 수 있습니다.

등고선 심볼

일반 탭의 *Groups* 프레임에서  아이콘을 클릭하면 기본 가시화 파라미터로 등고선을 표시합니다.


등고선 심볼 탭에서 선택한 그룹을 위한 등고선의 현재 가시화 옵션을 살펴보고 변경할 수 있습니다. 다음 그림 16.7 그림을 참조하세요:

슬라이드 바 또는 콤보박스를 사용해서 현재 그룹의 투명도를 설정합니다.

Load 버튼을 클릭해서 현재 그룹의 최소/최대 값을 조정합니다.

Interpolation 드롭다운 목록은 등고선을 렌더링하기 위한 *Linear*, *Discrete* 그리고 *Exact* 옵션 3 개를 담고 있습니다.

Color ramp 위젯은색상표 드롭다운 단축키 를 엽니다.

Label unit suffix 는 범례에 있는 값 뒤에 추가되는 라벨입니다.

범주화 *Mode* 에서'Continuous'를 선택하면, QGIS 가 *Min* 과 *Max* 값을 고려해서 범주 단계를 자동으로 생성합니다. 'Equal interval'을 선택하면, *Classes* 콤보박스에서 범주 단계의 개수를 선택한 다음 *Classify* 버튼을 누르기만 하면 됩니다.


Add values manually 버튼은 개별 색상표에 값을 추가합니다.
Remove selected row 아이콘은 개별 색상표에서 값을 삭제합니다. *Value* 열을 더블클릭하면 특정 값을 삽입할 수 있고, *Color* 열을 더블클릭하면 해당 값에 적용할 색상을 선택할 수 있는 *Change color* 대화창이 열립니다.

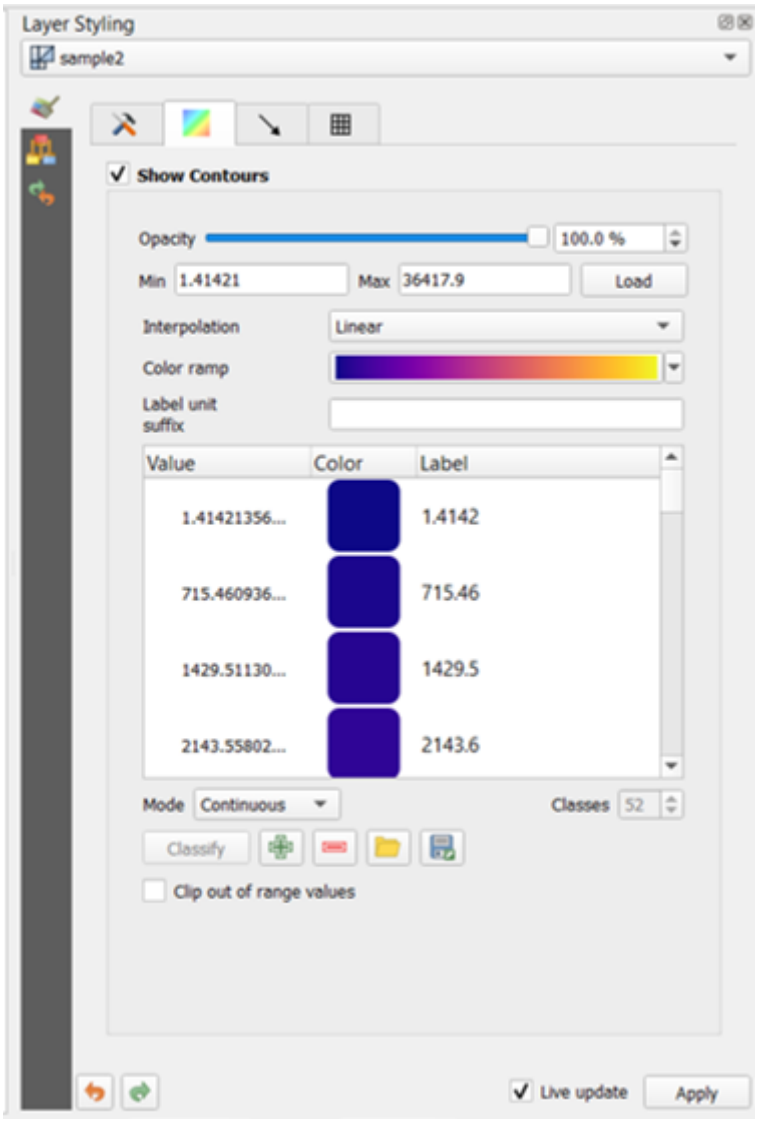


그림 16.7: 메시 레이어에서 등고선 스타일 작업하기

벡터 심볼

 일반 탭에서, 사용할 수 있는 경우  아이콘을 클릭해서 벡터를 표시하십시오. 맵 캔버스가 선택한 그룹의 벡터를 기본 파라미터로 표시할 것입니다.  벡터 심볼 탭을 선택하면 벡터를 위한 가시화 파라미터를 다음 그림과 같이 변경할 수 있습니다:

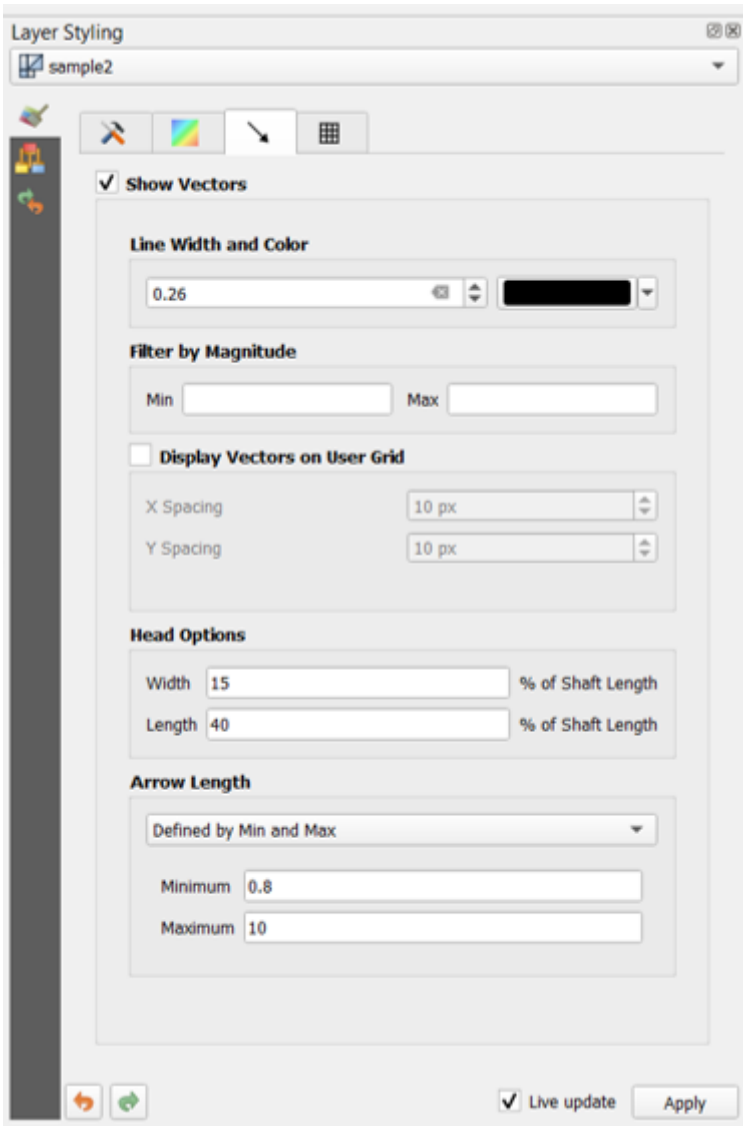


그림 16.8: 메시 레이어에서 벡터 스타일 작업하기

콤보박스를 통해 또는 값을 입력해서 라인 너비를 설정할 수 있습니다. 색상 위젯은 벡터에 적용할 색상을 선택할 수 있는 *Change color* 대화창을 엽니다.

벡터 값의 크기에 따라 벡터를 필터링하기 위한 *Min* 과 *Max* 값을 입력하십시오.

 *Display Vectors on User Grid* 옵션을 체크한 다음 *X spacing* 및 *Y spacing* 을 지정하면, QGIS 가 지정한 간격을 고려해서 벡터를 렌더링할 것입니다.

Head Options 화살촉 옵션을 통해 그 너비와 길이를 (백분율로) 지정해서 화살촉의 모양을 설정할 수 있습니다.

Arrow length: QGIS 는 벡터의 화살표 길이를 세 가지 서로 다른 방식으로 렌더링할 수 있습니다:

- **최소/최대 값으로 정의 (Defined by Min and Max)**: 사용자가 벡터의 최소/최대 길이를 지정하면, QGIS 는 그에 따라 벡터 가시화를 조정할 것입니다.
- **값의 크기에 따라 (Scale to magnitude)**: 사용할 (배율) 인자를 지정합니다.
- 고정 (Fixed): 모든 벡터를 동일한 길이로 표시합니다.

렌더링

QGIS 는  렌더링 탭에서 그리드를 표시할 수 있는 두 가지 가능성을 제시합니다. 다음 그림 16.9 그림을 참조하세요:

- *Native Mesh Rendering*: 사변형을 표시합니다.
- *Triangular Mesh Rendering*: 삼각형을 표시합니다.

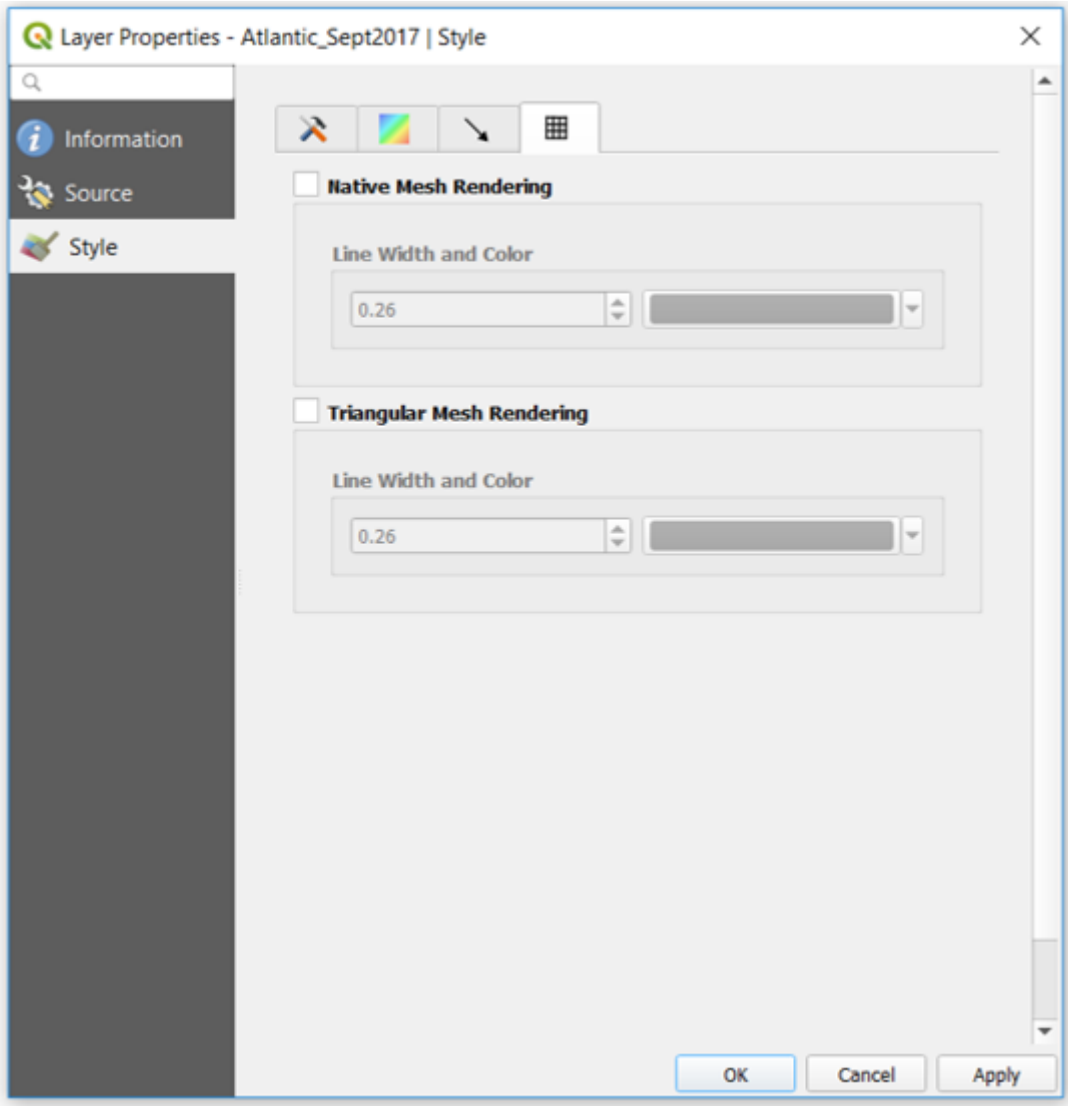


그림 16.9: 메시 렌더링

이 대화창에서 라인 너비 및 색상을 변경할 수 있으며, 두 그리드 렌더링을 모두 끌 수 있습니다.

CHAPTER 17

벡터 타일 작업

17.1 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 QGIS가 지도 제작 스타일을 적용해야 합니다.

17.2 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`

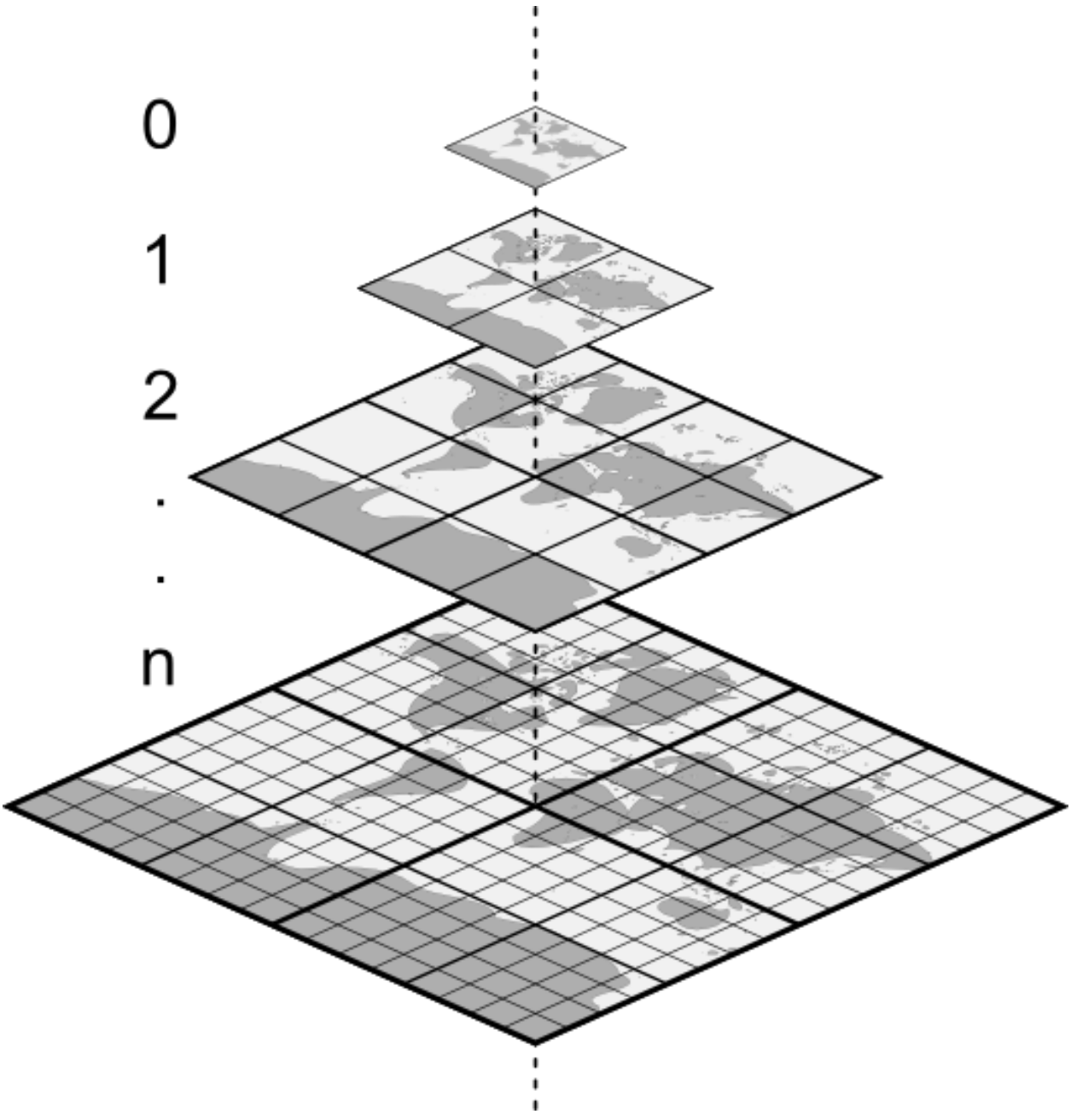


그림 17.1: 확대/축소 수준을 가진 벡터 타일의 피라미드 구조

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

인쇄 조판기가 제공하는 조판 및 인쇄 기능은 점차 증가하고 있습니다. QGIS 맵 캔버스, 텍스트 라벨, 이미지, 범례, 축척 막대, 기본 도형, 화살표, 속성 테이블 및 HTML 프레임과 같은 항목들을 추가할 수 있습니다. 각 항목의 크기를 조정하고 그룹화하고 정렬하고 배치하고 기울이며, 그 속성을 수정해서 사용자의 조판을 생성할 수 있습니다. 이 조판을 인쇄하거나 또는 이미지 포맷, PostScript, PDF 또는 SVG 파일로 내보낼 수 있습니다. 사용자가 만든 조판을 템플릿으로 저장해서 다른 세션에서 다시 불러올 수도 있습니다. 마지막으로, 지도책 생성기를 통해 템플릿 하나에 기반한 여러 맵을 생성할 수 있습니다.

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, 툴바에 있는  New Print Layout 아이콘을 클릭하거나 *File → 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. 명칭을 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.

조판 관리자 하단에서 새 인쇄 조판을 생성하고, 기초부터 보고서를 생성하거나 템플릿을 생성할 수 있습니다. 기본적으로, QGIS 는 사용자 프로파일과 응용 프로그램 템플릿 디렉터리 (프레임 하단에 있는 두 버튼으로 접근할 수 있습니다) 는 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.

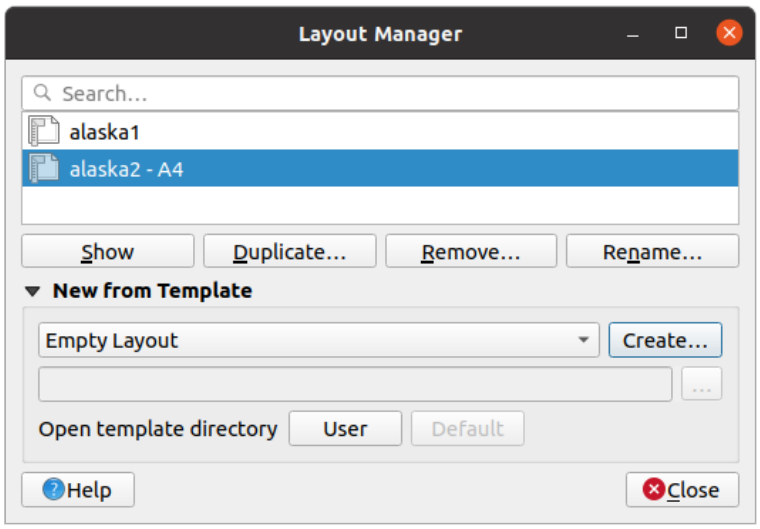


그림 18.1: 인쇄 조판 관리자

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 현재 QGIS 맵 캔버스, 텍스트 라벨, 이미지, 범례, 축척 막대, 기본 도형, 화살표, 속성 테이블 및 HTML 프레임과 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 내용물을 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.
- *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.)

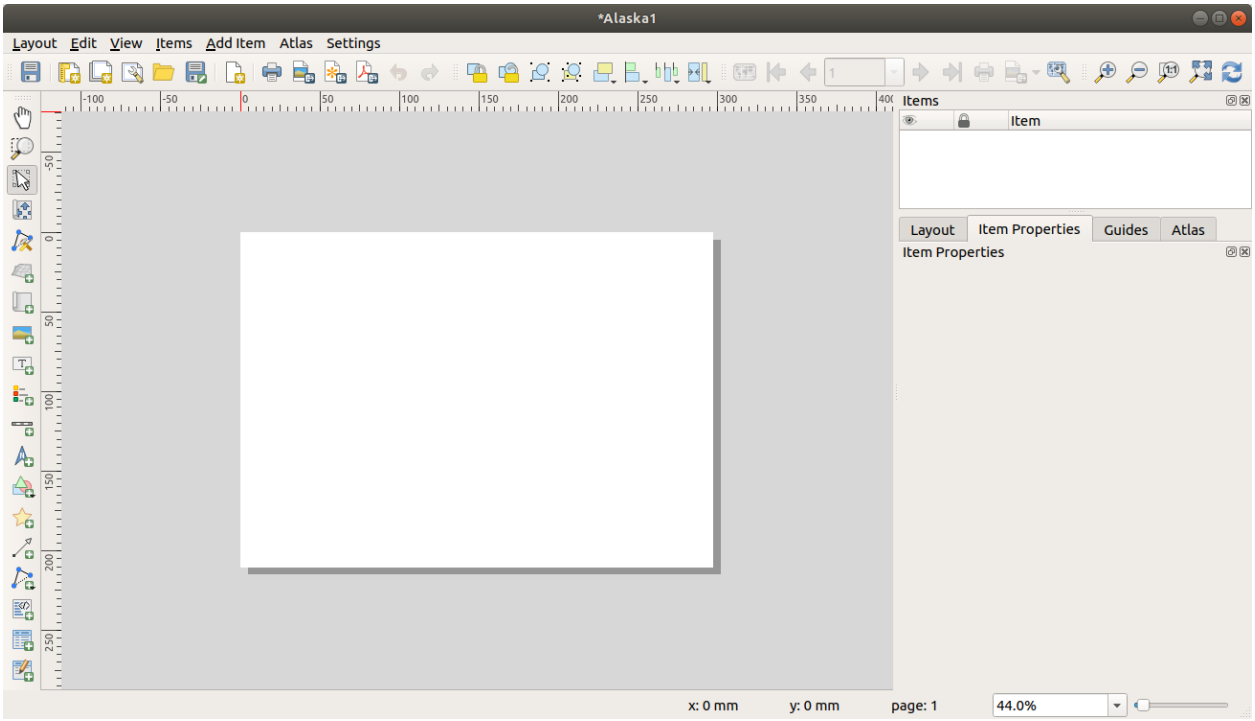


그림 18.2: 인쇄 조판기

- *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 또는 *View* → *Toolbars* → or *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.

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…			
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
Page Setup…	`Ctrl+Shift+P`		
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 18.1: 사용할 수 있는 도구들

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.

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
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.)

도구	단축키	툴바	참조
Group	`Ctrl+G`	*Actions*	항목 그룹화
Ungroup	`Ctrl+Shift+G`	*Actions*	항목 그룹화
Raise	`Ctrl+]`	*Actions*	정렬
Lower	`Ctrl+[`	*Actions*	정렬
Bring to Front	`Ctrl+Shift+]`	*Actions*	정렬
Send to Back	`Ctrl+Shift+[`	*Actions*	정렬
Lock Selected Items	`Ctrl+L`	*Actions*	항목 고정하기
Unlock All	`Ctrl+Shift+L`	*Actions*	항목 고정하기
Align Items		*Actions*	정렬
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.

도구	툴바	참조
Add Map	*Toolbox*	맵 항목
Add Picture	*Toolbox*	그림 항목
Add Label	*Toolbox*	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*	
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 3D Map	*Toolbox*	*3D* 맵 항목

지도책 메뉴

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
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.

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.

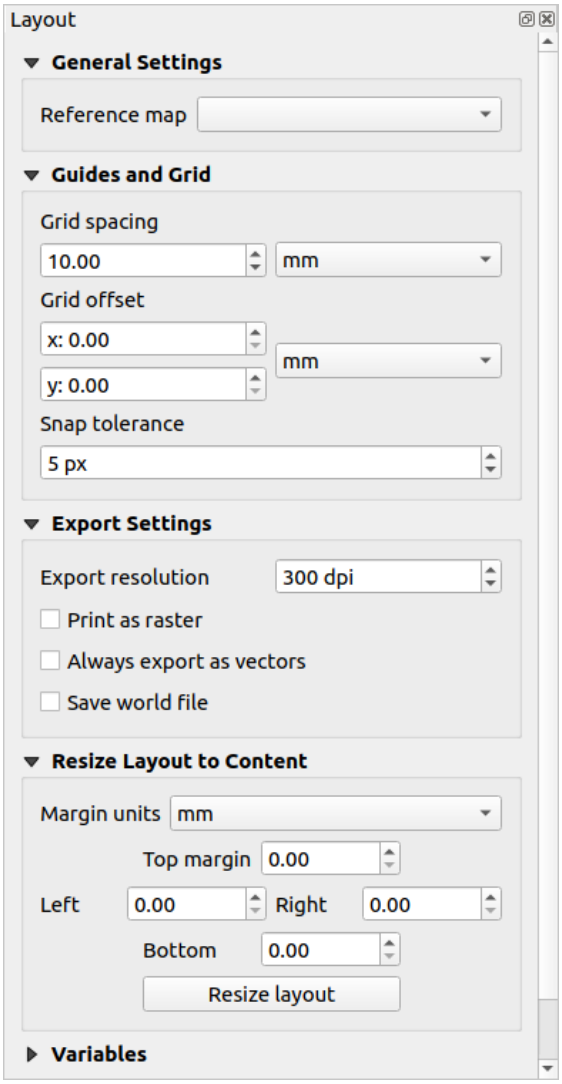


그림 18.3: 인쇄 조판기에서 조판 설정하기

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 파라미터들를 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 명칭을 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.

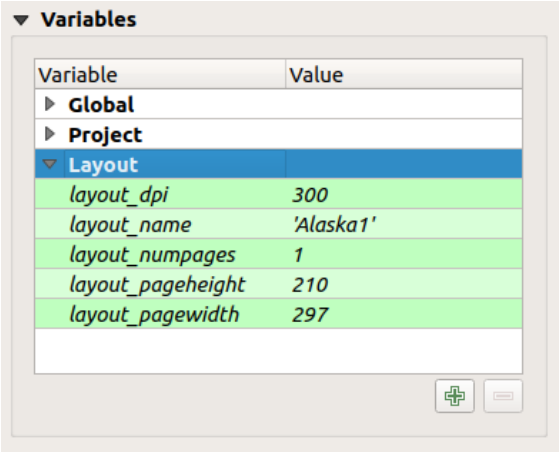


그림 18.4: 인쇄 조판기의 변수 편집기

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.

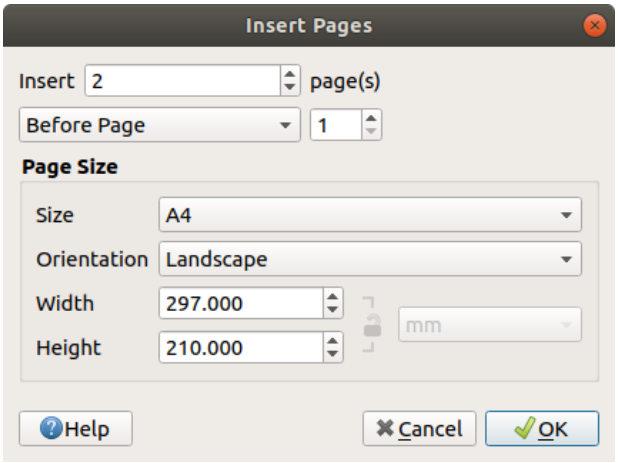


그림 18.5: 인쇄 조판기에서 새 페이지 생성하기

페이지 속성 업데이트하기

페이지를 생성한 후 페이지의 *Item Properties* 패널을 통해 어떤 페이지든 사용자 지정할 수 있습니다. 페이지를 오른쪽 클릭하고 *Page Properties…*를 선택하십시오. 다음과 같은 설정을 할 수 있는 *Item Properties* 패널이 열립니다:

- 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.)
- ☐ *Exclude page from exports* 옵션은조판기 산출물 에 현재 페이지를 그 내용과 함께 포함시켜야 할지 여부를 제어할 수 있습니다.
- 현재 페이지의 *Background* 배경을 사용자가 원하는색상 또는심볼 로 설정할 수 있습니다.

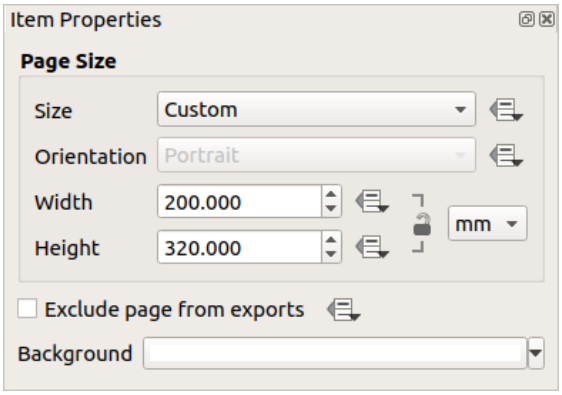


그림 18.6: 페이지 속성 대화창

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.

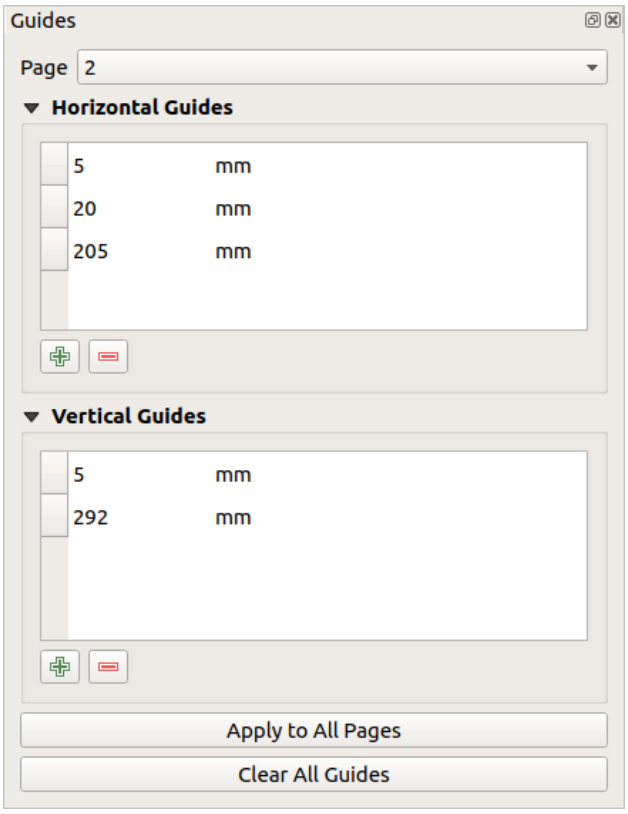

Guides
Page 2
Horizontal Guides
5 mm
20 mm
205 mm
Vertical Guides
5 mm
292 mm
Apply to All Pages
Clear All Guides

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.

선택한 모든 항목에 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 클릭해서 항목을 복사 또는 삭제하거나, 항목의속성 패널 을 열 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 실행하기만 하면, 선택한 액션 이후에 수행된 모든 액션이 제거될 것입니다.

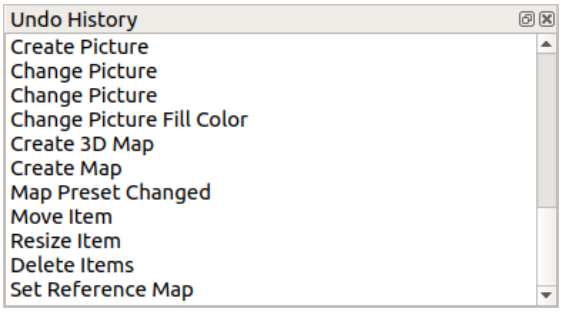


그림 18.8: 인쇄 조판기의 실행 취소 이력

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 둘 중 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.)

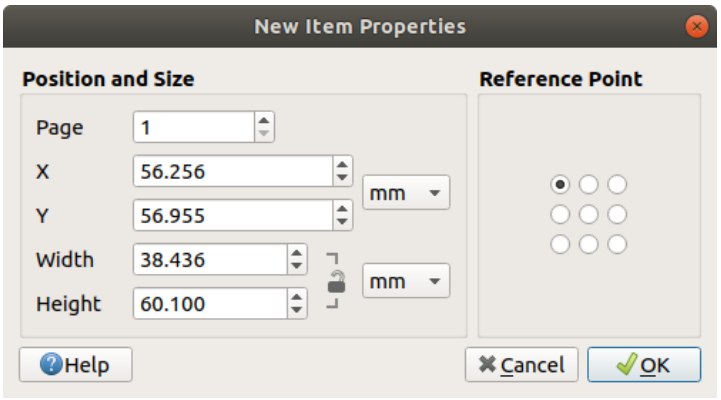


그림 18.9: 새 항목 속성 대화창

   - 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 만드는 자동화 도구들을 사용할 수 있습니다:

- 항목의 경계 (좌측, 우측, 상단 또는 하단)
- 항목의 수평 또는 수직 중심

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 *View* 메뉴 또는 *Actions* 툴바 또는 컨텍스트 메뉴에 있는 *Group* 버튼을 누르십시오. *Items* 패널에 `Group` 이라는 행이 추가되어, 다른 항목 패널 객체 *<layout_items_panel>* 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|unlock| 아이콘을 사용하면 고정된 모든 항목을 고정 해제할 수 있습니다.

정렬

Raise selected items 풀다운 메뉴에서 항목의 가시성 순위를 올리거나 내릴 수 있습니다. 인쇄 조판기 캔버스 상에서 항목을 선택한 다음 선택한 항목을 다른 항목들에 상대적으로 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.

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.)

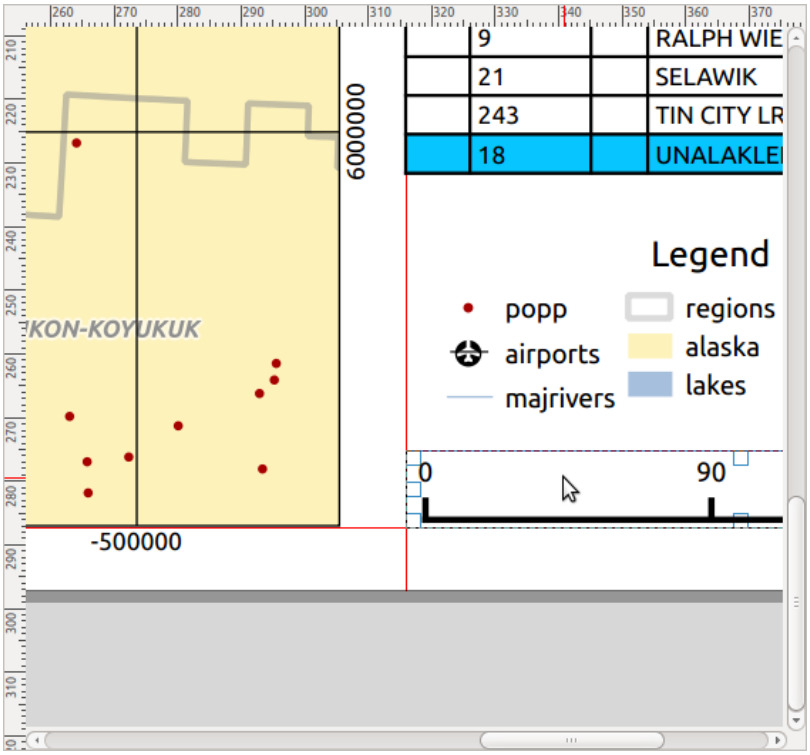


그림 18.10: 인쇄 조판기의 정렬 도우미 선

- *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 color* 메뉴는 배경 색상을 설정합니다. *Color*…버튼을 클릭하면 색상을 고를 수 있는 대화창이 열리거나, 또는 사용자 지정 설정에서 색상을 선택할 수 있습니다. 알파 필드 설정을 변경해서 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.
- *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.

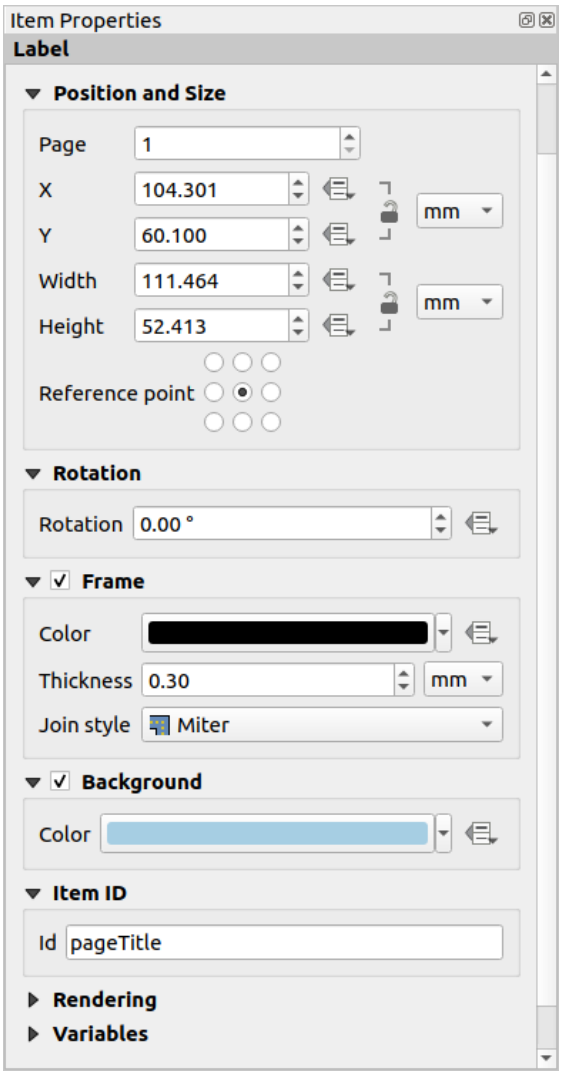


그림 18.11: 공통 항목 속성 그룹

위치 및 크기

새 항목 속성 대화창의 기능을 데이터 정의 기능을 사용해서 확장시키는 이 그룹은 항목을 정확하게 배치할 수 있게 해줍니다.

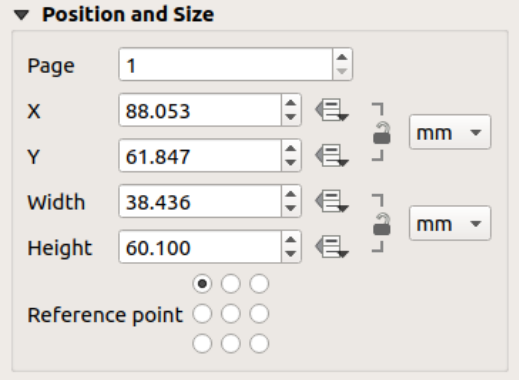


그림 18.12: 위치 및 크기

- 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.

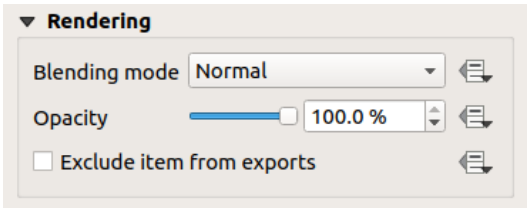


그림 18.13: 렌더링 모드

- *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.)
- *Transparency* : 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*: 모든 내보내기에서 특정 항목을 보이지 않도록 만들 수 있습니다. 이 체크박스를 활성화하면, 선택한 항목이 인쇄물, PDF 파일 등등에 포함되지 않을 것입니다.

변수

Variables 는 조판기 항목 수준에서 사용할 수 있는 모든 변수의 목록을 표시합니다. (전체, 프로젝트 및 조판기 수준의 모든 변수를 포함하고 있습니다.) 맵 항목은 지도의 축척, 범위 등과 같은 값에 쉽게 접근할 수 있는 맵 설정 변수도 포함하고 있습니다.

Variables 에서, 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:

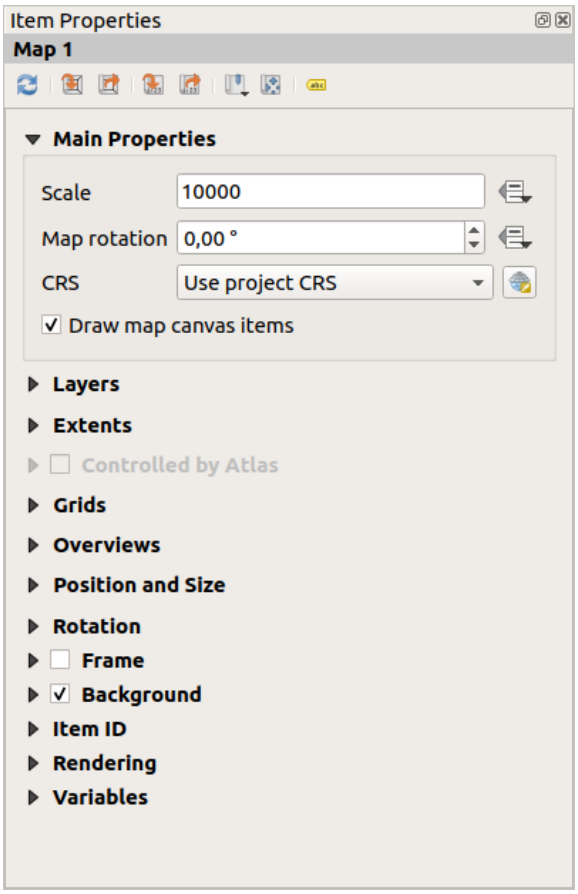


그림 18.14: 맵 항목 속성 패널

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 축척과 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.
- 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.
- *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 18.15 참조) 동기화를 끊을 수 있습니다.

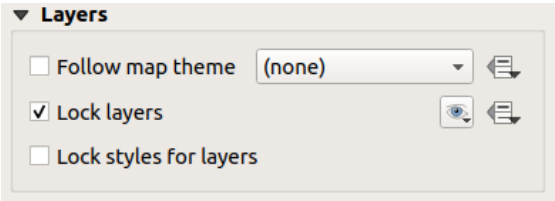


그림 18.15: 맵 레이어 그룹

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 18.16 을 참조하세요):

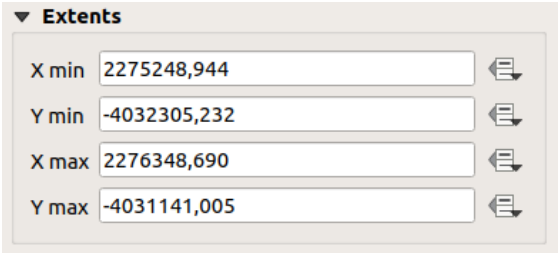


그림 18.16: 맵 범위 그룹

범위 (Extents) 영역은 맵 항목이 표시하는 영역의 `X` 및 `Y` 좌표를 표시합니다. 맵 항목이 표시하는 맵 캔버스 영역을 그리고/또는 맵 항목의 크기를 수정해서 두 값을 각각 직접 수정할 수 있습니다. *Set to Map Canvas Extent* 버튼을 클릭하면 조판기 맵 항목의 맵 범위를 메인 맵 캔버스의 범위로 설정합니다. *View Extent in Map Canvas* 버튼은 완전히 반대의 역할로, 메인 맵 캔버스의 범위를 조판기 맵 항목의 범위로 업데이트합니다.

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.

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.

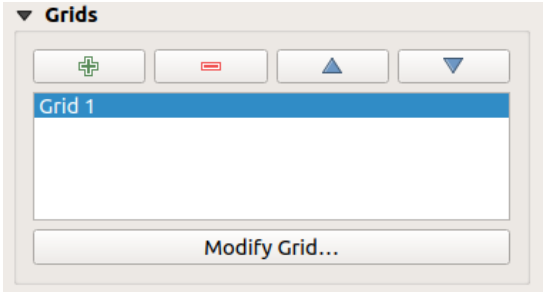


그림 18.17: 맵 그리드 그룹

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. 색상 과심볼 선택기 위젯을 사용해서 *Line style* 을 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 심볼을 표시합니다.
- 또는 프레임 및 주석만 표시할 수도 있습니다.

그리드 유형 외에도, 다음을 정의할 수 있습니다:

- *CRS*: 그리드의 좌표계를 정의합니다. 변경하지 않는 경우 맵 좌표계를 따를 것입니다. *Change* 버튼을 누르면 다른 좌표계로 설정할 수 있습니다. 설정한 후에도, 좌표계 선택 대화창의 *Predefined Coordinate Reference Systems* 부분에 있는 모든 그룹 이름 (예: **지리 좌표계**) 가운데 하나를 선택하면 기본값으로 변경시킬 수 있습니다.
- *Interval*: 그리드 참조용으로 사용할 간격 유형을 정의합니다. `Map Unit`, `Fit Segment Width`, `Millimeter` 또는 `Centimeter` 옵션을 선택할 수 있습니다.
  - `Fit Segment Width` 를 선택하면 맵 범위를 기반으로"보기 좋은 (pretty)"그리드 간격을 동적으로 선택할 것입니다. 이 옵션을 선택했다면 `Minimum` 및 `Maximum` 간격을 설정할 수 있습니다.
  - 다른 옵션들을 선택하면 `X` 및 `Y` 방향을 참조하는 두 연속되는 그리드들 사이의 거리를 설정할 수 있습니다.
- *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.

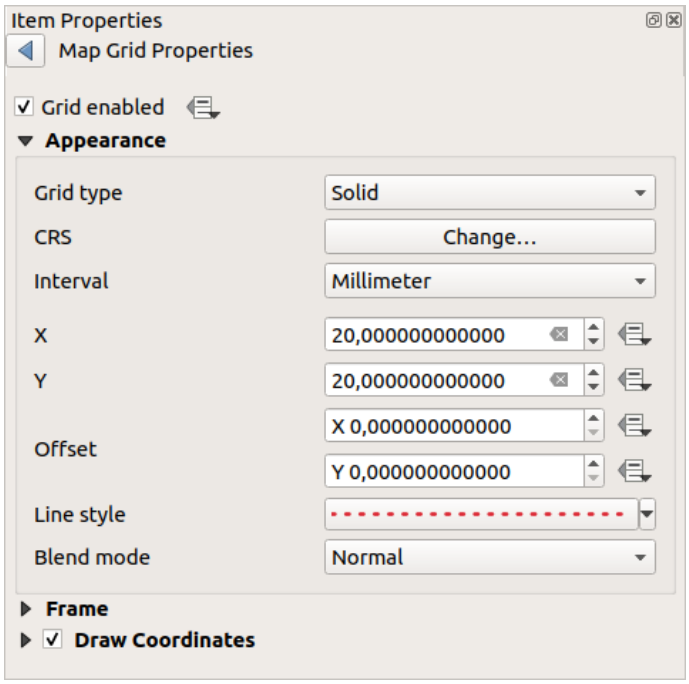


그림 18.18: 그리드 모양 대화창

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, `divisions` 옵션들을 `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.

좌표

Draw coordinates 옵션을 활성화하면 맵 프레임에 좌표를 추가할 수 있습니다. 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.

어느 주석을 표시할 것인지 선택할 수 있는 옵션도 있는데, `Show all`, `Latitude only`, `Longitude only` 또는 `Disable (none)` 가운데 하나를 선택할 수 있습니다. 맵이 기울어진 경우 이 옵션이 유용합니다. 맵 프레임 바깥쪽 또는 안쪽에 주석을 렌더링할 수 있습니다. 주석의 방향은 `Horizontal`, `Vertical ascending` 또는 `Vertical descending` 가운데 하나를 선택할 수 있습니다.

마지막으로, 주석 글꼴, 글꼴 색상, 맵 프레임에서 주석까지의 거리 그리고 렌더링한 좌표의 정밀도를 정의할 수 있습니다.

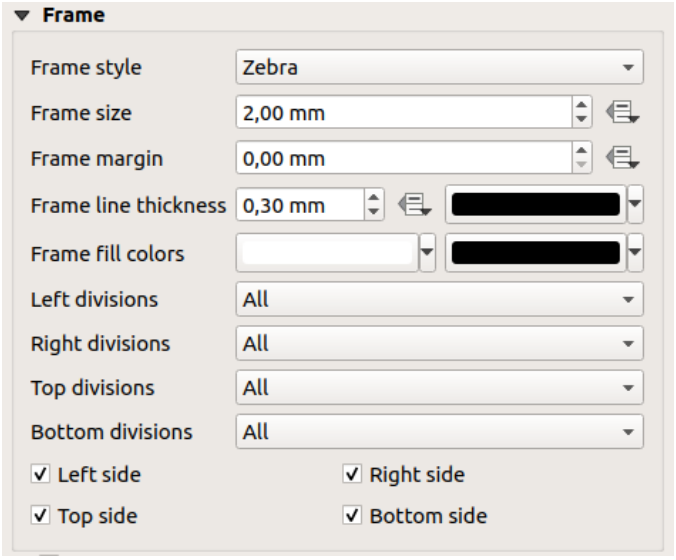


그림 18.19: 그리드 프레임 대화창

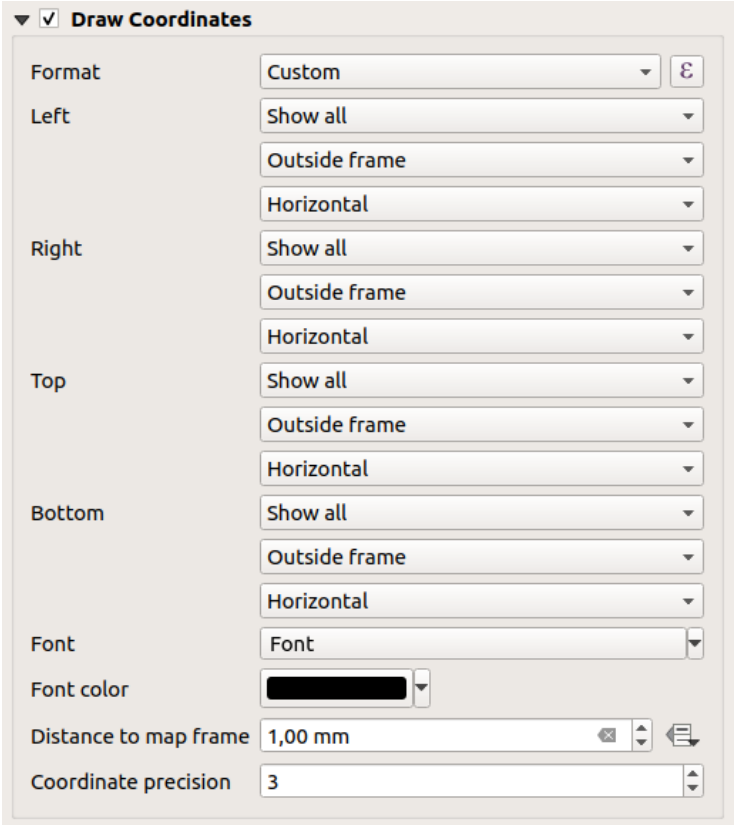


그림 18.20: 그리드 좌표 렌더링 대화창

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:

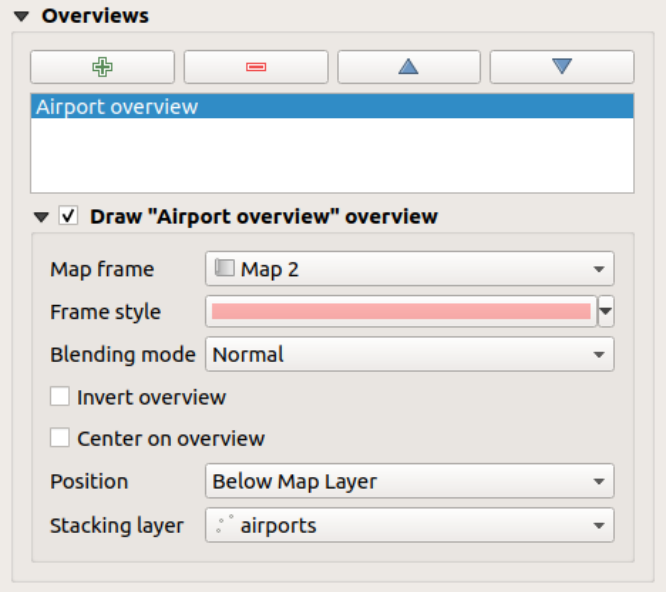


그림 18.21: 맵 오버뷰 그룹

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 18.21 을 참조하세요.) 이 그룹에서:

- 더블클릭하면 다른 명칭으로 변경할 수 있습니다.
- 및 버튼을 이용해서 오버뷰를 추가하거나 제거할 수 있습니다.
- 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 3D 맵 항목

3D 맵 항목을 사용해서*3D 맵 뷰* 를 표시합니다.  *Add 3D Map* 버튼을 클릭하고항목 생성 지침 을 따라 새 3D 맵 항목을 추가하십시오. 3D 맵 항목을 추가한 후대화형 조판기 항목 작업 에서 설명한 내용과 동일한 방법으로 수정할 수 있습니다.

새 3D 맵 항목은 기본적으로 비어 있습니다. *Item Properties* 패널에서 3D 맵 뷰의 속성을 설정하고 사용자 지정할 수 있습니다. 이 항목은공통 속성 이외에도 다음과 같은 기능을 가지고 있습니다 (그림 18.22 그림 참조):

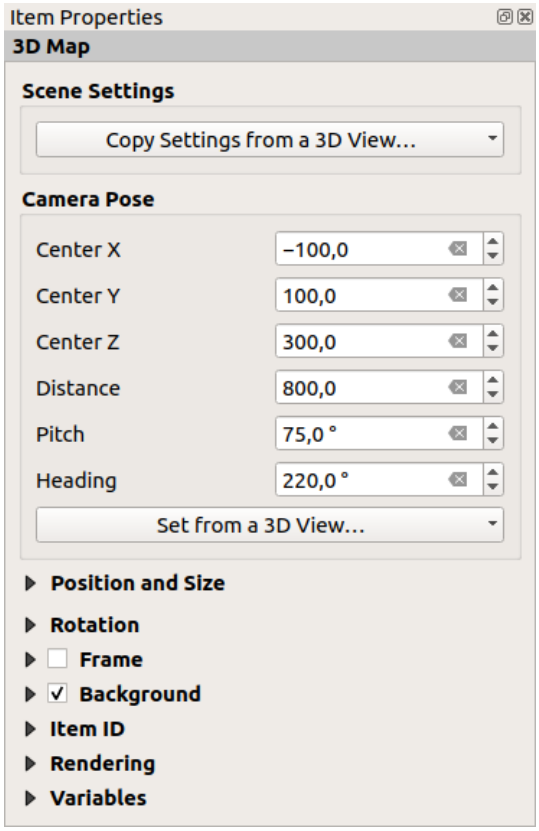


그림 18.22: 3D 맵 항목 속성

신 (Scene) 설정

Copy Settings from a 3D View…버튼을 클릭해서 표시할 3D 맵 뷰를 선택하십시오.

선택한 3D 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 3D 맵 항목을 3D 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* 도구로 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 18.23 참조):

주요 속성

Main properties 그룹에서 라벨 작성에 필요한 텍스트 (HTML 도 가능) 또는 표현식을 작성할 수 있습니다. 표현식을 표현식으로 해석되게 하려면, 표현식을 `[%` 와 `%]` 로 둘러싸야 합니다.

- *Render as HTML*: 이 옵션을 활성화하면 라벨을 HTML 코드로 해석시킬 수 있습니다. URL, 웹페이지로 연결되는 클릭할 수 있는 이미지, 또는 더 복잡한 무언가를 삽입할 수 있습니다.
- *Insert or Edit an expression*…버튼을 클릭하면표현식 도 사용할 수 있습니다. 대화창이 열리면 동일한 방법으로 공식을 작성하십시오. QGIS 가 괄호 문자를 자동적으로 추가합니다.

참고: 텍스트란에서 아무것도 선택하지 않았을 때 *Insert or Edit an expression*…버튼을 클릭하면 기존 텍스트에 새 표현식을 덧붙일 것입니다. 기존 텍스트를 업데이트하고 싶다면, *Insert an Expression* 버튼을 클릭하기 전에 텍스트에서 관심 부분을 선택해야 합니다.

HTML 렌더링과 표현식을 결합해서 고급 라벨 작업을 할 수 있습니다. 다음 코드는 그림 18.24 를 산출할 것입니다:

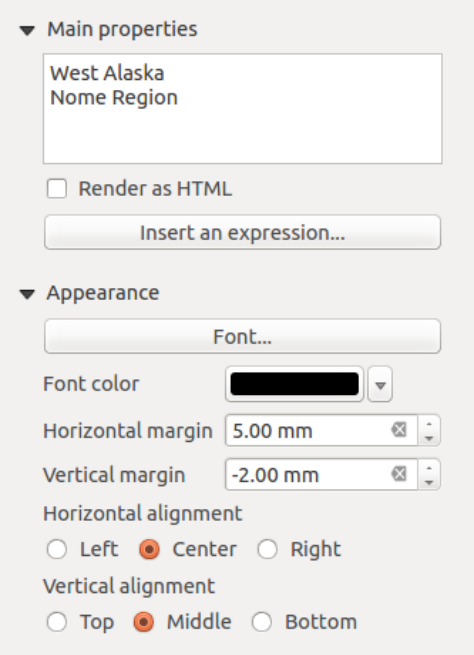


그림 18.23: 라벨 항목 속성 패널

```
<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 18.24: HTML 스타일 작업으로 라벨에 영향 주기

모양

- *Font*…버튼을 눌러서 *Font* 를 정의하거나, 또는색상 위젯 을 눌러 *Font color* 를 선택하십시오.
- 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*, *Middle*, *Bottom*

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
)
```

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 |label| *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 18.25 참조):

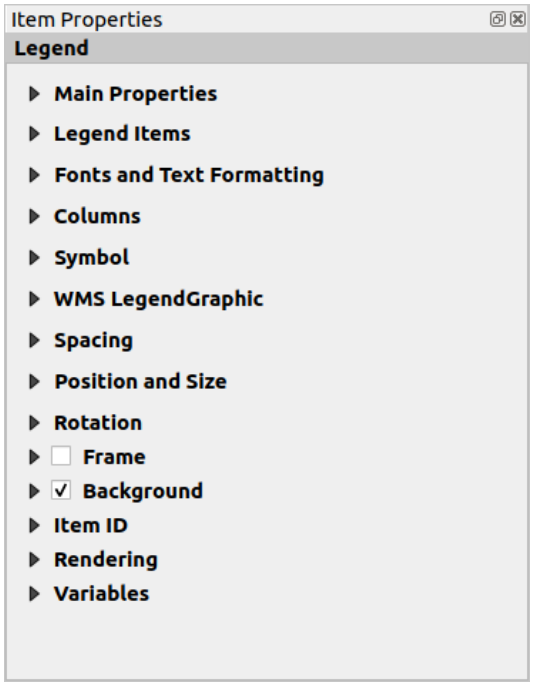


그림 18.25: 범례 항목 속성 패널

주요 속성

범례 *Item Properties* 패널의 *Main properties* 그룹은 다음 기능을 제공하고 있습니다. (그림 18.26 을 참조하세요.):

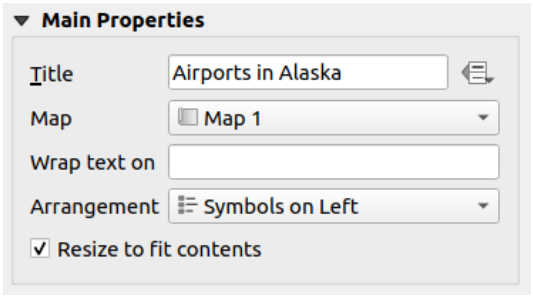


그림 18.26: 범례 주요 속성 그룹

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.

- 범례의 텍스트에 행갈이 문자를 설정할 수 있습니다. 해당 문자가 나타날 때마다 행을 바꿀 것입니다.

- 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.

범례 항목

범례 *Item Properties* 패널의 *Legend items* 그룹은 다음 기능을 제공하고 있습니다. (그림 18.27 을 참조하세요.):

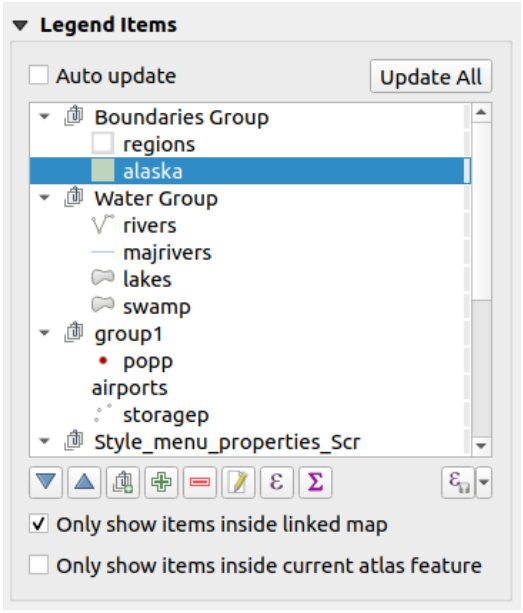


그림 18.27: 범례 항목 그룹

- *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, 레이어명을 편집하고, 필터를 추가할 수 있습니다.
  - 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)

- Σ 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. 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

ε 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. ε 버튼을 클릭해서 *Expression String Builder* 대화창을 여십시오.
3. 다음 표현식을 입력하십시오 (심볼 라벨은 편집되지 않았다고 가정합니다):

```
concat( @symbol_label,
        ' (', @symbol_count, ') - ',
        round( aggregate(@layer, 'sum', $area, filter:= "type"=@symbol_label)/
↪10000 ),
        'ha'
      )
```

4. *OK* 를 클릭하십시오.

글꼴

범례 *Item Properties* 패널의 *Fonts* 그룹은 다음 기능을 제공하고 있습니다:

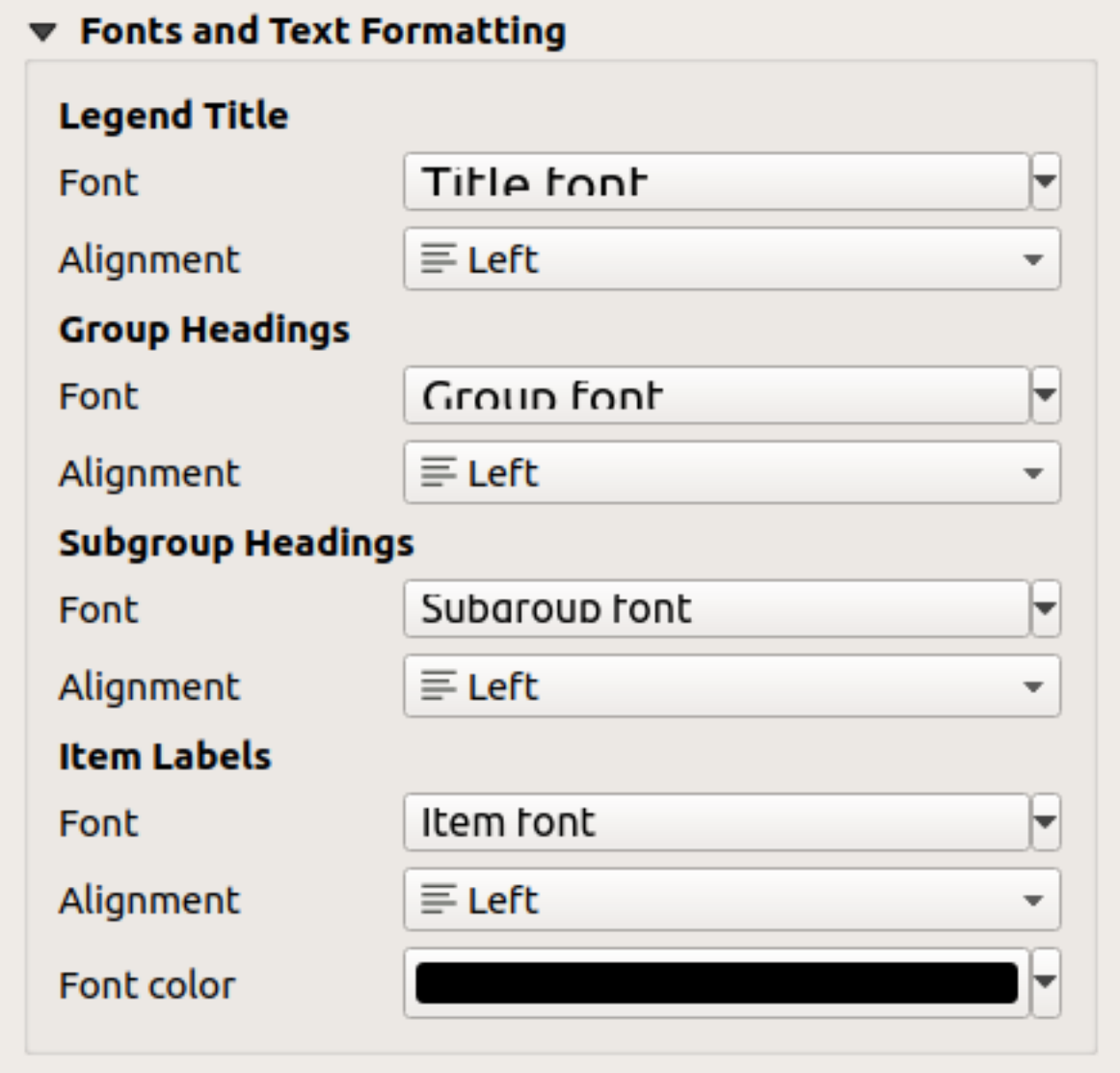


그림 18.28: 범례 글꼴 속성

- 글꼴 선택기 위젯을 사용해서 범례 항목에 있는 범례 제목, 그룹, 하위 그룹 및 항목 (피처) 의 글꼴을 변경할 수 있습니다.
- 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.
- 색상 선택기 를 이용해서 라벨의 *Color* 를 설정할 수 있습니다. 선택한 색상이 범례에 있는 모든 글꼴 항목에 적용될 것입니다.

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.

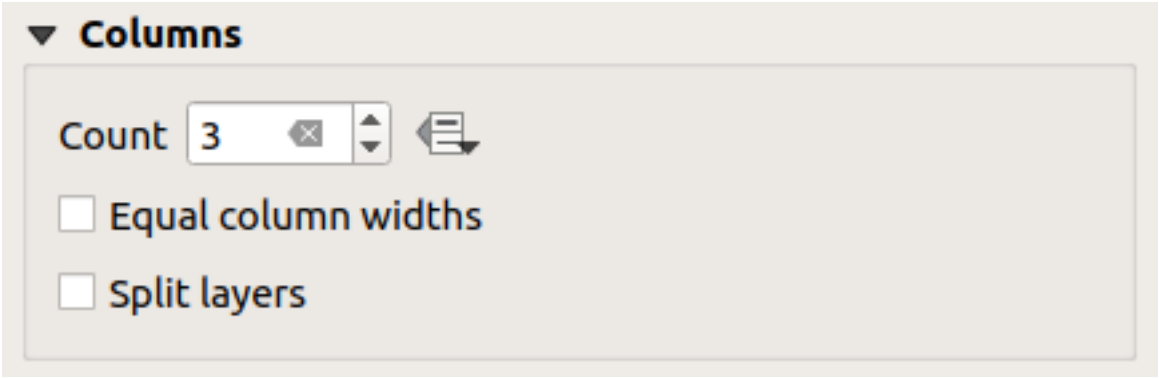


그림 18.29: 범례 열 설정

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 및 간격

범례 *Item Properties* 패널의 *WMS LegendGraphic* 및 *Spacing* 그룹은 다음 기능을 제공하고 있습니다. (그림 18.31 을 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.

이 대화창에서 제목, 그룹, 하위 그룹, 심볼, 라벨, 박스, 열 및 선 주위의 *Spacing* 을 사용자 지정할 수 있습니다.

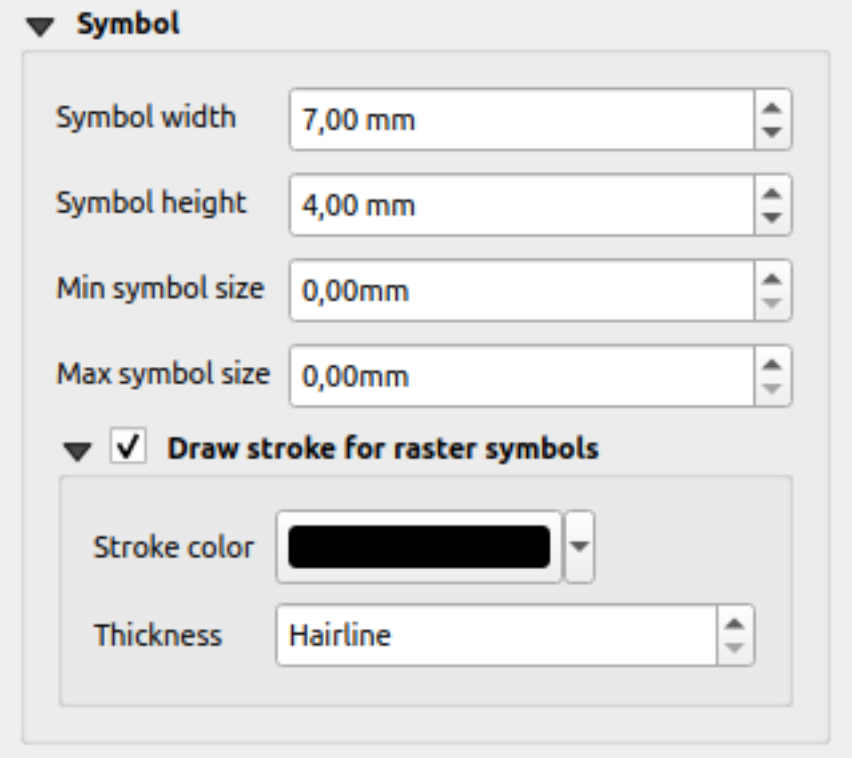


그림 18.30: 범례 심볼 환경 설정

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 18.32 참조):

주요 속성

축척 막대 *Item Properties* 패널의 *Main properties* 그룹은 다음 기능을 제공하고 있습니다. (그림 18.33 을 참조하세요.):

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`)

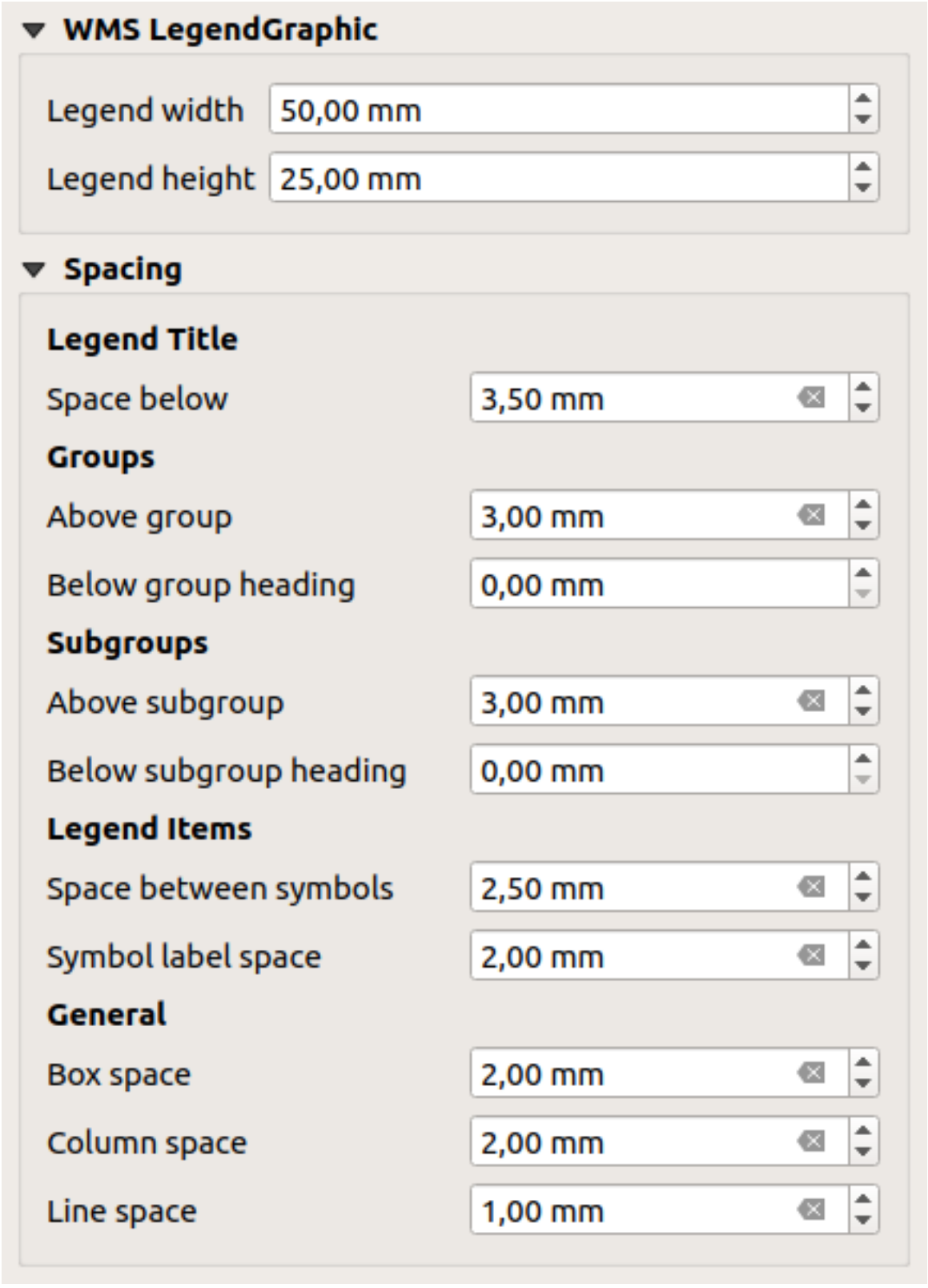


그림 18.31: WMS LegendGraphic 및 간격 그룹

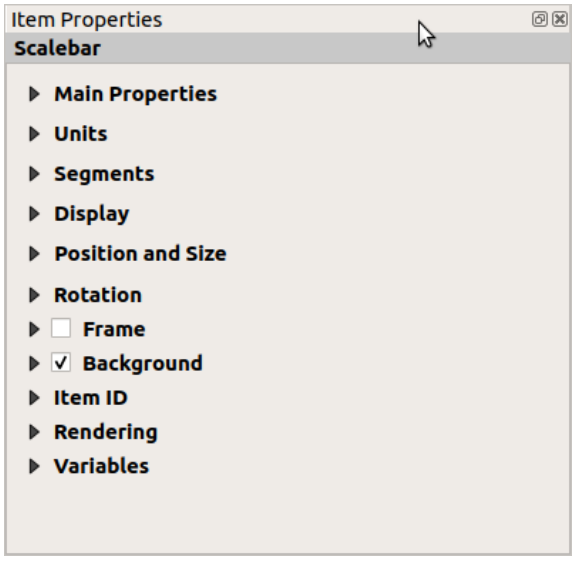


그림 18.32: 축척 막대 항목 속성 패널

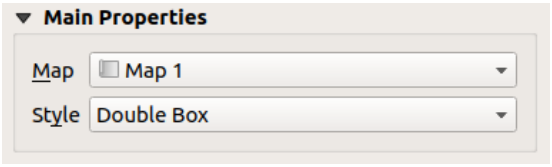


그림 18.33: 축척 막대 주요 속성 그룹

3. 속성을 적절하게 설정하십시오.

단위

축척 막대 *Item Properties* 패널의 *Units* 그룹은 표시될 단위와 텍스트 서식을 설정하기 위한 다음 기능을 제공하고 있습니다. (그림 18.34 를 참조하세요.):

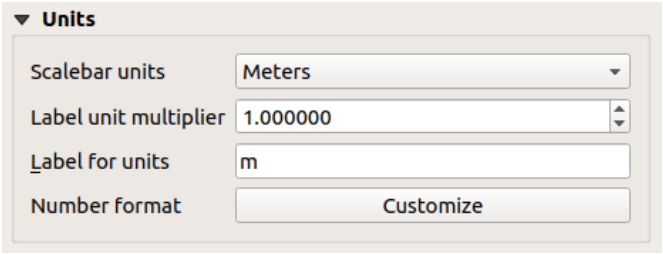


그림 18.34: 축척 막대 단위 그룹

- *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 매우 유용합니다. (예: 로케일 기본값은 천단위 구분자를 감추는 것이지만 천단위 구분자를 사용하고 싶은 경우)

선분

축척 막대 *Item Properties* 패널의 *Segments* 그룹은 선분과 그 분할의 개수와 크기를 환경 설정하기 위한 다음 기능을 제공하고 있습니다. (그림 18.35 를 참조하세요.):

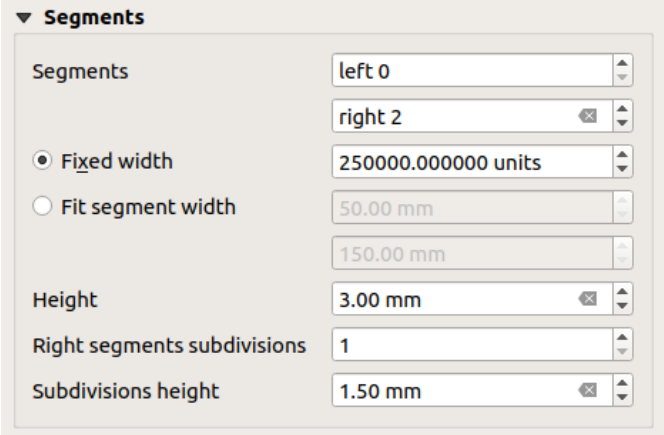


그림 18.35: 축척 막대 선분 그룹

- *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
- *Fixed width* 옵션에서 분할 1 개가 나타내는 거리를 설정할 수 있고, 또는 *Fit segment width* 옵션에서 축척 막대 크기를 mm 단위로 제한할 수도 있습니다. 후자의 경우, 맵 축척이 변경될 때마다 설정된 범위에 맞춰 축척 막대의 크기도 조정됩니다 (그리고 축척 라벨도 업데이트됩니다).
- *Height*: 축척 막대의 높이를 정의할 수 있습니다.
- *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.

표시

축척 막대 *Item Properties* 패널의 *Display* 그룹은 다음 기능을 제공하고 있습니다:

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.
- *Vertical label placement*: 라벨을 축척 막대 분할의 위 또는 아래 배치합니다.
- *Horizontal label placement*: 라벨을 축척 막대 분할의 경계 또는 중심에 중앙 정렬합니다.

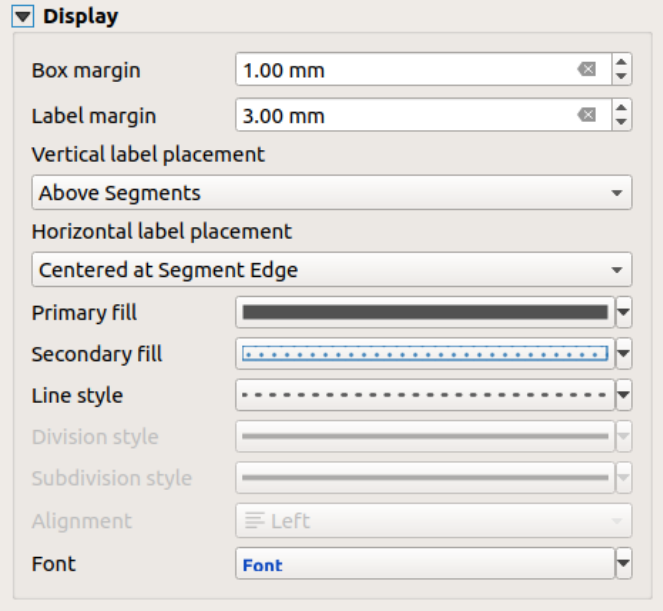


그림 18.36: 축척 막대 표시 그룹

- *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:

- 속성 테이블: 사전 정의된 규칙을 기반으로 레이어 속성의 하위 집합을 표시합니다.
- 고정 테이블: 정보가 레이어 독립적일 수 있는 수작업 텍스트 테이블을 삽입합니다.

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 18.37 참조):

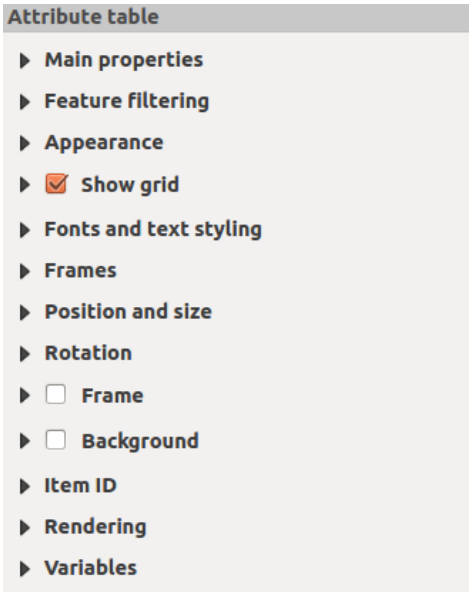


그림 18.37: 속성 테이블 항목 속성 패널

주요 속성

속성 테이블의 *Main properties* 그룹은 다음 기능을 제공하고 있습니다. (그림 18.38 을 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 18.41) 가 *Layer* 드롭다운 목록에 표시된 레이어의 테이블 구조이며 그대로 남아 있다는 사실을 기억하십시오. 즉 서로 다른 필드 (들) 을 가진 레이어에 데이터 정의 테이블을 설정하면 테이블에 비어 있는 열 (들) 이 생길 것입니다.

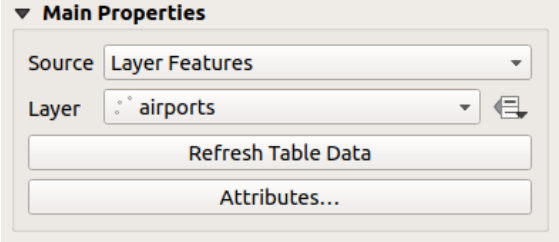


그림 18.38: 속성 테이블 주요 속성 그룹

Atlas 패널에서 *Generate an atlas* 옵션을 활성화한 경우 (지도책 생성 참조) 다음과 같은 *Source* 2 개를 추가로 선택할 수 있습니다:

- **Current atlas feature** (그림 18.39 참조): 레이어를 선택할 수 있는 옵션은 표시되지 않으며, 테이블 항목은 지도책 커버리지 레이어의 현재 피처에서 나온 속성을 가진 행만 표시할 것입니다.
- **Relation children** (그림 18.40 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.

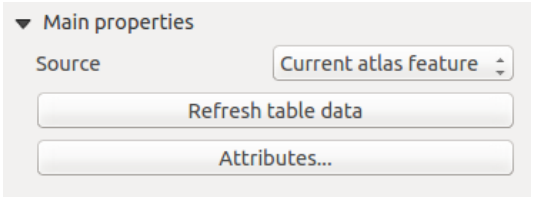


그림 18.39: 'Current atlas feature'에 대한 속성 테이블 주요 속성

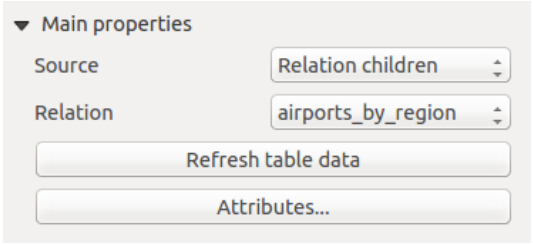


그림 18.40: 'Relation children'에 대한 속성 테이블 주요 속성

- *Attributes*···: *Select Attributes* 대화창을 엽니다. (그림 18.41 을 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.

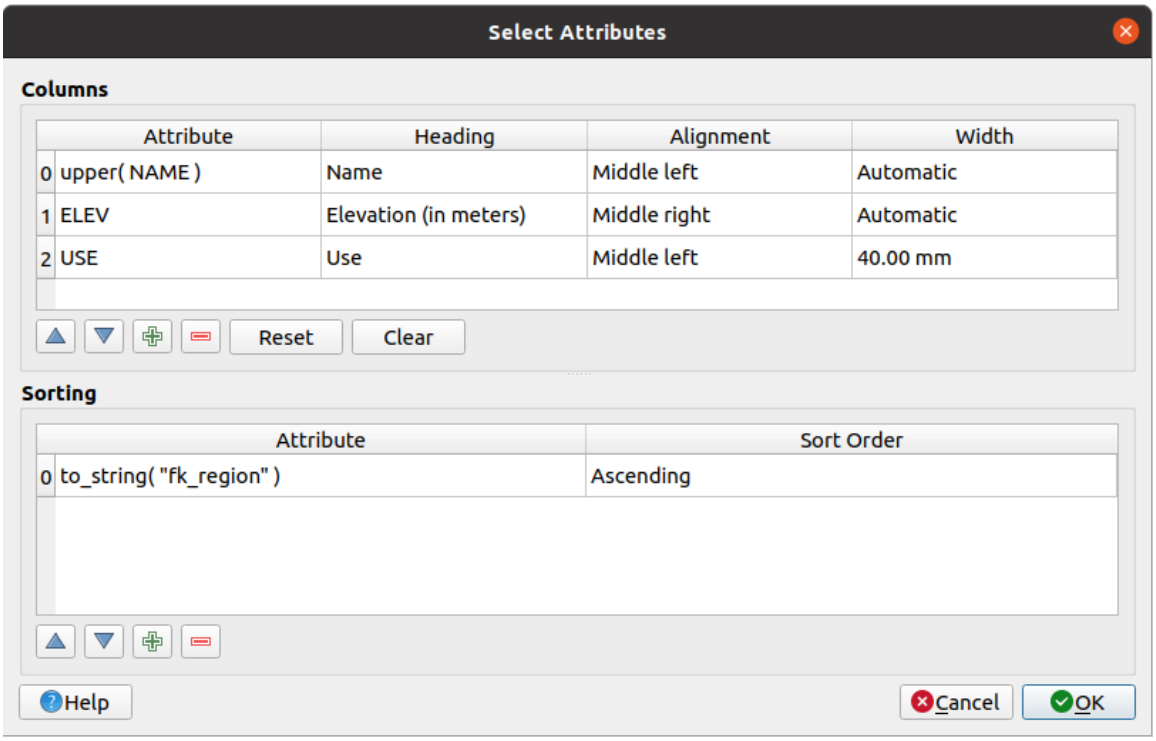


그림 18.41: 속성 테이블 속성 선택 대화창

- *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.
- *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 18.42 를 참조하세요.):

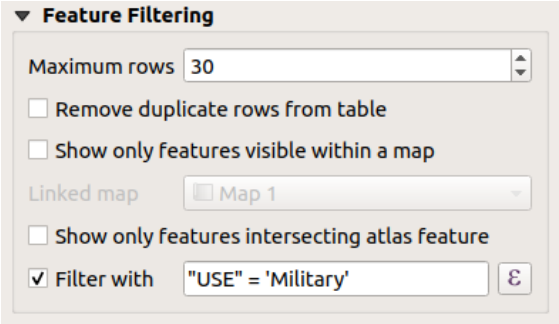


그림 18.42: 속성 테이블 피처 필터링 그룹

이 대화창에서 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.
- *Filter with* 옵션을 활성화하면 입력 란에 필터를 직접 입력하거나 또는 옆에 있는 ε 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 18.43 을 참조하세요.):

- *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.

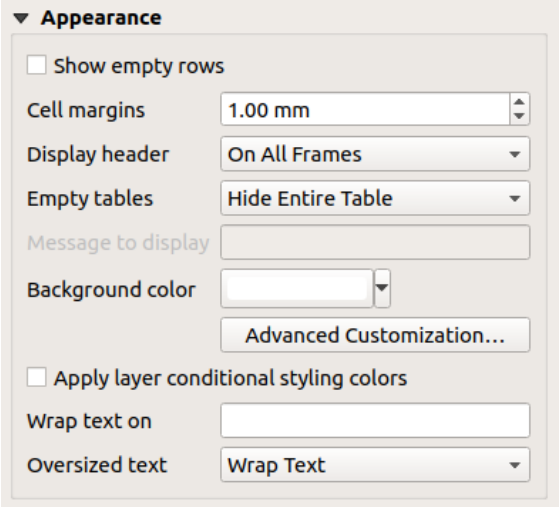


그림 18.43: 속성 테이블 모양 그룹

  - **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 18.44 를 참조하세요.)
- *Apply layer conditional styling colors*: 조판 속성 테이블 내부에 레이어가 가진조건부 테이블 서식 을 적용합니다 (현재 배경색과 전경색만 지원합니다). 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.

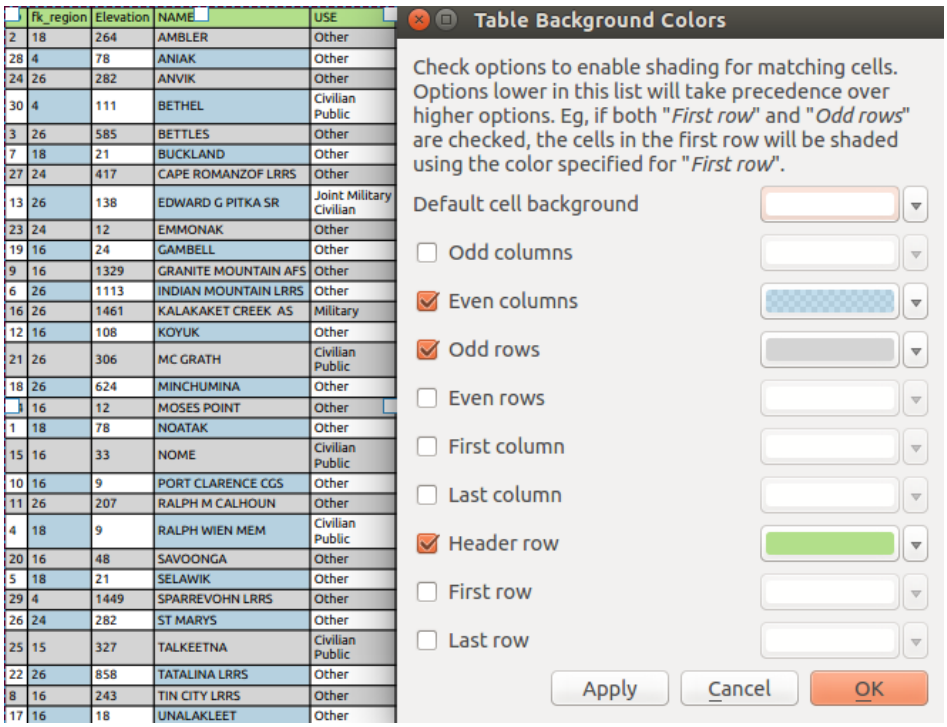


그림 18.44: 속성 테이블 고급 배경 대화창

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

Main properties 에서 *Edit table* …을 클릭하면 *Table designer* 를 작업할 수 있습니다:

- 테이블 내부를 클릭하면 텍스트를 직접 삽입할 수 있습니다.
- 상단 메뉴를 통해 다음 작업을 할 수 있습니다:
  - *File* 메뉴의 *Import Content From Clipboard*: 클립보드에 있는 내용을 불러옵니다 (직접 입력한 내용을 덮어씁니다).
  - *Edit*: 행 및 열에 대해 선택 작업을 할 수 있습니다.
  - *Include Header Row* 옵션은 물론, *Insert rows*, *Insert columns*, *Delete Rows*, *Delete Columns* 등의 작업을 할 수 있습니다.
- 우측에 있는 *Cell Contents* 부분에서 다음 작업을 할 수 있습니다:
  - *Formatting* 에서 선택한 셀의 텍스트 서식을 다음과 같이 정의할 수 있습니다:
    - ε 표현식 버튼을 클릭하면 셀에 정규 표현식을 입력할 수 있습니다.

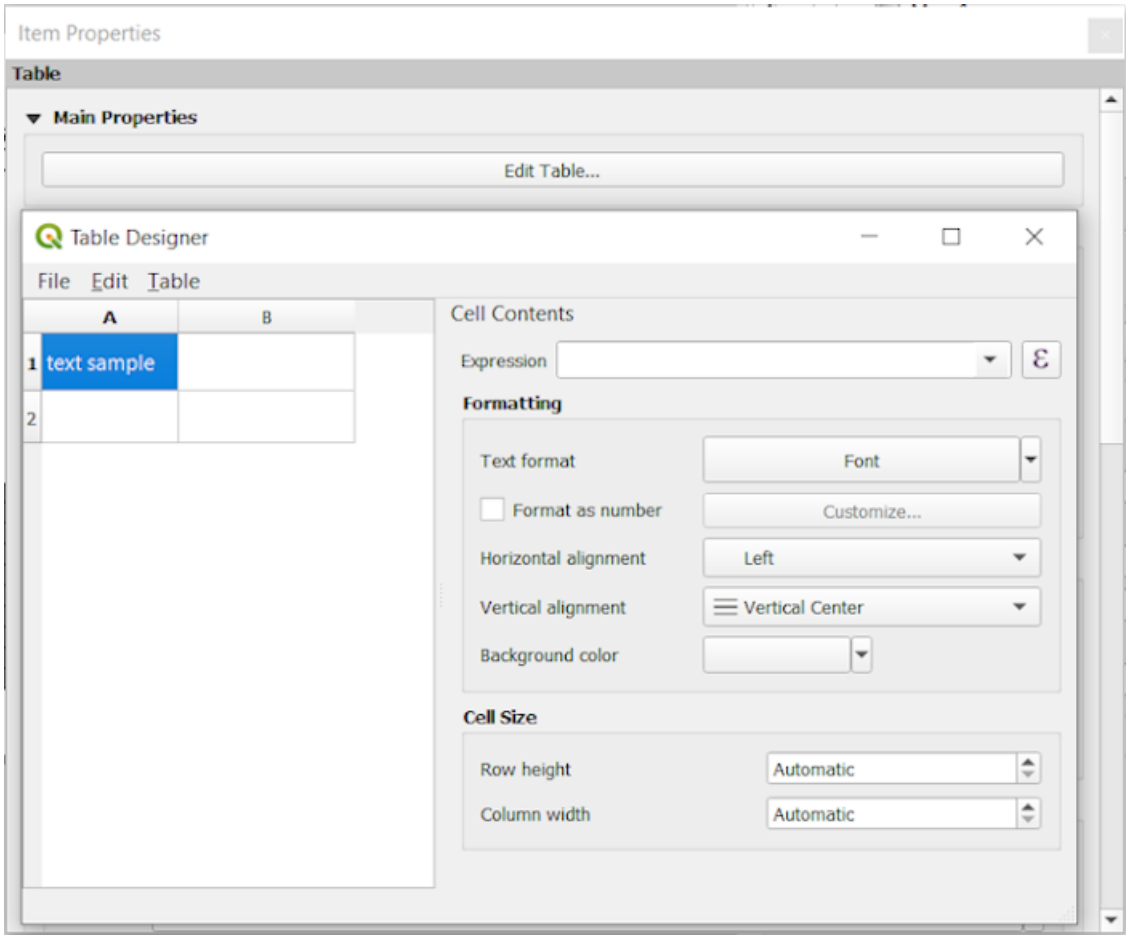


그림 18.45: 테이블 설계자를 띄운 고정 테이블 항목 속성 패널

    * *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 18.46 을 참조하세요.):

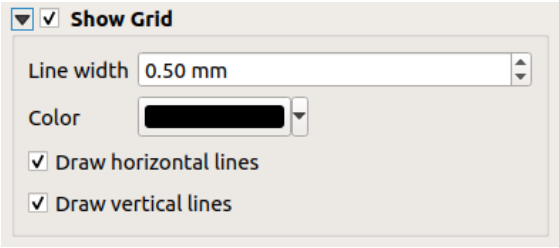


그림 18.46: 속성 테이블 그리드 표시 그룹

- *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 18.47 을 참조하세요.):



그림 18.47: 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.
- *Table heading* 의 경우 추가적으로 *Alignment* 옵션에서 `Follow column alignment` 를 선택하거나, 또는 `Left`, `Center`, `Right` 가운데 하나를 선택해서 열 정렬 설정을 무시할 수 있습니다. *Select Attributes* 대화창에서 열 정렬을 설정합니다. (그림 18.41 을 참조하세요.)

프레임

테이블 항목 속성의 *Frames* 그룹은 다음 기능을 제공하고 있습니다. (그림 18.48 을 참조하세요.):

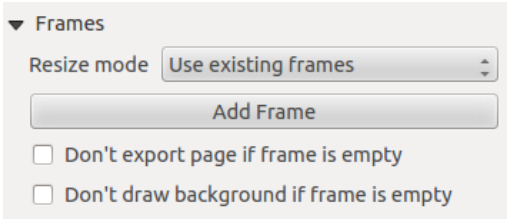


그림 18.48: 속성 테이블 프레임 그룹

- *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 그림과 방위표

Picture 항목은 사용자 맵을 그림, 로고 등으로 장식할 수 있게 해줍니다. 전용방위표 도구 대신, 이 항목을 이용해서 방위표를 추가할 수도 있습니다.

그림 항목

사용자의 파일 관리자에서 캔버스 위로 그림을 드래그해오거나, 항목 생성 지침 을 따라 Add Picture 도구를 사용하면 그림을 추가할 수 있습니다. 이후대화형 조판기 항목 작업 에서 설명하는 대로 새 그림 항목을 수정할 수 있습니다.

Add Picture 도구를 사용하면, 그림 항목은 *Item Properties* 패널에서 사용자 지정할 수 있는 빈 프레임이 될 것입니다. 이 항목은항목 공통 속성 이외에도 다음과 같은 기능을 가지고 있습니다 (그림 18.49 참조):

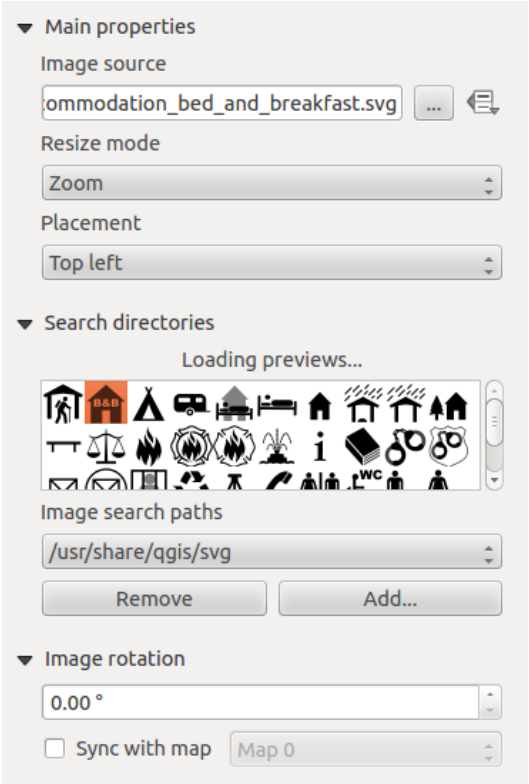


그림 18.49: 그림 항목 속성 패널

여러 가지 방법으로 (사용자가 표시하고자 하는 이미지를 선택하기 위한) *Image source* 를 설정할 수 있습니다:

1. *Main properties* 그룹에서 *image source* 의 … Browse 버튼을 클릭해서 사용자 컴퓨터에 있는 파일을 선택하십시오. 탐색기가 QGIS 가 제공하는 SVG 라이브러리 폴더를 열 것입니다. `.png` 또는 `.jpg` 같은 다른 이미지 포맷도 선택할 수 있습니다.
2. *Image source* 텍스트란에 소스를 직접 입력할 수 있습니다. 이미지를 가리키는 원격 URL 주소도 입력할 수 있습니다.
3. *Search directories* 부분에서 불러온 미리보기에서 이미지를 선택해서 이미지 소스를 설정할 수 있습니다. 기본적으로 *Settings* → *Options* → *System* → *SVG Paths* 에서 설정한 폴더에서 이 이미지들을 불러옵니다.
4. data defined override 버튼을 클릭하면 피처 속성으로부터, 또는 정규 표현식을 이용해서 이미지 소스를 설정할 수 있습니다.

참고: *Search directories* 그룹에 있는 *Add* 및 *Remove* 버튼을 이용해서 미리보기 이미지를 불러올 폴더 목록을 사용자 정의할 수 있습니다.

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* 모드에 따라 *Placement* 및 *Image rotation* 옵션이 비활성화될 수도 있습니다. *Placement* 옵션을 사용하면 이미지 프레임 내부에서 이미지의 위치를 선택할 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)"`

예시를 보고 싶다면 이 블로그 게시글 을 읽어보십시오.

Image rotation 필드에서 이미지를 기울일 수 있습니다. *Sync with map* 체크박스를 활성화하면, 선택한 맵 항목에 적용된 기울기와 이미지의 기울기를 동기화합니다. 이 기능은 방위표를 다음 가운데 하나에 맞춰 정렬할 수 있는 편리한 기능입니다:

- **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.

방위표 항목

항목 생성 지침 을 따라|addMap| *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.

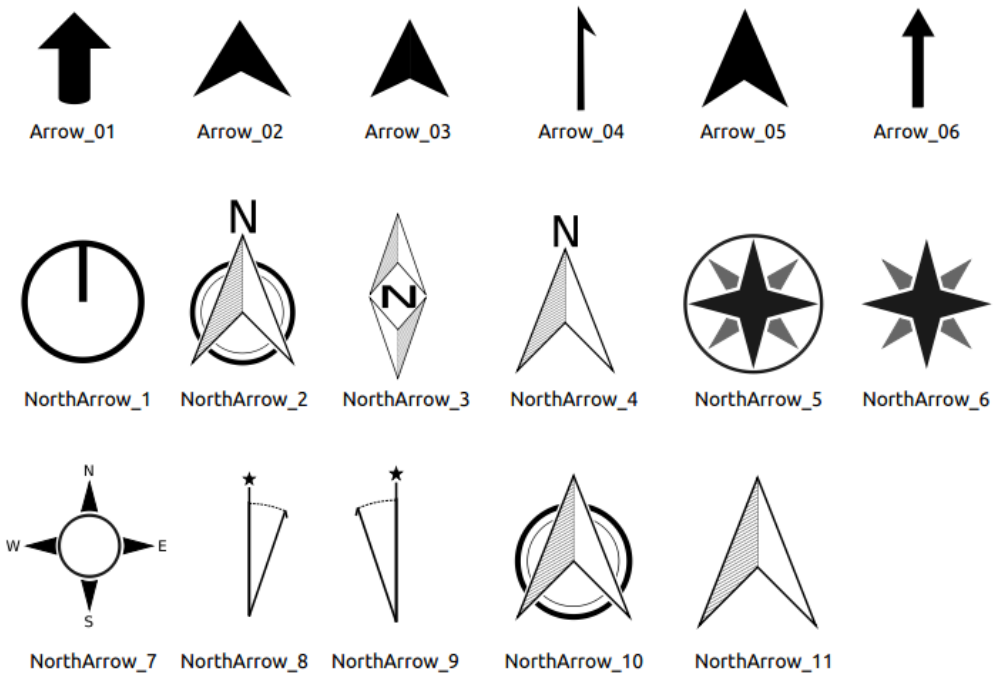


그림 18.50: SVG 라이브러리에서 선택하고 사용할 수 있는 방위표들

18.2.9 HTML 프레임 항목

웹사이트의 내용을 표시하는 프레임을 추가하거나, 또는 사용자 고유의 HTML 페이지를 생성하고 스타일을 적용할 수 있습니다. 항목 생성 지침 을 따라|label| *Add HTML* 도구로 HTML 프레임을 추가하고대화형 조판기 항목 작업 에서 설명한 내용과 동일한 방법으로 수정할 수 있습니다. HTML 프레임 생성시의 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 18.51 참조):

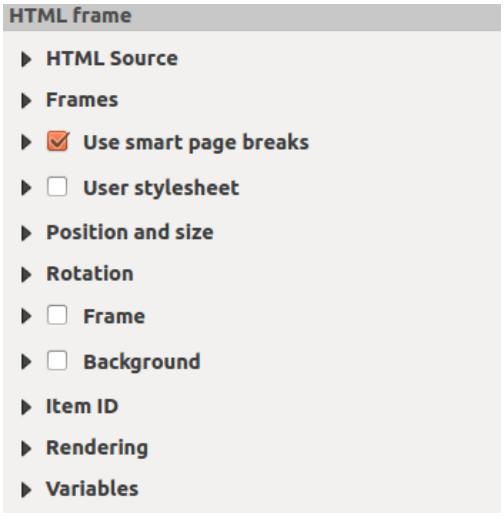


그림 18.51: HTML 프레임 항목 속성 패널

HTML 소스

HTML 프레임 *Item Properties* 패널의 *HTML Source* 그룹은 다음 기능을 제공하고 있습니다. (그림 18.52 를 참조하세요.):

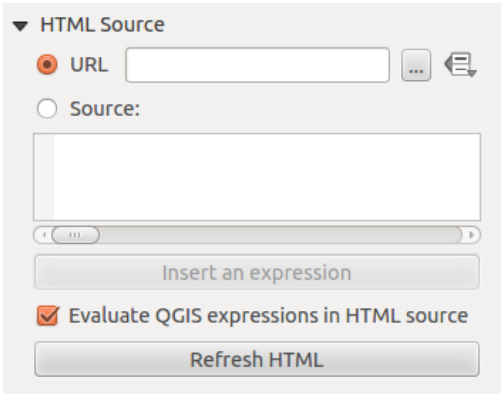


그림 18.52: HTML 프레임 항목의 HTML 소스 속성 그룹

- *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 안쪽 아무데나 클릭하십시오. 그러지 않으면 표현식이 사라질 것입니다.
- *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 18.53 을 참조하세요.):

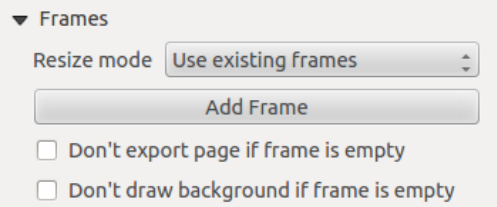


그림 18.53: HTML 프레임 항목의 프레임 속성 그룹

- *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 18.54 를 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.
- ☑ *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.

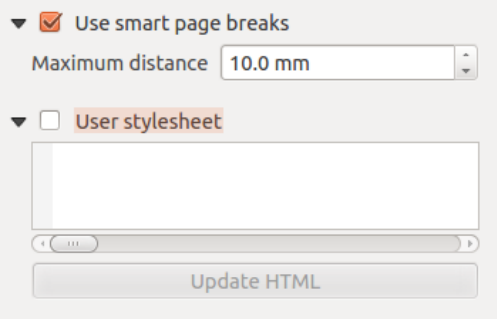


그림 18.54: HTML 프레임 항목의 스마트 페이지 나누기 사용 및 사용자 스타일시트 속성 그룹

```
h1 {color: #00ff00;
}
p {font-family: "Times New Roman", Times, serif;
   font-size: 20px;
}
```

- *Update HTML* 버튼을 클릭면 스타일시트 설정의 결과를 볼 수 있습니다.

18.2.10 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 18.55 참조):

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 서로 다른 단위로 설정할 수 있습니다. 모서리가 둥글어집니다.

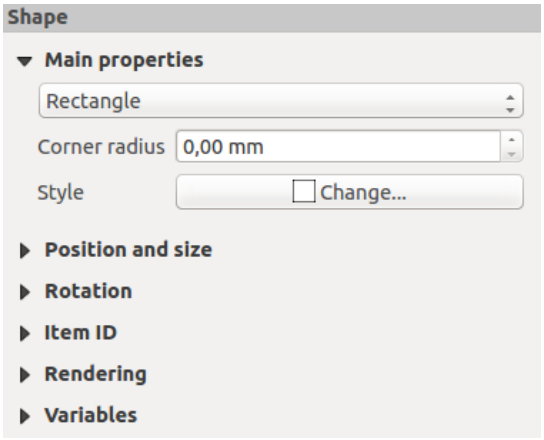


그림 18.55: 도형 항목 속성 패널

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. Select either Add Polygon or Add Polyline tool
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.

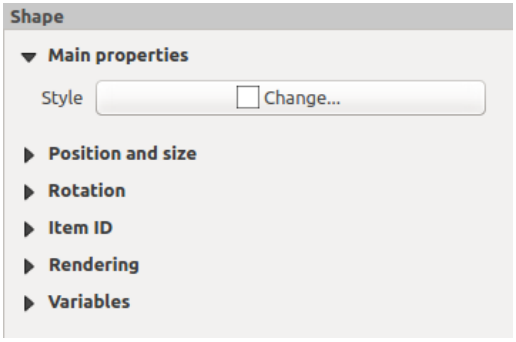


그림 18.56: 폴리곤 노드 도형 항목 속성 패널

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.
- 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.

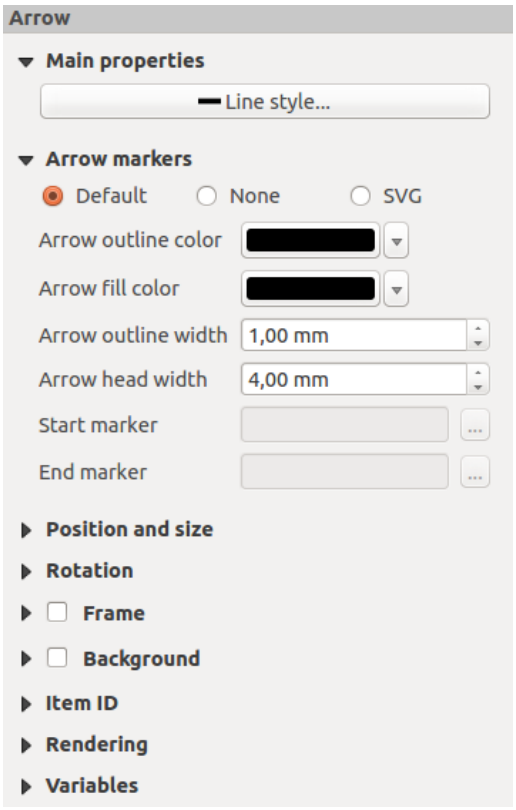


그림 18.57: 폴리라인 노드 도형 항목 속성 패널

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 18.58 은 앞에서 설명한 조판기 항목의 모든 유형을 포함하고 있는 인쇄 조판 예시를 보여주고 있습니다.

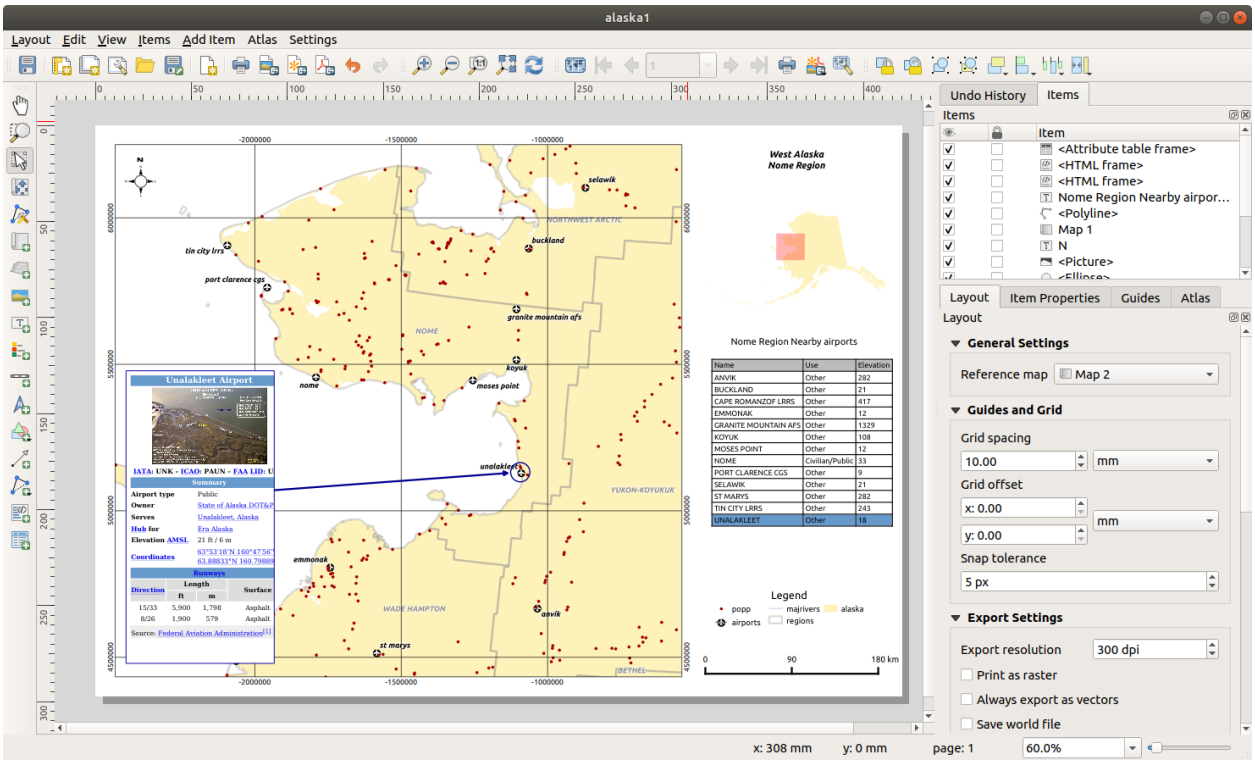


그림 18.58: 맵 뷰, 범례, 이미지, 축척 막대, 좌표, 텍스트 및 HTML 프레임을 추가한 인쇄 조판기

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

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.
   - *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 파일명은 같지만 확장자는 다른 (TIFF 는 `.tfw`, PNG 는 `.pnw`,JPEG 은 `jgw` 등등) *ESRI* 월드 파일을 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.

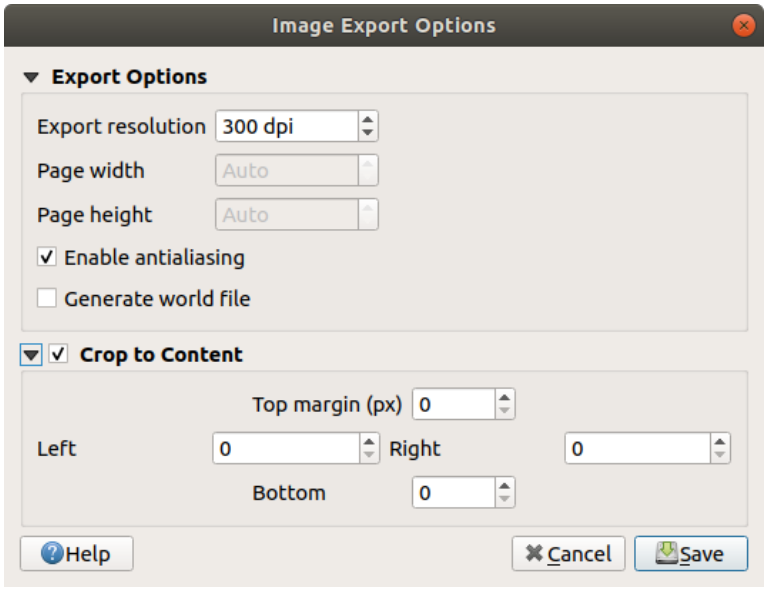


그림 18.59: 이미지 내보내기 옵션: 항목 범위로 산출물 크기를 조정

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 명칭을 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.)
- *Text export*: 텍스트 라벨을 제대로 된 텍스트 오브젝트로 내보낼지 (*Always export texts as text objects*) 경로로만 내보낼지 (*Always export texts as paths*) 설정합니다. 텍스트 라벨을 텍스트 객체로 내보낼 경우 (잉크스케이프 같은) 외부 응용 프로그램에서 일반 텍스트처럼 편집할 수 있습니다. **하지만** 렌더링의 질이 하락하는 부작용이 있습니다. **게다가** 버퍼 같은 특정 텍스트 설정이 적용된 경우 렌더링 오류가 생길 수도 있습니다. 경로로 내보내기를 추천하는 이유입니다.
- ☑ *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.

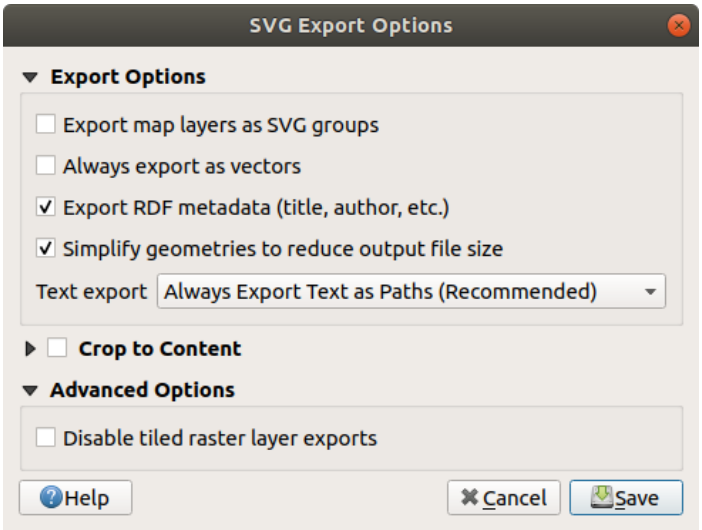


그림 18.60: SVG 내보내기 옵션

참고: 현재 SVG 출력물은 매우 기초적입니다. 이것은 QGIS 의 문제가 아니라, 기저 Qt 라이브러리의 문제입니다. Qt 향후 버전에서 해결되기를 바라고 있습니다.

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.)
   - ☑ *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.
   - *Text export*: 텍스트 라벨을 제대로 된 텍스트 객체로 내보낼지 (*Always export texts as text objects*) 경로로만 내보낼지 (*Always export texts as paths*) 설정합니다. 텍스트 라벨을 텍스트 객체로 내보낼 경우 (잉크스케이프 같은) 외부 응용 프로그램에서 일반 텍스트처럼 편집할 수 있습니다. **하지만** 렌더링의 질이 하락하는 부작용이 있습니다. **게다가** 버퍼 같은 설정이 적용된 특정 텍스트의 경우 렌더링 오류가 생길 수도 있습니다. 경로로 내보내기를 추천하는 이유입니다.
   - *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 (GeoPDF)*: 이 옵션을 활성화하면 지리참조 PDF 파일 을 생성합니다. (GDAL 버전 3 이상이 필요합니다.)
   - ☐ *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.)

참고: QGIS 3.10 버전부터, GDAL 3 을 설치했다면, GeoPDF 내보내기를 지원합니다. 다음과 같은 GeoPDF 특화 옵션을 사용할 수 있습니다:

- *Format*: GeoPDF 포맷을 지정합니다 — GeoPDF 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.

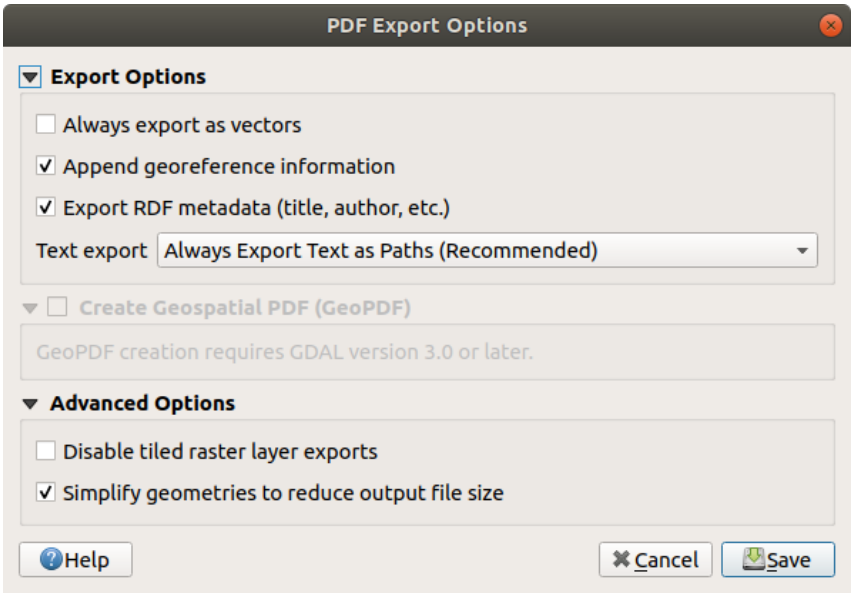


그림 18.61: PDF 내보내기 옵션

참고: 인쇄 조판을 지리 참조를 지원하는 포맷으로 (예: `PDF` 및 `TIFF`) 내보내면 기본적으로 지리 참조된 산출물을 생성합니다.

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 관련부모 또는 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 18.62 참조):

- *Generate an atlas*: 지도책 생성 기능을 활성화/비활성화합니다.
- *Configuration*

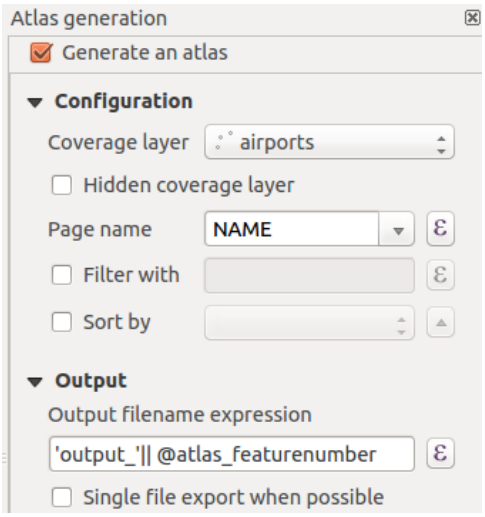


그림 18.62: 지도책 패널

  - *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 명칭을 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|arrowUp| 또는|arrowDown| 로 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.

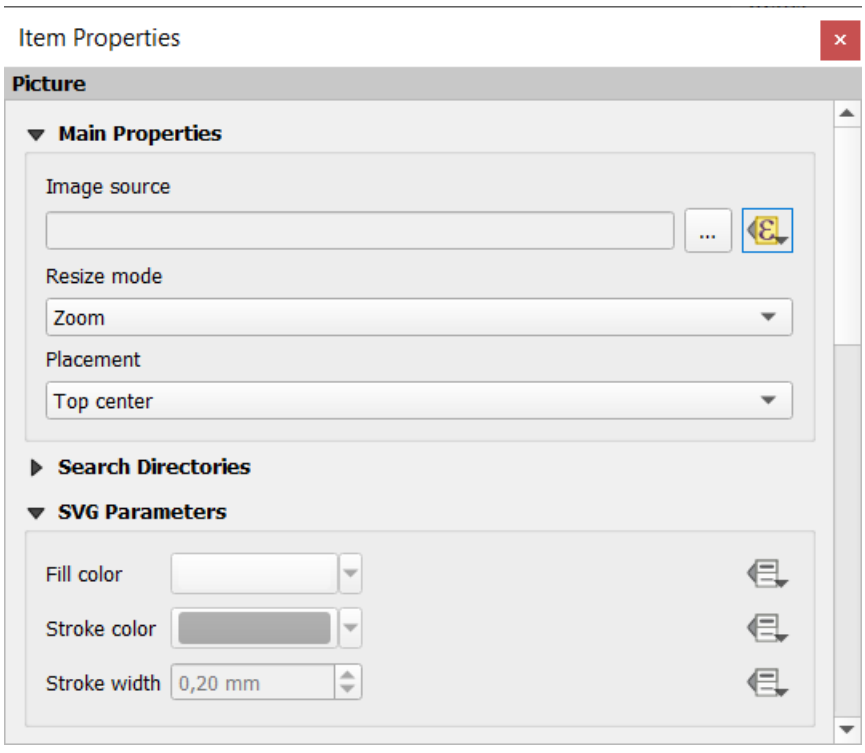
페이지 중심에 있는 맵 상단에 제목을 추가해봅시다. 라벨 항목을 선택하고 수평 정렬을 *Center* 로 설정하십시오. 그 다음 라벨을 올바른 위치로 이동시키고, *Reference point* 를 가운데 라디오 버튼으로 설정하십시오. 그리고 *X* 필드에 다음 표현식을 입력하십시오:

```
@layout_pagewidth / 2
```

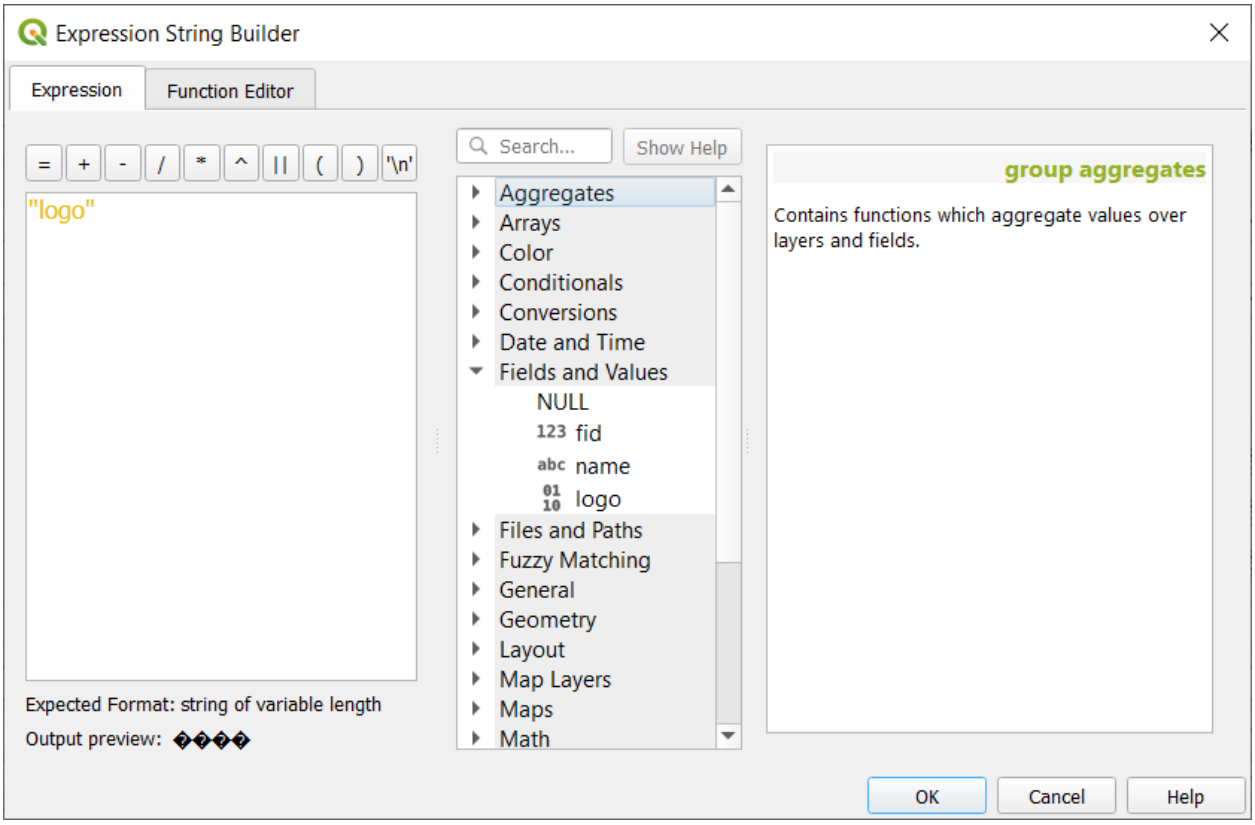
다른 모든 조판기 항목에 대해 비슷한 방법으로 위치를 설정할 수 있습니다. 이렇게 동적으로 위치를 설정하면 용지 방향이 포트레이트든 랜드스케이프든 조판기 항목들이 정확히 배치됩니다. 또 제목을 피처 속성으로 사용자 지정하거나 (표현식으로 라벨 사용자 지정 예시 참조), 이미지를 변경하거나, 용지 방향에 따라 범례의 열 개수를 재조정하는 등 더 많은 미세 조정도 할 수 있습니다.

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



그림 18.63: 지도책 미리보기 툴바

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 명칭을 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 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 자식 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 자식 피처의 동적인 불릿 (bullet, 작은 점) 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 자식 피처의 불릿 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.

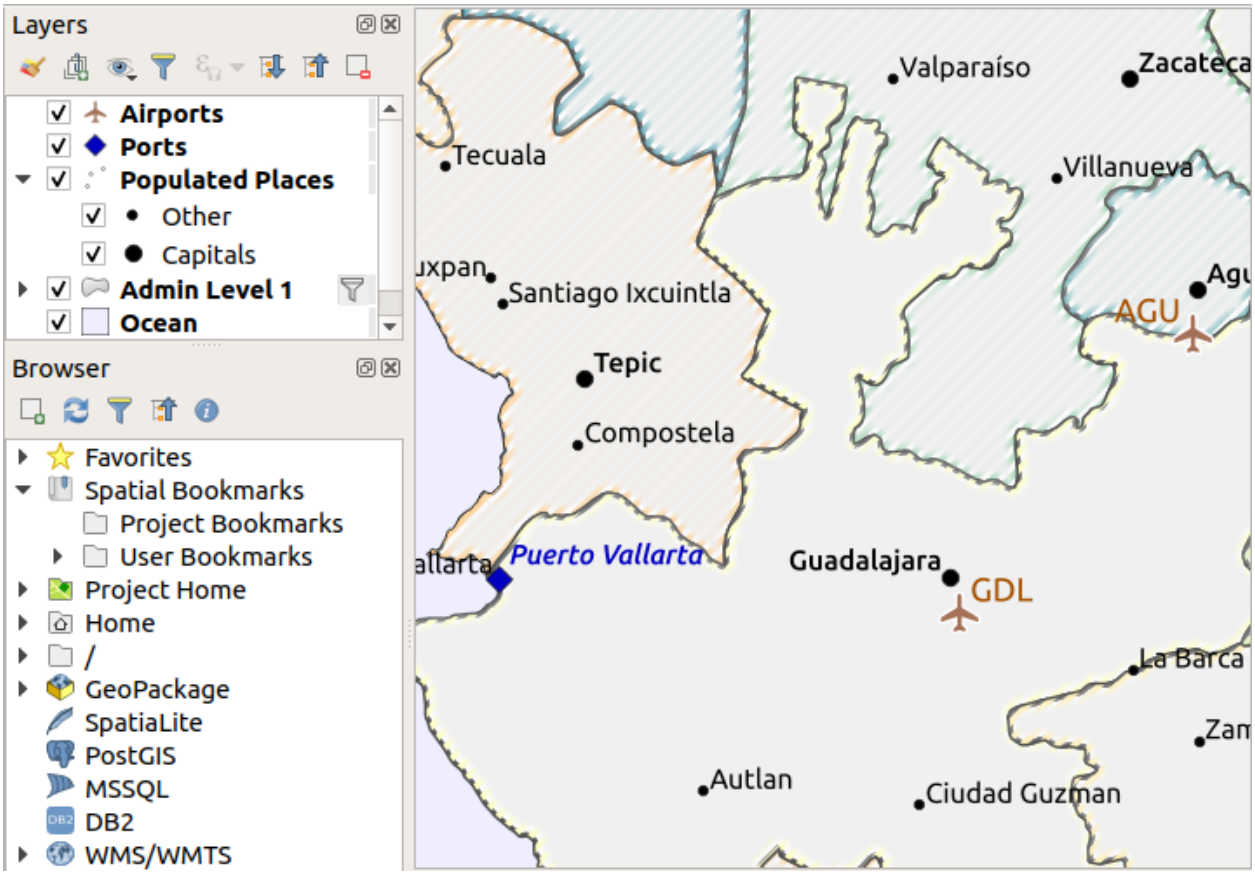
Project → New Report 메뉴 옵션을 선택하면, 비어 있는 보고서를 생성합니다. 처음에는 별로 볼 것도 없습니다 — 이 대화창은 왼쪽에 있는 *Report Organizer* 패널을 제외하면 인쇄 조판 설계자와 매우 비슷합니다:

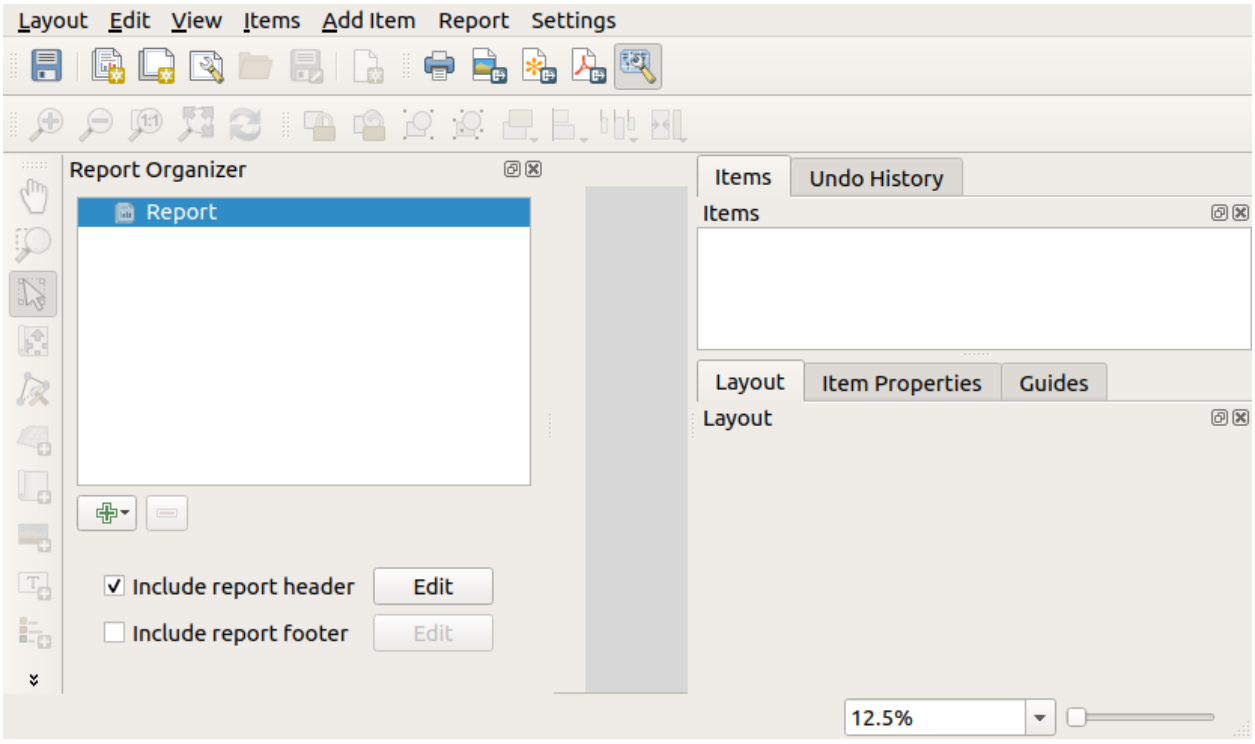
18.4.3 조판기 보고서 작업 공간

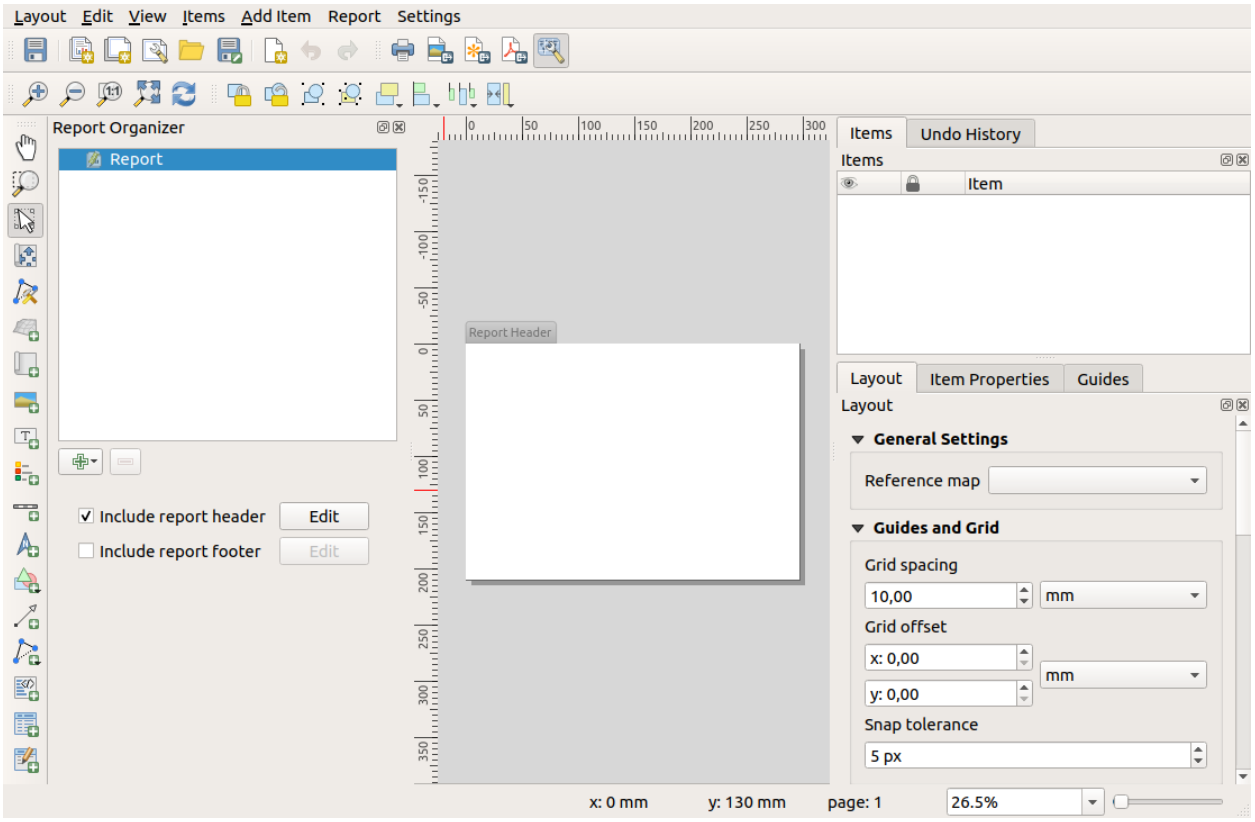
QGIS 보고서는 여러 개의 내포된 부분들로 이루어질 수 있습니다. 비어 있는 새 보고서는 처음에 주요 보고서 부분만 가지고 있습니다. 이 보고서 부분을 위한 옵션은 *Include report header* 와 *Include report footer* 뿐입니다. 이 옵션들을 활성화하면, 헤더가 보고서의 첫 번째 페이지 (들) 로 포함되고, 푸터가 마지막 페이지 (들) 를 구성할 것입니다. (필요한 경우 보고서의 개별 부분들을 여러 페이지로 만들 수 있습니다.) 헤더를 활성화하고 (*Include report header*) 옆에 있는 *Edit* 버튼을 클릭하십시오:

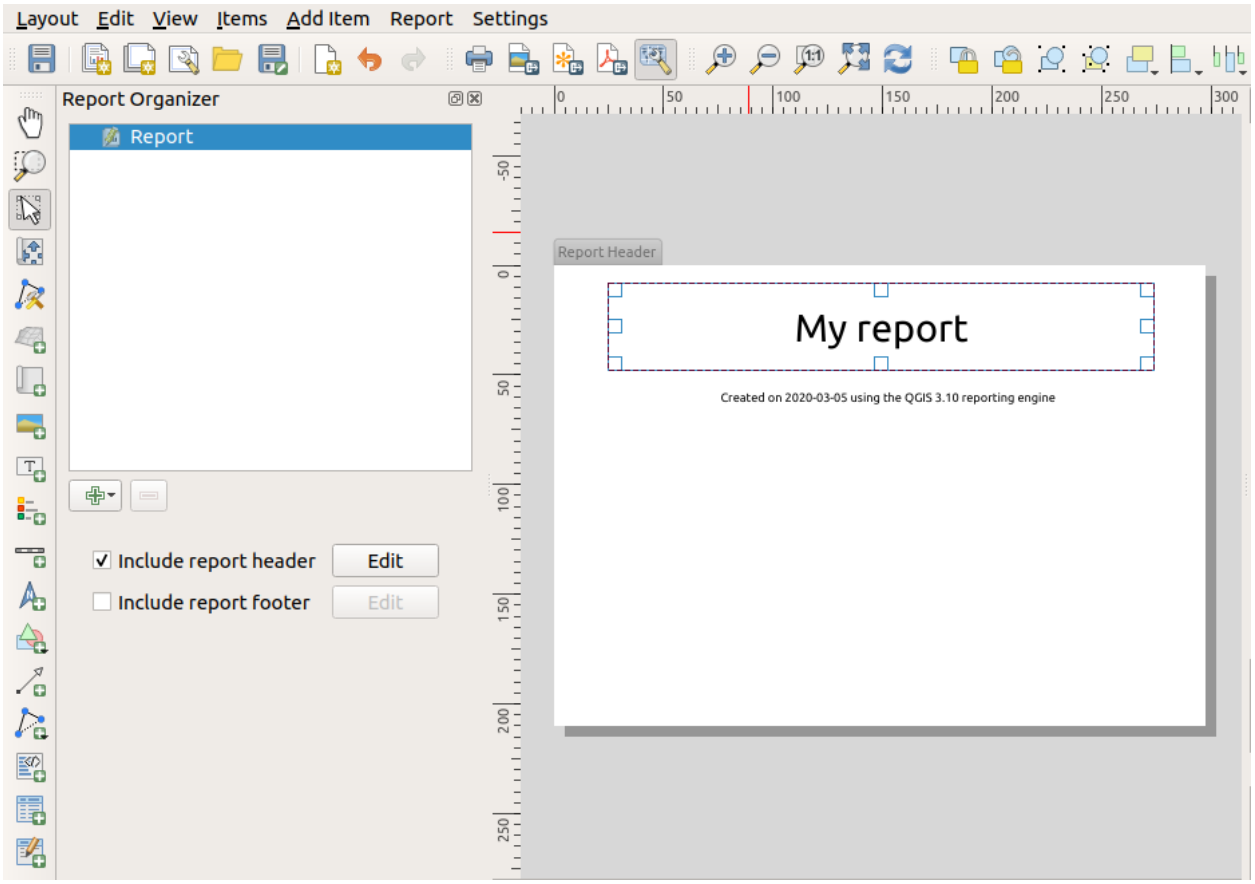
몇 가지 일이 일어납니다. 먼저, 설계자의 보고서 부분이 현재 편집중이라는 사실을 나타내는 편집 연필이 *Report Organizer* 의 *Report* 옆에 표시됩니다. 또 작은 *Report Header* 제목을 가진 새 페이지도 추가됩니다. 이 페이지는 기본적으로 랜드스케이프 방향이지만, 페이지를 오른쪽 클릭하고 *Page properties* 를 선택하면 이 속성을 (그리고 페이지의 다른 속성도) 변경할 수 있습니다. 이렇게 하면 페이지 *Item properties* 탭을 불러와서 페이지의 *Size*, *Width*, *Height* 는 물론 더 많은 속성을 설정할 수 있습니다.

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:

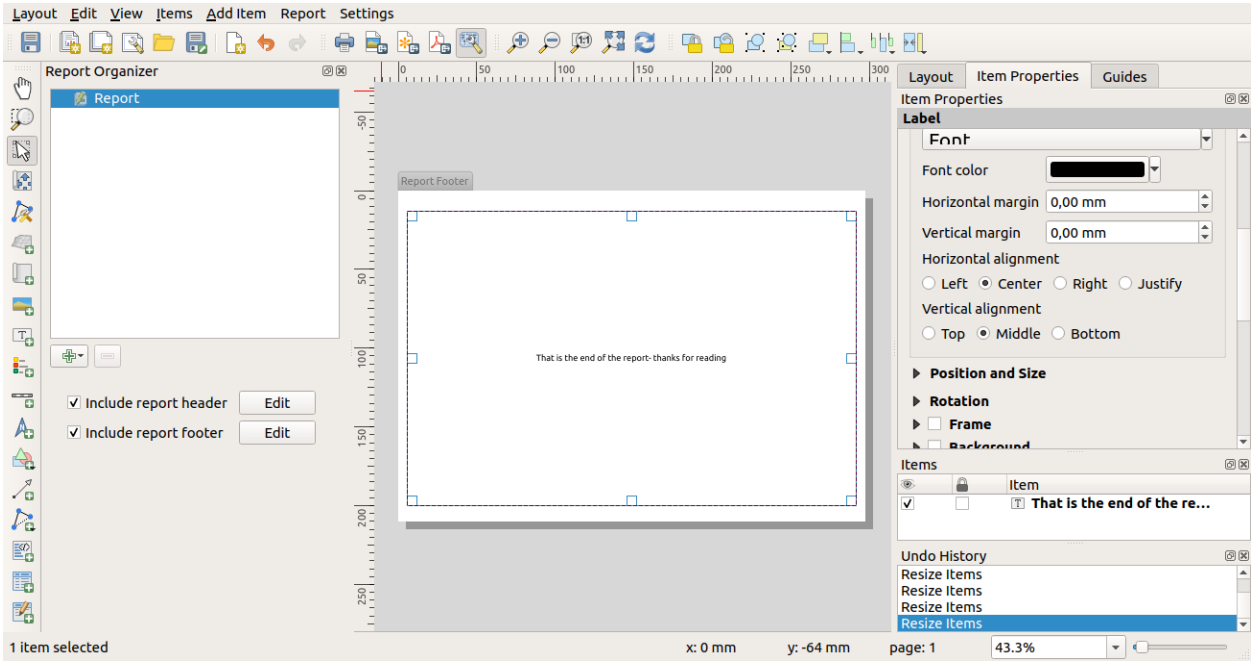

Layers
Airports
Ports
Populated Places
Other
Capitals
Admin Level 1
Ocean
Browser
Favorites
Spatial Bookmarks
Project Bookmarks
User Bookmarks
Project Home
Home
/
GeoPackage
SpatiaLite
PostGIS
MSSQL
DB2
WMS/WMTS
Valparaíso
Zacatecas
Tecuala
Villanueva
Santiago Ixcuintla
AGU
Tepic
Compostela
Puerto Vallarta
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%


Layout Edit View Items Add Item Report Settings
Report Organizer
Report
Include report header
Edit
Include report footer
Edit
Report Header
Items
Undo History
Items
Item
Layout
Item Properties
Guides
Layout
General Settings
Reference map
Guides and Grid
Grid spacing
10,00
mm
Grid offset
x: 0,00
y: 0,00
mm
Snap tolerance
5 px
x: 0 mm
y: 130 mm
page: 1
26.5%


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

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:

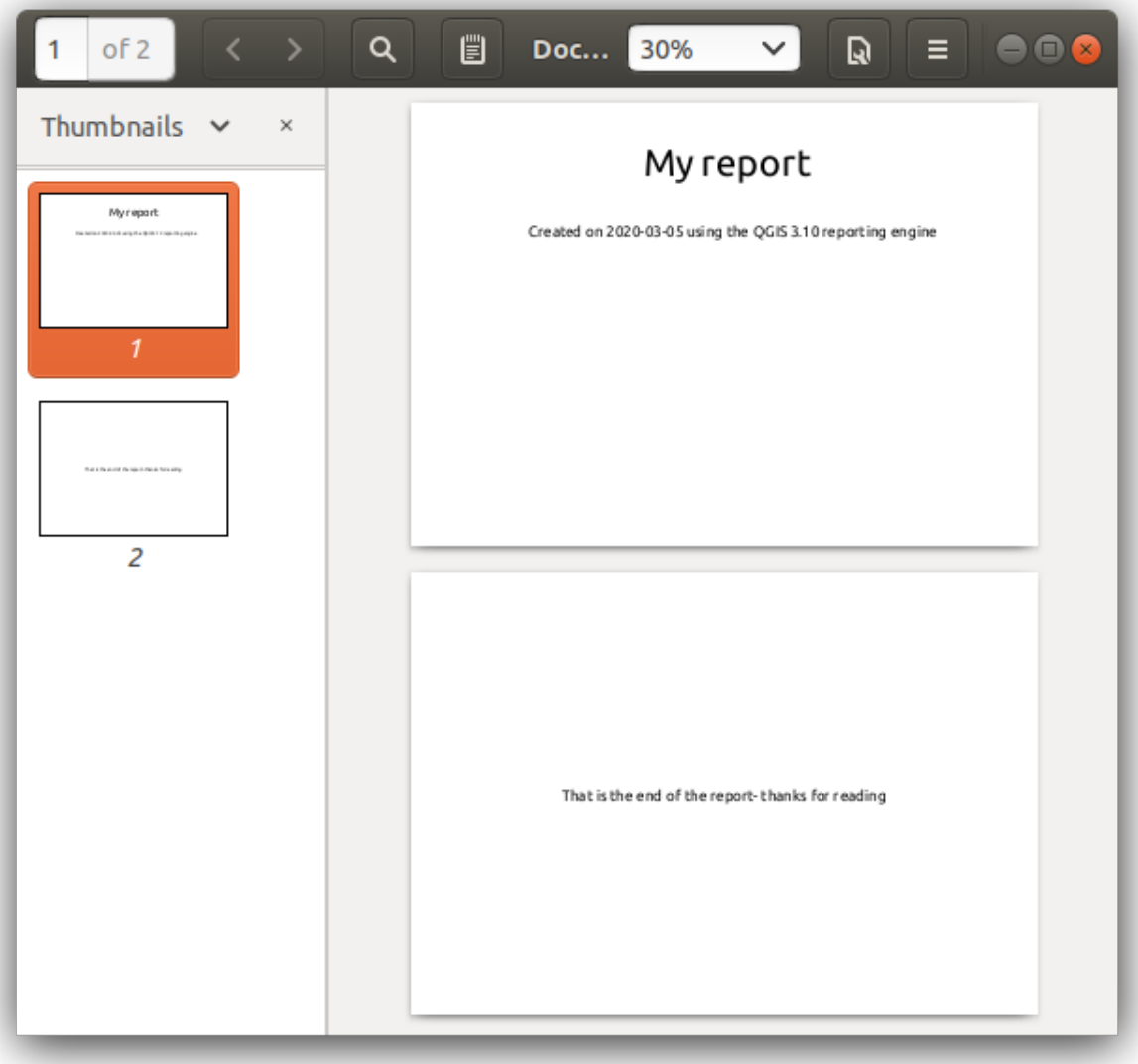
이 경우 *adm1name* 필드의 값을 사용해서'Admin Level 1'레이어의 모든 주 (☐) 를 반복 작업하도록 필드 그룹을 설정했습니다. 헤더 및 푸터를 포함시키는 기존 옵션은 물론 이 부분에 *body* 를 포함시키는 새 옵션도 있습니다. 이 옵션으로 본문을 추가하고 편집해보겠습니다:

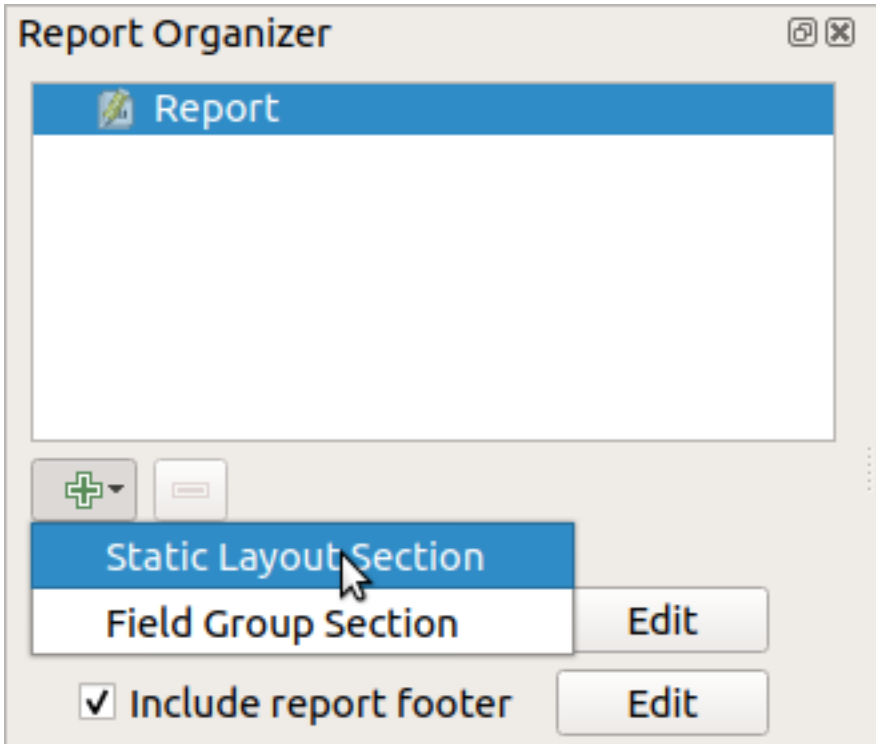
이제 맵과 주의 명칭을 표시하는 라벨로 본문을 구성합니다. 주의 명칭을 포함시키기 위해, *Add Item* ▷ *Add Label* 메뉴 옵션을 선택한 다음 *Insert or Edit an Expression*…을 통해 *Main Properties* 아래에서 텍스트를 데이터 정의했습니다.

다음 표현식을 사용했습니다 ('name'은 주의 명칭을 담고 있는'Admin Level 1'레이어 속성의 명칭입니다.):

```
[% "name" %]
```

맵이 현재 보고서 피처를 따르도록 설정합니다 (|checkbox| *Controlled by Report* 옵션을 체크해서 활성화합니다 — 지도책으로 제어 를 활성화하면 지도책이 현재 지도책 피처를 따르는 것과 동일합니다):


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 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:

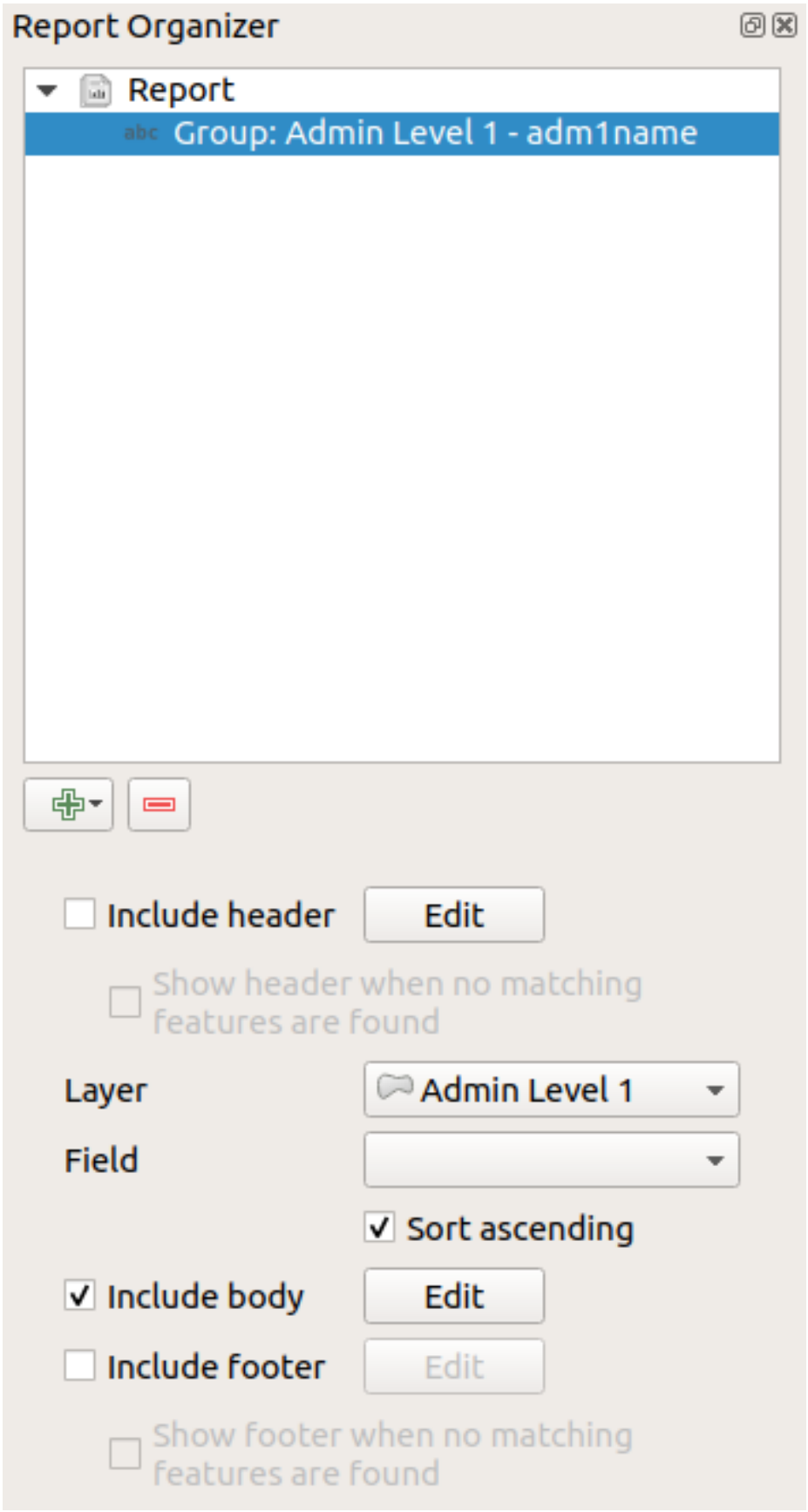
Field Group Section 의 피처들을 반복 작업할 때, 부모 그룹의 정의 필드 (이 경우 `adm1name`) 와 일치하도록 피처들을 필터링할 것입니다. 이때, 추가한 하위 부분이 *Populated Places* 레이어를 반복 작업하면서 일치하는 각 지역에 대한 본문 부분을 추가할 것입니다. 주목할 점은 *Populated Places* 레이어가 *adm1name* 부모 레이어의 정의 필드와 동일한 명칭을 가진 속성을 보유하고 있다는 것입니다. 즉 각 지역을 해당 지역이 있는 주에 태그한다는 뜻입니다. (운이 좋다면 사용자 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:

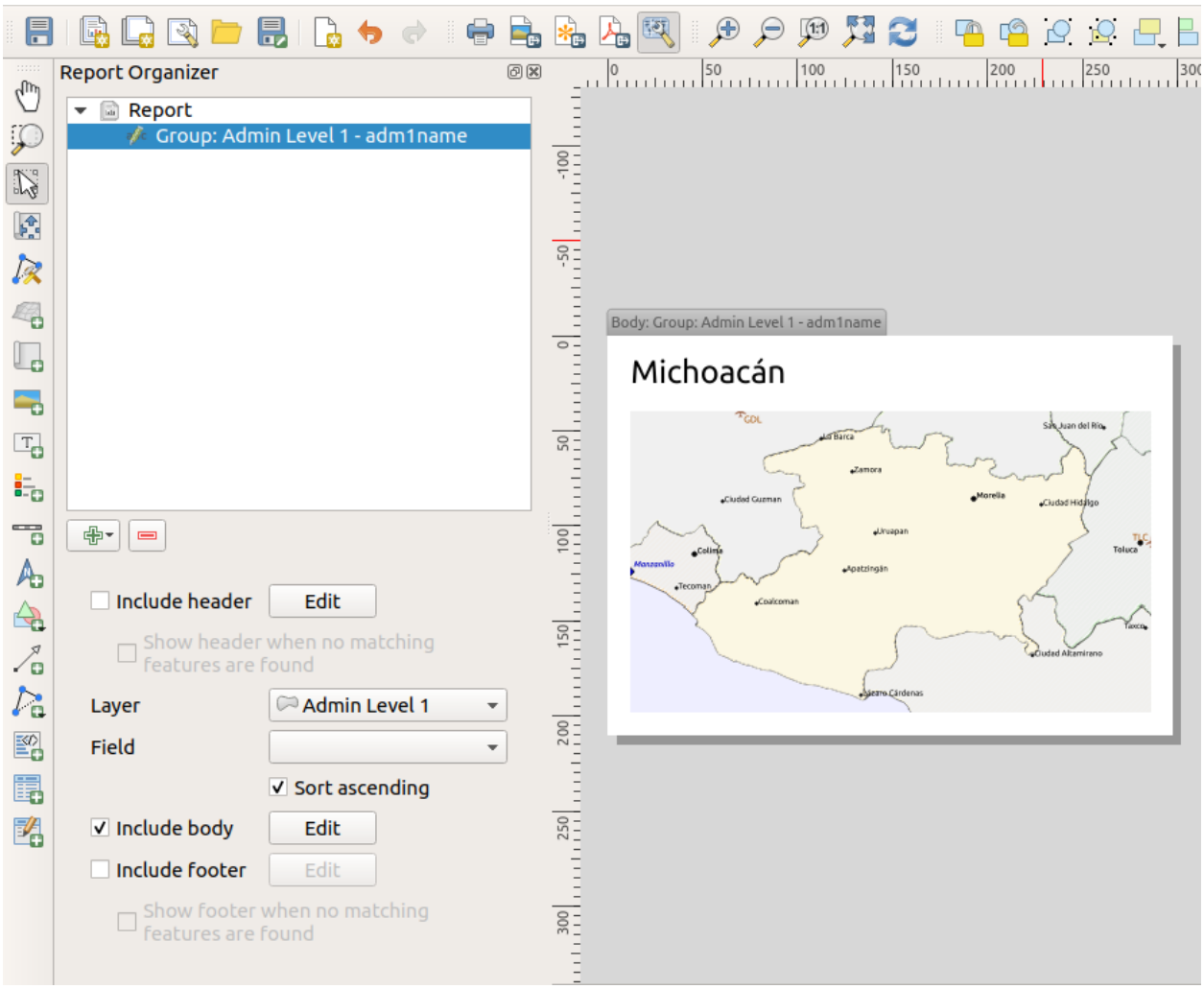
여기까지, 인구 밀집 지역 그룹에 대한 기초적인 본문을 지역의 맵과 일부 지역의 속성 테이블을 포함해서 생성했습니다. 이제 보고서는 보고서 헤더, 첫 번째 주에 대한 페이지, 해당 주에 있는 인구 밀집 지역 전체에 대한 페이지, 그 다음 주에 대한 페이지와 해당 주의 인구 밀집 지역 페이지, 그리고 마지막으로 보고서 푸터로 구성됐습니다. 인구 밀집 지역 그룹에 헤더를 추가하고자 했다면, 아래에 있는 그림과 같이 각 주의 인구 밀집 지역 페이지들 앞에 삽입되었을 것입니다.

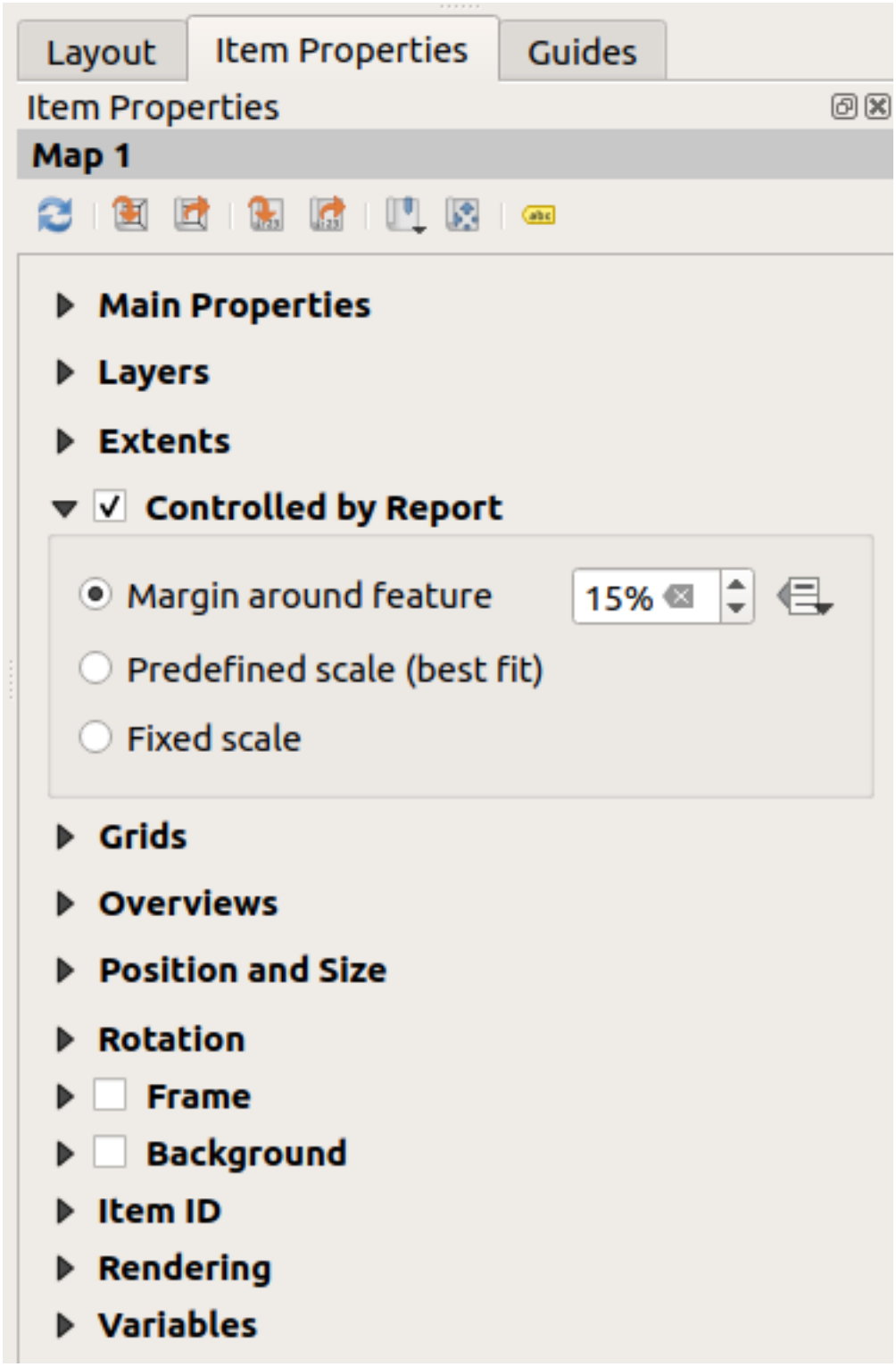
마찬가지로, 인구 밀집 지역 그룹에 푸터를 추가했다면 각 주의 마지막 지역 뒤에 삽입되었을 것입니다.

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 부모 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.


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

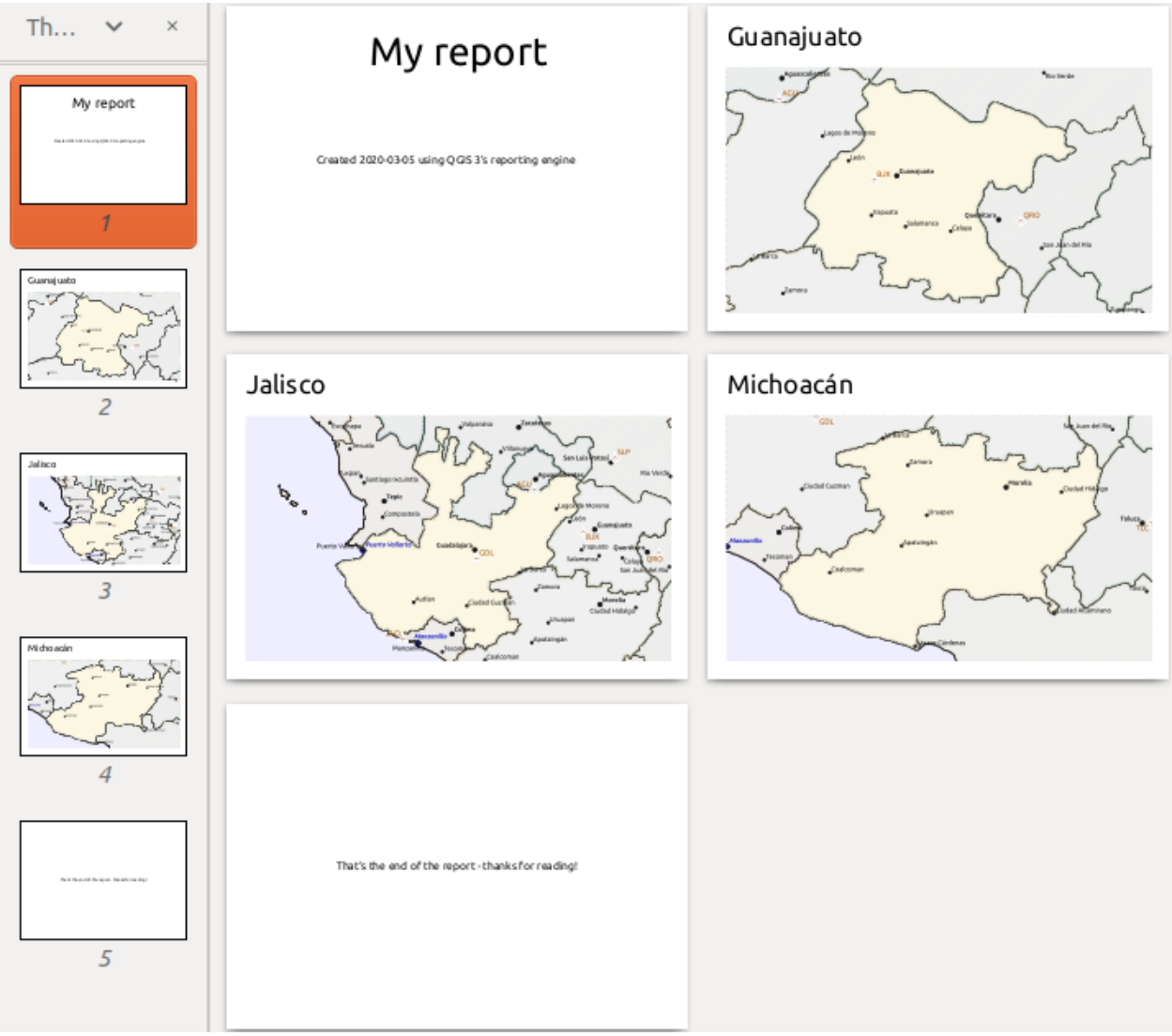
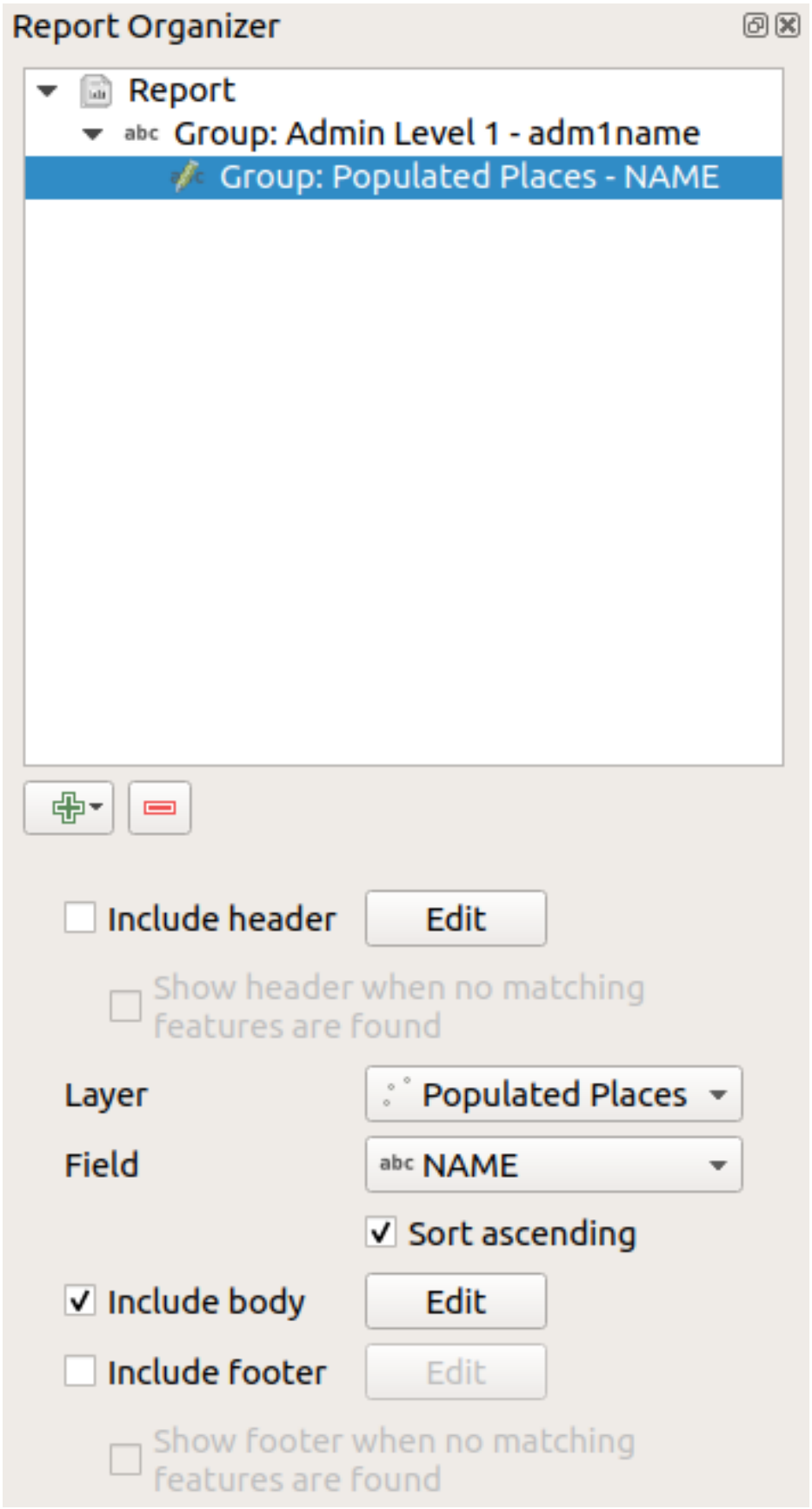
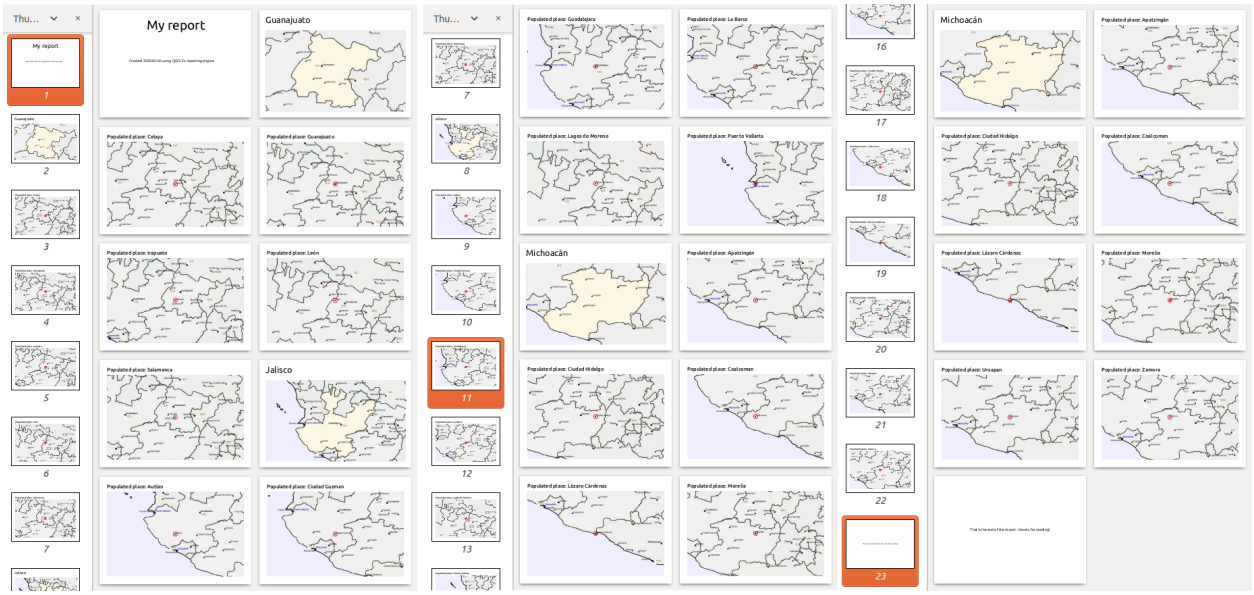


그림 18.64: 보고서 헤더, 각 주에 대한 페이지와 보고서 푸터


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



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.

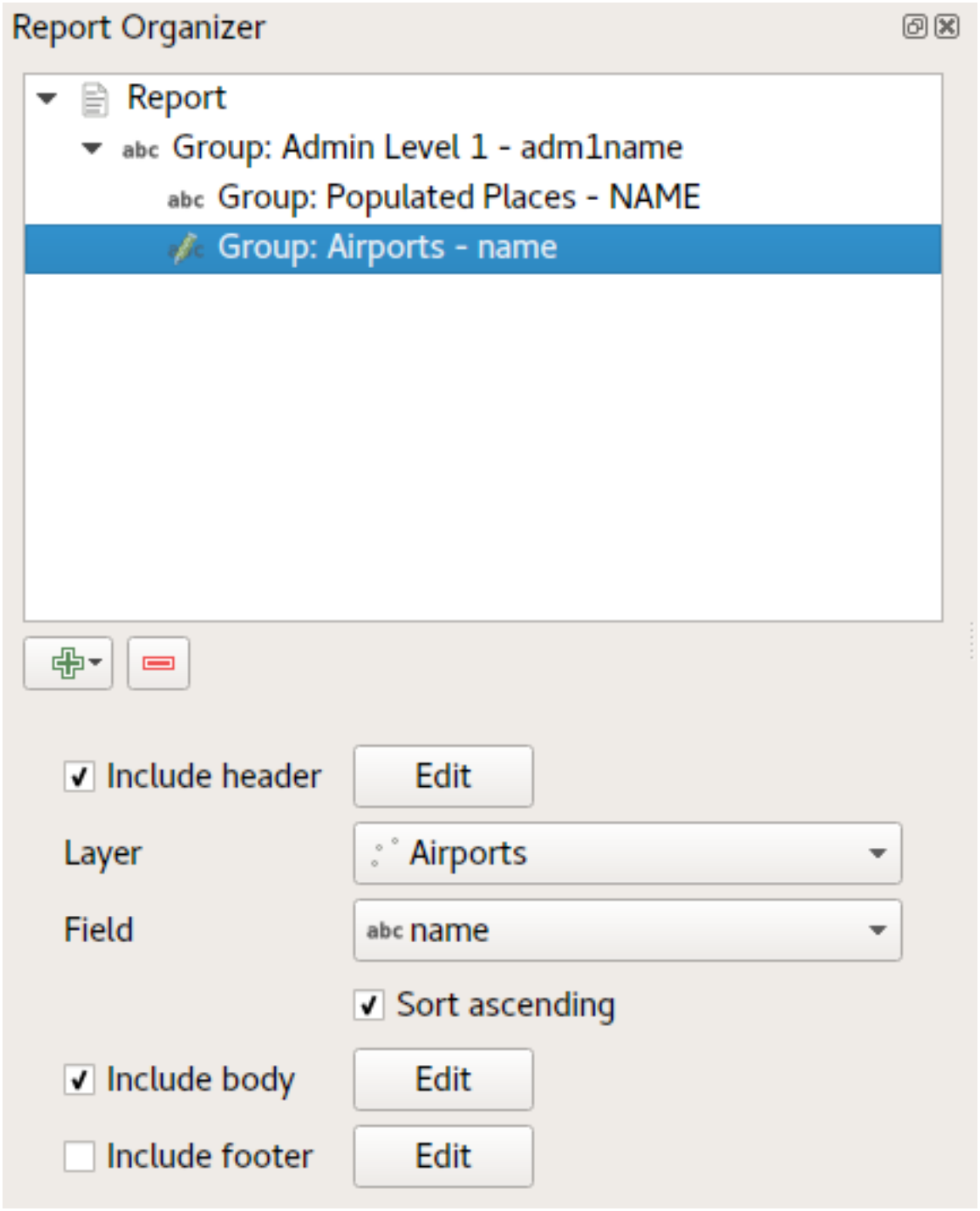
QGIS 는 보고서 이미지에 대해 절대 파일명에 의존합니다.

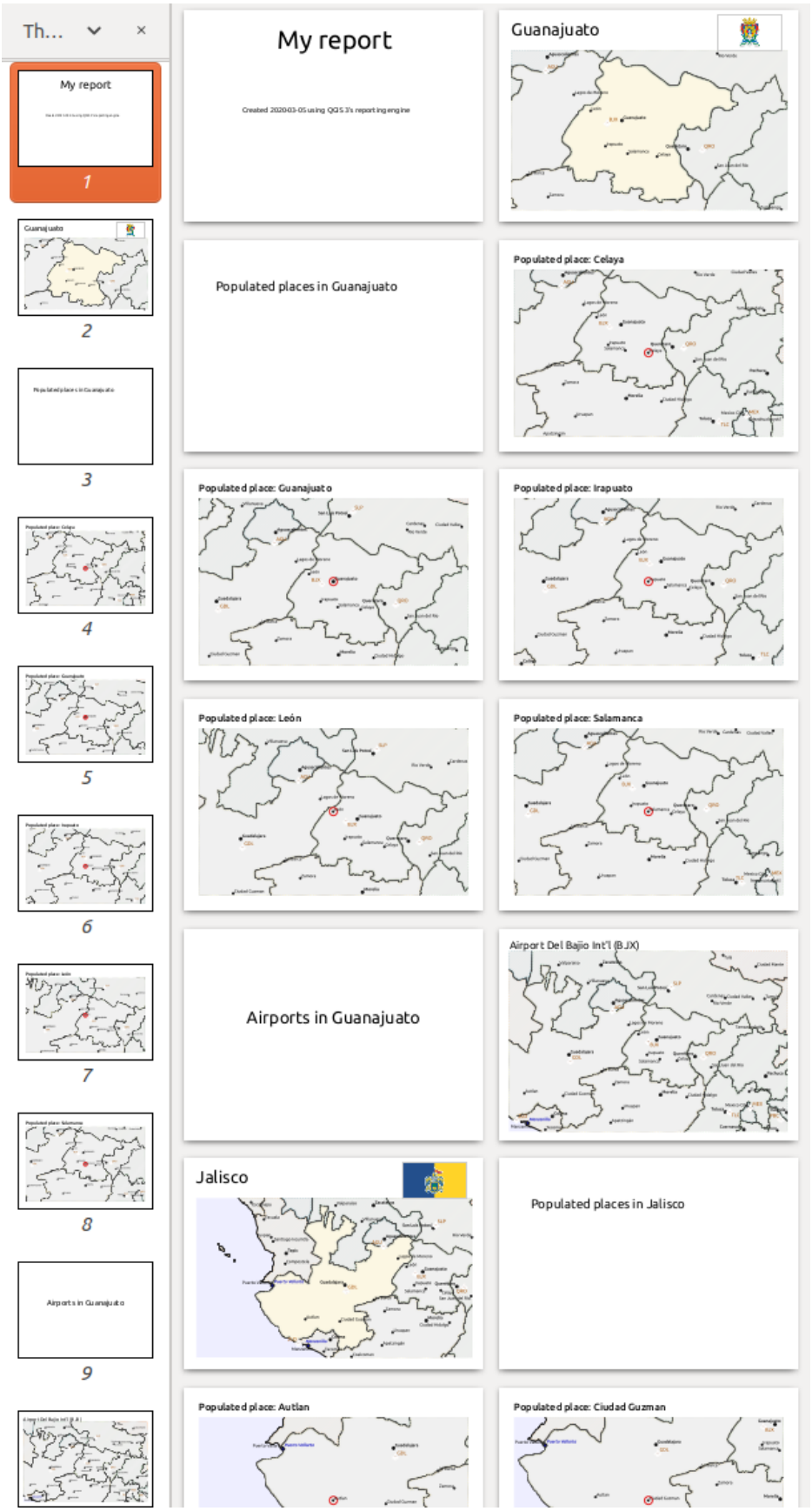
동적인 그림의 경우, 먼저 그룹의 본문 부분에 그림을 동일한 방법으로 추가하십시오. 그림 *Item properties* 패널에서 Data defined override 버튼으로 *Image Source* 를 설정한 다음 이미지의 절대 경로를 담고 있는 속성을 선택하거나, 또는 *Edit*…버튼을 클릭한 다음 이미지의 절대 경로를 생성하는 표현식을 입력하십시오.

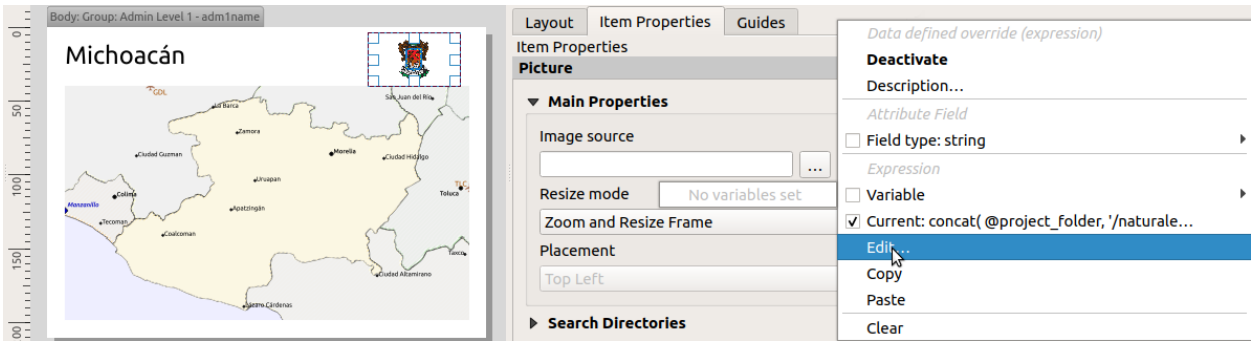
다음 예시 표현식은 프로젝트 파일이 위치한 디렉터리와 (`@project_path`) 파일명 생성 기반이 되는 속성을 (`adm1name`) 사용해서 그림을 가리키는 절대 경로를 지정하기 위해 문자열을 연결합니다. (이 경우 `adm1name` 속성의 문자열을 대문자로 변환한 다음 그 뒤에 `'_flag.png'` 를 붙여서 파일명을 생성합니다.)

```
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:

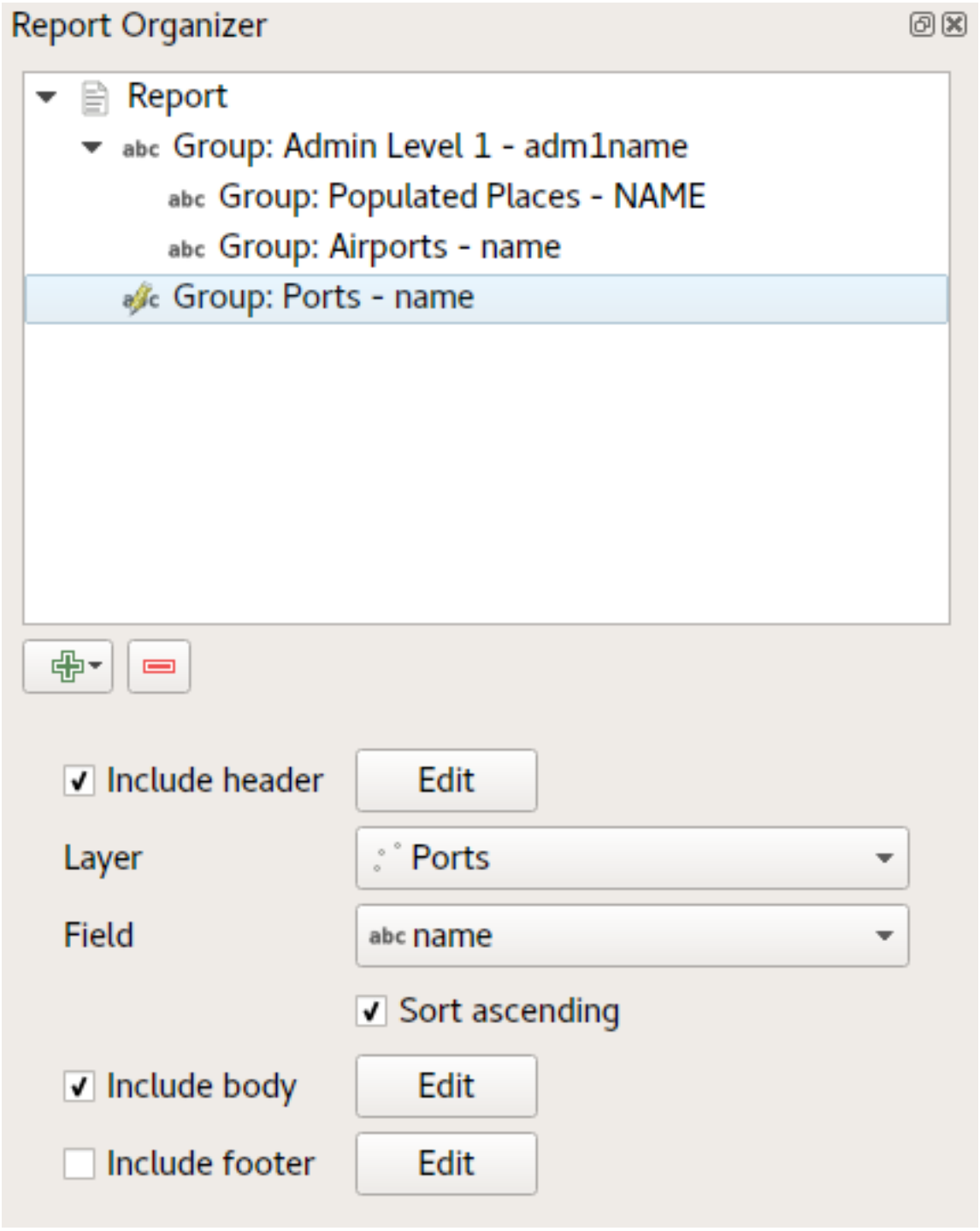
이렇게 내보내면 보고서의 마지막 부분에 다음과 같이 산출됩니다:

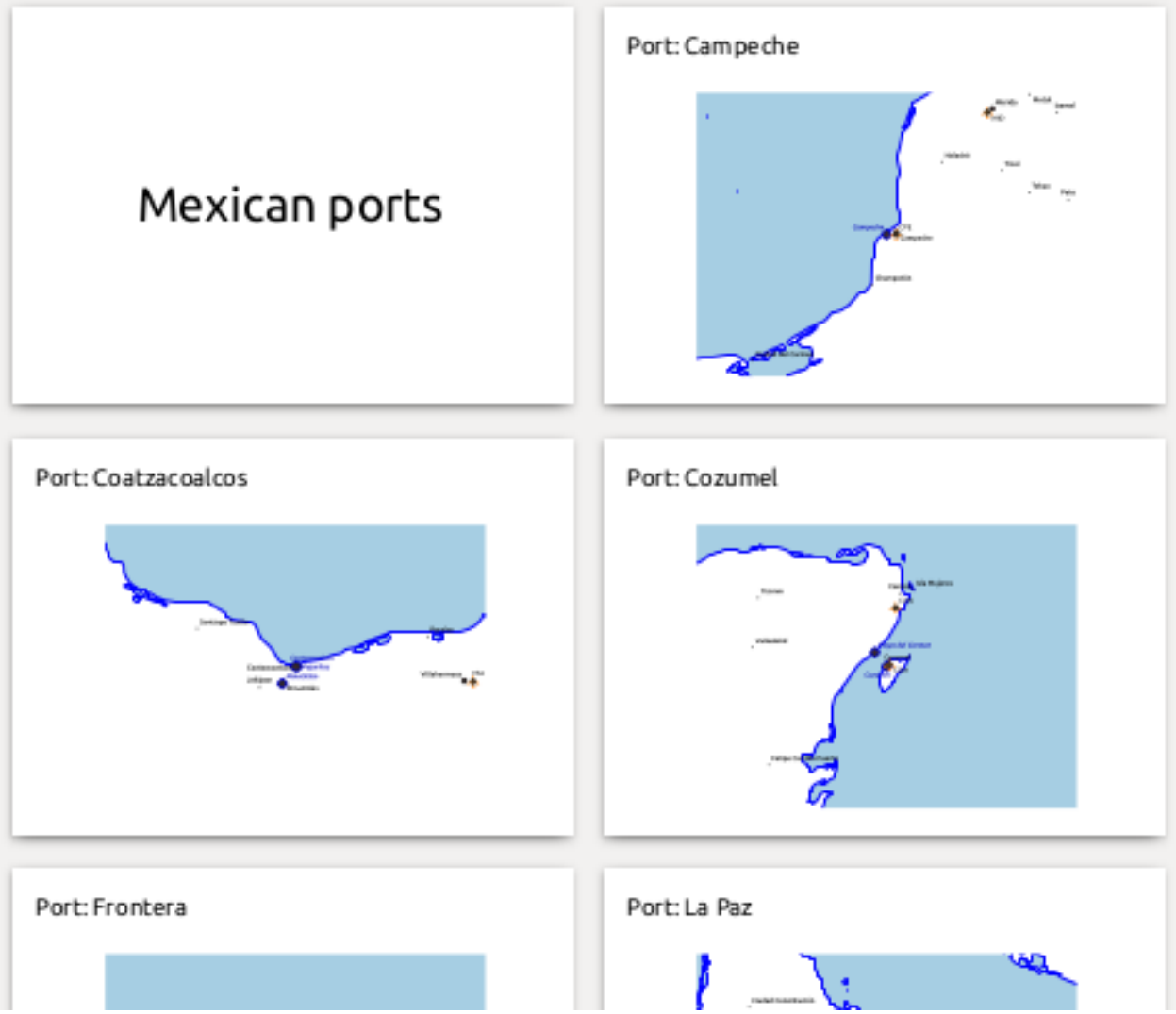
18.4.4 내보내기 설정

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.opengeospatial.org/ 사이트에서 더 자세한 정보를 찾아볼 수 있습니다.

QGIS 가 지원하는 중요한 OGC 사양은 다음과 같습니다:

- **WMS** –Web Map Service (*WMS/WMTS* 클라이언트)
- **WMTS** –Web Map Tile Service (*WMS/WMTS* 클라이언트)
- **WFS** –Web Feature Service (*WFS* 및 *WFS-T* 클라이언트)
- **WFS-T** –Web Feature Service - Transactional (*WFS* 및 *WFS-T* 클라이언트)
- **WCS** –Web Coverage Service (*WCS* 클라이언트)
- **WPS** –Web Processing Service
- **CSW** –Catalog Service for the Web
- **SFS** –Simple Features for SQL (*PostGIS* 레이어)
- **GML** –Geography Markup Language

OGC 서비스는 서로 다른 GIS 구현 및 데이터소스 사이에 지리공간 데이터를 교환하기 위해 점점 더 많이 이용되고 있습니다. QGIS 는 앞에서 소개한 사양들을 클라이언트로서, **SFS** 로서 (PostgreSQL/PostGIS 데이터 제공자의 지원을 통해, *PostGIS* 레이어 참조) 처리할 수 있습니다.

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 *Add Layer(s) from WMS(T) Server* 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:

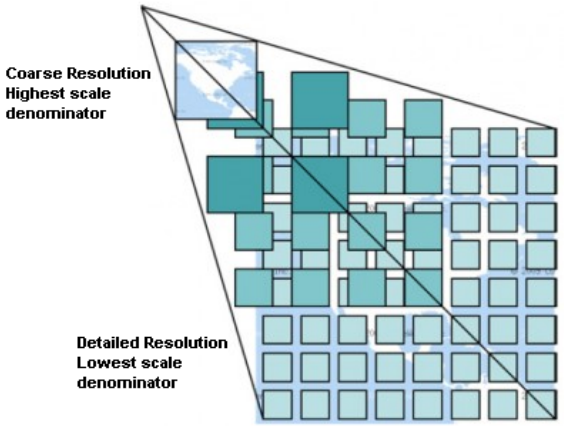


그림 19.1: WMTS 타일셋의 개념

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 예시가 바로 https://maps.wien.gv.at/basemap/1.0.0/WMTSCapabilities.xml 입니다.

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 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:

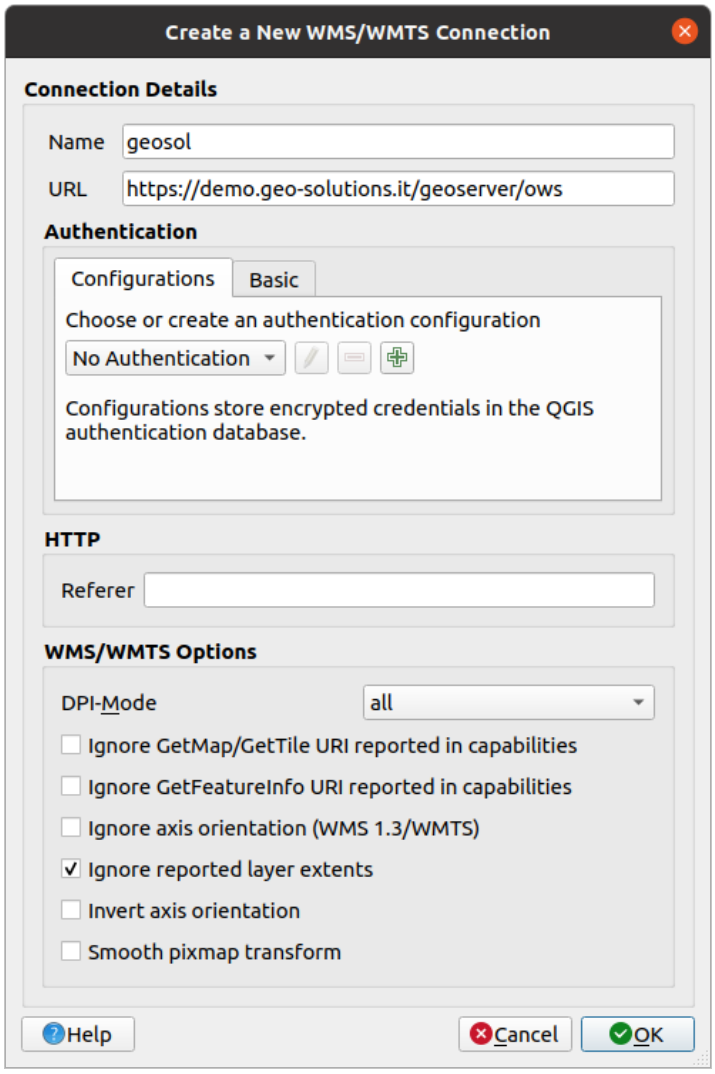


그림 19.2: WMS 서버에 연결 생성하기

- *Name*: 해당 연결을 위한 명칭입니다. 사용자가 다른 WMS 서버와 구별할 수 있도록, *Server Connections* 드롭다운 목록에 이 명칭을 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 를 참조하세요.

- HTTP *Referer*
- *DPI-Mode*: **all**, **off**, **QGIS**, **UMN** 및 **GeoServer** 옵션 가운데 선택할 수 있습니다.
- ☐ *Ignore GetMap/GetTile 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 axis orientation (WMS 1.3/WMTS)*
- ☐ *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.
- ☐ *Invert axis orientation*
- ☐ *Smooth pixmap transformation*

4. *OK* 를 클릭하십시오.

새 WMS 서버 연결을 생성하고 나면, 향후 QGIS 세션을 위해 새 연결을 보존합니다.

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:

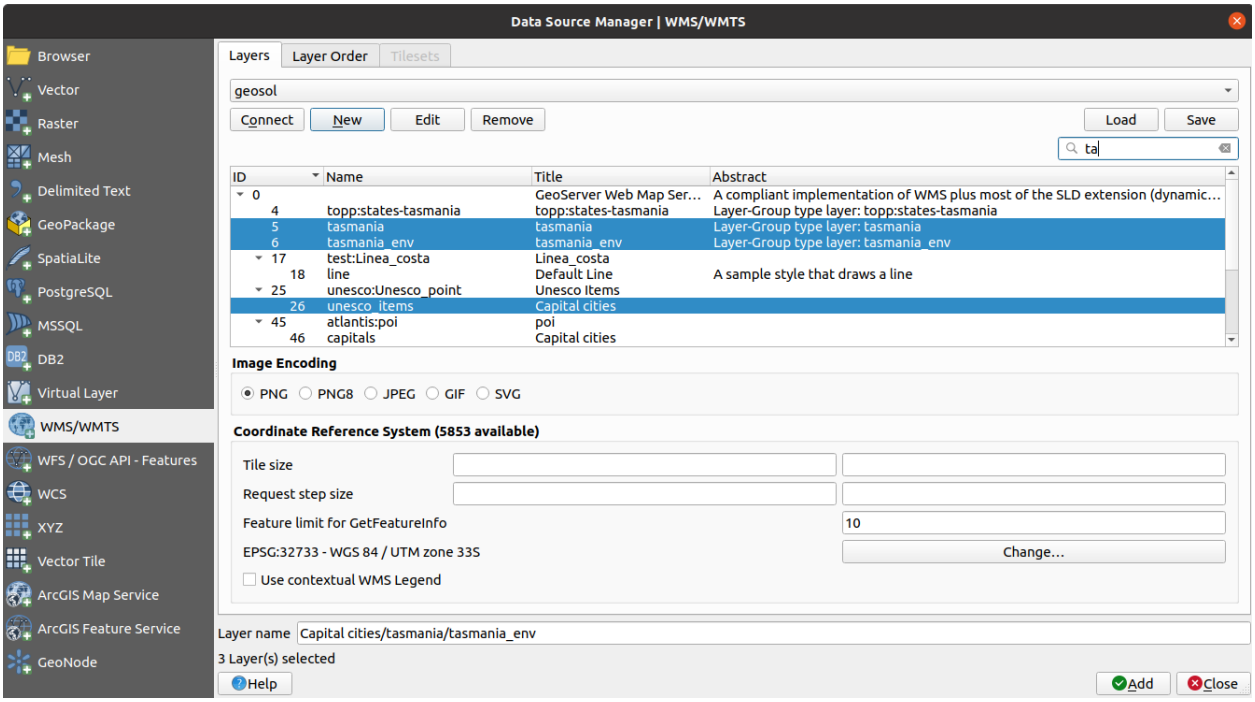


그림 19.3: 사용할 수 있는 레이어를 필터링해서 WMS 서버를 추가하는 대화창

- *ID*
- *Name*
- *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*

- *Feature limit for GetFeatureInfo* 는 서버에서 가져올 GetFeatureInfo 의 최대 개수를 정의할 수 있습니다.
- 목록에서 WMS 를 선택했다면, 맵 서버가 제공한 기본 투영체를 가진 필드가 나타납니다. *Change…*버튼을 클릭하면 WMS 의 기본 투영체를 WMS 서버가 지원하는 다른 좌표계로 변경할 수 있습니다.
- 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.

대화창의 하단에 있는 *Layer name* 텍스트 필드가 선택한 항목의 *Title* 을 표시합니다. 사용자 마음대로 이 제목을 변경할 수 있습니다. *Add* 버튼을 눌러 레이어 (들) 를 QGIS 로 불러오면 *Layers* 패널에 이 이름이 나타날 것입니다.

한 번에 여러 레이어를 선택할 수 있지만, 레이어 하나 당 이미지 스타일 하나만 선택할 수 있습니다. 여러 레이어를 선택한 경우, WMS 서버에서 레이어들을 하나로 결합한 다음, 한 번에 단일 레이어를 QGIS 로 전송합니다. 기본 이름은 원래 레이어 제목을 슬래시 (*/*) 로 구분한 목록입니다.

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.

팁: WMS 레이어 투명도

WMS 이미지의 투명도를 얼마나 활용할 수 있느냐는 어떤 이미지 인코딩을 사용했느냐에 따라 달라집니다. PNG 및 GIF 포맷은 투명도를 지원하는 반면, JPEG 은 지원하지 않습니다.

좌표계

좌표계란 QGIS 투영체를 뜻하는 OGC 용어입니다.

WMS 서버의 기능에 따라, 각 WMS 레이어를 여러 좌표계로 표현할 수 있습니다.

좌표계를 선택하려면, *Change…*버튼을 누르십시오. 그림 10.3 그림과 비슷한 대화창이 열릴 것입니다. WMS 버전의 대화창이 다른 점이라면 WMS 서버가 지원하는 좌표계만 표시될 것이라는 점이겠지요.

19.1.5 타일셋

WMTS(캐시된 WMS) 서비스를 다음과 같이 사용하는 경우

```
https://opencache.statkart.no/gatekeeper/gk/gk.open_wmts?\
  service=WMTS&request=GetCapabilities
```

서버가 제공한 *Tilesets* 탭을 통해 타일셋을 탐색할 수 있습니다. 타일 크기, 포맷, 지원 좌표계와 같은 추가 정보가 이 목록 테이블에 정리돼 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 send 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"
```

속성 살펴보기

WMS 서버를 추가하고 나면, 범례에서 서버를 오른쪽 클릭하고 *Properties* 를 선택해서 서버의 속성을 살펴볼 수 있습니다.

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.

19.1.7 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.

19.1.8 WMS 클라이언트 제약

QGIS 현재 버전이 사용 가능한 모든 WMS 클라이언트 기능을 포함하고 있지는 않습니다. 다음은 주목할 만한 일부 예외 사항에 대한 설명입니다.

WMS 레이어 설정 편집하기

 Add WMS layer 아이콘으로 WMS 레이어 추가 작업을 완료한 후에 설정을 변경할 방법은 없습니다. 설정을 변경하려면 레이어를 완전히 삭제한 다음 다시 시작해야 합니다.

인증이 필요한 WMS 서버

QGIS 현재 버전은 공개된 그리고 보안된 WMS 서비스를 지원하고 있습니다. 보안된 WMS 서버는 공개 인증서 (public authentication) 를 통해 접근할 수 있습니다. WMS 서버 추가 시 사용자가 (부가적인) 인증 정보를 추가할 수 있습니다. 자세한 내용은*WMS/WMTS 서버 선택하기* 를 참조하세요.

팁: 보안된 OGC 레이어에 접근하기

보안된 레이어에 기본 인증이 아닌 보안 방식으로 접근해야 하는 경우, 여러 인증 방식을 지원하고 있는 InteProxy 를 투명 프록시 모드로 사용할 수 있습니다. 더 자세한 정보는 https://inteproxy.wald.intevation.org/ 에 있는 InteProxy 사용자 설명서를 살펴보세요.

팁: QGIS WMS 맵서버

QGIS 는 1.7.0 버전부터 고유의 WMS 1.3.0 맵서버를 구현하고 있습니다. 자세한 내용은 QGIS-Server-manual 을 참조하세요.

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.

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 를 참조하세요.

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 WFS3(OGC API - Features) 를 편집 기능을 (WFS-T 를 통해) 포함해서 지원합니다.

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

게이트웨이 지오매틱스 (Gateway Geomatics) 의 WFS 서버를 예제로 삼아 레이어를 표시해보겠습니다. WFS 서버의 URL 은 https://demo.gatewaygeomatics.com/cgi-bin/wfs_gateway?REQUEST=GetCapabilities&VERSION=1.0.0&SERVICE=WFS 입니다.

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. `Gateway Geomatics` 를 명칭으로 입력하십시오.
5. (앞에서 언급한) URL 을 입력하십시오.

   참고: OGC API - Features(WFS3) 의 경우, 입력 URL 이 연결될 페이지 여야 합니다. 이때 연결될 페이지는 사용할 수 있는 모든 서비스 종단점까지 탐색할 수 있는 메인 페이지여야 합니다.

   ---

6. WFS 설정 대화창에서 다음과 같은 작업을 할 수 있습니다:
   - 서버의 WFS 버전을 알 수 있습니다. 표시되지 않는 경우, *Detect* 버튼을 클릭하면 자동으로 받아옵니다.
   - *maximum number of features*: `GetFetFeature` 요청 한 번으로 받아오는 피처의 최대 개수를 정의합니다. 비워놓는 경우, 제한을 두지 않습니다.
   - *Invert axis orientation*: 축의 방향을 반전시킵니다.
   - 그리고 WFS 버전에 따라:
     - *Ignore axis orientation (WFS 1.1/WFS 2.0)*: 버전 1.1, 2.0 인 경우 축의 방향을 무시하도록 강제할 수 있습니다.
     -  *Enable feature paging* 옵션을 활성화하면 *Page size* 란에 받아올 피처의 최대 개수를 지정할 수 있습니다. 비워놓는 경우, 서버 기본값을 적용합니다.

   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 를 참조하세요.

7. *OK* 를 클릭해서 연결을 생성하십시오.

사용자가 설정했을 수도 있는 어떤 프록시 설정이라도 적용된다는 사실을 기억하십시오.

이제 앞에서 설정한 연결로부터 WFS 레이어를 불러올 수 있습니다.

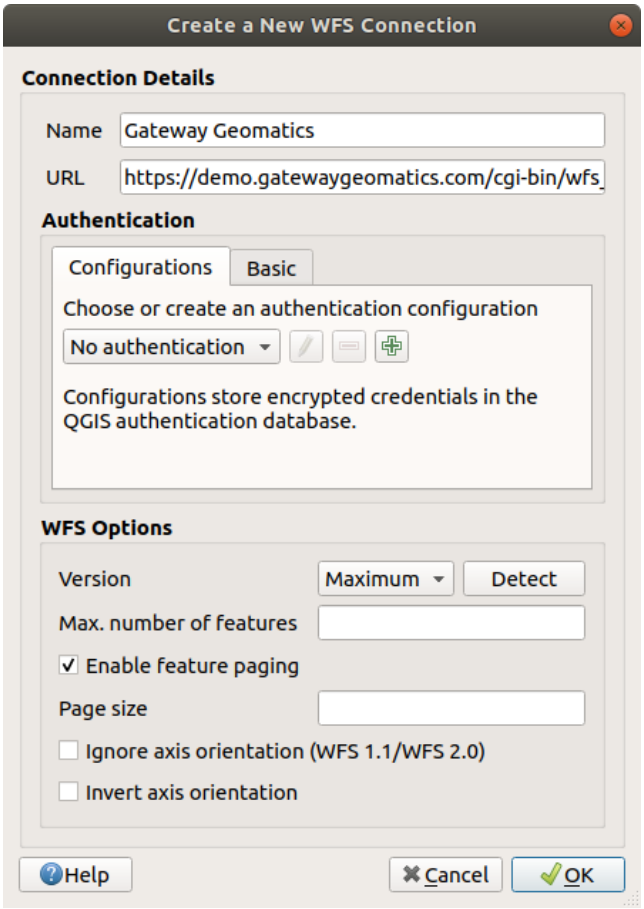


그림 19.4: WFS 서버와의 연결 생성하기

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:
   - *Use title for layer name*: 이 옵션을 체크하면 레이어의 *Name* 대신 *Layers* 패널에 있는 서버 상에서 정의된 레이어의 제목을 표시합니다.
   - *Only request features overlapping the view extent*: 이 옵션을 체크하면 뷰 범위와 겹치는 피처만 요청합니다.
   - *Change*: 이 버튼을 클릭하면 레이어의 좌표계를 변경할 수 있습니다.
   - *Build query* : 특정 피처를 받아오려면 이 버튼을 클릭해서, 또는 대상 레이어를 더블클릭해서 쿼리를 작성하십시오.
5. *Add* 버튼을 클릭하면 맵에 레이어를 추가합니다.

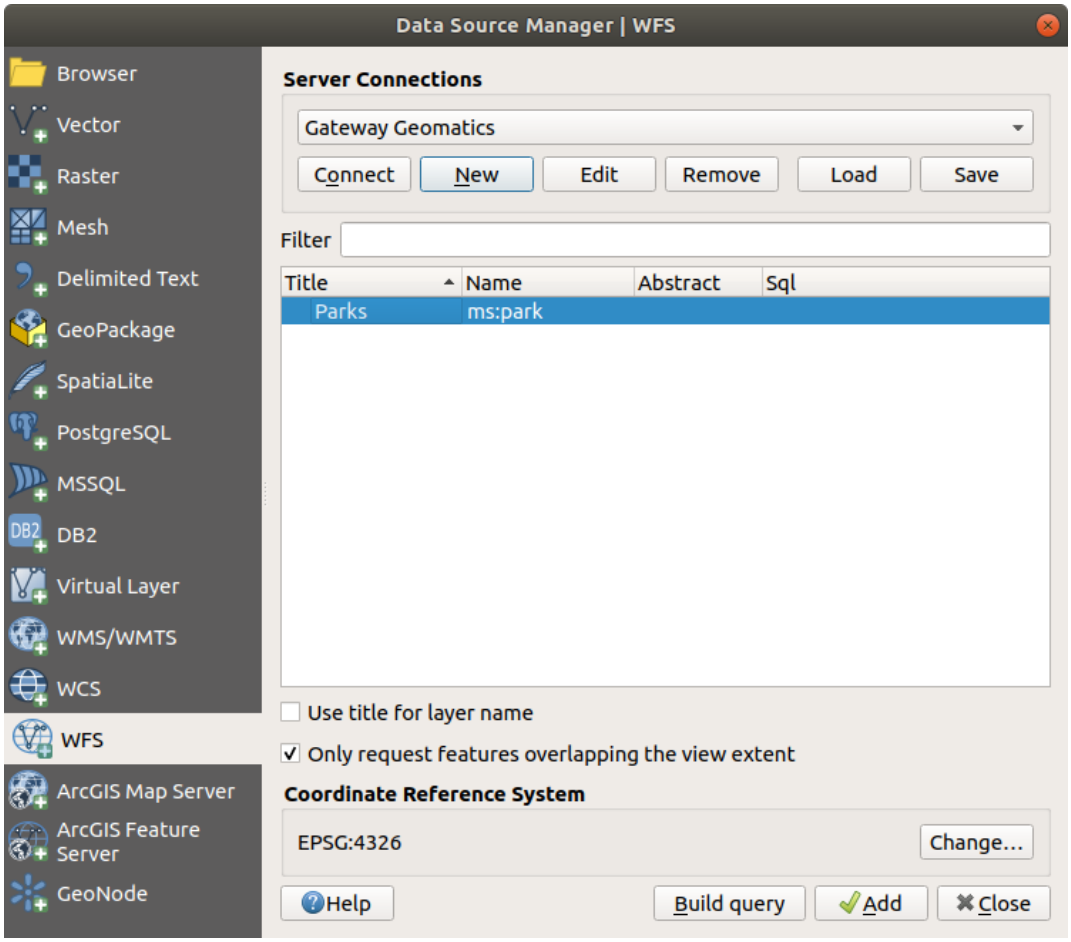


그림 19.5: WFS 레이어 추가하기

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.

참고: QGIS 는 백그라운드 다운로드, 점진적인 렌더링, 다운로드한 피처의 디스크 캐시 (on-disk caching), 그리고

자동 버전 탐지 등 여러 버전의 WFS 프로토콜을 지원합니다.

CHAPTER 20

GPS 데이터 작업

20.1 GPS 플러그인

20.1.1 GPS 란?

GPS(Global Positioning System) 는 GPS 수신기를 가지고 있다면 누구나 전 세계 어디에서든 자신의 위치를 정확하게 찾을 수 있는 위성 기반 시스템입니다. GPS 는 예를 들어 비행기, 배, 그리고 도보여행자들의 항법 보조 장치로 사용됩니다. GPS 수신기는 위성이 발신한 신호를 통해 경도, 위도, 그리고 (경우에 따라) 고도를 계산합니다. 수신기 대부분은 (**웨이포인트** 라는) 위치와, 계획된 **루트** 를 구성하는 일련의 위치와, 수신기의 시계열 이동을 추적하는 트랙로그 또는 **트랙** 을 저장할 수 있습니다. 웨이포인트, 루트, 그리고 트랙은 GPS 데이터의 세 가지 기본 피처 유형입니다. QGIS 는 웨이포인트를 포인트 레이어로, 루트 및 트랙을 라인스트링 레이어로 표시합니다.

참고: QGIS 는 GNSS 수신기도 지원하고 있습니다. 그러나 이 문서에서는 계속 GPS 라는 용어를 사용할 것입니다.

20.1.2 파일에서 GPS 데이터를 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, 먼저 플러그인을 불러와야 합니다. *Plugins* → *Plugin Manager*…메뉴 옵션을 선택하면 플러그인 관리자 대화창이 열립니다. *GPS Tools* 체크박스를 활성화하십시오. 이 플러그인을 불러오고 나면, 툴바와 *Layer* → *Create Layer* → 메뉴에 각각 다음 작은 휴대용 GPS 기기 아이콘이 나타날 것입니다:

- GPS Tools
- *Create new GPX Layer*

GPS 데이터를 작업하기 위해, QGIS 예시 데이터셋에 `qgis_sample_data/gps/national_monuments.gpx` 라는 예시 GPX 파일을 제공하고 있습니다. 예시 데이터에 관한 자세한 정보는예시 데이터 다운로드 를 참조하세요.

1. *Vector* → *GPS Tools* 메뉴 옵션을 선택하거나 툴바에 있는  GPS Tools 아이콘을 클릭한 다음 *Load GPX file* 탭을 선택하십시오. (그림 20.1 을 참조하세요.)
2. *Browse*…버튼을 눌러 `qgis_sample_data/gps/` 폴더를 찾은 다음, `national_monuments.gpx` GPX 파일을 선택하고 *Open* 버튼을 클릭하십시오.

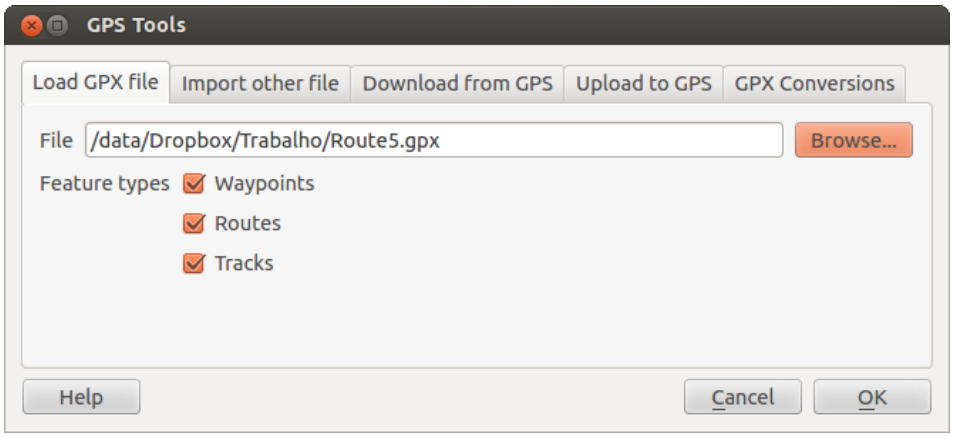


그림 20.1: *GPS* 도구 대화창

Browse…버튼으로 GPX 파일을 선택한 다음, 체크박스를 통해 해당 GPX 파일에서 불러오고자 하는 피처 유형을 선택하십시오. *OK* 버튼을 누르면 각 피처 유형을 개별 레이어로 불러올 것입니다. `national_monuments.gpx` 파일은 웨이포인트만 담고 있습니다.

참고: GPS 장치는 데이터를 서로 다른 좌표계로 저장할 수 있습니다. 사용자의 GPS 장치 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 사양 을 참조하세요.

20.1.3 GPSBabel

QGIS 가 GPX 파일을 사용하기 때문에, 다른 GPS 파일 포맷을 GPX 로 변환할 방법이 필요합니다. 자유 소프트웨어 GPSBabel 을 이용하면 많은 포맷들을 변환할 수 있습니다. GPSBabel 은 https://www.gpsbabel.org 에서 다운로드할 수 있습니다. 이 프로그램은 사용자 컴퓨터와 GPS 기기 간에 GPS 데이터를 전송할 수도 있습니다. QGIS 는 이런 작업에 GPSBabel 을 사용하므로, 이 프로그램을 설치하도록 권장합니다. 하지만, 단순히 GPX 파일에서 GPS 데이터를 불러오는 것만이 목적인 경우 이 프로그램이 필요하진 않습니다. GPSBabel 1.2.3 버전이 QGIS 가 공식적으로 지원하는 버전이지만, 이후 버전이라도 아무 문제없이 사용할 수 있을 것입니다.

20.1.4 GPS 데이터 가져오기

GPX 파일이 아닌 파일에서 GPS 데이터를 가져오려면, GPS 도구 대화창에 있는 *Import other file* 탭에서 불러올 수 있습니다. 이 탭에서 가져오고자 하는 파일 (및 파일 유형), 해당 파일에서 가져오고자 하는 피처 유형, 변환된 GPX 파일을 저장하고 싶은 위치, 그리고 새 레이어의 명칭을 지정합니다. 모든 GPS 데이터 포맷이 세 가지 피처 유형을 모두 지원하지는 않기 때문에, 여러 포맷의 경우 한두 가지 유형만 선택할 수 있을 것입니다.

20.1.5 GPS 데이터를 기기에서 다운로드

QGIS 는 GPSBabel 을 이용해서 GPS 기기로부터 새 벡터 레이어로 데이터를 직접 다운로드할 수 있습니다. 이 작업은 GPS 도구 대화창의 *Download from GPS* 탭에서 (그림 20.2 참조) 이루어집니다. 이 탭에서 GPS 기기의 유형, 기기가 연결된 포트 (또는 사용자의 GPS 기기가 지원하는 경우 USB), 다운로드하고자 하는 피처 유형, 데이터를 저장할 GPX 파일, 그리고 새 레이어의 명칭을 지정합니다.

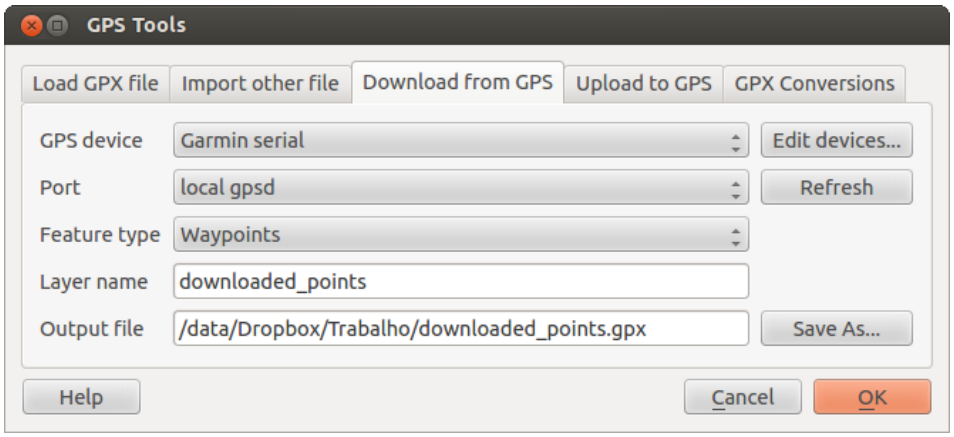


그림 20.2: 다운로드 도구

GPSBabel 은 *GPS device* 옵션에서 선택된 기기 유형을 이용해서 사용자의 GPS 기기와 통신을 시도합니다. 사용자 GPS 기기와 호환되는 유형이 없을 경우, 새 유형을 생성할 수 있습니다. (새 기기 유형 정의 를 참조하세요.)

Port 는 사용자의 OS 가 사용자 컴퓨터에 있는, GPS 기기가 연결된 물리적 포트를 참조하기 위해 사용하는 파일명 또는 기타 명칭일 수도 있습니다. USB 를 장비한 GPS 장치의 경우 그냥 USB 일 수도 있습니다.

- 리눅스 상에서는 `/dev/ttyS0` 또는 `/dev/ttyS1` 처럼 보일 것입니다.
- 윈도우 상에서는 `COM1` 또는 `COM2` 처럼 보일 것입니다.

OK 버튼을 누르면, 데이터가 기기에서 다운로드되어 QGIS 에 레이어로 나타날 것입니다.

20.1.6 GPS 데이터를 기기로 업로드

GPS 도구 대화창의 *Upload to GPS* 탭에서, QGIS 에 있는 벡터 레이어로부터 GPS 기기로 데이터를 직접 업로드할 수도 있습니다. 이 작업을 하려면, 그냥 업로드하고자 하는 레이어를 (GPX 레이어여야만 합니다) 선택한 다음, 사용자 GPS 기기 유형과 기기가 연결된 포트 (또는 USB) 를 지정하십시오. 다운로드 도구와 마찬가지로, 사용자 기기가 목록에 없을 경우 새 기기 유형을 생성할 수 있습니다.

QGIS 의 벡터 편집 능력과 결합한 경우 이 도구가 매우 유용합니다. QGIS 에 맵을 불러와서, 웨이포인트 및 루트를 생성한 다음, 사용자 GPS 기기로 업로드해서 사용할 수 있습니다.

20.1.7 새 기기 유형 정의

수많은 유형의 GPS 기기들이 있습니다. QGIS 개발자들이 모든 유형을 테스트할 수는 없으므로, *Download from GPS* 및 *Upload to GPS* 탭에 있는 기기 유형 목록에서 사용자의 GPS 기기와 호환되는 유형을 찾을 수 없는 경우, *Download from GPS* 및 *Upload to GPS* 탭에 있는 *Edit Devices* 버튼을 클릭하면 열리는 GPS 기기 편집기를 통해 사용자 자신의 기기를 정의해서 목록에 추가할 수 있습니다.

새 기기를 정의하려면, *New Device* 버튼을 눌러 기기 명칭과 사용자 기기용 다운로드 및 업로드 명령어를 입력한 다음, *Update Device* 버튼을 클릭하기만 하면 됩니다. 이 명칭은 — 어떤 문자열을 입력하더라도 — *Download from GPS* 및 *Upload to GPS* 탭에 있는 기기 유형 목록에 추가될 것입니다. 다운로드 명령어란 기기에서 데이터를 다운로드해서 GPX 파일로 저장하는 데 사용되는 명령어입니다. 대부분의 경우 GPSBabel 명령어일 테지만, GPX 파일을 생성할 수 있는 명령 줄 프로그램이라면 어떤 명령어라도 사용할 수 있습니다. QGIS 가 명령어를 실행하는 경우 `%type`, `%in`, and `%out` 키워드를 대체할 것입니다.

`%type` 은 웨이포인트를 다운로드하는 경우 `-w` 로, 루트를 다운로드하는 경우 `-r` 로, 트랙을 다운로드하는 경우 `-t` 로 대체될 것입니다. 이 문자열은 GPSBabel 에 어떤 피처 유형을 다운로드할지 알려주는 명령 줄 옵션입니다.

`%in` 은 *Download from GPS* 탭에서 선택한 포트명으로, `%out` 은 다운로드한 데이터를 저장할 GPX 파일명으로 대체될 것입니다. 따라서, `gpsbabel %type -i garmin -o gpx %in %out` 라는 다운로드 명령어로 ('Garmin serial'이라는 사전 정의된 기기 유형에 쓰이는 실제 다운로드 명령어입니다) 새 기기 유형을 생성한 다음 `/dev/ttyS0` 포트에서 웨이포인트 데이터를 다운로드해서 `output.gpx` 파일로 저장하는 경우, QGIS 키워드를 대체해서 `gpsbabel -w -i garmin -o gpx /dev/ttyS0 output.gpx` 라는 명령어를 실행할 것입니다.

업로드 명령어는 기기로 데이터를 업로드하는 데 사용되는 명령어입니다. 사용하는 키워드는 동일하지만, 업로드의 경우 `%in` 을 업로드하는 레이어를 위한 GPX 파일명으로, `%out` 을 포트명으로 대체합니다.

https://www.gpsbabel.org 웹사이트에서 GPSBabel 및 사용할 수 있는 명령 줄 옵션에 관해 더 많은 것을 알 수 있습니다.

새 기기 유형을 생성하고 나면, *Download from GPS* 및 *Upload to GPS* 탭에 있는 기기 유형 목록에 추가될 것입니다.

20.1.8 GPS 장치에서 포인트/트랙 다운로드

QGIS 는 앞에서 설명한 대로 GPSBabel 을 사용해서 프로젝트에 포인트/트랙을 직접 다운로드합니다. QGIS 는 처음부터 Garmin 기기에서 다운로드할 수 있는 사전 정의된 프로파일을 보유하고 있습니다. 안타깝게도 다른 프로파일을 생성하지 못 하게 하는 버그 #6318 이 있기 때문에, QGIS 가 GPS 도구를 통해 직접 다운로드할 수 있는 기기는 현재 Garmin USB 장치뿐입니다.

Garmin GPSMAP 60cs

MS 윈도우

https://www8.garmin.com/support/download_details.jsp?id=591 에서 Garmin USB 드라이버를 다운로드해서 설치하십시오.

장치를 연결하십시오. GPS 도구 대화창을 열고 `type=garmin serial` 및 `port=usb:` 로 지정하십시오. *Layer name* 및 *Output file* 텍스트 란에 명칭을 입력하십시오. 가끔 특정 폴더에 저장하는 데 문제가 있을 수도 있는데, `c:\temp` 같은 폴더를 사용하면 대부분 해결됩니다.

우분투/민트 GNU/리눅스

먼저, https://wiki.openstreetmap.org/wiki/USB_Garmin_on_GNU/Linux 에서 설명하고 있는 대로 기기의 권한 관련 문제를 해결해야 할 수도 있습니다. 이를 위해 다음 규칙을 담고 있는 `/etc/udev/rules.d/51-garmin.rules` 파일을 생성해볼 수 있습니다:

```
ATTRS{idVendor}=="091e", ATTRS{idProduct}=="0003", MODE="666"
```

그 다음, garmin_gps 커널 모듈을 불러오지 않았다는 것을 다음과 같이 확인해야 합니다:

```
rmmod garmin_gps
```

이제 GPS 도구를 사용할 수 있습니다. 안타깝게도 버그 #7182 때문에 QGIS 가 제대로 작동하기 전에 보통 몇 번 정도 멈출 겁니다.

BTGP-38KM DataLogger (블루투스만)

MS 윈도우

앞에서 언급한 버그 때문에 QGIS 안에서 데이터를 다운로드할 수 없기 때문에, 명령 줄 인터페이스에서 GPSBabel 을 사용하거나 또는 GPSBabel 의 인터페이스를 사용해야 합니다. 다음과 같은 명령어가 잘 작동합니다:

```
gpsbabel -t -i skytraq,baud=9600,initbaud=9600 -f COM9 -o gpx -F C:/GPX/aaa.gpx
```

우분투/민트 GNU/리눅스

윈도우에서와 동일한 명령어를 (또는 GPSBabel GUI 를 사용하는 경우 동일한 설정을) 사용하십시오. 리눅스 상에서 다음과 같은 메시지가 반환되는 일이 자주 있을 수도 있습니다:

```
skytraq: Too many read errors on serial port
```

이때 그냥 기기를 껐다가 켠 다음 다시 시도해보면 됩니다.

BlueMax GPS-4044 DataLogger (블루투스 및 USB 모두)

MS 윈도우

참고: 윈도우 7 에서 이 기기를 이용하려면 먼저 드라이버를 설치해야 합니다. http://blu-max.com/downloads.html 페이지에서 알맞은 드라이버를 다운로드하십시오.

GPSBabel 을 통해 다운로드하는 경우, USB 및 블루투스 모두 항상 다음과 같은 오류를 반환합니다:

```
gpsbabel -t -i mtk -f COM12 -o gpx -F C:/temp/test.gpx
mtk_logger: Can't create temporary file data.bin
Error running gpsbabel: Process exited unsuccessfully with code 1
```

우분투/민트 GNU/리눅스

USB 연결

USB 케이블을 연결한 다음, dmesg 명령어로 어떤-예를 들면 /dev/ttyACM3 같은-포트를 사용하고 있는지 알아내십시오. 그리고 평소와 같이 CLI 또는 GUI 에서 GPSBabel 을 사용하면 됩니다.

```
gpsbabel -t -i mtk -f /dev/ttyACM3 -o gpx -F /home/user/bluemax.gpx
```

블루투스 연결

블루투스 기기 관리자 (Blueman Device Manager) 를 사용, 기기와 페어링해서 시스템 포트를 통해 사용할 수 있도록 한 다음, GPSBabel 을 실행하십시오.

```
gpsbabel -t -i mtk -f /dev/rfcomm0 -o gpx -F /home/user/bluemax_bt.gpx
```

20.2 실시간 GPS 추적

QGIS 에서 실시간 GPS 추적 도구를 활성화하려면, *View* ▷ *Panels* ☑ *GPS Information Panel* 패널 체크박스를 체크하거나 `Ctrl+0` 조합키를 눌러야 합니다. 캔버스 좌측에 새 도킹 창이 열릴 것입니다.

이 GPS 추적 창에서 다음 3 개 화면을 볼 수 있습니다:

- GPS 위치 좌표를 표시하며 수동으로 꼭짓점 및 피처를 입력할 수 있는 인터페이스가 있습니다.
- 연결된 위성의 GPS 신호 강도를 표시합니다.
- GPS 옵션 화면입니다. (그림 20.5 참조)

GPS 수신기를 컴퓨터와 연결한 다음 (사용자 OS 가 해당 기기를 지원해야만 합니다) *Connect* 버튼을 누르면 QGIS 에 GPS 기기를 연결합니다. (이제 *Disconnect* 로 바뀐) 해당 버튼을 다시 클릭하면 사용자 컴퓨터에서 GPS 수신기를 연결 해제합니다. GNU/리눅스의 경우, 대부분의 GPS 수신기와의 연결을 지원하기 위해'gpsd'지원이 통합돼 있습니다. 따라서, QGIS 에 수신기를 연결하려면 먼저'gpsd'를 알맞게 환경 설정해야 합니다.

경고: 사용자의 위치를 캔버스에 기록하고 싶다면, 먼저 새 벡터 레이어를 생성한 다음 사용자의 트랙을 기록할 수 있게 편집 가능 상태로 전환해야 합니다.

20.2.1 위치 및 추가 속성

GPS 가 위성으로부터 신호를 받고 있다면, 사용자의 위치를 위도, 경도 그리고 고도로 — 추가 속성들과 함께 — 볼 수 있을 것입니다.

20.2.2 GPS 신호 강도

이 화면에서 신호를 받고 있는 위성의 신호 강도를 볼 수 있습니다.

20.2.3 GPS 옵션

연결에 문제가 생겼을 경우, 다음 가운데 하나로 전환할 수 있습니다:

- *Autodetect*
- *Internal*
- *Serial device*
- *gpsd* (사용자 GPS 가 연결돼 있는 호스트, 포트 및 기기를 지정)

Connect 버튼을 다시 클릭하면 GPS 수신기로 연결을 시작합니다.

편집 모드 상태라면 ☑ *Automatically save added features* 옵션을 활성화할 수 있습니다. 또는 ☑ *Automatically add points* 옵션을 활성화해서 맵 캔버스에 포인트를 지정한 크기와 색상으로 자동 추가할 수도 있습니다.

☑ *Cursor* 옵션을 활성화하면, 슬라이드 바 를 통해 캔버스 상에서 위치 표시 커서의 크기를 줄이거나 키울 수 있습니다.

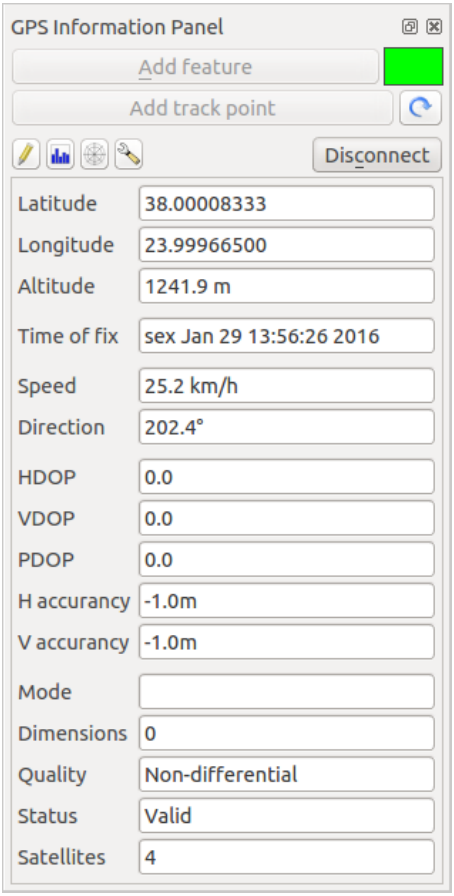


그림 20.3: GPS 위치 추적 및 추가 속성

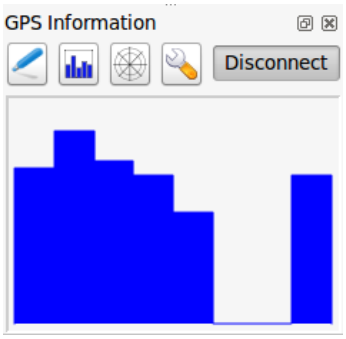


그림 20.4: GPS 추적 신호 강도

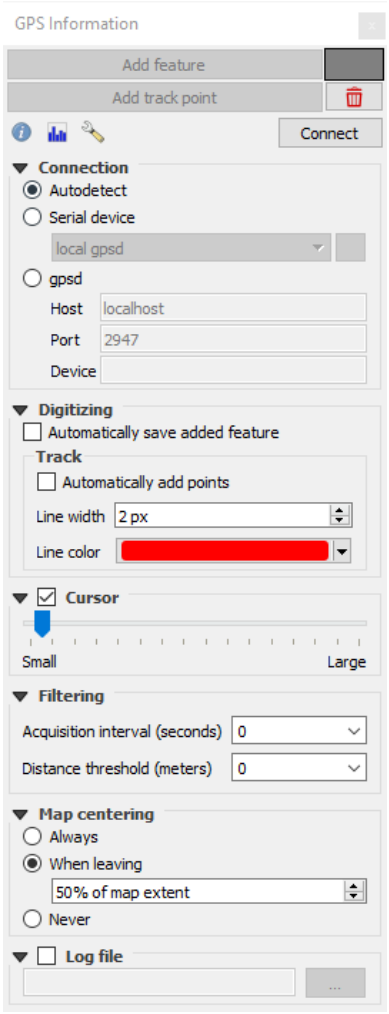


그림 20.5: GPS 추적 옵션 창

수신기가 정적 상태일 때도 커서를 활성화 상태로 유지하려면, *Acquisition interval (seconds)* 및 *Distance threshold (meters)* 파라미터도 설정하면 됩니다.

Map centering 옵션을 활성화하면 캔버스를 어떤 방식으로 업데이트할지 결정할 수 있습니다. 'always'를 선택하면 좌표가 기록될 때마다, 'when leaving'을 선택하면 기록되는 좌표가 캔버스 밖으로 나가는 시점에 맵 캔버스를 업데이트합니다. 또는'never'를 선택해서 맵 범위를 유지할 수도 있습니다.

마지막으로, *Log file* 옵션을 활성화하면 GPS 추적 관련 로그 메시지를 기록할 경로 및 파일명을 정의할 수 있습니다.

피처를 수동으로 설정하고 싶은 경우, Position 아이콘을 다시 클릭한 다음 *Add Feature* 또는 *Add Track Point* 버튼을 클릭해야 합니다.

20.2.4 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.

20.2.5 GPSMAP 60cs 사용법

MS 윈도우

GPSGate 라는 (공개 S/W 는 아니지만 자유 S/W 인) 미들웨어를 사용하면 이 GPS 기기를 가장 쉽게 사용할 수 있습니다.

프로그램을 구동해서 GPS 기기를 스캔하도록 하십시오. (USB 및 블루투스 둘 다 인식할 수 있습니다.) 그리고 QGIS 에서 *GPS Information Panel* 을 열고 GPS 옵션 화면에서 *Autodetect* 모드를 선택한 다음 *Connect* 버튼을 누르면 됩니다.

우분투/민트 GNU/리눅스

윈도우와 마찬가지로 미들웨어를 사용하는 것이 가장 쉬운 방법입니다. 이 경우에는 `gpsd` 를 다음과 같이 설치하십시오:

```
sudo apt install gpsd
```

그리고 `garmin_gps` 커널 모듈을 다음과 같이 불러오십시오:

```
sudo modprobe garmin_gps
```

그 다음 기기를 연결하고, `dmesg` 명령어로 실제 기기가 연결됐는지, 예를 들어 `/dev/ttyUSB0` 처럼 마운트됐는지 확인하십시오. 이제 다음과 같이 `gpsd` 를 구동할 준비가 끝났습니다:

```
gpsd /dev/ttyUSB0
```

마지막으로 QGIS 의 실시간 GPS 추적 도구와 연결하십시오.

20.2.6 BTGP-38KM DataLogger (블루투스만)

`gpsd` (리눅스) 또는 GPSGate (윈도우) 프로그램을 사용하는 것이 가장 쉬운 방법입니다.

20.2.7 BlueMax GPS-4044 DataLogger (블루투스 및 USB 둘 다)

MS 윈도우

GPS Information Panel 의 GPS 옵션 화면에서 *Autodetect* 모드를 선택하거나, 또는 올바른 포트를 지정하는 것만으로 – GPSGate 를 사용하거나 사용하지 않거나 – 블루투스 및 USB 연결 양쪽에서 모두 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

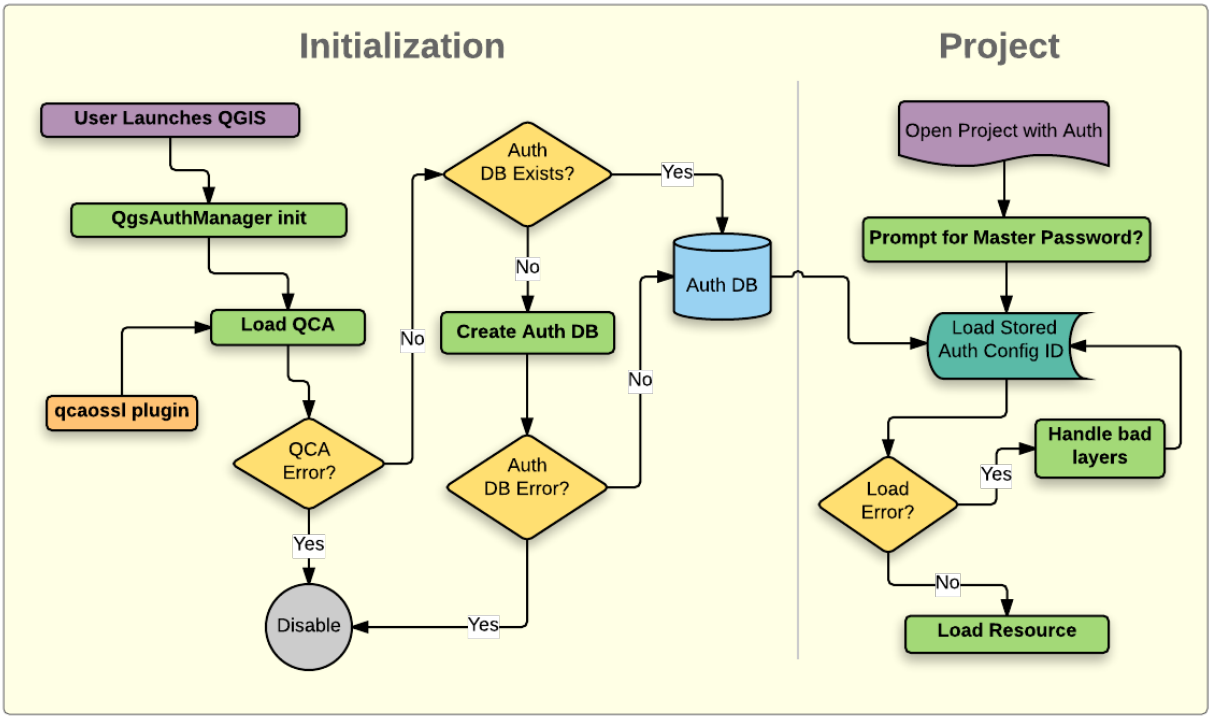


그림 21.1: 인증 시스템 개념도

21.1.1 인증 데이터베이스

새 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 항목으로 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 부모 디렉터리를 설정할 수 있습니다.

21.1.2 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.

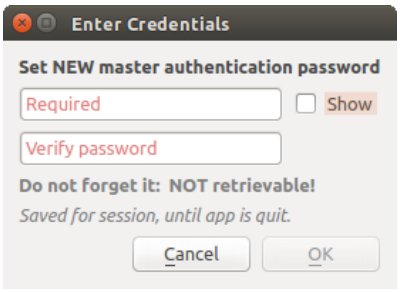


그림 21.2: 새 마스터 비밀번호 입력 대화창

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.

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 21.3: 마스터 비밀번호 리셋하기

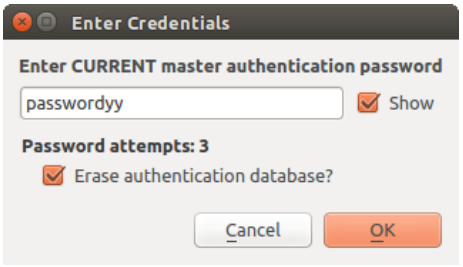


그림 21.4: 부정확한 시도를 세 번 한 후의 QGIS 비밀번호 메시지

21.1.3 인증 환경 설정

QGIS 옵션 대화창의 (*Settings* ▷ *Options*) *Authentication* 탭에 있는 **Configurations** 부분에서 인증 환경 설정을 관리할 수 있습니다.

새 환경 설정을 추가하려면  아이콘을, 그리고 환경 설정을 제거하려면  버튼을 클릭하십시오. 기존 환경 설정을 수정하려면  아이콘을 클릭하십시오.

OWS 서비스 연결과 같은 지정한 서비스 연결을 환경 설정할 때, 인증 환경 설정 관리에 대해 세 가지 유형의 작업 (추가, 편집 및 제거) 을 할 수 있습니다. 이를 위해, 환경 설정 선택기 안에 인증 데이터베이스에서 찾은 환경 설정을 완전히 관리할 수 있는 액션 버튼들이 있습니다. 이런 경우, 환경 설정을 좀 더 종합적으로 관리해야 할 필요가 없는 한, QGIS 옵션 대화창의 *Authentication* 탭에 있는 **Configurations** 부분으로 갈 필요가 없습니다.

인증 환경 설정을 생성하거나 편집할 때, 요구되는 정보는 명칭, 인증 방법, 그리고 인증 방법이 요구하는 기타 정보입니다. (사용할 수 있는 인증 유형에 관한 자세한 내용은인증 방법 을 참조하세요.)

21.1.4 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

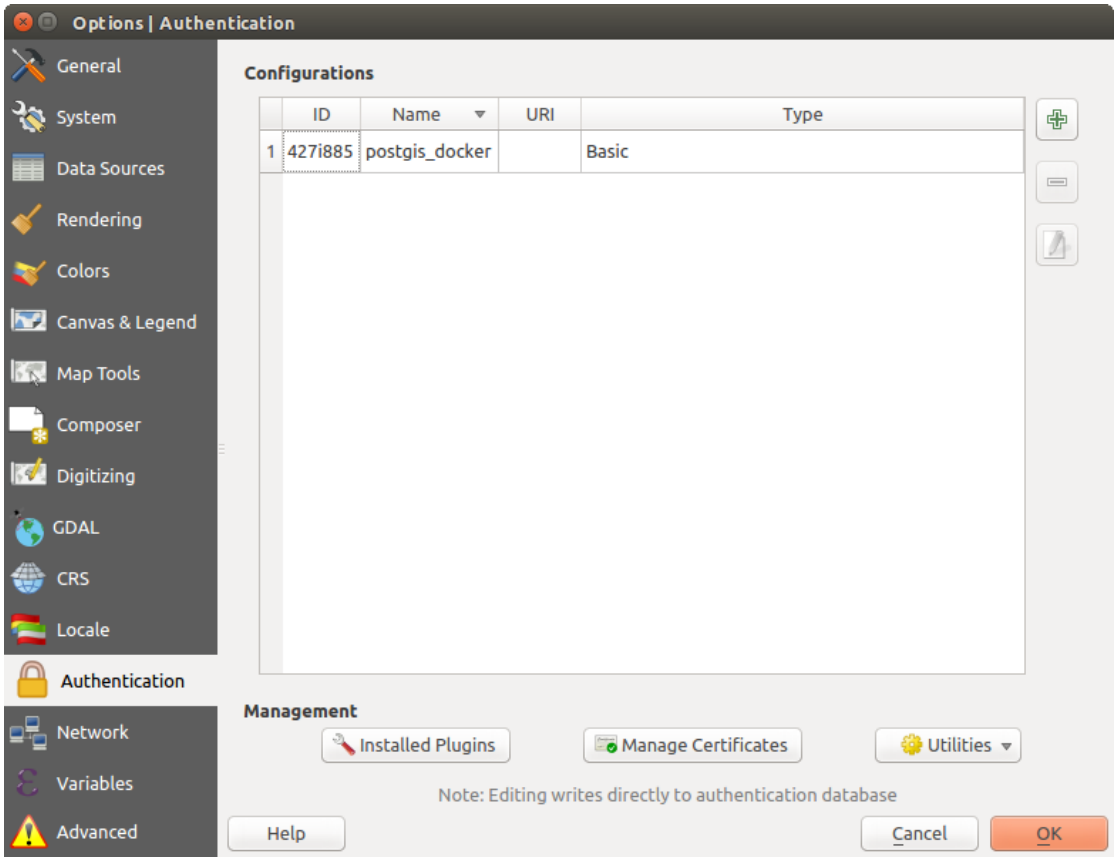


그림 21.5: 환경 설정 편집기

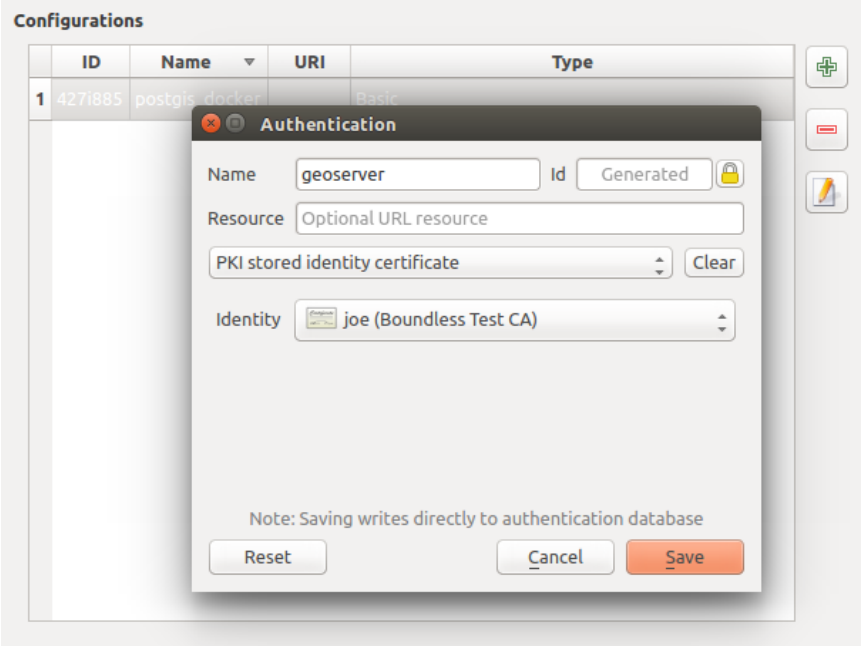


그림 21.6: 환경 설정 편집기 안에서 환경 설정 추가하기

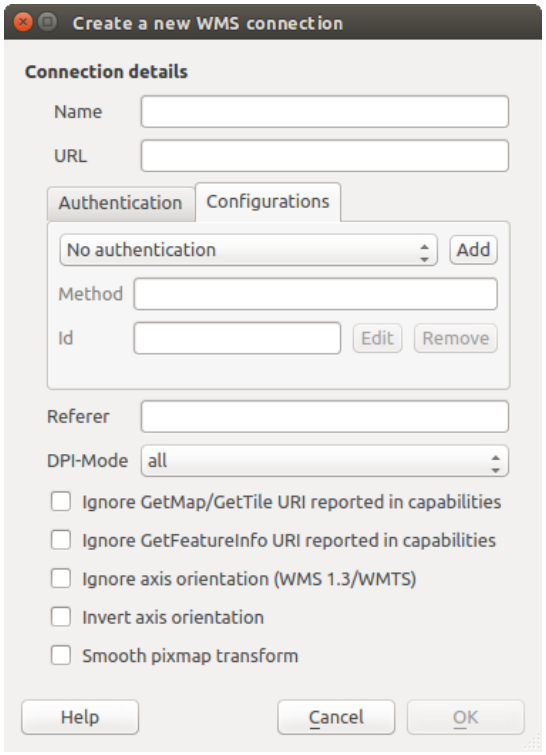


그림 21.7: *Add*, *Edit* 및 *Remove* 인증 환경 설정 버튼이 표시된 WMS 연결 대화창

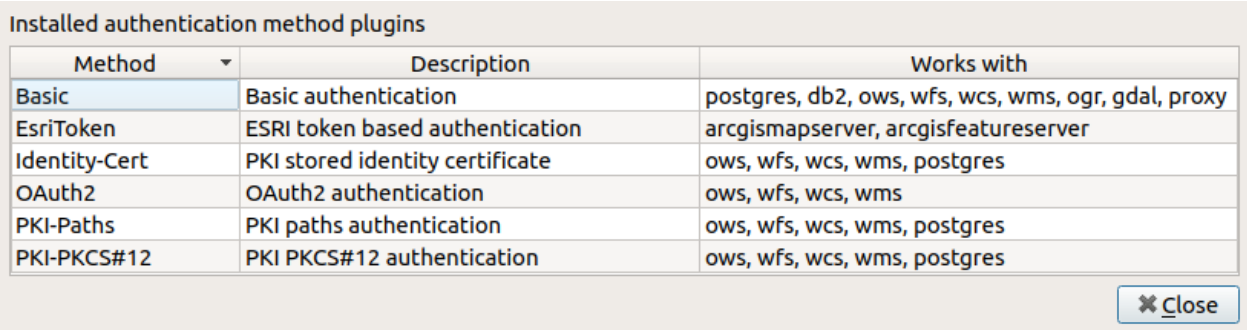


그림 21.8: 사용할 수 있는 인증 방법 플러그인 목록

설치본에 플러그인을 추가할 생각이라면, 사용자 플러그인이 기존 설치본과 동일한 버전을 대상으로 컴파일됐는지 확인하십시오.

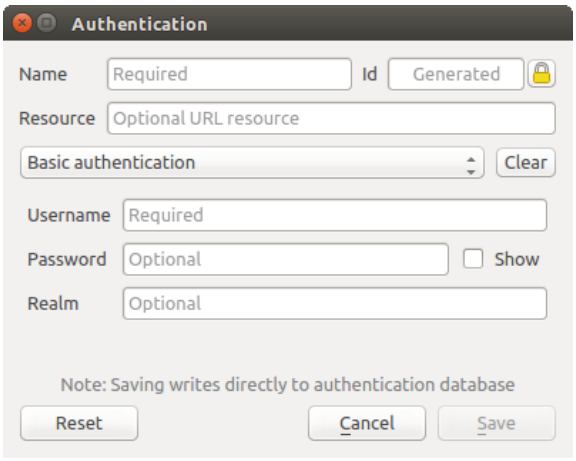


그림 21.9: HTTP 기본 인증 환경 설정

참고: *Resource* URL 은 현재 구현되지 않은 기능으로, 향후 지정한 URL 에 있는 리소스에 연결 시 특정 환경 설정을 자동 선택할 수 있도록 개발할 예정입니다.

21.1.5 마스터 비밀번호 및 인증 환경 설정 유틸리티

Settings ▸ *Options* 대화창의 *Authentication* 에서, 인증 데이터베이스 및 환경 설정을 관리할 수 있는 다음|settings| *Utilities* 액션의 콤보박스 목록을 찾을 수 있습니다:

- **마스터 비밀번호 입력**: 인증 데이터베이스 명령어 수행과는 무관하게 마스터 비밀번호 입력 대화창을 엽니다.
- **캐시된 마스터 비밀번호 삭제**: 마스터 비밀번호가 입력 대화창을 통해 설정되어있다면 이를 해제합니다.
- **마스터 비밀번호 리셋**: 마스터 비밀번호를 바꾸기 위한 (현재 비밀번호를 알고 있어야만 합니다) 대화창을 열고, 선택적으로 현재 데이터베이스를 백업합니다.
- **네트워크 인증 접근 캐시 삭제**: 모든 연결의 인증 캐시를 삭제합니다.
- **SSL 오류 발생 시 자동적으로 네트워크 인증 접근 캐시 삭제**: 연결 캐시는 연결이 실패하더라도 연결에 대한 모든 인증 데이터를 저장합니다. 사용자가 인증 환경 설정이나 인증서 기관을 변경하는 경우, 인증 캐시를 삭제하거나 QGIS 를 재시작해야 합니다. 이 옵션을 체크하면, SSL 오류가 발생해서 사용자가 연결을 중지하도록 선택할 때마다 인증 캐시를 자동적으로 삭제할 것입니다.
- **마스터 비밀번호를 사용자 월렛/키링과 통합**: 사용자 개인 월렛/키링에 마스터 비밀번호를 추가합니다.
- **사용자 월렛/키링에 마스터 비밀번호를 저장/업데이트**: 사용자 월렛/키링에 변경된 마스터 비밀번호를 업데이트합니다.
- **사용자 월렛/키링에서 마스터 비밀번호를 삭제**: 사용자 월렛/키링에서 마스터 비밀번호를 삭제합니다.
- **비밀번호 도우미 디버그 로그 활성화**: 인증 방법의 모든 로그 정보를 담게 될 디버그 도구를 활성화합니다.
- **캐시된 인증 환경 설정 삭제**: 네트워크 연결 속도를 높이는데 사용되는 환경 설정용 내부 조회 캐시를 지웁니다. QGIS 의 핵심 네트워크 접근 관리자의 캐시는 지워지지 않습니다. QGIS 를 다시 시작해야 하기 때문입니다.
- **모든 인증 환경 설정 제거**: 저장된 다른 레코드는 삭제하지 않은 채, 모든 환경 설정 레코드의 데이터베이스를 지웁니다.

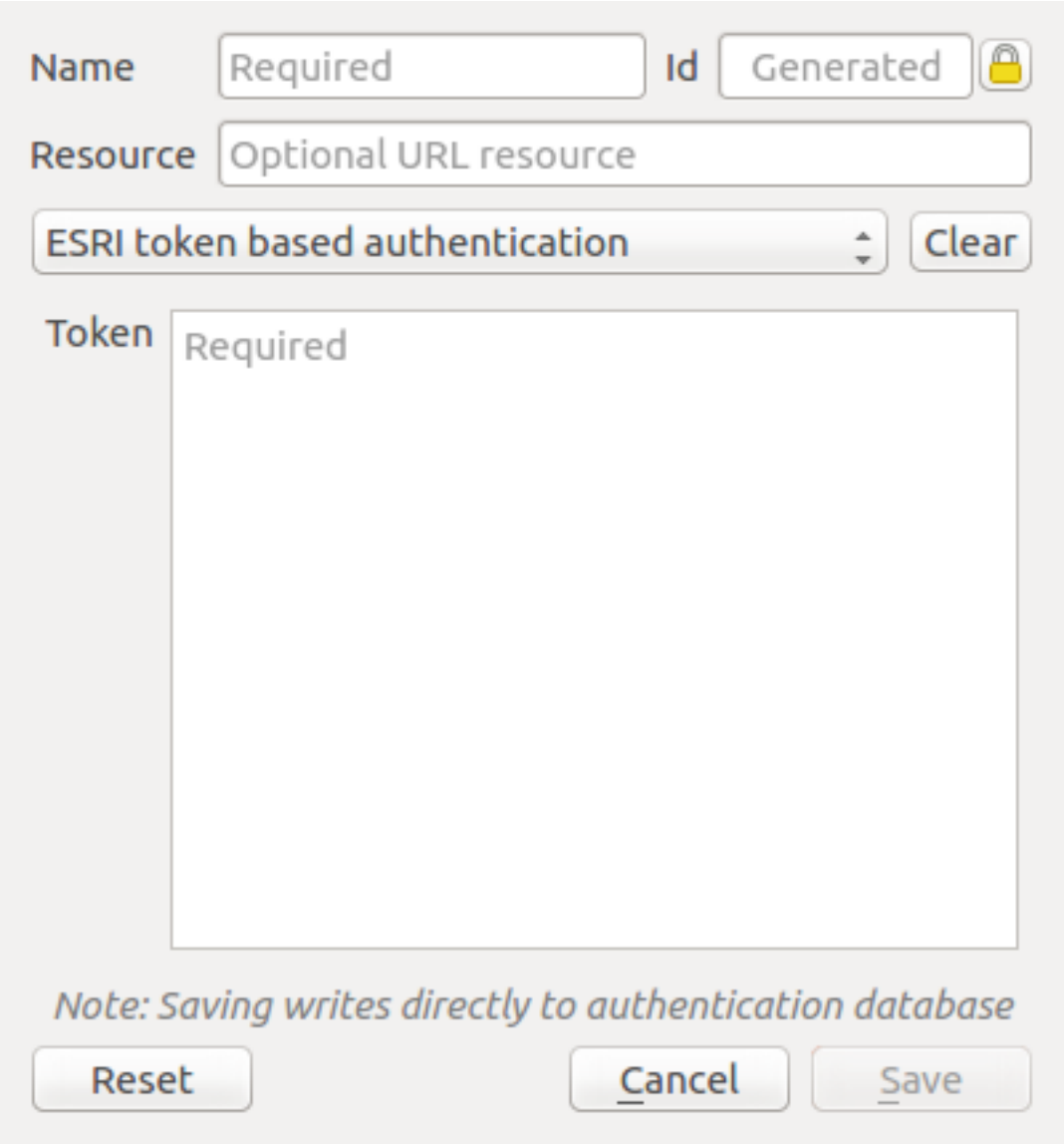


그림 21.10: ESRI 토큰 인증 환경 설정

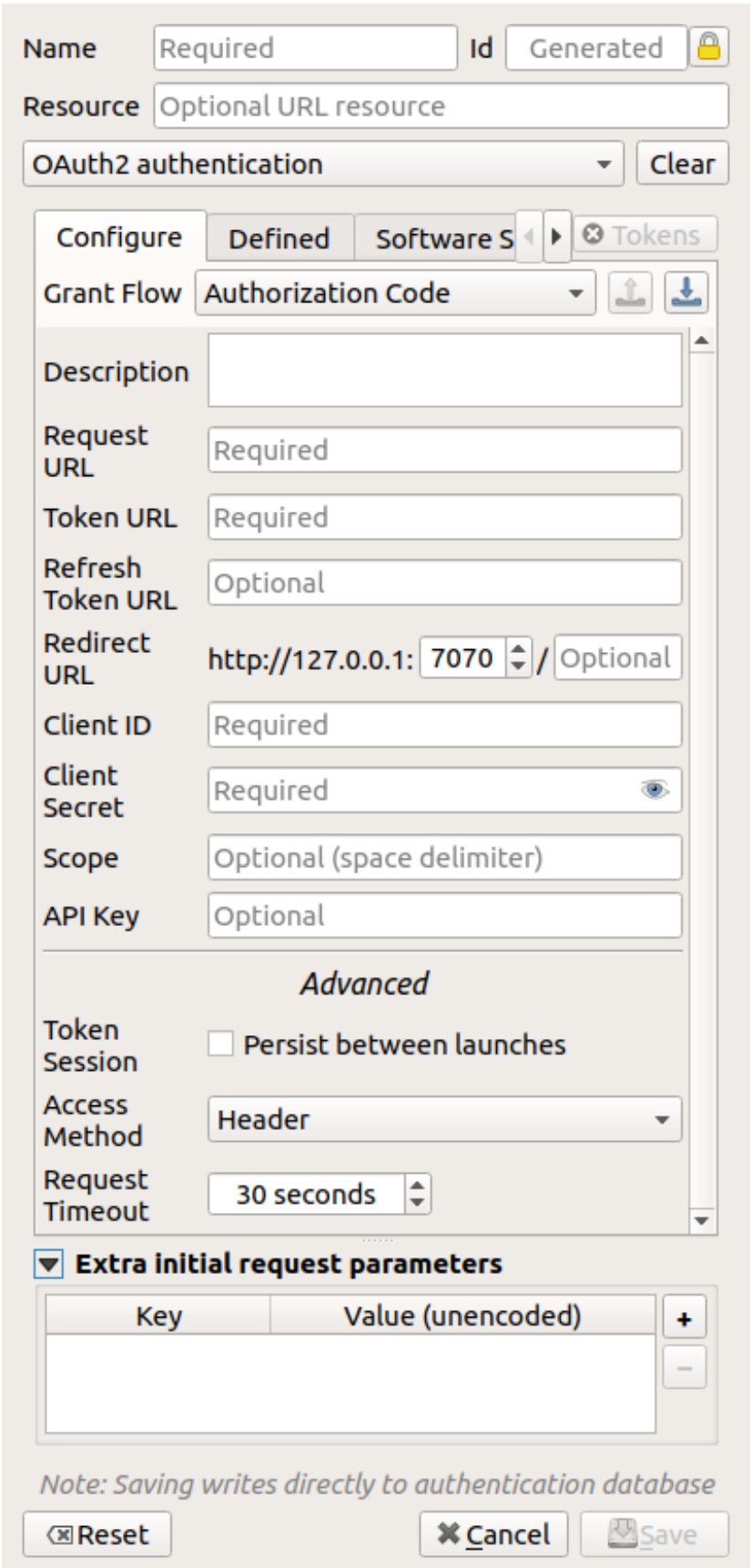


그림 21.11: OAuth2 인증 환경 설정

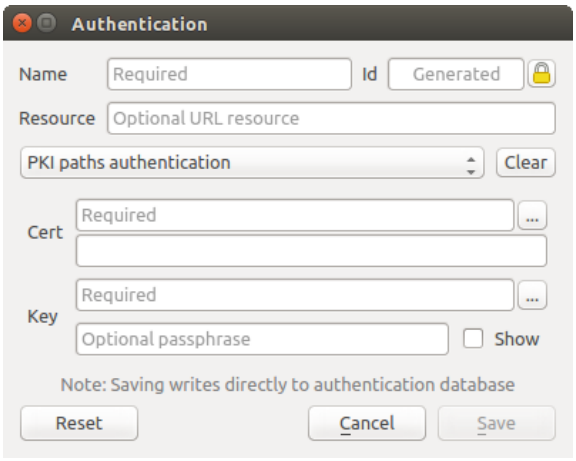


그림 21.12: PKI 경로 인증 환경 설정

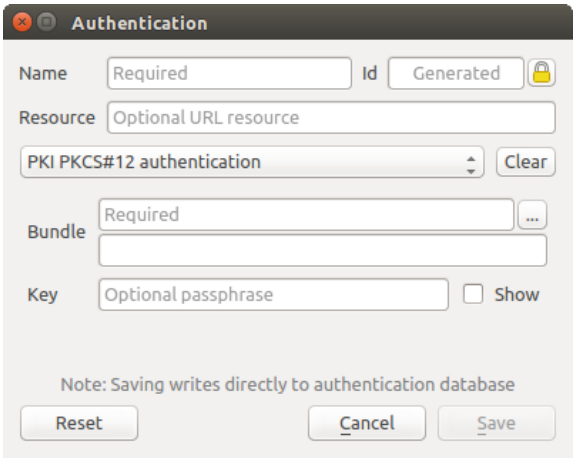


그림 21.13: PKI PKCS#12 파일 경로 인증 환경 설정

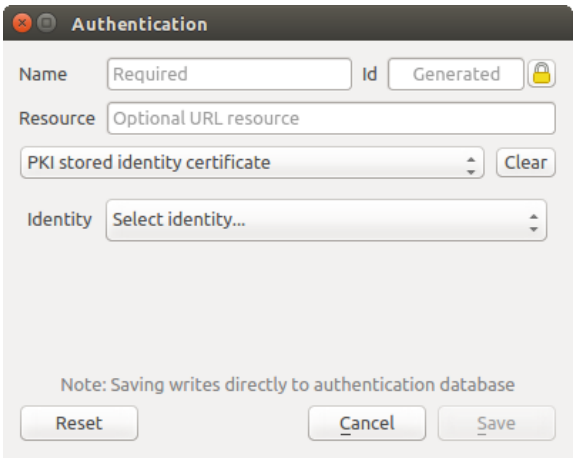


그림 21.14: 저장된 신원 인증 환경 설정

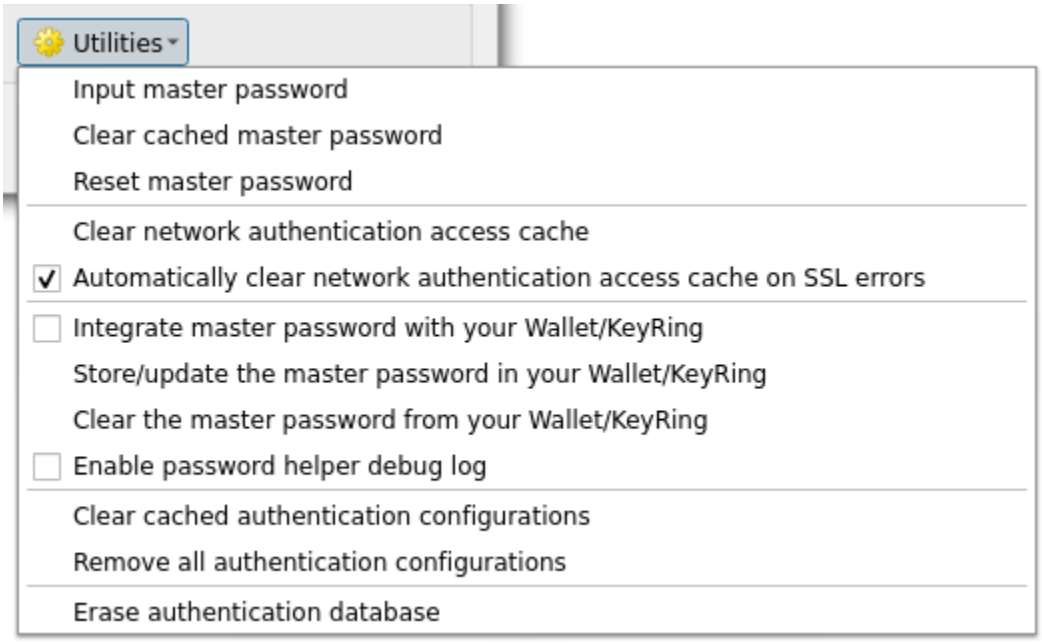


그림 21.15: 유틸리티 목록

- **인증 데이터베이스 지우기**: 현재 데이터베이스의 백업과 데이터베이스 테이블 구조의 완전한 리빌딩을 예약합니다. 이런 작업들은 나중에 실행되도록 예약됩니다. 프로젝트 불러오기와 같은 다른 작업이 이를 방해하지 않도록 하거나 일시적으로 누락된 데이터베이스로 인해 오류가 발생하지 않도록 하기 위해서입니다.

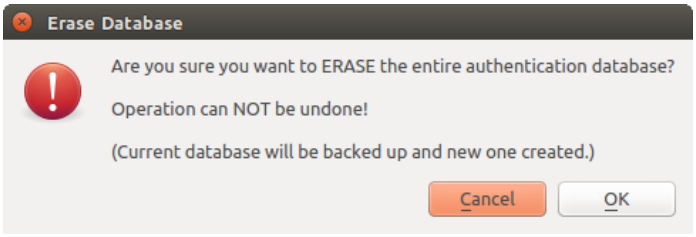


그림 21.16: DB 삭제 확인 대화창

21.1.6 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.

선택기 위젯을 사용할 때, 아무것도 선택하지 않았거나 선택할 인증 환경 설정이 없거나 또는 데이터베이스에서 이전에 할당한 환경 설정을 찾을 수 없는 경우 팝업 메뉴의 콤보박스에'No authentication'이라고 표시됩니다. *Method* 및 *Id* 텍스트 란은 읽기 전용으로, 선택한 인증 방법의 설명 및 환경 설정 ID 를 각각 표시합니다.

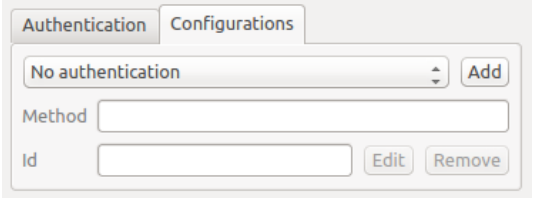


그림 21.17: 아무 인증도 없는 인증 환경 설정 선택기

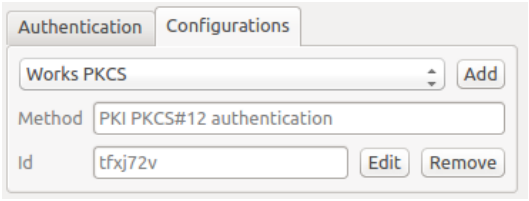


그림 21.18: 선택한 환경 설정을 표시한 인증 환경 설정 선택기

21.1.7 파이썬 바인딩

모든 클래스 및 공개 함수는 SIP 바인딩을 보유하고 있습니다. `QgsAuthCrypto` 를 제외하고 말이죠. 왜냐하면 파이썬을 통해서가 아니라 주 응용 프로그램이 마스터 비밀번호 해싱 및 인증 데이터베이스를 관리해야 하기 때문입니다. 파이썬 접근 관련 설명은보안 고려 사항 을 참조하세요.

21.2 사용자 인증 작업 흐름

21.2.1 HTTP(S) 인증

가장 흔한 리소스 연결 가운데 하나가 바로, 예를 들어 웹 매핑 서비스 같은 HTTP(S) 를 통한 연결로, 인증 방법 플러그인은 자주 이런 연결 유형을 위해 작업합니다. 인증 방법 플러그인은 HTTP 요청 객체에 접근해서 해당 요청은 물론 그 헤더도 조작할 수 있습니다. 따라서 수많은 형태의 인터넷 기반 인증을 할 수 있습니다. 표준적인 사용자명/비밀번호 인증 방법을 사용해서 HTTP(S) 를 통해 연결하는 경우, 연결 시 HTTP BASIC 인증을 시도할 것입니다.

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.

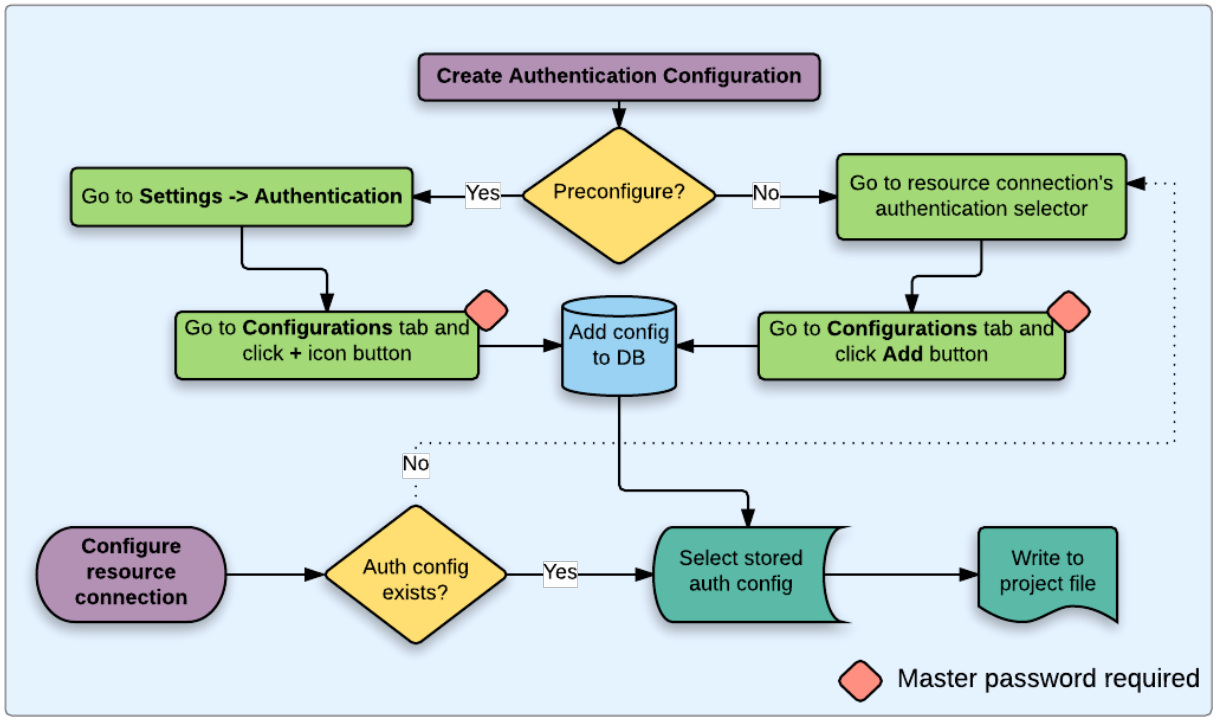


그림 21.19: 일반적인 사용자 작업 흐름

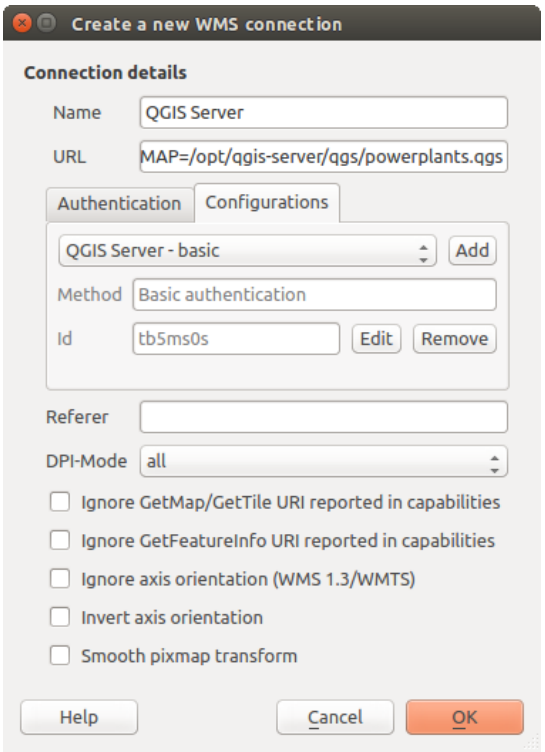


그림 21.20: HTTP BASIC 에 대한 WMS 연결 환경 설정하기

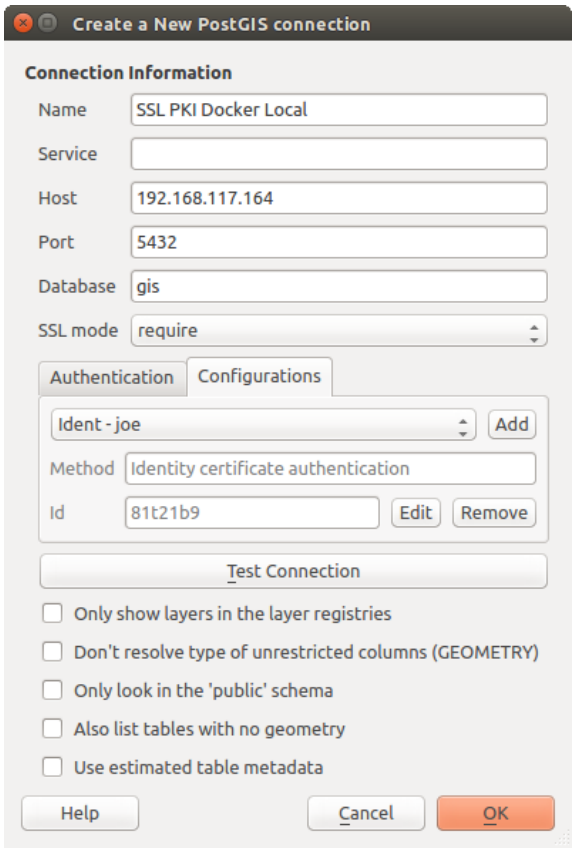


그림 21.21: PostgreSQL SSL-PKI 연결 환경 설정하기

21.2.3 PKI 인증

인증 시스템 내에서 PKI 항목을 환경 설정하는 경우, 데이터베이스로 항목을 가져오거나 또는 사용자 파일 시스템에 저장된 항목 파일을 참조하는 옵션을 선택할 수 있습니다. 해당 항목이 자주 변경되거나, 또는 시스템 관리자가 해당 항목을 교체할 예정인 경우 후자가 유용할 수도 있습니다. 두 옵션 모두, 데이터베이스 안에 있는 개인 키 (private key) 에 접근하는 데 필요한 암호 (passphrase) 를 저장해야 할 것입니다.

인증서 관리자 안에 있는 개별 편집기에서 모든 PKI 항목들을 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: 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.

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 유효하지 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.

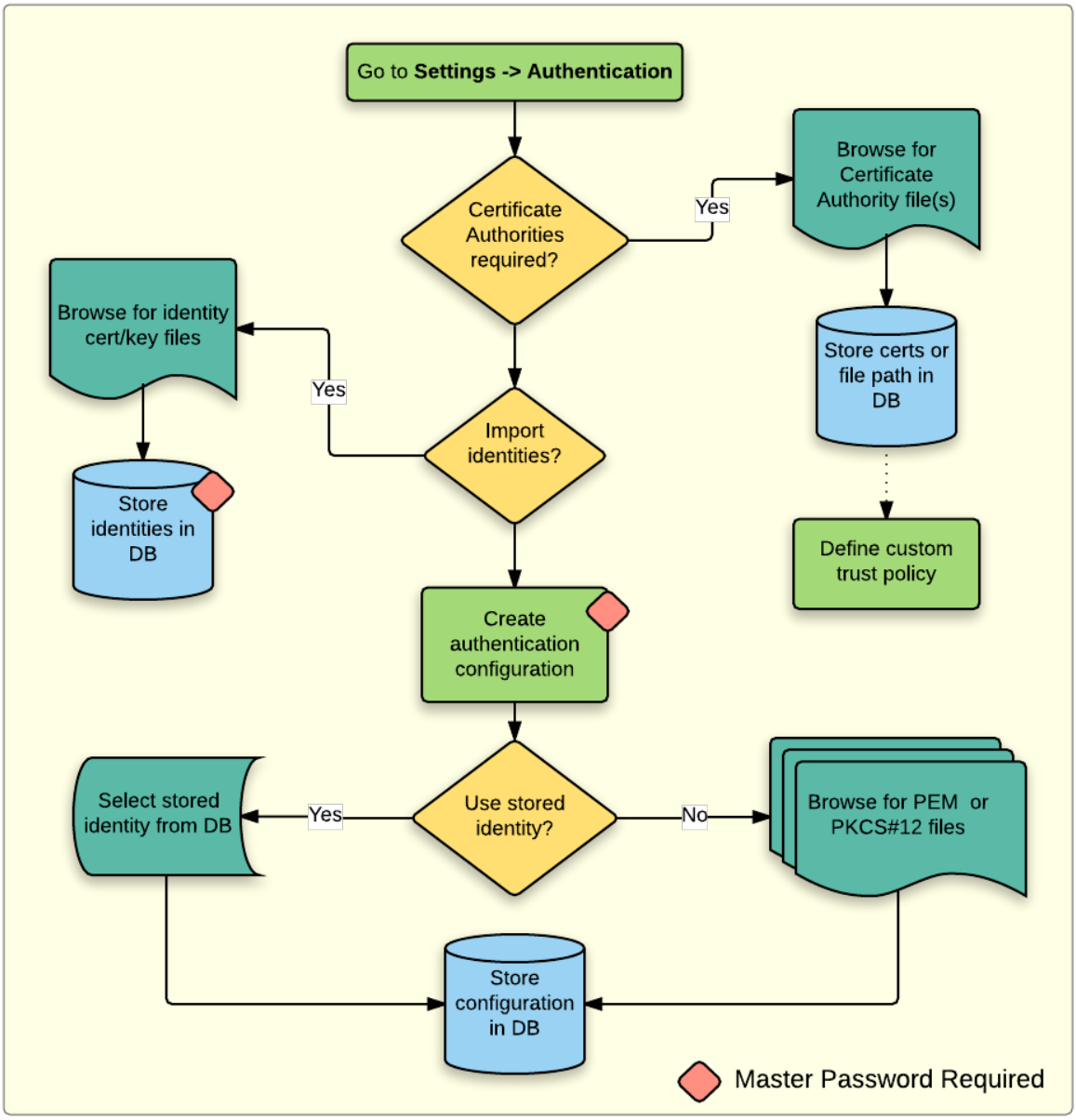


그림 21.22: PKI 환경 설정 작업 흐름

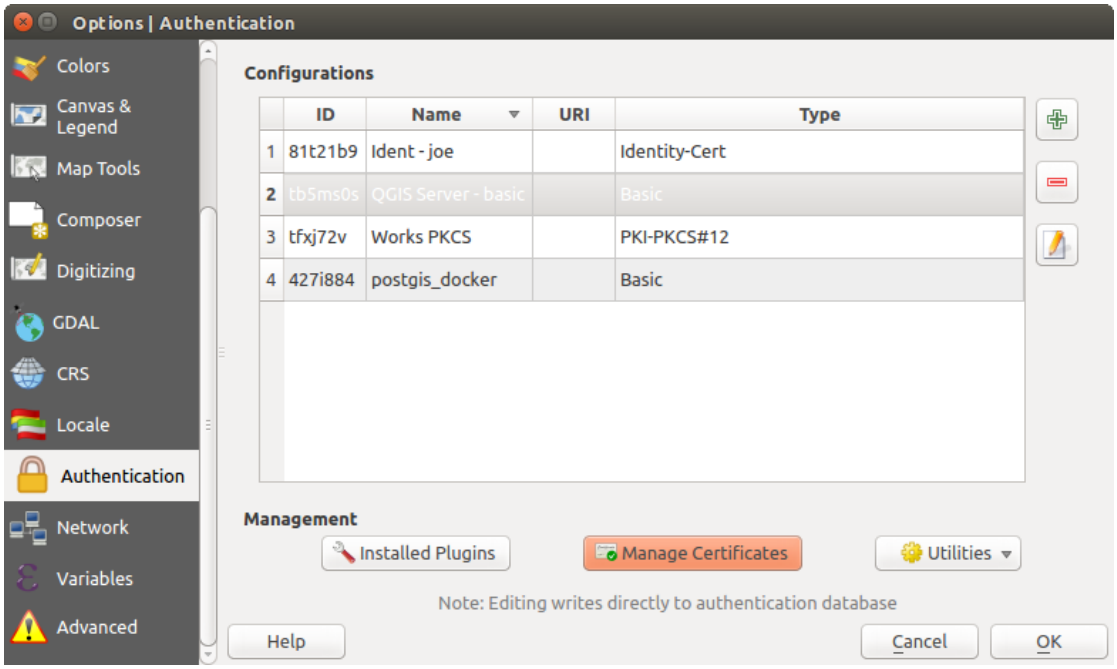


그림 21.23: 인증서 관리자 열기

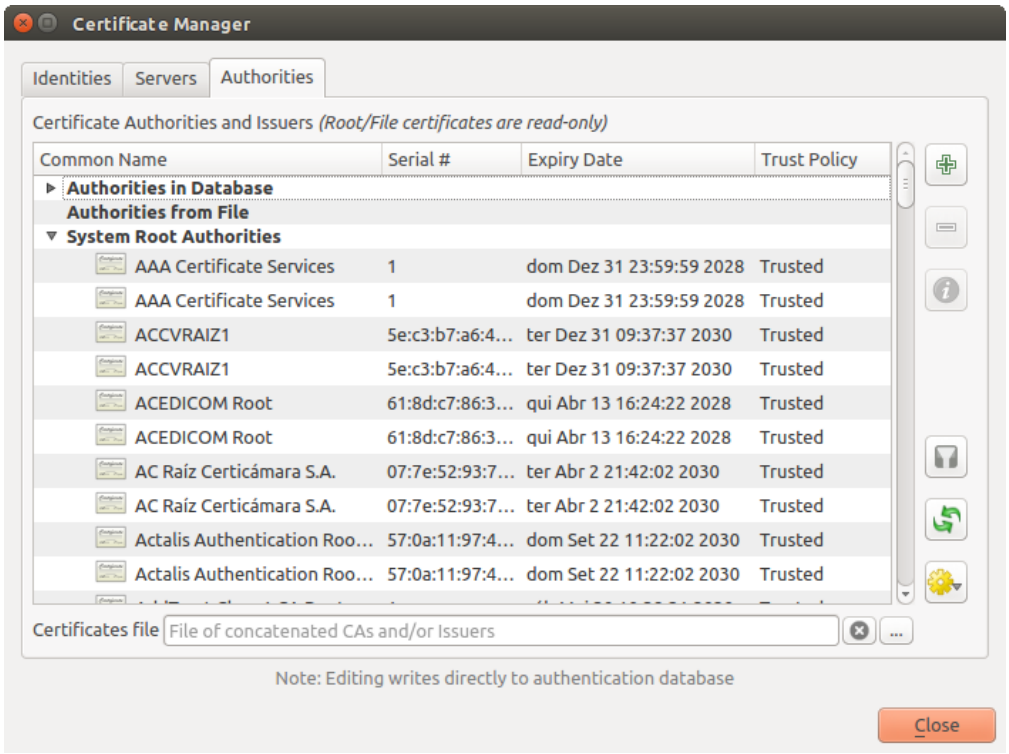


그림 21.24: 권한 편집기

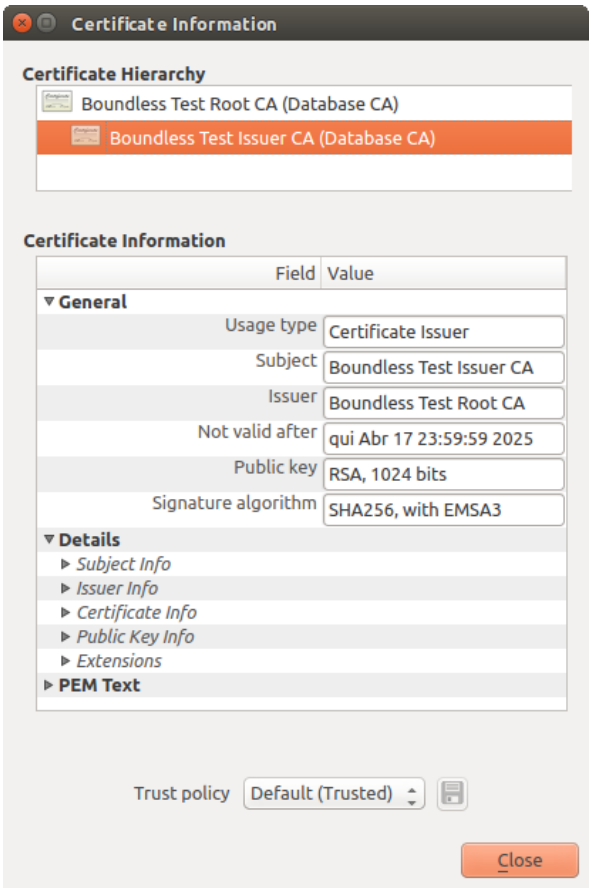


그림 21.25: 인증서 정보 대화창

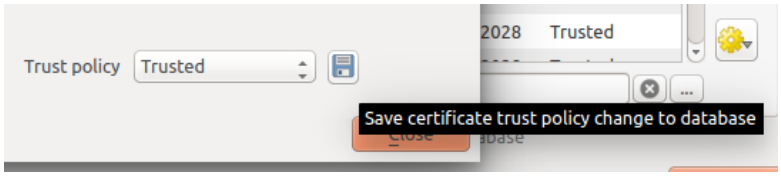


그림 21.26: 신뢰 정책의 변경 사항을 저장하기

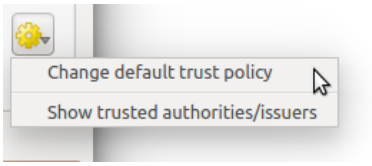


그림 21.27: 권한 옵션 액션

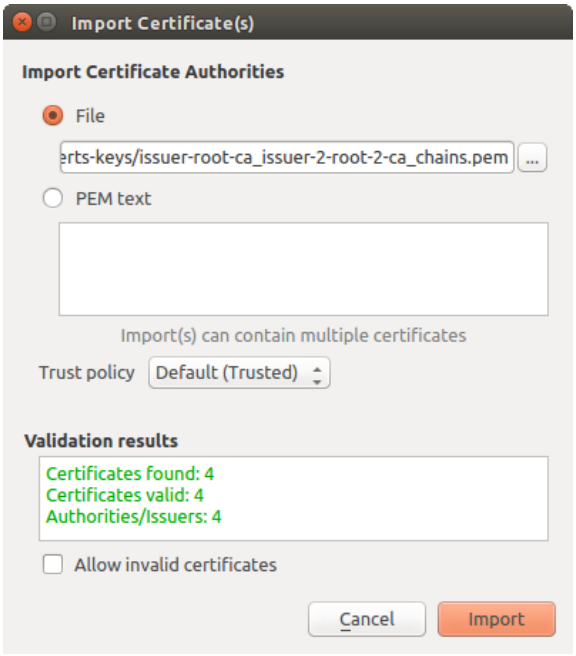


그림 21.28: 인증서 가져오기 대화창

참고: |radioButtonOn| *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 항목 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.

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) 항목으로 구성된 개인 신원 묶음을 지원합니다. 개인 키 또는 신원 묶음이 암호로 보호돼 있는 경우, 가져오기 전에 항목을 검증하기 위해 비밀번호를 입력해야 합니다. 마찬가지로, 신원 묶음에 있는 클라이언트 인증서가 유효하지 않을 경우 (예를 들어 유효 기간 전 또는 후인 경우) 신원 묶음을 가져올 수 없습니다.

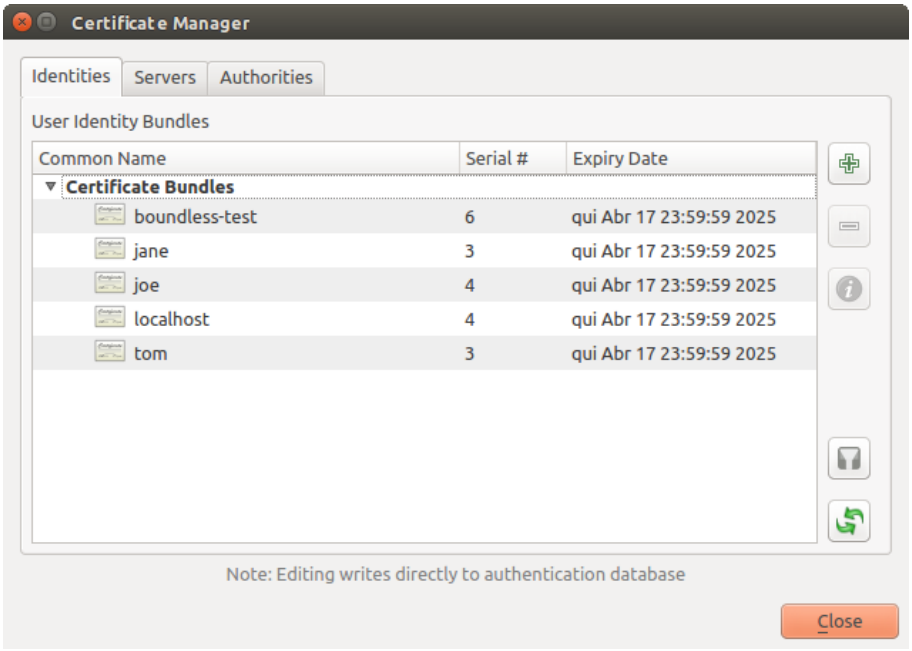


그림 21.29: 신원 편집기

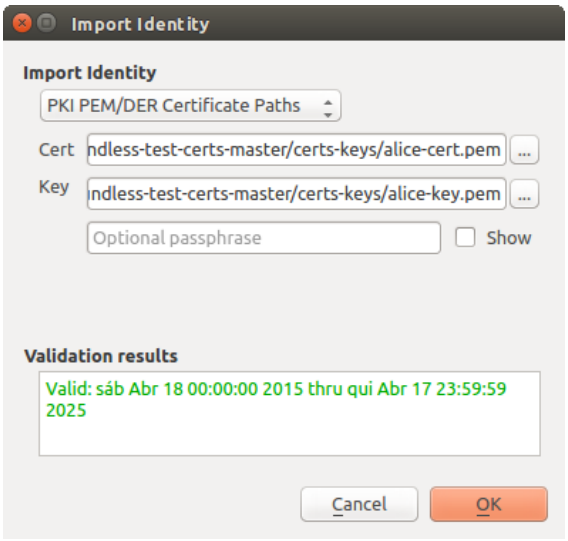


그림 21.30: PEM/DER 신원 묶음 가져오기

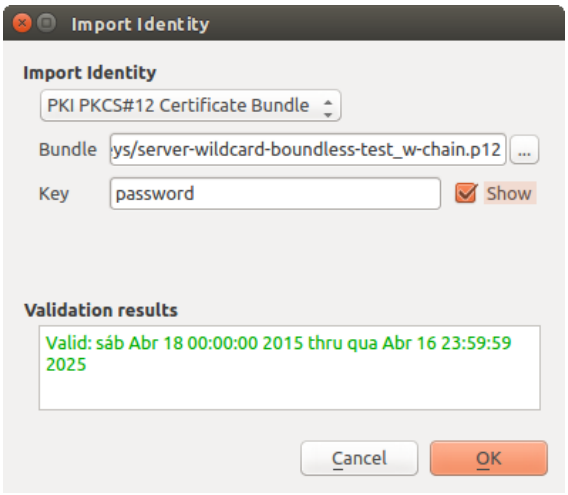


그림 21.31: PKCS#12 신원 묶음 가져오기

21.2.4 잘못된 레이어 처리

때때로, 프로젝트 파일에 저장된 인증 환경 설정 ID 가 더 이상 유효하지 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.

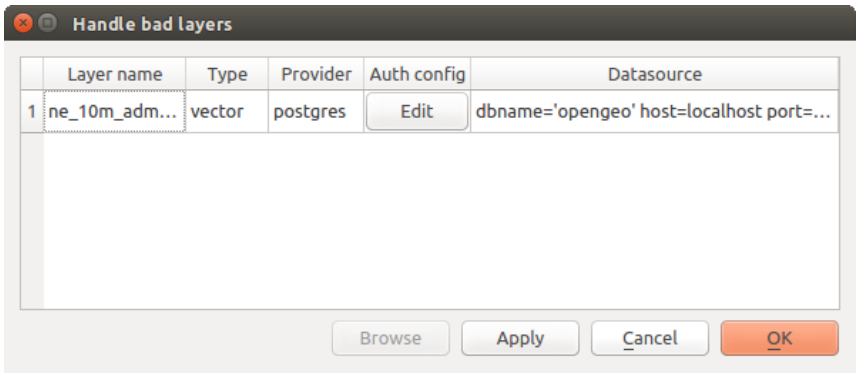


그림 21.32: 인증을 통해 잘못된 레이어 처리하기

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.

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 유효하지 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

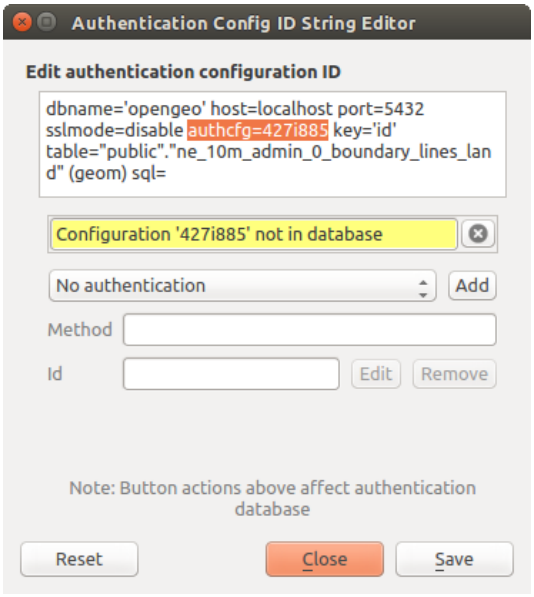


그림 21.33: 잘못된 레이어의 인증 환경 설정 ID 편집하기

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.

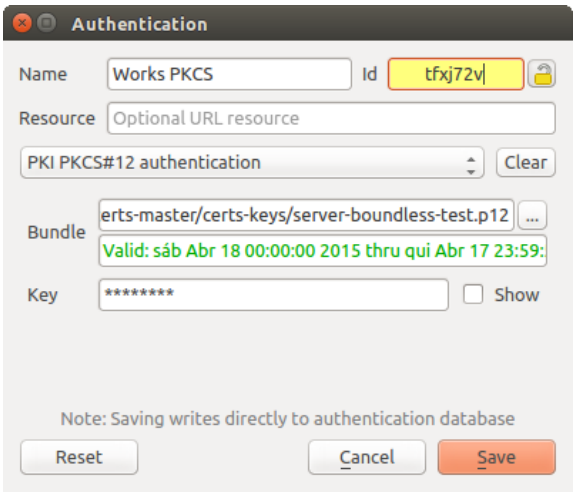


그림 21.34: 레이어의 인증 환경 설정 ID 변경하기 (고정 해제된 노란색 텍스트 란)

경고: 인증 환경 설정 ID 를 변경하는 것은 고급 작업으로 간주되며, 변경해야 하는 이유를 완전히 이해하고 있는 경우에만 변경해야 합니다. ID 를 편집하기 전에 클릭해야 하는|locked| 아이콘이 ID 텍스트 란 옆에 있는 이유입니다.

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

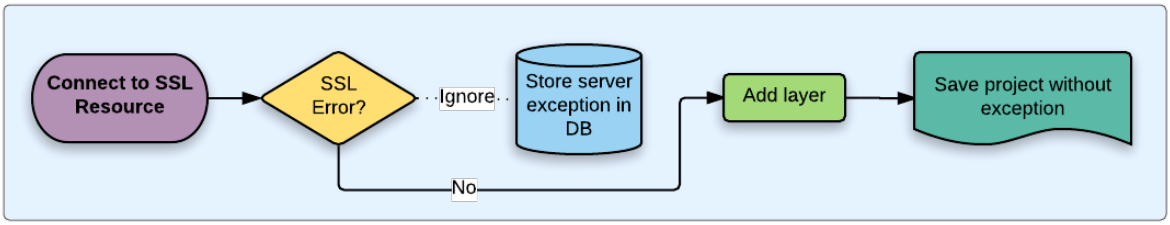


그림 21.35: SSL 서버 예외

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:

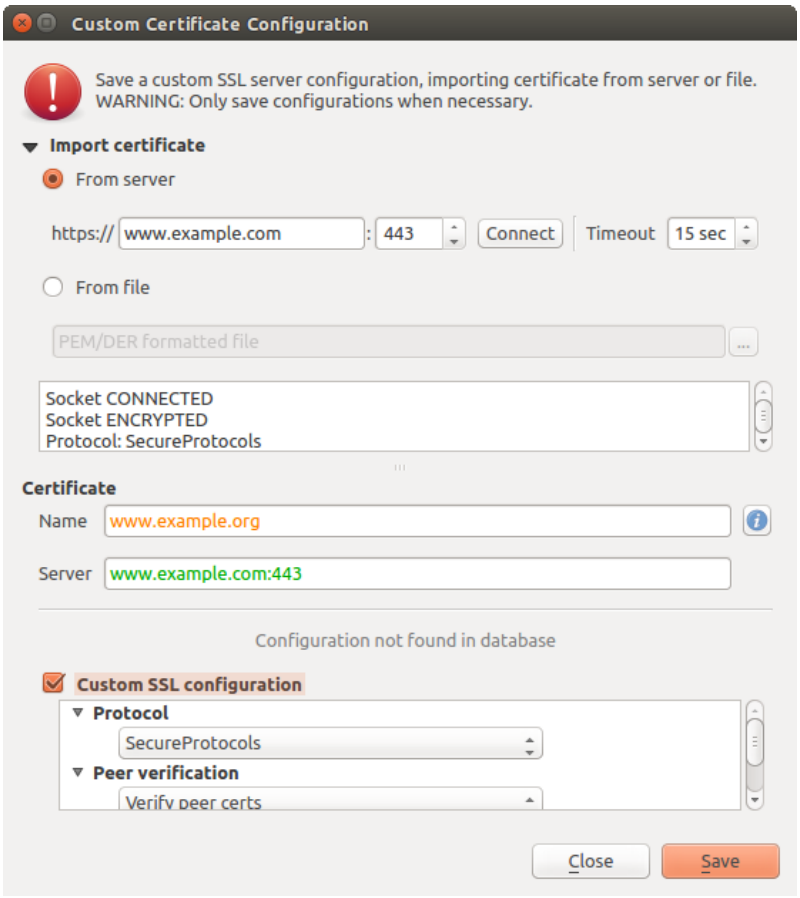


그림 21.36: 환경 설정을 직접 추가하기

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):

```
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.

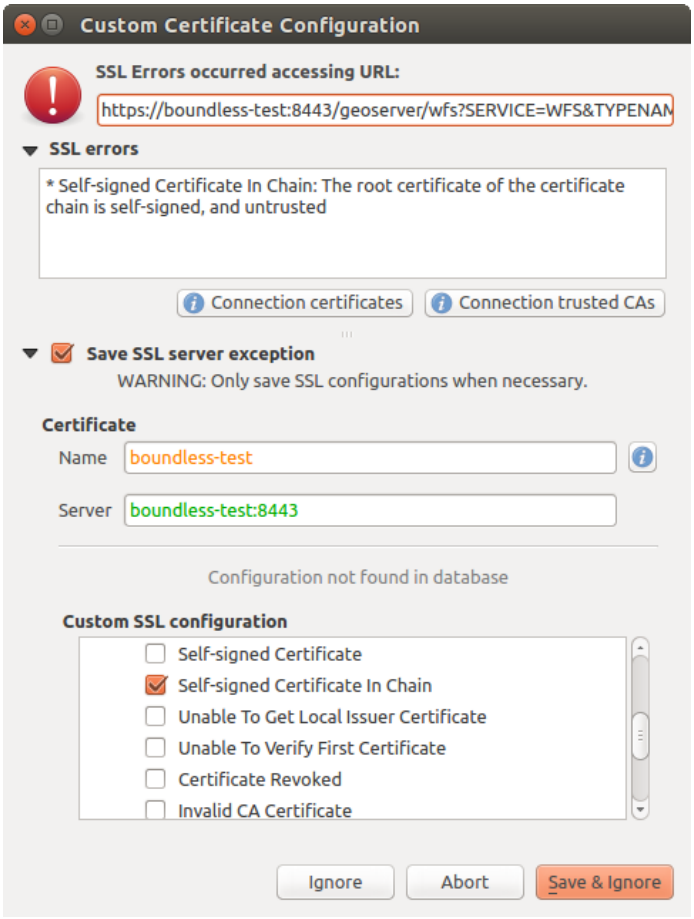


그림 21.37: SSL 오류 발생 도중 환경 설정을 추가하기

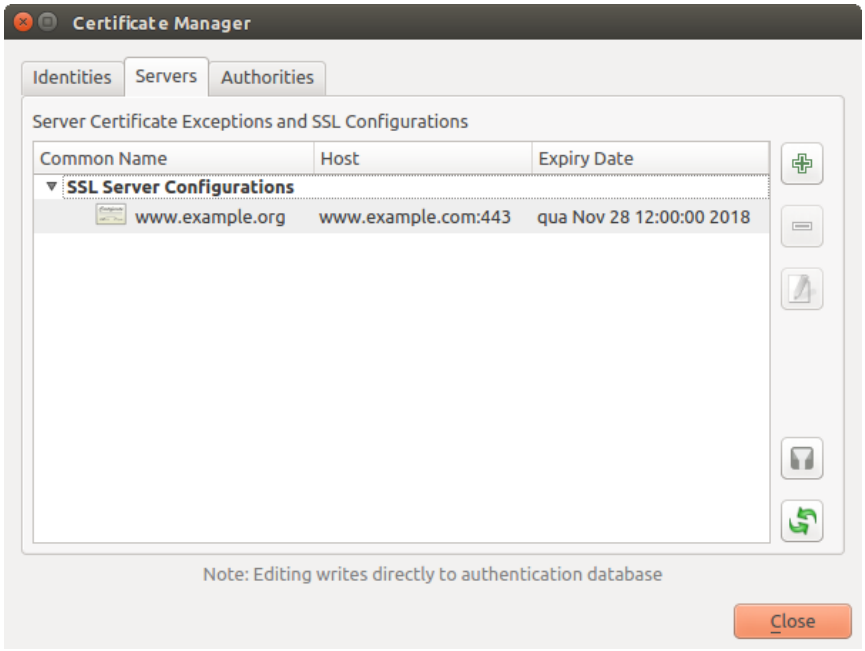


그림 21.38: 기존 SSL 환경 설정

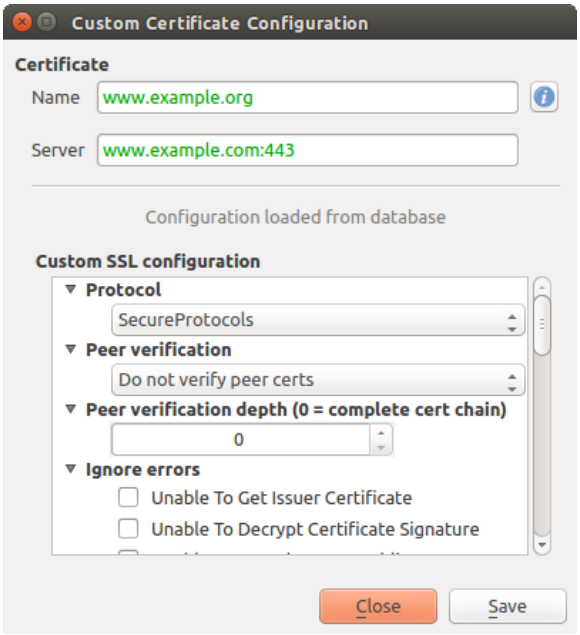


그림 21.39: 기존 SSL 환경 설정 편집하기

21.3 보안 고려 사항

마스터 비밀번호를 입력하고 나면, 파이어폭스 작동 방식과 비슷하게 API 가 열려 인증 데이터베이스에 있는 인증 환경 설정에 접근할 수 있습니다. 하지만 맨 처음 실행하는 경우, 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.

맨 처음 피처를 배포할 때의 이런 문제점에 대한 빠른 해결책은 그냥 대부분의 인증 시스템 용 PyQGIS 바인딩을 포함시키지 않는 것입니다.

강력하지는 않지만 단순한 또다른 해결책은 *Settings* ▷ *Options* 대화창의 *Authentication* 탭에 파이썬이 인증 시스템에 접근할 수 있는지 여부를 결정하는 (기본값이 'never'인) 콤보박스를 다음과 같이 추가하는 것입니다:

```
"Allow Python access to authentication system"
Choices: [ confirm once per session | always confirm | always allow | never]
```

이런 옵션의 설정은 파이썬이 접근할 수 없는, 예를 들어 인증 데이터베이스 같은 위치에 저장해야 하고, 또 마스터 비밀번호로 암호화해야 할 것입니다.

- 또다른 옵션은 사용자가 특별히 인증 시스템에 접근을 허용한 플러그인을 추적하는 것일 수도 있습니다.
- 어떤 플러그인이 실제로 호출했는지 알아내는 것이 힘들 수도 있지만 말이죠.
- 고유의 가상 환경에 존재할 가능성이 큰 샌드박스 플러그인은 인증된 또다른 플러그인에서 인증 환경 설정을 해킹하는 '교차 플러그인 (cross-plugin)' 해킹의 가능성을 줄여줄 것입니다. 교차 플러그인 통신 또한 제한해야 할 수도 있지만, 제 3 자 플러그인 사이의 통신만 제한할 수도 있을 겁니다.
- 검증된 플러그인 개발자에게 코드 서명 (code-signing) 인증서를 발급하는 것도 훌륭한 해결책 가운데 하나입니다. 플러그인을 불러올 때 플러그인의 인증서를 검증하면 되니까요. 필요한 경우 기존 인증서 관리자 대화창을 통해 플러그인과 관련된 인증서를 신뢰할 수 없다는 정책을 직접 설정할 수도 있습니다.
- 아니면, 파이썬이 민감한 인증 시스템 데이터에 아예 접근하지 못 하도록 할 수도 있습니다.

- 주 응용 프로그램에서 불러온 마스터 비밀번호와 인증 환경을 유지한 채로 오직 QGIS 핵심 위젯을 사용해서만, 또는 통합된 인증 시스템을 복제함으로써만 플러그인이 인증 환경 설정을 보유한 리소스와 작업할 수 있도록 허용할 수도 있습니다.

C++ 플러그인에도 동일한 보안 고려 사항이 적용됩니다. 다만 파이썬과는 달리 간단하게 제거할 수 있는 함수 바인딩이 없기 때문에, 접근을 제한하는 작업이 더 힘들어질 것입니다.

21.3.1 제약 사항

OpenSSL 적용과 관련한, 이해하기 힘든 사용권 계약 및 내보내기 문제점이 있습니다. Qt 가 SSL 인증서와 작업할 수 있게 하려면, Qt 가 OpenSSL 라이브러리에 접근할 수 있어야 합니다. Qt 를 어떻게 컴파일했느냐에 따라 다르지만, 기본값은 (내보내기 제한을 피하기 위해) 실행 시간 (run-time) 에 OpenSSL 라이브러리에 동적으로 링크를 거는 것입니다.

QGIS 인증서 권한 (QCA) 도 실행 시간에 `qca-ossl` (QCA-OpenSSL) 플러그인을 불러오기 때문에 QCA 에 링크를 걸어도 제약 사항이 발생하지 않는다는 비슷한 전략을 따릅니다. 패키지 생산자가 관련 플러그인을 포함시키는 경우, QGIS 가 어떤 OpenSSL 링크 연결 관련 제약 사항에도 걸리지 않도록 보장해야 하는 쪽은 패키지 생산자입니다…일 겁니다. 저도 확실히 몰라요. 전 변호사가 아니니까요.

실행 시간에 `qca-ossl` 플러그인을 찾을 수 없는 경우, 인증 시스템은 스스로를 안전하게 비활성화합니다.

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 2D 및 3D 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

예제를 위해 QGIS 알래스카 데이터셋 (예시 데이터 다운로드 참조) 을 사용할 것입니다. 알래스카 데이터셋은 벡터 레이어 3 개와 래스터 표고 맵 1 개를 보유한 소용량 GRASS `LOCATION` 예시 데이터를 포함하고 있습니다. `grassdata` 라는 새 폴더를 생성한 다음, https://qgis.org/downloads/data/ 에서 QGIS 알래스카 데이터셋 `qgis_sample_data.zip` 파일을 다운로드해서 `grassdata` 폴더에 압축 해제하십시오.

GRASS 웹사이트 https://grass.osgeo.org/download/sample-data/ 에서 더 많은 GRASS `LOCATION` 데이터를 다운로드할 수 있습니다.

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/OGR 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.)

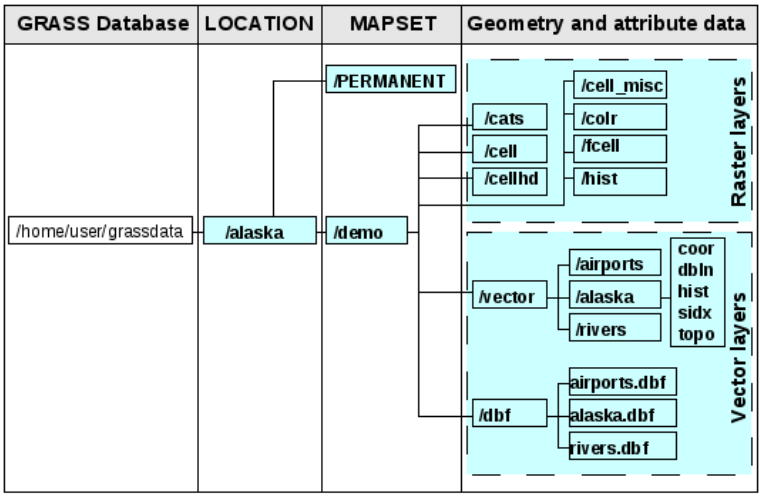


그림 22.1: 알래스카 로케이션에 있는 GRASS 데이터

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 명칭을 `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 명칭을 `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 알버 정적도법 (Alber's 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` shapefile 을 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 명칭을 — 이 예제에서는'alaska'를 — 입력한 다음 *Next* 버튼을 클릭하십시오.
7.  *Projection* 라디오 버튼을 체크해서 투영체 목록을 활성화한 다음 투영체를 선택하십시오.
8. 알버 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 요약을 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.

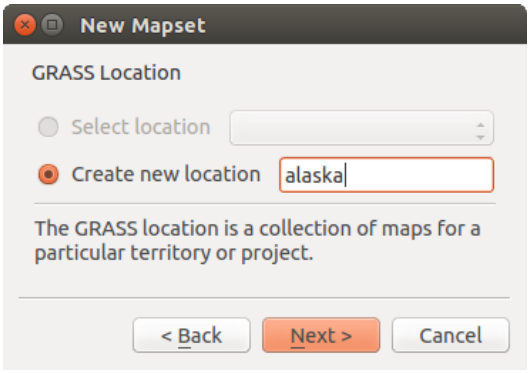


그림 22.2: QGIS 에서 새 GRASS 로케이션 또는 새 맵셋 생성하기

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 명칭을 입력하십시오. 마법사 대화창 하단에 기존 MAPSET 및 해당 소유자의 목록이 표시됩니다.
7. *Next* 버튼을 클릭해서 작업 내용이 정확한지 작업 요약을 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 명칭으로 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/documentation/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 명칭을 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 shapefile 을 생성하는 것과는 다릅니다. 왜냐하면 shapefile 은 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:

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.

표: GRASS 디지타이즈 작업 - GRASS 디지타이즈 작업 도구

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 명칭은 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, 명칭은 변경하지 마십시오. 왜냐하면 편집 작업을 다시 시작할 때 명칭을 사용해서 스타일을 리셋하기 때문입니다.

팁: 레이어 편집 중에 프로젝트 파일을 저장하지 마십시오. 레이어가 변경되지 않았을 경우 아무 의미도 가지지 못 하는'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.

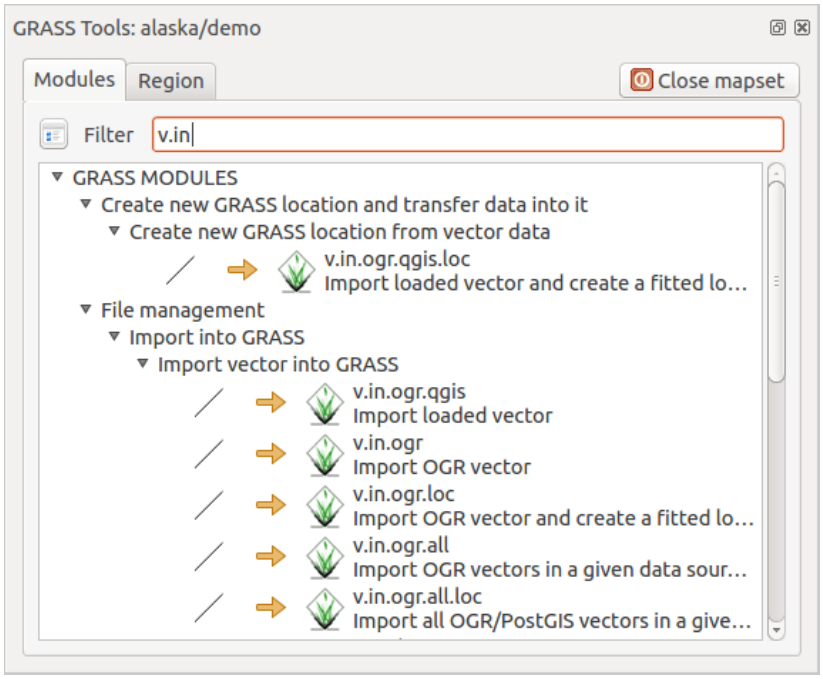


그림 22.3: GRASS 툴박스 및 모듈 트리

22.14.1 GRASS 모듈로 작업하기

GRASS 툴박스에 있는 GRASS 셸을 통해, 명령 줄 인터페이스에서 거의 모든 (300 개가 넘는) GRASS 모듈을 사용할 수 있습니다. 좀 더 사용자 친화적인 작업 환경을 제공하기 위해, GRASS 플러그인 툴박스 내부의 그래픽 인터페이스 대화창에서 약 200 개의 GRASS 모듈 및 기능을 사용할 수 있습니다.

GRASS 위키 사이트 에서, QGIS 3.16 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.

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.

산출물

Output 탭은 모듈의 산출물 상태에 관한 정보를 표시합니다. *Run* 버튼을 클릭했을 때, 모듈이 *Output* 탭으로

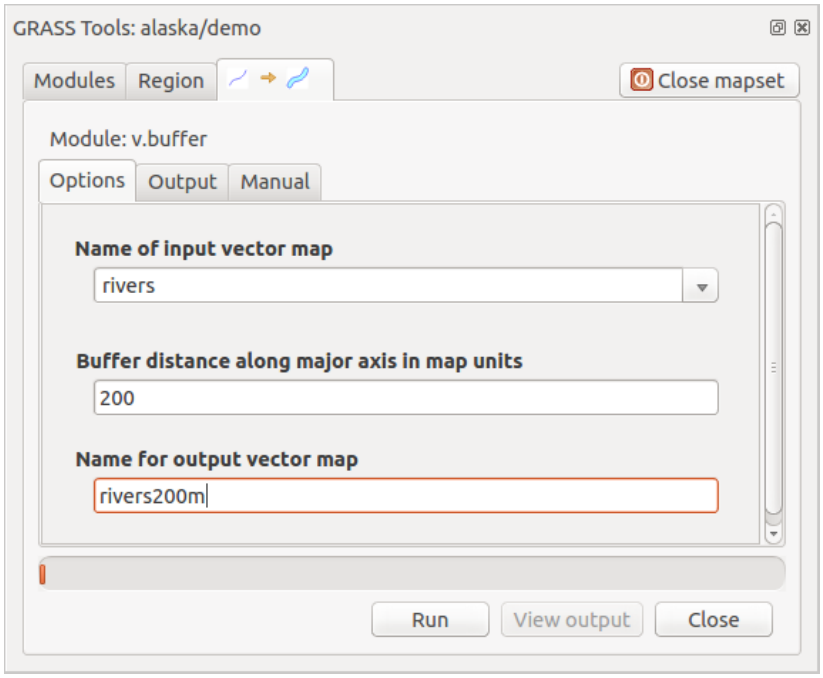


그림 22.4: GRASS 툴박스: 모듈 옵션

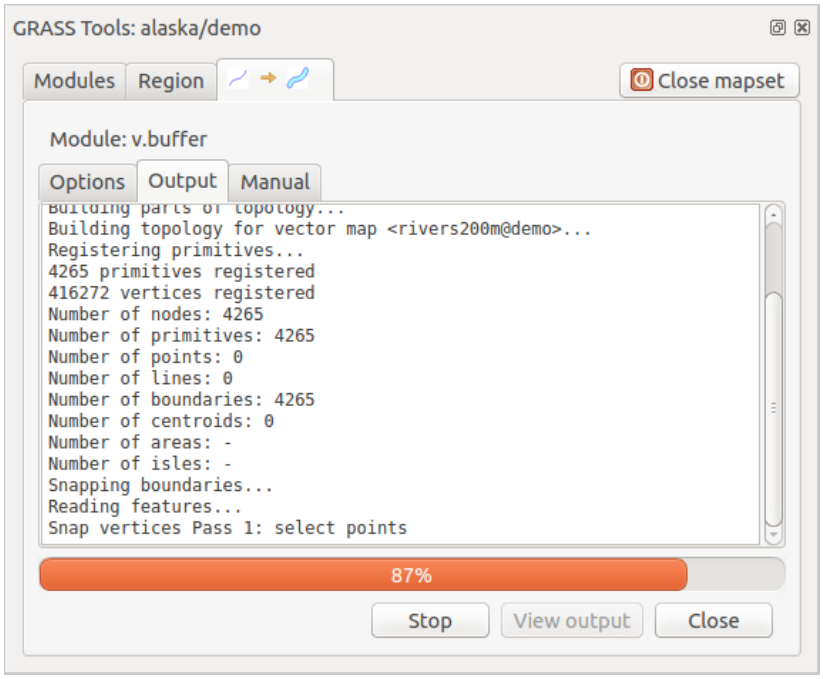


그림 22.5: GRASS 툴박스: 모듈 산출물

전환되어 사용자가 분석 과정에 관한 정보를 볼 수 있습니다. 모든 과정이 무사히 완료되면, 마지막으로 `Successfully finished` 메시지가 표시될 것입니다.

사용자 설명서

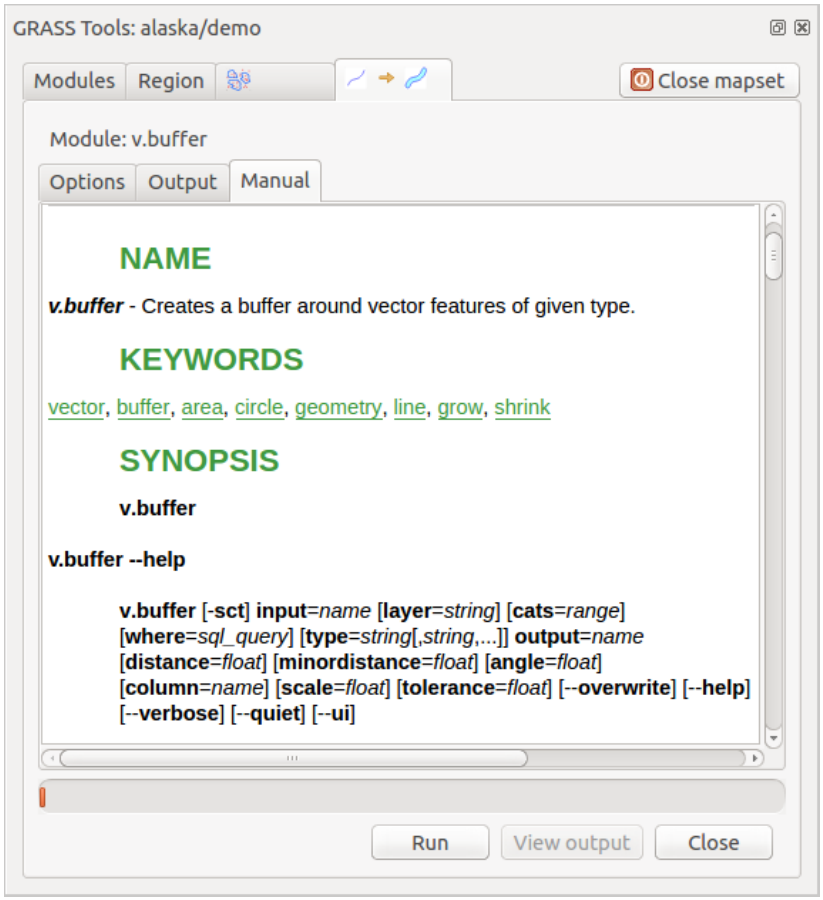


그림 22.6: GRASS 툴박스: 모듈 사용자 설명서

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.

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 명칭을 `ctour_100` 으로 입력하십시오.
- *Run* 버튼을 눌러 공간 처리 과정을 시작하십시오. 산출물 창에 `Successfully finished` 메시지가 나타낼 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 3D 효과 생성하기

표고 레이어를 표시하고 맵에 3D 효과를 주는 데 쓰이는 몇 가지 방법들이 있습니다. 앞에서 보인 등고선을 활용하는 것도 지형도 생산에 자주 쓰이는 인기 있는 방법 가운데 하나입니다. 3D 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 명칭으로 `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.

GRASS 셸 사용하기

QGIS 의 GRASS 플러그인은 GRASS 초보자로서 모든 모듈 및 옵션에 익숙하지 않은 사용자를 위해 설계돼 있습니다. 따라서, 툴박스에 있는 모듈 가운데 일부는 사용할 수 있는 모든 옵션을 표시하지 않으며, 일부는 아예 나타나지도 않습니다. GRASS 셸 (또는 콘솔) 은 사용자가 툴박스 트리에 나타나지 않는 이런 부가적인 GRASS 모듈을 사용할 수 있게 해주며, 또한 툴박스에 간소화된 기본 파라미터들만 표시되는 모듈의 부가적인 옵션들도 사용할 수 있게 해줍니다. 이번 예제에서는 앞에서 소개한 `r.shaded.relief` 모듈의 부가적인 옵션을 사용하는 법을 보여줄 것입니다.

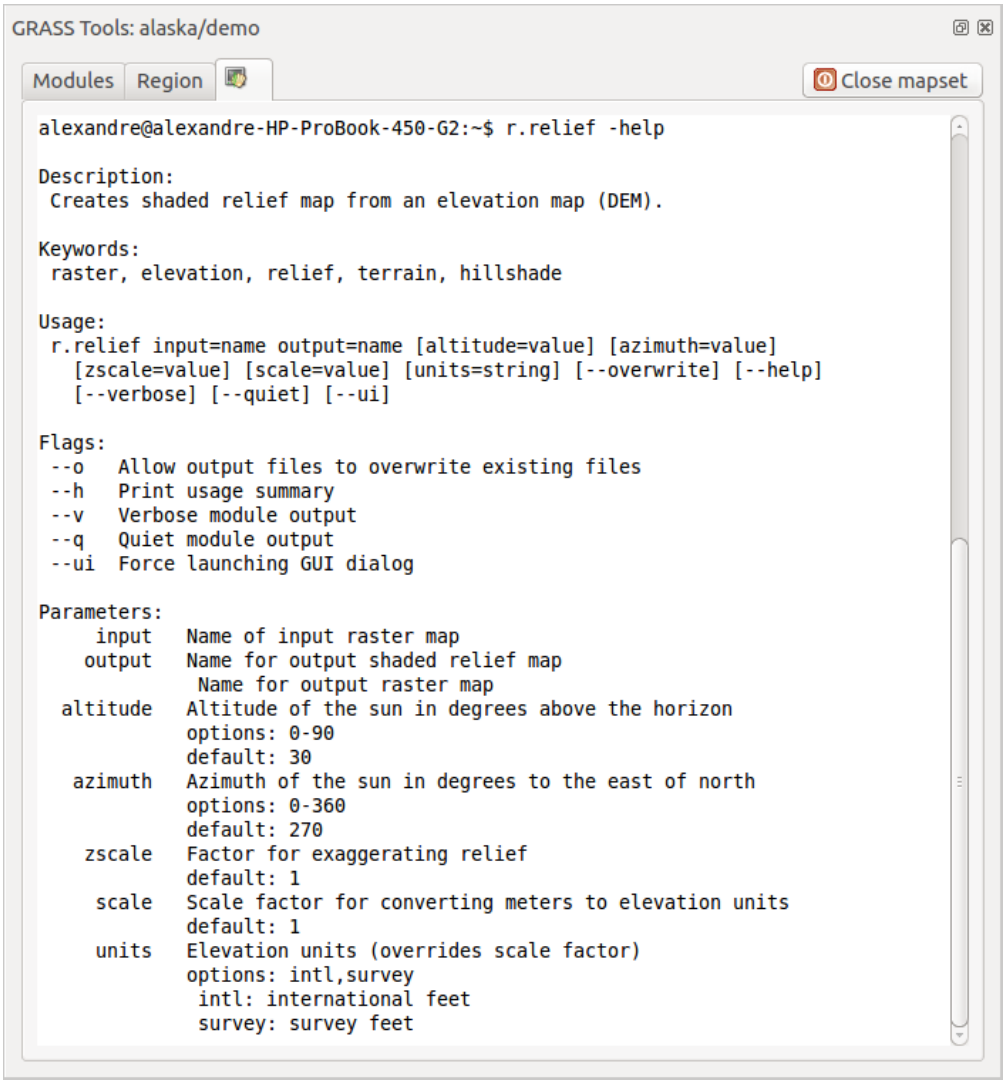


그림 22.8: GRASS 셸: *r.shaded.relief* 모듈

`r.shaded.relief` 모듈은 `zmult` 파라미터를 받을 수 있는데, 이 파라미터는 표고 값을 X/Y 좌표 단위를 기준으로 증가시켜서 음영기복 효과를 더욱 두드러지게 만듭니다.

- 앞에서와 마찬가지로 `gtopo30` 표고 래스터를 불러온 다음, GRASS 툴박스 대화창을 열고 GRASS 셸을 클릭하십시오. 셸 창에서 `r.shaded.relief map=gtopo30 shade=gtopo30_shade2 azimuth=315 zmult=3` 명령어를 입력한 다음 `Enter` 키를 누르십시오.
- 공간 처리 과정이 종료된 후, *Browse* 탭을 선택하고 새 `gtopo30_shade2` 래스터를 더블클릭해서 QGIS 맵 캔버스에 표시하십시오.

- 앞에서 설명한 대로, 범례에서 음영기복 래스터를 `gtopo30` 래스터 아래로 이동시킨 다음, 색상이 적용된 `gtopo30` 레이어의 투명도를 확인하십시오. 첫 번째 음영기복도를 사용했을 때보다 3D 효과가 더욱 두드러진다는 사실을 알 수 있을 겁니다.

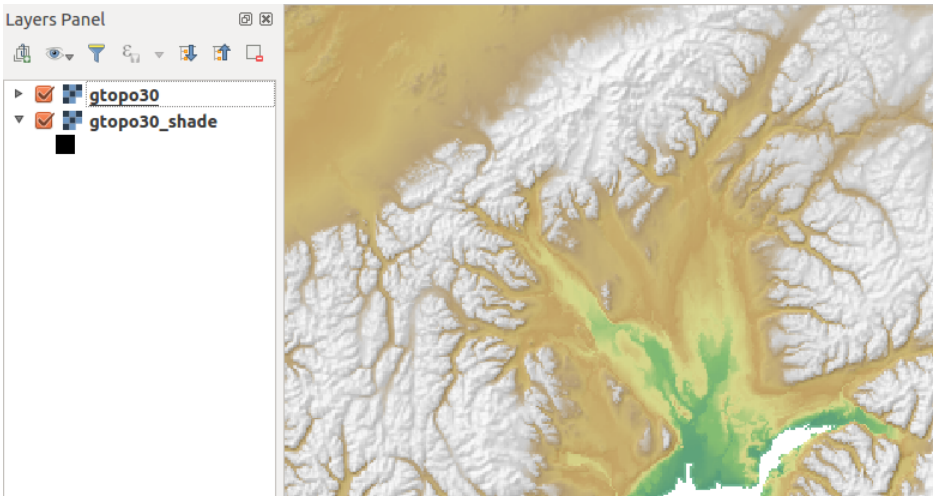


그림 22.9: GRASS *r.shaded.relief* 모듈로 생성한 음영기복도 표시하기

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 명칭을 `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>
```

사용자가 해당 모듈을 선택하면 파싱 작업자 (parser) 가 이 정의를 읽어들여 툴박스 안에 새 탭을 생성합니다. 새 모듈을 추가하는 방법, 모듈이 속한 그룹을 변경하는 방법 등에 대한 자세한 설명은 https://qgis.org/ko/site/getinvolved/development/addinggrasstools.html 웹페이지에서 찾아볼 수 있습니다.

CHAPTER 23

QGIS 공간 처리 프레임워크

23.1 소개

이 장에서는 QGIS 공간 처리 프레임워크를 소개합니다. QGIS 공간 처리 프레임워크는 QGIS 에서 내장 및 제 3 자 알고리즘을 호출하는 데 사용되는 지리 공간 처리 환경으로, 사용자가 공간 분석 업무를 더 생산적으로 더 쉽게 완수할 수 있도록 해줍니다.

핵심 플러그인 으로서, 공간 처리 프레임워크는 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. 대화 창을 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.
- *Graphical Modeler*: 모델 생성기 (modeler) 의 그래픽 인터페이스를 통해 여러 알고리즘을 결합시켜, 여러 하위 프로세스를 이용하는 단일 프로세스를 생성하는 작업 흐름을 정의합니다.

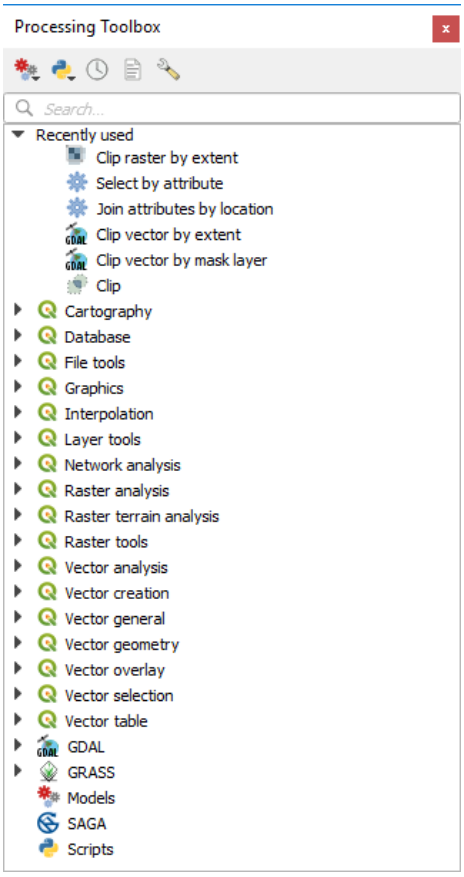


그림 23.1: 공간 처리 툴박스

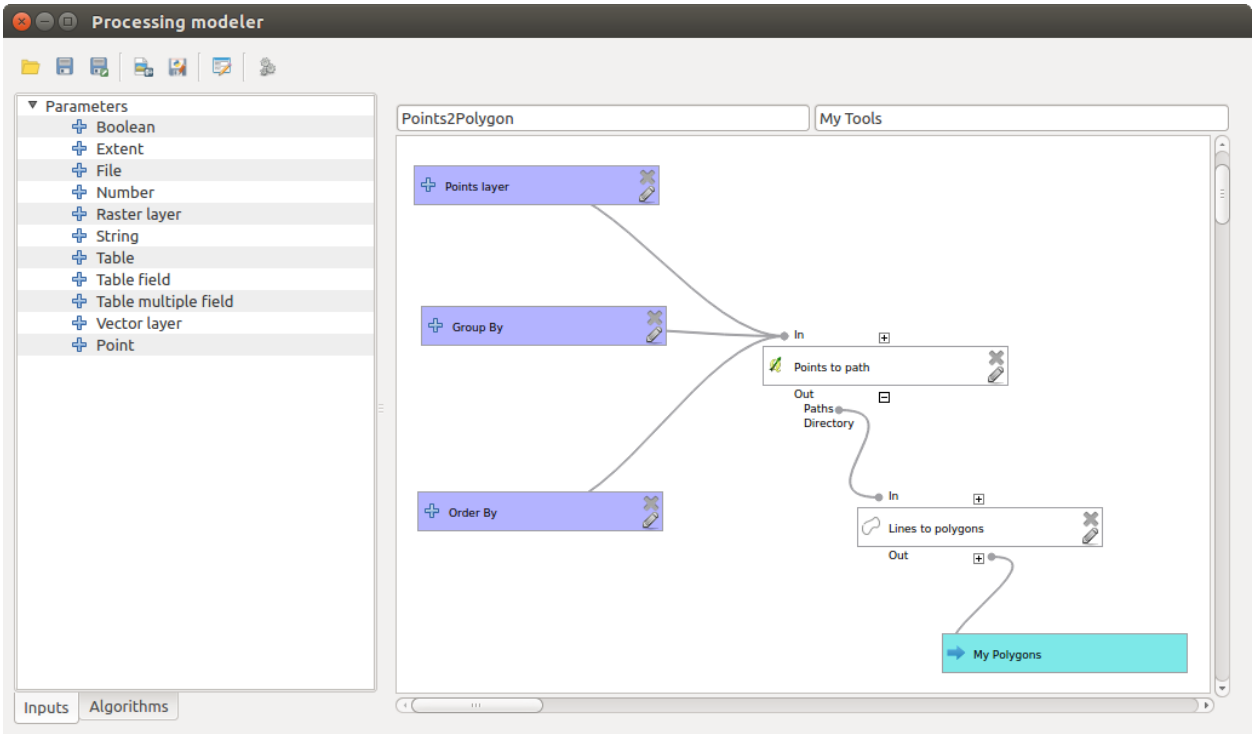


그림 23.2: 공간 처리 모델 생성기

- *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* 탭) 에서 알고리즘 작동 방식을 환경 설정할 수 있습니다. 환경 설정 파라미터들은 대화창의 왼쪽에서 선택할 수 있는 개별 그룹으로 구분돼 있습니다.

General 그룹에 몇 가지 흥미로운 파라미터가 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*: 유효하지 않은 객체를 필터링합니다.
- *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*: 산출물을 저장할 폴더를 설정합니다.

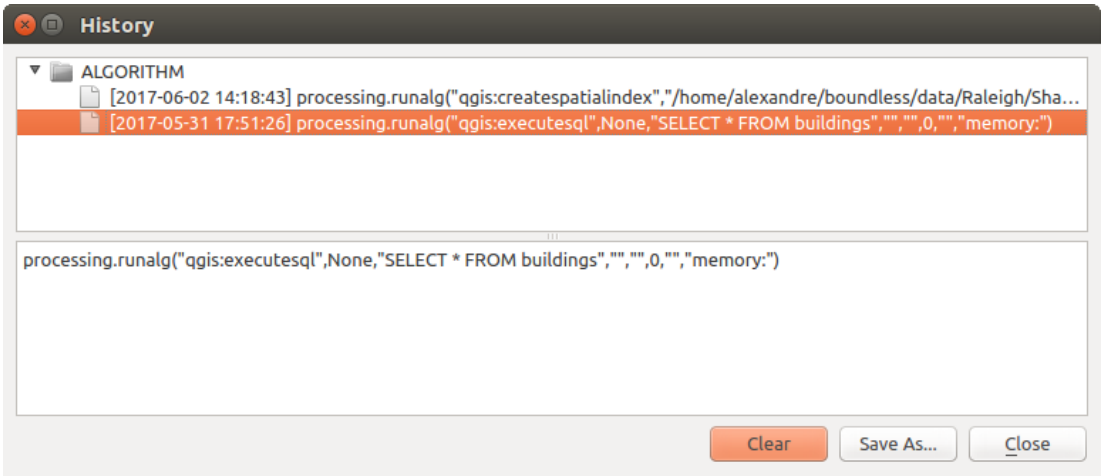


그림 23.3: 공간 처리 이력

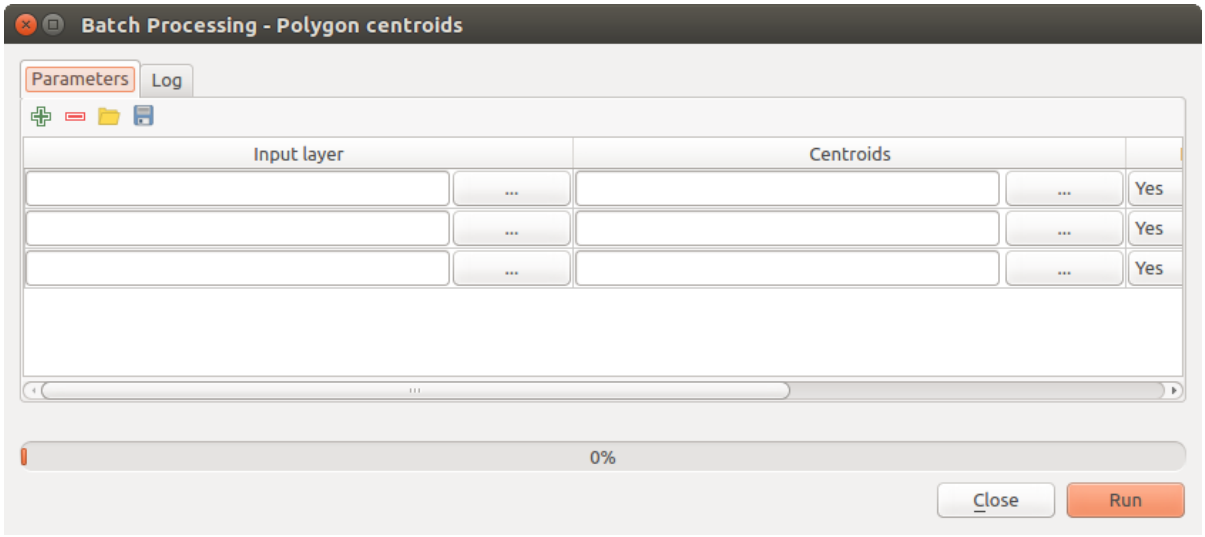


그림 23.4: 배치 공간 처리 인터페이스

- *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 명칭을 정의합니다. 몇몇 경우 고정된 명칭을 사용할 수도 있는데, 어떤 입력 레이어를 사용하든 상관없이 동일한 산출 레이어 명칭을 사용할 것이라는 뜻입니다. 다른 경우엔 입력 레이어의 명칭 또는 알고리즘 실행에 사용되는 몇몇 파라미터에 따라 산출 레이어 명칭이 달라지기도 합니다. 이 옵션을 활성화하면 대신 산출 파일명에서 레이어 명칭을 가져올 것입니다. 산출물이 임시 파일로 저장되는 경우, 이 임시 파일의 명칭은 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*: 알려진 문제가 있는 알고리즘을 표시합니다.
- *Show layer CRS definition in selection boxes*: 콤보 박스에 레이어 좌표계 정의를 표출합니다.
- *Show tooltip when there are disabled providers*: 비활성화된 제공자가 있는 경우 툴팁을 표출합니다.
- *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.

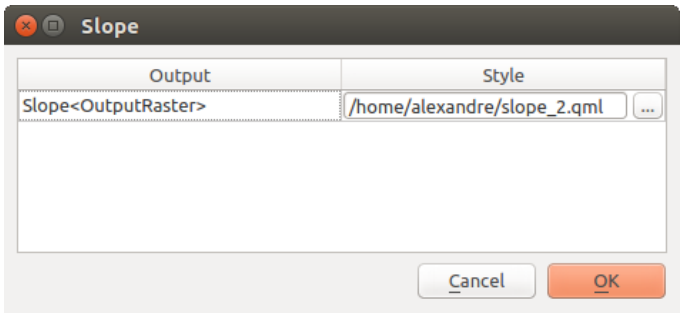


그림 23.5: 렌더링 스타일

  각 산출물별로 사용자가 원하는 스타일 파일 (`.qml`) 을 선택한 다음 *OK* 버튼을 클릭하십시오.
- *Temporary output folder path*: 임시 산출물 폴더 경로를 무시합니다.
- *Warn before executing if parameter CRS's do not match*: 파라미터 좌표계가 일치하지 않을 경우 실행 전에 경고합니다.

설정 대화창에는 알고리즘 제공자 *Providers* 그룹도 있습니다. 이 그룹의 각 항목은 해당 알고리즘을 툴박스에 표시할지 말지를 설정하는 *Activate* 옵션을 담고 있습니다. 또 알고리즘 제공자 가운데 일부는 고유의 환경 설정 옵션을 가지고 있는데, 이후 특정 알고리즘 제공자를 설명하는 부분에서 이 고유 옵션에 관해 설명하겠습니다.

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.

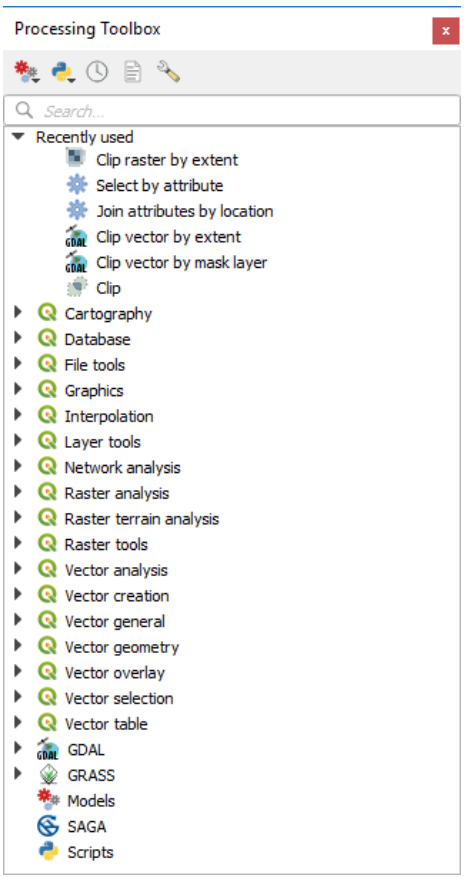


그림 23.6: 공간 처리 툴박스

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

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.

참고: 알고리즘 목록 최상단에는 가장 최근에 사용한 도구를 표시합니다. 어떤 도구를 재실행하려는 경우 유용합니다.

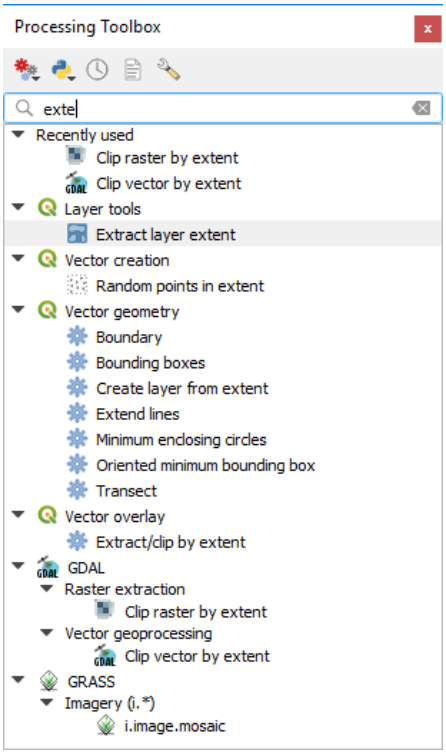


그림 23.7: 검색 결과를 표시하는 공간 처리 툴박스

어느 도구를 실행하려면, 툴박스에 있는 해당 명칭을 더블클릭하기만 하면 됩니다.

23.3.1 알고리즘 대화창

사용자가 실행하고자 하는 알고리즘의 명칭을 더블클릭하고 나면, 그림 23.8 과 비슷한 대화창이 열립니다. (이 그림은 `Centroids` 알고리즘에 대응하는 대화창입니다.)

이 대화창 왼쪽에는 탭이 2 개 (*Parameters* 와 *Log*) 있으며, 오른쪽에는 알고리즘 설명이, 하단에는 몇몇 버튼들이 있습니다.

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.

알고리즘의 특성에 따라 파라미터의 개수 및 유형이 달라지긴 해도, 대화창의 구조는 모두 비슷합니다. 대화창에 있는 파라미터는 다음 유형들 가운데 하나일 것입니다:

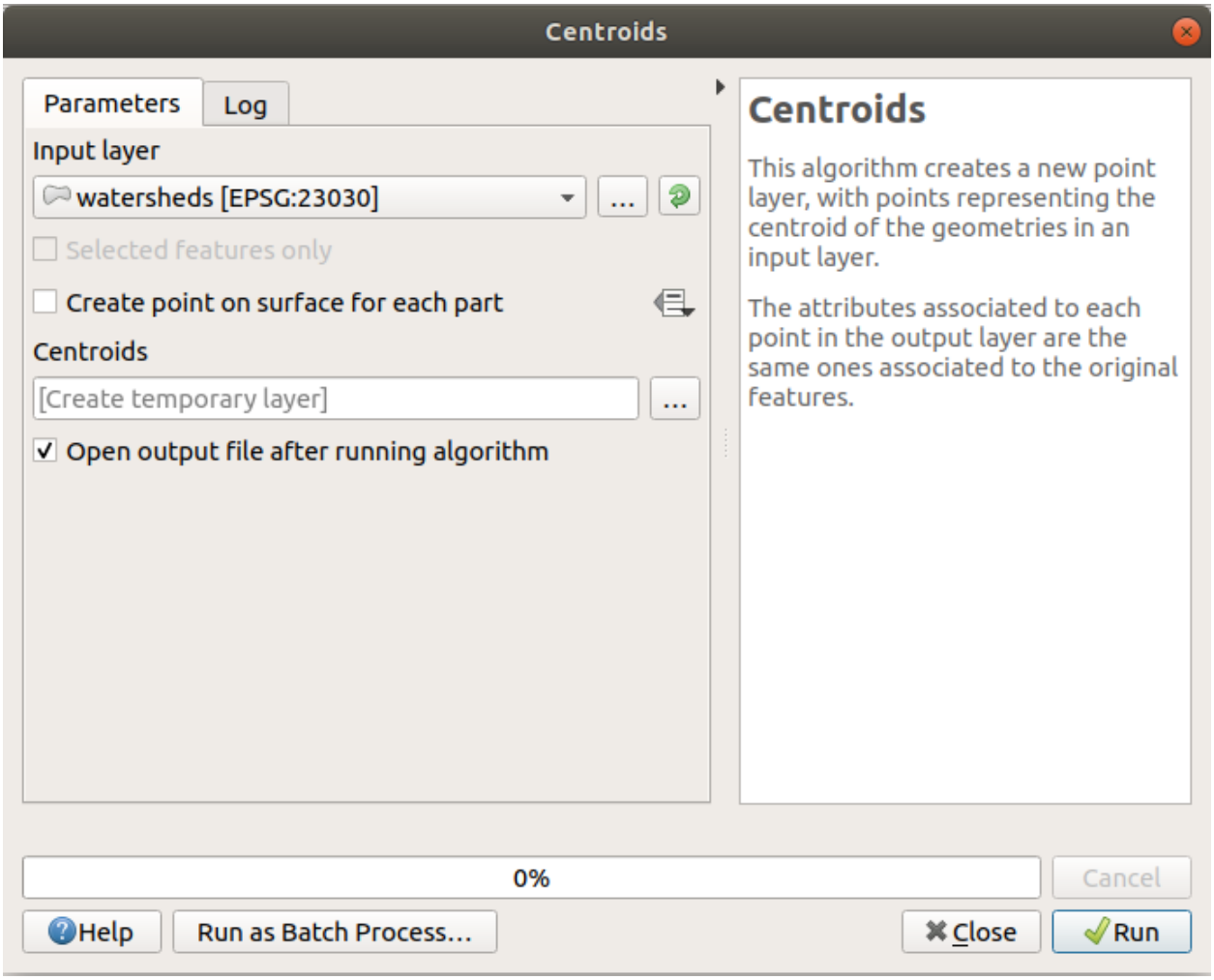


그림 23.8: 알고리즘 대화창 — 파라미터 탭

- **래스터 레이어**: QGIS 에서 (현재 열려 있는) 사용할 수 있는 모든 래스터 레이어의 목록 가운데 선택할 수 있습니다. 이 선택기의 오른쪽에 있는|browse| 버튼을 누르면 현재 QGIS 가 불러오지 않은 레이어를 나타내는 파일명을 선택할 수도 있습니다.
- **벡터 레이어**: QGIS 사용할 수 있는 모든 벡터 레이어의 목록 가운데 선택할 수 있습니다. 래스터 레이어의 경우와 마찬가지로 QGIS 가 불러오지 않은 레이어도 선택할 수 있는데, 알고리즘이 레이어의 속성 테이블에서 선택된 테이블 필드를 요구하지 않는 경우에만 선택할 수 있습니다. 알고리즘이 이런 필드를 요구하는 경우, 이미 열려 있는 레이어만 선택할 수 있습니다. 사용할 수 있는 필드명 목록을 가져오려면 먼저 레이어를 불러와야 하기 때문입니다.

  각 벡터 레이어 선택기 옆에는 다음 그림에 있는 반복기 (iterator) 버튼도 있습니다.

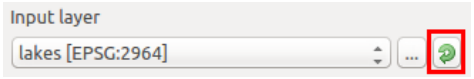


그림 23.9: 벡터 반복기 버튼

  벡터 레이어를 여러 개 선택해야 하는 알고리즘의 경우, 반복기 버튼들 가운데 하나만 켜고 끌 수 있습니다. 입력 벡터 레이어에 대응하는 버튼을 켜면, 알고리즘을 해당 레이어 전체에 대해 한 번 실행하는 대신 레이어의 모든 객체에 대해 반복 실행해서 알고리즘 실행 횟수만큼 산출물을 생성합니다. 레이어에 있는 모든 피처를 개별적으로 처리해야 하는 경우 해당 프로세스를 이 버튼으로 자동화할 수 있습니다.

참고: 기본적으로, 파라미터 탭은 각 레이어의 명칭과 함께 좌표계 설명도 표시할 것입니다. 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.

- **테이블**: QGIS 에서 사용할 수 있는 모든 테이블의 목록 가운데 선택할 수 있습니다. QGIS 는 비공간 테이블도 벡터 레이어처럼 불러오며, 사실 벡터 레이어처럼 취급합니다. 현재, 테이블이 필요한 알고리즘 실행 시 표시되는 사용할 수 있는 테이블의 목록은 dBase(`.dbf`) 또는 쉼표 구분 값 (`.csv`) 포맷의 파일에 있는 테이블로 제한되어 있습니다.
- **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.
- **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|browse| 버튼을 누르면 열리는좌표계 선택기 에서 선택할 수 있습니다.
- **범위 (extent)**: `xmin, xmax, ymin, ymax` 서식으로 된 모서리 좌표를 통해 직사각형을 정의하는 텍스트 란입니다. 값 선택기 오른쪽에 있는|browse| 버튼을 클릭하면, 다음 세 가지 옵션을 제공하는 팝업 메뉴가 나타납니다:
  - *Calculate from layer*: 불러온 레이어들 가운데 선택한 레이어의 경계 상자 좌표로 텍스트 란을 채웁니다.
  - *Use map canvas extent*: 맵 캔버스 범위의 좌표로 텍스트 란을 채웁니다.

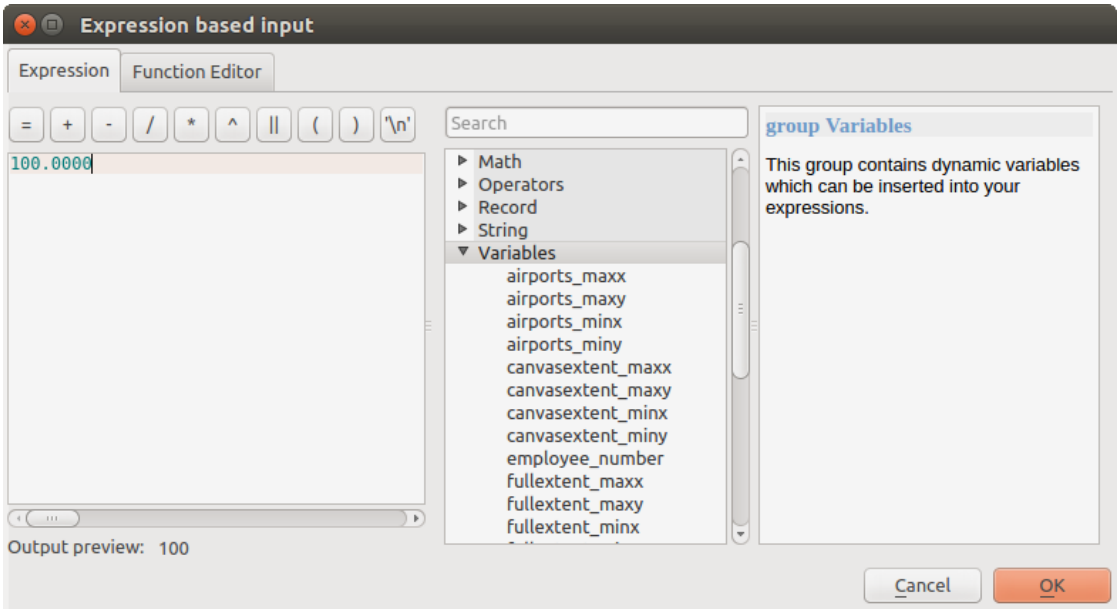


그림 23.10: 표현식 기반 입력

– *Draw on canvas*: 파라미터 대화창이 사라져서 캔버스 상에서 직사각형을 클릭 & 드래그할 수 있게 됩니다. 직사각형 영역을 정의하고 나면, 대화창이 범위 텍스트 란에 값을 담은 채 다시 나타납니다.

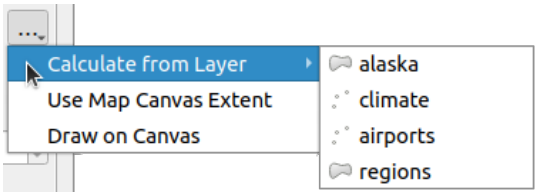


그림 23.11: 범위 선택기

- **항목 목록**: (래스터 레이어든 벡터 레이어든 또는 테이블이든 필드든) 해당 유형의 목록 가운데 선택할 수 있습니다. 옵션 왼쪽에 있는|browse| 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.

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.

Log 라는 또다른 탭이 *Parameters* 탭 옆에 있습니다. (다음 그림 23.14 참조) 이 탭은 알고리즘 실행 도중 알고리즘이 제공하는 정보를 표시해서 사용자가 그 과정을 추적하고 더 자세한 정보를 알 수 있게 해줍니다. 알고리즘 실행 정보는 *View* → *Panels* → *Log Messages Panel* 에도 산출됩니다.

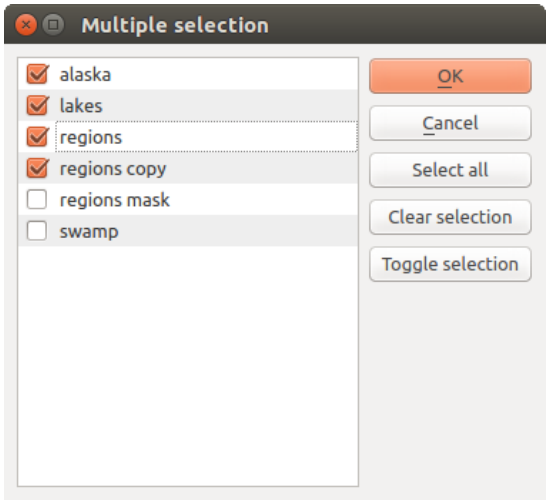


그림 23.12: 다중 선택

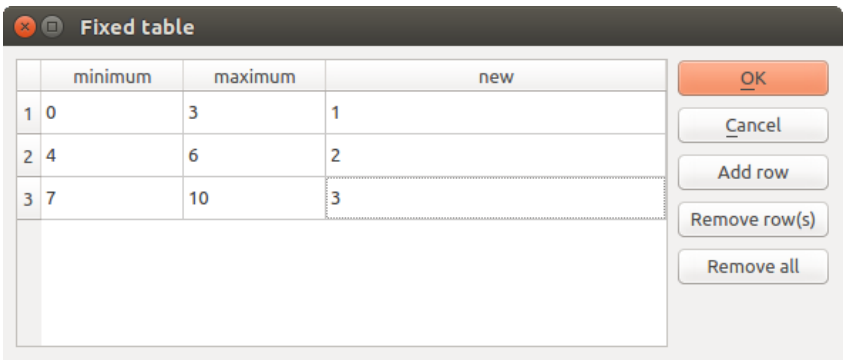


그림 23.13: 고정 테이블

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.

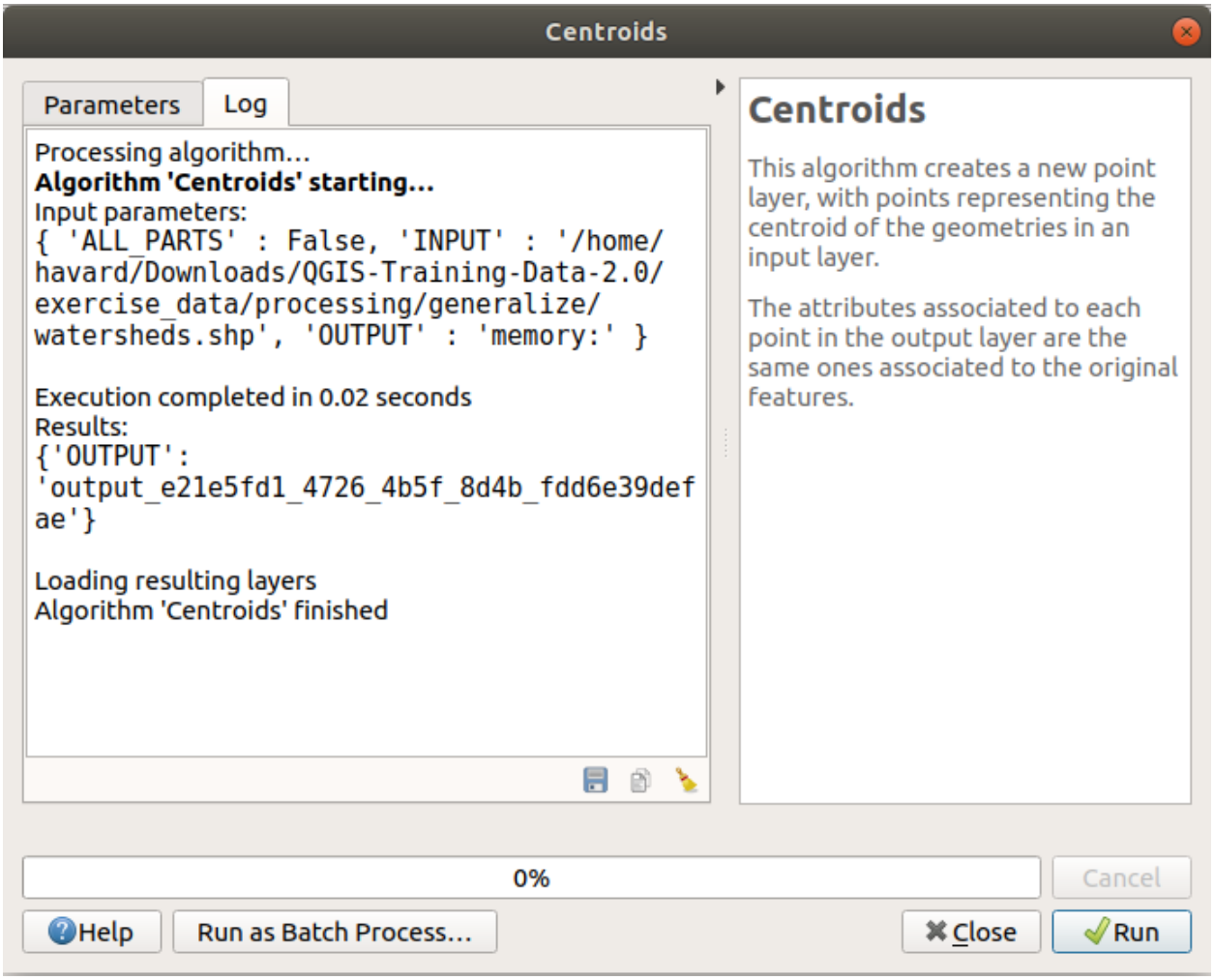


그림 23.14: 알고리즘 대화창 — 로그 탭

Log 탭 하단에 *Save Log to File*, *Copy Log to Clipboard* 및 *Clear Log* 버튼이 있습니다. 공간 처리 옵션 대화창의 *General* 그룹에서|checkbox| *Keep dialog open after running algorithm* 옵션을 활성화한 경우 이 도구들이 특히 유용합니다.

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.

Run as batch process 버튼을 누르면배치 프로세스 모드 를 촉발합니다. 이 모드에서 알고리즘의 중복된 인스턴스들을 서로 다른 파라미터로 환경 설정하고 실행할 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 명칭과 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

알고리즘이 생성한 데이터 객체는 다음 유형들 가운데 하나가 될 것입니다:

- 래스터 레이어
- 벡터 레이어
- 테이블
- HTML 파일 (텍스트 및 그래픽 산출물 용)

이런 데이터 객체들은 모두 디스크에 저장되며, 파라미터 탭에는 이런 각각의 산출물에 대응하는 텍스트란이 표시되어 사용자가 저장을 위한 산출물 채널을 입력할 수 있습니다. 산출물 채널은 산출되는 객체를 저장하기 위해 필요한 정보를 담고 있습니다. 거의 대부분은 파일로 저장되지만, 내장 알고리즘이 (외부 응용 프로그램을 사용하지 않는 알고리즘이) 생성한 벡터 레이어의 경우 PostGIS 또는 SpatiaLite 데이터베이스나 메모리 레이어로도 저장할 수 있습니다.

산출물 채널을 선택하려면, 텍스트 란 오른쪽에 있는|browse| 버튼을 누르기만 하면 됩니다. 사용 가능한 옵션을 표시한 작은 컨텍스트 메뉴가 나타날 것입니다.

거의 대부분의 경우, '파일로 저장하기'를 선택하게 됩니다. 해당 옵션을 선택했다면, 원하는 파일 경로를 선택할 수 있는 파일 저장 대화창이 열릴 것입니다. 대화창의 파일 포맷 선택기에 산출물 유형 및 실행 알고리즘에 따라 지원되는 파일 확장자가 표시됩니다.

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.

산출물 텍스트란에 어떤 파일명도 입력하지 않을 경우 (또는 컨텍스트 메뉴에 있는 해당 옵션을 선택하지 않는 경우) 산출물을 해당 기본 파일 포맷의임시 파일 로 저장하고, QGIS 종료시 삭제할 것입니다. (임시 레이어를 담고 있는 프로젝트를 저장하는 경우 이를 처리하십시오.)

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.

벡터 레이어를 반복 모드로 사용하는 알고리즘 실행 시, 입력한 파일 경로를 모든 생성 파일을 저장하는 기본 경로로 사용합니다. 즉 파일 경로의 파일명을 기본 명칭으로 삼아 반복 작업 차례대로 번호를 붙일 것이라는 뜻입니다. 이런 모든 생성 파일의 포맷도 파일 경로의 파일 확장자 (포맷) 를 따릅니다.

일부 알고리즘은 래스터 레이어 및 테이블 말고도, 그래픽 및 텍스트를 HTML 파일 포맷으로 생성할 수 있습니다. 이런 산출물은 알고리즘 실행 완료시 새 대화창에 표시됩니다. 이 대화창은 현재 세션 동안은 어느 알고리즘이 생성한 산출물이든 계속 가지고 있기 때문에, 언제든지 QGIS 메인 메뉴에서 *Processing* → *Results Viewer* 메뉴 옵션을 선택하면 볼 수 있습니다.

일부 외부 응용 프로그램이 (특정 확장자 제약 없이) 파일을 산출할 수도 있지만, 이런 파일들은 앞에서 설명한 어느 범주에도 속하지 않습니다. QGIS 는 이런 산출물 파일을 처리하지 않는데 (현재 QGIS 프로젝트에 불러오거나 포함시키지 않는데) 왜냐하면 대부분의 경우 이런 산출물 파일의 포맷이 QGIS 가 지원하는 포맷 또는 항목이 아니기 때문입니다. 예를 들면, LiDAR 데이터 용 LAS 파일이 그렇습니다. 해당 파일을 생성할 수는 있어도, 현재 실행 중인 QGIS 세션에 아무것도 새로 추가되지 않습니다.

다른 모든 산출물 유형의 경우, 알고리즘에게 파일을 생성한 다음 해당 산출물을 불러올지 여부를 알려줄 수 있는 체크박스가 존재합니다. 기본적으로, 모든 파일을 불러오도록 설정돼 있습니다.

산출물 선택은 지원하지 않습니다. 즉 모든 산출물을 생성한다는 뜻입니다. 하지만, 어떤 산출물을 저장하는 데 관심이 없을 경우 해당하는 체크박스를 체크 해제할 수 있는데, 본질적으로 산출물을 선택한 것처럼 작동합니다. (즉, 어쨌든 레이어를 생성하긴 하지만 텍스트 란을 비워놓을 경우 임시 파일로 저장해서 QGIS 종료 시 삭제할 것이라는 뜻입니다.)

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.

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 대화창에서 QGIS 공간 처리 알고리즘 및 스크립트의 하부 구조 (infrastructure) 도 간단하게 통합해서 점검할 수 있습니다. 어느 항목을 오른쪽 클릭하고 *Create Test…*메뉴를 선택하면, https://github.com/qgis/QGIS/blob/release-3_16/python/plugins/processing/tests/README.md 에 있는 지침에 따라 해당 알고리즘 및 파라미터를 사용해서 점검을 시작합니다.

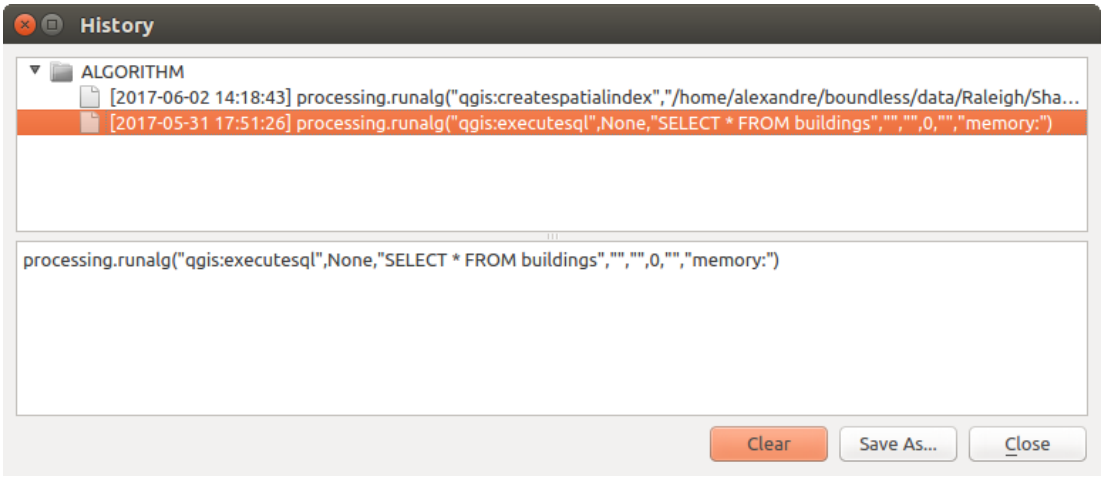


그림 23.15: 이력

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 그래픽 모델 생성기

그래픽 모델 생성기 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. 그래픽 모델 생성기를 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.

Processing 메뉴에서 (*Processing → Graphical Modeler*) 모델 생성기를 열 수 있습니다.

모델 생성기는 모델의 구조 및 모델이 나타내는 작업 흐름을 표시하는 작업 캔버스를 보유하고 있습니다. 대화창 좌측 부분에 모델에 새 항목을 추가할 수 있는 5 개의 패널이 있습니다:

1. `Model Properties`: 모델 및 모델을 담게 될 그룹의 이름을 지정할 수 있습니다.
2. `Inputs`: 사용자 모델을 형성할 모든 입력물
3. `Algorithms`: 사용할 수 있는 공간 처리 알고리즘
4. `Variables`: 공간 처리 모델 생성기에서만 사용할 수 있는 변수도 정의할 수 있습니다.
5. `Undo History`: 이 패널에 모델 생성기에서 일어나는 모든 일들이 등록될 것입니다. 즉 여기에서 사용자가 실수한 내용을 쉽게 취소할 수 있습니다.

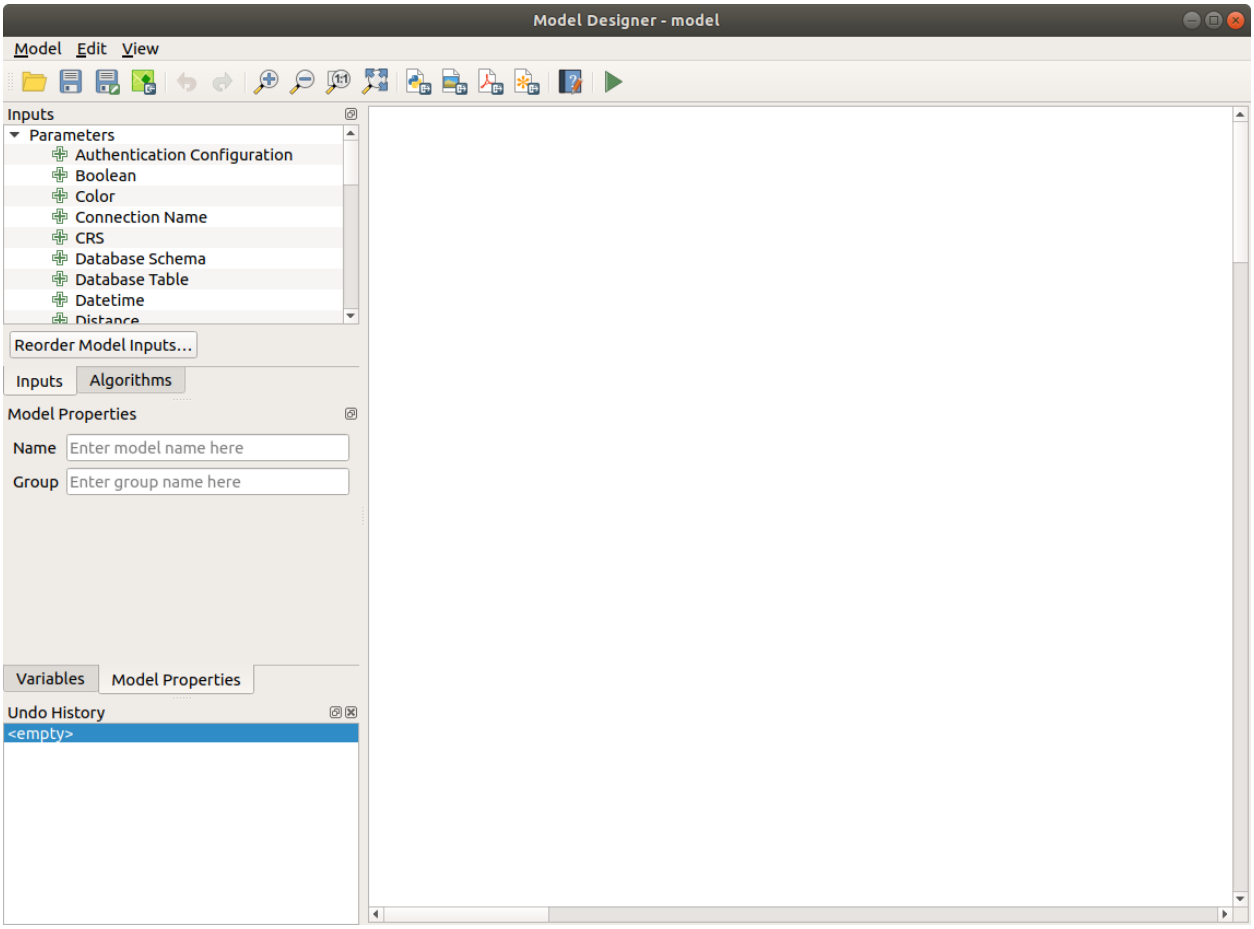


그림 23.16: 모델 생성기 대화창

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.

23.5.1 입력물 정의

모델 생성의 첫 단계는 모델에 필요한 입력물을 정의하는 것입니다. 모델 생성기 대화창 좌측에 있는 *Inputs* 패널에서 다음 항목들을 볼 수 있습니다:

- 인증 환경 설정
- 불 (Boolean)
- 색상
- 연결명
- 좌표 연산
- 좌표계
- 데이터베이스 스키마
- 데이터베이스 테이블
- 날짜 & 시간
- 거리
- 집합 (Enum)
- 표현식
- 범위 (Extent)
- 필드 집계
- 필드 맵퍼
- 파일/폴더
- 도형
- 맵 레이어
- 맵 테마
- 매트릭스
- 메시 레이어
- 다중 입력
- 숫자
- 포인트
- 인쇄 조판기
- 인쇄 조판 항목

- 범위 (Range)
- 래스터 밴드
- 래스터 레이어
- 축척
- 문자열
- TIN 생성 레이어
- 벡터 피처
- 벡터 필드
- 벡터 레이어
- 벡터 타일 작성자 레이어

참고: 입력물 위로 마우스를 가져가면 추가적인 정보를 가진 도움말 (tooltip) 을 표시할 것입니다.

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 유효값의 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.

`Comments` 탭에서 파라미터를 더 자세히 설명하기 위해 입력물에 더 많은 정보를 태그할 수 있습니다. 이런 코멘트는 모델 생성기 캔버스에서만 보이고 최종 알고리즘 대화창에서는 보이지 않습니다.

입력물을 추가할 때마다, 모델 생성기 캔버스에 새 항목이 추가됩니다.

파라미터 목록에서 입력물 유형을 모델 생성기 캔버스 상의 원하는 위치로 드래그해서 입력물을 추가할 수도 있습니다. 기존 입력물의 파라미터를 변경하고 싶다면, 그냥 입력물을 더블클릭하기만 하면 동일한 대화창이 뜰 것입니다.

23.5.2 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 패널에서 알고리즘을 찾을 수 있습니다. 공간 처리 툴박스와 거의 동일한 방식으로 알고리즘이 그룹으로 정리돼 있습니다.

모델에 알고리즘을 추가하려면, 입력물을 추가할 때와 마찬가지로 알고리즘 명칭을 더블 클릭하거나 모델 생성기 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.

두 대화창에 몇몇 차이가 있다는 것을 알 수 있습니다.

알고리즘 **입력물** 을 정의할 때 다음 네 가지 방식 가운데 하나를 선택할 수 있습니다:

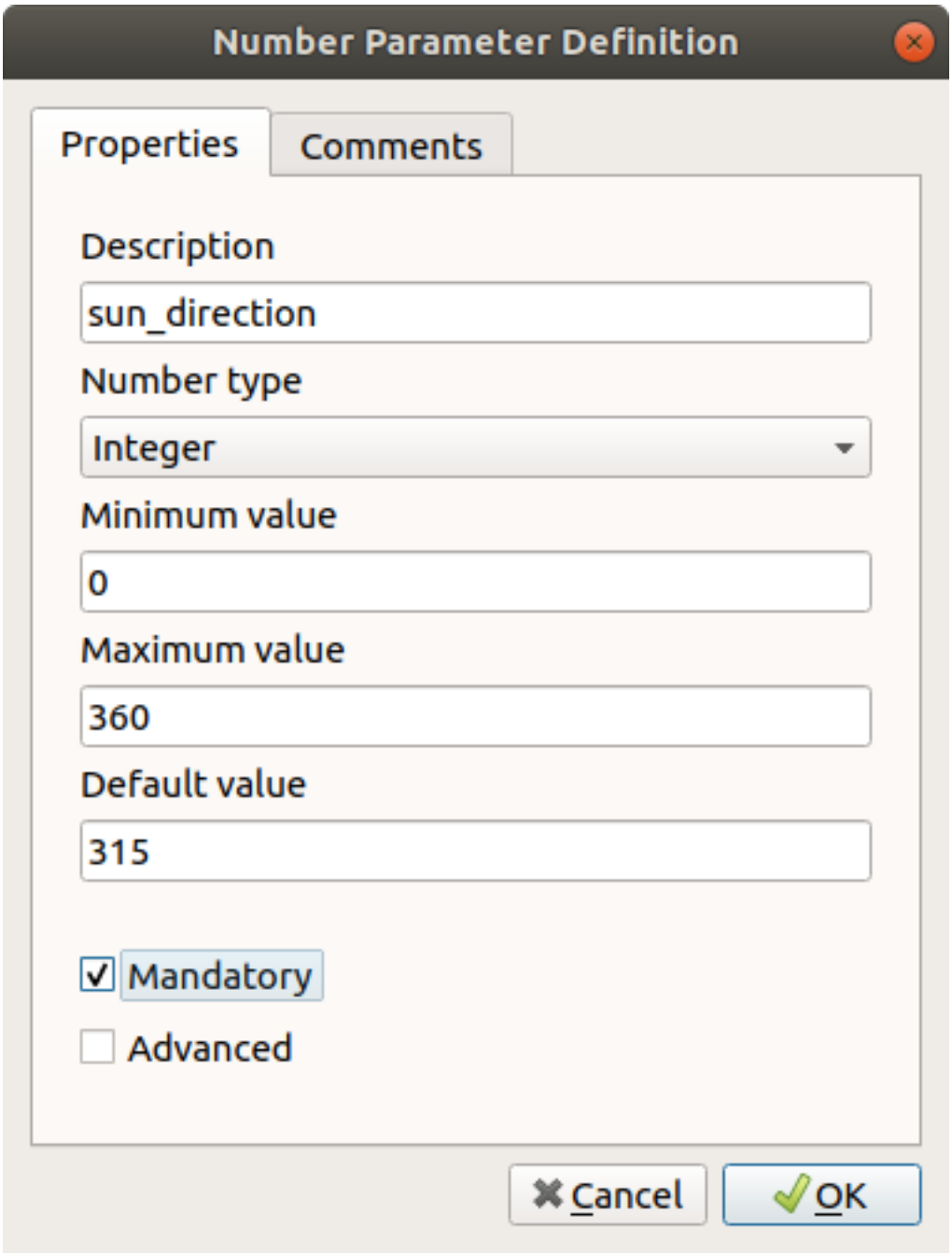


그림 23.17: 모델 파라미터 정의하기



그림 23.18: 캔버스에 표시되는 모델 파라미터

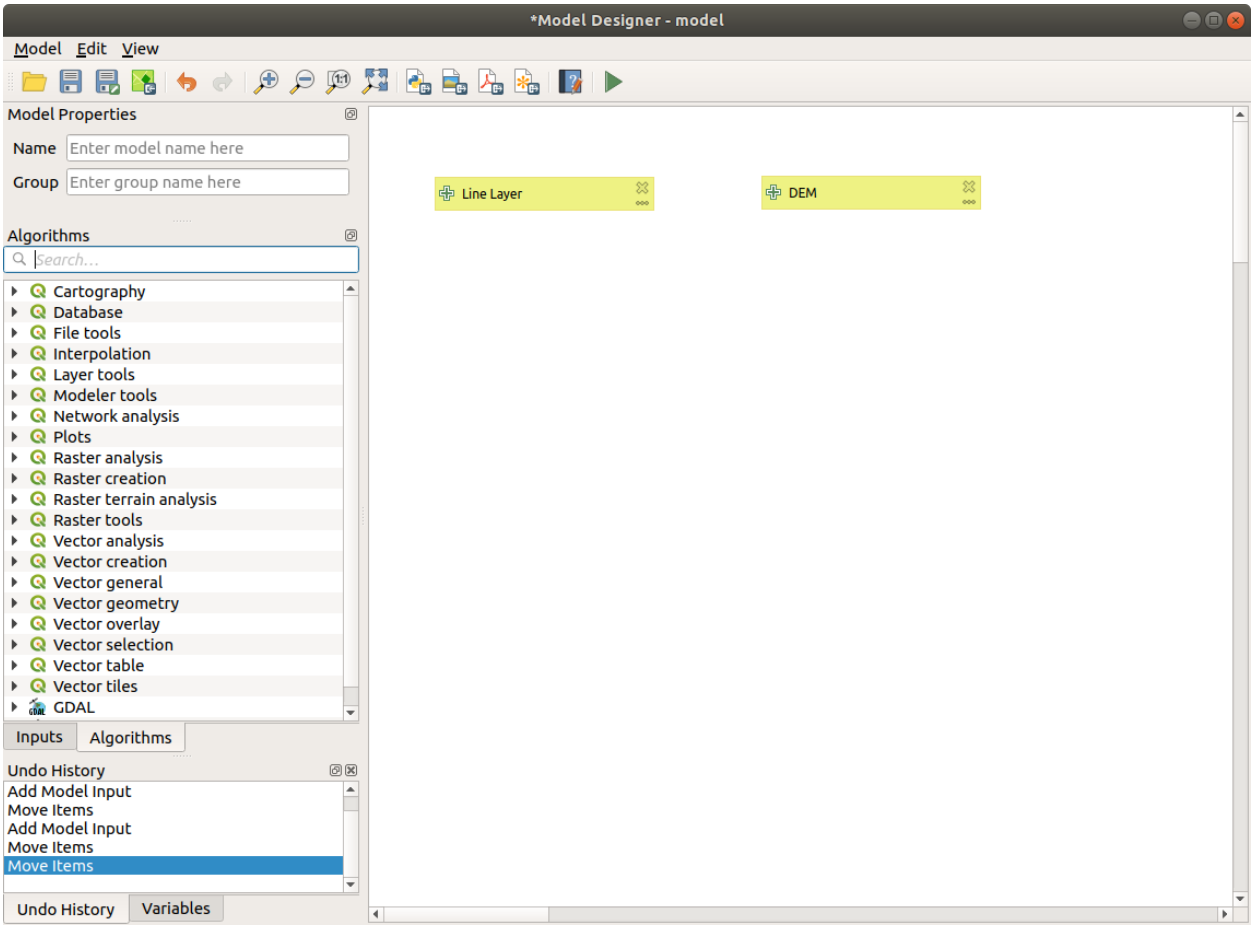


그림 23.19: 모델 입력물

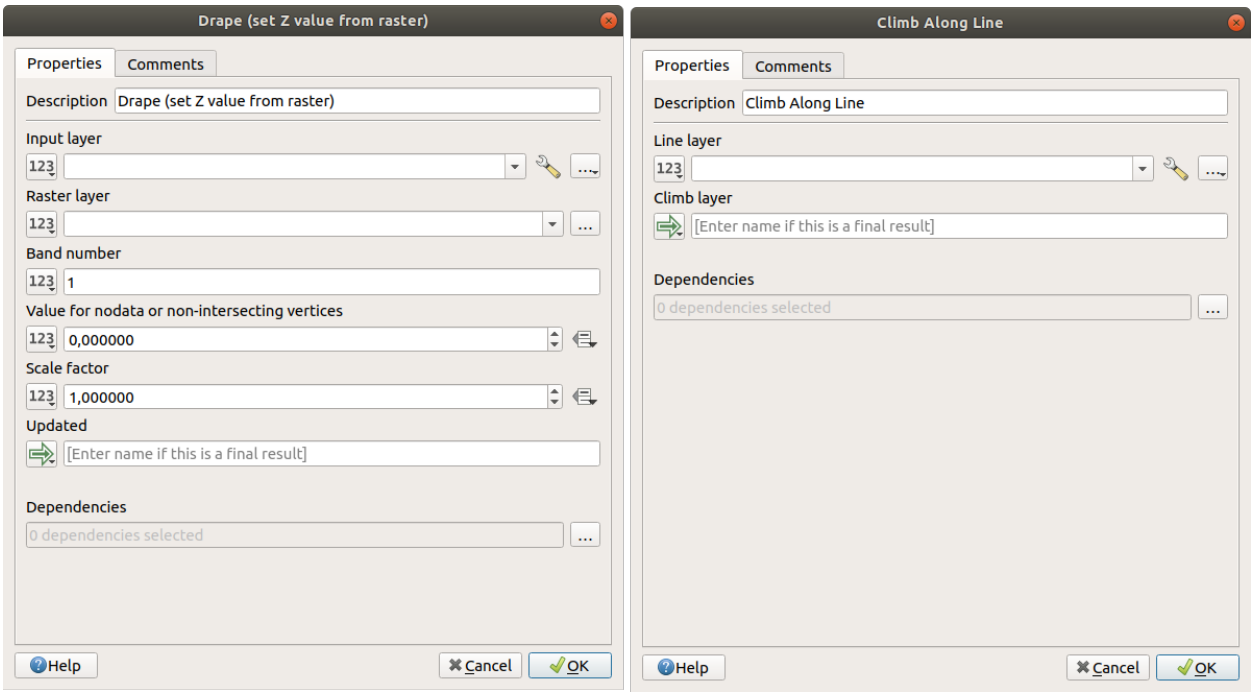


그림 23.20: 모델 알고리즘 파라미터

- Value: QGIS 프로젝트에 불러온 레이어에서 또는 폴더에서 레이어를 탐색해서 파라미터를 설정할 수 있습니다.
- Pre-calculated Value: 이 옵션을 사용하면 표현식 작성기를 열어 파라미터를 채우기 위한 사용자 고유의 표현식을 정의할 수 있습니다. 몇몇 다른 레이어의 통계를 가진, 사용할 수 있는 모델 입력물이 표현식 작성기의 검색 대화창 최상단에 **변수** 로 목록화되어 있습니다.
- Model Input: 사용자가 정의했던 모델의 입력물에서 파라미터를 가져오는 경우 이 옵션을 선택하십시오. 한번 클릭하기만 하면, 이 옵션이 해당 파라미터에 대해 적합한 모든 입력물의 목록을 표시할 것입니다.
- Algorithm Output: 이 옵션은 알고리즘의 입력 파라미터가 또다른 알고리즘의 산출물인 경우 유용합니다.

알고리즘 **산출물** 은 알고리즘의 산출물을 모델에서 사용할 수 있도록 하는 추가적인 Model Output 옵션을 가지고 있습니다.

알고리즘이 생성한 레이어가 또다른 알고리즘의 입력물로만 사용되는 경우, 해당 텍스트란을 편집하지 마십시오.

다음 그림에서 두 입력 파라미터가 Model Input 과 임시 산출 레이어로 정의된 것을 볼 수 있습니다:

이 모든 경우에, 툴박스에서 알고리즘을 호출할 때에는 볼 수 없었던 *Dependencies* 라는 추가 파라미터를 볼 수 있습니다. 이 파라미터는 한 알고리즘을 현재 알고리즘의 부모 로 명확하게 정의해서 알고리즘 실행 순서를 정의할 수 있습니다. 이렇게 하면 부모 알고리즘이 현재 알고리즘 이전에 실행되도록 강제할 것입니다.

이전 알고리즘의 산출물을 현재 알고리즘의 입력물로 사용하면, 암묵적으로 이전 알고리즘을 현재 알고리즘의 부모로 설정하는 것입니다. (그리고 모델 생성기 캔버스에도 그에 대응하는 화살표를 배치합니다.) 하지만, 어떤 경우 (예를 들면, 한 알고리즘은 PostGIS 데이터베이스에 대해 SQL 문을 실행하고 다른 알고리즘은 동일 데이터베이스로 레이어를 가져오는 경우) 현재 알고리즘이 또다른 알고리즘에 — 이 알고리즘의 어떤 산출 객체도 사용하지 않더라도 — 종속하고 있을 수도 있습니다. 이런 경우에는 그냥 *Dependencies* 파라미터에서 이전 알고리즘을 선택만 해주면 두 알고리즘이 정확한 순서로 실행될 것입니다.

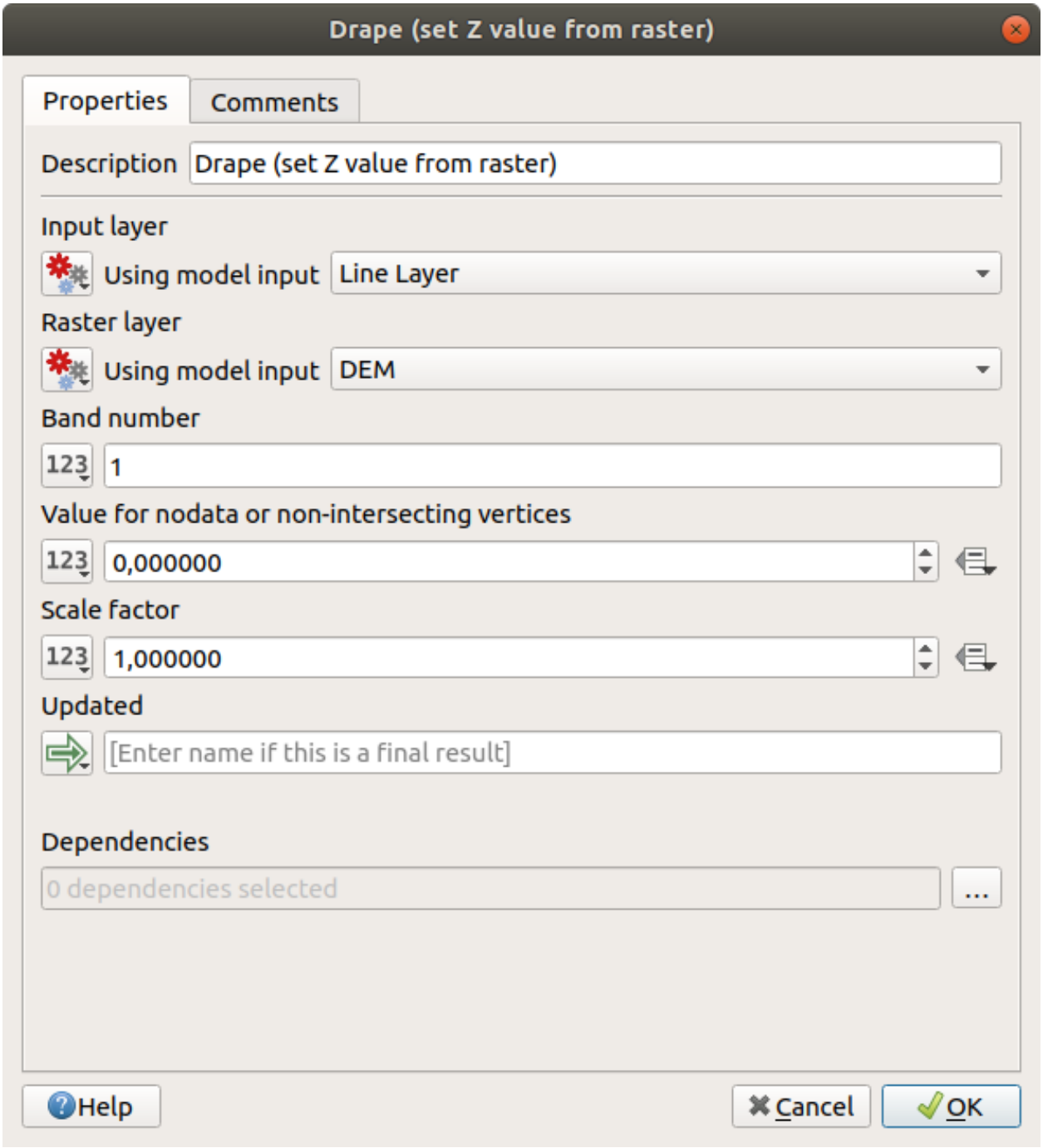


그림 23.21: 알고리즘 입력 및 출력 파라미터

모든 파라미터에 유효한 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

캔버스 안에서 항목을 드래그해서 위치를 변경할 수 있습니다. 모델 구조를 좀 더 명확하고 직관적으로 표현하는 데 유용합니다. 항목의 크기도 조정할 수 있습니다. 이 기능은 입력물 또는 알고리즘의 설명이 긴 경우 특히 유용합니다.

항목들 사이의 연결은 자동적으로 업데이트되며, 각 알고리즘의 상단 및 하단에 더하기 버튼을 볼 수 있습니다. 이 버튼을 클릭하면 알고리즘의 모든 입력물과 출력물의 목록이 나타나기 때문에, 알고리즘의 개요를 한 눈에 볼 수 있습니다.

마우스 휠을 사용해서 캔버스를 확대 및 축소할 수도 있습니다.

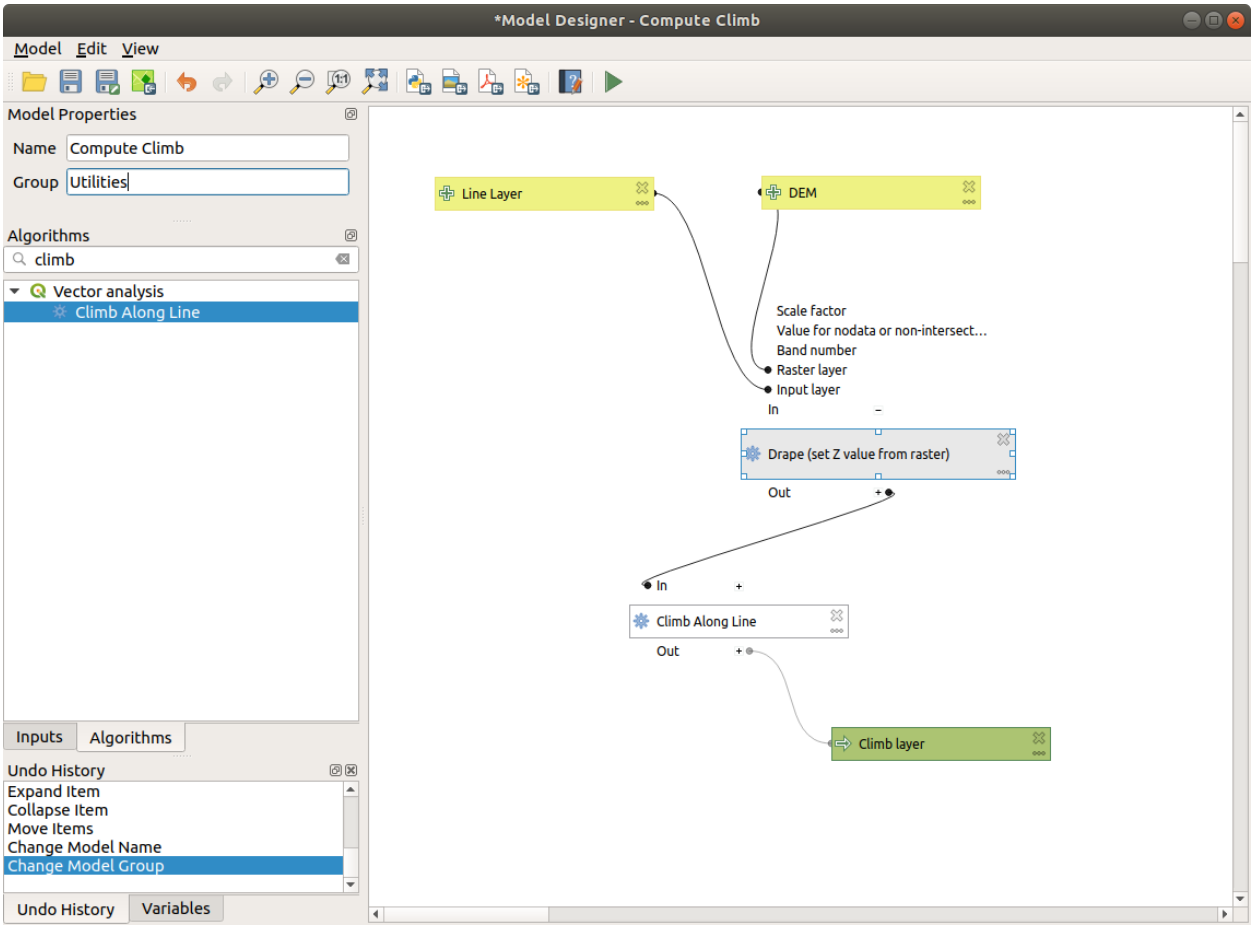


그림 23.22: 완성 모델

버튼을 클릭하면 언제라도 사용자의 알고리즘 모델을 실행할 수 있습니다. 하지만, 툴박스에서 이 모델을 사용하려면 툴박스가 그 내용을 새로고침할 수 있도록 모델을 저장하고 모델 생성기 대화창을 종료해야 합니다.

23.5.3 캔버스 및 항목과 대화형 작업

 , 버튼을 사용해서 모델 생성기 캔버스를 확대/축소할 수 있습니다. 이 버튼들은 기본적으로 메인 QGIS 툴바와 동일하게 작동합니다.

`Undo History` 패널의  및  버튼은 이전 상황으로 빠르게 롤백하는 데 매우 유용합니다. `Undo History` 패널은 사용자가 워크플로를 생성할 당시 작업했던 모든 내용의 목록을 가지고 있습니다.

먼저 항목들을 선택한 다음 마우스를 드래그하면 동시에 여러 항목들을 이동시키거나 크기를 조정할 수 있습니다.

캔버스에서 항목을 이동시키는 도중에 스냅하고 싶을 경우, *View* ▷ *Enable Snapping* 을 선택하면 됩니다.

Edit 메뉴는 사용자의 모델 항목과 대화형 작업할 수 있는 매우 유용한 옵션들을 담고 있습니다:

-  Select All: 모델의 모든 항목을 선택
- `Snap Selected Components to Grid`: 그리드에 항목을 스냅하고 정렬
-  Undo: 마지막 액션을 취소
-  Redo: 마지막 액션을 재실행
-  Cut: 선택한 항목을 잘라내기
-  Copy: 선택한 항목을 복사
-  Paste: 항목을 붙여넣기
-  Delete Selected Components: 모델에서 선택한 모든 항목을 삭제
- `Add Group Box`: 캔버스에 드래그할 수 있는 상자 를 추가합니다. 이 기능은 큰 모델의 모델 생성기 캔버스에 있는 항목들을 그룹화해서 워크플로를 지저분하지 않게 유지하는 데 매우 유용합니다. 예를 들면 예시의 모든 입력물을 함께 그룹화할 수도 있습니다:

상자의 이름과 색상을 변경할 수 있습니다. *View* ▷ *Zoom To* 와 함께 사용할 경우 그룹 상자가 매우 유용합니다. 함께 사용하면 모델의 특정 부분을 확대할 수 있게 해줍니다.

입력물의 순서 및 메인 모델 대화창에 목록화된 순서를 변경하고 싶을 수도 있습니다. `Input` 패널 하단에 `Reorder Model Inputs...` 버튼이 있는데, 이 버튼을 클릭하면 입력물의 순서를 변경할 수 있는 새 대화창이 뜹니다.

23.5.4 모델 저장 및 불러오기

 Save model 버튼을 클릭하면 현재 모델을 저장하고,  Open Model 버튼을 누르면 이전에 저장한 모델을 불러올 수 있습니다. 모델은 `.model3` 확장자로 저장됩니다. 모델 생성기 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 명칭 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* 메뉴를 통해 프로젝트 모델을 사용할 수 있습니다.

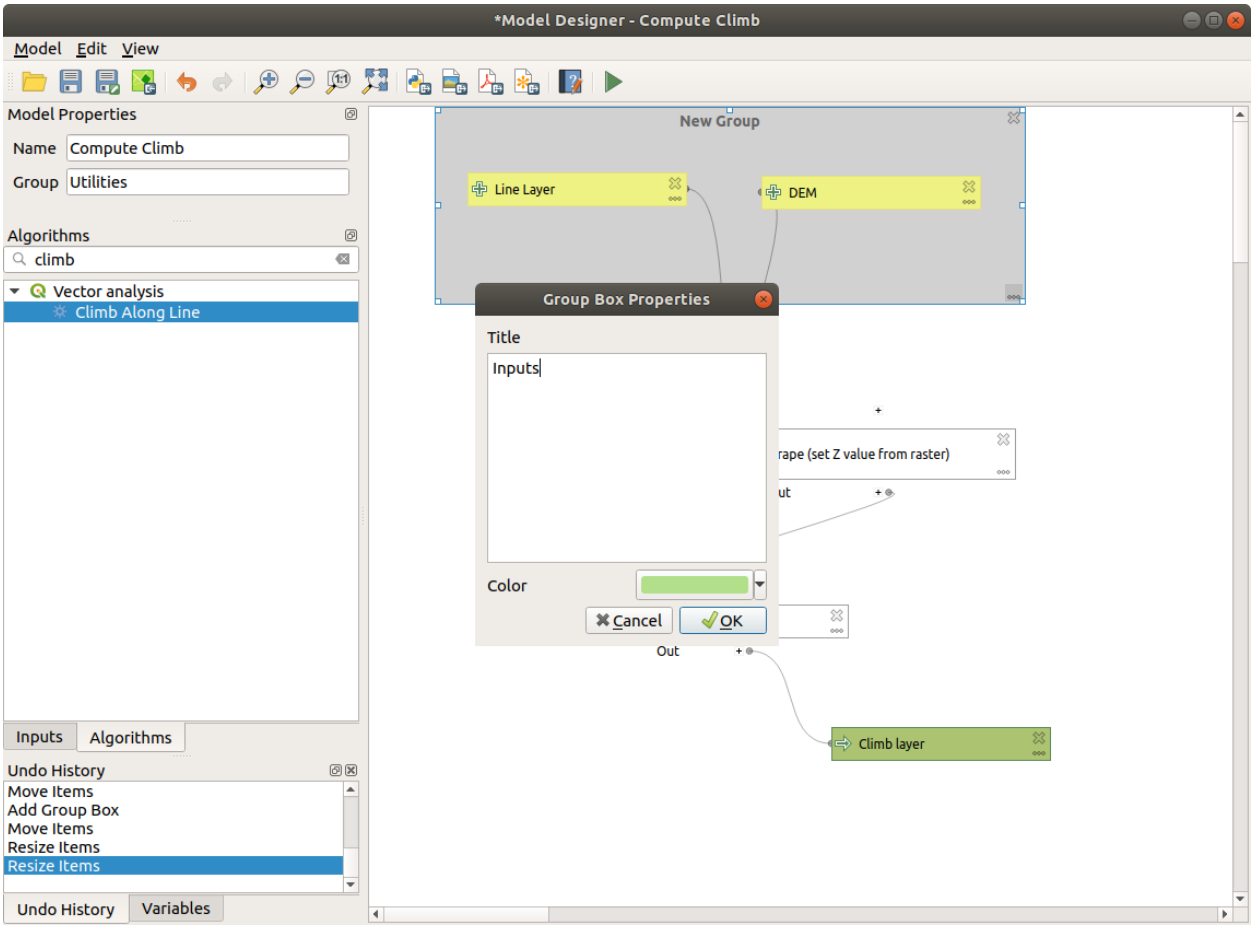


그림 23.23: 모델 그룹 상자

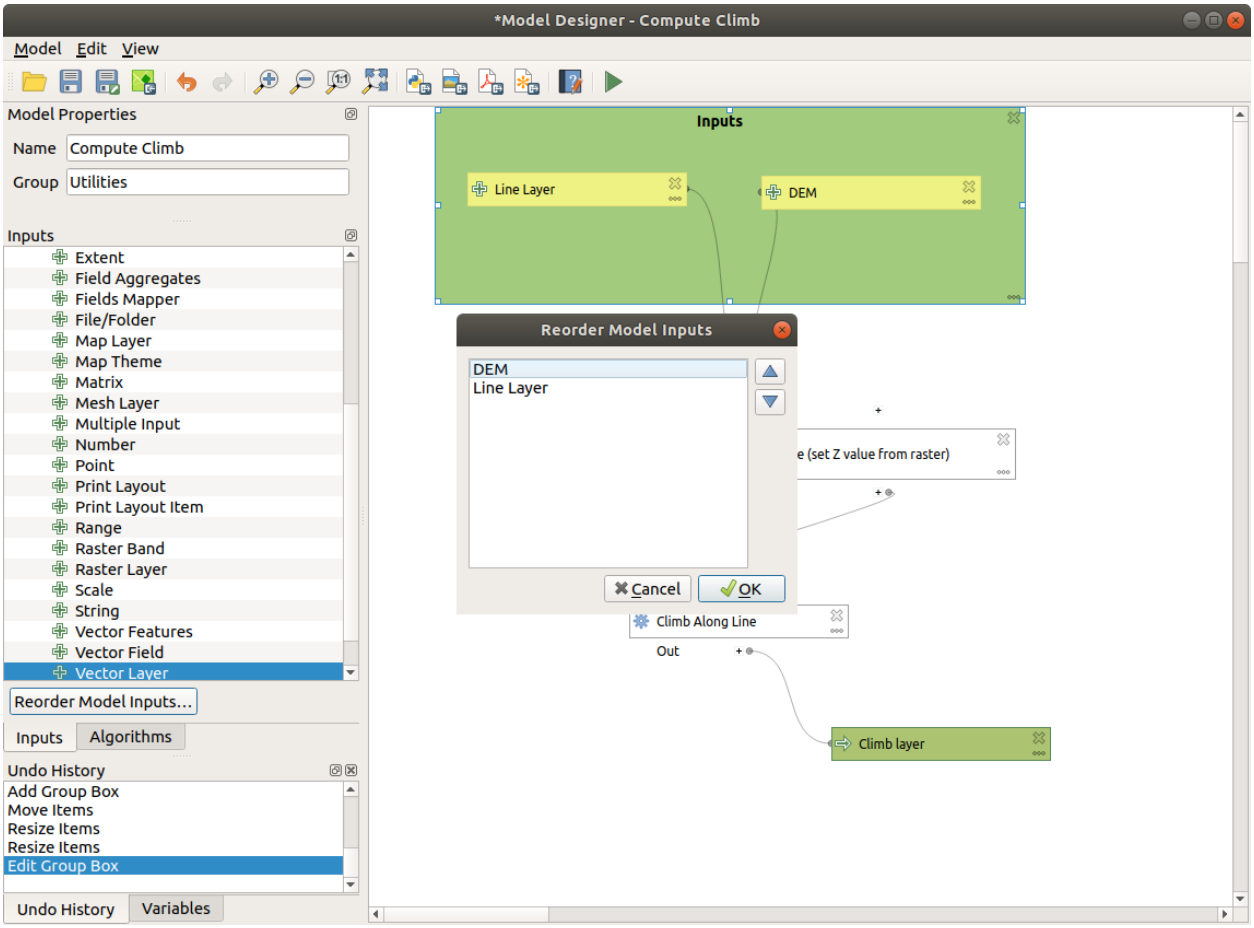


그림 23.24: 모델 입력물 재정렬

Processing Options 환경 설정 대화창의 *Modeler* 그룹에서 *Model folder* 를 설정할 수 있습니다.

`models` 폴더에서 불러온 모델은 툴박스뿐만 아니라 모델 생성기 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 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.

23.5.5 모델 편집

현재 생성 중인 모델을 편집해서 작업 흐름 및 모델 그 자체를 정의하는 알고리즘과 입력물 사이의 관계를 재정립할 수 있습니다.

캔버스에 있는 알고리즘을 오른쪽 클릭하면, 다음 그림과 같은 컨텍스트 메뉴가 나타납니다:

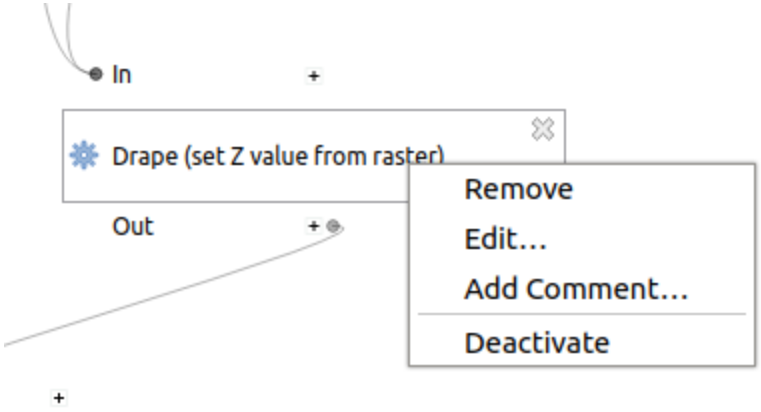


그림 23.25: 모델 생성기 오른쪽 클릭

Remove 옵션을 선택하면 선택한 알고리즘이 제거될 것입니다. 선택한 알고리즘에 종속되는 다른 알고리즘이 없는 경우에만 — 즉, 선택한 알고리즘의 산출물이 다른 알고리즘의 입력물로 사용되지 않는 경우에만 — 제거할 수 있습니다. 다른 알고리즘이 종속하고 있는 알고리즘을 제거하려 하는 경우, 다음 그림과 같은 경고 메시지를 보게 될 것입니다:

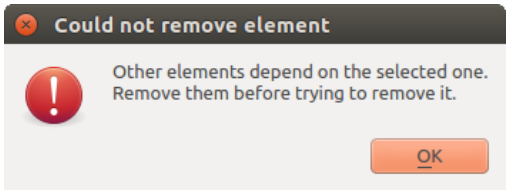


그림 23.26: 알고리즘을 삭제할 수 없습니다

Edit…옵션을 선택하면 해당 알고리즘의 파라미터 대화창이 열려서, 입력물 및 파라미터 값을 변경할 수 있습니다. 이 경우 모델에서 사용 가능한 모든 항목이 사용할 수 있는 입력물로 나타나지는 않을 것입니다. 모델이 정의하는 작업 흐름 상 더 진행된 단계에서 생성된 레이어 또는 값이 순환 종속성 (circular dependency) 을 유발하는 경우, 이를 사용할 수 없습니다.

새 값을 선택한 다음 이전과 같이 *OK* 버튼을 클릭하십시오. 모델 생성기 캔버스에서 모델 항목들 사이의 연결이 변경 사항에 맞춰 변경될 것입니다.

Add comment…옵션을 선택하면 알고리즘의 습성을 자세히 설명하기 위한 코멘트를 추가할 수 있습니다.

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.

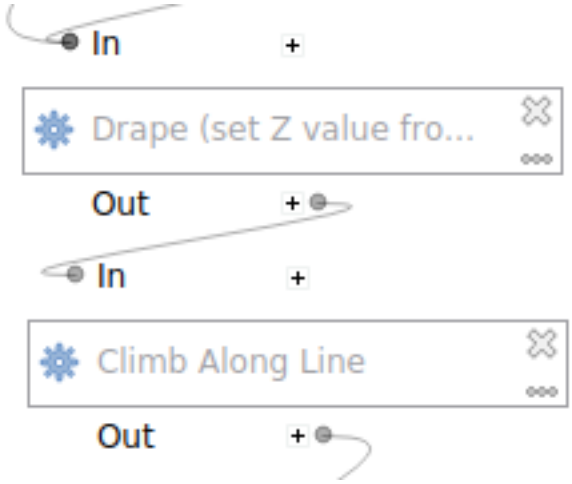


그림 23.27: 알고리즘이 비활성화된 모델

활성화 상태가 아닌 알고리즘을 오른쪽 클릭하는 경우, 선택한 알고리즘을 다시 활성화할 수 있는 *Activate* 메뉴가 나타날 것입니다.

23.5.6 모델 도움말 파일 및 메타 정보 편집

모델 작성자 내에서 사용자 모델에 관한 문서를 작성할 수 있습니다. 대화창 상단에 있는 Edit model help 버튼을 클릭하면 다음 그림과 같은 대화창이 열립니다.

대화창 상단에는 간소한 HTML 페이지 창이 있습니다. 입력 파라미터 및 알고리즘 산출물의 명칭과 모델 또는 모델 생성자에 대한 일반적인 설명을 함께 사용해서 이 페이지의 내용을 생성합니다. 처음 도움말 편집기를 열면, 이 모든 내용이 비어 있습니다. 그러나 대화창 좌하단에 있는 항목들을 이용하면 이를 편집할 수 있습니다. 좌하단 패널에서 항목을 선택한 다음 우하단 텍스트 란에 명칭 및 설명을 작성하십시오.

모델 도움말은 모델의 일부로 저장됩니다.

23.5.7 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 생성기 캔버스에서 Export as Script Algorithm… 을 클릭하거나 공간 처리 툴박스에 있는 모델명을 오른쪽 클릭한 다음 Export Model as Python Algorithm… 을 선택하십시오.

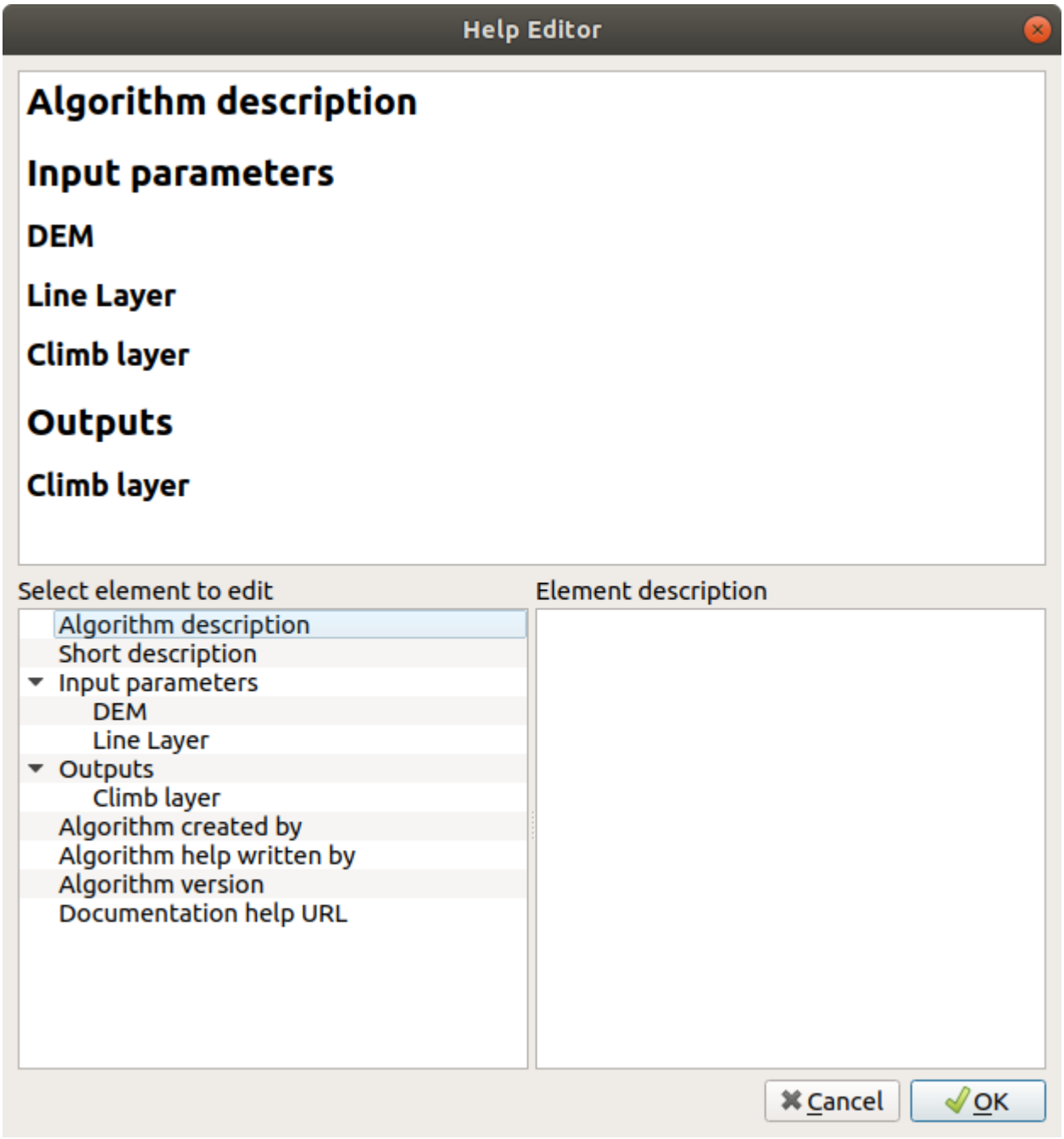


그림 23.28: 도움말 편집하기

23.5.8 사용 가능한 알고리즘에 관해

툴박스에서 실행할 수 있는 알고리즘 가운데 일부가 모델 설계 시 사용할 수 있는 알고리즘 목록에는 없다는 사실을 알아차렸을지도 모르겠습니다. 모델에 추가되는 알고리즘은 정확한 의미 체계 (semantic) 를 가지고 있어야만 합니다. 알고리즘이 이렇게 제대로 정의된 의미 체계를 보유하고 있지 않을 경우 (예를 들면 산출 레이어의 개수를 미리 알 수 없는 경우) 모델에 해당 알고리즘을 사용할 수 없습니다. 그러므로 이런 알고리즘은 모델 생성기 대화창의 사용 가능한 알고리즘 목록에 나타나지 않는 것입니다.

23.6 배치 프로세스 인터페이스

23.6.1 소개

모든 (모델 포함) 알고리즘을 배치 (batch) 프로세스로 실행할 수 있습니다. 단일 입력물 집합이 아니라 여러 집합을 사용해서 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.

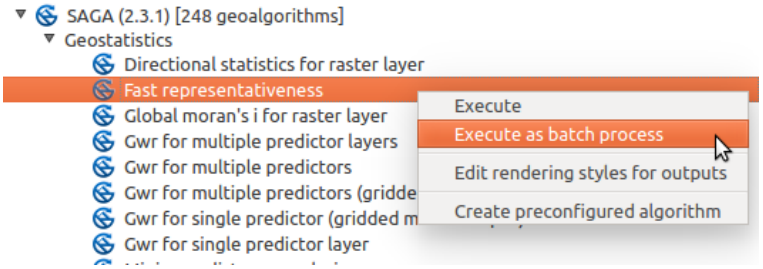


그림 23.29: 오른쪽 클릭으로 배치 프로세스 실행

선택한 알고리즘의 실행 대화창이 열리면, **[Run as batch process…]** 버튼을 눌러 배치 프로세스 작업 인터페이스도 열 수 있습니다.

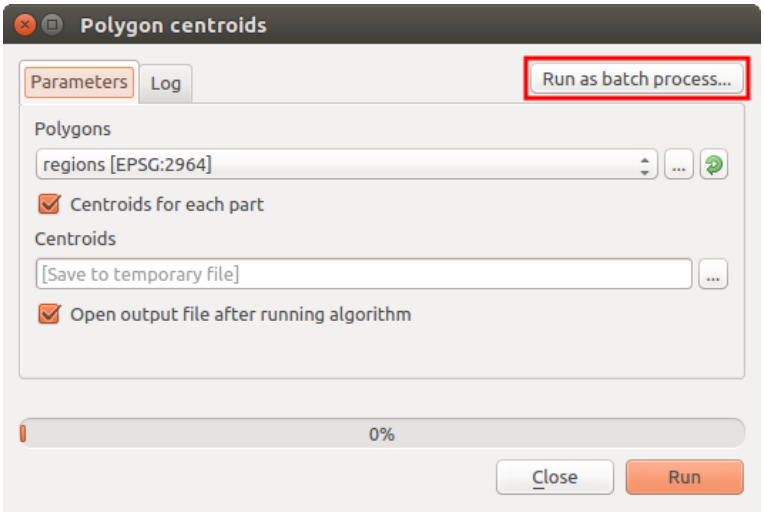


그림 23.30: 알고리즘 대화창에서 배치 프로세스 작업 시작

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. 다음 그림과 같은 표를 통해 이 값들을 설정합니다.

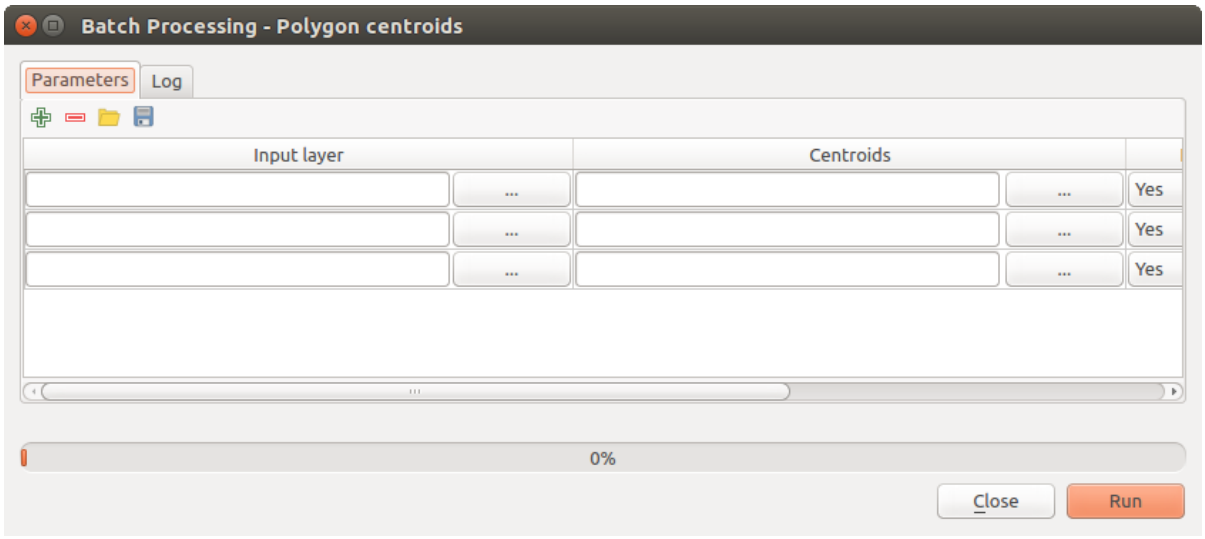


그림 23.31: 배치 프로세스 작업

이 표의 각 행은 알고리즘의 단일 실행을 나타내고, 각 셀은 파라미터 가운데 하나의 값을 담게 됩니다. 툴박스에서 알고리즘을 실행할 때 보게 되는 파라미터 대화창과 본질적으로 비슷하지만, 구조가 다릅니다.

이 표는 기본적으로 두 행만 담고 있을 겁니다. *Parameters* 탭 상단에 있는|signPlus| 또는|signMinus| 버튼으로 행을 추가 또는 제거할 수 있습니다.

표의 행 수를 (알고리즘 실행 횟수만큼) 설정하고 나면, 원하는 값을 입력해야 합니다.

23.6.3 파라미터 표 채우기

대부분의 파라미터의 경우 값 설정은 쉬운 일입니다. 파라미터 유형에 따라 그냥 값을 입력하거나, 사용할 수 있는 옵션 목록에서 값을 선택하면 됩니다.

입력 데이터 객체를 위한 파일명은 직접 입력하거나 또는, 좀 더 간편하게, 셀 우측에 있는 …버튼을 클릭하면 나타나는 컨텍스트 메뉴에서 현재 불러온 레이어에서 선택하는 옵션과 파일 시스템에서 선택하는 옵션 가운데 하나를 선택하면 됩니다. 두 번째 옵션을 선택하면 전형적인 파일 선택기 대화창이 열립니다. 파일 여러 개를 한 번에 선택할 수 있습니다. 입력 파라미터가 단일 데이터 객체를 받는데 파일 여러 개를 선택한 경우, 선택한 파일들을 각각 다른 행에 — 필요한 경우 행을 새로 추가해가면서 — 입력할 것입니다. 입력 파라미터가 다중 데이터 객체를 받는 경우, 셀 하나에 선택한 모든 파일을 쌍반점 (;) 으로 구분해서 입력할 것입니다.

파라미터 텍스트 란에 레이어 식별자를 직접 입력할 수 있습니다. 파일을 가리키는 전체 경로를 입력하거나, 현재 QGIS 프로젝트에 불러온 상태인 레이어의 명칭을 입력할 수 있습니다. 레이어명은 자동적으로 레이어의 소스 경로로 분석될 것입니다. 레이어 여럿이 동일한 명칭을 보유하고 있는 경우, 그 모호성 때문에 기대하지 않은 결과물이 산출될 수도 있습니다.

산출 데이터 객체는 언제나 파일로 저장되며, 툴박스에서 알고리즘을 실행할 때와는 달리 임시 파일 또는 데이터베이스로 저장할 수 없습니다. 파일 경로를 직접 입력하거나|browseButton| 버튼을 누르면 나타나는 파일 선택기 대화창을 통해 경로를 지정할 수 있습니다.

파일 경로를 선택하고 나면, 동일 열 (동일 파라미터) 에 있는 다른 셀을 자동 완성시킬 수 있는 새 대화창이 열립니다.

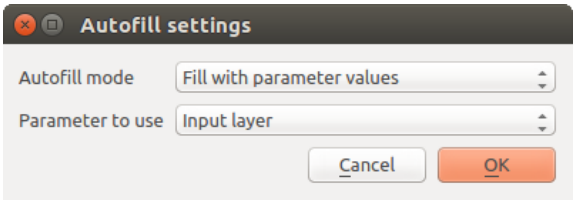


그림 23.32: 배치 프로세스 작업 - 저장

Autofill mode 옵션을 기본값 ('Do not autocomplete') 으로 선택할 경우, 그냥 파라미터 표에서 선택한 셀에 선택한 파일명을 입력할 것입니다. 다른 옵션을 선택하면, 선택한 셀 아래에 있는 모든 셀에 정의한 기준에 따라 파일명을 입력할 것입니다. 표에 값을 훨씬 쉽게 채울 수 있고, 배치 프로세스를 정의하는 데 드는 시간과 노력을 절약할 수 있는 방법입니다.

자동 완성 기능은 선택한 파일 경로에 상관 (correlative) 번호를 단순히 추가하거나, 또는 동일 행에 있는 다른 필드의 값을 붙입니다. 산출 데이터 객체를 입력 데이터 객체에 따라 명명할 수 있는 매우 유용한 방식입니다.

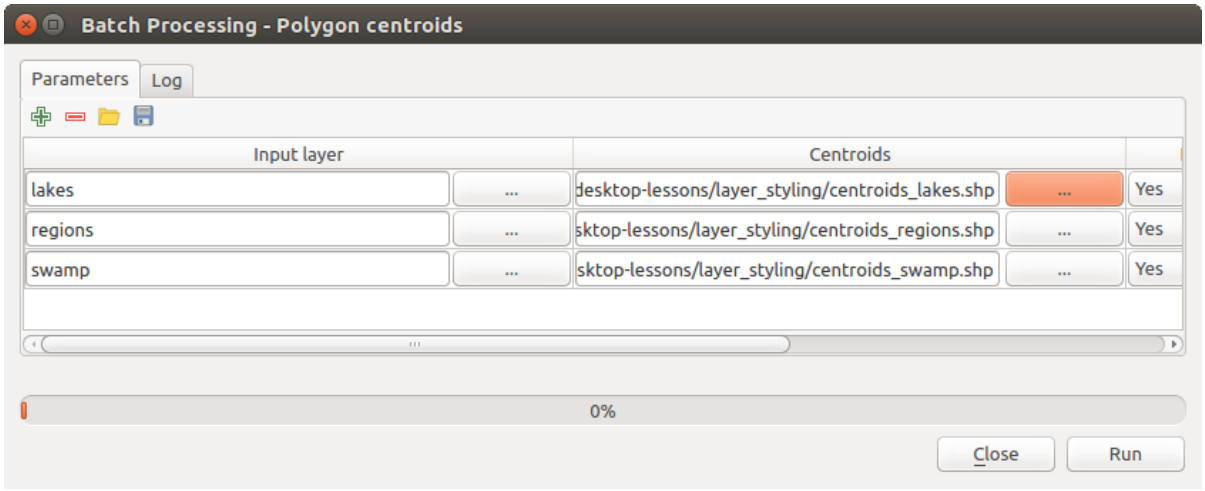


그림 23.33: 배치 프로세스 작업 - 파일 경로

23.6.4 배치 프로세스 실행

필요한 모든 값을 설정한 다음 배치 프로세스를 실행하려면, *OK* 버튼만 누르면 됩니다. 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, 그래픽 모델 생성기, 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 명칭입니다. 이 명칭은 툴박스에서 찾아볼 수 있는 명칭이 아니며, 오히려 유일한 명령 줄 명칭에 가깝습니다. 사용자 알고리즘의 정확한 명칭을 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 명칭과 함께 정리한 것입니다.

알고리즘의 명령 줄 명칭을 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 명칭이 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
```

(다음 페이지에 계속)

(이전 페이지에서 계속)

```
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:
            - int
            - float
            - QgsProperty

END_CAP_STYLE: End cap style

   Parameter type: QgsProcessingParameterEnum

   Available values:
```

(다음 페이지에 계속)

(이전 페이지에서 계속)

```
            - 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:
            - 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 PostGIS table
            - QgsProcessingOutputLayerDefinition
            - QgsProperty

----------------
Outputs
```

(다음 페이지에 계속)

(이전 페이지에서 계속)

```
----------------

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 명칭 (QGIS 범례에 표시되는 명칭) 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:
             - 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 명칭 또는 파일 경로를 각 입력 서술자로 사용할 수 있습니다.

- XXX 테이블의 필드: 사용할 필드의 명칭을 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()` 을 사용해서 동일한 알고리즘을 실행해도 산출물이 QGIS 맵 캔버스에 추가되지 않습니다. 그러나 `runAndLoadResults()` 로 실행하면 산출물을 맵 캔버스에 추가할 것입니다.

`run()` 메소드는 (알고리즘 설명에 표시되는) 하나 이상의 산출물 명칭을 키로 삼고 해당 산출물의 파일 경로를 값으로 하는 목록 (dictionary) 을 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
```

`load()` 메소드에 대응하는 파일 경로를 넘겨주면 피처 산출물을 불러올 수 있습니다. 또는 `run()` 대신 `runAndLoadResults()` 를 사용해서 즉시 불러올 수도 있습니다.

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) 명칭을 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__()

    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(
```

(다음 페이지에 계속)

(이전 페이지에서 계속)

```
            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. 다음은 알파벳순으로 정렬한 목록입니다:

- `QgsProcessingParameterAggregate`
- `QgsProcessingParameterAuthConfig`
- `QgsProcessingParameterBand`
- `QgsProcessingParameterBoolean`
- `QgsProcessingParameterColor`
- `QgsProcessingParameterCoordinateOperation`
- `QgsProcessingParameterCrs`
- `QgsProcessingParameterDatabaseSchema`

- QgsProcessingParameterDatabaseTable
- QgsProcessingParameterDateTime
- QgsProcessingParameterDistance
- QgsProcessingParameterEnum
- QgsProcessingParameterExpression
- QgsProcessingParameterExtent
- QgsProcessingParameterFeatureSink
- QgsProcessingParameterFeatureSource
- QgsProcessingParameterField
- QgsProcessingParameterFieldMapping
- QgsProcessingParameterFile
- QgsProcessingParameterFileDestination
- QgsProcessingParameterFolderDestination
- QgsProcessingParameterLayout
- QgsProcessingParameterLayoutItem
- QgsProcessingParameterMapLayer
- QgsProcessingParameterMapTheme
- QgsProcessingParameterMatrix
- QgsProcessingParameterMeshLayer
- QgsProcessingParameterMultipleLayers
- QgsProcessingParameterNumber
- QgsProcessingParameterPoint
- QgsProcessingParameterProviderConnection
- QgsProcessingParameterRange
- QgsProcessingParameterRasterDestination
- QgsProcessingParameterRasterLayer
- QgsProcessingParameterScale
- QgsProcessingParameterString
- QgsProcessingParameterVectorDestination
- QgsProcessingParameterVectorLayer
- QgsProcessingParameterVectorTileWriterLayers

구성자 (constructor) 가 받는 첫 번째 파라미터는 파라미터의 명칭이며, 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 유효하지 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 유효하지 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 실행전 및 실행후 스크립트 후크

스크립트를 알고리즘을 실행하기 전, 실행한 후에 실행되는 실행전 (pre-execution) 및 실행후 (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 명령 줄에서 공간 처리 알고리즘 사용

QGIS 는 사용자가 QGIS 데스크탑 자체를 실행하지 않고서도 명령 줄에서 공간 처리 알고리즘과 (내장 또는 플러그인이 제공하는) 모델을 직접 실행할 수 있는 `QGIS Processing Executor` 라는 도구를 제공합니다.

명령 줄에서 `qgis_process` 를 실행하면 다음과 같은 화면을 보게 될 것입니다:

```
QGIS Processing Executor - 3.16.8-Hannover 'Hannover' (3.16.8-Hannover)
Usage: C:\OSGeo4W\apps\qgis-ltr\bin\qgis_process.exe [--json] [command] [algorithm id␣
↪or path to model file] [parameters]

Options:

  --json          Output results as JSON objects

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 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
                   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.
```

참고: `qgis_process` 도구는 설치된 플러그인 가운데 `metadata.txt` 파일에 `hasProcessingProvider=yes` 를 명시한 플러그인만 식별하고, 활성화시키거나 불러올 수 있습니다.

`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 qgis: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 포맷을 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.

23.9 새 공간 처리 알고리즘을 파이썬 스크립트로 작성

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')

    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.
        """
```

(다음 페이지에 계속)

(이전 페이지에서 계속)

```
        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'
            )
        )
        self.addOutput(
            QgsProcessingOutputNumber(
                'NUMBEROFFEATURES',
                self.tr('Number of features processed')
            )
        )

    def processAlgorithm(self, parameters, context, feedback):
        """
        Here is where the processing itself takes place.
```

(다음 페이지에 계속)

(이전 페이지에서 계속)

```
        """
        # 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)

        # 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'],
```

(다음 페이지에 계속)

(이전 페이지에서 계속)

```
                '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 명칭을 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 명칭으로 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 명칭이고, 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 명칭과 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')

def bufferrasteralg(instance, parameters, context, feedback, inputs):
    """
    Description of the algorithm.
    (If there is no comment here, you will get an error)
    """
    input_featuresource = instance.parameterAsSource(parameters,
                                                     'INPUT', context)
    numfeatures = input_featuresource.featureCount()
    bufferdist = instance.parameterAsDouble(parameters, 'BUFFERDIST',
                                            context)
    rastercellsize = instance.parameterAsDouble(parameters, 'CELLSIZE',
                                                context)
    if feedback.isCanceled():
        return {}
    buffer_result = processing.run('native:buffer',
                                   {'INPUT': parameters['INPUT'],
                                    'OUTPUT': parameters['BUFFER_OUTPUT'],
                                    'DISTANCE': bufferdist,
```

(다음 페이지에 계속)

(이전 페이지에서 계속)

```
                                'SEGMENTS': 10,
                                'DISSOLVE': True,
                                'END_CAP_STYLE': 0,
                                'JOIN_STYLE': 0,
                                'MITER_LIMIT': 10
                                },
                               is_child_algorithm=True,
                               context=context,
                               feedback=feedback)
    if feedback.isCanceled():
        return {}
    rasterized_result = processing.run('qgis:rasterize',
                               {'LAYER': buffer_result['OUTPUT'],
                                'EXTENT': buffer_result['OUTPUT'],
                                'MAP_UNITS_PER_PIXEL': rastercellsize,
                                'OUTPUT': parameters['OUTPUT']
                               },
                               is_child_algorithm=True, context=context,
                               feedback=feedback)
    if feedback.isCanceled():
        return {}
    return {'OUTPUT': rasterized_result['OUTPUT'],
            'BUFFER_OUTPUT': buffer_result['OUTPUT'],
            '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 그래픽 모델 생성기 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 명칭 및 위치를 정의합니다.
- @alg.input 장식자를 사용해서 알고리즘의 입력물을 정의합니다.
- @alg.output 장식자를 사용해서 알고리즘의 산출물을 정의합니다.

23.9.3 공간 처리 알고리즘의 입력물 및 산출물 유형

다음은 공간 처리 프레임워크가 지원하는 입력물 및 산출물 유형을 대응하는 alg 장식자 상수와 함께 정리한 목록입니다. (`algfactory.py` 파일이 alg 상수의 전체 목록을 담고 있습니다.) 클래스명으로 정렬되어 있습니다.

입력물 유형

클래스	alg 상수	설명
QgsProcessingParameterAuthConfig	alg.AUTH_CFG	사용자가 사용할 수 있는 인증 환경 설정에서 선택하거나 또는 새 인증 환경 설정을 생성할 수 있게 해줍니다.
QgsProcessingParameterBand	alg.BAND	래스터 레이어의 밴드
QgsProcessingParameterBoolean	alg.BOOL	불 (boolean) 값
QgsProcessingParameterColor	alg.COLOR	색상
QgsProcessingParameterCoordinate-Operation	alg. COORDINATE_OPERATION	좌표 연산 (좌표계 변환용)
QgsProcessingParameterCrs	alg.CRS	좌표계
QgsProcessingParameterDatabas-eSchema	alg.DATABASE_SCHEMA	데이터베이스 스키마
QgsProcessingParameter-DatabaseTable	alg.DATABASE_TABLE	데이터베이스 테이블
QgsProcessingParameterDateTime	alg.DATETIME	날짜 & 시간 (또는 순수한 날짜 또는 시간)
QgsProcessingParameterDistance	alg.DISTANCE	거리 값을 위한 실수형 (double) 숫자 파라미터
QgsProcessingParameterEnum	alg.ENUM	사전 정의된 값들의 집합에서 선택할 수 있는 집합 (enumeration)
QgsProcessingParameterExpression	alg.EXPRESSION	표현식
QgsProcessingParameterExtent	alg.EXTENT	xmin, xmax, ymin, ymax 로 정의된 공간 범위
QgsProcessingParameterField	alg.FIELD	벡터 레이어의 속성 테이블에 있는 필드
QgsProcessingParameterFile	alg.FILE	기존 파일의 파일명
QgsProcessingParameterFileDesti-nation	alg.FILE_DEST	새로 생성된 산출물 파일의 파일명
QgsProcessingParameterFolderDes-tination	alg.FOLDER_DEST	폴더 (대상 폴더)
QgsProcessingParameterNumber	alg.INT	정수
QgsProcessingParameterLayout	alg.LAYOUT	조판
QgsProcessingParameterLayoutItem	alg.LAYOUT_ITEM	조판 항목
QgsProcessingParameterMapLayer	alg.MAPLAYER	맵 레이어
QgsProcessingParameterMapTheme	alg.MAP_THEME	프로젝트 맵 테마
QgsProcessingParameterMatrix	alg.MATRIX	매트릭스
QgsProcessingParameterMeshLayer	alg.MESH_LAYER	메시 레이어
QgsProcessingParameterMultiple-Layers	alg.MULTILAYER	레이어 집합
QgsProcessingParameterNumber	alg.NUMBER	숫자 값
QgsProcessingParameterPoint	alg.POINT	포인트
QgsProcessingParameterProvider-Connection	alg. PROVIDER_CONNECTION	데이터베이스 제공자에 대해 사용할 수 있는 연결
QgsProcessingParameterRange	alg.RANGE	숫자의 범위
QgsProcessingParameterRasterLayer	alg.RASTER_LAYER	래스터 레이어
QgsProcessingParameterRasterDes-tination	alg.RASTER_LAYER_DEST	래스터 레이어

다음 페이지에 계속

표 23.1 – 이전 페이지에서 계속

클래스	alg 상수	설명
QgsProcessingParameterScale	alg.SCALE	맵 축척
QgsProcessingParameterFeatureSink	alg.SINK	피처 싱크 (sink)
QgsProcessingParameterFeature-Source	alg.SOURCE	피처 소스
QgsProcessingParameterString	alg.STRING	텍스트 문자열
QgsProcessingParameterVectorLayer	alg.VECTOR_LAYER	벡터 레이어
QgsProcessingParameterVectorDes-tination	alg.VECTOR_LAYER_DEST	벡터 레이어

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
QgsProcessingOutputNumber	alg.INT	정수
QgsProcessingOutputLayerDefinition	alg.LAYERDEF	레이어 정의
QgsProcessingOutputMapLayer	alg.MAPLAYER	맵 레이어
QgsProcessingOutputMultipleLayers	alg.MULTILAYER	레이어 집합
QgsProcessingOutputNumber	alg.NUMBER	숫자 값
QgsProcessingOutputRasterLayer	alg.RASTER_LAYER	래스터 레이어
QgsProcessingOutputString	alg.STRING	텍스트 문자열
QgsProcessingOutputVectorLayer	alg.VECTOR_LAYER	벡터 레이어

23.9.4 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 생성기에서 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.

23.9.5 사용자에게 정보 제공하기

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 생성기, 파이썬 콘솔 등등) 실행되느냐에 따라 예외를 처리하고 사용자에게 알려줄 것입니다.

23.9.6 사용자 스크립트 문서 작성하기

`QgsProcessingAlgorithm` 클래스의 `helpString()` 및 `helpUrl()` 메소드를 과부하 (overload) 시키면 사용자 스크립트를 문서화할 수 있습니다.

23.9.7 플래그

`QgsProcessingAlgorithm` 클래스의 `flags()` 메소드를 무시하면 QGIS 에 사용자 알고리즘에 대해 더 많이 알려줄 수 있습니다. 예를 들어 QGIS 에 스크립트를 모델 생성기로부터 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.

23.9.8 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 생성기, 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. https://github.com/qgis/QGIS/blob/release-3_16/python/plugins/processing/algs/qgis 에서 코드를 찾아볼 수 있습니다.

23.10 외부 응용 프로그램 환경 설정

추가적인 응용 프로그램을 사용하면 공간 처리 프레임워크를 확장할 수 있습니다. 외부 응용 프로그램에 의존하는 알고리즘은 각각 고유의 알고리즘 제공자가 관리합니다. QGIS 플러그인 관리자를 사용하면 추가 제공자를 개별 플러그인으로 찾아서 설치할 수 있습니다.

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 그래픽 모델 생성기 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.

23.10.1 윈도우 사용자를 위한 소고

사용자가 고급 사용자가 아니며 윈도우 상에서 QGIS 를 실행하고 있는 경우, 이 장의 나머지 부분을 읽어보지 않아도 괜찮습니다. 단독형 설치 관리자를 사용해서 사용자 시스템에 QGIS 를 설치하십시오. 그러면 사용자 시스템에 자동적으로 SAGA 와 GRASS 를 설치하고 QGIS 에서 실행할 수 있도록 환경 설정하기 때문에, 별다른 환경 설정을 하지 않아도 이 제공자들의 모든 알고리즘을 사용할 수 있습니다. OSGeo4W 응용 프로그램을 통해 설치하는 경우, SAGA 및 GRASS 도 함께 설치하도록 선택했는지 확인하십시오.

23.10.2 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 Shapefile 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.

23.10.3 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.

23.10.4 SAGA

SAGA 가 QGIS 설치 관리자에 포함된 경우 QGIS 에서 SAGA 알고리즘을 실행할 수 있습니다.

사용자가 윈도우를 실행 중이라면, 단독형 설치 관리자와 OSGeo4W 설치 관리자 모두 SAGA 를 포함하고 있습니다.

SAGA 그리드 시스템 제약에 관해

입력 래스터 데이터 여러 개를 요구하는 SAGA 알고리즘 대부분은 입력 래스터가 동일한 그리드 시스템을 보유하고 있을 것을 요구합니다. 즉, 입력 래스터들이 동일한 지리 영역을 커버하고 동일한 셀 크기를 가지고 있어야만 하기 때문에 그에 대응하는 그리드들도 일치해야만 한다는 뜻입니다. QGIS 에서 SAGA 알고리즘을 호출하는 경우 사용자는 셀 크기나 범위에 상관없이 어떤 레이어든 사용할 수 있습니다. SAGA 알고리즘이 여러 래스터 레이어를 입력받아야 한다면 (해당 알고리즘이 서로 다른 그리드 시스템을 가진 레이어들을 처리할 수 없는 경우) QGIS 가 래스터들을 공통 그리드 시스템으로 리샘플링해서 SAGA 에 넘겨줍니다.

이때 공통 그리드 시스템은 사용자가 정의합니다. 설정 대화창에서 *SAGA* 그룹에 있는 파라미터 몇 가지를 사용해서 정의할 수 있습니다. 다음 두 가지 방법 가운데 하나로 대상 그리드 시스템을 설정합니다:

- 직접 설정합니다. 다음 파라미터의 값을 설정해서 범위를 정의합니다:
  - *Resampling min X*
  - *Resampling max X*
  - *Resampling min Y*
  - *Resampling max Y*
  - *Resampling cellsize*

  설정한 범위와 입력 레이어의 범위가 일치하지 않더라도 QGIS 가 입력 레이어들을 설정한 범위로 리샘플링할 것이라는 사실을 유념하십시오.
- 입력 레이어로부터 자동 설정합니다. 이 옵션을 선택하려면, *Use min covering grid system for resampling* 옵션을 체크하면 됩니다. 다른 모든 설정을 무시하고 모든 입력 레이어를 커버하는 최소 범위를 사용할 것입니다. 대상 레이어의 셀 크기는 입력 레이어들의 모든 셀 크기 가운데 가장 큰 크기를 사용합니다.

여러 래스터 레이어를 사용하지 않는 알고리즘의 경우, 또는 유일한 입력 그리드 시스템을 요구하지 않는 알고리즘의 경우 SAGA 호출 전에 리샘플링을 하지 않으므로 이런 파라미터들도 사용하지 않습니다.

다중 밴드 레이어에 대한 제약

QGIS 와 달리, SAGA 는 다중 밴드 레이어를 지원하지 않습니다. (RGB 또는 다중 스펙트럼 이미지 같은) 다중 밴드 레이어를 사용하고 싶다면 먼저 해당 레이어를 단일 밴드 이미지로 분할해야 합니다. RGB 이미지에서 이미지 3 개를 생성하는 `SAGA/Grid - Tools/Split RGB image` 알고리즘을 사용하거나 또는 단일 밴드 이미지를 추출하는 `SAGA/Grid - Tools/Extract band` 알고리즘을 사용하면 됩니다.

셀 크기에 대한 제약

SAGA 는 래스터 레이어의 X 및 Y 방향의 셀 크기가 동일하다고 가정하고 있습니다. 수평 및 수직 셀 크기가 다른 레이어를 작업하는 경우, 기대하지 않은 결과물이 나올 수도 있습니다. 이런 경우, SAGA 가 처리하기에 적합하지 않은 입력 레이어일 수도 있다는 경고 메시지가 공간 처리 로그에 추가될 것입니다.

로그 작업

QGIS 가 SAGA 를 호출할 때 SAGA 명령 줄 인터페이스를 사용해서 필요한 모든 작업을 수행하기 위한 명령어 집합을 넘겨줍니다. SAGA 는 그 진행 상태를 콘솔에 정보를 작성해서 표시하는데, 이 정보는 이미 완료된 공간 처리 작업의 백분율 및 부가적인 내용을 포함합니다. 알고리즘 실행 도중 이 산출 정보를 필터링해서 진행 상태 막대를 업데이트합니다.

QGIS 가 전송한 명령어는 물론 SAGA 가 출력한 부가 정보도 기타 공간 처리 로그 메시지와 함께 로그에 작성될 수 있습니다. QGIS 가 SAGA 알고리즘을 실행하는 과정에서 어떤 일이 일어나는지를 추적할 수 있는 유용한 정보입니다. 이런 로그 작업 기제를 활성화하려면, *Log console output* 및 *Log execution commands* 두 옵션을 활성화하십시오.

명령 줄 인터페이스를 통해 외부 응용 프로그램을 호출해서 사용하는 다른 제공자들 대부분도 비슷한 옵션을 가지고 있습니다. 공간 처리 설정 목록의 해당 제공자 부분에서 이런 옵션을 찾을 수 있을 겁니다.

23.10.5 R 스크립트

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.at.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.

QGIS 저장소에서 R 스크립트 추가하기

QGIS 에서의 R 통합은 사용자가 실행할 수 있는 사전 정의된 알고리즘 집합이 (공간 처리 *R* 제공자 플러그인과 함께 설치되는 몇몇 예시 스크립트를 제외하면) 없다는 점에서 SAGA 통합과는 다릅니다.

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.)

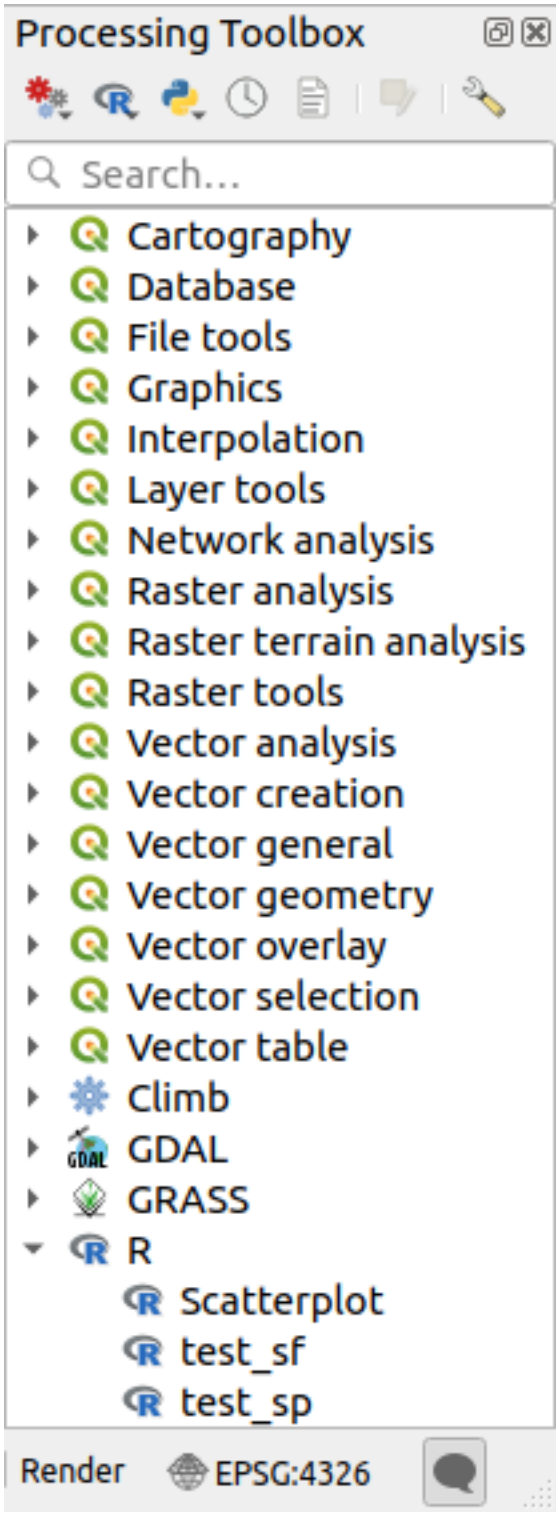
   가장 위에 있는 스크립트들은 공간 처리 *R* 제공자 플러그인의 예시 스크립트들입니다.
10. 만약 어떤 이유로든 *Processing Toolbox* 에 스크립트가 추가되지 않았다면, 다음 작업을 해볼 수 있습니다:
    1. 공간 처리 설정 대화창을 엽니다. (*Settings → Options → Processing* 탭)
    2. *Providers → R → R scripts folder* 를 선택합니다.
       - 우분투의 경우, 다음을 가리키는 경로를 설정하십시오 (또는, 더 나은 방법으로, PATH 에 포함시키십시오):

         /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.


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

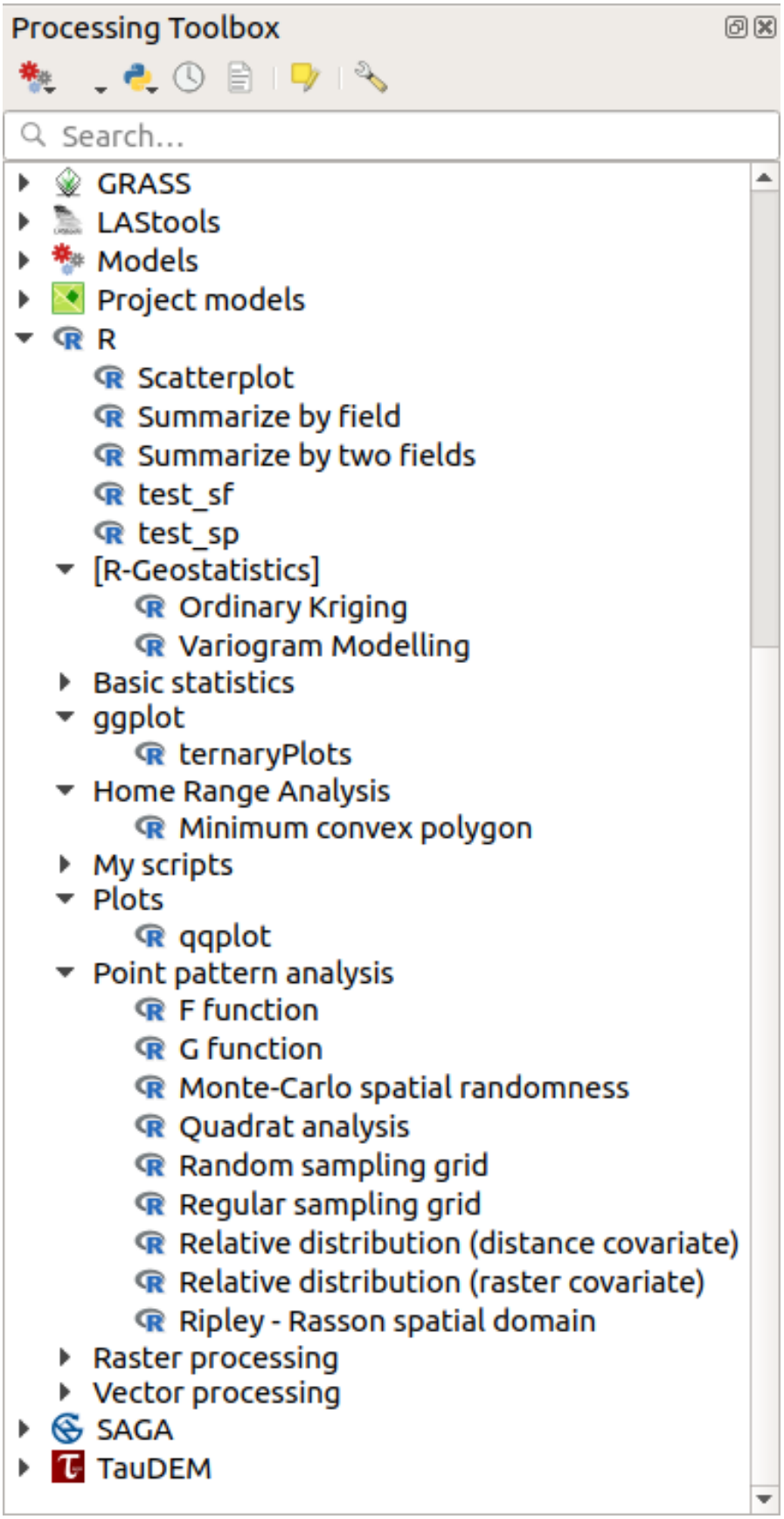
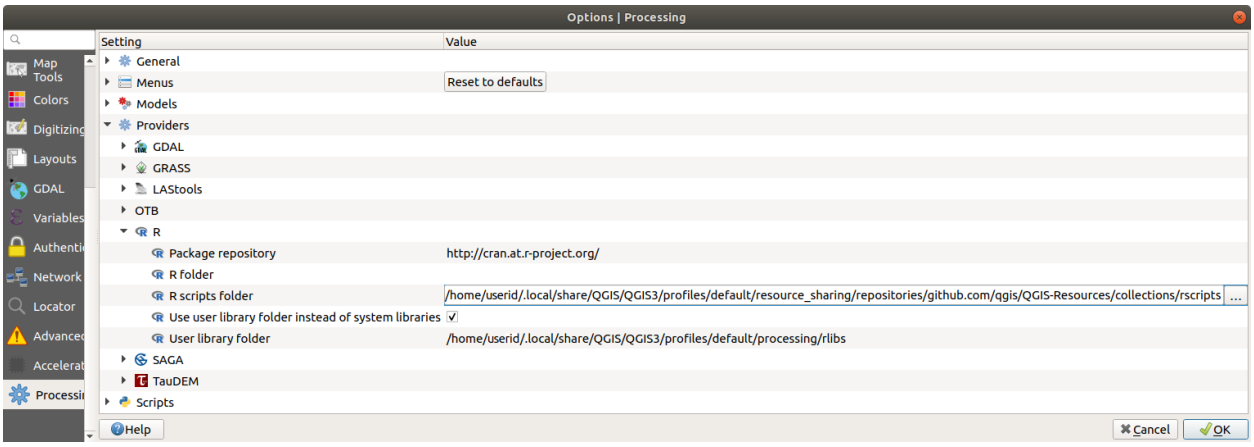
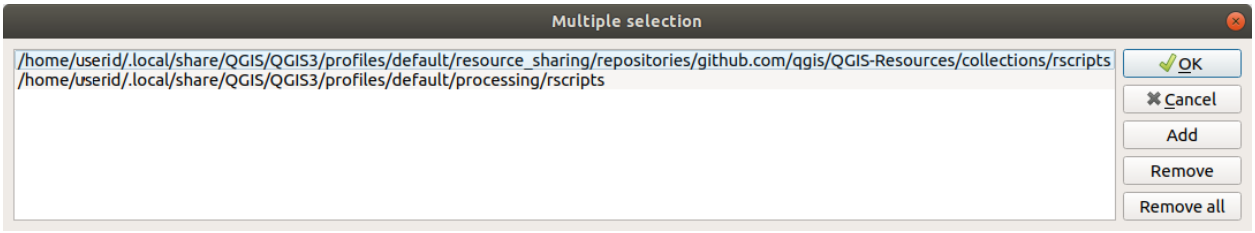


그림 23.34: 몇몇 R 스크립트가 표시된 *Processing Toolbox*





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 명칭 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

소개 에서*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 명칭을 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 명칭을 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 명칭과 (이 경우 `Output`) 동일한 명칭을 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. (앞에서 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**.

23.10.6 R 라이브러리

R 스크립트 `sp_test` 는 `sp` 및 `raster` R 패키지를 불러오려 합니다.

sf_test 실행시 설치되는 R 라이브러리

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

23.10.7 GRASS

GRASS 환경 설정은 SAGA 환경 설정과 그렇게 다르지 않습니다. 먼저, 윈도우 환경인 경우 GRASS 폴더를 가리키는 경로를 정의해야 합니다.

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 알고리즘은 계산 시 영역 (region) 을 사용합니다. SAGA 환경 설정에서와 비슷한 값들을 사용해서 이 영역을 직접 정의할 수도 있고, 또는 알고리즘 실행 시 사용되는 모든 입력 레이어를 커버하는 최소 범위를 받아 자동으로 정의하도록 할 수도 있습니다. 만약 두 번째 습성을 선호한다면, GRASS 환경 설정 파라미터 가운데 |checkbox| *Use min covering region* 옵션을 체크하면 됩니다.

23.10.8 LAStools

QGIS 에서 LAStools 를 사용하려면, 사용자 컴퓨터에 LAStools 를 다운로드해서 설치한 다음 QGIS 에 LAStools 플러그인을 (QGIS 공식 저장소에 있습니다) 설치해야 합니다.

리눅스 플랫폼에서 몇몇 도구를 실행하려면 Wine 이 필요할 것입니다.

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.

23.10.9 OTB 응용 프로그램

QGIS 공간 처리 프레임워크는 OTB 응용 프로그램을 완전하게 지원합니다.

OTB (Orfeo ToolBox) 는 원격탐사 (remote sensing) 데이터용 영상 처리 라이브러리입니다. OTB 는 영상 처리 기능을 사용할 수 있는 응용 프로그램도 제공합니다. OTB 쿡북 에서 응용 프로그램 목록과 그 문서들을 볼 수 있습니다.

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 여십시오. (*Settings* → *Options* → *Processing* 탭)
2. OTB 가 *Providers* 아래 있습니다.

1. *OTB* 탭을 펼치십시오.
2. |checkbox| *Activate* 옵션을 체크하십시오.
3. *OTB folder* 를 사용자가 OTB 를 설치한 위치로 설정하십시오.
4. *OTB application folder* 를 사용자가 OTB 응용 프로그램을 설치한 위치 (`<PATH_TO_OTB_INSTALLATION>/lib/otb/applications`) 로 설정하십시오.
5. *OK* 를 클릭해서 설정을 저장하고 대화 창을 닫습니다.

정확하게 설정했다면, *Processing Toolbox* 에서 OTB 알고리즘을 사용할 수 있습니다.

QGIS 공간 처리 프레임워크에서 사용할 수 있는 OTB 설정 문서

- **Activate**: OTB 제공자를 활성화하거나 비활성화하는 체크박스입니다. 유효하지 않은 OTB 설정을 저장하면 이 옵션을 비활성화할 것입니다.
- **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.

QGIS 와 OTB 버전 사이의 호환성

모든 (OTB 6.6.1 이후) OTB 버전은 최신 QGIS 버전과 호환됩니다.

문제 해결

QGIS 공간 처리 프레임워크에서 OTB 응용 프로그램 사용에 문제가 발생한 경우, OTB 버그 트래커 에 `qgis` 라벨을 사용해서 스레드를 여십시오.

OTB 및 QGIS 관련 추가 정보는 여기 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 환경 설정 없이도"설치 후 바로"작동할 것입니다.

QGIS 제공자는 플러그인의 알고리즘을 일부 포함하며, QGIS 고유 알고리즘도 추가합니다.

24.1.1 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.

제자리 피처 수정

파라미터

라벨	명칭	유형	설명
Input layer	INPUT	[vector: point]	기울기를 계산할 포인트 피처
Reference layer	REFERENCE_LAYER	[vector: any]	기울기 계산을 위해 가장 가까이 있는 피처를 찾아야 할 레이어
Maximum distance to consider 부가적	MAX_DISTANCE	[number] 기본값: 설정하지 않음	이 거리 내에서 참조 피처를 찾지 못할 경우, 포인트 피처에 기울기를 할당하지 않습니다.
Angle field name	FIELD_NAME	[string] 기본값: 'rotation'	기울기 값을 저장할 필드
Automatically apply symbology	APPLY_SYMBOLOGY	[boolean] 기본값: True	각도 필드 값을 사용해서 피처의 심볼 마커를 기울입니다.
Aligned layer	OUTPUT	[vector: point] 기본값: [Save to temporary file]	기울인 산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (TEMPORARY_OUTPUT) • 파일로 저장 • Geopackage 로 저장 • 데이터베이스 테이블에 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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 명칭을 가진 동일한 유형의 항목이 있을 경우 산출된 결합 데이터베이스에서 유일한 명칭을 가지도록 재명명할 것입니다.

더 보기:

프로젝트에서 스타일 데이터베이스 생성하기

파라미터

라벨	명칭	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: • 0 —심볼 • 1 —색상표 • 2 —텍스트 서식 • 3 —라벨 설정
Output style database	`OUTPUT`	[file] 기본값: `[Save to temporary file]`	선택한 스타일 항목을 결합한 산출 `.XML` 파일. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장

산출물

라벨	명칭	유형	설명
Color ramp count	`COLORRAMPS`	[number]	
Label settings count	`LABELSETTINGS`	[number]	
Output style database	`OUTPUT`	[file]	선택한 스타일 항목을 결합한 산출 `.XML` 파일
Symbol count	`SYMBOLS`	[number]	
Text format count	`TEXTFORMATS`	[number]	

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 **명칭** 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	명칭	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 명칭 비교시 대소문자를 구분합니다.
Ignore non-alphanumeric characters while matching	`TOLERANT`	[boolean] 기본값: False	True 인 (체크한) 경우, 범주 및 심볼의 명칭에 있는 알파벳 또는 숫자가 아닌 문자를 무시할 것입니다. 명칭이 일치하는 지 비교하는 동안 허용도가 크게 올라갑니다.

다음 페이지에 계속

표 24.1 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Non-matching categories 부가적	NON_MATCHING_CATEGORIES	[table] 기본값: [Skip output]	데이터베이스에 있는 어떤 심볼과도 일치하지 않는 카테고리용 산출 테이블. 다음 가운데 하나로 지정할 수 있습니다: • 산출물 건너뛰기 • 임시 레이어 생성 (TEMPORARY_OUTPUT) • 파일로 저장 • Geopackage 로 저장 • 데이터베이스 테이블에 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.
Non-matching symbol names 부가적	NON_MATCHING_SYMBOLS	[table] 기본값: [Skip output]	입력 스타일 데이터베이스의 심볼 가운데 어떤 범주와도 일치하지 않는 심볼용 산출 테이블. 다음 가운데 하나로 지정할 수 있습니다: • 산출물 건너뛰기 • 임시 레이어 생성 (TEMPORARY_OUTPUT) • 파일로 저장 • Geopackage 로 저장 • 데이터베이스 테이블에 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Non-matching categories	NON_MATCHING_CATEGORIES	[table]	지정한 스타일 데이터베이스에 있는 어떤 심볼과도 일치되지 않는 범주를 목록화
Non-matching symbol names	NON_MATCHING_SYMBOLS	[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 **명칭** 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	명칭	유형	설명
Input project (leave blank to use current) 부가적	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: • 0 — 심볼 • 1 — 색상표 • 2 — 텍스트 서식 • 3 — 라벨 설정
Output style database	OUTPUT	[file] 기본값: [Save to temporary file]	선택한 스타일 항목을 저장할 산출 .XML 파일. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장

산출물

라벨	명칭	유형	설명
Color ramp count	COLORRAMPS	[number]	색상표 개수
Label settings count	LABELSETTINGS	[number]	라벨 설정 개수
Output style database	OUTPUT	[file]	선택한 스타일 항목을 저장한 산출 .XML 파일
Symbol count	SYMBOLS	[number]	심볼 개수
Text format count	TEXTFORMATS	[number]	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 **명칭** 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	명칭	유형	설명
Atlas layout	`LAYOUT`	[layout]	내보낼 조판
Coverage layer 부가적	`COVERAGE_LAYER`	[vector: any]	지도책을 생성하는 데 사용할 레이어
Filter expression	`FILTER_EXPRESSION`	[expression]	지도책 객체들을 필터링하는 데 사용할 표현식
Sort expression 부가적	`SORTBY_EXPRESSION`	[expression]	지도책 객체들을 정렬하는 데 사용할 표현식
Reverse sort order 부가적	`SORTBY_REVERSE`	[boolean]	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	명칭	유형	설명
Map layers to assign to unlocked map item(s) 부가적	LAYERS	[enumeration] [layer]	내용물이 잠기지 않은 맵 항목 (들) 에 표시될 레이어들
Image format	EXTENSION	[list] 기본값: png	생성된 산출물 (들) 의 파일 포맷. 사용할 수 있는 포맷 목록은 OS 와 설치된 드라이버에 따라 달라집니다.
DPI 부가적	DPI 기본값: 설정하지 않음	[number]	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	명칭	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 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

지도책 조판을 PDF 로 내보내기

인쇄 조판 지도책을 PDF 파일 (들) 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	명칭	유형	설명
Atlas layout	LAYOUT	[layout]	내보낼 조판
Coverage layer 부가적	COVERAGE_LAYER	[vector: any]	지도책을 생성하는 데 사용할 레이어
Filter expression	FILTER_EXPRESSION	[expression]	지도책 객체들을 필터링하는 데 사용할 표현식
Sort expression 부가적	SORTBY_EXPRESSION	[expression]	지도책 객체들을 정렬하는 데 사용할 표현식
Reverse sort order 부가적	SORTBY_REVERSE	[boolean]	정렬 순서를 반전할지 결정합니다. 정렬 표현식을 지정한 경우 쓰입니다.

고급 파라미터

라벨	명칭	유형	설명
Map layers to assign to unlocked map item(s) 부가적	LAYERS	[enumeration] [layer]	내용물이 잠기지 않은 맵 항목 (들) 에 표시될 레이어들
DPI 부가적	DPI 기본값: 설정하지 않음	[number]	산출 파일 (들) 의 DPI. 설정하지 않을 경우 인쇄 조판 설정값을 사용할 것입니다.
Always export as vectors	FORCE_VECTOR	[boolean] 기본값: False	벡터 데이터를 벡터로 유지할지 여부를 결정합니다.
Append georeference information	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.
Disable tiled raster layer exports	DISABLE_TILED	[boolean] 기본값: False	래스터를 타일화할지 여부를 결정합니다.
Simplify geometries to reduce output file size	SIMPLIFY	[boolean] 기본값: True	산출 파일 용량을 줄이기 위해 도형들을 단순화할지 여부를 결정합니다.
Text export	TEXT_FORMAT	[list] 기본값: 0	텍스트를 경로로 내보낼지 또는 텍스트 객체로 내보낼지 여부를 결정합니다. 다음 옵션 가운데 하나를 선택할 수 있습니다: • 0 - 항상 텍스트를 경로로 내보내기 (추천) • 1 - 항상 텍스트를 텍스트 객체로 내보내기
PDF file	OUTPUT	[file] 기본값: [Save to temporary file]	산출 파일의 (경로 포함) 이름을 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장

산출물

라벨	명칭	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 **명칭** 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	명칭	유형	설명
Print layout	LAYOUT	[layout]	내보낼 조판
Image file	OUTPUT	[file] 기본값: [Save to temporary file]	산출 파일의 (경로 포함) 이름을 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장

고급 파라미터

라벨	명칭	유형	설명
Map layers to assign to unlocked map item(s) 부가적	LAYERS	[enumeration] [layer]	내용물이 잠기지 않은 맵 항목 (들) 에 표시될 레이어들
DPI 부가적	DPI 기본값: 설정하지 않음	[number]	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	명칭	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 **명칭** 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	명칭	유형	설명
Print Layout	LAYOUT	[layout]	내보낼 조판
PDF file	OUTPUT	[file] 기본값: [Save to temporary file]	산출 파일의 (경로 포함) 이름을 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장

고급 파라미터

라벨	명칭	유형	설명
Map layers to assign to unlocked map item(s) 부가적	LAYERS	[enumeration] [layer]	내용물이 잠기지 않은 맵 항목 (들) 에 표시될 레이어들
DPI 부가적	DPI 기본값: 설정하지 않음	[number]	산출 파일 (들) 의 DPI. 설정하지 않을 경우 인쇄 조판 설정값을 사용할 것입니다.
Always export as vectors	FORCE_VECTOR	[boolean] 기본값: False	벡터 데이터를 벡터로 유지할지 여부를 결정합니다.
Append georeference information	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.
Disable tiled raster layer exports	DISABLE_TILED	[boolean] 기본값: False	래스터를 타일화할지 여부를 결정합니다.
Simplify geometries to reduce output file size	SIMPLIFY	[boolean] 기본값: True	산출 파일 용량을 줄이기 위해 도형들을 단순화할지 여부를 결정합니다.
Text export	TEXT_FORMAT	[list] 기본값: 0	텍스트를 경로로 내보낼지 또는 텍스트 객체로 내보낼지 여부를 결정합니다. 다음 옵션 가운데 하나를 선택할 수 있습니다: • 0 - 항상 텍스트를 경로로 내보내기 (추천) • 1 - 항상 텍스트를 텍스트 객체로 내보내기
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	명칭	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 **명칭** 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	명칭	유형	설명
Print layout	`LAYOUT`	[enumeration]	현재 프로젝트의 인쇄 조판
Map item 부가적	`MAP`	[enumeration] 기본값: *All the map items*	사용자가 추출하려는 정보를 가진 맵 항목 (들). 아무 항목도 지정하지 않으면 모든 맵 항목을 처리합니다.
Extent	`OUTPUT`	[vector: polygon] 기본값: `[Create temporary layer]`	범위 (들) 를 위한 산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • 데이터베이스 테이블에 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

고급 파라미터

라벨	명칭	유형	설명
Overrride CRS 부가적	CRS	[crs] 기본값: *The layout CRS*	정보를 보고할 레이어용 좌표계를 선택합니다.

산출물

라벨	명칭	유형	설명
Map height	HEIGHT	[number]	
Extent	OUTPUT	[vector: polygon]	입력 조판 맵 항목 (들) 의 범위를 모두 담고 있는 산출 폴리곤 벡터 레이어
Map rotation	ROTATION	[number]	
Map scale	SCALE	[number]	
Map width	WIDTH	[number]	

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 **명칭** 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	명칭	유형	설명
Input layer	INPUT	[layer]	스타일을 적용하려는 입력 레이어
Style file	STYLE	[file]	`.qml` 스타일 파일을 가리키는 경로

산출물

라벨	명칭	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 **명칭** 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.

파라미터

라벨	명칭	유형	설명
Input layer	INPUT	[vector: polygon]	입력 폴리곤 레이어
Minimum number of colors	MIN_COLORS	[number] 기본값: 4	할당할 색상의 최소 개수. 최소값은 1, 최대값은 1000 입니다.
Minimum distance between features	MIN_DISTANCE	[number] 기본값: 0.0	가까이 있는 (그러나 접하지는 않는) 피처에 동일한 색상을 할당하는 일을 방지합니다. 최소값은 0.0 입니다.

다음 페이지에 계속

표 24.2 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Balance color assignment	BALANCE	[enumeration] 기본값: 0	옵션은 다음과 같습니다: • 0 — 피처 개수로 각 개별 색상 인덱스가 할당된 피처의 개수가 균형을 이루도록 시도합니다. • 1 — 할당 면적으로 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. • 2 — 색상들 사이의 거리로 동일한 색상의 피처 사이의 거리를 최대화하는 방향으로 색상을 할당합니다. 이 모드는 맵 전체에서 색상이 보다 균등하게 배포되도록 해줍니다.
Colored	OUTPUT	[vector: polygon] 기본값: [Create temporary layer]	산출 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (TEMPORARY_OUTPUT) • 파일로 저장 • Geopackage 로 저장 • 데이터베이스 테이블에 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

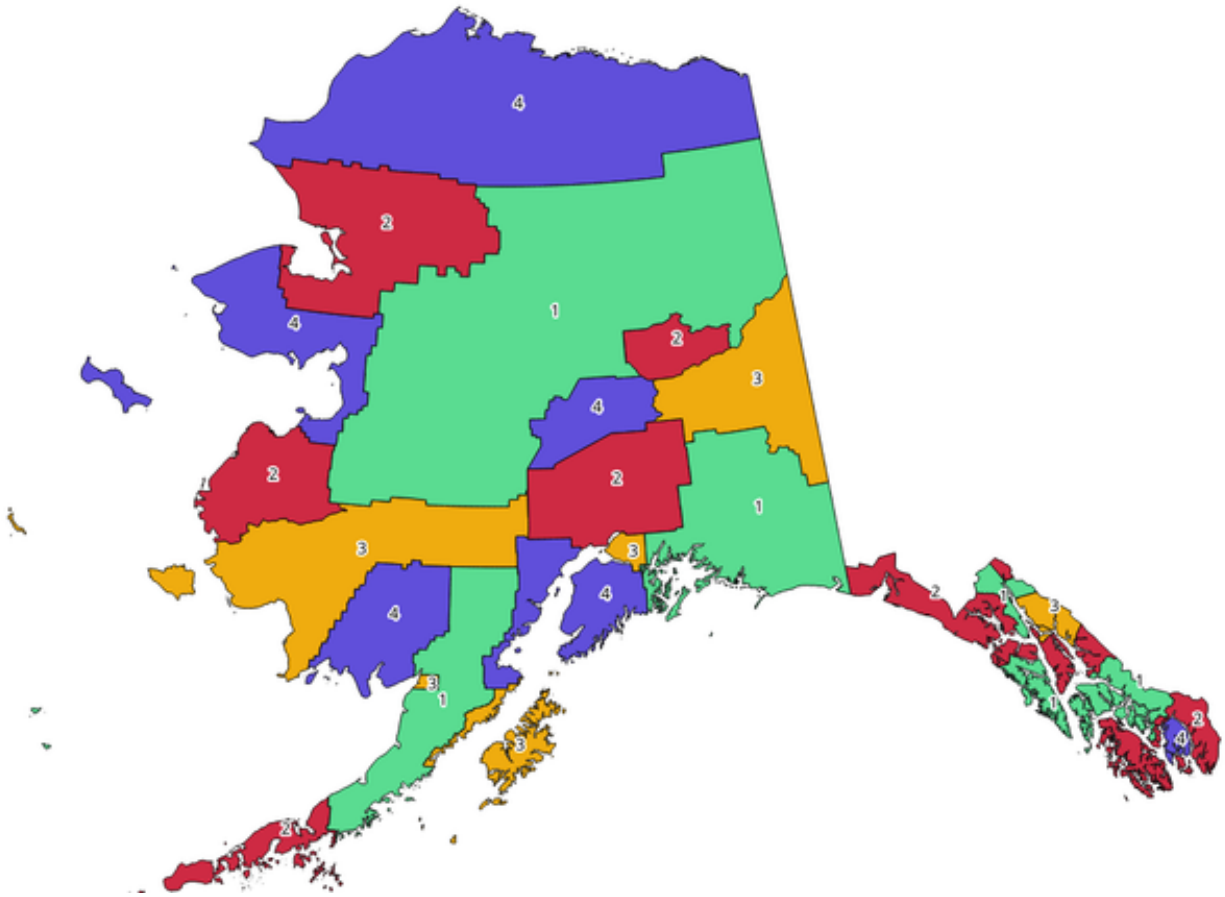


그림 24.1: 위상 색상 예시

24.1.2 데이터베이스

PostgreSQL 로 내보내기

새 관계를 생성하면서 벡터 레이어를 PostgreSQL 데이터베이스로 내보냅니다. 동일한 명칭을 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	명칭	유형	설명
Layer to import	`INPUT`	[vector: any]	데이터베이스에 추가할 벡터 레이어
Database (connection name)	`DATABASE`	[string]	데이터베이스 연결의 명칭. (데이터베이스명이 아닙니다) 기존 연결은 콤보박스에 표시될 것입니다.
Schema (schema name) 부가적	`SCHEMA`	[string] 기본값: 'public'	데이터를 저장할 스키마의 명칭. 새 스키마일 수도 있고 기존 스키마일 수도 있습니다.
Table to import to (leave blank to use layer name) 부가적	`TABLENAME`	[string] 기본값: ''	가져온 벡터 파일을 위한 테이블 명칭을 정의합니다. 아무것도 추가하지 않으면 레이어명을 사용할 것입니다.
Primary key field 부가적	`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] 기본값: 'geom'	새 PostGIS 테이블의 도형 열의 명칭을 정의합니다. 이 열에 객체에 대한 도형 정보를 저장합니다.
Encoding 부가적	`ENCODING`	[string] 기본값: 'UTF-8'	산출 레이어의 인코딩을 정의합니다.
Overwrite	`OVERWRITE`	[boolean] 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] 기본값: True	공간 인덱스를 생성할지 여부를 지정합니다.
Convert field names to lowercase	`LOWERCASE_NAMES`	[boolean] 기본값: True	입력 벡터 레이어의 필드명을 소문자로 변환합니다.
Drop length constraint on character fields	`DROP_STRING_LENGTH`	[boolean] 기본값: False	문자 필드의 길이 제약 조건을 제거할지 여부를 지정합니다.
Create single-part geometries instead of multi-part	`FORCE_SINGLEPART`	[boolean] 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

알고리즘 ID: `qgis: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 **명칭** 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	명칭	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) 부가적	`TABLENAME`	[string] 기본값: ‘’	가져온 벡터 파일을 위한 테이블 명칭을 정의합니다. 아무것도 지정하지 않으면 레이어명을 사용할 것입니다.
Primary key field 부가적	`PRIMARY_KEY`	[tablefield: any]	입력 벡터 레이어의 필드를 기본 키로 사용합니다.
Geometry column	`GEOMETRY_COLUMN`	[string] 기본값: ‘geom’	새 SpatiaLite 테이블의 도형 열의 명칭을 정의합니다. 이 열에 객체에 대한 도형 정보를 저장합니다.
Encoding 부가적	`ENCODING`	[string] 기본값: ‘UTF-8’	산출 레이어의 인코딩을 정의합니다.
Overwrite	`OVERWRITE`	[boolean] 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] 기본값: True	공간 인덱스를 생성할지 여부를 지정합니다.
Convert field names to lowercase	`LOWERCASE_NAMES`	[boolean] 기본값: True	입력 벡터 레이어의 필드명을 소문자로 변환합니다.

다음 페이지에 계속

표 24.4 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Drop length constraint on character fields	`DROP_STRING_LENGTH`	[boolean] 기본값: False	문자 필드의 길이 제약 조건을 제거할지 여부를 지정합니다.
Create single-part geometries instead of multi-part	`FORCE_SINGLEPART`	[boolean] 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

알고리즘 ID: `qgis: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 **명칭** 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	명칭	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] 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.
Save layer styles into GeoPackage	`SAVE_STYLES`	[boolean] 기본값: True	레이어 스타일을 저장합니다.
Destination GeoPackage	`OUTPUT`	[file] 기본값: `[Save to temporary file]`	GeoPackage 파일을 저장할 위치를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장…

산출물

라벨	명칭	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 **명칭** 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	명칭	유형	설명
Database (connection name)	`DATABASE`	[string]	데이터베이스 연결의 명칭. (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] 기본값: id	기본 키 필드를 (산출 테이블의 열) 설정합니다.
Geometry field name 부가적	`GEOMETRY_FIELD`	[string] 기본값: 'geom'	도형 열의 명칭을 (산출 테이블의 열) 설정합니다.

산출물

라벨	명칭	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 **명칭** 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	명칭	유형	설명
Database (connection name)	`DATABASE`	[string]	데이터베이스 연결의 명칭. (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 **명칭** 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. 이 알고리즘은 새 레이어를 생성하지 **않을 것입니다**. 이 알고리즘은 레이어 자체에 대해 쿼리를 실행하도록 설계되었습니다.

더 보기:

PostgreSQL 에서 *SQL* 실행, *SQL* 실행하기

SQL 쿼리의 몇몇 예시를 보고 싶다면*PostGIS SQL* 쿼리 예시 를 참조하세요.

파라미터

라벨	명칭	유형	설명
File Database	`DATABASE`	[vector]	연결할 SQLite/SpatiaLite 데이터베이스 파일
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: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 **명칭** 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

SQL 쿼리의 몇몇 예시를 보고 싶다면*PostGIS SQL* 쿼리 예시 를 참조하세요.

파라미터

라벨	명칭	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 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

24.1.3 파일 도구

파일 다운로드

URL 을 사용해서 (예를 들어 `http:` 또는 `file:` 을 사용해서) 지정한 파일을 다운로드합니다. 다시 말해 URL 을 복사/붙여넣기해서 파일을 다운로드할 수 있습니다.

파라미터

라벨	명칭	유형	설명
URL	`URL`	[string]	다운로드할 파일의 URL
File destination	`OUTPUT`	[string] 기본값: `[Save to temporary file]`	파일 저장 위치를 지정합니다. 다음 가운데 하나를 선택할 수 있습니다: • 산출물 건너뛰기 • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
File destination	`OUTPUT`	[string]	다운로드한 파일을 저장한 위치

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

24.1.4 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	명칭	유형	설명
Point layer	`INPUT`	[vector: point]	히트 맵에 사용할 포인트 벡터 레이어

다음 페이지에 계속

표 24.5 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Radius	RADIUS	[number] 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	[number] 기본값: 0.1	산출 래스터 레이어의 레이어 단위 픽셀 크기 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 부가적	RADIUS_FIELD	[tablefield: numeric]	입력 레이어의 속성 필드로부터 각 피처에 대한 검색 반경을 설정합니다.
Weight from field 부가적	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] 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. 여러 형태를 사용할 수 있습니다. (더 자세한 내용은 위키백과 페이지 를 참조하세요.) • 0 —4 차 다항 (Quartic) • 1 —삼각 (Triangular) • 2 —균일 (Uniform) • 3 —삼중가중 (Triweight) • 4 —예파네치니코프 (Epanechnikov)

다음 페이지에 계속

표 24.5 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Decay ratio (Triangular kernels only) 부가적	DECAY	[number] 기본값: *0.0*	삼각 커널 전용으로, 피처로부터의 거리가 멀어질수록 피처로부터의 열 (heat) 이 어떻게 낮아지는지 심도 있게 제어할 수 있습니다. • 값이 (최소값인) 0 인 경우 열이 지정한 반경의 중심에 집중되고 경계에서는 완전히 꺼질 것입니다. • 값이 0.5 인 경우 검색 반경의 경계에 있는 픽셀의 열이 중심에 있는 픽셀의 열의 반이 될 것입니다. • 값이 1 인 경우 열이 검색 반경의 원 전체에 걸쳐 균등하게 퍼질 것입니다. (이는'균일'커널과 동일합니다.) • 값이 1 보다 큰 경우 열이 경계로 갈수록 중심보다 높아질 것입니다.
Output value scaling	OUTPUT_VALUE	[enumeration] 기본값: *Raw*	산출 히트 맵 래스터의 값을 변경할 수 있습니다. 다음 가운데 하나를 선택할 수 있습니다: • 0 — 원본 (Raw) • 1 — 척도 변경 (Scaled)
Heatmap	OUTPUT	[raster] 기본값: [Save to temporary file]	커널 밀도값을 가진 산출 래스터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 24.2 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.

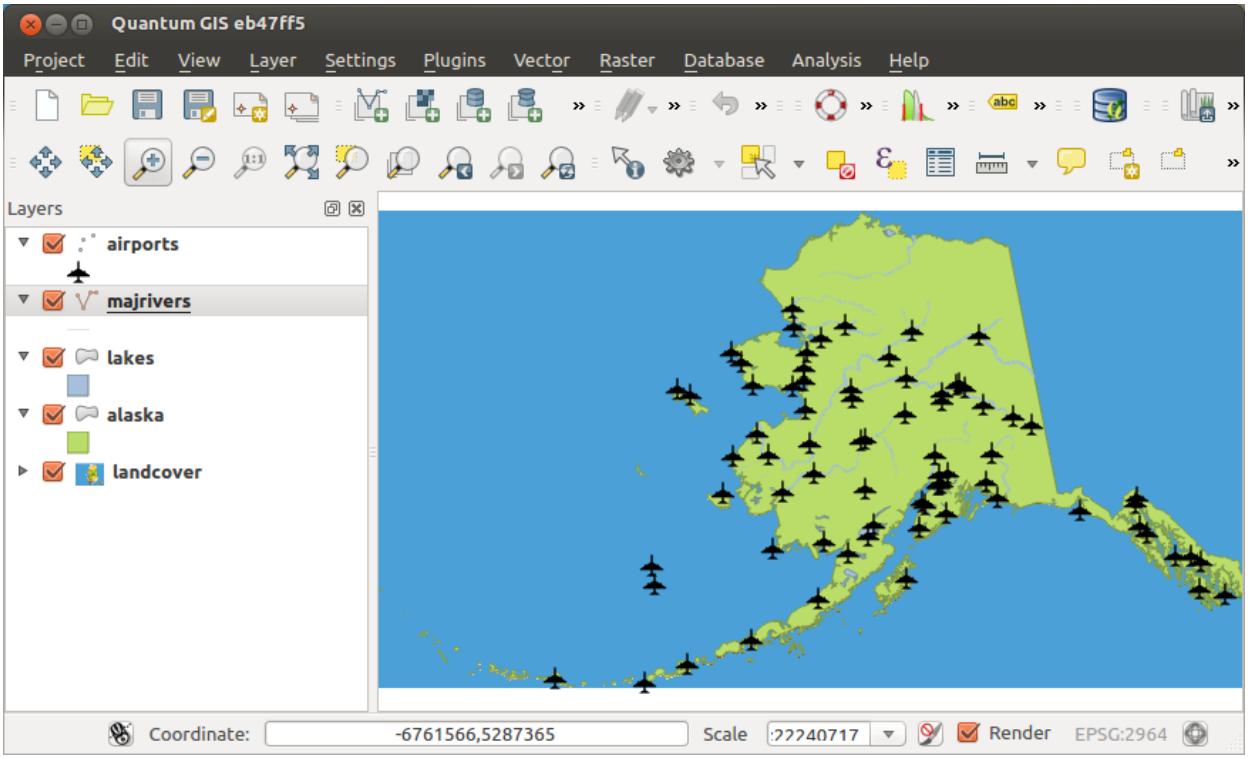


그림 24.2: 알래스카 주의 공항들

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 24.4 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 24.5 와 같습니다.

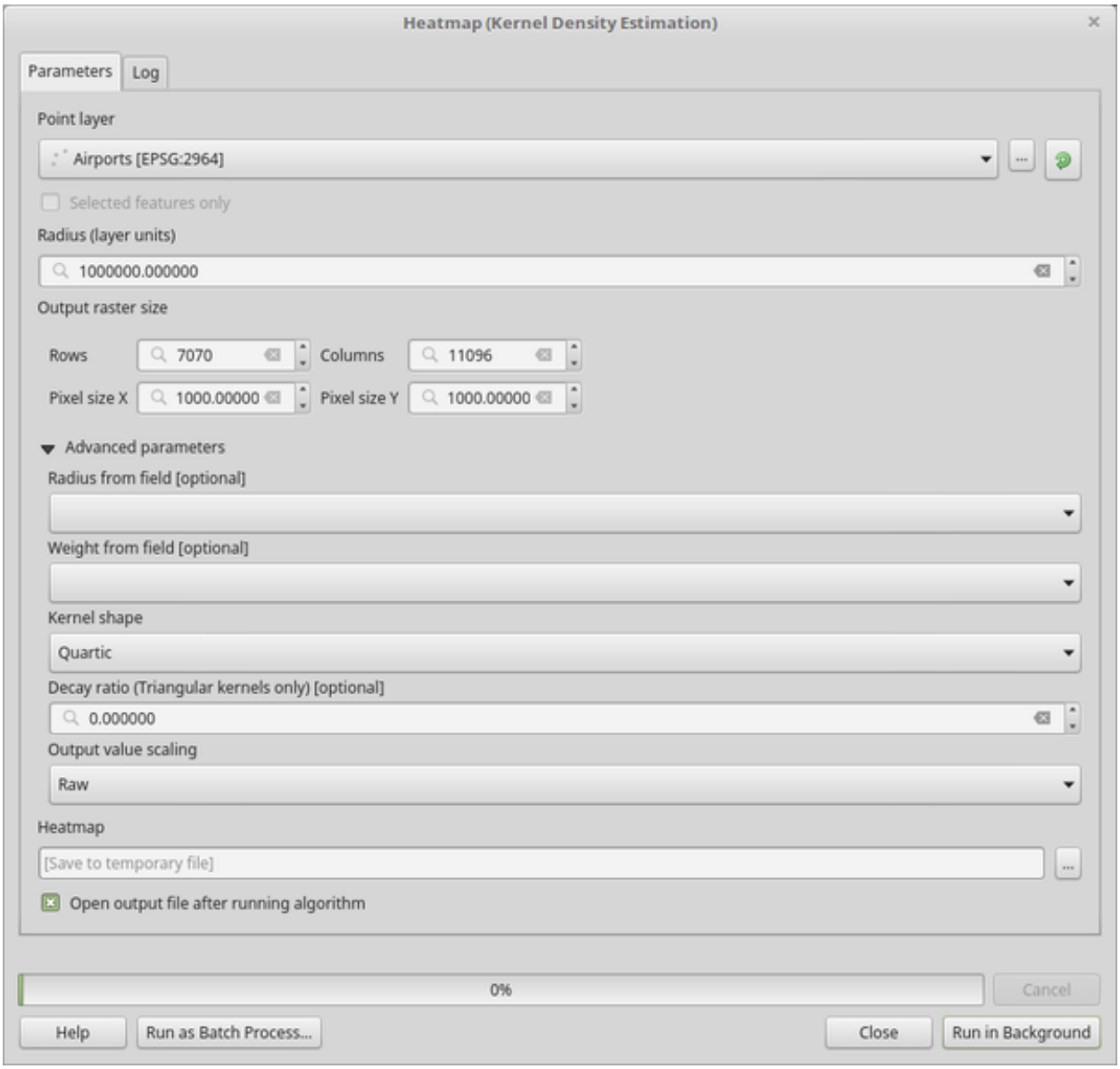


그림 24.3: 히트 맵 대화창

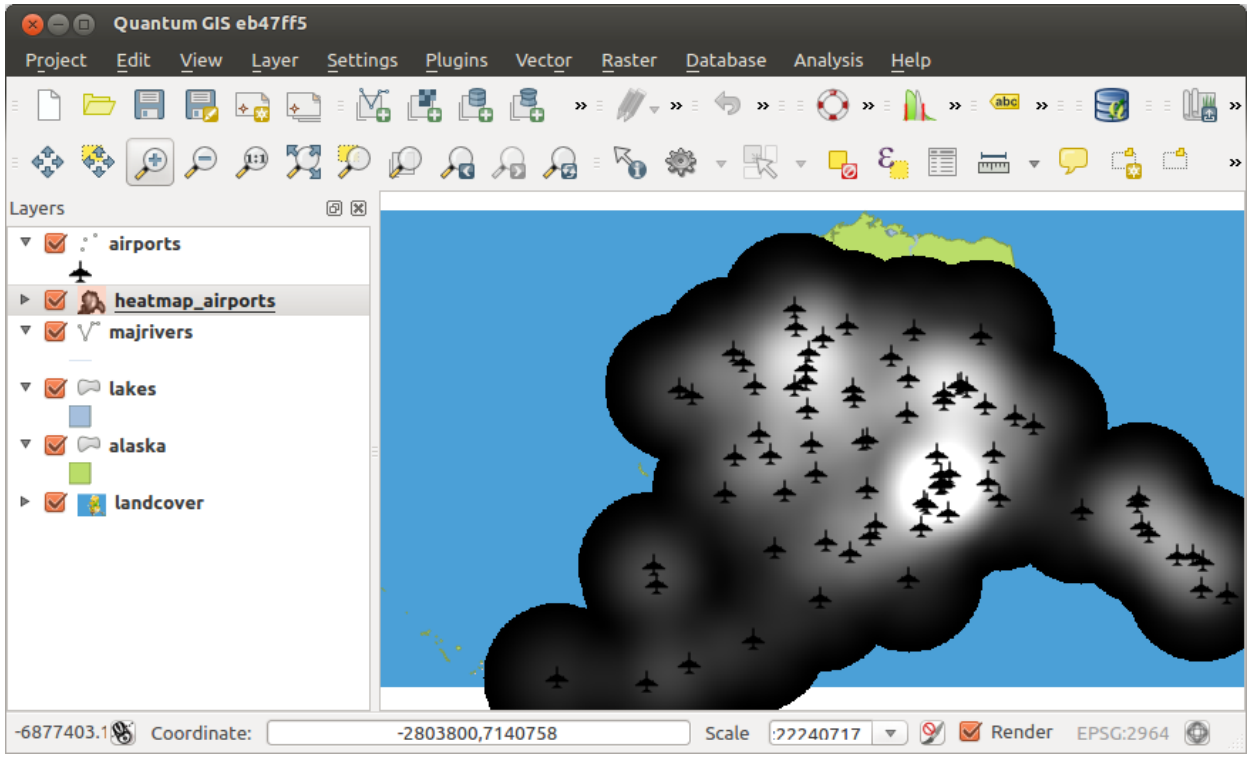


그림 24.4: 불러온 직후의 히트 맵은 회색 면처럼 보입니다

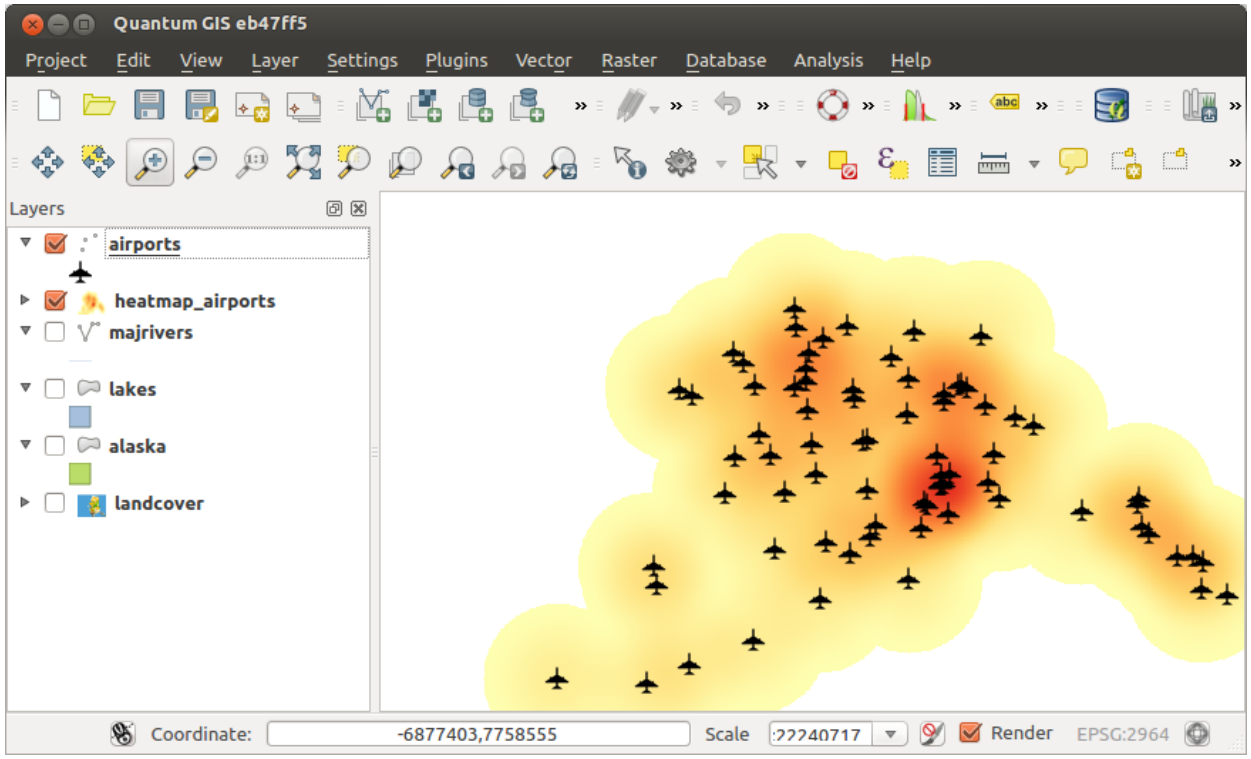


그림 24.5: 스타일을 적용한 알래스카 주 공항의 히트 맵

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 **명칭** 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	명칭	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.) 보간 데이터 문자열을 작성하기 위해 다음과 같은 GUI 항목이 제공됩니다: • **Vector layer** [vector: any] • **Interpolation attribute** [tablefield: numeric]: 보간 작업에 사용할 속성 • **Use Z-coordinate for interpolation** [boolean]: 레이어에 저장된 Z 값 (기본값: False) 추가된 각 레이어-필드 조합별로, 다음과 같은 유형을 선택할 수 있습니다: • *Points* • *Structured lines* • *Break lines* 문자열에서 레이어-필드 항목은 `'::|::'` 로 구분됩니다. 레이어-필드 항목의 하위 항목은 `'::~::'` 로 구분됩니다.
Distance coefficient P	`DISTANCE_COEFFICIENT`	[number] Default: 2.0	보간 작업을 위한 거리 계수 (coefficient) 를 설정합니다. 최소값은 0.0, 최대값은 100.0 입니다.

다음 페이지에 계속

표 24.7 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Extent (xmin, xmax, ymin, ymax)	`EXTENT`	[extent]	산출 래스터 레이어의 범위입니다. 맵 캔버스에서 범위를 선택하거나, 다른 레이어로부터 선택하거나, 또는 직접 입력해서 산출물의 범위를 선언해야 합니다.
Output raster size	`PIXEL_SIZE`	[number] 기본값: 0.1	산출 래스터 레이어의 레이어 단위 픽셀 크기 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]`	보간된 값을 가질 래스터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

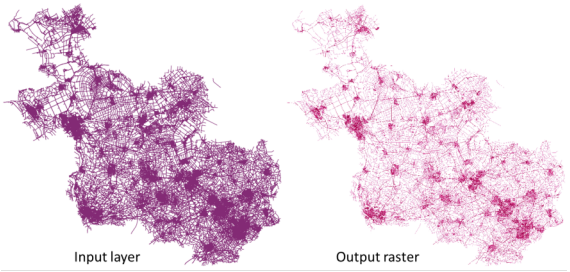


그림 24.6: 라인 밀도의 예. 입력 레이어 소스: 네덜란드 오퍼레이셀 주의 도로 (OSM)

파라미터

라벨	명칭	유형	설명
Input line layer	`INPUT`	[vector: any]	라인 객체를 담고 있는 입력 벡터 레이어
Weight field	`WEIGHT`	[number]	계산 도중 사용할 가중치 인자를 담고 있는 레이어 필드
Search Radius	`RADIUS`	[number] 기본값: 10	원형 구역의 반경. 여기에 단위를 지정할 수 있습니다.
Pixel size	`PIXEL_SIZE`	[number] 기본값: 10	산출 래스터 레이어의 레이어 단위 픽셀 크기. 래스터 픽셀은 정사각형입니다.
Line density raster	`OUTPUT`	[raster] 기본값: `[Save to temporary file]`	산출 래스터 레이어입니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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	명칭	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.) 보간 데이터 문자열을 작성하기 위해 다음과 같은 GUI 항목이 제공됩니다: • **Vector layer** [vector: any] • **Interpolation attribute** [tablefield: numeric]: 보간 작업에 사용할 속성 • **Use Z-coordinate for interpolation** [boolean]: 레이어에 저장된 Z 값 (기본값: False) 추가된 각 레이어-필드 조합별로, 다음과 같은 유형을 선택할 수 있습니다: • *Points* • *Structured lines* • *Break lines* 문자열에서 레이어-필드 항목은 `'::|::'` 로 구분됩니다. 레이어-필드 항목의 하위 항목은 `'::~::'` 로 구분됩니다.
Interpolation method	`METHOD`	[enumeration] 기본값: 0	사용할 보간법을 설정합니다. 다음 가운데 하나를 선택할 수 있습니다: • *Linear* • *Clough-Toucher (cubic)*

다음 페이지에 계속

표 24.11 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Extent (xmin, xmax, ymin, ymax)	EXTENT	[extent]	산출 래스터 레이어의 범위입니다. 맵 캔버스에서 범위를 선택하거나, 다른 레이어로부터 선택하거나, 또는 직접 입력해서 산출물의 범위를 선언해야 합니다.
Output raster size	PIXEL_SIZE	[number] 기본값: 0.1	산출 래스터 레이어의 레이어 단위 픽셀 크기 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]	삼각불규칙망 보간의 산출물을 저장할 래스터 레이어입니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.
Triangulation	TRIANGULATION	[vector: line] 기본값: [Skip output]	삼각불규칙망 보간의 산출물을 저장할 벡터 레이어입니다. 다음 가운데 하나로 저장할 수 있습니다: • 산출물 건너뛰기 • 임시 레이어 생성 (TEMPORARY_OUTPUT) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장

산출물

라벨	명칭	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 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

24.1.5 레이어 도구

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.

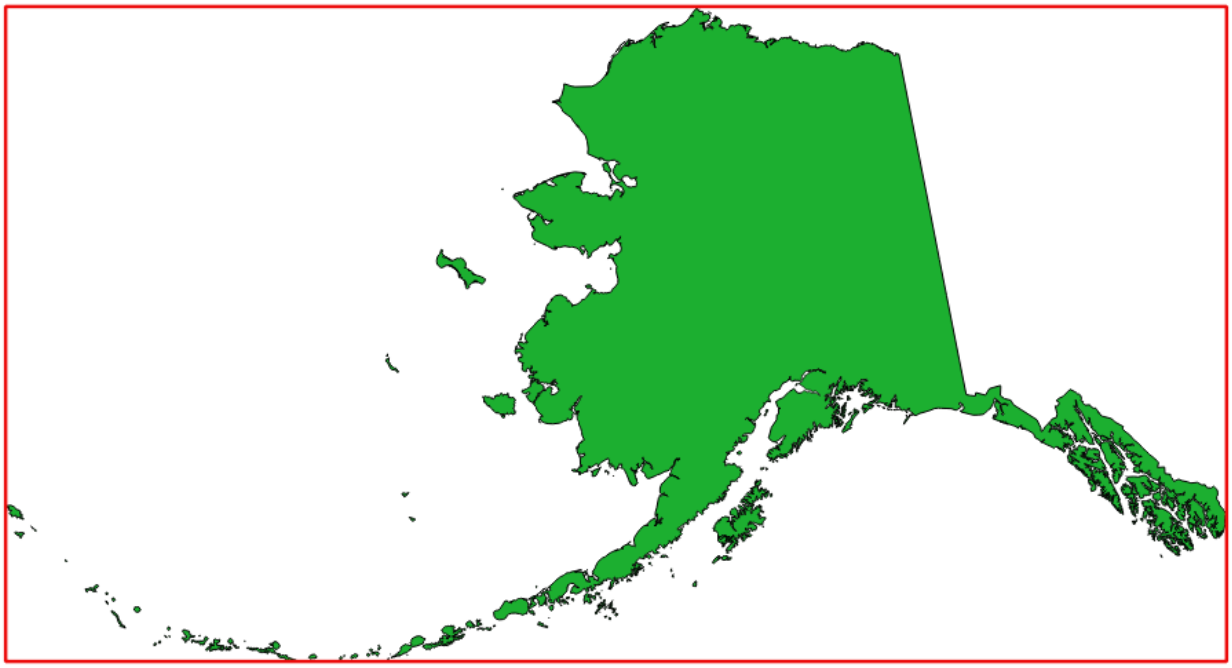


그림 24.7: 소스 레이어의 빨간색 경계 상자

기본 메뉴: *Vector → Research Tools*

파라미터

라벨	명칭	유형	설명
Layer	INPUT	[layer]	입력 레이어
Extent	OUTPUT	[vector: polygon] 기본값: [Create temporary layer]	산출물 범위용 폴리곤 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (TEMPORARY_OUTPUT) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Extent	OUTPUT	[vector: polygon]	범위를 (최소 경계 상자를) 가진 산출물 (폴리곤) 벡터 레이어

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

24.1.6 모델 생성기 도구

이 도구들은 그래픽 모델 생성기에서만 사용할 수 있습니다. 공간 처리 툴박스에서 사용할 수는 없습니다.

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	명칭	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 **명칭** 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	명칭	유형	설명
Layer	INPUT	[layer]	범례에 불러올 레이어
Loaded layer name	NAME	[string]	불러온 레이어의 명칭

산출물

라벨	명칭	유형	설명
Layer	OUTPUT	[same as input]	(재명명된) 불러온 레이어

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 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	명칭	유형	설명
Message	`MESSAGE`	[string]	표시할 메시지
Condition	`CONDITION`	[expression]	참인지 평가할 표현식

산출물

없음

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 **명칭** 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	명칭	유형	설명
Message	MESSAGE	[string]	표시할 메시지
Condition	CONDITION	[expression]	참인지 평가할 표현식

산출물

없음

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 **명칭** 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	명칭	유형	설명
Layer	INPUT	[layer]	재명명할 레이어
New name	NAME	[string]	레이어의 새 명칭

산출물

라벨	명칭	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 **명칭** 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	명칭	유형	설명
Use HTML	`USE_HTML`	[Boolean] 기본값: False	HTML 서식 사용 여부

산출물

라벨	명칭	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 **명칭** 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	명칭	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 **명칭** 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 생성기에서 문자열 2 개를 하나로 연결 (concatenation) 합니다.

파라미터

라벨	명칭	유형	설명
Input 1	INPUT_1	[string]	첫 번째 문자열
Input 2	INPUT_2	[string]	두 번째 문자열

산출물

라벨	명칭	유형	설명
Concatenation	CONCATENATION	[string]	연결된 문자열

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

24.1.7 망 분석

서비스 영역 (레이어로부터)

포인트 레이어에서 시작해서 어떤 거리 또는 시간 안에 도달할 수 있는 망 (network) 의 모든 경계 또는 경계의 일부를 반환합니다. 이 알고리즘은 망 내에서의 접근성을 평가할 수 있습니다. 예를 들어 지정한 값보다 비용을 초과하지 않으면서 (여기서 비용이란 거리 또는 시간일 수 있습니다) 도로망에서 탐색할 수 있는 위치는 어디인지 알 수 있습니다.

파라미터

라벨	명칭	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: • 0 — 최단 (Shortest) • 1 — 최속 (Fastest)
Travel cost (distance for "Shortest", time for "Fastest")	`TRAVEL_COST`	[number] 기본값: 0	최단 경로를 탐색하는 경우 (망 레이어 단위의) 거리로 추정한 값, 최속 경로의 경우 (시 단위의) 시간으로 추청한 값

Direction field 부가적	DIREC- TION_FIELD	[tablefield: string] 기본값: 0.0	망 경계에 대해 방향을 지정하는 데 사용하는 필드입니다. 이 필드에서 사용되는 값은 Value for forward direction, Value for backward direction 및 Value for both directions 파라미터 3 개로 지정됩니다. 순방향 (forward direction) 및 역방향 (backward direction) 은 일방통행 경계에 대응하며, "양방향 (both directions)" 은 쌍방통행 경계에 대응합니다. 피처가 이 필드에 값을 가지고 있지 않거나 아무 필드도 설정하지 않은 경우 (Default direction 파라미터로 지정한) 기본 방향을 사용합니다.
Value for forward direction 부가적	VALUE_FORWARD	[string] 기본값: '' (empty string)	순방향 경계를 식별하기 위해 방향 필드에서 설정한 값
Value for backward direction 부가적	VALUE_BACKWARD	[string] 기본값: '' (empty string)	역방향 경계를 식별하기 위해 방향 필드에서 설정한 값
Value for both directions 부가적	VALUE_BOTH	[string] 기본값: '' (empty string)	양방향 경계를 식별하기 위해 방향 필드에서 설정한 값
Default direction 부가적	DE- FAULT_DIRECTION	[enumeration] 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: • 0 — 순방향 • 1 — 역방향 • 2 — 양방향
Speed field 부가적	SPEED_FIELD	[tablefield: string]	최속 경로를 탐색하는 경우 망의 경계의 (km/h 단위의) 속도 값을 지정하는 필드 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) 부가적	DEFAULT_SPEED	[number] Default: 50.0	경계에 대해 아무 속도 필드도 지정하지 않은 경우 이동 시간을 계산하는 데 사용할 값
Topology tolerance 부가적	TOLERANCE	[number] 기본값: 0.0	지정한 허용 오차보다 가까이 있는 노드를 가진 라인 2 개를 연결된 것으로 간주합니다.

Include upper/lower bound points	IN- CLUDE_BOUNDS	[boolean] 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.

다음 페이지에 계속

표 24.15 – 이전 페이지에서 계속

Service area (lines)	OUTPUT_LINES	[vector: line] 기본값: [Create temporary layer]	서비스 영역을 담은 산출 라인 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 산출물 건너뛰기 • 임시 레이어 생성 (TEMPORARY_OUTPUT) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.
Service area (boundary nodes)	OUTPUT	[vector: point] 기본값: [Skip output]	서비스 영역의 경계 노드를 담은 산출 포인트 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 산출물 건너뛰기 • 임시 레이어 생성 (TEMPORARY_OUTPUT) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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: `qgis: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 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

서비스 영역 (포인트로부터)

포인트 피처에서 시작해서 지정한 거리 또는 시간 안에 도달할 수 있는 망 (network) 의 모든 경계 또는 경계의 일부를 반환합니다. 이 알고리즘은 망 내에서의 접근성을 평가할 수 있습니다. 예를 들어 지정한 값보다 비용을 초과하지 않으면서 (여기서 비용이란 거리 또는 시간일 수 있습니다) 도로망에서 탐색할 수 있는 위치는 어디인지 알 수 있습니다.

파라미터

라벨	명칭	유형	설명
Vector layer representing the network	`INPUT`	[vector: line]	커버할 망을 표현하는 라인 벡터 레이어
Start point (x, y)	`START_POINT`	[coordinates]	그 주위에 서비스 영역을 계산할 포인트의 좌표
Path type to calculate	`STRATEGY`	[enumeration] 기본값: 0	계산할 경로의 유형을 다음 가운데 하나로 선택할 수 있습니다: • 0 — 최단 (Shortest) • 1 — 최속 (Fastest)
Travel cost	`TRAVEL_COST`	[number] 기본값: 0	최단 경로를 탐색하는 경우 (망 레이어 단위의) 거리로 추정한 값, 최속 경로의 경우 (시 단위의) 시간으로 추청한 값
Advanced parameters	GUI 전용		고급 망분석 파라미터 그룹 — 다음 절 참조
Service area (lines)	`OUTPUT_LINES`	[vector: line] 기본값: `[Create temporary layer]`	서비스 영역을 담은 산출 라인 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 산출물 건너뛰기 • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.
Service area (boundary nodes)	`OUTPUT`	[vector: point] 기본값: `[Skip output]`	서비스 영역의 경계 노드를 담은 산출 포인트 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 산출물 건너뛰기 • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

고급 파라미터

라벨	명칭	유형	설명
Direction field 부가적	DIREC- TION_FIELD	[tablefield: string] 기본값: 0.0	망 경계에 대해 방향을 지정하는 데 사용하는 필드입니다. 이 필드에서 사용되는 값은 Value for forward direction, Value for backward direction 및 Value for both directions 파라미터 3 개로 지정됩니다. 순방향 (forward direction) 및 역방향 (backward direction) 은 일방통행 경계에 대응하며, "양방향 (both directions)" 은 쌍방통행 경계에 대응합니다. 피처가 이 필드에 값을 가지고 있지 않거나 아무 필드도 설정하지 않은 경우 (Default direction 파라미터로 지정한) 기본 방향을 사용합니다.
Value for forward direction 부가적	VALUE_FORWARD	[string] 기본값: '' (empty string)	순방향 경계를 식별하기 위해 방향 필드에서 설정한 값
Value for backward direction 부가적	VALUE_BACKWARD	[string] 기본값: '' (empty string)	역방향 경계를 식별하기 위해 방향 필드에서 설정한 값
Value for both directions 부가적	VALUE_BOTH	[string] 기본값: '' (empty string)	양방향 경계를 식별하기 위해 방향 필드에서 설정한 값
Default direction 부가적	DE- FAULT_DIRECTION	[enumeration] 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: • 0 — 순방향 • 1 — 역방향 • 2 — 양방향
Speed field 부가적	SPEED_FIELD	[tablefield: string]	최속 경로를 탐색하는 경우 망의 경계의 (km/h 단위의) 속도 값을 지정하는 필드 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) 부가적	DEFAULT_SPEED	[number] Default: 50.0	경계에 대해 아무 속도 필드도 지정하지 않은 경우 이동 시간을 계산하는 데 사용할 값
Topology tolerance 부가적	TOLERANCE	[number] 기본값: 0.0	지정한 허용 오차보다 가까이 있는 노드를 가진 라인 2 개를 연결된 것으로 간주합니다.
Include upper/lower bound points	IN- CLUDE_BOUNDS	[boolean] 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	명칭	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: `qgis: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 **명칭** 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.

파라미터

라벨	명칭	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: • 0 — 최단 (Shortest) • 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
Advanced parameters	GUI 전용		**고급 파라미터** 그룹:

다음 페이지에 계속

표 24.18 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Direction field 부가적	DIREC- TION_FIELD	[tablefield: string] 기본값: 0.0	망 경계에 대해 방향을 지정하는 데 사용하는 필드입니다. 이 필드에서 사용되는 값은 Value for forward direction, Value for backward direction 및 Value for both directions 파라미터 3 개로 지정됩니다. 순방향 (forward direction) 및 역방향 (backward direction) 은 일방통행 경계에 대응하며, "양방향 (both directions)" 은 쌍방통행 경계에 대응합니다. 피처가 이 필드에 값을 가지고 있지 않거나 아무 필드도 설정하지 않은 경우 (Default direction 파라미터로 지정한) 기본 방향을 사용합니다.
Value for forward direction 부가적	VALUE_FORWARD	[string] 기본값: '' (empty string)	순방향 경계를 식별하기 위해 방향 필드에서 설정한 값
Value for backward direction 부가적	VALUE_BACKWARD	[string] 기본값: '' (empty string)	역방향 경계를 식별하기 위해 방향 필드에서 설정한 값
Value for both directions 부가적	VALUE_BOTH	[string] 기본값: '' (empty string)	양방향 경계를 식별하기 위해 방향 필드에서 설정한 값
Default direction 부가적	DE- FAULT_DIRECTION	[enumeration] 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: • 0 —순방향 • 1 —역방향 • 2 —양방향
Speed field 부가적	SPEED_FIELD	[tablefield: string]	최속 경로를 탐색하는 경우 망의 경계의 (km/h 단위의) 속도 값을 지정하는 필드 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) 부가적	DEFAULT_SPEED	[number] Default: 50.0	경계에 대해 아무 속도 필드도 지정하지 않은 경우 이동 시간을 계산하는 데 사용할 값
Topology tolerance 부가적	TOLERANCE	[number] 기본값: 0.0	지정한 허용 오차보다 가까이 있는 노드를 가진 라인 2 개를 연결된 것으로 간주합니다.
			고급 파라미터 그룹 끝

다음 페이지에 계속

표 24.18 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Shortest path	OUTPUT	[vector: line]	최단 경로를 담은 산출 라인 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (TEMPORARY_OUTPUT) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Shortest path	OUTPUT	[vector: line]	각 시작점부터 종단점까지의 최단 또는 최속 경로를 담은 라인 레이어

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 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.

파라미터

라벨	명칭	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: • 0 — 최단 (Shortest) • 1 — 최속 (Fastest)
Start point (x, y)	START_POINT	[coordinates]	경로의 시작점을 표현하는 포인트 피처
Vector layer with end points	END_POINTS	[vector: point]	경로의 종단점으로 사용되는 피처를 가진 포인트 벡터 레이어

다음 페이지에 계속

표 24.19 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Direction field 부가적인 고급 파라미터	DIREC- TION_FIELD	[tablefield: string] 기본값: 0.0	망 경계에 대해 방향을 지정하는 데 사용하는 필드입니다. 이 필드에서 사용되는 값은 Value for forward direction, Value for backward direction 및 Value for both directions 파라미터 3 개로 지정됩니다. 순방향 (forward direction) 및 역방향 (backward direction) 은 일방통행 경계에 대응하며, "양방향 (both directions)" 은 쌍방통행 경계에 대응합니다. 피처가 이 필드에 값을 가지고 있지 않거나 아무 필드도 설정하지 않은 경우 (Default direction 파라미터로 지정한) 기본 방향을 사용합니다.
Value for forward direction 부가적인 고급 파라미터	VALUE_FORWARD	[string] 기본값: '' (empty string)	순방향 경계를 식별하기 위해 방향 필드에서 설정한 값
Value for backward direction 부가적인 고급 파라미터	VALUE_BACKWARD	[string] 기본값: '' (empty string)	역방향 경계를 식별하기 위해 방향 필드에서 설정한 값
Value for both directions 부가적인 고급 파라미터	VALUE_BOTH	[string] 기본값: '' (empty string)	양방향 경계를 식별하기 위해 방향 필드에서 설정한 값
Default direction 부가적인 고급 파라미터	DE- FAULT_DIRECTION	[enumeration] 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: • 0 —순방향 • 1 —역방향 • 2 —양방향
Speed field 부가적인 고급 파라미터	SPEED_FIELD	[tablefield: string]	최속 경로를 탐색하는 경우 망의 경계의 (km/h 단위의) 속도 값을 지정하는 필드 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) 부가적인 고급 파라미터	DEFAULT_SPEED	[number] Default: 50.0	경계에 대해 아무 속도 필드도 지정하지 않은 경우 이동 시간을 계산하는 데 사용할 값
Topology tolerance 부가적인 고급 파라미터	TOLERANCE	[number] 기본값: 0.0	지정한 허용 오차보다 가까이 있는 노드를 가진 라인 2 개를 연결된 것으로 간주합니다.

다음 페이지에 계속

표 24.19 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Shortest path	`OUTPUT`	[vector: line]	최단 경로를 담은 산출 라인 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Shortest path	`OUTPUT`	[vector: line]	각 시작점부터 종단점까지의 최단 또는 최속 경로를 담은 라인 레이어

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 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.

파라미터

라벨	명칭	고급 파라미터 여부	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: • 0 — 최단 (Shortest) • 1 — 최속 (Fastest)

다음 페이지에 계속

표 24.20 – 이전 페이지에서 계속

라벨	명칭	고급 파라미터 여부	유형	설명
Start point (x, y)	START_POINT		[coordinates]	경로의 시작점을 표현하는 포인트 피처
End point (x, y)	END_POINT		[coordinates]	경로의 종단점을 표현하는 포인트 피처
Direction field 부가적	DIRECTION_FIELD	X	[tablefield: string] 기본값: 0.0	망 경계에 대해 방향을 지정하는 데 사용하는 필드입니다. 이 필드에서 사용되는 값은 Value for forward direction, Value for backward direction 및 Value for both directions 파라미터 3 개로 지정됩니다. 순방향 (forward direction) 및 역방향 (backward direction) 은 일방통행 경계에 대응하며, "양방향 (both directions)"은 쌍방통행 경계에 대응합니다. 피처가 이 필드에 값을 가지고 있지 않거나 아무 필드도 설정하지 않은 경우 (Default direction 파라미터로 지정한) 기본 방향을 사용합니다.
Value for forward direction 부가적	VALUE_FORWARD	X	[string] 기본값: '' (empty string)	순방향 경계를 식별하기 위해 방향 필드에서 설정한 값
Value for backward direction 부가적	VALUE_BACKWARD	X	[string] 기본값: '' (empty string)	역방향 경계를 식별하기 위해 방향 필드에서 설정한 값
Value for both directions 부가적	VALUE_BOTH	X	[string] 기본값: '' (empty string)	양방향 경계를 식별하기 위해 방향 필드에서 설정한 값
Default direction 부가적	DEFAULT_DIRECTION	X	[enumeration] 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: • 0 —순방향 • 1 —역방향 • 2 —양방향

다음 페이지에 계속

표 24.20 – 이전 페이지에서 계속

라벨	명칭	고급 파라미터 여부	유형	설명
Speed field 부가적	SPEED_FIELD	X	[tablefield: string]	최속 경로를 탐색하는 경우 망의 경계의 (km/h 단위의) 속도 값을 지정하는 필드 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) 부가적	DEFAULT_SPEED	X	[number] Default: 50.0	경계에 대해 아무 속도 필드도 지정하지 않은 경우 이동 시간을 계산하는 데 사용할 값
Topology tolerance 부가적	TOLERANCE	X	[number] 기본값: 0.0	지정한 허용 오차보다 가까이 있는 노드를 가진 라인 2 개를 연결된 것으로 간주합니다.
Shortest path	OUTPUT		[vector: line]	최단 경로를 담은 산출 라인 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (TEMPORARY_OUTPUT) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Shortest path	OUTPUT	[vector: line]	각 시작점부터 종단점까지의 최단 또는 최속 경로를 담은 라인 레이어

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

24.1.8 도표

막대 그래프

범주 및 레이어 필드에서 막대 그래프 (bar plot) 를 색성합니다.

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	입력 벡터 레이어
Category field name	`NAME_FIELD`	[tablefield: any]	(X 축) 막대를 그룹화하는 데 사용할 범주 필드
Value field	`VALUE_FIELD`	[tablefield: any]	(Y 축) 그래프에 사용할 값
Bar plot	`OUTPUT`	[html] 기본값: `[Save to temporary file]`	그래프용 HTML 파일을 지정합니다. 다음 가운데 하나를 선택할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	입력 벡터 레이어
Category name field	`NAME_FIELD`	[tablefield: any]	(X 축) 상자를 그룹화하는 데 사용할 범주 필드
Value field	`VALUE_FIELD`	[tablefield: any]	(Y 축) 그래프에 사용할 값
Additional statistic lines	`MSD`	[enumeration] 기본값: 0	그래프에 추가할 부가적인 통계 정보입니다. 다음 가운데 하나를 선택할 수 있습니다: • 0 —평균 표시 • 1 —표준 편차 표시 • 2 —평균 및 표준 편차를 표시하지 않음
Box plot	`OUTPUT`	[html] 기본값: `[Save to temporary file]`	그래프용 HTML 파일을 지정합니다. 다음 가운데 하나를 선택할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Box plot	`OUTPUT`	[html]	그래프 HTML 파일. *Processing → Result Viewer* 메뉴로 볼 수 있습니다.

파이썬 코드

알고리즘 ID: `qgis:boxplot`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 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.

파라미터

라벨	명칭	유형	설명
Input table	`INPUT`	[vector: any]	입력 벡터 레이어
Category name field	`NAME_FIELD`	[tablefield: any]	(X 축) 상자를 그룹화하는 데 사용할 범주 필드
Value field	`VALUE_FIELD`	[tablefield: any]	(Y 축) 그래프에 사용할 값
Plot	`OUTPUT`	[html] 기본값: `[Save to temporary file]`	그래프용 HTML 파일을 지정합니다. 다음 가운데 하나를 선택할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

파라미터

라벨	명칭	유형	설명
Input layer	INPUT	[vector: any]	입력 벡터 레이어
Category name field	NAME_FIELD	[tablefield: any]	(X 축) 피처를 그룹화하는 데 사용할 범주 필드
Value field	VALUE_FIELD	[tablefield: any]	(Y 축) 그래프에 사용할 값
Polar plot	OUTPUT	[html] 기본값: [Save to temporary file]	그래프용 HTML 파일을 지정합니다. 다음 가운데 하나를 선택할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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	명칭	유형	설명
Input layer	`INPUT`	[raster]	입력 래스터 레이어
Band number	`BAND`	[raster band]	히스토그램에 사용할 래스터 밴드
number of bins	`BINS`	[number] 기본값: 10	히스토그램에 사용할 (X 축) 계급 간격 (bin) 의 개수. 최소값은 2 입니다.
Histogram	`OUTPUT`	[html] 기본값: `[Save to temporary file]`	그래프용 HTML 파일을 지정합니다. 다음 가운데 하나를 선택할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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	명칭	유형	설명
Input layer	INPUT	[vector: any]	입력 벡터 레이어
Attribute	FIELD	[tablefield: any]	(Y 축) 그래프에 사용할 값
number of bins	BINS	[number] 기본값: 10	히스토그램에 사용할 (X 축) 계급 간격 (bin) 의 개수. 최소값은 2 입니다.
Histogram	OUTPUT	[html] 기본값: [Save to temporary file]	그래프용 HTML 파일을 지정합니다. 다음 가운데 하나를 선택할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

파라미터

라벨	명칭	유형	설명
Input layer	INPUT	[vector: any]	입력 벡터 레이어
X attribute	XFIELD	[tablefield: any]	X 축에 사용할 필드
Y attribute	YFIELD	[tablefield: any]	Y 축에 사용할 필드
Scatterplot	OUTPUT	[html] 기본값: [Save to temporary file]	그래프용 HTML 파일을 지정합니다. 다음 가운데 하나를 선택할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	입력 벡터 레이어
X attribute	`XFIELD`	[tablefield: any]	X 축에 사용할 필드
Y attribute	`YFIELD`	[tablefield: any]	Y 축에 사용할 필드
Z attribute	`ZFIELD`	[tablefield: any]	Z 축에 사용할 필드
Histogram	`OUTPUT`	[html] 기본값: `[Save to temporary file]`	그래프용 HTML 파일을 지정합니다. 다음 가운데 하나를 선택할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Histogram	`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 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

24.1.9 래스터 분석

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.

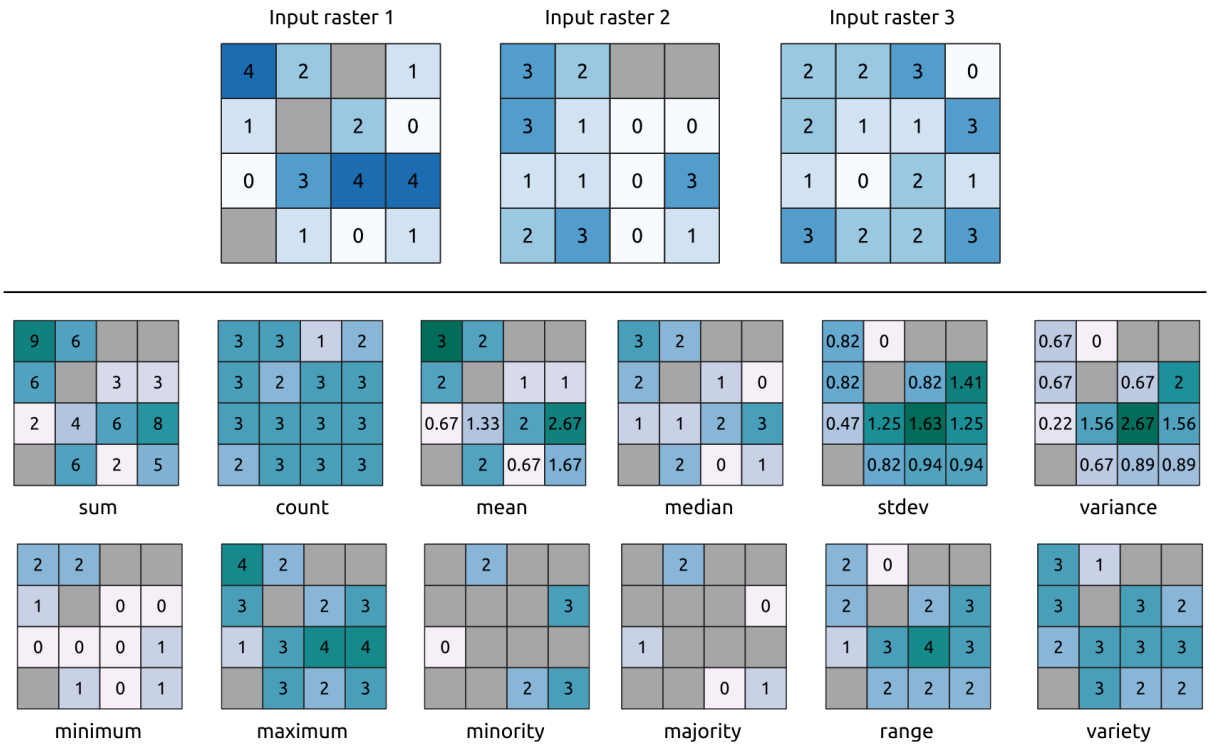


그림 24.8: 모든 통계 함수의 예. `NoData` (회색) 셀도 계산에 넣습니다.

파라미터

라벨	명칭	유형	설명
Input layers	INPUT	[raster] [list]	입력 래스터 레이어
Statistic	STATISTIC	[enumeration] 기본값: 0	다음과 같은 통계를 낼 수 있습니다: • 0 —합계 (Sum) • 1 —개수 (Count) • 2 —평균 (Mean) • 3 —중간값 (Median) • 4 —표준 편차 (Standard deviation) • 5 —분산 (Variance) • 6 —최소값 (Minimum) • 7 —최대값 (Maximum) • 8 —희귀값 (Minority, 가장 드문 값) • 9 —최빈값 (Majority, 가장 흔한 값) • 10 —범위 (Range, 최대 ~ 최소) • 11 —다양도 (Variety, 유일값 개수)
Ignore NoData values	IGNORE_NODATA	[boolean] 기본값: True	NODATA 셀을 무시하고 중첩한 모든 셀에 대해 통계를 계산합니다.
Reference layer	REF_LAYER	[raster]	산출 레이어 생성 시 (범위, 좌표계, 픽셀 크기를) 참조할 참조 레이어
Output no data value 부가적	OUTPUT_NO_DATA_VALUE	[number] 기본값: -9999.0	산출 레이어에서 NODATA 용으로 사용할 값
Output layer	OUTPUT	[same as input]	산출 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
CRS authority identifier	CRS_AUTHID	[crs]	산출 래스터 레이어의 좌표계
Extent	EXTENT	[extent]	산출 래스터 레이어의 공간 영역
Height in pixels	HEIGHT_IN_PIXELS	[integer]	산출 래스터 레이어의 픽셀 단위 높이
Output raster	OUTPUT	[raster]	결과물을 담고 있는 산출 래스터 레이어
Total pixel count	TOTAL_PIXEL_COUNT	[integer]	산출 래스터 레이어의 픽셀 개수
Width in pixels	WIDTH_IN_PIXELS	[integer]	산출 래스터 레이어의 픽셀 단위 너비

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 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.

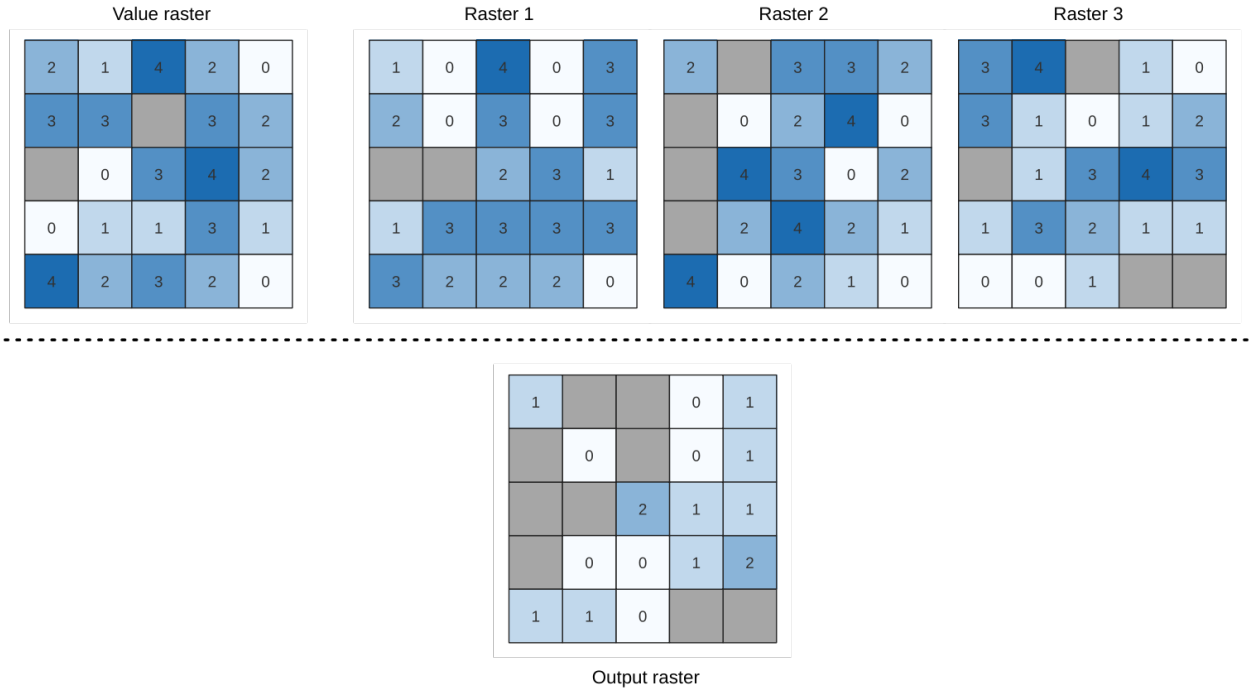


그림 24.9: 산출 래스터에 있는 각 셀의 값은 래스터 목록에서 대응하는 셀의 값이 값 래스터의 해당 셀의 값과 동일한 횟수를 나타냅니다. `NoData` (회색) 셀도 고려합니다.

더 보기:

빈도 초과, 빈도 미만

파라미터

기본 파라미터

라벨	명칭	유형	설명
Input value raster	INPUT_VALUE_RASTER	[raster]	입력 값 레이어는 샘플 레이어들을 위한 기준 레이어 역할을 합니다.
Value raster band	INPUT_VALUE_RASTER_BAND	[raster band] 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] 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] 기본값: [Save to temporary file]	산출 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장

고급 파라미터

라벨	명칭	유형	설명
Output no data value 부가적	OUTPUT_NO_DATA_VALUE	[number] 기본값: -9999.0	산출 레이어에서 NODATA 용으로 사용할 값

산출물

라벨	명칭	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_COUNT	[number]	
Height in pixels	HEIGHT_IN_PIXELS	[number]	산출 래스터 레이어의 행 개수
Total pixel count	TOTAL_PIXEL_COUNT	[integer]	산출 래스터 레이어의 픽셀 개수
Mean frequency at valid cell locations	MEAN_FREQUENCY_AT_LOCATION	[number]	
Count of value occurrences	OCCURRENCE_COUNT	[number]	
Width in pixels	WIDTH_IN_PIXELS	[integer]	산출 래스터 레이어의 열 개수

파이썬 코드

알고리즘 ID: `native:equaltofrequency`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

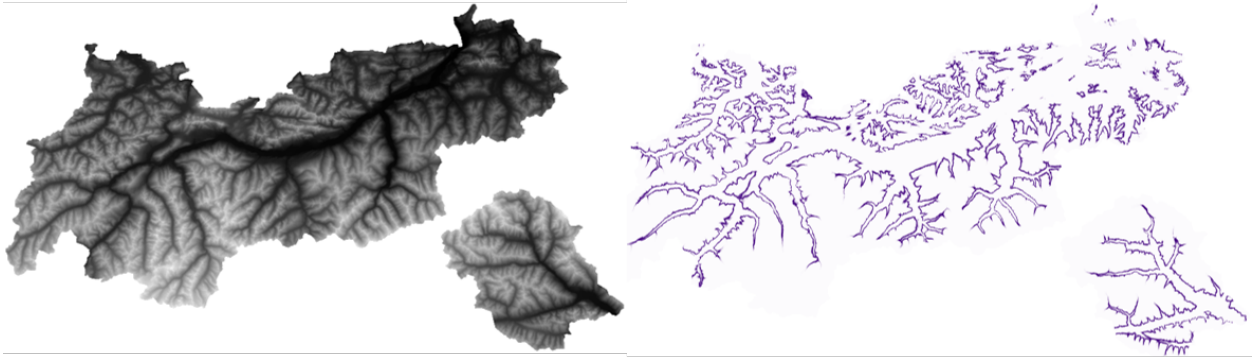
공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

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.

더 보기:

래스터 퍼지화 *(큰 값 멤버십)*, 래스터 퍼지화 *(선형 멤버십)*, 래스터 퍼지화 *(근접 멤버십)*, 래스터 퍼지화 *(거듭제곱 멤버십)*, 래스터 퍼지화 *(작은 값 멤버십)*



Input raster　　Fuzzified raster

그림 24.10: 래스터 퍼지화의 예. 입력 래스터 소스: 오스트리아 티롤 지역 - data.tirol.gv.at

파라미터

라벨	명칭	유형	설명
Input Raster	`INPUT`	[raster]	입력 래스터 레이어
Band Number	`BAND`	[raster band] 기본값: 래스터 레이어의 첫 번째 밴드	래스터가 다중 밴드인 경우, 퍼지화하려는 밴드를 선택하십시오.
Function midpoint	`FUZZYMIDPOINT`	[number] 기본값: 10	가우스 함수의 중점
Function spread	`FUZZYSPREAD`	[number] 기본값: 0.01	가우스 함수의 확산
Fuzzified raster	`OUTPUT`	[same as input]	산출 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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`	[extent]	산출 래스터 레이어의 공간 영역
Width in pixels	`WIDTH_IN_PIXELS`	[integer]	산출 래스터 레이어의 픽셀 단위 너비
Height in pixels	`HEIGHT_IN_PIXELS`	[integer]	산출 래스터 레이어의 픽셀 단위 높이
Total pixel count	`TOTAL_PIXEL_COUNT`	[integer]	산출 래스터 레이어의 픽셀 개수

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고

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

라벨	명칭	유형	설명
Input Raster	`INPUT`	[raster]	입력 래스터 레이어
Band Number	`BAND`	[raster band] 기본값: 래스터 레이어의 첫 번째 밴드	래스터가 다중 밴드인 경우, 퍼지화하려는 밴드를 선택하십시오.
Function midpoint	`FUZZYMIDPOINT`	[number] 기본값: 50	큰 값 함수의 중점
Function spread	`FUZZYSPREAD`	[number] 기본값: 5	큰 값 함수의 확산
Fuzzified raster	`OUTPUT`	[same as input]	산출 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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`	[extent]	산출 래스터 레이어의 공간 영역
Width in pixels	`WIDTH_IN_PIXELS`	[integer]	산출 래스터 레이어의 픽셀 단위 너비
Height in pixels	`HEIGHT_IN_PIXELS`	[integer]	산출 래스터 레이어의 픽셀 단위 높이
Total pixel count	`TOTAL_PIXEL_COUNT`	[integer]	산출 래스터 레이어의 픽셀 개수

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 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

라벨	명칭	유형	설명
Input Raster	`INPUT`	[raster]	입력 래스터 레이어
Band Number	`BAND`	[raster band] 기본값: 래스터 레이어의 첫 번째 밴드	래스터가 다중 밴드인 경우, 퍼지화하려는 밴드를 선택하십시오.
Low fuzzy membership bound	`FUZZYLOWBOUND`	[number] 기본값: 0	선형 함수의 하한
High fuzzy membership bound	`FUZZYHIGHBOUND`	[number] 기본값: 1	선형 함수의 상한
Fuzzified raster	`OUTPUT`	[same as input]	산출 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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`	[extent]	산출 래스터 레이어의 공간 영역
Width in pixels	`WIDTH_IN_PIXELS`	[integer]	산출 래스터 레이어의 픽셀 단위 너비
Height in pixels	`HEIGHT_IN_PIXELS`	[integer]	산출 래스터 레이어의 픽셀 단위 높이
Total pixel count	`TOTAL_PIXEL_COUNT`	[integer]	산출 래스터 레이어의 픽셀 개수

파이썬 코드

알고리즘 ID: `qgisfuzzifyrasterlinearmembership`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 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

라벨	명칭	유형	설명
Input Raster	INPUT	[raster]	입력 래스터 레이어
Band Number	BAND	[raster band] 기본값: 래스터 레이어의 첫 번째 밴드	래스터가 다중 밴드인 경우, 퍼지화하려는 밴드를 선택하십시오.
Function midpoint	FUZZYMIDPOINT	[number] 기본값: 50	근접 함수의 중점
Function spread	FUZZYSPREAD	[number] 기본값: 0.01	근접 함수의 확산
Fuzzified raster	OUTPUT	[same as input]	산출 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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	[extent]	산출 래스터 레이어의 공간 영역
Width in pixels	WIDTH_IN_PIXELS	[integer]	산출 래스터 레이어의 픽셀 단위 너비
Height in pixels	HEIGHT_IN_PIXELS	[integer]	산출 래스터 레이어의 픽셀 단위 높이
Total pixel count	TOTAL_PIXEL_COUNT	[integer]	산출 래스터 레이어의 픽셀 개수

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 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

라벨	명칭	유형	설명
Input Raster	`INPUT`	[raster]	입력 래스터 레이어
Band Number	`BAND`	[raster band] 기본값: 래스터 레이어의 첫 번째 밴드	래스터가 다중 밴드인 경우, 퍼지화하려는 밴드를 선택하십시오.
Low fuzzy membership bound	`FUZZYLOWBOUND`	[number] 기본값: 0	거듭제곱 함수의 하한
High fuzzy membership bound	`FUZZYHIGHBOUND`	[number] 기본값: 1	거듭제곱 함수의 상한
High fuzzy membership bound	`FUZZYEXPONENT`	[number] 기본값: 2	거듭제곱 함수의 지수
Fuzzified raster	`OUTPUT`	[same as input]	산출 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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`	[extent]	산출 래스터 레이어의 공간 영역
Width in pixels	`WIDTH_IN_PIXELS`	[integer]	산출 래스터 레이어의 픽셀 단위 너비
Height in pixels	`HEIGHT_IN_PIXELS`	[integer]	산출 래스터 레이어의 픽셀 단위 높이
Total pixel count	`TOTAL_PIXEL_COUNT`	[integer]	산출 래스터 레이어의 픽셀 개수

파이썬 코드

알고리즘 ID: `qgisfuzzifyrasterpowermembership`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 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

라벨	명칭	유형	설명
Input Raster	`INPUT`	[raster]	입력 래스터 레이어
Band Number	`BAND`	[raster band] 기본값: 래스터 레이어의 첫 번째 밴드	래스터가 다중 밴드인 경우, 퍼지화하려는 밴드를 선택하십시오.
Function midpoint	`FUZZYMIDPOINT`	[number] 기본값: 50	작은 값 함수의 중점
Function spread	`FUZZYSPREAD`	[number] 기본값: 5	작은 값 함수의 확산
Fuzzified raster	`OUTPUT`	[same as input]	산출 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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`	[extent]	산출 래스터 레이어의 공간 영역
Width in pixels	`WIDTH_IN_PIXELS`	[integer]	산출 래스터 레이어의 픽셀 단위 너비
Height in pixels	`HEIGHT_IN_PIXELS`	[integer]	산출 래스터 레이어의 픽셀 단위 높이
Total pixel count	`TOTAL_PIXEL_COUNT`	[integer]	산출 래스터 레이어의 픽셀 개수

파이썬 코드

알고리즘 ID: `qgisfuzzifyrastersmallmembership`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 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.

더 보기:

빈도와 동일, 빈도 미만

파라미터

기본 파라미터

라벨	명칭	유형	설명
Input value raster	`INPUT_VALUE_RASTER`	[raster]	입력 값 레이어는 샘플 레이어들을 위한 기준 레이어 역할을 합니다.
Value raster band	`INPUT_VALUE_RASTER`	[raster band] 기본값: 래스터 레이어의 첫 번째 밴드	샘플로 사용하려는 밴드를 선택하십시오.

다음 페이지에 계속

표 24.25 – 이전 페이지에서 계속

라벨	명칭	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] 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] 기본값: [Save to temporary file]	산출 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장

고급 파라미터

라벨	명칭	유형	설명
Output no data value 부가적	OUTPUT_NO_DATA_VALUE	[number] 기본값: -9999.0	산출 레이어에서 NODATA 용으로 사용할 값

산출물

라벨	명칭	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_COUNT	[number]	
Height in pixels	HEIGHT_IN_PIXELS	[number]	산출 래스터 레이어의 행 개수
Total pixel count	TOTAL_PIXEL_COUNT	[integer]	산출 래스터 레이어의 픽셀 개수
Mean frequency at valid cell locations	MEAN_FREQUENCY_PERMUTATION	[number]	
Count of value occurrences	OCCURRENCE_COUNT	[number]	
Width in pixels	WIDTH_IN_PIXELS	[integer]	산출 래스터 레이어의 열 개수

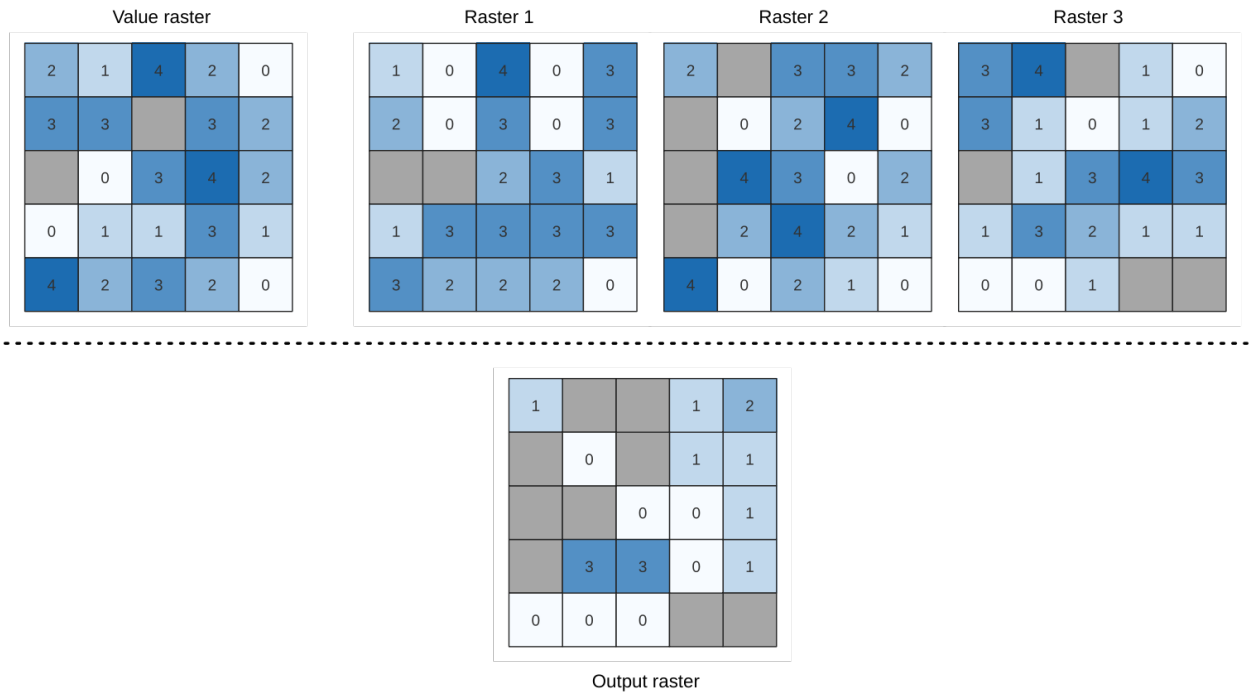


그림 24.11: 산출 래스터에 있는 각 셀의 값은 래스터 목록에서 대응하는 셀의 값이 값 래스터의 해당 셀의 값을 초과하는 횟수를 나타냅니다. NoData (회색) 셀도 고려합니다.

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 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

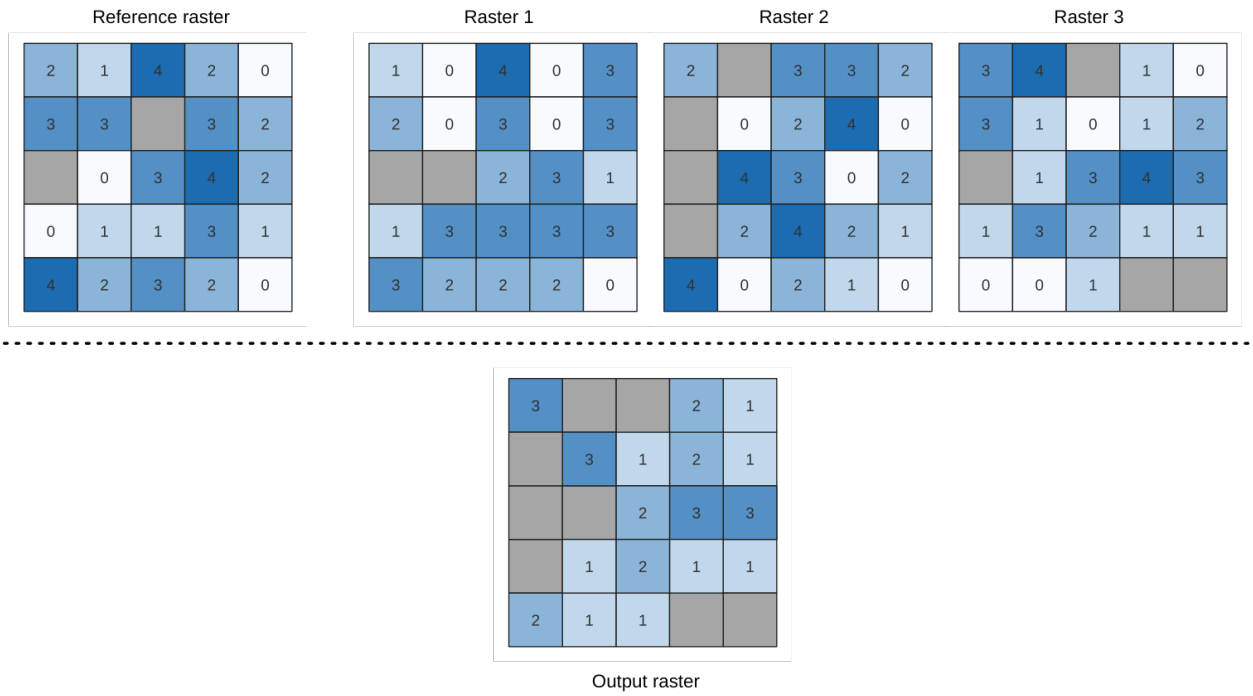
래스터 스택에서 가장 높은 위치

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	명칭	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] 기본값: False	이 옵션을 체크하지 않을 경우, 데이터 레이어 스택에 있는 모든 NODATA 셀이 산출 래스터에서 NODATA 셀로 산출될 것입니다.
Output layer	`OUTPUT`	[raster] 기본값: `[Save to temporary file]`	결과를 담을 산출 래스터를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장

고급 파라미터

라벨	명칭	유형	설명
Output no data value	`OUTPUT_NODATA_VALUE`	[number] 기본값: -9999.0	산출 레이어에서 NODATA 용으로 사용할 값

산출물

라벨	명칭	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

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.

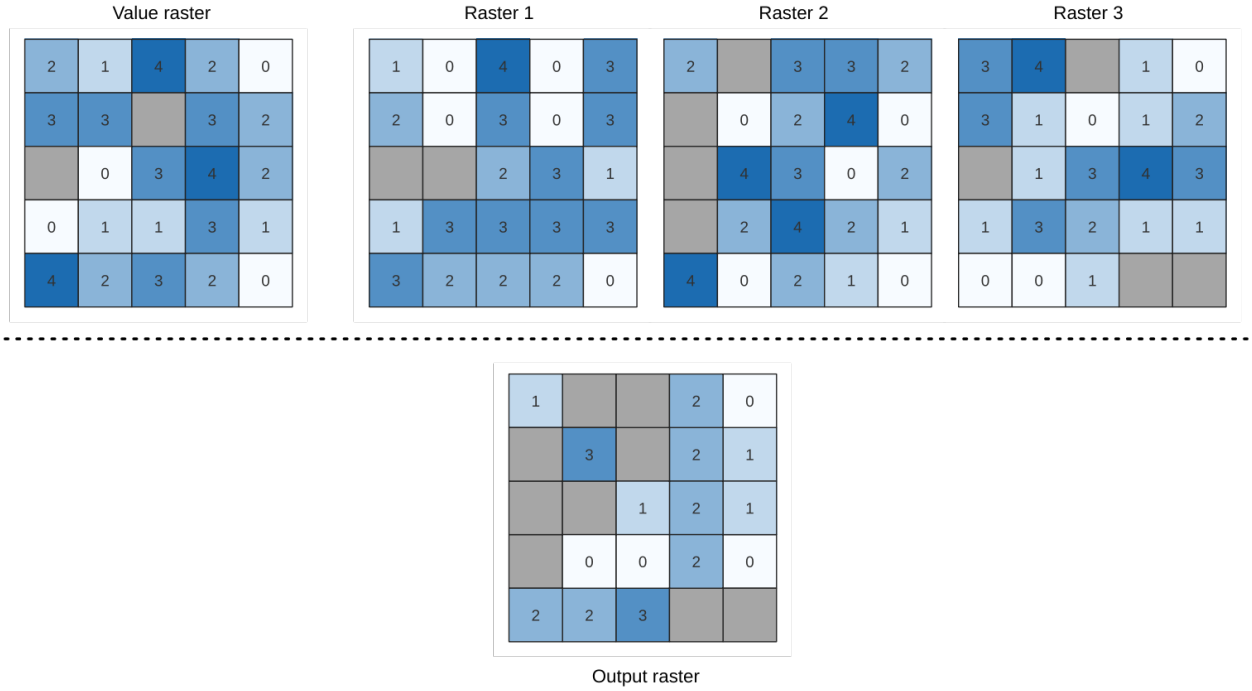


그림 24.12: 산출 래스터에 있는 각 셀의 값은 래스터 목록에서 대응하는 셀의 값이 값 래스터의 해당 셀의 값에 미만하는 횟수를 나타냅니다. `NoData` (회색) 셀도 고려합니다.

더 보기:

빈도와 동일, 빈도 초과

파라미터

기본 파라미터

라벨	명칭	유형	설명
Input value raster	`INPUT_VALUE_RASTER`	[raster]	입력 값 레이어는 샘플 레이어들을 위한 기준 레이어 역할을 합니다.
Value raster band	`INPUT_VALUE_RASTER_BAND`	[raster band] 기본값: 래스터 레이어의 첫 번째 밴드	샘플로 사용하려는 밴드를 선택하십시오.

다음 페이지에 계속

표 24.29 – 이전 페이지에서 계속

라벨	명칭	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] 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] 기본값: [Save to temporary file]	산출 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장

고급 파라미터

라벨	명칭	유형	설명
Output no data value 부가적	OUTPUT_NO_DATA_VALUE	[number] 기본값: -9999.0	산출 레이어에서 NODATA 용으로 사용할 값

산출물

라벨	명칭	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_COUNT	[number]	
Height in pixels	HEIGHT_IN_PIXELS	[number]	산출 래스터 레이어의 행 개수
Total pixel count	TOTAL_PIXEL_COUNT	[integer]	산출 래스터 레이어의 픽셀 개수
Mean frequency at valid cell locations	MEAN_FREQUENCY_PERMUTATION	[number]	
Count of value occurrences	OCCURRENCE_COUNT	[number]	
Width in pixels	WIDTH_IN_PIXELS	[integer]	산출 래스터 레이어의 열 개수

파이썬 코드

알고리즘 ID: `native:lessthanfrequency`

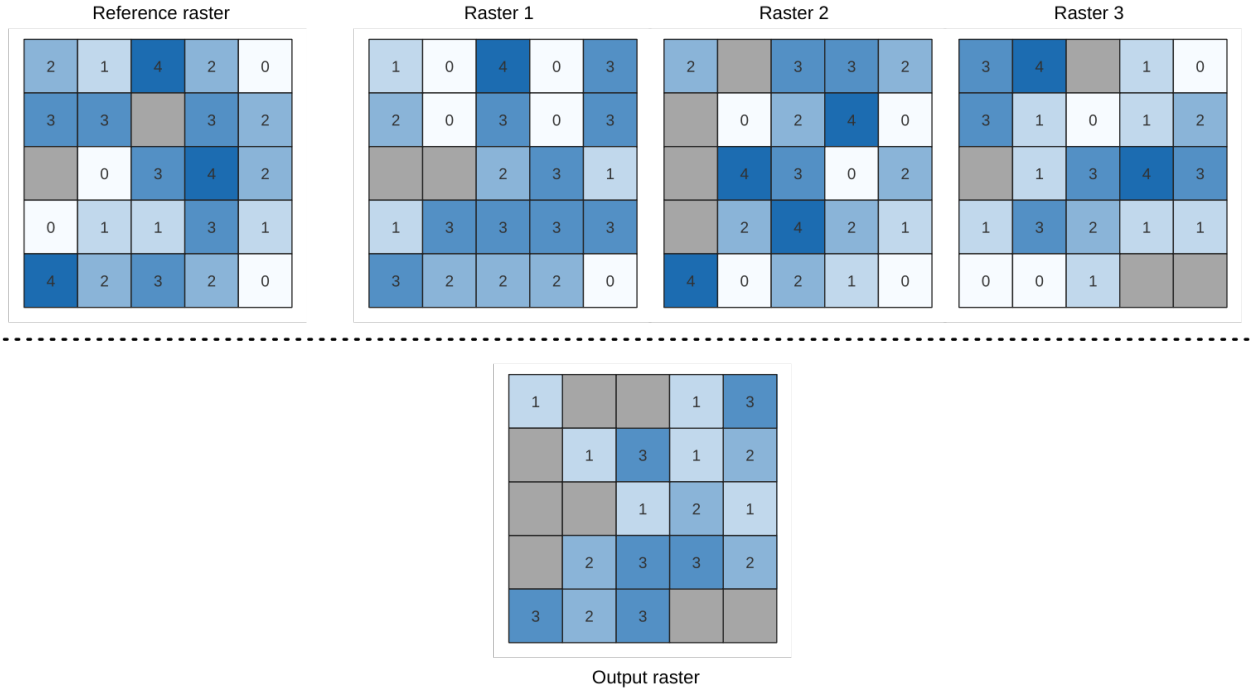
```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 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	명칭	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] 기본값: False	이 옵션을 체크하지 않을 경우, 데이터 레이어 스택에 있는 모든 NODATA 셀이 산출 래스터에서 NODATA 셀로 산출될 것입니다.
Output layer	OUTPUT	[raster] 기본값: [Save to temporary file]	결과를 담을 산출 래스터를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장

고급 파라미터

라벨	명칭	유형	설명
Output no data value	OUTPUT_NODATA_VALUE	[number] 기본값: -9999.0	산출 레이어에서 NODATA 용으로 사용할 값

산출물

라벨	명칭	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	[integer]	산출 래스터 레이어의 열 개수
Height in pixels	HEIGHT_IN_PIXELS	[integer]	산출 래스터 레이어의 행 개수
Total pixel count	TOTAL_PIXEL_COUNT	[integer]	산출 래스터 레이어의 픽셀 개수

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 **명칭** 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

라벨	명칭	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] 기본값: False	작업 수행 시 입력 파일에 있는 NODATA 값을 0 으로 취급
Output no data value	NO_DATA	[number] 기본값: -9999.0	산출 레이어에서 NODATA 용으로 사용할 값
Output data type	DATA_TYPE	[enumeration] 기본값: 5	산출 래스터 데이터 유형입니다. 다음 가운데 하나를 선택할 수 있습니다: • 0 —바이트 (Byte) • 1 —부호 있는 16 비트 정수형 (Int16) • 2 —부호 없는 16 비트 정수형 (UInt16) • 3 —부호 없는 32 비트 정수형 (UInt32) • 4 —부호 있는 32 비트 정수형 (Int32) • 5 —부동소수점 32 비트 실수형 (Float32) • 6 —부동소수점 64 비트 실수형 (Float64) • 7 —복잡 16 비트 정수형 (CInt16) • 8 —복잡 32 비트 정수형 (CInt32) • 9 —복잡 부동소수점 32 비트 실수형 (CFloat32) • 10 —복잡 부동소수점 64 비트 실수형 (CFloat64)
Output layer	OUTPUT	[raster]	산출 래스터 레이어

산출물

라벨	명칭	유형	설명
Extent	`EXTENT`	[extent]	산출 래스터 레이어의 범위
CRS authority identifier	`CRS_AUTHID`	[crs]	산출 래스터 레이어의 좌표계
Width in pixels	`WIDTH_IN_PIXELS`	[integer]	산출 래스터 레이어의 픽셀 단위 너비
Height in pixels	`HEIGHT_IN_PIXELS`	[integer]	산출 래스터 레이어의 픽셀 단위 높이
Total pixel count	`TOTAL_PIXEL_COUNT`	[integer]	산출 래스터 레이어의 픽셀 개수
NODATA pixel count	`NODATA_PIXEL_COUNT`	[integer]	산출 래스터 레이어의 NODATA 픽셀 개수
True pixel count	`TRUE_PIXEL_COUNT`	[integer]	산출 래스터 레이어에 있는 (값이 1 인) 참 픽셀의 개수
False pixel count	`FALSE_PIXEL_COUNT`	[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: `qgis: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 **명칭** 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

라벨	명칭	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] 기본값: False	작업 수행 시 입력 파일에 있는 NODATA 값을 0 으로 취급
Output no data value	NO_DATA	[number] 기본값: -9999.0	산출 레이어에서 NODATA 용으로 사용할 값
Output data type	DATA_TYPE	[enumeration] 기본값: 5	산출 래스터 데이터 유형입니다. 다음 가운데 하나를 선택할 수 있습니다: • 0 —바이트 (Byte) • 1 —부호 있는 16 비트 정수형 (Int16) • 2 —부호 없는 16 비트 정수형 (UInt16) • 3 —부호 없는 32 비트 정수형 (UInt32) • 4 —부호 있는 32 비트 정수형 (Int32) • 5 —부동소수점 32 비트 실수형 (Float32) • 6 —부동소수점 64 비트 실수형 (Float64) • 7 —복잡 16 비트 정수형 (CInt16) • 8 —복잡 32 비트 정수형 (CInt32) • 9 —복잡 부동소수점 32 비트 실수형 (CFloat32) • 10 —복잡 부동소수점 64 비트 실수형 (CFloat64)
Output layer	OUTPUT	[raster]	산출 래스터 레이어

산출물

라벨	명칭	유형	설명
Extent	EXTENT	[extent]	산출 래스터 레이어의 범위
CRS authority identifier	CRS_AUTHID	[crs]	산출 래스터 레이어의 좌표계
Width in pixels	WIDTH_IN_PIXELS	[integer]	산출 래스터 레이어의 픽셀 단위 너비
Height in pixels	HEIGHT_IN_PIXELS	[integer]	산출 래스터 레이어의 픽셀 단위 높이
Total pixel count	TOTAL_PIXEL_COUNT	[integer]	산출 래스터 레이어의 픽셀 개수
NODATA pixel count	NODATA_PIXEL_COUNT	[integer]	산출 래스터 레이어의 NODATA 픽셀 개수
True pixel count	TRUE_PIXEL_COUNT	[integer]	산출 래스터 레이어에 있는 (값이 1 인) 참 픽셀의 개수
False pixel count	FALSE_PIXEL_COUNT	[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: `qgis: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 **명칭** 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.

참고: 배치 프로세스 인터페이스 또는*QGIS* 파이썬 콘솔 에서 계산기를 사용하는 경우 사용할 파일을 지정해야 합니다. 대응하는 레이어는 (전체 경로가 아닌) 기본명을 사용해서 참조시켜야 합니다. 예를 들어, `path/to/my/rasterfile.tif` 위치에 있는 레이어를 사용하는 경우 해당 레이어의 첫 번째 밴드는 `rasterfile.tif@1` 로 참조될 것입니다.

더 보기:

래스터 계산기

파라미터

라벨	명칭	유형	설명
Layers	GUI 전용		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 명칭 및 밴드 번호로 참조됩니다. 예를 들어 `DEM` 이라는 레이어의 첫 번째 밴드는 `DEM@1` 로 참조될 것입니다.
Operators	GUI 전용		표현식란을 채울 수 있는 몇몇 계산기 버튼을 담고 있습니다.
Expression	`EXPRESSION`	[string]	산출 래스터 레이어를 계산하기 위해 사용할 수 있는 표현식입니다. 제공된 연산자 버튼을 사용해서 표현식란에 표현식을 직접 입력할 수 있습니다.

다음 페이지에 계속

표 24.37 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Predefined expressions	GUI 전용		사전 정의된 NDVI 표현식을 사용할 수도 있고, 또는 계산을 위한 새 표현식을 정의할 수도 있습니다. *Add…*버튼을 클릭하면 사전 정의 표현식을 불러올 수 있습니다. (그리고 사용자가 파라미터를 설정할 수 있지요.) *Save…*버튼을 클릭하면 새 표현식을 정의할 수 있습니다.
Reference layer(s) (used for automated extent, cellsize, and CRS) 부가적	LAYERS	[raster] [list]	범위, 셀 크기 및 좌표계를 가져올 레이어 (들) 입니다. 이 상자에서 레이어를 선택하면 사용자가 직접 다른 모든 파라미터를 지정해야 할 필요가 사라집니다. 래스터 레이어는 layer_name@band_number 처럼 명칭 및 밴드 번호로 참조됩니다. 예를 들어 DEM 이라는 레이어의 첫 번째 밴드는 DEM@1 로 참조될 것입니다.
Cell size (use 0 or empty to set it automatically) 부가적	CELLSIZE	[number]	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 extent (xmin, xmax, ymin, ymax) 부가적	EXTENT	[extent]	산출 래스터 레이어의 범위입니다. 범위를 지정하지 않은 경우, 선택한 참조 레이어 (들) 을 모두 커버하는 최소 범위를 사용할 것입니다.
Output CRS 부가적	CRS	[crs]	산출 래스터 레이어의 좌표계입니다. 좌표계를 지정하지 않은 경우, 첫 번째 참조 레이어의 좌표계를 사용할 것입니다.
Output	OUTPUT	[raster] 기본값: [Save to temporary file]	산출 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Output	OUTPUT	[raster]	계산된 값을 가진 산출 래스터 파일

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 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	명칭	유형	설명
Input layer	`INPUT`	[raster]	입력 래스터 레이어
Band number	`BAND`	[raster band] 기본값: 입력 레이어의 첫 번째 밴드	래스터가 다중 밴드인 경우, 통계를 얻고자 하는 밴드를 선택하십시오.
Output	`OUTPUT_HTML_FILE`	[html] 기본값: `[Save to temporary file]`	산출 레이어를 지정합니다: • 산출물 건너뛰기 • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Maximum value	`MAX`	[number]	
Mean value	`MEAN`	[number]	
Minimum value	`MIN`	[number]	
Output	`OUTPUT_HTML_FILE`	[html]	산출물 파일은 다음 정보를 담고 있습니다: • 파일 분석: 래스터 레이어의 경로 • 최소값: 래스터의 최소값 • 최대값: 래스터의 최대값 • 범위: 최대값과 최소값의 차 • 합계: 값들의 총합 • 평균값: 값들의 평균 • 표준 편차: 값들의 표준 편차 • 제곱 합계: 전체 평균과 각 관측값의 차를 제곱한 값의 합계

다음 페이지에 계속

표 24.38 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Range	RANGE	[number]	
Standard deviation	STD_DEV	[number]	
Sum	SUM	[number]	
Sum of the squares	SUM_OF_SQUARES	[number]	

파이썬 코드

알고리즘 ID: qgis: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 **명칭** 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 레이어에 있는 각 유일값의 개수와 면적을 반환합니다.

파라미터

라벨	명칭	유형	설명
Input layer	INPUT	[raster]	입력 래스터 레이어
Band number	BAND	[raster band] 기본값: 입력 레이어의 첫 번째 밴드	래스터가 다중 밴드인 경우, 통계를 얻고자 하는 밴드를 선택하십시오.
Unique values report	OUTPUT_HTML_FILE	[file] 기본값: [Save to temporary file]	산출 레이어를 지정합니다: • 산출물 건너뛰기 • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.
Unique values table	OUTPUT_TABLE	[table] 기본값: [Skip output]	유일값을 위한 테이블을 지정합니다: • 산출물 건너뛰기 • 임시 레이어 생성 • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
CRS authority identifier	CRS_AUTHID	[crs]	
Extent	EXTENT	[extent]	
Height in pixels	HEIGHT_IN_PIXELS	[number]	
NODATA pixel count	NODATA_PIXEL_COUNT	[number]	
Total pixel count	TOTAL_PIXEL_COUNT	[number]	
Unique values report	OUTPUT_HTML_FILE	[html]	산출 HTML 파일은 다음 정보를 담고 있습니다: • 파일 분석: 래스터 레이어의 경로 • 범위: 범위의 xmin, ymin, xmax, ymax 좌표 • 투영체: 레이어의 투영체 • 픽셀 단위 너비: 열의 개수와 픽셀 너비 크기 • 픽셀 단위 높이: 행의 개수와 픽셀 너비 크기 • 총 픽셀 개수: 모든 픽셀의 개수 • NODATA 픽셀 개수: NODATA 값을 가진 픽셀의 개수
Unique values table	OUTPUT_TABLE	[table]	다음 열 3 개를 가진 테이블: • *value*: 픽셀 값 • *count*: 해당 값을 가진 픽셀의 개수 • m^2: 해당 값을 가진 픽셀의 제곱미터 단위 총 면적
Width in pixels	WIDTH_IN_PIXELS	[number]	

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 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

라벨	명칭	유형	설명
Input Layer	INPUT	[raster]	입력 래스터 레이어
Band number	BAND	[raster band] 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.
Zones band number	ZONES_BAND	[raster band] 기본값: 래스터 레이어의 첫 번째 밴드	래스터가 다중 밴드인 경우, 구역을 정의하는 밴드를 선택하십시오.
Reference layer 부가적	REF_LAYER	[enumeration] 기본값: 0	산출 레이어에 구역을 결정할 때 참조물로 사용될 중심 (centroid) 을 계산하기 위해 사용되는 래스터 레이어입니다. 다음 가운데 하나를 선택할 수 있습니다: • 0 —입력 레이어 • 1 —구역 레이어
Statistics	OUTPUT_TABLE	[table]	계산된 통계를 가진 테이블

산출물

라벨	명칭	유형	설명
CRS authority identifier	CRS_AUTHID	[crs]	
Extent	EXTENT	[extent]	
Height in pixels	HEIGHT_IN_PIXELS	[number]	
NODATA pixel count	NODATA_PIXEL_COUNT	[number]	
Statistics	OUTPUT_TABLE	[table]	산출 레이어는 **각 구역별로** 다음 정보를 담고 있습니다: • 면적: 구역의 제곱 래스터 단위 면적 • 합계: 구역에 있는 픽셀 값의 총합 • 개수: 구역에 있는 픽셀들의 개수 • 최소값: 구역에 있는 최소 픽셀 값 • 최대값: 구역에 있는 최대 픽셀 값 • 평균: 구역에 있는 픽셀 값들의 평균

다음 페이지에 계속

표 24.42 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Total pixel count	TOTAL_PIXEL_COUNT	[number]	
Width in pixels	WIDTH_IN_PIXELS	[number]	

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 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	명칭	유형	설명
INPUT layer	INPUT	[raster]	표면을 표현하는 입력 래스터
Band number	BAND	[raster band] 기본값: 래스터 레이어의 첫 번째 밴드	래스터가 다중 밴드인 경우, 표면을 정의할 밴드를 선택하십시오.
Base level	LEVEL	[number] 기본값: 0.0	기준 또는 참조 값을 정의합니다. 부피를 Method 파라미터에 따라 (아래 참조) 계산하는 데 이 기준을 사용합니다.

다음 페이지에 계속

표 24.43 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Method	METHOD	[enumeration] 기본값: 0	래스터 픽셀 값과 Base level 의 차로 지정하는 부피 계산용 방법을 정의합니다. 다음 옵션 가운데 하나를 선택할 수 있습니다: • 0 —기준면 위만 셉니다: 기준면 위에 있는 픽셀만 부피에 추가할 것입니다. • 1 —기준면 아래만 셉니다: 기준면 아래 있는 픽셀만 부피에 추가할 것입니다. • 2 —기준면 아래의 부피를 뺍니다: 기준면 위에 있는 픽셀을 부피에 추가하고, 기준면 아래 있는 픽셀을 부피에서 뺄 것입니다. • 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] 기본값: [Save to temporary file]	산출 HTML 파일을 지정합니다. 다음 가운데 하나를 선택할 수 있습니다: • 산출물 건너뛰기 • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.
Surface volume table	OUTPUT_TABLE	[table] 기본값: [Skip output]	산출 테이블을 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 산출물 건너뛰기 • 임시 레이어 생성 (TEMPORARY_OUTPUT) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Volume	VOLUME	[number]	계산된 부피
Area	AREA	[number]	제곱 맵 단위의 면적
Pixel_count	PIXEL_COUNT	[number]	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	[table]	(부피, 면적 및 픽셀 개수를 담고 있는) 산출 테이블

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 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

라벨	명칭	유형	설명
Raster layer	INPUT_RASTER	[raster]	재범주화할 래스터 레이어
Band number	RASTER_BAND	[raster band] 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]	범주용 범위의 최소값을 가진 필드
Maximum class value field	MAX_FIELD	[tablefield: numeric]	범주용 범위의 최대값을 가진 필드
Output value field	VALUE_FIELD	[tablefield: numeric]	(대응하는 최소 및 최대값 사이의) 범주에 들어오는 픽셀에 할당될 값을 가진 필드
Output no data value	NO_DATA	[number] 기본값: -9999.0	NODATA 값에 적용할 값
Range boundaries	RANGE_BOUNDARIES	[enumeration] 기본값: 0	범주화를 위한 비교 규칙을 정의합니다. 다음 옵션 가운데 선택할 수 있습니다: • 0 — 최소값 < 값 <= 최대값 • 1 — 최소값 <= 값 < 최대값 • 2 — 최소값 <= 값 <= 최대값 • 3 — 최소값 < 값 < 최대값
Use no data when no range matches value	NO-DATA_FOR_MISSING	[boolean] 기본값: False	범주에 속하지 않는 값은 NODATA 값으로 산출될 것입니다. False 로 설정한 경우, 원본 값을 유지합니다.

다음 페이지에 계속

표 24.44 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Output data type	DATA_TYPE	[enumeration] 기본값: 5	산출 래스터 파일의 데이터 유형을 정의합니다. 다음 가운데 하나로 지정할 수 있습니다: • 0 —바이트 (Byte) • 1 —부호 있는 16 비트 정수형 (Int16) • 2 —부호 없는 16 비트 정수형 (UInt16) • 3 —부호 없는 32 비트 정수형 (UInt32) • 4 —부호 있는 32 비트 정수형 (Int32) • 5 —부동소수점 32 비트 실수형 (Float32) • 6 —부동소수점 64 비트 실수형 (Float64) • 7 —복잡 16 비트 정수형 (CInt16) • 8 —복잡 32 비트 정수형 (CInt32) • 9 —복잡 부동소수점 32 비트 실수형 (CFloat32) • 10 —복잡 부동소수점 64 비트 실수형 (CFloat64)
Reclassified raster	OUTPUT	[raster]	산출 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Reclassified raster	OUTPUT	[raster]	재범주화된 밴드 값을 가진 산출 래스터 레이어

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 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

라벨	명칭	유형	설명
Raster layer	INPUT_RASTER	[raster]	재범주화할 래스터 레이어
Band number	RASTER_BAND	[raster band] 기본값: 1	값을 재계산하고자 하는 래스터 밴드
Reclassification table	TABLE	[table]	각 범주의 범위 (Minimum 및 Maximum)를 설정할 값과 범주 안에 들어오는 밴드 값에 할당할 새 Value 로 채워질 열 3 개로 이루어진 테이블
Output no data value	NO_DATA	[number] 기본값: -9999.0	NODATA 값에 적용할 값
Range boundaries	RANGE_BOUNDARIES	[enumeration] 기본값: 0	범주화를 위한 비교 규칙을 정의합니다. 다음 옵션 가운데 선택할 수 있습니다: • 0 —최소값 < 값 <= 최대값 • 1 —최소값 <= 값 < 최대값 • 2 —최소값 <= 값 <= 최대값 • 3 —최소값 < 값 < 최대값
Use no data when no range matches value	NO-DATA_FOR_MISSING	[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. 옵션: • 0 —바이트 (Byte) • 1 —부호 있는 16 비트 정수형 (Int16) • 2 —부호 없는 16 비트 정수형 (UInt16) • 3 —부호 없는 32 비트 정수형 (UInt32) • 4 —부호 있는 32 비트 정수형 (Int32) • 5 —부동소수점 32 비트 실수형 (Float32) • 6 —부동소수점 64 비트 실수형 (Float64) • 7 —복잡 16 비트 정수형 (CInt16) • 8 —복잡 32 비트 정수형 (CInt32) • 9 —복잡 부동소수점 32 비트 실수형 (CFloat32) • 10 —복잡 부동소수점 64 비트 실수형 (CFloat64)

다음 페이지에 계속

표 24.45 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Reclassified raster	OUTPUT	[raster] 기본값: '[Save to temporary file]'	산출 래스터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Reclassified raster	OUTPUT	[raster] 기본값: '[Save to temporary file]'	산출 래스터 레이어

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 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.

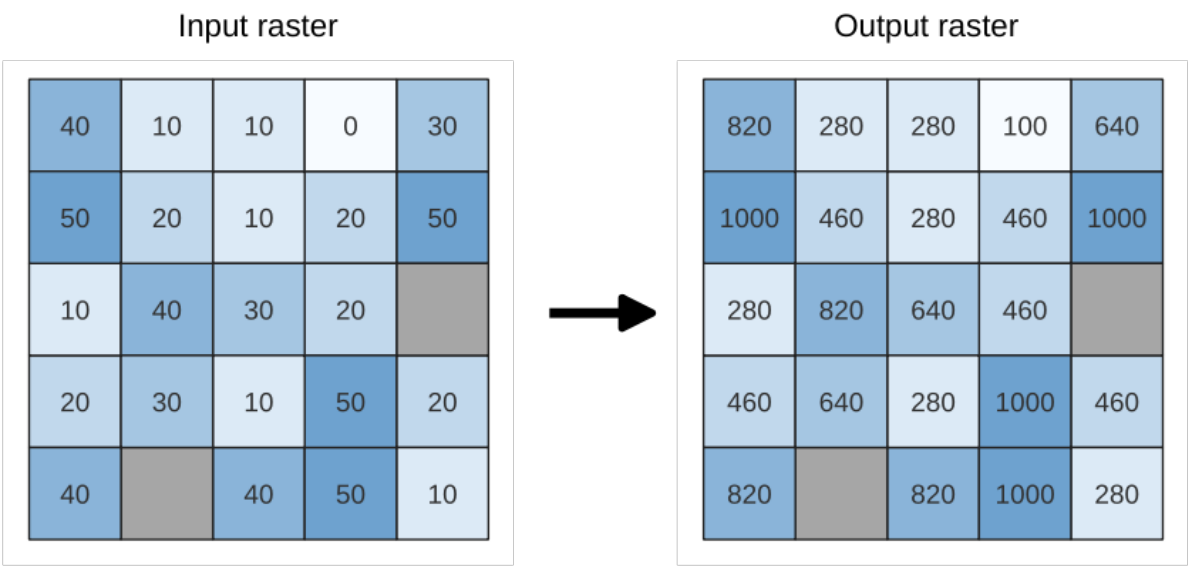


그림 24.13: 래스터 레이어의 값을 [0 - 50] 에서 [100 - 1000] 으로 크기 조정하기

파라미터

라벨	명칭	유형	설명
Input Raster	`INPUT`	[raster]	크기 조정에 사용할 래스터 레이어
Band number	`Band`	[raster band] 기본값: 입력 레이어의 첫 번째 밴드	래스터가 다중 밴드인 경우, 밴드를 선택하십시오.
New minimum value	`MINIMUM`	[number] 기본값: 0.0	크기 조정된 레이어에 사용할 최소 픽셀값
New maximum value	`MAXIMUM`	[number] 기본값: 255.0	크기 조정된 레이어에 사용할 최대 픽셀값
New NODATA value 부가적	`NODATA`	[number] 기본값: 설정하지 않음	NODATA 픽셀에 할당할 값. 이 값을 설정하지 않으면, 원본 NODATA 값을 유지합니다.
Rescaled	`OUTPUT`	[raster] 기본값: `[Save to temporary file]`	산출 래스터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장

산출물

라벨	명칭	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 **명칭** 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.

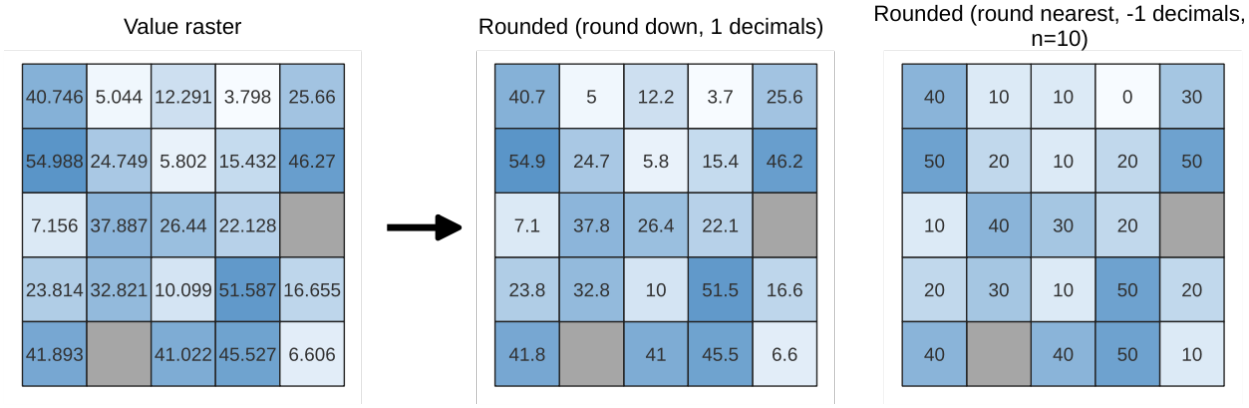


그림 24.14: 래스터의 값을 반올림하기

파라미터

기본 파라미터

라벨	명칭	유형	설명
Input raster	INPUT	[raster]	공간 처리할 래스터
Band number	BAND	[number] 기본값: 1	래스터의 밴드
Rounding direction	ROUNDING_DIRECTION	[list] 기본값: 1	대상의 값을 반올림할 방법을 선택하는 방식입니다. 다음 옵션 가운데 하나를 선택할 수 있습니다: 0 - 올림 1 - 반올림 2 - 내림
Number of decimals places	DECIMAL_PLACES	[number] 기본값: 2	반올림할 소수점 이하 자릿수. 셀 값을 밑 n 의 배수로 반올림하려면 음의 값을 사용하십시오.
Output raster	OUTPUT	[raster] 기본값: [Save to temporary file]	산출 파일을 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장

고급 파라미터

라벨	명칭	유형	설명
Base n for rounding to multiples of n	BASE_N	[number] 기본값: 10	DECIMAL_PLACES 파라미터가 음수일 경우, 래스터 값을 밑 n 값의 배수로 반올림합니다.

산출물

라벨	명칭	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 **명칭** 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	명칭	유형	설명
Input Point Layer	`INPUT`	[vector: point]	샘플링에 사용할 포인트 벡터 레이어
Raster Layer to sample	`RASTERCOPY`	[raster]	지정한 포인트 위치에서 샘플링할 래스터 레이어
Output column prefix	`COLUMN_PREFIX`	[string] 기본값: 'rvalue'	추가될 열의 명칭 앞에 붙일 접두어
Sampled Points 부가적	`OUTPUT`	[vector: point] 기본값: `[Create temporary layer]`	샘플링된 값을 담고 있는 산출 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • 데이터베이스 테이블에 저장… 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Sampled Points 부가적	`OUTPUT`	[vector: point]	샘플링된 값을 담고 있는 산출 레이어

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 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.

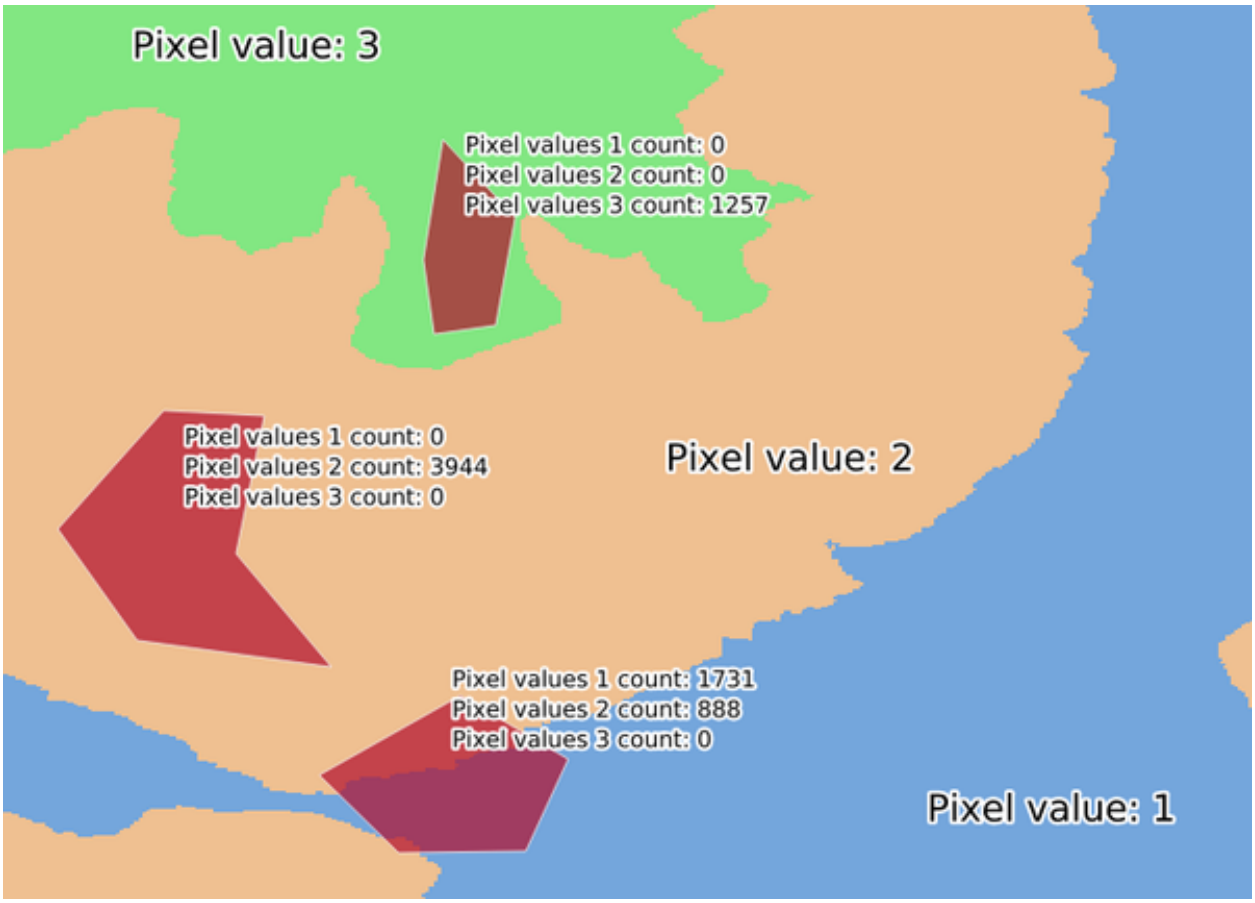


그림 24.15: 래스터 레이어 히스토그램 예시

파라미터

라벨	명칭	유형	설명
Raster layer	INPUT_RASTER	[raster]	입력 래스터 레이어
Band number	RASTER_BAND	[raster band] 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 부가적	[string] 기본값: 'HISTO_'	산출 열의 명칭 앞에 붙일 접두어
Output zones	OUTPUT	[vector: polygon] 기본값: [Create temporary layer]	산출 벡터 폴리곤 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (TEMPORARY_OUTPUT) • 파일로 저장 • Geopackage 로 저장 • 데이터베이스 테이블에 저장… 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Output zones 부가적	OUTPUT	[vector: polygon] 기본값: [Create temporary layer]	산출 벡터 폴리곤 레이어

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 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.

QGIS 3.16 이전 버전에서는 알고리듬이 레이어에 새 통계 필드를 추가하며 레이어를 제자리 (in-place) 편집했습니다. QGIS 3.16 버전부터, 이 통계를 담은 새 레이어를 산출합니다.

파라미터

라벨	명칭	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] 기본값: 입력 레이어의 첫 번째 밴드	래스터가 다중 밴드인 경우, 통계를 얻고자 하는 밴드를 선택하십시오.
Output column prefix	`COLUMN_PREFIX`	[string] 기본값: '_'	산출 열의 명칭 앞에 붙일 접두어
Statistics to calculate	`STATISTICS`	[enumeration] [list] 기본값: [0,1,2]	산출물을 위한 통계 연산자의 목록. 다음 옵션 가운데 선택할 수 있습니다: • 0 —개수 (Count) • 1 —합계 (Sum) • 2 —평균 (Mean) • 3 —중간값 (Median) • 4 —표준 편차 (St. dev.) • 5 —최소값 (Minimum) • 6 —최대값 (Maximum) • 7 —범위 (Range) • 8 —희귀값 (Minority) • 9 —최빈값 (Majority) • 10 —다양도 (Variety) • 11 —변동 (Variance)
Zonal Statistics `NEW in 3.16`	`OUTPUT`	[vector: polygon] 기본값: `[Create temporary layer]`	산출 벡터 폴리곤 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장… • GeoPackage 로 저장… • 데이터베이스 테이블에 저장… • 레이어에 추가… 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Zonal Statistics NEW in 3.16	OUTPUT	[vector: polygon]	통계를 추가한 입력 구역 벡터 레이어

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

24.1.10 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 범위 (xmin, xmax, ymin, ymax) 를 지정합니다. 다음 가운데 하나를 선택할 수 있습니다: • 캔버스 범위 사용 • 캔버스에서 범위 선택 • 레이어 범위 사용… 타일 크기의 배수가 되도록 내부적으로 확장될 것입니다.
Target CRS	`TARGET_CRS`	[crs] 기본값: Project CRS	산출 래스터 레이어 용 좌표계
Pixel size	`PIXEL_SIZE`	[number] 기본값: 0.1	맵 단위의 픽셀 크기 (X=Y) 입니다. 최소값은 0.01 입니다.
Constant value	`NUMBER`	[number] Default: 1	산출 래스터 레이어 용 상수 (constant) 픽셀 값입니다.
Constant	`OUTPUT`	[raster] 기본값: `[Save to temporary file]`	산출 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장…

고급 파라미터

라벨	이름	유형	설명
Output raster data type	`OUTPUT_TYPE` 기본값: 5	[enumeration]	산출 래스터 파일의 데이터 유형을 정의합니다. 다음 가운데 하나로 지정할 수 있습니다: • 0 —바이트 (Byte) • 1 —부호 있는 16 비트 정수형 (Integer16) • 2 —부호 없는 16 비트 정수형 (Unsigned Integer16) • 3 —부호 있는 32 비트 정수형 (Integer32) • 4 —부호 없는 32 비트 정수형 (Unsigned Integer32) • 5 —부동소수점 32 비트 실수형 (Float32) • 6 —부동소수점 64 비트 실수형 (Float64)

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 **명칭** 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 범위 (xmin, xmax, ymin, ymax) 를 지정합니다. 다음 가운데 하나를 선택할 수 있습니다: • 캔버스 범위 사용 • 캔버스에서 범위 선택 • 레이어 범위 사용… 타일 크기의 배수가 되도록 내부적으로 확장될 것입니다.
Target CRS	`TARGET_CRS`	[crs] 기본값: Project CRS	산출 래스터 레이어 용 좌표계
Pixel size	`PIXEL_SIZE`	[number] 기본값: 0.1	맵 단위의 픽셀 크기 (X=Y) 입니다. 최소값은 0.01 입니다.
Output raster	`OUTPUT`	[raster] 기본값: `[Save to temporary file]`	산출 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장…

고급 파라미터

라벨	이름	유형	설명
Output raster data type	`OUTPUT_TYPE` 기본값: 0	[enumeration]	산출 래스터 파일의 데이터 유형을 정의합니다. 다음 가운데 하나로 지정할 수 있습니다: • 0 —부호 있는 16 비트 정수형 (Integer16) • 1 —부호 없는 16 비트 정수형 (Unsigned Integer16) • 2 —부호 있는 32 비트 정수형 (Integer32) • 3 —부호 없는 32 비트 정수형 (Unsigned Integer32) • 4 —부동소수점 32 비트 실수형 (Float32) • 5 —부동소수점 64 비트 실수형 (Float64)
N	`N`	[number] 기본값: 10	
Probability	`PROBABILITY`	[number] 기본값: 0.5	

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 **명칭** 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 범위 (xmin, xmax, ymin, ymax) 를 지정합니다. 다음 가운데 하나를 선택할 수 있습니다: • 캔버스 범위 사용 • 캔버스에서 범위 선택 • 레이어 범위 사용… 타일 크기의 배수가 되도록 내부적으로 확장될 것입니다.
Target CRS	`TARGET_CRS`	[crs] 기본값: Project CRS	산출 래스터 레이어 용 좌표계
Pixel size	`PIXEL_SIZE`	[number] 기본값: 1.0	맵 단위의 픽셀 크기 (X=Y) 입니다. 최소값은 0.01 입니다.
Output raster	`OUTPUT`	[raster] 기본값: `[Save to temporary file]`	산출 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장…

고급 파라미터

라벨	이름	유형	설명
Output raster data type	`OUTPUT_TYPE` 기본값: 0	[enumeration]	산출 래스터 파일의 데이터 유형을 정의합니다. 다음 가운데 하나로 지정할 수 있습니다: • 0 —부동소수점 32 비트 실수형 (Float32) • 1 —부동소수점 64 비트 실수형 (Float64)
Lambda	`LAMBDA`	[number] 기본값: 1.0	

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 **명칭** 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 범위 (xmin, xmax, ymin, ymax) 를 지정합니다. 다음 가운데 하나를 선택할 수 있습니다: • 캔버스 범위 사용 • 캔버스에서 범위 선택 • 레이어 범위 사용… 타일 크기의 배수가 되도록 내부적으로 확장될 것입니다.
Target CRS	`TARGET_CRS`	[crs] 기본값: Project CRS	산출 래스터 레이어 용 좌표계
Pixel size	`PIXEL_SIZE`	[number] 기본값: 1.0	맵 단위의 픽셀 크기 (X=Y) 입니다. 최소값은 0.01 입니다.
Output raster	`OUTPUT`	[raster] 기본값: `[Save to temporary file]`	산출 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장…

고급 파라미터

라벨	이름	유형	설명
Output raster data type	`OUTPUT_TYPE` 기본값: 0	[enumeration]	산출 래스터 파일의 데이터 유형을 정의합니다. 다음 가운데 하나로 지정할 수 있습니다: • 0 —부동소수점 32 비트 실수형 (Float32) • 1 —부동소수점 64 비트 실수형 (Float64)
Alpha	`ALPHA`	[number] 기본값: 1.0	
Beta	`BETA`	[number] 기본값: 1.0	

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 **명칭** 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 범위 (xmin, xmax, ymin, ymax) 를 지정합니다. 다음 가운데 하나를 선택할 수 있습니다: • 캔버스 범위 사용 • 캔버스에서 범위 선택 • 레이어 범위 사용… 타일 크기의 배수가 되도록 내부적으로 확장될 것입니다.
Target CRS	`TARGET_CRS`	[crs] 기본값: Project CRS	산출 래스터 레이어 용 좌표계
Pixel size	`PIXEL_SIZE`	[number] 기본값: 1.0	맵 단위의 픽셀 크기 (X=Y) 입니다. 최소값은 0.01 입니다.
Output raster	`OUTPUT`	[raster] 기본값: `[Save to temporary file]`	산출 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장…

고급 파라미터

라벨	이름	유형	설명
Output raster data type	`OUTPUT_TYPE` 기본값: 0	[enumeration]	산출 래스터 파일의 데이터 유형을 정의합니다. 다음 가운데 하나로 지정할 수 있습니다: • 0 — 부호 있는 16 비트 정수형 (Integer16) • 1 — 부호 없는 16 비트 정수형 (Unsigned Integer16) • 2 — 부호 있는 32 비트 정수형 (Integer32) • 3 — 부호 없는 32 비트 정수형 (Unsigned Integer32) • 4 — 부동소수점 32 비트 실수형 (Float32) • 5 — 부동소수점 64 비트 실수형 (Float64)
Probability	`PROBABILITY`	[number] 기본값: 0.5	

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 **명칭** 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 범위 (xmin, xmax, ymin, ymax) 를 지정합니다. 다음 가운데 하나를 선택할 수 있습니다: • 캔버스 범위 사용 • 캔버스에서 범위 선택 • 레이어 범위 사용… 타일 크기의 배수가 되도록 내부적으로 확장될 것입니다.
Target CRS	`TARGET_CRS`	[crs] 기본값: Project CRS	산출 래스터 레이어 용 좌표계
Pixel size	`PIXEL_SIZE`	[number] 기본값: 1.0	맵 단위의 픽셀 크기 (X=Y) 입니다. 최소값은 0.01 입니다.
Output raster	`OUTPUT`	[raster] 기본값: `[Save to temporary file]`	산출 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장…

고급 파라미터

라벨	이름	유형	설명
Output raster data type	`OUTPUT_TYPE` 기본값: 0	[enumeration]	산출 래스터 파일의 데이터 유형을 정의합니다. 다음 가운데 하나로 지정할 수 있습니다: • 0 —부호 있는 16 비트 정수형 (Integer16) • 1 —부호 없는 16 비트 정수형 (Unsigned Integer16) • 2 —부호 있는 32 비트 정수형 (Integer32) • 3 —부호 없는 32 비트 정수형 (Unsigned Integer32) • 4 —부동소수점 32 비트 실수형 (Float32) • 5 —부동소수점 64 비트 실수형 (Float64)
Distribution parameter k	`K_PARAMETER`	[number] 기본값: 10	
Probability	`PROBABILITY`	[number] 기본값: 0.5	

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 **명칭** 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 범위 (xmin, xmax, ymin, ymax) 를 지정합니다. 다음 가운데 하나를 선택할 수 있습니다: • 캔버스 범위 사용 • 캔버스에서 범위 선택 • 레이어 범위 사용… 타일 크기의 배수가 되도록 내부적으로 확장될 것입니다.
Target CRS	`TARGET_CRS`	[crs] 기본값: Project CRS	산출 래스터 레이어 용 좌표계
Pixel size	`PIXEL_SIZE`	[number] 기본값: 1.0	맵 단위의 픽셀 크기 (X=Y) 입니다. 최소값은 0.01 입니다.
Output raster	`OUTPUT`	[raster] 기본값: `[Save to temporary file]`	산출 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장…

고급 파라미터

라벨	이름	유형	설명
Output raster data type	`OUTPUT_TYPE` 기본값: 0	[enumeration]	산출 래스터 파일의 데이터 유형을 정의합니다. 다음 가운데 하나로 지정할 수 있습니다: • 0 —부동소수점 32 비트 실수형 (Float32) • 1 —부동소수점 64 비트 실수형 (Float64)
Mean of normal distribution	`MEAN`	[number] 기본값: 0.0	
Standard deviation of normal distribution	`STDDEV`	[number] 기본값: 1.0	

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 **명칭** 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 범위 (xmin, xmax, ymin, ymax) 를 지정합니다. 다음 가운데 하나를 선택할 수 있습니다: • 캔버스 범위 사용 • 캔버스에서 범위 선택 • 레이어 범위 사용… 타일 크기의 배수가 되도록 내부적으로 확장될 것입니다.
Target CRS	TARGET_CRS	[crs] 기본값: Project CRS	산출 래스터 레이어 용 좌표계
Pixel size	PIXEL_SIZE	[number] 기본값: 1.0	맵 단위의 픽셀 크기 (X=Y) 입니다. 최소값은 0.01 입니다.
Output raster	OUTPUT	[raster] 기본값: [Save to temporary file]	산출 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장…

고급 파라미터

라벨	이름	유형	설명
Output raster data type	`OUTPUT_TYPE` 기본값: 0	[enumeration]	산출 래스터 파일의 데이터 유형을 정의합니다. 다음 가운데 하나로 지정할 수 있습니다: • 0 —부호 있는 16 비트 정수형 (Integer16) • 1 —부호 없는 16 비트 정수형 (Unsigned Integer16) • 2 —부호 있는 32 비트 정수형 (Integer32) • 3 —부호 없는 32 비트 정수형 (Unsigned Integer32) • 4 —부동소수점 32 비트 실수형 (Float32) • 5 —부동소수점 64 비트 실수형 (Float64)
Mean	`MEAN`	[number] 기본값: 1.0	

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 **명칭** 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 범위 (xmin, xmax, ymin, ymax) 를 지정합니다. 다음 가운데 하나를 선택할 수 있습니다: • 캔버스 범위 사용 • 캔버스에서 범위 선택 • 레이어 범위 사용… 타일 크기의 배수가 되도록 내부적으로 확장될 것입니다.
Target CRS	`TARGET_CRS`	[crs] 기본값: Project CRS	산출 래스터 레이어 용 좌표계
Pixel size	`PIXEL_SIZE`	[number] 기본값: 1.0	맵 단위의 픽셀 크기 (X=Y) 입니다. 최소값은 0.01 입니다.
Output raster	`OUTPUT`	[raster] 기본값: `[Save to temporary file]`	산출 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장…

고급 파라미터

라벨	이름	유형	설명
Output raster data type	`OUTPUT_TYPE` 기본값: 5	[enumeration]	산출 래스터 파일의 데이터 유형을 정의합니다. 다음 가운데 하나로 지정할 수 있습니다: • 0 —바이트 (Byte) • 1 —부호 있는 16 비트 정수형 (Integer16) • 2 —부호 없는 16 비트 정수형 (Unsigned Integer16) • 3 —부호 있는 32 비트 정수형 (Integer32) • 4 —부호 없는 32 비트 정수형 (Unsigned Integer32) • 5 —부동소수점 32 비트 실수형 (Float32) • 6 —부동소수점 64 비트 실수형 (Float64)
Lower bound for random number range	`LOWER_BOUND`	[number] 기본값: 0.0	
Upper bound for random number range	`UPPER_BOUND`	[number] 기본값: 0.0	

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 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

24.1.11 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.

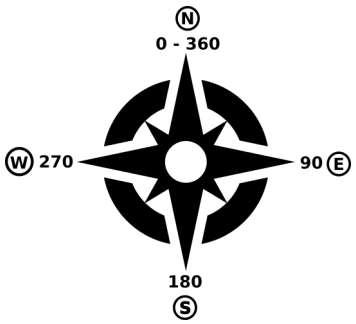


그림 24.16: 향 값

다음 그림은 색상표로 재범주화된 향 레이어입니다:

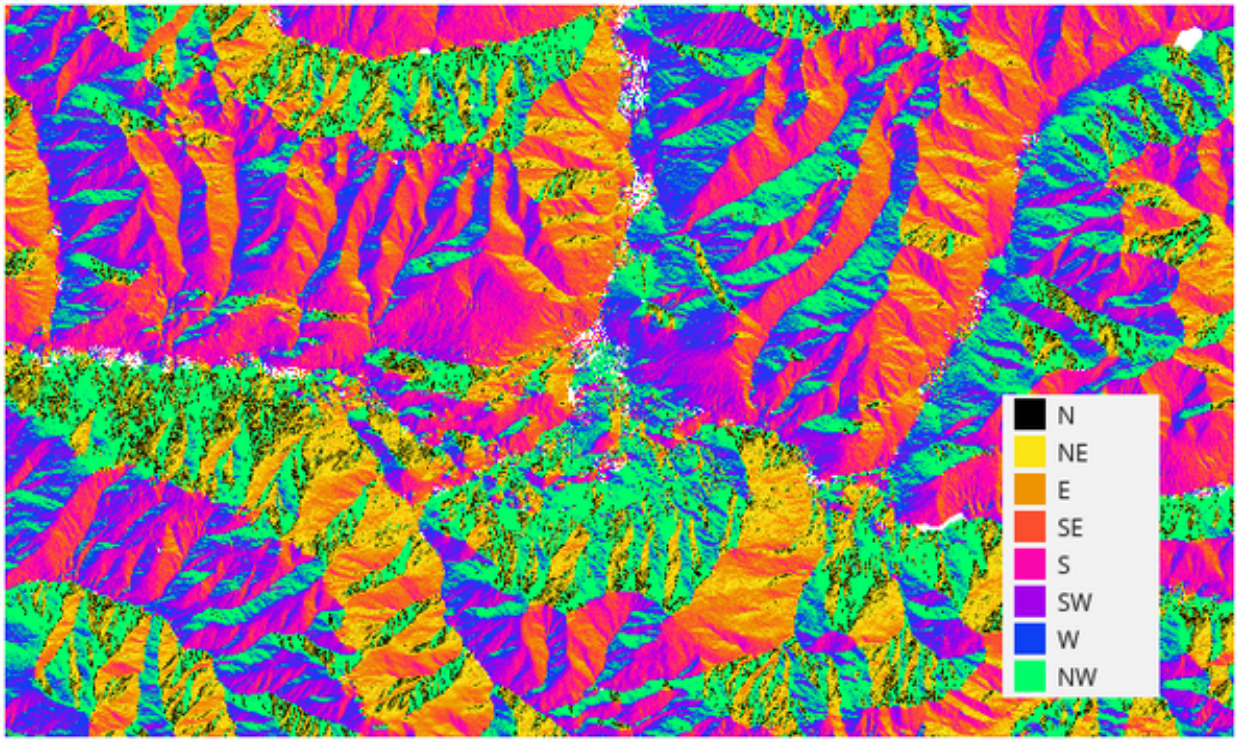


그림 24.17: 재범주화된 향 레이어

파라미터

라벨	명칭	유형	설명
Elevation layer	`INPUT`	[raster]	DTM 래스터 레이어
Z factor	`Z_FACTOR`	[number] 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]	산출 향 래스터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어로 저장 (`TEMPORARY_OUTPUT`) • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Aspect	OUTPUT	[raster]	산출 향 래스터 레이어

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 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.

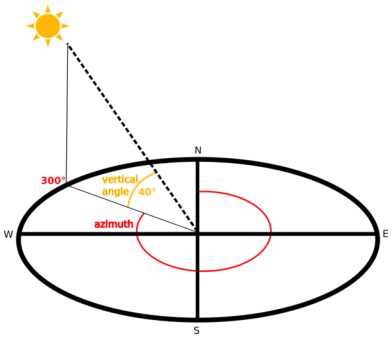


그림 24.18: 방위각 및 수직 각도

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:

파라미터

라벨	명칭	유형	설명
Elevation layer	INPUT	[raster]	DTM 래스터 레이어

다음 페이지에 계속

표 24.49 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Z factor	`Z_FACTOR`	[number] 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`	[number] 기본값: 300.0	태양의 수평 각도를 (도 단위 시계 방향으로) 설정합니다. 0 에서 360 까지의 범위로, 0 이 북쪽입니다.
Vertical angle	`V_ANGLE`	[number] 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]	산출 음영기복 래스터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어로 저장 (`TEMPORARY_OUTPUT`) • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Hillshade	`OUTPUT`	[raster]	산출 음영기복 래스터 레이어

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

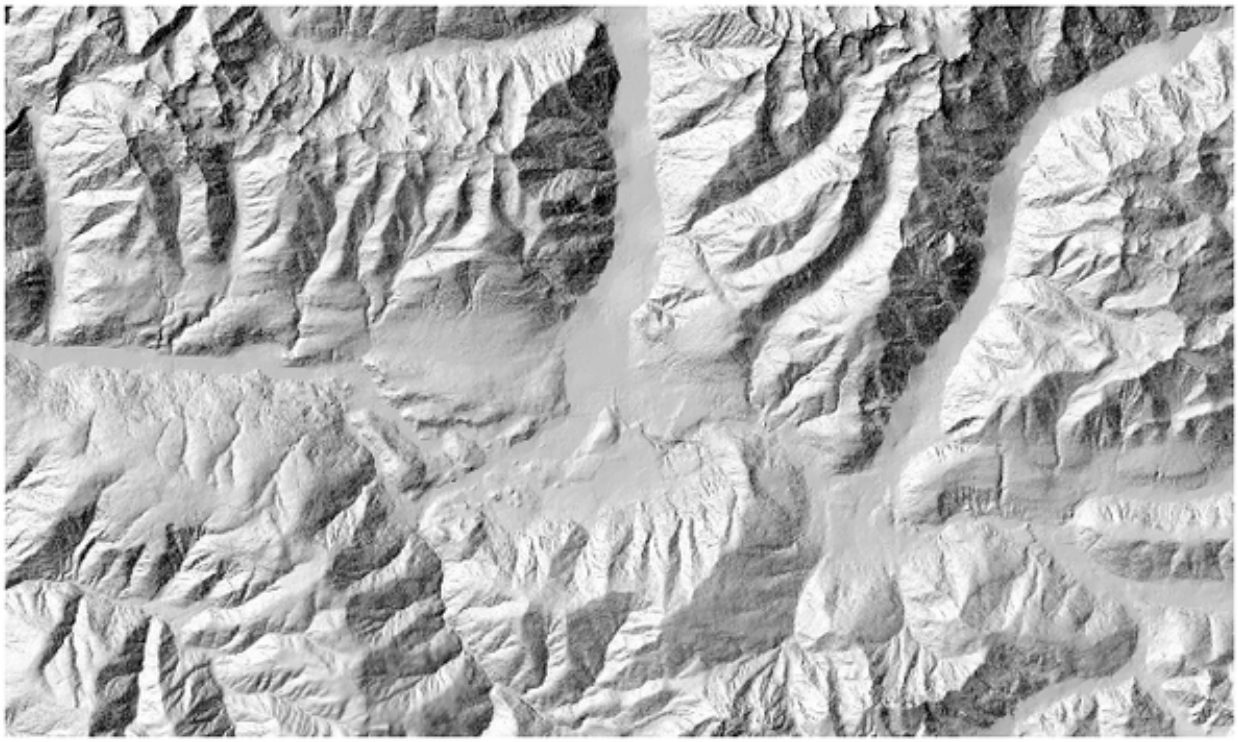


그림 24.19: 방위각이 300 이고 수직 각도가 45 인 음영기복 레이어

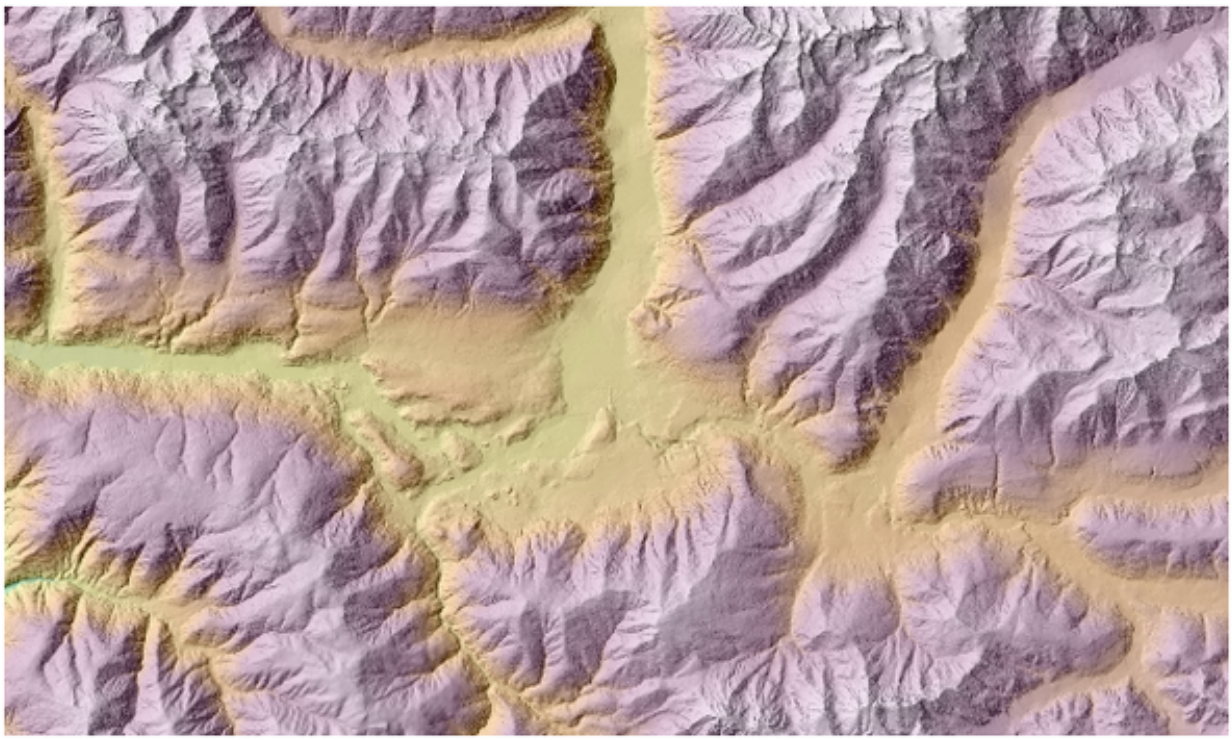


그림 24.20: 음영기복과 표고 레이어의 중첩

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	명칭	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`	[number] 기본값: 100.0	곡선들 사이의 수직 거리
Use % of area instead of absolute value	`USE_PERCENTAGE`	[boolean] 기본값: False	CSV 파일의 `Area` 필드에 절대 면적 대신 면적의 백분율을 작성합니다.
Hypsometric curves	`OUTPUT_DIRECTORY`	[folder]	고도분포곡선을 저장할 산출물 폴더를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어로 저장 (`TEMPORARY_OUTPUT`) • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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. 파일명은 `histogram_` 으로 시작해서, 그 뒤에 레이어명과 피처 ID 를 붙입니다.

	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

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 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

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.

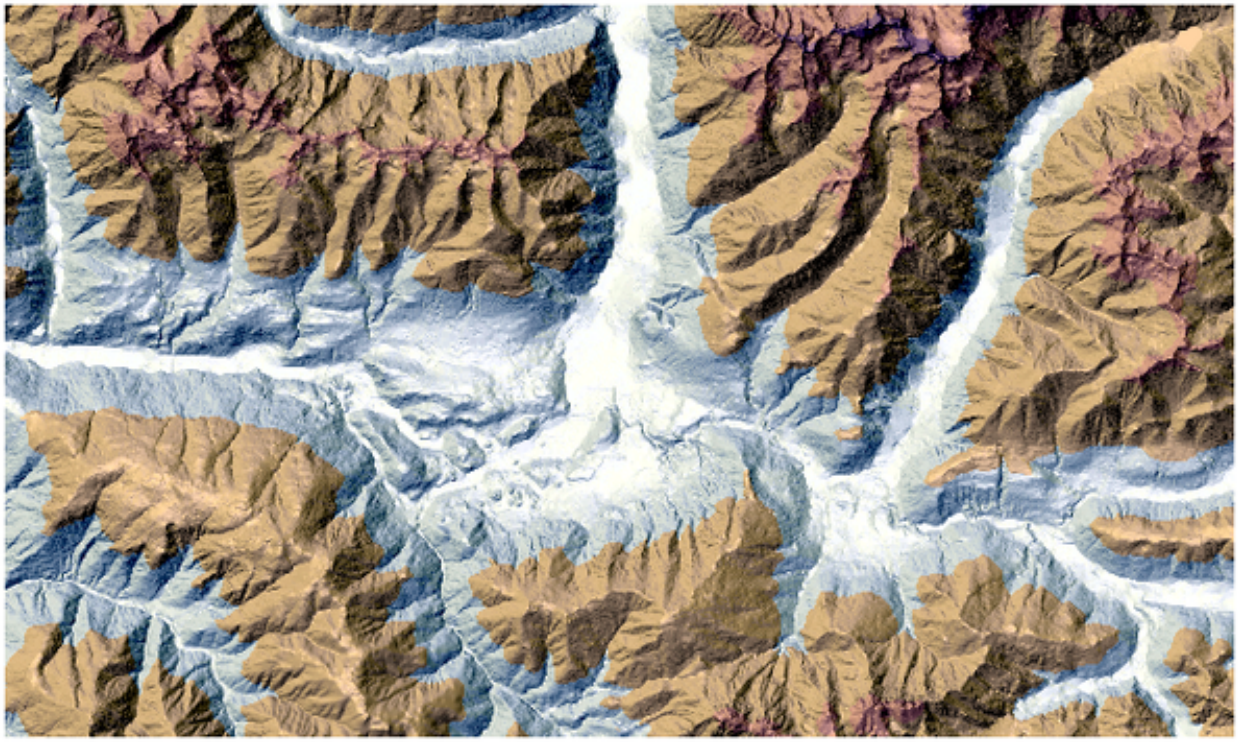


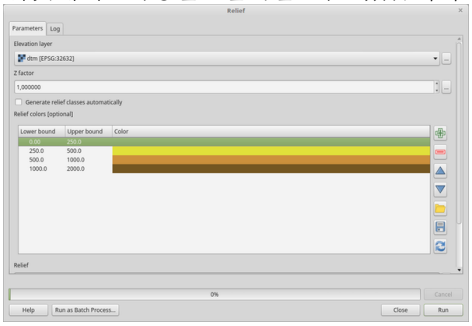
그림 24.21: 기복 레이어

파라미터

라벨	명칭	유형	설명
Elevation layer	INPUT	[raster]	DTM 래스터 레이어
Z factor	Z_FACTOR	[number] 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] 기본값: False	이 옵션을 활성화하면 알고리즘이 기복 색상 범주를 자동으로 생성할 것입니다.

다음 페이지에 계속

표 24.51 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Relief colors 부가적	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.  그림 24.22: 기복 색상 범주를 직접 설정하기 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] 기본값: [Save to temporary file]	산출 기복 래스터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어로 저장 (TEMPORARY_OUTPUT) • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.
Frequency distribution	FREQUENCY_DISTRIBUTION	[table] 기본값: [Skip output]	산출 도수 분포 (frequency distribution) 를 위한 CSV 테이블을 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 산출물 건너뛰기 • 임시 레이어로 저장 (TEMPORARY_OUTPUT) • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Relief	`OUTPUT`	[raster]	산출 기복 래스터 레이어
Frequency distribution	`OUTPUT`	[table]	산출 도수 분포

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 **명칭** 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.

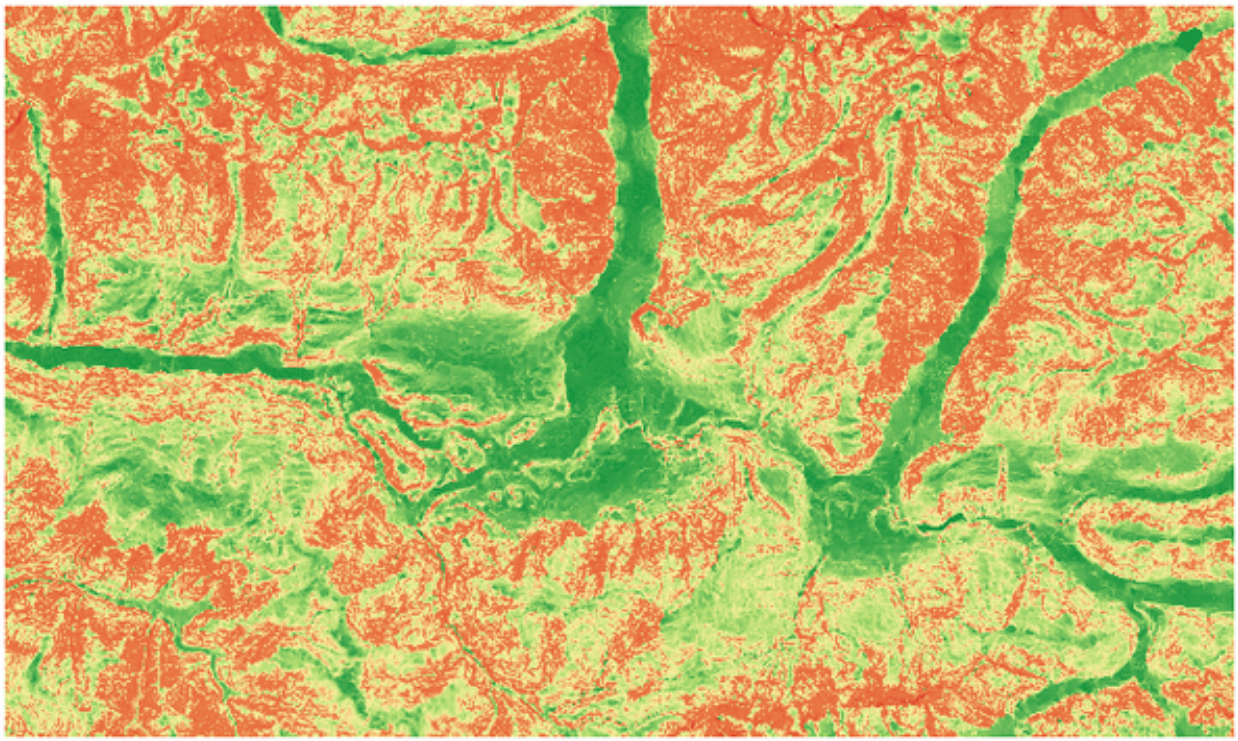


그림 24.23: 낮은 값 (빨간색) 에서 높은 값 (녹색) 까지의 변화를 보여주는 험상 레이어

파라미터

라벨	명칭	유형	설명
Elevation layer	`INPUT`	[raster]	DTM 래스터 레이어
Z factor	`Z_FACTOR`	[number] 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.
Ruggedness	`OUTPUT`	[raster] 기본값: `[Save to temporary file]`	산출 험상 래스터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어로 저장 (`TEMPORARY_OUTPUT`) • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Ruggedness	`OUTPUT`	[raster]	산출 험상 래스터 레이어

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

경사도

입력 래스터 레이어로부터 경사도 (slope) 를 계산합니다. 경사도란 지형의 기울어진 각도를 말하며, **도 단위** 로 표현됩니다.

다음 그림에서 표고값을 가진 DTM 레이어와 계산된 경사도를 볼 수 있습니다:

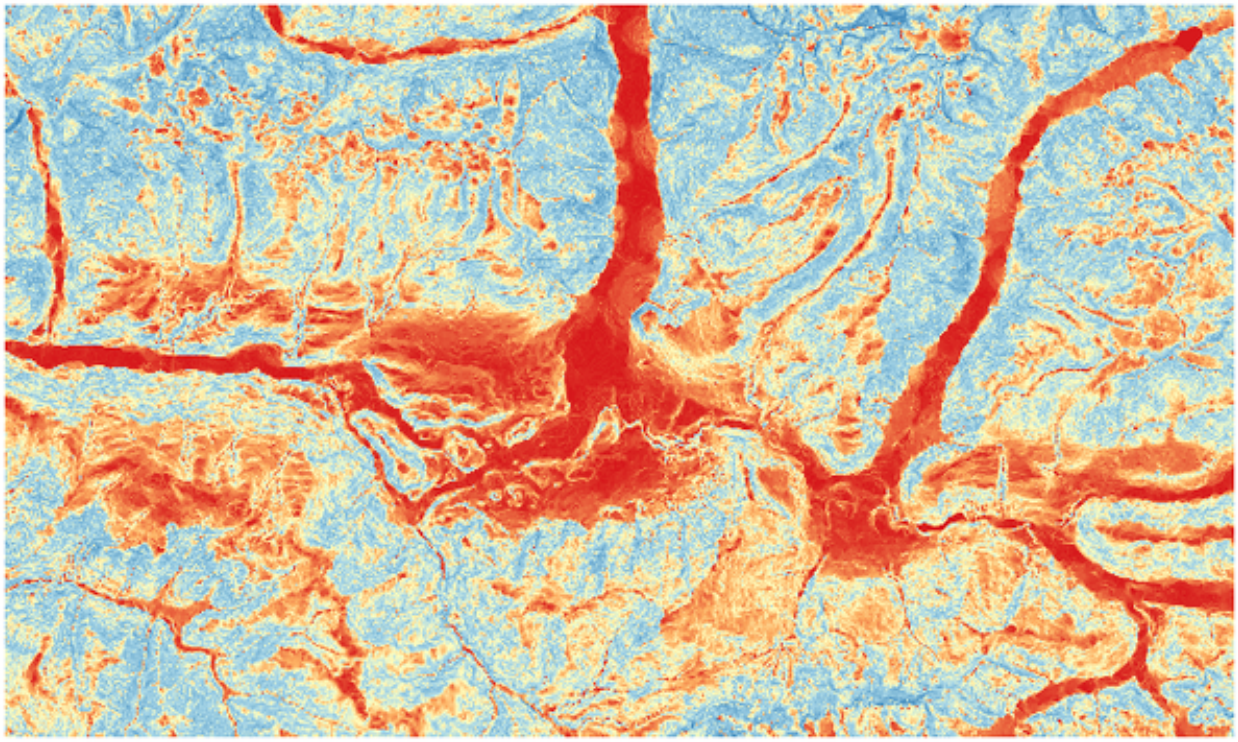


그림 24.24: 평평한 영역은 빨간색, 경사진 영역은 파란색

파라미터

라벨	명칭	유형	설명
Elevation layer	`INPUT`	[raster]	DTM 래스터 레이어
Z factor	`Z_FACTOR`	[number] 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] 기본값: `[Save to temporary file]`	산출 경사도 래스터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어로 저장 (`TEMPORARY_OUTPUT`) • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Slope	`OUTPUT`	[raster]	산출 경사도 래스터 레이어

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

24.1.12 래스터 도구

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	명칭	유형	설명
Minimum extent to render (xmin, xmax, ymin, ymax)	`EXTENT`	[extent]	산출 래스터 레이어의 범위를 지정합니다. 다음 가운데 하나로 선택할 수 있습니다: • 캔버스 범위 사용 • 캔버스에서 범위 선택 • 레이어 범위 사용 타일 크기의 배수가 되도록 내부적으로 확장될 것입니다.
Tile size	`TILE_SIZE`	[number] 기본값: 1024	산출 래스터 레이어의 타일 크기를 지정합니다. 최소값은 64 입니다.
Map units per pixel	`MAP_UNITS_PER_PIXEL`	[number] 기본값: 100.0	(맵 단위) 픽셀 크기입니다. 최소값은 0.0 입니다.
Make background transparent	`MAKE_BACKGROUND_TRANSPARENT`	[boolean] 기본값: False	투명한 배경을 가진 맵을 내보낼 수 있습니다. `True` 로 설정한 경우 (RGB 대신) RGBA 산출물을 생성할 것입니다.
Map theme to render 부가적	`MAP_THEME`	[enumeration]	렌더링을 위해 기존맵 테마 를 사용합니다.

다음 페이지에 계속

표 24.52 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Single layer to render 부가적	LAYER	[enumeration]	렌더링할 단일 레이어를 선택합니다.
Output layer	OUTPUT	[raster] 기본값: Save to temporary file	산출 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Output layer	OUTPUT	[raster]	산출 래스터 레이어

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 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.

파라미터

라벨	명칭	유형	설명
Input raster	INPUT	[raster]	공간 처리할 래스터
Band number	BAND	[number] 기본값: 1	래스터의 밴드
Fill value	FILL_VALUE	[number] 기본값: 1.0	NODATA 픽셀 용으로 사용할 값 설정
Output raster	OUTPUT	[raster] 기본값: [Save to temporary file]	산출 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장

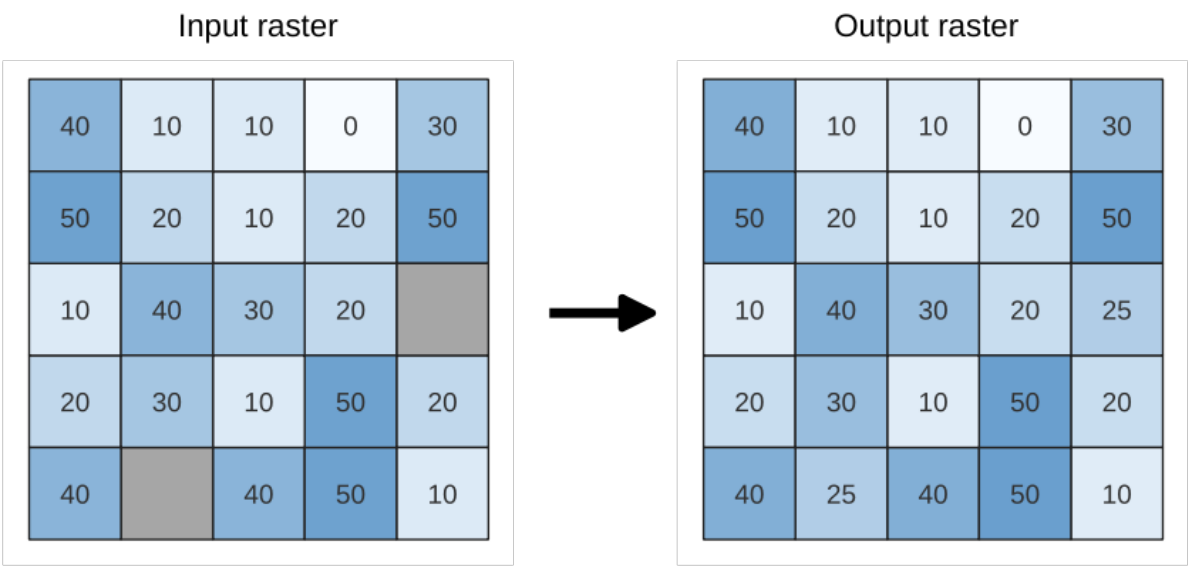


그림 24.25: 래스터의 NODATA 값 (회색) 채우기

산출물

라벨	명칭	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 **명칭** 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.

파라미터

라벨	명칭	유형	설명
Extent (xmin, xmax, ymin, ymax)	`EXTENT`	[extent]	타일의 범위를 지정합니다. 다음 가운데 하나로 선택할 수 있습니다: • 캔버스 범위 사용 • 캔버스에서 범위 선택 • 레이어 범위 사용 타일 크기의 배수가 되도록 내부적으로 확장될 것입니다.
Minimum zoom	`ZOOM_MIN`	[number] 기본값: 12	최소값은 0, 최대값은 25 입니다.
Maximum zoom	`ZOOM_MAX`	[number] 기본값: 12	최소값은 0, 최대값은 25 입니다.
DPI	`DPI`	[number] 기본값: 96	최소값은 48, 최대값은 600 입니다.
Background color 부가적	`BACK-GROUND_COLOR`	[color] 기본값: QColor(0, 0, 0, 0)	타일용 배경 색상을 선택합니다.
Tile format	`TILE_FORMAT`	[enumeration] 기본값: 0	다음 가운데 하나로 지정할 수 있습니다: • 0 —PNG • 1 —JPG
Quality (JPG only) 부가적	`QUALITY`	[number] 기본값: 75	최소값은 1, 최대값은 100 입니다.

다음 페이지에 계속

표 24.53 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Metatile size 부가적	METATILESIZE	[number] 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 부가적	TILE_WIDTH	[number] 기본값: 256	최소값은 1, 최대값은 4096 입니다.
Tile height 부가적	TILE_HEIGHT	[number] 기본값: 256	최소값은 1, 최대값은 4096 입니다.
Use inverted tile Y axis (TMS conventions) 부가적	TMS_CONVENTION	[boolean] 기본값: False	
Output directory	OUTPUT_DIRECTORY	[folder] 기본값: [Save to temporary folder]	산출 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 산출물 건너뛰기 • 임시 디렉터리에 저장 • 디렉터리에 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.
Output html (Leaflet)	OUTPUT_HTML	[html] 기본값: [Save to temporary file]	산출 HTML 파일을 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 산출물 건너뛰기 • 임시 파일로 저장 • 파일로 저장

산출물

라벨	명칭	유형	설명
Output directory	OUTPUT_DIRECTORY	[folder]	산출물 디렉터리 (타일용)
Output html (Leaflet)	OUTPUT_HTML	[html]	산출 HTML (리플렛) 파일

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 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	명칭	유형	설명
Extent (xmin, xmax, ymin, ymax)	`EXTENT`	[extent]	타일의 범위를 지정합니다. 다음 가운데 하나로 선택할 수 있습니다: • 캔버스 범위 사용 • 캔버스에서 범위 선택 • 레이어 범위 사용 타일 크기의 배수가 되도록 내부적으로 확장될 것입니다.
Minimum zoom	`ZOOM_MIN`	[number] 기본값: 12	최소값은 0, 최대값은 25 입니다.
Maximum zoom	`ZOOM_MAX`	[number] 기본값: 12	최소값은 0, 최대값은 25 입니다.
DPI	`DPI`	[number] 기본값: 96	최소값은 48, 최대값은 600 입니다.
Background color 부가적	`BACK-GROUND_COLOR`	[color] 기본값: QColor(0, 0, 0, 0)	타일용 배경 색상을 선택합니다.
Tile format	`TILE_FORMAT`	[enumeration] 기본값: 0	다음 가운데 하나로 지정할 수 있습니다: • 0 —PNG • 1 —JPG
Quality (JPG only) 부가적	`QUALITY`	[number] 기본값: 75	최소값은 1, 최대값은 100 입니다.
Metatile size 부가적	`METATILESIZE`	[number] 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] 기본값: `[Save to temporary file]`	산출 파일을 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 산출물 건너뛰기 • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Output file (for MBTiles)	OUTPUT_FILE	[file]	산출 파일

파이썬 코드

알고리즘 ID: qgis: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 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

24.1.13 벡터 분석

필드에 대한 기본 통계

벡터 레이어의 속성 테이블에 있는 필드에 대한 기본 통계를 생성합니다.

숫자, 날짜, 시간 및 문자열 필드를 지원합니다.

필드 유형에 따라 반환되는 통계가 달라질 것입니다.

통계는 HTML 파일로 생성되며 *Processing → Result Viewer* 메뉴로 볼 수 있습니다.

기본 메뉴: *Vector → Analysis Tools*

파라미터

라벨	명칭	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	OUTPUT_HTML_FILE	[html]	계산된 통계를 가진 HTML 파일

산출물

라벨	명칭	유형	설명
Statistics	OUTPUT_HTML_FILE	[html]	계산된 통계를 가진 HTML 파일
Count	COUNT	[number]	
Number of unique values	UNIQUE	[number]	
Number of empty (null) values	EMPTY	[number]	

다음 페이지에 계속

표 24.55 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Number of non-empty values	FILLED	[number]	
Minimum value	MIN	[same as input]	
Maximum value	MAX	[same as input]	
Minimum length	MIN_LENGTH	[number]	
Maximum length	MAX_LENGTH	[number]	
Mean length	MEAN_LENGTH	[number]	
Coefficient of Variation	CV	[number]	
Sum	SUM	[number]	
Mean value	MEAN	[number]	
Standard deviation	STD_DEV	[number]	
Range	RANGE	[number]	
Median	MEDIAN	[number]	
Minority (rarest occurring value)	MINORITY	[same as input]	
Majority (most frequently occurring value)	MAJORITY	[same as input]	
First quartile	FIRSTQUARTILE	[number]	
Third quartile	THIRDQUARTILE	[number]	
Interquartile Range (IQR)	IQR	[number]	

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 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 명칭을 가진 필드를 가지고 있는 경우, 기존 필드를 재명명할 것입니다. (“name_2”, “name_3”등으로 첫 번째로 일치하지 않는 명칭을 찾아 필드명을 수정할 것입니다.)

파라미터

라벨	명칭	유형	설명
Line layer	`INPUT`	[vector: line]	오르막을 계산할 라인 레이어입니다. Z 값을 가지고 있어야만 합니다.
Climb layer	`OUTPUT`	[vector: line]	산출 (라인) 레이어

산출물

라벨	명칭	유형	설명
Climb layer	`OUTPUT`	[vector: line]	오르막 계산에서 나온 결과를 가진 새 속성을 담고 있는 라인 레이어
Total climb	`TOTALCLIMB`	[number]	입력 레이어에 있는 모든 라인 도형에 대한 오르막 합계
Total descent	`TOTALDESCENT`	[number]	입력 레이어에 있는 모든 라인 도형에 대한 내리막 합계
Minimum elevation	`MINELEVATION`	[number]	레이어에 있는 도형 가운데 최저 표고
Maximum elevation	`MAXELEVATION`	[number]	레이어에 있는 도형 가운데 최고 표고

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 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.

기본 메뉴: *Vector → Analysis Tools*

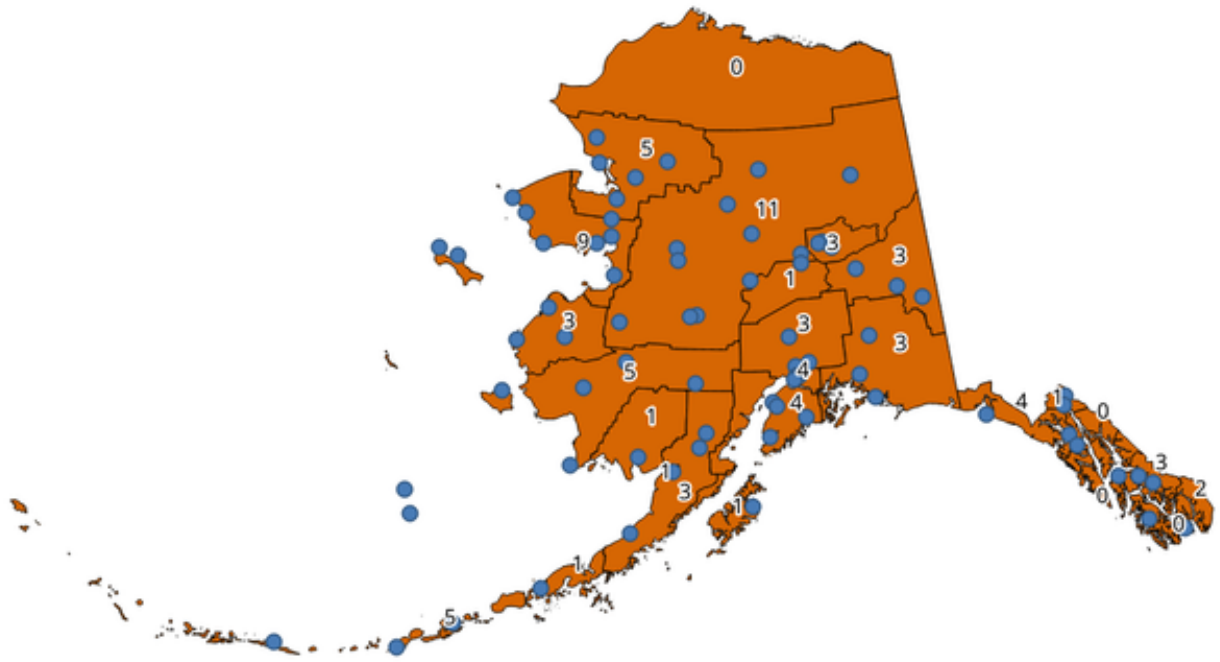


그림 24.26: 포인트 개수를 표시하는 폴리곤 라벨

파라미터

라벨	명칭	유형	설명
Polygons	`POLYGONS`	[vector: polygon]	포인트의 개수를 셀 피처를 가진 폴리곤 레이어
Points	`POINTS`	[vector: point]	개수를 셀 포인트 레이어
Weight field 부가적	`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 부가적	`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] 기본값: 'NUMPOINTS'	포인트 개수를 저장할 필드의 명칭
Count	`OUTPUT`	[vector: polygon]	산출 레이어를 지정합니다.

산출물

라벨	명칭	유형	설명
Count	`OUTPUT`	[vector: polygon]	포인트 개수를 가진 새 열을 담고 있는 속성 테이블을 가진 산출 레이어

DBSCAN 군집 형성

이상값 (noise) (DBSCAN) 알고리즘을 가진 응용 프로그램의 밀도 기반 공간 군집 형성의 2 차원 구현을 기반으로

표 24.57 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Cluster field name	FIELD_NAME	[string] 기본값: 'CLUSTER_ID'	연관 군집 번호를 저장할 필드의 명칭
Treat border points as noise (DBSCAN*) 부가적	DBSCAN*	[boolean] 기본값: False	True 로 설정하면, 군집의 경계 상에 있는 포인트를 군집되지 않은 포인트로 취급하고, 군집 내부에 있는 포인트만 군집된 것으로 태그합니다.
Clusters	OUTPUT	[vector: point]	군집 형성의 결과를 담은 벡터 레이어

산출물

라벨	명칭	유형	설명
Clusters	OUTPUT	[vector: point]	원본 피처와 함께 원본 피처가 속한 군집을 설정하는 필드를 가진 벡터 레이어
Number of clusters	NUM_CLUSTERS	[number]	발견된 군집의 개수

파이썬 코드

알고리즘 ID: qgis: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 **명칭** 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	명칭	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

다음 페이지에 계속

표 24.58 – 이전 페이지에서 계속

라벨	명칭	유형	설명
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: • 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. • 1 —표준 (N x T) 거리 매트릭스 • 2 —거리 매트릭스 요약 (평균, 표준 편차, 최소값, 최대값): 각 입력 포인트마다, 대상 포인트까지의 거리에 대한 통계를 보고합니다.
Use only the nearest (k) target points	NEAREST_POINTS	[number] 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]	

산출물

라벨	명칭	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 **명칭** 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.

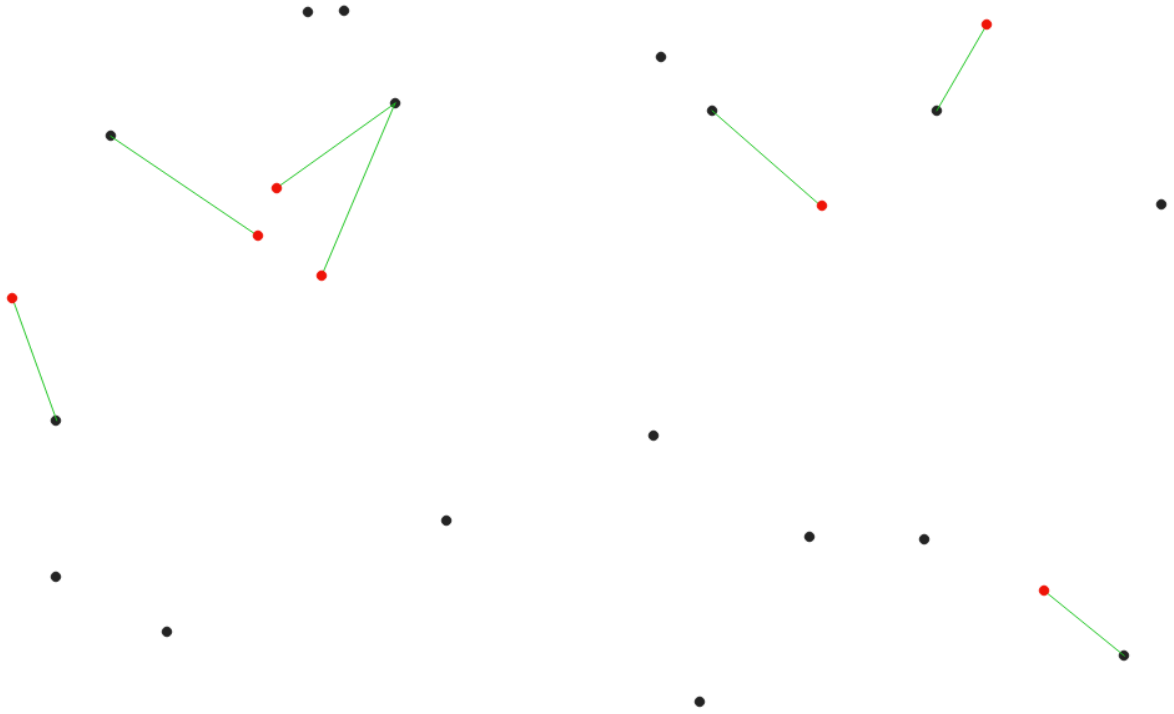


그림 24.27: 빨간색 입력 피처의 최근접 허브를 표시

더 보기:

최근접 허브까지의 거리 *(*포인트*)*, 최근접으로 속성 결합하기

파라미터

라벨	명칭	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 • 0 —미터 • 1 —피트 • 2 —마일 • 3 —킬로미터 • 4 —레이어 단위
Hub distance	`OUTPUT`	[vector: line]	거리 매트릭스 산출물을 저장할 라인 벡터 레이어

산출물

라벨	명칭	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 **명칭** 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	명칭	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 • 0 —미터 • 1 —피트 • 2 —마일 • 3 —킬로미터 • 4 —레이어 단위
Hub distance	`OUTPUT`	[vector: point]	거리 매트릭스 산출물을 저장할 포인트 벡터 레이어

산출물

라벨	명칭	유형	설명
Hub distance	`OUTPUT`	[vector: point]	입력 피처, 최근접 피처의 식별자, 계산된 거리의 속성을 가진 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 **명칭** 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.

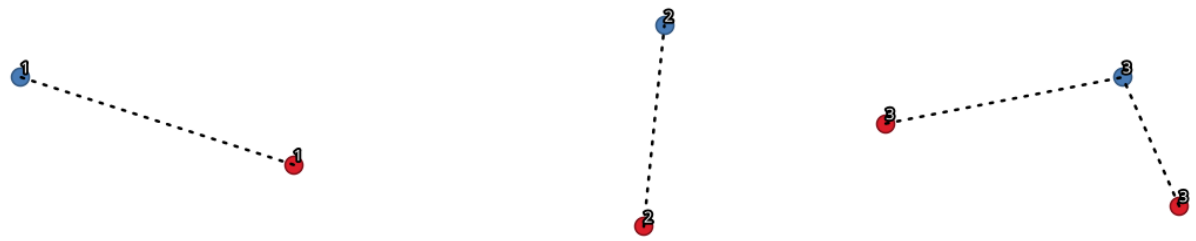


그림 24.28: 공통 필드/속성을 바탕으로 포인트 결합

파라미터

라벨	명칭	유형	설명
Hub layer	HUBS	[vector: any]	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) 부가적	HUB_FIELDS	[tablefield: any] [list]	복사할 허브 레이어의 필드 (들). 아무 필드도 선택하지 않을 경우 모든 필드를 복사합니다.
Spoke layer	SPOKES	[vector: any]	부가적인 스포크 포인트 레이어
Spoke ID field	SPOKE_FIELD	[tablefield: any]	결합을 위한 ID 를 가진 스포크 레이어의 필드
Spoke layer fields to copy (leave empty to copy all fields) 부가적	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.
Distance between vertices (geodesic lines only)	GEODESIC_DISTANCE	[number] 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.
Hub lines	OUTPUT	[vector: line]	산출되는 라인 레이어

산출물

라벨	명칭	유형	설명
Hub lines	`OUTPUT`	[vector: line]	산출되는 라인 레이어

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 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.

더 보기:

DBSCAN 군집 형성

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	분석할 레이어
Number of clusters	`CLUSTERS`	[number] 기본값: 5	피처를 통해 생성할 군집의 개수
Cluster field name	`FIELD_NAME`	[string] 기본값: 'CLUSTER_ID'	군집 개수 필드의 명칭
Clusters	`OUTPUT`	[vector: any]	생성된 군집을 담은 벡터 레이어

산출물

라벨	명칭	유형	설명
Clusters	`OUTPUT`	[vector: any]	원본 피처와 함께 원본 피처가 속한 군집을 지정하는 필드를 가진 벡터 레이어

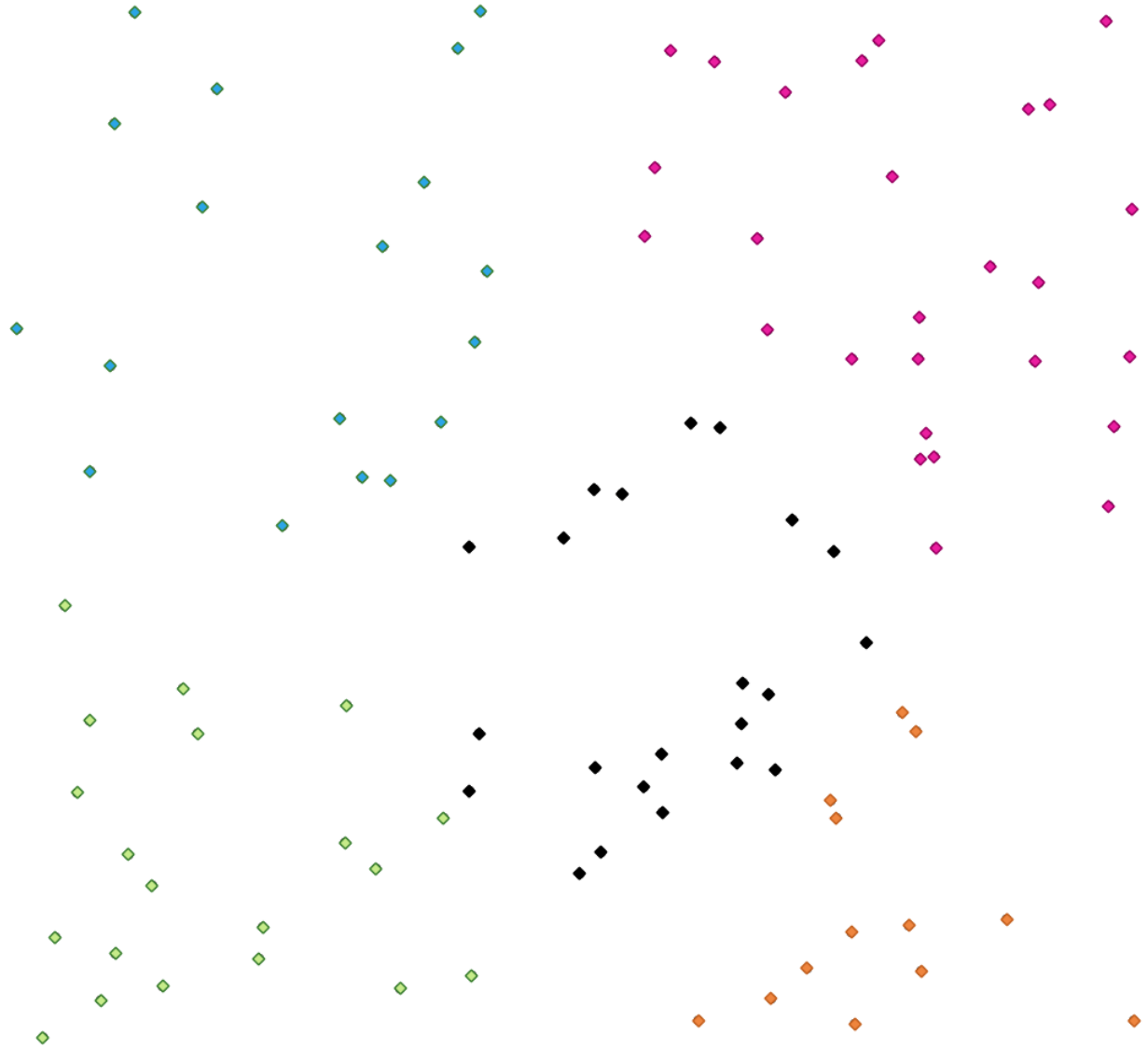


그림 24.29: 범주가 5 개인 포인트 군집들

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 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	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	분석할 레이어
Target field(s)	`FIELDS`	[tablefield: any]	분석할 필드
Unique values	`OUTPUT`	[table]	유일 값을 가진 요약 테이블 레이어
HTML report	`OUTPUT_HTML_FILE`	[html]	유일 값의 HTML 보고서. *Processing → Results viewer* 메뉴로 볼 수 있습니다.

산출물

라벨	명칭	유형	설명
Unique values	`OUTPUT`	[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`	[number]	입력 필드에 있는 유일 값들의 개수
UNIQUE_VALUES	`Unique values`	[string]	입력 필드에서 발견된 유일 값을 쉼표로 구분한 목록의 문자열

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 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	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	입력 벡터 레이어
Weight field 부가적	`WEIGHT`	[tablefield: numeric]	가중치를 적용한 평균을 작업하려는 경우 사용할 필드
Unique ID field	`UID`	[tablefield: numeric]	평균을 계산할 유일 필드
Mean coordinates	`OUTPUT`	[vector: point]	산출물을 저장할 (포인트 벡터) 레이어

산출물

라벨	명칭	유형	설명
Mean coordinates	`OUTPUT`	[vector: point]	산출되는 포인트 (들) 레이어

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

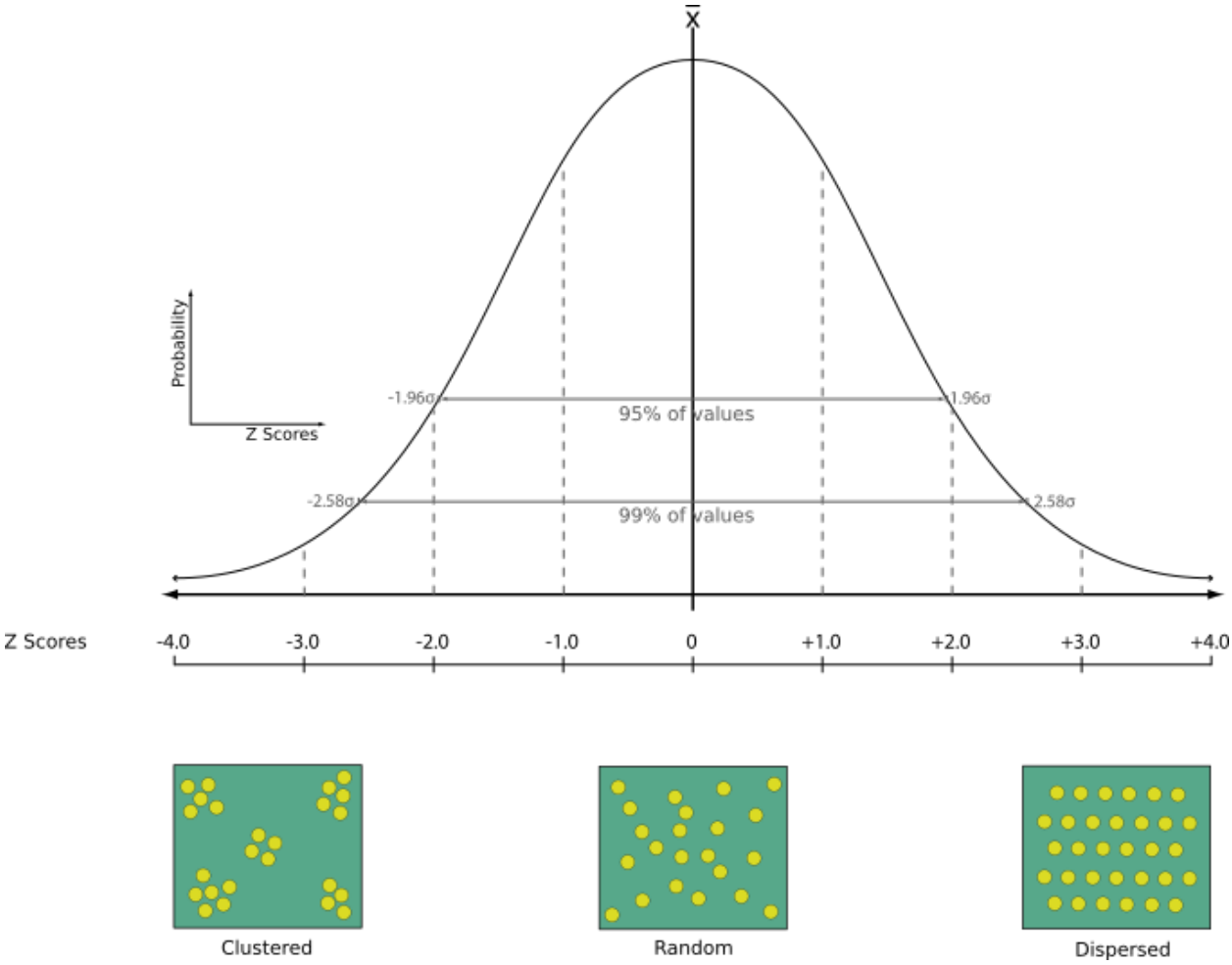
최근접 이웃 분석

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.



기본 메뉴: *Vector → Analysis Tools*

더 보기:

최근접으로 속성 결합하기

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: point]	통계를 계산할 포인트 벡터 레이어
Nearest neighbour	`OUTPUT_HTML_FILE`	[html]	계산된 통계를 가진 HTML 파일

산출물

라벨	명칭	유형	설명
Nearest neighbour	OUTPUT_HTML_FILE	[html]	계산된 통계를 가진 HTML 파일
Observed mean distance	OBSERVED_MD	[number]	관측 평균 거리
Expected mean distance	EXPECTED_MD	[number]	기대 평균 거리
Nearest neighbour index	NN_INDEX	[number]	최근접 이웃 지수
Number of points	POINT_COUNT	[number]	포인트의 개수
Z-Score	Z_SCORE	[number]	Z 점수 (Z-Score)

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 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.

파라미터

라벨	명칭	유형	설명
Input layer	INPUT	[vector: any]	입력 레이어
Overlap layers	LAYERS	[vector: any] [list]	중첩 레이어
Output layer	OUTPUT	[same as input] 기본값: [Create temporary layer]	산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (TEMPORARY_OUTPUT) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Output layer	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: `qgis: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 **명칭** 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	명칭	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) 부가적	VALUES_FIELD_NAME	[tablefield: any]	이 파라미터가 비어 있는 경우 개수만 계산할 것입니다.
Field(s) with categories	CATEGORIES_FIELD_NAME	[vector: any] [list]	(결합되어) 범주를 정의하는 필드들
Statistics by category	OUTPUT	[table]	생성된 통계를 저장할 테이블

산출물

라벨	명칭	유형	설명
Statistics by category	OUTPUT	[table]	통계를 담고 있는 테이블

분석되는 필드의 유형에 따라, 각 그룹화된 값에 대해 다음 통계를 반환합니다:

통계 정보	문자열	수치	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)	☑		
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 **명칭** 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.

기본 메뉴: *Vector → Analysis Tools*

파라미터

라벨	명칭	유형	설명
Lines	`LINES`	[vector: line]	입력 라인 벡터 레이어
Polygons	`POLYGONS`	[vector: polygon]	폴리곤 벡터 레이어
Lines length field name	`LEN_FIELD`	[string] 기본값: 'LENGTH'	라인 길이를 저장할 필드의 명칭
Lines count field name	`COUNT_FIELD`	[string] 기본값: 'COUNT'	라인 개수를 저장할 필드의 명칭
Line length	`OUTPUT`	[vector: polygon]	산출 폴리곤 벡터 레이어

산출물

라벨	명칭	유형	설명
Line length	`OUTPUT`	[vector: polygon]	라인 길이 및 개수 필드를 가진 산출 폴리곤 레이어

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

24.1.14 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.

제자리 피처 수정 옵션을 사용할 수 있습니다.

더 보기:

라인 오프셋시키기, 이동시킨 피처의 배열

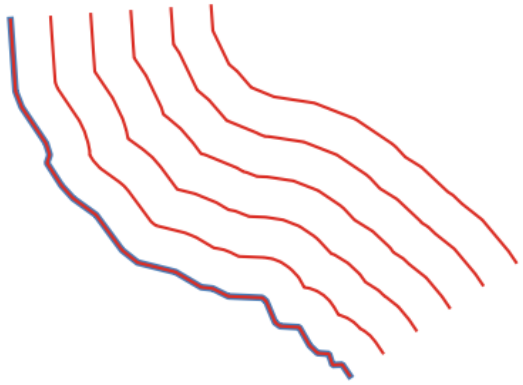


그림 24.30: 소스 피처는 파란색, 오프셋 피처는 빨간색

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: line]	오프셋에 사용할 입력 라인 벡터 레이어
Number of features to create	`COUNT`	[number] 기본값: 10	각 피처에 대해 생성할 오프셋 복사본의 개수
Offset step distance	`OFFSET`	[number] 기본값: 1.0	연속된 두 오프셋 복사본 사이의 거리
Segments	`SEGMENTS`	[number] 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: • 0 —둥글림 (Round) • 1 —마이터 (Miter) • 2 —베벨 (Bevel)
Miter limit	`MITER_LIMIT`	[number] 기본값: 2.0	마이터 결합 스타일에만 적용할 수 있으며, 마이터 결합 생성시 사용할 오프셋 곡선으로부터의 최장 거리를 제어합니다.
Offset lines	`OUTPUT`	[vector: line] 기본값: `[Create temporary layer]`	오프셋 피처를 담은 산출 라인 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Offset lines	`OUTPUT`	[vector: line]	오프셋 피처를 담은 산출 라인 레이어입니다. 원본 피처도 복사됩니다.

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 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.

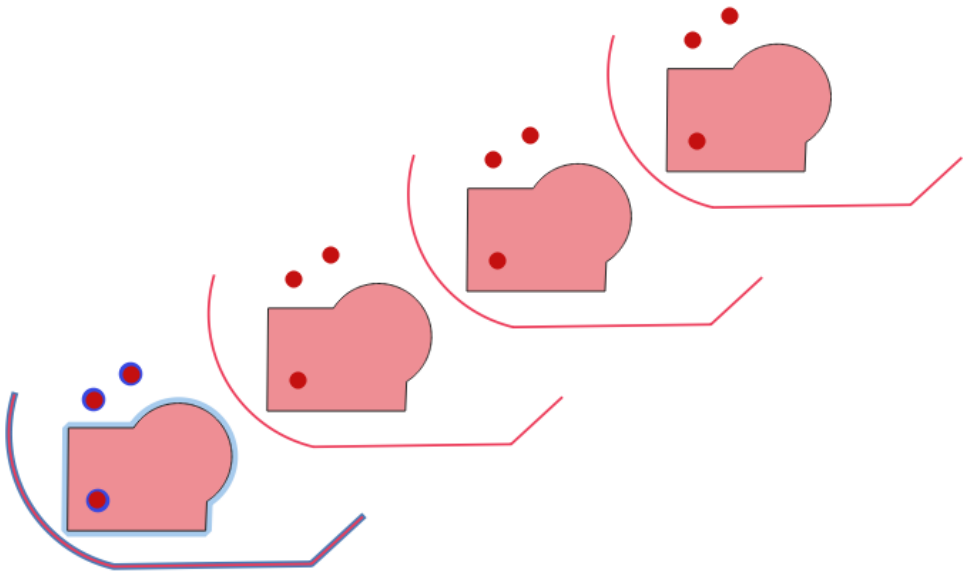


그림 24.31: 입력 레이어는 파란색, 이동된 피처를 담은 산출 레이어는 빨간색

제자리 피처 수정 옵션을 사용할 수 있습니다.

더 보기:

이동시키기 *(Translate)*, 오프셋 *(*평행*)* 라인 배열

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	이동시킬 입력 벡터 레이어
Number of features to create	`COUNT`	[number] 기본값: 10	각 피처에 대해 생성할 복사본의 개수
Step distance (x-axis)	`DELTA_X`	[number] 기본값: 0.0	X 축에 적용할 변위 (displacement)
Step distance (y-axis)	`DELTA_Y`	[number] 기본값: 0.0	Y 축에 적용할 변위 (displacement)
Step distance (z-axis)	`DELTA_Z`	[number] 기본값: 0.0	Z 축에 적용할 변위 (displacement)
Step distance (m values)	`DELTA_M`	[number] 기본값: 0.0	M 축에 적용할 변위 (displacement)
Translated	`OUTPUT`	[same as input] 기본값: `[Create temporary layer]`	이동된 피처 복사본을 담은 산출 벡터 레이어입니다. 원본 피처도 복사됩니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Translated	`OUTPUT`	[same as input]	이동된 피처 복사본을 담은 산출 벡터 레이어입니다. 원본 피처도 복사됩니다.

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 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:

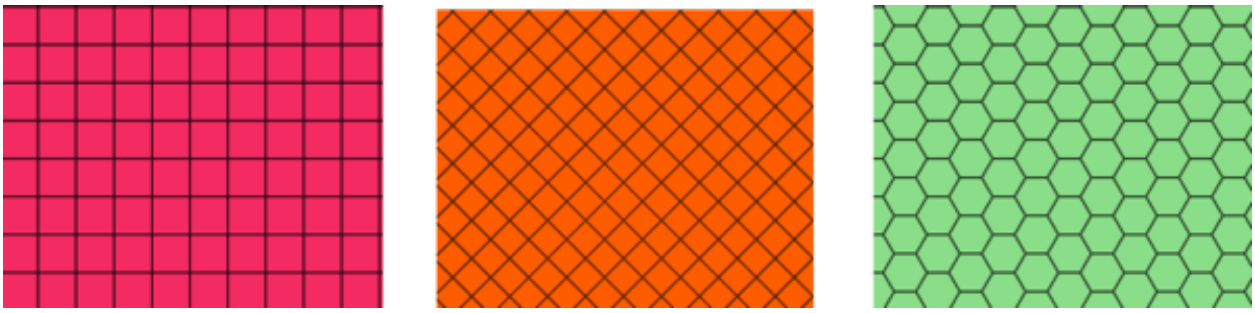


그림 24.32: 서로 다른 그리드 셀 형태

그리드에 있는 각 요소의 크기는 수평 및 수직 간격을 사용해서 정의됩니다.

산출 레이어의 좌표계를 반드시 정의해야만 합니다.

그리드의 범위 및 간격 값은 이 좌표계의 좌표 및 단위로 표현되어야만 합니다.

기본 메뉴: *Vector → Research Tools*

파라미터

라벨	명칭	유형	설명
Grid type	`TYPE`	[enumeration] 기본값: 0	그리드의 형태를 지정합니다. 다음 가운데 하나를 선택할 수 있습니다: • 0 —포인트 • 1 —라인 • 2 —직사각형 (폴리곤) • 3 —마름모꼴 (폴리곤) • 4 —육각형 (폴리곤)
Grid extent	`EXTENT`	[extent]	그리드의 범위
Horizontal spacing	`HSPACING`	[number] 기본값: 1.0	그리드 셀의 X 축 크기
Vertical spacing	`VSPACING`	[number] 기본값: 1.0	그리드 셀의 Y 축 크기
Horizontal overlay	`HOVERLAY`	[number] 기본값: 0.0	연속된 두 그리드 셀 사이의 X 축 중첩 거리
Vertical overlay	`VOVERLAY`	[number] 기본값: 0.0	연속된 두 그리드 셀 사이의 Y 축 중첩 거리
Grid CRS	`CRS`	[crs] 기본값: *Project CRS*	그리드에 적용할 좌표계

다음 페이지에 계속

표 24.65 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Grid	`OUTPUT`	[vector: any] 기본값: `[Create temporary layer]`	산출 그리드 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Grid	`OUTPUT`	[vector: any]	산출 그리드 벡터 레이어입니다. *Grid type* 에 따라 산출 도형 유형 (포인트, 라인 또는 폴리곤) 이 달라집니다.

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 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.

파라미터

라벨	명칭	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 부가적	`ZFIELD`	[tablefield: any]	Z 좌표를 담고 있는 필드
M field 부가적	`MFIELD`	[tablefield: any]	M 값을 담고 있는 필드

다음 페이지에 계속

표 24.66 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Target CRS	`TARGET_CRS`	[crs] 기본값: `EPSG:4326`	레이어에 사용할 좌표계입니다. 제공되는 좌표계가 규칙을 따른다고 가정합니다.
Points from table	`OUTPUT`	[vector: point] 기본값: `[Create temporary layer]`	산출 포인트 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Points from table	`OUTPUT`	[vector: point]	산출되는 포인트 레이어

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 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.

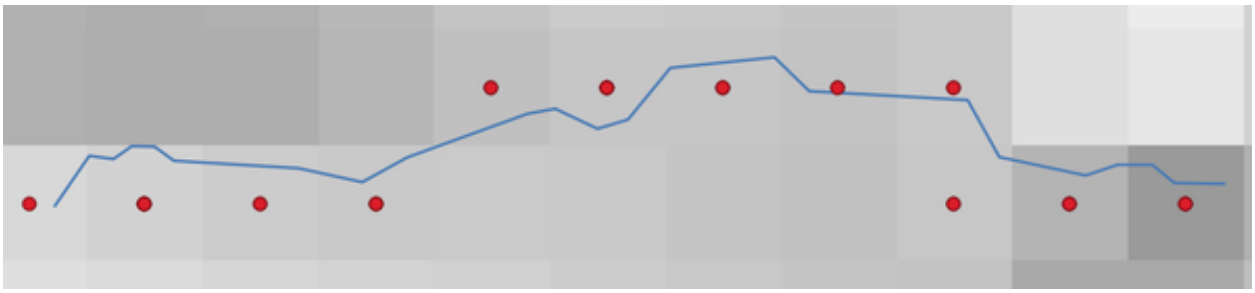


그림 24.33: 픽셀 중심 포인트

파라미터

라벨	명칭	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] 기본값: `[Create temporary layer]`	픽셀 중심을 담은 산출 포인트 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

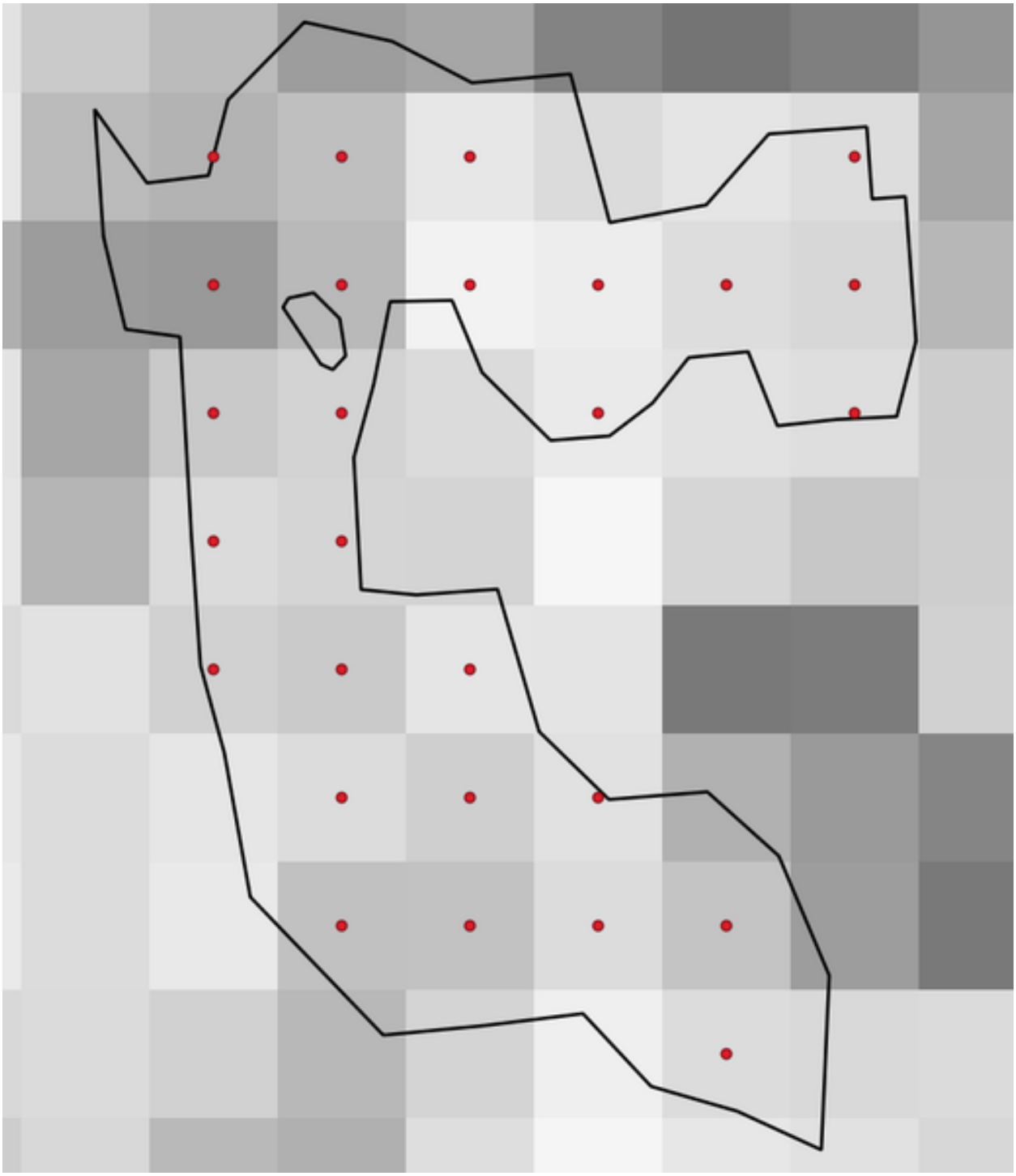


그림 24.34: 픽셀 중심 포인트

파라미터

라벨	명칭	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] 기본값: `[Create temporary layer]`	픽셀 중심을 담은 산출 포인트 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Points inside polygons	`OUTPUT`	[vector: point]	픽셀 중심을 담은 산출 포인트 레이어

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 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 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	명칭	유형	설명
Input folder	FOLDER	[folder]	지오태그된 사진을 담고 있는 소스 폴더를 가리키는 경로
Scan recursively	RECURSIVE	[boolean] 기본값: False	이 옵션을 활성화하면, 폴더는 물론 그 하위 폴더도 스캔할 것입니다.
Photos	OUTPUT	[vector: point] 기본값: [Create temporary layer]	지오태그된 사진에 대한 포인트 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (TEMPORARY_OUTPUT) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.
Invalid photos table 부가적	INVALID	[table] 기본값: [Skip output]	읽을 수 없거나 지오태그되지 않은 사진의 테이블을 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 산출물 건너뛰기 • 임시 레이어 생성 (TEMPORARY_OUTPUT) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 부가적	INVALID	[table]	읽을 수 없거나 지오태그되지 않은 사진의 테이블도 생성할 수 있습니다.

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

포인트를 경로로

입력 포인트 레이어에 있는 필드로 정의한 순서대로 포인트를 결합시켜, 포인트 레이어를 라인 레이어로 변환합니다. (순서 필드가 날짜 & 시간 필드인 경우, 서식을 지정해야만 합니다.)

라인 피처를 식별하기 위한 필드로 포인트를 그룹화할 수 있습니다.

라인 벡터 레이어는 물론, 산출 라인을 시작점과 일련의 (방위각에 상대적인) 방위/방향 및 거리로 서술하는 텍스트 파일도 산출할 수 있습니다.

파라미터

라벨	명칭	유형	설명
Input point layer	`INPUT`	[vector: point]	입력 포인트 벡터 레이어
Close path	`CLOSE_PATH`	[boolean] 기본값: False	이 옵션을 체크하면, 라인의 첫번째 및 마지막 포인트를 연결시켜 생성된 경로를 닫을 것입니다.
Order field	`ORDER_FIELD`	[tablefield: any]	포인트를 경로에 연결할 순서를 담고 있는 필드
Group field 부가적	`GROUP_FIELD`	[tablefield: any]	필드에 동일한 값을 가진 포인트 피처를 동일한 라인으로 그룹화할 것입니다. 이 파라미터를 설정하지 않으면, 모든 입력 포인트를 단일 경로로 연결합니다.
Date format (if order field is DateTime) 부가적	`DATE_FORMAT`	[string]	`Order field` 파라미터를 위한 서식입니다. `Order field` 가 날짜 & 시간 유형인 경우에만 이 파라미터를 지정하십시오.
Paths	`OUTPUT`	[vector: line] 기본값: `[Create temporary layer]`	경로를 담은 라인 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

다음 페이지에 계속

표 24.68 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Directory for text output	`OUTPUT_TEXT_DIR`	[folder] 기본값: `[Skip output]`	포인트와 경로를 서술하는 파일을 담을 디렉터리를 지정합니다. 다음 가운데 하나를 선택할 수 있습니다: • 산출물 건너뛰기 • 임시 디렉터리에 저장 • 디렉터리에 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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: `qgis: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 **명칭** 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.

포인트끼리 너무 가까워지는 일을 피하기 위해 최소 거리를 지정할 수 있습니다.

더 보기:

라인에 랜덤 포인트 생성하기

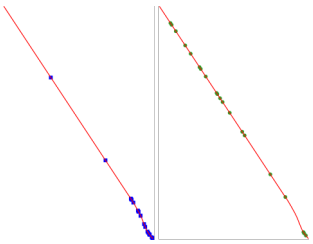


그림 24.35: 알고리즘 산출물의 예. 왼쪽: 라인을 따라 랜덤 포인트 생성하기, 오른쪽: 라인에 랜덤 포인트 생성하기

파라미터

라벨	명칭	유형	설명
Input point layer	`INPUT`	[vector: line]	입력 라인 벡터 레이어
Number of points	`POINTS_NUMBER`	[number] 기본값: 1	생성할 포인트의 개수
Minimum distance between points	`MIN_DISTANCE`	[number] 기본값: 0.0	포인트들 사이의 최소 거리
Random points	`OUTPUT`	[vector: point] 기본값: `[Create temporary layer]`	랜덤 포인트 산출물입니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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

라벨	명칭	유형	설명
Input extent	`EXTENT`	[extent]	랜덤 포인트용 맵 범위
Number of points	`POINTS_NUMBER`	[number] 기본값: 1	생성할 포인트의 개수
Minimum distance between points	`MIN_DISTANCE`	[number] 기본값: 0.0	포인트들 사이의 최소 거리
Target CRS	`TARGET_CRS`	[crs] 기본값: *Project CRS*	랜덤 포인트 레이어의 좌표계
Maximum number of search attempts given the minimum distance	`MAX_ATTEMPTS`	[number] 기본값: 200	포인트를 배치하기 위한 최대 시도 횟수
Random points	`OUTPUT`	[vector: point] 기본값: `[Create temporary layer]`	랜덤 포인트 산출물입니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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*

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: polygon]	영역을 정의하는 입력 폴리곤 레이어
Number of points	`POINTS_NUMBER`	[number] 기본값: 1	생성할 포인트의 개수
Minimum distance between points	`MIN_DISTANCE`	[number] 기본값: 0.0	포인트들 사이의 최소 거리
Random points	`OUTPUT`	[vector: point] 기본값: `[Create temporary layer]`	랜덤 포인트 산출물입니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.)

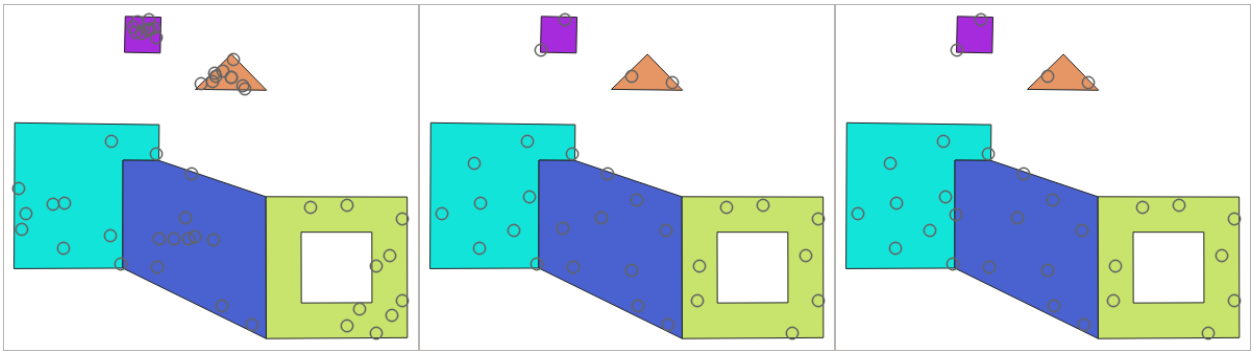


그림 24.36: 폴리곤 피처 당 포인트 10 개, 왼쪽: 최소 거리 = 0, 가운데: 최소 거리 = 1, 오른쪽: 최소 거리 = 1, 전체 수준 최소 거리 = 0

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

라벨	명칭	유형	설명
Input polygon layer	`INPUT`	[vector: line]	입력 폴리곤 벡터 레이어
Number of points for each feature	`POINTS_NUMBER`	[number] 기본값: 1	생성할 포인트의 개수
Minimum distance between points 부가적	`MIN_DISTANCE`	[number] 기본값: 0.0	한 폴리곤 피처 내부의 포인트들 사이의 최소 거리

다음 페이지에 계속

표 24.69 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Global minimum distance between points 부가적	MIN_DISTANCE_GLOBAL	[number] 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) 부가적	MAX_TRIES_PER_POINT	[number] 기본값: 10	포인트 당 최대 시도 횟수입니다. 포인트들 사이의 최소 거리가 (0 보다 크게) 설정된 경우에만 의미가 있습니다.
Random seed 부가적	SEED	[number] 기본값: 설정하지 않음	난수 생성기에 사용할 시드
Include polygon attributes	INCLUDE_POLYGON_ATTRIBUTES	[boolean] 기본값: True	설정한 경우, 포인트가 배치된 폴리곤에서 속성을 가져올 것입니다.
Random points in polygons	OUTPUT	[vector: point] 기본값: [Create temporary layer]	랜덤 포인트 산출물입니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (TEMPORARY_OUTPUT) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Random points in polygons	OUTPUT	[vector: point]	산출 랜덤 포인트 레이어
Number of features with empty or no geometry	FEATURES_WITH_EMPTY_OR_NO_GEOMETRY	[number]	
Total number of points generated	OUTPUT_POINTS	[number]	
Number of missed points	POINTS_MISSED	[number]	최소 거리 제약조건 때문에 생성하지 못 한 포인트의 개수
Number of features with missed points	POLYGONS_WITH_MISSED_POINTS	[number]	비어 있거나 도형이 없는 피처는 포함하지 않음

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 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*

더 보기:

폴리곤에 랜덤 포인트 생성하기

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: polygon]	입력 폴리곤 벡터 레이어
Sampling strategy	`STRATEGY`	[enumeration] 기본값: 0	사용할 샘플링 전략을 지정합니다. 다음 가운데 하나를 선택할 수 있습니다: • 0 —포인트 개수: 각 피처별 포인트의 개수 • 1 —포인트 밀도: 각 피처별 포인트의 밀도
Point count or density	`VALUE`	[number]  기본값: 1.0	선택한 *Sampling strategy* 에 따라 포인트의 개수 또는 밀도를 지정합니다.
Minimum distance between points	`MIN_DISTANCE`	[number] 기본값: 0.0	포인트들 사이의 최소 거리
Random points	`OUTPUT`	[vector: point] 기본값: `[Create temporary layer]`	랜덤 포인트 산출물입니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.)

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

라벨	명칭	유형	설명
Input line layer	INPUT	[vector: line]	입력 라인 벡터 레이어
Number of points for each feature	POINTS_NUMBER	[number] 기본값: 1	생성할 포인트의 개수

다음 페이지에 계속

표 24.72 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Minimum distance between points (per feature) 부가적	MIN_DISTANCE	[number] 기본값: 0.0	한 라인 피처 상의 포인트들 사이의 최소 거리
Global minimum distance between points 부가적	MIN_DISTANCE_GLOBAL	[number] 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) 부가적	MAX_TRIES_PER_POINT	[number] 기본값: 10	포인트 당 최대 시도 횟수입니다. 포인트들 사이의 최소 거리가 (0 보다 크게) 설정된 경우에만 의미가 있습니다.
Random seed 부가적	SEED	[number] 기본값: 설정하지 않음	난수 생성기에 사용할 시드
Include line attributes	INCLUDE_LINE_ATTRIBUTES	[boolean] 기본값: True	설정한 경우, 포인트가 배치된 폴리곤에서 속성을 가져올 것입니다.
Random points on lines	OUTPUT	[vector: point] 기본값: [Create temporary layer]	랜덤 포인트 산출물입니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (TEMPORARY_OUTPUT) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Random points on lines	OUTPUT	[vector: point]	산출 랜덤 포인트 레이어
Number of features with empty or no geometry	FEATURES_WITH_EMPTY_OR_NO_GEOMETRY	[number]	
Number of features with missed points	LINES_WITH_MISSED_POINTS	[number]	비어 있거나 도형이 없는 피처는 포함하지 않음
Total number of points generated	OUTPUT_POINTS	[number]	
Number of missed points	POINTS_MISSED	[number]	최소 거리 제약조건 때문에 생성하지 못 한 포인트의 개수

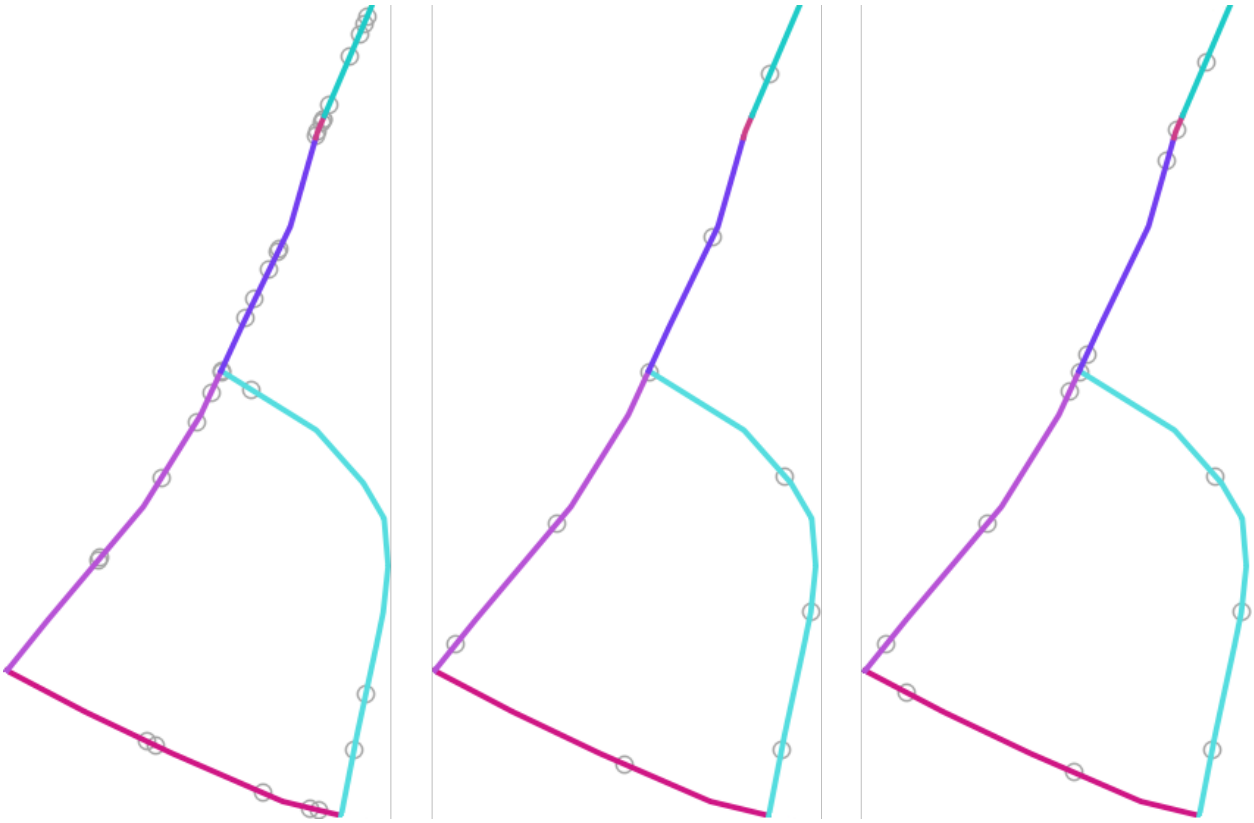


그림 24.37: 라인 피처 당 포인트 5 개, 왼쪽: 최소 거리 = 0, 가운데: 최소 거리 ≠ 0, 오른쪽: 최소 거리 ≠ 0, 전체 수준 최소 거리 = 0

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 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	명칭	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] 기본값: 'VALUE'	래스터 밴드 값을 저장할 필드의 명칭
Vector points	`OUTPUT`	[vector: point] 기본값: `[Create temporary layer]`	픽셀 중심을 담은 산출 포인트 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Vector points	`OUTPUT`	[vector: point]	픽셀 중심을 담은 산출 포인트 레이어

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 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	명칭	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] 기본값: 'VALUE'	래스터 밴드 값을 저장할 필드의 명칭
Vector polygons	`OUTPUT`	[vector: polygon] 기본값: `[Create temporary layer]`	픽셀 범위를 담은 산출 폴리곤 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Vector polygons	`OUTPUT`	[vector: polygon]	픽셀 범위를 담은 산출 폴리곤 레이어

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 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	명칭	유형	설명
Input extent (xmin, xmax, ymin, ymax)	`EXTENT`	[extent]	랜덤 포인트용 맵 범위
Point spacing/count	`SPACING`	[number] 기본값: 100	`Use point spacing` 파라미터의 활성화 여부에 따라 포인트들 사이의 간격 또는 포인트의 개수를 지정합니다.
Initial inset from corner (LH side)	`INSET`	[number] 기본값: 0.0	포인트를 좌상단에 상대적으로 오프셋시킵니다. 이 값은 X 및 Y 축 모두에 사용됩니다.
Apply random offset to point spacing	`RANDOMIZE`	[boolean] 기본값: False	이 파라미터를 활성화하면 포인트에 랜덤한 간격을 적용할 것입니다.
Use point spacing	`IS_SPACING`	[boolean] 기본값: True	이 파라미터를 비활성화하면 포인트 간격을 사용하지 않습니다.
Output layer CRS	`CRS`	[crs] 기본값: *Project CRS*	랜덤 포인트 레이어의 좌표계
Regular points	`OUTPUT`	[vector: point] 기본값: `[Create temporary layer]`	산출 정규 포인트 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

24.1.15 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:

Shapefile 투영체 정의하기, 투영체 찾기, 레이어 재투영하기

파라미터

라벨	명칭	유형	설명
Input layer	INPUT	[vector: any]	잘못된 좌표계를 가진 또는 좌표계가 없는 벡터 레이어
Assigned CRS	CRS	[crs] 기본값: EPSG:4326 - WGS84	벡터 레이어에 할당할 새 좌표계를 선택합니다.
Assigned CRS 부가적	OUTPUT	[same as input] 기본값: [Create temporary layer]	복사본만 담고 있는 산출 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (TEMPORARY_OUTPUT) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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	명칭	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: • 0 —프로젝트 북마크 • 1 —사용자 북마크
Name field	NAME_EXPRESSION	[expression]	생성된 북마크에 명칭을 부여할 필드 또는 표현식
Group field	GROUP_EXPRESSION	[expression]	생성된 북마크를 위한 그룹을 제공할 필드 또는 표현식

산출물

라벨	명칭	유형	설명
Count of bookmarks added	COUNT	[number]	

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 **명칭** 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	명칭	유형	설명
Bookmark source	`SOURCE`	[enumeration] [list] 기본값: [0,1]	북마크의 소스 (들) 을 선택합니다. 다음 가운데 하나 이상을 선택할 수 있습니다: • 0 —프로젝트 북마크 • 1 —사용자 북마크
Output CRS	`CRS`	[crs] 기본값: `EPSG:4326 - WGS84`	산출 레이어의 좌표계
Output	`OUTPUT`	[vector: polygon] 기본값: `[Create temporary layer]`	산출 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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	명칭	유형	설명
Input Layer	`INPUT`	[vector: any]	속성 인덱스를 생성하려는 벡터 레이어를 선택합니다.
Attribute to index	`FIELD`	[tablefield: any]	벡터 레이어의 필드

산출물

라벨	명칭	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 **명칭** 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	명칭	유형	설명
Input Layer	`INPUT`	[vector: any]	입력 벡터 레이어

산출물

라벨	명칭	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 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

Shapefile 투영체 정의하기

기존 Shapefile 포맷 데이터셋의 좌표계 (투영체) 를 지정한 좌표계로 설정합니다. 이 알고리즘은 Shapefile 포맷 데이터셋에서 `prj` 파일이 빠져 있는데, 사용자가 정확한 투영체를 알고 있는 경우 매우 유용합니다.

투영체 할당하기 알고리즘과는 반대로 이 알고리즘은 현재 레이어를 수정하지, 새 레이어를 산출하지는 않을 것입니다.

참고: Shapefile 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	명칭	유형	설명
Input layer	INPUT	[vector: any]	투영체 정보가 빠진 벡터 레이어
CRS	CRS	[crs]	벡터 레이어에 할당할 좌표계를 선택합니다.

산출물

라벨	명칭	유형	설명
	INPUT	[same as input]	정의된 투영체를 가진 입력 벡터 레이어

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 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.

더 보기:

도형 삭제, *NULL* 도형 제거하기, 속성으로 사본 삭제하기

파라미터

라벨	명칭	유형	설명
Input layer	INPUT	[vector: any]	사용자가 삭제하고 싶은 복제 도형이 담긴 레이어
Cleaned	OUTPUT	[same as input] 기본값: [Create temporary layer]	산출 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (TEMPORARY_OUTPUT) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Count of discarded duplicate records	DUPLICATE_COUNT	[number]	삭제한 복제 레코드의 개수
Cleaned	OUTPUT	[same as input]	복제 도형이 없는 산출 레이어
Count of retained records	RETAINED_COUNT	[number]	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 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

속성으로 사본 삭제하기

지정한 필드 (들) 만 고려해서 사본 행을 삭제합니다. 첫 번째로 일치하는 행을 유지하고, 사본은 삭제할 것입니다.

이 복제 레코드를 분석을 위한 개별 산출물로 저장할 수 있는 옵션이 있습니다.

더 보기:

복제 도형 삭제하기

파라미터

라벨	명칭	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] 기본값: [Create temporary layer]	유일 피처를 담고 있는 산출 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (TEMPORARY_OUTPUT) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

다음 페이지에 계속

표 24.75 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Filtered (duplicates) 부가적	DUPLICATES	[same as input] 기본값: [Skip output]	복사본만 담고 있는 산출 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 산출물 건너뛰기 • 임시 레이어 생성 (TEMPORARY_OUTPUT) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Filtered (duplicates) 부가적	DUPLICATES	[same as input] 기본값: [Skip output]	제거한 피처를 담고 있는 벡터 레이어입니다. 따로 지정하지 않는 이상 ([Skip output] 으로 내버려두면) 생성되지 않을 것입니다.
Count of discarded duplicate records	DUPLICATE_COUNT	[number]	삭제한 복제 레코드의 개수
Filtered (no duplicates)	OUTPUT	[same as input]	유일 피처를 담고 있는 벡터 레이어
Count of retained records	RETAINED_COUNT	[number]	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 **명칭** 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.

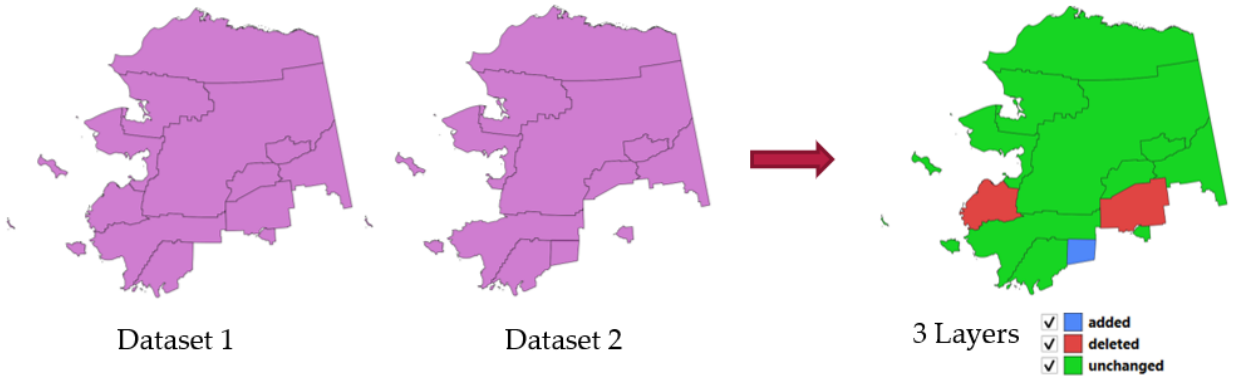


그림 24.38: 데이터셋 변경 사항 탐지의 예

파라미터

라벨	명칭	유형	설명
Original layer	ORIGINAL	[vector: any]	원본 버전으로 간주하는 벡터 레이어
Revised layer	REVISED	[vector: any]	개정된 또는 수정된 벡터 레이어
Attributes to consider for match 부가적	COMPARE_ATTRIBUTES	[tablefield: any] [list]	일치하는지 고려할 속성입니다. 기본적으로 모든 속성을 비교합니다.
Geometry comparison behavior 부가적	MATCH_TYPE	[enumeration] 기본값: 1	비교 규칙을 정의합니다. 다음 옵션 가운데 선택할 수 있습니다: • 0 — 완전 일치 (Exact Match): 도형의 순서 및 꼭짓점 개수도 포함 • 1 — 오차 허용 일치 (Tolerant Match, 위상적 동등성): 도형을 동등한 것으로 간주합니다.
Unchanged features	UNCHANGED	[vector: same as Original layer]	변경되지 않은 피처를 담고 있는 산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (TEMPORARY_OUTPUT) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

다음 페이지에 계속

표 24.77 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Added features	ADDED	[vector: same as Original layer]	추가된 피처를 담고 있는 산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (TEMPORARY_OUTPUT) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.
Deleted features	DELETED	[vector: same as Original layer]	삭제된 피처를 담고 있는 산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (TEMPORARY_OUTPUT) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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	[number]	변경되지 않은 피처의 개수
Count of features added in revised layer	ADDED_COUNT	[number]	개정 레이어에 추가된 피처의 개수
Count of features deleted from original layer	DELETED_COUNT	[number]	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 **명칭** 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.

☑ 제자리 피처 수정 옵션을 사용할 수 있습니다.

더 보기:

복제 도형 삭제하기, *NULL* 도형 제거하기

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	입력 벡터 레이어
Dropped geometries	`OUTPUT`	[table]	산출 무 (□) 도형 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Dropped geometries	`OUTPUT`	[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 **명칭** 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.

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	명칭	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** 명칭으로 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.
Unique identifier field 부가적	INPUT_UID_FIELD	[string]	유일 ID 를 가진 열을 지정합니다.
Geometry field 부가적	INPUT_GEOMETRY_FIELD	[string]	도형 필드를 지정합니다.

다음 페이지에 계속

표 24.79 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Geometry type 부가적	INPUT_GEOMETRY_TYPE	[enumeration] 기본값: 0	산출되는 도형의 유형을 선택합니다. 기본적으로 알고리즘이 자동 탐지할 것입니다. 다음 가운데 하나로 선택할 수 있습니다: • 0 —자동 탐지 (Autodetect) • 1 —도형 없음 • 2 —포인트 • 3 —라인스트링 • 4 —폴리곤 • 5 —멀티포인트 • 6 —멀티라인스트링 • 7 —멀티폴리곤
CRS 부가적	INPUT_GEOMETRY_CRS	[crs]	산출 레이어에 할당할 좌표계
SQL Output	OUTPUT	[vector: any] 기본값: [Create temporary layer]	쿼리가 생성한 산출 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (TEMPORARY_OUTPUT) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
SQL Output	OUTPUT	[vector: any]	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 **명칭** 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	명칭	유형	설명
Input Layer	`INPUT`	[vector: any]	저장할 피처를 선택한 레이어
Selected features	`OUTPUT`	[same as input] 기본값: `[Create temporary layer]`	선택한 피처를 저장할 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

Shapefile 인코딩 정보 추출

이 알고리즘은 Shapefile 에 내장된 속성 인코딩 정보를 추출합니다. 부가적인 `.cpg` 파일로 지정된 인코딩과 `.dbf` LDID 헤더 블록에 있는 모든 인코딩 상세 사항을 모두 고려합니다.

파라미터

라벨	명칭	유형	설명
Input Layer	`INPUT`	[vector: any]	인코딩 정보를 추출할 ESRI Shapefile (`.shp`) 레이어

산출물

라벨	명칭	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 **명칭** 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, *Shapefile* 투영체 정의하기, 레이어 재투영하기

파라미터

라벨	명칭	유형	설명
Input Layer	`INPUT`	[vector: any]	알 수 없는 투영체를 가진 레이어
Target area for layer (xmin, xmax, ymin, ymax)	`TARGET_AREA`	[extent]	레이어가 커버하는 영역. 범위를 지정하기 위한 옵션은 다음과 같습니다: • 캔버스 범위 사용 • 캔버스에서 범위 선택 • 레이어 범위 사용 범위 좌표를 직접 지정할 수도 있습니다. (xmin, xmax, ymin, ymax)
CRS candidates	`OUTPUT`	[table] 기본값: `[Create temporary layer]`	제안 좌표계 (EPSG 코드) 를 담을 테이블 (무 (☐) 도형 레이어) 을 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
CRS candidates	`OUTPUT`	[table]	기준을 만족시키는 모든 좌표계 (EPSG 코드) 를 담은 테이블

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 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.

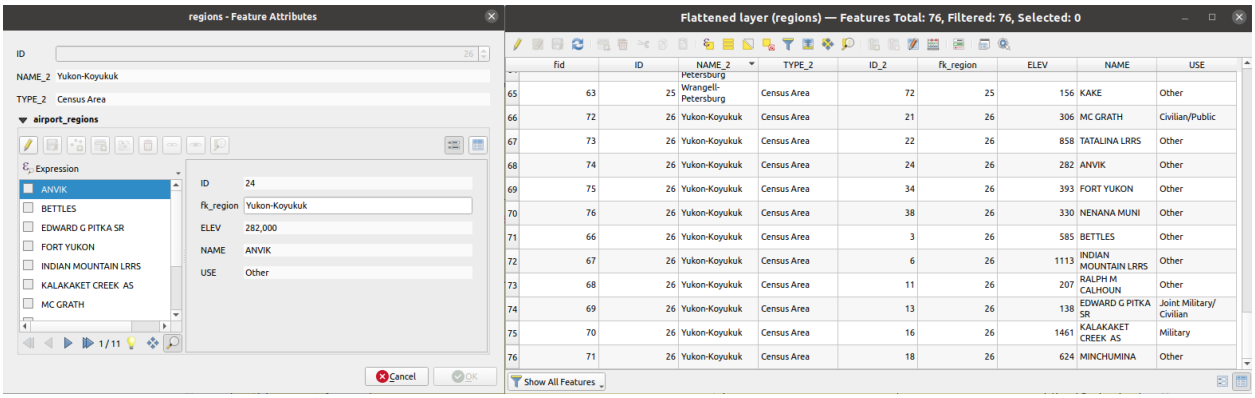


그림 24.39: 관련 자식을 가진 지역의 양식 (왼쪽) - 각 관련 자식마다 결합된 속성을 가진 복제 지역 피처 (오른쪽)

파라미터

라벨	명칭	유형	설명
Input Layer	`INPUT`	[vector: any]	비정규화 (de-normalize) 해야 할 관계를 가진 레이어
Flattened Layer 부가적	`OUTPUT`	[same as input] 기본값: `[Save to temporary file]`	산출 (평탄화) 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 • Geopackage 로 저장 • 데이터베이스 테이블에 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

더 보기:

최근접으로 속성 결합하기, 위치로 속성 결합하기

파라미터

라벨	명칭	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
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 부가적	FIELDS_TO_COPY	[tablefield: any] [list]	사용자가 추가하고자 하는 특정 필드를 선택합니다. 기본적으로 모든 필드를 추가합니다.
Join type	METHOD	[enumeration] 기본값: 1	최종 결합 레이어의 유형을 지정합니다. 다음 가운데 하나를 선택할 수 있습니다: • 0 —일치하는 각 피처마다 개별 피처를 생성합니다. (1 대다) • 1 —처음으로 일치하는 피처의 속성만 가져옵니다. (1 대 1)
Discard records which could not be joined	DISCARD_NONMATCHING	[boolean] 기본값: True	결합할 수 없는 피처를 유지하지 않으려면 이 파라미터를 활성화하십시오.
Joined field prefix 부가적	PREFIX	[string]	결합된 필드명에 접두어를 추가합니다. 결합된 필드를 더 쉽게 식별하고 명칭의 충돌을 방지하기 위해서입니다.
Joined layer	OUTPUT	[same as input] 기본값: [Create temporary layer]	결합된 속성을 저장할 산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (TEMPORARY_OUTPUT) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.
Unjoinable features from first layer	NON_MATCHING	[same as input] 기본값: [Skip output]	첫 번째 레이어에서 결합할 수 없는 피처를 저장할 산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 산출물 건너뛰기 • 임시 레이어 생성 (TEMPORARY_OUTPUT) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Number of joined features from input table	JOINED_COUNT	[number]	
Unjoinable features from first layer 부가적	NON_MATCHING	[same as input]	일치하지 않는 피처를 담은 벡터 레이어
Joined layer	OUTPUT	[same as input]	결합으로 추가된 속성을 가진 산출 벡터 레이어
Number of unjoinable features from input table 부가적	UNJOINABLE_COUNT	[number]	

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 **명칭** 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*

더 보기:

최근접으로 속성 결합하기, 필드값으로 속성 결합하기, 위치로 속성 결합하기 *(요약)*

파라미터

라벨	명칭	유형	설명
Input Layer	INPUT	[vector: any]	입력 벡터 레이어. 산출 레이어는 이 레이어의 피처와 두 번째 레이어에 있는 일치하는 피처의 속성으로 이루어질 것입니다.
Join layer	JOIN	[vector: any]	소스 레이어 속성 테이블에 이 벡터 레이어의 속성을 **추가** 할 것입니다.

다음 페이지에 계속

표 24.84 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Geometric predicate	PREDICATE	[enumeration] [list] 기본값: [0]	기하하적 기준을 선택합니다. 다음 가운데 하나 이상을 선택할 수 있습니다: • 0 —교차 (intersect) • 1 —포함 (contain) • 2 —동등 (equal) • 3 —접촉 (touch) • 4 —중첩 (overlap) • 5 —내부 (within) • 6 —공간 교차 (cross)
Fields to add (leave empty to use all fields) 부가적	JOIN_FIELDS	[tablefield: any] [list]	사용자가 추가하고자 하는 특정 필드를 선택합니다. 기본적으로 모든 필드를 추가합니다.
Join type	METHOD	[enumeration]	최종 결합 레이어의 유형을 지정합니다. 다음 가운데 하나를 선택할 수 있습니다: • 0 —일치하는 각 피처마다 개별 피처를 생성합니다. (1 대다) • 1 —처음으로 일치하는 피처의 속성만 가져옵니다. (1 대 1) • 2 —최대 중첩 영역을 가진 피처의 속성만 가져오기 (1 대 1)
Discard records which could not be joined	DISCARD_NONMATCHING	[boolean] 기본값: False	산출물에서 결합시킬 수 없는 입력 레이어 레코드를 제거합니다.
Joined field prefix 부가적	PREFIX	[string]	결합된 필드명에 접두어를 추가합니다. 결합된 필드를 더 쉽게 식별하고 명칭의 충돌을 방지하기 위해서입니다.
Joined layer	OUTPUT	[same as input] 기본값: [Create temporary layer]	결합된 속성을 저장할 산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (TEMPORARY_OUTPUT) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.
Unjoinable features from first layer	NON_MATCHING	[same as input] 기본값: [Skip output]	첫 번째 레이어에서 결합할 수 없는 피처를 저장할 산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 산출물 건너뛰기 • 임시 레이어 생성 (TEMPORARY_OUTPUT) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Number of joined features from input table	`JOINED_COUNT`	[number]	
Unjoinable features from first layer 부가적	`NON_MATCHING`	[same as input]	일치하지 않는 피처를 담은 벡터 레이어
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 **명칭** 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

파라미터

라벨	명칭	유형	설명
Input Layer	`INPUT`	[vector: any]	입력 벡터 레이어. 산출 레이어는 이 레이어의 피처와 두 번째 레이어에 있는 일치하는 피처의 속성으로 이루어질 것입니다.
Join layer	`JOIN`	[vector: any]	소스 레이어 속성 테이블에 이 벡터 레이어의 속성을 **추가** 할 것입니다.

다음 페이지에 계속

표 24.85 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Geometric predicate	PREDICATE	[enumeration] [list] 기본값: [0]	기하학적 기준을 선택합니다. 다음 가운데 하나 이상을 선택할 수 있습니다: • 0 —교차 (intersect) • 1 —포함 (contain) • 2 —동등 (equal) • 3 —접촉 (touch) • 4 —중첩 (overlap) • 5 —내부 (within) • 6 —공간 교차 (cross)
Fields to summarize (leave empty to use all fields) 부가적	JOIN_FIELDS	[tablefield: any] [list]	사용자가 추가하고 요약하고자 하는 특정 필드를 선택합니다. 기본적으로 모든 필드를 추가합니다.
Summaries to calculate (leave empty to use all fields) 부가적	SUMMARIES	[enumeration] [list] 기본값: []	각 피처마다 각 필드에 추가하고자 하는 요약 유형을 선택합니다. 다음 가운데 하나 이상을 선택할 수 있습니다: • 0 —개수 • 1 —유일값 • 2 —최소값 • 3 —최대값 • 4 —범위 (range) • 5 —합계 • 6 —평균 • 7 —중간값 • 8 —표준 편차 • 9 —희소값 (minority) • 10 —최빈값 (majority) • 11 —제 1 사분위수 • 12 —제 3 사분위수 • 13 —사분위수의 범위 • 14 —비어 있는 값 • 15 —존재하는 값 • 16 —최단 거리 • 17 —최장 거리 • 18 —평균 거리
Discard records which could not be joined	DISCARD_NONMATCHING	[boolean] 기본값: False	산출물에서 결합시킬 수 없는 입력 레이어 레코드를 제거합니다.
Joined layer	OUTPUT	[same as input] 기본값: [Create temporary layer]	결합된 속성을 저장할 산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (TEMPORARY_OUTPUT) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Joined layer	OUTPUT	[same as input]	결합으로부터 나온 요약된 속성을 가진 산출 벡터 레이어

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 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.

더 보기:

최근접 이웃 분석, 필드값으로 속성 결합하기, 위치로 속성 결합하기, 거리 매트릭스

파라미터

라벨	명칭	유형	설명
Input layer	INPUT	[vector: any]	입력 레이어
Input layer 2	INPUT_2	[vector: any]	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	DISCARD_NONMATCHING	[boolean] 기본값: False	산출물에서 결합시킬 수 없는 입력 레이어 레코드를 제거합니다.
Joined field prefix	PREFIX	[string]	결합된 필드의 접두어
Maximum nearest neighbors	NEIGHBORS	[number] 기본값: 1	최근접 이웃의 최대 개수
Maximum distance	MAX_DISTANCE	[number]	최대 검색 거리

다음 페이지에 계속

표 24.86 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Joined layer	OUTPUT	[same as input] 기본값: [Create temporary layer]	결합된 피처를 담고 있는 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (TEMPORARY_OUTPUT) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.
Unjoinable features from first layer	NON_MATCHING	[same as input] 기본값: [Skip output]	결합할 수 없는 피처를 담고 있는 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 산출물 건너뛰기 • 임시 레이어 생성 (TEMPORARY_OUTPUT) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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	[number]	입력 테이블에서 나온 결합된 피처의 개수
Number of unjoinable features from input table	UNJOINABLE_COUNT	[number]	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 **명칭** 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.

산출되는 레이어의 속성 테이블은 모든 입력 레이어의 필드들을 담게 될 것입니다. 동일한 이름이지만 서로 다른 유형인 필드들이 있을 경우 필드를 자동적으로 문자열 유형 필드로 변환해서 내보낼 것입니다. 원본 레이어명 및 그 소스를 저장한 새 필드도 추가됩니다.

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*

더 보기:

벡터 레이어 분할하기

파라미터

라벨	명칭	유형	설명
Input Layers	`LAYERS`	[vector: any] [list]	단일 레이어로 병합할 레이어들. 동일한 도형 유형 레이어이어야 합니다.
Destination CRS 부가적	`CRS`	[crs]	산출 레이어의 좌표계를 선택합니다. 좌표계를 지정하지 않은 경우, 첫 번째 입력 레이어의 좌표계를 사용할 것입니다.
Merged	`OUTPUT`	[same as input] 기본값: `[Create temporary layer]`	산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

파라미터

라벨	명칭	유형	설명
Input Layer	INPUT	[vector: any]	정렬할 입력 벡터 레이어
Expression	EXPRESSION	[expression]	정렬 작업에 사용할 표현식
Sort ascending	ASCENDING	[boolean] 기본값: True	이 파라미터를 활성화하면 벡터 레이어를 작은 값에서 큰 값으로 정렬할 것입니다.
Sort nulls first	NULLS_FIRST	[boolean] 기본값: False	이 파라미터를 활성화하면 NULL 값을 처음에 배치합니다.
Ordered	OUTPUT	[same as input] 기본값: [Create temporary layer]	산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (TEMPORARY_OUTPUT) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

Shapefile 수리하기

SHX 파일을 (재) 생성해서 망가진 Shapefile 데이터셋을 수리합니다.

파라미터

라벨	명칭	유형	설명
Input Shapefile	INPUT	[file]	없거나 깨진 SHX 파일을 가진 ESRI Shapefile 데이터셋을 가리키는 전체 경로

산출물

라벨	명칭	유형	설명
Repaired layer	OUTPUT	[vector: any]	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 **명칭** 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.

☑ 제자리 피처 수정 옵션을 사용할 수 있습니다.

더 보기:

투영체 할당하기, *Shapefile* 투영체 정의하기, 투영체 찾기

파라미터

라벨	명칭	유형	설명
Input Layer	`INPUT`	[vector: any]	재투영할 입력 벡터 레이어
Target CRS	`TARGET_CRS`	[crs] 기본값: `EPSG:4326 - WGS84`	대상 좌표계
Coordinate Operation 부가적	`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. 기준 *(datum)* 변환 을 읽어보십시오.
Reprojected	`OUTPUT`	[same as input] 기본값: `[Create temporary layer]`	산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

레이어를 지원하는 데이터셋 포맷에 대해, 부가적인 레이어명 파라미터를 이용해서 사용자 지정 문자열을 지정할 수 있습니다. GDAL 이 정의하는 데이터셋과 레이어 옵션들도 선택적으로 지정할 수 있습니다. 이 알고리즘에 대한 더 자세한 정보를 원한다면, 포맷 관련 온라인 GDAL 문서 를 읽어보세요.

파라미터

기본 파라미터

라벨	명칭	유형	설명
Vector features	INPUT	[vector: any]	입력 벡터 레이어
Saved features	OUTPUT	[same as input] 기본값: [Save to temporary file]	피처를 저장할 파일을 지정합니다. 다음 가운데 하나를 선택할 수 있습니다: • 임시 파일로 저장 • 파일로 저장

고급 파라미터

라벨	명칭	유형	설명
Layer name 부가적	LAYER_NAME	[string]	산출 레이어에 사용할 이름
GDAL dataset options 부가적	DATASOURCE_OPTIONS	[string]	산출 포맷의 GDAL 데이터셋 생성 옵션입니다. 개별 옵션을 쌍반점으로 구분하십시오.
GDAL layer options 부가적	LAYER_OPTIONS	[string]	산출 포맷의 GDAL 레이어 생성 옵션입니다. 개별 옵션을 쌍반점으로 구분하십시오.

산출물

라벨	명칭	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 **명칭** 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	명칭	유형	설명
Saved features	INPUT	[vector: any]	인코딩을 설정할 벡터 레이어
Encoding	ENCODING	[string]	현재 QGIS 세션에서 레이어에 할당할 텍스트 인코딩

산출물

라벨	명칭	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 **명칭** 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.

파라미터

라벨	명칭	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] 기본값: False	
Split	`OUTPUT`	[same as input] 기본값: `Create temporary layer`	산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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

라벨	명칭	유형	설명
Input Layer	`INPUT`	[vector: any]	입력 벡터 레이어
Unique ID field	`FIELD`	[tablefield: any]	분할 작업에 사용할 필드
Output directory	`OUTPUT`	[folder] 기본값: `[Save to temporary folder]`	산출 레이어를 저장할 디렉터리를 지정합니다. 다음 가운데 하나로 선택할 수 있습니다: • 임시 디렉터리에 저장 • 디렉터리에 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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	명칭	유형	설명
Input Layer	`INPUT`	[vector: any]	입력 벡터 레이어

산출물

라벨	명칭	유형	설명
Truncated layer	`OUTPUT`	[folder]	절단된 (비어 있는) 레이어

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 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

24.1.16 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`

기본 메뉴: *Vector → Geometry Tools*

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	입력 벡터 레이어
Calculate using	`CALC_METHOD`	[enumeration] 기본값: 0	기하학적 속성에 사용할 계산용 파라미터입니다. 다음 가운데 하나를 선택할 수 있습니다: • 0 — 레이어 좌표계 • 1 — 프로젝트 좌표계 • 2 — 타원체
Added geom info	`OUTPUT`	[same as input] 기본값: `[Create temporary layer]`	산출 레이어 (도형을 가진 입력 레이어의 복사본) 를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Added geom info	`OUTPUT`	[same as input]	입력 벡터 레이어에 도형 필드를 추가한 복사본

파이썬 코드

Algorithm ID: `qgis: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 **명칭** 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.

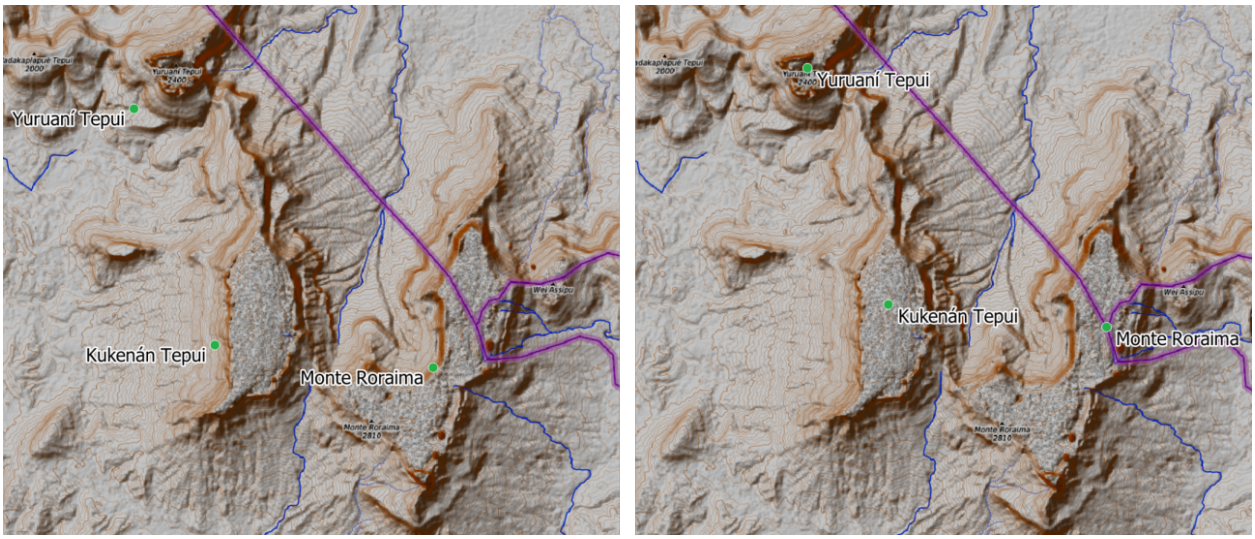


그림 24.40: 벡터 포인트 레이어 (녹색 점) 의 아핀 변형 (좌표 이동) 전 (왼쪽) 과 후 (오른쪽).

더 보기:

이동시키기 *(Translate)*

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	입력 벡터 레이어
Translation (x-axis)	`DELTA_X`	[number] 기본값: 0	X 축에 적용할 변위 (displacement)
Translation (y-axis)	`DELTA_Y`	[number] 기본값: 0	Y 축에 적용할 변위 (displacement)

다음 페이지에 계속

표 24.91 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Translation (z-axis)	DELTA_Z	[number] 기본값: 0	Z 축에 적용할 변위 (displacement)
Translation (m-values)	DELTA_M	[number] 기본값: 0	M 값에 적용할 오프셋
Scale factor (x-axis)	SCALE_X	[number] 기본값: 1	X 축에 적용할 척도값 (확장 또는 수축)
Scale factor (y-axis)	SCALE_Y	[number] 기본값: 1	Y 축에 적용할 척도값 (확장 또는 수축)
Scale factor (z-axis)	SCALE_Z	[number] 기본값: 1	Z 축에 적용할 척도값 (확장 또는 수축)
Scale factor (m-values)	SCALE_M	[number] 기본값: 1	M 값에 적용할 척도값 (확장 또는 수축)
Rotation around z-axis (degrees counter-clockwise)	ROTATION_Z	[number] 기본값: 0	기울기의 도 단위 각도
Transformed	OUTPUT	[same as input] 기본값: [Create temporary layer]	산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (TEMPORARY_OUTPUT) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

더 보기:

도형 모으기, 융해

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	입력 벡터 레이어
Group by expression	`GROUP_BY`	[tablefield: any] 기본값: 'NULL'	그룹화 필드를 선택합니다. *NULL* 값을 설정하면 모든 피처를 하나의 그룹으로 묶을 것입니다.

다음 페이지에 계속

표 24.93 – 이전 페이지에서 계속

라벨	명칭	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: •  버튼을 클릭해서 새 필드를 추가할 수 있습니다. •  버튼을 클릭해서 선택한 필드를 삭제할 수 있습니다. •  및  버튼을 클릭해서 필드 순서를 변경할 수 있습니다. •  버튼을 클릭해서 (입력 레이어 필드의) 기본값으로 리셋할 수 있습니다. 사용자가 정보를 검색하고 싶은 각 필드마다 다음을 정의해줘야 합니다: **Input expression [expression] (input)** 입력 레이어에서 나온 필드 또는 표현식 **Aggregate function [enumeration] (aggregate)** 집계값을 반환하기 위해 입력 표현식에 사용할함수 기본값: *concatenate* (문자열 데이터 유형), *sum* (숫자 데이터 유형) **Delimiter [string] (delimiter)** 예를 들어 concatenate 함수로 집계된 값들을 구분하기 위한 텍스트 문자열 기본값: , **Output field name [string] (name)** 산출 레이어의 집계된 필드의 명칭입니다. 기본적으로 입력 필드명을 그대로 유지합니다. **Type [enumeration] (type)** 산출 필드의 데이터 유형입니다. 다음 가운데 하나로 지정할 수 있습니다: • 1 —불 (Boolean) • 2 —정수형 (Integer) • 4 —부호 있는 64 비트 정수형 (Integer64) • 6 —배정밀도 64 비트 부동소수점 실수형 (Double) • 10 —문자열 • 14 —날짜 • 16 —날짜 & 시간 **Length [number] (length)** 산출 필드의 길이 **Precision [number] (precision)** 산출 필드의 정밀도

다음 페이지에 계속

표 24.93 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Load fields from layer	GUI 전용	[vector: any]	또다른 레이어에서 필드를 불러와 집계에 사용할 수 있습니다.
Aggregated	`OUTPUT`	[same as input] 기본값: `[Create temporary layer]`	산출 (집계) 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

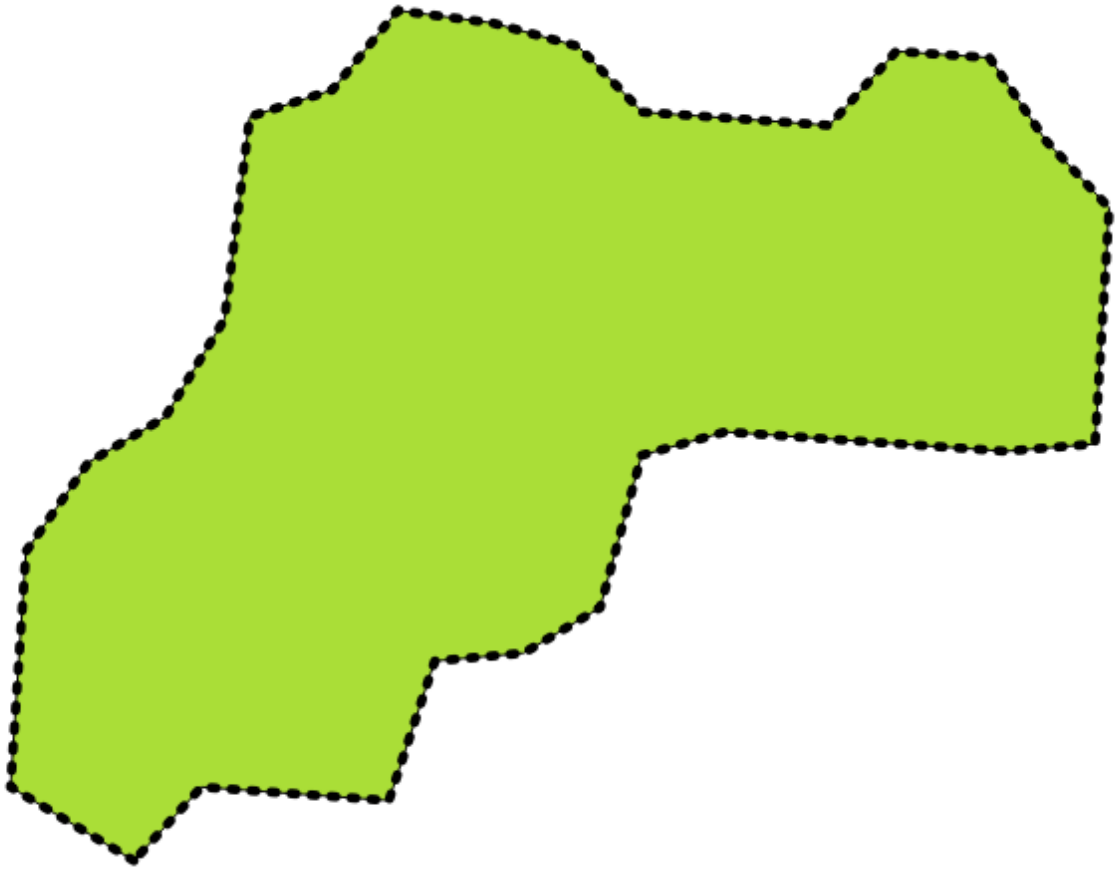


그림 24.41: 소스 폴리곤 레이어의 경계 (검은 점선)

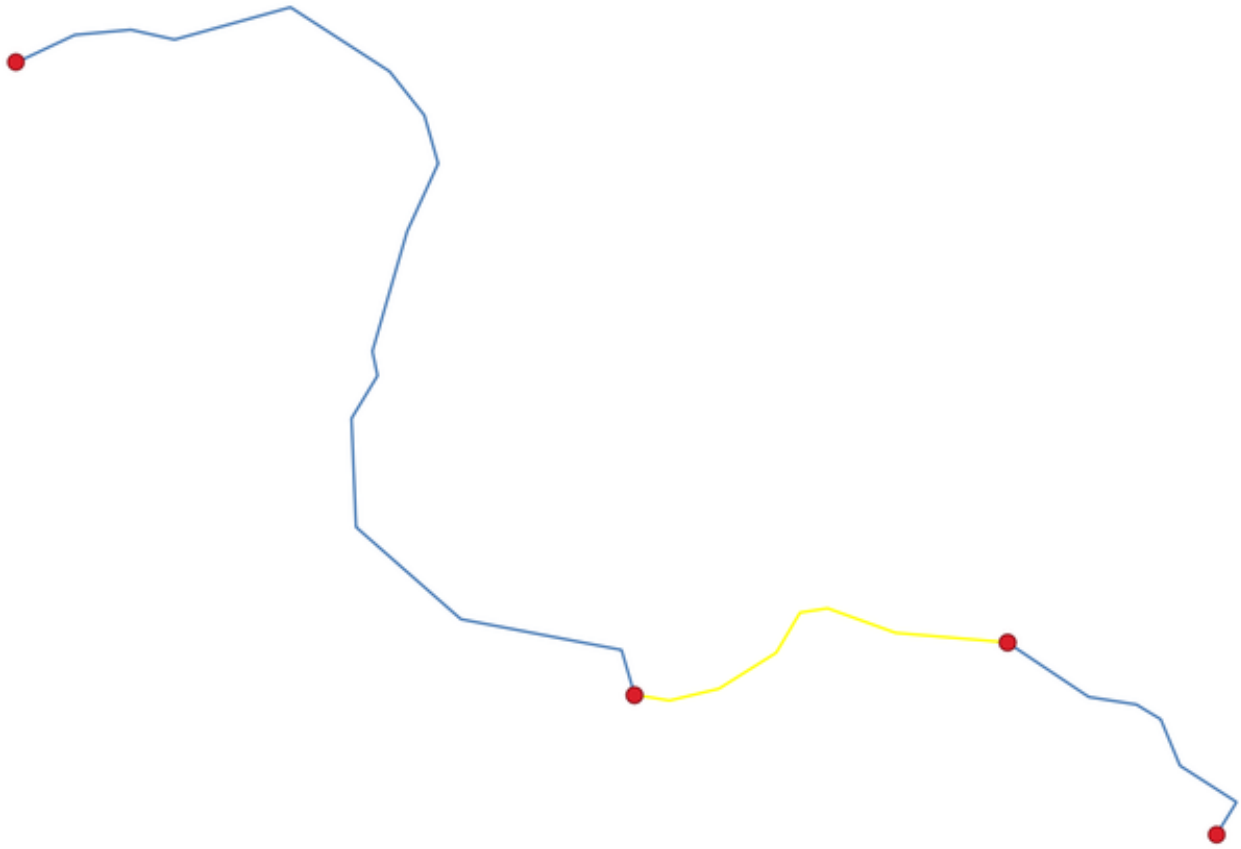


그림 24.42: 라인 레이어의 경계 (빨간 포인트), 노란색은 선택한 피처

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: line, polygon]	입력 라인 또는 폴리곤 벡터 레이어
Boundary	`OUTPUT`	[vector: point, line] 기본값: `[Create temporary layer]`	산출 (경계) 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

☑ 제자리 피처 수정 옵션을 사용할 수 있습니다.

더 보기:

최소 경계 도형

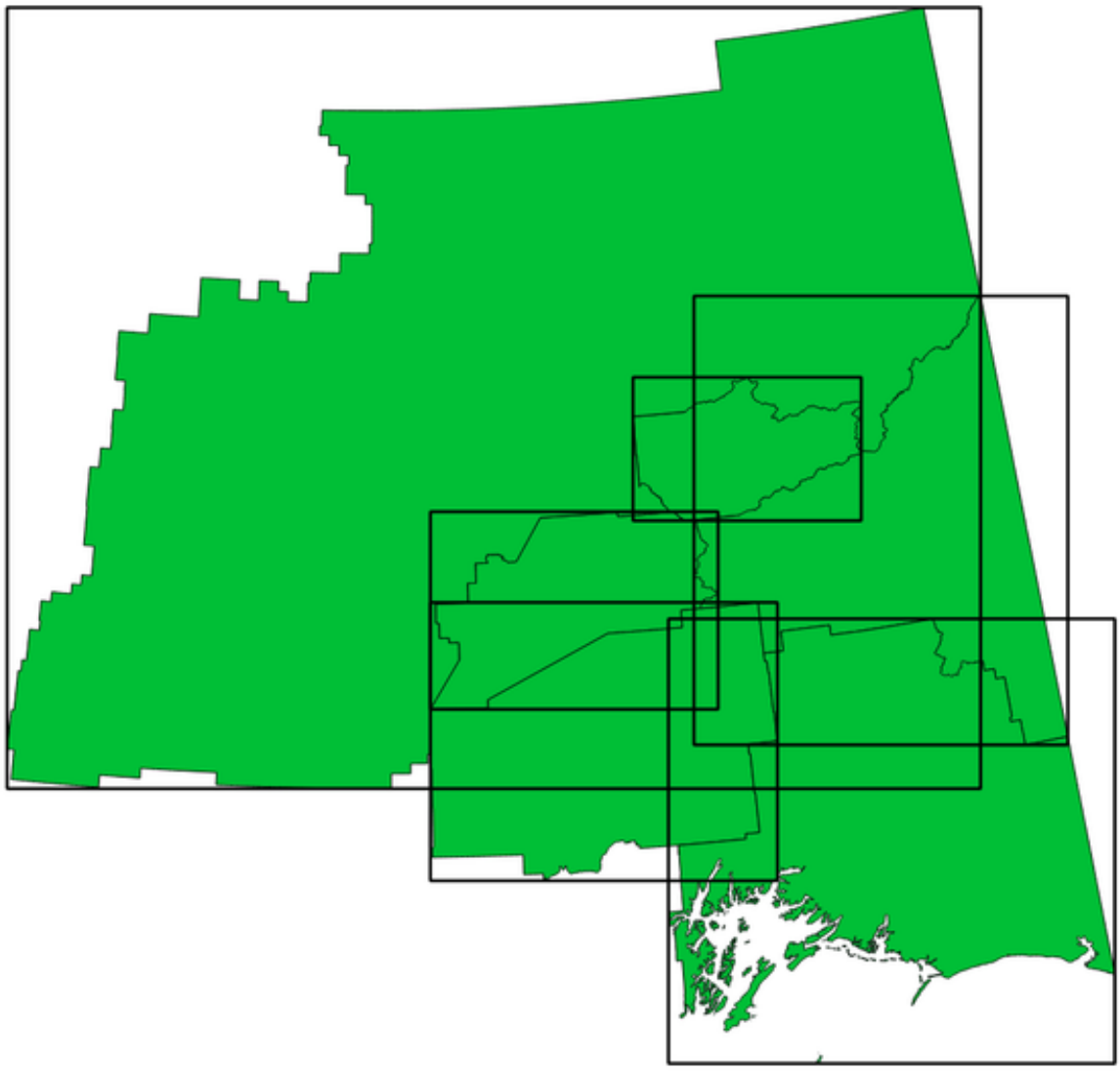


그림 24.43: 각 폴리곤 피처의 경계 상자를 표현하는 검은 선

파라미터

라벨	명칭	유형	설명
Input layer	INPUT	[vector: line, polygon]	입력 라인 또는 폴리곤 벡터 레이어
Bounds	OUTPUT	[vector: polygon] 기본값: [Create temporary layer]	산출 (경계 상자) 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (TEMPORARY_OUTPUT) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Bounds	OUTPUT	[vector: polygon]	입력 레이어의 경계 상자

파이썬 코드

알고리즘 ID: native:boundingboxes

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

버퍼

입력 레이어의 모든 피처에 대한 버퍼 영역을 고정 거리를 사용해서 계산합니다.

입력 폴리곤 레이어의 경우 음의 거리를 사용할 수 있습니다. 이 경우 버퍼는 더 작은 폴리곤으로 (역행 setback) 산출될 것입니다.

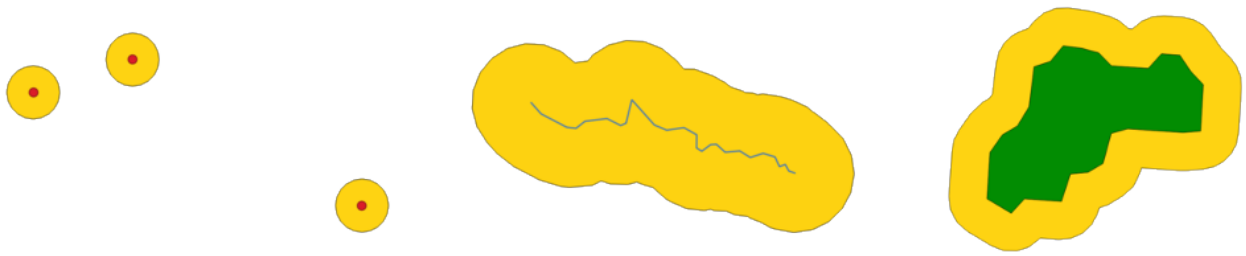


그림 24.44: 포인트, 라인 및 폴리곤의 (노란색) 버퍼

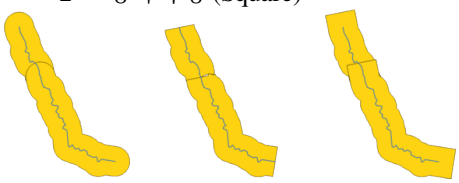
제자리 피처 수정 옵션을 사용할 수 있습니다.

기본 메뉴: *Vector → Geoprocessing Tools*

더 보기:

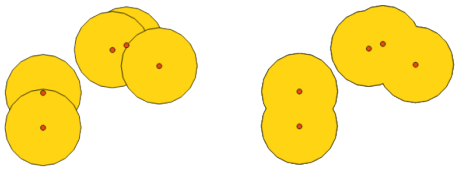
변동 거리 버퍼, 다중 고리 버퍼 *(상수 거리)*, 변동 너비 버퍼 *(M 값으로)*

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	입력 벡터 레이어
Distance	`DISTANCE`	[number] 기본값: 10.0	(각 피처의 경계로부터의) 버퍼 거리입니다. 오른쪽에 있는 데이터 정의 버튼을 클릭해서 반경을 계산할 필드를 선택할 수 있습니다. 이렇게 하면 각 피처별로 서로 다른 반경을 지정할 수 있습니다. (변동 거리 버퍼 참조)
Segments	`SEGMENTS`	[number] 기본값: 5	둥글린 (rounded) 오프셋을 생성하는 경우 사분원에 가깝게 만드는 데 사용할 라인 선분의 개수를 제어합니다.
End cap style	`END_CAP_STYLE`	[enumeration] 기본값: 0	버퍼에서 라인의 끝을 어떻게 처리할지 제어합니다. 다음 가운데 하나로 지정할 수 있습니다: • 0 —둥글림 (Round) • 1 —편평 (Flat) • 2 —정사각형 (Square)  그림 24.45: 둥글림, 편평 및 정사각형 캡 스타일
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: • 0 —둥글림 (Round) • 1 —마이터 (Miter) • 2 —베벨 (Bevel)
Miter limit	`MITER_LIMIT`	[number] 기본값: 2.0	마이터 결합 스타일에만 적용할 수 있으며, 마이터 결합 생성시 사용할 오프셋 곡선으로부터의 최장 거리를 제어합니다. (최소값은 1 입니다.)

다음 페이지에 계속

표 24.94 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Dissolve result	DISSOLVE	[boolean] 기본값: False	최종 버퍼를 융해 (dissolve) 합니다. True 로 설정하면 (활성화하면) 중첩하는 버퍼들을 새 피처로 융해 (결합) 할 것입니다.  그림 24.46: 표준 버퍼와 융해된 버퍼
Buffered	OUTPUT	[vector: polygon] 기본값: [Create temporary layer]	산출 (버퍼) 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (TEMPORARY_OUTPUT) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

제자리 피처 수정 옵션을 사용할 수 있습니다.

기본 메뉴: *Vector → Geometry Tools*

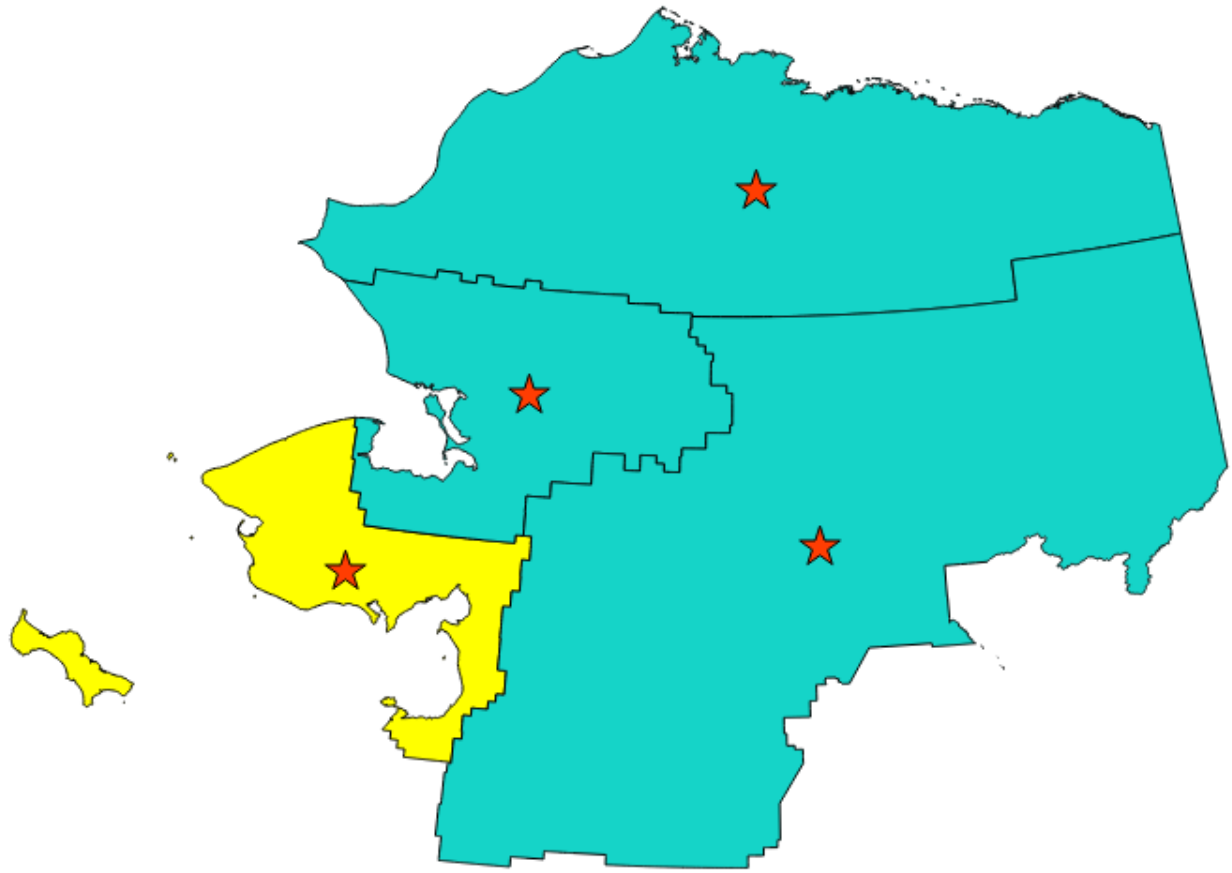


그림 24.47: 입력 레이어의 피처의 중심을 표현하는 빨간색 별

더 보기:

표면에 포인트 생성하기

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	입력 벡터 레이어
Create centroid for each part	`ALL_PARTS`	[boolean] 기본값: False	`True` 로 설정하면 (활성화하면) 도형의 각 부분별로 중심을 생성할 것입니다.
Centroids	`OUTPUT`	[vector: point] 기본값: `[Create temporary layer]`	산출 (중심) 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

유효성 점검하기

벡터 레이어의 도형에 대해 유효성 (validity) 점검을 수행합니다.

도형을 세 그룹 (유효함, 유효하지 않음 및 오류) 으로 분류해서 각 그룹별 피처를 담은 벡터 레이어를 생성합니다:

- **유효한 산출물** 레이어는 (위상 오류가 없는) 유효한 피처만 담고 있습니다.
- **유효하지 않은 산출물** 레이어는 알고리즘이 발견한 유효하지 않은 피처를 모두 담고 있습니다.
- **오류 산출물** 레이어는 유효하지 않은 피처를 발견한 위치를 가리키는 포인트 레이어입니다.

생성된 레이어들의 속성 테이블은 몇몇 (**오류** 레이어의 경우"message", **유효하지 않은** 레이어의 경우"FID"및"_errors", 그리고 **유효한** 레이어의 경우"FID"만) 추가 정보를 담을 것입니다:

생성된 각 레이어의 속성 테이블은 몇몇 (발견된 오류 개수 및 오류 유형) 추가 정보를 담을 것입니다:

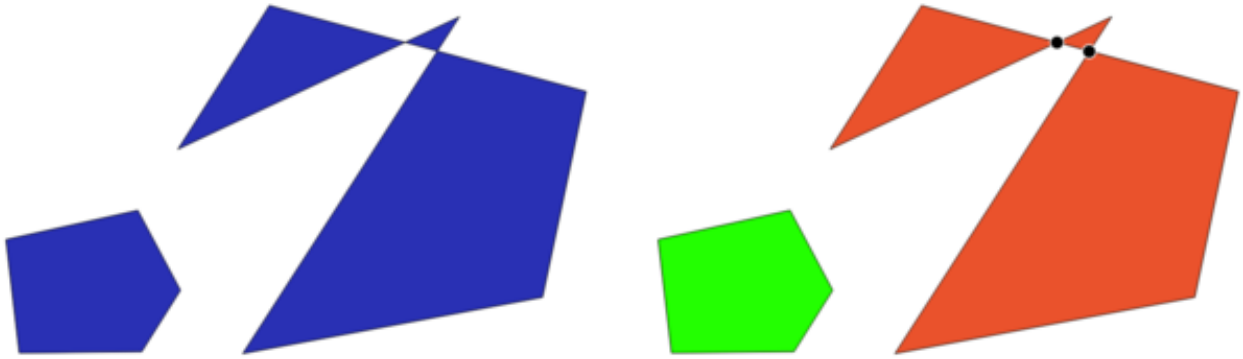


그림 24.48: 왼쪽: 입력 레이어, 오른쪽: 유효한 레이어 (녹색), 유효하지 않은 레이어 (주황색)

기본 메뉴: *Vector → Geometry Tools*

더 보기:

도형 고치기 and the core plugin 도형 점검기 플러그인

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT_LAYER`	[vector: any]	입력 벡터 레이어
Method	`METHOD`	[enumeration] 기본값: 2	유효성을 검증하기 위해 사용하는 메소드입니다. 다음 옵션 가운데 선택할 수 있습니다: • 0: 디지타이즈 작업 설정에서 선택한 메소드 • 1: QGIS • 2: GEOS
Ignore ring self intersection	`IGNORE_RING_SELF_INTERSECTION`	[boolean] 기본값: False	유효성 점검 작업시 자체 교차 (self-intersect) 하는 고리를 무시합니다.
Valid output	`VALID_OUTPUT`	[same as input] 기본값: `[Create temporary layer]`	소스 레이어의 유효한 피처의 복사본을 담고 있는 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 산출물 건너뛰기 • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

다음 페이지에 계속

표 24.95 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Invalid output	INVALID_OUTPUT	[same as input] 기본값: [Create temporary layer]	발견한 오류 (들) 의 요약 목록을 담은 _errors 필드와 함께 소스 레이어의 유효하지 않은 피처의 복사본을 담고 있는 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 산출물 건너뛰기 • 임시 레이어 생성 (TEMPORARY_OUTPUT) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.
Error output	ERROR_OUTPUT	[vector: point] 기본값: [Create temporary layer]	발견한 오류 (들) 를 서술하는 message 필드와 함께 감지된 유효성 문제점의 정확한 위치를 담고 있는 포인트 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 산출물 건너뛰기 • 임시 레이어 생성 (TEMPORARY_OUTPUT) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Count of errors	ERROR_COUNT	[number]	오류를 초래한 도형의 개수
Error output	ERROR_OUTPUT	[vector: point]	발견한 오류 (들) 를 서술하는 message 필드와 함께 감지된 유효성 문제점의 정확한 위치를 담고 있는 포인트 레이어
Count of invalid features	INVALID_COUNT	[number]	유효하지 않은 도형의 개수
Invalid output	INVALID_OUTPUT	[same as input]	발견한 오류 (들) 의 요약 목록을 담은 _errors 필드와 함께 소스 레이어의 유효하지 않은 피처의 복사본을 담고 있는 벡터 레이어
Count of valid features	VALID_COUNT	[number]	유효한 도형의 개수
Valid output	VALID_OUTPUT	[same as input]	소스 레이어의 유효한 피처의 복사본을 담고 있는 벡터 레이어

파이썬 코드

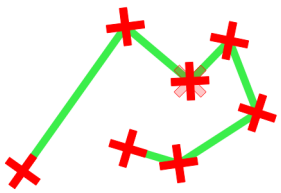
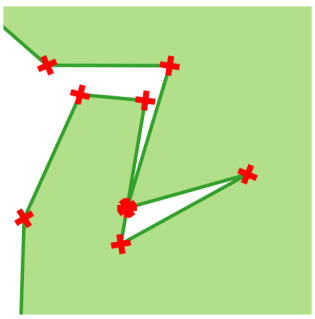
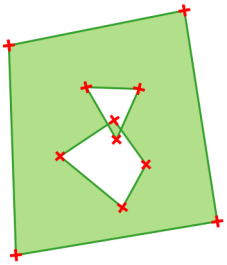
Algorithm ID: `qgis:checkvalidity`

```
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 **명칭** 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 24.97: GEOS 메소드를 사용하는 경우 다음과 같은 오류 메시지가 발생할 수 있습니다:

오류 메시지	설명	예시
Repeated point	이 오류는 어떤 꼭짓점이 반복되는 경우 발생합니다.	
Ring self-intersection	이 오류는 도형이 스스로 인접해서 고리를 생성하는 경우 발생합니다.	
Self-intersection	이 오류는 도형이 스스로 교차하는 경우 발생합니다.	
Topology validation error		
Hole lies outside shell		
Holes are nested		
Interior is disconnected		

다음 페이지에 계속

표 24.97 – 이전 페이지에서 계속

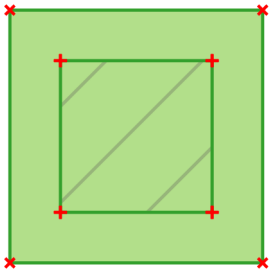
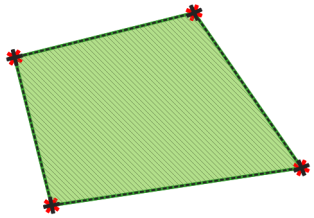
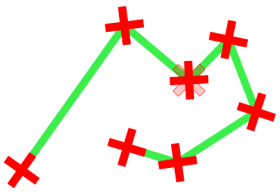
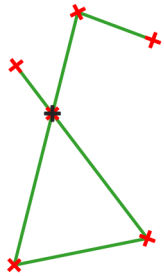
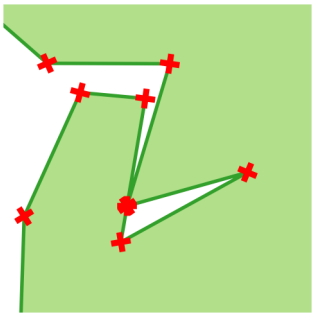
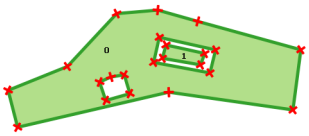
오류 메시지	설명	예시
Nested shells	이 오류는 폴리곤 도형이 또다른 폴리곤 도형 내부에 겹치는 경우 발생합니다.	
Duplicate rings	이 오류는 폴리곤 도형의 (외곽 또는 내곽) 고리 2 개가 동일한 경우 발생합니다.	
Too few points in geometry component		
Invalid coordinate	포인트 도형의 경우, 이 오류는 도형이 적절한 좌표쌍을 가지고 있지 않은 경우 발생합니다. 좌표쌍이 위도 값 및 경도 값을 이 순서대로 담고 있지 않습니다.	
Ring is not closed		

표 24.98: QGIS 메소드를 사용하는 경우 다음과 같은 오류 메시지가 발생할 수 있습니다:

오류 메시지	설명	예시
Segment %1 of ring %2 of polygon %3 intersects segment %4 of ring %5 of polygon %6 at %7		
Ring %1 with less than four points		
Ring %1 not closed		
Line %1 with less than two points		
Line %1 contains %n duplicate node(s) at %2	이 오류는 라인 상에 있는 연속되는 포인트들이 동일한 좌표를 가지고 있는 경우 발생합니다.	

다음 페이지에 계속

표 24.98 – 이전 페이지에서 계속

오류 메시지	설명	예시
Segments %1 and %2 of line %3 intersect at %4	이 오류는 라인이 스스로 교차하는 경우 (라인의 선분 2 개가 서로를 교차하는 경우) 발생합니다.	
Ring self-intersection	이 오류는 폴리곤 도형의 외곽 혹은 내곽 (섬) 고리/경계가 스스로 교차하는 경우 발생합니다.	
Ring %1 of polygon %2 not in exterior ring		
Polygon %1 lies inside polygon %2	이 오류는 멀티폴리곤 도형의 한 부분이 멀티폴리곤 도형의 구멍 (hole) 안에 있는 경우 발생합니다.	

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*

더 보기:

집계, 다중 부분으로 조성하기, 융해

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	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	명칭	유형	설명
Collected	`OUTPUT`	[same as input] 기본값: `[Create temporary layer]`	모아진 도형들을 저장할 산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 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 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

오목 껍질 (알파 형태)

입력 포인트 레이어에 있는 피처의 오목 껍질 (concave hull) 을 계산합니다.

더 보기:

볼록 껍질, 오목 껍질 *(k-*최근접 이웃*)*

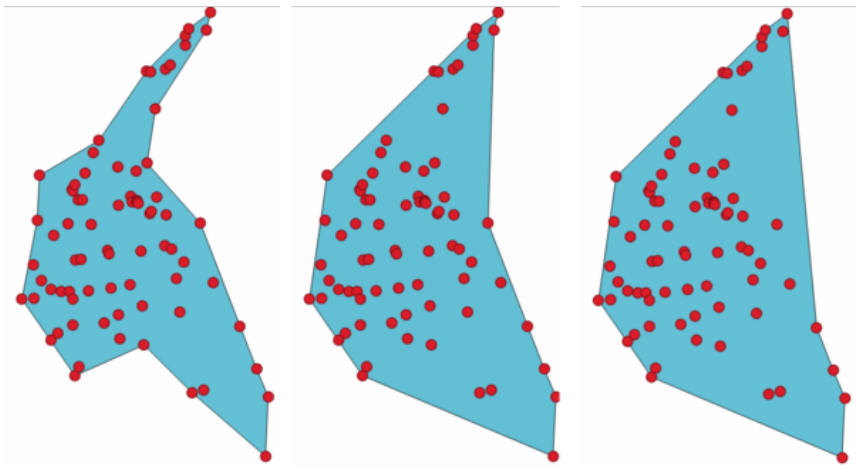


그림 24.49: 서로 다른 한계값 (0.3, 0.6, 0.9) 을 가진 오목 껍질들

파라미터

라벨	명칭	유형	설명
Input point layer	`INPUT`	[vector: point]	입력 포인트 벡터 레이어
Threshold	`ALPHA`	[number] 기본값: 0.3	0(최대 오목 껍질) 에서 1(볼록 껍질) 까지의 숫자
Allow holes	`HOLES`	[boolean] 기본값: True	최종 오목 껍질에 구멍을 허용할지 여부를 선택합니다.
Split multipart geometry into singlepart geometries	`NO_MULTIGEOMETRY`	[boolean] 기본값: True	다중 부분 도형 대신 단일 부분 도형을 원하는 경우 활성화하십시오.
Concave hull	`OUTPUT`	[vector: polygon] 기본값: `[Create temporary layer]`	산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Concave hull	`OUTPUT`	[vector: polygon]	산출 벡터 레이어

파이썬 코드

Algorithm ID: `qgis:concavehull`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 와 파라미터 명칭 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

오목 껍질 (k-최근접 이웃)

더 보기:

오목 껍질 (알파 형태)

파라미터

라벨	명칭	유형	설명
Input layer	INPUT	[vector: any]	입력 벡터 레이어
Number of neighboring points to consider (a lower number is more concave, a higher number is smoother)	KNEIGHBORS	[number] 기본값: 3	산출 폴리곤의 오목한 정도를 결정합니다. 개수가 낮을수록 포인트에 매우 근접하는 오목 껍질을 산출하고, 개수가 높을수록 더 볼록 껍질에 가까운 형태를 산출할 것입니다. (피처 개수 이상의 값은 볼록 껍질을 산출할 것입니다.) 최소값은 3 입니다.
Field 부가적	FIELD	[tablefield: any] 기본값: None	이 파라미터를 지정하면, 해당 필드의 각 유일값마다 (이 값을 사용해 피처를 선택해서) 오목 껍질 폴리곤 하나를 생성합니다.
Concave hull	OUTPUT	[vector: polygon] 기본값: [Create temporary layer]	산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (TEMPORARY_OUTPUT) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Concave hull	OUTPUT	[vector: polygon]	산출 벡터 레이어

파이썬 코드

Algorithm ID: `qgis:knearestconcavehull`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 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	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	입력 벡터 레이어
New geometry type	`TYPE`	[enumeration] 기본값: 0	산출 피처에 적용할 도형 유형입니다. 다음 가운데 하나로 지정할 수 있습니다: • 0 — 중심 (centroid) • 1 — 노드 • 2 — 라인스트링 • 3 — 멀티라인스트링 • 4 — 폴리곤
Converted	`OUTPUT`	[vector: any] 기본값: `[Create temporary layer]`	산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Converted	`OUTPUT`	[vector: any]	산출 벡터 레이어입니다. 파라미터에 따라 유형이 달라집니다.

파이썬 코드

Algorithm ID: `qgis: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 **명칭** 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.

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: line or polygon]	입력 벡터 레이어
Maximum distance tolerance	`DISTANCE`	[number] 기본값: 0.000001	꼭짓점의 원 위치와 변환된 만곡 도형에서의 위치 간에 허용되는 최장 거리
Maximum angle tolerance	`ANGLE`	[number] 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] 기본값: `[Create temporary layer]`	산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장… • GeoPackage 로 저장… • 데이터베이스 테이블에 저장… • 레이어에 추가… 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

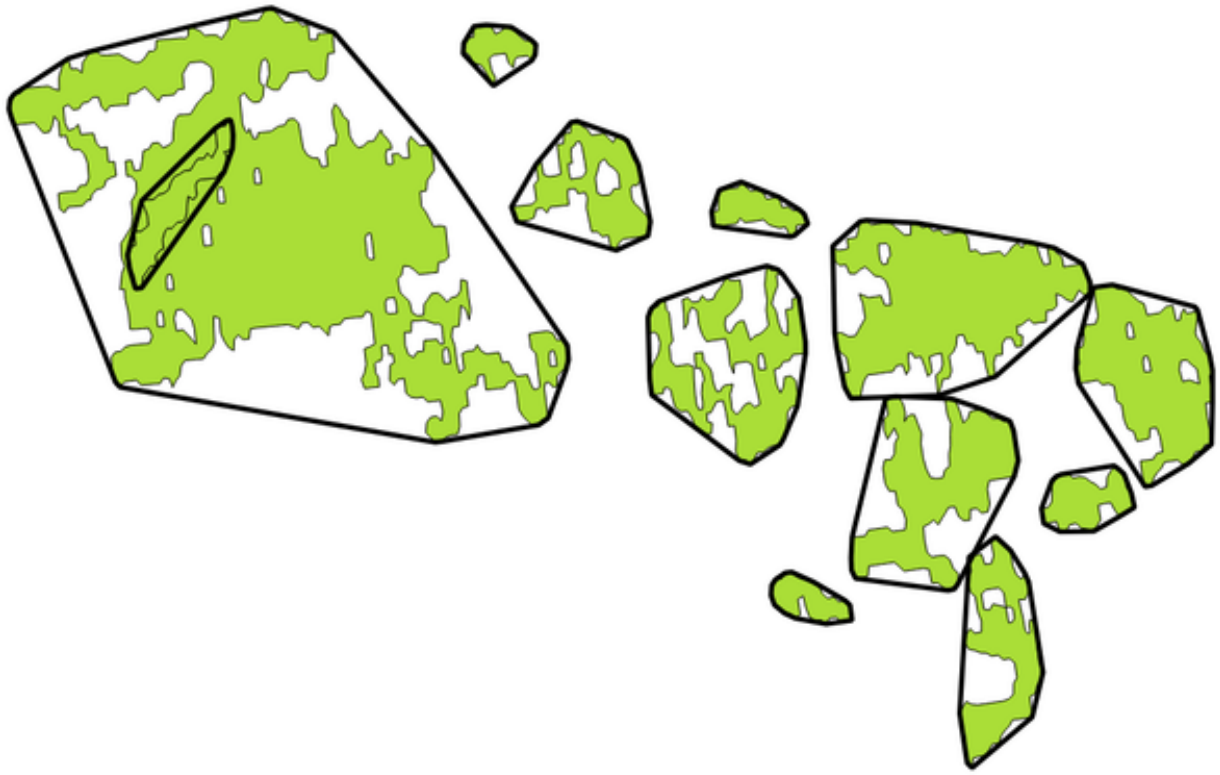


그림 24.50: 각 피처의 볼록 껍질을 표현하는 검은 선

제자리 피처 수정 옵션을 사용할 수 있습니다.

기본 메뉴: *Vector → Geoprocessing Tools*

더 보기:

최소 경계 도형, 오목 껍질 *(알파 형태)*

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	입력 벡터 레이어
Convex hull	`OUTPUT`	[vector: polygon] 기본값: `[Create temporary layer]`	산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Convex hull	`OUTPUT`	[vector: polygon]	산출 (볼록 껍질) 벡터 레이어

파이썬 코드

알고리즘 ID: `native:convexhull`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

범위로부터 레이어 생성하기

입력 레이어의 범위와 일치하는 도형을 가진 단일 피처를 담은 새 벡터 레이어를 생성합니다.

이 알고리즘은 모델에서 문자 그대로의 범위 (`xmin, xmax, ymin, ymax` 서식) 를 레이어 기반 입력을 요구하는 다른 알고리즘에 사용할 수 있는 레이어로 변환하기 위해 사용할 수 있습니다.

더 보기:

포인트로부터 레이어 생성하기

파라미터

라벨	명칭	유형	설명
Extent (xmin, xmax, ymin, ymax)	`INPUT`	[extent]	입력 범위
Extent	`OUTPUT`	[vector: polygon] 기본값: `[Create temporary layer]`	산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Extent	`OUTPUT`	[vector: polygon]	산출 (범위) 벡터 레이어

파이썬 코드

알고리즘 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 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

포인트로부터 레이어 생성하기

포인트 파라미터와 일치하는 도형을 가진 단일 피처를 담은 새 벡터 레이어를 생성합니다. 모델에서 포인트를 레이어 기반 입력을 요구하는 다른 알고리즘에 사용할 수 있는 포인트 레이어로 변환하기 위해 사용할 수 있습니다.

더 보기:

범위로부터 레이어 생성하기

파라미터

라벨	명칭	유형	설명
Point	`INPUT`	[coordinates]	좌표계 정보 (예: `397254,6214446 [EPSG:32632]`) 를 포함하는 입력 포인트입니다. 좌표계가 지정되지 않은 경우, 프로젝트 좌표계를 사용할 것입니다. 맵 캔버스를 클릭해서 포인트를 지정할 수 있습니다.
Point	`OUTPUT`	[vector: point] 기본값: `[Create temporary layer]`	산출 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Point	`OUTPUT`	[vector: point]	입력 포인트를 담고 있는 산출 포인트 벡터 레이어입니다.

파이썬 코드

알고리즘 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 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

쐐기 버퍼 생성하기

입력 포인트로부터 쐐기 (wedge) 모양 버퍼를 생성합니다.

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

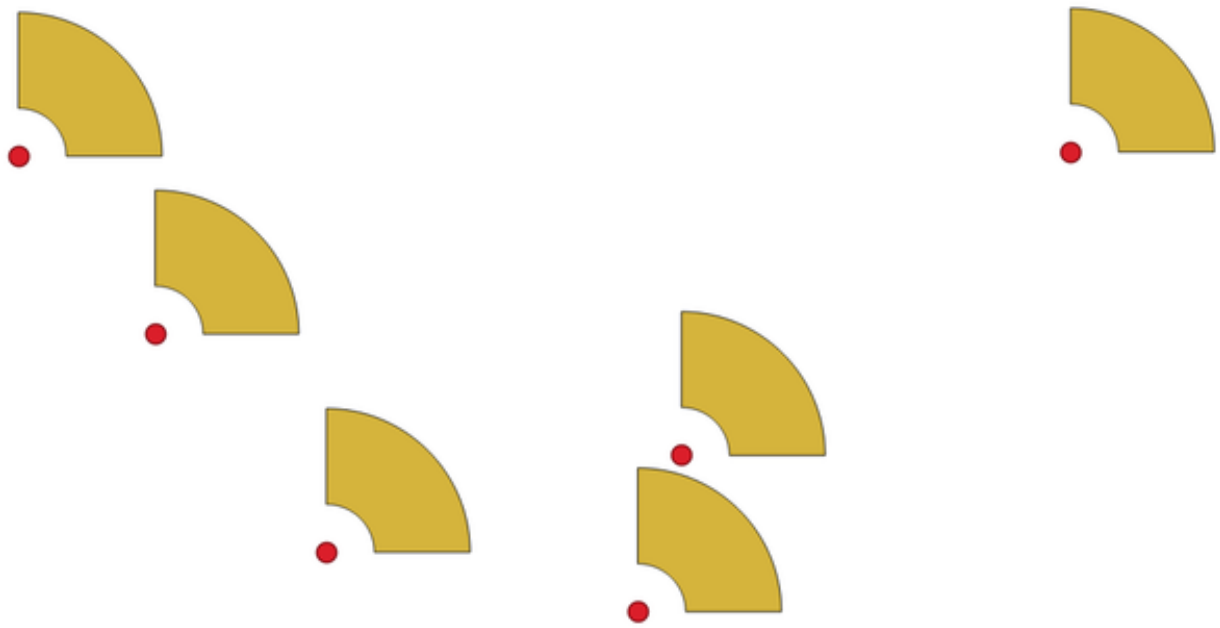


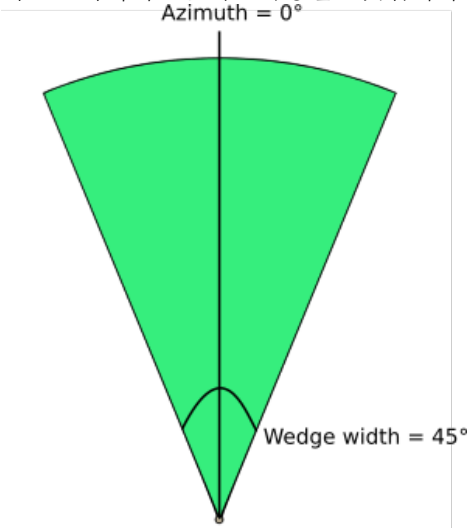
그림 24.51: 쐐기 버퍼

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: point]	입력 포인트 벡터 레이어
Azimuth (degrees from North)	`AZIMUTH`	[number] 기본값: 0.0	쐐기의 중간값 (도 단위) 각도

다음 페이지에 계속

표 24.101 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Wedge width (in degrees)	`WIDTH`	[number] 기본값: 45.0	버퍼의 (도 단위) 너비입니다. 쐐기는 방위각 방향의 양쪽으로 각도 너비의 반씩 확장될 것입니다.  그림 24.52: 쐐기 버퍼의 방위각 및 너비 값
Outer radius	`OUTER_RADIUS`	[number] 기본값: 1.0	쐐기의 외곽 크기 (길이) 입니다. 이 크기는 소스 포인트에서 쐐기 모양의 경계까지의 거리를 의미합니다.
Inner radius 부가적	`INNER_RADIUS`	[number] 기본값: 0.0	내곽 반경 값입니다. 0 으로 설정하면 쐐기가 소스 포인트에서 시작할 것입니다.
Buffers	`OUTPUT`	[vector: polygon] 기본값: `[Create temporary layer]`	산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

들로네 삼각분할

입력 포인트 레이어에 대응하는 들로네 삼각분할 (Delaunay triangulation) 을 통해 폴리곤 레이어를 생성합니다.

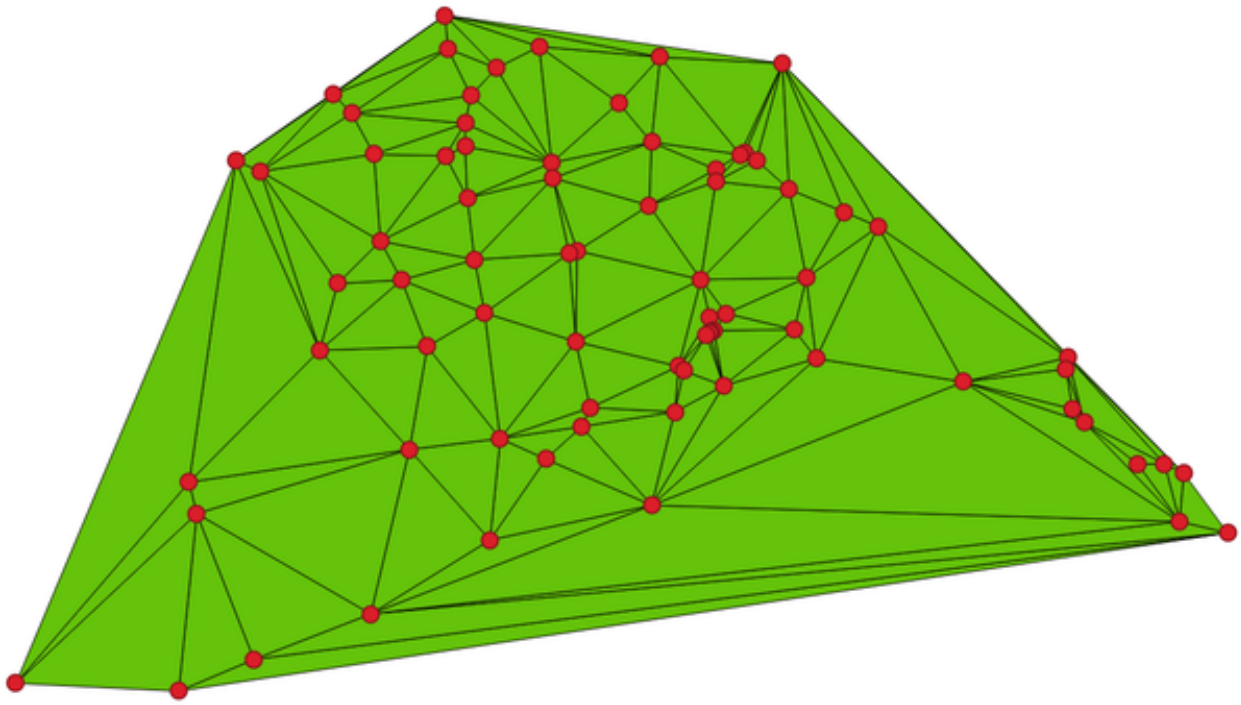


그림 24.53: 포인트에 대한 들로네 삼각분할

기본 메뉴: *Vector → Geometry Tools*

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: point]	입력 포인트 벡터 레이어
Delaunay triangulation	`OUTPUT`	[vector: polygon] 기본값: `[Create temporary layer]`	산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Delaunay triangulation	`OUTPUT`	[vector: polygon]	산출 (들로네 삼각분할) 벡터 레이어

파이썬 코드

Algorithm ID: `qgis: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 **명칭** 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.

제자리 피처 수정 옵션을 사용할 수 있습니다.

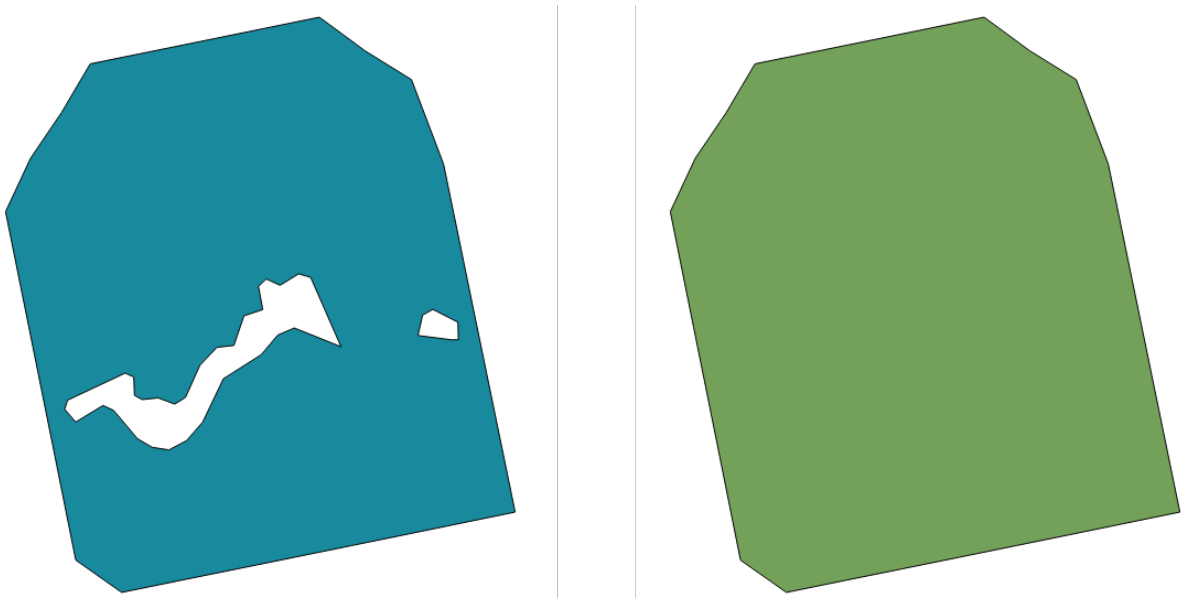


그림 24.54: 청소 이전과 이후

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: polygon]	입력 폴리곤 벡터 레이어
Remove holes with area less than 부가적	`MIN_AREA`	[number] 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] 기본값: `[Create temporary layer]`	산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

개수로 치밀화하기

폴리곤 레이어와 라인 레이어를 받아 도형이 원본보다 더 많은 꼭짓점을 가지는 새 레이어를 생성합니다.

도형이 Z 또는 M 값을 가지고 있을 경우 추가된 꼭짓점에 이 값을 선형적으로 보간할 것입니다.

각 선분에 추가할 새 꼭짓점의 개수는 입력 파라미터로 지정합니다.

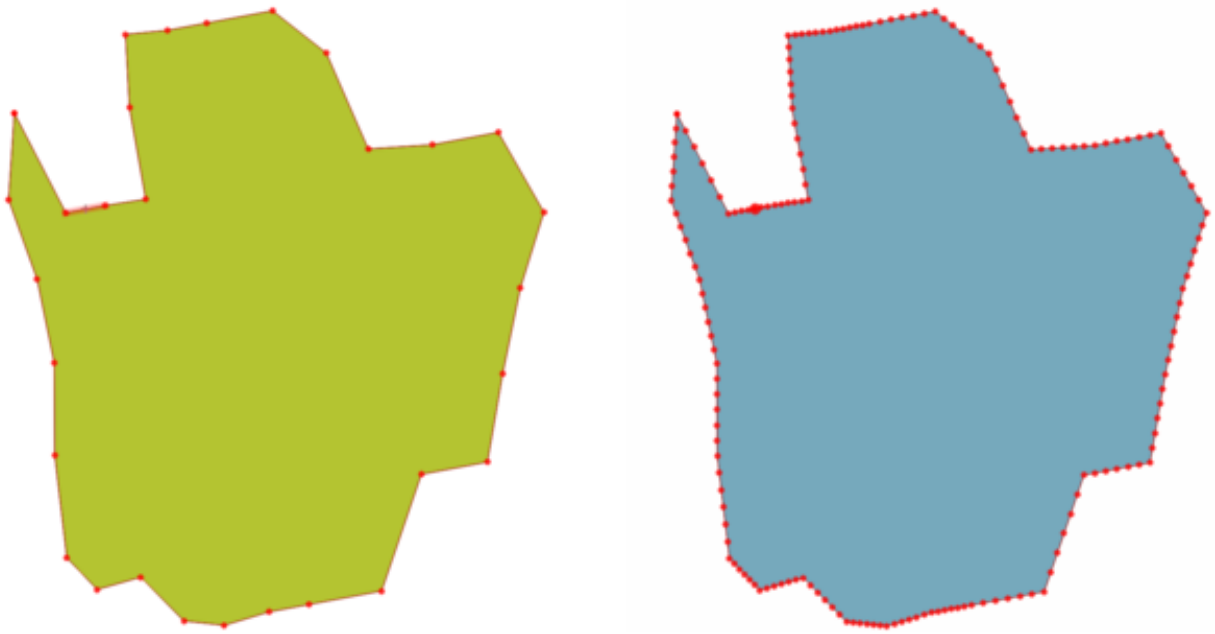


그림 24.55: 치밀화 이전과 이후의 꼭짓점을 표시하는 빨간 포인트

제자리 피처 수정 옵션을 사용할 수 있습니다.

기본 메뉴: *Vector → Geometry Tools*

더 보기:

간격으로 치밀화하기

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: line, polygon]	입력 라인 또는 폴리곤 벡터 레이어
Vertices to add	`VERTICES`	[number] 기본값: 1	각 선분에 추가할 새 꼭짓점의 개수
Densified	`OUTPUT`	[same as input] 기본값: `[Create temporary layer]`	산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

간격으로 치밀화하기

폴리곤 레이어와 라인 레이어를 받아 도형이 원본보다 더 많은 꼭짓점을 가지는 새 레이어를 생성합니다.

어느 두 꼭짓점 사이의 거리가 지정한 거리를 초과하지 않도록 각 선분 내부에 추가 꼭짓점을 규칙적으로 추가해서 도형을 치밀화 (densify) 합니다.

도형이 Z 또는 M 값을 가지고 있을 경우 추가된 꼭짓점에 이 값을 선형적으로 보간할 것입니다.

예시

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.

제자리 피처 수정 옵션을 사용할 수 있습니다.

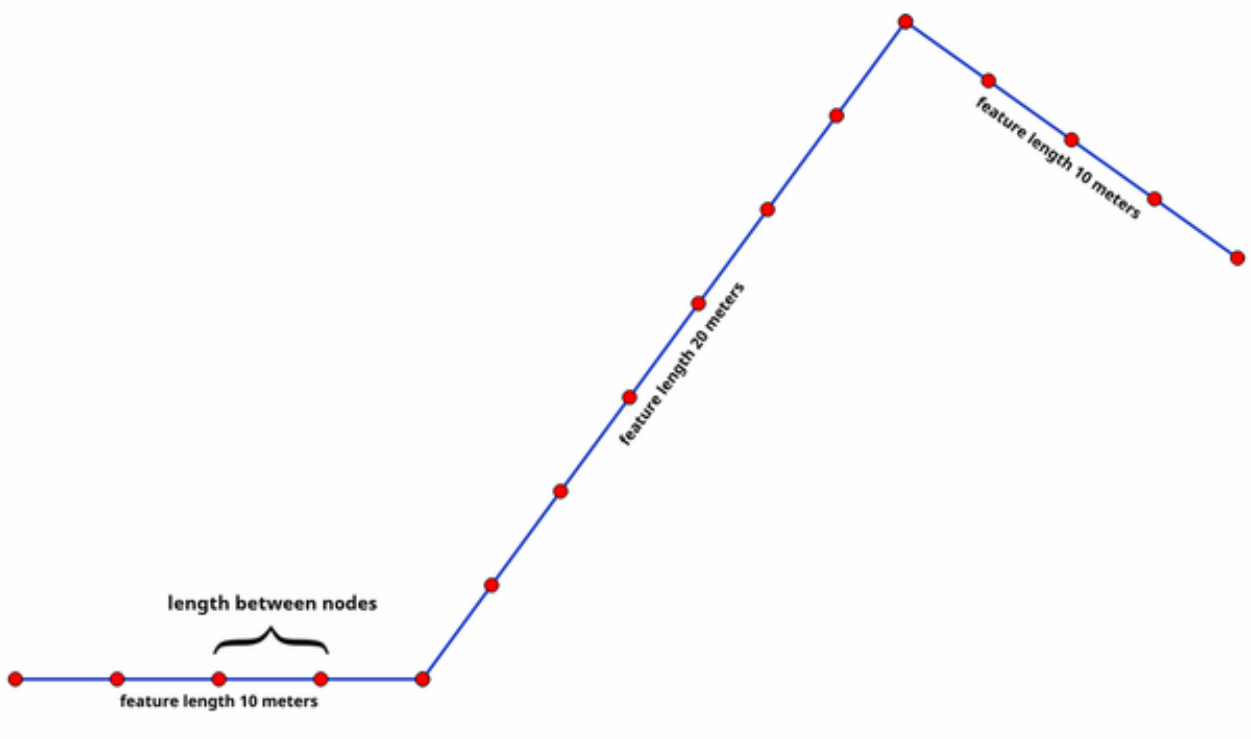


그림 24.56: 지정한 간격으로 도형 치밀화

더 보기:

개수로 치밀화하기

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: line, polygon]	입력 라인 또는 폴리곤 벡터 레이어
Interval between vertices to add	`INTERVAL`	[number] 기본값: 1.0	연속된 두 꼭짓점 사이의 최대 거리
Densified	`OUTPUT`	[same as input] 기본값: `[Create temporary layer]`	산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

산출 도형 모두 다중 도형으로 변환될 것입니다. 입력 레이어가 폴리곤 레이어인 경우, 융해되는 인접한 폴리곤들의 공통 경계는 지워질 것입니다.

산출 속성 테이블은 입력 레이어와 동일한 필드를 가질 것입니다. 산출 레이어의 필드에 있는 값들은 제일 먼저 처리될 입력 피처의 값들입니다.

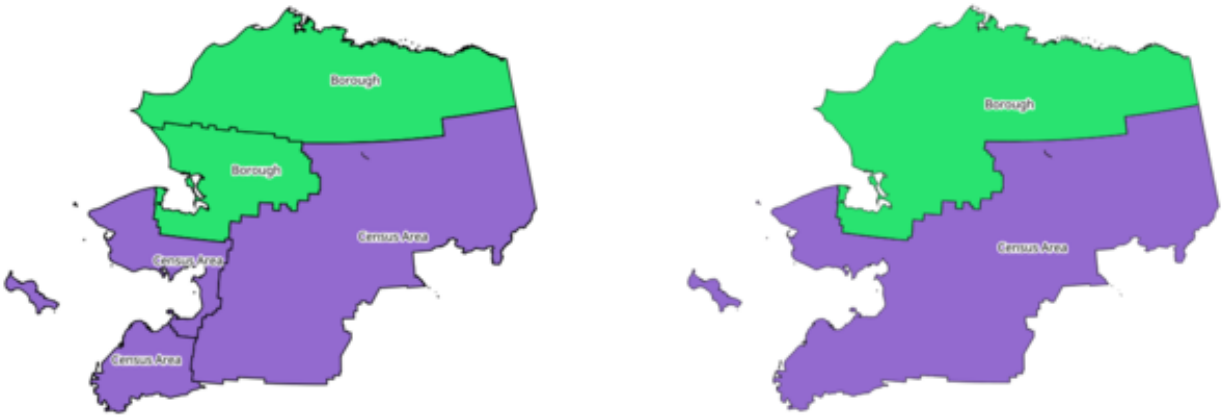


그림 24.57: 공통 속성을 바탕으로 폴리곤 레이어 융해

기본 메뉴: *Vector → Geoprocessing Tools*

더 보기:

집계, 도형 모으기

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	입력 벡터 레이어
Dissolve field(s) 부가적	`FIELD`	[tablefield: any] [list] 기본값: []	선택한 필드 (들) 에서 동일한 값을 가지고 있는 피처들을 단일 피처로 대체하고 도형들을 병합할 것입니다. 아무 필드도 선택하지 않은 경우 모든 피처를 융해시켜 단일 (다중 부분) 피처를 산출합니다.
Dissolved	`OUTPUT`	[same as input] 기본값: `[Create temporary layer]`	산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

더 보기:

래스터로부터 *M* 값 설정하기, *Z* 값 설정하기

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	입력 벡터 레이어
Raster layer	`RASTER`	[raster]	Z 값을 가진 래스터 레이어
Band number	`BAND`	[raster band] 기본값: 1	Z 값을 가져올 래스터 밴드
Value for nodata or non-intersecting vertices	`NODATA`	[number] 기본값: 0	꼭짓점이 래스터 (의 유효한 픽셀) 와 교차하지 않는 경우 사용할 값
Scale factor	`SCALE`	[number] 기본값: 1.0	척도 값: 밴드 값을 이 값으로 곱합니다.
Updated	`OUTPUT`	[same as input] 기본값: `[Create temporary layer]`	(래스터 레이어에서 나온 Z 값을 가진) 산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	Z 또는 M 값을 가진 입력 벡터 레이어
Drop M Values	`DROP_M_VALUES`	[boolean] 기본값: False	도형에서 M 값을 제거합니다.
Drop Z Values	`DROP_Z_VALUES`	[boolean] 기본값: False	도형에서 Z 값을 제거합니다.
Z/M Dropped	`OUTPUT`	[same as input] 기본값: `[Create temporary layer]`	산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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	명칭	유형	설명
Input layer	`INPUT`	[vector: polygon]	입력 폴리곤 벡터 레이어
Merge selection with the neighboring polygon with the	`MODE`	[enumeration] 기본값: None	선택한 폴리곤을 제거하기 위해 사용할 파라미터를 선택합니다: • 0 — 최대 면적 (Largest Area) • 1 — 최소 면적 (Smallest Area) • 2 — 최대 공유 경계 (Largest Common Boundary)
Eliminated	`OUTPUT`	[vector: polygon] 기본값: `[Create temporary layer]`	산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Eliminated	`OUTPUT`	[vector: polygon]	산출 폴리곤 벡터 레이어

파이썬 코드

Algorithm 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 **명칭** 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.

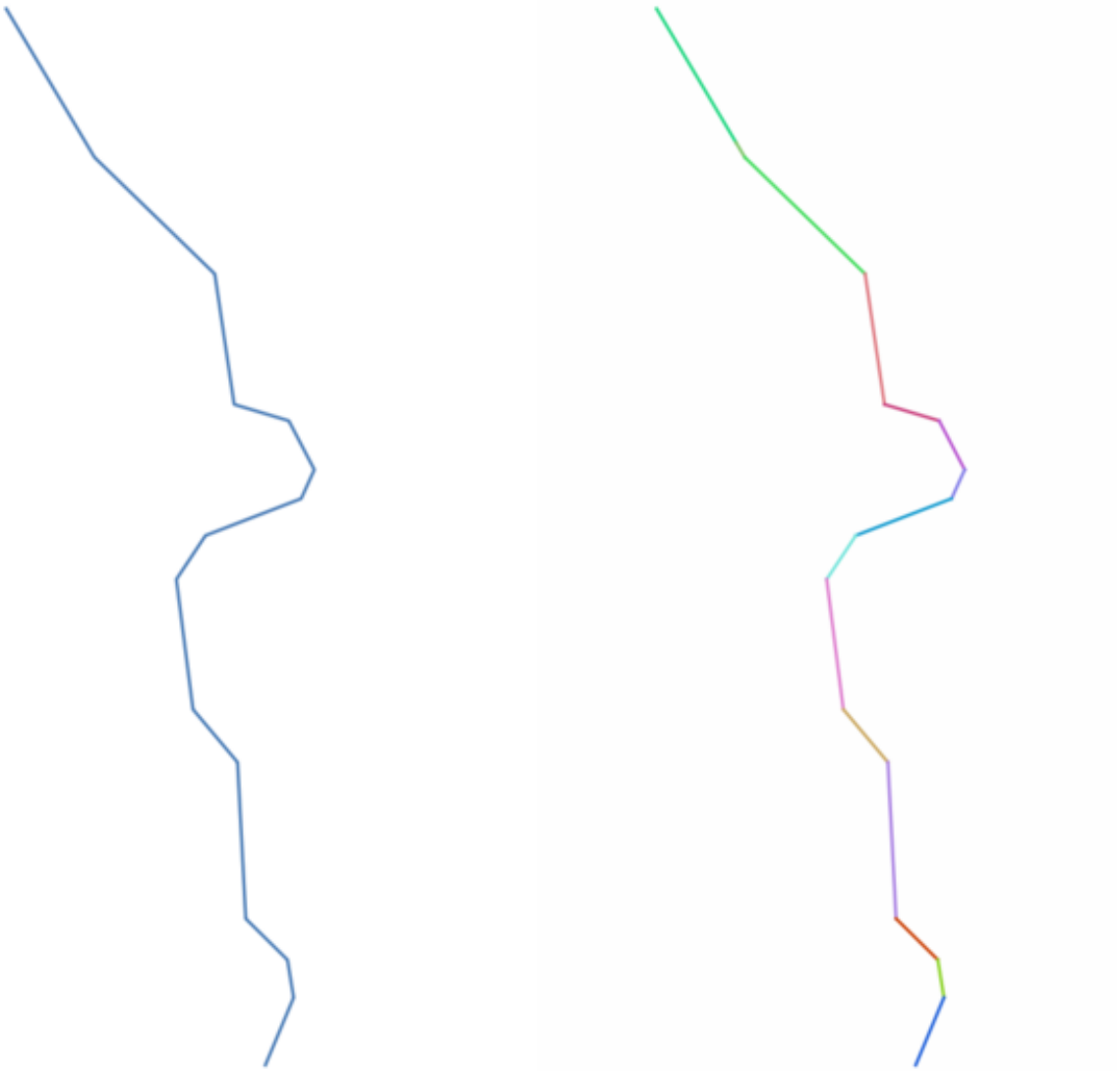


그림 24.58: 원본 라인 레이어와 조각낸 레이어

제자리 피처 수정 옵션을 사용할 수 있습니다.

더 보기:

세분화, 라인 부스트링 생성하기

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: line]	입력 라인 벡터 레이어
Exploded	`OUTPUT`	[vector: line] 기본값: `[Create temporary layer]`	산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

라인 연장하기

라인의 시작과 끝에서 양을 지정해서 라인 도형을 연장합니다.

라인의 첫 번째와 마지막 선분 방향을 따라 라인을 연장합니다.

제자리 피처 수정 옵션을 사용할 수 있습니다.

더 보기:

라인 부스트링 생성하기

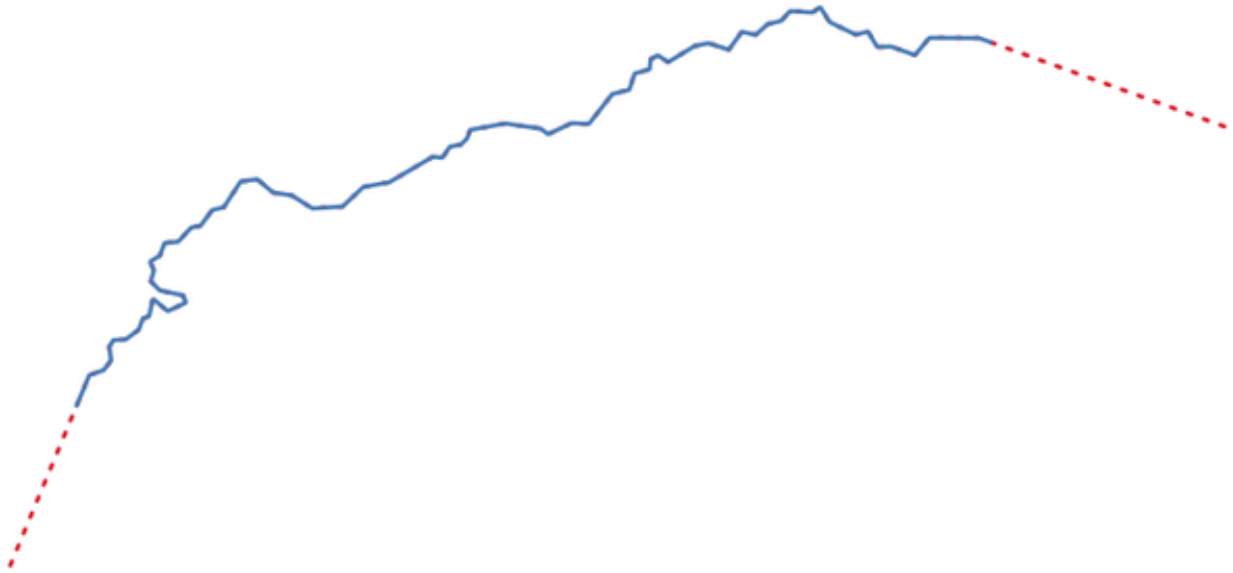


그림 24.59: 원본 레이어의 처음과 마지막 연장을 표현하는 빨간 점선

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: line]	입력 라인 벡터 레이어
Start distance	`START_DISTANCE`	[number]	라인의 첫 번째 선분을 연장할 거리 (시작점)
End distance	`END_DISTANCE`	[number]	라인의 마지막 선분을 연장할 거리 (시작점)
Extended	`OUTPUT`	[vector: line] 기본값: `[Create temporary layer]`	산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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)* 은 파라미터 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

M 값 추출하기

도형에서 피처 속성으로 M 값을 추출합니다.

기본적으로 각 피처의 첫 꼭짓점의 M 값만 추출하지만, 도형의 모든 M 값에 대해 합계, 평균, 최소값 및 최대값을 포함하는 통계를 계산할 수 있는 옵션이 있습니다.

더 보기:

Z 값 추출하기, *M* 값 설정하기, *M/Z* 값 제거

파라미터

라벨	명칭	유형	설명
Input layer	INPUT	[vector: any]	입력 벡터 레이어

다음 페이지에 계속

표 24.103 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Summaries to calculate	SUMMARIES	[enumeration] 기본값: [0]	도형의 M 값에 대한 통계를 계산합니다. 다음 가운데 하나 이상을 지정할 수 있습니다: • 0 —첫 번째 • 1 —마지막 • 2 —개수 • 3 —합계 • 4 —평균 • 5 —중간값 • 6 —표준 편차 (population) • 7 —최소값 • 8 —최대값 • 9 —범위 (range) • 10 —소수 기준 (Minority) • 11 —다수 기준 (Majority) • 12 —다양도 (Variety) • 13 —제 1 사분위수 • 14 —제 3 사분위수 • 15 —사분위수의 범위
Output column prefix	COLUMN_PREFIX	[string] 기본값: 'm_'	산출 (M) 열의 명칭 앞에 붙일 접두어
Extracted	OUTPUT	[same as input] 기본값: [Create temporary layer]	산출 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (TEMPORARY_OUTPUT) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

더 보기:

꼭짓점 추출하기, *M* 값으로 꼭짓점 필터링하기, *Z* 값으로 꼭짓점 필터링하기

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	입력 벡터 레이어
Vertex indices	`VERTICES`	[string] 기본값: '0'	추출할 꼭짓점의 인덱스를 가리키는 쉼표로 구분된 문자열
Vertices	`OUTPUT`	[vector: point] 기본값: `[Create temporary layer]`	산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Vertices	OUTPUT	[vector: point]	입력 레이어 도형에서 지정한 꼭짓점을 담고 있는 산출 (포인트) 벡터 레이어

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 **명칭** 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.

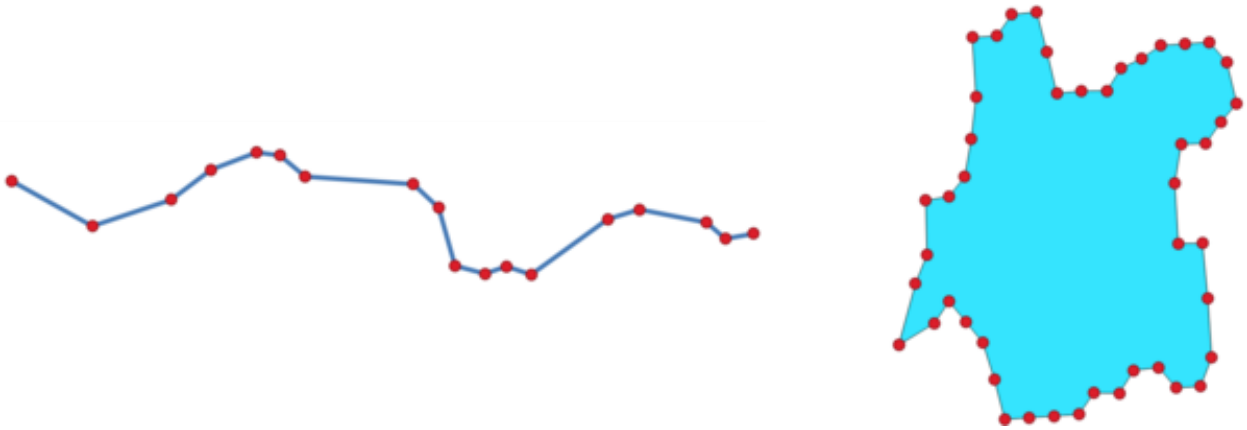


그림 24.60: 라인 및 폴리곤 레이어에서 추출한 꼭짓점들

기본 메뉴: *Vector → Geometry Tools*

더 보기:

특정 꼭짓점 추출하기, *M* 값으로 꼭짓점 필터링하기, *Z* 값으로 꼭짓점 필터링하기

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	입력 벡터 레이어
Vertices	`OUTPUT`	[vector: point] 기본값: `[Create temporary layer]`	산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Vertices	`OUTPUT`	[vector: point]	입력 레이어 도형의 꼭짓점을 담고 있는 산출 (포인트) 벡터 레이어

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 **명칭** 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	명칭	유형	설명
Input layer	INPUT	[vector: any]	입력 벡터 레이어
Summaries to calculate	SUMMARIES	[enumeration] 기본값: [0]	도형의 Z 값에 대한 통계를 계산합니다. 다음 가운데 하나 이상을 지정할 수 있습니다: • 0 —첫 번째 • 1 —마지막 • 2 —개수 • 3 —합계 • 4 —평균 • 5 —중간값 • 6 —표준 편차 (population) • 7 —최소값 • 8 —최대값 • 9 —범위 (range) • 10 —소수 기준 (Minority) • 11 —다수 기준 (Majority) • 12 —다양도 (Variety) • 13 —제 1 사분위수 • 14 —제 3 사분위수 • 15 —사분위수의 범위
Output column prefix	COLUMN_PREFIX	[string] 기본값: 'z_'	산출 (Z) 열의 명칭 앞에 붙일 접두어
Extracted	OUTPUT	[same as input] 기본값: [Create temporary layer]	산출 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (TEMPORARY_OUTPUT) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

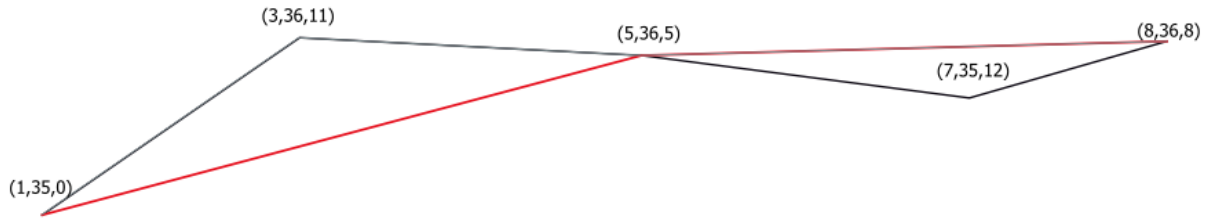


그림 24.61: 검은 라인에서 M 값이 10 이하인 꼭짓점만 잇는 빨간 라인

참고: 입력 도형의 속성 및 사용된 필터에 따라, 이 알고리즘이 생상하는 산출 도형이 더 이상 유효하지 않을 수도 있습니다.

더 보기:

Z 값으로 꼭짓점 필터링하기, 꼭짓점 추출하기, 특정 꼭짓점 추출하기

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: line, polygon]	꼭짓점을 제거할 입력 라인 또는 폴리곤 벡터 레이어
Minimum 부가적	`MIN`	[number] 기본값: *Not set*	허용되는 최소 M 값
Maximum 부가적	`MAX`	[number] 기본값: *Not set*	허용되는 최대 M 값
Filtered	`OUTPUT`	[same as input] 기본값: `[Create temporary layer]`	산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

참고: 입력 도형의 속성 및 사용된 필터에 따라, 이 알고리즘이 생상하는 산출 도형이 더 이상 유효하지 않을 수도 있습니다. 산출 도형의 유효성을 보장하려면도형 고치기 알고리즘을 실행해야 할 수도 있습니다.

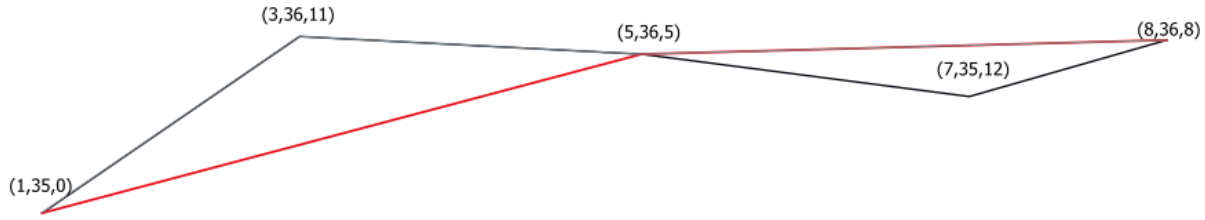


그림 24.62: 검은 라인에서 Z 값이 10 이하인 꼭짓점만 잇는 빨간 라인

더 보기:

M 값으로 꼭짓점 필터링하기, 꼭짓점 추출하기, 특정 꼭짓점 추출하기

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: line, polygon]	꼭짓점을 제거할 입력 라인 또는 폴리곤 벡터 레이어
Minimum 부가적	`MIN`	[number] 기본값: *Not set*	허용되는 최소 Z 값
Maximum 부가적	`MAX`	[number] 기본값: *Not set*	허용되는 최대 Z 값
Filtered	`OUTPUT`	[same as input] 기본값: `[Create temporary layer]`	산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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 유효하지 않은 도형의 유효한 표현을 생성하려 합니다. 이미 유효한 도형은 다른 처리 없이 그대로 반환됩니다. 이 알고리즘은 항상 다중 도형 레이어를 산출합니다.

참고: 산출물에서 M 값은 제거될 것입니다.

제자리 피처 수정 옵션을 사용할 수 있습니다.

더 보기:

유효성 점검하기

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	입력 벡터 레이어
Fixed geometries	`OUTPUT`	[same as input] 기본값: `[Create temporary layer]`	산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: polygon]	입력 벡터 레이어
Reoriented	`OUTPUT`	[vector: polygon] 기본값: `[Create temporary layer]`	산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: line]	입력 라인 벡터 레이어
Split	`OUTPUT`	[vector: line] 기본값: `[Create temporary layer]`	산출 라인 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	입력 벡터 레이어
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: • 0 — 폴리곤 • 1 — 라인 • 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: any] 기본값: `[Create temporary layer]`	산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Modified geometry	`OUTPUT`	[vector: any]	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 **명칭** 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 24.63: 라인의 시작에서 500m 거리에 보간된 포인트

더 보기:

도형을 따라 포인트 생성하기

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: line, polygon]	입력 라인 또는 폴리곤 벡터 레이어
Distance	`DISTANCE`	[number] 기본값: 0.0	라인의 시작으로부터의 거리
Interpolated points	`OUTPUT`	[vector: point] 기본값: `[Create temporary layer]`	산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

파라미터

라벨	명칭	유형	설명
Polygons	`INPUT`	[vector: polygon]	입력 폴리곤 벡터 레이어
Parts to keep	`PARTS`	[number] 기본값: 1	남길 부분의 개수. 1 로 설정하면 피처의 가장 큰 부분만 남길 것입니다.
Parts	`OUTPUT`	[vector: polygon] 기본값: `[Create temporary layer]`	산출 폴리곤 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

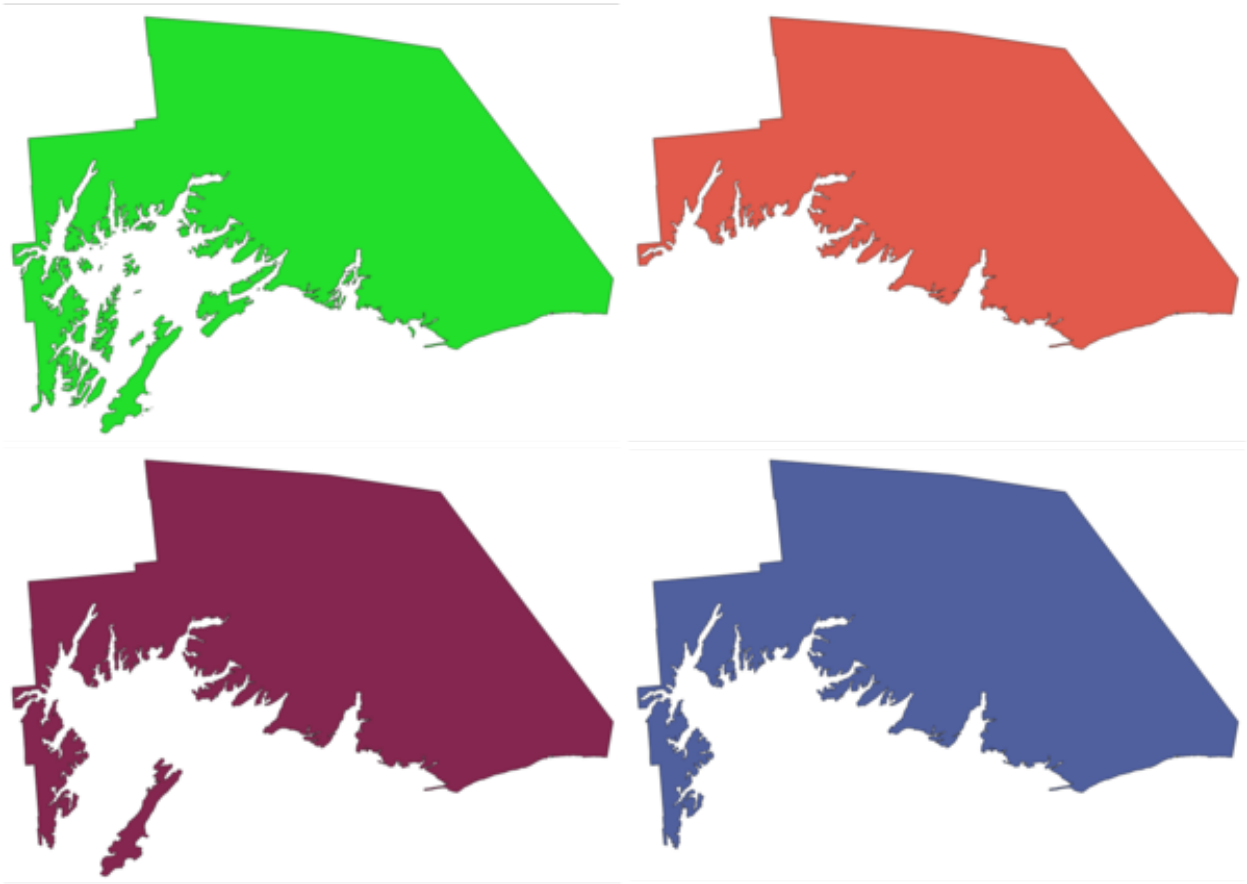


그림 24.64: 좌상단부터 시계 방향으로: 원본 다중 부분 피처, 가장 큰 1, 2, 3 개의 부분을 남긴 피처들

산출물

라벨	명칭	유형	설명
Parts	`OUTPUT`	[vector: polygon]	각 피처의 가장 큰 N 개의 부분을 담은 산출 폴리곤 벡터 레이어

파이썬 코드

Algorithm ID: `qgis: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 **명칭** 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.

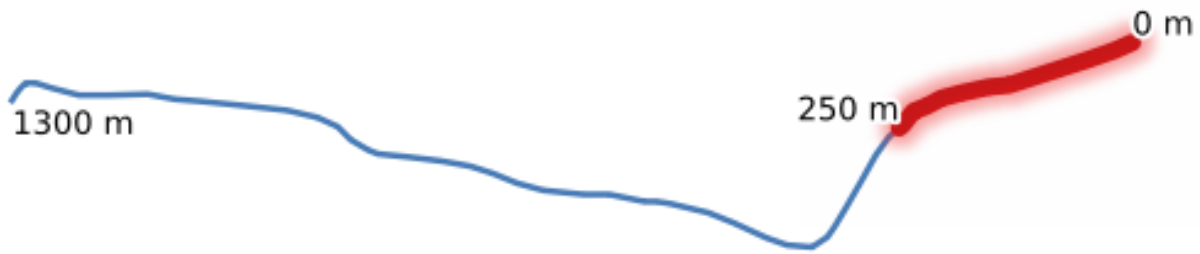


그림 24.65: 시작 거리를 0m 로, 종단 거리를 250m 로 설정한 부스트링 라인

제자리 피처 수정 옵션을 사용할 수 있습니다.

더 보기:

라인 연장하기

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: line]	입력 라인 벡터 레이어
Start distance	`START_DISTANCE`	[number]	입력 라인을 따라 산출 피처의 시작점까지의 거리
End distance	`END_DISTANCE`	[number]	입력 라인을 따라 산출 피처의 종단점까지의 거리
Substring	`OUTPUT`	[vector: line] 기본값: `[Create temporary layer]`	산출 라인 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

기본 메뉴: *Vector → Geometry Tools*

더 보기:

폴리곤을 라인으로, 폴리곤화, 도형 유형 변환하기

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: line]	입력 라인 벡터 레이어
Polygons	`OUTPUT`	[vector: polygon] 기본값: `[Create temporary layer]`	산출 폴리곤 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Polygons	`OUTPUT`	[vector: polygon]	산출 폴리곤 벡터 레이어

파이썬 코드

Algorithm 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 **명칭** 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.

제자리 피처 수정 옵션을 사용할 수 있습니다.

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: line]	입력 라인 벡터 레이어
Merged	`OUTPUT`	[vector: line] 기본값: `[Create temporary layer]`	산출 라인 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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

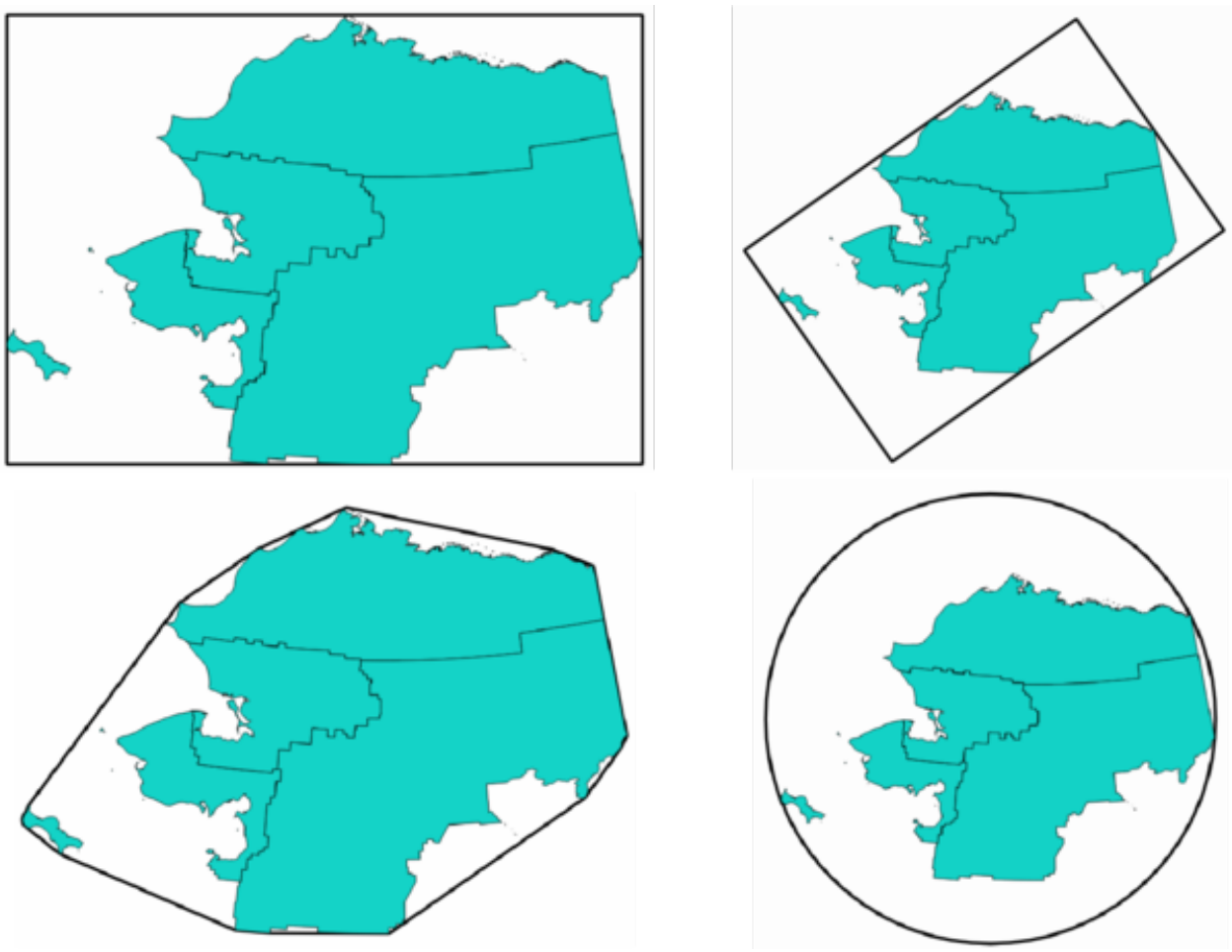


그림 24.66: 좌상단부터 시계 방향으로: 엔벨로프, 기울어진 직사각형, 원, 볼록 껍질

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	입력 벡터 레이어
Field 부가적	`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] 기본값: 0	외함 도형 유형을 지정합니다. 다음 가운데 하나를 선택할 수 있습니다: • 0 — 엔벨로프 (경계 상자) • 1 — 최소 기울어진 직사각형 • 2 — 최소 외함 원 • 3 — 볼록 껍질
Bounding geometry	`OUTPUT`	[vector: polygon] 기본값: `[Create temporary layer]`	산출 폴리곤 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Bounding geometry	`OUTPUT`	[vector: polygon]	산출 (경계) 폴리곤 벡터 레이어

파이썬 코드

Algorithm 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 **명칭** 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.

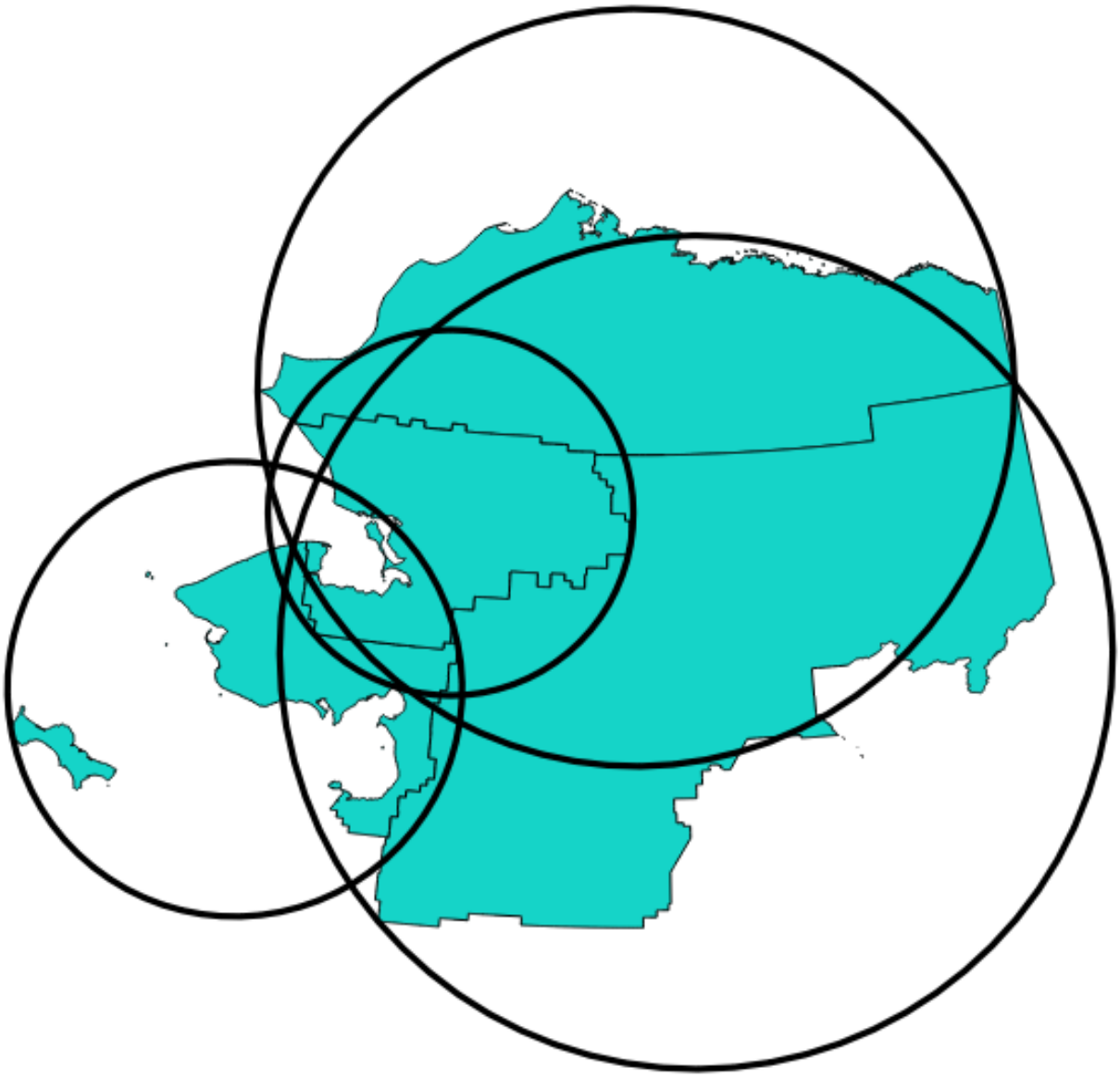


그림 24.67: 각 피처를 외함하는 원들

제자리 피처 수정 옵션을 사용할 수 있습니다.

더 보기:

최소 경계 도형

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	입력 벡터 레이어
Number of segment in circles	`SEGMENTS`	[number] 기본값: 72	원에 가깝게 만드는 데 사용할 선분의 개수입니다. 최소값은 8, 최대값은 100000 입니다.
Minimum enclosing circles	`OUTPUT`	[vector: polygon] 기본값: `[Create temporary layer]`	산출 폴리곤 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Minimum enclosing circles	`OUTPUT`	[vector: polygon]	산출 폴리곤 벡터 레이어

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 **명칭** 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.

제자리 피처 수정 옵션을 사용할 수 있습니다.

더 보기:

버퍼, 변동 거리 버퍼, 직사각형, 타원형, 마름모, 한쪽 버퍼 생성하기

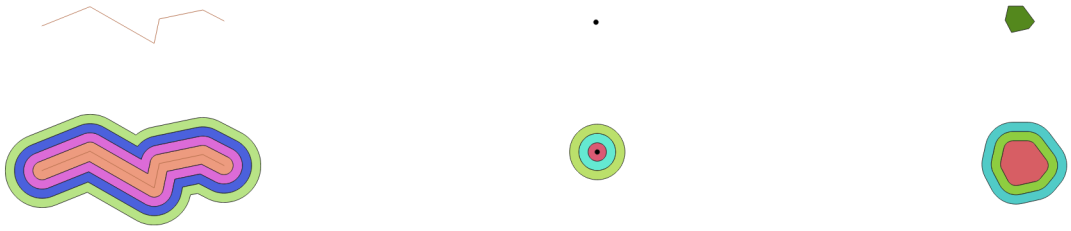


그림 24.68: 라인, 포인트 및 폴리곤 레이어의 다중 고리 버퍼

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	입력 벡터 레이어
Number of rings	`RINGS`	[number] 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`	[number] 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] 기본값: `[Create temporary layer]`	산출 폴리곤 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Multi-ring buffer (constant distance)	`OUTPUT`	[vector: polygon]	산출 폴리곤 벡터 레이어

파이썬 코드

알고리즘 ID: `native:multiringconstantbuffer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 각 파라미터의 명칭 및 값으로 구성합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

다중 부분에서 단일 부분으로

입력 레이어의 다중 부분 피처를 단일 부분 피처로 분할합니다.
산출 레이어의 속성은 입력 레이어와 동일하지만 개별 피처별로 나뉩니다.



그림 24.69: 왼쪽: 다중 부분 소스 레이어, 오른쪽: 단일 부분 산출 레이어

☑ 제자리 피처 수정 옵션을 사용할 수 있습니다.

기본 메뉴: *Vector ▸ Geometry Tools*

더 보기:

도형 모으기, 다중 부분으로 조성하기

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	입력 벡터 레이어
Single parts	`OUTPUT`	[same as input] 기본값: `[Create temporary layer]`	산출 폴리곤 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

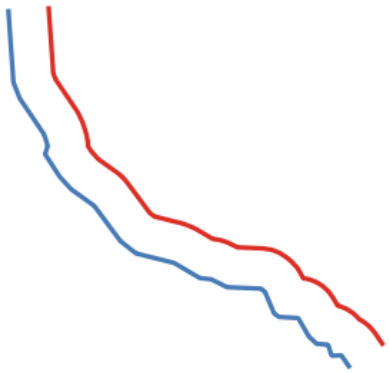


그림 24.70: 소스 피처는 파란색, 오프셋 피처는 빨간색

제자리 피처 수정 옵션을 사용할 수 있습니다.

더 보기:

오프셋 *(*평행*)* 라인 배열, 이동시키기 *(Translate)*

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: line]	입력 라인 벡터 레이어
Distance	`DISTANCE`	[number] 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. (변동 거리 버퍼 참조)
Segments	`SEGMENTS`	[number] 기본값: 8	둥글린 (rounded) 오프셋을 생성하는 경우 사분원에 가깝게 만드는 데 사용할 라인 선분의 개수를 제어합니다.

다음 페이지에 계속

표 24.109 – 이전 페이지에서 계속

라벨	명칭	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: • 0 —둥글림 (Round) • 1 —마이터 (Miter) • 2 —베벨 (Bevel)
Miter limit	MITER_LIMIT	[number] 기본값: 2.0	마이터 결합 스타일에만 적용할 수 있으며, 마이터 결합 생성시 사용할 오프셋 곡선으로부터의 최장 거리를 제어합니다. (최소값은 1 입니다.)
Offset	OUTPUT	[vector: line] 기본값: [Create temporary layer]	산출 (오프셋) 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (TEMPORARY_OUTPUT) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

제자리 피처 수정 옵션을 사용할 수 있습니다.

더 보기:

최소 경계 도형

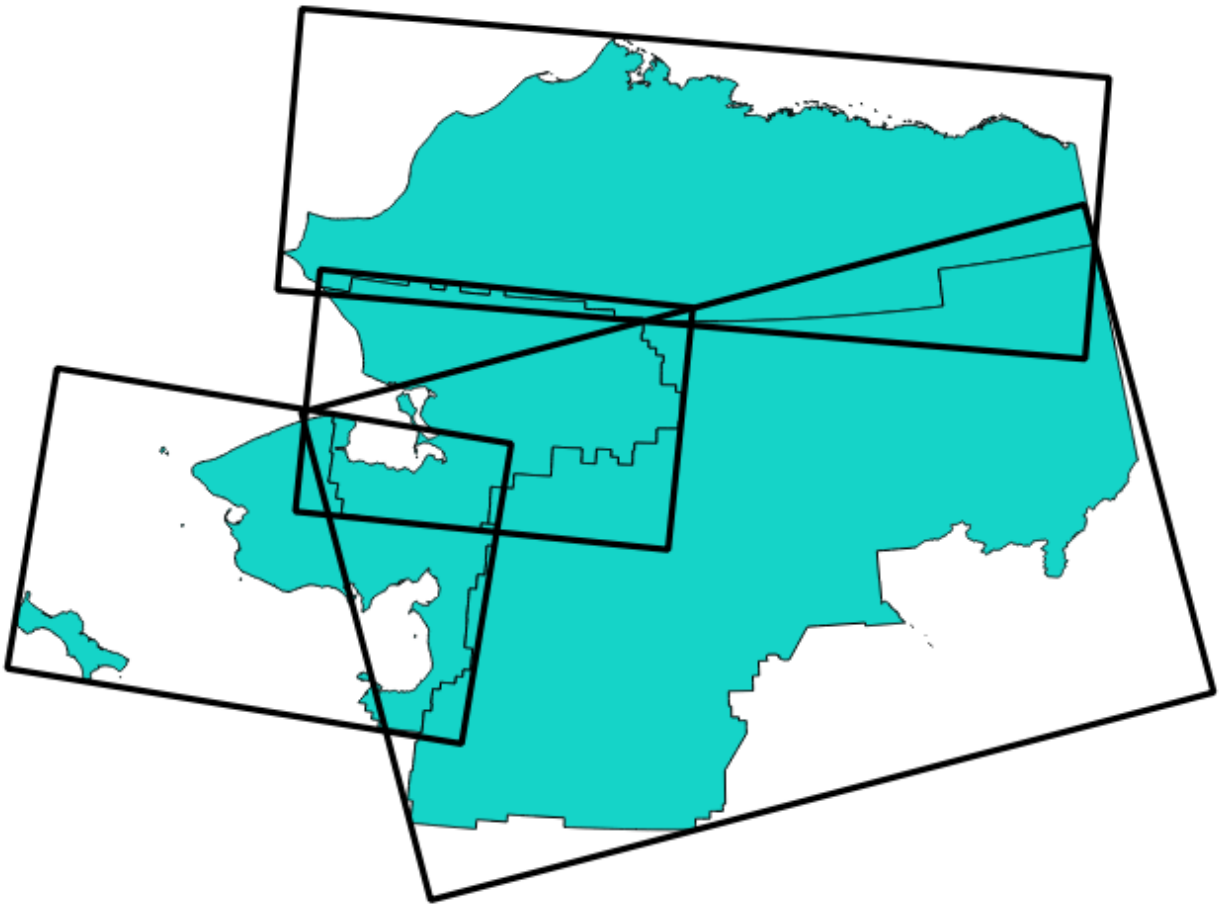


그림 24.71: 기울어진 최소 경계 상자

파라미터

라벨	명칭	유형	설명
Input layer	INPUT	[vector: any]	입력 벡터 레이어
Bounding boxes	OUTPUT	[vector: polygon] 기본값: [Create temporary layer]	산출 폴리곤 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (TEMPORARY_OUTPUT) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Bounding boxes	OUTPUT	[vector: polygon]	산출 폴리곤 벡터 레이어

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 **명칭** 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.

제자리 피처 수정 옵션을 사용할 수 있습니다.

파라미터

라벨	명칭	유형	설명
Input layer	INPUT	[vector: line, polygon]	입력 라인 또는 폴리곤 벡터 레이어

다음 페이지에 계속

표 24.110 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Maximum angle tolerance (degrees)	ANGLE_TOLERANCE	[number] 기본값: 15	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	[number] 기본값: 1000	최대 반복 횟수를 크게 설정할수록 처리 시간은 더 걸리지만 더 직교 도형에 가까운 도형을 산출할 것입니다.
Orthogonalized	OUTPUT	[same as input] 기본값: [Create temporary layer]	산출 폴리곤 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (TEMPORARY_OUTPUT) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

제자리 피처 수정 옵션을 사용할 수 있습니다.

더 보기:

중심

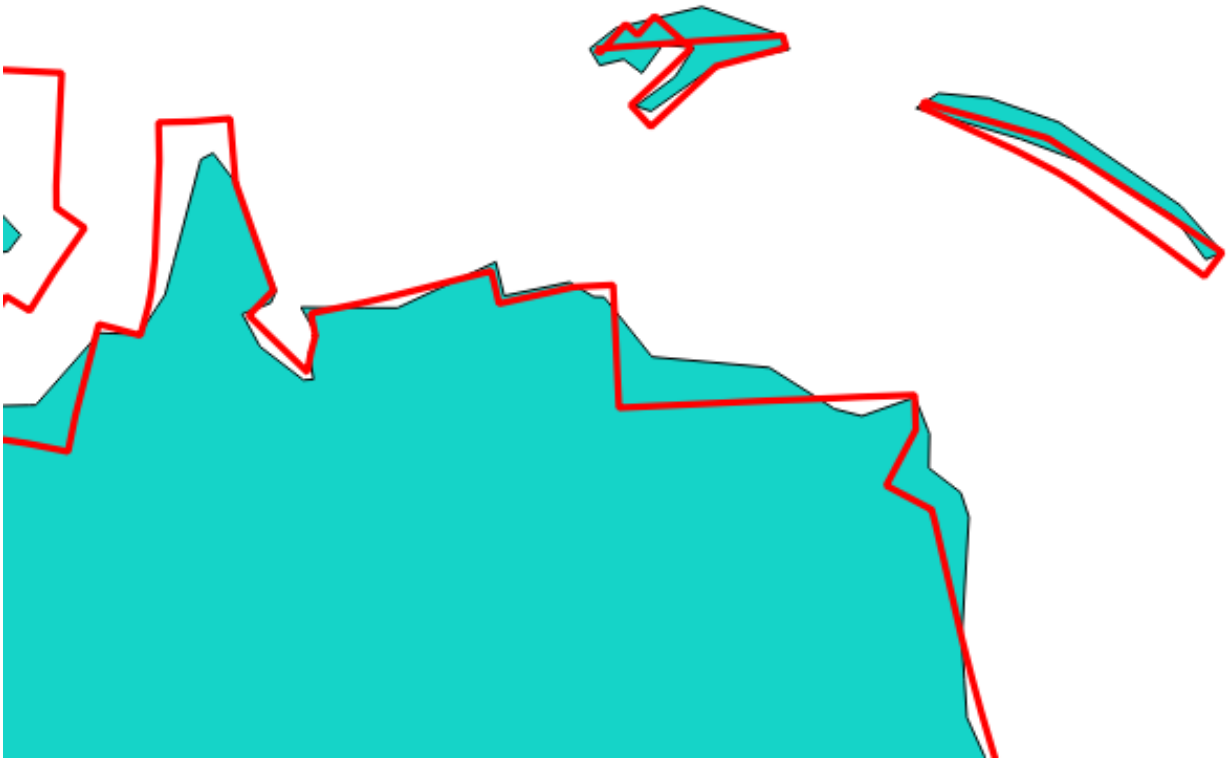


그림 24.72: 파란색: 소스 레이어, 빨간색: 직교화된 산출물

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	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]`	산출 포인트 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

더 보기:

라인에 포인트를 보간하기

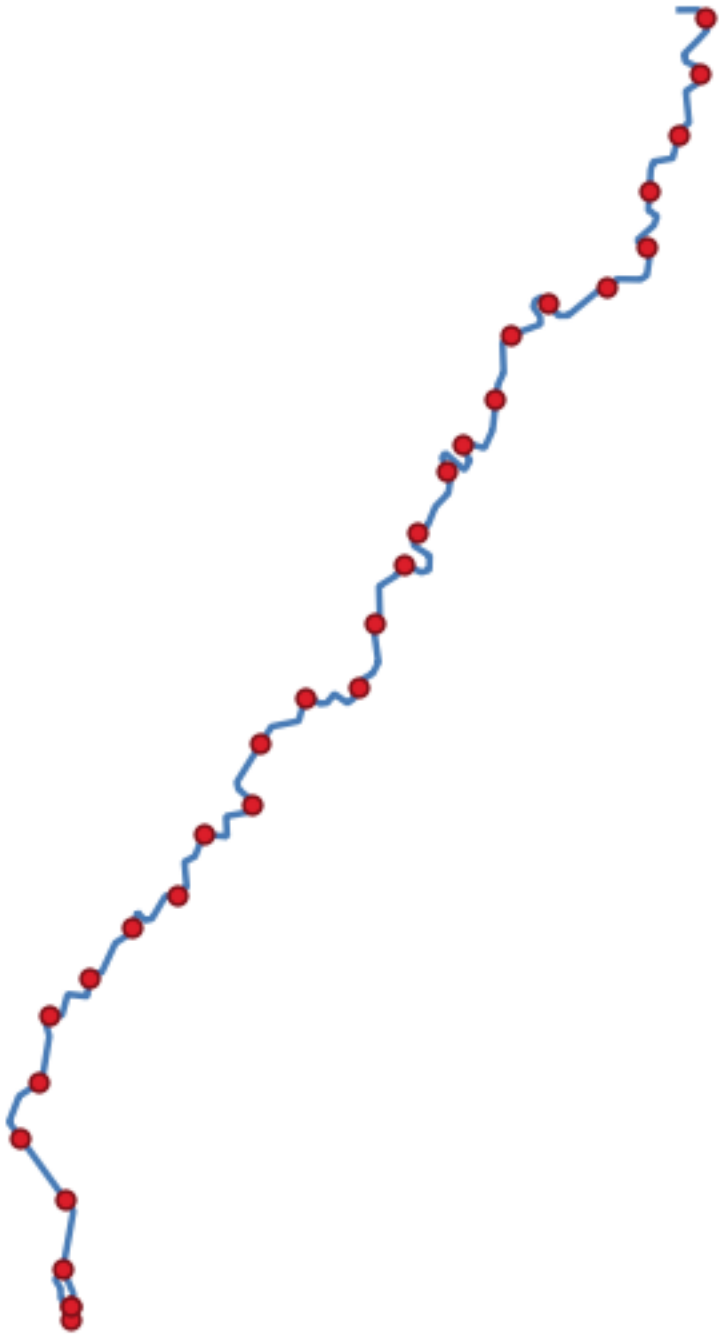


그림 24.73: 소스 라인 레이어를 따라 생성된 포인트

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: line, polygon]	입력 라인 또는 폴리곤 벡터 레이어
Distance	`DISTANCE`	[number] 기본값: 1.0	라인을 따라 연속된 두 포인트 사이의 거리
Start offset	`START_OFFSET`	[number] 기본값: 0.0	첫 번째 포인트의 위치를 나타내는 입력 라인 시작점에서의 거리
End offset	`END_OFFSET`	[number] 기본값: 0.0	더 이상 포인트 피처를 생성해서는 안 되는 위치를 나타내는 입력 라인 종단점에서의 거리
Interpolated points	`OUTPUT`	[vector: point] 기본값: `[Create temporary layer]`	산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Interpolated points	`OUTPUT`	[vector: point]	입력 레이어의 라인 또는 폴리곤 경계를 따라 배치된 피처를 담은 포인트 벡터 레이어

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 **명칭** 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	명칭	유형	설명
Input layer	`INPUT`	[vector: point]	입력 포인트 벡터 레이어
Minimum distance to other points	`PROXIMITY`	[number] 기본값: 1.0	이 거리 미만의 포인트 피처들을 근접하다고 판단합니다. 근접한 피처들을 함께 분포시킵니다.
Displacement distance	`DISTANCE`	[number] 기본값: 1.0	근접한 피처들을 배치할 원의 반경
Horizontal distribution for two point case	`HORIZONTAL`	[boolean] 기본값: False	단 2 개의 포인트만 근접해 있다고 식별된 경우, 원 위에 수직이 아닌 수평으로 정렬합니다.
Displaced	`OUTPUT`	[vector: point] 기본값: `[Create temporary layer]`	산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Displaced	`OUTPUT`	[vector: point]	입력 포인트 벡터 레이어

파이썬 코드

Algorithm 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 **명칭** 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.

계산된 도달불능극에서 폴리곤 경계선까지의 거리를 산출 레이어의 새 속성으로 저장할 것입니다.



그림 24.74: 도달불능극

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: polygon]	입력 벡터 레이어
Tolerance	`TOLERANCE`	[number] 기본값: 1.0	계산을 위한 허용 오차를 설정합니다.
Point	`OUTPUT`	[vector: point] 기본값: `[Create temporary layer]`	산출 폴리곤 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Point	OUTPUT	[vector: point]	산출 포인트 벡터 레이어

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 **명칭** 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.

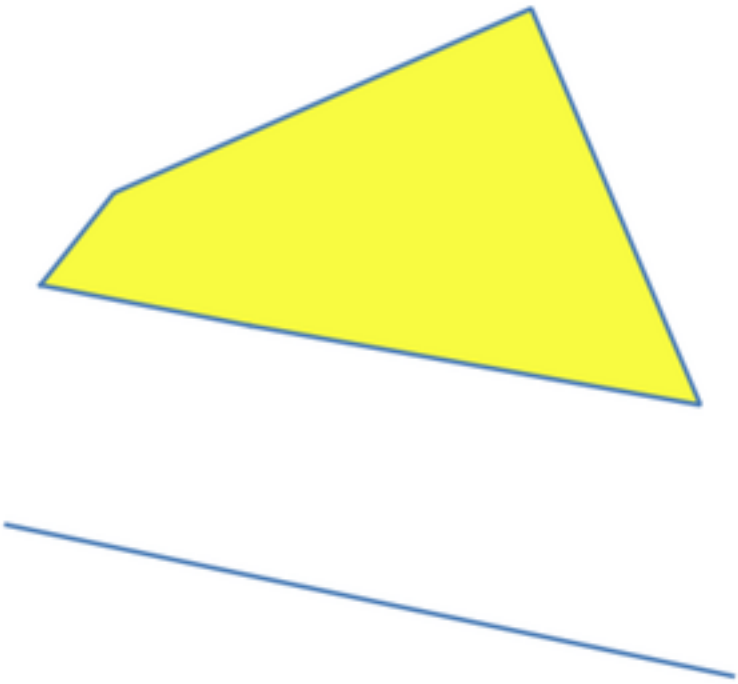


그림 24.75: 닫힌 라인으로부터 생성된 노란색 폴리곤

참고: 폴리곤으로 변형되기 위해서는 라인 레이어가 닫힌 도형을 가지고 있어야만 합니다.

더 보기:

폴리곤을 라인으로, 라인을 폴리곤으로, 도형 유형 변환하기

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: line]	입력 라인 벡터 레이어
Keep table structure of line layer 부가적	`KEEP_FIELDS`	[boolean] 기본값: False	입력 레이어의 필드를 (값이 아닌 테이블 구조만) 유지하려면 체크하십시오.
Polygons from lines	`OUTPUT`	[vector: polygon] 기본값: `[Create temporary layer]`	산출 폴리곤 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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 24.76: 알고리즘이 산출한 검은색 라인

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: polygon]	입력 폴리곤 벡터 레이어
Lines	`OUTPUT`	[vector: line] 기본값: `[Create temporary layer]`	산출 라인 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

제자리 피처 수정 옵션을 사용할 수 있습니다.

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: point]	입력 포인트 벡터 레이어
Bearing (degrees from North)	`BEARING`	[number] 기본값: 0.0	북쪽에 시작하는 시계 방향 도 (°) 단위 각도
Distance	`DISTANCE`	[number] 기본값: 1.0	도형을 오프셋시킬 레이어 단위 거리
Projected	`OUTPUT`	[vector: point] 기본값: `[Create temporary layer]`	산출 포인트 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

☑ 제자리 피처 수정 옵션을 사용할 수 있습니다.

더 보기:

집계, 도형 모으기

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	입력 벡터 레이어
Multiparts	`OUTPUT`	[same as input] 기본값: `[Create temporary layer]`	산출 다중 부분 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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 24.77: 동적 파라미터를 가진 서로 다른 버퍼 형태

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: point]	입력 포인트 벡터 레이어
Buffer shape	`SHAPE`	[enumeration]	사용할 형태를 지정합니다. 다음 가운데 하나를 선택할 수 있습니다: • 0 —직사각형 • 1 —타원형 (oval) • 2 —마름모꼴
Width	`WIDTH`	[number] 기본값: 1.0	버퍼 형태의 너비
Height	`HEIGHT`	[number] 기본값: 1.0	버퍼 형태의 높이
Rotation 부가적	`ROTATION`	[number] 기본값: None	버퍼 형태의 기울기
Number of segment	`SEGMENTS`	[number] 기본값: 36	완전한 원을 위한 선분의 개수 (타원형 형태)
산출물	`OUTPUT`	[vector: polygon] 기본값: `[Create temporary layer]`	산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
산출물	`OUTPUT`	[vector: polygon]	버퍼 형태를 가진 산출 벡터 레이어

파이썬 코드

알고리즘 ID: `native:rectanglesovalsdiamonds`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

복제 꼭짓점 제거하기

꼭짓점 제거가 도형의 유효성을 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.

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.

제자리 피처 수정 옵션을 사용할 수 있습니다.

더 보기:

꼭짓점 추출하기, 특정 꼭짓점 추출하기, 복제 도형 삭제하기

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	입력 벡터 레이어
Tolerance	`TOLERANCE`	[number] 기본값: 0.000001	지정한 거리보다 가까이 있는 꼭짓점들을 복제 꼭짓점으로 간주합니다.
Use Z value	`USE_Z_VALUE`	[boolean] 기본값: False	*Use Z Value* 파라미터를 참으로 설정한 경우, Z 값도 검사해서 X 및 Y 좌표가 동일하지만 Z 값은 다른 꼭짓점들을 그대로 남길 것입니다.

다음 페이지에 계속

표 24.115 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Cleaned	OUTPUT	[same as input] 기본값: [Create temporary layer]	산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (TEMPORARY_OUTPUT) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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	명칭	유형	설명
Input layer	INPUT	[vector: any]	NULL 이 아닌 도형을 가진 입력 벡터 레이어
Also remove empty geometries	REMOVE_EMPTY	[boolean]	

다음 페이지에 계속

표 24.116 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Non null geometries	`OUTPUT`	[same as input] 기본값: `[Create temporary layer]`	NULL 이 아닌 (그리고 비어 있지도 않은) 도형을 저장할 산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.
Null geometries	`NULL_OUTPUT`	[same as input] 기본값: `[Skip output]`	NULL (그리고 비어 있는) 도형을 저장할 산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 산출물 건너뛰기 • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

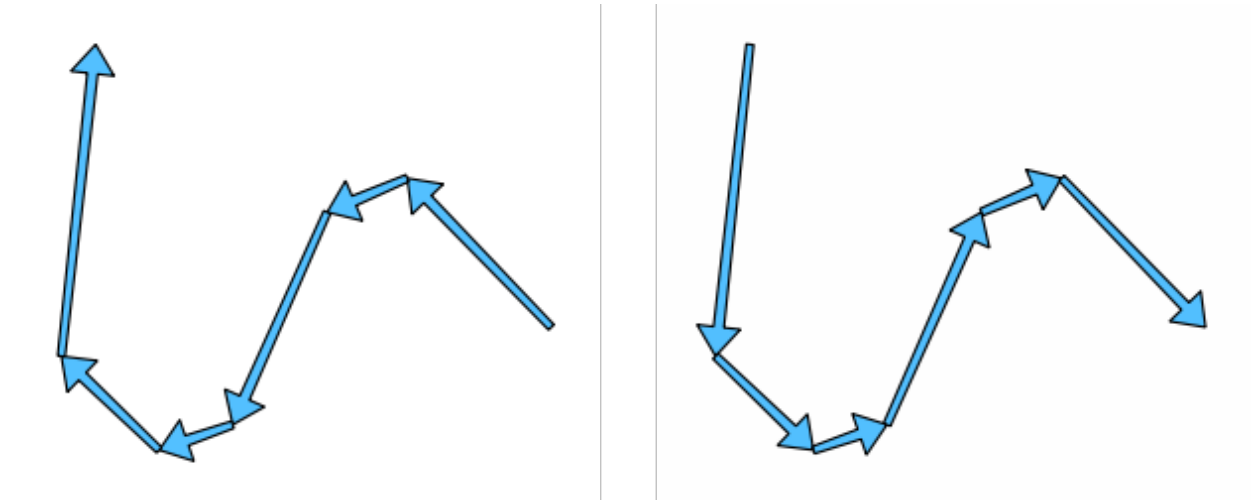


그림 24.78: 방향 반전 이전과 이후

제자리 피처 수정 옵션을 사용할 수 있습니다.

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: line]	입력 라인 벡터 레이어
Reversed	`OUTPUT`	[vector: line] 기본값: `[Create temporary layer]`	산출 라인 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

제자리 피처 수정 옵션을 사용할 수 있습니다.

더 보기:

이동시키기 *(Translate)*, *X* 좌표와 *Y* 좌표 바꾸기

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	입력 벡터 레이어
Rotation (degrees clockwise)	`ANGLE`	[number] 기본값: 0.0	기울기의 도 단위 각도
Rotation anchor point (x, y) 부가적	`ANCHOR`	[point] 기본값: None	피처 기울이기의 기준이 될 포인트의 X, Y 좌표. 이 파라미터를 설정하지 않는 경우 각 피처의 중심을 기준으로 기울입니다.
Rotated	`OUTPUT`	[same as input] 기본값: `[Create temporary layer]`	기울인 도형을 담을 산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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	명칭	유형	설명
Input layer	`INPUT`	[vector: line, polygon]	입력 라인 또는 폴리곤 벡터 레이어
Maximum angle between vertices (degrees)	`ANGLE`	[number ] 기본값: 5.0	직선화된 도형에 있는 연속된 두 꼭짓점 사이의 최대 허용 반경 각도
Segmentized	`OUTPUT`	[same as input] 기본값: `[Create temporary layer]`	선분화된 도형을 담을 산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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	명칭	유형	설명
Input layer	`INPUT`	[vector: line, polygon]	입력 라인 또는 폴리곤 벡터 레이어
Maximum offset distance	`DISTANCE`	[number] 기본값: 1.0	원본 곡선과 선분화된 표현 사이의 최대 허용 오프셋 거리를 레이어 단위로 지정합니다.
Segmentized	`OUTPUT`	[same as input] 기본값: `[Create temporary layer]`	선분화된 도형을 담을 산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Segmentized	`OUTPUT`	[same as input]	선분화된 도형을 담은 산출 벡터 레이어

파이썬 코드

알고리즘 ID: `native:segmentizebymaxdistance`

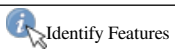
```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 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.

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	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	입력 벡터 레이어
M Value	`M_VALUE`	[number] 기본값: 0.0	피처 도형에 할당할 M 값
M Added	`OUTPUT`	[same as input] 기본값: `[Create temporary layer]`	산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

더 보기:

늘어뜨리기 *(*래스터로부터 *Z* 값 설정하기*)*, *M* 값 설정하기

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	입력 벡터 레이어
Raster layer	`RASTER`	[raster]	M 값을 가진 래스터 레이어
Band number	`BAND`	[raster band] 기본값: 1	M 값을 가져올 래스터 밴드
Value for nodata or non-intersecting vertices	`NODATA`	[number] 기본값: 0.0	꼭짓점이 래스터 (의 유효한 픽셀) 와 교차하지 않는 경우 사용할 값
Scale factor	`SCALE`	[number] 기본값: 1.0	척도 값: 밴드 값을 이 값으로 곱합니다.

다음 페이지에 계속

표 24.117 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Updated	`OUTPUT`	[same as input] 기본값: `[Create temporary layer]`	업데이트된 M 값을 담을 산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

팁: 

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	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	입력 벡터 레이어
Z Value	`Z_VALUE`	[number] 기본값: 0.0	피처 도형에 할당할 Z 값
Z Added	`OUTPUT`	[same as input] 기본값: `[Create temporary layer]`	산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

제자리 피처 수정 옵션을 사용할 수 있습니다.

기본 메뉴: *Vector → Geometry Tools*

더 보기:

평탄화, 개수로 치밀화하기, 간격으로 치밀화하기



그림 24.79: 좌상단에서 시계 방향으로: 소스 레이어와 증가하는 단순화 허용 오차

파라미터

라벨	명칭	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: • 0 — 거리 (더글라스-포이커) • 1 — 그리드에 스냅 • 2 — 면적 (비쉬왈링감)
Tolerance	`TOLERANCE`	[number] 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]`	산출 (단순화된) 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Simplified	`OUTPUT`	[same as input]	산출 (단순화된) 벡터 레이어

파이썬 코드

알고리즘 ID: `native:simplifygeometries`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

한쪽 버퍼 생성하기

라인의 한쪽에만 지정된 거리로 버퍼를 계산합니다.
버퍼는 언제나 폴리곤 레이어로 산출됩니다.



그림 24.80: 동일 라인 벡터 레이어의 왼쪽 대 오른쪽 버퍼

더 보기:

버퍼

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: line]	입력 라인 벡터 레이어
Distance	`DISTANCE`	[number] 기본값: 10.0	버퍼 거리
Side	`SIDE`	[enumeration]	버퍼를 생성할 쪽을 지정합니다. 다음 가운데 하나를 선택할 수 있습니다: • 0 –왼쪽 • 1 –오른쪽
Segments	`SEGMENTS`	[number] 기본값: 8	둥글린 (rounded) 오프셋을 생성하는 경우 사분원에 가깝게 만드는 데 사용할 라인 선분의 개수를 제어합니다.
Join style	`JOIN_STYLE`	[enumeration]	라인에 있는 모서리를 오프셋시킬 때 둥글림, 마이터, 또는 베벨 결합을 사용할지 여부를 지정합니다. 다음 옵션 가운데 하나를 선택할 수 있습니다: • 0 —둥글림 (Round) • 1 —마이터 (Miter) • 2 —베벨 (Bevel)

다음 페이지에 계속

표 24.119 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Miter limit	MITER_LIMIT	[number] 기본값: 2.0	마이터 결합 스타일에만 적용할 수 있으며, 마이터 결합 생성시 사용할 오프셋 곡선으로부터의 최장 거리를 제어합니다. (최소값은 1.0 입니다.)
Buffer	OUTPUT	[vector: polygon] 기본값: [Create temporary layer]	산출 (버퍼) 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (TEMPORARY_OUTPUT) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

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.

제자리 피처 수정 옵션을 사용할 수 있습니다.

더 보기:

단순화, 개수로 치밀화하기, 간격으로 치밀화하기



그림 24.81: 반복 횟수를 늘릴수록 더 매끄러워지는 도형

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: line, polygon]	입력 라인 또는 폴리곤 벡터 레이어
Iterations	`ITERATIONS`	[number] 기본값: 1	반복 횟수를 늘리면 더 매끄러운 도형을 (더 많은 꼭짓점과 함께) 산출합니다.
Offset	`OFFSET`	[number] 기본값: 0.25	이 값을 늘리면 입력 라인/경계선으로부터 평탄화된 라인/경계선을 더 멀리 이동시킬 것입니다.
Maximum node angle to smooth	`MAX_ANGLE`	[number] 기본값: 180.0	이 각도 값 미만의 노드를 전부 평탄화할 것입니다.
Smoothed	`OUTPUT`	[same as input] 기본값: `[Create temporary layer]`	산출 (평탄화) 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

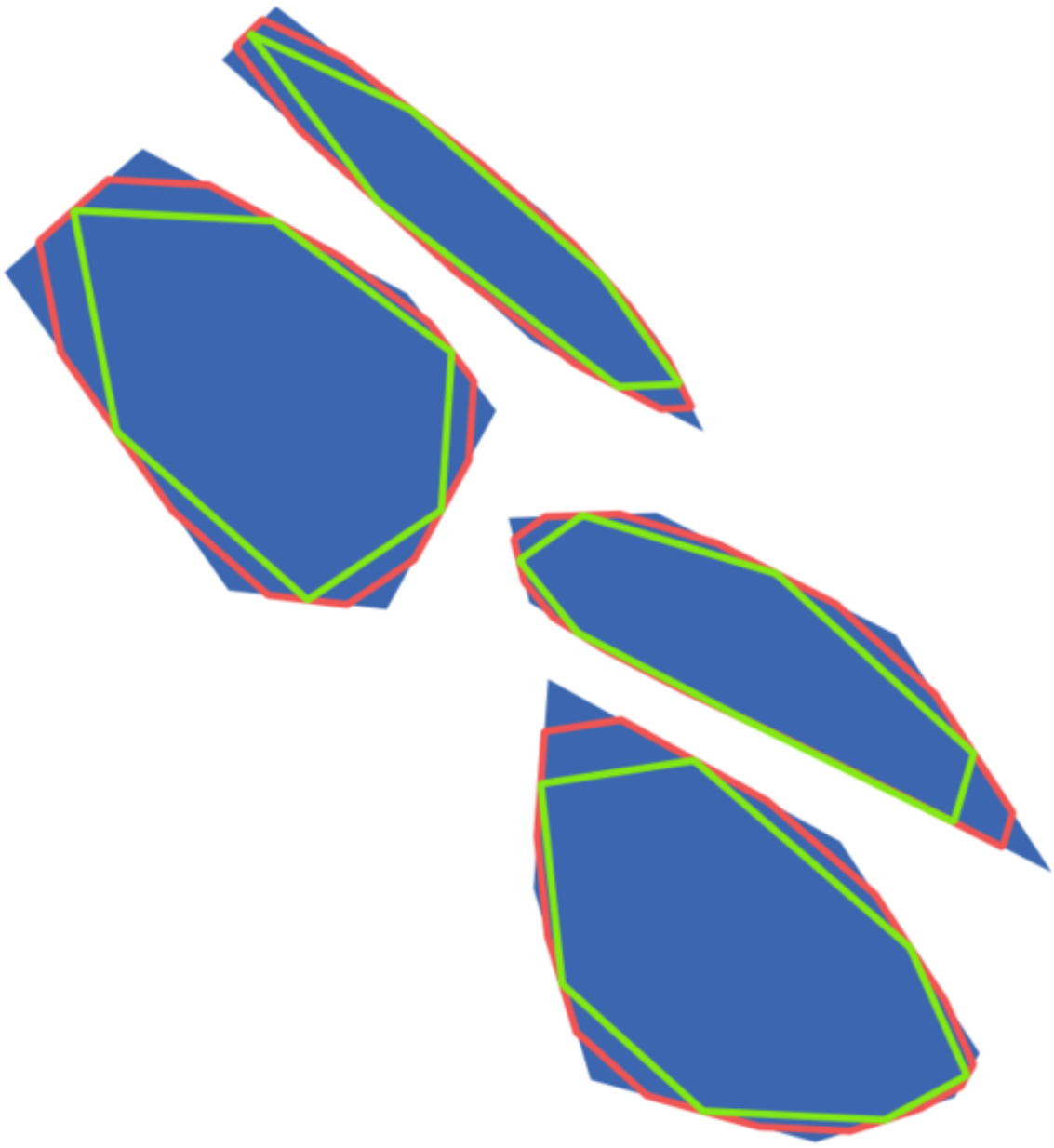


그림 24.82: 파란색은 입력 레이어, 빨간 라인은 오프셋 값 0.25, 녹색 라인은 오프셋 값 0.50

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.

더 보기:

그리드에 포인트를 스냅시키기

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	입력 벡터 레이어
Reference layer	`REFERENCE_LAYER`	[vector: any]	스냅시킬 벡터 레이어
Tolerance	`TOLERANCE`	[number] 기본값: 10.0	입력 꼭짓점을 스냅시키려면 참조 레이어의 도형에 얼마나 가까워야 하는지 제어합니다.

다음 페이지에 계속

표 24.121 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Behavior	BEHAVIOR	[enumeration] 기본값: 0	기존 노드 또는 (이동시킬 꼭짓점에 가장 근접한 포인트를 가진) 선분에 스냅시킬 수 있습니다. 다음 스냅 옵션 가운데 선택할 수 있습니다: • 0 —노드를 정렬시키는 편을 선호, 필요한 위치에 추가 꼭짓점을 삽입 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. • 1 —최근접 포인트를 선호, 필요한 위치에 추가 꼭짓점을 삽입 노드인지 선분인지 상관없이 가장 가까운 포인트에 스냅합니다. 허용 가능한 오차 범위 내에 있을 경우 도형들이 서로를 정확히 따르게 만들기 위해 새 노드를 삽입할 것입니다. • 2 —노드를 정렬시키는 편을 선호, 새 꼭짓점 삽입 안 함 선분이 노드보다 가까이 있어도 노드에 스냅하는 편을 선호합니다. 새 노드를 삽입하지 않을 것입니다. • 3 —최근접 포인트를 선호, 새 꼭짓점 삽입 안 함 노드인지 선분인지 상관없이 가장 가까운 포인트에 스냅합니다. 새 노드를 삽입하지 않을 것입니다. • 4 —종단 포인트만 이동, 노드 정렬 선호 라인의 시작/종단 포인트에만 스냅합니다. (포인트 피처에도 스냅할 것이고, 폴리곤 피처는 수정되지 않을 것입니다.) 노드에 스냅하는 편을 선호합니다. • 5 —종단 포인트만 이동, 최근접 포인트 선호 라인의 시작/종단 포인트에만 스냅합니다. (포인트 피처에도 스냅할 것이고, 폴리곤 피처는 수정되지 않을 것입니다.) 가장 가까운 포인트에 스냅합니다. • 6 —종단 포인트와 종단 포인트만 스냅 라인의 시작/종단 포인트를 다른 라인의 시작/종단 포인트에만 스냅합니다. • 7 —고정 노드 (anchor node) 에 스냅 (단일 레이어에서만)

다음 페이지에 계속

표 24.121 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Snapped geometry	OUTPUT	[same as input] 기본값: [Create temporary layer]	산출 (스냅된) 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (TEMPORARY_OUTPUT) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

참고: 그리드에 스냅시키면 일부 모서리에서 유효하지 않은 도형을 생성할 수도 있습니다.

제자리 피처 수정 옵션을 사용할 수 있습니다.

더 보기:

레이어에 도형을 스냅시키기

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	입력 벡터 레이어
X Grid Spacing	`HSPACING`	[number] 기본값: 1.0	X 축 상의 그리드 간격
Y Grid Spacing	`VSPACING`	[number] 기본값: 1.0	Y 축 상의 그리드 간격
Z Grid Spacing	`ZSPACING`	[number] 기본값: 0.0	Z 축 상의 그리드 간격
M Grid Spacing	`MSPACING`	[number] 기본값: 0.0	M 축 상의 그리드 간격
Snapped	`OUTPUT`	[same as input] 기본값: `[Create temporary layer]`	산출 (스냅된) 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: line]	입력 라인 벡터 레이어
Maximum line length	`LENGTH`	[number] 기본값: 10.0	산출 라인의 최장 길이
Split	`OUTPUT`	[vector: line] 기본값: `[Create temporary layer]`	산출 라인 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Split	`OUTPUT`	[vector: line]	새 라인 벡터 레이어 — 피처 도형의 길이는 LENGTH 파라미터에서 지정한 길이 이하입니다.

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 **명칭** 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.

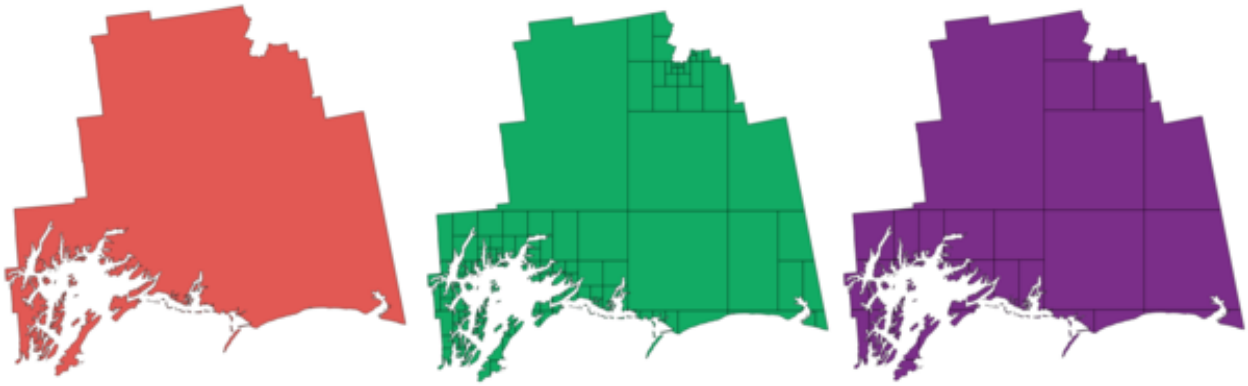


그림 24.83: 왼쪽: 입력 레이어, 중앙: 최대 노드 값 100, 오른쪽: 최대 노드 값 200

참고: 도형을 세분화하면 유효하지 않거나 자체 교차하는 도형 부분들을 생성할 수도 있습니다.

제자리 피처 수정 옵션을 사용할 수 있습니다.

더 보기:

라인 조각내기, 라인 부스트링 생성하기

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	입력 벡터 레이어
Maximum nodes in parts	`MAX_NODES`	[number] 기본값: 256	각 새 도형 부분이 가질 수 있는 꼭짓점의 최대 개수입니다. 값이 높을수록 하위 부분의 개수가 적어집니다.
Subdivided	`OUTPUT`	[same as input] 기본값: `[Create temporary layer]`	산출 (세분화된) 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

제자리 피처 수정 옵션을 사용할 수 있습니다.

더 보기:

이동시키기 *(Translate)*, 기울이기

파라미터

라벨	명칭	유형	설명
Input layer	INPUT	[vector: any]	입력 벡터 레이어
Swapped	OUTPUT	[same as input] 기본값: [Create temporary layer]	산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (TEMPORARY_OUTPUT) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

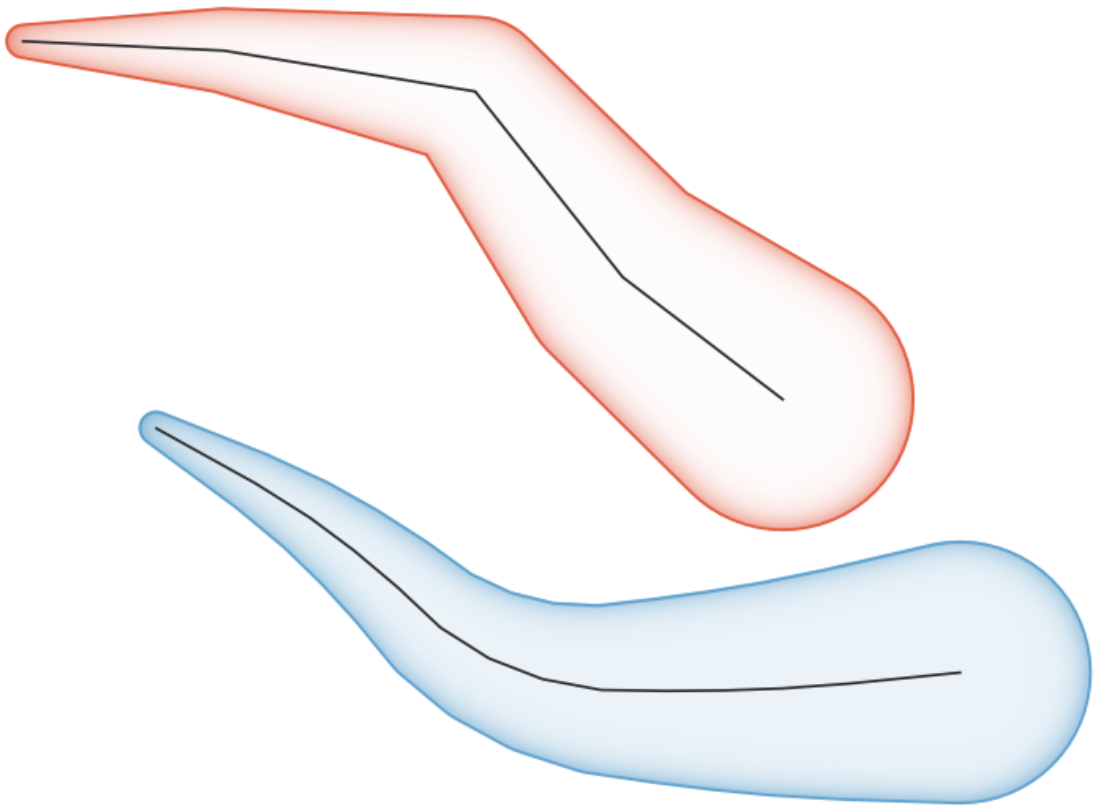


그림 24.84: 줄어드는 버퍼 예시

더 보기:

변동 너비 버퍼 *(M 값으로)*, 버퍼, 쐐기 버퍼 생성하기

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: line]	입력 라인 벡터 레이어
Start width	`START_WIDTH`	[number] 기본값: 0.0	라인 피처의 시작 포인트에 적용되는 버퍼 반경을 설정합니다.
End width	`END_WIDTH`	[number] 기본값: 0.0	라인 피처의 종단 포인트에 적용되는 버퍼 반경을 설정합니다.
Segments	`SEGMENTS`	[number] 기본값: 16	둥글린 (rounded) 오프셋을 생성하는 경우 사분원에 가깝게 만드는 데 사용할 라인 선분의 개수를 제어합니다.
Buffered	`OUTPUT`	[vector: polygon] 기본값: `[Create temporary layer]`	산출 (버퍼) 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Buffered	`OUTPUT`	[vector: polygon]	산출 (버퍼) 폴리곤 레이어

파이썬 코드

알고리즘 ID: `native:taperedbuffer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

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.

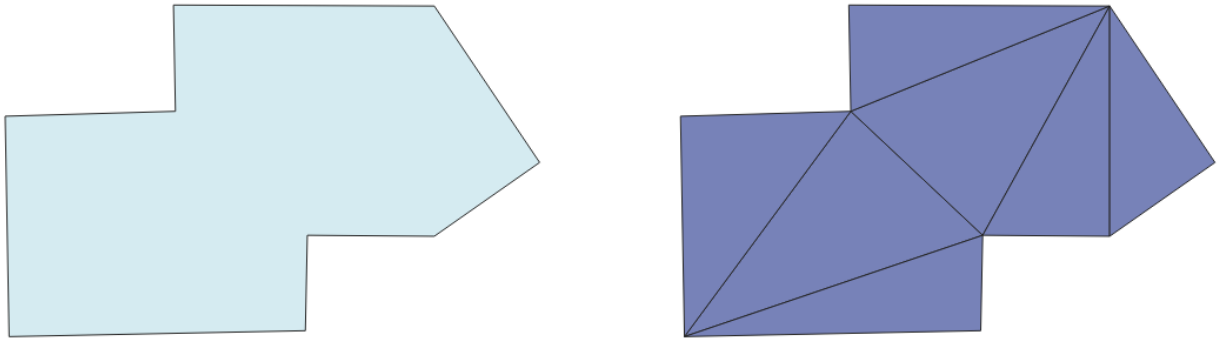


그림 24.85: 모자이크화된 폴리곤 (오른쪽)

제자리 피처 수정 옵션을 사용할 수 있습니다.

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: polygon]	입력 폴리곤 벡터 레이어
Tesselated	`OUTPUT`	[vector: polygon] 기본값: `[Create temporary layer]`	산출 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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)

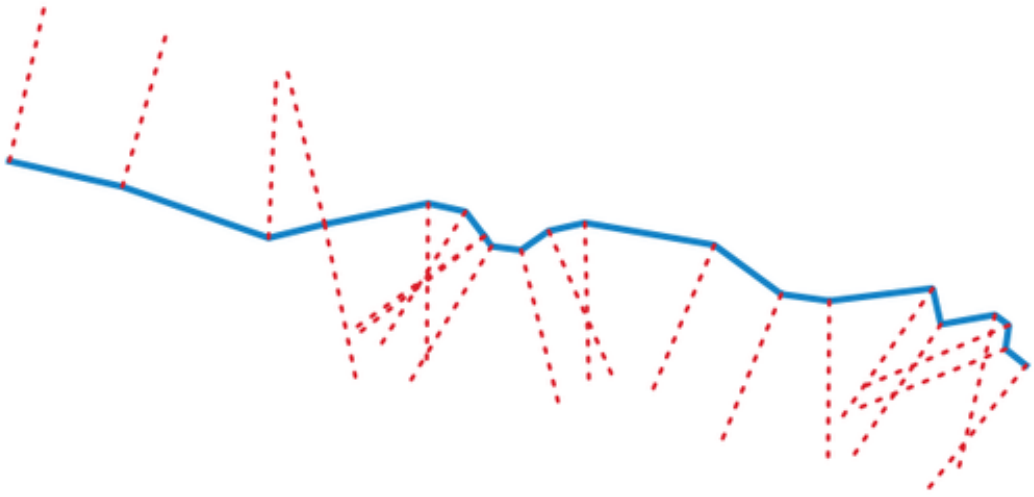


그림 24.86: 입력 라인 레이어의 절개선을 나타내는 빨간 점선

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: line]	입력 라인 벡터 레이어
Length of the transect	`LENGTH`	[number] 기본값: 5.0	절개선의 맵 단위 길이
Angle in degrees from the original line at the vertices	`ANGLE`	[number] 기본값: 90.0	절개선의 각도를 변경합니다.
Side to create the transect	`SIDE`	[enumeration]	절개선의 쪽을 설정합니다. 다음 옵션 가운데 선택할 수 있습니다: • 0 — 왼쪽 • 1 — 오른쪽 • 2 — 양쪽
Transect	`OUTPUT`	[vector: line] 기본값: `[Create temporary layer]`	산출 라인 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

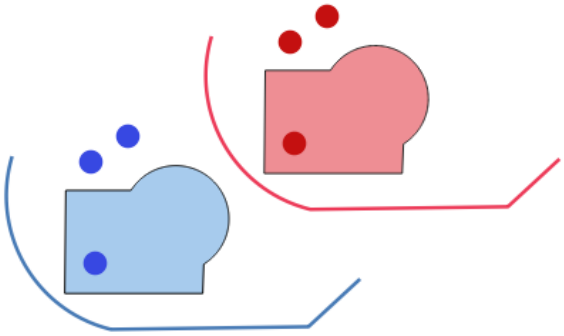


그림 24.87: 입력 레이어는 파란색, 이동된 피처를 담은 산출 레이어는 빨간색

제자리 피처 수정 옵션을 사용할 수 있습니다.

더 보기:

이동시킨 피처의 배열, 라인 오프셋시키기, 기울이기, *X* 좌표와 *Y* 좌표 바꾸기

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	입력 벡터 레이어
Offset distance (x-axis)	`DELTA_X`	[number] 기본값: 0.0	X 축에 적용할 변위 (displacement)
Offset distance (y-axis)	`DELTA_Y`	[number] 기본값: 0.0	Y 축에 적용할 변위 (displacement)
Offset distance (z-axis)	`DELTA_Z`	[number] 기본값: 0.0	Z 축에 적용할 변위 (displacement)
Offset distance (m values)	`DELTA_M`	[number] 기본값: 0.0	M 축에 적용할 변위 (displacement)
Translated	`OUTPUT`	[same as input] 기본값: `[Create temporary layer]`	산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

변동 거리 버퍼

입력 레이어의 모든 피처에 대한 버퍼 영역을 계산합니다.

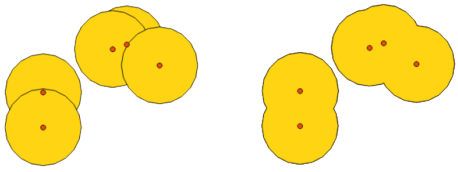
지정한 피처에 대한 버퍼 크기를 속성으로 정의합니다. 즉 서로 다른 피처가 서로 다른 버퍼 크기를 가질 수 있다는 의미입니다.

참고: 이 알고리즘은그래픽 모델 생성기 에서만 사용할 수 있습니다.

더 보기:

버퍼

파라미터

라벨	명칭	유형	설명
Input layer	INPUT	[vector: any]	입력 벡터 레이어
Distance field	DISTANCE	[tablefield: numeric]	버퍼의 거리 반경을 정의할 속성
Segments	SEGMENTS	[number] 기본값: *5*	둥글린 (rounded) 오프셋을 생성하는 경우 사분원에 가깝게 만드는 데 사용할 라인 선분의 개수를 제어합니다.
Dissolve result	DISSOLVE	[boolean] 기본값: *False*	최종 버퍼를 융해할지 여부를 선택합니다. True 로 설정하면 입력 피처를 전부 커버하는 단일 피처를 산출할 것입니다.  그림 24.88: 일반 버퍼 대 융해된 버퍼

다음 페이지에 계속

표 24.126 – 이전 페이지에서 계속

라벨	명칭	유형	설명
End cap style	`END_CAP_STYLE`	[enumeration]	버퍼에서 라인의 끝을 어떻게 처리할지 제어합니다.  그림 24.89: 둥글림, 편평 및 정사각형 캡 스타일
Join style	`JOIN_STYLE`	[enumeration]	라인에 있는 모서리를 오프셋시킬 때 둥글림, 마이터, 또는 베벨 결합을 사용할지 여부를 지정합니다.
Miter limit	`MITER_LIMIT`	[number] 기본값: 2.0	마이터 결합 스타일에만 적용할 수 있으며, 마이터 결합 생성시 사용할 오프셋 곡선으로부터의 최장 거리를 제어합니다.

산출물

라벨	명칭	유형	설명
Buffer	`OUTPUT`	[vector: polygon]	버퍼 폴리곤 벡터 레이어

파이썬 코드

Algorithm 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 **명칭** 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

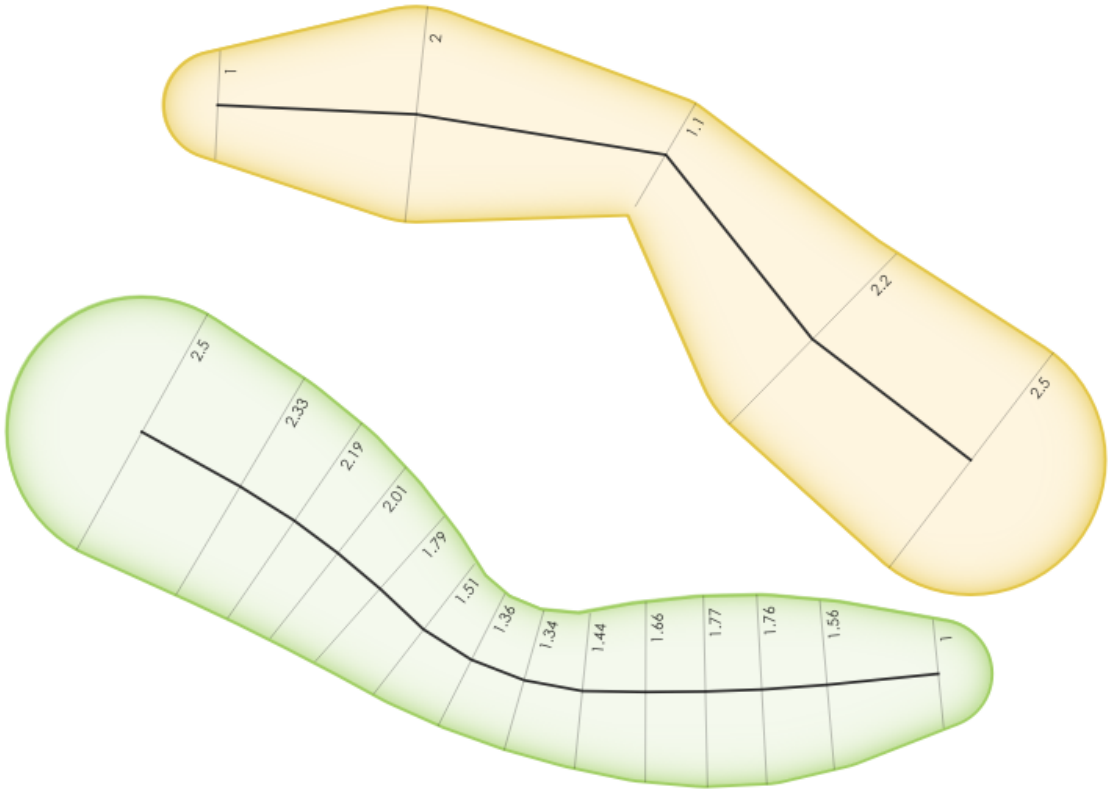


그림 24.90: 변동 버퍼 예시

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: line]	입력 라인 벡터 레이어
Segments	`SEGMENTS`	[number] 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]`	산출 (버퍼) 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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*

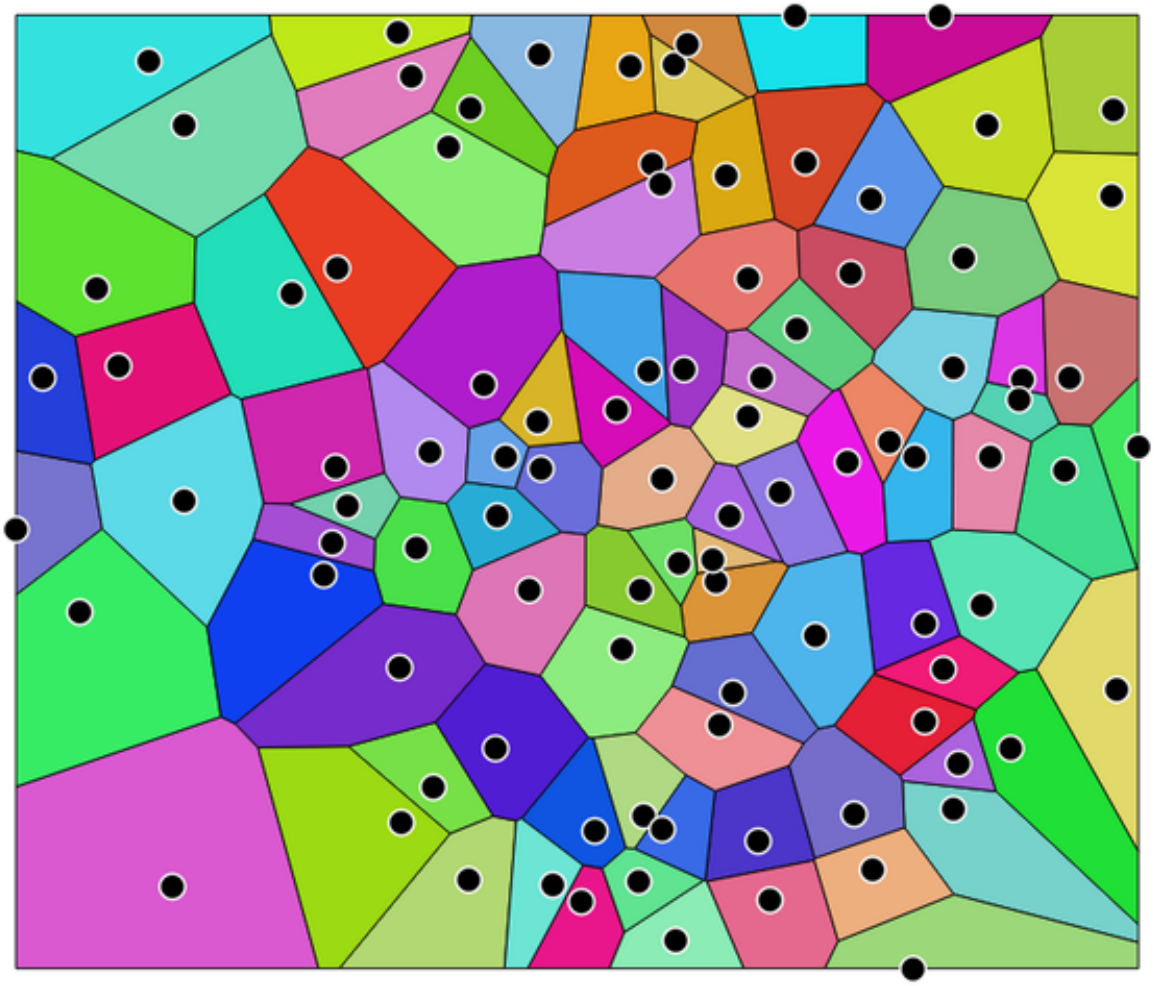


그림 24.91: 보로노이 폴리곤

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: point]	입력 포인트 벡터 레이어
Buffer region (% of extent)	`BUFFER`	[number] 기본값: 0.0	산출 레이어의 범위는 입력 레이어의 범위보다 이만큼 더 커질 것입니다.
Voronoi polygons	`OUTPUT`	[vector: polygon] 기본값: `[Create temporary layer]`	보로노이 폴리곤을 담을 산출 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Voronoi polygons	`OUTPUT`	[vector: polygon]	입력 포인트 벡터 레이어에 대한 보로노이 폴리곤

파이썬 코드

Algorithm ID: `qgis: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 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

24.1.17 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

잘라내기 작업으로 피처의 면적 또는 길이와 같은 속성 (property) 이 변경되긴 하지만, 피처의 속성 (attribute) 은 **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.

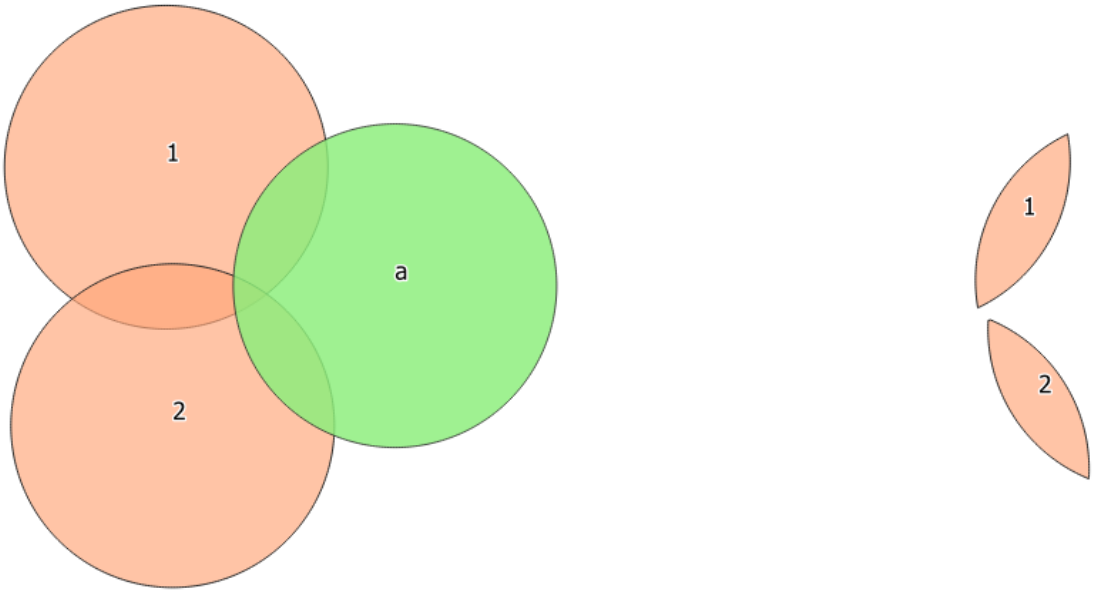


그림 24.92: 잘라내기 작업: 피처 2 개를 가진 입력 레이어와 피처 1 개를 가진 중첩 레이어 (왼쪽) 및 명확하게 보여주기 위해 이동시킨 산출 피처들 (오른쪽)

☑ 제자리 피처 수정 옵션을 사용할 수 있습니다.

기본 메뉴: *Vector → Geoprocessing Tools*

더 보기:

교차 *(Intersection)*, 차감하기 *(Difference)*

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	잘라낼 피처를 담고 있는 레이어
Overlay layer	`OVERLAY`	[vector: polygon]	잘라내기 피처를 담고 있는 레이어
Clipped	`OUTPUT`	[same as input] 기본값: `[Create temporary layer]`	중첩 (잘라내기) 레이어 안에 들어오는 입력 레이어의 피처를 담을 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Clipped	OUTPUT	[same as input]	중첩 레이어가 잘라낸 입력 레이어의 피처를 담고 있는 레이어

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 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.

속성은 수정하지 않습니다. (경고 참조)

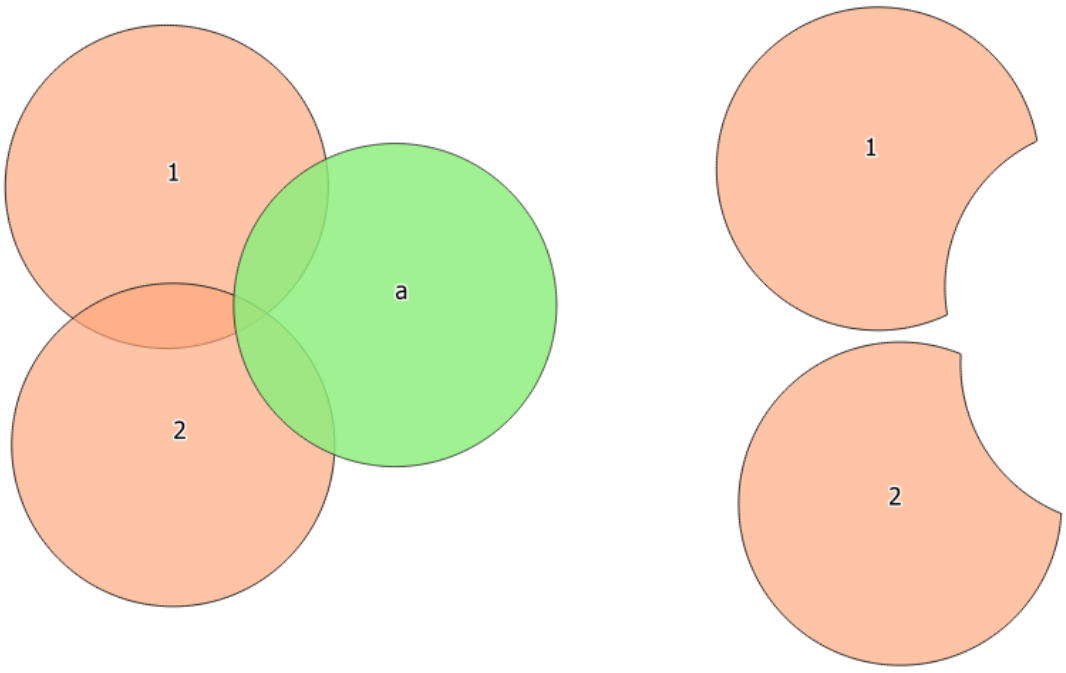


그림 24.93: 차감 작업: 피처 2 개를 가진 입력 레이어와 피처 1 개를 가진 중첩 레이어 (왼쪽) 및 명확하게 보여주기 위해 이동시킨 산출 피처들 (오른쪽)

제자리 피처 수정 옵션을 사용할 수 있습니다.

기본 메뉴: *Vector → Geoprocessing Tools*

더 보기:

대칭 차감, 잘라내기 *(Clip)*

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	피처 (의 일부분) 를 추출할 레이어
Overlay layer	`OVERLAY`	[vector: any]	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] 기본값: `[Create temporary layer]`	중첩 레이어 안에 들어오지 않는 입력 레이어의 피처 (의 일부분) 를 담을 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Difference	`OUTPUT`	[same as input]	중첩 레이어와 중첩하지 않는 입력 레이어의 피처 (의 일부분) 를 담은 레이어

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

범위로 추출/잘라내기

지정한 범위 안에 들어오는 피처만 담고 있는 새 벡터 레이어를 생성합니다.

범위와 교차하는 모든 피처를 포함할 것입니다.

더 보기:

잘라내기 *(Clip)*

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	피처 (의 일부분) 를 추출할 레이어
Extent (xmin, xmax, ymin, ymax)	`EXTENT`	[extent]	잘라내기용 범위
Clip features to extent	`CLIP`	[boolean] 기본값: False	이 파라미터를 활성화하면 산출물 유형을 통일하기 위해 산출 도형을 자동적으로 다중 도형으로 변환할 것입니다. 거기에 도형 전체를 산출하는 대신, 도형을 범위로 잘라낼 것입니다.
Extracted	`OUTPUT`	[same as input] 기본값: `[Create temporary layer]`	잘라내기 범위 안에 들어오는 입력 레이어의 피처를 담을 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Extracted	`OUTPUT`	[same as input]	잘라낸 피처를 담고 있는 레이어

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 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.
속성은 수정하지 않습니다. (경고 참조)

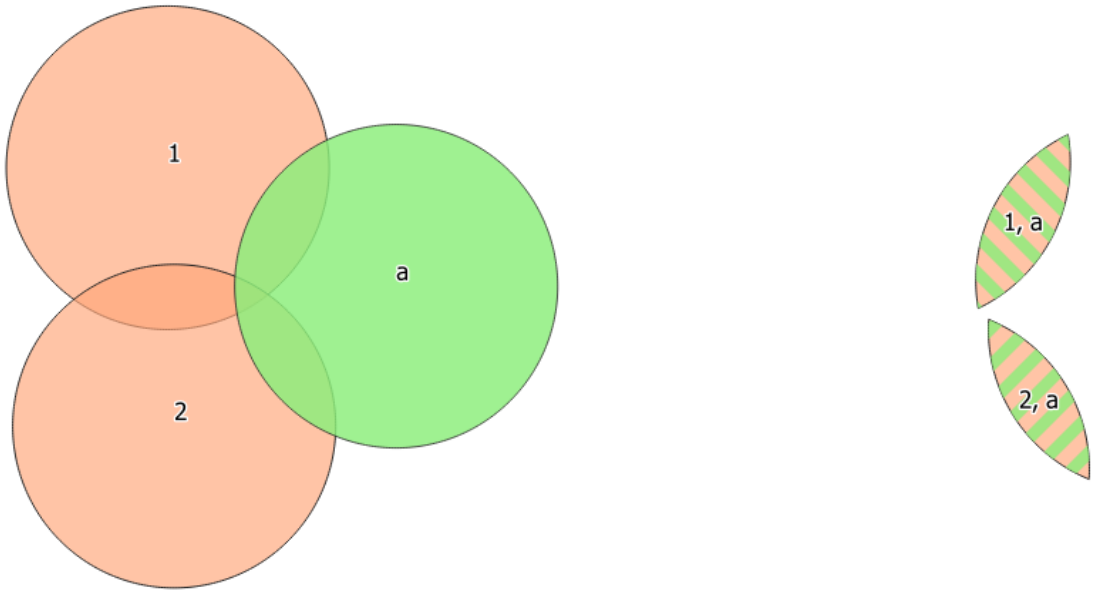


그림 24.94: 교차 작업: 피처 2 개를 가진 입력 레이어와 피처 1 개를 가진 중첩 레이어 (왼쪽) 및 명확하게 보여주기 위해 이동시킨 산출 피처들 (오른쪽)

기본 메뉴: *Vector → Geoprocessing Tools*

더 보기:

잘라내기 *(Clip)*, 차감하기 *(Difference)*

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	피처 (의 일부분) 를 추출할 레이어
Overlay layer	`OVERLAY`	[vector: any]	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) 부가적	`INPUT_FIELDS`	[tablefield: any] [list] 기본값: None	산출물에 남길 입력 레이어의 필드 (들). 아무 필드도 선택하지 않을 경우 모든 필드를 복사합니다.

다음 페이지에 계속

표 24.127 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Overlay fields to keep (leave empty to keep all fields) 부가적	`OVERLAY_FIELDS`	[tablefield: any] [list] 기본값: None	산출물에 남길 중첩 레이어의 필드 (들). 아무 필드도 선택하지 않을 경우 모든 필드를 복사합니다.
Overlay fields prefix 부가적	`OVERLAY_FIELDS_PREFIX`	[string]	입력 레이어 필드와 명칭 충돌을 피하기 위해 교차 레이어의 필드명 앞에 붙일 접두어
Intersection	`OUTPUT`	[same as input] 기본값: `[Create temporary layer]`	중첩 레이어의 하나 이상의 피처와 중첩하는 입력 레이어의 피처 (의 일부분) 를 담을 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Intersection	`OUTPUT`	[same as input]	중첩 레이어와 중첩하는 입력 레이어의 피처 (의 일부분) 를 담은 레이어

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 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.

기본 메뉴: *Vector → Analysis Tools*

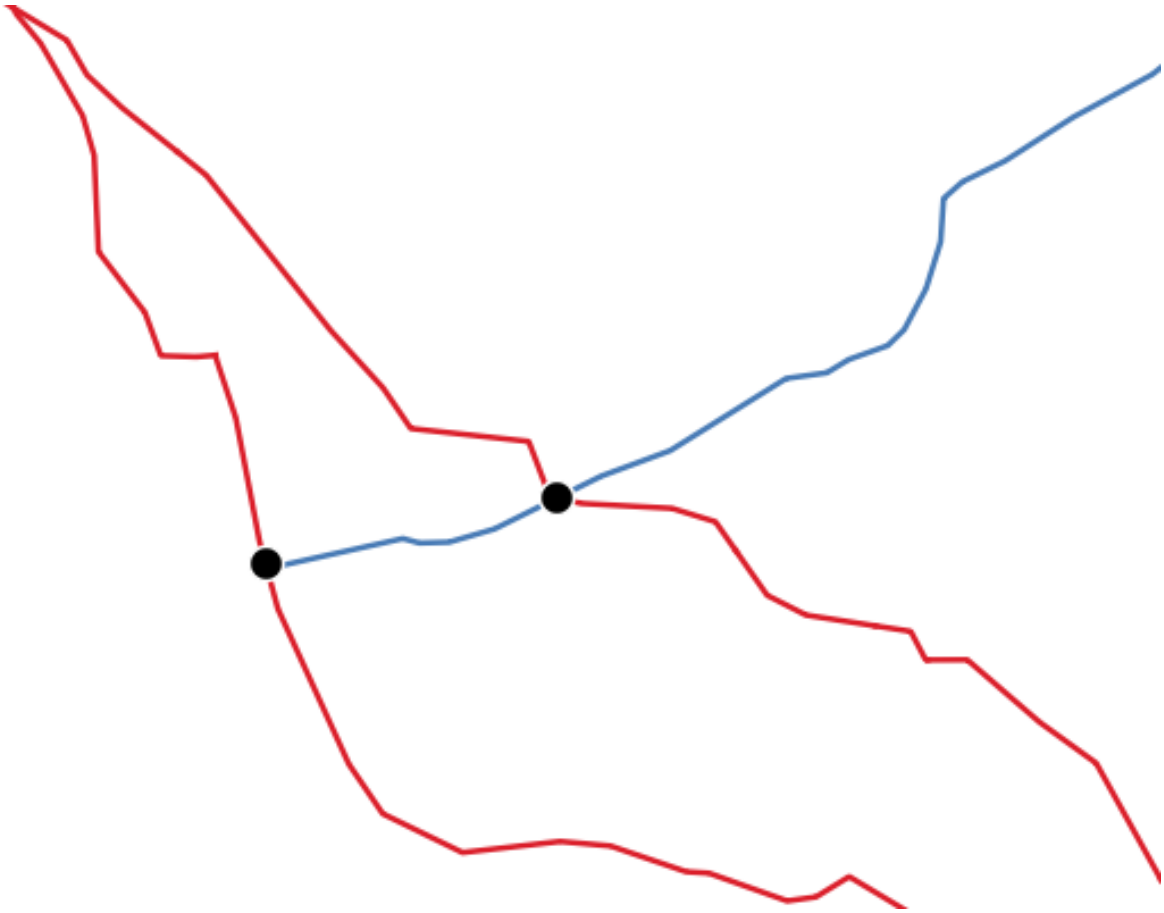


그림 24.95: 교차점 포인트

파라미터

라벨	명칭	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) 부가적	INPUT_FIELDS	[tablefield: any] [list] 기본값: None	산출물에 남길 입력 레이어의 필드 (들). 아무 필드도 선택하지 않을 경우 모든 필드를 복사합니다.
Intersect fields to keep (leave empty to keep all fields) 부가적	INTERSECT_FIELDS	[tablefield: any] [list] 기본값: None	산출물에 남길 교차 레이어의 필드 (들). 아무 필드도 선택하지 않을 경우 모든 필드를 복사합니다.
Intersect fields prefix 부가적	OVERLAY_FIELDS_PREFIX	[string]	입력 레이어 필드와 명칭 충돌을 피하기 위해 교차 레이어의 필드명 앞에 붙일 접두어
Intersection	OUTPUT	[vector: point] 기본값: [Create temporary layer]	입력 및 중첩 레이어의 라인들의 교차점 포인트를 담을 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Intersections	OUTPUT	[vector: point]	교차점 포인트를 담은 포인트 벡터 레이어

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 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 라인을 이용해서 분할합니다. 두 레이어에 있는 도형들 사이의 교차점을 분할점으로 삼습니다.

산출물은 분할된 피처를 위한 다중 도형을 담을 것입니다.

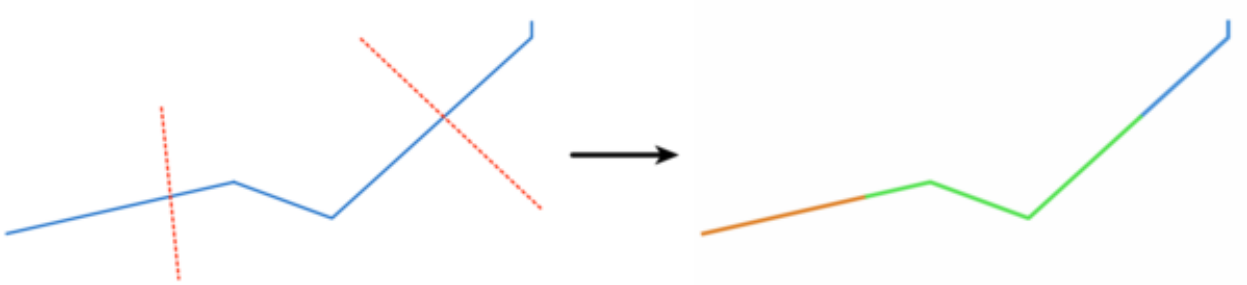


그림 24.96: 라인 분할하기

제자리 피처 수정 옵션을 사용할 수 있습니다.

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: line, polygon]	분할할 라인 또는 폴리곤을 담고 있는 레이어
Split layer	`LINES`	[vector: line]	분절점을 정의하는 데 사용되는 라인을 가진 라인 레이어
Split	`OUTPUT`	[same as input] 기본값: `[Create temporary layer]`	입력 레이어에서 분할된 (분할 레이어의 라인과 교차하는) 라인/폴리곤 피처를 담을 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Split	`OUTPUT`	[same as input]	입력 레이어에서 분할된 라인 또는 폴리곤을 가진 산출 벡터 레이어

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 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.

속성은 수정하지 않습니다. (경고 참조)

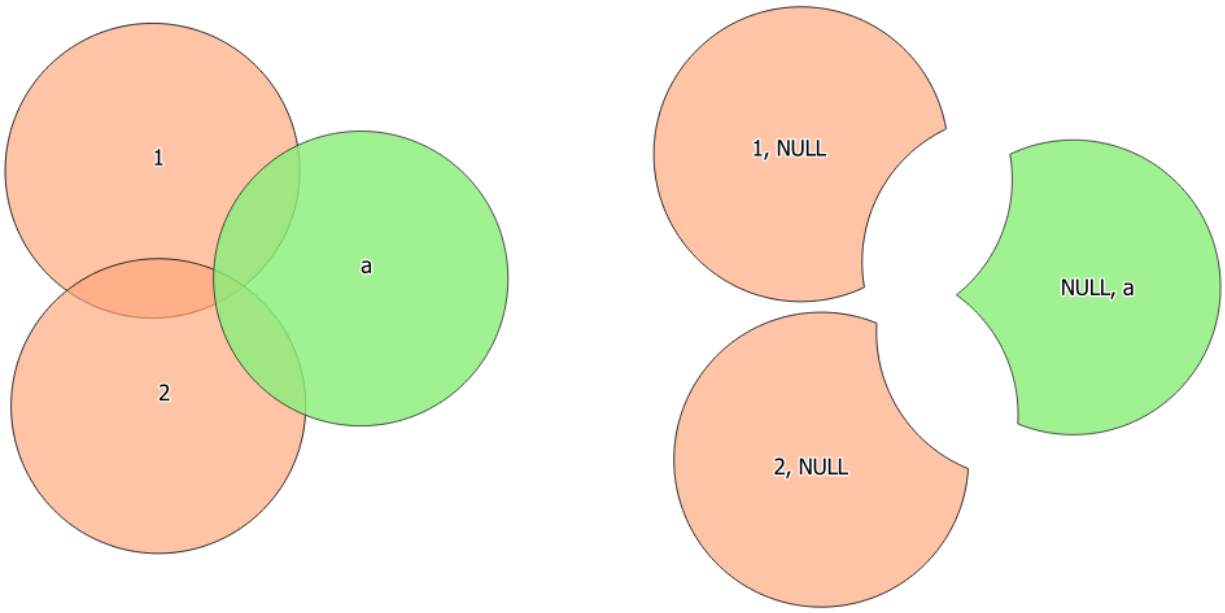


그림 24.97: 대칭 차감 작업: 피처 2 개를 가진 입력 레이어와 피처 1 개를 가진 중첩 레이어 (왼쪽) 및 명확하게 보여주기 위해 이동시킨 산출 피처들 (오른쪽)

기본 메뉴: *Vector → Geoprocessing Tools*

더 보기:

차감하기 *(Difference)*, 잘라내기 *(Clip)*, 교차 *(Intersection)*

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	피처 (의 일부분) 를 추출할 첫 번째 레이어
Overlay layer	`OVERLAY`	[vector: any]	피처 (의 일부분) 를 추출할 두 번째 레이어. 도형 유형이 입력 레이어와 동일한 편이 좋습니다.
Overlay fields prefix 부가적	`OVERLAY_FIELDS_PREFIX`	[string]	입력 레이어 필드와 명칭 충돌을 피하기 위해 중첩 레이어의 필드명 앞에 붙일 접두어
Symmetrical difference	`OUTPUT`	[same as input] 기본값: `[Create temporary layer]`	입력 및 중첩 레이어에서 서로 중첩하지 않는 피처 (의 일부분) 를 담을 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Symmetrical difference	`OUTPUT`	[same as input]	각 레이어에서 다른 레이어와 중첩하지 않은 피처 (의 일부분) 를 담은 레이어

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 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.

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. 통합 레이어의 속성 테이블은 중첩하지 않는 피처의 경우 각 원본 레이어의 속성값으로, 중첩하는 피처의 경우 두 레이어의 속성값을 담게 됩니다.

참고: `union(A,B)` 알고리즘의 경우, 레이어 A 의 도형들 또는 레이어 B 의 도형들 가운데 중첩하는 것이 있다면 오류가 발생합니다. 모든 중첩하는 부분들을 처리하려면 `union(union(A,B))` 와 같이 실행해야 합니다. 예를

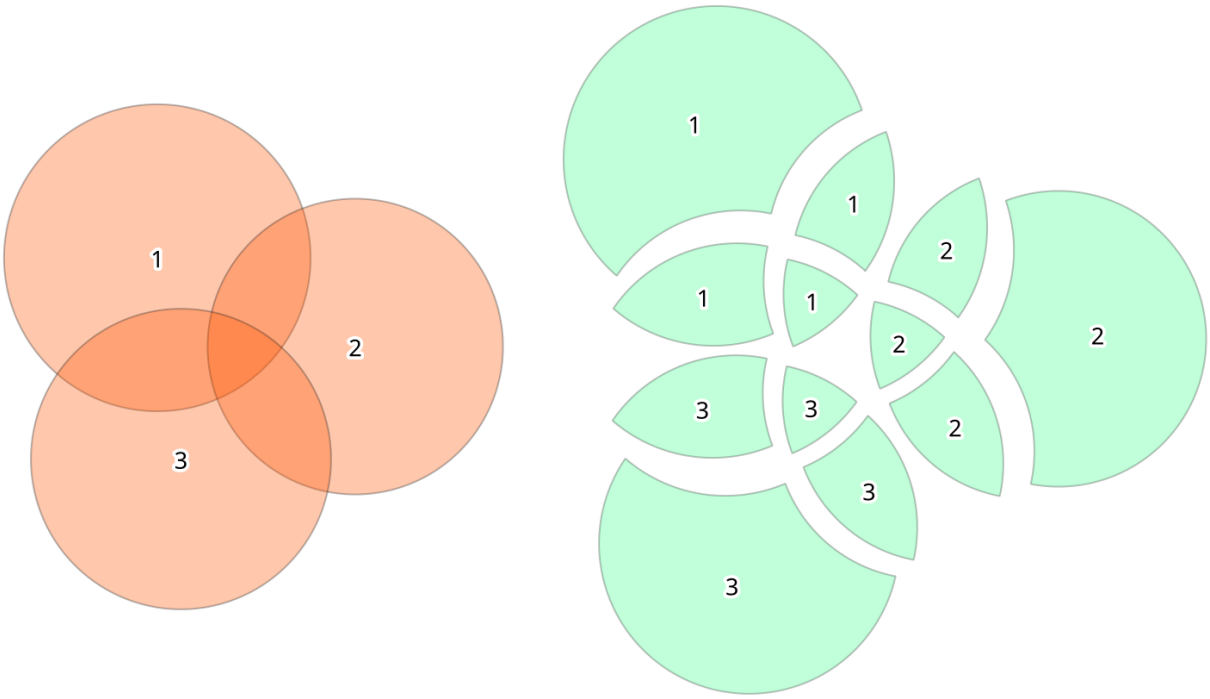


그림 24.98: 통합 작업: 중첩하는 피처 3 개를 가진 단일 입력 레이어 (왼쪽) 및 명확하게 보여주기 위해 이동시킨 산출 피처들 (오른쪽)

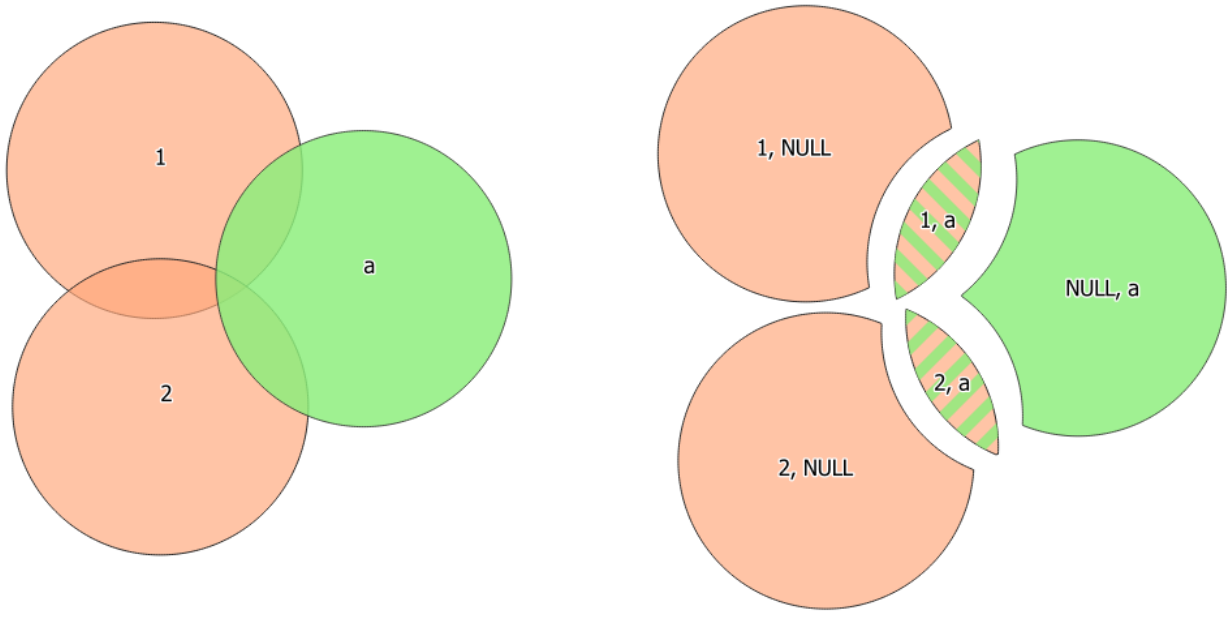


그림 24.99: 통합 작업: 피처 2 개를 가진 입력 레이어와 피처 1 개를 가진 중첩 레이어 (왼쪽) 및 명확하게 보여주기 위해 이동시킨 산출 피처들 (오른쪽)

들면 산출된 단일 레이어 `X=union(A,B)` 에 대해 `union(X)` 를 실행하듯이 말이죠.

기본 메뉴: *Vector → Geoprocessing Tools*

더 보기:

잘라내기 *(Clip)*, 차감하기 *(Difference)*, 교차 *(Intersection)*

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	모든 교차점에서 분할할 입력 벡터 레이어
Overlay layer 부가적	`OVERLAY`	[vector: any]	첫 번째 레이어로 결합될 레이어들. 도형 유형이 입력 레이어와 동일한 편이 좋습니다.
Overlay fields prefix 부가적	`OVERLAY_FIELDS_PREFIX`	[string]	입력 레이어 필드와 명칭 충돌을 피하기 위해 중첩 레이어의 필드명 앞에 붙일 접두어
Union	`OUTPUT`	[same as input] 기본값: `[Create temporary layer]`	입력 레이어 및 중첩 레이어에서 (분할된 그리고 복제된) 피처를 담을 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Union	`OUTPUT`	[same as input]	공간 처리된 레이어 (들) 에서 나온 중첩하는 그리고 중첩하지 않는 부분들을 모두 담고 있는 레이어

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

24.1.18 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	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	피처를 추출할 레이어
Selection attribute	`FIELD`	[tablefield: any]	레이어의 필터링 필드
Operator	`OPERATOR`	[enumeration] 기본값: 0	다음과 같은 많은 연산자를 사용할 수 있습니다: • 0 —= • 1 —≠ • 2 —> • 3 —>= • 4 —< • 5 —<= • 6 —begins with (…으로 시작) • 7 —contains (포함) • 8 —is null (NULL 값임) • 9 —is not null (NULL 값 아님) • 10 —does not contain (포함하지 않음)
Value 부가적	`VALUE`	[string]	평가할 값
Extracted (attribute)	`OUTPUT`	[same as input] 기본값: `[Create Temporary Layer]`	일치하는 피처를 저장할 산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

다음 페이지에 계속

표 24.130 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Extracted (non-matching)	`FAIL_OUTPUT`	[same as input] 기본값: `[Skip output]`	일치하지 않는 피처를 저장할 산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 산출물 건너뛰기 • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장

산출물

라벨	명칭	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

알고리즘 ID: `qgis: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 **명칭** 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	명칭	유형	설명
Input layer	INPUT	[vector: any]	입력 벡터 레이어
Expression	EXPRESSION	[expression]	벡터 레이어를 필터링할 표현식
Matching features	OUTPUT	[same as input] 기본값: [Create Temporary Layer]	일치하는 피처를 저장할 산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (TEMPORARY_OUTPUT) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.
Non-matching	FAIL_OUTPUT	[same as input] 기본값: [Skip output]	일치하지 않는 피처를 저장할 산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 산출물 건너뛰기 • 임시 레이어 생성 (TEMPORARY_OUTPUT) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장

산출물

라벨	명칭	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

알고리즘 ID: `qgis: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 **명칭** 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.

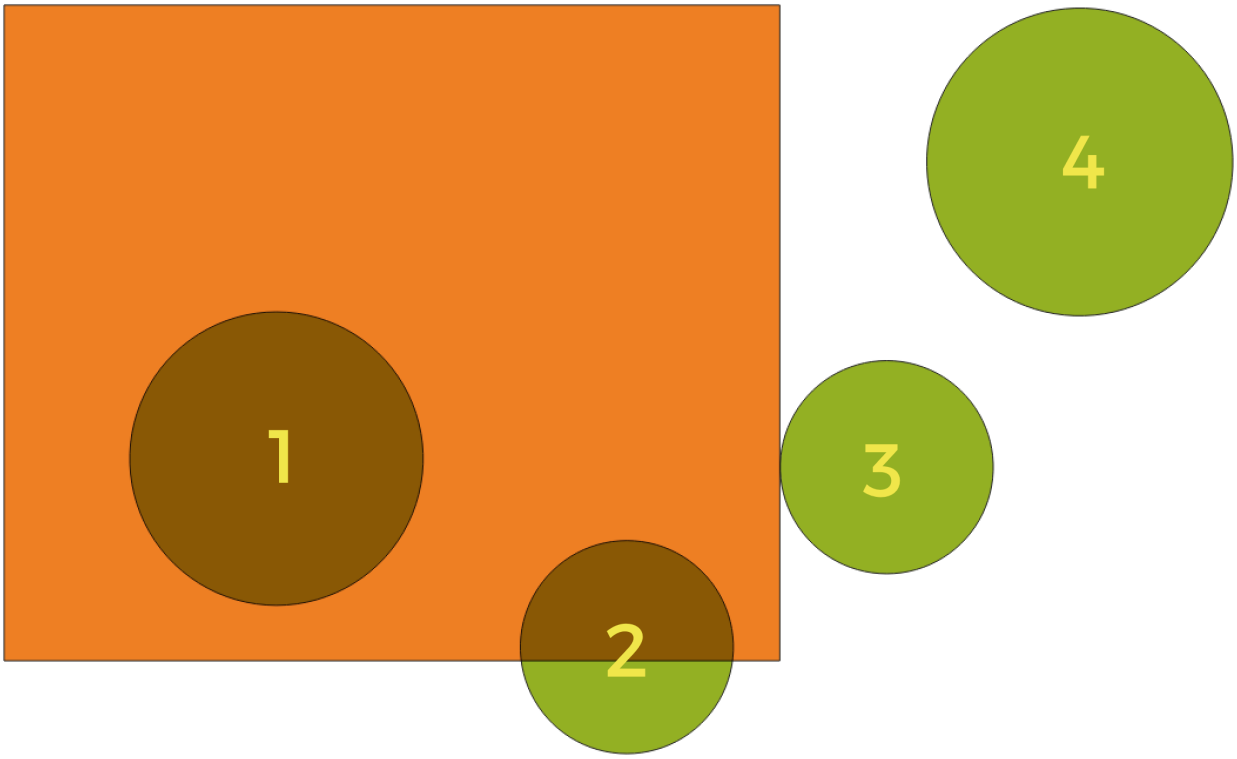


그림 24.100: 이 예시에서, 피처를 선택하고자 하는 데이터셋 (소스 벡터 레이어) 은 녹색 원으로 이루어져 있고, 주황색 직사각형은 이를 비교할 데이터셋 (교차 벡터 레이어) 입니다.

다음과 같은 기하학적 술어 (predicate) 를 사용할 수 있습니다:

Intersect 도형이 다른 도형과 교차하는지 검증합니다. 도형들이 공간적으로 교차하는 (공간의 어떤 부분이라도 공유 — 중첩 또는 접촉하는) 경우 1(참) 을 반환하고, 교차하지 않는 경우 0(거짓) 을 반환합니다. 앞의 그림에서, 이 조건은 원 1, 2, 3 을 선택할 것입니다.

Contain 도형 b 의 포인트 가운데 도형 a 외부에 있는 포인트가 하나도 없고, 도형 b 내부의 포인트 가운데 최소한 포인트 1 개가 도형 a 내부에 있는 경우에만 1(참) 을 반환합니다. 앞의 그림에서 이 조건은 어떤 원도 선택하지 않지만, a 와 b 의 순서를 바꿀 경우 직사각형이 원을 완전하게 포함하기 때문에 직사각형을 선택할 것입니다. 이 조건은 *are within* 조건의 정반대입니다.

Disjoint 도형들이 공간의 어떤 부분도 함께 공유하지 않는 (중첩하지도, 접촉하지도 않는) 경우 1(참) 을 반환합니다. 이 조건은 원 4 만 선택합니다.

Equal 도형이 정확하게 동일한 경우에만 1(참) 을 반환합니다. 이 조건은 아무 원도 선택하지 않을 것입니다.

Touch 도형이 다른 도형과 접하는지 검증합니다. 도형들이 최소한 포인트 1 개를 공유하지만 각 도형의 내부가 교차하지 않는 경우 1(참) 을 반환합니다. 이 조건은 원 3 만 선택합니다.

Overlap 도형이 다른 도형과 중첩하는지 검증합니다. 도형들이 동일한 차원이며 공간을 공유하지만 서로를 완전히 담고 있지 않은 경우 1(참) 을 반환합니다. 이 조건은 원 2 만 선택합니다.

Are within 도형이 다른 도형 내부에 있는지 검증합니다. 도형 a 가 도형 b 내부에 완전히 들어가 있는 경우 1(참) 을 반환합니다. 이 조건은 원 1 만 선택합니다.

Cross 입력 도형들이 내부 포인트 전부가 아닌 일부를 공유하고, 공간 교차 (cross) 가 입력 도형의 최고 차원보다 낮은 차원에서 발생하는 경우 1(참) 을 반환합니다. 예를 들어, 폴리곤과 공간 교차하는 라인은 라인으로서 공간 교차할 것입니다 (라인을 선택합니다). 공간 교차하는 두 라인은 포인트로서 공간 교차할 것입니다 (포인트를 선택합니다). 두 폴리곤은 폴리곤으로서 공간 교차합니다 (선택하지 않습니다).

더 보기:

위치로 선택하기

파라미터

라벨	명칭	유형	설명
Extract features from	`INPUT`	[vector: any]	입력 벡터 레이어
Where the features (geometric predicate)	`PREDICATE`	[enumeration] [list] 기본값: [0]	선택을 위한 공간 조건을 지정합니다. 다음 가운데 하나 이상을 지정할 수 있습니다: • 0 — 교차 (intersect) • 1 — 포함 (contain) • 2 — 격리 (disjoint) • 3 — 동등 (equal) • 4 — 접촉 (touch) • 5 — 중첩 (overlap) • 6 — 내부 (are within) • 7 — 공간 교차 (cross) 조건을 하나 이상 선택한 경우, 적어도 하나는 (OR 연산자) 추출할 피처를 만족시켜야 합니다.
By comparing to the features from	`INTERSECT`	[vector: any]	교차 벡터 레이어
Extracted (location)	`OUTPUT`	[same as input] 기본값: `[Create temporary layer]`	비교 레이어의 하나 이상의 피처와 선택한 공간 관계 (들) 을 가지는 피처를 저장할 산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장

산출물

라벨	명칭	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

알고리즘 ID: `qgis: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 **명칭** 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.

더 보기:

랜덤하게 선택하기

파라미터

라벨	명칭	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: • 0 — 선택한 피처의 개수 • 1 — 선택한 피처의 백분율
Number/percentage of selected features	NUMBER	[number] 기본값: 10	선택할 피처들의 개수 또는 백분율

다음 페이지에 계속

표 24.132 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Extracted (random)	`OUTPUT`	[vector: any] 기본값: `[Create temporary layer]`	랜덤하게 선택한 피처를 저장할 산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 랜덤하게 선택한 피처를 담고 있는 벡터 레이어

산출물

라벨	명칭	유형	설명
Extracted (random)	`OUTPUT`	[same as input]	입력 레이어에서 랜덤하게 선택한 피처를 담고 있는 벡터 레이어

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 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	명칭	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] 기본값: 0	랜덤 선택 메소드를 설정합니다. 다음 가운데 하나를 선택할 수 있습니다: • 0 — 선택한 피처의 개수 • 1 — 선택한 피처의 백분율
Number/percentage of selected features	`NUMBER`	[number] 기본값: 10	선택할 피처들의 개수 또는 백분율
Extracted (random stratified)	`OUTPUT`	[same as input] 기본값: `[Create temporary layer]`	랜덤하게 선택한 피처를 저장할 산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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	명칭	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: • 0 — 선택한 피처의 개수 • 1 — 선택한 피처의 백분율
Number/percentage of selected features	`NUMBER`	[number] 기본값: 10	선택할 피처들의 개수 또는 백분율

산출물

라벨	명칭	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 **명칭** 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	명칭	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: • 0 — 선택한 피처의 개수 • 1 — 선택한 피처의 백분율
Number/percentage of selected features	`NUMBER`	[number] 기본값: 10	선택할 피처들의 개수 또는 백분율

산출물

라벨	명칭	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 **명칭** 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	명칭	유형	설명
Input layer	INPUT	[vector: any]	피처를 선택할 벡터 레이어
Selection attribute	FIELD	[tablefield: any]	레이어의 필터링 필드
Operator	OPERATOR	[enumeration] 기본값: 0	다음과 같은 많은 연산자를 사용할 수 있습니다: • 0 —= • 1 —≠ • 2 —> • 3 —>= • 4 —< • 5 —<= • 6 —begins with (…으로 시작) • 7 —contains (포함) • 8 —is null (NULL 값임) • 9 —is not null (NULL 값 아님) • 10 —does not contain (포함하지 않음)
Value 부가적	VALUE	[string]	평가할 값
Modify current selection by	METHOD	[enumeration] 기본값: 0	알고리즘의 선택 작업을 어떻게 관리해야 할지 지정합니다. 다음 가운데 하나를 선택할 수 있습니다: • 0 —새 선택 집합 생성 • 1 —현재 선택 집합에 추가 • 2 —현재 선택 집합에서 제거 • 3 –현재 선택 집합 내에서 선택

산출물

라벨	명칭	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 **명칭** 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	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	입력 벡터 레이어
Expression	`EXPRESSION`	[expression]	입력 레이어를 필터링할 표현식
Modify current selection by	`METHOD`	[enumeration] 기본값: 0	알고리즘의 선택 작업을 어떻게 관리해야 할지 지정합니다. 다음 가운데 하나를 선택할 수 있습니다: • 0 —새 선택 집합 생성 • 1 —현재 선택 집합에 추가 • 2 —현재 선택 집합에서 제거 • 3 –현재 선택 집합 내에서 선택

산출물

라벨	명칭	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 **명칭** 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.

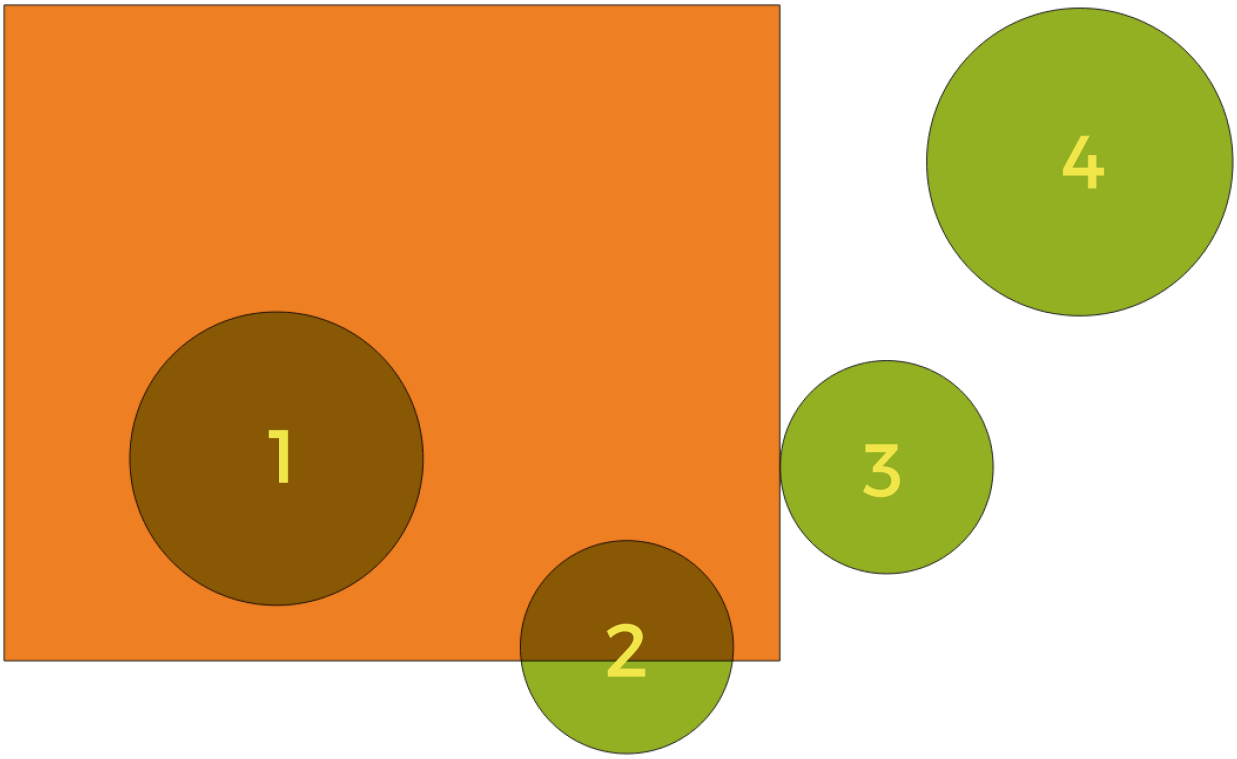


그림 24.101: 이 예시에서, 피처를 선택하고자 하는 데이터셋 (소스 벡터 레이어) 은 녹색 원으로 이루어져 있고, 주황색 직사각형은 이를 비교할 데이터셋 (교차 벡터 레이어) 입니다.

다음과 같은 기하학적 술어 (predicate) 를 사용할 수 있습니다:

Intersect 도형이 다른 도형과 교차하는지 검증합니다. 도형들이 공간적으로 교차하는 (공간의 어떤 부분이라도 공유 — 중첩 또는 접촉하는) 경우 1(참) 을 반환하고, 교차하지 않는 경우 0(거짓) 을 반환합니다. 앞의 그림에서, 이 조건은 원 1, 2, 3 을 선택할 것입니다.

Contain 도형 b 의 포인트 가운데 도형 a 외부에 있는 포인트가 하나도 없고, 도형 b 내부의 포인트 가운데 최소한 포인트 1 개가 도형 a 내부에 있는 경우에만 1(참) 을 반환합니다. 앞의 그림에서 이 조건은 어떤 원도 선택하지 않지만, a 와 b 의 순서를 바꿀 경우 직사각형이 원을 완전하게 포함하기 때문에 직사각형을 선택할 것입니다. 이 조건은 *are within* 조건의 정반대입니다.

Disjoint 도형들이 공간의 어떤 부분도 함께 공유하지 않는 (중첩하지도, 접촉하지도 않는) 경우 1(참) 을 반환합니다. 이 조건은 원 4 만 선택합니다.

Equal 도형이 정확하게 동일한 경우에만 1(참) 을 반환합니다. 이 조건은 아무 원도 선택하지 않을 것입니다.

Touch 도형이 다른 도형과 접하는지 검증합니다. 도형들이 최소한 포인트 1 개를 공유하지만 각 도형의 내부가 교차하지 않는 경우 1(참) 을 반환합니다. 이 조건은 원 3 만 선택합니다.

Overlap 도형이 다른 도형과 중첩하는지 검증합니다. 도형들이 동일한 차원이며 공간을 공유하지만 서로를 완전히 담고 있지 않은 경우 1(참) 을 반환합니다. 이 조건은 원 2 만 선택합니다.

Are within 도형이 다른 도형 내부에 있는지 검증합니다. 도형 a 가 도형 b 내부에 완전히 들어가 있는 경우 1(참) 을 반환합니다. 이 조건은 원 1 만 선택합니다.

Cross 입력 도형들이 내부 포인트 전부가 아닌 일부를 공유하고, 공간 교차 (cross) 가 입력 도형의 최고 차원보다 낮은 차원에서 발생하는 경우 1(참) 을 반환합니다. 예를 들어, 폴리곤과 공간 교차하는 라인은 라인으로서 공간 교차할 것입니다 (라인을 선택합니다). 공간 교차하는 두 라인은 포인트로서 공간 교차할 것입니다 (포인트를 선택합니다). 두 폴리곤은 폴리곤으로서 공간 교차합니다 (선택하지 않습니다).

기본 메뉴: *Vector → Research Tools*

더 보기:

위치로 추출하기

파라미터

라벨	명칭	유형	설명
Select features from	`INPUT`	[vector: any]	입력 벡터 레이어
Where the features (geometric predicate)	`PREDICATE`	[enumeration] [list] 기본값: [0]	선택을 위한 공간 조건을 지정합니다. 다음 가운데 하나 이상을 지정할 수 있습니다: • 0 — 교차 (intersect) • 1 — 포함 (contain) • 2 — 격리 (disjoint) • 3 — 동등 (equal) • 4 — 접촉 (touch) • 5 — 중첩 (overlap) • 6 — 내부 (are within) • 7 — 공간 교차 (cross) 조건을 하나 이상 선택한 경우, 적어도 하나는 (OR 연산자) 추출할 피처를 만족시켜야 합니다.
By comparing to the features from	`INTERSECT`	[vector: any]	교차 벡터 레이어
Modify current selection by	`METHOD`	[enumeration] 기본값: 0	알고리즘의 선택 작업을 어떻게 관리해야 할지 지정합니다. 다음 가운데 하나를 선택할 수 있습니다: • 0 — 새 선택 집합 생성 • 1 — 현재 선택 집합에 추가 • 2 — 현재 선택 집합 내에서 선택 • 3 — 현재 선택 집합 내에서 제거

산출물

라벨	명칭	유형	설명
Input layer	INPUT	[same as input]	선택한 피처를 가진 입력 레이어

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

24.1.19 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	명칭	유형	설명
Input layer	INPUT	[vector: any]	입력 벡터 레이어
Field name	FIELD_NAME	[string] 기본값: 'AUTO'	자동 증가하는 값을 가진 필드의 명칭
Start values at 부가적	START	[number] 기본값: 0	증가하는 번호의 맨 처음 숫자를 선택합니다.
Group values by 부가적	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 부가적	SORT_EXPRESSION	[expression]	표현식을 사용해서 레이어 전체 수준에서 피처들을 정렬하거나, 필드 그룹을 설정한 경우 필드 그룹을 기반으로 피처를 정렬합니다.
Sort ascending	SORT_ASCENDING	[boolean] 기본값: True	`sort expression` 을 설정한 경우, 이 옵션을 사용해서 피처에 값을 할당하는 순서를 제어할 수 있습니다.

다음 페이지에 계속

표 24.136 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Sort nulls first	SORT_NULLS_FIRST	[boolean] 기본값: False	sort expression 을 설정한 경우, 이 옵션을 사용해서 *NULL* 값을 맨 처음 정렬시킬시 또는 마지막으로 정렬시킬지 설정할 수 있습니다.
Incremented	OUTPUT	[same as input] 기본값: [Create temporary layer]	자동 증가 필드를 가질 산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (TEMPORARY_OUTPUT) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Incremented	OUTPUT	[same as input]	자동 증가 필드를 가진 벡터 레이어

파이썬 코드

알고리즘 ID: qgis: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 **명칭** 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 명칭 및 특성은 파라미터로 정의합니다.

입력 레이어에는 새 속성을 추가하지 않지만 그 대신 새 레이어를 생성할 것입니다.

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	입력 레이어
Field name	`FIELD_NAME`	[string]	새 필드의 명칭
Field type	`FIELD_TYPE`	[enumeration] 기본값: 0	새 필드의 유형을 다음 가운데 선택할 수 있습니다: • 0 — 정수형 • 1 — 부동소수점 실수형 • 2 — 문자열
Field length	`FIELD_LENGTH`	[number] 기본값: 10	필드의 길이
Field precision	`FIELD_PRECISION`	[number] 기본값: 0	필드의 정밀도. 부동소수점 필드 유형에 유용합니다.
Added	`OUTPUT`	[same as input] 기본값: `[Create temporary layer]`	산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (`TEMPORARY_OUTPUT`) • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Added	`OUTPUT`	[same as input]	새 필드를 추가한 벡터 레이어

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 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	명칭	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.
Output field name	`FIELD_NAME`	[string] 기본값: 'NUM_FIELD'	인덱스를 담고 있는 새 필드의 명칭
Layer with index field	`OUTPUT`	[vector: any] 기본값: `[Create temporary layer]`	인덱스를 담고 있는 숫자 필드를 가질 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 산출물 건너뛰기 • 임시 레이어 생성 • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.
Class summary	`SUMMARY_OUTPUT`	[table] 기본값: `[Skip output]`	대응하는 유일값에 맵핑된 범주 필드의 요약을 담을 테이블을 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 산출물 건너뛰기 • 임시 레이어 생성 • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Layer with index field	`OUTPUT`	[same as input]	인덱스를 담고 있는 숫자 필드를 가진 벡터 레이어
Class summary	`SUMMARY_OUTPUT`	[table] 기본값: `[Skip Output]`	대응하는 유일값에 맵핑된 범주 필드의 요약을 담은 테이블

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 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.)

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: point]	입력 레이어
Coordinate system	`CRS`	[crs] 기본값: "EPSG:4326"	생성된 X 및 Y 필드에 사용할 좌표계
Field prefix 부가적	`PREFIX`	[string]	입력 레이어 필드와 명칭 충돌을 피하기 위해 새 필드명 앞에 붙일 접두어
Added fields	`OUTPUT`	[vector: point] 기본값: `[Create temporary layer]`	산출 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Added fields	`OUTPUT`	[vector: point]	입력 레이어와 동일하지만 새 `x` 및 `y` 필드 2 개를 가진 산출 레이어

파이썬 코드

알고리즘 ID: `qgis:addxyfieldstolayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 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.

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	입력 벡터 레이어
Result field name	`FIELD_NAME`	[string] 기본값: 'NewField'	새 필드의 명칭
Field type	`FIELD_TYPE`	[enumeration] 기본값: 0	새 필드의 데이터 유형입니다. 다음 가운데 하나로 지정할 수 있습니다: • 0 —정수형 • 1 —부동소수점 실수형 • 2 —문자열
Field length	`FIELD_LENGTH`	[number] 기본값: 10	필드의 길이
Field precision	`FIELD_PRECISION`	[number] 기본값: 3	필드의 정밀도. 부동소수점 필드 유형에 유용합니다.
Global expression 부가적	`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.
Formula	`FORMULA`	[string]	값을 구할 파이썬 공식입니다. 예를 들어 입력 폴리곤 레이어의 면적을 계산하려면 다음 공식을 추가하면 됩니다: `value = $geom.area()`

다음 페이지에 계속

표 24.140 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Calculated	OUTPUT	[same as input] 기본값: [Create temporary layer]	새로 계산된 필드를 가질 산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

파라미터

라벨	명칭	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] 기본값: [Create temporary layer]	남아 있는 필드를 담을 산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Remaining fields	`OUTPUT`	[same as input]	남아 있는 필드를 담은 벡터 레이어

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 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.

PostgreSQL HStore 는 PostgreSQL 및 OGR 에서 (`other_tags` 필드를 가진 OSM 파일 을 읽어오는 경우) 사용되는 단순 키-값 저장소입니다.

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	입력 벡터 레이어
HStore field	`FIELD`	[tablefield: any]	삭제할 필드 (들)
Expected list of fields separated by a comma 부가적	`EXPECTED_FIELDS`	[string] 기본값: ''	추출할 필드의 쉼표 구분 목록입니다. 이 키들을 제거해서 HStore 필드를 업데이트할 것입니다.
Exploded	`OUTPUT`	[same as input] 기본값: `[Create temporary layer]`	산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Exploded	OUTPUT	[same as input]	산출 벡터 레이어

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 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	명칭	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] 기본값: `[Save to a temporary folder]`	산출 파일을 저장할 폴더를 지정합니다. 다음 가운데 하나로 선택할 수 있습니다: • 임시 디렉터리에 저장 • 디렉터리에 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Folder	FOLDER	[folder]	산출 파일을 담고 있는 폴더

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

피처 필터링하기

입력 레이어에서 피처를 필터링해서 하나 또는 여러 개의 산출물로 전용합니다. 가능한 모든 입력 레이어들이 공통으로 가지고 있는 속성명을 하나도 모르는 경우, 가능한 방법은 `$id` 및 `uuid` 같은 피처 도형 및 일반 레코드 기제 (mechanism) 에 대한 필터링뿐입니다.

참고: 이 알고리즘은그래픽 모델 생성기 에서만 사용할 수 있습니다.

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	입력 레이어
Outputs and filters (하나 이상)	`OUTPUT_<name of the filter>`	[same as input]	필터링된 (사용한 필터 개수의) 산출 레이어들

산출물

라벨	명칭	유형	설명
산출물 (하나 이상)	`native:filter_1:OUTPUT_<name of filter>`	[same as input]	필터링된 피처를 가진 (사용한 필터 개수의) 산출 레이어들

파이썬 코드

알고리즘 ID: `qgis:featurefilter`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 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 계산기는그래픽 모델 생성기 에서 매우 유용하게 사용됩니다.

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	계산할 레이어
Output field name	`FIELD_NAME`	[string]	계산 결과를 저장할 필드의 명칭
Output field type	`FIELD_TYPE`	[enumeration] 기본값: 0	필드의 데이터 유형입니다. 다음 가운데 하나로 지정할 수 있습니다: • 0 —부동소수점 실수형 • 1 —정수형 • 2 —문자열 • 3 —날짜
Output field width	`FIELD_LENGTH`	[number] 기본값: 10	산출 필드의 길이 (최소값 0)
Field precision	`FIELD_PRECISION`	[number] 기본값: 3	산출 필드의 정밀도 (최소 0, 최대 15)
Create new field	`NEW_FIELD`	[boolean] 기본값: True	산출 필드를 새 필드로 생성할지 여부
Formula	`FORMULA`	[expression]	결과를 계산하는 데 사용할 공식
Output file	`OUTPUT`	[vector: any] 기본값: `[Save to temporary file]`	산출 레이어를 지정합니다.

산출물

라벨	명칭	유형	설명
Calculated	`OUTPUT`	[vector: any]	계산된 필드 값을 가진 산출 레이어

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 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 명칭을 수정할 수 있습니다.

원본 레이어는 수정하지 않습니다. 지정한 필드 맵핑에 따라 수정된 속성 테이블을 담고 있는 새 레이어를 생성합니다.

레이어 필드를 재구성 (refactor) 하면:

- 필드 명칭 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.

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	수정할 레이어

다음 페이지에 계속

표 24.143 – 이전 페이지에서 계속

라벨	명칭	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: •  버튼을 클릭해서 새 필드를 추가할 수 있습니다. •  버튼을 클릭해서 선택한 필드를 삭제할 수 있습니다. •  및  버튼을 클릭해서 선택한 필드의 순서를 변경할 수 있습니다. •  버튼을 클릭해서 기본 뷰로 리셋할 수 있습니다. 재사용하고 싶은 각 필드마다, 다음 옵션을 설정해줘야 합니다: ***Source expression*** **(expression) [expression]** 입력 레이어에서 나온 필드 또는 표현식 ***Field name*** **(name) [string]** 산출 레이어의 필드명입니다. 기본적으로 입력 필드명을 그대로 유지합니다. ***Type*** **(type) [enumeration]** 산출 필드의 데이터 유형입니다. 다음 가운데 하나로 지정할 수 있습니다: • Date (14): 날짜 • DateTime (16): 날짜 & 시간 • Double (6): 64 비트 실수형 • Integer (2): 정수형 • Integer64 (4): 64 비트 정수형 • String (10): 문자열 • Boolean (1): 불 ***Length*** **(length) [number]** 산출 필드의 길이 ***Precision*** **(precision) [number]** 산출 필드의 정밀도 *Load fields from layer* 에서 다른 레이어의 필드를 필드 목록으로 불러올 수 있습니다.
Refactored	OUTPUT	[vector: any] 기본값: [Create temporary layer]	산출 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

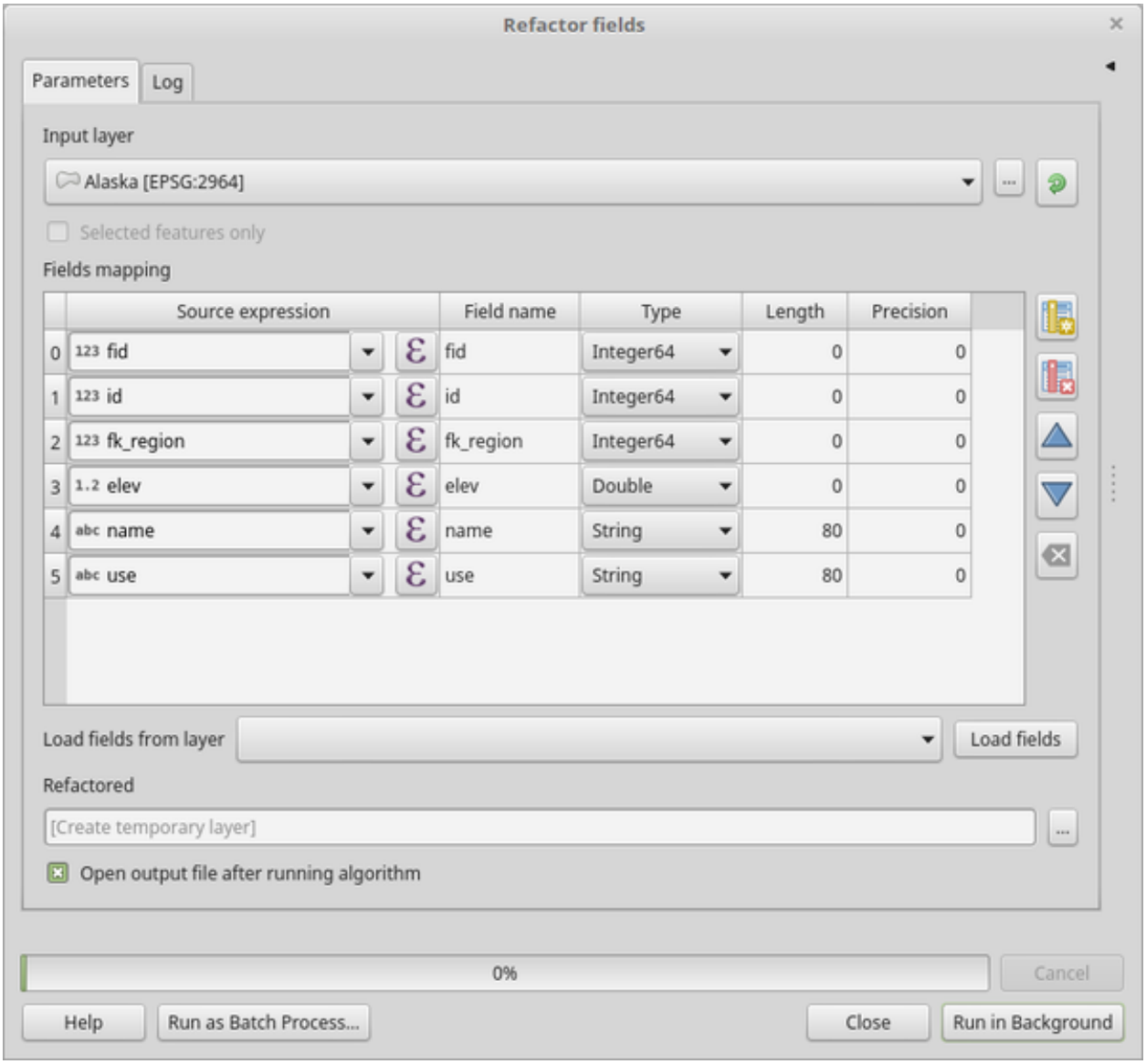


그림 24.102: 필드 재구성 대화창

산출물

라벨	명칭	유형	설명
Refactored	OUTPUT	[vector: any]	재구성된 필드를 가진 산출 레이어

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

벡터 필드명 바꾸기

벡터 레이어의 기존 필드를 재명명합니다.

원본 레이어는 수정하지 않습니다. 재명명된 필드를 담고 있는 속성 테이블을 가진 새 레이어를 생성합니다.

더 보기:

필드 재구성하기

파라미터

라벨	명칭	유형	설명
Input layer	INPUT	[vector: any]	입력 벡터 레이어
Field to rename	FIELD	[string]	변경할 필드
New field name	NEW_NAME	[string]	새 필드명
Renamed	OUTPUT	[vector: same as input] 기본값: [Create temporary layer]	산출 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Renamed	`OUTPUT`	[vector: same as input]	재명명된 필드를 가진 산출 레이어

파이썬 코드

알고리즘 ID: `qgis: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 **명칭** 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	명칭	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] 기본값: `[Create Temporary Layer]`	산출 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 • 파일로 저장 • Geopackage 로 저장 • PostGIS 테이블로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

24.1.20 벡터 타일

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`	[number] 기본값: 0	타일셋이 데이터를 제공하기 위한 최저 확대/축소 수준. 0 에서 24 사이로 설정합니다.
Maximum zoom level	`MAX_ZOOM`	[number] 기본값: 3	타일셋이 데이터를 제공하기 위한 최고 확대/축소 수준. 0 에서 24 사이로 설정합니다.
Extent 부가적	`EXTENT`	[extent] 기본값: 설정하지 않음	렌더링된 맵 영역의 최대 범위. 경계는 모든 확대/축소 수준이 커버하는 영역을 정의해야만 합니다.
Metadata: Name 부가적	`META_NAME`	[string]	타일셋의 이름
Metadata: Description 부가적	`META_DESCRIPTION`	[string]	타일셋 내용의 설명
Metadata: Attribution 부가적	`META_ATTRIBUTION`	[string]	데이터 소스 그리고/또는 맵 용 스타일을 설명하는 속성 문자열
Metadata: Version 부가적	`META_VERSION`	[string]	타일셋의 버전. MBTiles 사양이 아니라 타일셋 자체의 리비전 (revision) 을 참조하는 버전입니다.
Metadata: Type 부가적	`META_TYPE`	[string]	타일셋의 유형. `overlay` 또는 `baselayer` 가운데 하나를 값으로 쓸 수 있습니다.
Metadata: Center 부가적	`META_CENTER`	[string]	맵의 기본 뷰의 중심. (쉼표로 분리된 숫자 문자열: 경도, 위도, 확대/축소 수준) 예: `-122.1906,37.7599,11`
Destination MBTiles	`OUTPUT`	[vector tiles] 기본값: [Save to temporary file]	산출 MBTiles 파일을 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 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 **명칭** 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`	[number] 기본값: 0	타일셋이 데이터를 제공하기 위한 최저 확대/축소 수준. 0 에서 24 사이로 설정합니다.
Maximum zoom level	`MAX_ZOOM`	[number] 기본값: 3	타일셋이 데이터를 제공하기 위한 최고 확대/축소 수준. 0 에서 24 사이로 설정합니다.
Extent 부가적	`EXTENT`	[extent] 기본값: 설정하지 않음	렌더링된 맵 영역의 최대 범위. 경계는 모든 확대/축소 수준이 커버하는 영역을 정의해야만 합니다.
Output directory	`OUTPUT_DIRECTORY`	[folder] 기본값: [Save to temporary folder]	산출 벡터 타일 폴더를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 디렉터리에 저장 • 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 **명칭** 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

경사 방향

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).

이 알고리즘은 GDAL DEM 유틸리티 에서 파생되었습니다.

기본 메뉴: *Raster → Analysis*

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[raster]	입력 표고 래스터 레이어
Band number	`BAND`	[raster band] 기본값: 1	표고로 사용할 밴드의 번호
Return trigonometric angle instead of azimuth	`TRIG_ANGLE`	[boolean] 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] 기본값: False	이 옵션을 활성화하면 평야 지대에 -9999 값 대신 0 값을 삽입할 것입니다.
Compute edges	`COMPUTE_EDGES`	[boolean] 기본값: False	표고 래스터에서 경계를 생성합니다.
Use Zevenbergen&Thorne formula instead of the Horn's one	`ZEVENBERGEN`	[boolean] 기본값: False	매끄러운 풍경을 위해 체벤베르겐 & 손 (Zevenbergen&Thorne) 공식을 활성화합니다.

다음 페이지에 계속

표 24.145 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Additional creation options 부가적	`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) 프로세스의 경우: 파이프 문자 (|) 로 복수의 옵션들을 분리하십시오.
Additional command-line parameters 부가적	`EXTRA`	[string] 기본값: None	부가적인 GDAL 명령 줄 옵션을 추가합니다.
Aspect	`OUTPUT`	[raster] 기본값: `[Save to temporary file]`	산출 경사 방향 래스터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

이 알고리즘은 GDAL DEM 유틸리티 에서 파생되었습니다.

파라미터

라벨	명칭	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
Matching mode	`MATCH_MODE`	[enumeration] 기본값: 2	다음 가운데 하나로 지정할 수 있습니다: • 0 —색상을 엄격하게 맞춤 • 1 —최근접 RGBA 4 배수 (quadruple) 사용 • 2 —부드럽게 혼합된 색상 사용
Additional creation options 부가적	`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) 프로세스의 경우: 파이프 문자 (|) 로 복수의 옵션들을 분리하십시오.
Additional command-line parameters 부가적	`EXTRA`	[string] 기본값: None	부가적인 GDAL 명령 줄 옵션을 추가합니다.
Color relief	`OUTPUT`	[raster] 기본값: `[Save to temporary file]`	산출 경사 방향 래스터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

이 알고리즘은 GDAL NODATA 채우기 유틸리티 에서 파생되었습니다.

기본 메뉴: *Raster → Analysis*

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[raster]	입력 래스터 레이어
Band number	`BAND`	[raster band] 기본값: 1	작업할 밴드. NODATA 값은 0 값으로 표현되어야만 합니다.
Maximum distance (in pixels) to search out for values to interpolate	`DISTANCE`	[number] 기본값: 10	모든 방향에서 보간할 값을 찾기 위해 검색할 픽셀들의 개수
Number of smoothing iterations to run after the interpolation	`ITERATIONS`	[number] 기본값: 0	보간 작업의 결과를 평탄화하기 위해 (0 번 이상) 실행할 3x3 필터링의 횟수
Do not use default validity mask for the input band	`NO_MASK`	[boolean] 기본값: False	사용자 정의 유효성 마스크를 활성화합니다.
Validity mask	`MASK_LAYER`	[raster]	채워야 할 영역을 정의하는 래스터 레이어
Additional creation options 부가적	`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) 프로세스의 경우: 파이프 문자 (|) 로 복수의 옵션들을 분리하십시오.

다음 페이지에 계속

표 24.147 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Additional command-line parameters 부가적	EXTRA	[string] 기본값: None	부가적인 GDAL 명령 줄 옵션을 추가합니다.
Filled	OUTPUT	[raster] 기본값: [Save to temporary file]	산출 래스터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

이 알고리즘은 GDAL 그리드 유틸리티 에서 파생되었습니다.

기본 메뉴: *Raster → Analysis*

더 보기:

GDAL 그리드 예제

파라미터

라벨	명칭	유형	설명
Point layer	INPUT	[vector: point]	입력 포인트 벡터 레이어

다음 페이지에 계속

표 24.148 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Data metric to use	METRIC	[enumeration] 기본값: 0	다음 가운데 하나로 지정할 수 있습니다: • 0 —최소값: 그리드 노드 검색 타원에서 발견된 최소값 • 1 —최대값: 그리드 노드 검색 타원에서 발견된 최대값 • 2 —범위: 그리드 노드 검색 타원에서 발견된 최소값 및 최대값의 차 • 3 —개수: 그리드 노드 검색 타원에서 발견된 데이터 포인트의 개수 • 4 —평균 거리: 그리드 노드 (검색 타원의 중심) 와 그리드 노드 검색 타원에서 발견된 모든 데이터 포인트 사이의 평균 거리 • 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	[number] 기본값: 0.0	검색 타원의 첫 번째 (기울기 각도가 0 인 경우 X 축) 반경
The second radius of search ellipse	RADIUS_2	[number] 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	[number] 기본값: 0.0	타원 기울기의 도 단위 각도. 타원은 반시계 방향으로 기울어집니다.
Minimum number of data points to use	MIN_POINTS	[number] 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	[number] 기본값: 0.0	비어 있는 포인트를 채울 NODATA 마커
Z value from field 부가적	Z_FIELD	[tablefield: numeric]	보간 작업용 필드
Additional creation options 부가적	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) 프로세스의 경우: 파이프 문자 (|) 로 복수의 옵션들을 분리하십시오.
Additional command-line parameters 부가적	EXTRA	[string] 기본값: None	부가적인 GDAL 명령 줄 옵션을 추가합니다.

다음 페이지에 계속

표 24.148 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Output data type	`DATA_TYPE`	[enumeration] 기본값: 5	산출 래스터 파일의 데이터 유형을 정의합니다. 다음 가운데 하나로 지정할 수 있습니다. • 0 —바이트 (Byte) • 1 —부호 있는 16 비트 정수형 (Int16) • 2 —부호 없는 16 비트 정수형 (UInt16) • 3 —부호 없는 32 비트 정수형 (UInt32) • 4 —부호 있는 32 비트 정수형 (Int32) • 5 —32 비트 부동소수점 실수형 (Float32) • 6 —부동소수점 64 비트 실수형 (Float64) • 7 —복잡 16 비트 정수형 (CInt16) • 8 —복잡 32 비트 정수형 (CInt32) • 9 —복잡 부동소수점 32 비트 실수형 (CFloat32) • 10 —복잡 부동소수점 64 비트 실수형 (CFloat64)
Interpolated (data metrics)	`OUTPUT`	[raster] 기본값: `[Save to temporary file]`	보간된 값을 가진 산출 래스터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

이 알고리즘은 GDAL 그리드 유틸리티 에서 파생되었습니다.

더 보기:

GDAL 그리드 예제

파라미터

라벨	명칭	유형	설명
Point layer	`INPUT`	[vector: point]	입력 포인트 벡터 레이어
Weighting power	`POWER`	[number] Default: 2.0	가중치 제곱지수
Smoothing	`SMOOTHING`	[number] 기본값: 0.0	평탄화 파라미터
The radius of the search circle	`RADIUS`	[number] 기본값: 1.0	검색 원의 반경
Maximum number of data points to use	`MAX_POINTS`	[number] 기본값: 12	이 값을 초과하는 개수의 포인트를 검색하지 않습니다.
Minimum number of data points to use	`MIN_POINTS`	[number] 기본값: 0	평균을 낼 데이터 포인트의 최소 개수. 이 값보다 적은 포인트가 발견된 그리드 노드는 비어 있다고 간주되어 NODATA 마커로 채워질 것입니다.
Nodata	`NODATA`	[number] 기본값: 0.0	비어 있는 포인트를 채울 NODATA 마커
Z value from field 부가적	`Z_FIELD`	[tablefield: numeric]	보간 작업용 필드
Additional creation options 부가적	`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) 프로세스의 경우: 파이프 문자 (|) 로 복수의 옵션들을 분리하십시오.
Additional command-line parameters 부가적	`EXTRA`	[string] 기본값: None	부가적인 GDAL 명령 줄 옵션을 추가합니다.

다음 페이지에 계속

표 24.149 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Output data type	`DATA_TYPE`	[enumeration] 기본값: 5	산출 래스터 파일의 데이터 유형을 정의합니다. 다음 가운데 하나로 지정할 수 있습니다. • 0 —바이트 (Byte) • 1 —부호 있는 16 비트 정수형 (Int16) • 2 —부호 없는 16 비트 정수형 (UInt16) • 3 —부호 없는 32 비트 정수형 (UInt32) • 4 —부호 있는 32 비트 정수형 (Int32) • 5 —32 비트 부동소수점 실수형 (Float32) • 6 —부동소수점 64 비트 실수형 (Float64) • 7 —복잡 16 비트 정수형 (CInt16) • 8 —복잡 32 비트 정수형 (CInt32) • 9 —복잡 부동소수점 32 비트 실수형 (CFloat32) • 10 —복잡 부동소수점 64 비트 실수형 (CFloat64)
Interpolated (IDW with NN search)	`OUTPUT`	[raster] 기본값: `[Save to temporary file]`	보간된 값을 가진 산출 래스터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

이 알고리즘은 GDAL 그리드 유틸리티 에서 파생되었습니다.

기본 메뉴: *Raster → Analysis*

더 보기:

GDAL 그리드 예제

파라미터

라벨	명칭	유형	설명
Point layer	`INPUT`	[vector: point]	입력 포인트 벡터 레이어
Weighting power	`POWER`	[number] Default: 2.0	가중치 제곱지수
Smothing	`SMOOTHING`	[number] 기본값: 0.0	평탄화 파라미터
The first radius of search ellipse	`RADIUS_1`	[number] 기본값: 0.0	검색 타원의 첫 번째 (기울기 각도가 0 인 경우 X 축) 반경
The second radius of search ellipse	`RADIUS_2`	[number] 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`	[number] 기본값: 0.0	타원 기울기의 도 단위 각도. 타원은 반시계 방향으로 기울어집니다.
Maximum number of data points to use	`MAX_POINTS`	[number] 기본값: 0	이 값을 초과하는 개수의 포인트를 검색하지 않습니다.
Minimum number of data points to use	`MIN_POINTS`	[number] 기본값: 0	평균을 낼 데이터 포인트의 최소 개수. 이 값보다 적은 포인트가 발견된 그리드 노드는 비어 있다고 간주되어 NODATA 마커로 채워질 것입니다.
Nodata	`NODATA`	[number] 기본값: 0.0	비어 있는 포인트를 채울 NODATA 마커
Z value from field 부가적	`Z_FIELD`	[tablefield: numeric]	보간 작업용 필드
Additional creation options 부가적	`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) 프로세스의 경우: 파이프 문자 (|) 로 복수의 옵션들을 분리하십시오.
Additional command-line parameters 부가적	`EXTRA`	[string] 기본값: None	부가적인 GDAL 명령 줄 옵션을 추가합니다.

다음 페이지에 계속

표 24.150 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Output data type	`DATA_TYPE`	[enumeration] 기본값: 5	산출 래스터 파일의 데이터 유형을 정의합니다. 다음 가운데 하나로 지정할 수 있습니다. • 0 —바이트 (Byte) • 1 —부호 있는 16 비트 정수형 (Int16) • 2 —부호 없는 16 비트 정수형 (UInt16) • 3 —부호 없는 32 비트 정수형 (UInt32) • 4 —부호 있는 32 비트 정수형 (Int32) • 5 —32 비트 부동소수점 실수형 (Float32) • 6 —부동소수점 64 비트 실수형 (Float64) • 7 —복잡 16 비트 정수형 (CInt16) • 8 —복잡 32 비트 정수형 (CInt32) • 9 —복잡 부동소수점 32 비트 실수형 (CFloat32) • 10 —복잡 부동소수점 64 비트 실수형 (CFloat64)
Interpolated (IDW)	`OUTPUT`	[raster] 기본값: `[Save to temporary file]`	보간된 값을 가진 산출 래스터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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 들로네 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.

이 알고리즘은 GDAL 그리드 유틸리티 에서 파생되었습니다.

파라미터

라벨	명칭	유형	설명
Point layer	`INPUT`	[vector: point]	입력 포인트 벡터 레이어
Search distance	`RADIUS`	[number] Default: -1.0	보간할 포인트가 들로네 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`	[number] 기본값: 0.0	비어 있는 포인트를 채울 NODATA 마커
Z value from field 부가적	`Z_FIELD`	[tablefield: numeric]	보간 작업용 필드
Additional creation options 부가적	`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) 프로세스의 경우: 파이프 문자 (`|`) 로 복수의 옵션들을 분리하십시오.
Additional command-line parameters 부가적	`EXTRA`	[string] 기본값: None	부가적인 GDAL 명령 줄 옵션을 추가합니다.

다음 페이지에 계속

표 24.151 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Output data type	`DATA_TYPE`	[enumeration] 기본값: 5	산출 래스터 파일의 데이터 유형을 정의합니다. 다음 가운데 하나로 지정할 수 있습니다. • 0 —바이트 (Byte) • 1 —부호 있는 16 비트 정수형 (Int16) • 2 —부호 없는 16 비트 정수형 (UInt16) • 3 —부호 없는 32 비트 정수형 (UInt32) • 4 —부호 있는 32 비트 정수형 (Int32) • 5 —32 비트 부동소수점 실수형 (Float32) • 6 —부동소수점 64 비트 실수형 (Float64) • 7 —복잡 16 비트 정수형 (CInt16) • 8 —복잡 32 비트 정수형 (CInt32) • 9 —복잡 부동소수점 32 비트 실수형 (CFloat32) • 10 —복잡 부동소수점 64 비트 실수형 (CFloat64)
Interpolated (Linear)	`OUTPUT`	[raster] 기본값: `[Save to temporary file]`	보간된 값을 가진 산출 래스터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

이 알고리즘은 GDAL 그리드 유틸리티 에서 파생되었습니다.

기본 메뉴: *Raster → Analysis*

더 보기:

GDAL 그리드 예제

파라미터

라벨	명칭	유형	설명
Point layer	`INPUT`	[vector: point]	입력 포인트 벡터 레이어
The first radius of search ellipse	`RADIUS_1`	[number] 기본값: 0.0	검색 타원의 첫 번째 (기울기 각도가 0 인 경우 X 축) 반경
The second radius of search ellipse	`RADIUS_2`	[number] 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`	[number] 기본값: 0.0	타원 기울기의 도 단위 각도. 타원은 반시계 방향으로 기울어집니다.
Minimum number of data points to use	`MIN_POINTS`	[number] 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`	[number] 기본값: 0.0	비어 있는 포인트를 채울 NODATA 마커
Z value from field 부가적	`Z_FIELD`	[tablefield: numeric]	보간 작업용 필드
Additional creation options 부가적	`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) 프로세스의 경우: 파이프 문자 (|) 로 복수의 옵션들을 분리하십시오.
Additional command-line parameters 부가적	`EXTRA`	[string] 기본값: None	부가적인 GDAL 명령 줄 옵션을 추가합니다.

다음 페이지에 계속

표 24.152 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Output data type	`DATA_TYPE`	[enumeration] 기본값: 5	산출 래스터 파일의 데이터 유형을 정의합니다. 다음 가운데 하나로 지정할 수 있습니다. • 0 —바이트 (Byte) • 1 —부호 있는 16 비트 정수형 (Int16) • 2 —부호 없는 16 비트 정수형 (UInt16) • 3 —부호 없는 32 비트 정수형 (UInt32) • 4 —부호 있는 32 비트 정수형 (Int32) • 5 —32 비트 부동소수점 실수형 (Float32) • 6 —부동소수점 64 비트 실수형 (Float64) • 7 —복잡 16 비트 정수형 (CInt16) • 8 —복잡 32 비트 정수형 (CInt32) • 9 —복잡 부동소수점 32 비트 실수형 (CFloat32) • 10 —복잡 부동소수점 64 비트 실수형 (CFloat64)
Interpolated (moving average)	`OUTPUT`	[raster] 기본값: `[Save to temporary file]`	산출 래스터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

이 알고리즘은 GDAL 그리드 유틸리티 에서 파생되었습니다.

기본 메뉴: *Raster → Analysis*

더 보기:

GDAL 그리드 예제

파라미터

라벨	명칭	유형	설명
Point layer	`INPUT`	[vector: point]	입력 포인트 벡터 레이어
The first radius of search ellipse	`RADIUS_1`	[number] 기본값: 0.0	검색 타원의 첫 번째 (기울기 각도가 0 인 경우 X 축) 반경
The second radius of search ellipse	`RADIUS_2`	[number] 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`	[number] 기본값: 0.0	타원 기울기의 도 단위 각도. 타원은 반시계 방향으로 기울어집니다.
Nodata	`NODATA`	[number] 기본값: 0.0	비어 있는 포인트를 채울 NODATA 마커
Z value from field 부가적	`Z_FIELD`	[tablefield: numeric]	보간 작업용 필드
Additional creation options 부가적	`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) 프로세스의 경우: 파이프 문자 (|) 로 복수의 옵션들을 분리하십시오.
Additional command-line parameters 부가적	`EXTRA`	[string] 기본값: None	부가적인 GDAL 명령 줄 옵션을 추가합니다.

다음 페이지에 계속

표 24.153 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Output data type	`DATA_TYPE`	[enumeration] 기본값: 5	산출 래스터 파일의 데이터 유형을 정의합니다. 다음 가운데 하나로 지정할 수 있습니다. • 0 —바이트 (Byte) • 1 —부호 있는 16 비트 정수형 (Int16) • 2 —부호 없는 16 비트 정수형 (UInt16) • 3 —부호 없는 32 비트 정수형 (UInt32) • 4 —부호 있는 32 비트 정수형 (Int32) • 5 —32 비트 부동소수점 실수형 (Float32) • 6 —부동소수점 64 비트 실수형 (Float64) • 7 —복잡 16 비트 정수형 (CInt16) • 8 —복잡 32 비트 정수형 (CInt32) • 9 —복잡 부동소수점 32 비트 실수형 (CFloat32) • 10 —복잡 부동소수점 64 비트 실수형 (CFloat64)
Interpolated (Nearest neighbour)	`OUTPUT`	[raster] 기본값: `[Save to temporary file]`	보간된 값을 가진 산출 래스터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

이 알고리즘은 GDAL DEM 유틸리티 에서 파생되었습니다.

기본 메뉴: *Raster → Analysis*

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[raster]	입력 표고 래스터 레이어
Band number	`BAND`	[raster band] 기본값: 1	표고 정보를 담고 있는 밴드
Z factor (vertical exaggeration)	`Z_FACTOR`	[number] 기본값: 1.0	산출 표고 래스터의 높이를 강조하는 인자
Scale (ratio of vert. units to horiz.)	`SCALE`	[number] 기본값: 1.0	수직 단위와 수평 단위의 비율
Azimuth of the light	`AZIMUTH`	[number] 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`	[number] 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] 기본값: False	표고 래스터에서 경계를 생성합니다.
Use Zevenbergen&Thorne formula (instead of the Horn's one)	`ZEVENBERGEN`	[boolean] 기본값: False	매끄러운 풍경을 위해 체벤베르겐 & 손 (Zevenbergen&Thorne) 공식을 활성화합니다.
Combined shading	`COMBINED`	[boolean] 기본값: False	
Multidirectional shading	`MULTIDIRECTIONAL`	[boolean] 기본값: False	
Additional creation options 부가적	`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) 프로세스의 경우: 파이프 문자 (|) 로 복수의 옵션들을 분리하십시오.
Additional command-line parameters 부가적	`EXTRA`	[string] 기본값: None	부가적인 GDAL 명령 줄 옵션을 추가합니다.

다음 페이지에 계속

표 24.154 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Hillshade	OUTPUT	[raster] 기본값: [Save to temporary file]	보간된 값을 가진 산출 래스터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

이 알고리즘은 GDAL 근사 검은색 유틸리티 에서 파생되었습니다.

기본 메뉴: *Raster → Analysis*

파라미터

라벨	명칭	유형	설명
Input layer	INPUT	[raster]	입력 표고 래스터 레이어
How far from black (white)	NEAR	[number] 기본값: 15	픽셀값이 검은색/하얀색/사용자 지정 색상으로 간주되려면 검은색/하얀색/사용자 지정 색상과 얼마나 다를 수 있는지 선택합니다.

다음 페이지에 계속

표 24.155 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Search for nearly white pixels instead of nearly black	`WHITE`	[boolean] 기본값: False	근사 검정색 픽셀 대신 근사 하얀색 (255) 픽셀을 검색합니다.
Additional creation options 부가적	`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) 프로세스의 경우: 파이프 문자 (|) 로 복수의 옵션들을 분리하십시오.
Additional command-line parameters 부가적	`EXTRA`	[string] 기본값: None	부가적인 GDAL 명령 줄 옵션을 추가합니다.
Nearblack	`OUTPUT`	[raster] 기본값: `[Save to temporary file]`	산출 래스터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

이 알고리즘은 GDAL 근접성 유틸리티 에서 파생되었습니다.

기본 메뉴: *Raster → Analysis*

파라미터

라벨	명칭	유형	설명
Input layer	INPUT	[raster]	입력 표고 래스터 레이어
Band number	BAND	[raster band] 기본값: 1	표고 정보를 담고 있는 밴드
A list of pixel values in the source image to be considered target pixels 부가적	VALUES	[string] 기본값: ''	소스 이미지에서 대상 픽셀로 간주되기 위한 대상 픽셀값의 목록입니다. 이 파라미터를 지정하지 않는 경우, 0 값이 아닌 모든 픽셀을 대상 픽셀로 간주할 것입니다.
Distance units	UNITS	[enumeration] 기본값: 1	생성된 거리를 픽셀 단위로 나타낼지 또는 지리참조 좌표로 나타낼지 설정합니다. 다음 가운데 하나를 선택할 수 있습니다: • 0 — 지리참조 좌표 • 1 — 픽셀 좌표
The maximum distance to be generated 부가적	MAX_DISTANCE	[number] 기본값: 0.0	생성할 거리의 최대값입니다. 이 거리를 초과하는 픽셀에 대해서는 NODATA 값을 사용할 것입니다. NODATA 값을 지정하지 않는 경우, 산출 밴드에 산출 밴드의 NODATA 값을 쿼리할 것입니다. 산출 밴드가 NODATA 값을 가지고 있지 않은 경우, 65535 값을 사용할 것입니다. *Distance units* 의 값에 따라 거리를 해석합니다.
Value to be applied to all pixels that are within the maxdist of target pixels 부가적	REPLACE	[number] 기본값: 0.0	대상 픽셀에서 최대 거리보다 가까이 있는 모든 픽셀에 거리 값 대신 적용할 값을 지정합니다.
Nodata value to use for the destination proximity raster 부가적	NODATA	[number] 기본값: 0.0	산출 래스터에 사용할 NODATA 값을 지정합니다.
Additional creation options 부가적	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) 프로세스의 경우: 파이프 문자 (|) 로 복수의 옵션들을 분리하십시오.
Additional command-line parameters 부가적	EXTRA	[string] 기본값: None	부가적인 GDAL 명령 줄 옵션을 추가합니다.

다음 페이지에 계속

표 24.156 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Output data type	`DATA_TYPE`	[enumeration] 기본값: 5	산출 래스터 파일의 데이터 유형을 정의합니다. 다음 가운데 하나로 지정할 수 있습니다. • 0 —바이트 (Byte) • 1 —부호 있는 16 비트 정수형 (Int16) • 2 —부호 없는 16 비트 정수형 (UInt16) • 3 —부호 없는 32 비트 정수형 (UInt32) • 4 —부호 있는 32 비트 정수형 (Int32) • 5 —32 비트 부동소수점 실수형 (Float32) • 6 —부동소수점 64 비트 실수형 (Float64) • 7 —복잡 16 비트 정수형 (CInt16) • 8 —복잡 32 비트 정수형 (CInt32) • 9 —복잡 부동소수점 32 비트 실수형 (CFloat32) • 10 —복잡 부동소수점 64 비트 실수형 (CFloat64)
Proximity map	`OUTPUT`	[raster] 기본값: `[Save to temporary file]`	산출 래스터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

이 알고리즘은 GDAL DEM 유틸리티 에서 파생되었습니다.

기본 메뉴: *Raster → Analysis*

파라미터

라벨	명칭	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.
Additional creation options 부가적	`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) 프로세스의 경우: 파이프 문자 (|) 로 복수의 옵션들을 분리하십시오.
Roughness	`OUTPUT`	[raster] 기본값: `[Save to temporary file]`	산출 래스터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

이 알고리즘은 GDAL 거르기 (sieve) 유틸리티 에서 파생되었습니다.

기본 메뉴: *Raster → Analysis*

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[raster]	입력 표고 래스터 레이어
Threshold	`THRESHOLD`	[number] 기본값: 10	이 크기보다 작은 래스터 폴리곤만 제거할 것입니다.
Use 8-connectedness	`EIGHT_CONNECTEDNESS`	[boolean] 기본값: False	4 연결성 (connectedness) 대신 8 연결성을 사용
Do not use the default validity mask for the input band	`NO_MASK`	[boolean] 기본값: False	
Validity mask 부가적	`MASK_LAYER`	[raster]	기본값 대신 사용할 유효성 마스크
Additional command-line parameters 부가적	`EXTRA`	[string] 기본값: None	부가적인 GDAL 명령 줄 옵션을 추가합니다.
Sieved	`OUTPUT`	[raster] 기본값: `[Save to temporary file]`	산출 래스터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

경사도

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.

이 알고리즘은 GDAL DEM 유틸리티 에서 파생되었습니다.

기본 메뉴: *Raster → Analysis*

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[raster]	입력 표고 래스터 레이어
Band number	`BAND`	[raster band] 기본값: 1	표고 정보를 담고 있는 밴드
Ratio of vertical units to horizontal	`SCALE`	[number] 기본값: 1.0	수직 단위와 수평 단위의 비율
Slope expressed as percent (instead of degrees)	`AS_PERCENT`	[boolean] 기본값: False	경사도를 도 단위 대신 백분율로 표현합니다.
Compute edges	`COMPUTE_EDGES`	[boolean] 기본값: False	표고 래스터에서 경계를 생성합니다.
Use Zevenbergen&Thorne formula (instead of the Horn's one)	`ZEVENBERGEN`	[boolean] 기본값: False	매끄러운 풍경을 위해 체벤베르겐 & 손 (Zevenbergen&Thorne) 공식을 활성화합니다.
Additional creation options 부가적	`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) 프로세스의 경우: 파이프 문자 (|) 로 복수의 옵션들을 분리하십시오.
Additional command-line parameters 부가적	`EXTRA`	[string] 기본값: None	부가적인 GDAL 명령 줄 옵션을 추가합니다.
Slope	`OUTPUT`	[raster] 기본값: `[Save to temporary file]`	산출 래스터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

이 알고리즘은 GDAL DEM 유틸리티 에서 파생되었습니다.

기본 메뉴: *Raster → Analysis*

파라미터

라벨	명칭	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.
Additional creation options 부가적	`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) 프로세스의 경우: 파이프 문자 (|) 로 복수의 옵션들을 분리하십시오.
Terrain Ruggedness Index	`OUTPUT`	[raster] 기본값: `[Save to temporary file]`	산출 래스터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

이 알고리즘은 GDAL DEM 유틸리티 에서 파생되었습니다.

기본 메뉴: *Raster → Analysis*

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[raster]	입력 표고 래스터 레이어
Band number	`BAND`	[raster band] 기본값: 1	표고 값으로 사용할 밴드의 번호
Compute edges	`COMPUTE_EDGES`	[boolean] 기본값: False	표고 래스터에서 경계를 생성합니다.
Additional creation options 부가적	`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) 프로세스의 경우: 파이프 문자 (|) 로 복수의 옵션들을 분리하십시오.
Terrain Ruggedness Index	`OUTPUT`	[raster] 기본값: `[Save to temporary file]`	산출 래스터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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	명칭	유형	설명
Input layer	`INPUT`	[raster]	변환할 래스터 레이어
Band number	`BAND`	[raster band] 기본값: 입력 레이어의 첫 번째 밴드	래스터가 다중 밴드인 경우, 변환하려는 밴드를 선택하십시오.
Output comma-separated values	`CSV`	[boolean] 기본값: False	산출 파일이 쉼표 구분 값 (CSV) 유형이어야 하는지를 설정합니다.
XYZ ASCII file	`OUTPUT`	[file] 기본값: `[Save to temporary file]`	산출 파일을 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

이 알고리즘은 GDAL pct2rgb 유틸리티 에서 파생되었습니다.

기본 메뉴: *Raster → Conversion*

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[raster]	입력 8 비트 래스터 이미지
Band number	`BAND`	[raster band] 기본값: 입력 레이어의 첫 번째 밴드	래스터가 다중 밴드인 경우, 변환하려는 밴드를 선택하십시오.
Generate a RGBA file	`RGBA`	[boolean] 기본값: False	산출 파일이 RGBA 유형이어야 하는지를 설정합니다.
PCT to RGB	`OUTPUT`	[file] 기본값: `[Save to temporary file]`	산출 파일을 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

이 알고리즘은 GDAL 폴리곤화 유틸리티 에서 파생되었습니다.

기본 메뉴: *Raster → Conversion*

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[raster]	입력 래스터 레이어
Band number	`BAND`	[raster band] 기본값: 입력 레이어의 첫 번째 밴드	래스터가 다중 밴드인 경우, 사용하려는 밴드를 선택하십시오.
Name of the field to create	`FIELD`	[string] 기본값: 'DN'	연속 영역의 속성을 가질 필드의 명칭을 지정합니다.
Use 8-connectedness	`EIGHT_CONNECTEDNESS`	[boolean] 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).
Additional command-line parameters 부가적	`EXTRA`	[string] 기본값: None	부가 GDAL 명령 줄 옵션을 추가합니다.

다음 페이지에 계속

표 24.164 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Vectorized	OUTPUT	[vector: polygon] 기본값: [Save to temporary file]	산출 폴리곤 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

이 알고리즘은 GDAL 번역 (translate) 유틸리티 에서 파생되었습니다.

파라미터

라벨	명칭	유형	설명
Input layer	INPUT	[raster]	입력 래스터 레이어
Selected band(s)	BANDS	[raster band] [list] 기본값: None	새 래스터를 생성하기 위해 사용할 밴드의 정렬된 목록
Additional creation options 부가적	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) 프로세스의 경우: 파이프 문자 (|) 로 복수의 옵션들을 분리하십시오.

다음 페이지에 계속

표 24.165 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Output data type	DATA_TYPE	[enumeration] 기본값: 0	산출 래스터 파일의 데이터 유형을 정의합니다. 다음 가운데 하나로 지정할 수 있습니다. • 0 —입력 레이어의 데이터 유형 사용 • 1 —바이트 (Byte) • 2 —부호 있는 16 비트 정수형 (Int16) • 3 —부호 없는 16 비트 정수형 (UInt16) • 4 —부호 없는 32 비트 정수형 (UInt32) • 5 —부호 있는 32 비트 정수형 (Int32) • 6 —부동소수점 32 비트 실수형 (Float32) • 7 —부동소수점 64 비트 실수형 (Float64) • 8 —복잡 16 비트 정수형 (CInt16) • 9 —복잡 32 비트 정수형 (CInt32) • 10 —복잡 부동소수점 32 비트 실수형 (CFloat32) • 11 —복잡 부동소수점 64 비트 실수형 (CFloat64)
Converted	OUTPUT	[raster] 기본값: Save to temporary file	산출 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

이 알고리즘은 GDAL rgb2pct 유틸리티 에서 파생되었습니다.

기본 메뉴: *Raster → Conversion*

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[raster]	입력 (RGB) 래스터 레이어
Number of colors	`NCOLORS`	[number] 기본값: 2	산출 이미지가 담게 될 색상의 개수. 2 에서 256 까지의 값을 사용할 수 있습니다.
RGB to PCT	`OUTPUT`	[raster] 기본값: `[Save to temporary file]`	산출 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

이 알고리즘은 GDAL 번역 (translate) 유틸리티 에서 파생되었습니다.

기본 메뉴: *Raster → Conversion*

파라미터

라벨	명칭	유형	설명
Input layer	INPUT	[raster]	입력 래스터 레이어
Override the projection of the output file 부가적	TARGET_CRS	[crs]	산출 파일용 투영체를 지정합니다.
Assign a specified nodata value to output bands 부가적	NODATA	[number] 기본값: 설정하지 않음	산출 래스터에 있는 NODATA 에 사용할 값을 정의합니다.
Copy all subdatasets of this file to individual output files	COPY_SUBDATASETS	[boolean] 기본값: False	하위 데이터셋 (subdataset) 용 개별 파일을 생성합니다.
Additional creation options 부가적	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) 프로세스의 경우: 파이프 문자 (|) 로 복수의 옵션들을 분리하십시오.
Additional command-line parameters 부가적	EXTRA	[string] 기본값: None	부가 GDAL 명령 줄 옵션을 추가합니다.

다음 페이지에 계속

표 24.167 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Output data type	DATA_TYPE	[enumeration] 기본값: 0	산출 래스터 파일의 데이터 유형을 정의합니다. 다음 가운데 하나로 지정할 수 있습니다. • 0 —입력 레이어의 데이터 유형 사용 • 1 —바이트 (Byte) • 2 —부호 있는 16 비트 정수형 (Int16) • 3 —부호 없는 16 비트 정수형 (UInt16) • 4 —부호 없는 32 비트 정수형 (UInt32) • 5 —부호 있는 32 비트 정수형 (Int32) • 6 —부동소수점 32 비트 실수형 (Float32) • 7 —부동소수점 64 비트 실수형 (Float64) • 8 —복잡 16 비트 정수형 (CInt16) • 9 —복잡 32 비트 정수형 (CInt32) • 10 —복잡 부동소수점 32 비트 실수형 (CFloat32) • 11 —복잡 부동소수점 64 비트 실수형 (CFloat64)
Converted	OUTPUT	[raster] 기본값: [Save to temporary file]	산출 (번역된) 래스터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

이 알고리즘은 GDAL 왜곡 (warp) 유틸리티 에서 파생되었습니다.

기본 메뉴: *Raster ▸ Extraction*

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[raster]	입력 래스터
Clipping extent	`EXTENT`	[extent]	산출 래스터에 사용될 범위. 지정한 경계 상자 안에 들어오는 픽셀만 산출물에 포함될 것입니다.
Assign a specified nodata value to output bands 부가적	`NODATA`	[number] 기본값: None	산출 래스터에 있는 NODATA 에 사용할 값을 정의합니다.
Additional creation options 부가적	`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) 프로세스의 경우: 파이프 문자 (|) 로 복수의 옵션들을 분리하십시오.
Output data type	`DATA_TYPE`	[enumeration] 기본값: 0	산출 래스터 파일의 포맷을 정의합니다. 옵션: • 0 —입력 레이어 데이터 유형 사용 • 1 —바이트 (Byte) • 2 —부호 있는 16 비트 정수형 (Int16) • 3 —부호 없는 16 비트 정수형 (UInt16) • 4 —부호 없는 32 비트 정수형 (UInt32) • 5 —부호 있는 32 비트 정수형 (Int32) • 6 —부동소수점 32 비트 실수형 (Float32) • 7 —부동소수점 64 비트 실수형 (Float64) • 8 —복잡 16 비트 정수형 (CInt16) • 9 —복잡 32 비트 정수형 (CInt32) • 10 —복잡 부동소수점 32 비트 실수형 (CFloat32) • 11 —복잡 부동소수점 64 비트 실수형 (CFloat64)

다음 페이지에 계속

표 24.168 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Additional command-line parameters 부가적	EXTRA	[string] 기본값: None	부가 GDAL 명령 줄 옵션을 추가합니다.
Clipped (extent)	OUTPUT	[raster] 기본값: '[Save to temporary file]'	산출 래스터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

이 알고리즘은 GDAL 왜곡 (warp) 유틸리티 에서 파생되었습니다.

기본 메뉴: *Raster → Extraction*

파라미터

라벨	명칭	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

다음 페이지에 계속

표 24.169 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Assign a specified nodata value to output bands 부가적	NODATA	[number] 기본값: None	산출 래스터에 있는 NODATA에 사용할 값을 정의합니다.
Create an output alpha band	ALPHA_BAND	[boolean] 기본값: False	산출물용 알파 밴드를 생성합니다. 그러면 알파 밴드가 픽셀에 대한 투명도 값을 담습니다.
Match the extent of the clipped raster to the extent of the mask layer	CROP_TO_CUTLINE	[boolean] 기본값: True	이 파라미터를 활성화하면 산출 래스터에 벡터 레이어 범위를 적용합니다.
Keep resolution of input raster	KEEP_RESOLUTION	[boolean] 기본값: False	이 파라미터를 활성화하면 산출 래스터의 해상도를 변경하지 않을 것입니다.
Set output file resolution	SET_RESOLUTION	[boolean] 기본값: False	이 파라미터를 활성화하면 산출물 해상도 (셀 크기) 를 지정합니다.
X Resolution to output bands 부가적	X_RESOLUTION	[number] 기본값: None	산출 래스터의 셀의 너비
Y Resolution to output band 부가적	Y_RESOLUTION	[number] 기본값: None	산출 래스터의 셀의 높이
Use multithreaded warping implementation	MULTITHREADING	[boolean] 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 creation options 부가적	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) 프로세스의 경우: 파이프 문자 (|) 로 복수의 옵션들을 분리하십시오.

다음 페이지에 계속

표 24.169 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Output data type	DATA_TYPE	[enumeration] 기본값: 0	산출 래스터 파일의 포맷을 정의합니다. 옵션: • 0 —입력 레이어 데이터 유형 사용 • 1 —바이트 (Byte) • 2 —부호 있는 16 비트 정수형 (Int16) • 3 —부호 없는 16 비트 정수형 (UInt16) • 4 —부호 없는 32 비트 정수형 (UInt32) • 5 —부호 있는 32 비트 정수형 (Int32) • 6 —부동소수점 32 비트 실수형 (Float32) • 7 —부동소수점 64 비트 실수형 (Float64) • 8 —복잡 16 비트 정수형 (CInt16) • 9 —복잡 32 비트 정수형 (CInt32) • 10 —복잡 부동소수점 32 비트 실수형 (CFloat32) • 11 —복잡 부동소수점 64 비트 실수형 (CFloat64)
Additional command-line parameters 부가적	EXTRA	[string] 기본값: None	부가 GDAL 명령 줄 옵션을 추가합니다.
Clipped (mask)	OUTPUT	[raster] 기본값: '[Save to temporary file]'	산출 래스터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

이 알고리즘은 GDAL 등고선 유틸리티 에서 파생되었습니다.

기본 메뉴: *Raster → Extraction*

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[raster]	입력 래스터
Band number	`BAND`	[raster band] 기본값: 1	등고선을 생성할 래스터 밴드
Interval between contour lines	`INTERVAL`	[number] 기본값: 10.0	등고선 사이의 간격을 지정한 표고 래스터의 단위로 정의합니다. (최소값: 0)
Attribute name (if not set, no elevation attribute is attached) 부가적	`FIELD_NAME`	[string] 기본값: 'ELEV'	표고를 넣을 속성을 위한 이름을 지정합니다.
Offset from zero relative to which to interpret intervals 부가적	`OFFSET`	[number] 기본값: 0.0	
Produce 3D vector	`CREATE_3D`	[boolean] 기본값: False	2 차원 대신 3 차원 벡터를 생성하도록 강제합니다. 모든 꼭짓점에 표고를 포함시킵니다.
Treat all raster values as valid	`IGNORE_NODATA`	[boolean] 기본값: False	데이터셋에 있는 모든 NODATA 값을 무시합니다.
Input pixel value to treat as "nodata" 부가적	`NODATA`	[number] 기본값: None	산출 래스터에 있는 NODATA 에 사용할 값을 정의합니다.
Additional command-line parameters 부가적	`EXTRA`	[string] 기본값: None	추가적인 GDAL 명령 줄 옵션을 추가합니다. 대응하는 GDAL 유틸리티 문서를 참조합니다.

다음 페이지에 계속

표 24.170 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Contours	OUTPUT	[vector: line] 기본값: '[Save to temporary file]'	산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

이 알고리즘은 GDAL 등고선 유틸리티 에서 파생되었습니다.

파라미터

라벨	명칭	유형	설명
Input layer	INPUT	[raster]	입력 래스터
Band number	BAND	[raster band] 기본값: 1	등고선을 생성할 래스터 밴드
Interval between contour lines	INTERVAL	[number] 기본값: 10.0	등고선 사이의 간격을 지정한 표고 래스터의 단위로 정의합니다. (최소값: 0)
Offset from zero relative to which to interpret intervals 부가적	OFFSET	[number] 기본값: 0.0	
Attribute name for minimum elevation of contour polygon 부가적	FIELD_NAME_MIN	[string] 기본값: 'ELEV_MIN'	등고선 폴리곤의 최저 표고를 넣을 속성을 위한 이름을 지정합니다. 아무 이름도 지정하지 않으면 최저 표고 속성을 추가하지 않습니다.

다음 페이지에 계속

표 24.171 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Attribute name for maximum elevation of contour polygon 부가적	FIELD_NAME_MAX	[string] 기본값: 'ELEV_MAX'	등고선 폴리곤의 최고 표고를 넣을 속성을 위한 이름을 지정합니다. 아무 이름도 지정하지 않으면 최고 표고 속성을 추가하지 않습니다.
Produce 3D vector	CREATE_3D	[boolean] 기본값: False	2 차원 대신 3 차원 벡터를 생성하도록 강제합니다. 모든 꼭짓점에 표고를 포함시킵니다.
Treat all raster values as valid	IGNORE_NODATA	[boolean] 기본값: False	데이터셋에 있는 모든 NODATA 값을 무시합니다.
Input pixel value to treat as "nodata" 부가적	NODATA	[number] 기본값: None	산출 래스터에 있는 NODATA 에 사용할 값을 정의합니다.
Additional command-line parameters 부가적	EXTRA	[string] 기본값: None	추가적인 GDAL 명령 줄 옵션을 추가합니다. 대응하는 GDAL 유틸리티 문서를 참조합니다.
Contours	OUTPUT	[vector: polygon] 기본값: '[Save to temporary file]'	산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

이 알고리즘은 GDAL addo 유틸리티 에서 파생되었습니다.

기본 메뉴: *Raster → Miscellaneous*

파라미터

기본 파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[raster]	입력 래스터 레이어
Remove all existing overviews	`CLEAN`	[boolean] 기본값: False	래스터에서 기존 오버뷰를 제거합니다. 기본값은 제거하지 않는다입니다.

고급 파라미터

라벨	명칭	유형	설명
Overview levels	LEVELS	[string] 기본값: '2 4 8 16'	입력 래스터 레이어의 원본 해상도에서 계산할 오버뷰 수준의 개수를 정의합니다. 기본적으로 4 개의 수준을 계산합니다.
Resampling method 부가적	RESAMPLING	[enumeration] 기본값: 0	정의된 리샘플링 방법으로 오버뷰를 계산합니다. 다음과 같은 리샘플링 방법을 사용할 수 있습니다: • 0 – 최근접 이웃 (nearest) • 1 – 평균 (average) • 2 – 가우스 (gauss) • 3 – 3 차 회선 (Cubic Convolution) (cubic) • 4 – B-스플라인 회선 (cubicspline) • 5 – 란초시 창이 있는 싱크 (Lánczos Windowed Sinc) (lanczos) • 6 – 평균 MP (average_mp) • 7 – 스펙트럼/페이즈 공간에서의 평균 (Average in Mag/Phase Space) (average_magphase) • 8 – 모드 (mode)
Overviews format 부가적	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: • 0 – 내부 저장 (가능한 경우) • 1 – 외부 저장 (GTiff .ovr) • 2 – 외부 저장 (ERDAS Imagine .aux)
Additional command-line parameters 부가적	EXTRA	[string] 기본값: None	부가 GDAL 명령 줄 옵션을 추가합니다.

산출물

라벨	명칭	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 **명칭** 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.

이 알고리즘은 GDAL buildvrt 유틸리티 에서 파생되었습니다.

기본 메뉴: *Raster → Miscellaneous*

파라미터

기본 파라미터

라벨	명칭	유형	설명
Input layers	`INPUT`	[raster] [list]	GDAL 이 지원하는 래스터 레이어
Resolution	`RESOLUTION`	[enumeration] 기본값: 0	모자이크의 산출 해상도. 기본적으로 래스터 파일들의 평균 해상도를 선택할 것입니다. 옵션: • 0 — 평균 (`average`) • 1 — 최고 (`highest`) • 2 — 최저 (`lowest`)
Place each input file into a separate band	`SEPARATE`	[boolean] 기본값: False	'참'으로 설정하면 VRT 의 개별 적층 (stacked) 밴드에 각 래스터 파일을 넣도록 정의할 수 있습니다.
Allow projection difference	`PROJ_DIFFERENCE`	[boolean] 기본값: False	산출 밴드들이 입력 래스터 레이어들의 투영체에서 추출한 서로 다른 투영체를 가지도록 허용합니다.
Virtual	`OUTPUT`	[raster] 기본값: `[Save to temporary file]`	산출 래스터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장

고급 파라미터

라벨	명칭	유형	설명
Add alpha mask band to VRT when source raster has none	ADD_ALPHA	[boolean] 기본값: False	소스 래스터에 알파 마스크 밴드가 없을 경우, VRT 에 알파 마스크 밴드를 추가합니다.
Override projection for the output file 부가적	ASSIGN_CRS	[crs] 기본값: None	산출 파일용 투영체를 무시합니다. 재투영하지 않습니다.
Resampling algorithm	RESAMPLING	[enumeration] 기본값: 0	사용할 리샘플링 알고리즘입니다. 다음 가운데 하나를 선택할 수 있습니다: • 0 —최근접 이웃 (nearest) • 1 —이중선형 (bilinear) • 2 —3 차 회선 (Cubic Convolution) (cubic) • 3 —B-스플라인 회선 (cubicspline) • 4 —란초시 창이 있는 싱크 (Lánczos Windowed Sinc) (lanczos) • 5 —평균 (average) • 6 —모드 (mode)
Nodata value(s) for input bands (space separated) 부가적	SRC_NODATA	[string] 기본값: None	입력 밴드 (들) 를 위한 공백으로 구분된 NODATA 값 (들)
Additional command-line parameters	EXTRA	[string] 기본값: None	부가 GDAL 명령 줄 옵션을 추가합니다.

산출물

라벨	명칭	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 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

gdal2tiles

OSGeo 타일 맵 서비스 사양 을 따라 작은 타일들과 메타데이터를 담은 디렉터리를 생성합니다. OpenGIS 웹 맵 타일 서비스 구현 표준 도 참조하세요. 구글 지도, 오픈레이어, 리플렛 (Leaflet) 을 기반으로 한 뷰어를 가진 간단한 웹 페이지도 함께 생성합니다. 웹 브라우저에서 사용자 맵을 온라인으로 탐색하려면, 웹 서버에 생성된 디렉터리를 업로드하기만 하면 됩니다.

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.

이 알고리즘은 GDAL gdal2tiles 유틸리티 에서 파생되었습니다.

파라미터

기본 파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[raster]	GDAL 이 지원하는 래스터 레이어
Tile cutting profile	`PROFILE`	[enumeration] 기본값: 0	다음 가운데 하나로 지정할 수 있습니다: • 0 —메르카토르 (`mercator`) • 1 —측지선 (`geodetic`) • 2 —래스터 (`raster`)
Zoom levels to render 부가적	`ZOOM`	[string] 기본값: ' '	
Web viewer to generate	`VIEWER`	[enumerate] 기본값: 0	다음 가운데 하나로 지정할 수 있습니다: • 0 —모두 (`all`) • 1 —구글 지도 (`google`) • 2 —오픈레이어 (`openlayers`) • 3 —리플렛 (`leaflet`) • 4 —없음 (`none`)
Title of the map 부가적	`TITLE`	[string] 기본값: ' '	
Copyright of the map	`COPYRIGHT`	[string] 기본값: ' '	
Output directory	`OUTPUT`	[folder] 기본값: `[Save to temporary folder]`	타일을 저장할 산출물 폴더를 지정합니다. 다음 가운데 하나를 선택할 수 있습니다: • 임시 디렉터리에 저장 • 디렉터리에 저장

고급 파라미터

라벨	명칭	유형	설명
Resampling method	RESAMPLING	[enumeration] 기본값: 0	사용할 리샘플링 알고리즘입니다. 다음 가운데 하나를 선택할 수 있습니다: • 0 —평균 (average) • 1 —최근접 이웃 (nearest) • 2 —이중선형 (bilinear) • 3 —3 차 회선 (Cubic Convolution) (cubic) • 4 —B-스플라인 회선 (cubicspline) • 5 —란초시 창이 있는 싱크 (Lánczos Windowed Sinc) (lanczos) • 6 —에일리어싱 제거 (antialias)
The spatial reference system used for the source input data 부가적	SOURCE_CRS	[crs] 기본값: None	
Transparency value to assign to the input data 부가적	NODATA	[number] 기본값: 0.0	
URL address where the generated tiles are going to be published 부가적	URL	[string] 기본값: ''	
Google Maps API key (http://code.google.com/apis/maps/signup.html) 부가적	GOOGLE_KEY	[string] 기본값: ''	사용자의 구글 지도 API 키
Bing Maps API key (https://www.bingmapsportal.com/) 부가적	BING_KEY	[string] 기본값: ''	사용자의 빙 맵 API 키
Generate only missing files	RESUME	[boolean] 기본값: False	
Generate KML for Google Earth	KML	[boolean] 기본값: False	
Avoid automatic generation of KML files for EPSG:4326	NO_KML	[boolean] 기본값: False	

산출물

라벨	명칭	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 **명칭** 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.

이 알고리즘은 GDAL 병합 유틸리티 에서 파생되었습니다.

기본 메뉴: *Raster → Miscellaneous*

파라미터

기본 파라미터

라벨	명칭	유형	설명
Input layers	INPUT	[raster] [list]	입력 래스터 레이어
Grab pseudocolor table from first layer	PCT	[boolean] 기본값: False	색상 작업에 첫 번째 레이어의 의사색상 테이블을 사용할 것입니다.
Place each input file into a separate band	SEPARATE	[boolean] 기본값: False	개별 밴드에 각 입력 파일을 배치합니다.

다음 페이지에 계속

표 24.175 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Output data type	DATA_TYPE	[enumeration] 기본값: 5	산출 래스터 파일의 포맷을 정의합니다. 다음 가운데 하나로 지정할 수 있습니다: • 0 —바이트 (Byte) • 1 —부호 있는 16 비트 정수형 (Int16) • 2 —부호 없는 16 비트 정수형 (UInt16) • 3 —부호 없는 32 비트 정수형 (UInt32) • 4 —부호 있는 32 비트 정수형 (Int32) • 5 —32 비트 부동소수점 실수형 (Float32) • 6 —부동소수점 64 비트 실수형 (Float64) • 7 —복잡 16 비트 정수형 (CInt16) • 8 —복잡 32 비트 정수형 (CInt32) • 9 —복잡 부동소수점 32 비트 실수형 (CFloat32) • 10 —복잡 부동소수점 64 비트 실수형 (CFloat64)
Merged	OUTPUT	[raster] 기본값: [Save to temporary file]	산출 래스터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장

고급 파라미터

라벨	명칭	유형	설명
Input pixel value to treat as "nodata" 부가적	NODATA_INPUT	[number] 기본값: None	병합되는 파일들에서 이 픽셀값을 가진 픽셀을 무시합니다.
Assign specified "nodata" value to output 부가적	NODATA_OUTPUT	[number] 기본값: None	산출 밴드에 지정한 NODATA 값을 할당합니다.
Additional creation options 부가적	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) 프로세스의 경우: 파이프 문자 (|) 로 복수의 옵션들을 분리하십시오.
Additional command-line parameters	EXTRA	[string] 기본값: None	부가 GDAL 명령 줄 옵션을 추가합니다.

산출물

라벨	명칭	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 **명칭** 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.

GDAL 영상융합 을 참조하세요.

파라미터

기본 파라미터

라벨	명칭	유형	설명
Spectral dataset	`SPECTRAL`	[raster]	입력 (스펙트럼) 래스터 레이어
Panchromatic dataset	`PANCHROMATIC`	[raster]	입력 (전정색 (□□□)) 래스터 레이어
산출물	`OUTPUT`	[raster] 기본값: `[Save to temporary file]`	산출 (선명해진) 래스터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장

고급 파라미터

라벨	명칭	유형	설명
Resampling algorithm	RESAMPLING	[enumeration] 기본값: 2	사용할 리샘플링 알고리즘입니다. 다음 가운데 하나를 선택할 수 있습니다: • 0 —최근접 이웃 (nearest) • 1 —이중선형 (bilinear) • 2 —3 차 회선 (Cubic Convolution) (cubic) • 3 —B-스플라인 회선 (cubicspline) • 4 —란초시 창이 있는 싱크 (Lánczos Windowed Sinc) (lanczos) • 5 —평균 (average)
Additional creation options 부가적	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) 프로세스의 경우: 파이프 문자 (|) 로 복수의 옵션들을 분리하십시오.
Additional command-line parameters 부가적	EXTRA	[string] 기본값: None	부가 GDAL 명령 줄 옵션을 추가합니다.

산출물

라벨	명칭	유형	설명
산출물	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 **명칭** 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.

GDAL 래스터 계산기 유틸리티 문서 를 참조하세요.

더 보기:

래스터 계산기

파라미터

기본 파라미터

라벨	명칭	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 부가적	`INPUT_B`	[raster] 기본값: None	두 번째 입력 래스터 레이어
Number of raster band for B 부가적	`BAND_B`	[raster band]	입력 레이어 B 의 밴드
Input layer C 부가적	`INPUT_C`	[raster] 기본값: None	세 번째 입력 래스터 레이어
Number of raster band for C 부가적	`BAND_C`	[raster band]	입력 레이어 C 의 밴드
Input layer D 부가적	`INPUT_D`	[raster] 기본값: None	네 번째 입력 래스터 레이어
Number of raster band for D 부가적	`BAND_D`	[raster band]	입력 레이어 D 의 밴드
Input layer E 부가적	`INPUT_E`	[raster] 기본값: None	다섯 번째 입력 래스터 레이어
Number of raster band for E 부가적	`BAND_E`	[raster band]	입력 레이어 E 의 밴드
Input layer F 부가적	`INPUT_F`	[raster]	여섯 번째 입력 래스터 레이어
Number of raster band for F 부가적	`BAND_F`	[raster band] 기본값: None	입력 레이어 F 의 밴드

다음 페이지에 계속

표 24.177 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Calculation in gdalnumeric syntax using +-/* or any numpy array functions (i.e. logical_and())	FORMULA	[string] 기본값: ' '	계산 공식입니다. 다음은 몇몇 예시입니다: • A*(A>0) —래스터 A 의 값이 0 을 초과하는 경우 래스터 A 의 값을 산출합니다. 0 을 초과하지 않으면, 0 을 산출합니다. • 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. • 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. • sqrt(A*A+B*B) —래스터 A 의 값의 제곱과 래스터 B 의 값의 제곱의 합의 제곱근을 산출합니다.
Set output nodata value 부가적	NO_DATA	[number] 기본값: None	NODATA 용으로 사용할 값
Output raster type	RTYPE	[enumeration] 기본값: 5	산출 래스터 파일의 포맷을 정의합니다. 다음 가운데 하나로 지정할 수 있습니다: • 0 —바이트 (Byte) • 1 —부호 있는 16 비트 정수형 (Int16) • 2 —부호 없는 16 비트 정수형 (UInt16) • 3 —부호 없는 32 비트 정수형 (UInt32) • 4 —부호 있는 32 비트 정수형 (Int32) • 5 —32 비트 부동소수점 실수형 (Float32) • 6 —부동소수점 64 비트 실수형 (Float64)
Calculated	OUTPUT	[raster] 기본값: [Save to temporary file]	산출 (계산된) 래스터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장

고급 파라미터

라벨	명칭	유형	설명
Additional creation options 부가적	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) 프로세스의 경우: 파이프 문자 (|) 로 복수의 옵션들을 분리하십시오.
Additional command-line parameters 부가적	EXTRA	[string] 기본값: ''	부가 GDAL 명령 줄 옵션을 추가합니다.

산출물

라벨	명칭	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 **명칭** 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.

이 알고리즘은 GDAL 정보 유틸리티 에서 파생되었습니다.

기본 메뉴: *Raster → Miscellaneous*

파라미터

기본 파라미터

라벨	명칭	유형	설명
Input layer	INPUT	[raster]	입력 래스터 레이어

다음 페이지에 계속

표 24.179 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Force computation of the actual min/max values for each band	`MIN_MAX`	[boolean] 기본값: False	데이터셋에 있는 각 밴드에 대해 실제 최소/최대 값을 계산하도록 강제합니다.
Read and display image statistics (force computation if necessary)	`STATS`	[boolean] 기본값: False	이미지 통계를 읽고 표시합니다. 이미지에 저장된 통계가 없는 경우 강제로 계산합니다.
Suppress GCP info	`NO_GCP`	[boolean] 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] 기본값: False	메타데이터를 인쇄하지 않습니다. 일부 데이터셋은 수많은 메타데이터 문자열을 담고 있을 수도 있습니다.
Layer information	`OUTPUT`	[html] 기본값: `[Save to temporary file]`	산출물용 HTML 파일을 지정합니다. 다음 가운데 하나를 선택할 수 있습니다: • 임시 파일로 저장 • 파일로 저장

고급 파라미터

라벨	명칭	유형	설명
Additional command-line parameters 부가적	`EXTRA`	[string] 기본값: None	부가 GDAL 명령 줄 옵션을 추가합니다.

산출물

라벨	명칭	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 **명칭** 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.

이 알고리즘은 GDAL 타일 재작업 유틸리티 에서 파생되었습니다.

파라미터

기본 파라미터

라벨	명칭	유형	설명
Input files	`INPUT`	[raster] [list]	입력 래스터 파일
Tile width	`TILE_SIZE_X`	[number] 기본값: 256	타일의 픽셀 단위 너비 (최소값 0)
Tile height	`TILE_SIZE_Y`	[number] 기본값: 256	타일의 픽셀 단위 높이 (최소값 0)
Overlap in pixels between consecutive tiles	`OVERLAP`	[number] 기본값: 0	
Number of pyramid levels to build	`LEVELS`	[number] 기본값: 1	최소값: 0
Output directory	`OUTPUT`	[folder] 기본값: `[Save to temporary folder]`	타일을 저장할 산출물 폴더를 지정합니다. 다음 가운데 하나를 선택할 수 있습니다: • 임시 디렉터리에 저장 • 디렉터리에 저장
CSV file containing the tile(s) georeferencing information	`OUTPUT_CSV`	[file] 기본값: `[Skip output]`	타일을 저장할 산출 파일을 지정합니다. 다음 가운데 하나를 선택할 수 있습니다: • 산출물 건너뛰기 • 임시 파일로 저장 • 파일로 저장

고급 파라미터

라벨	명칭	유형	설명
Source coordinate reference system 부가적	`SOURCE_CRS`	[crs] 기본값: None	

다음 페이지에 계속

표 24.182 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Resampling method	RESAMPLING	[enumeration] 기본값: 0	사용할 리샘플링 알고리즘입니다. 다음 가운데 하나를 선택할 수 있습니다: • 0 —최근접 이웃 (nearest) • 1 —이중선형 (bilinear) • 2 —3 차 회선 (Cubic Convolution) (cubic) • 3 —B-스플라인 회선 (cubicspline) • 4 —란초시 창이 있는 싱크 (Lánczos Windowed Sinc) (lanczos)
Column delimiter used in the CSV file 부가적	DELIMITER	[string] 기본값: ';'	타일 (들) 의 지리참조 정보를 담고 있는 CSV 파일에 사용할 구분자
Additional creation options 부가적	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) 프로세스의 경우: 파이프 문자 (|) 로 복수의 옵션들을 분리하십시오.
Additional command-line parameters 부가적	EXTRA	[string] 기본값: ''	부가 GDAL 명령 줄 옵션을 추가합니다.
Output data type	DATA_TYPE	[enumeration] 기본값: 5	산출 래스터 파일의 포맷을 정의합니다. 다음 가운데 하나로 지정할 수 있습니다: • 0 —바이트 (Byte) • 1 —부호 있는 16 비트 정수형 (Int16) • 2 —부호 없는 16 비트 정수형 (UInt16) • 3 —부호 없는 32 비트 정수형 (UInt32) • 4 —부호 있는 32 비트 정수형 (Int32) • 5 —32 비트 부동소수점 실수형 (Float32) • 6 —부동소수점 64 비트 실수형 (Float64) • 7 —복잡 16 비트 정수형 (CInt16) • 8 —복잡 32 비트 정수형 (CInt32) • 9 —복잡 부동소수점 32 비트 실수형 (CFloat32) • 10 —복잡 부동소수점 64 비트 실수형 (CFloat64)
Build only the pyramids	ONLY_PYRAMIDS	[boolean] 기본값: False	
Use separate directory for each tile row	DIR_FOR_ROW	[boolean] 기본값: False	

산출물

라벨	명칭	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 **명칭** 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.

이 알고리즘은 GDAL 타일 인덱스 유틸리티 에서 파생되었습니다.

기본 메뉴: *Raster → Miscellaneous*

파라미터

기본 파라미터

라벨	명칭	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 명칭입니다.
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 24.183 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Tile index	OUTPUT	[vector: polygon] 기본값: [Save to temporary file]	인덱스를 작성할 폴리곤 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장

고급 파라미터

라벨	명칭	유형	설명
Transform geometries to the given CRS 부가적	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 부가적	CRS_FIELD_NAME	[string]	각 파일의 좌표계를 저장할 필드의 명칭입니다.
The format in which the CRS of each tile must be written	CRS_FORMAT	[enumeration] 기본값: 0	좌표계의 서식을 지정합니다. 다음 가운데 하나를 선택할 수 있습니다: • 0 –자동 (AUTO) • 1 –Well-known text (WKT) • 2 –EPSG (EPSG) • 3 –Proj.4 (PROJ)

산출물

라벨	명칭	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 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

가시권 분석

입력 DEM 래스터에 있는 사용자 정의 포인트에 대해 Wang2000 에 정의된 메소드를 이용해서 가시권 (viewshed) 래스터를 계산합니다.

파라미터

기본 파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[raster]	입력 표고 래스터 레이어
Band number	`BAND`	[raster band] 기본값: 1	표고로 사용할 밴드의 번호
Observer location	`OBSERVER`	[point]	관찰자의 위치
Observer height	`OBSERVER_HEIGHT`	[number] 기본값: 1.0	관찰자의 DEM 단위로 된 고도
Target height	`TARGET_HEIGHT`	[number] 기본값: 1.0	대상 (target) 요소의 DEM 단위로 된 고도
Maximum distance from observer to compute visibility	`MAX_DISTANCE`	[number] 기본값: 100.0	관찰자로부터 가시권을 계산할 DEM 단위로 된 최장 거리
산출물	`OUTPUT`	[raster] 기본값: `[Save to temporary file]`	산출 래스터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장

고급 파라미터

라벨	명칭	유형	설명
Additional creation options 부가적	`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) 프로세스의 경우: 파이프 문자 (|) 로 복수의 옵션들을 분리하십시오.
Additional command-line parameters	`EXTRA`	[string] 기본값: None	부가 GDAL 명령 줄 옵션을 추가합니다.

산출물

라벨	명칭	유형	설명
산출물	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 **명칭** 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.

이 알고리즘은 GDAL 편집 유틸리티 에서 파생되었습니다.

기본 메뉴: *Raster → Projections*

파라미터

라벨	명칭	유형	설명
Input layer	INPUT_LAYER	[raster]	입력 래스터 레이어
Desired CRS	CRS	[crs]	산출 레이어의 투영체 (좌표계)

산출물

라벨	명칭	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 **명칭** 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.

이 알고리즘은 GDAL 좌표계 정보 유틸리티 에서 파생되었습니다.

기본 메뉴: *Raster → Projections*

파라미터

라벨	명칭	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] 기본값: False	이 파라미터를 활성화하면 투영체 정보를 담은 `.prj` 파일도 생성합니다.

산출물

라벨	명칭	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 **명칭** 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.

이 알고리즘은 GDAL 왜곡 (warp) 유틸리티 에서 파생되었습니다.

기본 메뉴: *Raster → Projections*

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[raster]	재투영할 입력 래스터 레이어
Source CRS 부가적	`SOURCE_CRS`	[crs]	입력 래스터 레이어의 좌표계를 정의합니다.
Target CRS 부가적	`TARGET_CRS`	[crs] 기본값: `EPSG:4326`	산출 레이어의 좌표계
Resampling method to use	`RESAMPLING`	[enumeration] 기본값: 0	사용할 픽셀값 리샘플링 메소드를 선택합니다. 선택할 수 있는 옵션은 다음과 같습니다: • 0 — 최근접 이웃 • 1 — 이중선형 (Bilinear) • 2 — 3 차 회선 (Cubic Convolution) • 3 — B-스플라인 회선 • 4 — 란초시 창이 있는 싱크 (Lánczos Windowed Sinc) • 5 — 평균 • 6 — 모드 • 7 — 최대값 • 8 — 최소값 • 9 — 중간값 • 10 — 제 1 사분위수 (First Quartile) • 11 — 제 3 사분위수 (Third Quartile)
Nodata value for output bands 부가적	`NODATA`	[number] 기본값: None	산출 밴드에 사용할 NODATA 값을 설정합니다. 설정하지 않을 경우, 소스 데이터셋에서 NODATA 값을 복사할 것입니다.

다음 페이지에 계속

표 24.187 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Output file resolution in target georeferenced units 부가적	TARGET_RESOLUTION	[number] 기본값: None	재투영한 결과를 저장할 산출 파일의 해상도를 정의합니다.
Additional creation options 부가적	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) 프로세스의 경우: 파이프 문자 (|) 로 복수의 옵션들을 분리하십시오.
Output data type	DATA_TYPE	[enumeration] 기본값: 0	산출 래스터 파일의 포맷을 정의합니다. 옵션: • 0 —입력 레이어의 데이터 유형 사용 • 1 —바이트 (Byte) • 2 —부호 있는 16 비트 정수형 (Int16) • 3 —부호 없는 16 비트 정수형 (UInt16) • 4 —부호 없는 32 비트 정수형 (UInt32) • 5 —부호 있는 32 비트 정수형 (Int32) • 6 —부동소수점 32 비트 실수형 (Float32) • 7 —부동소수점 64 비트 실수형 (Float64) • 8 —복잡 16 비트 정수형 (CInt16) • 9 —복잡 32 비트 정수형 (CInt32) • 10 —복잡 부동소수점 32 비트 실수형 (CFloat32) • 11 —복잡 부동소수점 64 비트 실수형 (CFloat64)
Georeferenced extents of output file to be created 부가적	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.)
CRS of the target raster extent 부가적	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] 기본값: False	이미지 무리를 처리하고 입력/출력 작업을 동시에 수행하기 위해 스레드 2 개를 사용할 것입니다. 계산 자체는 다중 스레드 작업이 아니라는 사실을 기억하십시오.

다음 페이지에 계속

표 24.187 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Additional command-line parameters 부가적	EXTRA	[string] 기본값: None	부가 GDAL 명령 줄 옵션을 추가합니다.
Reprojected	OUTPUT	[raster] 기본값: '[Save to temporary file]'	산출 래스터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Reprojected	OUTPUT	[raster] 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 **명칭** 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.

이 알고리즘은 ogr2ogr 유틸리티 에서 파생되었습니다.

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	입력 벡터 레이어
Additional creation options 부가적	`OPTIONS`	[string] 기본값: ''(추가 옵션 없음)	부가적인 GDAL 생성 옵션
Converted	`OUTPUT`	[same as input]	산출 벡터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다. `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.

산출물

라벨	명칭	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 **명칭** 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.

이 알고리즘은 GDAL 래스터화 유틸리티 에서 파생되었습니다.

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	입력 벡터 레이어
Input raster layer	`INPUT_RASTER`	[raster]	입력 래스터 레이어
Field to use for a burn-in value 부가적	`FIELD`	[tablefield: numeric]	픽셀 값을 설정하기 위해 사용할 속성 필드를 정의합니다.
Add burn in values to existing raster values	`ADD`	[boolean] 기본값: False	거짓으로 설정하면, 픽셀에 선택한 필드의 값을 할당합니다. 참으로 설정하면, 입력 래스터 레이어의 값에 선택한 필드의 값을 추가합니다.
Additional command-line parameters 부가적	`EXTRA`	[string] 기본값: ''	부가 GDAL 명령 줄 옵션을 추가합니다.

산출물

라벨	명칭	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 **명칭** 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.

이 알고리즘은 GDAL 래스터화 유틸리티 에서 파생되었습니다.

파라미터

라벨	명칭	유형	설명
Input layer	INPUT	[vector: any]	입력 벡터 레이어
Input raster layer	INPUT_RASTER	[raster]	입력 래스터 레이어
A fixed value to burn	BURN	[number] 기본값: 0.0	덮어쓸 값
Add burn in values to existing raster values	ADD	[boolean] 기본값: False	거짓으로 설정하면, 픽셀에 고정값을 할당합니다. 참으로 설정하면, 입력 래스터 레이어의 값에 고정값을 추가합니다.
Additional command-line parameters 부가적	EXTRA	[string] 기본값: ''	부가 GDAL 명령 줄 옵션을 추가합니다.

산출물

라벨	명칭	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 **명칭** 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.

이 알고리즘은 GDAL 래스터화 유틸리티 에서 파생되었습니다.

기본 메뉴: *Raster → Conversion*

파라미터

라벨	명칭	유형	설명
Input layer	INPUT	[vector: any]	입력 벡터 레이어
Field to use for a burn-in value 부가적	FIELD	[tablefield: numeric]	픽셀용 속성을 선택해야 할 속성 필드를 정의합니다.

다음 페이지에 계속

표 24.190 – 이전 페이지에서 계속

라벨	명칭	유형	설명
A fixed value to burn 부가적	BURN	[number] 기본값: 0.0	모든 피처의 밴드에 덮어쓸 고정값
Output raster size units	UNITS	[enumeration] 기본값: 0	산출 래스터의 크기/해상도를 정의할 때 사용할 단위입니다. 다음 가운데 하나를 선택할 수 있습니다: • 0 —픽셀 • 1 —지리참조된 단위
Width/Horizontal resolution	WIDTH	[number] 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	[number] 기본값: 0.0	산출 래스터의 (크기 단위가"픽셀"인 경우) 높이를, 또는 (크기 단위가"지리참조된 단위"인 경우) 수직 해상도를 설정합니다.
Output extent	EXTENT	[extent]	산출 래스터 레이어의 범위입니다. 범위를 지정하지 않은 경우, 선택한 참조 레이어 (들) 을 커버하는 최소 범위를 사용할 것입니다.
Assign a specified nodata value to output bands 부가적	NODATA	[number] 기본값: 0.0	산출 밴드에 지정한 NODATA 값을 할당합니다.
Additional creation options 부가적	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) 프로세스의 경우: 파이프 문자 (|) 로 복수의 옵션들을 분리하십시오.
Output data type	DATA_TYPE	[enumeration] 기본값: 5	산출 래스터 파일의 포맷을 정의합니다. 옵션: • 0 —바이트 (Byte) • 1 —부호 있는 16 비트 정수형 (Int16) • 2 —부호 없는 16 비트 정수형 (UInt16) • 3 —부호 없는 32 비트 정수형 (UInt32) • 4 —부호 있는 32 비트 정수형 (Int32) • 5 —32 비트 부동소수점 실수형 (Float32) • 6 —부동소수점 64 비트 실수형 (Float64) • 7 —복잡 16 비트 정수형 (CInt16) • 8 —복잡 32 비트 정수형 (CInt32) • 9 —복잡 부동소수점 32 비트 실수형 (CFloat32) • 10 —복잡 부동소수점 64 비트 실수형 (CFloat64)

다음 페이지에 계속

표 24.190 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Pre-initialize the output image with value 부가적	INIT	[number]	산출 이미지의 밴드를 이 값으로 사전 초기화합니다. 산출물 파일에서 NODATA 값으로 마크하지 않습니다. 모든 밴드에 동일한 값을 사용합니다.
Invert rasterization	INVERT	[boolean] 기본값: False	고정값을 덮어쓰거나, 또는 이미지에서 지정한 폴리곤 안에 들어가지 않는 모든 부분에 첫 번째 피처 관련 값을 덮어씁니다.
Rasterized	OUTPUT	[raster] 기본값: 'Save to temporary file'	산출 래스터 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다. 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.

산출물

라벨	명칭	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 **명칭** 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

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	입력 벡터 레이어
Geometry column name	`GEOMETRY`	[string] 기본값: 'geometry'	사용할 입력 레이어 도형 열의 명칭
Buffer distance	`DISTANCE`	[number] 기본값: 10.0	최소값: 0.0
Dissolve by attribute 부가적	`FIELD`	[tablefield: any] 기본값: None	융해 작업에 사용할 필드
Dissolve results	`DISSOLVE`	[boolean] 기본값: False	참으로 설정하면 결과물을 융해합니다. 융해 작업용 필드를 설정하지 않으면, 모든 버퍼를 단일 피처로 융해합니다.
Produce one feature for each geometry in any kind of geometry collection in the source file	`EXPLODE_COLLECTIONS`	[boolean] 기본값: False	
Additional creation options 부가적	`OPTIONS`	[string] 기본값: ''(추가 옵션 없음)	부가적인 GDAL 생성 옵션
Buffer	`OUTPUT`	[vector: polygon] 기본값: `[Save to temporary file]`	산출 버퍼 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

이 알고리즘은 GDAL ogr2ogr 유틸리티 에서 파생되었습니다.

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	입력 벡터 레이어
Clip extent	`EXTENT`	[extent]	산출 벡터 파일에 사용되어야 할 경계 상자를 정의합니다. 대상 좌표계의 좌표로 이 경계 상자를 정의해야 합니다.
Additional creation options 부가적	`OPTIONS`	[string] 기본값: ''(추가 옵션 없음)	부가적인 GDAL 생성 옵션
Clipped (extent)	`OUTPUT`	[same as input] 기본값: `[Save to temporary file]`	산출 (잘라낸) 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Clipped (extent)	`OUTPUT`	[same as input]	산출 (잘라낸) 레이어입니다. 기본 포맷은 "ESRI Shapefile"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 **명칭** 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.

이 알고리즘은 GDAL ogr2ogr 유틸리티 에서 파생되었습니다.

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	입력 벡터 레이어
Mask layer	`MASK`	[vector: polygon]	입력 벡터 레이어에 잘라내기 범위로 사용할 레이어
Additional creation options 부가적	`OPTIONS`	[string] 기본값: ‘’(추가 옵션 없음)	부가적인 GDAL 생성 옵션
Clipped (mask)	`OUTPUT`	[same as input] 기본값: `[Save to temporary file]`	산출 (마스크로 잘라낸) 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Clipped (mask)	`OUTPUT`	[same as input]	산출 (마스크로 잘라낸) 레이어입니다. 기본 포맷은“ESRI Shapefile”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 **명칭** 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

라벨	명칭	유형	설명
Input layer	INPUT	[vector: any]	융해시킬 입력 레이어
Dissolve field 부가적	FIELD	[tablefield: any]	융해 작업에 사용할 입력 레이어의 필드
Geometry column name	GEOMETRY	[string] 기본값: 'geometry'	융해 작업에 사용할 입력 레이어 도형 열의 명칭
Produce one feature for each geometry in any kind of geometry collection in the source file	EXPLODE_COLLECTIONS	[boolean] 기본값: False	소스 파일에 있는 모든 유형의 도형 집합의 각 도형에 대해 피처 1 개를 생성합니다.
Keep input attributes	KEEP_ATTRIBUTES	[boolean] 기본값: False	입력 레이어의 모든 속성을 유지합니다.
Count dissolved features	COUNT_FEATURES	[boolean] 기본값: False	산출 레이어에 융해된 피처의 개수를 포함시킵니다.
Compute area and perimeter of dissolved features	COMPUTE_AREA	[boolean] 기본값: False	융해된 피처의 면적 및 둘레를 계산해서 산출 레이어에 포함시킵니다.
Compute min/max/sum/mean for attribute	COMPUTE_STATISTICS	[boolean] 기본값: False	지정한 숫자 속성에 대한 통계 (최소값, 최대값, 합계 및 평균) 를 계산해서 산출 레이어에 포함시킵니다.
Numeric attribute to calculate statistics on 부가적	STATISTICS_ATTRIBUTE	[tablefield: numeric]	통계를 계산할 숫자 속성
Additional creation options 부가적	OPTIONS	[string] 기본값: ''(추가 옵션 없음)	부가적인 GDAL 생성 옵션
Dissolved	OUTPUT	[same as input] 기본값: [Save to temporary file]	산출 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Dissolved	`OUTPUT`	[same as input]	융해된 도형을 담은 산출 다중 부분 도형 레이어

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 **명칭** 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

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: line]	입력 라인 레이어
Geometry column name	`GEOMETRY`	[string] 기본값: 'geometry'	사용할 입력 레이어 도형 열의 명칭
Offset distance (left-sided: positive, right-sided: negative)	`DISTANCE`	[number] 기본값: 10.0	
Additional creation options 부가적	`OPTIONS`	[string] 기본값: ''(추가 옵션 없음)	부가적인 GDAL 생성 옵션
Offset curve	`OUTPUT`	[vector: line] 기본값: `[Save to temporary file]`	산출 라인 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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

라벨	명칭	유형	설명
Input layer	INPUT	[vector: line]	입력 라인 레이어
Geometry column name	GEOMETRY	[string] 기본값: 'geometry'	사용할 입력 레이어 도형 열의 명칭
Buffer distance	DISTANCE	[number] 기본값: 10.0	
Buffer side	BUFFER_SIDE	[enumeration] 기본값: 0	다음 가운데 하나로 지정할 수 있습니다: • 0 — 오른쪽 • 1 — 왼쪽
Dissolve by attribute 부가적	FIELD	[tablefield: any] 기본값: None	융해 작업에 사용할 필드
Dissolve all results	DISSOLVE	[boolean] 기본값: False	참으로 설정하면 결과물을 융해합니다. 융해 작업용 필드를 설정하지 않으면, 모든 버퍼를 단일 피처로 융해합니다.
Produce one feature for each geometry in any kind of geometry collection in the source file	EXPLODE_COLLECTIONS	[boolean] 기본값: False	
Additional creation options 부가적	OPTIONS	[string] 기본값: ''(추가 옵션 없음)	부가적인 GDAL 생성 옵션

다음 페이지에 계속

표 24.194 – 이전 페이지에서 계속

라벨	명칭	유형	설명
One-sided buffer	OUTPUT	[vector: polygon] 기본값: [Save to temporary file]	산출 버퍼 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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

라벨	명칭	유형	설명
Input layer	INPUT	[vector: line]	입력 라인 레이어
Geometry column name	GEOMETRY	[string] 기본값: 'geometry'	사용할 입력 레이어 도형 열의 명칭
Distance from line start represented as a fraction of line length	DISTANCE	[number] 기본값: 0.5 (라인의 중간)	
Additional creation options 부가적	OPTIONS	[string] 기본값: ''(추가 옵션 없음)	부가적인 GDAL 생성 옵션

다음 페이지에 계속

표 24.195 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Points along line	OUTPUT	[vector: point] 기본값: [Save to temporary file]	산출 포인트 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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

벡터 레이어 집합을 담고 있는 가상 벡터 레이어를 생성합니다. 산출 가상 벡터 레이어는 현재 프로젝트에서 열리지 않을 것입니다.

이 알고리즘은 또다른 알고리즘이 여러 레이어가 필요하지만 레이어를 지정하는 vrt 파라미터가 단 하나의 인자만 받는 경우에 특히 유용합니다.

파라미터

라벨	명칭	유형	설명
Input datasources	INPUT	[vector: any] [list]	가상 벡터를 작성하는 데 사용하려는 벡터 레이어들을 선택합니다.
Create "unioned" VRT	UNIONED	[boolean] 기본값: False	모든 벡터들을 단일 vrt 파일로 통합하고자 할 경우 활성화하십시오.
Virtual vector	OUTPUT	[same as input] 기본값: [Save to temporary file]	복사본만 담고 있는 산출 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	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 **명칭** 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.

이 알고리즘은 GDAL ogr2ogr 유틸리티 에서 파생되었습니다.

파라미터

라벨	명칭	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`
SQL dialect	DIALECT	[enumeration] 기본값: 0	사용할 SQL 방언 (dialect) 을 지정합니다. 다음 가운데 하나를 선택할 수 있습니다: • 0 — 없음 • 1 — OGR SQL • 2 — SQLite
Additional creation options 부가적	OPTIONS	[string] 기본값: ''(추가 옵션 없음)	부가적인 GDAL 생성 옵션
SQL result	OUTPUT	[vector: any]	산출 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다. `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.

산출물

라벨	명칭	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 **명칭** 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. |checkbox| *Save Username* 과 *Save Password* 가 체크되어 있는지 확인하십시오. 그런 다음에야 이 알고리즘을 사용할 수 있습니다.

이 알고리즘은 GDAL ogr2ogr 유틸리티 에서 파생되었습니다.

파라미터

라벨	명칭	유형	설명
Database (connection name)	DATABASE	[string]	연결할 PostgreSQL 데이터베이스
Input layer	INPUT	[vector: any]	데이터베이스로 내보낼 OGR 이 지원하는 벡터 레이어
Shape encoding 부가적	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: • 0 — 자동 • 1 — 없음 • 2 — 도형 • 3 — 포인트 • 4 — 라인스트링 • 5 — 폴리곤 • 6 — 도형집합 • 7 — 멀티포인트 • 8 — 멀티폴리곤 • 9 — 멀티라인스트링
Assign an output CRS 부가적	A_SRS	[crs] 기본값: None	데이터베이스 테이블의 산출 좌표계를 정의합니다.

다음 페이지에 계속

표 24.197 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Reproject to this CRS on output 부가적	T_SRS	[crs] 기본값: None	산출물에 이 좌표계로 재투영/변형합니다.
Override source CRS 부가적	S_SRS	[crs] 기본값: None	입력 레이어의 좌표계를 무시합니다.
Schema (schema name) 부가적	SCHEMA	[string] 기본값: 'public'	데이터베이스 테이블의 스키마를 정의합니다.
Table to export to (leave blank to use layer name) 부가적	TABLE	[string] 기본값: ''	데이터베이스로 가져올 테이블의 명칭을 정의합니다. 테이블명은 기본적으로 입력 벡터 파일의 명칭입니다.
Primary Key (new field) 부가적	PK	[string] 기본값: 'id'	데이터베이스 테이블의 기본 키가 될 속성 필드를 정의합니다.
Primary Key (existing field, used if the above option is left empty) 부가적	PRIMARY_KEY	[tablefield: any] 기본값: None	데이터베이스 테이블의 기본 키가 될 내보낸 레이어의 속성 필드를 정의합니다.
Geometry column name 부가적	GEOCOLUMN	[string] 기본값: 'geom'	도형 정보를 담게 될 데이터베이스의 속성 필드를 정의합니다.
Vector dimensions 부가적	DIM	[enumeration] 기본값: 0 (2 차원)	가져올 벡터 파일이 2 차원인지 3 차원인지 정의합니다. 다음 가운데 하나로 지정할 수 있습니다: • 0 —2 차원 • 1 —3 차원
Distance tolerance for simplification 부가적	SIMPLIFY	[string] 기본값: ''	가져올 벡터 도형을 단순화하기 위한 거리 허용 오차를 정의합니다. 기본값은 단순화하지 않는 것입니다.
Maximum distance between 2 nodes (densification) 부가적	SEGMENTIZE	[string] 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) 부가적	SPAT	[extent] 기본값: None	산출 테이블에 들어갈 지정 범위에서 피처를 선택할 수 있습니다.
Clip the input layer using the above (rectangle) extent 부가적	CLIP	[boolean] 기본값: False	입력 레이어를 사용자가 사전에 정의한 범위로 잘라낼 것입니다.

다음 페이지에 계속

표 24.197 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Select features using a SQL "WHERE" statement (Ex: column="value") 부가적	WHERE	[string] 기본값: ''	입력 레이어에서 어떤 피처를 선택해야 할지를 SQL "WHERE" 선언문으로 정의합니다.
Group N features per transaction (Default: 2000) 부가적	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 부가적	OVERWRITE	[boolean] 기본값: True	데이터베이스에 동일한 명칭을 가진 테이블이 있는 경우, 그리고 이 옵션을 참으로 설정한 경우, 테이블을 덮어쓸 것입니다.
Append to existing table 부가적	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 부가적	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 부가적	LAUNDER	[boolean] 기본값: False	이 옵션을 참으로 설정하면 (열의 명칭을 소문자로 변환한다거나, 공백 및 기타 유효하지 않은 문자를 제거하는 등의) 기본 습성을 막을 수 있습니다.
Do not create Spatial Index 부가적	INDEX	[boolean] 기본값: False	산출 테이블에 공간 인덱스를 생성하는 일을 막습니다. 기본값은 공간 인덱스를 추가하는 것입니다.
Continue after a failure, skipping the failed feature 부가적	SKIPFAILURES	[boolean] 기본값: False	
Promote to Multipart 부가적	PROMOTETOMULTI	[boolean] 기본값: True	산출 테이블에서 피처의 도형 유형을 다중 부분으로 변환합니다.
Keep width and precision of input attributes 부가적	PRECISION	[boolean] 기본값: True	열 속성을 입력 데이터를 따르도록 수정하는 일을 막습니다.
Additional creation options 부가적	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 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

PostgreSQL 로 내보내기 (새 연결)

PostGreSQL 데이터베이스로 벡터 레이어를 가져옵니다. PostGIS 데이터베이스로의 새 연결을 생성해야만 합니다.

이 알고리즘은 GDAL ogr2ogr 유틸리티 에서 파생되었습니다.

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	데이터베이스로 내보낼 OGR 이 지원하는 벡터 레이어
Shape encoding 부가적	`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: • 0 — 자동 • 1 — 없음 • 2 — 도형 • 3 — 포인트 • 4 — 라인스트링 • 5 — 폴리곤 • 6 — 도형집합 • 7 — 멀티포인트 • 8 — 멀티폴리곤 • 9 — 멀티라인스트링
Assign an output CRS 부가적	`A_SRS`	[crs] 기본값: None	데이터베이스 테이블의 산출 좌표계를 정의합니다.
Reproject to this CRS on output 부가적	`T_SRS`	[crs] 기본값: None	산출물에 이 좌표계로 재투영/변형합니다.
Override source CRS 부가적	`S_SRS`	[crs] 기본값: None	입력 레이어의 좌표계를 무시합니다.

다음 페이지에 계속

표 24.198 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Host 부가적	HOST	[string] 기본값: 'localhost'	데이터베이스의 호스트명
Port 부가적	PORT	[string] 기본값: '5432'	PostgreSQL 데이터베이스 서버가 열어놓은 포트 번호
Username 부가적	USER	[string] 기본값: ''	데이터베이스에 로그인하는 데 쓰이는 사용자명
Database name 부가적	DBNAME	[string] 기본값: ''	데이터베이스의 명칭
Password 부가적	PASSWORD	[string] 기본값: ''	데이터베이스에 연결하기 위해 사용자명과 함께 쓰이는 비밀번호
Schema (schema name) 부가적	SCHEMA	[string] 기본값: 'public'	데이터베이스 테이블의 스키마를 정의합니다.
Table name, leave blank to use input name 부가적	TABLE	[string] 기본값: ''	데이터베이스로 가져올 테이블의 명칭을 정의합니다. 테이블명은 기본적으로 입력 벡터 파일의 명칭입니다.
Primary Key (new field) 부가적	PK	[string] 기본값: 'id'	데이터베이스 테이블의 기본 키가 될 속성 필드를 정의합니다.
Primary Key (existing field, used if the above option is left empty) 부가적	PRIMARY_KEY	[tablefield: any] 기본값: None	데이터베이스 테이블의 기본 키가 될 내보낸 레이어의 속성 필드를 정의합니다.
Geometry column name 부가적	GEOCOLUMN	[string] 기본값: 'geom'	도형 정보를 담게 될 데이터베이스의 속성 필드를 정의합니다.
Vector dimensions 부가적	DIM	[enumeration] 기본값: 0 (2 차원)	가져올 벡터 파일이 2 차원인지 3 차원인지 정의합니다. 다음 가운데 하나로 지정할 수 있습니다: • 0 — 2 차원 • 1 — 3 차원
Distance tolerance for simplification 부가적	SIMPLIFY	[string] 기본값: ''	가져올 벡터 도형을 단순화하기 위한 거리 허용 오차를 정의합니다. 기본값은 단순화하지 않는 것입니다.
Maximum distance between 2 nodes (densification) 부가적	SEGMENTIZE	[string] 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) 부가적	SPAT	[extent] 기본값: None	산출 테이블에 들어갈 지정 범위에서 피처를 선택할 수 있습니다.
Clip the input layer using the above (rectangle) extent 부가적	CLIP	[boolean] 기본값: False	입력 레이어를 사용자가 사전에 정의한 범위로 잘라낼 것입니다.

다음 페이지에 계속

표 24.198 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Fields to include (leave empty to use all fields) 부가적	FIELDS	[string] [list] 기본값: []	가져온 벡터 파일에서 유지할 필드를 정의합니다. 아무것도 선택하지 않으면 모든 필드를 가져옵니다.
Select features using a SQL "WHERE" statement (Ex: column="value") 부가적	WHERE	[string] 기본값: ''	산출 레이어를 위해 어떤 피처를 선택해야 할지를 SQL "WHERE" 선언문으로 정의합니다.
Group N features per transaction (Default: 2000) 부가적	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 부가적	OVERWRITE	[boolean] 기본값: True	데이터베이스에 동일한 명칭을 가진 테이블이 있는 경우, 그리고 이 옵션을 참으로 설정한 경우, 테이블을 덮어쓸 것입니다.
Append to existing table 부가적	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 부가적	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 부가적	LAUNDER	[boolean] 기본값: False	이 옵션을 참으로 설정하면 (열의 명칭을 소문자로 변환한다거나, 공백 및 기타 유효하지 않은 문자를 제거하는 등의) 기본 습성을 막을 수 있습니다.
Do not create Spatial Index 부가적	INDEX	[boolean] 기본값: False	산출 테이블에 공간 인덱스를 생성하는 일을 막습니다. 기본값은 공간 인덱스를 추가하는 것입니다.
Continue after a failure, skipping the failed feature 부가적	SKIPFAILURES	[boolean] 기본값: False	
Promote to Multipart 부가적	PROMOTETOMULTI	[boolean] 기본값: True	산출 테이블에서 피처의 도형 유형을 다중 부분으로 변환합니다.
Keep width and precision of input attributes 부가적	PRECISION	[boolean] 기본값: True	열 속성을 입력 데이터를 따르도록 수정하는 일을 막습니다.
Additional creation options 부가적	OPTIONS	[string] 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 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

벡터 정보

OGR 이 지원하는 데이터소스에 대한 정보 목록을 담은 정보 파일을 생성합니다. *Processing* –> *Results viewer* 창에 산출물을 표시하며, HTML 파일로 작성할 수도 있습니다. 이 정보 파일은 도형 유형, 피처 개수, 공간 범위, 투영체 정보 등등 수많은 정보를 담고 있습니다.

이 알고리즘은 GDAL ORG 정보 유틸리티 에서 파생되었습니다.

파라미터

라벨	명칭	유형	설명
Input layer	`INPUT`	[vector: any]	입력 벡터 레이어
Summary output only 부가적	`SUMMARY_ONLY`	[boolean] 기본값: True	
Suppress metadata info 부가적	`NO_METADATA`	[boolean] 기본값: False	
Layer information	`OUTPUT`	[html] 기본값: `[Save to temporary file]`	파일 정보를 담은 산출 HTML 파일을 지정합니다. 다음 가운데 하나를 선택할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다. 아무 HTML 파일도 정의하지 않으면 산출물을 임시 파일로 작성할 것입니다.

산출물

라벨	명칭	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 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

24.3 LAStools 알고리즘 제공자

LAStools 는 LiDAR 데이터 공간 처리를 위한 매우 효율적인 멀티코어 명령 줄 도구의 집합입니다.

24.3.1 blast2dem

설명

스트리밍을 사용해서 구현된 단절없는 (seamless) 들로네 삼각분할을 통해 (수십 억에 달하는) 포인트를 대용량 표고 래스터, 강도 (intensity) 래스터, 또는 RGB 래스터로 변환합니다.

더 자세한 정보는 blast2dem 웹페이지와 온라인 README 파일을 참조하세요.

파라미터

라벨	명칭	유형	설명
verbose	VERBOSE	[boolean] 기본값: False	콘솔에 좀 더 서술적인 제어 산출물을 생성합니다.
open LAStools GUI	GUI	[boolean] 기본값: False	사전에 지정한 입력 파일을 가지고 있는 LAStools 의 GUI 를 실행합니다.
input LAS/LAZ file	INPUT_LASLAZ	[file]	래스터화될 포인트를 담고 있는 LAS/LAZ 포맷 파일입니다.

다음 페이지에 계속

표 24.200 – 이전 페이지에서 계속

라벨	명칭	유형	설명
filter (by return, classification, flag)	FILTER_RETURN_CLASS_FLAGS	[enumeration] 기본값: 0	어떤 포인트가 차후 래스터화될 임시 TIN을 구성하는 데 사용될지 지정합니다. 다음 가운데 하나를 선택할 수 있습니다: • 0 —— • 1 −keep_last • 2 −keep_first • 3 −keep_middle • 4 −keep_single • 5 −drop_single • 6 −keep_double • 7 −keep_class 2 • 8 −keep_class 2 8 • 9 −keep_class 8 • 10 −keep_class 6 • 11 −keep_class 9 • 12 −keep_class 3 4 5 • 13 −keep_class 2 6 • 14 −drop_class 7 • 15 −drop_withheld • 16 —drop_synthetic • 17 —drop_overlap • 18 —keep_withheld • 19 —keep_synthetic • 20 —keep_keypoint • 21 —keep_overlap
step size / pixel size	STEP	[number] 기본값: 1.0	TIN 이 래스터화되는 그리드의 셀 크기를 지정합니다.
Attribute	ATTRIBUTE	[enumeration] 기본값: 0	래스터화될 속성을 지정합니다. 다음 가운데 하나를 선택할 수 있습니다: • 0 —표고 • 1 —경사도 • 2 —강도 (intensity) • 3 —RGB
Product	PRODUCT	[enumeration] 기본값: 0	속성을 래스터 값으로 어떻게 변환할지 지정합니다. 다음 가운데 하나를 선택할 수 있습니다: • 0 —실제 값 • 1 —음영기복 • 2 —회색조 • 3 —적외선 (false color)
Use tile bounding box (after tiling with buffer)	USE_TILE_BB	[boolean] 기본값: False	래스터화 영역을 타일 경계 상자로 제한합니다. (입력물이 lastile 로 생성한 LAS/LAZ 타일인 경우에만 의미가 있습니다.)
additional command line parameter(s) 부가적	ADDITIONAL_OPTIONS	[string] 기본값: ‘’	이 메뉴에서는 사용할 수 없지만 (고급) LAStools 사용자는 알고 있는 다른 명령 줄 스위치를 지정합니다.

다음 페이지에 계속

표 24.200 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Output raster file	`OUTPUT_RASTER`	[raster] 기본값: `[Skip output]`	산출 래스터를 어디에 저장할지 지정합니다. 적외선 영상 (false color), 회색조 색상표, 음영기복의 경우 TIF, PNG 및 JPG 같은 이미지 래스터를 사용합니다. 실제 값의 경우 TIF, BIL, IMG, ASC, DTM, FLT, XYZ 및 CSV 같은 값 래스터를 사용합니다. 다음 가운데 하나로 저장할 수 있습니다: • 산출물 건너뛰기 • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Output raster file	`OUTPUT_RASTER`	[raster]	산출 래스터

파이썬 코드

알고리즘 ID: `lastools:blast2dem`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

24.3.2 blast2iso

설명

스트리밍을 사용해서 구현된 단절없는 (seamless) 들로네 삼각분할을 통해 (수십 억에 달하는) 포인트를 동위 등고선 (iso-contour) 라인으로 변환합니다.

더 자세한 정보는 blast2iso 웹페이지와 온라인 README 파일을 참조하세요.

파라미터

라벨	명칭	유형	설명
verbose	VERBOSE	[boolean] 기본값: False	콘솔에 좀 더 서술적인 제어 산출물을 생성합니다.
open LAStools GUI	GUI	[boolean] 기본값: False	사전에 지정한 입력 파일을 가지고 있는 LAStools 의 GUI 를 실행합니다.
input LAS/LAZ file	INPUT_LASLAZ	[file]	등위선 라인을 생성하기 위해 사용할 포인트를 담고 있는 파일입니다.
smooth underlying TIN	SMOOTH	[number] 기본값: 0	임시 TIN 을 평탄화해야 하는지, 한다면 몇 번이나 반복해야 하는지를 지정합니다.
extract isoline with a spacing of	ISO_EVERY	[number] 기본값: 10.0	등위선 라인을 어떤 간격으로 추출할지를 지정합니다. (등고선 간격)
clean isolines shorter than (0 = do not clean)	CLEAN	[number] 기본값: 0.0	지정한 길이보다 짧은 등위선을 뺍니다.
simplify segments shorter than (0 = do not simplify)	SIMPLIFY_LENGTH	[number] 기본값: 0.0	지정한 길이보다 짧은 등위선을 가장 기본적인 형태로 단순화시킵니다.
simplify segment pairs with area less than (0 = do not simplify)	SIMPLIFY_AREA	[number] 기본값: 0.0	연속 선분으로 이루어진 요철 (bump) 부분 가운데 지정한 크기보다 작은 면적을 가진 요철을 가장 기본적인 형태로 단순화시킵니다.
additional command line parameter(s) 부가적	ADDITIONAL_OPTIONS	[string] 기본값: ' '	이 메뉴에서는 사용할 수 없지만 (고급) LAStools 사용자는 알고 있는 다른 명령 줄 스위치를 지정합니다.
Output vector file	OUTPUT_VECTOR	[vector: line] 기본값: [Skip output]	어디에 산출 벡터를 저장할지 지정합니다. SHP 또는 WKT 산출 파일을 사용합니다. 사용자의 입력 LiDAR 파일이 지리 좌표 (경도/위도) 인 경우 또는 지리참조 정보를 가지고 있는 경우에만, KML 산출 파일도 생성할 수 있습니다. 다음 가운데 하나로 저장할 수 있습니다: • 산출물 건너뛰기 • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Output vector file	OUTPUT_VECTOR	[vector: line]	등위선을 담은 산출 라인 벡터 레이어

파이썬 코드

알고리즘 ID: `lastools:blast2iso`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

24.3.3 las2dem

설명

사용자가 정의한 단계 크기로 래스터화된 임시 들로네 삼각분할을 통해 (2 억에 달하는) 포인트를 표고 래스터, 강도 (intensity) 래스터, 또는 RGB 래스터로 변환합니다.

더 자세한 정보는 las2dem 웹페이지와 온라인 README 파일을 참조하세요.

파라미터

라벨	명칭	유형	설명
verbose	`VERBOSE`	[boolean] 기본값: False	콘솔에 좀 더 서술적인 제어 산출물을 생성합니다.
run new 64 bit executable	`CPU64`	[boolean] 기본값: False	
open LAStools GUI	`GUI`	[boolean] 기본값: False	사전에 지정한 입력 파일을 가지고 있는 LAStools 의 GUI 를 실행합니다.
input LAS/LAZ file	`INPUT_LASLAZ`	[file]	래스터화될 포인트를 담고 있는 LAS/LAZ 포맷 파일입니다.

다음 페이지에 계속

표 24.202 – 이전 페이지에서 계속

라벨	명칭	유형	설명
filter (by return, classification, flags)	FILTER_RETURN_CLASS_FLAGS	[enumeration] 기본값: 0	어떤 포인트가 차후 래스터화될 임시 TIN을 구성하는 데 사용될지 지정합니다. 다음 가운데 하나를 선택할 수 있습니다: • 0 — • 1 —keep_last • 2 —keep_first • 3 —keep_middle • 4 —keep_single • 5 —drop_single • 6 —keep_double • 7 —keep_class 2 • 8 —keep_class 2 8 • 9 —keep_class 8 • 10 —keep_class 6 • 11 —keep_class 9 • 12 —keep_class 3 4 5 • 13 —keep_class 3 • 14 —keep_class 4 • 15 —keep_class 5 • 16 —keep_class 2 6 • 17 —drop_class 7 • 18 —drop_withheld • 19 —drop_synthetic • 20 —drop_overlap • 21 —keep_withheld • 22 —keep_synthetic • 23 —keep_keypoint • 24 —keep_overlap
step size / pixel size	STEP	[number] 기본값: 1.0	TIN 이 래스터화되는 그리드의 셀 크기를 지정합니다.
Attribute	ATTRIBUTE	[enumeration] 기본값: 0	래스터화할 속성을 지정합니다. 다음 가운데 하나를 선택할 수 있습니다: • 0 —표고 • 1 —경사도 • 2 —강도 (intensity) • 3 —RGB • 4 —edge_longest • 5 —edge_shortest
Product	PRODUCT	[enumeration] 기본값: 0	속성을 래스터 값으로 어떻게 변환할지 지정합니다. 다음 가운데 하나를 선택할 수 있습니다: • 0 —실제 값 • 1 —음영기복 • 2 —회색조 • 3 —적외선 (false color)

다음 페이지에 계속

표 24.202 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Use tile bounding box (after tiling with buffer)	USE_TILE_BB	[boolean] 기본값: False	래스터화 영역을 타일 경계 상자로 제한합니다. (입력물이 lastile 로 생성한 LAS/LAZ 타일인 경우에만 의미가 있습니다.)
additional command line parameter(s) 부가적	ADDITIONAL_OPTIONS	[string] 기본값: ' '	이 메뉴에서는 사용할 수 없지만 (고급) LAStools 사용자는 알고 있는 다른 명령 줄 스위치를 지정합니다.
Output raster file	OUTPUT_RASTER	[raster] 기본값: [Skip output]	산출 래스터를 어디에 저장할지 지정합니다. 적외선 영상 (false color), 회색조 색상표, 음영기복의 경우 TIF, PNG 및 JPG 같은 이미지 래스터를 사용합니다. 실제 값의 경우 TIF, BIL, IMG, ASC, DTM, FLT, XYZ 및 CSV 같은 값 래스터를 사용합니다. 다음 가운데 하나로 저장할 수 있습니다: • 산출물 건너뛰기 • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Output raster file	OUTPUT_RASTER	[raster]	산출 래스터

파이썬 코드

알고리즘 ID: `lastools:las2dem`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

24.3.4 las2iso

설명

등위선을 따라 그릴 임시 들로네 삼각분할을 생성해서 (파일 당 2 억에 달하는) 점구름을 등위선 라인으로 변환합니다.

더 자세한 정보는 las2iso 웹페이지와 온라인 README 파일을 참조하세요.

파라미터

라벨	명칭	유형	설명
verbose	VERBOSE	[boolean] 기본값: False	콘솔에 좀 더 서술적인 제어 산출물을 생성합니다.
run new 64 bit executable	CPU64	[boolean] 기본값: False	
open LAStools GUI	GUI	[boolean] 기본값: False	사전에 지정한 입력 파일을 가지고 있는 LAStools 의 GUI 를 실행합니다.
input LAS/LAZ file	INPUT_LASLAZ	[file]	등위선 라인을 생성하기 위해 사용할 포인트를 담고 있는 파일입니다.
smooth underlying TIN	SMOOTH	[number] 기본값: 0	임시 TIN 을 평탄화해야 하는지, 한다면 몇 번이나 반복해야 하는지를 지정합니다.
extract isoline with a spacing of	ISO_EVERY	[number] 기본값: 10.0	등위선 라인을 어떤 간격으로 추출할지를 지정합니다. (등고선 간격)
clean isolines shorter than (0 = do not clean)	CLEAN	[number] 기본값: 0.0	지정한 길이보다 짧은 등위선을 뺍니다.
simplify segments shorter than (0 = do not simplify)	SIMPLIFY_LENGTH	[number] 기본값: 0.0	지정한 길이보다 짧은 등위선을 가장 기본적인 형태로 단순화시킵니다.
simplify segment pairs with area less than (0 = do not simplify)	SIMPLIFY_AREA	[number] 기본값: 0.0	연속 선분으로 이루어진 요철 (bump) 부분 가운데 지정한 크기보다 작은 면적을 가진 요철을 가장 기본적인 형태로 단순화시킵니다.
additional command line parameter(s) 부가적	ADDITIONAL_OPTIONS	[string] 기본값: ''	이 메뉴에서는 사용할 수 없지만 (고급) LAStools 사용자는 알고 있는 다른 명령 줄 스위치를 지정합니다.
Output vector file	OUTPUT_VECTOR	[vector: line] 기본값: [Skip output]	어디에 산출 벡터를 저장할지 지정합니다. SHP 또는 WKT 산출 파일을 사용합니다. 사용자의 입력 LiDAR 파일이 지리 좌표 (경도/위도) 인 경우 또는 지리참조 정보를 가지고 있는 경우에만, KML 산출 파일도 생성할 수 있습니다. 다음 가운데 하나로 저장할 수 있습니다: • 산출물 건너뛰기 • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Output vector file	`OUTPUT_VECTOR`	[vector: line]	등위선을 담은 산출 라인 벡터 레이어

파이썬 코드

알고리즘 ID: `lastools:las2iso`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

24.3.5 las2las_filter

설명

las2las 알고리즘을 사용해서 서로 다른 속성을 바탕으로 LiDAR 포인트를 필터링하고, 필터링된 포인트 하위 집합을 새 LAZ 또는 LAS 파일로 작성합니다.

더 자세한 정보는 las2las 웹페이지와 온라인 README 파일을 참조하세요.

파라미터

라벨	명칭	유형	설명
verbose	`VERBOSE`	[boolean] 기본값: False	콘솔에 좀 더 서술적인 제어 산출물을 생성합니다.
run new 64 bit executable	`CPU64`	[boolean] 기본값: False	
open LAStools GUI	`GUI`	[boolean] 기본값: False	사전에 지정한 입력 파일을 가지고 있는 LAStools 의 GUI 를 실행합니다.
input LAS/LAZ file	`INPUT_LASLAZ`	[file]	등위선 라인을 생성하기 위해 사용할 포인트를 담고 있는 파일입니다.

다음 페이지에 계속

표 24.204 – 이전 페이지에서 계속

라벨	명칭	유형	설명
filter (by return, classification, flags)	FILTER_RETURN_CLASS_FLAGS	[enumeration] 기본값: 0	반환 (return), 범주화, 또는 플래그와 같은 여러 옵션을 바탕으로 포인트를 필터링합니다. 다음 가운데 하나로 지정할 수 있습니다: • 0 —– • 1 −keep_last • 2 −keep_first • 3 −keep_middle • 4 −keep_single • 5 −drop_single • 6 −keep_double • 7 −keep_class 2 • 8 −keep_class 2 8 • 9 −keep_class 8 • 10 −keep_class 6 • 11 −keep_class 9 • 12 −keep_class 3 4 5 • 13 —keep_class 3 • 14 —keep_class 4 • 15 —keep_class 5 • 16 —keep_class 2 6 • 17 —drop_class 7 • 18 —drop_withheld • 19 —drop_synthetic • 20 —drop_overlap • 21 —keep_withheld • 22 —keep_synthetic • 23 —keep_keypoint • 24 —keep_overlap

다음 페이지에 계속

표 24.204 – 이전 페이지에서 계속

라벨	명칭	유형	설명
second filter (by return, classification, flags)	FILTER_RETURN_CLASS_FLAGS_2	[enumeration] 기본값: 0	반환 (return), 범주화, 또는 플래그와 같은 여러 옵션을 바탕으로 포인트를 필터링합니다. 다음 가운데 하나로 지정할 수 있습니다: • 0 —— • 1 –keep_last • 2 –keep_first • 3 –keep_middle • 4 –keep_single • 5 –drop_single • 6 –keep_double • 7 –keep_class 2 • 8 –keep_class 2 8 • 9 –keep_class 8 • 10 –keep_class 6 • 11 –keep_class 9 • 12 –keep_class 3 4 5 • 13 —keep_class 3 • 14 —keep_class 4 • 15 —keep_class 5 • 16 —keep_class 2 6 • 17 —drop_class 7 • 18 —drop_withheld • 19 —drop_synthetic • 20 —drop_overlap • 21 —keep_withheld • 22 —keep_synthetic • 23 —keep_keypoint • 24 —keep_overlap

다음 페이지에 계속

표 24.204 – 이전 페이지에서 계속

라벨	명칭	유형	설명
filter (by coordinate, intensity, GPS time, …)	FILTER_COORDS_INTENS…	[enumeration] 기본값: 0	값을 인자로 요구하는 다른 옵션 여러 개를 바탕으로 포인트를 필터링합니다. 다음 가운데 하나로 지정할 수 있습니다: • 0 —– • 1 −drop_x_above • 2 −drop_x_below • 3 −drop_y_above • 4 −drop_y_below • 5 −drop_z_above • 6 −drop_z_below • 7 −drop_intensity_above • 8 −drop_intensity_below • 9 −drop_gps_time_above • 10 −drop_gps_time_below • 11 −drop_scan_angle_above • 12 −drop_scan_angle_below • 13 −keep_point_source • 14 −drop_point_source • 15 −drop_point_source_above • 16 −drop_point_source_below • 17 −keep_user_data • 18 −drop_user_data • 19 −drop_user_data_above • 20 −drop_user_data_below • 21 −keep_every_nth • 22 −keep_random_fraction • 23 −thin_with_grid
value for filter (by coordinate, intensity, GPS time, …)	FILTER_COORDS_INTENS…	[number] 기본값: None	앞에서 선택한 필터에 인자로 사용할 값

다음 페이지에 계속

표 24.204 – 이전 페이지에서 계속

라벨	명칭	유형	설명
second filter (by coordinate, intensity, GPS time, …)	FILTER_COORDS_INTENSITY2	[enumeration] 기본값: 0	값을 인자로 요구하는 다른 옵션 여러 개를 바탕으로 포인트를 필터링합니다. 다음 가운데 하나로 지정할 수 있습니다: • 0 —— • 1 −drop_x_above • 2 −drop_x_below • 3 −drop_y_above • 4 −drop_y_below • 5 −drop_z_above • 6 −drop_z_below • 7 −drop_intensity_above • 8 −drop_intensity_below • 9 −drop_gps_time_above • 10 −drop_gps_time_below • 11 −drop_scan_angle_above • 12 −drop_scan_angle_below • 13 −keep_point_source • 14 −drop_point_source • 15 −drop_point_source_above • 16 −drop_point_source_below • 17 −keep_user_data • 18 −drop_user_data • 19 −drop_user_data_above • 20 −drop_user_data_below • 21 −keep_every_nth • 22 −keep_random_fraction • 23 −thin_with_grid
value for second filter (by coordinate, intensity, GPS time, …)	FILTER_COORDS_INTENSITY2_VALUE	[number] 기본값: None	앞에서 선택한 필터에 인자로 사용할 값
additional command line parameter(s) 부가적	ADDITIONAL_OPTIONS	[string] 기본값: “	이 메뉴에서는 사용할 수 없지만 (고급) LAStools 사용자는 알고 있는 다른 명령 줄 스위치를 지정합니다.
Output LAS/LAZ file	OUTPUT_LASLAZ	[file] 기본값: [Skip output]	어디에 산출 점구름을 저장할지 지정합니다. 압축된 산출물의 경우 LAZ 파일을, 압축되지 않은 산출물의 경우 LAS 파일을, 그리고 ASCII 의 경우 TXT 파일을 사용합니다. 다음 가운데 하나로 저장할 수 있습니다: • 산출물 건너뛰기 • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Output LAS/LAZ file	`OUTPUT_LASLAZ`	[file]	산출 LAS/LAZ 포맷 파일

파이썬 코드

알고리즘 ID: `lastools:las2las_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 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

24.3.6 las2las_project

폴더에 있는 LAS/LAZ 파일을 또다른 좌표계로 변형합니다.

파라미터

라벨	명칭	유형	설명
verbose	`VERBOSE`	[boolean] 기본값: False	콘솔에 좀 더 서술적인 제어 산출물을 생성합니다.
run new 64 bit executable	`CPU64`	[boolean] 기본값: False	
open LAStools GUI	`GUI`	[boolean] 기본값: False	사전에 지정한 입력 파일을 가지고 있는 LAStools 의 GUI 를 실행합니다.
input LAS/LAZ file	`INPUT_LASLAZ`	[file]	입력 LAS/LAZ 파일
source projection	`SOURCE_PROJECTION`	[enumeration] 기본값: 0	다음 가운데 하나로 지정할 수 있습니다: • 0 —– • 1 —epsg (EPSG) • 2 —utm (UTM) • 3 —sp83 (NAD83) • 4 —sp27 (NAD27) • 5 —longlat (경도/위도) • 6 —latlong (위도/경도) • 7 —ecef (ECEF)

다음 페이지에 계속

표 24.205 – 이전 페이지에서 계속

라벨	명칭	유형	설명
source utm zone	SOURCE_UTM	[enumeration] 기본값: 0	다음 가운데 하나로 지정할 수 있습니다: • 0 —— • 1 −1 (북반구) • 2 −2 (북반구) • 3 −3 (북반구) • 4 −4 (북반구) • 5 −5 (북반구) • 6 −6 (북반구) • 7 −7 (북반구) • 8 −8 (북반구) • 9 −9 (북반구) • 10 −10 (북반구) • 11 −11 (북반구) • 12 −12 (북반구) • 13 −13 (북반구) • 14 −14 (북반구) • 15 −15 (북반구) • 16 −16 (북반구) • 17 −17 (북반구) • 18 −18 (북반구) • 19 −19 (북반구) • 20 −20 (북반구) • 21 −21 (북반구) • 22 −22 (북반구) • 23 −23 (북반구) • 24 −24 (북반구) • 25 −25 (북반구) • 26 −26 (북반구) • 27 −27 (북반구) • 28 −28 (북반구) • 29 −29 (북반구) • 30 −30 (북반구) • 31 −31 (북반구) • 32 −32 (북반구) • 33 −33 (북반구) • 34 −34 (북반구) • 35 −35 (북반구) • 36 −36 (북반구) • 37 −37 (북반구) • 38 −38 (북반구) • 39 −39 (북반구) • 40 −40 (북반구) • 41 −41 (북반구) • 42 −42 (북반구) • 43 −43 (북반구) • 44 −44 (북반구) • 45 −45 (북반구) • 46 −46 (북반구) • 47 −47 (북반구) • 48 −48 (북반구) • 49 −49 (북반구) • 50 −50 (북반구) • 51 −51 (북반구) • 52 −52 (북반구) • 53 −53 (북반구) • 54 −54 (북반구) • 55 −55 (북반구) • 56 −56 (북반구)

표 24.205 – 이전 페이지에서 계속

라벨	명칭	유형	설명
source state plane code	SOURCE_SP	[enumeration] 기본값: 0	다음 가운데 하나로 지정할 수 있습니다: • 0 —– • 1 −AK_10 • 2 −AK_2 • 3 −AK_3 • 4 −AK_4 • 5 −AK_5 • 6 −AK_6 • 7 −AK_7 • 8 −AK_8 • 9 −AK_9 • 10 −AL_E • 11 −AL_W • 12 −AR_N • 13 −AR_S • 14 −AZ_C • 15 −AZ_E • 16 −AZ_W • 17 −CA_I • 18 −CA_II • 19 −CA_III • 20 −CA_IV • 21 −CA_V • 22 −CA_VI • 23 −CA_VII • 24 −CO_C • 25 −CO_N • 26 −CO_S • 27 −CT • 28 −DE • 29 −FL_E • 30 −FL_N • 31 −FL_W • 32 −GA_E • 33 −GA_W • 34 −HI_1 • 35 −HI_2 • 36 −HI_3 • 37 −HI_4 • 38 −HI_5 • 39 −IA_N • 40 −IA_S • 41 −ID_C • 42 −ID_E • 43 −ID_W • 44 −IL_E • 45 −IL_W • 46 −IN_E • 47 −IN_W • 48 −KS_N • 49 −KS_S • 50 −KY_N • 51 −KY_S • 52 −LA_N • 53 −LA_S • 54 −MA_I • 55 −MA_M • 56 −MD

표 24.205 – 이전 페이지에서 계속

라벨	명칭	유형	설명
target projection	TAR-GET_PROJECTION	[enumeration] 기본값: 0	다음 가운데 하나로 지정할 수 있습니다: • 0 —— • 1 —epsg (EPSG) • 2 —utm (UTM) • 3 —sp83 (NAD83) • 4 —sp27 (NAD27) • 5 —longlat (경도/위도) • 6 —latlong (위도/경도) • 7 —ecef (ECEF)

다음 페이지에 계속

표 24.205 – 이전 페이지에서 계속

라벨	명칭	유형	설명
target utm zone	TARGET_UTM	[enumeration] 기본값: 0	다음 가운데 하나로 지정할 수 있습니다: • 0 —— • 1 −1 (북반구) • 2 −2 (북반구) • 3 −3 (북반구) • 4 −4 (북반구) • 5 −5 (북반구) • 6 −6 (북반구) • 7 −7 (북반구) • 8 −8 (북반구) • 9 −9 (북반구) • 10 −10 (북반구) • 11 −11 (북반구) • 12 −12 (북반구) • 13 −13 (북반구) • 14 −14 (북반구) • 15 −15 (북반구) • 16 −16 (북반구) • 17 −17 (북반구) • 18 −18 (북반구) • 19 −19 (북반구) • 20 −20 (북반구) • 21 −21 (북반구) • 22 −22 (북반구) • 23 −23 (북반구) • 24 −24 (북반구) • 25 −25 (북반구) • 26 −26 (북반구) • 27 −27 (북반구) • 28 −28 (북반구) • 29 −29 (북반구) • 30 −30 (북반구) • 31 −31 (북반구) • 32 −32 (북반구) • 33 −33 (북반구) • 34 −34 (북반구) • 35 −35 (북반구) • 36 −36 (북반구) • 37 −37 (북반구) • 38 −38 (북반구) • 39 −39 (북반구) • 40 −40 (북반구) • 41 −41 (북반구) • 42 −42 (북반구) • 43 −43 (북반구) • 44 −44 (북반구) • 45 −45 (북반구) • 46 −46 (북반구) • 47 −47 (북반구) • 48 −48 (북반구) • 49 −49 (북반구) • 50 −50 (북반구) • 51 −51 (북반구) • 52 −52 (북반구) • 53 −53 (북반구) • 54 −54 (북반구) • 55 −55 (북반구) • 56 −56 (북반구)

표 24.205 – 이전 페이지에서 계속

라벨	명칭	유형	설명
target state plane code	TARGET_SP	[enumeration] 기본값: 0	다음 가운데 하나로 지정할 수 있습니다: • 0 —– • 1 −AK_10 • 2 −AK_2 • 3 −AK_3 • 4 −AK_4 • 5 −AK_5 • 6 −AK_6 • 7 −AK_7 • 8 −AK_8 • 9 −AK_9 • 10 −AL_E • 11 −AL_W • 12 −AR_N • 13 −AR_S • 14 −AZ_C • 15 −AZ_E • 16 −AZ_W • 17 −CA_I • 18 −CA_II • 19 −CA_III • 20 −CA_IV • 21 −CA_V • 22 −CA_VI • 23 −CA_VII • 24 −CO_C • 25 −CO_N • 26 −CO_S • 27 −CT • 28 −DE • 29 −FL_E • 30 −FL_N • 31 −FL_W • 32 −GA_E • 33 −GA_W • 34 −HI_1 • 35 −HI_2 • 36 −HI_3 • 37 −HI_4 • 38 −HI_5 • 39 −IA_N • 40 −IA_S • 41 −ID_C • 42 −ID_E • 43 −ID_W • 44 −IL_E • 45 −IL_W • 46 −IN_E • 47 −IN_W • 48 −KS_N • 49 −KS_S • 50 −KY_N • 51 −KY_S • 52 −LA_N • 53 −LA_S • 54 −MA_I • 55 −MA_M • 56 −MD

표 24.205 – 이전 페이지에서 계속

라벨	명칭	유형	설명
additional command line parameter(s) 부가적	ADDITIONAL_OPTIONS	[string] 기본값: ''	이 메뉴에서는 사용할 수 없지만 (고급) LAStools 사용자는 알고 있는 다른 명령 줄 스위치를 지정합니다.
Output LAS/LAZ file	OUTPUT_LASLAZ	[folder] 기본값: [Save to temporary folder]	산출 점구름을 저장할 폴더를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 산출물 건너뛰기 • 임시 디렉터리에 저장 • 디렉터리에 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Output LAS/LAZ file	OUTPUT_LASLAZ	[file]	산출 LAS/LAZ 포맷 파일

파이썬 코드

알고리즘 ID: `lastools:las2las_project`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

24.3.7 las2las_transform

설명

las2las 알고리즘을 사용해서 서로 다른 속성을 바탕으로 LiDAR 포인트를 필터링하고, 필터링된 포인트 하위 집합을 새 LAZ 또는 LAS 파일로 작성합니다.

더 자세한 정보는 las2las 웹페이지와 온라인 README 파일을 참조하세요.

파라미터

라벨	명칭	유형	설명
verbose	VERBOSE	[boolean] 기본값: False	콘솔에 좀 더 서술적인 제어 산출물을 생성합니다.
run new 64 bit executable	CPU64	[boolean] 기본값: False	

다음 페이지에 계속

표 24.206 – 이전 페이지에서 계속

라벨	명칭	유형	설명
open LAStools GUI	GUI	[boolean] 기본값: False	사전에 지정한 입력 파일을 가지고 있는 LAStools 의 GUI 를 실행합니다.
input LAS/LAZ file	INPUT_LASLAZ	[file]	병합할 포인트를 담고 있는 첫 번째 파일
transform (coordinates)	TRANSFORM_COORDINATE1	[enumeration] 기본값: 0	아래에서 지정한 값으로 X, Y, 또는 Z 좌표를 변형 (translate) 하거나, 척도를 변경 (scale) 하거나, 고정 (clamp) 시킵니다. 다음 가운데 하나로 지정할 수 있습니다: • 0 —— • 1 −translate_x • 2 −translate_y • 3 −translate_z • 4 −scale_x • 5 −scale_y • 6 −scale_z • 7 −clamp_z_above • 8 −clamp_z_below
value for transform (coordinates)	TRANSFORM_COORDINATE1	[string] 기본값: ''	앞의 파라미터에서 선택한 변형 (transform) 작업이 어느 정도의 변형, 크기 조정, 또는 고정을 할지 지정하는 값입니다.
second transform (coordinates)	TRANSFORM_COORDINATE2	[enumeration] 기본값: 0	아래에서 지정한 값으로 X, Y, 또는 Z 좌표를 변형 (translate) 하거나, 척도를 변경 (scale) 하거나, 고정 (clamp) 시킵니다. 다음 가운데 하나로 지정할 수 있습니다: • 0 —— • 1 −translate_x • 2 −translate_y • 3 −translate_z • 4 −scale_x • 5 −scale_y • 6 −scale_z • 7 −clamp_z_above • 8 −clamp_z_below
value for second transform (coordinates)	TRANSFORM_COORDINATE2	[string] 기본값: ''	앞의 파라미터에서 선택한 변형 (transform) 작업이 어느 정도의 변형, 크기 조정, 또는 고정을 할지 지정하는 값입니다.

다음 페이지에 계속

표 24.206 – 이전 페이지에서 계속

라벨	명칭	유형	설명
transform (intensities, scan angles, GPS times, …)	TRANSFORM_OTHER1	[enumeration] 기본값: 0	아래에서 지정한 값으로 X, Y, 또는 Z 좌표를 변형 (translate) 하거나, 척도를 변경 (scale) 하거나, 고정 (clamp) 시킵니다. 다음 가운데 하나로 지정할 수 있습니다: • 0 —— • 1 −scale_intensity • 2 −translate_intensity • 3 −clamp_intensity_above • 4 −clamp_intensity_below • 5 −scale_scan_angle • 6 −translate_scan_angle • 7 −translate_gps_time • 8 −set_classification • 9 −set_user_data • 10 −set_point_source • 11 −scale_rgb_up • 12 −scale_rgb_down • 13 −repair_zero_returns
value for transform (intensities, scan angles, GPS times, …)	TRANSFORM_OTHER1_ARG	[string] 기본값: ˚	앞의 파라미터에서 선택한 변형 (transform) 작업이 어느 정도의 변형, 크기 조정, 고정, 또는 설정을 할지 지정하는 값입니다.
second transform (intensities, scan angles, GPS times, …)	TRANSFORM_OTHER2	[enumeration] 기본값: 0	아래에서 지정한 값으로 X, Y, 또는 Z 좌표를 변형 (translate) 하거나, 척도를 변경 (scale) 하거나, 고정 (clamp) 시킵니다. 다음 가운데 하나로 지정할 수 있습니다: • 0 —— • 1 −scale_intensity • 2 −translate_intensity • 3 −clamp_intensity_above • 4 −clamp_intensity_below • 5 −scale_scan_angle • 6 −translate_scan_angle • 7 −translate_gps_time • 8 −set_classification • 9 −set_user_data • 10 −set_point_source • 11 −scale_rgb_up • 12 −scale_rgb_down • 13 −repair_zero_returns
value for second transform (intensities, scan angles, GPS times, …)	TRANSFORM_OTHER2_ARG	[string] 기본값: ˚	앞의 파라미터에서 선택한 변형 (transform) 작업이 어느 정도의 변형, 크기 조정, 고정, 또는 설정을 할지 지정하는 값입니다.

다음 페이지에 계속

표 24.206 – 이전 페이지에서 계속

라벨	명칭	유형	설명
operations (first 7 need an argument)	OPERATION	[enumeration] 기본값: 0	다음 가운데 하나로 지정할 수 있습니다: • 0 —— • 1 −set_point_type • 2 −set_point_size • 3 −set_version_minor • 4 −set_version_major • 5 −start_at_point • 6 −stop_at_point • 7 −remove_vlr • 8 −auto_reoffset • 9 −week_to_adjusted • 10 −adjusted_to_week • 11 —auto reoffset • 12 —scale_rgb_up • 13 —scale_rgb_down • 14 —remove_all_vlrs • 15 —remove_extra • 16 —clip_to_bounding_box
argument for operation	OPERATIONARG	[string] 기본값: ‘’	앞에서 선택한 작업에 인자로 사용할 값
additional command line parameter(s) 부가적	ADDITIONAL_OPTIONS	[string] 기본값: ‘’	이 메뉴에서는 사용할 수 없지만 (고급) LAStools 사용자는 알고 있는 다른 명령 줄 스위치를 지정합니다.
Output LAS/LAZ file	OUTPUT_LASLAZ	[file] 기본값: [Skip output]	어디에 산출 점구름을 저장할지 지정합니다. 압축된 산출물의 경우 LAZ 파일을, 압축되지 않은 산출물의 경우 LAS 파일을, 그리고 ASCII 의 경우 TXT 파일을 사용합니다. 다음 가운데 하나로 저장할 수 있습니다: • 산출물 건너뛰기 • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Output LAS/LAZ file	OUTPUT_LASLAZ	[file]	산출 (병합된) LAS/LAZ 포맷 파일

파이썬 코드

알고리즘 ID: `lastools:las2las_transform`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

24.3.8 las2txt

설명

LAS/LAZ 파일을 텍스트 파일로 변형합니다.

파라미터

라벨	명칭	유형	설명
verbose	VERBOSE	[boolean] 기본값: False	
run new 64 bit executable	CPU64	[boolean] 기본값: False	
open LAStools GUI	GUI	[boolean] 기본값: False	
input LAS/LAZ file	INPUT_LASLAZ	[file] 기본값: None	
parse_string	PARSE	[string] 기본값: 'xyz'	
additional command line parameters 부가적	ADDITIONAL_OPTIONS	[string] 기본값: ''	이 메뉴에서는 사용할 수 없지만 (고급) LAStools 사용자는 알고 있는 다른 명령 줄 스위치를 지정합니다.
Output ASCII file	OUTPUT_GENERIC	[file] 기본값: [Create temporary layer]	산출 파일을 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 레이어 생성 (TEMPORARY_OUTPUT) • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Output ASCII file	OUTPUT_GENERIC	[file]	산출 파일

파이썬 코드

알고리즘 ID: `lastools:las2txt`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

24.3.9 lasindex

설명

< 알고리즘 설명 추가할 것 >

파라미터

라벨	명칭	유형	설명
verbose	VERBOSE	[boolean] 기본값: False	
run new 64 bit executable	CPU64	[boolean] 기본값: False	
open LAStools GUI	GUI	[boolean] 기본값: False	
input LAS/LAZ file	INPUT_LASLAZ	[file] 기본값: None	
append *.lax file to *.laz file	APPEND_LAX	[boolean] 기본값: False	
is mobile or terrestrial LiDAR (not airborne)	MOBILE_OR_TERRESTRIAL	[boolean] 기본값: False	
additional command line parameters 부가적	ADDITIONAL_OPTIONS	[string] 기본값: ' '	이 메뉴에서는 사용할 수 없지만 (고급) LAStools 사용자는 알고 있는 다른 명령 줄 스위치를 지정합니다.

산출물

이 알고리즘은 아무것도 산출하지 않습니다.

파이썬 코드

알고리즘 ID: `lastools:lasindex`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

24.3.10 lasgrid

사용자가 특정 (예: 최소값, 최대값, 평균 등) 메소드를 사용해서 래스터 상에 정의한 단계 크기를 가진 대용량 점구름에서 (예를 들어 표고, 강도, 범주화, 스캔 각도 등의) 선택한 속성으로 그리드를 생성합니다.

더 자세한 정보는 lasgrid 웹페이지와 온라인 README 파일을 참조하세요.

파라미터

라벨	명칭	유형	설명
verbose	`VERBOSE`	[boolean] 기본값: False	콘솔에 좀 더 서술적인 제어 산출물을 생성합니다.
run new 64 bit executable	`CPU64`	[boolean] 기본값: False	
open LAStools GUI	`GUI`	[boolean] 기본값: False	사전에 지정한 입력 파일을 가지고 있는 LAStools 의 GUI 를 실행합니다.
input LAS/LAZ file	`INPUT_LASLAZ`	[file]	래스터화될 포인트를 담고 있는 LAS/LAZ 포맷 파일입니다.

다음 페이지에 계속

표 24.209 – 이전 페이지에서 계속

라벨	명칭	유형	설명
filter (by return, classification, flags)	FIL-TER_RETURN_CLASS_FLAGS	[enumeration] 기본값: 0	그리드 작업에 사용할 포인트의 하위 집합을 지정합니다. 다음 가운데 하나를 선택할 수 있습니다: • 0 — • 1 −keep_last • 2 −keep_first • 3 −keep_middle • 4 −keep_single • 5 −drop_single • 6 −keep_double • 7 −keep_class 2 • 8 −keep_class 2 8 • 9 −keep_class 8 • 10 −keep_class 6 • 11 −keep_class 9 • 12 −keep_class 3 4 5 • 13 —keep_class 3 • 14 —keep_class 4 • 15 —keep_class 5 • 16 —keep_class 2 6 • 17 —drop_class 7 • 18 —drop_withheld • 19 —drop_synthetic • 20 —drop_overlap • 21 —keep_withheld • 22 —keep_synthetic • 23 —keep_keypoint • 24 —keep_overlap
step size / pixel size	STEP	[number] 기본값: 1.0	TIN 이 래스터화되는 그리드의 셀 크기를 지정합니다.
Attribute	ATTRIBUTE	[enumeration] 기본값: 0	래스터화할 속성을 지정합니다. 다음 가운데 하나를 선택할 수 있습니다: • 0 —표고 • 1 —강도 • 2 —RGB • 3 —범주화
Method	METHOD	[enumeration] 기본값: 0	어떤 셀 안에 떨어지는 속성을 어떻게 래스터 값으로 변환할지 지정합니다. 다음 가운데 하나를 선택할 수 있습니다: • 0 —최저 (lowest) • 1 —최고 (heighest) • 2 —평균 (average) • 3 —표준 편차 (stddev)
use tile bounding box (after tiling with buffer)	USE_TILE_BB	[boolean] 기본값: False	래스터화 영역을 타일 경계 상자로 제한합니다. (입력물이 lastile 로 생성한 LAS/LAZ 타일인 경우에만 의미가 있습니다.)

다음 페이지에 계속

표 24.209 – 이전 페이지에서 계속

라벨	명칭	유형	설명
additional command line parameter(s) 부가적	ADDITIONAL_OPTIONS	[string] 기본값: ''	이 메뉴에서는 사용할 수 없지만 (고급) LAStools 사용자는 알고 있는 다른 명령 줄 스위치를 지정합니다.
Output raster file	OUTPUT_RASTER	[raster] 기본값: [Skip output]	산출 래스터를 어디에 저장할지 지정합니다. 적외선 영상 (false color), 회색조 색상표, 음영기복의 경우 TIF, PNG 및 JPG 같은 이미지 래스터를 사용합니다. 실제 값의 경우 TIF, BIL, IMG, ASC, DTM, FLT, XYZ 및 CSV 같은 값 래스터를 사용합니다. 다음 가운데 하나로 저장할 수 있습니다: • 산출물 건너뛰기 • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Output raster file	OUTPUT_RASTER	[raster]	산출 래스터

파이썬 코드

알고리즘 ID: `lastools:lasgrid`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

24.3.11 lasinfo

파라미터

라벨	명칭	유형	설명
verbose	VERBOSE	[boolean] 기본값: False	콘솔에 좀 더 서술적인 제어 산출물을 생성합니다.
run new 64 bit executable	CPU64	[boolean] 기본값: False	
open LAStools GUI	GUI	[boolean] 기본값: False	사전에 지정한 입력 파일을 가지고 있는 LAStools 의 GUI 를 실행합니다.
input LAS/LAZ file	INPUT_LASLAZ	[file]	정보를 가져올 파일

다음 페이지에 계속

표 24.210 – 이전 페이지에서 계속

라벨	명칭	유형	설명
compute density	COMPUTE_DENSITY	[boolean] 기본값: False	
repair bounding box	REPAIR_BB	[boolean] 기본값: False	
repair counters	REPAIR_COUNTERS	[boolean] 기본값: False	
histogram	HISTO1	[enumeration] 기본값: 0	첫 번째 히스토그램을 지정합니다. 다음 가운데 하나를 선택할 수 있습니다: • 0 —– • 1 —x • 2 —y • 3 —z • 4 —intensity • 5 —classification • 6 —scan_angle • 7 —user_data • 8 —point_source • 9 —gps_time • 10 —X • 11 —Y • 12 —Z • 13 —attribute0 • 14 —attribute1 • 15 —attribute2
bin size	HISTO1_BIN	[number] 기본값: 1.0	
histogram	HISTO2	[enumeration] 기본값: 0	두 번째 히스토그램을 지정합니다. 다음 가운데 하나를 선택할 수 있습니다: • 0 —– • 1 —x • 2 —y • 3 —z • 4 —intensity • 5 —classification • 6 —scan_angle • 7 —user_data • 8 —point_source • 9 —gps_time • 10 —X • 11 —Y • 12 —Z • 13 —attribute0 • 14 —attribute1 • 15 —attribute2
bin size	HISTO2_BIN	[number] 기본값: 1.0	

다음 페이지에 계속

표 24.210 – 이전 페이지에서 계속

라벨	명칭	유형	설명
histogram	HISTO3	[enumeration] 기본값: 0	세 번째 히스토그램을 지정합니다. 다음 가운데 하나를 선택할 수 있습니다: • 0 — • 1 —x • 2 —y • 3 —z • 4 —intensity • 5 —classification • 6 —scan_angle • 7 —user_data • 8 —point_source • 9 —gps_time • 10 —X • 11 —Y • 12 —Z • 13 —attribute0 • 14 —attribute1 • 15 —attribute2
bin size	HISTO3_BIN	[number] 기본값: 1.0	
additional command line parameter(s) 부가적	ADDITIONAL_OPTIONS	[string] 기본값: ''	이 메뉴에서는 사용할 수 없지만 (고급) LAStools 사용자는 알고 있는 다른 명령 줄 스위치를 지정합니다.
Output ASCII file	OUTPUT_GENERIC	[file] 기본값: [Skip output]	어디에 산출물을 저장할지 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 산출물 건너뛰기 • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Output ASCII file	OUTPUT_GENERIC	[file]	산출 파일을 담고 있는 파일

파이썬 코드

알고리즘 ID: `lastools:lasinfo`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

24.3.12 lasmerge

최대 7 개의 LAS/LAZ 파일을 하나로 병합합니다.

파라미터

라벨	명칭	유형	설명
verbose	VERBOSE	[boolean] 기본값: False	콘솔에 좀 더 서술적인 제어 산출물을 생성합니다.
run new 64 bit executable	CPU64	[boolean] 기본값: False	
open LAStools GUI	GUI	[boolean] 기본값: False	사전에 지정한 입력 파일을 가지고 있는 LAStools 의 GUI 를 실행합니다.
files are flightlines	FILES_ARE_FLIGHTLINES	[boolean] 기본값: False	
apply file source ID	APPLY_FILE_SOURCE_ID	[boolean] 기본값: False	
input LAS/LAZ file	INPUT_LASLAZ	[file]	병합할 포인트를 담고 있는 첫 번째 파일
2nd file 부가적	FILE2	[file]	병합할 두 번째 파일
3rd file 부가적	FILE3	[file]	병합할 세 번째 파일
4th file 부가적	FILE4	[file]	병합할 네 번째 파일
5th file 부가적	FILE5	[file]	병합할 다섯 번째 파일
6th file 부가적	FILE6	[file]	병합할 여섯 번째 파일
7th file 부가적	FILE7	[file]	병합할 일곱 번째 파일
additional command line parameter(s) 부가적	ADDITIONAL_OPTIONS	[string] 기본값: ''	이 메뉴에서는 사용할 수 없지만 (고급) LAStools 사용자는 알고 있는 다른 명령 줄 스위치를 지정합니다.
Output LAS/LAZ file	OUTPUT_LASLAZ	[file] 기본값: [Skip output]	어디에 산출 점구름을 저장할지 지정합니다. 압축된 산출물의 경우 LAZ 파일을, 압축되지 않은 산출물의 경우 LAS 파일을, 그리고 ASCII 의 경우 TXT 파일을 사용합니다. 다음 가운데 하나로 저장할 수 있습니다: • 산출물 건너뛰기 • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Output LAS/LAZ file	OUTPUT_LASLAZ	[file]	산출 (병합된) LAS/LAZ 포맷 파일

파이썬 코드

알고리즘 ID: `lastools:lasmerge`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

24.3.13 lasprecision

파라미터

라벨	명칭	유형	설명
verbose	VERBOSE	[boolean] 기본값: False	콘솔에 좀 더 서술적인 제어 산출물을 생성합니다.
open LAStools GUI	GUI	[boolean] 기본값: False	사전에 지정한 입력 파일을 가지고 있는 LAStools 의 GUI 를 실행합니다.
input LAS/LAZ file	INPUT_LASLAZ	[file]	입력 점구름 파일
additional command line parameter(s) 부가적	ADDITIONAL_OPTIONS	[string] 기본값: ' '	이 메뉴에서는 사용할 수 없지만 (고급) LAStools 사용자는 알고 있는 다른 명령 줄 스위치를 지정합니다.
Output ASCII file	OUTPUT_GENERIC	[file] 기본값: [Skip output]	어디에 산출 ASCII 파일을 저장할지 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 산출물 건너뛰기 • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Output ASCII file	OUTPUT_GENERIC	[file]	산출 ASCII 파일

파이썬 코드

알고리즘 ID: `lastools:lasprecision`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

24.3.14 lasquery

설명

< 알고리즘 설명 추가할 것 >

파라미터

라벨	명칭	유형	설명
verbose	VERBOSE	[boolean] 기본값: False	콘솔에 좀 더 서술적인 제어 산출물을 생성합니다.
open LAStools GUI	GUI	[boolean] 기본값: False	사전에 지정한 입력 파일을 가지고 있는 LAStools 의 GUI 를 실행합니다.
input LAS/LAZ file	INPUT_LASLAZ	[file]	입력 점구름 파일
area of interest	AOI	[extent]	범위
additional command line parameter(s) 부가적	ADDITIONAL_OPTIONS	[string] 기본값: ' '	이 메뉴에서는 사용할 수 없지만 (고급) LAStools 사용자는 알고 있는 다른 명령 줄 스위치를 지정합니다.

산출물

파이썬 코드

알고리즘 ID: `lastools:lasquery`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

24.3.15 lasvalidate

파라미터

라벨	명칭	유형	설명
input LAS/LAZ file	INPUT_LASLAZ	[file]	입력 점구름 파일
save report to '*_LVS.xml'	ONE_REPORT_PER_FILE	[boolean]	
additional command line parameter(s) 부가적	ADDITIONAL_OPTIONS	[string] 기본값: ''	이 메뉴에서는 사용할 수 없지만 (고급) LAStools 사용자는 알고 있는 다른 명령 줄 스위치를 지정합니다.
Output XML file	OUTPUT_GENERIC	[file] 기본값: [Skip output]	어디에 산출 XML 파일을 저장할지 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 산출물 건너뛰기 • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Output XML file	OUTPUT_GENERIC	[file]	산출 XML 파일

파이썬 코드

알고리즘 ID: `lastools:lasvalidate`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

24.3.16 laszip

파라미터

라벨	명칭	유형	설명
verbose	VERBOSE	[boolean] 기본값: False	콘솔에 좀 더 서술적인 제어 산출물을 생성합니다.
run new 64 bit executable	CPU64	[boolean] 기본값: False	

다음 페이지에 계속

표 24.215 – 이전 페이지에서 계속

라벨	명칭	유형	설명
open LAStools GUI	GUI	[boolean] 기본값: False	사전에 지정한 입력 파일을 가지고 있는 LAStools 의 GUI 를 실행합니다.
input LAS/LAZ file	INPUT_LASLAZ	[file]	ZIP 파일로 압축할 파일
only report size	REPORT_SIZE	[boolean] 기본값: False	
create spatial indexing file (*.lax)	CREATE_LAX	[boolean] 기본값: False	
append *.lax into *.laz file	APPEND_LAX	[boolean] 기본값: False	
additional command line parameter(s) 부가적	ADDITIONAL_OPTIONS	[string] 기본값: ''	이 메뉴에서는 사용할 수 없지만 (고급) LAStools 사용자는 알고 있는 다른 명령 줄 스위치를 지정합니다.
Output LAS/LAZ file	OUTPUT_LASLAZ	[file] 기본값: [Skip output]	어디에 산출 점구름을 저장할지 지정합니다. 압축된 산출물의 경우 LAZ 파일을, 압축되지 않은 산출물의 경우 LAS 파일을, 그리고 ASCII 의 경우 TXT 파일을 사용합니다. 다음 가운데 하나로 저장할 수 있습니다: • 산출물 건너뛰기 • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Output LAS/LAZ file	OUTPUT_LASLAZ	[file]	산출 파일

파이썬 코드

알고리즘 ID: `lastools:laszip`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

24.3.17 txt2las

파라미터

라벨	명칭	유형	설명
verbose	VERBOSE	[boolean] 기본값: False	콘솔에 좀 더 서술적인 제어 산출물을 생성합니다.
run new 64 bit executable	CPU64	[boolean] 기본값: False	
open LAStools GUI	GUI	[boolean] 기본값: False	사전에 지정한 입력 파일을 가지고 있는 LAStools 의 GUI 를 실행합니다.
input LAS/LAZ file	INPUT_LASLAZ	[file]	ZIP 파일로 압축할 파일
parse lines as	PARSE	[string] 기본값: 'xyz'	
skip the first n lines	SKIP	[number] 기본값: 0	
resolution of x and y coordinate	SCALE_FACTOR_XY	[number] 기본값: 0.01	
resolution of z coordinate	SCALE_FACTOR_Z	[number] 기본값: 0.01	
resolution of z coordinate	SCALE_FACTOR_Z	[number] 기본값: 0.01	
source projection	PROJECTION	[enumeration] 기본값: 0	다음 가운데 하나로 지정할 수 있습니다: • 0 —- • 1 —epsg (EPSG) • 2 —utm (UTM) • 3 —sp83 (NAD83) • 4 —sp27 (NAD27) • 5 —longlat (경도/위도) • 6 —latlong (위도/경도) • 7 —ecef (ECEF)
source epsg code	EPSG_CODE	[number]	

다음 페이지에 계속

표 24.216 – 이전 페이지에서 계속

라벨	명칭	유형	설명
utm zone	`UTM`	[enumeration] 기본값: 0	다음 가운데 하나로 지정할 수 있습니다: • 0 —— • 1 −1 (북반구) • 2 −2 (북반구) • 3 −3 (북반구) • 4 −4 (북반구) • 5 −5 (북반구) • 6 −6 (북반구) • 7 −7 (북반구) • 8 −8 (북반구) • 9 −9 (북반구) • 10 −10 (북반구) • 11 −11 (북반구) • 12 −12 (북반구) • 13 −13 (북반구) • 14 −14 (북반구) • 15 −15 (북반구) • 16 −16 (북반구) • 17 −17 (북반구) • 18 −18 (북반구) • 19 −19 (북반구) • 20 −20 (북반구) • 21 −21 (북반구) • 22 −22 (북반구) • 23 −23 (북반구) • 24 −24 (북반구) • 25 −25 (북반구) • 26 −26 (북반구) • 27 −27 (북반구) • 28 −28 (북반구) • 29 −29 (북반구) • 30 −30 (북반구) • 31 −31 (북반구) • 32 −32 (북반구) • 33 −33 (북반구) • 34 −34 (북반구) • 35 −35 (북반구) • 36 −36 (북반구) • 37 −37 (북반구) • 38 −38 (북반구) • 39 −39 (북반구) • 40 −40 (북반구) • 41 −41 (북반구) • 42 −42 (북반구) • 43 −43 (북반구) • 44 −44 (북반구) • 45 −45 (북반구) • 46 −46 (북반구) • 47 −47 (북반구) • 48 −48 (북반구) • 49 −49 (북반구) • 50 −50 (북반구) • 51 −51 (북반구) • 52 −52 (북반구) • 53 −53 (북반구) • 54 −54 (북반구) • 55 −55 (북반구) • 56 −56 (북반구)

표 24.216 – 이전 페이지에서 계속

라벨	명칭	유형	설명
state plane code	SP	[enumeration] 기본값: 0	다음 가운데 하나로 지정할 수 있습니다: • 0 —– • 1 −AK_10 • 2 −AK_2 • 3 −AK_3 • 4 −AK_4 • 5 −AK_5 • 6 −AK_6 • 7 −AK_7 • 8 −AK_8 • 9 −AK_9 • 10 −AL_E • 11 −AL_W • 12 −AR_N • 13 −AR_S • 14 −AZ_C • 15 −AZ_E • 16 −AZ_W • 17 −CA_I • 18 −CA_II • 19 −CA_III • 20 −CA_IV • 21 −CA_V • 22 −CA_VI • 23 −CA_VII • 24 −CO_C • 25 −CO_N • 26 −CO_S • 27 −CT • 28 −DE • 29 −FL_E • 30 −FL_N • 31 −FL_W • 32 −GA_E • 33 −GA_W • 34 −HI_1 • 35 −HI_2 • 36 −HI_3 • 37 −HI_4 • 38 −HI_5 • 39 −IA_N • 40 −IA_S • 41 −ID_C • 42 −ID_E • 43 −ID_W • 44 −IL_E • 45 −IL_W • 46 −IN_E • 47 −IN_W • 48 −KS_N • 49 −KS_S • 50 −KY_N • 51 −KY_S • 52 −LA_N • 53 −LA_S • 54 −MA_I • 55 −MA_M • 56 −MD

표 24.216 – 이전 페이지에서 계속

라벨	명칭	유형	설명
additional command line parameter(s) 부가적	`ADDITIONAL_OPTIONS`	[string] 기본값: ''	이 메뉴에서는 사용할 수 없지만 (고급) LAStools 사용자는 알고 있는 다른 명령 줄 스위치를 지정합니다.
Output LAS/LAZ file	`OUTPUT_LASLAZ`	[file] 기본값: `[Skip output]`	어디에 산출 점구름을 저장할지 지정합니다. 압축된 산출물의 경우 LAZ 파일을, 압축되지 않은 산출물의 경우 LAS 파일을, 그리고 ASCII 의 경우 TXT 파일을 사용합니다. 다음 가운데 하나로 저장할 수 있습니다: • 산출물 건너뛰기 • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
output LAS/LAZ file	`OUTPUT_LASLAZ`	[file]	산출 파일

파이썬 코드

알고리즘 ID: `lastools:txt2las`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

24.4 TauDEM 알고리즘 제공자

TauDEM (Terrain Analysis Using Digital Elevation Models) 은 수치 표고 모델 (Digital Elevation Model; DEM) 이 표현하는 지형으로부터 수문학적 정보를 추출하고 분석하기 위한 DEM 도구들의 집합입니다. 유타 주립 대학 (Utah State University; USU) 에서 수문학적 수치 표고 모델 분석 및 유역 묘사 (watershed delineation) 를 위해 이 소프트웨어를 개발했습니다.

TauDEM 은 윈도우 용 독립형 명령 줄 실행 프로그램 및 다른 시스템 상에서 컴파일하고 사용하기 위한 소스 코드를 함께 묶어 배포하고 있습니다.

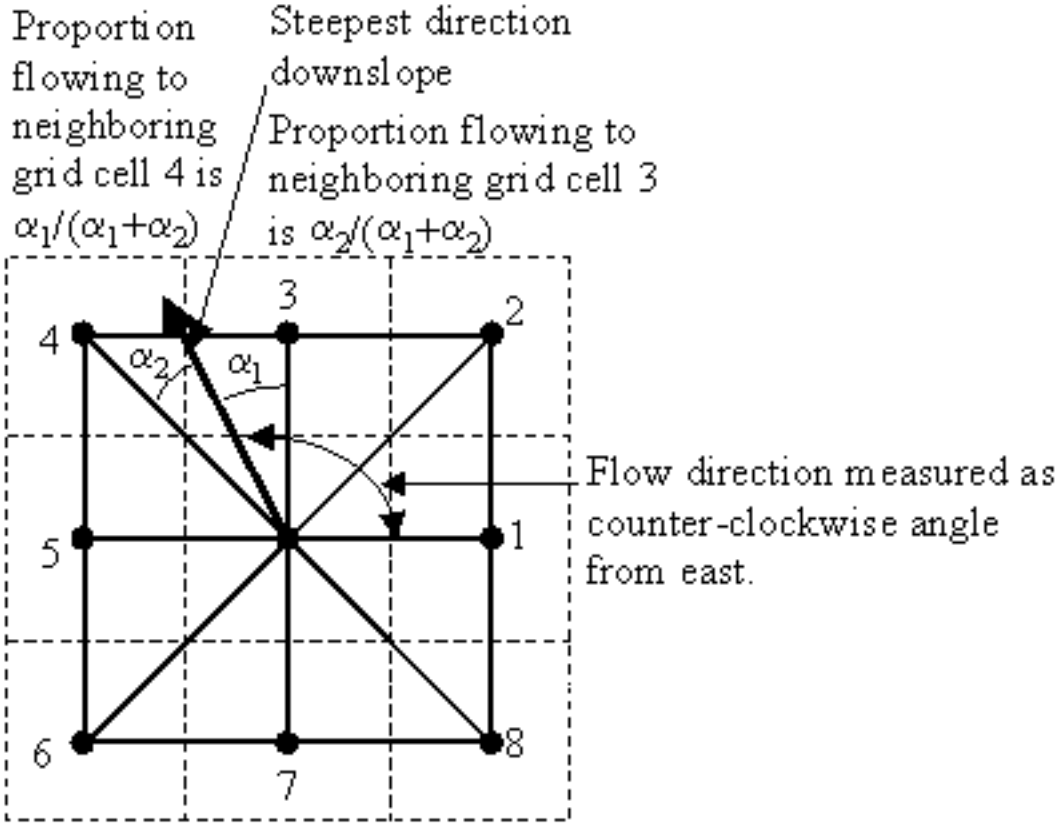
참고: 공간 처리 패키지는 인터페이스 설명만 담고 있기 때문에, 사용자가 TauDEM 5.0.6 을 직접 설치하고 공간 처리 워크프레임을 알맞게 환경 설정해야 한다는 사실을 기억하십시오.

TauDEM 알고리즘에 관한 문서는 공식 TauDEM 문서 에서 파생됐습니다.

24.4.1 기본 그리드 분석

D-Infinity 기여 영역

이 알고리즘은 다중 유수 방향 D-Infinity 접근법을 사용해서 등고선 길이 단위당 기여 영역을 나타내는 특정 집수 영역의 그리드를 계산합니다. D-Infinity 유수 방향은 직육면체로 이루어진 그리드 상에 있는 편평 삼각면 (planar triangular facet) 에 대해 가장 내리막 경사가 심한 방향으로 정의됩니다. 각 그리드 셀 별 기여도는 그리드 셀 길이 (또는 부가적인 가중치 그리드를 사용하는 경우, 가중치 그리드에서 나온 값) 로 계산합니다. 이렇게 되면 각 그리드 셀 별 기여 영역은 각 셀 고유의 기여도에 D-Infinity 유수 모델에 따라 해당 셀로 어느 정도 배수되는 오르막 이웃들의 기여도를 합산해서 계산합니다. 각 셀에서 유수 방향은 각도가 기수 (0, π/2, π, 3π/2) 또는 서수 (π/4, 3π/4, 5π/4, 7π/4) 방향인 경우 단일 이웃으로 모두 배수됩니다. 또는 각도가 교각 (direct angle) 사이라면 인접 이웃 2 개 방향으로 배수됩니다. 후자의 경우 유수 방향 각도가 인접 이웃 셀에 대한 교각에 얼마나 가깝냐에 따라 유수의 비율이 정해집니다. 이 알고리즘이 사용하는 등고선 길이란 그리드 셀 크기를 말합니다. 특정 집수 영역의 산출 단위는 그리드 셀 크기와 동일한 길이 단위입니다.



부가적인 가중치 그리드를 사용하지 않는 경우, 셀 개수를 (셀 면적을 셀 길이로 나눈) 그리드 셀 길이와 곱한 값으로 나타낸 특정 집수 영역 (등고선 길이 단위 당 오르막 영역) 을 산출합니다. 이때 특정 집수 영역의 정의에 따라 그리드 셀 길이가 실질적인 등고선 길이라고 가정하며, 유수 방향에 따른 등고선 길이의 차이점을 구분하지 않습니다. 부가적인 가중치 그리드를 사용하는 경우, 어떤 크기 조정도 없이 가중치의 요약을 직접 산출합니다.

부가적인 유출구 포인트 shapefile 을 사용하는 경우, 유출구 셀과 해당 셀의 (D-infinity 유수 모델에 따른) 오르막 셀들이 있는 영역만 평가합니다.

기본적으로, 이 도구는 경계 오염을 확인합니다. 경계 오염이란 영역 바깥에 있는 그리드 셀을 계산에 넣지 않기 때문에 기여 영역 값이 저평가될 수도 있다는 가능성으로 정의됩니다. 경계 오염은 배수가 경계로부터 안 쪽으로 흐르거나 또는 표고값이 NODATA 인 영역에서 발생합니다. 이 알고리즘은 이를 인식하고 기여 영역에 대해 NODATA 를 보고합니다. 경계선에서 영역으로 들어오는 유수 경로를 따라 NODATA 값들이 쭉 나열되는 것을 흔히

볼 수 있습니다. 이는 기대대로의 효과로서, 해당 그리드 셀들의 기여 영역이 사용할 수 있는 데이터 영역 외부에 있는 지형에 의존하기 때문에 그 기여 영역을 알 수 없다는 사실을 나타냅니다. 사용자가 경계 오염이 문제가 되지 않는다는 사실을 알고 있거나 또는 이런 문제를 무시하고자 하는 경우, 예를 들면 유역 외곽선을 따라 DEM 을 잘라낸 경우라면 경계 오염 확인 옵션을 끌 수도 있습니다.

파라미터

라벨	명칭	유형	설명
D-infinity flow directions	DINF_FLOWDIR	[raster]	삼각면 (triangular facet) 의 가장 심한 경사도를 사용하는 D-infinity 유수법에 기반한 유수 방향 그리드입니다. 유수 방향은 3x3 블록 중심 그리드의 삼각면 8 개 가운데 가장 심한 내리막 경사의 방향으로 결정됩니다. 동쪽에서 반시계 방향으로 0 에서 2π 사이의 (부동소수점) 연속량인 라디안 단위 각도로 유수 방향을 인코딩합니다. 이렇게 하면 일반적으로 가장 심한 내리막 경사를 가진 삼각면을 정의하는 이웃 셀 2 개 사이의 비율로서 그리드에서 산출되는 유수를 해석합니다.
Outlets 부가적	OUTLETS	[vector: point]	관심 유출구 포인트를 정의하는 포인트 shapefile 입니다. 이 파일을 입력하는 경우, 해당 유출구 셀의 주변 오르막 셀들만 평가 영역 안에 있다고 간주합니다.
Weight grid 부가적	WEIGHT_GRID	[raster]	각 셀에 대해 유수 기여도를 지정하는 그리드입니다. 이 (가중치 또는 부하값으로 불리기도 하는) 기여도는 기여 영역 집적에 사용됩니다. 이 파일을 입력하지 않는 경우, 셀 개수를 (셀 면적을 셀 길이로 나눈) 그리드 셀 길이와 곱한 값으로 나타낸 특정 집수 영역 (등고선 길이 단위 당 오르막 영역) 을 산출합니다.

다음 페이지에 계속

표 24.217 – 이전 페이지에서 계속

라벨	명칭	유형	설명
Check for edge contamination	EDGE_CONTAMINATION	[boolean] 기본값: True	이 도구가 경계 오염을 확인해야 할지 여부를 나타내는 플래그입니다. 경계 오염이란 영역 바깥에 있는 그리드 셀을 계산에 넣지 않기 때문에 기여 영역 값이 저평가될 수도 있다는 가능성으로 정의됩니다. 경계 오염은 배수가 경계로부터 안 쪽으로 흐르거나 또는 표고값이 NODATA 인 영역에서 발생합니다. 이 알고리즘은 이를 인식하고 이런 셀에 대해 NODATA 를 보고합니다. 경계선에서 영역으로 들어오는 유수 경로를 따라 NODATA 값들이 쭉 나열되는 것을 흔히 볼 수 있습니다. 이는 기대대로의 효과로서, 해당 그리드 셀들의 기여 영역이 사용할 수 있는 데이터 영역 외부에 있는 지형에 의존하기 때문에 그 기여 영역을 알 수 없다는 사실을 나타냅니다. 사용자가 경계 오염이 문제가 되지 않는다는 사실을 알고 있거나 또는 이런 문제를 무시하고자 하는 경우, 예를 들면 유역 외곽선을 따라 DEM 을 잘라낸 경우라면 이 옵션을 끌 수도 있습니다.
D-infinity specific catchment area	DINF_CONTRIB_AREA	[raster] 기본값: [Save to temporary file]	산출 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
D-infinity specific catchment area	DINF_CONTRIB_AREA	[raster]	다중 유수 방향 D-infinity 접근법을 사용한 등고선 길이 단위 당 기여 영역인 특정 집수 영역 그리드입니다. 각 그리드 셀별 기여 영역은 각 셀 고유의 기여도에 D-infinity 유수 모델에 따라 해당 셀로 배수되는 오르막 이웃들의 기여도를 합산해서 계산합니다.

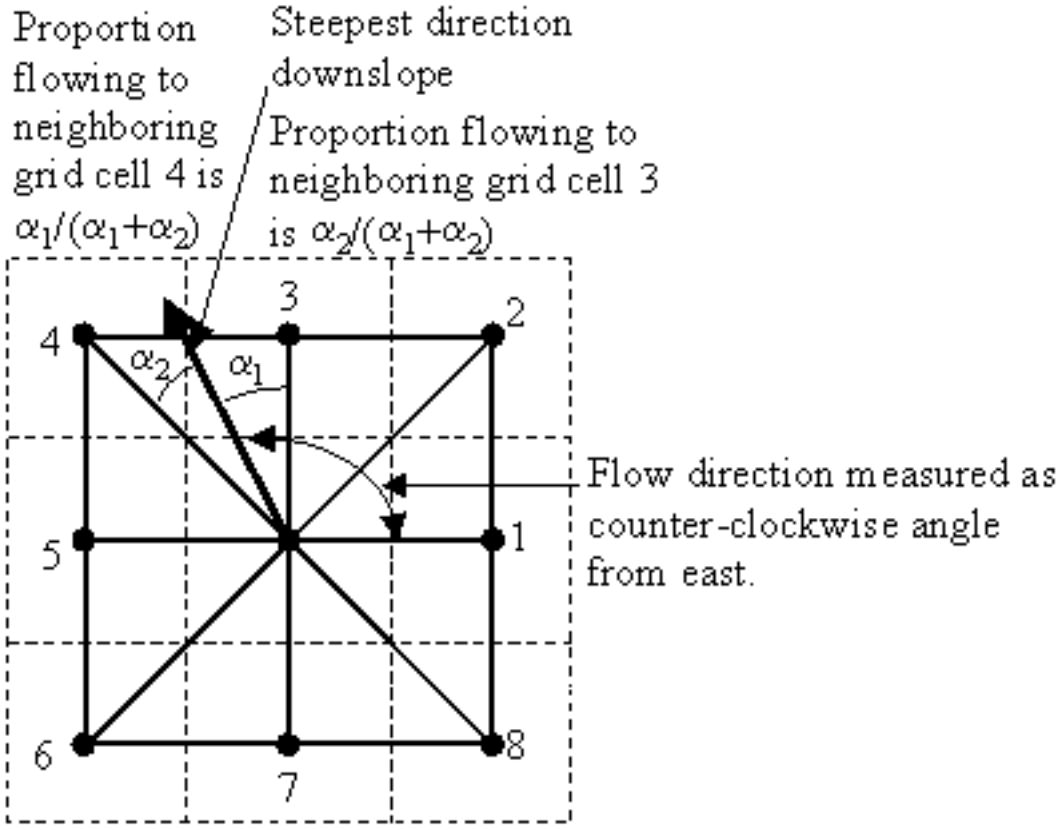
알고리즘 ID: `taudem:areadinf`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

D-infinity 유수 방향

삼각면 (triangular facet) 의 가장 심한 경사도를 사용하는 D-infinity 유수법에 기반한 유수 방향을 할당합니다 (Tarboton, 1997, "A New Method for the Determination of Flow Directions and Contributing Areas in Grid Digital Elevation Models", Water Resources Research, 33(2): 309-319). 유수 방향은 블록 중심 그리드 상의 삼각면 가운데 가장 심한 내리막 경사의 방향으로 정의됩니다. 동쪽에서 반시계 방향으로 0 에서 2π 사이의 (부동소수점) 연속량인 라디안 단위 각도로 유수 방향을 인코딩합니다. 유수 방향 각도는 관심 그리드 셀을 중심으로 하는 3x3 그리드 셀 창을 형성하는 삼각면 8 개 가운데 가장 심한 내리막 경사의 방향으로 정의됩니다. 이렇게 하면 일반적으로 가장 심한 내리막 경사를 가진 삼각면을 정의하는 이웃 셀 2 개 사이의 비율로서 그리드에서 산출되는 유수를 해석합니다.



해당 그리드 셀의 중심 표고를 표현하는 데 쓰이는 각 표고 값에 블록 중심 표현법을 사용합니다. 편평 삼각면 8 개가 각 그리드 셀과 그 이웃 8 개 사이에 형성됩니다. 각 편평 삼각면은 중심에서 바깥쪽으로 그리는 경우 중심 포인트에 있는 삼각면의 45 도 (π/4 라디안) 각도 범위 안 또는 바깥에 있을 수도 있는 내리막 벡터를 가지고 있습니다. 경사 벡터 각도가 삼각면 각도 안에 있을 경우, 해당 삼각면의 가장 심한 경사의 유수 방향을 나타냅니다. 경사 벡터 각도가 삼각면 각도 바깥에 있을 경우, 해당 삼각면의 가장 심한 경사의 유수 방향은 가장 심한 경사를 가진 경계를 따르게 됩니다. 그리드 셀의 경사 및 유수 방향은 모든 8 개 삼각면 가운데 가장 심한 내리막 벡터의 크기 및 방향을 따릅니다. 고저차/거리, 즉 경사각의 탄젠트로 경사를 측정합니다.

모든 경사 벡터가 양의 값 (내리막) 이 아닌 경우, 평탄 지역의 유수 방향을 결정하기 위한 Garbrecht & Martz (1997) 방법을 사용해서 유수 방향을 정합니다. 평탄 지역을 고지대에서 저지대로 향하는 배수 경로의 일부로 취급하는 방법입니다. 기존 하천을 따라 배수를 강제하기 위한 유수 경로 그리드를 입력할 수도 있고, 하지 않을 수도 있습니다. 입력하는 경우, 유수 방향을 정하는 과정에서 표고를 우선합니다.

함몰부를 채우지 않은 DEM 에 D-infinity 유수 방향 알고리즘을 적용할 수도 있지만, 그렇게 하면 함몰부의 최저 포인트에서 D-infinity 유수 방향 및 경사를 NODATA 로 산출할 것입니다.

파라미터

라벨	명칭	유형	설명
Pit filled elevation	`PIT_FILLED`	[raster]	표고 값을 보유한 그리드를 지정합니다. 일반적으로 이 그리드는 함몰부 제거 도구의 산출물로, 함몰부를 제거한 표고입니다. 함몰부란 수치 표고 모델 (DEM) 에 있는 높은 지형으로 완전히 둘러쌓인 낮은 표고 영역을 말합니다. 일반적으로 함몰부는 DEM 에서 유수 공간 처리를 방해하는, 디지타이즈 작업 과정 중의 오류로 간주됩니다. 따라서 함몰부의 표고를 해당 영역의 물이 배수될 때까지 올려 함몰부를 제거합니다. 사용자 DEM 에 있는 함몰부가 실재한다고 믿을 만한 이유가 있다면 이 단계를 건너뛰어도 됩니다. 몇몇 함몰부가 실제로 존재하고 제거해서는 안 되는 반면 다른 몇몇 함몰부는 제거해야 할 오류인 경우, 실재 함몰부의 가장 낮은 지점에 NODATA 표고값을 삽입해야 합니다. NODATA 값은 유수 영역의 경계를 정의하는 역할을 하며, 함몰부 제거 도구는 유수가 경계를 향해 흘러가도록 표고를 높이기 때문에, 필요한 경우 이렇게 NODATA 값을 영역 내부에 설정해주면 함몰부가 제거되는 일을 막을 수 있습니다.
D-infinity flow directions	`DINF_FLOWDIR`	[raster] 기본값: `[Save to temporary file]`	산출 유수 방향 래스터를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.
D-infinity slope	`DINF_SLOPE`	[raster] 기본값: `[Save to temporary file]`	산출 경사도 래스터를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
D-infinity flow directions	DINF_FLOWDIR	[raster]	삼각면 (triangular facet) 의 가장 심한 경사도를 사용하는 D-infinity 유수법에 기반한 유수 방향 그리드입니다. 유수 방향은 3x3 블록 중심 그리드의 삼각면 8 개 가운데 가장 심한 내리막 경사의 방향으로 결정됩니다. 동쪽에서 반시계 방향으로 0 에서 2π 사이의 (부동소수점) 연속량인 라디안 단위 각도로 유수 방향을 인코딩합니다. 이렇게 하면 일반적으로 가장 심한 내리막 경사를 가진 삼각면을 정의하는 이웃 셀 2 개 사이의 비율로서 그리드에서 산출되는 유수를 해석합니다.
D-infinity slope	DINF_SLOPE	[raster]	D. G. 타보튼 (Tarboton) 이 1997 년 수자원 연구 (Water Resources Research) 33(2) 호 309 ~ 319 쪽에 발표한 논문"A New Method for the Determination of Flow Directions and Contributing Areas in Grid Digital Elevation Models"에서 설명한 D-infinity 방법을 사용해서 평가한 경사 그리드를 지정합니다. 각 그리드 셀을 중심으로 한 8 개의 삼각면 가운데 가장 심한 바깥 방향 경사를 경사 값으로 보유하고 있는 그리드입니다. 고저차/거리, 즉 경사각의 탄젠트로 경사를 측정합니다.

알고리즘 ID: `taudem:dinfflowdir`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

D8 기여 영역

이 알고리즘은 단일 방향 D8 유수 모델을 사용해서 기여 영역의 그리드를 생성합니다. 각 그리드 셀 별로 기여도를 (또는 부가적인 가중치 그리드를 사용하는 경우, 가중치 그리드에서 나온 값으로) 계산합니다. 각 그리드 셀 별 기여 영역은 각 셀 고유의 기여도에 D8 유수 모델에 따라 해당 셀로 배수되는 오르막 이웃들의 기여도를 합산해서 계산합니다.

부가적인 유출구 포인트 shapefile 을 사용하는 경우, 유출구 셀과 해당 셀의 (D8 유수 모델에 따른) 오르막 셀들이 있는 영역만 평가합니다.

기본적으로, 이 도구는 경계 오염을 확인합니다. 경계 오염이란 영역 바깥에 있는 그리드 셀을 계산에 넣지 않기 때문에 기여 영역 값이 저평가될 수도 있다는 가능성으로 정의됩니다. 경계 오염은 배수가 경계로부터 안 쪽으로 흐르거나 또는 표고값이 NODATA 인 영역에서 발생합니다. 이 알고리즘은 이를 인식하고 기여 영역에 대해 NODATA 를 보고합니다. 경계선에서 영역으로 들어오는 유수 경로를 따라 NODATA 값들이 쭉 나열되는 것을 흔히 볼 수 있습니다. 이는 기대대로의 효과로서, 해당 그리드 셀들의 기여 영역이 사용할 수 있는 데이터 영역 외부에 있는 지형에 의존하기 때문에 그 기여 영역을 알 수 없다는 사실을 나타냅니다. 사용자가 경계 오염이 문제가 되지 않는다는 사실을 알고 있거나 또는 이런 문제를 무시하고자 하는 경우, 예를 들면 유역 외곽선을 따라 DEM 을 잘라낸 경우라면 경계 오염 확인 옵션을 끌 수도 있습니다.

파라미터

라벨	명칭	유형	설명
D8 flow directions	`D8_FLOWDIR`	[raster]	각 셀 별로 그 인접 또는 대각선 방향의 이웃 8 개 셀 가운데 가장 심한 내리막 경사를 가지고 있는 셀의 방향으로 정의한 D8 유수 방향 그리드입니다. 이 그리드는*D8 유수 방향* 알고리즘의 산출물로 얻을 수 있습니다.
Outlets 부가적	`OUTLETS`	[vector: point]	관심 유출구 포인트를 정의하는 포인트 shapefile 입니다. 이 파일을 입력하는 경우, 해당 유출구 셀의 주변 오르막 셀들만 평가 영역 안에 있다고 간주합니다.
Weight grid 부가적	`WEIGHT_GRID`	[raster]	각 셀에 대해 유수 기여도를 지정하는 그리드입니다. 이 (가중치 또는 부하값으로 불리기도 하는) 기여도는 기여 영역 집적에 사용됩니다. 이 파일을 입력하지 않는 경우, 각 그리드 셀 별로 유수 기여도를 추정할 것입니다.
Check for edge contamination	`EDGE_CONTAMINATION`	[boolean] 기본값: True	이 도구가 경계 오염을 확인해야 할지 여부를 나타내는 플래그입니다. 경계 오염이란 영역 바깥에 있는 그리드 셀을 계산에 넣지 않기 때문에 기여 영역 값이 저평가될 수도 있다는 가능성으로 정의됩니다. 경계 오염은 배수가 경계로부터 안 쪽으로 흐르거나 또는 표고값이 NODATA 인 영역에서 발생합니다. 이 알고리즘은 이를 인식하고 이런 셀에 대해 NODATA 를 보고합니다. 경계선에서 영역으로 들어오는 유수 경로를 따라 NODATA 값들이 쭉 나열되는 것을 흔히 볼 수 있습니다. 이는 기대대로의 효과로서, 해당 그리드 셀들의 기여 영역이 사용할 수 있는 데이터 영역 외부에 있는 지형에 의존하기 때문에 그 기여 영역을 알 수 없다는 사실을 나타냅니다. 사용자가 경계 오염이 문제가 되지 않는다는 사실을 알고 있거나 또는 이런 문제를 무시하고자 하는 경우, 예를 들면 유역 외곽선을 따라 DEM 을 잘라낸 경우라면 이 옵션을 끌 수도 있습니다.
D8 specific catchment area	`D8_CONTRIB_AREA`	[raster] 기본값: `[Save to temporary file]`	산출 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
D8 specific catchment area	D8_CONTRIB_AREA	[raster]	각 셀 고유의 기여도에 D8 유수 모델에 따라 해당 셀로 배수되는 오르막 이웃들의 기여도를 합산해서 계산한 기여 영역 값들의 그리드입니다.

알고리즘 ID: `taudem:aread8`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

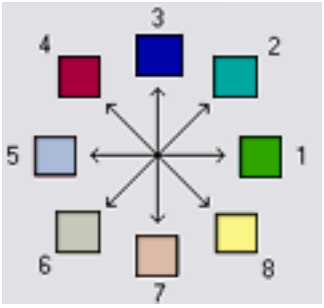
공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

D8 유수 방향

이 알고리즘은 그리드를 2 개 생성합니다. 첫 번째 그리드는 각 그리드 셀에서 가장 심한 내리막 경사를 계산해서 인접 또는 대각선 방향의 이웃들 가운데 어느 방향으로 흘러가는지에 대한 정보를 담고 있습니다. 두 번째 그리드는 가장 심한 내리막 경사의 방향으로 평가한 경사도를 고저차/거리라는, 예를 들면 각도의 탄젠트 값으로 담고 있습니다. DEM 영역의 경계에 인접한, 또는 DEM 의 NODATA 값에 인접한 모든 그리드 셀의 유수 방향은 NODATA 로 보고됩니다. 평탄 지역의 경우, Garbrecht & Martz (1997) 방법을 사용해서 고지대에서 저지대를 향해 유수 방향을 할당합니다. 함몰부를 채우지 않은 DEM 에 D8 유수 방향 알고리즘을 적용할 수도 있지만, 각 함몰부의 최저점에서 유수 방향 및 경사도를 NODATA 로 산출할 것입니다.

D8 유수 방향 코딩:

- 1 –동
- 2 –북동
- 3 –북
- 4 –북서
- 5 –서
- 6 –남서
- 7 –남
- 8 –남동



J. 가르브레히트와 L. W. 마츠가 1997 년 수문학 저널 (Journal of Hydrology) 193 호 204 ~ 213 쪽에 공저한 논문 "The Assignment of Drainage Direction Over Flat Surfaces in Raster Digital Elevation Models"에서 설명한 방법에 따라 유수가 평탄 지역에서 어느 방향으로 흘러가는지 결정합니다.

파라미터

라벨	명칭	유형	설명
Pit filled elevation	`PIT_FILLED`	[raster]	표고 값을 보유한 그리드를 지정합니다. 일반적으로 이 그리드는 함몰부 제거 도구의 산출물로, 함몰부를 제거한 표고입니다. 함몰부란 수치 표고 모델 (DEM) 에 있는 높은 지형으로 완전히 둘러쌓인 낮은 표고 영역을 말합니다. 일반적으로 함몰부는 DEM 에서 유수 공간 처리를 방해하는, 디지타이즈 작업 과정 중의 오류로 간주됩니다. 따라서 함몰부의 표고를 해당 영역의 물이 배수될 때까지 올려 함몰부를 제거합니다. 사용자 DEM 에 있는 함몰부가 실재한다고 믿을 만한 이유가 있다면 이 단계를 건너뛰어도 됩니다. 몇몇 함몰부가 실제로 존재하고 제거해서는 안 되는 반면 다른 몇몇 함몰부는 제거해야 할 오류인 경우, 실재 함몰부의 가장 낮은 지점에 NODATA 표고값을 삽입해야 합니다. NODATA 값은 유수 영역의 경계를 정의하는 역할을 하며, 함몰부 제거 도구는 유수가 경계를 향해 흘러가도록 표고를 높이기 때문에, 필요한 경우 이렇게 NODATA 값을 영역 내부에 설정해주면 함몰부가 제거되는 일을 막을 수 있습니다.
D8 flow directions	`D8_FLOWDIR`	[raster] 기본값: `[Save to temporary file]`	산출 유수 방향 래스터를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.
D8 slope	`D8_SLOPE`	[raster] 기본값: `[Save to temporary file]`	산출 경사도 래스터를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
D8 flow directions	`D8_FLOWDIR`	[raster]	각 셀 별로 그 인접 또는 대각선 방향의 이웃 8 개 셀 가운데 가장 심한 내리막 경사를 가지고 있는 셀의 방향으로 정의한 D8 유수 방향 그리드입니다.
D8 slope	`D8_SLOPE`	[raster]	D8 유수 방향에서 경사도를 지정하는 그리드입니다. 경사도는'고저차/거리'로 측정합니다.

알고리즘 ID: `taudem:d8flowdir`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

그리드 망

각 그리드 셀이 각각 1) 최장 경로, 2) 총 경로, 3) 스트랄러 순서 번호를 담고 있는 그리드 3 개를 생성합니다. 이 값들은 D8 유수 모델이 정의한 망 (network) 으로부터 파생됐습니다.

최장 오르막 길이란 각 셀로 배수되는 가장 멀리 있는 셀에서 시작한 유수 경로의 길이입니다. 총 오르막 경로 길이란 각 그리드 셀의 그리드 망 오르막 경로의 전체 길이입니다. 셀 크기 및 인접 방향인지 또는 대각선 방향인지를 고려해서 셀 중심 간의 길이를 측정합니다.

스트랄러 순서 (Strahler order) 란 다음과 같이 정의됩니다: D8 유수 방향 그리드가 유수 경로망을 정의합니다. 주 (source) 유수 경로가 스트랄러 순서 1 번입니다. 서로 다른 순서의 유수 경로 2 개가 하류 (downstream) 로 합쳐지는 경우 유수 경로의 순서는 가장 높은 순서를 따르게 됩니다. 동일한 순서의 유수 경로 2 개가 하류로 합쳐지는 경우 유수 경로 순서는 1 만큼 증가합니다. 2 개 이상의 유수 경로가 하류로 합쳐지는 경우 유수 경로 순서는 가장 높은 유수 경로 순서의 최대값, 또는 두 번째로 높은 유수 경로 순서에 1 을 더한 값으로 계산됩니다. 어느 한 포인트에서 2 개 이상의 유수 경로가 합쳐지는 경우 스트랄러 순서는 일반적으로 이렇게 정의됩니다.

부가적인 임계값 마스크 그리드를 입력하는 경우, 임계값 이상의 값을 가진 마스크 그리드 영역에 있는 그리드 셀만을 평가합니다. 해당 영역 안에 있는 어느 그리드 셀도 해당 셀로 배수되지 않는 셀을 주 (1 번 순서) 그리드 셀로 평가하고, 이런 유수 경로 2 개가 합쳐지는 경우에만 순서 규칙에 따라 순서를 누적 (propagate) 합니다. 또 임계값 이상의 값을 가진 마스크 그리드 영역에 있는 경로만 집계해서 길이를 평가합니다.

부가적인 유출구 포인트 shapefile 을 사용하는 경우, 유출구 셀과 해당 셀의 (D8 유수 모델에 따른) 오르막 셀들이 있는 영역만 평가합니다.

파라미터

라벨	명칭	유형	설명
D8 flow directions	`D8_FLOWDIR`	[raster]	각 셀 별로 그 인접 또는 대각선 방향의 이웃 8 개 셀 가운데 가장 심한 내리막 경사를 가지고 있는 셀의 방향으로 정의한 D8 유수 방향 그리드입니다. 이 그리드는*D8 유수 방향* 알고리즘의 산출물로 얻을 수 있습니다.
Mask Grid 부가적	`MASK_GRID`	[raster]	분석할 영역을 결정하는 데 사용하는 그리드를 지정합니다. 마스크 그리드의 값이 마스크 임계값 (다음 파라미터 참조) 이상인 경우, 해당 셀을 분석 영역에 포함시킵니다. 이 도구는 경계 오염 플래그를 보유하고 있지 않지만, 경계 오염 분석을 해야 하는 경우*D8 기여 영역* 처럼 경계 오염을 지원하는 알고리즘이 산출한 마스크 그리드를 사용하면 됩니다.
Mask threshold 부가적	`THRESHOLD`	[number] 기본값: 100.0	어느 그리드 셀이 분석할 영역에 해당하는지 결정하기 위해 마스크 그리드 값 >= 마스크 임계값 계산에 사용하는 입력 파라미터를 지정합니다.
Outlets 부가적	`OUTLETS`	[vector: point]	관심 유출구 포인트를 정의하는 포인트 shapefile 입니다. 이 파일을 입력하는 경우, 해당 유출구 셀의 주변 오르막 셀들만 평가 영역 안에 있다고 간주합니다.
Longest upslope length	`LONGEST_PATH`	[raster] 기본값: `[Save to temporary file]`	가장 긴 오르막 (upslope) 길이를 가질 산출 래스터를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.
Total upslope length	`TOTAL_PATH`	[raster] 기본값: `[Save to temporary file]`	총 오르막 (upslope) 길이를 가질 산출 래스터를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.
Strahler network order	`STRAHLER_ORDER`	[raster] 기본값: `[Save to temporary file]`	스트랄러 망 순서를 가질 산출 래스터를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Longest upslope length	LONGEST_PATH	[raster]	각 그리드 셀에서 끝나는 최장 오르막 D8 유수 경로의 길이를 지정하는 그리드입니다. 셀 크기 및 인접 방향인지 또는 대각선 방향인지를 고려해서 셀 중심 간의 길이를 측정합니다.
Total upslope length	TOTAL_PATH	[raster]	총 오르막 경로 길이란 각 그리드 셀의 D8 유수 그리드 망 오르막 경로의 전체 길이입니다. 셀 크기 및 인접 방향인지 또는 대각선 방향인지를 고려해서 셀 중심 간의 길이를 측정합니다.
Strahler network order	STRAHLER_ORDER	[raster]	각 셀 별 스트랄러 순서를 지정하는 그리드입니다. D8 유수 방향 그리드가 유수 경로망을 정의합니다. 주 (source) 유수 경로가 스트랄러 순서 1 번입니다. 서로 다른 순서의 유수 경로 2 개가 하류 (downstream) 로 합쳐지는 경우 유수 경로의 순서는 가장 높은 순서를 따르게 됩니다. 동일한 순서의 유수 경로 2 개가 하류로 합쳐지는 경우 유수 경로 순서는 1 만큼 증가합니다. 2 개 이상의 유수 경로가 하류로 합쳐지는 경우 유수 경로 순서는 가장 높은 유수 경로 순서의 최대값, 또는 두 번째로 높은 유수 경로 순서에 1 을 더한 값으로 계산됩니다. 어느 한 포인트에서 2 개 이상의 유수 경로가 합쳐지는 경우 스트랄러 순서는 일반적으로 이렇게 정의됩니다.

알고리즘 ID: `taudem:gridnet`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

함몰부 제거

DEM 에 있는 모든 함몰부를 식별해서, 표고를 그 경계 주변의 최저 배수 포인트 수준까지 올립니다. 함몰부란 수치 표고 모델 (DEM) 에 있는, 높은 지형으로 완전히 둘러쌓인 낮은 표고 영역을 말합니다. 일반적으로 함몰부는 DEM 에서 유수 경로 작업을 방해하는 오류로 간주됩니다. 따라서 함몰부의 표고를 해당 영역의 물이 배수될 때까지 올려 함몰부를 제거합니다. 배수 포인트 (pour point) 란 함몰부로 배수되는"유역 (watershed)"경계 상의 최저 포인트를 말합니다. 사용자 DEM 에 있는 함몰부가 실재한다고 믿을 만한 이유가 있다면 이 단계를 건너뛰어도 됩니다. 몇몇 함몰부가 실제로 존재하고 제거해서는 안 되는 반면 다른 몇몇 함몰부는 제거해야 할 오류인 경우, 실재 함몰부의 가장 낮은 지점에 NODATA 표고값을 삽입해야 합니다. NODATA 값은 유수 영역의 경계를 정의하는 역할을 하며, 함몰부 제거 도구는 유수가 경계를 향해 흘러가도록 표고를 높이기 때문에, 필요한 경우 이렇게 NODATA 값을 영역 내부에 설정해주면 함몰부가 제거되는 일을 막을 수 있습니다.

파라미터

라벨	명칭	유형	설명
Elevation	`ELEVATION`	[raster]	지형 분석 및 하천 묘사를 위한 기본 입력물 역할을 하는 DEM 그리드를 지정합니다.
Depression mask 부가적	`DEPRESSION_MASK`	[raster]	
Consider only 4 way neighbors	`FOUR_NEIGHBOURS`	[boolean] 기본값: False	
Pit removed elevation	`PIT_FILLED`	[raster] 기본값: `[Save to temporary file]`	함몰부를 채운 산출 레이어를 지정합니다. 다음 가운데 하나로 저장할 수 있습니다: • 임시 파일로 저장 • 파일로 저장 이 파라미터에서 파일 인코딩도 변경할 수 있습니다.

산출물

라벨	명칭	유형	설명
Pit removed elevation	`PIT_FILLED`	[raster]	유수 경로가 영역 밖으로 향하도록 함몰부를 제거한 표고 값 그리드입니다.

알고리즘 ID: `taudem:pitremove`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

24.4.2 특수 그리드 분석

하천까지의 D8 거리

하천 그리드 셀을 만날 때까지 D8 유수 모델에 따라 내리막 방향으로 이동해서 각 그리드 셀 별 하천까지의 수평 거리를 계산합니다.

파라미터

라벨	명칭	유형	설명
D8 Flow Direction Grid		[raster]	이 입력물은 모든 셀에서의 유수가 전부 가장 심한 내리막 경사 방향에 있는 단일 이웃 셀로 흐르는 D8 방법을 사용해서 인코딩된 유수 방향 그리드입니다. 이 그리드는 D8 유수 방향 알고리즘의 산출물로 얻을 수 있습니다.
Stream Raster Grid		[raster]	하천을 나타내는 그리드를 지정합니다. 하천망 분석 도구 모음 가운데 몇몇 도구들로 이런 그리드를 생성할 수 있습니다. 하지만, **하천망 분석** 도구 모음은 하천이 아닌 셀인 경우 0, 하천 셀인 경우 1 값을 가지는 그리드만 생성할 수 있습니다. 이 알고리즘은 1 을 초과하는 값을 가진 그리드도 입력받을 수 있는데, `Threshold` 파라미터와 함께 사용하면 하천의 위치를 결정할 수 있습니다. 즉 하천을 정의하는 데 일반'하천 래스터 그리드'는 물론 기여 영역 그리드도 사용할 수 있다는 뜻입니다. 이 입력 그리드의 값은 (long) 정수형 값이어야 하며, 정수가 아닌 값일 경우 정수로 잘라낸 다음 평가할 것입니다.
Threshold		[number] 기본값: 50	이 값은 `Stream Raster Grid` 상에서 하천 위치를 결정하기 위한 한계값 역할을 합니다. `Threshold` 값 이상의 `Stream Raster Grid` 값을 보유한 셀을 하천으로 해석합니다.

산출물

라벨	명칭	유형	설명
Output Distance to Streams		[raster]	이 산출 그리드는 하천 래스터 그리드에 있는 하천에 대해 D8 유수 방향 그리드가 정의하는 유수 방향을 따르는 수평 거리를 지정합니다.

파이썬 코드

알고리즘 ID: `taudem:d8hdisttostrm`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

D-Infinity 산사태 유출

이 알고리즘은 산사태의 영향을 받은 영역과 해당 영역에 있는 각 셀 별 유수 경로 길이를 식별합니다. 소스 영역에서 영향 영역까지의 경사가 알파 각도 (alpha angle) 라는 한계값 각도 미만인 포인트 까지인 각 소스 영역 셀 주변의 모든 내리막 셀들이 영향 영역에 들어갈 수 있습니다. 이 도구는 유수 방향을 결정하기 위해 D-infinity 다중 유수 방향 메소드를 사용합니다. 이 도구는 유수가 영향 영역을 과장할 수도 있는 일부 내리막 셀들로 극소량 분산되도록 할 가능성이 크기 때문에, 이런 과장된 분산을 피하기 위해 한계값 비율을 설정할 수 있습니다. 여기서 유수 경로 길이란 해당 셀에서 가장 급한 경사를 보유한 소스 셀까지의 거리를 말합니다.

소스 영역 주변의 모든 내리막 포인트는 잠재적으로 영향 영역에 들어갈 수 있지만, 소스 영역에서 영향 영역까지의 경사가 알파 각도라는 한계값 각도 미만인 포인트를 넘어서는 포인트는 아닙니다.

Elevations

10	10	10	10	10	10	Yellow cell is the source
10	9	9	9	9	10	Green: downslope of source
10	9	8	7	6.99	10	
10	9	9	8	6.98	10	
10	9	8	7	6.97	10	
10	10	10	10	6.96	10	

Straight-line distance from highest point of source

0	1	2	3	4	5	Yellow cell is the source
1	1.414214	2.236068	3.162278	4.123106	5.09902	Green: downslope of source
2	2.236068	2.828427	3.605551	4.472136	5.385165	
3	3.162278	3.605551	4.242641	5	5.830952	
4	4.123106	4.472136	5	5.656854	6.403124	
5	5.09902	5.385165	5.830952	6.403124	7.071068	

Drop in elevation from highest point in source

0	0	0	0	0	0	Yellow cell is the source
0	1	1	1	1	0	Green: downslope of source
0	1	2	3	3.01	0	
0	1	1	2	3.02	0	
0	1	2	3	3.03	0	
0	0	0	0	3.04	0	

2 The cell size (a fiddle factor for me to make sensible values)
18 The threshold angle for being in the runout zone

The slope angle from the highest point in the source to each cell

0	0	0	0	0	0	Yellow cell is the source
0	19	13	9	7	0	Green: downslope of source
0	13	19	23	19	0	Grey cells are BOTH
0	9	8	13	17	0	downslope of the source AND
0	7	13	17	15	0	have a sufficiently steep
0	0	0	0	13	0	angle to be in the runout zone

소스 포인트에서 평가 포인트까지의 직선 거리를 사용해서 경사를 측정합니다.

그런데 유수 경로를 따라 각도를 측정하는 것이 물리적으로 더 말이 됩니다. 유수 경로를 따라 각도를 측정하도록 코딩하는 일이 직선 각도 측정 코딩만큼이나 쉽기 때문에, 이 두 방법을 전환할 수 있는 옵션을 제공할 것입니다. 산사태 유출을 평가할 수 있는 가장 현실적인 방법은 각 포인트에 가장 급한 각도를 가진 소스 포인트를 추적하는 것입니다. 이렇게 하면 회귀적 오르막 유수 대수 (algebra) 접근법으로 어느 그리드 셀 및 해당 셀로 흐르는 모든 오르막 이웃들을 찾을 것입니다. 해당 그리드 셀에 대한 각도를 계산해서 그 각도가 알파 각도를 초과하는 경우 해당 셀을 유출 구역으로 포함시키는 데 이 오르막 이웃들의 정보를 사용할 것입니다. 이 과정에서 어느 그리드

셀의 최대 각도는 해당 셀로 흘러드는 이웃들에 대한 최대 각도를 보유한 셀 집합에서 나올 것이라고 가정합니다. 유수 경로를 따라 각도를 계산하는 경우 이 가정은 언제나 참이지만, 유수 경로가 스스로에 대해 역전할 때 직선을 따라 각도를 계산하는 경우 이 가정이 거짓이 될 가능성도 존재합니다.

D-infinity 다중 유수 방향 필드는 각 그리드 셀로부터 흘러나오는 모든 유수에 대해 0 ~ 1 사이에서 변화하며, 총합이 1 이 되는 비율 (`Pik`) 을 사용해서 다중 내리막 이웃들로의 유수를 할당합니다. 어느 그리드 셀을 유수가 흘러드는 내리막 그리드 셀로 집계하기 전에 이 비율이 초과해야 하는 한계값 `T` 를 지정하는 것이 바람직할 수도 있습니다. 예를 들어 극소량의 유수가 흘러드는 그리드 셀로의 분산을 피하기 위해 `Pik > T` (한계값 0.2) 같은 조건을 설정할 수 있습니다. 사용자가 `T` 값을 입력해서 설정해야 합니다. 모든 오르막 그리드 셀이 `T` 를 사용해야 하는 경우 0 으로 입력할 수도 있습니다.

산사태 소스 위치는 양의 값이 산사태가 유발될 수도 있는 곳을 그리고 0 값이 아닌 곳을 나타내도록 구성된 (파일명의 접미어가 `*ass` 인, 예를 들면 `demass` 인) (short) 정수형 그리드로 입력해야 합니다.

다음 그리드들을 산출합니다:

- rz −0 값이 해당 그리드 셀이 유출 구역에 포함되지 않으며 0 이상의 값이 해당 그리드 셀은 유출 구역에 포함된다는 사실을 나타내는 유출 구역 (runout zone) 지표입니다. 관련 산사태 소스 위치에 대한 각도에 정보가 존재할 수도 있기 때문에, 소스 위치에 대한 각도에 이 변수를 (도 단위로) 할당할 것입니다.
- dm −해당 포인트에 대한 가장 급한 각도를 보유한 소스 위치로부터 유수를 따라 계산한 거리입니다.

파라미터

라벨	명칭	유형	설명
D-Infinity Flow Direction Grid		[raster]	D-infinity 방법으로 지정한 유수 방향 그리드입니다. 유수 방향을 라디안 단위로, 동쪽에서 반시계방향으로 측정합니다. D-Infinity 유수 방향 알고리즘이 이 그리드를 생성할 수 있습니다.
Pit Filled Elevation Grid		[raster]	표고 값 그리드를 입력합니다. 일반 규칙에 따라, 구덩이를 제거한 표고 값 그리드를 입력할 것을 권합니다. 구덩이란 일반적으로 DEM 상의 유수 분석을 방해하는 오류로 간주됩니다. 구덩이 제거 도구가 산출한 그리드를 사용하면 됩니다. 구덩이의 표고를 해당 영역의 물이 배수될 때까지 올려 구덩이를 제거한 표고 값을 담고 있는 그리드입니다.
Avalanche Source Site Grid		[raster]	경험 및 지도 해석을 통해 공통적으로 직접 식별한 눈사태 소스 영역 그리드입니다. 눈사태 소스 위치는 양의 값이 눈사태가 유발될 수도 있는 곳을 그리고 0 값이 아닌 곳을 나타내도록 구성된 (파일명의 접미어가 `*ass` 인, 예를 들면 `demass` 인) (short) 정수형 그리드로 입력해야 합니다.
Proportion Threshold		[number] 기본값: 0.2	이 값은 유수 방향을 결정하기 위한 D-infinity 다중 유수 방향 메소드를 사용함으로써 발생하는 유수의 분산을 제약하는 데 쓰이는 한계값 비율입니다. D-infinity 다중 유수 방향 메소드는 종종 영향 영역을 과장할 수도 있는 일부 내리막 셀들로 유수를 극소량 분산시키기 하기 때문에, 이런 과장된 분산을 피하기 위해 한계값 비율을 설정할 수 있습니다.
Alpha Angle Threshold		[number] 기본값: 18	이 값은 알파 각도라는 한계값 각도로 소스 셀 주변의 어느 내리막 셀이 영향 영역에 들어가는지 결정하는 데 사용됩니다. 소스 영역에서 영향 영역까지의 경사가 한계값 각도 미만인 포인트까지인 각 소스 영역 셀 주변의 내리막 셀들만 영향 영역에 포함됩니다.
Measure distance along flow path		[boolean] 기본값: True	이 옵션에서 경사각을 계산하는 데 쓰이는 거리를 측정할 방법을 선택합니다. 이 옵션이 참 인 경우 유수 경로를 따라 측정하고, 거짓 인 경우 소스 셀에서 평가 셀까지의 직선을 따라 거리를 측정합니다.

산출물

라벨	명칭	유형	설명
Runout Zone Grid		[raster]	이 그리드는 0 값이 해당 그리드 셀이 유출 구역에 포함되지 않으며 0 이상의 값이 해당 그리드 셀은 유출 구역에 포함된다는 사실을 나타내는 유출 구역 지표를 사용해서 산사태 유출 구역 (영향 영역) 을 식별합니다. 관련 산사태 소스 위치에 대한 각도에 정보가 존재할 수도 있기 때문에, 소스 위치에 대한 각도에 이 변수를 (도 단위로) 할당할 것입니다.
Path Distance Grid		[raster]	이 그리드는 각 셀에 대한 가장 급한 각도를 보유한 소스 위치로부터 유수를 따라 계산한 거리를 담고 있습니다.

파이썬 코드

알고리즘 ID: `taudem:dinfavalanche`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

D-Infinity 농도 제약 집적

이 알고리즘은 표시자 (indicator) 그리드 (gd) 가 표시하는 지역에 걸쳐 어떤 물질이 농도 (concentration) 또는 용해도 (solubility) 한계값 Csol 로 유수에 무제한 공급되는 상황에 적용됩니다. 공급 영역에 있는 어떤 물질이 유수에 농도 또는 용해도 한계값으로 흘러드는 경우, 영역 내 각 위치의 해당 물질 농도 그리드를 산출합니다. 먼저 유수를 입력 유효 유출 가중치 그리드 (Effective Runoff Weight Grid)(개념상 과잉 침전) 의 D-infinity 가중치 기여 영역으로 계산합니다. 공급 영역 (표시자 그리드) 전체의 물질 농도는 농도 한계값입니다. 해당 물질이 D-infinity 유수 방향으로 내리막 이동하는 동안, 유수 변화에 따라 희석되는 것은 물론 셀에서 셀로 이동하면서 부분적으로 (1 차) 감퇴 (decay) 하게 됩니다. 감퇴 곱수 그리드 (decay multiplier grid) 는 그리드 셀 `x` 에서 다음 내리막 셀로 이동하면서 부분적으로 (1 차) 감소되는 양을 지정합니다. 배출 shapefile 을 사용하는 경우, 이 도구는 shapefile 이 지정하는 위치에 유수를 기여하는 영역 부분만을 평가합니다. 어느 화합물이 어느 구역에 걸쳐 유수에 농도 또는 용해도 한계값으로 흘러들고 해당 구역에서 나오는 유수가 감퇴 (decay) 또는 감쇠 (attenuation) 될 수도 있는 경우, 이 알고리즘은 해당 화합물을 무제한 공급할 수 있는 지역에서 오염 물질 또는 해당 화합물을 추적하는 데 쓸모가 있습니다.

(0, 1) 표시자 함수 `i(x)` 를 사용해서 물질 공급 영역을 묘사하기 위해 표시자 그리드 (`gd`) 를 사용합니다. `A[]` 는 D-Infinity 기여 영역 함수를 사용해서 평가하는 가중 집적 연산자를 나타냅니다. `Effective Runoff Weight Grid` (실질 유출 가중치 그리드) 는 `w(x)` 로 나타내는 유수로의 공급량 (예: 지표 유수인 경우 과잉 강수량) 을 지정합니다. 특정 배출량은 다음과 같이 지정합니다:

```
Q(x)=A[w(x)]
```

이 가중 집적 `Q(x)` 는 `Overland Flow Specific Discharge Grid` (지표 유수 특정 배출 그리드) 로 산출됩니다. 물질 공급 영역 전체의 농도는 한계값입니다. (여기서 한계값이란 포화도 또는 한계 용해도입니다.) 만약 `i(x) = 1` 인 경우:

```
C(x) = Csol, and L(x) = Csol Q(x),
```

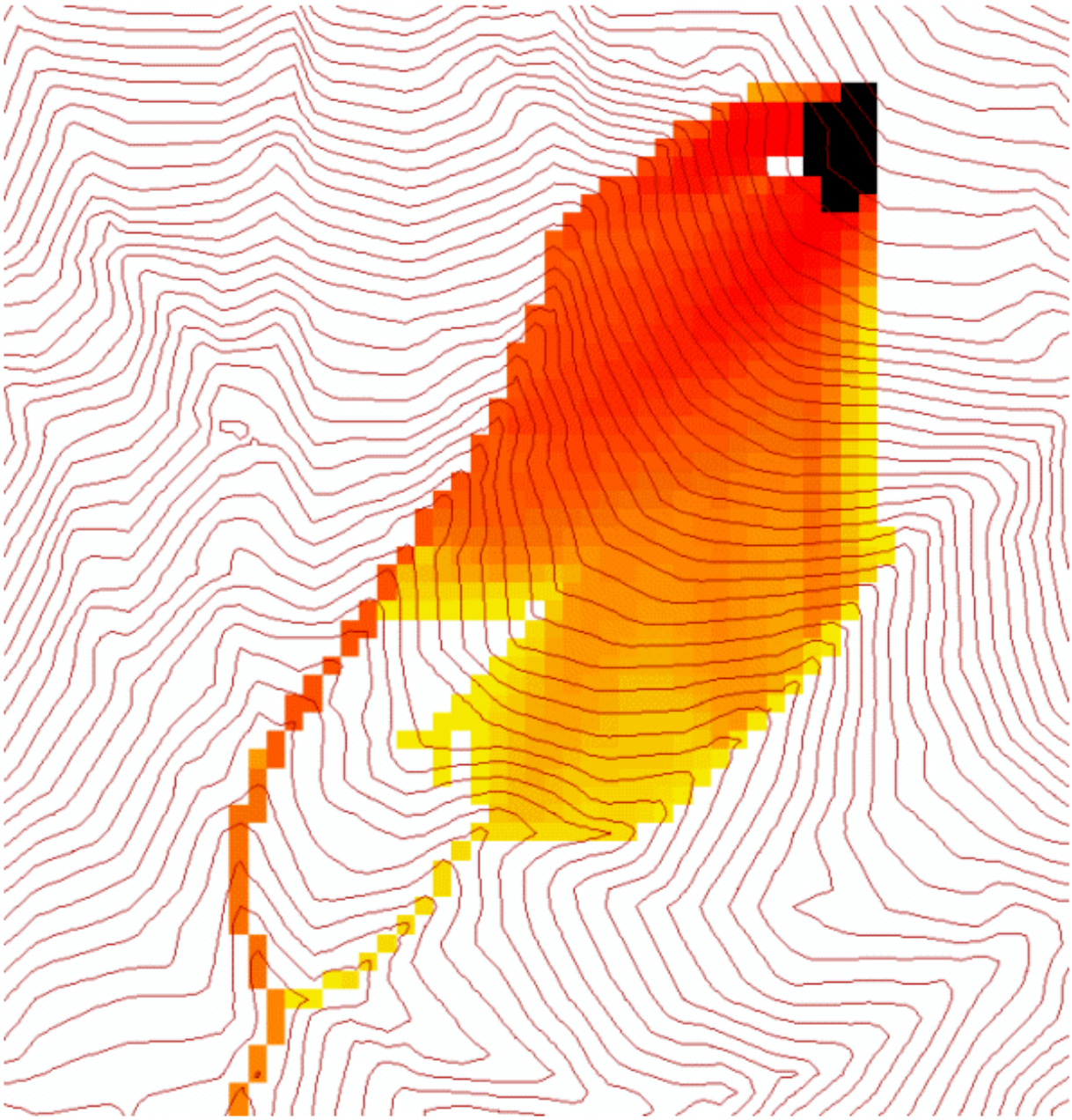
이때 `L(x)` 는 유수가 운반하고 있는 질량을 나타냅니다. 나머지 위치들의 경우, 질량 집적 및 희석에 의한 농도를 통해 질량을 다음과 같이 결정합니다:

$$L(x) = L(i, j) = \sum_{k \text{ contributing neighbors}} p_k d(i_k, j_k) L(i_k, j_k)$$

$$C(x) = L(x)/Q(x)$$

여기서 `d(x) = d(i, j)` 가 그리드 셀 `x` 에서 다음 내리막 셀로 이동하면서 부분적으로 (1 차) 감소되는 양을 지정하는 감퇴 곱수입니다. 이동 (또는 체류) 를 셀 간의 유수와 연관된 `t(x)` 와 곱할 수 있는 경우, `d(x)` 를 `exp(-k t(x))` 로 나타낼 수도 있습니다. 이때 `k` 가 1 차 감퇴 매개 변수입니다. 농도 그리드의 산출물이

`C(x)` 입니다. 배출 shapefile 을 사용하는 경우, 이 알고리즘은 shapefile 이 지정하는 위치에 유수를 기여하는 영역 부분만을 평가합니다.



고정 한계값 농도에서 유수로 배출된 또는 분할된 오염 물질을 추적하는 데 유용합니다.

파라미터

라벨	명칭	유형	설명
D-Infinity Flow Direction Grid		[raster]	D-infinity 방법으로 유수 방향을 지정한 그리드입니다. 유수 방향을 라디안 단위로, 동쪽에서 반시계방향으로 측정합니다. D-Infinity 유수 방향 알고리즘이 이 그리드를 생성할 수 있습니다.
Disturbance Indicator Grid		[raster]	물질 공급 영역인 소스 구역을 나타내는 그리드로, 해당 구역 내부는 1, 영역의 나머지 부분은 전부 0 또는 NODATA 여야만 합니다.
Decay Multiplier Grid		[raster]	내리막 그리드 셀에 집적되기 전 각 그리드 셀을 나가는 유수에 곱할 인수를 지정하는 그리드입니다. 감쇠 또는 감퇴되는 물질의 이동을 시뮬레이션하는 데 사용할 수도 있습니다. 이동 (또는 체류) 를 셀 간의 유수와 연관된 `t(x)` 와 곱할 수 있는 경우, `d(x)` 를 `exp(-k t(x))` 로 나타낼 수도 있습니다. 이때 `k` 가 1 차 감퇴 매개 변수입니다.
Effective Runoff Weight Grid		[raster]	D-infinity 가중 기여 영역이 `Overland Flow Specific Discharge Grid` 를 평가할 때 쓰이는 입력량 (개념상 실질 유출량 또는 과잉 침전량) 을 지정하는 그리드입니다.
Outlets shapefile 부가적		[vector: point]	이 부가적인 입력 파일은 관심 배출 포인트를 정의하는 포인트 shapefile 입니다. 이 파일을 입력하는 경우, 해당 배출 포인트의 오르막 영역만을 평가할 것입니다.
Concentration Threshold		[number] 기본값: 1.0	농도 또는 용해도 한계값입니다. 물질 공급 영역 전체의 농도가 이 한계값입니다.
Check for edge contamination		[boolean] 기본값: True	이 옵션으로 경계 오염을 확인해야 할지 여부를 결정합니다. 경계 오염이란 기여 영역 결정 시 영역 외부에 있는 그리드 셀을 고려하지 않기 때문에 값이 과소평가될 수도 있다는 가능성으로 정의됩니다.

산출물

라벨	명칭	유형	설명
Concentration Grid		[raster]	유수 속 관심 화합물의 농도를 산출해서 지정하는 그리드입니다.

파이썬 코드

알고리즘 ID: `taudem:dinfconclimaccum`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

D-Infinity 감퇴 집적

D-Infinity 감퇴 집적 알고리즘은 어느 물질의 양이 D-Infinity 유수 방향으로 집적되지만, 셀에서 셀로 이동하면서 1 차 감퇴되는 영역 내부에 있는 각 위치에 집적되는 양의 그리드를 생성합니다. 기본적으로, 각 그리드 셀의 양 분포는 너비 단위 당 집적량을 지정하는 셀 길이이지만, 부가적으로 가중치 그리드로 표현할 수도 있습니다. 감퇴 곱수 그리드는 그리드 셀 `x` 에서 다음 내리막 셀로 이동하면서 부분적으로 (1 차) 감소되는 양을 지정합니다.

감퇴 집적 연산자 `DA[.]` 는 각 그리드 위치에서 유수 방향으로 이동하지만 셀에서 셀로 이동하면서 1 차 감퇴된다고 가정하는 `m(i, j)` 로 표현되는 질량 하중 항목 `m(x)` 를 입력받아, 각 위치 `DA(x)` 에서 집적된 질량을 산출합니다. 각 그리드 셀에서 `m` 의 집적량을 숫자로 평가할 수 있습니다.

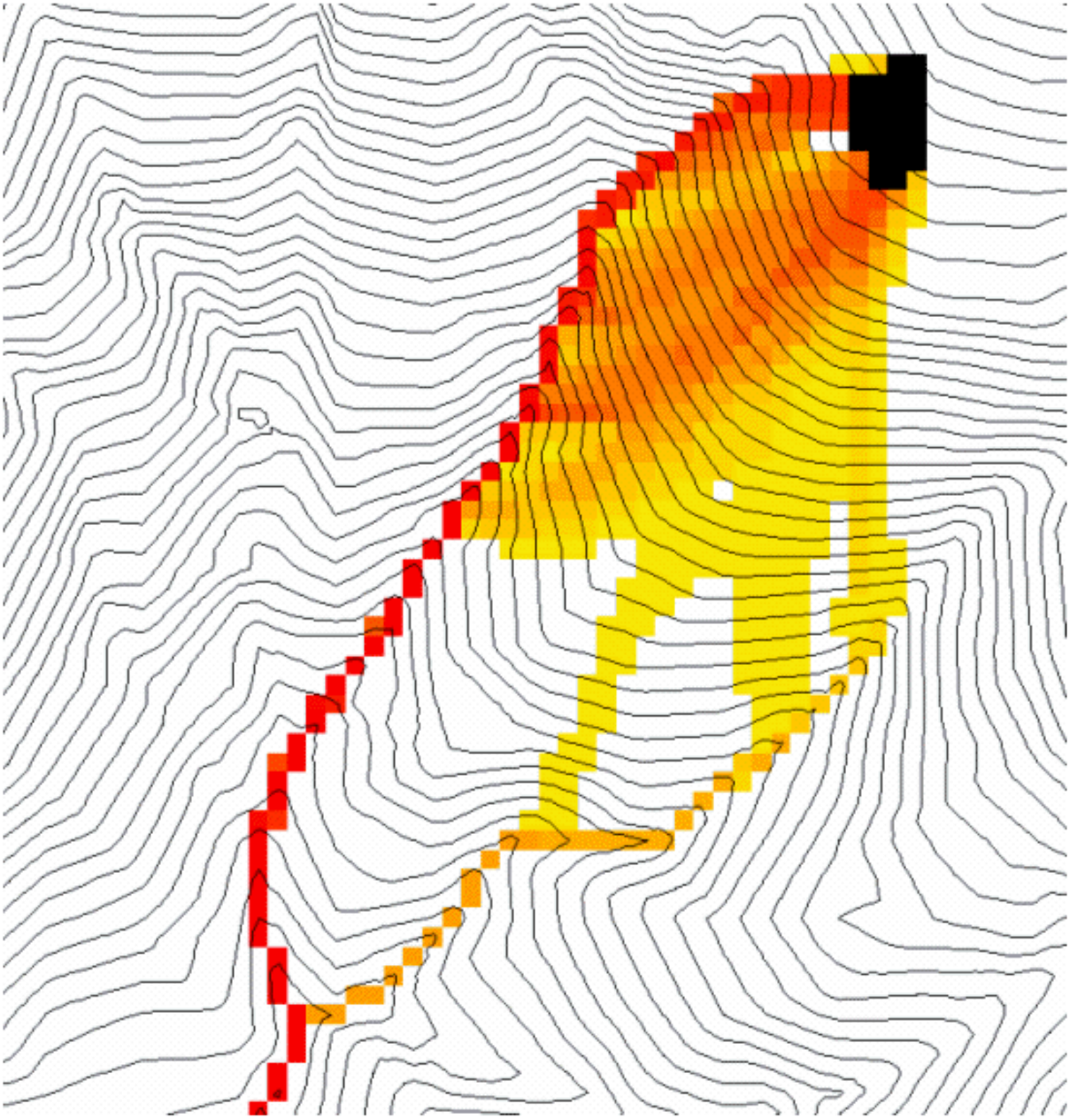
$$DA[m(x)] = DA(i, j) = m(i, j)\,\Delta^2 + \sum_{k \text{ contributing neighbors}} p_k d(i_k, j_k) DA(i_k, j_k)$$

여기서 `d(x) = d(i, j)` 가 그리드 셀 `x` 에서 다음 내리막 셀로 이동하면서 부분적으로 (1 차) 감소되는 양을 지정하는 감퇴 곱수입니다. 이동 (또는 체류) 를 셀 간의 유수와 연관된 `t(x)` 와 곱할 수 있는 경우, `d(x)` 를 `exp(-k t(x))` 로 나타낼 수도 있습니다. 이때 `k` 가 1 차 감퇴 매개 변수입니다. 질량 하중 `m(x)` 를 표현하기 위해 가중치 그리드를 사용합니다. 가중치를 따로 지정하지 않을 경우 1 로 간주합니다. 배출 shapefile 을 사용하는 경우, 이 알고리즘은 shapefile 이 지정하는 위치에 유수를 기여하는 영역 부분만을 평가합니다.

감퇴 또는 감쇠되는 오염 물질 또는 화합물을 추적하는 데 유용합니다.

파라미터

라벨	명칭	유형	설명
D-Infinity Flow Direction Grid		[raster]	D-infinity 방법으로 유수 방향을 지정한 그리드입니다. 유수 방향을 라디안 단위로, 동쪽에서 반시계방향으로 측정합니다. D-Infinity 유수 방향 알고리즘이 이 그리드를 생성할 수 있습니다.
Decay Multiplier Grid		[raster]	내리막 그리드 셀에 집적되기 전 각 그리드 셀을 나가는 유수에 곱할 인수를 지정하는 그리드입니다. 감쇠되는 물질의 이동을 시뮬레이션하는 데 사용할 수도 있습니다.
Weight Grid 부가적		[raster]	집적 계산에 사용되는 가중치 (하중) 를 지정하는 그리드입니다. 이 부가적인 그리드를 지정하지 않는 경우, 가중치를 너비 단위 당 집적을 지정하는 선형 그리드 셀 크기로 간주합니다.
Outlets Shapefile 부가적		[vector: point]	이 부가적인 입력 파일은 관심 배출 포인트를 정의하는 포인트 shapefile 입니다. 이 파일을 입력하는 경우, 해당 배출 포인트의 오르막 영역만을 평가할 것입니다.
Check for edge contamination		[boolean] 기본값: True	이 옵션으로 경계 오염을 확인해야 할지 여부를 결정합니다. 경계 오염이란 기여 영역 결정 시 영역 외부에 있는 그리드 셀을 고려하지 않기 때문에 값이 과소평가될 수도 있다는 가능성으로 정의됩니다.



산출물

라벨	명칭	유형	설명
Decayed Specific Catchment Area Grid		[raster]	D-Infinity 감퇴 집적 알고리즘은 어느 물질의 양이 D-Infinity 유수 방향으로 집적되지만, 셀에서 셀로 이동하면서 1 차 감퇴되는 영역 내부에 있는 각 위치에 집적되는 양의 그리드를 생성합니다.

파이썬 코드

알고리즘 ID: `taudem:dinfdecayaccum`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

D-Infinity 내리막 거리

이 알고리즘은 D-infinity 유수 모델을 사용해서 하천까지의 내리막 거리를 계산합니다. D-infinity 유수 모델은 다중 유수 방향 모델입니다. 각 그리드 셀에서 흘러나가는 유수가 2 개까지의 내리막 그리드 셀로 분산되기 때문입니다. 따라서 어떤 그리드 셀로부터 하천까지의 거리를 하나로 정의할 수 없습니다. 특정 그리드 셀에서 시작하는 유수가 서로 다른 여러 셀에서 하천과 합류할 수 있기 때문입니다. 통계적 방법을 선택해서 하천까지의 유수 경로 거리를 최장 거리, 최단 거리, 또는 가중 평균 거리로 구할 수 있습니다. 또는 총 직선 경로 (피타고라스), 직선 경로의 수평 구성 요소, 직선 경로의 수직 구성 요소, 또는 총 지표 유수 경로 같은 여러 거리 측정 방법 가운데 하나를 선택할 수도 있습니다.

파라미터

라벨	명칭	유형	설명
D-Infinity Flow Direction Grid		[raster]	D-infinity 방법으로 지정한 유수 방향 그리드입니다. 유수 방향을 라디안 단위로, 동쪽에서 반시계방향으로 측정합니다. D-Infinity 유수 방향 알고리즘이 이 그리드를 생성할 수 있습니다.
Pit Filled Elevation Grid		[raster]	표고 값 그리드를 입력합니다. 일반 규칙에 따라, 구덩이를 제거한 표고 값 그리드를 입력할 것을 권합니다. 구덩이란 일반적으로 DEM 상의 유수 분석을 방해하는 오류로 간주됩니다. 구덩이 제거 도구가 산출한 그리드를 사용하면 됩니다. 구덩이의 표고를 해당 영역의 물이 배수될 때까지 올려 구덩이를 제거한 표고 값을 담고 있는 그리드입니다.
Stream Raster Grid		[raster]	하천인 경우 그리드 셀 값을 1, 하천이 아닌 경우 셀 값을 0 으로 해서 하천을 나타내는 그리드입니다. 하천망 분석 도구 모음 가운데 하나로 이런 그리드를 생성할 수 있습니다.
Weight Path Grid 부가적		[raster]	거리 계산에 사용되는 가중치 (하중) 을 지정하는 그리드입니다. 예를 들면 버퍼를 통과하는 유수 거리만 계산하도록 할 경우 이 그리드를 사용할 수도 있습니다. 이 경우 버퍼 내부의 가중치가 1 이 되고 외부의 가중치는 0 이 됩니다. 또는 가중치가 지표면을 이동하기 위한, 어쩌면 이동 시간 또는 이동 과정 중의 감쇠를 나타내는 일종의 비용 함수를 반영할 수도 있습니다. 이 파일을 입력하지 않는 경우, 각 그리드 셀의 하중을 1 로 가정할 것입니다.
Statistical Method		[enumeration] 기본값: 2	하천까지의 내리막 거리를 계산하기 위한 통계적 방법을 지정합니다. D-infinity 유수 모델에서, 각 그리드 셀에서 흘러나가는 유수는 2 개까지의 내리막 그리드 셀로 분산됩니다. 따라서 어떤 그리드 셀로부터 하천까지의 거리를 하나의 값으로 정의할 수 없습니다. 특정 그리드 셀에서 시작하는 유수가 서로 다른 여러 셀에서 하천과 합류할 수 있기 때문입니다. 하천까지의 거리를 최장 (최대값), 최단 (최소값), 또는 하천까지의 내리막 거리의 가중 평균으로 정의할 수 있습니다. 옵션: • 0 –최소값 • 1 –최대값 • 2 –평균
Distance Method		[enumeration] 기본값: 1	하천까지의 내리막 거리를 계산하기 위한 거리 측정법을 지정합니다. 총 직선 경로 (피타고라스), 직선 경로의 수평 구성 요소 (수평), 직선 경로의 수직 구성 요소 (수직), 또는 총 지표 유수 경로 (지표) 같은 여러 거리 측정 방법 가운데 하나를 선택할 수 있습니다. 옵션: • 0 –피타고라스 • 1 –수평 • 2 –수직 • 3 –지표
Check for edge contamination		[boolean] 기본값: True	이 도구가 경계 오염을 확인해야 할지 여부를 나타내는 플래그입니다. 경계 오염이란 영역 바깥에 있는 그리드 셀을 계산에 넣지 않기 때문에 값이 저평가될 수도 있다는 가능성으로 정의됩니다. 내리막 거리라는 맥락에서, 어느 그리드 셀에서 나와 내리막을 따라

산출물

라벨	명칭	유형	설명
D-Infinity Drop to Stream Grid		[raster]	D-infinity 유수 모델 및 선택한 통계적 방법과 경로 측정법을 사용해서 계산한 하천까지의 거리를 담고 있는 산출 그리드입니다.

파이썬 코드

알고리즘 ID: `taudem:dinfdistdown`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

D-Infinity 오르막 거리

이 알고리즘은 각 셀에서 능선 (ridge) 셀까지의 거리를 D-infinity 유수 역방향을 따라 계산합니다. 능선 셀이란 더 이상 오르막 방향으로 기여하지 않는 그리드 셀을 말합니다. 다중 유수 경로가 어떤 그리드 셀에 수렴하는 경우, 어떤 그리드 셀을 지정하더라도 해당 셀은 오르막 능선 셀을 여럿 보유할 수 있습니다. 이 도구는 이런 유수 경로에 대한 최장 거리, 최단 거리, 그리고 대기 유수 평균 (waited flow average) 3 개의 통계법을 사용할 수 있습니다. 이런 통계법에서 사용되는 변수는 지정한 모든 셀에 대해 사용자가 지정한 한계값 (t) 을 초과하는 비율의 유수를 기여하는 그리드 셀만 오르막으로 간주합니다. t=0.5 로 설정하면 모든 그리드 셀에서 유수 경로 하나만 선택할 것입니다. 이렇게 되면 유수가 내리막 그리드 셀 2 개로 나뉘는 D-infinity 유수 모델보다는 D8 유수 모델에 더 가까워질 것입니다. 마지막으로, 총 직선 경로 (피타고라스), 직선 경로의 수평 구성 요소, 직선 경로의 수직 구성 요소, 또는 총 지표 유수 경로와 같은, 측정할 수 있는 부가적인 경로가 몇 개 존재합니다.

파라미터

라벨	명칭	유형	설명
D-Infinity Flow Direction Grid		[raster]	D-infinity 방법으로 지정한 유수 방향 그리드입니다. 유수 방향을 라디안 단위로, 동쪽에서 반시계방향으로 측정합니다. D-Infinity 유수 방향 알고리즘이 이 그리드를 생성할 수 있습니다.
Pit Filled Elevation Grid		[raster]	표고 값 그리드를 입력합니다. 일반 규칙에 따라, 구덩이를 제거한 표고 값 그리드를 입력할 것을 권합니다. 구덩이란 일반적으로 DEM 상의 유수 분석을 방해하는 오류로 간주됩니다. 구덩이 제거 도구가 산출한 그리드를 사용하면 됩니다. 구덩이의 표고를 해당 영역의 물이 배수될 때까지 올려 구덩이를 제거한 표고 값을 담고 있는 그리드입니다.
Slope Grid		[raster]	경사 값을 가진 입력 그리드입니다. 이 그리드는 고저차를 거리로 나눈 값으로 경사를 측정하며, D-Infinity 유수 방향 알고리즘의 산출물인 경우가 대부분입니다.
Statistical Method		[enumeration] 기본값: 2	하천까지의 내리막 거리를 계산하기 위한 통계적 방법을 지정합니다. D-infinity 유수 모델에서, 각 그리드 셀에서 흘러나가는 유수는 2 개까지의 내리막 그리드 셀로 분산됩니다. 따라서 어떤 그리드 셀로부터 하천까지의 거리를 하나의 값으로 정의할 수 없습니다. 특정 그리드 셀에서 시작하는 유수가 서로 다른 여러 셀에서 하천과 합류할 수 있기 때문입니다. 하천까지의 거리를 최장 (최대값), 최단 (최소값), 또는 하천까지의 내리막 거리의 가중 평균으로 정의할 수 있습니다. 옵션: • 0 – 최소값 • 1 – 최대값 • 2 – 평균
Distance Method		[enumeration] 기본값: 1	하천까지의 내리막 거리를 계산하기 위한 거리 측정법을 지정합니다. 총 직선 경로 (피타고라스), 직선 경로의 수평 구성 요소 (수평), 직선 경로의 수직 구성 요소 (수직), 또는 총 지표 유수 경로 (지표) 같은 여러 거리 측정 방법 가운데 하나를 선택할 수 있습니다. 옵션: • 0 – 피타고라스 • 1 – 수평 • 2 – 수직 • 3 – 지표
Proportion Threshold		[number] 기본값: 0.5	비율 한계값 파라미터입니다. 지정한 모든 셀에 대해 사용자가 지정한 한계값 `t` 를 초과하는 비율의 유수를 기여하는 그리드 셀만 오르막으로 간주합니다. `t=0.5` 로 설정하면 모든 그리드 셀에서 유수 경로 하나만 선택할 것입니다. 이렇게 되면 유수가 내리막 그리드 셀 2 개로 나뉘는 D-infinity 유수 모델보다는 D8 유수 모델에 더 가까워질 것입니다.
Check for edge contamination		[boolean] 기본값: True	이 옵션으로 경계 오염을 확인해야 할지 여부를 결정합니다. 경계 오염이란 영역 외부에 있는 그리드 셀을 고려하지 않기 때문에 값이 과소평가될 수도 있다는 가능성으로 정의됩니다.

산출물

라벨	명칭	유형	설명
D-Infinity Distance Up		[raster]	D-infinity 유수 모델 및 선택한 통계적 방법과 경로 측정법을 사용해서 계산한 능선까지의 거리를 담고 있는 산출 그리드입니다.

파이썬 코드

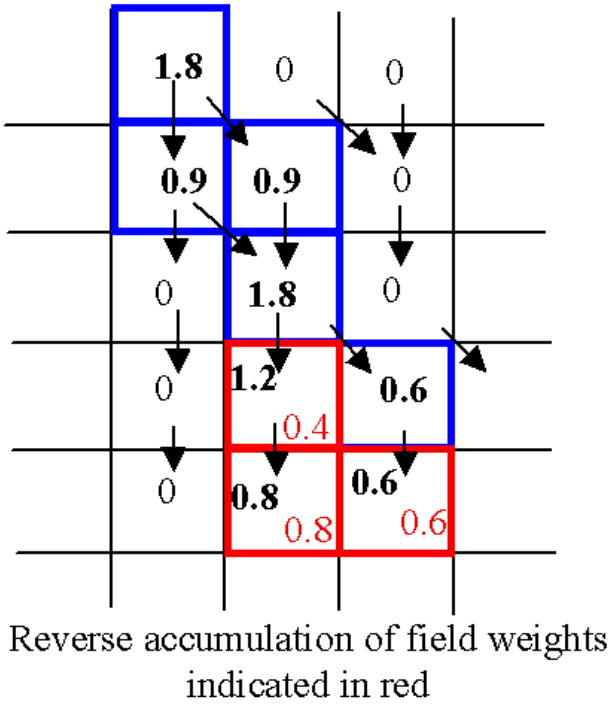
알고리즘 ID: `taudem:dinfdistup`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

D-Infinity 역 집적

이 알고리즘은 가중 기여 영역 판정과 비슷한 방식으로 동작합니다. 다만 각 그리드 셀에서 내리막 가중 하중을 집적하는 것이 아니라, 그 유수 방향의 역방향을 따라 오르막 가중 하중을 역산해서 집적량을 계산합니다. 이 알고리즘은 최대 내리막 그리드에 있는 각 그리드 셀에서 나오는 내리막 가중 하중의 최대값도 산출합니다.



Reverse accumulation of field weights indicated in red



내리막 방향으로 영향을 주는 활동 때문에 일어날 수도 있는 재해를 평가하고 매핑하기 위해 이 알고리즘을 개발했습니다. 토사 유출을 증가시키는 토지 관리 활동을 한 예로 들 수 있습니다. 토사 유출은 때로 산사태 또는

석편류 (debris flow) 를 일으키기도 하기 때문에, 이 가중 그리드를 지형 안정성 지도로 사용할 수도 있습니다. 그러면 역 집적으로 각 그리드 셀에서 토사 유출을 증가시킬 수도 있는 활동의 위험성 지표로서 내리막 불안정 지형의 면적을 측정할 수 있습니다. 현지에 어떤 영향을 끼칠 가능성이 없다고 해도 말입니다.

파라미터

라벨	명칭	유형	설명
D-Infinity Flow Direction Grid		[raster]	D-infinity 방법으로 지정한 유수 방향 그리드입니다. 유수 방향을 라디안 단위로, 동쪽에서 반시계방향으로 측정합니다. D-Infinity 유수 방향 알고리즘이 이 그리드를 생성할 수 있습니다.
Weight Grid		[raster]	집적 계산에 사용되는 가중치 (하중) 를 지정하는 입력 그리드입니다.

산출물

라벨	명칭	유형	설명
Reverse Accumulation Grid		[raster]	**역 집적** 알고리즘의 산출 그리드입니다. 이 알고리즘은 가중 기여 영역 판정과 비슷한 방식으로 동작합니다. 다만 각 그리드 셀에서 내리막 가중 하중을 집적하는 것이 아니라, 그 유수 방향의 역방향을 따라 오르막 가중 하중을 역산해서 집적량을 계산합니다.
Maximum Downslope Grid		[raster]	각 그리드 셀에서 내리막 가중 하중의 최대값을 나타내는 산출 그리드입니다.

파이썬 코드

알고리즘 ID: `taudem:dinfrevaccum`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

D-Infinity 이송 제약 집적 - 2

공급 및 이송시키기 위한 유수 능력에 따라 제약될 수도 있는 어떤 (침전물 같은) 물질의 이송 및 퇴적을 계산하기 위해 이 알고리즘을 개발했습니다. 이 알고리즘은 어느 그리드 셀에서 나가는 이송량은 공급량 및 이송 능력 `Tcap` 가운데 최소값이라는 규칙을 따르는 (침전물 이송 같은) 물질 유동량을 집적합니다. 어떤 그리드 셀의 총 공급량은 오르막 그리드 셀에서 이송된 양의 합계 `Tin` 에 해당 셀의 (침식 등의) 공급 기여량 `E` 를 더해서 계산합니다. 이 알고리즘은 총 공급량에서 실제 이송량을 뺀 퇴적량 `D` 도 산출합니다.

여기에서 `E` 는 공급량입니다. 각 그리드 셀에서 `Tout` 은 내리막 그리드 셀의 경우 `Tin` 이 되며, 이송 제약 집적 `tla` 로 산출됩니다. `D` 는 퇴적량 `tdep` 입니다. 이 알고리즘은 이송된 물질에 붙어 있는 어떤 합성물 (오염물질) 의 농도를 판정할 수 있는 옵션을 제공하는데, 다음과 같이 판정합니다:

$$T_{out} = \min(E + \sum T_{in}, T_{cap})$$

$$D = E + \sum T_{in} - T_{out}$$

$$L_{in} = \sum T_{in} C_{in}$$

이때 `Lin` 은 들어오는 합성물 하중이고 `Cin` 및 `Tin` 은 각각 오르막 그리드 셀에서 들어오는 농도 및 이송량을 뜻합니다.

$$T_{out} < \sum T_{in}$$

만약

이거나 또는

일 때 `Cs` 는 해당 셀이 공급하는 농도를 뜻하며, 우항에서 두 번째 항목의 차는 해당 그리드 셀에서 추가로 공급하는 농도를 뜻합니다. 따라서,

이 알고리즘은 각 그리드 셀에서 `Cout` 을 계산해서 농도 그리드를 산출합니다.

배출 (outlet) shapefile 을 사용한 경우, 이 알고리즘은 shapefile 이 지정한 위치로 유수를 기여하는 영역 부분만 평가할 것입니다.

이송 제약 집적은 침전물 이송 비율과 침전물에 부착된 오염물질의 공간 의존성을 포함하는 침식 및 침전을 실어나르는 모델 작업에 유용합니다.

$$L_{out} = L_{in}\left(T_{out} / \sum T_{in}\right)$$

$$L_{out} = L_{in} + C_s\left(T_{out} - \sum T_{in}\right)$$

파라미터

라벨	명칭	유형	설명
D-Infinity Flow Direction Grid		[raster]	D-infinity 방법으로 지정한 유수 방향 그리드입니다. 유수 방향을 라디안 단위로, 동쪽에서 반시계방향으로 측정합니다. D-Infinity 유수 방향 알고리즘이 이 그리드를 생성할 수 있습니다.
Supply Grid		[raster]	이송 제약 집적 함수에 물질의 공급량 (하중) 을 제공하는 그리드입니다. 침식에 응용할 경우, 이 그리드가 각 그리드 셀에서 공급되는 침식 분산탈리 (detachment) 또는 침전물을 지정할 것입니다.
Transport Capacity Grid		[raster]	이송 제약 집적 함수를 위해 각 그리드 셀에서 이송 능력을 제공하는 그리드입니다. 침식에 응용할 경우, 이 그리드가 실어나르는 유수의 이송 능력을 지정할 것입니다.
Input Concentration Grid		[raster]	이송 제약 집적 함수에 공급량 가운데 관심 화합물의 농도를 제공하는 그리드입니다. 침식에 응용할 경우, 이 그리드가 침식된 침전물에 부착된, 예를 들어 인 (□) 의 농도를 지정할 것입니다.
Outlets Shapefile 부가적		[vector: point]	이 부가적인 입력 파일은 관심 배출 포인트를 정의하는 포인트 shapefile 입니다. 이 파일을 입력하는 경우, 해당 배출 포인트의 오르막 영역만을 평가할 것입니다.
Check for edge contamination		[boolean] 기본값: True	이 옵션으로 경계 오염을 확인해야 할지 여부를 결정합니다. 경계 오염이란 결과물 산출 시 영역 외부에 있는 그리드 셀을 고려하지 않기 때문에 값이 과소평가될 수도 있다는 가능성으로 정의됩니다.

$$C_{out} = L_{out} / T_{out}$$

산출물

라벨	명칭	유형	설명
Transport Limited Accumulation Grid		[raster]	이 그리드는 이송 능력 제약을 반영한 집적된 공급량의 가중 집적 그리드입니다. 모든 그리드 셀에서 나오는 이송량이 해당 그리드 셀에 대한 (해당 셀의 공급량에 들어오는 이송량을 더한) 총 공급량과 이송 능력 가운데 최소값이라는 규칙을 따르는 물질 유동을 집적해서 계산한 이송율을 산출합니다.
Deposition Grid		[raster]	이송 제약 집적에서 산출된 퇴적량을 제공하는 그리드입니다. 퇴적량이란 각 그리드 셀로 들어오는 이송량에서 그리드 셀에서 나가는 이송 능력을 뺀 나머지를 말합니다. 퇴적 그리드는 들어오는 이송량과 해당 셀의 공급량을 더한 값에서 나가는 이송량을 뺀 값으로 계산합니다.
Output Concentration Grid		[raster]	공급량 그리드에 입력 농도를 지정한 경우, (침전물과 같은) 이송되는 물질에 부착 또는 결합돼 있는 화합물 (오염물질) 의 농도를 계산해서 지정한 이 그리드도 산출합니다.

파이썬 코드

알고리즘 ID: 알 수 없음

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

D-Infinity 이송 제약 집적

공급 및 이송시키기 위한 유수 능력에 따라 제약될 수도 있는 어떤 (침전물 같은) 물질의 이송 및 퇴적을 계산하기 위해 이 알고리즘을 개발했습니다. 이 알고리즘은 어느 그리드 셀에서 나가는 이송량은 공급량 및 이송 능력 `Tcap` 가운데 최소값이라는 규칙을 따르는 (침전물 이송 같은) 물질 유동량을 집적합니다. 어떤 그리드 셀의 총 공급량은 오르막 그리드 셀에서 이송된 양의 합계 `Tin` 에 해당 셀의 (침식 등의) 공급 기여량 `E` 를 더해서 계산합니다. 이 알고리즘은 총 공급량에서 실제 이송량을 뺀 퇴적량 `D` 도 산출합니다.

$$T_{out} = \min(E + \sum T_{in}, T_{cap})$$

$$D = E + \sum T_{in} - T_{out}$$

여기에서 E 는 공급량입니다. 각 그리드 셀에서 Tout 은 내리막 그리드 셀의 경우 Tin 이 되며, 이송 제약 집적 tla 로 산출됩니다. D 는 퇴적량 tdep 입니다. 이 알고리즘은 이송된 물질에 붙어 있는 어떤 합성물 (오염물질) 의 농도를 판정할 수 있는 옵션을 제공하는데, 다음과 같이 판정합니다:

$$L_{in} = \sum T_{in} C_{in}$$

이때 Lin 은 들어오는 합성물 하중이고 Cin 및 Tin 은 각각 오르막 그리드 셀에서 들어오는 농도 및 이송량을 뜻합니다.

$$T_{out} < \sum T_{in}$$

만약

$$L_{out} = L_{in} \left(T_{out} / \sum T_{in} \right)$$

이거나 또는

일 때 Cs 는 해당 셀이 공급하는 농도를 뜻하며, 우항에서 두 번째 항목의 차는 해당 그리드 셀에서 추가로 공급하는 농도를 뜻합니다. 따라서,

이 알고리즘은 각 그리드 셀에서 Cout 을 계산해서 농도 그리드를 산출합니다.

배출 (outlet) shapefile 을 사용한 경우, 이 알고리즘은 shapefile 이 지정한 위치로 유수를 기여하는 영역 부분만 평가할 것입니다.

이송 제약 집적은 침전물 이송 비율과 침전물에 부착된 오염물질의 공간 의존성을 포함하는 침식 및 침전을 실어나르는 모델 작업에 유용합니다.

$$L_{out} = L_{in} + C_s\left(T_{out} - \sum T_{in}\right)$$

$$C_{out} = L_{out} / T_{out}$$

파라미터

라벨	명칭	유형	설명
D-Infinity Flow Direction Grid		[raster]	D-infinity 방법으로 지정한 유수 방향 그리드입니다. 유수 방향을 라디안 단위로, 동쪽에서 반시계방향으로 측정합니다. D-Infinity 유수 방향 알고리즘이 이 그리드를 생성할 수 있습니다.
Supply Grid		[raster]	이송 제약 집적 함수에 물질의 공급량 (하중) 을 제공하는 그리드입니다. 침식에 응용할 경우, 이 그리드가 각 그리드 셀에서 공급되는 침식 분산탈리 (detachment) 또는 침전물을 지정할 것입니다.
Transport Capacity Grid		[raster]	이송 제약 집적 함수를 위해 각 그리드 셀에서 이송 능력을 제공하는 그리드입니다. 침식에 응용할 경우, 이 그리드가 실어나르는 유수의 이송 능력을 지정할 것입니다.
Outlets Shapefile 부가적		[vector: point]	이 부가적인 입력 파일은 관심 배출 포인트를 정의하는 포인트 shapefile 입니다. 이 파일을 입력하는 경우, 해당 배출 포인트의 오르막 영역만을 평가할 것입니다.
Check for edge contamination		[boolean] 기본값: True	이 옵션으로 경계 오염을 확인해야 할지 여부를 결정합니다. 경계 오염이란 결과물 산출 시 영역 외부에 있는 그리드 셀을 고려하지 않기 때문에 값이 과소평가될 수도 있다는 가능성으로 정의됩니다.

산출물

라벨	명칭	유형	설명
Transport Limited Accumulation Grid		[raster]	이 그리드는 이송 능력 제약을 반영한 집적된 공급량의 가중 집적 그리드입니다. 모든 그리드 셀에서 나오는 이송량이 해당 그리드 셀에 대한 (해당 셀의 공급량에 들어오는 이송량을 더한) 총 공급량과 이송 능력 가운데 최소값이라는 규칙을 따르는 물질 유동을 집적해서 계산한 이송율을 산출합니다.
Deposition Grid		[raster]	이송 제약 집적에서 산출된 퇴적량을 제공하는 그리드입니다. 퇴적량이란 각 그리드 셀로 들어오는 이송량에서 그리드 셀에서 나가는 이송 능력을 뺀 나머지를 말합니다. 퇴적 그리드는 들어오는 이송량과 해당 셀의 공급량을 더한 값에서 나가는 이송량을 뺀 값으로 계산합니다.

파이썬 코드

알고리즘 ID: `taudem:dinftranslimaccum`

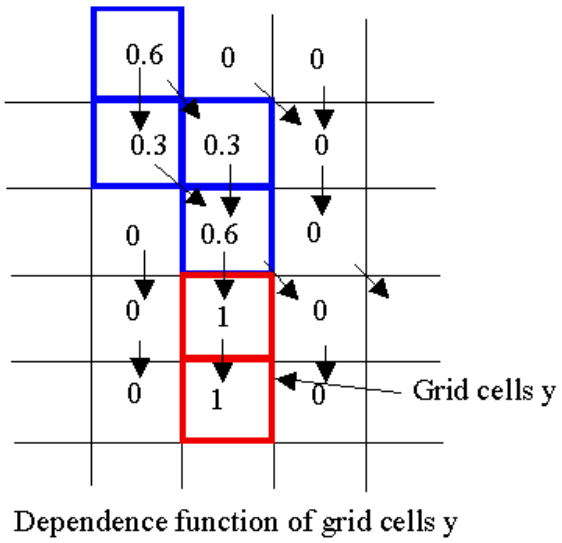
```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

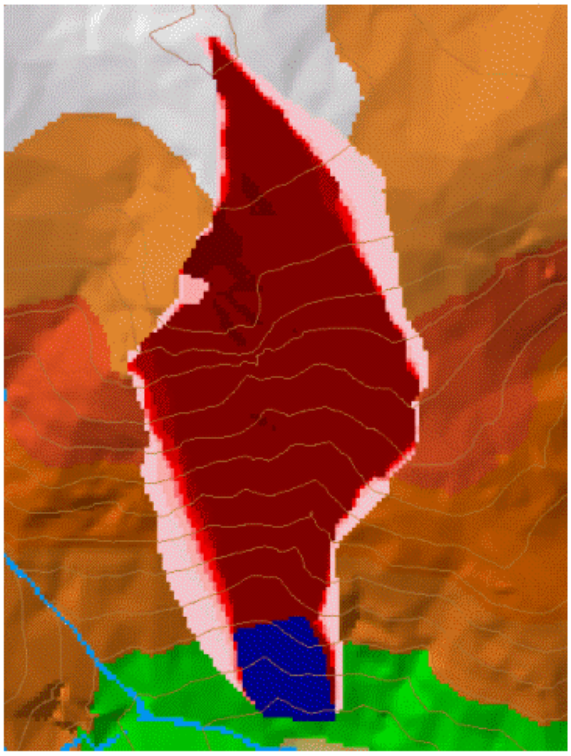
D-Infinity 오르막 의존

D-Infinity 오르막 의존 알고리즘은 영역 안에 있는 각 그리드 셀이 대상 그리드 셀 집합에 기여하는 양을 정량화합니다. 각 그리드 셀에서 여러 내리막 그리드 셀로 흐르는 유수의 비율을 D-Infinity 유수 방향 알고리즘으로 계산합니다. 이 내리막 유수 비율을 따라 각 그리드 셀에서 시작되어 대상 영역으로 흐르는 유수량을 정의합니다. 각 그리드 셀에서 내리막 그리드 셀들을 검사하는 내리막 회귀를 사용해서 오르막 영향을 판정, 산출된 맵으로 대상 영역을 흘러가는 유수가 시작된 오르막 영역을 식별하거나, 또는 대상 영역의 유수가 의존하는 영역을 식별합니다.

다음 그림은 `x` 영역 (파랑색) 에 있는 각 소스 포인트가 대상 포인트 혹은 `y` 영역 (빨강색) 에 기여하는 양을 묘사하고 있습니다. 가중 기여 영역 함수의 지표를 `I(y; x)` 로 나타내는 경우, 특정 그리드 셀 `y` 에서 그리드 셀 `x` 로 가중 기여치를 단위값 (1) 로 지정하면, 오르막 의존은 `D(x; y) = I(y; x)` 로 표현됩니다.



Dependence function of grid cells y



이 알고리즘은, 예를 들어 대상 지역으로 들어오는 유수 또는 유수에 포함된 [오염] 물질이 어디에서 시작되었는지 추적하는 데 유용합니다.

파라미터

라벨	명칭	유형	설명
D-Infinity Flow Direction Grid		[raster]	관심 그리드 셀을 중심으로 하는 3x3 그리드 셀 크기의 창에 형성된 8 개의 삼각면 가운데 가장 심한 내리막 경사 방향으로 유수 방향 각도를 결정하는 D-Infinity 방법으로 유수 방향을 지정하는 그리드입니다. D-Infinity 유수 방향 알고리즘이 이 그리드를 생성할 수 있습니다.
Destination Grid		[raster]	오르막 방향에서 유수를 받아들일 수도 있는 대상 영역을 나타내는 그리드입니다. `y` 영역 내부는 1, 영역의 나머지 부분은 전부 0 이어야만 합니다.

산출물

라벨	명칭	유형	설명
Output Upslope Dependence Grid		[raster]	영역 내부에 있는 각 소스 포인트가 대상 그리드가 정의하는 영역에 기여하는 양을 정량화하는 그리드입니다.

파이썬 코드

알고리즘 ID: `taudem:dinfupdependence`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

내리막 평균 경사

이 알고리즘은 D8 내리막 방향에서 사용자가 지정한 거리 안의 경사의 평균을 계산합니다. 이때 거리는 수평 맵 단위로 지정해야 합니다.

파라미터

라벨	명칭	유형	설명
D8 Flow Direction Grid		[raster]	이 입력물은 모든 셀에서의 유수가 전부 가장 심한 내리막 경사 방향에 있는 단일 이웃 셀로 흐르는 D8 방법을 사용해서 인코딩된 유수 방향 그리드입니다. 이 그리드는 D8 유수 방향 알고리즘의 산출물로 얻을 수 있습니다.
Pit Filled Elevation Grid		[raster]	표고 값 그리드를 입력합니다. 일반 규칙에 따라, 구덩이를 제거한 표고 값 그리드를 입력할 것을 권합니다. 구덩이란 일반적으로 DEM 상의 유수 분석을 방해하는 오류로 간주됩니다. 구덩이 제거 도구가 산출한 그리드를 사용하면 됩니다. 구덩이의 표고를 해당 영역의 물이 배수될 때까지 올려 구덩이를 제거한 표고 값을 담고 있는 그리드입니다.
Downslope Distance		[number] 기본값: 50	경사를 계산할 (평균 맵 단위) 내리막 거리 입력 파라미터입니다.

산출물

라벨	명칭	유형	설명
Slope Average Down Grid		[raster]	사용자 지정 거리 범위 안에서 D8 내리막 방향으로 경사의 평균을 계산한 산출 경사 그리드입니다.

파이썬 코드

알고리즘 ID: `taudem:slopeavedown`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

영역 전체 경사 비율

특정 집수 영역 (기여 영역) 에 대한 경사의 비율을 계산합니다. 이 알고리즘은 좀 더 일반적인 `ln(a/tan beta)` 습윤 지수와 대수적으로 관계가 있지만, 경사가 0 인 경우 0 으로 나누는 오류를 피하기 위해 분모에 기여 영역을 넣습니다.

파라미터

라벨	명칭	유형	설명
Slope Grid		[raster]	경사 값을 가진 입력 그리드입니다. D8 유수 방향 또는 D-Infinity 유수 방향 알고리즘 가운데 하나를 사용해서 이 그리드를 생성할 수 있습니다.
Specific Catchment Area Grid		[raster]	각 셀의 고유 기여에 해당 셀로 흘러드는 오르막 이웃들의 기여를 더한 기여 영역 값을 제공하는 그리드입니다. 그리드 셀의 개수 (또는 가중치의 총체)로 기여 영역을 집계합니다. D8 기여 영역 또는 D-Infinity 기여 영역 알고리즘 가운데 하나를 사용해서 이 그리드를 생성할 수 있습니다.

산출물

라벨	명칭	유형	설명
Slope Divided By Area Ratio Grid		[raster]	특정 집수 영역 (기여 영역) 에 대한 경사 비율 그리드입니다. 좀 더 일반적인 `ln(a/tan beta)` 습윤 지수와 대수적으로 관계가 있지만, 경사가 0 인 경우 0 으로 나누는 오류를 피하기 위해 분모에 기여 영역을 넣습니다.

파이썬 코드

알고리즘 ID: `taudem:slopearearatio`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

지형 습윤 지수 (TWI)

지형 습윤 지수 (Topographic Wetness Index) 를 계산합니다.

파라미터

라벨	명칭	유형	설명
Slope		[raster]	경사 값을 가진 입력 그리드입니다. D8 유수 방향 또는 D-Infinity 유수 방향 알고리즘 가운데 하나를 사용해서 이 그리드를 생성할 수 있습니다.
Specific catchment area		[raster]	각 셀의 고유 기여에 해당 셀로 흘러드는 오르막 이웃들의 기여를 더한 기여 영역 값을 제공하는 그리드입니다. 그리드 셀의 개수 (또는 가중치의 총체) 로 기여 영역을 집계합니다. D8 기여 영역 또는 D-Infinity 기여 영역 알고리즘 가운데 하나를 사용해서 이 그리드를 생성할 수 있습니다.

산출물

라벨	명칭	유형	설명
Wetness index		[raster]	지형습윤지수 (TWI) 그리드

파이썬 코드

알고리즘 ID: `taudem:twi`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

24.4.3 하천망 분석

하류 연결하기

이 알고리즘은 입력 래스터의 (예: 그리드로 변환된 HUC 의) 각 구역마다 가장 큰 AreaD8 을 가진 포인트를 식별합니다. 식별된 포인트를 배수구 (outlet) 로 간주하고, OGR 파일을 생성합니다. 유수 방향을 사용해서 각 배수구를 사용자가 제어할 수 있는 (기본값은 1) 그리드 셀 개수만큼 하류 방향으로 이동시킵니다. 해당 포인트가 이동해 간 위치의 ID 를 iddown 으로 간주합니다. 초기 포인트를 가진 파일과 이동한 포인트를 가진 파일, 2 개의 OGR 파일을 생성합니다. 두 파일 모두 ID, iddown 및 AreaD8 을 담고 있습니다.

파라미터

라벨	명칭	유형	설명
D8 flow directions		[raster]	모든 셀에서의 유수가 전부 가장 심한 내리막 경사 방향에 있는 단일 이웃 셀로 흐르는 D8 방법을 사용해서 인코딩된 유수 방향 그리드입니다.
D8 contribution area		[raster]	D8 알고리즘을 사용해서 해당 셀 고유의 기여와 해당 셀로 흘러드는 모든 오르막 이웃에서 나오는 기여를 더한 값을 각 셀 별 그리드 셀 개수 (또는 가중치의 총체) 라는 측면에서 계산한 기여 영역 값을 제공하는 그리드입니다. 일반적으로 이 그리드는 **"D8 기여 영역"**알고리즘의 산출물입니다.
Watershed		[raster]	GageWatershed 함수 또는 StreamReachWatershed 함수로 그린 집수구역 그리드입니다. 다른 (예: HUC) 집수구역 래스터도 집수구역 그리드로 사용할 수 있습니다.
Grid cells move to downstream		[number]	유수 방향을 기반으로 하류 방향으로 이동한 그리드 셀의 개수

산출물

라벨	명칭	유형	설명
Outlets		[vector: point]	각 포인트가 각 구역마다 가장 큰 기여 영역을 가진 집수구역 그리드에서 생성된 포인트 OGR 파일
Moved Outlets		[vector: point]	이동한 관심 배수구를 정의하는 포인트 OGR 파일. 각 배수구는 유수 방향을 사용해서 지정한 그리드 셀 개수만큼 하류 방향으로 이동합니다.

파이썬 코드

알고리즘 ID: `taudem:connectdown`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

D8 오르막 극값

입력 그리드에서 D8 유수 모델을 바탕으로 오르막 극값 (최대 또는 최소 가운데 하나) 을 판정합니다. 이 알고리즘은 원래 (낙차 분석에 따른) 최적의 하천망을 산출하는 경사와 면적의 곱의 한계값을 식별하기 위한 하천 래스터를 생성하기 위해 개발됐습니다.

부가적인 배출 포인트 shapefile 을 사용하는 경우, 배출 셀과 해당 셀의 (D8 유수 모델에 따른) 오르막 셀들이 있는 영역만 평가합니다.

기본적으로, 이 도구는 경계 오염을 확인합니다. 경계 오염이란 영역 바깥에 있는 그리드 셀을 계산에 넣지 않기 때문에 결과가 저평가될 수도 있다는 가능성으로 정의됩니다. 경계 오염은 배수가 경계로부터 안 쪽으로 흐르거나

또는 표고값이 NODATA 인 영역에서 발생합니다. 이 알고리즘은 이를 인식하고 이 그리드 셀들의 결과에 대해 NODATA 를 보고합니다. 경계선에서 영역으로 들어오는 유수 경로를 따라 NODATA 값들이 쭉 나열되는 것을 흔히 볼 수 있습니다. 이는 기대대로의 효과로서, 해당 그리드 셀들의 결과가 사용할 수 있는 데이터 영역 외부에 있는 지형에 의존하기 때문에 그 기여 영역을 알 수 없다는 사실을 나타냅니다. 사용자가 경계 오염이 문제가 되지 않는다는 사실을 알고 있거나 또는 이런 문제를 무시하고자 하는 경우, 예를 들면 유역 외곽선을 따라 DEM 을 잘라낸 경우라면 경계 오염 확인 옵션을 끌 수도 있습니다.

파라미터

라벨	명칭	유형	설명
D8 Flow Directions Grid		[raster]	각 셀 별로 그 인접 또는 대각선 방향의 이웃 8 개 셀 가운데 가장 심한 내리막 경사를 가지고 있는 셀의 방향으로 정의한 D8 유수 방향 그리드입니다. 이 그리드는 **"D8 유수 방향"**알고리즘의 산출물로 얻을 수 있습니다.
Upslope Values Grid		[raster]	오르막 최대값 또는 최소값을 선택한 값의 그리드입니다. 가장 흔히 쓰이는 값은 낙차 분석에 따라 하천 래스터를 생성하는 경우 필요한 경사와 면적의 곱입니다.
Outlets Shapefile 부가적		[vector: point]	관심 배출 포인트를 정의하는 포인트 shapefile 입니다. 이 파일을 입력하는 경우, 알고리즘이 해당 배출 셀의 주변 오르막 영역만 평가할 것입니다.
Check for edge contamination		[boolean] 기본값: True	알고리즘이 경계 오염을 확인할지 여부를 나타내는 옵션입니다.
Use max upslope value		[boolean] 기본값: True	오르막 경사의 최대값 또는 최소값 가운데 어느 쪽을 계산할지 나타내는 옵션입니다.

산출물

라벨	명칭	유형	설명
Extreme Upslope Values Grid		[raster]	오르막 경사의 최대값/최소값 그리드입니다.

파이썬 코드

알고리즘 ID: `taudem:d8flowpathextremeup`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

측정기 유역

측정기 유역 (Gage Watershed) 그리드를 계산합니다. 각 그리드 셀에 해당 셀이 다른 어떤 측정기도 통과하지 않고 직접 흘러드는 측정기의 (`id` 열에서 나온) 식별자 라벨을 부여합니다.

파라미터

라벨	명칭	유형	설명
D-infinity flow directions	`DINF_FLOWDIR`	[raster]	D-infinity 유수법을 기반으로 한 유수 방향 그리드
D8 Flow Directions Grid		[raster]	각 셀 별로 그 인접 또는 대각선 방향의 이웃 8 개 셀 가운데 가장 심한 내리막 경사를 가지고 있는 셀의 방향으로 정의한 D8 유수 방향 그리드입니다. 이 그리드는 **"D8 유수 방향"**알고리즘의 산출물로 얻을 수 있습니다.
Gages Shapefile		[vector: point]	유역을 묘사할 측정기들을 정의하는 포인트 shapefile 입니다. 이 shapefile 은 `id` 열을 가지고 있어야 합니다. 이 shapefile 에서 각 포인트로 직접 흘러드는 그리드 셀들에 이 `id` 라벨을 부여할 것입니다.

산출물

라벨	명칭	유형	설명
Gage Watershed Grid		[raster]	각 측정기 유역을 식별하는 그리드입니다. 각 그리드 셀에 해당 셀이 다른 어떤 측정기도 통과하지 않고 직접 흘러드는 측정기의 (`id` 열에서 나온) 식별자 라벨을 부여합니다.
Downstream Identifiers File		[file]	유역의 내리막 연결성을 지정하는 텍스트 파일입니다.

파이썬 코드

알고리즘 ID: `taudem:gagewatershed`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

길이 면적 하천원

오르막 경로 길이, D8 기여 영역 그리드 입력, 그리고 파라미터 M 과 y 를 바탕으로 A >= (M)(Ly) 를 평가하는 지표 (indicator) 그리드 (1, 0) 을 생성합니다. 이 그리드는 하천원 (stream source) 일 가능성이 높은 그리드 셀들을 표시합니다. 이 알고리즘은 실험적인 메소드로, 하천 L ~ A 0.6 에 대해 앞의 내용을 서술하고 있는 핵의 법칙 을 그 이론적 기반으로 삼고 있습니다. 하지만 산사면 (hillslope) 의 경우 평행 유수 L ~ A 를 대상으로 합니다. 따라서 산사면에서 하천으로의 전이를 L ~ A 0.8 로 표현할 수도 있는데, 그렇다면 그리드 셀이 A > M (L (1/0.8)) 인 경우 하천 셀이라고 식별할 수 있을 것입니다.

파라미터

라벨	명칭	유형	설명
Length Grid		[raster]	각 셀 별 최장 오르막 길이를 담고 있는 그리드입니다. 최장 오르막 길이는 각 셀로 흘러드는 셀 가운데 가장 멀리 있는 셀에서 시작되는 유수 경로의 길이로 계산합니다. 이 길이는 셀 크기 및 방향의 인접/대각선 여부를 고려해서 셀 중심점 간의 거리로 측정합니다. 어떤 셀을 하천 셀로 간주할 것인지 결정하기 위한 A >(M)(Ly) 공식에 사용되는 길이 L 이 바로 이 최장 오르막 길이입니다. 이 그리드는 **"그리드 망"** 알고리즘의 산출물입니다.
Contributing Area Grid		[raster]	D8 알고리즘을 사용해서 계산한 각 셀 별 기여 영역 값 그리드입니다. 어느 셀의 기여 영역 값은 해당 셀 고유의 기여와 해당 셀로 흘러드는 모든 오르막 이웃에서 나오는 기여를 더한 값으로, 셀 개수로 측정됩니다. 일반적으로 이 그리드는 **"D8 기여 영역"**알고리즘의 산출물입니다. 길이 면적 하천원 알고리즘에서 하천으로의 전이 여부를 결정하는 A > (M)(Ly) 공식 안의 비교 대상인 기여 영역 A 가 바로 이 기여 영역 값입니다.
Threshold		[number] 기본값: 0.03	하천의 시작을 식별하는 공식 A > (M)(Ly) 에 사용되는 곱수 한계값 M 파라미터입니다.
Exponent		[number] 기본값: 1.3	하천의 시작을 식별하는 공식 A > (M)(Ly) 에 사용되는 지수 (exponent) y 파라미터입니다. 분기 체계에서, 핵의 법칙은 1/y = 0.6 (또는 0.56) 인 L = 1/M A(1/y) 를 제안하고 있습니다. (y 가 약 1.7 입니다.) 평행 유수 체계에서 L 은 A 에 비례합니다. (y 가 약 1 입니다.) 이 메소드는 그 사이 어딘가에 있는 지수 y 를 사용해서 이 두 패러다임 사이의 전이를 식별하고자 합니다. (y 가 약 1.3 입니다.)

산출물

라벨	명칭	유형	설명
Stream Source Grid		[raster]	최장 오르막 경로 길이, D8 기여 영역 그리드 입력, 그리고 파라미터 M 과 y 를 바탕으로 A >= (M)(L^y) 를 평가하는 지표 (indicator) 그리드 (1, 0) 입니다. 이 그리드는 하천원 (stream source) 일 가능성이 높은 그리드 셀들을 표시합니다.

파이썬 코드

알고리즘 ID: `taudem:lengtharea`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

배출 포인트를 하천으로 이동

하천 래스터 그리드에서 하천 셀과 정렬되지 않은 배출 포인트를 내리막 D8 유수 방향을 따라 하천 래스터 셀과 만날 때까지, 또는 그리드 셀들의 `max_dist` 개수가 판정될 때까지, 또는 유수 경로가 영역을 벗어날 때까지 (예를 들어 D8 유수 방향의 NODATA 값이 나올 때까지) 이동시킵니다. 이 알고리즘의 산출 파일은 새로운 배출 shapefile 로, 가능한 경우 각 배출 포인트를 하천 래스터 그리드와 일치하도록 이동시킨 shapefile 입니다. 이 새로운 shapefile 에는 각 배출 포인트가 변경된 내용을 나타내는 `dist_moved` 필드가 추가돼 있습니다. 이미 하천 셀 위에 있는 포인트들은 이동되지 않으며, 해당 포인트의 `dist_moved` 필드엔 0 값이 할당됩니다. 초기에 하천 셀 위에 없던 포인트들은 내리막 D8 유수 방향을 따라 다음 사항 가운데 하나가 일어날 때까지 이동됩니다: 1) 그리드 셀의 `max_dist` 개수를 넘기 전에 하천 그리드 셀과 만나는 경우, 포인트를 이동시키고 `dist_moved` 필드에 포인트가 몇 개의 그리드 셀들을 이동했는지 나타내는 값을 할당합니다. 2) 그리드 셀의 `max_dist` 개수를 넘거나, 또는 3) 영역 밖으로 나가는 경우 (예를 들어 D8 유수 방향의 NODATA 값을 만나는 경우) 배출 포인트를 이동시키지 않고 `dist_moved` 필드에 -1 값을 할당합니다.

파라미터

라벨	명칭	유형	설명
D8 Flow Direction Grid		[raster]	각 셀 별로 그 인접 또는 대각선 방향의 이웃 8 개 셀 가운데 가장 심한 내리막 경사를 가지고 있는 셀의 방향으로 정의한 D8 유수 방향 그리드입니다. 이 그리드는 **"D8 유수 방향"**알고리즘의 산출물로 얻을 수 있습니다.
Stream Raster Grid		[raster]	이 입력물은 하천인 경우 그리드 셀 값을 1, 나머지 셀 값을 0 으로 해서 하천의 위치를 나타내는 지표 (indicator) 그리드 (1, 0) 입니다. 그리드입니다. **"하천망 분석"**도구 모음 가운데 하나로 이런 그리드를 생성할 수 있습니다.
Outlets Shapefile		[vector: point]	관심 포인트 또는 배출 포인트를 정의하는 포인트 shapefile 입니다. 이상적으로는 이런 포인트들이 하천 위에 위치해 있어야 하지만, shapefile 포인트 위치가 하천 래스터 그리드에 대해 정확히 등록돼 있지 않을 수도 있기 때문에 하천 위에 정확히 위치하지 않을 수도 있습니다.
Maximum Number of Grid Cells to traverse		[number] 기본값: 50	이 입력 파라미터는 입력 배출 shapefile 안에 있는 포인트들이 산출 배출 shapefile 로 저장되기 전에 이동할 최대 그리드 셀 개수입니다.

산출물

라벨	명칭	유형	설명
Output Outlet Shapefile		[vector: point]	관심 포인트 또는 배출 포인트를 정의하는 포인트 shapefile 입니다. 이 파일이 보유한 포인트 개수는 입력 배출 shapefile 이 보유한 포인트 개수와 동일합니다. 입력 포인트가 하천 위에 존재하는 경우, 포인트는 이동하지 않습니다. 입력 포인트가 하천 위에 존재하지 않는 경우, 포인트를 내리막 D8 유수 방향을 따라 하천을 만날 때까지 또는 최장 거리까지 이동시킵니다. `dist_moved` 필드가 이 파일에 추가되는데, 포인트가 이동한 셀 개수를 담고 있습니다. 포인트가 원래 하천 위에 위치한 경우 이 필드의 값은 0 이며, 하천이 최장 거리 내에 없기 때문에 이동하지 않았을 경우 -1 이고, 또는 포인트가 이동한 경우 이동한 셀 개수입니다.

파이썬 코드

알고리즘 ID: `taudem:moveoutletstostreams`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

더글러스-포이커

더글러스-포이커 알고리즘 이 생성한 오르막 만곡 그리드 셀들의 지표 (indicator) 그리드 (1, 0) 을 생성합니다.

이 알고리즘은 먼저 DEM 을 중심, 테두리, 그리고 대각선 방향에 가중치를 부여하는 커널을 통해 평활화합니다. 그 다음 Peuker & Douglas (1975) 방법을 사용해서 오르막 만곡 그리드 셀을 식별합니다. (이 방법은 Band (1986) 에서도 설명하고 있습니다.) 이 기술은 전체 그리드를 표시한 다음, 그리드 셀 4 개로 이루어진 각 사분면을 한번에 (single pass) 평가해서 가장 높은 셀의 표시를 지웁니다. 나머지 표시된 셀들을"오르막 만곡"으로 간주하는데, 이를 살펴보면 수로망과 닮아 있습니다. 이 원시 수로망은 일반적으로 연결성이 부족해서 솎아내기 (thinning) 를 해야 합니다. Band (1986) 에서 이런 문제들을 자세하게 논의하고 있습니다.

파라미터

라벨	명칭	유형	설명
Elevation Grid		[raster]	표고 값을 보유한 그리드를 지정합니다. 일반적으로 이 그리드는 함몰부 제거 도구의 산출물로, 구덩이를 제거한 표고입니다.
Center Smoothing Weight		[number] 기본값: 0.4	이 알고리즘이 오르막 만곡 그리드 셀을 식별하기 전에 커널이 DEM 을 평활화하기 위해 사용하는 중심 가중치 파라미터입니다.

다음 페이지에 계속

표 24.234 – 이전 페이지에서 계속

Side Smoothing Weight		[number] 기본값: 0.1	이 알고리즘이 오르막 만곡 그리드 셀을 식별하기 전에 커널이 DEM 을 평활화하기 위해 사용하는 테두리 가중치 파라미터입니다.
Diagonal Smoothing Weight		[number] 기본값: 0.05	이 알고리즘이 오르막 만곡 그리드 셀을 식별하기 전에 커널이 DEM 을 평활화하기 위해 사용하는 대각선 방향 가중치 파라미터입니다.

산출물

라벨	명칭	유형	설명
Stream Source Grid		[raster]	더글러스-포이커 알고리즘이 생성한 오르막 만곡 그리드 셀들의 지표 (indicator) 그리드 (1, 0) 로, 이를 살펴보면 수로망과 닮아 있습니다. 이 원시 수로망은 일반적으로 연결성이 부족해서 솎아내기 (thinning) 를 해야 합니다. Band (1986) 에서 이런 문제들을 자세하게 논의하고 있습니다.

파이썬 코드

알고리즘 ID: `taudem:peukerdouglas`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

참고

- Band, L. E., (1986), "Topographic partition of watersheds with digital elevation models", Water Resources Research, 22(1): 15-24.
- Peuker, T. K. and D. H. Douglas, (1975), "Detection of surface-specific points by local parallel processing of discrete terrain elevation data", Comput. Graphics Image Process., 4: 375-387.

포이커 더글러스 하천

DEM 곡률 기반 방법을 사용해서 하천의 위치를 식별하는 하천 지표 그리드 (1,0) 를 생성하기 위해"더글러스-포이커 (Douglas-Peuker)", "D8 기여 영역 (D8 Contributing Area)", "하천 낙차 분석 (Stream Drop Analysis)"그리고 "한계값으로 하천 정의 (Stream Definition by Threshold)"도구의 기능을 결합합니다. 이 방법은 먼저 DEM 을 중심, 테두리, 그리고 대각선 방향에 가중치를 부여하는 커널을 통해 평활화합니다. 그 다음 Peuker & Douglas (1975) 방법을 사용해서 오르막 만곡 그리드 셀을 식별합니다. (이 방법은 Band (1986) 에서도 설명하고 있습니다.) 이 기술은 전체 그리드를 표시한 다음, 그리드 셀 4 개로 이루어진 각 사분면을 한번에 (single pass) 평가해서 가장 높은 셀의 표시를 지웁니다. 나머지 표시된 셀들을'오르막 만곡'으로 간주하는데, 이를 살펴보면 수로망과 닮아 있습니다. 이 원시 수로망은 일반적으로 연결성이 부족해서 솎아내기 (thinning) 를 해야 합니다. Band (1986) 에서 이런 문제들을 자세하게 논의하고 있습니다. 여기에서는 이런 오르막 만곡 셀들만을 이용, D8 기여 영역을 계산해서

이 그리드 셀들을 솎아내고 연결합니다. 그 다음 이 셀들의 개수에 대한 누적 한계값을 사용해서 이 한계값을 사용자가 선택적으로 설정하거나, 낙차 분석 (drop analysis) 을 통해 결정하는 하천망을 매핑합니다.

낙차 분석을 이용하는 경우, 누적 한계값을 위한 값을 지정하는 대신"Number"파라미터의 단계 개수를 사용해서 강하 분석 파라미터"Lowest"와"Highest"사이의 범위를 검색, 누적 한계값을 결정합니다. 낙차 분석의 과학적인 방법론을 알고 싶다면 Tarboton, et al. (1991, 1992) 그리고 Tarboton and Ames (2001) 를 참조하십시오. 선택된 누적 한계값의 값은 t-통계량의 절대값이 2 미만일 때 가장 작은 값입니다. 이 값은 낙차 분석 테이블 텍스트 파일에 작성됩니다. 배출구가 지정된 경우에만 낙차 분석을 할 수 있습니다. 전체 그리드 영역을 분석할 경우, 한계값이 변하면서 경계로 배수되는 더 짧은 하천이 한계값 기준을 충족시키지 못 해 분석에서 제외될 수도 있기 때문입니다. 이렇게 되면 배수 밀도를 정의하는 데 문제가 생겨, 서로 상이한 영역들에서 평가된 통계를 비교하는 데 일관성이 떨어지게 됩니다.

파라미터

산출물

라벨	명칭	유형	설명
Stream source		[raster]	더글러스-포이커 알고리즘이 생성한 오르막 만곡 그리드 셀들의 지표 (indicator) 그리드 (1, 0) 로, 이를 살펴보면 수로망과 닮아 있습니다. 이 원시 수로망은 일반적으로 연결성이 부족해서 솎아내기 (thinning) 를 해야 합니다. Band (1986) 에서 이런 문제들을 자세하게 논의하고 있습니다.

파이썬 코드

알고리즘 ID: `taudem:peukerdouglasstreamdef`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

경사-면적 조합

경사, 특정 집수 영역 그리드 입력물, 그리고 `m` 과 `n` 파라미터를 바탕으로 하는 경사-면적 값 (`Sm`) (`An`) 을 담고 있는 그리드를 생성합니다. 이 알고리즘은 경사-면적 하천 래스터 묘사 방법의 일부로 사용하기 위해 개발됐습니다.

파라미터

라벨	명칭	유형	설명
Slope Grid		[raster]	경사 값을 가진 입력 그리드입니다. 이 그리드는 고저차를 거리로 나눈 값으로 경사를 측정하며, **"D-Infinity 유수 방향"**알고리즘의 산출물입니다.

다음 페이지에 계속

표 24.237 – 이전 페이지에서 계속

Contributing Area Grid		[raster]	각 셀의 고유 기여에 해당 셀로 흘러드는 오르막 이웃들의 기여를 더한 각 셀 별 집수 영역 값을 제공하는 그리드입니다. 이 그리드는 **"D-Infinity 기여 영역"**알고리즘을 사용해서 얻을 수 있습니다.
Slope Exponent		[number] 기본값: 2	경사-면적 그리드를 생성하는 `(Sm)(An)` 공식에 사용되는 경사 지수 (exponent) `m` 파라미터입니다.
Area Exponent		[number] 기본값: 1	경사-면적 그리드를 생성하는 `(Sm)(An)` 공식에 사용되는 면적 지수 (exponent) `n` 파라미터입니다.

산출물

라벨	명칭	유형	설명
Slope Area Grid		[raster]	경사 그리드, 특정 집수 영역 그리드, 경사 지수 파라미터 `m`, 그리고 면적 지수 파라미터 `n` 으로 계산한 경사-면적 값 `(Sm) (An)` 을 담고 있는 그리드입니다.

파이썬 코드

알고리즘 ID: `taudem:slopearea`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

경사 면적 하천 정의

경사, 특정 집수 영역 그리드 입력물, 그리고 `m` 과 `n` 파라미터를 바탕으로 하는 경사-면적 값 `(Sm) (An)` 을 담고 있는 그리드를 생성합니다. 이 알고리즘은 경사-면적 하천 래스터 묘사 방법의 일부로 사용하기 위해 개발됐습니다.

파라미터

라벨	명칭	유형	설명
D8 flow directions		[raster]	
D-infinity Contributing Area		[raster]	각 셀의 고유 기여에 해당 셀로 흘러드는 오르막 이웃들의 기여를 더한 각 셀 별 집수 영역 값을 제공하는 그리드입니다. 이 그리드는 **"D-Infinity 기여 영역"**알고리즘을 사용해서 얻을 수 있습니다.
Slope		[raster]	경사 값을 가진 입력 그리드입니다. 이 그리드는 고저차를 거리로 나눈 값으로 경사를 측정하며, **"D-Infinity 유수 방향"**알고리즘의 산출물입니다.
Mask grid		[raster]	
Outlets		[vector: point]	

다음 페이지에 계속

표 24.239 – 이전 페이지에서 계속

Pit-filled grid for drop analysis		[raster]	
D8 contributing area for drop analysis		[raster]	
Slope Exponent		[number] 기본값: 2	경사-면적 그리드를 생성하는 `(Sm)(An)` 공식에 사용되는 경사 지수 (exponent) `m` 파라미터입니다.
Area Exponent		[number] 기본값: 1	경사-면적 그리드를 생성하는 `(Sm)(An)` 공식에 사용되는 면적 지수 (exponent) `n` 파라미터입니다.
Accumulation threshold		[number]	
Minimum threshold		[number]	
Maximum threshold		[number]	
Number of drop thresholds		[number]	
Type of threshold step		[enumeration] 기본값: 0	옵션: • 0 –로그 • 1 –선형
Check for edge contamination		[boolean]	
Select threshold by drop analysis		[boolean]	

산출물

라벨	명칭	유형	설명
Stream raster		[raster]	
Slope area		[raster]	경사 그리드, 특정 집수 영역 그리드, 경사 지수 파라미터 `m`, 그리고 면적 지수 파라미터 `n` 으로 계산한 경사-면적 값 `(Sm) (An)` 을 담고 있는 그리드입니다.
Maximum upslope		[raster]	
Drop analysis		[file]	

파이썬 코드

알고리즘 ID: `taudem:slopeareastreamdef`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

한계값으로 하천 정의

어떤 그리드를 입력받아 입력값이 한계값 이상인 셀들을 식별하는 지표 (indicator) 그리드 (1, 0) 을 산출합니다. 일반적으로 집적원 (accumulated source) 영역 그리드를 입력받아 하천 래스터 그리드를 생성합니다. 부가적인 입력 마스크 그리드를 사용하는 경우, 마스크 값이 0 이상인 셀 영역만 평가하도록 제한합니다. D-infinity 기여 영역 그리드 (`*.sca`) 를 마스크 그리드로 사용하는 경우, 경계 오염 마스크로 기능합니다. 한계값 논리는 다음과 같습니다:

```
src = ((ssa >= thresh) & (mask >= s0)) ? 1:0
```

파라미터

라벨	명칭	유형	설명
Accumulated Stream Source Grid		[raster]	이 그리드는 일반적으로 유역 (watershed) 의 일부 특성 또는 특성들의 조합을 집적합니다. 정확한 특성 (들) 은 사용되는 하천망 래스터 알고리즘에 따라 달라집니다. 이 그리드는 산출되는 하천망이 연속하도록 그리드 셀 값이 내리막 D8 유수 방향을 따라 점증하는 속성을 보유해야 합니다. 집적 알고리즘이 이 그리드를 주로 산출하긴 해도, 최장 오르막 함수 같은 다른 알고리즘도 이런 그리드를 생성할 수 있습니다.
Threshold		[number] 기본값: 100	어느 셀을 하천 셀로 간주해야 하는지 여부를 결정하기 위해 이 파라미터와 집적 하천원 (Accumulated Stream Source) 그리드 (`*ssa`) 내부의 값을 비교합니다. `*ssa` 값이 이 한계값 이상인 그리드 셀을 하천으로 식별합니다.
Mask Grid 부가적		[raster]	이 부가적인 입력물은 관심 영역을 마스크하기 위한 그리드로, 산출물은 이 그리드에서 0 이상의 값을 가진 영역만 커버합니다. 일반적으로 D-Infinity 기여 영역 그리드를 마스크로 사용해서 하천망을 D-Infinity 기여 영역 안으로 제한해서 묘사합니다. 이는 경계 오염 마스크의 기능과 동일합니다.

산출물

라벨	명칭	유형	설명
Stream Raster Grid		[raster]	이 입력물은 하천인 경우 그리드 셀 값을 1, 나머지 셀 값을 0 으로 해서 하천의 위치를 나타내는 지표 (indicator) 그리드 (1, 0) 입니다.

파이썬 코드

알고리즘 ID: `taudem:threshold`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

낙차 분석으로 하천 정의

"하천 낙차 분석 (Stream Drop Analysis)"그리고"한계값으로 하천 정의 (Stream Definition by Threshold)"도구의 기능을 결합합니다. 이 방법은 (입력 파라미터로 결정된) 일련의 한계값을 입력 누적 하천 소스 그리드 (`ssa`) 에 적용해서 하천 낙차 통계 테이블 (`drp.txt`) 을 산출합니다. 그 다음 하천 셀들의 지표 (1,0) 그리드인 하천 래스터 그리드를 산출합니다. 누적 하천 소스 값이 하천 낙차 통계로 결정된 최적 한계값 이상인 셀을 하천 셀로 정의합니다. `*sca` 파일을 경계 오염 마스크로 이용하는 기능을 복제하기 위해 마스크 입력물을 포함하는 옵션이 있습니다. 한계값 논리는 다음과 같아야 합니다: `src = ((ssa >= thresh) & (mask >=0)) ? 1:0`

파라미터

산출물

파이썬 코드

알고리즘 ID: `taudem:streamdefdropanalysis`

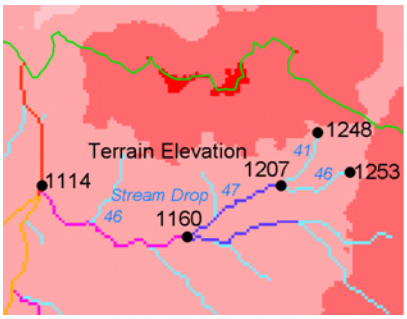
```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

하천 낙차 분석

입력 집적 하천원 그리드 (`*ssa`) 에 (입력 파라미터로 결정된) 일련의 한계값을 적용해서 그 결과를 하천 낙차 통계 테이블 `*drp.txt` 파일로 산출합니다. 이 알고리즘은 하천을 묘사하는 데 사용되는 지형학적으로 객관적인 한계값을 결정하기 위해 개발됐습니다. 낙차 분석은 한계값들의 범위에서 하천망을 평가하고, 산출되는 스트랄러 하천의 일관적인 낙차 속성을 조사해서 올바른 한계값을 자동적으로 선택하려 합니다. 근본적으로 낙차 분석은 1 차 하천의 평균 하천 낙차가 고차 하천의 평균 하천 낙차와 통계적으로 다른지를 T-검증을 사용해서 판정합니다. T-검증으로 두드러진 차이가 드러나는 경우 하천망이 이"법칙"을 따르지 않는 것이기 때문에 더 큰 한계값을 선택해야 합니다. T-검증으로 두드러진 차이가 드러나지 않는 최소 한계값을 선택하면 지형학적으로 일관적인 하천 낙차"법칙"을 따르는 최고 해상도의 하천망을 산출하며, 또한 이 최소 한계값은 DEM 에서 하천의"객관적인" 또는 자동적인 매핑을 위해 선택되는 한계값이기도 합니다. 하천망 래스터를 구축하는 데 이 알고리즘을 사용할 수 있습니다. 이때 집적 하천망 그리드에서 집적된 정확한 유역 특성 (들) 은 하천망 래스터 구축에 사용되는 메소드에 따라 달라집니다.

Broscoe (1959) 에서 일관적인 하천 낙차"법칙"을 식별했습니다. 하천 묘사 한계값을 결정하기 위해 이 법칙을 사용하는 데 쓰이는 과학에 대해 알고 싶다면, Tarboton et al. (1991, 1992), Tarboton & Ames (2001) 을 참조하십시오.



파라미터

라벨	명칭	유형	설명
D8 Contributing Area Grid		[raster]	D8 알고리즘을 사용해서 계산한 각 셀 별 기여 영역 값 그리드입니다. 어느 셀의 기여 영역 값은 해당 셀 고유의 기여와 해당 셀로 흘러드는 모든 오르막 이웃에서 나오는 기여를 더한 값으로, 셀 개수로 측정됩니다. 일반적으로 이 그리드는 **"D8 기여 영역"** 알고리즘의 산출물입니다. 하천 낙차 테이블에 보고된 배수 농도를 평가하는 데 이 그리드를 사용합니다.
D8 Flow Direction Grid		[raster]	각 셀 별로 그 인접 또는 대각선 방향의 이웃 8 개 셀 가운데 가장 심한 내리막 경사를 가지고 있는 셀의 방향으로 정의한 D8 유수 방향 그리드입니다. 이 그리드는 **"D8 유수 방향"**알고리즘의 산출물로 얻을 수 있습니다.
Pit Filled Elevation Grid		[raster]	표고 값을 보유한 그리드를 지정합니다. 일반적으로 이 그리드는 함몰부 제거 도구의 산출물로, 구덩이를 제거한 표고입니다.
Accumulated Stream Source Grid		[raster]	이 그리드는 내리막 D8 유수 방향을 따라 점증해야만 합니다. 하천의 시작을 결정하기 위해 이 그리드와 일련의 한계값을 비교합니다. 이 그리드는 **"D8 기여 영역"**알고리즘으로, 또는 **"D8 유수 경로 극값"** 알고리즘의 최대값 옵션을 사용해서 유역의 일부 특성들 또는 특성들의 조합을 집적해서 생성하는 경우가 많습니다. 정확한 방법은 사용하는 알고리즘에 따라 달라집니다.
Outlets Shapefile		[vector: point]	낙차 분석 수행 대상인 배출 상류를 정의하는 포인트 shapefile 입니다.
Minimum Threshold		[number] 기본값: 5	이 파라미터는 낙차 분석으로 가능한 한계값을 탐색하는 범위의 가장 낮은 값입니다. 이 기술은 t-통계의 절대값이 2 미만인 범위에 있는 최소 한계값을 찾습니다. 낙차 분석에 쓰이는 과학에 대해 알고 싶다면 Tarboton et al. (1991, 1992), Tarboton & Ames (2001) 을 참조하십시오.
Maximum Threshold		[number] 기본값: 500	이 파라미터는 낙차 분석으로 가능한 한계값을 탐색하는 범위의 가장 높은 값입니다. 이 기술은 t-통계의 절대값이 2 미만인 범위에 있는 최대 한계값을 찾습니다. 낙차 분석에 쓰이는 과학에 대해 알고 싶다면 Tarboton et al. (1991, 1992), Tarboton & Ames (2001) 을 참조하십시오.

다음 페이지에 계속

표 24.243 – 이전 페이지에서 계속

Number of Threshold Values		[number] 기본값: 10	이 파라미터는 낙차 분석으로 가능한 한계값을 탐색하는 범위를 나누기 위한 단계의 개수입니다. 이 기술은 t-통계의 절대값이 2 미만인 범위에 있는 최소 한계값을 찾습니다. 낙차 분석에 쓰이는 과학에 대해 알고 싶다면 Tarboton et al. (1991, 1992), Tarboton & Ames (2001) 을 참조하십시오.
Spacing for Threshold Values		[enumeration] 기본값: 0	이 파라미터는 낙차 분석으로 가능한 한계값을 탐색할 때 로그 간격 (logarithmic spacing) 을 사용해야 할지 또는 선형 간격을 사용해야 할지를 나타내는 파라미터입니다. 옵션: • 0 –로그 • 1 –선형

산출물

라벨	명칭	유형	설명

D-Infinity Drop to Stream Grid		[file]	다음 헤더 줄을 보유한 쉼표 구분 텍스트 파일입니다:
			`Threshold,DrainDen,NoFirstOrd,NoHighOrd,`
			`↪MeanDFirstOrd,MeanDHighOrd,`
			`↪StdDevFirstOrd,StdDevHighOrd,T`
			이 파일은 조사한 각 한계값 별 데이터 한 줄, 그 다음으로 최적 한계값을 나타내는 요약 줄을 담고 있습니다. 이 기술은 t-통계의 절대값이 2 미만인 범위에 있는 최소 한계값을 찾습니다. 낙차 분석에 쓰이는 과학에 대해 알고 싶다면 Tarboton et al. (1991, 1992), Tarboton & Ames (2001) 을 참조하십시오.

파이썬 코드

알고리즘 ID: `taudem:dropanalysis`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

참고

- Broscoe, A. J., (1959), "Quantitative analysis of longitudinal stream profiles of small watersheds", Office of Naval Research, Project NR 389-042, Technical Report No. 18, Department of Geology, Columbia University, New York.
- Tarboton, D. G., R. L. Bras and I. Rodriguez-Iturbe, (1991), "On the Extraction of Channel Networks from Digital Elevation Data", Hydrologic Processes, 5(1): 81-100.
- Tarboton, D. G., R. L. Bras and I. Rodriguez-Iturbe, (1992), "A Physical Basis for Drainage Density", Geomorphology, 5(1/2): 59-76.
- Tarboton, D. G. and D. P. Ames, (2001), "Advances in the mapping of flow networks from digital elevation data", World Water and Environmental Resources Congress, Orlando, Florida, May 20-24, ASCE, https://www.researchgate.net/publication/2329568_Advances_in_the_Mapping_of_Flow_Networks_From_Digital_Elevation_Data.

하천 범위 및 유역

이 알고리즘은 하천 래스터 그리드로부터 벡터망 및 shapefile 을 생성합니다. 하천 래스터를 따라 유수 경로를 연결하기 위해 유수 방향 그리드를 사용합니다. 각 하천 구간의 스트랄러 순서를 계산합니다. 각 하천 구간 (직선 유역) 으로 흘러드는 소유역 (subwatershed) 도 묘사하며, 하천 직선 유역 (Stream Reach) shapefile 안에 있는 WSNO(유역 번호) 에 대응하는 값 식별자를 라벨로 표시합니다.

이 알고리즘은 하천망을 스트랄러 순서 체계에 따라 순서를 부여합니다. 다른 어떤 하천도 흘러들지 않는 하천이 1 차 (order 1) 입니다. 서로 다른 순서의 하천 구간 2 개가 합쳐질 경우 그 하류 구간의 순서는 해당 하천 구간들 가운데 더 높은 순서를 따릅니다. 동일한 순서의 하천 구간 2 개가 합쳐질 경우 그 하류 구간의 순서는 1 만큼 증가합니다. 하천 구간 2 개 이상이 합쳐질 경우 유수 경로 순서는 가장 높은 구간 순서의 최대값, 또는 두 번째로 높은 구간 순서에 1 을 더한 값으로 계산됩니다. 어느 한 지점에서 2 개 이상의 구간이 합쳐지는 경우 스트랄러 순서는 일반적으로 이렇게 정의됩니다. 하천망의 지형적 연결성은 하천망 트리 파일에 저장되며, 하천망을 따라 각 그리드 셀의 좌표 및 속성은 하천망 좌표 파일에 저장됩니다.

이 알고리즘은 하천 래스터 그리드를 하천망의 소스로 사용하며, 유수 방향 그리드를 하천망 내부의 연결점을 추적하는 데 사용합니다. 하천망 좌표 파일 안에 표고 및 기여 영역 속성을 결정하는 데 표고 그리드 및 기여 영역 그리드를 사용합니다. 관측 지점의 상류 및 하류 유역을 쉽게 표현하기 위해 하천 구간을 논리적으로 나누는 데 배출 shapefile 에 있는 포인트들을 사용합니다. 이 알고리즘은 배출 shapefile 에 있는 `id` 속성 필드를 하천망 트리 파일의 식별자로 사용합니다. 그 다음 하천망 트리 및 좌표 파일에 있는 벡터망 표현 텍스트를 shapefile 로 변환합니다. 더 많은 속성들도 평가합니다. 이 알고리즘은 산출 유역 그리드에 하천망으로 흘러드는 전체 영역을 단일 값으로 표현해서 단일 유역을 묘사하는 옵션도 가지고 있습니다.

파라미터

라벨	명칭	유형	설명
Pit Filled Elevation Grid		[raster]	표고 값을 보유한 그리드를 지정합니다. 일반적으로 이 그리드는 함몰부 제거 도구의 산출물로, 구덩이를 제거한 표고입니다.
D8 Flow Direction Grid		[raster]	각 셀 별로 그 인접 또는 대각선 방향의 이웃 8 개 셀 가운데 가장 심한 내리막 경사를 가지고 있는 셀의 방향으로 정의한 D8 유수 방향 그리드입니다. 이 그리드는 **"D8 유수 방향"**알고리즘의 산출물로 얻을 수 있습니다.

다음 페이지에 계속

표 24.245 – 이전 페이지에서 계속

D8 Drainage Area		[raster]	D8 알고리즘을 사용해서 해당 셀 고유의 기여와 해당 셀로 흘러드는 모든 오르막 이웃에서 나오는 기여를 더한 값을 각 셀 별 그리드 셀 개수 (또는 가중치의 총체) 라는 측면에서 계산한 기여 영역 값을 제공하는 그리드입니다. 일반적으로 이 그리드는 **"D8 기여 영역"**알고리즘의 산출물입니다. 하천망 좌표 파일의 기여 영역 속성을 결정하기 위해 이 그리드를 사용합니다.
Stream Raster Grid		[raster]	하천인 경우 그리드 셀 값을 1, 나머지 셀 값을 0 으로 해서 하천을 나타내는 지표 (indicator) 그리드입니다. **"하천망 분석"**도구 모음 가운데 하나로 이런 그리드를 생성할 수 있습니다. 하천 래스터 그리드를 하천망의 소스로 사용합니다.
Outlets Shapefile as Network Nodes 부가적		[vector: point]	관심 지점을 정의하는 포인트 shapefile 입니다. 이 파일을 사용하는 경우, 알고리즘이 이 배출 지점 상류의 하천망만을 묘사할 것입니다. 또한, 관측 지점의 상류 및 하류 유역을 쉽게 표현하기 위해 하천 구간을 논리적으로 나누는 데 배출 shapefile 에 있는 포인트들을 사용합니다. 이 알고리즘은 배출 shapefile 이 정수형 속성 필드 `id` 를 가지고 있을 것을 요구합니다. 하천망 트리 파일이 `id` 값을 식별자로 사용하기 때문입니다.
Delineate Single Watershed		[boolean] 기본값: True	산출 유역 그리드에 하천망으로 흘러드는 전체 영역을 단일 값으로 표현해서 단일 유역을 묘사하도록 하는 옵션입니다. 이 옵션을 거짓으로 설정하면 각 하천 구간 별로 개별 유역을 묘사합니다. 기본값은 `False` (개별 유역) 입니다.

산출물

라벨	명칭	유형	설명
Stream Order Grid		[raster]	하천 순서 그리드는 스트랄러 순서 체계에 따라 순서를 부여한 하천 셀 값을 담고 있습니다. 스트랄러 순서 체계는 다른 어떤 하천도 흘러들지 않는 하천 구간을 1 차 (order 1) 로 정의합니다. 서로 다른 순서의 하천 구간 2 개가 합쳐질 경우 그 하류 구간의 순서는 해당 하천 구간들 가운데 더 높은 순서를 따릅니다. 동일한 순서의 하천 구간 2 개가 합쳐질 경우 그 하류 구간의 순서는 1 만큼 증가합니다. 하천 구간 2 개 이상이 합쳐질 경우 유수 경로 순서는 가장 높은 구간 순서의 최대값, 또는 두 번째로 높은 구간 순서에 1 을 더한 값으로 계산됩니다. 어느 한 지점에서 2 개 이상의 구간이 합쳐지는 경우 스트랄러 순서는 일반적으로 이렇게 정의됩니다.
Watershed Grid		[raster]	이 산출 그리드는 각 구간 유역을 유일 ID 번호로 식별하거나, 또는 단일 유역 옵션을 참으로 설정한 경우 하천망으로 흘러드는 전체 영역을 단일 ID 로 식별합니다.

다음 페이지에 계속

표 24.246 – 이전 페이지에서 계속

Stream Reach Shapefile		[vector: line]	이 산출물은 하천망 내부의 링크를 지정하는 폴리라인 shapefile 입니다. 속성 테이블에 있는 열들은 다음과 같습니다: • LINKNO – 링크 번호입니다. 각 링크 (합류점 사이의 수로 구간) 에 관련된 유일한 숫자입니다. 랜덤한 숫자로, 공간 처리 횟수에 따라 달라집니다. • DSLINKNO –하류 링크의 링크 번호입니다. `-1` 은 하류 링크가 존재하지 않는다는 뜻입니다. • USLINKNO1 – 첫 번째 상류 링크의 링크 번호입니다. (예를 들어 소스 링크의 경우처럼, `-1` 은 상류 링크가 존재하지 않는다는 뜻입니다.) • USLINKNO2 – 두 번째 상류 링크의 링크 번호입니다. (예를 들어 소스 링크의 경우 또는 하천 구간이 논리적으로 나누어지지만 하천망은 두 갈래로 나뉘지 않는 내부 관측 지점의 경우처럼, `-1` 은 두 번째 상류 링크가 존재하지 않는다는 뜻입니다.) • DSNODEID –하천 구간의 하류 종단점에 있는 노드의 노드 식별자입니다. 이 식별자는 배출 shapefile 에서 노드를 지정하기 위해 사용되는 `id` 속성 필드와 대응합니다. • Order –스트랄러 하천 순서 • Length – 링크의 길이입니다. 기저 DEM 그리드의 수평 맵 단위를 사용합니다. • Magnitude –링크의 슈리브 규모 (Shreve Magnitude) 입니다. 상류 소스의 총 개수를 뜻합니다. • DS_Cont_Ar – 링크의 하류 종단점에 있는 배수 영역입니다. 일반적으로 이 영역은 하류 종단점의 상류 그리드 셀 1 개인데, 왜냐하면 하류 종단점 그리드 셀에 있는 배수 영역이 하천이 합류하는 영역을 포함하기 때문입니다. • Drop –링크의 시작점에서 종단점 사이의 표고 낙차입니다. • Slope –링크의 평균 경사입니다. (낙차를 길이로 나눈 값입니다.) • Straight_L – 링크의 시작점에서 종단점까지의 직선 거리입니다. • US_Cont_Ar –링크의 상류 종단점에 있는 배수 영역입니다. • WSNO – 유역 번호입니다. 링크로 직접 흘러드는 유역의 식별 번호를 지정하는 `*w.shp` 파일 및 `*w` 그리드 파일을 교차 참조합니다. • DOUT_END – 링크의 하류 종단점에서 (예를 들어 하천망에서 가장 하류 지점인) 최종 배출 지점까지의 거리입니다. • DOUT_START –링크의 상류 종단점에서 최종 배출 지점까지의 거리입니다. • DOUT_MID –링크의 중앙 지점에서 최종 배출 지점까지의 거리입니다.

다음 페이지에 계속

표 24.246 – 이전 페이지에서 계속

Network Connectivity Tree		[file]	이 산출물은 하천망 트리 파일에 저장된 하천망의 지형적 연결성을 상세히 설명하는 텍스트 파일입니다. 다음과 같은 열들을 담고 있습니다: • 링크 번호 (랜덤 – 공간 처리 횟수에 따라 달라집니다.) • 하천망 좌표 파일 (`*coord.dat`) 에서 시작 지점 번호 (인덱스가 0 부터 시작합니다.) • 하천망 좌표 파일 (`*coord.dat`) 에서 종단 지점 번호 (인덱스가 0 부터 시작합니다.) • 다음 (하류) 링크 번호 (링크 번호를 가리킵니다. 예를 들어 최종 링크의 경우처럼, `-1` 은 상류 링크가 존재하지 않는다는 뜻입니다.) • 첫 번째 이전 (상류) 링크 번호. (링크 번호를 가리킵니다. `-1` 은 상류 링크가 존재하지 않는다는 뜻입니다.) • 두 번째 이전 (상류) 링크 번호. (링크 번호를 가리킵니다. `-1` 은 상류 링크가 존재하지 않는다는 뜻입니다. 이전 링크 하나만 `-1` 인 경우, 하천 구간이 논리적으로 나누어지지만 하천망은 두 갈래로 나뉘지 않는다는 뜻입니다.) • 링크의 스트랄러 순서 • 링크의 하류 종단점에 있는 관측 지점의 식별자 (`-1` 은 하류 종단점이 관측 지점이 아니라는 뜻입니다.) • 링크의 하천망 규모 (슈리브의 정의에 따라 상류 소스의 개수로 계산합니다.)
Network Coordinates		[file]	이 산출물은 하천망을 따라 각 포인트의 좌표 및 속성을 담고 있는 텍스트 파일입니다. 다음과 같은 열들을 담고 있습니다: • X 좌표 • Y 좌표 • 최종 링크의 하류 종단점까지의 수로 거리 • 표고 • 기여 영역

파이썬 코드

알고리즘 ID: `taudem:streamnet`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

공간 처리 툴박스에 있는 알고리즘 위에 마우스를 가져가면 알고리즘 *ID* 를 표시합니다. 파라미터 목록 *(dictionary)* 은 파라미터 **명칭** 및 값을 제공합니다. 파이썬 콘솔에서 공간 처리 알고리즘을 어떻게 실행하는지 자세히 알고 싶다면콘솔에서 공간 처리 알고리즘 사용 을 참조하세요.

24.5 OTB 응용 프로그램 제공자

OTB (Orfeo ToolBox) 는 원격탐사 (remote sensing) 데이터용 영상 처리 라이브러리입니다. OTB 는 영상 처리 기능을 사용할 수 있는 응용 프로그램도 제공합니다. OTB 쿡북 에서 응용 프로그램 목록과 그 문서들을 볼 수 있습니다.

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.

참고: 플러그인 관리자 설정 에 따르면, QGIS 주 인터페이스는 설치된 플러그인용 업데이트 또는 새 플러그인을 사용할 수 있는 경우 상태 바 오른쪽에 그 사실을 알려주는 아이콘을 표시할 수 있습니다.

25.1.2 플러그인 대화창

사용자는 플러그인 대화창의 메뉴를 통해 플러그인을 서로 다른 방식으로 설치, 설치 제거, 업그레이드할 수 있습니다. 각 플러그인은 오른쪽 패널에 표시되는 다음 메타데이터를 보유하고 있습니다:

- 실험적인 플러그인인지 알려주는 정보
- 설명
- 평점 (사용자가 선호하는 플러그인을 평가할 수 있습니다!)
- 태그
- 홈페이지, 트래커 및 코드 저장소를 가리키는 유용한 몇몇 링크
- 작성자 (들)
- 사용할 수 있는 버전

대화창 상단에 있는 *Search* 기능으로 메타데이터 정보 (작성자, 명칭, 설명 등등) 를 사용해서 어떤 플러그인이라도 찾을 수 있습니다. ( *Settings* 탭을 제외한) 거의 모든 탭에서 검색 기능을 사용할 수 있습니다.

설정 탭

사용자는  *Settings* 탭에서 사용자 응용 프로그램에 어떤 플러그인을 표시할 수 있는지 환경 설정할 수 있습니다. 다음 옵션들을 사용할 수 있습니다:

-  *Check for updates on startup*: 새 플러그인이나 플러그인 업데이트를 사용할 수 있을 때마다, QGIS 가 사용자에게 알려줄 것입니다. 'QGIS 를 시작할 때마다', '하루에 한 번', '사흘에 한 번', '1 주일에 한 번', '2 주일에 한 번', '1 달에 한 번'알려주도록 선택할 수 있습니다.
-  *Show also experimental plugins*: QGIS 가 보통 생산적으로 활용하기엔 적합하지 않은 개발 초기 단계에 있는 플러그인을 표시할 것입니다.
-  *Show also deprecated plugins*: 중요도가 떨어져 더 이상 사용되지 않고 앞으로는 사라지게 될 플러그인은 QGIS 에서 더 이상 사용할 수 없는 함수를 사용하기 때문에, 이런 플러그인을'deprecated'로 분류합니다. 이런 플러그인은 보통 생산적으로 활용하기엔 적합하지 않습니다. 이 옵션을 체크하면 이런 플러그인도 인식 불가능한 플러그인 목록에 표시합니다.

QGIS 는 기본적으로 (QGIS 3.0 의 경우) *Plugin repositories* 부분에 https://plugins.qgis.org/plugins/plugins.xml?qgis=3.0 이라는 URL 과 함께 공식 플러그인 저장소를 가지고 있습니다. 외부 작성자 저장소를 추가하려면, *Add*…버튼을 클릭하고 *Repository Details* 양식에 명칭과 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.

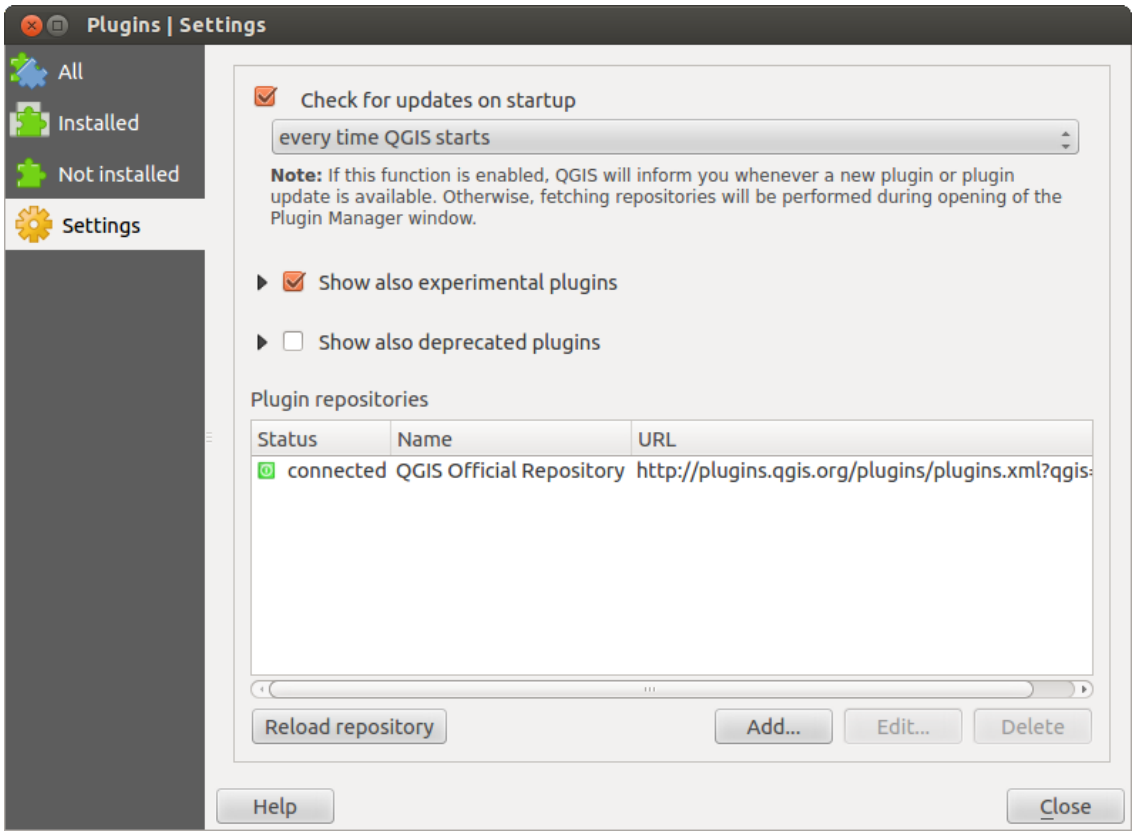


그림 25.1: *Settings* 탭

전체 탭

All 탭은 핵심 및 외부 플러그인 둘 다를 포함한 사용할 수 있는 모든 플러그인의 목록을 담고 있습니다. 플러그인의 새 버전을 찾으려면 *Upgrade All* 버튼을 클릭하십시오. 또, 플러그인이 목록에 있지만 설치돼 있지 않은 경우 *Install Plugin* 버튼을 클릭하면 됩니다. 이미 설치돼 있는 경우 *Uninstall Plugin* 버튼을 클릭한 다음 *Reinstall Plugin* 버튼을 클릭하십시오. 설치한 플러그인은 체크박스를 사용해서 일시적으로 비/활성화시킬 수 있습니다.

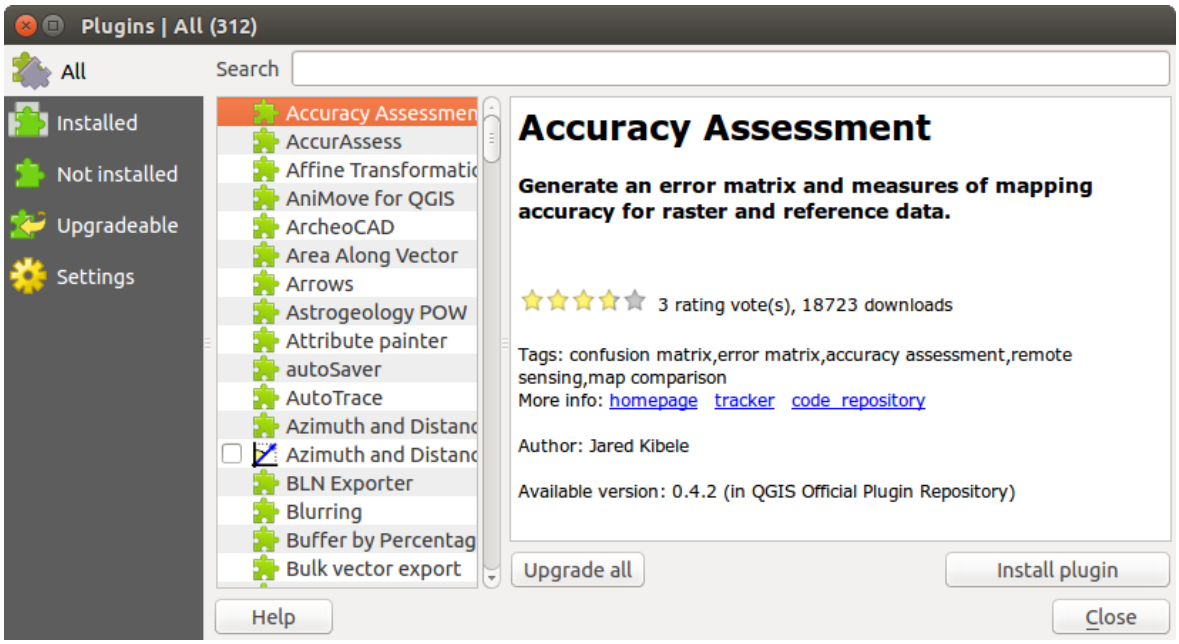


그림 25.2: *All* 탭

설치됨 탭

Installed 탭은 사용자가 설치 제거할 수 없는 핵심 플러그인 목록을 담고 있습니다. *Uninstall Plugin* 과 *Reinstall Plugin* 버튼을 사용해서 언제나 외부 플러그인을 삭제하고 재설치해서 이 목록을 늘릴 수 있습니다. 이 탭에서도 *Upgrade All* 버튼을 사용할 수 있습니다.

설치되지 않음 탭

Not installed 탭은 설치되지 않았지만 사용할 수 있는 모든 플러그인의 목록을 담고 있습니다. *Install Plugin* 버튼을 사용해서 QGIS 에 플러그인을 구현할 수 있습니다.

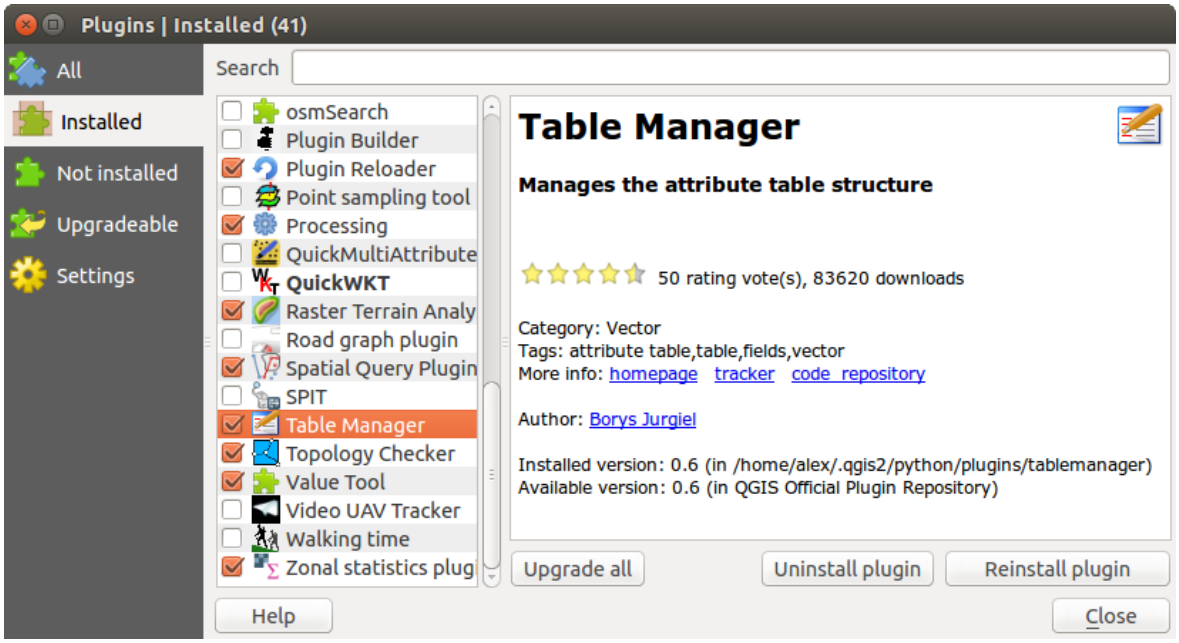


그림 25.3: *Installed* 탭



그림 25.4: *Not installed* 탭

업그레이드 가능 및 신규 탭

새 플러그인이 저장소에 추가되거나 또는 설치한 플러그인의 새 버전이 배포된 경우 *Upgradeable* 및 *New* 탭이 활성화됩니다. *Settings* 메뉴에서 *Show also experimental plugins* 옵션을 체크한 경우, 실험적인 플러그인도 목록에 표시돼 사용자가 새로 나올 도구를 미리 테스트해볼 수 있습니다.

Install Plugin, *Upgrade Plugin* 또는 *Upgrade All* 버튼을 사용해서 플러그인을 설치할 수 있습니다.

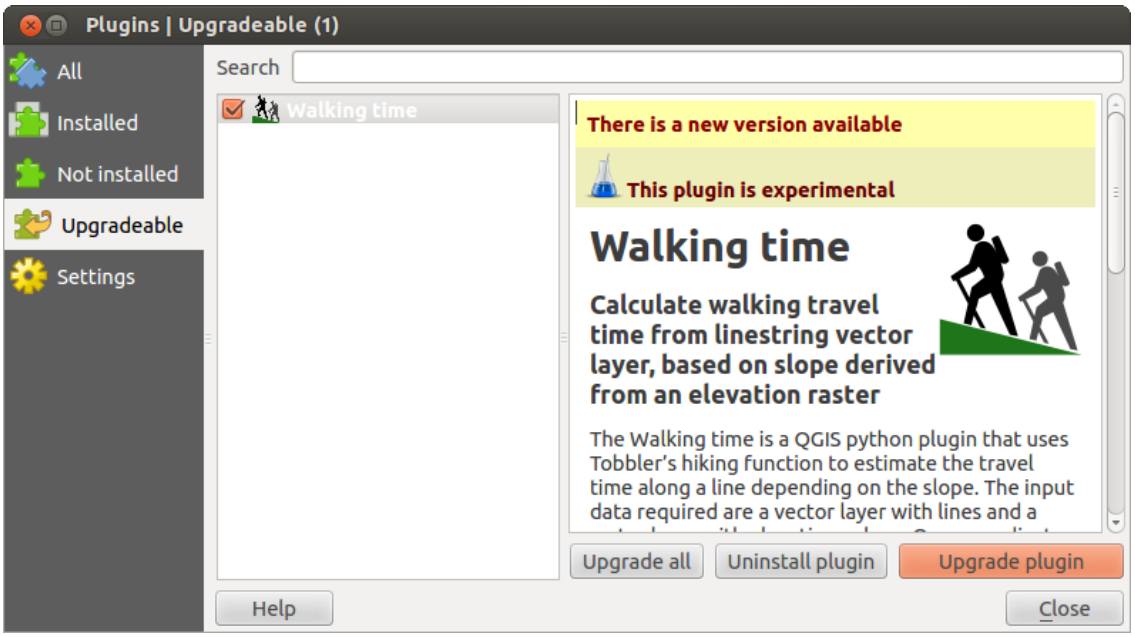


그림 25.5: *Upgradeable* 탭

인식 불가능 탭

Invalid 탭은 설치돼 있지만 어떤 이유로든 (의존성 상실, 불러오는 중 오류 발생, 현재 QGIS 버전과 호환되지 않는 함수 등등) 현재 작동하지 않는 모든 플러그인의 목록을 담고 있습니다. *Reinstall Plugin* 버튼을 클릭해서 인식 불가능해진 플러그인을 고쳐볼 수도 있지만, 다른 방법으로 (새 라이브러리 설치, 호환되는 또다른 플러그인 탐색, 또는 망가진 플러그인을 업그레이드해서) 문제를 해결해야 하는 경우가 대부분입니다.

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.

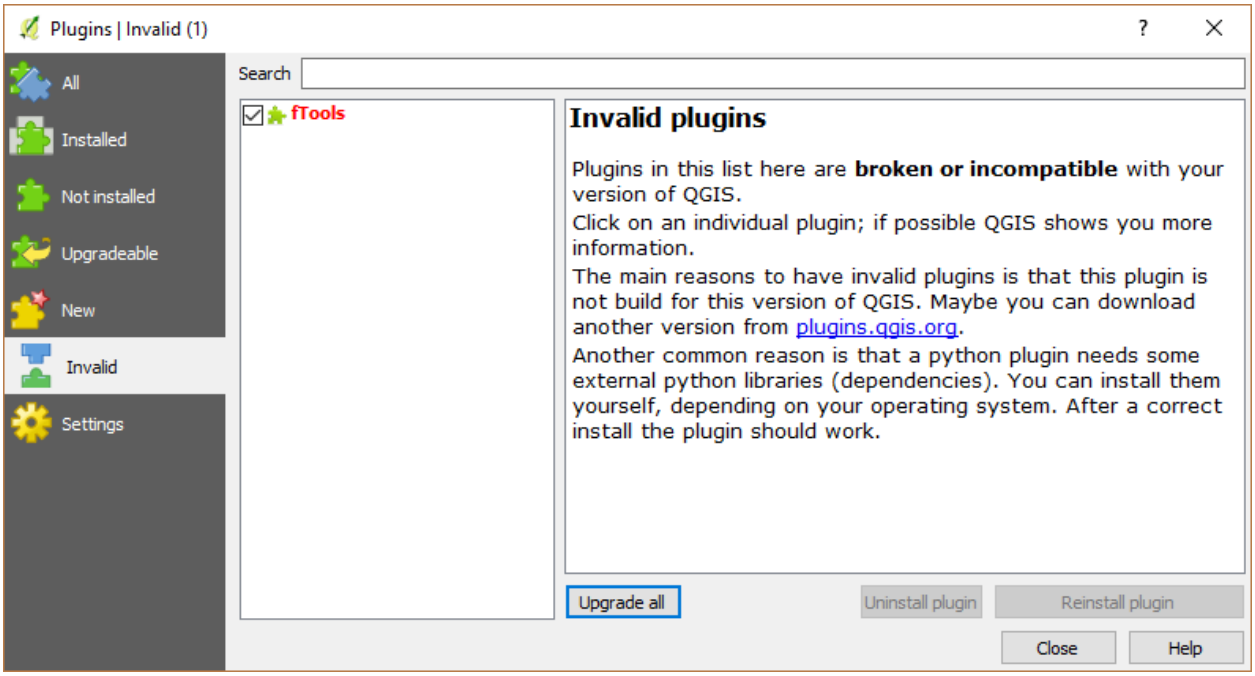


그림 25.6: *Invalid* 탭

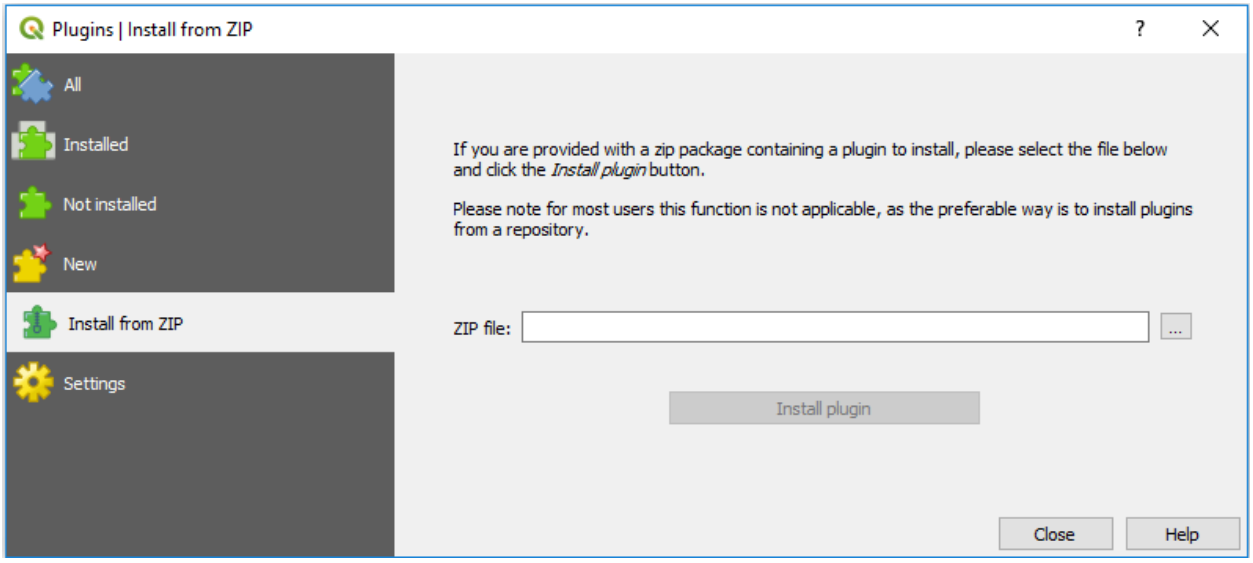


그림 25.7: *Install from zip* 탭

25.2 QGIS 핵심 플러그인 사용하기

25.2.1 데이터베이스 관리자 플러그인

DB 관리자 플러그인은 단일 사용자 인터페이스 내에서 QGIS 가 지원하는 공간 데이터베이스 포맷 (PostGIS, SpatiaLite, GeoPackage, Oracle Spatial, 가상 레이어) 을 통합하고 관리하기 위한 주요 도구로 개발됐습니다. DB Manager 플러그인은 여러 기능을 제공합니다. QGIS 탐색기에서 DB 관리자로 레이어를 드래그하면, 사용자 공간 데이터베이스로 해당 레이어를 가져올 것입니다. 공간 데이터베이스들 사이에서 테이블을 드래그 & 드롭하면 해당 테이블을 가져올 것입니다

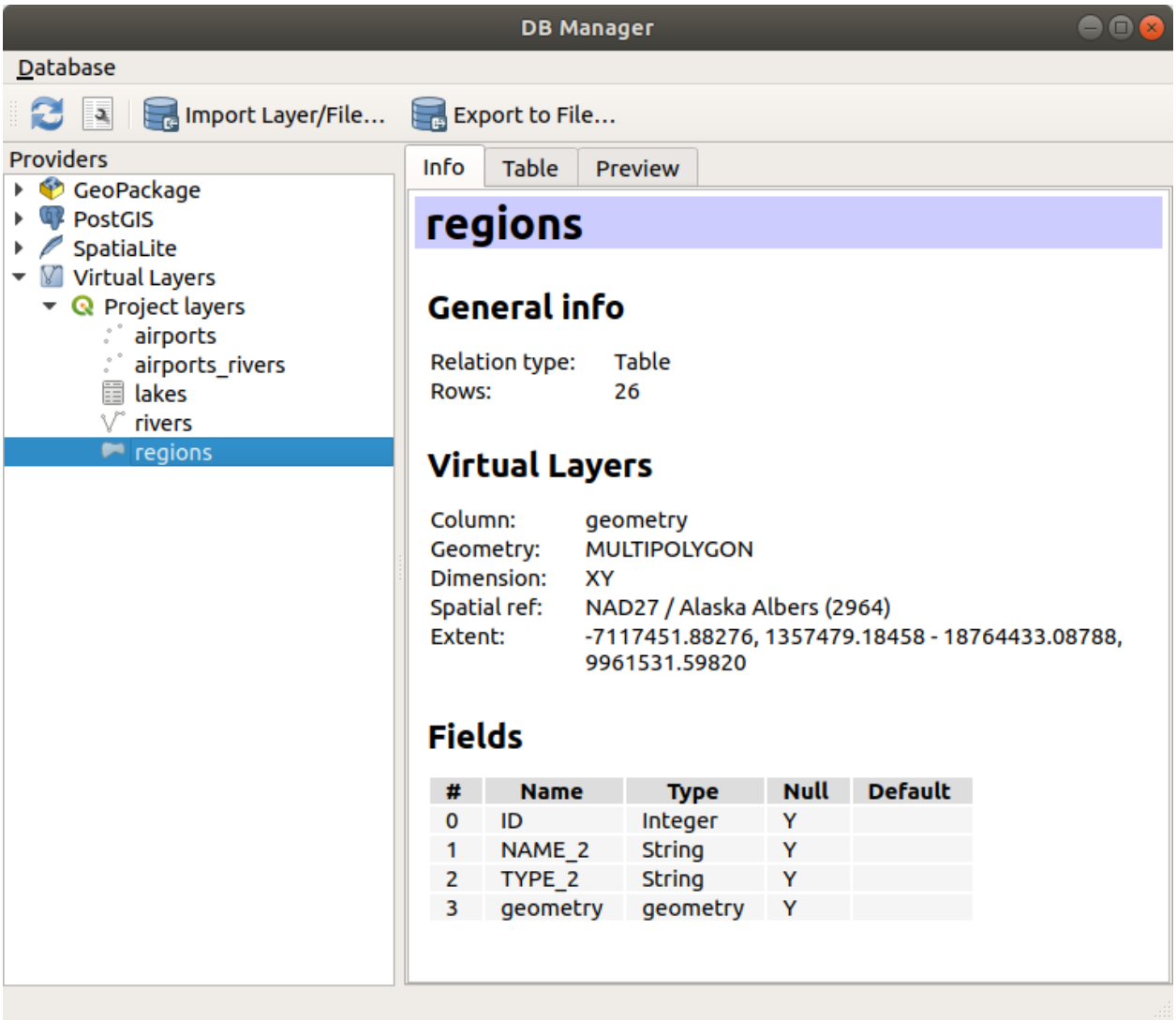


그림 25.8: 데이터베이스 관리자 대화창

Database 메뉴를 통해 기존 데이터베이스에 연결할 수 있고, SQL 창을 열 수 있으며, DB 관리자 플러그인을 종료할 수 있습니다. 기존 데이터베이스에 연결하고 나면, (PostGIS / PostgreSQL 같은 DBMS 들 관련) *Schema* 와 *Table* 메뉴가 나타납니다.

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 Logging…*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.

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.

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.

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 환경 설정

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

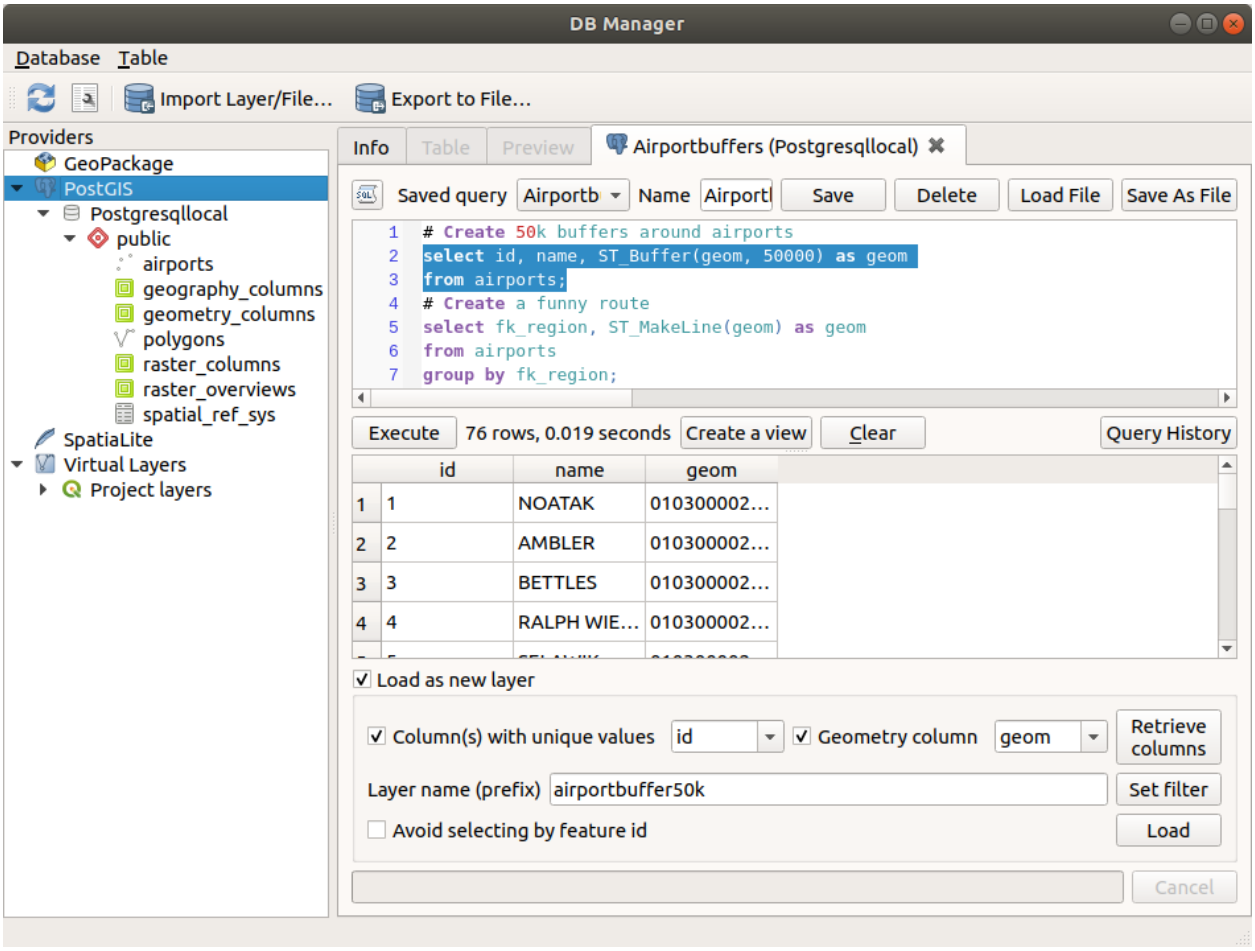


그림 25.9: 데이터베이스 관리자 SQL 창에서 SQL 쿼리 실행하기

  - ☑ 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.

어떤 행을 선택하면 오류의 위치를 볼 수 있습니다. *Error* (default), *Feature*, *Don't move*, 및 *Highlight contour of selected features* 액션들 가운데 하나를 선택하면 이 습성을 변경할 수 있습니다.

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.

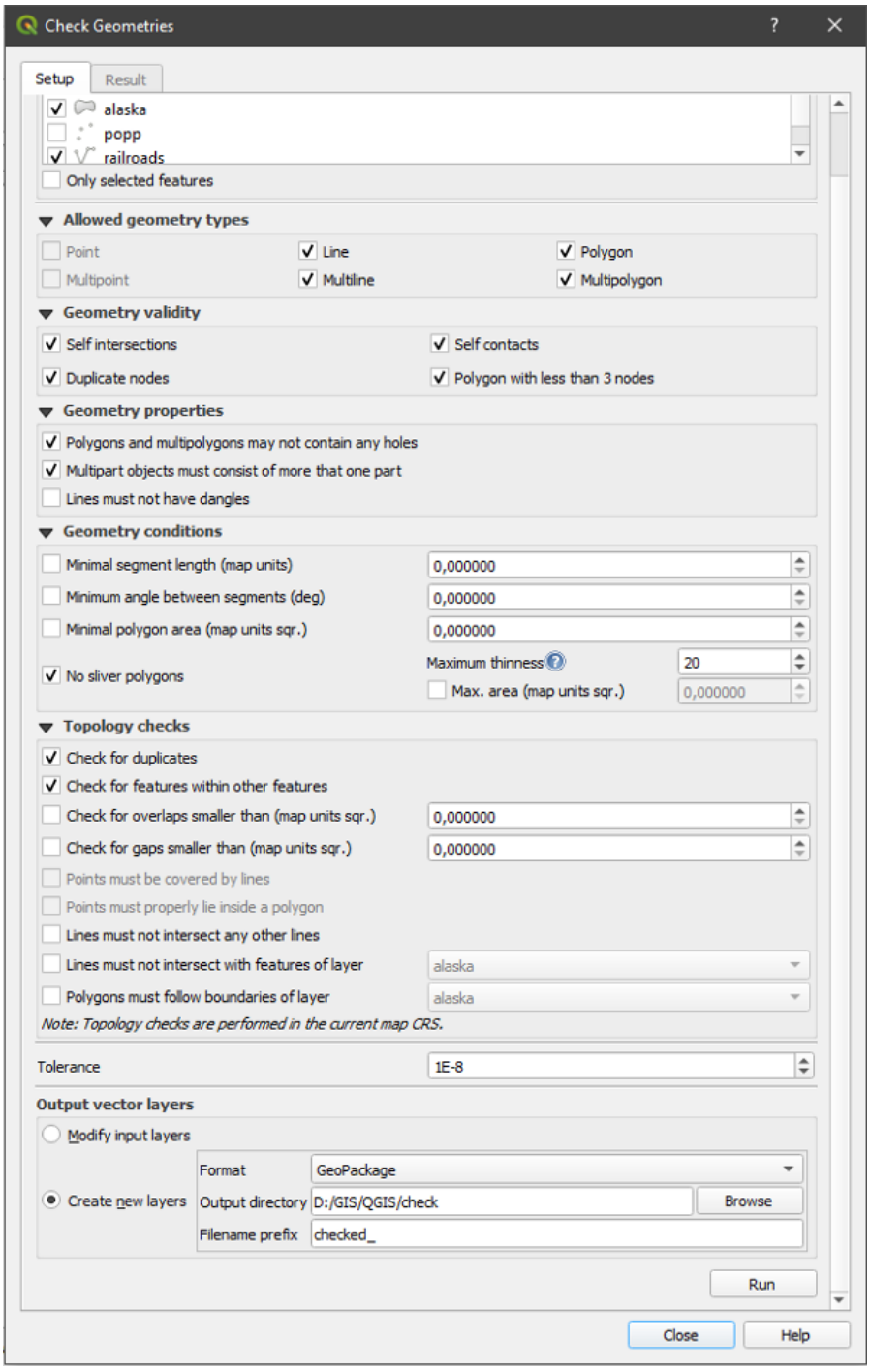


그림 25.10: 도형 점검기 플러그인

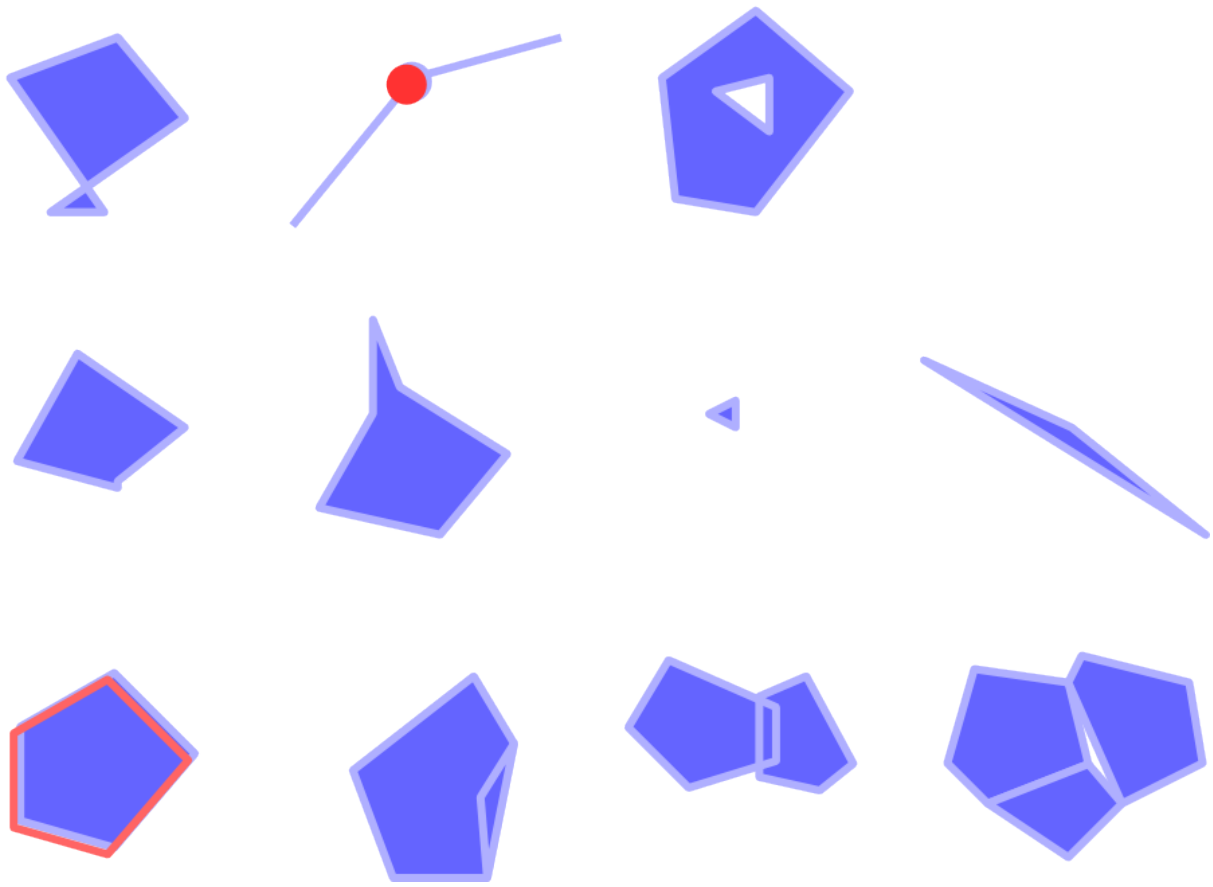


그림 25.11: 도형 점검기 플러그인이 지원하는 서로 다른 점검들

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

개요

메타검색 (MetaSearch) 은 메타데이터 카탈로그 서비스와 대화형으로 작업하는 QGIS 플러그인으로, OGC 웹용 카탈로그 서비스 (CSW) 표준을 지원합니다.

메타검색은 QGIS 내부에서 메타데이터 카탈로그를 검색할 수 있는 쉽고 직관적인 접근법과 사용자 친화적인 인터페이스를 제공하고 있습니다.

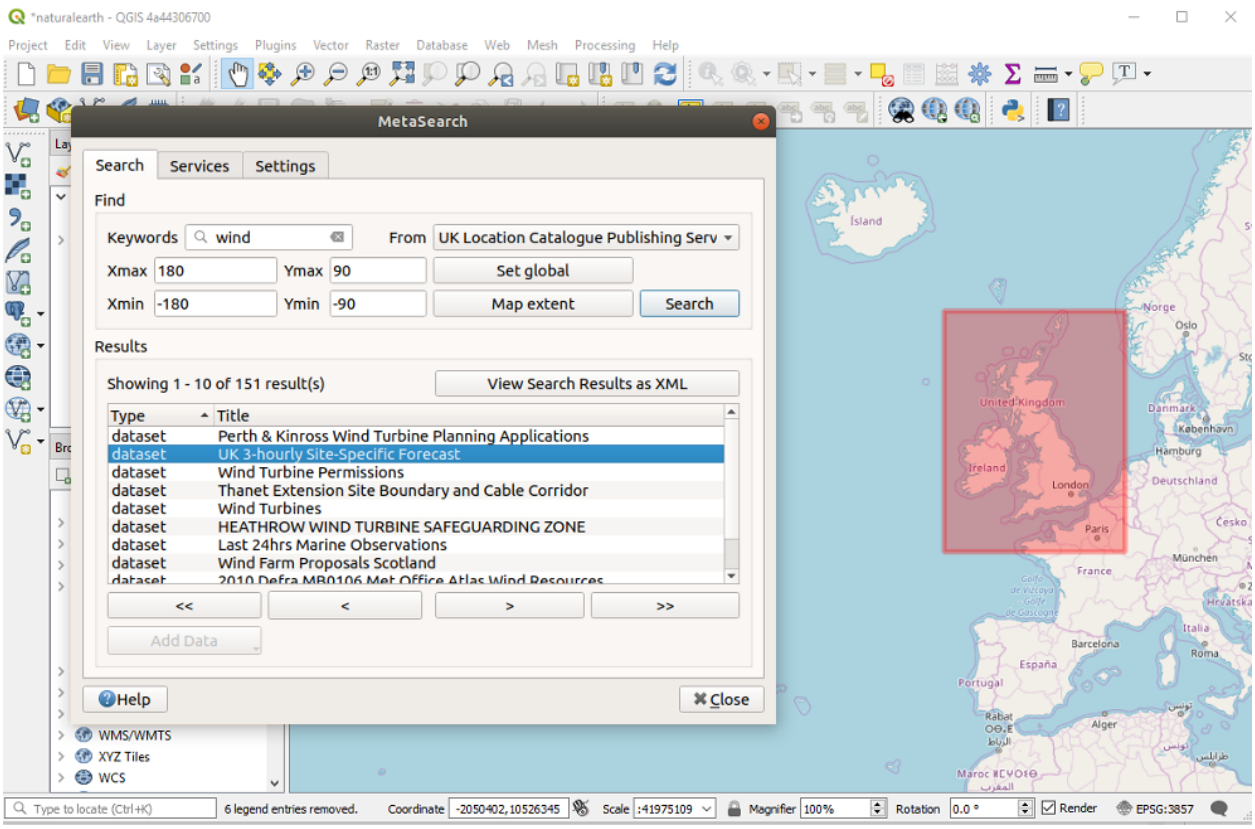


그림 25.12: 메타검색으로 서비스를 검색한 결과

QGIS 에서 메타데이터 카탈로그 작업하기

QGIS 는 기본적으로 메타검색을 모든 의존성과 함께 내장하고 있으며, QGIS 플러그인 관리자에서 활성화할 수 있습니다.

웹용 카탈로그 서비스 (CSW)

CSW(웹용 카탈로그 서비스) 는 OGC(개방형 공간 정보 컨소시엄) 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 클릭한 하십시오.
2. *Name* 란에 서비스명을 입력하고 *URL* 에 (종단점) 서비스 URL 을 입력하십시오. (전체 GetCapabilities URL 이 아니라) 기본 URL 만 입력하면 된다는 점을 기억하세요.
3. CSW 가 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. *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 name="Data.gov CSW" url="https://catalog.data.gov/csw-all"/>
    <csw name="Geonorge - National CSW service for Norway" url="https://www.geonorge.
↪no/geonetwork/srv/eng/csw"/>
    <csw name="Geoportale Nazionale - Servizio di ricerca Italiano" url="http://www.
↪pcn.minambiente.it/geoportal/csw"/>
    <csw name="LINZ Data Service" url="http://data.linz.govt.nz/feeds/csw"/>
    <csw name="Nationaal Georegister (Nederland)" url="http://www.
↪nationaalgeoregister.nl/geonetwork/srv/eng/csw"/>
    <csw name="RNDT - Repertorio Nazionale dei Dati Territoriali - Servizio di ricerca
↪" url="http://www.rndt.gov.it/RNDT/CSW"/>
    <csw name="UK Location Catalogue Publishing Service" url="http://csw.data.gov.uk/
↪geonetwork/srv/en/csw"/>
```

(다음 페이지에 계속)

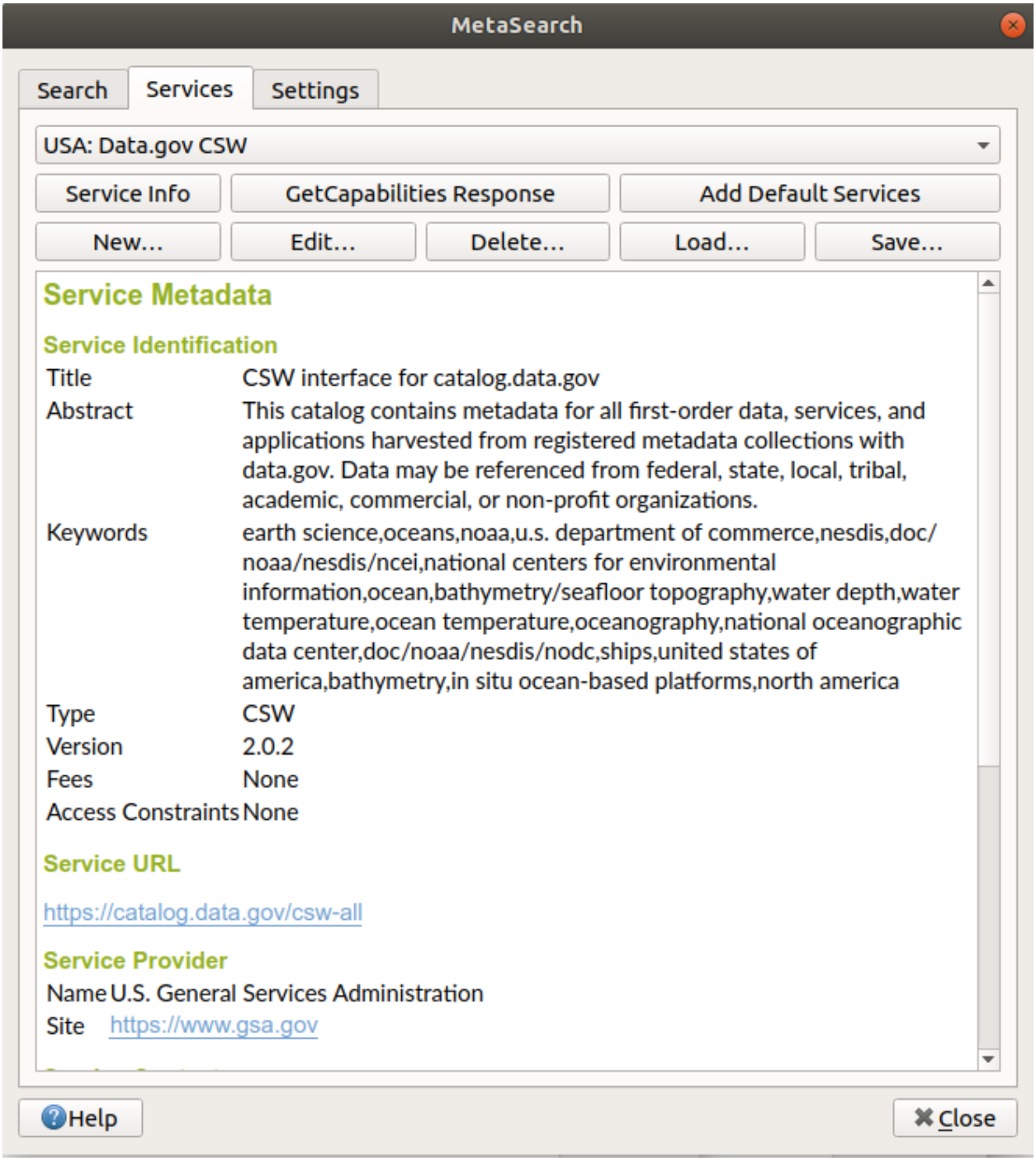


그림 25.13: 카탈로그 서비스 관리하기

(이전 페이지에서 계속)

```
    <csw 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) XML 응답을 살펴보고 싶다면, *GetCapabilities Response* 버튼을 누르십시오. Capabilities XML 을 표시한 개별 창이 열릴 것입니다.

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.
- **Bounding box**: *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 Search Results as XML* 버튼을 클릭하면 서비스 응답을 원시 XML 포맷으로 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.
- 레코드가 OGC 지원 웹 서비스 (WMS/WMTS, WFS, WCS, ArcGIS Map Service, ArcGIS Feature Service 등) 인 경우, *Add Data* 버튼이 활성화됩니다. 이 버튼을 클릭하면, 메타검색이 해당 레코드가 유효한 OGC 웹 서비스인지 검증할 것입니다. 그 다음 적절한 QGIS 연결 목록에 OGC 웹 서비스를 추가하고, 적합한 연결 대화창을 열 것입니다.

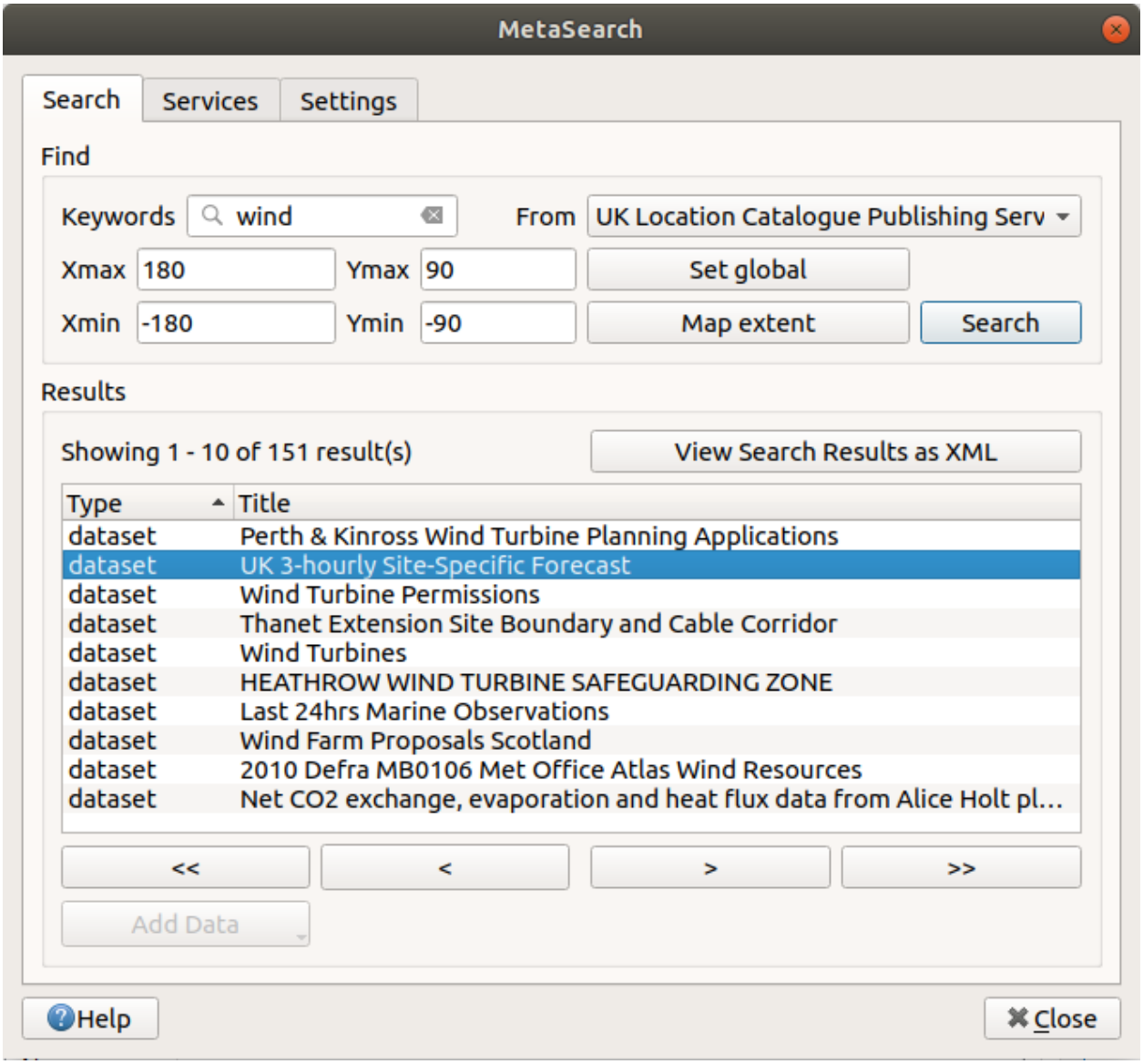


그림 25.14: 카탈로그 서비스 검색하기

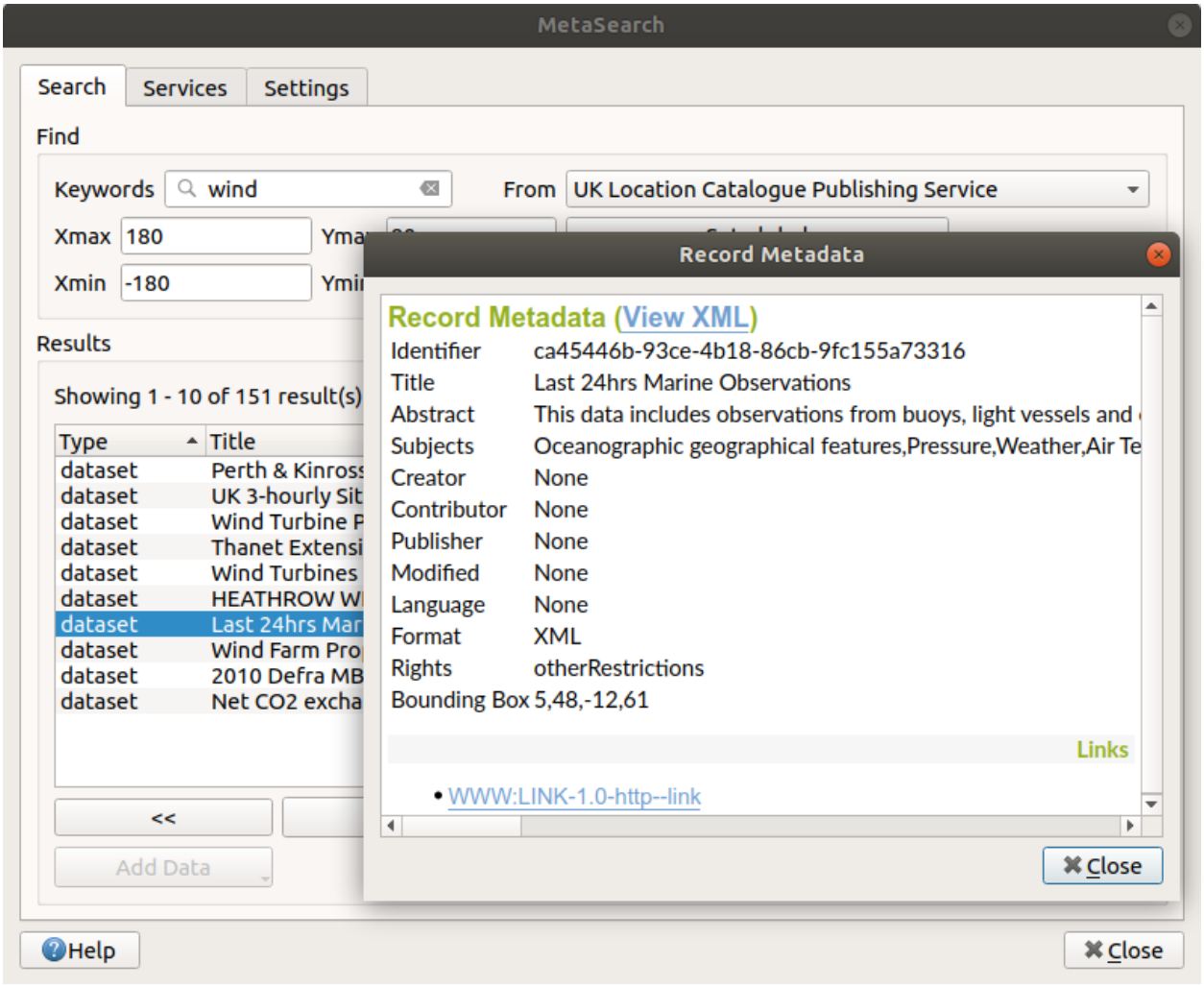


그림 25.15: 메타데이터 레코드 표시

설정

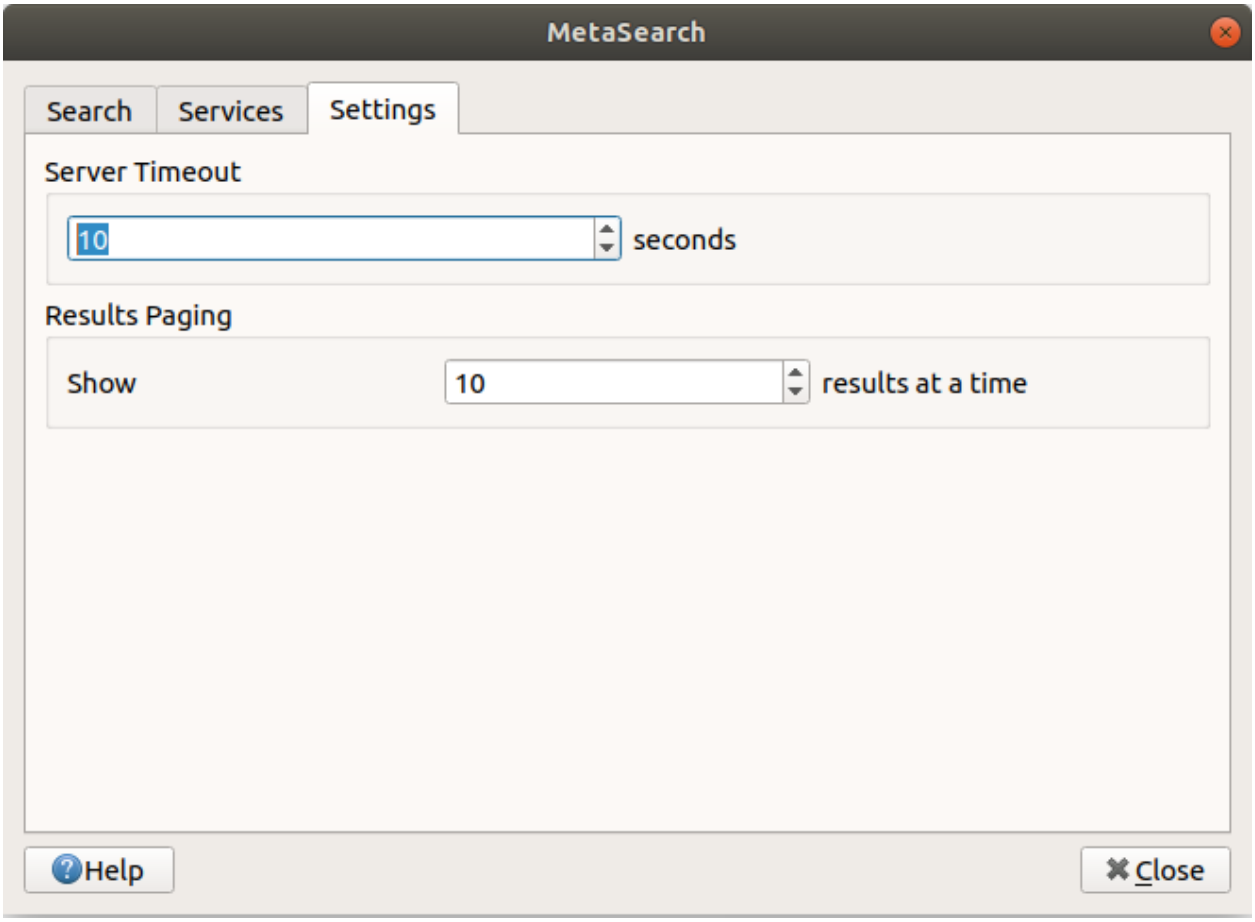


그림 25.16: 메타검색 설정하기

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.
- *Results paging*: 메타데이터 카탈로그를 검색할 때, 페이지 당 결과를 몇 개나 표시할지 설정합니다. 기본값은 10 입니다.

웹용 카탈로그 서비스 (CSW) 서버 오류

CSW 가 웹 브라우저에서 작동할 경우도 있지만, 메타검색은 아닙니다. 그 원인은 CSW 서버의 환경 설정 때문일 수도 있습니다. CSW 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 CSW 서버에도 일반적으로 적용할 수 있습니다.

25.2.4 오프라인 편집 작업 플러그인

데이터를 수집하는 경우, 현장에서 노트북 또는 휴대폰으로 오프라인 작업을 해야 하는 일이 흔합니다. 다시 온라인에 접속할 때, 오프라인에서 변경한 내용을 마스터 데이터소스 (예: PostGIS 데이터베이스) 와 동기화해야 합니다. 여러 사람이 동일한 데이터셋을 동시에 작업하고 있다면, 사람들이 동일한 객체를 변경하지 않았다고 해도 편집 내용을 수작업으로 합치기란 어려운 일입니다.

 Offline Editing 플러그인은 데이터소스 (보통 PostGIS 또는 WFS-T) 의 내용을 SpatiaLite 또는 GeoPackage 데이터베이스로 복사해서 전용 테이블에 오프라인 편집 내용을 저장함으로써 동기화 작업을 자동화합니다. 다시 온라인에 접속한 다음, 마스터 데이터셋에 오프라인 편집 내용을 적용할 수 있습니다.

플러그인을 사용하려면:

1. (예를 들어 PostGIS 또는 WFS-T 데이터소스에서 나온) 벡터 레이어를 가진 프로젝트를 여십시오.
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.
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.

25.2.5 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.

위상 점검기 플러그인을 사용하면 위상 규칙을 매우 쉽게 생성할 수 있습니다.

포인트 레이어 상에서 다음 규칙들을 사용할 수 있습니다:

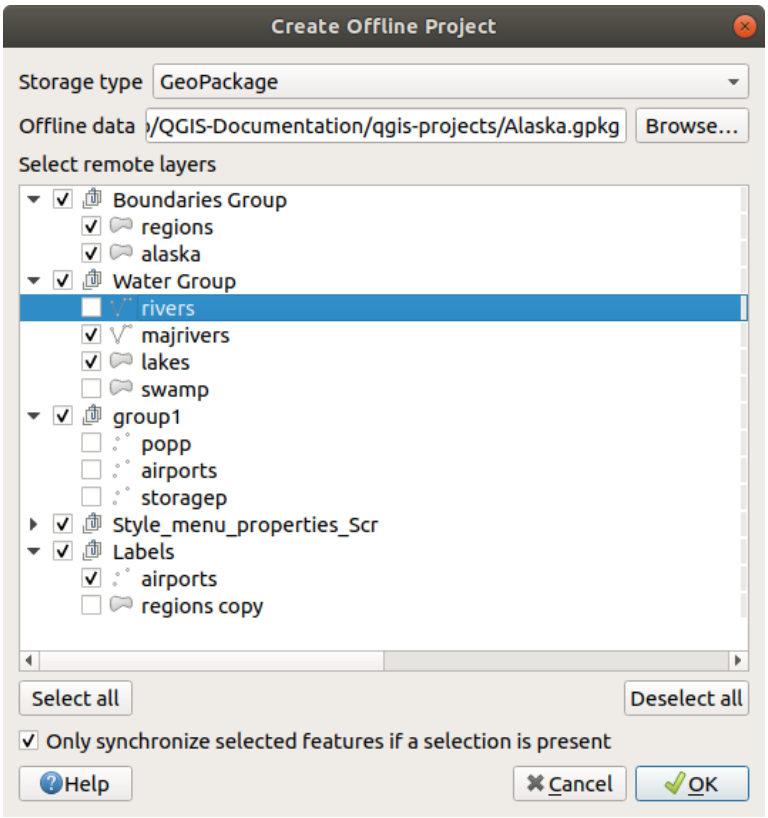


그림 25.17: 오프라인 프로젝트 생성하기

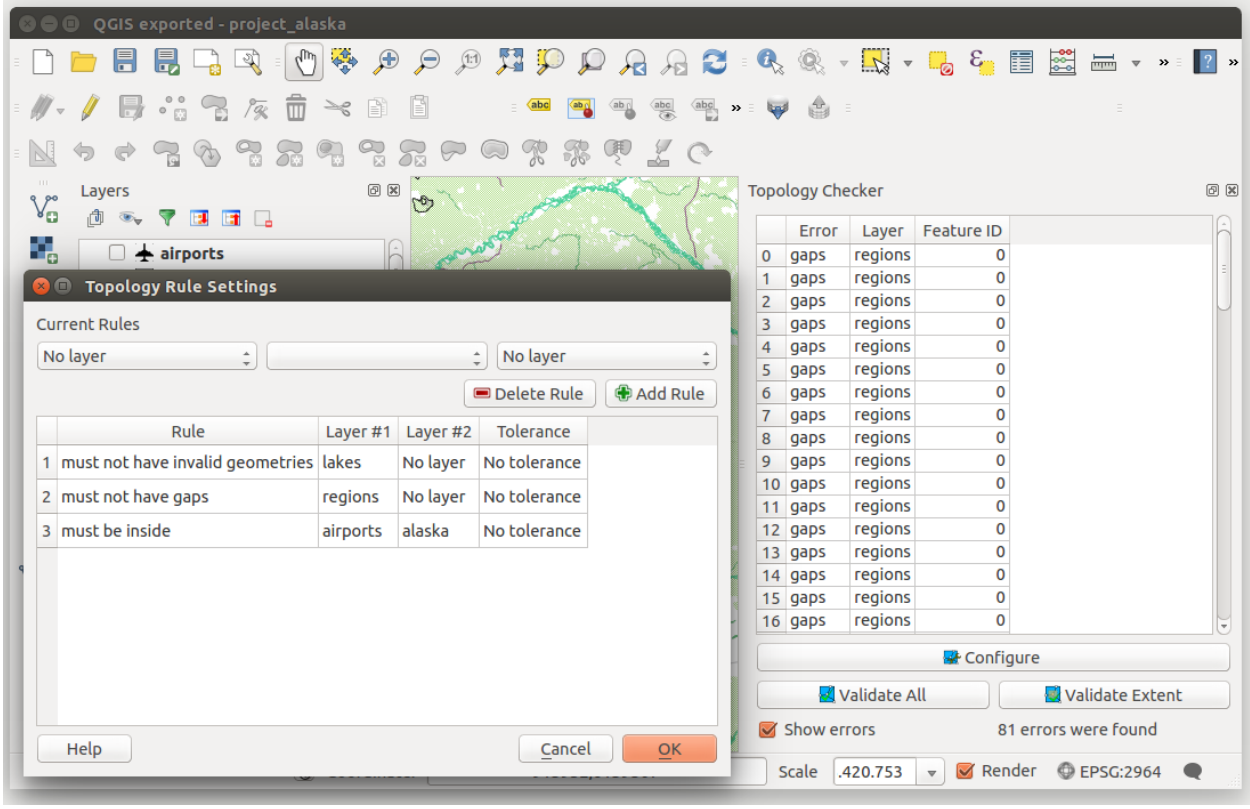


그림 25.18: 위상 점검기 플러그인

- **Must be covered by**: 이 규칙은 사용자 프로젝트에서 벡터 레이어를 선택할 수 있습니다. 지정한 벡터 레이어로 커버되지 않는 포인트를'오류'항목으로 분류합니다.
- **Must be covered by endpoints of**: 이 규칙은 사용자 프로젝트에서 라인 레이어를 선택할 수 있습니다.
- **Must be inside**: 이 규칙은 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**: 도형의 무결성을 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**: 이 규칙은 사용자 프로젝트에서 포인트 레이어를 선택할 수 있습니다.
- **Must not have dangles**: 라인 레이어에 있는 오버슛을 표시할 것입니다.
- **Must not have duplicates**: 어느 라인 객체가 두 번 이상 표현되는 경우, 해당 라인을'오류'로 분류합니다.
- **Must not have invalid geometries**: 도형의 무결성을 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**: 도형의 무결성을 점검합니다. 다음은 무결한 도형을 정의하는 규칙 가운데 일부입니다:
  - 폴리곤 고리는 닫혀 있어야만 합니다.
  - 구멍을 정의하는 고리는 외부 경계선을 정의하는 고리 내부에 있어야만 합니다.
  - 고리가 스스로 교차해서는 안 됩니다. (혼자 접촉하거나 공간교차해서는 안 됩니다.)
  - 고리는 한 포인트에서 접촉하는 경우 외엔 서로 접촉해서는 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 다중 폴리곤으로 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.

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
	GPS 도구	GPS 데이터를 불러오고 내보내는 도구입니다.	*GPS* 플러그인
	GRASS	GRASS 기능	*GRASS GIS* 통합
	메타검색 카탈로그 클라이언트 (MetaSearch Catalog Client)	메타데이터 카탈로그 서비스 (CSW) 와 대화형 작업을 합니다.	메타검색 카탈로그 클라이언트
	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

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 *<PyQGIS-Developer-Cookbook>* 을 참조) 를 제공합니다. 물론 QGIS 에는 파이썬 콘솔도 있습니다.

QGIS 파이썬 콘솔은 파이썬 명령어 실행을 위한 대화형 셸 (shell) 입니다. 그리고 사용자가 파이썬 스크립트를 편집하고 저장할 수 있는 파이썬 파일 편집기가 있습니다. 콘솔과 편집기 모두 PyQScintilla2 패키지를 기반으로 합니다. 콘솔을 열려면 *Plugins* → *Python Console* (`Ctrl+Alt+P`) 메뉴 옵션을 선택하십시오.

25.3.1 대화형 콘솔

대화형 콘솔은 툴바, 입력 영역 그리고 출력 영역으로 이루어져 있습니다.

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 속성을 환경 설정할 수 있는 대화창을 엽니다. (파이썬 콘솔 설정 참조)

- Help…: 현재 문서를 탐색합니다.

콘솔

콘솔의 주요 기능은 다음과 같습니다:

- 다음 API 들을 위한 코드 완성, 문법 구문 강조, 그리고 팝업 도움말:
  - 파이썬
  - PyQGIS
  - PyQt5
  - QScintilla2
  - osgeo-gdal-ogr
- `Ctrl+Alt+Space`: 파이썬 콘솔 설정 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.
- `_api` 를 입력하면 QGIS C++ API 문서를 엽니다.
- `_pyqgis` 를 입력하면 QGIS Python API 문서를 엽니다.
- `_cookbook` 을 입력하면 PyQGIS 개발자 쿡북 을 엽니다.

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.

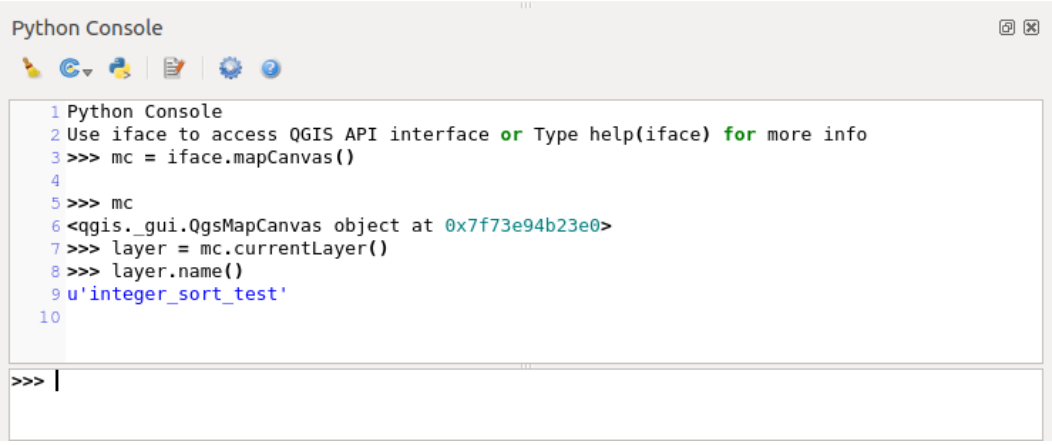


그림 25.19: 파이썬 콘솔

25.3.2 코드 편집기

편집기 위젯을 활성화하려면  Show Editor 버튼을 클릭하십시오. 편집기는 파이썬 파일을 편집하고 저장할 수 있으며, 사용자 코드를 관리하기 위한 고급 기능들 (코드 주석 추가 및 삭제, 문법 확인, http://codepad.org/ 를 통해 코드 공유 등등) 을 제공합니다. 주요 기능은 다음과 같습니다:

- 다음 API 들을 위한 코드 완성, 문법 구문 강조, 그리고 팝업 도움말:
  - 파이썬
  - PyQGIS
  - PyQt5
  - QScintilla2
  - osgeo-gdal-ogr
- `Ctrl+Space`: 자동 완성 목록 보기
- http://codepad.org/ 를 통해 코드 조각 공유하기
- `Ctrl+4`: 문법 확인하기
- 검색란 (데스크탑 환경 기본 단축키로 엽니다. 일반적으로 `Ctrl+F` 조합키를 누르면 됩니다.):
  - 데스크탑 환경 기본 단축키 (`Ctrl+G` 및 `Shift+Ctrl+G`) 를 사용해서 다음/이전 찾기
  - 검색창 입력 시 첫 번째 일치 항목을 자동으로 찾기
  - 검색창을 열 때 처음 찾은 문자열을 선택하도록 설정하기
  - `Esc` 키를 눌러 검색창 닫기
- 객체 조사기: 클래스 및 함수 탐색기
- 마우스 클릭으로 객체 정의로 가기 (객체 조사기에서)
- 컨텍스트 메뉴의  *Run Selected* 를 선택해서 코드 조각 실행하기
- 컨텍스트 메뉴의 *Run Script* 를 선택해서 전체 스크립트를 실행하기 (이렇게 하면 `.pyc` 확장자를 가진 바이트코드 파일을 컴파일합니다.)

참고: 코드 편집기에서 스크립트 일부 또는 전체를 실행하면 콘솔의 출력 영역에 결과를 산출합니다.

팁: 옵션 저장하기

콘솔 위젯 상태를 저장하려면, 닫기 버튼을 사용해서 파이썬 콘솔을 종료해야 합니다. 이렇게 하면 다음 시작 시 도형이 복구되도록 저장할 수 있습니다.

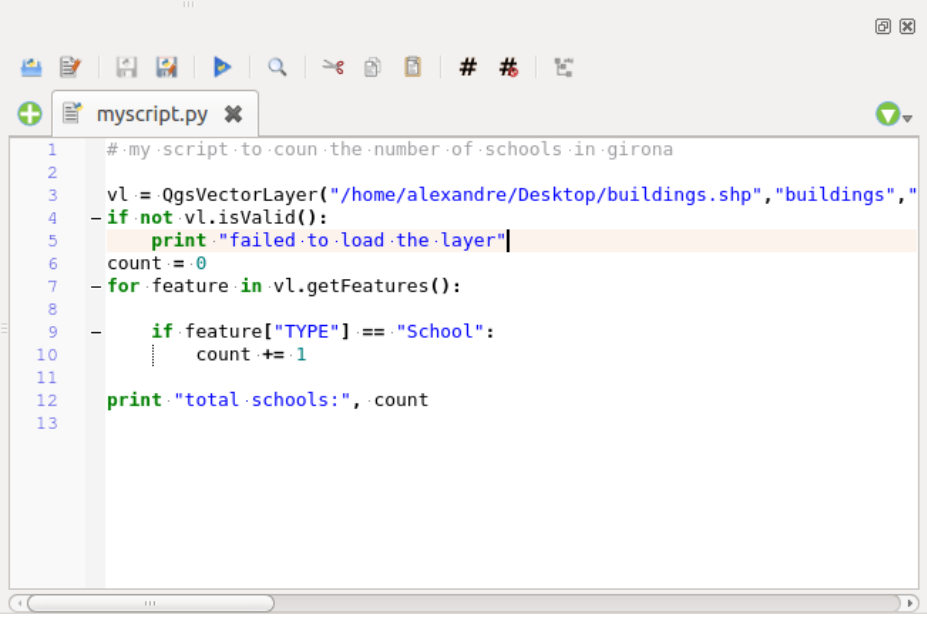


그림 25.20: 파이썬 콘솔 편집기

CHAPTER 26

도움 및 지원

26.1 메일링 리스트

QGIS 는 지금도 활발히 개발 중이기에 항상 사용자가 기대한대로 동작하지는 않습니다. 도움을 얻을 수 있는 좋은 방법은 QGIS-users 메일링 리스트에 가입하는 것입니다. 여러 사람이 당신의 질문과 그에 대한 답변을 보게 될 것이고, 많은 이들에게 도움이 될 것입니다.

26.1.1 QGIS 사용자

이 메일링 리스트는 일반적인 QGIS 논의뿐만 아니라, 설치 및 사용에 대한 구체적인 질문에도 사용됩니다. 사용자는 다음 URL 을 방문해서 QGIS-users 메일링 리스트를 구독하실 수 있습니다: https://lists.osgeo.org/mailman/listinfo/qgis-user

26.1.2 QGIS 개발자

만약 당신이 보다 기술적인 문제에 맞닥뜨린 개발자라면, QGIS-developer 메일링 리스트에 참여하는 편이 좋습니다. 이 리스트에서 QGIS 관련 사용자 경험 (UX) 및 사용성 문제도 제기하거나 찾아보거나 토의할 수 있습니다. https://lists.osgeo.org/mailman/listinfo/qgis-developer 에서 개발자 메일링 리스트를 구독하세요.

26.1.3 QGIS 커뮤니티 팀

이 메일링 리스트는 문서, 도움말, 사용자 지침서, 웹사이트, 블로그, 메일링 리스트, 포럼, 번역 활동 등에 대한 주제를 다룹니다. 만약 당신이 사용자 지침서 작업에 참여하고 싶으시다면, 이 메일링리스트가 당신이 질문을 시작할 적절한 지점입니다. https://lists.osgeo.org/mailman/listinfo/qgis-community-team 에서 메일링 리스트를 구독하세요.

26.1.4 QGIS 번역

이 메일링 리스트는 번역 활동에 대해 다룹니다. 만약 당신이 웹사이트, 사용자 지침서나 그래픽 사용자 인터페이스 (GUI) 의 번역에 참여하고 싶으시다면, 이 메일링 리스트가 당신이 질문을 시작할 적절한 지점입니다. https://lists.osgeo.org/mailman/listinfo/qgis-tr 에서 메일링 리스트를 구독하세요.

26.1.5 QGIS 프로젝트 운영 위원회

이 메일링 리스트에서는 QGIS 의 종합적 관리와 방향에 대한 커뮤니티 운영 이슈에 관한 토의가 이루어집니다. https://lists.osgeo.org/mailman/listinfo/qgis-psc 에서 메일링 리스트를 구독하세요.

26.1.6 QGIS 사용자 그룹

현지에서 QGIS 를 홍보하고 프로그램 개발에 공헌하기 위해, 일부 QGIS 커뮤니티는 QGIS 사용자 그룹으로 조직화하기도 합니다. 이 그룹들은 현지의 문제점들을 토의하고, 지역별 또는 국가별 사용자 모임을 주최하며, 특정 기능들의 후원자를 모집하고 있습니다. https://qgis.org/en/site/forusers/usergroups.html 에서 현재 사용자 그룹 목록을 살펴볼 수 있습니다.

어떤 메일링 리스트를 구독하시더라도 좋습니다. 질문에 답해주고 경험을 공유하는 것이 메일링 리스트에 기여하는 것임을 기억해 주십시오.

26.2 IRC

사용자는 IRC 로 참여할 수도 있습니다. irc.freenode.net 의 #qgis 채널에 가입해서 우리를 방문해 주세요. 채널의 많은 이들이 다른 일을 하고 있기에 당신의 질문을 발견할 때까지 시간이 걸릴 수 있으니, 질문에 대한 응답을 기다려 주세요. IRC 상에서 이루어지는 토의를 놓쳤다고 해도 걱정하지 마세요! 모든 토의를 로그로 저장하기 때문에, 쉽게 찾아볼 수 있습니다. http://irclogs.geoapt.com/qgis/ 를 방문해서 IRC-logs 를 읽어보면 됩니다.

26.3 상업적 지원

QGIS 를 상업적으로도 지원할 수 있습니다. 자세한 정보를 원한다면 https://qgis.org/en/site/forusers/commercial_support.html 웹사이트를 살펴보세요.

26.4 버그 추적기

QGIS-users 메일링 리스트가 일반적인'QGIS 에서 XYZ 를 하려면 어떻게 해야 하나요'유형의 질문에 유용할지라도, 사용자가 QGIS 의 버그에 대해 알려주고 싶을 수도 있습니다. 사용자는 QGIS 버그 추적기 를 이용해 버그 리포트를 제출할 수 있습니다.

당신이 제출한 버그가 항상 생각했던 (그 심각성에 따라 다른) 우선 순위를 가질 수는 없다는 점을 명심해주십시오. 어떤 버그는 해결을 위해 엄청난 개발 노력을 필요로 할 수도 있으며, 항상 이를 위해 모든 노력을 쏟을 수만은 없습니다.

기능 요청은 버그 추적을 위한 티켓 시스템을 이용해 보내실 수 있습니다. 꼭 `Feature request` 유형을 선택해 주세요.

만약 사용자가 버그를 발견하고 스스로 해결했다면, Github QGIS 프로젝트 에 풀 요청 (Pull Request) 을 제출할 수 있습니다.

더 자세한 정보를 원한다면 버그, 기능 및 이슈 와 패치 및 풀 요청 제출 을 읽어보세요.

26.5 블로그

QGIS 커뮤니티는 https://plugins.qgis.org/planet/ 에서 웹블로그도 운영하고 있는데, 사용자 및 개발자를 위한 흥미로운 기사들을 제공합니다. 수많은 다른 QGIS 블로그도 존재하며, 사용자 스스로 QGIS 블로그를 작성해서 기여할 수도 있답니다!

26.6 플러그인

https://plugins.qgis.org 웹사이트는 공식 QGIS 플러그인 웹 포털입니다. '공식 QGIS 플러그인 저장소'를 통해 사용자가 QGIS 플러그인 안정 버전 및 실험적 버전 목록을 찾아볼 수 있습니다.

26.7 위키

마지막으로, https://github.com/qgis/QGIS/wiki 에서 WIKI 웹사이트를 운영하고 있습니다. QGIS 개발, 배포 계획, 다운로드 사이트 링크, 메시지 번역에 대한 조언 등에 관련된 다양하고 유용한 정보를 찾아볼 수 있습니다. 확인해보세요. 훌륭한 내용들이 있답니다!

CHAPTER 27

공헌자

QGIS 는 헌신적인 자원봉사자와 조직들로 이루어진 팀이 개발하는 오픈소스 프로젝트입니다. 우리는 인종, 신념, 성별 및 삶의 방식에 상관없이 모든 이들을 환영하는 커뮤니티가 되려 합니다. 언제라도, 참여 하실 수 있습니다.

27.1 저자

다음은 QGIS 문서 전체를 작성하고, 검토하고, 업데이트하는 데 시간과 열정을 쏟아준 사람들의 명단입니다.

팀 서튼 (Tim Sutton)	이브 자콜랑 (Yves Jacolin)	야코브 란스토르프 (Jacob Lanstorp)	개리 E. 셔먼 (Gary E. Sherman)	리하르트 두이펜포오르데 (Richard Duivenvoorde)
타라 에이선 (Tara Athan)	아니타 그라저 (Anita Graser)	아르노 모르반 (Arnaud Morvan)	개빈 맥컬레이 (Gavin Macaulay)	루카 카사그란데 (Luca Casagrande)
K. 코이 (K. Koy)	위고 메르시 (Hugo Mercier)	아크바르 굼비라 (Akbar Gumbira)	마리 실베스트리 (Marie Silvestre)	위르겐 E. 피셔 (Jürgen E. Fischer)
프란 라가 (Fran Raga)	에릭 고다드 (Eric Goddard)	마르틴 도비아시 (Martin Dobiáš)	디타르트 얀젠 (Diethard Jansen)	사베르 라즈무에이 (Saber Razmjooei)
나가세 고 (Ko Nagase)	나이얼 도슨 (Nyall Dawson)	마티아스 쿤 (Matthias Kuhn)	안드레아스 노이만 (Andreas Neumann)	아리수 상트-안나 (Harrissou Sant-anna)
마넬 클로 (Manel Clos)	데이빗 윌리스 (David Willis)	라리사 유네크 (Larissa Junek)	폴 블로티에르 (Paul Blottière)	제바스티안 디트리히 (Sebastian Dietrich)
크리스 마요 (Chris Mayo)	슈테판 홀 (Stephan Holl)	망누스 호만 (Magnus Homann)	베른하르트 스트뢰블 (Bernhard Ströbl)	알레산드로 파소티 (Alessandro Pasotti)
네드 호닝 (Ned Horning)	라딤 블라제크 (Radim Blažek)	요슈아 아르노트 (Joshua Arnott)	루카 만가넬리 (Luca Manganelli)	마르코 후겐토블러 (Marco Hugentobler)
안드르 마누 (Andre Mano)	미에 빈스트루프 (Mie Winstrup)	프랭크 소콜릭 (Frank Sokolic)	뱅상 피카베트 (Vincent Picavet)	장-록 모레알 (Jean-Roc Morreale)
앤디 앨런 (Andy Allan)	빅터 올라야 (Victor Olaya)	타일러 미첼 (Tyler Mitchell)	레네-뤽 디혼트 (René-Luc D'Hont)	마르코 베르나조키 (Marco Bernasocchi)
일카 린네 (Ilkka Rinne)	베르너 마호 (Werner Macho)	크리스 버코우트 (Chris Berkhout)	니홀라스 두간 (Nicholas Duggan)	조너선 윌리츠 (Jonathan Willitts)
데이빗 애들러 (David Adler)	라르스 루트만 (Lars Luthman)	브렌단 모렐리 (Brendan Morely)	라이몬트 니이센 (Raymond Nijssen)	카슨 J. Q. 파머 (Carson J. Q. Farmer)
야카 크란츠 (Jaka Kranjc)	메진 워르쿠 (Mezene Worku)	패트릭 선터 (Patrick Sunter)	스티븐 코드웰 (Steven Cordwell)	스테판 블루멘트라트 (Stefan Blumentrath)
안디 슈미트 (Andy Schmid)	뱅상 모라 (Vincent Mora)	알레샨드르 네투 (Alexandre Neto)	히엔 쩐꽝 (Hiển Trần-Quang)	알렉산드레 부스케츠 (Alexandre Busquets)
주앙 가스파르 (João Gaspar)	탐 크랄리디스 (Tom Kralidis)	알레한데르 브루이 (Alexander Bruy)	파올로 카발리니 (Paolo Cavallini)	밀로 판 데어 린덴 (Milo Van der Linden)
피터 어스츠 (Peter Ersts)	우자발 간디 (Ujaval Gandhi)	도미닉 켈러 (Dominic Keller)	조반니 만기 (Giovanni Manghi)	막시밀리안 크람바흐 (Maximilian Krambach)
아네 크히슬라 (Anne Ghisla)	디크 그로스캄프 (Dick Groskamp)	우로시 프렐로즈니크 (Uroš Preloznik)	스테판 브뤼너 (Stéphane Brunner)	QGIS 한국어 번역자
시키 졸탄 (Zoltán Siki)	호바르드 트베이테 (Håvard Tveite)	마테오 게타 (Mattheo Ghetta)	살바토레 라로사 (Salvatore Larosa)	콘스탄티노스 니콜라우 (Konstantinos Nikolaou)
탐 채드윈 (Tom Chadwin)	래리 셰이퍼 (Larry Shaffer)	네이선 우드로 (Nathan Woodrow)	마르티나 사바레세 (Martina Savarese)	고도프레도 콘트레라스 (Godofredo Contreras)
아스트리드 엠데 (Astrid Emde)	루이지 피렐리 (Luigi Pirelli)	토마 그라티에 (Thomas Gratier)	조반니 알레그리 (Giovanni Allegri)	GiordanoPezzola
파올로 코르티 (Paolo Corti)	투도르 버러스쿠 (Tudor Bărăscu)	마닝 삼발레 (Maning Sambale)	클라우디아 A. 엥겔 (Claudia A. Engel)	카야마 요이치 (Yoichi Kayama)
오토 다사우 (Otto Dassau)	데니스 로우차우트 (Denis Rouzaud)	닉 베어먼 (Nick Bearman)	embelding	ajazepk
Ramon	Andrei	zstadler	icephale	로사 아길라르 (Rosa Aguilar)

27.2 번역자

QGIS 는 여러 언어를 지원하는 응용 프로그램으로, 몇 가지 언어로 번역된 문서가 출판되어 있습니다. 그 외 다른 많은 언어로도 번역 중이며 번역 분량이 일정 수준에 다다르면 배포될 예정입니다. 어떤 언어의 번역을 향상시키거나 새로운 언어로 번역을 요청하려면, https://qgis.org/ko/site/getinvolved/index.html 웹페이지를 참조해주십시오.

현재 진행된 번역은 다음 분들 덕분입니다:

언어	공헌자
인도네시아어	에미르 하르타토 (Emir Hartato), 이 마데 아놈바와 (I Made Anombawa), 자누아르 V. 시마르마타 (Januar V. Simarmata), 무함마드 이크나울 하크 시레가르 (Muhammad Iqnaul Haq Siregar), 트리아스 아디티야 (Trias Aditya)
중국어 (번체)	캘빈 응예 (Calvin Ngei), 장 전 (Zhang Jun), 리처드 셰 (Richard Xie)
네덜란드어	카를로 판 리이스비이크 (Carlo van Rijswijk), 디크 그로스캄프 (Dick Groskamp), 디타르트 얀젠 (Diethard Jansen), 라이몬트 니이센 (Raymond Nijssen), 리하르트 두이펜포오르데 (Richard Duivenvoorde), 빌렘 호프만 (Willem Hoffman)
핀란드어	마티 맨튀넨 (Matti Mäntynen), 카리 미코넨 (Kari Mikkonen)
프랑스어	아르노 모르반 (Arnaud Morvan), 오귀스탱 로셰 (Augustin Roche), 디디에 반덴 베르게 (Didier Vanden Berghe), Dofabien, 에티엔 트리마유 (Etienne Trimaille), 프랑시스 가스크 (Francis Gasc), 하리수 생탄나 (Harrissou Sant-anna), 장-로크 모레알 (Jean-Roc Morreale), 제레미 가르니오 (Jérémy Garniaux), 로익 부스코 (Loïc Buscoz), Lsam, 마르크-앙드레 사이아 (Marc-André Saia), 마리 실베스트르 (Marie Silvestre), 마티외 보사르트 (Mathieu Bossaert), 마티외 라트 (Mathieu Lattes), 마예울 카우프만 (Mayeul Kauffmann), 메데릭 리브로 (Médéric Ribreux), 메흐디 셈샤위 (Mehdi Semchaoui), 미카엘 두샹 (Michael Douchin), 니콜라 보아스토 (Nicolas Boisteault), 니콜라 로샤르 (Nicolas Rochard), 파스칼 오브스테타르 (Pascal Obstetar), 로뱅 프레스트 (Robin Prest), 로드 베라 (Rod Bera), 스테판 앙리오 (Stéphane Henriod), 스테판 포사마이 (Stéphane Possamai), sylther, 실뱅 베이디 (Sylvain Badey), 실뱅 메이야 (Sylvain Maillard), 뱅상 피카베 (Vincent Picavet), 자비에 타르듀 (Xavier Tardieu), 얀 레베유-메네 (Yann Leveille-Menez), yoda89
갈리시아어	샨 비에이로 (Xan Vieiro)
독일어	위르겐 E. 피셔 (Jürgen E. Fischer), 오토 다사우 (Otto Dassau), 슈테판 홀 (Stephan Holl), 베르너 마호 (Werner Macho)
힌두어	하리시 쿠마르 솔란키 (Harish Kumar Solanki)
이탈리아어	알레산드로 판나 (Alessandro Fanna), 안네 기슬라 (Anne Ghisla), 플라비오 리골론 (Flavio Rigolon), 줄리아노 쿠르티 (Giuliano Curti), 루카 카사그란데 (Luca Casagrande), 루카 델루치 (Luca Delucchi), 마르코 브라이다 (Marco Braida), 마테오 게타 (Matteo Ghetta), 모리치오 나폴리타노 (Maurizio Napolitano), 미켈레 베네벤티 (Michele Beneventi), 미켈레 페레티 (Michele Ferretti), 로베르토 안젤레티 (Roberto Angeletti), 파올로 카발리니 (Paolo Cavallini), 스테파노 캄푸스 (Stefano Campus)
일본어	바바 요시히코 (Baba Yoshihiko), 아카기 미노루 (Minoru Akagi), 야마테 노리히코 (Norihiro Yamate), 미즈타니 다카유키 (Takayuki Mizutani), 누이무라 다카유키 (Takayuki Nuimura), 가야마 요이치 (Yoichi Kayama)
한국어	OSGeo 한국어 지부
폴란드어	안제이 시비옹데르 (Andrzej Świąder), 보리스 유르기엘 (Borys Jurgiel), 에벨리나 크라프차크 (Ewelina Krawczak), 야쿠프 보브로프스키 (Jakub Bobrowski), 마테우시 워스코트 (Mateusz Łoskot), 미하우 쿠와흐 (Michał Kułach), 미하우 스모치크 (Michał Smoczyk), 밀레나 노보타르스카 (Milena Nowotarska), 라도스와프 파시오크 (Radosław Pasiok), 로베르트 슈체파네크 (Robert Szczepanek), 토마시 파울 (Tomasz Paul)
포르투갈어	알레샨드르 네투 (Alexandre Neto), 두아르트 카헤이라 (Duarte Carreira), 지오바니 망기 (Giovanni Manghi), 주앙 가스파르 (João Gaspar), 조아나 시몽이스 (Joana Simões), 레안드루 인판티니 (Leandro Infantini), 넬송 실바 (Nelson Silva), 페드루 팔레이루 (Pedro Palheiro), 페드루 페레이라 (Pedro Pereira), 히카르두 세나 (Ricardo Sena)
포르투갈어 (브라질)	아르투르 나니 (Arthur Nanni), 펠리피 소드레 바후스 (Felipe Sodré Barros), 레오니다스 데스코비 필류 (Leônidas Descovi Filho), 마르셀루 소아리스 소자 (Marcelo Soares Souza), 나르셀리우 지 사 페레이라 필류 (Narcélio de Sá Pereira Filho), 시드네이 스샤베를리 고베이아 (Sidney Schaberle Goveia)
루마니아어	알렉스 버데스쿠 (Alex Bădescu), 보그단 파쿠라르 (Bogdan Pacurar), 제오르지아나 이오아노비치 (Georgiana Ioanovici), 로누트 로시페스쿠-에네스쿠 (Lonut Losifescu-Enescu), 소린 컬리니커 (Sorin Călinică), 투도르 버러스쿠 (Tudor Bărăscu)
러시아어	알레한데르 브루이 (Alexander Bruy), 아르템 포포우 (Artem Popov)
스페인어	카를로스 다빌라 (Carlos Dávila), 디아나 갈린도 (Diana Galindo), 에뒨 아마도 (Edwin Amado), 가브리엘라 아와드 (Gabriela Awad), 하비에르 세사르 알다리스 (Javier César Aldariz), 마예울 카우프만 (Mayeul Kauffmann), 프란 라가 (Fran Raga)
우크라이나어	알레한데르 브루이 (Alexander Bruy)

27.3 번역 통계

QGIS 3.16 장기지원 배포판의 번역을 위한 활동 현황입니다.

(last update: 2022-03-11)

문자열의 개수	번역 언어의 개수	전반적인 번역률
32361	**59**	**12.3%**

언어	번역률 (%)	언어	번역률 (%)	언어	번역률 (%)
알바니아어	0.23	아랍어	4.02	아제르바이잔어	0.02
바스크어	1.42	벵골어	0.19	불가리아어	2.59
버마어	0.1	카탈루냐어	1.51	중국어 (간체)	8.53
중국어 (번체)	0.69	크로아티아어	0.12	체코어	6.0
덴마크어	0.66	네덜란드어	100.0	에스토니아어	1.3
핀란드어	1.81	프랑스어	98.49	갈리시아어	0.59
조지아어	0.11	독일어	21.73	그리스어	0.37
히브리어	0.74	힌두어	0.31	헝가리어	9.3
이그보어	0.01	인도네시아어	2.77	이탈리아어	88.87
일본어	71.07	커바일어	0.11	한국어	88.58
라오어	0.0	리투아니아어	6.06	마케도니아어	0.13
말레이어	0.05	말라얄람어	0.1	마라티어	0.19
몽골어	0.11	응코어	1.82	노르웨이 보크몰어	3.32
펀자브어	0.0	페르시아어	0.48	폴란드어	1.85
포르투갈어 (브라질)	37.01	포르투갈어 (포르투갈)	8.5	루마니아어	30.61
러시아어	14.94	세르비아어	0.11	슬로바키아어	1.55
슬로베니아어	3.2	스페인어	96.0	스웨덴어	1.19
타갈로그어	0.1	타밀어	0.52	텔루구어	0.03
태국어	0.11	터키어	2.82	우크라이나어	2.37
우르두어	0.0	베트남어	0.33		

CHAPTER 28

부록

28.1 부록 A: GNU 일반 공중 사용 허가서

2 판, 1991 년 6 월

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

누구든지 본 사용 허가서를 있는 그대로 복제하고 배포할 수 있습니다. 그러나 본문에 대한 수정은 허용되지 않습니다. 추가로, 본 한국어 번역물은 참조용으로 법적 효력이 없으며, 영문 사용 허가서의 이해를 돕기 위해 번역된 문서임을 밝힙니다.

전문 (□□)

소프트웨어에 적용되는 대부분의 사용 허가서 (license) 들은 소프트웨어에 대한 수정과 공유의 자유를 제한하려는 것을 그 목적으로 합니다. 그러나 GNU 일반 공중 사용 허가서 (이하, "GPL"이라고 칭합니다.) 는 자유 소프트웨어에 대한 수정과 공유의 자유를 모든 사용자들에게 보장하기 위해서 성립된 것입니다. 자유 소프트웨어 재단이 제공하는 대부분의 소프트웨어들은 GPL 에 의해서 관리되고 있으며, 몇몇 소프트웨어에는 별도의 사용 허가서인 GNU 라이브러리 일반 공중 사용 허가서 (GNU Library General Public License) 를 대신 적용하기도 합니다. 자유 소프트웨어란, 이를 사용하려고 하는 모든 사람에 대해서 동일한 자유와 권리가 함께 양도되는 소프트웨어를 말하며 프로그램 저작자의 의지에 따라 어떠한 종류의 프로그램에도 GPL 을 적용할 수 있습니다. 따라서 여러분이 만든 프로그램에도 GPL 을 적용할 수 있습니다.

자유 소프트웨어를 언급할 때 사용되는"자유"라는 단어는 무료 (□□) 를 의미하는 금전적인 측면의 자유가 아니라 구속되지 않는다는 관점에서의 자유를 의미하며, GPL 은 자유 소프트웨어를 이용한 복제와 개작, 배포와 수익 사업 등의 가능한 모든 형태의 자유를 실질적으로 보장하고 있습니다. 여기에는 원시 코드 (source code) 의 전부 또는 일부를 원용해서 개선된 프로그램을 만들거나 새로운 프로그램을 창작할 수 있는 자유가 포함되며, 자신에게 양도된 이러한 자유와 권리를 보다 명확하게 인식할 수 있도록 하기 위한 규정도 포함되어 있습니다.

GPL 은 GPL 안에 소프트웨어를 양도받을 사용자의 권리를 제한하는 조항과 단서를 별항으로 추가시키지 못하게 함으로써 사용자들의 자유와 권리를 실제적으로 보장하고 있습니다. 자유 소프트웨어의 개작과 배포에 관계하고 있는 사람들은 이러한 무조건적인 권리 양도 규정을 준수해야만 합니다.

예를 들어 GPL 프로그램을 배포할 경우에는 프로그램의 유료 판매나 무료 배포에 관계없이 자신이 해당 프로그램에 대해서 가질 수 있었던 모든 권리를, 프로그램을 받게될 사람에게 그대로 양도해 주어야 합니다. 이 경우, 프로그램의 원시 코드를 함께 제공하거나 원시 코드를 구할 수 있는 방법을 확실히 알려주어야 하고 이러한 모든 사항들을 사용자들이 분명히 알 수 있도록 명시해야 합니다.

자유 소프트웨어 재단은 다음과 같은 두 가지 단계를 통해서 사용자들을 권리를 보호합니다. (1) 소프트웨어에 저작권을 설정합니다. (2) 저작권의 양도에 관한 실정법에 의해서 유효한 법률적 효력을 갖는 GPL 을 통해 소프트웨어를 복제하거나 개작 및 배포할 수 있는 권리를 사용자들에게 부여합니다.

자유 소프트웨어를 사용하는 사람들은 반복적인 재배포 과정을 통해 소프트웨어 자체에 수정과 변형이 일어날 수도 있으며, 이는 최초의 저작자가 만든 소프트웨어가 갖고 있는 문제가 아닐 수 있다는 개연성을 인식하고 있어야 합니다. 우리는 개작과 재배포 과정에서 다른 사람에 의해 발생된 문제로 인해 프로그램 원저작자들의 신망이 훼손되는 것을 원하지 않습니다. GPL 에 자유 소프트웨어에 대한 어떠한 형태의 보증도 규정하지 않는 이유는 이러한 점들이 고려되었기 때문이며, 이는 프로그램 원저작자와 자유 소프트웨어 재단의 자유로운 활동을 보장하는 현실적인 수단이기도 합니다.

특허 제도는 자유 소프트웨어의 발전을 위협하는 요소일 수밖에 없습니다. 자유 프로그램을 재배포하는 사람들이 개별적으로 특허를 취득하게 되면, 결과적으로 그 프로그램이 독점 소프트웨어가 될 가능성이 있습니다. 자유 소프트웨어 재단은 이러한 문제에 대처하기 위해서 어떠한 특허에 대해서도 그 사용 권리를 모든 사람들 (이하, "공중 (□□)"이라고 칭합니다.) 에게 자유롭게 허용하는 경우에 한해서만 자유 소프트웨어와 함께 사용할 수 있다는 것을 명확히 밝히고 있습니다.

복제 (copying) 와 개작 (modification) 및 배포 (distribution) 에 관련된 구체적인 조건과 규정은 다음과 같습니다.
복제와 개작 및 배포에 관한 조건과 규정

0. 본 허가서는 GNU 일반 공중 사용 허가서의 규정에 따라 배포될 수 있다는 사항이 저작권자에 의해서 명시된 모든 컴퓨터 프로그램 저작물에 대해서 동일하게 적용됩니다. 컴퓨터 프로그램 저작물 (이하, "프로그램"이라고 칭합니다.) 이란 특정한 결과를 얻기 위해서 컴퓨터 등의 정보 처리 능력을 가진 장치 (이하, "컴퓨터"라고 칭합니다.) 내에서 직접 또는 간접으로 사용되는 일련의 지시 및 명령으로 표현된 창작물을 의미하고, "2 차적 프로그램"이란 전술한 프로그램 자신 또는 저작권법의 규정에 따라 프로그램의 전부 또는 상당 부분을 원용하거나 다른 언어로의 번역을 포함할 수 있는 개작 과정을 통해서 창작된 새로운 프로그램과 이와 관련된 저작물을 의미합니다. (이후로 다른 언어로의 번역은 별다른 제한없이 개작의 범위에 포함되는 것으로 간주합니다.) "피양도자"란 GPL 의 규정에 따라 프로그램을 양도받은 사람을 의미하고, "원 (□) 프로그램"이란 프로그램을 개작하거나 2 차적 프로그램을 만들기 위해서 사용된 최초의 프로그램을 의미합니다.

   본 허가서는 프로그램에 대한 복제와 개작 그리고 배포 행위에 대해서만 적용됩니다. 따라서 프로그램을 실행시키는 행위에 대한 제한은 없습니다. 프로그램의 결과물 (output) 에는, 그것이 프로그램을 실행시켜서 생성된 것인지 아닌지의 여부에 상관없이 결과물의 내용이 원프로그램으로부터 파생된 2 차적 프로그램을 구성했을 때에 한해서 본 허가서의 규정들이 적용됩니다. 2 차적 프로그램의 구성 여부는 2 차적 프로그램 안에서의 원프로그램의 역할을 토대로 판단합니다.

1. 적절한 저작권 표시와 프로그램에 대한 보증이 제공되지 않는다는 사실을 각각의 복제물에 명시하는 한, 피양도자는 프로그램의 원시 코드를 자신이 양도받은 상태 그대로 어떠한 매체를 통해서도 복제하고 배포할 수 있습니다. 복제와 배포가 이루어 질 때는 본 허가서와 프로그램에 대한 보증이 제공되지 않는다는 사실에 대해서 언급되었던 모든 내용을 그대로 유지시켜야 하며, 영문판 GPL 을 함께 제공해야 합니다.

   배포자는 복제물을 물리적으로 인도하는데 소요된 비용을 청구할 수 있으며, 선택 사항으로 독자적인 유료 보증을 설정할 수 있습니다.

2. 피양도자는 자신이 양도받은 프로그램의 전부나 일부를 개작할 수 있으며, 이를 통해서 2 차적 프로그램을 창작할 수 있습니다. 개작된 프로그램이나 창작된 2 차적 프로그램은 다음의 사항들을 모두 만족시키는 조건에 한해서, 제 1 조의 규정에 따라 또다시 복제되고 배포될 수 있습니다.

   a) 파일을 개작할 때는 파일을 개작한 사실과 그 날짜를 파일 안에 명시해야 합니다.

   b) 배포하거나 공표하려는 저작물의 전부 또는 일부가 양도받은 프로그램으로부터 파생된 것이라면, 저작물 전체에 대한 사용 권리를 본 허가서의 규정에 따라 공중에게 무상으로 허용해야 합니다.

   c) 개작된 프로그램의 일반적인 실행 형태가 대화형 구조로 명령어를 읽어 들이는 방식을 취하고 있을 경우에는, 적절한 저작권 표시와 프로그램에 대한 보증이 제공되지 않는다는 사실, (별도의 보증을 설정한 경우라면 해당 내용) 그리고 양도받은 프로그램을 본 규정에 따라 재배포할 수 있다는 사실과 GPL 사본을 참고할 수 있는 방법이 함께 포함된 문구가 프로그램이 대화형 구조로 평이하게 실행된 직후에 화면 또는 지면으로 출력되도록 작성되어야 합니다. (예외 규정: 양도받은 프로그램이 대화형

구조를 갖추고 있다 하더라도 통상적인 실행 환경에서 전술한 사항들이 출력되지 않는 형태였을 경우에는 이를 개작한 프로그램 또한 관련 사항들을 출력시키지 않아도 무방합니다.)

위의 조항들은 개작된 프로그램 전체에 적용됩니다. 만약, 개작된 프로그램에 포함된 특정 부분이 원프로그램으로부터 파생된 것이 아닌 별도의 독립 저작물로 인정될 만한 상당한 이유가 있을 경우에는 해당 저작물의 개별적인 배포에는 본 허가서의 규정들이 적용되지 않습니다. 그러나 이러한 저작물이 2 차적 프로그램의 일부로서 함께 배포된다면 개별적인 저작권과 배포 기준에 상관없이 저작물 모두에 본 허가서가 적용되어야 하며, 전체 저작물에 대한 사용 권리는 공중에게 무상으로 양도됩니다.

이러한 규정은 개별적인 저작물에 대한 저작자의 권리를 침해하거나 인정하지 않으려는 것이 아니라, 원프로그램으로부터 파생된 2 차적 프로그램이나 수집 저작물의 배포를 일관적으로 규제할 수 있는 권리를 행사하기 위한 것입니다.

원프로그램이나 원프로그램으로부터 파생된 2 차적 프로그램을 이들로부터 파생되지 않은 다른 저작물과 함께 단순히 저장하거나 배포할 목적으로 동일한 매체에 모아 놓은 집합물의 경우에는, 원프로그램으로부터 파생되지 않은 다른 저작물에는 본 허가서의 규정들이 적용되지 않습니다.

3. 피양도자는 다음 중 하나의 항목을 만족시키는 조건에 한해서 제 1 조와 제 2 조의 규정에 따라 프로그램 (또는 제 2 조에서 언급된 2 차적 프로그램) 을 목적 코드 (object code) 나 실행물 (executable form) 의 형태로 복제하고 배포할 수 있습니다.

   a) 목적 코드나 실행물에 상응하는 컴퓨터가 인식할 수 있는 완전한 원시 코드를 함께 제공해야 합니다. 원시 코드는 제 1 조와 제 2 조의 규정에 따라 배포될 수 있어야 하며, 소프트웨어의 교환을 위해서 일반적으로 사용되는 매체를 통해 제공되어야 합니다.

   b) 배포에 필요한 최소한의 비용만을 받고 목적 코드나 실행물에 상응하는 완전한 원시 코드를 배포하겠다는, 최소한 3 년간 유효한 약정서를 함께 제공해야 합니다. 이 약정서는 약정서를 갖고 있는 어떠한 사람에 대해서도 유효해야 합니다. 원시 코드는 컴퓨터가 인식할 수 있는 형태여야 하고 제 1 조와 제 2 조의 규정에 따라 배포될 수 있어야 하며, 소프트웨어의 교환을 위해서 일반적으로 사용되는 매체를 통해 제공되어야 합니다.

   c) 목적 코드나 실행물에 상응하는 원시 코드를 배포하겠다는 약정에 대해서 자신이 양도받은 정보를 함께 제공해야 합니다. (제 3 항은 위의 제 2 항에 따라 원시 코드를 배포하겠다는 약정을 프로그램의 목적 코드나 실행물과 함께 제공 받았고, 동시에 비상업적인 배포를 하고자 할 경우에 한해서만 허용됩니다.)

   저작물에 대한 원시 코드란 해당 저작물을 개작하기에 적절한 형식을 의미합니다. 실행물에 대한 완전한 원시 코드란 실행물에 포함된 모든 모듈들의 원시 코드와 이와 관련된 인터페이스 정의 파일 모두, 그리고 실행물의 컴파일과 설치를 제어하는데 사용된 스크립트 전부를 의미합니다. 그러나 특별한 예외의 하나로서, 실행물이 실행될 운영체제의 주요 부분 (컴파일러나 커널 등) 과 함께 (원시 코드나 바이너리의 형태로) 일반적으로 배포되는 구성 요소들은 이러한 구성 요소 자체가 실행물에 수반되지 않는 한 원시 코드의 배포 대상에서 제외되어도 무방합니다.

   목적 코드나 실행물을 지정한 장소로부터 복제해 갈 수 있게 하는 방식으로 배포할 경우, 동일한 장소로부터 원시 코드를 복제할 수 있는 동등한 접근 방법을 제공한다면 이는 원시 코드를 목적 코드와 함께 복제되도록 설정하지 않았다고 하더라도 원시 코드를 배포하는 것으로 간주됩니다.

4. 본 허가서에 의해 명시적으로 이루어 지지 않는 한 프로그램에 대한 복제와 개작 및 하위 허가권 설정과 배포가 성립될 수 없습니다. 이와 관련된 어떠한 행위도 무효이며 본 허가서가 보장한 권리는 자동으로 소멸됩니다. 그러나 본 허가서의 규정에 따라 프로그램의 복제물이나 권리를 양도받았던 제 3 자는 본 허가서의 규정들을 준수하는 한, 배포자의 권리 소멸에 관계없이 사용상의 권리를 계속해서 유지할 수 있습니다.

5. 본 허가서는 서명이나 날인이 수반되는 형식을 갖고 있지 않기 때문에 피양도자가 본 허가서의 내용을 반드시 받아들여야 할 필요는 없습니다. 그러나 프로그램이나 프로그램에 기반한 2 차적 프로그램에 대한 개작 및 배포를 허용하는 것은 본 허가서에 의해서만 가능합니다. 만약 본 허가서에 동의하지 않을 경우에는 이러한 행위들이 법률적으로 금지됩니다. 따라서 프로그램 (또는 프로그램에 기반한 2 차적 프로그램) 을 개작하거나 배포하는 행위는 이에 따른 본 허가서의 내용에 동의한다는 것을 의미하며, 복제와 개작 및 배포에 관한 본 허가서의 조건과 규정들을 모두 받아들이겠다는 의미로 간주됩니다.

6. 피양도자에 의해서 프로그램 (또는 프로그램에 기반한 2 차적 프로그램) 이 반복적으로 재배포될 경우, 각 단계에서의 피양도자는 본 허가서의 규정에 따른 프로그램의 복제와 개작 및 배포에 대한 권리를 최초의

양도자로부터 양도받은 것으로 자동적으로 간주됩니다. 프로그램 (또는 프로그램에 기반한 2 차적 프로그램) 을 배포할 때는 피양도자의 권리의 행사를 제한할 수 있는 어떠한 사항도 추가할 수 없습니다. 그러나 피양도자에게, 재배포가 일어날 시점에서의 제 3 의 피양도자에게 본 허가서를 준수하도록 강제할 책임은 부과되지 않습니다.

7. 법원의 판결이나 특허권 침해에 대한 주장 또는 특허 문제에 국한되지 않은 그밖의 이유들로 인해서 본 허가서의 규정에 배치되는 사항이 발생한다 하더라도 그러한 사항이 선행하거나 본 허가서의 조건과 규정들이 면제되는 것은 아닙니다. 따라서 법원의 명령이나 합의 등에 의해서 본 허가서에 위배되는 사항들이 발생한 상황이라도 양측 모두를 만족시킬 수 없다면 프로그램은 배포될 수 없습니다. 예를 들면, 특정한 특허 관련 허가가 프로그램의 복제물을 직접 또는 간접적인 방법으로 양도받은 임의의 제 3 자에게 해당 프로그램을 무상으로 재배포할 수 있게 허용하지 않는다면, 그러한 허가와 본 사용 허가를 동시에 만족시키면서 프로그램을 배포할 수 있는 방법은 없습니다.

   본 조항은 특정한 상황에서 본 조항의 일부가 유효하지 않거나 적용될 수 없을 경우에도 본 조항의 나머지 부분들을 적용하기 위한 의도로 만들어 졌습니다. 따라서 그 이외의 상황에서는 본 조항을 전체적으로 적용하면 됩니다.

   본 조항의 목적은 특허나 저작권 침해 등의 행위를 조장하거나 해당 권리를 인정하지 않으려는 것이 아니라, GPL 을 통해서 구현되어 있는 자유 소프트웨어의 배포 체계를 통합적으로 보호하기 위한 것입니다. 많은 사람들이 배포 체계에 대한 신뢰있는 지원을 계속해 줌으로써 소프트웨어의 다양한 분야에 많은 공헌을 해 주었습니다. 소프트웨어를 어떠한 배포 체계로 배포할 것인가를 결정하는 것은 전적으로 저작자와 기증자들의 의지에 달려있는 것이지, 일반 사용자들이 강요할 수 있는 문제는 아닙니다.

   본 조항은 본 허가서의 다른 조항들에서 무엇이 중요하게 고려되어야 하는 지를 명확하게 설명하기 위한 목적으로 만들어진 것입니다.

8. 특허나 저작권이 설정된 인터페이스로 인해서 특정 국가에서 프로그램의 배포와 사용이 함께 또는 개별적으로 제한되어 있는 경우, 본 사용 허가서를 프로그램에 적용한 최초의 저작권자는 문제가 발생하지 않는 국가에 한해서 프로그램을 배포한다는 배포상의 지역적 제한 조건을 명시적으로 설정할 수 있으며, 이러한 사항은 본 허가서의 일부로 간주됩니다.

9. 자유 소프트웨어 재단은 때때로 본 사용 허가서의 개정판이나 신판을 공표할 수 있습니다. 새롭게 공표될 판은 당면한 문제나 현안을 처리하기 위해서 세부적인 내용에 차이가 발생할 수 있지만, 그 근본 정신에는 변함이 없을 것입니다.

   각각의 판들은 판번호를 사용해서 구별됩니다. 특정한 판번호와 그 이후 판을 따른다는 사항이 명시된 프로그램에는 해당 판이나 그 이후에 발행된 어떠한 판을 선택해서 적용해도 무방하고, 판번호를 명시하고 있지 않은 경우에는 자유 소프트웨어 재단이 공표한 어떠한 판번호의 판을 적용해도 무방합니다.

10. 프로그램의 일부를 본 허가서와 배포 기준이 다른 자유 프로그램과 함께 결합하고자 할 경우에는 해당 프로그램의 저작자로부터 서면 승인을 받아야 합니다. 자유 소프트웨어 재단이 저작권을 갖고 있는 소프트웨어의 경우에는 자유 소프트웨어 재단의 승인을 얻어야 합니다. 우리는 이러한 요청을 수락하기 위해서 때때로 예외 기준을 만들기도 합니다. 자유 소프트웨어 재단은 일반적으로 자유 소프트웨어의 2 차적 저작물들을 모두 자유로운 상태로 유지시키려는 목적과 소프트웨어의 공유와 재활용을 증진시키려는 두가지 목적을 기준으로 승인 여부를 결정할 것입니다.

보증의 결여

11. 본 허가서를 따르는 프로그램은 무상으로 양도되기 때문에 관련 법률이 허용하는 한도 내에서 어떠한 형태의 보증도 제공되지 않습니다. 프로그램의 저작권자와 배포자가 공동 또는 개별적으로 별도의 보증을 서면으로 제공할 때를 제외하면, 특정한 목적에 대한 프로그램의 적합성이나 상업성 여부에 대한 보증을 포함한 어떠한 형태의 보증도 명시적이나 묵시적으로 설정되지 않은“있는 그대로의”상태로 이 프로그램을 배포합니다. 프로그램과 프로그램의 실행에 따라 발생할 수 있는 모든 위험은 피양도자에게 인수되며 이에 따른 보수 및 복구를 위한 제반 경비 또한 피양도자가 모두 부담해야 합니다.

12. 저작권자나 배포자가 프로그램의 손상 가능성을 사전에 알고 있었다 하더라도 발생된 손실이 관련 법규에 의해 보호되고 있거나 이에 대한 별도의 서면 보증이 설정된 경우가 아니라면, 저작권자나 프로그램을 원래의 상태 또는 개작한 상태로 제공한 배포자는 프로그램의 사용이나 비작동으로 인해 발생된 손실이나 프로그램 자체의 손실에 대해 책임지지 않습니다. 이러한 면책 조건은 사용자나 제 3 자가 프로그램을 조작함으로써

발생된 손실이나 다른 소프트웨어와 프로그램을 함께 동작시키는 것으로 인해서 발생된 데이터의 상실 및 부정확한 산출 결과에만 국한되는 것이 아닙니다. 발생된 손실의 일반성이나 특수성 뿐 아니라 원인의 우발성 및 필연성도 전혀 고려되지 않습니다.

GPL 에 대한 QGIS Qt 의 예외

추가로, 특별한 예외로서, QGIS 개발팀은 Qt 라이브러리를 사용한 이 프로그램의 코드를 링크할 수 있도록 허용합니다. QGIS 가 사용하는 무료 및 상용 Qt 라이브러리는 Qt/비 (□) 상용 윈도우, Qt/윈도우, Qt/X11, Qt/맥, 그리고 Qt/내장형 버전을 포함하지만, Qt 와 동일한 사용 허가서를 사용하는 수정된 Qt 버전도 포함할 수 있습니다. QGIS 개발팀은 이 둘을 포함한 결합물의 링크를 배포할 수 있도록 허용합니다. Qt 를 제외한, 사용된 모든 코드에 대한 모든 측면에서 GNU 일반 공중 사용 허가서를 따라야만 합니다. 사용자가 이 파일을 수정하는 경우 사용자 버전의 파일까지 이 예외를 적용할 수도 있지만 그렇게 해야 할 의무는 없습니다. 사용자 버전의 파일에 이 예외를 적용하고 싶지 않다면, 사용자 버전에서 이 예외 선언문을 삭제하십시오.

28.2 부록 B: GNU 자유 문서 사용 허가서

1.3 판, 2008 년 11 월 3 일

Copyright (C) 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.

https://fsf.org/

누구든지 본 사용 허가서를 있는 그대로 복제하고 배포할 수 있습니다. 그러나 본문에 대한 수정은 허용되지 않습니다. 추가로, 본 한국어 번역물은 참조용으로 법적 효력이 없으며, 영문 사용 허가서의 이해를 돕기 위해 번역된 문서임을 밝힙니다.

제 0 조. 전문

본 사용 허가서의 목적은 첫째, 매뉴얼이나 책 또는 다른 문서들을"자유"롭게 만들기 위한 것입니다. 여기서 말하는"자유"란 무료가 아닌 구속되지 않는다는 관점에서의 자유를 의미합니다. 즉, 상업적이든 비상업적이든 간에 누구나 그것을 수정하거나 그렇지 않은 상태에서 복제 및 재배포할 수 있는 실질적인 자유를 보장하기 위한 것입니다. 둘째, 본 사용 허가서는 저작자나 발행인에게 다른 사람들이 가한 수정에 책임지지 않고 그들의 저작물에 대한 공로를 인정 받을 수 있는 길을 보장하기 위한 것입니다.

본 사용 허가서는 일종의"카피레프트"입니다. 즉, 문서의 2 차적 저작물 또한 같은 의미에서 자유로워야 한다는 것을 의미합니다. 이것은 자유 소프트웨어를 위해서 고안된 카피레프트 사용 허가서인 GNU 일반 공중 사용 허가서를 보완합니다.

자유 소프트웨어는 자유 문서를 필요로 하기 때문에 본 사용 허가서는 자유 소프트웨어 매뉴얼에 사용되기 위해서 고안되었습니다. 자유 프로그램에는 소프트웨어에서와 같은 자유가 제공되는 매뉴얼이 함께 수반되어야 합니다. 그러나 본 사용 허가서가 단지 소프트웨어 매뉴얼에만 한정되는 것은 아닙니다. 이것은 문서의 주제나 그것이 인쇄물로 발행되었는지의 여부에 상관없이 모든 종류의 문서 저작물에 사용될 수 있습니다. 교육이나 참고를 목적으로 하는 저작물에는 원칙적으로 본 사용 허가서를 사용할 것을 추천합니다.

제 1 조. 적용 범위와 정의

본 사용 허가서는 GNU 자유 문서 사용 허가서 (이하, "GFDL"이라고 칭합니다.) 의 규정에 따라 배포될 수 있다는 사항이 저작권자에 의해서 명시된, 매체의 종류에 무관하게 제작된 모든 매뉴얼과 문서 저작물에 적용될 수 있으며, 공중의 누구라도 피양도자가 될 수 있습니다. 저작권자의 이러한 명시는 GFDL 의 조건들을 만족시키는 한, 저작물을 사용하는데 있어 사용료와 유효 기간이 없는 전세계적으로 유효한 허가를 부여합니다. 본 사용 허가서에서 사용되는 문서 (□□, document) 는 GFDL 에 의해서 양도된 매뉴얼 또는 문서 저작물을 의미합니다. 또한 공중 (□□, public) 이란 불특정 다수의 사람을 의미하고, 피양도자 (□□□□, licensee) 란 GFDL 에 의해서 매뉴얼 또는 문서 저작물을 양도받은 사람을 의미합니다. 복제 또는 개작의 대상이 되는 독자적인 문서 창작물의 최초 발행물은 원문서 (□□□, original document) 라고 합니다.

문서의 수정판 (□□□, modified version) 이란 원문서가 그대로 복제 또는 개작되거나, 다른 언어로 번역된 문서의 전부나 일부를 포함하고 있는 저작물을 의미합니다.

문서의 2 차구성부 (□□□□□, secondary section) 란 문서의 저작자나 발행인과 문서의 전체 주제 (또는 관련 내용) 과의 관계만을 설명하거나, 문서의 전체 주제와 직접적인 관계가 없는 내용을 포함하고 있는 머리말과 목차 등의 서두나 부록 부분을 가리킵니다. (따라서 문서가 수학 교과서의 일부였을 경우, 이 문서의 2 차구성부에는 수학적 설명이 포함된 부분이 들어 있지 않습니다.) 2 차구성부에 포함될 수 있는 내용은 문서의 주제나 관련 사항에 대한 개정 이력이나 법률적, 상업적, 철학적, 윤리적, 정치적 입장 등입니다.

변경 불가 부분 (□□□□□□, invariant section) 이란 문서가 본 사용 허가서에 의해서 배포된다는 사실이, 해당 부분이 변경 불가 부분이라는 제목과 함께 명시된 2 차 구성부의 한 형태를 의미합니다. 만약 2 차구성부의 정의에 부합되지 않는 부분 (section) 이 있다면, 이러한 부분은 변경 불가 부분으로 설정할 수 없습니다. 문서의 변경 불가 부분에는 내용이 없을 수도 있습니다. 만약 문서 안에 변경 불가 부분에 대한 구분이 명시되어 있지 않다면, 해당 문서에는 변경 불가 부분이 없는 것입니다.

표지구절 (□□□□, cover texts) 이란 문서가 본 사용 허가서에 의해서 배포된다는 사실이 문서의 앞 표지나 뒷 표지에 언급되는 짧은 문장을 의미합니다. 앞 표지구절은 5 음절 안팎으로, 그리고 뒷 표지구절은 25 음절 안팎으로 기술되어야 합니다.

문서의 투명 (□□, transparent) 복제물이란 문서를 수정할 수 있게 그 사양이 공중에게 공개되어 있고 일반적인 문서 편집기, (픽셀로 구성된 이미지의 경우) 일반적인 페인트 프로그램, (그림의 경우) 널리 사용되는 그림 편집기로 그 내용을 출력시키거나 직접 수정하기에 용이하며 조판 프로그램에 입력하기에 적당하거나 조판 프로그램에 입력할 수 있는 다양한 형태의 포맷으로 자동으로 번역되기에 용이한 형태로 만들어진 기계로 판독 가능한 복제물을 의미합니다. 투명 파일이라 하더라도 피양도자가 문서를 개작하는 것을 방해하거나 금지하기 위해서 기존의 마크업이나 마크업이 없는 상태를 변경시킨 파일 포맷은 투명한 것이 아닙니다. 투명 복제물이 아닌 것을 불투명 (□□□, opaque) 복제물이라고 합니다.

투명 복제물로 적절한 예는 마크업이 포함되지 않은 평범한 ASCII 포맷과 Texinfo 입력 포맷, LaTeX 입력 포맷, 공개적으로 이용되는 DTD 를 사용하는 SGML 이나 XML, 그리고 사람이 직접 개작할 수 있는 표준 규약을 준수하는 간단한 형식의 HTML, PostScript 또는 PDF 포맷입니다. 투명 이미지 포맷의 예로는 PNG, XCF 그리고 JPG 가 있습니다. 불투명 복제물에는 PostScript 와 PDF, 독점 워드 프로세서에서만 읽고 편집할 수 있는 독점 포맷, 일반적으로 통용되지 않는 DTD 와 처리 도구가 필요한 SGML 및 XML 포맷, 그리고 출력 목적만을 위해서 특정 워드 프로세서로 자동 생성한 HTML, PostScript 또는 PDF 포맷이 포함됩니다.

제목 페이지 (title page) 란 인쇄된 책의 경우에는 문서의 제목이 표시된 페이지 자체뿐만 아니라 보다 쉽게 이해되는데 필요하다고 판단되어 본 사용 허가서가 제목 페이지에 함께 포함시킬 것을 규정한 후속 페이지들을 모두 의미합니다. 제목 페이지가 없는 저작물의 경우, 제목 페이지는 본문이 시작되기 전에 저작물의 제목에 가장 근접한 형태가 나타난 페이지를 의미합니다.

발행인 (publisher) 이란 문서의 복제물을 공중에게 배포하는 한 개인 또는 한 단체를 의미합니다.

"XYZ 표제"부분은 그 제목이 정확히 XYZ 이거나 XYZ 를 다른 언어로 번역한 다음 괄호 안에 XYZ 를 포함시킨 형태로 표제를 붙인 전체 문서 중 일부분을 의미합니다.(XYZ 는"감사의 글", "헌사", "추천사"또는"개정이력"등과 같은 특정한 부분의 표제를 가리킵니다.) 문서를 개작할 때"표제를 보존"한다는 것은 이러한 정의에 따라"XYZ 표제"가 붙은 부분이 남아 있도록 한다는 것을 뜻합니다.

문서에 본 사용 허가서가 적용된다는 점을 명시한 부분 다음에"보증의 결여 (Warranty Disclaimer)"에 대한 부분을 추가시킬 수 있습니다. "보증의 결여"에 대한 부분은 문맥상 본 사용 허가서에 포함된 것으로 간주되지만, 그 의미는 보증이 결여된다는 사실 자체만에 국한됩니다. "보증의 결여"라는 말에 포함될 지 모를 다른 종류의 함축들은 모두 무효이며 본 사용 허가서의 효력에 영향을 주지 않습니다.

제 2 조. 동일 복제

문서의 피양도자는 본 사용 허가서에 어떠한 사항도 추가하지 않은 상태에서 본 사용 허가서와 저작권 사항 그리고 문서의 모든 복제물에 본 사용 허가서가 동일하게 적용된다는 사항을 명시하고 이를 문서의 복제물과 함께 제공하는 한, 어떠한 정보 매체에 의해서도 상업적이나 비상업적인 목적으로 문서를 복제하거나 배포할 수 있습니다. 문서를 복제하거나 배포할 경우에는 임의의 피양도자가 문서를 열람하거나 복제할 수 없도록 방해하거나 통제할 수 있는 어떠한 기술적 수단도 사용해서는 안됩니다. 그러나 복제물을 제공하는데 따른 보상을 청구할 수는 있습니다. 만약, 충분히 많은 양의 복제물을 배포할 경우에는 제 3 조의 규정들을 함께 준수해야만 합니다.

또한 위의 조건을 준수하는 한, 피양도자는 문서의 복제물을 대여하거나 공개적으로 전시할 수 있습니다.

제 3 조. 대량 복제

만약, 인쇄물 또는 일반적으로 인쇄된 표지를 갖고 있는 매체의 형태로 문서의 복제물을 100 부 이상 발행하며, 문서의 사용 허가서가 표지 구절의 사용을 규정하고 있는 경우에는 인쇄물의 앞 표지와 뒷 표지에 삽입될 앞 표지 구절과 뒷 표지 구절이 명확하고 읽기 쉬운 형태로 모든 복제물에 포함되어야 합니다. 또한 복제물의 발행인에 대한 정보가 양쪽 표지 모두에 명확하고 읽기 쉬운 형태로 명시되어야 합니다. 앞 표지에는 문서의 완전한 제목이, 제목을 구성하는 모든 문자들이 동일한 수준의 식별력을 가질 수 있도록 표시되어야 합니다. 표지에는 추가적인 문장이나 도형, 그림 등의 요소를 추가하는 것이 가능합니다. 문서의 제목이 유지되고 이러한 조건들을 만족하는 한, 표지만 변경시킨 복제물은 표지 이외의 다른 부분에 대한 동일 복제로 간주됩니다.

앞 표지나 뒷 표지에 표시될 표지 구절의 내용이 너무 많아서 읽기 힘든 경우에는 실제 표지에는 (적당한 만큼만) 기재하고, 나머지 내용들은 인접 페이지에 표시할 수 있습니다.

100 부 이상의 불투명 복제물을 발행하거나 배포하는 경우, 모든 불투명 복제물에 기계 판독이 가능한 투명 복제물을 함께 첨부하거나, 추가된 것이 없는 완벽한 투명 복제물이 있는 공개적인 접근이 가능한 컴퓨터 네트워크의 위치를 각 불투명 복제물에 명시하여 네트워크를 사용하는 일반 공중이 공개 표준을 준수하는 네트워크 프로토콜을 이용하여 비용없이 익명으로 다운로드 받을 수 있도록 해야 합니다. 후자의 경우, 불투명 복제물을 대량으로 배포하기 시작할 때에는 매우 신중한 접근을 해야 하는데, 불투명 복제물을 공중에게 (본인이 직접 또는 대리인을 통해서, 또는 소매업자를 통해서) 배포한 마지막 시점으로부터 적어도 1 년 뒤까지 투명 복제물이 명시된 위치에서 접근될 수 있는 상태로 확실히 남아있도록 해야 합니다.

강제 조항은 아니지만, 복제물을 대량으로 배포하기 전에 충분한 시간적 여유를 두고 문서의 저작자와 연락해서 저작자에게 문서의 최신 개정판을 제공할 수 있는 기회를 주어야 합니다.

제 4 조. 개작

문서의 수정판은 수정판이 명백하게 본 사용 허가서에 의해서 관리되는 조건 하에서 제 2 조와 제 3 조의 규정에 의해서 복제 및 배포될 수 있습니다. 즉, 수정판은 문서의 역할을 유지해야 하며, 수정판의 복제물을 양도받은 임의의 피양도자는 개작과 배포에 대한 동일한 권리를 양수받게 됩니다. 또한 수정판에 대해서 다음의 규정들을 준수해야만 합니다.

A. 수정판의 제목 페이지에는 (표지가 있다면 표지에도) 문서와 그 이전 판의 문서와 구별되는 제목을 사용해야 합니다. (문서의 개정 이력란이 존재한다면 이러한 사실이 등재되어야 합니다.) 그러나 이전 판의 발행인이 허락한다면, 이전 판과 같은 제목을 사용할 수 있습니다.

B. 원저작자들이 이 규정을 면제한 경우가 아니라면, 수정판의 제목 페이지에는 적어도 5 명의 문서의 원저작자 (5 명보다 적다면 원저작자 모두) 와 함께, 개작에 책임이 있는 1 인 이상의 개인 또는 단체를 저작자로 명시해야 합니다.

C. 수정판의 발행인 성명을 제목 페이지에 발행인으로서 명시합니다.

D. 문서의 모든 저작권 표시를 수정판에 유지해야 합니다.

E. 문서의 저작권 표시 부분에 자신이 개작한 것에 대한 적절한 저작권 사항을 추가해서 수정판에 표시합니다.

F. 저작권 표시 바로 다음에, 본 사용 허가서의 규정 하에 공중이 수정판을 사용할 수 있다는 사용 허가 표시를 본 사용 허가서의 부록에 나와있는 형식으로 포함시킵니다.

G. 문서의 사용 허가 표시에 포함되어 있던 변경 불가 부분의 목록과 명시할 것을 요구한 표지 구절을 수정판에도 모두 그대로 유지시킵니다.

H. 본 사용 허가서를 변경 없이 그대로 포함시킵니다.

I. "개정이력 (□□, history) 표제"부분과 이"표제를 보존"하고 적어도 수정판의 제목, 연도, 수정판 저작자, 발행인에 대한 항목을 제목 페이지에서 명시한 것과 동일하게 이 부분에 추가합니다. 문서에"개정이력 표제" 부분이 없을 경우에는 제목 페이지와 동일하게 문서의 제목과 연도, 저작자, 발행인을 명시한 이력 부분을 새롭게 만들고 앞에서 언급한 대로 수정판에 대한 사항을 추가합니다.

J. 공중이 투명 복제물에 접근할 수 있게 하기 위해서 문서에 명시한 네트워크 주소가 존재한다면 이를 수정판에도 그대로 유지시킵니다. 문서의 이전 판에 포함되어 있던 네트워크 주소가 문서에 기재되어 있는 경우에도, 이를 소급해서 수정판에 그대로 유지합니다. 네트워크 주소는"개정 이력 표제"부분에 기재될 수도

있습니다. 만일 네트워크 주소가 문서보다 적어도 4 년 전에 발행된 저작물을 위한 것이거나 네트워크 주소가 최초로 포함된 문서의 발행인이 허락했다면 이를 생략할 수 있습니다.

K. "감사의 글 (acknowledgements) 표제"또는"헌사 (dedications) 표제"부분이 있다면, "표제를 보전"하고 이 부분에 기재되어 있는 기여자에 대한 감사의 글과 헌사의 내용 및 어조를 수정판에도 모두 유지합니다.

L. 문서의 모든 변경 불가 부분은 제목과 본문을 변경하지 않고 수정판에 그대로 유지시킵니다. 장 (chapter) 또는 절 (section) 번호나 이에 상당하는 것은 변경 불가 부분의 제목의 일부분으로 간주되지 않습니다.

M. "추천사 (endorsements) 표제"부분은 수정판에서 모두 누락시킵니다. 이러한 부분이 수정판에 포함되어서는 안됩니다.

N. 기존의 어떠한 부분도 수정판에서"추천사 표제"부분으로 표제를 바꾸지 말고 제목을 개명하는 부분이 변경 불가 부분의 어떠한 표제와도 충돌되지 않도록 합니다.

O. "보증의 결여"부분은 모두 그대로 보존합니다.

만일 수정판이 문서에 포함되어 있지 않던 새로운 서두 부분이나 부록을 2 차 구성부의 형태로 포함하게 되면, 이러한 부분의 전체나 일부를 선택에 따라 변경 불가 부분으로 설정할 수 있습니다. 변경 불가 부분을 새롭게 설정하기 위해서는 수정판의 사용권 허가 표시 부분에 포함되어 있는 변경 불가 부분 목록에 제목을 추가시킵니다. 이때 그 제목들은 다른 부분의 제목들과 구별되어야 합니다.

수정판에만 한정된 추천사가 다양한 주체들에 의해서 제공될 경우에는, 예를 들어 동료들의 비평문이나 수정판을 특정한 표준의 권위있는 정의로 인정한다는 관련 기관의 승인이 있을 경우에는"추천사 표제"부분을 추가할 수 있습니다.

수정판의 표지 구절 목록 말미에는 앞 표지 구절과 뒷 표지 구절로 각각 5 단어와 25 단어 미만의 문장을 덧붙일 수 있습니다. 한 개인 또는 한 단체는 (또는 단체에 의해서 만들어진 협약을 통해서) 오직 한 개의 문장만을 각각 앞 표지 구절과 뒷 표지 구절에 추가할 수 있습니다. 만약 문서의 표지 구절에 이미 특정인이나 특정인이 대표하는 단체의 협약에 의해서 포함된 문장이 존재할 경우에는 동일인에 의해서 표지 구절 문장이 추가될 수 없습니다. 그러나 문서의 발행인으로부터 명시적인 승인을 받은 경우에는 기존의 문장을 수정판에서 새로운 문장으로 대체할 수 있습니다.

문서의 저작자 (들) 과 발행인 (들) 은 본 사용 허가서를 통해서 수정판을 선전하는데 그들의 이름이 사용되거나, 명시적 또는 묵시적인 형태로 그들의 이름이 수정판을 추천하는데 사용되는 것을 허용한 것은 아닙니다.

제 5 조. 문서의 결합

수정판에 대해서 정의된 제 4 조의 규정에 따라서 본 사용 허가서에 의해서 특정 문서를 다른 문서들과 결합할 수 있습니다. 단, 문서를 결합할 때는 결합 저작물을 구성하는 개별 문서들의 변경 불가 부분들을 결합 문서에 그대로 포함시켜야 하며 그 목록을 결합 저작물의 저작권 표시 부분에 명시해야 합니다. 또한 모든"보증의 결여"부분을 그대로 보존해야 합니다.

결합 저작물에는 본 사용 허가서의 복제물 1 부만 포함시키면 되며 여러 개의 동일한 변경 불가 부분 또한 하나로 통합될 수 있습니다. 만약, 동일한 이름을 갖는 변경 불가 부분이 여러 개 존재하지만 그 내용이 다른 경우에는 각각의 내용과 관련된 저작자와 발행자가 알려져 있을 경우에는 해당 정보를 각 부분의 말미에 괄호안에 명시하고 그렇지 않은 경우에는 숫자를 이용해서 구분합니다. 결합 저작물의 저작권 표시 부분에 있는 변경 불가 부분 목록에 포함된 제목도 같은 방식으로 조정합니다.

원문서에 존재하던"개정 이력 표제"부분은 모두 통합하여, 단일한"개정 이력 표제"부분을 결합 저작물 안에 유지해야 합니다. "감사의 글 표제"와"헌사 표제"부분도 같은 방식으로 조정합니다. 단, "추천사 표제"부분은 모두 삭제해야 합니다.

제 6 조. 문서의 수집

본 사용 허가서에 의해서 배포된 문서들을 모아서 구성된 수집 저작물을 만들 수 있습니다. 또한 개별 문서에 포함되어 있던 본 사용 허가서의 복제물들을 한 개로 대체하여 수집 저작물에 포함시킬 수 있습니다. 이 경우, 다른 모든 부분들은 본 사용 허가서에 규정된 제 2 조 동일 복제 규정을 준수해야 합니다.

수집 저작물로부터 하나의 문서를 발췌해서 개별 배포할 경우에는 본 사용 허가서의 복제물을 발췌한 문서에 첨부하고 그 이외의 다른 부분들은 모두 제 2 조에 규정된 동일 복제 조항을 준수해야 합니다.

제 7 조. 독자적 저작물과의 집합 저작물 구성

문서 또는 문서의 2 차적 저작물을 개별적이고 독자적인 다른 문서나 저작물과 함께 대량 저장 매체 또는 배포 매체에 구성한 편집물을 만들 경우, 저작물의 구성에 따른 편집 저작권이 개별 저작물이 허용한 범위를 넘어 편집물의 사용자의 법적 권리를 제한하는 데 사용되지 않는 한, 이러한 편집물을"집합 (aggregate) 저작물"이라고 부릅니다. 문서가 집합 저작물에 포함될 경우, 편집 과정에서 문서와 함께 구성된 독자적 저작물이 문서로부터 파생된 것이 아니라면 본 사용 허가서가 적용되지 않습니다.

제 3 조의 표지 구절에 대한 요구가 집합 저작물 안의 문서의 복제물에 적용되는 경우에는, 문서의 양이 전체 편집물의 1/2 보다 작은 경우에는 문서의 표지 구절은 편집물 안에서 문서가 위치해 있는 곳의 표지 부분, 또는 전자적 형태의 문서였을 경우에는 전자적으로 이에 해당하는 부분에 포함되어도 무방합니다. 그러나 그렇지 않은 경우에는 표지 구절이 전체 편집물의 인쇄된 표지 부분에 나타나야 합니다.

제 8 조. 번역

번역은 일종의 개작으로 간주됩니다. 따라서 문서의 번역물은 제 4 조의 규정에 따라 배포될 수 있습니다. 변경 불가 부분을 번역물로 대체하기 위해서는 저작권자의 명시적인 승인을 얻어야 합니다. 그러나 변경 불가 부분의 전체 또는 일부에 대한 번역문을 변경 불가 부분의 원문과 함께 표시할 경우에는 저작권자로부터 별도의 승인을 얻을 필요가 없습니다. 본 사용 허가서와 사용권 고지 사항 및 보증의 결여 부분에 대한 번역판을 첨부할 경우에는 본 사용 허가서와 사용권 고지 사항 및 보증의 결여 부분의 영문 원판들을 함께 제공해야 합니다. 영문 원판과 번역판 사이에 충돌이 발생할 경우에는 영문 원판이 우선합니다.

문서의 일부분이"감사의 글"과"헌사", "개정이력"표제를 갖고 있을 경우에는 (제 1 조에 따른) 표제를 보존시키는 (제 4 조의) 요건은 번역에 따라 전형적으로 실제 표제를 바꿀 필요가 있을 것입니다.

제 9 조. 권리의 소멸

본 허가서에 의해서 명시적으로 이루어지지 않는 한 문서에 대한 복제와 개작, 양도, 배포가 성립될 수 없습니다. 이와 관련된 어떠한 행위도 무효이며 본 허가서가 보장한 권리는 자동으로 소멸됩니다.

그러나 본 허가서의 규정 위반을 모두 중지하는 경우, 특정 저작권자가 피양도인에게 양도한 사용권은 (a) 저작권자가 명시적으로 그리고 최종적으로 피양도자의 사용권을 소멸시키지 않는 한 또는 소멸시킬 때까지 잠정적으로 (b) 저작권자가 위반 중지일 이후 60 일이 경과한 날짜까지 피양도자에게 위반 사실을 합당한 수단으로 고지하는 데 실패한 경우 영구히 회복됩니다.

특정 저작권자가 피양도인에게 양도한 사용권은 저작권자가 피양도인에게 위반 사실을 합당한 수단으로 고지하고, 피양도인이 해당 저작권자로부터 (어떤 저작물에 대해서든) 본 사용 허가서의 위반 사실을 고지받은 것이 처음이며, 피양도인이 고지를 받은 날짜로부터 30 일 이내에 위반 사실을 해결한 경우 영구히 회복됩니다.

본 허가서의 규정에 따라 피양도인으로부터 문서의 복제물이나 권리를 양도받았던 제 3 자는 피양도인의 권리 소멸에 관계없이 사용상의 권리를 계속해서 유지할 수 있습니다. 피양도인의 권리가 소멸되고 영구히 회복되지 않는 경우, 제 3 자에게 양도한 동일한 저작물의 일부 또는 전체의 복사물을 사용할 권리도 소멸됩니다.

제 10 조. 본 사용 허가서의 향후 개정

자유 소프트웨어 재단은 때때로 본 사용 허가서의 개정판이나 신판을 공표할 수 있습니다. 새롭게 공표될 판은 당면한 문제나 현안을 처리하기 위해서 세부적인 내용에 차이가 발생할 수 있지만, 그 근본 정신에는 변함이 없을 것입니다. https://www.gnu.org/copyleft/ 의 내용을 참고하시기 바랍니다.

본 사용 허가서의 각각의 판들은 판번호를 사용해서 구별됩니다. 본 사용 허가서의 특정한 판번호와"그 이후 판"을 따른다는 사항이 명시된 문서에는 해당 판이나 그 이후에 자유 소프트웨어 재단이 발행한 (초안이 아닌) 어떠한 판을 선택해서 적용해도 무방합니다. 문서가 판번호를 명시하고 있지 않은 경우에는 자유 소프트웨어 재단이 발행한 (초안이 아닌) 어떠한 판번호의 판을 적용해도 무방합니다. 문서에 대리인이 본 사용 허가서의 어떠한 향후 버전을 사용할 것인지 결정할 수 있다고 명시돼 있고, 해당 대리인이 어떠한 판번호를 공개적으로 선택한 경우, 해당 문서에 해당 판번호를 적용할 수 있는 권리를 영구히 갖게 됩니다.

제 11 조. 재사용 허가

"다중 저자 협력 프로젝트 사이트 (Massive Multiauthor Collaboration Site)"(이하"MMC 사이트") 란 저작권을 취득할 수 있는 저작물을 출판하며 모든 사람이 저작물을 편집할 수 있는 현저한 기능도 제공하는 월드 와이드 웹 서버를 의미합니다. 누구나 편집할 수 있는 공개 위키가 이러한 서버의 한 예입니다. 이러한 사이트가 담고 있는"다중 저자

협력 프로젝트 (Massive Multiauthor Collaboration)"(이하"MMC") 란 MMC 사이트 상에서 저작권을 취득할 수 있는 저작물들의 어떤 결합 저작물도 출판된다는 사실을 의미합니다.

"CC-BY-SA"란 크리에이티브 커먼즈 (Creative Commons Corporation) 가 발행한 저작자표시-동일조건변경허락 (Attribution-Share Alike) 3.0 사용 허가서 및 동일 단체가 발행하는 해당 사용 허가서의 향후 카피레프트 판을 의미합니다. 크리에이티브 커먼즈는 캘리포니아 주 샌프란시스코 시에 주요 사업장을 두고 있는 비영리 법인입니다.

"포함 (incorporate)"이란 문서 전체 또는 일부를 다른 문서의 일부로 발행 또는 재발행한다는 의미입니다.

MMC 가 본 사용 허가서의 규정을 따르며 모든 저작물들이 해당 MMC 가 아닌 다른 장소에서 본 사용 허가서를 따라 처음 발행되었고 나중에 해당 MMC 로 전체 또는 일부가 포함되었으며 (1) 앞 표지 구절 및 뒷 표지 구절 또는 변경 불가 부분이 없고, (2) 2008 년 11 월 1 일 이전에 포함된 경우라면"재사용 허가를 승인할 수 있습니다".

MMC 가 재사용 허가를 승인할 수 있는 경우라면 MMC 사이트의 운영자가 동일 사이트 상에 2009 년 8 월 1 일 이전에 사이트 안에 있던 MMC 를 CC-BY-SA 를 따라 재발행할 수도 있습니다.

부록: 문서에 GFDL 을 적용하는 방법

작성한 문서에 본 사용 허가서를 적용하기 위해서는 본 사용 허가서의 복제물을 문서에 첨부하고 다음과 같은 저작권 및 사용 허가 표시를 제목 페이지 다음에 추가합니다.

```
Copyright © YEAR  YOUR NAME.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.3
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
A copy of the license is included in the section entitled "GNU
Free Documentation License".
```

만약, 변경 불가 부분과 앞 표지 구절 및 뒷 표지 구절이 있는 경우라면 다음과 같이 변경할 수 있습니다.

```
with the Invariant Sections being LIST THEIR TITLES, with the
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
```

만약 변경 불가 부분만 있고 앞 표지 구절과 뒷 표지 구절이 없는 경우 또는 이 세 가지의 다른 조합인 경우라면 상황에 맞춰 앞의 두 가지 대안을 병합합니다.

만약, 문서가 프로그램 코드의 예를 상당 부분 포함하고 있다면 GNU 일반 공중 사용 허가서와 같은 자유 소프트웨어 사용 허가서를 사용해서 프로그램 코드가 자유소프트웨어에서 사용될 수 있도록 사용 허가를 병행할 것을 추천합니다.

28.3 부록 C: QGIS 파일 포맷

28.3.1 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
- 범례
- 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 28.1: 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 28.2: QGS 파일의 펼쳐진 최상위 수준 `ProjectLayers` 태그

28.3.2 QLR ? 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.

```
−<qlr>
  +<layer-tree-group name="" checked="Qt::Checked" expanded="1"></layer-tree-group>
  −<maplayers>
    −<maplayer autoRefreshEnabled="0" labelsEnabled="0" autoRefreshTime="0" readOnly="0" refreshOnNotifyMessage=""
     geometry="Line" simplifyDrawingTol="1" simplifyMaxScale="1" styleCategories="AllStyleCategories" simplifyDrawingHints="1"
     maxScale="0" simplifyLocal="1" hasScaleBasedVisibilityFlag="0" type="vector" refreshOnNotifyEnabled="0" minScale="1e+8"
     simplifyAlgorithm="0">
      +<extent></extent>
        <id>inputnew_6740bb2e_0441_4af5_8dcf_305c5c4d8ca7</id>
      +<datasource></datasource>
      +<keywordList></keywordList>
        <layername>inputnew</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 enableorderby="0" type="singleSymbol" forceraster="0" symbollevels="0"></renderer-v2>
      +<customproperties></customproperties>
        <blendMode>0</blendMode>
        <featureBlendMode>0</featureBlendMode>
        <layerOpacity>1</layerOpacity>
      +<geometryOptions removeDuplicateNodes="0" geometryPrecision="0"></geometryOptions>
      +<fieldConfiguration></fieldConfiguration>
      +<aliases></aliases>
        <excludeAttributesWMS/>
        <excludeAttributesWFS/>
      +<defaults></defaults>
      +<constraints></constraints>
      +<constraintExpressions></constraintExpressions>
        <expressionfields/>
      +<attributeactions></attributeactions>
      +<attributetableconfig sortExpression="" actionWidgetStyle="dropDown" sortOrder="0"></attributetableconfig>
      +<conditionalstyles></conditionalstyles>
        <editform tolerant="1">../src/qgisplugins/qgisbostaskdepplugin/data</editform>
        <editforminit/>
        <editforminitcodesource>0</editforminitcodesource>
        <editforminitfilepath/>
        <editforminitcode></editforminitcode>
        <featformsuppress>0</featformsuppress>
        <editorlayout>generatedlayout</editorlayout>
        <editable/>
        <labelOnTop/>
        <widgets/>
        <previewExpression>"FID"</previewExpression>
        <mapTip/>
    </maplayer>
  </maplayers>
</qlr>
```

그림 28.3: QLR 파일의 최상위 수준 태그들

28.3.3 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.)

28.4 부록 D: QGIS R 스크립트 문법

Scuola Superiore Sant'Anna 의 지원으로 마테오 게타 (Matteo Ghetta) 가 작성함

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
```

28.4.1 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`

28.4.2 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)

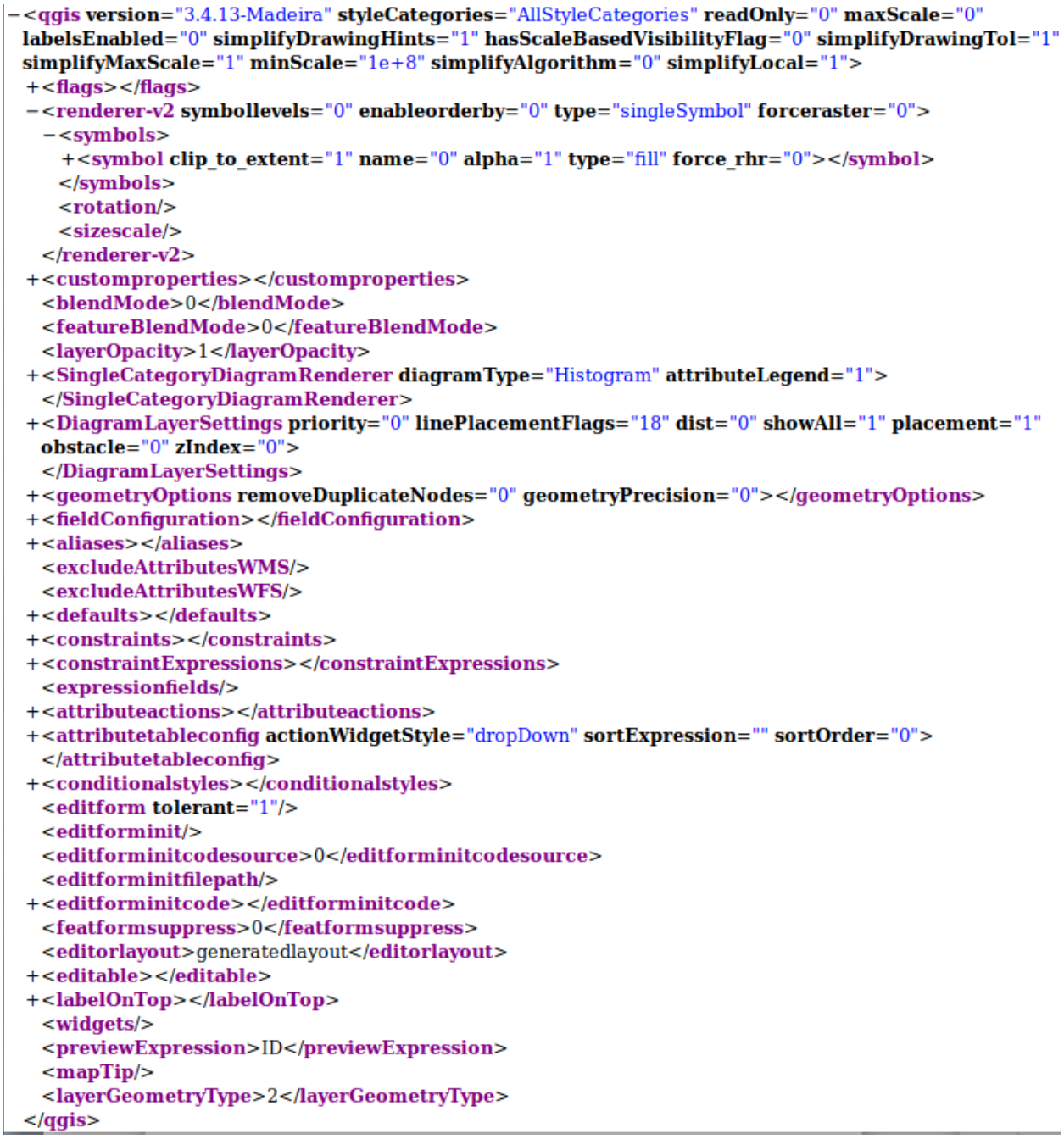


그림 28.4: QML 파일의 최상위 수준 태그들 (`renderer_v2` 및 그 하위 `symbol` 태그만 펼쳐져 있습니다)

- 결과물 뷰어 에 R 산출물을 표시하려면, 사용자가 표시하고자 하는 산출물을 생산하는 명령어 앞에 > 문자를 삽입하십시오.

28.4.3 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)
```

28.4.4 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]]),
    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 공간 처리 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.

CHAPTER 29

참고 문헌 및 웹사이트

GDAL-SOFTWARE-SUITE. 지리정보 데이터 추상화 라이브러리 (Geospatial data abstraction library). https://gdal.org, 2013.

GRASS-PROJECT. 지리 자원 분석 지원 시스템 (Geographic resource analysis support system). https://grass.osgeo.org, 2013.

NETELER, M., AND MITASOVA, H. 오픈소스 GIS: GRASS 의 GIS 접근 (Open source gis: A grass gis approach), 2008.

OGR-SOFTWARE-SUITE. 지리정보 데이터 추상화 라이브러리 (Geospatial data abstraction library). https://gdal.org, 2013.

OPEN-GEOSPATIAL-CONSORTIUM. 웹 맵 서비스 (1.1.1) 구현 사양 (Web map service (1.1.1) implementation specification). https://portal.opengeospatial.org, 2002.

OPEN-GEOSPATIAL-CONSORTIUM. 웹 맵 서비스 (1.3.0) 구현 사양 (Web map service (1.3.0) implementation specification). https://portal.opengeospatial.org, 2004.

POSTGIS-PROJECT. PostgreSQL 을 위한 공간 지원 (Spatial support for postgresql). http://postgis.refractions.net/, 2013.