



QGIS Documentation Guidelines

QGIS Project

3 月 15, 2026

1	A Step By Step Contribution	3
1.1	Using the GitHub web interface	4
1.1.1	Fork QGIS-Documentation	4
1.1.2	Make changes	4
1.1.3	Modify files	6
1.1.4	Share your changes via Pull Request	6
1.1.5	Delete your merged branch	9
1.2	Using Git command line tools	10
1.2.1	Local repository	10
1.2.2	Add another remote repository	11
1.2.3	Update your base branch	11
1.2.4	Contribute to your production branch	12
1.2.5	Share your Changes	13
1.2.6	Clean-up your local and remote repository	13
1.3	Further reading	13
2	编写指南	15
2.1	编写文档	15
2.1.1	标题	15
2.1.2	Lists	16
2.1.3	Indentation	16
2.1.4	Inline Tags	16
2.1.5	Labels/references	17
2.1.6	Figures and Images	18
2.1.7	Tables	20
2.1.8	Index	21
2.1.9	Special Comments	22
2.1.10	Code Snippets	22
2.1.11	Footnotes	22
2.2	Managing Screenshots	22
2.2.1	Add new Screenshots	22
2.2.2	Translated Screenshots	23
2.3	Documenting Processing algorithms	23
3	在 PyQGIS Cookbook 编写代码	27
3.1	How to write testable code snippets	27
3.1.1	Doctest sphinx directives	27
3.1.2	Grouping tests	28
3.2	How to test snippets on your local machine	29
4	文档指南	31

4.1	成为译者	31
4.1.1	Transifex	31
4.1.2	加入项目	32
4.1.3	翻译	32
4.2	翻译流程	32
4.3	翻译一个文件	33
4.3.1	在 Transifex 中进行翻译	34
4.3.2	在 Qt Linguist 中进行翻译	35
4.3.3	翻译手册	37
4.3.4	翻译规则总结	38
5	替换	39
5.1	用法	39
5.2	常见的替换	40
5.2.1	平台的图标	40
5.2.2	菜单项	40
5.3	工具栏按钮图标	40
5.3.1	管理图层和概览	40
5.3.2	工程	41
5.3.3	编辑	42
5.3.4	Identify result	42
5.3.5	数字化和高级数字化	42
5.3.6	网孔	43
5.3.7	地图导航和属性	44
5.3.8	选择和表达式	44
5.3.9	标签和图表	45
5.3.10	装饰	46
5.3.11	帮助	46
5.3.12	颜色	46
5.4	其他基本图标	46
5.5	属性工作表	47
5.6	投影和配准工具	48
5.7	打印布局	48
5.8	图层属性	49
5.9	插件	50
5.9.1	数据处理	50
5.9.2	各种核心插件	52
5.9.3	Grass 整合	52

QGIS Documentation is available at <https://www.qgis.org/resources/hub/#documentation>. As the writing process is going on, a build is automatically run every day (see bottom of the page for exact time) for all supported versions (testing, Long Term Release (LTR) and next-to-be LTR).

QGIS Documentation source files are available at <https://github.com/qgis/QGIS-Documentation>. They are mainly written using the reStructuredText (reST) format syntax, coupled with some scripts from the Sphinx toolset to post-process the HTML output.

以下章节将指导您学习：

- 如何使用 ‘git <<https://git-scm.com/>>’ 系统和储存文档的 GitHub 平台来管理文档源文件
- 如何修改文本，提供屏幕截图…依照规定的方式
- 如何共享并确保将您的修改推送到官方文档中。

If you are looking for general information about how to contribute to the QGIS project, you may find help at [Get Involved in the QGIS Community](#).

A Step By Step Contribution

Note

Though QGIS-Documentation is used to demonstrate the process, all commands and steps shown below also apply to QGIS-Website.

If you are reading these lines, it is certainly because you are willing to contribute to writing QGIS documentation and are looking for a how-to. You have come to the right place! The current document will guide you through the different ways to achieve this objective, showing you the main steps to follow, the tricks you can use and the traps you should be aware of.

For any help, do not hesitate to either ask in a comment on the issue report you are trying to fix or write to the [QGIS-community-team list](#). Read general information on [QGIS community support](#).

Let's now dive into the process.

Documentation sources are stored using the Git version control system and are available on GitHub at <https://github.com/qgis/QGIS-Documentation>. A list of issues to fix and features to explain can be found at <https://github.com/qgis/QGIS-Documentation/issues>.

Tip

If you are a first-time contributor and do not know where to start from, you may be interested in tackling our [welcoming issue reports](#).

There are two main ways, not mutually exclusive, to modify the files:

1. *Using the GitHub web interface*
2. *Using Git command line tools.*

1.1 Using the GitHub web interface

The GitHub web interface allows you to do the following:

- edit files
- preview and commit your changes
- make a pull request to have your changes inserted into the main repository
- create, update, or delete branches

If you are not yet familiar with Git and GitHub vocabulary, you may want to read the GitHub [Hello-world](#) project to learn some basic vocabulary and actions that will be used below.

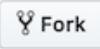
Note

If you are fixing a reported issue

If you are making changes to fix an [issue](#), add a comment to the issue report to assign it to yourself. This will prevent more than one person from working on the same issue.

1.1.1 Fork QGIS-Documentation

Assuming you already have a [GitHub account](#), you first need to fork the source files of the documentation.

Navigate to the [QGIS-Documentation repository](#) page and click on the  button in the upper right corner.

In your GitHub account you will find a QGIS-Documentation repository (<https://github.com/<YourName>/QGIS-Documentation>). This repository is a copy of the official QGIS-Documentation repository where you have full write access and you can make changes without affecting the official documentation.

1.1.2 Make changes

There are different ways to contribute to QGIS documentation. We show them separately below, but you can switch from one process to the other without any harm.

Alternative 1: Use the `Edit on GitHub` shortcut

Pages on the QGIS documentation website can be edited quickly and easily by clicking on the `Edit on GitHub` link:

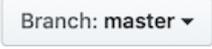
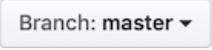
- at the top right of each page,
- or in the drop-down menu at the bottom of the left sidebar.

1. This will open the file in the `qgis:master` branch with a message at the top of the page telling you that you don't have write access to this repo and your changes will be applied to a new branch of your repository.

Note

If you do have write access to the QGIS repository, no message is displayed and your changes will likely be saved directly in the main repository instead of your fork. However, to keep the repository clean, it is recommended that even contributors with write access work from their forks when editing documentation.

2. Do your changes. Since the documentation is written using the reStructureText syntax, depending on your changes, you may need to rely on the *writing guidelines*.

3. Once you have finished modifying the file, press *Commit changes...* and add a short title explaining your changes. You can also provide a more detailed explanation, if necessary.
4. Press *Propose changes*, saving the modifications to your repository. This will generate a dedicated `branch` (`patch-xxx`) in your repository and the web browser will open the *Comparing changes* page.
 - If the issue you are addressing does not need any further changes, jump to the *Share your changes via Pull Request* section below.
 - If there are additional changes that you want to make before submitting them to QGIS, follow these steps:
 1. Navigate to your fork of QGIS-Documentation (<https://github.com/<YourName>/QGIS-Documentation>)
 2. Click on  and search for the new `patch-xxx` branch.
 3. Select it. The  button will now display *Branch: patch-xxx*.
 4. Jump down to the *Modify files* section below.

Alternative 2: Create an ad hoc branch in your documentation repository

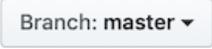
You can edit files directly from your fork of the QGIS Documentation.

First, make sure that your `master` branch is up to date with upstream, i.e. `qgis/QGIS-Documentation:master` branch. To do so:

1. Go to the main page of your repository, i.e. <https://github.com/<YourName>/QGIS-Documentation>. The `master` branch should be active with a mention whether it is up to date with `qgis/QGIS-Documentation:master` or not.

If it has commits ahead the upstream branch, you better use the previous *shortcut button alternative* until you align your `master` branch.

If it only has commits behind:

1. Expand the *Sync fork* drop-down menu on the right. You can:
 - *Compare* the branches and see new changes in the upstream branch you do not yet have in yours
 - *Fetch and merge*: brings changes from the upstream branch into yours.
2. Click *Fetch and merge*: after the process, your branch is mentioned as up to date with `qgis/QGIS-Documentation:master`.
2. Click on  in the upper left corner of your forked QGIS-Documentation repository and enter a unique name in the text field to create a new `branch`. It is advised that the name of the new branch relates to the problem you intend to fix. Convenient for identifying the branch later.
3. Press *Create branch <branch_name> from master*. The  button should now display `branch_name`.
4. You are ready to start new changes on top of your new branch.

Attention

Do your changes in an ad hoc branch, never in the `master` branch

By convention, avoid making changes in your `master` branch except when you merge the modifications from the `master` branch of `qgis/QGIS-Documentation` into your copy of the QGIS-Documentation repository. Separate branches allow you to work on multiple problems at the same time without interfering with other branches. If you make a mistake you can always delete a branch and start over by creating a new one from the `master` branch.

1.1.3 Modify files

1. Make sure the branch you want to modify is active: its name is displayed in the top left menu or in the page URL.
2. Browse the source files to the file that needs modifications.
3. Press the  Edit this file button.
4. Update the text following the *writing guidelines*
5. Once you have finished modifying the file, press *Commit changes...* and add a short title explaining your changes. You can also provide a more detailed explanation, if necessary.
6. Make sure *Commit directly to the <branch_name> branch* is selected.
7. Press *Commit changes* to save your modifications in the branch.
8. To update or add a new image file:
 1. Browse your branch to the folder of the file to update or add.
 2. Top right, use the *Add files ► Upload files* menu to select and load files from your drive. You can find hints for formatting an image for the docs at *Managing Screenshots*.
 3. If a file of the same name already exists, it gets overwritten.
 4. Commit once again your changes to the target branch.
9. Repeat the previous steps for any other files that need to be updated to fix the issue.

1.1.4 Share your changes via Pull Request

You need to make a [pull request](#) to integrate your changes into the official documentation.

Start a new pull request

1. If you used the [Edit on GitHub](#) link, after you commit your changes, GitHub automatically opens a new page comparing the changes you made in your `patch-xxx` branch to the `qgis/QGIS-Documentation:master` branch.
2. In all the cases, the comparing changes page can be reached by going to the "Pull requests" page of your repository and click on *New pull request*.

Compare changes

If you see two dialog boxes, one that says `base:master` and the other `compare:branch_name` (see figure), this will only merge your changes from one of your branches to your master branch. To fix this click on the *compare across forks* link.

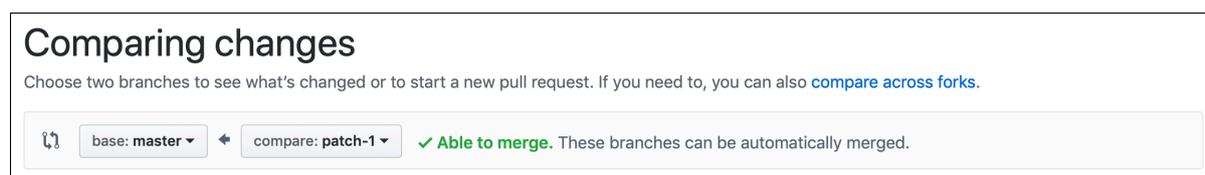


Fig. 1.1: If your *Comparing changes* page looks like this, click on the *compare across forks* link.

You should see four drop-down menus. These will allow you to compare the changes that you have made in your branch with the master branch that you want to merge into. They are:

- **base fork:** the fork that you want to merge your changes into
- **base:** the branch of the base fork that you want to merge your changes into
- **head fork:** the fork that has changes that you want to incorporate into the base fork
- **compare:** the branch with those changes

Select `qgis/QGIS-Documentation` as the base fork with `master` as base, set the head fork to your repository `<YourName>/QGIS-Documentation`, and set `compare` to your modified branch.

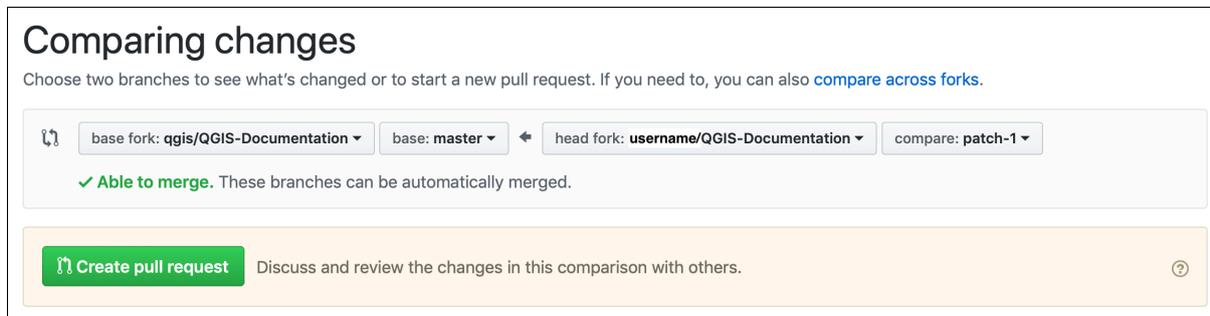


Fig. 1.2: Comparing changes between `qgis/QGIS-Documentation` and your repository

A green check with the words **Able to merge** shows that your changes can be merged into the official documentation without conflicts.

Click the *Create pull request* button.

Warning

If you see **✗ Can't automatically merge.**

This means that there are **conflicts**. The files that you are modifying are not up to date with the branch you are targeting because someone else has made a commit that conflicts with your changes. You can still create the pull request but you'll need to fix any *conflicts* to complete the merge.

Tip

Though being translated, the **latest version** of QGIS documentation is still maintained and existing issues are fixed. If you are fixing issues for a different release, change **base** from `master` to the appropriate `release_...` branch in the steps above.

Describe your pull request

A text box will open: fill in a title and any relevant description of the changes you are proposing.

If this relates to a particular *issue*, add the issue number to your comments. This is done by entering `#` and the issue number (e.g. `#1234`). If preceded by terms like `fix` or `close`, the concerned issue will be closed as soon as the pull request is merged.

Add links to any documentation pages that you are changing.

Click on *Create pull request*.

Review and comment pull request

As seen above, anyone can submit modifications to the documentation through pull requests. Likewise anyone can review pull requests with questions and [comments](#). Perhaps the writing style doesn't match the project guidelines, the change is missing some major details or screenshots, or maybe everything looks great and is in order. Reviewing helps to improve the quality of the contribution, both in form and substance.

To review a pull request:

1. Navigate to the [pull requests page](#) and click on the pull request that you want to comment on.
2. At the bottom of the page you will find a text box where you can leave general comments about the pull request.
3. To add comments about specific lines,

1. Click on  **Files changed** and find the file you want to comment on. You may have to click on < > Display the source diff to see the changes.
2. Scroll to the line you want to comment on and click on the . That will open a text box allowing you to leave a comment.

Specific line comments can be published either:

- as single comments, using the *Add single comment* button. They are published as you go. Use this only if you have few comments to add or when replying to another comment.
- or as part of a review, pressing the *Start a review* button. Your comments are not automatically sent after validation, allowing you to edit or cancel them afterwards, to add a summary of the main points of the review or global instructions regarding the pull request and whether you approve it or not. This is the convenient way since it's more flexible and allows you to structure your review, edit the comments, publish when you are ready and send a single notification to the repository followers and not one notification for each comment. [Get more details](#).



Fig. 1.3: Commenting a line with a change suggestion

Line comments can embed suggestions that the pull request writer can apply to the pull request. To add a suggestion, click the  **Insert a suggestion** button on top of the comment text box and modify the text within the suggestion block.

Tip

Prefer committing suggestions to your pull request as a batch

As a pull request author, when directly incorporating reviewers' feedback in your pull request, avoid using the *Commit suggestion* button at the bottom of the comment when you have many suggestions to address and prefer adding them as a batch commit, that is:

1. Switch to the  **Files changed** tab
2. Press *Add suggestion to batch* for each rewording you'd like to include. You will see a counter increasing as you go.
3. Press any of the *Commit suggestions* button when you are ready to apply the suggestions to your pull request, and enter a message describing the changes.

This will add all the modifications to your branch as a single commit, resulting in a more legible history of changes and less notifications for the repository followers. Incidentally, proceeding as this will also save you many clicks.

Make corrections

A new pull request will automatically be added to the [Pull requests list](#). Other editors and administrators will review your pull request and they may make suggestions or ask for corrections.

A pull request will also trigger automated build checks (eg, for rst formatting, python code syntaxes), and reports are displayed at the bottom of the page. If an error is found, a red cross will appear next to your commit. Click on the red cross or on *Details* in the summary section at the bottom of the pull request page to see the details of the error. You'll have to fix any reported errors or warnings before your changes are committed to the `qgis/QGIS-Documentation` repository.

You can make modifications to your pull request until it is merged with the main repository, either to improve your request, to address requested modifications, or to fix a build error:

1. Click on the  **Files changed** tab in your pull request page
2. Press ... menu on the right of the filename that you want to modify, and select *Edit file*.
3. The file opens in edit mode; add your modifications and commit.

Any additional changes will be automatically added to your pull request if you make those changes to the same branch that you submitted in your pull request. For this reason, you should only make additional changes if those changes relate to the issue that you intend to fix with that pull request.

If you want to fix another issue, create a new branch for those changes and repeat the steps above.

An administrator will merge your contribution after any build errors are corrected, and after you and the administrators are satisfied with your changes.

1.1.5 Delete your merged branch

You can delete the branch after your changes have been merged. Deleting old branches saves you from having unused and outdated branches in your repository.

1. Navigate to your fork of the QGIS-Documentation repository (<https://github.com/<YourName>/QGIS-Documentation>).
2. Click on the *Branches* tab. Below *Your branches* you'll see a list of your branches.
3. Click on the  *Delete this branch* icon to delete any unwanted branches.

1.2 Using Git command line tools

The GitHub web interface is an easy way to update the QGIS-documentation repo with your contributions, but it doesn't offer tools to:

- group your commits and clean your change history
- fix possible conflicts with the main repo
- build the documentation to test your changes

You need to [install Git](#) on your hard drive in order to get access to more advanced and powerful tools and have a local copy of the repository. Some basics you may often need are exposed below. You'll also find rules to care about even if you opt for the web interface.

In the code samples below, lines beginning with \$ show commands you should type while # are comments.

1.2.1 Local repository

Now you are ready to get a local clone of **your** QGIS-Documentation repository.

You can clone your QGIS repository using the web URL as follows:

```
# move to the folder in which you intend to store the local repository
$ cd ~/Documents/Development/QGIS/
$ git clone https://github.com/<YourName>/QGIS-Documentation.git
```

The former command line is simply an example. You should adapt both the path and the repository URL, replacing <YourName> with your GitHub user name.

Check the following:

```
# Enter the local repository
$ cd ./QGIS-Documentation
$ git remote -v
origin https://github.com/<YourName>/QGIS-Documentation.git (fetch)
origin https://github.com/<YourName>/QGIS-Documentation.git (push)
$ git branch
* master
```

- *origin* is the name of the remote repository of your QGIS-Documentation repository.
- *master* is the default main branch. You should never use it to contribute! **Never!**

Alternatively you can clone your QGIS repository using the SSH protocol:

```
# move to the folder in which you intend to store the local repository
$ cd ~/Documents/Development/QGIS/
$ git clone git@github.com:<YourName>/QGIS-Documentation.git
```

Tip

Permission denied (publickey) error?

If you get a Permission denied (publickey) error with the former command, there may be a problem with your SSH key. See [GitHub help](#) for details.

Check the following if you used the SSH protocol:

```
# Enter the local repository
$ cd ./QGIS-Documentation
$ git remote -v
origin git@github.com:<YourName>/QGIS-Documentation.git (fetch)
origin git@github.com:<YourName>/QGIS-Documentation.git (push)
$ git branch
* master
```

You can start to work here but in the long term process you will get a lot of issues when you will push your contribution (called Pull Request in GitHub process) as the master branch of the qgis/QGIS-Documentation repository will diverge from your local/remote repository. You then need to keep track of the main remote repository and work with branches.

1.2.2 Add another remote repository

To be able to follow the work in the main project, add a new remote repository in your local repository. This new remote repository is the QGIS-Documentation repository from QGIS project:

```
$ git remote add upstream https://github.com/qgis/QGIS-Documentation.git
$ git remote -v
origin https://github.com/<YourName>/QGIS-Documentation.git (fetch)
origin https://github.com/<YourName>/QGIS-Documentation.git (push)
upstream https://github.com/qgis/QGIS-Documentation.git (fetch)
upstream https://github.com/qgis/QGIS-Documentation.git (push)
```

Similarly, you can use the SSH protocol to add a remote repository in your local repository:

```
$ git remote add upstream git@github.com:qgis/QGIS-Documentation.git
$ git remote -v
origin git@github.com:<YourName>/QGIS-Documentation.git (fetch)
origin git@github.com:<YourName>/QGIS-Documentation.git (push)
upstream git@github.com:qgis/QGIS-Documentation.git (fetch)
upstream git@github.com:qgis/QGIS-Documentation.git (push)
```

So now you have the choice between two remote repository:

- *origin* to push your local branch in **your** remote repository
- *upstream* to merge (if you have right to do so) your contribution to the official one OR to update your master branch of local repository from the master branch of the official repository.

Note

upstream is just a label, a kind of standard name but you can call it as you want.

1.2.3 Update your base branch

Before working on a new contribution, you should always update your master branch in your local repository. Assuming you are willing to push changes to the testing documentation, run the following command lines:

```
# switch to master branch (it is easy to forget this step!)
$ git checkout master
# get "information" from the master branch in the upstream repository
# (aka qgis/QGIS-Documentation's repository)
$ git fetch upstream master
# merge update from upstream/master to the current local branch
# (which should be master, see step 1)
$ git merge upstream/master
```

(continues on next page)

(continued from previous page)

```
# update your remote repository (aka <YourName>/QGIS-Documentation)
$ git push origin master
```

Now you have your local and remote repositories which both have their `master` branch up to date with the official `master` branch of QGIS-Documentation. You can start to work on your contribution.

Note

Switch the branch if you wish to contribute to released doc

Along with the testing documentation, we continue to fix issues in the [latest release](#), meaning that you can also contribute to it. Follow the previous section sample code, replacing `master` with the corresponding branch of the latest documentation.

1.2.4 Contribute to your production branch

Now that your base branch is updated, you need to create a dedicated branch in which you add your contribution. Always work on a branch other than the base branch! Always!

```
# Create a new branch
$ git checkout -b myNewBranch
# checkout means go to the branch
# and -b flag creates a new branch if needed, based on current branch
# Let's check the list of existing branches (* indicates the current branch)
$ git branch
master
release_2.18
...
* myNewBranch
# You can now add your contribution, by editing the concerned file(s)
# with any application (in this case, vim is used)
$ vim myFile
# once done
$ git add myFile
$ git commit
```

Few words about commit/push commands:

- try to commit only one contribution (atomic change) i.e. address only one issue
- try to explain carefully what you change in the title of your commit and in the description. The first line is a title and should start by an upper case letter and have 80 characters length, don't end with a `..`. Be concise. Your description can be longer, end with a `.` and you can give much more details.
- use a `#` with a number to refer to an issue. Prefix with `Fix` if you fix the ticket: your commit will close the ticket.

Now that your changes are saved and committed in your local branch, you need to send them to your remote repository in order to create pull request:

```
$ git push origin myNewBranch
```

1.2.5 Share your Changes

Now you can go to your GitHub repository and *create a Pull Request* as exposed in a previous section. Ensure you create a PR from your branch to the remote branch you are targetting in the official QGIS-Documentation repository.

1.2.6 Clean-up your local and remote repository

After your PR has been merged into the official QGIS-Documentation, you can delete your branch. If you work a lot this way, in few weeks you will get a lot of unuseful branches. So keep your repository clean this way:

```
# delete local branch
$ git branch -d myNewBranch
# Remove your remote myNewBranch by pushing nothing to it
$ git push origin :myNewBranch
```

And do not forget to update the `master` branch in your local repository!

1.3 Further reading

- Other than the GitHub web interface and the Git command line tools exposed above, there are also [GUI applications](#) you can use to create and manage your contributions to the documentation.
- When the changes in the pull request are conflicting with recent changes pushed to the target branch, the conflicts need to be resolved before a merge is possible:
 - if the conflict relates to few competing lines, a *Resolve conflicts* button is available in the GitHub pull request page. Press the button and resolve the issue as explained at [Resolving a merge conflict on GitHub](#)
 - if the conflict involves files renaming or removal, then you'd need to resolve the conflict using Git command lines. Typically, you have to first rebase your branch over the target branch using `git rebase targetBranch` call and fix the conflicts that are reported. Read more at [Resolving a merge conflict using the command line](#)
- Sometimes, at the end of the proofreading process, you may end up with changes split into multiple commits that are not necessarily worth it. Git command lines help you squash these commits to a smaller number and more meaningful commit messages. Some details at [Using git rebase on the command line](#)

总的来说，当你为 QGIS 项目创建 reST 格式文档时，需要遵守 Python 文档样式指南。为了方便起见，我们提供了一套通用规则，用于编写下面的 QGIS 文档。

2.1 编写文档

2.1.1 标题

文档的每个网页对应一个 .rst 文件。

Sections used to structure the text are identified through their title which is underlined (and overlined for the first level). Same level titles must use same character for underline adornment. Precede section titles with a blank line. In QGIS Documentation, you should use following styles for chapter, section, subsection and minisec.

```
*****  
Chapter  
*****  
  
Section  
=====  
  
Subsection  
-----  
  
Minisec  
.....  
  
Subminisec  
^^^^^^^^^^
```

2.1.2 Lists

Lists are useful for structuring the text. Here are some simple rules common to all lists:

- Start all list items with a capital letter
- Do not use punctuation after list items that only contain a single simple sentence
- Use period (.) as punctuation for list items that consist of several sentences or a single compound sentence

2.1.3 Indentation

Indentation in reStructuredText should be aligned with the list or markup *marker*. It is also possible to create block quotes with indentation. See the [Specification](#)

```
#. In a numbered list, there should be
   three spaces when you break lines
#. And next items directly follow

* Nested lists
* Are also possible
* And when they also have
  a line that is too long,
  the text should be naturally
  aligned
* and be in their own paragraph

However, if there is an unindented paragraph, this will reset the numbering:

#. This item starts at 1 again
```

2.1.4 Inline Tags

You can use tags to emphasize items.

- **Menu GUI:** to mark a complete sequence of menu selections, including selecting submenus and choosing a specific operation, or any subsequence of such a sequence.

```
:menuselection:`menu --> submenu`
```

- **Dialogs and Tab titles:** Labels presented as part of an interactive user interface including window titles, tab titles, button and option labels.

```
:guilabel:`title`
```

- **Filenames and directories**

```
:file:`README.rst`
```

- **Icons with popup text**

```
|icon| :sup:`popup_text`
```

(see *image* below).

- **Keyboard shortcuts**

```
:kbd:`Ctrl+B`
```

will show Ctrl+B

When describing keyboard shortcuts, the following conventions should be used:

- Letter keys are displayed using uppercase: S
- Special keys are displayed with an uppercase first letter: Esc
- Key combinations are displayed with a + sign between keys, without spaces: Shift+R

• **User text**

```
``label``
```

• **Layers, databases, tables or columns names**

When referring to layers, databases, tables or columns, format as inline code:

```
``layer name``
``database_name``
``table_name``
``column_name``
```

2.1.5 Labels/references

Anchors inside the text can be used to create hyperlinks to sections or pages.

The example below creates the anchor of a section (e.g., Label/reference title)

```
.. _my_anchor:

Label/reference
-----
```

To call the reference in the **same page**, use

```
see my_anchor_ for more information.
```

which will return:

see *my_anchor* for more information.

Notice that it will jump to the line/thing following the 'anchor'. You do not need to use apostrophes, but you do need to have empty lines after the anchor.

Another way to jump to the same place **from anywhere in the documentation** is to use the `:ref:` role.

```
see :ref:`my_anchor` for more information.
```

which will create a link with the caption instead (in this case the title of this section!):

see *Labels/references* for more information.

So, reference 1 (*my_anchor*) and reference 2 (*Labels/references*). Because the reference often displays a full caption, it is not really necessary to use the word *section*. Note that you can also use a custom caption to describe the reference:

```
see :ref:`Label and reference <my_anchor>` for more information.
```

which returns:

see *Label and reference* for more information.

2.1.6 Figures and Images

Pictures

To insert an image, use

```
.. figure:: /static/common/logo.png
   :width: 10 em
```

which returns



Icon substitution

You can add an image inside a text paragraph (e.g., as a tool icon). To do so, you first need to create an alias (also named substitution), which is a reference name used to display the icon. To ensure consistency across the documents and help in the use of the icons, we maintain a list in the `substitutions.txt` file at the root of this repository. Find more details in the [替换](#) chapter.

Using an icon substitution is usually achieved through these steps:

1. Add the icon substitution in the `substitutions.txt` file as below. If a substitution already exists for the icon you want to use, then skip this step.

```
.. |logo| image:: /static/common/logo.png
   :width: 1 em
```

2. Call it in your paragraph:

```
My paragraph begins here with |logo|.
```

3. Add (again) the icon substitution at the end of the file you are modifying. This helps connecting the replacement text with the actual image, and can be done by copy-pasting it from `substitutions.txt` or by executing the `scripts/find_set_subst.py` script.

This is how the example will be displayed:

My paragraph begins here with .

Figure

```
.. _figure_logo:
.. figure:: /static/common/logo.png
   :width: 20 em
   :align: center

A caption: A logo I like
```

The result looks like this:



Fig. 2.1: A caption: A logo I like

To avoid conflicts with other references, always begin figure anchors with `_figure_` and use terms that easily connect to the figure caption. While only the centered alignment is mandatory for the image, feel free to use any other options for figures (such as `width`, `height`, `scale...`) if needed.

The scripts will insert an automatically generated number before the caption of the figure in the generated HTML and PDF versions of the documentation.

To use a caption (*see My caption*) just insert indented text after a blank line in the figure block.

A figure can be referenced using the reference label like this:

```
see :numref:`figure_logo`
```

renders like this:

see Fig. 2.1

This is the preferred way of referencing figures.

Note

For `:numref:` to work, the figure **must have a caption**.

It is possible to use `:ref:` instead of `:numref:` for reference, but this returns the full caption of the image.

```
see :ref:`figure_logo`
```

renders like this:

see *A caption: A logo I like*

2.1.7 Tables

You can make a simple table like this:

```

=====
x          y          z
=====
1          2          3
4          5          5
=====
    
```

It will render like this:

x	y	z
1	2	3
4	5	5

Tables should have a caption. You can use an explicit reST `table` directive to add a caption to simple tables or grid tables. Add the caption on the same line as the directive and indent the table at least one space.

You can also add a [hyperlink target](#) before a table in order to reference it elsewhere. To avoid conflicts with other references, always begin hyperlink targets with `_table_` and use terms relevant to the table caption.

Here is an example of a more complicated grid table with a caption and a hyperlink target:

```

.. _table-grid-caption:
.. table:: Grid table with caption

+-----+-----+
| Windows | macOS |
+-----+-----+
| |win|   | |osx| |
+-----+-----+
| and of course not to forget |nix| |
+-----+-----+
    
```

其结果为:

Table 2.1: Grid table with caption

Windows	macOS
	X
and of course not to forget 	

You may find it easier to use [list tables](#) or [CSV tables](#) to make complex tables. Add the caption after the `list-table` or `csv-table` directive.

Here is an example of a list table with a caption and a hyperlink target:

```

.. _table-list-caption:
.. list-table:: List table with caption
:header-rows: 1
:widths: 20 20 20 40

* - What
  - Purpose
  - Key word
    
```

(continues on next page)

(continued from previous page)

```

- Description
* - **Test**
- ``Useful test``
- complexity
- Geometry. One of:

  * Point
  * Line
    
```

其结果为:

Table 2.2: List table with caption

What	目标	Key word	描述
Test	Useful test	complexity	Geometry. One of: <ul style="list-style-type: none"> • 点 • Line

Use `:numref:` roles to reference tables like this:

```
see :numref:`table-grid-caption` or :numref:`table-list-caption`
```

其结果为:

see [Table 2.1](#) or [Table 2.2](#)

Note

You must add a caption to your table in order to create a cross reference with the `:numref:` role.

2.1.8 Index

An index is a handy way to help the reader find information in a document. QGIS documentation provides some essential indices. There are a few rules that help us provide a set of indices that are really useful (coherent, consistent and really connected to each other):

- An index should be human readable, understandable and translatable; an index can be made from many words but you should avoid any unneeded `_`, `-`... characters to link them i.e., `Loading layers` instead of `loading_layers` or `loadingLayers`.
- Capitalize only the first letter of the index unless the word has a particular spelling. E.g., `Loading layers`, `Atlas generation`, `WMS`, `pgsql2shp`.
- Keep an eye on the existing [Index list](#) in order to reuse the most convenient expression with the right spelling and avoid unnecessary duplicates.

Several index tags exist in reST. You can use the inline `:index:` tag within normal text:

```
QGIS can load several :index:`Vector formats` supported by GDAL ...
```

Or you can use the `.. index::` block-level markup which links to the beginning of the next paragraph. Because of the rules mentioned above, it is recommended to use the block-level tag:

```
.. index:: WMS, WFS, Loading layers
```

It is also recommended to use index parameters such as `single`, `pair` and `see`, in order to build a more structured and interconnected index table. See [Index generating](#) for more information on index creation.

2.1.9 Special Comments

Sometimes, you may want to emphasize some points of the description, either to warn, remind or give some hints to the user. In QGIS Documentation, we use reST special directives such as `.. warning::`, `.. seealso::`, `.. note::` and `.. tip::`. These directives generate frames that highlight your comments. See [Paragraph Level markup](#) for more information. A clear and appropriate title is required for both warnings and tips.

```
.. tip:: **Always use a meaningful title for tips**
```

Begin tips with a title that summarizes what it is about. This helps users to quickly overview the message you want to give them, and decide on its relevance.

2.1.10 Code Snippets

You may also want to give examples and insert code snippets. In this case, write the comment below a line with the `::` directive inserted. For a better rendering, especially to apply color highlighting to code according to its language, use the code-block directive, e.g. `.. code-block:: xml`. More details at [Showing code](#).

Note

Text in special comments will be translated, but text in code-block frames will not be translated. So keep comments in code blocks as short as possible and avoid comments unrelated to code.

2.1.11 Footnotes

This is for creating a footnote (showing as example¹)

```
blabla [1]_
```

Which will point to:

2.2 Managing Screenshots

2.2.1 Add new Screenshots

Here are some hints to create new, nice looking screenshots. The images should be placed in an image (`img/`) folder that is located in the same folder as the referencing `.rst` file.

- You can find some prepared QGIS projects that are used to create screenshots in the `./qgis-projects` folder of this repository. This makes it easier to reproduce screenshots for the next version of QGIS. These projects use the QGIS [Sample Data](#) (aka Alaska Dataset), which should be unzipped and placed in the same folder as the QGIS-Documetation Repository.
- Reduce the window to the minimal space needed to show the feature (taking the whole screen for a small modal window > overkill)
- The less clutter, the better (no need to activate all the toolbars)
- Don't resize them in an image editor; the size will be set into the `.rst` files if necessary (downscaling the dimensions without properly upping the resolution > ugly)
- Cut the background
- Make the top corners transparent if the background is not white

¹ Updates of core plugins

- Set print size resolution to 135 dpi (e.g., in GIMP scale down the image using *Image* ► *Scale Image* and setting "X/Y" to 135 pixels/in, and export it through *File* ► *Export...*). This way, images will be at original size in HTML and at a good print resolution in the PDF.

You can also use ImageMagick convert command to do a batch of images:

```
convert -units PixelsPerInch input.png -density 135 output.png
```

- Save them as .png (to avoid .jpeg artifacts)
- The screenshot should show the content according to what is described in the text

 **Tip**

If you are on Ubuntu, you can use the following command to remove the global menu function and create smaller application screens with menus:

```
sudo apt autoremove appmenu-gtk appmenu-gtk3 appmenu-qt
```

2.2.2 Translated Screenshots

Here are some additional hints for those that want to create screenshots for a translated user guide:

Translated images should be placed in a `img/<your_language>/` folder. Use the same filename as the 'original' English screenshot.

2.3 Documenting Processing algorithms

If you want to write documentation for Processing algorithms, consider these guidelines:

- Processing algorithm help files are part of QGIS User Guide, so use the same formatting as the User Guide and other documentation.
- Each algorithm documentation should be placed in the corresponding **provider** folder and **group** file, e.g. the algorithm *Voronoi polygon* belongs to the *QGIS* provider and to the group *vectorgeometry*. So the correct file to add the description is: `source/docs/user_manual/processing_algs/qgis/vectorgeometry.rst`.

 **Note**

Before starting to write the guide, check if the algorithm is already described. In this case, you can enhance the existing description.

- It is **extremely** important that each algorithm has an *anchor* that corresponds to the provider name + the unique name of the algorithm itself. This allows the Help button to open the Help page of the correct section. The anchor should be placed **above** the title, e.g. (see also the *Labels/references* section):

```
.. _qgisvoronoipolygons:
Voronoi polygons
-----
```

To find out the algorithm name you can just hover the mouse on the algorithm in the Processing toolbox.

- Avoid using "This algorithm does this and that..." as the first sentence in the algorithm description. Try to use more general expressions like:

Takes a point layer and generates a polygon layer containing the...

- Avoid describing what the algorithm does by replicating its name and please don't replicate the name of the parameter in the description of the parameter itself. For example if the algorithm is Voronoi polygon consider to describe the Input layer as Layer to calculate the polygon from.
- Indicate in the description whether the algorithm has a default shortcut in QGIS or supports in-place editing.
- Add images! A picture is worth a thousand words! Use .png format and follow the general guidelines for documentation (see the *Figures and Images* section for more info). Put the image file in the correct folder, i.e. the img folder next to the .rst file you are editing.
- If necessary, add links in the "See also" section that provide additional information about the algorithm (e.g., publications or web-pages). Only add the "See also" section if there is really something to see. As a good practice, the "See also" section can be filled with links to similar algorithms.
- Give clear explanation for algorithm parameters and outputs: take inspiration from existing algorithms.
- Avoid duplicating detailed description of algorithm options. Add this information in the parameter description.
- Avoid adding information about the vector geometry type in the algorithm or parameter description, as this information is already available in the parameter descriptions.
- Add the default value of the parameter, e.g.:

```
* - **Number of points**
- ``NUMBER_OF_POINTS``
- [numeric: integer]

Default: 1
- Number of points to create
```

- Describe the *type* of input supported the parameters. There are several types available you can pick one from:

Parameter/Output type	描述	Visual indicator
Point vector layer	vector: point	
Line vector layer	vector: line	
Polygon vector layer	vector: polygon	
All spatial vector layers	vector: geometry	
Geometryless vector layer	vector: table	
Generic vector layer	vector: any	
Vector field numeric	tablefield: numeric	1.2
Vector field string	tablefield: string	abc
Vector field generic	tablefield: any	
Raster layer	raster	
Raster band	raster band	
HTML file	html	
Expression	expression	ε
Line geometry	geometry: line	
Point geometry	coordinates	
Extent	extent	
CRS	crs	
Enumeration	enumeration	
List	list	
Integer value	numeric: integer	<input type="text" value="1,00"/>
Decimal value	numeric: double	<input type="text" value="1,00"/>

continues on next page

Table 2.3 – continued from previous page

Parameter/Output type	描述	Visual indicator
字符串	string	Display name <input type="text" value="lakes.shp"/>
布尔	boolean	<input checked="" type="checkbox"/>
Folder path	folder	
文件	file	
Matrix	matrix	
图层	layer	
Same output type as input type	same as input	
Definition	definition	
Map layers	layer list	
范围	range	
AuthConfig	authconfig	
网孔	mesh	
Layout	layout	
LayoutItem	layoutitem	
Color	color	
Scale	scale	
Map theme	map theme	

- Study an existing and well documented algorithm, and copy all the useful layouts.
- When you are finished, just follow the guidelines described in *A Step By Step Contribution* to commit your changes and make a Pull Request

Here is an example of an existing algorithm to help you with the layout and the description:

```

.. _qgiscountpointsinpolygon:

Count points in polygon
-----
Takes a point and a polygon layer and counts the number of points from the
point layer in each of the polygons of the polygon layer.
A new polygon layer is generated, with the exact same content as the input
polygon layer, but containing an additional field with the points count
corresponding to each polygon.

.. figure:: img/count_points_polygon.png
:align: center

The labels in the polygons show the point count

An optional weight field can be used to assign weights to each point.
Alternatively, a unique class field can be specified. If both options
are used, the weight field will take precedence and the unique class field
will be ignored.

``Default menu``: :menuselection: `Vector --> Analysis Tools`

Parameters
.....

.. list-table::
:header-rows: 1
:widths: 20 20 20 40
:class: longtable

* - Label
  - Name
    
```

(continues on next page)

```

- Type
- Description
* - **Polygons**
- ``POLYGONS``
- [vector: polygon]
- Polygon layer whose features are associated with the count of
  points they contain
* - **Points**
- ``POINTS``
- [vector: point]
- Point layer with features to count
* - **Weight field**

  Optional
  - ``WEIGHT``
  - [tablefield: numeric]
  - A field from the point layer.
    The count generated will be the sum of the weight field of the
    points contained by the polygon.
* - **Class field**

  Optional
  - ``CLASSFIELD``
  - [tablefield: any]
  - Points are classified based on the selected attribute and if
    several points with the same attribute value are within the
    polygon, only one of them is counted.
    The final count of the points in a polygon is, therefore, the
    count of different classes that are found in it.
* - **Count field name**
- ``FIELD``
- [string]

  Default: 'NUMPOINTS'
- The name of the field to store the count of points
* - **Count**
- ``OUTPUT``
- [vector: polygon]

  Default: [Create temporary layer]
- Specification of the output layer type (temporary, file,
  GeoPackage or PostGIS table).
  Encoding can also be specified.

```

Outputs

.....

```

.. list-table::
   :header-rows: 1
   :widths: 20 20 20 40

```

```

* - Label
  - Name
  - Type
  - Description
* - **Count**
- ``OUTPUT``
- [vector: polygon]
- Resulting layer with the attribute table containing the
  new column with the points count

```

在 PyQGIS Cookbook 编写代码

If you are planning to add or update some chapters of the PyQGIS-Developer-Cookbook, then you should follow some rules to enable automatic testing of the code snippets.

Testing is really important because it allows automatic checking of the code. Code snippets with errors or code that uses outdated methods will fail and the notification will help you fix the problems.

For testing, we use the [Sphinx doctest extension](#). Refer to the extension documentation for more detailed information.

3.1 How to write testable code snippets

Writing testable code snippets is not so different from the *old* method. Basically, you need to use a different Sphinx *directive*.

3.1.1 Doctest sphinx directives

Instead of embedding the code in a `.. code-block:: python` directive (which would highlight the code syntax automatically), you now need to embed it in a `.. testcode::`. That is, instead of this:

```
.. code-block:: python

    crs = QgsCoordinateReferenceSystem("EPSG:4326")
    assert crs.isValid()
```

You now use this:

```
.. testcode::

    crs = QgsCoordinateReferenceSystem("EPSG:4326")
    assert crs.isValid()
```

After you wrote the example code, you should add some *assertion* that will evaluate the code and that will be run automatically.

In the above example, you are creating a `crs` and with `assert crs.isValid()` you **test** if it is valid. If the code has a wrong python syntax or the `crs.isValid()` returns `False`, this code snippet will fail during testing.

To successfully run the tests on snippets, you must import all the classes and declare any variables used in the code snippets. You can include those in the code snippet itself (visible in the HTML pages) or you can add them to a `.. testsetup::` directive (hidden in the HTML pages). The `.. testsetup::` needs to be placed before the `.. testcode::`:

```
.. testsetup::  
  
    from qgis.core import QgsCoordinateReferenceSystem  
  
.. testcode::  
  
    crs = QgsCoordinateReferenceSystem("EPSG:4326")  
    assert crs.isValid()
```

If the code snippet doesn't create objects (and therefore you cannot use something like `assert object.isValid()`), you can test the code using the `print()` method, then add the expected results within a `.. testoutput::` directive to compare the expected output:

```
.. testcode::  
  
    print("QGIS CRS ID:", crs.srsid())  
    print("PostGIS SRID:", crs.postgisSrid())  
  
.. testoutput::  
  
    QGIS CRS ID: 3452  
    PostGIS SRID: 4326
```

By default, the content of `.. testoutput::` is shown in the HTML output. To hide it from the HTML use the `:hide:` option:

```
.. testoutput::  
    :hide:  
  
    QGIS CRS ID: 3452  
    PostGIS SRID: 4326
```

Note

If the code snippet contains any print statements, you **MUST** add a `testoutput` with the expected outputs; otherwise the test will fail.

3.1.2 Grouping tests

For each reST document, the code snippets are tested sequentially, which means you can use one `.. testsetup::` for all the following code snippets and that later snippets will have access to variables declared in earlier ones in the document.

Alternatively, you can use groups to break down the examples on the same page in different tests.

You add the code snippet to groups by adding one or more group names (separated by commas) in the respective directive:

```
.. testcode:: crs_crsfromID [, morenames]  
  
    crs = QgsCoordinateReferenceSystem("EPSG:4326")  
    assert crs.isValid()
```

The `doctest` will pick each group snippets and run them independently.

Note

Use group names that make sense with the related content. Use something similar to <chapter>_<subchapter>, for example: crs_intro, crs_fromwkt. In case of failures, this will help identifying where the failures occur.

If you don't declare any group, the code snippet will be added to a group named `default`. If instead, you use `*` as a group name, the snippet will be used in all testing groups, something normally useful to use in the test setup:

```
.. testsetup:: *
    from qgis.core import QgsCoordinateReferenceSystem
```

3.2 How to test snippets on your local machine

Note

Instructions are valid for Linux system.

To test Python code snippets, you need a *QGIS* installation. For this, there are many options. You can:

- Use your system *QGIS* installation with *Sphinx* from a Python virtual environment:

```
make -f venv.mk doctest
```

- Use a manually built installation of *QGIS*. You'd need to:

1. Create a custom Makefile extension on top of the `venv.mk` file, for example a `user.mk` file with the following content:

```
# Root installation folder
QGIS_PREFIX_PATH = /home/user/apps/qgis-master

include venv.mk
```

Or

```
# build output folder
QGIS_PREFIX_PATH = /home/user/dev/QGIS-build-master/output

include venv.mk
```

2. Then, use it to run target `doctest`:

```
make -f user.mk doctest
```

- Run target `doctest` inside the official *QGIS* docker image:

```
make -f docker.mk doctest
```

You have to install [Docker](#) first because this uses a docker image with *QGIS* in it.

本手册旨在帮助译者。首先说明如何加入翻译团队，然后介绍技术层面的翻译流程，接着通过一个从英文 .rst 文档翻译成荷兰语的实际案例进行说明，最后总结了翻译规则。

Note

尽管这些指南聚焦于 QGIS 文档，但下文所述的方法和规则同样适用于 QGIS 应用程序。

4.1 成为译者

QGIS 项目始终在寻找愿意投入更多时间将 QGIS 翻译成外语的人士——他甚至可能负责协调整个翻译工作。

我们正在努力改进我们的项目管理过程，并在每个人都有特定责任领域的人之间更均匀地分配负载，因此您必须做出的任何贡献都将受到极大的赞赏。

如果您希望自荐担任某种新语言的协调员，请尽管申请。若多人同时申请同一语言的协调员职位，请相互联系并协商如何分工合作。

4.1.1 Transifex

网上翻译平台 [Transifex](#) 适用于所有 QGIS 翻译：桌面应用程序本身（或 GUI）和文档。所以你的第一件事就是需要一个登录和开始的帐户。

4.1.2 加入项目

1. 前往 <https://explore.transifex.com/qgis/>
2. 您可以浏览我们正在翻译的项目，并通过各种统计数据了解其目标语言：
 - **QGIS Desktop**：用于 QGIS 应用程序（QGIS 桌面版和 QGIS 服务器）中所有文本内容，
 - **QGIS Documentation**：用于“官方长期支持文档 LTR <<https://docs.qgis.org/latest/>>”
3. 点击您希望协助翻译的项目
4. 点击右侧的:guilabel:‘JOIN THIS PROJECT’（加入此项目），系统将提示您注册。
5. 创建您的账户，或使用第三方平台账户登录。请通过您收到的邮件中的链接验证账户。
6. 登录
7. 随后会弹出窗口，供您选择希望协助翻译的目标语言。请注意，为简化流程，我们仅列出目标语言（例如 French (fr)），而**不**包含国家参数（如 French (France) (fr_FR)）。仅当语言存在显著差异时（例如葡萄牙的葡萄牙语与巴西的葡萄牙语），才可能允许不同版本。

搜索您的目标语言，即您希望协助将 QGIS 翻译成的语言，**不一定是**您会说的所有语言：

- 如果显示为“already added”（已添加），请选择该语言并点击:guilabel:‘Join Project’（加入项目）。
 - 如果未标记为“已添加”，请选择该语言并点击 *Request language*（申请语言）。请谨记：翻译整个项目可能需要数天甚至数周的工作量！再次强调（抱歉重复），这并非让您选择您会说的所有语言。
8. 现在您需要等待语言协调员或项目维护人员处理您的请求。请求获批后，您将收到邮件通知。若一周内未收到回复，请考虑联系 Transifex 中该语言的协调员，或致信‘**QGIS 翻译者邮件列表** <<https://lists.osgeo.org/mailman/listinfo/qgis-tr>>’。
 9. 您也可以加入其他 QGIS 项目，在各处提供帮助。

4.1.3 翻译

请求获批后，您即可翻译所选项目中的任意文本。只需单击您的语言，选择要翻译的章节（transifex 中称做资源），再单击“翻译”即可。很简单，对吧？

为帮助您完成高质量翻译，下方提供了一些说明。我们强烈建议您仔细阅读。

Tip

快速访问 Transifex 中的可翻译文件

如果您在当前文档中发现错误或缺失的翻译，可看向页面左下角，点击写着“QGIS Documentation+版本号”的下拉菜单中的“Translate page”（翻译此页）链接，直接跳转到 Transifex 中的源文件并进行修改。

4.2 翻译流程

QGIS 文档使用英文编写，采用 .rst 文件格式。为提供翻译版本：

1. 预构建脚本会在 /QGIS-Documentation/locale/en 文件夹中为英文生成名为 .po 的翻译文件。
2. .po 文件中的句子会被推送至 Transifex 网站平台，供译者使用编辑器将英文翻译成其目标语言。
3. 当某个文件翻译完成度达到 100% 时，其译文会自动拉取回文档仓库，目录存放在 /QGIS-Documentation/locale/<language>。

4. 在下一文档构建时（每天至少一次——页面底部可见时间），脚本会复用这些译文生成翻译后的输出。
5. 对于未完全翻译的文件，脚本每两周从 Transifex 拉取已翻译的字符串到 GitHub，并在下一次构建时一并发布。
6. 每当 .rst 文件更新时，英文版 .po 文件也会同步更新，并将变更推送至 Transifex 中对应的文件。这意味着，若一个已翻译的 .rst 文档新增了一段内容，只有这些新增或修改的句子会被加入到已翻译的 .po 文件中，需要译者处理。

Note

QGIS Desktop 翻译的特殊性

与 QGIS 文档翻译的主要区别在于：QGIS 应用程序不使用 .po 文件，而是将构成特定版本应用程序的 .py、.cpp、.yaml 文件中的所有可翻译字符串，以单个 .ts 文件（例如 qgis-application/qgis_en.ts (branch release-3_30)）的形式推送和拉取至 Transifex。译文会在开发分支中每日拉取到 GitHub，并在每个版本发布时同步。

目前 QGIS 使用两种不同的工具进行翻译：

- **Transifex web platform** 是翻译 QGIS **最简单且推荐的方式**。它透明地执行上述流程，将所有可翻译文本集中呈现给译者。您只需选择要翻译的文件即可。译文文件将保存在平台上，直至下一次版本推送。
- **Qt Linguist** 是 Qt 开发工具，要求译者从源代码中本地拉取 .po`（或 :file:.ts`）文件，完成翻译后再推送回去。

请注意，无论您选择哪种工具，翻译规则都是相同的。

4.3 翻译一个文件

为了解释翻译，我们可以用热力图插件做例子。在这个例子当中，我们将把英语翻译成荷兰语，其他语言也是同样的道理

该文档的源文件可在此处找到：

```
QGIS-Documentation/source/docs/user_manual/plugins/plugins_heatmap.rst
```

那么，我为何选择这份文档？

1. 它包括图片，说明文字，标题，参考和替换。
2. 我写它是为了方便我翻译;-)

构建流程已创建了英文版 .po 文件，可在此处找到：

```
QGIS-Documentation/locale/en/LC_MESSAGES/docs/user_manual/plugins/plugins_heatmap.  
↔po
```

对应的荷兰语 .po 文件（基本是副本）可在此处找到：

```
QGIS-Documentation/locale/nl/LC_MESSAGES/docs/user_manual/plugins/plugins_heatmap.  
↔po
```

在此文件旁边，您会看到一个小小的 .mo 文件，表明该文件尚无任何翻译内容。

4.3.1 在 Transifex 中进行翻译

要使用 Transifex 平台进行翻译，您需要：

1. 创建一个 *Transifex* 账户并加入 *QGIS* 项目。
2. 一旦你加入语言团队后，单击对应项目（此处为 *QGIS Documentation*）。系统将显示可用语言及其翻译完成率列表。

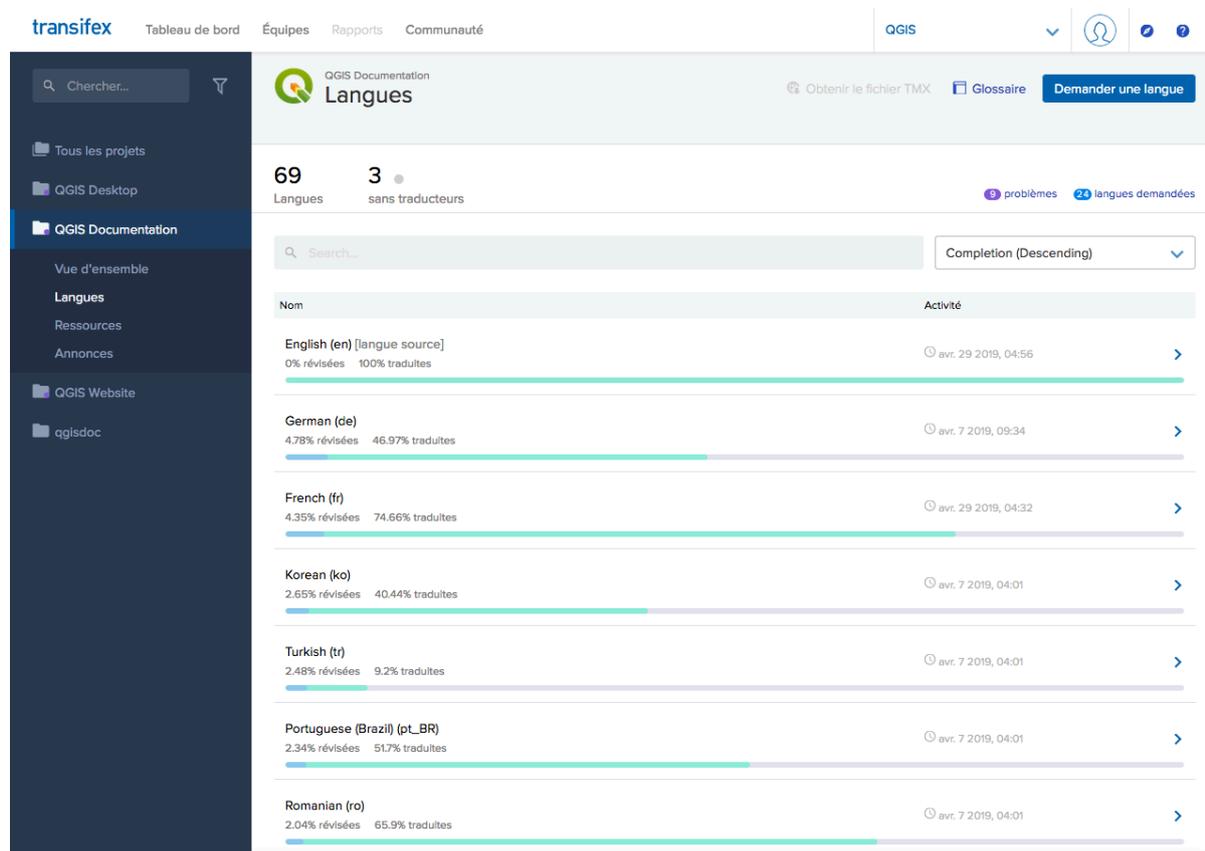


Fig. 4.1: 在 Transifex 菜单中选择要翻译的语言

3. 将鼠标悬停在您的语言上，然后单击以下任一选项：
 - *View resources*（查看资源）：显示可翻译的 .po 文件及其翻译比例、字符串数量等元数据。
 - *Translate*（翻译）：打开翻译界面，列出所有可用的 .po 文件
4. 找到您想翻译的文件（本例中为 docs_user-manual_plugins_plugins-heatmap，即热力图插件文件），或选择任意未完成的文件并单击：文件中的字符串将被加载，您可使用界面进行筛选、翻译或提出建议……

💡 Tip

单击任一页面左下角的下拉菜单中的 *Translate page* 链接。它会直接把你带到在 Transifex 中对应的翻译页面。

5. 您所需要做的就是选择每个文本并遵循指导。

有关使用 *transifex Web Editor* 的更多信息，请参见：<https://help.transifex.com/en/articles/6318216-translating-with-the-web-editor>.

4.3.2 在 Qt Linguist 中进行翻译

使用 Qt Linguist 时，您需要：

1. 手动获取 .po 或 .ts 文件。可通过 Transifex 平台下载，或从源代码仓库（GitHub）的 locale/\$language 文件夹中获取，
2. 在本地翻译
3. 将修改后的文件上传回其来源（Transifex 或 GitHub）。

虽然可通过 Transifex 下载和上传可翻译文件，但不建议采用此方式。因为 Transifex 没有版本控制系统，您上传的文件会直接覆盖现有文件，可能覆盖其他人在平台上所做的修改。

首次在 Qt Linguist 中打开文件时，您会看到以下对话框：



Fig. 4.2: 在 Linguist 菜单中选择翻译语言

目标语言应正确填写。源语言可保持默认（语言为 POSIX，国家/地区为任意国家）。

点击 *OK* 按钮后，Qt Linguist 将加载所有句子，您即可开始翻译，参见 Fig. 4.3。

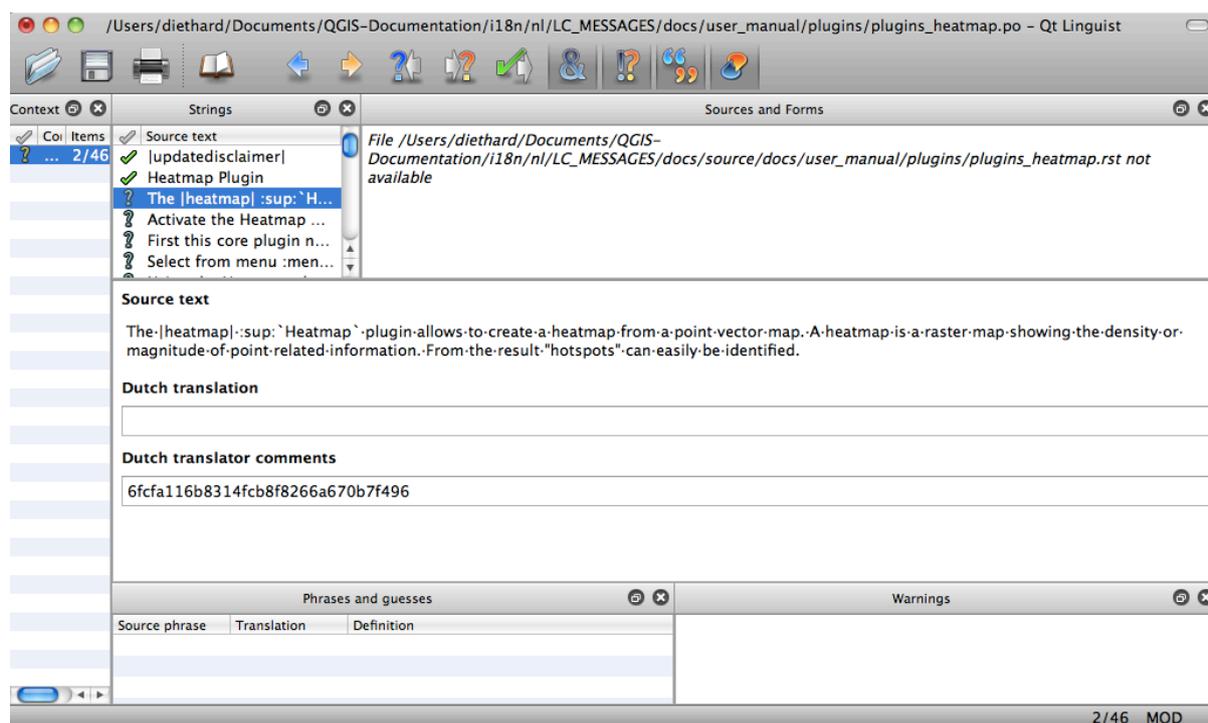


Fig. 4.3: 使用 Linguist 菜单进行翻译

菜单中提供以下便于使用的按钮：

-  按钮是最重要的按钮。若条目需要翻译，请在文本框中输入译文后点击此按钮；若无需翻译，请保持文本框为空并同样点击此按钮，表示该条目已完成，继续处理下一条目。
-  按钮用于跳转到上一个翻译条目。
-  按钮用于跳转到下一个翻译条目。
-  按钮会跳转到第一个仍需翻译的条目。当原文档发生变更，仅需翻译少量新增或修改的句子时，此功能非常实用。
-  按钮会反向查找并跳转到第一个仍需翻译的条目。

有关 Qt Linguist 的更多使用信息，请参见 <https://doc.qt.io/archives/qt-5.15/linguist-translators.html>

Warning

若要从源代码仓库下载待翻译内容，请勿使用 master 分支。对于翻译，始终存在专用的翻译分支——当某版本的英文文档完全更新后即会创建。例如，要翻译 QGIS 2.8 的手册，必须使用 manual_en_v2.8 分支。

4.3.3 翻译手册

现在我们开始翻译 `plugin_heatmap` 手册!

大多数句子的翻译都较为直接。在本次翻译过程中, 我将指出哪些部分 (rst 语句) 需要特殊处理。

下面我们看到一个有趣的待翻译句子:

The `|heatmap|` `:sup:`Heatmap`` plugin allows to create a heatmap from a point vector map. A heatmap is a raster map showing the density or magnitude of point related information. From the result "hotspots" can easily be identified.

该句子包含两个 rst 语句:

1. | 之间的 `|heatmap|` 是替换词, 这些词不应该被翻译! 这将被替换为热力图插件图标!
2. `:sup:`Heatmap`` 之中的 `:sup:` 语句是一个叠加语句, 并将下面的文本打印得更高一些。这用于显示当您鼠标悬停在工具栏项目上方时出现的弹出文本, 在 QGIS 应用程序中实际翻译时, 这可能会有所不同。在荷兰的情况下, 它不是!

该句子中所有其他纯文本均可翻译!

下一个翻译条目包含 `:ref:` 语句, 通常用于引用手册中其他章节! `:ref:` 后的文本是唯一标识符, **切勿修改!**

First this core plugin needs to be activated using the Plugin Manager (see Section `:ref:`load_core_plugin``). After activation the heatmap icon `|heatmap|` can be found in the Raster Toolbar.

在这种情况下, `load_core_plugin` 是放置在具有标题的第一项之前的唯一引用标识符。ref 语句将被替换为标题的文本, 并转换为超链接。当这个引用所引的标题被翻译时, 所有对这个标题的引用也会被自动翻译。

下一条目包含 rst 标签 `:menuselection:`, 其后文本为 QGIS 应用程序菜单中实际显示的内容。若该内容在应用程序中已被翻译, 则此处也应相应修改。

Select from menu `:menuselection:`View --> Toolbars --> Raster`` to activate the Raster Toolbar when it is not yet activated.

在上面的项目 “View ->” 实际上被翻译成 “Beeld ->”, 因为这是荷兰本地化的 QGIS 应用程序中使用的翻译。

再往下, 我们会遇到以下较棘手的翻译条目:

The `|heatmap|` `:sup:`Heatmap`` tool button starts the Dialog of the Heatmap plugin (see `:numref:`figure_heatmap_settings``).

它包含一个指向图示 `figure_heatmap_settings_` 的引用。与章节引用类似, 这个引用不应该被改变!! 该引用在 rst 文档中的定义未包含在 .po 文件中, 因此无法修改。这意味着图示引用不可翻译。生成 HTML 时, 您将看到 `figure_heatmap_settings`; 生成 PDF 时, `figure_heatmap_settings_` 会被替换为图号。

下一个包含 rst 属性的翻译条目如下:

****Input Point dialog****: Provides a selection of loaded point vector maps.

请勿删除上述行中的星号。它们用于将所包含文本以粗体显示。该文本通常是对话框中的内容, 在应用程序中可能已被翻译。

下面的翻译项包含 `:guilabel:` rst 标签。

When the `|checkbox|` `:guilabel:`Advanced`` checkbox is checked it will give access to additional advanced options.

:guilabel: 标签的文本 “Advanced” 在 QGIS 应用程序中很可能已被翻译，因此此处可能需要修改！
以下翻译条目包含 “airports”。引号用于将文本显示为另一种字体。此处为字面值，**无需翻译**。

```
For the following example, we will use the ``airports`` vector point
layer from the QGIS sample dataset (see :ref:`label_sampledata`).
Another excellent QGIS tutorial on making heatmaps can be found on
`https://www.qgistutorials.com
<https://www.qgistutorials.com/en/docs/creating_heatmaps.html>`_.
```

此条目还包含一个超链接，包含 URL 和外部显示文本。URL 当然应保持不变，但您可以修改读者可见的外部文本“<https://www.qgistutorials.com>”。**切勿删除超链接末尾的下划线**，它是链接的必要组成部分！

4.3.4 翻译规则总结

1. 请勿修改两个 | 之间的字符，例如 |bronze|、|checkbox|、|labels|、|selectString|、|addLayer|……这些是用于替换图像的特殊标签。
2. 请勿修改以 :ref:、:file:、:numref: 等角色开头的引用，除非它们包含标题。若包含标题，可翻译标题，但 **必须保留链接部分**（即 < 与 > 之间的文本）不变。

Tip

当引用包含标题时，Transifex 可能在英文源文本中用数字替代链接部分。请点击源文本中的该数字，将引用链接添加到正在翻译的标题旁边。

3. 请勿修改以下划线结尾的引用，例如 figure_labels_1_
4. 请勿修改超链接中的 URL，但可修改外部描述文本。请保留超链接末尾紧邻的下划线，且不要添加额外空格（如 >_）
5. 请修改 :index:、:sup:、:guilabel: 和 :menuselection: 标签引号内的文本。请核实在 QGIS 应用程序中是否/如何翻译，并相应调整。**切勿修改标签本身**。
6. 双星号或双引号之间的文本通常表示值或字段名，有时需要翻译，有时则不需要。
7. 请注意，必须使用与源文本 **完全相同**（数量一致）的特殊字符，如 `、``、*、**、::。这些字符影响信息的呈现效果。
8. 由特殊字符或标签包裹的文本，开头和结尾均不得包含空格。
9. 译文字符串末尾不要以新段落结束，否则在 HTML 生成时该文本将不会被正确处理。

遵循上述规则，译文文档将呈现良好效果！

如有任何疑问，请联系 [QGIS 社区团队](#) 或 [QGIS 翻译团队](#)。

5.1 用法

为了方便在 QGIS 手册中使用图标，在 [QGIS-Documentation repository](#) 的 `/source/substitutions.txt` 文件中定义了每个图标的替换，下面列出了其中的一些替换。因此，当您想在文档中使用 QGIS 应用程序中的图标时，很有可能已经有可以/应该使用的替代。

如果没有替代品：

1. 请检查文档仓库中 `/static/common` 文件夹内是否存在该图标。若未找到对应图像，则需从 [QGIS 代码仓库](#) 中查找并复制图标文件（通常位于 `默认主题目录` 下），然后以 `.png` 格式粘贴至文档仓库的 `/static/common` 文件夹中。为便于维护和后续更新，建议尽可能保留原始文件名。
2. 按照下面的示例在 `/substitutions.txt` 文件中创建对替换的引用。替换文本应该从文件名和 `camelCase` 中派生：

```
.. |dataSourceManager| image:: /static/common/mActionDataSourceManager.png
:width: 1.5em
.. |splitLayer| image:: /static/common/split_layer.png
:width: 1.5em
```

3. 使用新的替换更新文档的目标部分。
4. （可选但非常建议）将替换添加到下面的列表中。
5. 在使用它的文件末尾的替换列表中添加新的替换引用，或者运行方便的 `scripts/find_set_subst.py` 脚本。

```
# from the repository main folder
python3 scripts/find_set_subst.py
```

5.2 常见的替换

下面给出了在编写文档时使用的一些图标及其替换。可以在手册的许多地方使用/找到。

5.2.1 平台的图标

Icon	替换	Icon	替换
	logo		qt
	kde		nix
	osx		win

5.2.2 菜单项

Icon	替换	Icon	替换
	checkbox		unchecked
	radioButtonOn		radioButtonOff
	selectNumber		selectString
	selectColor		selectColorRamp
	hamburgerMenu		degrees
	inputText		slider

5.3 工具栏按钮图标

5.3.1 管理图层和概览

Icon	替换	Icon	替换
	dataSourceManager		
	addOgrLayer		addSensorThingsLayer
	addRasterLayer		addMssqlLayer
	addDelimitedTextLayer		addSpatialiteLayer
	addPostgisLayer		addOracleLayer
	addAfsLayer		addMeshLayer
	addVectorTileLayer		addXyzLayer
	addVirtualLayer		addWmsLayer
	addWcsLayer		addWfsLayer
	addPointCloudLayer		addGpsLayer

continues on next page

Table 5.3 – continued from previous page

Icon	替换	Icon	替换
	addTiledSceneLayer		addHanaLayer
	newVectorLayer		newSpatialiteLayer
	newGeoPackageLayer		createMemory
	newVirtualLayer		newMeshLayer
	newGpx		stac
	dbManager		gdal
	geoPackage		spatialite
	virtualLayer		wms
	wcs		wfs
	pointCloudLayer		gps
	tiledSceneLayer		hana
	dbSchema		
	inOverview		addAllToOverview
	removeAllFromOverview		removeLayer
	showAllLayers		hideAllLayers
	showPresets		showSelectedLayers
	hideSelectedLayers		hideDeselectedLayers
	toggleAllLayers		toggleSelectedLayers
	addLayer		
	indicatorTemporal		indicatorNonRemovable
	indicatorEmbedded		indicatorFilter
	indicatorMemory		indicatorNoCRS
	indicatorBadLayer		favourites
	indicatorLayerError		indicatorNotes
	indicatorLowAccuracy		indicatorOffline

5.3.2 工程

Icon	替换	Icon	替换
	fileNew		fileOpen
	fileSave		fileSaveAs
	fileExit		user

5.3.3 编辑

Icon	替换	Icon	替换
	undo		redo
	editCopy		editPaste
	editCut		saveEdits
	editableEdits		
	circle2Points		circle2TangentsPoint
	circle3Points		circle3Tangents
	circleCenterPoint		ellipseCenter2Points
	ellipseCenterPoint		ellipseExtent
	ellipseFoci		rectangle3PointsDistance
	rectangle3PointsProjected		rectangleCenter
	rectangleExtent		regularPolygon2Points
	regularPolygonCenterCorner		regularPolygonCenterPoint

5.3.4 Identify result

Icon	替换	Icon	替换
	expandTree		collapseTree
	expandNewTree		formView
	deselectAll		editCopy
	filePrint		
	identifyByRectangle		identifyByPolygon
	identifyByFreehand		identifyByRadius

5.3.5 数字化和高级数字化

Icon	替换	Icon	替换
	cad		cadConstruction
	cadParallel		cadPerpendicular
	floater		
	toggleEditing		allEdits
	tracing		snapping
	snappingVertex		snappingSegment
	snappingArea		snappingCentroid

continues on next page

Table 5.7 – continued from previous page

Icon	替换	Icon	替换
	snappingMiddle		snappingEndpoint
	capturePoint		capturePolygon
	captureLine		captureCurveFromFeature
	deleteSelectedFeatures		
	circularStringCurvePoint		circularStringRadius
	vertexTool		vertexToolActiveLayer
	digitizeWithSegment		digitizeShape
	streamingDigitize		digitizeWithCurve
	moveFeature		moveFeatureCopy
	moveFeatureLine		moveFeatureCopyLine
	moveFeaturePoint		moveFeatureCopyPoint
	rotateFeature		rotatePointSymbols
	scaleFeature		
	offsetCurve		offsetPointSymbols
	simplify		reshape
	addRing		addPart
	fillRing		
	deleteRing		deletePart
	mergeFeatures		mergeFeatureAttributes
	splitFeatures		splitParts
	reverseLine		
	allowIntersections		avoidIntersectionsCurrentLayer
	avoidIntersectionsLayers		snappingSelf

5.3.6 网孔

Icon	替换	Icon	替换
	meshDigitizing		meshReindex
	meshSelectExpression		meshSelectPolygon
	meshTransformByExpression		meshEditForceByVectorLines
	vertexCoordinates		

5.3.7 地图导航和属性

Icon	替换	Icon	替换
	pan		panToSelected
	zoomIn		zoomOut
	zoomActual		zoomFullExtent
	zoomToLayer		zoomToSelected
	zoomLast		zoomNext
	zoomInXAxis		refresh
	identify		mapTips
	showBookmarks		newBookmark
	measure		measureArea
	measureBearing		measureAngle
	newMap		new3DMap
	tiltUp		tiltDown
	3dNavigation		play
	camera		shadow
	editCutDisabled		temporalNavigationOff
	temporal		temporalNavigationAnimated
	temporalNavigationFixedRange		newElevationProfile
	temporalNavigationMovie		
	elevation		

5.3.8 选择和表达式

Icon	替换	Icon	替换
	selectRectangle		selectPolygon
	selectFreehand		selectRadius
	selectAll		deselectAll
	invertSelection		expressionSelect
	deselectActiveLayer		
	selectDistance		selectLocation
	selectAllTree		select
	selectAdd		selectRemove
	formSelect		dataDefine
	expression		dataDefineOn

continues on next page

Table 5.10 – continued from previous page

Icon	替换	Icon	替换
	dataDefineExpressionOn		dataDefineError
	dataDefineExpressionError		
	addExpression		
	expressionFilter		filterMap

5.3.9 标签和图表

Icon	替换	Icon	替换
	labelingSingle		labelingNone
	labelingRuleBased		labelingObstacle
	piechart		diagramNone
abc	text		histogram
	stackedBar		stackedDiagram
	createAnnotationLayer		annotationLayer
	formAnnotation		htmlAnnotation
	actionText		textAlongLine
	addImage		textInsideRect
	labelbackground		labelbuffer
	labelformatting		labelplacement
	labelshadow		render
	labelcallout		
	labelAnchorCenter		labelAnchorCustom
	labelAnchorEnd		labelAnchorStart
	pinLabels		showHideLabels
	moveLabel		rotateLabel
	showPinnedLabels		showUnplacedLabel
	changeLabelProperties		autoPlacementSettings
	labelingRules		

5.3.10 装饰

Icon	替换	Icon	替换
	copyrightLabel		addGrid
	titleLabel		northArrow
	scaleBar		addMap
	addImage		

5.3.11 帮助

Icon	替换	Icon	替换
	helpContents		qgisHomePage
	success		
	helpSponsors		contextHelp

5.3.12 颜色

Icon	替换	Icon	替换
	colorBox		colorPicker
	colorSwatches		colorWheel

5.4 其他基本图标

Icon	替换	Icon	替换
	arrowLeft		arrowRight
	arrowDown		arrowUp
	symbologyAdd		symbologyRemove
	projectProperties		options
	interfaceCustomization		keyboardShortcuts
	copyrightLabel		northArrow
	scaleBar		tracking
	gpsTrackBarChart		
	gpsConnect		gpsDisconnect
	gpsDestinationLayer		addTrackPoint
	recenter		reset

continues on next page

Table 5.15 – continued from previous page

Icon	替换	Icon	替换
	folder		extents
	settings		start
	properties		deleteSelected
	browserExpand		browserCollapse
	codeEditor		add
	relations		layoutItem3DMap
	stopwatch		sensor
	clearItem		warning

5.5 属性工作表

Icon	替换	Icon	替换
	openTable		openTableSelected
	openTableVisible		openTableEdited
	selectedToTop		
	selectAll		invertSelection
	panToSelected		zoomToSelected
	copySelected		editPaste
	expressionSelect		deleteSelectedFeatures
	newAttribute		deleteAttribute
	editTable		
	newTableRow		calculateField
	refresh		formView
	conditionalFormatting		multiEdit
	dock		actionRun
	duplicateFeature		zoomTo
	panTo		highlightFeature
	handleStoreFilterExpressionChecked		
	handleStoreFilterExpressionUnchecked		

5.6 投影和配准工具

Icon	替换	Icon	替换
	geographic		crs
	customProjection		setProjection
	projectionDisabled		projectionEnabled
	transformation		gdalScript
	georefRun		pencil
	linkQGisToGeoref		linkGeorefToQGis
	fullHistogramStretch		

5.7 打印布局

Icon	替换	Icon	替换
	newLayout		layoutManager
	duplicateLayout		
	newReport		newPage
	atlasSettings		atlas
	filePrint		saveMapAsImage
	saveAsSVG		saveAsPDF
	addBasicShape		addBasicCircle
	addBasicTriangle		addBasicRectangle
	addNodesShape		editNodesShape
	addPolygon		addPolyline
	addArrow		northArrow
	add3DMap		addMap
	elevationProfile		copyProfileSettings
	addLegend		addHtml
	addManualTable		addTable
	addImage		addMarker
	label		scaleBar
	select		moveItemContent
	setToCanvasScale		setToCanvasExtent
	viewScaleInCanvas		viewExtentInCanvas

continues on next page

Table 5.18 – continued from previous page

Icon	替换	Icon	替换
	raiseItems		lowerItems
	moveItemsToTop		moveItemsToBottom
	alignLeft		alignRight
	alignHCenter		alignVCenter
	alignTop		alignBottom
	distributeLeft		distributeRight
	distributeTop		distributeBottom
	distributeHCenter		distributeVCenter
	distributeHSpace		distributeVSpace
	resizeShortest		resizeTallest
	resizeNarrowest		resizeWidest
	resizeSquare		groupItems
	lockItems		unlockAll
	locked		unlocked
	lockRepeating		lockedGray

5.8 图层属性

Icon	替换	Icon	替换
	symbology		labelingSingle
	sourceFields		general
	metadata		action
	display		rendering
	join		diagram
	labelmask		temporal
	legend		dependencies
	3d		system
	elevationscale		layerTree
	editMetadata		overlay
	digitizing		auxiliaryStorage
	history		stylePreset
	search		pyramids
	transparency		rasterHistogram

continues on next page

Table 5.19 – continued from previous page

Icon	替换	Icon	替换
	singleSymbol		nullSymbol
	graduatedSymbol		categorizedSymbol
	25dSymbol		ruleBasedSymbol
	invertedSymbol		heatmapSymbol
	pointDisplacementSymbol		pointClusterSymbol
	mergedFeatures		
	meshContours		meshContoursOff
	meshVectors		meshVectorsOff
	meshFrame		meshAveraging
	singleColor		paletted
	singleBandPseudocolor		multibandColor
	pointCloudExtent		
	sum		sort
	paintEffects		mapIdentification
	styleManager		iconView
	joinNotEditable		joinedLayerNotEditable
	joinHasNotUpsertOnEdit		filterTableFields
	symbologyEdit		
	sharingImport		sharingExport

5.9 插件

5.9.1 数据处理

Icon	替换	Icon	替换
	processingAlgorithm		processingModel
	processingHistory		processingResult
	menu		runSelected
	processSelected		editHelpContent
	saveAsPython		modelOutput
	qgsProjectFile		addToProject
	fieldInteger		
	meanCoordinates		extractLayerExtent
	selectRandom		vectorGrid
	convexHull		buffer

continues on next page

Table 5.20 – continued from previous page

Icon	替换	Icon	替换
	intersect		union
	symmetricalDifference		clip
	difference		dissolve
	checkGeometry		exportGeometry
	delaunay		centroids
	polygonToLine		extractVertices
	lineToPolygon		nearestNeighbour
	splitLayer		heatmap
	showRasterCalculator		showMeshCalculator
	regularPoints		addGeometryAttributes
	basicStatistics		uniqueValues
	collect		simplify_2
	createGrid		distanceMatrix
	lineIntersections		mergeLayers
	sumPoints		sumLengthLines
	randomPointsInPolygons		randomPointsWithinPolygon
	randomPointsOnLines		randomPointsWithinExtent
	multiToSingle		
	grid		tiles
	merge		rasterClip
	contour		proximity
	polygonize		rasterize
	sieve		nearblack
	projectionAdd		projectionExport
	8To24Bits		24To8Bits
	rasterInfo		rasterOverview
	vrt		voronoi
	translate		warp
	iterate		terminal
	queryHistory		storedqueries

5.9.2 各种核心插件

基本安装时提供的标准配置，但在初始安装时不加载

Icon	替换	Icon	替换
	showPluginManager		installPluginFromZip
	pythonFile		runConsole
	showEditorConsole		clearConsole
	scriptOpen		searchEditorConsole
	searchRegex		replace
	commentEditorConsole		formatCode
	classBrowserConsole		codepadConsole
	syntaxErrorConsole		
	offlineEditingCopy		offlineEditingSync
	plugin		metasearch
	geometryChecker		topologyChecker
	fromSelectedFeature		sqlQueryBuilder

5.9.3 Grass 整合

Icon	替换	Icon	替换
	grassLogo		grassRegion
	grassTools		grassNewMapset
	grassOpenMapset		grassCloseMapset