QGIS Documentation Guidelines

QGIS Project

2020 년 12 월 09 일

Contents

QGIS 문서는 태평양 표준시 0 시, 오전 8 시, 오후 4 시에 서버 상에서 자동으로 빌드될 것입니다. https://docs.qgis.org 에서 현재 상태를 알 수 있습니다.

https://github.com/qgis/QGIS-Documentation 에서 QGIS 문서 소스 파일을 찾을 수 있습니다. 대부분 reStructuredText(reST) 포맷 문법으로 작성되었으며, 일부 스크립트는 HTML 산출물을 후처리하는 Sphinx 도구 모음으로 작성되기도 했습니다. 이 도구에 대한 일반적인 정보를 알고 싶다면, http://docutils.sourceforge.net/docs/ref/rst/restructuredtext.html 또는 https://www.sphinx-doc.org/en/master/usage/restructuredtext/basics.html 을 참조하세요.

이 문서의 나머지 부분에서 다음과 같은 내용을 배울 수 있을 것입니다:

- git 시스템 및 문서 소스 파일이 저장된 GitHub 플랫폼을 사용해서 문서 소스 파일을 어떻게 관리하는지
- 규칙에 따라 텍스트를 어떻게 수정하고, 스크린샷을 어떻게 업로드할 수 있는지
- 사용자의 수정 사항을 어떻게 공유하고, 공식 문서로 푸시하도록 보장할 수 있는지

QGIS 프로젝트에 참여하는 방법에 대한 전체적인 정보를 알고 싶다면, QGIS 커뮤니티에 참여하기 웹페이지를 방문해주십시오.

CHAPTER 1

공헌단계

참고: 이번 과정을 설명하는 데 QGIS 문서를 예로 들긴 했지만, 다음 단락부터 보여줄 모든 명령어 및 단계는 QGIS 웹사이트에도 적용할 수 있습니다.

이 라인을 읽고 있다는것은 QGIS 문서와 방법을 고려하고 있다는것입니다. 정확하게 오셨습니다! 현재문서는 다양한 방식으로 목적을 달성하기위해 도와드리며 고려할 사항을 알려드립니다.

도움이 필요하시면 문제리포트에 코맨트를 남기거나 〈QGIS-커뮤니티팀 목록 <https://lists.osgeo.org/mailman/listinfo/qgis-community-team>〉 _ 에 쓰십시오. 자세한 내용은 <https://qgis.org/en/site/getinvolved/document.html> 를 참조하십시오.

과정으로 이동하겠습니다.

문서의 원본은 https://github.com/qgis/QGIS-Documentation 에 GitHub 에 git 버전으로 저장되어 있습니다. 파일을 수정하는 방식은 다음 두가지입니다:

1. GitHub 웹인터페이스 1 사용
2. Git 커맨드라인툴 1 사용.

1.1 GitHub 웹인터페이스 사용

GitHub 웹인터페이스에서 실행할 수 있는 작업:

- 파일편집
- 변경사항 미리보기와 코맨트
- 메인 저장소에 삽입할 변경사항을 요청
- 브랜치 생성, 업데이트 삭제

git 와 GitHub 어휘에 익숙하지 않다면 GitHub 〈Hello-world <https://guides.github.com/activities/hello-world/>〉 _ 프로젝트에서 아래에서 사용할 기본적인 어휘와 액션을 학습하십시오.

참고: 보고된 이슈을 수정하려면

〈문제점 <https://github.com/qgis/QGIS- Documentation/issue>〉 _ 를 수정하려면 코맨트를 자신의 이슈리포트에 적용하십시오. 동일한 문제를 여러사람이 해결하려는것을 방지할 수 있습니다.

1.1.1 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 Fork 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 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.

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.
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. When you finish, make a short comment about your changes and click on *Propose changes*. This will generate a new branch (`patch-xxx`) in your repository.
4. After you click on *Propose changes*, github will navigate to the *Comparing changes* page.
   - If you're done making changes, skip to *Compare changes* in 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 Branch: master ▾ and search for the `patch-xxx` branch. Select this patch branch. The Branch: master ▾ button will now say *Branch: patch-xxx*
     3. Jump down to *Modify files* below.

참고: The `Edit on GitHub` shortcut is also available in the drop-down menu at the bottom of the left sidebar.

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

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

Click on Branch: master ▾ 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 . The name of the new branch should relate to the problem you intend to fix. The Branch: master ▾ button should now say *Branch: branch_name*

팁: 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 3. Modify files

1. Browse the source files of your fork of QGIS-Documentation to the file that needs to be modified
2. Make your modifications following the *writing guidelines*
3. When you finish, navigate to the **Commit Changes** frame at the bottom of the page, make a short comment about your changes, and click on *Commit Changes* to commit the changes directly to your branch. Make sure *Commit directly to the branch_name branch.* is selected.
4. Repeat the previous steps for any other file that needs to be updated to fix the issue

1.1.4 4. Share your changes via Pull Request

You need to make a pull request to integrate your changes into the official documentation.

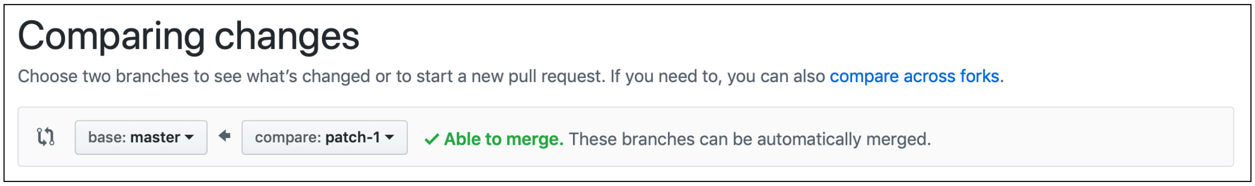
> **참고: If you used an `Edit on GitHub` link**
>
> After you commit your changes GitHub will automatically open a new page comparing the changes you made in your `patch-xxx` branch to the `qgis/QGIS-Documentation` master branch.
>
> Skip to *Step 2* below.

1. Start a new pull request

Navigate to the main page of the QGIS-Documentation repository and click on *New pull request*.

2. 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.



Comparing changes

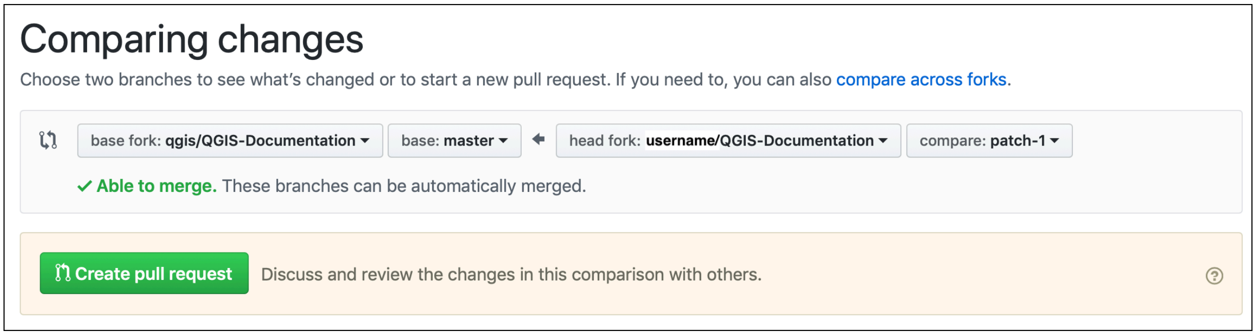
Choose two branches to see what's changed or to start a new pull request. If you need to, you can also compare across forks.

✓ Able to merge. These branches can be automatically merged.

그림 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



Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also compare across forks.

base fork: qgis/QGIS-Documentation ▾ | base: master ▾ | ← | head fork: username/QGIS-Documentation ▾ | compare: patch-1 ▾

✓ Able to merge. These branches can be automatically merged.

Create pull request — Discuss and review the changes in this comparison with others.

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

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.

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.

경고: 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.

팁: 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.

3. Describe your pull request

A text box will open: fill in any relevant comments for the issue you are addressing.

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*.

4. Review and comment pull request

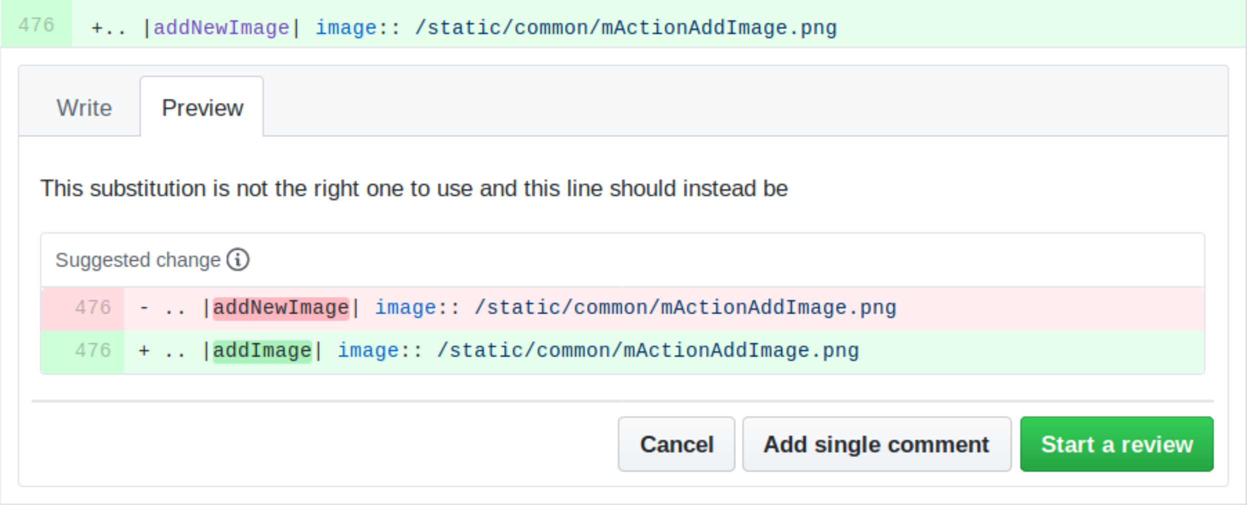
As seen above, anyone can submit modifications to the documentation through pull requests. Likewise anyone can review pull requests with questions and comments. Perhaps the writing style doesn't match the project guidelines, the change is missing some major details or screenshots, or maybe everything 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.



```
476  +.. |addNewImage| image:: /static/common/mActionAddImage.png
```

This substitution is not the right one to use and this line should instead be

Suggested change

```
476  - .. |addNewImage| image:: /static/common/mActionAddImage.png
476  + .. |addImage| image:: /static/common/mActionAddImage.png
```

그림 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.

팁: 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.

5. 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 a Travis CI build which automatically checks your contribution for build errors. If Travis CI finds an error, 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.

To make changes click on the Files changed tab in your pull request page and click the pencil button ✎ next to the filename that you want to modify.

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 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 Git 명령 프롬프트 도구 이용

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
- 작성자의 변경 사항을 테스트하기 위한 문서를 빌드

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.

다음에 나오는 코드 예시에서, # 로 시작하는 줄은 주석을 의미하지만 $ 로 시작하는 줄은 작성자가 입력해야 하는 명령어를 나타냅니다.

1.2.1 로컬 저장소

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
```

이 명령줄은 예시일 뿐입니다. 작성자는 `<YourName>` 을 작성자의 사용자명으로 대체하여 경로 및 저장소 URL 둘 다 작성자의 로컬 환경에 맞춰야 합니다.

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* 은 작성자의 QGIS 문서 저장소의 원격 저장소의 명칭입니다.
- *master* 는 기본 주 분기입니다. 기고하는 데 절대로 이 분기를 이용해서는 안 됩니다! **절대로요!!**

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
```

팁: 권한 거부 (퍼블릭키) 오류?

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 또다른 원격 저장소 추가

주 프로젝트에서 작업을 추적할 수 있으려면, 작성자의 로컬 저장소에 새 원격 저장소를 추가하십시오. 이 새 원격 저장소가 QGIS 프로젝트에서 나온 QGIS 문서 저장소입니다.

```
$ 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)
```

이제 두 원격 저장소들 사이에서 선택할 수 있습니다.

- **작성자** 의 원격 저장소에 작성자의 로컬 분기를 푸시하는 *origin*
- 공식 문서로 작성자의 작업 내용을 (그럴 권한이 있을 경우) 통합하거나, 공식 저장소의 마스터 분기에서 나온 작성자의 로컬 저장소의 마스터 분기를 업데이트하는 *upstream*

참고: *upstream* 은 라벨일 뿐으로, 일종의 표준 명칭이지만 원하는 대로 명명할 수 있습니다.

1.2.3 작성자의 기반 분기 업데이트

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
```

(다음 페이지에 계속)

(이전 페이지에서 계속)

```
# (which should be master, see step 1)
$ git merge upstream/master
# 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.

참고: 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 작성자의 작업 분기로 기고

이제 작성자의 기반 분기가 업데이트됐으니, 작성 내용을 추가할 전용 분기를 생성해야 합니다. 언제나 기반 분기가 아닌 다른 분기에서 작업하십시오! 언제나요!

```
# 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
```

커밋/푸시 (commit/push) 명령어에 대해:

- 오직 하나의 (원자 단위의 변경) 내용만, 예를 들어 오직 문제점 하나만 다루도록 해보십시오.
- 작성자의 커밋 제목 및 설명에 무엇을 변경했는지 꼼꼼히 서술해보십시오. 첫 줄은 제목으로, 대문자로 시작해야 하고 문자 80 개의 길이 제한이 있으며 `.` 로 끝나서는 안 됩니다. 제목은 간결하게 적으십시오. 설명은 더 길어도 되고, `.` 로 끝나며, 더 상세하게 서술할 수 있습니다.
- 어떤 문제점인지 식별하기 위해 `#` 뒤에 문제점 번호를 적으십시오. 해당 버그 티켓을 해결했다면 그 앞에 `Fix` 를 입력하십시오. 작성자의 커밋이 해당 티켓을 폐지할 것입니다.

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 작성자의 변경 사항 공유

이제 작성자의 GitHub 저장소로 가서 앞 단락에서 설명한 대로풀 요청을 생성 할 수 있습니다. 작성자의 분기에서 목표인 공식 QGIS 문서 저장소에 있는 원격 분기로 풀 요청을 생성했는지 확인하십시오.

1.2.6 작성자의 로컬 및 원격 저장소 청소

작성자의 풀 요청이 공식 QGIS 문서에 통합된 다음, 작성자의 분기를 삭제할 수 있습니다. 이런 방식으로 많은 작업을 하는 경우, 몇 주만 지나도 쓸데없는 분기들을 많이 보유하게 될 겁니다. 따라서 다음 방법을 통해 작성자의 저장소를 말끔히 유지하십시오.

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

그리고 작성자의 로컬 저장소에 있는 master 분기를 업데이트하는 것도 잊지 마십시오!

1.3 참고문헌

- 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 https://help.github.com/articles/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 https://help.github.com/articles/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 https://help.github.com/articles/using-git-rebase-on-the-command-line/

CHAPTER 2

Writing Guidelines

In general, when creating reST documentation for the QGIS project, please follow the Python documentation style guidelines. For convenience, we provide a set of general rules we rely on for writing QGIS documentation below.

2.1 문서 작성

2.1.1 표제

문서의 .rst 파일 하나가 웹페이지 하나와 대응합니다.

밑줄 (1 단계 제목의 경우 윗줄도 포함) 그어진 제목을 통해 텍스트의 구조를 구성하는 절 (section) 을 식별합니다. 동일한 단계의 제목은 반드시 동일한 특수문자로 밑줄을 작성해야 합니다. QGIS 문서의 경우, 장 (chapter), 절 (section), 항 (subsection), 목 (minisec) 에 대해 다음과 같은 스타일을 사용해야 합니다.

```
********
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 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`
```

(다음*image* 를 참조하십시오)

- **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`

- **사용자 텍스트**

```
``label``
```

2.1.4 Labels/references

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

다음은 절 (예: 라벨/참조 제목) 의 앵커를 생성하는 예시입니다.

```
.. _my_anchor:

Label/reference
---------------
```

동일 페이지 안에서 참조를 호출하려면 다음과 같이 작성합니다.

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

다음과 같은 내용을 반환할 것입니다.

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.

문서 어디에서든 동일한 장소로 넘어가는 또다른 방법은 `:ref:` 기능을 이용하는 것입니다.

```
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.

2.1.5 Figures and Images

그림

다음은 이미지를 삽입하는 예시입니다.

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

다음과 같은 그림을 반환합니다.

별명

You can put an image inside text or add an alias to use everywhere. To use an image inside a paragraph, first create an alias in the `source/substitutions.txt` file:

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

and then call it in your paragraph:

```
My paragraph begins here with a nice logo |nice_logo|.
```

This is how the example will be displayed:

My paragraph begins here with a nice logo .

To allow preview rendering in GitHub that is as close as possible to HTML rendering, you will also need to add the image replacement call at the end of the file you changed. This can be done by copy-pasting it from `substitutions.txt` or by executing the `scripts/find_set_subst.py` script.

참고: Currently, to ensure consistency and help in the use of QGIS icons, a list of aliases is built and available in the 별명 chapter.

도형

```
.. _figure_logo:

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

   A caption: A logo I like
```

결과는 다음과 같습니다.

그림 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 그림 2.1

This is the preferred way of referencing figures.

참고: 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*

It is also possible (but not recommended) to use the following mechanism:

```
(see Figure_logo_).
```

It will render like this:

(see *Figure_logo*).

You can use uppercase if you want. This mechanism can only be used in the same `.rst` file.

도표

A simple table can be coded like this

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

It will render like this:

x	y	z
1	2	3
4		5

Use a \ (backslash) followed by an empty space to leave an empty space.

You can also make more complicated tables and reference them:

```
.. _my_drawn_table:

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

My drawn table, mind you this is unfortunately not regarded as a caption

You can reference it like this: my_drawn_table_.
```

결과물은 다음과 같습니다.

<table>
<tr><td>Windows</td><td>macOS</td></tr>
<tr><td>⊞</td><td>X</td></tr>
<tr><td colspan="2">and of course not to forget 🐧</td></tr>
</table>

My drawn table, mind you this is unfortunately not regarded as a caption

You can reference to it like this *my_drawn_table*.

For even more complex tables, it is easier to use `list-table`:

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

   * - What
     - Purpose
     - Key word
     - Description
   * - **Test**
     - ``Useful test``
     - complexity
     - Geometry.  One of:

       * Point
       * Line
```

결과물은 다음과 같습니다.

What	목적	Key word	설명
Test	`Useful test`	complexity	Geometry. One of: • 포인트 • Line

2.1.6 색인

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 RST. You can use the inline `:index:` tag within normal text:

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

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.

2.1.7 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.
```

2.1.8 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.

참고: While texts in note, tip and warning frames are translatable, be aware that code block frames do not allow translation. So avoid comments not related to the code and keep comments as short as possible.

2.1.9 주석

Please note: Footnotes are not recognized by any translation software and it is also not converted to pdf format properly. So, if possible, don't use footnotes within any documentation.

This is for creating a footnote (showing as example[1])

```
blabla [1]_
```

다음을 가리키게 됩니다.

2.2 Managing Screenshots

2.2.1 새 스크린샷 추가

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 placed in the same folder as the QGIS-Documentation 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)

[1] Updates of core plugins

- 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
- Set print size resolution to `135 dpi` (e.g. in Gimp set the print resolution *Image ▯ Print size* and save). 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

팁: 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 english 〈original〉 screenshot.

2.3 공간 처리 알고리즘 문서 작성

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 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`.

  참고: 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.:

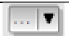
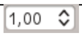
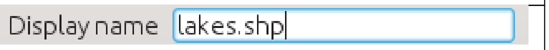
```
* - **Number of points**
  - ``NUMBER_OF_POINTS``
  - [number]

    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	설명	Visual indicator
Point vector layer	`vector: point`	⁘
Line vector layer	`vector: line`	∨
폴리곤 벡터 레이어	`vector: polygon`	▱
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`	
Table layer	`table`	▤
표현식	`expression`	ε
Point geometry	`coordinates`	
범위 (Extent)	`extent`	
좌표계	`crs`	🌐

다음 페이지에 계속

표 2.1 – 이전 페이지에서 계속

Parameter/Output type	설명	Visual indicator
Enumeration	`enumeration`	
List	`list`	
숫자	`number`	 1,00
문자열	`string`	 Display name lakes.shp
불 (Boolean)	`boolean`	
Folder path	`folder`	
파일	`file`	
매트릭스	`matrix`	
레이어	`layer`	
Same output type as input type	`same as input`	
Definition	`definition`	
포인트	`point`	
MultipleLayers	`multipleLayers`	
범위	`range`	
AuthConfig	`authconfig`	
메시 (mesh)	`mesh`	
Layout	`layout`	
LayoutItem	`layoutitem`	
색상	`color`	
축척	`scale`	

- Study an existing and well documented algorithm, and copy all the useful layouts.
- When you are finished, just follow the guidelines described in 공헌단계 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
..........
```

(다음 페이지에 계속)

(이전 페이지에서 계속)

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

   * - Label
     - Name
     - 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
```

(다음 페이지에 계속)

(이전 페이지에서 계속)

```
   :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
```

CHAPTER 3

Writing code in the PyQGIS Cookbook

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(4326, QgsCoordinateReferenceSystem.PostgisCrsId)
   assert crs.isValid()
```

You now use this:

```
.. testcode::

   crs = QgsCoordinateReferenceSystem(4326, QgsCoordinateReferenceSystem.PostgisCrsId)
   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(4326, QgsCoordinateReferenceSystem.PostgisCrsId)
   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
```

참고: 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 rst 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(4326, QgsCoordinateReferenceSystem.PostgisCrsId)
   assert crs.isValid()
```

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

참고: 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 usefull to use in the test setup:

```
.. testsetup:: *

   from qgis.core import QgsCoordinateReferenceSystem
```

3.2 How to test snippets on your local machine

참고: 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.

CHAPTER 4

번역 지침

This manual is aiming to help the translator. First the general process of how technically a translation is done is explained. Later the translation is explained from an actual English rst document that is translated to Dutch. Finally a summary of *Rules of translation* is given.

참고: 이 지침들이 QGIS 문서에 중점을 두고 있긴 하지만, 다음 절부터 설명하는 방법 및 규칙은 QGIS 응용프로그램이나 웹사이트 번역에도 적용할 수 있습니다.

4.1 번역 과정

QGIS Documentation is written in English with `.rst` files. In order to provide translations:

1. A prebuild script creates translation files named `.po` files for the English language in the folder `/QGIS-Documentation/locale/en`.
2. These 《originals》 are then copied by the script to the `locale` folders for other languages.
3. The sentences in the `.po` files are pushed to the Transifex web platform, and made available for translators who can begin to translate from English to their language with the editor.

4. At the end of the day, a script pulls back all validated translations
5. At the next build of the documentation (which occurs at least once a day), a script reuses the sentences to create translated output
6. When afterwards an .rst document is updated a new .po file is created in the English part. The contents of this new file will be merged with already existing .po files for each language. This means that when a new line is added to an .rst document that was already translated, only the new/updated sentences are added in the translated .po file and needs to be translated. The amount of work for updating translations for next release should be relatively small.

참고: The process above is the same followed to translate QGIS website, QGIS Desktop and QGIS Server. The difference with the applications is that instead of .po files, all the translatable strings in the .py, .cpp, .yaml and others…files that shape the application are pushed to and pulled from transifex as a single .ts file.

현재 QGIS 를 번역하는 데 쓰이는 도구는 두 가지입니다.

- The Transifex web platform, the easiest and recommended way to translate QGIS, transparently does the process described above and pulls all the translatable texts in one place for the translator. Just pick the files you want and translate. Translated files are stored in the platform until another release is pushed.
- Qt Linguist, a Qt development tool, requires the translator to pull locally the .po (or .ts) files from the source code, translate and then push back.

어떤 도구를 선택하든, 번역 규칙은 동일하다는 사실을 명심하십시오.

4.2 파일 번역

어떻게 번역 작업을 해야 하는지 설명하기 위해, 열지도 플러그인을 예시로 이용하겠습니다. 이 예시를 영어에서 네덜란드 어로 번약할 텐데, 모든 언어의 다른 문서를 번역할 때도 실질적으로 동일합니다.

문서의 소스를 다음 위치에서 찾을 수 있습니다.

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

어째서 이 문서를 예시로 선택했을까요?

1. It includes images, captions, headers, references and replacements.
2. 내가 직접 작성했으니 번역하기도 쉽기 때문이지요;-)

The build process has created the English .po file which can be found here:

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

The equivalent Dutch .po file (basically a copy) can be found here:

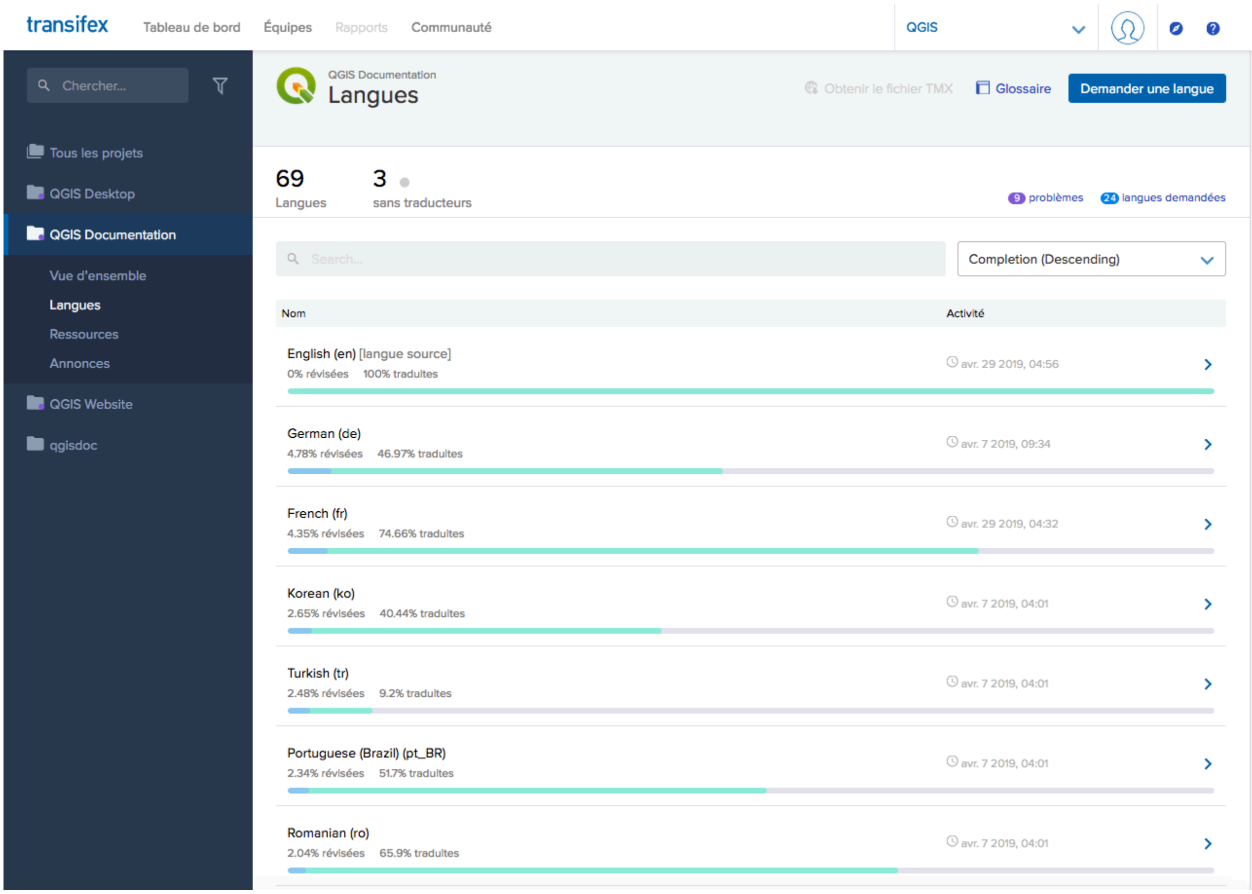
```
QGIS-Documentation/locale/nl/LC_MESSAGES/docs/user_manual/plugins/plugins_heatmap.po
```

Along this file you will see a tiny .mo file which indicates that it does not hold any translations yet.

4.2.1 Transifex 로 번역

In order to translate using Transifex, you need to:

1. create an account on Transifex and join the QGIS project.
2. Once you are part of a language team, click on the corresponding project (in this case `QGIS Documentation`). A list of available languages with their ratio of translation is displayed.



Nom	Activité
English (en) [langue source] 0% révisées 100% traduites	avr. 29 2019, 04:56
German (de) 4.78% révisées 46.97% traduites	avr. 7 2019, 09:34
French (fr) 4.35% révisées 74.66% traduites	avr. 29 2019, 04:32
Korean (ko) 2.65% révisées 40.44% traduites	avr. 7 2019, 04:01
Turkish (tr) 2.48% révisées 9.2% traduites	avr. 7 2019, 04:01
Portuguese (Brazil) (pt_BR) 2.34% révisées 51.7% traduites	avr. 7 2019, 04:01
Romanian (ro) 2.04% révisées 65.9% traduites	avr. 7 2019, 04:01

그림 4.1: Select language for translation in the Transifex menu

3. Hover over your language and click either:
   - *View resources*: translatable `.po` files with their ratio of translation, number of strings and some more metadata are now displayed.
   - or *Translate*: opens the interface of translation with all the available `.po` files
4. Identify the file you'd like to translate (in our case we are looking for the `docs_user-manual_plugins_plugins-heatmap`, the heatmap plugin file) or any unfinished file and click on it: strings in the files are loaded and you can use the interface to filter, translate, suggest translation⋯

> **팁:** For the documentation or the website, clicking the `Fix me` link in the footer of a page brings you directly to its corresponding translation page in Transifex.

5. All you need to do is select each text and translate following the *guidelines*.

For further information on the use of Transifex Web Editor, see https://docs.transifex.com/translation/translating-with-the-web-editor.

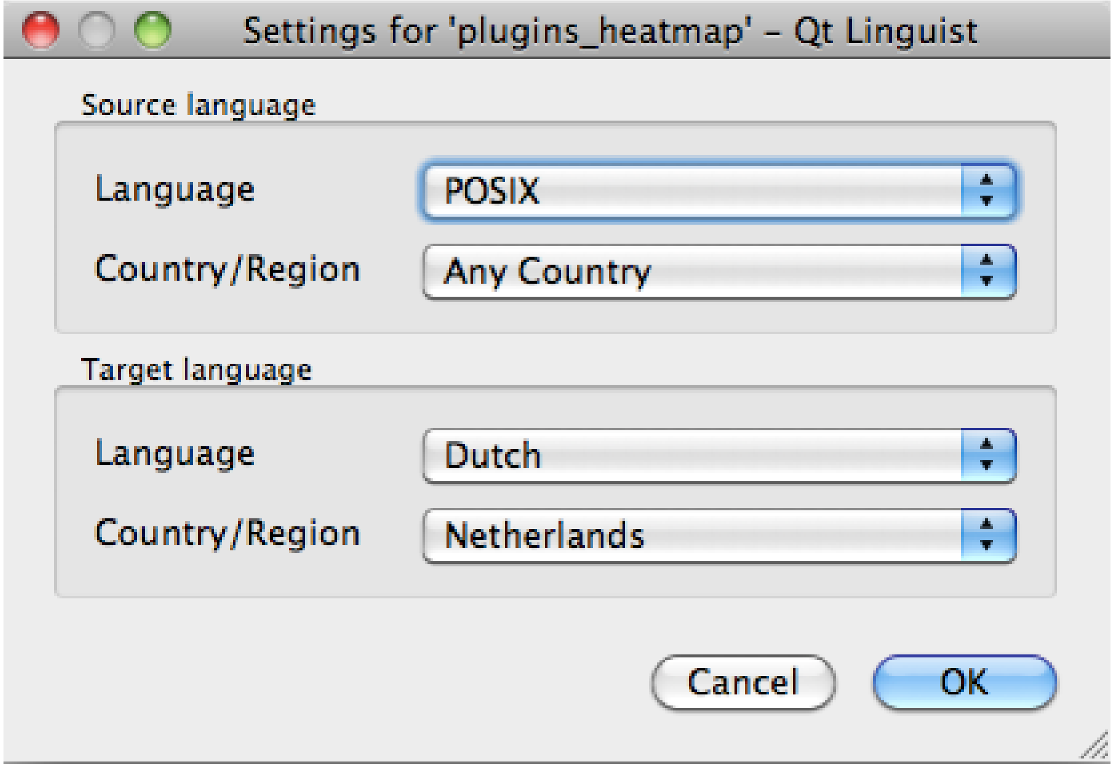
4.2.2 Qt Linguist 로 번역

With Qt Linguist, you need to:

1. manually grab the `.po` or `.ts` file(s). This can be achieved by downloading the file(s) either from Transifex platform or from the `locale/$language` folder of the source repository (in GitHub),
2. proceed to the translation locally
3. upload the modified files to their sources (Transifex or GitHub).

While downloading and uploading translatable files can be done with Transifex, it's not advised to use this process. Since there's no versioning system on Transifex, the file you upload will simply replace the existing one and potentially overwrite any modification made by others on the platform in the meantime.

When you open the file in Qt Linguist for the first time you will see the following dialog:



Settings for 'plugins_heatmap' – Qt Linguist

Source language

Language: POSIX

Country/Region: Any Country

Target language

Language: Dutch

Country/Region: Netherlands

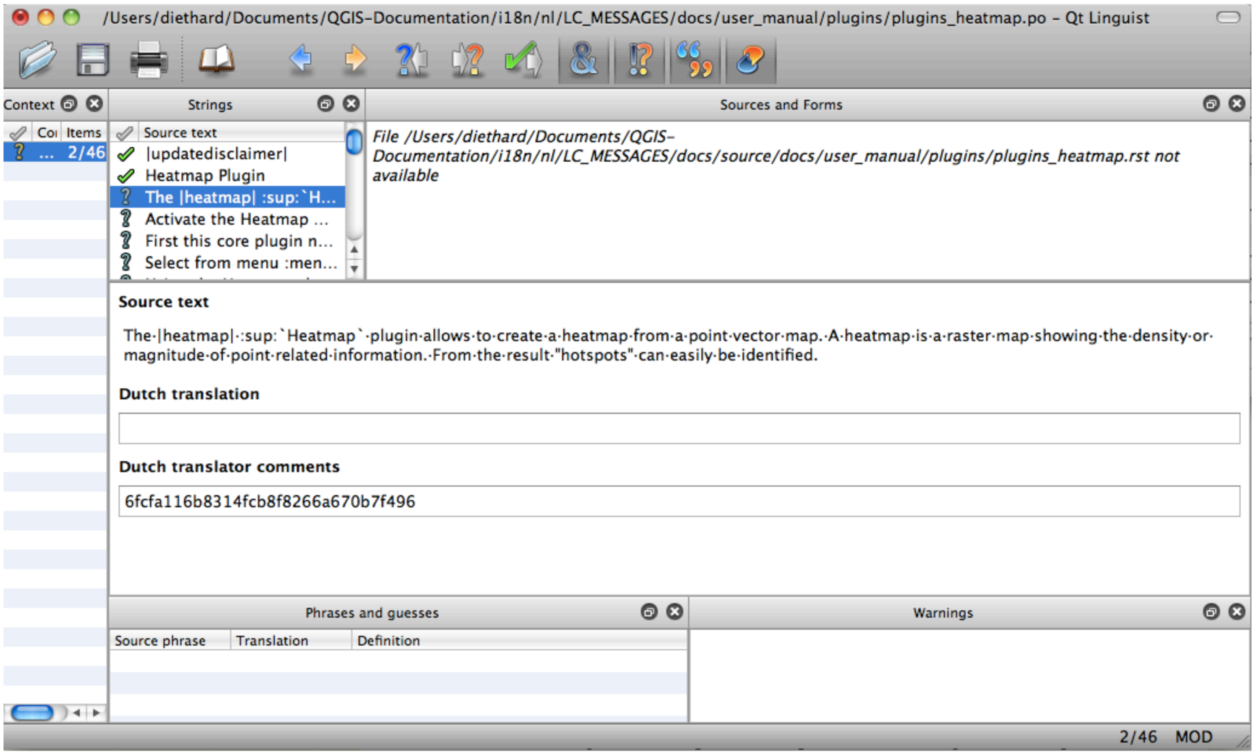
그림 4.2: Select language for translation in linguist menu

목표 언어를 정확하게 선택해야 합니다. 소스 언어는 언어 POSIX, 국가/지역 Any Country 를 그대로 놔두어도 상관없습니다.

When you press the *OK* button Qt Linguist is filled with sentences and you can start translating, see *Figure_translation_menu*.

메뉴에 사용하기에 편리한 다음 버튼들이 있습니다.

- ✓ 번역 완료 다음 (Translation Done Next) 버튼은 가장 중요한 버튼입니다. 어떤 항목을 번역해야 할 경우, 텍스트 영역에 번역을 입력한 다음 이 버튼을 클릭하십시오. 번역해야 할 항목이 아니라면 번역 텍스트


/Users/diethard/Documents/QGIS-Documentation/i18n/nl/LC_MESSAGES/docs/user_manual/plugins/plugins_heatmap.po - Qt Linguist

File /Users/diethard/Documents/QGIS-Documentation/i18n/nl/LC_MESSAGES/docs/source/docs/user_manual/plugins/plugins_heatmap.rst not available

Source text

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.

Dutch translation

Dutch translator comments

6fcfa116b8314fcb8f8266a670b7f496

그림 4.3: Qt Linguist 메뉴를 사용해 번역하기

영역을 비워놓은 채 역시 이 버튼을 클릭하십시오. 항목 번역을 완료했고 다음 항목을 이어서 번역하겠다는 의미입니다.

- 이전 항목 (Goto Previous) 버튼은 이전에 번역한 항목으로 가는 데 쓰입니다.
- 다음 항목 (Goto Next) 버튼은 다음에 번역할 항목으로 가는 데 쓰입니다.
- 다음 할 일 (Next Todo) 버튼은 번역해야 할 다음 첫 번째 항목으로 넘어갑니다. 원본 문서가 변경돼 새로운/업데이트된 문장들만 번역해야 할 경우 유용합니다.
- 이전 할 일 (Previous Todo) 버튼은 이전 방향으로 검색해서 번역해야 할 첫 번째 항목으로 넘어갑니다.

For further information on the use of Qt Linguist, see https://doc-snapshots.qt.io/qt5-5.12/linguist-translators.html

> **경고:** If you want to download content to translate from the source repository, never do this in the `master` branch. For translations there are always translation branches available, once a document is fully updated in English for a certain version. As an example, to translate the manual of QGIS 2.8, you have to use the manual_en_v2.8 branch.

4.2.3 사용자 설명서 번역

이제 열지도 플러그인 설명서를 번역해봅시다!

대부분의 문장은 번역하는 데 복잡하지 않을 겁니다. 이 번역 과정에서는 어떤 부분 (reST 표현) 에 특별한 번역이 필요한지 짚어보겠습니다.

다음에 번역할 문단이 흥미로워 보이는군요.

```
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.
```

이 문단은 reST 표현을 두 개 담고 있습니다.

1. `|heatmap|` : `|` 사이에 있는 단어들은 별명으로, 이 단어들을 번역해서는 안 됩니다! 이 표현은 열지도 플러그인 아이콘으로 교체될 겁니다!
2. `:sup:`Heatmap`` : `:sup:` 표현은 중첩 선언으로 그 다음에 오는 텍스트를 좀 더 높게 표시합니다. 이 표현은 마우스 커서를 툴바 항목 위에 올렸을 때 나타나는 팝업 텍스트를 보여주기 위해 쓰입니다. QGIS 응용 프로그램에서 실제로 번역된 경우 이 텍스트도 번역해야 할 수도 있습니다만, 이번 네덜란드 어 예시에선 번역하지 않습니다!

이 문단의 다른 모든 평문들은 번역할 수 있습니다!

다음 번역 항목은 `:ref:` 표현을 담고 있는데, 설명서 안 또다른 위치를 참조하는 데 흔히 쓰이곤 하죠! `:ref:` 표현 뒤에 나오는 텍스트는 유일 식별자이기 때문에 번역해선 안 됩니다!

```
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.
```

In this case `load_core_plugin` is a unique reference identifier placed before an rst item that has a caption. The ref statement will be replaced with the text of the header and turned into a hyperlink. When the header this reference is referring to is translated, all references to this header will be automatically translated as well.

다음 항목엔 QGIS 응용 프로그램의 메뉴에 실제로 표시되는 텍스트를 담고 있는 reST 태그 `:menuselection:` 이 있는데, 응용 프로그램이 번역된 경우 당연히 이 메뉴 텍스트도 번역해야겠죠.

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

이 항목에서 실제로 《View –>》 를 《Beeld –>》 로 번역했는데, 네덜란드 어로 현지화된 QGIS 응용 프로그램에 쓰인 번역이기 때문입니다.

조금 더 가면 다음과 같은 까다로운 번역 항목을 보게 됩니다.

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

It holds a reference to a figure `figure_heatmap_settings_`, and like a reference to a section this reference should not be changed!! The reference definition from the rst-document is not included in the `.po` file and can therefore not be changed. This means the reference to figures can not be translated. When HTML is created you will see `figure_heatmap_settings`. When a PDF document is created `figure_heatmap_settings_` is replaced with a figure number.

reST 속성을 가진, 다음으로 번역할 항목은 다음과 같습니다.

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

이 문장에서 별표를 제거하지 마십시오. 별표 사이에 담긴 텍스트는 볼드체로 표시될 것입니다. 텍스트 자체는 대화창에 포함된 텍스트일 경우가 많아, 응용 프로그램에서도 번역될 가능성이 큽니다.

다음은 :guilabel: reST 태그를 담고 있는 번역 항목입니다.

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

:guilabel: 태그의 텍스트 Advanced 는 QGIS 응용 프로그램에서도 번역될 가능성이 크기 때문에 번역해야 할 필요가 있을 겁니다!

다음 번역 항목은 ``airports`` 를 포함하고 있습니다. 이 인용 부호는 텍스트에 또다른 텍스트 폰트 속성을 부여하는 데 쓰입니다. 이 경우 ``airports`` 는 글자 그대로의 값으로 번역할 필요가 없습니다.

```
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>`_.
```

This item also includes a hyperlink with an url and an external presentation. The url should of course be left intact, you are allowed to change the external text https://www.qgistutorials.com which is visible by the reader. Never remove the underscore at the end of the hyperlink which forms an essential part of it!!

4.2.4 번역 규칙 요약

1. |bronze|, |checkbox|, |labels|, |selectString|, |addLayer| 와 같이 | 문자 두 개 사이의 텍스트를 변경하지 마십시오. 이미지를 대체하기 위해 쓰이는 특수 태그입니다.
2. :ref: 또는 :file: 태그로 시작하는 참조를 변경하지 마십시오.
3. figure_labels_1_ 처럼 언더바로 끝나는 참조를 변경하지 마십시오.
4. 하이퍼링크 내부에 있는 URL 을 변경하지 마십시오. 그러나 외부 설명 텍스트는 변경해도 됩니다. 하이퍼링크 마지막에 있는 언더바를 공백을 추가하지 말고 (>`_) 내버려두십시오.
5. :index:, :sup:, :guilabel: 그리고 :menuselection: 태그 뒤에 붙는 인용 부호 안의 텍스트를 번역하십시오. QGIS 응용 프로그램에서 해당 텍스트를 번역했는지, 했다면 어떻게 번역했는지 확인하십시오. 태그 자체는 변경하지 마십시오.
6. 이중 별표 그리고 이중 인용 부호 사이의 텍스트는 값 또는 파일명을 나타내는 경우가 많으며, 때로는 번역해야 하고 때로는 그대로 두어야 합니다.
7. `, ``, *, **, :: 과 같은 특수 문자를 소스 텍스트와 정확히 동일한 개수로 작성하도록 유의해야 합니다. 이런 특수 문자는 제공하는 정보의 스타일에 기여합니다.
8. 특수 문자 또는 태그에 담긴 텍스트는 공백으로 시작해서도 끝나서도 안 됩니다.
9. 번역된 문자열을 새 단락으로 끝내지 마십시오. 만약 그렇게 하면 HTML 생성 시 텍스트가 번역되지 않을 것입니다.

앞에서 정리한 규칙을 준수한다면 보기 좋은 번역 문서를 얻게 될 겁니다!

질문이 있다면, QGIS 커뮤니티 팀 또는 QGIS 번역 팀 으로 연락해주십시오.

CHAPTER 5

별명

5.1 사용법

To ease the use of icons in QGIS manuals, replacements are defined for each icon in `/source/substitutions.txt` file at QGIS-Documentation repository and some of these substitutions are listed below. Thus, when you want to use an icon from QGIS application in the documentation there is a big chance that there is already a substitution that can/should be used.

별명이 존재하지 않을 경우:

1. check the documentation repository whether the icon is available in `/static/common` folder. If no image, then you need to find and copy the icon image file from QGIS repository (often under https://github.com/qgis/QGIS/blob/release-3_10/images/themes/default folder) and paste (in `.png` format) under `/static/common` folder. For convenience and update, it's advised to keep filename when possible.
2. create the reference to the substitution in the `/source/substitutions.txt` file following the example below. The replacement text should be in camelCase:

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

3. run the `scripts/find_set_subst.py` script to update the substitution definitions in the rst files and include the new substitution(s).
4. (선택적) 아이콘을 가리키는 참조를 추가하고 그 별명을 다음 목록에 추가하십시오.

5.2 일반 별명

다음은 문서 작성 시 사용할 수 있는 아이콘과 그 별명의 예시입니다. 사용자 설명서의 여러 군데에서 찾아볼 수 있고, 사용할 수 있습니다.

5.2.1 플랫폼 아이콘

아이콘	별명	아이콘	별명
Q	|logo|		
K	|kde|	🐧	|nix|
X	|osx|	⊞	|win|

5.2.2 메뉴 항목

아이콘	별명	아이콘	별명
☑	|checkbox|	□	|unchecked|
◉	|radioButtonOn|	○	|radioButtonOff|
1,00 ⇕	|selectNumber|	... ▼	|selectString|
■ ▼	|selectColor|	▬ ▼	|selectColorRamp|
General	|tab|	°	|degrees|
Display name lakes.shp	|inputText|	○───	|slider|

5.3 툴바 버튼 아이콘

5.3.1 레이어 및 미리보기 관리

아이콘	별명	아이콘	별명
	|dataSourceManager|		
	|addOgrLayer|	DB2	|addDb2Layer|
	|addRasterLayer|		|addMssqlLayer|
	|addDelimitedTextLayer|		|addSpatiaLiteLayer|
	|addPostgisLayer|		|addOracleLayer|
	|addAfsLayer|		|addAmsLayer|
	|addMeshLayer|		
	|addVirtualLayer|		|addWmsLayer|
	|addWcsLayer|		|addWfsLayer|
	|newVectorLayer|		|newSpatiaLiteLayer|
	|newGeoPackageLayer|		|createMemory|
	|newVirtualLayer|		
	|dbManager|	GDAL	|gdal|
	|geoPackage|		|spatialite|
	|virtualLayer|		|wms|
	|wcs|		|wfs|
	|dbSchema|		
	|inOverview|		|addAllToOverview|
	|removeAllOVerview|		|removeLayer|
	|showAllLayers|		|hideAllLayers|
	|showMapTheme|		|showSelectedLayers|
	|hideSelectedLayers|		|hideDeselectedLayers|
	|toggleAllLayers|		
	|addLayer|		
	|indicatorEmbedded|		|indicatorFilter|
	|indicatorMemory|	?	|indicatorNoCRS|
⚠	|indicatorBadLayer|	☆	|favourites|

5.3.2 파일

아이콘	별명	아이콘	별명
	|fileNew|		|fileOpen|
	|fileSave|		|fileSaveAs|
	|fileExit|		

5.3.3 편집

아이콘	별명	아이콘	별명
	|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 식별 결과

아이콘	별명	아이콘	별명
	|expandTree|		|collapseTree|
	|expandNewTree|		|formView|
	|deselectAll|		|editCopy|
	|filePrint|		
	|identifyByRectangle|		|identifyByPolygon|
	|identifyByFreehand|		|identifyByRadius|

5.3.5 디지타이즈 작업 및 고급 디지타이즈 작업

아이콘	별명	아이콘	별명
	|cad|		|cadConstruction|
	|cadParallel|		|cadPerpendicular|
	|toggleEditing|		|allEdits|
	|tracing|		|snapping|
	|capturePoint|		|capturePolygon|
	|captureLine|		|deleteSelectedFeatures|
	|circularStringCurvePoint|		|circularStringRadius|
	|vertexTool|		|vertexToolActiveLayer|
	|moveFeature|		|moveFeatureCopy|
	|moveFeatureLine|		|moveFeatureCopyLine|
	|moveFeaturePoint|		|moveFeatureCopyPoint|
	|rotateFeature|		|rotatePointSymbols|
	|offsetCurve|		|offsetPointSymbols|
	|simplifyFeatures|		|reshape|
	|addRing|		|addPart|
	|fillRing|		
	|deleteRing|		|deletePart|
	|mergeFeatures|		|mergeFeatAttributes|
	|splitFeatures|		|splitParts|
	|reverseLine|		

5.3.6 맵 조작 및 속성

<table>
<tr><th>아이콘</th><th>별명</th><th>아이콘</th><th>별명</th></tr>
<tr><td></td><td>|pan|</td><td></td><td>|panToSelected|</td></tr>
<tr><td></td><td>|zoomIn|</td><td></td><td>|zoomOut|</td></tr>
<tr><td>1:1</td><td>|zoomActual|</td><td></td><td>|zoomFullExtent|</td></tr>
<tr><td></td><td>|zoomToLayer|</td><td></td><td>|zoomToSelected|</td></tr>
<tr><td></td><td>|zoomLast|</td><td></td><td>|zoomNext|</td></tr>
<tr><td></td><td>|refresh|</td><td></td><td></td></tr>
<tr><td></td><td>|identify|</td><td></td><td>|mapTips|</td></tr>
<tr><td></td><td>|showBookmarks|</td><td></td><td>|newBookmark|</td></tr>
<tr><td></td><td>|measure|</td><td></td><td>|measureArea|</td></tr>
<tr><td></td><td>|measureAngle|</td><td></td><td></td></tr>
<tr><td></td><td>|newMap|</td><td></td><td>|new3DMap|</td></tr>
<tr><td></td><td>|tiltUp|</td><td></td><td>|tiltDown|</td></tr>
<tr><td></td><td>|3dNavigation|</td><td>▶</td><td>|play|</td></tr>
</table>

5.3.7 선택 및 표현식

<table>
<tr><th>아이콘</th><th>별명</th><th>아이콘</th><th>별명</th></tr>
<tr><td></td><td>|selectRectangle|</td><td></td><td>|selectPolygon|</td></tr>
<tr><td></td><td>|selectFreehand|</td><td></td><td>|selectRadius|</td></tr>
<tr><td></td><td>|selectAll|</td><td></td><td>|deselectAll|</td></tr>
<tr><td></td><td>|invertSelection|</td><td></td><td>|expressionSelect|</td></tr>
<tr><td></td><td>|selectAllTree|</td><td></td><td>|select|</td></tr>
<tr><td></td><td>|formSelect|</td><td></td><td>|dataDefined|</td></tr>
<tr><td>ε</td><td>|expression|</td><td></td><td>|dataDefineOn|</td></tr>
<tr><td></td><td>|dataDefineExpressionOn|</td><td></td><td>|dataDefineError|</td></tr>
<tr><td></td><td>|dataDefineExpressionError|</td><td></td><td></td></tr>
<tr><td></td><td>|addExpression|</td><td></td><td></td></tr>
<tr><td></td><td>|expressionFilter|</td><td></td><td>|filterMap|</td></tr>
</table>

5.3.8 Labels and Diagrams

아이콘	별명	아이콘	별명
abc	|labeling|		|labelingNone|
abc abc	|labelingRuleBased|		|labelingObstacle|
	|piechart|		|diagramNone|
abc	|text|		|histogram|
	|annotation|	T	|textAnnotation|
	|formAnnotation|	</>	|htmlAnnotation|
*	|svgAnnotation|		|autoPlacement|
	|labelbackground|	abc	|labelbuffer|
+ab < c	|labelformatting|		|labelplacement|
	|labelshadow|		|render|
abc	|labelcallout|		
ab	|pinLabels|	abc	|showHideLabels|
abc	|moveLabel|	abc	|rotateLabel|
ab	|showPinnedLabels|	abc	|showUnplacedLabel|
abc	|changeLabelProperties|		

5.3.9 지도 장식

아이콘	별명	아이콘	별명
C	|copyrightLabel|		|addGrid|
T	|titleLabel|	N	|northArrow|
	|scaleBar|		|addMap|
	|addImage|		

5.3.10 도움말

아이콘	별명	아이콘	별명
?	|helpContents|		|qgisHomePage|
✓	|success|		
2010	|helpSponsors|	?	|contextHelp|

5.3.11 색상

아이콘	별명	아이콘	별명
	|colorBox|		|colorPicker|
	|colorSwatches|		|colorWheel|

5.4 기타 기본 아이콘

아이콘	별명	아이콘	별명
▼	|arrowDown|	▲	|arrowUp|
✚	|signPlus|	▬	|signMinus|
	|projectProperties|		|options|
	|interfaceCustomization|		|keyboardShortcuts|
	|copyrightLabel|		|northArrow|
	|scaleBar|		|tracking|
	|gpsImporter|		|gpsTrackBarChart|
	|folder|		|extents|
	|settings|	▶	|start|
	|3dconfigure|		|deleteSelected|
▶	|browserExpand|	▼	|browserCollapse|

5.5 속성 테이블

아이콘	별명	아이콘	별명
	|openTable|		|selectedToTop|
	|selectAll|		|invertSelection|
	|panToSelected|		|zoomToSelected|
	|copySelected|		|editPaste|
	|expressionSelect|		|deleteSelectedFeatures|
	|newAttribute|		|deleteAttribute|
	|newTableRow|		|calculateField|
	|refresh|		|formView|
	|conditionalFormatting|		|multiEdit|
	|dock|		|actionRun|
	|duplicateFeature|		
	|panTo|		|highlightFeature|

5.6 투영 및 지리참조자

아이콘	별명	아이콘	별명
	|geographic|		|crs|
	|customProjection|		|setProjection|
	|projectionDisabled|		|projectionEnabled|
	|georefRun|		|pencil|
	|linkQGisToGeoref|		|linkGeorefToQGis|
	|coordinateCapture|		|fullHistogramStretch|

5.7 인쇄 조판기

아이콘	별명	아이콘	별명
	|newLayout|		|layoutManager|
	|duplicateLayout|		
	|newReport|		|newPage|
	|atlasSettings|		|atlas|

다음 페이지에 계속

표 5.1 – 이전 페이지에서 계속

아이콘	별명	아이콘	별명
	|filePrint|		|saveMapAsImage|
	|saveAsSVG|		|saveAsPDF|
	|addBasicShape|		|addBasicCircle|
	|addBasicTriangle|		|addBasicRectangle|
	|addNodesShape|		|editNodesShape|
	|addPolygon|		|addPolyline|
	|addArrow|	N	|northArrow|
	|add3DMap|		|addMap|
	|addLegend|	</>	|addHtml|
	|addImage|		|addTable|
T	|label|		|scaleBar|
	|select|		|moveItemContent|
1:23	|setToCanvasScale|		|setToCanvasExtent|
1:23	|viewScaleInCanvas|		|viewExtentInCanvas|
	|raiseItems|		|lowerItems|
	|moveItemsToTop|		|moveItemsToBottom|
	|alignLeft|		|alignRight|
	|alignHCenter|		|alignVCenter|
	|alignTop|		|alignBottom|
	|resizeShortest|		|resizeTallest|
	|resizeNarrowest|		|resizeWidest|
	|resizeSquare|		
	|lockItems|		|unlockAll|
	|locked|		|unlocked|
	|lockedRepeat|		|lockedGray|
	|groupItems|		

5.8 Layer Properties

아이콘	별명	아이콘	별명
	|symbology|	abc	|labeling|
	|sourceFields|		|general|
i	|metadata|		|action|
	|display|		|rendering|
	|join|		|diagram|
	|legend|		|dependencies|
	|3d|		|system|
	|editMetadata|		|overlay|
	|digitizing|		|auxiliaryStorage|
	|history|		|stylePreset|
	|search|		|pyramids|
	|transparency|		|rasterHistogram|
	|singleSymbol|		|nullSymbol|
	|graduatedSymbol|		|categorizedSymbol|
	|25dSymbol|		|ruleBasedSymbol|
□	|invertedSymbol|	●	|heatmapSymbol|
	|pointDisplacementSymbol|		|pointClusterSymbol|
	|meshcontours|		|meshcontoursoff|
↘	|meshvectors|	↘	|meshvectorsoff|
▦	|meshframe|		
Σ	|sum|	A Z	|sort|
★	|paintEffects|		|mapIdentification|
a	|styleManager|		|iconView|
	|joinNotEditable|		|joinedLayerNotEditable|
	|joinHasNotUpsertOnEdit|		|filterTableFields|
	|symbologyEdit|		

5.9 플러그인

5.9.1 공간 처리

아이콘	별명	아이콘	별명
	|processing|		|processingModel|
	|processingHistory|		|processingResult|
	|processSelected|		
	|qgsProjectFile|		|addToProject|
	|mean|		|layerExtent|
	|randomSelection|		|vectorGrid|
	|convexHull|		|buffer|
	|intersect|		|union|
	|symDifference|		|clip|
	|difference|		|dissolve|
	|checkGeometry|		|exportGeometry|
	|delaunay|		|centroids|
	|toLines|		|extractNodes|
	|splitLayer|		
	|showRasterCalculator|		|heatmap|
	|showMeshCalculator|		

5.9.2 Various Core Plugins

기본 설치 시 제공되는 표준 플러그인이지만, 첫 설치 시 자동으로 로드되지 않는 플러그인입니다.

아이콘	별명	아이콘	별명
	|showPluginManager|		|installPluginFromZip|
	|pythonFile|		|runConsole|
	|showEditorConsole|		|clearConsole|
	|offlineEditingCopy|		|offlineEditingSync|
	|plugin|		|interpolation|
	|gdalScript|		|metasearch|
	|geometryChecker|		|topologyChecker|
	|fromSelectedFeature|		|sqlQueryBuilder|

5.9.3 GRASS 통합

아이콘	별명	아이콘	별명
	|grass|		|grassRegion|
	|grassTools|		|grassNewMapset|
	|grassOpenMapset|		|grassCloseMapset|

5.9.4 eVis 플러그인

아이콘	별명	아이콘	별명
	|eventBrowser|	ID	|eventId|
	|evisConnect|		