



QGIS Documentation Guidelines

QGIS Project

2020 年 12 月 09 日

目次

第 1 章	貢献のための一歩一歩	3
1.1	GitHub のウェブインタフェースを使用する	4
1.1.1	QGIS-Documentation をフォークする	5
1.1.2	変更を行う	5
1.1.3	ファイルを修正する	6
1.1.4	あなたの行った変更をプルリクエストを通して共有する	6
1.1.5	マージされたブランチを削除する	10
1.2	Git コマンドラインツールを使用する	11
1.2.1	ローカルリポジトリ	11
1.2.2	もうひとつのリモートリポジトリを追加	12
1.2.3	ベースブランチを更新する	13
1.2.4	制作ブランチに取り組む	14
1.2.5	変更を共有する	14
1.2.6	ローカルおよびリモートリポジトリをクリーンアップ	15
1.3	より詳しく知りたい場合は	15
第 2 章	執筆のためのガイドライン	17
2.1	ドキュメントを書く	18
2.1.1	見出し	18
2.1.2	リスト	18
2.1.3	行内タグ	19
2.1.4	Labels/references	20
2.1.5	図と画像	20
2.1.6	索引	24
2.1.7	特別なコメント	25
2.1.8	短いコード	25
2.1.9	脚注	25
2.2	スクリーンショットを管理する	26
2.2.1	新しいスクリーンショットを追加	26
2.2.2	翻訳されたスクリーンショット	27
2.3	プロセッシングアルゴリズムのドキュメントを作成する	27
第 3 章	PyQGIS クックブック中でコードを書く	33
3.1	テスト可能なコードスニペットの書き方	33
3.1.1	sphinx ディレクティブを文書化する	33
3.1.2	テストをグループ化する	35

3.2	ローカルマシンでスニペットをテストする方法	36
第 4 章	翻訳ガイドライン	37
4.1	翻訳のプロセス	37
4.2	ファイルを翻訳する	38
4.2.1	Transifex での翻訳	39
4.2.2	Qt Linguist による翻訳	40
4.2.3	マニュアルを翻訳する	42
4.2.4	翻訳ルールのまとめ	44
第 5 章	置換参照と定義	45
5.1	利用方法	46
5.2	一般的な置換	46
5.2.1	プラットフォームアイコン	47
5.2.2	メニューアイテム	47
5.3	ツールバーボタンアイコン	48
5.3.1	レイヤー管理と全体図	48
5.3.2	ファイル	49
5.3.3	編集	49
5.3.4	識別結果	49
5.3.5	デジタイズと高度なデジタイジング	50
5.3.6	地図ナビゲーションと属性	51
5.3.7	選択と式	51
5.3.8	ラベルとダイヤグラム	52
5.3.9	整飾	52
5.3.10	ヘルプ	52
5.3.11	色	53
5.4	他の基本的アイコン	53
5.5	属性テーブル	54
5.6	投影とジオリファレンス	54
5.7	印刷レイアウト	54
5.8	レイヤプロパティ	56
5.9	プラグイン	57
5.9.1	プロセッシング	57
5.9.2	コア・プラグインのいろいろ	57
5.9.3	Grass 統合	58
5.9.4	eVis プラグイン	58

QGIS ドキュメントは、米国太平洋時間の太平洋標準時の午前 0 時、午前 8 時、午後 4 時にサーバ上で自動作成されます。現在の状態は <https://docs.qgis.org> で入手できます。

QGIS ドキュメントのソースファイルは <https://github.com/qgis/QGIS-Documentation> で入手できます。これらは主に reStructuredText (reST) 形式の構文を使用して記述され、HTML 出力を後処理するための Sphinx ツールセットからのスクリプトと組み合わせられています。これらのツールの一般的な情報については <http://docutils.sourceforge.net/docs/ref/rst/restructuredtext.html> または <https://www.sphinx-doc.org/en/master/usage/restructuredtext/basics.html> 参照。

以下の章では次のようなことを学ぶことができます。

- how to manage the documentation source files using [git](#) system and the [GitHub](#) platform on which they are stored
- ルールに準拠した方法でテキストを変更し、スクリーンショットを提供する方法
- あなたの行った変更を確実に公式ドキュメントに取り込み共有する方法

QGIS プロジェクトに貢献する方法に関する一般的な情報を探している場合は、[QGIS コミュニティに参加する](#) にヘルプを見つけることができます。

第 1 章

貢献のための一步一步

- *GitHub* のウェブインタフェースを使用する
 - *QGIS-Documentation* をフォークする
 - 変更を行う
 - * 選択肢 1: *Edit on GitHub* ショートカットを使用する
 - * 選択肢 2: あなたのドキュメンテーションレポジトリに一時的な専用ブランチを作成する
 - ファイルを修正する
 - あなたの行った変更をプルリクエストを通して共有する
 - * 新しくプルリクエストを始める
 - * 変更を比較する
 - * プルリクエストを説明する
 - * プルリクエストをレビューしコメントする
 - * 訂正を加える
 - マージされたブランチを削除する
- *Git* コマンドラインツールを使用する
 - ローカルリポジトリ
 - もうひとつのリモートリポジトリを追加
 - ベースブランチを更新する
 - 制作ブランチに取り組む
 - 変更を共有する
 - ローカルおよびリモートリポジトリをクリーンアップ

- より詳しく知りたい場合は

注釈: ここでは手順の説明のために QGIS ドキュメントを使用しますが、以下で説明するコマンドおよび手順はすべて、QGIS ウェブサイトにも当てはまります。

あなたがこの文章を読まれているということはきっと、QGIS ドキュメントに貢献しようという気持ちがあって、そのための方法を探されているのでしょう。ここで提供しようとしているのは、まさしくそれです。現在この文書では、目的を達成するための複数の方法を一通り案内し、従うべき主な手順を示し、使用可能な小技と知っておくべき落とし穴を教えています。

何か助けが必要なときには迷わず、修正しようとしている Issue レポートのコメントに書き込むか、[QGIS コミュニティチームメーリングリスト](#) に投稿をしてください。より詳しくは [ドキュメントを書く](#) をご覧ください。

さあ、では集中して始めてみましょう。

ドキュメントのソースは Git バージョンコントロールシステムを使用して保存されており、GitHub <https://github.com/qgis/QGIS-Documentation> で利用することができます。ファイルに変更を加えるには主に 2 つの方法がありますが、どちらか一方の方法しか選べないというわけではありません。

1. [GitHub のウェブインタフェース](#)を使用する
2. [Git コマンドラインツール](#)を使用する。

1.1 GitHub のウェブインタフェースを使用する

GitHub のウェブインタフェースでは次のことを行うことができます。

- ファイルを編集する
- 変更をプレビューし、コミットする
- 変更がメインリポジトリに挿入されるようにプルリクエストを行う
- ブランチを作成、更新または削除する

git や GitHub で使われる言葉にまだ馴染みがない場合は、GitHub の [Hello-world](#) プロジェクトを読んで、以下でも使われる基本的な語彙と操作について学んだ方がいいでしょう。

注釈: 報告された Issue をあなたが修正中なら

もしあなたが [issue](#) を修正するための変更を加えている最中なら、issue レポートにコメントをして、それをあなた自身に割り当ててください。これにより、複数の人が同じ issue に取り組むことを防ぐことができます。

1.1.1 QGIS-Documentation をフォークする

GitHub アカウント はすでに取得しているものと仮定すると、まず最初にするべきは、ドキュメントのソースファイルをフォークすることです。

QGIS-Documentation のリポジトリ ページに移動して、右上隅の  ボタンをクリックします。

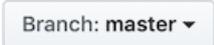
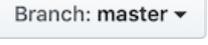
ご自身の GitHub アカウントに、QGIS-Documentation リポジトリ (<https://github.com/<YourName>/QGIS-Documentation>) が作られていることと思います。このリポジトリは公式の QGIS-Documentation リポジトリのコピーです。あなたに完全な書込権限が与えられていて、公式のドキュメントに影響を与えることなく変更を加えることができます。

1.1.2 変更を行う

QGIS ドキュメンテーションに貢献するにはいくつかの異なった方法があります。以下ではそれらを別々に示しますが、あるプロセスから別のプロセスに切り替えることに何ら問題はありません。

選択肢 1: Edit on GitHub ショートカットを使用する

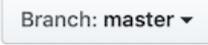
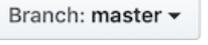
QGIS ドキュメントのそれぞれのページは、ページの右上にある Edit on GitHub というリンクをクリックすると、素早く簡単に編集することができます。

1. Edit on GitHub をクリックすると、`qgis:master` ブランチのファイルが開きます。ページの上部には、あなたにはこのリポジトリへの書込権限がないため、変更はあなたのリポジトリの新しいブランチで適用される旨を知らせるメッセージが、表示されます。
2. 変更を行います。ドキュメントは `reStructureText` シンタックスを使用して書かれていますので、変更の内容によっては、執筆のためのガイドライン を参照しながら行う必要があるかもしれません。
3. 終了したら、行った変更について短いコメントを書いて、*Propose changes* をクリックします。これによってあなたのリポジトリに新しい **ブランチ** (`patch-xxx`) が作成されます。
4. *Propose changes* をクリックすると、GitHub の *Comparing changes* ページに移動します。
 - すべての変更が終わったら、下の **プルリクエストで変更をシェアする** のセクションの、**変更を比較** にスキップしてください。
 - QGIS に送信する前に追加の変更が必要な場合は、次の手順に従ってください。
 1. フォークしたあなたの QGIS-Documentation (<https://github.com/<YourName>/QGIS-Documentation>) リポジトリに移動します。
 2.  をクリックして ``patch-xxx`` ブランチを探し、このブランチを選択します。  ボタンが *Branch: patch-xxx* に変わります。
 3. 下の **ファイルを修正する** に飛んでください。

注釈: Edit on GitHub ショートカットは左サイドバーの一番下のドロップダウンメニューからも利用できます。

選択肢 2: あなたのドキュメンテーションレポジトリに一時的な専用ブランチを作成する

あなたがフォークした QGIS-Documentation で直接ファイルを編集できます。

フォークした QGIS-Documentation レポジトリの左上隅にある  をクリックして、テキストフィールドに一意の名前を入力して新しい **ブランチ** を作成します。新しいブランチの名前は、修正しようとしている問題に関連していなければなりません。すると  ボタンは *Branch: branch_name* となるはずです。

ちなみに: 変更はこの一時的な専用ブランチで行うこと。絶対に `master` ブランチでは行わないこと

`qgis/QGIS-Documentation` の `master` ブランチからあなたの QGIS-Documentation レポジトリに変更をマージする場合を除いて、慣例として `master` ブランチでは変更を行わないでください。問題ごとに別々のブランチを使用することによって、他のブランチに干渉することなく、同時に複数の問題に取り組むことができます。間違えた場合は、いつでもブランチを削除して、`master` ブランチから新しいブランチを作成してやり直すことができます。

1.1.3 ファイルを修正する

1. フォークした QGIS-Documentation のソースファイルをブラウザして、修正する必要があるファイルに移動します。
2. 執筆のためのガイドライン に従いながら修正を行います。
3. 終了したら、ページの一番下にある **Commit Changes** フレームに移動し、行った変更について短いコメントを書き、そして *Commit Changes* をクリックしてあなたのブランチに直接変更をコミットします。 *Commit directly to the branch_name branch.* が選択されていることを確認してください。
4. 問題を修正するために更新する必要がある他のファイルについて上記の手順を繰り返します。

1.1.4 あなたの行った変更をプルリクエストを通して共有する

あなたの変更を公式ドキュメントに統合するためにはプルリクエストをする必要があります。

注釈: Edit on GitHub リンクを使用して始めた場合は

変更をコミットした後、GitHub はあなたの `patch-xxx` ブランチで行った変更を `qgis/QGIS-Documentation` マスターブランチと比較する新しいページを自動的に開きます。

下の *Step 2* にスキップしてください。

新しくプルリクエストを始める

QGIS-Documentation リポジトリのメインページに行き、 *New pull request* をクリックします。

変更を比較する

一方が `base:master`、もう一方が `compare:branch_name` (図を参照) という 2 つのダイアログボックスが表示されている場合は、あなたの行った変更はあなたのリポジトリの中で、変更を加えたブランチからあなたのマスターブランチへとマージされるだけです。これを修正するために compare across forks と表示されているリンクをクリックします。`

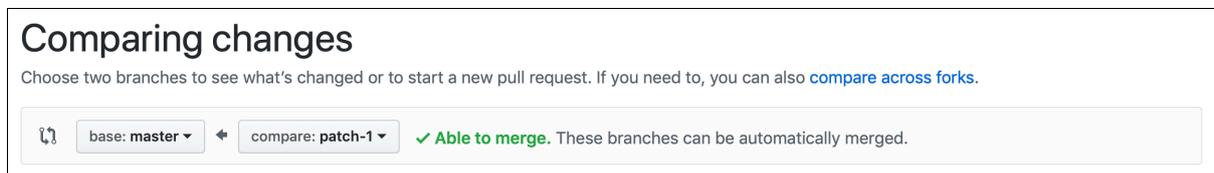


図 1.1 *Comparing changes* ページがこのようなものだったら、 *compare across forks* リンクをクリックします。

4 つのドロップダウンメニューが表示されていることと思います。このドロップダウンメニューによって、あなたのブランチで行った変更を、変更をマージしたい相手である公式のマスターブランチと比較することが可能になります。4 つのドロップダウンメニューは以下の通りです。

- **base fork** : あなたの変更をマージしたい相手先のフォーク
- **base** : あなたの変更をマージしたい base fork のブランチ
- **head fork** : base fork に組み込みたい変更があるフォーク
- **compare** : 変更が行われたブランチ

base fork で `qgis/QGIS-Documentation` を、base で `master` を選択します。head fork をあなたのリポジトリ `<YourName>/QGIS-Documentation` に、compare をあなたが変更を行ったブランチに設定します。

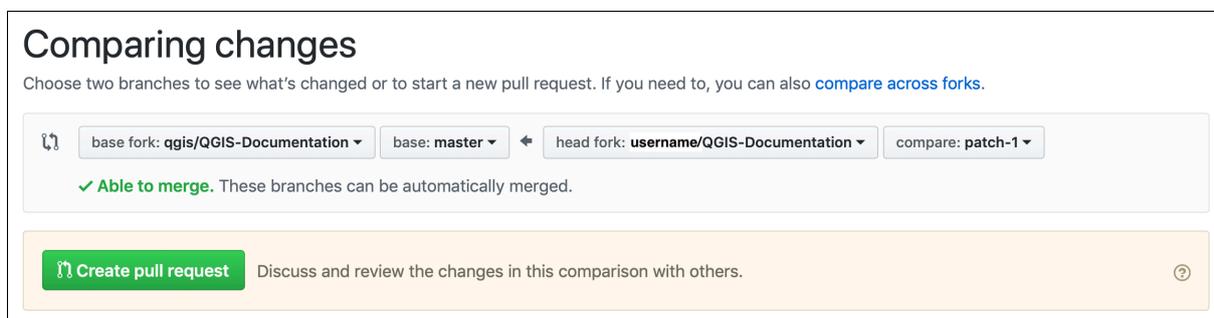


図 1.2 `qgis/QGIS-Documentation` とあなたのリポジトリとの間で変更を比較している

Able to merge と書いてある緑色のチェックマークは、あなたの変更が衝突することなく公式のドキュメントにマージできることを示しています。

Create pull request ボタンをクリックします。

警告: **X Can't automatically merge.** と表示されたら

これは **競合** があることを意味します。他の誰かがあなたの変更と競合するコミットをしたので、あなたが修正中のファイルは対象のブランチの最新の状態を反映していません。それでもプルリクエストは作成できますが、マージを完了するには **競合** を修正する必要があります。

ちなみに: **最新バージョン** の QGIS ドキュメントは、翻訳されてはいますが、保守は継続しており、問題が見つかれば修正が行われています。特定リリースの問題を修正する場合は、上記の手順で **base** を `master` から適切な `release_...` ブランチに変更してください。

プルリクエストを説明する

テキストボックスが開きます。対処している問題に関するコメントを入力します。

これが特定の **issue** に関連する場合は、コメントに **issue** 番号を追加します。これは # と **issue** 番号 (例 #1234) を入力することによって行われます。直前に `fix` や `close` のような語が置かれている場合は、プルリクエストがマージされると、すぐにその **issue** はクローズされます。

あなたの変更を行ったすべてのドキュメントのページへのリンクを含めてください。

Create pull request をクリックします。

プルリクエストをレビューしコメントする

上で見たように、誰でもプルリクエストを通してドキュメントの修正を提案することができます。同様に、誰でも質問や **コメント** でプルリクエストをレビューすることが可能です。おそらくそれは、プロジェクトのガイドラインに沿っていない執筆スタイルだったり、重要な細部やスクリーンショットが変更欠けていたり、あるいはすべてが素晴らしくて申し分なさそうだ、ということだったりするでしょう。レビューはドキュメントの形式と実質の両面において、貢献の質を高める助けになります。

プルリクエストをレビューするには

1. **pull requests** の **ページ** へ移動して、コメントをしたいプルリクエストをクリックします。
2. ページの一番下に、このプルリクエストについて全体的なコメントを残すことのできるテキストボックスがあると思います。
3. 特定の行についてコメントをするには、

1.  **Files changed** をクリックして、コメントをしたいファイルを探します。変更を確認するために *Display the source diff* をクリックしないとイケないかもしれません。
2. コメントしたい行までスクロールして、 をクリックします。テキストボックスが開きますので、コメントを残すことができます。

特定の行に対するコメントは、次のいずれかの方法で公開することができます。

- *Add single comment* ボタンを使って、単独のコメントとして公開する。この場合コメントはその都度即座に公開されます。コメント数が少ない場合や他のコメントにコメントする場合にのみ、この方法を使ってください。
- *Start a review* ボタンを押して、レビューの一部として公開する。コメントは検証後に自動的に送信されないため、後で編集やキャンセルしたり、レビューの主要な点を要約したりそのプルリクエストについての全体的な指示を追加したり、そのプルリクエストを承認するかどうかを述べたりすることができます。こちらの方がよりよい方法でしょう。より柔軟ですし、レビューを構造化したりコメントを編集したりして準備ができてから公開することができますし、リポジトリのフォロワーにコメントごとに通知を送るのではなく、ひとつだけ通知を送ることができるからです。より詳しくは [を参照してください](#)。



図 1.3 特定の行に対して修正の提案とともにコメントを行う

行に対するコメントには提案を埋め込むことができ、プルリクエストの作成者はこれをプルリクエストに適用することができます。提案を追加するには、コメントテキストブロックの上にある  **Insert a suggestion** ボタンをクリックして、提案ブロックの中のテキストを修正します。

ちなみに：プルリクエストへの提案をバッチとして追加したいときは

プルリクエストの作成者としてレビュワーからのフィードバックを直接プルリクエストに組み込むときに、対処すべき提案の数が多いためにそれらをバッチコミットとして追加したい場合は、コメントの一番下にある *Commit suggestion* ボタンを使用するのは避けてください。すなわち、以下のようにします。

1.  **Files changed** タブに移動します。
2. 取り込みたい提案のそれぞれで *Add suggestion to batch* を押します。押すごとにカウンターが増えるの

が分かると思います。

3. 取り込みたいすべての提案をプルリクエストに適用する準備ができたら、どれでもいいので *Commit suggestions* ボタンを押して、この変更を説明するメッセージを入力します。

これによってすべての修正がひとつのコミットとしてブランチに追加されます。結果として、変更履歴が読みやすいものとなり、リポジトリのフォロワーに送信される通知の数も減ります。ついでに言えば、この処置によってあなたのクリック数も大きく減らすことができます。

訂正を加える

新しいプルリクエストは自動的に [プルリクエストリスト](#) に追加されます。他の編集者や管理者があなたのプルリクエストを見直し、提案をしたり修正を求めたりするかもしれません。

プルリクエストによって [Travis CI ビルド](#) も実行されます。これはあなたの貢献をビルドエラーについて自動的にチェックします。Travis CI がエラーを見つけた場合は、コミットの横に赤い×印が表示されます。エラーの詳細を見るには、プルリクエストページの下部にある概要セクションの赤い十字または「詳細」をクリックしてください。変更が `qgis/QGIS-Documentation` リポジトリにコミットされる前に、報告されたエラーや警告を修正する必要があります。

メインリポジトリにマージされるまでは、プルリクエストに修正を加えることができます。それはプルリクエストを改善するためだったり、提案された修正に対処するためだったり、あるいはビルドエラーを修正するためだったりするでしょう。

変更するにはプルリクエストのページで  `Files changed` をクリックして、変更したいファイル名の横にある鉛筆ボタン  をクリックします。

プルリクエストで使用したのと同じブランチで変更を加えた場合、追加された変更はすべてプルリクエストに自動的に追加されます。このため、追加の変更は、プルリクエストで修正しようとしている問題にその変更が関連する場合にのみ、行うようにしてください。

別の問題を解決したい場合は、それらの変更に対して新しいブランチを作成して上記のステップを繰り返します。

ビルドエラーが修正され、あなたと管理者が変更で満足したら、管理者はあなたの貢献をマージします。

1.1.5 マージされたブランチを削除する

変更がマージされた後でブランチを削除できます。古いブランチを削除すると、未使用のブランチや古いブランチをリポジトリに保存しておく必要がなくなります。

1. フォークしたあなたの `QGIS-Documentation` リポジトリ (<https://github.com/<YourName>/QGIS-Documentation>) に移動します。
2. `Branches` タブをクリックします。 `Your branches` の下にあなたのブランチがあることと思います。

3. 不要なブランチの  Delete this branch アイコンをクリックして削除します。

1.2 Git コマンドラインツールを使用する

GitHub のウェブインターフェイスは、簡単なやり方で QGIS-documentation レポジトリの更新に貢献することができますが、下記のためのツールは提供していません。

- 複数のコミットをまとめることによって変更履歴をきれいにする
- メインリポジトリとの間で生じうる衝突を解決する
- あなたの変更をテストするためにドキュメントをビルドする

より高度で強力なツールへアクセスし、リポジトリのコピーをローカル環境に持つためには、ハードドライブ上に `git` をインストールする必要があります。しばしば必要となる基本的な事項は以下で説明されています。そこではウェブインターフェイスを使用する場合であっても気をつけるべきルールを学ぶことができますでしょう。

以下のコードサンプルでは、`$` で始まる 行はあなたが入力すべきコマンドを示します。一方、`#` で始まる 行はコメントです。

1.2.1 ローカルリポジトリ

さあ、QGIS-Documentation リポジトリの**あなた用の**コピーをローカル環境に取得する準備はできましたね。

ウェブ URL を使った以下のコマンドで、QGIS ドキュメンテーションリポジトリを複製し取得することができます。

```
# 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
```

上記のコマンドラインはただ単に一例です。<YourName> を自分のユーザー名で置き換え、ローカルパスとリポジトリ URL の両方を実際に合わせて変更しなければなりません。

次に以下の確認を行ってください。

```
# 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` は、あなたの QGIS-Documentation リポジトリの、リモートリポジトリにつけられた名前です。

- *master* はデフォルトのメインブランチです。貢献するにはこのブランチを使用してはいけません。絶対にです！

SSH プロトコルを使用して QGIS ドキュメンテーションリポジトリを複製することもできます。

```
# 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
```

ちなみに: **Permission denied (publickey)** エラーが出た時は？

上記のコマンドで **Permission denied (publickey)** というエラーが出た場合は、おそらくはあなたの SSH key に問題があります。詳細は [GitHub のヘルプ](#) を参照してください。

SSH プロトコルを使用した場合は確認は以下のようになります。

```
# 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
```

これで始めることができますが、長い間のうちには、あなたの貢献をプッシュした際 (GitHub のプロセスではプルリクエストと呼びます) に、たくさん問題が生じると思います。これは公式の `qgis/QGIS-Documentation` リポジトリの `master` ブランチが、あなたのローカル/リモートリポジトリからだんだんと差異を増してずれていくことによるものです。このため、常にメインリモートリポジトリの状態を追跡したうえで、ブランチにおける作業を行う必要があります。

1.2.2 もうひとつのリモートリポジトリを追加

メインプロジェクトで行われた作業を追跡できるようにするために、ローカルリポジトリに新しいリモートリポジトリを追加します。この新しいリモートリポジトリは、QGIS プロジェクト公式の `QGIS-Documentation` リポジトリです。

```
$ 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)
```

ローカルリポジトリにリモートリポジトリを追加するときにも、同様に SSH プロトコルを使うことができます。

```
$ 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)
```

これで2つのリモートリポジトリの、どちらかを選ぶことができるようになりました。

- *origin* はあなたの リモートリポジトリに、あなたのローカルブランチをプッシュするときに使用します。
- *upstream* は、あなたの貢献を公式のリポジトリにマージしたり（その権限がある場合）あなたのローカルリポジトリのマスターブランチを公式リポジトリのマスターブランチに従って更新したりする際に使用します。

注釈: *upstream* は標準の名前のようになっていますが、実際はただのラベルですので、あなたの好きなように名前をつけることができます。

1.2.3 ベースブランチを更新する

新しい貢献に取り組む前には、必ず、ローカルリポジトリのマスターリポジトリをアップデートしなければなりません。

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

これであなたのローカルリポジトリとリモートリポジトリ双方の *master* ブランチを、公式 QGIS-Documentation リポジトリの *master* ブランチに一致するようアップデートしましたので、貢献の作業を始めることができます。

注釈: リリースドキュメントに貢献したい場合はブランチを移動すること

testing 版ドキュメントとともに、[latest release](#) 版ドキュメントについても問題を修正する作業が続けられていますので、このドキュメントに対して貢献することも可能です。コード中の *master* を最新の対応するブランチに置き換えた上で、前セクションのサンプルコードに従ってください。

1.2.4 制作ブランチに取り組む

ベースブランチを最新に更新しましたので、次は自分の貢献を追加する専用のブランチを作成する必要があります。作業は必ずベースブランチ以外のブランチで行ってください！これは常にです！

```
# 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
```

commit/push コマンドについて一言：

- ひとつの貢献（それ以上分割することが不可能な変更）だけをコミットするようにしてください。すなわち一度にひとつの問題だけに取り組んでください。
- コミットのタイトルおよび説明文の中で、変更の内容を丁寧に説明するようにしてください。最初の行はタイトルです。大文字で始め、80文字以内に収め、最後には . を付けしないでください。簡潔にしてください。コミットの説明文は長くなってもよいので、より多く詳細について語るすることができます。最後は . で終了してください。
- Issue を参照するには、# を頭につけた Issue 番号を使用してください。チケットを修正する場合は Fix をその前につけておくと、コミットによってチケットが閉じられます。

さあ変更が保存されローカルブランチにコミットされました。プルリクエストを作成するためには、リモートリポジトリに送信する必要があります。

```
$ git push origin myNewBranch
```

1.2.5 変更を共有する

これであなたの github のリポジトリに行って、前のセクションで触れたように [プルリクエストを作成](#) することができます。作成したプルリクエストが、自分のブランチから公式の QGIS-Documentation リポジトリ中のターゲットとしているリモートブランチへものであることを確認してください。

1.2.6 ローカルおよびリモートリポジトリをクリーンアップ

プルリクエストが公式の QGIS-Documentation にマージされたら、あなたの制作ブランチは削除してかまいません。この方法で多くの作業をこなした場合、数週間のうちに用済みのブランチがたくさんできると思えます。ですので以下のようにしてあなたのリポジトリをきれいに保ちましょう。

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

またローカルリポジトリ中の `master` ブランチを更新して最新の状態を保つことも忘れないでください！

1.3 より詳しく知りたい場合は

- 上記の Github ウェブインターフェイスと `git` コマンドラインツール以外にも、ドキュメントへの貢献を作成および管理するために使用できる **GUI アプリケーション** があります。
- プルリクエストの変更が、ターゲットブランチにプッシュされた最近の変更と競合している場合は、マージが可能になるよう、先にこの競合を解決する必要があります。
 - 競合が競合する数行に関連している場合は、Github のプルリクエストのページに *Resolve conflicts* ボタンがあります。 <https://help.github.com/articles/resolving-a-merge-conflict-on-github/> で説明されているようにボタンを押し、問題を解決して下さい
 - 競合がファイルの名前変更または削除を伴う場合は、`git` コマンドラインを使用して競合を解決する必要があります。典型的には、最初に `git rebase targetBranch` 呼び出しを使ってターゲットブランチの上にあなたのブランチをリベースし、報告された衝突を修正しなければなりません。詳細は <https://help.github.com/articles/resolving-a-merge-conflict-using-the-command-line/> をご覧ください。
- 場合によっては、校正プロセスの最後に、変更が複数のコミットに分割されてしまうことがあります。分割されたコミットが必ずしもそれだけの価値があるわけではありません。Git コマンドラインは、これらのコミットをより少数の、より意味のあるコミットメッセージに変換するのに役立ちます。より詳しくは <https://help.github.com/articles/using-git-rebase-on-the-command-line/> を参照してください。

第 2 章

執筆のためのガイドライン

- ドキュメントを書く
 - 見出し
 - リスト
 - 行内タグ
 - *Labels/references*
 - 図と画像
 - * 画像
 - * 置換
 - * 図
 - * 表
 - 索引
 - 特別なコメント
 - 短いコード
 - 脚注
- スクリーンショットを管理する
 - 新しいスクリーンショットを追加
 - 翻訳されたスクリーンショット
- プロセッシングアルゴリズムのドキュメントを作成する

総じて、QGIS プロジェクトのために reST ドキュメントを作成するときには、[Python documentation style guidelines](#) に従ってください。簡便のために以下に、QGIS ドキュメントを書く際に依拠すべき一般的なルー

ルを、一通り示します。

2.1 ドキュメントを書く

2.1.1 見出し

ドキュメントのそれぞれのウェブページには、ひとつの `.rst` ファイルが対応しています。

テキストを構造化するために使用されるセクションはそれらのタイトルを通じて識別されます。タイトルには下線（及び第 1 レベルに対して上線）が引かれます。同じレベルのタイトルは下線装飾のために同じ文字を使用する必要があります。QGIS 文書では、章、セクション、サブセクションと minisec に対して以下のスタイルを使用する必要があります。

```
*****  
Chapter  
*****  
  
Section  
=====
```

Subsection

Minisec
.....

Subminisec
^^^^^^^^^^

2.1.2 リスト

リストはテキストを構造化するのに役立ちます。こちらはすべてのリストに共通な簡単な規則のいくつかです：

- すべての項目を大文字で始めてください
- 単一の単文のみを含むリスト項目の後に句読点を使用しないでください
- 複数の文または 1 つの複合文からなるリスト項目の句読点としてピリオド（.）を使用します

2.1.3 行内タグ

You can use tags to emphasize items.

- **メニュー GUI** : サブメニューを選択したり、特定の操作、またはこのような配列の何らかの部分配列を選択するなど、メニュー選択の完全な配列をマークします。

```
: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`
```

(以下の画像を参照してください)。

- **キーボードショートカット**

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

will show Ctrl+B

キーボードショートカットを説明するときは、次の規則を使用してください：

- Letter keys are displayed using uppercase: S
- Special keys are displayed with an uppercase first letter: Esc
- キーの組み合わせはキー同士の間空白を入れずに + 記号を表示し、Shift+R のように表示します。

- **ユーザーテキスト**

```
` `label` `
```

2.1.4 Labels/references

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

以下の例は、セクション（例えば、ラベル/参照タイトル）のアンカーを作成します

```
.. my_anchor:  
  
Label/reference  
-----
```

同じページ中の参照を呼び出すために使用するの

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

これが返すのは：

詳細については [my_anchor](#) を参照してください。

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.

文書内のどこからでも同じ場所にジャンプする別の方法は、`:ref:` 役割を使用することです。

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

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

詳細については [Labels/references](#) を参照。

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:

詳細については [ラベルや参照](#) を参照。

2.1.5 図と画像

画像

画像を挿入するには、使用します

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

これを返します



置換

テキスト内には画像を置くか、どこでも使用される別名を追加できます。段落内で画像を使用するには、最初に `source/substitutions.txt` ファイル中に別名を作成します：

```
.. |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.

注釈: Currently, to ensure consistency and help in the use of QGIS icons, a list of aliases is built and available in the [置換参照と定義](#) chapter.



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

   A caption: A logo I like
```

結果は以下のようになります。

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.



図 2.1 キャプション：私の好きなロゴ

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

キャプションを使用するには (*My caption* を参照)、図ブロックに空白行の後ろに字下げテキストを挿入します。

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

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

renders like this:

see 図 2.1

This is the preferred way of referencing figures.

注釈: 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:

キャプション：私の好きなロゴ 参照

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.

表

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_.

```

結果 :

ウィンドウズ	macOS
	
そしてもちろん🐧を忘れないように	

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

`my_drawn_table` のように参照できます。

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
```

結果：

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

2.1.6 索引

索引は読者がドキュメント内で必要な情報を見つけるのを手助けする手軽な方法です。QGIS ドキュメントは必須の索引項目のみを提供しています。本当に有用な（よく整理され、首尾一貫し、相互に関連づけられた）索引項目のセットのみを提供する助けとなるルールがいくつかあります。

- 索引は、人間が読めるもの、理解できるもの、翻訳可能なものでなければなりません。索引は複数の単語から作ることができますが、それらを繋ぐ不要な `_` や `-` などは使用しないようにしてください。たとえば `loading_layers` や `loadingLayers` ではなく、`Loading layers` としてください。
- 特別な綴りを持つ単語でない限りは、索引語の最初の文字のみを大文字にします。たとえば、`Loading layers`、`Atlas generation`、`WMS`、`pgsql2shp` などとします。
- 常に現在の [索引語リスト](#) に注意を払うことで、より好適な表現を正しい綴りで再利用し、不要な重複は避けるようにしてください。

RST には索引用のタグがあります。行内タグ `:index:` は通常のテキスト中で次のように使います。

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

ブロックレベルのマークアップの `.. index::` を使うこともできます。これは次の段落の始まりにリンクします。上記ルールにより、ブロックレベルタグの使用を推奨します。

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

single、pair、see のような索引パラメータの使用も推奨します。これはより構造化され相互に関連した索引語テーブルを作るのに有用です。索引作成についてのより詳しい情報は [Index generating](#) を参照してください。

2.1.7 特別なコメント

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 短いコード

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](#).

注釈: 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 脚注

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.

これは、脚注を作成するためのものです（例として示します^{*1}）

```
blabla [1]_
```

これが指しているのは：

*1 コアプラグインの更新

2.2 スクリーンショットを管理する

2.2.1 新しいスクリーンショットを追加

以下は、新しいスクリーンショットを、見た目良く作成するためのヒントです。画像ファイルは、参照元の `.rst` ファイルがあるフォルダの中の、画像フォルダ (`img/`) に置きます。

- スクリーンショットを作成するために使うことのできる QGIS プロジェクトが、このリポジトリの `./qgis-projects` フォルダに用意されています。これは次期バージョン QGIS のためのスクリーンショットの再作成を多少なりとも容易にします。このプロジェクトでは QGIS の [サンプルデータ](#) (またの名を Alaska データセット) を使用しています。サンプルデータは QGIS-Documentation リポジトリと同じフォルダになければなりません。
- ウィンドウは説明に必要な範囲で最小にします (小さなモーダルウィンドウのためだけに全画面表示するのは過剰です)。
- ごちゃごちゃしていないほど良いです (すべてのツールバーをアクティブにする必要はありません)。
- 画像編集ソフトでリサイズしないでください。サイズは必要であれば `.rst` ファイル中で設定されます (解像度を適切に上げることなく大きさを縮小すると、画像が汚くなります)。
- 背景はカットします。
- 背景が白でない場合は上部の角を透明にします。
- 印刷サイズ解像度を 135 dpi に設定します (たとえば Gimp で印刷解像度を設定するにはメニューで *Image Print size* を選択して保存します)。これによって、画像は HTML ではオリジナルサイズで、PDF では印刷に適した解像度となります。ImageMagick の `convert` コマンドを使えば、この処理を次のようにバッチ処理で行うこともできます。

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

- `.png` で保存します (`.jpeg` は避けてください)。
- スクリーンショットはテキストの記載に従った内容を表示していなければなりません。

ちなみに: 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 翻訳されたスクリーンショット

翻訳されたユーザガイド用にスクリーンショットを作成したい人向けのヒントです。

翻訳された画像は `img/<your_language>/` フォルダに置きます。英語のオリジナルのスクリーンショットと同じファイル名を使用してください。

2.3 プロセシングアルゴリズムのドキュメントを作成する

プロセシングアルゴリズムのドキュメントを書きたいときは、以下のガイドラインを考慮してください。

- 処理アルゴリズムのヘルプファイルは、QGIS ユーザーガイドの一部であるので、ユーザーガイドとその他の文書と同じフォーマットを使用します
- 各アルゴリズムのドキュメントは対応する プロバイダ フォルダと グループ ファイル中に置いてください。アルゴリズム ボロノイポリゴンは *QGIS* プロバイダと *vectorgeometry* グループに属します。そのため、説明を追加する正しいファイルは `source/docs/user_manual/processing_algs/qgis/vectorgeometry.rst` です。

注釈: ガイドを書き始める前に、そのアルゴリズムの説明がないかどうかを確認してください。すでにある場合は、既存の説明を拡張できます。

- 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
-----
```

アルゴリズム名を調べるには、「プロセシング」ツールボックスのアルゴリズム上にマウスを置くだけです。

- 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.
- 説明中ではアルゴリズムに QGIS のデフォルトショートカットがあるか、またはインプレース編集をサポートするのを示します。

- Add images! A picture is worth a thousand words! Use .png format and follow the general guidelines for documentation (see the [図と画像](#) 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.
- アルゴリズムのパラメータと出力について明確に説明してください。既存のアルゴリズムからヒントを得てください。
- 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
```

- サポートされている入力のタイプを説明してください。1つを選べる利用できるタイプはいくつがあります：

Parameter/Output type	説明	Visual indicator
Point vector layer	vector: point	
Line vector layer	vector: line	
Polygon vector layer	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	
式	expression	
Point geometry	coordinates	
領域	範囲	

次のページに続く

表 2.1 – 前のページからの続き

Parameter/Output type	説明	Visual indicator
CRS	crs	
Enumeration	enumeration	
List	list	
数値	number	<input type="text" value="1,00"/>
文字列	string	Display name <input type="text" value="lakes.shp"/>
真偽値	boolean	<input checked="" type="checkbox"/>
Folder path	フォルダ	
ファイル	file	
行列	matrix	
レイヤー	layer	
Same output type as input type	same as input	
Definition	definition	
点	point	
MultipleLayers	multipleLayers	
Range	range	
AuthConfig	authconfig	
メッシュ	mesh	
Layout	layout	
LayoutItem	layoutitem	
Color	color	
Scale	scale	

- Study an existing and well documented algorithm, and copy all the useful layouts.
- When you are finished, just follow the guidelines described in [貢献のための一歩一歩](#) to commit your changes and make a Pull Request

こちらがレイアウトと説明の助けになる 既存アルゴリズム の例です:

```

.. _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

```

(次のページに続く)

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
```

```
* - 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
```

(前のページからの続き)

```
* - **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
```


第 3 章

PyQGIS クックブック中でコードを書く

- テスト可能なコードスニペットの書き方
 - `sphinx` ディレクティブを文書化する
 - テストをグループ化する
- ローカルマシンでスニペットをテストする方法

PyQGIS-Developer-Cookbook の章を追加または更新することを計画している方は、コードスニペットの自動テストを可能にするためにいくつかの規則に従ってください。

テストはコードの自動チェックを可能にするので非常に重要です。エラーのあるコードスニペットや古いメソッドを使用するコードは失敗し、その通知は問題の解決に役立ちます。

テストのために、`Sphinx doctest` 拡張機能を使います。より詳細な情報については拡張機能のドキュメントを参照してください。

3.1 テスト可能なコードスニペットの書き方

テスト可能なコードスニペットを書くことは古い方法とそれほど変わりません。基本的には、異なる Sphinx のディレクティブを使う必要があります。

3.1.1 sphinx ディレクティブを文書化する

コードを `.. code-block:: python` ディレクティブに埋め込む代わりに（そうすると自動的にコード構文が強調表示される）、今度は `.. testcode::` に埋め込む必要があります。つまり、これの代わりに:

```
.. code-block:: python

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

今度はこれを使用します:

```
.. testcode::

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

サンプルコードを書いた後は、コードを評価して自動的に実行される アサーション を追加する必要があります。

上記の例では、crs を作成し、`assert crs.isValid()` でそれが有効かどうかをテストします。コードの Python 構文が間違っていたり `crs.isValid()` が `False` を返した場合、このコードスニペットはテスト中に失敗します。

スニペットでテストを正しく実行するには、すべてのクラスをインポートし、コードスニペットで使用されている変数を宣言する必要があります。それらはコードスニペット自体 (HTML ページに表示される) に含めることも、`.. testsetup ::` ディレクティブに追加することもできます (HTML ページには表示されません)。`.. testsetup::` は `.. testcode::` の前に置く必要があります:

```
.. testsetup::

    from qgis.core import QgsCoordinateReferenceSystem

.. testcode::

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

コードスニペットがオブジェクトを作成しない (したがって、`assert object.isValid()` のようなものが使用できない) 場合、`print()` メソッドを使用してコードをテストし、期待される結果を `.. testoutput ::` ディレクティブ内に追加して、期待される出力を比較できます:

```
.. testcode::

    print("QGIS CRS ID:", crs.srsid())
    print("PostGIS SRID:", crs.postgisSrid())

.. testoutput::

    QGIS CRS ID: 3452
    PostGIS SRID: 4326
```

デフォルトでは、`.. testoutput::` の内容は HTML 出力に表示されます。HTML から隠すには `:hide:` を使います:

```
.. testoutput::
    :hide:
```

(次のページに続く)

(前のページからの続き)

```
QGIS CRS ID: 3452
PostGIS SRID: 4326
```

注釈: コードスニペットに `print` 文が含まれている場合は、期待される出力とともに `testoutput` を追加しなければなりません。そうでなければテストは失敗します。

3.1.2 テストをグループ化する

各 `rst` 文書ごとに、コードスニペットは順番にテストされます。つまり、以下のすべてのコードスニペットに対して1つの `.. testsetup::` を使用でき、後のスニペットでは文書内の前のスニペットで宣言された変数にアクセスできます。

あるいは、グループを使用して、異なるテスト中の同じページの例を分類することもできます。

それぞれのディレクティブに1つ以上のグループ名を(コンマで区切って)追加して、コードスニペットをグループに追加します:

```
.. testcode:: crs_crsfromID [, morenames]

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

`doctest` は各グループのスニペットを選び、それらを独立して実行します。

注釈: 関連するコンテンツで意味のあるグループ名を使用してください。 `<chapter>_<subchapter>` のようなものを使用してください、例えば `crs_intro`、`crs_fromwkt`。障害が発生した場合は、障害が発生した場所を特定するのに役立ちます。

グループを宣言しないと、コードスニペットは `default` という名前のグループに追加されます。代わりに、グループ名として `*` を使用すると、スニペットはすべてのテストグループで使用されます。これは通常、テスト設定で使用すると便利です。

```
.. testsetup:: *

    from qgis.core import QgsCoordinateReferenceSystem
```

3.2 ローカルマシンでスニペットをテストする方法

注釈: 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. それから、それを使って target `doctest` を実行します。

```
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.

第 4 章

翻訳ガイドライン

- 翻訳のプロセス
- ファイルを翻訳する
 - *Transifex* での翻訳
 - *Qt Linguist* による翻訳
 - マニュアルを翻訳する
 - 翻訳ルールのまとめ

このマニュアルは翻訳者の助けとなるよう書かれました。最初に、翻訳の全体のプロセスが技術的にどのように進行するかを説明します。次に実際の翻訳がどのように行われるかを、英語の `.rst` ドキュメントからオランダ語への翻訳に例をとって説明します。最後に [翻訳ルールのまとめ](#) を見てもらいます。

注釈: このガイドラインでは QGIS ドキュメントに焦点を当てますが、下記の方法やルールは QGIS アプリケーションや QGIS ウェブサイトの翻訳にも適用可能です。

4.1 翻訳のプロセス

QGIS のドキュメントは `.rst` ファイルを使い英語で書かれています。翻訳は次のプロセスを経て提供されます。

1. プレビルドスクリプトによって、`.po` ファイルと呼ばれる翻訳用のファイルが、まず英語のために `/QGIS-Documentation/locale/en` フォルダの中に作成されます。
2. 次にスクリプトによって、この「オリジナル」が、その他の言語のために `locale` フォルダにコピーされます。
3. `.po` ファイルの中の文章が *Transifex* のウェブプラットフォームにプッシュされ、英語から各言語へ

と、翻訳者がエディタを使って翻訳できるようになります。

4. 1日の終わりに、すべての検証済みの翻訳がスクリプトによって取得され、元の .po ファイルに反映されます。
5. これらの翻訳は、ドキュメントの次のビルド（少なくとも1日に1回は行われます）の際に、翻訳版ドキュメントを作成するために、スクリプトによって再度利用されます。
6. 原文の .rst ドキュメントが更新されると、新しく英語の .po ファイルが作られます。この新しいファイルの内容は、各言語の既存の .po ファイルへとマージされます。言い換えると、翻訳済みの .rst ドキュメントに新たな文章が追加されても、翻訳済みの .po には新しい（あるいは修正された）文章だけが追加され、その部分だけを翻訳し直せばよいということです。このため、次のリリースのための翻訳の更新作業は、比較的少く済むはずでです。

注釈: 上記のプロセスは、QGIS ウェブサイト、QGIS デスクトップアプリケーション、QGIS サーバのいずれの翻訳でも、同じように行われます。ただしアプリケーションの翻訳では、.po ファイルではなく、.py ファイルや .cpp ファイル、.yaml ファイルなどのアプリケーションを形作るファイルの中のすべての翻訳可能な文字列が、単一の .ts ファイルとして transifex との間でやりとりされるという違いがあります。

現在、ふたつの異なるツールが、QGIS での翻訳を行うために使用されています。

- [Transifex のウェブプラットフォーム](#) は、QGIS を翻訳するのに一番簡単でお勧めの方法です。上で説明したプロセスは透過的に行われ、またすべての翻訳可能なテキストは1箇所に集められています。翻訳者は希望のファイルを選んで翻訳するだけです。別のリリースがブッシュされ翻訳を再開するまで、翻訳済みファイルはこのプラットフォームの中に保存されています。
- [Qt Linguist](#) は Qt の開発ツールです。翻訳者は .po (もしくは .ts) ファイルをソースコードからローカル環境に取得し、翻訳した後は送り返す必要があります。

どんなツールを選ぼうと、翻訳のルールは同じであることに注意してください。

4.2 ファイルを翻訳する

ヒートマッププラグインの例を使って翻訳の作業を説明します。この例では英語からオランダ語に翻訳を行いますが、他のドキュメントや言語でもほぼ同じでしょう。

ヒートマッププラグインのドキュメントのソースは以下の場所にあります。

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

なんでこのドキュメントを例として選んだのかですって？

1. このドキュメントには画像、キャプション、ヘッダー、参照、置換があるからです
2. このドキュメントを書いたのは私なので、私には翻訳が容易だからです (^o^)

ビルドプロセスが英語の .po ファイルを以下の場所に生成しています。

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

これに対応するオランダ語の .po ファイル（基本的にコピーです）は以下の場所にあります。

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

このファイルと並んでごく小さなサイズの .mo ファイルがあるのに気づかれると思いますが、今の所このファイルにはなんの翻訳も含まれていません。

4.2.1 Transifex での翻訳

Transifex を使って翻訳するには、以下に従ってください。

1. Transifex でアカウントを作成し、QGIS プロジェクトに参加します。
2. Languages チームの一員になったら、該当するプロジェクト（ここでは QGIS Documentation です）をクリックします。翻訳が可能な言語の一覧が、その進捗の割合とともに表示されます。

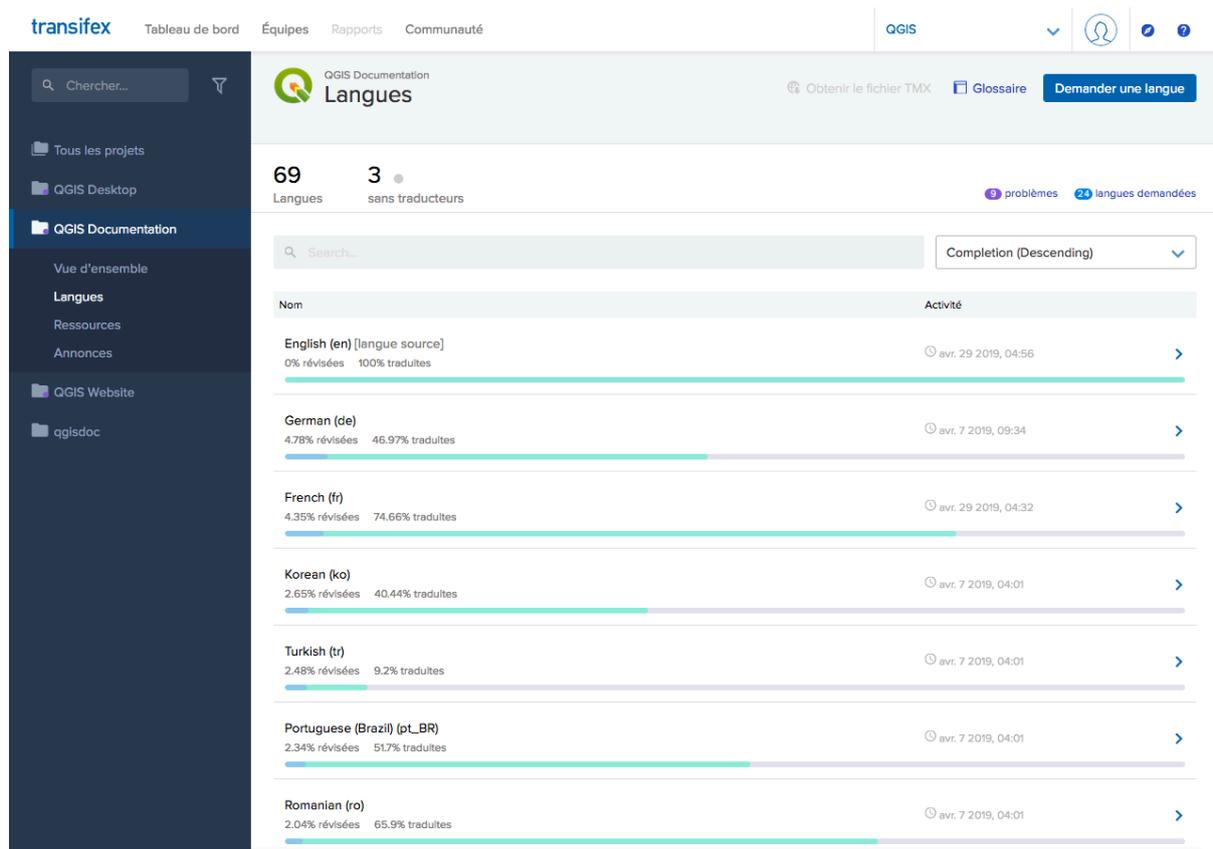


図 4.1 Transifex のメニューで翻訳する言語を選んでください

3. 翻訳する言語の上にカーソルを持ってきて、次のいずれかをクリックしてください。

- リソースを表示: 翻訳可能な .po ファイルが、その翻訳の割合と文字列数とその他のメタデータと

ともに表示されます。

- 翻訳: 翻訳可能なすべての .po ファイルと翻訳インターフェースが開きます。
4. 翻訳したいファイル (この例では docs_user-manual_plugins_plugins-heatmap という名前の heatmap plugin のファイルを探します) が、もしくはなんでもいいので翻訳の終わっていないファイルを見つけて、それをクリックしてください。するとそのファイル中の文字列が読み込まれ、絞り込みや翻訳、提案などのインターフェースを利用できるようになります。

ちなみに: ドキュメントとウェブサイトの翻訳では、ページのフッターにある [このページを修正](#) のリンクをクリックすれば、Transifex の対応する翻訳ページに直接移動することができます。

- 5.あとは個々のテキストを選んで、下記の *guidelines* に従って翻訳するだけです。

Transifex の Web エディターについてのより詳しい情報は、<https://docs.transifex.com/translation/translating-with-the-web-editor> を参照してください。

4.2.2 Qt Linguist による翻訳

Qt Linguist を使って翻訳するには、以下に従ってください。

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. ローカルで翻訳に取りかかります。
3. 修正を行ったファイルを Transifex か GitHub の元の場所へアップロードします。

翻訳可能なファイルを Transifex を使ってダウンロードしたりアップロードしたりすることは可能ですが、この手順はお勧めできません。というのも、Transifex にはバージョン管理システムがないために、あなたがアップロードしたファイルは単純に既存のファイルを置き換えるだけなので、その間に Transifex で他の人によって行われた変更を上書きしてしまう可能性があるからです。

Qt Linguist で初めてそのファイルを開くときは、次のようなダイアログが表示されます。

ターゲット言語は正しく選択する必要があります。ソース言語は Language は POSIX、Country/Region は Any Country のままでかまいません。

OK ボタンを押すと、Qt Linguist に文章が読み込まれ、翻訳を開始することが可能になります。 [Figure_translation_menu](#) を参照してください。

メニューには使うと便利な以下のようなボタンがあります。

-  [翻訳を完了して次へ] ボタンは、最も重要なボタンです。翻訳が必要な項目では、テキストフィールドに翻訳を入力してこのボタンを押してください。翻訳を必要としない項目では、テキストフィールドは空のままこのボタンを押してください。これで翻訳完了となり、次の項目に進みます。
-  [前へ行く] ボタンで、その前の翻訳項目に移動することができます。



図 4.2 linguist メニューから翻訳する言語を選んでください

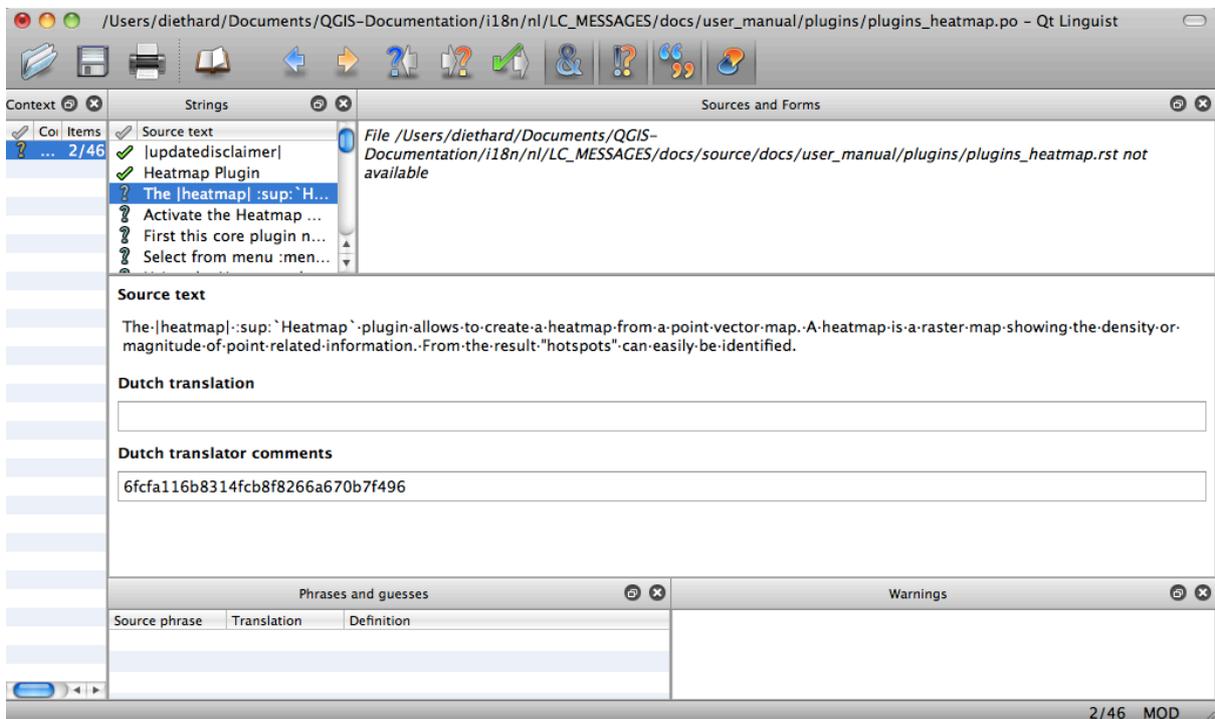


図 4.3 Qt Linguist のメニューを使用して翻訳

-  [次へ行く] ボタンで、その次の翻訳項目に移動することができます。
-  [次のすべき] ボタンは、翻訳が必要な次の翻訳項目にジャンプします。ソースファイルの変更によって追加や修正が行われた文だけを翻訳する必要があるときに便利です。
-  [前のすべき] ボタンは逆方向に検索を行い、最初に見つかった未翻訳の項目にジャンプします。

Qt Linguist の使用についてのより詳しい情報は、 <https://doc-snapshots.qt.io/qt5-5.12/linguist-translators.html> を参照してください。

警告: 翻訳のためにソースリポジトリからファイルをダウンロードするときは、絶対に `master` ブランチで行なってはなりません。ある特定のバージョンで英語での更新が完了した後は、常に翻訳用ブランチが有効になっています。翻訳のためにはそのブランチを使用してください。例えば QGIS 2.8 の翻訳では、`manual_en_v2.8` ブランチを使用してください。

4.2.3 マニュアルを翻訳する

それでは `plugin_heatmap` マニュアルの翻訳を始めましょう！

センテンスのほとんどはそのまま翻訳するべきです。特別な翻訳をしなければならない部分 (rst 文) を、この翻訳セッションの中で挙げていきます。

以下で翻訳の対象として興味深いセンテンスを見ていきましょう。

```
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.
```

このセンテンスには 2 つの rst 構文があります。

- `|heatmap|` のように `|` の間にある言葉は置換されるので決して翻訳してはいけません。ここではヒートマッププラグインのアイコンに置換されます。
- `:sup:`Heatmap`` のような `:sup:` 文は上付き文です。続くテキストを少し高く出力します。これはツールバー項目の上にマウスを当てた時に表示されるポップアップテキストを表示するために使用されますが、実際の QGIS アプリケーションでの翻訳とは異なる場合があります。オランダ語の場合はそうではありません。

このセンテンス中のそれ以外のプレーン・テキストはすべて翻訳してかまいません。

次の文章には、通常マニュアルの別のセクションを参照するために使用される `:ref:` 文が含まれています。`:ref:` 文に続くテキストは一意的識別子であるため、変更してはいけません。

```
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. `ref` 文は見出しテキストに置換され、ハイパーリンクになります。参照されている見出しが翻訳された時は、この見出しを参照しているすべての `ref` 文も自動的に同じように翻訳されます。

次の文章には `rst`-タグ `:menuselection:` があります。後ろに続くテキストは実際に QGIS アプリケーションのメニューで表示されるテキストです。これはアプリケーションでは翻訳されているかもしれませんが、その場合は変更すべきです。

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

ここではオランダ語へのローカライズ QGIS アプリケーションで使用される翻訳ですので、上記の「View -->」は実際には「Beeld -->」と翻訳されます。

次のように、もう少しトリッキーな文章のこともあります。

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

この文章には、図 `figure_heatmap_settings_` への参照があります。セクションへの参照と同様に、この参照を変更してはいけません。`rst` ドキュメントでの参照定義は `.po` ファイルには含まれていないため、これを変更することはできません。このことは図への参照は翻訳できないことを意味します。HTML 生成時には `figure_heatmap_settings` と表示されます。PDF 生成の際には `figure_heatmap_settings_` は図番号に置き換えられます。

次の `rst` 属性付きの文章はこのようなものです。

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

上の行内の二重星印を削除しないでください。二重星印に挟まれたテキストは太字で印刷されます。テキストそのものは多くの場合、ダイアログ中のテキストであり、アプリケーションでは翻訳されているかもしれませんが。

以下の文章には `:guilabel:` `rst` タグが含まれています。

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

`guilabel` タグのテキスト `Advanced` は QGIS アプリケーション中でおそらく翻訳されるのでたぶん変更する必要があります！

以下の文章には ``airports`` が含まれています。引用符は、テキストに別のテキストフォントを与えるために使用されています。このケースでは、リテラル値であり、翻訳を必要としません。

```
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>`_.
```

この文章には、URL と表示テキストの両方を持つハイパーリンクも含まれています。URL はもちろん手をつけずそのまま残すべきですが、読者に表示される方のテキストである `https://www.qgistutorials.com` は変更することができます。ハイパーリンクの末尾のアンダースコアは本質的な部分なので決して削除してはいけません。

4.2.4 翻訳ルールのまとめ

1. `|bronze|`、`|checkbox|`、`|labels|`、`|selectString|`、`|addLayer|` ... のような、2 つの文字 | の間のテキストを変更しないでください。これらは画像に置換するために使用される特殊なタグです。
2. `:ref:` や `:file:` タグで始まる参照を変更しないでください。
3. `figure_labels_1_` のようなアンダースコアで終わる参照を変更しないでください。
4. ハイパーリンクの URL を変更しないでください。ただし読者に表示される記述は変更できます。ハイパーリンクの末尾のアンダースコア (`>`_`) は空白を加えたりせずそのまま残しておいてください。
5. `:index:`、`:sup:`、`:guilabel:`、`:menuselection:` タグに続く引用符内のテキストは変更してください。それが QGIS アプリケーションで翻訳されているかどうか (翻訳されていたらどう翻訳されているか) を確認してください。タグ自体は変更しないでください。
6. 二重星と二重引用符の間のテキストは多くの場合、値もしくはフィールド名を示しています。翻訳の必要があることもあれば、ないこともあります。
7. ```、````、`*`、`**`、`::` などの特殊文字は、正確に元のテキストと同じ文字を同じ数だけ使用するように注意してください。これらは情報の見た目を整えるのに貢献しています。
8. 特殊文字やタグで囲まれたテキストの前後にスペースを入れないでください。
9. 翻訳文字列の後に段落を開始して終了しないでください。そうすると HTML 生成時にテキストが翻訳されません。

以上のルールを守ってドキュメントの品質を良くしましょう！

ご質問がございましたら、[QGIS コミュニティチーム](#) または [QGIS 翻訳チーム](#) までご連絡ください。

第 5 章

置換参照と定義

- 利用方法
- 一般的な置換
 - プラットフォームアイコン
 - メニューアイテム
- ツールバーボタンアイコン
 - レイヤー管理と全体図
 - ファイル
 - 編集
 - 識別結果
 - デジタイズと高度なデジタイジング
 - 地図ナビゲーションと属性
 - 選択と式
 - ラベルとダイヤグラム
 - 整飾
 - ヘルプ
 - 色
- 他の基本的アイコン
- 属性テーブル
- 投影とジオリファレンサ
- 印刷レイアウト

- レイヤプロパティ
- プラグイン
 - プロセッシング
 - コア・プラグインのいろいろ
 - *Grass* 統合
 - *eVis* プラグイン

5.1 利用方法

QGIS マニュアルでアイコンを手軽に使えるように、アイコンの置換が [QGIS-Documentation](#) リポジトリの `/source/substitutions.txt` ファイルで定義されています。そこで定義されている置換のいくつかを以下に示します。このためドキュメント中で QGIS アプリケーションのアイコンを使用したい場合は、使用できる、あるいは使用すべき置換がすでにある可能性が大きいです。

置換が存在しない場合は、以下のようにして追加することができます。

1. 目的のアイコン画像があるかどうか、ドキュメンテーションリポジトリの `/static/common` フォルダ内を確認してください。画像がない場合は、[QGIS](#) リポジトリからアイコン画像ファイルを探して (https://github.com/qgis/QGIS/blob/release-3_10/images/themes/default フォルダにあることが多いです) `/static/common` フォルダに `.png` フォーマットでコピーします。利便性と後の更新のことを考えると、ファイル名は可能な限り元のままにしておくことを勧めます。
2. 以下の例に従って、置換への参照を `/source/substitutions.txt` ファイルに作成します。置換テキストはキャメルケースにする必要があります。

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

3. `scripts/find_set_subst.py` スクリプトを実行して `rst` ファイル中の置換定義を更新し、新しい置換が含まれるようにしてください。
4. (オプション) 以下のリストに、アイコンおよびその置換への参照を追加します。

5.2 一般的な置換

以下はドキュメントを書くときに使われるアイコンとその置換です。マニュアルの多くの場所で使用されているのを見ることができるでしょう。

5.2.1 プラットフォームアイコン

アイコン	置き換え	アイコン	置き換え
	logo		
	kde		nix
	osx		win

5.2.2 メニューアイテム

アイコン	置き換え	アイコン	置き換え
	checkbox		unchecked
	radioButtonOn		radioButtonOff
	selectNumber		selectString
	selectColor		selectColorRamp
	tab		degrees
	inputText		slider

5.3 ツールバーボタンアイコン

5.3.1 レイヤー管理と全体図

アイコン	置き換え	アイコン	置き換え
	dataSourceManager		
	addOgrLayer		addDb2Layer
	addRasterLayer		addMssqlLayer
	addDelimitedTextLayer		addSpatialiteLayer
	addPostgisLayer		addOracleLayer
	addAfsLayer		addAmsLayer
	addMeshLayer		
	addVirtualLayer		addWmsLayer
	addWcsLayer		addWfsLayer
	newVectorLayer		newSpatialiteLayer
	newGeoPackageLayer		createMemory
	newVirtualLayer		
	dbManager		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 ファイル

アイコン	置き換え	アイコン	置き換え
	fileNew		fileOpen
	fileSave		fileSaveAs
	fileExit		

5.3.3 編集

アイコン	置き換え	アイコン	置き換え
	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 識別結果

アイコン	置き換え	アイコン	置き換え
	expandTree		collapseTree
	expandNewTree		formView
	deselectAll		editCopy
	filePrint		
	identifyByRectangle		identifyByPolygon
	identifyByFreehand		identifyByRadius

5.3.5 デジタイズと高度なデジタイジング

アイコン	置き換え	アイコン	置き換え
	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 地図ナビゲーションと属性

アイコン	置き換え	アイコン	置き換え
	pan		panToSelected
	zoomIn		zoomOut
	zoomActual		zoomFullExtent
	zoomToLayer		zoomToSelected
	zoomLast		zoomNext
	refresh		
	identify		mapTips
	showBookmarks		newBookmark
	measure		measureArea
	measureAngle		
	newMap		new3DMap
	tiltUp		tiltDown
	3dNavigation		play

5.3.7 選択と式

アイコン	置き換え	アイコン	置き換え
	selectRectangle		selectPolygon
	selectFreehand		selectRadius
	selectAll		deselectAll
	invertSelection		expressionSelect
	selectAllTree		select
	formSelect		dataDefined
	expression		dataDefineOn
	dataDefineExpressionOn		dataDefineError
	dataDefineExpressionError		
	addExpression		
	expressionFilter		filterMap

5.3.8 ラベルとダイアグラム

アイコン	置き換え	アイコン	置き換え
	labeling		labelingNone
	labelingRuleBased		labelingObstacle
	piechart		diagramNone
abc	text		histogram
	annotation		textAnnotation
	formAnnotation		htmlAnnotation
	svgAnnotation		autoPlacement
	labelbackground	abc	labelbuffer
+ab < c	labelformatting		labelplacement
	labelshadow		render
abc	labelcallout		
	pinLabels		showHideLabels
	moveLabel		rotateLabel
	showPinnedLabels		showUnplacedLabel
	changeLabelProperties		

5.3.9 整飾

アイコン	置き換え	アイコン	置き換え
	copyrightLabel		addGrid
	titleLabel		northArrow
	scaleBar		addMap
	addImage		

5.3.10 ヘルプ

アイコン	置き換え	アイコン	置き換え
	helpContents		qgisHomePage
	success		
	helpSponsors		contextHelp

5.3.11 色

アイコン	置き換え	アイコン	置き換え
	colorBox		colorPicker
	colorSwatches		colorWheel

5.4 他の基本的アイコン

アイコン	置き換え	アイコン	置き換え
	arrowDown		arrowUp
	signPlus		signMinus
	projectProperties		options
	interfaceCustomization		keyboardShortcuts
	copyrightLabel		northArrow
	scaleBar		tracking
	gpsImporter		gpsTrackBarChart
	folder		extents
	settings		start
	3dconfigure		deleteSelected
	browserExpand		browserCollapse

5.5 属性テーブル

アイコン	置き換え	アイコン	置き換え
	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 投影とジオリファレンサ

アイコン	置き換え	アイコン	置き換え
	geographic		crs
	customProjection		setProjection
	projectionDisabled		projectionEnabled
	georefRun		pencil
	linkQgisToGeoref		linkGeorefToQgis
	coordinateCapture		fullHistogramStretch

5.7 印刷レイアウト

アイコン	置き換え	アイコン	置き換え
	newLayout		layoutManager
	duplicateLayout		
	newReport		newPage

次のページに続く

表 5.1 – 前のページからの続き

アイコン	置き換え	アイコン	置き換え
	atlasSettings		atlas
	filePrint		saveMapAsImage
	saveAsSVG		saveAsPDF
	addBasicShape		addBasicCircle
	addBasicTriangle		addBasicRectangle
	addNodesShape		editNodesShape
	addPolygon		addPolyline
	addArrow		northArrow
	add3DMap		addMap
	addLegend		addHtml
	addImage		addTable
	label		scaleBar
	select		moveItemContent
	setToCanvasScale		setToCanvasExtent
	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 レイヤプロパティ

アイコン	置き換え	アイコン	置き換え
	symbology		labeling
	sourceFields		general
	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		
	sum		sort
	paintEffects		mapIdentification
	styleManager		iconView
	joinNotEditable		joinedLayerNotEditable
	joinHasNotUpsertOnEdit		filterTableFields
	symbologyEdit		

5.9 プラグイン

5.9.1 プロセッシング

アイコン	置き換え	アイコン	置き換え
	processing		processingModel
	processingHistory		processingResult
	processSelected		
	qgsProjectFile		addToProject
	mean		layerExtent
	randomSelection		vectorGrid
	convexHull		buffer
	intersect		union
	symDifference		clip
	difference		dissolve
	checkGeometry		exportGeometry
	delanay		centroids
	toLines		extractNodes
	splitLayer		
	showRasterCalculator		heatmap
	showMeshCalculator		

5.9.2 コア・プラグインのいろいろ

基本インストールで標準で提供されます。しかし初期状態ではロードはされていません。

アイコン	置き換え	アイコン	置き換え
	showPluginManager		installPluginFromZip
	pythonFile		runConsole
	showEditorConsole		clearConsole
	offlineEditingCopy		offlineEditingSync
	plugin		interpolation
	gdalScript		metasearch
	geometryChecker		topologyChecker
	fromSelectedFeature		sqlQueryBuilder

5.9.3 Grass 統合

アイコン	置き換え	アイコン	置き換え
	grass		grassRegion
	grassTools		grassNewMapset
	grassOpenMapset		grassCloseMapset

5.9.4 eVis プラグイン

アイコン	置き換え	アイコン	置き換え
	eventBrowser		eventId
	evisConnect		