



QGIS Documentation Guidelines

QGIS Project

20 mag 2025

1	Una contribuzione passo dopo passo	3
1.1	Utilizzo dell'interfaccia web di GitHub	4
1.1.1	Segnala QGIS-Documentation	4
1.1.2	Apportare modifiche	5
1.1.3	Modificare file	6
1.1.4	Condividi le tue modifiche via Pull Request	6
1.1.5	Eliminare il tuo ramo di fusione	10
1.2	Usare gli strumenti a riga di comando di Git	10
1.2.1	Repository locale	10
1.2.2	Aggiungere un altro repository remoto	11
1.2.3	Aggiornare il tuo ramo base	12
1.2.4	Contribuire al tuo ramo di produzione	12
1.2.5	Condividere i tuoi cambiamenti	13
1.2.6	Pulire il tuo repository locale e remoto	13
1.3	Ulteriori letture	13
2	Writing Guidelines	15
2.1	Writing Documentation	16
2.1.1	Headlines	16
2.1.2	Lists	16
2.1.3	Indentation	16
2.1.4	Inline Tags	17
2.1.5	Labels/references	17
2.1.6	Figures and Images	18
2.1.7	Indice	21
2.1.8	Commenti speciali	22
2.1.9	Frammenti di codice	22
2.1.10	Note a piè di pagina	22
2.2	Gestione delle schermate	22
2.2.1	Aggiungere nuove schermate	22
2.2.2	Screenshot tradotti	23
2.3	Documentare gli algoritmi di Processing	23
3	Scrivere codice nel PyQGIS Cookbook	29
3.1	Come scrivere parti di codice testabili	29
3.1.1	Direttive Doctest sphinx	29
3.1.2	Raggruppare i test	31
3.2	Come testare pezzi di codice sulla tua macchina locale	31
4	Linee guida per la traduzione	33
4.1	Processo di traduzione	33

4.2	Traduci un file	34
4.2.1	Traduzione in Transifex	35
4.2.2	Traduzione in QT Linguist	36
4.2.3	Traducurre il manuale	38
4.2.4	Sommario delle regole per la traduzione	39
5	Sostituzioni	41
5.1	Uso	42
5.2	Sostituzioni comuni	42
5.2.1	Icone di piattaforma	42
5.2.2	Oggetti del menu	43
5.3	Icone pulsanti barra degli strumenti	43
5.3.1	Gestione layer e panoramica	43
5.3.2	Progetto	44
5.3.3	Modifica	45
5.3.4	Identity result	45
5.3.5	Digitalizzazione e digitalizzazione avanzata	45
5.3.6	Mesh	46
5.3.7	Orientazione della mappa e attributi	47
5.3.8	Selezione ed Espressioni	47
5.3.9	Etichette e Diagrammi	48
5.3.10	Decorazioni	48
5.3.11	Guida	49
5.3.12	Colori	49
5.4	Altre icone di base	49
5.5	Tabella degli attributi	50
5.6	Proiezioni e Georeferenziatore	50
5.7	Layout di stampa	51
5.8	Proprietà Layer	52
5.9	Plugin	53
5.9.1	Processing	53
5.9.2	Varie Plugin di Base	54
5.9.3	Integrazione con GRASS	55

La documentazione di QGIS è disponibile all'indirizzo <https://docs.qgis.org>. Mentre il processo di scrittura è in corso, ogni giorno viene eseguita automaticamente una compilazione (vedere in fondo alla pagina per l'ora esatta) per tutte le [versioni supportate](#) (testing, Long Term Release (LTR) e prossima LTR).

I file sorgente della Documentazione QGIS sono disponibili all'indirizzo <https://github.com/qgis/QGIS-Documentation>. Sono scritti principalmente utilizzando la sintassi del formato reStructuredText (reST), abbinata ad alcuni script del set di strumenti Sphinx per post-processare l'output HTML. Per informazioni generali su questi strumenti, vedere <https://docutils.sourceforge.io/docs/ref/rst/restructuredtext.html> o <https://www.sphinx-doc.org/en/master/usage/restructuredtext/basics.html>.

I seguenti capitoli ti guideranno per imparare:

- come gestire i file sorgente della documentazione usando [git](#) sistema e su [GitHub](#) piattaforma in cui sono memorizzate
- come modificare testi, fornire screenshot... in modo conforme
- come condividere e garantire che le tue modifiche siano inserite nei documenti ufficiali.

Se stai cercando le informazioni generali su come contribuire al progetto QGIS, puoi trovare aiuto in [Get Involved in the QGIS Community](#).

Una contribuzione passo dopo passo

- *Utilizzo dell'interfaccia web di GitHub*
 - *Segnala QGIS-Documentation*
 - *Apportare modifiche*
 - * *Alternativa 1: usa la scorciatoia Modifica su GitHub*
 - * *Alternativa 2: creare un ramo ad hoc nel tuo repository di documentazione*
 - *Modificare file*
 - *Condividi le tue modifiche via Pull Request*
 - * *Inizia una nuova richiesta di pull*
 - * *Confronta le modifiche*
 - * *Descrivi la tua richiesta di pull*
 - * *Rivedere e commentare la richiesta di pull*
 - * *Apportare correzioni*
 - *Eliminare il tuo ramo di fusione*
- *Usare gli strumenti a riga di comando di Git*
 - *Repository locale*
 - *Aggiungere un altro repository remoto*
 - *Aggiornare il tuo ramo base*
 - *Contribuire al tuo ramo di produzione*
 - *Condividere i tuoi cambiamenti*
 - *Pulire il tuo repository locale e remoto*
- *Ulteriori letture*

Nota: Sebbene la documentazione QGIS sia utilizzata per illustrare il processo, tutti i comandi e i passaggi illustrati

di seguito si applicano anche al sito Web QGIS.

Se stai leggendo queste righe è certamente perché sei disposto a contribuire a scrivere la documentazione di QGIS e stai cercando un modo per farlo. Sei venuti nel posto giusto, il presente documento ti guiderà attraverso i diversi modi per raggiungere questo obiettivo, mostrandoti i principali passi da seguire, i trucchi che puoi usare e le trappole a cui porre attenzione.

Per qualsiasi aiuto non esitare a chiedere postando un commento sul problema che stai cercando di risolvere o scrivendo alla **lista QGIS-community-team** <<https://lists.osgeo.org/mailman/listinfo/qgis-community-team>>`. Maggiori dettagli su [Partecipa alla documentazione](#).

Buttiamoci ora nel processo.

I sorgenti della documentazione sono memorizzati usando il sistema di controllo di versione git e sono disponibili su GitHub a <https://github.com/qgis/QGIS-Documentation>. Una lista di problemi da risolvere e di funzioni da spiegare può essere trovata su <https://github.com/qgis/QGIS-Documentation/issues>.

Suggerimento: Se sei un contributore alle prime armi e non sai da dove cominciare, potresti essere interessato ad affrontare i nostri «rapporti di benvenuto» <<https://github.com/qgis/QGIS-Documentation/contribute>>`.

Ci sono due modi principali, che non si escludono a vicenda, per modificare i file:

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

1.1 Utilizzo dell'interfaccia web di GitHub

L'interfaccia web di GitHub permette di fare quanto segue:

- modifica file
- vedere l'anteprima e fare il commit delle tue modifiche
- effettuare una richiesta di pull per l'inserimento delle modifiche nel repository principale
- creare, aggiornare o eliminare rami

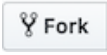
Se non si ha ancora familiarità con git e con il vocabolario di GitHub, si consiglia di leggere il progetto GitHub [Hello-world](#) per imparare alcuni vocaboli e azioni di base che saranno usati di seguito.

Nota: Se stai risolvendo un problema segnalato

Se stai facendo delle modifiche per risolvere un [issue](#), aggiungi un commento al report del problema per assegnarlo a te stesso. Questo impedirà a più di una persona di lavorare sullo stesso problema.

1.1.1 Segnala QGIS-Documentation

Supponendo che tu abbia già un account GitHub <<https://github.com/join>>`, devi prima fare una duplicazione dei file sorgente della documentazione.

Naviga alla pagina [QGIS-Documentation repository](#) e clicca sul pulsante  in alto a destra.

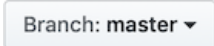
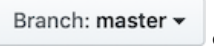
Nel tuo account GitHub troverai un repository QGIS-Documentation (<https://github.com/1/QGIS-Documentation>). Questo repository è una copia del repository ufficiale di QGIS-Documentation dove hai pieno accesso in scrittura e puoi fare modifiche senza influenzare la documentazione ufficiale.

1.1.2 Apportare modifiche

Ci sono diversi modi per contribuire alla documentazione di QGIS. Li mostriamo separatamente qui sotto, ma puoi passare da un processo all'altro senza alcun problema.

Alternativa 1: usa la scorciatoia **Modifica su GitHub**

Le pagine sul sito web della documentazione di QGIS possono essere modificate rapidamente e facilmente cliccando sul link **Modifica su GitHub** in alto a destra di ogni pagina.

1. Questo aprirà il file nel ramo `qgis:master` con un messaggio in cima alla pagina che ti dice che non hai accesso in scrittura a questo repo e le tue modifiche saranno applicate a un nuovo ramo del tuo repository.
2. Fai le tue modifiche. Poiché la documentazione è scritta usando la sintassi `reStructureText`, a seconda delle tue modifiche, potresti aver bisogno di fare riferimento alle [writing guidelines 1](#).
3. Al termine, fai un breve commento sulle modifiche e clicca su *Proponi modifiche*. Questo genererà un nuovo **branch** (`patch-xxx`) nel tuo repository.
4. Dopo aver cliccato su *Propose changes*, github passerà alla pagina *Comparing changes*.
 - Se hai finito di apportare modifiche, passa a *Compare changes* nella sezione *Share your changes via Pull Request* qui sotto.
 - Se ci sono ulteriori modifiche che vuoi fare prima di inviarle a QGIS, segui questi passi:
 1. Vai al tuo ramo di `QGIS-Documentation` (<https://github.com/<YourName>/QGIS-Documentation>)
 2. Clicca su  e cerca il ramo `patch-xxx`. Seleziona questo ramo della patch. Il pulsante  ora dirà *Branch: patch-xxx`*.
 3. Vai a *Modify files* qui sotto.

Nota: La scorciatoia **Edit on GitHub** è anche disponibile nel menu a discesa in fondo alla barra laterale sinistra.

Alternativa 2: creare un ramo ad hoc nel tuo repository di documentazione

Puoi modificare i file direttamente dal tuo ramo della Documentazione QGIS.

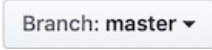
Per prima cosa, assicurati che il tuo ramo `master` sia aggiornato con il ramo `qgis:master`. Per fare ciò:

1. Vai alla pagina principale del tuo repository, ad esempio <https://github.com/<YourName>/QGIS-Documentation>. Il ramo `master` dovrebbe essere attivo con una indicazione se è aggiornato con `qgis/QGIS-Documentation:master` o no.

Se ha dei commit in precedenza rispetto al ramo upstream, è meglio usare il precedente *shortcut button alternative* finché non allinei il tuo ramo `master`.

Se ha solo dei commit arretrati:

1. Espandi il menu a discesa *Fetch Upstream* sulla destra. Puoi
 - *Compare* i rami e vedi le nuove modifiche nel repository principale
 - *Fetch and merge*: trasferisce i cambiamenti dal ramo upstream al tuo.
2. Facciamo clic su *Fetch and merge*: dopo il processo, il tuo ramo viene indicato come aggiornato con `qgis/QGIS-Documentation:master`.

2. Clicca su  nell'angolo in alto a sinistra del repository di QGIS-Documentation e inserisci un nome univoco nel campo di testo per creare un nuovo [branch](#). Il nome del nuovo ramo dovrebbe essere correlato al problema che intendi risolvere. Il pulsante  dovrebbe ora riportare *Branch: branch_name*.
3. Sei pronto a iniziare nuovi cambiamenti su di esso.

Attenzione: Fai le tue modifiche in un ramo ad hoc, mai nel ****branch master**.

Per convenzione, evita di fare cambiamenti nel tuo ramo `master` tranne quando unisci le modifiche dal ramo `master` di `qgis/QGIS-Documentation` nella tua copia del repository `QGIS-Documentation`. I rami separati ti permettono di lavorare su più problemi allo stesso tempo senza interferire con altri rami. Se fai un errore puoi sempre cancellare un ramo e ricominciare creandone uno nuovo dal ramo `master`.

1.1.3 Modificare file

1. Sfoglia i file sorgente del tuo ramo della QGIS-Documentation fino al file che deve essere modificato
2. Fai le tue modifiche seguendo le *writing guidelines*
3. Quando hai finito, vai al riquadro **Commit Changes** in fondo alla pagina, fai un breve commento sulle tue modifiche e clicca su *Commit Changes* per impegnare le modifiche direttamente nel tuo ramo. Assicurati che *Commit directly to the branch_name branch* sia selezionato.
4. Ripeti i passi precedenti per qualsiasi altro file che deve essere aggiornato per correggere il problema

1.1.4 Condividi le tue modifiche via Pull Request

Devi fare una richiesta di pull per integrare le tue modifiche nella documentazione ufficiale.

Nota: Se hai usato una [Modifica su GitHub link](#)

Dopo aver fatto il commit delle tue modifiche GitHub aprirà automaticamente una nuova pagina confrontando le modifiche che hai fatto nel tuo ramo `patch-xxx` con il ramo `master` di `qgis/QGIS-Documentation`.

Passa a [Step 2](#) qui sotto.

Inizia una nuova richiesta di pull

Vai alla pagina principale del repository [QGIS-Documentation](#) e clicca su *New pull request*.

Confronta le modifiche

Se vedi due finestre di dialogo, una che dice `base:master` e l'altra `compare:branch_name` (vedi figura), questo fonderà solo le tue modifiche da uno dei tuoi rami al tuo ramo `master`. Per risolvere questo problema clicca sul link *compare across forks*.

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: master ← compare: patch-1 ✓ **Able to merge.** These branches can be automatically merged.

Fig. 1.1: Se la tua pagina *Comparing changes* assomiglia a questa, clicca sul link *compare across forks*.

Dovresti vedere quattro menu a tendina. Questi ti permetteranno di confrontare i cambiamenti che hai fatto nel tuo ramo con il ramo master in cui vuoi fonderti. Essi sono:

- **base fork:** il ramo in cui vuoi fondere le tue modifiche
- **base:** il ramo della diramazione di base in cui vuoi fondere le tue modifiche
- **head fork:** il ramo che ha modifiche che si vogliono incorporare nel ramo base
- **compare:** il ramo con tali modifiche

Seleziona `qgis/QGIS-Documentation` come ramo base con `master` come base, imposta la diramazione principale al tuo repository `<YourName>/QGIS-Documentation`, e imposta `compare` al tuo ramo modificato.

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.

Fig. 1.2: Comparare le modifiche tra `qgis/QGIS-Documentation` e il tuo repository

Una spunta verde con le parole **Able to merge** mostra che le tue modifiche possono essere unite alla documentazione ufficiale senza conflitti.

Clicca sul pulsante *Create pull request*.

Avvertimento: Se vedi **✗ Can't automatically merge.**

Questo significa che ci sono **conflicts**. I file che stai modificando non sono aggiornati con il ramo a cui stai puntando, perché qualcun altro ha fatto un commit che è in conflitto con le tue modifiche. Puoi ancora creare la richiesta di pull, ma devi correggere tutti i **conflicts** per completare la fusione.

Suggerimento: Sebbene sia stata tradotta, la **latest version** della documentazione di QGIS è ancora mantenuta e i problemi esistenti sono risolti. Se stai correggendo problemi per una release diversa, cambia **base** da `master` al ramo appropriato `release_...` nei passi di cui sopra.

Descrivi la tua richiesta di pull

Si aprirà una casella di testo: inserisci qualsiasi commento pertinente al problema che stai affrontando.

Se questo si riferisce ad un particolare issue <<https://github.com/qgis/QGIS-Documentation/issues>>, aggiungi il numero del problema ai tuoi commenti. Questo viene fatto inserendo # e il numero del problema (e.g. #1234). Se preceduto da termini come `fix` o `close`, il problema in questione verrà chiuso non appena la richiesta di pull verrà inserita.

Aggiungi i link alle pagine di documentazione che stai cambiando.

Clicca su *Create pull request*.

Rivedere e commentare la richiesta di pull

Come visto in precedenza, chiunque può inviare modifiche alla documentazione attraverso le richieste di pull. Allo stesso modo chiunque può rivedere le richieste di pull con domande e `comments`. Forse lo stile di scrittura non corrisponde alle linee guida del progetto, la modifica manca di alcuni dettagli importanti o di schermate, o forse tutto sembra perfetto e in ordine. La revisione aiuta a migliorare la qualità del contributo, sia nella forma che nella sostanza.

Per rivedere una richiesta di pull:

1. Vai alla pagina delle richieste di pull <<https://github.com/qgis/QGIS-Documentation/pulls>> e clicca sulla richiesta di pull su cui vuoi fare un commento.
2. In fondo alla pagina troverai una casella di testo dove puoi lasciare commenti generali sulla richiesta di pull.
3. Per aggiungere commenti su linee specifiche,

1. Clicca su  **Files changed** e trova il file che vuoi commentare. Potresti dover cliccare su *Display the source diff* per vedere le modifiche.
2. Scorri fino alla linea che vuoi commentare e clicca su . Questo aprirà una casella di testo che ti permetterà di lasciare un commento.

Commenti specifici sulla linea possono essere pubblicati:

- come commenti singoli, usando il pulsante *Add single comment*. Vengono pubblicati man mano. Usalo solo se hai pochi commenti da aggiungere o quando replichi su un altro commento.
- o come parte di una revisione, premendo il pulsante *Avvia una revisione*. I commenti non vengono inviati automaticamente dopo la convalida, consentendo di modificarli o cancellarli in seguito, di aggiungere un riassunto dei punti principali della revisione o istruzioni globali riguardanti la richiesta di pull e se approvarla o meno. Questo è il modo più opportuno, perché è più flessibile e consente di strutturare la revisione, modificare i commenti, pubblicare quando si è pronti e inviare un'unica notifica ai follower del repository e non una notifica per ogni commento. Maggiori dettagli <<https://docs.github.com/en/pull-requests/collaborating-with-pull-requests/reviewing-changes-in-pull-requests/reviewing-proposed-changes-in-a-pull-request>>.



Fig. 1.3: Commentare una linea con un suggerimento di modifica

I commenti di riga possono incorporare suggerimenti che l'autore della richiesta di pull può applicare alla richiesta di pull. Per aggiungere un suggerimento, fai clic sul pulsante  *Insert a suggestion* in cima alla casella di testo del commento e modifica il testo all'interno del blocco del suggerimento.

Suggerimento: Proferire il commit di suggerimenti alla tua richiesta di pull come un batch

Come autore di una richiesta di pull, quando incorpori direttamente il feedback dei revisori nella tua richiesta di pull, evita di usare il pulsante *Commit suggestion* in fondo al commento quando hai molti suggerimenti da affrontare e preferisci aggiungerli come un batch commit, cioè:

1. Passa alla scheda  *Files changed*
2. Premi *Add suggestion to batch* per ogni riformulazione che vuoi includere. Vedrai un contatore che aumenta man mano che vai avanti.
3. Premi uno qualsiasi dei pulsanti *Commit suggestions* quando sei pronto ad applicare i suggerimenti alla tua richiesta di pull, e inserisci un messaggio che descriva le modifiche.

Questo aggiungerà tutte le modifiche al tuo ramo come un singolo commit, risultando in una storia più leggibile dei cambiamenti e meno notifiche per i fruitori del repository. Per inciso, procedere in questo modo ti farà anche risparmiare molti clic.

Apportare correzioni

Una nuova richiesta di pull verrà automaticamente aggiunta alla lista delle richieste di pull <<https://github.com/qgis/QGIS-Documentation/pulls>>_. Altri redattori e amministratori esamineranno la tua richiesta di pull e potranno dare suggerimenti o chiedere correzioni.

Una richiesta di pull innescherà anche controlli automatici di compilazione (ad esempio, per la formattazione rst, sintassi del codice python), e i report sono visualizzati in fondo alla pagina. Se viene trovato un errore, una croce rossa apparirà accanto al tuo commit. Clicca sulla croce rossa o su *Details* nella sezione di riepilogo in fondo alla pagina della richiesta di pull per vedere i dettagli dell'errore. Dovrai correggere qualsiasi errore o avvertimento segnalato prima che le tue modifiche siano inviate al repository `qgis/QGIS-Documentation`.

Puoi fare modifiche alla tua richiesta di pull fino a quando non viene unita al repository principale, sia per migliorare la tua richiesta, sia per indirizzare le modifiche richieste, sia per correggere un errore di compilazione.

Per apportare modifiche clicca sulla scheda  *Files changed* nella tua pagina di richiesta di pull e clicca sul pulsante *matita*  accanto al nome del file che vuoi modificare.

Qualsiasi modifica aggiuntiva sarà automaticamente aggiunta alla tua richiesta di pull se fai queste modifiche allo stesso ramo che hai presentato nella tua richiesta di pull. Per questo motivo, dovresti apportare modifiche aggiuntive solo se tali modifiche si riferiscono al problema che intendi risolvere con quella richiesta di pull.

Se vuoi risolvere un altro problema, crea un nuovo ramo per quelle modifiche e ripeti i passi di cui sopra.

Un amministratore inserirà il tuo contributo dopo che ogni errore di compilazione è stato corretto e dopo che tu e gli amministratori siete soddisfatti delle tue modifiche.

1.1.5 Eliminare il tuo ramo di fusione

Puoi eliminare il ramo dopo che le tue modifiche sono state incorporate. Cancellare i vecchi rami ti evita di avere rami inutilizzati e superati nel tuo repository.

1. Naviga al tuo ramo del repository QGIS-Documentation (<https://github.com/<YourName>/QGIS-Documentation>).
2. Clicca sulla scheda *Branches*. Sotto *Your branches* vedrai una lista dei tuoi rami.
3. Clicca sull'icona  Delete this branch per cancellare qualsiasi ramo indesiderato.

1.2 Usare gli strumenti a riga di comando di Git

L'interfaccia web di GitHub è un modo semplice per aggiornare il repo della documentazione di QGIS con i tuoi contributi, ma non offre strumenti per:

- raggruppare i tuoi commit e pulire la tua cronologia delle modifiche
- correggere possibili conflitti con il repo principale
- costruire la documentazione per testare le tue modifiche

Hai bisogno di [install git](#) sul tuo disco rigido per avere accesso a strumenti più avanzati e potenti e avere una copia locale del repository. Alcune nozioni di base di cui potresti avere spesso bisogno sono esposte qui sotto. Troverai anche delle regole a cui badare anche se opti per l'interfaccia web.

Negli esempi di codice qui sotto, le linee che iniziano con \$ mostrano i comandi che dovresti digitare, mentre # sono commenti.

1.2.1 Repository locale

Ora sei pronto per ottenere un clone locale del **tuo** repository QGIS-Documentation.

Puoi copiare il tuo repository QGIS usando l'URL web come segue:

```
# 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
```

La prima linea di comando è semplicemente un esempio. Dovresti adattare sia il percorso che l'URL del repository, sostituendo <YourName> con il tuo nome utente github.

Verifica quanto segue:

```
# 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* è il nome del repository remoto del tuo repository QGIS-Documentation.
- *master* è il ramo principale di default. Non dovresti mai usarlo per contribuire! ****Mai!**

In alternativa puoi clonare il tuo repository QGIS usando il protocollo SSH:

```
# 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
```

Suggerimento: Errore di autorizzazione negata (chiave pubblica)?

Se ricevi un errore Permission denied (publickey) con il comando precedente, potrebbe esserci un problema con la chiave SSH. Vedere la guida di [GitHub](#) per i dettagli.

Verifica quanto segue se hai usato il protocollo SSH:

```
# 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
```

Puoi iniziare a lavorare qui ma nel processo a lungo termine avrai un sacco di problemi quando spingerai il tuo contributo (chiamato Pull Request nel processo di github) come il ramo principale del repository qgis/QGIS-Documentation divergerà dal tuo repository locale/remoto. Devi quindi tenere traccia del repository remoto principale e lavorare con i rami.

1.2.2 Aggiungere un altro repository remoto

Per poter seguire il lavoro nel progetto principale, aggiungi un nuovo repository remoto nel tuo repository locale. Questo nuovo repository remoto è il repository QGIS-Documentation del progetto QGIS:

```
$ 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)
```

Allo stesso modo, puoi usare il protocollo SSH per aggiungere un repository remoto nel tuo repository locale:

```
$ 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)
```

Così ora hai la scelta tra due repository remoti:

- *origin* per effettuare il push del tuo ramo locale nel **tuo** repository remoto
- *upstream* per fondere (se hai il diritto di farlo) il tuo contributo a quello ufficiale O per aggiornare il tuo ramo master del repository locale dal ramo master del repository ufficiale.

Nota: *upstream* è solo un'etichetta, una specie di nome standard ma puoi chiamarlo come vuoi.

1.2.3 Aggiornare il tuo ramo base

Prima di lavorare su un nuovo contributo, dovresti sempre aggiornare il ramo master nel tuo repository locale. Supponendo che tu sia disposto ad inviare le modifiche alla documentazione di test, esegui le seguenti linee di comando:

```
# 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
```

Ora hai i tuoi repository locali e remoti che hanno entrambi il loro ramo master aggiornato con il ramo ufficiale master di QGIS-Documentation. Puoi iniziare a lavorare sul tuo contributo.

Nota: Cambia il ramo se vuoi contribuire al doc rilasciato

Insieme alla documentazione di test, continuiamo a correggere i problemi nella [latest release](#), il che significa che puoi anche contribuire ad essa. Segui il codice di esempio della sezione precedente, sostituendo master con il ramo corrispondente dell'ultima documentazione.

1.2.4 Contribuire al tuo ramo di produzione

Ora che il tuo ramo base è aggiornato, devi creare un ramo dedicato in cui aggiungere tuoi contributi. Lavora sempre su un ramo diverso da quello di base! Sempre!

```
# 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
```

Poche parole sui comandi commit/push:

- cercare di fare un solo commit (cambiamento atomico), cioè affrontare un solo problema
- cerca di spiegare attentamente cosa cambi nel titolo del tuo commit e nella descrizione. La prima riga è un titolo e dovrebbe iniziare con una lettera maiuscola e avere una lunghezza di 80 caratteri, non finire con un .. Sii conciso. La tua descrizione può essere più lunga, termina con un . e puoi dare molti più dettagli.
- usa un # con un numero per riferirti ad un problema. Usa il prefisso Fix se correggi il ticket: il tuo commit chiuderà il ticket.

Ora che le tue modifiche sono salvate e modificate nel tuo ramo locale, hai bisogno di inviarle al tuo repository remoto per creare una richiesta di pull:


```
$ git push origin myNewBranch
```

1.2.5 Condividere i tuoi cambiamenti

Ora puoi andare al tuo repository github e [create a Pull Request](#) come esposto in una sezione precedente. Assicurati di creare un PR dal tuo ramo al ramo remoto a cui stai puntando nel repository ufficiale di QGIS-Documentation.

1.2.6 Pulire il tuo repository locale e remoto

Dopo che la tua PR è stata fusa nella documentazione ufficiale di QGIS, puoi cancellare il tuo ramo. Se lavori molto in questo modo, in poche settimane avrai un sacco di rami inutili. Quindi mantieni il tuo repository pulito in questo modo:

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

E non dimenticare di aggiornare il ramo principale nella tua repository locale!

1.3 Ulteriori letture

- Oltre all'interfaccia web di Github e agli strumenti a riga di comando git esposti sopra, ci sono anche applicazioni [GUI](#) che puoi usare per creare e gestire i tuoi contributi alla documentazione.
- Quando i cambiamenti nella richiesta di pull sono in conflitto con i recenti cambiamenti inseriti nel ramo di destinazione, i conflitti devono essere risolti prima che una fusione sia possibile:
 - se il conflitto riguarda poche linee concorrenti, nella pagina della richiesta di pull su GitHub è disponibile un pulsante *Resolve conflicts*. Premere il pulsante e risolvere il problema come spiegato in [Resolving a merge conflict on GitHub](#).
 - se il conflitto riguarda la ridenominazione o la rimozione di file, allora devi risolverlo usando le linee di comando di git. In genere, devi prima eseguire il rebase del tuo ramo sul ramo di destinazione usando la chiamata `git rebase targetBranch` e correggere i conflitti che vengono segnalati. Per saperne di più: [Resolving a merge conflict using the command line](#)
- A volte, alla fine del processo di revisione, puoi ritrovarti con modifiche suddivise in più commit che non necessariamente sono validi. Le linee di comando di Git aiutano a ridurre questi commit a un numero minore e a messaggi di commit più significativi. Alcuni dettagli su [Using git rebase on the command line](#)

Writing Guidelines

- *Writing Documentation*
 - *Headlines*
 - *Lists*
 - *Indentation*
 - *Inline Tags*
 - *Labels/references*
 - *Figures and Images*
 - * *Pictures*
 - * *Replacement*
 - * *Figure*
 - * *Tables*
 - *Indice*
 - *Commenti speciali*
 - *Frammenti di codice*
 - *Note a piè di pagina*
- *Gestione delle schermate*
 - *Aggiungere nuove schermate*
 - *Screenshot tradotti*
- *Documentare gli algoritmi di Processing*

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 Writing Documentation

2.1.1 Headlines

To each webpage of the documentation corresponds a `.rst` file.

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

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

Subsection

Minisec
.....

Subminisec
^^^^^^^^

2.1.2 Lists

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

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

2.1.3 Indentation

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

```
#. In a numbered list, there should be  
   three spaces when you break lines  
#. And next items directly follow  
  
* Nested lists  
* Are also possible  
* And when they also have  
  a line that is too long,  
  the text should be naturally  
  aligned  
* and be in their own paragraph
```

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

```
#. This item starts at 1 again
```

2.1.4 Inline Tags

You can use tags to emphasize items.

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

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

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

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

- **Filenames and directories**

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

- **Icons with popup text**

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

(see *image* below).

- **Keyboard shortcuts**

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

will show Ctrl+B

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

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

- **User text**

```
`label`
```

- **Layer names** When referring to layers, format as inline code:

```
`layer name`
```

2.1.5 Labels/references

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

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

```
.. _my_anchor:
Label/reference
-----
```

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

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

which will return:

see [my_anchor](#) for more information.

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

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

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

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

see [Etichette/riferimenti](#) for more information.

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

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

which returns:

see [Label and reference](#) for more information.

2.1.6 Figures and Images

Pictures

To insert an image, use

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

which returns



Replacement

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.

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

Figure

```
.. _figure_logo:

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

   A caption: A logo I like
```

The result looks like this:



Fig. 2.1: A caption: A logo I like

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

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

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

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

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

renders like this:

see Fig. 2.1

This is the preferred way of referencing figures.

Nota: 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 *Una didascalia: Un logo che mi piace*

Tables

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_`.

The result:

Windows	macOS
	
e, naturalmente, non dimenticate 	

La mia tabella disegnata, badate che questo non è purtroppo considerato come una didascalia

Puoi fare riferimento ad esso in questo modo `my_drawn_table`.

Per tabelle ancora più complesse, è più comodo usare `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

```

The result:

Che cosa	Azione	Parola chiave	Descrizione
Test	Useful test	complexity	Geometria. Una di: • Punto • Linea

2.1.7 Indice

Un indice è un modo pratico per aiutare il lettore a trovare le informazioni in un documento. La documentazione di QGIS fornisce alcuni indici essenziali. Ci sono alcune regole che ci aiutano a fornire un insieme di indici che siano veramente utili (coerenti, consistenti e realmente collegati tra loro):

- Un indice dovrebbe essere leggibile, comprensibile e traducibile; un indice può essere composto da molte parole ma dovresti evitare qualsiasi carattere `_`, `-`... non necessario per collegarle, ad esempio, `Loading layers` invece di `loading_layers` o `loadingLayers`.
- Metti in maiuscolo solo la prima lettera dell'indice a meno che la parola non abbia un'ortografia particolare. Per esempio, `Loading layers`, `Atlas generation`, `WMS`, `pgsql2shp`.
- Tieni d'occhio la lista esistente [Index list](#) per riutilizzare l'espressione più conveniente con la giusta ortografia ed evitare inutili duplicati.

Esistono diversi tag di indice in RST. Puoi usare il tag inline `:index:` all'interno del testo normale:

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

Oppure puoi usare il markup a livello di blocco `.. index::` che rimanda all'inizio del paragrafo successivo. A causa delle regole menzionate sopra, si raccomanda di usare il tag block-level:

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

Si raccomanda anche di usare parametri di indice come `single`, `pair` e `see`, per costruire una tabella di indice più strutturata e interconnessa. Vedi [Index generating](#) per maggiori informazioni sulla creazione di indici.

2.1.8 Commenti speciali

A volte, potresti voler enfatizzare alcuni punti della descrizione, sia per avvertire, ricordare o dare qualche suggerimento all'utente. In QGIS Documentation, usiamo le istruzioni speciali reST come `.. warning::`, `.. seealso::`, `.. note::` e `.. tip::`. Queste istruzioni generano dei frame che evidenziano i tuoi commenti. Vedi `Paragraph Level markup` <<https://www.sphinx-doc.org/en/master/usage/restructuredtext/directives.html#paragraph-level-markup>> per maggiori informazioni. Un titolo chiaro e appropriato è richiesto sia per gli avvisi che per i suggerimenti.

```
.. 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.
```

2.1.9 Frammenti di codice

Potresti anche voler dare esempi e inserire frammenti di codice. In questo caso, scrivi il commento sotto una linea con la clausola `::` inserita. Per una visualizzazione migliore, specialmente per applicare l'evidenziazione dei colori al codice in base alla sua lingua, usa la clausola `code-block::` `xml`. Maggiori dettagli in [Showing code](#).

Nota: Mentre i testi nei riquadri delle note, dei suggerimenti e degli avvisi sono traducibili, sii consapevole che i riquadri dei blocchi di codice non ammettono la traduzione. Quindi evita i commenti non relativi al codice e mantieni i commenti il più corti possibile.

2.1.10 Note a piè di pagina

Nota bene: le note a piè di pagina non vengono riconosciute da nessun software di traduzione e non vengono nemmeno convertite correttamente in formato pdf. Quindi, se possibile, non usare le note a piè di pagina in nessuna documentazione.

Questo è per creare una nota a piè di pagina (mostrando come esempio¹)

```
blabla [1]_
```

Che punterà a:

2.2 Gestione delle schermate

2.2.1 Aggiungere nuove schermate

Ecco alcuni suggerimenti per creare nuove schermate dall'aspetto gradevole. Le immagini dovrebbero essere messe in una cartella `image (img/)` che si trova nella stessa cartella del file di riferimento `.rst`.

- Puoi trovare alcuni progetti QGIS pronti che vengono utilizzati per creare le schermate nella cartella `/qgis-projects` di questo repository. In questo modo è più facile riprodurre le schermate per la prossima versione di QGIS. Questi progetti utilizzano i [Sample Data](#) (aka Alaska Dataset), che devono essere decompressi e collocati nella stessa cartella del QGIS-Documentation Repository.
- Riduci la finestra al minimo spazio necessario per mostrare la struttura (prendi tutto lo schermo per una piccola finestra modale > superfluo).

¹ Aggiornamenti dei plugin di base

- Meno disordine c'è, meglio è (non c'è bisogno di attivare tutte le barre degli strumenti)
- Non ridimensionarli in un editor di immagini; la dimensione sarà impostata nei file `.rst` se necessario (ridimensionare le dimensioni senza aumentare correttamente la risoluzione > difficile).
- Taglia lo sfondo
- Imposta gli angoli superiori trasparenti se lo sfondo non è bianco
- Imposta la risoluzione di stampa a 135 dpi (per esempio in Gimp imposta la risoluzione di stampa *Image ► Print size* e salva). In questo modo le immagini saranno alle dimensioni originali in html e ad una buona risoluzione di stampa nel PDF. Puoi anche usare il comando ImageMagick convert per fare un batch di immagini:

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

- Salvali come `.png` (per evitare gli artefatti di `.jpeg`)
- Lo screenshot dovrebbe mostrare il contenuto secondo quanto descritto nel testo

Suggerimento: Se sei su Ubuntu, puoi usare il seguente comando per rimuovere la funzione di menu globale e creare schermate di applicazioni più piccole con menu:

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

2.2.2 Screenshot tradotti

Ecco alcuni suggerimenti aggiuntivi per coloro che vogliono creare screenshot per una guida utente tradotta:

Le immagini tradotte dovrebbero essere messe in una cartella `img/1/`. Usa lo stesso nome di file della schermata inglese «originale».

2.3 Documentare gli algoritmi di Processing

Se vuoi scrivere la documentazione per gli algoritmi di Processing, considera queste linee guida:

- I file di aiuto dell'algoritmo di Processing fanno parte della Guida utente di QGIS, quindi usano la stessa formattazione della Guida utente e dell'altra documentazione.
- La documentazione di ogni algoritmo dovrebbe essere messa nella corrispondente cartella **provider** e nel file **group**, per esempio l'algoritmo *Voronoi polygon* appartiene al provider *QGIS* e al gruppo *vectorgeometry*. Quindi il file corretto per aggiungere la descrizione è: `source/docs/user_manual/processing_algs/qgis/vectorgeometry.rst`.

Nota: Prima di iniziare a scrivere la guida, controllare se l'algoritmo è già descritto. In tal caso, è possibile migliorare la descrizione esistente.

- È **estremamente** importante che ogni algoritmo abbia un *ancora* che corrisponde al nome del provider + il nome univoco dell'algoritmo stesso. Questo permette al pulsante Help di aprire la pagina di aiuto della sezione corretta. L'ancora dovrebbe essere posta **sopra** il titolo, ad esempio (vedi anche la sezione *Etichette/riferimenti*):

```
.. _qgisvoronoipolygons:

Voronoi polygons
-----
```

Per trovare il nome dell'algoritmo puoi semplicemente passare il mouse sull'algoritmo nella casella degli strumenti di Processing.

- Evita di usare «Questo algoritmo fa questo e quello...» come prima frase della descrizione dell'algoritmo. Cerca di usare espressioni più generali come:

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

- Evita di descrivere cosa fa l'algoritmo replicando il suo nome e per favore non replicare il nome del parametro nella descrizione del parametro stesso. Per esempio se l'algoritmo è Voronoi polygon puoi descrivere il Layer in ingresso come Layer da cui calcolare il poligono.
- Indica nella descrizione se l'algoritmo ha una scorciatoia predefinita in QGIS o supporta la modifica sul posto
- Aggiungete immagini! Un'immagine vale più di mille parole! Usa il formato .png e segui le linee guida generali per la documentazione (vedi la sezione [Figure e Immagini](#) per maggiori informazioni). Metti il file immagine nella cartella corretta, cioè la cartella img accanto al file .rst che stai modificando.
- Se necessario, aggiungi dei link nella sezione «Vedi anche» che forniscono informazioni aggiuntive sull'algoritmo (ad esempio, pubblicazioni o pagine web). Aggiungi la sezione «Vedi anche» solo se c'è veramente qualcosa da vedere. Come buona pratica, la sezione «Vedi anche» può essere riempita con link ad algoritmi simili.
- Dai una spiegazione chiara dei parametri e dei risultati dell'algoritmo: prendi ispirazione dagli algoritmi esistenti.
- Evita di duplicare la descrizione dettagliata delle opzioni dell'algoritmo. Aggiungi queste informazioni nella descrizione del parametro.
- Evita di aggiungere informazioni sul tipo di geometria vettoriale nella descrizione dell'algoritmo o del parametro, poiché queste informazioni sono già disponibili nelle descrizioni dei parametri.
- Aggiungi il valore predefinito del parametro, ad esempio:

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

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

- Descrivi il *tipo* di dati in ingresso supportati dai parametri. Ci sono diversi tipi disponibili che puoi scegliere:

Tipo di parametro/risultato	Descrizione	Indicatore visuale
Layer vettore punto	vector: point	
Layer vettore linea	vector: line	
Layer vettoriale poligono	vector: polygon	
Layer vettoriale generico	vector: any	
Campo vettoriale numerico	tablefield: numeric	1.2
Campo vettoriale stringa	tablefield: string	abc
Campo generico vettore	tablefield: any	
Layer raster	raster	
Banda raster	raster band	
File HTML	html	
Layer tabella	table	
Espressione	expression	
Geometria punto	coordinates	
Estensione	extent	
SR	crs	
Numerazione	enumeration	

continues on next page

Tabella 2.1 – continua dalla pagina precedente

Tipo di parametro/risultato	Descrizione	Indicatore visuale
Elenco	list	
Numero	number	1,00 ▾
Stringa	string	Display name lakes.shp
Booleano	boolean	☒
Percorso cartella	folder	
File	file	
Matrice	matrix	
Layer	layer	
Stesso tipo in uscita del tipo in ingresso	same as input	
Definizione	definition	
Punto	point	
Multilayer	multipleLayers	
Intervallo	range	
Autoconfigurazione	authconfig	
Mesh	mesh	
Layout	layout	
LayoutItem	layoutitem	
Colore	color	
Scala	scale	

- Studia un algoritmo esistente e ben documentato, e copia tutti i layout utili.
- Quando hai finito, segui le linee guida descritte in [Una contribuzione passo dopo passo](#) per confermare le tue modifiche e fare una richiesta di Pull.

Ecco un esempio di un existing algorithm per esserti d'aiuto con il layout e la descrizione:

```
.. _qgiscountpointsinpolygon:

Count points in polygon
-----

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

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

   The labels in the polygons show the point count

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

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

Parameters
.....

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

   * - Label
```

(continues on next page)

```

- 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
   :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

```

Scrivere codice nel PyQGIS Cookbook

- *Come scrivere parti di codice testabili*
 - *Direttive Doctest sphinx*
 - *Raggruppare i test*
- *Come testare pezzi di codice sulla tua macchina locale*

Se hai intenzione di aggiungere o aggiornare alcuni capitoli del PyQGIS-Developer-Cookbook, allora dovresti seguire alcune regole per permettere il test automatico dei blocchi di codice.

I test sono molto importanti perché permettono il controllo automatico del codice. I blocchi di codice con errori o il codice che usa metodi obsoleti fallirà e la notifica ti aiuterà a risolvere i problemi.

Per i test, usiamo l'estensione «Sphinx doctest» <<https://www.sphinx-doc.org/en/master/usage/extensions/doctest.html>>_. Fai riferimento alla documentazione dell'estensione per informazioni più dettagliate.

3.1 Come scrivere parti di codice testabili

Scrivere pezzi di codice testabili non è così diverso dal *vecchio* metodo. Fondamentalmente, è necessario utilizzare una diversa «direttiva» di Sphinx.

3.1.1 Direttive Doctest sphinx

Invece di nidificare il codice in una frase `.. code-block:: python` (che evidenzierebbe automaticamente la sintassi del codice), ora devi incorporarlo in un `.. testcode::`. Cioè, invece di questo:

```
.. code-block:: python

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

Ora usa questo:

```
.. testcode::

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

Dopo che hai scritto il codice di esempio, dovresti aggiungere qualche *attestazione* che valuterà il codice e che verrà eseguita automaticamente.

Nell'esempio di cui sopra, stai creando un SR e con `assert crs.isValid()` fai un **test** se è valido. Se il codice ha una sintassi python sbagliata o la `crs.isValid()` restituisce `False`, questo frammento di codice fallirà durante il test.

Per eseguire con successo i test sugli snippet, devi importare tutte le classi e dichiarare tutte le variabili usate negli snippet di codice. Puoi includerle nello snippet stesso (visibile nelle pagine HTML) o puoi aggiungerle ad una frase `.. testsetup::` (nascosta nelle pagine HTML). La frase `.. testsetup::` deve essere messa prima della frase `.. testcode::`:

```
.. testsetup::

    from qgis.core import QgsCoordinateReferenceSystem

.. testcode::

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

Se lo snippet di codice non crea oggetti (e quindi non puoi usare qualcosa come `assert object.isValid()`), puoi testare il codice usando il metodo `print()`, quindi aggiungi i risultati attesi all'interno di una direttiva `.. testoutput::` per confrontare l'output atteso:

```
.. testcode::

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

.. testoutput::

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

Per impostazione predefinita, il contenuto di `.. testoutput::` viene mostrato nell'output HTML. Per nascondere dall'HTML usa l'opzione `:hide:`:

```
.. testoutput::
    :hide:

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

Nota: Se il frammento di codice contiene delle istruzioni di stampa, DEVI aggiungere un `testoutput` con gli output previsti; altrimenti il test fallirà.

3.1.2 Raggruppare i test

Per ogni documento rst, gli snippet di codice sono testati in modo sequenziale, il che significa che si può usare un `.. testsetup::` per tutti gli snippet di codice successivi e che gli snippet successivi avranno accesso alle variabili dichiarate in quelli precedenti nel documento.

In alternativa, puoi usare i gruppi per suddividere gli esempi sulla stessa pagina in diversi test.

Aggiungi il frammento di codice ai gruppi aggiungendo uno o più nomi di gruppo (separati da virgole) nella rispettiva direttiva:

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

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

Il doctest sceglierà ciascun gruppo di snippet e li eseguirà indipendentemente.

Nota: Usa nomi di gruppo che abbiano senso con il contenuto correlato. Usa qualcosa di simile a `<chapter>_<subchapter>`, per esempio: `crs_intro`, `crs_fromwkt`. In caso di errori, questo aiuterà a identificare dove si verificano gli errori.

Se non dichiari alcun gruppo, lo snippet di codice sarà aggiunto a un gruppo chiamato `default`. Se invece usi `*` come nome del gruppo, lo snippet sarà usato in tutti i gruppi di test, cosa normalmente utile da usare nella configurazione dei test:

```
.. testsetup:: *

    from qgis.core import QgsCoordinateReferenceSystem
```

3.2 Come testare pezzi di codice sulla tua macchina locale

Nota: Le istruzioni sono valide per il sistema Linux.

Per testare pezzi di codice Python, hai bisogno di un'installazione *QGIS*. Per questo, ci sono molte opzioni. Puoi:

- Utilizzare la tua installazione *QGIS* di sistema con *Sphinx* da un ambiente virtuale Python:

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

- Utilizzare un'installazione costruita manualmente di *QGIS*. Avrai bisogno di:

1. Creare un'estensione personalizzata Makefile sopra il file `venv.mk`, per esempio un file `user.mk` con il seguente contenuto:

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

include venv.mk
```

O:

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

include venv.mk
```

2. Poi, usatelo per eseguire il test `doctest`:

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

- Esegui il test `doctest` all'interno dell'immagine originale di *QGIS*:

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

Devi installare prima [Docker](#) perché questo usa un'immagine con QGIS originale al suo interno.

Linee guida per la traduzione

- *Processo di traduzione*
- *Traduci un file*
 - *Traduzione in Transifex*
 - *Traduzione in QT Linguist*
 - *Traducurre il manuale*
 - *Sommario delle regole per la traduzione*

Questo manuale ha lo scopo di aiutare il traduttore. Prima viene spiegato il processo generale di come viene fatta tecnicamente una traduzione. In seguito viene spiegata la traduzione da un vero e proprio documento inglese che viene tradotto in olandese. Infine viene dato un riassunto delle *Rules of translation*.

Nota: Sebbene queste linee guida si focalizzino sulla documentazione di QGIS, i metodi e le regole descritte in seguito sono applicabili anche alle traduzioni dell'applicazione QGIS e del sito web.

4.1 Processo di traduzione

La documentazione di QGIS è scritta in inglese con file `.rst`. Al fine di fornire traduzioni:

1. Uno script di precostruzione crea file di traduzione denominati `.po` per la lingua inglese nella cartella `/QGIS-Documentation/locale/en`.
2. Le frasi nei file `.po` sono caricate sulla piattaforma web Transifex, e rese disponibili per i traduttori che possono iniziare a tradurre dall'inglese alla loro lingua con l'editor.
3. Quando un file viene tradotto al 100%, le stringhe tradotte vengono automaticamente trasferite nel repository della documentazione, in `/QGIS-Documentation/locale/<language>`.
4. Alla successiva creazione della documentazione (che avviene almeno una volta al giorno - si veda l'orario in fondo alla pagina), uno script riutilizza le frasi per creare un output tradotto.
5. Per i file non completamente tradotti, uno script estrae ogni due settimane le stringhe tradotte da Transifex a Github e anche queste vengono pubblicate nella build successiva.

6. Ogni volta che un file `.rst` viene aggiornato, il file `.po` inglese viene aggiornato e le modifiche vengono trasferite al file corrispondente in Transifex. Ciò significa che quando si aggiunge un nuovo paragrafo a un documento `.rst` già tradotto, solo le frasi nuove/aggiornate vengono aggiunte al file `.po` tradotto e devono essere tradotte.

Nota: Tradurre le specificità di QGIS Desktop

La differenza principale nella traduzione delle applicazioni QGIS è che invece dei file `.po`, tutte le stringhe traducibili nei file `.py`, `.cpp`, `.yaml` che danno forma a una particolare versione dell'applicazione vengono inviate e prelevate da Transifex come un singolo file `.ts` (ad esempio, `qgis-application/qgis_en.ts` (branch `release-3_30`)). Le traduzioni vengono estratte su Github nel ramo di sviluppo (quotidianamente) e al momento del rilascio (per ogni versione rilasciata).

Due diversi strumenti sono attualmente utilizzati per la traduzione di QGIS:

- La [piattaforma web Transifex](#), il modo più semplice e consigliato per tradurre QGIS, fa in modo trasparente il processo descritto sopra e raccoglie tutti i testi traducibili in un posto per il traduttore. Basta scegliere i file che vuoi e tradurre. I file tradotti sono immagazzinati nella piattaforma fino a quando un'altra release viene rilasciata.
- [Qt Linguist](#), uno strumento di sviluppo di Qt, richiede al traduttore di estrarre localmente i file `.po` (o `.ts`) dal codice sorgente, tradurli e poi rispedirli.

Da notare che qualsiasi strumento tu scelga, le regole di traduzione sono le stesse.

4.2 Traduci un file

Per spiegare come funziona la traduzione, useremo il plugin Mappa di concentrazione come esempio. In questo esempio, tradurremo da Inglese a Olandese, ma sarà praticamente lo stesso per gli altri documenti in tutte le lingue.

Il sorgente del documento può essere trovato qui:

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

Perché ho scelto questo documento?

1. Include immagini, didascalie, intestazioni, riferimenti e sostituzioni.
2. L'ho scritto perché così è più facile per me tradurre ;-)

Il processo di generazione ha creato il file inglese `.po` che può essere trovato qui:

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

L'equivalente file olandese `.po` (praticamente una copia) può essere trovato qui:

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

Insieme a questo file vedrete un piccolo file `.mo` che indica che non contiene ancora alcuna traduzione.

4.2.1 Traduzione in Transifex

Per tradurre usando Transifex, devi:

1. avere un account su Transifex e unirsi al progetto QGIS <https://qgis.org/en/site/getinvolved/translate.html#join-a-project>.
2. Una volta che fai parte di un team linguistico, clicca sul progetto corrispondente (in questo caso QGIS Documentation). Viene visualizzato un elenco di lingue disponibili con il loro rapporto di traduzione.

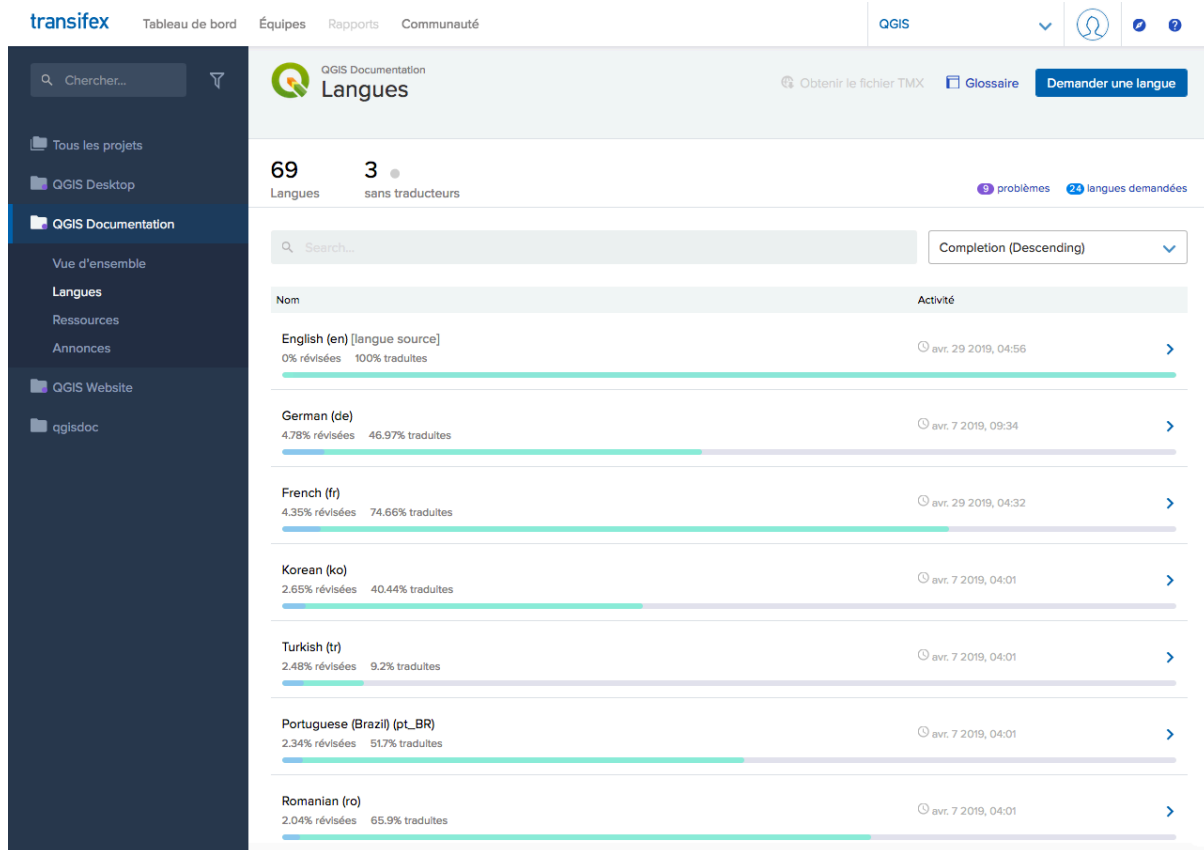


Fig. 4.1: Seleziona la lingua per la traduzione nel menu Transifex

3. Spostati sulla tua lingua e clicca su uno dei due:
 - *View resources*: sono ora visualizzati i file traducibili .po con il loro rapporto di traduzione, il numero di stringhe e alcuni altri metadati.
 - o *Translate*: apre l'interfaccia di traduzione con tutti i file .po disponibili
4. Identifica il file che vorresti tradurre (nel nostro caso stiamo cercando il docs_user-manual_plugins_plugins-heatmap, il file del plugin heatmap) o qualsiasi file non finito e clicca su di esso: le stringhe nei file vengono caricate e puoi usare l'interfaccia per filtrare, tradurre, suggerire la traduzione...

Suggerimento: Per la documentazione o il sito web, cliccando sul link `Fix me` nel piè di pagina di una pagina si arriva direttamente alla pagina di traduzione corrispondente in Transifex.

5. Tutto quello che devi fare è selezionare ogni testo e tradurre seguendo le [guidelines](#).

Per ulteriori informazioni sull'uso di Transifex Web Editor, vedere <https://help.transifex.com/en/articles/6318216-translating-with-the-web-editor>.

4.2.2 Traduzione in Qt Linguist

Con Qt Linguist, devi:

1. prendere manualmente il(i) file `.po` o `.ts`. Questo può essere ottenuto scaricando il(i) file o dalla piattaforma Transifex o dalla cartella `locale/$language` del repository dei sorgenti (in GitHub),
2. procedere alla traduzione in locale
3. caricare i file modificati nei loro sorgenti (Transifex o GitHub).

Anche se il download e l'upload di file traducibili può essere fatto con Transifex, non è consigliabile utilizzare questo processo. Dal momento che non c'è un sistema di versioning su Transifex, il file che caricate sostituirà semplicemente quello esistente e potenzialmente sovrascriverà qualsiasi modifica fatta nel frattempo da altri sulla piattaforma.

Quando apri il file in Qt Linguist per la prima volta vedrai la seguente finestra di dialogo:



Fig. 4.2: Seleziona la lingua per la traduzione nel menu linguistico

La lingua tradotta può essere riempita correttamente. La lingua sorgente può essere lasciata così come lingua POSIX e Nazione/Regione o Tutte le nazioni.

Quando premi il pulsante *OK* Qt Linguist viene riempito di frasi e puoi iniziare a tradurre, vedi Fig. 4.3.

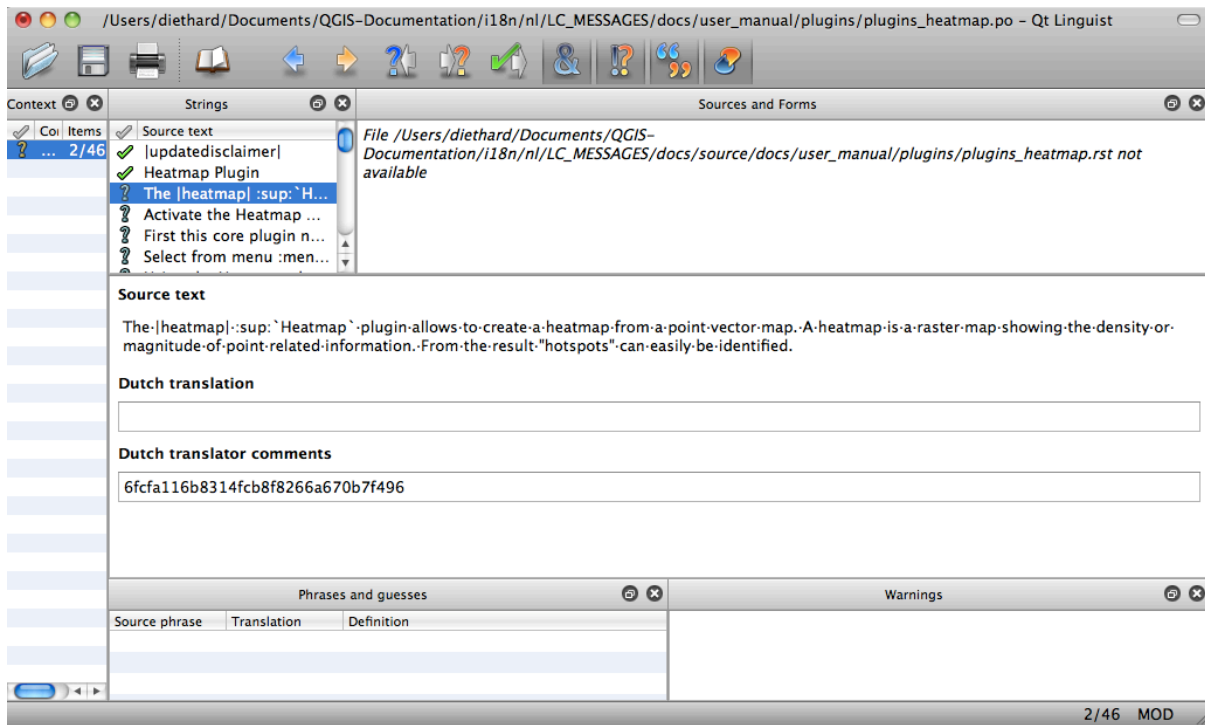


Fig. 4.3: Traduzione usando il menu linguistico

Nel menù puoi vedere i seguenti bottoni che sono preferibilmente da utilizzare.

-  Il pulsante Translation Done Next è il più importante. Se la voce ha bisogno di traduzione, inserisci una traduzione nel campo di testo, poi premi questo pulsante. Se la voce non ha bisogno di traduzione, lascia il campo di testo per la traduzione vuoto e premi questo pulsante che indica che la voce è stata completata e continua con la voce successiva.
-  Il pulsante Goto Previous può essere usato per andare alla voce di traduzione precedente.
-  Il pulsante Goto Next può essere usato per andare alla prossima voce di traduzione.
-  Il pulsante Next Todo passa alla prima voce di traduzione che ha ancora bisogno di una traduzione. Comodo quando il documento originale è cambiato e solo alcune frasi nuove/modificate devono essere tradotte.
-  Il pulsante Previous Todo cerca all'indietro e salta alla prima voce di traduzione che trova che ha ancora bisogno di una traduzione.

Per ulteriori informazioni sull'uso di Qt Linguist, vedere <https://doc.qt.io/qt-5/linguist-translators.html>.

Avvertimento: Se vuoi scaricare contenuti da tradurre dal repository dei sorgenti, non farlo mai nel ramo `master`. Per le traduzioni ci sono sempre branche di traduzione disponibili, una volta che un documento è completamente aggiornato in inglese per una certa versione. Per esempio, per tradurre il manuale di QGIS 2.8, devi usare la branca `manual_en_v2.8`.

4.2.3 Traducurre il manuale

Ora possiamo iniziare a tradurre il manuale del plugin_heatmap

La traduzione della maggior parte delle frasi dovrebbe essere semplice. Durante questa sessione di traduzione indicherò quali parti (istruzioni rst) hanno bisogno di una traduzione speciale.

Qui sotto vediamo una frase interessante da tradurre:

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.

Questa frase contiene due istruzioni rst:

1. «**lMappa di concentrazione**» le parole tra «|» sono sostituite e queste non dovrebbero mai essere tradotte. Questo sostituirà l'icona del plugin Mappa di concentrazione.
2. :sup:`Heatmap`, l'istruzione :sup: è una dichiarazione di superposizione e stampa il testo seguente un po' più in alto. Questo è usato per mostrare i testi popup che appaiono quando passi il mouse sopra l'elemento della barra degli strumenti e questo può essere diverso quando viene effettivamente tradotto nell'applicazione QGIS. Nel caso olandese non lo è!

Tutti gli altri file di testo in questa frase possono essere tradotti!

Il prossimo oggetto di traduzione contiene la dichiarazione :ref: che è comunemente usata per riferirsi ad un'altra sezione da qualche parte nel manuale! Il testo che segue una dichiarazione :ref: non dovrebbe mai essere cambiato perché è un identificatore univoco!

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 questo caso load_core_plugin è un identificatore di riferimento univoco posto prima di un elemento rst che ha una didascalia. L'istruzione ref sarà sostituita con il testo dell'intestazione e trasformata in un collegamento ipertestuale. Quando l'intestazione a cui questo riferimento si riferisce viene tradotta, anche tutti i riferimenti a questa intestazione saranno automaticamente tradotti.

L'elemento successivo contiene il rst-tag :menuselection: seguito dal testo effettivamente visualizzato in un menu nell'applicazione QGIS, questo può essere tradotto nell'applicazione e quindi dovrebbe essere cambiato.

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

Nel punto sopra «View -->» è effettivamente tradotto in «Beeld -->» perché questa è la traduzione usata nell'applicazione QGIS localizzata in olandese.

Un po' più avanti incontriamo il seguente elemento di traduzione complicato:

The |heatmap| :sup:`Heatmap` tool button starts the Dialog of the Heatmap plugin (see :numref:`figure\_heatmap\_settings`).

Contiene un riferimento a una figura figure_heatmap_settings_, e come un riferimento a una sezione questo riferimento non dovrebbe essere cambiato! La definizione di riferimento dal documento rst non è inclusa nel file .po e quindi non può essere cambiata. Questo significa che il riferimento alle figure non può essere tradotto. Quando viene creato l'HTML si vedrà figure_heatmap_settings_. Quando un documento PDF viene creato figure_heatmap_settings_ viene sostituito da un numero di figura.

Il successivo oggetto di traduzione con attributi rst è l'oggetto seguente:

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

Non rimuovere le stelle nella linea precedente. Il testo che contiene verrà stampato in grassetto. Il testo stesso è spesso incluso nella finestra di dialogo stessa e può anche essere tradotto nell'applicazione.

Il seguente elemento di traduzione contiene il tag rst `:guilabel:`.

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

Il testo `Advanced` del tag `guilabel` può benissimo essere tradotto nell'applicazione QGIS e probabilmente deve essere cambiato!

Il seguente elemento di traduzione contiene ```airports```. Le virgolette sono usate per dare al testo un altro carattere di testo. In questo caso è un valore letterale e non ha bisogno di traduzione.

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>`_`.`

Questa voce include anche un collegamento ipertestuale con un url e una presentazione esterna. L'url deve ovviamente essere lasciato intatto, sei autorizzato a cambiare il testo esterno `https://www.qgistutorials.com` che è visibile dal lettore. Non rimuovere mai il trattino basso alla fine del collegamento ipertestuale che ne costituisce una parte essenziale!

4.2.4 Sommario delle regole per la traduzione

1. Non cambiare il testo tra due caratteri | come ad esempio |bronze|, |checkbox|, |labels|, |selectString|, |addLayer| ... Questi sono tag speciali usati per essere sostituiti da immagini
2. Non cambiare i riferimenti che iniziano con nomi come `:ref:`, `:file:`, `:numref:` a meno che non includano un titolo. In questo caso, puoi tradurre il titolo ma mantenere invariato il collegamento (cioè il testo tra `< e >`)

Suggerimento: Quando viene fornito un titolo per un riferimento, Transifex può visualizzare un numero nel testo sorgente inglese in sostituzione della parte di collegamento. Clicca sul numero nel testo sorgente per aggiungere il link di riferimento accanto al titolo da tradurre.

3. Non cambiare i riferimenti che finiscono con un trattino basso come `figure_labels_1_`.
4. Non cambiare l'url nei collegamenti ipertestuali, ma puoi cambiare la descrizione esterna. Lascia il trattino basso alla fine del collegamento ipertestuale, senza spaziatura aggiuntiva (`>`_``)
5. Cambia il testo all'interno delle virgolette che seguono i tag `:index:`, `:sup:`, `:guilabel:` e `:menuselection:`. Controlla se/come viene tradotto nell'applicazione QGIS. Non cambiare il tag stesso.
6. Il testo tra le doppie stelle e le doppie virgolette spesso indica valori o nomi di campi, a volte hanno bisogno di traduzione a volte no.
7. Fate attenzione ad usare esattamente gli stessi (stesso numero di) caratteri speciali del testo sorgente come ```, ````, `*`, `**`, `:`. Questi contribuiscono alla cosmesi delle informazioni fornite
8. Non iniziare né terminare la stringa di testo con caratteri speciali o tag con uno spazio
9. Non terminare le stringhe tradotte con un nuovo paragrafo, altrimenti il testo non sarà tradotto durante la generazione dell'html.

Atteniti alle regole di cui sopra e il documento tradotto avrà un bell'aspetto!

Per qualsiasi domanda, contatta il [QGIS Community Team](#) o il [QGIS Translation Team](#).

- *Uso*
- *Sostituzioni comuni*
 - *Icone di piattaforma*
 - *Oggetti del menu*
- *Icone pulsanti barra degli strumenti*
 - *Gestione layer e panoramica*
 - *Progetto*
 - *Modifica*
 - *Identity result*
 - *Digitalizzazione e digitalizzazione avanzata*
 - *Mesh*
 - *Orientazione della mappa e attributi*
 - *Selezione ed Espressioni*
 - *Etichette e Diagrammi*
 - *Decorazioni*
 - *Guida*
 - *Colori*
- *Altre icone di base*
- *Tabella degli attributi*
- *Proiezioni e Georeferenziatore*
- *Layout di stampa*
- *Proprietà Layer*
- *Plugin*

- *Processing*
- *Varie Plugin di Base*
- *Integrazione con GRASS*

5.1 Uso

Per facilitare l'uso delle icone nei manuali QGIS, le sostituzioni sono definite per ogni icona nel file `/source/substitutions.txt` al [repository QGIS-Documentation](#) e alcune di queste sostituzioni sono elencate qui sotto. Così, quando vuoi usare un'icona dell'applicazione QGIS nella documentazione c'è una buona probabilità che ci sia già una sostituzione che può/deve essere usata.

Se non esiste alcuna sostituzione:

1. controllare nel repository della documentazione se l'icona è disponibile nella cartella `/static/common`. Se non c'è l'immagine, è necessario trovare e copiare il file dell'immagine dell'icona dal repository [QGIS](#) (spesso sotto la cartella `default themes`) e incollarlo (in formato `.png`) nella cartella `/static/common`. Per comodità e aggiornamento, si consiglia di mantenere il nome del file quando possibile.
2. creare il riferimento alla sostituzione nel file `/sostituzioni.txt` seguendo l'esempio seguente. Il testo della sostituzione deve essere derivato dal nome del file e in camelCase:

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

3. Aggiornare la(e) sezione di destinazione dei documenti, usando la nuova sostituzione.
4. (opzionale ma altamente raccomandato) aggiungere la sostituzione all'elenco sottostante.
5. Aggiungere il riferimento alla nuova sostituzione nell'elenco delle sostituzioni alla fine del(i) file in cui viene utilizzata, oppure eseguire il consueto script `scripts/find_set_subst.py`.

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

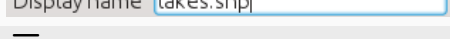
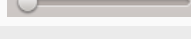
5.2 Sostituzioni comuni

Di seguito sono date alcune icone e la loro sostituzione da usare quando si scrive la documentazione. Possono essere usate/trovate in molti posti nei manuali.

5.2.1 Icone di piattaforma

Icona	Sostituzione	Icona	Sostituzione
	logo		
	kde		nix
	osx		win

5.2.2 Oggetti del menu

Icona	Sostituzione	Icona	Sostituzione
	checkbox		unchecked
	radioButtonOn		radioButtonOff
	selectNumber		selectString
	selectColor		selectColorRamp
	tab		degrees
	inputText		slider
	hamburgerMenu		

5.3 Icone pulsanti barra degli strumenti

5.3.1 Gestione layer e panoramica

Icona	Sostituzione	Icona	Sostituzione
	dataSourceManager		
	addOgrLayer		
	addRasterLayer		addMssqlLayer
	addDelimitedTextLayer		addSpatialiteLayer
	addPostgisLayer		addOracleLayer
	addAfsLayer		addMeshLayer
	addVectorTileLayer		addXyzLayer
	addVirtualLayer		addWmsLayer
	addWcsLayer		addWfsLayer
	addPointCloudLayer		addGpsLayer
	addTiledSceneLayer		addHanaLayer
	newVectorLayer		newSpatialiteLayer
	newGeoPackageLayer		createMemory
	newVirtualLayer		newMeshLayer
	newGpx		
	dbManager		gdal
	geoPackage		spatialite
	virtualLayer		wms
	wcs		wfs
	pointCloudLayer		gps
	tiledSceneLayer		hana

continues on next page

Tabella 5.1 – continua dalla pagina precedente

Icona	Sostituzione	Icona	Sostituzione
	dbSchema		
	inOverview		addAllToOverview
	removeAllFromOverview		removeLayer
	showAllLayers		hideAllLayers
	showPresets		showSelectedLayers
	hideSelectedLayers		hideDeselectedLayers
	toggleAllLayers		toggleSelectedLayers
	addLayer		
	indicatorTemporal		indicatorNonRemovable
	indicatorEmbedded		indicatorFilter
	indicatorMemory		indicatorNoCRS
	indicatorBadLayer		favourites
	indicatorLayerError		indicatorNotes
	indicatorLowAccuracy		indicatorOffline

5.3.2 Progetto

Icona	Sostituzione	Icona	Sostituzione
	fileNew		fileOpen
	fileSave		fileSaveAs
	fileExit		user

5.3.3 Modifica

Icona	Sostituzione	Icona	Sostituzione
	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 Identity result

Icona	Sostituzione	Icona	Sostituzione
	expandTree		collapseTree
	expandNewTree		formView
	deselectAll		editCopy
	filePrint		
	identifyByRectangle		identifyByPolygon
	identifyByFreehand		identifyByRadius

5.3.5 Digitalizzazione e digitalizzazione avanzata

Icona	Sostituzione	Icona	Sostituzione
	cad		cadConstruction
	cadParallel		cadPerpendicular
	floater		
	toggleEditing		allEdits
	tracing		snapping
	snappingVertex		snappingSegment
	snappingArea		snappingCentroid

continues on next page

Tabella 5.2 – continua dalla pagina precedente

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

5.3.6 Mesh

Icona	Sostituzione	Icona	Sostituzione
	meshDigitizing		meshReindex
	meshSelectExpression		meshSelectPolygon
	meshTransformByExpression		meshEditForceByVectorLines
	vertexCoordinates		

5.3.7 Orientazione della mappa e attributi

Icona	Sostituzione	Icona	Sostituzione
	pan		panToSelected
	zoomIn		zoomOut
	zoomActual		zoomFullExtent
	zoomToLayer		zoomToSelected
	zoomLast		zoomNext
	zoomInXAxis		refresh
	identify		mapTips
	showBookmarks		newBookmark
	measure		measureArea
	measureBearing		measureAngle
	newMap		new3DMap
	tiltUp		tiltDown
	3dNavigation		play
	temporal		temporalNavigationOff
	temporalNavigationFixedRange		temporalNavigationAnimated
	newElevationProfile		

5.3.8 Selezione ed Espressioni

Icona	Sostituzione	Icona	Sostituzione
	selectRectangle		selectPolygon
	selectFreehand		selectRadius
	selectAll		deselectAll
	invertSelection		expressionSelect
	deselectActiveLayer		
	selectDistance		selectLocation
	selectAllTree		select
	selectAdd		selectRemove
	formSelect		dataDefine
	expression		dataDefineOn
	dataDefineExpressionOn		dataDefineError
	dataDefineExpressionError		
	addExpression		
	expressionFilter		filterMap

5.3.9 Etichette e Diagrammi

Icona	Sostituzione	Icona	Sostituzione
	labelingSingle		labelingNone
	labelingRuleBased		labelingObstacle
	piechart		diagramNone
	text		histogram
	stackedBar		
	createAnnotationLayer		annotationLayer
	textAnnotation		svgAnnotation
	formAnnotation		htmlAnnotation
	actionText		textAlongLine
	labelbackground		labelbuffer
	labelformatting		labelplacement
	labelshadow		render
	labelcallout		
	labelAnchorCenter		labelAnchorCustom
	labelAnchorEnd		labelAnchorStart
	pinLabels		showHideLabels
	moveLabel		rotateLabel
	showPinnedLabels		showUnplacedLabel
	changeLabelProperties		autoPlacementSettings

5.3.10 Decorazioni

Icona	Sostituzione	Icona	Sostituzione
	copyrightLabel		addGrid
	titleLabel		northArrow
	scaleBar		addMap
	addImage		

5.3.11 Guida

Icona	Sostituzione	Icona	Sostituzione
	helpContents		qgisHomePage
	success		
	helpSponsors		contextHelp

5.3.12 Colori

Icona	Sostituzione	Icona	Sostituzione
	colorBox		colorPicker
	colorSwatches		colorWheel

5.4 Altre icone di base

Icona	Sostituzione	Icona	Sostituzione
	arrowLeft		arrowRight
	arrowDown		arrowUp
	symbologyAdd		symbologyRemove
	projectProperties		options
	interfaceCustomization		keyboardShortcuts
	copyrightLabel		northArrow
	scaleBar		tracking
	gpsTrackBarChart		
	gpsConnect		gpsDisconnect
	gpsDestinationLayer		addTrackPoint
	recenter		reset
	folder		extents
	settings		start
	properties		deleteSelected
	browserExpand		browserCollapse
	codeEditor		add
	relations		layoutItem3DMap
	stopwatch		sensor
	clearItem		

5.5 Tabella degli attributi

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

5.6 Proiezioni e Georeferenziatore

Icona	Sostituzione	Icona	Sostituzione
	geographic		crs
	customProjection		setProjection
	projectionDisabled		projectionEnabled
	transformation		gdalScript
	georefRun		pencil
	linkQGisToGeoref		linkGeorefToQGis
	fullHistogramStretch		

5.7 Layout di stampa

Icona	Sostituzione	Icona	Sostituzione
	newLayout		layoutManager
	duplicateLayout		
	newReport		newPage
	atlasSettings		atlas
	filePrint		saveMapAsImage
	saveAsSVG		saveAsPDF
	addBasicShape		addBasicCircle
	addBasicTriangle		addBasicRectangle
	addNodesShape		editNodesShape
	addPolygon		addPolyline
	addArrow		northArrow
	add3DMap		addMap
	elevationProfile		copyProfileSettings
	addLegend		addHtml
	addManualTable		addTable
	addImage		addMarker
	label		scaleBar
	select		moveItemContent
	setToCanvasScale		setToCanvasExtent
	viewScaleInCanvas		viewExtentInCanvas
	raiseItems		lowerItems
	moveItemsToTop		moveItemsToBottom
	alignLeft		alignRight
	alignHCenter		alignVCenter
	alignTop		alignBottom
	distributeLeft		distributeRight
	distributeTop		distributeBottom
	distributeHCenter		distributeVCenter
	distributeHSpace		distributeVSpace
	resizeShortest		resizeTallest
	resizeNarrowest		resizeWidest
	resizeSquare		groupItems
	lockItems		unlockAll

continues on next page

Tabella 5.3 – continua dalla pagina precedente

Icona	Sostituzione	Icona	Sostituzione
	locked		unlocked
	lockRepeating		lockedGray

5.8 Proprietà Layer

Icona	Sostituzione	Icona	Sostituzione
	symbolology		labelingSingle
	sourceFields		general
	metadata		action
	display		rendering
	join		diagram
	labelmask		temporal
	legend		dependencies
	3d		system
	elevationscale		layerTree
	editMetadata		overlay
	digitizing		auxiliaryStorage
	history		stylePreset
	search		pyramids
	transparency		rasterHistogram
	singleSymbol		nullSymbol
	graduatedSymbol		categorizedSymbol
	25dSymbol		ruleBasedSymbol
	invertedSymbol		heatmapSymbol
	pointDisplacementSymbol		pointClusterSymbol
	mergedFeatures		
	meshcontours		meshcontoursoff
	meshvectors		meshvectorsoff
	meshframe		meshaveraging
	singleColor		paletted
	singlebandPseudocolor		multibandColor
	pointCloudExtent		
	sum		sort

continues on next page

Tabella 5.4 – continua dalla pagina precedente

Icona	Sostituzione	Icona	Sostituzione
	paintEffects		mapIdentification
	styleManager		iconView
	joinNotEditable		joinedLayerNotEditable
	joinHasNotUpsertOnEdit		filterTableFields
	symbologyEdit		
	sharingImport		sharingExport

5.9 Plugin

5.9.1 Processing

Icona	Sostituzione	Icona	Sostituzione
	processingAlgorithm		processingModel
	processingHistory		processingResult
	menu		
	processSelected		editHelpContent
	saveAsPython		modelOutput
	qgsProjectFile		addToProject
	fieldInteger		
	meanCoordinates		extractLayerExtent
	selectRandom		vectorGrid
	convexHull		buffer
	intersect		union
	symmetricalDifference		clip
	difference		dissolve
	checkGeometry		exportGeometry
	delaunay		centroids
	polygonToLine		extractVertices
	lineToPolygon		nearestNeighbour
	splitLayer		heatmap
	showRasterCalculator		showMeshCalculator
	regularPoints		addGeometryAttributes
	basicStatistics		uniqueValues
	collect		simplify_2
	createGrid		distanceMatrix

continues on next page

Tabella 5.5 – continua dalla pagina precedente

Icona	Sostituzione	Icona	Sostituzione
	lineIntersections		mergeLayers
	sumPoints		sumLengthLines
	randomPointsInPolygons		randomPointsWithinPolygon
	randomPointsOnLines		randomPointsWithinExtent
	multiToSingle		
	grid		tiles
	merge		rasterClip
	contour		proximity
	polygonize		rasterize
	sieve		nearblack
	projectionAdd		projectionExport
	8To24Bits		24To8Bits
	rasterInfo		rasterOverview
	vrt		voronoi
	translate		warp
	iterate		terminal

5.9.2 Varie Plugin di Base

Standard forniti con l'installazione di base, ma non caricate con l'installazione iniziale

Icona	Sostituzione	Icona	Sostituzione
	showPluginManager		installPluginFromZip
	pythonFile		runConsole
	showEditorConsole		clearConsole
	offlineEditingCopy		offlineEditingSync
	plugin		metasearch
	geometryChecker		topologyChecker
	fromSelectedFeature		sqlQueryBuilder

5.9.3 Integrazione con GRASS

Icona	Sostituzione	Icona	Sostituzione
	grassLogo		grassRegion
	grassTools		grassNewMapset
	grassOpenMapset		grassCloseMapset