

Contribuer au  
logiciel libre  
sans être expert

1 - Qui suis-je ?

# 1.1 - Qui suis-je ?

Vincent Poulailleau

<https://github.com/vpoulailleau>

<https://vpoulailleau.wordpress.com/>

[vpoulailleau@gmail.com](mailto:vpoulailleau@gmail.com)

<https://www.linkedin.com/in/vpoulailleau/>

# L'essentiel : Qui suis-je ?

Vincent Poulailleau

<https://www.linkedin.com/in/vpoulailleau/>

# 2 - Le logiciel libre

## 2.1 - C'est quoi ?

## 2.1.1 - Le logiciel libre ?

Liberté d'utiliser le logiciel, pour quelque usage que ce soit

Liberté d'étudier le fonctionnement du programme, et de l'adapter à vos propres besoins

Liberté de redistribuer des copies à tout le monde

Liberté d'améliorer le programme et de publier vos améliorations

<https://www.gnu.org/philosophy/free-sw.fr.html>

## 2.1.2 - Le logiciel libre

Libre  $\neq$  gratuit

Libre  $\neq$  open source



## 2.1.3 - Examples

Linux

Firefox

Chromium, VSCodium

LibreOffice

VLC

## 2.1.4 - Examples

Gimp

Filezilla

PostgreSQL

Kodi

CPython

## 2.1.5 - Contre exemples

Microsoft Windows

Microsoft Office

Google Chrome

Apple iOS

Adobe Photoshop

# L'essentiel : C'est quoi ?

Un logiciel libre respecte 4 libertés fondamentales : utiliser, étudier, redistribuer, modifier

Libre  $\neq$  gratuit

Libre  $\neq$  open source

## 2.2 - Type de contributions

## 2.2.1 - Code

### Écriture de code

Ajout de fonctionnalité

Correction de bug

Pas toujours facile

## 2.2.2 - Autres contributions

Faire de la pub !

Faire un don

Créer un logo

Trier des rapports de bug, reproduire les bugs

Faire des relectures

Traduire un logiciel

Traduire une documentation

...

# L'essentiel : Type de contributions

Écriture de code

Traduire une documentation

Et tant d'autres façons



# L'essentiel : Le logiciel libre

§ C'est quoi ?

§ Type de contributions

# 3 - Notre projet

# 3.1 - Traduction de la documentation de Python

## 3.1.1 - CPython

CPython est l'interpréteur classique de Python

```
python3.8 mon_script.py
```

## 3.1.2 - CPython

Le projet CPython :

<https://github.com/python/cpython>

350 000 lignes de C

800 000 lignes de Python

230 000 lignes de reStructuredText

C'est la documentation

Source LOC : <https://talkpython.fm/episodes/transcript/240/a-guided-tour-of-the-cpython-source-code>

## 3.1.3 - Documentation

### Traduction de la documentation

Par des bénévoles

Raymond Hettinger : « Mon salaire a quasiment été doublé tous les ans par Guido »

<https://docs.python.org/3/> devient  
[https://docs.python.org/\*\*fr\*\*/3/](https://docs.python.org/fr/3/)

# L'essentiel : Traduction de la documentation de Python

Notre projet : proposer des traductions

# 3.2 - reStructuredText



# 3.2.1 - reStructuredText

Documentation dans un format textuel

Quelques signes de ponctuation particuliers

Pour faire un titre

Pour faire des hyper-liens

Pour mettre en italique

...

<http://docutils.sourceforge.net/docs/user/rst/quickref.html>

<http://docutils.sourceforge.net/docs/user/rst/cheatsheet.txt>

Appelé aussi RST

## 3.2.2 - RST

Du texte normal

Ceci est un texte tout à fait habituel,  
un retour à la ligne ne commence pas  
un autre paragraphe.

Ceci est un autre paragraphe, un unique  
saut de ligne est équivalent à une espace.

## 3.2.3 - RST

### Des titres

Mon titre principal

=====

Première partie

-----

Un joli paragraphe.

Deuxième partie

-----

Un autre paragraphe.

## 3.2.4 - RST

### Emphase

Un texte avec `*emphase*` est généralement en `*italique*`.

Un texte avec une `**emphase forte**` est généralement en `**gras**`.

# 3.2.5 - RST

## Des liens

Ceci est un lien : `https://en.wikipedia.org/wiki/ReStructuredText`

Ou encore : ``Lien Wikipédia <https://en.wikipedia.org/wiki/ReStructuredText>`_`

Ou : `j'adore Python_ au point de faire un lien vers Python_`

`.. _Python: http://www.python.org/`

## 3.2.6 - RST

### Police à chasse fixe

Ce ``texte`` est interprété selon la configuration du lecteur reStructuredText.

Ce ```texte``` est dans une police à chasse fixe (qui rappelle le code).

## 3.2.7 - RST

### Balises particulières

`:term: `un mot avec une définition``

`:term: `texte à afficher <un mot avec une définition>``

`:class: `HTTPServer``

`:exc: `KeyError``

Et d'autres avec une syntaxe similaire.

## 3.2.8 - RST

### Blocs

Ceci introduit un bloc ::

Ce texte est dans le bloc,  
magnifique, non ?

Notez l'espace insécable avant le ``::``

.. code-block:: python

```
print("Vive la doc en français !")
```



## 3.2.9 - RST

Je dois connaître tout ça ???

Non !

Par contre il ne faut pas casser le balisage existant

# L'essentiel : reStructuredText

Documentation dans un format textuel

Un retour à la ligne ne crée pas de paragraphe

Pour mettre en italique, entourer avec des \*

Blocs introduits par ::

Ne pas casser le balisage existant

## 3.3 - Outils

# 3.3.1 - Navigateur internet

Il vous faut un navigateur internet

Et un accès à internet (on peut faire des choses sans, mais pas tout)

<https://github.com/python/python-docs-fr>

Accord de contribution à la documentation (à lire absolument)

« vous acceptez de fournir vos améliorations [...] sous licence CC0 »

« vous pouvez demander à ce que votre contribution à la documentation soit publiquement reconnue »

<https://github.com/python/python-docs-fr/blob/3.8/CONTRIBUTING.rst>

Source d'inspiration de ce document

## 3.3.2 - GitHub

Il vous faut un compte GitHub

Enfin, c'est la méthode recommandée

<https://github.com/join>

## 3.3.3 - Git

Il vous faut un client git pour interagir avec GitHub

<https://git-scm.com/book/fr/v2/D%C3%A9marrage-rapide-Installation-de-Git>

<https://git-scm.com/book/fr/v2/D%C3%A9marrage-rapide-Param%C3%A9trage-%C3%A0-la-premi%C3%A8re-utilisation-de-Git>

## 3.3.4 - Un éditeur

Les fichiers à traduire sont des fichiers textuel avec l'extension « .po »

Il nous faut un éditeur

Éditeur de texte habituel

Éditeur de fichier « .po »

En ligne : <https://localise.biz/free/poeditor>

À installer : <https://poedit.net/>

Édition > Préférences > Informations sur le traducteur

<https://github.com/python/python-docs-fr/blob/3.8/TRANSLATORS>

## 3.3.5 - Padpo

<https://github.com/vpoulailleau/padpo>

```
pip install padpo --upgrade
```

Dans un environnement virtuel (Python 3.7 et plus)

Vérifie grammaire, orthographe, espaces  
insécables...



# L'essentiel : Outils

Navigateur internet

Accord de contribution

GitHub

Git

Éditeur

Padpo

# 3.4 - Contribution

## 3.4.1 - Git

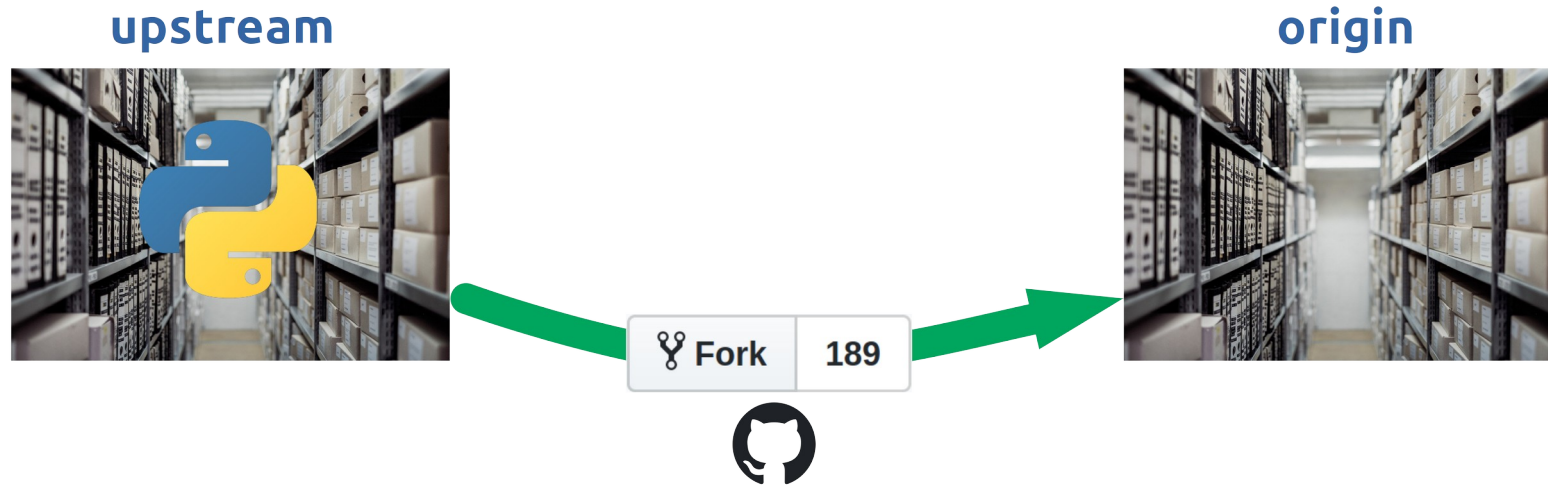
3.4.1.1 - La première fois

# 3.4.1.1.1 - Git

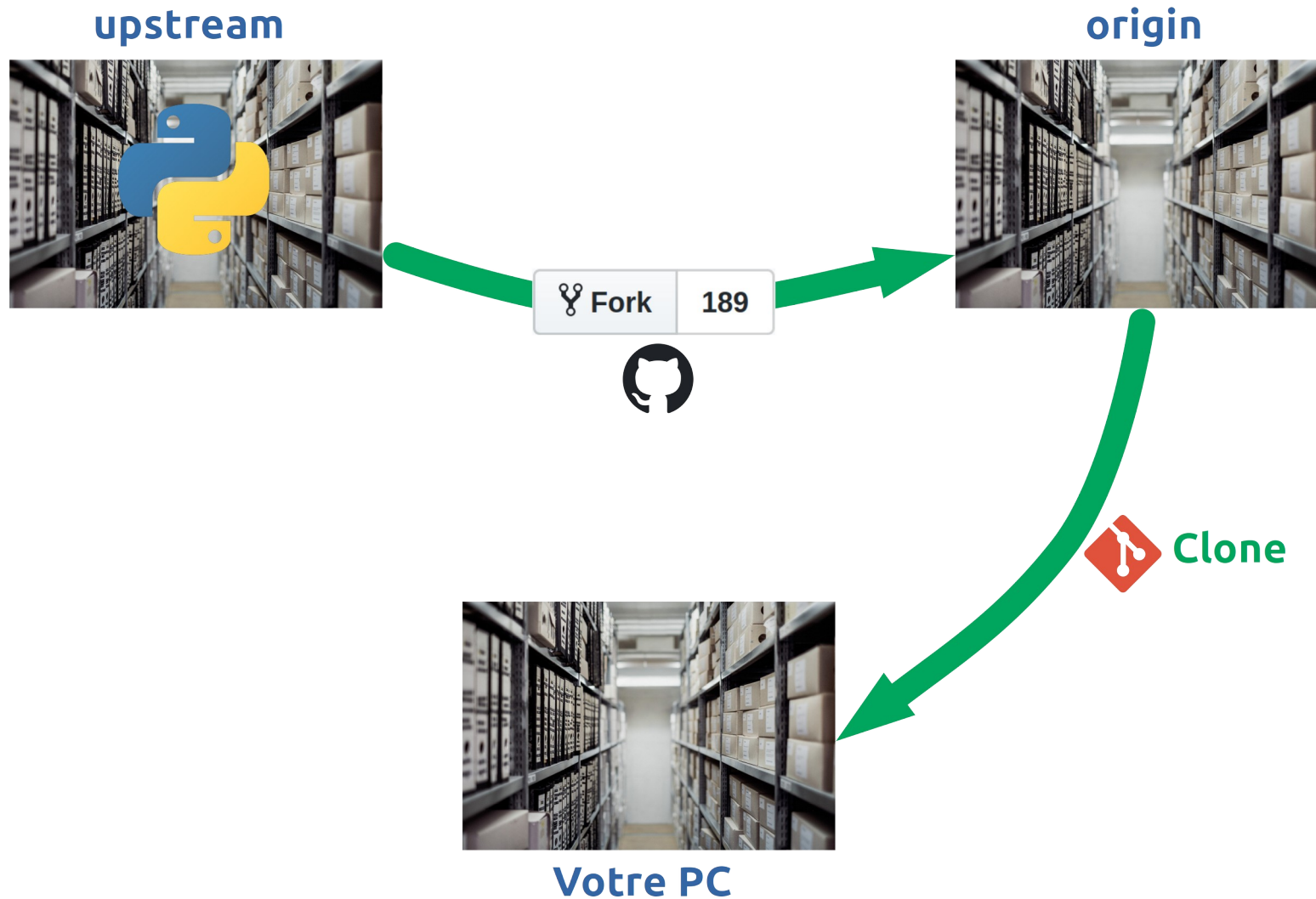
upstream



## 3.4.1.1.2 - Git



# 3.4.1.1.3 - Git



# L'essentiel : La première fois

Fork

Clone



3.4.1.2 - À chaque  
nouvelle traduction

# 3.4.1.2.1 - Git

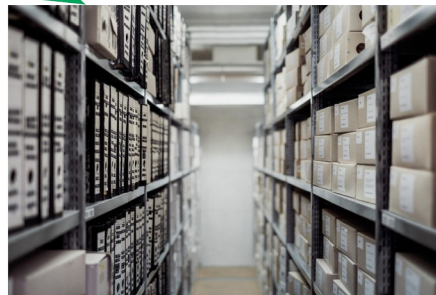
upstream



origin



Fetch + branche



Votre PC

# 3.4.1.2.2 - Git

upstream



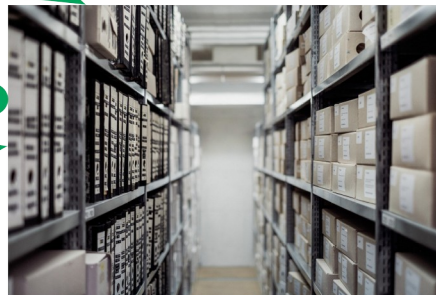
origin



Fetch + branche

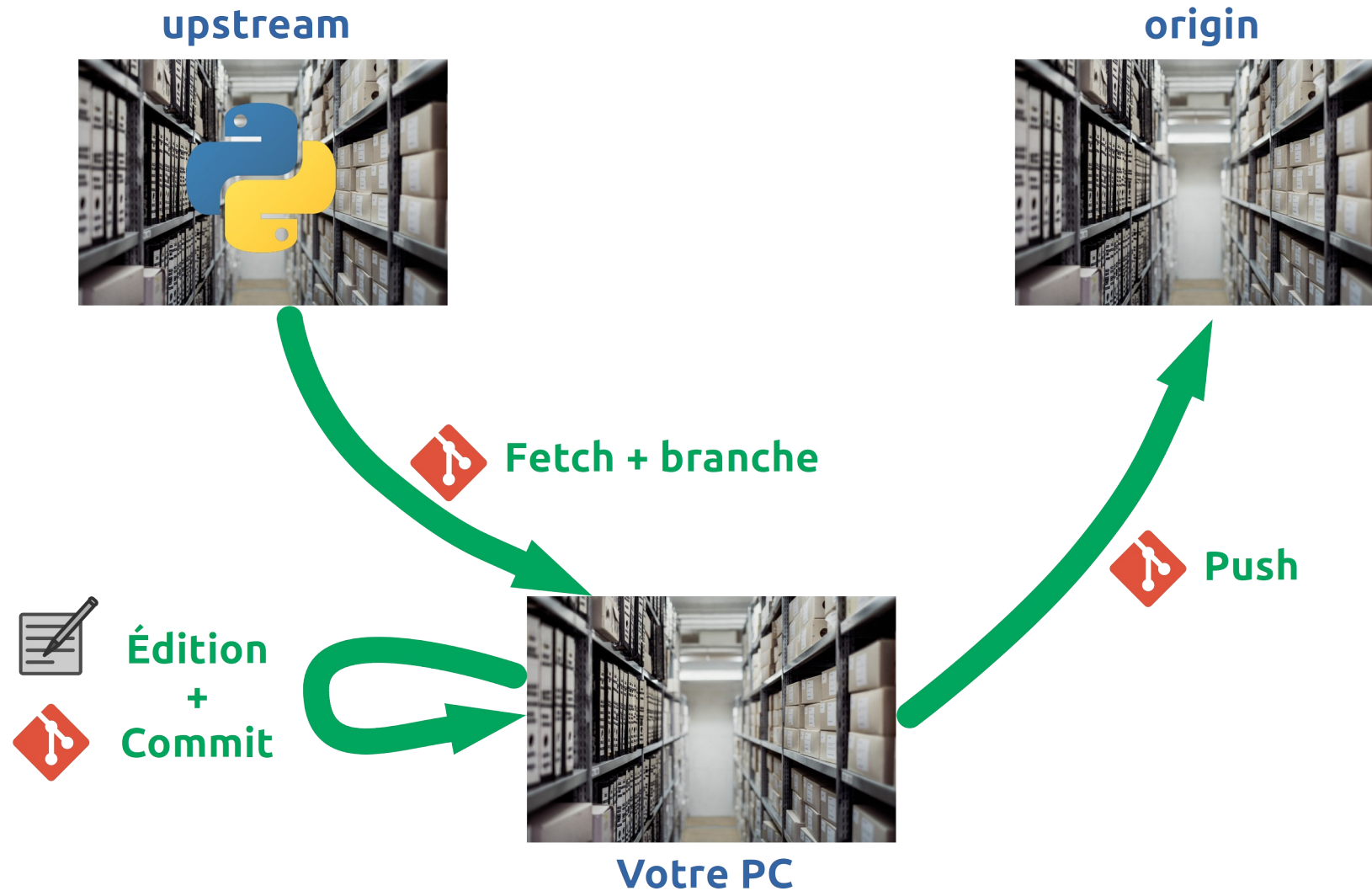


Édition  
+  
Commit

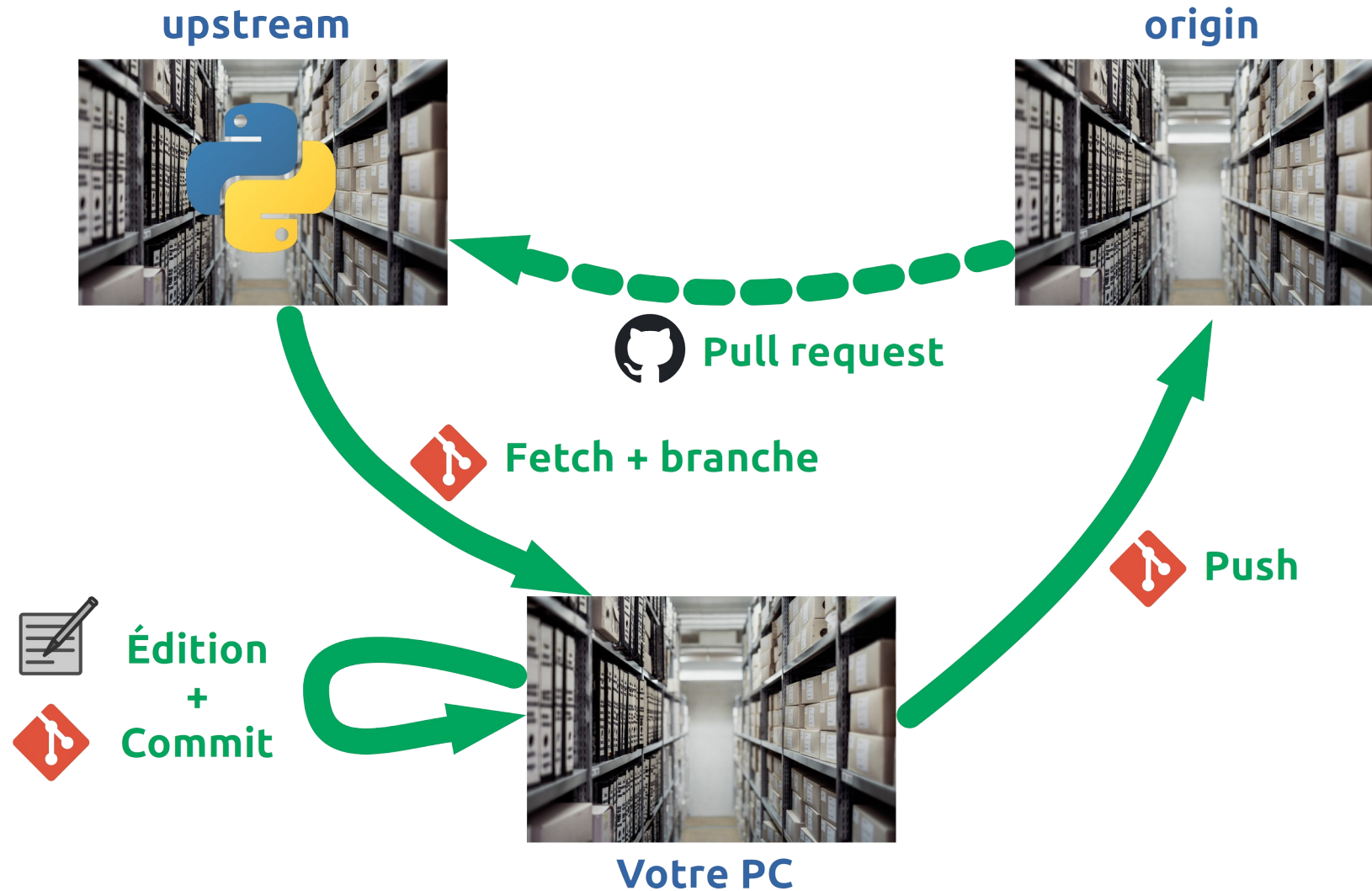


Votre PC

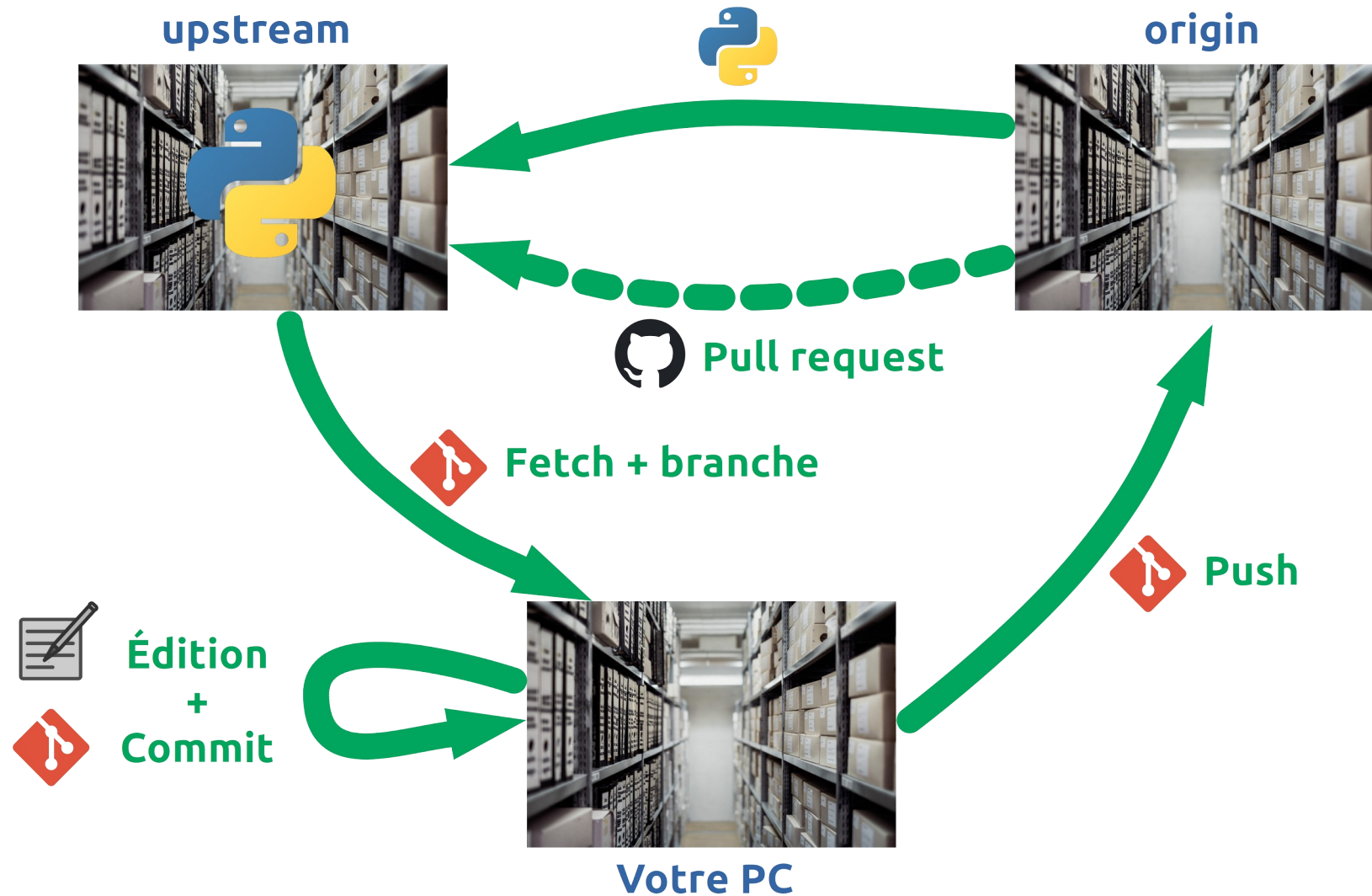
# 3.4.1.2.3 - Git



# 3.4.1.2.4 - Git



# 3.4.1.2.5 - Git





# L'essentiel : À chaque nouvelle traduction

Fetch

Branch

Édition

Commit

Push

Pull request

Fusion

# L'essentiel : Git

§ La première fois

§ À chaque nouvelle traduction



## 3.4.2 - Traduire

## 3.4.2.1 - Quoi traduire ?

Commencez petit !

Prenez quelque chose qui vous plaît

Voulant savoir comment fonctionne `pathlib`, j'ai traduit la documentation

`potodo`

Montre les réservations (diapo suivante)

Montre les fuzzy

Lançable avec `make todo`

Ou source `~/venvs/python-docs-i18n/bin/activate` puis `potodo`

## 3.4.2.2 - Réserve

Annoncez sur quoi vous travaillez

Pour éviter que quelqu'un fasse le même travail que vous

Oralement, pour ceux avec vous ici

***Et*** en créant un ticket sur GitHub, pour le reste du monde

<https://github.com/python/python-docs-fr/issues/new/choose>

## 3.4.2.3 - Git

```
git fetch upstream
```

```
git checkout -b  
ma_nouvelle_branche_a_moi  
upstream/3.8
```

Choisissez un nom de branche qui dit ce sur quoi vous travaillez, pas `ma_nouvelle_branche_a_moi`

## 3.4.2.4 - Éditer le fichier

Éditez le fichier avec l'éditeur de votre choix

Pensez à sauvegarder !

## 3.4.2.5 - Vérifiez votre traduction

```
padpo -i mon_fichier.po
```

Adaptez le nom du fichier, par exemple `library/json.po`

Signale aussi les problèmes sur les parties non modifiées par vous

Corrigez votre partie, et éventuellement les autres parties

```
make verifs
```

## 3.4.2.6 - git

```
git add mon_fichier.po
```

```
git commit -m "Traduction de  
mon_fichier.po"
```

```
git push -u origin
```

Donne un lien pour créer la pull request

Mettre en commentaire de pull request : Fix #9999 où 9999 est le numéro du ticket de réservation

## 3.4.2.7 - Relecture

Votre pull request va être relue

Vous pouvez relire les pull requests des autres

Soyez attentifs aux remarques (par mail, ou par la page de la pull request)

Apportez les corrections si nécessaires (diapo suivante)



## 3.4.2.8 - Correction

Pour prendre en compte les remarques

Par l'interface web pour appliquer les suggestions

Sinon par git

```
git checkout ma_nouvelle_branche_a_moi
```

```
git pull
```

Édition du document

```
git add mon_fichier.po
```

```
git commit -m "prise en compte des remarques"
```

```
git push
```

# L'essentiel : Traduire

Faire une petite traduction sur un fichier non déjà réservé

Réservez le fichier

Mettez votre dépôt à jour

Créez une branche

Éditer le fichier

Vérifiez votre traduction

Publier votre traduction

Faites une pull request

Soyez attentifs aux remarques de relecture

Prenez en compte les remarques

## 3.4.3 - Conseils

## 3.4.3.1 - Anglais

Si vous devez laisser un mot anglais, il faut le mettre en italique

*\*un mot\** en reStructuredText

## 3.4.3.2 - Futur

if you do this, it will produce that

Devient : si vous faites ceci, il se produit cela

Pas de futur, sauf concordance des temps

### 3.4.3.3 - should

you should indique généralement une obligation dans la documentation

Ne pas traduire par vous devriez

Traduire par vous devez

## 3.4.3.4 - Caractères spéciaux

Il faut traduire en français correct

Utilisation de É, Ç, «», et autres espaces insécables

<https://github.com/python/python-docs-fr/blob/3.8/CONTRIBUTING.rst#les-caract%C3%A8res-sp%C3%A9ciaux>

Utilisez un clavier BÉPO ;-)

Utilisez un clavier AZERTY à la nouvelle norme NF (qui inclut aussi BÉPO)

Installez un logiciel sinon

## 3.4.3.5 - Guillemets

"Hi" devient « Salut »

Avec des espaces insécables

`print("Hi")` devient `print("Hi")`



## 3.4.3.6 - ::

:: introduit un bloc en reStructuredText

S'il est à la fin d'une phrase, il faut une espace insécable avant

Il va se passer ceci ::

## 3.4.3.7 - Glossaire

En cas de doute sur une traduction

<https://github.com/python/python-docs-fr/blob/3.8/CONTRIBUTING.rst#glossaire>

Le glossaire Python est déjà traduit :  
<https://docs.python.org/fr/3/glossary.html>

Regardez la traduction dans Wikipédia

pogrep

## 3.4.3.8 - Aide

Cela fait beaucoup d'informations en peu de temps

Demandez de l'aide, aidez vous

N'ayez pas peur, tout ce que vous ferez sera relu

Par des gens sympathiques !

Et peut-être un jour aussi par des robots !

## 3.4.3.9 - À vous

C'est parti : à vous !

Un petit youpi par pull request créée

Un gros youpi par pull request acceptée et fusionnée

# L'essentiel : Conseils

Mots anglais en italique

Pas de futur (no future...)

you should => vous devez

Attention aux caractères spéciaux

Attention aux guillemets

Une espace insécable avant ::

Des glossaires existent

Demandez de l'aide

N'ayez pas peur

# L'essentiel : Contribution

§ Git

§ Traduire

§ Conseils

# L'essentiel : Notre projet

§ Traduction de la documentation de Python

§ reStructuredText

§ Outils

§ Contribution