

1. Introduction

Git est un outil de gestion de versions distribue qui permet de suivre les modifications d'un projet, de revenir en arriere en cas d'erreur et de collaborer a plusieurs efficacement. Nous avons travaille sur un projet Python, avec deux branches, puis nous avons fusionne les branches.

Installation de Git sous Ubuntu/Debian (Machine virtuelle, connexion internet, acces root)

```
sudo apt update  
sudo apt install git
```

Verification:

```
git --version
```

```
(kuroro@kuroroLucifer)~[/fac/SE/projet-git-demo]  
$ git --version  
git version 2.47.2  
  
(kuroro@kuroroLucifer)~[/fac/SE/projet-git-demo]  
$ █
```

2. Configuration de Git

```
git config user.name="Votre Nom"  
git config user.email="votre.email@gmail.com"
```

```
(kuroro@kuroroLucifer)~[/fac/SE/projet-git-demo]  
$ git config --global user.email "felixkiffance@gmail.com"  
  
(kuroro@kuroroLucifer)~[/fac/SE/projet-git-demo]  
$ git config --global user.name "yonkotech"  
  
(kuroro@kuroroLucifer)~[/fac/SE/projet-git-demo]  
$ █
```

Voir la configuration

```
git config --list
```

```
(kuroro@kuroroLucifer)-[~/fac/SE/projet-git-demo]
$ git config --list
user.name=yonkotech
user.email=felixkiffance@gmail.com
core.repositoryformatversion=0
core.filemode=true
core.bare=false
core.logallrefupdates=true

(kuroro@kuroroLucifer)-[~/fac/SE/projet-git-demo]
$ █
```

3. Utilisation de base

Initialiser un depot :

git init

```
(kuroro@kuroroLucifer)-[~/fac/SE/projet-git-demo]
$ git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/kuroro/fac/SE/projet-git-demo/.git/

(kuroro@kuroroLucifer)-[~/fac/SE/projet-git-demo]
$ █
```

Ajouter un fichier au suivi

git add nom_du_fichier

Sauvegarder les changements:

git commit -m "Message du commit"

```

(kuroro@kuroroLucifer)-[~/fac/SE/projet-git-demo]
$ echo "# Projet Git Demo" > README.md

(kuroro@kuroroLucifer)-[~/fac/SE/projet-git-demo]
$ git add README.md

(kuroro@kuroroLucifer)-[~/fac/SE/projet-git-demo]
$ git commit -m "Ajout du README"

[master (root-commit) 4617b56] Ajout du README
1 file changed, 1 insertion(+)
create mode 100644 README.md

(kuroro@kuroroLucifer)-[~/fac/SE/projet-git-demo]
$ █

```

Voir l'historique

git log

```

(kuroro@kuroroLucifer)-[~/fac/SE/projet-git-demo]
$ git log --oneline --graph
* 0c83311 (HEAD -> master, feature) Ajout du script hello.py
* 4617b56 Ajout du README

(kuroro@kuroroLucifer)-[~/fac/SE/projet-git-demo]
$ ls
hello.py  README.md

(kuroro@kuroroLucifer)-[~/fac/SE/projet-git-demo]
$ █

```

4. Gestion des branches

git branch nom_de_branche

git checkout nom_de_branche

```

(kuroro@kuroroLucifer)-[~/fac/SE/projet-git-demo]
$ git branch feature

(kuroro@kuroroLucifer)-[~/fac/SE/projet-git-demo]
$ git checkout feature
Switched to branch 'feature'

(kuroro@kuroroLucifer)-[~/fac/SE/projet-git-demo]
$ echo "print('Hello world')" > hello.py

(kuroro@kuroroLucifer)-[~/fac/SE/projet-git-demo]
$ git add hello.py

(kuroro@kuroroLucifer)-[~/fac/SE/projet-git-demo]
$ git commit -m "Ajout du script hello.py"
[feature 0c83311] Ajout du script hello.py
1 file changed, 1 insertion(+)
create mode 100644 hello.py

(kuroro@kuroroLucifer)-[~/fac/SE/projet-git-demo]
$ ls
hello.py  README.md

(kuroro@kuroroLucifer)-[~/fac/SE/projet-git-demo]
$ 

```

Fusion\

git merge nom_de_branche

```

(kuroro@kuroroLucifer)-[~/fac/SE/projet-git-demo]
$ git checkout master
Switched to branch 'master'

(kuroro@kuroroLucifer)-[~/fac/SE/projet-git-demo]
$ git merge feature
Updating 4617b56..0c83311
Fast-forward
 hello.py | 1 +
1 file changed, 1 insertion(+)
create mode 100644 hello.py

(kuroro@kuroroLucifer)-[~/fac/SE/projet-git-demo]
$ █

```

5. Travailler avec GitHub

Créer un nouveau dépôt GitHub

Aller sur <https://github.com>

Créer un nouveau repo

Copier l'URL du repo

Conclusion

Ce projet nous a permis de maîtriser les bases de Git sous Linux et de comprendre l'importance du versionnement collaboratif. Git est un outil incontournable dans toute démarche de développement moderne.