

Introduction à Git

Git est un système de contrôle de version décentralisé qui permet de suivre les modifications d'un projet, de collaborer efficacement et de conserver l'historique des changements.

1. Afficher les commandes disponibles

```
git
```

Affiche la liste des commandes principales de Git.

2. Configurer l'identité utilisateur (globalement)

```
git config --global user.name 'ezynsegnane'  
git config --global user.email 'ezynsegnane@gmail.com'
```

Enregistre le nom et l'e-mail globalement sur la machine.

3. Vérifier la configuration

```
git config user.name  
git config user.email  
git config --list
```

4. Initialiser un dépôt Git

```
git init ourproject
```

Crée un dépôt local dans le dossier ourproject.

5. Accéder et lister les fichiers

```
ls  
cd ourproject  
ls -a
```

ls -a affiche les fichiers cachés comme .git.

6. Vérifier l'état du dépôt

```
git status
```

Montre les fichiers modifiés, ajoutés ou non suivis.

7. Créer un fichier

```
touch python.py
```

Crée un fichier vide nommé `python.py`.

8. Ajouter un fichier à l'index

```
git add python.py
```

Place le fichier dans le *staging area*.

9. Créer un commit

```
git commit -m 'Cr ation du fichier python.py'
```

Enregistre les changements avec un message.

10. Afficher l'historique

```
git log
```

11. Revenir à un commit précédent

```
git checkout <id_du_commit>
```

Permet de naviguer dans l'historique (lecture seule).

12. Gérer les branches

```
git branch          # liste les branches
git branch ssd      # cr er une branche
git checkout ssd     # basculer vers ssd
git checkout master  # revenir  master
```

13. Fusionner une branche

```
git merge ssd
```

Fusionne la branche `ssd` dans `master`.

GitHub : travailler à distance

14. Vérifier la connexion distante

```
git remote -v
```

15. Lier à un dépôt GitHub

```
git remote add origin https://github.com/ezynsegnane/gitRibo.git
```

16. Pousser le dépôt local

```
git push -u origin master  
# ou forcer :  
# git push origin master --force
```

17. Récupérer depuis GitHub

```
git pull origin master
```

18. Cloner un dépôt GitHub

```
git clone https://github.com/utilisateur/nom-du-repo.git
```

Copie un dépôt distant sur votre machine locale.

19. Supprimer un fichier versionné

```
git rm fichier.py
```

Supprime un fichier du projet et de l'historique Git.

20. Réinitialiser un fichier modifié

```
git checkout -- fichier.py
```

Annule les modifications locales d'un fichier.

21. Réinitialiser le dernier commit (mais garder les fichiers)

```
git reset --soft HEAD~1
```

Annule le commit tout en conservant les fichiers dans le stage.

22. Réinitialiser complètement (commit + fichiers)

```
git reset --hard HEAD~1
```

Annule totalement le dernier commit et les fichiers.

23. Sauvegarder temporairement les modifications (stash)

```
git stash
```

Met de côté les modifications non commit.

24. Restaurer les modifications mises en stash

```
git stash apply
```

25. Afficher les différences dans les fichiers

```
git diff
```

Compare les modifications entre le fichier et l'index.

26. Voir les détails d'un commit spécifique

```
git show <id_commit>
```

27. Rejouer les commits d'une branche sur une autre (rebase)

```
git rebase nom_branche
```

Réécrit l'historique de manière linéaire.