

1. Installation de Git

Tout d'abord, assurez-vous que Git est installé sur votre système. Vous pouvez le vérifier en ouvrant un terminal ou une invite de commandes et en tapant :

```
git --version
```

Si Git n'est pas installé, téléchargez-le et installez-le depuis [le site officiel de Git](#).

2. Configuration initiale de Git

Après l'installation, ouvrez un terminal et configurez votre nom d'utilisateur et votre adresse e-mail. Ces informations seront utilisées dans vos futurs commits :

```
git config --global user.name "Votre Nom"
git config --global user.email "votreemail@example.com"
```

3. Initialiser un nouveau dépôt Git

Pour commencer à suivre un projet avec Git, naviguez dans le dossier de votre projet et utilisez :

```
cd chemin/vers/votre/projet
git init
```

Cela crée un nouveau sous-dossier `.git` contenant tous les fichiers nécessaires au fonctionnement de Git.

4. Suivi des fichiers et premiers commits

Pour commencer à suivre un fichier, utilisez la commande `git add`. Pour ajouter tous les fichiers d'un projet :

```
git add .
```

Pour enregistrer vos changements dans le dépôt, utilisez :

```
git commit -m "Votre message de commit"
```

Envoyer votre code sur GitHub :

Pour la première fois, vous devrez définir la branche principale et pousser vos changements :

```
git push -u origin main
```

Pour les envois suivants, utilisez simplement :

```
git push
```

5. Vérifier le statut de vos fichiers

Utilisez `git status` pour voir quels fichiers sont suivis et lesquels ont des modifications :

```
git status
```

6. Voir l'historique des commits

Pour voir l'historique des commits, utilisez :

```
git log
```

7. Branches avec Git

Pour créer une nouvelle branche :

```
git branch nom_de_la_branche
```

Pour basculer entre les branches :

```
git checkout nom_de_la_branche
```

8. Fusionner des branches

Pour fusionner une branche dans votre branche actuelle :

```
git merge nom_de_la_branche
```

9. Récupérer un projet

Pour récupérer une version de projet correspondant à un commit spécifique

```
git reset --hard numero_du_commit
```

10. Revenir à un commit antérieur (sans supprimer les commits suivants)

Si tu souhaites revenir à un commit antérieur tout en conservant les commits suivants (et pouvoir y revenir plus tard), tu peux utiliser la commande `git checkout` :

```
git checkout numero_du_commit
```

Cela te permet de naviguer vers un commit précédent, mais sans changer l'historique de la branche actuelle.

11. Revenir à la version du commit en cours en annulant toutes les modifications non commises

Pour annuler toutes les modifications non commises et revenir à l'état du commit en cours, tu peux utiliser la commande suivante :

```
git checkout -- .
```

Cela rétablit tous les fichiers modifiés (mais non ajoutés ou commis) à leur version du commit actuel.

12. Récupérer un fichier supprimé

Si tu as supprimé un fichier et que tu veux le récupérer (avant d'avoir fait un commit), tu peux utiliser cette commande :

```
git checkout -- chemin/vers/le/fichier
```

Cela restaure le fichier supprimé à son état dans le dernier commit.

Si tu veux récupérer des fichiers supprimés à partir d'un commit précédent, utilise la commande suivante, en remplaçant `numero_du_commit` par le numéro du commit où le fichier existait encore :

```
git checkout numero_du_commit -- chemin/vers/le/fichier
```

13. Rattacher un projet local à un dépôt GitHub

1. **Créer un dépôt sur GitHub** : Allez sur GitHub, connectez-vous, et créez un nouveau dépôt (repository).
2. **Lier votre dépôt local au dépôt GitHub** :

```
git remote add origin https://github.com/votre_nom_d_utilisateur/nom_du_dépôt.git
```

14 Conclusion

Voici les commandes Git fondamentales, de l'initialisation d'un dépôt à la fusion de branches et au rattachement d'un projet à GitHub. Git est un outil très puissant avec beaucoup plus de fonctionnalités et de nuances. Je vous encourage à expérimenter et à apprendre davantage pour maîtriser pleinement cet outil essentiel au développement logiciel.

15 Pour aller plus loin ...

- Site officiel du projet Git : <https://git-scm.com/>
- La documentation officielle complète sous la forme d'un livre : <https://git-scm.com/book/fr/v2>
- Une série de tutoriels/vidéos sur Git par Grafikart : <https://grafikart.fr/tutoriels/git>