

# Git, une introduction

## Introduction à Git

- Description de Git : Git est un système de contrôle de version distribué largement utilisé pour la gestion des projets de développement de logiciels. Il permet de suivre les modifications apportées aux fichiers, de collaborer avec d'autres développeurs et de revenir à des versions antérieures du code.
- Mots et concepts clés :
  - Dépôt (repository) : Un dépôt Git est l'endroit où tous les fichiers et historiques de version sont stockés.
  - Commit : Un commit représente une version spécifique du code avec une description des modifications apportées.
  - Branche (branch) : Une branche est une ligne de développement indépendante permettant de travailler sur des fonctionnalités ou des corrections de bugs sans affecter la branche principale.
  - Fusion (merge) : La fusion est le processus d'intégration des modifications d'une branche dans une autre.
  - Clone : Cloner un dépôt Git signifie créer une copie locale complète du dépôt distant sur votre machine.
  - Pousser (push) : Pousser des modifications signifie envoyer les commits locaux vers le dépôt distant.
  - Récupérer (pull) : Récupérer les modifications signifie obtenir les dernières modifications du dépôt distant et les fusionner avec votre copie locale.

## Les commandes Git le plus communes

- `git add` : Ajoute des fichiers à l'index pour les inclure dans le prochain commit.
- `git commit` : Crée un nouveau commit avec les modifications actuelles de l'index et ajoute un message de description.
- `git push` : Envoie les commits locaux vers le dépôt distant spécifié.
- `git checkout` : Permet de basculer entre les branches existantes ou de créer une nouvelle branche.
- `git branch` : Affiche, crée ou supprime des branches.
- `git merge` : Fusionne les modifications d'une branche dans une autre branche.

Voici quelques options couramment utilisées pour ces commandes :

- `git add` :
  - `-A` : Ajoute tous les fichiers modifiés, supprimés et nouvellement créés à l'index.
  - `-p` : Permet une sélection interactive des modifications à ajouter à l'index.

- `git commit` :
  - `-m "message"` : Spécifie un message de commit en ligne de commande.
  - `-a` : Ajoute automatiquement tous les fichiers modifiés à l'index et crée un commit.
- `git push` :
  - `<remote>` : Spécifie le dépôt distant vers lequel pousser les modifications.
  - `<branch>` : Spécifie la branche locale à pousser vers le dépôt distant.
- `git checkout` :
  - `-b <branch>` : Crée une nouvelle branche et bascule dessus.
- `git branch` :
  - `-a` : Affiche toutes les branches locales et distantes.
  - `-d <branch>` : Supprime une branche locale.
- `git merge` :
  - `<branch>` : Spécifie la branche à fusionner dans la branche actuelle.

## Copie locale et copie à distance du dépôt

- Copie locale : Lorsque vous clonez un dépôt Git, vous créez une copie complète du dépôt sur votre machine locale. Vous pouvez travailler sur cette copie, apporter des modifications, créer des branches et des commits.
- Copie à distance : Le dépôt distant est la version centrale du dépôt qui peut être partagée avec d'autres développeurs. Vous pouvez pousser vos commits locaux vers le dépôt distant pour les partager avec l'équipe. Vous pouvez également récupérer les modifications du dépôt distant et les fusionner dans votre copie locale.

## Exemple d'utilisation de Git en équipe

Supposons que vous travaillez en équipe sur un projet avec les branches suivantes :

- `main` : Branche principale du projet
- `feature` : Branche pour développer une nouvelle fonctionnalité
- `bugfix` : Branche pour corriger des bugs

Voici les étapes pour collaborer et pousser les branches sur le dépôt distant :

1. Clonez le dépôt distant sur votre machine :

```
git clone <url_du_dépôt>
```

2. Créez une nouvelle branche pour travailler sur une fonctionnalité :

```
git checkout -b feature
```

3. Faites des modifications dans la branche **feature** et effectuez un commit :

```
git add <fichiers_modifiés>  
git commit -m "Ajout de la nouvelle fonctionnalité"
```

4. Poussez la branche **feature** sur le dépôt distant :

```
git push origin feature
```

5. Lorsque la fonctionnalité est prête, fusionnez la branche **feature** dans la branche **main** :

```
git checkout main  
git merge feature
```

6. Poussez la branche **main** fusionnée sur le dépôt distant :

```
git push origin main
```

## Sources

Blog : Vous pouvez consulter le blog “Atlassian Git Tutorial” qui propose un tutoriel complet sur Git et ses fonctionnalités : [Lien vers le blog Atlassian Git Tutorial](#)

Documentation officielle de Git : [Lien vers la documentation Git](#)