Cours : L'utilisation des branches dans Git

24/01/2025

1 Introduction

Les branches sont l'une des fonctionnalités les plus puissantes de Git. Elles permettent de travailler sur différentes versions d'un projet sans affecter le code principal, facilitant ainsi le développement parallèle, la gestion de fonctionnalités ou corrections, et l'expérimentation.

2 Qu'est-ce qu'une branche ?

Une branche est un pointeur sur un commit spécifique dans l'historique Git. Elle permet de créer une "copie" du projet sur laquelle on peut travailler indépendamment. La branche par défaut dans Git est appelée `master` (ou `main` dans les versions récentes).

Schéma conceptuel

```
master -> Commit A -> Commit B -> Commit C
```

Si vous créez une nouvelle branche appelée `feature`, elle pointe également sur le dernier commit (C) mais devient indépendante :

```
master  -> Commit A -> Commit B -> Commit C
                          \
feature ------------------> (Nouvelle branche)
```

3 Pourquoi utiliser des branches ?

- **Développement parallèle** : Travailler sur une fonctionnalité ou correction sans perturber le code principal.
- **Collaboration** : Chaque développeur peut travailler sur sa propre branche, puis intégrer les changements.
- **Expérimentation** : Tester de nouvelles idées sans risque d'affecter le projet principal.
- **Organisation** : Isoler les tâches pour un suivi clair.

4 Commandes de base pour gérer les branches

4.1 Créer une branche

```
git branch <nom_de_branche>
```

Exemple :

```
git branch feature-login
```

4.2 Afficher les branches

```
git branch
```

4.3 Basculer entre les branches

```
git checkout <nom_de_branche>
```

Exemple :

```
git checkout feature-login
```

4.4 Créer une branche et basculer dessus (raccourci)

```
git checkout -b <nom_de_branche>
```

4.5 Fusionner une branche

Pour intégrer les modifications d'une branche dans une autre :

1. Basculer sur la branche cible (par exemple, `master`) :

```
    git checkout master

```

2. Fusionner la branche source (par exemple, `feature-login`) :

```
    git merge feature-login

```

4.6 Supprimer une branche

Une fois la branche fusionnée ou obsolète, vous pouvez la supprimer :

```
git branch -d <nom_de_branche>
```

Pour forcer la suppression :

```
git branch -D <nom_de_branche>
```