

Cours Git

Introduction à Git

Git est un système de gestion de versions distribué, très utilisé pour le développement logiciel. Il permet de suivre l'évolution d'un projet, de collaborer à plusieurs, de gérer différentes versions du code et de fusionner des modifications.

Créer un dépôt Git

Pour démarrer avec Git sur un nouveau projet :

- Placez-vous dans le dossier de votre projet :
- `cd /chemin/vers/mon_projet`

📖 Initialisez le dépôt :

- `git init`

Cela crée un dossier caché `.git` qui contient tout l'historique et la configuration du dépôt

Pour cloner un dépôt existant (par exemple depuis GitHub) :

- `git clone https://url_du_depot.git`

Cela crée un dossier local avec tout l'historique du projet

Créer et gérer des branches

Créer une branche à partir de main :

- `git branch nom_branche`

Vous pouvez créer autant de branches que nécessaire, par exemple :

- `git branch feature1`
- `git branch feature2`
- `git branch correction_bug`

Chaque branche est une copie du code à l'instant où elle est créée

Créer et se placer directement sur une nouvelle branche :

- `git checkout -b nom_branche`

Cela crée la branche et vous y positionne immédiatement

Se déplacer entre les branches

Pour lister les branches :

- `git branch`

La branche active est précédée d'un astérisque.

Pour changer de branche :

- `git checkout nom_branche`

Depuis Git 2.23, il est recommandé d'utiliser :

- `git switch nom_branche`

pour plus de clarté

Travailler et pousser ses modifications

Ajouter les fichiers modifiés à l'index :

- `git add fichier.txt`

ou pour tout ajouter :

- `git add .`

Valider les modifications :

- `git commit -m "Message explicite sur les modifications"`

Envoyer la branche sur le dépôt distant :

- `git push origin nom_branche`

Si c'est la première fois que vous poussez cette branche :

- `git push --set-upstream origin nom_branche`

Cela crée la branche sur le dépôt distant et la lie à votre branche locale

Pull Request et gestion de la branche main

Créer une Pull Request (PR) :

- Sur GitHub ou GitLab, après avoir poussé votre branche, cliquez sur "Compare & pull request" ou "New merge request".
- Décrivez vos changements et soumettez la PR pour relecture.

Actualiser la branche main avec les modifications d'une autre branche :

1. Placez-vous sur main :
 - `git checkout main`
2. Récupérez les dernières modifications du dépôt distant :
 - `git pull origin main`
3. Fusionnez la branche de fonctionnalité (après validation de la PR) :
 - `git merge nom_branche`

Ou, si la PR a déjà été fusionnée sur GitHub, il suffit de tirer les changements :

- `git pull origin main`

pour mettre à jour votre branche locale.

Récapitulatif des commandes principales

Action	Commande
Initialiser un dépôt	<code>git init</code>
Cloner un dépôt	<code>git clone URL</code>
Créer une branche	<code>git branch nom_branche</code>
Créer + se placer sur une branche	<code>git checkout -b nom_branche</code> ou <code>git switch -c nom_branche</code>
Lister les branches	<code>git branch</code>
Changer de branche	<code>git checkout nom_branche</code> ou <code>git switch nom_branche</code>
Ajouter des fichiers	<code>git add .</code>
Valider (commit)	<code>git commit -m "message"</code>

Action	Commande
Pousser une branche	<code>git push origin nom_branche</code>
Mettre à jour main	<code>git pull origin main</code>
Fusionner une branche dans main	<code>git merge nom_branche</code>

Conclusion

Avec ces commandes, vous pouvez :

- Initialiser un projet Git,
- Créer et naviguer entre plusieurs branches,
- Travailler et pousser vos modifications,
- Gérer les pull requests et fusionner vos changements sur la branche principale.

Ce sont les bases essentielles pour collaborer efficacement avec Git sur un projet logiciel moderne, en équipe.