

Aide-mémoire Git

1. Initialisation d'un dépôt Git

- ~ `git init` : initialise un nouveau dépôt Git dans le répertoire courant.

Exemple : `git init`

- ~ `git clone URL` : clone un dépôt Git distant dans un nouveau répertoire local.

Exemple : `git clone https://github.com/user/repo.git`

- ~ `git config --global user.name "NOM"` : configure le nom d'utilisateur pour tous les dépôts locaux.

Exemple : `git config --global user.name "John Doe"`

- ~ `git config --global user.email "EMAIL"` : configure l'adresse e-mail pour tous les dépôts locaux (ne pas négliger cette instruction !).

Exemple : `git config --global user.email "johndoe@example.com"`

- ~ `git add` : ajoute tout le répertoire à l'index.

- ~ `git add FILE` : ajoute un fichier modifié ou nouvellement créé à l'index.

Exemple : `git add file.txt`

- ~ `git commit -m "MESSAGE"` : crée un instantané du code source actuel et le stocke dans l'historique des versions de Git.

Exemple : `git commit -m "Ajout d un nouveau fichier"`

2. Gestion des branches Git

- ~ `git branch` : affiche la liste des branches locales.

Exemple : `git branch`

- ✓ `git branch BRANCH` : crée une nouvelle branche à partir de la branche actuelle.

Exemple : `git branch feature`

- ✓ `git checkout BRANCH` : bascule vers une branche spécifique et met à jour le répertoire de travail.

Exemple : `git checkout feature`

- ✓ `git merge BRANCH` : fusionne une branche spécifique dans la branche actuelle.

Exemple : `git merge feature`

- ✓ `git branch -d BRANCH` : supprime une branche locale spécifique.

Exemple : `git branch -d feature`

- ✓ `git push origin BRANCH` : envoie une branche locale vers le dépôt distant.

Exemple : `git push origin feature`

3. Gestion des commits Git

- ✓ `git log` : affiche l'historique des commits pour la branche actuelle.

Exemple : `git log`

- ✓ `git diff COMMIT1 COMMIT2` : affiche les différences entre deux commits spécifiques.

Exemple : `git diff abc123 def456`

- ✓ `git reset COMMIT` : annule tous les commits après le commit spécifié et conserve les modifications en tant que modifications non validées.

Exemple : `git reset abc123`

- ✓ `git revert COMMIT` : annule le commit spécifié en créant un nouveau commit qui annule les modifications apportées par le commit spécifié.

Exemple : `git revert abc123`

- ✓ `git cherry-pick COMMIT` : applique les modifications apportées par le commit spécifié à la branche actuelle.

Exemple : `git cherry-pick abc123`

- ✓ `git bisect` : utilise une recherche binaire pour trouver le commit qui a introduit un bogue.

4. Gestion des conflits de fusion

- ✓ `git merge --abort` : annule une fusion en cours et restaure l'état précédent de la branche actuelle.

Exemple : `git merge --abort`

- ✓ Résoudre manuellement les conflits de fusion en éditant les fichiers en conflit et en utilisant les commandes `git add` et `git commit` pour marquer les conflits comme résolus.
- ✓ Utiliser des outils tiers tels que **KDiff3** ou **Meld** pour résoudre les conflits de fusion.

5. Gestion des tags Git

- ✓ `git tag` : affiche la liste des tags existants pour la branche actuelle.

Exemple : `git tag`

- ✓ `git tag TAG` : crée un nouveau tag pour le commit actuel.

Exemple : `git tag v1.0.0`

~ `git show TAG` : affiche les informations sur un tag spécifique.

Exemple : `git show v1.0.0`

~ `git push origin TAG` : envoie un tag local vers le dépôt distant.

Exemple : `git push origin v1.0.0`

~ `git tag -d TAGNAME` : supprime le tag `TAGNAME`.