

Résumé complet des commandes Git (avec explications)

1. Vérifier les branches

```git branch```

Montre les branches locales (sur ton ordinateur).

La branche active a une étoile `*`.

```git branch -a```

Liste les branches locales + celles présentes sur GitHub (`origin/...`).

2. Changer de branche

```git switch nom_branch```

Change la branche active sans risque.

```git checkout nom_branch```

Ancienne méthode (encore utilisée mais moins recommandée).

3. Créer et se déplacer dans une nouvelle branche

```git switch -c nouvelle_branch```

Crée une nouvelle branche puis se place dessus.

```git checkout -b nouvelle_branch```

Ancienne méthode équivalente.

4. Envoyer une nouvelle branche sur GitHub

```git push -u origin nom_branch```

Explications :

- **```git push```** : envoie la branche vers GitHub
- **```-u```** : configure un suivi automatique (tracking)
- **```origin```** : nom du dépôt distant par défaut
- **```nom_branch```** : la branche locale à envoyer

Une fois le -u fait, un simple **`git push`** suffit.

5. Pousser tes modifications (push)

`git push`

Envoie tes commits vers GitHub.

Fonctionne sans argument si la branche est déjà suivie.

6. Récupérer les modifications du serveur (pull)

`git pull`

Télécharge les dernières modifications de la branche sur GitHub et les fusionne localement.

7. Ajouter des fichiers avant commit

`git add nom_fichier`

Ajoute un fichier au prochain commit.

`git add .`

Ajoute tous les fichiers modifiés.

8. Créer un commit

`git commit -m "message clair"`

Enregistre une nouvelle version du code.

Le message doit décrire ce que tu as modifié.

9. Voir l'état du dépôt

`git status`

Affiche :

- la branche active

- les fichiers modifiés

- les fichiers à committer

- les fichiers non suivis

10. Voir l'historique des commits

****`git log`****

Affiche tout l'historique des commits.

****`git log --oneline`****

Version concise (id court + message).

11. Supprimer une branche

Localement :

****`git branch -d nom_branche`****

Sur GitHub :

****`git push origin --delete nom_branche`****

12. Mettre à jour les infos distantes

****`git fetch`****

Met à jour les références distantes (n'affecte pas le code local).