Git : les commandes du quotidien

Les commandes Git dont vous avez besoin chaque jour, rangées par usage, avec une ligne d'explication pour chacune.

1. VÉRIFIER
`git status`
Ce qui a changé

2. PRÉPARER
`git add`
Ce qui partira

3. ENREGISTRER
`git commit`
Une étape de votre travail

4. PARTAGER
`git push`
Vers le dépôt distant

1. CONFIGURER GIT (UNE SEULE FOIS)

`git config --global user.name "Prénom Nom"`	Nom qui signera vos commits
`git config --global user.email "vous@exemple.fr"`	Adresse e-mail de vos commits
`git config --global init.defaultBranch main`	Nommer « main » la branche par défaut
`git config --list`	Afficher la configuration

2. DÉMARRER UN DÉPÔT

`git init`	Créer un dépôt dans le dossier courant
`git clone <url>`	Copier un dépôt distant sur votre machine
`git clone <url> <dossier>`	Copier dans le dossier de votre choix

3. ENREGISTRER DES CHANGEMENTS

`git status`	Fichiers modifiés, préparés ou non suivis
`git add <fichier>`	Préparer un fichier pour le prochain commit
`git add .`	Préparer tous les changements du dossier
`git add -p`	Choisir les modifications morceau par morceau
`git commit -m "message"`	Enregistrer les changements préparés
`git commit --amend`	Corriger le dernier commit, tant qu'il n'est pas envoyé

4. CONSULTER L'HISTORIQUE

`git log --oneline`	Historique, un commit par ligne
`git log --oneline --graph --all`	Toutes les branches, en arbre
`git diff`	Changements pas encore préparés
`git diff --staged`	Changements préparés, pas encore enregistrés
`git show <commit>`	Détail d'un commit
`git blame <fichier>`	Qui a modifié chaque ligne, et quand

5. TRAVAILLER AVEC LES BRANCHES

`git branch`	Lister les branches locales
`git switch -c <nom>`	Créer une branche et passer dessus
`git switch <nom>`	Passer sur une branche existante
`git merge <nom>`	Fusionner une branche dans la branche courante
`git rebase main`	Rejouer vos commits au-dessus de main
`git branch -d <nom>`	Supprimer une branche déjà fusionnée

> **Bon à savoir :** un bon message de commit dit en une ligne ce que le commit change, par exemple `Ajoute la page contact`.

6. ÉCHANGER AVEC LE DÉPÔT DISTANT

`git remote -v`	Voir les dépôts distants
`git fetch`	Télécharger les nouveautés sans les appliquer
`git pull`	Récupérer et intégrer les changements
`git push`	Envoyer vos commits
`git push -u origin <branche>`	Envoyer une nouvelle branche et la suivre

7. METTRE DE CÔTÉ

`git stash`	Mettre de côté les modifications en cours
`git stash list`	Lister ce qui est mis de côté
`git stash pop`	Réappliquer le dernier élément mis de côté

8. REVENIR EN ARRIÈRE

`git restore <fichier>`	Abandonner les modifications d'un fichier
`git restore --staged <fichier>`	Retirer un fichier de la préparation, sans perdre le travail
`git reset --soft HEAD~1`	Défaire le dernier commit, en gardant les changements
`git revert <commit>`	Annuler un commit déjà partagé, par un nouveau commit
`git reflog`	Retrouver un commit que l'on croyait perdu

9. EXEMPLE COMPLET : CORRIGER UN BUG SUR UNE BRANCHE

`git switch main`	Partir de la branche principale
`git pull`	La mettre à jour
`git switch -c correction-formulaire`	Créer une branche pour la correction
`git status`	Après vos modifications : vérifier les fichiers touchés
`git add .`	Préparer les fichiers
`git commit -m "Corrige la validation du formulaire"`	Enregistrer la correction
`git push -u origin correction-formulaire`	Envoyer la branche, puis ouvrir une demande de fusion (pull request) sur GitHub

> **Le fichier .gitignore** liste ce que Git doit ignorer : par exemple `.env` (mots de passe), `vendor/` ou `node_modules/` (dépendances téléchargées). Créez-le à la racine du dépôt, avant le premier commit.

POUR ALLER PLUS LOIN

Formation « Git et GitHub : gestion de versions pour débutants »

lapolaris.fr/formations/fondamentaux/git-github-gestion-versions-debutants

Formation « Git, GitHub Actions et Cyber Resilience Act : pipeline sécurisé pour développement .NET »

lapolaris.fr/formations/devops-deploiement/git-github-actions-cyber-resilience-act-developpement-dotnet

Cet aide-mémoire est libre de partage, à condition de citer la source : lapolaris.fr, hors utilisation commerciale (licence CC BY-NC 4.0).