

MEMENTO DES COMMANDES

Creation d'un dépôt et ajout d'un commit

Commande	Effet
<code>git init <directory></code>	Créer un dépôt git vide dans le répertoire spécifié ou initialise le répertoire courant en tant que dépôt git.
<code>git config --global user.name <name></code>	Définir le nom de l'auteur/autrice à utiliser pour les nouveaux commits.
<code>git status</code>	Afficher l'état du dépôt et la liste des fichiers inclus ou non pour le prochain commit.
<code>git add <dossier></code>	Inclure (<i>stage</i>) tous les changements dans <code><dossier></code> pour le commit
<code>git add <fichier></code>	Inclure les changements du <code><fichier></code> pour le commit
<code>git add -A</code>	Inclure tous les changements pour le commit
<code>git rm <fichier></code>	Enlever (<i>unstage</i>) <code><fichier></code> du prochain commit.
<code>git diff</code>	Afficher les lignes modifiées depuis le dernier commit.
<code>git commit -m "<message>"</code>	Valider les modifications sélectionnées (<i>staged</i>) pour créer un nouveau commit avec le message <code><message></code> .
<code>git log</code> ou <code>git log --oneline --all --graph</code>	Afficher l'historique des commits
<code>git remote add <name> <url></code>	Ajouter une connexion de votre dépôt courant à un dépôt sur un serveur.
<code>git push</code>	Pousser les nouveaux commits sur le serveur (principal).

Téléchargement et exploration d'un dépôt simple

Commande	Effet
<code>git clone <url></code>	Cloner en local un dépôt depuis l'adresse <code><url></code> généralement un serveur ou un forge.
<code>git pull</code>	Récupérer les dernières modification (#réflexe).
<code>git log --oneline</code>	Afficher l'historique avec une ligne par commit.
<code>tig</code>	Un outil plus sympa que git log pour explorer l'historique.
<code>git diff HEAD <num_commit></code>	Affiche la différence entre le commit actuel (HEAD) et le commit <code><num_commit></code> .
<code>git diff HEAD HEAD~1</code>	Affiche la différence entre le commit actuel (HEAD) et le précédent (HEAD~1).
<code>git checkout <num_commit></code>	Charge la version du code au niveau du commit <code><num_commit></code> . La "tête" se déplace au niveau de ce commit (HEAD détachée).
<code>git checkout master</code> ou <code><nom_branch></code>	Positionne HEAD au niveau du dernier commit de la branche.
<code>git reflog</code>	Affiche une liste des dernières positions de HEAD. (quand on est perdu !!!)

Les branches et les merges

Commande	Effet
<code>git branch</code>	Affiche la liste des branches
<code>git checkout <nom_branche></code>	Basculer sur la branche <code><nom_branche></code>
<code>git checkout -b <nom_branche></code>	Créer une nouvelle branche et basculer dessus
<code>git diff <branche_1> <branche_2></code>	Comparer deux branches pour voir les différences
<code>git merge <nom_branche></code>	Fusionner la branche <code><nom_branche></code> avec la branche courante.

Corriger ses erreurs

Commande	Effet
<code>git commit --amend</code>	Ajouter des modifications au commit précédent pour le corriger ou simplement changer le message du commit précédent.
<code>git reset <commit></code>	Réinitialiser le HEAD au commit indiqué en gardant les modifications.
<code>git reset --hard <commit></code>	Réinitialiser le HEAD au commit indiqué en perdant les modifications.
<code>git rebase <branche></code>	(plus complexe) Reconstruire l'historique de la branche courant à partir d'une autre branche en résolvant les conflits à chaque commit

Lexique git

Concept	Explication
Un commit	Une version validée du code avec un auteur / une autrice, un message et un identifiant unique.
Une branche	Une suite de commits avec un nom contenant une version du logiciel.
HEAD	Le commit actuellement sélectionné dans le dépôt.
remote	Un dépôt git sur un serveur par exemple la forge framagit.
origin	Le nom du remote par défaut.
master	La branche par défaut, généralement la branche principale.