

Mémento des commandes git

Creation d'un dépôt et ajout d'un commit

<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 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> <code>git add <fichier></code> <code>git add -A</code>	Inclure (stage) tous les changements dans <dossier> pour le commit Inclure les changements du <fichier> pour le commit Inclure tous les changements pour le commit
<code>git rm <fichier></code>	Enlever (unstage) <fichier> 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 (staged) pour créer un nouveau commit avec le message <message>.
<code>git log</code>	Display the entire commit history using the default format.
<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.

Téléchargement et exploration d'un dépôt simple

<code>git clone <url></code>	Cloner en local un dépôt depuis l'adresse <url> généralement un serveur ou un forge.
<code>git pull</code>	Récupérer les dernières modifications (#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 <num_commit>.
<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

	<num_commit>. La "tête" se déplace au niveau de ce commit (HEAD détachée).
git checkout master ou <nom_branch>	Positionne HEAD au niveau du dernier commit de la branche.
git reflog	Affiche une liste des dernières positions de HEAD. (quand on est perdu !!!)

Les branches et les merges

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

Corriger ses erreurs

git commit --amend	Ajouter des modifications au commit précédent pour le corriger ou simplement changer le message du commit précédent.
git reset	Annuler le dernier commit en gardant les modifications.
git reset --hard	Annuler le dernier commit et supprimer les modifications
git rebase <base>	

Lexique git

Un commit	Une version validée du code avec un auteur, un message et un numéro.
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.