

Git & GitHub

Un commit : version d'un code à un instant t

Les commandes

git init : transformer un dossier en dépôt Git

git add (nom_du_fichier) : indexer un fichier dans le dépôt Git (ajouter dans les fichiers à tracer **à chaque fois** qu'on modifie le fichier)

git commit [-m "Message"] : enregistrer les modifications (en écrivant un msg pour le détail des modif)

NB : Si on a déjà ajouté le fichier à l'index, on peut après une modification directement faire **git commit -a -m "Message"**

git log : affiche la liste de tous les commits effectués (quitter en appuyant sur la touche **Q**)

git status : consulter les informations sur le repository.

git checkout SHAduCommit (ou le mot **master** si on veut revenir au commit le plus récent) : se positionner sur un commit donné.

git reset --hard : annuler toutes les modifications d'un fichier qui n'a pas encore été commité

git clone (lien url fournit par GitHub) : télécharger un repository depuis GitHub

git push origin : envoyer les modifications effectuées depuis une branche sur Git(machine locale) sur GitHub.

git pull origin (nom de la branche) : récupérer les modifications effectuées sur un repository depuis GitHub sur Git (machine locale).

git branch : pour voir la liste des branches (il y a une étoile devant le nom de la branche sur laquelle on est situé : la branche master par défaut)

git branch (nom de la branche) : créer une nouvelle branche

git checkout (nom de la branche) : se positionner sur une branche

NB : On peut directement faire **git checkout -b** (nom de la branche) :pour créer une branche et s'y positionner directement

git branch -d (nom de la branche) : supprimer une branche.

git merge (nom de la branche) : fusionner « nom de la branche » avec la branche dans la quelle on se trouve actuellement.

git blame (nom du fichier) : voir dans le détail qui a modifié quoi sur le fichier (renvoie en complément le début du SHA de chaque commit effectué).

git show (debut du SHA) : renvoie tous les détails du commit

git stash : mettre de côté les modifications qu'on a pas terminé

git stash pop : continuer les modifications qu'on avait mis de côté

git stash apply : commiter directement les modifications qu'on avait mis de côté

Pour modifier le code sur GitHub

create a new branch => modify the code on that branch => come up to master branch and make a pull request => compare the master branch to the new branch => merge on master branch => delete the new branch created.

Pour contribuer à un projet par pull request

Retrouver le projet auquel on veut contribuer => Faire un fork du projet => cloner le projet forké sur sa machine locale (Git) => créer une branche et faire des modifications là-dessus => commiter et pushé la branche sur GitHub => faire un compare & pull request sur Github.