

## Système de contrôle de version :

SCM "source code management" (autres outils que Git)

Système décentralisé : système qui adopte un modèle de redondance.

Git utilise un système décentralisé.

Git hub : site web, plateforme de travail collaboratif

Repository . espace disque dans lequel est stocké et sauvegardé nos notes/ projet

Git enregistre chaque ajout, modification, suppression

Fonctionne avec des instantanées.

Premier commit :

Commandes :

- Git init : permet d'initialiser un projet Git ( ajoute un dossier .git )
- Git status : permet de savoir dans quels dossiers il y a eu des modifications dans le dossier depuis sa création ou depuis son dernier commit en les affichant en rouge. Elle permet également de savoir quel dossier ont été ajouté à l'enregistrement pour le prochain commit en les affichant en vert ( pour cela il faut utiliser la commande git add ).
- Git add . : ajoute tous les dossiers qui sont dans le dossier courant à la partie qui vas être commit
- Git commit -a effectue un git add et un git commit en même temps
- Git commit -m "texte here" : effectue un enregistrement de l'état actuel du dossier et le stock dans le .git au nom de ce que l'on a mis dans les parenthèses.
- git remote add origin [git@github.com:math09/lo.git](https://github.com/math09/lo.git)
- git push -u origin main
- git branch -M main : renomme la branche actuelle en "main"
- git rm -le nom du dossier : permet d'enlever un document du suivi (notamment les dependencies)
- git reset permet d'enlever un document des documents prêt à commit
- git restore : permet de revenir au dernier commit en enlevant les modifications
- git log (--pretty=oneline) (--graph)
- git branch "nom de la branche" : ajoute une branche
- git checkout "nom de la branche" : permet de changer de branche
- git checkout -b "nom de la branche" : permet de créer une branche et de se mettre directement dessus
- git pull : permet de récupérer les dernières modifications effectuées sur la branche actuelle
- git log --graph --abbrev-commit --decorate --format=format:'%C(bold blue)%h%C(reset) - %C(bold green)(%ar)%C(reset) %C(white)%s%C(reset) %C(dim white)- %an%C(reset)%C(bold yellow)%d%C(reset)' --all

- `git push --set-upstream origin nom du fichier` : permet de push une branche qui n'existe pas encore sur le repository distant

Checkout : changement de branche

`ssh-keygen -t ed25519 -C "your_email@example.com"` : génère une clé ssh (une clé publique et une clé privée ( la clé publique est a ajouter dans git hub pour permettre de push)

`.gitignore` : fichier qui répertorie l'ensemble des fichiers qui ne sont pas suivis par git

créer une copie d'un projet sur une autre branche pour ne pas impacter la principale

Une branche est un pointeur vers la branche principale (main). C'est une copie du projet de base sur laquelle on peut travailler sans causer de problème sur la branche principale.

HEAD : désigne-le commit sur lequel on travail (le dernier)

Ordre à suivre :

`git commit` ➔ `git pull` ➔ `git push`