

# Principales Commande GIT : Mémento

## I) Création de la clé SSH

La clé SSH devra être créée et enregistrée sur GitHub avant de push ou pull une requête car, elle est obligatoire, sinon ça ne marche pas (elle sert à chiffrer et déchiffrer les échanges).

- `ssh-keygen -t rsa -b 4096 -C « ton_email@exemple.com »`
  - Commande servant à créer la clé SSH.
  - A la question suivante (entrer un fichier où sauvegarder la clé), faites un retour chariot pour qu'elle soit enregistrée à l'endroit par défaut.
  - Quand ils demandent une passphrase, veuillez en entrer une assez longue mais facile à retenir pour vous, car elle sert à vous authentifier lors d'un push ou pull et il faut la rentrer à chacune de ces deux opérations (c'est le mot de passe).
  - Voir le site suivant pour toutes les informations sur la création de clés SSH (en anglais) :
    - <https://help.github.com/articles/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent/#generating-a-new-ssh-key> .
- `clip < ~/.ssh/id_rsa.pub`
  - Copie la clé (publique) SSH dans votre presse papier
  - Il ne reste plus qu'à aller l'ajouter à votre compte GitHub :
    - Log in GitHub → Settings → SSH and GPG keys → New or Add SSH key.
    - Dans la partie Title, vous pouvez par exemple mettre le nom de la machine sur laquelle est la clé SSH car la clé servira à faire le lien entre votre compte GitHub et cette machine.
    - Collez ce que vous avez dans votre presse-papier dans la partie Key.
    - Validez avec Add SSH key.
    - Entrez votre mot de passe GitHub et validez.
  - Voir le site suivant pour toutes les informations sur l'ajout d'une clé SSH à GitHub (en anglais) :
    - <https://help.github.com/articles/adding-a-new-ssh-key-to-your-github-account/> .

## II) Manipulation GIT

Cette partie décrit les principales commandes GIT et leurs effets. Le tutoriel Youtube suivant peut aussi aidé à prendre en main ce gestionnaire de version :

[https://www.youtube.com/playlist?list=PLlxQJeQRaKDRBd\\_FazeI7gLq5wyr7f7J](https://www.youtube.com/playlist?list=PLlxQJeQRaKDRBd_FazeI7gLq5wyr7f7J) .

- git config –global user.name « Prénom NOM »
  - Définit votre nom et prénom (il y a toujours double tiret devant global).
- git config –global user.mail [mail@example.fr](mailto:mail@example.fr)
  - Définit votre mail comme [mail@example.fr](mailto:mail@example.fr) .
- git init
  - Initialise une gestion de version d'un projet (à utiliser à la racine du répertoire du projet).
- git status
  - Permet de vérifier les modifications faites depuis le dernier commit.
- git log
  - Permet de vérifier la liste de commit déjà fait.
- git add nom\_fichier
  - Indexe les modification du fichier nom\_fichier, ou le nouveau fichier nom\_fichier, pour le prochain commit.
- git add -A
  - Indexe toutes les modifications faites dans les fichiers déjà indexés et indexe les nouveaux fichiers créés pour le prochain commit (à part ceux dont les noms sont précisés dans un fichier .gitignore).
- git commit -m « Phrase exemple »
  - Commit toutes les modifications, ajouts ou suppressions indexés et met pour description du commit « Phrase exemple » (en gros, cela crée une nouvelle version du produit).
- git branch nom\_branch
  - Crée une nouvelle branche appelée nom\_branch.
  - La branche principale d'un répertoire GIT est toujours la branche « master ».
- git checkout nom\_branch
  - On se positionne sur la branche nom\_branch.
- git checkout id\_commit
  - On se positionne sur le commit identifié par id\_commit (visible à côté de commit quand on fait un 'git log').
  - Attention : les commits suivants le commit où l'on se positionne ne seront plus accessibles car on les détruit et on reprend le travail à partir du commit où l'on se repositionne.

- `git remote add origin url_depot_github`
  - Ajoute un lien entre votre GIT local et un GitHub qui aura pour nom origin et pour dépôt GitHub celui dont l'adresse est url\_depot\_github.
  - « origin » est le nom habituel qu'on donne au lien que l'on fait avec le répertoire du projet sur le site de gestion de version (GitHub). On peut mettre un autre nom si l'on veut mais si on utilise un seul site il est plus fréquent que l'on mette origin.
  - La remote (ou lien) créée nous servira à mettre à jour ou à récupérer le travail sur le site GitHub.
- `git push origin nom_branch`
  - On pousse le dernier commit fait sur la branche nom\_branch locale vers le dépôt distant configuré par la remote origin (par exemple, un répertoire GitHub).
  - Attention : il vaut mieux se créer chacun une branche ou on push plutôt que de push sur la branch master car si on y push alors que quelqu'un d'autre l'a modifié on pourrait perdre du travail même si normalement, il interdit de push si on a pas récupéré les dernières modifications avec un pull.
  - Même si on va récupérer la dernière version avec un pull et qu'on met nos fichiers modifiés dans le répertoire si nos camarades les avaient modifiés on perdrait leurs modifications.
- `git pull origin nom_branch`
  - On récupère le travail de la branche nom\_branch, sur le dépôt distant configuré par la remote origin.
  - Attention : si vous avez des fichiers que vous avez modifié sur votre dépôt local vous risquez de perdre ces modifications si ces fichiers sont sur le dépôt distant (et qu'ils n'ont pas vos modifications).
- `git clone url_depot_github`
  - Crée un répertoire local avec tous les fichiers du dépôt dont l'adresse est url\_depot\_github (clone tout les fichiers du dépôt distant vers ce nouveau répertoire local).
  - Équivaut à télécharger le dossier du dépôt GitHub dans le dossier où l'on se situe quand on lance la commande.