

# GitHub / Terminal

GitHub = Un des services utilisant les repositories Git, mais il en existe d'autres (GitLab, BitBucket, Azur, etc) !

## Utiliser le terminal

**cd** Aller à un dossier (change directory)

**cd ~** Retourner à la base, dossier "home"

**ls** Lister les fichiers/dossiers

**ls -l** Lister en détail dans l'ordre alphabetic

**ls -a** Lister tous, même les fichiers cachés

**mkdir NomDuDossier** Crée un dossier

**rm** Supprime, remove

**echo > Hello World > msg.txt** Crée un fichier msg.txt contenant le message Hello World

**touch fichier.html** Crée un fichier

## Git

**git init** Créer un repo Git (crée un fichier .git "invisible" dans le dossier), il suffit de refaire un git init pour l'enlever

**git status** Permet de voir les fichiers modifiés, non-trackés, etc

**git add exemple.php** Ajouter un fichier dans le stage (le préparer à être commit)(pareil pour les fichiers supprimés, qu'il faut aussi add et commit)

**git add \*.html** Permet d'add tous les fichier .html (m\* prendrait tous les fichiers commençant par m, juste une \* prendrait tout, mais il faut faire attention dans ce cas)

**git restore erreur.txt** Permet de restaurer un fichier au dernier état commit, même s'il y a eu une sauvegarde, ou supprimé

**git restore --staged fichier.txt** Retire un fichier staged

**git diff fichier.txt** Permet de voir les changements apportés

**git commit -m "Commentaire"** Commit le repo (toujours mettre un message, qui décrit les derniers changement par exemple)

**git log** Montrer les log

**git log -5** Montrer les 5 derniers log

**git config --global user.name "Julie De Deken"** Défini l'username global, on peut retirer le global si on veut que ce repo aie un nom différent

**git remote add origin urlDuRepoSurGitHub** Remote de base, on le nomme traditionnellement origin, crée une connexion entre le local et le remote

**git remote -v** Pour vérifier le(s) remotes

**git push** Pusher les commit en ligne (le premier push nécessite la commande `git push --set-upstream origin master`, mais il suffit de le copier coller quand git le dit)

**git clone urlDuReposurGitHub** Clôner, copier le repository sur un autre ordinateur

**git pull** Prendre les commits du serveur depuis l'autre ordi

**esc :wq /** pour sortir de la console infernale

## GitHub

- Créer un nouveau repository, on peut le faire en son nom, ou alors avec une organisation
- Généralement on donne le nom de son dossier à son repo
- Ne surtout pas

toucher quand on  
importe un repo  
déjà existant  
(dossier qui a été  
git init)

**Initialize this repository with:**

Skip this step if you're importing an existing repository.

☐ **Add a README file**

This is where you can write a long description for your project. [Learn more.](#)

☐ **Add .gitignore**

Choose which files not to track from a list of templates. [Learn more.](#)

☐ **Choose a license**

A license tells others what they can and can't do with your code. [Learn more.](#)

- Git init → Git remote add origin url → Git add → Git commit  
→ Git Push

## Utiliser sur un autre ordi

- Me placer dans le bon dossier, dans lequel je veux cloner mon repo
- git clone <https://github.com/julieddkn/20210622-github.git>
- D'un ordi ou de l'autre, c'est la même marche à suivre pour ajouter sur le repo → add, commit, push
- Dans l'autre sens, git pull