**Liste de commande dans Git**

**Mkdir** : *mkdir myRepositoryName* crée un nouveau dossier

**Init** : *git init* active un dossier comme repository Git (on doit être placé dedans)

**Touch** : *touch nameOfTheFile.ExtensionOfTheFile* crée un nouveau fichier

**Cat** : *cat nameOfTheFile.ExtensionOfTheFile* affiche le contenu d’un fichier

**Ls**:*ls* affiche les différents fichiers du répertoire

**Add** : *git add nameOfTheFile.ExtensionOfTheFile* ajoute le fichier à l’index Git ou tous les fichiers avec *git add .*

**Commit** : *git commit* ou *git commit -m « message »* enregistre les modifications après un add. Si on a déjà ajouté le fichier, utiliser plutôt *git commit -a -m « message »* à la place de add + commit

**Log** : *git log* affiche la liste de tous les commits réalisés référencés par leurs SHA (une clé de hachage). Pour quitter *q*

**Checkout** : *git checkout SHADuCommit* pour se positionner sur un commit donné ou *git checkout master* pour se positionner sur le plus récent commit

**Clone**:  *git clone linkGivenByGithub* copie un dossier sur internet ou autre, sur son ordinateur

**Push**: *git push origin master* après avoir commit, envoie le dossier modifié sur origin (= remote = Github) avec la branche master

**Pull** : *git pull origin master* envoie dans mon répertoire local les dernières modifications de la branche master située sur mon remote origin

**Branch**: *git branch myNewBranch* crée une nouvelle branche séparée de celle sur laquelle nous sommes (de base master). *Git branch* pour savoir sur quelle branche nous nous trouvons.

**Checkout**: *git checkout myNewBranch* pour se positionner sur une branche donnée (master pour revenir à la principale). *Git checkout -b myNewBranch* regroupe l’opération branch + checkout

**Merge**: *git merge branchB* pour fusionner la branche B avec la branche A sur laquelle nous sommes positionnés

**Blame**: *git blame nameOfTheFile.extension* liste toutes les modifications à travers les commit qui ont été faites, faisant correspondres les SHA

**Show**: *git show startOfTheSHA montre les différences entre le dernier commit avant celui du SHA et celui du SHA lui-même*

**Gitignore**:créer un fichier texte nommé .gitignore permet d’y inscrire à l’intérieur la liste des fichiers que nous ne voulons pas ajouter à Git

**Stash**: *git stash* permet de mettre de côté des modifications sans faire de commit, et on peut ensuite revenir dessus avec la commande *git stash pop* ou  *git stash apply* pour garder le contenu du stash dans le stash.