Documentation Github

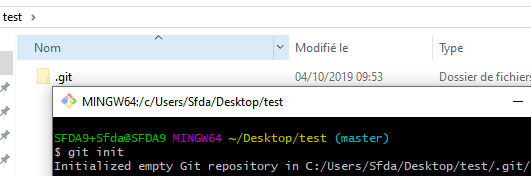
1. **Histoire :** Git a été créé le 7 avril 2005, en 2016 ils annoncent que plus de 12 millions d’utilisateurs sont inscrits. Il a été créé en C, C++, Python, Perl. Git a été fait par Linux après la séparation avec une société qui a développé BitKeeper. Les intérêts de Git sont d’être simple, rapide, pouvoir gérer les gros projets et de gérer les projets en sous-dossier (branche).

**Utilité :** Git permet de stocker des fichiers en arborescence gratuitement, simplement à partir d’une invite de commande basée en Linux (Gitbash). On peut gérer les versions de nos programmes et voir le code depuis internet.

**Installation :** A partir du site <https://git-scm.com/downloads>, Cliquez sur le nom de votre OS (Système d’exploitation), Puis un téléchargement devrait se lancer automatiquement. Si ce n’est pas le cas, cliquez sur le lien pour le télécharger. Ensuite, ouvrez votre fichier téléchargé. Cliquez sur Next, puis choisissez l’endroit où enregistrer l’installation. Appuyez sur Next, cochez les cases que vous pensez nécessaire, puis 2 fois Next, choisissez sur quel éditeur de texte vous voulez utiliser. Cliquez sur Next jusqu’à la fin. L’installation est fini, il ne reste qu’à ouvrir Gitbash en faisant clique droit dans un dossier.

1. **Commandes :**

git init : permet de configurer un nouveau / existant projet. C’est la premiere commande a faire.



git config : permet de configurer les préférences de l’utilisateur. Comme configurer son prenom et nom, son adresse mail ...



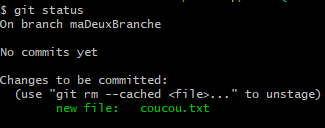
git checkout : peut être utilisée pour créer des branches ou pour basculer entre elles. Pour créer une branche, on utilise le -b puis le nom de notre branche entre guillemets.



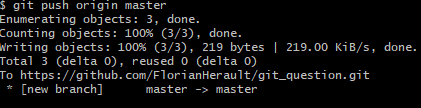
git add : permet d’ajouter des fichiers à l’index. Par exemple, la commande suivante ajoutera un fichier nommé coucou.txt dans le répertoire :



git status : permet de voir les nouveaux fichiers créés, la branche actuelle, etc. Affiche la liste des fichiers modifiés ainsi que les fichiers qui doivent encore être ajoutés ou validés.



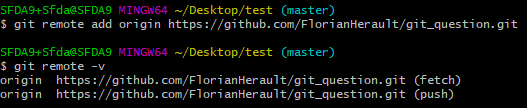
git push : Un simple push envoie toutes les données sur Github.



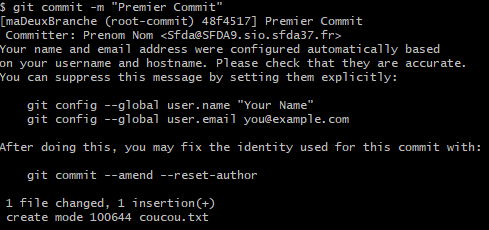
git branch : peut être utilisée pour répertorier, créer ou supprimer des branches.



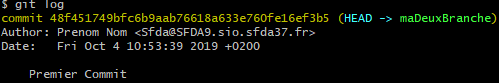
git remote : permet à un utilisateur de se connecter à un dépôt distant. Le -v permet de connaître auquel le fichier est connecté.



git commit : La commande git commit permet de valider les modifications apportées.



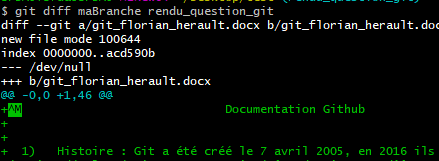
git log : génère le log d’une branche. Cela donne celui qui a créer, la date, le nom du commit.



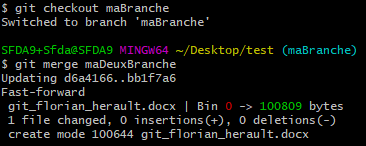
git pull : Pour fusionner toutes les modifications présentes à distance dans le répertoire de travail local, ou juste récupérer les fichiers.



git diff : Permet de connaître les différences entre deux branches, les fichiers qui changent. Utile pour savoir les conflits.



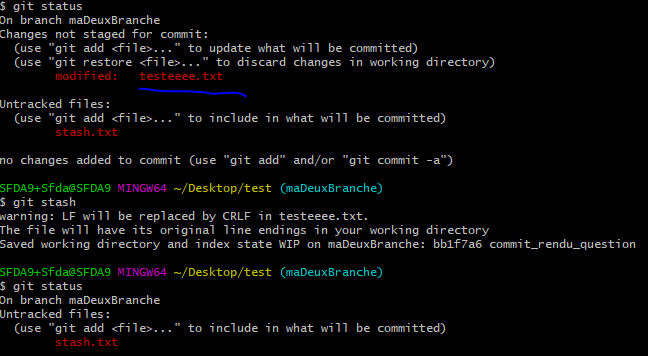
git merge : Permet de fusionner deux branches. Pour cela aller sur une branche puis utiliser merge sur l’autre.



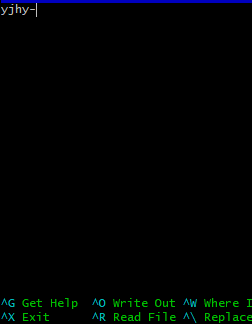
git fetch : Cette commande télécharge les données des commits qui n'ont pas encore été récupérées dans votre branche locale puis fusionne ensuite ces données.



git stash : Annule les modifications non sauvegarder. Très utile pour faire un retour, par exemple si on a modifier des fichiers par erreur.



nano : permet de modifier l’intérieur d’un fichier (un .txt par exemple). On peut écrire ce qu’on veut. Une fois fini, faites contrôle + X, puis il vous demande si vous voulez sauvegarder. Appuyez sur y pour sauvegarder.



git clone : Permet de cloner un fichier vers un autre, donc de créer un fichier identique à celui d’origine. Cela copie tous les fichiers à l'intérieur. On peut aussi mettre un lien github.

