

## Gestion de git en ligne de commandes

Après installation de git et configuration comme ceci:

**git config --global user.name "Mon nom"**

**git config --global user.email "mon@email.fr"**

On peut démarrer git bash dans le répertoire de travail

On peut effectuer les commandes **cd ..** ou **ls**

On va créer un dossier avec la commande **mkdir test**

On se place dans ce dossier test avec **cd test**

On initialise le dépôt avec **git init**

On peut démarrer par la création d'un fichier readme : **notepad.exe**

**readme.txt**

Le fichier readme.txt s'ouvre et on peut par exemple écrire à l'intérieur :

"Ma première phrase"

On peut écrire la commande **git status** pour connaître le status sur l'état de notre versioning.

On va informer git qu'il y a un nouveau fichier à gérer avec la commande

**git add readme.txt**

Si on refait git status on verra que notre fichier readme.txt est prêt à être ajouté

On doit valider nos changements avec la commande **git commit -m**

"initial load"

**git status** nous montre que nos modifications ont été validées

On ouvre à nouveau le fichier readme.txt on le modifie en ajoutant une deuxième phrase comme "Ma deuxième phrase"

Avec **git status** on peut voir que le fichier readme.txt a été modifié

On crée un deuxième fichier nommé credits.txt avec la commande

**touch credits.txt**

On peut lister le nombre de fichiers qu'il y a dans notre répertoire test avec la commande **ls**

On présente tous les fichiers à git avec **git add .**

On vérifie avec **git status**

On valide toutes les modifications avec **git commit -m "deuxieme modifs"**

On va voir l'historique des modifications avec la commande **gitk**

## **Création et gestion des branches**

On crée une branche nommée "anouchka" : **git branch anouchka**

On liste les branches : **git branch**

Pour faire des modifications sur la branche anouchka : **git checkout anouchka**

git va alors activer les modifications sur la branche anouchka

On va donc éditer à nouveau notre fichier credits.txt : **notepad.exe credits.txt**

Toutes les modifications ne vont pas impacter la branche principale master

On fait les commandes suivantes:

**git status**

**git add .**

**git commit -m "feature"**

**git status**

On va fusionner les modifications de la branche anouchka dans la branche master. Pour cela, revenons à notre branche master avec la commande **git checkout master**

On va maintenant dire à la branche master de fusionner avec la branche anouchka avec la commande **git merge anouchka**

Ouvrez et modifiez les fichiers readme.txt et credits.txt avec les double commandes : **notepad.exe readme.txt** et **notepad.exe credits.txt**

Créer un fichier logs : **mkdir.exe logs**

On se place dans le dossier logs: **cd logs**

On ajoute un fichier log pour enregistrer l'historique des modifications:

**touch log.dat**

On revient dans notre dossier test : **cd ..**

On fait : **git add .**

**git commit -m "add folder logs"**

On repart sur notre branche anouchka : **git checkout anouchka**  
En tapant la commande **ls** on constate que notre dossier logs n'y est pas.

On va fusionner les modifications de la branche master à la branche anouchka: **git merge master**

On peut voir les changements dans la branche anouchka: **ls**

Lors d'un conflit, on peut voir la liste des branches et les modifications avec : **git --all**

Dans le dossier projet, clique droit on peut accéder à "git gui here" pour voir les différences des modifications.

Pour faire la fusion manuellement en modifiant le document en conflit, on ouvrira avec : **notepad.exe readme.txt**

**git add .**

**git commit -m "fix merge"**

**git status**

**gitk --all**

On supprime la branche anouchka **git branch -D anouchka**

arrêt à 8:09