

TP1 : GIT

Partie 1

- 1-
 - a. Indiquer votre nom pour savoir qui a fait le commit à l'aide de la commande :
git config --global user.name « votre nom »
 - b. Pour s'assurer que votre nom a bien changé vous pouvez taper :
git config --global user.name
 - c. De meme pour votre adresse mail vous pouvez la changer à l'aide de la commande :
git config -- global user.email « votre email »

```
C:\Users\NB>git config --global user.name "Samya"
C:\Users\NB>git config --global user.name
Samya
C:\Users\NB>git config --global user.email "S.bouhaddour@ems1.ma"
```

- d. Avant de commencer n'importe quel projet il faut toujours l'initialiser, à l'aide de la commande :
git init
- e. Après l'exécution de cette commande qu'est ce qui a changé ?

2-

- a. Alors maintenant on va ajouter 2 fichiers à notre dossier (par exemple test1.txt et test2.txt) et les ajouter à Git à l'aide de la commande suivante :
git add text1.txt
git add text2.txt
git status
Sinon vous pouvez utiliser la commande **Git add -A** pour ajouter tous les fichiers.

```
C:\Users\NB>git add test1.txt
C:\Users\NB>git add test2.txt
```

```
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
   new file:   test1.txt
   new file:   test2.txt
```

- b. Modifier l'un des fichiers et retaper la commande : Git status : Qu'est ce qui s'est passé ?

```
No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   test1.txt
    new file:   test2.txt

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   test2.txt
```

c. Pour capturer un instantané des changements actuellement stagnés du projet on utilise la commande :

git commit

Cette commande nous permet de créer une version sûre du projet. Un nouvel éditeur est ouvert avec les détails de votre commit.

```
Invite de commandes - git commit

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch master
#
# Initial commit
#
# Changes to be committed:
#   new file:   test1.txt
#   new file:   test2.txt
#
# Changes not staged for commit:
#   modified:   test2.txt
#
```

Pour sortir de cette éditeur **tapez :x ! puis entrer pour enregistrer** toutes modifications, sinon **:q ! puis entrer pour sortir sans enregistrer**.

d. On peut ajouter un commentaire pour décrire notre commit à l'aide de l'option **-m** : **git commit -m « message »**

```
C:\Users\NB>git commit -m "initial commit"
[master (root-commit) 5e1dc07] initial commit
2 files changed, 2 insertions(+)
create mode 100644 test1.txt
create mode 100644 test2.txt
```

e. Qu'est que la commande **git status** vous affiche après le commit ?

f. Pour faire le suivie des commit effectués, il suffit de taper la commande : **git log**

```
C:\Users\NB>git log
commit 5e1dc078edecf1a7f9b6e7eb83e383d02aa2cb43 (HEAD -> master)
Author: Samya <S.bouhaddour@emsi.ma>
Date: Tue Feb 27 18:35:24 2024 +0100

    initial commit
```

Partie 2

Pour tout le monde :

1. Créez un compte sur GitHub :

Rendez-vous sur le site web de GitHub (<https://github.com>) et suivez les instructions pour créer un compte si vous n'en avez pas déjà un.

2. Installez GitHub Desktop :

Téléchargez et installez GitHub Desktop à partir du site officiel : <https://desktop.github.com/>

3. Ouvrez GitHub Desktop :

Lancez l'application GitHub Desktop sur votre ordinateur et connectez-vous avec votre compte GitHub.

Une personne du groupe :

a. Créez un dépôt nommé "TP Git" :

Sur GitHub Desktop, cliquez sur "File" (Fichier) en haut à gauche, puis sur "New Repository" (Nouveau dépôt). Nommez le dépôt "TP-Git" et créez-le.

b. Créez un fichier avec NotePad++ (ou autre) dans le dépôt sur votre ordinateur.

c. Remplissez le fichier avec un bout de code (n'importe quel langage, il peut être un fichier texte).

d. Sauvegardez le fichier.

e. Retournez à GitHub Desktop, vous devriez voir le fichier répertorié dans les fichiers non suivis.

f. Faites un commit : Saisissez un message de commit significatif dans le champ prévu à cet effet, puis cliquez sur "Commit" (Valider).

g. Faites un push vers le dépôt GitHub en ligne.

h. Publiez votre dépôt et ajoutez les autres en collaborateurs sur le dépôt. Ils reçoivent une invitation par mail, ils doivent l'accepter.

Ensuite, les autres, chacun à son tour :

a. Faites un pull : Sur GitHub Desktop, cliquez sur "Repository" (Dépôt) en haut, puis sur "Pull" (Récupérer) pour obtenir les dernières modifications du dépôt.

b. Regardez l'historique pour vérifier que les changements sont là.

c. modifier/ajouter du contenu au fichier dans le dossier du dépôt sur votre ordinateur.

Partie à faire individuellement (chacun sur son ordinateur, simultanément)

1. Ouvrez GitHub Desktop et clonez le dépôt précédemment créé si ce n'est déjà fait.

2. Ajoutez/modifier dans le fichier. Ne pas oublier de sauvegarder le fichier.

3. Ajoutez le fichier dans GitHub Desktop.

4. Faites un commit. (pas de push)

5. Observez et comparez les historiques de chacun.

Une personne fait un push. Les autres personnes ne font rien.

1. Une des 2 autres personnes fait le pull sur son ordinateur (cliquez sur "push" si "pull" n'est pas affiché, et cliquez "close" sur le message d'erreur qui s'affiche).

2. Résolvez le conflit de merge ensemble (il faut éditer le fichier).

3. Une fois le merge terminé, faites un commit et un push.

4. La dernière personne fait le pull sur son ordinateur.

5. Résolvez le conflit de merge ensemble pour avoir toutes les blagues.

6. Les autres peuvent faire un pull pour récupérer toutes les blagues.

7. Comparez à nouveau vos historiques.