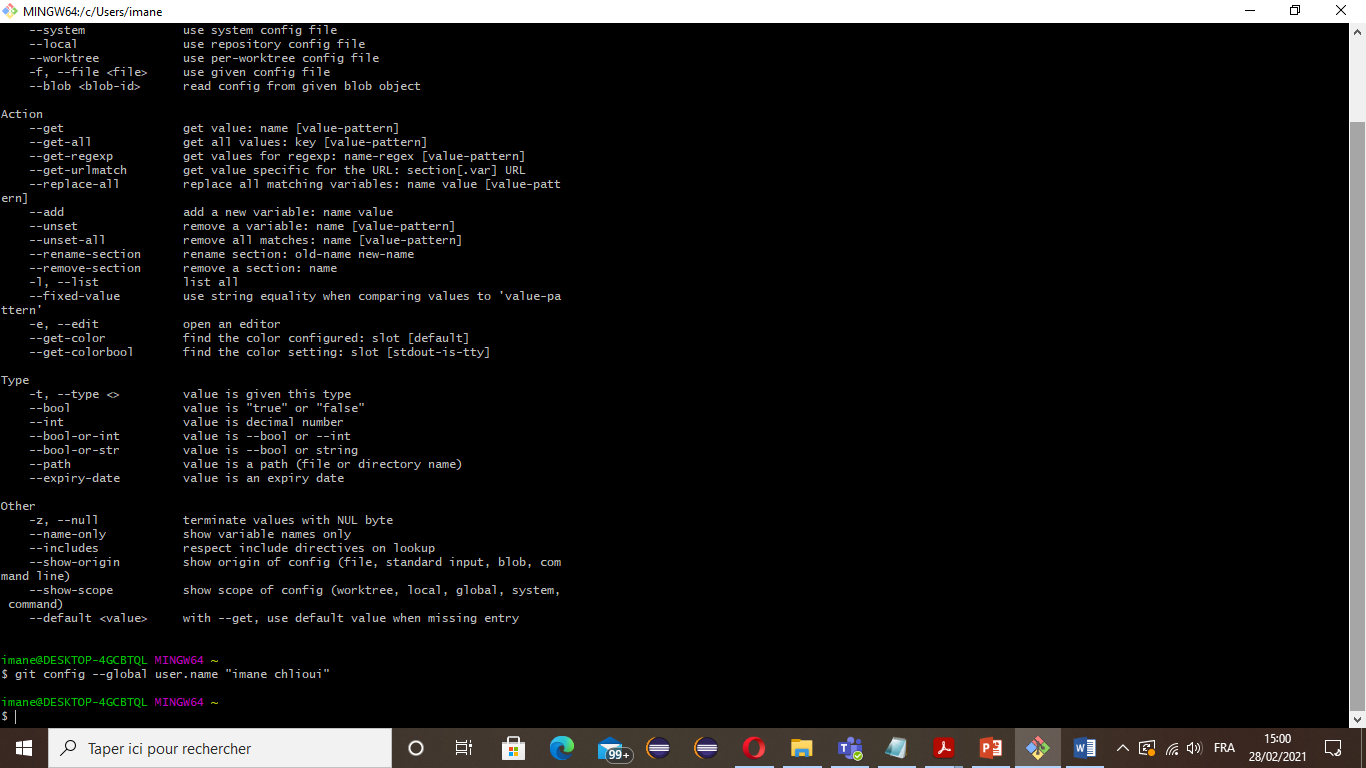
**TP1 : GIT**

* Indiquer votre nom pour savoir qui a fait le commit à l’aide de la commande :

**Git config –global user.name « votre nom »**

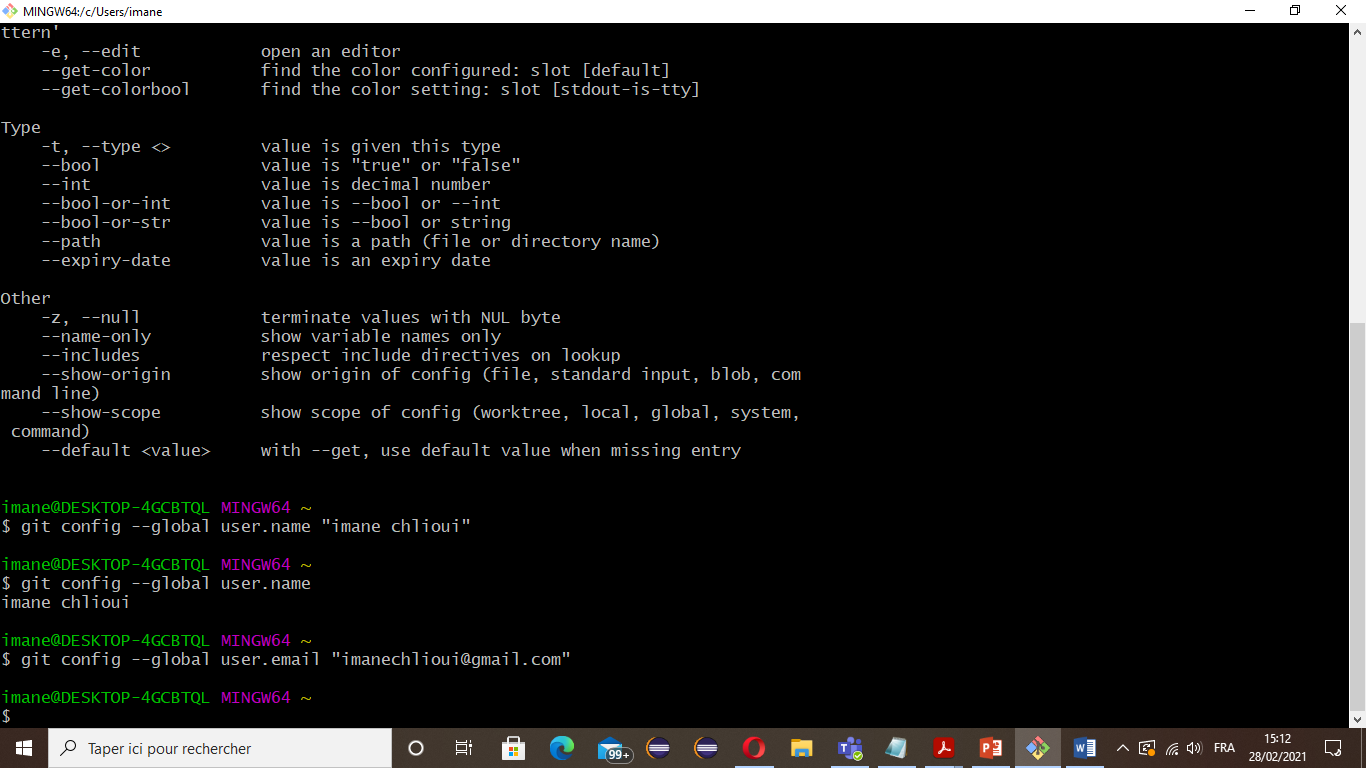


* Pour s’assurer que votre nom a bien changé vous pouvez taper :

**Git config –global user.name**

* De meme pour votre adresse mail vous pouvez la changer à l’aide de la commande :

**Git config -- global user.email « votre email »**



* Avant de commencer n’importe quel projet il faut toujours l’initialiser, à l’aide de la commande : **Git init**

1. Après l’exécution de cette commande qu’est ce qui a changé ?

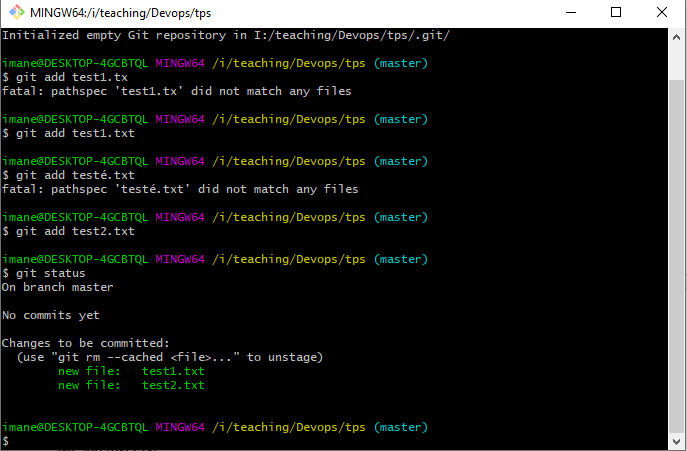
* Alors maintenant on va ajouter 2 fichiers à notre dossier (par exp test1.txt et test2.txt) et les ajouter à Git pour suivre les changements

**Git add text1.txt**

Ou bien la commande **Git add –A** pour ajouter tous les fichiers

**Git add text2.txt**

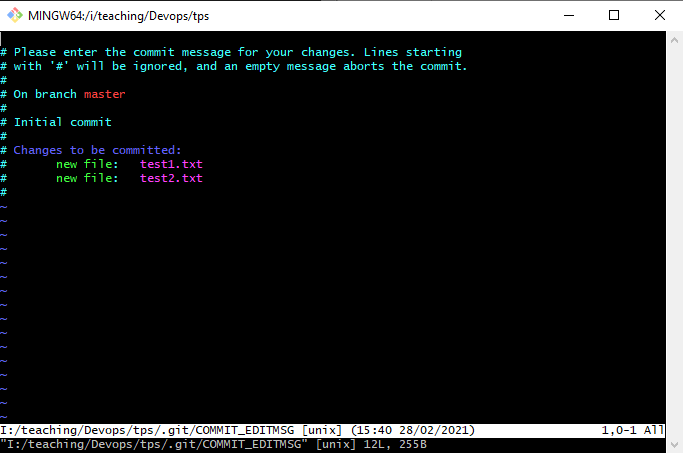
**Git status**

****

1. Modifier l’un des fichiers et retaper la commande : **Git status** : Qu’est ce qui s’est passé ?

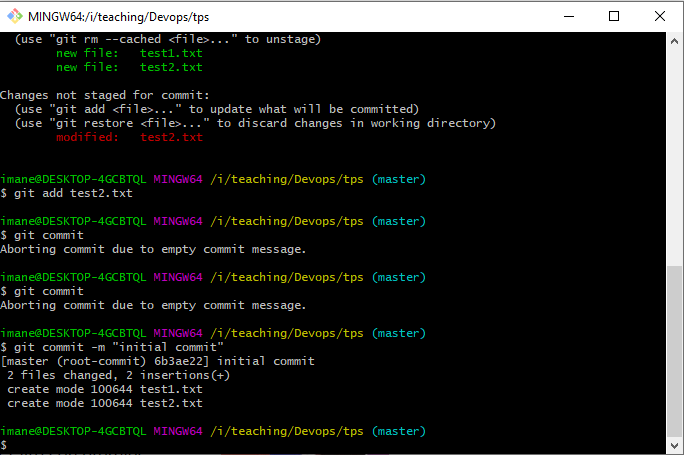
* Pour capturer un instantané des changements actuellement stagés du projet on utilise la commande **Git commit**. Cette commande nous permet de créer une version sure du projet.

Un nouvel éditeur est ouvert avec les détails de votre commit :



* Pour sortir de cette éditeur tapez **:x !** puis entrer pour enregistrer toutes modifications, sinon **:q !** puis entrer pour sortir sans enregistrer
* On peut ajouter un commentaire pour décrire notre commit à l’aide de l’option **–m** :

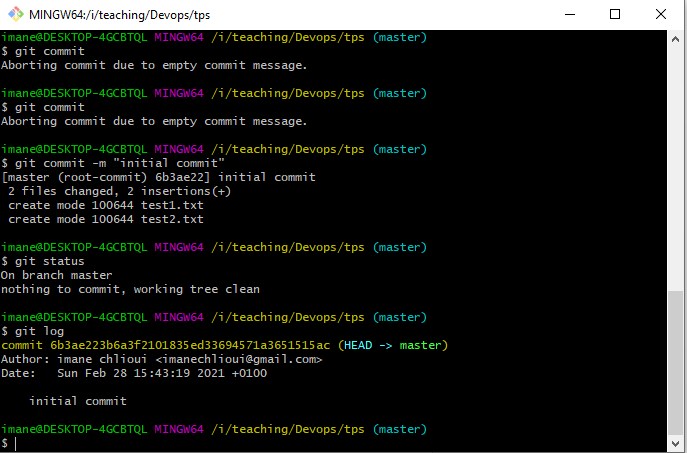
**Git commit –m « message »**



1. Qu’est que la commande Git status vous affiche après le commit ?

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

**Git log**

****

**Exercice à faire :**

Exercice à faire par groupe de 2 ou 3 : Pour tout le monde :

* Créez un compte sur GitHub.
* Installez GitHub Desktop, puis ouvrez-le.

Une personne du groupe :

* Créez un dépôt nommé TP Git

Une personne :

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

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

3. Sauvegardez le fichier.

4. Reprenez GitHub Desktop et ajoutez le fichier.

5. Faites un commit.

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

7. Publiez votre dépot 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 :

8. Faites un pull

9. Regardez l’historique pour vérifier que les changements sont là.

10. modifier/ajouter dans le fichier.

11. Refaites les étapes 3 à 6 ci-dessus.

N’hésitez pas à répéter cela plusieurs fois, pour être sûrs de bien comprendre !

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

* + Ouvrez GitHub Desktop et clonez le dépôt précédemment créé si ce n’est déja fait.
  + Ajoutez/modifier dans le fichier. Ne pas oublier de sauver le fichier.
  + Ajoutez le fichier dans GitHub Desktop.
  + Faites un commit. (pas de push)
  + Observez et comparez les historiques de chacun.

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

* + 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).
  + Résolvez le conflit de merge ensemble (il faut éditer le fichier).
  + Une fois le merge terminé, faites un commit et un push.
  + La dernière personne fait le pull sur son ordinateur.
  + Résolvez le conflit de merge ensemble pour avoir toutes les blagues.
  + Les autres peuvent faire un pull pour récupérer toutes les blagues.
  + Comparez à nouveau vos historiques.