

## EXERCICE PARIE 3 - Solution proposée

### 1- Qu'est-ce qu'un commit

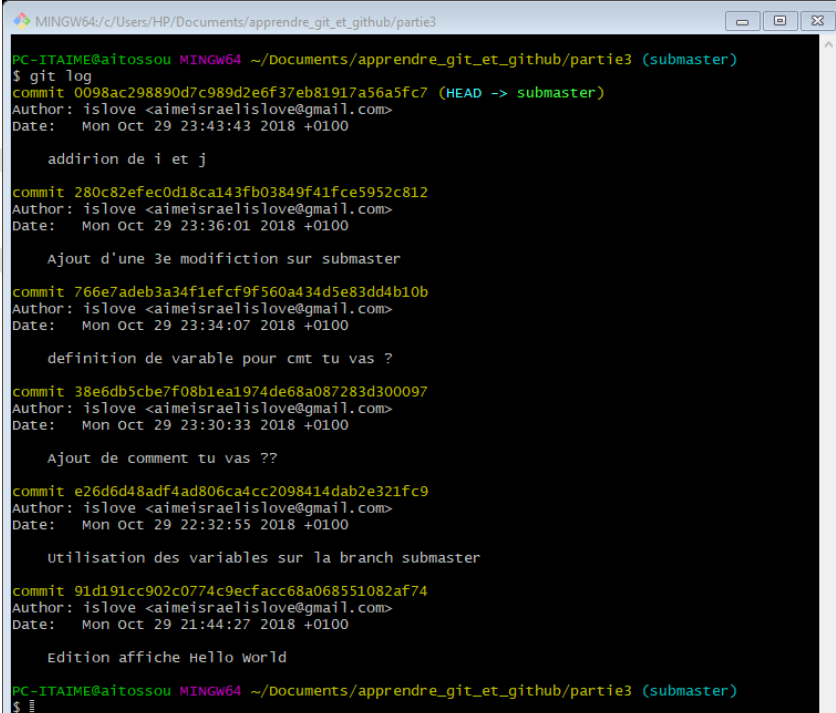
Dans tous les domaines les travaux que ce soit technique ou pas ont besoin d'être sauvegarder. Cette sauvegarde peut se faire sur un support papier ou numérique. Ainsi dans le domaine de l'informatique, notamment dans les systèmes de bases de données, de révision de fichier (par exemple du code) les termes d'archivage, de soumission, de validation ou encore l'anglais **commit** désignent l'enregistrement. Cet enregistrement entérine l'exécution de la tâche préalablement confiée, marquant à la fois la fin de la demande de transaction et le début de l'exécution de la tâche confiée, qui devra être exécutée automatiquement. Git est un système de contrôle de version distribué gratuit et en licence open source. Il a été conçu pour traiter aussi bien des petits ou de très gros projets avec rapidité et efficacité.

Commande: **git commit -a -m "message du commit"** (permet de faire les commits. L'option -a permet d'ajouter à l'index et l'option -m permet ajouter un message détaillant le commit).

### 2- À quoi sert la commande : **git log**

Après avoir créé plusieurs **commits** (sauvegardes) ou si vous avez cloné (copié) un dépôt ayant un historique de commits, vous souhaitez probablement revoir le fil des événements. Pour ce faire, la commande **git log** est l'outil le plus basique et le plus puissant. Ainsi on dira que git log permet d'afficher toutes les sauvegardes, enregistrements effectués. Pour bien illustré ces propos voici une sauvegarde en capture d'écran.

Commande : **git log**



```
MINGW64~/c/Users/HP/Documents/apprendre_git_et_github/partie3
PC-ITAIME@aitossou MINGW64 ~/Documents/apprendre_git_et_github/partie3 (submaster)
$ git log
commit 0098ac298890d7c989d2e6f37eb81917a56a5fc7 (HEAD -> submaster)
Author: islove <aimeisraelislove@gmail.com>
Date: Mon Oct 29 23:43:43 2018 +0100

    addition de i et j

commit 280c82efec0d18ca143fb03849f41fce5952c812
Author: islove <aimeisraelislove@gmail.com>
Date: Mon Oct 29 23:36:01 2018 +0100

    Ajout d'une 3e modification sur submaster

commit 766e7adeb3a34f1efcf9f560a434d5e83dd4b10b
Author: islove <aimeisraelislove@gmail.com>
Date: Mon Oct 29 23:34:07 2018 +0100

    definition de variable pour cmt tu vas ?

commit 38e6db5cbe7f08b1ea1974de68a087283d300097
Author: islove <aimeisraelislove@gmail.com>
Date: Mon Oct 29 23:30:33 2018 +0100

    Ajout de comment tu vas ??

commit e26d6d48adf4ad806ca4cc2098414dab2e321fc9
Author: islove <aimeisraelislove@gmail.com>
Date: Mon Oct 29 22:32:55 2018 +0100

    utilisation des variables sur la branch submaster

commit 91d191cc902c0774c9ecfacc68a068551082af74
Author: islove <aimeisraelislove@gmail.com>
Date: Mon Oct 29 21:44:27 2018 +0100

    Edition affiche Hello world

PC-ITAIME@aitossou MINGW64 ~/Documents/apprendre_git_et_github/partie3 (submaster)
$
```

### 3- Qu'est-ce qu'une branche

La commande `branch` permet de gérer tout ce qui a trait aux branches (ajout, listing, suppression, renommage).

Commande: `git init`

Commande: `git branch nom-de-branche` (Permet de lister les branches) et `git branch -b nom-de-branche` (Permet de lister les branches et de s'y positionner)

Commande : `git merge master` (cela signifie que la branche courante est **<disk-jaune>**)  
Ainsi la branche master est fusionné dans la branche **<disk-jaune>**.

Autres commandes:

```
git branch <ma-branche>           # Permet de créer une nouvelle branche <ma-branche>
git branch -m <rename-branche>    # Renomme la branche courante en <rename-branche>
git branch -d <ma-branche>        # Permet de supprimer une branche.
```

Dans capture master est la branche principale et submaster est une autre branche. Remarquer le mot « HEAD » qui indique qu'elle est la branche courante.

Il faut noter qu'une branche reste une branche qu'elle soit master ou pas. Le nom master juste une convention de nom pour se retrouver quand on en a plusieurs.