

# Présentation de Git pour le cours OpenClassrooms



Git permet de conserver la trace des différentes versions d'un travail (document, code, etc.)

Le but de ce texte est de présenter trois concepts de *git* :

- La notion de *commit*
- La commande *git log*
- La notion de *branche*

## Qu'est-ce qu'un *commit* ?

Un *commit* est en quelque sorte une *version* de mon travail, dont je décide de sauvegarder l'état à un moment donné.

Il est important de bien penser son *message de commit*, afin de pouvoir ensuite savoir clairement à quoi se rapporte une version, même plusieurs mois après sa réalisation.

(Les choses nous paraissent toujours évidentes au moment où l'on sauvegarde un *commit*, mais nous oublions très vite avec le temps ...)

Pour sauvegarder un *commit*, il faut avoir auparavant :

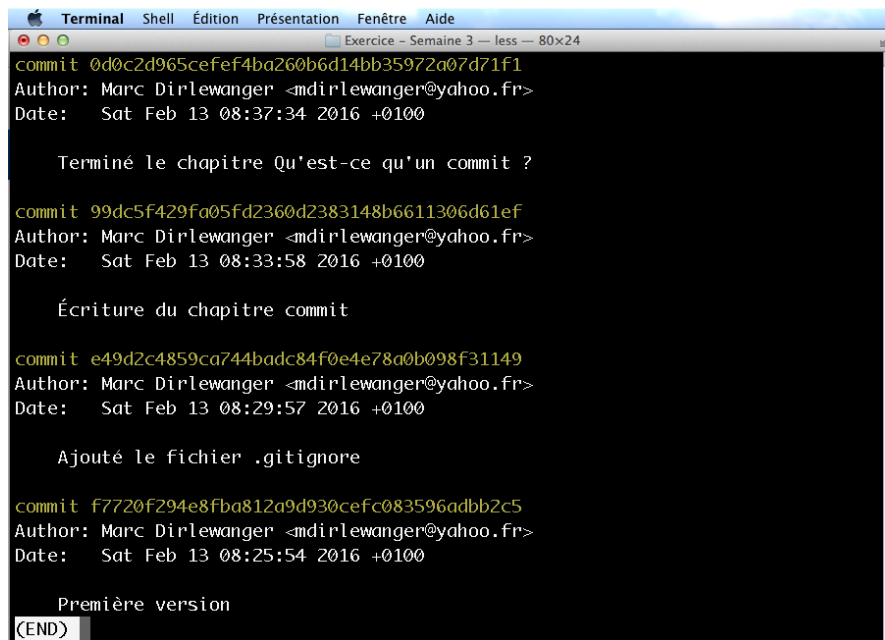
- effectué la commande *git init* dans le répertoire concerné
- ajouté le ou les fichiers avec la commande *git add*

Pour créer le *commit*, il faut ensuite utiliser la commande *git commit*.

```
mbpdemalewanger:Exercice - Semaine 3 admin$ git commit -am "Terminé le chapitre  
Qu'est-ce qu'un commit ?"  
[master 0d0c2d9] Terminé le chapitre Qu'est-ce qu'un commit ?  
1 file changed, 0 insertions(+), 0 deletions(-)  
mbpdemalewanger:Exercice - Semaine 3 admin$
```

## À quoi sert la commande `git log`

La commande `git log` sert à consulter l'historique des différents *commits* d'un projet.



```
Terminal Shell Édition Présentation Fenêtre Aide
Exercice - Semaine 3 — less — 80x24

commit 0d0c2d965cefef4ba260b6d14bb35972a07d71f1
Author: Marc Dirlewanger <mdirlewanger@yahoo.fr>
Date: Sat Feb 13 08:37:34 2016 +0100

    Terminé le chapitre Qu'est-ce qu'un commit ?

commit 99dc5f429fa05fd2360d2383148b6611306d61ef
Author: Marc Dirlewanger <mdirlewanger@yahoo.fr>
Date: Sat Feb 13 08:33:58 2016 +0100

    Écriture du chapitre commit

commit e49d2c4859ca744badc84f0e4e78a0b098f31149
Author: Marc Dirlewanger <mdirlewanger@yahoo.fr>
Date: Sat Feb 13 08:29:57 2016 +0100

    Ajouté le fichier .gitignore

commit f7720f294e8fba812a9d930cefc083596adb2c5
Author: Marc Dirlewanger <mdirlewanger@yahoo.fr>
Date: Sat Feb 13 08:25:54 2016 +0100

    Première version

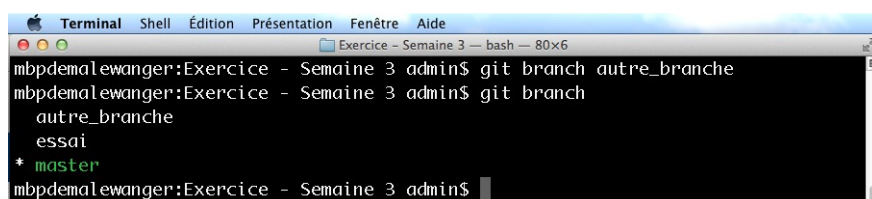
(END)
```

Sur cette copie d'écran, nous pouvons voir quatre commits effectués sur le texte que vous être en train de lire. Notez les différents messages de commit, qui permettent de savoir rapidement ce qui a été fait lors de chaque commit.

## Qu'est-ce qu'une *branche* ?

Je vais créer une nouvelle *branche* lorsque je désire avancer dans une certaine direction dans mon travail, mais sans être sûr d'appliquer ces changements par la suite.

Pour créer une branche, nous allons utiliser la commande `git branch`.



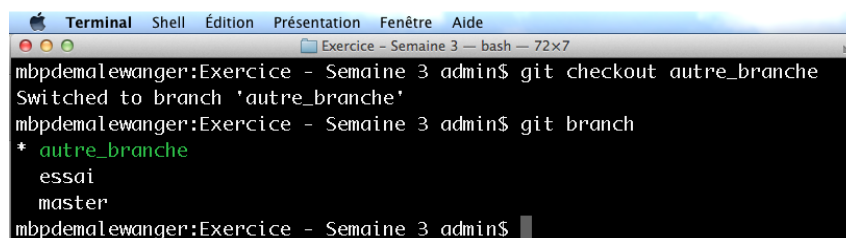
```
Terminal Shell Édition Présentation Fenêtre Aide
Exercice - Semaine 3 — bash — 80x6

mbpdemalewanger:Exercice - Semaine 3 admin$ git branch
mbpdemalewanger:Exercice - Semaine 3 admin$ git branch
autre_branche
essai
* master
mbpdemalewanger:Exercice - Semaine 3 admin$
```

La commande `git branch` avec un nom va créer une nouvelle branche.  
La commande `git branch` seule affichera la liste des branches disponibles.

Chaque branche pourra avoir ses propres *commits*, différents de ceux des autres branches.

La commande `git checkout` permet ensuite de se placer sur l'une ou l'autre branche.



```
Terminal Shell Édition Présentation Fenêtre Aide
Exercice - Semaine 3 — bash — 72x7

mbpdemalewanger:Exercice - Semaine 3 admin$ git checkout autre_branche
Switched to branch 'autre_branche'
mbpdemalewanger:Exercice - Semaine 3 admin$ git branch
* autre_branche
essai
master
mbpdemalewanger:Exercice - Semaine 3 admin$
```