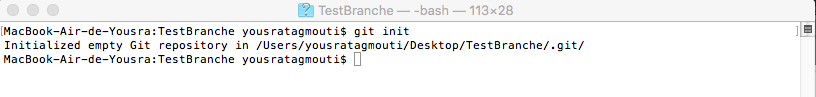
**Gestion des branches**

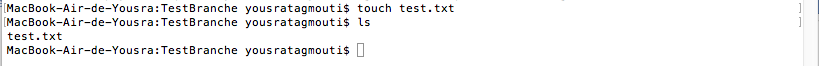
1. **Création des branches**

L’une des grandes forces de git est la gestion des branches

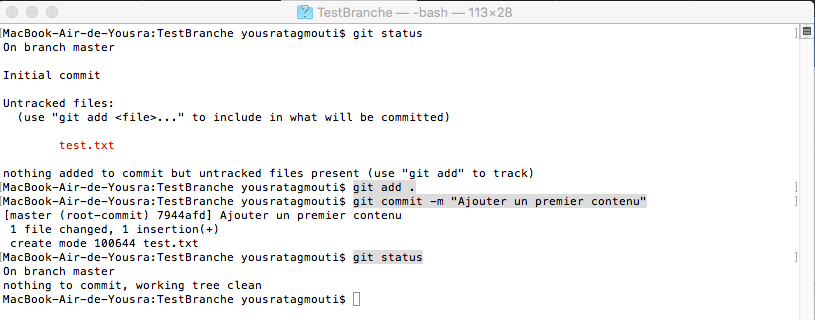
1. Pour tester les branches nous allons travailler sur un nouveau répertoire (**TestBranche** par exemple)
2. Initialiser le répertoire avec **git init** tel que ci-dessous :



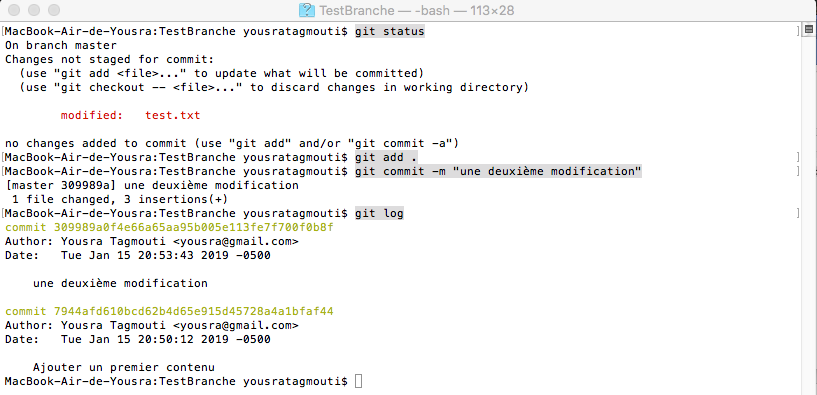
1. Ajouter un fichier texte dans ce répertoire :



1. Ajouter ensuite un texte de votre choix au fichier test.txt, et faire votre premier commit, pour un résultat tel que ci-dessous :



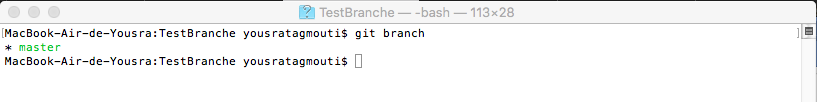
1. Ajouter ensuite un autre texte et faire un autre commit, pour un résultat tel que ci-dessous :



* À ce niveau vous allez partir sur une nouvelle branche faire des modifications pour décider par la suite si vous voulez remettre ces modifications dans la branche principale ou pas.

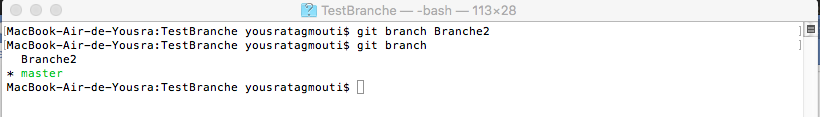
1. Vous pouvez voir la branche sur laquelle vous êtes avec la commande :

**git branch**



1. Créer une nouvelle branche du nom **Branche2** avec la commande :

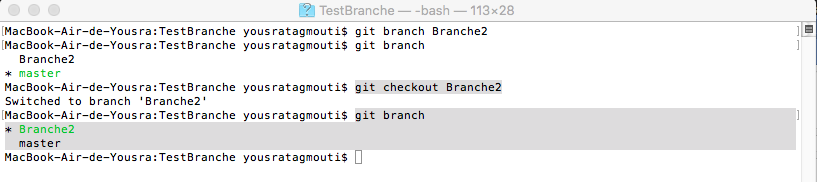
**git branch Branche2**



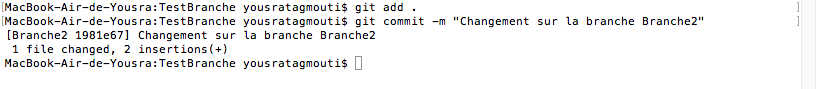
* Vous pouvez remarquer que vous avez maintenant deux branches

1. Changer de branche pour passer de la branche **master** à la branche **Branche2,** avec la commande ci-dessous :

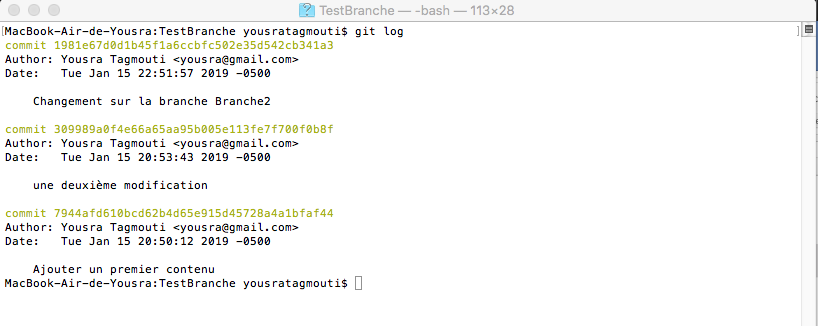
**git checkout Branche2**



1. Faire un changement de votre choix sur le fichier **test.txt** et faire ensuite un commit



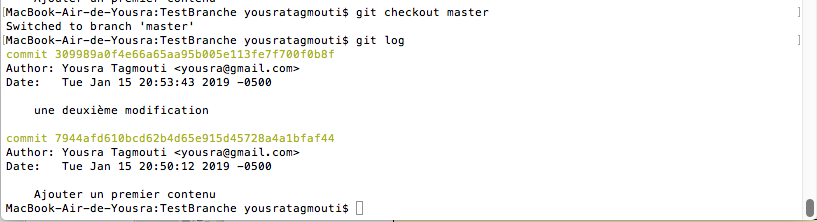
1. Si vous faites maintenant un **git log** vous pouvez voir les différents commits, tel que ci-dessous :



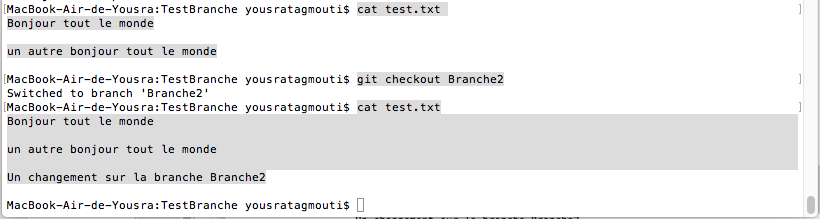
1. Vous pouvez voir la différence au niveau des commits entre les branches **master** et **Branche2**
2. Si les changements faits sur la branche **Branche2** ne vous conviennent plus, vous pouvez tout simplement basculer sur la branche master avec la commande :

**git checkout master**

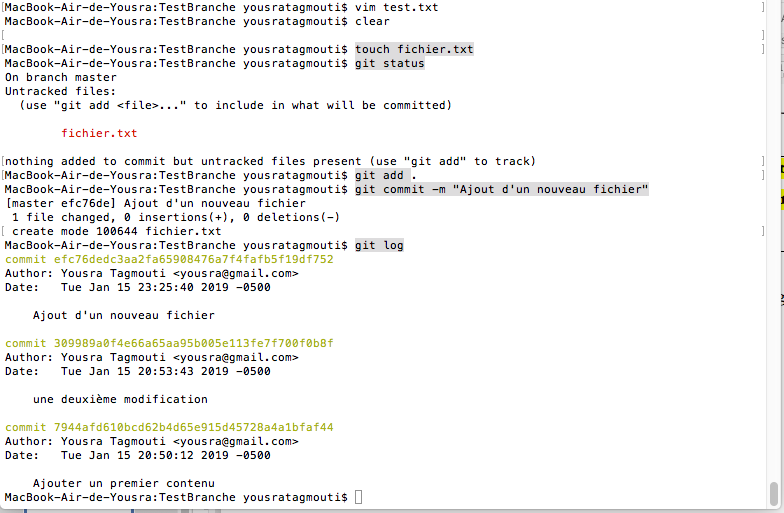
1. Vous pouvez **voir** au niveau des commits que les commits faits sur la branche **Branche2** n’apparaissent plus



1. Si vous regardez le contenu du fichier test.txt, vous pouvez voir que le contenu n’inclut pas l’ajout fait sur la branche **Branche2**



1. Retourner sur master pour faire un nouveau changement de votre choix, tel que ci-dessous, par exemple :



1. **Fusionner des branches**

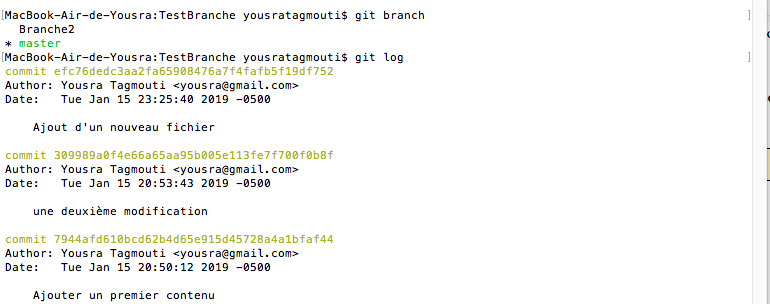
* Même si vous travaillez sur plusieurs branches, à la fin d’un projet toutes branches utiles doivent êtres fusionnées

1. Faire un merge des branches **master** et **Branche2,** avec la commande ci-dessous :

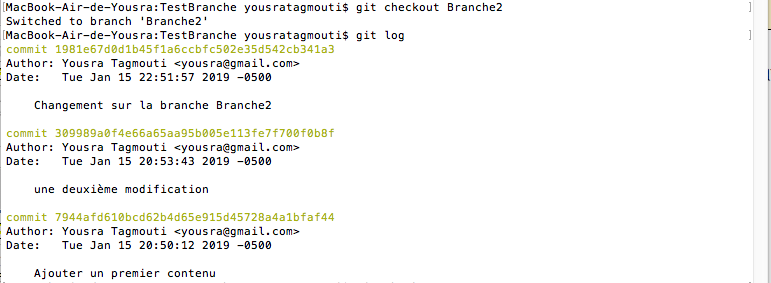
**git merge Branche2**

1. Vous pouvez ainsi voir le regroupement des commits des deux branches, tel que ci-dessous :

L’ensemble des commits de la branche **master**



L’ensemble des commits de la branche **Branche2**



Après la fusion des deux branches, vous pouvez voir l’ensemble des commits dans la branche **master**

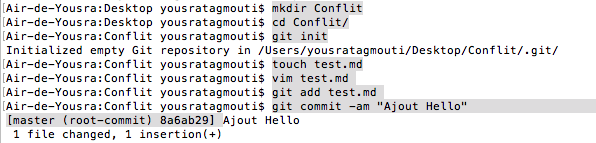


1. **Résoudre un conflit**

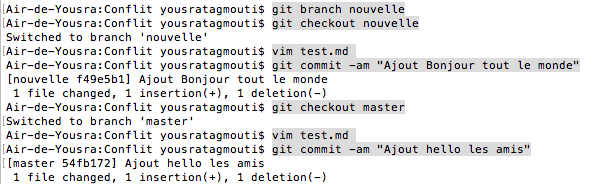
Si vous travaillez à plusieurs sur le même fichier et mêmes lignes et que vous voulez faire une fusion (merge), il y a un risque de conflit au moment de la fusion.

**Créer une situation de conflit**

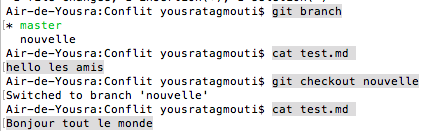
1. Créer un nouveau répertoire de projet **Conflit**
2. Créer sur la branche **master** un fichier **test.md** tel que ci-dessous :



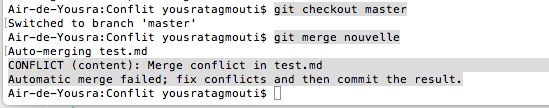
1. Ajouter une nouvelle branche **nouvelle** et modifier le contenu du fichier **test.md** sur la première ligne



1. Vérifier que votre scénario ressemble à ci-dessous :



1. Le merge se fera à partir de la branche **master** et causera un conflit tel que ci-dessous :



1. Si vous ouvrez maintenant le fichier **test.md** avec n’importe quel éditeur vous pouvez voir :

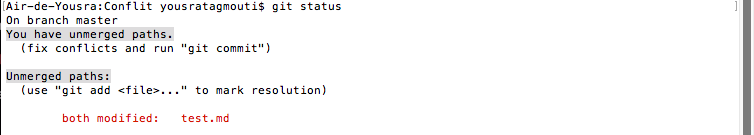


1. Maintenant il faut choisir quoi garder, vous pouvez même garder les deux lignes en modifiant directement le contenu tel que ci-dessous :

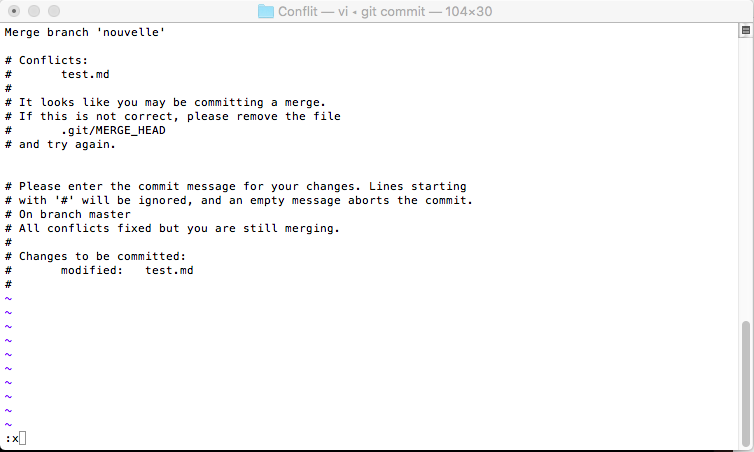
hello les amis

Bonjour tout le monde

1. Enregistrer vos modifications
2. Maintenant vous pouvez faire un git status pour un résultat tel que ci-dessous :



1. Remarquez les informations sur le merge
2. Maintenant vous pouvez faire un add classique et un commit sans commentaire et vous vous retrouvez sur un contenu tel que ci-dessous :



1. Vous pouvez enregistrer
2. À ce niveau, l’ensemble de vos commits ressemble au contenu ci-dessous :

