**Utilisation d’un dépôt distant sur GitHub**

Le remote est un dépôt distant qui peut servir à :

* Avoir un backup de votre travail, au lieu de tout garder en interne
* Permettre une collaboration sur un projet à plusieurs, de telle manière à ce que chaque personne puisse collaborer et avoir accès aux contributions des autres.
* Pour créer un remote allons utiliser GitHub

**Créer un compte sur GitHub**

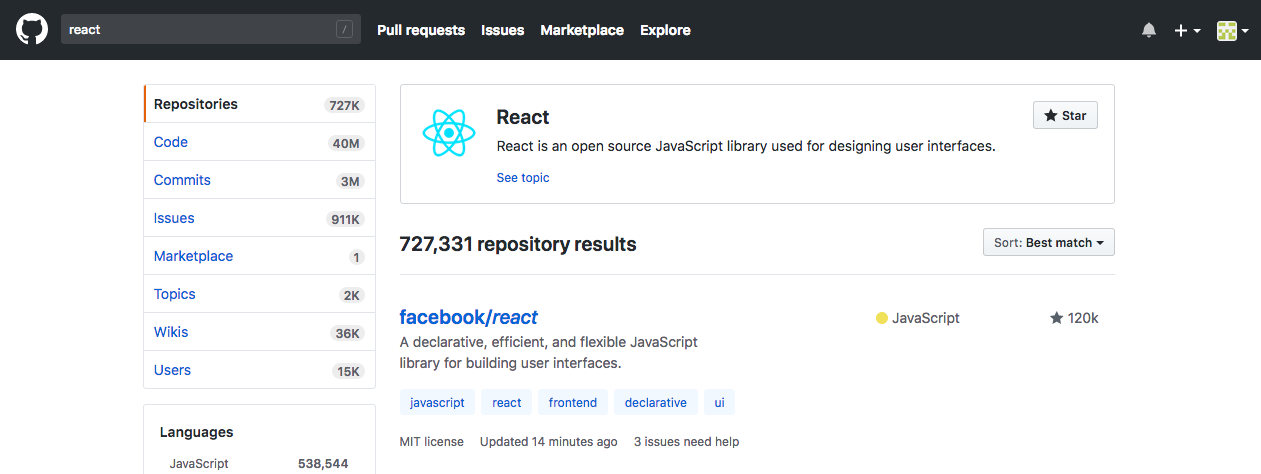
1. Rendez-vous sur GitHub : <https://github.com/>
2. Créez-vous un compte. Faites attention, le **username** est votre identité publique. Choisissez quelque chose qui vous ressemble, et gardez à l’esprit que ce compte pourrait servir dans votre emploi futur.

Nous allons voir quelques fonctionnalités intéressantes sur GitHub :

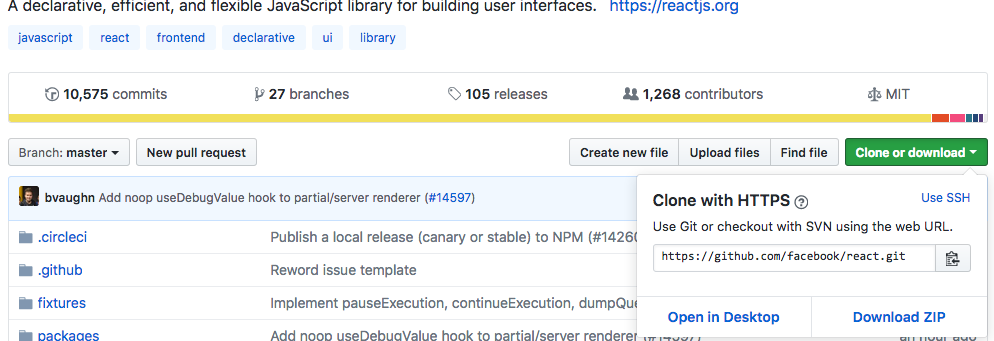
1. **Récupérer le contenu d’un répertoire de projet**

**Ils existent déjà plusieurs projets en ligne qui utilisent Git et ces projets sont hébergés sur GitHub, lorsqu’on essaie de récupérer le contenu de l’un de ces projets on appelle cela le clonage (clonning)**

1. Connectez-vous à GitHub pour récupérer l’URL du projet qui nous intéresse, avec l’objectif de créer une copie de ce projet sur votre machine (clonage).
2. Nous allons choisir de cloner une librairie public disponible sur GitHub **react** permettant d’améliorer le gestion d’interface graphique avec JavaScript.

****

1. Cliquer sur le nom du répertoire correspondant à la librairie **react** et ensuite sur le bouton **Clone or download**

****

1. Copier ensuite l’URL du projet, que vous allez utiliser sur le terminal de Git
2. Créer un répertoire **ProjetReact** dans le répertoire **simpleprojet** avec la commande ci-dessous :

**Mkdir ProjetReact**

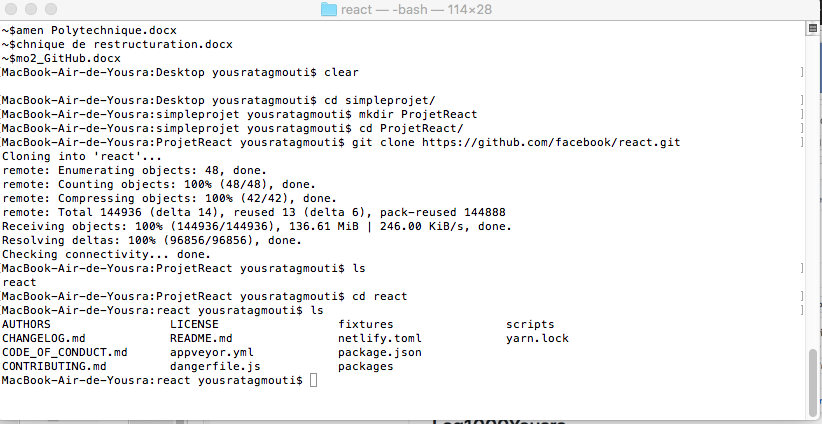
1. Placez-vous dans le répertoire **ProjetReact** avec la commande :

**cd ProjetReact**

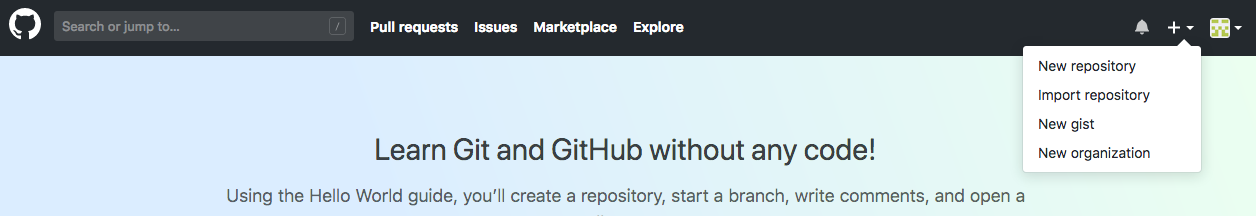
1. Cloner ensuite la librairie **react** dans ce répertoire en utilisant l’url du projet, avec la commande ci-dessous :

**git clone https://github.com/facebook/react.git**

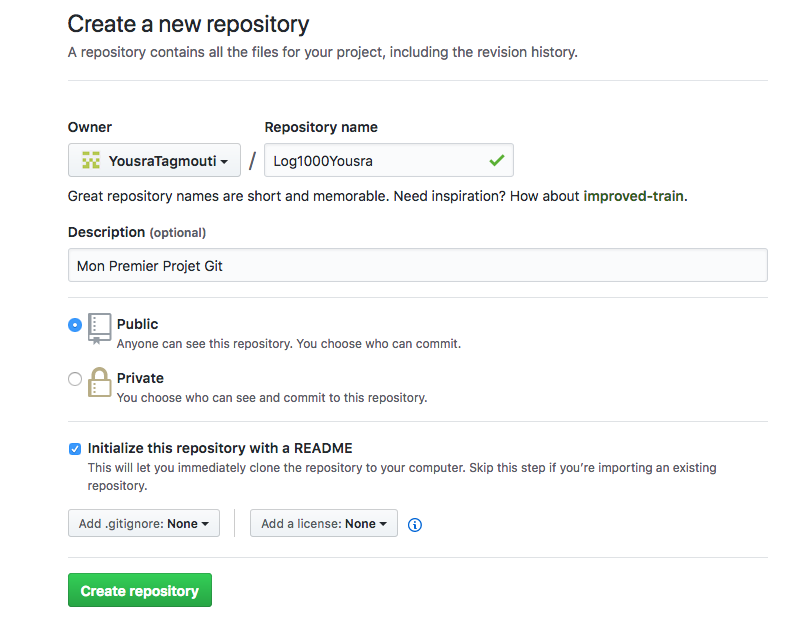
Après chargement, si vous listez le contenu de votre dépôt local se sera le même que celui disponible en ligne sur GitHub



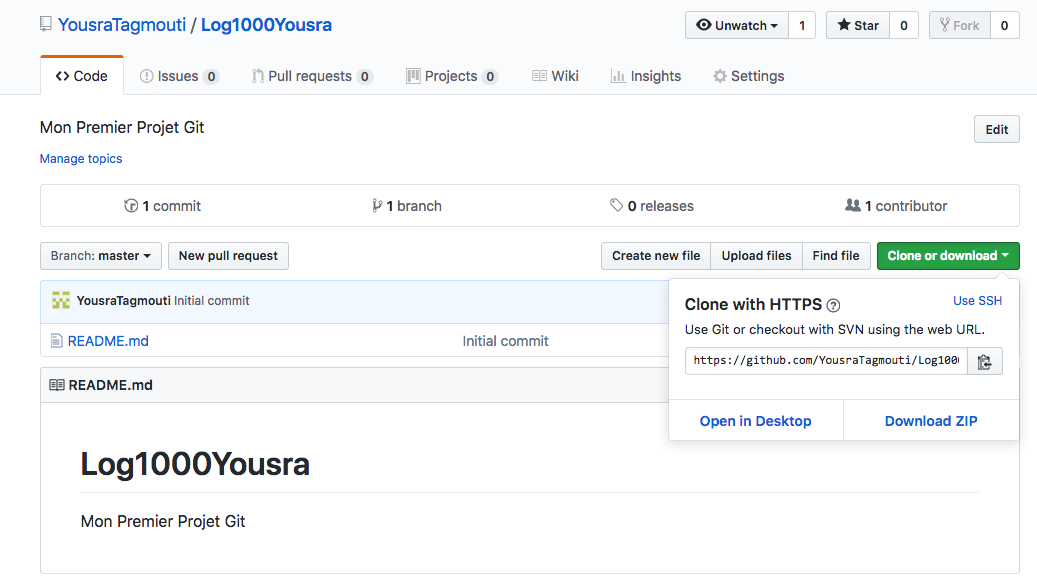
1. **Créer un répertoire de travail sur GitHub**
2. Connectez-vous à GitHub et cliquez sur **New repository**

****

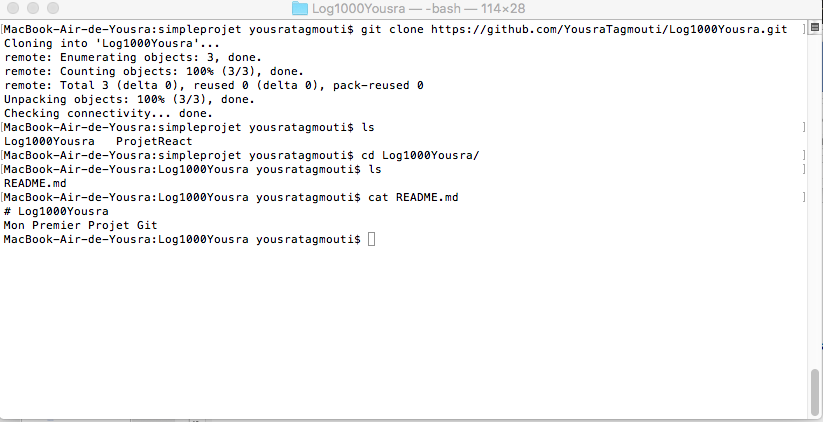
1. Donner un nom et une description à votre répertoire de projet



1. Il faut mettre votre projet en public (si vous voulez que votre projet soit privé il faut payer)
2. Initialisez votre projet avec un README (l’utilité est d’avoir un premier fichier dans votre répertoire, si vous voulez le cloner sur votre machine)

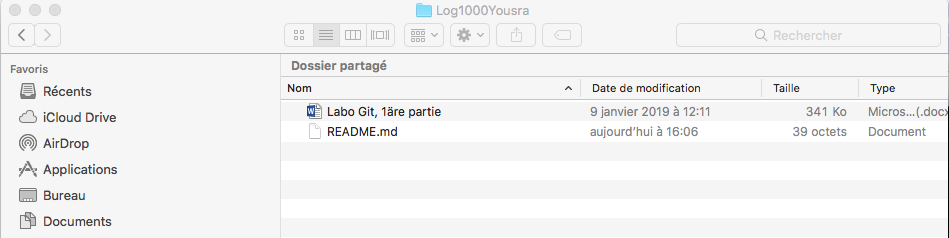


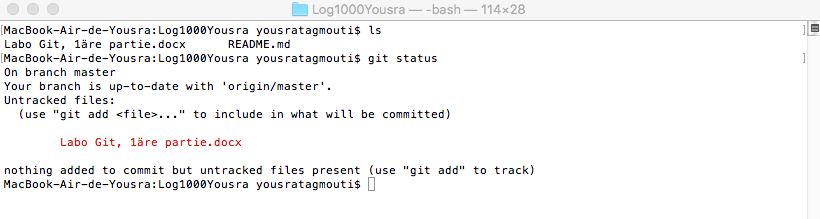
1. Vous pouvez ensuite cloner votre nouveau répertoire sur votre machine…vous savez maintenant comment faire !



Pour cloner un projet vous avez la possibilité d’utiliser une authentification http, tel que ci-dessus ou l’option SSH qui est plus pratique, mais pour cela il faut générer une clé SSH. Le lien suivant : <https://help.github.com/articles/connecting-to-github-with-ssh/> vous explique toutes les étapes à suivre.

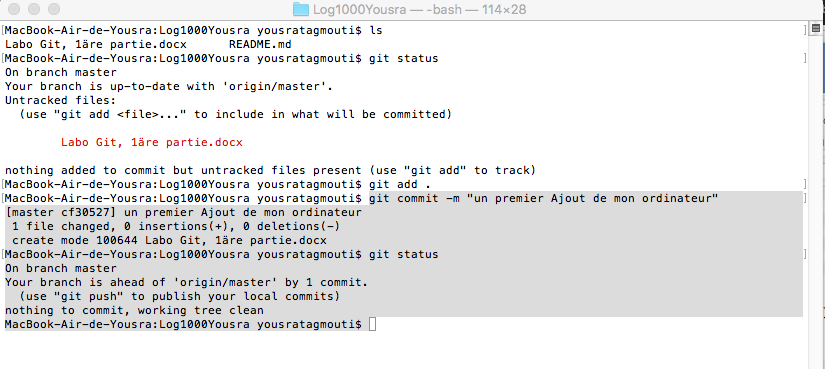
1. **Contribuer à un projet sur GitHub**
2. Une fois que vous avez copier un projet sur votre ordinateur, vous travaillez un peu dessus, ensuite l’idée est de partager votre travail sur le Web … non pas sur Facebook ! mais sur GitHub.
3. Ajoutez au répertoire créé à l’étape précédente (pour moi c’est Log1000Yousra) un fichier de votre choix, dans mon cas c’est le fichier Word **LaboGit,1ere partie.docx** tel que ci-dessous :



1. Si vous regardez maintenant le statut de votre projet je pouvez voir un changement, tel que ci-dessous :
2. Tel que vous devez le savoir à ce niveau, vous devez faire un git add pour indexer vos changements et ensuite un commit, tel que ci-dessous :

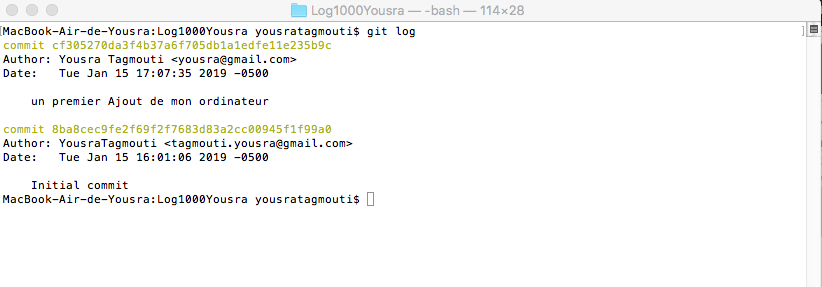
**git add .**

**git commit -m "un premier Ajout de mon ordinateur"**



1. Si vous voulez voir vos commits vous pouvez faire la commande :

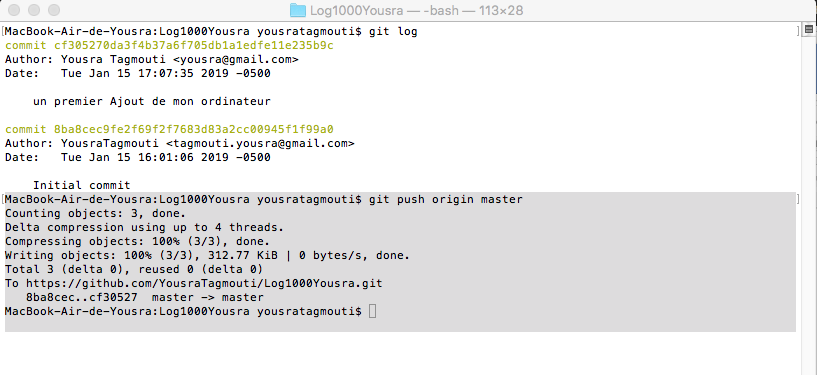
**git log**



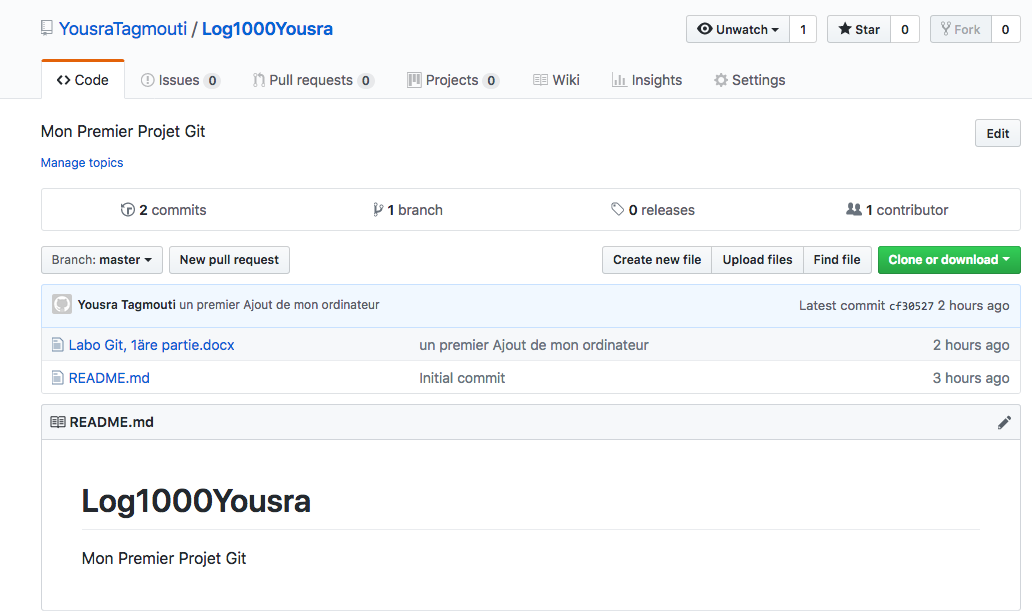
1. Vous pouvez envoyer vos changements sur GitHub avec la commande ci-dessous :

**git push origin master**

* **origin**: est le nom donné au remote par défaut, ceci dit nous pouvons créer d’autres remote si nous voulons envoyer nos changements sur d’autres destinations
* **master**: est le nom de la branche par défaut sur laquelle vous travaillez



1. Maintenant vous pouvez voir les changements sur votre répertoire (**Log100Yousra**) tel que ci-dessous :



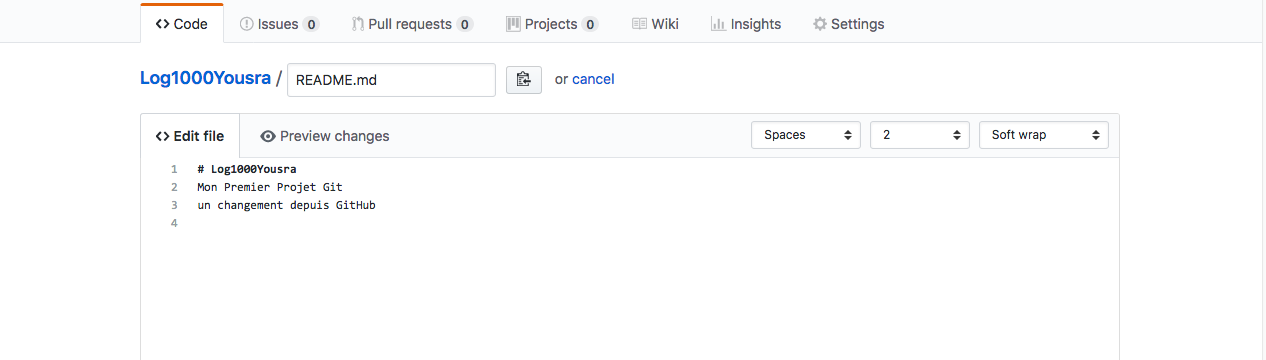
Vous pouvez remarquer :

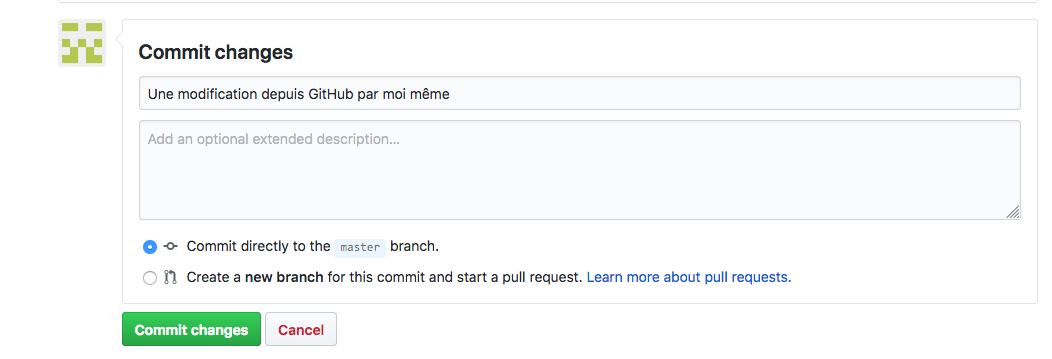
* L’ajout du fichier Word
* L’ajout d’un nouveau commit
* **Les commits sur GitHub sont les mêmes que nous pouvons obtenir avec la commande git Log,**
* **Sauf que le format sur GitHub est plus agréable et**
* **Si vous cliquez sur un commit sur GitHub vous pouvez voir les détails d’ajouts sur chaque commit**

1. **Récupérer les changements sur GitHub**
2. Vous voulez récupérer sur votre dépôt local les changements faits par exemple par une autre personne sur le projet

* **Tel que vu à l’étape précédente, nous pouvons cloner le répertoire du projet,**
* **Mais cloner le répertoire d’un projet copie l’ensemble du contenu**
* **Sauf qu’à ce niveau nous voulons uniquement copier les changements**
* **Pour récupérer les derniers changements c’est à dire les derniers commits fait sur le projet nous avons deux possibilités utiliser git pull ou encore git fetch avec git merge**
* **Imaginons que vous et votre ami travaillés sur le même projet sur GitHub, vous faites des push et votre ami aussi, mais vous savez que vous devez avoir les derniers changements sur votre dépôt local**

1. Vous pouvez faire une simulation de ce scénario en modifiant par vous-même un document du projet directement sur GitHub, tel que le fichier **README** par exemple



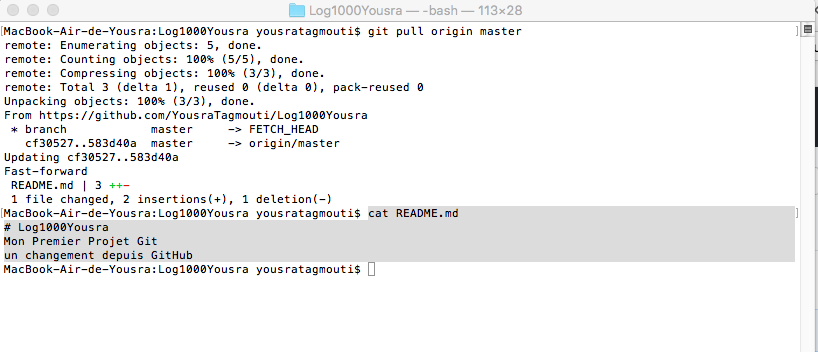


**À ce niveau les changements sont uniquement sur GitHub et non sur votre dépôt local**

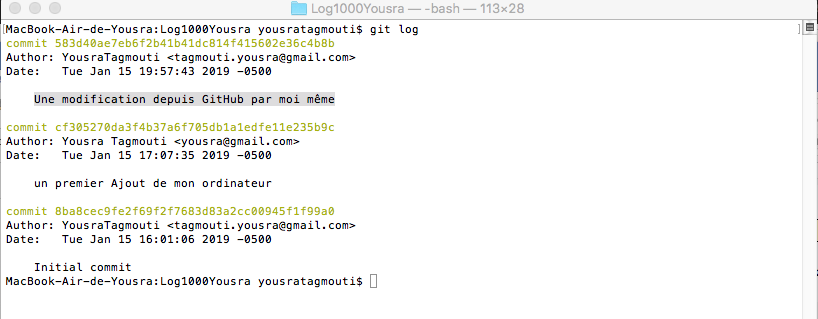
1. Pour pouvoir récupérer les modifications de README sur votre dépôt local vous pouvez faire par exemple la commande ci-dessous :

**git pull origin master**

1. Vous pouvez voir les changements sur le fichier README de votre dépôt local

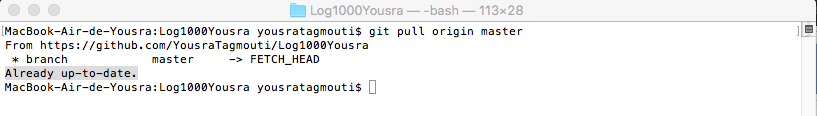


1. Vous pouvez aussi voir que le nombre de commits a également changé

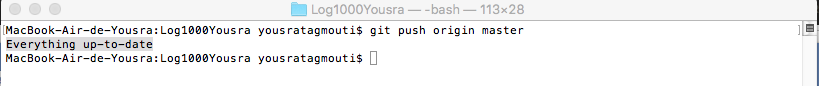


* **Il est important de vous synchroniser avec le dépôt distant le plus souvent possible**
* **Vous ne pouvez pas faire de push sur Git Hub tant que vous n’êtes pas synchroniser avec le dépôt distant**

1. Si maintenant vous faites un git pull, vous avez un message qui vous informe que vous êtes à jour, tel que ci-dessous :

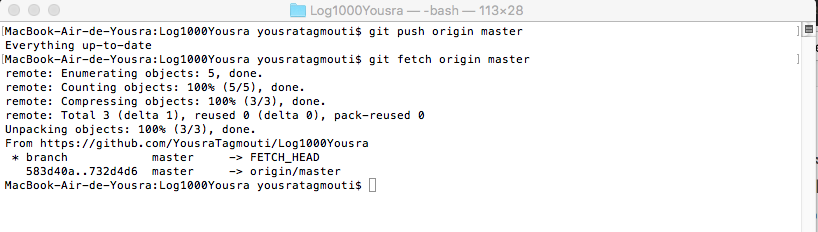


Vous avez la même information si vous faites un git push, tel que ci-dessous :

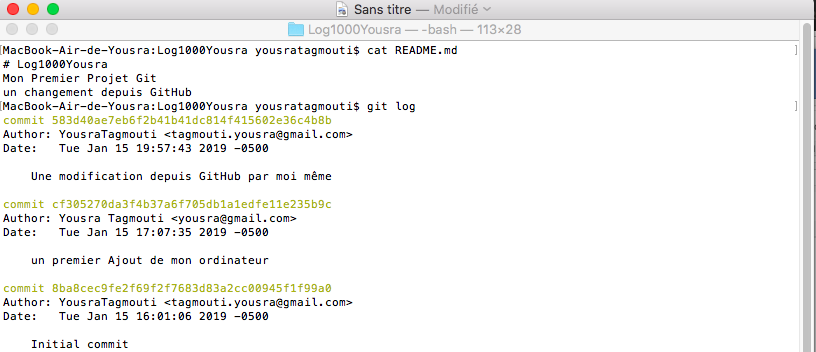


1. Apportez d’autres changement au fichier README, tel que dans l’étape précédente.
2. Vous pouvez faire la même chose qu’avec un **pull** avec les commandes ci-dessous :

**git fetch origin master**



Remarquez que vous n’avez encore rien récupéré jusqu’à maintenant



Pour voir un changement vous devez faire un merge tel que ci-dessous :

**git merge origin master**

