

# Notes de cours Git-GitHub

## Les commandes de bases sur Git

Configurer compte avec Github :

```
git config --global user.name "tmarmey"
```

```
git config --global user.email "tmarmey@gmail.com"
```

## Activer un repository Git, se placer dans le dossier que l'on veut et faire :

```
git init
```

## L'activation du repository Git génère un index.

Lorsqu'on rajoute un nouveau fichier dans le repository il faut rajouter ce fichier à l'index par la commande :

```
git add nomDuFichier.extension
```

## Ajouter tous les fichiers du dossier courant :

```
git add .
```

## Enregistrement des modifications dans le rep

```
git commit -m "ajout de fichier" (le -m permet de rajouter un message au commit)
```

## Pour se positionner sur un commit de l'historique

```
git checkout SHADuCommit
```

## Revenir au commit le plus récent

```
git checkout master
```

## Créer un nouveau commit qui fait l'inverse du commit précédent

```
git revert SHADuCommit
```

## Modifier le message du dernier commit

```
git commit --amend -m "votre message"
```

## **Annuler tous les changements qui ne sont pas encore commités**

git reset - -hard

## **Pour créer une nouvelle branche**

se placer au commit voulu avec checkout et faire :

git checkout -b nouvelle-branche

## **Pour supprimer une branche**

se placer à l'extérieur de celle ci et faire :

git branch -d nom-de-la-branche

## **Copier un repository depuis GitHub**

git clone adresseDuRepository

## **Rebase**

git rebase master bug1

transplante bug1 sur l'actuelle branche master. Si on est déjà en train de bosser sur bug1 on peut se contenter de taper git rebase master

git checkout master

Switche sur master

git merge bug1

Fusionne bug1 dans master

git branch -d bug1

fusionne bug1 dans master

## **Savoir qui a modifié un fichier**

git blame nomDuFichier.extension

## **Voir les détails du commit**

git show numSHA

## **Ignorer des fichiers dans Git**

Telles que des varibales de configuration, des mots de passe, etc :

utiliser le fichier *.gitignore*

Indiquer leur chemin complet et sauter une ligne entre chaque fichier

## Eviter les commits superflus

En cas “d’urgence”, si la tâche actuelle n’est pas fini et donc ne nécessite pas de commit, mais qu’une autre tâche doit être faite, il faut mettre de côté ce qui est entrain d’être fait, pour pouvoir y revenir après. Utiliser :

```
git stash
```

(permet de mettre en pause les modifications pour travailler sur un autre endroit du projet)

```
git stash pop
```

(permet de revenir à l’endroit où vous étiez arrêtés, ici pop va effacer le stash donc si l’on veut refaire une pause etc... il faut ainsi refaire un *git stash*)

```
git stash apply
```

Si l’on veut garder les modifications dans le stash utiliser :

## Les fonctionnalités de bases sur GitHub

Créer un nouveau repository sur GitHub et celui ci n’est pas encore créé sur notre machine : cocher l’option “initialise with README”

## Pour envoyer ses modifications sur GitHub

il faut d’abord faire ses commits et ensuite faire :

```
git push origin master
```

(origin constitue le remote choisi, par défaut c’est origin et pour nous c’est github, mais si nous avons plusieurs remotes il faudrait spécifier et donc changer origin)

## Ajouter clef SSH pour push sans identifiant et mot de passe

- \$ ssh-keygen -t rsa -b 4096 -C "[your\\_email@example.com](mailto:your_email@example.com)"
  - Enter a file in which to save the key (/home/you/.ssh/id\_rsa): [Press enter]
  - Start the ssh-agent : \$ eval "\$(ssh-agent -s)"
  - \$ ssh-add ~/.ssh/id\_rsa
- Ajouter la clef au compte GitHub :
- copier la clef avec xclip : \$ sudo apt-get install xclip
  - \$ xclip -sel clip < ~/.ssh/id\_rsa.pub
  - Settings/SSH and GPG keys
  - New SSH key et Add SSH key
  - \$ ssh -T [git@github.com](mailto:git@github.com)

## Pour récupérer les dernières modifications

```
git pull origin master
```

## **Contribuer à un projet :**

- forker le repo de l'auteur (c'est à dire copier le repo sur votre compte GitHub)
- cloner le fork sur votre machine
- Créer une nouvelle branche
- Faire des modifs sur cette nouvelle branche
- Envoyer les modifications sur GitHub sur votre propre branche
- Proposer votre contribution au projet "pull request"