GIT

Les vidéos du site openclassroom sont très bien

https://openclassrooms.com/courses/gerer-son-code-avec-git-et-github

Je vous propose d’aller regarder ce que lucas a fait c’est très bien

<http://bioinfo-fr.net/git-premiers-pas>

<http://bioinfo-fr.net/git-usage-collaboratif>

# 1)Pour installer GIT et l’utiliser

## 1-1) Installation

Aller le télécharger ici (sous linux un *sudo apt-get install* *git-core* devrait suffir ^^)

<http://git-scm.com/downloads>

Ouvrir le fichier télécharger et suivre les instructions en laissant toutes les valeurs par default.

## 1-2) Configuration

Une fois l’installation terminée, aller sur le terminal de l’ordi pour définir notre nom et email qui seront utile pour créer le compte gratuit GitHub:

git config --global user.name "Votre nom"

git config --global user.email "you@example.com"

Pour avoir de la couleur

git config --global color.diff auto

git config --global color.status auto

git config --global color.branch auto

pour vérifier si ça a marcher

vim ~/.gitconfig

## 1-3) Créer son 1er commit

créer un fichier

ce placer dans ce fichier a l’aide du terminal

création d’un répertoire git :

git init

Mettre dans notre nouveau dossier les fichiers concernés par le travail

Pour vérifier ce qui a été fait

git status

vérifier les changement effectuer sur son ordinateur

git status

en rouge les fichiers modifier

en vert les fichiers ajouté sur git

git add nom\_du\_fichier\_rouge

permet de faire passer un fichier rouge en vert

git commit –m « message concernant le type d’ajout fait avec git add »

ici on enregistre l’ajout effectuer avec le git add précédent en y ajoutant un petit message d’info qui concerne les modifications effectuées.

# 2)Utilisation de GitHub

## 1-1)S’inscrire si ce n’est pas déjà fait sur le site

<https://github.com>

## 1-2)Clonage de répertoire.

Aller sur le dossier projet\_ADT

Cliquer sur HTTPS

Et copier le lien

Allez dans sa console

Ce rendre dans son dossier concernant le projet

Puis faire un

Git clone https://github.com/VictorYt/P\_ADT\_r\_BIG\_2015.git

la console demandera surement votre login et mot\_de\_passe du site GitHub (si c’est votre permier clone)

avec la commande

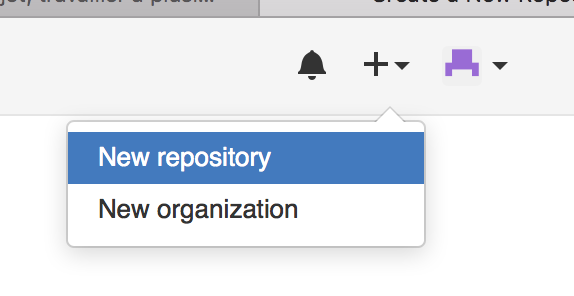
ls

on voit alors que le fichier cloner est bien présent. (pout nous Projet\_ADT)

## 1-3)Mettre un repertoire sur GitHub (pas necessaire si j’en fait dejà un)

Allez sur son compte GitHub

Appuyer sur le + et « New repository » (comme ci dessous)



Puis donner un nom et choisir la categorie public (sinon c’est payant)

Bref je l’ai deja fait car vous aurrait juste à le récupérer

# 3) Recevoir et envoyer les modifications

## 3-1) récupérer les modifications

il suffit tout simplement de faire un

git pull

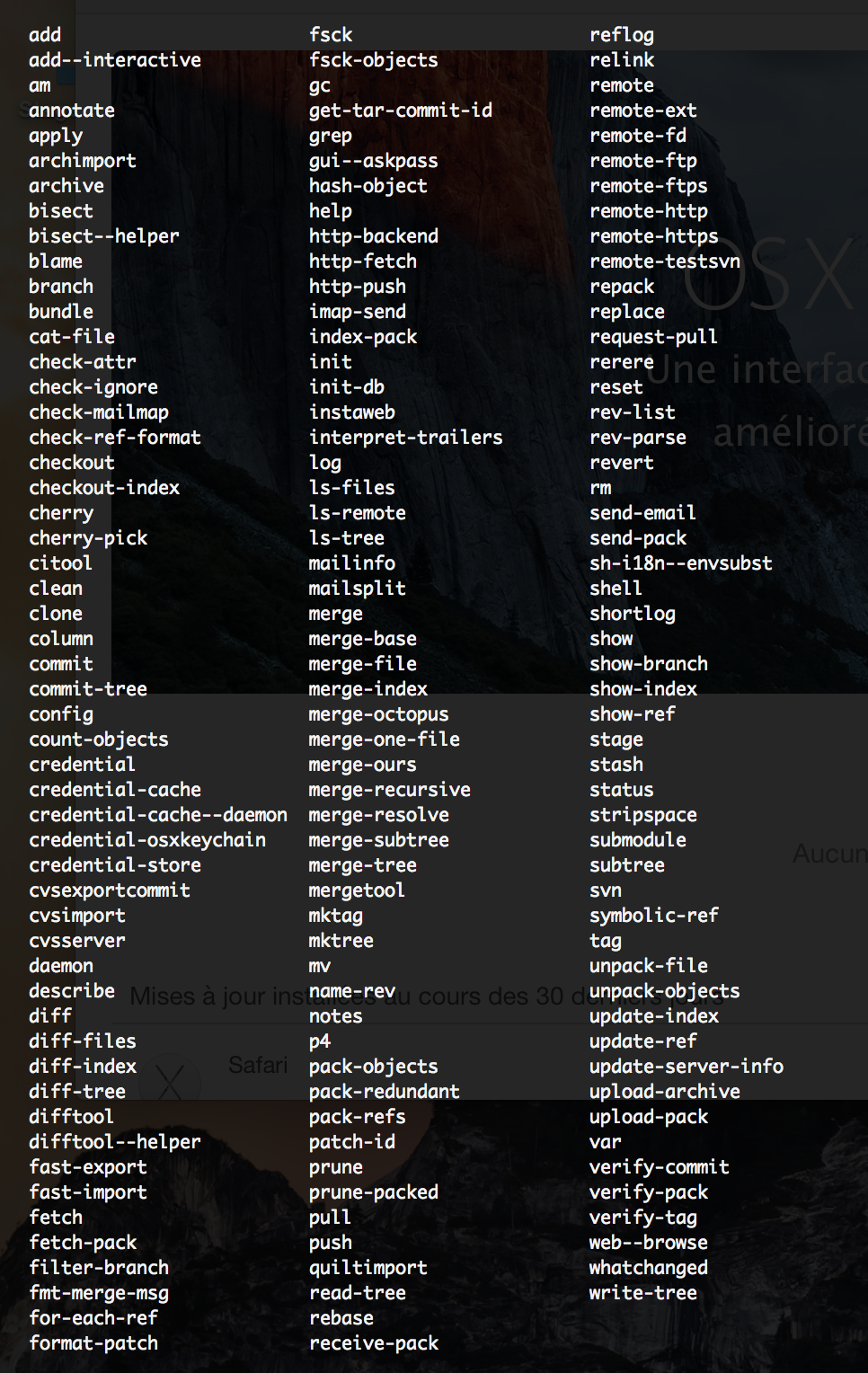
Si le git clone a fonctionné avant il ramènera tous les fichiers concerner par le clone (en gros tous ce qui a été modifier dans notre projet sera mis a jour sur votre ordi)

## 3-2)Envoyer les modifications

C’est le total inverse de la commande précédente

**LES COMMANDES A ABSOLUMENT CONNAITRE :**

git help –a pour avoir les différentes commandes



vérifier les changement effectuer sur son ordinateur

git status

en rouge les fichiers modifier

en vert les fichiers ajouté sur git

git add nom\_du\_fichier\_rouge

permet de faire passer un fichier rouge en vert

git commit –m « message concernant le type d’ajout fait avec git add »

ici on enregistre l’ajout effectuer avec le git add précédent en y ajoutant un petit message d’info qui concerne les modifications effectuées.

git commit –a –m « message concernant le type d’ajout fait avec git add »

ici le –a permet de ne pas avoir a refaire un git add si le fichier a déjà été add et que des modification l’on déjà concerné

git log

Permet d’avoir l’historique des ajouts, author, date, message. Et le sha (identifiant du git)

La modification la plus récente est tout en haut

Pour quitter le log appuyer sur « q »

git checkout sha sur lequel on souhaite revenir

ça permet de revenir sur un git plus vieux pour lequel il n’y avait pa encore d’erreur.

Le sha est le code après commit lorsque l’on affiche le git log.

git checkout master

Permet a un git checkout de revenir au dernier git ajouté (le plus recent) après être revenue sur un plus vieux avec la commande ci-dessus.

git help –g Et voilà pour le guide

