DOCUMENTATION OF GITHUB SERVICE

Brice POIRIER & Ronan JAMES – 2019

DESCRIPTION

Git is a Cloud-based team documentation. GitHub is the most know service, and is widely use for all professionals developers. In this document we will explain how to get start and use efficiently this service.

CONTENTS

I. GETTING STARTED WITH GITHUB

1. CREATE A GITHUB ACCOUNT

a. Go to github.com

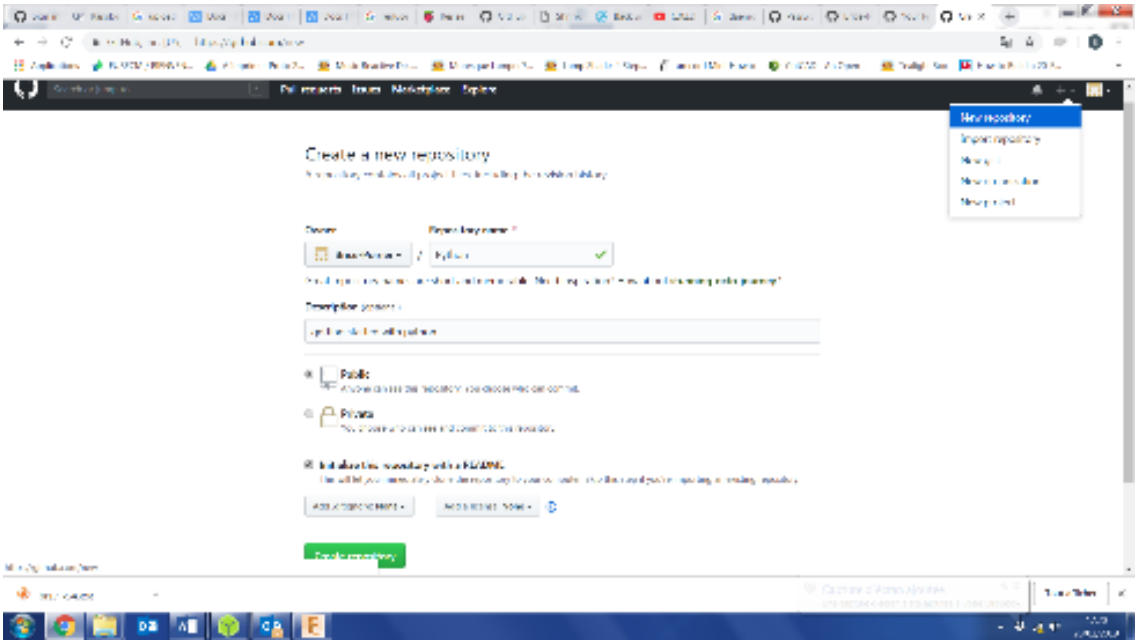
b. Sign up and create an account

c. Sign in to access your account

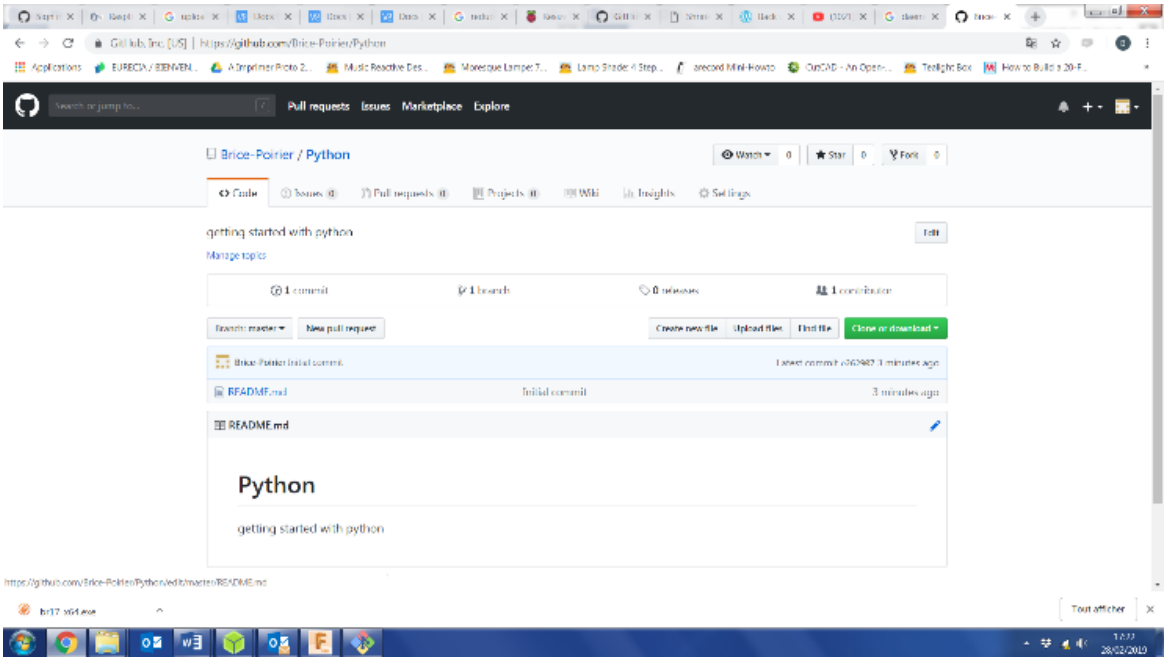
2. CREATE A GITHUB REPOSITORY

   a. Set up a new repository by clicking on the + icon, top right of the page

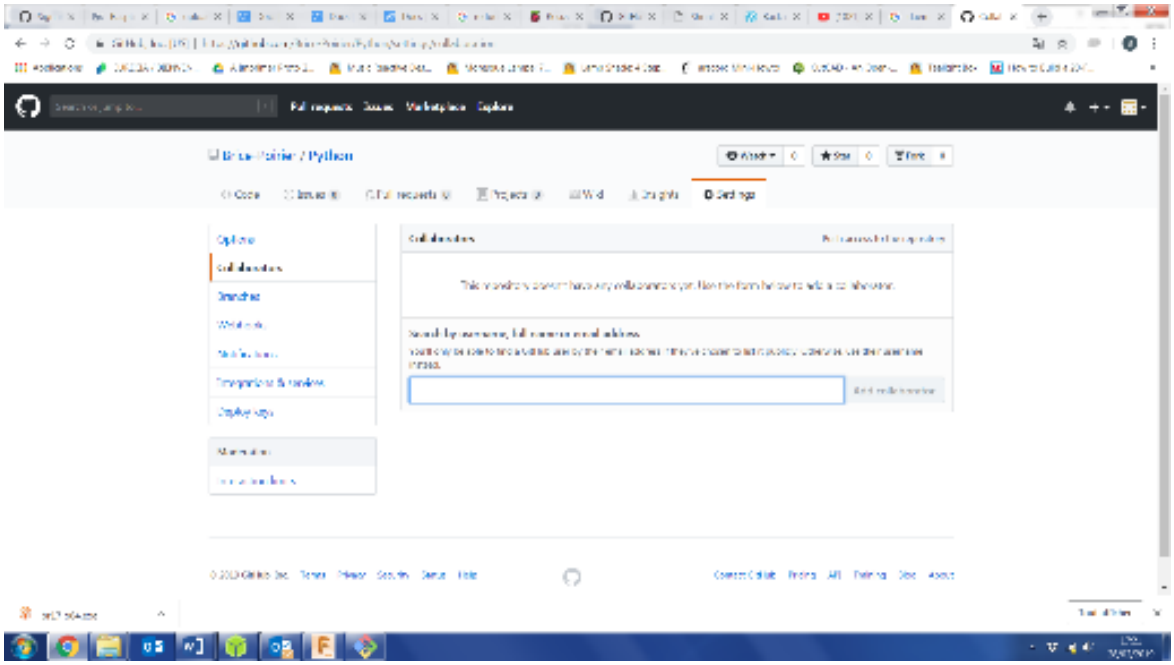
   b. **Give a name to your repository**, and Describe its purpose



   c. You end up on this screen



d. Invite other users to collaborate to your repository in the section “Settings / collaborators”



3. INSTALL GITHUB ON YOUR COMPUTER

Let us move to the local side of git. What git does is that it synchronize and allow different versions of your code to coexist. But we need a software to interface the local version and the cloud.

a. Go to https://git-scm.com/

b. Download the version that suits your system

c. Install it, just say yes to the already ticked option

d. Once installed, either lick on the icon on your desktop or look for git-bash in the menu and launch it. Where / what is git bash ?

e. Create a Git folder on your computer at the root of a partition.
CAUTION : that folder will be your local git hub, where all the repos will be stored.

f. Go back to git bash and go to your git folder using :
Eg. If git is in your C partition roo, you can type cd /c/Git
CAUTION : it is case-sensitive!

g. If you log for the first time, Log in using those two commands :
git config --global user.name “Name”
git config --global user.email "youremail@domain.com"

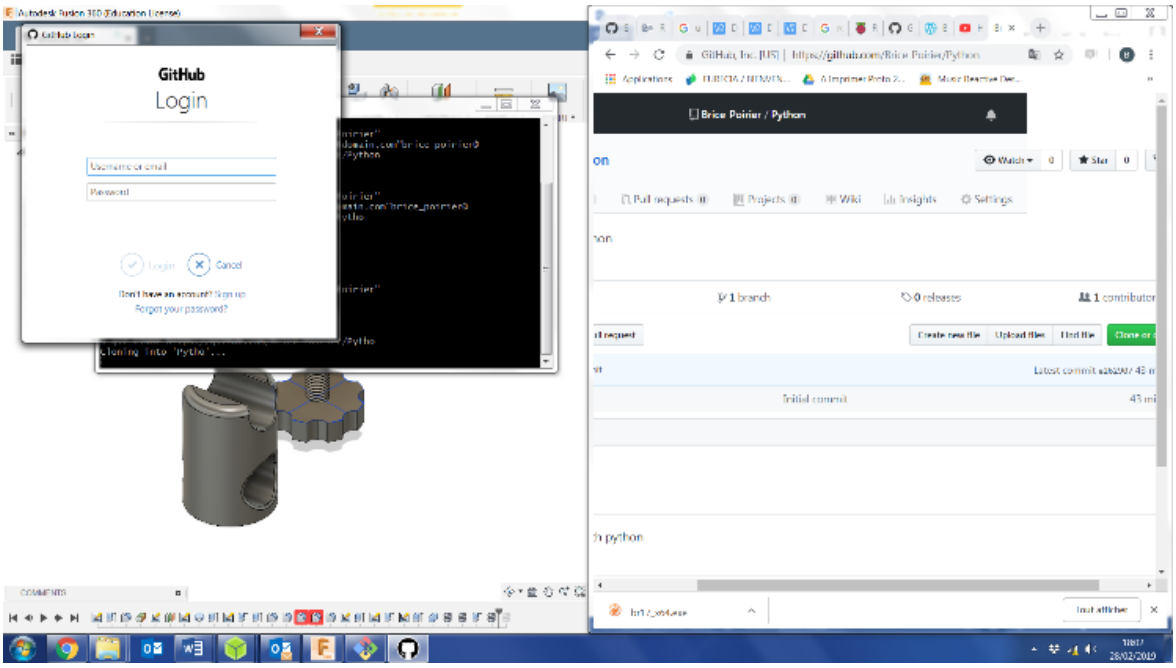
II. USE VERSIONNING WITH GITHUB

1. CLONE YOUR REPOSITORY ON YOUR GITHUB FOLDER

a. Go to your Github folder
In our exemple, cd /c/Git

b. Copy the URL of your repository, and clone it :
git clone “https://github.com/PROTO204/Students-SETI-2019"

CAUTION : The first time you will, a window will pop-up and ask for your credentials; fill in your login and password and it’ll start downloading your online repo onto the Git folder



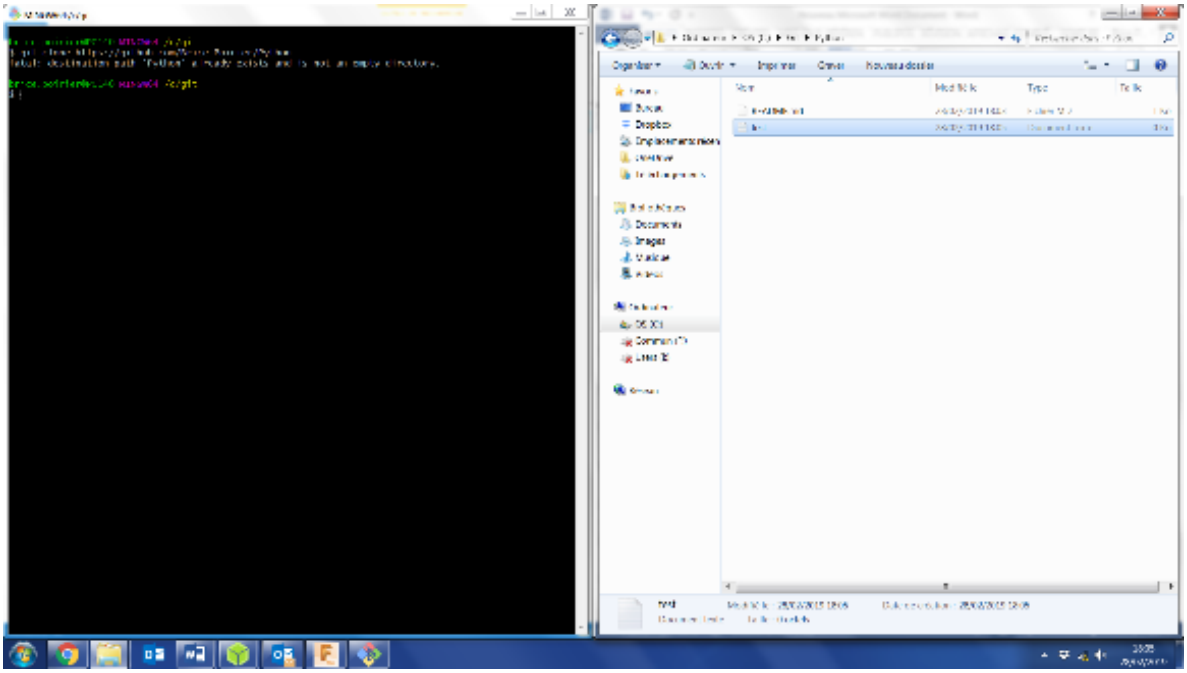
2. KEEP YOUR LOCAL REPOSITORY UP DATED WITH THE ONLINE REPOSITORY

Github does not synchronize automatically, so you have to check that your local repository is up to date when to code on your computer. The command is straight forward:

a. git pull “https://github.com/PROTO204/Students-SETI-2019"

3. COMMIT THE ONLINE GITHUB REPOSITORY

In this example, we have created a file test.txt in our local folder, and we want to send it to the online Github repository.



a. First, we need to make Git aware of the files inside the folder and tell it those files are relevant to the project.

```
cd Python
git add (filenames.ext)
```

b. Make sure your command was taken into account

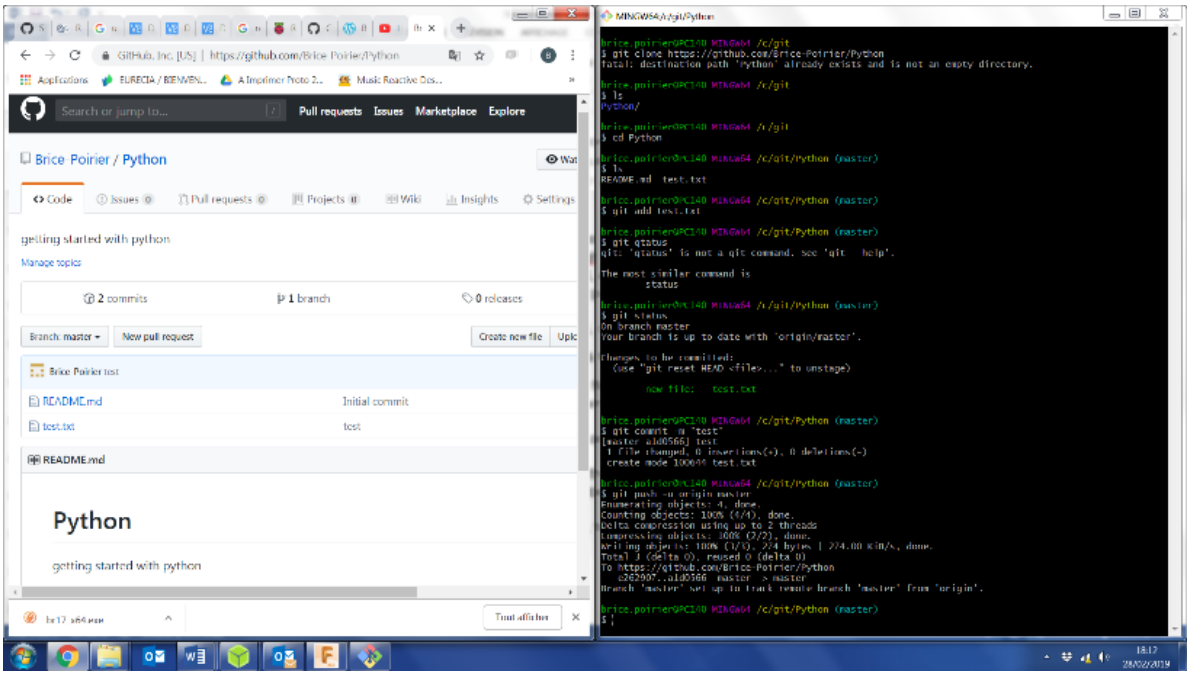
```
git status
```

c. Now we need to integrate those files to the package that will be sent online

```
git commit –m “describe here your commit : what the file is, what it does..”
```

d. And finally we send the package online, to the master branch for now

```
git push –u origin master
```



CAUTION : if your local repository is not up to date, you will not be allowded to push your commit. If so, make sure to git pull before git push :

git pull “https://github.com/PROTO204/Students-SETI-2019"

4. USING GIT ON A DAILY BASIS

Because there will be many of you working on one project, before adding anything, you’ll need to make sure you have the latest version of the project or else, Git will not accept any modification.
In order to download the latest version, you’ll need to type “Git pull”. That command will fetch the files on the cloud repo and merge them with your local version.

Try and do one commit at a time, if you want to modify something, you can use Git reset before you commit, that will cancel previous commits and allow you a fresh start.
$ git reset --soft HEAD^ (1)
$ edit (2)
$ git commit -a -c ORIG_HEAD (3)

(1) This is most often done when you remembered what you just committed is incomplete, or you misspelled your commit message, or both. Leaves working tree as it was before "reset".
(2) Make corrections to working tree files.
(3) "reset" copies the old head to .git/ORIG_HEAD; redo the commit by starting with its log message. If you do not need to edit the message further, you can give -C option instead.

Sometimes, you might get mixed up, commits are piling up and GIt tells you you are x commits ahead and refuse to push. There are several options, depending on what you want to do:

- git push : move your changes to the remote (this might get rejected if there are already other changes on the remote)
- do nothing and keep coding, sync another day (not recommanded)
- git pull : get the changes (if any) from the remote and merge them into your changes
- git pull –rebase : as above, but try to redo your commits on top of the remote changes

WANT TO LEARN MORE ?

Most common problems will have been answered on stack overflow or this useful website https://ohshitgit.com/