**Software Carpentry Saturday October 3**

**Github**

Slides on Github:

<http://slides.com/connorosborn/intro-to-git/live#/>

Etherpad:

<https://etherpad-mozilla.org/swc-iplant-2015-10-141>

Dropbox:

<https://www.dropbox.com/s/zr0vit95f7ed0mc/shell-history.txt>

General info about the workshop:

<http://naupaka.github.io/2015-10-03-ua-iplant/>

Software Carpentry website:

<http://software-carpentry.org/>

Atmosphere:

<https://atmo.iplantcollaborative.org/application#instances/f2aac81c-0b7d-460f-864b-2433e78e6047>

All what **git** (the tool) cares about is the succession of **commits**. **Github** is the platform, which gathers the gits. **Add** send what I want on the **stage** (which is not visible), then **commit**, then **push** on github to make it visible. The only purpose of the stage is the commits, necessary to keep track of all changes I’ve done and why I’ve done them.

create an empty folder: **mkdir**

create an empty file: **touch**

Let’s create a commit and push it on github.

Zip file = binary files! Github doesn’t like binary so DON’T COMMIT A ZIP FILE!

1. modify on github and then get it on your computer (download, copy from download to directory,

2. modify on your computer and get it on github (add, commit, push)

1. Download the zip file Sunday-scripts on github

2. copy it from Downloads on my computer to the directory I want to be in (my folder software-carpentry-october-2015)

$ **cp ~/Downloads/Sunday\_scripts.zip .** *#the ‘.’ at the end means copy it in the current directory*

$ ls

README.md Sunday\_scripts.zip

3. unzip the zip file

$ **unzip Sunday\_scripts.zip**

$ ls

README.md cleanData.sh setupProject.sh

Sunday\_scripts.zip combine.sh

$ **git status** *#means what the fuck is happening!*

“Untracked files”

Sunday\_scripts.zip

cleanData.sh

combine.sh

setupProject.sh

note: remember don’t add a binary file!

4. add files on the “stage” (see below)

$ **git add** cleanData.sh combine.sh setupProject.sh

note: git doesn’t add directories, only files.

5. commit the added files

$ **git commit**

note: you can do **git commit –m “**blabla**”** or **git commit**. It will open an editor, which gives more flexibility to give a title to the commit and then some additional information about what I did.

6. send it on github

$ **git push origin master**

Bilan, what we did:

1. we moved all our codes in the same directory
2. we cleaned our data by running the file data
3. we committed and pushed these changes back on github