**Git Commands**

**Git init:** initialised the working directory, this tells git to start tracking all changes in the directory. A hidden directory named .git will be created in the directory

**Git status:** show new files or files that have been amended. If no file is added, git will mention that there are untracked file.

**Git add file\_name :** allow to track changes on the file, the file is added to the “staging area”

If we want to add several file we can separate them with a space.

**Git commit file\_name --m ”message”:** save the changes, git will keep a log of this file. Message is to describe the changes done it is useful to track the different version of the file.

A file must be “add” before it can be “commit.

**Git commit file\_name:** same as above except that a prompt screen for comments will appear . Use I and Esc to swap between edit and view mode. “:wq” to leave the prompt.

It is more common to use the other commit command.

**Git log:** all the changes in the directory will be displayed, change here are changes “commit”, this is the reason a message is required for each file “commit”.

**Git diff file\_name:** display the change between last commit and current file.

**Git reset:** it can be used after a file has been added to the “staging area” (git add).

This will remove the file from the staging area. Use “.” for all files or list the files separated by a space.

**git diff -- staged:** compare the file between staged file and working directory one

**Git revert:** revert to a previous commited file. Often use with HEAD (Git revert HEAD) to revert to the previous committed file or HEAD~x to revert to the x+1 previous commit.

Git revert will create a copy to the commit we want to revert to and use this commit. Git revert do not alter historical commit.

**Working on Branch**

A branch is a virtual environment that will allow work to be done on the file without affecting the main file. The work done on the branch can later be merged with the main (master) branch.

**Git branch:** display all the branches of the directory

**Git branch branch\_name:** create a branch

**Git checkout branch\_name:** change working branch.

**Git diff branch1..branch2:** changes between two branches will be displayed. Same as diff feature branch has to be “add” and “commit” in order to have diff.

**git merge --no-ff branch\_name –m “message”:** merge the branch to the master file.

The command has to be executed after checkout master. We can’t merge from the virtual branch.

**Git branch –D branch\_name:** delete the branch, it is always better to delete branch after we finished working on it.

**Commands related to Github**

Repository = project/folder

After a repository has been created in Github, the platform will provide an URL/SSH Key. This will be the URL to use to upload the files into Github.

**Git remote add orgin SSHkey**: it will add the SSH key locally so we can later push file to this repository.

**Git push SSHkey branch\_name:** upload the branch into the repository

**Git pull SSHkey branch\_name:** download from the remote to the local environemnt

**Git Fetch SSHkey branch\_name:** will download the file in a different branch and not affect the current version.

**Git branch –r:** will display all the remote branches

Forking = Fork option in github allows to copy a project to our own repository. Then from there we can clone.

**Git clone “ssh key”:** clone in the computer the file.

**Navigation**

**Pwd:** shows the current directory

**Cd:** change directory to go to the selected directory. Full path can be written in one line using “/” between directories. If it used without path it will revert to the original directory

**Cd .. :** Move up by 1 directory

**Mkdir:** create a folder. Add the name of the folder after mkdir.

**Mv “current\_name new\_name”**: rename file or directory

**Rm “file name”:** remove the file

**Ls:** will show the list of file in the directory

**Ls –a:** will display all the file including the hidden one. Hidden usually starts with a “.”

**Cat “file name”:** open/access the file name

**Cp:** copy a file into a different folder, the new folder written after a space

Full path has to be written

Cp must be followed by the option needed (-a, -r, etc…)