Git & GitHub Study (freeCodeCamp)

**Git** – Free and open-source version control system.

**Version Control** – The management of changes to document, computer programs, large web sites, and other collections of information.

Terms:

**Directory** – Folder

**Terminal** or **Command Line** – Interface for Text Commands

**CLI** – Command Line Interface

**CD** – Change Directory

**Code Editor** – Word Processor for Writing Code

**Repository** – Project, or the folder/place where your project is kept

**GitHub** – A website to host your repositories online

Git Commands:

**Clone** – Bring a repository that is hosted somewhere like github into a folder on your local machine

**Add** – Track your files and changes in Git

**Commit** - save your files in Git

**Push** – upload git commits to a remote repo, like GitHub.

**Pull** – Download changes from remote repo to your local machine, the opposite of push

~ ssh-keygen -t rsa -b 4096 -C [marwindacallos@yahoo.com](mailto:marwindacallos@yahoo.com)

Generate SSH Keys and added to computer

Or getting your SSH Keys just watch this <https://www.youtube.com/watch?v=WgZIv5HI44o&feature=youtu.be&ab_channel=SyntaxByte>

**Passphrase** – password 09987450702Zz

-,

**ls -la** – to list everything in repository even hidden files

1. We go at view then terminal or ctrl + `

1 – We clone first doing git clone then the ssh link,

2. then targeting the file or moving by using cd demo-repo(file)

3. Then track the file first by using git status, then git add . or the git add (file)

4. Then commit and there should be a message using git commit -m ‘(message)’

5. then git push

OR

Git remote origin [git@github.com:marwindacallos/demo-repo2.git](mailto:git@github.com:marwindacallos/demo-repo2.git)

Git push -u origin master (use this to just comment git push u)

**GIT BRANCHING**

MASTER BRANCH – (main branch)

Git branch (so we can see our branch)

Git checkout (so we can switch our branch)

FEAUTURE BRANCH – (re-edit branch) adding we use (git checkout -b feature/description-11

Git branch -d (file) so you can delete

Now I am in the 2 branch

HOT FIX BRANCH –

Git merge (merged branch)

-u = --set-upstream

Git diff (see the difference)

Hit q after

Git commit -a (add and -m at the same time for new modified files)

MAKING MISTAKE

git reset README.md (undoing add)

git reset HEAD~1 (undoing commit)

git log (then copy the commit )

git reset (then the hash of the commit)

git reset --hard (hash) – this will do completely remove

FORK IS FOR NOT FOR PERSONAL bECAUSE YOU ALREADY HAVE ACCESS INTO IT

FORK IS FOR SOMEONE TO HAVE AN ACCESS TO CHANGE SOMETHING

DELETE THE BRANCH WHEN IT’S ALREADY MERGE AND THE PROBLEM IS ALREADY SOLVED

Also we make branch for fixing problems so we will not have a future bugs.