**Git and GitHub**

**Git**: Usually used for developers sharing work among themselves.

* Nothing to do with coding {Keeps tracks of old version or old documentation after updated}
* In this case, Dev / testers is very useful... let’s say the client want a change and decided he/she want the old version back.
* When a file is changed or updated, Git will ask to keep the old version save & store. After save, you can access the old version at any time.

**GitHub**: (Remote server/ computer) a web-based interface that uses Git, the open-source version control software that lets multiple people make separate changes to web pages at the same time.

* GitHub is a server that manages Git. (the brain)
* Allows dev/ tester to share work with each other
* Allow manager to create Abstract method and share with employees
* Git allows and encourages you to have multiple local branches that can be entirely independent of each other. The creation, merging, and deletion of those lines of development takes seconds.

See the diagram for more information...

**How to install:**

*Before installed: in CMD*

*{C:\Users\ddebe>git --version*

*'git' is not recognized as an internal or external command,*

*Operable program or batch file.}*

In google search, download git

Click first – “https://www.git-scm.com/downloads”

Click windows

Select – window portable & download – (*64-bit operating system, x64-based processor*)

Setup click Next,

Choosing default editor – Use notepad

Override {main}

Next,

Install.

After Install:

*{C:\Users\ddebe>git –version git version 2.43.0.windows.1}*

**Key Words to use in interviews:**

* I push my code to GitHub, or I push my code to remote Repository
* I have got Git install in my local machine

**Global Configuration (needs full name & email address):**

* Right click 🡪 Open Git Bash Here
* $ 🡪 git config –global user.name “Daniel Debesay”
* $ 🡪 git config –global user.email ddebesay@gmail.com
* $ 🡪 git config –global init.defaultBranch main
* Set Up is complete!
  + To verify:
    - **$ 🡪 git config –list** {you name & email should appear}

**Local Repository and Remote**

* **Saving into Staging of Local Repository.**

Inside folder,

* $ *pwd* {/c/Users/ddebe/Desktop/GitPractice}
* $ *git init* {create new file on folder – meaning git is monitoring
* $ git status
  + **On branch main:** meaning it is on main branch
  + **No commits yet:** meaning you have not made any changes inside file yet
  + **Untracked files:** meaning not send to staging or Local Repository
  + **$ git add:** this will add the file into Staging
  + **($ git add .):** used this if you are adding more than one file. After checking the status... both files should be green.
  + **$ git commit –m “message”** : this will add into local repository.
  + SAVE YOUR WORK EVERY 2 HRS!!
* **Creating and saving into Branch:**
  + **$ git branch** { will give you how many branches you have}
  + **$ git branch ‘nameOfBranch’:** will create a new branch
  + **$ git checkout ‘nameofnewBranch’:** will switch into new branch
    - Now you can create new files and work on this new branch, however it will be ahead of the Main Branch. Main branch will not be able to read this. You will need to Merge and delete the new branch.
* **$ git merge ‘nameofnewBranch’:** {note you need to be on main Branch

in order to work}

* **$ git branch -d ‘nameofnewBranch’ :** deleting the new branch.
* **Send your Local Repository to Remote Repository:** 
  + *GitHub.com - > sign Up account*
  + *Origin : remote is local until you add Origin*
  + *In GitHub Home Page, click right corner logo -> Settings -> Developer settings -> Personal access tokens -> Tokens (classic) -> then create access Token. Should receive access token & Address of repository*
    - *$ git remote add origin “address of repository”*
    - *$ git branch -M main*
    - *$ git Push -u origin main*
    - *After Once, all you must do is $ git push*
* **Cloning Lead / coworkers Remote Repository into Local:**
  + *Copy the Address of repository*
  + *In the file you want to Paste, open git Bash*
  + *$ git clone “Lead Address of repository”*
  + *Complete !!*