

GIT Lessons

A system for keeping track of changes that happen across a set of files. It is a system that works in files by monitoring the folder the file is in.

How do we use it?

- Install the system on your computer if your computer does not already have it. The system is call Git.
- Open any folder you want to keep track of in a terminal (just any terminal - in the command prompt window or a code editor like VSCode).
- To monitor the folder and keep track of any changes in it and its content, tell Git to start monitoring that folder by using the git command "git init". (Git system must be installed in the computer). Git will create a hidden folder in the folder you are tracking called (.git). Just ignore it, no worries about this.
- Make any changes in the folder (editing its files) you may like and Git will start monitoring the folder for any changes. Use the command "git add ." to add all changes to the Git system.
- You can take snapshots (commits) of this folders and Git will keep track of them. After adding the changes to git, use the command (git commit -m "a message describing the changes") to tell Git take a snapshot of the changes.
- Git will keep records of this snapshots(commits), so that you can always refer to them when you need them. Git keeps this chain of changes in a chain called branch. The original chain(branch) when it starts keeping this record has a name mostly "main" or "master".
- Because in software, so many people can be working on different files in the folder you're tracking. So person A (Glory) working in that folder can create a new chain for herself from the main chain, this is called branching.
- Use the command "git branch 'new-branch_name'". Now a new chain called 'new-branch' name' is created and Glory can start editing files in that folder and making snapshots in her own chain. Ify may also create her own branch and keeps taking her snapshots as she makes changes. To move into that branch, use the command "git checkout 'branch-name'"
- This snapshots(commits) technically create individual folders for their branch.
- Now when all is done and Ify and Glory can converge their branches back to the main branch with all the changes they have made. Use the command "git checkout 'branch-name'" to move between the branches.
- The main branch manager verifies their branches to make sure there's no

conflict and merges it to the main. To merge the alternate branches to the master branch, use the command “git merge ‘branch-name’”.

- If there’s any conflict in the files, the main branch manager, rectifies and validates.