

Git

Git stands for
Global Information Tracker

* uses →

-) helps knowing the history of the code
-) which person made which changes at what time
-) sharing code with other people

Note: All the history, including creation of a file, deleting it etc. is all stored in .git folder (which is hidden)

* commands →

1. git status informs us about untracked files
2. git add. adds a change in the working directory to the staging area
3. git commit -m "<message>" commits a snapshot of all changes in the working directory
4. git push upload local repo content to a remote repository
 add -f to force push
5. git restore --staged <filename> will restore the file to the last commit version

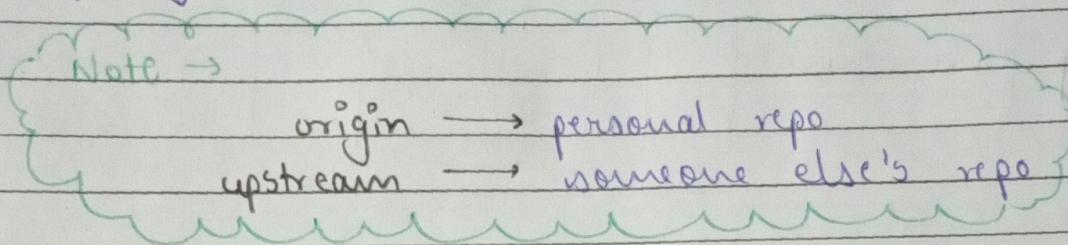
6. `git log` allows us to look at the entire history of the project
7. `git reset "commit id"` deletes any commit that is above the specified commit id
8. `git stash` takes uncommitted changes (both staged and unstaged) and saves them for later
9. `git stash pop.` will take content from stash file and apply them to the current working file
10. `git remote add origin <repo url>` used to attach a url to a local repo
11. `git push` Q - why use branches?
allows us to work on different versions of our code simultaneously.
11. `git branch feature`, used to create a new branch
branch name (head changes to feature)
12. `git checkout main` used to switch b/w branches
(head changes to main)
13. `git merge feature` now we've merged feature with the master/main branch
14. `git clone <url>` to clone any repo to local folder

Q- Why to fork?

Forking enables us to modify someone else's code by forking it and then later cloning it to make changes.

- Upstream url → The repository from where the code repo is forked is called as upstream url.

eg → git remote add upstream "url"



- Pull requests →

Pull requests lets us tell others about changes you've pushed to a branch in a repo. Once a pull request is opened, upon further discussion if everything goes well, changes are merged in the base branch.

- Merge conflicts →

It is an event that occurs when git is unable to automatically resolve differences in code between two commits.