GIT Commands

* Github platform allows us to host git repos
* To initialize empty git repository in our folder we use : **git init**
* To see hidden files in our folder : **ls -a**
* To create a file in linux: **touch filename.txt**
* To see the changes done and review, it shows what branch are you in and what files you made changes everything : **git status**
* **Untracked file** : means you changes your file locally but people are not aware of it
* To send the files to staging area( history for adding it to hub) : **git add .** //all filles will be sent to staging area

or **git add** **filename** //only selected files

* To commit/send the changes to repo from staging area : **git commit -m “file is added”** (-m represents a message or comment)
* Again if you edit your file in your local and press **git status ,** you will see again the files changes under
* When you bymistakenly sent files to your staging area by git add . , you remove it by **git restore –staged filename.txt**
* To see history of the project (all the commits were made ):- **git log**
* To delete the file in local  **rm -rf filename.txt**
* And you need to tell and remove that in git also so check **git status** , you will see deleted files, they do **git add .** or **git add file name**, then do **git commit -m “deleting the file”**

This means you are deleting these files in history also

* In git log you will see all the commit id if you want to undo the deleted file commit then you can do **git reset commitId** it will move everything to staging area
* When you wanted to save all the file in a staging area and wait , if need to commit i will do that later, save the files without committing or putting it in project history , the you can use **git stash**

**Hosting our own project :**

* To add your git repo url to your project : **git remote add origin url-link** (origin is the git given name to your url)
* To send your commits to git (url) is : **git push origin branch**(main or something)

Branching:

* Main node or say main branch name will be : **main**
* **Never commit on main branch/ always create a separate branch when working on new feature**
* To create a branch: **git branch branch\_name**
* To Go to that branch from our branch : **git checkout branch\_name** (after moving to your branch , now the commits will be in your branch instead of main)
* To merge your branch(after finalizing the code) to main branch , **git merge branch\_name**  //mergeing is done using pull request