To initialize a repository, go to the folder that you want to use as the respository and run

*git init*

To check the status of a respository, run

*git status*

To add a file to the repository, first we copy the file to the repository and add to it by running the command

*git add <filenames>*

or if you want to add all the files that got listed git status output

*git add .*

Then we commit the changes to the repository by running

*git commit -m “<commit comment>”*

To get the history in commits, run

*git log*

To go back to a previous commit, run

*git checkout <commit hash shown for the commit you want to go to>*

To get the list of all branches, run

*git branch*

Creating a new branch, run

*git branch <branch-name>*

Then to use the branch, run

*git checkout <branch-name>*

Then any subsequent operations you do like git add etc. will be on this branch. Basically same as would be if you were to checkout a specific commit. The changes/commits that you make to the branch does not get reflected on the *master* branch, till you merge them.

To merge a branch into the master, switch to the master branch and then run

*git merge <branch-name>*

To delete a branch, run

*git branch -d <branch-name>*

Useful reference material : <https://www.notion.so/Introduction-to-Git-ac396a0697704709a12b6a0e545db049>

Youtube Video : <https://www.youtube.com/watch?v=USjZcfj8yxE>

**GitHub**

Useful reference material : <https://www.notion.so/Introduction-to-GitHub-202af6f64bbd4299b15f238dcd09d2a7>

Create a repository in GitHub using the UI. Then link that repository to the one in the local machine. To do so, go to your local repository and run the command

*git remote add origin <URL of the repo in GitHub>* 🡪 Here “origin” is acting like a reference to the URL. It can be anything

You can see the remote repo links that you have by running

*git remote -v*

To push the files from the local repository into the repo in GitHub, run the command

*git push -u origin master* 🡪 here “origin” is the reference to the URL that was made earlier. “master” is the branch where we want to push to go into

To get the latest commits from GitHub on a branch, run the command

*git pull origin master* 🡪 “origin” is the URL and “master” is the branch

You can also create new branches in the local repository, create content and push to GitHub. Do the following:

*git branch <branch-name>*

*git checkout <branch-name>*

After make the changes in the files,

*git add <filenames>*

*git commit -m “<comment for the commit>”*

*git push -u origin <branch-name>* 🡪 This will create a new branch in GitHub if not existing and push the files into that.

You can also Clone an existing repository in GitHub into your local repository. As a good practice do NOT clone a repository into another already existing local repository. So make a new directory locally into which you want to clone an existing repository from GitHub. Change to the local directory and run

*git clone <GitHub repository URL>*

You will not be allowed to push back changes to the GitHub repository unless explicit access is given. If you want to contribute to an existing open source project on GitHub, you will first create a *fork* (almost like a branch), push changes into that and then raise a *pull* request to the main branch. To do so, go to the GitHub repo, click on “Fork” button and choose to which GitHub account you want it forked to. Then we clone the fork into the local repo.

Make the changes you need, do the add, commit and then push to GitHub forked repo.

Then in GitHub against that repo, click on “Pull request” against the master branch in your forked repo. This will create a pull request and also change the repo to the original repo, you can see the changes. In that click on “Create pull request” button, fill in the form with the comments and finally click on “Create pull request”. The repo owners have to access the changes that you have submitted.