

You first create a project and do git init inside it.

Then, create a new repository on the GitHub.

In the local repository, add the command:

Git remote add origin https://.... {Here, origin is the default name of the remote repository }

Now, if we do git remote, it’ll show origin, which simply means that we are on the origin.

If you do ‘git remote –v’, you’ll get the following two lines as output:

origin https://github.com/Hritik-Arora/GitLearning.git (fetch)

origin https://github.com/Hritik-Arora/GitLearning.git (push)

This kind of means that a connection has been established between our local and remote git repository.

Now, to push your code in the repository, do:

Git push -u origin master {-u means upstream, i.e. we want to push our code upstream to the repository origin, and in the master branch}

If we do git branch –r, the output is origin/master. Without going much deep, it means that we have a connection between master branch of our local and master branch of origin repository.

When you clone a git repository inside a folder, you’ll simply get a folder inside your folder with the name of the repository. Now that is a git repository. Go inside it and do git status and git log.

Now, if after cloning, I do a change to the local repository, and then commit it, it is commited in the local only. Then, if I do git status, it’ll show that I am ahead of origin/master by one commit.

Do git push to push to remote repository.

Some important terms before starting next topic:

**Remote**-tracking **branches** are references to the state of **remote branches. Remote tracking branches is thus a term associated with our local repository. Remote tracking branches are a kind of connection between local and remote repositories.**

**FETCHING**

**Git fetch**

This fetches the branches. The remote tracking branches are updated. After doing git fetch, you’ll get something like: master -> origin/master. This simply means that our master will point to remote tracking branch origin/master. The updates will not yet be seen on the repository. Now, do git checkout origin/master, and you’ll see the changes. This means that origin/master has the updated code, but master does not. So, the task is simple, move to master branch and merge master with the origin/master. Git merge origin/master, and your local master branch will be updated with the code.

**GIT PULL**

Now, th**e** two steps above (Git fetch and git merge origin/master) can be combined by just writing git pull.

**DELETING REPOSITORY**

To delete GitHub repository, go to github page, click on settings and the on delete repository.

To delete the from the local (the connection with remote repository, simply type:

Git remote rm origin 🡪This deleted the connection between local and remote repository (named origin). After doing this, if we type git remote, it wouldn’t show us origin.