# GIT:

It is used for version control system to maintain the different versions of a code

Common commands:

$ sign means waiting for your command

*- git config --global user.name "Your name here"*

*- git config --global user.email "Your\_email@example.com"*

*- git config --list # for the details of your git hub account*

You can create github repository in your git hub account and then create the same named repository in your local system and link both of the repository

1. Create a repository in your account

2.Use *“mkdir”* "same named directory that you have created in the git-hub account"

3. Change the directory to the newly created directory by using the command cd

4. Use *git init* to intialise the local git repository

5. Point your local repository to the github repository that you just created in git-hub using the foillowing command

*git remote add origin https://github.com/yourUsername/yourRepo.git*

Now you have linked up your local repository with you remote repository

6. Downloading or forking an existing repo, used to collaborate by downloading a copy of existing repo and making the changes and then pushing to the existing repo

*git clone https://github.com/user/repoYouWantToFork.git*

This will create a local copy of the repo you wanted to fork

Some other commands:

* Suppose that you have added new files to your repo, then you need to tell git that they need to be tracked for version control. We need to do this before committing.
  + git add . : it adds all the new files in the working directory
  + git add –u : updates tracking of files that were changed names or deleted
  + git –A : does both
* Suppose you have changes that you want to commit and save as an intermediate version
  + Git commit –m “message” where message is meaningful description of the changes, however it updates only the local repository
* You have saved all the commits and now want to push it to the remote directory. You need the following command:
  + git push
* When you are working on a live project and that particular project is being used by many people then you try to isolate your work in different branches. So the commands for starting such branches are as follows:
  + git checkout –b “branch name”: to create a branch
  + git branch : to see on which branch you are working on
  + git checkout master : to switch back to your master branch type

