Git is a version control which allows us to track changes, undo changes, compare versions. Git is a free and open source tool.

Github or Gitlab is used to host our source code on internet then we use hosting service called Github. Github is owned by Microsoft,

Github is a servicefor hosting files on the internet that are managed using git.

cd🡪Change directory

make a new directory(mkdir)—creates a new directory

list files in current directory(dir)

Git itself is hosted on github

We can install git using below link

<https://git-scm.com/download/>

Below are the commands we can execute to get the username and email address of github account

Git config user.name

Git config user.email

If there is no configuration done we have to execute below commands

Git config –global user.name “venkatvarma”

**git init** command will create a Repository

**git status** will check the repo status

**git clone “url”** to clone the report

if it s public repo token is not required we can execute the below command to clone the repo

**git clone https:// github.com/kvenkatvarma/TestRepo.git**

if it is a private repo then we need a token to clone the repo. below is the command

**git clone https://ghp\_2hP5X9Kmh7NlmXtaIVTINXIR5EVEsD2McUrc@github.com/kvenkatvarma/TestRepo.gi**

**t**

After creating a file in vs code we have to execute below commands

**git add :filename”**

if we want to add two files then we have to execute **git add .**

**git commit -m “Aded a file(here we can add any comment”**

Below is the command to connect to remote branch

**git remote add origin** [**https://github.com/kvenkatvarma/GIT\_SAMPLE.git**](https://github.com/kvenkatvarma/GIT_SAMPLE.git)

Below is the command to push the changes to remote branch

Git push -u origin master

**Git fetch** will download the changes form the github remote repository but we will not see those changes in the working directory

**Git pull** will directly fetch the changes form the remote repository and updates the files in the working directory

**Head** It will always points to most recent commits in the master branch

**Git branch <branch-name>** is used to create a new branch

**Git switch branch name** to change the from one branch to another branch

**git branch -m <branch name>** to rename the branch. Here m=Move

**git branch -D <branch name>** to delete the branch. If we are deleting a specific branch then we should not be in that branch

Suppose I created a new branch and made changes in a file. Now I want to merge the changes to master branch. So first we have to switch to master branch and then we have to execute **git merge new branch name**

**git diff** is used to compare the difference between new commit and previous commit

**git diff firstbranch secondbranch** to check the diff in file between two branches

we can also use **git checjout branchname** instead of **git switch branchname**

**git log --oneline** will give us the first 7 characters the commit unique identifier

**git restore** suppose I made changes in one file and I did not committed it. Now I want to restrore the file to previous commit. So for that we can make use of **git restore filename** command

**git restore --source HEAD~1 file.txt** this command is used if we want to restore from a specific commit.

For example I committed 3 times. In first commit “Added First” , in second commit “Added Second” and in third commit “Addd third”. Now if I want to till 2 commits we have to specify **below command**

**git restore --source HEAD~1 file.txt** Here 1 means we are skipping 1 commit

**git reset** allows us to remove the commits and reset the branch

git reset –hard will remove the commits

**git revert** command will create a new commit that undoes the work from the previous commit but keeps those commit in the branch