GIT TUTORIAL REFFERENCES:

|  |  |  |
| --- | --- | --- |
| Working Directory | Staging Area | Local Repository |
| Creating new textfile(eg..FirstCode.txt)(or)  Codefile. | Adding all the files  (or) Adding Selected Files. | Save & Commit  Repository. |

Working Directory 🡪 (git add<file name>) 🡪 Staging area.

Staging area 🡪 (git commit -m “ --msg--") 🡪 LocalRepository.

**Git Commands:**

1. git init -> Initializing the empty git repository in currentdirectory.
2. git status 🡪 Shows all statements
3. git init -b main (or) git branch -M master 🡪 to change the branch if we needed.
4. git clone (add “https/github.com/ your code name “) =>to add the local respository. In add code in github code name in local respository.
5. git config --global user.name “----u.name----”
6. git config --global user.mail “----u.mail----”

* When we config and the initialization are done we want create a repository in the Github login and it will gives a https:request like git remote add origin [www.github.in](http://www.github.in) we should run the command in code editor ……
* Every time we made the changes in the Working directory or coding file it should be added as git add<file name> and also use the commit using the git commit -m “--” after that we want to push the code using the git push -u URL main.

1. git add <file name> 🡪 Working directory to staging area ..
2. git commit -m “Commit message”🡪(to commit the current staged files to git repository with the commit message)
3. git log 🡪 shows the current branch and all the Commits log of commits in the local repository.
4. git remote add origin (URL of Github Repository.)

10.git remote 🡪listing of all the URL

11.git push -u URL main(send the data from local remote repository)

12.git diff 🡪In Working Directory

13. git diff –staged 🡪 (InStaged area)

14.git rm –cached creds.txt 🡪(proper method for removing or deleting the File in the Working directory.)

15.git log enter q

**Descriptions about Git:**

->When we try to add the new commit .We have to choose the filename as using (git add <filename>)and next we have to write the commit using the (git commit -m “Commit message”).

->git log (will show the commits)

->git diff (helps to test the changes done only in the Working directory (or) editing file..)After this ,If we check the status using (git status)nothing commit is there ..It will be already sended to the **staged area** …And then we have to work with (git diff --staged)🡪then it will show the all changes done in it.

Tags :

🡪git tag tagname -m “—msg—or version”,

🡪git push origin (tagname)->whatever we have created in the tag version we just need to push the message with this push method…

**Descriptions about Tags:**

->Tag is used for updating the coding files .. If we have to update the codes we have to create the tag for keeping the update..

->If we created a tag and updated for whole file it will updated as single version..

🡪 for creating tag using comman like (git tag tagname) and if we have to push means git push origin (tagname)..

Branches:

🡪git checkout -b branchname🡪(If we want to add branch)

🡪git switch branch -name 🡪(If we want to change the branches)

🡪git switch -c branch-name 🡪(If we use switch to create branch using switch )

🡪git branch --all🡪(If we want to see all branches)

🡪git switch 🡪(If we want to go previous branch)

🡪git branch -d branchname 🡪(If we want to delete branch)

🡪git merge branchname 🡪(If we want to merge branches)

Description about branch:

* They are two branches default (main & master)..We have to work always work with the main..Because the push method is only for main(git push -u origin main)….If we it is in master we have to change to main(git branch -M main),(git switch main)..
* If we are adding some new files we have to create a branch like Verison 1 or anything using the command (git switch -c b.name)& (git checkout -b branchname).
* If we have to check the current branch(git branch) .
* If we have to switch the branch (main to branchname you given),

(git switch branchname). In that normal all files default saves in the main.Because whenever you are creating a new files you have to create a new branch for updating purpose.

* After adding files in the newbranch we have push the branch using git push origin (branchname)..
* When we switch the branches it should be show the files presented in that similar branches.