Git Version control system

Version control

the management of changes to documents,computer programs,large websites, and other collections of information

Terms

directory -- folder

cli --- command line interface

ls- la to list all the items including hidden files (la) for mac

COde editor -- word processor for writing code

repository -- proect, or folder/place where your project is

Github -- a website to host your repositories online

Git commands

clone -- bring a repository that is hosted somewhere like github into a folder on your local machine

add -- track your files and changes in git

commit -- save your files in git

push --upload git commits to a remote repo, like github

pull -- download changes from remote repo to your local machine, the opposite of push

.md file

Git ls-tree –r branchname -- to list all the files

Git rm filename -- to remove fiel

Git commit –m ‘message’ --

git status -- which shows the changes eventhough not tracked

git add \*/file -- then we can track

git commit -m -- to commit the changes

SSH keys :

to push changes from local machine to git hub you have to prove that you are owner of this account for that you need to generate the ssh key by using this command

ssh-keygen -t rsa -b 4096 -C "email@example.com" -- here t stands for type of encryption and emailid

ssh-keygen is also enough

enter 'filename' to store the key

sshkey -- private

sshkey -- public (we have to copy this we will enter in github)

after adding to github

then we need ssh key to ssh agent

1.eval $(ssh-agent -s)

2.ssh-add ~/(inpath)/filename

PROCESS

git init -- create repository locally

git add /\* -- to track files

git commit -m 'message' -- to commit changes

git commit –am ‘message ‘ -- only works with modified file not for the new files

git push origin master -- to push

git remote add origin url -- create a repository in github and paste here

git remote -v -- to check

git push -u origin master -- -u is used here to avoid typing whole in the next time

Git branching:

Master Branch -- default or main branch

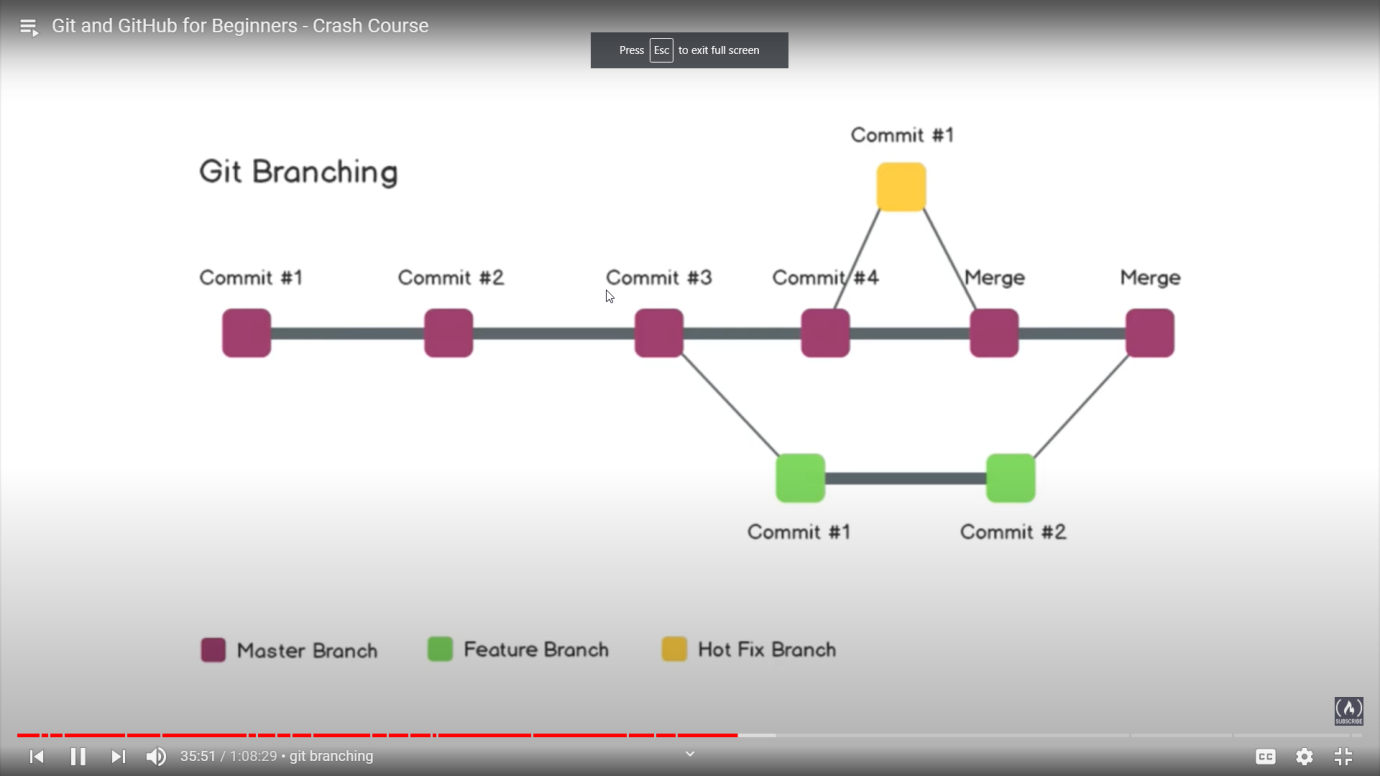
Feature Branch –

Hot fix Branch --

You cant see the edits or commits which is done in feature branch in master branch and vice versa.

After doing all in github

Git pull – to pull the code from github



Git branching commands:

Git branch -- to check how many branches are there and in which branch we are (\*) by this symbol

Git checkout – to switch between branches

Git checkout –b new\_branch\_name -- -b refers to create branch with branch name

If you want to add any new features switch to feture branch then do the changes and commit those changes then you can roll back to the master branch and you wont see those changes

And before merging feature branch to master you can use ‘diff’

Git diff featurebranch --

Git merge featurebranch –

Other way :

Git checkout featurebranch

Git status

Git push -- error will come because we had master branch but we don’t have feature branch

Git push –u origin featurebranch --

Now when you come to homepage of your repository you can see the notification of feature branch by clicking that you can create new feature branch and then you can see the commits in feature branch which are not available in feature branch and then you can merge it.

Now the master branch is merged with feature branch in github

Then we need to merge in local machine

Git pull --

Now we deleting the feature branch

Git branch –d featurebranch -- to delete the branch

Merge – conflicts:

Problems:

In master branch I wrote one line and in feature branch I added another line so when I want to checkout master I can do easily but for example I added the second line in master branch as well then I need to commit those changes to switch between branches

So before checkout you need to commit changes

Git merge master -- Then you need to solve conflict in any editor

And then Git commit –am

Undoing in Git :

Git reset -- after adding you can do reset but not after committing

Git reset HEAD~1 -- it will unstage and uncommit

Git log -- to see the commits and stages

Git reset commitid

Git reset –hard commitid -- it will not just unstage or uncommit but it also removes the commits and you will get that data.

Forking :

Copying the entire repository of other

Default it will be in the dev branch