Git and Github

(FreeCodeCamp - git and github crash course)

#creating a repository

Dashboard → + (at top right of screen) → new repository

creating a new / importing existing one if needed

Repository name → (project name)

Description (optional) → (describe a line about project)

public / private → choose public for all
choose private for us

☐ → for a initializing readme file
else skip

create Repository

create a new file and commit later perform changes and update commit

code → create a file (Readme.md)

↓

#Demo
some description
:::

↓

commit new file (create) ← open the file

(edit the file) → chane something in file

↓

Commit changes (updates)

history of commits ← 2 commits

↳ see two commits one is new so green another is red later green since we changed order → one with newer (+) if white ⇒ stayed same as before.

install Git (which contains GitBash)

create a folder git named and run gitbash in it.

Terminal of gitbash.

pulling down

→ git git clone (copy clone or download location/url in repository) (folder)

→ git cd demo-repo (changes in Terminals)

→ demo-repo git:(master) ls/ls

o/p:- all folders, files, details

↳ add some changes to file and push it

→ git status (command)

o/p:- modified : Readme.md. } Red color

↳ write an file like index.html in same demo-repo folder

→ git status

o/p:- modified : Readme.md
untracked :
index.html } Red color

↳ we should track the untracked files

→ git add . (for all) (called period)

→ git clear (clears terminal)

→ git status

o/p:- modified : Readme.md
new file: index.html

↳ now we need to commit

→ git commit -m (message) "added index.html babai" -m "some description"

o/p:- info of changes

↳ Here we still saved changes locally

→ git push (push to remote repository where project is hosted)

SSH Keys

To push into github account we should prove that we are owner, to connect local machine to github online we use SSH keys

→ ssh-keygen -t rsa -b 4096 -C "V.srinivassonny7@gmail.com"
Enter file in which we want to save key : mykey

... } leave empty

↳ now key is generated to search for key,

→ ls | grep mykey

op:- mykey
mykey.pub → public for all
if .pri ⇒ private for us

→ cat mykey.pub

op:- (full key)

↳ copy whole key (or) use this,

→ pbcopy < ~/mykey.pub

↳ goto account → settings → SSH and GPG keys

New SSH key

mykey

(paste the key copied) → Add SSH key

confirm account password

To add SSH key to ssh-agent ⇒ (help.github.com ⇒ generating a new SSH key and adding to ssh-agent)

~~git push~~

→ git push origin master

↳ 3 commits (2 online, 1 pushed offline)

↳ create demo-repo2

→ cd demo-repo2

→ ls (nothing since not a repository)

↳ write and create a readme.md file

→ git init } (creating a local repository)

→ git status

o/p:- untracked :
readme.md } Red

→ git add readme.md

→ git status

newfile : Readme.md } green

→ git commit -m "created Readme amma" -m "description"

o/p:- changes / additions / updations info with msg.

↳ (to push live)

→ git push origin master

o/p:- error:- since local repository no connection to github

↳ create empty repository with same name in github

↳ create repository.

↳ (copy git@github.com:.....git)

→ git remote add origin (paste that link we copied)

→ git remote -v → (created default as origin master)

→ git push -u origin master ↳ (next time use only git push)

git branching

→ demo-repo git:(master) git branch

op:- * master } 1 Branch (already open)
(END)

↳ press q

→ git checkout -b (creating) feature1

op:- switched to feature1

→ git branch

op:- * feature1
master
(END)

→ git checkout master

op:- switched to master

→ git branch (use Tab)
→ git checkout feature1 (auto completes name)

↳ change something in readme.md (M near filename => Modified) (in vscode)

→ git status
Readme.md } Red

→ git add readme.md
→ git commit -m "updated readme"

→ git checkout master (changes don't appear in master Readme.md)

→ git diff feature1
↳ press q
→ git checkout feature1

→ git status

→ git push

↳ but use,

→ git push -u origin feature1

O/P:- pushed to github

...

give some instructions also.

↳ pull request ⇒ feature1 to master once pr → pull request is merged

we can delete feature1 branch

↳ goto github ⇒ see compare & pull request

(base: ... ← compare: ...)

write about changes → create pull request → see commends, commits, files changes.

Resolve (not necessary) (+) in any line ↳ to comment on that line

↳ merge pull request → confirm → merged successfully

changes made to master branch

→ git checkout master → but no changes in files locally since we only made it online)

↳ To get changes locally

→ git pull → (used upstring before)

O/P:- changes done

↳ local changes done in master

↳ delete feature1

→ git branch -d feature1

→ git branch

↳ To mange merges conflicts

→ git checkout -b quick-test

↳ add something in index.html

→ git status

modified : index.html } Red

→ git diff

↳ press q

→ ~~git add~~ git commit -am "done" → adds and commits with msg at same time

↳ only for modified files not newer files

→ git checkout master

↳ no change in master

↳ change with 2nd line something

↳ now if we merge both 2nd lines conflicts.

→ git branch

→ git checkout quick-test

op:- error :- overwritten

please commit

→ git status

modified : index.html.

→ git commit -am "added new 2nd line"

→ git checkout quick-test

→ git diff master

→ git merge master

Error- conflict

↳ In vscode it shows options on how to change conflicts

↳ In others delete not needed lines and keep only needed lines.

→ git status

both modified : index.html

→ git diff

→ git commit -am "merged/updated with master"

Undoing in git

↳ add another line in Readme.md.

→ git status

{red

→ git add Readme.md

→ git status

{green

↳ but since we found later that line is not needed/ wrong one

→ git reset

→ git status

{Red

→ git add Readme.md

→ git commit -m "added wrong line"

→ git status

→ git reset HEAD~1 → 1 commit further one

→ git status

→ git diff

↳ press q

→ git log

op:- all commits with latest ones on top

↳ To goto some commit copy Hash id (long string of gerraco colors)

↳ use space to see still old commits

→ git reset (copied hash of commit) → still wrong lines exists

→ git reset --hard (another some commit hash)

op:- removes all wrong lines completely

Forking

↳ To make changes in some other repos we need to fork a pull request

↳ click on fork in other's repository → select our account

changes only to my forked not other's one

now we can commit and update

for that we create a pull request

↳ pull request → create pull request for other's / for our master → merge

⇒ our master is fixed with others repository + our files also