

GitHub classes



Github

- Folders that we upload on github is called repository.

Repository

- Readme file - stores relative info
- commit - change (github tracks your changes)
- To add readme file learn basic html.

Git ⇒ It is installed on the computer

Configuring Git

- git config --global user.name "My name".
you can also use local if you want to use a different account to make changes in the repo
- " " --global user.email "Someone@email.com"
- git --list

Clone & status

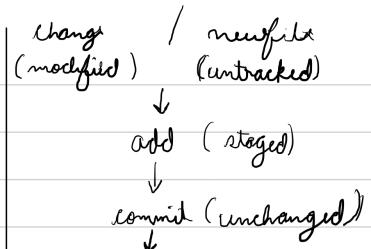
Clone :- Cloning a repository on our local machine
git clone <- some link ->

[remote] [local]
github on laptop

Status :- display status of the codes

Different status shown are

- ① Untracked \Rightarrow new file that git doesn't yet track
- ② Modified \Rightarrow changed
- ③ Staged \Rightarrow file is ready to be committed
- ④ Unmodified \Rightarrow unchanged.



push

Add & commit

add \Rightarrow adds new or changed files in your working directory to the git staging area.

git add <file name> / git add \rightarrow This is to add all the files, if there is a file.

commit \Rightarrow It is a record of changes

git commit -m "some message"

even after all this the files won't show on github. we use git push

Push command

push :- upload local repos content to remote repos.

git push origin main

init command

init :- used to create new git repos

different commands

git init → to create new repo → link of the new repo created on github

git remote add origin <link> This is the name of new remote repo

git remote -v → to verify remote whatever name you use here will be used

git branch → to check branch

git branch -M name of new branch (main) → to rename branch

git push origin main

Note if you use → git push -u origin main then from next time you only need to type (git push) and it will push in origin main only

Git Branches

- Branches are made so that different people can work on the same project at the same time and no one has to wait for the other person to complete since they are working on their own branch.

Branch commands

• git branch → to check branch

• git branch -M main → To rename branch (Here name is changed to main)

• git checkout <branch name> → to navigate to different branch.

• git checkout -b <new branch name> → to create new branch.

• git branch -d <branch name> → to delete branch

→ You won't be able to delete the branch you are currently working on

Merging code

Way 1

- `git diff < branch name >` → to compare commits, branches, files & more
- `git merge < branch name >` → to merge 2 branch

Way 2

- Create a PR → PR is a pull request

Pull request

It lets you tell others about changes you've pushed to a branch in a repo on github

process of pull request is from 57:00 - 1:00 in the video

Pull command

- `git pull origin main`

This is used to fetch and download content from remote repo and immediately update the local repo to match that content.

Merge conflicts

An event that takes place when git is unable to automatically resolve difference in code between 2 commits

Undoing Changes

Case 1 : staged changes (add →)

- `git reset <- file name ->`
- `git reset` → for multiple files

Case 2 : committed changes (for one commit)

- `git reset HEAD~1`

Case 3 : committed changes (for many commits)

- `git reset <- commit hash ->`
- `git reset --hard <- commit hash ->`

These bases are different for every commits and can be checked by using the command
• `git log`

Fork

A fork is a new repository that shares code and visibility settings with the original “upstream” repository.

Fork is a rough copy.