

28

* What is git?

Free & Open Source Version Control System



Tools that help to track change in code.

- 1] Track History
- 2] Help to collaborate
- 3] Scalable
- 4] Fast

* What is github?

Website where we host repositories online.

* How to Create Repo :-

1] Type Repository Name

2] Description

3] type of repo : Public
Private

∴ Help to other people & see repo to recruiter.

4] Add README.md file md = Mark down

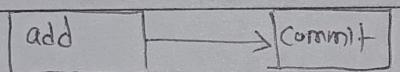
∴ Give a brief about the project (e.g. why we create project, benefits, description, demo)

5] Click on **Create repository** button.

* Commit :-

Commit mean change in actual repo / Disk

there are two step :



* Using git :-

1] Command Line (Most popular)

2] IDE / Code Editor (like vs code)

3] GUI (like gitkraken)

1] Command Line :-

1] Check git is install or not

\$ git --version

∴ if not then goto git scm website & download it.

2] If you want to see all git command then use

\$ git

Configure git :-

git config --global user.name "My name"

e.g. git config --global user.email "someone@gmail.com"

The command you provided is used to set your username in git globally.

By replacing "myname" with your actual name.

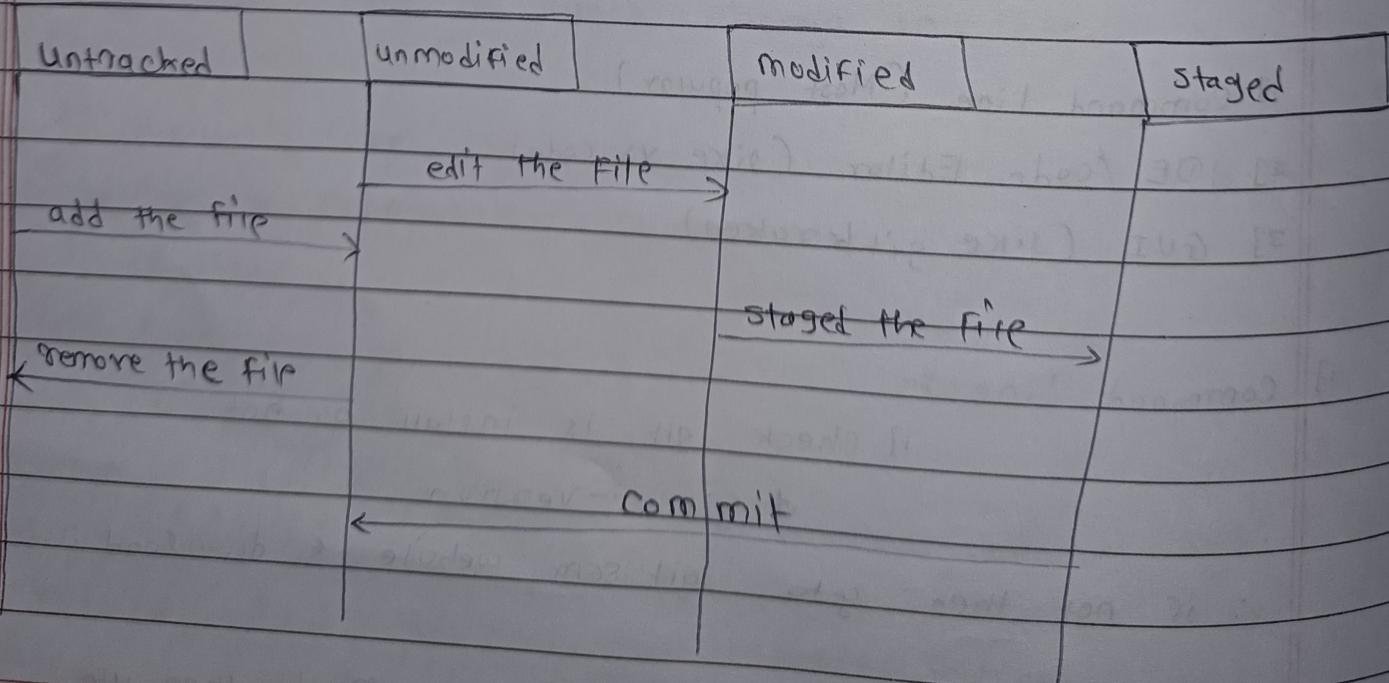
* Basic Command :

- clone - cloning a repository on our local machine
- status - display the state of code

git clone <- Some link ->

git status

File status life cycle



Github account contain three type of link

- ✓ 1] HTTPS (Use this) :: for beginner
- 2] SSH
- 3] GitHub CLI

e.g. git clone <https://github.com/Sample/demo.git>

Benefit of Clone command

- 1) Create copy to local system
- 2) Change (modify) allowed

if you change in README file then color change to yellow

4 "M" symbol add ~~mean~~ ^{new} M → ~~unmodified~~ if you add another file
 "M" mean modified
 "U" mean ~~unmodified~~ ^{tracked}
~~unmodified~~ untracked

Before change/add some content/file in repo you goto that repository.

cd repo-name

* Commit : add → commit

1) add - add new or changed files in your working directory to the git staging area.

git add <file-name> (for single file)

git add . (for all file)

2) Commit - it is the record of change

git commit -m "some message" (for all file)

3) Push : upload local repo content to remote repo

git push origin main (for all file push to remote site)

* init Command

- init - used to create a new git repo
`git init`

`git remote add origin <link>`

`git remote -v` (to verify remote)

`git branch` (to check branch)

`git branch -M main` (to rename branch)

`git push origin main`

- 'git init' is a command used to initialize a new git repository in the current directory.
- When you run this command, git creates a new directory 'git' in the current directory, which contains all the necessary files and subdirectories for the repository.

Note: if you want to add / commit simultaneously for single change

`git commit -am "added"`

`git push -u origin main`

* Workflow :-

GitHub

Code change



Commit

Local Git

Code change



"add"

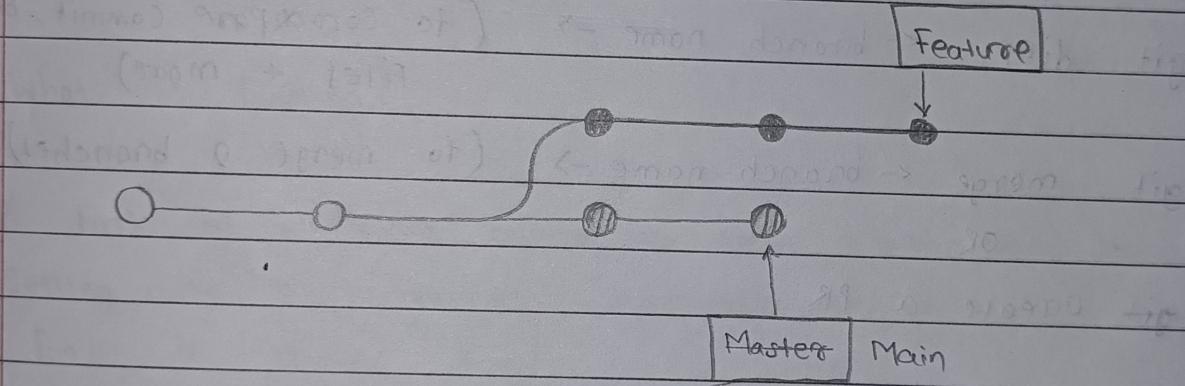
stage a change

"Commit" Commit change



"Push"

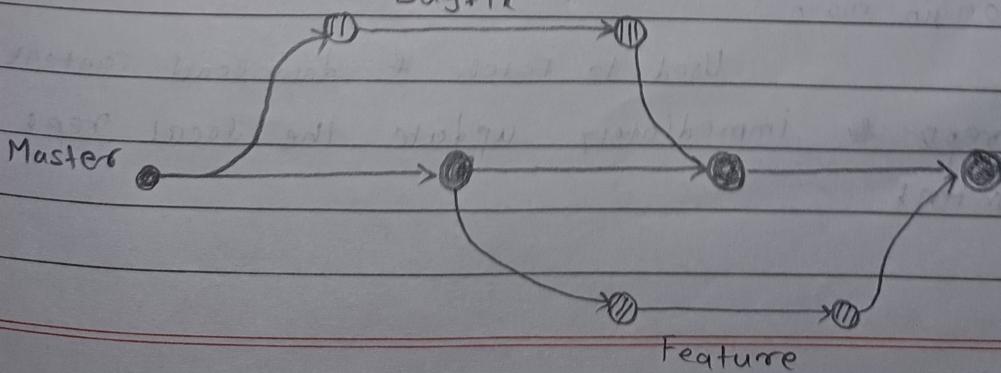
* Git Branches :-



Benefits :- 1) Isolation of work 2) Parallel development.

3) Code review & collaboration 4) Stable release
5) Version control.

Bugfix



* Branch Command :-

git branch (to check branch)

git branch -M main (to rename branch)

git checkout <-branch name-> (to navigate)

git checkout -b <new branch name-> (to create new branch)

git branch -d <-branch name-> (to delete branch)

* Merge Code :

git diff <-branch name-> (to compare commit, branches, files + more)

git merge <-branch name-> (to merge 2 branches)
OR

git Create a PR

* Pull request : if ~~lets~~ gets you tell others about change you have pushed to a branch in a repository on Github.

* git pull origin main :

Used to fetch & download content from a remote repo & immediately update the local repo to match that content



Fixing mistake

Case 1 : Staged Changes

git reset <- file name ->

git reset

2] Case 2 : Committed changes (for one commit)

git reset HEAD~1

3] Case 3 : Committed changes (for many commit)

git reset <- commit hash ->

git reset --hard <- commit hash ->

* What is Forking?

A fork is a new repository that shares code + visibility
Setting with the original "upstream" repository.
fork is rough copy.