Git & GitHub

Git :-

- Free & open source Version Control System.
  ↓
  tools that help to track changes in code.

Advantages:- 1) track history
2) help to collaborate

Github:-

- website where we host repositories online.

NOTE:- README.md }→ Description
↓
markdown

Using Git :-

```
git --version
```

}→ MAC / Terminal

Configuring Git :-

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

git config --global user.email "someone@email.com"
```

Basic Commands:-

- Clone: Cloning a repository on our local machine
- status: displays the state of the code

Clone:-

Syntax: git clone $\underbrace{\text{https:// — — —}}_{\text{link}}$

Status:-

Syntax: git status

Untracked
Unmodified
modified
staged

- add: add new or changed files in your working directory to the Git staging area.

  Syntax: git add filename } → single file

  git add . } → All files

- commit: is the record of change.

  Syntax: git commit -m "some message"

- push: upload local repo content to remote repo.

Syntax: `git push origin main`

- init: used to create a new git repo

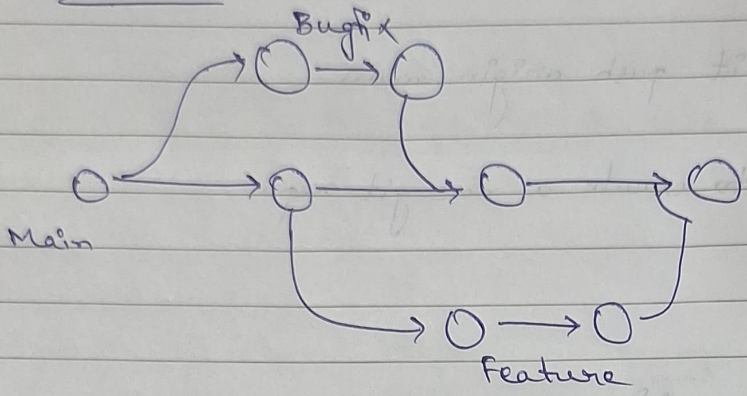
```
git init

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 push -u origin main }→ After this no need
                            to write origin main
                              everytime.
```

Workflow :-

Github	Local Git	
code changes	code changes	
↓	↓	
commit	"add"	stage a change
	↓	
	"commit"	commit changes
	↓	
	"push"	

Git Branches:→



Branch Commands:→

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)

Merging Code:→

- git diff <-branch name-> (to compare commits, branches, files & more)

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

OR

Create a PR

Basic Commands :-

- git pull origin main

used to fetch and download content from a 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 differences in code b/w two commits.

Fixing Mistakes :-

Case 1: staged changes

```
git reset <-file name->
git reset
```

Case 2: commited changes (for one commit)

```
git reset HEAD~1
```

$\underbrace{\text{HEAD}}$ ↳ ptr to last commit

Case 3: commited changes (for many commits)

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

Forking :-

- A fork is a new repository that shares code and visibility settings with the original "upstream" repository.
- Fork is a rough copy.