✯ GIT & GITHUB ✯

APNA COLLEGE

* **Git** :- Git is a version control system.
  → is a tool that helps to track changes in code
  - Popular
  - free & open source
  - Fast & scalable

* **Uses of Git** :- 1) track the History
  2) Collaborate

☐ To check Git version
→ git --version

☐ To view list of files
→ ls

☐ To check working directory
→ Pwd

☐ To check all commits
→ git log

* **Configuring Git**

→ git config --global user.name "My Name"
→ git config --global user.email "someone@email.com"
→ git config --list

* **Clone & Status**

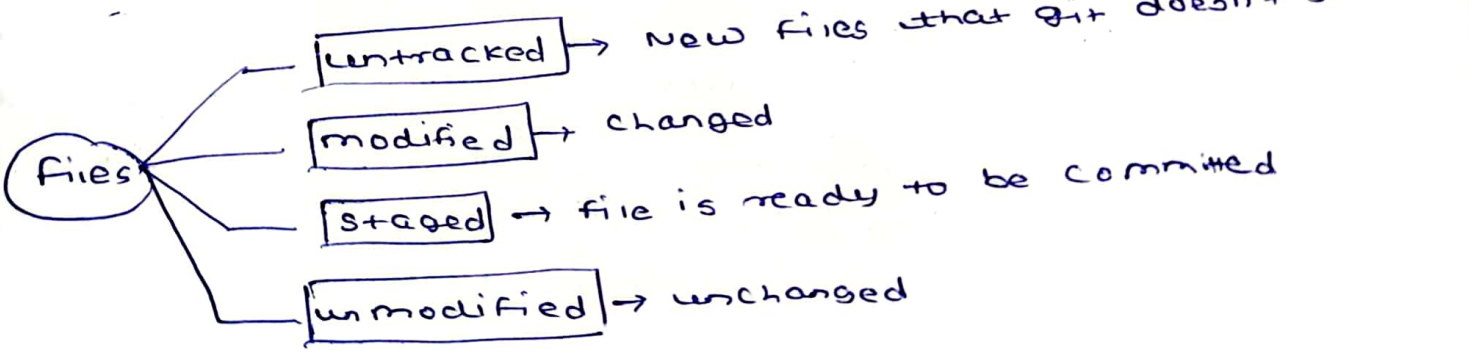
remote [Github] files
local [Laptop/PC] files

Clone - Cloning a repository in our local machine

→ git clone <some link>

Status - display the state of the code

→ git status



Change (modified) / New file (untracked)

↓

add (staged)

↓

commit (unchanged)

* Add & Commit *

add - adds new or changed files in your working directory to the Git staging Area.

→ git add <file name> } --- to add single file

or

→ git add . } ---- to add all files

or

→ git add -A

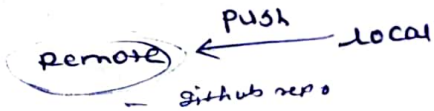
Commit:- It is the record of change

→ git commit -m "some message"

* Push Command:

Push- upload local repo content to remote repo.

→ git push origin main/master



* 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

Note: If we used Push command with upstream (-u) like git ~~add~~ push -u origin main.

It means we have to push all nesst local files or code in same (origin main) branch so after that we don't need to write origin main in push command just write ↵

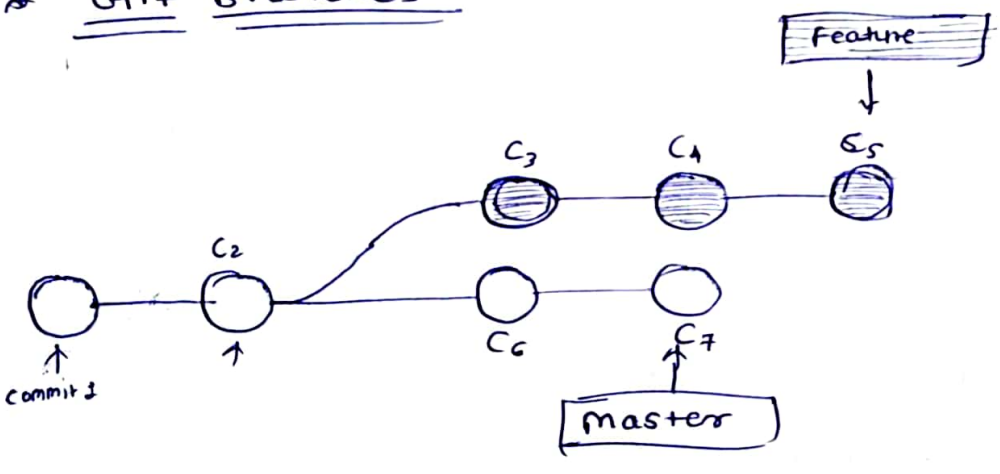
→ git push

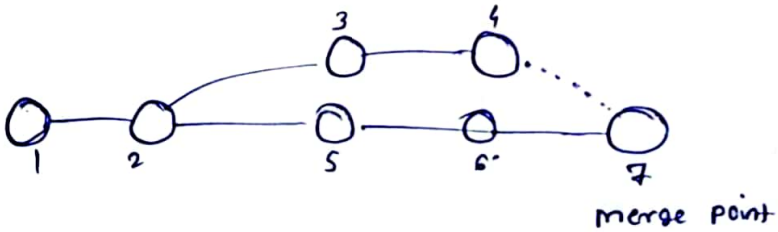
* Workflow:

Local git

Github Repo
↓
clone
↓
changes
↓
add
↓
commit
↓
Push

* Git Branches





→ git branch (to check Branch)

→ git branch -M main (to rename Branch)

→ git checkout <branch name> (to Navigate) → To go from one branch to another

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

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

* merging code

way 1

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

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

way 2

Create a PR (pull request)

* pull Request:

It lets you tell others about changes you've pushed to branch in a repository on GitHub.

* pull command:-

→ git pull origin main

Used to fetch and download content from a remote repo and immediately update the local repo to match the content.

* merge conflicts resolving *:-

An event that takes place when Git is unable to automatically resolve differences in code between two commits.

1) PR
2) git merge

* Undoing changes :-

Case 1 : Staged changes (add)

→ git reset <file name>

→ git reset

Case 2 : Commited changes (For one commit)

→ git reset HEAD~1

Case 3 : Commited changes (For many commits)

→ git reset <commit hash>

→ git reset --hard <commit hash>

* Fork :-

A Fork is a new repository that shares code and visibility setting with the original "upstream" repository.

Fork is a rough copy.
