GITHUB

My notes (Apna College)

Git

Version Control System is a tools that helps to track changes in code

Git is a Version Control System. It is :

popular

free & Open Source

fast & scalable

Github

Website that allows developers to store and manage their code using Git.

folder (*Repository*)

https://github.com

Github Account

- Create a new repository : apnacollege-demo

- Make our first commit

Setting up Git

Visual Studio Code

Windows (Git Bash)

Mac (Terminal)

git --version

Configuring Git

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

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

git config --list

```
[shradhakhapra@Shradhas-MacBook-Air ~ % git config --global user.name "Student Ap]
naCollege"
[shradhakhapra@Shradhas-MacBook-Air ~ % git config --global user.email "student@a]
pnacollege.in"
[shradhakhapra@Shradhas-MacBook-Air ~ % git config --list                        ]
credential.helper=osxkeychain
user.name=Student ApnaCollege
user.email=student@apnacollege.in
shradhakhapra@Shradhas-MacBook-Air ~ %
```

Clone & Status

remote

[Github]

local

[laptop / PC]

Clone - Cloning a repository on our local machine

git clone <- some link ->

status - displays the state of the code

git status

EXPLORER

GITDEMO

OUTLINE

TIMELINE

PROBLEMS OUTPUT **TERMINAL** DEBUG CONSOLE

zsh

shradhakhapra@Shradhas-MacBook-Air GitDemo % git clone https://github.com/student-a
demo.git
Cloning into 'apnacollege-demo'...

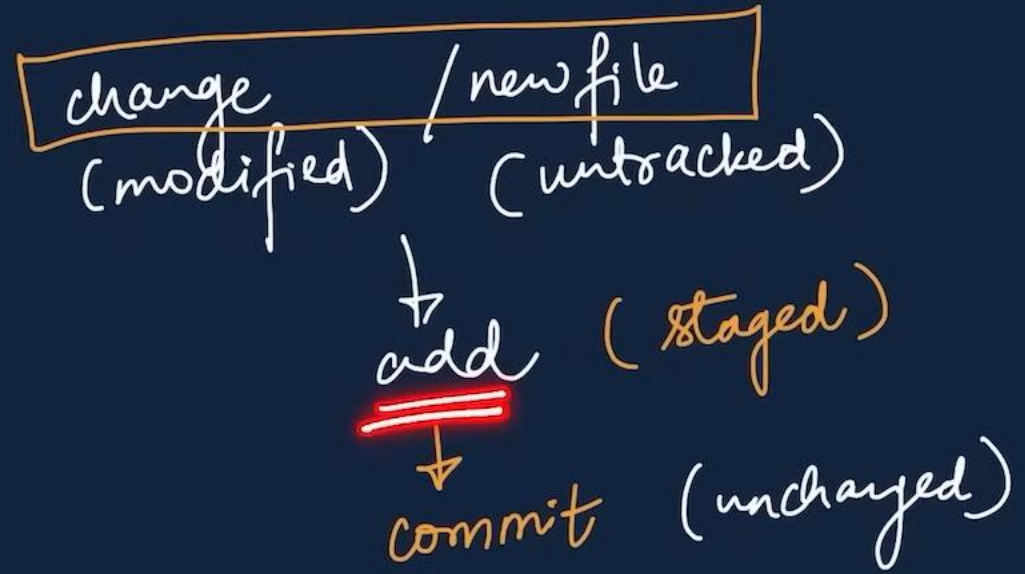**untracked**

new files that git doesn't yet track

modified

changed

staged

file is ready to be committed

unmodified

unchanged

change / new file
(modified) (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 ->

commit - it is the record of change

git commit -m "some message"

apnacollege-demo

apnacollege-demo > README.md > # apnacollege-demo

```
1   # apnacollege-demo
2
3   This is my first Git Repository.
4   <br>
5   Author - Shradha (Apna College)
6
```

PROBLEMS   OUTPUT   TERMINAL   DEBUG CONSOLE   zsh - apnacollege-demo

shradhakhapra@Shradhas-MacBook-Air apnacollege-demo % git add .
shradhakhapra@Shradhas-MacBook-Air apnacollege-demo %

OUTLINE

TIMELINE

Push Command

push - upload local repo content to remote repo

git push origin main

Init Command

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

git init

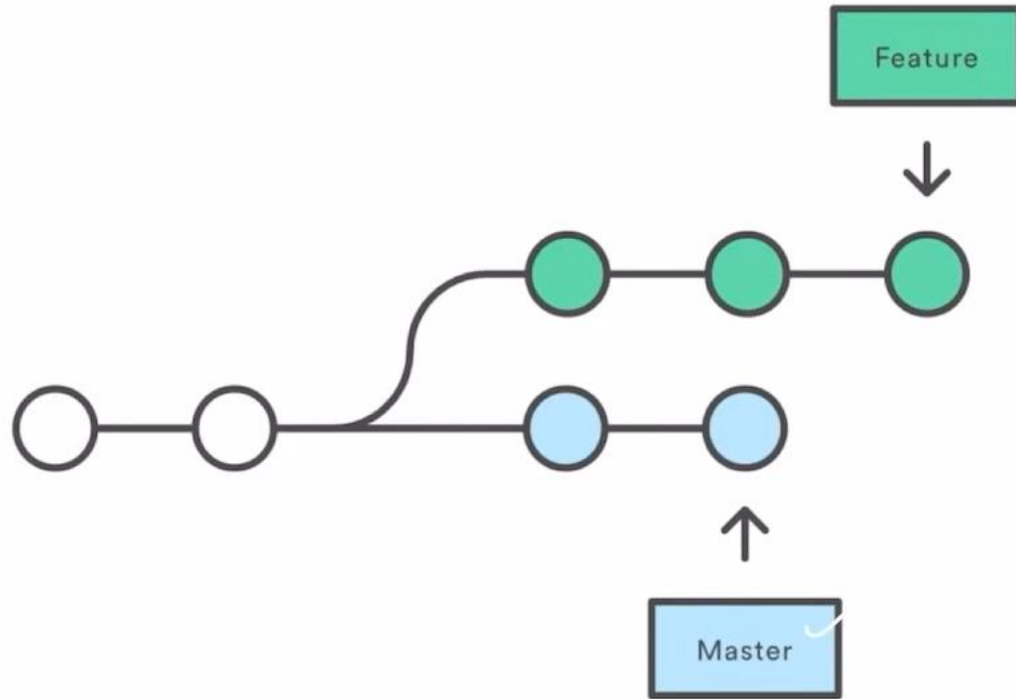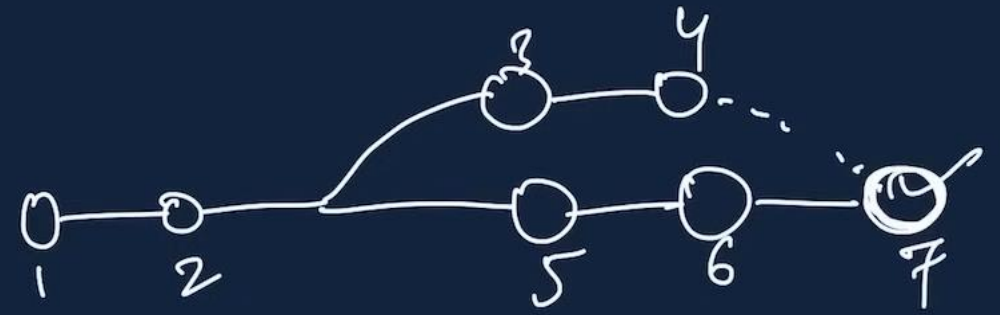*git remote add origin <- link ->*

git remote -v   (to verify remote)

git branch   (to check branch)

git branch -M main   (to rename branch)

git push origin main

Git 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

Way 1

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

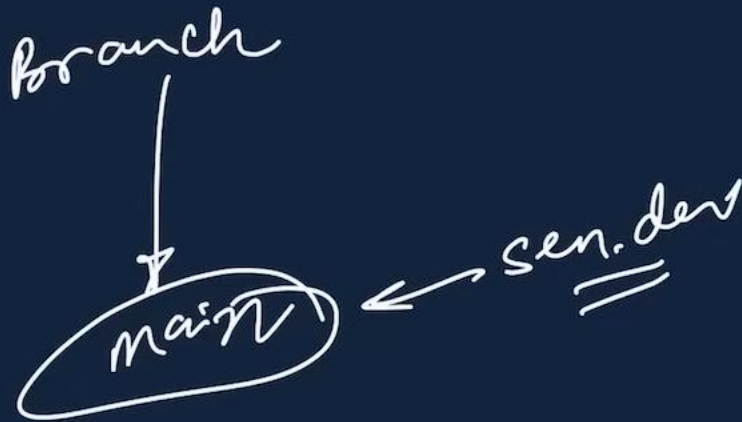*git merge <- branch name->*   (to merge 2 branches)

Way 2

Create a PR

Pull Request

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

student-apnacollege / localrepo

Type / to search

<> Code   ⊙ Issues   ⑂ Pull requests   ▷ Actions   ⊞ Projects   📖 Wiki   🛡 Security   📈 Insights   ⚙ Settings

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also **compare across forks**.

⇅   base: **main** ▾   ←   compare: **feature1** ▾   ✓ **Able to merge.** These branches can be automatically merged.

Add new feature

Write   Preview   H B *I* ≔ <> 🔗 ≔ ≔ ☑ @ ⟷ ↩ ⊟

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.   ℳ↓

Create pull request ▾

ⓘ Remember, contributions to this repository should follow our **GitHub Community Guidelines**.

Reviewers   ⚙
No reviews

Assignees   ⚙
No one—assign yourself

Labels   ⚙
None yet

Projects   ⚙
None yet

Milestone
No mileston

Developm
Use Closing
automatical

Helpful resour

Resolving Merge Conflicts

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

    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 ->

Undoing Changes

HEAD~1

Case 1 : staged changes (add ~)

    git reset <- file name ->

    git reset

latest

1 — 2 — 3

HEAD

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 ->
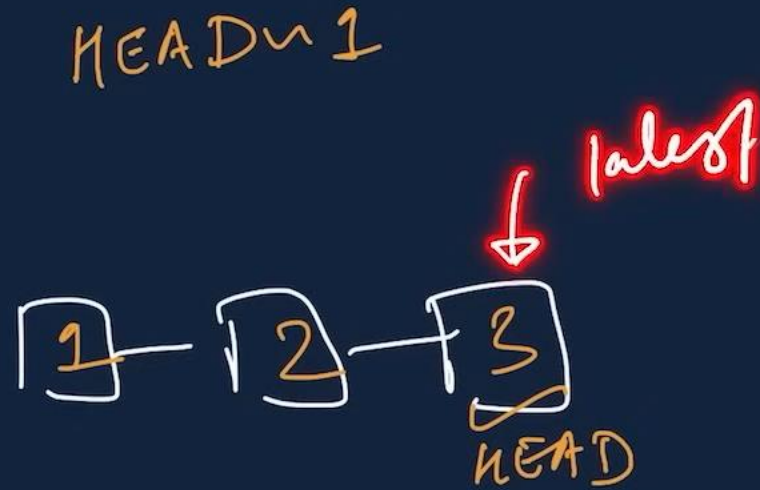
Fork

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

Fork is a rough copy.