Git & Github

↓ software (VCS)

↓ service provider

version control system

track files for changes

Path :-

→ clear Basics

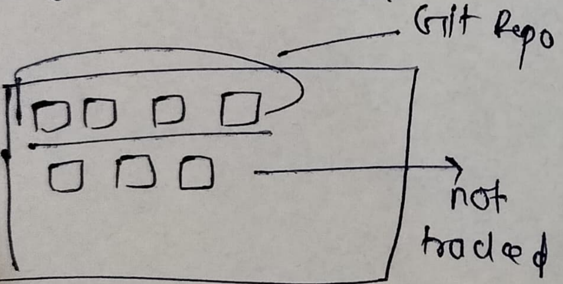
→ Use it daily

→ face the problem → solve the problem

Repo → Repostiary (folder)

→Repo is git on system

(1) git --version



-we have to mention we folder to track even git is installed in local system.

(2) git status — check the status

(3) git init (one time per project)

.git → a hidden folder to keep history of all files & folders, sub-folders

inside .git

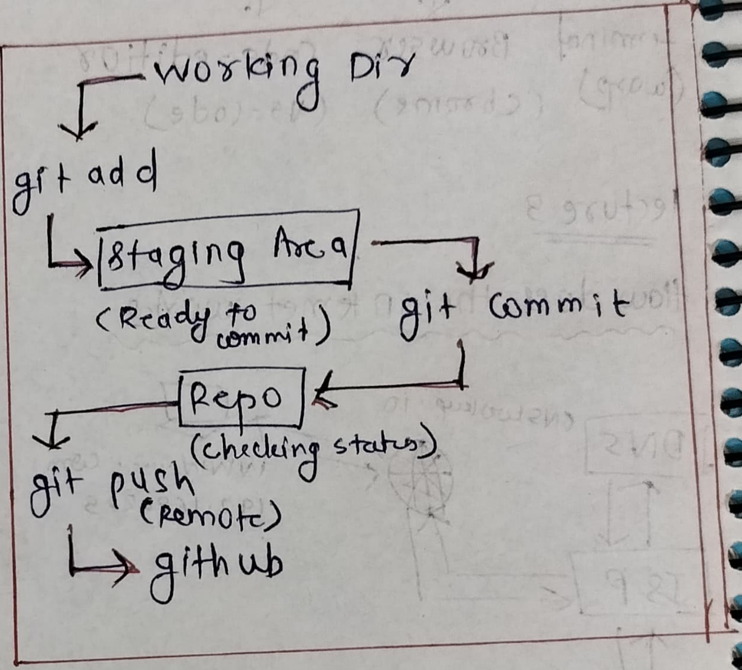
HEAD description info/ refs/
config hooks/ objects/

commit

↓

check point (like game)

Write → Add → Commit
(i) (ii) (iii)



git add testone.txt
(Add to staging Area)

git commit -m "add file one"
git status

git log

(Details commit id, Author, date, message)

Git follows the principle of Atomic commits i.e one commit does one job..

(i) keep commits centric to one feature, one component or fix

(ii) present or Past commit message

- Depends of Present tense, Imperative ij

~~git config~~

git config --global
(Get & set repo or global options)

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

```
git config --global core.editor
                      "code --wait"
```

git log -oneline
"gives log in oneline"

→ ~~.gitgonee~~ .gitignore

(Env, holds all sensitive keys

.gitignore

```
.env
folder/
node-modules/
```

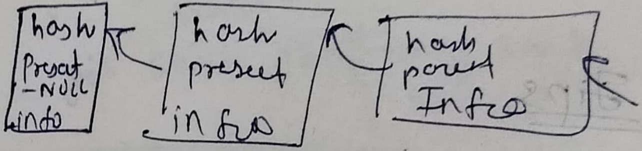
Tip:-

.gitignore generator (Google)

Where it is stored (git config--)

cat .gitconfig (Home dir)
on
Windows
(users\)

→ Working of commit



how to check this hashs

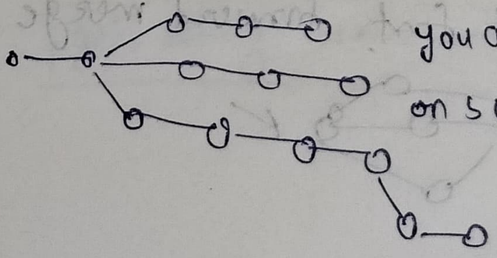
.git

Git Branches

git branch
* main

git branch nav-bar

(like an alternative timeline)



you are always on some branch

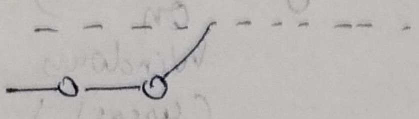
→ HEAD → main
head points to where a branch is currently at.

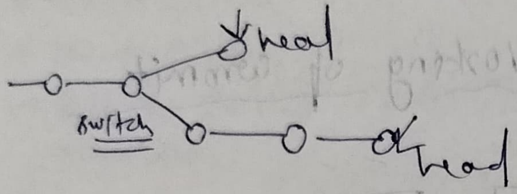
create branch

```
git branch. nav-bar
```

move to new branch

```
git checkout nav-bar
```



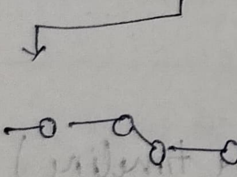


Tips

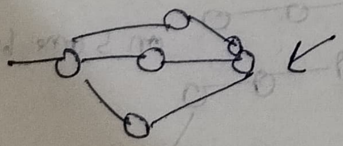
(i) commit before switching to another branch

(ii) go to .git folder & checkout HEAD file.

→ Merging of Branches



→ Not fast forward merge



```
git merge nav-bar
```

" comt "

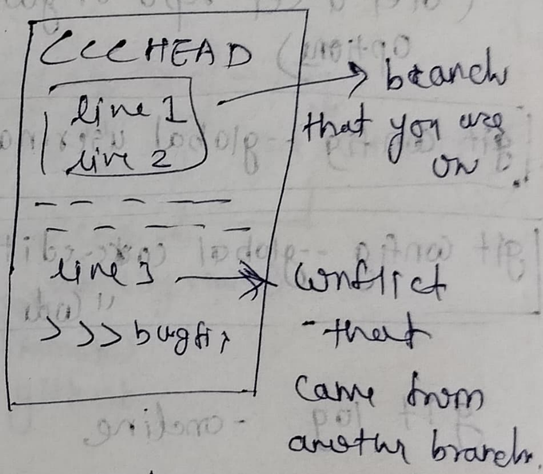
Delete branch

```
git branch -d nav-bar
```

Shortcuts

```
git checkout -b footer
```

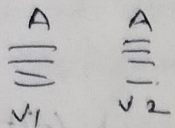
→ git tries best to resolve conflicts



keep whatever you want, remove markers & save.

Diff / stash

Git diff (informative command)
shows difference between



(compare working with staging)

How to read diff

- ★ a → file 1 & b → file 2 (same file over time)

symbol.

- ★ - - - file 1
- ★ + + + file 2

{ Indicates changes in file

- ★ changes in lines & little preview of it

```
git diff --staged
git diff 9cdf011 9dd877
```

```
git diff branchone..branchtwo
```

Git Stash (very Carful).

→ Create a repo, work & Commit on main

→ switch to another branch & work

→ conflicty changes do not allow to switch branch, without commit

```
git switch -c fixbug
```

new branch & switched to it

git stash (you can switch branch) (temporary shelf)

git stash pop (Bring Back changes

```
git stash list
```

- git stash apply (apply changes & keep them in stash)

More Commands

git checkout <hash>
git switch main
git checkout HEAD~2
git restore filename

```
git reflog
```

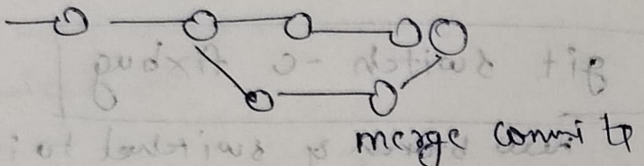
→ rare used

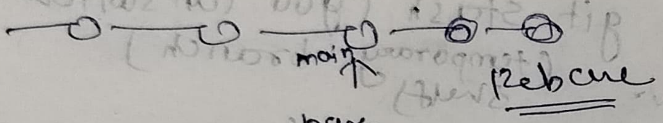
Git rebase ((scary))

Rebase

→ alternative to merging

→ clean up tool (clean up commits).





tip:-

if you are on main branch never run this command.

```
git rebase main
```

github

Generate SSH key & connect to computer.

→ Github

Git is a software and Github is a service to host git online

Github → collaboration + Backup & opensource

Gitlab or BitBucket

```
git clone <URL>
git config --global user.name " "
      "          "     user.email " "
```

- setup SSH keys to connect with github, github uses SSH to allows you to push code. Password based code push is not allowed.

```
git remote -v
git remote add name url
git remote add origin <URL>
git remote rename oldname newname
git remote remove name
```

```
git push <remote> <branch>
git push origin main
```

```
git push -u origin main
```

setup upstream that allows you to run future command.

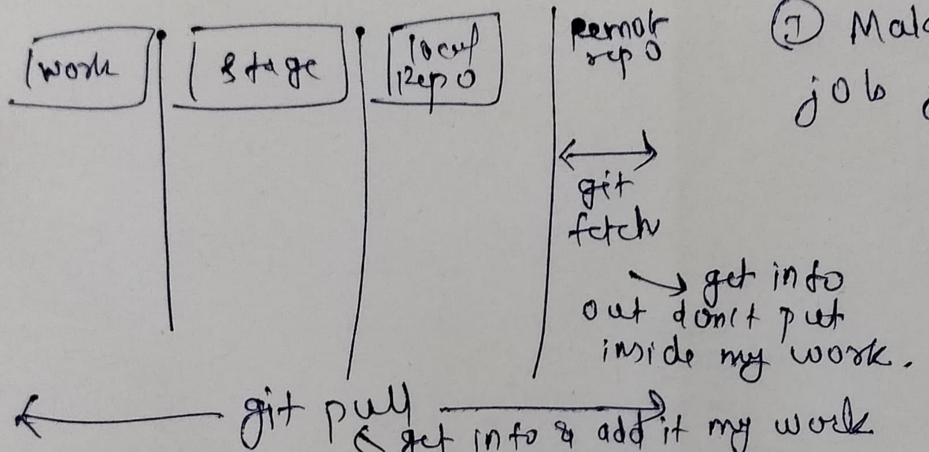
git push

When we clone a repo, you get just main branch connected, rest of remote branches are not configured.

git switch branch-name

connects remote branch to local

git branch -r


work
stage
local Repo
remote repo
git fetch
get info out don't put inside my work.
git pull
get info & add it my work

git pull = git fectch + git merge

git pull origin main (changes will be merged to main)

Features

- Gist
- codespaces
- pages
- Adding collaborators

Open Source Contribution

Path :-

1. Talk
2. open an issue
3. get the issue
4. work & Add value
5. Make PR & iterate over it
6. Have patience
7. Making PR is not a job guarantee