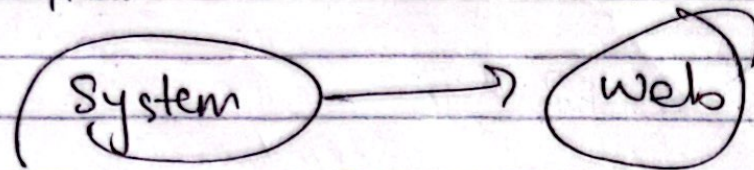


Note

Git → runs on PC

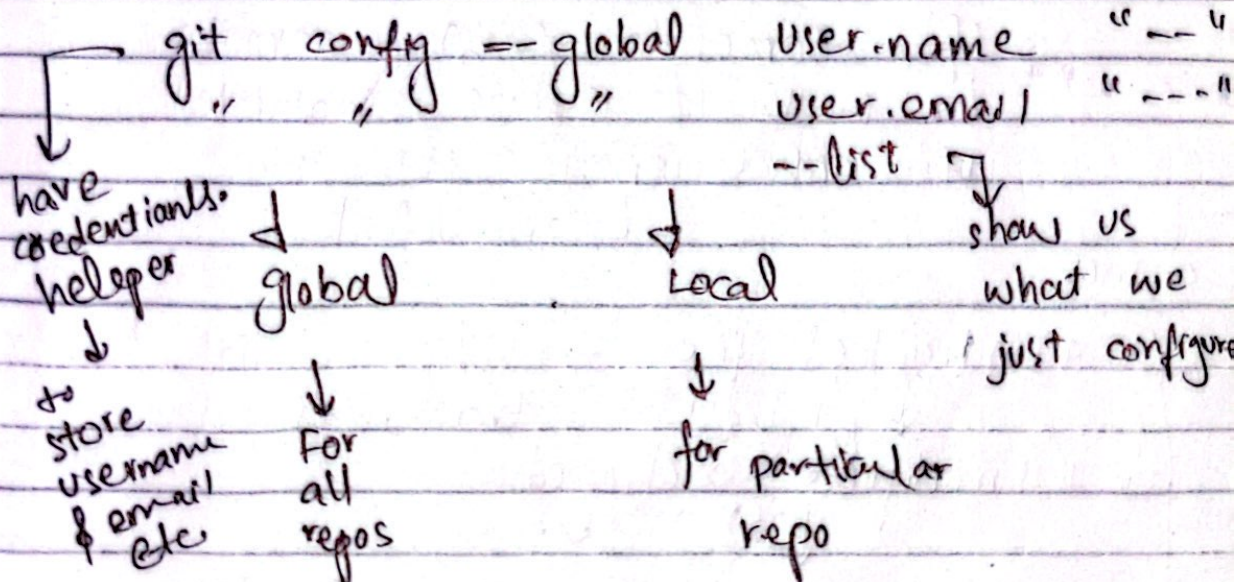
- ① Track the history
- ② Collaborations

GitHub → website



.md → markdown

# Configuration Git → changes → GitHub



~ → symbol is for - root directory  
cloud/web/GitHub → remote  
local - PC

# command

⇒ clone → clone on our system

git clone (Link)

→ cd → change directory



- Tab for auto complete
- Clear to clear terminal
- ls → list files
- hidden files — ls -a  
↳ all files

• git → github file

⇒ Status → display status of code

-git status

if same on local = remote  
↳ nothing to update, (uptodate)  
if not shows us.  
↳ modified

### Statuses

↳ + untracked file → git is not tracking  
↓ or  
↳ + modified → changed  
↓ (add)  
↳ + staged → file is ready to be committed  
↓ (commit)  
↳ + unmodified → unchanged

⇒ Add → adds new or changed file to the git staging area  
git add filename (only one file)

git add . (all files)



⇒ commit → it is the record of change

git commit -m "message here"  
↓  
Try to be as meaningful as possible

⇒ Push → upload local repo content to the remote repo.  
→ github repo name (e.g. PL1)

git push origin main  
↓  
Basic command  
↓  
branch

Remote ← push local  
↓  
default

⇒ Init Command

→ init → used to create a new git repo

git init

git remote add origin (link)  
// adding remote repo for local repo



⇒ `git remote -v`  
(verify remote)

⇒ `git branch` → shows us branch

⇒ `git branch -M (newname)`  
// for renaming a branch

⇒ `git push -u origin main`  
↓  
set upstream  
if used, no need to  
add origin main again  
& again  
↓  
(git push)

## # Git Branches

`git branch` → check branch

`git branch -M newname` → rename.

`git checkout branchname` → Navigate  
b/w branches

`git checkout -b newbranch` → new branch  
name created.



→ git branch -d branchname → to delete branch  
↓  
(not current branch).

## # Merging code

Way 1

- git diff branchname  
↓

to compare commits, branches, files & more

- git merge branch name

↓ to merge two branches

- abort → to abort merge.

Way 2

Create a Pull Request (PR)

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

→ Senior dev or repo owner needs to merge on our request or



=> pull command

git pull origin main

- used to fetch and download contents from a remote repo and update the local repo to match that content.

### # Resolving Merge conflicts

↓  
Take place when git is unable to automatically resolve difference in code b/w two branches.

- Manually tell git which branch change/commit we want.

### # undoing changes

(Case 1) staged changes

git reset (filename)

git reset



⇒ git log → show an commit

(Case 2) Committed changes (for one commit)

git reset HEAD ~ 1

HEAD ~ 1  
└───┘  
Latest commit

~ 1 → Head ko 1  
step picky ly  
jao.

(Case 3) Committed changes (for many commits)

git reset commit hash

git reset --hard commit hash

hash → code (id)  
↓  
unique.

changes removed from code as well.



## ⇒ Fork

- fork is a new repo that shares code and visibility setting with the original "upstream" repo.
- Fork is a rough copy.

## ⇒ Stash

↳ want to store changes in a box, when you need that you can get from that box. and if not, they are saved there, not discarded.

- git stash

↳ saved with random name.

- git stash -m "Name/message".

↳ saved with name.

- git stash list.

↳ shows how many changes.

- git stash apply.

↳ last (latest) stash apply, come back in code.



→ git stash apply stash@{2}

↓  
stash number  
you want  
to apply.  
check number  
using stash list.

→ git stash pop

↳ latest stash got deleted.

→ git stash drop stash@{2}

↳ delete by name/id.

→ git stash clear

↳ all stash deleted.

→ git stash branch "branchname" stashname  
stash@{0}

↳ create a new branch  
with that stash code and  
with that time ~~stash~~ branch code  
as well.