

بِسْمِ اللّٰهِ الرَّحْمٰنِ الرَّحِيْمِ

MUHAMMAD HARIS

Road To MACHINE LEARNING ار. شادی اللہ

"اللّٰهُ تَعَالٰی فَضْلُ سَنَامَلَنَ بِهِ مَكْلُونٌ هُوَ حَاجَاتٌ عَلَى"

PHASE 1 : COURSE 3

Complete Git and Github Course

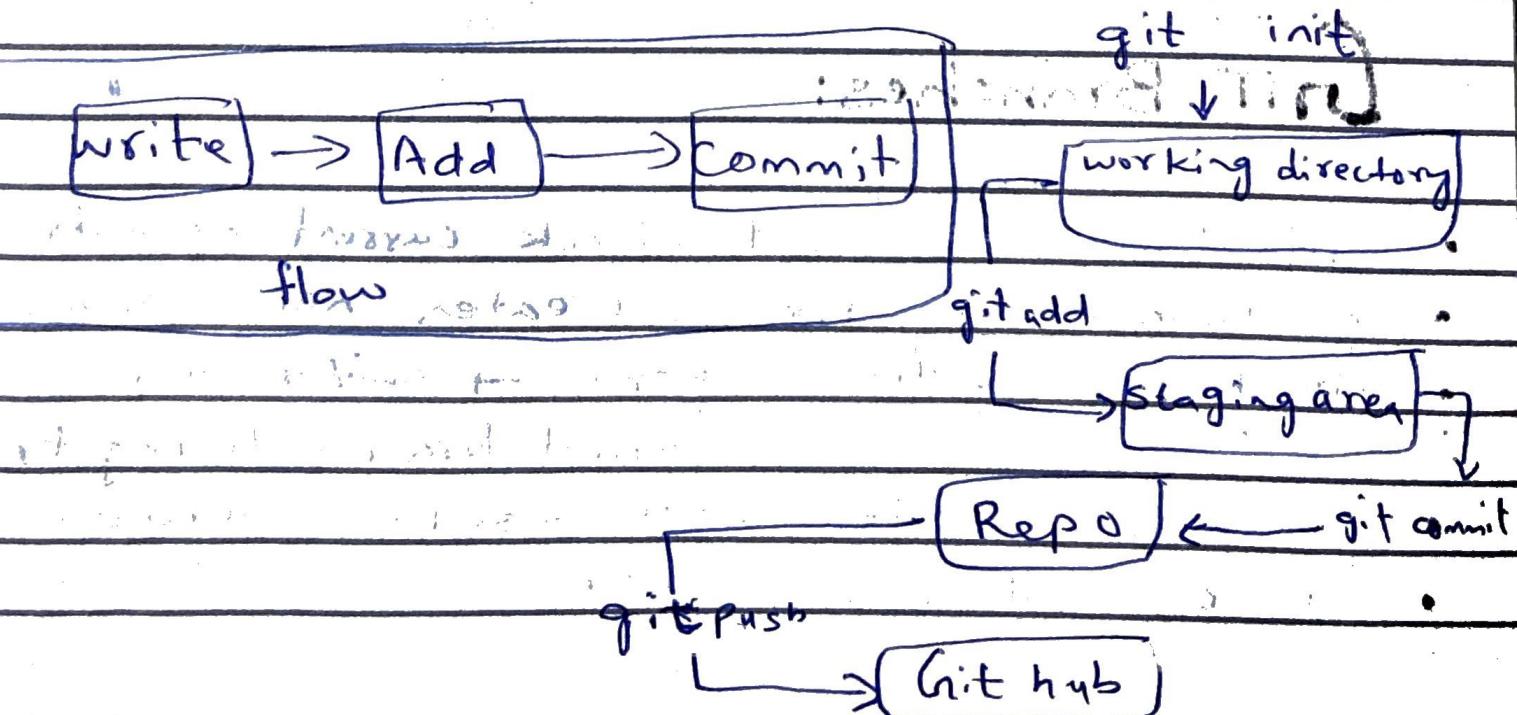
By
Chai Aus Code

Commands:

- git --version
- cd → to go to desire folder
- pwd → check where we are now

Creating Repository

- git status → check repo is exist or not
- git init → to initialize
- ls -la → to check hidden files
- ls -l → - files and folder starts with d.



- `git add .` → Means all files
- `git add file.txt file2.txt` → Means add by choice
- `git rm --cached <file>` → to unstaged
- `git commit -m "message"`
- `git log` → to check history of Repo
- `git log --oneline` → shows commit history in one line
- `git` don't track empty ~~files~~ / folders
so we add file `.gitkeep` to let folder not be deleted
for his presence and then `git track` it
- `.gitignore` files to ignore change
and dont want to track `.gitkeep`, `.env`
`.vscode` files, it's our choice

Git Branches:

- `git branch` → to check current branch
- `git branch bug-fix` → creates new branch
- `git branch switch bug-fix` → switch from current branch to bug-fix
- `git switch -c dark-mode` → switch and create
- `git checkout orange-mode`

Merging

CREATE (1/n)

bug-fix → merge master

- git merge bug-fix (if we change master and then wants to update bug-fix then we perform merge on (bug-fix))
- git branch -m <old-branch-name> <new-branch-name>
 - To rename a branch
- git branch -d <branch-name> → for delete

Git DIFF

- git diff → used for comparison and list
 - git diff --staged (compares staging area)
 - git diff <branch-name-one> <branch-name-two> (compared branches)
 - git diff <commit-hash-one>..<commit-hash-two>
 - we can use this or space both same.
- We use diff and compares anything there are multiple commands.

GIT STASH

→ Stash is a way to save your changes in a temporary location.

- `git stash` → switch branches with commit

- `git stash save "..."` → Give stash a name

- `git stash list` → view stash list

- `git stash apply` → undo stash

- `git stash apply stash@{0}` → choice

- `git stash pop` → bring hidden stuff back
 - uses when we don't want to apply stash and just want to see uncommitted things. Then and also delete from stash list

- `stash apply` → don't remove from stash list

- `stash pop` → delete from stash list.

- `git stash clear` → clear stash list tip

- `git stash drop stash@{0}` → clean choice

- `git stash apply stash@{0} <branch-name>`

GIT TAG

- `git tag <tag-name>`

- `git tag -a <tag-name> -m "Release 1.0"`

- `git tag` → list all tags

- `git tag <tag-name> <commit-hash>`

- `git push origin <tag-name>` → push to remote repo

- `git tag -d <tag-name>` → del tag on local

- `git push origin :<tag-name>` → del tag on remote repo

REBASE

- Ensure you are on the branch you want to rebase
- 90-95% don't rebase on master or main branch.
- git checkout on sub branch
 - then check that you are desired branch
- git rebase master
- used to avoid extra commits
- used to realigned the base
- git rebase --onto , git merge --onto

REFLOG

- git reflog → History of your head moves
- git reflog <commit - hash>
- git reset --hard <commit - hash>
 - more prefer
not prefer
 - ↑
 - ↓
 - reverses lost commits or
change
- (git reflog <commit - hash>
 - get reset --hard Head@{1} → choice