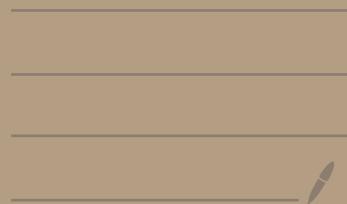


GIT STARTERS



1- 'git init' → Powers your folder to be managed by git, and initializes a new repository. It also creates a .git folder that has all the relevant logic to manage versions of your project.

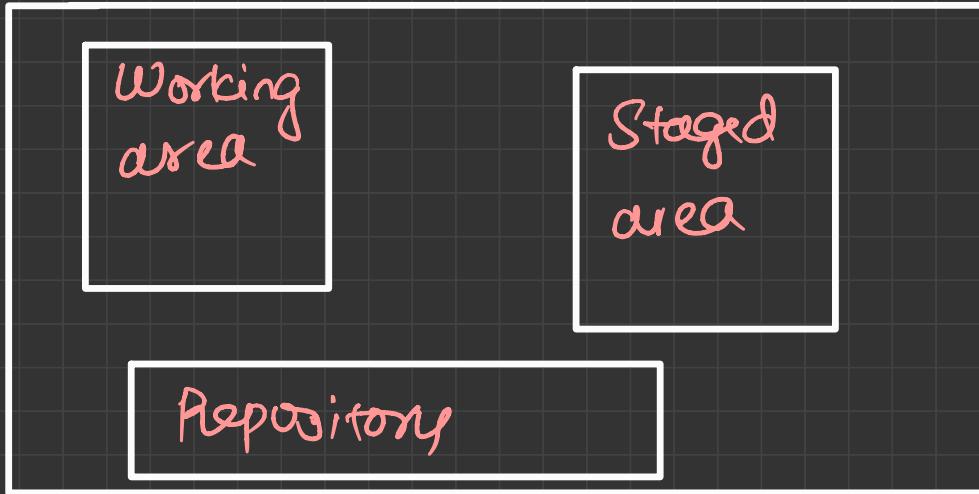
2. 'git status'

On branch master

No commits yet



Any file change
lie in any
one of these
regions.



Initially any change you do lies in 'working area'. You technically haven't started tracking it using git.

3. 'Working area' - There can be a bunch of files that are not currently handled by git. It means the changes done or to be done in those files are not managed by git yet. A file in working area is considered to be not in the staging area.

When we do 'git status' & we see a bunch of files in the 'untracked files', then these files are said to be in the working area.

4. Staging / Staged Area : tells you what all files are going to part of next version. This staging area is the place where git knows what changes will be done from the last version to the next version.

How do move files to 'staging area'?

5. `git add <file-name>`

→ Moves file from 'untracked region' to staging area.

6. `git rm --cached <file_name>`

→ Move file from staging area to working area.

7. Repository Area: This area actually contains the details of all your previous registered

version. The files in this area are already managed by git & knows their version history.

How to register a Version?

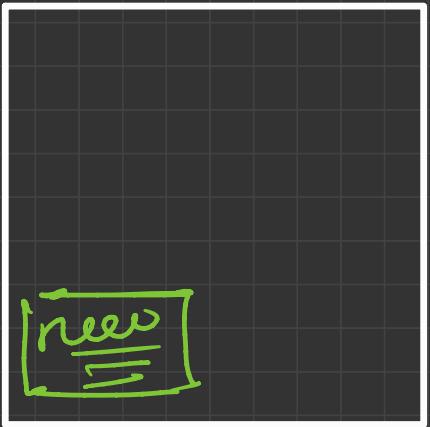
In git, versions are managed by 'commits'.

8. commit: commit is a particular version of project. It captures a particular snapshot of the project's staged changes and creates a version out of it.
git commit → Register staging changes.

9. 'git log' → list down all the commits of the repository - If you want to exit out of git log prompt, press 'q'.

10. git restore <file-name> - Delete everything in the staging area that is not yet committed.

11. git restore --staged <file-name> - Remove file changes from staging to working area.



→ Repo → committed

git status

changes will be in
working area.

git add <file>

→ staging area

Not git restore won't
work.

[Moves from staging
area to working
area]

git restore --staged
<file>

git restore <file>

12. Difference between git rm & git
restore

If you want to remove the whole file back to untracked state, then we do git rm, otherwise if we just want the changes to be moved to working area we do git restore.

13. git diff commit1 commit2 - gives difference of all file changes between two commits.

14. git commit -m `{your message}` - Commit with msg.

Connect Local repo & GITHUB REPO .

15. `git remote add origin <url>`

Add new link to the remote repo.

16. `git remote rm <name-of-remote>`

This command deletes a remote connection

17. `git push origin main`
↳ upload changes .

18. `git remote` - Lists down all remote connection names.

19. Remote connection - It helps you link two git repos. for uploading & downloading changes for each.

20. `git remote rename <old-name> <new name>` : Rename the remote connectn.

NOTE: Name of the remote connectn. is always used to establish connection b/w the repos.

21. `git add <file1> <file2>` :
Adds multiple file changes together
in the staging area.
22. `git add .` :
Add all files from working repo.
to staging repo.
23. `git pull <remote-name> <branch name>`
Download recent changes.

Recommended practice to do:

- make changes
- git add <file>
- git commit
- git pull
- git push

NOTE: Merge conflicts can occur if multiple people try to make changes to the same file, and then collaborate -