(Git)

⇒ On Git installation
- → Git Command line tool
- → Git bash (terminal program)

⇒ On Git terminal :-

- $ Pwd

⇒ Tells us current directory

- $ cd Desktop or $ cd /c

⇒ Takes us in Desktop or /c (folder)

- To configure name

⇒ $ git config --global user.name "..."

- To config email.

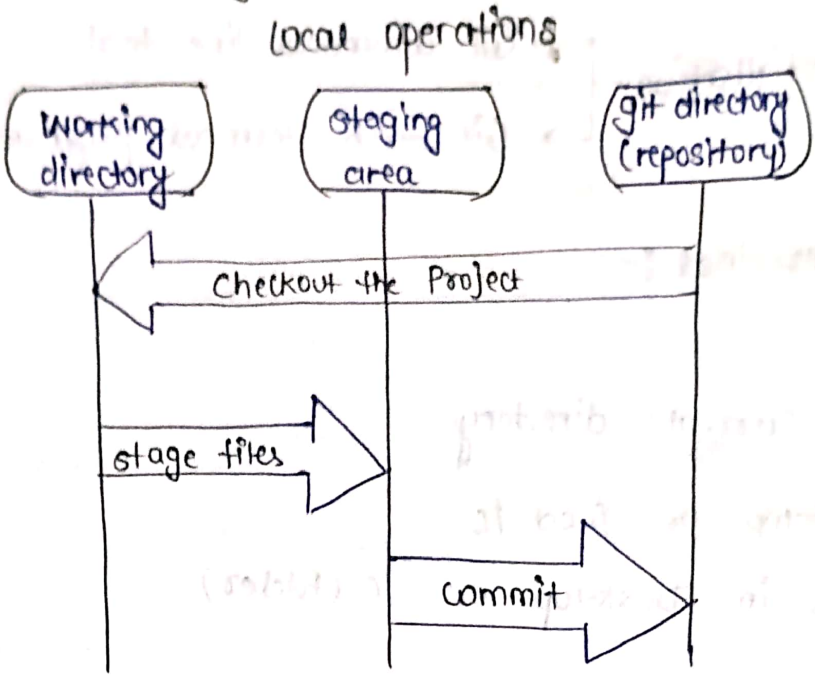
⇒ $ git config --global user.email "..."

- To get list

⇒ $ git config --list

-

Git → Three Stages Architecture



Track a project in Git :-

- check status of your repository -
  git status
- Initialize a repository in Git :-
  git init
- Stage all files :-
  git add --a
- Stage particular file :-
  git add file_name
- Commit files and write a msg -
  git commit -m "Initial Commit"
- To check all files update :-
  git log

- To delete a git repository -

  rm -rf .git where git is a folder/ repository created by git when we initialized the repository.

- To clone a repository -

  git clone "link or path of repository"

- To go to current working directory -

  pwd - (working directory)

- To list all directories -

  ls - (list)

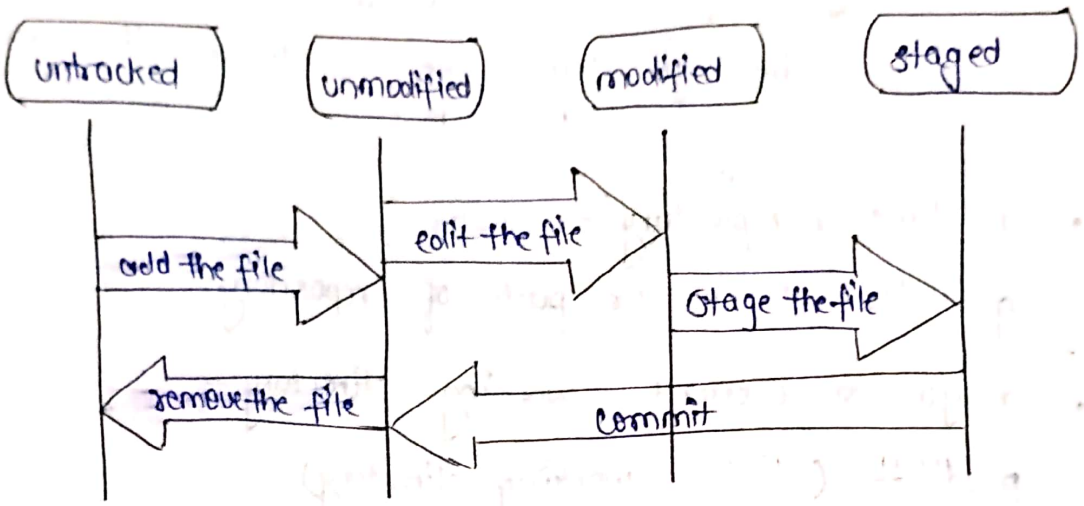
- To change directory -

  cd - (change directory)

- To quit from a directory or process -

  q

File Status Lifecycle :-



⇒ To create a blank file in repository / folder:

touch file_name

Ex:- touch error.log

- Create a file to ignore such blank files with name of .gitignore -

touch .gitignore

* Put the files you want to ignore in gitignore

Ex:- *.log → file

dir/ → all folder or directory

/dir/ → first folder or directory only

*.log → all files with .log extension

- To see difference b/w staged files & Current working file -

git diff

- To see difference b/w committed files -

git diff --staged

→ To commit and stage files in one line -

git commit -a -m "first commit"

↳ it only commit track files.
not untrack files.

- To remove / delete a file from folder -

git rm file_name

- To ~~move~~ rename a file -

git mv old name of file new name of file

Ex:- git mv first.txt first_replaced.txt

- To untrack a file -

git rm --cached file_name

Deep dive into log :-

- To show the difference b/w committments -

git log -p

- To show particular no. of commits -

git log -p -3

- To show commits in short -

git log -stat

- To show commits in one line / short / full -

git log --pretty = oneline
git log --pretty = short
git log --pretty = full

- To show a particular time commits -

  git log --since = 0.years or 0.weeks or 0.days

- To show commits in a format -

  git log --pretty = format: "%h -- %an"

⇒(use git scm website for more format)

- To merge your current commit with particular commit & edit it in that particular commit -

→ Open that previous particular commit

→ git commit --amend

In editor → Do changes

At INDEX in bottom type for Exit -

↳ Escape for type

↳ then [:wq]

Unstaging or unmodifying in Git files :-

- To unstage a file -

  git restore --staged file_name

- To unmodified a file -

  git checkout --space file_name

- To unmodified all modified files :-

  git checkout -f

Alias :-

Ex:- git config --global alias.st status

or

Ex:- git config --global alias.unstage "restore --staged --"

(Work with Git hub)

[remote a repository]

- create folder
- initialize it & then commit it.
- Generate a SSH key

  ssh-keygen (-t rsa -b 4096 -C "email")x
- see path open folder copy key in .pub file and paste it on Git hub.
- To remote it (origin creation) :-

  git remote add origin git@github etc, etc. copy from your repository
- To see all origin of push & pull —

  git remote -v
- To push the file —

  git push -u origin master
- To push a branch in remote —

  ~~push~~ git push origin branch name
- To delete a branch from remote —

  git push -d origin branch name

[Branch in Git]

1) Create a branch —

git checkout -b branch_name

- Switch a branch -

git checkout branch_name

Ex:- git checkout master

- merging of branches :-

git merge branch_name

- To see All branches —

git branch -v

- To check merged branch —

git branch --merged

- To check unmerged branch —

git branch --no-merged

- To delete a branch —

git branch -d issue2

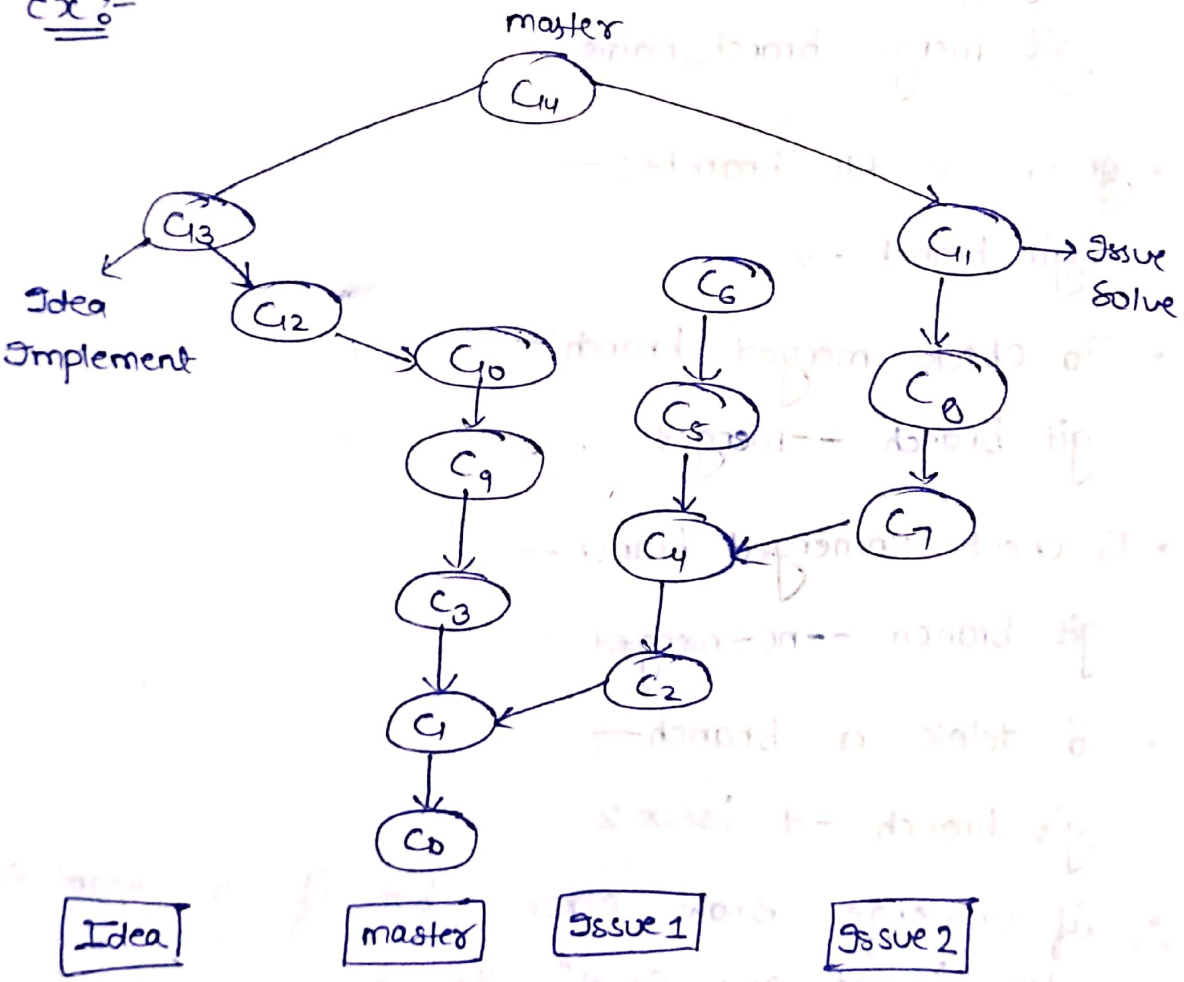
∴ if unmerge show error but if you want to delete it at any Cond^{n}. then —

git branch -D issue2

Branching Workflow :-

- Long running branches
  - 1) master
  - 2) development
  - 3) PU
- Topic branches

Ex :-



Happy

Happy ...