

Git Quick Notes

- Basic Command:

`Git init` → For making/create a git repo in a particular folder.

`Git status` → Provide the current status of the git repo.

`Git add.` & `Git add --a` → To stage the situation.

`git commit` → Syntax. `git commit -m "<comment>"`

- User input information

① `git config --global user.name "<name>"`

→ To check name.

`git config --global user.name`

② `git config --global user.email "<Email>"`

→ To check Email

`git config --global user.Email`

③ `git --help.`

NOTE `rm -rf. git` → To delete the git folder in the folder.

- Cloning a remote git repository from Git hub.

Search & open the repository to be ~~ton~~ cloned with SSH. And copy the SSH address of it.

`git clone <URL/SSH address> - "new name"` → If want else no need.

few important Linux Command:

- ① `pwd` → present working directory. (In which directory you are write now.)
- ② `ls` → List condence. (List all the items in which you are).
- ③ `cd` → change directory [Syntax: `cd (<new directory name>)`]

NOTE To add a folder/file blank in the folder we use.

`touch <file name.type>`

- To make folders ignored By git. (git Ignore)

Steps: `touch gitignore`

then add the file name to be ignored in the file created with file extend & then save it.

- Git Diff. ① `git diff` → Will compare working directory to staging area.
② `git diff -- staged` → It compares staging area with commit area.

NOTE Git: skipping the staging area

`git commit -a -m "<message>"`

- Renamed | Renaming file in git:-

- ① `git rm <file names>` → To delete a file & stage it automatically through git.
- ② `git mv <original name> <new name>` → To rename a file & put automatically in staging area.
- ③ `git rm -- cached <file name>` → To untrack a file from git

git Log: Viewing & changing commits in Git:

- ① `git log` → To check ^{all} log entry
- ② `git log -p` → also provide all log commits
- ③ `git log -1` → Tell the last commit only.
- ④ `git log ->` → This will give all detail but in very summaries way.
- ⑤ `git log --pretty = oneline` → Summaries all entries in one line each commit.
- `git log --pretty = short`
- `git log --pretty = full` → Give detail of author & commit.

→ Filter in log:-

- `git log --since = 2. days` → last 2 days entries.
→ 2. weeks → 2. years etc.

Unstaging & unmodifying files in Git:-

- ① Unstaging `git restore --staged <file>`

- ② Unmodifying (get back to last commit)

`git checkout -- <file>`

`git checkout -F` → Take everything back to last commit & undo all the changes.

To upload all the files to the online repository.

First creat a blank repo. On Github then

Step: 1) `git remote add origin <SSH repo>`

2) `git branch -M main`

3) `git push -u origin main.`

- Setting Alias in git: (Use to make short form of the command.)

`git config --global alias.<alie name> <Command>`

Eg: `git config --global alias.st status`

- Creating & Switching branches in git: → Make sure to commit before changing branch.

① To make new branch

`git checkout -b <branch names>`

② To get back to master branch

`git checkout <master/main name>`

③ To get back to the branch

`git checkout <sidebranch name>`

→ few other branch command:

`git branch` → To get the list of available branches.

`git branch -v` → It gives last commits in all the branches.

`git branch -d <branch names>` → To delete any branch

NOTE `git merge <branch names>` → To merge the in the current directory.

- Pushing git branches to remote repository:

① `git push origin <branch name>` → To publish new branch in repository.

② `git push origin <local branch name> : <newname>` → To publish with new name.

③ `git push` → To update the file in the existing repository.

NOTE `git branch --merged`
→ give names of already merged

`git branch --no-merged`
→ give name of not merged branch.