Git hub & Git

→ Git Bash command -

- pwd - present working directories
- cd - change directory
- cd desktop - change to desktop
- cd /c - change directory to c drive

} These are linux based Command.

→ First create a new folder and open git bash from there using Right click.
Or
- we can Run command line in git bash to create directories.
  - mkdir <directory Name>

→ How to view all directories in Git bash.
- ls → to view all directories.
- ls -a → to view all the hidden folders.

→ If we want to create directories then following command need to run.
- mkdir <folder name> - if it is two diff. name then two fodd. created.
- mkdir "<folder name>" - only one folder will be created.

→ we can remove directory by following command.
- rmdir "<name of fol>" - will remove the dir.

→ Commands for Registerar your user name.
- git config --global user.name "Amit"

→ Command for Register your email.
- git config --global user.email "yamit7188@gmail.com"

→ To view all the available list.
- git config --list

Git command

→ Git status – to see git status

→ git init – to create git repository and it will auto create .git file.

→ git add -a – all files on now to stage

→ git commit -m "massege" – to create a snapshot.

→ git log – last activity can be view

→ git add <file.name> – put particular file in staging area.

→ rm -rf .git – delete .git repository

→ git done <URL> – to create a directory

→ gitignore

- to create a file with the help of touch command
  - touch <error.log>
- If we want to ignore any file the
  Stp-1 – Create another file with the help of
  gi touch .gitignore
  Stp 2 – and put file name inside .gitignore which you want to ignore.

→ if there are multiple file wit same extension and you want to ignore them all then follow below steps:
  - put *.log <or any extension> inside .gitignore file.

→ git diff - It compare the file in staging area and the file is in modified area.
or
It compare staging area wit working directory

→ git diff --staged - It compare previous Commit with it present staging area.

→ Skipping the staging area -

- git commit -a -m "Direct Commit" — just a comments
  ↳ This command put all tracked file in staged and make a Commit
  but if there is an utracked file then it will skipp it.

→ Rename and move

→ git rm <filenam.txt>
[This command will remove the file and put it into stage.

→ git mv <filenam.txt> <nawnam.tx>
└ This command will rename the file and put it into stage.

→ git rm --cached <file.name>
↳ This command will untrack the file
It mean if there is a file on track it will remove that file from track and put it on untracked.

→ Git log: Viewing & changing Commit

→ git log -p
  ↳ activity with diff

→ git log -p -2 (or any number)
  ↳ It will show activities wit number of lines.

→ git log --stat
  ↳ It is same as git log -p but in short.

→ git log --pretty=oneline
  ↳ log in oneline

→ git log --pretty=short
  ↳ In short

→ git log --pretty=full
  ↳ full details.

→ git log --since=2.day/week/month/years
  ↳ use as filter.

→ git log pretty=format:"%h -- %an"
  ↳ for details visit
    ↳ git scm useful option for git log format

→ git commit --amend
  ↳ This command will merge the change with its previous content.

↳ [vim editor

press i for insert

Press [esc] → [:] → [q] → [w] → [↵]

→ Unstagging and unmodifying.

→ git restore --staged <file.name>

↳ unStaging file.

→ git checkout -- <file.name>

↳ unmodify the file.

↳ match with its previous commit.

→ git checkout -f

↳ Complete file matched with its previous commits (use only if there are large number of file).

→ git Remote:

→ git remote add origin < url >

↳ This connect wit git hub and put Repository name origin.

→ git remote -v

↳ This will show you push and pull URL

→ git push -u origin master

↳ This command will push repository to git hub.

Alias:-

→ git config --global alias.st Status

↳ we can use git st instead of git status.

→ git config --global alias.unstage "restore --staged"

Creating Branches with git:-

→ git checkout -b <branch name>

↳ divert to new branch and it will create new branch.

→ git checkout master

↳ change to master branch.

→ git branch

↳ To view all branches.

→ git checkout <branch name>

↳ Switch to any branch.

→ git merge <branch name>

↳ This command will merge with master Branch

→ git branch --merged

↳ This will show merged branch

→ git branch --no-merged

↳ This will show unmerged branch

→ git branch -d <branch name>

↳ This will delete branch

→ **branching workflow :-**

→ **push branches to git hub.**

~~git branch~~

Step 1 - on git Switch to that branch which you want to push.

& → git check out <branchname>

Step 2:- Now use command to push it

git push origin <branch name>

→ If you want to rename branch while pushing then:

→ git push origin <branch name> : <Rename branch>

Note: If you want to merge any branch to master first go to master branch then run command

git merge <branch name>

→ If you want to delete any branches from origin then:

↳ git push -d origin <branch name>