(begin 🡪 stage 🡪commit(photo) )

After making folder you should do **git init**

**After making change you should do**

Git add file.txt (to bring it in stage)

Git commit -m “modify the file” =commit=

1. How to add files
   1. git add .
   2. git add abc.txt xyz.txt
   3. git add -A
2. How to check status
   1. Git status
   2. Git status -s short for precise to file only
   3. Git status -u for untract files only
   4. Git status -b for branch only
3. How to commit
   1. Git commit -m “YOUr commit”
4. To back to previous commit of that particular file
   1. Git checkout (FileName)
5. To back to previous commit of all files
   1. Git checkout -f
6. To check the log
   1. Git log
7. To check the last N log and view the change it made
   1. Git log –p – (N-> 1,2,3,4,5,6,)
8. To check the last edit that is not yet in stage. It compare stage with working tree
   1. Git diff
9. To compare new stage area with last commit
   1. Git diff --staged
10. To get list of files
    1. Ls
11. To create a file
    1. touch abc.html
12. to remove the file
    1. delete file from folder
       1. git rm file.txt
    2. to remove file from track list not from the folder
       1. git rm cached file.txt
13. to make a new branch
    1. git branch (branchname)
14. to move to new branch
    1. git checkout branchname
15. advance version of making branch and directly checkout into the branch
    1. git checkout -b (branchName)
16. to merge the branches
    1. go to branch in which add features
       1. git merge (branchNmae)
    2. to add specific file from one branch to another branch
       1. git checkout (BranchName) (filename)
17. to check difference in branch
    1. git diff (branchNameofOtherBranch)
       1. git diff (belowOption) branchName
       2. **--name-only** Shows only the names of the files that are different between the branched
       3. **--stat** Provides a summary of the changes, including the number of files changed and the number of insertions/deletions
       4. **--color-words** Highlights the differences within lines using color.
       5. **-w** Ignores whitespace changes.
18. to delete the branch
    1. git branch -d branch\_name
    2. if you have not merge it above command does not work
       1. git branch -D branch\_name
19. to connect with github
    1. git remote add (origin 🡨 NAME ->) httpsxxxxxxxxxxxxxxxxxx
20. to check remote respositreis
    1. git remote
21. to get remote link
    1. git remote -v
22. to delete the remote
    1. first check remote by
       1. git remote
    2. then
       1. git remote remove {name}
23. to push from local to github
    1. git push -u origin (<- remote name that we have set above) master (Name of our local branch agar kisi or local branch sy krna hai push tu uska name)
24. to pull from github to local
    * 1. git pull origin master (agr kisi or branch sy pull krna tu master ko branch k name sy replace)
25. Agr only one file pull krni hai tu
    1. Git checkout origin master file name
26. Agr koi respositery ko download krna hai tu
    1. Git clone httpxxxxxxxxxxx

first of all create a folder in c

then open cmd in that folder path

in cmd type command (git inint) it make it stage

to check status use (git status) in cmd

\*created file in that folder not tracked by the git until you add this file in git through cmd commands\*

git add filename.txt (YA) git add filename1.txt filename2.txt filename3.txt (YA) git add . /add k bd space hai/for all files in the folder/ YA / git add -A (to save all files)

if you stage a wrong file or not to stage file do (git reset ) in cmd to remove file from stage

(git reset --hard) it takeoff the file stage and also remove the modification you apply on the file

git status

git commit {it will open vim command press I then write initial commit then press Esc then :wq enter you will get come to bash cmd}

git commit -m"file names and animals added"

to clean the terminal “clear”

git ls-files (to get list of files)

git log /to view author and created date changes time and date/

edit in file then again commit it with

git commit -m"we edit in file \*your message is\*"

if you want that git not track some file in folder then you have to make a file having no name but .gitignore header inside it mention the file name which you want to ignore if folder then simple mention the folder name

git branch /\*Show list of branches\*/

git branch -v /\*this commannd is use to show log of branches\*/

git branch hello /\*this command is use to create new branch\*/

git checkout \_\_xyz\_\_ /\*this command is use to go on the branch\*/

git log master..dev /\*jo item dev ko pass hain lekin master k pass nai woh show hongay\*/

git log dev..master /\*jo item master ko pass hain lekin dev k pass nai woh show hongay\*/

/\*Agar alag alag branch ny same file mei same line mei change kia tu conflict ajai ga jo terminal mei solve krna muskil hai magar ui mei easy hai \*/

git merge ------ /\*space mei branch ka naam ai ga \*/

Agar apko code commit kiye bagir branch change krni hotu stash krdo satash(temporarily save changes)

git stash /\*to save temporarily\*/

git stash save <name>/\*to save stash with given name\*/

git stash list /\*name of all stash\*/

git stash pop /\*it apply the last stash\*/

/\*same file per ek sy ziyada stash tu hojai ga lekin ek ko apply krnay k bd add or commit bhi krna hai phir dosra stash apply hojai ga\*/

/\*alag alag file mei bhut sara stash bn sktay hain\*/

git stash apply <name>/\*we apply stash that we save with name and it also seen in log history\*/

git stash apply stash@{} /\*by giving name we apply out required stash\*/or stash no

git stash clear /\*clear list of stash\*/

--------------------------------------------github--------------------------------------------------------

1) git clone url

git push origin master /\*it also be written as git push only\*/

git fetch (it bring changes on your machine from server but it not change the file until merge is called)

git merge (it merge the file by commit changes init bring by the fetch)

git pull (it direactly fetch and merge both in single command)

git remote -v(/\*show list your folder connected with\*/)

-----------------------------------From Local to GitHUb-------------------------------------------------------------------------

first make folder (git init) then create repo on github

then git remote add origin hhttp:xxxxx

git remote

git remote -v(/\*show list your folder connected with\*/)

for push :git push -u origin master

for pull :git pull origin master

git remote show origin /\*To get url for pull and push\*/