ghp\_gDXRDqA0f2A4y0hxMhAkOKVzAAbxei1TpffR

**GIT COMMANDS(Kmgreddy@1994,** **kandadimurali**

1. TO SEE THE DETAILS OF PRESENT WORKING DIRECTORY ; pwd
2. To see the details of folder list ; ls
3. T see detais in folder ; cat file name

4. To clear the github command page ; clear or ctrl+l

1. To check git is installed or not : git –version
2. To initialize git repo ; git init
3. To see hidden files : ls –la
4. To full directory with details : ls –lrht
5. To see files on workspace ; git status {red indicate work space; green indicates index stage}
6. move files fom workspace to indexstage : git add filename or .(dot) or \* or –a
7. To check status : **git status**
8. move files fom index to local repository: git commit –m “explanation for files” (filename only for particular folder) we need permission to use this command for that wee need to submit our mail and name through folloing instruction and example ;1 git config --global user.email kmgreddy143@gmail.com : 2 git config --global user.name "muralidharreddy"
9. check files commited or not : git log
10. to create new directory or folder : mkdir name
11. to enter the directory : cd name
12. to create new fies : touch name
13. to see only one commit details : git show commit number (commit no is available in commit log)
14. to see git user details : git config –list
15. to switch the git user :
16. to move file from local repository to index stage : git reset --soft commit id -1
17. to move file from local repository to work space : git reset –mixed commit id -1
18. to move file from index to work space : git reset head filename
19. enter for editing file data : vim file name
20. edit file data : press I and change editing requires
21. after completing edit ; 1)press esc :w for saving edit data 2)press :wq for save and exit 3) press :q! for exit without saving
22. delete files : rm –rf filename ( delete all command is rm –rf \*)
23. delete multiples : rm –rf file1 file2
24. delete directory/folder : rm –rf directory name
25. vvv
26. vvv
27. to download / clone data from git account : git clone <url from git account>
28. enter into clone directory
29. mention require data and do git add and commit
30. then push the updated data
31. to receive data after from github exclude of previous data command : git pull
32. if git push is not accepting data use command of git pull –rebase ( this command will work to receive changed data from files)
33. to delete file in central from gitbash
34. first delete file
35. second add files
36. third commit
37. and use push then files will be deleted
38. to give permission for others to push the file to repositorty
39. open repository setting and select collaborate in access
40. add people by mail or username of github account
41. vvv
42. to see log in shortlist: git log –-oneline
43. to see top require lines : git log –-oneline –n
44. to find with author name : git log –-authorname
45. to find with name line : git log –-authorname –n
46. to see a word in message: git log –-grep name
47. to se log with data wise: git log –-(befor/since) date
48. to see log with data : git log –n
49. vvv
50. vvv
51. to see branch list : git branch
52. to add branch : git branch <name of require branch>
53. to shift branch: git checkout <name of require branch>
54. to create branch and enter into branch in single command: git checkout –b <name of require branch>
55. to push new branch to central : git push origin <name of require branch> branch should be main
56. whenever you created new branch all data will be copied from which branch you created
57. to delete branch: git branch –d/D <name of require branch>
58. to delete branch in central: git push origin –d <name of require branch>
59. vvv
60. to copy data from one branch to another branch: git merge <name of require branch>
61. to see sll branches: git branch –a/r
62. to merge only specific file use command of : git cherry-pick cid no
63. To merge two different files data into one file data : git merge master
64. Create shortcut for command : git config –global alias.s “status”

**Amend:**

1. To change commit message of last commit: git commit --amend –m <label message>
2. To commit extra file to last commit :git commit --amend –m <label name>

**Tags:**

1. To see tags: git tag
2. To add tags: git tag <tag name>
3. To see tag detais: git checkout <tag name>
4. To push tags to central: git push tags
5. To push required tag to central: git push origin <tag name>
6. To tag previous data: git tag <tagname> cid no
7. To delete tag : git tag –d <tagname>
8. To delete tagin central : git push origin –d <tagname>

**Stash :**

1. To move file to slash: git stash save <label>
2. To see stag files: git stash list
3. To receive only one file : git stash apply stash@1/2
4. To bring back file from stash

Git stash apply (apply means copy and paste)

Git stash pop (pop means cut and paste)

Git stash drop (drop means delete in stash)

1. To see details of stash folder : git stash show

**Merge:**

1. It is used for merging data for two branches’ if we use merge cid data will be in correct order and create with one extra cid: git merge <dir name>

**Rebase:**

1. Rebase is used for merging data for two branches, if we use rebase cid data will change the order and no extra cid will be create: git rebase <dir name>

**Pull=feth+merge**

1. If you use pull command data will be tranfered from central repo to origin and origin to local repo directly: git pull

**Fetch;**

1. If you use only fetch command the data will be transferred to origin only: git fetch
2. You want to move origin to local repo: git merge origin

**Revert:**

1. If you want to revert of commit : git revert cid

**Gitignore:**

1. If you add any file to .gitignore that file should be hidden and if you dane any work in that file the work and status of that file will not be able to record of that file