1. Git init 🡪 to initialize file in selected path/area or folder.

2. Git config --global user.name “Name” 🡪 to create user name.

3. Git Config --global user.email “email id” 🡪 to create user email.

4. Git Config --list 🡪 to check user configuration details.

5. Touch (File Name) 🡪 to create one file; more files add file name space file name.

6. Git status 🡪 to check file status (red : untracked files are in local system)

7. Git add 🡪 (file name for 1 file ) (. & \* & -A for all files) to add in stagining / index area (green :it changes to commit).

8. Git commit –m “label name or file description” 🡪 to move local repository.

9. Git commit –m “label name in double quotation mark only” then file name 🡪 for single file to commit.

10. Git log 🡪 to check files status in local repository area, committed or not.

11. Git clone 🡪 copy link from github clone link, in required repository to download Ream me.md file for access to push. (it will download total repository).

12. Git push 🡪 Send all local repository data to github (Central repository ) to push we have to enter inside the repository folder using cd ..

13. Git show (copy id ex. 97df354df) 🡪 to see inside committed files list.

14. cd .. 🡪 one step back ◘ cd 🡪 Home ♦ cd 🡪 double time“C”user path♦

Cd /c 🡪 enter C drive ♦ Ls 🡪 list files♣ Ls-La 🡪 list with hidden files.

Ls-a 🡪 list of all files with hidden. **/** 🡪 Enter or Root.

15. Control panel 🡪 user accounts 🡪 credential manager 🡪 windows credentials 🡪 to change/delete github account details in local repository for central repository.

16. Git log --oneline 🡪 it shows first 7 characteristics and label name in oneline.

17. Git log ̶ n ( n=number) 🡪 shows from newly added files.

18. Git log --oneline -n 🡪 it shows only latest created log in one-line.

19. Git log –author=author name. (Ex-Arjunsdevops) 🡪 to check author log.

20. Git log --since=year-month-date 🡪 it will show since nundi the log to today.

21. Git log --until=year-month-date 🡪 it will show all logs until varaku the given date.

22. Git log --oneline --decorate 🡪 it shows all listed files.

23. Git reset --soft idnumber 🡪 to reset the file from local repository to index/stagining area.

24. Git reset head then file name for 1 file or enter dot / \* for all 🡪 to move files from index/stagining area to local system.

25. Git reset --mixed and enter commit id 🡪 to reset from local repository to local system directly without Index/stagining area.

26. Git branch 🡪 to see the list of branches.

27. Git branch *branch-2* 🡪 to create new branch (Ex.. - branch-2)

28. Git checkout required branch name 🡪 to change that branch.

29. Git hub collaboration access 🡪 enter repository – settings – collaborators – add collaborator then can assess git hub for modify.

30. Git pull 🡪 it pull all updated files in github repository.

31. Git push origin - - d branch name 🡪 to delete remote branch in git hub.

32. Git branch -d branch name 🡪 to delete branch (before you delete habe to checkout the branch first).

33. Git branch -**D**  branch name 🡪 to delete forcefully.

34. Git merge branch name 🡪 to merge other branch files in required branch. To merge have to enter in target branch only.

35. Git rm –r file name 🡪 to remove a file or folder and commit again to remove permanently.

36. Git rm –rf 🡪 to remove forcefully file or folder.

37. Git push origin masterbranch/branchname 🡪 to push github repository anyway.

38. Git pull origin branch name 🡪 to pull from github repository anyway.

39. Git log - - summary 🡪 view changes detailed.

40. Conflict error 🡪 because 2 branches have same name with different code/matter that file then it will show conflict error.

41. Git merge - - abort 🡪 to return back from the conflict error (Temporary).

42.Stash memory 🡪 It is only temporary storage memory. We can only move all our staging /index area files to save temporary and can roll back it.

43.Git stash list 🡪 to check stash list.

44.Git stash save “label name” 🡪 Move to stash all files in index/staging area.

45. Git stash show –p stash memory name 🡪 To see stash file details.

46. Git stash pop 🡪 to send back to index area (it only send stash 0 file)

47. Git stash pop stash@{1} or stash memory name 🡪 to move required file.

48. Git stash apply 🡪 🡪 Copy to index area (it only Copy stash 0 file)

49. Git stash apply stash@{9} or stash memory name 🡪 to copy required file.

50. Git stash drop🡪To drop 0 *&* Git stash drop stash@{7}🡪to drop requires.

51. Alias Command 🡪 it’s another name to a command what we create like a shortcut or short name.

52. Git config - - global alias.L “log” 🡪 L is a short cut to log command **ex**. Git l is git log.

53. Git config - - global - - unset alias.l or user.name or any config to remove.

54. Git commit -am “label name” 🡪 moved files or reset files or edited files only can move from workspace to local repository area without index area using this command. \*not for new files.

55. vi .gitignore 🡪 vi to create and edit data in created file & .gitignore to hide the names what we entered inside the document as a data. Suppose we create 2 files using same name inside the .gitignore data names it will create but I wont appear in your log. If you removed the name in .gitignore it will appear. \* .gitignore is only to hide documents.

56.