1) To set name and email

$ git config --global user.email "rajdevaliya169@gmail.com"

$ git config --global user.name "Raj"

2) To watch the name and email

$ git config user.email

$ git config user.name

3) git status

4) git init --- new hidden .git folder banse ---

5) git add –-a or .

6) git commit -m “message”

7) git log

8) rm -rf .git --- to remove init ----

9) git clone

[https://github.com/tensorflow/tensorflow.git Rename](https://github.com/tensorflow/tensorflow.git Rename10)

[10](https://github.com/tensorflow/tensorflow.git Rename10)) rm -rf tensorflow/ --- to delete ---

11) rmdir /s tensorflow/ --- to delete ----

12) touch error.log ---- to make empty file----

13) touch .gitignore --- write file names in it that you don’t want to track

14) git diff ----comparition between staging file and modified file----

15) git diff –staged – in staging area – to check last staging what we did , only works in staging area---

16) git commit -a -m “message” ---- bypassing staging area ----

17) git rm filename ----to remove file ----

18) git mv name.txt rename.txt ---- to rename file name ---

19) if file is on tracking before moving into .gitignore, git will still track file despite in .gitignore

20) git rm –cached filename --- to untrack from above situation ----

21) git log -p ---- what is changed in file with commit ---

22) git log -p -3 ----to check first commits only ----

23) git log –pretty=oneline ---to see commits in oneline----

24) git log –stat ----to see commits in short ---

25) $ git log --pretty=short

26) $ git log --pretty=full

27) $ git log --since=3.days ---to filter log as per time---

28) $ git log --pretty=format: “%h -- %ln”

29) git restore filename ---remove from staging area, changes will be removed---

30) git checkout –-filename ---- to get previous stage of file ----

31) git checkout -f --- works same as above but for all file---

>>>>>>>Remote <<<<<<<<

32) git remote add origin <https://github.com/rajdevaliya/practice_repo.git>

33) git remote -v ---- to check repository available----

34) git remote

35) git remote rm origin(name) ----to remove remote connection----

36) ls -al ~/.ssh ----to check existing ssh key available ---

37) ls -a ----to check all files with hidden ones---

38) ssh-keygen -t ed25519 -C "rajdevaliya169@gmail.com" –to create ssh key---

39) rm -f ~/.ssh ---to remove ssh keys if stored in pwd ---

40) eval "$(ssh-agent -s)" --- to check ssh agent is working or not ---

41) ssh-add ~/.ssh/id\_ed25519 --- to add ssh key in agent ----

42) clip < ~/.ssh/id\_ed25519.pub ---- to copy ssh key ----

43) tail < ~/.ssh/id\_ed25519.pub --- to see that key ---

44) git config --global alias.st 'status' ---- to make command short as per out convenience

45) git checkout -b develop ---to make branch of develop---

46) git checkout main --- to switch from branch—

47) git branch ---to see available branch---

48) git push -d origin develop --to delete from remote---

49) <https://gist.github.com/aprilmintacpineda/f101bf5fd34f1e6664497cf4b9b9345f>

To generate multiple ssh keys for multiple github account

50) <https://www.freecodecamp.org/news/how-to-make-your-first-pull-request-on-github-3/>

Pull requests