**#gives name so that everyone can know who did the commit**git config –global user.name “Sanidhya”

git config –global user.email “[guptasanidhya6@gmail.com](mailto:guptasanidhya6@gmail.com)”

**# to list out config details**

Git config –list

**# to change editor in linux only**

Git config –global core.editor emacs

Git config –global core.editor vim

**#to get details of your repo**

Get status

**# to initialize a new git**

Git init

**#to add files in staging area (for commit)**

**#For all files**

Git add -- a

# to commit files

Git commit –m “add a message inside string”

#to see all your commits

Git log

#to remove the .git folder

Rm –rf .git

#to copy our own repo or some one else

Git clone “link of clone without quotes”

# linux command to create a new file

Touch “filename with extension

#create a git ignore file

Touch .gitignore

#to compare two filles staged and unstaged file

Git diff

# to compare two files between staged and commit area

Git diff -- staged

**# if you want to commit only tracked files which may be modified and don’t want to commit untracked file**

Git commit –a –m “Direct commit”

#to remove file from git

#this command also add it to staged layer

Git rm “file name with extension without quotes”

# to rename a file use

Git mv old\_filename.ex new\_filename.ex

#if we have a file which is getting tracked from the past and we want to add it in git ignore so

We have to add file first in git ignore and then run command

Git rm --cached “filename”

**Git View**

Git log #to view all git commits

Git log –p #also gives what we added and removed in git files

Git log –p {1 to n number} # will gives you last n commits

Git log - - stat # gives you basic description detail of all commit

Git log - - pretty=oneline #gives you comment lines of all git commits

Git log - - pretty=short #gives you short description

Git log - - pretty=full #gives you some extra description

Git log - - since=2.days # gives you details of last two months

You can also format the gitlog you can find it in git scm the formatting options

Examples https://git-scm.com/docs/pretty-formats

Git log - -pretty=format:”%h -- % ae” #gives you hash and email

Git log - -pretty=format:”%h -- % an” #gives you hash and name

To edit the last commit

Git commit - - amend #edit in vim editor

To unstage a file from staging

Git restore - - staged filename

If I changed some file and lost the important data

We can get the last commit file data using

Git checkout - - filename

Git checkout –f #undo the previous commit

#working with remote

Git remote add origin “link” #origin is a name by default we can change it

Git remote #shows where the files are going

Git remote –v #gives fetch and push link details

If you want to apply alias (short hand for commands)

Git config - - global alias.st ‘status’ #change the status to st

**Branching**

Git checkout –b branch\_name #to create a new branch and switch to it

Git checkout branchname # for just switching the branch

Git branch #it will show all branches

Git branch –v “to get branches with commit and hash”

Git branch - -merged #to get the merged branch

Git branch - -no-merged master #to get the non merged branch

Git branch –d “branchname” ***#to delete a branch*** first it will give error that the branch is not merged if you use capital D then it will not give error

If you are in different branch and want to push it to services

Git push origin branchname #remember you should be in same branch

git push origin local\_branch\_name:\_service\_branch\_name ***#you can change the name on services it will act as a new branch as get tracked***

**If you want to delete the branches from the github use**

Git push –d origin branch\_name

Merging

Do merging after commits only

Git merge branchname #first check you should be in master or main branch in which you want to change

***If there is any conflict during merge***

Choose either the current change or incoming change

Git status

Git add .

Git commit #you can write merge inside editor also