Hi hello

git init => intiates/makes git folder/repository

git status => inform about git status

rm -rf .git => deletes git folder/repository

git add --a => add all files tracking to git. unmodified stage

git commit -m “comment” => moves files to staged stage

git pwd => in which folder you are (present working directory)

git ls => list content of folder in which you are

git cd => change directory

git clone [link] => pull the directory of that particular link

touch [name].log => for making a new log file with particular name

in directory

touch .gitignore => makes a file in which if name of any file is indexed then that file is ignored by git status

.GITIGNORE folder entries:

\*.log - git ignores all files of .log type

dir/ - git ignores dir named folder

/dir/ - git ignores outside dir named folder

static/dir - git ignores dir named folder inside static folder, but other content of static folder won’t be ignored.

git diff => compare working directory and staging area

git diff --staged => compare your previous commit to your current staging area

git log => Shows all commited files

git commit -m -a “comment” => Skip staging area

git rm demo.txt => revomes demo.txt file and directly moves file to staging area

git mv first.txt second.txt => rename first.txt to second .txt

git rm --cached [file name] => make untracked file tracked

git log => used for exploring repository’s history

q => to quit log repository

git log -p => to know about changes in log

git log -p 3 => shows only 3 commits

git log --stat => tells in short about all changes in a commits

git log --pretty=oneline =>shows all commits in one line

git log --pretty=short/full =>shows all commits in short/full

git log --since=2.days(or months/years) => shows all commits within 2 days

git log --pretty=format:”%h -- %ae” => for retrieving infromation with own formatting type

git restore --staged [file name] => restore file from staged to unstaged area

git checkout -- [file name] => unmodify file to previous commit position

git checkout -f => take back the all repository’s file to previous commit state