git --version

git init (Converts folder into git repo)

git clone [URL] (clones the project at the current location)

git status (current branch, compares current branch with remote branch, commit status)

git branch (Shows current branch with all other (locals only))

git branch -r (Shows all remote branches)

git branch -a (Shows all local and remote branches)

git branch [branchName] (created a new branch)

git checkout [branchName] (switch to a new branch)

git checkout –b [branchName] (creates and switch to new branch)

git branch –d [branchName] (Deletes branch)

git merge [branchName] (merges [branchName](In which changes were done) with current branch(Master)).

git diff [file name] (Difference between file of current location repo and remote repo )

git add . / git add [file name] (Add file to staging area)

git commit –m “changes done” (commit through which we can track and identify changes done)

git log (Shows all the logs for all commits done one this repo)

git log -5 (shows latest 5 logs)

git show [commitHash] (shows the changes done in this commit)

git reset --hard HEAD^ / hashCode of commit (removes commit , removes it from staging area, undo the code changes done from the file)

git reset--soft [HEAD^ / hashCode of commit] (removes commit only, staging area and code changes are not disturbed)

git pull (merges commits from remote branch to local current branch)

git fetch (to check if there are any changes between remote and local branch)

git remote add origin [remote url] (add remote urls with origin as ref)

git remote –v (shows remote urls for fetch and push(only if remote urls are added))

git push –u origin [branchName] (push changes current branch changes to origin)

git restore --staged [fileName] (removes file from staging area)

git restore [fileName] (undo the code changes done in the file)

git commit --amend (Editing the commit done)

touch .gitignore (creates file .gitignore, file name inside this will not be tracked in git.)