* Git init (to initialize git, .git gets added), git add presentflename (. Will add all the present files to staging area), git commit -m (to commit the file to local repository)
* Git commit -am “” (Express command, after file is initialized and modified. Adding and commitiong for the second time).
* Empty directory cannt be added to git.
* rm -rf .git (to remove git functionality, execute froom master branch).
* Rm -rf \* (to remove all the files).
* Git ignore file (vim .gitignore): all the files mentioned in this git ignore file will not be effected by git init, or git commands.
* Git log : To check history of all commits, Git log –oneline (to see precise commit details). Git log –graph –oneline –pretty (shows the history in graph format).
* Git status (to check the status of the files)
* Git checkout commit# will take you to the commit
* Git reset –hard commitid will remove the details before that commit.
* Git reset –soft commitid will remove the commits and file to the unstaged area, you need to add and commit again
* Git reset –mixed commitid will remove the commits and take the file to the untracked area.
* Git commit –amend -m “message” (amend the last commits, that is merging the last two commits into one.
* Git branch brancname : creates a branchname
* Git checkout branchname : takes you to branch
* Git checkout -b branchname: express way to create and moving to a new branch.
* Git merge branchname: to merge the two branches
* Git cherry-pick to pick only single commit from the branch and merge it into the master
* Git rebase -i HEAD~commitnumber (rebase helps in shuffling the commits, replace pick with squash number to combine two commits, or rearrange the commits in your preferred order).
* Git clean, -n is for starting the removal process of files that are not for git, -f is to remove the files, -I will show you the options to select by pattern (like mention .txt), or by numbers, or by asking for each file.
* git reflog: This command will show all the files copy in temporary state. Like if you remove the file during hard reset we can retrieve the file from reflog. The reflog log will show all the files both that are removed and present with their commit it. We can checkout to the commit id we are looking for, commit it, and then create a new branch,