* ***git init*** -> To initialize a repository.
* ***git status*** -> To check the status of a directory or to track files.
* ***git add <file name>*** -> To add a particular file.
* ***git restore <file name>*** -> To remove all the new changes made in a tracked file and restore the previous version. After running the remove command, the modifications will be removed. After this, on running status command, there will be no need to add the file to staging area again because the previous version is already getting tracked.
* ***git add - - a*** -> To add all the untracked files to the staging area.
* ***git rm - - cached <file name>*** -> To remove file from staging area. This can be used when has not been committed even once otherwise use restore - -staged.
* ***git commit -m “msg to be added with the commit”*** -> To commit the files to the local repository.
* ***rm -rf .git*** -> to make a directory a normal one and stop tracking the file. It will delete the .git folder.
* ***git diff*** ->to get the difference between working directory and staging area. Same file should be present in both the areas with modified content in the working directory, only then this will work. If a new file/directory which has not been tracked is present in wd then git diff will not mention it.
* ***git diff - - staged*** -> to get the difference between staging area and the last commit.
* ***git commit -a -m “committing a modified file”*** -> to commit a modified file which is getting tracked in the local repository without using git add explicitly. This won’t work for untracked files.
* On deleting a file, the change has to be added to staging area and should also be committed. After adding it, it can be brought back to the directory by using the git restore command.
* On renaming a file, git will consider it deletion and addition of new file when you run status command. After adding it to staging area, it will know that file has been renamed and then you can use the commit command to commit it. You can use restore command to undo the renaming of the file.
* ***git rm <file name>*** -> Use this to delete a file and commit the change without adding the update to the staging area. This worked after committing the file.
* ***git mv <old file name> <New file name>*** -> Use this to rename a file and commit the change without adding the update to the staging area. mv stands for move.
* If a tracked file is included in .gitignore then it should be removed from tracking area by using ***git rm - - cached <filename>***
* ***git log*** -> Used to get all the commit history.
* ***git log*** ***- -oneline*** -> Used to get partial SHA value and commit message in one line.
* ***git log -p*** -> Used to get the commit history and also the changes made in the commit.
* ***git log -p -2*** -> Used to get the commit history and the changes made in them in the last 2 commits. 2 can be changed to any other integer value too.
* ***git --stat*** -> Used to get changes in the form of total number of insertions and deletions in a particular file.
* ***git log --pretty=short*** -> Used to get commit history in short with only author details.
* ***git log - -pretty=long*** -> Used to get commit history with the details of the author and the person who made the commit.
* ***git log - -pretty=oneline*** -> Used to get full SHA value and commit message in one line.
* ***git log - -pretty=format:”%h -- %ae”*** -> To get commit history in the format specified between the double quotes. Here abbreviated SHA value with author’s email will be displayed.

For example: 40f18c470 -> charlesr.harris@gmail.com

* ***git commit --amend*** -> To merge one commit with another commit. After this step Vim editor will open. Press I to add your message and then press Escape key. Type :wq to quit. Your commit will be merged with the previous commit and don’t forget to stage the changes before using git commit - -amend.
* ***git restore --staged <file name>*** -> Use this to un-stage a file from the staging area. It should have been committed previously.
* ***git checkout --<filename>*** -> To undo changes in the working directory that have not been staged yet.
* ***git checkout -f*** -> To bring a file to its last committed state when the change is just in the working directory or in the staging area. Works for both, working directory and staging area.
* ***Refer to this for generating an ssh key and adding it:*** [***SSHGeneration***](https://docs.github.com/en/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agen)