**GIT COMMANDS AND IT’S USAGE**

**To Initialize a directory as a git repository:**

git init

**To add all files to the staging area:**

git add .

**To stop tracking a folder using git:**

rm -rf .git

**To commit the staging area:**

git commit -m “<message>”

**To get the difference between the current working directory with the staging area**

git diff

**To get the difference between the current working directory with the latest commit**

git diff –staged

**To skip the staging area and commit directly:**

git add -a -m “<your message>”

**To ignore the files:**

Create a file .gitignore

Add file names or directory names or the extension name

For example: .docx, .pdf, /static (directory name), /static/ (to ignore only the particular directory and not the sub directory)

**To rename the files:**

git mv <old file name> <new file name>

**To delete the files:**

git rm <file name>

**To clone existing git repository:**

git clone <repository url> <file name> (this is optional If you want the new repository to clone it in a new folder with a name pro)

**To log all the commits:**

git log

**To log all the commits with their differences in all the commits:**

git log –p

**To log all the commit in a single line :**

git log --pretty=oneline

**To log all the commits with less information:**

git log --pretty=short

**To log all the commits within specific period of time:**

git log --since=2.days(you can use different time period such as days, months or years)

**To log all the commits with different formats:**

Visit google search = git scm git log useful commands to dive deeper into git log commands

For example, log all the commits with abbreviated hash and the author name, such as “nkjdsfbs -- jaypatel”, to get such results use the following command:

git log --pretty=format:”%h -- %an”

**To unstage files added in staging area:**

git restore --staged <file name>

**To restore a file to its last commit:**

git checkout -- <filename>