**Command Prompt:**

- **ls** or **list**  (*to list all the files in the current directory*)

- **cd <location>** (*to change the directory you are currently in*)

- **cd ..** (*to go back to previous directory*)

- **touch <file\_name>** (*to create a new file with any extension*)

**Vim Editor Tutorial:**

- How to write something in VIM Editor?

ANS: press " **i** " key on keyboard.

- How to exit from VIM Editor?

ANS: 1st press "**Esc**" key

then write= " **:wq** " and press "enter".

**Git Commands:**

- **git config --global user.email "email\_ID"** (*to change the email id in git*)

- **git config --global user.name "name"** (*to change the Username in git*)

-**git config --global --edit** (*to change both Username & Email at the same time*)

- **git --help** (*for command help*)

- **git init** (*write in the directory where the code is saved*)

- **git status** (*to check the status of the files*)

- **git status -s** (*to check short status*)

- **git add <file\_name>** (*to add the code file in the "****staging area****"*)

- **git add -A** (*to add all the files in the "****Staging area****"*)

- **git commit -m "message"** (*to* ***commit*** *the file which is in the* ***staging area***)

- **git commit -a -m "message"** (*commit* ***all*** *the file* ***without*** *taking them to the* ***staging area***) (*avoid using this command*)

- **git checkout** (*to get the previous commit back*)

- **git checkout -f** (*to get the pervious commit back of all the files*)

- **git log** (*to get all information about all the commits done in the past*)

- **git diff** (*it will* ***compare*** *the working tree or say* ***current working progress*** *to the* ***staging area*** *project*)

- **git diff --staged** (*it will compare the* ***present*** *"staged area" to the "last commit"*)

- **git rm -f <file\_name>** (*this command will remove a file from the* ***staged area*** *as well as from the* ***original directory*** *also*)

- **git rm --cached <file\_name>** (*this command will remove the file from* ***only*** *the* ***staging area****)*

-**git rm --cached \*** *(this command will remove* ***all*** *file from* ***staging area****)*

- **touch .gitignore** (*to create a file which will ignore some files which you don't want to track*)

- **git branch <Branch\_Name>** (*to create* ***new branch***)

**- git branch** (*to* ***list*** *all the branch*)

- **git checkout** **<branch\_Name>** (*to get/****switch*** *into a specific branch*)

- **git checkout -b <branch\_Name>** (*to* ***create*** *a new branch and* ***switch*** *to that branch simultaneously*)

- **git merge <branch\_Name>** (*to* ***merge*** *the other branches with master branch*)

- **git remote add <remote\_name> <url>** (*this command will* ***link*** *my* ***local repository*** *which is in my pc with the* ***remote*** *or server repository which is the* ***GitHub***)

ex- git remote add origin http://www.github.com/bisesjklj

- **git remote set-url origin <new\_url>** (this is used to change the remote location or repository of a git repo)

- **git remote** (*this will show my* ***all-remote*** *names*)

- **git remote -v** (*this will show fetch and push* ***links*** *created for my remote repo*)

- **git push <remote> <branch\_Name>** (*this will* ***push*** *only* ***specified branch*** *from local repo to the remote repo*)

ex- git push origin master

- **git push <remote> --all** (*this will* ***push all the branches*** *from local repo to remote repo*)

- **git push -u <remote> <branch\_Name>** (*this will set an* ***upstream*** *for a branch so that whenever we* ***push*** *any update, we can directly write* ***<git push>*** *and it will push update to the same branch without writing the whole* ***<git push origin master>*** *command*)