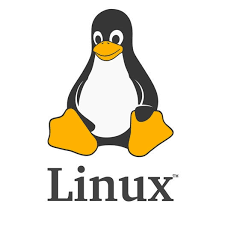
**Linux Commands** 

**pwd**: to check the present working directory

**ls**: list all the directories

**ls -a**: list all including hidden files and directories

**mkdir**: create a new directory

**mkdir –p A/B/C/…**: make nested directories [where A, B, C are the names of the directories

**cd**: change directory

**cd ..**: go to previous directory

**cat <*file name*>**: to view contents of a file

**sudo**: to grant super-user permissions to any file

**chmod**: to change permissions of any given file

**tail**: This command prints the last N numbers of data of the given input. By default, it prints 10 lines.

**sort**: This command is used to sort the results of search either alphabetically or numerically. It also sorts files and directories.

**chown**: This command is used to change the file Owner or group.

**vim**: This is a text editor used in Linux. It stands for “Vi Improved”.

**history**: This command is used to view the previously executed command.

**diff**: This command is used to find the difference between two files.

**ssh-keygen**: This command is used to generate a public/private authentication key pair.

**ssh user@host:** connect to the host as a user.

**clear:** to clear the screen all at once

**Git commands** 

**git clone [url]:** to clone a repository to your local.

**git init:** initialise an existing directory as a Git repository.

**git add <*file name*>**: to stage a file so that it can be committed.

**git commit -m “*message*”**: to commit the added file along with a message.

**git push origin <*branch name*>**: to push the changes made to the file in the local to the GitHub repository.

**git reset <*file name*>**: unstage a file while retaining the changes in working directory.

**git diff**: difference of what is changed but not staged.

**git branch**: list your branches. \* Indicates the currently active branch.

**git branch <*branch-name>***: create a new branch at the current commit

**git checkout**: switch to another branch and check it out into your working directory

**git merge <*branch*>**: merge the specified branch’s history into the current one

**git log**: show all commits in the current branch’s history

**git rm <*file*>**: delete the file from project and stage the removal for commit.

**git fetch [alias]**: fetch down all the branches from that Git remote

**git merge [alias]/[branch]**: merge a remote branch into your current branch to bring it up to date

**git pull**: fetch and merge any commits from the tracking remote branch

**git pull –rebase**: combining your local unpublished changes with the latest published changes on your remote

**git stash**: Save modified and staged changes

**git stash pop**: write working from top of stash stack

**git stash drop**: discard the changes from top of stash stack