UNIX

Tree-(to view everything in tree form)

pwd-(shows path of present working directory)

mkdir-(to make directory/folder)

cd directoryname/fooldername-(to go inside a directory or folder)

touch filename-(to create a file)

rm filename/filepath-(to remove a file)

rmdir directory/foldername-(to remove empty directory)

rm -r directoryname/foldername-(to delete directory along with the files in it)

clear-(to clear the screen)

cp filenametobecopied nameofcopiedfile(to make a copy of one file)

mv filename/filepath newfilename/filepath-(used to move or rename a file/folder/directory)

vi filename(to enter a file and check content inside it)
I (to insert into the file or to edit the file)
press esc after the changes are done
:wq-(here w writes the changes and q saves it)
:q!-(forcefully quit a file without writing/saving the changes)
cat filename(to check the content)

note: vi command can also be used to create a new file and directly add up a data into it at the same time (this file will only be getting saved when you press :wq else it will not)

| (pipe symbol is used to merge two queries together)
example
cat filename | wc
here cat filename goes to the file content
and wc gets the word count and in the output the word count is printed

UNIX

```
ln filenameone newfilename
```

(ln is used to hard link a file with a new file)
(hard linked file works as a backup file for anything. After the link if any change is made in one file it will be reflecting in the other also)

```
ln -s filename newfilename
```

(this command is used to make a shortcut or a soft copy of anything
this soft copy only work till the original file exists else it will not
also it wont work if you change the path of the file)

```
vi filename.sh
```

(to create a shell file
this file can execute multiple files all at onces)

./filename.sh (to execute a shell file)

permissions
ls -l shows permission
1 three for users
next 3 for groups
and last 3 for others

```
chmod +x filename.sh
```

(to add executable permission onto a file only for users)

```
chmod 777 filename.sh
```

(to add all permission onto a file for all-users, groups, others)

```
chmod 666 filename.sh
```

(to add read/write permission onto a file for all-users, groups, others)

```
chmod 764 filename.sh
```

(here the first 7 provides all permissions to the users, the 6 provides read and write permission to the groups, 4 provides the read permission only to others

the combination of these 3 decides what all permission one file hold for what user)

ps (it is used to check out for all running processes)
top (shows the top running processes)

UNIX

kill -9 processID
(it is used to kill an specific process)

whoami (tells you the current user name)

who (tells you the current user name and)

going to (cd /etc) and then going to (cat passwd) shows you all the users of the ubuntu

sudo useradd username (helps to add new user)

foreground processes (they block the terminal until it is finished)

background process (they will be happening in the background and you can have this in background and execute other commands till that time)

to make a normal command as a background command add & sign in the end
example- **sleep 5 &**

jobs (used to check all the running background processes)

fg processid %
(it is used to bring process from background to foreground)

wildcards (*, ?)
used to check for a file with an specific value
* is used when you are not sure

FOR LOOP Syntax

```
for variable in item1 item2 item3
        do
                echo “$variable”
done
```

```
for variable in {1..10..2}
do
            echo “$variable”
done
```

WHILE LOOP SYNTAX

```
num=1
while [ $num -le/-ge 10 ]
do
            echo “$variable”
            num=$((num+1))
done
```

UNTIL LOOP SYNTAX

```
num=1
until [ $num -le/-ge 10 ]
do
            echo “$variable”
            num=$((num+1))
done
```

note: while loop will go into the loop till the time the condition is satisfied
the first time the condition is not satisfied it will take the control out of the loop
whereas
until link will check for all the values which satisfy the condition and will still continue to check the remaining values even after it finds one not acceptable value