

# Bash shell scripting

what is Command?

→ A command is a line that tells the terminal to perform an action.

Syntax of Commands

Command (option) (something)

> echo -n "hello"

> killall name of process (xfce4, patedit, etc)

> cal month & year

Commands

> whoami

> ls

> ls -a

> pwd

> ls -l {more details}

no of links  
↑ owner group  
n n n size

(dewrte-re-re) 4 Macbook [staff] 136 Date  
directory name

Macbook

owner can  
Read, write,  
execute the file

ewr {e-re-re}

group  
refers to which

everyone else  
who is not the  
owner nor the  
group

this file  
assigned

staff

-R { Recursively }  
done work recursively

Read/write +

- represent that + { Giver file - - - }

It is a file & ~~EW-~~  
doesn't belong any  
directory

> ls -la namefolder { details of hidden file in - }

> file a.txt { gives information files }

> cd Desktop { change directory }

> cd .. { move back 1 folder }

{ cd.../.. } Back twice

> open file/folderName { open the respective }

{ open . } { open current directory in Finder }

> touch name { create file }

> mkdir name { create directory }

> mv oldname newname { rename the file }

> mv name FolderName { move a particular folder }

> mv Desktop/does/file.txt pwd { move file to pwd }

> cp main.txt copyName.txt { copy the files }

> nano filename { edit file }

> cat filename { show content }

> rm filename { remove file }

\* replace  
with  
anything  
{ a1.txt  
all.txt  
a111.txt  
a1111.txt }

> mv ax Desktop/does/NewFolder

a1

a11

a111

a1111

## ⊕ Redirection Command :-

↑ Redirection

> echo " something " > newtext.txt

{ If it does not exist it creates }

If exists otherwise replace all the content  
with " something "

> echo " AIBI " >> newtext.txt

cat { something }  
          { AIBI }

> cat ext1.txt ext2.txt { concatenate the 2 files }

## ⊕ Pipe Command { | }

> ls -A -Poldel | less =

ls -A -Poldel > kmPorary.txt

cat kmPorary.txt

less kmP -- txt.

② less =

same as the right sides  
commands

## ⊕ find

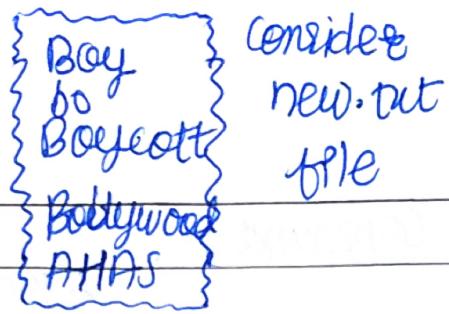
> find newFolder/ -name name.txt

② { find the name.txt in newFolder }

> find newFolder/ -type d -name directory

find files

\$ grep



> grep bo new.txt → bo

> grep BO new.txt → Boy, Boycott, Bolly...

> grep -i bo new.txt → Boy, bo, Boycott, Bolly...

> grep s b.txt → {printing the line containing s }

> grep s -v b.txt → {print other than s }

\$ if any file is not readable, writable or executable use 'sudo' command after that user will act as superuser {root} give access all the stuff.

OR

\$ {change the ownership}

who wanna make owner

> sudo chown [ ] filename

chgrp → {change group}

\$ change the permission of the particular file

u for the owner, 'g' for the group, 'o' for others

> chmod u=rw filename

> chmod +x filename → Executable to everyone  
-x → None

\* change owner & permission for folder :-

> sudo chown -R bhuwanthambu:root /home/Bhudeep  
chgrp -R

\* variable

> myvar=573

myvar<sup>x</sup>=573, myvar=573

> echo "\$myvar".

> unset myvar 2 unset the variable?

assigning command to variable

> mycommand=ls.

~~echo~~

> \$mycommand { is equally ls? }

\* \$(command) Command Substitution

> d=\$(ls)

> echo \$d

---

---

---

another way writing

\$() = ~ ~

{ Back tick }

> read -s Password

\* Read

> read -p "Enter name" \$add\_line{

> read myvar

> echo \$myvar

---

---

## Introduction to Shell Scripting :-

⊕ The terminal usually allows just one command at the time

⊕ Shell script allows you to combine & even multiple commands together  
 ↳ also you can use if else & loops.

### The first script & how it works

⊕ Arithmetic

> echo \$((number+1))

⊕ IF

if [ 3 -eq 3 ]; then  
 echo "

{  
 3 -eq 3 }  
 3 -ne 4 }  
 3 -gt 1 }  
 3 -lt 7 }  
 3 -ge 3 }  
 3 -le 3 }

fi

⊕ IF ELSE

if [ \$age -gt 18 ]; then

echo "

else echo "

fi

else [ --- ]; then

echo.

`== equal`    `!= [ ! cond ]`

1/1

⊕ Logic Condition

↑ OR

`if [ cond1 -o cond2 ] ; then`

↑ AND

`if [ cond1 -a cond2 ] ; then`

⊕ cond" to check empty strings

`if [ -z "$str" ] ; then`

`echo "empty"`

↑

⊕ file exists

`if [ -e $myfile ] ;`

--

⊕ directory

`if [ -d $myfile ]`

-e: readable

⊕ if file empty

`if [ ! -s $myfile ] empty`

OR

NOT empty

∅ loops

∅ for i in {1..5}  
do  
echo "i=\$i"  
done

∅ while

number=1  
while [ \$number -le 7 ]  
do  
echo  
number=\$((number+1))  
done

∅ sleep num

for --  
do  
echo  
sleep 2  
done

∅ case

> echo "select A to --"  
> echo "select B to --"  
> read choice  
> case \$choice in  
1) echo "you choose A";;  
2) echo "you choose B";;  
\*)  
esac

## \$ exit status

> echo \$? if it 0 correct

> \$ Not correct (second issue)

## \$ functions

```
mydate() {  
    date  
}  
> mydate  
hello "B", "T"
```

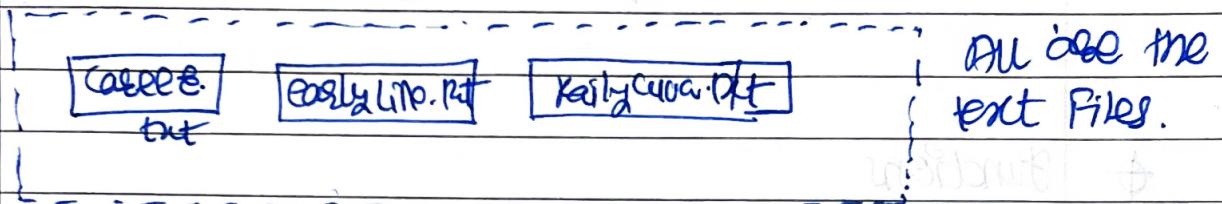
```
sum() {  
    return($1+$2)  
}
```

```
sum "1" "2"  
echo "return value $?"  
} only return the return  
not exactly now
```

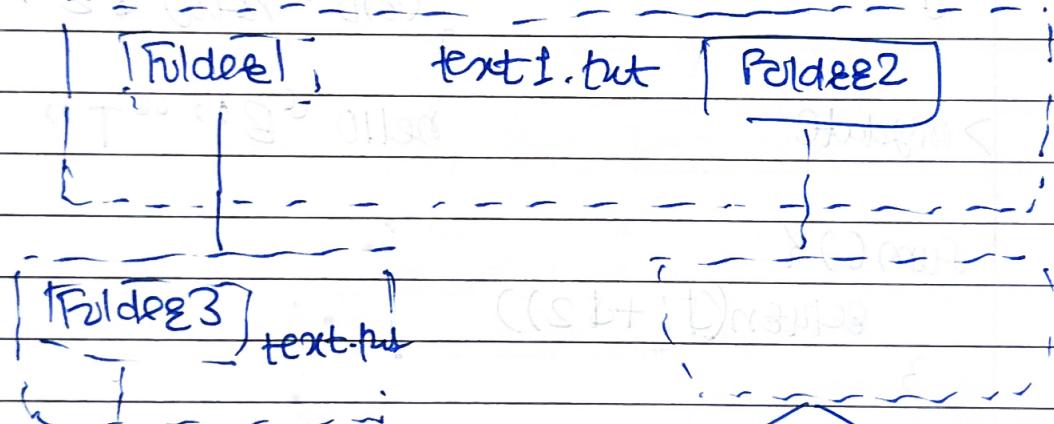
## \$ variables

all variables are global unless we write  
local a=35

Project 1 :- Finding the hot word into files which are only present in the current directory.



Project 2 :- Finding the hot word into multiple files of folders or within subfolders.



String matching with all the files in the directory