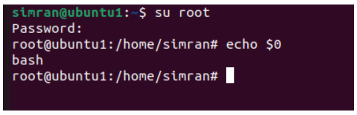
Shell Scripting:

"shell" generally refers to a command-line interface (CLI) that allows users to interact with the operating system by typing text-based commands. The default shell in Ubuntu is called Bash (Bourne Again SHell), but other shells like Zsh, Fish, and others are also available and can be installed.

Check the default shell in VM

echo $0



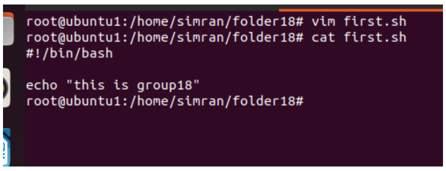
```
simran@ubuntu1:~$ su root
Password:
root@ubuntu1:/home/simran# echo $0
bash
root@ubuntu1:/home/simran# 
```

Here a script consists of set of commands and it get executed sequentially at once.

Here with scripts we can introduce automation

Example1:

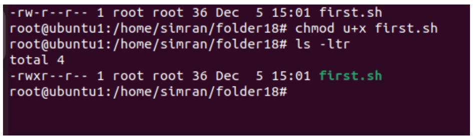
Make a script with .sh extension to print a text



```
root@ubuntu1:/home/simran/folder18# vim first.sh
root@ubuntu1:/home/simran/folder18# cat first.sh
#!/bin/bash

echo "this is group18"
root@ubuntu1:/home/simran/folder18#
```

Whenever we want to execute a script. In file the very first line should about the interpreter we are going to use or the shell #!/bin/bash

In order to run the script the file should have execute permission also



```
-rw-r--r-- 1 root root 36 Dec  5 15:01 first.sh
root@ubuntu1:/home/simran/folder18# chmod u+x first.sh
root@ubuntu1:/home/simran/folder18# ls -ltr
total 4
-rwxr--r-- 1 root root 36 Dec  5 15:01 first.sh
root@ubuntu1:/home/simran/folder18#
```

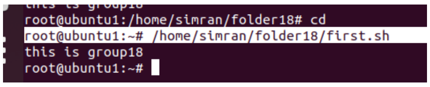
To execute the file type:

When at the current location only you have .sh file

./first.sh

When you are not at the exact location where file is present then mention the whole path

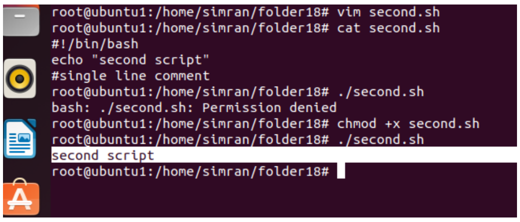
/home/simran/folder18/first.sh



```
this is group18
root@ubuntu1:/home/simran/folder18# cd
root@ubuntu1:~# /home/simran/folder18/first.sh
this is group18
root@ubuntu1:~# 
```

How to add comments in .sh file:

#for single line comment



```
root@ubuntu1:/home/simran/folder18# vim second.sh
root@ubuntu1:/home/simran/folder18# cat second.sh
#!/bin/bash
echo "second script"
#single line comment
root@ubuntu1:/home/simran/folder18# ./second.sh
bash: ./second.sh: Permission denied
root@ubuntu1:/home/simran/folder18# chmod +x second.sh
root@ubuntu1:/home/simran/folder18# ./second.sh
second script
root@ubuntu1:/home/simran/folder18# 
```

Multi-line Comment:

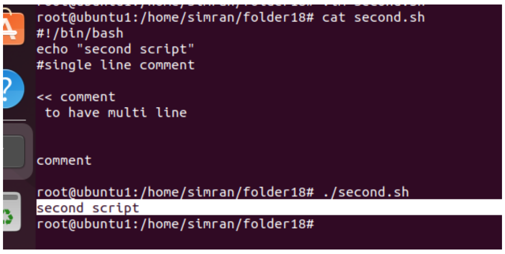
<<comment

Type the

Text here

Comment

(to end the multi-line comment just mention the very first word you have used after <<)



```
root@ubuntu1:/home/simran/folder18# cat second.sh
#!/bin/bash
echo "second script"
#single line comment

<< comment
 to have multi line


comment

root@ubuntu1:/home/simran/folder18# ./second.sh
second script
root@ubuntu1:/home/simran/folder18#
```

Variables:

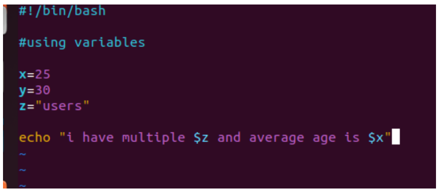
How you define variables-

X=25

Y=30

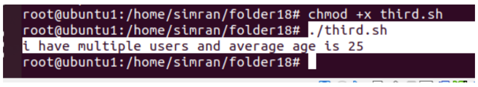
How to use variables in script-

echo “avg age of users is $Y”



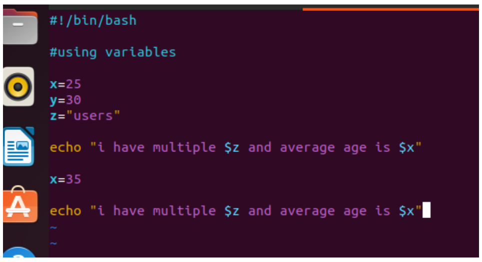
```
#!/bin/bash

#using variables

x=25
y=30
z="users"

echo "i have multiple $z and average age is $x"
~
~
```

Now execute the script:

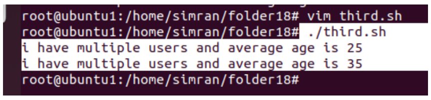


```
root@ubuntu1:/home/simran/folder18# chmod +x third.sh
root@ubuntu1:/home/simran/folder18# ./third.sh
i have multiple users and average age is 25
root@ubuntu1:/home/simran/folder18#
```

If we change the variable value in between of the script



```
#!/bin/bash

#using variables

x=25
y=30
z="users"

echo "i have multiple $z and average age is $x"

x=35

echo "i have multiple $z and average age is $x"
~
~
```



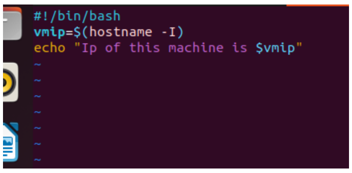
```
root@ubuntu1:/home/simran/folder18# vim third.sh
root@ubuntu1:/home/simran/folder18# ./third.sh
i have multiple users and average age is 25
i have multiple users and average age is 35
root@ubuntu1:/home/simran/folder18#
```

How to store the output of any command inside a variable:

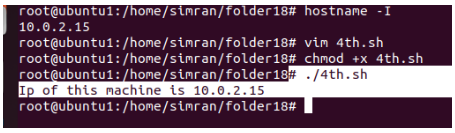
We have a command to check the ip of our vm

hostname -I

Let’s store this command inside a variable and then print the value of that variable



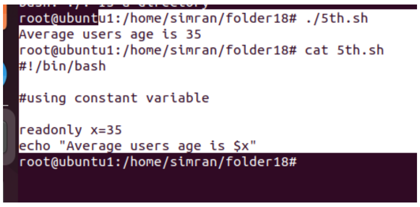
```
#!/bin/bash
vmip=$(hostname -I)
echo "Ip of this machine is $vmip"
~
~
~
~
~
~
~
```



```
root@ubuntu1:/home/simran/folder18# hostname -I
10.0.2.15
root@ubuntu1:/home/simran/folder18# vim 4th.sh
root@ubuntu1:/home/simran/folder18# chmod +x 4th.sh
root@ubuntu1:/home/simran/folder18# ./4th.sh
Ip of this machine is 10.0.2.15
root@ubuntu1:/home/simran/folder18#
```

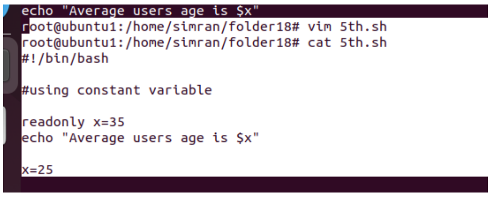
How to make a constant variable

Just mention readonly before mentioning any variable



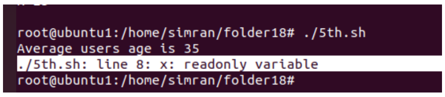
```
root@ubuntu1:/home/simran/folder18# ./5th.sh
Average users age is 35
root@ubuntu1:/home/simran/folder18# cat 5th.sh
#!/bin/bash

#using constant variable

readonly x=35
echo "Average users age is $x"
root@ubuntu1:/home/simran/folder18#
```

If I add another value to that variable



```
echo "Average users age is $x"
root@ubuntu1:/home/simran/folder18# vim 5th.sh
root@ubuntu1:/home/simran/folder18# cat 5th.sh
#!/bin/bash

#using constant variable

readonly x=35
echo "Average users age is $x"

x=25
```

Now try to execute it

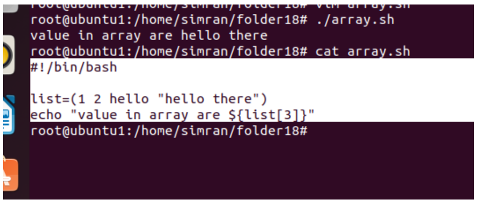


```
root@ubuntu1:/home/simran/folder18# ./5th.sh
Average users age is 35
./5th.sh: line 8: x: readonly variable
root@ubuntu1:/home/simran/folder18#
```

See it Is showing error for the variable

How array work here:

- Index number starts from 0 only
- You can put space separated values in the array



```
root@ubuntu1:/home/simran/folder18# ./array.sh
value in array are hello there
root@ubuntu1:/home/simran/folder18# cat array.sh
#!/bin/bash

list=(1 2 hello "hello there")
echo "value in array are ${list[3]}"
root@ubuntu1:/home/simran/folder18#
```

To get all values of array:

Just use → echo(“all values in array are ${list[*]}

How to find total number of values present in array:

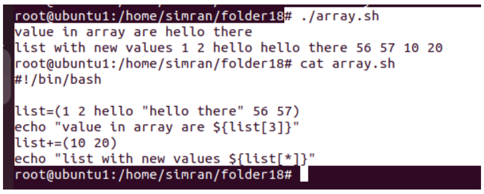
echo “length of array is ${#list[*]}”

how to get specific values from array:

echo”values from 3-4 index ${list[*]:3:2}”

This means starting from 3th index we want to see 2 more values from the index

How to update the array with new values:

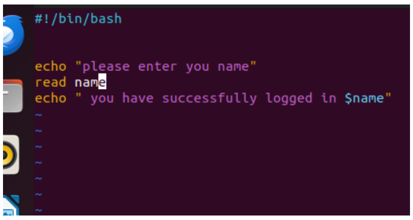


```
root@ubuntu1:/home/simran/folder18# ./array.sh
value in array are hello there
list with new values 1 2 hello hello there 56 57 10 20
root@ubuntu1:/home/simran/folder18# cat array.sh
#!/bin/bash

list=(1 2 hello "hello there" 56 57)
echo "value in array are ${list[3]}"
list+=(10 20)
echo "list with new values ${list[*]}"
root@ubuntu1:/home/simran/folder18#
```

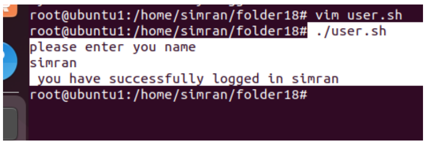
String Operations:

User Interactive scripts:

To take input from user we use read command



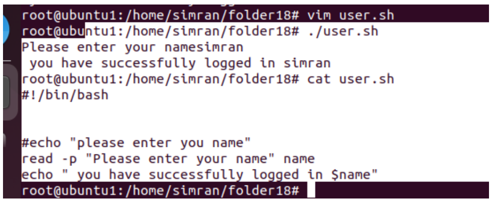
```
#!/bin/bash

echo "please enter you name"
read name
echo " you have successfully logged in $name"
~
~
~
~
~
~
~
```



```
root@ubuntu1:/home/simran/folder18# vim user.sh
root@ubuntu1:/home/simran/folder18# ./user.sh
please enter you name
simran
 you have successfully logged in simran
root@ubuntu1:/home/simran/folder18#
```

With read command we can also use -p option to print the message

read -p “enter the message here “   name



```
root@ubuntu1:/home/simran/folder18# vim user.sh
root@ubuntu1:/home/simran/folder18# ./user.sh
Please enter your namesimran
 you have successfully logged in simran
root@ubuntu1:/home/simran/folder18# cat user.sh
#!/bin/bash


#echo "please enter you name"
read -p "Please enter your name" name
echo " you have successfully logged in $name"
root@ubuntu1:/home/simran/folder18#
```

Arithmetic Operations:

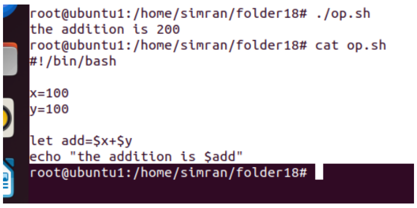
Here we use the let command

X=100

Y=100

let add=$x+$y

Echo” the total is $add”



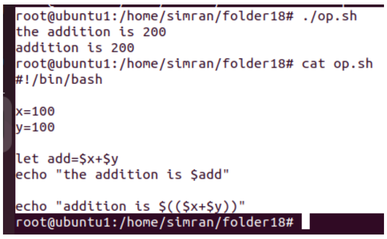
```
root@ubuntu1:/home/simran/folder18# ./op.sh
the addition is 200
root@ubuntu1:/home/simran/folder18# cat op.sh
#!/bin/bash

x=100
y=100

let add=$x+$y
echo "the addition is $add"
root@ubuntu1:/home/simran/folder18# 
```

Another way to directly perform the operation

X=100

Y=100

echo”the addition is $(($x+$y))”



```
root@ubuntu1:/home/simran/folder18# ./op.sh
the addition is 200
addition is 200
root@ubuntu1:/home/simran/folder18# cat op.sh
#!/bin/bash

x=100
y=100

let add=$x+$y
echo "the addition is $add"

echo "addition is $(($x+$y))"
root@ubuntu1:/home/simran/folder18# 
```

If-Else :

If[condition]

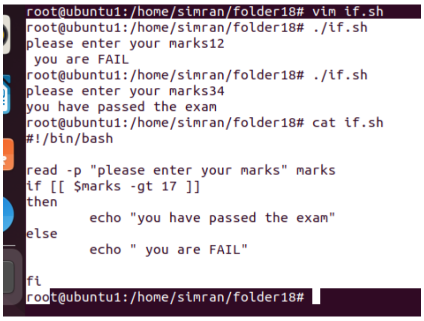
Then

    Echo “print”

Else

    Echo”print”

Fi (to close it)



```
root@ubuntu1:/home/simran/folder18# vim if.sh
root@ubuntu1:/home/simran/folder18# ./if.sh
please enter your marks12
 you are FAIL
root@ubuntu1:/home/simran/folder18# ./if.sh
please enter your marks34
you have passed the exam
root@ubuntu1:/home/simran/folder18# cat if.sh
#!/bin/bash

read -p "please enter your marks" marks
if [[ $marks -gt 17 ]]
then
        echo "you have passed the exam"
else
        echo " you are FAIL"

fi
root@ubuntu1:/home/simran/folder18# 
```

Comparison operator which you can use

-eq/== →equal

-ge → greaterthanorequalto

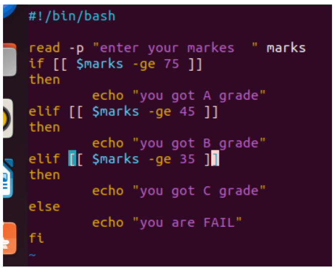
-le →lessthanorequalto

-ne / != →not equal

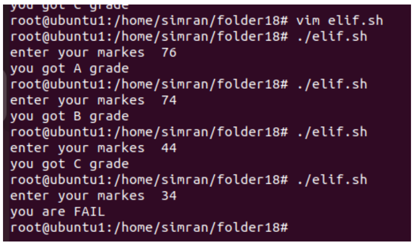
-gt →greaterthan

-lt →lessthan

Example of Elif in order to add multiple conditions:

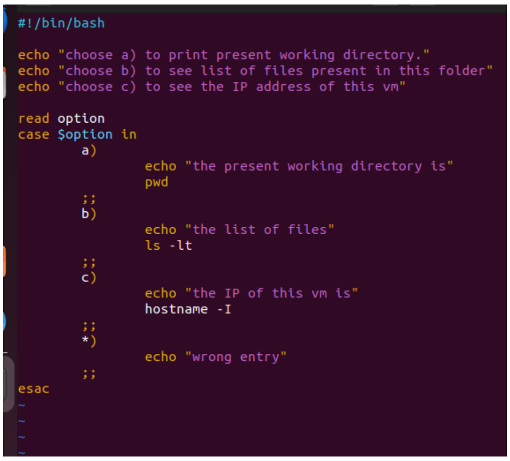


```
#!/bin/bash

read -p "enter your markes  " marks
if [[ $marks -ge 75 ]]
then
        echo "you got A grade"
elif [[ $marks -ge 45 ]]
then
        echo "you got B grade"
elif [[ $marks -ge 35 ]]
then
        echo "you got C grade"
else
        echo "you are FAIL"
fi
~
```

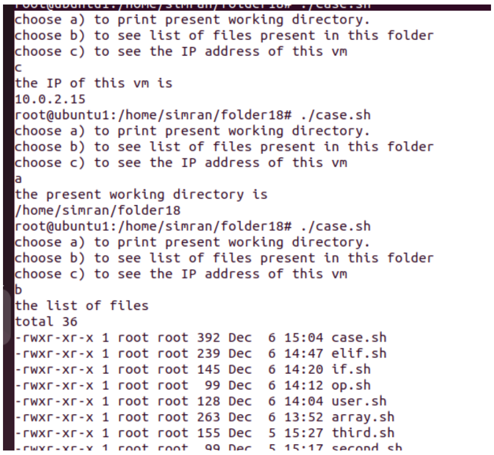


```
you got C grade
root@ubuntu1:/home/simran/folder18# vim elif.sh
root@ubuntu1:/home/simran/folder18# ./elif.sh
enter your markes  76
you got A grade
root@ubuntu1:/home/simran/folder18# ./elif.sh
enter your markes  74
you got B grade
root@ubuntu1:/home/simran/folder18# ./elif.sh
enter your markes  44
you got C grade
root@ubuntu1:/home/simran/folder18# ./elif.sh
enter your markes  34
you are FAIL
root@ubuntu1:/home/simran/folder18#
```

CASE statements:



```
#!/bin/bash

echo "choose a) to print present working directory."
echo "choose b) to see list of files present in this folder"
echo "choose c) to see the IP address of this vm"

read option
case $option in
        a)
                echo "the present working directory is"
                pwd
        ;;
        b)
                echo "the list of files"
                ls -lt
        ;;
        c)
                echo "the IP of this vm is"
                hostname -I
        ;;
        *)
                echo "wrong entry"
        ;;
esac
~
~
~
~
```



```
root@ubuntu1:/home/simran/folder18# ./case.sh
choose a) to print present working directory.
choose b) to see list of files present in this folder
choose c) to see the IP address of this vm
c
the IP of this vm is
10.0.2.15
root@ubuntu1:/home/simran/folder18# ./case.sh
choose a) to print present working directory.
choose b) to see list of files present in this folder
choose c) to see the IP address of this vm
a
the present working directory is
/home/simran/folder18
root@ubuntu1:/home/simran/folder18# ./case.sh
choose a) to print present working directory.
choose b) to see list of files present in this folder
choose c) to see the IP address of this vm
b
the list of files
total 36
-rwxr-xr-x 1 root root 392 Dec  6 15:04 case.sh
-rwxr-xr-x 1 root root 239 Dec  6 14:47 elif.sh
-rwxr-xr-x 1 root root 145 Dec  6 14:20 if.sh
-rwxr-xr-x 1 root root  99 Dec  6 14:12 op.sh
-rwxr-xr-x 1 root root 128 Dec  6 14:04 user.sh
-rwxr-xr-x 1 root root 263 Dec  6 13:52 array.sh
-rwxr-xr-x 1 root root 155 Dec  5 15:27 third.sh
-rwxr-xr-x 1 root root  99 Dec  5 15:17 second.sh
```