**Shell Scripting**

Script file has extension of .sh (without extension also it can be script file)

+++++++++++++++++++++++++++++++++++++++++++++

**To create file** we can run,

touch test.sh (file can be created using vi also)

To edit we can use nano or vi

If vi is used, then press **i** to insert data then **esc:wq enter** to exit

+++++++++++++++++++++++++++++++++++++++++++++

Shell script file always start with

**#!/bin/bash**

+++++++++++++++++++++++++++++++++++++++++++++

**To print content**,

use echo

echo Hello World

e.g.

#!/bin/bash

echo Hello World

+++++++++++++++++++++++++++++++++++++++++++++

**To run the script file**

use ./file\_name

File won’t get run as it has no execute permission

To add execute permission, run

chmod u+x file\_name

+++++++++++++++++++++++++++++++++++++++++++++

**Variables**:

name=abhishek

age=27

echo My name is $name and age is $age

Variables are printed using $ sign

\*\* Note: there should not be any space between variable name and its value, else we get as variable not declared error

+++++++++++++++++++++++++++++++++++++++++++++

**To store output of linux command in variable**

e.g. hostname command gives name of host and groups command gives group names in linux

To store output of this command in a variable

inside $() pass linux command

host=$(hostname)

group=$(groups)

echo $host

echo $groups

+++++++++++++++++++++++++++++++++++++++++++++

**To read user input**

User read command

e.g.

echo what is your name?

read name (at this it waits for user input)

echo welcome $name

+++++++++++++++++++++++++++++++++++++++++++++

**If else conditions**

Operators

| Equal to | -eq or == |
| --- | --- |
| Not equal to | -ne or != |
| Greater than | -gt |
| Less than | -lt |
| Greater than equal to | -ge |
| Less than equal to | -le |

e.g.

echo what is your age?

read age

if [ $age -ge 18 ] → see the spaces between characters

then

echo eligible

else

echo not eligible

fi →(endif)

+++++++++++++++++++++++++++++++++++++++++++++

**Switch Case:**

Note\*\*: use **;;** after each case

case must be end with **esac**

It will read as if choice is in 1 then execute this or choice is in 2 execute this etc.

If no condition match then to handle that use **\*)**

#!/bin/bash

echo enter the choice

echo 1=get date

echo 2=get current directory

echo 3=get list of files

read choice

case $choice in

1)date;;

2)pwd;;

3)ls -l;;

\*)echo invalid choice

Esac

+++++++++++++++++++++++++++++++++++++++++++++

**For loop**

Different Syntax:

for i in 1 2 3 4 5

or

for i in {1..5}

or

for i in abhishek joshi

**do**

**What to do**

**done**

e.g.

for i in 1 2 3 4 5

do

echo Number is $i

done

+++++++++++++++++++++++++++++++++++++++++++++

**While Loop**

num=0

while [ $num -le 10 ]

do

echo num is $num

let num++ → let command used to use expressions

Done

+++++++++++++++++++++++++++++++++++++++++++++

**Read from a file**

Using for loop, we can get content of the file

Say one file is present name.txt

cat command is used to read file

e.g.

#!/bin/bash

file\_name='/home/abhishek/Documents/test7/name.txt'

for content in $(cat $file\_name)

do

echo name is $content

done

+++++++++++++++++++++++++++++++++++++++++++++

**Get name of script that is being run**

echo Name of script is ${0}

We get o/p as ./name\_of\_script.sh

+++++++++++++++++++++++++++++++++++++++++++++

**Functions**

**syntax:**

function f\_name {

echo inside function

}

or

f\_name() {

Echo inside\_function

}

To call function

f\_name

**Function with parameters**

Parameters passed to function are taken in order as $1, $2 resp.

echo function with parameters

addition() {

num1=$1

num2=$2

let sum=$num1+$num2

echo sum is $sum

}

addition 10 20 → calling function with parameter

addition 20 5

+++++++++++++++++++++++++++++++++++++++++++++

**ROUND AND UID**

These are two keywords used in linux

Random produce random number between 0 and 32767

In any terminal run the command

echo $RANDOM

UID: 0 for root user, this can be used to check whether scrip or file is ran from root user or not

Run from any terminal

echo $UID

+++++++++++++++++++++++++++++++++++++++++++++

**Print content of file from script using redirects**

file=’path\_of\_file\_to\_read’

read text < $file → content of file are redirected to variable text

echo $text

+++++++++++++++++++++++++++++++++++++++++++++

**Save output of script to file**

For this we can use

> → for write mode

>> → for append mode

Say, i want to store output of pwd command to file

pwd > output.txt

+++++++++++++++++++++++++++++++++++++++++++++

To store errors in files also

Suppose pwdd is not a command

So error of this need to store in to file

For that use 2>&1 command

pwdd > error.txt 2>&1

+++++++++++++++++++++++++++++++++++++++++++++

**/dev/null**

Suppose we neither want to display output of command on terminal nor write to file in that case we can use dev/null

In any terminal

e.g. pwd &>/dev/null

+++++++++++++++++++++++++++++++++++++++++++++

**Passing arguments to scripts**

Arguments can be passed while running the script also,

e.g. ./test.sh 10 20

We can access these arguments inside scripts as

$# → number of argos passed

$@ → gives all arguments passed

$1, $2 → gives arguments one by one

+++++++++++++++++++++++++++++++++++++++++++++

**Sleep Command**

To give delay between two executions

syntax: sleep 10s/m/h/d

+++++++++++++++++++++++++++++++++++++++++++++

**Stop/exit script**

To stop execution of script

echo abhishek

echo joshi

exit

echo roll

+++++++++++++++++++++++++++++++++++++++++++++

**Constant variable**

Use readonly before variable name,

readonly name=abhishek

echo $name

name=joshi

echo $name

+++++++++++++++++++++++++++++++++++++++++++++

**Exit status ($?) of last command**

If last command is successful then it gives $? = 0 else $? = 1

From this we can check that whether our last command is successful or not

e.g. in any terminal type

ping -c1 '[www.google.com](http://www.google.com)' → ping google 1 time

Then type

$? → 0

After that type ping -c1 'localhost.com'

Then type

$? → 1