```
---------------------------------------------------------------
12-26-2020
https://youtu.be/RhOhupxTz4M
---------------------------------------------------------------
shell script
---------------------------------------------------------------

[root@script01 ~]# ls;pwd;echo;hostname;echo;lsblk;echo;df -h;echo;cat /etc/fstab

you can literally have unlimeted command running from CLI
this kind of commands is considered shell script

---------------------------------------------------------------
so you these command and put in a file, it bomes a bash script

---------------------------------------------------------------
Bash Script
---------------------------------------------------------------
#!                    #< --- shebang
/bin/bash             #< --- location of bash command


#!/bin/bash           #< --- you have to have this to become script

---------------------------------------------------------------

#!/bin/bash


ls;pwd;echo;hostname;echo;lsblk;echo;df -h;echo;cat /etc/fstab;

---------------------------------------------------------------
Method of executing the script file
-----------------------------------

[root@script01 ~]# ./script

---------------------------------------------------------------
Permission
----------
The script file must have execute permission

---------------------------------------------------------------

[root@script01 ~]# ls -l
total 4
-rw-r--r--. 1 root root 78 Dec 26 13:45 script
[root@script01 ~]# ./script
bash: ./script: Permission denied

[root@script01 ~]# chmod 755 script      #< ---execute permission added

[root@script01 ~]# ls -l
total 4
-rwxr-xr-x. 1 root root 78 Dec 26 13:45 script

---------------------------------------------------------------
execute the script
------------------


[root@script01 ~]# ./script  #< --- All the commands in the file will run

---------------------------------------------------------------
if you want to redirec the conent of the out to a file

[root@script01 ~]# ./script >> output.txt

```

```
output.txt will hold the output from the script

------------------------------------------------------------
Script Identification
---------------------
The script itself does not need any extension, but generally its
good idea to have one

Exmaples: .sh  .scr .scrp


[root@script01 ~]# ls -l
total 12
-rw-r--r--. 1 root root 2096 Dec 26 13:57 output.txt
-rwxr-xr-x. 1 root root   78 Dec 26 13:45 script        #< ---
-rwxr-xr-x. 1 root root   39 Dec 26 14:00 template.scr  #< ---

------------------------------------------------------------
comment out
-----------
# hash is used for comments or disabling specific line in script

#!/bin/bash


#ls;pwd;echo;hostname;echo;lsblk;echo;df -h;echo;cat /etc/fstab;  #< --- blind
ls;pwd;echo;hostname;echo;lsblk;echo;df -h;echo;cat /etc/fstab;
#ls;pwd;echo;hostname;echo;lsblk;echo;df -h;echo;cat /etc/fstab;  #< --- blind

------------------------------------------------------------


#!/bin/bash

echo "This is lsblk command"   #< --- This command will run

lsblk                          #< --- This command will run

echo "df -h is commented out"  #< --- This command will run

#df -h                   #< --- This command will not run

------------------------------------------------------------
basic.sh
--------

#!/bin/bash

#This is basic script with clean output

echo "-----This is output of ls -l----"

ls -l

echo "-----This is output of pwd----"

pwd

echo "-----This is output of hostname----"

hostname

echo "-----This is output of lsblk----"

lsblk

echo "-----This is output of df -h----"

df -h
```

```
echo "-----This is output of cat /etc/fstab----"

-------------------------------------------------------------
commandlocation.scr
-------------------

#!/bin/bash

#This is a commandlocation.scr


df -h
echo

echo "this is ls -l `ls -l` "  #< ---use backtick ` if you have to use
                                     the command middle of the line


-------------------------------------------------------------
Arithematic Operators
---------------------

+  addition
-  subtraction
*  multiplication  #< ---when using inside script use \*
/  division
-------------------------------------------------------------
expr           #< --- expr is built-in command, stands for expression
-----

[root@script01 ~]# expr 2 + 2
4
[root@script01 ~]# expr 2 - 2
0
[root@script01 ~]# expr 2 / 2
1
[root@script01 ~]# expr 2 \* 2
4

-------------------------------------------------------------
booblean operators    #< --- comparision ture or false
-----------------------------------------------------

-eq  ==  equal to
-ne  !=  not equal to
-gt  >   greater than
-lt  <   less than
-le  <=  less than or equl to
-ge  >=  greater than or equal to

|| means "or"  - two pipes - this is key above "Enter"

&& mesn  "and"

$? - to check the exit code

-------------------------------------------------------------
read - this is built in command, provides chance to user for input
-------------------------------------------------------------



[root@script01 ~]# read
zafar                  #< --- input is sitting memory

There is no way to recall whats in memory

[root@script01 ~]# read name1  #< ---you need to give a place holder [variable], name1
```

```
is variable
zafar                           #< --- now its sitting memory

[root@script01 ~]# echo $name1 #< --- echo command is used to read from memory
zafar

--------------------------------
another example
---------------


[root@script01 ~]# read name1
This is a linux course
[root@script01 ~]# echo $name1
This is a linux course


--------------------------------
another example
---------------
[root@script01 ~]# echo $name1
This is a linux course
[root@script01 ~]# read name2
haroon
[root@script01 ~]# read name3
burhan
[root@script01 ~]# read name4
adil
[root@script01 ~]# echo $name1 $name2 $name3 $name4 $name2  #< ---extracted from memory
This is a linux course haroon burhan adil haroon

------------------------------------------------------------

[root@script01 ~]# read num1
45
[root@script01 ~]# read num2
55
[root@script01 ~]# echo $num1 $num2
45 55
[root@script01 ~]# expr $num1 + $num2
100
[root@script01 ~]# expr $num1 - $num2
-10
[root@script01 ~]# expr $num1 \* $num2
2475
[root@script01 ~]# expr $num1 / $num2
0

------------------------------------------------------------

[root@script01 ~]# read num1
45
[root@script01 ~]# read num2
55
[root@script01 ~]# echo $num1 $num2
45 55
[root@script01 ~]# expr $num1 + $num2
100
[root@script01 ~]# expr $num1 - $num2
-10
[root@script01 ~]# expr $num1 \* $num2
2475
[root@script01 ~]# expr $num1 / $num2
0

------------------------------------------------------------

[root@script01 ~]# echo $name1 $num2 $name2 $num1 $name3 $name4 $name2
This is a linux course 55 haroon 45 burhan adil haroon
```

```
------------------------------------------------------------
 cal.sc
-------

#!/bin/bash

#This is a simple calculator

echo "Enter a first number"

read num1

echo "Enter a second number"

read num2

total=`expr $num1 + $num2`    #< --- total is place holder [variable - made up]

echo "The sum of $num1 and $num2 is : $total" #<

------------------------------------------------------------
in reality a script is extracted and executed on CLI by the system
------------------------------------------------------------

[root@script01 ~]# echo "Enter a first number"
Enter a first number
[root@script01 ~]# read num1
45
[root@script01 ~]# echo "Enter a second number"
Enter a second number
[root@script01 ~]# read num2
90
[root@script01 ~]# total=`expr $num1 + $num2`
[root@script01 ~]# echo "The sum of $num1 and $num2 is : $total"
The sum of 45 and 90 is : 135
[root@script01 ~]# echo "The sum of $num1 and $num2"
The sum of 45 and 90
[root@script01 ~]# echo $total
135

------------------------------------------------------------
you get same result running at CLI
------------------------------------------------------------

[root@script01 ~]# echo "Enter a first number";read num1;echo "Enter a second
number";read num2;total=`expr $num1 + $num2`;echo "The sum of $num1 and $num2 is :
$total"
Enter a first number
30
Enter a second number
40
The sum of 30 and 40 is : 70

------------------------------------------------------------
variable - place holder
-----------------------

What is variable

Variable is a placeholder for a input, commands etc..,
variable are defined by person writing a script
------------------------------------------------------------
Two types of Variable - static and dynamic
------------------------------------------------------------

Static variable
---------------

```

```
stvariable.st
-------------

[root@script01 ~]# chicago=illinois
[root@script01 ~]# newyork=ny
[root@script01 ~]# echo chicago
chicago
[root@script01 ~]# echo $chicago
illinois
[root@script01 ~]# echo $newyork
ny



#!/bin/bash

#This is a static vairable exmaple

chicago=illinois
newyork=ny


echo "$chicago is in midwest"
echo
echo "$newyork is in eastcoast"

-----------------------------------------------------------------
[root@script01 ~]# ./stvariable.st
illinois is in midwest

ny is in eastcoast

-----------------------------------------------------------------

Dynamic Variable
----------------

dyvariable.dy
-------------

#!/bin/bash

#This is a dynamic [changes] variable

echo "Enter a place in midwest"

read midcity

echo "Enter a place in east coast"

read eastcity

echo
echo "$midcity is in midwest"
echo
echo "$eastcity is in east coast"

-------------------------------------------------------

[root@script01 ~]# ./dyvariable.dy
Enter a place in midwest
st louis
Enter a place in east coast
new jersey

st louis is in midwest

new jersey is in east coast
```

```
[root@script01 ~]# ./dyvariable.dy
Enter a place in midwest
springfield
Enter a place in east coast
boston

springfield is in midwest

boston is in east coast

-----------------------------------------------------------------
Using Alias inside the script
-----------------------------------------------------------------

aliaszafar.scr
--------------


#!/bin/bash

#This is a exmample of using alias

shopt -s expand_aliases #< ---type exactly the way it is, it will set and unset once
the script is done

alias haroon="ls;pwd;echo;hostname;echo;lsblk;echo;df -h;echo;cat /etc/fstab;"

haroon

-----------------------------------------------------------------
12-27-2020

-----------------------------------------------------------------
```