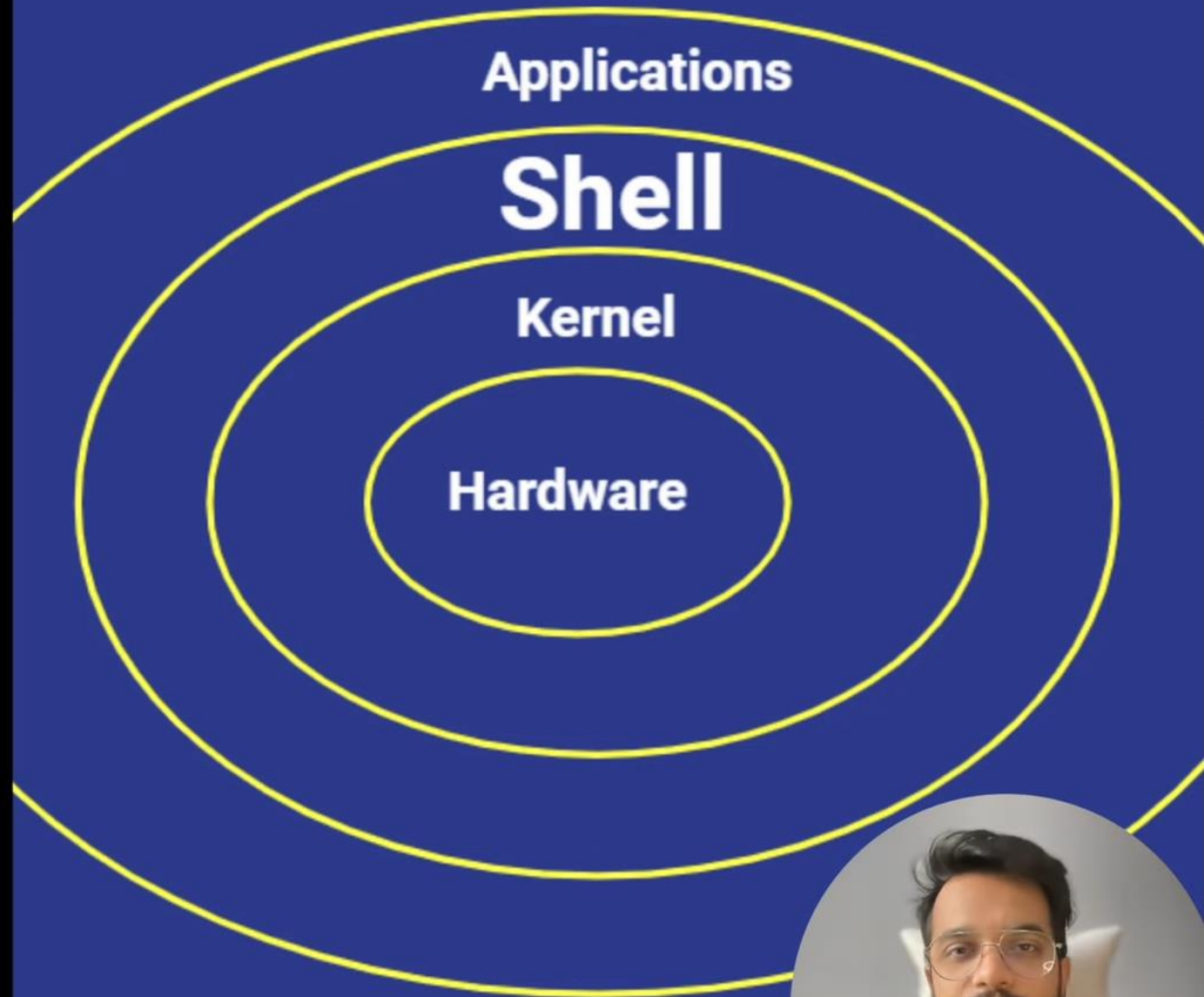


Shell Scripting

by Dev Namdev

WHAT IS LINUX SHELL?

A shell provide an environment to a user to execute commands and interact with kernel.



There are different types of shell

- bash
- sh
- ksh
- tsh
- fish
- zsh



WHAT IS MY SHELL TYPE?

You can check using
`echo $0`



```
[root@redhat01 script]#  
[root@redhat01 script]#  
[root@redhat01 script]# echo $0  
-bash  
[root@redhat01 script]# █
```



```
[root@redhat01 script]#  
[root@redhat01 script]#  
[root@redhat01 script]# cat /etc/shells  
/bin/sh  
/bin/bash  
/usr/bin/sh  
/usr/bin/bash  
[root@redhat01 script]# █
```



WHAT IS SHELL SCRIPTING?

- Shell script consist of set of commands to perform a task.
- All the commands execute sequentially.
- Some task like file manipulation, program execution, user interaction, automation of task etc can be done



MPRASHANT

WHAT IS SHEBANG?

```
#!/bin/bash
```



HOW TO RUN A SCRIPT..

- Make sure script has execute permission `rwx`
- Run using
 - `./script.sh`
 - `/path/script.sh`
 - `bash script.sh`
- `Ctrl+C` to terminate
- `Ctrl+z` to stop



MPRASHANT

FIRST BASIC SCRIPT...

```
#!/bin/bash
```

```
echo "Hello World!"
```



COMMENTS

Using #

```
#This is a comment
```

Multi-line comment

```
<<comment
```

```
...
```

```
your comment here
```

```
...
```

```
comment
```



MPRASHANT

WHAT ARE VARIABLES?

```
VAR_NAME=value
```

```
VAR_NAME=$(hostname)
```

```
echo $VAR_NAME
```



```
#Defining variables
```

```
name="Prashant"
```

```
age=30
```

```
echo "My name is $name and age is $age"
```

```
name="Paul"
```

```
echo "New name is ${name}"
```

```
HOSTNAME=$(hostname)
```

```
echo "Name of the server is $HOSTNAME"
```

```
PWD=$(pwd)
```

```
echo "We are in path $PWD"
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```



```
[root@redhat01 script]#  
[root@redhat01 script]# bash 03_vardemo.sh  
My name is Prashant and age is 30  
New name is Paul  
Name of the server is redhat01  
[root@redhat01 script]# vi 03_vardemo.sh  
[root@redhat01 script]# bash 03_vardemo.sh  
My name is Prashant and age is 30  
New name is Paul  
Name of the server is redhat01  
We are in path /root/script  
[root@redhat01 script]#
```



CONSTANT VARIABLE?

Once you defined a variable and don't wanna change it until end of the script.

```
readonly var_name="Hi"
```



```
#!/bin/bash
```

```
#Defining variables
```

```
readonly name="Prashant"
```

```
age=30
```

```
echo "My name is $name and age is $age"
```

```
name="Paul"
```

```
echo "New name is ${name}"
```

```
HOSTNAME=$(hostname)
```

```
echo "Name of the server is $HOSTNAME"
```

```
PWD=$(pwd)
```

```
echo "We are in path $PWD"
```

```
~
```

```
~
```

```
~
```

```
~
```




```
[root@redhat01 script]#  
[root@redhat01 script]# vi 03_vardemo.sh  
[root@redhat01 script]# bash 03_vardemo.sh  
My name is Prashant and age is 30  
03_vardemo.sh: line 9: name: readonly variable  
New name is Prashant  
Name of the server is redhat01  
We are in path /root/script  
[root@redhat01 script]#
```



Arrays

7



ARRAYS

How to define an array?

```
myArray=( 1 2 Hello "Hey man")
```

⇒

How to get values from an array?

```
echo "${myArray[0]}"
```

```
echo "${myArray[1]}"
```



```
#!/bin/bash
```

```
#Arrays
```

```
myArray=( 1 10.5 Raju "Hey buddy" )
```

```
echo "${myArray[2]}"
```

```
echo "${myArray[3]}"
```

```
echo "All the values in my array is ${myArray[*]}"
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```



```
[root@redhat01 script]#  
[root@redhat01 script]# bash 04_arrays.sh  
Raju  
Hey buddy  
[root@redhat01 script]# vi 04_arrays.sh  
[root@redhat01 script]# bash 04_arrays.sh  
Raju  
Hey buddy  
All the values in my array is 1 10.5 Raju Hey buddy  
[root@redhat01 script]#
```



ARRAYS

How to get length of array?

```
echo "${#myArray[*]}"
```

How to get specific values?

```
echo "${myArray[*]:1}"
```

```
echo "${myArray[*]:1:2}"
```



```
#!/bin/bash
```

```
#Arrays
```

```
myArray=( 1 10.5 Raju "Hey buddy" Bow )
```

```
echo "${myArray[2]}"
```

```
echo "${myArray[3]}"
```

```
echo "All the values in my array is ${myArray[*]}"
```

```
echo "No. of values in my array is ${#myArray[*]}"
```

```
echo "Values from index 1-3 ${myArray[*]:1:3}"
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```



```
[root@redhat01 script]#  
[root@redhat01 script]# vi 04_arrays.sh  
[root@redhat01 script]# bash 04_arrays.sh  
Raju  
Hey buddy  
All the values in my array is 1 10.5 Raju Hey buddy Bow  
No. of values in my array is 5  
Values from index 1-3 10.5 Raju Hey buddy  
[root@redhat01 script]#
```



MPRASHANT

ARRAYS

How to update an array?

```
myArray+=( 5 6 8 )
```



```
#!/bin/bash
```

```
#Arrays
```

```
myArray=( 1 10.5 Raju "Hey buddy" Bow )
```

```
echo "${myArray[2]}"
```

```
echo "${myArray[3]}"
```

```
echo "All the values in my array is ${myArray[*]}"
```

```
echo "No. of values in my array is ${#myArray[*]}"
```

```
echo "Values from index 1-3 ${myArray[*]:1:3}"
```

```
myArray+=( New 10 20 Alex )
```

```
echo "All the values in my array is ${myArray[*]}"
```

```
myArray[2]=Raja
```

```
echo "All the values in my array is ${myArray[*]}"
```

```
unset myArray[2]
```

```
echo "All the values in my array is ${myArray[*]}"
```



```
[root@redhat01 script]#  
[root@redhat01 script]# vi 04_arrays.sh  
[root@redhat01 script]# bash 04_arrays.sh  
Raju  
Hey buddy  
All the values in my array is 1 10.5 Raju Hey buddy Bow  
No. of values in my array is 5  
Values from index 1-3 10.5 Raju Hey buddy  
All the values in my array is 1 10.5 Raju Hey buddy Bow New 10 20 Alex  
All the values in my array is 1 10.5 Raja Hey buddy Bow New 10 20 Alex  
All the values in my array is 1 10.5 Hey buddy Bow New 10 20 Alex  
[root@redhat01 script]#
```



ARRAYS

KEY-VALUE

```
declare -A myArray
```

```
myArray=( [name]=Paul [age]=20 )
```

```
echo "${myArray[name]}"
```



```
#!/bin/bash
```

```
declare -A myarray
```

```
myarray=( [name]=Paul [city]=London [age]=30 )
```

```
echo "${myarray['city']}"
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```



```
[root@redhat01 script]#  
[root@redhat01 script]# vi 05_key-value.sh  
[root@redhat01 script]# bash 05_key-value.sh  
Paul  
[root@redhat01 script]# vi 05_key-value.sh  
[root@redhat01 script]# bash 05_key-value.sh  
London  
[root@redhat01 script]#
```



String Operations



MPRASHANT

STRING OPERATIONS

```
myVar="Hello World!"
```

```
length=${#myVar}
```

```
upper=${x^^}
```

```
lower=${y,,}
```

```
replace=${myVar/World/Buddy}
```

```
slice=${myVar:6:11}
```




```
#!/bin/bash
```

```
myvar="Hey buddy, how are you??"
```

```
echo "Length of the value is ${#myvar}"
```

```
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~
```



5,40



```
[root@redhat01 script]#  
[root@redhat01 script]# vi 06_string_ops.sh  
[root@redhat01 script]# bash 06_string_ops.sh  
Length of the value is 24  
[root@redhat01 script]#
```



```
#!/bin/bash
```

```
myvar="Hey buddy, how are you??"
```

```
echo "Length of the value is ${#myvar}"
```

```
echo "Upper case is ---- ${myvar^^}"
```

```
echo "Lower case is ---- ${myvar,,}"
```

```
echo "${myvar/buddy/Prashant}"
```

```
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~
```





```
[root@redhat01 script]#  
[root@redhat01 script]# vi 06_string_ops.sh  
[root@redhat01 script]# bash 06_string_ops.sh  
Length of the value is 24  
Upper case is ---- HEY BUDDY, HOW ARE YOU??  
Lower case is ---- hey buddy, how are you??  
Hey Prashant, how are you??  
[root@redhat01 script]#
```





```
#!/bin/bash
```

```
myvar="Hey buddy, how are you??"
```

```
echo "Length of the value is ${#myvar}"
```

```
echo "Upper case is ---- ${myvar^^}"
```

```
echo "Lower case is ---- ${myvar,,}"
```

```
echo "${myvar/buddy/Prashant}"
```

```
echo "${myvar:4:5}"
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
^[
```





```
[root@redhat01 script]#  
[root@redhat01 script]# vi 06_string_ops.sh  
[root@redhat01 script]# bash 06_string_ops.sh  
Length of the value is 24  
Upper case is ---- HEY BUDDY, HOW ARE YOU??  
Lower case is ---- hey buddy, how are you??  
Hey Prashant, how are you??  
[root@redhat01 script]#  
[root@redhat01 script]# vi 06_string_ops.sh  
[root@redhat01 script]# bash 06_string_ops.sh  
Length of the value is 24  
Upper case is ---- HEY BUDDY, HOW ARE YOU??  
Lower case is ---- hey buddy, how are you??  
Hey Prashant, how are you??  
buddy  
[root@redhat01 script]#
```



Taking User Input

```
#!/bin/bash
```

```
read -p "Please enter your name " name
```

```
echo "Welcome $name"
```

~
~
~
~
~
~
~
~
~
~
~
~
~
~
~





```
[root@redhat01 script]#  
[root@redhat01 script]# vi 07_user_input.sh  
[root@redhat01 script]# bash 07_user_input.sh  
Please enter your name Prashant  
Welcome Prashant  
[root@redhat01 script]# █
```



Arithmetic Operations





```
#!/bin/bash
```

```
x=10
```

```
y=20
```

```
sum=$((x+y))
```

```
echo "Sum of two numbers are $sum"
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

8,33





```
[root@redhat01 script]#  
[root@redhat01 script]# vi 08_airth_ops.sh  
[root@redhat01 script]#  
[root@redhat01 script]# bash 08_airth_ops.sh  
[root@redhat01 script]# vi 08_airth_ops.sh  
[root@redhat01 script]#  
[root@redhat01 script]# bash 08_airth_ops.sh  
Sum of two numbers are 10+20  
[root@redhat01 script]# █
```



```
#!/bin/bash
```

```
x=10
```

```
y=20
```

```
let sum=$x+$y
```

```
echo "Sum of two numbers are $sum"
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

7,5





```
[root@redhat01 script]#  
[root@redhat01 script]# vi 08_airth_ops.sh  
[root@redhat01 script]#  
[root@redhat01 script]# bash 08_airth_ops.sh  
Sum of two numbers are 30  
[root@redhat01 script]#
```

I



HOW TO USE EXPRESSIONS

Using `let` command

```
let a++
```

```
let a=5*10
```

```
((a++))
```

```
((a=5*10))
```





```
#!/bin/bash
```

```
x=10
```

```
y=20
```

```
let sum=$x+$y
```

```
echo "Sum of two numbers are $sum"
```

```
echo "Mul of two numbers are $((($x*$y)))"
```

```
~  
~  
~  
~  
~  
~  
~  
~  
~  
~
```

10,38





```
[root@redhat01 script]#  
[root@redhat01 script]# vi 08_airth_ops.sh  
[root@redhat01 script]# bash 08_airth_ops.sh  
Sum of two numbers are 30  
Mul of two numbers are 200  
[root@redhat01 script]#
```

I



Conditional Statement

If-Else

MPRASHANT

IF-ELSE

```
if [ $marks -gt 40 ]
then
    echo "You are PASS"
else
    echo "You are FAIL"
fi
```



```
#!/bin/bash
```

```
read -p "Enter your total marks: " marks
```

```
if [ $marks -gt 40 ]
```

```
then
```

```
    echo "You are Passed!!!"
```

```
else
```

```
    echo "You are Failed!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
```

```
fi
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

10,3



```
[root@redhat01 script]#  
[root@redhat01 script]# ls  
01_basic.sh      03_vardemo.sh   05_key-value.sh  07_user_input.sh  
02_comments.sh  04_arrays.sh    06_string_ops.sh 08_arith_ops.sh  
[root@redhat01 script]#  
[root@redhat01 script]# vi 09_if-else.sh  
[root@redhat01 script]# bash 09_if-else.sh  
Enter your total marks: 50  
You are Passed!!!  
[root@redhat01 script]#  
[root@redhat01 script]# bash 09_if-else.sh  
Enter your total marks: 39  
You are Failed!!!!!!!!!!!!!!!!!!!!  
[root@redhat01 script]# vi 09_if-else.sh  
[root@redhat01 script]# █
```



```
#!/bin/bash
```

```
read -p "Enter your total marks: " marks
```

```
if [ $marks -gt 40 ]
```

```
then
```

```
    echo "You are Passed!!!"
```

```
    echo "Second line"
```

```
else
```

```
    echo "You are Failed!!!!!!!!!!!!!!!!!!!!!!"
```

```
fi
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

8,19-2



```
[root@redhat01 script]#  
[root@redhat01 script]# ls  
01_basic.sh      03_vardemo.sh  05_key-value.sh  07_user_input.sh  
02_comments.sh  04_arrays.sh   06_string_ops.sh 08_airth_ops.sh  
[root@redhat01 script]#  
[root@redhat01 script]# vi 09_if-else.sh  
[root@redhat01 script]# bash 09_if-else.sh  
Enter your total marks: 50  
You are Passed!!!  
[root@redhat01 script]#  
[root@redhat01 script]# bash 09_if-else.sh  
Enter your total marks: 39  
You are Failed!!!!!!!!!!!!!!!!!!!!  
[root@redhat01 script]# vi 09_if-else.sh  
[root@redhat01 script]# bash 09_if-else.sh  
Enter your total marks: 50  
You are Passed!!!  
Second line  
[root@redhat01 script]#
```



OPERATORS

Equal	-eq / ==
Greaterthanorequalto	-ge
Lessthanorequalto	-le
Not Equal	-ne / !=
Greater Than	-gt
Less Than	-lt



MPRASHANT



ELIF

```
if [ $marks -ge 80 ]
then
    echo "First Division"
elif [ $marks -ge 60 ]
then
    echo "Second Division"
else
    echo "Fail"
fi
```



```
#!/bin/bash
```

```
read -p "Enter your total marks: " marks
```

```
if [ $marks -ge 80 ]
then
    echo "A grade"
elif [ $marks -ge 60 ]
then
    echo "B Grade"
else
    echo "Failed"
fi
```

```
~
~
~
~
~
~
~
~
~
```

14, 1



```
[root@redhat01 script]#  
[root@redhat01 script]# ls  
01_basic.sh      03_vardemo.sh   05_key-value.sh  07_user_input.sh  09_if-else.sh  
02_comments.sh  04_arrays.sh    06_string_ops.sh 08_arith_ops.sh  
[root@redhat01 script]#  
[root@redhat01 script]# vi 10_elif.sh  
[root@redhat01 script]# bash 10_elif.sh  
Enter your total marks: 85  
A grade  
[root@redhat01 script]# bash 10_elif.sh  
Enter your total marks: 80  
A grade  
[root@redhat01 script]# bash 10_elif.sh  
Enter your total marks: 79  
B Grade  
[root@redhat01 script]# bash 10_elif.sh  
Enter your total marks: 50
```



CASE

```
echo "Hey choose an option"
echo "a = To see the current date"
echo "b = list all the files in current dir"

read choice

case $choice in
    a) date;;
    b) ls;;
    *) echo "Non a valid input"
esac
```



```
#!/bin/bash
```

```
echo "Hey Choose an option"  
echo "a = To print current date, time"  
echo "b = To print current working directory"  
echo "c = To list all the files in current directory"
```

```
read choice
```

```
case $choice in  
    a) date;;  
    b) pwd;;  
    c) ls;;  
    *) echo "You have provided invalid input"
```

```
esac
```

```
~  
~  
~  
~  
~
```

```
^[
```



```
[root@redhat01 script]#  
[root@redhat01 script]# bash 11_casedemo.sh  
Hey Choose an option  
a = To print current date, time  
b = To print current working directory  
c = To list all the files in current directory  
a  
Wednesday 08 November 2023 11:26:23 PM EET  
[root@redhat01 script]#  
[root@redhat01 script]# bash 11_casedemo.sh  
Hey Choose an option  
a = To print current date, time  
b = To print current working directory  
c = To list all the files in current directory  
b  
/root/script  
[root@redhat01 script]#
```





```
#!/bin/bash
```

```
echo "Hey Choose an option"  
echo "a = To print current date, time"  
echo "b = To print current working directory"  
echo "c = To list all the files in current directory"
```

```
read choice
```

```
case $choice in  
    a)  
        date  
        hostname  
        ;;  
    b) pwd;;  
    c) ls;;  
    *) echo "You have provided invalid input"
```

```
esac
```

```
~
```

```
~
```

13, 11 -



```
[root@redhat01 script]#  
[root@redhat01 script]# bash 11_casedemo.sh  
Hey Choose an option  
a = To print current date, time  
b = To print current working directory  
c = To list all the files in current directory  
1  
You have provided invalid input  
[root@redhat01 script]# vi 11_casedemo.sh  
[root@redhat01 script]# bash 11_casedemo.sh  
Hey Choose an option  
a = To print current date, time  
b = To print current working directory  
c = To list all the files in current directory  
a  
Wednesday 08 November 2023 11:28:35 PM EET  
redhat01  
[root@redhat01 script]#
```



Logical Operators

&& ||

MPRASHANT

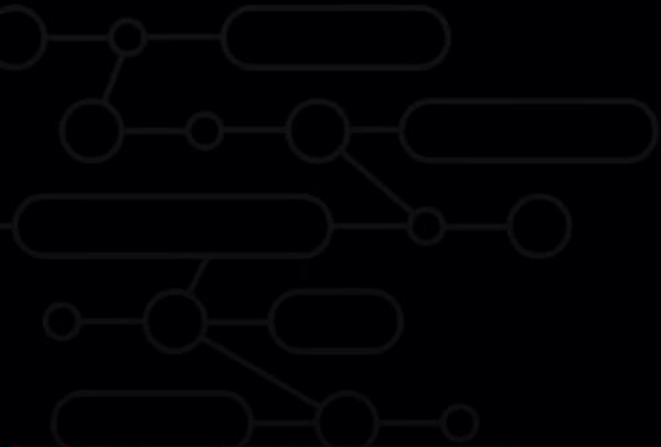
LOGICAL OPERATORS

```
condition1 && condition2
```

If both conditions are true then true else false

```
condition1 || condition2
```

If any of the condition is true then true



```
#!/bin/bash
```

```
echo "Check if you are eligible for work"
```

```
read -p "What is your age? " age
```

```
if [ $age -gt 18 ] && [ $age -le 60 ]  
then  
    echo "You can work"  
else  
    echo "Sorry you are not eligible for work"  
fi
```

```
~  
~  
~  
~  
~  
~  
~  
~
```



```
[root@redhat01 script]#  
[root@redhat01 script]# vi 12_logical_ops.sh  
[root@redhat01 script]# bash 12_logical_ops.sh  
Check if you are eligible for work  
What is your age? 15  
Sorry you are not eligible for work  
[root@redhat01 script]# bash 12_logical_ops.sh  
Check if you are eligible for work  
What is your age? 25  
You can work  
[root@redhat01 script]# bash 12_logical_ops.sh  
Check if you are eligible for work  
What is your age? 61  
Sorry you are not eligible for work  
[root@redhat01 script]#
```



LOGICAL
OPERATORS

`condition1 && condition2 || condition3`



Execute `condition2` only when `1` is true
else execute `condtion3`



```
#!/bin/bash
```

```
read -p "Enter your age: " age
```

```
[ $age -ge 18 ] && echo "Adult" || echo "Minor"
```

```
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~
```



```
[root@redhat01 script]#  
[root@redhat01 script]#  
[root@redhat01 script]# vi 14_ternary_ops.sh  
[root@redhat01 script]# vi 14_ternary_ops.sh  
[root@redhat01 script]#  
[root@redhat01 script]# bash 14_ternary_ops.sh  
Enter your age: 18  
Adult  
[root@redhat01 script]# bash 14_ternary_ops.sh  
Enter your age: 17  
Minor  
[root@redhat01 script]# █
```



Loops

For Loop

MPRASHANT

FOR LOOP

```
for i in 1 2 3 4 5
do
    echo "Number is $i"
done
```

Other ways to write For loop

```
for j in Raju Sham Baburao
for p in {1..20}
```



```
#!/bin/bash
```

```
for num in 10 20 30 40 50
do
    echo "Number is $num"
done
```

```
█
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

7,1



```
[root@redhat01 script]#  
[root@redhat01 script]# ls  
01_basic.sh          05_key-value.sh      09_if-else.sh        13_logical_or.sh  
02_comments.sh      06_string_ops.sh     10_elif.sh           14_ternary_ops.sh  
03_var_demo.sh       07_user_input.sh     11_casedemo.sh  
04_arrays.sh         08_arith_ops.sh      12_logical_ops.sh  
[root@redhat01 script]#  
[root@redhat01 script]# vi 15_forloop.sh  
[root@redhat01 script]# bash 15_forloop.sh  
Number is 10  
Number is 20  
Number is 30  
Number is 40  
Number is 50  
[root@redhat01 script]#
```



```
#!/bin/bash
```

```
for num in 10 20 30 40 50  
do
```

```
    echo "Number is $num"
```

```
done
```

```
for name in Raju Sham Paul Alex  
do
```

```
    echo "Name is $name"
```

```
done
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
^[
```



```
[root@redhat01 script]#  
[root@redhat01 script]# vi 15_forloop.sh  
[root@redhat01 script]# bash 15_forloop.sh  
Number is 10  
Number is 20  
Number is 30  
Number is 40  
Number is 50  
Name is Raju  
Name is Sham  
Name is Paul  
Name is Alex  
[root@redhat01 script]#
```

I



```
#!/bin/bash
```

```
for num in 10 20 30 40 50
do
    echo "Number is $num"
done
```

```
for name in Raju Sham Paul Alex
do
    echo "Name is $name"
done
```

```
for i in {1..10}
do
    echo "$i"
done
```

```
~
```

17,4



```
Number is 20  
Number is 30  
Number is 40  
Number is 50  
Name is Raju  
Name is Sham  
Name is Paul  
Name is Alex
```

```
1
```

```
2
```

```
3
```

```
4
```

```
5
```

```
6
```

```
7
```

```
8
```

```
9
```

```
10
```

```
[root@redhat01 script]#
```



MPRASHANT

ITERATE VALUES FROM FILE..

```
items="/home/paul/file.txt"
```

```
for item in $(cat $items)
do
    echo $item
done
```




```
[root@redhat01 script]#  
[root@redhat01 script]# vi servers.txt
```



```
server1  
server2  
serverqa  
serverDEV  
serverPROD
```

```
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~
```

```
5,11
```



```
#!/bin/bash
```

```
FILE="/root/script/servers.txt"
```

```
for server in $(cat $FILE)
do
    echo "Server is $server"
done
```

```
█
~
~
~
~
~
~
~
~
~
~
```

10,1



```
[root@redhat01 script]#  
[root@redhat01 script]# vi servers.txt  
[root@redhat01 script]#  
[root@redhat01 script]# vi 16_for_with_file.sh  
[root@redhat01 script]#  
[root@redhat01 script]# pwd  
/root/script  
[root@redhat01 script]# vi 16_for_with_file.sh  
[root@redhat01 script]# bash 16_for_with_file.sh  
Server is server1  
Server is server2  
Server is serverqa  
Server is serverDEV  
Server is serverPROD  
[root@redhat01 script]# █
```



For Loop With an Array

```
#!/bin/bash
```

```
#Arrays
```

```
myArray=( 1 10.5 Raju "Hey buddy" Bow )
```

```
for value in ${myArray[*]}
```

```
do
```

```
    echo "Value is $value"
```

```
done
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

6,24



```
[root@redhat01 script]#  
[root@redhat01 script]# vi 17_for_with_array.sh  
[root@redhat01 script]# bash 17_for_with_array.sh  
Value is 1  
[root@redhat01 script]# vi 17_for_with_array.sh  
[root@redhat01 script]# bash 17_for_with_array.sh  
Value is 1  
Value is 10.5  
Value is Raju  
Value is Hey  
Value is buddy  
Value is Bow  
[root@redhat01 script]# █
```



MPRASHANT

WHILE LOOP

```
count=0
```

```
num=10
```

```
while [ $count -le $num ]  
do
```

```
    echo "Numbers are $count"
```

```
    let count++
```

```
done
```




```
#!/bin/bash
```

```
count=0
```

```
num=10
```

```
while [ $count -le $num ]
```

```
do
```

```
    echo "Value of count is $count"  
    ((count++))
```

```
done
```

```
~  
~  
~  
~  
~  
~  
~
```

```
^[
```



```
[root@redhat01 script]#  
[root@redhat01 script]# vi 18_while_demo.sh  
[root@redhat01 script]# bash 18_while_demo.sh  
Value of count is 0  
Value of count is 1  
Value of count is 2  
Value of count is 3  
Value of count is 4  
Value of count is 5  
Value of count is 6  
Value of count is 7  
Value of count is 8  
Value of count is 9  
Value of count is 10  
[root@redhat01 script]# █
```



MPRASHANT

UNTIL LOOP..



```
a=10
```

```
until [ $a -eq 1 ]
```

```
do
```

```
    echo $a
```

```
    a=`expr $a - 1`
```

```
done
```



```
#!/bin/bash
```

```
counter=10
```

```
until [ $counter -eq 1 ]
```

```
do
```

```
    echo "Counter is $counter"
```

```
    let counter--
```

```
done
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

8, 15-22



```
[root@redhat01 script]#  
[root@redhat01 script]# vi 19_until_demo.sh  
[root@redhat01 script]# bash 19_until_demo.sh  
Counter is 10  
Counter is 9  
Counter is 8  
Counter is 7  
Counter is 6  
Counter is 5  
Counter is 4  
Counter is 3  
Counter is 2  
[root@redhat01 script]#
```



Infinite Loop

MPRASHANT

INFINITE LOOP..

```
while true
do
    echo "Hi"
    sleep 2s
done
```



```
#!/bin/bash
```

```
while true  
do  
    echo "Hello user"  
    sleep 2s  
done
```

```
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~
```

6, 10-17



What are Functions

MPRASHANT

WHAT ARE FUNCTIONS?

- Block of code which perform some task and run when it is called.

- Can be reuse many times in our program which lessen our lines of code.

- We can pass arguments to the me



MPRASHANT

HOW TO MAKE FUNCTIONS?

```
function myfun {  
    echo "Hi"  
}
```

```
myfun() {  
    echo "Hello"  
}
```

To call the function
myfun



```
#!/bin/bash
```

```
function welcomeNote {  
    echo "-----"  
    echo "Hello & Welcome!!"  
    echo "-----"  
    echo  
}
```

```
welcomeNote
```



11,1



```
[root@redhat01 script]#  
[root@redhat01 script]# vi 21_function_demo.sh  
[root@redhat01 script]# bash 21_function_demo.sh  
[root@redhat01 script]# vi 21_function_demo.sh  
[root@redhat01 script]# bash 21_function_demo.sh  
-----  
Hello & Welcome!!  
-----  
[root@redhat01 script]# █
```





```
#!/bin/bash
```

```
welcomeNote() {  
    echo "-----"  
    echo "Hello & Welcome!!"  
    echo "-----"  
    echo
```

```
welcomeNote  
welcomeNote  
welcomeNote
```

```
~  
~  
~  
~  
~  
~
```

3,16





```
[root@redhat01 script]#  
[root@redhat01 script]# vi 21_function_demo.sh  
[root@redhat01 script]# bash 21_function_demo.sh
```

```
-----  
Hello & Welcome!!  
-----
```

```
-----  
Hello & Welcome!!  
-----
```

```
-----  
Hello & Welcome!!  
-----
```

```
[root@redhat01 script]#
```



MPRASHANT

HOW TO USE ARGUMENTS IN FUNCTIONS?

```
addition() {  
    local num1=$1  
    local num2=$2  
    let sum=$num1+$num2  
    echo "Sum of $num1 and $num2 is $sum"  
}
```

```
myfun 12 13
```




```
#!/bin/bash
```

```
welcomeNote() {  
    echo "-----"  
    echo "Welcome $1"  
    echo "-----"  
    echo  
}
```

```
read -p "Enter your name: " name
```

```
welcomeNote $name
```

```
~  
~  
~  
~  
~  
~  
~
```

```
^[
```



```
[root@redhat01 script]#  
[root@redhat01 script]#  
[root@redhat01 script]# vi 22_function_with_args.sh  
[root@redhat01 script]# bash 22_function_with_args.sh  
Enter your name: Paul  
-----  
Welcome Paul  
-----  
  
[root@redhat01 script]# bash 22_function_with_args.sh  
Enter your name: Alex  
-----  
Welcome Alex  
-----  
  
[root@redhat01 script]# █
```



Argument Passing in Shell Script

MPRASHANT

HOW TO USE ARGUMENTS IN FUNCTIONS?

```
addition() {  
    local num1=$1  
    local num2=$2  
    let sum=$num1+$num2  
    echo "Sum of $num1 and $num2 is $sum"  
}
```

```
myfun 12 13
```



ARGUMENTS IN SCRIPT...

```
#myscript.sh arg1 arg2..
```



How to access these arguments inside our script?

To get no. of arguments : \$#

To display all arguments : \$@

To use or display a argument: \$1 \$2 ..



```
#!/bin/bash
```

```
#Accessing the arguments
```

```
echo "First argument is $1"
```

```
echo "Second argument is $2"
```

```
echo "No. of arguments are $#"
```

```
echo "All the arguments are $@"
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

8,32





```
[root@redhat01 script]#  
[root@redhat01 script]# vi 23 args.sh  
[root@redhat01 script]# bash 23 args.sh server1 server2 server3  
First argument is server1  
Second argument is server2  
No. of arguments are 3  
All the arguments are server1 server2 server3  
[root@redhat01 script]#
```



```
#!/bin/bash
```

```
#Accessing the arguments
```

```
echo "First argument is $1"
```

```
echo "Second argument is $2"
```

```
echo "No. of arguments are $#"
```

```
echo "All the arguments are $@"
```

```
for arg in $@
```

```
do
```

```
    echo "Argumet is $arg"
```

```
done
```

```
~  
~  
~  
~  
~
```

12, 22-





```
[root@redhat01 script]#  
[root@redhat01 script]# vi 23_args.sh  
[root@redhat01 script]# bash 23_args.sh server1 server2 server3 server4  
First argument is server1  
Second argument is server2  
No. of arguments are 4  
All the arguments are server1 server2 server3 server4  
Argument is server1  
Argument is server2  
Argument is server3  
Argument is server4  
[root@redhat01 script]#
```



```
#!/bin/bash
```

```
if [ $# -eq 0 ]  
then
```

```
    echo "Kindly provide atleast one argument"  
    exit 1
```

```
fi
```

```
#Accessing the arguments
```

```
echo "First argument is $1"  
echo "Second argument is $2"
```

```
echo "No. of arguments are $#"  
echo "All the arguments are $@"
```

```
for arg in $@  
do  
    echo "Argumet is $arg"
```

6,7-14





```
[root@redhat01 script]#  
[root@redhat01 script]# vi 23_args.sh  
[root@redhat01 script]#  
[root@redhat01 script]# bash 23_args.sh  
Kindly provide atleast one argument  
[root@redhat01 script]# █
```



Other Useful Concepts

MPRASHANT

USEFUL
CONCEPTS...

break - to stop the loop

continue - to stop current iteration
of loop and start next iteration



```
#!/bin/bash
```

```
#example os break in a loop
```

```
num=6
```

```
for i in 1 2 3 4 5 6 7 8 9  
do
```

```
    #break the loop if num found
```

```
    if [ $num -eq $i ]
```

```
    then
```

```
        echo "$num is found"
```

```
        break
```

```
    fi
```

```
    echo "Number is $i"
```

```
done
```

```
~
```

```
~
```

```
"24_break.sh" 16L, 197B
```

```
15,19-2
```



```
[root@redhat01 script]#  
[root@redhat01 script]# vi 24_break.sh  
[root@redhat01 script]# bash 24_break.sh  
Number is 1  
Number is 2  
Number is 3  
Number is 4  
Number is 5  
6 is found  
[root@redhat01 script]#
```

I



USEFUL CONCEPTS...

`sleep` – to create delay between two executions ex: `sleep 1s/1m`

`exit` – to stop script at a point

exit status `$?` – gives you status of previous command if that was successful




```
[root@redhat01 script]#  
[root@redhat01 script]# vi 23_args.sh  
[root@redhat01 script]#  
[root@redhat01 script]# pwd  
/root/script  
[root@redhat01 script]# echo $?  
0  
[root@redhat01 script]# █
```



```
[root@redhat01 script]#  
[root@redhat01 script]# vi 23_args.sh  
[root@redhat01 script]#  
[root@redhat01 script]# pwd  
/root/script  
[root@redhat01 script]# echo $?  
0  
[root@redhat01 script]# rm ABC.txt  
rm: cannot remove 'ABC.txt': No such file or directory  
[root@redhat01 script]#
```



```
[root@redhat01 script]#  
[root@redhat01 script]# vi 23_args.sh  
[root@redhat01 script]#  
[root@redhat01 script]# pwd  
/root/script  
[root@redhat01 script]# echo $?  
0  
[root@redhat01 script]# rm ABC.txt  
rm: cannot remove 'ABC.txt': No such file or directory  
[root@redhat01 script]#  
[root@redhat01 script]# echo $?  
1  
[root@redhat01 script]#
```

I



Script For Connectivity Check

```
[root@redhat01 script]#  
[root@redhat01 script]# ping -c 1 www.google.com  
PING www.google.com (142.250.180.196) 56(84) bytes of data.  
64 bytes from bud02s33-in-f4.1e100.net (142.250.180.196): icmp_seq=1  
ttl=128 time=18.8 ms  
  
--- www.google.com ping statistics ---  
1 packets transmitted, 1 received, 0% packet loss, time 0ms  
rtt min/avg/max/mdev = 18.827/18.827/18.827/0.000 ms  
[root@redhat01 script]#  
[root@redhat01 script]# █
```



```
[root@redhat01 script]#  
[root@redhat01 script]# ping -c 1 www.google.com  
PING www.google.com (142.250.180.196) 56(84) bytes of data.  
64 bytes from bud02s33-in-f4.1e100.net (142.250.180.196): icmp_seq=1  
ttl=128 time=18.8 ms
```

```
--- www.google.com ping statistics ---  
1 packets transmitted, 1 received, 0% packet loss, time 0ms  
rtt min/avg/max/mdev = 18.827/18.827/18.827/0.000 ms
```

```
[root@redhat01 script]#  
[root@redhat01 script]# echo $?  
0
```

```
[root@redhat01 script]# ping -c 1 localhost.com  
PING localhost.com (74.125.224.72) 56(84) bytes of data.
```

```
--- localhost.com ping statistics ---  
1 packets transmitted, 0 received, 100% packet loss, time 0
```

```
[root@redhat01 script]# █
```



```
#!/bin/bash
```

```
read -p "Which site you want to check? " site
```

```
ping -c 1 $site
```

```
if [ $? -eq 0 ]
```

```
then
```

```
    echo "Connection was successful!!"
```

```
else
```

```
    echo "Failed to connect"
```

```
fi
```

```
~  
~  
~  
~
```

14,1



```
[root@redhat01 script]#  
[root@redhat01 script]# vi 25_connectivity_check.sh  
[root@redhat01 script]# bash 25_connectivity_check.sh  
Which site you want to check? www.google.com  
PING www.google.com (142.251.208.100) 56(84) bytes of data.  
64 bytes from bud02s41-in-f4.1e100.net (142.251.208.100): icmp_seq=1  
ttl=128 time=11.1 ms  
  
--- www.google.com ping statistics ---  
1 packets transmitted, 1 received, 0% packet loss, time 0ms  
rtt min/avg/max/mdev = 11.058/11.058/11.058/0.000 ms  
Connection was successful!!  
[root@redhat01 script]#
```




```
[root@redhat01 script]#
```

```
[root@redhat01 script]# bash 25_connectivity_check.sh
```

```
Which site you want to check? localhost.com
```

```
PING localhost.com (74.125.224.72) 56(84) bytes of data.
```

```
--- localhost.com ping statistics ---
```

```
1 packets transmitted, 0 received, 100% packet loss, time 0ms
```

```
Failed to connect
```

```
[root@redhat01 script]#
```



MPRASHANT



USEFUL CONCEPTS...

basename - strip directory info and only give filename

dirname - strip the filename and gives directory path

realpath - gives you full path for a file





```
[root@redhat01 script]#  
[root@redhat01 script]# basename /root/script/01_basic.sh  
01_basic.sh  
[root@redhat01 script]#  
[root@redhat01 script]# dirname /root/script/01_basic.sh  
/root/script  
[root@redhat01 script]# █
```



```
[root@redhat01 script]#  
[root@redhat01 script]#  
[root@redhat01 script]# realpath 01_basic.sh  
/root/script/01_basic.sh  
[root@redhat01 script]#
```

I



Check Existance Of Files or Directory

MPRASHANT

CHECK IF FILE/DIR EXIST

`if [-d folder_name]` If folder exists

`[! -d folder_name]` If folder not exists

`if [-f file_name]` If file exists

`if [! -f file_name]` If file not exists



```
#!/bin/bash
```

```
FILE="/root/script/servers.txt"
```

```
if [ ! -f $FILE ]  
then
```

```
    echo "File is not found!!"  
    exit 1
```

```
fi
```

```
for server in $(cat $FILE)  
do
```

```
    echo "Server is $server"
```

```
done
```

```
~  
~
```

8,8-15



MPRASHANT

BASH VARIABLES...

RANDOM – A random integer between 0 and 32767 is generated

UID – User ID of the user logged in




```
[root@redhat01 script]#  
[root@redhat01 script]# echo $RANDOM  
3248  
[root@redhat01 script]# echo $RANDOM  
11709  
[root@redhat01 script]# echo $RANDOM  
28522  
[root@redhat01 script]# echo $RANDOM  
28909  
[root@redhat01 script]# echo $RANDOM  
17638  
[root@redhat01 script]# echo $RANDOM  
5773  
[root@redhat01 script]# █
```





```
[root@redhat01 script]#  
[root@redhat01 script]# vi 26_dice_game.sh  
[root@redhat01 script]# bash 26_dice_game.sh  
Number is 1  
[root@redhat01 script]# bash 26_dice_game.sh  
Number is 2  
[root@redhat01 script]# bash 26_dice_game.sh  
Number is 5  
[root@redhat01 script]# bash 26_dice_game.sh  
Number is 5  
[root@redhat01 script]# bash 26_dice_game.sh  
Number is 2  
[root@redhat01 script]# bash 26_dice_game.sh  
Number is 4  
[root@redhat01 script]# bash 26_dice_game.sh  
Number is 1  
[root@redhat01 script]# bash 26_dice_game.sh  
Number is 5  
[root@redhat01 script]# █
```



```
[root@redhat01 script]#
```

```
[root@redhat01 script]# echo $UID
```

```
0
```

```
[root@redhat01 script]# █
```





```
[root@redhat01 script]#  
[root@redhat01 script]# echo $UID  
0  
[root@redhat01 script]# su paul  
[paul@redhat01 script]$  
[paul@redhat01 script]$ echo $UID  
1000  
[paul@redhat01 script]$
```



Redirection in scripts

> >>





```
[root@redhat01 script]#  
[root@redhat01 script]# pwd  
/root/script  
[root@redhat01 script]#  
[root@redhat01 script]# pwd > sample.txt  
[root@redhat01 script]# cat sample.txt  
/root/script  
[root@redhat01 script]#
```



```
[root@redhat01 script]#  
[root@redhat01 script]# pwd  
/root/script  
[root@redhat01 script]#  
[root@redhat01 script]# pwd > sample.txt  
[root@redhat01 script]# cat sample.txt  
/root/script  
[root@redhat01 script]# hostname > sample.txt  
[root@redhat01 script]#  
[root@redhat01 script]# cat sample.txt  
redhat01  
[root@redhat01 script]#
```




```
[root@redhat01 script]#  
[root@redhat01 script]# pwd  
/root/script  
[root@redhat01 script]#  
[root@redhat01 script]# pwd > sample.txt  
[root@redhat01 script]# cat sample.txt  
/root/script  
[root@redhat01 script]# hostname > sample.txt  
[root@redhat01 script]#  
[root@redhat01 script]# cat sample.txt  
redhat01  
[root@redhat01 script]# pwd >> sample.txt  
[root@redhat01 script]#  
[root@redhat01 script]# cat sample.txt  
redhat01  
/root/script  
[root@redhat01 script]# █
```



MPRASHANT

In case if you don't wanna print the
output of a command on terminal or
write in a file,

WHAT IS /DEV/NULL

We can redirect the output to
`/dev/null`

Example:

```
#cd /root &> /dev/null
```



```
#!/bin/bash
```

```
read -p "Which site you want to check? " site
```

```
ping -c 1 $site &> /dev/null
```

```
if [ $? -eq 0 ]
```

```
then
```

```
    echo "Connection was successful!!"
```

```
else
```

```
    echo "Failed to connect"
```

```
fi
```

```
~  
~  
~  
~
```

5,29





root@redhat01:~/script



```
[root@redhat01 script]#  
[root@redhat01 script]# vi 25_connectivity_check.sh  
[root@redhat01 script]# bash 25_connectivity_check.sh  
Which site you want to check? www.google.com  
Connection was successful!!  
[root@redhat01 script]# █
```



Run Script in BackGround

nohup

`nohup ./script.sh &`



```
[root@redhat01 ~]#  
[root@redhat01 ~]# cd /root/script/  
[root@redhat01 script]#  
[root@redhat01 script]# chmod 777 sample.sh  
[root@redhat01 script]#  
[root@redhat01 script]# nohup ./sample.sh &  
[1] 112512  
[root@redhat01 script]# nohup: ignoring input and appending output to 'nohup.out'  
  
[root@redhat01 script]# █
```



Automate Your Script with AT Command



```
[root@redhat01 script]#
```

```
[root@redhat01 script]#
```

```
[root@redhat01 script]# at 02:30 PM 10 November 2023
```



```
[root@redhat01 script]#  
[root@redhat01 script]#  
[root@redhat01 script]# at 02:30 PM 10 November 2023  
warning: commands will be executed using /bin/sh  
at> echo "hello"  
at> <EOT>  
job 10 at Fri Nov 10 14:30:00 2023  
[root@redhat01 script]#  
[root@redhat01 script]# atq  
10          Fri Nov 10 14:30:00 2023 a root  
[root@redhat01 script]# █
```



```
[root@redhat01 script]#  
[root@redhat01 script]#  
[root@redhat01 script]# at 02:30 PM 10 November 2023  
warning: commands will be executed using /bin/sh  
at> echo "hello"  
at> <EOT>  
job 10 at Fri Nov 10 14:30:00 2023  
[root@redhat01 script]#  
[root@redhat01 script]# atq  
10          Fri Nov 10 14:30:00 2023 a root  
[root@redhat01 script]#  
[root@redhat01 script]# atrm 10  
[root@redhat01 script]# █
```



Automate Using cronjobs

4:11

For scheduling only one time, use AT

at 12:09 PM

<your_command>

Ctrl + D

atq to check scheduled job

atrm <id> to remove the schedule



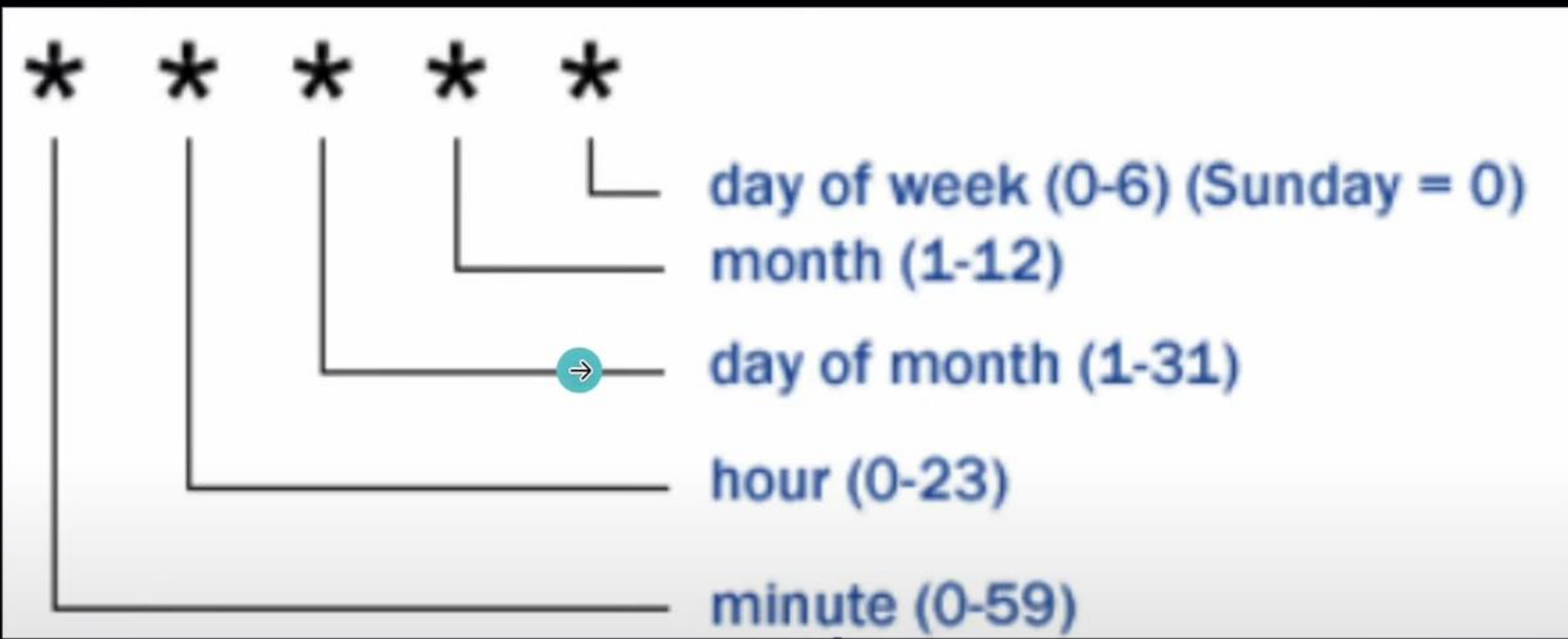
To check the existing jobs - `crontab -l`

To add new job - `crontab -e`



```
* * * * cd /home/paul/scripts && ./create_file.sh
```





MPRASHANT

1

Monitoring free RAM space



```
[paul@redhat01 projects]$  
[paul@redhat01 projects]$ ls  
[paul@redhat01 projects]$  
[paul@redhat01 projects]$ vi ram_status.sh  
[paul@redhat01 projects]$  
[paul@redhat01 projects]$ free  
              total          used          free      shared  buff/cache   availa  
ble  
Mem:           1740064      1508756          77112      163084     402132     231  
308  
Swap:           2134012      1482300          651712
```

[paul@redhat01 projects]\$ █



```
[paul@redhat01 projects]$
```

```
[paul@redhat01 projects]$ free -h
```

	total	used	free	shared	buff/cache	available
Mem:	1.7Gi	1.4Gi	77Mi	156Mi	389Mi	227Mi
Swap:	2.0Gi	1.4Gi	636Mi			

```
[paul@redhat01 projects]$ █
```



```
[paul@redhat01 projects]$
```

```
[paul@redhat01 projects]$ free -mt
```

	total	used	free	shared	buff/cache	available
Mem:	1699	1474	74	160	393	
Swap:	2083	1447	636			
Total:	3783	2921	710			

```
[paul@redhat01 projects]$
```



```
[paul@redhat01 projects]$  
[paul@redhat01 projects]$ free -mt  
                total          used          free          shared  buff/cache          availa  
ble  
Mem:            1699           1474             74           160           393  
224  
Swap:           2083           1447           636  
Total:          3783           2921           710  
[paul@redhat01 projects]$  
[paul@redhat01 projects]$ free -mt | grep "Total"  
Total:        3783           2905           727  
[paul@redhat01 projects]$ █
```

I



```
[paul@redhat01 projects]$  
[paul@redhat01 projects]$ free -mt  
total used free shared buff/cache availa  
ble  
Mem: 1699 1474 74 160 393  
224  
Swap: 2083 1447 636  
Total: 3783 2921 710  
[paul@redhat01 projects]$  
[paul@redhat01 projects]$ free -mt | grep "Total"  
Total: 3783 2905 727  
[paul@redhat01 projects]$  
[paul@redhat01 projects]$ free -mt | grep "Total" | awk '{print $4}'  
723  
[paul@redhat01 projects]$
```



```
#!/bin/bash
```

```
FREE_SPACE=$(free -mt | grep "Total" | awk '{print $4}')
```

```
TH=500
```

```
if [[ $FREE_SPACE -lt $TH ]]
```

```
then
```

```
    echo "WARNING, RAM is running low"
```

```
else
```

```
    echo "RAM Space is sufficient - $FREE_SPACE M"
```

```
fi
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
:
```



```
[paul@redhat01 projects]$
```

```
[paul@redhat01 projects]$ bash ram status.sh
```

```
RAM Space is sufficient - 715 M
```

```
[paul@redhat01 projects]$
```



2

Monitoring free DISK space and sent an Alert Email



```
[paul@redhat01 projects]$  
[paul@redhat01 projects]$ vi fs_status.sh  
[paul@redhat01 projects]$  
[paul@redhat01 projects]$ df  
Filesystem                1K-blocks      Used Available Use% Mounted on  
devtmpfs                   4096            0       4096    0% /dev  
tmpfs                      870032         7840    862192    1% /dev/shm  
tmpfs                      348016         31244   316772    9% /run  
/dev/mapper/rhel-root     42520284      5250448 37269836 13% /  
/dev/mapper/rhel-home    20756480      276064  20480416  2% /home  
/dev/sda2                 1038336       261356   776980   26% /boot  
/dev/sda1                  613160         7140   606020    2% /boot/efi  
tmpfs                     174004         140    173864    1% /run/user/1000  
[paul@redhat01 projects]$ █
```



```
[paul@redhat01 projects]$  
[paul@redhat01 projects]$ df -H  
Filesystem                Size      Used Avail Use% Mounted on  
devtmpfs                   4.2M         0   4.2M   0% /dev  
tmpfs                       891M     8.1M   883M   1% /dev/shm  
tmpfs                       357M     32M   325M   9% /run  
/dev/mapper/rhel-root      44G     5.4G    39G  13% /  
/dev/mapper/rhel-home      22G     283M    21G   2% /home  
/dev/sda2                   1.1G     268M   796M  26% /boot  
/dev/sda1                   628M     7.4M   621M   2% /boot/efi  
tmpfs                       179M     136k   179M   1% /run/user/1000  
[paul@redhat01 projects]$  
[paul@redhat01 projects]$ df -H | egrep -v "Filesystem|tmpfs"  
/dev/mapper/rhel-root      44G     5.4G    39G  13% /  
/dev/mapper/rhel-home      22G     283M    21G   2% /home  
/dev/sda2                   1.1G     268M   796M  26% /boot  
/dev/sda1                   628M     7.4M   621M   2% /boot/efi  
[paul@redhat01 projects]$ █
```



```
[paul@redhat01 projects]$  
[paul@redhat01 projects]$ df -H | egrep -v "Filesystem|tmpfs"  
/dev/mapper/rhel-root    44G  5.4G   39G   13% /  
/dev/mapper/rhel-home    22G  283M   21G    2% /home  
/dev/sda2                1.1G  268M   796M   26% /boot  
/dev/sda1                628M  7.4M   621M    2% /boot/efi  
[paul@redhat01 projects]$  
[paul@redhat01 projects]$ df -H | egrep -v "Filesystem|tmpfs" | grep "sda2"  
/dev/sda2                1.1G  268M   796M   26% /boot  
[paul@redhat01 projects]$
```

I



```
[paul@redhat01 projects]$  
[paul@redhat01 projects]$ df -H | egrep -v "Filesystem|tmpfs"  
/dev/mapper/rhel-root    44G  5.4G  39G  13% /  
/dev/mapper/rhel-home    22G  283M  21G   2% /home  
/dev/sda2                1.1G  268M  796M  26% /boot  
/dev/sda1                628M   7.4M  621M   2% /boot/efi  
[paul@redhat01 projects]$  
[paul@redhat01 projects]$ df -H | egrep -v "Filesystem|tmpfs" | grep "sda2"  
/dev/sda2                1.1G  268M  796M  26% /boot  
[paul@redhat01 projects]$  
[paul@redhat01 projects]$ df -H | egrep -v "Filesystem|tmpfs" | grep "sda2" |  
awk '{print $5}'  
26%  
[paul@redhat01 projects]$ df -H | egrep -v "Filesystem|tmpfs" | grep "sda2" |  
awk '{print $5}' |█
```



```
[paul@redhat01 projects]$  
[paul@redhat01 projects]$ df -H | egrep -v "Filesystem|tmpfs"  
/dev/mapper/rhel-root    44G  5.4G  39G  13% /  
/dev/mapper/rhel-home    22G  283M  21G   2% /home  
/dev/sda2                1.1G  268M  796M  26% /boot  
/dev/sda1                628M   7.4M  621M   2% /boot/efi  
[paul@redhat01 projects]$  
[paul@redhat01 projects]$ df -H | egrep -v "Filesystem|tmpfs" | grep "sda2"  
/dev/sda2                1.1G  268M  796M  26% /boot  
[paul@redhat01 projects]$  
[paul@redhat01 projects]$ df -H | egrep -v "Filesystem|tmpfs" | grep "sda2" |  
awk '{print $5}'  
26%  
[paul@redhat01 projects]$ df -H | egrep -v "Filesystem|tmpfs" | grep "sda2" |  
awk '{print $5}' | tr -d %  
26  
[paul@redhat01 projects]$
```



```
#!/bin/bash
```

```
#monitoring the free fs space disk
```

```
FU=$(df -H | egrep -v "Filesystem|tmpfs" | grep "sda2" | awk '{print $5}' | tr -d %)
```

```
if [[ $FU -ge 80 ]]
```

```
then
```

```
    echo "Warning, disk space is low"
```

```
else
```

```
    echo "All good"
```

```
█
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
-- INSERT --
```

```
11,2-
```



```
[paul@redhat01 projects]$  
[paul@redhat01 projects]$ vi fs_status.sh  
[paul@redhat01 projects]$ vi fs_status.sh  
[paul@redhat01 projects]$ bash fs_status.sh  
All good  
[paul@redhat01 projects]$  
[paul@redhat01 projects]$ █
```

