**SHELL SCRIPT**

***Variables*  
##Variable**

Name="Sushant"

Mobile=55565656

echo" $name $Mobile"

**##type=2 to store command output in variable**

HOSTNAME= $(hostname)

present\_working\_D=$(pwd)

DATE=$(date)

echo "system name is $HOSTNAME , present working dir is $present\_working\_D , Todays date and time is $DATE"

**##type=3 define constant variable will not be override (FIXED)**

readonly Company="Audi"

echo "company name is = $Company "

**#type=4 Array**

#!/bin/bash

#Array

My\_Array=(100 200 300 Apple Oneplus "Nice cool")

echo "${My\_Array[0]}"

echo "${My\_Array[3]}"

#To print all array

echo "values of all array ${My\_Array[\*]}"

#To print lenth of array

echo "The Complte length of arry is:: ${#My\_Array[\*]}"

#To print specfic values of arry

#I want to print values from Index number 2 and after that I need 2 values from Array

echo "to print 2-3 index value= ${My\_Array[\*]:2:3}"

echo "to print specific values from index= ${My\_Array[\*]:2}"

echo "to print all number after index 2 ${My\_Array[@]:2}"

#Updating/adding new values in arry

My\_Array+=( New 999 888 77)

#Print array details including all info

echo "printing array including newaly added values= ${My\_Array[\*]}"

**Output:**

100

Apple

Nice cool

values of all array 100 200 300 Apple Oneplus Nice cool

The Complte length of arry is:: 6

to print 2-3 index value= 300 Apple Oneplus

to print specific values from index= 300 Apple Oneplus Nice cool

to print all number after index 2 300 Apple Oneplus Nice cool

printing array including newaly added values= 100 200 300 Apple Oneplus Nice cool New 999 888 77

**#Type=3 Store key value data in variable**

#!/bin/bash

#How to store key value pairs

declare -A My\_Array

My\_Array=([name]=Sushant [age]=27 [hobby]=tt )

echo "name of winner is ${My\_Array[name]}"

echo "hobby of winner is ${My\_Array[hobby]}"

echo "age is winner is ${My\_Array[age]}"

**Output=**

name of winner is Sushant

hobby of winner is tt

age is winner is 27

***#String Operation***

Vi string\_ops.sh

#!/bin/bash

#string operation

My\_Var="Hello Buddy!!! Vishay ahe bhava"

#TO print length of string

My\_var\_length=${#My\_Var}

echo "length of variable is == $My\_var\_length"

#To do Uppercase and lowercase

echo "string in uppercase---${My\_Var^^}"

echo "string in lowercase---${My\_Var,,}"

#To replace a string

newstring=${My\_Var/Hello/Hi}

echo "new string---$newstring"

#To slice a string

echo "printing slice of string ${My\_Var:5:7}"

**Output:**

length of variable is == 31

string in uppercase---HELLO BUDDY!!! VISHAY AHE BHAVA

string in lowercase---hello buddy!!! vishay ahe bhava

new string---Hi Buddy!!! Vishay ahe bhava

printing slice of string Buddy!

***#User\_interaction***

#!/bin/bash

#take input value form user

#echo "what is your name buddy" #curser in new line

read -p "What is your name buddy?" name #curser in same line

#read name

echo "user name is $name"

***#Arithmetic expressions***

**1.Using let keyword**

**2.Using double bracket $(())**

#!/bin/bash

#Math calculation

x=10

y=20

#Multiply

let mul=$x\*$y

echo "multiplication of x and y is : $mul"

#Add

let add=$x+$y

echo "addition of x and y is : $add"

#Sub

echo "substaraction of x and y is : $(($y-$x))"

***IF-ELSE***

if [[ $age -gt 20 ]]

then

echo "you are Adult...."

else

echo "you are Kid...."

fi

***EL-IF***

#!/bin/bash

#To get input from the user check grade of student

read -p "insert your SSC marks %: " marks

if [[ $marks -gt 100 ]]

then

echo "ERROR...Please insert correct %"

elif [[ $marks -ge 80 ]]

then

echo "you passed with DIST class"

elif [[ $marks -ge 60 ]]

then

echo "you passed with First class........................."

elif [[ $marks -ge 40 ]]

then

echo "second division"

else

echo "FAILED................."

Fi

***CASE STATEMENT***

#To avoid multiple elif we can use CASE function/class

!#/bin/bash

echo "press a for to get current date and time"

echo "press b for to get list of scripts"

echo "press c for to get present working directory"

echo "press d to check disk space utilization"

read choice

case $choice in

a) echo "Todays date is: "

date

echo "Thank you"

;; #for end

b)ls -ltr;;

c)pwd;;

d)df -Th;;

\*) echo "ERROR....Please provide valid input"

Esac

***LOGICAL\_OPERATORS***

**Condition1 && Condition2 ⇒ if both conditions are true then else false.**

**Condition1 || Condition2 ⇒If any of condition is true then then true.**

**1. && AND confition**

!#/bin/bash

#Taking input form user

read -p "insert your age? " age

read -p "insert your country name? " country

if [[ $age -ge 18 ]] && [[ $country == "India" ]] #== for string -geor -gt for int/num

then

echo "YUP...you can able to vote India."

else

echo "Better luck next time. you can't vote in India."

fi

**2. | | OR condition**

!#/bin/bash

#Taking input form user

read -p "insert your country name..?" country

if [[ $country == India ]] || [[ $country == Nepal ]]

then

echo "you can travel in both country without passport"

else

echo "please make your passport"

Fi

**3. && and || AND OR operator**

!#/bin/bash

read -p "insert your age..?" age

read -p "insert country name ..?" country

if [[ $age -gt 18 ]] && [[ $age -lt 60 ]] || [[ $country == India ]]

then

echo "you are eligible"

else

echo " SORRY............TRY after 18+"

fi

***LOOPS***

***1.FOR LOOP:It is used to go through each item in a list or a number range, one by one.***

#!/bin/bash

**EX:A**

for i in {1..10} #way 1

for i in 1 2 3 4 5 6 7 8 9 #way2

do

echo "number is $i"

done

#!/bin/bash

for (( i=1; i<=5; i++ )) #way3

do

echo "Index: $i"

done

**EX:B:with string**

for name in APPLE SAMSUNG OPPO VIVO ONEPLUS REALME NOKIA

do

echo "Mobile popular companies $name"

done

**EX:4#Iterate values from file**

FILE="/root/linux/shell-script/names18.txt"

for name in $(cat $FILE)

do

echo "students names are:: $name"

done

**EX:5#store values in Array**

My\_Array=( laptop phone tab earphone charger powerbank cover glass )

length\_MA=${#My\_Array[\*]}

for (( i=0;i<$length\_MA;i++ ))

do

echo "smartphone accessories are :: ${My\_Array[$i]}"

Done

***WHILE LOOP:Run commands as long as a condition is true.***

**EX:1**

#!/bin/bash

#While loop

count=5

num=20

while [[ $count -lt $num ]]

do

echo "values are :: $count"

let count++

Done  
  
**EX:2**

#!/bin/bash

count=1

while [ $count -le 5 ]

do

echo "Count: $count"

((count++))

done

**EX:3: countdown**

#!/bin/bash

count=10

echo "countdown starts...."

while [[ $count -gt 0 ]]

do

echo "countdown starts:: $count"

((count--))

sleep 1

done

echo "time up...."

**EX:4 ping** [**google.com**](http://google.com)

#!/bin/bash

while ! ping -c1 google.com &>/dev/null

do

echo "connection timeout and 100% packet loss"

done

echo "ping is sucessfull"