CONDITION

if (condition satisfied)

{

<execure block>

else

<execure block>

}

Unix style

if [[ condition satisfies ]]

then

<execute block>

else

<executre bnlock>

fi

if [[ condition satisfies ]]

then

<execute block>

elif [[ condition satisfies ]]

then

<execute block>

elif [[ condition satisfies ]]

then

<execureb block>

else

<executre bnlock>

fi

RULES

== ===> comparing string

-eq ===> comparing numbers

-gt ===> greater than

-ge ==> greater than or equal to

-lt ===> lesser than

-le ==> lesser than or equal to

-ne ===> not equal to

&& ==> and condition

|| ===> or condtiob

=~ ==> match as

SHELL SCRIPT

#!/bin/bash

echo "Enter the input"

Output:

Praveen 🡺true

Ajay 🡺almost true

Abhi 🡺not false

Others 🡺false

read a

if [[ $a == "praveen" ]]

then

echo "True"

elif [[ $a == "ajay" ]]

then

echo "Almost True"

elif [[ $a == "abhi" ]]

then

echo "Not false"

else

echo "False"

fi

Praveen, 9:39 AM

#!/bin/bash

Output:

Praveen,PRAVEEN🡺true

Ajay ,AJAY 🡺almost true

Abhi,ABHI 🡺not false

Others 🡺false

If others will came atuomalically,

touch $a

ls –ltr $a

echo $a >>$a

cat $a

stat $a

echo "Enter the input"

read a

if [[ $a == "praveen" ]] || [[ $a == "PRAVEEN" ]]

then

echo "True"

uptime

elif [[ $a == "ajay" ]] || [[ $a == "AJAY" ]]

then

echo "Almost True"

uname

elif [[ $a == "abhi" ]] || [[ $a == "ABHI" ]]

then

echo "Not false"

df -h

else

echo "False"

touch $a

ls -ltr $a

echo $a >>$a

cat $a

stat $a

fi

Praveen, 9:47 AM

#!/bin/bash

echo "Enter the input1"

read a

echo "Enter the input2"

read b

if [[ $a == $b ]]

then

echo "Both inputs are same"

mkdir $a

touch $a/$b

c="$a$b"

echo $c >>$a/$b

echo "Below are contents of file $a/$b"

cat $a/$b

else

echo "Both inputs are not same"

uptime

fi

Praveen, 9:57 AM

#!/bin/bash

echo "Enter the input1"

read a

echo "Enter the input2"

read b

echo "Enter the input3"

read c

if [[ $a == $b ]] && [[ $b == $c ]] && [[ $c == $a ]]

then

echo "All 3 inputs are same"

c="$a$b$c"

useradd $c

usermod -c "$a" $c

grep $c /etc/passwd

elif [[ $a == $b ]] && [[ $b != $c ]] && [[ $c != $a ]]

then

echo "All 3 inputs are not same"

c="$a$c"

groupadd $c

grep $c /etc/group

fi

Praveen, 10:03 AM

#!/bin/bash

echo "Enter the input"

read a

echo "Entered input is $a"

revs=$(echo $a |rev)

if [[ $a == $revs ]]

then

echo "String $a is palindrome"

else

echo "String $a is not palindrome"

fi

Praveen, 10:12 AM

#!/bin/bash

echo "Enter the content to be checked"

read con

col=$(grep $con ty.txt |wc -l)

if [[ $col -gt 0 ]]

then

echo "Content $con present in file"

echo "Below are Lines where $con present in file"

grep -in "$con" ty.txt

else

echo "Content $con not present in file"

fi

Praveen, 10:18 AM

#!/bin/bash

echo "Enter the content to be checked"

read con

echo "Enter the filename with path"

read fil

col=$(grep $con $fil |wc -l)

if [[ $col -gt 0 ]]

then

echo "Content $con present in file $fil"

echo "Below are Lines where $con present in file $fil"

grep -in "$con" $fil

else

echo "Content $con not present in file $fil"

fi

=====================================================================================

NUMBERS

Praveen, 9:31 AM

#!/bin/bash

echo "Enter the number1"

read n1

echo "Enter the number2"

read n2

if [[ $n1 -eq $n2 ]]

then

echo "Both inputs are same"

out=$(($n1+$n2))

echo "Sum of $n1 and $n2 is $out"

uptime

elif [[ $n1 -gt $n2 ]]

then

echo "Number $n1 is greater than $n2"

out=$(($n1\*$n2))

echo "Product of $n1 and $n2 is $out"

uname

elif [[ $n1 -lt $n2 ]]

then

echo "Number $n1 is lesser than $n2"

out=$(($n1-$n2))

echo "Difference of $n1 and $n2 is $out"

free

fi

Praveen, 9:41 AM

#!/bin/bash

echo "Enter the number1"

read n1

echo "Enter the number2"

read n2

echo "Enter the number3"

read n3

if [[ $n1 -eq $n2 ]]

then

echo "Both inputs are same"

out=$(($n1+$n2-$n3))

echo "Sum of $n1 and $n2 and difference of $n3 is $out"

uptime

elif [[ $n1 -gt $n2 ]]

then

echo "Number $n1 is greater than $n2"

out=$(($n1\*$n2+$n3))

echo "Product of $n1 and $n2 and adding $n3 is $out"

uname

elif [[ $n1 -lt $n2 ]]

then

echo "Number $n1 is lesser than $n2"

echo "Enter the input number"

read usrin

out=$(($n1-$n2+$usrin))

echo "Difference of $n1 and $n2 and addingg $usrin is $out"

free

fi

Praveen, 9:49 AM

#!/bin/bash

echo "Enter the number1"

read n1

echo "Enter the number2"

read n2

echo "Enter the number3"

read n3

if [[ $n1 -eq $n2 ]] && [[ $n2 -eq $n3 ]] && [[ $n3 -eq $n1 ]]

then

echo "All inputs are same"

co=$(($n1+$n2+$n3))

echo "Sum of $n1 $n2 $n3 is $co"

elif [[ $n1 -eq $n2 ]] && [[ $n2 -gt $n3 ]] && [[ $n3 -lt $n1 ]]

then

echo "$n1 and $n2 is equal and $n2 is greater than $n3 and $n3 is lesser than $n1"

co=$(($n1\*$n2\*$n3))

echo "Product of $n1 $n2 $n3 is $co"

fi

Praveen, 9:56 AM

#!/bin/bash

echo "Enter the Number"

read p

if [[ $p%2 -eq 0 ]]

then

echo "Number $p is even"

else

echo "Number $p is odd"

fi

~

Praveen, 10:06 AM

#!/bin/bash

echo "Enter the input"

read p

if [[ $p =~ ^[0-9]+$ ]]

then

echo "Input $p contains number"

elif [[ $p =~ ^[a-z]+$ ]]

then

echo "Input $p contains alphanet"

elif [[ $p =~ [a-z] ]] && [[ $p =~ [0-9] ]]

then

echo "Input $p contains both alphabet and number"

fi

#!/bin/bash

echo "Enter the Number"

read p

if [[ $p =~ ^[0-9]+$ ]]

then

echo "Valid input $p provided"

if [[ $p%2 -eq 0 ]]

then

echo "Number $p is even"

else

echo "Number $p is odd"

fi

else

echo "Invalid input $p provided"

fi

Praveen, 10:24 AM

#!/bin/bash

echo "Enter the number"

read n1

echo "Enter the number"

read n2

echo -e "Enter '1' for additon\n'2' for Multiplication\n'3' for subtraction\n'4' for divison\n'5' for all operation"

read usropt

if [[ $usropt -eq 1 ]]

then

echo "Peforming addition"

co=$(($n1+$n2))

echo "Sum of $n1 and $n2 is $co"

elif [[ $usropt -eq 2 ]]

then

echo "Perfroming multiplication"

co=$(($n1\*$n2))

echo "Product of $n1 and $n2 is $co"

elif [[ $usropt -eq 3 ]]

then

echo "Performing subtraction"

co=$(($n1-$n2))

echo "Diferernce of $n1 and $n2 is $co"

elif [[ $usropt -eq 4 ]]

then

echo "Perfroming division"

co=$(($n1/$n2))

echo "Divison of $n1 and $n2 is $co"

elif [[ $usropt -eq 5 ]]

then

co=$(($n1+$n2))

mu=$(($n1\*$n2))

su=$(($n1-$n2))

echo "Sum of $n1 and $n2 is $co"

echo "Product of $n1 and $n2 is $mu"

echo "Diferernce of $n1 and $n2 is $su"

fi

#!/bin/bashecho "Enter the Full file path to be checked"read filif [[ -f $fil ]]thenecho "File $fil exsists"elseecho "$File $fil doesnt exsists\nIn next step it should user confirmation for creating file"echo "Enter 'yes' to create file"read usrif [[ $usr == "yes" ]]thenecho "User confirmed to create file $fil"touch $filif [[ -f $fil ]]thenecho "File $fil got created successfully post user confirmation"ls -ltr $filstat $filelseecho "File $fil not created successfully due to permission issue"fielseecho "User not confirmed to create file $fil"fi

#!/bin/bashecho "Enter the Full file path to be checked"read filif [[ -f $fil ]]thenecho "File $fil exsists"elseecho "$File $fil doesnt exsists\nIn next step it should user confirmation for creating file"echo "Enter 'yes' to create file"read usrif [[ $usr == "yes" ]]thenecho "User confirmed to create file $fil"touch $filif [[ -f $fil ]]thenecho "File $fil got created successfully post user confirmation"ls -ltr $filstat $filecho "Enter the Content1"read con1echo "Enter the content2"read con2echo "Enter the content3"read con3echo -e "$con1\n$con2\n$con3" >>$filecho "Below are content of file $fil"cat $filecho "Below are reversed content of file $fil"tac $fills -ltr $filecho -e "Enter to '1' to change permission\n'2' To change owner\n'3' to change group\n'4' to know size of file\n'5' to compress the file\n'6' Hardlink\n'7' to copy the file\n'8' To know count of lines\n'9' Apply chattr"read optif [[ $opt -eq 1 ]]thenecho "Enter the new permission"read perchmod $per $fills -ltr $filelif [[ $opt -eq 2 ]]thenecho "Enter new owner"read ownchown $own $fills -ltr $filelif [[ $opt -eq 3 ]]thenecho "Enter the new group"read grpchgrp $grp $fills -ltr $filelif [[ $opt -eq 4 ]]thenecho "Below are size of file $fil"du -sh $filelif [[ $opt -eq 5 ]]thengzip $fills -ltr $fil.gzelif [[ $opt -eq 6 ]]thenecho "Enter the hardlink filename"read hdln $fil $hdls -litr $fil $hdelif [[ $opt -eq 7 ]]thenecho "Enter the destination path with file"read copyfilecp -v $fil $copyfilels -ltr $copyfileelif [[ $opt -eq 8 ]]thenecho "BELOW ARE COUNT ofLines of file $fil"wc -l $filelif [[ $opt -eq 9 ]]thenchattr +i $fillsattr $filelif [[ $opt -eq 10 ]]thenecho "Enter the new permission"read perecho "Enter the new owner"read ownecho "Enter the grp"read grpchmod $per $filchown $own $filchgrp $grp $fills -ltr $filfielseecho "File $fil not created successfully due to permission issue"fielseecho "User not confirmed to create file $fil"fi

#!/bin/bashfor ((i=0;i<=20;i++))doecho $idone

#!/bin/bashfor ((i=0;i<=20;i++))doif [[ $i%2 -eq 0 ]]thenecho "Number $i is even"uptimeelif [[ $i -eq 15 ]]thenecho "Number is $i"df -helseecho "Number $i is odd"unamefiecho "==========================================="done

#!/bin/bashecho "Enter the number"read nfor ((i=0;i<=20;i++))doif [[ $i -ge 0 ]] && [[ $i -le 5 ]]thenecho "Number $i is greater than 0 and lesser than 5"adds=$(($i+$n))echo "Sum of $i and $n is $adds"elif [[ $i -gt 5 ]] && [[ $i -le 10 ]]thenecho "Number $i is greater than 5 and lesser than 10"subs=$(($i-$n))echo "Difference of $i and $n is $subs"elif [[ $i -eq 11 ]]thenecho "Number is $i"uptimeelif [[ $i -gt 10 ]] && [[ $i -le 15 ]]thenecho "Number $i is greater than 10 and lesser than 15"prd=$(($i\*$n))echo "Product of $i and $n is $prd"elseecho "Number is $i"unamefiecho "========================================================="done

#!/bin/bashecho "Enter the content1"read con1echo "ENTER THE contebr2"read con2for i in "$con1" "$con2"dofor file in "f1.txt" "f2.txt"docol=$(grep $i $file |wc -l)if [[ $col -gt 0 ]]thenecho "Content $i present in file $file"elseecho "Content $i not present in file $file"fidonedone

#!/bin/bashfor i in "abhi" "rahulharsh" "sujeethguna"docol=$(grep "^$i" /etc/passwd|cut -d ":" -f1| grep "^$i$"|wc -l)if [[ $col -gt 0 ]]thenecho "User $i already exsist in server"elseecho "User $i doesnt exsists in server"fidone

**Server handling**

#!/bin/bash#for i in 192.168.209.144 192.168.209.145 192.168.209.149for ((j=144;j<=150;j++))doi="192.168.209.$j"ping -c2 $i >/dev/nullif [[ $? -eq 0 ]]thenecho "Server $i is pinging"ssh $i "exit"if [[ $? -eq 0 ]]thenecho "Server $i is sshable"ssh $i "uptime;hostname;useradd shivu\_praku\_sujee"elseecho "Server $i is not sshable"fielseecho "Server $i is not pinging"fidone

#!/bin/bashfor i in shiv55 rahulharsh joyjoyjoydocol=$(grep $i /etc/passwd |wc -l)if [[ $col -gt 0 ]]thenecho "User $i already exsists in server"coc=$(getent shadow $i|cut -d ":" -f2|wc -c)if [[ $coc -gt 3 ]]thenecho "Password already set for user $i and Below is passowrd entry"getent shadow $ielseecho "Password Not set for user $i, Now we are setting the password"echo "Enter the password"read -s pasecho $pas|passwd $i --stdincoc\_pas=$(getent shadow $i|cut -d ":" -f2 |wc -c)if [[ coc\_pas -gt 3 ]]thenecho "Password successfully set for user $i"getent shadow $ielseecho "Password not set for user $i"fifielseecho "User $i doesnt exsists in server"useradd $iecho $i |passwd $i --stdingetent passwd $igetent shadow $ifiecho "===================================="done