**Shell Scripting Examples:**

**Exa: ss1.sh**

# This is my 1st shell script

echo "Hello Yuvi"

Exa: ss2.sh

# This is my 2nd script file

banner ========================

HELLO WORLD

banner ========================

Exa: ss3.sh

# Using variable and read command to take input from people

echo "Please enter your name"

read my\_name

echo "Hi $my\_name, How are you?"

Exa: ss4.sh

# This script rename files to file.name where name is the logged in name of user

name=$1

set `who am i`

mv $name $name.$1

Exa: ss5.sh

# Arithmetic operation

a=30 b=20

echo `expr $a + $b`

echo `expr $a - $b`

echo `expr $a \\* $b`

echo `expr $a / $b`

echo `expr $a % $b`

Exa: ss6.sh

#Arthmetic script

echo "Please enter the bumbers:"

read a

read b

echo `expr $a + $b`

echo `expr $a - $b`

echo `expr $a \\* $b`

echo `expr $a / $b`

echo `expr $a % $b`

Exa ss7.sh

# Arithmetic operaters

echo "Please enter the values:"

read a

read b

read c

read d

echo =`expr $a \\* \( $b + $c \) / $d`

Exa : ss8.sh

# Arthmetic operation with float(exa: 4.5) values

echo "please enter the values:"

read a

read b

c=`echo $a + $b | bc`

d=`echo $a - $b | bc`

e=`echo $a \\* $b | bc`

f=`echo $a / $b | bc`

echo $c $d $e $f

Exa: ss9.sh

# Describing escape sequencces

echo -e " Hey Yuvi,\nwhats going on" #\n for new line

Exa: ss10.sh

#tput in action

tput clear

echo -e "Total number of rows on screen=\c"

tput lines

echo -e "Total number of lines on screec=\c"

tput cols

tput cup 15 20

tput bold

echo "Hellow world"

echo -e "\033[om Bye Bye"

Exa: string.sh

# string checks

str1="Hello"

str2="world"

str3=""

[ "$str1" = "$str2" ] # to check the both the strings are same

echo $? # to check the status of command. 0 for successfull and 1 for failed

[ "$str1" != "$str2" ] # to check both the strings are not equal

echo $?

[ -n "$str1" ] # to calculate the length of variable is getter than is zero.

echo $?

[ -z "$str3" ] # to check the string a nul value.

echo $?

Exa: and.sh

echo "Enter a number between 50 and 100:"

read num

if [ $num -ge 50 -a $num -le 100 ]

then

echo " The number is in between 50 to 100"

else

echo "The numbe not within the range"

fi

Exa: append.sh

echo "Please enter the file name: "

read fname

if [ -f $fname ]

then

if [ -w $fname ]

then

echo " Type the text to append and precc ctrl+d to quit"

cat >> $fname

else

echo "You don't have the write permission"

fi

fi

Exa: case.sh

echo "Enter a character: "

read var

case $var in

[a-z])

echo " This is a lowercase alphabates

;;

[A-Z])

echo " This is a uppercase alphabate"

;;

[0-9])

echo " This is a numiric digit"

;;

?)

echo " It's special character"

;;

\*)

echo " more than one character"

;;

esac

Exa: if-then.sh

# if-then statement in action

echo "Please provide the file names:"

read source

read dest

if mv $source $dest

then

echo "File has been renamed successfully."

fi

Exa: if-then-else.sh

# if-then-else

echo "Plese provide the source and destination file or folder name: "

read source

read dest

if [ -f $source ] # -f used for file, -d for directory, -s for file size

then

if [ -f $dest ]

then

echo "one file is already exist as targeted file name."

else

if mv $source $dest

then

echo "renamed has been successfully"

else

echo "unable to rename the file/folder"

fi

fi

else

echo "file named as $source is not exist"

fi

Exa: if-elif.sh

# if-elif condition with numbers

echo "Enter the number between 10 and 20: "

read num

if [ $num -lt 10 ] # -lt stands for lessthan

then

echo "That was less than the 10"

elif [ $num -gt 20 ] # -gt stands for gaterthan

then

echo "That number is gater than 20"

elif [ $num -eq 10 ] # -eq stands for equals

then

echo "This number is equals to 10"

else

echo "Yes....., This number is within the range..Thanks"

fi

Exa: count.sh

echo "Please enter a chareacter"

read char

if [ `echo $char | wc -c` -eq 2 ] # count the character -W for word -L for line

then

echo "you entered a chanracter" # -le stands for lessthan equal

else # -ge stands for gatertahn equal

echo "Invalid character"

fi

Exa: or.sh

echo "Please enter a lowercase character"

read var

if [ `echo $var | wc -c` -eq 2 ]

then

if [ $var = a -o $var = e -o $var = i -o $var = o -o $var = u ] # -o for or

then

echo "It's a vowel"

else

echo "it's not a vowel"

fi

else

echo "it's a invalid character"

fi

Exa: while.sh

# while loop demo

count=1

while [ $count -le 10 ]

do

echo $count

count=`expr $count + 1`

done

Exa: until.sh

#until loop demo

count=1

until [ $count -ge 10 ]

do

echo $count

count=`expr $count + 1`

done

Exa: compare\_real.sh

# comparing the real number

echo "Please enter the number: "

read a

read b

if [ "$a" = "$b" ]

then

echo "Both the values are same"

else

echo "both the values are different"

fi

Exa: for-loop.sh

# for loop demo

for item in \*

do

if [ -f $item ] # -f for file and -d for directory

then

echo $item

fi

done

1. Write : write command is used to communicate or pass the message to the other logged in user of the machine.

Syntax: write <username>

<your message>

Exa: write alok

Hi alok, I’m going for tea.

1. mesg : mesg command is used to enable the messaging option.

Syntax: mesg y # to enable

mesg n # to disable

Exa: IFS.sh

# internal field separator(IFS)

line="Cricket:is:my:life"

IFS=: # IFS is used to seperate the words of line using specified seperator(:).

set $line

echo $1

echo $2

echo $3

echo $4

Exa: pinfo.sh # passwd info

echo "Enter User Name:"

read logname

line=`grep $logname /etc/passwd`

IFS=:

set $line

echo "User Name: $1"

echo "User ID: $3"

echo "GroupID: $4"

echo "Comment: $5"

echo "Home Folder: $6"

echo " Shell: $7"

Exa: read.sh

echo " Enter the file Name:"

read fname

if [ -z "$fname" ]

then

exit

fi

terminal=`tty` # tty holds the text of terminal.

exec < $fname

count=1

while read line

do

echo $count. $line

count=`expr $count + 1`

done

exec < $terminal

Eax: sleep.sh

echo "Enter the sentance: "

read str

for word in $str

do

echo $word

sleep 5

done

Exa: count1.sh

echo " Enter the file Name:"

read fname

terminal=`tty`

exec < $fname

nol=0

now=0

while read line

do

nol=`expr $nol + 1`

set $line

now=`expr $now + $#`

done

echo "Number of lines:$nol"

echo " Number of words:$now"

exec < $terminal

Exa: help.sh

for cmd in `cat commandlist` #commandlist is file which holds the commands

do

man $cmd >> helpfile

done

Exa: commandlist

cat

date

cal

touch

Exa: nested-loop.sh

a=1

while [ $a -le 3 ]

do

b=1

while [ $b -le 3 ]

do

c=1

while [ $c -le 3 ]

do

echo "$a$b$c"

c=`expr $c + 1`

done

b=`expr $b + 1`

done

a=`expr $a + 1`

done

Exa: break.sh

count=1

while [ $count -le 10 ]

do

if [ $count -eq 6 ]

then

break

fi

echo $count

count=`expr $count + 1`

done

echo " We are out of the while loop"

Exa: continue.sh

count=0

while [ $count -le 9 ]

do

count=`expr $count + 1`

if [ $count -eq 5 ]

then

continue

fi

echo $count

done

echo "We are out of the loop."

Exa: check.sh

echo -e "Enter username:\c"

read logname

time=0

while true #infinite loop

do

who | grep "$logname" > /dev/null

if [ $? -eq 0 ]

then

echo " $logname has logged in.."

if [ $time -ne 0 ] # -ne for notequal

then

echo "$logname was $time minutes late in logging in."

fi

exit

else

time=`expr $time + 1`

sleep 1

fi

done

Exa: function.sh

youtube()

{

echo "Good morning"

}

byebye()

{

cal

}

* chmod +X function.sh : to make function file as a executable
* . function.sh : to execute function.sh
* Youtube : to call the youtube function inside function.sh file
* ls ; cal ; banner "Yuvi" : used multiple command at a time.