1. **system variables**

#!/bin/bash

#system variables

echo $SHELL

echo $BASH

echo $BASH\_VERSION

echo $HOME

echo $PWD

echo $USER

1. **user variables**

#user variables

a=10

echo $a

1. **read user inputs**

#read user inputs

echo "enter a place :"

read place

echo "enetered a place : $place"

echo "enter the places :"

read place1 place2 place3

echo "entered the places : $place1,$place2,$place3"

read -sp "user name :" user\_var

echo "user name is : $user\_var"

read -sp "password :" pass\_var

echo "password is : $pass\_var"

1. **arguments and their values**

#Print total arguments and their values

echo "print the total arguments :" $#

echo "print the arguments values :" $@

echo "first-->" $1

echo "second-->" $2

1. **ifelse (comparison operators(= =, -eq)... compare two numbers)**

#ifelse statement... comparison operators(==)... compare two numbers (example:1)

a=10

b=20

if [ $a == $b]

then

echo "$a a is equal to b"

else

echo "$a a is not equal to b"

fi

#Ifelse statement... comparison operators (== (or) -eq) ... compare two numbers (example:2)

echo "enter a value :"

read a

echo "enter b value"

read b

if [ $a == $b]

then

echo "a is equal to b"

else

echo "a is not equal to b"

fi

**(or)**

echo "enter a value :"

read a

echo "enter b value"

read b

if [ $a -eq $b ]

then

echo "a is equal to b"

else

echo "a is not equal to b"

fi

1. **ifelse file operators**[find the existing file(-f) and directories(-d) and read(-r) write(-w) execute(-x) permissions]

#ifelse file operators [find the existing file(-f) and directories(-d) and read(-r) write(-w) execute(-x) permissions]

#ifelse file operator (-f)

echo "enter the file name"

read file\_name

if [ -f $file\_name ]

then

echo "file $file\_name found"

else

echo "file $file\_name not found"

fi

#ifelse directory operator(-d)

echo "enter the directory name"

read directory\_name

if [ -d $directory\_name ]

then

echo "dirctory $directory\_name found"

else

echo "directory $directory\_name not found"

fi

#ifelse read permission operator(-r)

echo "enter the directory name"

read directory\_name

if [ -r $directory\_name ]

then

echo "read permissions $directory\_name found"

else

echo " read permissions $directory\_name not found"

fi

#Ifelse write permission operator(-w)

echo "enter the directory name"

read directory\_name

if [ -w $directory\_name ]

then

echo " write permissions $directory\_name found"

else

echo " write permissions $directory\_name not found"

fi

#ifelse execute permission operator(-x)

echo "enter the directory name"

read directory\_name

if [ -x $directory\_name ]

then

echo " execute permissions $directory\_name found"

else

echo " execute permissions $directory\_name not found"

fi

1. **ifelse [ and (-a,&),or (-o,||)] operators**

#ifelse and operator (-a)

echo "enter ssc percentage :"

read ssc

echo "enter inter percentage :"

read inter

if [ $ssc -ge 70 -a $inter -ge 65 ]

then

echo "candidate eligible"

else

echo "candidate not eligible"

fi

**(or)**

#ifelse and operator (&)

echo "enter ssc percentage :"

read ssc

echo "enter inter percentage :"

read inter

if [ $ssc -ge 70 ] & [ $inter -ge 65 ]

then

echo "candidate eligible"

else

echo "candidate not eligible"

fi

#ifelse or operator (-o)

echo "enter ssc percentage :"

read ssc

echo "enter inter percentage :"

read inter

if [ $ssc -ge 70 -o $inter -ge 65 ]

then

echo "candidate eligible"

else

echo "candidate not eligible"

fi

**(or)**

#ifelse or operator (||)

echo "enter ssc percentage :"

read ssc

echo "enter inter percentage :"

read inter

if [ $ssc -ge 70 ] || [ $inter -ge 65 ]

then

echo "candidate eligible"

else

echo "candidate not eligible"

fi

1. elif (comparison operators(-gt: greater than ))

#elif statement... comparison operators(-gt: greater than)... compare of multiple numbers (example:1)

echo "enter a value :"

read a

echo "enter b value :"

read b

echo "enter c value :"

read c

echo "enter d value :"

read d

if [ $a -gt $b -a $a -gt $c -a $a -gt $d ]

then

echo "$a a is big"

elif [ $b -gt $c -a $b -gt $d ]

then

echo "$b b is big"

elif [ $c -gt $d ]

then

echo "$c c is big"

else

echo "$d d is big"

fi

**(or)**

#elif statement... comparison operators(-gt: greater than )... compare of multiple numbers (example:2)

echo -e "enter a value : \c"

read -r a

echo "enter b value : \c"

read -r b

echo -e "enter c value : \c"

read -r c

echo -e "enter d value : \c"

read -r d

if [ $a -gt $b -a $a -gt $c -a $a -gt $d ]

then

echo "$a a is big"

elif [ $b -gt $c -a $b -gt $d ]

then

echo "$b b is big"

elif [ $c -gt $d ]

then

echo "$c c is big"

else

echo "$d d is big"

fi

**9.Arithmetic Operators**

Arithmetic Operators are:

+: Addition

-: Subtraction

\*: Multiplication

/: Division

ex1:

#!/bin/bash

#arithmetic operators (integers=1, 2,.. )

echo "enter first number :"

read num1

echo "enter second number :"

read num2

echo "addition is :" $(( num1 + num2 ))

echo "subtraction is :" $(( num1 - num2 ))

echo "multiplication is :" $(( num1 \* num2 ))

echo "division is :" $(( num1 / num2 ))

**(or)**

ex2:

#arithmetic operators ( floats=2.5, 3.7, 4.2, ....)

echo "enter first number :"

read num1

echo "enter second number :"

read num2

echo "scale=2;$num1+$num2" | bc

echo "scale=2;$num1-$num2" | bc

echo "scale=2;$num1\*$num2" | bc

echo "scale=2;$num1/$num2" | bc

**10.FOR LOOP**

1. **Create number of files with .txt and append the content in all .txt files.**

#!/bin/bash

for i in raju{1..4}.txt

do

echo "hi hello" >> $i

done

1. **Create number of files with .txt using touch command.**

#!/bin/bash

touch raju{1..4}.txt

1. **Create number of files and append the different content in different files.**

#!/bin/bash

for i in {1..10}

do

echo "Number$i" >> file$i

done

1. **Form a Right-angle triangle with numbers.**

#! /bin/bash

echo "enter a value n :"

read n

for((i=1;i<=n;i++))

do

for((k=1;k<=i;k++))

do

echo -ne "$k"

done

echo

done