**Relational operators**

#!/bin/sh

a=10

b=20

if [ $a -eq $b ]

then

echo "$a -eq $b : a is equal to b"

else

echo "$a -eq $b: a is not equal to b"

fi

if [ $a -ne $b ]

then

echo "$a -ne $b: a is not equal to b"

else

echo "$a -ne $b : a is equal to b"

fi

if [ $a -gt $b ]

then

echo "$a -gt $b: a is greater than b"

else

echo "$a -gt $b: a is not greater than b"

fi

if [ $a -lt $b ]

then

echo "$a -lt $b: a is less than b"

else

echo "$a -lt $b: a is not less than b"

fi

if [ $a -ge $b ]

then

echo "$a -ge $b: a is greater or equal to b"

else

echo "$a -ge $b: a is not greater or equal to b"

fi

if [ $a -le $b ]

then

echo "$a -le $b: a is less or equal to b"

else

echo "$a -le $b: a is not less or equal to b"

fi

The above script will generate the following result −

10 -eq 20: a is not equal to b

10 -ne 20: a is not equal to b

10 -gt 20: a is not greater than b

10 -lt 20: a is less than b

10 -ge 20: a is not greater or equal to b

10 -le 20: a is less or equal to b

**file test operators**

#!/bin/sh

file="/var/www/tutorialspoint/unix/test.sh"

if [ -r $file ]

then

echo "File has read access"

else

echo "File does not have read access"

fi

if [ -w $file ]

then

echo "File has write permission"

else

echo "File does not have write permission"

fi

if [ -x $file ]

then

echo "File has execute permission"

else

echo "File does not have execute permission"

fi

if [ -f $file ]

then

echo "File is an ordinary file"

else

echo "This is sepcial file"

fi

if [ -d $file ]

then

echo "File is a directory"

else

echo "This is not a directory"

fi

if [ -s $file ]

then

echo "File size is zero"

else

echo "File size is not zero"

fi

if [ -e $file ]

then

echo "File exists"

else

echo "File does not exist"

fi

The above script will produce the following result −

File has read access

File has write permission

File has execute permission

File is an ordinary file

This is not a directory

File size is zero

File exists

**Looping Statements Ex:**

#!/bin/sh

a=0

while [ $a -lt 10 ]

do

echo $a

if [ $a -eq 5 ]

then

break

fi

a=`expr $a + 1`

done

Upon execution, you will receive the following result −

0

1

2

3

4

5

#!/bin/sh

for var1 in 1 2 3

do

for var2 in 0 5

do

if [ $var1 -eq 2 -a $var2 -eq 0 ]

then

break 2

else

echo "$var1 $var2"

fi

done

done

Upon execution, you will receive the following result. In the inner loop, you have a break command with the argument 2. This indicates that if a condition is met you should break out of outer loop and ultimately from the inner loop as well.

1 0

1 5

#!/bin/sh

NUMS="1 2 3 4 5 6 7"

for NUM in $NUMS

do

Q=`expr $NUM % 2`

if [ $Q -eq 0 ]

then

echo "Number is an even number!!"

continue

fi

echo "Found odd number"

done

Upon execution, you will receive the following result −

Found odd number

Number is an even number!!

Found odd number

Number is an even number!!

Found odd number

Number is an even number!!

Found odd number