**Assignments :**

-----------------

**Conditional Operators**

There are 3 conditional statements which can be used in bash programming

**if statement**

Syntax:

if [ expression ]

then

statement

fi

Program

#Initializing two variables

a=10

b=20

#Check whether they are equal

if [ $a == $b ]

then

echo "a is equal to b"

fi

#Check whether they are not equal

if [ $a != $b ]

then

echo "a is not equal to b"

fi

Output

$bash -f main.sh

a is not equal to b

2. if-else statement

Syntax:

if [ expression ]

then

statement1

else

statement2

fi

Program:

#Initializing two variables

a=20

b=20

if [ $a == $b ]

then

#If they are equal then print this

echo "a is equal to b"

else

#else print this

echo "a is not equal to b"

fi

Output:

$bash -f main.sh

a is equal to b

Elif

Program

#!/bin/bash

if [ 0 -eq 1 ]; then

echo '0=1'

else

if [ 0 -eq 2 ]; then

echo '0=2'

else

echo '0!=2'

fi

fi

Output

$ ./test.sh

0!=2

**looping statements**

**while Loops**

It is possible to use a while loop as part of the body of another while loop.

Syntax

while command1 ; # this is loop1, the outer loop

do

Statement(s) to be executed if command1 is true

while command2 ; # this is loop2, the inner loop

do

Statement(s) to be executed if command2 is true

done

Statement(s) to be executed if command1 is true

done

Example:

#!/bin/sh

a=0

while [ "$a" -lt 10 ] # this is loop1

do

b="$a"

while [ "$b" -ge 0 ] # this is loop2

do

echo -n "$b "

b=`expr $b - 1`

done

echo

a=`expr $a + 1`

done

Output

0

1 0

2 1 0

3 2 1 0

4 3 2 1 0

5 4 3 2 1 0

6 5 4 3 2 1 0

7 6 5 4 3 2 1 0

8 7 6 5 4 3 2 1 0

9 8 7 6 5 4 3 2 1 0

**For Loop**

For loop is another type of looping statement to execute a set of commands for a certain number of times.

Syntax

for var in list

do

command 1

command 2

done

Program

for p\_name in Stan Kyle Cartman

do

echo $p\_name

done

Output

Stan

Kyle

Cartman