There are total 3 looping statements which can be used in bash programming

1. while statement
2. for statement
3. until statement

To alter the flow of loop statements, two commands are used they are,

1. break
2. continue

Their descriptions and syntax are as follows:

**while statement:**Here command is evaluated and based on the result loop will executed, if command raise to false then loop will be terminated

while <condition>

do

<command 1>

<command 2>

<etc>

done

**Example:** This program will print the value of ‘a’ five times, from 1 to 5.

a=1

while [ $a -le 5 ]

do

echo “value of a=” $a

a=`expr $a + 1`

done

**for statement**: The for loop operate on lists of items. It repeats a set of commands for every item in a list.

**Syntax:**

for <var> in <value1 value2 ... valuen>

do

<command 1>

<command 2>

<etc>

Done

**Example:** This program will add 1+2+3+4+5 and the result will be 15

for i in 1 2 3 4 5

do

sum=`expr $sum + $i`

done

echo $sum

**until statement:**The until loop is executed as many as times the condition/command evaluates to false. The loop terminates when the condition/command becomes true.

**Syntax:**

until <condition>

do

<command 1>

<command 2>

<etc>

done

This program will print the value of ‘a’ two times from 1 to 2.

a=1

until [ $a -ge 3 ]

do

echo “value of a=” $a

a=`expr $a + 1`

done

**Break and Continue Keywords**

Both “break” and “continue” are used to transfer control of the program to another part of the program. It is used within loops to alter the flow of the loop and terminate the loop or skip the current iteration.

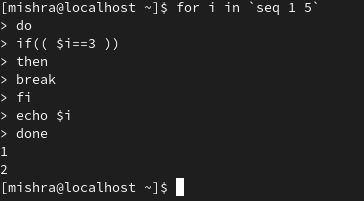
### **break**

The break statement is used to terminate the loop and can be used within a while, for, until, and select loops.

**Syntax**

break

Using break command in a loop.



It can be seen that when the value of i is 3, the loop execution is terminated and hence i is only printed upto 2.

**Example:** Using break with the value of N. Consider an example:

for i in `seq 1 5`

do

for j in `seq 1 5`

do

if(( $j== 2 ))

then

break 2

fi

echo "value of j is $j"

done

echo "value of i is $i"

done

As the value of break is 2, when the value of j is 2, both the loops are exited in a single step. So, the only value of j is printed when it was 1.

break keyword in linux

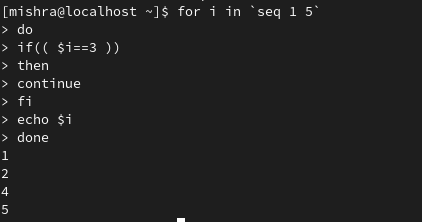
### **continue**

Continue is a command which is used to skip the remaining command inside the loop for the current iteration in for, while, and until loop.

**Syntax:**

continue

Using continue command in a loop



**Example:** Using continue with the value of N. Consider an example:

for i in `seq 1 5`

do

for j in `seq 1 5`

do

if(( $j== 2 ))

then

continue 2

fi

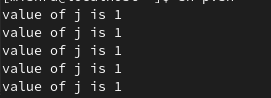
echo "value of j is $j"

done

echo "value of i is $i"

done

Continue skips both the loop when the value of j is 2 and hence, the code only executes when the value of j is 1.



#### **Difference between break and continue**

| **Sr. No.** | **break** | **continue** |
| --- | --- | --- |
| 1 | It terminates the execution of the loop for all the remaining iterations. | It skips the execution of the loop for only the current iteration. |
| 2 | It allows early termination of the loop. | It allows early execution of the next iteration. |
| 3 | It stops the execution of loops. | It stops the execution of the loop only for the current iteration. |
| 4 | The code after the loop which was terminated is continued. | The code in the loop continues its execution skipping the current iteration. |