

Introduction to Loops

Sometimes a programmer wants to execute a group of statements a certain number of times. This can be done using loops. Based on this loops are further classified into following main types;

- for loop
- while loop

The for Loop

for loops can iterate over a sequence of iterable objects in python. Iterating over a sequence is nothing but iterating over strings, lists, tuples, sets and dictionaries.

Example: iterating over a string:

```
name = 'Abhishek'
for i in name:
    print(i, end=" ")
```

Output:

A, b, h, i, s, h, e, k,

Example: iterating over a list:

```
colors = ["Red", "Green", "Blue", "Yellow"]
for x in colors:
    print(x)
```

Output:

Red
Green
Blue
Yellow

Similarly, we can use loops for lists, sets and dictionaries.

range():

What if we do not want to iterate over a sequence? What if we want to use for loop for a specific number of times?

Here, we can use the range() function.

Example:

```
for k in range(5):  
    print(k)
```

Output:

```
0  
1  
2  
3  
4
```

Here, we can see that the loop starts from 0 by default and increments at each iteration.

But we can also loop over a specific range.

Example:

```
for k in range(4,9):  
    print(k)
```

Output:

```
4  
5  
6  
7  
8
```

Python while Loop

As the name suggests, while loops execute statements while the condition is True. As soon as the condition becomes False, the interpreter comes out of the while loop.

Example:

```
count = 5  
while (count > 0):  
    print(count)
```

```
count = count - 1
```

Output:

```
5  
4  
3  
2  
1
```

Here, the count variable is set to 5 which decrements after each iteration. Depending upon the while loop condition, we need to either increment or decrement the counter variable (the variable count, in our case) or the loop will continue forever.

Else with While Loop

We can even use the else statement with the while loop. Essentially what the else statement does is that as soon as the while loop condition becomes False, the interpreter comes out of the while loop and the else statement is executed.

Example:

```
x = 5  
while (x > 0):  
    print(x)  
    x = x - 1  
else:  
    print('counter is 0')
```

Output:

```
5  
4  
3  
2  
1  
counter is 0
```

Do-While loop in python

do..while is a loop in which a set of instructions will execute at least once (irrespective of the condition) and then the repetition of loop's body will depend on the condition passed at the end of the while loop. It is also known as an exit-controlled loop.

How to emulate do while loop in python?

To create a do while loop in Python, you need to modify the while loop a bit in order to get similar behavior to a do while loop.

The most common technique to emulate a do-while loop in Python is to use an infinite while loop with a break statement wrapped in an if statement that checks a given condition and breaks the iteration if that condition becomes true:

Example

```
while True:
    number = int(input("Enter a positive number: "))
    print(number)
    if not number > 0:
        break
```

Output

```
Enter a positive number: 1
1
Enter a positive number: 4
4
Enter a positive number: -1
-1
```

Explanation

This loop uses True as its formal condition. This trick turns the loop into an infinite loop. Before the conditional statement, the loop runs all the required processing and updates the breaking condition. If this condition evaluates to true, then the break statement breaks out of the loop, and the program execution continues its normal path.