

Python - While Loop

Loop is a very popular phrase in programming jargon. A program, by default, follows a sequential execution of statements. If the program flow is directed towards any of the earlier statements in the program, it constitutes a loop. However, sending it unconditionally causes an infinite loop, which is not desired.

Python uses the `while` and `for` keywords to constitute a conditional loop, by which repeated execution of a block of statements is done until a Boolean expression is true. Here, you will learn about the while loop.

The following is the while loop syntax.

Syntax:

```
while [boolean expression]:  
    statement1  
    statement2  
    ...  
    statementN
```

Python keyword `while` has a conditional expression followed by the `:` symbol to start a block with increased indent. This block has statements to be executed repeatedly. Such a block is usually referred to as the body of the loop. The body will keep executing till the condition remains true. If and when it turns out to be false, the program will come out of the loop. Consider the following example.

Example: while loop

```
num = 0  
while num < 5:  
    num = num + 1  
    print("num =", num)
```

Result:

```
num = 1  
num = 2  
num = 3
```

```
num = 4  
num = 5
```

Here the repetitive block after the while statement involves incrementing the value of an integer variable and printing it. Before the block begins, the variable num is initialized to 0. Till it is less than 5, num is incremented by 1 and printed to display the sequence of numbers, as above.

Let us develop a Python program that successively takes a number as input from the user and calculates the average, as long as the user enters a positive number. Here, the repetitive block (the body of the loop) asks the user to input a number, adds it cumulatively and keeps the count if it is non-negative.

Example: while loop

```
num=0  
count=0  
sum=0  
while num>=0:  
    num=int(input("enter any number .. -1 to exit\n"))  
    if num>=0:  
        count=count+1 #this counts number of inputs  
        sum=sum+num # this adds input number cumulatively.  
avg=sum/count  
print ("numbers input:",count, "average:",avg)
```

When a negative number is provided by the user, the loop terminates and displays the average of the numbers provided so far. A sample run of the above code is below:

Result:

```
enter any number .. -1 to exit  
10  
enter any number .. -1 to exit  
20  
enter any number .. -1 to exit  
30  
enter any number .. -1 to exit
```

-1

numbers input: 3 average: 20.0