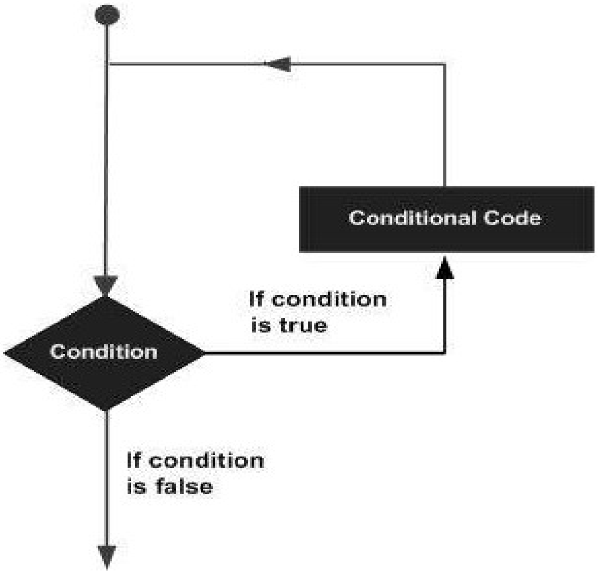
Programming and Problem Solving through Python Language
O Level / A Level

Chapter -4: Operators, Expressions and Python Statements

Loop Statement

A loop statement allows us to execute a statement or group of statements multiple times.



Loop Types

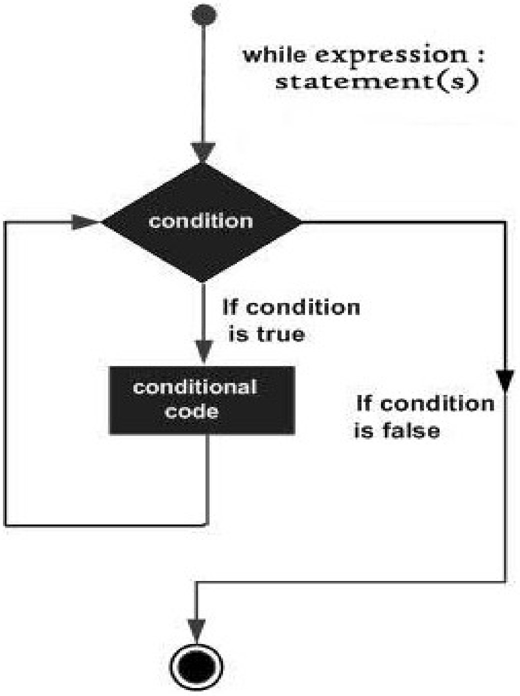
1. **while loop** : Repeats a statement or group of statements while a given condition is TRUE. It tests the condition before executing the loop body.
2. **for loop** : Executes a sequence of statements multiple times and abbreviates the code that manages the loop variable.
3. **nested loops** : You can use one or more loop inside any another **while**, **for** or **do..while** loop.

While Loop

- A **while** loop statement repeatedly executes a statement as long as a given condition is true.
- Syntax

```
while expression:
      statement(s)
```

- **statement(s)** may be a single statement or a block of statements. The **condition** may be any expression, and true is any non-zero value. The loop iterates while the condition is true.
- When the condition becomes false, program control passes to the line immediately following the loop.
- In Python, all the statements indented by the same number of character spaces after a programming construct are considered to be part of a single block of code. Python uses indentation as its method of grouping statements.



Example

```
# Program to print 1 t 10

count = 0
while (count < 11):
    print ('The count is:', count)
    count = count + 1
```

When the above code is executed, it produces the following result −

```
The count is: 0
The count is: 1
The count is: 2
The count is: 3
The count is: 4
The count is: 5
The count is: 6
The count is: 7
The count is: 8
```

```
# Program to print 1 ,3,5,7,9

count = 1
while (count < 10):
     print ('The count is:', count)
     count = count + 2
```

When the above code is executed, it produces the following result −

```
The count is: 1
The count is: 3
The count is: 5
The count is: 7
The count is: 9
```

```
# Program to print sum of 1 to 10

count = 1
sum = 0
while (count <= 10):
     sum=sum+count
     count = count + 1

print ('The sum is:', sum)
```

When the above code is executed, it produces the following result −

```
The sum is: 55
```

```
# Program to factorial of number

count = 1
fact = 1
while (count < = 5):
     fact = fact * count
     count = count + 1

print ('The fact is:', fact)
```

When the above code is executed, it produces the following result −

```
The fact is: 120
```

Using else Statement with Loops

If the **else** statement is used with a **while** loop, the **else** statement is executed when the condition becomes false.

```
count = 0
while count < 5:
   print (count, " is  less than 5")
   count = count + 1
else:
   print (count, " is not less than 5")
```

When the above code is executed, it produces the following result −

```
0 is less than 5
1 is less than 5
2 is less than 5
3 is less than 5
4 is less than 5
5 is not less than 5
```

```
count = 1
sum = 0
while count <= 10:
   sum = sum + count
   count = count + 1
else:
   print (" Sum : " , sum)
```

When the above code is executed, it produces the following result −

```
Sum : 55
```

The break Statement

With the **break** statement we can stop the loop even if the while condition is true:

Example

#Exit the loop when i is 3:

```
i = 1
while i < 6:
   print(i)
   if i == 3:
      break
 i += 1
```

Output

```
1

2
```

The continue Statement

With the continue statement we can stop the current iteration, and continue with the next:

Example

```
#Continue to the next iteration if i is 3:
i = 0
while i < 6:
    i += 1
    if i == 3:
        continue
    print(i)
```

Output

```
1
2
4
5
```

Assignment

1. Write the program to display the first 10 terms of the following series :
    a. 1 , 3, 5,…………
    b. 2 , 4 , 6 …………
    c. 1 , 4 , 9 , 16…….
    d. 1.5 , 3.0 , 4.5 , 6.0 ……..
    e. -5 , -10 , -15, -20 ……….
2. Write a program to input any 20 numbers (including positive and negative). Perform the following tasks
    a. Count and display the positive numbers
    b. Count and display the negative numbers
    c. Display the sum of positive numbers
    d. Display the sum of negative numbers
3. Write a program to calculate and display the sum of all odd numbers and even numbers between a range of numbers from m to n where m < n. Input m and n.
4. Write a program to print the 10 multiples of any entered number.
5. Write a program to display the sum of 10 natural numbers.
6. Write a program to calculate and display the factorial of a entered number.