Control Flow (Conditionals and Loops)

Python programming language provides the following types of loops to handle looping requirements.

- **While Loop**
- **For Loop**
- **Nested Loops**
- **Loop Control**

1.While Loop:-

Until a specified criterion is true, a block of statements will be continuously executed in a Python while loop. And the line in the program that follows the loop is run when the condition changes to false.

Syntax of While Loop:-

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

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 for While Loop:-

```
count = 0
while (count < 3):
    count = count+1
```

```
    print("Hello Geek")
```

Output:-

Hello Geek

Hello Geek

Hello Geek

2.For Loop

In Python, there is no C style for loop, i.e., for (i=0; i<n; i++). There is a “for in” loop which is similar to for each loop in other languages.

Syntax of Python for Loop:-

```
for iterator_var in sequence:
    statements(s)
```

Example For loop:-

```
print("List Iteration")
l = ["geeks", "for", "geeks"]
for i in l:
    print(i)

# Iterating over a tuple (immutable)
print("\nTuple Iteration")
t = ("geeks", "for", "geeks")
```

```
for i in t:
    print(i)


# Iterating over a String
print("\nString Iteration")
s = "Geeks"
for i in s :
    print(i)


# Iterating over dictionary
print("\nDictionary Iteration")
d = dict()
d['xyz'] = 123
d['abc'] = 345
for i in d :
    print("%s %d" %(i, d[i]))
```

Output :-

```
List Iteration
geeks
for
geeks


Tuple Iteration
```

geeks

for

geeks

String Iteration

G

e

e

k

s

Dictionary Iteration

xyz 123

abc 345

Nested Loops :-

Python programming language allows using one loop inside another loop. The following section shows a few examples to illustrate the concept.

Syntax of Nested Loop:-

The syntax for a nested for loop statement in Python programming language is as follows:

for iterator_var in sequence:

```
   for iterator_var in sequence:
      statements(s)
      statements(s)
```

Example For Nested Loop:-

```
# Running outer loop from 2 to 3


for i in range(2, 4):


   # Printing inside the outer loop
   # Running inner loop from 1 to 10
   for j in range(1, 11):


      # Printing inside the inner loop
      print(i, "*", j, "=", i*j)
   # Printing inside the outer loop


   print()
```

Output:-

```
2 * 1 = 2

2 * 2 = 4

2 * 3 = 6

2 * 4 = 8

2 * 5 = 10
```

2 * 6 = 12

2 * 7 = 14

2 * 8 = 16

2 * 9 = 18

2 * 10 = 20

3 * 1 = 3

3 * 2 = 6

3 * 3 = 9

3 * 4 = 12

3 * 5 = 15

3 * 6 = 18

3 * 7 = 21

3 * 8 = 24

3 * 9 = 27

3 * 10 = 30

Nested while Loop