

Lecture-07: Loops in Python

Loops in Python:

Python programming language provides following types of loops to handle looping requirements. Python provides three ways for executing the loops. While all the ways provide similar basic functionality, they differ in their syntax and condition checking time.

If else:

if condition:

```
# execute these statements
```

else:

```
# execute these statements
```

GALGOTIAS
UNIVERSITY

Example:

```
a = 17
```

```
b = 20
```

```
if ( a is b ):
```

```
    print "Line 1 - a and b have same identity"
```

```
else:
```

```
    print "Line 1 - a and b do not have same identity"
```

OUTPUT

```
Line 1 - a and b do not have same identity
```

GALGOTIAS
UNIVERSITY

While Loop:

In python, while loop is used to execute a block of statements repeatedly until a given a condition is satisfied. And when the condition becomes false, the line immediately after the loop in program is executed.

Syntax:

while expression:

 statement(s)

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.

UNIVERSITY

School of Basic and Applied Sciences

Course Code : BSCM 304

Course Name: Programming Using Python

Example:

```
# while loop
count = 0
while (count < 3):
    count = count + 1
    print("Hello Geek")
```

Output:

```
Hello Geek
Hello Geek
Hello Geek
```



GALGOTIAS
UNIVERSITY

Using else statement with while loops:

As discussed above, while loop executes the block until a condition is satisfied. When the condition becomes false, the statement immediately after the loop is executed.

The else clause is only executed when your while condition becomes false. If you break out of the loop, or if an exception is raised, it won't be executed.

If else like this:

if condition:

```
# execute these statements
```

else:

```
# execute these statements
```

GALGOTIAS
UNIVERSITY

And while loop like this are similar

while condition:

```
# execute these statements
```

else:

```
# execute these statements
```

```
#Python program to illustrate
```

```
# combining else with while
```

```
count = 0
```

```
while (count < 3):
```

```
    count = count + 1
```

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

```
else:
```

```
    print("In Else Block")
```

Output:

Hello Geek

Hello Geek

Hello Geek\n

Else Block



GALGOTIAS
UNIVERSITY

Single statement while block:

Just like the if block, if the while block consists of a single statement then we can declare the entire loop in a single line as shown below:

```
# Python program to illustrate  
# Single statement while block  
count = 0  
while (count == 0):  
    print("Hello Geek")
```

GALGOTIAS
UNIVERSITY

for in Loop:

For loops are used for sequential traversal. For example: traversing a list or string or array etc. In Python, there is no C style for loop, i.e., for (i=0; i<n; i++). There is “for in” loop which is similar to [for each](#) loop in other languages. Let us learn how to use for in loop for sequential traversals.

Syntax:

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

It can be used to iterate over a range and iterators.

GALGOTIAS
UNIVERSITY

School of Basic and Applied Sciences

Course Code : BSCM 304

Course Name: Programming Using Python

```
# Python program to illustrate  
# Iterating over range 0 to n-1  
n = 4  
for i in range(0, n):  
    print(i)
```

Output :

0
1
2
3



GALGOTIAS
UNIVERSITY

Example:

Iterating over a list

```
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)
```



GALGOTIAS
UNIVERSITY

Output:

List Iteration

Geeks

for

geeks

Tuple Iteration

Geeks

for

geeks



GALGOTIAS
UNIVERSITY

School of Basic and Applied Sciences

Course Code : BSCM 304

Course Name: Programming Using Python

Example:

```
# 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]))
```



GALGOTIAS
UNIVERSITY

School of Basic and Applied Sciences

Course Code : BSCM 304

Course Name: Programming Using Python

Output:

String Iteration

G

e

e

e

s

Dictionary Iteration

xyz 123

abc 345



GALGOTIAS
UNIVERSITY

References:

1. Introduction to Computation and Programming using Python, by John Guttag, PHI Publisher
2. Fundamentals of Python first Programmes by Kenneth A Lambert, Copyrighted material Course Technology Inc. 1 st edition (6th February 2009)
3. <https://www.tutorialspoint.com/python/index.htm>
4. <https://www.geeksforgeeks.org/python-programming-language>

GALGOTIAS
UNIVERSITY

*****END OF THE LECTURE*****

*****THANK YOU*****

GALGOTIAS
UNIVERSITY