In this lecture, The sixth part of the first module was covered. Which is “Loops”

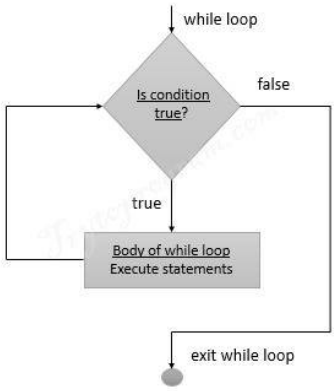
Loops are basically a technique to do mundane iteration tasks multiple times. Aloop contains 3 parts. Those are,

- **Initialization:** This is the gateway point of a loop. Basically, It’s the point where a loop starts.
- **Condition:** This is basically a controlling technique. If we need to stop a loop or run it for a specific number of times, Then that’s the condition on which the loop is dependent.
- **Increment/Decrement:** According to a situation, We can either make the loop go down from above which is known as “decrease” and the vice versa is known as “increase”.

There are 3 types of loops in Python. Those are,

- While loop
- For loop
- Nested Loop

While Loop:



This loop's role is basically the "AS LONG AS" part of a code. It ensures that the code's snippet(Where it is used) runs or iterates the code until the condition/statement becomes untrue or false. It's not necessary to have a specific number on which based upon that code will run. We can also run a loop infinite times in a code using a while loop.

Syntax:

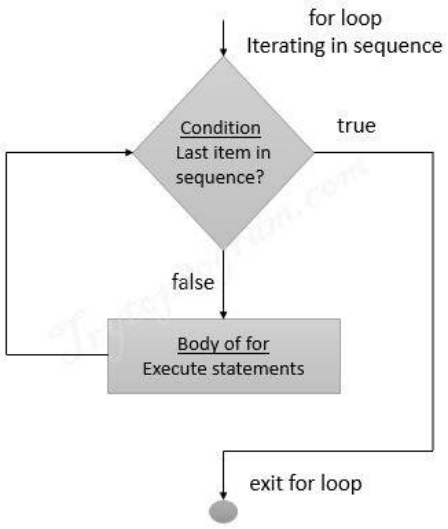
```
while(expression):
        statement(s)
```

EG:

```
i = 5
while (i > 8):
  print ('This is while loop')
  i++
```

Here in this program while loop won't be executed because in the initial test i > 8 will return FALSE as the value of i is 5.

For Loop:



This loop's role is basically the "TILL/UNTIL" part of a code. It ensures that the code's snippet(Where it is used) runs or iterates the code a specific number of times on which based upon that code will run.

Syntax:

```
for var in sequence:
    statement(s)
```

Note:

Range function syntax: range(start, stop, step)

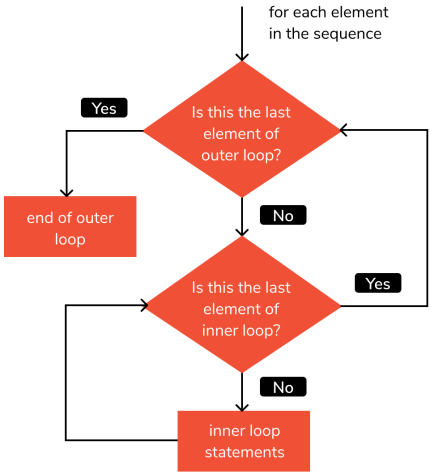
EG:

```
string = 'YOLO'
for k in string:
  print(k)
```

Output

```
Y
O
L
O
```

Nested Loop:



This basically means a loop inside a loop or loops inside loops. If one or multiple loop contains inside another loop, then the loop that is coontaining all the loops inside it and not being inside of any loop itself is called NESTER LOOP.

Syntax:

```
for variable1 in sequence:
    for variable2 in sequence:
        statement (s)
```

EG:

```
for i in range(1, 4):
  print("i =", i, "->", end = " ")
  for j in range(1, 11):
    print("{:2d}".format(i * j), end = " ")
  print()
```

Output:

```
i = 1 -> 1 2 3 4 5 6 7 8 9 10
i = 2 -> 2 4 6 8 10 12 14 16 18 20
i = 3 -> 3 6 9 12 15 18 21 24 27 30
```

Sources:

1. http://www.trytoprogram.com/python-programming/python-while-loop/
2. http://www.trytoprogram.com/python-programming/python-for-loop/
3. https://www.faceprep.in/python/nested-loops-in-python/