FOR LOOP

Like all the other functions we have seen so far in the video tutorials, for loop is also just a programming function that iterates a statement or a number of statements based on specific boundaries under certain defined conditions, that are the basis of the loop.

Note that the statement that the loop iterates must be present inside the body of the loop.

Regarding loops, iteration means going through some code again and again. In programing it has the same meaning, the only difference is that the iteration depends upon certain conditions and upon its fulfillment, the iteration stops, and the compiler moves forward.

The concept of the loop could be easily understood using an example of songs playlist. When we like a song, we set it on repeat, and then it automatically starts playing again and again. The same concept is used in programming, we set a part of code for looping and the same part of the code executes until the certain condition that we provided is fulfilled. You must be thinking that in song playlist the song keeps on playing until we stop it, the same scenario can be made in case of loops, if we put a certain condition that the loop could not fulfill, then it will continue to iterate endlessly until stopped by force.

An example of where loop could be helpful to us could be in areas where a lot of data has to be printed on the screen and physically writing that many printing statements could be difficult or in some cases impossible.

Why do we use loops?

- Complex problems can be simplified using loops
- Less amount of code required for our program
- Lesser code so lesser chance or error
- Saves a lot of time
- Can write code that is practically impossible to be written
- Programs that require too many iterations such as searching and sorting algorithms can be simplified using loops

How to write a for-loop?

For loop basically depends upon the elements it has to iterate instead of the statement being true or false.

Syntax

for <variable> in <sequence>:

Explanation

Firstly, the first value will be assigned in the variable. Secondly all the statements in the body of the loop are executed with the same value.

Thirdly, once step second is completed then variable is assigned the next value in the sequence and step second is repeated.

Finally, it continues till all the values in the sequence are assigned in the variable and processed.

Example

```python
for i in range(6):
    print(i)



# -----OUTPUT-----
0
1
2
3
4
5
```

```python
num = 2
for a in range(1, 6):
    print(num * a)



# -----OUTPUT-----
2
4
6
8
10
```

Example to find Sum of first 10 Numbers

```python
sum = 0

for i in range(1, 11):
    sum += i

print("Sum is = %d" % sum)



# -----OUTPUT-----
Sum is = 55
```

Using For-Loop with IF and BREAK Statement

Break statement stops the Loop at that Position.

```
for i in range(1, 20):
    if i == 5:
        break
    print(i)



# -----OUTPUT-----
1
2
3
4
```

Using For-Loop with IF and CONTINUE Statement

Continue statement skips all the code written below that and goes to next loop.

```
for i in range(1, 10):
    if i == 4:
        continue
    print(i)



# -----OUTPUT-----
1
2
3
5
6
7
8
9
```

Steps in For-Loop

Runs the loop adding step value to previous Loop value.

```python
for i in range(1, 20, 2):
    print(i)



# -----OUTPUT-----
1
3
5
7
9
11
13
15
17
19
```

Negative For-Loop

```python
for i in range(10, 1, -1):
    print(i)


# -----OUTPUT-----
10
9
8
7
6
5
4
3
2
```

Questions

1. Print first 10 Natural numbers.
2. Print all the even numbers from 1 to 100.
3. Find sum of N numbers where N is input from User.
4. Check whether number entered by User is Prime or Not.
5. Check whether number entered by User is Armstrong or not.

HW

1. Print multiplication table of a number entered by user.
2. Program to display all the prime numbers within a range.

Nested For-Loop

Loops defined within another Loop are called Nested Loops. Nested loops are used to iterate matrix elements or to perform complex computation.

When an outer loop contains an inner loop in its body it is called Nested Looping.

Syntax

for <expression>:

      for <expression>:

           Body

Example 1

```
for i in range(1, 4):
    for j in range(1, i+1):
        print(i)
print()


# -----OUTPUT-----
1
2
2
3
3
3
```

Example 2

```
for i in range(1, 6):
    for j in range(5, i-1, -1):
        print("*", end="")
    print()


# -----OUTPUT-----
*****
****
***
**
*
```

Questions

1. Make the following Patterns

   a)
   ```
   *
   **
   ***
   ****
   *****
   ```

   b)
   ```
   1
   1 2
   1 2 3
   1 2 3 4
   1 2 3 4 5
   ```

   c)
   ```
   5
   5 4
   5 4 3
   5 4 3 2
   5 4 3 2 1
   ```

   d)
   ```
   5 5 5 5 5
   4 4 4 4
   3 3 3
   2 2
   1
   ```

   e)
   ```
           1
         2 1
       3 2 1
     4 3 2 1
   5 4 3 2 1
   ```

   f)
   ```
           5
         4 5
       3 4 5
     2 3 4 5
   1 2 3 4 5
   ```

2. Write a program to display following Output

   a) 1 2 3 4 5 6 7 8 9 10

   b) 2 4 6 8 10 12 14 16 18 20

   c) 3 6 9 12 15 18 21 24 27 30

   d) 4 8 12 16 20 24 28 32 36 40

   e) 5 10 15 20 25 30 35 40 45 50