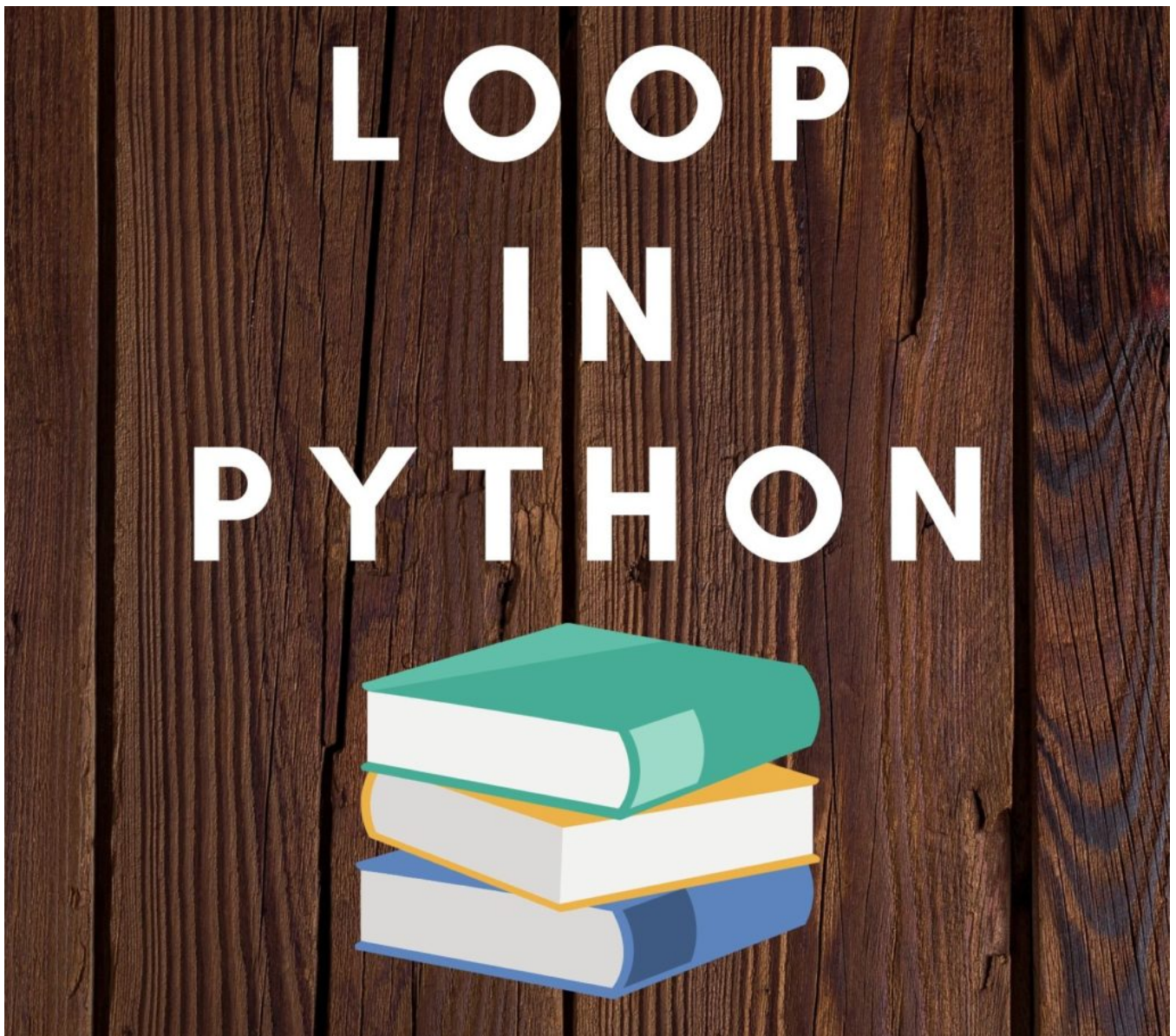




Python Tutorial: Do-While loop in Python

Python / March 14, 2020



This is the **Python Tutorial: Do-While loop in Python**. In this Python Tutorial, you will learn different loop used in python. At the end of this tutorial, you will have full knowledge of Loop in Python.

Hello, & Welcome!

Part 7- Python Tutorial: Loop in Python?

In this article of Python Tutorial, you will learn following-

1. For Loop
2. While Loop
3. Nested Loop
4. Do-While loop
5. Break Statement
6. Continue Statement

For Loop-

For loop is used to perform some operation repeatedly until the condition gets false.

It is used in traversing a list, strings, and arrays.

Suppose we create a list called colors-

- `colors=['red','green','yellow','black']`

And we want to traverse the whole list, then we can for loop.

Syntax of for loop is-

```
for iteration_variable_name in List_Name:  
    statements(S)
```

Let's take an example of colors list, and perform traversing-

```
colors=['red','green','yellow','black']  
for c in colors:  
    print(c)
```

OUTPUT-

```
red  
green  
yellow  
black
```

- You can use any name for iteration_variable_name. Here we use 'c'.

Slicing via For Loop-

You can perform slicing as we have done in [List](#) and [string](#). If you wanna get some particular element from the whole list, then you can perform this task via for loop.

Let's see in the example below-

```
colors=['red','green','yellow','black']  
for c in colors[1:3]:  
    print(c)
```

OUTPUT-

```
green  
yellow
```

- You are wondering, How it is printing green and yellow as output?
- Don't worry! ... I'll tell you how?
- Here, we are getting only 'green' and 'yellow', because 'green' index position is 1 and 'yellow' index position is 2. And we give colors[1:3]. It starts at index 1(green) to index 2(yellow).

While Loop-

While loop runs until the certain condition is true, but as the condition becomes false, it terminates.

Syntax of while loop-

```
while condition:  
    statement(S)
```

Let's have a look at while loop example-

```
age=25  
num=0  
while num<age:  
    print(num)  
    num+=1
```

OUTPUT -

```
1  
2  
3  
.  
.  
.  
24
```

- It will print num till 24 times because at 25 it gets terminated, as the condition becomes false.

Nested Loop-

You can perform nesting in loops as in **if-else**, which means you can use a loop inside another loop.

You can use nesting in for loop as well as while loop.

Syntax of Nested For loop-

```
for iteration_variable_name in List_Name:
    for iteration_variable_name in List_Name:
        statements(S)
        statements(S)
```

Syntax of Nested While loop-

```
while condition:
    while condition:
        statement(S)
        statement(S)
```

- **Note-** You can perform nesting of a loop by using any loop inside another type of loop. For example, you can use while loop inside for loop and similarly for loop inside while loop.

Do-While loop-

Do you know? Python doesn't provide a feature of a Do-While loop, But if you wanna use it in python, then you can create a program using a Do-While loop

The Do-While loop works similarly as a while loop but with one difference.

The Do-While loop first executes and then check the condition, which means it executes once, no matter the condition is true or false.

Syntax of Do-While loop-

```
do{
    statements
} while(condition):
```

Break Statement-

If you wanna break a code, or want to stop the loop before traversing through the whole list, you can use Break Statement. In other words, Break statement abort the execution of current loop.

Let's have a look at an example-

```
colors=['red','green','yellow','black']
for c in colors:
    print(c)
    if c=="green":
        break
```

OUTPUT -

red
green

- Here, The Break statement, break the loop as c="green".

Continue Statement-

If you use Continue Statement in the loop, then it stops the current iteration of the loop, and continue with the next. It sounds a bit confusing, but don't worry!... Let's see in the example.

```
colors=['red','green','yellow','black']
for c in colors:
    if c=="green":
        continue
    print(c)
```

OUTPUT -

red
yellow
black

- Look in the output, it prints only three colors- red, yellow and black. Not green. Because we use Continue Statement, which stops the current iteration as

```
if c=="green":  
    continue
```

- And continue with next. That's why it prints red, yellow and black.
- If you are still confused then in simple words continue statement skip the 'green' color because we give a condition if c='green', then skip the green color and print the rest of the colors.
- I hope now you understand Continue Statment. 😊

That's all for Loop in Python, I hope now you have a better understanding of Loop in Python.

congratulations! You successfully learned Loop in [Python](#).



In the next tutorial, we will start learning [Ranges in Python](#)

Till then, Enjoy Learning Python!

All the Best!

Are you ML Beginner and confused, from where to start ML, then read my BLOG – [How do I learn Machine Learning?](#)

If you are looking for Machine Learning Algorithms, then read my Blog – [Top 5 Machine Learning Algorithm](#).

If you are wondering about Machine Learning, read this Blog- [What is Machine Learning?](#)

Thank YOU!

Though of the Day...

“Live as if you were to die tomorrow. Learn as if you were to live forever.”

– Mahatma Gandhi

[← Previous Post](#)

[Next Post →](#)

Leave a Comment

Your email address will not be published. Required fields are marked *

Type here..

Name*

Email*

☐ Save my name, email, and website in this browser for the next time I comment.

Post Comment »

Follow Us on Twitter-



Follow us on Pinterest-



Our Facebook Group-



Copyright © 2020 MLTut

Powered by MLTut