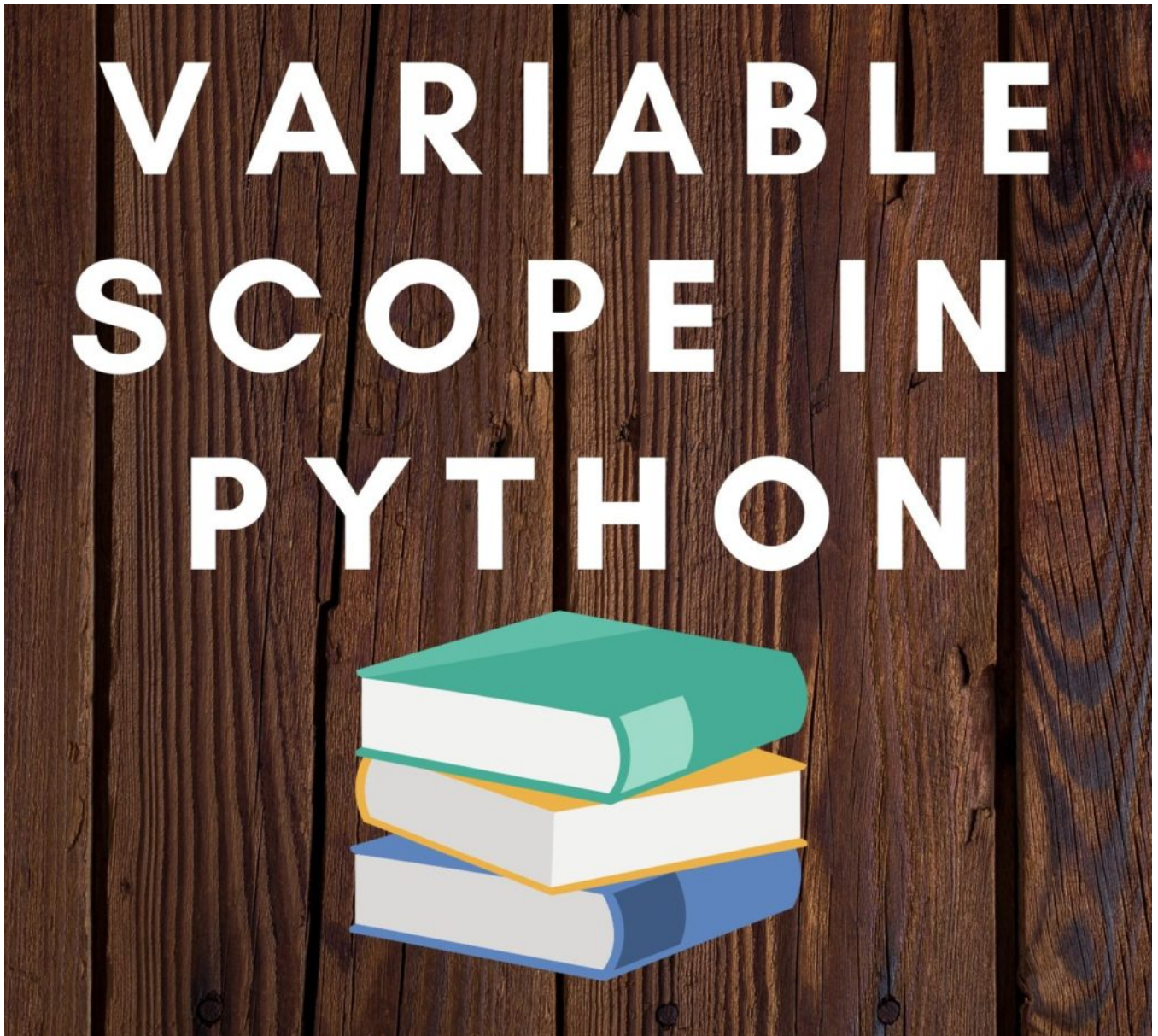




Variable Scope in Python-Complete Tutorial

Python / March 18, 2020



This **Variable Scope in Python-Complete Tutorial** is made for you to give complete knowledge of the Variable Scope in Python. At the end of this tutorial, you will have full knowledge of the **Variable Scope in Python**.

Hello, & Welcome! to the **Variable Scope in Python-Complete Tutorial**.

Part 10- Variable Scope in Python-Complete Tutorial

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

1. What is Variable Scope?
2. Types of Variable Scope.
3. Python Global Scope.
4. Python Local Scope.
5. Global Keyword.

What is Variable Scope?

Variable Scope in Python shows the visibility of the variable in the code. The visibility of variables declares on the basis of the variable definition location. Where the variable is defined, its scope decides at the same time.

You will understand the whole concept in a few minutes.

Types of Variable Scope-

There are basically two types of Variable Scope-

1. Local Scope
2. Global Scope

Python Global Scope-

Global Scope is when you define a variable outside the function. The accessibility of the global variable is inside the function as well as outside the function.

Let's have a look at the example of Global Variable-

```
my_name='Alex'           #global variable
def print_name():
    print('name inside the function',my_name)
print_name()
print('name outside the function',my_name)
```

OUTPUT-

```
name inside the function-Alex
name outside the function-Alex
```

Here, we declare variable **my_name** as a global variable. This means we can use **my_name** variable inside the function as well as outside the function. The scope of the variable is global because we define **my_name** outside the function.

After looking at the example, I hope you have a clear idea about the Global Variable definition. Let's revise it once- **The variable which is declared outside the function is known as global variable.**

Python Local Scope-

Local Scope in python is when you define a variable inside the function.

If you define the variable inside the function so it is a **local variable**. Which means it is only accessible inside the function, and not outside the function.

Let's understand with the help of an example-

```
my_name='Alex'
def print_name():
    my_name='John'
    print('name inside the function',my_name)
print_name()
print('name outside the function',my_name)
```

OUTPUT-

```
name inside the function-John
name outside the function-Alex
```

Here, **my_name='Alex'** is a Global variable, but **my_name='John'** is a local variable. Which overrides the global variable value inside the function, but not outside the function. Therefore, as you see in the output 'John' print as the name inside a function. And 'Alex' print as a name outside the function.

You get an error when you don't define the variable as global (outside the function). But try to access a local variable in a global scope.

Let's understand with the help of an example-

```
def print_name():
    my_name='John'
print_name()
print('name outside the function',my_name)
```

OUTPUT-

```
NameError: name 'my_name' is not defined.
```

You will get an error because, in the example, we define **my_name** inside the function. Therefore its scope is local and we can not access it globally.

Global Keyword-

As we saw in the global keyword example, **my_name='John'** override the value of **my_name='Alex'** inside the function. But if you want that it doesn't override the value inside the function, then you can do it by using **Global Keyword**.

Let's see how Global Keyword works-

```
my_name='Alex'
def print_name():
    global my_name          #define local variable as global
    my_name='John'
```

```
print('name inside the function',my_name)
print_name()
print('name outside the function',my_name)
```

OUTPUT-

```
name inside the function-Alex
name outside the function-Alex
```

As you can see, by using **Global Keyword**, the value of the local variable is not printed. The only global variable value is printed two times.

Due to **Global Keyword**, the value gets changes globally inside and outside the function.

That's all for the Variable Scope in Python. I hope now you have a better understanding of the Variable Scope in Python.

Congratulations! You successfully learned the Variable Scope in [Python](#).



In the next tutorial, we will start learning [Dictionaries 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