Lambda Function and Global keyword

BY ROHIT S. AGRAWAL

Anonymous/Lambda Function

In Python, an anonymous function is a function that is defined without a name.

While normal functions are defined using the ‘def’ keyword in Python, anonymous functions are defined using the lambda keyword

Hence, anonymous functions are also called lambda functions.

Syntax of Lambda Function

lambda arguments:expression

Lambda Function

Try to create a lambda function

squares = lambda a:a**2

print(squares(3))

This function has no name.

It returns a function object which is assigned to the identifier squares

Use of Lambda Function

We use lambda functions when we require a nameless function for a short period of time.

In Python, we generally use it as an argument to a higher-order function (a function that takes in other functions as arguments)

Lambda functions are used along with built-in functions like filter(),map() etc.

Lambda with filter()

The filter() function in Python takes in a function and a list as arguments.

The function is called with all the items in the list and a new list is returned which contains items for which the function evaluates to True.

Here we use the filter() function to filter out only even numbers from a list.

```
# Program to filter out only the even items from a list

my_list = [1, 5, 4, 6, 8, 11, 3, 12]

new_list = list(filter(lambda x: (x%2 == 0) , my_list))

print(new_list)
```

Lambda with map()

The map() function in Python takes in a function and a list.

The function is called with all the items in the list and a new list is returned which contains items returned by that function for each item.

Here we will use of map() function to double all the items in a list.

```
# Program to double each item in a list using map()

my_list = [1, 5, 4, 6, 8, 11, 3, 12]

new_list = list(map(lambda x: x * 2 , my_list))

print(new_list)
```

Global Variables

A variable declared outside of the function or in global scope is known as a global variable.

This means that a global variable can be accessed inside or outside of the function.

```
x = "global"

def func():
    print("x inside:", x)

func()

print("x outside:", x)
```

Try to change the value of x inside the function

Local Variables

- A variable declared inside the function's body or in the local scope is known as a local variable.

```
def func():
    y = "local"
func()
print(y)
```

- Write a program to use both local and global variables in the same code.

Nonlocal Variables

Nonlocal variables are used in nested functions whose local scope is not defined.

This means that the variable can be neither in the local nor the global scope.

We use nonlocal keywords to create nonlocal variables.

```
def outer():
    x = "local"
    def inner():
        nonlocal x
        x = "nonlocal"
        print("inner:", x)
    inner()
    print("outer:", x)
outer()
```

What is the global keyword

- In python, ‘global’ keyword allows you to modify the variable outside of the current scope.
- It is used to create a global variable and make changes to the variable in a local context.
- When we create a variable inside a function, it is local by default.
- When we define a variable outside of a function, it is global by default. You don't have to use ‘global’ keyword.
- We use global keyword to read and write a global variable inside a function.
- Use of global keyword outside a function has no effect.

Lets try…

Try - Accessing global Variable From Inside a Function.

```
c = 1 # global variable

def add():
    print(c)

add()
```

Modifying Global Variable From Inside the Function

```
c = 1 # global variable

def add():
    c = c + 2 # increment c by 2
    print(c)

add()
```

Lets try..

Changing Global Variable From Inside a Function using global

```
c = 0 # global variable
def add():
    global c
    c = c + 2 # increment by 2
    print("Inside add():", c)
add()
print("In main:", c)
```