

Python from Scratch

Python Scope

Lesson 23

- **Local Scope**
- **Function Inside Function**
- **Global Scope**
- **Naming Variables**
- **Global Keyword**
- **Python - Scope Exercises**

Python Scope

A variable is only available from inside the region it is created. This is called **scope**.

Local Scope

A variable created inside a function belongs to the *local scope* of that function, and can only be used inside that function.

Example

A variable created inside a function is available inside that function:

```
def myfunc():  
    x = 300  
    print(x)  
  
myfunc()
```

Function Inside Function

As explained in the example above, the variable `x` is not available outside the function, but it is available for any function inside the function:

Example

The local variable can be accessed from a function within the function:

```
def myfunc():  
    x = 300  
    def myinnerfunc():  
        print(x)  
    myinnerfunc()  
  
myfunc()
```

Global Scope

A variable created in the main body of the Python code is a global variable and belongs to the global scope.

Global variables are available from within any scope, global and local.

Example

A variable created outside of a function is global and can be used by anyone:

```
x = 300

def myfunc():
    print(x)

myfunc()

print(x)
```

Naming Variables

If you operate with the same variable name inside and outside of a function, Python will treat them as two separate variables, one available in the global scope (outside the function) and one available in the local scope (inside the function):

Example

The function will print the local `x`, and then the code will print the global `x`:

```
x = 300

def myfunc():
    x = 200
    print(x)

myfunc()

print(x)
```

Global Keyword

If you need to create a global variable, but are stuck in the local scope, you can use the `global` keyword.

The `global` keyword makes the variable global.

Example

If you use the `global` keyword, the variable belongs to the global scope:

```
def myfunc():  
    global x  
    x = 300
```

```
myfunc()
```

```
print(x)
```

Also, use the `global` keyword if you want to make a change to a global variable inside a function.

Example

To change the value of a global variable inside a function, refer to the variable by using the `global` keyword:

```
x = 300
```

```
def myfunc():  
    global x  
    x = 200
```

```
myfunc()
```

```
print(x)
```