Python Fundamentals day 6

Today's Agenda

- Introduction to Functions
- Why Functions?
- Different types of functions
- User Defined functions in detail

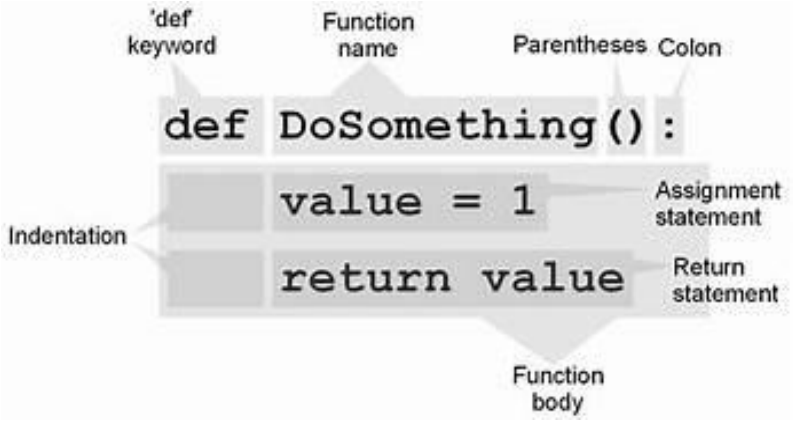
Introduction to Functions

Definition: A Function is a set of statements that takes inputs, performs some specific task and produces output.

How do i declare a function in python?

The four steps to defining a function in Python are the following:

- Use the keyword def to declare the function and follow this up with the function name.
- Add parameters to the function: they should be within the parentheses of the function. End your line with a colon.
- Add statements that the functions should execute.
- End your function with a return statement if the function should output something.



Why Functions?

After getting to know how a python function looks like, let us now understand why functions by considering few examples.

Let us consider two operations of adding two numbers and dividing two numbers as shown below.

```
a = 10
b = 20
c = a+b
print(c)

d = 100
e = 10
f=d/e
print(f)
```

Above lines of codes will perform addition and division of two numbers, but if in case you want to perform addition of two numbers again, the only option is to copy the same lines of code and use them again as shown below:

```
a = 10
b = 20
c = a+b
print(c)

d = 100
e = 10
f=d/e
print(f)
```

```
a = 10
b = 20
c = a+b
print(c)
```

But as a programmer this is not the right approach which one should follow, the efficient way of doing this is by following dry approach which stands for DO-NOT-REPEAT-YOURSELF.

So, the solution is including the lines of code inside a block and give it a suitable name. That way you can use the same block of code multiple times by making use of name given to it.

Using the above approach, we can achieve:

Reusability and Modularity.

Reusability refers to resuing the same block of codes multiple times.

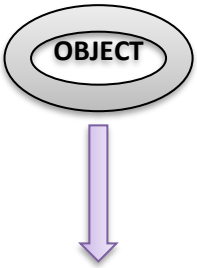
Modularity refers to dividing the entire code into different logical blocks of code.

From object orientation perspective, the behaviour of an object is taken care by functions in python.

```
a = 10
b = 20
c = a+b
print(c)
```

```
d = 100
e = 10
f=d/e
print(f)
```

1.Reusability
2.Modularity



Types of Functions in python

Python consists of four types of functions:

1. User Defined Functions
2. Built-in-functions.
3. Lambda functions.
4. Recursive Functions.

User Defined Functions: These functions are defined or created by user.

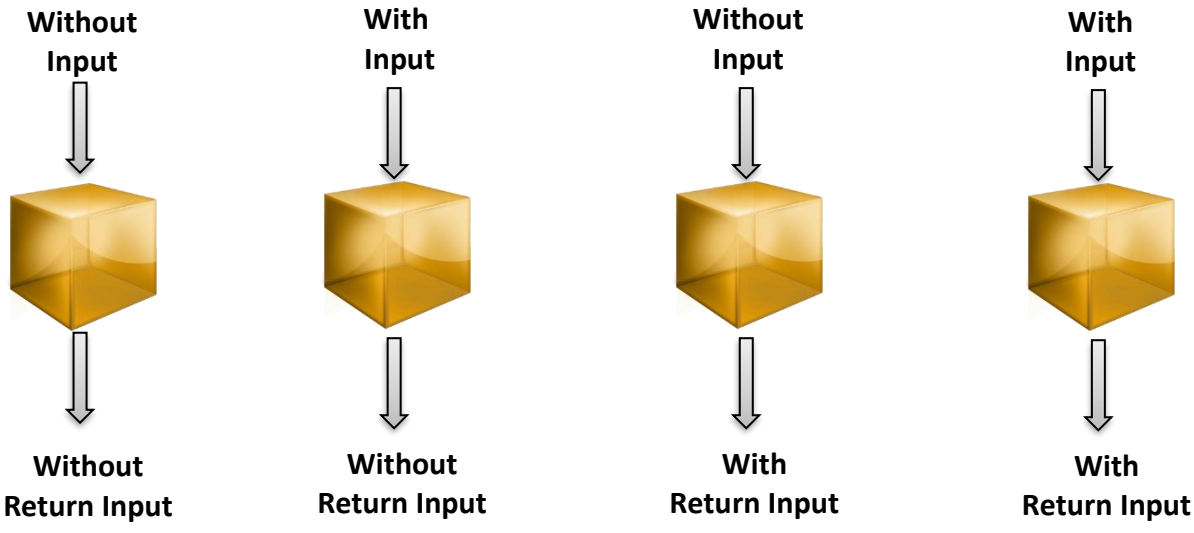
Built-in-functions: These functions are already defined in Python libraries and we can call them directly.

Lambda functions: A lambda function can take any number of arguments, but can only have one expression.

Recursive Functions: A Python function which is defined to call itself during its execution is known as python recursive function.

User defined Functions

A user defined function in python can be declared in four different ways as shown below:



1.Function that takes no input and does not return any output

Below is the code which consists of a function mul() which falls in the first category of user defined functions which is a function that takes no input and returns no output.

In the code given below there is not only mul(), it also has main() inside which we are calling mul().

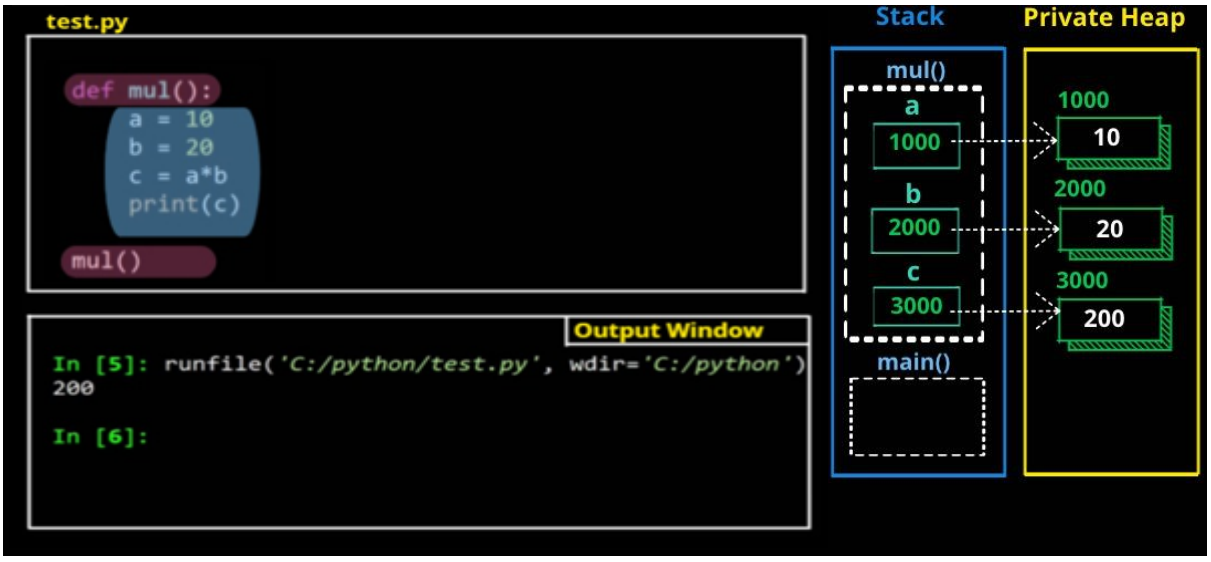
All such statements which are not present inside any functions are automatically placed within the main() function and is automatically called in python. The moment a method gets called its stack frame gets created.

The first function that gets called in a python program is always the main().

Whenever a main function gets called its stack frame gets created on stack which simply shows main() has begun execution.

Inside main(), we are calling mul() which results in the creation of stack frame of mul() on stack.

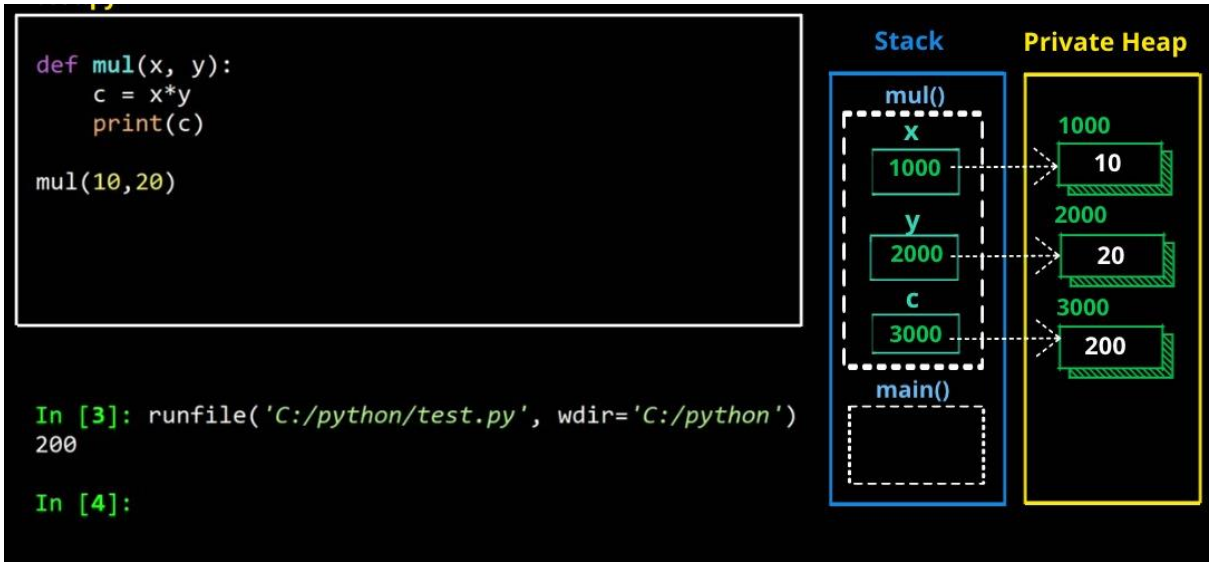
Whatever is present at the top of stack is always the one which is currently executing and hence mul() starts executing inside which we are creating three objects a,b,c. The references are created inside the stack frame of mul() and objects are created on private heap. Once mul() finishes execution, its stack frame gets deleted and then the control is given back to main(). After the execution of main(), its stack frame also gets deallocated memory on stack. If all the stack frames are destroyed, all the references also gets deleted. Now the objects on heap do not have any reference pointing to them, once pyhton encounters there no references pointing to objects, it treats those objects as garbage objects and deletes them. To perform deletion of garbage objects, python provides a software called as garbage collector which deallocates memory for all the garbage objects.



2.Function that takes input and does not return any output

Below is the code which consists of a function mul() which falls in the second category of user defined functions which is a function that takes input and returns no output.

While calling a method that requires arguments, one must pass number of arguments accepted by the methods as shown below where we are passing 10,20 while calling mul(10,20). 10 and 20 are now stored inside x and y.



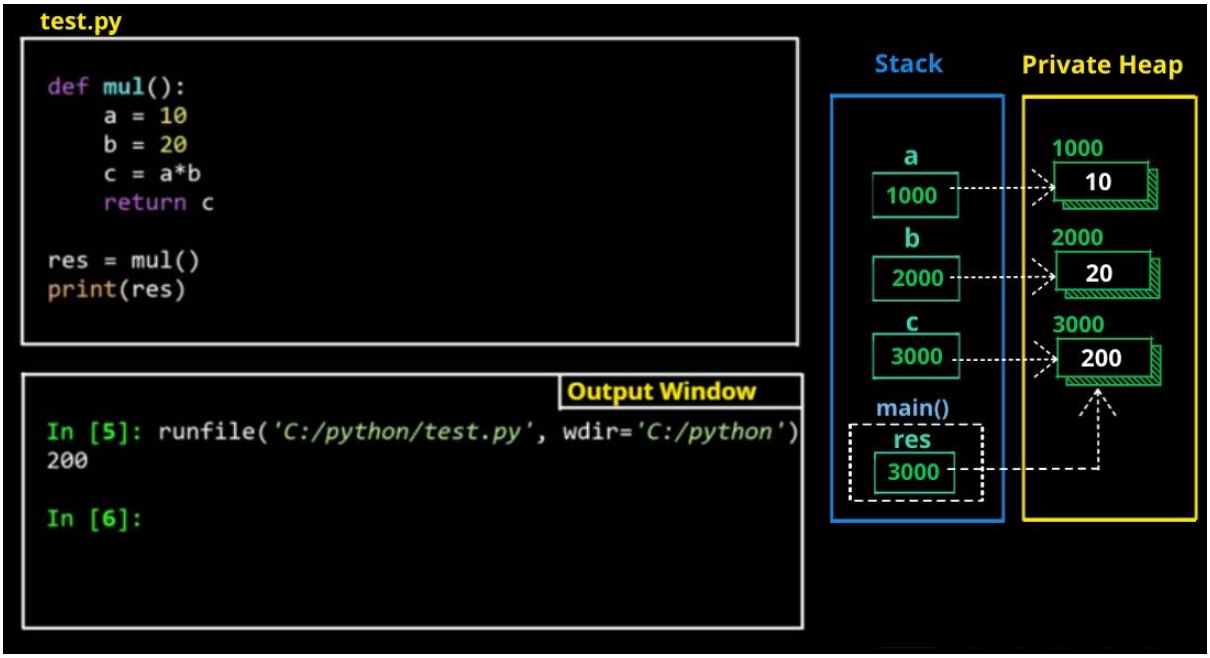
3. Function that takes no input but returns output

Below is the code which consists of a function mul() which falls in the third category of user defined functions which is a function that takes no input and returns output.

One must make use of return statement inside the function to return a value.

Once return statement gets executed, control goes to the statement which had called that function. Now it is the choice of the caller to collect the value returned by function or not. In the below code the returned value is collected inside res which is created inside the stack frame of main() as shown in figure below.

Note: Pyhton functions can return muliple values.



4.Function that takes input and returns output

Below is the code which consists of a function mul() which falls in the fourth category of user defined functions which is a function that takes two inputs x,y and returns output.

Based on situations, we must make use of different functions.

For example, you need your bank account details and to do so you must give your details as input upon which your details will be returned to you.

This way, based on different scenarios, different types of functions should be used to achieve the required task.