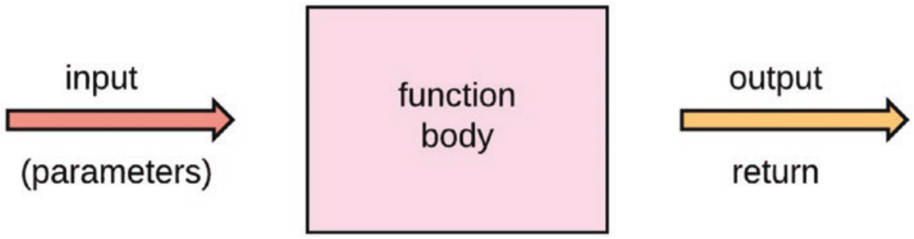

Figure 9-5. Functions

A function receives data into its parameter list during a function call. The inputed data is used to complete the function execution. At the end of its execution, a function always returns a result – this result could be 'None' or a specific data value.

Functions are treated as first-class objects in Python. That means a function can be passed as data into another function, the result of a function execution can also be a function, and a function can also be stored as a variable.

Functions are visualized as a black box that receives a set of objects as input, executes some code, and returns another set of objects as output.

User-Defined Functions

A function is defined using the def keyword. The syntax for creating a function is as follows:

```
def function-name(parameters):
    statement(s)
```

Let's create a simple function:

```
def squares(number):
    return number**2

squares(2)
'Output': 4
```

Here's another function example:

```
def _mean_(*number):
    avg = sum(number)/len(number)
    return avg

_mean_(1,2,3,4,5,6,7,8,9)
'Output': 5.0
```

The ∗ before the parameter number indicates that the variable can receive any number of values, which is implicitly bound to a tuple.

Lambda Expressions

Lambda expressions provide a concise and succinct way to write simple functions that contain just a single line. Lambdas now and again can be very useful, but in general, working with **def** may be more readable. The syntax for lambdas are as follows:

```
lambda parameters: expression
```

Let's see an example:

```
square = lambda x: x**2
square(2)
'Output': 4
```

Packages and Modules

A module is simply a Python source file, and packages are a collection of modules. Modules written by other programmers can be incorporated into your source code by using **import** and **from** statements.

import Statement

The **import** statement allows you to load any Python module into your source file. It has the following syntax:

```
import module_name [as user_defined_name][,...]
```

where the following is optional:

```
[as user_defined_name]
```

Let us take an example by importing a very important package called **numpy** that is used for numerical processing in Python and very critical for machine learning.