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.

```
import numpy as np

np.abs(-10)   # the absolute value of -10
'Output': 10
```

from Statement

The **from** statement allows you to import a specific feature from a module into your source file. The syntax is as follows:

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

Let's see an example:

```
from numpy import mean


mean([2,4,6,8])
'Output': 5.0
```

This chapter provides the fundamentals for programming with Python. Programming is a very active endeavor, and competency is gained by experience and repetition. What is presented in this chapter provides just enough to be dangerous.

In the next chapter, we'll introduce NumPy, a Python package for numerical computing.