PACKAGE

We create two simple files a.py and b.py just for the sake of filling the package with modules.

The content of a.py:

**def** bar():

**print**("Hello, function 'bar' from module 'a' calling")

The content of b.py:

**def** foo():

**print**("Hello, function 'foo' from module 'b' calling")

We will also add an empty file with the name \_\_init\_\_.py inside of simple\_package directory.

Let's see what happens, when we import simple\_package from the interactive Python shell, assuming that the directory simple\_package is either in the directory from which you call the shell or that it is contained in the search path or environment variable "PYTHONPATH" (from your operating system):

**import** **simple\_package**

simple\_package/a

#### **OUTPUT:**

**---------------------------------------------------------------------------**

**NameError** Traceback (most recent call last)

**<ipython-input-3-347df8a711cc>** in <module>

**----> 1** simple\_package/a

**NameError**: name 'a' is not defined

simple\_package/b

#### **OUTPUT:**

**---------------------------------------------------------------------------**

**NameError** Traceback (most recent call last)

**<ipython-input-4-e71d2904d2bd>** in <module>

**----> 1** simple\_package/b

**NameError**: name 'b' is not defined

We can see that the package simple\_package has been loaded but neither the module "a" nor the module "b"! We can import the modules a and b in the following way:

**from** **simple\_package** **import** a, b

a.bar()

b.foo()

#### **OUTPUT:**

Hello, function 'bar' from module 'a' calling

Hello, function 'foo' from module 'b' calling

As we have seen at the beginning of the chapter, we can't access neither "a" nor "b" by solely importing simple\_package.

Yet, there is a way to automatically load these modules. We can use the file \_\_init\_\_.py for this purpose. All we have to do is add the following lines to the so far empty file \_\_init\_\_.py:

import simple\_package.a

import simple\_package.b

It will work now:

**import** **simple\_package**

simple\_package.a.bar()

simple\_package.b.foo()

#### **OUTPUT:**

Hello, function 'bar' from module 'a' calling

Hello, function 'foo' from module 'b' calling