Packages (1A)

Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using OpenOffice.

Package (1)

modules are files
containing Python <u>statements</u> and <u>definitions</u>,
like <u>function</u> and <u>class</u> <u>definitions</u>.

how to <u>bundle</u> <u>multiple</u> modules together
to form a package.

a package is basically a directory
      with Python files
      and a file with the name **__init__.py**.

every directory inside of the Python path,
      contains a file named **__init__.py**,
will be treated as a package by Python.

several modules into a package.

Package (2)

packages are a way of
structuring Python's module namespace
by using "dotted module names".

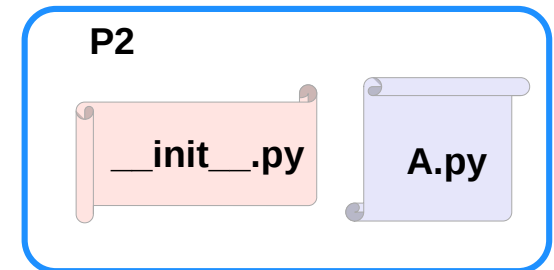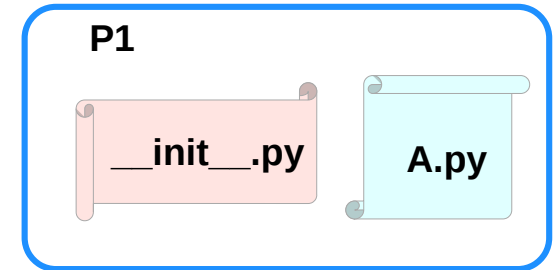**A.B** stands for
a submodule named **B**
in a package named **A**.

two different packages like **P1** and **P2**
can both have modules with the same name,
let's say **A**, for example.

The submodule **A** of the package **P1** and
the submodule **A** of the package **P2** can be totally different.

P1.A
P2.A

A package is imported like a "normal" module.

P1

__init__.py   A.py

P2

__init__.py   A.py

Creating a package (1)

Need a directory.

The name of this directory will be
the name of the package,

assume we want to create "simple_package".
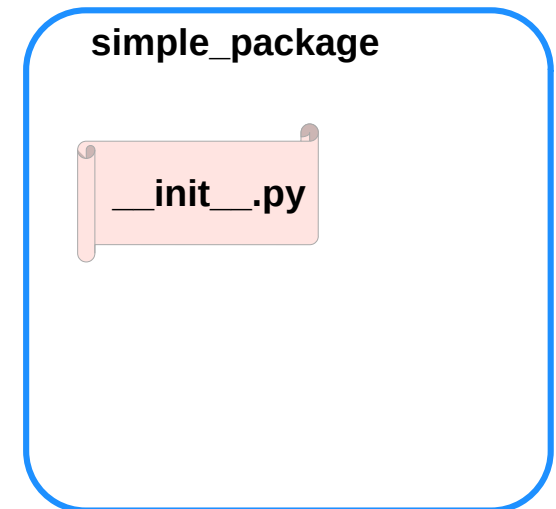
must create directory "simple_package"
and this directory needs to contain the "__init__.py" file

this file can be empty, or
can contain valid Python code.

this code will be executed when a package is imported,

so it can be used to initialize a package,

e.g. to make sure that some other modules are imported or
some values set.

simple_package

__init__.py

Young Won Lim
4/5/24

Creating a package (2)

put all of the Python files which will be the submodules
into the directory for a package.

create two simple files **a.py** and **b.py**

a.py:

```
def bar():
    print("Hello, function 'bar' from module 'a' calling")
```
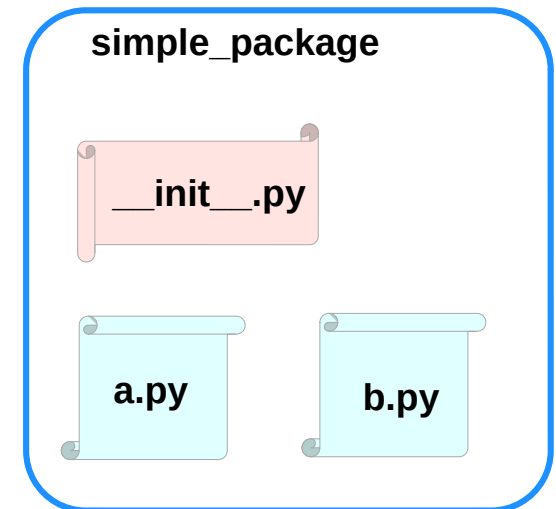
b.py:

```
def foo():
    print("Hello, function 'foo' from module 'b' calling")
```

an empty file with the name __**init**__.py
inside of simple_package directory

__**init**__.py:

```
empty file
```

simple_package

__init__.py

a.py   b.py

Creating a package (3-1)

import **simple_package** from the interactive Python shell,

assuming that the directory **simple_package** is

either in the <u>directory</u> from which you <u>call the shell</u> or

that it is contained in the search path or

environment variable "PYTHONPATH" (from your operating system):

simple_package

__init__.py

a.py   **b.py**

https://www.w3schools.com/python/python_modules.asp

import simple_package
simple_package/a

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

NameError   Traceback (most recent call last)
<ipython-input-4-e71d2904d2bd> in <module>
----> 1 simple_package/b
NameError: name 'b' is not defined

simple_package

__init__.py

a.py   b.py
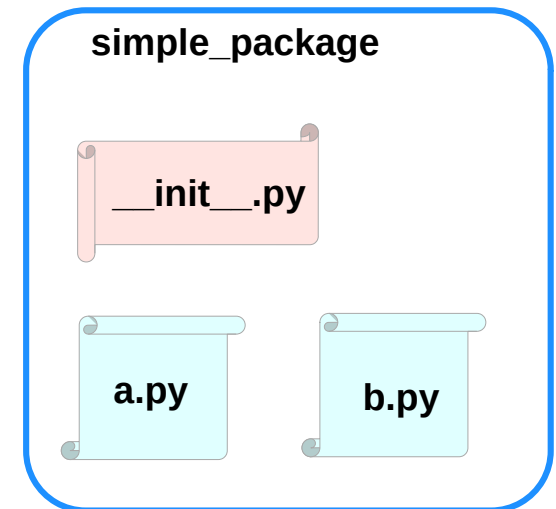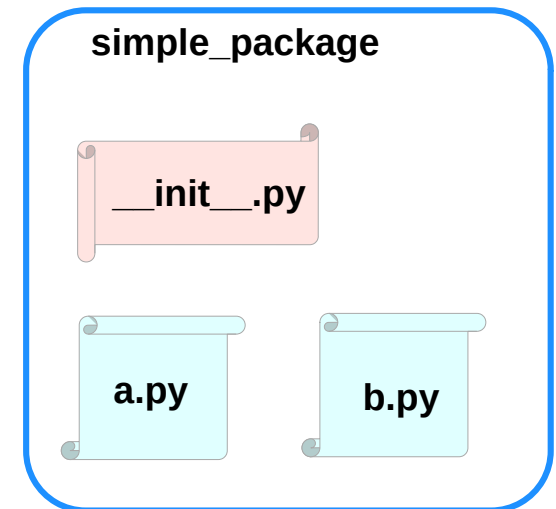
https://www.w3schools.com/python/python_modules.asp

Creating a package (4)

the package **simple_package** has been <u>loaded</u>
but <u>neither</u> the module "**a**" <u>nor</u> the module "**b**" has been <u>loaded</u>

<u>can't</u> <u>access</u> neither "**a**" nor "**b**"
by solely <u>importing</u> **simple_package**.

must <u>import</u> the modules **a** and **b** as follows

from simple_package import a, b

a.bar()
b.foo()

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

simple_package

__init__.py

a.py   b.py

Creating a package (4)

to <u>automatically</u> <u>load</u> these modules.
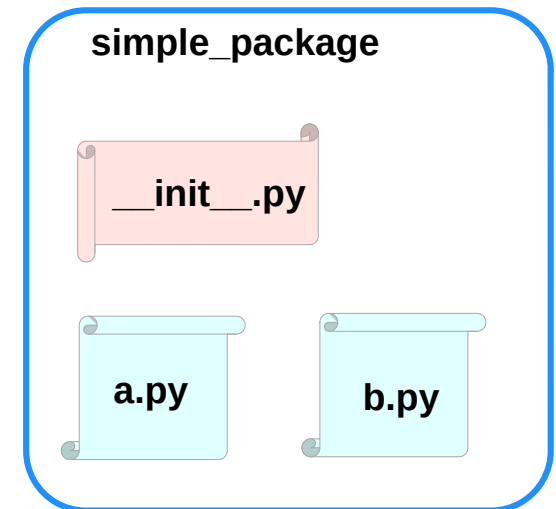
add the following lines to the file **__init__.py**:

import simple_package.a
import simple_package.b

Then

import simple_package
simple_package.a.bar()
simple_package.b.foo()

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

simple_package

__init__.py

a.py

b.py

https://www.w3schools.com/python/python_modules.asp

Package Examples (1)

```
sound
|-- effects
|   |-- __init__.py
|   |-- echo.py
|   |-- reverse.py
|   `-- surround.py
|-- filters
|   |-- __init__.py
|   |-- equalizer.py
|   |-- karaoke.py
|   `-- vocoder.py
|-- formats
|   |-- __init__.py
|   |-- aiffread.py
|   |-- aiffwrite.py
|   |-- auread.py
|   |-- auwrite.py
|   |-- wavread.py
|   `-- wavwrite.py
`-- __init__.py
```

sound

  __init__.py

  effects

    __init__.py

    echo.py
    reverse.py
    surround.py

  filters

    __init__.py

    equalizer.py
    karaoke.py
    vocoder.py

  formats

    __init__.py

    aiffread.py   **aurwrite.py**
    aifwrite.py   **wavred.py**
    auread.py   **wavwrite.py**

Young Won Lim
4/5/24

sound1

effects/__init__.py
print("effects package is getting imported!")

effects/echo.py
def func1():
    print("Function func1 has been called!")
  print("Module echo.py has been loaded!")

effects/reverse.py
def func1():
    print("Function func1 has been called!")
print("Module reverse.py has been loaded!")

effects/surround.py
def func1():
    print("Function func1 has been called!")

filters/__init__.py
print("filters package is getting imported!")

filters/equalizer.py
def func1():
    print("Function func1 has been called!")
print("Module equalizer.py has been loaded!")

filters/karaoke.py
def func1():
    print("Function func1 has been called!")
print("Module karaoke.py has been loaded!")

filters/vocoder.py
def func1():
    print("Function func1 has been called!")
print("Module vocoder.py has been loaded!")

formats/__init__.py
print("formats package is getting imported!")

formats/aiffread.py
def func1():
    print("Function func1 has been called!")
print("Module aiffread.py has been loaded!")

formats/aiffwrite.py
def func1():
    print("Function func1 has been called!")
print("Module aiffwrite.py has been loaded!")

formats/auread.py
def func1():
    print("Function func1 has been called!")
print("Module auread.py has been loaded!")

formats/auwrite.py
def func1():
    print("Function func1 has been called!")
print("Module auwrite.py has been loaded!")

formats/wavread.py
def func1():
    print("Function func1 has been called!")
print("Module wavread.py has been loaded!")

formats/wavwrite.py
def func1():
    print("Function func1 has been called!")
print("Module wavwrite.py has been loaded!")

https://www.w3schools.com/python/python_modules.asp

Young Won Lim
4/5/24