Packages (1A)

Young Won Lim
6/26/24

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

This document was produced by using OpenOffice.

Package (1)

modules are
files containing Python statements and definitions,
like function and class definitions.

to bundle multiple modules together,
create a package.

a package is
basically a directory
with several Python files (modules)
and a special file **__init__.py**

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

https://python-course.eu/python-tutorial/packages.php

Submodules in a package

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

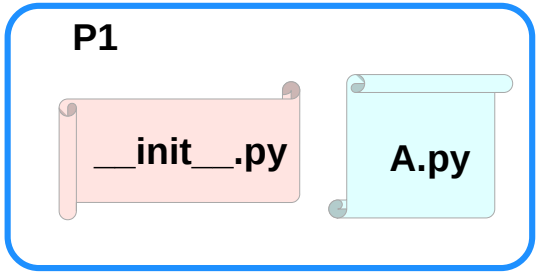
A.B stands for
a submodule named **B**
in a package named **A**.

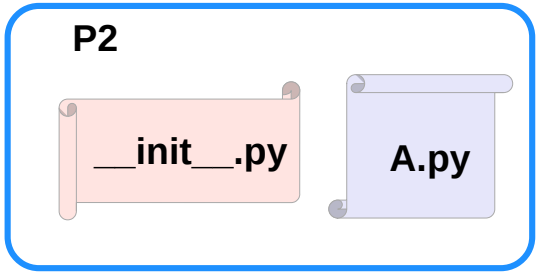
two different packages like **P1** and **P2**
can both have modules with the same name,
let's say **A**, for example.

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.





https://python-course.eu/python-tutorial/packages.php

Creating a package

to create a package, we need a directory.

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

assume we want to create "simple_package" package

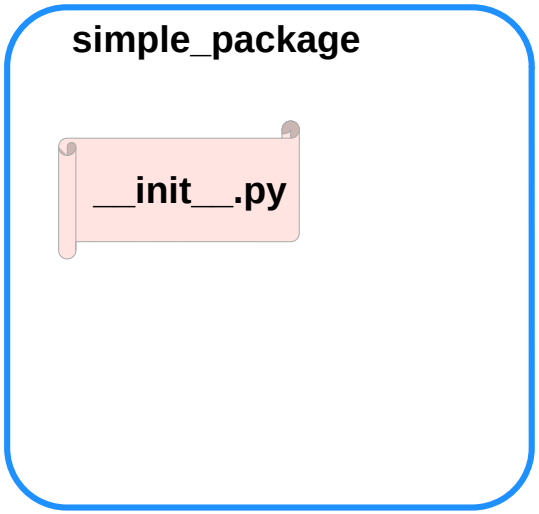
must create directory "simple_package"
and this directory needs to contain the "**__init__.py**" file

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.



https://python-course.eu/python-tutorial/packages.php

Examples of creating a package (1)

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: ➡ submodule **a**

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

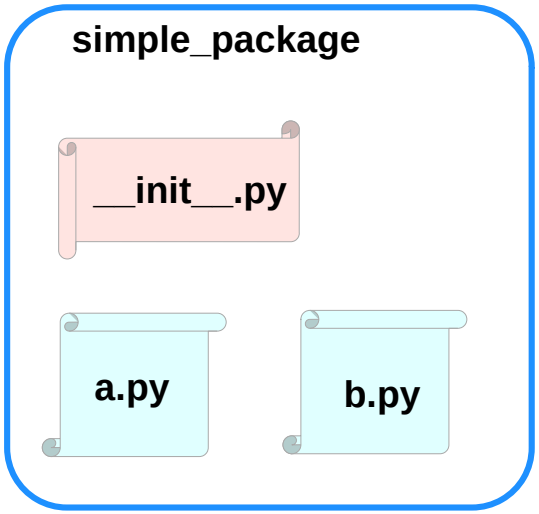
b.py: ➡ submodule **b**

```
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
```



https://python-course.eu/python-tutorial/packages.php

Examples of creating a package (2)

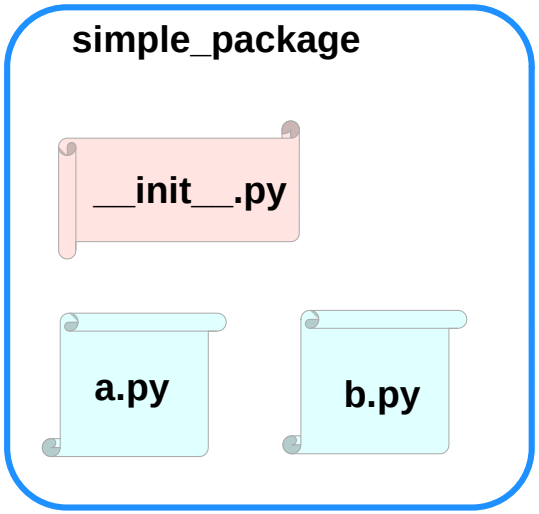
import **simple_package** from the interactive Python shell,

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):



https://python-course.eu/python-tutorial/packages.php

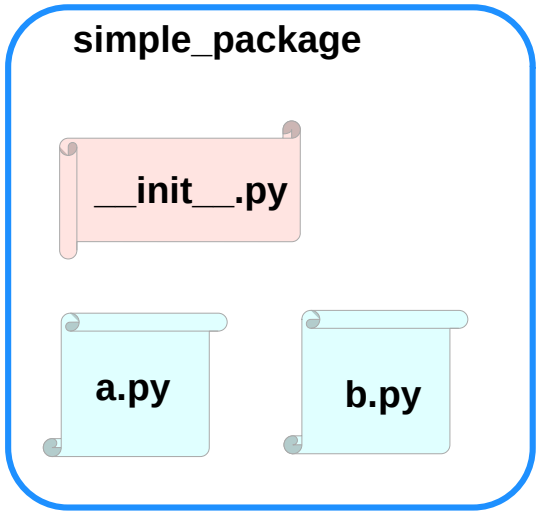
Examples of creating a package (3)

```
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
```



https://python-course.eu/python-tutorial/packages.php

Examples of creating a package (4)

the package **simple_package** has been loaded
but neither the module "**a**" nor the module "**b**" has been loaded

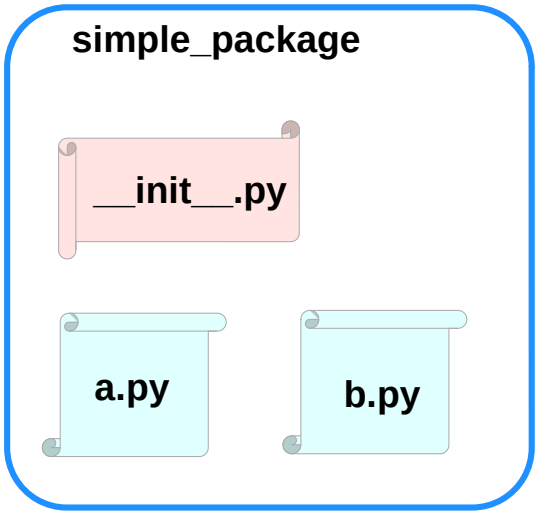
can't access neither "**a**" nor "**b**"
by solely importing **simple_package**.

must import 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
```



https://python-course.eu/python-tutorial/packages.php

Examples of creating a package (5)

to automatically load these modules.

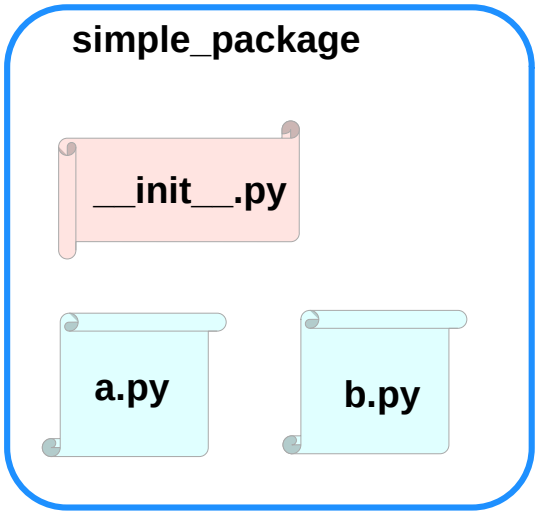
add the following lines to the file __init__.py:

```
import simple_package.a
import simple_package.b
```

Then

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

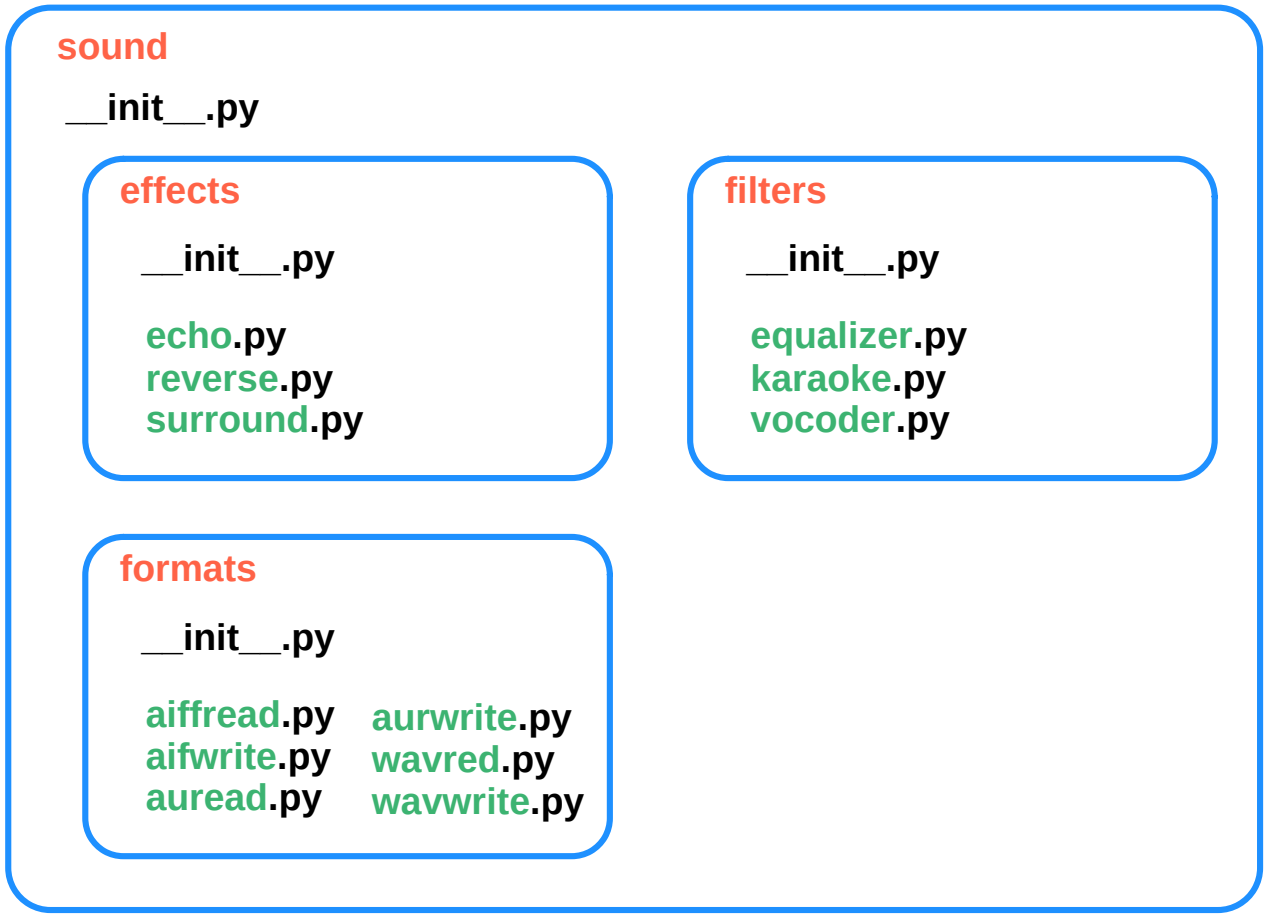
```
Hello, function 'bar' from module 'a' calling
Hello, function 'foo' from module 'b' calling
```



https://python-course.eu/python-tutorial/packages.php

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
```



https://python-course.eu/python-tutorial/packages.php

Package sound1 (1)

__init__.py
```
print("sound1 package is getting imported!")
```

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://python-course.eu/python-tutorial/packages.php

Package sound1 (2)

If we import the package **sound1**
by using the statement **import sound1**,
the package **sound1** but not the subpackages
effects, **filters** and **formats** will be imported

The reason for this consists in the fact
that the file **__init__.py** doesn't contain any code
for importing subpackages:

```
import sound1
print(sound1)
print(sound1.effects)
```

OUTPUT:

```
<module 'sound1' from '/data/Dropbox (Bodenseo)/
Bodenseo Team Folder/melisa/notebooks_en/
sound1/__init__.py'>
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-2-0b6d7fed3b24> in <module>
      3 print(sound1)
      4
----> 5 print(sound1.effects)
AttributeError: module 'sound1' has no attribute 'effects'
```

If you also want to use the package **effects**,
you have to import it explicitly with import **sound.effects**:

```
import sound1.effects
print(sound1.effects)
```

```
<module 'sound1.effects' from '/data/Dropbox (Bodenseo)/
Bodenseo Team Folder/melisa/notebooks_en/
sound1/effects/__init__.py'>
```

It is possible to have the submodule importing done automatically
when importing the sound1 module.

We will change now to **sound2** to demonstrate how to do this.

We use the same files as in **sound1**,
but we will add the code line **import sound2.effects**
into the file **__init__.py** of the directory **sound2**.

```
"""An empty sound package
This is the sound package, providing hardly anything!"""
import sound2.effects
print("sound2.effects package is getting imported!")
)
```

https://python-course.eu/python-tutorial/packages.php

Package sound2

__init__.py
```
import sound2.effects
print("sound2 package is getting imported!")
```

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!")
```

import sound2.effects
in *__init__.py* of the package sound2

when the package sound2,
the subpackage effects will also
be automatically loaded:

import sound2

```
sound2 package is getting imported!
```

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://python-course.eu/python-tutorial/packages.php

Package sound3

__init__.py
```
from . import effects
print("sound3 package is getting imported!")
```

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!")
```

Instead of using an absolute path we could have imported the effects-package relative to the sound2 package.

```
import sound2.effects   # absolute path

import effects          # relative path
```

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!")
```

import sound3

effects package is getting imported!

It is also possible to automatically import the package formats, when we are importing the effects package.

We can also do this with a relative path, which we will include into the __init__.py file of the directory effects:

```
from .. import formats
```

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://python-course.eu/python-tutorial/packages.php