模块

创建模块(my_module.py)

```
A = 100

def add(a, b):
    return a + b
```

导入模块 - import

```
❯ ipython
... 省略一些输出 ...
In [1]: import my_module

In [2]: my_module.A
Out[2]: 100

In [3]: my_module.add(1, 2)
Out[3]: 3

In [4]: my_module.B
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-7-715ed2da0652> in <module>()
----> 1 my_module.B

AttributeError: module 'my_module' has no attribute 'B'
```

导入模块 - from + import

```
In [5]: from my_module import A, add

In [6]: A
Out[6]: 100

In [7]: add(1, 2)
Out[7]: 3

In [8]: from my_module import *
```

不建议是用「from X import *」的理由

1. 不好跟踪
2. 导入的变量没有被用到，提倡按需导入
3. 命名空间污染

import 如何工作?

第一次导入模块要做三个步骤

1. 搜索并找到模块文件
2. 在必要时把模块文件编译成字节码
3. 执行模块的代码来创建所定义的对象

搜索路径顺序

1. 程序的主目录
2. PYTHONPATH系统变量
3. 标准库目录
4. .pth文件

sys.path

```
In : import sys

In : sys.path
Out:
['',
 '/Users/dongwm/Library/Python/2.7/bin',
 '/Users/dongwm/.venvburrito/lib/python2.7/site-packages',
 ...
 '/Users/dongwm/pipenv',
 '/Users/dongwm/.ipython']
```

搜索文件类型

1. a.py 代码源文件
2. a.pyc 字节码文件
3. 目录a 作为包导入
4. a.so/a.dll/a.pyd 编译扩展文件
5. 用c编译好的内置模块
6. zip文件包

my_module.py生成的字节码文件效果(Python 3)

```
In : !tree
.
├── __pycache__
│   └── my_module.cpython-36.pyc
└── my_module.py

1 directory, 2 files
```

__name__ 和 __main__

```
def run():
    print('Run')


run()
```

```
def run():
    print('Run')


if __name__ == '__main__':
    run()
```

```
❯ python test.py
Run

❯ ipython
In : import test
Run
```

```
❯ python test.py
Run

❯ ipython
In : import test
```

模块包

```
❯ tree
.
├── dir1
│   ├── __init__.py
│   ├── a.py
│   └── dir2
│       ├── __init__.py
│       └── b.py
```

```
In : import dir1

In : dir1
Out: <module 'dir1' from 'dir1/__init__.py'>

In : import dir1.dir2

In : from dir1 import a

In : from dir1.dir2 import b

In : dir1.a
Out: <module 'dir1.a' from 'dir1/a.py'>

In : dir1.dir2.b
Out: <module 'dir1.dir2.b' from 'dir1/dir2/b.py'>
```

延伸阅读

1. https://www.digitalocean.com/community/tutorials/how-to-write-modules-in-python-3
2. https://docs.python.org/3/tutorial/modules.html