

一、模块

python 模块是一个python文件，以.py结尾，包含了python对象定义和python语句。模块可以定义函数，类和变量，模块里也可能包含可执行的代码。

1.1 导入模块

- import 模块名
- from 模块名 import 功能名
- from 模块名 import *
- from 模块名 as 别名
- from 模块名 import 功能名 as 别名

1.1.1 import

```
import 模块名
Import 模块名1, 模块名2...

# 调用功能
模块名.功能名()
```

1.1.2 from import

- 语法

```
from 模块名 import 功能1, 功能2, 功能3
```

- 体验

```
from math import sqrt
print(sqrt(9)) #好处在于不需要写 模块名 直接写功能名
```

1.1.3 from import*

- 语法

```
from 模块名 import *
```

- 体验

```
from math import*  
print(sqrt(9))    #导入该模块的所有功能
```

1.1.4 as 定义别名

```
# 模块定义别名  
import 模块名 as 别名  
  
# 功能定义别名  
from 模块名 import 功能 as 别名
```

```
import time as tt  
  
tt.sleep(2)  
print('hello')  
  
from time import sleep as s1  
s1(2)  
print('hello')
```

1.2 制作模块

在python中 每一个python文件 都可以作为一个模块 模块的名字就是文件的名字，也就是说自定义模块名必须要符合标识符命名规则

1.2.1 定义模块

新建一个python文件， 命名为my_module.py 并定义testA函数

```
def testA(a,b):  
    print(a + b)
```

1.2.2 测试模块

```
def testA(a,b):  
    print(a + b)  
  
# name 是模块标识符 如果使用位置在本模块 即为__main__ 否则即为Python文件名  
# 那么在其他文件 导入该模块文件 name 就不是 main 所以testA就不会执行
```

```
if __name__ == '__main__':  
    testA(1,1)
```

```
import module1  
  
module1.testA(2,2)
```

1.3 模块定位顺序

当导入一个模块，python解释器对模块位置的搜索顺序是：

1. 当前目录
2. 如果不在当前目录，python则搜索在shell变量pythonpath下的每一个目录
3. 如果找不到python会查看默认路径。UNIX下，默认路径一般为/usr/local/lib/python

模块搜索路径存储在system模块的sys.path变量中。变量里包含当前目录，pythonpath和由安装过程决定的默认目录

注意：

- 自己的文件名不要和已有的模块名重复，否则导致模块功能无法使用
- 使用from模块名import功能的时候，如果功能名字重复，调用到的是最后定义或者导入的功能

1.4 __all__

如果一个模块文件中有__all__变量当使用from xxx import*导入时，只能导入这个列表中的元素

```
__all__ = ['testA']  
  
def testA():  
    print('testA')  
  
def testB():  
    print('testB')  
  
from test12 import*  
testA()  
testB() #报错
```