包导入：

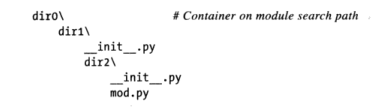
对于dir0\dir1\dir2\mod.py，可以在Python用如下方式导入：

import dir1.dir2.mod

但需要遵循下列要求：

1. dir0必须在Python搜索路径下（主目录、PYTHONPATH、.pth文件、标准链接库目录）
2. dir1和dir2下必须包含\_\_init\_\_.py文件。\_\_init\_\_.py可以完全为空。
3. dir0下不需要\_\_init\_\_.py文件，即使存在也会被忽略。

目录结构如下：

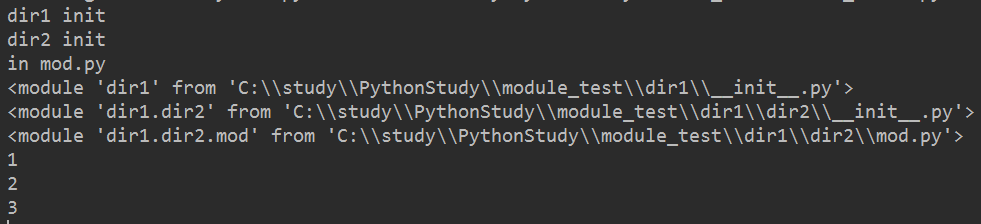


当首次执行包导入时，Python会自动执行\_\_init\_\_.py文件中的所有语句。

例：程序module\_test/module\_test5.py

import dir1.dir2.mod  
print(dir1)  
print(dir1.dir2)  
print(dir1.dir2.mod)  
print(dir1.x)  
print(dir1.dir2.y)  
print(dir1.dir2.mod.z)

输出为：



包导入的优点：

1. 包导入提供了程序文件的目录信息；
2. 包导入可以简化PYTHONPATH和.pth文件搜索路径设置。
3. 包导入让导入的文件更明确，从而解决同名文件的模糊性。

包相对导入：

例：程序module\_test/dir1/add.py

待相对导入的模块

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

程序module\_test/dir1/mod1.py

执行相对导入的py文件

# 该py文件不能直接运行  
# 相对导入只能发生在包内  
# 只能使用from语句  
from .add import add # 相对导入  
print("add()", add(1, 2))

程序module\_test/module\_test6.py

import dir1.mod1  
print("包相对导入", dir1.mod1.add(5, 6))

输出为：

dir1 init

add() 3

包相对导入 11

相对导入会先搜索导入语句所在包，然后再搜索sys.path路径。