

Python及其应用

主讲人：钱惠敏

E-mail: amandaqian@hhu.edu.cn

第8讲 模块

8.1 模块简介

8.2 模块的 `_name_`

8.3 `dir()`函数

8.4 包

8.1 模块简介

➤ 内置电池（batteries included）

基础代码库，覆盖了网络通信、文件处理、数据库接口、图形系统、XML处理

➤ 第三方工具包

- ✓ **Pandas**: 数据处理与分析
- ✓ **Numpy**: 科学计算
- ✓ **BeautifulSoup4**: 处理HTML文档
- ✓ **Matplotlib**: 绘图
- ✓ **Scikit-learn**: 基于SciPy和Numpy的开源机器学习模块
- ✓ **Tensorflow**: 深度学习框架

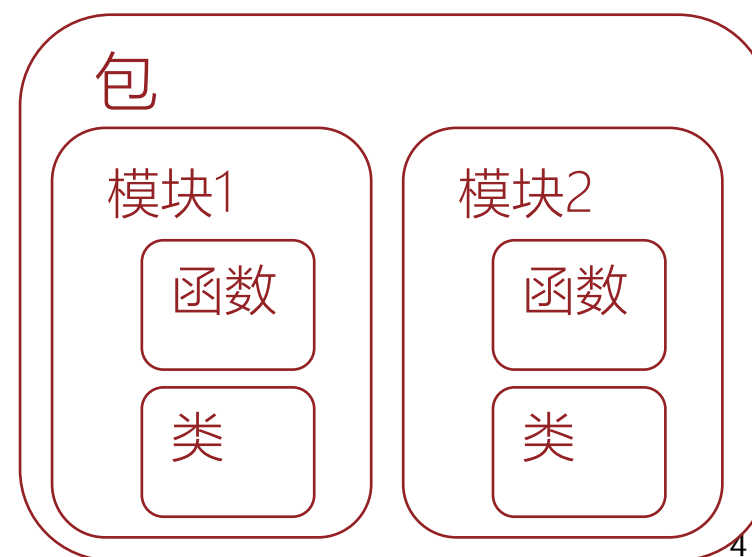
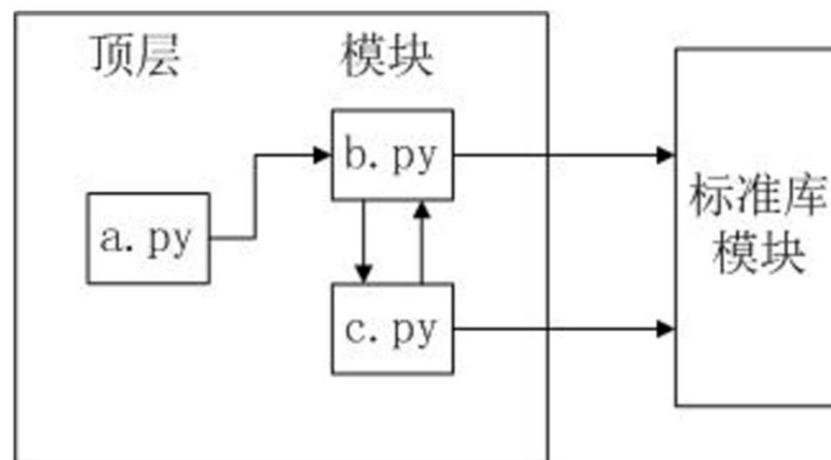
8.1 模块简介（续1）

➤Python程序的架构

将一个程序分割为源代码文件的集合以及将这些部分连接在一起的方法

➤Python源代码文件：*.py

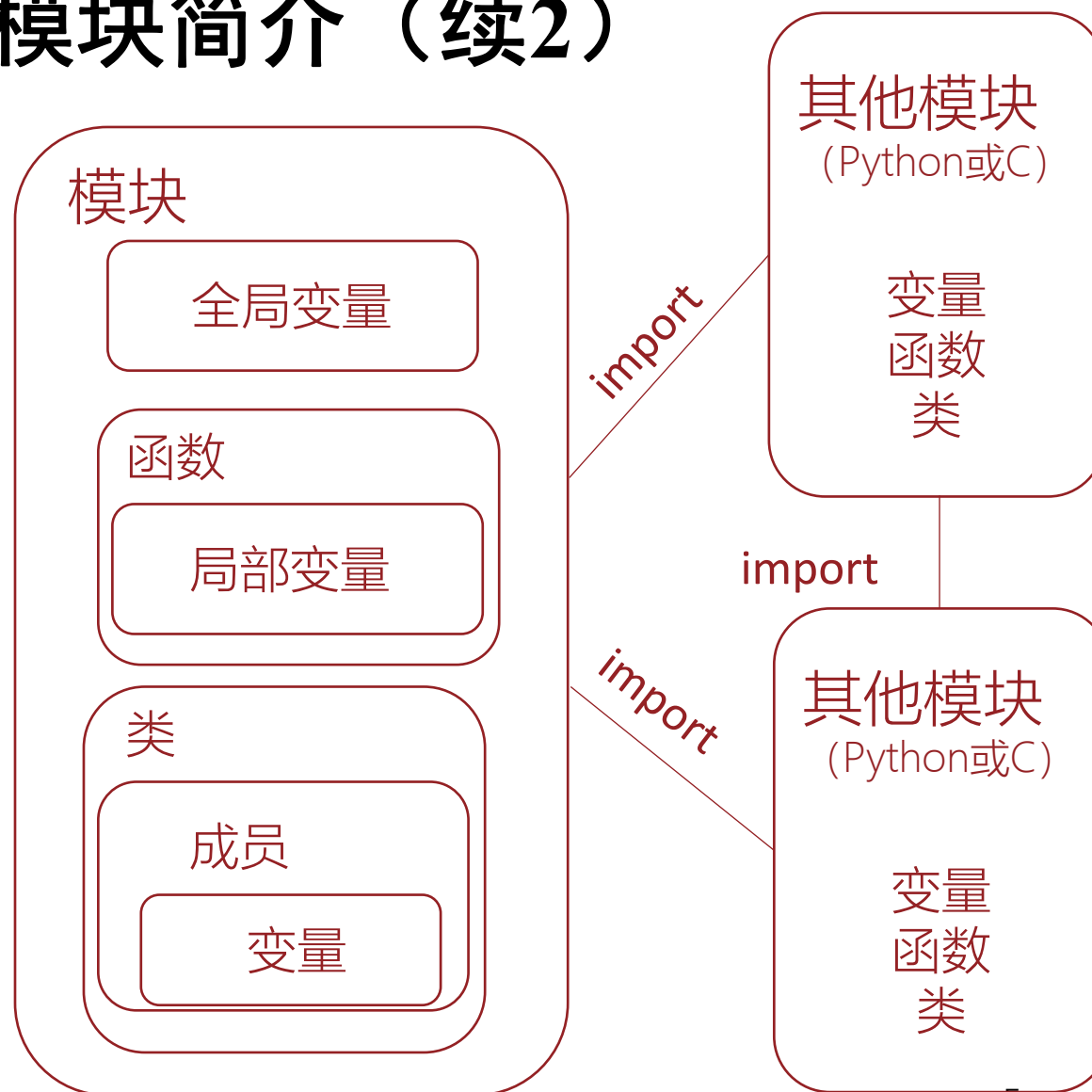
- ✓ 一个py文件是一个模块（module）
- ✓ 多个模块可以组成一个包（package）



8.1 模块简介（续2）

➤ 模块的执行环境

模块包含变量、函数、类以及其他的模块（如果导入的话），而函数也有自己的本地变量



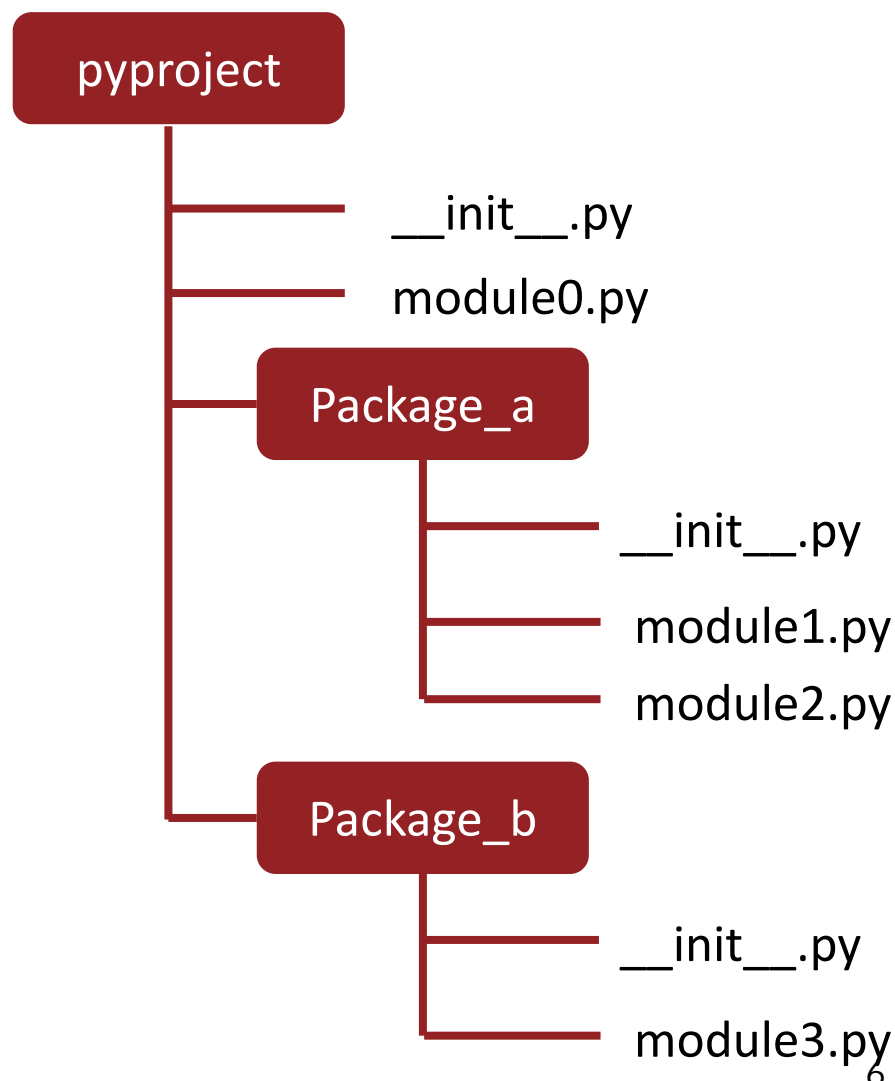
8.1 模块简介（续3）

➤ 模块的作用

模块是Python中的最高级别组织单元，它将程序代码和数据封装起来以便重用

➤ 模块的三个角色：

- ✓ 代码重用
- ✓ 系统命名空间的划分（模块可理解为变量名的封装，即模块就是命名空间）
- ✓ 实现共享服务和数据



8.1 模块简介（续4）

```
#support.py模块  
def print_func(par):  
    print ("Hello:",par)  
    return
```

```
import support  
print_func('Kelly')
```

➤ 模块定义存成.py文件

➤ 引入模块使用import语句

`import module1 [,module2 [...]]`

"import 模块名称 as 别名"

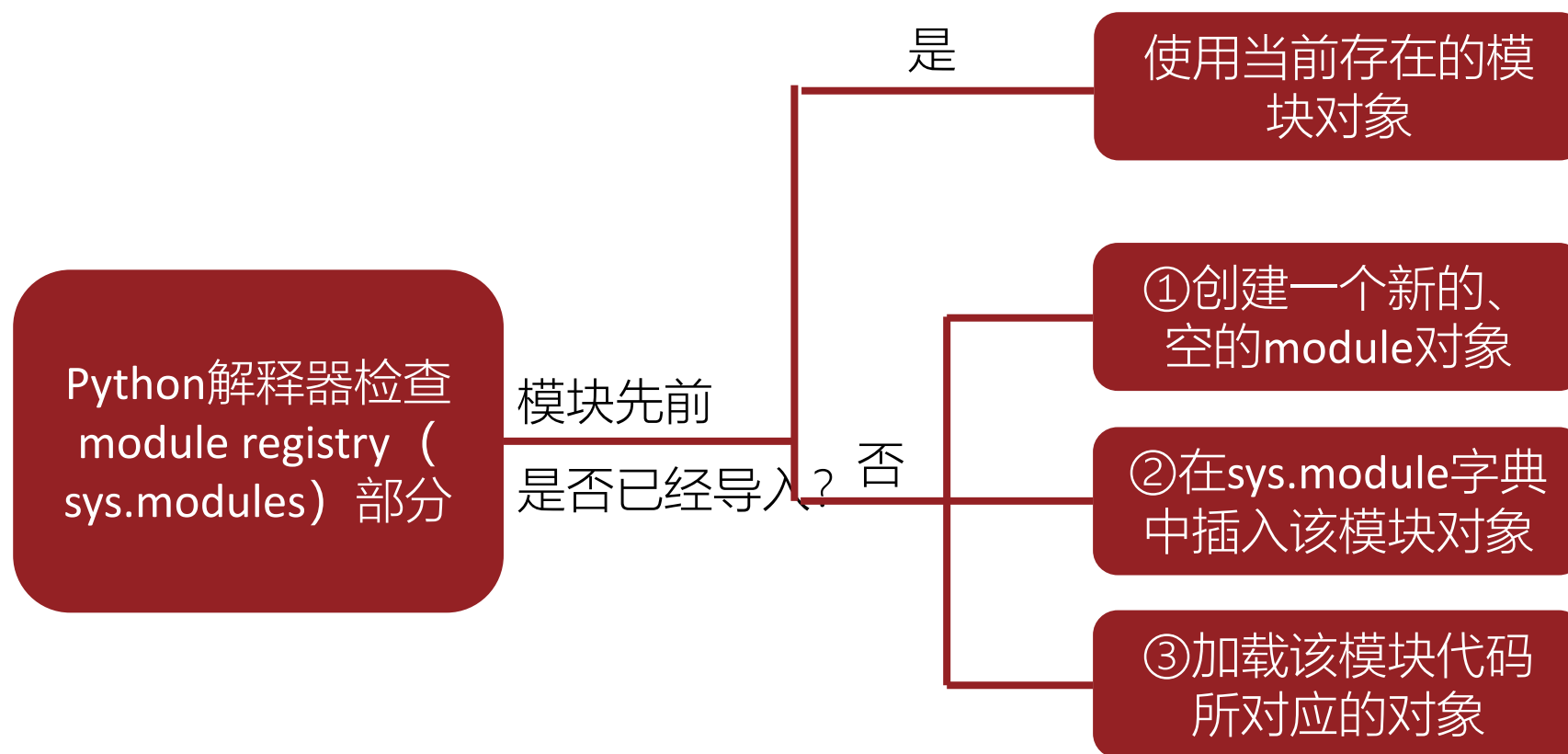
`import numpy as np`

“from 模块名称 import 函数名称”

`from numpy import sqrt`

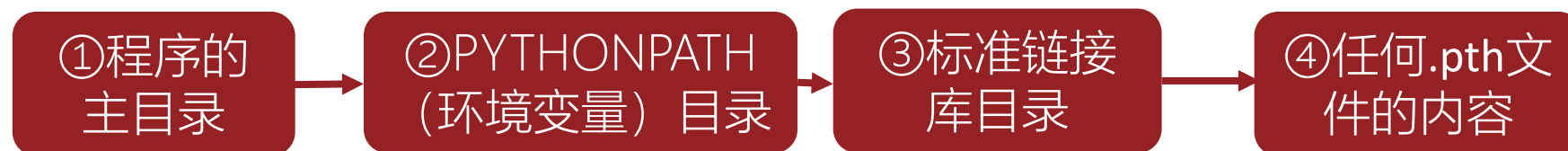
8.1 模块简介（续5）

➤ import模块时，Python都做了哪些事情？



8.1 模块简介（续6）

➤ import搜索的路径顺序



- ✓ 以上四个组件组合起来就变成了 `sys.path`，其保存了模块搜索路径在机器上的实际配置，可以通过打印内置的 `sys.path` 列表来查看这些路径
- ✓ 搜索路径①和③是系统自定义的，而②和④可以用于拓展路径，从而加入自己的源代码目录

8.2 模块的__name__

➤ 当一个模块被第一次引入的时候，这个模块的主块将被运行。假如只想在程序本身被使用的时候运行主块，而在它被别的模块引入的时候不运行主块，我们该怎么完成。

__name__属性完成。

每个Python模块都有它的__name__，如果它是'__main__'，这说明这个模块被用户单独运行，我们可以进行相应的恰当操作。

```
# Filename: using_name.py
def main():
    if __name__ == '__main__':
        print('This program is being run by itself.')
    else:
        print('I am being imported from another module')

main()
print(__name__)
```

8.3 dir() 函数

➤ 内置的函数 `dir()` 可以找到模块内定义的所有名称。以一个字符串列表的形式返回：

```
# Filename: mymodule.py
def sayhi():
    print ('Hi, this is mymodule speaking.')
```

```
import mymodule
dir(mymodule)
```

```
['__builtins__',
'__doc__',
'__file__',
'__name__',
'__package__',
'sayhi',
'version']
```

如果没有给定参数，那么 `dir()` 函数会罗列出当前定义的所有名称。

8.4 包（package）

- package包是一组module的集合，一个文件夹下面只要有个__init__.py 文件，这个文件夹就可以看作是一个包。
- 包的创建方法：
 - ✓ 先在当前目录创建一个目录testpackage
 - ✓ 在testpackage下创建一个空文件__init__.py
 - ✓ 在testpackage中创建一个testmodule.py，里面编写任意代码
 - ✓ 启动Python，运行下列代码

```
>>> import testpackage.testmodule  
>>> testpackage.testmodule.sayhi()  
Hi, this is mymodule speaking.
```

8.4 包 (package)

- 包是一种组织模块的方法，提供了一个命名空间，防止发生名字冲突。
- 包中还可以有包，所以这种方式可以很好的组织一个树状结构，用来管理多个模块。

