Python程序设计：模块、包、库与框架

2023-2024

王斌辉 副教授

南开大学软件学院

模块 module

- 为了使代码更容易维护，提高代码重用价值，可以将一组相关功能的代码（可能包含多个函数与类）写入一个单独的.py文件中，以便在其它场景导入使用，这样的.py文件称为模块
- 模块的导入语法：
  - **import** module1[, module2[,... moduleN]]
  - **from** modname **import** name1[, name2[, ... nameN]]
  - **from**...**import** *

此类声明不应被过多使用

模块 module

- 搜索路径：
  - 当前目录
  - 如果都找不到，Python会察看默认路径
  - 如不在当前目录， 则搜索在 shell 变量 PYTHONPATH 下的每个目录
  - 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：
    - Windows 系统：set PYTHONPATH=c:\python27\lib;
    - UNIX 系统：set PYTHONPATH=/usr/local/lib/python

模块 module

- 搜索路径：
  - __name__
    - if __name__ == "__main__":

      检测来保护的代码块仅会在模块被用来填充 __main__ 命名空间时而非普通的导入时被执行
  - __loader__
  - __package__
  - __spec__
  - __path__：不是包的模块不应该具有该属性
  - __file__
  - __cached__

包 package

- 包是一个有层次的文件目录结构，它定义了由n个模块或n个子包

```
package/
        __init__.py
        subpackage1/
                __init__.py
                moduleX.py
                moduleY.py
        subpackage2/
                __init__.py
                moduleZ.py
        moduleA.py
```

__init__.py中可以设置外界通过包名所能访问的模块

- 当外界使用import ...语句导入包后，就可以通过.访问我们指定对外提供的模块了
- 当外界使用from ... import ...直接导入包内的模块时，不论导入的模块是否在__init__.py中被指定了，都可以正常被导入

- 相对导入使用前缀点号，一个前缀点号表示相对导入从当前包开始。 两个或更多前缀点号表示对当前包的上级包的相对导入，第一个点号之后的每个点号代表一级

库library

- 参照其它编程语言的一个称呼，完成一定功能的代码集合，具体表现可以是一个模块，也可以是包

框架framework

- 一个架构层面的概念:
- 从库功能的角度来看：解决一个开放性问题而设计的具有一定约束性的支撑结构
- 通过一个框架，可以快速实现一个问题解决的骨架，到时按照框架角色去填充，交互就可以完成一个质量好，维护性高的项目
  - 如Web框架：Flask、Django...