# 模块与包

## 课程目标

* 能够使用 import 导入模块中所有的成员
* 能够使用 from…import… 导入模块中的部分成员

## 课程内容

* 模块
* 包

### 模块

#### 1.1模块概念

模块指包含了python程序的源文件，每一个独立的.py文件具有特定的功能，其功能可以被其他python程序使用，该文件就构成了一个模块

模块的名称必须满足**标识符**命名规则

#### 1.2模块中的资源

模块中以下资源可以被外界使用：

类（定义）

函数

变量（全局）

#### 1.3模块资源导入

格式一（全部导入）：

导入格式：

import 模块名

模块资源使用格式：

模块名.类名

模块名.函数名

模块名.变量名

格式二（局部导入）：

导入格式：

from 模块名称 import 资源名称

from 模块名称 import 资源名称1，资源名称2

from 模块名称 import \*

模块资源使用格式：

类名

函数名

变量名

注意：

如果多次导入模块中具有同名的资源，后导入的模块资源覆盖先导入的模块资源

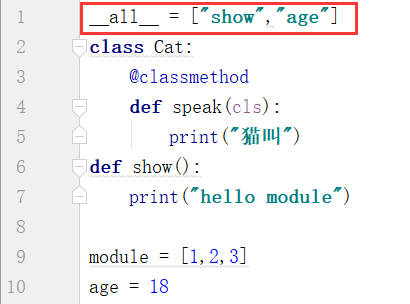
#### 1.4局部导入资源访问控制

如果使用格式二中的全部资源导入方式

from 模块名称 import \*

可以在模块文件中第一行设置资源访问控制

\_\_all\_\_ = ["资源名称1","资源名称2"]



说明：\_\_all\_\_列表中出现名称的资源可以被其他文件中使用

#### 1.5模块文件的执行代码处理

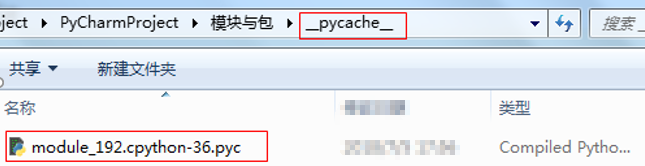
导入模块的同时会将原模块对应的.py文件进行执行，包括书写的运行代码，通过下列方式屏蔽原始模块中的运行代码：

if \_\_name\_\_ == "\_\_main\_\_":

执行代码

#### 1.6模块资源运行工作原理

模块资源导入时，解释器第一次读取原始资源文件，并生成缓存文件。 再次使用时直接使用缓存文件，加速程序执行。



### 包

#### 2.1包的概念

包就是项目结构中的文件夹/目录

#### 2.2包的作用

包用于将不同的模块进行分类管理，包本身不具有具体的含义， 仅用于文件分层管理

注意事项：放入包中的模块使用时需要在模块名前面添加包名，中间使用.分隔