

[笔记][LIKE-Python-6][01]

Python

[笔记][LIKE-Python-6][01]

002. 包和模块-基本概念

003. 包和模块-作用

004. 包和模块-分类

005. 包和模块-创建

006. 包和模块-基本信息

002. 包和模块-基本概念

概念解释

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写入一个单独的.py文件中；	
模块	• 供别人导入使用；这个.py文件就被称作是一个模块
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	
包	• 包含__init__.py文件的目录，这个目录下一定得有这个__init__.py
具体表现形式 • 文件和其它模块或子包	
参照其他编程语言的一个喊法	
库	• 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	
一个架构层面的概念；	
框架	• 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	
例如：Web框架 • Django	
Flask	

003. 包和模块-作用

1. 编写好一些“轮子”，“工具”代码，供其他模块调用 • 有效的对程序进行分解；方便代码的管理和维护

2. 可以防止同一模块内的命名重复问题

004. 包和模块-分类

按照包和模块的提供方可以分为三类：

- 标准包/模块
- 三方包/模块
- 自定义包/模块

标准包/模块

安装Python后, 自动帮我们安装好的一些模块; 可以直接导入使用

每个安装Python的用户电脑上都有这些模块 • 在你电脑上写的使用某个系统模块的代码, 到其他人电脑上, 同样能够运行满足我们基本的功能需求

特殊 • 内建包/模块 • 当我们在编码的过程当中, python自动帮我们导入的模块; • builtins

我们使用这些功能的时候, 不需要手动的导入该模块, 直接使用就可以

内置模块：因为经常使用，所以 Python 自动帮我们导入

- 对于 Python 2
`import __builtin__` 内置模块不需要手动导入
`print(dir(__builtin__))` 查看模块有哪些东西
- 对于 Python 3
`import builtins`

三方包/模块

有些功能，系统模块没有实现，或者实现了，但我们使用起来比较复杂

就有开发人员，开发了一些使用更加方便的模块；供其他开发人员使用 • 有可能也使用到了系统的某些模块

有可能没有使用

如果你的代码使用了某个三方模块；到时候，把代码拷贝给别人后，其他开发人员，需要下载安装才可以 • 别人也需要先安装同样的三方模块后才能运行

自定义包/模块

我们自己写的一些好用的模块

发布出去给别人用, 就变成了三方包/模块

005. 包和模块-创建

创建模块

直接创建一个.py文件即可

创建包

创建一个文件夹, 文件夹内务必创建一个 `__init__.py` 这个文件

其实3.3版本往后可以不用创建

但是为了代码版本兼容, 以及做一些其他包处理操作, 目前还是建议创建

`__init__.py` 文件作用

第一次导入包之后都会执行 `__init__.py`

创建多层级包

在包里面直接创建另外一个包即可, 可以无限级嵌套

006. 包和模块-基本信息

模块/包的名称

模块的名称没有 `.py`

包的名称就是文件夹的名称

查看标准包/模块的位置

- `import os` 此时 `PyCharm` 会智能提示
- `print(os.__file__)`

2 与 3 的不兼容说的就是两者之间包和模块发生了一些变化。

查看源代码和 `API`

- `Ctrl + 单击` 跳转查看源码实现
- `print(dir(模块名/包名))` 查看 `API`

完成于 201811100943