

中国大学 MOOC 课程

# 《Python 语言程序设计》

课后练习（第 7 周）



北京理工大学

Python 语言教学团队

## 【说明】

本文是中国大学 MOOC 课程《Python 语言程序设计》第 7 周的课后学习内容，预估学习完成时间约 40 分钟。

本周课后练习内容包括 1 道编程题和 1 段学习内容。

对于尚未安装 Python 运行环境的同学，请根据第 1 周课程内容介绍的步骤安装 Python 3.5.1 或者 Python 3.5.2 版本解释器，如果操作系统兼容性有问题，可以安装 Python 3.4 版本解释器。

## 【学习资料】

### 1. pyinstaller 库的使用

PyInstaller 是一个十分有用的第三方库,它能够在 Windows、Linux、Mac OS X 等操作系统下将 Python 源文件打包,通过对源文件打包,Python 程序可以在没有安装 Python 的环境中运行,也可以作为一个独立文件方便传递和管理。PyInstaller 需要在命令行(控制台)下用 pip 工具安装,如下:

```
:>pip install pyinstaller
```

或

```
:>pip3 install pyinstaller
```

PyInstaller 的官方网站网址为 <http://www.pyinstaller.org/>。

PyInstaller 库会自动将 PyInstaller 命令安装到 Python 解释器目录中,与 pip 或 pip3 命令路径相同,因此可以直接使用。使用 PyInstaller 库十分简单,

请注意,由于 PyInstaller 不支持源文件命中有英文句号 (.) 存在,假设 dpython.py 文件在 D:\codes 目录中,命令如下:

```
:>pyinstaller dpython.py
```

或

```
:>pyinstaller D:\codes\dpython.py
```

执行完毕后，源文件所在目录将生成 dist 和 build 两个文件夹。其中，build 目录是 pyinstaller 存储临时文件的目录，可以安全删除。最终的打包程序在 dist 内部的 dpython 目录中。目录中其他文件是可执行文件 dpython.exe 的动态链接库。

可以通过-F 参数对 Python 源文件生成一个独立的可执行文件，如下：

```
: \>pyinstaller -F dpython.py
```

执行后在 dist 目录中出现了 dpython.exe 文件，没有任何依赖库，执行它即可。

使用 PyInstaller 库需要注意以下问题：

- 文件路径中不能出现空格和英文句号 (.)；
- 源文件必须是 UTF-8 编码，暂不支持其他编码类型。采用 IDLE 编写的源文件都保存为 UTF-8 编码形式，可直接使用。

PyInstaller 有一些常用参数，如表 7.1 所示。

表 7.1: pyinstaller 命令的常用参数

参数	功能
-h, --help	查看帮助
-v, --version	查看 PyInstaller 版本
--clean	清理打包过程中的临时文件
-D, --onedir	默认值，生成 dist 目录
-F, --onefile	在 dist 文件夹中只生成独立的打包文件

<code>-p DIR, --paths DIR</code>	添加 Python 文件使用的第三方库路径
<code>-i &lt;.ico or .exe,ID or .icns&gt;, --icon &lt;.ico or .exe,ID or .icns &gt;</code>	指定打包程序使用的图标（icon）文件

PyInstaller 命令不需要在 Python 源文件中增加代码，只需要通过命令行进行打包即可。`-F` 参数最为常用，对于包含第三方库的源文件，可以使用`-p` 添加第三方库所在路径。如果第三方库由 pip 安装且在 Python 环境目录中，则不需要使用`-p` 参数。

## 【课后练习】

### 1. pyinstaller 库的实例

请使用 pyinstaller 库将第 6 周课后练习中“哈姆雷特词频统计”程序打包成单独文件。

（上述内容仅供个人学习使用，禁止转载）

面向过程和面向对象有何区别？