

# Package Guide

杨资璋（翻译）

2021 年 10 月 10 日

## 1 打包和分发项目

这个章节包含如何设置、打包和分发你自己的 Python 项目的基本知识。它假设你已经熟悉了安装包的内容。

这个章节并不以包含所有 Python 项目开发的最佳实践作为目标。例如，它不为版本控制、文档或者测试提供指南或者工具推荐。

对于更多的参考资料，查看 `setuptools` 文档中的构建和分发包章节，但是注意其中的一些建议内容可能已经过时了。如果发生冲突，优先考虑 Python 打包用户指南的建议。

### 1.1 打包和分发的要求

- 首先，确保你已经实现了安装包的要求
- 安装 “twine”： `python3 -m pip install twine`；你需要使用它来将你的包分发上传至 PyPI

### 1.2 配置你的项目

#### 1.2.1 初始文件

**setup.py**。存在于你的项目根目录下的 `setup.py` 是最重要的文件。例如，见 PyPA 样本项目中的 `setup.py`。

`setup.py` 有两个主要功能：

- 在这个文件中，你的项目的多个方面被设置。`setup.py` 的主要特性是它包含一个全局 `setup()` 函数。传入这个函数的关键字参数就是项目的特定细节是如何被定义的。下面的章节解释了最相关的参数。
- 它是运行多个与打包任务相关的命令的命令行接口。运行 `python setup.py -help-command` 可以得到可用命令的列表。

**setup.cfg**。它是包含 `setup.py` 命令的默认选项的初始文件。例如，见 PyPA 样本项目中的 `setup.cfg`。

**README.rst/README.md**。所有的项目都应该包含一个涵盖项目目标的 `readme` 文件。最常见的格式是以“`rst`”为扩展名的 `reStructuredText`，虽然这并不是一个必须要求；`Markdown` 的多种变种也是被支持的（见 `setup()` 的 `long_description_content_type` 参数）。

例如，见 PyPA 样本项目中的 `README.md`。

**MANIFEST.in**。当你需要打包不会自动包含在源分发的额外文件时，你需要一个 `MANIFEST.in` 文件。对于写一个 `MANIFEST.in` 文件的细节，包括默认包括什么的列表，见通过 `MANIFEST.in` 在源分发中包含文件。

例如，见 PyPA 样本项目中的 `MANIFEST.md`。

**LICENSE.txt**。每个包都应该包含一个详述分发用法的许可文件。在很多司法管辖区，没有显示许可的包不能被合法使用或者分发给除了版权所有者的其他人。如果你不确定选择哪种许可，你可以使用例如 `GitHub` 的许可选择的资源或者询问你的律师。

例如，见 PyPA 样本项目中的 `LICENSE.txt`。

**<your package>**。虽然这并不是要求的，但是将你的 `Python` 模块和包放在和你的项目名字相同或者非常接近的单独顶层包中是最普遍的做法。

例如，见 PyPA 样本项目中的 `sample` 包。

### 1.2.2 setup() 参数

## 1.3 在“开发模式”下开发

## 1.4 打包你的项目

## 1.5 将你的项目上传至 PyPI