

# Python打包分发工具setuptools - pip+wheel 安装、详解及示例

已于 2022-03-15 09:13:59 修改

版权声明：本文为博主原创文章，遵循 [CC 4.0 BY-SA](#) 版权协议，转载请附上原文出处链接和本声明。

本文链接：[https://blog.csdn.net/weixin\\_44462773/article/details/123417045](https://blog.csdn.net/weixin_44462773/article/details/123417045)

## Python打包分发工具setuptools

### 一、介绍

### 二、打包步骤

#### 2.1 前提条件

##### 2.1.1 python环境准备

##### 2.1.2 pip安装

##### 2.1.3 wheel安装

#### 2.2 编写setup.py文件

#### 2.3 分发

#### 2.4 安装 Wheel

#### 2.5 使用

## 一、介绍

Python项目的打包、发布和部署的常用方法比较：

1. distutils - python自带的基本安装工具, 适用于非常简单的应用场景使用, 不支持依赖包的安装  
通过distutils来打包, 生成安装包, 安装python包等工作, 需要编写名为setup.py python脚本文件。
2. setuptools - 针对 distutils 做了大量扩展, 尤其是加入了包依赖机制。不支持python3, 安装完setuptools后会有easy\_install
3. distribute - 类似于setuptools, 支持python3, 安装完distribute后会有easy\_install。
4. easy\_install - setuptools 和 distribute 自带的安装脚本, 也就是一旦setuptools或distribute安装完毕, easy\_install 也便可用了。
5. pip - 目标是取代easy\_install。easy\_install 有很多不足: 安装事务是非原子操作, 只支持 svn, 没有提供卸载命令, 安装一系列包时需要写脚本; pip 解决了以上问题, 已然成为新的事实标准, virtualenv 与它已经成为一对好搭档;
6. distutils2 - setuptools 和 distribute 的诞生是因为 distutils 的不济, 进而导致目前分化的状况。它将成为 Python 3.3 的标准库 packaging, 并在其它版本中以distutils2 的身份出现; 换句话说, 它和 pip 将联手结束目前混乱的状况。
7. virtualenv - 用来创建隔离的python环境, 处理python环境的多版本和模块依赖。

本文主要介绍了基于setuptools 通过pip+wheel的方式对python项目进行打包。

## 二、打包步骤

### 2.1 前提条件

#### 2.1.1 python环境准备

1、首先要确定你机器上已经安装了python环境，可通过以下命令查看本机的python版本：

win+R 进入cmd命令行 -> 进入python安装目录 -> 输入命令 python 查看python 版本

```
管理员: C:\Windows\system32\cmd.exe - python
Microsoft Windows [版本 10.0.17763.2686]
(c) 2018 Microsoft Corporation。保留所有权利。

C:\Users\Administrator>f:

F:\>cd F:\python\python-3.8.10

F:\python\python-3.8.10>python
Python 3.8.10 (tags/v3.8.10:3d8993a, May 3 2021, 11:48:03) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

## 2.1.2 pip安装

注：如果有出现“pip’不是内部或外部命令，也不是可运行的程序或批处理文件’异常时，  
处理方案：‘pip’不是内部或外部命令，也不是可运行的程序或批处理文件 – 解决方案。

1、pip是python的一个模块，在使用之前，要确认一下该模块是否存在：

输入命令：`python -m ensurepip`，

```
管理员: C:\Windows\system32\cmd.exe
Microsoft Windows [版本 10.0.17763.2686]
(c) 2018 Microsoft Corporation。保留所有权利。

C:\Users\Administrator>f:

F:\>cd F:\python\python-3.8.10

F:\python\python-3.8.10>python -m ensurepip
WARNING: Ignoring invalid distribution -ip (f:\python\python-3.8.10\lib\site-packages)
WARNING: Ignoring invalid distribution -ip (f:\python\python-3.8.10\lib\site-packages)
Looking in links: c:\Users\Administrator\AppData\Local\Temp\tmpfp_dkt0p
Requirement already satisfied: setuptools in f:\python\python-3.8.10\lib\site-packages (56.0.0)
Processing c:\users\administrator\appdata\local\temp\tmpfp_dkt0p\pip-21.1.1-py3-none-any.whl
WARNING: Ignoring invalid distribution -ip (f:\python\python-3.8.10\lib\site-packages)
Installing collected packages: pip
WARNING: Ignoring invalid distribution -ip (f:\python\python-3.8.10\lib\site-packages)
Successfully installed pip-21.1.1
```

2、将pip升级：

输入命令：`python -m pip install --upgrade pip`

```
F:\python\python-3.8.10>python -m pip install --upgrade pip
WARNING: Ignoring invalid distribution -ip (f:\python\python-3.8.10\lib\site-packages)
WARNING: Ignoring invalid distribution -ip (f:\python\python-3.8.10\lib\site-packages)
Requirement already satisfied: pip in f:\python\python-3.8.10\lib\site-packages (21.1.1)
Collecting pip
  Using cached pip-22.0.4-py3-none-any.whl (2.1 MB)
WARNING: Ignoring invalid distribution -ip (f:\python\python-3.8.10\lib\site-packages)
Installing collected packages: pip
  Attempting uninstall: pip
    WARNING: Ignoring invalid distribution -ip (f:\python\python-3.8.10\lib\site-packages)
    Found existing installation: pip 21.1.1
    Uninstalling pip-21.1.1:
      Successfully uninstalled pip-21.1.1
WARNING: Ignoring invalid distribution -ip (f:\python\python-3.8.10\lib\site-packages)
Successfully installed pip-22.0.4
WARNING: Ignoring invalid distribution -ip (f:\python\python-3.8.10\lib\site-packages)
WARNING: Ignoring invalid distribution -ip (f:\python\python-3.8.10\lib\site-packages)
WARNING: Ignoring invalid distribution -ip (f:\python\python-3.8.10\lib\site-packages)
```

### 2.1.3 wheel安装

Wheel 也是一种 built 包，而且是官方推荐的打包方式。也许你曾经遇见或使用过 egg 包，但现在 wheel 是官方推荐的打包方式（<https://wheel.readthedocs.io/en/stable/>）。

wheel 包的优点：

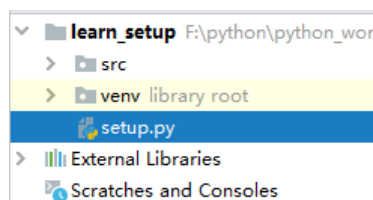
- 1.更快地安装纯 Python 和本机 C 扩展包。
- 2.避免安装时执行任意代码。（避免 setup.py）
- 3.在 Windows 或 macOS 上安装 C 扩展不需要编译器。
- 4.为测试和持续集成提供更好的缓存。
- 5.创建 .pyc 文件作为安装的一部分，以确保它们与使用的 Python 解释器匹配。
- 6.跨平台和机器的更一致的安装。

使用 wheel 打包，首先要安装 wheel，输入命令：`pip3 install wheel`

```
F:\python\python-3.8.10>pip3 install wheel
WARNING: Ignoring invalid distribution -ip (f:\python\python-3.8.10\lib\site-packages)
WARNING: Ignoring invalid distribution -ip (f:\python\python-3.8.10\lib\site-packages)
Collecting wheel
  Downloading wheel-0.37.1-py2.py3-none-any.whl (35 kB)
WARNING: Ignoring invalid distribution -ip (f:\python\python-3.8.10\lib\site-packages)
Installing collected packages: wheel
WARNING: Ignoring invalid distribution -ip (f:\python\python-3.8.10\lib\site-packages)
Successfully installed wheel-0.37.1
WARNING: Ignoring invalid distribution -ip (f:\python\python-3.8.10\lib\site-packages)
WARNING: Ignoring invalid distribution -ip (f:\python\python-3.8.10\lib\site-packages)
```

## 2.2 编写setup.py文件

在项目与src平级的目录下新增文件setup.py:



编辑setup.py文件内容：

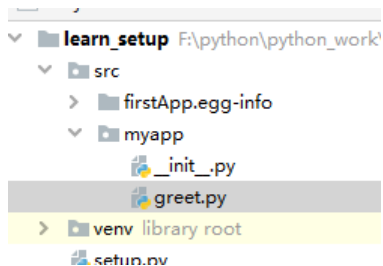
```
1 from setuptools import setup, find_packages
2 setup(
3     name='firstApp', # 应用名
4     version='1.0.0', # 版本号
5     packages=find_packages('src'), # 包含所有src中的包
6     package_dir={'': 'src'}, # 告诉distutils包都在src下
7
8     # 打包时需要打包的数据文件，如图片，配置文件等
9     package_data = {
10         # 任何包中含有.txt文件，都包含它
11         '': ['*.xml'],
12     },
13     # 项目依赖的 Python 库，使用 pip 安装本项目时会自动检查和安装依赖
14     install_requires=['numpy'],
15     # 指定项目依赖的 Python 版本。
16     python_requires='>=3',
17     # 是否使用静态文件，为true时静态文件生效，否则不起作用
18     include_package_data=True,
19     # 此项需要，否则卸载时报windows error
20     zip_safe=False,
21     setup_cfg=True
22     #其他信息暂时没加，后面会加附录进行说明
23 )
```

文件格式内容说明，参考链接: [Python打包分发工具setuptools](#)。

另外，为了测试，我们在src下面加个python包，目录结构及内容如下：

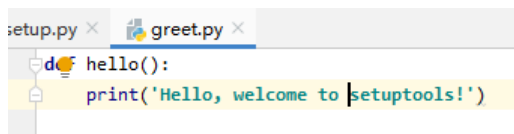
新增包结构：myapp

新增文件：init.py 和greet.py



greet.py编辑内容为：

```
def hello(): print('Hello, welcome to setuptools!')
```



## 2.3 分发

有了上面的 setup.py 文件，我们就可以打出各种安装包，主要分为两类：sdist 和 bdist。

使用 bdist 可以打出 built distribution，和源码包相比，由于预先构建好，所以安装更快：

Format	Description	Notes
gztar	gzipped tar file (.tar.gz)	Unix 默认
bztar	bzipped tar file (.tar.bz2)	
xztar	xzipped tar file (.tar.xz)	
ztar	compressed tar file (.tar.Z)	
tar	tar file (.tar)	
zip	zip file (.zip)	Windows 默认
rpm	RPM	
pkgtool	Solaris pkgtool	
sdux	HP-UX swinstall	
wininst	self-extracting ZIP file for Windows	
msi	Microsoft Installer.	CSDN @黎陌MLing

使用上，和 sdist 一样，可以使用 --formats 指定包格式。如：`python setup.py bdist --formats=rpm`  
同时为了简化操作，setuptools 提供了如下命令：

Command	Formats	Notes
bdist_dumb	tar, gztar, bztar, xztar, ztar, zip	Windows 默认 zip, Unix 默认 gztar
bdist_rpm	rpm, srpm	
bdist_wininst	wininst	
bdist_msi	msi	

所以打 rpm 包可以使用：`python setup.py bdist_rpm`



使用wheel打包，命令行进入待打包的工程目录，  
输入命令：`python setup.py sdist bdist_wheel`

```
F:\python\python_work\learn_setup>python setup.py sdist bdist_wheel
F:\python\python_work\learn_setup>src\data1
F:\python\python-3.8.10\lib\distutils\dist.py:274: UserWarning: Unknown distribution option: 'setup_cfg'
  warnings.warn(msg)
running sdist
running egg_info
writing src\firstApp.egg-info\PKG-INFO
writing dependency_links to src\firstApp.egg-info\dependency_links.txt
writing requirements to src\firstApp.egg-info\requires.txt
writing top-level names to src\firstApp.egg-info\top_level.txt
reading manifest file 'src\firstApp.egg-info\SOURCES.txt'
reading manifest template 'MANIFEST.in'
warning: no files found matching '*.xml'
warning: no files found matching 'LICENSE.txt'
writing manifest file 'src\firstApp.egg-info\SOURCES.txt'
```

CSDN @黎陌MLing

成功后 /dist目录 存在两个文件:

本地磁盘 (F:) > python > python\_work > learn\_setup > dist

名称	修改日期	类型	大小
 firstApp-1.0.0.tar	2022/3/14 17:05	WinRAR 压缩文件	5 KB
 firstApp-1.0.0-py3-none-any.whl	2022/3/14 17:05	WHL 文件	5 KB

执行成功后，目录下除了 dist 和 \*.egg-info 目录外，还有一个 build 目录用于存储打包中间数据。wheel 包的名称如 **firstApp-1.0.0-py3-none-any.whl**，其中 py3 指明只支持 Python3。

可以使用参数 **--universal**，包名如 **firstApp-1.0.0-1.0.0-py2.py3-none-any.whl**，表明 wheel 包同时支持 Python2 和 Python3。使用 universal 也成为通用 wheel 包，反之称为纯 wheel 包。

注：应用开发过程中会频繁变更，每次安装都需要先卸载旧版本很麻烦。使用 develop 开发模式安装的话，实际代码不会拷贝到 site-packages 下，而是除一个指向当前应用的链接（\*.egg-link）。

这样当前位置的源码改动就会马上反映到 site-packages。使用如下：**python setup.py develop**

如需卸载，使用 **pip uninstall**

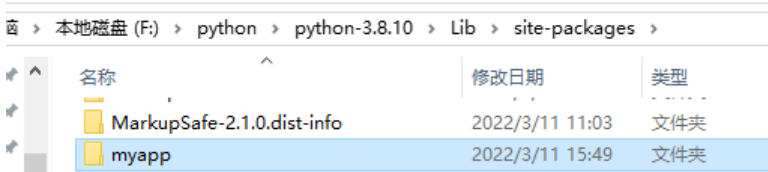
## 2.4 安装 Wheel

可以使用 pip 安装到本地 Python 的 site-packages 目录，命令：**pip install dist/文件名.whl**

这里我的命令是：**pip install dist/firstApp-1.0.0-py3-none-any.whl**

```
F:\python\python-3.8.10>pip install dist/firstApp-1.0.0-py3-none-any.whl --force-reinstall
WARNING: Ignoring invalid distribution -ip (f:\python\python-3.8.10\lib\site-packages)
WARNING: Ignoring invalid distribution -ip (f:\python\python-3.8.10\lib\site-packages)
Processing f:\python\python-3.8.10\dist\firstapp-1.0.0-py3-none-any.whl
Collecting numpy
  Downloading numpy-1.22.3-cp38-cp38-win_amd64.whl (14.7 MB)
  | 14.7 MB 3.2 MB/s
WARNING: Ignoring invalid distribution -ip (f:\python\python-3.8.10\lib\site-packages)
Installing collected packages: numpy, firstApp
  Attempting uninstall: numpy
    WARNING: Ignoring invalid distribution -ip (f:\python\python-3.8.10\lib\site-packages)
    Found existing installation: numpy 1.21.3
    Uninstalling numpy-1.21.3:
      Successfully uninstalled numpy-1.21.3
  Attempting uninstall: firstApp
    Found existing installation: firstApp 1.0.0
    Can't uninstall 'firstApp'. No files were found to uninstall.
WARNING: Ignoring invalid distribution -ip (f:\python\python-3.8.10\lib\site-packages)
WARNING: Ignoring invalid distribution -ip (f:\python\python-3.8.10\lib\site-packages)
Successfully installed firstApp-1.0.0 numpy-1.22.3
WARNING: Ignoring invalid distribution -ip (f:\python\python-3.8.10\lib\site-packages)
```

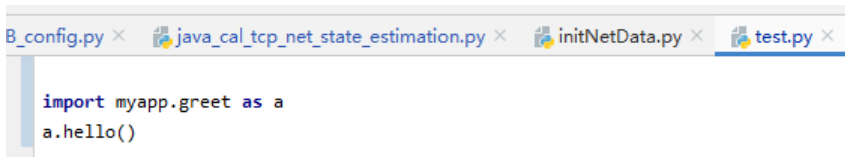
安装成功后，会在python安装路径\Lib\site-packages目录下看到功能模块：



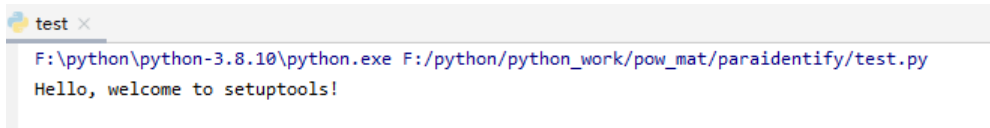
## 2.5 使用

现在和其他使用 pip 安装的三方库一样使用：  
测试内容：

```
1 | import myapp.greet as a
2 | a.hello()
```



输出为：



以上就是 Python 打包和分发的全部内容，当然是最简单的功能。更复杂的打包，比如模块过滤，非 py 文件打包，作者信息等常见的需求，都被实现在 setup() 内。  
详细介绍 setup()可参考链接: [Python打包分发工具setuptools](#).