

# 如何构建自己的python模块

## 前言

根据最佳实践或是默认约定，组织清晰合理的模块或项目目录结构。减少不必要的重复工作和杂乱无章的维护工作，快速进入业务开发。

## 1,目录结构

建立和模块名称一致的目录，比如我的模块名称为my1, 那就新建项目名秀称为my1。现在模块内的目录和文件结构如下图，并分别进行了说明。

- 目录说明：

docs目录，包含 了模块的文档;

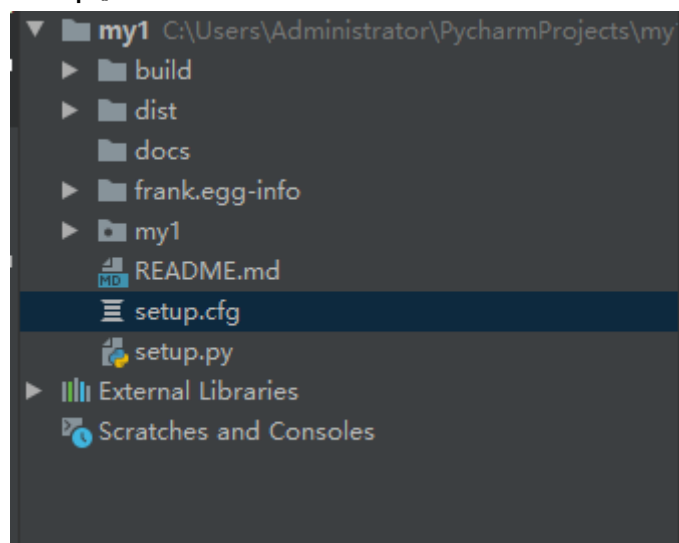
my1目录，至少要有有一个包的名称和项目的名称相同，注意这这是一个python的包，不是目录。

- 文件说明：

README.md 包含了模块的说明文 件，一般的内类容有模块功能; 依赖说明；安 装说明；联系人; 项目开发人员的联系方式等信息。这个文 件中的类容尽该可能详细，方便其他人尽可能简单快捷了解模块的基本信息。

setup.cfg,包含了模块的setup工具的配置信息，如果没有特别需求，保持 为空就可以了。

setup.py 包含了模块的安装信息，这是来自于python内置包的setuptools,调用了setup()函数，可以根据需求定义函数的常见信息。



比如我这个setup.py例子中的内容

```

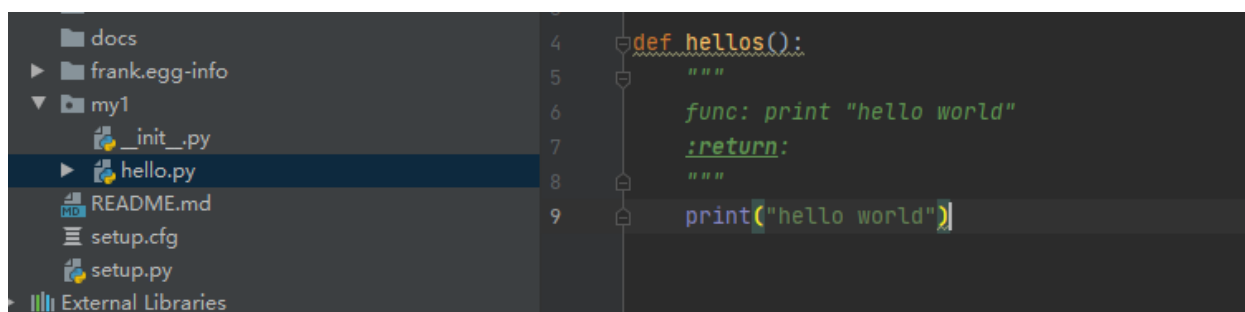
from setuptools import setup
from setuptools import find_packages

setup(name='frank',
      version='0.0.1',
      description='my1',
      author='my1',
      author_email='my1@gmail.com',
      install_requires=[],
      extras_require={
          'h5py': ['h5py'],
          'visualize': ['pydot-ng'],
      },
      packages=find_packages())

```

## 2, 模块发布

假设我现在在模块根目录下的my1的包,新建了一个python文件 hello.py,并在其中定义了一个函数hellos(),这个函数就是一个简单的输出"hello word"的功能.



```

def hellos():
    """
    func: print "hello world"
    :return:
    """
    print("hello world")

```

接下就可以发布模块了,比如可以使用操作系统做压缩文件或是直接使用目录分发(为了不引入更多的复杂性,本文的例子不使用版本管理工具,比如git,svn等)。

## 3,安装

现在模块使用者从开发者那里获取了安装目录或是压缩文件,如果是压缩文件,那首先要在操作系统中使用工具进行解压缩。

进入到操作系统的shell中 ( windows 是cmd命令提示符,linux是shell ), 切换到模块目录,使用python setup.py install进行安装

```

C:\Users\Administrator\PycharmProjects\my1>python.exe setup.py install

```

## 4,使用

模块安装完成，就可以在其他项目是进行引用了，引用方式

```
import prackagename
```

```
import argparse
import my1

def main():
    """the entrance of this file"""
    my1.hello.hellos()

if __name__ == '__main__':
    main()
```

## 后记

模块的目录或是文 件可以根据需求进行动态增加或是需求，但是我们要记住符合大多数人或是项目的约定和习惯.