

项目结构

一、试验介绍

1.1 实验内容

本实验阐述了一个完整的 Python 项目结构，你可以使用什么样的目录布局以及怎样发布软件到网络上。

1.2 知识点

- 创建项目，编写 `__init__` 文件
- 使用 `setuptools` 模块，编写 `setup.py` 和 `MANIFEST.in` 文件
- 创建源文件的发布版本
- 项目注册&上传到 PyPI

1.3 实验环境

- python3.5
- Xfce终端
- Vim

1.4 适合人群

本课程属于初级级别课程，不仅适用于那些有其它语言基础的同学，对没有编程经验的同学也非常友好

二、实验步骤

2.1 实验准备

我们的实验项目名为 *factorial*。

```
$ mkdir factorial
$ cd factorial/
```

2.2 主代码

我们给将要创建的 Python 模块取名为 *myfact*，因此我们下一步创建 *myfact* 目录。

```
$ mkdir myfact
$ cd myfact/
```

主代码将在 *fact.py* 文件里面。

```
"myfact module"

def factorial(num):
    """
    返回给定数字的阶乘值

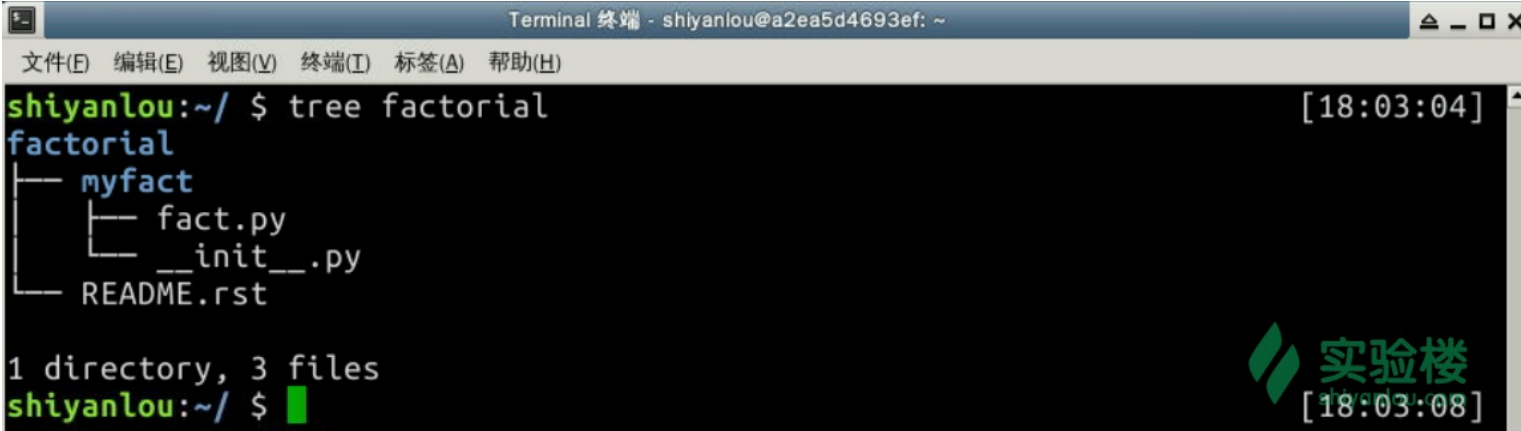
    :arg num: 我们将计算其阶乘的整数值

    :return: 阶乘值，若传递的参数为负数，则为 -1
    """
    if num >= 0:
        if num == 0:
            return 1
        return num * factorial(num - 1)
    else:
        return -1
```

我们还有模块的 `__init__.py` 文件，内容如下：

```
from fact import factorial
__all__ = [factorial, ]
```

我们还在 *factorial* 目录下添加了一个 *README.rst* 文件。因此，目录结构看起来像下面这样：



```
Terminal 终端 - shiyanlou@a2ea5d4693ef: ~
文件(E) 编辑(E) 视图(V) 终端(T) 标签(A) 帮助(H)
shiyanlou:~/ $ tree factorial
factorial
├── myfact
│   ├── fact.py
│   └── __init__.py
└── README.rst

1 directory, 3 files
shiyanlou:~/ $
```

The screenshot shows a terminal window with the title "Terminal 终端 - shiyanlou@a2ea5d4693ef: ~". The menu bar includes "文件(E)", "编辑(E)", "视图(V)", "终端(T)", "标签(A)", and "帮助(H)". The command `tree factorial` is executed, showing the directory structure of the `factorial` package. The output shows a directory named `factorial` containing a subdirectory `myfact` (which has `fact.py` and `__init__.py`) and a file `README.rst`. A summary line indicates "1 directory, 3 files". The prompt `shiyanlou:~/ $` is shown at the bottom. A watermark for "实验楼" (Experiment Tower) is visible in the bottom right corner.

2.3 MANIFEST.in

现在我们要写一个 `MANIFEST.in` 文件，它用来在使用 *sdist* 命令的时候找出将成为项目源代码压缩包一部分的所有文件。

```
include *.py
include README.rst
```

如果你想要排除某些文件，你可以在这个文件中使用 *exclude* 语句。

2.4 安装 python-setuptools 包

我们使用 *virtualenv* (这里不示范步骤)。

```
$ sudo pip3 install setuptools
```

2.5 setup.py

最终我们需要写一个 `setup.py`，用来创建源代码压缩包或安装软件。

```
#!/usr/bin/env python3
"""Factorial project"""
from setuptools import find_packages, setup

setup(name = 'factorial',
      version = '0.1',
      description = "Factorial module.",
      long_description = "A test module for our book.",
      platforms = ["Linux"],
      author="ShiYanLou",
      author_email="support@shiyancelou.com",
      url="https://www.shiyancelou.com/courses/596",
      license = "MIT",
      packages=find_packages()
)
```

name 是项目名称，*version* 是发布版本，*description* 和 *long_description* 分别是项目介绍，项目长描述。*platforms* 是此模块的支持平台列表。*find_packages()* 是一个能在你源目录下找到所有模块的特殊函数，packaging docs (<https://packaging.python.org/en/latest/distributing.html>)。

2.5.1. setup.py 用例

要创建一个源文件发布版本，执行以下命令。

```
$ python3 setup.py sdist
```

执行完毕会返回类似下面的信息：

```
running sdist
running egg_info
creating factorial.egg-info
writing factorial.egg-info/PKG-INFO
writing top-level names to factorial.egg-info/top_level.txt
writing dependency_links to factorial.egg-info/dependency_links.txt
writing manifest file 'factorial.egg-info/SOURCES.txt'
reading manifest file 'factorial.egg-info/SOURCES.txt'
reading manifest template 'MANIFEST.in'
writing manifest file 'factorial.egg-info/SOURCES.txt'
running check
creating factorial-0.1
creating factorial-0.1/factorial.egg-info
creating factorial-0.1/myfact
making hard links in factorial-0.1...
hard linking MANIFEST.in -> factorial-0.1
hard linking README.rst -> factorial-0.1
hard linking setup.py -> factorial-0.1
hard linking factorial.egg-info/PKG-INFO -> factorial-0.1/factorial.egg-info
hard linking factorial.egg-info/SOURCES.txt -> factorial-0.1/factorial.egg-info
hard linking factorial.egg-info/dependency_links.txt -> factorial-0.1/factorial.egg-info
hard linking factorial.egg-info/top_level.txt -> factorial-0.1/factorial.egg-info
hard linking myfact/__init__.py -> factorial-0.1/myfact
hard linking myfact/fact.py -> factorial-0.1/myfact
Writing factorial-0.1/setup.cfg
creating dist
Creating tar archive
removing 'factorial-0.1' (and everything under it)
```

我们能在 *dist* 目录下看到一个 tar 压缩包。

```
$ ls dist/
factorial-0.1.tar.gz
```

记住尝试安装代码时使用 `virtualenv`。

执行下面的命令从源代码安装。

```
$ sudo python3 setup.py install
```

学习更多可前往 packaging.python.org

(<https://packaging.python.org/en/latest/distributing.html>)。

2.6. Python Package Index (PyPI)

你还记得我们经常使用的 *pip* 命令吗？有没有想过这些包是从哪里来的？答案是 *PyPI*。这是 Python 的软件包管理系统。

为了实验，我们会使用 *PyPI* 的测试服务器 <https://testpypi.python.org/pypi> (<https://testpypi.python.org/pypi>)。

2.6.1. 创建账号

首先在这个链接 (https://testpypi.python.org/pypi?%3Aaction=register_form) 注册账号。你会收到带有链接的邮件，点击这个链接确认你的注册。

创建 `~/.pypirc` 文件，存放你的账号详细信息，其内容格式如下：

```
[distutils]
index-servers = pypi
                testpypi

[pypi]
repository: https://upload.pypi.org/legacy/
username: <username>
password: <password>

[testpypi]
repository: https://test.pypi.org/legacy/
username: <username>
password: <password>
```

替换 `<username>` 和 `<password>` 为您新创建的帐户的详细信息。在这里，由于我们是到 `testpypi` 的网页上去注册账号，即将相应的服务上传到 `testpypi`，所以在这里，你只需修改 `[testpypi]` 的用户名和密码

记得在 `setup.py` 中更改项目的名称为其它的名字来测试下面的指令，在接下来的命令中我将项目名称修改为 `factorial2`，为了不重复，需要自行修改至其它名

称。

2.6.2. 上传到 TestPyPI 服务

下一步我们会将我们的项目到 TestPyPI 服务。这通过 `twine` 命令完成。

我们也会使用 `-r` 把它指向测试服务器。

```
$ sudo pip3 install twine
$ twine upload dist/* -r testpypi
```

执行完毕会返回类似下面的信息,

```
Uploading distributions to https://test.pypi.org/legacy/
Uploading factorial2-0.1.tar.gz
```

在这里你也可以使用下面的命令上传到 PyPI 服务上，但这里需要注意，在 `~/.pypirc` 里面，你需要到 <https://pypi.python.org> 页面，按照上面的步骤去注册一个账号，然后到 `~/.pypirc` 的 `[pypi]` 下填写相应的用户名和密码。testpypi 和 pypi 的账号密码并不通用。

```
$ twine upload dist/* -r pypi
```

现在如果你浏览这个页面 (<https://testpypi.python.org/pypi/factorial2/0.1>)，你会发现你的项目已经准备好被别人使用了。

安全 | https://testpypi.python.org/pypi

应用 GitHub 收件箱 - yingque96 sf 橙子 - SegmentFault Python 语言参考 2.开始 - selenium-p cssselect: CSS Select

PyPI - the Python Package Index

The Python Package Index is a repository of software for the Python programming language. There are currently **17069** packages here.

To contact the PyPI admins, please use the [Support](#) or [Bug reports](#) links.

PACKAGE INDEX >>

- Browse packages
- List trove classifiers
- RSS (latest 40 updates)
- RSS (newest 40 packages)
- Terms of Service
- PyPI Tutorial
- PyPI Security
- PyPI Support
- PyPI Bug Reports
- PyPI Discussion
- PyPI Developer Info

ABOUT >>

NEWS >>

DOCUMENTATION >>

DOWNLOAD >>

COMMUNITY >>

FOUNDATION >>

CORE DEVELOPMENT >>

Welcome yingque96

[Your details \(Logout\)](#)

Your packages:

[factorial2](#)

Get Packages

To use a package from this index either "`pip install package`" (get `pip`) or download, unpack and "`python setup.py install`" it.

Package Authors

Submit packages with "`python setup.py upload`". The index [hosts package docs](#). You may also use the [web form](#). You must [register](#). Testing? Use [testpypi](#).

Infrastructure

To interoperate with the index use the [JSON](#), [OAuth](#), [XML-RPC](#) or [HTTP](#) interfaces. Use [local mirroring or caching](#) to make installation more robust.

Updated	Package	Description
2017-07-07	cloudshell-cp-aws 1.3.0.455	A repository for projects providing out of the box capabilities within CloudShell to define AWS instances in CloudShell and leverage Amazon Cloud Computing capabilities to deploy and connect apps in CloudShell sandboxes.
2017-07-07	factorial2 0.1	Factorial module.

实验楼 shiyanlou.com

三、总结

本实验使用了 `setuptools` 包，并完成了较为完整的项目创建&发布流程，最后还将项目发布到了网络（PyPI）上。

*本课程内容，由作者授权实验楼发布，未经允许，禁止转载、下载及非法传播。

上一节：测试 (/courses/596/labs/2052/document)

下一节：Flask 介绍 (/courses/596/labs/2054/document)