

Python计算生态构建

嵩天

Python第三方库的发布

高天

Python第三方库的发布

单元开篇

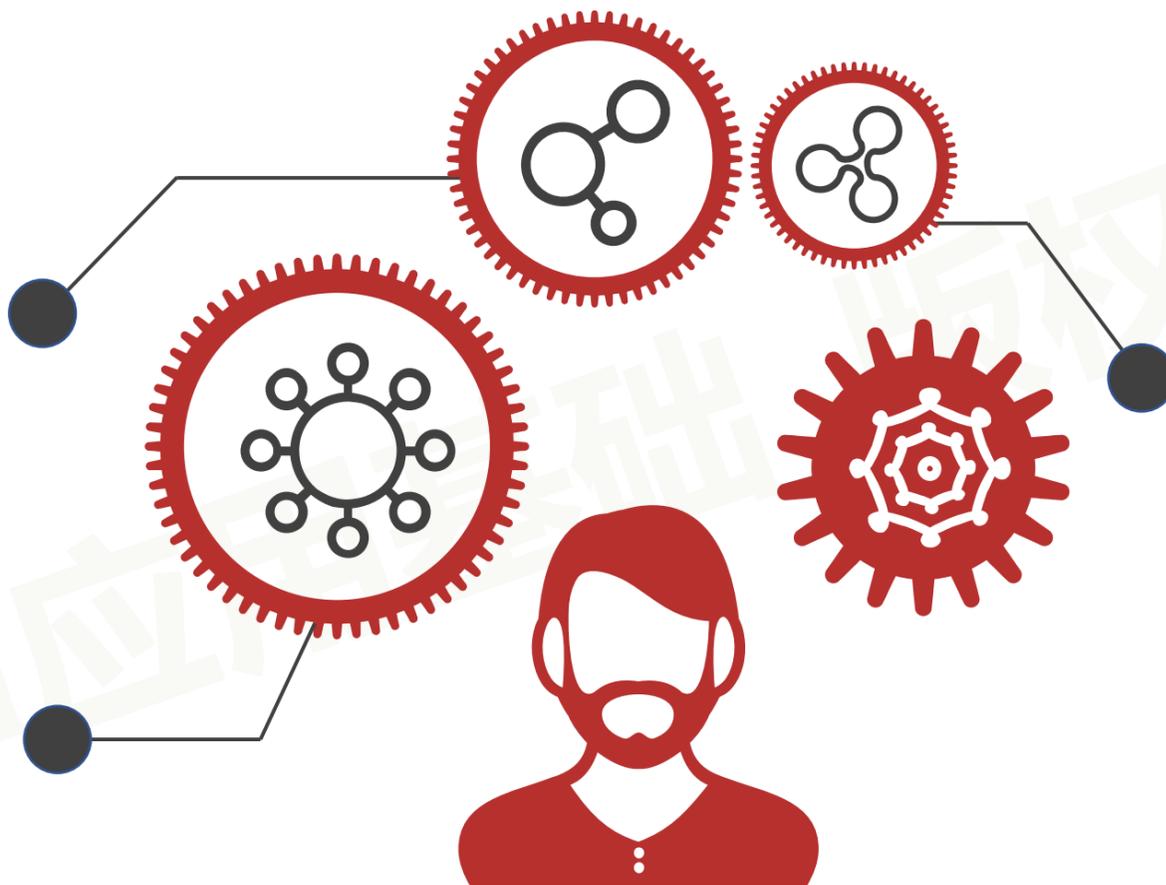
单元开篇

避免断更, 请加微信501863613

(2) 第三方库的发布流程

(1) 发布前的准备

(3) 第三方库的发布概念



Python第三方库的发布

Python第三方库的发布

发布前的准备

发布前的准备

Python第三方库发布

- PyPI: Python Package Index, 用来登记第三方库信息
- Github、bitbucket: 存储第三方库源代码及文档
- 目标: 通过pip进行安装和管理、源代码和文档网络可管理

发布前的准备

一些基本概念

- 项目 project: PyPI上一组发布和文件的统称
- 发布 release: 项目的一个特定版本, 每个发布有一个确定的版本号
- 文件 file: 即package, 一次发布包含的具体文件

发布前的准备

发布工具的使用



发布前的准备

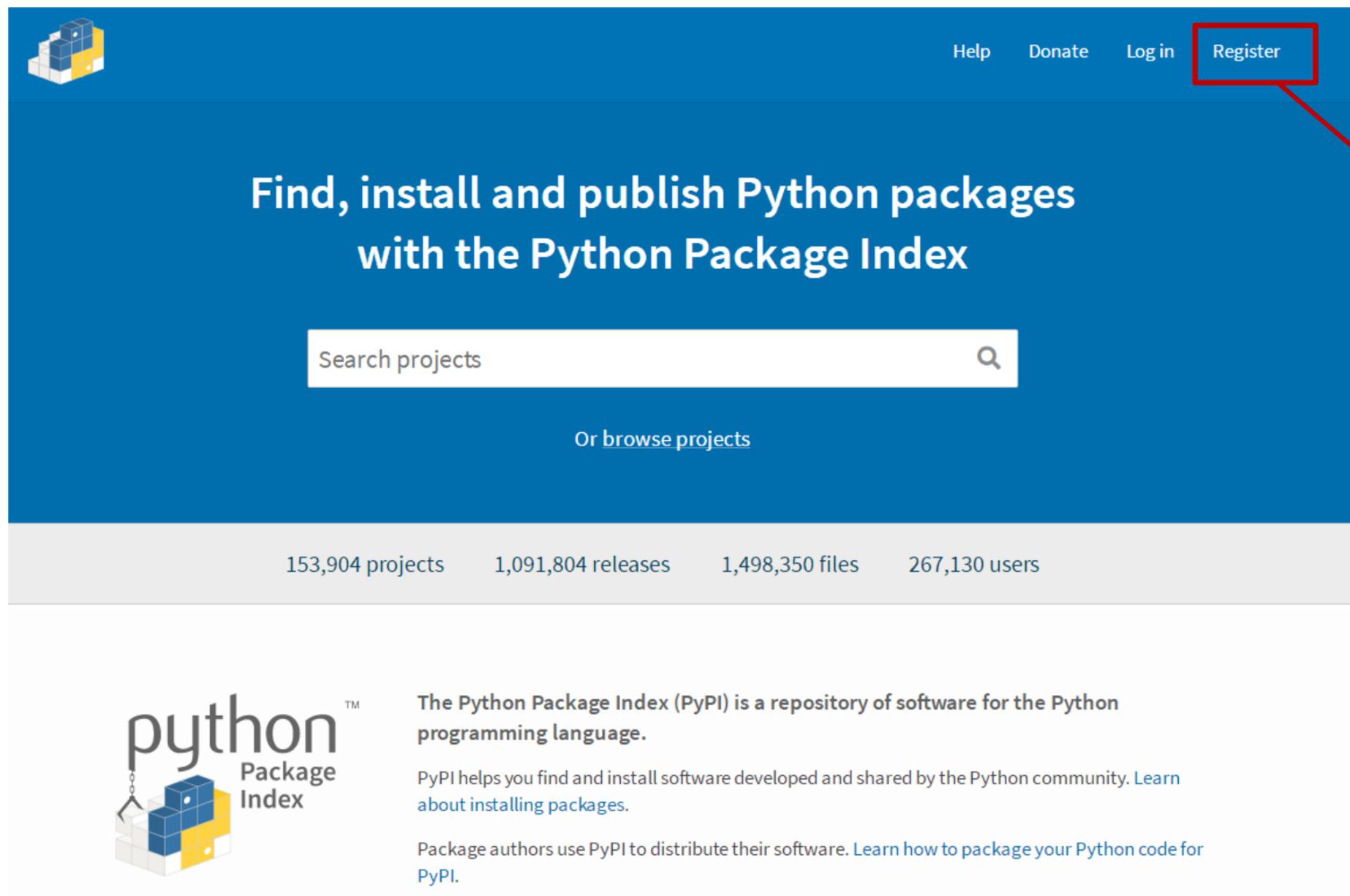
避免断更, 请加微信501863613

更新工具

- 更新setuptools、wheel、twine工具

```
python -m pip install --user --upgrade setuptools wheel twine
```

发布前的准备



注册账号

Python第三方库的发布

第三方库的发布流程

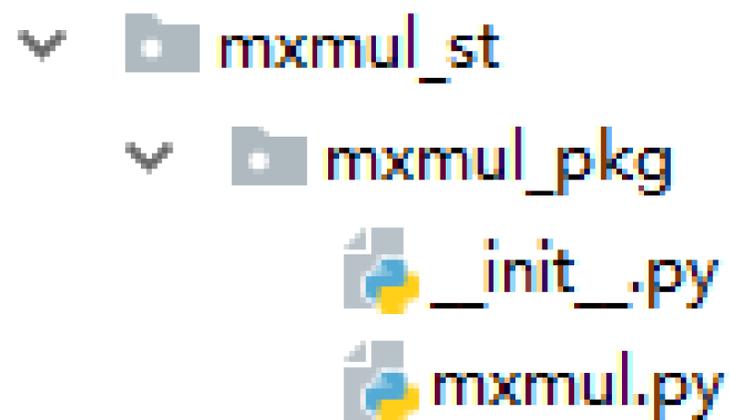
基本发布方法

待发布项目：矩阵乘法库

- 含一个.py文件：mxmul.py
- 代码参考：实例1

基本发布方法

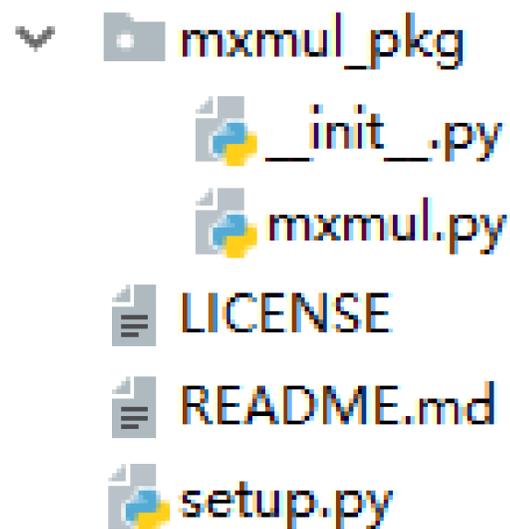
第一步：整理目录结构



- mxmul_st: 最顶层目录，相当于当前项目
 - 所有setuptools命令在最顶层目录中执行
- mxmul_pkg: 包目录，分发的主体
 - 所有源代码及层次化命名空间放到这里

基本发布方法

第二步：创建其他相关文件



- `setup.py`: 配置发布信息的文件，很重要
- `README.md`: markdown格式的说明文件
- `LICENSE`: 版权声明文件

基本发布方法

避免断更, 请加微信501863613

```
→ import setuptools
→
→ with open("README.md", "r") as fh:
    long_description = fh.read()

setuptools.setup(
    name="mxmul_pkg",
    version="0.1.0",
    author="Tian Song",
    author_email="songtian@bit.edu.cn",
    description="An example for teaching how to publish a Python package",
    long_description=long_description,
    long_description_content_type="text/markdown",
    url="https://github.com/pypa/sampleproject",
    packages=setuptools.find_packages(),
    classifiers=[
        "Programming Language :: Python :: 3",
        "License :: OSI Approved :: MIT License",
        "Operating System :: OS Independent",
    ],
)
```

setup.py

格式化程序

基本发布方法

setup.py

```
import setuptools

with open("README.md", "r") as fh:
    long_description = fh.read()

setuptools.setup(
    name="mxmul_pkg",
    version="0.1.0",
    author="Tian Song",
    author_email="songtian@bit.edu.cn",
    description="An example for teaching how to publish a Python package",
    long_description=long_description,
    long_description_content_type="text/markdown",
    url="https://github.com/pypa/sampleproject",
    packages=setuptools.find_packages(),
    classifiers=[
        "Programming Language :: Python :: 3",
        "License :: OSI Approved :: MIT License",
        "Operating System :: OS Independent",
    ],
)
```

各部分尽量填写

基本发布方法

README.md

```
# Example Package
```

```
This is a simple example package. You can use  
[Github-flavored Markdown] (https://guides.github.com/features/mastering-markdown/)  
to write your content.
```

Markdown格式，自由编写

基本发布方法

Copyright (c) 2018 The Python Packaging Authority

LICENSE

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

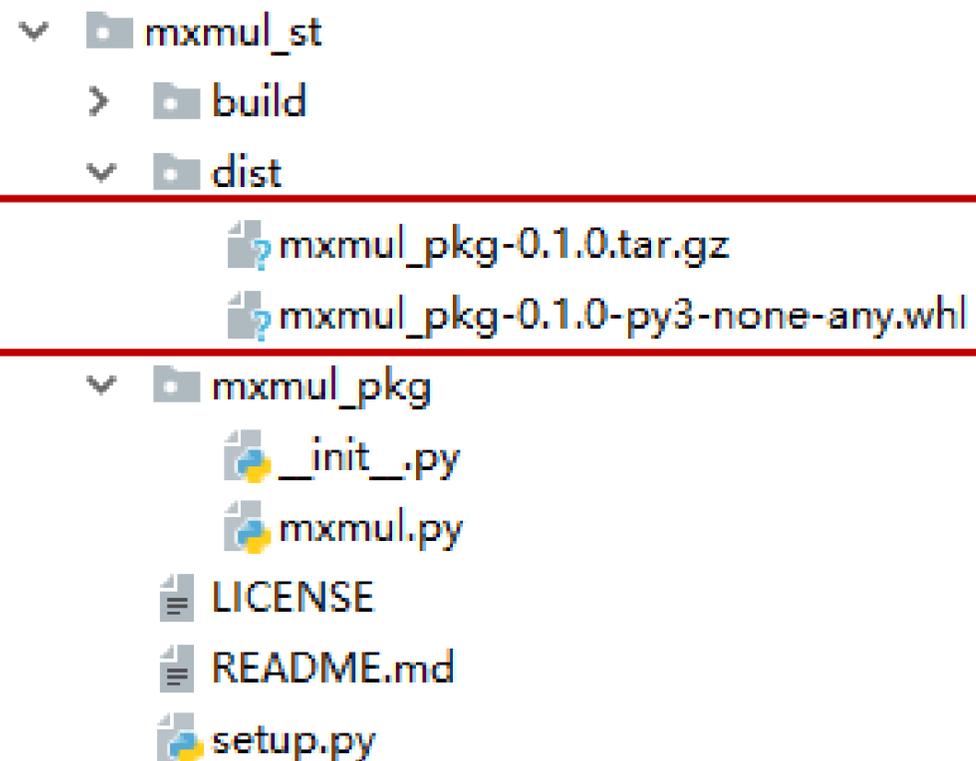
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

文本格式，自由编写

基本发布方法

第三步：执行打包命令



`python setup.py sdist bdist_wheel`

源发布

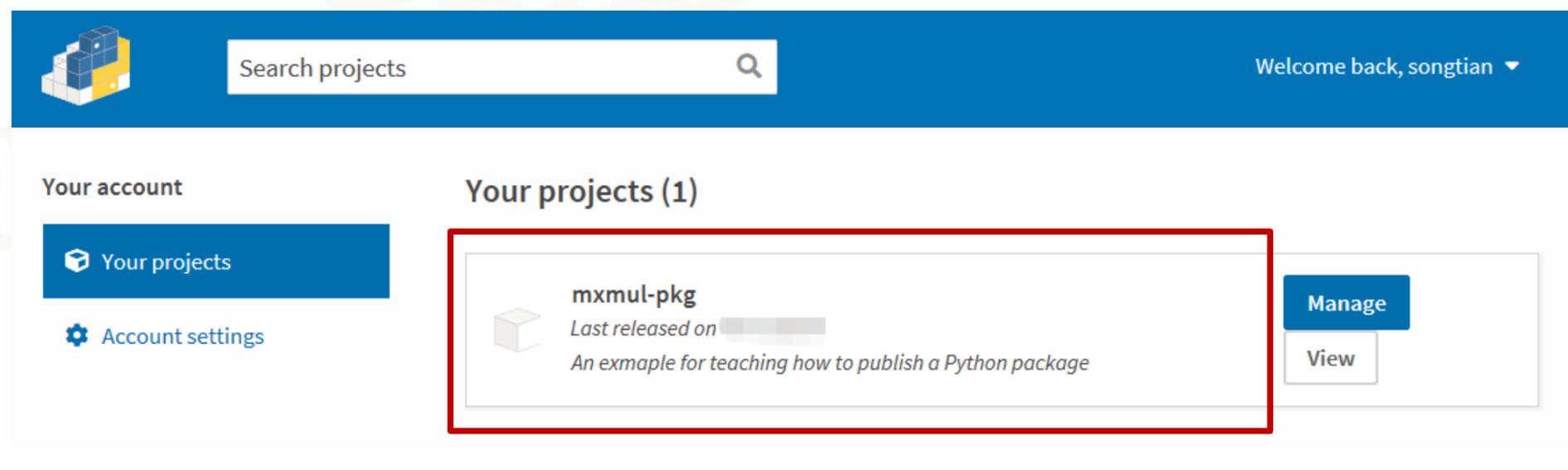
可执行发布

基本发布方法

避免断更, 请加微信501863613

第四步：执行发布命令

```
twine upload dist/*
```



基本发布方法

整体步骤

- 第一步：整理目录结构
- 第二步：创建其他相关文件，主要是setup.py
- 第三步：执行打包命令：`python setup.py ...`
- 第四步：执行发布命令：`twine upload dist/*`

Python第三方库的发布

第三方库的发布概念

与发布 Distribution 相关的概念

- 打包发布: Built Distribution, 对文件整理打包后进行发布
- 源发布: Source Distribution, 发布源代码的方式, 简称 sdist
- 可执行发布: Binary Distribution, 发布编译后可执行代码的方式

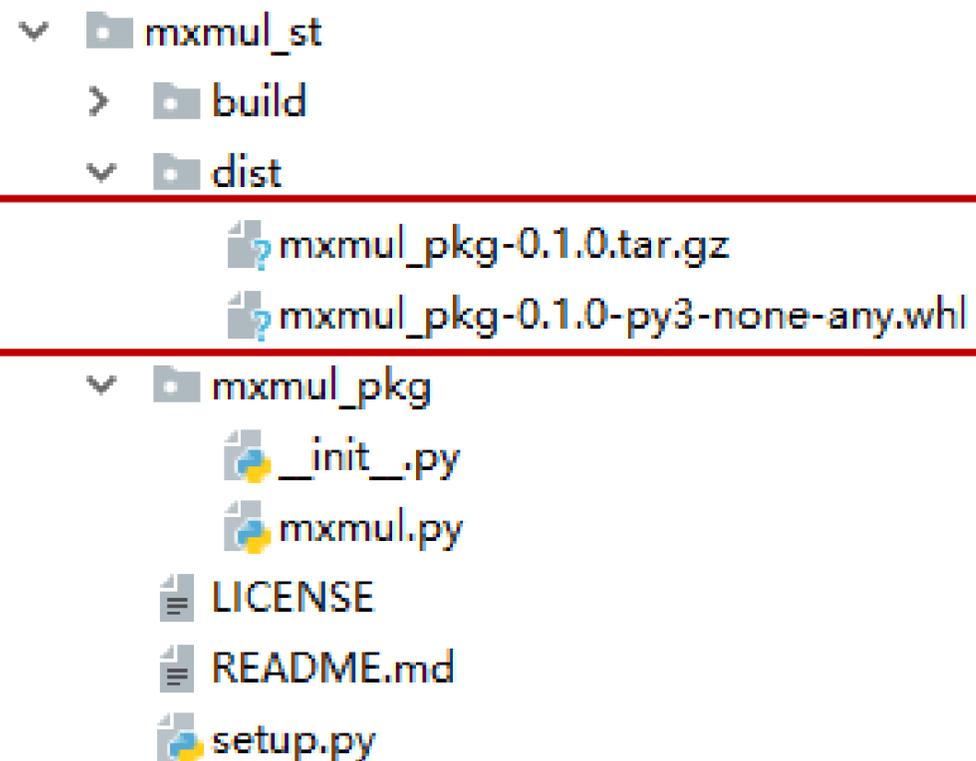
与发布 Distribution相关的概念逻辑

- 打包发布：一次发布一个打包文件，比如 .tar.gz 或 wheel文件
- 源发布：对于纯Python编写的库，源发布有最佳的适用性
- 可执行发布：对于各类多语言的扩展库，节省了本机编译需求，适用性较好

基本发布方法

避免断更, 请加微信501863613

第三步：执行打包命令



`python setup.py sdist bdist_wheel`

源发布

可执行发布

与打包格式相关的概念

- Egg: 一种打包格式, 已经被wheel替换
- Wheel: 一种打包格式, Python主流打包方式, PEP427, zip压缩
- Egg和Wheel都是打包格式, 可以包括源代码或可执行代码

Python第三方库的发布

单元小结

单元小结

避免断更, 请加微信501863613

(2) 第三方库的发布流程

四个步骤: 目录、setup.py、打包、发布

(1) 发布前的准备

基本概念、发布工具、账号

(3) 第三方库的发布概念

打包发布、源发布、可执行发布

Python第三方库的发布

 Python ▶ 123

Thank you