



亲自动手写一个Python库



地球的外...

Linux云计算和Python推动市场提升的学习研究者。

103 人赞同了该文章

分享一篇文章，比较详细的介绍了Python库的创作和发布过程，很有意义，原文在：[亲自动手写一个Python库（一）](#)和[亲自动手写一个python库（二）](#)。

引子

学习编程以来，接触过Basic,C/C++,Swift,JavaScript和Python五种语言，其中最爱的当属Python，简洁的语法和丰富的库让我一直沉迷于此，尽管最近实习工作中用的是C++。

最近一年，我将大把的时间投入到学习编程中，收获了满满的成就感，希望未来工作之后，还能保持对编程的爱。学习Python以后，经常感慨开源的伟大，我也一直希望自己能成为这光荣世界的一员，所以趁着工作中需要，利用业余时间开发一个Python库解决CAA开发中遇到的问题。

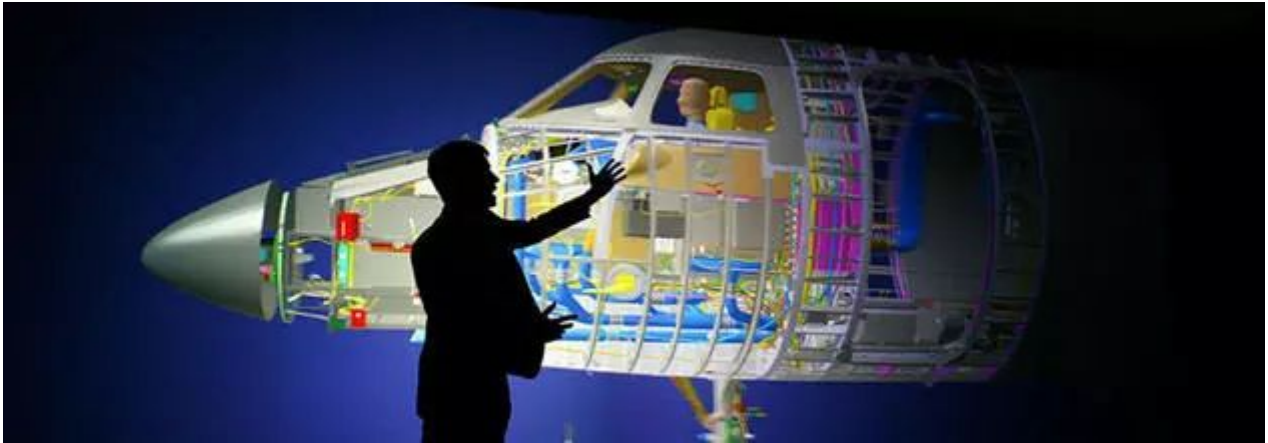
需求

▲ 赞同 103



● 2 条评论

行设计。从上世纪60年代CAD软件开始商用至今，历经技术的不断变革和各家公司的兼联合，已经形成达索、西门子、PTC“三国争霸”的局面。我实习公司主推的是达索的CAD软件，即CATIA。在CAD领域，我们将利用厂商提供的API在已经成型的CAD软件上增加一些程序（功能）称之为“二次开发”，上面所述的DS CAA就是对CATIA的二次开发。



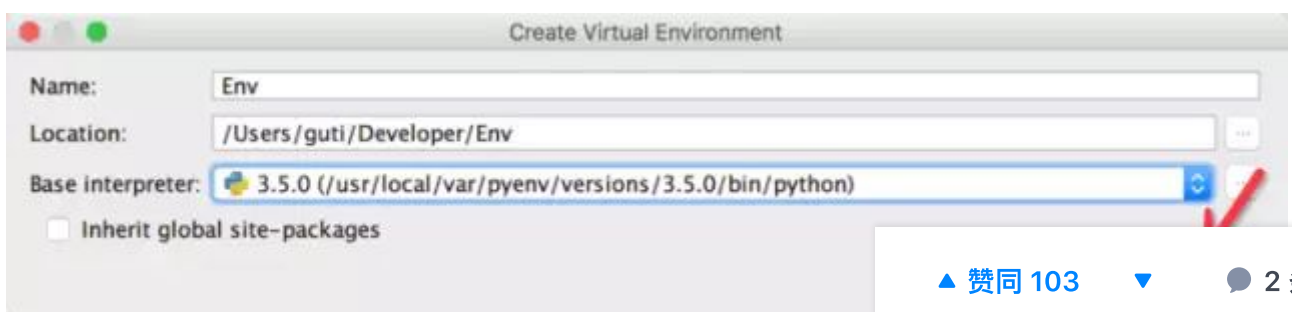
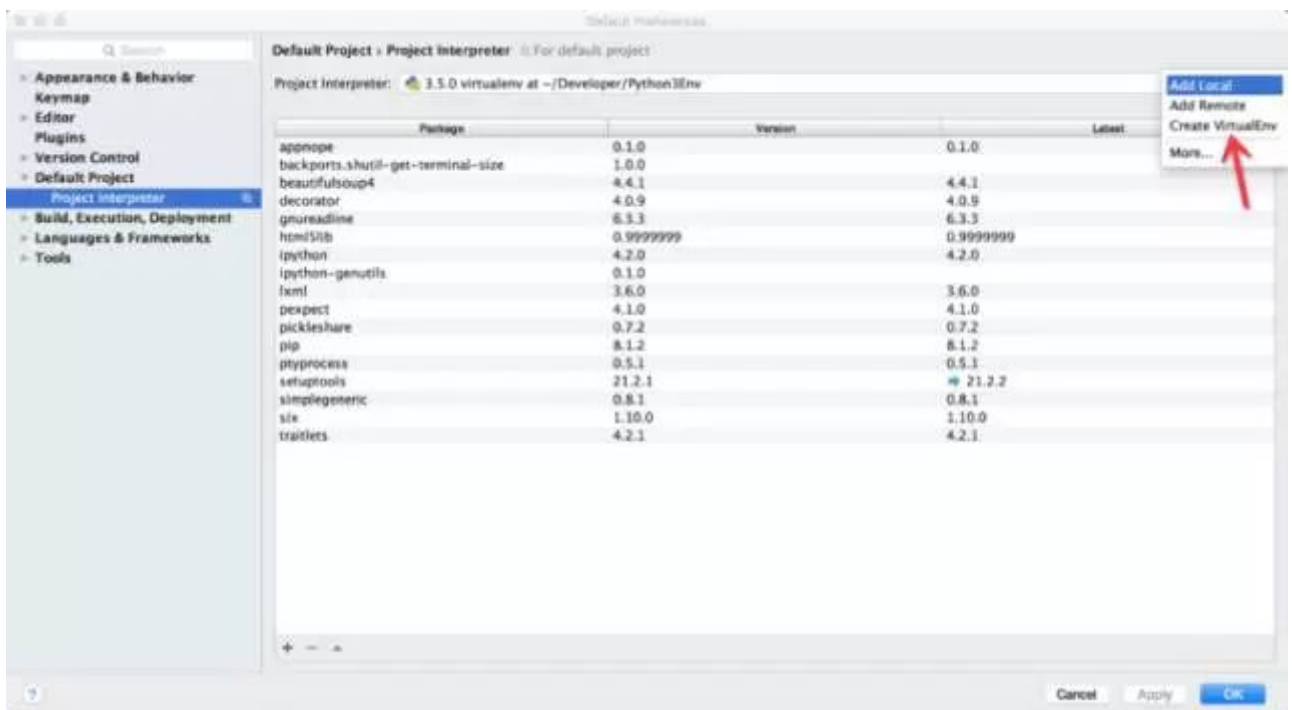
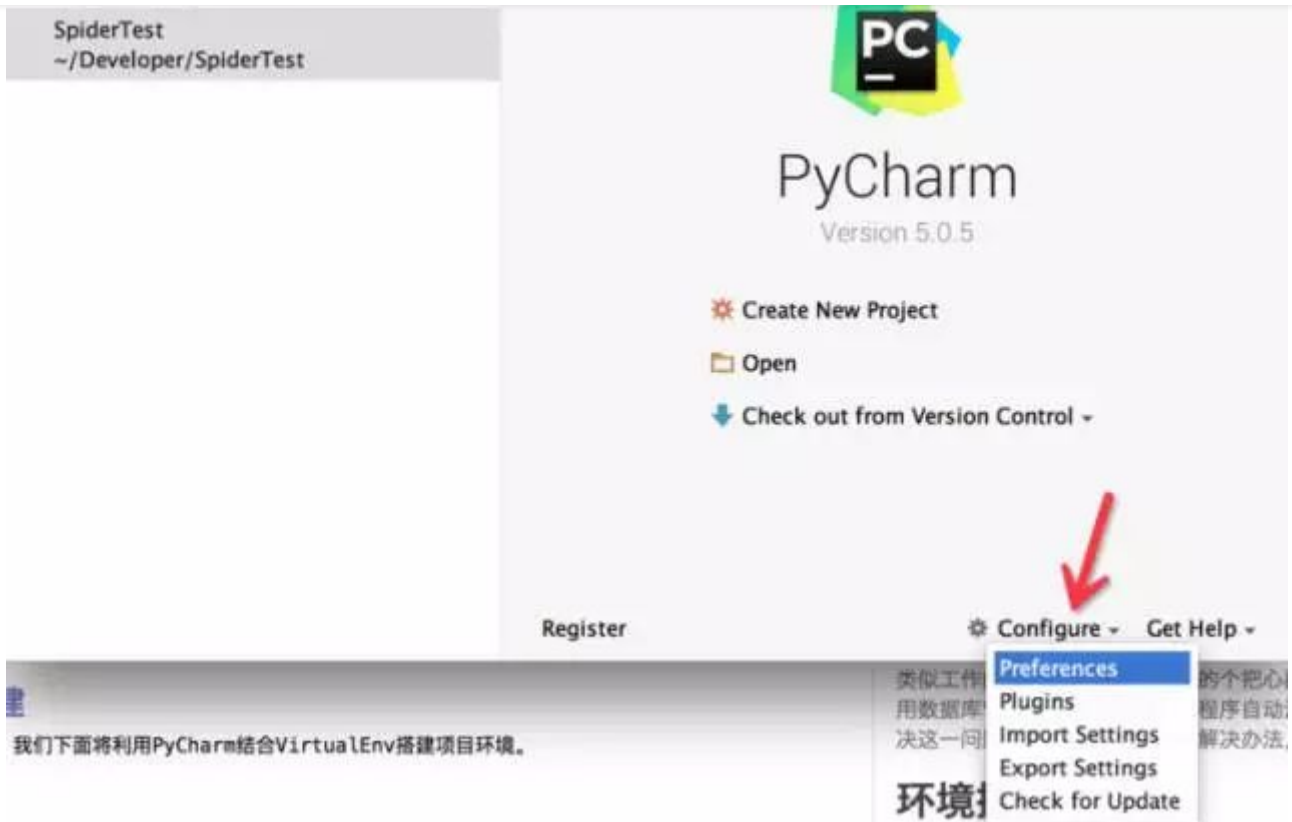
DS CAA采用的开发语言是C++，由于是完全的基于SDK和API的开发，往往开发工作要受限于达索的接口和框架。开发中大量的精力不得不花费在阅读达索API接口文档，查找API，查找模块，查找框架之中，做过类似工作的人一定能理解这其中的个把心酸。在慢慢的工作中，逐渐有了想法，将散落于文档中的API接口用数据库管理，根据自己所写的程序自动添加头文件，模块和框架。当然采用最爱的Python写一个库来解决这一问题，也许并不是最好的解决办法，但我愿意就好，也希望能帮助的苦逼的CAA开发者。

环境搭建

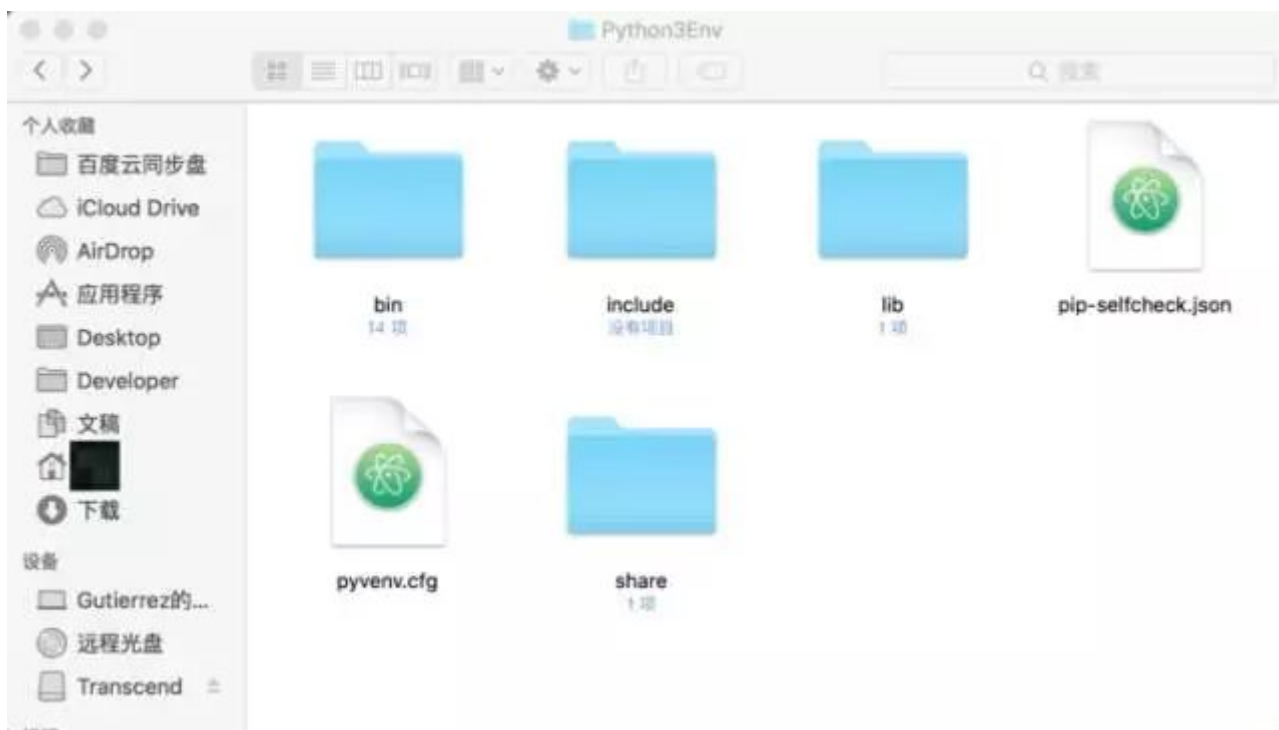
废话已经说了太多，我们下面将利用PyCharm结合VirtualEnv搭建项目环境。

安装好Pycharm后，点击configure->Preferecnces进入设置页面，找到Project Interpreter设置选项，该选项显示了当前可用的Python解释器，点击右上角设置图标，然后选择Create VirtualEnv来创建一个Python虚拟环境，虚拟环境的好处是你可以根据自己的虚拟Python环境为所欲为，而不用担心搞坏系统的Python环境，笔者曾经将Mac上的Python环境搞崩溃后，只得默默重装系统。

此外，Pycharm还可以方便的帮你管理虚拟环境的Python包，点击左下角+ 即可添加库到当前环境中。



```
// 激活虚拟环境
$ source ~/Developer/Python3Env/bin/activate
// 退出虚拟环境
$ deactivatea
```



接下来就用Pycharm创建一个项目，只需注意选择一个虚拟环境即可，我给自己的项目起了个还不错的名“CAAFinder”，项目已经放在GitHub上面，如果有CAA开发者看到，欢迎和我一起完善项目，也欢迎你可以测试它。即使你不是CAA开发者，也希望在Github上面点个赞。

CAAFinder

一款采用Python开发的达索CAA助手

主要功能

- Workspace管理
- 头文件、模块、框架自动补全
- 搭建快速测试模块
- snippet管理

开发日志

协作

欢迎大家一起来参与，联系方式：geekluca@qq.com

项目文档结构

首先我们先构建出自己项目的文档结构，首先是LICENSE和README.md，这两个文件并不是必要的，但是最好将其加上，方便别人了解你的项目。README.md是关于本项目的介绍，LICENSE是表示该项目所遵循的开源协议，一般MIT用的比较多，其他协议你可以百度谷歌了解一下。

接下来是打包库需要的一些文件，主要有MANIFEST.in和setup.py两个文件。MANIFEST.in表示打包时不会被自动包含进去的附加文件清单。setup.py是打包程序配置文件，需要设置一些选项。此外，setup.cfg是包含setup.py额外配置信息的文件，我没有用到这其中的设置，所以本项目中其是空的。

caafinder文件夹就是我们要写的Python程序，我们要注意这个文件夹里必须有__init__.py这个文件。

```
|— LICENSE
|— README.md
|— MANIFEST.in
|— caafinder
| |— __init__.py
| |— workspace.py
| |— database.py
|— setup.cfg
|— setup.py
```

```
from setuptools import setup, find_packages

setup(
    name = 'caafinder',
    version = '0.1.4',
    keywords='ds caa',
    description = 'a library for DS CAA Developer',
    license = 'MIT License',
    url = 'https://github.com/Gutier14/CAAFinder',
    author = 'Luca Liu',
    author_email = 'geekluca@qq.com',
    packages = find_packages(),
    include_package_data = True,
    platforms = 'any',
    install_requires = [],
)
```

可以参考一下：

- PyPA sample project这是一个官方示例库结构。
- Python Packaging User Guide这是Python打包的官方英文文档。

关于模块

模块也就是库，我们自己写程序时，通常通过import XX来导入一个需要的模块，在Python中，一个py文件就是一个模块，如一个abc.py的文件就是一个名字叫abc的模块。

你也许还想到，如果不同的人编写的模块名相同怎么办？为了避免模块名冲突，Python又引入了按目录来组织模块的方法，称为包（Package），包是一个带有__init__.py的文件夹，如图中文档所示，我们有一个包caafinder，因为它的子级有一个__init__.py，包caafinder拥有database和workspace两个模块。__init__.py可以是空文件，也可以有Python代码，因为__init__.py本身就是一个模块，而它的模块名就是caafinder。

```
|— caafinder
| |— __init__.py
| |— workspace.py
| |— database.py
```

通过执行下列语句来进行打包：

```
python setup.py xxx
```

其中xxx可以是下列几种方式中其中一个：

sdist	create a source distribution (tarball, zip file, etc.)
bdist	create a built (binary) distribution
bdist_dumb	create a "dumb" built distribution
bdist_rpm	create an RPM distribution
bdist_wininst	create an executable installer for MS Windows
bdist_egg	create an "egg" distribution

举个例子：

```
python setup.py sdist #生成的文件支持 pip
```

此时在根目录出现了dist文件夹，里面有name-version.tar.gz这个文件，这就是我们要发布到PyPI的压缩包了。

发布到PyPI

首先我们需要在PyPI上注册一个帐号，并在本地用户根目录下创建文件~/.pypirc，这样以后就不需要输入帐号密码了。

```
[distutils]
index-servers=pypi

[pypi]
repository = https://pypi.python.org/pypi
username = <username>
password = <password>
```

接下来，需要在PyPI网站上注册一个项目，网站提供三种方式注册，选择一种即可，最简单的是通过上传打包时生成的PKG-INFO文件，生成项目信息。此步骤只需在第一次发布时操作。


```
twine upload dist/*
```

此时在网页上就可以看到自己的源代码包啦，并且可以通过使用`pip install packagename`,就可以使用我们自己写的Python库了。



祝大家玩的开心，希望大家能到Github上帮忙点个赞。

你想更深入了解学习Python知识体系，你可以看一下我们花费了一个多月整理了上百小时的几百个知识点体系内容：

[【超全整理】《Python自动化全能开发从入门到精通》笔记全放送](#)

发布于 2017-12-19

[Python](#) [Python 入门](#) [Python 库](#)

文章被以下专栏收录



Python头条

专注分享高价值Python技术知识，欢迎添加群聊：515237230

进入专栏

▲ 赞同 103 ▼

● 2 条评论



首发于
Python头条



最好用的两款Python IDE

云同学



VS Code + Ana
的Python环境

挖掘机小王子

2 条评论

⇌ 切换为时间排序

写下你的评论...



李钊

1 年前

插眼

👍 赞



袁小艳

1 年前

能否直接用Python对Catia做二次开发

👍 赞

▲ 赞同 103 ▼

💬 2 条评论