

目录

前言	1.1
Python虚拟环境概述	1.2
virtualenv	1.3
pipenv	1.4
常见问题和心得	1.4.1
附录	1.5
参考资料	1.5.1

隔离Python项目环境：虚拟环境

- 最新版本： `v1.0`
- 更新时间： `20210705`

简介

介绍用于隔离Python项目环境的虚拟环境，主要是virtualenv和pipenv。

源码+浏览+下载

本书的各种源码、在线浏览地址、多种格式文件下载如下：

Gitbook源码

- [crifan/isolate_python_project_virtual_environment](#): 隔离Python项目环境：虚拟环境

如何使用此Gitbook源码去生成发布为电子书

详见：[crifan/gitbook_template: demo how to use crifan gitbook template and demo](#)

在线浏览

- 隔离Python项目环境：虚拟环境 [book.crifan.com](#)
- 隔离Python项目环境：虚拟环境 [crifan.github.io](#)

离线下载阅读

- 隔离Python项目环境：虚拟环境 PDF
- 隔离Python项目环境：虚拟环境 ePub
- 隔离Python项目环境：虚拟环境 Mobi

版权说明

此电子书教程的全部内容，如无特别说明，均为本人原创和整理。其中部分内容参考自网络，均已备注了出处。如有发现侵犯您的版权，请通过邮箱联系我 `admin` 艾特 `crifan.com`，我会尽快删除。谢谢合作。

鸣谢

感谢我的老婆陈雪的包容理解和悉心照料，才使得我 crifan 有更多精力去专注技术专研和整理归纳出这些电子书和技术教程，特此鸣谢。

更多其他电子书

本人 crifan 还写了其他 100+ 本电子书教程，感兴趣可移步至：

[crifan/crifan_ebook_readme: Crifan的电子书的使用说明](#)

crifan.com，使用署名4.0国际(CC BY 4.0)协议发布 all right reserved,
powered by Gitbook最后更新： 2021-07-17 12:23:11

Python虚拟环境概述

Python项目开发期间，常涉及到不同的Python环境：

- Python版本不同
 - Python的大版本不同：Python2或Python3
 - Python的小版本不同
 - Python2：比如 Python 2.6 / Python 2.7
 - Python3：比如 Python 3.6 / Python 3.7 / Python 3.8 /...
- 每个项目所安装的库的版本不同
 - 比如
 - 某些项目需要某个库的特定的版本：1.1.0b3 的 sqlalchemy

Python虚拟环境工具对比

- pyenv
 - 概述
 - Python 3.3 和 3.4 中创建虚拟环境的推荐工具
 - 从 Python 3.6 之后
 - 不推荐使用：pyenv
 - 推荐使用：venv
- venv
 - 概述
 - Python 3.3 之后，标准库自带的模块，虚拟环境创建和管理工具：venv
 - 原理和作用类似于virtualenv，在一定程度上能够替代 virtualenv
 - 目前来说，社区用 virtualenv 更多，暂时没太多人说要用 venv 取代掉 virtualenv
 - venv 是 Python3.3 才有的，Python2.X 不能使用
 - 官网
 - venv — Creation of virtual environments — Python 3.8.2rc1 documentation
 - <https://docs.python.org/3/library/venv.html>
- virtualenv
 - 概述
 - 之前常用的，虚拟环境工具
 - virtualenv 同时支持 Python2.X 和 Python3.X
 - 特别是在当前的生产环境中 Python2.X 还占有很大比例的情况下，我们依然需要 virtualenv

- `virtualenv`: 很常用的工具, 用于创建虚拟环境
- 隔绝不同项目, 使用不同Python环境和版本
- 官网 PyPA 也很认可
- `virtualenvwrapper`: 一堆的`virtualenv`的扩展的集合
 - 内含工具
 - `mkvirtualenv`
 - `lssitepackages`
 - `workon`: 切换多个虚拟环境
- `pipenv`
 - 概述
 - `requests`的作者写的
 - 希望把 `Pipfile`, `pip`, `virtualenv` 集成到一起
 -
- 另外: Python多版本管理工具
 - `pyenv`
 - 概述
 - 第三方的、开源的多版本 Python 管理工具
 - 用以管理在一台机器上多个 Python 发行版本的共存问题
 - 隔离多个Python版本
 - 比如一台 Linux 机器上同时安装 `Python2.7`、`Python3.4`、`Python3.5` 三个版本的管理
 - `pyenv-virtualenv`
 - `pyenv`的插件: 同时可以使用`pyenv`和 `virtualenv`
 - `pyenv-virtualenvwrapper`: 把 `virtualenvwrapper` 集成到了 `pyenv`

Python虚拟环境工具结论

- 结论
 - 优先推荐: 简单方便好用的 `virtualenv`
 - 其次推荐: `locking`有点耗时, 但也很好用的: `pipenv`

crifan.com, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved,
powered by Gitbook最后更新: 2021-07-17 12:19:02

virtualenv

virtualenv 是目前流行度最高的Python虚拟环境管理工具之一。

最大特点，简单方便易用。

安装

```
pip install virtualenv
```

使用

- 概述
 - 创建: `virtualenv venv`
 - 进入: `source venv/bin/activate`
 - 管理库: 用 `pip`
 - 安装: `pip install xxx`
 - 保存虚拟环境=冻结依赖的库: `pip freeze > requirements.txt`
 - 恢复虚拟环境=恢复安装依赖的库: `pip install -r requirements.txt`
 - 退出: `deactivate`
 - 删除: `rm -rf venv`

创建虚拟环境

- 语法

```
virtualenv YourVirtualEnvName
```

- 最常用写法

```
virtualenv venv
```

- 大家约定俗成把虚拟环境名称写成: `venv`
 - 你看到项目根目录下有个 `venv` , 往往就可以判断出, 当前项目Python虚拟环境工具是 `virtualenv` 了
- 举例

```
□ virtualenv venv
Using base prefix '/Users/limao/.pyenv/versions/3.8
New python executable in /Users/limao/dev/xxx/AppCr
Also creating executable in /Users/limao/dev/xxx/Ap
Installing setuptools, pip, wheel...
done.
```

激活虚拟环境并进入

- 概述

- Mac/Linux

```
source venv/bin/activate
```

- 或

```
. venv/bin/activate
```

- Win

```
venv\Scripts\activate.bat
```

- 详解

Platform	Shell	Command to activate virtual environment
POSIX	bash/zsh	<pre>\$ source <venv>/bin/activate</pre>
	fish	<pre>\$. <venv>/bin/activate.fish</pre>
	csh/tcsh	<pre>\$ source <venv>/bin/activate.csh</pre>
	PowerShell Core	<pre>\$ <venv>/bin/Activate.ps1</pre>
Windows	cmd.exe	<pre>C:\> <venv>\Scripts\activate.bat</pre>
	PowerShell	<pre>PS C:\> <venv>\Scripts\Activate.ps1</pre>

- 说明

- 进入虚拟环境后，你会看到，终端最前面多出个 (venv)，表示你进入了虚拟环境了。

- 举例

- (venv) limao@xxx □ ~/dev/xxx/AppCrawler □ □
master ● □

注:

- 其他很多虚拟环境工具，都是类似的逻辑，直接删除对应文件夹即可
 - `virtualenv`
 - `venv`
 - `Anaconda environment`
 - `pyenv`
- 而有些虚拟环境工具，是有专门的命令的（在虚拟环境根目录中执行）
 - `pipenv`
 - `pipenv --rm`
 - `virtualenvwrapper`
 - `rmvirtualenv`

附录

查看virtualenv版本

查看此处virtualenv工具的版本:

```
□ virtualenv --version
16.7.9
```

文档

- 官方文档
 - Virtualenv — virtualenv 20.4.8.dev5+g5657d56 documentation
 - <https://virtualenv.pypa.io/en/latest/>

crifan.com, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved,
powered by Gitbook最后更新: 2021-07-17 12:19:52

pipenv

- pipenv
 - 概述: pipenv是Python虚拟环境管理工具
 - 优点: 功能简单易用
 - 缺点
 - locking卡死或耗时很长
 - 历史上(2020年前后), 很长一段时间, locking卡死问题, 一直没解决, 导致几乎不可用
 - 一个简单的 `pipenv install xxx` 安装可能瞬间就完成, 但是后续locking可能要数个小时
 - 最新 20210705, locking问题, 基本上解决
 - 现在 `pipenv install xxx` 后, locking可能要耗时, 几十秒 或者 几分钟
 - 算还能忍, 虽然不爽, 但基本可用

安装

```
pip install pipenv
```

使用

- 概述
 - 创建虚拟环境

```
pipenv install
```
 - 创建虚拟环境+安装库

```
pipenv install xxx
```
 - 进入虚拟环境

```
pipenv shell
```
 - 管理库
 - 用 pipenv 或 pip
 - 用 pipenv : 无需进入虚拟环境, 即可去安装和管理库

```
pipenv install xxx
```



```
→ xxx git:(master) x ll /Users/crifan/.local/share/virtua
total 0
drwxr-xr-x  8 crifan  staff   256B  4 21 14:47 AutocarData-
drwxr-xr-x  7 crifan  staff   224B  5 18 15:58 xxx-SCpLPEyZ
drwxr-xr-x  7 crifan  staff   224B  4 26 16:29 testDownload
→ xxx git:(master) x ll /Users/crifan/.local/share/virtua
total 8
drwxr-xr-x 21 crifan  staff   672B  5 18 15:58 bin
drwxr-xr-x  3 crifan  staff    96B  5 18 15:58 include
drwxr-xr-x  3 crifan  staff    96B  5 18 15:58 lib
-rw-r--r--  1 crifan  staff    61B  5 18 15:58 pip-selfche
```

进入虚拟环境

```
pipenv shell
```

举例：

```
→ xxx git:(master) x pipenv shell
Spawning environment shell (/bin/zsh). Use 'exit' to leave.
. /Users/crifan/.local/share/virtualenvs/xxx-SCpLPEyZ/bin/act
→ xxx git:(master) x . /Users/crifan/.local/share/virtuale
```

进去后，对应的python就是你虚拟环境中的python了：

```
→ xxx git:(master) x which python
/Users/crifan/.local/share/virtualenvs/xxx-SCpLPEyZ/bin/pyt
```

管理Python库

用pipenv管理python库

语法：

```
pipenv install xxx
```

举例：

```
pipenv install requests  
  
pipenv install pyspider  
  
pipenv install flask-restful  
  
pipenv install mysql  
pipenv install mysql-connector-python  
  
pipenv install pymysql  
  
pipenv install openpyxl  
  
pipenv install gunicorn  
  
pipenv install flask-pymongo  
  
pipenv install flask-cors
```

特殊:

- 安装某库(A), 指定所依赖的其他库(B)
 - 举例

```
pipenv install "celery[redis]"
```

用pip管理Python库

```
pipenv shell  
  
pip install browsermob-proxy
```

注: 通过 pip 安装的库, 无法自动进入 pipenv 的管理系统的内部逻辑

-》比如 pipenv graph 就无法看到对应的库了

-》就不方便 pipenv 的自动管理已安装的库了

查看已安装的库

```
pipenv graph
```

举例:

```
→ xxxDemoServer git:(master) x pipenv graph
celery==4.1.0
- billiard [required: >=3.5.0.2,<3.6.0, installed: 3.5.0.2]
- kombu [required: <5.0,>=4.0.2, installed: 4.1.0]
  - amqp [required: >=2.1.4,<3.0, installed: 2.2.2]
    - vine [required: >=1.1.3, installed: 1.1.4]
  - pytz [required: >dev, installed: 2018.4]
Flask-Cors==3.0.4
- Flask [required: >=0.9, installed: 1.0.2]
- click [required: >=5.1, installed: 6.7]
- itsdangerous [required: >=0.24, installed: 0.24]
- Jinja2 [required: >=2.10, installed: 2.10]
  - MarkupSafe [required: >=0.23, installed: 1.0]
  - Werkzeug [required: >=0.14, installed: 0.14.1]
- Six [required: Any, installed: 1.11.0]
Flask-PyMongo==0.5.1
- Flask [required: >=0.8, installed: 1.0.2]
- click [required: >=5.1, installed: 6.7]
- itsdangerous [required: >=0.24, installed: 0.24]
- Jinja2 [required: >=2.10, installed: 2.10]
  - MarkupSafe [required: >=0.23, installed: 1.0]
  - Werkzeug [required: >=0.14, installed: 0.14.1]
- PyMongo [required: >=2.5, installed: 3.6.1]
Flask-RESTful==0.3.6
- aniso8601 [required: >=0.82, installed: 3.0.0]
- Flask [required: >=0.8, installed: 1.0.2]
- click [required: >=5.1, installed: 6.7]
- itsdangerous [required: >=0.24, installed: 0.24]
- Jinja2 [required: >=2.10, installed: 2.10]
  - MarkupSafe [required: >=0.23, installed: 1.0]
  - Werkzeug [required: >=0.14, installed: 0.14.1]
- pytz [required: Any, installed: 2018.4]
- six [required: >=1.3.0, installed: 1.11.0]
unicorn==19.8.1
openpyxl==2.5.3
- et-xmlfile [required: Any, installed: 1.0.1]
- jdcal [required: Any, installed: 1.4]
PyMySQL==0.8.1
redis==2.10.6
requests==2.18.4
- certifi [required: >=2017.4.17, installed: 2018.4.16]
- chardet [required: <3.1.0,>=3.0.2, installed: 3.0.4]
- idna [required: >=2.5,<2.7, installed: 2.6]
- urllib3 [required: <1.23,>=1.21.1, installed: 1.22]
```

目的=用途：可以清晰的看出安装了哪些库，每个库的依赖有哪些。

比如我此处关心的：`celery==4.1.0`

举例2：

```
# pipenv graph
Flask-Cors==3.0.4
- Flask [required: >=0.9, installed: 1.0.1]
- click [required: >=5.1, installed: 6.7]
- itsdangerous [required: >=0.24, installed: 0.24]
- Jinja2 [required: >=2.10, installed: 2.10]
  - MarkupSafe [required: >=0.23, installed: 1.0]
- Werkzeug [required: >=0.14, installed: 0.14.1]
- Six [required: Any, installed: 1.11.0]
Flask-PyMongo==0.5.1
- Flask [required: >=0.8, installed: 1.0.1]
- click [required: >=5.1, installed: 6.7]
- itsdangerous [required: >=0.24, installed: 0.24]
- Jinja2 [required: >=2.10, installed: 2.10]
  - MarkupSafe [required: >=0.23, installed: 1.0]
- Werkzeug [required: >=0.14, installed: 0.14.1]
- PyMongo [required: >=2.5, installed: 3.6.1]
Flask-RESTful==0.3.6
- aniso8601 [required: >=0.82, installed: 3.0.0]
- Flask [required: >=0.8, installed: 1.0.1]
- click [required: >=5.1, installed: 6.7]
- itsdangerous [required: >=0.24, installed: 0.24]
- Jinja2 [required: >=2.10, installed: 2.10]
  - MarkupSafe [required: >=0.23, installed: 1.0]
- Werkzeug [required: >=0.14, installed: 0.14.1]
- pytz [required: Any, installed: 2018.4]
- six [required: >=1.3.0, installed: 1.11.0]
unicorn==19.8.1
openpyxl==2.5.3
- et-xmlfile [required: Any, installed: 1.0.1]
- jdcal [required: Any, installed: 1.4]
PyMySQL==0.8.0
```

删除虚拟环境

确保在虚拟环境根目录下，去运行：

```
pipenv --rm
```

举例：

```
→ server pipenv --rm
Removing virtualenv (/Users/crifan/.local/share/virtualenvs/
```

Pipenv的缺点

本地和别处Python版本不一致导致代码运行逻辑出错

有时候，由于无法完美复制虚拟环境，（Mac）本地是python 3.6，（CentOS）服务器中是Python 3.4，导致部分语法支持不同，而代码出错：

详见：

【已解决】Python中两个星号**参数去传递给函数出错：SyntaxError invalid syntax

不能完美复制虚拟环境，比如：

本地python 3.6，线上只有python 3.4

造成问题：

1. 【已解决】Python中两个星号**参数去传递给函数出错：SyntaxError invalid syntax
2. 【已解决】脚本编写系统Python返回的dict对应的一级topic没有排序

附录

pipenv的help


```
→ pipenv --help
Usage: pipenv [OPTIONS] COMMAND [ARGS]...

Options:
  --where          Output project home information.
  --venv           Output virtualenv information.
  --py             Output Python interpreter information.
  --envs          Output Environment Variable options.
  --rm            Remove the virtualenv.
  --bare          Minimal output.
  --completion    Output completion (to be eval'd).
  --man           Display manpage.
  --three / --two Use Python 3/2 when creating virtualenv.
  --python TEXT   Specify which version of Python virtualenv
  --site-packages Enable site-packages for the virtualenv.
  --version       Show the version and exit.
  -h, --help     Show this message and exit.

Usage Examples:
  Create a new project using Python 3.6, specifically:
  $ pipenv --python 3.6

  Install all dependencies for a project (including dev):
  $ pipenv install --dev

  Create a lockfile containing pre-releases:
  $ pipenv lock --pre

  Show a graph of your installed dependencies:
  $ pipenv graph

  Check your installed dependencies for security vulnerabilities:
  $ pipenv check

  Install a local setup.py into your virtual environment:
  $ pipenv install -e .

  Use a lower-level pip command:
  $ pipenv run pip freeze

Commands:
  check    Checks for security vulnerabilities and against
           provided in Pipfile.
  clean    Uninstalls all packages not specified in Pipfile.
  graph    Displays currently-installed dependency graph.
  install  Installs provided packages and adds them to Pipfile
           (if is given), installs all packages.
  lock     Generates Pipfile.lock.
  open     View a given module in your editor.
```

```
run      Spawns a command installed into the virtualenv
shell    Spawns a shell within the virtualenv.
sync    Installs all packages specified in Pipfile.lock
uninstall Un-installs a provided package and removes it
update   Runs lock, then sync.
```

crifan.com, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved,
powered by Gitbook最后更新: 2021-07-17 12:17:07

常见问题和心得

Warning Python 3.6 was not found on your system

背景：用pipenv去创建和初始化虚拟环境，但是报错：

```
# pipenv install
Warning: Python 3.6 was not found on your system
You can specify specific versions of Python with:
$ pipenv --python path/to/python
```

错误原因：此处已有带 Pipfile 是别处（本地Mac）的系统创建的，其 Python 是 3.6 。当前系统（在线CentOS服务器）Python 是 Python 3.4 ，没有希望的 Python 3.6 ，所以报错。

解决办法：

此处解决问题思路是：

- 要么重新安装Mac本地的Python3为Python 3.4，重新弄出pipenv的环境
- 要么重新安装服务器中的Python3为Python 3.6，这样就和本地Mac的Pipfile一致了

不过此处情况稍微有点特殊：

两种方式都不想做，觉得一是麻烦，二是考虑到目前Python3（Flask的）代码不是很多，Python3.4和Python3.6差异不是很大，至少短期内用起来没有问题

所以打算将就着凑合用：

在服务器端，使用本地Mac中的 Python 3.6 的Pipfile，但是创建出来的 Python 3.4 的虚拟环境

具体步骤：

```
pipenv install --python 3.4
```

即可。

The script virtualenv is installed in which is not on PATH

背景:

```
pip3 install pipenv --user
```

安装后看到警告:

```
The script virtualenv is installed in '/Users/crifan/Library/Python/3.6/bin'
Consider adding this directory to PATH or, if you prefer, adjust your 'PATH' environment variable to include this folder.
The script virtualenv-clone is installed in '/Users/crifan/Library/Python/3.6/bin'
Consider adding this directory to PATH or, if you prefer, adjust your 'PATH' environment variable to include this folder.
The scripts pewtwo, pipenv and pipenv-resolver are installed in '/Users/crifan/Library/Python/3.6/bin'
Consider adding this directory to PATH or, if you prefer, adjust your 'PATH' environment variable to include this folder.
Successfully installed certifi-2018.4.16 pipenv-11.10.0 virtualenv-17.1.0
```

原因: pipenv安装后, 默认没有把对应的pipenv (以及其他相关的 pewtwo, pipenv-resolver, virtualenv, virtualenv-clone) 的所在路径

此处是: `/Users/crifan/Library/Python/3.6/bin`, 加入到PATH 中, 这会导致命令行中找不到pipenv (和另外几个)

解决办法: 按照提示, 把上述路径加到PATH中。

具体步骤:

把:

```
export PATH="/Users/crifan/Library/Python/3.6/bin:$PATH"
```

加到 (系统启动脚本) `.bashrc` 中

-> 为了以后每次启动电脑后, PATH中都包含该路径。

备注: 为了当前不重启 (终端或系统) 就使得PATH生效, 所以去:

```
source ~/.bashrc
```

然后此刻PATH中即可包含该路径。

-> 命令行中就可以找到这些 `pipenv` 等工具了

-> 后续命令行中运行 `pipenv` 就不会报错找不到了

关于如何在pipenv中运行python代码

举例:

```
pipenv run python testRestApi.py
```

crifan.com, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved,
powered by Gitbook最后更新: 2021-07-17 12:17:47

附录

下面列出相关参考资料。

crifan.com, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved,
powered by Gitbook最后更新: 2021-07-17 12:12:40

参考资料

- 管理好Python的库：包管理器
- pip · 管理好Python的库：包管理器
- 【已解决】Python3中选择合适的虚拟环境工具
- requests · Python心得：http网络库
- 【已解决】pipenv安装后警告：The script virtualenv is installed in which is not on PATH
- 【已解决】Mac本地用pipenv搭建Flask开发环境
- 【已解决】PyCharm中调试pipenv虚拟环境中文件出错：FileNotFoundError Errno 2 No such file or directory pipenv run python py
- 【基本解决】pipenv用已有pipfile去创建环境失败：Warning Python 3.6 was not found on your system
- 【已解决】Mac本地安装Python3的pipenv虚拟环境
- 【已解决】pipenv install PySpider卡在：Locking [packages] dependencies
- 【已解决】Mac中升级Python 3的pipenv的版本
- Python爬虫框架：PySpider
- 【已解决】把Python3的pipenv的Flask部署到CentOS服务器上
- 【已解决】Flask中获取REST API接口的传递进来的参数
- 【已解决】pyspider中运行result_worker出错：ModuleNotFoundError No module named mysql
- 【已解决】Python中如何操作mysql
- 【已解决】gunicorn运行Flask的app出错：gunicorn.errors.HaltServer HaltServer Worker failed to boot 3
- 【已解决】后台用Flask封装百度的语音合成功能对外提供REST API接口
- 【已解决】Mac本地用gunicorn部署启动Flask应用
- 【已解决】Flask中连接远程MongoDB数据库的gridfs并返回查询到的文件数据
- 【已解决】Mac中.bashrc中的export的PATH没生效
- 【已解决】Safari浏览器中页面出错：Origin is not allowed by Access-Control-Allow-Origin
- 【基本解决】pipenv用已有pipfile去创建环境失败：Warning Python 3.6 was not found on your system – 在路上
- 【基本解决】如何利用已有Pipfile和Pipfile.lock创建pipenv的虚拟环境
- 【已解决】Mac本地用Celery实现延时执行任务
- 【已解决】Flask中新增Celery周期任务去定期更新Azure的token
- 【已解决】把Celery+Redis集成到在线Flask中且用supervisor去管理后台服务

- [【已解决】PyCharm中如何调试pipenv的虚拟环境中的python3的文件](#)
- [【已解决】Python中两个星号**参数去传递给函数出错：SyntaxError invalid syntax](#)
- [【已解决】剧本编写系统Python返回的dict对应的一级topic没有排序](#)
-
- [【已解决】Mac本地用pipenv搭建Flask开发环境](#)
- [【已解决】PyCharm中调试pipenv虚拟环境中文件出错：FileNotFoundError Errno 2 No such file or directory pipenv run python py](#)
- [【已解决】给自动抓包工具安装Python虚拟环境](#)
- [【已解决】windows中还原恢复virtualenv的Python虚拟环境](#)
- [【已解决】mac中删除virtualenv的虚拟环境](#)
- [【已解决】给ruleResultPostProcess初始化Python虚拟环境和安装依赖的库](#)
- [【未解决】Selenium中如何打开url后忽略加载部分资源](#)
- [【已解决】Python中两个星号**参数去传递给函数出错：SyntaxError invalid syntax](#)
-
- [12. 虚拟环境和包 — Python 3.6.14 文档](#)
- [28.3. venv — 创建虚拟环境 — Python 3.6.14 文档](#)
- [Pipenv & 虚拟环境 — The Hitchhiker's Guide to Python](#)
-

crifan.com, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved,
powered by Gitbook最后更新: 2021-07-17 12:12:34