



如何开始一个开源Python项目

2018.06.30

——王斌鑫

01

项目结构

02

LICENSE

03

包管理

04

测试

05

CI

06

发布包

07

Makefile

08

文档

以 [pydu](#) 为例

pydu

pypi v0.7.0 license MIT License python 2.7, 3.5, 3.6 Linux passing Windows passing codecov 86% contributors 3

pydu (python data structures and utils) is a library of useful data structures and utils for Python 2 and 3, which collected from open source projects and created by contributors.

Installation

To install pydu, simply:

```
$ pip install pydu
```

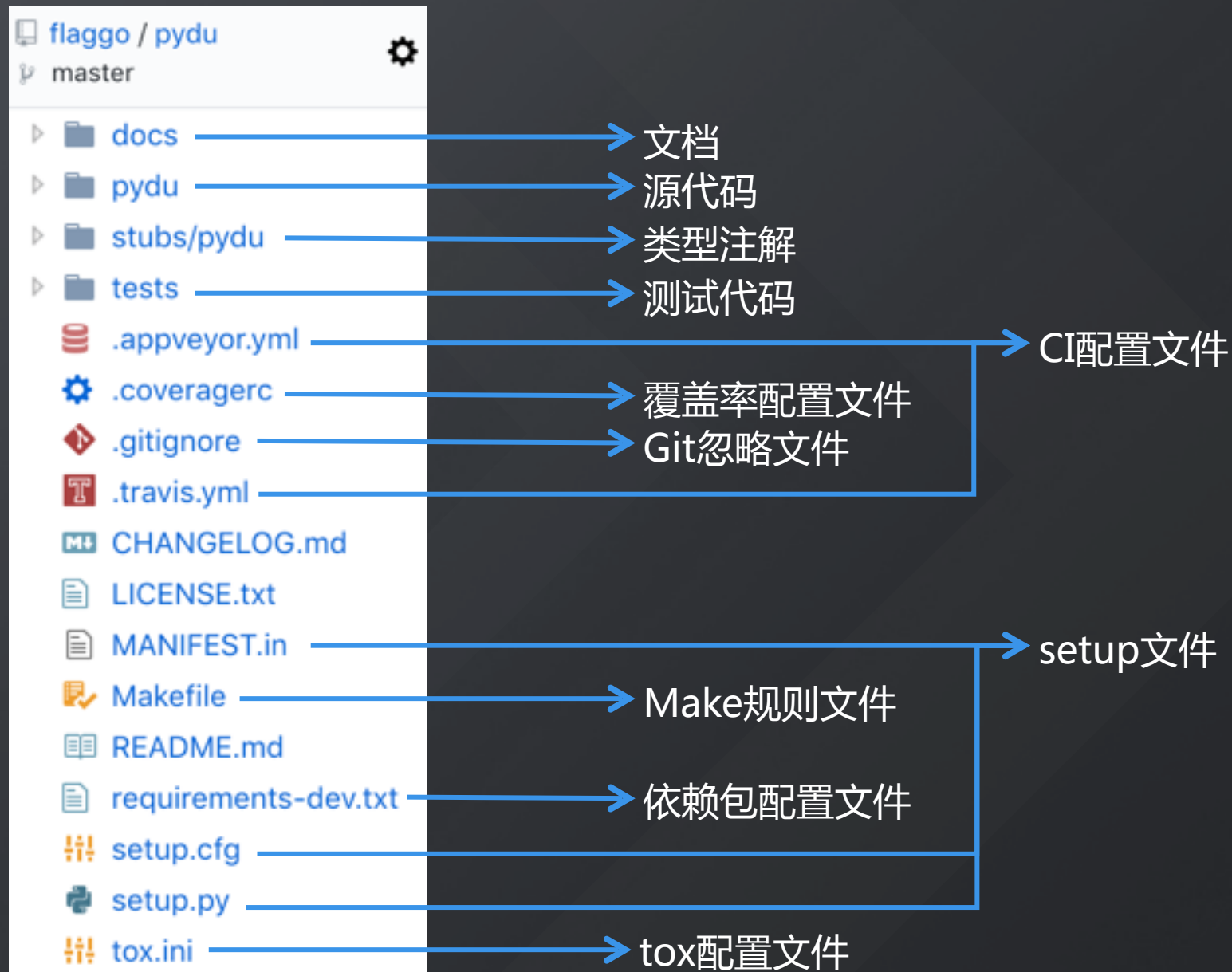
Document

Fantastic documentation is available at: [English](#) | [中文版](#).



项目结构

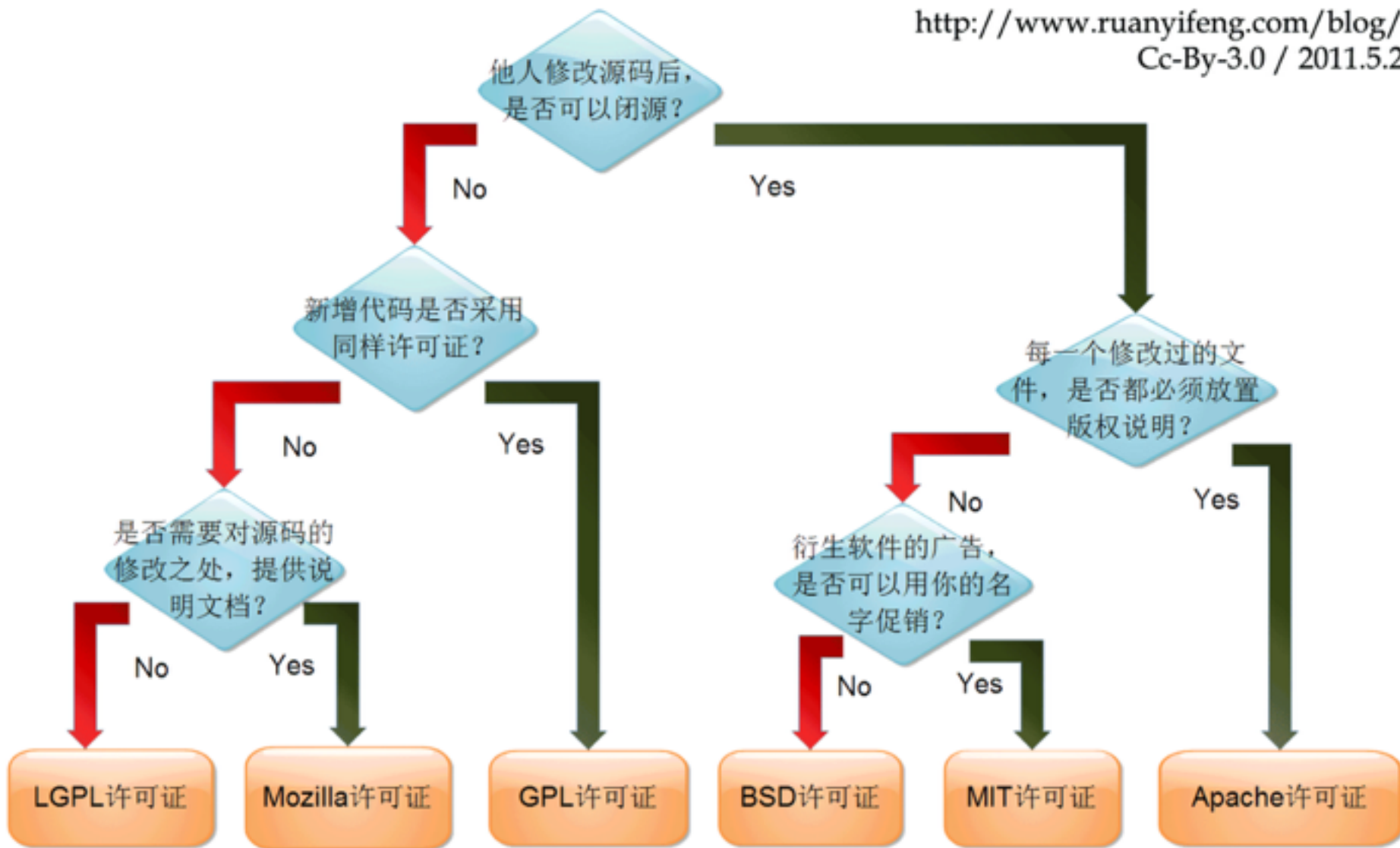
项目结构





LICENSE

<http://www.ruanyifeng.com/blog/Cc-By-3.0/>
2011.5.2





包管理

pip
+
virtualenv

```
$ virtualenv venv  
$ source venv/bin/source  
$ pip install requests  
$ deactivate
```

pipenv

```
$ pipenv install requests
```



测试

测试——测试框架



unittest

```
import unittest

class TestStringMethods(unittest.TestCase):

    def test_split(self):
        s = 'hello world'
        self.assertEqual(s.split(), ['hello', 'world'])
        with self.assertRaises(TypeError):
            s.split(2)
```



nose2

```
from nose2.tools.such import helper

def test_split():
    s = 'hello world'
    assert s.split() == ['hello', 'world']
    helper.assertRaises(TypeError, s.split, 2)
```



pytest

```
import pytest

def test_split():
    s = 'hello world'
    assert s.split() == ['hello', 'world']
    with pytest.raises(TypeError):
        s.split(2)
```

测试——测试框架

	优雅性	错误信息友好度	报告	用例自动发现	并行执行	社区活跃度
unittest	一般	一般	一般	一般	不支持	标准库，官方维护
nose2	一般	一般	较丰富	强	支持	一般
pytest	优雅	好	丰富	强	支持	活跃

测试——测试工具tox

跨解释器

支持CPython2.7, 3.4及更高,
Jython和pypy;
基于不同解释器创建多个虚拟
环境进行测试。

跨解释器

tox

与CI集成

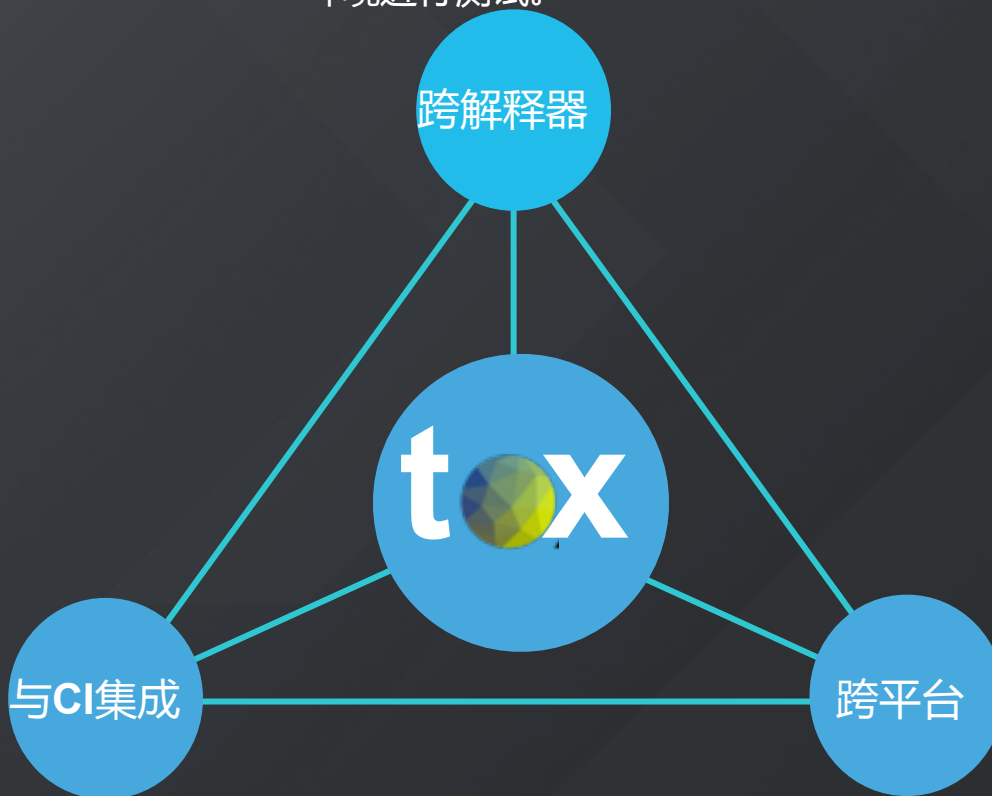
与主流CI服务器, 如Jenkins、
Travis、Appveyor完美集成。

与CI集成

跨平台

支持Windows和Unix/Linux环
境。

跨平台



测试——测试工具tox



配置文件 tox.ini

```
# content of: tox.ini, put in same dir as setup.py
[tox]
envlist = py27,py36

[testenv]
# install pytest in the virtualenv where commands will be executed
deps = pytest
# whatever extra steps before testing might be necessary
commands =
    # or any other test runner that you might use
    pytest
```



命令行执行

```
$ tox
```

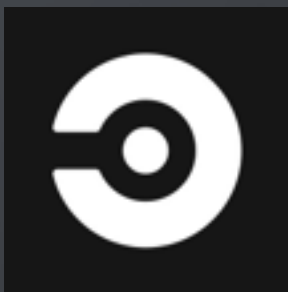


CI



Travis CI

- 快速设置
- 实时构建视图
- 支持PR构建
- 预先安装数据库服务
- 每次构建清理VM
- 支持Mac、Linux、iOS系统



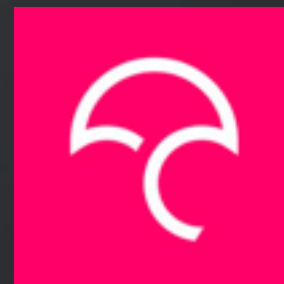
Circle CI

- workflow编排
- 全面支持Docker
- 选择所需的CPU/RAM
- 支持诸多语言，提供对应工具链、框架
- 强大的缓存
- 易于调试
- 安全性
- 高度可视化
- 支持Linux和Mac



AppVeyor

- 支持GitHub、Bitbucket、GitLab、VSTS等仓库
- 通过YAML或UI配置化构建
- 隔离、干净的构建环境
- 内置部署和NuGet服务器
- 支持分支和PR构建
- 个人支持和充满活力的社区
- 支持Windows、Linux系统



Codecov

- 一行代码上传报告
- 完美合并报告
- 无缝查看报告

CI——Travis CI



配置文件 .travis.yml

```
language: python
python:
  - "2.7"
  - "3.5"
  - "3.6"

sudo: false

cache: pip

install:
  - pip install tox codecov

script:
  - tox -e ${echo py$TRAVIS_PYTHON_VERSION | tr -d .}

after_success:
  - codecov

notifications:
  email:
    recipients:
      - wangbinxin001@126.com
    on_success: always
    on_failure: always
```



Travis CI 构建界面

flaggo / pydu build passing

Current Branches Build History Pull Requests More options

✓ v0.7.0 update changelog for v0.7.0 #299 passed Restart build

Commit 36d8579

Compare v0.7.0

Tag v0.7.0

Prodesire authored and committed

Ran for 1 min 45 sec

Total time 3 min 35 sec

2 days ago

Build Jobs

✓ # 299.1		</> Python: 2.7	no environment variables...	1 min 4 sec	
✓ # 299.2		</> Python: 3.5	no environment variables...	1 min 7 sec	
✓ # 299.3		</> Python: 3.6	no environment variables...	1 min 24 sec	

Search all repositories

CI——Appveyor



配置文件 .appveyor.yml

```
build: off

environment:
  matrix:
    - PYTHON: "C:\\Python27-x64"
      PYTHON_VERSION: "2.7.x"
      PYTHON_ARCH: "64"
      TOXENV: "py27"

    - PYTHON: "C:\\Python35-x64"
      PYTHON_VERSION: "3.5.x"
      PYTHON_ARCH: "64"
      TOXENV: "py35"

    - PYTHON: "C:\\Python36-x64"
      PYTHON_VERSION: "3.6.x"
      PYTHON_ARCH: "64"
      TOXENV: "py36"

install:
  - "SET PATH=%PYTHON%;%PYTHON%\\Scripts;%PATH%"
  - "%CMD_IN_ENV% pip install tox codecov"

test_script:
  - "%CMD_IN_ENV% tox"

on_success:
  - "%CMD_IN_ENV% codecov"
```



Appveyor 构建界面

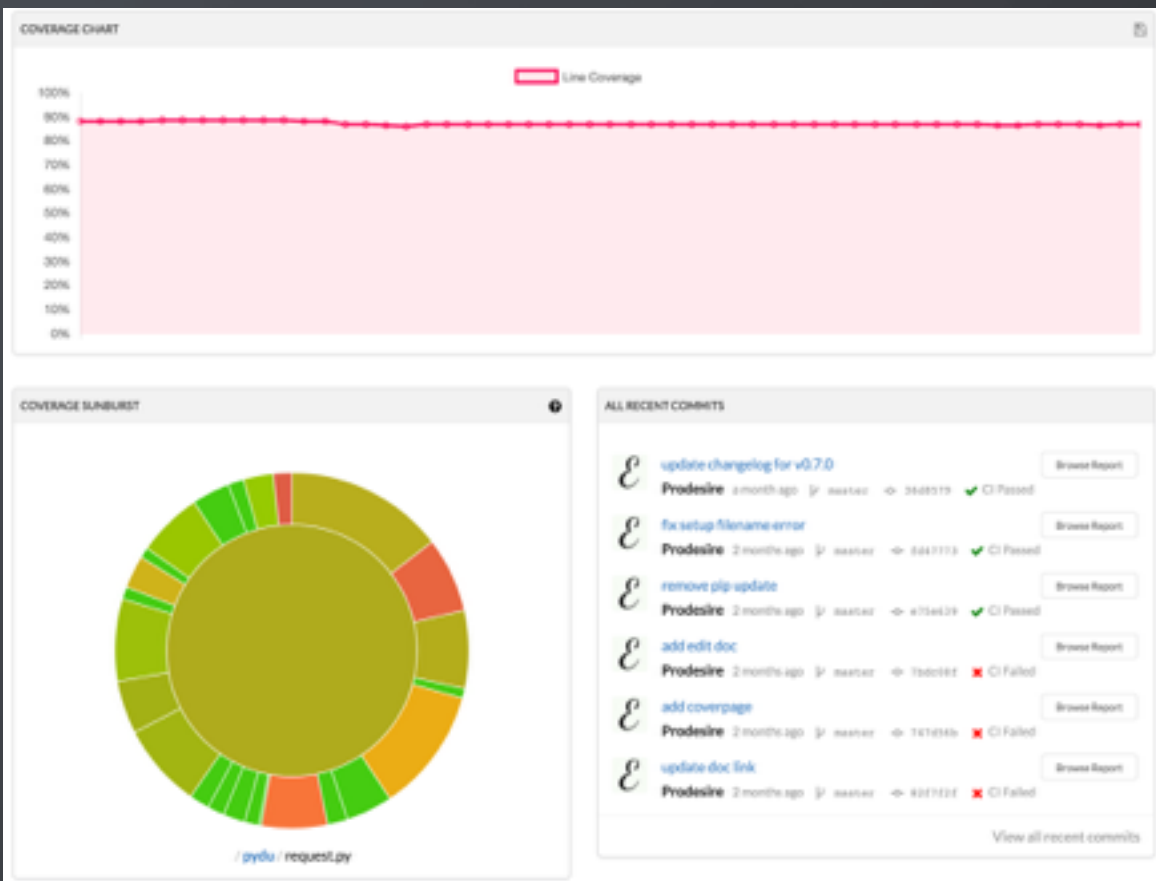
The screenshot shows the Appveyor build interface for a project named 'pydu'. The top navigation bar includes links for LATEST BUILD, HISTORY, DEPLOYMENTS, EVENTS, and SETTINGS. On the right, there are buttons for NEW BUILD and RE-BUILD COMMIT. The main section displays the latest build for the 'update changelog for v0.7.0' commit, which was pushed a month ago by Prodesire. The build is currently on the master branch, version v0.7.0, with commit hash 36d8579b. It was completed 2 days ago in 2 minutes and 59 seconds. A JOBS button is visible next to the build details. Below this, a table lists the individual build jobs.

JOB NAME	TESTS	DUR...
Environment: PYTHON=C:\Python27-x64, PYTHON_VERSION=2.7.x, PYTHON_AR...		46 sec
Environment: PYTHON=C:\Python35-x64, PYTHON_VERSION=3.5.x, PYTHON_AR...		50 sec
Environment: PYTHON=C:\Python36-x64, PYTHON_VERSION=3.6.x, PYTHON_AR...		54 sec

CI—Codecov



趋势概览



各文件覆盖率

Files					Coverage
archive.py	141	114	8	19	80.85%
cmd.py	74	69	2	3	93.24%
compat.py	53	53	0	0	100.00%
console.py	37	31	2	4	83.78%
convert.py	43	43	0	0	100.00%
dict.py	96	87	1	8	90.62%
dt.py	18	18	0	0	100.00%
environ.py	34	32	2	0	94.11%
exception.py	19	19	0	0	100.00%
functional.py	3	3	0	0	100.00%
inspect.py	75	56	1	18	74.66%
iter.py	14	14	0	0	100.00%
list.py	23	23	0	0	100.00%
misc.py	93	86	0	7	92.47%
network.py	68	76	5	7	86.36%
path.py	25	25	0	0	100.00%
platform.py	12	12	0	0	100.00%
process.py	21	0	0	21	0.00%
request.py	85	61	1	23	71.76%
set.py	23	23	0	0	100.00%



发布包

发布包



setup.py

```
38 open_kwargs = {} if PY2 else {'encoding': 'utf-8'}
39 setup(
40     name="pydu",
41     version=__version__,
42     description="Useful data structures, utils for Python.",
43     long_description=open('README.md', **open_kwargs).read(),
44     author="Prodesire",
45     author_email='wangbinxin001@126.com',
46     license='MIT License',
47     url="https://github.com/Prodesire/pydu",
48     cmdclass={'test': PyTest},
49     tests_require=test_requirements,
50     packages=find_packages(),
51     classifiers=[
52         'Operating System :: OS Independent',
53         'Intended Audience :: Developers',
54         'License :: OSI Approved :: MIT License',
55         'Programming Language :: Python',
56         'Programming Language :: Python :: Implementation',
57         'Programming Language :: Python :: 2',
58         'Programming Language :: Python :: 2.7',
59         'Programming Language :: Python :: 3',
60         'Programming Language :: Python :: 3.5',
61         'Programming Language :: Python :: 3.6',
62         'Topic :: Software Development :: Libraries'
63     ],
64 )
```



setup.cfg

```
[bdist_wheel]
universal = 1

[metadata]
license_file = LICENSE.txt
```



MANIFEST.in

```
include requirements-dev.txt README.md CHANGELOG.md LICENSE.txt
```



发布

```
$ pip install 'twine>=1.5.0'
$ python setup.py sdist
$ twine upload dist/*
$ rm -rf build dist *.egg-info .eggs
```



Makefile

Makefile

```
# Env
export PYTHONDONTWRITEBYTECODE=1
TEST_PATH=./tests
DEFAULT_PYTHON2='python -c "import sys;print(sys.version_info.major)" | grep 2'
PY2=${if ${DEFAULT_PYTHON2},python,python2}
PY3=${if ${DEFAULT_PYTHON2},python3,python}

# Func
.PHONY: docs

help:
    @echo "\033[32minit\033[0m"
    @echo "    Init environment for pydu."
    @echo "\033[32mtest\033[0m"
    @echo "    Run pytest with Python 2 and 3."
    @echo "\033[32mtest-py2\033[0m"
    @echo "    Run pytest with Python 2."
    @echo "\033[32mtest-py3\033[0m"
    @echo "    Run pytest with Python 3."
    @echo "\033[32mcoverage\033[0m"
    @echo "    Run pytest and report coverage."
    @echo "\033[32mpublish\033[0m"
    @echo "    Publish pydu to PyPI."
    @echo "\033[32mdocs\033[0m"
    @echo "    Make docs for pydu."
    @echo "\033[32mclean\033[0m"
    @echo "    Remove python and build artifacts."
    @echo "\033[32mclean-ptyc\033[0m"
    @echo "    Remove python artifacts."
    @echo "\033[32mclean-build\033[0m"
    @echo "    Remove build artifacts."

init:
    pip install -r requirements-dev.txt
```

```
test: test-py2 test-py3

test-py2: clean-ptyc
    ${PY2} -m pytest --color=yes ${TEST_PATH}

test-py3: clean-ptyc
    ${PY3} -m pytest --color=yes ${TEST_PATH}

coverage:
    coverage run --source=pydu -m pytest tests
    coverage report

publish:
    pip install 'twine>=1.5.0'
    python setup.py sdist
    twine upload dist/*
    rm -rf build dist *.egg-info .eggs

docs:
    cd docs && make html
    @echo "\033[95m\n\nBuild successful! View the docs homepage at docs/_build/html/index.html.\n\033[0m"

clean: clean-ptyc clean-build

clean-ptyc:
    find . -name '*.pyc' -exec rm -f {} +
    find . -name '*.pyo' -exec rm -f {} +
    find . -name '*~' -exec rm -f {} +
    find . -name '__pycache__' -exec rm -rf {} +

clean-build:
    rm -rf build dist *.egg-info .eggs
```



文档

项目名 ←

Badge ←

介绍 ←

安装 ←

文档 ←

pydu

pypi **v0.7.0** **license** MIT License **python** 2.7, 3.5, 3.6 **Linux** passing **Windows** passing **codecov** 86% **contributors** 3

pydu (python data structures and utils) is a library of useful data structures and utils for Python 2 and 3, which collected from open source projects and created by contributors.

Installation

To install pydu, simply:

```
$ pip install pydu
```

Document

Fantastic documentation is available at: [English](#) | [中文版](#).

文档——工具：Sphinx

pydu
latest

Search docs

Archive 存档

Cmd 命令

Compat 兼容性

Console 控制台

Convert 转换

Dict 字典

Environ 环境

Exception 异常

Functional 函数式

Inspect 检查

Iter 迭代

List 列表

Miscellanea 综合

Network 网络

Path 路径

Platform 平台

Process 进程

Request 请求

Set 集合

String 字符串

System 系统

Read the Docs

v: latest

Docs » Archive 存档

Edit on GitHub

Archive 存档

提供归档相关工具，如解压。

```
pydu.archive.extract(path, to_path="", ext="")
```

解压tar或zip文件，可指定 `to_path` 解压到特定目录。它支持很多文件格式，包括 `.tar`、`.tar.bz2`、`.tar.gz`、`.tgz`、`.taz`、`.zip`。如果给定的 `path` 不包含文件格式，则可指定 `ext` 参数来说明文件格式。

```
>>> from pydu.archive import extract
>>> extract('foobar.tgz', '/tmp')
>>> extract('foobar', '/tmp', ext='.tgz')
>>> extract('foobar', '/tmp')
Traceback (most recent call last):
...   AttributeError: pydu.archive.UnrecognizedArchiveFormat: Path not a recognized archive
```

Previous

Next

© Copyright 2017, Prodesire. Revision cbfb1a03.

Built with Sphinx using a theme provided by Read the Docs.

- reStructuredText语法
- 多种主题风格，如 readthedocs风格
- 支持国际化
- 需构建静态页面

文档——工具：docsify

搜索

pydu

模块

Archive 归档

- archive.extract

Command 命令

Compat 兼容性

Console 控制台

Convert 转换

Dict 字典

Date and Time 时间

Environment 环境

Exception 异常

Functional 函数式

Inspect 检查

Iter 迭代

List 列表

Miscellanea 综合

Network 网络

Path 路径

Platform 平台

En 中文

Edit Document

archive

提供归档相关工具。如解压。

archive.extract

```
extract(path, to_path='', ext='')
```

解压tar或zip文件。可指定 `to_path` 解压到特定目录。它支持很多文件格式，包括 "`.tar`、`.tar.bz2`、`.tar.gz`、`.tgz`、`.tz2`、`.zip`。如果给定的 "`path`" 不包含文件格式，则可指定 `ext` 参数来说明文件格式。

```
>>> from pydu.archive import extract
>>> extract('foobar.tgz', '/tmp')
>>> extract('foobar', '/tmp', ext='.tgz')
>>> extract('foobar', '/tmp')
Traceback (most recent call last):
...   AttributeError: pydu.archive.UnrecognizedArchiveFormat: Path not a recognized archive
```

pydu Copyright © 2017-2018 Prodesire. Powered By docsify.

- Markdown语法
- 多种主题风格，如vue风格
- 支持国际化
- 无需构建静态页面

谢谢
聆听



dreamlofter



<https://www.zhihu.com/people/prodesire>



<https://github.com/Prodesire>