
python-packaging Documentation

0.1

Scott Torborg

11 18, 2017

Contents

1	3
2	7
3	9
4	11
5	13
6	15
7	17
8	21

Python

:

- pip or easy_install.
- .
- .
- .
- .

python **funniest**

CHAPTER 1

:

```
def joke():  
    return (u'How do you tell HTML from HTML5?'  
            u'Try it out in Internet Explorer.'  
            u'Does it work?'  
            u'No?'  
            u'It\'s HTML5.')
```

Python,

1.1

Python :

-
- pypi
- ()

Python module **funniest**

1.2

funniest :

```
funniest/  
    funniest/  
        __init__.py  
        setup.py
```

, funniest.git. funniest, Python module.

, joke() __init__.py:

```
def joke():
    return (u'How do you tell HTML from HTML5?'
            u'Try it out in Internet Explorer.'
            u'Does it work?'
            u'No?'
            u'It\'s HTML5.')
```

setup setup.py, setuptools.setup(),:

```
from setuptools import setup

setup(name='funniest',
      version='0.1',
      description='The funniest joke in the world',
      url='http://github.com/storborg/funniest',
      author='Flying Circus',
      author_email='flyingcircus@example.com',
      license='MIT',
      packages=['funniest'],
      zip_safe=False)
```

python:

```
$ python setup.py install
```

.,.:

```
$ python setup.py develop
```

python:

```
>>> import funniest
>>> print funniest.joke()
```

1.3 PyPI

setup.py PyPI.

(pypi.python.org):

```
$ python setup.py register
```

PyPI, ..

PyPI **funniest** :

<http://pypi.python.org/pypi/funniest/0.1>

URLgit, . clonegit .

:

```
$ python setup.py sdist
```

dist/funniest-0.1.tar.gz,.,.,.

PyPI:

```
$ python setup.py sdist upload
```

```
...:
```

```
$ python setup.py register sdist upload
```

setup.py:

```
$ python setup.py --help-commands
```

1.4

```
, easy_install:
```

```
easy_install funniest
```

```
pip
```

```
$ pip install funniest
```

```
,()
```

1.5

```
,
```

```
, text,:
```

```
funniest/
  funniest/
    __init__.py
    text.py
    setup.py
```

```
__init__.py
```

```
from .text import joke
```

```
text.py
```

```
def joke():
    return (u'How do you tell HTML from HTML5?'
            u'Try it out in Internet Explorer.'
            u'Does it work?'
            u'No?'
            u'It\'s HTML5.')
```

```
funniest/funniest/.
```

1.6 (.gitignore, etc)

```
.gitignore,...
```

.gitignore:

```
# Compiled python modules.  
*.pyc  
  
# Setuptools distribution folder.  
/dist/  
  
# Python egg metadata, regenerated from source files by setuptools.  
/*.egg-info
```

1.7

. Python, .

, , , ,

CHAPTER 2

Python, PyPI

setuptools, .

funniest joke Markdown.

__init__.py

```
from markdown import markdown

def joke():
    return markdown(u'How do you tell HTML from HTML5?'
                    u'Try it out in **Internet Explorer**.'
                    u'Does it work?'
                    u'No?'
                    u'It\'s HTML5.')
```

markdown. setup.py install_requires:

```
from setuptools import setup

setup(name='funniest',
      version='0.1',
      description='The funniest joke in the world',
      url='http://github.com/storborg/funniest',
      author='Flying Circus',
      author_email='flyingcircus@example.com',
      license='MIT',
      packages=['funniest'],
      install_requires=[
          'markdown',
      ],
      zip_safe=False)
```

python setup.py develop

```
$ python setup.py develop
running develop
```

```
running egg_info
writing requirements to funniest.egg-info/requirements.txt
writing funniest.egg-info/PKG-INFO
writing top-level names to funniest.egg-info/top_level.txt
writing dependency_links to funniest.egg-info/dependency_links.txt
reading manifest file 'funniest.egg-info/SOURCES.txt'
writing manifest file 'funniest.egg-info/SOURCES.txt'
running build_ext
Creating /.../site-packages/funniest.egg-link (link to .)
funniest 0.1 is already the active version in easy-install.pth

Installed /Users/scott/local/funniest
Processing dependencies for funniest==0.1
Searching for Markdown==2.1.1
Best match: Markdown 2.1.1
Adding Markdown 2.1.1 to easy-install.pth file

Using /.../site-packages
Finished processing dependencies for funniest==0.1
```

funniest, pip install funniest markdown.

2.1 PyPI

, setuptools, PyPI, dependency_links URL, URL, setuptoolsURL.

, GithubURL:

```
setup(
    ...
    dependency_links=['http://github.com/user/repo/tarball/master#egg=package-1.0']
    ...
)
```

```
setuptools.setup() ,..:
```

```
from setuptools import setup

setup(name='funniest',
      version='0.1',
      description='The funniest joke in the world',
      long_description='Really, the funniest around.',
      classifiers=[
          'Development Status :: 3 - Alpha',
          'License :: OSI Approved :: MIT License',
          'Programming Language :: Python :: 2.7',
          'Topic :: Text Processing :: Linguistic',
      ],
      keywords='funniest joke comedy flying circus',
      url='http://github.com/storborg/funniest',
      author='Flying Circus',
      author_email='flyingcircus@example.com',
      license='MIT',
      packages=['funniest'],
      install_requires=[
          'markdown',
      ],
      include_package_data=True,
      zip_safe=False)
```

```
classifiers, http://pypi.python.org/pypi?%3Aaction=list_classifiers.
```

3.1 README / Long Description

README, PyPI `long_description`. `reStructuredText`, .

funniest, :

```
funniest/  
  funniest/  
    __init__.py  
  setup.py  
  README.rst  
  MANIFEST.in
```

README.rst

```
Funniest  
-----  
  
To use (with caution), simply do::  
  
    >>> import funniest  
    >>> print funniest.joke()
```

MANIFEST.in

```
include README.rst
```

setuptoolsREADME.rst, Python.

setup.py

```
from setuptools import setup  
  
def readme():  
    with open('README.rst') as f:  
        return f.read()  
  
setup(name='funniest',  
      version='0.1',  
      description='The funniest joke in the world',  
      long_description=readme(),  
      classifiers=[  
          'Development Status :: 3 - Alpha',  
          'License :: OSI Approved :: MIT License',  
          'Programming Language :: Python :: 2.7',  
          'Topic :: Text Processing :: Linguistic',  
      ],  
      keywords='funniest joke comedy flying circus',  
      url='http://github.com/storborg/funniest',  
      author='Flying Circus',  
      author_email='flyingcircus@example.com',  
      license='MIT',  
      packages=['funniest'],  
      install_requires=[  
          'markdown',  
      ],  
      include_package_data=True,  
      zip_safe=False)
```

GitHubBitBucket, README.rst .

CHAPTER 4

funniest. funniest...:

```
funniest/  
  funniest/  
    __init__.py  
    tests/  
      __init__.py  
      test_joke.py  
  setup.py  
  ...
```

test_joke.py..., unittest.TestCase:

```
from unittest import TestCase  
  
import funniest  
  
class TestJoke(TestCase):  
    def test_is_string(self):  
        s = funniest.joke()  
        self.assertTrue(isinstance(s, basestring))
```

Nose()

\$ pip install nose \$ nosetests

setup.py, Nose.:

```
setup(  
    ...  
    test_suite='nose.collector',  
    tests_require=['nose'],  
)
```

,:

```
$ python setup.py test
```

setuptools nose.

Python. setuptools/PyPI, Python .
, funniest-joke .
setuptools.setup() scripts console_scripts .

5.1 scripts

:

```
funniest/  
    funniest/  
        __init__.py  
        ...  
    setup.py  
    bin/  
        funniest-joke  
    ...
```

bin/funniest-joke:

```
#!/usr/bin/env python  
  
import funniest  
print funniest.joke()
```

“setup()“:

```
setup(  
    ...  
    scripts=['bin/funniest-joke'],  
    ...  
)
```

, setuptoolsPATH:

```
$ funniest-joke
```

Python, funniest-joke shell.

5.2 console_scripts

'entry point'. setuptools 'entry points', . console_scripts 'entry points'.

console_scripts Python *functions* ().

,:

```
funniest/  
  funniest/  
    __init__.py  
    command_line.py  
    ...  
  setup.py  
  ...
```

command_line.py ():

```
import funniest  
  
def main():  
    print funniest.joke()
```

,:

```
$ python  
>>> import funniest.command_line  
>>> funniest.command_line.main()  
...
```

setup.py main()

```
setup(  
    ...  
    entry_points = {  
        'console_scripts': ['funniest-joke=funniest.command_line:main'],  
    }  
    ...  
)
```

, funniest-joke.setuptools, , ,

CHAPTER 6

python, . . . setuptools, .

MANIFEST.in, MANIFEST.in, .:

```
include README.rst
include docs/*.txt
include funniest/data.json
```

site-packages, setup() include_package_data=True.

: ()Python(funniest/funniest/data.json)., __file__.

CHAPTER 7

Python, :

```
funniest/  
  funniest/  
    __init__.py  
    command_line.py  
    tests/  
      __init__.py  
      test_joke.py  
      test_command_line.py  
  MANIFEST.in  
  README.rst  
  setup.py  
  .gitignore
```

:

funniest/__init__.py

```
from markdown import markdown  
  
def joke():  
    return markdown(u'How do you tell HTML from HTML5?'  
                    u'Try it out in **Internet Explorer**.'  
                    u'Does it work?'  
                    u'No?'  
                    u'It\'s HTML5.')
```

funniest/command_line.py

```
from . import joke  
  
def main():  
    print joke()
```

funniest/tests/__init__.py

(empty)

funniest/tests/test_joke.py

```
from unittest import TestCase

import funniest

class TestJoke(TestCase):
    def test_is_string(self):
        s = funniest.joke()
        self.assertTrue(isinstance(s, basestring))
```

funniest/tests/test_command_line.py

```
from unittest import TestCase

from funniest.cmd import main

class TestCmd(TestCase):
    def test_basic(self):
        main()
```

MANIFEST.in

```
include README.rst
```

README.rst

```
Funniest
-----

To use (with caution), simply do::

    >>> import funniest
    >>> print funniest.joke()
```

setup.py

```
from setuptools import setup

def readme():
    with open('README.rst.example') as f:
        return f.read()

setup(name='funniest',
      version='0.1',
      description='The funniest joke in the world',
      long_description=readme(),
      classifiers=[
          'Development Status :: 3 - Alpha',
          'License :: OSI Approved :: MIT License',
          'Programming Language :: Python :: 2.7',
          'Topic :: Text Processing :: Linguistic',
      ],
      keywords='funniest joke comedy flying circus',
```

```
url='http://github.com/storborg/funniest',
author='Flying Circus',
author_email='flyingcircus@example.com',
license='MIT',
packages=['funniest'],
install_requires=[
    'markdown',
],
test_suite='nose.collector',
tests_require=['nose', 'nose-cover3'],
entry_points={
    'console_scripts': ['funniest-joke=funniest.command_line:main'],
},
include_package_data=True,
zip_safe=False)
```

.gitignore

```
# Compiled python modules.
*.pyc

# Setuptools distribution folder.
/dist/

# Python egg metadata, regenerated from source files by setuptools.
/*.egg-info
/*.egg
```


CHAPTER 8

Scott Torborg - storborg@gmail.com.

I wrote this tutorial in an attempt to improve the state of Python packages at large. Tools like [virtualenv](#) and [pip](#), as well as improvements to [setuptools](#), have made the Python package ecosystem a delight to work in.

However, I frequently run across packages I want to use that don't interoperate correctly with others, or find myself in situations where I'm not sure exactly how to structure things. This is an attempt to fix that.

This documentation is written in reStructuredText and built with [Sphinx](#). The source is open and hosted at <http://github.com/storborg/python-packaging>.

To build the HTML version, just do this in a clone of the repo:

```
$ make html
```

Contributions and fixes are welcome, encouraged, and credited. Please submit a pull request on GitHub or email me.
[openmartin](#) [Github](#)

http://python-packaging-zh.readthedocs.org/zh_CN/latest/index.html

: Python 2.xPython 3.x

:

Setuptools Documentation [setuptools documentation](#).

Python Packaging User Guide “Python Packaging User Guide” (PyPUG) Python.