

SHARE YOUR CODE!

PYTHON PACKAGING WITHOUT COMPLICATION

daveops.com/pycon2017
dave@forgac.com
[@tylerdave](#)

```
def beautiful_code():  
    """This code is so beautiful."""  
    print('Hello, PyCon!')
```

*“I want to share my code!
What do I need to do?”*

PACKAGE

DOCUMENTATION

SOURCE HOSTING

TESTS

CI

LICENSE

CONTRIBUTING

WE NEED

Code

Package

Documentation

Source Hosting

Tests

CI

License

Contributing

*“But I just wanted to share
my code!”*

TL;DR

TL;DR



TERMINOLOGY

MODULE

SAVED PYTHON CODE

IMPORT PACKAGE

NAMESPACE / DIRECTORY

DISTRIBUTION PACKAGE

SHAREABLE / INSTALLABLE

SOURCE DISTRIBUTION

SDIST

BUILT DISTRIBUTION

WHEEL

WHEEL

WHEEL

UNIVERSAL

WHEEL

UNIVERSAL

PURE PYTHON

WHEEL

UNIVERSAL

PURE PYTHON

PLATFORM

HISTORY

HISTORY

COPY / REPLACE

HISTORY

COPY / REPLACE

PYPA

HISTORY

COPY / REPLACE

PYPA

PYPUG

<https://packaging.python.org/>

“Let's make a package!”

USING COOKIECUTTER

```
$ cookiecutter cookiecutter-python-package
...
full_name (default is "Dave Forgac")?
email (default is "tylerdave@tylerdave.com")?
github_username (default is "tylerdave")?
project_name (default is "Example")? Example Package
repo_name (default is "boilerplate")? Example-Package
project_short_description (default is "A short description")?
Example package for PyCon talk.
release_date (default is "2017-05-21")?
year (default is "2017")?
version (default is "0.1.0")?
```

LAYOUT

```
.
├── data
│   └── data_file
├── DESCRIPTION.rst
├── MANIFEST.in
├── README.rst
├── sample
│   ├── __init__.py
│   └── package_data.dat
├── setup.cfg
├── setup.py
└── tests
    ├── __init__.py
    └── test_sample.py
```

CODE

```
.  
├── data  
│   └── data_file  
├── DESCRIPTION.rst  
├── MANIFEST.in  
├── README.rst  
├── sample  
│   ├── __init__.py  
│   └── package_data.dat  
├── setup.cfg  
├── setup.py  
└── tests  
    ├── __init__.py  
    └── test_sample.py
```


SETUP.PY

```
.
├── data
│   └── data_file
├── DESCRIPTION.rst
├── MANIFEST.in
├── README.rst
├── sample
│   ├── __init__.py
│   └── package_data.dat
├── setup.cfg
├── setup.py
└── tests
    ├── __init__.py
    └── test_sample.py
```

SETUP.CFG

```
.
├── data
│   └── data_file
├── DESCRIPTION.rst
├── MANIFEST.in
├── README.rst
├── sample
│   ├── __init__.py
│   └── package_data.dat
├── setup.cfg
├── setup.py
└── tests
    ├── __init__.py
    └── test_sample.py
```

MANIFEST.IN

```
.
├── data
│   └── data_file
├── DESCRIPTION.rst
├── MANIFEST.in
├── README.rst
├── sample
│   ├── __init__.py
│   └── package_data.dat
├── setup.cfg
├── setup.py
└── tests
    ├── __init__.py
    └── test_sample.py
```

README.RST

```
.
├── data
│   └── data_file
├── DESCRIPTION.rst
├── MANIFEST.in
├── README.rst
├── sample
│   ├── __init__.py
│   └── package_data.dat
├── setup.cfg
├── setup.py
└── tests
    ├── __init__.py
    └── test_sample.py
```

SETUP.PY (1/2)

```
from setuptools import setup, find_packages
```

```
setup(
```

```
    name='Example Package',
    version='0.1.0',      # PEP440-compliant version
    description='Example package for PyCon talk.',
    long_description='Displayed on PyPI project page.',
    url='https://github.com/tylerdave/Example-Package',
    author='Dave Forgac',
    author_email='tylerdave@tylerdave.com',
    packages=find_packages(exclude=['docs', 'tests'])
    install_requires=['requests'],
    package_data={
        'sample': ['package_data.dat']
    }
```

SETUP.PY (1/2)

```
from setuptools import setup, find_packages
```

```
setup(
```

```
    name='Example Package',  
    version='0.1.0',      # PEP440-compliant version  
    description='Example package for PyCon talk.',  
    long_description='Displayed on PyPI project page.',  
    url='https://github.com/tylerdave/Example-Package',  
    author='Dave Forgac',  
    author_email='tylerdave@tylerdave.com',  
    packages=find_packages(exclude=['docs', 'tests'])  
    install_requires=['requests'],  
    package_data={  
        'sample': ['package_data.dat']  
    }
```

SETUP.PY (1/2)

```
from setuptools import setup, find_packages

setup(
    name='Example Package',
    version='0.1.0',    # PEP440-compliant version
    description='Example package for PyCon talk.',
    long_description='Displayed on PyPI project page.',
    url='https://github.com/tylerdave/Example-Package',
    author='Dave Forgac',
    author_email='tylerdave@tylerdave.com',
    packages=find_packages(exclude=['docs', 'tests'])
    install_requires=['requests'],
    package_data={
        'sample': ['package_data.dat']
    }
)
```

SETUP.PY (1/2)

```
from setuptools import setup, find_packages

setup(
    name='Example Package',
    version='0.1.0',    # PEP440-compliant version
    description='Example package for PyCon talk.',
    long_description='Displayed on PyPI project page.',
    url='https://github.com/tylerdave/Example-Package',
    author='Dave Forgac',
    author_email='tylerdave@tylerdave.com',
    packages=find_packages(exclude=['docs', 'tests'])
    install_requires=['requests'],
    package_data={
        'sample': ['package_data.dat']
    }
)
```


SETUP.PY (1/2)

```
from setuptools import setup, find_packages

setup(
    name='Example Package',
    version='0.1.0',    # PEP440-compliant version
    description='Example package for PyCon talk.',
    long_description='Displayed on PyPI project page.',
    url='https://github.com/tylerdave/Example-Package',
    author='Dave Forgac',
    author_email='tylerdave@tylerdave.com',
    packages=find_packages(exclude=['docs', 'tests'])
    install_requires=['requests'],
    package_data={
        'sample': ['package_data.dat']
    }
```

SETUP.PY (2/2)

...

```
entry_points={  
    'console_scripts': [  
        'hello=example_package.cli:say_hello',  
    ],  
}
```

```
license='MIT',
```

```
classifiers=[
```

```
    'Development Status :: 3 - Alpha',
```

```
    'License :: OSI Approved :: MIT License',
```

```
    'Programming Language :: Python :: 2',
```

```
    'Programming Language :: Python :: 2.7',
```

```
    'Programming Language :: Python :: 3',
```

```
    'Programming Language :: Python :: 3.4',
```

```
    'Programming Language :: Python :: 3.5',
```

```
    'Programming Language :: Python :: 3.6'
```

SETUP.PY (2/2)

```
...
entry_points={
    'console_scripts': [
        'hello=example_package.cli:say_hello',
    ],
}
license='MIT',
classifiers=[
    'Development Status :: 3 - Alpha',
    'License :: OSI Approved :: MIT License',
    'Programming Language :: Python :: 2',
    'Programming Language :: Python :: 2.7',
    'Programming Language :: Python :: 3',
    'Programming Language :: Python :: 3.4',
    'Programming Language :: Python :: 3.5',
    'Programming Language :: Python :: 3.6'
```

SETUP.PY (2/2)

```
...
entry_points={
    'console_scripts': [
        'hello=example_package.cli:say_hello',
    ],
}
license='MIT',
classifiers=[
    'Development Status :: 3 - Alpha',
    'License :: OSI Approved :: MIT License',
    'Programming Language :: Python :: 2',
    'Programming Language :: Python :: 2.7',
    'Programming Language :: Python :: 3',
    'Programming Language :: Python :: 3.4',
    'Programming Language :: Python :: 3.5',
    'Programming Language :: Python :: 3.6'
```

SETUP.CFG

```
[bdist_wheel]  
universal = 1
```

```
[flake8]  
exclude = docs
```

MANIFEST.IN

```
include LICENSE  
include README.rst  
  
recursive-include tests *  
recursive-exclude * __pycache__  
recursive-exclude * *.py[co]
```

README.RST

```
=====
Example Package
=====

License: MIT

Documentation: https://example.com

Usage
-----

``sample --help``
```

LICENSE

TESTS

./tests

tox.ini

Docs

`./docs`

CI

`.travis.yml`

REQUIREMENTS.TXT

Also Common

(RST | MD | TXT)

HISTORY or CHANGES or CHANGELOG
CONTRIBUTING
AUTHORS

“Now let's share our code!”

GIT INIT

```
cd Example-Package  
git init  
git add .  
git commit -m 'initial commit'
```

RUN TESTS

tox

...

OK

	summary
py27: commands succeeded	
py34: commands succeeded	
py35: commands succeeded	
py36: commands succeeded	

COMMIT

```
git add .  
git commit -m "add hello pycon functionality"
```

SERVICES

GITHUB

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner





Repository name



Great repository names are short and memorable. Need inspiration? How about **stunning-octo-goggles**.

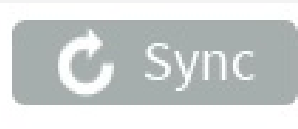
Description (optional)

- ☒  **Public**
Anyone can see this repository. You choose who can commit.
- ☐  **Private**
You choose who can see and commit to this repository.

☐ Initialize this repository with a README

TRAVIS CI

<https://travis-ci.org/>



or

travis enable

GIT PUSH

```
To git@github.com:tylerdave/Example-Package.git
* [new branch]      master -> master
Branch master set up to track remote branch master
from origin.
```

SEE BUILDS SUCCEEDING!

Current

Branches


Build History

Pull Requests

✓

🔗

master add tests

 Dave Forgac authored and committed

1 passed

🐱 Commit ae34900

🐱 Compare 304379b..ae34900

🕒 ran for 38 sec

Build Jobs

✓	# 1.1	🐧 </> Python: 2.6	📦 no environment variables set	🕒 9 sec
✓	# 1.2	🐧 </> Python: 2.7	📦 no environment variables set	🕒 7 sec
✓	# 1.3	🐧 </> Python: 3.3	📦 no environment variables set	🕒 7 sec
✓	# 1.4	🐧 </> Python: 3.4	📦 no environment variables set	🕒 7 sec
✓	# 1.5	🐧 </> Python: pypy	📦 no environment variables set	🕒 8 sec

(55/68)



Read the Docs

PYPI

<https://pypi.python.org/>

PYPI

<https://pypi.python.org/>

TEST PYPI

<https://testpypi.python.org/>

SAVE PYPI SETTINGS

```
$HOME/.pypirc
```

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

VIRTUALENV

PEW

VIRTUALENVWRAPPER

PACKAGING TOOLS

```
pip install wheel  
pip install twine
```

BUILD DIST. FILES

```
./setup.py sdist
```

```
./setup.py bdist_wheel
```

or

```
make dist
```

UPLOAD

twine upload dist/*

INSTALL

```
pip install example-package
```

DEVELOP MODE

```
./setup.py develop
```

or

```
pip install -e .
```


RINSE, REPEAT

VERSIONS

BUMPVERSION

VERSIONEER

RECAP

RECAP

~~DIY~~

RECAP

~~DIY~~

USE THE TOOLS

RECAP

~~DIY~~

USE THE TOOLS

SHARE YOUR CODE!

THANK YOU!
PLEASE LEAVE FEEDBACK

daveops.com/pycon2017

dave@forgac.com

[@tylerdave](#)