Publishing (**P**erfect) **P**ython **P**ackages on **P**yPI

Mark Smith
Nexmo

Publishing (**P**erfect) **P**ython **P**ackages on **P**yPI

Mark Smith
Nexmo

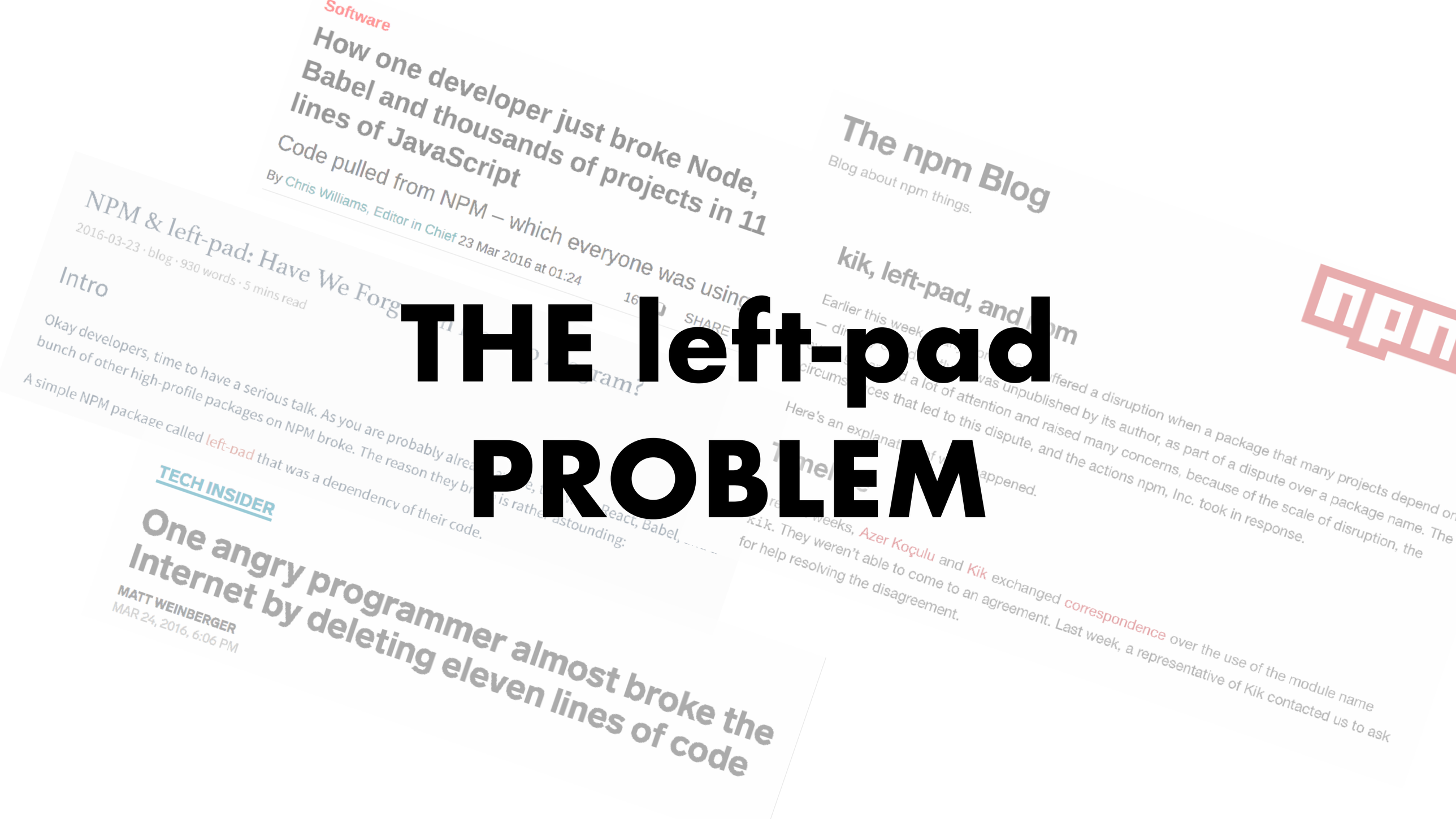
@Judy2k

Mark Smith
Developer Advocate
Nexmo

nexmo®

The Vonage® API Platform

Tap Takeover!



THE left-pad PROBLEM

@Judy2k

THE left-pad **PROBLEM**

left-pad was
the **SOLUTION**

COPY & PASTE

IS NOT

HOW YOU SHOULD

SHARE CODE

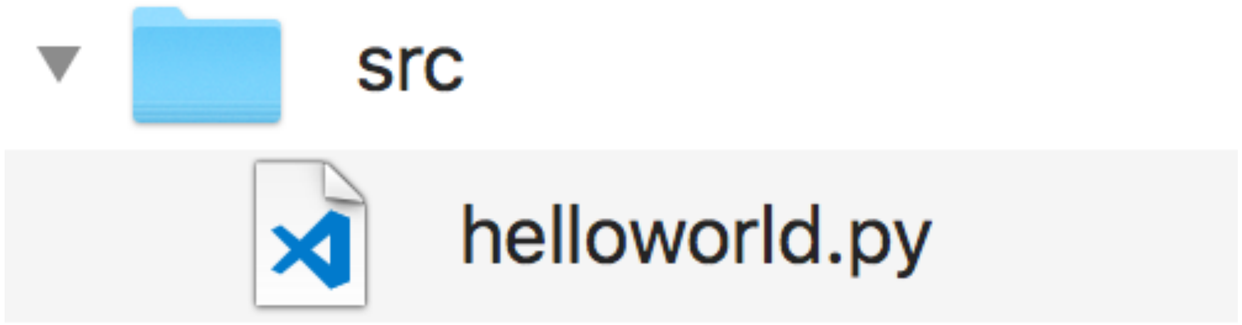
PUBLISH
YOUR CODE
MAKE PYTHON
BETTER

Make A **Package**

helloworld.py

```
def say_hello(name=None):
    if name is None:
        return "Hello, World!"
    else:
        return f"Hello, {name}!"
```

helloworld



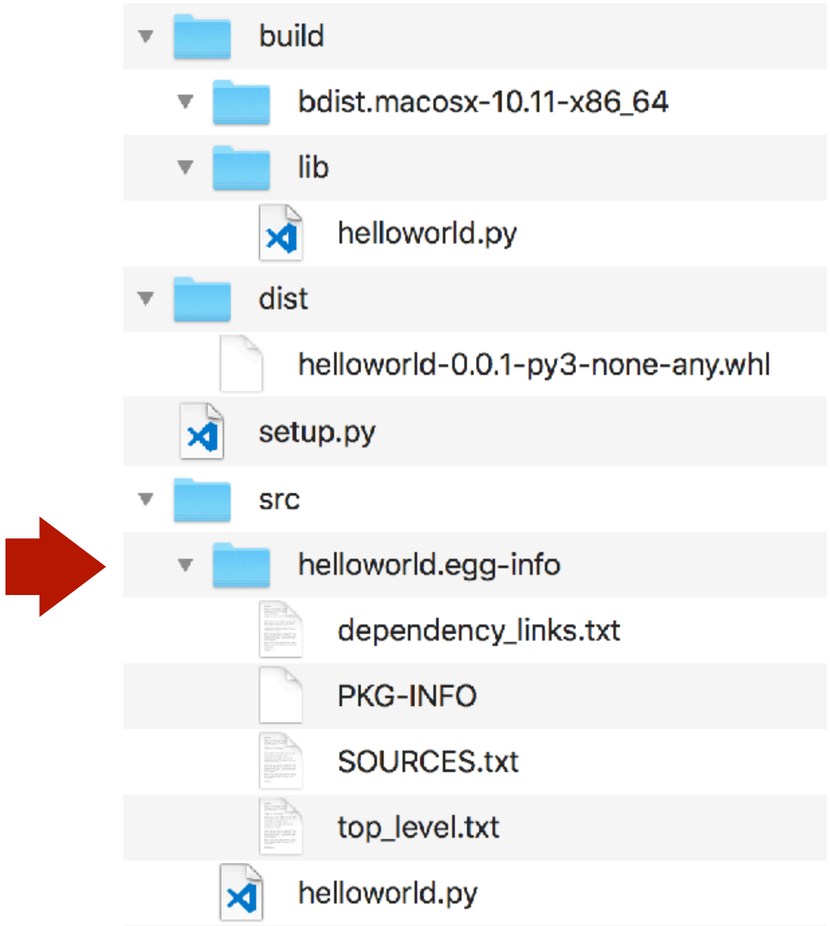
setup.py

```
from setuptools import setup

setup(
    name='helloworld',
    version='0.0.1',
    description='Say hello!',
    py_modules=["helloworld"],
    package_dir={'': 'src'},
)
```

Let's Test It!

```
$ python setup.py bdist_wheel
running bdist_wheel
…
copying src/helloworld.py -> build/lib
…
creating ‘/path/to/helloworld/dist/
helloworld-0.0.1-py3-none-any.whl’ and adding
'.' to it
removing build/bdist.macosx-10.11-x86_64/wheel
```


build
bdist.macosx-10.11-x86_64
lib
helloworld.py
dist
helloworld-0.0.1-py3-none-any.whl
setup.py
src
helloworld.egg-info
dependency_links.txt
PKG-INFO
SOURCES.txt
top_level.txt
helloworld.py

@judy2k

Install it locally

```
$ pip install -e .
Obtaining file:///path/to/helloworld
Installing collected packages: helloworld
  Running setup.py develop for helloworld
Successfully installed helloworld
```

Let's Test It!

```
$ python

>>> from helloworld import say_hello

>>> say_hello()
'Hello, World!'

>>> say_hello("Everybody")
'Hello, Everybody!'
```

Perfect It

gitignore.io

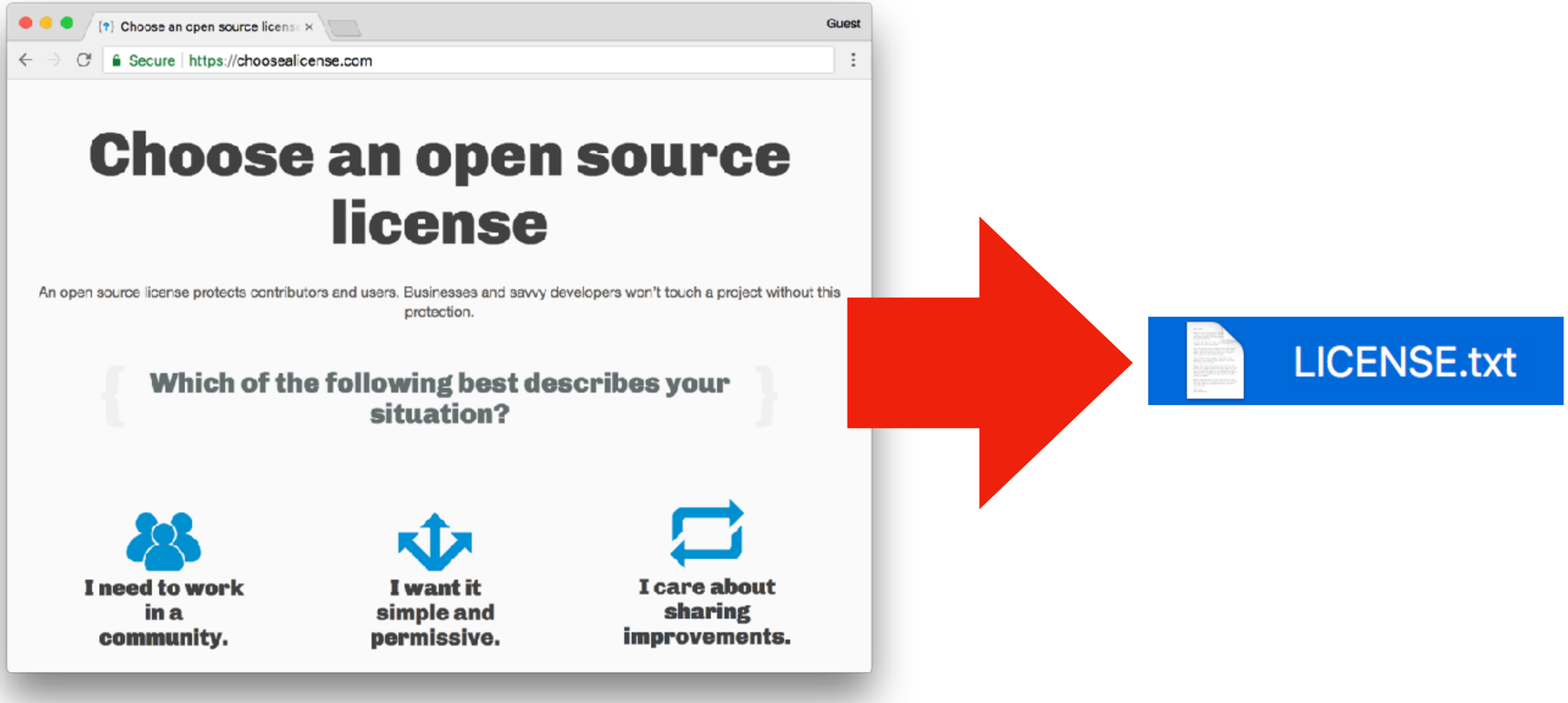
Create useful .gitignore files for your project

Search Operating Systems, IDEs, or Programming Languages

Create

Source Code | Command Line Docs | Watch Video Tutorial

choosealicense.com



setup.py

```
setup(
    ...
    classifiers=[
        "Programming Language :: Python :: 3",
        "Programming Language :: Python :: 3.6",
        "Programming Language :: Python :: 3.7",
        "License :: OSI Approved :: GNU General Public
License v2 or later (GPLv2+)",
        "Operating System :: OS Independent",
    ],
)
```

https://pypi.org/classifiers/

@judy2k

ReStructured Text

- Pythonic
- Powerful
- Can use Sphinx

Markdown

- More Widespread
- Simpler
- Can use MkDocs

README.md

```
# Hello World

This is an example project demonstrating how to publish a
python module to PyPI.
```

README.md

…

Installation

Run the following to install:

```python
pip install helloworld
```

README.md

````
…
## Usage

```python
from helloworld import say_hello

# Generate "Hello, World!"
say_hello()

# Generate "Hello, Everybody!"
say_hello("Everybody")
```
````

@judy2k

Add to **setup.py**

```
from setuptools import setup

with open("README.md", "r") as fh:
    long_description = fh.read()

setup(
    …
    long_description=long_description,
    long_description_content_type="text/markdown",
)
```

Library Dependencies

```
$ pip install blessings

# setup.py
setup(
    ...
    install_requires = [
        "blessings ~= 1.7",
    ],
)
```

Library Dependencies

```
# setup.py
setup(
    ...
    install_requires = [
        "blessings ~= 1.7",
    ],
)
```

Library Dependencies

```
$ pip install -e .
```

Test with pytest?

First let's declare dev dependencies

Development Dependencies

```
# setup.py
setup(
    ...
    extras_require = {
        "dev": [
            "pytest>=3.7",
        ],
    },
)
```

Update The Readme!

```
# Developing Hello World

To install helloworld, along with the tools you
need to develop and run tests, run the following
in your virtualenv:

```bash
$ pip install -e .[dev]
```
```

Update The Readme!

```
$ pip install -e .[dev]


...
Collecting chardet<3.1.0,>=3.0.2 (from requests!
=2.15,!=2.16,>=2.5.0->twine->helloworld-
judy2k==0.0.1)
...
Successfully installed Pygments-2.4.2
atomicwrites-1.3.0 attrs-19.1.0 ... helloworld-
judy2k idna-2.8 ... urllib3-1.25.3 wcwidth-0.1.7
webencodings-0.5.1 zipp-0.5.1
```

Install vs Extras

- **install_requires**
  - is for production dependencies (Flask, Click, Numpy, Pandas)
  - versions should be as relaxed as possible (>3.0,<4.0)
- **extras_require**
  - is for optional requirements: (Pytest, Mock, Coverage.py)
  - versions should be as specific as possible

requirements.txt

- is for **Apps** being deployed on machines **you control**.
- Uses **fixed** version numbers, eg: **requests==2.22.0**
- Is generated with **pip freeze > requirements.txt**

test_helloworld.py

```
from helloworld import say_hello


def test_helloworld_no_params():
    assert say_hello() == "Hello, World!"

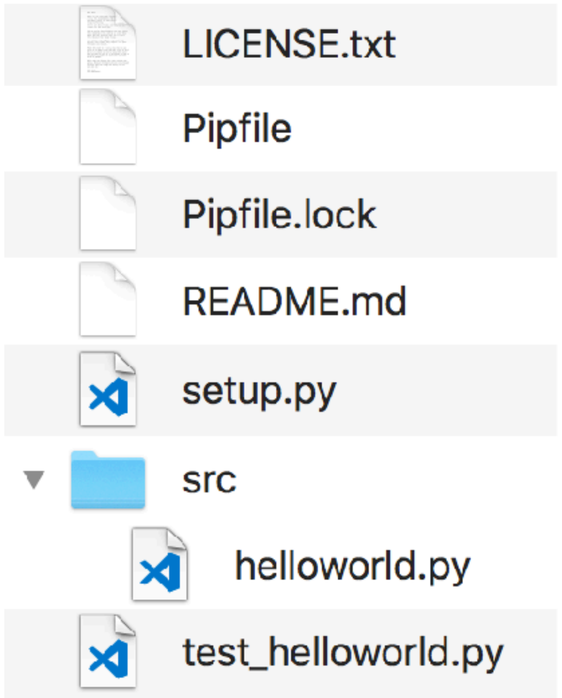

def test_helloworld_with_param():
    assert say_hello("Everyone") == "Hello, Everyone!"
```

Running tests

```
$ pytest
============== test session starts ==============
platform darwin -- Python 3.6.3, pytest-3.7.2,
py-1.5.4, pluggy-0.7.1
rootdir: /path/to/helloworld, inifile:
collected 2 items

test_helloworld.py ..                          [100%]
============ 2 passed in 0.02 seconds ===========
```

helloworld


LICENSE.txt
Pipfile
Pipfile.lock
README.md
setup.py
src
helloworld.py
test_helloworld.py

Source Distribution

```
$ python setup.py sdist
running sdist
...
warning: check: missing required meta-data: url
warning: check: missing meta-data: either (author and
author_email) or (maintainer and maintainer_email) must be
supplied
...
Creating tar archive
removing 'helloworld-0.0.1' (and everything under it)
```

setup.py

```
setup(
    ...
    url="https://github.com/judy2k/helloworld",
    author="Mark Smith",
    author_email="mark.smith@vonage.com",
)
```

Test It?

```
$ tar tzf dist/helloworld-0.0.1.tar.gz
helloworld-0.0.1/
helloworld-0.0.1/PKG-INFO
helloworld-0.0.1/README.md
helloworld-0.0.1/setup.py
helloworld-0.0.1/setup.cfg
helloworld-0.0.1/src/
helloworld-0.0.1/src/helloworld.egg-info/
helloworld-0.0.1/src/helloworld.egg-info/PKG-INFO
helloworld-0.0.1/src/helloworld.egg-info/SOURCES.txt
helloworld-0.0.1/src/helloworld.egg-info/top_level.txt
helloworld-0.0.1/src/helloworld.egg-info/dependency_links.txt
helloworld-0.0.1/src/helloworld.py
```

LICENSE.txt?
test_helloworld.py

Check Manifest

```
$ pip install check-manifest

$ check-manifest --create

$ git add MANIFEST.in
```

Test It?

```
helloworld-0.0.1/LICENSE.txt
helloworld-0.0.1/MANIFEST.in
helloworld-0.0.1/PKG-INFO
✅ helloworld-0.0.1/README.md
helloworld-0.0.1/setup.cfg
helloworld-0.0.1/setup.py
helloworld-0.0.1/src/
helloworld-0.0.1/src/helloworld.egg-info/
helloworld-0.0.1/src/helloworld.egg-info/PKG-INFO
helloworld-0.0.1/src/helloworld.egg-info/SOURCES.txt
helloworld-0.0.1/src/helloworld.egg-info/dependency_links.txt
helloworld-0.0.1/src/helloworld.egg-info/top_level.txt
helloworld-0.0.1/src/helloworld.py
✅ helloworld-0.0.1/test_helloworld.py
```

Publish It!

Build It

```
$ python setup.py bdist_wheel sdist

$ ls dist/
helloworld-0.0.1-py3-none-any.whl
helloworld-0.0.1.tar.gz
```

Push To PyPI

```
# Stick this in `extras_require`
$ pip install twine

$ twine upload dist/*
username: USER
password:
Uploading distributions to https://
upload.pypi.org/legacy/
Uploading helloworld-0.0.1-py3-none-any.whl
Uploading helloworld-0.0.1.tar.gz
```

helloworld-judy2k 0.0.1

```
pip install helloworld-judy2k
```

Latest version

Last released: Yesterday

Say hello!

Navigation

- Project description
- Release history
- Download files

Project links

Project description

Hello World

This is an example project demonstrating how to publish a python module to PyPI.

Installation

Run the following to install:

Productionize? It

tox.ini

```
[tox]
envlist = py36,py37

[testenv]
deps = pytest
commands = pytest
```

Testing With Tox

```
$ pip install tox
$ tox
...
================= test session starts ================
platform darwin -- Python 3.6.3, pytest-3.7.2, py-1.5.4,
pluggy-0.7.1
rootdir: /path/to/helloworld, inifile:
collected 2 items

test_helloworld.py ..
[100%]

============== 2 passed in 0.02 seconds ==============
...
```

@judy2k

Testing With Tox

```
...
_____________________________ summary _____________________________
  py36: commands succeeded
  py37: commands succeeded
  congratulations :)
```

Testing With Tox

```
============= test session starts =============
platform darwin -- Python 3.6.3, pytest-3.7.2,
py-1.5.4, pluggy-0.7.1
rootdir: /path/to/helloworld, inifile:
collected 2 items

test_helloworld.py ..
[100%]

=========== 2 passed in 0.02 seconds ===========
```

.travis.yml

```
language: python
sudo: false

python:
  - "3.6"
  - "3.7-dev"

install:
  - pip install tox

script:
  - tox -v -e py
```

Extra Credit

- Badges!
  - **Code Coverage** (Coveralls, codecov.io)
  - **Quality Metrics** (Code Climate, Landscape.io)
- Manage versioning with **bumpversion**
- **Test** on OSX & Windows
- **More Documentation**
  - Contributors Section
  - Code of Conduct

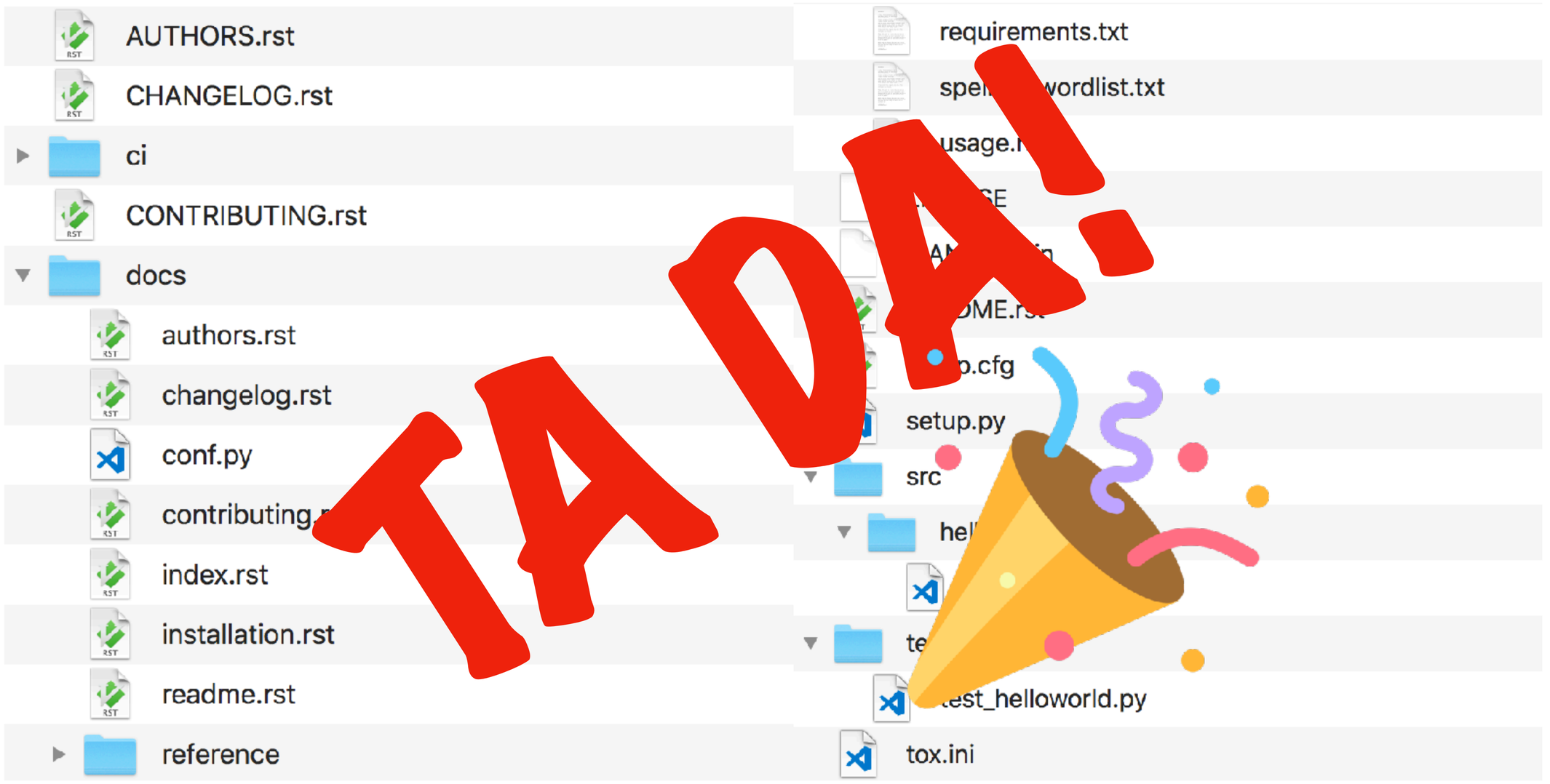
Don't **Do** This!

@judy2k

Use Cookiecutter

```
$ pip install cookiecutter

$ cookiecutter gh:ionelmc/cookiecutter-pylibrary
... Lots of questions ...
... Copy in your code and tests ...
... Some minor file tweaks ...
... DONE!
```


AUTHORS.rst
CHANGELOG.rst
ci
CONTRIBUTING.rst
docs
authors.rst
changelog.rst
conf.py
index.rst
installation.rst
readme.rst
reference
requirements.txt
setup.py
src
tox.ini
TA DA!

@judy2k

Things Are Changing...

- Move metadata from **setup.py** to **setup.cfg**
- Move to **pyproject.toml**
  - **Poetry** (cookiecutter & virtualenv & setup.py)
  - **Flit** (setup.py)
  - **Hatch** (cookiecutter & virtualenv & twine)

Slides & Code:
bit.ly/perfectpypi

Follow Me On Twitter: **@Judy2k**!