python

Erik Swanson

@swans_one
github.com/swans-one

What is a Python Package

How Hard Is Packaging?

The “Hard” Way

The Easy Way

Private/Internal Packaging

Bonus Topics

What is a Python Package?

Individually distributable libraries and utilities

```
$ pip install requests
$ python
>>> from requests import get
```

```
$ pip install django
$ django-admin startproject my-new-project
```

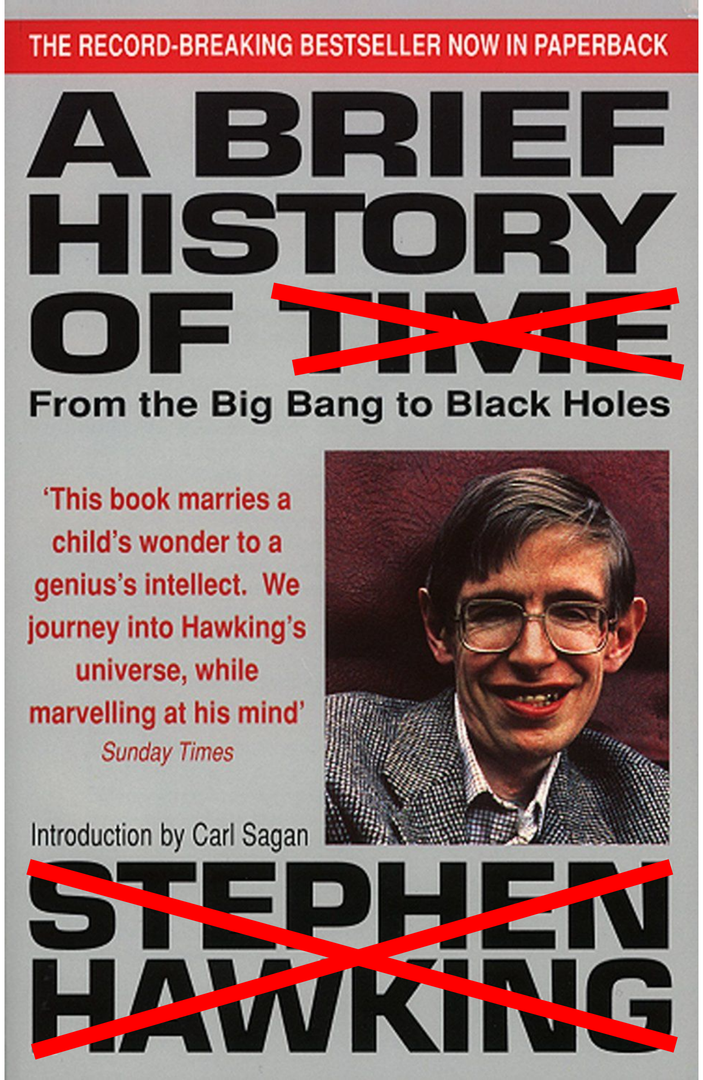
We can do it too!

```
$ pip install my-kitten-counter-package

$ count-kittens

$ python

>>> from count_kittens import kitten_counter
```

Why Package Your Code?

- Other people can use it
- Useable in multiple projects
- Higher code quality
  - Separation of Concerns
  - Public Scrutiny
- Easy Upgrades / Rollbacks

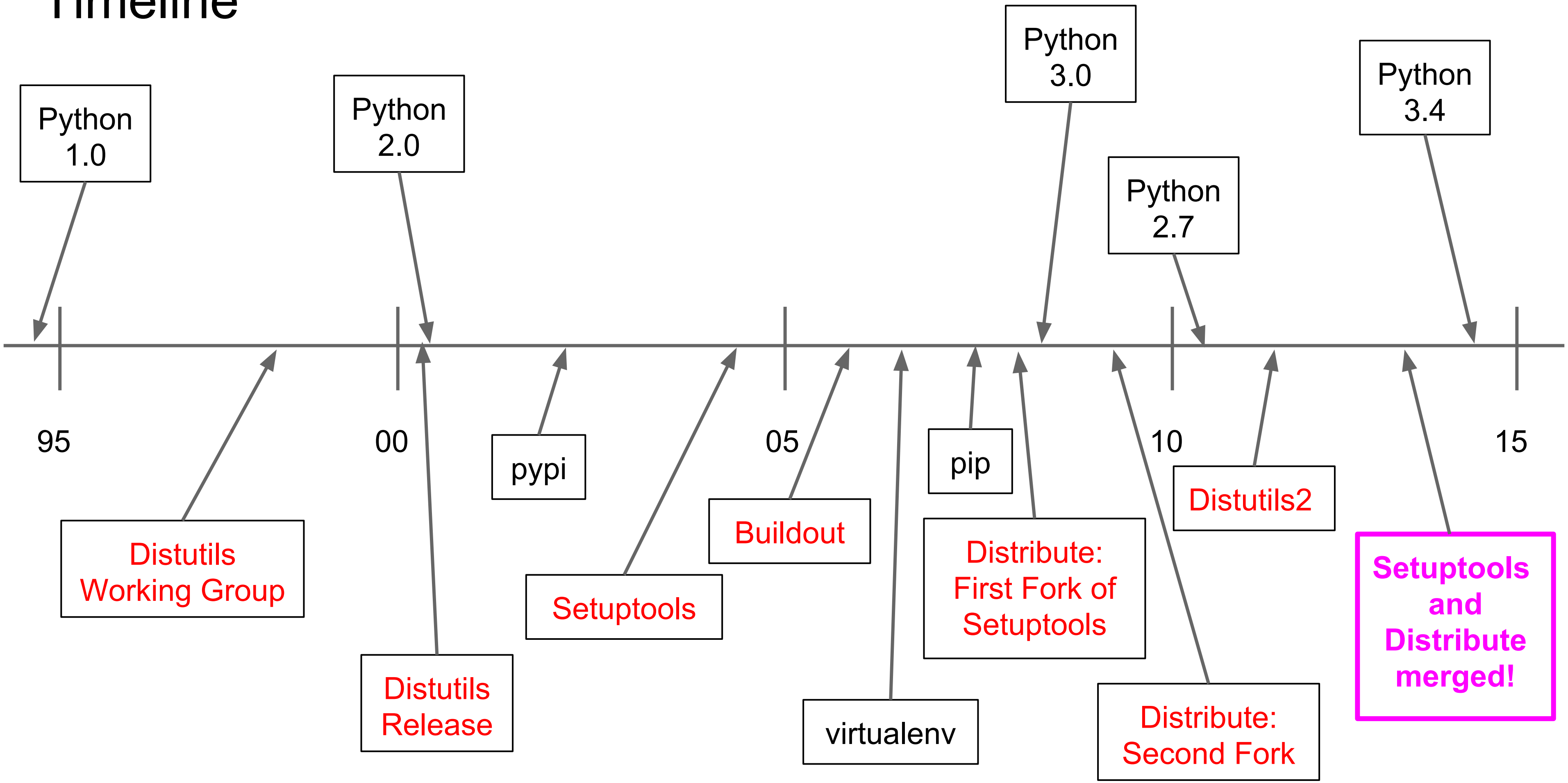
How Hard Is Packaging?


THE RECORD-BREAKING BESTSELLER NOW IN PAPERBACK
A BRIEF HISTORY OF ~~TIME~~
From the Big Bang to Black Holes
'This book marries a child's wonder to a genius's intellect. We journey into Hawking's universe, while marvelling at his mind'
Sunday Times
Introduction by Carl Sagan
~~STEPHEN HAWKING~~

PACKAGING

Erik Swanson

Timeline


Python 1.0
Python 2.0
Python 3.0
Python 2.7
Python 3.4
95
00
05
10
15
Distutils Working Group
Distutils Release
pypi
Setuptools
Buildout
virtualenv
pip
Distribute: First Fork of Setuptools
Distribute: Second Fork
Distutils2
Setuptools and Distribute merged!

A single source of truth: PyPUG

The “Python Packaging User Guide” (PyPUG) aims to be the authoritative resource on how to package, publish and install Python distributions using current tools.

python-packaging-user-guide.readthedocs.org/

The “Hard” Way

What goes into a package?

- A Python module, or modules
- Some additional metadata
    - setup.py
    - setup.cfg
    - README.rst

```
my_count_kittens_package/
    setup.py
    setup.cfg
    README.rst

    count_kittens/
        __init__.py
        kitten_counter.py

        tests/
            __init__.py
            test_kitten_counter.py
```

README.rst

- Like a README.md
- Uses reStructuredText
- Looks just as nice on github

setup.cfg

- setup.cfg is an ini file
- It contains option defaults for `setup.py` commands.
- Fine for it to be blank at first

setup.py

- Main source of metadata
- Just python code
- Entry point for packaging tasks

```
$ python setup.py <command>
```

```python
from setuptools import setup, find_packages

setup(
    name=’kitten_counter’,
    description=’Easily to count kittens’,
    packages=find_packages(),
    install_requires=[‘Django>=1.8’]
    # more arguments...
)
```

Example

Install Requirements

```
$ pip install setuptools
$ pip install twine
$ pip install wheel
```

Uploading to PyPi

- Where publicly available packages are stored
- pypi.python.org
- You’ll need to register for an account

Register / Upload your package

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

Maybe try TestPyPi

- If you want practice
- Packages are regularly cleared out
- Totally separate from PyPi
- testpypi.python.org

Register/Upload

```
$ python setup.py register
  -r https://testpypi.python.org
$ python setup.py sdist bdist_wheel upload
  -r https://testpypi.python.org
```

~/.pypirc

- Make it easy to specify servers to upload to
- ini file
- On windows, need a `HOME` environ var
- Example:
  https://python-packaging-user-guide.readthedocs.org/en/latest/distributing/#create-an-account

```
[distutils]
index-servers=
    pypi
    pypitest

[pypitest]
repository = https://testpypi.python.org/pypi
username = <your user name goes here>
password = <your password goes here>

[pypi]
repository = https://pypi.python.org/pypi
username = <your user name goes here>
password = <your password goes here>
```

Register/Upload

```
$ python setup.py register -r testpypi
$ python setup.py sdist bdist_wheel upload
  -r testpypi
```

The Easy Way

Start with a template!

github.com/pypa/sampleproject/blob/master/setup.py

```
$ git clone
https://github.com/pypa/sampleproject.git

$ mv sampleproject my_project

$ cd my_project

$ rm -rf .git

# Make your edits, write your code
```

Upload to PyPi

```
$ python setup.py register

$ python setup.py sdist bdist_wheel upload
```

Private Packages

Local Package Creation

```
$ python setup.py sdist --formats=zip,gztar

$ cd dist/

$ cp my_package-1.3.5.zip ~/LocalPython

$ pip install ~/LocalPython/my_package-1.3.5.zip
```

DevPi

- Host your own pypi.python.org
- Cache external python packages
- http://doc.devpi.net/latest/

```
$ pip install internal_package
  -i https://my.devpi.server/root/
```

Don’t accidentally Upload to PyPi

```
setup(
    classifiers=[
        ‘ProgrammingLanguage :: Python’,
        ‘Private :: DoNotUpload’
    ]
)
```

Bonus Topics!

Adding Extra Data Files

pythonhosted.org/setuptools/setuptools.html#including-data-files

```
setup(
    #...
    package_data={
        ‘’: [‘*.txt’, ‘*.png’],
        ‘counter’: [‘defaults.cfg’]
    }
)
```

Distributing Python Scripts

```
setup(
    #...
    entry_points={
        'console_scripts': [
            'ck=kitten_counter.scripts:main’,
        ]},
    #...
)
```

Python 2 and Python 3 compatibility

- You can use a single codebase
- Use the `six` library
  - Provides a compatibility layer
  - https://pypi.python.org/pypi/six
- “Cheat Sheets” python-future.org/compatible_idioms.html

```
import six

for k, v in six.iteritems({‘abc’: 123}):
    print(k, v)
```

Using Anaconda.org

conda.pydata.org/docs/build_tutorials/pkgs.html

```
# Really easy if it’s already on Pypi!

$ conda skeleton pypi my_package

$ anaconda login

$ anaconda upload /path/to/my_package.tar.bz
```

GPG Signing

```
$ pip install twine

$ python setup.py sdist bdist_wheel

$ twine upload -r pypi dist/* -i

# https://pypi.python.org/pypi/twine
```

Thanks For Coming!

python-packaging-user-guide.readthedocs.org

Questions? Also, come to Build Night!