

What happens if I delete setup.py?

Heather White · @eevelweezel

To start with...

- Python packaging is developer-centric: more options, fine-grained control
- **Packaging tools** alone are a broad topic: build (setuptools, build, hatchling) vs project management (poetry, hatch, venv)
 - Some tools span categories (pdm, flit)
 - This talk will focus on **build**
- In software, **stable** is a synonym for **slowly rotting**
 - **Best practice** is constantly evolving
 - Python packaging has changed a lot over the years: wheels vs eggs, setuptools vs easy_install, now we have pip and not distutils, etc.

A Brief Timeline

- distutils introduced in Python 2.0 (released October 2000)
 - First introduced “from distutils import setup” and setup.py
 - Deprecated in 3.10, removed in Python 3.12
- setuptools announced in 2004, as an enhancement for distutils
 - provided all the necessary tools to build and upload a project to PyPI
 - upload was deprecated in 2013 (replaced by twine)
- pip released in 2011 (has had many improvements since then)
- 2015: PEP-517 & PEP-518 paved the way for new-style packaging
 - PEP-517 - separate the build system from the runtime & define a build API
 - PEP-518 - specification of pyproject.toml

Pyproject.toml: PEP-517, PEP-518, and Beyond

- These PEPs define a minimal spec that a build system should follow
 - Must implement `build_wheel` and `build_sdist` methods
 - Build metadata can be separated from `setup.py` using `setup.cfg` (2019), or later, `pyproject.toml` (2020)
- Reasons to use an alternate build system
 - Better isolation and environment management
 - Ease of use
 - Because you want to
- You're also welcome to keep using `setuptools`, it supports new-style packaging

Demos

1. Let's look at what's in:
 - a. twisted's setup.py (circa v17.1.0), setup.cfg, and pyproject.toml (circa now)
 - b. cryptography's pyproject.toml [build-system] element (circa now)
2. Let's repack my sparklepony project:
<https://github.com/eevelweezel/sparklepony>
 - a. Upgrade to pyproject.toml
 - b. Setup pre-commit
 - c. Build a wheel & sdist, then upload to test.pypi.org
 - i. Yes, there's a TestPyPI, and you, too, can use it!
 - ii. Extra credit: setup a test.pypi.org account and try these steps yourself with sparklepony!

Further reading

PyPA Packaging Docs: <https://packaging.python.org/en/latest/>

Sparklepony: <https://github.com/eevelweezel/sparklepony>

TestPyPI: <https://test.pypi.org>

Getting Started with Setuptools (circa 2009):

https://pythonhosted.org/an_example_pypi_project/setuptools.html

Bernát Gábor's Python Packaging - Past, Present, Future:

<https://bernat.tech/posts/pep-517-518/>

Heather White (Wææzəl)

eevel.weeziel@gmail.com

<https://github.com/eevelweeziel>

