Reusable, Reproducible, Useful Computational Science in Python

An interactive tutorial presented by Charles Tapley Hoyt. If you want to get in touch, tweet me @cthoyt. If you want to see an old incarnation of this tutorial, check out the June 2020 version.

This document can be accessed at https://bit.ly/cth-reproducibility-2021 and the associated slides at https://bit.ly/cth-reproducibility-2021-pres. They're both available under the CC-BY-4.0 license.

Recording of this event: https://www.youtube.com/watch?v=f6brWkO9OiE

Abstract

This tutorial will teach you how to make sure your Python code is reusable, reproducible, and ultimately useful for others. We will cover organization of Python code, packaging with setuptools, installing code in development mode with pip, unit testing with unittest/pytest, linting with black, code quality assurance with flake8, documentation with Sphinx, automation with Tox, continuous integration with GitHub Actions, documentation hosting with ReadTheDocs, implementation of command line interfaces using Click, versioning with bumpversion, archiving on Zenodo, and ultimately, deployment to PyPI. This tutorial will be appropriate for Python programmers of ***all*** skill levels and for projects of ***all*** sizes. It's free to attend and will be posted on YouTube after.

Logistics

Duration	3 Hours
Venue	Online (Zoom)
Host	Prof. Dr. Olga Vitek on behalf of the May Institute at Northeastern University
Link	https://northeastern.zoom.us/j/94948087309
Date	Wednesday, July 28, 2021
Start time	8 AM PST - 11 AM PST 11 AM EST - 1 PM EST 17.00 CEST - 20.00 CEST

Prerequisites

This isn't *really* a tutorial about coding in Python, but more about all of the best tools to use while/after you're doing it. This means that anyone, with any level of proficiency in Python can join and learn something - even if you're just at `Hello World!`. Most of the ideas from this talk can also be brought to other languages, so even if your favorite is another, you might still learn something.

This tutorial will be live-coded, but you can feel free to follow along and ask questions! Before the talk, I'll post a link to a repository with all the code ready in case you'd rather copy/paste or read in your favorite text editor or IDE. If you want to follow along, you'll need the following free accounts before we begin:

- GitHub (https://github.com)
- Zenodo (https://zenodo.org; use GitHub account to sign in)

- Read the Docs (https://readthedocs.io; use GitHub account to sign in)
- Test PyPI (https://test.pypi.org; it's like the real PyPI but for learning and testing)

You'll also have to be comfortable using the command line and have the following software installed:

- Git (https://git-scm.com) and optionally the GitHub Desktop GUI (https://desktop.github.com/)
- Python 3 (https://www.python.org preferably 3.8+ because I might throw in a walrus operator :=)
- Make (should already be installed on *nix systems, sorry Windows users)

It would be best to work in an IDE like PyCharm, have a nice git GUI set up (you're not a martyr; I use GitHub Desktop), and have the following `tox` python package installed and ready to go (it will do all of the hard work for us).

Do You Like Baking?

I always thought it was funny on baking shows when they put something in the oven, then pull the same thing out of the next oven, already done. What great planning! In that spirit, the code for this tutorial, already finished, is available at https://github.com/cthoyt-teaches-reproducibility/iter-together-FINISHED before we even begin. This will allow you to quickly copy/paste code out of it if you're trying to keep up, or later go back and review at your own pace.

Outline

Here's my proposed outline of what's going to happen. Maybe I can even get fancy and tag the times we get to each of these on our live stream for later!

1. Make a repository
2. Choose a License (https://choosealicense.com)
3. Update .gitignore (https://gitignore.io)
4. Fix README and outline project
5. Make package structure and add version.py
6. Begin setup.cfg and setup.py
7. Introduce pyroma, tox, and add first environment to tox.ini
8. Set up GitHub Actions
9. Finish setup.cfg
10. Make function stub
11. Introduce flake8 and add environment to tox.ini
12. Make tests structure and write unit tests
13. Introduce pytest and add environment to tox.ini
14. Implement function and check tests pass
15. Introduce coverage configuration
16. Introduce sphinx, create documentation, and build documentation
17. Add documentation tests (doc8, docs, and readme) to tox.ini
18. ?Set up ReadTheDocs (https://readthedocs.org)
19. Introduce click, implement CLI, add CLI tests, add CLI documentation
20. Introduce MANIFEST.in and add check-manifest environment to tox.ini
21. ?Set up Zenodo (https://zenodo.org)

22. ?Introduce bumpversion and *tox -e finish*
23. ?Upload to Test PyPI
24. ?The *One True Test*
25. Writing a README
26. Badges, Badges Everywhere 👋
27. This time, with feeling (Cookiecutter)
28. Demo: fix https://github.com/PatWalters/clusterama

Extra Resources

My Blog posts on python packaging and projects:

- https://cthoyt.com/2020/06/03/how-to-code-with-me-organization.html
- https://cthoyt.com/2020/04/25/how-to-code-with-me-flake8.html

My YouTube playlist of cool talks from PyCon etc. on python programming, documentation and more:

- https://www.youtube.com/playlist?list=PLPFmTfhIBiumfYT3rsa35fHJxabB78er1