



Best practices

Using environments

One thing that's helped me tremendously is having separate environments for Python 2 and Python 3. I used `conda create -n py2 python=2` and `conda create -n py3 python=3` to create two separate environments, `py2` and `py3`. Now I have a general use environment for each Python version. In each of those environments, I've installed most of the standard data science packages (numpy, scipy, pandas, etc.). Remember that when you set up an environment initially, you'll only start with the standard packages and whatever packages you specify in your `conda create` statement.

I've also found it useful to create environments for each project I'm working on. It works great for non-data related projects too like web apps with Flask. For example, I have an environment for my personal blog using [Pelican](#).

Sharing environments

When sharing your code on GitHub, it's good practice to make an environment file and include it in the repository. This will make it easier for people to install all the dependencies for your code. I also usually include a pip `requirements.txt` file using `pip freeze` ([learn more here](#)) for people not using conda.

More to learn

To learn more about conda and how it fits in the Python ecosystem, check out this article by Jake Vanderplas: [Conda myths and misconceptions](#). And here's the [conda documentation](#) you can reference later, and a link to a [cheatsheet](#) to help you basic conda commands.

[NEXT](#)

