Summary

At this moment, you must have completed the following steps before moving to the next lesson:

1. Install and navigate through the Anaconda
2. Download Python packages in Anaconda Terminal
3. Setup and manage one or more environments

Below are a few best practices that you can consider to incorporate in your programming habit.

Using Environments

One thing that’s helped me tremendously is having separate environments for Python 2 and Python 3. I used the commands below to create two separate environments - py2_env and py3_env,

```
conda create -n py2_env python=2
```

```
conda create -n py3_env python=3
```

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 in addition to 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. You can do this using conda as:

```
conda env export > environment.yaml
```

Share the List of Dependencies

For users not using conda, you may want to share the list of packages installed in the current environment. You can use pip to generate such a list as requirements.txt file using:

```
pip freeze > requirements.txt
```

Later, you can share this requirements.txt file with other users over Github. Once a user (or yourself) switches to another environment, you can install all the packages mentioned in the `requirements.txt` file using:

```
pip install -r requirements.txt
```

You can learn more here about using pip instead of conda. This will make it easier for people to install all the dependencies for your code.

Recommended Read

- To learn more about conda and how it fits in the Python ecosystem, check out this article by Jake Vanderplas: Conda myths and misconceptions.
- Here's the conda glossary documentation for your reference.