An Introduction to Python Programming for Environmental Professionals

A brief primer to the Python Programming Language, which is widely used in data science and has significant potential for addressing many environmental topics.
Software, sensors, unmanned systems, and machine learning are becoming increasingly used for environmental applications such as monitoring water and air quality, assessing invasive species, and quantifying habitat losses. As a result, advanced computing skills are becoming essential for the management of environmental systems. Despite the importance of these skills for effective management of natural resources and the environment, many higher education programs provide little training in data science and computing skills needed to effectively make use of the new data revolution that is sweeping the planet. The educational curriculum needs to adapt to provide environmental students with the training in computing skills to address emerging problems such as climate change, energy sustainability, water and air quality degradation, and habitat destruction.

Software and data literacy is expected to grow in importance and will likely become a core competency for many environmental professionals. Universities need to rapidly create and grow a pool of skilled environmental data scientists to meet this demand. This article provides a brief introduction to the Python Programming Language, which is widely used in data science and has significant potential for addressing many environmental topics. Due to its popularity, Python already has a wide range of educational materials that can be extended to teach new skills that are increasingly important for environmental applications. An example application of Python for environmental modeling is presented that is publicly available for interested readers to explore.

**Python for Environmental Applications**

Python is perhaps the most commonly used programming language right now for data science on account of its readability and open-source nature, which provides flexibility and facilitates extensibility into new data domains. Languages described as “open source” can easily be modified to include new features and fix bugs. Python is designed to be easy to read and extend, which allows new users to perform complex tasks without having to re-invent the wheel.

In addition to Python or a data-focused language, other important software and data skills needed for data science are basic knowledge of the command line and revision control. Command-line knowledge is important because it underlies the interactions of operating systems, and code development often requires interactions with files and programs at this level. Familiarity with revision control software such as Git is needed for collaboration and development. Python (or R), revision control and command line skills are commonly taught at Software Carpentry workshops, which are provided free and run by volunteers (including the author) worldwide. The Software Carpentry curriculum provides a possible pathway for introduction of data science skills to environmental professionals.

I started using Python when I was searching for a language for a sediment remediation design application. I needed a programming language to make a user interface, manage a database of chemical and material properties, solve differential equations, and provide graphical output. I also wanted to be able to distribute the software to other users (environmental consultants) without an expensive license. Python and its auxiliary “libraries” provided all these features and many more. Libraries are bundles of code that provide new features to a Python installation. There are now over 70,000 libraries in the Python Package Index, including many environmental applications.

Table 1 summarizes some important libraries that are available in Python for environmental and data science applications. Having all these features in a single language facilitates usage of Python as “glue” to connect disparate applications. For example, I have developed additional libraries in Python to automate the entire process of constructing a hydrologic model, including the scraping of data from online websites, processing input time series, performing geospatial calculations, and calibrating hydrological models.

Python can be used for scraping data from online databases, performing advanced mathematics, visualizing environmental data, and developing machine learning models. Machine learning is a tool that can make connections between disparate datasets such as those encountered in environmental systems that is being used in almost every aspect of modern life.

There are many different interfaces available to run Python

<table>
<thead>
<tr>
<th>Python Library</th>
<th>Environmental Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numeric Python (NumPy)</td>
<td>Solution of linear system of equations, Image processing</td>
</tr>
<tr>
<td>Scientific Python (SciPy)</td>
<td>Statistical analysis, numerical integration methods, solution of differential equations</td>
</tr>
<tr>
<td>Math Plotting Library (Matplotlib)</td>
<td>Data visualization</td>
</tr>
<tr>
<td>Python Data Analysis Library (Pandas)</td>
<td>Database queries, time series manipulation</td>
</tr>
<tr>
<td>Geospatial Data Abstraction Library (GDAL)</td>
<td>Geographic information system data processing</td>
</tr>
<tr>
<td>Scikit-learn</td>
<td>Machine learning</td>
</tr>
</tbody>
</table>
Perhaps the easiest way to start working with Python is through the usage of Jupyter notebooks. Jupyter provides an interactive platform for coding and software demonstrations through a web browser interface that is ideal for educational purposes. The Anaconda Python Distribution includes common scientific Python packages, as well as many of the packages related to data analytics and big data. Downloading and installing Anaconda allows immediate access to the standard Python libraries as well as many of those listed in Table 1.

Jupyter allows for basic text formatting using Markdown, which is a language used on web pages that enables formatted output including headings, bullets, numbered lists, and tables. Complex mathematical expressions can also be developed in Jupyter notebooks using LaTeX syntax. These tools enable rapid development of environmental models, including documentation. If environmental education programs provided an introduction to an application-focused language in an environment like Jupyter early in a course in the curriculum, then reinforced their application throughout the environmental coursework, students could learn how to apply these tools and develop more advanced computing skills needed for complex environmental systems later. Environmental professionals could also learn these skills through professional development workshops.
A Jupyter notebook has been made accessible online that shows how Python can be used to build an interactive version of the Streeter-Phelps model and placed in a Github repository for public access. Figure 1 shows the heading of the notebook before and after the Markdown code is converted into output in the web browser. The figure illustrates how to develop a header, table, and equations in Jupyter by writing Markdown code on the left and the resulting formatted results on the right.

Figure 2 shows four cells that demonstrate how to import libraries (NumPy) and write functions, which in this case are the DO and BOD as a function of the distance and other Streeter-Phelps model parameters. Figure 3 shows how to create a plot of the model output values every 0.2 km for a 50 km river using Matplotlib, which can be placed in-line in Jupyter notebooks. The graph illustrates that for these model parameters, the DO drops below healthy levels of 5 mg/L for about 15 km of this reach of the river.

This example shows how to build a documented, modifiable interface to a model in only 20 lines of Python code. Additional examples can easily be developed for air quality and other environmental models such as the Gaussian Dispersion model for point sources, box model, etc. A variety of papers have been published recently describing ways to use Python for environmental data science problems.

Environmental education needs to embrace the new data revolution, which has potential to help solve many challenging issues. Data science tools can be used to describe complex environmental models, document parameters and equations the models utilize, and illustrate the model output with spectacular visuals. Jupyter notebooks that make use of Python can be published online and distributed to model users using Github. Approaches such as the example in this article can be developed to illustrate to students how to use software and data science tools to study environmental applications. Other important skills include command line usage, revision control, and high-level languages such as Python, Matlab, and R.

---

**References**

4. Lambert, D.J.; Wu, M. Development of an Open-Source Software Package for Watershed Modeling with the Hydrological Simulation Program in Fortran; *Environmental Modeling & Software* 2015, 68, 166-174; https://doi.org/10.1016/j.envsoft.2015.02.018.
8. LaTeX/ Mathematics—Wikibooks, open books for an open world. See https://en.wikibooks.org/wiki/LaTeX/Mathematics.
11. Kadiyala, A.; Kumar, A. Applications of Python to Evaluate Environmental Data Science Problems; *Environmental Progress & Sustainable Energy* 2017, 36 (6), 1588-1586.