

Programming with Python

The best way to learn how to program is to do something useful, so this introduction to Python is built around a common scientific task: data analysis.

Our real goal isn't to teach you Python, but to teach you the basic concepts that all programming depends on. We use Python in our lessons because:

1. we have to use *something* for examples;
2. it's free, well-documented, and runs almost everywhere;
3. it has a large (and growing) user base among scientists; and
4. experience shows that it's easier for novices to pick up than most other languages.

But the two most important things are to use whatever language your colleagues are using, so that you can share your work with them easily, and to use that language *well*.

We are studying inflammation in patients who have been given a new treatment for arthritis, and need to analyze the first dozen data sets of their daily inflammation. The data sets are stored in [comma-separated values](#) (reference.html#comma-separated-values) (CSV) format: each row holds information for a single patient, and the columns represent successive days. The first few rows of our first file look like this:

```
0,0,1,3,1,2,4,7,8,3,3,3,10,5,7,4,7,7,12,18,6,13,11,1
1,7,7,4,6,8,8,4,4,5,7,3,4,2,3,0,0
0,1,2,1,2,1,3,2,2,6,10,11,5,9,4,4,7,16,8,6,18,4,12,5,12,7,11,5,1
1,3,3,5,4,4,5,5,1,1,0,1
0,1,1,3,3,2,6,2,5,9,5,7,4,5,4,15,5,11,9,10,19,14,12,17,7,12,11,7,4,2,1
0,5,4,2,2,3,2,2,1,1
0,0,2,0,4,2,2,1,6,7,10,7,9,13,8,8,15,10,10,7,17,4,4,7,6,15,6,4,9,1
1,3,5,6,3,3,4,2,3,2,1
0,1,1,3,3,1,3,5,2,4,4,7,6,5,3,10,8,10,6,17,9,14,9,7,13,9,1
2,6,7,7,9,6,3,2,2,4,2,0,1,1
```

We want to:

- load that data into memory,
- calculate the average inflammation per day across all patients, and
- plot the result.

To do all that, we'll have to learn a little bit about programming.

Learners need to understand the concepts of files and directories (including the working directory) and how to start a Python interpreter before tackling this lesson. This lesson references the Jupyter (IPython) Notebook although it can be taught through any Python interpreter. The commands in this this lesson pertain to Python 2.7.

✓ Getting ready

You need to download some files to follow this lesson:

1. Make a new folder in your Desktop called `python-novice-inflammation`.
2. Download [python-novice-inflammation-data.zip](#) (`./python-novice-inflammation-data.zip`) and move the file to this folder.
3. If it's not unzipped yet, double-click on it to unzip it. You should end up with a new folder called `data`.
4. You can access this folder from the Unix shell with:

```
$ cd && cd Desktop/python-novice-inflammation/data
```

Topics

1. [Analyzing Patient Data](#) (01-numpy.html)
2. [Repeating Actions with Loops](#) (02-loop.html)
3. [Storing Multiple Values in Lists](#) (03-lists.html)
4. [Analyzing Data from Multiple Files](#) (04-files.html)
5. [Making Choices](#) (05-cond.html)
6. [Creating Functions](#) (06-func.html)
7. [Errors and Exceptions](#) (07-errors.html)
8. [Defensive Programming](#) (08-defensive.html)
9. [Debugging](#) (09-debugging.html)
10. [Command-Line Programs](#) (10-cmdline.html)

Other Resources

- [Reference](#) (reference.html)
- [Discussion](#) (discussion.html)
- [Instructor's Guide](#) (instructors.html)

Software Carpentry (<http://software-carpentry.org>)

Source (<https://github.com/swcarpentry/python-novice-inflammation>)

Contact (<mailto:admin@software-carpentry.org>)

License ([LICENSE.html](#))