Idealistic Python Course

September 9-13, 2019

Room: CCO Kleingruppen 03.003
Virchowweg 6
Charité Universitätsmedizin
Campus Charité Mitte
10117 Berlin

This is an introductory course on Python programming, with no assumed programming or other computing knowledge.

Although the narrow aim is to turn you into a Python programmer, the broader one is to show you how to work effectively on data, including via programming. So in addition to Python, we will spend a lot of time working with the bash shell (also a programming language) from the command line, and writing small shell scripts (i.e., programs written in the shell language). You'll also learn something about programming-related topics such source code control (via git & github) and code testing. There are many additional topics we could cover, depending on interests. E.g., plotting, useful packages (BioPython, numpy, & pandas), how the web works, etc. We'll collectively decide what to focus on.

When non-programmers working with data run into a problem they cannot solve with pre-existing tools (e.g., excel), the reaction is usually either to ask someone else for help, to do it manually, or to give up. In this course we will add another option: write a program. The course aim is concrete and modest: turn you into someone who, when faced with a data problem, thinks "I can write a quick script to solve that problem" (and then does so). Many people go back to their regular lives after taking courses, don't put into practice what they learned, and eventually forget it all. I don't want that to happen with our course. We'll aim to turn everyone into an active programmer. To get you going on a path that fundamentally changes how you work with data for the rest of your career. It's impossible to learn everything about programming in a week, but it is possible to take the crucial first steps and to make them stick.

Although I will formally lead the course, it will only be successful if it is a group effort. Attitude is extremely important. We'll get past the initial learning hump, teach each other (so if you already know a bit more, please come with the intention of helping others catch you up), learn by doing, learn where to find help, and end up with a small group who can continue to encourage and help each other in the longer term. Hence “idealistic”.

There is no a priori course agenda. We will work on simple, relevant real problems, rather than working our way through abstract programming concepts and exercises. We’ll solve practical problems that are just over the horizon of what one can do manually. Things that are very hard, error prone, and time-consuming by hand but easy and fast with a little code. If you have data and/or problems you'd like us to work on, please bring them.

Running the course is a big time commitment for me, so I will expect a similar commitment from you :-) If you sign up, please plan to dedicate as much time as possible to the course. That means spending your days in the class, and preferably spending your evenings and the weekend reading, programming, learning, doing homework tasks, etc. Make sure you have the time to really throw yourself into the course. If you can't commit this time around, don't worry, I will run the course again.

You’ll need to bring a laptop, ideally running Linux or Mac OS X. It’s not easy, and slows the whole class down, to support people using Windows. However, it can be done via Cygwin, the new Windows Subsystem for Linux, or a virtual machine such as VMware. Let me know in advance if you only have Windows.

Although you wont be expected to know anything ahead of time, the more you do know, the better. So:

- It would be good to familiarize yourself a little with the command line (sometimes referred to as “the terminal”, “the shell”, or bash).
- It would be good to know how to use a text editor, for editing plain text files. You cannot use a tool like Microsoft Word for this because it has its own special and obscure file format. I suggest trying Sublime Text (http://www.sublimetext.com), though there are many options - any of them should be fine. If unsure, ask.
- You could try learning some Python (version 3) ahead of time. E.g., by doing, at https://www.codecademy.com/learn/python or by reading, at http://learnpythonthehardway.org

These three things are closely tied together in what you’ll be doing. You'll write code in the Python language. You'll do that in a text editor. You'll then run those programs on the command line (using bash). You must become comfortable with those three tools. The more you know about them before we start, the better.