Programming in Python

Instructor: Manuel Arriaga (marriaga@stern.nyu.edu)

Course objectives

Who this course is aimed at and what we will be doing:

This course is for people with no prior programming experience who want to develop a working knowledge of how to write computer programs, get data from the web and build their own web applications.

We'll learn the basic elements of programming through the language Python. At the end of this class, you can expect to be fairly proficient in using Python to build applications, understand enough about programming to be able to quickly pick up other languages (particularly scripting languages such as JavaScript, Ruby or Perl), and have a good understanding of what it takes to plan, analyze, design and implement a software application. You will understand how computer applications—be they large- or small-scale—work and you’ll be able to build working prototypes to illustrate and market your own ideas.

Who this course is not for and what we will not be doing:

(1) If you have programming experience, this probably is not the right course for you.

(2) Although programming skills are pretty much easily transferable across devices and architectures, we will not be covering mobile app development.

Scope of the course

Python and Cherrypy. Designed (by Guido van Rossum) to be simple, readable, and uncomplicated, Python is about as intuitive as a programming language can get. Cherrypy is a web-framework, software that is designed to make the development (and maintenance) of a web app as painless as possible.

Python2 vs. Python3. Unfortunately, there are two versions of Python and they aren't fully
compatible. We'll be using Python3 (current version: 3.5.2) mainly because Python2 is slated to go away.

**HTML, CSS, and JavaScript.** HTML is the markup language used to create web pages. CSS is the style sheet language that integrates with HTML to define the 'look and feel' of a website (e.g., CSS is what gives all Stern pages the same look). JavaScript is a programming language frequently used *within* web pages to give them additional functionality. We won’t have the time to cover JavaScript in class but you will have the tools and knowledge to pick it up quickly if necessary.

**Evaluation and learning components**

**Mini Quizzes:** We’ll have several, very short, quizzes mainly to reinforce points made in class and also to help you get your hands dirty. Most quizzes will be online 'do whenever you have the time', though some may be in-class. All quizzes are open book and you’re welcome to check your solutions on your computers. Quizzes will be lightly graded so make a good faith effort and you’ll do fine. One or two quiz scores will be dropped in computing your quiz grade so no worries if you mess one up.

**Home assignments:** We’ll have a few home assignments as well. Like the quizzes, assignments are not meant to be diagnostic but rather to help you practice and learn so they will be very (very!) lightly graded. You can consult with others, ask me questions, use Google for help, but do try them on your own first. Because I’ll either discuss the solution in class or put it up on the course site, late submissions will not be accepted (sorry!).

**Project:** There is no better way to learn something than to go out and use it so start thinking about a web application that you think you’d like to build. The expectation is that you use the material we’ll cover in this class to plan, design, and implement a small web application. Your project grade will depend on how well your work illustrates your understanding of the course material. Final submission will include a design report, Python code, and an in-class "speed-date" presentation and demonstration.

**Participation:** Given the nature of our course, there most likely won’t be scope for a whole lot of in-class discussion. However, as we progress through the material I will frequently pause for you to have an opportunity to ask questions and demonstrate how you would apply what we just covered.

**Computers and the class**

Computers are a requirement for this course and you are expected to bring one for every class.
We'll do a lot of programming—the best way to learn is to see something in action and Python is an especially good language for making things happen. Make sure that your laptops have sufficient charge for the duration of the class.

**Python runs on all mainstream operating systems (Linux, Mac, Windows):** Before our first session, follow the instructions online to make sure that you have Python 3 installed.

**Reading materials**

There is no required text for this class. You will find an abundance of materials on the web on how to program in Python (and how to use Cherrypy), since these are widely used tools all over the globe. With this said, if you wish to get a book I recommend *Think Python, 2nd Edition* by Allen B. Downey.

You can also opt to follow the [Python tutorial](https://www.python.org) on python.org.