Programming in Python: Spring 2016

An online version of the syllabus is available at http://hardeep-johar.github.io/Web-Apps-Programming-in-Python-Spring-2016.html

What we will cover

- **Python**: A relatively easy but powerful programming language
- **Web scraping**: Gathering data from web pages (using Python)
- **Data analysis**: Using python for data analysis
- **Web app construction**: Using python to build a server side web application

Who should take this course

Frankly, everyone! We live in an information age, surrounded by data and with computer programs popping up around us all the time. If you want to move effectively through this information space you need to understand what programming is all about. You need to know what is possible with computers, you need to be able to communicate effectively with the IT people who work for you, and you need to be able to demonstrate the usefulness of your own ideas by building prototype applications. This class will give you the tools to make all this possible.

No programming background necessary!

Though we cover a lot of ground, we start at point zero so no prior programming experience is required or even necessary. All you need is the desire to learn and a willingness to work hard. Come with those two prerequisites and you will be surprised how delightfully fulfilling and useful programming can be.

Contact/Meeting times

Prof. Hardeep Johar

**Class hours**: Tuesday. 6:00pm to 9:15pm

Interested? Want more information?

- View the course content summary
- Read the Syllabus
- or email me at hj2203@columbia.edu

Content Summary

<table>
<thead>
<tr>
<th>Part 1: Python basics and web scraping</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>What we'll learn</strong>: the basics of <strong>Python programming</strong></td>
</tr>
<tr>
<td><strong>What we'll do</strong>: <strong>scrape the web!</strong>: go to the web and get data</td>
</tr>
<tr>
<td><strong>What you may find surprising!</strong> how easy web scraping can be! If you've never done programming before, you'll be surprised at the rudimentary nature of a computer's 'brain'. And, you may be surprised at how hard</td>
</tr>
</tbody>
</table>
it is to change your mind set and 'think' like a computer.

Part 2: Repetition, modularity, and data analysis

**What we'll learn**: how to do stuff again and again and the basics of data analysis using Python

**What we'll do**: crawl the web for data, learn about APIs

**What you may find surprising!** That the Web is no mystery and digging around inside it is actually fairly simple. And once you have the data on your computer, using Python's data analysis tools is a breeze!

Part 3: Building your own web app

**What we'll learn**: object oriented programming; using Django and the Model-View-Controller paradigm to design and build your own web application. We'll also learn the basics of SQL, a popular language for interacting with databases

**What we'll do**: build your own web application

**What you may find surprising!** Going from your idea to a working prototype of a web application is simple, as long as you know where you want to go!

---

**Syllabus details**

**Course objectives**

The goal of this class is to give you a working knowledge of how to write computer programs, to get data from the web, and to build your 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, implement and support software applications. We'll tour Python libraries for data analysis and, finally, we'll put all these components together to learn how to crawl the web for data and to build web apps.

Your reward, at the end of the course (assuming you do everything seriously!) will be substantial. You'll understand how computer applications - large scale ones or small scale ones - work and you'll be able to build working prototypes to illustrate and market your own ideas. But, be aware that this is an intensive course that will require a lot of work and a substantial time commitment.

**Expectations**

**What you can expect from me.**

- That I will treat every student with respect and consideration.
- That I will answer every question (email, canvas, in class, office hours) in a timely fashion.
- That I will be there for you to help you take your project work as far as is possible.

**What I expect from you**
That you complete all course work in time.
That you arrive in class and be in your seat by the start of the class.
That you don't hesitate to ask for help from me or from the Teaching Assistant.
That you understand that this is a difficult class and that the demands on your time will likely be more than that in your other classes.
That, above all, you remember that the goal is to learn and to enjoy the process of learning.

Scope of the course

Python and Django. Designed (by Guido van Rossum) to be simple, readable, and uncomplicated, Python is about as intuitive as a programming language can get. Django is a web-framework, software that is designed to make the development and maintenance of a website as painless as possible. Django is written in Python for Python and runs on most web servers. With Python and Django you can prototype your web app 'off-server' and then, once you've tested it, easily port it to a web server.

Python2 vs. Python3. Unfortunately, there are two versions of Python and they aren't fully compatible. We'll be using Python3 (current version: 3.4.2) mainly because Python2 is slated to go away (though, in the open source world, that doesn't mean a whole lot!) but also because it is a better language.

HTML, CSS, and JavaScript. HTML is a markup language. Pieces of text are 'tagged' (bold, headings, list elements, buttons, forms) and these tags are interpreted by the browser when it renders a web page. CSS is a style sheet language that integrates with HTML to create a common 'look and feel' for a website (CSS is what gives all Columbia pages the same look). JavaScript is a programming language much used on websites (because client computers can run javascript programs). We won't have the time to cover JavaScript 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: Demonstrate engagement in the course by asking questions. I'll respond to every question, either online or, if the response is of general interest, in the classroom. Some classes have mandatory attendance (I'll let you know which ones).

Exam: The course has one in-class exam toward the end of the semester. The exam will be a short â€œin-classâ€œ closed book, and closed computer exam though you are welcome to bring in a one page cheat sheet.

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!

**Mac vs Windows:** Either is fine but, if you have the choice, then please use a Mac. It is much easier to install needed libraries on a Mac than it is on a Windows machine. In particular, if you have a Mac and are using some sort of Windows emulator then please use Mac OS-X and not the Windows emulator for the work you do in this class. The double redirection will make everything a lot slower. But, either Mac or Windows will work so don't worry if you're a Windows user.

**Texts**

There is no required text for this class because, unfortunately, most books are designed for people who want to become computer programmers. However, I strongly suggest that you purchase the Mark Lutz book below so that you have something to read and work with outside the class. In the class, our focus is on usability and therefore we'll use a mix of online resources, class slides and notes, and sample programs as a substitute for formal texts. The internet is an almost endless resource with excellent tutorials on almost everything and answers to practically every question you might have and I'll point you to these resources as we move along the course.

- **Learning Python, 5th Edition Powerful Object-Oriented Programming**, Mark Lutz. O'Reilly Media, 2013. No one can call this 1600 page tome a lightweight, and it isn't. Easily the best book to keep around when you want to delve deeper into the language.
- **Python Programming Fundamentals**, Kent D. Lee. Springer-Verlag, 2015. This is a slimmer book but it covers most of what we need.

**Online resources**

- [Python documentation at Python.org](https://docs.python.org/)
- [Python tutorial at Codeacademy](https://www.codecademy.com/learn/learn-python-2)
- [Learn python the hard way](https://learnpythonthehardway.org/book/)
- [The Django tutorial](https://docs.djangoproject.com/en/3.0/)