

Web Apps Programming in Python

Prof. Hardeep Johar

hj2203@columbia.edu

Course Objective

The goal of this class is to give you a working knowledge of what it takes to build a web application. We'll learn the basics of the Model-View-Controller paradigm, the most common approach to building client-server applications (like web applications), learn the basics of constructing and using a data model in conjunction with a web application, see how to build authentication into an application using Facebook and Google APIs, and learn how to make pages dynamic with JavaScript.

This is a fairly technical course and, at the end of the week, the expectation is that you will have developed a prototype web application that you will demo to the rest of the class. We will review the basics of Python in half a day so you should have had some programming experience prior to taking the class!

Class format

We'll do a lot of programming so be ready with your laptops. Each session will be a mix of lectures (though you should be ready to follow along) and working on your web application. The TAs and I will be available to help.

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.

Prerequisites

Knowledge of the basics of python (lists, dictionaries, functions, objects) is a necessary prerequisite for this class. You must have either taken B8136, Data Analytics in Python (B8139) or passed the exemption exam (B0001).

Software

- **python:** We'll use the latest version of Python, 3.8/3.9, from python.org. (<https://www.python.org/downloads/>) or from anaconda (<https://www.anaconda.com/products/individual>)
- **PyCharm:** PyCharm is a Python and Django development environment. You need to download the professional version of PyCharm. One year student licenses are available at <https://www.jetbrains.com/student/> (you'll get a license to all their products but you only need to download the PyCharm professional edition).

Topics

Basics:

1. the MVC (Model View Controller) paradigm
2. virtual environments (building a python virtual environment)
3. deploying an app on an external host (most likely [heroku](#))

Django: Django is a web-framework, software that is designed to make the development and maintenance of a website as painless as possible. Django is an MVC ([or MTV - I'll explain](#)) framework. We'll examine the structure of a Django app to see how we can build an MVC app.

Database servers: Database servers are applications that make databases resources accessible to other programs. We will use a database server (sqlite3 and PostgreSQL) to store content data and to record data about users who visit the web app we're building. Luckily, Django hides SQL (the language of relational databases) and we don't have to learn how to use SQL

HTML/CSS: The language of web pages. 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 formats for a website. Mostly, you're going to have to read this up on your own. [This tutorial](#)

JavaScript: A high-level language used to make web pages interactive and is often embedded inside the html on a web page. We'll learn the basics of JavaScript, enough to add basic interactivity to our web pages.

Other stuff: web scraping, APIs, maps, and whatever fun stuff we can fit in

Evaluation components

- **Quick quizzes:** Quizzes will cover basics of python and any material that we've covered in class
- **Individual assignments:** A few short programming assignments, mainly for practicing material covered in class.
- **Project:** The cornerstone of the class. You will work in small groups and the expectation is that, at the end of the semester, you'll have a working prototype of a web application that incorporates all the elements that we cover in the class.
- **Project presentation:** You will, as a group, present your applications in a to-be-determined format
- **Participation and attendance:** Attendance is mandatory and you'll be docked points for unexcused absences