

Intro to Coding Using Python at the Worcester Public Library

Melody Friedenthal

ABSTRACT

The Worcester Public Library (WPL) offers several Digital Learning courses to our adult patrons, and among them is “Intro to Coding Using Python”. This 6-session class teaches basic programming concepts and the vocabulary of software development. It prepares students to take more intensive, college-level classes.

The Bureau of Labor Statistics predicts a bright future for software developers, web developers, and software engineers. WPL is committed to helping patrons increase their “hireability” and we believe our Python class will help patrons break into these lucrative and gratifying professions... or just have fun.

HISTORY AND DETAILS OF OUR CLASS

I came to librarianship from a long career in software development, so when I joined the Worcester Public Library in January 2018 as a Public Services Librarian, my manager proposed that I teach a class in programming. She asked me to research what language would be best. Python got high marks for ease of use, flexibility, growing popularity, and a very active online community.

Once I selected a language, I had to choose an environment to teach it in – or so I thought. I had absolutely no experience in front of a classroom, and few pedagogical skills, so I sought out an online Python course within which to teach.

I decided to use the Code Academy (CA) website as our programming environment. CA has self-guided classes in a number of subjects and the free Beginning Python course seemed to be just what we needed. I went through the whole class myself before using it as courseware.

My intent was to help students register for CA, then, each day, teach them the concepts in that day’s CA lesson. They would then be set to do the online lesson and assignments.

We first offered Python in June 2018. Problems with CA came up right from the start: students registered for the wrong class (despite the handout explicitly naming the correct class) and CA frequently tried to upsell to a not-free Python class.

Since CA’s classes are MOOCs (Massive Open Online Courses), the developers built in an automated way of correcting student code: embedded behind each web page of the course, there’s code that examines the student’s code and decides whether it is acceptable or not. Good in theory, not so good in practice. CA’s “code-behind” is flawed and sometimes prevented students from advancing to the next lesson.

Melody Friedenthal (mfriedenthal@mywpl.org) is a Public Services Librarian, Worcester Public Library.

Moreover, some of the CA tasks were inane. For example, one lesson incorporated a kind of *Mad Libs* game. This is where the instructions ask, for example, for 13 nouns and 11 adjectives, and these are combined with set sentences to generate a silly story. This assignment turned out to be too long and difficult to fulfill, preventing students from advancing.

Although I used CA the first few times I offered the class, I subsequently abandoned it and wrote my own classroom material.

After determining that CA wasn't appropriate, I chose an online IDE where the students could code independently. This platform worked well when I tested it ahead of time, but when the whole class tried to log on at once, we received denial-of-service error messages. Hurriedly moving on to Plan C, I chose Thonny, a free Python IDE which we downloaded to each PC in the Lab (see <https://thonny.org/>).

Each student receives a free manual (see figure 1), which I wrote. Every time I've offered this class I've edited the manual, clarifying those topics the students had a hard time with. I've also added new material, including commands students have shown me. It is now 90 pages long, written in Microsoft Word, and printed in color. We use soft binders with metal fasteners.

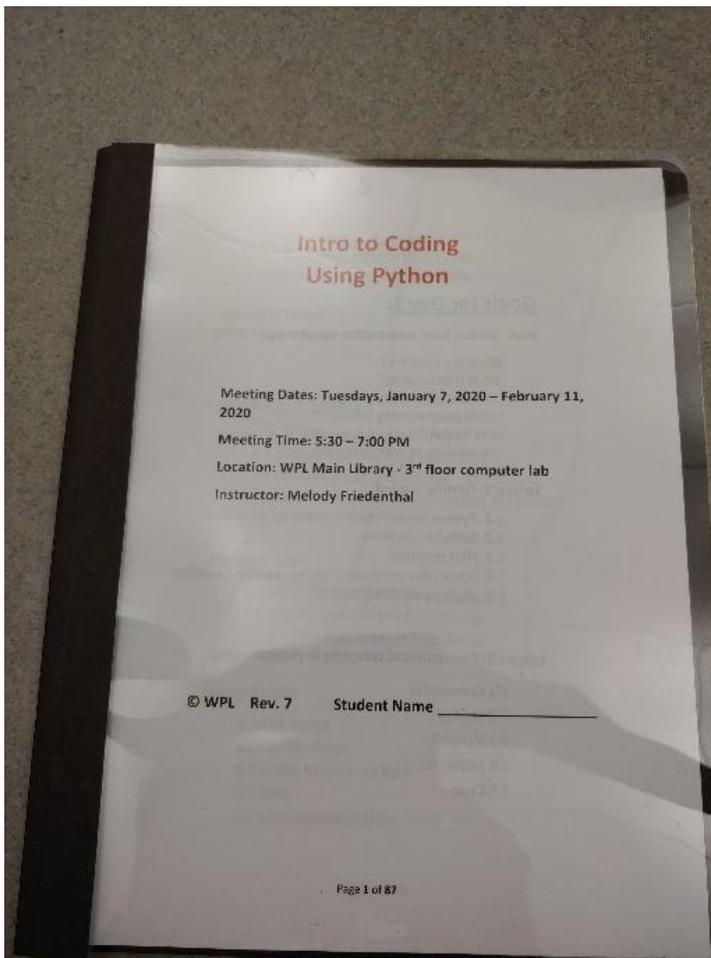


Figure 1. *Intro to Coding Using Python* manual developed for the course.

The manual consists of the following sections:

- Cover: course name, dates we meet, time class starts and ends, location, instructor's name, manual version number, and a place for the student to write their own name.
- Syllabus: goals for each of the six sessions. This is aspirational.
- Basic information about programming, including an online alternative to Thonny, for students who don't have a computer at home and wish to use our public computers for homework.
- Lessons 1 – 17: "Hello World" and beyond.
- Lesson 18: Object Oriented Design, which I consider to be advanced, optional material. Skipped if time is pressing or the class isn't ready for it.
- Lesson 19: Wrap-up:
 - How to write good code.
 - How to debug.
 - List of suggested topics for further study.
 - Online resources for Python forums and community.
- List of WPL's print resources on Python and programming.
- Relevant comic strips and cartoons.

In March 2019, my manager asked me to start assigning homework. If a student attends all six sessions and makes a decent attempt at each assignment, at the sixth session they receive a Certificate of Completion. The certificate has the WPL name & logo, the student's name, and my signature. Typically three or four students earn a certificate.

Homework is emailed to me as an attachment. This class meets on Tuesday evenings and I tell students to send me their homework as soon as possible. Inevitably, several students don't email me until the following Monday.

While I don't give out grades, I do spend considerable time reviewing homework, line by line, and I email back detailed feedback.

When the January 2020 course started, I found that between October's class and January, Outlook implemented a security protocol which removes certain file extensions from incoming email. And – you can see where this is going – the .py Python extension was one of them. I told students to rename their Python code files from xxxx.py to xxxx.py.doc, where "xxxx" is their program name. This fools Outlook into thinking the file is a Microsoft Word document and the email is delivered to me intact. When it arrives, I remove the .doc extension from the attachment and save it to a student-specific file. Then I open the file in Thonny and review it.

Physically, our Computer Lab contains an instructor's computer and twelve student computers (see figure 2). It also has a projector which projects the active window from the instructor's computer onto a screen: usually the class manual. I use dry erase markers in a variety of colors to illustrate the concepts on a whiteboard. There is also a supply of pencils on hand for student note-taking use.

The class is offered once per season. Although the classroom can accommodate twelve students, we set our maximum registration to fourteen, which allows us to maximize attendance even if patrons cancel or don't show up. And if all fourteen do attend the first class, we have two laptops I

can bring into the Lab. We also maintain a small waitlist, usually of five spots. We've offered this class seven times and the registration and waitlists have been full every time. Sometimes we have to turn students away.



Figure 2. Classroom at Worcester Public Library.

However, we had a problem with registered patrons not showing up, so last spring we implemented a process where, about a week before class starts, I email each student, asking them confirm their continued interest in the class. I tell them that if they are no longer interested—or don't respond - I will give the seat we reserved for them to another interested patron (from the waitlist).

In this email I also outline how the course is structured and that they can each earn a Certificate of Completion. I tell them class starts promptly at 5:30 and to please plan accordingly. Some students don't check their email. Some patrons show up without ever registering; they are told registration is required and to try again in a few months.

I keep track of attendance on an Excel spreadsheet. Here in Worcester, MA, weather is definitely a factor for our winter sessions.

Over time I've made the class more dynamic. I have a student read a paragraph in the manual aloud. I've switched around the order of some lessons, in response to student questions. I have them play a game to teach Boolean logic:

"If you live in Worcester **And** you love pizza, stand up!" ... then:

"If you live in Worcester **Or** you love pizza, stand up!"

Students range from experienced programmers (of other languages), to people with no experience but great aptitude, to people who just never seem to "get it". This material is technical and I try hard to communicate the concepts but I lose a few students every time.

We ask our patrons for feedback on all of our programs. Our Python students have written:

- "... the classes were formatted in an organized manner that was beginner friendly"
- "The manual is a big help. I'm thankful that the program is free."
- "... coding is fun and I learned a new skill."
- "This made me think critically and helped me understand where my errors in the programs were."

WPL is proud to offer classes that make a difference in our patrons' lives.