

**Free CS For All: PY4E Python for Everyone**

Tony Tam

San Francisco State University

ITEC 830 - Sprint 2023

Instructor - Dr. Brian Beatty

May 10, 2023

## Introduction

After being a software engineer and engineering manager for 27 years, I would like to leverage my experience to introduce computer science to eighth to 12 graders. I am inspired by Dr. Beatty's use of free textbooks in our ITEC courses to make learning more accessible. Even though there are widely adopted commercial LMS and learning tools available to learn to code, there are also very capable free, open content, LMS, and engagement platforms to teach computer science.

I have been a hands-on coder for many years, but for the last seven years, my role has been in people management, hence my coding skills have atrophied. In this paper, I will demonstrate how I personally brush up my coding skills in Python using PY4E.com (Python for Everyone), and how PY4E can teach children programming via a 100% free content, engagement, and assessment learning management system.

## What is PY4E

PY4E (Python For Everyone) (*PY4E - Python for Everybody*, n.d.) is a free, online, and open course to learn the programming language Python. Python is one of the most popular programming languages used for developing websites, data analysis, and general-purpose tools. PY4E is built on top of a free and open-source MOOC platform called Tsugi. (Severance, n.d.). The website, content, and MOOC platform is created by Charles Severance who is a Clinic Professor at the University of Michigan. He is a self-proclaimed "long-time advocate of open source educational technology and open educational resources to empower teachers." (*Charles R. Severance Short Biography*, n.d.). The same course is offered on the Coursera platform and has

220,000+ ratings and 2,845,864 enrolled. (*Programming for Everybody (Getting Started With Python)*, n.d.). Even though not everyone who is enrolled completes MOOC lessons, the statistics on Coursera are an indication of the wide interest in a Python course.

PY4E provides content (see Figure 1), engagement, and assessment tools to support the learner to gain foundational knowledge to learn how to code in Python.

Figure 1: List of 17 lessons available on PY4E

PY4E
Lessons
Discussions
Assignments
Instructor 
Book

## Python for Everybody (PY4E)

Hello and welcome to my site where you learn Python even if you have no programming background.

<b>1: Installing Python</b>  The first task is to work through the installation steps including	<b>2: Why Program?</b>  We learn why one might want to learn to program, and look at the	<b>3: Variables, expressions and statements</b>  We learn how to make variables
<b>4: Conditional Execution</b>  We look at how Python executes some statements and skips others.	<b>5: Functions</b>  Take a brief look at how Python implements the 'store and use	<b>6: Loops and Iterations</b>  We look at how Python repeats statements using looping
<b>7: Strings</b>  We look at how Python stores and manipulates textual data using	<b>8: Files</b>  We learn how to open data files on your computer and read through	<b>9: Lists</b>  We look at Python's simplest data structure - the list. Lists can store
<b>10: Dictionaries</b>  The dictionary data structures allows us to store multiple values	<b>11: Tuples</b>  The tuple is a Python data structure that is like a simple and	<b>12: Regular Expressions</b>  Regular Expressions allow us to search for patterns in strings and
<b>13: Network Programming</b>  We take a quick look at how data moves across the network using	<b>14: Using Web Services</b>  Web services allow a program to access data available in a different	<b>15: Object-Oriented Programming</b>  We do a quick look at how Python
<b>16: Databases</b>  Databases give us very fast random access to large amounts	<b>17: Data Visualization</b>  In this section, we learn to scrape data from the network, store the	

### **Free Content: Videos, Slides, References**

Each lesson in P4E has links to Youtube videos created by the instructor, slides used in the videos, and reference guides to deep dive into concepts, quizzes to guide student learning and hands-on projects to cement knowledge. See Figure 2 for an example of the Network Programming lesson on how the content is layout.

In the series of videos hosted on Youtube where the instructor explains concepts or they work through a programming example. Most of these videos are at most 12 minutes long. This is helpful for students who learn by watching and Youtube is a very familiar video player for children. The instructor makes extensive use of analogies to explain abstract concepts such as comparing network ports to telephone extensions. I highly recommend watching the videos to make connections and gain an understanding of abstract computer science concepts.

As a lesson plan, I would design my course to ask my students to watch these videos and write a reflection using Google Docs about 3 things they learn from each of the videos and also one question or topic they would lead a discussion in our synchronous class.

PY4E also makes PowerPoint slides reference in the Youtube video freely available if a teacher or student needs to go through the material at their own pace. Since students learn better in their own ways, having content in different formats is yet another way PY4E helps make learning more accessible.

As a lesson plan in my course, I would ask each student to screen capture their favorite slide in a Google doc and annotate the screenshot with something new they learn from that slide, and share that with their peers.

The reference guides are dense material consisting of explanations of concepts Dr. Chuck goes over in the video and slides. The material is helpful to answer the quizzes or to complete

the hands-on coding exercise. I will remind my students to make heavy use of the reference guides to help them through the coding portion.

As an experienced programmer, I leveraged the reference guides to refresh my memory. I skipped the slides and videos and jumped into the Autograder hands-on exercises and the quizzes to check my knowledge. For my students, I would encourage students with no to little computer science background to watch Youtube videos, PowerPoint slides, and reference guides. For more advanced students, they could skim over the slides and read the reference guides in detail to fill out their knowledge.

Figure 2: A single lesson showing the videos, slides, references, and tools

# Network Programming

[← Previous](#)
[All \(13 / 17\)](#)
[→ Next](#)

We take a quick look at how data moves across the network using the HyperText Transport Protocol (HTTP) and how we write programs to read data across the network.



## Videos

- ▶ Networks and Packets - Part 1
- ▶ Servers and Protocols - Part 2
- ▶ Using HTTP in Python - Part 3
- ▶ Worked Example: Sockets
- ▶ Characters, ASCII, and Unicode - Part 4
- ▶ Using urllib in Python - Part 5
- ▶ Worked Example: Urllib
- ▶ Beautiful Soup in Python - Part 6
- ▶ Worked Example: BeautifulSoup



## Slides

- ▶ Pythonlearn-12-HTTP.pptx [↗](#)



## References

- ▶ Chapter 12: Networked Programs [↗](#)
- ▶ ASCII Coding and Binary [↗](#)
- ▶ Characters, Symbols and the Unicode Miracle - Computerphile [↗](#)

### • Discussions:

- ▶ Network Programming

### • Tools:

- ▶ Autograder: Request-Response Cycle
- ▶ Autograder: Scraping HTML Data with BeautifulSoup
- ▶ Autograder: Following Links with BeautifulSoup
- ▶ Quiz: HTTP

### **Free Engagement: Discussion Boards and Peer Feedback**

PY4E offers a discussion board for students to help each other and ask for feedback for their hands-on projects. Unfortunately, this is not really well done because there are a lot of students posting their full solutions which it's not helpful as a learner.

In the first lesson, there is a peer review system where you have to upload two screenshots of your installation of Python. You have to review at least two other submissions and give out a six-point grade. (See Figure 3) There is a lot of value to this system if some of the other lessons also incorporate peer review.

As a lesson plan, I would ask students to submit their code to Github for each lesson and ask for students to review at least two other submissions. Since no two students write the exact same code, as part of learning, when a student is asked to review and comment on other people's code, they also learn by giving meaningful feedback.



Figure 3: Peer reviews using uploaded screenshots

Welcome Tony Tam from Python for Everybody

### Peer Review: Installing and Running Python Screen Shots

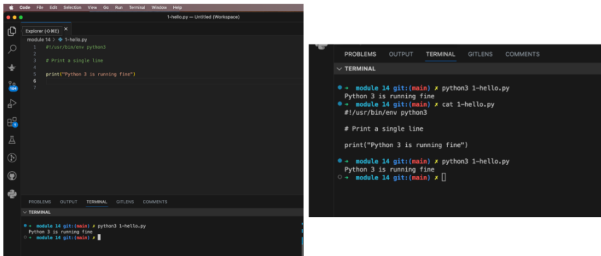
Install Python and a programming text editor and write a program that prints one line other than 'hello world', then take two screen shots and upload them below. You should use the command line to execute the Python program you wrote in the text editor. Please do *not* use the IDLE Python Shell, the Python Interpreter (>>>), or a shortcut in your text editor to run the code. Later in the class when we start reading files, we will need to be able to run Python programs from particular directories. See the videos for details.

This is a relatively simple assignment. The goal is to simply show that each student has Python installed on their desktop or laptop and can take screen shots. Please make your comments to help the student that you are reviewing.

[Review other students](#)

You have reviewed 4 other student submissions. You must review at least 2 submissions for full credit on this assignment. You *can* review up to 8 submissions if you like.

#### Your Submission:



Click on each image/pdf to see a larger view of the image.

You have the following grades from other students:

Points	Comments	Action
6	Good job!	<a href="#">Flag</a>
6		<a href="#">Flag</a>

Your overall score from your peers: 6

[Delete Your Submission](#)

Your current grade is 100%

### Free Assessment: Autograder and quizzes

For each lesson in PY4E, there are 10 multiple-choice questions that attempt to assess the student's understanding of the concepts of that lesson. The student can retake the quiz as many times as they would like, but they do have to wait at least 10 minutes to retake the entire quiz. As a professional, I went through every quiz and when I missed one or two questions, the quiz tells me which questions I answer incorrectly. The assessment tool for the quiz doesn't tell me what is a correct response which is better for the learner to figure out on their own. I find the delay helpful for me to take a breath and come back later and retake the entire quiz again. It was surprisingly motivating for me to come back again to try to get 100% on the 14 lessons.

As a lesson plan, I would ask my students to work on quizzes on their own and try multiple times in order to get a full grade. Mastery of the content is important in order to be prepared for the next lessons. Computer science is like math in the way that complicated concepts are built on top of foundational knowledge.

Finally, each lesson has one to three hands-on projects where the students are expected to modify or write code from scratch to accomplish a goal. For example, in lesson 13 with Network Programming, one of the hands-on projects is "*Scraping Numbers from HTML using BeautifulSoup: In this assignment you will write a Python program similar to <http://www.py4e.com/code3/urllink2.py>. The program will use urllib to read the HTML from the data files below, and parse the data, extracting numbers and compute the sum of the numbers in the file.*"

The student has to enter in a text box the sum of the numbers, and the code, and the Autograder will give the student 100% or 0%. The student can try as many times as they want.

What is really unique is the dataset the student uses is unique to them, so no two students will have the same answer for the sum of the numbers. (See Figure 4)

As the student progresses, they see their grade in the Assignment tab. Even though there isn't an LMS for the teacher to review progress, this is a personalized self-paced LMS. As a student, I find myself very motivated to get a perfect score on the coding and quizzes. (Figure 5)

For my lesson plan, I would ask students to share their assignment tab with me weekly via email so I can track their progress and help them to get unblocked.

Figure 4: Coding exercise and submission page

Your current grade on this assignment is: 100%

**Scraping Numbers from HTML using BeautifulSoup** In this assignment you will write a Python program similar to <http://www.py4e.com/code3/urllink2.py>. The program will use `urllib` to read the HTML from the data files below, and parse the data, extracting numbers and compute the sum of the numbers in the file.

We provide two files for this assignment. One is a sample file where we give you the sum for your testing and the other is the actual data you need to process for the assignment.

- **Sample data:** [http://py4e-data.dr-chuck.net/comments\\_42.html](http://py4e-data.dr-chuck.net/comments_42.html) (Sum=2553)
- **Actual data:** [http://py4e-data.dr-chuck.net/comments\\_1811344.html](http://py4e-data.dr-chuck.net/comments_1811344.html) (Sum ends with 87)

You do not need to save these files to your folder since your program will read the data directly from the URL. **Note:** Each student will have a distinct data url for the assignment - so only use your own data url for analysis.

#### Data Format

The file is a table of names and comment counts. You can ignore most of the data in the file except for lines like the following:

```
<tr><td>Modu</td><td><span class="comments">90</span></td></tr>
<tr><td>Kenzie</td><td><span class="comments">88</span></td></tr>
<tr><td>Hubert</td><td><span class="comments">87</span></td></tr>
```

You are to find all the `<span>` tags in the file and pull out the numbers from the tag and sum the numbers.

Look at the sample code provided. It shows how to find all of a certain kind of tag, loop through the tags and extract the various aspects of the tags.

```
...
# Retrieve all of the anchor tags
tags = soup('a')
for tag in tags:
    # Look at the parts of a tag
    print 'TAG:', tag
    print 'URL:', tag.get('href', None)
    print 'Contents:', tag.contents[0]
    print 'Attrs:', tag.attrs
```

You need to adjust this code to look for `span` tags and pull out the text content of the span tag, convert them to integers and add them up to complete the assignment.

#### Sample Execution

```
$ python3 solution.py
Enter - http://py4e-data.dr-chuck.net/comments_42.html
Count 50
Sum 2...
```

#### Turning in the Assignment

Enter the sum from the actual data and your Python code below:

Sum:  (ends with 87)

Python code:

Figure 5: Grades for quizzes and coding exercises

PY4E		Lessons	Discussions	Assignments	Instructor 	Book
<input checked="" type="checkbox"/>	Autograder: Exercise 3.1				Score: 100	
<input checked="" type="checkbox"/>	Autograder: Exercise 3.3				Score: 100	
<input checked="" type="checkbox"/>	Quiz: Conditional Execution				Score: 100	
Functions						
<input checked="" type="checkbox"/>	Autograder: Exercise 4.6				Score: 100	
<input checked="" type="checkbox"/>	Quiz: Functions				Score: 100	
Loops and Iterations						
<input checked="" type="checkbox"/>	Autograder: Exercise 5.2				Score: 100	
<input checked="" type="checkbox"/>	Quiz: Loops and Iterations				Score: 90	
Strings						
<input checked="" type="checkbox"/>	Autograder: Exercise 6.5				Score: 100	
<input checked="" type="checkbox"/>	Quiz: Strings				Score: 100	
Files						
<input checked="" type="checkbox"/>	Autograder: Exercise 7.2				Score: 100	
<input checked="" type="checkbox"/>	Quiz: Files				Score: 100	
Lists						
<input checked="" type="checkbox"/>	Autograder: Exercise 8.4				Score: 100	
<input checked="" type="checkbox"/>	Autograder: Exercise 8.5				Score: 100	
<input checked="" type="checkbox"/>	Quiz: Lists				Score: 100	
Dictionaries						
<input checked="" type="checkbox"/>	Autograder: Exercise 9.4				Score: 100	
<input checked="" type="checkbox"/>	Quiz: Dictionaries				Score: 90.9091	
Tuples						
<input checked="" type="checkbox"/>	Autograder: Exercise 10.2				Score: 100	
<input checked="" type="checkbox"/>	Quiz: Tuples				Score: 100	
Regular Expressions						
<input checked="" type="checkbox"/>	Autograder: Regular Expressions				Score: 100	
<input checked="" type="checkbox"/>	Quiz: Regular Expressions				Score: 100	
Network Programming						
<input checked="" type="checkbox"/>	Autograder: Request-Response Cycle				Score: 100	
<input checked="" type="checkbox"/>	Autograder: Scraping HTML Data with BeautifulSoup				Score: 100	
<input checked="" type="checkbox"/>	Autograder: Following Links with BeautifulSoup				Score: 100	
<input checked="" type="checkbox"/>	Quiz: HTTP				Score: 90.9091	

## **Conclusion**

PY4E offers a truly 100% free course in computer programming. The free and open content (videos, presentations, reference), interactive quizzes, coding platform, and assessment all make this a fantastic resource for students to learn to program in a self-paced and no-pressure environment. Even an experienced programmer like myself found the platform enjoyable to use to brush up on my coding skills. Charles Severance's commitment to a free and open education platform is very inspiring and has already educated thousands of children and adults.

## References

- Charles R. Severance Short Biography*. (n.d.). Dr. Chuck. Retrieved May 10, 2023, from <http://www.dr-chuck.com/dr-chuck/resume/bio.htm>
- Programming for Everybody (Getting Started with Python)*. (n.d.). Coursera. Retrieved May 10, 2023, from <https://www.coursera.org/learn/python?specialization=python>
- PY4E - Python for Everybody*. (n.d.). PY4E - Python for Everybody. Retrieved May 10, 2023, from <https://py4e.com>
- Severance, C. (n.d.). *TSUGI - Tsugi Framework for Building Learning Tools*. TSUGI - Tsugi Framework for Building Learning Tools. Retrieved May 10, 2023, from <https://www.tsugi.org/>