Course Syllabus

**Welcome to Using Python to Access Web Data,** taught by Charles Severance!

This course will show how one can treat the Internet as a source of data. We will scrape, parse, and read web data as well as access data using web APIs. We will work with HTML, XML, and JSON data formats in Python. This course will cover Chapters 11-13 of the textbook “Python for Everybody”. To succeed in this course, you should be familiar with the material covered in Chapters 1-10 of the textbook and the first two courses in this specialization. These topics include variables and expressions, conditional execution (loops, branching, and try/except), functions, Python data structures (strings, lists, dictionaries, and tuples), and manipulating files. This course covers Python 3.

**Participation Strategies**

Engaged learning looks different for everybody. In this course, we hope you will define your own measures of success and engage with the material in a way that best suits your needs. We recognize and celebrate the diverse ways learners engage in courses. As you go through this course, we hope you will reflect on your unique skills, needs, and aspirations, and engage in the course material in a way that aligns with your own goals. While the course provides time estimates for completion, you should feel empowered to engage in the material in whatever ways make sense to you.

**Course Schedule & Grading Policy**

|  |  |
| --- | --- |
| LTI Item: Peer Review: Installing and Running Python Screen Shots | 9% |
| Quiz: Regular Expressions | 7% |
| LTI Item: Extracting Data with Regular Expressions | 8% |
| Quiz: Networks and Sockets | 7% |
| LTI Item: Understanding the Request / Response Cycle | 8% |
| Quiz: Reading Web Data from Python | 7% |
| LTI Item: Scraping HTML Data with BeautifulSoup | 8% |
| LTI Item: Assignment: Following Links in HTML Using BeautifulSoup | 8% |
| Quiz: eXtensible Markup Language | 7% |
| LTI Item: Extracting Data from XML | 8% |
| Quiz: REST, JSON, and APIs | 7% |
| LTI Item: Extracting Data from JSON | 8% |
| LTI Item: Using the GeoJSON API | 8% |

You can see the grading breakdown below for each assignment:

**Ground Rules**

We expect everyone to be mindful of what they say and its potential impact on others. The goal is to have respectful discussions that do not violate the community space created for these conversations. Here are some productive ways to engage in this course:

* **Participate:** This is a community. Read what others have written and share your thoughts.
* **Stay curious:** Learn from experts and each other by listening and asking questions, not making assumptions.
* **Keep your passion positive:** When replying to a discussion forum post, respond with thoughts on what was said, not about the person who posted. Avoid using all caps, too many exclamation points, or aggressive language.
* **Acknowledge discomfort:** The topics discussed in this course might be challenging or hard to talk about. Stick with it and remember the benefits of having these tough conversations that surface from multiple perspectives.

We expect all learners to abide by our full [Learner Engagement Policy](https://docs.google.com/document/d/18Ucy-lT33FmkDDYdtxHEJ_nRytVLr_XGRVq2moAsPB4/edit). We will specifically be monitoring this course for language that could be considered inflammatory, incivil, racist, or otherwise unacceptable for this learning space, and we will remove language deemed such.

*Please note that external study groups on applications like WhatsApp are not affiliated or endorsed by the University of Michigan. We strongly discourage joining external groups and instead recommend interacting with your fellow learners within the platform. Please express caution if you do join or post any personal information in these forums or in these groups. These forums are publicly accessible and any information you post may be collected, published, or used in an exploitative manner (scams, etc).*

**Academic Honesty**

All submitted work should be your own and academic dishonesty is not allowed. Academic dishonesty can be defined as:

* Copying answers
* Copying words, ideas, or other materials from another source without giving credit to the original author
* Copying from your peers within the course
* Employing or allowing another person to alter or revise your work, and then submitting the work as your own

**Please don’t share or reuse solutions to assignments which is an academic integrity concern. Please do not:**

* Share complete assignment code in the course discussion forums
* Upload completed assignments to public websites with the goal of sharing solutions. (You can share your work and ideas for professional purposes though).
* Take a peer’s solution and submit it as your own

**Course Support**

Questions and discussion of course material should take place within the course itself. Please do not contact instructors or teaching assistants off the platform, as responding to individual questions is virtually impossible. We encourage you to direct your questions to the discussion forum, where your question might be answered by a fellow learner or one of our course team members. For technical help please contact the [Coursera Learner Help Center](https://learner.coursera.help/hc/en-us) or use the support forums.

**Accessibility**

We are committed to developing accessible learning experiences for the widest possible audience. We recognize that learners with disabilities (including but not limited to visual impairments, hearing impairments, cognitive disabilities, or motor disabilities) might need more specific accessibility-related support to achieve learning goals in this course.

Please use the [accessibility feedback form](https://forms.gle/sxxDQvstDYJF9PtR6) to let us know about any accessibility challenges such as urgent issues that keep you from making progress in the course (e.g., missing or inadequate alt-text, captioning errors).

**Diversity, Equity, Inclusion, and Justice**

**We welcome all learners to this course.** People like you are joining from all over the world and we value this diversity. We strive to create a community of mutual respect and trust, where people from all backgrounds, identities and views are valued and heard without the threat of bias, harassment, intimidation, or discrimination. We pay attention to your feedback, how different types of learners experience this course, and aim to make improvements so the course can best serve everyone. We hope you enjoy learning about topics that are important to you.

Python Regular Expression Quick Guide

Here is Dr. Chuck's RegEx "cheat sheet". You can also download it here:

<https://www.py4e.com/lectures3/Pythonlearn-11-Regex-Handout.txt>

|  |  |
| --- | --- |
| **^** | Matches the beginning of a line |
| **$** | Matches the end of the line |
| **.** | Matches any character |
| **\s** | Matches whitespace |
| **\S** | Matches any non-whitespace character |
| **\*** | Repeats a character zero or more times |
| **\*?** | Repeats a character zero or more times (non-greedy) |
| **+** | Repeats a character one or more times |
| **+?** | Repeats a character one or more times (non-greedy) |
| **[aeiou]** | Matches a single character in the listed set |
| **[^XYZ]** | Matches a single character *not* in the listed set |
| **[a-z0-9]** | The set of characters can include a range |
| **(** | Indicates where string extraction is to start |
| **)** | Indicates where string extraction is to end |

For more information about using regular expressions in Python, see <https://docs.python.org/3/howto/regex.html>

Welcome Yash Tank from Using Python to Access Web Data

Your current grade on this assignment is: 100%

**Finding Numbers in a Haystack**

In this assignment you will read through and parse a file with text and numbers. You will extract all the numbers in the file and compute the sum of the numbers.

**Data Files**

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/regex_sum_42.txt> (There are 90 values with a sum=445833)
* Actual data: <http://py4e-data.dr-chuck.net/regex_sum_1682994.txt> (There are 88 values and the sum ends with 887)

These links open in a new window. Make sure to save the file into the same folder as you will be writing your Python program. **Note:** Each student will have a distinct data file for the assignment - so only use your own data file for analysis.

**Data Format**

The file contains much of the text from the introduction of the textbook except that random numbers are inserted throughout the text. Here is a sample of the output you might see:

Why should you learn to write programs? 7746

12 1929 8827

Writing programs (or programming) is a very creative

7 and rewarding activity. You can write programs for

many reasons, ranging from making your living to solving

8837 a difficult data analysis problem to having fun to helping 128

someone else solve a problem. This book assumes that

everyone needs to know how to program ...

The sum for the sample text above is **27486**. The numbers can appear anywhere in the line. There can be any number of numbers in each line (including none).

**Handling The Data**

The basic outline of this problem is to read the file, look for integers using the **re.findall()**, looking for a regular expression of **'[0-9]+'** and then converting the extracted strings to integers and summing up the integers.

**Turn in Assignment**

Top of Form

Enter the sum from the actual data and your Python code below:  
Sum:  (ends with 887) 

Python code:

import re  
  
digits = 0  
count = 0  
list = list()  
with open('regex\_sum\_1682994.txt') as file:  
 for lines in file:  
 lines = lines.rstrip()  
 x = re.findall('[0-9]+', lines)  
 for i in x:  
 count += 1  
 digits += int(i)  
 list.append(i)  
  
print(digits, count)

Bottom of Form

**Optional: Just for Fun**

There are a number of different ways to approach this problem. While we don't recommend trying to write the most compact code possible, it can sometimes be a fun exercise. Here is a a redacted version of two-line version of this program using list comprehension:

Python 2

import re

print sum( [ \*\*\*\*\*\* \*\*\* \* in \*\*\*\*\*\*\*\*\*\*('[0-9]+',\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*.read()) ] )

Python 3:

import re

print( sum( [ \*\*\*\*\*\* \*\*\* \* in \*\*\*\*\*\*\*\*\*\*('[0-9]+',\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*.read()) ] ) )

Please don't waste a lot of time trying to figure out the shortest solution until you have completed the homework. List comprehension is mentioned in Chapter 10 and the **read()** method is covered in Chapter 7.

# If You Want to Learn More

This chapter covers networking at a very high level. If you want to learn more, there is both a free book and a Coursera course that I would recommend:

* [Introduction to Networking](http://www.net-intro.com/) (free textbook)
* [Internet History, Technology, and Security](https://www.coursera.org/learn/internet-history) (Coursera Course)

Neither of these is essential for this course or the Python Specialization as we quickly move from how the network works to how to write Python code using the urllib library - which makes the very complex Internet protocols exceedingly simple.

# Notes Regarding the Use of BeautifulSoup

The sample code for this course and textbook examples use BeautifulSoup to parse HTML. The examples in the textbook and in this class work with BeautifulSoup 3 and Python 3.8.

**Using BeautifulSoup 3 (only for Python 3.8 or Python 3.9)**

If you want use our samples "as is", download our Python 3 version of BeautifulSoup 3 from

<http://www.py4e.com/code3/bs4.zip>

You must unzip this into a "bs4" folder and have that folder as a sub-folder of the folder where you put our sample code like:

<http://www.py4e.com/code3/urllinks.py>

**Using BeautifulSoup 4 with Python 3.10**

Instructions for Windows 10:

* pip install beautifulsoup4
* if the bs4.zip file was downloaded, delete it

Instructions for MacOS:

* p​ip3 install beautifulsoup4
* if the bs4.zip file was downloaded or you have a bs4 folder, delete it