Extracting Data from XML

In this assignment you will write a Python program somewhat similar to https://py4e.com/code3/geoxml.py. The program will prompt for a URL, read the XML data from that URL using **urllib** and then parse and extract the comment counts from the XML data, compute the sum of the numbers in the file and enter the sum,

This course uses a third-party tool, Extracting Data from XML, to enhance your learning experience. The tool will reference basic information like your name, email, and Coursera ID.

I, **Ishaan Narula**, understand that submitting work that isn't my own may result in permanent failure of this course or deactivation of my Coursera account.

Learn more about Coursera's Honor Code

Open Tool

Done

Welcome Ishaan Narula from Using Python to Access Web Data

Extracting Data from XML

In this assignment you will write a Python program somewhat similar to http://www.py4e.com/code3/geoxml.py. The program will prompt for a URL, read the XML data from tha using **urllib** and then parse and extract the comment counts from the XML data, 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.xml (Sum=2553)
- Actual data: http://py4e-data.dr-chuck.net/comments_913449.xml (Sum ends with 78)

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 - only use your own data url for analysis.

Data Format and Approach

The data consists of a number of names and comment counts in XML as follows:

```
<comment>
  <name>Matthias</name>
  <count>97</count>
</comment>
```

You are to look through all the <comment> tags and find the <count> values sum the numbers. The closest sample code that shows how to parse XML is geoxml.py. But since nesting of the elements in our data is different than the data we are parsing in that sample code you will have to make real changes to the code.

To make the code a little simpler, you can use an XPath selector string to look through the entire tree of XML for any tag named 'count' with the following line of code:

```
counts = tree.findall('.//count')
```

Take a look at the Python ElementTree documentation and look for the supported XPath syntax for details. You could also work from the top of the XML down to the comments no and then loop through the child nodes of the comments node.

Sample Execution

```
$ python3 solution.py
Enter location: http://py4e-data.dr-chuck.net/comments_42.xml
Retrieving http://py4e-data.dr-chuck.net/comments_42.xml
Retrieved 4189 characters
Count: 50
Sum: 2...
```

Turning in the Assignment

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

Sum: (ends with 78) Submit Assignment

Python code:

Select Language ▼