Analyzing Structured Data in Python

WEEK 2: ELEMENTTREE

25 March 2022

CDCS Python Course Series

Instructor: Lucy Havens

RECAP: WHAT IS XML?

E**x**tensible **M**arkup **L**anguage

Similar to HTML (**H**ypertext **M**arkup **L**anguage)

Organizes data **hierarchically**

Reference: DataCamp's Python XML Tutorial with ElementTree: Beginner's Guide

XML VS. HTML

- XML provides data only
- XML used to *carry* data
- XML tag names are completely *customizable*

- HTML has information about how to *display* data
- HTML tag names are predefined

Reference: https://www.w3schools.com/xml/xml_whatis.asp

RECAP: XML IN THE REAL WORLD

- Web publishing
- Business applications (sending data between different technology systems)
- Digital metadata formats
- Harvesting data through APIs
- Downloading data dumps
- **Also: the Text Encoding Initiative (TEI)**
  - Standard approach to adding annotations to documents
  - Developed particularly for the humanities and social sciences

References: https://www.ibm.com/docs/en/i/7.3?topic=introduction-uses-xml, https://tei-c.org

ASSIGNMENT 2

Watch the videos below from LinkedIn Learning's "Python: XML, JSON, and the Web" course:

1.2 Quick Overview of XML:
https://www.linkedin.com/learning/python-xml-json-and-the-web/quick-overview-of-xml?u=50251009

6.3 The ElementTree API:
https://www.linkedin.com/learning/python-xml-json-and-the-web/the-elementtree-api?u=50251009

Complete the following online tutorial in your own Jupyter Notebook:

Turn XML data into CSV data: https://www.geeksforgeeks.org/xml-parsing-python/

Find or create your own XML file to parse and analyze with ElementTree!

What questions can you ask about it using the methods and functions in ElementTree?
Can you extract some of the XML data and put it in a DataFrame using Pandas?

If you're not sure where to find XML data, you could try one of these:

- *https://www.oldbaileyonline.org/browse.jsp?foo=bar&path=sessionsPapers/17800628.xml&div=t17800628-33&xml=yes*
- *sample.xml at https://data.mendeley.com/datasets/rth2kr5hxf/2*

Assignment 2

How did it go?

Were you able to parse your own XML data?

What data did you use?

USING AN API

Application Programming Interface (API): a way to access data, often through a URL (a web address)

ElementTree is an API

You can use other APIs with ElementTree (e.g., for Google Maps)

Sometimes you'll need to request a **key**, which you must include in the URL you use to obtain data; other times you won't need a key

USING A EUROPEANA API

Europeana provides many APIs: https://pro.europeana.eu/page/apis

- Some require a key: https://pro.europeana.eu/pages/get-api
- OpenSearch does not: https://pro.europeana.eu/page/search#opensearch

APIs can power some fun interfaces…

- https://culturepics.org/colour/#/

USING A EUROPEANA API

Let's look at XML data with a Europeana API:

https://api.europeana.eu/record/v2/opensearch.rss?searchTerms=blue&count=100&startIndex=100

- Must specify values for:
    - searchTerms
    - count
    - startIndex

LET'S CODE!

WHAT IF...

...we want to *change* XML data that we've parsed from an API or a file?

...we want to *create our own* XML data and save it to a file?

LET'S CODE!

WRITING EFFICIENT CODE

List Comprehension: a shorter, faster way to iterate than looping

```
t = []
for elem in root.iter():
  t.append(elem.tag)
```

```
t = [elem.tag for elem in root.iter]
```

WRITING EFFICIENT CODE

If you're looping through large amounts of data (i.e., high-resolution images), you can use a **generator** instead of list comprehension.

```
t = [elem.tag for elem in root.iter]
t = (elem.tag for elem in root.iter)
```

Generators create (and delete) items on the fly, rather than storing all items in memory simultaneously.

LIST COMPREHENSION OR GENERATOR?

Generators:

- For large data/lots of memory needed
- Returns an iterable, which isn’t mutable or indexable, can’t be sliced
- Good when you want to iterate over data *only once*

List Comprehension:

- When you want to iterate over data multiple times
- When you want to access and change the data iterated over
- Returns a list, which is mutable and indexable and can be sliced

Final Thoughts

Good programming is a balance of...

- Readability
  - Commenting your code (# like this)
  - Naming variables intuitively
  - Consistent conventions for naming variables, functions
- Efficiency
  - For you
  - For your machine's memory

MORE ON XML AND ELEMENTTREE

A Roadmap to XML Parsers in Python
https://realpython.com/python-xml-parser/#xmletreeelementtree-a-lightweight-pythonic-alternative

W3Schools XML Tutorial
https://www.w3schools.com/xml/default.asp

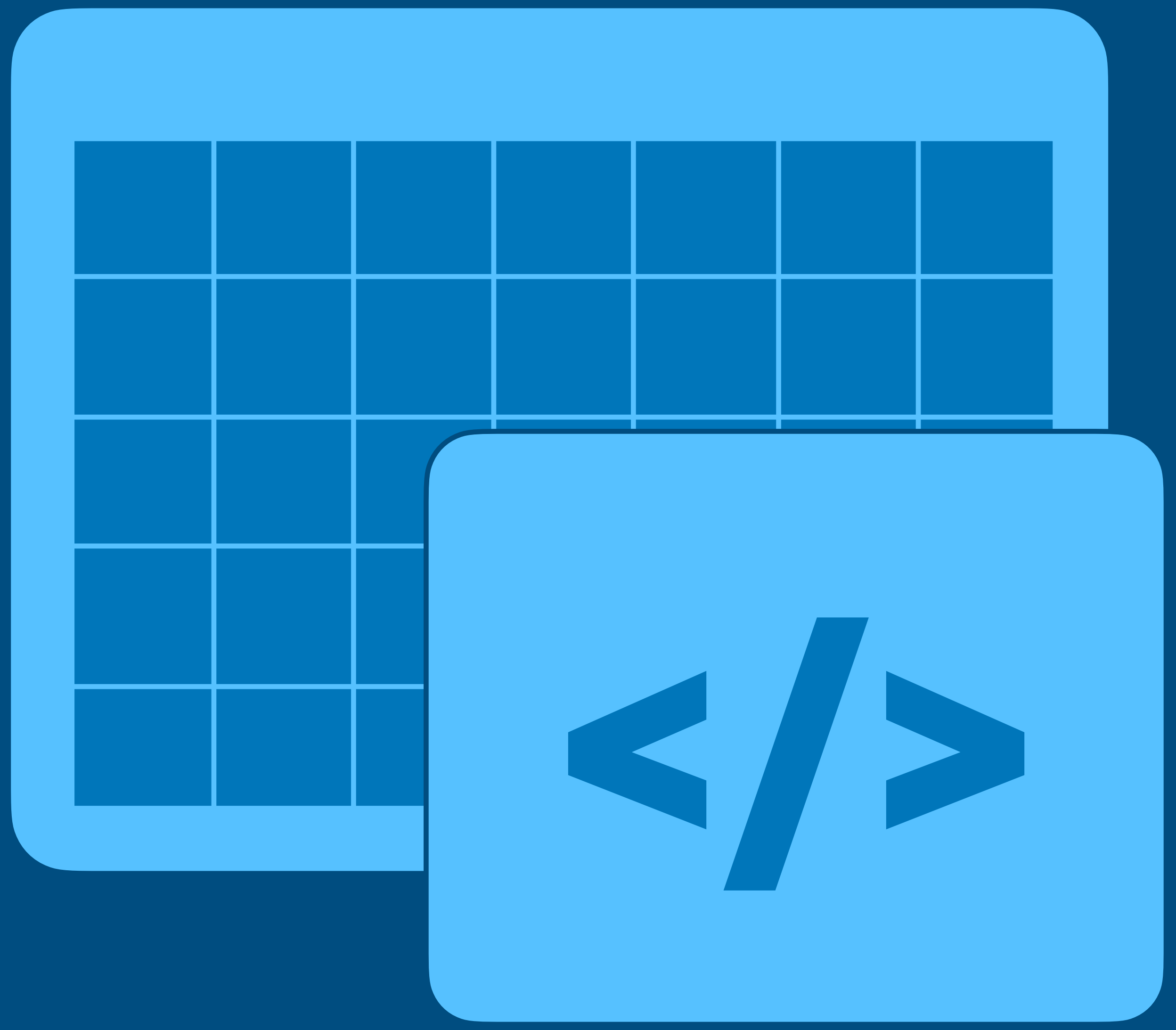
ElementTree Documentation
https://docs.python.org/3/library/xml.etree.elementtree.html

MORE PYTHON COURSES WITH THE CDCS

Machine Learning with Python
23 March - 13 April (2 days a week)

Text Analysis with Python's NLTK Library
11 April - 22 April (2 days a week)

The CDCS has many resources on it's website for courses and self-guided learning!



THANK YOU

I hope you enjoyed the course!

Feedback Survey: https://forms.office.com/r/YYNrqvuNr8