Pre-Assignment: (0 points)

For this assignment we will be performing data analysis. We will acquire the data by web scraping using XQuery and XPath. In order to begin this assignment it is necessary to set up Jupyter Notebook or get acquainted with Google Colab. Both of these tools are useful for performing data analysis. Additionally, it is important to install BaseX as it will be vital when developing and running queries.

Sub-Assignment 1:

First we will begin with understanding how XML documents are structured. XML documents are structured such that:

- Everything in it is a node
  - Elements are element nodes
  - Text is a text node
  - Attribute is an attribute node
  - Comments are comment nodes
- There is tree organization
  - It starts at the root and branches into leaves

You can learn more about XML structure here:

https://www.w3schools.com/xml/default.asp

XML and HTML are structured very similarly. So, if you have an understanding of the XML structure you will be able to pick up the HTML structure.

Both HTML and XML can be structured using a node tree. Using the terms parent, child and sibling, we can describe the relationship between the various nodes and develop the tree.

For the first sub assignment, choose a web page (a sub reddit, an ecommerce website, something with a lot of data, etc) and develop the tree structure. This should be done in an easy to understand format, so that we can easily see the structure of the web page. For this sub assignment please provide the URL in addition to the tree.

Sub Assignment 2:

For the second part of this assignment we will look at parsing an HTML file. The recommended library is lxml, as it is “easy-to-use” and “feature-rich”, as you can see if you take a look at their website :).

If you have the package installer you can easily install it: `pip install lxml`

Using the same web page you created a node tree for, begin parsing the HTML file using lxml. The best way to select specific information in an HTML or XML file is by using XPath.

Once you have successfully scraped the relevant data, make sure to print out all your data points.

Sub Assignment 3:

Now that you have relevant data, perform analysis of the data you have found. Create at least 5 visualizations of your data. If you do not have enough data for 5 visualizations, go back and scrape more data.

Make sure that these visualizations are accurate and convey meaningful information, as this is what you will be graded on for this section.

Sub-Assignment 4:

For the last part of this assignment we will look at XML files and XQuery. For this part of the project you will need to save an XML file from a web page. Some good options for possible XML files, are at the following links (you can use these or find your own):

- https://www.w3schools.com/python/python_booleans.asp
- https://www.javatpoint.com/selenium-tutorial
- Any wikipedia page

Create a sitemap using this resource: https://www.xml-sitemaps.com/ and then make sure to save the XML file.

When choosing a webpage for this part of the project, make sure that more than 1 page is indexed.

Now, we will begin querying the XML data. For the purpose of this project, write 5 queries for your XML data and provide the corresponding output. I recommend that you use BaseX to process your XQuery requests.

Useful links:

https://docs.python-guide.org/scenarios/scrape/

https://lxml.de/tutorial.html

https://tomassetti.me/parsing-html/

https://www.edureka.co/blog/html-vs-xml/

https://www.w3.org/TR/xquery-31/

https://www.xml-sitemaps.com/

https://basex.org/basex/xquery/

https://www.xml-sitemaps.com/download/www.javatpoint.com-da44bd744/sitemap.xml?view=1

https://data-lessons.github.io/library-webscraping-DEPRECATED/xpath/

https://towardsdatascience.com/how-to-use-python-and-xpath-to-scrape-websites-99eaed73f1dd