Workshop - Week 3: COMP20008

XML

1. What is the difference between XML and HTML? When would it be more appropriate to use XML instead of HTML? When would it be more appropriate to use HTML instead of XML?

2. Type in (or copy and paste) the following XML data using a text editor.

   ```
   <?xml version="1.0" encoding="utf-8"?>

   <queen title="Queen Elizabeth II" marriedTo="Philip, Duke of Edinburgh">
        <prince title="Charles, Prince of Wales" marriedTo="Lady Diana Spencer">
             <prince title="Prince William of Wales" />
             <prince title="Prince Henry of Wales" />
        </Prince>
        <princess title="Anne, Princess Royal" />
        <prince title="Andrew, Duke of York" />
        <prince title="Edward, Earl of Wessex" >
   </queen>
   ```

   Save the file and name it *royal.xml*. Load the file in the browser. Notice that the browser would display some errors. In fact, there are actually two syntax errors in the documents. The XML file is not well-formed. Find those errors and fix them. Save the file as *royal2.xml* (put in the same directory as *royal.xml*) and load it in the browser to check it works.

3. Examine the corrected file royal2.xml and answer the following questions:

   - Draw the XML tree that corresponds to this XML fragment
   - How many XML elements are there in the whole XML tree? What attributes belong to the first child of the root element? What are their values?
   - Why do you think title is an attribute and not an element? Under what circumstances would it be worthwhile to make it an element?
   - Why is prince an element and not an attribute? Are there circumstances where it would be worthwhile to make it an attribute?

4. Now load up the Jupyter notebook file *XMLandJSON-exercise.ipynb* and do exercises 5a) and 5b). Make sure the *royal2.xml* is in the same directory as the notebook file.

JSON

Consider the following description of a book using XML

```
<?xml version="1.0" encoding="utf-8"?>
<book id="book001">
   <author>Salinger, J. D.</author>
   <title>The Catcher in the Rye</title>
   <price>44.95</price>
   <language>English</language>
   <publish_date>1951-07-16</publish_date>
   <publisher>Little, Brown and Company</publisher>
   <isbn>0-316-76953-3</isbn>
   <description>A story about a few important days in the life of Holden Caulfield</description>
</book>
```

5. Represent the XML file as JSON. Create the new file, give it the name book.json, and save as 'Text'. Begin with the following text and expand from there:

   ```
   {
      "id": "book001",
      "author": "Salinger, J. D."
   }
   ```

   Validate your JSON solution against JSONLint.

6. In your JSON solution, add Spanish and German as two extra languages represented as an array. Save this file as *book2.json*. Validate it on JSONLint.

   Now modify the publish date parameter. Make this an array of two objects that have properties of edition (first, second) and date (1951-07-16,1979-01-01) respectively. Save this file as *book3.json*. Validate it on JSONLint.

Preprocessing

7. Download, open and study the file smoking_data_us_1995_2010-fixed.csv, showing United States population smoking data from 1995 to 2010. In the first twenty rows, there are seven errors. Identify the errors and suggest one plausable reason. Where possible fix the errors manually and save the new spreadsheet as *smoking-info-corrected.csv*

8. Suggest some tools and/or methods you could use to automatically fix the problems in the file above, assuming these types of errors occurred throughout the file rather than in just the first twenty rows.