Week 11 Commentary

Document the page you have developed including how the data is fed onto your site.

The page I have developed this week is a page that links to a page on the website OpenWeatherMap (which is a weather website that has lots of features and services with its weather data, as I'll elaborate on soon), and gets the weather data for a location (in my case I chose Arbroath, UK) and displays it in the units of your choice. As I already had an account with OpenWeatherMap, I signed in and created a new application, which generates a new API Key, which I knew would be needed for a new program call such as this.

After this had been created, I went to the page on their website which lists and explains how to use their API for whatever you need it for. From this, I was able to create a URL that specified my options, such as the Town whose weather I required, and the unit I would like it to be in, as by default it uses the Kelvin unit of measurements, which is not what I wanted to be used. I also had to paste in the API Key, which was created earlier. From there, I tested that this URL worked, which it did, and then amended the 'About Us' page on the CMSS website to include a link to view this data. Although the page does not update itself automatically, it will update with new weather information when the page is refreshed manually.

Advantages of XSLT

One advantage of XSLT is that is allows for the proper styling of XML files, such as the RSS Feed that I was able to style last week, which I did via XSLT (eXtensible Stylesheet Language Transformations). This means that for the average viewer of an RSS feed it is easy to read and not simply unformatted XML, which to the average person is not going to the most user friendly. By comparison, with a small amount of XSL added, it can be much easier to read, especially on mobile devices, where RSS is more commonly used.

Another advantage of XSLT is that it allows for a great amount of control over the functions of web technologies, such as the fact that it includes XPath. XPath allows for the developer to select nodes in an XML document, by which I mean an object like an element, attribute, or text etc. This allows the developer to in effect call parts of a web page or document and transpose this onto another document. This may seem like aspects of the Document Object Model, and to an extent it is, although whereas the DOM would require a method like `Document.getElementById()`, XPath is more universal, and works with a wider variety of languages including XML which is what has been used both in this and the previous week.