Styling XML with XSL

Author: Mihai Rotaru

Date: 1 Dec 2011

This blog post will describe the process of employing XSLT stylesheets for styling an XML document. It is very similar to using CSS - the stylesheet must be created, and the XML document should be told about the stylesheet. But that's where the similarities stop; the two methods are very different in their approach.

Central to XSLT-based styling is the concept of a 'template' - it is XSLT's equivalent for CSS rules. XSLT will try to match the template's pattern with the XML document; when it finds a match, it will perform the transformations described in the template body.

Here's an XSL which would display the movies.xml file as a table:

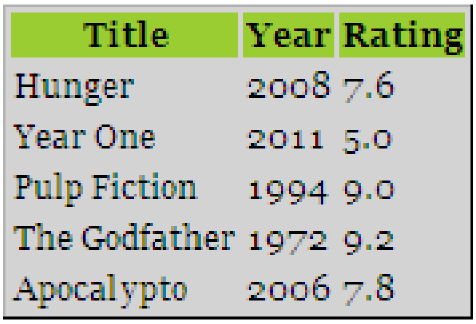
```
<?xml version="1.0"?>
<xsl:stylesheet
version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform" >
<xsl:template match="/">
<html>
<body style="font:12px Georgia;">
<table style="border: 1px outset; background-color: lightgray;">
<!--Generate table headings-->
<tr bgcolor="#9acd32">
<th>Title</th>
<th>Year</th>
<th>Rating</th>
</tr>

<!--Generate table rows-->
<xsl:for-each select="movies/movie">
<tr>
<td><xsl:value-of select="title"/></td>
<td><xsl:value-of select="year"/></td>
<td><xsl:value-of select="rating"/></td>
</tr>
</xsl:for-each>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

The XML file needs to know about this XSL stylesheet; so this line needs to be added to the XML file, assuming the XSL has the 'movies-xsl-style.xsl':

```
<?xml-stylesheet type="text/xsl" href="movies-xsl-style.xsl"?>
```

This is how the XML file was rendered by Firefox 8.0:



Title	Year	Rating
Hunger	2008	7.6
Year One	2011	5.0
Pulp Fiction	1994	9.0
The Godfather	1972	9.2
Apocalypto	2006	7.8

Which is exactly how I wanted it to look. This example illustrates the advantages and potential drawbacks of using XSL fro styling XML documents.

Among the drawbacks, the most obvious one is the learning curve - although XSL stylesheets are XML documents themselves, they have numerous tags with a specific role (more precisely, those inside the http://www.w3.org/1999/XSL/Transform namespace), which means that it is still a technology that one would need to spend considerable amounts of time to learn. It can be considered a fully-fledged programming language, with loops, if's and variables.

Browser support is also quite limited for XSLT 2.0; however, this doesn't diminish it's usefulness on the server side. Instead of serving the XML + XSL documents and relying on the client's browser to perform the transformations, the transformations could be run on the server (using tools such as Saxon or AltovaXML), generating XHTML files which would then be served to the clients.

One of the most important advantages of using XSL to style XML documents is the fact that it respects the principle of keeping the data and presentation separate, while providing the same styling capabilities as with CSS stylesheets, and much more powerful constructs.