

Hypermedia and the Web

May 21, 2003

XSLT – Extensible Stylesheet Language for Transformations

Compare/contrast with CSS:

CSS is used to change display characteristics of primarily HTML documents. But, CSS has limitations:

- CSS can't change the order of elements in a document
- CSS cannot perform computations (summing together multiple element values, for example)
- CSS cannot combine multiple documents

XSLT is a language for transforming XML documents into other forms.

Can transform XML into: HTML, XML, PDF, SVG, VRML, Java code, text files, etc.

Helps answer the question, how do I display XML? Convert to HTML, then view in a browser.

Related standard: XSL-FO – XML Stylesheet Language, Formatting Objects – a language for converting XML into page descriptions, such as PDF documents. Can be viewed as similar to Latex.

A few high-level observations:

XSLT stylesheets are XML documents – use XML to represent the XSLT language itself

XSLT uses rule-based pattern matching. XPath is used in the creation of the patterns. Typical form: if you see part of the document that looks like *pattern*, perform *action*.

Ways you might want to use XSLT:

- Convert XML into multiple display representations (for Web browser, handheld, cell phone, print)
- Data conversion for data interchange among different formats
- Have Web site content available in logical form, and have the actual Web site generated from the XML. More easily supports site redesign.
- Have documents in a relatively neutral format that is richer than text.

Hello World example (From XSLT, Doug Tidwell, O'Reilly, 2001)

Input XML:

```
<?xml version="1.0"?>
<greeting>
  Hello, World!
</greeting>
```

XSLT stylesheet:

```
<xsl:stylesheet
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  version="1.0">
  <xsl:output method="html"/>

  <xsl:template match="/">
    <xsl:apply-templates select="greeting"/>
  </xsl:template>

  <xsl:template match="greeting">
    <html>
      <body>
        <h1>
          <xsl:value-of select="."/>
        </h1>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```

Output:

```
<html>
<body>
<h1>
  Hello, World!
</h1>
</body>
</html>
```

Steps performed during the example:

- XSLT stylesheet is read and parsed
- XML source document is read and parsed into a tree structure
- XSLT processor is at the root of the source document (context set to root)
- Since there is a template that matches the document root, evaluate that template.
- The template's instruction is to evaluate any templates that apply to greeting elements
- There is one greeting element at the document root, so evaluate the second template instruction
- Elements that are not in the XSL namespace are passed through to the output (these are the HTML tags html, body, h1)
- The `xsl:value-of` puts into the output the value of the node matching the XPath expression, which in this case is the current context (i.e., the greeting node)
- Remaining HTML tags pass through to output
- No other templates match, are done.

Note:

`xsl:stylesheet` element provides `xsl` namespace declaration, and version of XSLT. XSLT processors complain if these are missing.

`xsl:output` describes which output format is being used (in this case HTML, but many other are possible – “xml” is a common one)

Built-in template rule:

```
<xsl:template match="*/">  
  <xsl:apply-templates/>  
</xsl:template>
```

Ensures that XML documents will be processed, even if your stylesheet only includes more specific matching rules, and omits a matching rule for the root element.