

Definition and some examples

# XSL

EXtensible Stylesheet Language

By **Hamid Moravi-Porari**

## **XSL - More than a Style Sheet Language**

The World Wide Web Consortium (W3C) started to develop XSL because there was a need for an XML-based style sheet language.

XSLT stands for XSL Transformations. In this tutorial you will learn how to use XSLT to transform XML documents into other formats, like XHTML. XSL describes how the XML document should be displayed!

XSL consists of three parts:

- XSLT - a language for transforming XML documents
- XPath - a language for navigating in XML documents
- XSL-FO - a language for formatting XML documents

### **What is XSLT?**

- XSLT stands for XSL Transformations
- XSLT is the most important part of XSL
- XSLT transforms an XML document into another XML document
- XSLT uses XPath to navigate in XML documents
- XSLT is a W3C Recommendation

### **XSLT = XSL Transformations**

XSLT is the most important part of XSL.

XSLT is used to transform an XML document into another XML document, or another type of document that is recognized by a browser, like HTML and XHTML. Normally XSLT does this by transforming each XML element into an (X)HTML element.

With XSLT you can add/remove elements and attributes to or from the output file. You can also rearrange and sort elements, perform tests and make decisions about which elements to hide and display, and a lot more.

A common way to describe the transformation process is to say that *XSLT transforms an XML source-tree into an HTML result-tree.*

### **XSLT Uses XPath**

XSLT uses XPath to find information in an XML document. XPath is used to navigate through elements and attributes in XML documents.

### **How does it Work?**

In the transformation process, XSLT uses XPath to define parts of the source document that should match one or more predefined templates. When a match is found, XSLT will transform the matching part of the source document into the result document.

Java API for XML <http://java.sun.com/xml>

To perform XSLT transformation we need:

- An XML data source, e.g. *discussionForumHome.xml*

```
<?xml version="1.0" encoding="UTF-8"?>
<discussionForumHome>
  <messageBoard id="1" name="Java Programming"/>
  <messageBoard id="2" name="XML Programming"/>
  <messageBoard id="3" name="XSLT Programming"/>
</discussionForumHome>
```

- An XSLT Stylesheet, e.g. *discussionForumHome.xslt*

```
<?xml version="1.0" encoding="UTF8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="html"/>
```

```
<!-- match the document root -->
<xsl:template match="/">
  <html>
    <head>
      <title> Discussion Forum Home page</title>
    </head>
    <body>
      <h1> Discussion Forum Home Page</h1>
      <h3> Please select a message board to view:</h3>
      <ul>
        <xsl:apply-templates select="discussionForumHome/messageBoard"/>
      </ul>
    </body>
  </html>
</xsl:template>
```

```
<!-- match a <messageBoard> element -->
<xsl:template match="messageBoard">
  <li>
    <a href="viewForum?id={@id}">
      <xsl:value-of select="@name"/>
    </a>
  </li>
</xsl:template>
```

</xsl:stylesheet>

- XSLT processor

Apache Xalan.jar and Xerces.jar have to be used for running XSLTs. Add these jar files to your CLASSPATH and run:

***Java org.apache.xalan.xslt.Process -IN discussionForumHome.xml -XSL discussionForumHome.xslt -OUT > outfile.html***

Output is outfile.html with following text:

```
<html>
<head>
META http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title> Discussion Forum Home page</title>
</head>
<body>
<h1> Discussion Forum Home Page</h1>
<h3> Please select a message board to view:</h3>
<ul>
<li>
<a href="viewForum?id=1">Java Programming</a>
</li>
<li>
<a href="viewForum?id=2">XML Programming</a>
</li>
<li>
<a href="viewForum?id=3">XSLT Programming</a>
</li>
</ul>
</body>
</html>
```