

XSL

In some ways, XSL (eXtensible Style Language) is to XML as CSS is to HTML. XSL provides a mechanism to specify how to render (display) the elements of an XML file. XSL goes beyond CSS in that it also provides mechanisms to choose & manipulate XML data. XSL transforms the XML data to another file format (html).

Creating an XSL file

Open a text document (notepad) type XSL instructions in that file. Save that file with .Xsl extension like f2.xsl

Example for an XSL file is given below

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
<xsl:template match="/">
* page code here *
</xsl:template>
</xsl:stylesheet>
```

The style sheet (XSL file) begins with an XML declaration and <xsl:stylesheet > tag

XSL - XML Declaration

You should include an XML declaration at the top of your XSL documents. The attribute *version* defines what version of XML you are using.

```
<?xml version="1.0" ?>
```

XSL - Stylesheet Root Element

Every XSL file must have the root element *xsl:stylesheet*. This root element has two attributes that must be included:

- **version** - the version of XSL
- **xmlns:xsl** - the XSL namespace, which is a URI to w3.org. That may be <http://www.w3.org/1999/XSL/Transform> OR <http://www.w3.org/TR/WD-xsl>

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

</xsl:stylesheet>
```

Importing XSL file into an XML file

xml-stylesheet

xml-stylesheet is a special declaration in XML for linking XML with stylesheets. Place this after your XML declaration to link your XML file to your XSL code.

xml-stylesheet has two attributes:

1. **type**: the type of file being linked to. We will be using the value *text/xsl* to specify XSL.
2. **href**: the location of the file. If you saved your XSL and XML file in the same directory, you

can simply use the XSL filename.

Let the following xml file is saved as class.xml. In this Xml file we imported one XSL file class.xsl. Make sure that both your XSL and XML file are in the same directory. If we open class.xml file using notepad content will be displayed as given below. If we open class.xml file using Internet Explorer content of the xml file will be displayed according to the instructions given in the Xsl file.

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="class.xsl"?>
<class>
  <student>Jack</student>
  <student>Harry</student>
  <student>Rebecca</student>
  <teacher>Mr. Bean</teacher>
</class>
```

Save this XML file as class.xml and your XML file will be linked to your XSL file. After you have finished this step you can continue learning XSL!

Working of XSL

The purpose of XSL is to take an XML document and transform it into something new. To transform an XML document, XSL must be able to do two things well:

1. Find information in the XML document
2. Add additional text and/or data

Both of these items are taken care of with the very important XSL element *xsl:template*.

xsl:template

Match Attribute of xsl:template

To find information in an XML document you will need to use xsl:template's *match* attribute. It is in this attribute that you use your knowledge of XPath to find information in your XML document.

We will be using above class.xml as our example XML document.

If we wanted to find *student* elements we would set the *match* attribute to a simple XPath expression: *student*.

Every time we find a student let's print something out. The text we want printed must go between the opening and closing tags of `<xsl:template>`. Let's have it print out "Found a learner!" The following XSL code will find *student* elements in the XML and output "Found a learner!" for each student element.

```
<?xml version="1.0" ?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="student">
    Found a learner!
  </xsl:template>
</xsl:stylesheet>
```

Save this file as class.xsl. Then open class.xml. You can see the output given below. This simple XSL does not work in Firefox because the output is not well-formed XML. However, Internet Explorer 6.0+ will view it just fine.

XSL Output:

Found a learner! Found a learner! Found a learner! Mr. Bean

Our XML document had three students and one teacher, so our XSL printed out "Found a learner!" three times, followed by the *unmatched* teacher element's text "Mr. Bean". In a later lesson we will show you how to filter out the unwanted text "Mr. Bean".

xsl:apply-templates

The file a2.xml given below uses stylesheet manager1.xsl for displaying the content of it. By changing the content of manager1.xsl different outputs will get. Outputs for different manager1.xsl files are given below.

A2.xml

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="manager1.xsl"?>
<company>
<name>Asme</name>
<manager>
    <name>phil</name>
    <title>Chief</title>
</manager>
<division>
    <manager>
        <name>james</name>
        <title>officer</title>
    </manager>
</division>
</company>
```

manager1.xsl

<pre><?xml version="1.0"?> <xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl"> <xsl:template match="/"> <html><xsl:apply-templates select="//manager"/> hai</html> </xsl:template> <xsl:template match="manager"> <p><xsl:value-of select="title"/> <xsl:value-of select="name"/></p> </xsl:template> </xsl:stylesheet></pre>	<p>output</p> <p>Chief phil</p> <p>officer james</p> <p>hai</p>
--	--

<?xml version="1.0"?>	output
-----------------------	---------------

<pre><xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl"> <xsl:template match="/"> <html><xsl:apply-templates select="company/manager"/> </html> </xsl:template> <xsl:template match="manager"> <p><xsl:value-of select="title"/> <xsl:value-of select="name"/></p> </xsl:template> </xsl:stylesheet></pre>	Chief phil
<pre><?xml version="1.0"?> <xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl"> <xsl:template match="/"> <html><xsl:apply-templates select="company/division/manager"/> </html> </xsl:template> <xsl:template match="manager"> <p><xsl:value-of select="title"/> <xsl:value-of select="name"/></p> </xsl:template> </xsl:stylesheet></pre>	output officer james
<pre><?xml version="1.0"?> <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"> <xsl:template match="manager"> <p><xsl:value-of select="title"/> <xsl:value-of select="name"/></p> </xsl:template> </xsl:stylesheet></pre>	Asme Chiefphil officerjames
<pre><?xml version="1.0"?> <xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl"> <xsl:template match="manager"> <p><xsl:value-of select="title"/> <xsl:value-of select="name"/></p> </xsl:template> </xsl:stylesheet></pre>	Output blank

To display the content of file a3.xml in table format, we can use XSL (we can use the file s2.xml given below to display the content of a3.xml in table format.)

A3.xml

```
<?xml version="1.0" encoding="ISO8859-1" ?>
<?xml-stylesheet type="text/xsl" href="s2.xml"?>
<CATALOG>
  <CD> <TITLE>Empire</TITLE> <ARTIST>Bob</ARTIST> </CD>
```

```

    <CD> <TITLE>Burles</TITLE> <ARTIST>Dylan</ARTIST></CD>
</CATALOG>

```

s2.xsl

```

<?xml version='1.0'?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
<xsl:template match="">
<html> <body> <table border="2" > <tr><th>Title</th> <th>Artist</th> </tr>
  <xsl:for-each select="CATALOG/CD">
    <tr> <td><xsl:value-of select="TITLE"/></td>
      <td><xsl:value-of select="ARTIST"/></td>
    </tr>
  </xsl:for-each>
</table> </body> </html>
</xsl:template>
</xsl:stylesheet>

```

S2.XSL

```

<?xml version='1.0'?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
<xsl:template match="">
  <html><body><table border="2" > <tr><th>Title</th><th>Artist</th></tr>
  <xsl:apply-templates select="CATALOG/CD"/>
  </table></body></html>
</xsl:template>

<xsl:template match="CATALOG/CD">
  <tr> <td><xsl:value-of select="TITLE"/></td>
    <td><xsl:value-of select="ARTIST"/></td>
  </tr>
</xsl:template>
</xsl:stylesheet>

```

OUTPUT

Title	Artist
Empire	Bob
Burles	Dylan

Example showing how the content of XML file is displayed in the form of unordered list**A1.xml**

```

<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="xmlstyle.xml"?>
<LANGLIST>
<TITLE>List of Items Important to Markup Languages</TITLE>
<TITLE1>Languages</TITLE1>
<LIST1>
<LANGUAGES>SGML</LANGUAGES>
<LANGUAGES>XML</LANGUAGES>
<LANGUAGES>HTML</LANGUAGES>
</LIST1>
<TITLE2>Other Support</TITLE2>
<LIST2>
<OTHER>DTD</OTHER>

```

```
<OTHER>DSSSL</OTHER>
<OTHER>Style Sheets</OTHER>
</LIST2>
</LANGLIST>
```

xmlstyle.xsl

```
<?xml version="1.0"?>
```

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
```

```
<xsl:template match="/">
<xsl:apply-templates select="LANGLIST/TITLE" />
<xsl:apply-templates select="LANGLIST/TITLE1" />
<xsl:apply-templates select="LANGLIST/LIST1" />
<xsl:apply-templates select="LANGLIST/TITLE2" />
<xsl:apply-templates select="LANGLIST/LIST2" />
</xsl:template>
```

```
<xsl:template match="TITLE">
    <b><u><xsl:value-of /></u></b><BR/>
```

```
</xsl:template>
```

```
<xsl:template match="TITLE1">
    <b><xsl:value-of /></b><BR/>
```

```
</xsl:template>
```

```
<xsl:template match="LIST1">
    <UL style=" color: '#000000'; font-size: '18'">
    <xsl:for-each select="LANGUAGES">
        <LI style=" list-style-type: 'square'; font-size: '16'"><xsl:value-of /></LI>
    </xsl:for-each>
    </UL>
```

```
</xsl:template>
```

```
<xsl:template match="TITLE2">
    <xsl:value-of /><BR/>
```

```
</xsl:template>
```

```
<xsl:template match="LIST2">
    <i><UL><xsl:for-each select="OTHER">
        <LI> <xsl:value-of /> </LI>
    </xsl:for-each>
    </UL></i>
```

```
</xsl:template>
```

```
</xsl:stylesheet>
```

output

List of Items Important to Markup Languages

- SGML
- XML
- HTML

Other Support

- *DTD*
- *DSSSL*
- *Style Sheets*

xsl:value-of

Although using XSL to print out static messages is fun to play around with, using an XML document's elements and their contents is probably more useful in real world use. This section will teach you how to get the value of elements and attributes in an XML document using the *xsl:value-of* element.

XML Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="class.xsl"?>
<class>
    <student>Jack</student>
    <student>Harry</student>
    <student>Rebecca</student>
    <teacher>Mr. Bean</teacher>
</class>
```

XSL - Element Values

Our XSL code has been outputting "Found a learner!" for each *student* element found. Let's replace that string with the student's actual name, by using *xsl:value-of* to get the element's contents.

XSL Code:

```
<?xml version="1.0" ?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="class">
    <html> <body>
        <xsl:apply-templates select="student"/>
    </body> </html>
</xsl:template>
<xsl:template match="student">
    <p> <xsl:value-of select="."/> </p>
</xsl:template>
</xsl:stylesheet>
```

xsl:value-of uses the *select* attribute to choose elements. Because we are already at the correct element, *student*, we use a period "." to select the current element's data. The period is a special character to use when you want to select the same element that you matched with template's *match* attribute.

If you were outputting the XSL to a file, this is what the output would look like.

XSL Output:(is output of this HTML file)

```
<html>
<body>
<p>Jack</p>
<p>Harry</p>
<p>Rebecca</p>
</body>
</html>
```

XSL - Attribute Values

xsl:value-of can also be used to retrieve the value of attributes of XML elements. We have slightly modified our XML document to include an *id* attribute,

XML Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="class.xsl"?>
<class>
    <student id="1">Jack</student>
    <student id="2">Harry</student>
    <student id="3">Rebecca</student>
    <teacher id="1">Mr. Bean</teacher>
</class>
```

Let's output each student's id number before their name to make a nice list. To select an attribute use the at sign "@" followed by the name of the attribute.

XSL Code :

```
<?xml version="1.0" ?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="class">
    <html> <body>
    <xsl:apply-templates select="student"/>
    </body> </html>
</xsl:template>
<xsl:template match="student">
    <p> <xsl:value-of select="@id"/> - <xsl:value-of select="."/> </p>
</xsl:template>
</xsl:stylesheet>
```

The XSL output is shown below:

XSL Output:

<pre><html> <body> <p>1 - Jack</p> <p>2 - Harry</p> <p>3 - Rebecca</p> </body> </html></pre>	<pre>1 - Jack 2 - Harry 3 - Rebecca</pre>
--	---

Xml file

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/xsl" href="simple.xsl"?>
<breakfast_menu>
<food>
    <name>Belgian Waffles</name>
    <price>$5.95</price>
    <description>
        two of our famous Belgian Waffles
    </description>
    <calories>650</calories>
```



```

</food>
</breakfast_menu>
Simple.xsl.
<?xml version="1.0" encoding="ISO-8859-1" ?>
<xsl:version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" >
< html><body>
<xsl:for-each select="breakfast_menu/food">
    [<xsl:value-of select="name" />    <xsl:value-of select="price" /> ],
    <xsl:value-of select="description" />
    <span style="font-style:italic">
    (    <xsl:value-of select="calories" />    calories per serving)
    </span>
</xsl:for-each>
</body>
</html>

```

Output

[Belgian Waffles\$5.95], two of our famous Belgian Waffles (650 calories per serving)

- 1) Explain how we can display the content of an XML file using XSL
- 2) Explain <xsl:template>, <xsl:value-of> and <xsl:apply-templates> tags
- 3) Explain the basics of XSL language
- 4) Write the difference between <xsl:value-of select="description" /> and <xsl:value-of select="@id"/>
- 5) Explain how XML documents can be embedded into HTML document.
- 6) Briefly explain XSL style sheets.
- 7) Describe in detail the principles of displaying XML Data in HTML browser as HTML tables.
- 8) Explain the process of converting XML to HTML with XSL minimalist