Birkbeck
UNIVERSITY OF LONDON

XML

Session 7

Extensible Stylesheet Language Transformations (XSLT)

Introduction and use of XSLT

Birkbeck UNIVERSITY OF LONDON

- It is an XML language used to transform an XML document into another format
- Other formats include:
  - HTML
  - Another XML markup family, e.g. MathML
  - Text
  - When the output is HTML or XML, you can combine it with CSS for further presentation fine-tuning

XSLT – When to use it

Birkbeck UNIVERSITY OF LONDON

- When a document is stored in one format and needs to be converted to another.
  - Transform an XML document to an HTML document
  - Transform an XML document to a PDF document
- When you need to compact a document
- When you want to remove unnecessary markup not required for processing by your application
- Use as a front end to query a database.
  A script can generate requests that the database responds to. The stylesheet can then transform the results of the query into HTML.

Mechanism of an XSL processor

Birkbeck UNIVERSITY OF LONDON

- The transformation of an XML document is done through a “XSL processor”
- Most modern browsers have built-in processors
  - Other browsers can use JavaScript
- Programming languages that support XML such as Java, C#, C++, PHP have a built-in XSL processor.

XSLT – Processing

- XSLT views an XML document as a “tree” structure of nodes.
- In XSLT elements, attributes, processing instructions, comments and text are considered as nodes.
- Each match of a node is called the “context node”.

XSLT syntax structure

- XSLT is written using XML and must have a root element
- The correct way to declare an XSL stylesheet according to the W3C XSLT Recommendation is:

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

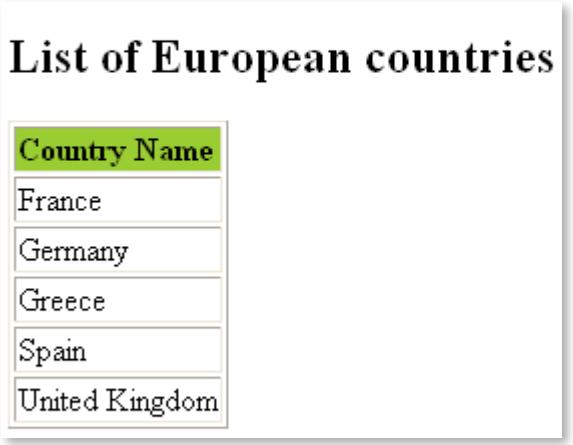
or

```
<xsl:transform version="1.0“ xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
```

- The root element must include the correct XSLT namespace URI and use the standard “xsl” prefix
- The root element contains the XSL instructions for the output of the XML document

Example XML and outcome

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl"
href= "Country.xsl"?>
<europe>
   <country>France</country>
   <country>Germany</country>
   <country>Greece</country>
   <country>Spain</country>
   <country>United Kingdom</country>
</europe>
```



Filename: Country.xml

Example XSLT

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
    <xsl:template match="/">
        <html>
            <head>
                <link rel="stylesheet" type="text/css" href="Country.css"/>
            </head>
            <body>
                <h2>List of European countries</h2>
                <table>
                    <tr class="header">
                        <th>Country Name</th>
                    </tr>
                    <xsl:for-each select="europe/country">
                        <tr>
                            <td>
                                <xsl:value-of select="."/>
                            </td>
                        </tr>
                    </xsl:for-each>
                </table>
            </body>
        </html>
    </xsl:template>
</xsl:stylesheet>
```

Filename: Country.xsl

XSLT Templates

- Definition and purpose in XSLT
  - XSL templates match the elements that must make up the output

    Example: <xsl:template match="root">

    Specifying a template in the output is done as follows:

    <xsl:apply-templates select="body">

  - XSL Templates contains the markup and/or text that fits with the XML to output

    Example:

```
<xsl:template match="table">
   <p>This table contains <strong><xsl:value-of select = "count(//*/row)" />
   </strong> rows.</p>
</xsl:template>
```

- Using <xsl:template match="/"> and <xsl:apply-templates>

  - Template rules are modules that describes how a particular part of your XML should be output.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
    <xsl:template match="/">
        <html>
            <body>
                <xsl:apply-templates/>
            </body>
        </html>
    </xsl:template>
</xsl:stylesheet>
```

  - To create a template rule:
    - a) Type <xsl:template to begin the template rule.
    - b) Type match="/" where "/" is a pattern that identifies the section of the XML document to which the template may be applied (in our example the root and hence the whole document).
    - c) Specify what should happen when a node is found that matches the pattern – in this instance we specify the <xsl:apply-templates/> which means apply all templates to the current element or the current element's child nodes.

- You can further modularise your XML by declaring other templates for various nodes like the following example:

```
<?xml version="1.0" encoding=”UTF-8"?>
<xsl:stylesheet version="1.0“ xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
    <html>
        <body>
            <h2>My CD Collection</h2>
        <xsl:apply-templates/>
        </body>
    </html>
</xsl:template>
<xsl:template match="cd">
    <p>
        <xsl:apply-templates select="title"/>
        <xsl:apply-templates select="artist"/>
    </p>
</xsl:template>
<xsl:template match="title">
    Title: <span style="color:#ff0000">
    <xsl:value-of select="."/></span> <br />
</xsl:template>
<xsl:template match="artist">
    Artist: <span style="color:#00ff00”><xsl:value-of select="."/></span> <br />
</xsl:template>
</xsl:stylesheet>
```

http://www.w3schools.com/Xsl/xsl_apply_templates.asp

XSLT – Output control

- Templates control the output of the XSL processor
- Other XSLT elements can further control how source XML is output
  - Calculated values
  - Conditional testing
  - Element/attribute creation
  - Sorting
  - Default values, parameters and constants

XSLT - Declaring an Element

- Calculated values

  <xsl:value-of select="heading"/>
  Gets the value of a node or calculates the XPath expression.

- Conditional testing

  <xsl:if test="…">
  Content is output if test is true.

- Element/attribute creation

  <xsl:element name="…">
  Adds element to output tree

- Declaring an Attribute

  <xsl:attribute name="…">
  When used in conjunction with <xsl:element>, this creates an attribute for that element
  The content of this element creates the attribute value.

- Sorting

  <xsl:sort select="…" order="ascending|descending">
  Sort node set using specified order, often used in conjunction with <xsl:for-each> and <xsl:apply-templates>

- Default values

  <xsl:param name="…" value="…">
  Sets default value for a template or an external application can set this parameter value

- Parameters

  Use <xsl:value-of select="$variableName"/> to extract the value inside a template

- Constants

  <xsl:variable name="…" value="…">
  Define a constant value for use elsewhere in the document (usually in a descendent template)

- Others

  <xsl:choose>

  List of choices evaluated in sequence specified by <xsl:when test="…"> elements

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:param name="browser"/>
<xsl:choose>
   <xsl:when test="$browser='ie'">
      <!-- Do something here -->
   </xsl:when>
  <xsl:when test="$browser='mo'">
      <!-- Do something else here -->
  </xsl:when>
</xsl:choose>
 </xsl:stylesheet>
```

- <xsl:output method="html|xml|text" omit-xml-declaration="yes|no"/>
  Instructs what format processor must use for output
  Specify XML if you require XHTML (HTML output can be pre HTML 4 format)

- <xsl:for-each select="…">
  Matches in sequence each element in a list of the same name (node set)

XSLT templates programming syntax for outputting

- <xsl:template match="/">
  Match root node of XML source
- <xsl:template match="*">
  Match all elements
- <xsl:template match="@*">
  Match all attributes
- <xsl:value-of select="."/>
  Output the value of the current node
- <xsl:template match="text()">
  Match text nodes
- <xsl:for-each match="node()">
  Match each child node in sequence
- <xsl:value-of select="@attributeName"/>
  Output the specified attribute's value
- {@attributeName}
  Shortcut to output the value of a specified attribute