CS 104: Web Technology - II Assignment #4

Anup Adhikari
anup.adhikari@gandakiuniversity.edu.np

Gandaki University January 19, 2023

1 Introduction to XSLT

XSLT (eXtensible Stylesheet Language Transformations) is a language used to transform XML documents into other formats, such as HTML or XHTML. It is a W3C recommendation and is a part of the XSL (eXtensible Stylesheet Language) specification. XSLT is used to transform XML documents into other formats, such as HTML or XHTML, for the purpose of displaying the data on the web or in other applications.

1.1 Use of XSLT

1. It can be used to transform XML data into HTML, PDF, or any other format that supports XML. XSLT is also used for transforming XML data into other types of data, such as JSON, CSV, and more.
2. Another common use of XSLT is in the creation of reports or other types of data visualization. For example, an XSLT style sheet can be used to transform an XML document containing financial data into an HTML table, making it easy to view and analyze the data.
3. XSLT is also used in data integration and data exchange, where data from different sources needs to be transformed into a common format. It can be used to transform data from one XML schema to another, or to convert data from one application's data format to another.
4. XSLT is also used in the development of web services, to transform the XML data sent and received by the service into more human-readable formats.

Overall, XSLT allows us to transform XML documents into other formats, making the data more accessible and usable for a variety of purposes.

1.1.1 Using XSLT in XML Documents

XSLT documents will render in an appealing format if used wisely.

basic.xml

```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" href="basic.xsl"?>
<data>
</data>
```

basic.xsl

```
<?xml version="1.0" ?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
  <xsl:template match="/">
    <html>
    <body><h1>Hello</h1></body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```

1.2 XSL Elements

The XSLT namespace has the URI http://www.w3.org/1999/XSL/Transform.

XSLT processors must use the XML namespaces mechanism [XML Names] to recognize elements and attributes

1.2.1 Stylesheet Element

```
<xsl:stylesheet
    id = id
    extension-element-prefixes = tokens
    exclude-result-prefixes = tokens
    version = number>
    <!-- Content: (xsl:import*, top-level-elements) -->
  </xsl:stylesheet>
```

A stylesheet is represented by an xsl:stylesheet element in an XML document. xsl:transform is allowed as a synonym for xsl:stylesheet.

Literal Result Element as Stylesheet

A simplified syntax is allowed for stylesheets that consist of only a single template for the root node.

```
<html xsl:version="1.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    xmlns="http://www.w3.org/TR/xhtml1/strict">
<head>
  <title>Expense Report Summary</title>
</head>
<body>
  <p>Total Amount: <xsl:value-of select="expense-report/total"/></p>
</body>
</html>
```

has the same meaning as:

```
<xsl:stylesheet version="1.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    xmlns="http://www.w3.org/TR/xhtml1/strict">
<xsl:template match="/">
<html>
<head>
<title>Expense Report Summary</title>
</head>
<body>
<p>Total Amount: <xsl:value-of select="expense-report/total"/></p>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

A literal result element that is the document element of a stylesheet must have an xsl:version attribute, which indicates the version of XSLT that the stylesheet requires.

1.2.2 Understanding the Template Rules

<xsl:template> Element

The <xsl:template> element is used to build templates.

The match attribute is used to associate a template with an XML element. The match attribute can also be used to define a template for the entire XML document. The value of the match attribute is an XPath expression (i.e. match="/" defines the whole document).

<xsl:value-of> Element

The <xsl:value-of> element can be used to extract the value of an XML element and add it to the output stream of the transformation:

```
td><xsl:value-of select="catalog/cd/title"/></td>
<td><xsl:value-of select="catalog/cd/artist"/></td>
```

<xsl:for-each> Element

The XSL <xsl:for-each> element can be used to select every XML element of a specified node-set:

```
<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>
```

<xsl:sort> Element

To sort the output, simply add an <xsl:sort> element inside the <xsl:for-each> element in the XSL file:

```
<td><xsl:value-of select="catalog/cd/title"/></td>
<td><xsl:value-of select="catalog/cd/artist"/></td>
```

<xsl:if> Element

To put a conditional if test against the content of the XML file, add an <xsl:if> element to the XSL document.

```
<xsl:if test="expression">
  ...some output if the expression is true...
</xsl:if>
```

```
<xsl:for-each select="catalog/cd">
    <xsl:sort select="artist"/>
    <tr>
        <td><xsl:value-of select="title"/></td>
        <td><xsl:value-of select="artist"/></td>
    </tr>
</xsl:for-each>
```

<xsl:choose> Element

The <xsl:choose> element is used in conjunction with <xsl:when> and <xsl:otherwise> to express multiple conditional tests.

```
<?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>
  <table border="1">
    <tr bgcolor="#9acd32">
      <th>Title</th>
      <th>Artist</th>
```

```
    </tr>
    <xsl:for-each select="catalog/cd">
    <tr>
      <td><xsl:value-of select="title"/></td>
—>  <xsl:choose>
        <xsl:when test="price &gt; 10">
          <td bgcolor="#ff00ff">
          <xsl:value-of select="artist"/></td>
        </xsl:when>
        <xsl:otherwise>
          <td><xsl:value-of select="artist"/></td>
        </xsl:otherwise>
—>  </xsl:choose>
    </tr>
    </xsl:for-each>
  </table>
  </body>
  </html>
</xsl:template>

</xsl:stylesheet>
```

<xsl:apply-templates> Element

The <xsl:apply-templates> element applies a template to the current element or to the current element's child nodes.

If we add a "select" attribute to the <xsl:apply-templates> element, it will process only the child elements that matches the value of the attribute. We can use the "select" attribute to specify in which order the child nodes are to be processed.

2 Lab Objectives

The Objectives of lab are:

1. To understand the basics of XSLT document to transform a XML document to the valid HTML document
2. To understand the basic template Rules

3 Examples

3.1 Simple XML and XSLT document

Major steps to transform XML into XHTML using XSLT.

1. Correct Style Sheet Declaration

```
<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">
```

2. Start with a Raw XML Document
3. Create an XSL Style Sheet (use .xsl extension)
4. Link the XSL Style Sheet to the XML Document

XML Document

data.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<catalog>
    <cd>
    <title>Empire Burlesque</title>
    <artist>Bob Dylan</artist>
    <country>USA</country>
    <company>Columbia</company>
    <price>10.90</price>
    <year>1985</year>
    </cd>
    <cd>
    <title>Hide your heart</title>
    <artist>Bonnie Tyler</artist>
    <country>UK</country>
    <company>CBS Records</company>
    <price>9.90</price>
    <year>1988</year>
    </cd>
    <cd>
    <title>Greatest Hits</title>
    <artist>Dolly Parton</artist>
    <country>USA</country>
    <company>RCA</company>
    <price>9.90</price>
    <year>1982</year>
    </cd>
    <cd>
    <title>Still got the blues</title>
    <artist>Gary Moore</artist>
    <country>UK</country>
    <company>Virgin records</company>
    <price>10.20</price>
    <year>1990</year>
    </cd>
</catalog>
```

XSLT Code

```xml
<?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>
    <table border="1">
    <tr bgcolor="#9acd32">
        <th style="text-align:left">Title</th>
        <th style="text-align:left">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>
```

Src: https://www.w3schools.com/xml/tryxslt.asp?xmlfile=cdcatalog&xsltfile=cdcatalog

4 Questions

Question 1

1. Create an xml document "data.xml" consisting of names of fruit items with price.
2. Create an XSLT stylesheet called "transform.xsl" to display it in a table containing columns, serial no, Fruits Name, Price