

XML- eXtensible Markup Language

mariaiuliana.dascalu@gmail.com

University POLITEHNICA of Bucharest
Department of Engineering in Foreign Languages

“It’s not XML that’s in decline. It’s the knowledge about XML that is in decline...”

<https://blog.wimtenbrink.nl/2014/02/21/is-xml-in-decline/>

Agenda

- XSL (the eXtensible Stylesheet Language)
- XSLT (XSL Transformation)
- XPath
- XML Namespaces

XML CSS

- It is possible to use CSS to format an XML document.

```
CATALOG
{
background-color: #ffffff;
width: 100%;
}
CD
{
display: block;
margin-bottom: 30pt;
margin-left: 0;
}
TITLE
{
color: #FF0000;
font-size: 20pt;
}
ARTIST
{
color: #0000FF;
font-size: 20pt;
}
COUNTRY, PRICE, YEAR, COMPANY
{
display: block;
color: #000000;
margin-left: 20pt;
}
```

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/css" href="cd_catalog.css"?>
<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>
  .
  .
  .
</CATALOG>
```

The result

Empire Burlesque Bob Dylan

USA
Columbia
10.90
1985

Hide your heart Bonnie Tyler

UK
CBS Records
9.90
1988

Greatest Hits Dolly Parton

USA
RCA
9.90
1982

XSL

- XSL is the preferred style sheet language of XML.
- XSL (the eXtensible Stylesheet Language) is far more sophisticated than CSS

<?xml version="1.0" encoding="ISO-8859-1"?>

<?xml-stylesheet type="text/xsl" href="simple.xml"?>

- **XSL** is an XML-based language used for stylesheets that can be used to transform XML documents into other document types and formats.
- **XSL** is a family of recommendations for defining XML document transformation and presentation.
- It consists of three parts.

XSL parts

- XSL Transformations ([XSLT](#))
 - a language for transforming XML
- XML Path Language ([XPath](#))
 - an expression language used by XSLT to access or refer to parts of an XML document. (XPath is also used by the [XML Linking](#) specification)
- XSL Formatting Objects ([XSL-FO](#))
 - an XML vocabulary for specifying formatting semantics

XSLT

- XSLT became a W3C Recommendation 16. November 1999.
- most important part of XSL
- transforms input document (*source tree*) into a particular way in a specified output document (*result tree*).
- built on a structure known as an XSL template:
- Example:

```
<xsl:template match="/movie/title">
```

```
  <xsl:value-of select="."/>
```

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

- selects one/all movies
- check first the root template; if that doesn't match, the next, etc.

XSLT Browsers

- nearly all major browsers support XML and XSLT.
- **Mozilla Firefox**
 - supports XML, XSLT, and XPath from v 3.
- **Google Chrome**
 - supports XML, XSLT, and XPath from v 1.
- **Opera**
 - \geq v 9, XML, XSLT (and CSS). V 8 only XML + CSS.
- **Apple Safari**
 - Safari supports XML and XSLT from v 3.
- **Internet Explorer**
 - \geq v 6, XML, Namespaces, CSS, XSLT, and XPath. V 5 **NOT !** compatible

XSLT Elements in common use

- **<xsl:stylesheet>**
used as root element of nearly all XSLT stylesheets.
- **<xsl:stylesheet version="version number" xmlns="path to W3C namespace">**
- Version number: current number of XSLT specification from [W3C](http://www.w3.org/1999/XSL/Transform).
- xmlns is the path to the XML Namespace defined by the W3C for the XSLT Transformation language. Currently, that path is <http://www.w3.org/1999/XSL/Transform>.

Example Correct Style Sheet Declaration

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

```
▼<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>Unchain my heart</title>
    <artist>Joe Cocker</artist>
    <country>USA</country>
    <company>EMI</company>
    <price>8.20</price>
    <year>1987</year>
  </cd>
</catalog>
```

The <xsl:template> Element

- An XSL style sheet consists of one or more set of rules that are called templates.
- A template contains rules to apply when a specified node is matched.
- The value of the match attribute is an XPath expression.
- The **match="/"** attribute associates the template with the root of the XML source document.
- Example:

```
<xsl:template match="/">
  <html>
    <body> <h2>My CD Collection</h2>
      <table border="1"> <tr bgcolor="#9acd32"> <th>Title</th>
        <th>Artist</th> </tr> <tr> <td>.</td> <td>.</td> </tr> </table>
    </body>
  </html>
</xsl:template>
```

Result:

My CD Collection

Title	Artist
.	.

The <xsl:value-of> Element

- used to extract the value of a selected node and add it to the output stream of the transformation
- a forward slash (/) selects subdirectories.

```
<xsl:template match="/">
  <html>
  <body> <h2>My CD Collection</h2>
    <table border="1"> <tr bgcolor="#9acd32"> <th>Title</th>
    <th>Artist</th> </tr> <tr> <td>.</td> <td></td> </tr> </table>
  </body> <xsl:value-of select="catalog/cd/title"/>
</html>
</xsl:template> <xsl:value-of select="catalog/cd/artist"/>
```

My CD Collection

Result:

Title	Artist
Empire Burlesque	Bob Dylan

The <xsl:for-each> Element

- can be used to select every XML element of a specified node-set
- Example:

Result:

```
<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>
        <td><xsl:value-of select="artist"/></td>
      </tr>
    </xsl:for-each>
  </table>
  </body>
  </html>
</xsl:template>
```

My CD Collection

Title	Artist
Empire Burlesque	Bob Dylan
Hide your heart	Bonnie Tyler
Greatest Hits	Dolly Parton
Still got the blues	Gary Moore
Eros	Eros Ramazzotti
One night only	Bee Gees
Sylvias Mother	Dr.Hook
Maggie May	Rod Stewart
Romanza	Andrea Bocelli

Filtering the Output

- `<xsl:for-each select="expression">`
- Example:
`<xsl:for-each select="catalog/cd[artist='Bob Dylan']">`
- Legal filter operators are:
 - `=` (equal)
 - `!=` (not equal)
 - `<` less than
 - `>` greater than

The <xsl:sort> Element

- add an <xsl:sort> element inside the <xsl:for-each> element

```
<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>
```


The <xsl:if> Element

- add the <xsl:if> element inside the <xsl:for-each> element in the XSL file

```
<xsl:for-each select="catalog/cd">
  <xsl:if test="price > 10">
    <tr>
      <td><xsl:value-of select="title"/></td>
      <td><xsl:value-of select="artist"/></td>
    </tr>
  </xsl:if>
</xsl:for-each>
```

The <xsl:choose> Element

```
<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 > 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>
```

My CD Collection

Title	Artist
Empire Burlesque	Bob Dylan
Hide your heart	Bonnie Tyler
Greatest Hits	Dolly Parton
Still got the blues	Gary Moore
Eros	Eros Ramazzotti
One night only	Bee Gees
Sylvias Mother	Dr.Hook
Maggie May	Rod Stewart
Romanza	Andrea Bocelli
When a man loves a woman	Percy Sledge
Black angel	Savage Rose
1999 Grammy Nominees	Many

Other Elements

- `<xsl:apply-templates>`
call other templates repeatedly
- `<xsl:output>`
output method to use, e.g., xml, text, or html
- `<xsl:element>`
dynamically create generic elements during the transformation process
- `<xsl:text>`
insert some specified text (PCDATA)
- `<xsl:copy-of>`
take sections of source tree and copy them directly to the result tree

XML Tools

- **Oxygen**, XML Spy, Stylus Studio
 - XML Editor, Schema Editor, Query Editor
- Apache Xerxes and family
 - XML Parser, XML Schema validator, XSLT Engine
- MXQuery, Saxon, ...
 - XQuery Engines