Dynamic Web Development

Lecture 4

More on XML 1

XML Technologies

There are a range of languages which enable you to process XML documents.

XPATH XSLT	Lecture 4 - More on XML 1
XSL-FO XML-DOM	Lecture 5 - More on XML 2

XPATH

XPath is a way of referring to a particular part of an XML document. It is very similar to the way in which you refer to a file on a disk.

XSLT (Transformation Style Sheets) make use of XPath during the transformation process.

XQuery and XPointer also make use of XPath expressions, and make use of some of the same functions.

XPath Terminology

An XML document is represented by a tree structure in the computer's memory.

The tree is made up of nodes. There are seven types:

- document (root)
- element
- attribute
- text
- namespace
- processing-instruction
- comment

Examples are based on this XML

```
<bookstore>

 <book>
    <title lang="eng">Harry Potter</title>
    <price>29.99</price>
 </book>

 <book>
    <title lang="eng">Learning XML</title>
    <price>39.95</price>
 </book>

</bookstore>
```

XPath Syntax

XPath uses path expressions to select individual nodes, or sets of nodes, in an XML document.

`nodename`	Selects all child nodes of the node
`/`	Selects from the root node
`//`	Selects nodes that match the selection no matter where they are
`.`	Selects the current node
`..`	Selects the parent of the current node
`@`	Selects attributes

Selecting Nodes: Examples

`bookstore`	Selects all the child nodes of the bookstore element
`/bookstore`	Selects the root element bookstore
`bookstore/book`	Selects all book elements that are children of bookstore
`//book`	Selects all book elements, no matter where they are in the document
`bookstore//book`	Select all book elements that are below the bookstore element
`//@lang`	Select all attributes named lang, no matter where they are in the document

Predicates

Predicates are used to find a specific node or a node that contains a specific value.

Predicates are always enclosed in square brackets

Predicates: Examples

`/bookstore/book[1]`	The second book element that is a child of bookstore
`/bookstore/book[last()]`	The last book element that is a child of bookstore
`/bookstore/book[last()-1]`	The last but one element that is a child of bookstore
`/bookstore/book[position()<3]`	The first two book elements that are children of bookstore

`//title[@lang]`	All title elements that have an attribute named lang
`//title[@lang='eng']`	All title elements that have an attribute named lang, with a value of 'eng'
`/bookstore/book[price>35.00]`	All the book elements of bookstore that have a price element greater than 35.00
`/bookstore/book[price>35.00]/title`	All the title elements of the book elements ..etc.

Wildcards

`*`	Any element node
`@*`	Any attribute node
`node()`	Any node of any kind
`/bookstore/*`	All child nodes of bookstore
`//*`	All elements in the document
`//title[@*]`	All title elements that have any attribute

XSLT vs XSL-FO

Are XSL-FO and XSLT the same thing?

Styling is both about **transforming** and **formatting** information. When the World Wide Web Consortium (W3C) made their first XSL Working Draft, it contained the language syntax for both transforming and formatting XML documents.

Later, the Working Group at W3C split the original draft into separate Recommendations:

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

XSLT

Extensible Stylesheet Language Transformations

A language which can convert one XML-based language into another.

XSL Transformation (XSLT)

Can reuse data (variables)

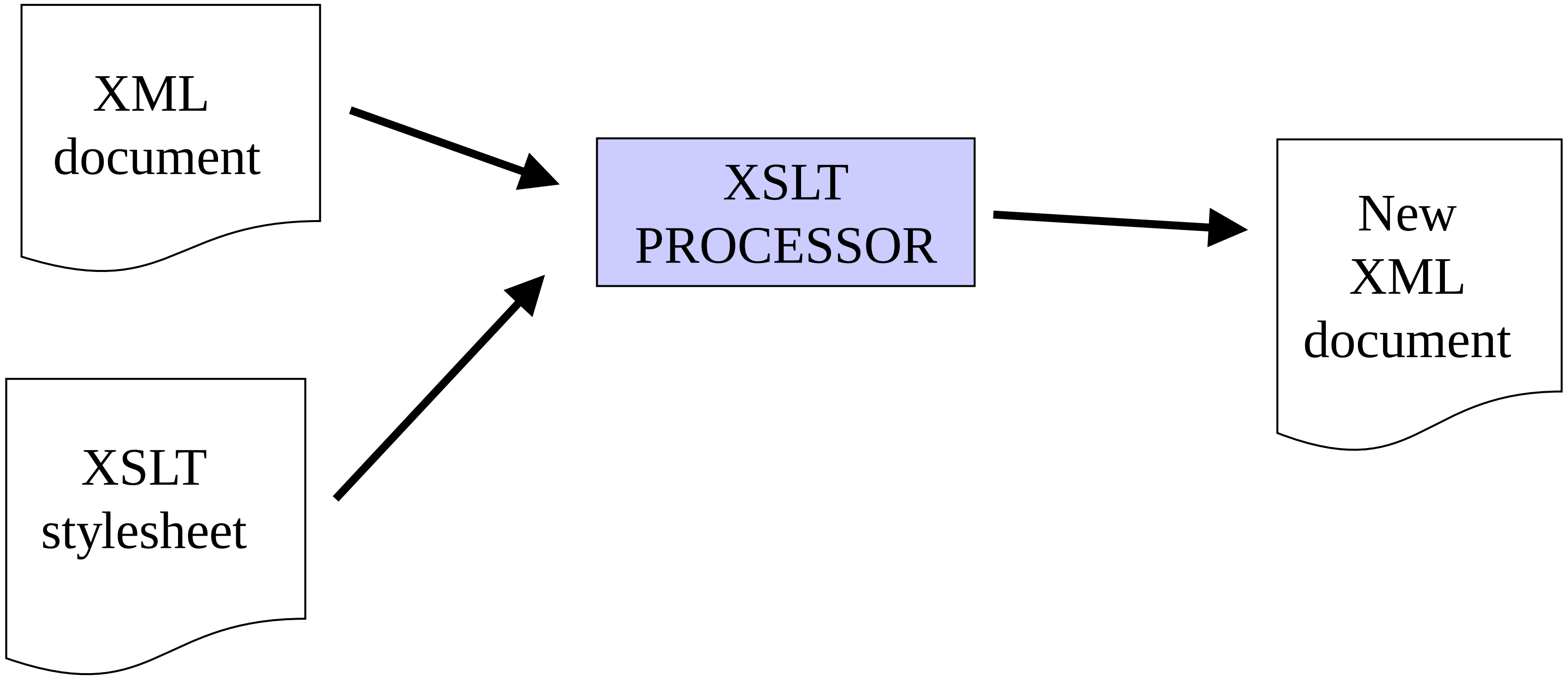
Can conditionally select document data.

Can calculate quantities

Can generate dynamic text eg page numbers

XSLT Processor

A piece of software that applies an XSLT stylesheet to an XML document.



Location of XSLT processor

Built into a web browser

- Internet Explorer has one called MSXML

Built into a web server

- Apache server has a module called Cocoon

Standalone program

- Saxon (http://saxon.sourceforge.net)
- Apache has one called Xalan

XSLT Stylesheets

An XSLT stylesheet doesn’t just specify how the document should appear in a browser.

It actually specifies how to change the document into a different XML-based language.

It can change one tag into another – not just change how it is displayed in a browser.

Example

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

<booker>
  <award>
    <author>Kingsley Amis</author>
    <title>The Old Devils</title>
    <year>1986</year>
  </award>
  <award>
    <author>Margaret Atwood</author>
    <title>The Blind Assassin</title>
    <year>2000</year>
  </award>
  <award>
    <author>Pat Barker</author>
    <title>The Ghost Road</title>
    <year>1995</year>
  </award>


and so on
```

Here is the XML document

XSL-T stylesheet

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">

   <html>
      <body>
         <h2 align="center">Author, Book</h2>
            <table border="2" bordercolor="black" align="center">
               <tr bgcolor="blue">
                  <th color="white">Author</th>
                  <th color="white">Book Title</th>
               </tr>
               <xsl:for-each select="booker/award">
                  <tr>
                     <td><xsl:value-of select="author" /></td>
                     <td><xsl:value-of select="title" /></td>
                  </tr>
               </xsl:for-each>
            </table>
      </body>
   </html>

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

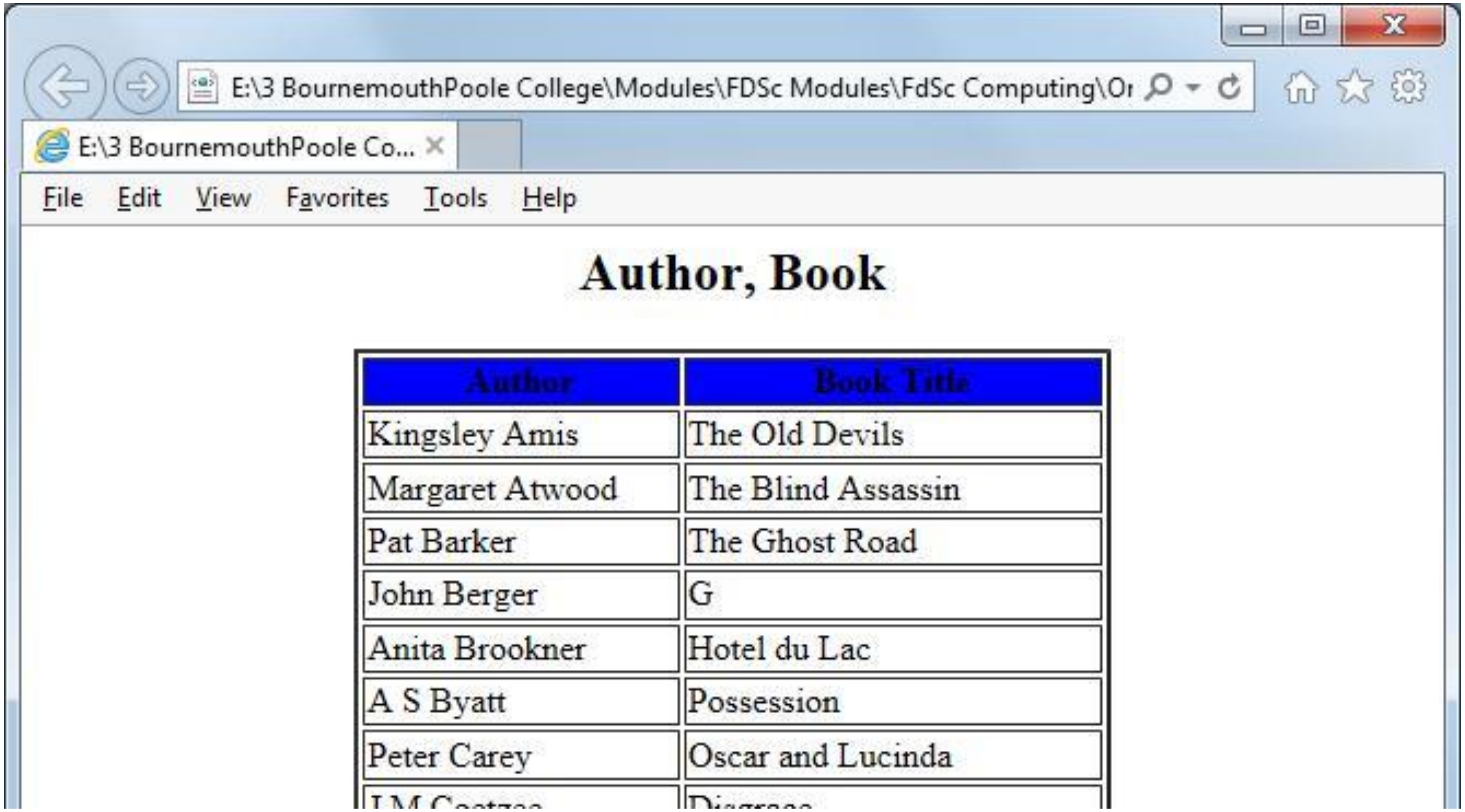
Any tag without a prefix is assumed to be in the default <html> namespace

Will produce this output

```
<html>
    <body>
        <h2 align="center">Author, Book</h2>
        <table border="2" bordercolor="black" align="center">
            <tr bgcolor="blue">
                <th color="white">Author</th>
                <th color="white">Book Title</th>
            </tr>
            <tr>
                <td>Kingsley Amis</td>
                <td>The Old Devils</td>
            </tr>
            <tr>
                <td>Margaret Atwood</td>
                <td>The Blind Assassin</td>
            </tr>
            <tr>
                <td>Pat Barker</td>
                <td>The Ghost Road</td>
            </tr>
            and so on. . . . .

        </table>
    </body>
</html>
```

will produce this


E:\3 BournemouthPoole College\Modules\FDSc Modules\FdSc Computing\Or
E:\3 BournemouthPoole Co...
File
Edit
View
Favorites
Tools
Help

Author, Book

Author	Book Title
Kingsley Amis	The Old Devils
Margaret Atwood	The Blind Assassin
Pat Barker	The Ghost Road
John Berger	G
Anita Brookner	Hotel du Lac
A S Byatt	Possession
Peter Carey	Oscar and Lucinda
J M Coetzee	Disgrace

bookerb.xsl

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:template match="/">

    <html>
        <body>
            <h2 align="center">Author, Book</h2>
                <table border="2" bordercolor="black" align="center">
                    <tr bgcolor="blue">
                        <th color="white">Author</th>
                        <th color="white">Book Title</th>
                        <th color="white">Year</th>
                    </tr>
                    <xsl:for-each select="booker/award">
                        <tr>
                            <td><xsl:value-of select="author" /></td>
                            <td><xsl:value-of select="title" /></td>
                            <td><xsl:value-of select="year" /></td>
                        </tr>
                    </xsl:for-each>
                </table>
        </body>
    </html>

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

bookerc.xsl

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:template match="/">

    <html>
        <body>
            <h2 align="center">Author, Book</h2>
                <table border="2" bordercolor="black" align="center">
                    <tr bgcolor="blue">
                        <th color="white">Author</th>
                        <th color="white">Book Title</th>
                        <th color="white">Year</th>
                    </tr>
                    <xsl:for-each select="booker/award">
                      <xsl:sort select="year" />
                        <tr>
                            <td><xsl:value-of select="author" /></td>
                            <td><xsl:value-of select="title" /></td>
                            <td><xsl:value-of select="year" /></td>
                        </tr>
                    </xsl:for-each>
                </table>
        </body>
    </html>

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

bookerd.xsl

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:template match="/">

    <html>
        <body>
            <h2 align="center">Author, Book</h2>
                <table border="2" bordercolor="black" align="center">
                    <tr bgcolor="blue">
                        <th color="white">Author</th>
                        <th color="white">Book Title</th>
                        <th color="white">Year</th>
                    </tr>
                    <xsl:for-each select="booker/award">
                      <xsl:sort select="year" />
                        <xsl:if test="year &gt; 1990">
                            <tr>
                                <td><xsl:value-of select="author" /></td>
                                <td><xsl:value-of select="title" /></td>
                                <td><xsl:value-of select="year" /></td>
                            </tr>
                        </xsl:if>
                    </xsl:for-each>
                </table>
        </body>
    </html>

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

bookere.xsl

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:template match="/">

    <html>
        <body>
            <h2 align="center">Author, Book</h2>
                <table border="2" bordercolor="black" align="center">
                    <tr bgcolor="blue">
                        <th color="white">Author</th>
                        <th color="white">Book Title</th>
                        <th color="white">Year</th>
                    </tr>
                    <xsl:for-each select="booker/award">
                        <xsl:choose>
                          <xsl:when test="year &lt; 1985">
                            <tr>
                               <td bgcolor="red"><xsl:value-of select="author" /></td>
                               <td bgcolor="red"><xsl:value-of select="title" /></td>
                               <td bgcolor="red"><xsl:value-of select="year" /></td>
                            </tr>
                          </xsl:when>
                          <xsl:otherwise>
                            <tr>
                               <td><xsl:value-of select="author" /></td>
                               <td><xsl:value-of select="title" /></td>
                               <td><xsl:value-of select="year" /></td>
                            </tr>
                          </xsl:otherwise>
                        </xsl:choose>
                    </xsl:for-each>
            </table>
        </body>
    </html>

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