PRESENTATION TECHNOLOGIES

1) XSL
2) XFORMS
3) XHTML

XSL & XSLT:

XSL stands for EXtensible Stylesheet Language.

What is XSLT?

XSLT stands for XSL Transformations. XSLT is the most important part of XSL. XSLT transforms an XML document into another XML document. XSLT uses XPath to navigate in XML documents. XSLT is a W3C Recommendation. We want to **transform** the following XML document ("cdcatalog.xml") into XHTML:

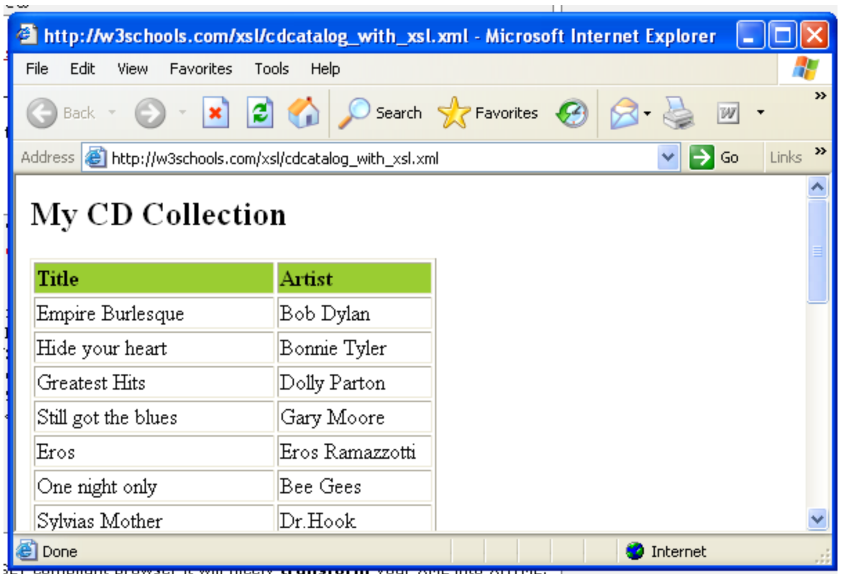
```
<?xml version="1.0" encoding="ISO-8859-1"?>
        <catalog>
        <cd>
        <title>Empire Burlesque</title>
        <artist>Bob
Dylan</artist><country>USA</country><company>Columbia</company><price>
10.90</price>
        <year>1985</year>
        </cd> . . .

</catalog>
```

Then you create an XSL Style Sheet ("cdcatalog.xsl") with a transformation template:

```
<?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>My CD Collection</h2>
                    <table border="1">
                    <tr bgcolor="#9acd32">
                    <th align="left">Title</th>
                    <th 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>
```

The result is:


http://w3schools.com/xsl/cdcatalog_with_xsl.xml - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address http://w3schools.com/xsl/cdcatalog_with_xsl.xml
Go
Links
Done
Internet

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

XFORMS:

XForms is the next generation of HTML forms. XForms is richer and more flexible than HTML forms. XForms will be the forms standard in XHTML 2.0. XForms is platform and device independent. XForms separates data and logic from presentation. XForms uses XML to define form data. XForms stores and transports data in XML documents. XForms contains features like calculations and validations of forms. XForms reduces or eliminates the need for scripting. XForms is a W3C Recommendation. The XForms Model. The XForms model is used to describe the data. The data model is an instance (a template) of an XML document. The XForms model defines a data model inside a <model> element:

```
<model>
<instance>
<person>
<fname/>
<lname/>
</person>
```

```
                    </instance>
                    <submission id="form1" action="submit.asp" method="get"/>
                    </model>
```

The XForms Model

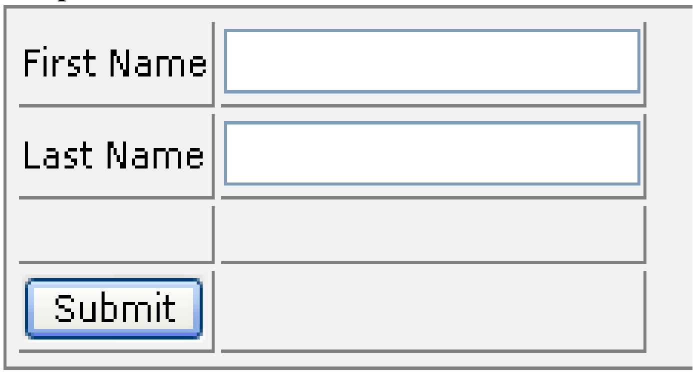
The XForms **model** is used to **describe** the data. The data model is an instance (a template) of an XML document. The XForms model defines a data model inside a <model> element:

```
          <model>
          <instance>
          <person>
          <fname/>
          <lname/>
          </person>
          </instance>
          <submission id="form1" action="submit.asp" method="get"/>
          </model>
```

All together it looks as below

```
          <xforms>
          <model>
          <instance>
          <person><fname/><lname/></person></instance>
          <submission id="form1" action="submit.asp" method="get"/>
          </model>
          <input ref="fname"><label>First Name</label></input><input


ref="lname"><label>Last Name</label></input><submit submission="form1">
          <label>Submit</label>
          </submit>
          </xforms>
```

Output seems like:



XHTML:

XHTML stands for EXtensible HyperText Markup Language. XHTML is aimed to replace HTML. XHTML is almost identical to HTML 4.01. XHTML is a stricter and cleaner version of HTML. XHTML is HTML defined as an XML application. XHTML is a W3C Recommendation. XHTML elements must be properly nested. XHTML elements must always be closed. XHTML elements must be in lowercase. XHTML documents must have one root element.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
        <head>
        <title>simple document</title>
        </head>
        <body><p>a simple paragraph</p></body>
</html>
```

The 3 Document Type Definitions :

1) DTD specifies the syntax of a web page in SGML.
2) DTD is used by SGML applications, such as HTML, to specify rules that apply to the markup of documents of a particular type, including a set of element and entity declarations.
3) XHTML is specified in an SGML document type definition or 'DTD'.

An XHTML DTD describes in precise, computer-readable language, the allowed syntax and grammar of XHTML markup. There are currently 3 XHTML document types:

i. STRICT
ii. TRANSITIONAL
iii. FRAMESET

XHTML 1.0 specifies three XML document types that correspond to three DTDs:

i. Strict
ii. Transitional
iii. Frameset

XHTML 1.0 Strict:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

We can use this when you want really clean markup, free of presentational clutter. We can use this together with Cascading Style Sheets.

XHTML 1.0 Transitional:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

We can use this when you need to take advantage of HTML's presentational features and when you want to support browsers that don't understand Cascading Style Sheets.

XHTML 1.0 Frameset:

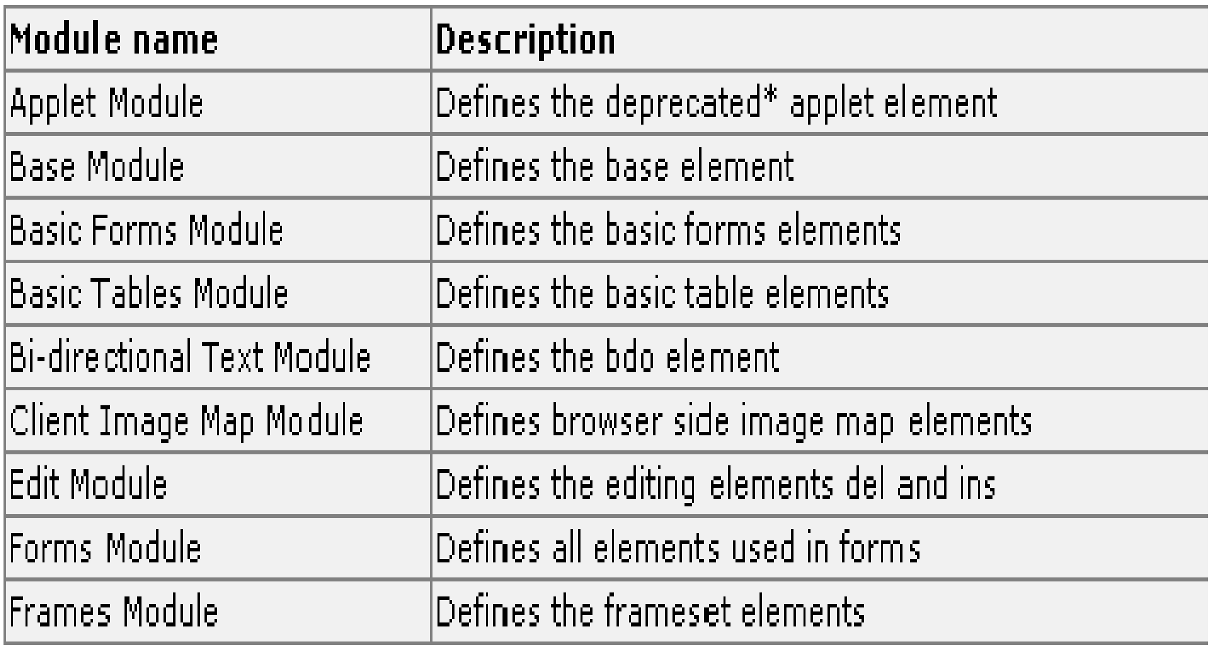
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

We can use this when you want to use HTML Frames to partition the browser window into two or more frames.

Why XHTML Modularization?

By splitting XHTML into modules, the W3C (World Wide web Consortium) has created small and well-defined sets of XHTML elements that can be used separately for small devices, or combined with other XML standards into larger and more complex applications.

Some of the modules are as below:



Module name	Description
Applet Module	Defines the deprecated* applet element
Base Module	Defines the base element
Basic Forms Module	Defines the basic forms elements
Basic Tables Module	Defines the basic table elements
Bi-directional Text Module	Defines the bdo element
Client Image Map Module	Defines browser side image map elements
Edit Module	Defines the editing elements del and ins
Forms Module	Defines all elements used in forms
Frames Module	Defines the frameset elements

TRANSFORMATION:

- XSLT
- XLINK
- XPATH
- XQuery

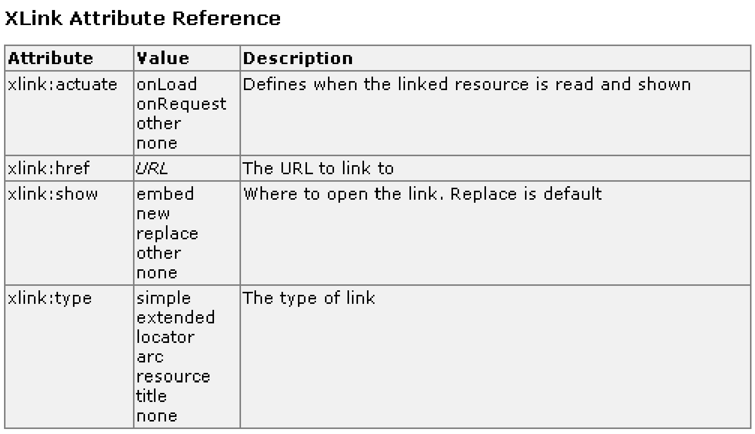
XLINK:

XLink Syntax:

In HTML, we know (and all the browsers know!) that the <a> element defines a hyperlink. However, this is not how it works with XML. In XML documents, you can use whatever element names you want - therefore it is impossible for browsers to predict what hyperlink elements will be called in XML documents.The solution for creating links in XML documents was to put a marker on elements that should act as hyperlinks.

Example:

```
<?xml version="1.0"?>
<homepages xmlns:xlink="http://www.w3.org/1999/xlink">
<homepage xlink:type="simple" xlink:href="http://www.w3schools.com">Visit W3Schools</homepage><homepage xlink:type="simple" xlink:href="http://www.w3.org">Visit W3C</homepage>
</homepages>
```



XLink Attribute Reference

Attribute	Value	Description
xlink:actuate	onLoad onRequest other none	Defines when the linked resource is read and shown
xlink:href	*URL*	The URL to link to
xlink:show	embed new replace other none	Where to open the link. Replace is default
xlink:type	simple extended locator arc resource title none	The type of link

XPATH:

XPath is a syntax for defining parts of an XML document. XPath uses path expressions to navigate in XML documents. XPath contains a library of standard functions
XPath is a major element in XSLT. XPath is a W3C Standard.

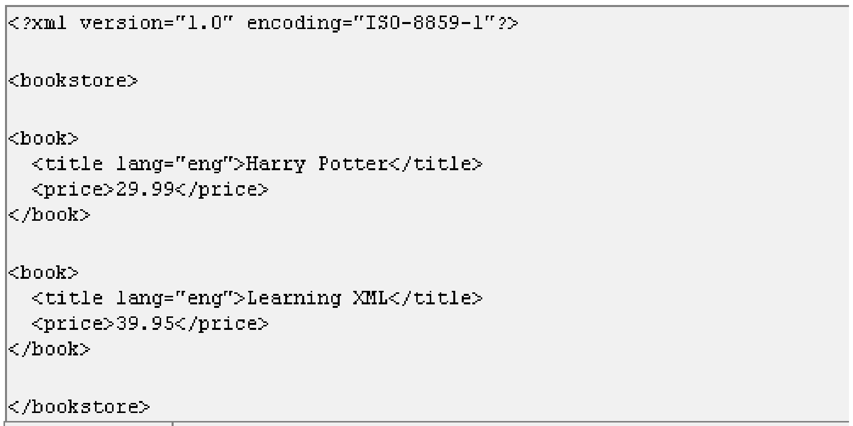
XPath Terminology

Nodes:

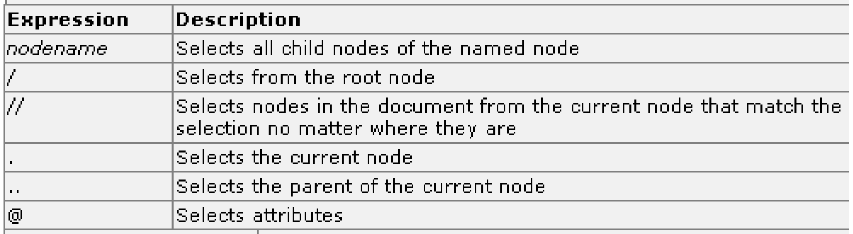
In XPath, there are seven kinds of nodes: element, attribute, text, namespace, processing-instruction, comment, and document (root) nodes. XML documents are treated as trees of nodes. The root of the tree is called the document node (or root node).

Relationship of Nodes

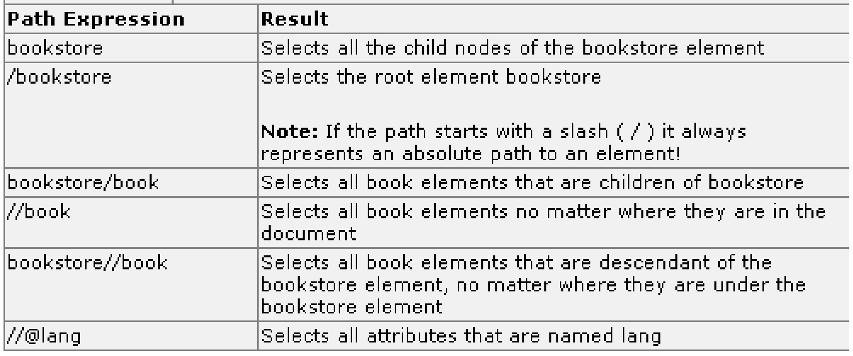
i. Parent
ii. Children
iii. Siblings
iv. Ancestors
v. Descendants



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

<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>
```

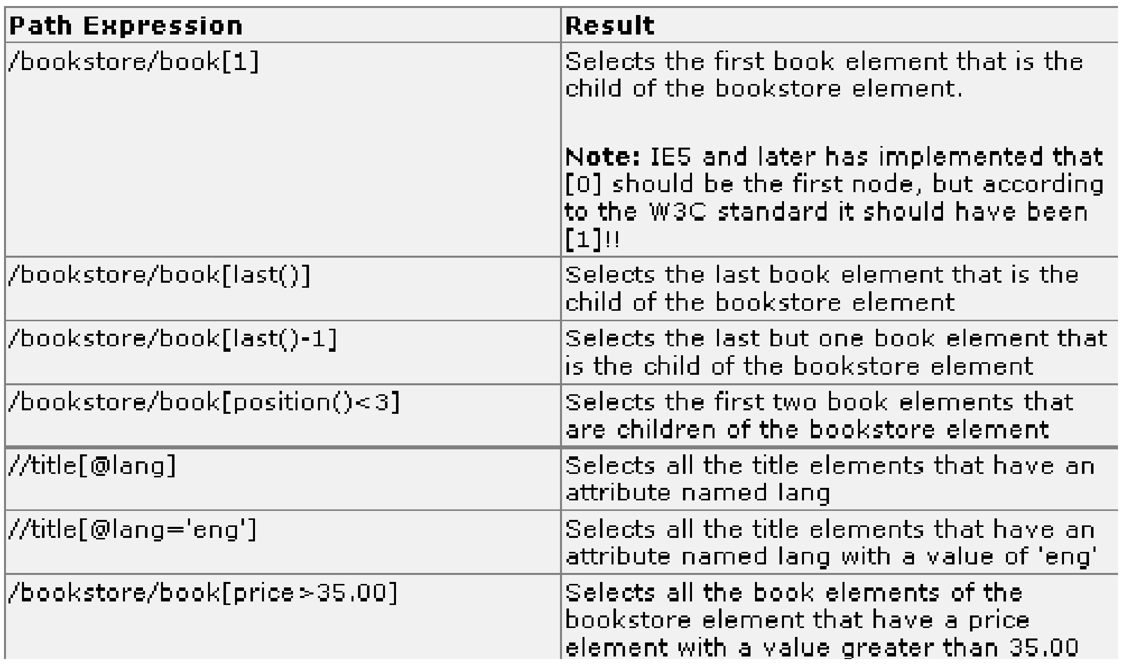


Expression	Description
nodename	Selects all child nodes of the named node
/	Selects from the root node
//	Selects nodes in the document from the current node that match the selection no matter where they are
.	Selects the current node
..	Selects the parent of the current node
@	Selects attributes



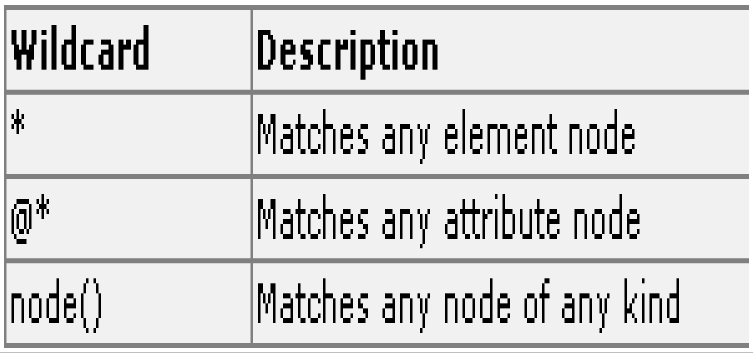
Path Expression	Result
bookstore	Selects all the child nodes of the bookstore element
/bookstore	Selects the root element bookstore **Note:** If the path starts with a slash (/) it always represents an absolute path to an element!
bookstore/book	Selects all book elements that are children of bookstore
//book	Selects all book elements no matter where they are in the document
bookstore//book	Selects all book elements that are descendant of the bookstore element, no matter where they are under the bookstore element
//@lang	Selects all attributes that are named lang

Predicates:

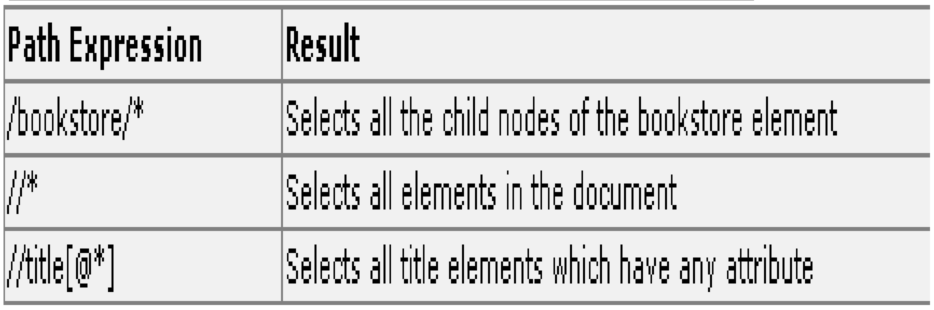


Path Expression	Result
/bookstore/book[1]	Selects the first book element that is the child of the bookstore element. **Note:** IE5 and later has implemented that [0] should be the first node, but according to the W3C standard it should have been [1]!!
/bookstore/book[last()]	Selects the last book element that is the child of the bookstore element
/bookstore/book[last()-1]	Selects the last but one book element that is the child of the bookstore element
/bookstore/book[position()<3]	Selects the first two book elements that are children of the bookstore element
//title[@lang]	Selects all the title elements that have an attribute named lang
//title[@lang='eng']	Selects all the title elements that have an attribute named lang with a value of 'eng'
/bookstore/book[price>35.00]	Selects all the book elements of the bookstore element that have a price element with a value greater than 35.00

Selecting Unknown Nodes:

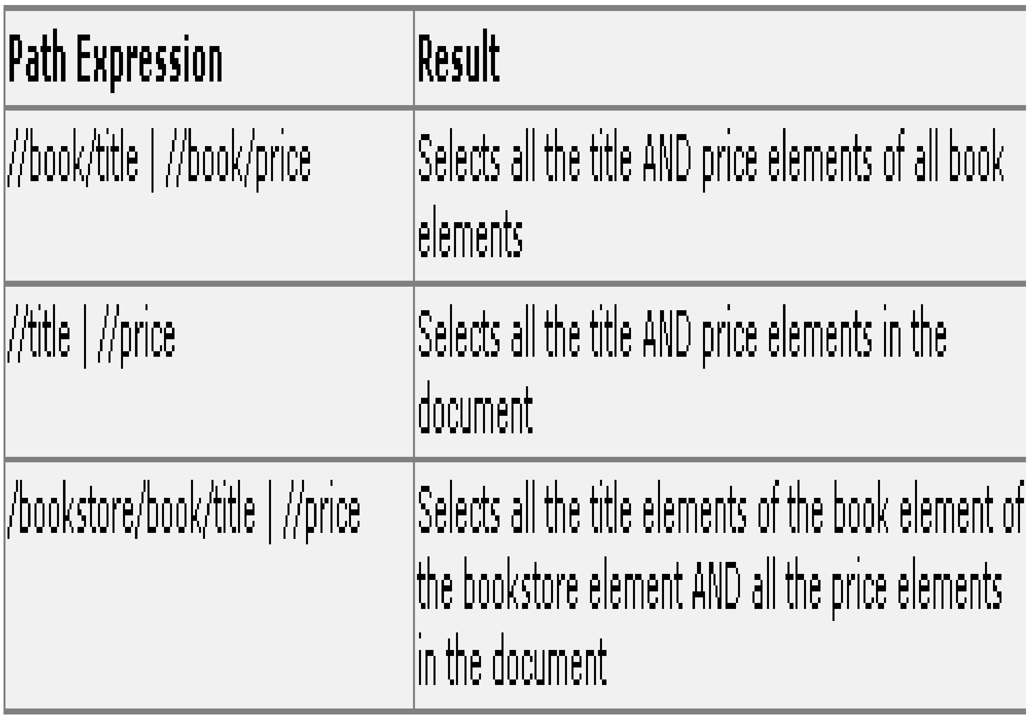


Wildcard	Description
*	Matches any element node
@*	Matches any attribute node
node()	Matches any node of any kind



Path Expression	Result
/bookstore/*	Selects all the child nodes of the bookstore element
//*	Selects all elements in the document
//title[@*]	Selects all title elements which have any attribute

Selecting several paths:



Path Expression	Result
//book/title | //book/price	Selects all the title AND price elements of all book elements
//title | //price	Selects all the title AND price elements in the document
/bookstore/book/title | //price	Selects all the title elements of the book element of the bookstore element AND all the price elements in the document

XQuery:

XQuery is ***the*** language for querying XML data. XQuery for XML is like SQL for databases. XQuery is built on XPath expressions. XQuery is supported by all the major database engines (IBM, Oracle, Microsoft, etc.). XQuery is a W3C Recommendation .

```
<title lang="en">XQuery Kick Start</title>
 <author>James McGovern</author>
 <author>Per Bothner</author>
 <author>Kurt Cagle</author>
 <author>James Linn</author>
 <author>Vaidyanathan Nagarajan</author>
 <year>2003</year>
 <price>49.99</price>
 </book>
- <book category="WEB">
 <title lang="en">Learning XML</title>
 <author>Erik T. Ray</author>
 <year>2003</year>
 <price>39.95</price>
 </book>
 </bookstore>
```

Functions:

XQuery uses functions to extract data from XML documents. The doc() function is used to open the "books.xml" file:

doc("books.xml"), Path Expressions

XQuery uses path expressions to navigate through elements in an XML document.The following path expression is used to select all the title elements in the "books.xml" file: doc("books.xml")/bookstore/book/title(/bookstore selects the bookstore element, /book selects all the book elements under the bookstore element, and /title selects all the title elements under each book element), The XQuery above will extract the following:

```
<title lang="en">Everyday Italian</title>
<title lang="en">Harry Potter</title>
<title lang="en">XQuery Kick Start</title>
<title lang="en">Learning XML</title>
```

Predicates:

XQuery uses predicates to limit the extracted data from XML documents. The following predicate is used to select all the book elements under the bookstore element that have a price element with a value that is less than 30: doc("books.xml")/bookstore/book[price<30]The XQuery above will extract the following:

```
    <book category="CHILDREN">
    <title lang="en">Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
<price>29.99</price>
    </book>
```

With FLWOR:

FLWOR is an acronym for **"For, Let, Where, Order by, Return"**. The **for** clause selects all book elements under the bookstore element into a variable called $x.The **where** clause selects only book elements with a price element with a value greater than 30.The **order by** clause defines the sort-order. Will be sort by the title element.The **return** clause specifies what should be returned. Here it returns the title elements.

Example: doc("books.xml")/bookstore/book[price>30]/title

The following FLWOR expression will select exactly the same as the path expression above:

for $x in doc("books.xml")/bookstore/book where $x/price>30 return $x/title

The result will be:

```
<title lang="en">XQuery Kick Start</title>
<title lang="en">Learning XML</title>
```

With FLWOR you can sort the result:

for $x in doc("books.xml")/bookstore/book where $x/price>30 order by $x/title return $x/title

Source : http://nprcet.org/e%20content/Misc/e-Learning/IT/VIII%20Sem/IT1451%20-%20XML%20and%20Web%20Services.pdf