

Unit 6

XPath

XPath is a major element in the XSLT standard.

- XPath can be used to navigate through elements and attributes in an XML document.

XPath Path Expressions

XPath uses path expressions to select nodes or node-sets in an XML document.

These path expressions look very much like the path expressions you use with traditional computer file systems:



XPath Nodes

In XPath, there are seven kinds of nodes: element, attribute, text, namespace, processing-instruction, comment, and root nodes.

- XML documents are treated as trees of nodes. The topmost element of the tree is called the root element.

```
<?xml version="1.0" encoding="UTF-8"?>

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

Example of nodes in the XML document above:

<bookstore> (root element node)

<author>J K. Rowling</author> (element node)

lang="en" (attribute node)

Relationship of Nodes

Parent: Each element and attribute has one parent.

- In the following example; the book element is the parent of the title, author, year, and price:

- `<book>`
 - `<title>Harry Potter</title>`
 - `<author>J K. Rowling</author>`
 - `<year>2005</year>`
 - `<price>29.99</price>`
- `</book>`

Children

Element nodes may have zero, one or more children.

- In the following example; the title, author, year, and price elements are all children of the book element:
- `<book>`
 - `<title>Harry Potter</title>`
 - `<author>J K. Rowling</author>`
 - `<year>2005</year>`
 - `<price>29.99</price>`
- `</book>`

Siblings

Nodes that have the same parent.

- In the following example; the title, author, year, and price elements are all siblings:
- `<book>`
 - `<title>Harry Potter</title>`
 - `<author>J K. Rowling</author>`
 - `<year>2005</year>`
 - `<price>29.99</price>`
- `</book>`

- Ancestors

A node's parent, parent's parent, etc.

- In the following example; the ancestors of the title element are the book element and the bookstore element:
- `<bookstore>`

`<book>`

`<title>Harry Potter</title>`

`<author>J K. Rowling</author>`

`<year>2005</year>`

`<price>29.99</price>`

`</book>`

`</bookstore>`

- Descendants

A node's children, children's children, etc.

- In the following example; descendants of the bookstore element are the book, title, author, year, and price elements:

- `<bookstore>`

```
<book>  
  <title>Harry Potter</title>  
  <author>J K. Rowling</author>  
  <year>2005</year>  
  <price>29.99</price>  
</book>
```

```
</bookstore>
```

Selecting Nodes

- XPath uses path expressions to select nodes in an XML document. The node is selected by following a path or steps. The most useful path expressions are listed below:

Expression	Description
<i>nodename</i>	Selects all nodes with the name " <i>nodename</i> "
/	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

- In the table below we have listed some path expressions and the result of the expressions:

Path Expression	Result
bookstore	Selects all nodes with the name "bookstore"
/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

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

Predicates are always embedded in square brackets.

- In the table below we have listed some path expressions with predicates and the result of the expressions:

Path Expression	Result
/bookstore/book[1]	Selects the first book element that is the child of the bookstore element.
/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='en']	Selects all the title elements that have a "lang" attribute with a value of "en"
/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
/bookstore/book[price>35.00]/title	Selects all the title elements of the book elements of the bookstore element that have a price element with a value greater than 35.00

XPath Axes

- An axis represents a relationship to the context (current) node, and is used to locate nodes relative to that node on the tree.

AxisName	Result
ancestor	Selects all ancestors (parent, grandparent, etc.) of the current node
ancestor-or-self	Selects all ancestors (parent, grandparent, etc.) of the current node and the current node itself
attribute	Selects all attributes of the current node
child	Selects all children of the current node
descendant	Selects all descendants (children, grandchildren, etc.) of the current node
descendant-or-self	Selects all descendants (children, grandchildren, etc.) of the current node and the current node itself
following	Selects everything in the document after the closing tag of the current node
following-sibling	Selects all siblings after the current node
namespace	Selects all namespace nodes of the current node
parent	Selects the parent of the current node
preceding	Selects all nodes that appear before the current node in the document, except ancestors, attribute nodes and namespace nodes
preceding-sibling	Selects all siblings before the current node
self	Selects the current node