



# XPath - Basic concepts

Systems Integration course

# XPath

- XPath is a language for finding information in a XML document
- Is used to navigate through elements and attributes
- It has a specific syntax
  - XPath path expressions
  - XPath functions

# XML document example

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

<bookstore>
  <book>
    <title lang="en">Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2001</year>
    <price>26.76</price>
  </book>
</bookstore>
```

Relation ship of nodes:

- **parent:** each element and attribute has one parent
- **children:** element nodes may have zero, one or no children
- **siblings:** nodes that have the same parent
- **ancestors:** a node's parent, parent's parent, etc. (the ancestors of the title element are the book element and the bookstore element)
- **descendants:** a node's children, children's children, etc.

# Selecting nodes

nodename	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

```
<?xml version="1.0" encoding="utf-8"?>
<bookstore>
  <book category="CHILDREN">
    <title>Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book category="WEB">
    <title>Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
    <rate>3</rate>
  </book>
  <book category="BIOGRAPHIES" isbn="111111">
    <title>Steve Jobs</title>
    <author>Walter Isaacson</author>
    <year>2011</year>
    <price>21.95</price>
  </book>
</bookstore>
```

# Selecting nodes

<code>bookstore</code>	Selects all nodes with the name "bookstore"
<code>/bookstore</code>	Selects the root element bookstore <b>Note:</b> If the path starts with a slash ( / ) it always represents an absolute path to an element
<code>bookstore/book</code>	Selects all book elements that are children of bookstore
<code>//book</code>	Selects all book elements no matter where they are in the document
<code>bookstore//book</code>	Selects all book elements that are descendant of the bookstore element, no matter where they are under the bookstore element
<code>//@category</code>	Selects all attributes that are named category

```
<?xml version="1.0" encoding="utf-8"?>
<bookstore>
  <book category="CHILDREN">
    <title>Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book category="WEB">
    <title>Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
    <rate>3</rate>
  </book>
  <book category="BIOGRAPHIES" isbn="111111">
    <title>Steve Jobs</title>
    <author>Walter Isaacson</author>
    <year>2011</year>
    <price>21.95</price>
  </book>
</bookstore>
```

# Predicates

- Predicates are used to find a specific node or a node that contains a specific value
- Predicates are always embedded in square brackets [ ]

# Predicates

<code>/bookstore/book[1]</code>	Selects the first book element that is the child of the bookstore element
<code>/bookstore/book[last()-1]</code>	Selects the last but one book element that is the child of the bookstore element
<code>/bookstore/book[position()&lt;3]</code>	Selects the first two book elements that are children of the bookstore element
<code>//book[@isbn]</code>	Selects all the book elements that have an attribute named isbn
<code>//book[@category='WEB']</code>	Selects all the book elements that have an attribute named category with a value of 'WEB'
<code>/bookstore/book[year&gt;2010]</code>	Selects all the book elements of the bookstore element that have a year element with a value greater than 2010
<code>/bookstore/book[year=2005]/title</code>	Selects all the title elements of the book elements of the bookstore element that have a year element with a value 2005

```
<?xml version="1.0" encoding="utf-8"?>
<bookstore>
  <book category="CHILDREN">
    <title>Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book category="WEB">
    <title>Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
    <rate>3</rate>
  </book>
  <book category="BIOGRAPHIES" isbn="111111">
    <title>Steve Jobs</title>
    <author>Walter Isaacson</author>
    <year>2011</year>
    <price>21.95</price>
  </book>
</bookstore>
```

# Select unknown nodes

<code>/bookstore/*</code>	Selects all the child nodes of the bookstore element
<code>//*</code>	Selects all elements in the document
<code>@*</code>	Matches any attribute node
<code>//book[@*]</code>	Selects all book elements which have any attribute

```
<?xml version="1.0" encoding="utf-8"?>
<bookstore>
  <book category="CHILDREN">
    <title>Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book category="WEB">
    <title>Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
    <rate>3</rate>
  </book>
  <book category="BIOGRAPHIES" isbn="111111">
    <title>Steve Jobs</title>
    <author>Walter Isaacson</author>
    <year>2011</year>
    <price>21.95</price>
  </book>
</bookstore>
```

**Using the | operator in an XPath expression it possible to select several paths**

<code>//book/title   //book/year</code>	Selects all the title AND year elements of all book elements
<code>//title   //author</code>	Selects all the title AND author elements in the document
<code>/bookstore/book/title   //year</code>	Selects all the title elements of the book element of the bookstore element AND all the year elements in the document



# XPath operators

	computes two nodes-sets, e.g. //book   //game
--	---

- +, -, \*, div, =, !=, <, <=, >, >=, or, and, mod  
(modulus - division remainder)

ex.:

- year=2001, year<2013, price>10.25 and  
price<15.00, rate=5 or rate=4, 5 mod 2, 8  
div 4

# XPath axes

- An axis defines a node-set relative to the current node
- Syntax:

`axisname::nodetest[predicate]`

<code>ancestor</code>	Selects all ancestors (parent, grandparent, etc.) of the current node
<code>ancestor-or-self</code>	Selects all ancestors (parent, grandparent, etc.) of the current node and the current node itself
<code>attribute</code>	Selects all attributes of the current node
<code>child</code>	Selects all children of the current node
<code>descendant</code>	Selects all descendants (children, grandchildren, etc.) of the current node
<code>descendant-or-self</code>	Selects all descendants (children, grandchildren, etc.) of the current node and the current node itself
<code>following</code>	Selects everything in the document after the closing tag of the current node
<code>parent</code>	Selects the parent of the current node
<code>self</code>	Selects the current node
<code>...</code>	and many more...

# XPath axes

## Examples:

<code>child::book</code>	Selects all book nodes that are children of the current node
<code>attribute::isbn</code>	Selects the isbn attribute of the current node
<code>child::*</code>	Selects all element children of the current node
<code>attribute::*</code>	Selects all attributes of the current node
<code>child::text()</code>	Selects all text node children of the current node
<code>child::node()</code>	Selects all children of the current node
<code>descendant::book</code>	Selects all book descendants of the current node
<code>ancestor::book</code>	Selects all book ancestors of the current node

# XPath axes

- An axis defines a node-set relative to the current node



more information in the lecture class

# Free tools

- There are many online free tools to help testing XPath expressions

FREEFORMATTER.COM    HTTPS    Contact    Like 515    G+

### Formatters

- JSON Formatter
- HTML Formatter
- XML Formatter
- SQL Formatter
- Batch Formatter (new!)

### Validators

- JSON Validator
- HTML Validator
- XML Validator - XSD

### XPath Tester

- Credit Card Number Generator & Validator
- Regular Expression Tester
- Java Regular Expression Tester
- Cron Expression Generator - Quartz

### Encoders & Decoders

- Url Encoder & Decoder
- Base 64 Encoder & Decoder
- QR Code Generator

### Code Minifiers / Beautifier

- JavaScript Beautifier
- CSS Beautifier
- JavaScript Minifier
- CSS Minifier

### Converters

- XSD Generator
- XSLT (XSL Transformer)
- XML to JSON Converter
- JSON to XML Converter
- CSV to XML Converter

### XPath Tester / Evaluator

Allows you to test your XPath expressions/queries against a XML file. This tool runs better than other existing XPath online tools as it supports most of the XPath functions (string(), number(), name(), string-length() etc.) and does not limit you to working against nodes. It fully supports XPath 2.0 / 3.0 specification. See the [XPath Examples](#) section for details.

The XPath tester fully supports XML namespaces. See the XPath Examples section for details. The namespace prefix "fn" and "math" are reserved to XPath functions.

\*The maximum size limit for file upload is 2 megabytes. Results bigger than 500k will be written to a new window for performance reason and to prevent your browser from being unresponsive.

#### XML Input

Option 1: Copy-paste your XML document here

Option 2: Or upload your XML document

Choose file No file chosen UTF-8

#### XPath expression

Include the XML item type in output:

Yes

TEST XPATH TEST XPATH IN NEW WINDOW

<https://www.freeformatter.com/xpath-tester.html>

<http://www.xpathtester.com/xpath>