POLITÉCNICO DE LEIRIA
ESCOLA SUPERIOR
DE TECNOLOGIA E GESTÃO

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 "*nodename*"
/	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

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
//@category	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

/bookstore/book[1]	Selects the first 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
//book[@isbn]	Selects all the book elements that have an attribute named isbn
//book[@category='WEB']	Selects all the book elements that have an attribute named category with a value of 'WEB'
/bookstore/book[year>2010]	Selects all the book elements of the bookstore element that have a year element with a value greater than 2010
/bookstore/book[year=2005]/ title	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

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

//book/title | //book/year	Selects all the title AND year elements of all book elements
//title | //author	Selects all the title AND author elements in the document
/bookstore/book/title | //year	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]
```

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
descendent	Selects all descendants (children, grandchildren, etc.) of the current node
descendent-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
parent	Selects the parent of the current node
self	Selects the current node
...	and many more...

XPath axes

Examples:

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

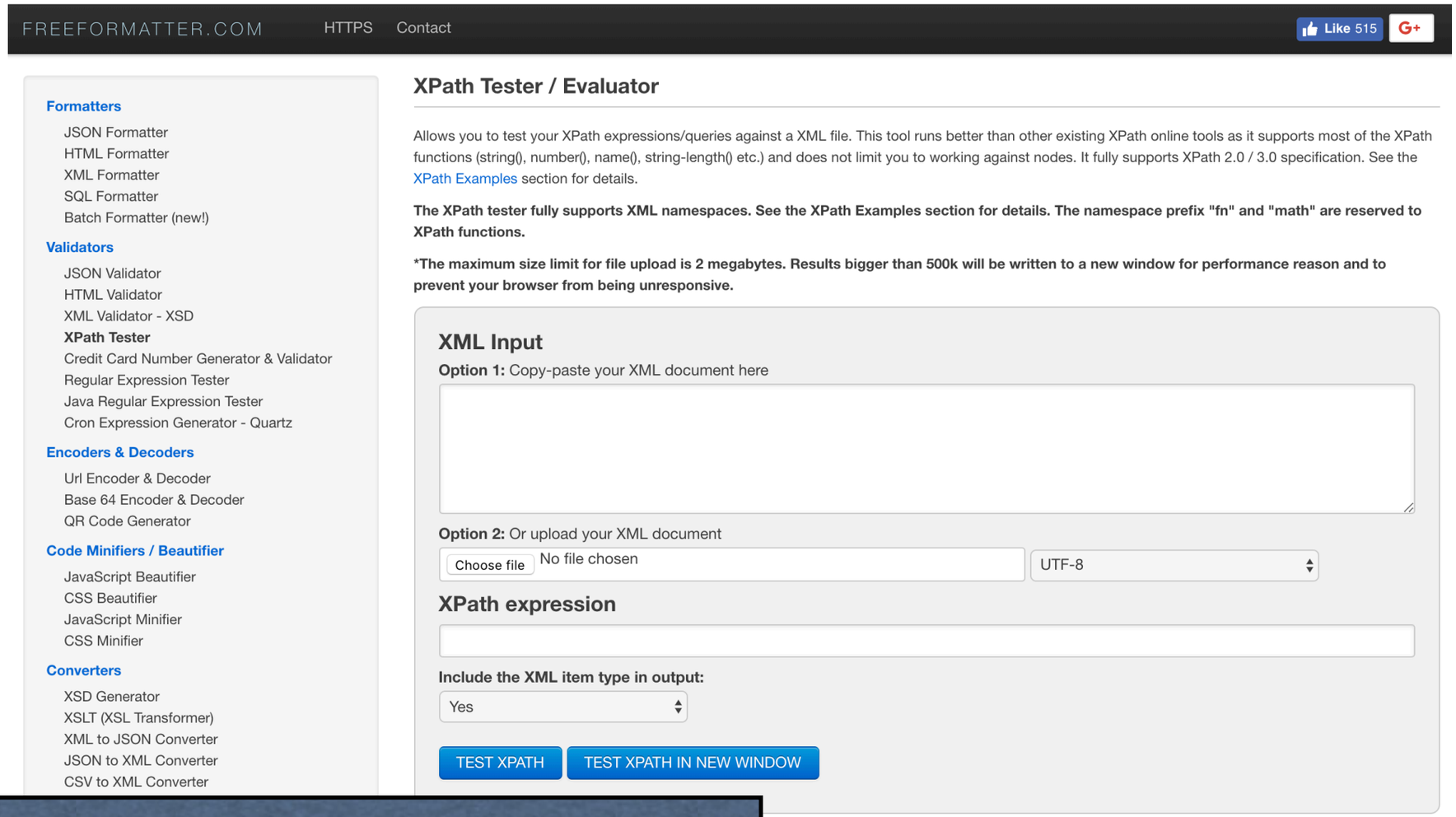
XPath axes

- 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



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

http://www.xpathtester.com/xpath