

# XPath

## XPath? What is it ?

**XPath** is a notation for pointing into and navigating around XML documents

**XPath** is used for uniqueness and scope descriptions in **XML Schema**, pattern matching and selection in **XSLT**, selection and iteration in **XQuery**, and it is also a key component of **Xlink** and **XPointer**

**XPath** started out as a simple language known as **XPath 1.0**; through interactions with **XQuery** query language it evolved into much larger language **XPath 2.0**

## XPath data model

XML document may be viewed as a hierarchical structure called as **XML tree**

**XML tree** consists of various kinds of nodes organized in **n-ary tree**

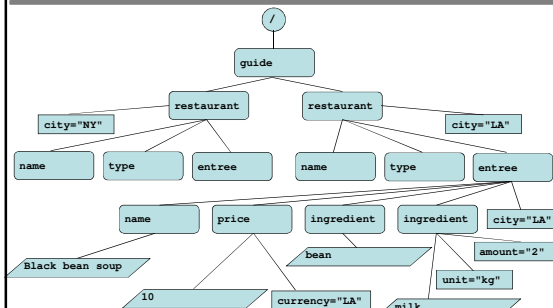
The nodes in **XML tree** can be of the following kinds:

- (1) text nodes that correspond to a fragment of actual information represented by XML document,
- (2) element nodes that define a logical grouping of information represented by its descendants,
- (3) attribute nodes that are refinements of an element name,
- (4) comment nodes,
- (5) processing instruction nodes,
- (6) root node that represents the entire document.

## A sample XML document

```
<guide>
  <restaurant city="LA">
    <name>Blues by the Bay</name>
    <type>vegetarian</type>
    <entree date="01-JAN-2006">
      <name>Black bean soup</name>
      <price currency="AUD">10</price>
      <ingredient amount="2" unit="kg">bean</ingredient>
      <ingredient>milk</ingredient>
    </entree>
    <entree date="02-JAN-2006">
      <name>Asparagus timbale</name>
      <price currency="US$">2.04</price>
      <ingredient>asparagus</ingredient>
      <ingredient>onion</ingredient>
      <ingredient amount="2" unit="cup">sugar</ingredient>
    </entree>
  </restaurant>
  <restaurant city="NY">
    <name>Lazy Lobster</name>
    <type>seafood</type>
    <dinner date="23-OCT-2005">
      <name>Grilled prawns</name>
      ...
    </dinner>
  </restaurant>
</guide>
```

## A sample XML tree



## XPath components

**Location step** consists of: **axis**, **node test**, zero or more **predicates**

**axis::node test [exp-1] [exp-2] [exp-3] ...**

**child::guide**

**child::entree[name="Grilled tomatoes"]**

**child::restaurant[attribute::city="LA"]**

**descendant::restaurant[price > 5.00]**

**parent::entree[attribute::date="07-MAR-2003"]**

05 XPath

XPath components

Location path is a sequence of location steps separated by '/'

/child::guide/  
child::restaurant[name="Lazy Lobster"]

/child::guide/  
child::restaurant[attribute::city="LA"/  
descendant::entree[price > 5.00]]

A location step starts at a context node and evaluates to a sequence of nodes

Location path evaluates to a sequence of nodes in a given XML tree

© Janusz R. Getta CSC1235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 7

05 XPath

Example: location path

Evaluation of location path

/descendant::B/child::E/descendant::F

Context node /

descendant::B

child::E

descendant::F

© Janusz R. Getta CSC1235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 8

05 XPath

Axes

An axis is a sequence of nodes located relative to the context node

XPaths supports 12 axes:

- (1) **child**: the children of the context node
- (2) **descendant**: descendants of the context node
- (3) **parent**: the unique parent node
- (4) **ancestor**: all ancestors of the context node, from the parent to the root
- (5) **following-sibling**: the right-hand siblings of the context node
- (6) **preceding-sibling**: the left-hand siblings of the context node
- (7) **following**: all nodes appearing strictly later in the document than the context node, but excluding descendants
- (8) **preceding**: all nodes appearing strictly earlier in the document than the context node, but excluding ancestors
- (9) **attribute**: all attribute nodes of the context node
- (10) **self**: the context node itself

© Janusz R. Getta CSC1235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 9

05 XPath

Axes

An axis is a sequence of nodes located relative to the context node

XPaths supports 12 axes:

- (11) **descendant-or-self**: the concatenation of the self and descendant sequences
- (12) **ancestor-or-self**: the concatenation of the self and ancestor sequence

Forward axis: **child, descendant, following-sibling, following, self, descendant-or-self**

Backward axis: **parent, ancestor, preceding-sibling, preceding**

© Janusz R. Getta CSC1235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 10

05 XPath

Example: axes

parent

child

© Janusz R. Getta CSC1235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 11

05 XPath

Example: axes

descendants

ancestor

© Janusz R. Getta CSC1235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 12

2

05 XPath

### Example: axes

following-sibling      preceding-sibling

© Janusz R. Getta      CSC1235/MCS9235/CSC1835 Databases, SCIT, Autumn 2015      13

05 XPath

### Example: axes

following      preceding

© Janusz R. Getta      CSC1235/MCS9235/CSC1835 Databases, SCIT, Autumn 2015      14

05 XPath

### Node tests

A **node test** filters the kinds of nodes in the specified axis

Sample node tests:

- (1) **text()** : selects only the character data nodes
- (2) **comment()** : selects only the comment nodes
- (3) **processing-instruction()** : selects only the processing instruction nodes
- (4) **node()** : selects all nodes
- (5) **\*** : selects all nodes of a certain kind depending on the axis in for the node test; for the attribute axis, **\*** selects all attribute nodes; for any other axis, **\*** selects all element nodes
- (6) **name** : selects the nodes with a given name
- (7) **\*:localname** : selects the nodes with the given name in any namespace
- (8) **prefix:\*** : selects the nodes as **\*** but only those in the given namespace

© Janusz R. Getta      CSC1235/MCS9235/CSC1835 Databases, SCIT, Autumn 2015      15

05 XPath

### Predicates

**Predicates** are XPath expressions of various types that are evaluated as Boolean conditions

A **number** corresponds to **true** when its value equals to the current context position; a **string** corresponds to **true** when it has non-zero length; a **sequence** corresponds to **true** when it has non-zero length

```

/
child::guide/
child::restaurant/
child::entree[price > 5.00]/
child::price

/
descendant::entree/
descendant::price[attribute::currency="AU$"]

```

© Janusz R. Getta      CSC1235/MCS9235/CSC1835 Databases, SCIT, Autumn 2015      16

05 XPath

### Typical location paths

Select all restaurant nodes that are children of guide node

```
/child::guide/child::restaurant
```

Select an attribute date in entree node

```
/descendant::entree/attribute::date
```

Select all children nodes of entree node

```
/descendant::entree/child::*
```

Select all attributes of entree node

```
/descendant::entree/attribute::*
```

Select all text child nodes of entree node

```
/descendant::entree/child::text()
```

Select all child nodes of the entree node

```
/descendant::entree/child::node()
```

© Janusz R. Getta      CSC1235/MCS9235/CSC1835 Databases, SCIT, Autumn 2015      17

05 XPath

### Typical location paths

Select a name of each restaurant

```
/descendant::restaurant/child::name
```

Select the currency of each price

```
/descendant::entree/child::price/attribute::currency
```

Select all character data in the guide document

```
/descendant::*/child::text()
```

Select all ingredients that have attribute amount

```
/descendant::ingredient[attribute::amount]
```

Select all ingredients that have a given value attribute amount

```
/descendant::ingredient[attribute::amount=2]
```

Select a name of entree that uses an ingredient onion

```
/descendant::ingredient[self::ingredient="onion"]/parent::entree/child::name/text()
```

© Janusz R. Getta      CSC1235/MCS9235/CSC1835 Databases, SCIT, Autumn 2015      18

05 XPath
<b>Typical location paths</b>
Select all ingredients that have a given value attribute amount and given value of attribute unit
<code>/descendant::ingredient [attribute::amount=2 and attribute::unit="kg"]</code>
Select the first ingredient
<code>/descendant::ingredient[1]</code>
Select the second ingredient
<code>/descendant::ingredient[2]</code>
Select name of restaurant
<code>/descendant::restaurant/child::name/text()</code>
Select all ingredients of entree Asparagus timbale
<code>/descendant::entree[child::name="Asparagus timbale"] /descendant::ingredient</code>
© Janusz R. Getta CSC1235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 19

05 XPath
<b>Abbreviations</b>
If no axis is used then <b>child</b> axis is used as a default
<b>attribute::axis</b> can be replaced with <b>@</b> character
Entire location path fragment
<b>descendant-or-self::node()</b> can be written as <b>//</b>
Character <b>.</b> abbreviates a location step <b>self::node()</b>
String <b>..</b> abbreviates a location step <b>parent::node()</b>
© Janusz R. Getta CSC1235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 20

05 XPath
<b>Typical abbreviated location paths</b>
Select all restaurant nodes
<code>/guide/restaurant</code>
Select the root node
<code>/</code>
Select all children nodes of the root node
<code>/*</code>
Select all attributes of a restaurant node
<code>/guide/restaurant/@*</code>
Select all text child nodes of a entree/name node
<code>/guide/restaurant/entree/name/text()</code>
© Janusz R. Getta CSC1235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 21

05 XPath
<b>Typical abbreviated location paths</b>
Select a name of each restaurant
<code>//restaurant/name</code>
Select the currency of each price
<code>//entree/price/@currency</code>
Select all character data in the guide document
<code>//text()</code>
Select all ingredients that have attribute amount
<code>//ingredient[@amount]</code>
Select all ingredients that have a given value attribute amount
<code>//ingredient[@amount=2]</code>
© Janusz R. Getta CSC1235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 22

05 XPath
<b>Typical abbreviated location paths</b>
Select all ingredients that have a given value attribute amount and given value of attribute unit
<code>//ingredient[@amount=2 and @unit="kg"]</code>
Select the first ingredient
<code>//ingredient[1]</code>
Select the second ingredient
<code>//ingredient[2]</code>
Select the contents of element name of all restaurants
<code>//restaurant/name/text()</code>
Select all ingredients of entree Asparagus timbale
<code>//entree[name="Asparagus timbale"]//ingredient</code>
© Janusz R. Getta CSC1235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 23

05 XPath
<b>Selecting unknown nodes</b>
Select all child nodes of guide element
<code>/guide/*</code>
Select all elements in a document
<code>/*</code>
Select all ingredient elements which have an attribute
<code>//ingredient[@*]</code>
© Janusz R. Getta CSC1235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 24

05 XPath
<b>Selecting several paths</b>
Select all name AND price elements of all entree elements
<code>//entree/name   //entree/price</code>
Select all name AND type elements in a document
<code>//name   //type</code>
© Janusz R. Getta    CSC1235/MCS9235/CSC1835 Databases, SCIT, Autumn 2015    25

05 XPath
<b>References</b>
Elmasri R., Navathe S., Fundamentals of Database Systems, 6th edition, chapter 12 XML: Extensible Markup Language, pp. 420-448
<a href="http://www.uow.edu.au/~jrg/235/HOMEWORK">http://www.uow.edu.au/~jrg/235/HOMEWORK</a>
9.1 How to use XPath query language ?
© Janusz R. Getta    CSC1235/MCS9235/CSC1835 Databases, SCIT, Autumn 2015    26