Ch. 3: XPath Patterns and Expressions

Seoul National University, Internet Database Laboratory

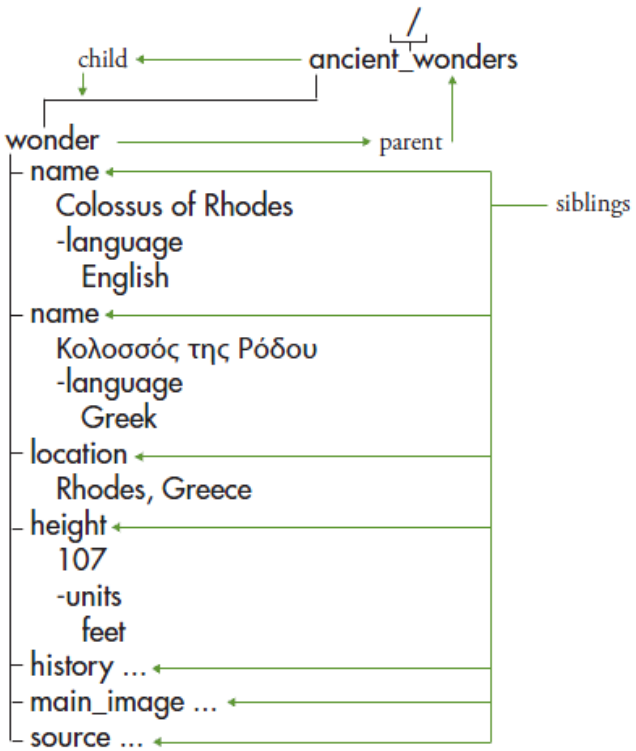
July, 2015

Contents

- XPath
- Locating Nodes
- Determining the Current Node
- Referring to the Current Node
- Selecting a Node's Children
- Conditionally Selecting Nodes
- Creating Absolute Location Paths
- Selecting All the Descendants

Locating Nodes [1/3]

- XPath
  - A language for selecting nodes and node sets by specifying their location paths in the XML document
  - Node: An individual piece of the XML document
    - Element, attribute or some text content
- Everything in the tree is a node
  - Root node
    - The top of the node tree
  - Child node
  - Parent node
  - Sibling node
  - Descendant node
  - Ancestor node



Locating Nodes [2/3]

- Location Paths
  - Relative location path
    - Consists of a sequence of location steps separated by " / "
    - Each step selects a node or node set relative to the current node
  - Absolute location path
    - Relative location path starting at the root node
      - " / " : selects the root node of the XML document
  - Relative location paths are most commonly used
    - They generate the resulting node set relative to the current node
    - This is typically the context in which you are working

Determining the Current Node [1/2]

- Developing an XSLT style sheet
  - Specify what to process next with respect to the current node

XML

```
<?xml version="1.0"?>

<ancient_wonders>
  <wonder>
    <name language="English" >
      Colossus of Rhodes</name>
    <name language="Greek" >
      Κολοσσός της Ρόδου</name>
    <location> Rhodes, Greece </location>
    <height units= "feet"> 107 </height>
    <history> ... </history>
    <main_image ... />
    <source ... />
  </wonder>
</ancient_wonders>
```

XSLT

```
<xsl:template match="/">
...
  <h2>Overview</h2>
 <xsl:apply-templates select=
   "ancient_wonders/wonder">
  <xsl:sort select="height" order=
   "descending" data-type="number" />
 </xsl:apply-templates>
...
</xsl:template>

<xsl:template match="wonder">
 <tr><td><a>...
   <strong><xsl:value-of select=
     "name[@language='English']"/>
   </strong></a><br/>
   <xsl:apply-templates select="
     name[@language!='English']"/>
...
</xsl:template>

<xsl:template match=
  "name[@language!='English']">
  (<em>
    <xsl:value-of select="."/></em>)
</xsl:template>
```

- "." ➔ To refer to the current node

Referring to the Current Node [2/2]

- HTML Result

```
...
<tr><td><a href="#Great Pyramid of Giza">
        <strong>  Great Pyramid of Giza</strong></a>
    <br/></td>
   <td>Giza, Egypt</td>
   <td>455</td></tr>

<tr><td><a href="#Lighthouse of Alexandria">
        <strong>Lighthouse of Alexandria</strong></a>
    <br/> (<em>ο Φάρος τ˜ης ’Αλεξανδρείας</em>)
    </td>
    <td>Alexandria, Egypt</td>
    <td>384</td></tr>

<tr><td><a href="#Mausoleum at Halicarnassus">
   <strong> Mausoleum at Halicarnassus </strong></a>
   <br/> (<em>Μαυσωλε˜ιον ’Αλικαρνασσεύς</em>)
   </td>
   <td>Bodrum, Turkey</td>
   <td>135</td></tr>
...
```

Selecting Node's Children

XML

Fig 3.6

```
...
<history>
<year_built era="BC"> 282 </year_built>
<year_destroyed era="BC">226</year_destroyed>
<how_destroyed> earthquake </how_destroyed>
  <story> In 294 BC, the people of
  the island of Rhodes began
  building a colossal statue of
  the sun god Helios. They
  believed ...
  </story>
</history>
...
```

Fig 3.7

XSLT

```
...
<xsl:template match="history">
...
  was built in
 <xsl:value-of select="year_built"/>
 <xsl:text> </xsl:text>
 <xsl:value-of  select="year_built/@era"/>
 <xsl:choose>
   <xsl:when        test="year_destroyed != 0">
      and was destroyed by
     <xsl:value-of  select="how_destroyed"/> in
     <xsl:value-of  select="year_destroyed"/>
     <xsl:text> </xsl:text>
     <xsl:value-of  select="year_destroyed/@era"/>.
   </xsl:when>
   <xsl:otherwise>
    is still standing today.
   </xsl:otherwise>
 </xsl:choose>
<br \><br \>
</xsl:template>
```

- "xsl:text" element: To add literal text to the output
  - Cannot contain any other elements
  - Often used to handle special characters, such as "&" or ">" or white space

Selecting a Nodes around the current node [2/2]

- "*" : to select all the current node's children
- "..": To select the current node's parent
- ../ sibling: the child of the current node's parent
- ../sibling/ niece: the child of the sibling of the current node
- ../@attribute: Attribute of the parent node
- ../*: All the child elements of the parent of the current node

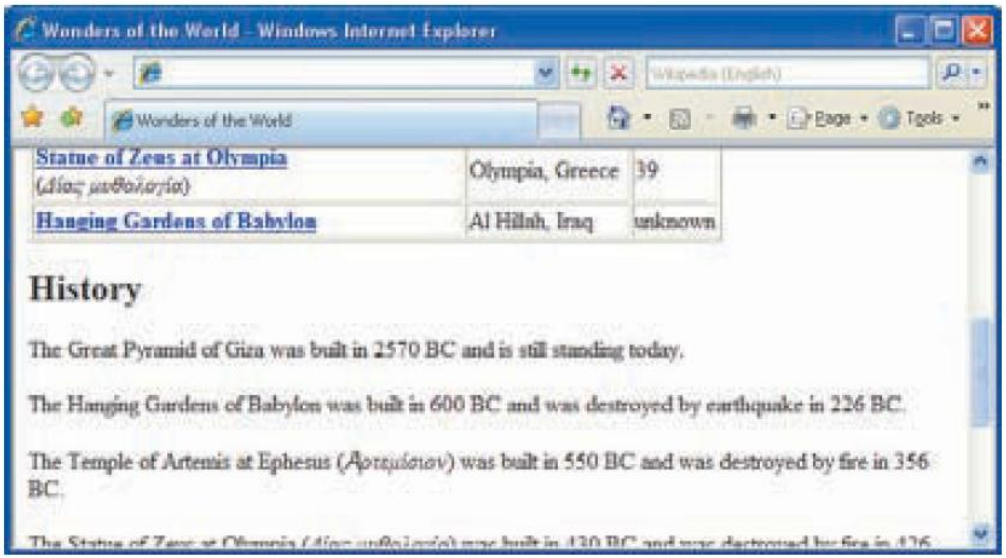
Selecting a Node's Parent

XML

```
...
<wonder>
  <name language="English">
    Colossus of Rhodes</name>
  <name language="Greek">
    Κολοσσός της Ρόδου</name>
...
  <history>
    <year_built era="BC">
      282</year_built>
    <year_destroyed era="BC">
      226</year_destroyed>
    <how_destroyed>
      earthquake</how_destroyed>
    <story>In 294 BC, ...</story>
  </history>
...
```

XSLT

```
...
<xsl:template match="history">
...
  The <xsl:value-of select=
        "../name[@language='English']"/>
  <xsl:apply-templates select=
    "../name[@language!='English']"/>
  was built in <xsl:value-of select="year_built"/>
...
```



Selecting a Node's Attributes

- "/@*" : wildcard to select all the node's attributes
- "/**@attribute**": specify the name of the attribute

XSLT

```
...
<xsl:template match="history">
...
  The <xsl:value-of select=
    "../name[@language='English']"/>
  <xsl:apply-templates select=
    "../name[@language!='English']"/>
  was built in <xsl:value-of
    select="year_built"/>
  <xsl:text> </xsl:text>
  <xsl:value-of
    select = "year_built/@era"/>
...
```

HTML

```
...
<h2>History</h2>
<a name="Great Pyramid of Giza"/>
  The Great Pyramid of Giza was built
  in 2570 BC and is still standing
  today.<br/><br/>
<a name="Hanging Gardens of
  Babylon"/>The Hanging Gardens of
  Babylon was built in 600 BC
...
```

Conditionally Selecting Nodes [1/2]

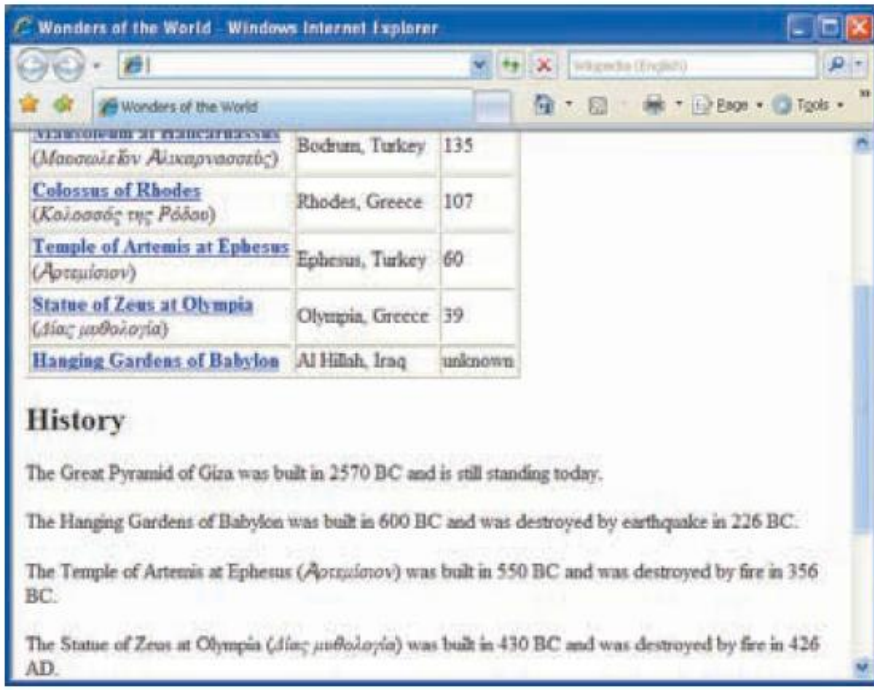
- Conditionally select nodes

XML

```
...
<wonder>
  <name language="English">
    Statue of Zeus at Olympia</name>
  <name language="Greek">
    Κολοσσός της Ρόδου</name>
  <location>Olympia, Greece</location>
  <height units="feet">39</height>
  <history>
...
</wonder>
...
```

XSLT

```
...
<xsl:template match=
   "name[@language!='English']">
  (<em><xsl:value-of select="."/>
    </em>)
</xsl:template>
...
```



Conditionally Selecting Nodes [2/2]

- **[@language]:** Select all the current node's elements that have a language attribute

- Multiple predicates
  - **Name[@language='English'][position()=last()]**
    - Select the name elements that have a language attribute equal to "English" and that are the last node in the set

- **[last()]/@***
    - All the attributes of the last element of the current node set

- Make sure you type square brackets
    - Not curly ones
    - Not parentheses

Creating Absolute Location Paths

- Absolute location paths
  - One that do not rely on the current node
- To create an absolute location path
  - Relative location from root node

XML

```
...
<xsl:template match="wonder">
  <tr><td>
   <a><xsl:attribute name=
     "href">#<xsl:value-of select=
     "name[@language='English']"/>
     </xsl:attribute>
     <strong>

     <xsl:value-of select=
     "/ancient_wonders/wonder/name
     [@language='English']"/>
     </strong></a><br/>
...
```

HTML

```
...
 <a href="#Great Pyramid of
   Giza"><strong>Colossus of Rhodes
   </strong></a><br/></td>
<td>Giza, Egypt</td><td>455</td>
...
 <a href="#Lighthouse of
    Alexandria"><strong>Colossus of
    Rhodes</strong></a>
...
```

Selecting All the Descendants [1/2]

- Type " // "
  - Two forward slashes
  - To select all the descendants of the root node
- Type " .// "
  - A period followed by two forward slashes
  - To select all the descendants of the current node
- " //*/@file "
  - All nodes that have an attribute *named* file

Selecting All the Descendants [2/2]

- Selecting All the Descendants

XML

```
...
 <wonder>
  <name language="English">
    Lighthouse of Alexandria</name>
...
 <main_image file="lighthouse.jpg"
   w="528" h="349"/>
 <source sectionid="112"
   newspaperid="53"/>
</wonder>
...
```

XSLT

```
...
 <xsl:template match="/">
  <html><head><title>Wonders of the
    World</title></head>
   <body>
    <xsl:apply-templates select="//*/@file" />
   </body></html>
 </xsl:template>

<xsl:template match="//*/@file" >
   <xsl:value-of select="."/> <br />
</xsl:template>
...
```