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**
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

- 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>...</main_image>
    <source>...</source>
  </wonder>
</ancient_wonders>
```

- "." ➔ To refer to the current node

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

<xsl:template match="name[@language!='English']">
  (\em<xsl:value-of select="/"/>)
</xsl:template>
```
Referring to the Current Node [2/2]

- HTML Result

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

<tr><td><a href="#Lighthouse of Alexandria">
    <strong>Lighthouse of Alexandria</strong></a>
    <br/>
    (ο Φάρος της 'Αλεξανδρείας)
    Alexandria, Egypt</td>
    <td>384</td></tr>

<tr><td><a href="#Mausoleum at Halicarnassus">
    <strong>Mausoleum at Halicarnassus</strong></a>
    (Μαυσωλείον 'Αλικαρνασσεύς)
    Bodrum, Turkey</td>
    <td>135</td></tr>

...
“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

- "*": 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>
</wonder>

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:template>...
Selecting a Node’s Attributes

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

XSLT

```xml
... 
<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

```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>
        ...
    </history>
</wonder>
```

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

XSLT

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

---

**Statue of Zeus at Olympia**

- **Location:** Olympia, Greece
- **Height:** 39 feet
- **History:**

**Colette of Rhodes**

- **Location:** Rhodes, Greece
- **Height:** 107 feet
- **History:**

**Temple of Artemis at Ephesus (Artemision)**

- **Location:** Ephesus, Turkey
- **Height:** 60 feet
- **History:**

**Statue of Zeus at Olympia (Dias μυθολογία)**

- **Location:** Olympia, Greece
- **Height:** 39 feet
- **History:**

**History**

- **The Great Pyramid of Giza**
  - Built in 2570 BC and is still standing today.
- **The Hanging Gardens of Babylon**
  - Built in 600 BC and was destroyed by earthquake in 226 BC.
- **The Temple of Artemis at Ephesus (Artemision)**
  - Built in 550 BC and was destroyed by fire in 356 BC.
- **The Statue of Zeus at Olympia (Dias μυθολογία)**
  - Built in 430 BC and was destroyed by fire in 426 AD.
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

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

- Selecting All the Descendants

**XML**

```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**

```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>
```

![Screenshot of the webpage](image-url)