

XPath 2.0 확장

(6주차)

학습개요

- 학습 목표

- XPath 2.0의 데이터 모델을 설명할 수 있다.
- XPath 2.0 표현식이 반환하는 시퀀스의 특징에 대해 설명할 수 있다.
- XPath 2.0 표현식에서 사용하는 함수에 대해 설명할 수 있다.

- 학습 내용

- XPath 2.0 소개
- XPath 2.0 데이터 모델
- XPath 2.0 시퀀스
- XPath 2.0 표현식과 함수
- XPath 2.0 실습

XPath 2.0

- W3C 2007년 1월 권고안 발표
- 데이터 모델의 근간 변화
 - 노드 집합(1.0) => 시퀀스(2.0)
- XPath 2.0 표현식에서 사용 가능한 조건, 반복, 향상된 함수 제공
- XSLT 2.0와 XQuery 1.0에서 사용하도록 설계
- XPath 1.0의 슈퍼 셋으로 XPath 1.0와 호환성을 유지
- <http://www.w3.org/TR/xpath20/>

XDM

- XPath 2.0 Data Model
- XML 스키마에 대한 지원
 - 스키마를 기반으로 맞춤법, 경로, 타입 에러 등에 대한 경고를 지원
 - 값의 타입에 대해 엄격해짐
 - 예를 들어 문자열에 대해 산술 연산을 하면 오류 발생

시퀀스 타입

- XPath 2.0 표현식의 반환값은 시퀀스
- 시퀀스는 구분자로 쉼표를 사용하며, 여러 항목(노드 또는 원자값)의 목록
(1, 'XPath 2.0', 20000)
- 시퀀스는 중첩의 의미가 없음
(1, 'XPath 2.0', ('홍길동', 225), 20000) => (1, 'XPath 2.0', '홍길동', 225, 20000)
- 연산의 반환값
(1 to 5) => (1, 2, 3, 4, 5)
- 함수의 반환값
reverse(1 to 5) => (5, 4, 3, 2, 1)

avg()

[XML 1.0의 경우]

```
...  
<td>  
<xsl:value-of select="format-number(sum(products/product/price) div  
count(products/product/price[.!=0]), '###,##0')"/>  
</td>  
...
```

[XML 2.0의 경우]

```
...  
<td>  
<xsl:value-of select="format-number(avg(products/product/price[.!=0]), '###,##0')"/>  
</td>  
...
```

min(), max()

```
...  
<td>  
<xsl:value-of select="format-number(min(products/product/price[!=0]), '###,##0')"/>  
</td>  
<td>  
<xsl:value-of select="format-number(max(products/product/price[!=0]), '###,##0')"/>  
</td>  
...
```

upper-case(), lower-case()

```
...  
<xsl:for-each select="//product">  
  <tr>  
    <td>  
      <xsl:value-of select="lower-case(name)"/>  
    </td>  
    <td>  
      <xsl:value-of select="upper-case(name)"/>  
    </td>  
  </tr>  
</xsl:for-each>  
...
```


if-then-else

```
...  
<xsl:variable name="ascending" as="xs:boolean" select="true()" />  
<xsl:for-each select="//product">  
  <xsl:sort select="price" data-type="number" order="{if ($ascending) then  
    'ascending' else 'descending'}"/>  
  <tr>  
    <td>  
      <xsl:value-of select="name"/>  
    </td>  
    <td align="right">  
      <xsl:value-of select="format-number(price, '###,##0')"/>  
    </td>  
  </tr>  
</xsl:for-each>  
...
```

distinct-values()

```
...  
<xsl:for-each select="distinct-values(products/product/name)">  
  <xsl:sort select="." order="ascending"/>  
  <xsl:value-of select="."/>  
  <xsl:choose>  
    <xsl:when test="position()=last()"><br/></xsl:when>  
    <xsl:when test="position()=last()-1">,</xsl:when>  
    <xsl:otherwise>,</xsl:otherwise>  
  </xsl:choose>  
</xsl:for-each>  
...
```

for

```
...  
<xsl:variable name="status"  
  select="for $i in (//product)  
    return (if ($i/stock != 0 and $i/price != 0) then 1 else 0)"/>  
<xsl:text>주문가능한 항목 수: </xsl:text>  
<xsl:value-of select="sum($status)"/> <br/>  
...
```

current-date(), current-time(), current-dateTime()

```
...  
<xsl:variable name="orderYear" select="format-date(current-date(), '[Y0001'])  
  cast as xs:integer" />  
<xsl:variable name="orderMonth" select="format-date(current-date(), '[M01'])  
  cast as xs:integer" />  
<xsl:variable name="orderDate" select="format-date(current-date(), '[D01'])  
  cast as xs:integer" />  
<xsl:text>주문일자: </xsl:text>  
<xsl:value-of select="concat($orderYear, '/', $orderMonth, '/', $orderDate)"/>  
<xsl:text>배송일자: </xsl:text>  
<xsl:value-of select="concat($orderYear, '/', $orderMonth, '/', $orderDate + 1)"/>  
...
```

학습정리

- XPath 2.0은 데이터 모델의 근간을 노드 집합에서 시퀀스로 변경함
- 시퀀스는 구분자로 쉼표를 사용하며, 여러 항목(노드 또는 원자 값)의 목록
- XPath 2.0은 표현식에서 사용 가능한 if-then-else, for, avg(), min(), max(), distinct-values() 등의 향상된 함수를 제공
- XPath 2.0은 XSLT 2.0과 XQuery 1.0에서 사용
- XPath 2.0은 XPath 1.0의 슈퍼 셋으로 XPath 1.0과 호환성을 유지