

제 7 장 XPath 이해하기

- 1 절. XPath 소개
- 2 절. XPath 데이터 모델
- 3 절. Location Path
- 4 절. Location Step
- 5 절. 단축형 Location Step
- 6 절. XPath 함수
- 7 절. XPath를 이용하는 스펙들

1. XPath 소개

- XPath는 XML문서에 대한 경로를 지정하기 위한 언어로, Version 1.0이 최신 권고안이다.
- XPath는 XSLT와 XPointer 언어에 사용될 목적으로 설계된 언어이다.
- XPath는 디렉토리 구조와 유사한 형태로 XML 문서의 특정 노드에 접근한다.(by 절대/상대 경로)
- XPath에서는 문자열, 숫자 또는 boolean 데이터 타입과 관련된 기본적인 함수들도 가지고 있다.

2. XPath의 데이터 모델

2-1 트리(Tree) 구조

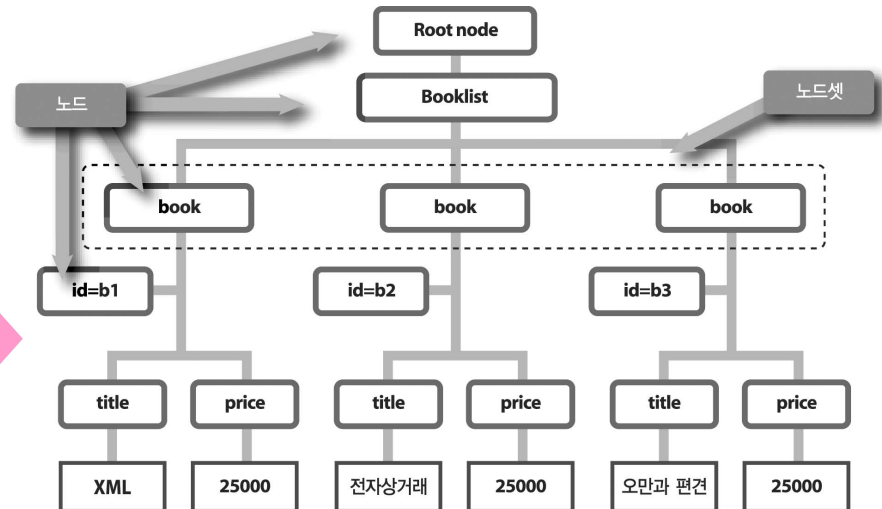
- XPath는 XML 문서를 한 개의 트리(Tree) 모델로 취급한다.

2-2 노드와 노드셋의 이해

- 노드 : 트리 구조에서 매듭에 해당한다.
- 노드셋(Node-set) : 동일 종류의 노드들의 집합이다.
- XML 문서에서의 구성 요소를 노드 또는 노드셋으로 표현할 수 있다.

XML 문서

```
<?xml version="1.0" encoding="euc-kr"?>
<booklist>
  <book id="b1">
    <title>XML</title>
    <price>25000</price>
  </book>
  <book id="b2">
    <title>전자상거래</title>
    <price>25000</price>
  </book>
</booklist>
```



2. XPath의 데이터 모델

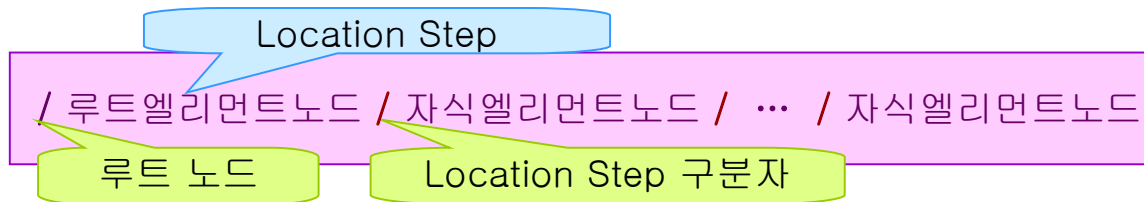
- XPath의 데이터 모델을 구성하는 노드들.
- 노드들은 XPath 표현식에 의해 선택, 조작될 수 있는데, XPath 표현식은 Location Path, 그리고 XPath 함수들로 구성된다.

노드 종류	설명
루트 노드 (root node)	<ul style="list-style-type: none">• XML 문서 자체를 표현하는 최상위 노드를 말한다.• 루트 엘리먼트의 부모 노드이다.
엘리먼트 노드 (element node)	<ul style="list-style-type: none">• 엘리먼트를 표현하는 노드이다.• 루트 엘리먼트도 엘리먼트 노드이다.
속성 노드 (attribute node)	<ul style="list-style-type: none">• 속성(attribute)을 표현하는 노드이다.• 네임스페이스 선언은 속성 노드로 표현하지 않는다.
텍스트 노드 (text node)	<ul style="list-style-type: none">• 콘텐츠 내용인 문자 데이터를 표현하는 노드이다.• 트리 구조상 최말단에 있는 노드이다.
네임스페이스 노드 (namespace node)	<ul style="list-style-type: none">• 네임스페이스 선언을 표현하는 노드이다.
프로세싱 지시자 노드(processing instruction node)	<ul style="list-style-type: none">• 프로세싱 지시자를 표현하는 노드이다.
주석 노드 (comment node)	<ul style="list-style-type: none">• 주석을 표현하는 노드이다.

3. Location Path

- Location Path는 XML 문서의 각 노드들에 대한 위치를 지정하기 위한 XPath의 표현식이다.
- Location Path는 절대 경로(Absolute Location Path)와 상대 경로(Relative Location Path)로 나누어 진다.

Location Path의 구성



(1) 절대 경로

- 루트 노드부터 시작하는 경로를 절대 경로라 한다. 절대 경로는 반드시 '/'로 시작해야 한다.

문법	/루트엘리먼트노드/자식엘리먼트노드/.../자식엘리먼트노드
----	---------------------------------

(2) 상대 경로

- 현재 노드부터 시작하는 경로를 상대 경로라 한다. 다음은 다양한 상대 경로를 지정 방법이다.

문법	<ul style="list-style-type: none">① 자식엘리먼트노드/.../자식엘리먼트노드 또는 ./자식엘리먼트노드/.../자식엘리먼트노드② ../형제엘리먼트노드③ ../형제엘리먼트노드/ 자식엘리먼트노드/.../자식엘리먼트노드
----	---

3. Location Path

(3) 절대 경로와 상대 경로의 예제

XML 문서	<pre><booklist> <book id="b1"> <title>XML 기초</title> <author>이규미</author> <price>25000</price> </book> </booklist></pre>
XML 문서를 트리 구조로 모델링	<pre>graph TD Booklist[Booklist] --> book[book] book --> title[title] book --> author[author] book --> price[price] title --> title_text[XML 기초] author --> author_text[이규미] price --> price_text[25000]</pre>
절대 경로 예	price 노드를 지정하는 절대 경로 /booklist/book/price
상대 경로 예	현재의 위치가 book에 있을 경우, 하위의 price를 상대 경로로 표현 ./price 또는 price

3. Location Path

3-2 Location Step 작성 문법

문법	Axis::NodeTest[Predicate]
구성 요소	설명
Axis	노드를 찾기 위한 검색 방향을 지정한다.
NodeTest	찾을 노드의 이름을 지정한다
Predicate	필터링을 하기 위한 표현식을 기술한다.

3. Location Path

(1) Axis의 종류

Axis	설명
ancestor	▪ 조상 노드들
ancestor-or-self	▪ 노드 자신과 조상 노드들
attribute	▪ 노드에 속한 애트리뷰트 노드들 ▪ 기준이 되는 노드(context node)가 엘리먼트 노드일 경우에만 의미가 있는 축임 ▪ 단축 구문으로 사용이 가능함 (@기호 사용) ex) child::book/attribute::id → book/@id로 표현함
child	▪ 자식 노드들 ▪ 단축 구문으로 사용이 가능함 (child:: 생략 가능) ex) child::book/child::title → book/title로 표현함
descendant	▪ 자손 노드들
descendant-or-self	▪ 노드 자신과 자손 노드들 ▪ 단축 구문으로 사용이 가능함 (//기호 사용) ex) /descendant-or-self::node()/child::book → //book으로 표현하여 모든 book 요소를 검색함

지정된 축의 모든 노드를 선택

3. Location Path

(1) Axis의 종류 - 계속

Axis	설명
following	<ul style="list-style-type: none">▪ XML 문서에서의 순서상 다음에 오는 노드들▪ 자손 노드들과 조상 노드들은 제외됨
following-sibling	<ul style="list-style-type: none">▪ XML 문서를 트리로 표현했을 때 다음에 오는 형제 노드들
namespace	<ul style="list-style-type: none">▪ 노드에 속한 네임스페이스 노드들▪ 기준이 되는 노드(context node)가 엘리먼트 노드일 경우에만 의미가 있는 축임
parent	<ul style="list-style-type: none">▪ 부모 노드▪ 단축 구문으로 사용이 가능함 (.. 기호 사용) ex) parent::node()/child::title → ../title 로 표현함
preceding	<ul style="list-style-type: none">▪ XML 문서에서의 순서상 전에 오는 노드들▪ 자손 노드들과 조상 노드들은 제외됨
preceding-sibling	<ul style="list-style-type: none">▪ XML 문서를 트리로 표현했을 때 전에 오는 형제 노드들
self	<ul style="list-style-type: none">▪ 노드 자신▪ 단축 구문으로 사용이 가능함 (. 기호 사용) ex)self::node()/descendant-or-self::node()/child::title → ../title 로 표현함

3. Location Path

(2) NodeTest

NodeTest	설명
comment()	▪ 지정된 축의 모든 주석 노드들을 선택한다.
text()	▪ 지정된 축의 모든 텍스트 노드들을 선택한다.
processing-instruction()	▪ 지정된 축의 모든 processing-instruction 노드들을 선택한다.
processing-instruction(타겟명)	▪ 지정된 타겟명과 동일한 이름을 갖는 processing-instruction 노드들을 선택.
node()	▪ 지정된 축의 모든 노드들을 선택한다.
*	▪ 모든 노드들을 선택한다.
접두사:*	▪ 지정된 네임스페이스 접두사를 갖는 모든 노드들을 선택한다. ▪ 네임스페이스 선언이 없으면 접두사는 생략 가능하다.
접두사:로컬이름	▪ 지정된 네임스페이스 접두사를 갖고 로컬이름과 동일한 노드들을 선택한다. ▪ 네임스페이스 선언이 없으면 접두사는 생략 가능하다.

3. Location Path

(3) Predicate에 올 수 있는 표현

- Predicate에는 NodeTest에서 선택된 노드 중에서 특별한 노드를 선택하기 위한 필터링과 관련된 표현이 온다.

노드명[노드명 연산자 “값”] 또는 노드명[@속성명 연산자 “값”]


- 주로 사용되는 Predicate 표현

NodeTest[Predicate 표현]	설명
노드명[노드명=”값”]	자식 노드가 주어진 값을 콘텐츠 내용을 가지고 있는 노드 선택
노드명[@속성명=”값”]	주어진 속성값을 가지고 있는 노드 선택

3. Location Path

(4) 연산자의 종류

- 다음은 XPath의 표현식에서 사용할 수 있는 연산자들의 목록이다.

데이터 타입	연산자		우선순위
node-set	프레디카드(predicate)	‘[‘ 와 ‘]’	높음
node-set	경로 연산	/ //	
node-set	유니온(union)연산		
숫자	음의 부호	-	
숫자	승제 연산	* div mod	
숫자	가감 연산	+ -	
Boolean	비교 연산	< > <= >=	
Boolean	등가 비교 연산	= !=	
Boolean	AND 연산	and	
Boolean	OR 연산	or	낮음

4. Location Step

5. 단축형 Location Step

6. XPath 함수

6-1 XPath 함수의 종류

- XPath 표현식에 사용할 수 있는 함수의 종류
 - 노드 집합과 관련된 함수들(Node Set Functions)
 - 문자열과 관련된 함수들 (String Functions)
 - 참 또는 거짓과 관련된 함수들(Boolean Functions)
 - 숫자와 관련된 함수들(Number Functions)

6-2 노드 집합과 관련된 함수들

함수명	설명
<i>number last()</i>	마지막 노드의 위치를 리턴한다.
<i>number position()</i>	현재 노드(context node)의 위치를 리턴한다.
<i>number count(node-set)</i>	노드의 수를 리턴한다.
<i>node-set id(object)</i>	고유한 ID에 의해 엘리먼트들을 선택한다.
<i>string local-name(node-set?)</i>	노드 이름의 네임스페이스 정보를 제외한 로컬 부분을 되돌린다.
<i>string name(node-set?)</i>	노드의 네임스페이스 정보를 포함한 Qname을 리턴한다.
<i>string namespace-uri(node-set?)</i>	노드 이름의 네임스페이스 URI를 리턴한다.

6. XPath 함수

(1) *number last()*

- last() 함수는 마지막 노드의 위치(position) 값을 리턴한다.

XSL
문서

```
<xsl:apply-templates select="booklist/book[position()=last()]"> 또는  
<xsl:apply-templates select="booklist/book[last()]">
```

(2) *number position()*

- position() 함수는 현재 노드의 위치값을 리턴한다.

XSL
문서

```
<xsl:apply-templates select="booklist/book[position()=2]"> 또는  
<xsl:apply-templates select="booklist/book[2]">
```

(3) *number count(node-set)*

- count() 함수는 인자로 받은 노드셋(노드 집합)에 있는 노드수를 리턴한다.

XSL
문서

```
<xsl:value-of select="count(booklist/book)" />
```


6. XPath 함수

6-3 문자열과 관련된 함수들

함수명	설명
<i>string</i> string (<i>object</i> ?)	▪ 지정된 객체의 값을 <i>string</i> 타입으로 변환하여 그 값을 리턴한다.
<i>string</i> concat (<i>string</i> , <i>string</i> , <i>string</i> *)	▪ 지정된 문자열을 연결(concatenation)하여 리턴한다.
<i>boolean</i> starts-with (<i>string</i> , <i>string</i>)	▪ 첫번째 인자로 지정된 문자열이 두 번째 인자의 문자열로 시작하는지를 검사하여 <i>true</i> 또는 <i>false</i> 를 리턴한다.
<i>boolean</i> contains (<i>string</i> , <i>string</i>)	▪ 첫번째 인자로 지정된 문자열이 두 번째 인자의 문자열을 포함하고 있는지를 검사하여 <i>true</i> 또는 <i>false</i> 를 리턴한다.
<i>string</i> substring-before (<i>string</i> , <i>string</i>)	▪ 첫번째 인자로 지정된 문자열에서 두 번째 인자의 문자열이 출현하는 첫번째 위치를 찾아서 그 앞에 오는 문자열들을 리턴한다. ▪ 첫번째 인자로 지정된 문자열이 두 번째 인자의 문자열을 포함하고 있지 않으면 빈 문자열(empty string)을 리턴한다.

6. XPath 함수

6-3 문자열과 관련된 함수들 - (계속)

함수명	설명
<i>string</i> substring-after (<i>string</i> , <i>string</i>)	<ul style="list-style-type: none">첫번째 인자로 지정된 문자열에서 두 번째 인자의 문자열이 출현하는 첫번째 위치를 찾아서 그 뒤에 오는 문자열들을 리턴한다.첫번째 인자로 지정된 문자열이 두 번째 인자의 문자열을 포함하고 있지 않으면 빈 문자열(empty string)을 리턴한다.
<i>string</i> substring (<i>string</i> , <i>number</i> , <i>number</i> ?)	<ul style="list-style-type: none">첫번째 인자로 지정된 문자열 중에서 두 번째 인자로 지정된 위치부터 세 번째 인자로 지정된 길이만큼 문자열을 추출하여 리턴한다.문자열의 첫번째 문자 위치는 1이다.세 번째 인자가 지정되지 않으면 두 번째 인자로 지정된 위치부터 문자열 끝까지의 내용을 추출하여 리턴한다.
<i>number</i> string-length (<i>string</i> ?)	<ul style="list-style-type: none">지정된 문자열이 포함하고 있는 문자의 개수를 리턴한다.
<i>string</i> normalize-space (<i>string</i> ?)	<ul style="list-style-type: none">문자열의 제일 앞과 제일 뒤에 오는 공백 문자를 제거하고, 문자열 중간에 오는 공백 문자들을 하나의 공백 문자(0x20)로 치환하여 리턴한다.
<i>string</i> translate (<i>string</i> , <i>string</i> , <i>string</i>)	<ul style="list-style-type: none">첫번째 인자의 문자열에서 두 번째 인자 문자열에 대해 세 번째 인자의 문자열로 대체시킨 문자열을 리턴한다.

6. XPath 함수

(1) *string* string(*object*?)

- string() 함수는 지정된 객체의 값을 string 타입으로 변환하여 그 값을 리턴한다.
 - 인자가 노드셋일 경우에는 첫번째 노드의 문자열 값을 리턴한다.
 - 숫자는 숫자 변환 규칙에 따라 문자열로 변환한다.
 - 0은 문자열 0으로 변환
 - 양수 무한대는 문자열 'Infinity'로 변환, 음수 무한대는 문자열 '-Infinity'로 변환
 - 정수는 소수점 없이 문자열로 변환
 - 실수는 소수점을 포함하는 문자열로 변환
 - boolean false는 문자열 false로, boolean true는 문자열 true로 변환한다.

XSL
문서

```
<xsl:value-of select="string(book/title)" />
```

6. XPath 함수

(2) *string* concat(*string*, *string*, *string**)

- concat() 함수는 인자로 넘어온 문자열들을 연결하여 새로운 문자열로 리턴한다.

XSL
문서

```
<xsl:value-of  
    select="concat(book/title, book/author, book/publisher)" />
```

(3) *boolean* starts-with(*string*, *string*)

- starts-with() 함수는 첫번째 파라미터가 두 번째 파라미터 문자열로 시작하는 경우에는 true를 리턴하고, 그렇지 않은 경우에는 false를 리턴한다.

XSL
문서

```
<xsl:apply-templates  
    select="/booklist/book[starts-with(title, '사')]" >
```

(4) *boolean* contains(*string*, *string*)

- contains() 함수는 첫번째 인자가 두 번째 인자의 문자열을 포함하고 있으면 true를 리턴하고, 그렇지 않은 경우에는 false를 리턴한다.

XSL
문서

```
<xsl:apply-templates  
    select="/booklist/book[contains(title, 'XML')]" >
```

6. XPath 함수

(5) *string* substring-before(*string*, *string*)

- 첫번째 인자로 지정된 문자열 중에서, 두 번째 인자로 지정된 문자열을 만나기 전까지의 문자열들을 리턴한다. 그러나 첫번째 인자로 지정된 문자열이 두 번째 인자의 문자열을 포함하고 있지 않으면 빈 문자열(empty string)을 리턴한다.

XSL
문서

```
<xsl:value-of select="substring-before(title, 'XML')" />
```

(6) *string* substring-after(*string*, *string*)

- 첫번째 인자로 지정된 문자열에서 두 번째 인자의 문자열 뒤에 오는 문자열들을 리턴한다.

XSL
문서

```
<xsl:value-of select="substring-after(title, '사랑')" />
```

(7) *string* substring(*string*, *number*, *number?*)

- 첫번째 인자 문자열에 대해 두 번째 인자에서 지정한 포인터부터 세 번째 인자에서 지정한 길이만큼의 문자열을 리턴한다. 세 번째 인자를 지정하지 않으면, 두 번째 인자에서부터 문자열 끝까지를 리턴하게 된다.

XSL
문서

```
<xsl:value-of select="substring(title, 4, 6)" />
```

6. XPath 함수

(8) *number* string-length(*string?*)

- string-length() 함수는 문자열의 문자 개수를 리턴한다.

XSL
문서

```
<xsl:value-of select="string-length(title)" />
```

(9) *string* normalize-space(*string?*)

- normalize-space() 함수는 문자열의 앞과 뒤에 오는 공백 문자를 제거하고, 인접한 하나 이상의 공백 문자들을 하나의 공백 문자로 변환한 문자열을 리턴한다.

XSL
문서

```
<xsl:value-of select="normalize-space(title)" />
```

(10) *string* translate(*string*, *string*, *string*)

- translate() 함수는 첫번째 인자의 문자열에 포함되어 있는 두 번째 인자 문자열을 세 번째 인자 문자열로 대체시킨 문자열을 리턴한다.

XSL
문서

```
<xsl:value-of select="translate(title, 'XML',  
'xml' )" />
```

6. XPath 함수

6-4 참 또는 거짓과 관련된 함수들

함수명	설명
<i>boolean</i> boolean (<i>object</i>)	▪ 지정된 객체의 값을 boolean 타입으로 변환하여 그 값을 리턴한다.
<i>boolean</i> not (<i>boolean</i>)	▪ 지정된 값이 true이면 false, false이면 true를 리턴한다.
<i>boolean</i> true ()	▪ true를 리턴한다.
<i>boolean</i> false ()	▪ false를 리턴한다.
<i>boolean</i> lang (<i>string</i>)	▪ 기준이 되는 노드에서 유효한 ‘xml:lang’ 속성값이 지정된 문자열과 일치하면 true를, 그렇지 않으면 false를 리턴한다. ▪ ‘xml:lang’ 속성값은 자신이 속한 엘리먼트 뿐만 아니라 그 엘리먼트의 자손 엘리먼트들에 대해서도 유효하다.

6. XPath 함수

(1) *boolean* boolean(*object*)

- boolean() 함수는 인자로 지정된 객체의 값을 boolean 타입으로 변환하여 그 값을 리턴한다.
 - 인자가 숫자일 경우 0은 false로, 그 외의 모든 경우에는 true로 변환한다.
 - 노드셋은 빈 노드셋(empty node-set)인 경우에는 false, 그 외의 경우에는 true로 변환한다.
 - 문자열은 문자열의 길이가 0인 경우에는 false, 그 외의 경우에는 true로 변환한다.

XSL 문서	<xsl:value-of select="boolean(title)" />
--------	--

(2) *boolean* not(*boolean*)

- not() 함수는 인자로 지정된 객체의 값이 true이면 false로, false이면 true로 지정한다.

XSL 문서	<xsl:value-of select="not(title)" />
--------	--------------------------------------

(3) *boolean* true()

- true() 함수는 true값을 리턴한다.

XSL 문서	<xsl:value-of select="true()" />
--------	----------------------------------

(4) *boolean* false()

- false() 함수는 false값을 리턴한다.

XSL 문서	<xsl:value-of select="false()" />
--------	-----------------------------------

6. XPath 함수

6-5 숫자와 관련된 함수들

함수 명	설명
<i>number</i> number (<i>object?</i>)	▪ 지정된 객체의 값을 <i>number</i> 타입으로 변환하여 그 값을 리턴한다.
<i>number</i> sum (<i>node-set</i>)	▪ 지정된 노드셋에 속하는 모든 노드들의 스트링-값을 <i>number</i> 타입으로 변환한 후 그 값들을 더하여 리턴한다.
<i>number</i> floor (<i>number</i>)	▪ 지정된 값보다 작거나 같은 정수 값들 중에서 가장 큰 값을 리턴한다.
<i>number</i> ceiling (<i>number</i>)	▪ 지정된 값보다 크거나 같은 정수 값들 중에서 가장 작은 값을 리턴한다.
<i>number</i> round (<i>number</i>)	▪ 지정된 값과 가장 가까운 정수 값을 리턴한다. ▪ 지정된 값과 가장 가까운 정수 값이 둘이면 그 중 큰 값을 리턴한다.

6. XPath 함수

(1) *number* number(*object?*)

- number() 함수는 인자로 지정된 객체의 값을 number 타입으로 변환하여 그 값을 리턴한다. 객체를 숫자(number)형으로 변환할 경우 다음과 같은 규칙을 따른다.
 - 문자열은 가장 근접한 숫자로 변환된다.
 - boolean 형의 true는 1로, false는 0으로 변환된다.

XSL 문서	<xsl:value-of select="number(price)" />
--------	---

(2) *number* sum(*node-set*)

- sum() 함수는 노드셋 인자에서 각 노드들의 콘텐츠 내용을 숫자로 변환시킨후 합을 리턴한다.

XSL 문서	<xsl:value-of select="sum(/booklist/book/price)" />
--------	---

6. XPath 함수

(3) *number floor(number)*

- floor() 함수는 인자로 지정된 값보다 작거나 같은 정수 값들 중에서 가장 큰 값을 리턴한다. 즉 소수점 버림에 해당된다.

XSL 문서	<xsl:value-of select="floor(price)" />
--------	--

(4) *number ceiling(number)*

- ceiling() 함수는 인자로 지정된 값보다 크거나 같은 정수 값들 중에서 가장 작은 값을 리턴한다. 즉 소수점 무조건 올림에 해당된다.

XSL 문서	<xsl:value-of select="ceiling(price)" />
--------	--

(5) *number round(number)*

- round() 함수는 인자로 지정된 값과 가장 가까운 정수를 리턴한다. 즉 소수 반올림에 해당한다(.5는 반올림된다).

XSL 문서	<xsl:value-of select=" round(price)" />
--------	---

7. XPath를 이용하는 스펙들