

# Xsl / XPath

김 현

전통문화연구회 AI 고전번역학 강의

[hyeon@aks.ac.kr](mailto:hyeon@aks.ac.kr)



이 저작물(PPT)의 인용 표시 방법:  
김현, 「Xsl/XPath」, 전통문화연구회 'AI 고전번역학' 강의 자료 (2026)



1. XSL이란?

2. XSL Elements

3. XSL에 의한 문서 표현

4. XPath란?

5. 필터

6. Xpath 함수

### ❖ XSL (eXtensible Stylesheet Language)

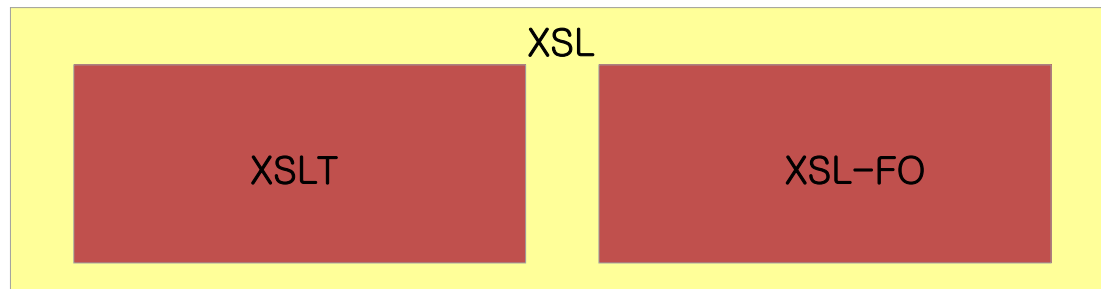
- 문서의 스타일을 정의하기 위한 언어

### ❖ XSLT (XSL Transformation)

- XML 문서의 구조를 다른 구조로 변환시키는 마크업 언어
- XML → HTML, XML → WML, XML → XML

### ❖ XSL-FO (XSL Formatting Objects)

- XML 문서를 非 XML 문서(XML 권고안 문법에 따라 작성된 텍스트 형식의 파일이 아닌 문서)로 변환하는 방법
- XML 문서 → PDF 문서



## ❖ XML 문서

```
<?xml version="1.0" encoding="UTF-8"?>  
<?xml-stylesheet type="text/xsl" href="mystyle.xsl"?>  
.....
```

## ❖ XSL 문서 (mystyle.xsl)

```
<?xml version="1.0" encoding="utf-8"?>  
  
<xsl:stylesheet version="1.0"  
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">  
  .....  
</xsl:stylesheet>
```



1. XSL이란?

2. XSL Elements

3. XSL에 의한 문서 표현

4. XPath란?

5. 필터

6. Xpath 함수

## ❖ 기본 요소

- **xsl:stylesheet**
  - XSLT의 최상위 요소
- **xsl:output**
  - XSLT를 통해 변환된 결과가 어떠한 문서인지를 표시(XML, HTML, TEXT)
  - Indent 속성 ..... 요소 사이의 공백 표시/제거
- **xsl:template**
  - XML 문서의 어느 부분을 어떻게 변환할 것인지 결정한다.
  - match 속성을 이용하여 문맥 노드의 위치를 XPath 표현식으로 지정한다.
- **xsl:apply-templates**
  - 문맥 노드의 하위에 있는 자식 요소의 처리를 지시한다.
  - select 속성을 이용하여 특정 노드를 선택할 수 있다.
- **xsl:value-of**
  - 요소나 속성의 내용을 문자열 형식으로 출력한다.
  - select 속성을 이용하여 특정 노드를 선택할 수 있다.

## ❖ 제어 요소

### ■ **xsl:if**

- test 속성의 조건이 참일 경우의 처리를 지시한다.

### ■ **xsl:choose, xsl:when, xsl:otherwise**

- test 속성의 조건이 참일 경우와 거짓일 경우의 처리를 지시한다.

### ■ **xsl:for-each**

- select 속성을 통해 선택한 노드를 반복적으로 처리한다.

### ■ **xsl:sort**

- xml 문서의 내용을 순차적으로 정렬한다.
- <apply-templates>나 <for-each>의 자식 요소로 사용할 수 있다.
- order 속성으로 차순(ascending|descending)을 지정
- data-type 속성으로 숫자/문자 여부(number|text)를 지정

#### ❖ XML 문서 생성 요소

##### ▪ **xsl:element**

- 결과 트리를 출력할 때 새로운 요소를 생성한다.

##### ▪ **xsl:attribute**

- 결과 트리에 출력되는 요소의 속성값을 동적으로 할당한다.



1. XSL이란?

2. XSL Elements

3. XSL에 의한 문서 표현

4. XPath란?

5. 필터

6. Xpath 함수

## 예제

### XML: 문서의 내용

```
<hanmun_text punctuation_scheme="
standard">
<heading>
任實縣量後陳傳令<annotation>甲戌十月
</annotation>
</heading>
<p>
官自在朝時，素聞外邑之量後陳白徵之寃。故
到任後採取民瘼，則本縣亦有己巳改量時，乍
起執實，旋又陳棄者至於八九十結，而方在白
徵之中。聞來不勝驚慘<이두>是乎矣</이두>
。但本縣亦入於年分稍實中，番庫全灾處亦不
得一一蒙頃，則况此量後陳之初不入於舉論中
者哉。夙夜憂歎，寢食不安，百爾思量，計無
所施。而身為牧民之官，亦不忍坐視吾民之白
地納稅。故反復揣摩，艱辛變通，或捐廩而自
備，或從他而鳩合，雖未得逐庫盡頃，從其陳
田結數之多少，或折半或分數減給計料。而各
面書負等看審裳紙中懸陳者，有難準信<이
두>同</이두>
。陳地佃夫處捧單後，親自逐庫摘奸。次如是
消詳，傳令風約及佃夫等，須悉勸懇之意，從
實書單以納<이두>是乎矣</이두>
。若不體官家之至意，或有以實爲陳者，則如
此之人，乃是全無人心者，當隨其現露，面任
及佃夫等依田政律各別嚴刑懲殲，斷不饒貸事。
</p>
</hanmun_text>
```

#### ○ 任實縣量後陳傳令 【甲戌十月】

官自在朝時，素聞外邑之量後陳白徵之寃。故到任後採取民瘼，則本縣亦有己巳改量時，乍起執實，旋又陳棄者至於八九十結，而方在白徵之中。聞來不勝驚慘【是乎矣】。但本縣亦入於年分稍實中，番庫全灾處亦不得一一蒙頃，則况此量後陳之初不入於舉論中者哉。夙夜憂歎，寢食不安，百爾思量，計無所施。而身為牧民之官，亦不忍坐視吾民之白地納稅。

### XSL: 문서의 모양

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="html" encoding="UTF-8"
indent="yes"/>

  <xsl:template match="/">
    <div class="text-container" style="line-height: 1.8;
font-size: 1.1em;">
      <xsl:apply-templates select="//hanmun_text |
//translation"/>
    </div>
  </xsl:template>

  <xsl:template match="hanmun_text | translation">
    <div class="content-unit">
      <xsl:apply-templates/>
    </div>
  </xsl:template>

  <xsl:template match="title">
    <span style="color: navy; font-size: 1.5em;
vertical-align: baseline; margin: 0 2px;">
      § <xsl:apply-templates/><br/>
    </span>
  </xsl:template>

  <xsl:template match="heading">
    <span style="color: darkcyan; font-size: 1.25em;
vertical-align: baseline; margin: 0 2px;">
      <br/>○ <xsl:apply-templates/><br/>
    </span>
  </xsl:template>

  .....
  .....

</xsl:stylesheet>
```

## ○ 任實縣量後陳傳令 【甲戌十月】

官自在朝時，素聞外邑之量後陳白徵之冤。故到任後採取民瘼，則本縣亦有己巳改量時，乍起執實，旋又陳棄者至於八九十結，而方在白徵之中。聞來不勝驚慘〔是乎矣〕。但本縣亦入於年分稍實中，畚庫全災處亦不得一一蒙頌，則况此量後陳之初不入於舉論中者哉。夙夜憂歎，寢食不安，百爾思量，計無所施。而身為牧民之官，亦不忍坐視吾民之白地納稅。故反復揣摩，艱辛變通，或捐廩而自備，或從他而鳩合，雖未得逐庫盡頌，從其陳田結數之多少，或折半或分數減給計料。而各面書負等看審裳紙中懸陳者，有難準信〔同〕。陳地佃夫處捧單後，親自逐庫摘奸。次如是消詳，傳令風約及佃夫等，須悉勲懇之意，從實書單以納〔是乎矣〕。若不體官家之至意，或有以實為陳者，則如此之人，乃是全無人心者，當隨其現露，面任及佃夫等依田政律各別嚴刑懲礪，斷不饒貸事。



1. XSL이란?

2. XSL Elements

3. XSL에 의한 문서 표현

4. XPath란?

5. 필터

6. Xpath 함수

## ❖ XPath란?

- XML 문서 내의 특정 부분(element, attribute)을 찾아가기 위해 사용되는 경로 표기 언어
- 1999년 11월 W3C 권고안으로 XPath 1.0 발표.

## ❖ XPath의 용도

- 스타일 쉬트
  - 스타일을 적용할 요소를 찾아가는 경로 표시자로 사용
  - xsl:template 요소의 match 속성 혹은 xsl:apply-templates, xsl:value-of 요소의 select 속성에 사용.
- 데이터베이스
  - 데이터 상에서 검색 대상 요소를 찾아내는 경로 표시자로 사용
  - XML Data Type Method에서 사용

## ❖ 노드(node)

- XML 문서를, 루트(root)라는 하나의 개체에서 시작하여 여러 개의 연결된 개체로 벌어져 나간 나무(tree) 형태의 자료로 이해하고, 서로 연결된 각각의 구성 요소를 노드(node)라고 명명
- XML 문서 상에서 명시적으로 확인되는 노드는 요소(element)와 속성(attribute)이지만, XPath에서는 정교하고 유연한 조작을 위해 다음을 모두 노드로 간주
  - 루트(root)
  - 요소(element)
  - 속성(attribute)
  - 텍스트(text)
  - 이름공간(namespace)
  - 프로세싱 명령문(processing instruction)
  - 주석(comment)
- 노드 집합(node collection) : XML 문서상의 모든 노드 중에서 특정 조건에 부합하는 노드들의 집합

# 루트 노드와 문맥 노드

## ❖ 루트 노드(root node)

- XML 문서 자체를 지시하는 가상적인 최상위 노드
- 문서 루트(document root)라고도 하며, ("/")로 표기

## ❖ 문맥 노드(context node)

- 템플릿(template)의 적용의 시작점이 되는 노드

## ❖ 위치 경로: 문맥 노드의 위치를 지정하는 방법

- /
  - 루트 노드를 선택
- /tag
  - 루트 노드의 자식인 모든 <tag> 요소를 선택
  - 문맥 노드가 루트 노드에서 <tag> 요소로 이동
- tag
  - 문맥 노드의 자식인 모든 <tag> 요소를 선택
  - 문맥 노드가 상위 노드에서 <tag> 요소로 이동
- /tag/stag/sstag
  - 루트 노드의 자식 <tag> 요소를 찾은 후
  - <tag> 요소의 자식 <stag> 요소를 찾은 후
  - <stag> 요소의 자식 <sstag> 요소를 선택
  - 문맥 노드는 최종적으로 선택된 <sstag> 요소로 이동
- @attr
  - 문맥 노드의 attr 속성을 선택
- .
  - 문맥 노드 자체를 선택
- ..
  - 문맥 노드의 상위 노드를 선택

### ❖ 재귀적 내림 연산자(Recursive Descent Operator) "//"

- 찾고자 하는 노드의 위치와 상관없이 노드 이름을 찾아 경로 이동
- **//인명** → 문서 내의 모든 <인명> 요소를 선택.
- **//@id** → 문서 내의 모든 id 속성을 선택

### ❖ OR 연산자 "|"

- 어떤 노드가 하나 이상의 패턴과 일치할 경우 템플릿을 활성화
- **인명 | 지명** → 노드 이름이 <인명> 또는 <지명>인 경우의 처리 지시



1. XSL이란?

2. XSL Elements

3. XSL에 의한 문서 표현

4. XPath란?

5. 필터

6. Xpath 함수

## ❖ 노드 선택 필터 “[ ]”

- 특정 조건을 만족하는 노드의 선택
- 필터는 문맥 노드의 위치에 영향을 주지 않는다.
- 분야[소분야]
  - 문맥 노드의 자식 중 <소분야> 자식 요소를 가진 모든 <분야> 요소 선택
  - 문맥 노드가 <소분야> 요소가 아니라 <분야> 요소까지만 이동
- 인명[@id]
  - 문맥 노드의 자식 중 id 속성을 가진 모든 <인명> 요소만 선택
- 분야[소분야 = 'A']
  - 문맥 노드의 자식 중 <소분야> 요소의 값이 “A”인 자식 노드를 가진 모든 <분야> 요소 선택
- 유형[. = 'B']
  - 문맥 노드의 자식 중 값이 ‘B’인 모든 <유형> 요소
- //명단[인명/@id = 'C']
  - id 속성 값이 ‘C’ 인 <인물> 요소를 자식으로 가진 모든 <명단> 요소

※응용

- /프로젝트[@code = 'D']/명단/인명[@id = 'C']

## ❖ 필터 조건문에 사용하는 연산자

## ❖ 비교 연산자

연산자	기능
=	같다
!=	같지 않다
&lt;	작다
&lt;=	작거나 같다
&gt;	크다
&gt;=	크거나 같다

## ❖ 관계 연산자

연산자	기능
and	두 조건을 모두 만족
or	적어도 하나를 만족



1. XSL이란?

2. XSL Elements

3. XSL에 의한 문서 표현

4. XPath란?

5. 필터

6. Xpath 함수

## ❖ 노드 함수

- **node()**
  - 노드 그 자체를 반환
- **text()**
  - 노드의 PCDATA 내용을 반환

## ❖ 위치 함수

- **count()**
  - 노드 집합에서 노드의 갯수를 반환
- **position()**
  - 노드 집합 내의 노드의 위치를 문서 순서대로 반환
  - "tag[position() = 2]"는 "tag[2]"와 동일
- **last()**
  - 노드 집합에서 마지막 노드의 위치를 반환

## ❖ 수치 함수

▪ **number()**

- PCDATA를 수치값으로 변환하여 반환

▪ **sum()**

- 노드 집합 내의 모든 수치값을 합하여 반환

## ❖ 부울 함수

▪ **boolean()**

- XPath expression을 진. 위를 판단하여 true 혹은 false를 반환
- 수치값: 0인 경우-> false, 0이 아닌 경우 -> true
- 문자열: 1개 이상의 문자로 구성된 경우 -> true
- 노드 집합: empty인 경우 -> false

## ❖ 문자열 함수

- **string()**
  - 임의의 값을 문자열로 변환하여 반환
- **string-length()**
  - 공백을 포함한 문자열의 길이를 반환
- **concat()**
  - 2개 이상의 문자열을 합하여 하나의 문자열로 반환
- **contains()**
  - 두 번째 문자열의 내용이 첫 번째 문자열에 포함되었는지 진위를 반환
- **substring(), substring-after(), substring-before()**
  - 문자열의 일부를 반환
- **translate()**
  - 두 번째 문자열에 속하는 문자를 세 번째 문자열에 속하는 문자로 변환