

# XSL의 이해

김 현

한국학중앙연구원 인문정보학교실

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



이 저작물(PPT)의 인용 표시 방법:  
김현, 「XSL의 이해」, '전자문서와 하이퍼텍스트' 수업 자료 (2018)



1. XSL이란?

2. XSL Elements

3. XSL에 의한 문서 표현

### ❖ XSL (eXtensible Stylesheet Language)

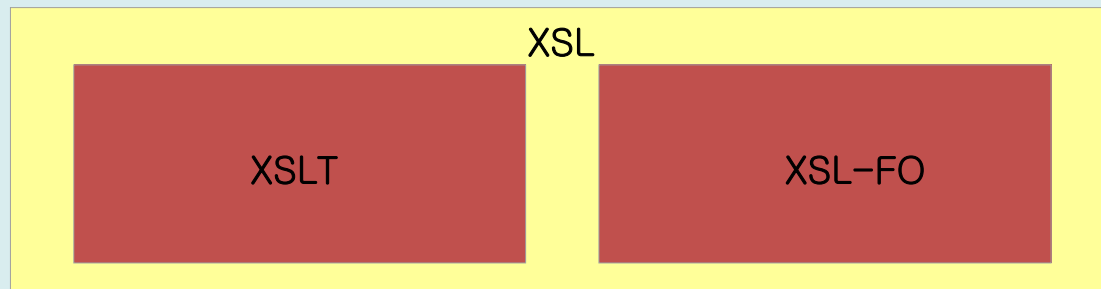
- 문서의 스타일을 정의하기 위한 언어

### ❖ 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"  
  xmlns:html="http://www.w3.org/1999/xhtml">  
  .....  
</xsl:stylesheet>
```



1. XSL이란?

2. XSL Elements

3. XSL에 의한 문서 표현

## ❖ 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

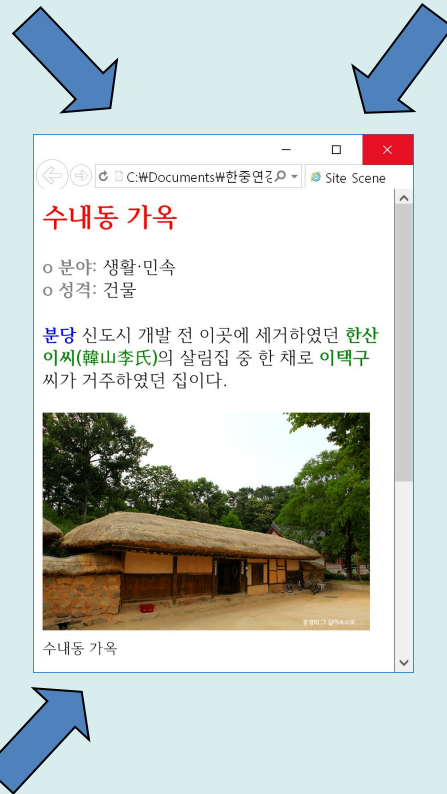
## 예제

### DTD: 문서의 구조

```
<?xml version="1.0" encoding="utf-8"?>
<!ENTITY % 고유명사 "인명지명">
<!ELEMENT 항목 (항목명, 메타데이터, 본문)>
  <!ATTLIST 항목 ID CDATA #REQUIRED>
<!ELEMENT 항목명 (#PCDATA)>
<!ELEMENT 메타데이터 (분야, 유형)>
<!ELEMENT 분야 (#PCDATA)>
<!ELEMENT 성격 (#PCDATA)>
<!ELEMENT 본문 (#PCDATA|%고유명사;)*>
<!ELEMENT 인명 (#PCDATA)>
  <!ATTLIST 인명 유형 CDATA #IMPLIED>
<!ELEMENT 지명 (#PCDATA)>
  <!ATTLIST 지명 유형 CDATA #IMPLIED>
```

### XML: 문서의 내용

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet type="text/xsl" href="mystyle.xml"?>
<항목 ID="10001332"
  IRI="http://digerati.aks.ac.kr/terms/heritage#수내동_가옥">
  <항목명>수내동 가옥</항목명>
  <메타데이터>
    <분야>생활·민속</분야>
    <성격>건물</성격>
  </메타데이터>
  <본문>
  <지명>분당</지명>신도시 개발 전 이곳에 세거하였던 <인명
  유형="성씨">한산이씨(韓山李氏)</인명>의 살림집 중
  한 채로 <인명>이택구</인명>씨가 거주하였던 집이다.
  </본문>
</항목>
```



### XSL: 문서의 모양

```
<?xml version="1.0" encoding="utf-8"?>
<!-- XSL for 항목_Display 1.0 written by Hyeon Kim 2016. 6. 9. -->
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:html="http://www.w3.org/1999/xhtml"
  xmlns:dc="http://purl.org/dc/terms/"
  version="XHTML+RDFa 1.0" >
  <xsl:output method="xml" indent="yes"/>
  <xsl:template match="/">
    <html:html
      xmlns:dc="http://purl.org/dc/terms/"
      version="XHTML+RDFa 1.0" >
      <xsl:apply-templates />
    </html:html>
  </xsl:template>
  .....
  <xsl:template match="인명">
    <html:font style="color:green;font-weight:bold">
      <xsl:value-of select="."/>
    </html:font>
  </xsl:template>
</xsl:stylesheet>
```



← → ↻ C:\Documents\한중연결 Site Scene

## 수내동 가옥

- 분야: 생활·민속
- 성격: 건물

분당 신도시 개발 전 이곳에 세거하였던 **한산 이씨(韓山李氏)**의 살림집 중 한 채로 **이택구** 씨가 거주하였던 집이다.



수내동 가옥

```

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

<항목 ID="10001332" IRI="http://digerati.aks.ac.kr/terms/heritage#수내동_가옥">
  <항목명>
    수내동 가옥
  </항목명>
  <메타데이터>
    <분야>생활·민속</분야>
    <성격>건물</성격>
  </메타데이터>
  <본문>
    <지명>분당</지명> 신도시 개발 전 이곳에 세거하였던
      <인명 유형="성씨">한산이씨(韓山李氏)</인명>의 살림집 중 한 채로
      <인명>이택구</인명>씨가 거주하였던 집이다.
    <삽도 url="house_a.jpg">수내동 가옥</삽도>
    <삽도 url="house_b.jpg">수내동 가옥</삽도>
  </본문>
</항목>

```

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<!ENTITY % 고유명사 "인명|지명">
```

```
<!ELEMENT 항목 (항목명, 메타데이터, 본문)>
```

```
  <!ATTLIST 항목 ID CDATA #REQUIRED>
```

```
<!ELEMENT 항목명 (#PCDATA)>
```

```
<!ELEMENT 메타데이터 (분야, 성격)>
```

```
<!ELEMENT 분야 (#PCDATA)>
```

```
<!ELEMENT 성격 (#PCDATA)>
```

```
<!ELEMENT 본문 (#PCDATA|%고유명사;)*>
```

```
<!ELEMENT 인명 (#PCDATA)>
```

```
  <!ATTLIST 인명 유형 CDATA #IMPLIED>
```

```
<!ELEMENT 지명 (#PCDATA)>
```

```
  <!ATTLIST 지명 유형 CDATA #IMPLIED>
```

#### 예제 분석: XSD 문서(1)

```
<?xml version="1.0" encoding="UTF-8"?>  
<xs:schema xmlns:xs=http://www.w3.org/2001/XMLSchema elementFormDefault="qualified">
```

```
<xs:element name="항목">  
  <xs:complexType>  
    <xs:sequence>  
      <xs:element ref="항목명"/>  
      <xs:element ref="메타데이터"/>  
      <xs:element ref="본문"/>  
    </xs:sequence>  
    <xs:attributeGroup ref="attlist.항목"/>  
  </xs:complexType>  
</xs:element>  
<xs:attributeGroup name="attlist.항목">  
  <xs:attribute name="ID" use="required"/>  
</xs:attributeGroup>  
<xs:element name="항목명" type="xs:string"/>
```

```
<xs:element name="메타데이터">  
  <xs:complexType>  
    <xs:sequence>  
      <xs:element ref="분야"/>  
      <xs:element ref="성격"/>  
    </xs:sequence>  
  </xs:complexType>  
</xs:element>  
<xs:element name="분야" type="xs:string"/>  
<xs:element name="성격" type="xs:string"/>
```

#### 예제 분석: XSD 문서(2)

```
<xs:element name="본문">
  <xs:complexType mixed="true">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element ref="인명"/>
      <xs:element ref="지명"/>
    </xs:choice>
  </xs:complexType>
</xs:element>
```

```
<xs:element name="인명">
  <xs:complexType mixed="true">
    <xs:attributeGroup ref="attlist.인명"/>
  </xs:complexType>
</xs:element>
<xs:attributeGroup name="attlist.인명">
  <xs:attribute name="유형"/>
</xs:attributeGroup>
```

```
<xs:element name="지명">
  <xs:complexType mixed="true">
    <xs:attributeGroup ref="attlist.지명"/>
  </xs:complexType>
</xs:element>
<xs:attributeGroup name="attlist.지명">
  <xs:attribute name="유형"/>
</xs:attributeGroup>
```

```
</xs:schema>
```

```
<?xml version="1.0" encoding="utf-8"?>

<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:html="http://www.w3.org/1999/xhtml" >

  <xsl:output method="xml" indent="yes"/>
  .....
  <xsl:template match="/" >
    .....
    <xsl:apply-templates />
    .....
  </xsl:template>
  .....
  <xsl:template match="....." >
    .....
    <xsl:value-of select="." />
    .....
  </xsl:template>
  .....
</xsl:stylesheet>
```



#### 예제 분석: XSL 문서(2)

```
<xsl:template match="/">
  <html:html xmlns="http://www.w3.org/1999/xhtml"
    xmlns:dc="http://purl.org/dc/terms/" version="XHTML+RDFa 1.0" >
    <xsl:apply-templates />
  </html:html>
</xsl:template>

<xsl:template match="항목">
  <html:html>
    <html:head>
      <html:title>
        <xsl:value-of select="./@ID"/>
      </html:title>
      <html:link>
        <xsl:attribute name="rel">dc:relation</xsl:attribute>
        <xsl:attribute name="href"><xsl:value-of select="./@IRI"/></xsl:attribute>
      </html:link>
      <html:style>
        body {font-family:함초롬바탕}
        img {border:0}
        a {text-decoration:none}
      </html:style>
    </html:head>
    <html:body>
      <xsl:apply-templates/>
    </html:body>
  </html:html>
</xsl:template>
```

#### 예제 분석: XSL 문서(3)

```
<xsl:template match="항목명">
  <html:font style="color:red;font-weight:bold;font-size:18pt">
    <xsl:value-of select="."/>
  </html:font>
  <html:br/> <html:br/>
</xsl:template>

<xsl:template match="메타데이터">
  <xsl:apply-templates/>
  <html:br/>
</xsl:template>

<xsl:template match="분야">
  <html:font style="color:grey;font-weight:bold">o 분야: </html:font>
  <xsl:value-of select="."/>
  <html:br/>
</xsl:template>

<xsl:template match="성격">
  <html:font style="color:grey;font-weight:bold">o 성격: </html:font>
  <xsl:value-of select="."/>
  <html:br/>
</xsl:template>
```

#### 예제 분석: XSL 문서(4)

```
<xsl:template match="본문">
  <xsl:apply-templates/>
  <html:br/>
</xsl:template>
```

```
<xsl:template match="지명">
  <html:font style="color:blue;font-weight:bold">
    <xsl:value-of select="."/>
  </html:font>
</xsl:template>
```

```
<xsl:template match="인명">
  <html:font style="color:green;font-weight:bold">
    <xsl:value-of select="."/>
  </html:font>
</xsl:template>
```

```
<xsl:template match="text()">
  <xsl:value-of select="."/>
</xsl:template>
```

```
<xsl:template match="삽도">
  <html:br/>
  <html:br/>
  <html:img>
    <xsl:attribute name="width">300</xsl:attribute>
    <xsl:attribute name="src"><xsl:value-of select="./@url" /></xsl:attribute>
  </html:img>
  <!-- 간단히 이렇게 쓰는 것도 가능
  <html:img width="300" src="{./@url}" />
  -->
  <html:br/>
  <html:font size="-1"><xsl:value-of select="."/></html:font>
  <html:br/>
</xsl:template>
```