



<XML>

<TAG>

<DATA>

<TEXT>

</XML>

# XML의 이해

전통문화연구회 시고전번역학 입문 강의 자료

2026.04.16

서소리

# XML(eXtensible Markup Language)

## 정의 및 역할

- 문서의 구조적인 형식과 내용 요소들이 컴퓨터가 식별할 수 있는 명시적 정보로 기술될 수 있도록 하기 위한 마크업 언어
- 텍스트 데이터의 각 요소에 의미(tag)를 부여함으로써 시스템 간의 원활한 데이터 교환과 처리를 가능하게 함

## 표준화 및 발전 과정

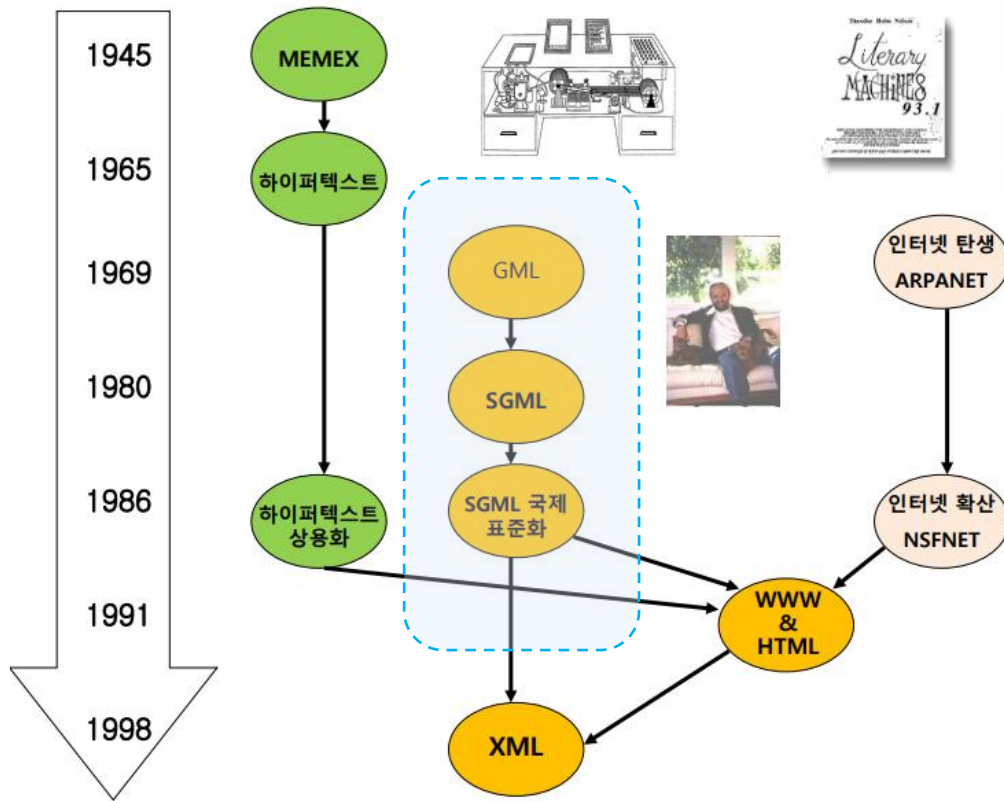
- 월드 와이드 웹(WWW)의 표준화를 주도하고 있는 W3C(World Wide Web Consortium)에서 관리
- 1998년 최초 권장안(Recommendation) 제시 이후, 현재까지 다양한 개선안과 응용 기술의 표준화 방안을 제공

## CCTI와의 관계

- CCTI(Classical Chinese Text Interpreter)의 근간을 이루는 기본 데이터 형식
- 원문, 표점, 번역, 주석 등 고전의 데이터를 체계적으로 구조화하는 도구

## XML 문서의 핵심 특성

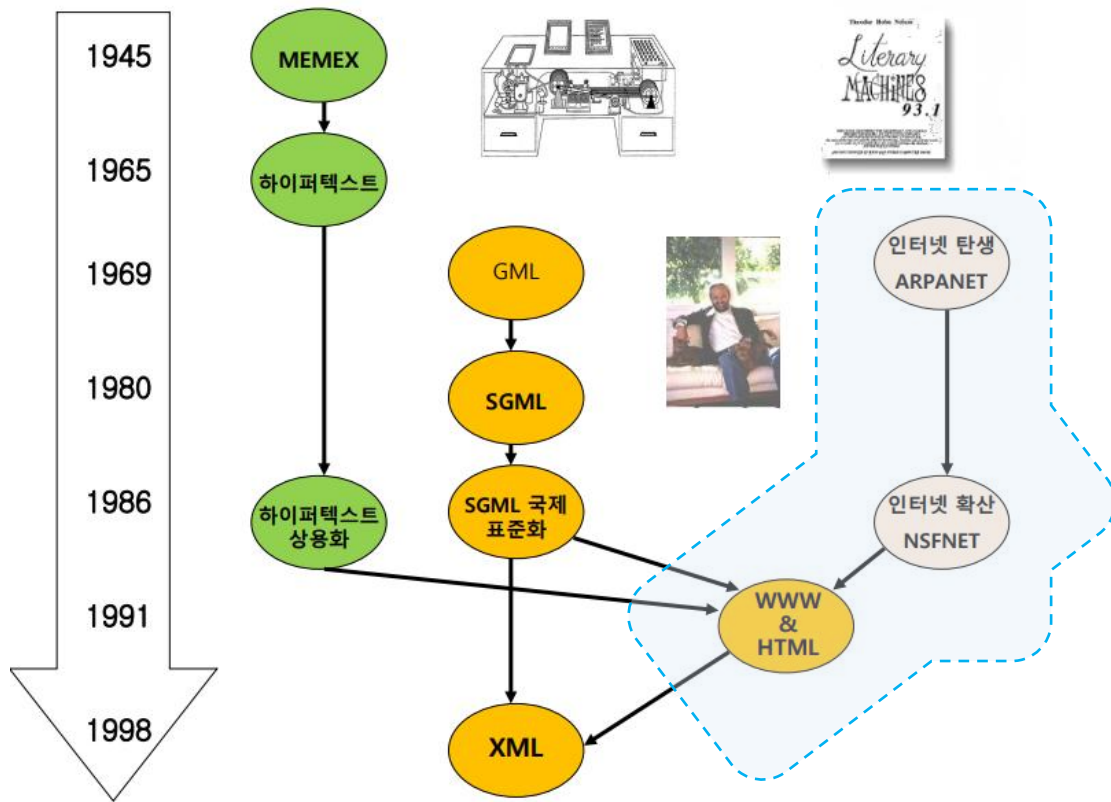
- 문서를 '문서의 구조', '문서의 내용', '문서의 모양' 세 가지의 영역으로 분리하여 관리



## SGML(Standard Generalized Markup Language)

☞ 지식 구조화의 시초: 표준 범용 마크업 언어

- 문서에 담긴 정보 요소와 그 요소들의 관계를 컴퓨터가 논리적으로 이해할 수 있도록 설계된 마크업 언어
- 1969년 IBM의 찰스 골드파브(Charles Goldfarb)와 동료들이 개발한 GML(Generalized Markup Language)에 뿌리를 둬
- 1986년 국제표준기구(ISO)에 의해 구조적 데이터 표현의 표준안(ISO 8879)으로 채택됨
- 주요 특징:
  - 구조 중심적 기술: 문서의 논리적 구조를 텍스트로 표현.
  - 장치 독립성: 하드웨어/소프트웨어에 상관없이 문서 교환 가능.
  - 엄격한 형식 검증: 문서 구조 정의(DTD)를 통해 일관된 문서 형식을 요구.
- 한계: 지나치게 방대하고 복잡한 문법 규격으로 인해 일반 사용자가 웹 환경에서 다루기에는 진입장벽이 높았음

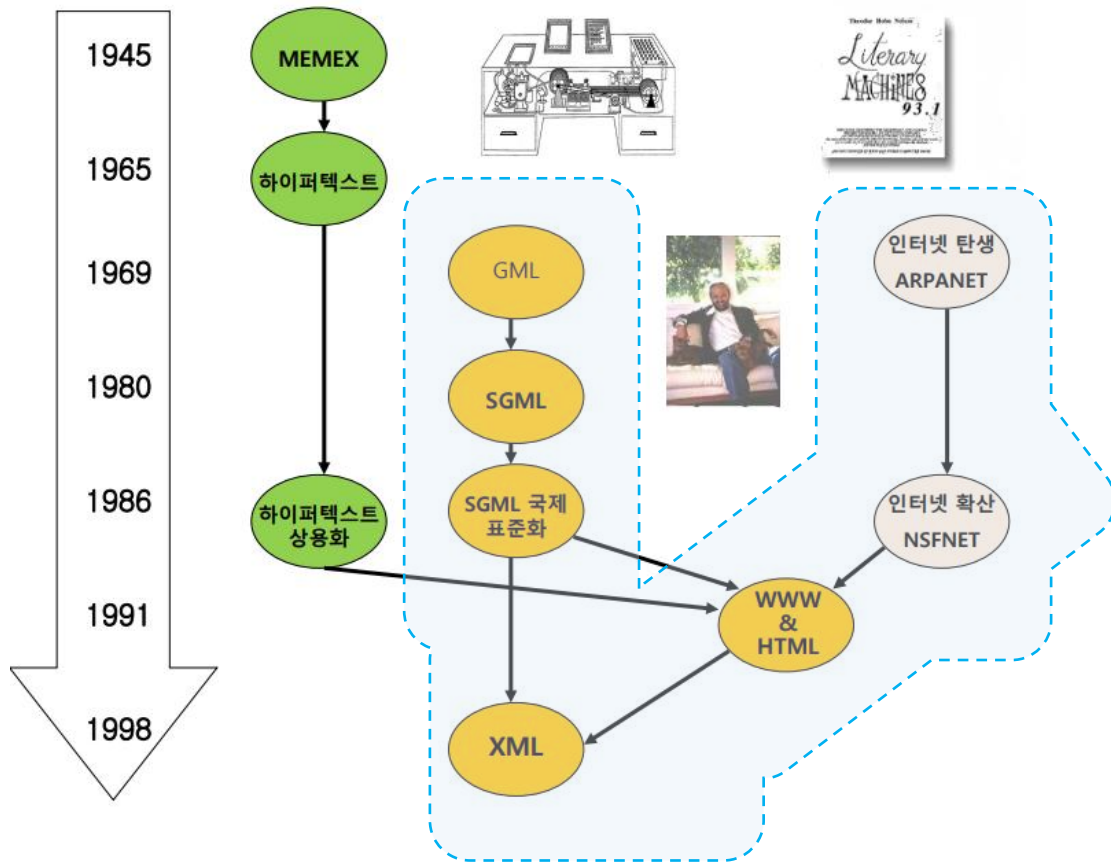


## HTML(Hypertext Markup Language)

☞ 웹의 대중화와 하이퍼텍스트의 탄생

- 웹 브라우저에서 통용되는 하이퍼텍스트(Hypertext) 형식의 전자문서를 제작하는 언어
- SGML과의 관계: SGML의 문법을 응용하여 탄생했으나, SGML이 추구했던 '내용-구조-모양'의 엄격한 독립성보다는 웹에서의 즉각적인 출력에 집중함
- SGML의 복잡한 제약에서 벗어나 누구나 쉽게 문서를 만들고 연결(Link)할 수 있게 함으로써 월드 와이드 웹(WWW)의 폭발적 성장을 견인
- HTML 마크업 종류:

구조 마크업	제목(<h1>), 단락(<p>), 목록(<ul>) 등 문서의 뼈대 형성
표현 마크업	굵게(<b>), 기울임(<i>), 글자 색상 등 시각적 효과 부여
하이퍼링크 마크업	문서와 문서를 연결하는 핵심 태그(<a>)

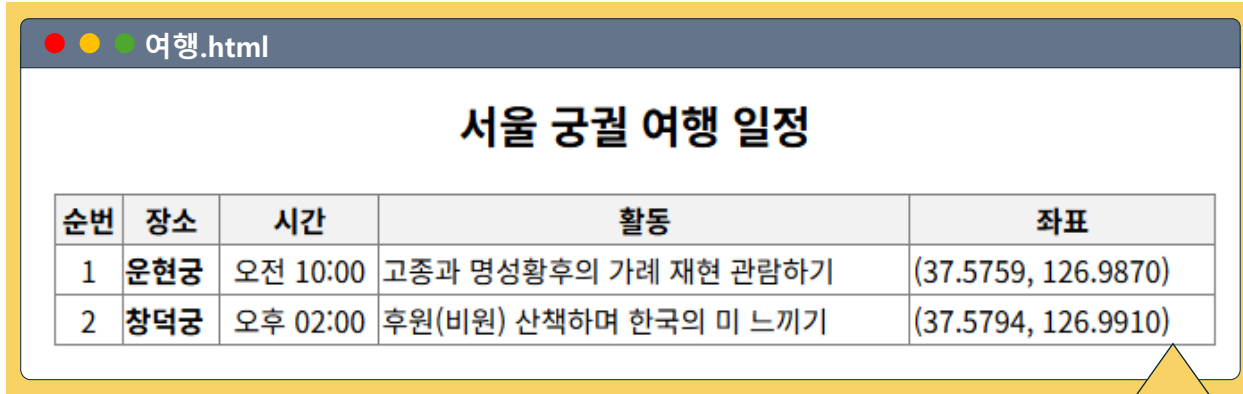


## XML(eXtensible Markup Language)

☞ SGML의 엄격함과 HTML의 유연한 결합

- 탄생 배경: HTML이 간과했던 '문서의 의미 구조'를 회복하면서도, SGML의 복잡함을 걷어내어 인터넷 환경에 최적화된 언어
  - 핵심 가치: HTML이 표현하지 못하는 문서 내부 정보 요소의 성격을 명확하게 표현 가능
  - XML의 3요소 분리:
    - 문서의 구조: 데이터 간의 위계와 규칙 정의 (DTD / XSD)
    - 문서의 내용: 실제 기록된 정보와 지식 데이터 (XML)
    - 문서의 모양: 데이터가 사용자에게 보여지는 시각적 형식 (XSL)
  - 분리의 장점: 데이터의 유연성 확보
- 데이터 내용(원본)을 수정하지 않고도 모양(Style)만 교체하거나, 모양을 유지한 채 내용(Data)만 확장·수정할 수 있는 유연성 보장

## ❖ HTML에서 구조·내용·모양의 통합과 한계



순번	장소	시간	활동	좌표
1	운현궁	오전 10:00	고종과 명성황후의 가례 재현 관람하기	(37.5759, 126.9870)
2	창덕궁	오후 02:00	후원(비원) 산책하며 한국의 미 느끼기	(37.5794, 126.9910)

### 시각적 구현 중심의 문서 구성

- 정보를 화면에 어떻게 보여줄지를 중심으로 문서 구성
- <table>, <tr>, <td>, <b> 등은 정보의 성격이 아닌 시각적 형태(표, 행, 열, 굵게)만을 정의함

### 구조와 모양의 혼재

- 문서의 뼈대(구조)와 꾸밈(모양)이 HTML 태그 하나에 뒤섞여 있어, 디자인을 바꾸려면 문서 전체의 구조를 뜯어고쳐야 하는 비유연성 발생
- 데이터가 늘어나거나 형식이 바뀔 때마다 시각적 코드(<td> 등)를 일일이 수정해야 함

### 데이터 의미의 불명확성

- 코드상에서는 모두 동일한 <td>(표의 칸)로 취급되어 컴퓨터가 각 데이터의 실질적 의미를 식별 불가

여행.html



```

<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <title>서울 궁궐 여행 일정 - 표 형식</title>
</head>
<body>
  <h2 style="color: #2c3e50; font-family: sans-serif; text-align:center;">서울 궁궐 여행
  일정</h2>
  <table border="1" style="border-collapse: collapse; width: 100%;">
    <tr style="background-color: #f2f2f2;">
      <th>순번</th>
      <th>장소</th>
      <th>시간</th>
      <th>활동</th>
      <th>좌표</th>
    </tr>
    <tr>
      <td>1</td>
      <td><b>운현궁</b></td>
      <td>오전 10:00</td>
      <td>고종과 명성황후의 가례 재현 관람하기</td>
      <td>(37.5759, 126.9870)</td>
    </tr>
    <tr>
      <td>2</td>
      <td><b>창덕궁</b></td>
      <td>오후 02:00</td>
      <td>후원(비원) 산책하며 한국의 미 느끼기</td>
      <td>(37.5794, 126.9910)</td>
    </tr>
  </table>
</body>
</html>

```

### 여행.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="여행.xsl"?>
<여행정보 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="여행.xsd">
  <제목>서울 궁궐 여행 일정</제목>
  <일정 순번="1">
    <장소>운현궁</장소>
    <좌표 경도="126.9870" 위도="37.5759"/>
    <시간>오전 10:00</시간>
    <활동>고종과 명성황후의 가례 재현 관람하기</활동>
  </일정>
  <일정 순번="2">
    <장소>창덕궁</장소>
    <좌표 경도="126.9910" 위도="37.5794"/>
    <시간>오후 02:00</시간>
    <활동>후원(비원) 산책하며 한국의 미 느끼기</활동>
  </일정>
</여행정보>
```

### 여행.XSL

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/여행정보">
  <xsl:for-each select="일정">
    <tr>
      <td align="center"><xsl:value-of select="@순번"/></td>
      <td><b><xsl:value-of select="장소"/></b></td>
      <td align="center"><xsl:value-of select="시간"/></td>
      <td><xsl:value-of select="활동"/></td>
      <td>(<xsl:value-of select="좌표/@위도"/>, <xsl:value-of select="좌표/@경도"
"/></td>
    </tr>
  </xsl:for-each>
</xsl:template>
</xsl:stylesheet>
```

## ❖ XML에서 구조·내용·모양의 분리와 독립성



### XML: 문서의 내용 정의

시각적 서식 없이 내용과 의미 태그만으로 구성된 순수 데이터 파일

순번	장소	시간	활동	좌표
1	운현궁	오전 10:00	고종과 명성황후의 가례 재현 관람하기	(37.5759, 126.9870)
2	창덕궁	오후 02:00	후원(비원) 산책하며 한국의 미 느끼기	(37.5794, 126.9910)



### XSL: 문서의 모양 정의

데이터를 어떻게 보여줄지 정의한 모양 파일



### XSD: 문서의 구조 정의

데이터가 지켜야 할 구조와 제약 조건을 선언한 설계도

### 여행.XSD

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="여행정보">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="제목" type="xs:string"/>
      <xs:element name="일정" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="장소" type="xs:string"/>
            <xs:element name="좌표">
              <xs:complexType>
                <xs:attribute name="경도" type="xs:decimal" use="required"/>
                <xs:attribute name="위도" type="xs:decimal" use="required"/>
              </xs:complexType>
            </xs:element>
            <xs:element name="시간" type="xs:string"/>
            <xs:element name="활동" type="xs:string"/>
          </xs:sequence>
          <xs:attribute name="순번" type="xs:integer" use="required"/>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>
```

## ❖ XML: 문서의 내용(Content)과 의미 구조(Semantics)를 담은 독립적 데이터 문서

여행.xml + 여행.xsl + 여행.xsd

### 서울 궁궐 여행 일정

순번	장소	시간	활동	좌표
1	운현궁	오전 10:00	고종과 명성황후의 가례 재현 관람하기	(37.5759, 126.9870)
2	창덕궁	오후 02:00	후원(비원) 산책하며 한국의 미 느끼기	(37.5794, 126.9910)

- XML은 어떻게 보여줄지보다 '무엇을 담고 있는지'를 기술하는 데 집중
- 태그를 통해 각 정보 요소의 성격을 명시적으로 표현
- XML 문서의 첫 부분에 문서의 모양과 구조를 선언하여 외부 문서와 논리적으로 결합
- [사례] 여행.xml :
  - 이 문서가 참조할 스타일시트(여행.xsl)를 연결하고(Ⓐ), 문서가 준수할 스키마(여행.xsd)를 선언함(Ⓑ)
  - 전체 정보를 <일정>이라는 반복 가능한 단위로 구조화
  - 각 <일정>에 포함된 주요 정보(<장소>, <시간>, <활동>)를 계층적으로 구성함

여행.xml

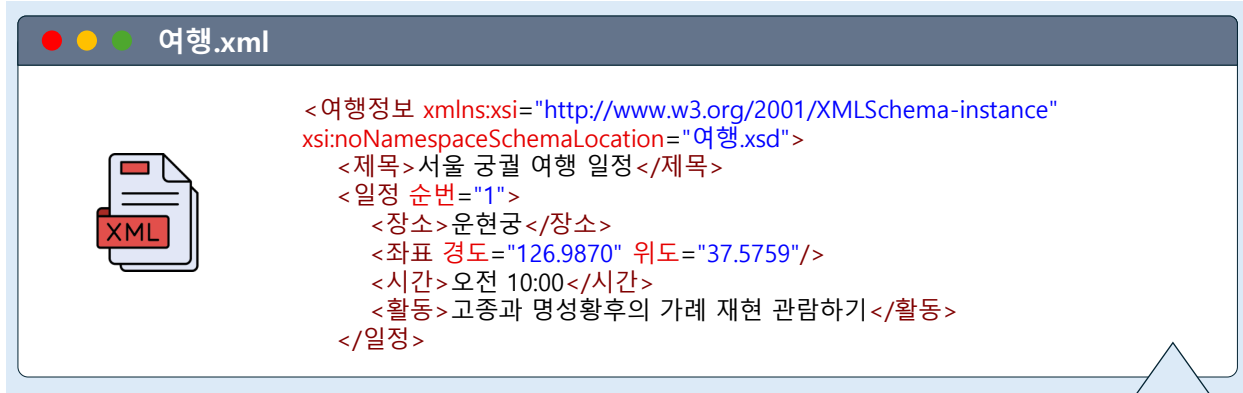


```

<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="여행.xsl"?>
<여행정보 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="여행.xsd">
  <제목>서울 궁궐 여행 일정</제목>
  <일정 순번="1">
    <장소>운현궁</장소>
    <좌표 경도="126.9870" 위도="37.5759"/>
    <시간>오전 10:00</시간>
    <활동>고종과 명성황후의 가례 재현 관람하기</활동>
  </일정>
  <일정 순번="2">
    <장소>창덕궁</장소>
    <좌표 경도="126.9910" 위도="37.5794"/>
    <시간>오후 02:00</시간>
    <활동>후원(비원) 산책하며 한국의 미 느끼기</활동>
  </일정>
</여행정보>

```

## ❖ XSD(XML Schema Definition): 문서의 구조(Structure &amp; Validation)



- XML 문서가 준수해야 할 구조적 설계도이자 데이터의 제약 조건을 정의하는 문서
- [사례] 여행.xsd :
  - 루트 요소 선언: 최상위에 <여행정보> 요소가 위치함을 명시(Ⓐ)
  - 복합 타입 정의: <여행정보>가 여러 하위 요소를 가질 수 있는 요소임을 명시(Ⓑ)
  - 순서 및 반복 제약: <일정> 요소가 나타나는 순서와 반복 횟수 지정(Ⓒ)
  - 데이터 타입 지정: <좌표>의 경위도 속성이 숫자 형식이며 반드시 존재해야 함을 규정(Ⓓ)
- 문서의 품질 검증 체계:
  - Well-formed(잘 짜인 문서): XML의 기본 문법 규칙을 준수한 문서
  - Valid(유효한 문서): Well-formed이면서, 참조하는 XSD의 구조적 규칙까지 완벽히 만족하는 문서

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" >
  <xs:element name="여행정보" >
    <xs:complexType>
      <xs:sequence>
        <xs:element name="제목" type="xs:string"/>
        <xs:element name="일정" maxOccurs="unbounded" >
          <xs:complexType>
            <xs:sequence>
              <xs:element name="장소" type="xs:string"/>
              <xs:element name="좌표" >
                <xs:complexType>
                  <xs:attribute name="경도" type="xs:decimal" use="required"/>
                  <xs:attribute name="위도" type="xs:decimal" use="required"/>
                </xs:complexType>
              </xs:element>
              <xs:element name="시간" type="xs:string"/>
              <xs:element name="활동" type="xs:string"/>
            </xs:sequence>
            <xs:attribute name="순번" type="xs:integer" use="required"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

## ❖ XSL(eXtensible Stylesheet Language): 문서의 모양(Presentation & Transformation)

여행.xml + 여행.xsl + 여행.xsd

**서울 궁궐 여행 일정**

순번	장소	시간	활동	좌표
1	운현궁	오전 10:00	고종과 명성황후의 가례 재현 관람하기	(37.5759, 126.9870)
2	창덕궁	오후 02:00	후원(비원) 산책하며 한국의 미 느끼기	(37.5794, 126.9910)

- XML 데이터를 웹 브라우저에 표시하거나 다른 형식으로 변환하는 방법을 정의하는 언어
- 핵심 원칙: 데이터 원본은 전혀 수정하지 않은 채, 보여주는 방식(presentation)만 독립적으로 결정
- [사례] 여행.xsl :
  - 템플릿 매칭: <여행정보> 요소를 찾아 전체적인 출력 틀(HTML Table) 생성(a)
  - 반복 처리: XML 내의 모든 <일정> 요소마다 동일한 서식 적용(b)
  - 데이터 추출: 특정 경로의 데이터 값을 가져와 HTML 셀(<td>) 안에 삽입(c)
- 동일한 XML 데이터를 활용하여 표, 연표, 지도 등 다양한 인터페이스로의 무한한 변신 가능

여행.xsl



```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/여행정보">
    <xsl:for-each select="일정">
      <tr>
        <td align="center"><xsl:value-of select="@순번"/></td>
        <td align="center"><b><xsl:value-of select="장소"/></b></td>
        <td align="center"><xsl:value-of select="시간"/></td>
        <td><xsl:value-of select="활동"/></td>
        <td>(<xsl:value-of select="좌표/@위도"/>, <xsl:value-of select="좌표/@경도"/>)</td>
      </tr>
    </xsl:for-each>
  </xsl:template>
</xsl:stylesheet>
  
```

“같은 데이터가 다른 형식으로 재표현될 수 있다”



여행\_타임라인.xsl

여행.xml

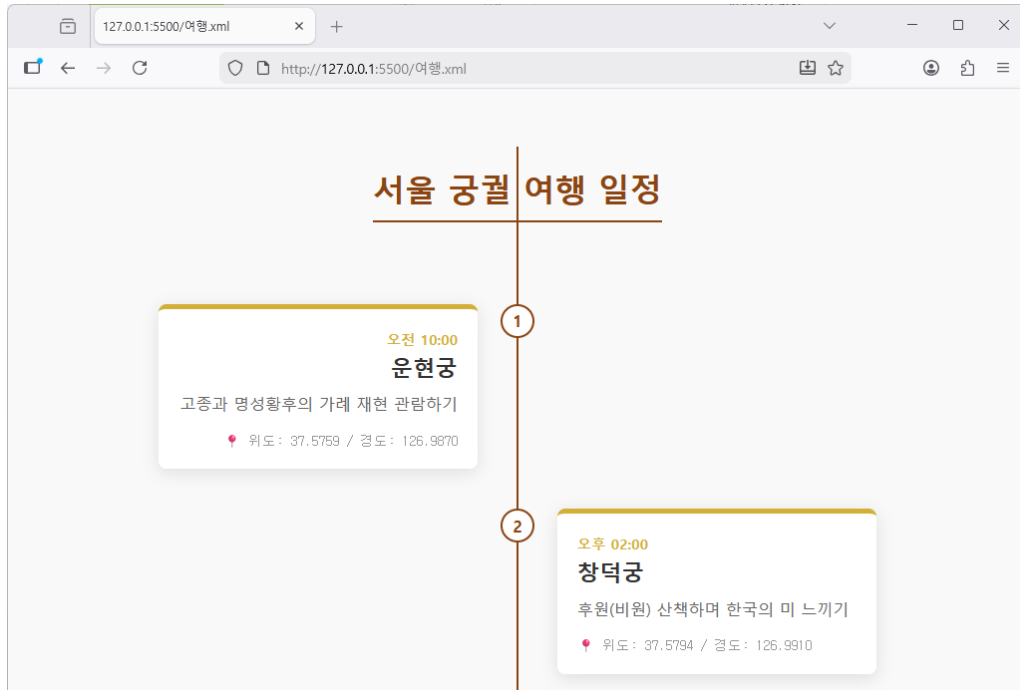
```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="여행.xsl"?>
<여행정보 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="여행.xsd">
  <제목>서울 궁궐 여행 일정</제목>
  <일정 순번="1">
    <장소>운현궁</장소>
    <좌표 경도="126.9870" 위도="37.5759"/>
    <시간>오전 10:00</시간>
    <활동>고종과 명성황후의 가례 재현 관람하기</활동>
  </일정>
  <일정 순번="2">
    <장소>창덕궁</장소>
    <좌표 경도="126.9910" 위도="37.5794"/>
    <시간>오후 02:00</시간>
    <활동>후원(비원) 산책하며 한국의 미 느끼기</활동>
  </일정>
</여행정보>
```



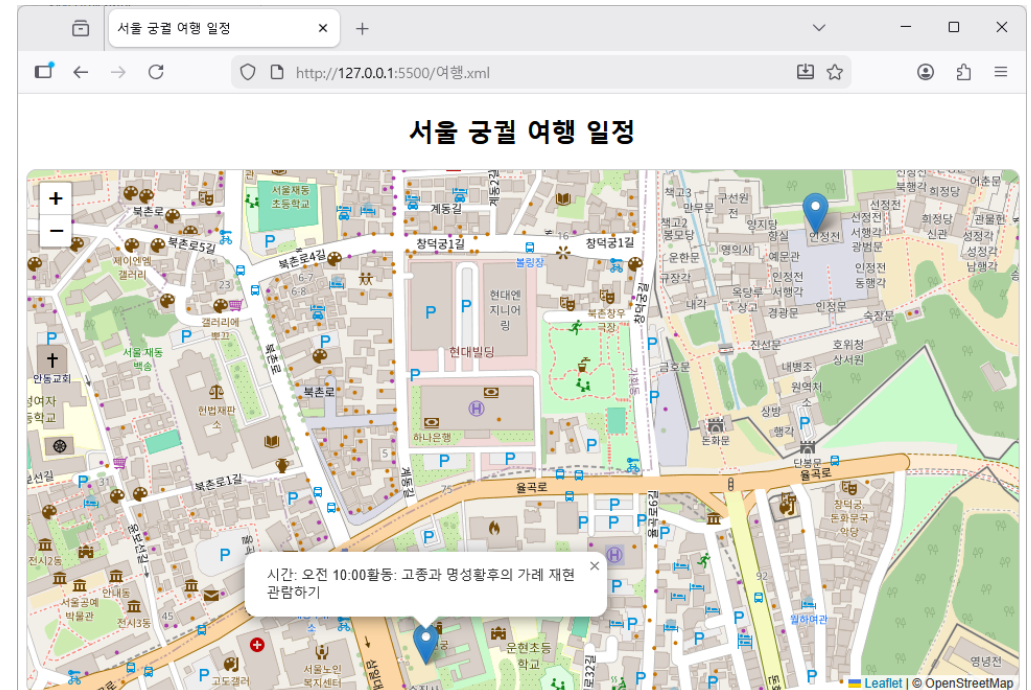
여행\_지도.xsl

“데이터(XML)는 하나, 표현(XSL)은 무한대  
원본의 훼손 없는 다중 매체 출력”

일정의 순번대로 흐름을 시각화한 형태로 표현



좌표 데이터를 활용한 지도 시각화로 위치 정보 표현



## 국사편찬위원회 『조선왕조실록』 XML 예시

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <!DOCTYPE level2 SYSTEM "history.dtd">
3  <level2 id="wca_107">
4  <front>
5  <biblioData type="L">
6  <title>
7  <mainTitle>太宗實錄 七年</mainTitle>
8  </title>
9  <date>
10 <dateOccured date="1407" type="서기"/>
11 <dateOccured type="간지">丁亥</dateOccured>
12 <dateOccured type="재위연도">태종 7</dateOccured>
13 <dateOccured type="개국연호">조선 개국(開國) 16</dateOccured>
14 <dateOccured type="중국연호">명 영락(永樂) 5</dateOccured>
15 <dateOccured type="단기">3740</dateOccured>
16 <dateOccured type="일본연호"/>
17 <dateOccured type="display">명 영락(永樂) 5</dateOccured>
18 </date>
19 </biblioData>
20 </front>
21 <text>
22 <content>
23 <paragraph align="center">七年(1407)<annotation type="원주">
24 <noteContent>[<index num="0029940_0" sort="K" type="지명">大明</index>
25 <index num="0029941_0" type="연호">永樂</index>五年]</noteContent>
26 </annotation>
27 </paragraph>
28 </content>
29 </text>
30 <level3 id="wca_107010">
31 <front>
32 <biblioData type="L">
33 <title>
34 <mainTitle>太宗實錄 七年 春正月</mainTitle>
35 </title>
36 </biblioData>
37 </front>

```

## XML 문서를 구성하는 두 요소

- **마크업(markup)**: 문서의 구조, 의미, 형식을 정의하기 위해 사용하는 모든 기호와 태그
- **내용(content)**: 태그 사이에 담긴 실제로 전달하고자 하는 순수 문자 데이터

## ‘마크업을 한다’는 것의 의미

- 단순한 텍스트 입력을 넘어, 비정형 데이터에 ‘의미의 테두리’를 입히는 과정
- 컴퓨터가 식별 가능한 ‘구조화된 정보’로 변환하는 전문적인 데이터 편찬 행위
- 이때 마크업을 실행하기 위해 사용하는 구체적인 기호를 ‘태그(Tag)’라고 함

<장소>운현궁 노락당</장소>

<일정>

<장소>운현궁 노락당</장소>

<특이사항/>

</일정>

**태그(tag):** 정보를 구별하고 성격을 나타내는 표식

- **시작 태그:** 특정 정보 단위의 시작을 알림. 예: <장소>
- **종료 태그:** 해당 정보 단위의 종료를 알림. 슬래시(/)로 시작함. 예: </장소>
- **단일 태그:** 내용(content)이 없는 요소에서 시작과 종료를 하나로 결합한 형태. 예: <특이사항/>

**요소(Element):** 구조화된 데이터의 기본 단위

- 시작 태그, 내용, 종료 태그를 포함한 독립적인 정보 객체
- 단순 요소(simple element): 하위 요소 없이 텍스트 내용만을 포함하는 경우
  - 예: <장소>운현궁 노락당</장소>
- 빈 요소(empty element): 내용 없이 존재 자체나 속성만으로 정보를 전달하는 경우
  - 예: <특이사항/>
- 복합 요소(complex element): 하나 이상의 하위 요소를 포함하여 계층 구조를 형성하는 경우
  - 예: <일정><장소>운현궁 노락당</장소><활동/></일정>

<장소 유형="건조물">운현궁 노락당</장소>

<사건 시작일="1866년2월15일" 종료일="1866년3월22일">고종.명성황후 가례</사건>

### 속성(Attribute)

- 요소에 대한 추가 정보

### 속성값(Attribute Value)

- 속성에 들어가는 실제 값

무엇을 요소로 두고 무엇을 속성으로 둘지는 설계의 문제

```
<여행정보>
  <제목>서울 궁궐 여행 일정</제목>
  <일정 순번="1">
    <장소>운현궁</장소>
    <좌표 경도="126.9870" 위도="37.5759"/>
    <시간>오전 10:00</시간>
    <활동>고종과 명성황후의 가례 재현 관람하기</활동>
  </일정>
</여행정보>
```

## 계층적 구조

- 요소가 다른 요소를 포함함으로써 형성되는 상하 관계
- 정보 간의 종속성과 포함 관계를 명확히 하여, 데이터의 위계를 논리적으로 설계 가능

## 계층 구조의 주요 구성 요소

- 루트 요소(root element): 문서 전체를 감싸는 최상위 요소. 하나의 XML 문서에는 반드시 단 하나의 루트 요소만 존재함. 예: <여행정보>
- 부모 요소(parent element): 하위 요소를 포함하고 있는 상위 요소. 예: <일정>은 <장소>, <좌표>, <시간>, <활동>의 부모 요소
- 자식 요소(child element): 특정 요소에 포함되어 있는 하위 요소. 예: <장소>는 <일정>의 하위 요소

```
<여행정보>  
  <제목>서울 궁궐 여행 일정</제목>  
  <일정 순번="1">  
    <장소>운현궁</장소>  
    <좌표 경도="126.9870" 위도="37.5759"/>  
    <시간>오전 10:00</시간>  
    <활동>고종과 명성황후의 가례 재현 관람하기</활동>  
  </일정>  
</여행정보>
```

### 빈 요소(empty element)

- 내용(텍스트) 없이 존재하는 요소

내용은 없지만 의미 정보는 존재할 수 있음



<본문>

<장소>운현궁</장소>의 안채인 <장소>노락당</장소> 당호를 쓴 편액이다. 노락당은 운현궁의 가장 중심이 되는 건물로서 그 규모는 궁궐에 비하여 손색없을 정도로 화려하고 웅장하였다. <인물>고종</인물>의 명에 따라 대제학 <인물>김병학(金炳學)</인물>이 지은 "<문헌>노락당기(老樂堂記)</문헌>"를 보면 "노락당과 하늘 사이가 한자 다섯 치 밖에 안된다"고 하여 노락당의 위용을 잘 표현하고 있다. 고종과 <인물>명성황후</인물>의 <사건>가례행사</사건>를 치룬 곳이기도 하다. 노락당의 편액 글씨는 <인물>신관호(申觀浩 1810-1888, 이후 신헌申樞으로 개명)</인물>가 썼다.

</본문>

### 문중 요소(In-Text Element)

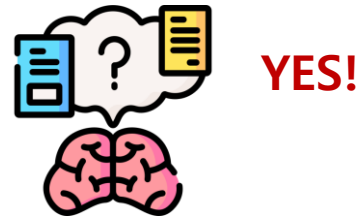
- 텍스트 내부에 직접 삽입되는 정보 요소
- 문서의 외형적 구조뿐만 아니라, 문장 속에 혼재된 지식 개체를 정밀하게 추출·식별하는 방법

### 문중 요소의 주요 기능

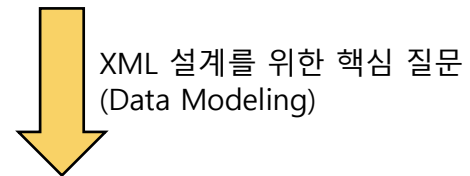
- 의미 구조의 가시화: 비정형 텍스트 안에 포함된 장소, 인물, 사건, 문헌 등을 마크업하여 데이터로서의 의미를 부여
- 세밀한 정보화: 단순한 텍스트 나열을 넘어, 문장 내부의 핵심 정보 단위를 기계적으로 식별 가능한 상태로 전환
- 맥락의 보존: 원문 텍스트의 서사적 흐름을 유지하면서도, 그 내부의 지식 구조를 입체적으로 드러냄



XML은 특정 분야에 고착된 언어가 아니므로,  
마크업의 대상과 범위에 대한 **절대적인 기준은 존재하지 않음**



"어떠한 형태로 만들어야 하는가"에 정해진 모범 답안은 없기에,  
자료의 성격과 연구 목적에 부합하는 **[전자문서(XML) 설계]** 과정이 반드시 선행되어야 함



- 정보 단위의 확정: 문헌의 방대한 내용 중 무엇을 독립적인 정보 단위로 간주할 것인가?
  - 요소(Element)의 선택: 무엇을 핵심적인 정보 요소로 표현할 것인가?
  - 속성(Attribute)의 할당: 무엇을 요소에 관한 부가적인 속성으로 처리할 것인가?
- 계층 구조의 최적화: 정보 간의 종속 관계를 어떻게 계층화(Hierarchy)하여 지식의 위계를 세울 것인가? 등