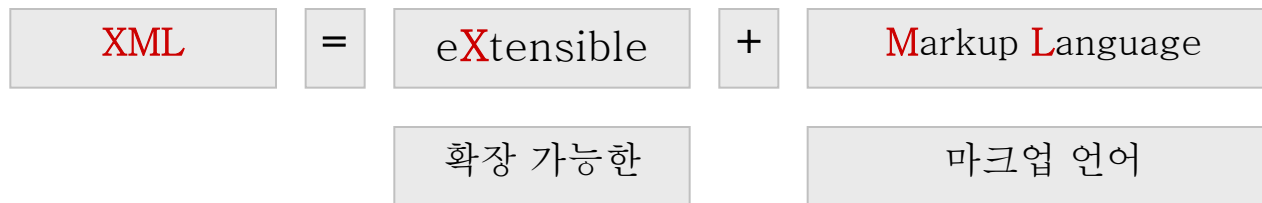


01. XML 기초

1. XML 소개

1-1. XML의 어원적 뜻

- XML(eXtensible Markup Language)은 “확장 가능한 마크업 언어”이다.



확장 가능

-기존에 없던 것을 새로 만들 수 있다.

마크업 언어

-문서의 내용을 구체적으로 전달하기 위한 추가적인 정보 표시 언어이다.
(예: HTML의 각종 태그)

XML은 “기존에 없던 것을 새로 만들 수 있는 마크업 언어”이다.
XML은 새로운 마크업 언어를 개발하기 위한 수단으로 사용된다.

1. XML 소개

1-2 마크업 언어

- **마크업 언어**란 문서의 내용에 추가적인 정보를 표시하기 위한 언어를 의미한다.
- 문서의 내용과 마크업 언어와의 관계



- 마크업을 사용하여 문서의 내용을 구조적으로 구분함으로써 내용을 쉽게 이해할 수 있게 된다.
- 마크업 언어는 문서의 내용을 조직화 하고 구조화 시켜 문서의 내용을 정확하게 전달한다.

1. XML 소개

1-3 확장 가능한 의미

- “확장 가능”이란 “기존에 없던 것을 새로 만들 수 있다.”는 의미이다.

고정 DTD

HTML 문서의 작성 예

```
<html>
  <body>
    <table>
      <tr>
        <td>XML 기초</td>
        <td>홍길동</td>
      </tr>
    </table>
  </body>
</html>
```

정해진 **태그**만 사용할 수 있음

사용자 정의
DTD, Schema

XML 문서의 작성 예

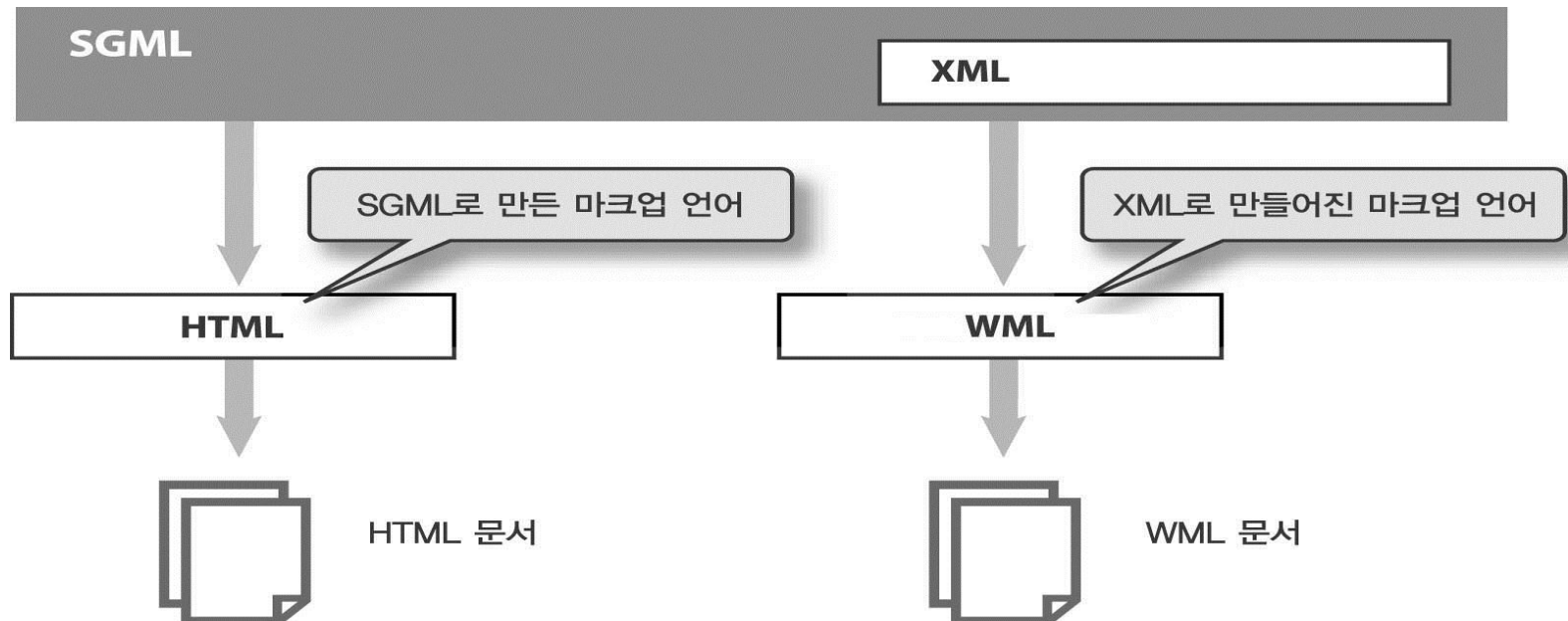
```
<?xml version="1.0"?>
<booklist>
  <book>
    <title>XML 기초</title>
    <author>홍길동</author>
  </book>
</booklist>
```

새로운 **마크업**을 개발하여 문서를
작성할 수 있음

1. XML 소개

1-4 XML은 메타언어

- XML은 **메타언어(Meta-Language)**로써 마크업 언어를 개발하기 위한 언어이다.
- 대표적인 메타언어는 **SGML(Standard Generated Markup Language)**이며, HTML은 SGML로 개발된 웹 문서 작성용 마크업 언어이다.
- XML은 SGML과 같은 메타언어이므로 새로운 마크업 언어를 만드는 도구로 사용된다. “XML로 문서를 작성한다”라는 말은 XML로 개발된 특정 마크업 언어로 문서를 작성하는 것을 의미한다.



1. XML 소개

1-5. SGML과 XML 비교

- SGML은 1986년 ISO(International Standardization Organization)에서 표준으로 발표.
- 문법이 복잡하고, 응용 S/W를 구현하기 어려워 주로 공공기관 및 출판업체에서 제한적으로 사용.
- 1996년 W3C의 SGML Work Group에서 SGML의 부분집합(Subset)으로서 SGML의 장점을 취하고, 복잡성을 배제한 XML을 개발

1-6. W3C에서 하는 일

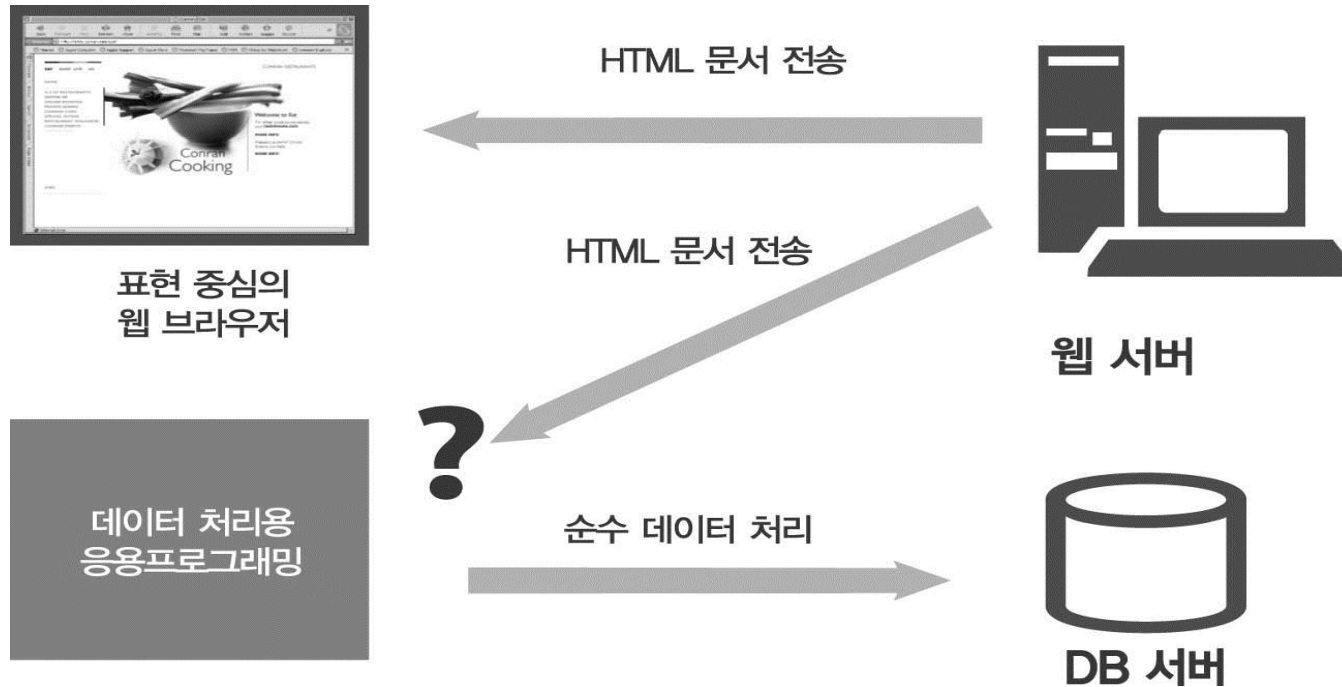
- 1994년 10월 Tim Berners-Lee(web의 창시자) 설립.
- W3C에서는 각종 스펙(Spec)을 연구하고 발표하는 일을 한다.

국제표준화기구 (ISO)에서 정한 규격 → 표준(standardization)
W3C에서 정한 규격 → 권고안(Recommendation)

2. XML 문서 사용 용도

2-1. 데이터 전달용 문서

- 웹 서버가 데이터의 내용과 표현 형식을 HTML 문서에 함께 담아서 클라이언트로 보내면 웹 브라우저는 태그가 의미하는 표현 방법으로 그 데이터를 디스플레이 해준다.
- HTML 문서를 **표현 중심**이 아닌, **정보 전달**용으로 사용했을 때의 문제점
→ HTML 문서 내용에서 데이터를 추출할 수는 있지만, 데이터의 의미는 알기 어려움



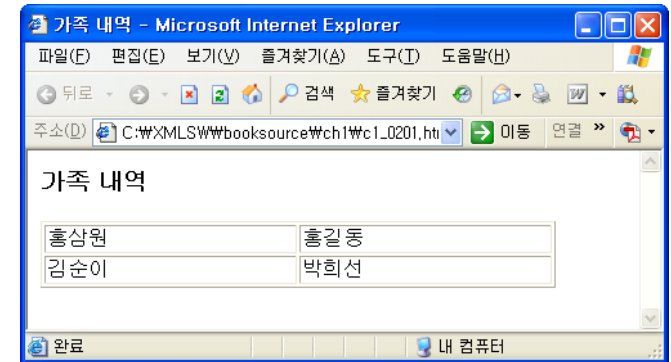
2. XML 문서 사용 용도

2-1. 데이터 전달용 문서(HTML의 한계점)

- HTML 문서는 순수 데이터를 추출하기 어렵다.

HTML 문서

```
<html>
<head>
  <title>가족 내역</title>
</head>
<body>
  <h3>가족 내역</h3>
  <table border="1" width="400">
    <tr> <td>홍삼원</td> <td>홍길동</td> </tr>
    <tr> <td>김순이</td> <td>박희선</td> </tr>
  </table>
</body>
</html>
```



- 데이터 추출에는 성공 했지만, 데이터가 의미하는 내용은 알 수 없다.
- HTML 문서를 데이터 처리용 응용프로그램에서 이용한다면 추출한 데이터를 적절히 이용할 수 없다.

2. XML 문서 사용 용도

2-1. 데이터 전달용 문서(계속)

< 해결 방법 >

- ① HTML에서 가족 관계를 나타내는 태그를 만든다. ➔ 불가능(이미 태그들이 고정되어 있음)
- ② HTML 문서의 문제점을 극복한 XML 문서를 사용한다.

XML 문서

```
<?xml version="1.0"?>
<가족관계>
  <아버지>홍삼원</아버지>
  <어머니>김순이</어머니>
  <아들>홍길동</아들>
  <며느리>박희순</며느리>
</가족관계>
```

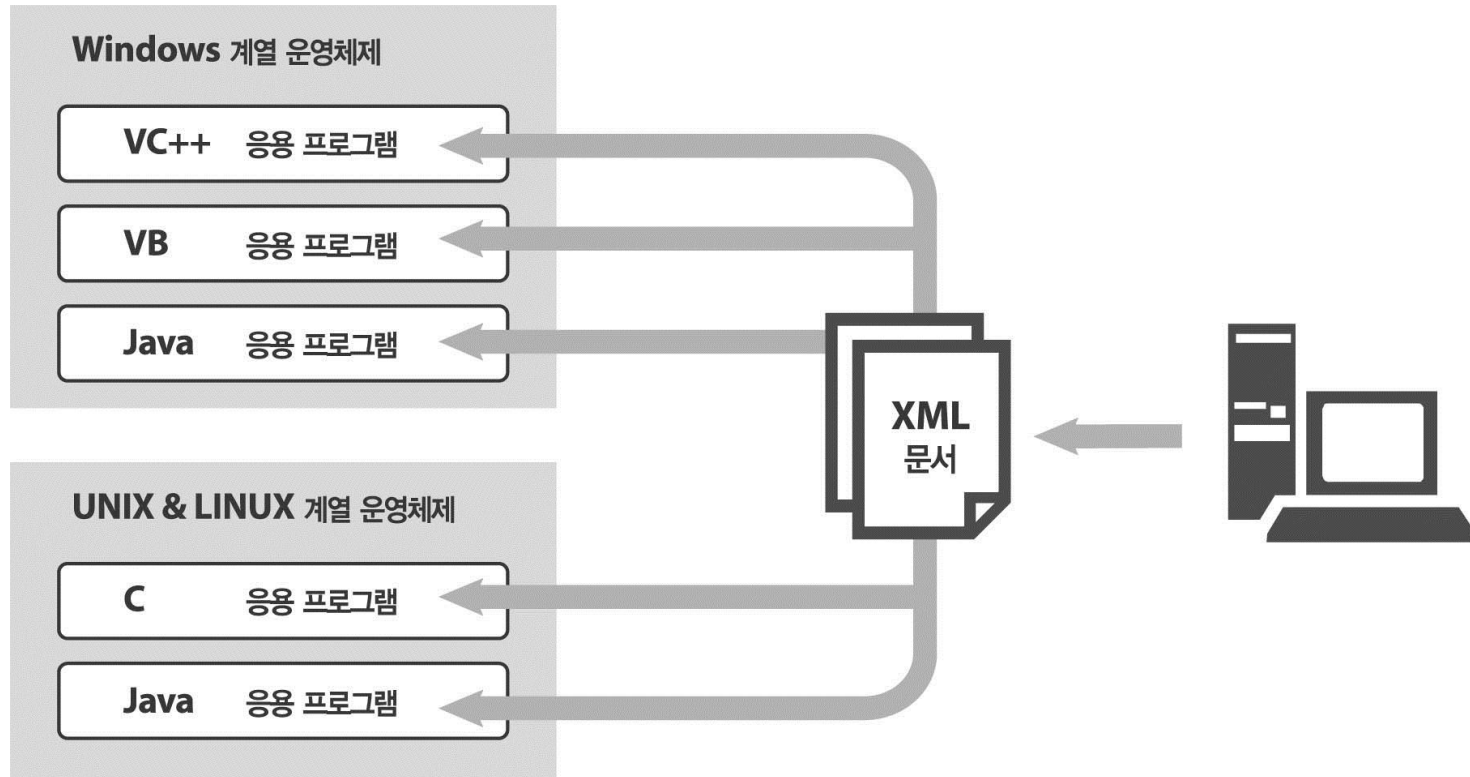
조직화, 구조화된
데이터 전달

➔ XML 문서는 응용 프로그램으로 하여금 쉽고 정확하게 데이터를 처리할 수 있도록 데이터를 조직화되고 구조화 시킨다.

2. XML 문서 사용 용도

2-1. 데이터 전달용 문서(계속)

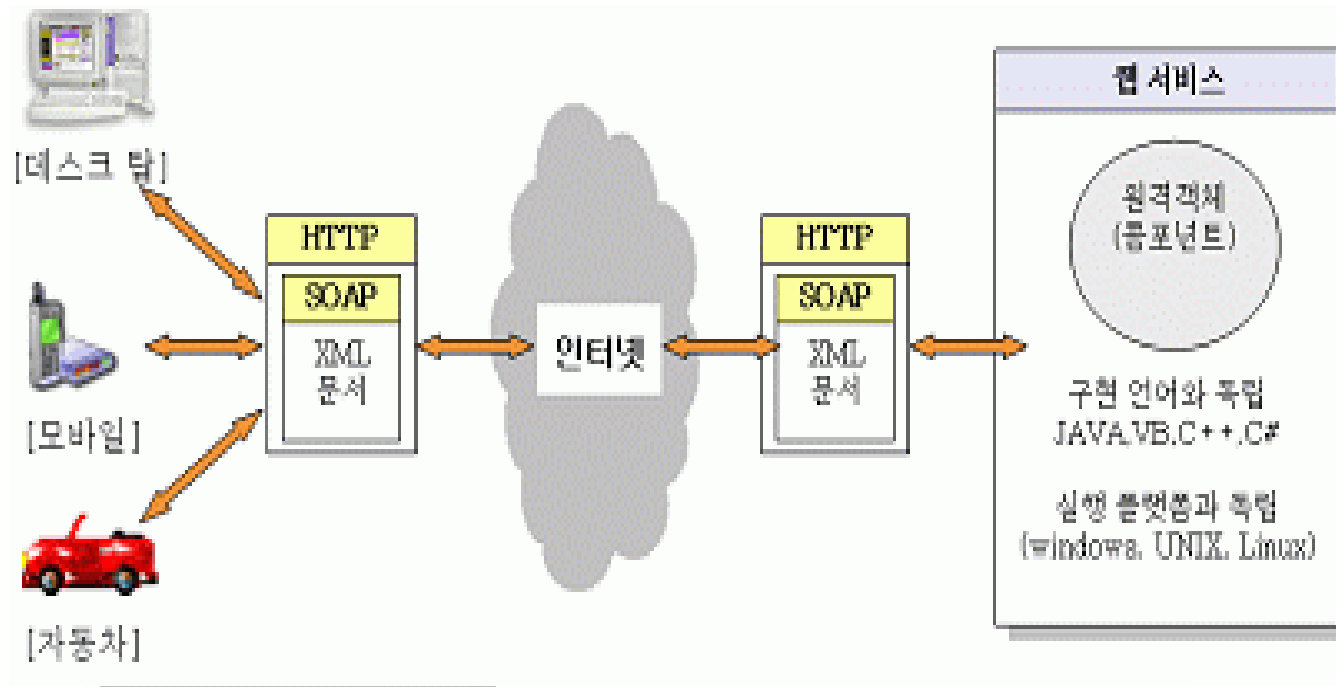
- 데이터를 처리하는 응용프로그램도 XML 문서가 Text형식의 **유니코드(Unicode)**로 작성되기 때문에 유니코드를 인식하는 모든 운영 체제상의 응용 프로그램과도 호환성 있는 문서가 된다.



2. XML 문서 사용 용도

2-2. XML 문서의 적용 분야

- XML 문서의 핵심은 시스템 통합
- 이기종 시스템 정보 교환 분야
- 웹 서비스 분야
 - 유비 쿼터스(Ubiquitous) : “언제 어디서나 동시에 존재한다”. Anywhere, Anytime, Any device → 3A



2. XML 문서 사용 용도

2-3. HTML에 대한 잘못된 오해

- XML이 HTML을 대신하지는 않는다. XML은 HTML의 대체 언어가 아니며, 이 두 가지 언어는 상호 보완적인 언어로 앞으로도 계속 사용될 것이다.
- XML은 HTML을 확장한 것이 아니다. XML이 데이터 전달용으로 사용될 경우 HTML 문서 보다 더욱더 좋은 해결책일 뿐이다.
- 전송용 데이터는 XML로 개발된 마크업 언어에 의해 XML 문서로 조직화, 구조화시켜 네트워크를 통해 원격 전송하면 좋다.
- 표현 중심의 문서를 작성할 때는 HTML이 가장 좋은 언어이다.

02. XML 문서작성방법

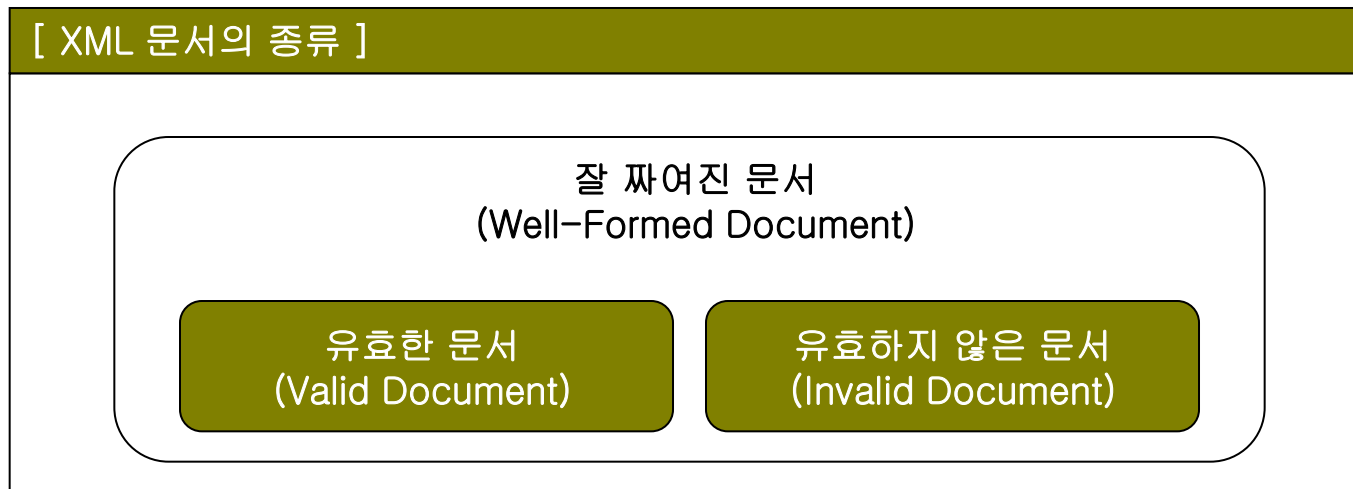
1. XML 문서의 종류

1-1 잘 짜여진(Well-Formed XML Document) 문서

XML 1.0 권고안에 언급되어 있는 문법(Spec)을 잘 지켜서 작성된 문서를 말한다.

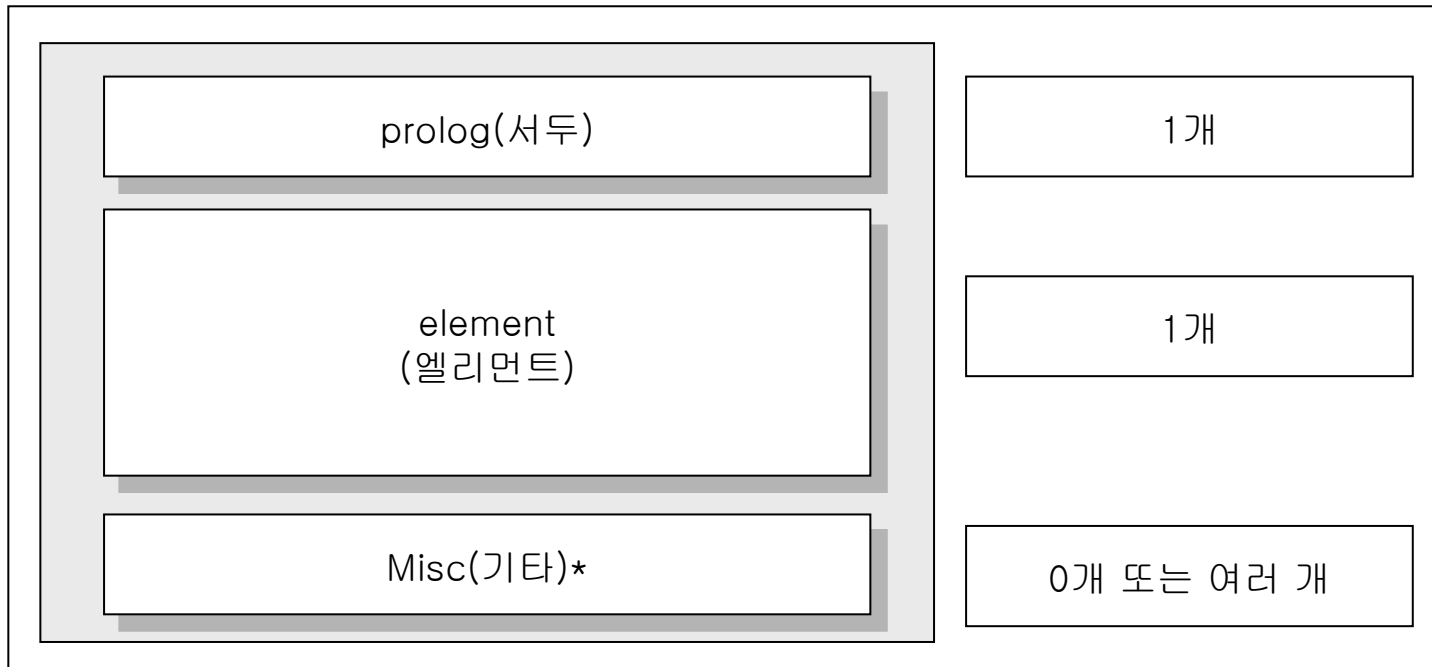
1-2 유효한(Validate) 문서

유효한 문서란 잘 짜인 문서이면서, XML로 개발된 특정 마크업 언어에 맞게 작성된 문서를 의미한다. 즉, DTD나 Schema에서 작성한 규정을 따르는 문서인 것이다.



2. XML 문서의 구조

- XML 문서의 구조는 서두와 몸체 그리고 기타 부분으로 구성된다.
- 서두는 XML 선언, **PI**(**P**rocessing **I**nstruction), 문서 유형 선언으로 구성된다.
- 몸체 부분에는 한 개의 루트 엘리먼트(Root Element)와 루트 엘리먼트의 자식(하위, Child) 엘리먼트로 작성되어야 한다.
- 기타 부분은 주석 및 PI, 공백으로 구성 되는데 일반적으로 생략된다.



2. XML 문서의 구조

XML 문서

```
<?xml version="1.0" encoding="euc-kr"?>
```

```
<!-- 문서 유형 선언 -->
```

```
<!DOCTYPE booklist SYSTEM "bml.dtd">
```

```
<!-- 문서의 구조를 xhtml 문서로 변경 -->
```

```
<?xml-stylesheet type="text/xsl" href="bml.xsl"?>
```

```
<booklist>
```

```
  <book id="b1" kind="k2">
```

```
    <title>XML 기초서</title>
```

```
    <author>신민철</author>
```

```
    <publisher>프리렉</publisher>
```

```
  </book>
```

```
<book id="b2" kind="k1">
```

```
  <title>가을엔 사랑을 느끼세요</title>
```

```
  <author>이사랑</author>
```

```
  <publisher>가을문화사</publisher>
```

```
</book>
```

```
</booklist>
```

[서두]

[엘리먼트]

3. XML 선언

3-1 XML 선언 문법

- 현재 작성중인 문서가 XML 문서임을 명시적으로 표현하는 것이 XML 선언이다.
- XML 선언은 반드시 XML 문서 첫 줄에 기술되어야 한다.

```
<?xml version="버전번호" encoding="인코딩방식" standalone="yes|no"?>
```

- XML 선언의 시작은 “<?xml”로 시작하며 ‘<?’와 ‘xml’ 문자 사이에 공백을 두어서는 안 된다.

잘못된 선언 예제

① <!-- XML 문서 선언 -->
 <?xml version="1.0" encoding="euc-kr"?>

② <? xml version="1.0" encoding="euc-kr"?>

3. XML 선언

3-2 XML 선언에서 사용되는 속성들

(1) version (버전)

- XML 선언에서 version 속성은 반드시 기술해야 한다.

```
<?xml version="1.0"?> or <?xml version='1.0'?>
```

(2) encoding (인코딩)

- 작성하는 XML 문서를 어떤 인코딩 방식으로 저장할 것인가를 지정한다.
- 생략되면 XML 프로세서는 디폴트 유니코드 인코딩 방식인 UTF-8로 처리한다.

```
<?xml version="1.0" encoding="euc-kr"?>  
<?xml version='1.0' encoding="UTF-8"?>
```

(3) standalone (스탠드얼론)

- 파서가 XML 문서를 해석할 때 외부 DTD 문서 참조여부를 파서에게 전달한다.
- 생략되면 XML 프로세서는 디폴트로 'no' 값으로 처리하게 된다.

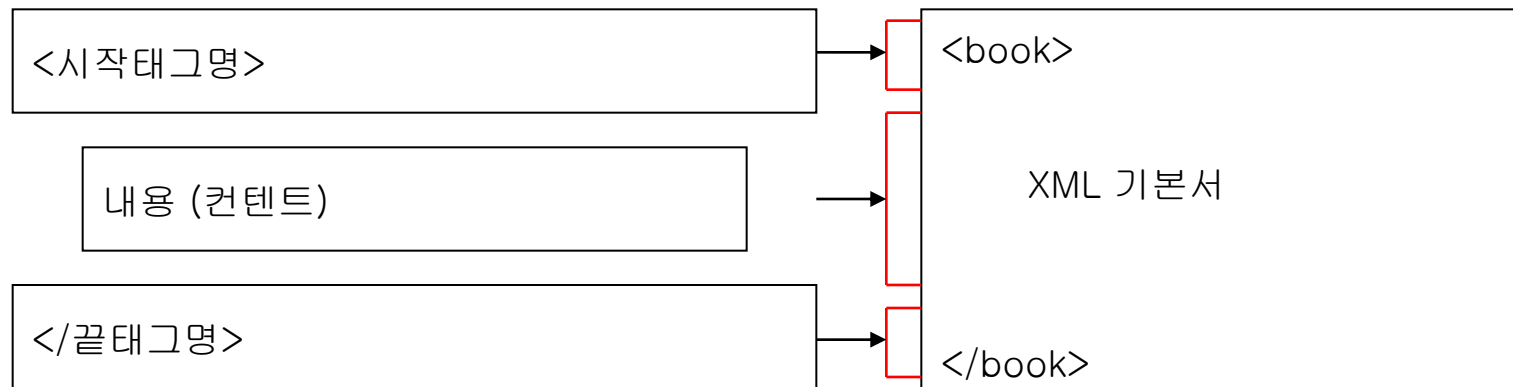
```
<?xml version="1.0" encoding="euc-kr" standalone="no"?>  
<?xml version='1.0' encoding="UTF-8" standalone="yes"?>
```

4. 엘리먼트

4-1 엘리먼트 작성 문법

- 모든 XML 문서는 단 하나의 루트 엘리먼트(root element)를 갖는다.
- 엘리먼트는 시작 태그와 끝 태그로 구성되며 태그명은 동일해야 한다.
- 엘리먼트는 추가적인 정보를 나타내는 속성(Attribute)을 가질 수 있다.
- 시작태그와 끝 태그 사이에는 엘리먼트의 실질적인 내용(Content)이 오는데, 문자 데이터 및 자식 엘리먼트가 올 수 있다.

[엘리먼트의 구성 요소]



4. 엘리먼트

4-2 엘리먼트의 종류

(1) 내용을 가지는 엘리먼트

- 문자 데이터나 자식 엘리먼트를 내용으로 갖는 엘리먼트

```
<book>  
  <title>자연과 인간</title>  
</book>
```

(2) 내용이 없는 빈 엘리먼트(Empty Element)

- 문자 데이터나 자식 엘리먼트를 갖지 않는 엘리먼트

```
<image src="./temp/image1.gif"></image>  
또는  
<image src="./temp/image1.gif"/>
```

4. 엘리먼트

4-3 엘리먼트 작성시 주의할 점

엘리먼트 작성시 주의할 점	
1	시작태그와 끝태그는 반드시 짝을 이루어야 한다. 단 내용이 없는 빈 엘리먼트는 시작태그의 끝에 '/'를 붙여주어 시작태그인 동시에 끝태그임을 표시한다.
2	속성은 반드시 속성명=속성값 형태로 사용해야 하며 속성값은 반드시 “ ” 또는 ‘ ’로 감싸야 한다. 한 엘리먼트에 같은 속성명은 두 개 이상 올 수 없다.
3	태그를 나타내는 ‘<’ 문자는 엘리먼트 내용(Content)인 문자 데이터 및 속성값으로 사용할 수 없다. ‘>’ 문자는 사용해도 되지만 가급적 사용하지 않는다.
4	태그 이름 중간에는 공백 문자가 올 수 없으며, 시작태그와 끝태그 이름은 같아야 한다.
5	엘리먼트는 중첩되어 작성될 수 없다. 이것은 시작태그와 끝태그 사이에 다른 엘리먼트의 시작태그가 존재할 수 없음을 의미다.
6	태그 이름은 이름 작성 규칙을 따라야 한다.

4. 엘리먼트

이름 작성 규칙	
1	이름은 문자(한글도 포함)로 시작할 수 있지만 숫자나 ‘.’으로 시작할 수는 없다.
2	두 번째 문자부터는 숫자 및 “_”, “-“, “.” 도 가능하다.
3	태그 이름에 공백을 포함시킬 수는 없다.
4	‘:’ 문자는 쓸 수는 있지만 네임스페이스에 관련된 기호이므로 사용하지 않는다.
5	태그 이름은 대소문자를 구별하므로 철자가 같다고 해서 같은 태그가 아니다.
6	시작태그의 ‘<’ 다음에 공백 문자가 올 수 없으며, 시작과 종료태그 이름은 같아야 한다.
7	태그 이름은 xml로 시작할 수 없다.

< 올바른 태그 이름 >

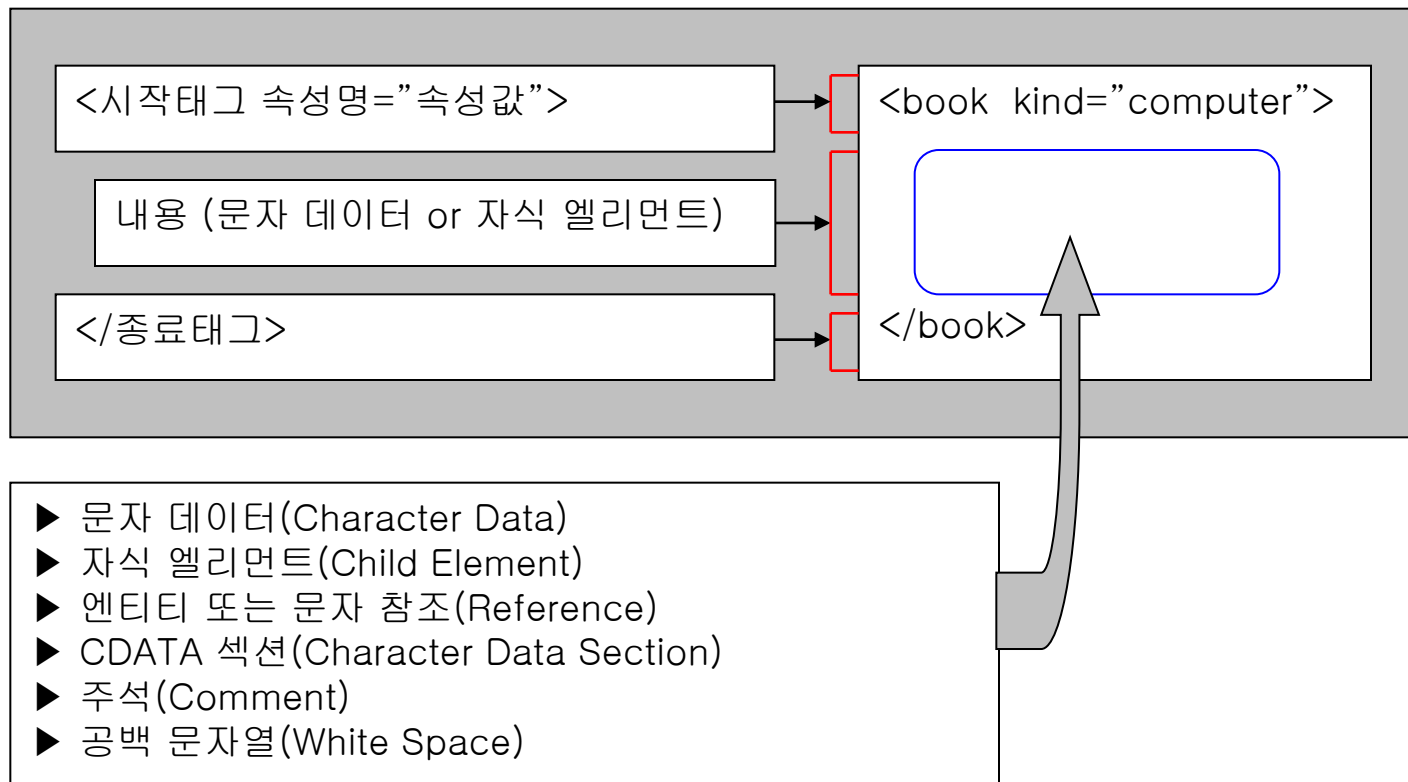
<book>
<_book>
<책>
<book1>
<book-1>
<Book>

< 잘못된 태그 이름 >

<7Book>	첫 글자는 숫자를 사용할 수 없다.
<c++>	‘_’, ‘-’, ‘.’, ‘:’ 이외의 특수 문자는 사용할 수 없다.
<book list>	태그 이름에 공백을 사용할 수 없다.
< book>	‘<’ 다음에 공백을 두어서는 안 된다.
<xml-book>	태그 이름이 xml로 시작하면 안 된다.

5. 엘리먼트 내용

5-1 엘리먼트 내용으로 올 수 있는 것들



5. 엘리먼트 내용

5-2 문자 데이터

- 문자 데이터(Character Data)란 XML 프로세서가 해석할 수 있는 내용 중에서 **마크업**을 제외한 부분을 말한다.

< XML 1.0 권고안의 마크업(markup) >

■ XML 선언	<?xml version="1.0" encoding="euc-kr"?>
■ 문서 유형 선언	<!DOCTYPE booklist SYSTEM "bml.dtd">
■ 프로세싱 지시자(PI)	<?xml-stylesheet type="text/xsl" href="bml.xsl"?>
■ 주석	<!-- 주석 내용 -->
■ 시작태그 및 끝태그	<book> </book>
■ 빈 엘리먼트 태그	<imgae src="image1.gif"/>
■ 엔티티 참조	DTD에 정의되어 있는 엔티티 참조 (예) &pub1;
■ 문자 참조	
진수; 진수;
■ CDATA 섹션 구분자	<![CDATA[문자 데이터]]>
■ 최상위 공백 문자열	XML 문서 구성요소 중 루트 엘리먼트 외부에 있는 공백 문자열
■ Text 선언	<?xml version="1.0" encoding="euc-kr"?>

5. 엘리먼트 내용

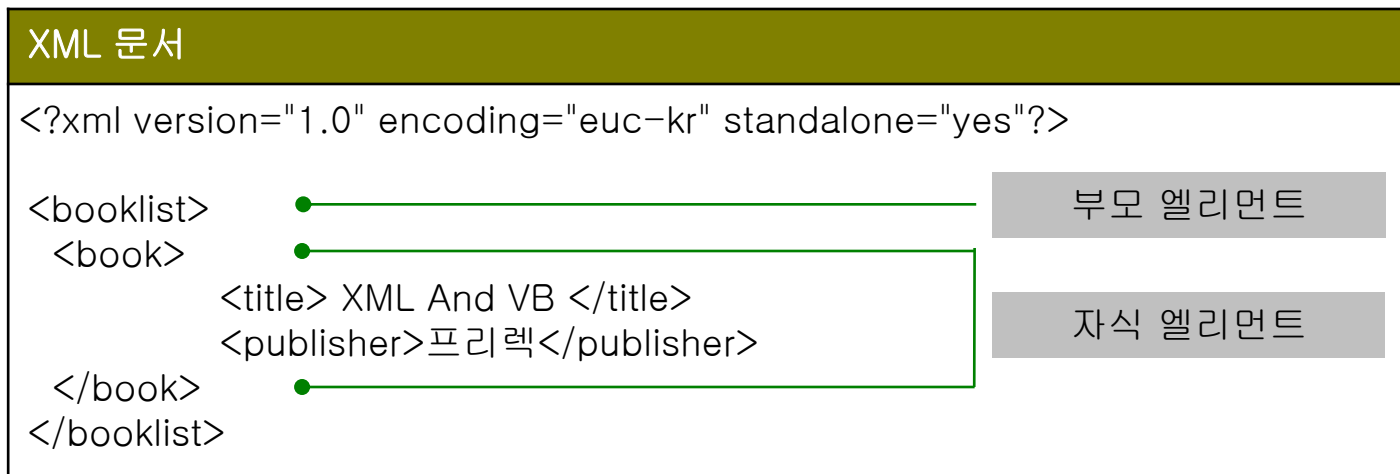
- 문자 데이터 내에는 ‘&’ 문자와 ‘<’ 문자를 사용할 수 없다.
- ‘&’ 문자는 엔티티 참조의 시작을 의미하며, ‘<’ 문자는 엘리먼트의 태그, **CDATA 섹션**의 시작을 의미하기 때문이다.
- 빌트인(Built-in) 엔티티**의 참조 또는 문자 참조로 사용이 가능하다.

문자	엔티티 참조	문자 참조	사용 예
&	&	16진수(&) 10진수(&)	<title> XML &amp; Java </title>
<	<	16진수(<) 10진수(<)	<식> 3 &lt; 5 </식> <식> 3 &#x3C; 5 </식>
>	>	> >	<식> 5 &gt; 3 </식> 속성값에 > 문자를 삽입할 경우
“	"	" "	<book kind=” &quot; computer &quot; ”> 속성값에 “ 문자를 삽입할 경우
‘	'	' '	<book kind=’mincheol &apos; s book’> 속성값에 ‘ 문자를 삽입할 경우

5. 엘리먼트 내용

5-3 자식 엘리먼트

- 엘리먼트의 내용으로 자식(Child) 엘리먼트를 포함할 수 있다.



5-4 엔티티 참조

- 엔티티 참조(Entity Reference)는 자주 쓰이는 내용을 엔티티로 정의하고, XML 문서에서 엔티티로 정의된 내용과 동일한 내용이 작성되어야 할 부분에 엔티티 참조를 사용한다.

5. 엘리먼트 내용

5-5 문자 참조

- 문자 참조(Character Reference)는 문자 집합 코드표상에 언급되어 있는 코드값을 직접 사용하여 문자를 나타내는 것이다.

&#x(16진수 코드값); 또는 &#(10진수 코드값);

문자	16진수	10진수
스페이스	 	
탭				

XML 문서

```
<?xml version="1.0" encoding="euc-kr"
standalone="yes"?>
<booklist>
  <book>
    <title>대망</title>
    <publisher>믿음 문화사</publisher>
    <price>&#xA5;20</price>
  </book>
</booklist>
```

엔화

5. 엘리먼트 내용

5-6 CDATA 섹션

- 대부분의 문자 데이터인 **PCDATA**(**P**arsed **C**haracter **D**ATA)는 XML 파서가 해석(Parsing)하는 데이터를 말한다.
- CDATA 섹션(Section)** 내에 정의된 문자 데이터는 XML 프로세서가 해석하지 않고 바로 응용프로그램(Application)에게 전달한다.

```
<![CDATA[ 문자 데이터 ]]>
```

XML 문서

```
<?xml version="1.0"?>
<booklist>
  <book>
    <title>
      <![CDATA[ XML & JAVA ]]>
    </title>
  </book>
</booklist>
```

5. 엘리먼트 내용

5-7 공백 문자열

- 공백 문자열(White Space)이란 하나 이상의 공백 문자들로 구성된 문자열을 말한다. XML 1.0 권고안에서는 스페이스(#x20), 탭(#x9), 캐리지 리턴(#xd), 라인 피드(#xa)만을 공백 문자로 분류하고 있다. XML 프로세서는 공백 문자열을 하나의 공백 문자로 취급한다.

6. 속성

6-1 속성 작성 문법

- 속성(Attribute)은 엘리먼트에 추가적인 정보를 줄 수 있다.
- 속성은 속성명=속성값으로 표기하며, '=' 앞과 뒤에 공백을 둘 수 있다.
- 한 개의 엘리먼트는 같은 속성명을 2번 이상 기술할 수 없다.

```
<시작태그명 속성명="속성값" 또는 속성명='속성값'>  
  내용  
</끝태그명>
```

- 속성명은 XML 권고안에 언급되어 있는 이름 작성 규칙을 따라야 한다.

속성 사용 예제

```
<book id="id" kind='컴퓨터'>  
<image src="background.gif"/>
```

7. 주석

- 주석은 XML 문서를 작성하는 사람은 물론 XML 문서를 읽는 사람이 좀더 쉽게 이해할 수 있도록 덧붙인 설명이다.

```
<!-- 주석 내용 -->
```

XML 문서

```
<?xml version="1.0" encoding="euc-kr" standalone="yes"?>
```

```
<!-- 루트 엘리먼트 -->
```

```
<booklist>
```

```
<!--
```

```
<book>
```

```
<title>XML 기초서</title>
```

```
<publisher>&pub1;</publisher>
```

```
</book>
```

```
-->
```

```
<book>
```

```
<title> <!-- 알기 쉬운 --> XML 기초서</title>
```

```
<publisher>&pub1;</publisher>
```

```
</book>
```

```
</booklist>
```

8. 프로세싱 지시자

- 프로세싱 지시자(Processing Instruction)는 PI는 해당 XML 문서를 처리하는 응용프로그램(Application)에게 XML 문서의 처리 방법을 지시하는 내용을 담고 있다.