제 3 장 XML 문서 작성 방법

1 절. XML 문서의 종류

2 절. EBNF 표기법

3 절. XML 문서 구조

4 절. XML 선언

5 절. 인코딩 및 유니코드

6 절. 엘리먼트(Element)

7 절. 엘리먼트(Element) 내용

8 절. 속성(Attribute)

9 절. 주석(Comment)

10 절. 프로세싱 지시자

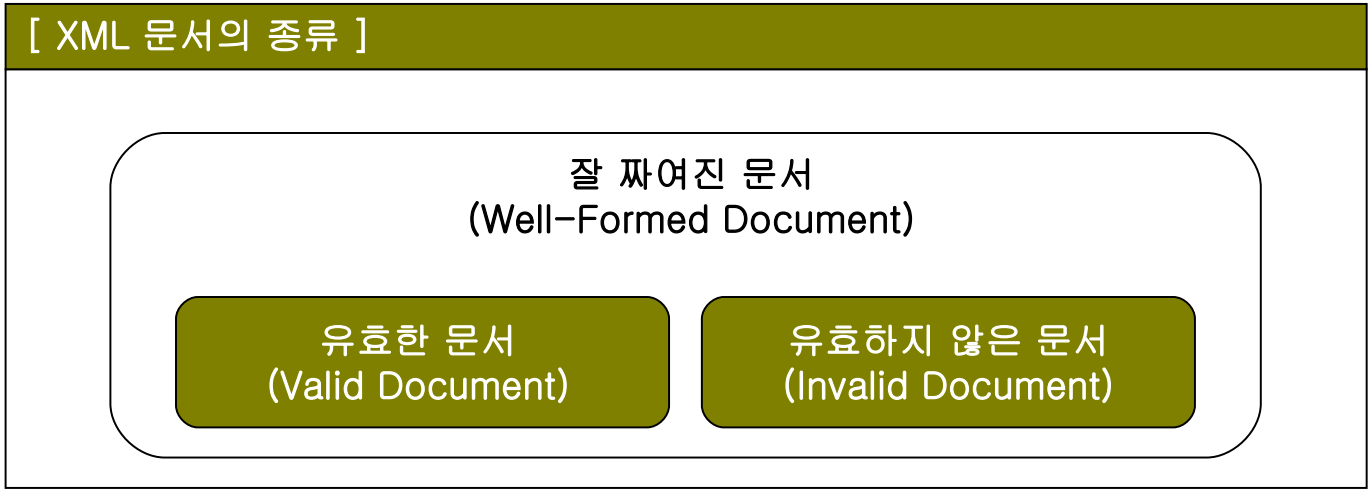
1. XML 문서의 종류

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. EBNF 표기법

2-1 EBNF 문법 형태

- XML 1.0 권고안의 공식 문법은 EBNF(Extended Backus-Naur Form) 표기법을 사용한다.

기호 ::= 표현식 (Symbol ::= Expression)

< 예제 >
Char1 ::= [a-z]
VersionNum ::= [0-9]
Word1 ::= “version” 또는 Word1 ::= ‘version’

2. EBNF 표기법(계속)

2-2 패턴 결합

- 표현식은 다음과 같은 패턴과 결합하여 좀더 복잡한 형태를 가질 수 있다.

패턴	설명
A?	A표현이 올 수도 안 올 수도 있다(옵션).
A, B	A표현이 먼저 오고 B표현이 나중에 온다(순차).
A|B	A와 B 표현 중 하나만 와야 한다(선택).
A-B	A표현이 와야 되지만 B표현과는 일치되지 않아야 한다.
A+	A표현이 최소한 한 개 이상이 와야 한다.
A*	A표현이 안 올 수 있고, 한 개 이상이 와도 된다.

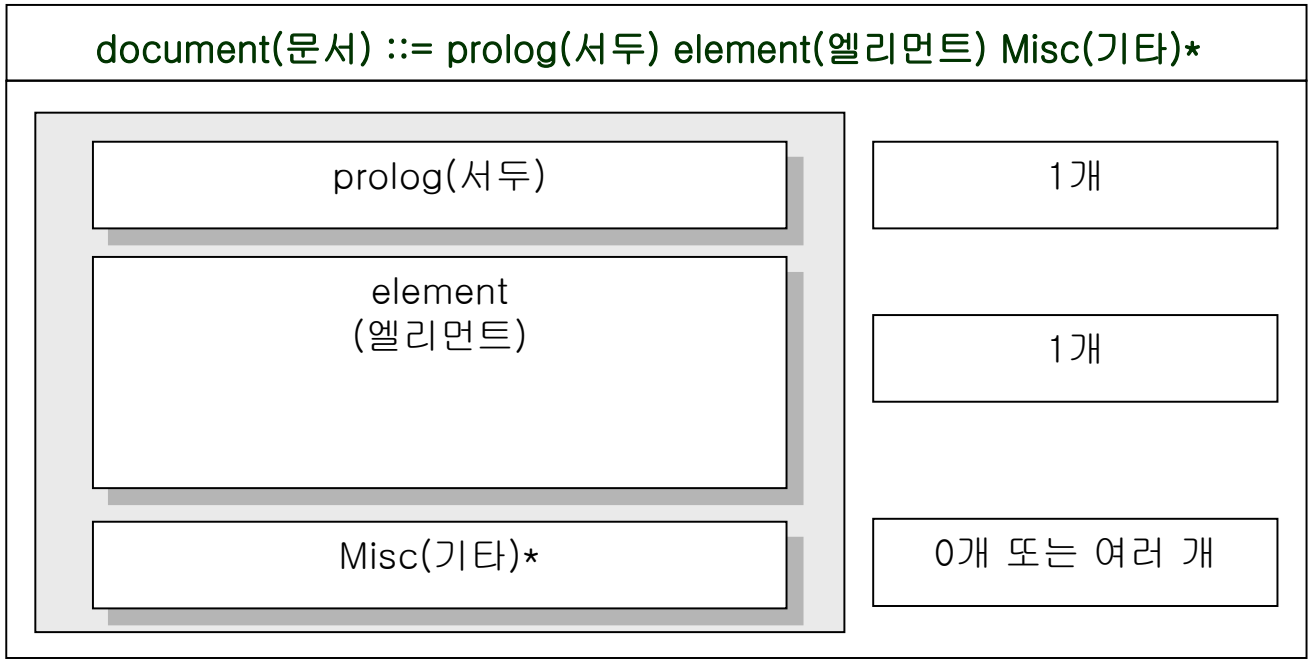
- 사용 예제

```
Version ::= S 'version' Eq ("'" VersionNum "'" | '"' VersionNum '"')
Eq ::= S? '=' S?
VersionNum ::= ([a-zA-Z0-9.] | '-')+
```

3. XML 문서의 구조

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

< XML 1.0 권고안의 문서에 대한 EBNF 문법 >



3. XML 문서의 구조(계속)

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>Struts & XML</title>
    <author>홍길동</author>
    <publisher>중앙출판사</publisher>
  </book>
<book id="b2" kind="k1">
    <title>가을엔 사랑을 느끼세요</title>
    <author>이사랑</author>
    <publisher>가을문화사</publisher>
  </book>
</booklist>
```

[서두]

[엘리먼트]

4. XML 선언

4-1 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"?>

4. XML 선언(계속)

4-2 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"?>
```

5. 인코딩 및 유니코드

5-1 인코딩 (Encoding)

- XML 문서는 단순 텍스트 형식의 파일이기 때문에 여러 언어 형태로 작성될 수 있다.
- 문자 코드(Character Code)란 문자들의 집합과 이 문자들을 나타내기 위한 숫자(문자 코드)들을 1대 1로 연결시켜 놓은 것을 말한다.
- 문자 코드를 컴퓨터가 이해할 수 있는 코드 형태로 만들어 주는 것을 문자 인코딩(Character Encoding)이라고 한다.

5-2 KS C 5601과 EUC-KR 인코딩

- 한국표준문자집합(KS C 5601)은 한국표준협의회가 한국공업표준(Korea Industrial Standard)으로 정한 정보처리분야(C)의 5601번 표준안이며, 2 바이트를 사용해서 완성형 한글을 표현하는 방법을 기술하고 있다.
- EUC-KR 인코딩은 Bell Laboratories에서 개발한 확장 유닉스 코드 (Extended UNIX Code)의 변형으로, 유닉스 운영체제에서 영어는 KS C 5636을, 한글은 KS C 5601을 사용하는 것을 말한다. 즉 ASCII 문자 코드는 1바이트로 표현하고 'KS C 5601'에 언급되어 있는 한글 문자 코드는 2 바이트로 표현한다.

5. 인코딩 및 유니코드(계속)

5-3 유니코드(Unicode)

- 유니코드는 기존 언어의 인코딩 체계를 모두 포함할 수 있도록 고안된 거대한 문자 집합이다.
- 유니코드 인코딩 방식을 사용하여 문서를 작성하면 하나의 문서를 다양한 언어로 작성할 수 있다.

http://www.unicode.org/standard/WhatIsUnicode.html

- UTF-8 방식은 기존의 ASCII 문자 코드 체계와 호환이 가능하기 때문에 인터넷상에서 문서를 교환하기 위한 기본적인 인코딩으로 사용되고 있다.
- UTF-16은 2바이트를 사용하여 모든 문자 코드를 표현한다.
- 영어로 작성된 문서인 경우에는 대부분의 문자들이 ASCII 문자 인코딩에 적합하기 때문에 UTF-16보다 UTF-8이 결과적으로 파일 사이즈를 작게 만든다. 하지만 영어 이외의 다른 문자가 있는 경우에는 UTF-8보다 UTF-16이 파일 사이즈를 더 적게 만든다.

5-4 XML 문서 인코딩

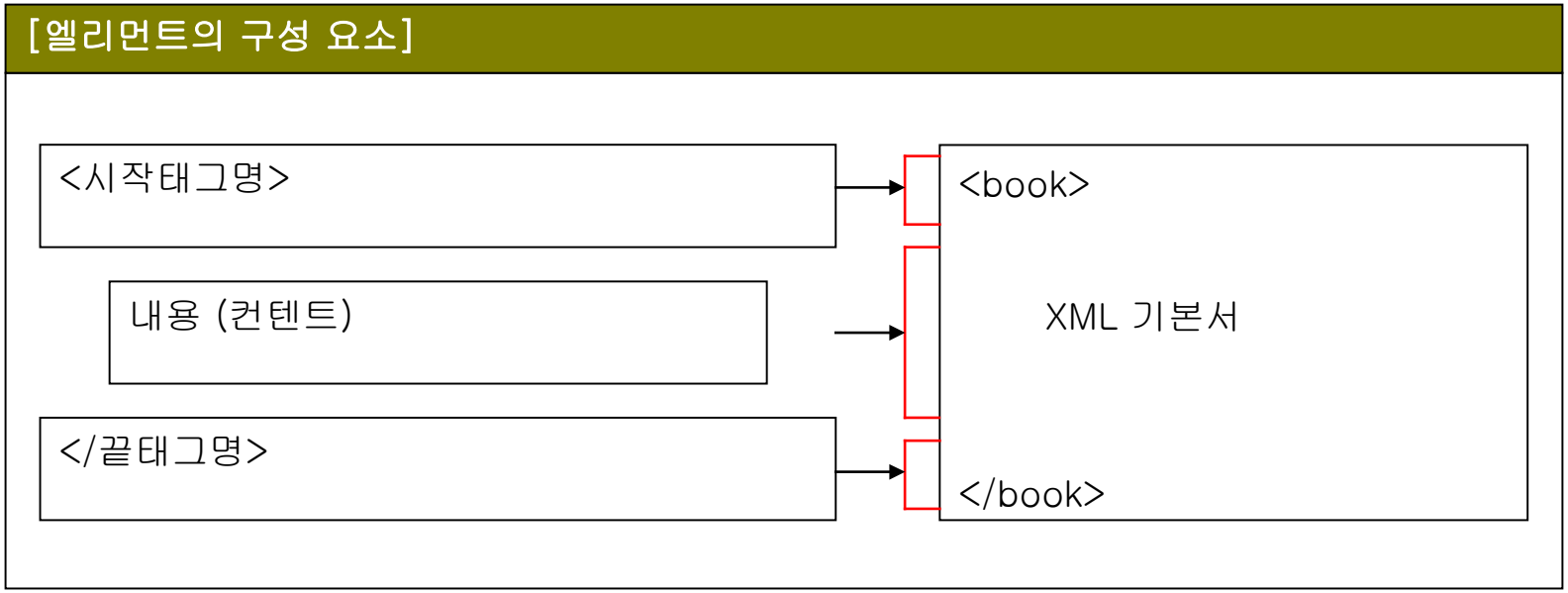
- XML 권고안은 모든 XML 문서는 유니코드 인코딩 방식인 UTF-8 인코딩 방식으로 저장할 것을 기본으로 하고 있지만, “euc-kr” 인코딩 방식도 ASCII는 1바이트로 한글은 2바이트로 저장되기 때문에 encoding 속성값으로 “euc-kr” 을 지정해도 무관하다.

6. 엘리먼트

6-1 엘리먼트 작성 문법

- 모든 XML 문서는 단 하나의 루트 엘리먼트(root element)를 갖는다.
- 엘리먼트는 시작 태그와 끝 태그로 구성되며 태그명은 동일해야 한다.
- 엘리먼트는 부가적인 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]



6. 엘리먼트

6-2 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="D:₩temp₩image1.gif"/>
  또는
<image src="D:₩temp₩image1.gif"></image>
```

6. 엘리먼트

6-3 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.

6. 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 >

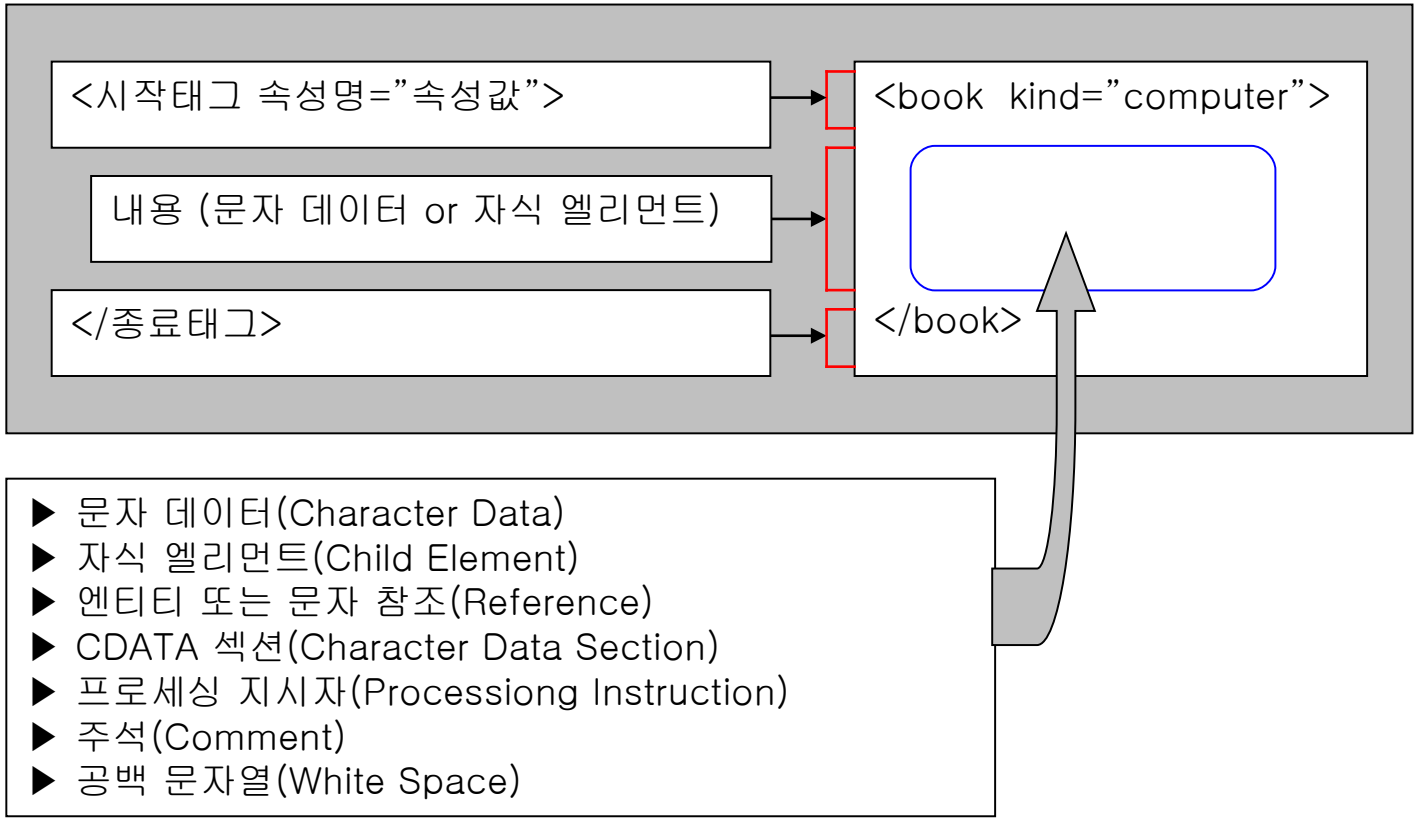
```
<book>
<_book>
<책>
<book1>
<book-1>
<Book>
```

< 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.

7. 엘리먼트 내용

7-1 엘리먼트 내용으로 올 수 있는 것들



7. 엘리먼트 내용

7-2 문자 데이터

- 문자 데이터(Character Data)란 XML 프로세서가 해석할 수 있는 내용 중에서 마크업을 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	&#10진수; &#x16진수;
■ 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”?>

7. 엘리먼트 내용

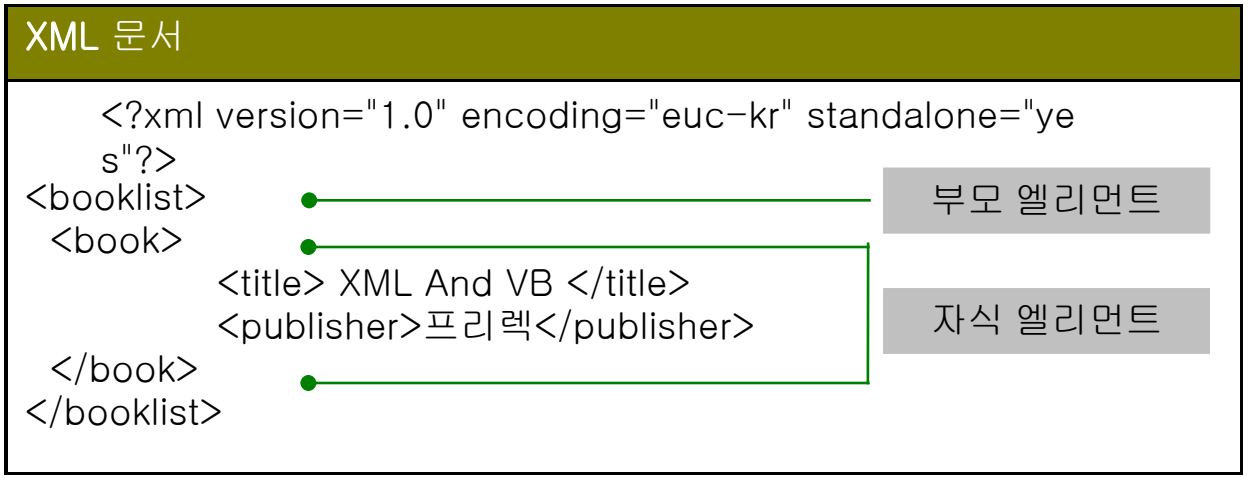
- 문자 데이터 내에는 '&' 문자와 '<' 문자를 사용할 수 없다. '&' 문자는 엔티티 참조의 시작을 의미하며, '<' 문자는 엘리먼트의 태그, CDATA 섹션의 시작을 의미하기 때문이다.
- 빌트인(Built-in) 엔티티의 참조 또는 문자 참조로 사용이 가능하다.

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

7. 엘리먼트 내용

7-3 자식 엘리먼트

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



7-4 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.

7. 엘리먼트 내용

7-5 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	&#x20;	&#32;
탭	&#x9;	&#9;

XML 문서

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

엔화

7. 엘리먼트 내용

7-6 CDATA 섹션

- 대부분의 문자 데이터인 PCDATA(Parsed Character DATA)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 &amp; JAVA ]]>
    </title>
  </book>
</booklist>
```

7. 엘리먼트 내용

7-7 프로세싱 지시자 (Processing Instruction)

- 일반적으로는 엘리먼트의 내용(Content) 부분에 프로세싱 지시자가 오지 않는다. 프로세싱 지시자는 문서 서두(prolog) 부분에 기술되어 응용프로그램이 XML 문서를 처리하기 위한 정보를 전달할 목적으로 사용된다. (9절)

7-8 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.

8. 속성

8-1 속성 작성 문법

- 속성(Attribute)은 엘리먼트에 부가적인 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"/>
```

9. 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 </title>
      <publisher>&pub1;</publisher>
  </book>
  -->
  <book>
    <title> <!-- 알기 쉬운 --> XML</title>
      <publisher>&pub1;</publisher>
  </book>
</booklist>
```

10. 프로세싱 지시자

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

```
<?타겟이름 지시자?>
```

- 타켓(target) 이름은 태그 이름 작성 규칙과 동일하다.
- 타겟은 프로세싱 지시자를 처리하는 응용프로그램을 식별하는 역할을 하며, 지시자는 해당 응용프로그램이 어떻게 문서를 처리하라는 내용에 해당 된다.

XML 문서

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

<?데이터뷰어 디폴트모양:막대 그래프?>
<?웹브라우저 디폴트모양:테이블?>

<data>
  <x>20</x>
  <y>35</y>
</data>
```

XML 문서가 데이터뷰어 및 웹 브라우저에서 사용된다고 가정