

* XML 프로그래밍
: 체계적으로 배우는 XML 원리와 응용

홍성용 지음

누구를 위한 책인가

웹 기초 학습을 마치고 처음으로 XML을 배우려는 컴퓨터 관련학과 학생과 e-비즈니스 관련학과 학생을 위한 책입니다. XML 원천기술을 이해시키는 데 초점을 두고, XML 기본 기술부터 응용 기술까지 차근차근 짚어나가기 때문에 비전공자와 일반인에게도 추천합니다. 다만 3부 예제부터 자바와 데이터베이스를 활용한 XML 응용 프로그래밍을 진행하므로 관련 지식이 있다면 더욱 쉽게 따라올 수 있습니다.

무엇을 다루는가

준비 학습 : XML 시작(1~4장)

XML 학습을 위해 필요한 기본 내용을 다룹니다. XML의 필요성과 적용 분야, 관련 표준, 사용 환경과 개발 환경을 알아본 후 XML 문서의 구조와 기본 문법을 배웁니다. 그리고 XML 전용 편집기를 이용해 직접 XML 문서를 작성해보고, XML 문서 화면 표현에 사용되는 기술인 CSS에 대해서도 살펴봅니다.

기본 학습 : XML 문서 구조 정의와 조작(5~10장)

XML 문서 구조 정의에 사용하는 문법인 DTD와 XML 스키마를 학습하고 실습해봅니다. 그리고 XML을 좀 더 효율적으로 작성하면서 구조적으로 짜임새 있게 만들어 주는 문법인 XPath, XLink, XPointer의 개념과 사용법도 익힙니다. 마지막으로 실전에서 자주 사용되는 기술인 XML 조작을 위한 XSLT, DOM, SAX 프로그래밍 등을 배웁니다.

응용 학습1 : XML 데이터베이스 활용(11~12장)

XML의 실제 활용 목적을 더욱 극대화시키는 기술인 XML 데이터베이스에 대해 알아봅니다. XML과 데이터베이스의 매핑 관계를 알아본 후 상용 DBMS 중 많이 사용하는 MS-SQL Server를 기반으로 XML 데이터를 데이터베이스로 저장하고 활용하는 방법을 익힙니다. 기존의 데이터베이스 시스템 환경을 응용할 수 있는 응용 학습 단계입니다.

응용 학습2 : 웹 서버 구축과 XML 활용(13장)

XML과 JSP 기술을 활용하여 웹 서버를 구축하는 방법에 대해 알아봅니다. 웹 서버의 개념과 웹 서버 구축에 필요한 기술을 알아본 후, 실제로 실습 환경을 구축하고 JSP 프로그램에서 XML을 어떻게 활용하는지 살펴봅니다.

예제소스

실습에 필요한 자료는 아래 주소에서 내려받을 수 있습니다.

<http://www.hanbit.co.kr/exam/4181>

개발도구

책에서 사용하는 개발도구는 무료로 사용할 수 있는 것으로, 인터넷에서 내려받을 수 있습니다.

① XML 에디터(전체 장)

에디트플러스 <http://www.editplus.co.kr/download.html>

XML Spy <http://www.altova.com/download.html>

XML Copy Editor <http://xml-copy-editor.sourceforge.net>

② 데이터베이스 관리 프로그램(11~13장)

MS-SQL Server <http://www.microsoft.com/ko-kr/download/default.aspx>

③ 웹 서버 구축 관련 프로그램(13장)

아파치 톰캣 <http://tomcat.apache.org>

JDBC <http://www.microsoft.com/ko-kr/download/default.aspx>

자바 JDK <http://www.oracle.com>

강의 보조 자료 및 저자 운영 홈페이지

한빛 홈페이지에서 '교수회원'으로 가입하신 분은 인증 후 교수용 강의 보조 자료를 제공받을 수 있습니다. 홈페이지 우측 상단의 <교수회원전용> 아이콘을 클릭해주세요.

<http://www.hanbit.co.kr>

자주 변경되는 내용을 관리하기 위해 저자가 운영하는 홈페이지입니다.

<http://cafe.naver.com/go2web>

Chapter 02

XML 구조와 문법

01 XML 문서 구조

02 XML 구조와 문법

03 네임스페이스의 개념과 사용

04 객체지향과 XML 설계

요약

연습문제

학습목표

- ▶ XML 문서의 구조를 이해하고 학습한다.
- ▶ XML 문서의 제작 방법을 익히고, 학습한다.
- ▶ XML 설계 방법과 문법 구조를 학습하고 실습한다.
- ▶ 네임스페이스의 개념을 이해하고 학습한다.
- ▶ 객체지향과 XML 설계 기법을 학습한다.

Q&A

Q1 정형화의 일반적인 의미는 무엇인가?

정형화는 일정한 형식이나 틀에 맞게 작성한 구조를 말한다. 예를 들면, XML에서 '시작 태그가 있으면 항상 끝 태그가 있어야 한다'와 같은 일정한 형식에 맞춰 작성한 문서를 정형식 문서(well-formed document)라고 한다. 또한 데이터베이스(database)와 같이 정형화된 데이터만 다루는 구조에서는 데이터베이스에 만들어진 스키마 구조의 형식에 맞는 데이터를 정형화한 데이터라고 할 수 있다.

Q2 XML에서 정형식 XML 문서와 유효한 XML 문서의 차이점은 무엇인가?

정형식 XML 문서는 XML 문법에 맞춰 만들어진 문서를 말한다. 이에 비해 유효한 XML 문서는 일정한 규칙을 별도로 만들어서 적용한 XML 문서이다. 따라서 유효한 XML 문서는 모두 정형식 XML 문서라고 할 수 있다. 이러한 규칙을 별도로 정의해서 만든 문서로는 DTD(Document Type Definition) 또는 XML 스키마가 있다.

XML은 정형식 문서 구조로, 일반적인 구조를 살펴보면 다음과 같다.



그림 2-1 XML 문서의 구성

XML 문서는 도입부에 ❶ XML 선언^{XML declaration}이 필요하다. 다음으로 XML 문서의 유효성을 검증하기 위한 ❷ DTD, XML 스키마 선언, 네임스페이스 선언이 올 수 있다. 다만 이 부분은 선택 사항이므로 필요한 경우에 선언할 수 있다. 마지막으로 ❸ XML 태그와 데이터로 이루어진 몸체^{body}가 있다.

XML 문서의 기본 구조를 좀 더 자세히 살펴보면 다음과 같다.

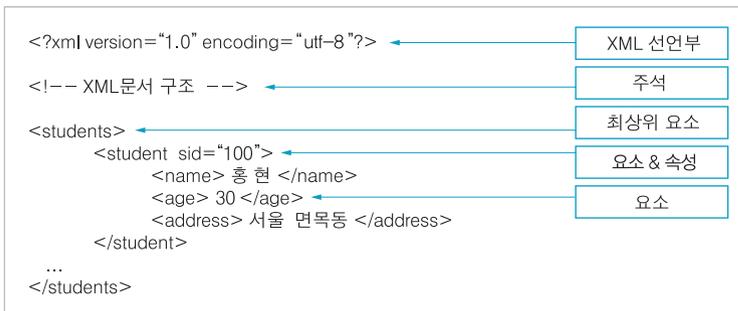


그림 2-2 XML 문서 형태

일반적인 XML 문서는 [그림 2-2]와 같다. XML 선언 이후에는 필요한 부분에 주석을 넣을 수 있다. 몸체는 기본적으로 요소와 속성으로 이루어지며, 요소는 다시 최상위 요소와 하위

요소로 나뉜다. 요소는 태그(< >)로 둘러싼 데이터를 말하고, 속성은 요소에 추가적인 데이터를 나타내는 부분으로, 요소 한 개에 속성 여러 개를 나타낼 수 있다. XML 문서는 요소나 속성에 따라 데이터를 나타낸다. 다음은 XML 문서를 작성해서 웹 브라우저로 출력한 화면이다.



그림 2-3 브라우저로 출력한 XML 문서

XML 문서는 최상위 요소를 하나만 가질 수 있지만, 하위 요소는 여러 개의 계층 구조로 가질 수 있다. 최상위 요소에 속한 하위 요소 중 하나의 그룹을 인스턴스라고 한다. [그림 2-3]에서 최상위 요소는 <students>이고, 인스턴스는 세 개가 된다.

1 XML 선언부

XML 문서임을 선언하는 부분이다. 선언부는 반드시 입력해야 하며, 속성으로 버전과 인코딩을 지정한다. XML 선언부는 반드시 시작과 끝에 <? ~ ?> 태그를 붙여야 한다. 문서 분석 애플리케이션은 처리 명령문으로 이것이 XML 문서임을 판단한다.

```

<?xml version="버전" encoding="문자 코드" standalone="yes or no"?>
    ①         ②         ③         ④
    
```

TIP W3C에서는 파서나 XML 처리기가 XML 문서를 제대로 처리할 수 있게 XML 선언문을 반드시 포함할 것을 권장한다.

- ① <?xml로 시작해야 하고, 공백이 없어야 한다. ② 버전은 필수 속성이고, ③ 인코딩과 ④ 독

립형 여부는 선택 속성이다.

▼ 사용 예

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

저자 한마디 XML 문서 구조

W3C에서 정의한 XML 문서 구성은 'document ::= prolog element Misc*'이다. 여기서 프롤로그(prolog)는 파서가 XML 문서를 어떻게 해석하고 파싱할지 알려주는 부분이다. 요소(element)는 데이터를 기술하는 부분이며, 속성 여러 개를 표현할 수 있다. Misc는 Miscellaneous의 약자로, 주석이나 처리 명령어(PI, Processing Instructions) 등이 있음을 정의한다.

XML 선언에서는 버전을 가장 먼저 표기해야 하고, 생략할 수는 없다. 다음으로 인코딩 정보와 독립성 여부를 지정한다. 인코딩 기본값은 UTF-8(8비트 인코딩)을 사용하지만, 한글을 사용하려면 euc-kr(16비트 인코딩)로 지정해야 한다. 독립성 여부는 XML 문서를 단독으로 사용할 수 있는지, 문서 외부에 어떠한 마크업 선언이 있어야만 애플리케이션으로 전달하는 데이터가 만들어질 수 있는지를 나타낸다. 독립성 여부는 생략이 가능하고 기본값은 yes이다.

Tip XML 선언에서 xml, version, encoding은 소문자로 표기하고, 순서대로 입력한다.

Tip 인코딩에 대한 내용은 <http://www.w3.org/International>을 참고하기 바란다.

잘못된 XML 선언문을 살펴보면서, XML 선언문 사용 예를 익혀보자.

■ XML 선언에서 시작 태그인 <?와 xml 문자 사이에는 공백이 없어야 한다

▼ students_error_1.xml

```
<? xml version="1.0" encoding="utf-8"?> <?와 xml 사이에 공백이 있는 오류
↑
<!-- XML 문서 구조 : 선언부 공백 오류 -->

<students>
  <student sid="100">
    <name>홍 현</name>
    <age>30</age>
    <address>서울 면목동</address>
  </student>
</students>
```

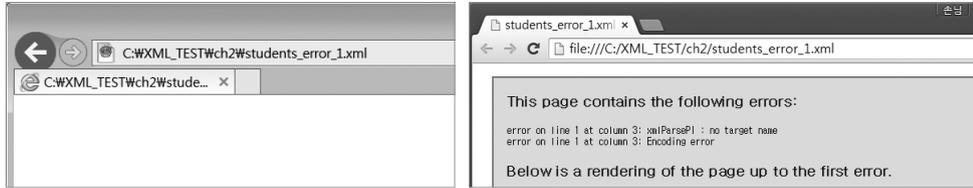


그림 2-4 오류 화면 1

TIP 웹 브라우저마다 오류를 표시하는 방법이 다르다. 인터넷 익스플로러 11 버전은 오류 메시지를 별도로 표현하지 않고 있으며 크롬은 오류 메시지를 출력하여 나타내고 있다. 자세한 오류 표현은 각 브라우저에서 실행하여 결과를 살펴보길 바란다.

저자 한마디 XML 문자 코드

문자를 표현하기 위해 각 문자에 붙인 번호이다. XML 규격에서 정한 문자 코드 체계는 UTF-8, UTF-16, ISO-10646-UCS 등이 있다. XML 규격에서 정하고 있는 필수 문자 코드는 UTF-8, UTF-16이지만 생략할 수 있다. 한글을 사용하려면 euc-kr이나 KSC5601을 지정한다. 이 책에서는 UTF-8을 사용할 것이다.

■ XML 선언에 version 속성은 반드시 표기해야 한다

▼ students_error_2.xml

```
<?xml encoding="utf-8"?> ← version 속성이 빠져서 오류
<!-- XML 문서 구조 : 버전 속성 생략 오류 -->

<students>
  <student sid="100">
    <name>홍 현</name>
    <age>30</age>
    <address>서울 면목동</address>
  </student>
</students>
```

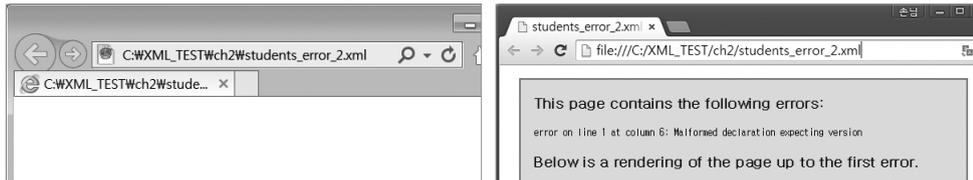


그림 2-5 오류 화면 2

TIP 인터넷 익스플로러는 빈 화면을 출력하고 크롬의 경우 오류 메시지를 출력한다.

■ XML 선언 속성은 version, encoding 순으로 표현해야 한다

▼ students_error_3.xml

```
<?xml encoding="utf-8" version="1.0" standalone="yes"?> ← 속성 표현 순서가 잘못되어 오류
<!-- XML 문서 구조 -->

<students>
  <student sid="100">
    <name>홍 현</name>
    <age>30</age>
    <address>서울 면목동</address>
  </student>
</students>
```

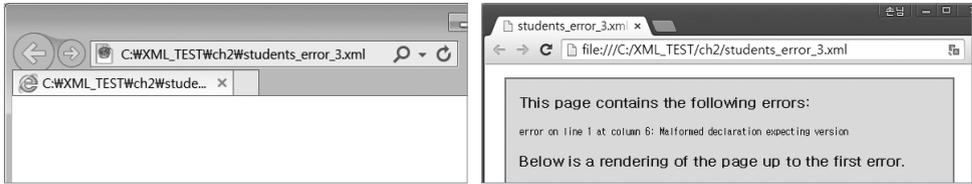


그림 2-6 오류 화면 3

2 XML 주석

HTML과 마찬가지로 XML 문서에서도 주석을 사용할 수 있다. 주석은 `<!-- ... -->` 형태로 표기하며, XML 선언부 이후라면 어느 위치에나 올 수 있다.

```
<!-- comment -->
```

▼ 사용 예

```
<!-- 이곳은 주석을 표현하는 부분입니다. -->
```

XML 주석은 DTD나 XML 스키마와 같은 문서 정의 문서에서도 사용할 수 있다. 주석은 XML 파서가 제거해서 애플리케이션에 전달할 수도 있고, 제거하지 않고 전달할 수도 있다.

주석은 복잡한 프로그램이나 공동 작업으로 만들어진 프로그램에서 유용하게 사용할 수 있다. XML 주석의 여러 가지 사례를 통해 문서의 잘못된 표현을 살펴보자.

```

<!-- 잘못된 주석 ❶ -->
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<students <!-- 잘못된 주석 ❷ -->>
<!-- 잘못된 -- 주석 ❸ -->
    
```

- ❶ XML 주석은 XML 선언 이전에 표현할 수 없다.
- ❷ XML 주석은 다른 요소의 내부에 표현할 수 없다.
- ❸ XML 주석은 내부에 ‘--’ 문자를 사용할 수 없다.

3 XML 요소

요소는 XML 문서 내에 데이터를 표현하기 위한 태그 정의와 데이터를 말한다.

3.1 XML 요소 서식

XML에서는 시작 태그(<>)로 시작해서 종료 태그(</>)로 끝내야 하며, 시작 태그와 종료 태그 사이에는 요소 콘텐츠를 표기한다. 이러한 요소의 내용 부분을 PCDATA^{Parsed Character DATA}라고 한다. 요소 구조를 보면 다음과 같다.

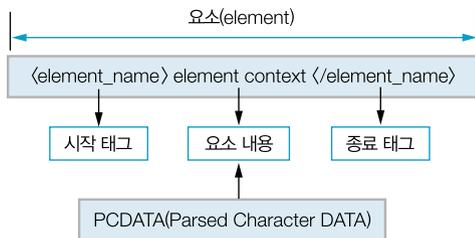


그림 2-7 요소의 구조

3.2 요소 정의와 사용법

요소를 정의하고 사용할 때 주의해야 할 사항을 알아보자.

- 시작 태그 다음에는 반드시 종료 태그가 있어야 한다. 이때 요소명은 시작 태그와 종료 태그가 항상 일치해야 한다.

```
<student> 데이터, 요소 등 </student>
```

- 모든 요소명은 대소문자를 구분하기 때문에 시작 태그와 종료 태그는 대문자를 반드시 일치시켜야 한다.

```
<STUDENT>           </student> (X)  
<STUDENT>           </STUDENT> (O)
```

- 요소와 텍스트 데이터를 함께(mixed) 나타낼 수 있다. 예를 들어, 주소를 나타내기 위해 '서울'이라는 텍스트 부분과 '<dong>'이라는 요소를 함께 표현할 수 있다.

```
<address> 서울  
    <dong> 면목동 </dong>  
</address>
```

- 요소 내용이 비어 있을 때는 요소 내용에 아무 것도 쓰지 않거나 시작 태그와 종료 태그를 통합해서 표현할 수 있다.

```
<telephone>     </telephone> (O)  
<telephone/>           (O)
```

- 태그를 겹쳐 쓰는(overlap) 것을 허용하지 않는다. 태그를 겹쳐 쓰면 XML 파서가 태그의 시작과 끝을 인식할 수 없기 때문에 오류가 발생한다. 따라서 요소 내용에 요소를 표현하는 내부 형태로 작성해야 한다.

```
<student>  
  <name> 홍 현 <age> </name> ] (X)  
                        30 </age>  
</student>
```

- XML 문서는 최상위 요소를 하나만 가질 수 있다. 최상위 요소는 루트 요소 또는 문서 요소라고도 하는데, XML 문서의 최상위 도메인(domain)으로 구성한다.

잘못된 문서 형식

```
<students>
  <student> ... </student>
</students>
<students>
  <student> ... </student>
</students>
```

올바른 문서 형식

```
<students>
  <student> ... </student>
<student> ... </student>
  <student> ... </student>
  <student> ... </student>
</students>
```

- 요소명이 XML 이름 부여 규칙(naming convention)에 적합해야 한다.

잘못된 문서 형식

```
<3student>
  <na me> 홍 현 </na me>
</3student>
```

올바른 문서 형식

```
<student3>
  <name> 홍 현 </name>
</student3>
```

HTML에서는 태그의 요소명이 정해져 있는 반면, XML에서는 태그의 요소명을 직접 정의해서 사용한다. XML의 가장 큰 장점이자 매력이라고 볼 수 있다. 요소명은 보통 요소의 내용이 무엇인가에 따라 의미 명칭을 나타내게 되며 영문 혹은 한글, 한자로 나타낼 수 있다.

3.3 XML 이름 부여 규칙

XML 이름 부여 규칙을 살펴보면 다음과 같다.

- 문자열을 'xml'로 시작할 수 없다(단, 대문자나 대문자를 포함하면 사용할 수 있다).
- 요소명의 첫 문자는 문자나 '_' 기호로 시작해야 하고 숫자로 시작할 수 없다.
- 요소명의 처음과 중간에는 공백을 사용할 수 없다(단, 공백 문자로 끝날 수는 있다).
- 예약어를 사용할 수 없다.

저자 한마디 요소명 정하기

XML의 가장 큰 장점은 문서 내에서 태그를 자유롭게 정의할 수 있다는 것이다. 그러나 자유롭게 정의할 수 있다는 것은 그만큼 잘못되거나 무의미하게 정의할 수 있는 여지가 많다는 의미이다. 예를 들면, 자신이 만든 문서라고 해서 '이름'을 표현할 때 <n>으로 표현하거나 '주소'를 <a>로 표현하면 다른 사람이 읽었을 때 무엇을 의미하는지 알 수 없다. 그러므로, 되도록 요소명은 데이터에 의미성을 부여하여 결정하는 것이 좋다. 이러한 이유로 XML 문서를 공유하려면 표준화(standardization)가 필요하고, 문서 구조 정의(DTD)가 필요하다. 또한, 동일한 요소명을 구별할 수 있게 네임스페이스(namespace) 개념을 사용해야 한다.

다음은 요소명을 나타내는 예이다.

▼ 자동차의 명칭을 나타내기 위한 요소명

```
<car_name> 산타페 2.0 </car_name> ← 영문일 때  
<차명칭> 산타페 2.0 </차명칭> ← 한글일 때
```

▼ 주소를 나타내기 위한 요소명

```
<address> 서울 용산구 한남동 </address> ← 영문일 때  
<주소> 서울 용산구 한남동 </주소> ← 한글일 때
```

▼ 전화번호를 나타내기 위한 요소명

```
<telephone> 02-123-6639 </telephone> ← 영문일 때  
<전화번호> 02-123-6639 </전화번호> ← 한글일 때
```

3.4 요소 내용 구성

요소 내용은 여러 개를 나타낼 수 있다. 즉, 요소 내용 안에 요소를 가질 수 있다는 의미이다. 이러한 구성은 한 XML 문서에 서로 다른 구조의 요소를 분류하여 나타낼 수 있을 뿐만 아니라 같은 구조의 요소를 여러 번 반복해서 나타낼 수도 있다. 다음 예제를 보면서 좀 더 자세히 살펴보자.

▼ 같은 구조의 요소를 반복해서 나타내고 있는 XML 문서

```
<students>  
  <student>  
    <sid> 100 </sid>  
    <name> 홍현 </name>  
    <age> 30 </age>  
    <address> 서울 면목동 </address>  
  </student>  
  <student>  
    <sid> 200 </sid>  
    <name> 오석영 </name>  
    <age> 27 </age>  
    <address> 서울 옥수동 </address>  
  </student>  
</students>
```

<student> 요소에 여러 개의 요소를 포함

같은 구조의 요소임

<student> 요소에 여러 개의 요소를 포함

▼ 다른 구조의 요소를 나타내고 있는 XML 문서

```

<students>
  <student>
    <sid>100 </sid>
    <name>홍현 </name>
    <age>30 </age>
    <address>서울 면목동 </address>
  </student>
  <course>
    <cno>c123 </cno>
    <cname>XML 프로그래밍 </cname>
    <credit>3 </credit>
  </course>
</students>

```

<student> 요소에 여러 개의 요소를 포함
 <course> 요소에 여러 개의 요소를 포함
 다른 구조의 요소임 →

내용에 대한 요소 포함 관계는 계층 형태의 트리 구조로 나타낼 수 있다. 실제로 XML 문서를 설계하려면 계층 구조의 트리로 표현하는 것이 가장 바람직하다. 앞에서 본 XML 문서의 구조를 트리 구조로 표현하면 [그림 2-8], [그림 2-9]와 같다.

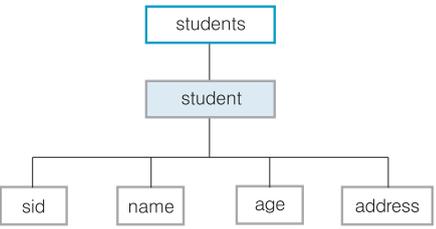


그림 2-8 요소 내용에 같은 구조의 요소를 포함하는 계층 구조

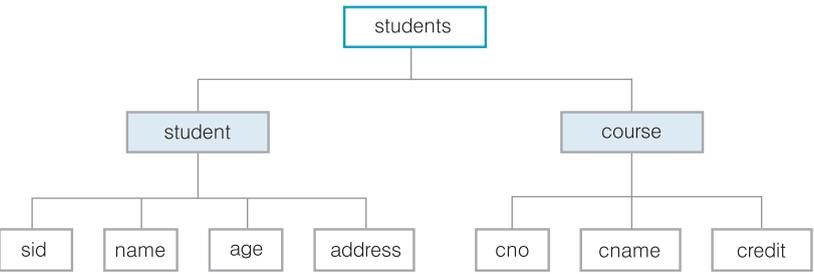


그림 2-9 요소 내용에 다른 구조의 요소를 포함하는 계층 구조

XML 문서 구조를 트리 구조로 만들 수 없다면 그 XML 문서는 잘못 만들어진 것이다. 일반적

으로 [그림 2-9]와 같은 트리는 구조적으로 문제가 없지만 실제 설계 구조로는 좋지 않다. 포함 관계로 볼 때 한 학생(student)이 어떤 과목(course)을 수강하고 있는지 나타내려면, 과목 정보를 학생 정보 안에 포함시키는 것이 좋다. 그렇게 표현하면 여러 학생에 대한 인스턴스를 나타낼 때 쉽게 알아볼 수 있고, 검색할 때도 용이하게 접근할 수 있다. 다음 두 구조를 살펴보자.

(a) 학생에 과목 정보를 포함한 XML 문서

```

<students>
  <student>
    <sid> 100 </sid>
    <name> 홍현 </name>
    <age> 30 </age>
    <address> 서울 면목동 </address>
    <course>
      <cno> c123 </cno>
      <cname> XML 프로그래밍 </cname>
      <credit> 3 </credit>
    </course>
  </student>

  <student>
    <sid> 200 </sid>
    <name> 오석영 </name>
    <age> 27 </age>
    <address> 서울 옥수동 </address>
    <course>
      <cno> c339 </cno>
      <cname> 자료구조 </cname>
      <credit> 3 </credit>
    </course>
  </student>
</students>

```

(b) 학생 정보와 과목 정보를 분리한 XML 문서

```

<students>
  <student>
    <sid> 100 </sid>
    <name> 홍현 </name>
    <age> 30 </age>
    <address> 서울 면목동 </address>
  </student>
  <course>
    <cno> c123 </cno>
    <cname> XML 프로그래밍 </cname>
    <credit> 3 </credit>
  </course>

  <student>
    <sid> 200 </sid>
    <name> 오석영 </name>
    <age> 27 </age>
    <address> 서울 옥수동 </address>
  </student>
  <course>
    <cno> c339 </cno>
    <cname> 자료구조 </cname>
    <credit> 3 </credit>
  </course>
</students>

```

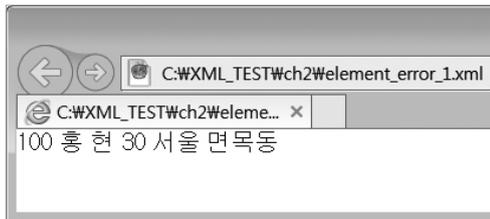
두 문서 모두 학생들의 정보를 다루기 위한 XML 문서이다. 그러나 (b)는 (a)보다 문서의 구조가 좋지 않다. (b)는 ‘홍현’이 어떤 과목을 신청했는지 알 수 없다. 물론, 과목에 대한 부분을 별도의 XML 문서로 만들 수도 있고, 학생 정보 안에 <cno> 요소를 포함시켜 관계성을 유지할 수도 있다. 문서의 구조 설계는 수없이 많은 형태로 만들어질 수 있다. 그러나 얼마나 유용하고, 효율적이며 사용 목적에 적합하게 문서 구조를 작성했는지가 무척 중요하다는 사실을 잊지 말아야 한다.

XML 요소의 여러 가지 사례를 통해 잘못된 문서 표현을 살펴보자.

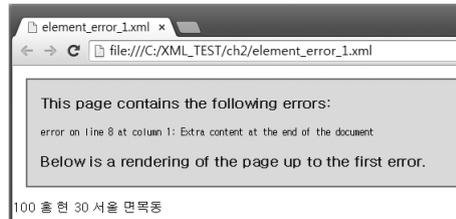
■ XML 문서의 본체에는 반드시 한 개의 최상위 요소가 있어야만 한다

▼ element_error_1.xml

```
<?xml version="1.0" encoding="utf-8"?>
<student>
  <sid>100 </sid>
  <name>홍 현 </name>
  <age>30 </age>
  <address>서울 면목동 </address>
</student>
<student>
  <sid>200 </sid>
  <name>오석영 </name>
  <age>27 </age>
  <address>서울 옥수동 </address>
</student>
```



(a) IE : 일부 데이터만 출력



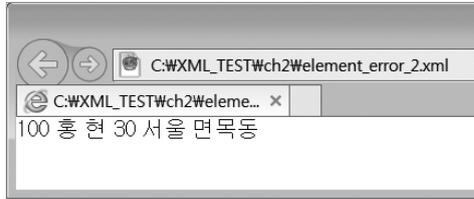
(b) 크롬 : 오류 메시지와 일부 데이터만 출력

그림 2-10 오류 화면 4

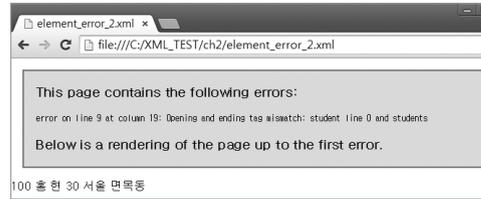
■ 요소를 나타내는 시작 태그와 종료 태그는 반드시 한 쌍(pair)으로 존재해야 한다

▼ element_error_2.xml

```
<?xml version="1.0" encoding="utf-8"?>
<students>
  <student>
    <sid>100 </sid>
    <name>홍 현 </name>
    <age>30 </age>
    <address>서울 면목동 </address>
  </students> ← <student>의 종료 태그가 없음
```



(a) IE : 데이터 출력, 그러나 XML 문서 구조의 내부 오류



(b) 크롬 : 오류 메시지와 데이터만 출력

그림 2-11 오류 화면 5

■ 다른 요소의 시작 태그나 종료 태그를 겹쳐 쓰지(overlap) 말아야 하며, 이름 부여 규칙(naming convention)에 적합해야 한다

▼ element_error_3.xml

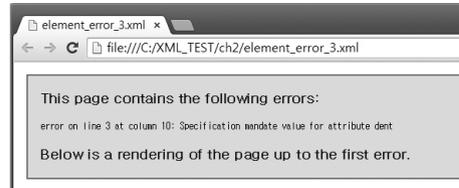
```

<?xml version="1.0" encoding="utf-8" ?>
<students>
  <stu dent> ← 요소명 사이에 공백이 있음
    <sid>100 </sid>
    <name>홍 현 </name>
    <age>30 <address></age> ← 요소의 시작 태그와 종료 태그를 겹쳐 씀
    서울 면목동 </address>
  </stu dent>
</students>

```



(a) IE : 데이터 출력 안됨



(b) 크롬 : 오류 메시지만 출력

그림 2-12 오류 화면 6

TIP 브라우저마다 오류 표현 방식이 약간씩 다르므로, 오류 출력 결과를 정확하게 알기 위해서는 브라우저마다 XML 오류 문서를 직접 실행하여 보길 권장한다.

4 XML 속성

속성은 요소 내에 정보나 데이터를 표현하기 위한 방법이다. XML 문서에서는 한 요소에 대하여 추가적으로 여러 속성을 표현할 수 있으며, 속성은 시작 태그의 일부로 표현한다. 속성은 속성명과 속성값으로 이루어지며 여러 속성을 표현할 경우에는 속성과 속성 사이를 공백space으로 구분한다.

```
<element_name attribute="attribute_value" attribute="attribute_value" ... >
```

XML 문서의 속성 구조는 다음과 같다.

```
<student sid="100" >
  ↑      ↑      ↑
  요소명 속성명 속성값
```

4.1 속성 선언 시 주의 사항

속성을 정의하고 사용할 때 알아야 할 중요한 사항을 몇 가지 정리해보자.

- 속성은 반드시 값을 가져야 한다. 물론, 빈 문자열을 포함할 수도 있다.

```
<student sid="100" telephone=" " > 홍 현 </student>
```

- 속성값은 큰 따옴표(")나 작은 따옴표(')로 감싸야 한다. 속성값을 나타내기 위해서 큰 따옴표(")와 작은 따옴표(')를 동시에 사용할 수는 있지만 겹쳐서 사용해서는 안 된다.

```
<student sid='100' telephone="02-123-6399"> 홍 현 </student> (○)
<student address="서울 '면목동' 00번지"> 홍 현 </student> (○)
```

- 속성명 부여 방법은 요소명 부여 방법과 같다. 즉, 대소문자를 구별하며, 'xml'이라는 문자열로 시작할 수 없고, 숫자로 속성명을 시작할 수 없다.

```
<student sid="100" Sid="100" SID="100"> 홍 현 </student>
```

- 한 요소에 같은 이름의 속성을 두 개 이상 선언할 수 없다.

```
<student name="홍현" name="오석영" name="황규정"></student>    (×)
```

속성은 한 요소 내에서 여러 가지로 표현할 수 있다. 그러면 요소로 표현하는 것과 속성으로 표현하는 것에 어떤 차이가 있을까? 둘 다 데이터를 표현한다는 목적은 동일하지만 실제 문서를 설계하고 제작할 때는 중요한 차이점이 있다.

먼저 속성으로 설계할 경우, 문서 크기를 줄일 수 있다는 장점이 있지만, 속성 자체가 요소에 종속적이기 때문에 확장성이 떨어지는 단점이 있다. 이에 비해 요소로 설계할 경우, 가독성이 좋아 의미를 쉽게 파악할 수 있으며, 이는 곧 향후에 문서 내용을 추가하거나 확장할 때 용이하다는 장점을 끌어낸다.

요소와 속성 표현을 비교해서 전환할 수 있는 경우와 그렇지 않은 경우를 살펴보면 차이점을 좀더 구체적으로 살펴보자. 다음과 같이 한 학생에 대한 정보를 표현하기 위해 학번, 이름, 나이, 주소 등을 표현하는 것은 문제가 없다.

```
<student>
  <sid>100 </sid>
  <name>홍현 </name>
  <age>30 </age>
  <address>서울 면목동 </address>
</student>
```

```
<student sid="100"
  name="홍현"
  age="30"
  address="서울 면목동">
</student>
```

그러나 다음과 같이 학생의 키나 전화번호 정보를 표현하기 위한 상황을 생각해보자.

```
<student height="170"> 황규정
</student>
```

↓inch로 표현 어려움

```
<student> 황규정
  <height unit="cm">170 </height>
</student>
```

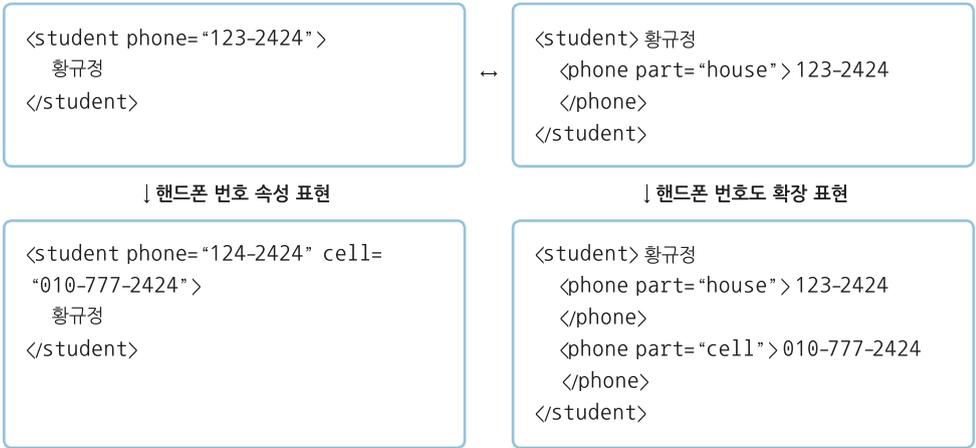
↓inch로도 확장 표현

```
<student height="170cm"> 황규정
</student>
<student height="66.9inch"> 황규정
</student>
```

```
<student> 황규정
  <height unit="cm">170 </height>
  <height unit="inch">66.9 </height>
</student>
```

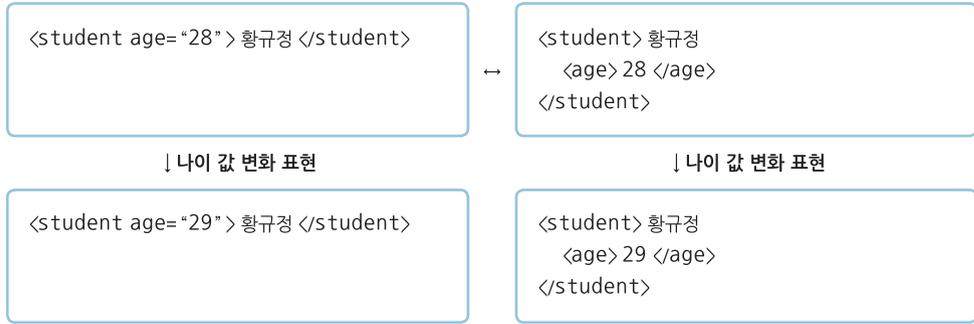
학생의 키 정보를 나타내기 위해 속성을 사용했다면, 한 학생의 키 정보를 하나의 단위로 밖에 표현하지 못할 것이다. 그러나 키에 대한 정보를 요소로 표현하고, 단위를 속성으로 표현한다면, 단위에 따른 키 정보를 다양하게 표현할 수 있다. 물론, 속성명을 다르게 설계하여 표현할 수도 있지만(예를 들어, height_cm="170" height_inch="66.9"), 일반적으로 요소로 표현하는 것이 좋다. 요소로 표현하면 속성으로 표현하는 것에 비해 확장하기 편리하다.

다른 예를 살펴보자. 학생 전화번호를 표현하기 위해 속성을 사용한다면, 하나의 전화번호를 표현하기에는 좋다. 하지만 다른 전화번호 즉, 핸드폰이나 아르바이트, 직장 전화번호를 표현해야 할 경우, 확장성을 고려해서 요소로 표현하는 것이 좋다.



TIP 요소는 확장성이 용이하고 속성은 단일값을 표현하기에 적합하다.

그렇다면 어떤 경우 속성으로 표현하는 것이 좋을까? 표현하고자 하는 데이터가 단순하고, 의존적인 경우 속성으로 표현하는 것이 좋다. 즉 하나의 데이터 값만 표현하고자 하는 경우라면 굳이 요소로 표현할 필요가 없을 것이다. 예를 들면, 예 혹은 아니오로 표현하는 경우, 나이를 표현하는 경우가 이에 해당한다.

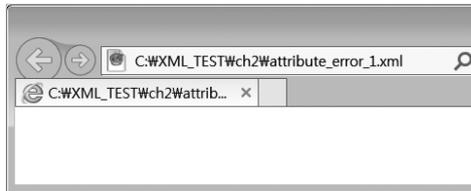


XML 속성에 의한 여러 가지 사례를 통해 잘못된 문서의 표현을 살펴보자.

■ 큰 따옴표와 작은 따옴표를 겹쳐서 사용할 수 없다

▼ attribute_error_1.xml

```
<?xml version="1.0" encoding="utf-8"?>
<students>
  <student
    sid="100"
    name="홍현"
    age="30"
    address="서울 '면목동' 00번지">
  </student>
</students>
```



(a) IE : 데이터 출력 안됨



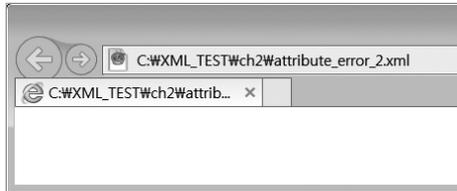
(b) 크롬 : 오류 메시지만 출력

그림 2-13 오류 화면 7

■ 여러 개의 속성을 표현할 때는 쉼표(,)를 사용하지 않는다

▼ attribute_error_2.xml

```
<?xml version="1.0" encoding="utf-8"?>
<students>
  <student
    sid="100",
    age="30",
    address="서울 면목동" >
  </student>
</students>
```



(a) IE : 데이터 출력 안됨



(b) 크롬 : 오류 메시지만 출력

그림 2-14 오류 화면 8