

1. XML의 정의

■ ‘eXtensible Markup Language’의 약자

■ Extensible : 확장가능하다는 의미

여기서의 확장가능은 markup이 확장가능하다는 뜻이 된다. 즉 우리가 어떤 실제 데이터를 표현하고자 할 때 이 실제 데이터에 특정 정보를 덧붙일 때 어떤 기호나 표시를 미리 정해진 범위 내에서 사용하는 것이 아니라 필요에 따라 이를 사용자가 새롭게 정의하는 것이 가능하다는 것을 의미한다. (여기서 새롭게 정의한다는 것은 사용자가 정말 아무런 규칙없이 정의하는 것이 아니라 정해진 규칙 안에서 사용자가 새롭게 정의할 수 있다는 뜻이다.)

■ Markup : 조판지시라는 뜻

인쇄업계에서 출판할 원고의 특정 부분을 어떻게 출력할 것인가를 인쇄업자에게 지시하는 내용을 편집자가 써 넣은 것은 의미하는 용어로 사용되었다. 즉 출력하고자 하는 데이터에 어떤 정보를 덧붙이고자 할 때 이를 표시하기 위한 방법이며 이런 부가 정보를 우리는 메타 데이터(meta data)라고도 한다.

■ Language : 언어

컴퓨터에서 사용하는 언어라는 뜻인데 컴퓨터 분야에서 보통 언어라고 하며 프로그래밍 언어를 뜻하지만 XML에서의 언어는 프로그래밍을 위한 언어는 아니다. XML에서의 언어는 어떤 정보를 표현하기 위한 수단을 의미한다.

■ 어떤 정보를 표현하는데 있어 필요한 부가 정보를 함께 기술하기 위한 방법을 사용자가 새롭게 정의하여 사용할 수 있는 프로그래밍 언어

2. 문서의 기본구조

- 내용 : 실제 본문내용(데이터)
- 구조 : 문서의 구성 정보(제목, 장, 절, 각주, 문단 등)
- 스타일 : 문서를 표현하기 위한 정보(글꼴, 글자색, 크기, 여백 등)

3. XML의 발전 과정

■ SGML(Standard Generalized Markup Language)

- ① 기술적 문서 마크업을 생성하는 기법을 제공함
- ② 다양한 입력으로부터 내용, 구조를 표준화하기 위한 방법 제공
- ③ 시스템 독립적 / 논리적인 정보만을 제공하기 때문에 포맷정보와 내용을 분리함
- ④ 많은 장점을 가지고 있지만, 구성과 문법이 복잡하여 사용하기 힘들
- ⑤ 쉽게 사용하기 위해 만들어진 마크업 언어가 HTML임

■ HTML(Hyper Text Markup Language)

- ① 웹에서 정보를 표현할 목적으로 제안한 마크업 언어
- ② SGML 태그중 일부를 미리 정의하여 사용하기 쉽게 만들
- ③ 인터넷의 발달로 가장 많이 쓰이는 마크업 언어

■ HTML의 단점

- ① 사용할 수 있는 태그가 한정적임 : 유연성과 확장성이 떨어짐
- ② 정보를 표현하는데 집중 : 태그 내용이 어떤 의미인지 알 수 없음
- ③ 구조화된 정보를 표현하고 검색하기 어려움 : 정보간 연관성을 표현할 수 없음
- ④ 문서의 유효성을 검증하기 어려움
- ⑤ 전체 문서만 표현이 가능함(일부만 표현하기 어려움) : 문서의 조작이 어려움

■ XML의 탄생

- ① SGML과 HTML의 문제점을 해결할 수 있는 차세대 마크업 언어
- ② SGML로부터 정보의 구조적인 기술방법과 정보내용과 표시정보를 분리한 구조를 차용함
- ③ HTML로부터 인터넷의 이용 환경을 이어 받음

4. XML의 기본 개념

■ 구조적인 데이터로 표현하여 작성함

문서의 기본구조 내용 : 실제 본문내용(데이터) 구조 : 문서의 구성 정보(제목, 장, 절, 각주, 문단 등) 스타일 : 문서를 표현하기 위한 정보(글꼴, 글자색, 크기, 여백 등) XML의 발전 과정 SGML(Standard Generalized Markup Language) 기술적 문서 마크업을 생성하는 기법을 제공함 다양한 입력으로부터 내용, 구조를 표준화하기 위한 방법 제공 시스템 독립적 / 논리적인 정보만을 제공하기 때문에 포맷정보와 내용을 분리함 많은 장점을 가지고 있지만, 구성과 문법이 복잡하여 사용하기 힘들 쉽게 사용하기 위해 만들어진 마크업 언어가 HTML

1. 문서의 기본구조

- 내용 : 실제 본문내용(데이터)
- 구조 : 문서의 구성 정보(제목, 장, 절, 각주, 문단 등)
- 스타일 : 문서를 표현하기 위한 정보(글꼴, 글자색, 크기, 여백 등)

2. XML의 발전 과정

- SGML(Standard Generalized Markup Language)
 - ① 기술적 문서 마크업을 생성하는 기법을 제공함
 - ② 다양한 입력으로부터 내용, 구조를 표준화하기 위한 방법 제공
 - ③ 시스템 독립적 / 논리적인 정보만을 제공하기 때문에 포맷정보와 내용을 분리함
 - ④ 많은 장점을 가지고 있지만, 구성과 문법이 복잡하여 사용하기 힘들
 - ⑤ 쉽게 사용하기 위해 만들어진 마크업 언어가 HTML임

■ HTML과 비슷해 보이지만 다름

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
  </head>
  <body>
    <h1>Hello World!</h1>
  </body>
</html>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE List SYSTEM 'dtdEx.dtd'>
<List>
  <personal number="100">
    <name>마루치</name>
    <c_phone>010</c_phone>
    <email>maru@&hb;.co.kr</email>
  </personal>
</List>
```

■ 텍스트이지만 읽히는 것만을 목적으로 하지 않음

■ 확장할 수 있음

- ① XML(eXtensible Markup Language) : extensible의 뜻이 ‘확장가능한’
- ② 문서의 데이터도 계속 늘어날 수 있고, 사용자가 태그를 직접 정의해서 사용함

■ 기술의 집합

- ① Xlink, XPointer, XFragments, CSS, XSL, XSLT 등 기술의 집합
- ② 이 외에도 보안, 저장기술 등 다양한 분야의 기술과 접목함

■ 새로운 기술이 아니라 발전된 기술

SGML과 HTML의 장점을 발전시켜 만든 마크업 언어

■ XML은 HTML을 XHTML로 이끌었음

- ① 기존 HTML이 XML의 영향을 받아 변형한 것이 XHTML
- ② XHTML은 확장형 HTML임

■ XML은 모듈

다른 포맷을 조합하거나 재사용함으로써 새로운 문서 포맷을 정의하는 것을 허용함

■ XML은 RDF와 시맨틱 웹의 토대

- ① 시맨틱 웹이란?

Semantic Web은 웹상에 존재하는 여러 가지 자원(웹 문서, 각종 파일들, 서비스 등)간에 관계나 의미의 정보를 컴퓨터가 스스로 처리할 수 있는 형태(온톨로지)로 표현하고 자동화된 기계(컴퓨터)가 처리하도록 하는 프레임워크이자 기술이다.

온톨로지(Ontology)란 사람들이 사물에 대해 생각하는 바를 추상화하고 공유한 모델로 정형화되어 있고, 개념의 타입이나 사용상의 제약조건들이 명시적으로 정의된 기술을 의미한다.

- ② RDF란?

Resource Description Framework(자원묘사 프레임워크)의 약자로 XML의 문제를 해결하기 위해 나온 기술로, XML은 사람이 이해하기는 어렵지 않으나, 기계(컴퓨터)가 이해하는 데 무리가 있다. 즉, 태그의 사용법은 정의 내릴 수 있으나, 컴퓨터가 스스로 태그간의 연관성을 추측할 수는 없다. 그렇기 때문에 RDF를 이용하면 컴퓨터가 문서의 태그간의 의미와 문서의 의미를 표시할 수 있고, 자동화 처리가 한결 쉬워진다.

RDF는 데이터가 처음 생성되거나 업데이트 될 때 생성되는 정보이며, 검색엔진의 로봇들이 데이터를 찾는 키워드이며, 웹 페이지의 지적 권리를 나타내는 지표이다. 또 웹을 인식하며 돌아다니는 기계들이 정보를 인식하고 교환할 수 있도록 정보 상호 운용성을 제공한다.

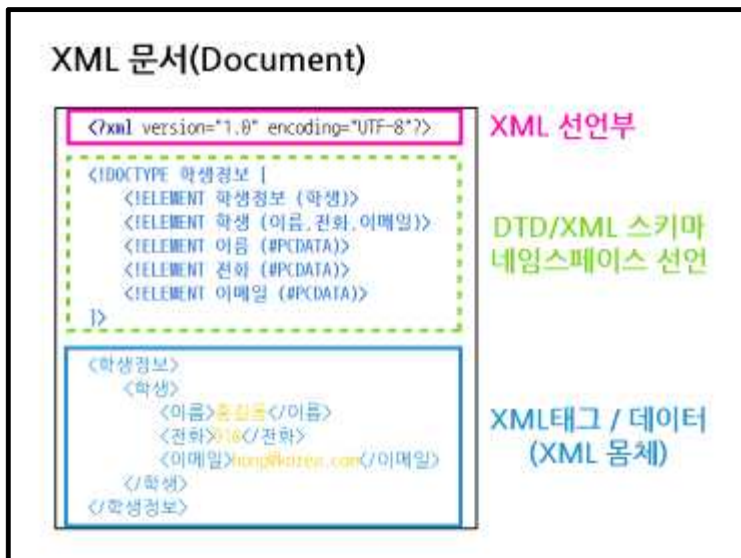
■ 라이선스 제약이 없고, 플랫폼 독립적이며, 많은 자원이 있음

- ① XML은 무료로 개발해서 사용할 수 있음
- ② W3C에서 기술지원을 하고 있기 때문에, 별도의 개발 비용도 없음

5. XML의 활용분야

- 메타데이터 : 데이터에 대한 정보를 가지는 데이터
- 소프트웨어 설계, 배포, 설정
- 멀티미디어 그래픽(png, svg, vml, jsml 등)
- 전자 상거래
- 과학, 교육
- 무선 웹 콘텐츠
- 보안 등

6. XML의 문서 구조



■ XML 선언부

```
<?xml version="버전" [ encoding="문자코드" standalone="yes or no" ]?>
```

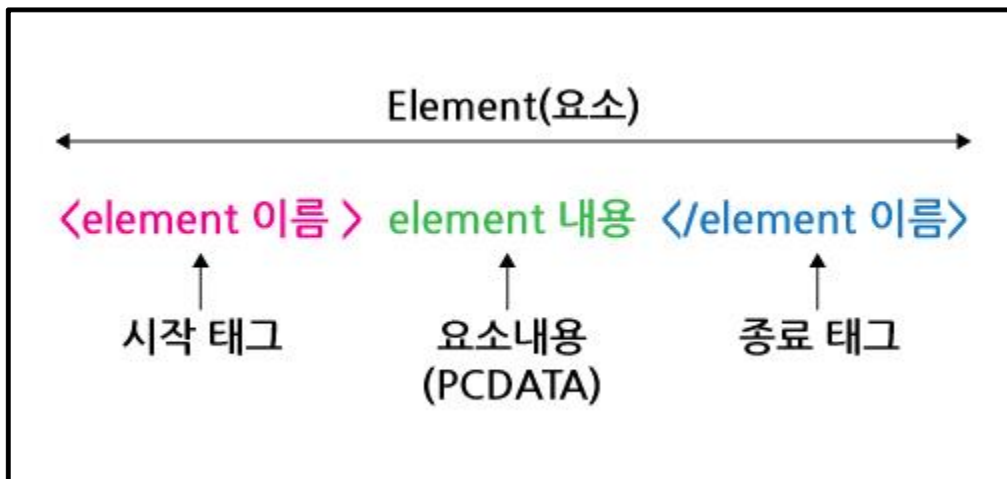
- ① ‘<?xml’ 로 시작하고 공백이 없어야 함
- ② 버전은 필수속성, 인코딩과 독립형 여부는 선택 속성

■ XML 주석

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

■ XML 요소(Element)

- ① XML 문서 내의 데이터를 표현하기 위한 태그 정의와 데이터를 말함



- ② XML몸체에 반드시 한 개의 최상위 요소가 있어야만 함(root element)
③ 시작과 종료태그는 반드시 한 쌍(pair)으로 존재해야 함
④ 다른요소의 시작과 종료태그를 겹치면 안됨

■ XML 속성(Attribute)

- ① 요소 내의 정보나 데이터를 표현하기 위한 방법
② 하나의 요소에 여러개의 속성을 표현할 수 있음
③ 속성은 속성명과 속성값으로 이루어짐

```
<person id = "100">
```

↑ ↑ ↑

요소명 속성명 속성값

7. XML의 문서 제작

■ XML의 종류

- ① Well-formed 문서 : XML의 문법에 맞게 만들어진 오류없는 문서
- ② 유효한(Valid) 문서 : DTD로 정의된 규칙대로 만들어진 검증된 문서

■ Well-formed 문서 예제

```
XMLEx01.java
<?xml version="1.0" encoding="UTF-8"?>
<!-- XML 주석 -->
<persons><!--root element -->
    <person code = "810225-1000000"><!--code는 속성 -->
        <name>박지성</name>
        <age>29</age>
        <phone>010-777-7777</phone>
    </person>

    <person code = "900905-2000000">
        <name>김연아</name>
        <age>20</age>
        <phone>010-999-9999</phone>
    </person>
</persons>
```

■ DTD 선언 : [!DOCTYPE 루트 엘리먼트이름 [...내부DTD...]>

■ 요소 선언

```
<!ELEMENT 이름 (내부 엘리먼트이름 or #PCDATA or EMPTY) >
```

■ 선언자 기호

선언자 기호	설 명
무기호	특별한 기호가 없으면 지정한 요소를 1번만 정의
,	선언한 요소의 순서대로 자식 요소를 사용
	선언한 요소의 사용에 따라 자식 요소를 골라 사용
()	() 요소를 그룹으로 선언해서 정의
*	선언한 요소를 0번 이상 사용할 수 있음(0~n)
+	선언한 요소를 1번 이상 사용할 수 있음(1~n)
?	선언한 요소를 0, 1번 사용할 수 있음

PCDATA란?

PCDATA는 Parsed Character DATA 의 약자로서 XML 파서(해석기)가 해석하는 문자 데이터를 의미함.
해석을 한다면 말은 문자 데이터 안에 XML 권고안에 규정되어 있는 문자로 쓰여 졌는지 조사하고,
또한 엔티티 참조가 있다면 엔티티 참조에 해당하는 값으로 치환하는 작업을 의미함

■ 속성 선언

<!ATTLIST 요소이름 속성이름 속성타입 속성_기본값 데이터_기본값>

■ 속성의 기본값

속성 기본값	설 명
속성 초기값	속성 값을 지정하지 않는 경우에는 초기값을 사용하여 정의함
#FIXED	XML 문서에서 속성을 정의할 수도 있고, 안할 수도 있음. 명시한다면 초기값과 일치
#IMPLIED	XML에서 속성을 명시할 수도 있고, 안할 수도 있음(초기값 X)
#REQUIRED	속성을 반드시 명시적으로 사용해야 함(초기값 X)

■ 속성 타입

속성 타입	설 명
열거형	사용자가 정의해 놓은 여러 속성 중 하나를 선택하도록 정의
ID	속성 값이 유일한 문자 데이터를 의미, 요소를 구분하는 데 사용(숫자로 시작 불가)
IDREF	속성 값이 문서 내부에서 선언한 ID를 참조
IDREFS	속성 값이 문서 내부에서 선언한 여러 ID를 참조
NMTOKEN	ID와 유사하고 속성 값을 XML 문서 작성 규칙에 맞게 작성해야 함
NMTOKENS	공백 문자로 구분한 NMTOKEN을 여러 개 사용 가능
ENTITY	외부의 이진 데이터 참조
ENTITYS	외부의 이진 데이터 이름을 여러 개 참조

■ DTD 문서 예제

XMLe02.java

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

<!DOCTYPE 학생정보 [
    <!ELEMENT 학생정보 (학생)>
    <!ELEMENT 학생 (이름,전화,이메일)>
    <!ELEMENT 이름 (#PCDATA)>
    <!ELEMENT 전화 (#PCDATA)>
    <!ELEMENT 이메일 (#PCDATA)>
]>

<학생정보>
    <학생>
        <이름>손흥민</이름>
        <전화>010</전화>
        <이메일>son@korea.com</이메일>
    </학생>
</학생정보>
```

XMLe03.java

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

<!DOCTYPE 회원목록 [
    <!ELEMENT 회원목록 (회원+,이달의회원,신규회원)>
    <!ELEMENT 회원 (이름,전화번호)>
    <!ELEMENT 이름 (#PCDATA)>
    <!ELEMENT 전화번호 (#PCDATA)>
    <!ELEMENT 이달의회원 EMPTY>
    <!ELEMENT 신규회원 EMPTY>

    <!ATTLIST 회원 등급 (준회원|정회원|우수회원) "준회원">
    <!ATTLIST 회원 회원번호 ID #REQUIRED>

    <!ATTLIST 이달의회원 회원번호 IDREF #REQUIRED>

    <!ATTLIST 신규회원 회원번호 IDREFS #REQUIRED>
]>

<회원목록>
    <회원 등급="정회원" 회원번호="A123">
        <이름>Michael</이름>
        <전화번호>010</전화번호>
    </회원>
    <회원 회원번호="B125">
        <이름>김연아</이름>
        <전화번호>010</전화번호>
    </회원>
    <이달의회원 회원번호="A123"></이달의회원>
```

```
<신규회원 회원번호="B125 A123"/>
</회원목록>
```

■ 연습문제

XMLEx04.java

루트엘리먼트는 사원이며 사원 안에는 명함이라는 엘리먼트를 두어야 한다.
그리고 명함은 중복이 가능하지만 반드시 1개는 존재해야 한다.
명함에는 회사명과 부서, 성명 그리고 주소와 전화, 팩스 같은
하위엘리먼트들을 순서적으로 가진다.
여기서 전화는 중복이 가능하지만 생략해서는 안되며,
팩스는 생략이 가능하지만 중복을 허용하지 않는다.
성명에는 사번이라는 속성을 정의하고, 속성 값은 파싱을 하지 않는
문자 데이터가 들어오며 반드시 속성을 정의해야 한다.
주소의 하위 엘리먼트는 '시'와 '구' 아니면 '도'와 '시'가
정의된 후 '동'이 순서적으로 정의 되어야 한다.

XMLEx04.java

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

```
<!--
```

루트엘리먼트는 사원이며 사원 안에는 명함이라는 엘리먼트를 두어야 한다.

그리고 명함은 중복이 가능하지만 반드시 1개는 존재해야 한다.

명함에는 회사명과 부서, 성명 그리고 주소와 전화, 팩스 같은

하위엘리먼트들을 순서적으로 가진다.

여기서 전화는 중복이 가능하지만 생략해서는 안되며,

팩스는 생략이 가능하지만 중복을 허용하지 않는다.

성명에는 사번이라는 속성을 정의하고, 속성 값은 파싱을 하지 않는

문자 데이터가 들어오며 반드시 속성을 정의해야 한다.

주소의 하위 엘리먼트는 '시'와 '구' 아니면 '도'와 '시'가

정의된 후 '동'이 순서적으로 정의 되어야 한다.

```
-->
```

```
<!DOCTYPE 사원 [
```

```
<!ELEMENT 사원 (명함)+>
```

```
<!ELEMENT 명함 (회사명,부서,성명,주소,전화+,팩스?)>
```

```
<!ELEMENT 회사명 (#PCDATA)>
```

```
<!ELEMENT 부서 (#PCDATA)>
```

```
<!ELEMENT 성명 (#PCDATA)>
```

```
<!ELEMENT 주소 (((시,구)|(도,시)),동)>
```

```
<!ELEMENT 시 (#PCDATA)>
```

```
<!ELEMENT 구 (#PCDATA)>
```

```
<!ELEMENT 도 (#PCDATA)>
```

```
<!ELEMENT 동 (#PCDATA)>
```

```
<!ELEMENT 전화 (#PCDATA)>
```

```
<!ELEMENT 팩스 (#PCDATA)>
```

```
<!ATTLIST 성명 사번 CDATA #REQUIRED>
```