

XML, XLink, Namespace, DTD, Schema, CSS, XHTML 등 ... 만약 XML을 새로이 접하게 된다면, 이와같은 용어는 매우 어렵게 느껴질것이다. 이 요약은 XML의 초보자들에게 기본적인 개념에 대하여 10 가지를 제공 함으로써 나무를 통하여 숲을 볼 수 있도록 하기 위한것이다. 그리고 만약에 XML에 대하여 발표를 하게 된다면, 이 10 가지에 대하여 시작해보지 않으시겠습니까?

### 1. XML은 구조적인 데이터를 위한것이다

스프레드 시트, 주소록, 형상 파라미터, 비즈니스 트랜잭션, 그리고 기술 도면 등 구조를 가지는 데이터는 사물을 포함한다. XML은 텍스트 포맷을 설계하기 위한 규칙(가이드 라인이나 관례라고 생각할 수 있다)들의 집합이며, 이러한 규칙을 통하여 데이터를 체계화한다. XML은 프로그래밍 언어가 아니며, 따라서 XML을 배우거나 사용하기 위하여 프로그래머가 될 필요는 없다. XML은 컴퓨터가 쉽게 데이터를 생성하고, 읽고, 데이터 구조가 모호하지 않도록 하게 한다. XML은 언어를 설계하는데 일상적인 오류를 범하지 않도록 하며, 확장 가능하고, 플랫폼 독립적이며, 국제화와 지역화를 지원한다. XML은 유니코드를 충실히 지원한다.

### 2. XML은 다소 HTML과 같이 보인다

HTML과 같이 XML은 태그('<'와 '>' 등의 꺾쇠괄호)와 속성(이름="값"형태)을 이용한다. HTML이 각 태그와 속성의 의미가 무엇이고, 태그들 가운데의 텍스트가 어떻게 브라우저에 보일것인지를 기술하는 반면에, XML은 태그를 데이터들의 분리자로 사용할 뿐만 아니라 데이터에 대한 해석을 완전히 그것을 읽는 응용 프로그램에 맡긴다. 바꾸어 말하면, 만약 "<p>" 태그가 XML에 있다면, 이를 절(paragraph)로 추측하지 말아야 한다. 문맥에 의존적으로 값(price), 파라미터(parameter), 사람(person), 또다른 p...가 될 수 있다.

### 3. XML은 텍스트이며, 읽혀지는것만을 의미하지 않는다

스프레드 시트, 주소록, 그리고 다른 구조 데이터를 만들어내는 프로그램들은 종종 그 데이터를 바이너리나 다른 텍스트 형태로 디스크에 저장한다. 텍스트 포맷의 장점은 필요하다면 사람으로 하여금 그 데이터를 생산해내는 프로그램 없이 볼 수 있으며, 급할때는 텍스트 편집기로 텍스트 포맷을 읽을 수 있다. 또한, 텍스트 포맷은 개발자로 하여금 응용 프로그램을 더욱 쉽게 디버그 할 수 있도록 한다. HTML과 같이 XML 화일은 텍스트 화일이고, 사람들은 이를 직접 읽을 필요는 없으나, 필요하다면 읽을 수는 있다. HTML과 다른것은, XML 화일을 위한 규칙은 엄격하다. 태그들이 일치하지 않거나, 속성이 큰따옴표 없이 사용되면, XML 화일은 사용할 수 없게 된다. 반면에 HTML에서는 그러한 것에 대하여 관대하고, 또 명시적으로 허용한다. 공식적 XML 명세는 어플리케이션들이 잘못된 XML 화일의 생성을 만드는것을 금하고 있으며, 만약 XML 파일이 잘못되었다면, 그 즉시 처리를 중단하고, 에러를 알려야 한다.

### 4. XML은 크기가 커진다

XML이 텍스트 포맷이고 데이터를 구별하기 위하여 태그를 사용하기 때문에 XML 화일은 바이너리 포맷과 비교하면, 거의 그 크기가 크다. 텍스트 포맷의 장점은 명백하며(3번 항목 참조), 단점은 다른 방법으로 상쇄될 수 있다. 디스크 크기는 사용되어지는 것보다는 그리 고비용이 아니며, zip과 gzip과 같은 압축 프로그램은 빠르고 좋게 화일을 압축한다. 더욱이, 모뎀 프로토콜과 웹의 핵심 프로토콜인 HTTP/1.1과 같은 통신 프로토콜은 데이터를 즉각적으로 압축하고, 바이너리 포맷만큼 효과적으로 대역폭을 절약할 수 있다.

### 5. XML은 기술의 집합이다

XML 1.0 은 "태그"와 "속성"을 정의하는 명세이다. XML 1.0 이면에는 중요하고 자주 요청되는 일들에 유용한 서비스를 제공하는 모듈의 집합인 "XML 군(family)"이 있다. Xlink는 XML 화일에 하이퍼링크를 기술하는 표준 방법을 설명하고 있다. XPointer와 XFragments는 XML 문서의 일부분을 가르키기 위한 구문이다. XPointer는 URL과 비슷하나, URL이 웹상의 문서만을 가르키는 대신에 XPointer는 XML 내의 데이터 일부분을 가르킨다. CSS는 스타일 시트 언어로서 XML을 HTML과 같이 보이도록 하는데 적용된다. XSL은 스타일 시트를 정의할 수 있도록 개선된 언어이다. XSL은 태그들과 속성들을 재배치하고, 추가나 삭제하기 위하여 사용되는 변환 언어인 XSLT를 기반으로 한다. DOM은 프로그래밍 언어에서 XML(및 HTML) 화일을 조작하기 위한 표준 함수들의 집합이다. XML Schemas1과 2는 개발자로 하여금 정확하게 XML 기반의 포맷의 구조를 정의하는데 이용된다. 이밖에도 더 많은 모듈과 툴들을 이용할 수 있으며, 개발중에 있다. W3C의 기술 문서 페이지를 지켜보기 바란다.

### 6. XML은 새롭지만, 그렇게 새로운 것은 아니다

XML의 개발은 1996년에 시작되었고, 1998년 2월에 W3C의 권고안으로 제정되었으나 다소 미완성의 기술이었다. 사실, 기술은 새로운 것이 아니었다. XML 이전에 1980년대 중반에 개발되고 1986년에 ISO 표준으로 제정되었으며, 대용량 문서 프로젝트에 광범위하게 사용되는 SGML이 있었으며, 1990년에 개발된 HTML이 있었다. XML 설계자는 HTML의 시도를 참고로 단순히 SGML의 장점만을 취하여 SGML과 못지않은 기능을 제공하였으며, 더욱 사용하기 쉽도록 하였다. 그러나, 일부 진화된 모습을 혁명으로 말하기는 어렵다... 그리고, SGML이 대부분 기술 문서에 이용되고, 다른 종류의 데이터에는 덜 이용되는 반면에, XML은 그렇지 않다고 말할 수 있다.

### 7. XML은 HTML에서 XHTML로 이끌었다

중요한 XML 응용은 문서 포맷이고, HTML이후의 W3C의 XHTML이 바로 그것이다. XHTML은 HTML과 많은 부분에서 같은 엘리먼트가 이용된다. 문법은 XML의 규칙을 준수하기 위하여 약간만 변경되었다. "XML 기반의" 문서는 XML로부터 문법을 상속받았으며, 어떤 부분에서는 제약(예를 들어, XHTML은 "<p>"은 허용하지만, "<r>"은 허용하지 않는다)을 받으며; 그 문법에 의미(XHTML에서는 "<p>"이 "절(paragraph)"을 나타내며, "가격(price)", "사람(person)" 또다른 것을 의미하지 않는다)를 추가하기도 한다.



### 8. XML은 모듈식이다

XML은 다른 포맷을 조합하거나 재사용함으로써 새로운 문서 포맷을 정의하는것을 허용한다. 두개의 포맷이 독립적으로 개발되었을때 같은 이름의 엘리먼트나 어트리뷰트를 가질 수 있으며, 이를 조합할때는 주의하여야 한다. ("<p>"는 어떤 포맷에서는 "절(paragraph)"을 의미하지만, 다른 포맷에서는 "사람(person)"을 의미한다. 이름 혼동을 배제하기 위해서 포맷이 조합될 때, XML은 이름공간 메커니즘을 제공한다. XSL과 RDF는 이름공간을 사용하는 XML 기반 포맷의 좋은 예이다. XML 스키마는 두개의 스키마를 결합하여 합병된 문서 구조를 가진 또다른 스키마를 만들어내기 용이하게 함으로써, XML 문서 구조 정의의 단계에서 모듈화를 지원을 반영하기 위하여 설계되었다.



### 9. XML은 RDF와 시맨틱 웹의 토대이다

W3C의 RDF는 음악 연주 리스트, 사진 수집, 서지 목록과 같은 자원 설명과 메타데이터 응용을 지원하기 위한 XML 텍스트 포맷이다. 예를 들어, RDF는 개인 연락 리스트로부터의 정보를 이용하여 웹 사진 앨범의 사람을 식별하며, 자동적으로 웹상의 사진에 있는 사람들에게 메시지를 보내는데 이용된다. HTML이 문서, 메뉴



시스템, 그리고 폼 어플리케이션을 초창기 웹에 통합한 것과 같이, RDF는 어플리케이션과 에이전트를 시맨틱 웹에 통합한다. 사람이 대화에 사용되는 단어의 의미에 합의하는것이 필요하듯이, 컴퓨터도 효과적으로 커뮤니케이션하기 위한 단어의 의미에 합의하는 메커니즘이 필요하다. 어떤 분야(예를 들어 쇼핑 또는 제조)에서 용어의 형식적인 설명은 온톨로지를 필요하게 되고, 그러한것은 시맨틱 웹의 필수적인 부분이다. RDF, 온톨로지, 그리고 의미의 표현은 컴퓨터가 사람에게 일하는것을 도와주게 되고, 그러한 모든것이 시맨틱 웹 활동의 주제이다.

### 10. XML은 라이선스 제약이 없으며, 플랫폼 독립적이고, 많은 지원이 있다

프로젝트에 기본으로서 XML을 선택함으로써, XML 관련 툴(이들은 이미 원하는 것을 할 수 있다)을 위한 크고 활동적인 커뮤니티를 접근하는 이득을 얻고, 기술적인 부분에서의 기술자들의 경험을 얻는 것이다. XML을 채택하는 것은 데이터 베이스를 위하여 SQL을 선택하는 것과 약간 비슷하다. - 자신의 데이터 베이스를 만들어야 하며, 그를 위한 프로그램과 조작을 위한 프로시저어를 만들어야 하지만, 이용할 수 있는 많은 툴이 있고, 도움을 줄 수 있는 많은 사람들이 있다는 부분에서 비슷한 것이다. 그리고, XML이 라이선스에 제약이 없기 때문에, 어떠한 사람에게 어떠한 것을 지불하지 않고도 자신의 소프트웨어를 만들 수 있다. 또한 많은 지원과 성장적인 지원의 의미는 단일 회사에 묶여 있는 것이 아니라는 것이다. XML은 항상 최고의 해결책은 아니지만, 항상 고려할만한 가치는 있다.



W3C 커뮤니케이션 팀, w3t-comm@w3.org

<http://www.w3c.or.kr/Translation/XML-in-10-points.html.kr> 에서 원문을 보실 수 있습니다.