



The Future of Web Standards, HTML5

윤 석 찬

(주)다음커뮤니케이션

Channy Yun

Daum Communications Corp.

<http://channy.creation.net/blog>

About Me

- My Job
 - 2004~ Technical Evangelist in Daum Communications.
 - <http://dna.daum.net>
- Interests
 - Web Standards, Open Source, Web 2.0
- Activities
 - 2002~ Mozilla L10N ko module owner
 - <http://www.mozilla.or.kr>
 - 2004~ Web Standards Group in Korea
 - <http://webstandard.or.kr>

Agenda

- 목적
 - 차세대 웹 표준의 방향에 대한 모색
 - 보다 적극적인 웹 표준 활동 참여
- 웹 표준 동향
 - 제 2 웹 브라우저 전쟁 (W3C vs. WHATWG)
 - W3C
 - XHTML2 (CSS3, XForm, XMLEvents, Xframes...)
 - Rich Web Clients (XMLHttpRequest, Widget...)
 - WHATWG
 - WebForm 2.0
 - WebApplication 1.0
 - ECMAScript
 - E4X, Javascript 2.0
- 향후 전망 및 참여 방법

World Wide Web Consortium(W3C)

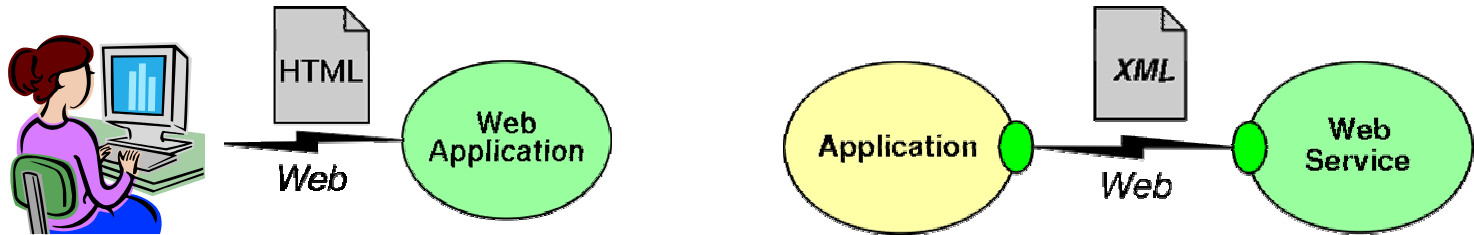
- 운영 현황
 - 시작: 웹을 발명한 Tim Berners-Lee에 의해 1994년 시작
 - 400개 회원사, 65명의 직원 (2006년 현재)
 - MIT (US), ERCIM (프랑스), 게이오대(Japan) 외 15개 지역 사무국 운영
- 운영 방법
 - 강력한 표준안 제정 과정
 - 특허 무료 정책
 - 공감대, 개방성, 상호 운용성 중시
 - 40개 이상의 다른 표준 단체와 연계 작업
 - 국제 표준에 맞는 합리적인 운영
- 주요 목표
 - 웹 문서에서 하나의 '웹'을 지향
 - 데이터서비스: XML, 웹 서비스(Web Services), 시맨틱 웹(Semantic Web)



W3C 주요 목표

- XML : 다수의 플랫폼이 지원, 무상 특허로 이용 가능
 - 기계 및 인간 모두에 대한 접근성 가능, 국제화 용이
 - XML은 읽기 쉽고, 검증하기 용이하며 어플리케이션이 아닌 데이터로 관리
 - 주요 이슈
 - 텍스트 메시징의 문제 (바이너리 XML)
 - 웹에서 XML 전환이 사실상 실패
- 웹 서비스 : 기계-기계 상호 교환 (cf. 사람-기계 상호 교환)
 - XML 기반 메시지 처리: 플랫폼 독립적인 모델
 - 확장성을 고려한 프레임워크: 상호 운용성 확보
 - 기계가 처리할 수 있는 형식: 통합 적합성 확보
 - 유연한 연결성 : 확장성 고려
- 시맨틱 웹
 - 데이터와 의미를 가진 웹
 - 외양 기준 웹을 기계가 처리 가능한 의미 있는 웹으로 전환 하는 것이 목표

W3C의 기술 비전



- 모든 것 위의 하나의 웹(Web)

- 하나의 웹(One Web)

- 웹 기술은 모든 디바이스를 통해 상호 교환 가능해야 함, 컴퓨터, 임베디드, 사람, 집, 직장, 교통, 산업, 의료 분야 전 시스템

- 상호 교환 기술: XHTML, XForms, CSS, DOM, SVG, SMIL, Voice, MMI
 - 리치 웹 기술: Compound Doc Formats, Web Applications
 - 모바일 웹 기술: Mobile Web Initiative
 - 유비쿼터스 기술: Embedded systems, Ubiquitous Web

XHTML

- XHTML: XML 기반 마크업 언어
 - XHTML 1.0: HTML 4을 XML화(Well-formed) 시킴
 - HTML의 전환 전략으로 만들어짐
 - XHTML 1.1: XHTML 1.0 Strict을 완성 시킴
 - XHTML Basic: 휴대폰, TV 및 프린터에서도 사용 가능
 - XHTML2: 진정한 XML 기반 마크업 언어
- XHTML 2.0 주요 특징
 - XML 기반 원리에 충실
 - 구조적 마크업에 치중
 - 더 많은 시맨틱 마크업 채택
 - 사용성 및 접근성, 국제화 강화
 - XForm, XMLEvents 사용
 - 스크립트 사용 지양 및 장치 독립성 확대



XHTML 2.0 (1)

- Generic XML

- 모든 외양은 CSS로 선언 가능하게 함
- 모든 Element에 하이퍼 링크 추가

```
<div href="block.html">New Blogck</div>
<h3 href="home.html">ABCD</h3>
```

- 모든 외양 태그 삭제

```
 >Browser
Images</img>
```

- 구조적 마크업 대거 확장

- ~~
~~ 대신 ~~<hr/>~~, ~~<h1/>~~, ~~<h2/>~~ 대신 ~~~~,

```
<1> I think that I shall never see </1>
<1> A poem lovely as a tree </1>
```

~~<h2>Chapter 1</h2>~~ 및 ~~<h>~~ 신설

```
<section>
<h3>Section 1</h3>
```

```
<section> <h>Chapter 1</h>
...
<section> <h>Section 1</h>
... </section>...
</section>
```


XHTML 2.0 (2)

- 시맨틱 태그 강화
 - <meta> 및 <link>에 있는 title 같은 속성을 모든 태그에서 사용 가능
`<h property="title">My Life and Time</h>, <p role="note">This is note</p>`
- 주요 이슈
 - HTML에 대한 하위 호환성(Backward Compatibility) 포기를 전제로 함
 - , <applet> 등 외부 객체를 표현은 <object>로 통일
 - <form>은 확장 가능한 XForm을 제공
 - XML Events 사용을 위해 <script>를 <handler>로 대체
 - URL 연속성을 위해 <frame>을 Xframes로 대체
- 가장 큰 이슈
 - 과감한 변화에 따른 비용 증가에 대한 효용성 부족
 - 브라우저 벤더의 저항
 - 웹은 W3C의 방향에 따라 움직여 주지 않고 있음
 - MathML, SVG

W3C Rich Web Clients

- W3C 내 새로운 그룹 형성
 - 2005년 10월 결성
 - Ajax 및 위젯 등 동적 UI 및 XML기반 웹 어플리케이션에 뒤늦게 호응
- Web Application Formats W/G
 - Web Forms 2.0 2006-08-21 Working Draft
 - XML Binding Language (XBL) 2.0 2006-09-07 Last Call Working Draft
 - Client-Side Web Applications (Widgets) Requirements 2006-11-09 Working Draft
 - Widgets 1.0 2006-11 Working Draft
- Web APIs W/G
 - XMLHttpRequest Object, Window Object 1.0.
 - DOM Level 3 Events.
 - File Upload, Selectors API, Remote Events for XML (REX) 1.0

W3C에 대한 비판

- 기술 사양에 대한 비판
 - IE6을 따르는 웹 저작 환경에서 XML 스펙의 과도한 확장으로 현실 웹과 괴리 (웹의 변화 속도에 맞추지 못하는 상황 지속)
 - 너무 어렵고 긴 구현 스펙(DOM3 219p, SVG 719p)으로 인한 구현 어려움
 - Web 2.0 및 Rich Web 등 새로운 변화에 대응하지 못하고, 문서 위주의 어려운 스펙을 남발하고 있음
 - SOAP, WSDL 등 웹 서비스 역시 현실 웹에서 거의 사용되지 않고 있으며 그나마 REST방식으로만 진행 중
- 조직에 대한 비판
 - 서버 기반 회원사 증가로 클라이언트 기반(브라우저) 회원사들의 배제
 - 방대한 조직으로 인해 핵심에 집중하지 않은 너무 큰 이상적인 목표 치중
 - 각 W/G의 책임감 있는 운영 부재 및 외부와의 커뮤니케이션 단절
 - 각 W/G의 방만한 회원사 운영 및 비 생산적인 활동
 - 재정적으로 독립되지 못한 상태로 특정 회원사의 이익만을 대변할 가능성이 큼
 - 현재 W3C 회원사가 점점 줄고 있는 상태임

WHATWG

- 설립 계기
 - 2004. 4 W3C 웨어플리케이션 CDF 워크샵 (<http://www.w3.org/2004/04/webapps-cdf-ws/>)
 - 2004. 6 WHATWG 결성 (Web Hypertext Application Technology Working Group, <http://whatwg.org>)
- 주요 목표
 - 웹 페이지 주요 기능인 HTML을 확장.
 - 지금의 “웹 수준”에 맞는 “적합한 웹” 개발 모델 제시
 - 리치 웹 어플리케이션 및 동적 웹 서비스 제시
- 주요 멤버: Mozilla, Opera, Apple, Google
- 활동 방법
 - 공개된 메일링 리스트를 통해 의견 청취
 - 오픈 소스 방법론을 통한 스펙 개발

WHATWG 작업 스펙

- WebForm 2.0 (기초 작업 완료)
 - 웹 저작자들이 Form을 확장하여 사용할 수 있도록 함
 - 웹 어플리케이션 제작 시 요구 기능 추가
- Web Application 1.0 (작업 중)
 - HTML에 시맨틱과 구조적 마크업 확장
 - 고급 위젯에 사용할 수 있는 메뉴, 컨트롤, 툴바 정의
 - 브라우저 환경 개선 및 오프라인 브라우징, 통신 처리 개선
- Web Controls 1.0 (작업 미 시작)
 - 자바스크립트와 CSS에 다양한 위젯 개발 기능을 부여
 - XBL2의 스펙에도 영향을 줌

WebForm 2.0 - Form 속성 확장

- Type 속성값 확장을 통한 시맨틱 처리

```
<label>E-mail address: <input type="email"
  name="addr"></label>
```

```
<label>Start date: <input type="date" name="start"></label>
```

- 필수 요소에 대한 처리 기능

```
<label>Quantity: <input type="number" required="required"
  name="qty"></label>
```

- Form에 입력되는 값의 범위 측정

```
<label>Meeting time: <input type="time" min="09:00"
  max="17:00" name="mt"></label>
```

- Form 입력 값 유효성 체크

```
var myControl = document.forms[0].addr;
```

```
if (myControl.value == 'a@b.c') {
```

```
  myControl.setCustomValidity('You must enter your real address.');
```

```
}
```

```
<label>Home page: <input type="url" name="hp"
```

```
  required="required" oninvalid="alert('You must enter a valid home
  page address.');" return false;">
```

```
</label>
```

WebForm 2.0 – 중복 폼 양식 템플릿

```
<!DOCTYPE html>
<html>
<head><title>Form Repeat Demo</title></head>
<body>
<form action="echo" method="post" enctype="multipart/form-data">
<table>
<thead><tr><th>상품명</th> <th>갯수</th><th>관리</th></tr>
</thead>
<tbody>
<tr repeat="0">
<td><input type="text" name="name_0" value="Digital Camera"></td>
<td><input type="text" name="count_0" value="2"></td>
<td><button type="remove">Delete Row</button></td>
</tr>
<tr repeat="template" id="row">
<td><input type="text" name="name_[row]" value=""></td>
<td><input type="text" name="count_[row]" value="1"></td>
<td><button type="remove">Delete Row</button></td>
</tr>
</tbody>
</table>
<p><button type="add" template="row" >Add Row</button>
<button type="submit">Submit</button></p>
</form>
</body>
</html>
```

WebForm 2.0 – Form 이벤트 수정

- 입력 시 확인 이벤트 추가

```
<p>Search: <input type="text" oninput="search(value)"></p>
```

- Form 유효성 확인 이벤트 추가

```
<label><input name="byte" type="number" min="0" max="255" required="required"
oninvalid="failed(event)" /></label>
<output name="error"/>
```

```
<script type="text/javascript"> <![CDATA[
function failed(event) {
    var form = event.target.form;
    form.error.value = '아래와 같은 오류가 있습니다.';
    if (event.target.validity.typeMismatch)
        form.error.value = '숫자만 허용 됩니다..';
    else if (event.target.validity.rangeUnderflow)
        form.error.value = '숫자가 0보다는 커야 합니다.';
    else if (event.target.validity.rangeOverflow)
        form.error.value = '숫자가 255이하여야 합니다.';
    event.preventDefault();
}
]]> </script>
</form>
```

- 기타 Form 이벤트 관련 버그 수정

WebForm 2.0- 기타

- 폼 처리 방법 명시
 - Form Submit에 대한 절차 명세를 정함
 - 문자 인코딩이 다른 문자가 포함 될 경우 처리 방법 재기술
 - 프로토콜별 Form Submit에 처리 방법 재 기술
 - 추가된 요소 및 속성에 대한 DOM 처리 방법 기술
 - 추가된 요소 및 속성에 대한 CSS 처리 방법 기술
- 경과 사항
 - 2006년 8월에 W3C에 Working Draft로 제공
 - <http://www.w3.org/TR/web-forms-2/>

Web Application 1.0 – 시맨틱 및 구조적 마크업

- 섹션 구조 태그 추가

`<header>`, `<nav>`, `<section>`, `<article>`, `<aside>`, `<footer>`

- 시맨틱 태그 추가

```
<code>var i: Integer; i := <m>1.1</m>; end.</code>  
<t lang="fr">je ne sais quoi</t>  
<time datetime="2006-09-23">a saturday</time>  
<meter min="0" max="100" value="75"></meter>  
<meter value="0.25">Low activity,</meter>  
Progress: <progress><span id="p">0</span>%</progress>
```

- 콘텐츠 임베딩

``, `<iframe>`, `<embed>`, `<object>`-`<param>` `<figure>`

- 테이블 모델 추가

`createCaption()`, `createTHead()`, `deleteTFoot()` ...

Web Application 1.0 – 정적 데이터 처리

- 전통 브라우저에서 HTML (DOM)처리의 문제
 - HTML상의 DOM Tree 오류 수정
 - 동적 마크업 추가 - innerHTML()
 - 이벤트 핸들러 추가 – AddEventListener()
 - HTML 문서에 API 추가 - document.createElement()
- 표준 데이터 포맷 정의
 - 숫자, 날짜 형식 정의
- 다양한 `<link type="xxx">` `` (By Microformat)
 - Contact, bookmark, license, sidebar, help, icon, tag...
 - `<link rel="feed alternate" href="data.xml">`
 - `This`
 - `(Help)`

Web Application 1.0 – 동적 데이터 처리

- Datagrid : 데이터를 찾는 새로운 방법 (New Table?)
 - 행과 열을 가지는 새로운 데이터 트리 구조
 - Datagrid는 새로운 이벤트 모델로 제어가 가능하다
 - 데이터 얻기: `getRowCount()`, `getCaptionText()`
 - 데이터 편집: `setRow()`, `editCell()`, `updateEverything()`
 - 데이터 핸들: `selectAll()`, `invert()`, `clear()`
- 스크립트 실행 방식 정의
 - `async/비동기적 실행`, `defer/실행 연기`
- 신규 이벤트 요소
 - `<command>`: 신규 이벤트 처리 방법
 - `<menu>`: `command`의 목록
- 데이터 편집
 - `contenteditable` 속성 제공

```
<datagrid>
  <ol>
    <li> row 0 </li>
    <li> row 1
      <ol>
        <li> row 1,0 </li>
      </ol>
    </li>
    <li> row 2 </li>
  </ol>
</datagrid>
```

Web Application 1.0 – 브라우저 환경 지원

- 브라우저 윈도우를 제어하여 드래그앤 드롭, 선택, 명령 실행 및 취소 기능을 제공할 수 있음
 - 세션 히스토리 처리 기능 e.g go(), forward()
 - 브라우저 상태 별 처리 기능 e.g. navigator.offline()
 - 사용자 정의 프로토콜 및 핸들러 e.g. fax:// registerContentHandler()
 - 클라이언트 기반 세션 스토리지 e.g. Firefox DOM Storage
 - 드래그앤 드롭 e.g. draggrable 속성, dragStart(), drag(), drop()
 - 명령 실행/취소 기능 e.g removeUndo(), redo(), selectAll()
 - 텍스트 선택 기능 e.g selection(), selectStart()
- 멀티미디어
 - 〈canvas〉, var a = new Audio("test.wav");

Web Application 1.0 – 통신

- DOM 이벤트의 서버 통신

- application/x-dom-event-stream

```
<event-source src="http://stocks.example.com/ticker.php"
  id="stock">
<script type="text/javascript">
document.getElementById('stock').addEventListener('stock
  change',
  function () {
    var data = event.data.split('\n');
    updateStocks(data[0], data[1], data[2]);
  }, false);
</script>
```

- TCP/IP 통신, P2P 통신 모두 지원
- 도메인 교차 데이터 통신 지원

ECMAScript

- 표준화 진행 상황

- 1997년 [ECMA](#)가 JavaScript를 표준화
- 1998년 ISO 표준으로 개정 (Edition 2)
- 1999년 추가 사항 (Edition 3)
- 2004년 6월 [E4X](#) (ECMAScript for XML)
 - first class object subtype and literal syntax
- 2006년 ECMA Edition 4 제정 중

- 주요 변화 상황

- 대부분 렌더링 엔진은 ECMAScript Edition 4를 99% 이상 지원
- Adobe가 ActionScript Engine을 오픈 소스로 제공
- 대부분 JS 엔진은 오픈 소스로 나오고 있음

현재 상황

- Tim Berners-Lee, “Reinventing HTML” (2006-10-27)
 - <http://dig.csail.mit.edu/breadcrumbs/node/166>
 - Well-formed와 Semantic 마크업의 장점에도 불구하고 HTML에서 XHTML로의 전환은 실패했음을 인정
 - 새로운 HTML 개선 워킹 그룹 개설 제의
- 새로운 표준화 작업 진행 전망
 - WHATWG에서 스펙에 대한 공개 Feedback 시작
 - Web Form 2.0: XForm과 통합 가능성 있음
 - Web Application 1.0: W3C에 HTML5 제안
 - 기존 Rich Web Clients와 공조 가능
- 변수: Internet Explorer 8?

향후 전망

- 새로운 웹 전쟁
 - 제 1 브라우저 전쟁 = 벤더간 경쟁 (Netscape vs. Internet Explorer)
 - 제 2 브라우저 전쟁 = 표준안 경쟁 (WHATWG vs. W3C)
- 시사점
 - 문서 형식으로 만들어진 웹 표준 구도를 벗어나 정보의 접근성 및 호환성은 유지하면서 어플리케이션 기능을 제공할 수 있는 보다 확장된 표준 필요.
 - Ajax의 출현으로 Rich Web 환경이 도래하고 있으며 데이터 구조를 유지하면서 동적 기능을 제공하는 확장 필요.
 - 오픈 소스와 MVC 모델, 빠른 개발을 위한 프레임웍의 출현 등으로 서버 기반 프로그래밍에 비해 클라이언트 기반 프로그래밍의 전성 시대가 열릴 수 있음
 - 클라이언트 기반 개발자들에게 새로운 비전을 열어 줄 수 있음

표준화 참여 방법

- W3C
 - W3C 테크니컬 문서 참조 <http://www.w3.org/TR/>
 - 표준안 단계별의 Last Call 및 Issue 추적
 - W3C 한국 사무국에 의견 제시 (<http://www.w3.or.kr>)
- WHATWG
 - WHATWG 위키 및 블로그: Feedback 및 제안
 - <http://blog.whatwg.org>, <http://wiki.whatwg.org>
 - 공개 메일링 리스트: 진행 사항 확인
 - <http://listserver.dreamhost.com/pipermail/whatwg-whatwg.org/>
 - 표준 스펙 테스트 : 웹 브라우저 구현 단계 별로 테스트
 - Firefox DOM Storage,
<http://channy.creation.net/work/firefox/domstorage/>

웹 표준화 커뮤니티

- 한국 웹 표준 프로젝트
 - <http://webstandard.or.kr>
- CSS 디자인 코리아
 - <http://standardmag.com>
- 모질라 한국 커뮤니티
 - <http://www.mozilla.or.kr>