




김동완
dwkim@nextree.co.kr

강의 요구 기술

- 본 강의는 아래 기술에 대한 이해를 필요로 합니다.
 - 인터넷
 - 웹

- 개발 도구
 - Text Editor (Notepad, Ultraedit, EditPlus)
 - 웹브라우저(Firefox, Opera, Internet Explorer, Chrome, Safari...)

웹 표준과 HTML5

- 
1. 웹이란
 2. 웹 표준
 3. HTML4.01, xHTML1.x
 4. xHTML2.x HTML5
 5. CSS2.0 & CSS3.0
 6. Javascript & AJAX
 7. Canvas, HTML5 Javascript API (Web apps W/G)
 8. HTML5와 Mobile

ONE STEP AHEAD

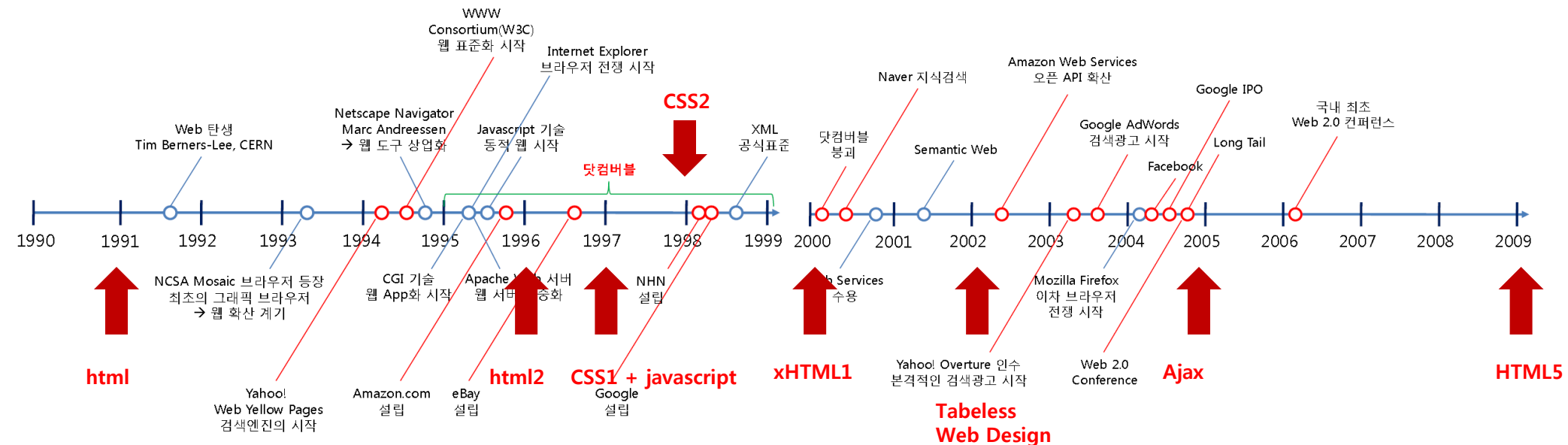
1. 웹의 역사
2. 웹 패러다임의 변화
3. 웹구성요소
4. 플랫폼으로서의 웹
5. HTML5

웹의 역사



들어가기전에 : 인터넷의 시작

- 1969년 : ARPANET 미국국방성 ARPA(Advanced Research Projects Agency)에서 일정지역에 대한 폭탄폭격과 같은 긴급 사태시에도 장애를 받지않고 제 기능을 발휘할 수 있는 통신망 구축을 연구하여 네트워크를 개발, 최초의 연구목적의 네트워크
- 1970~80년대 : 미국내 50개 대학, 연구소들이 알파넷(ARPANET)에 연결, 일반에 공개, 네트워크에 연결
- 1990년대 : **일반상업적인 목적의 네트워크**이 되면서 현재의 인터넷으로 발전
 - 1991년 : 초기 e-mail, FTP, Newgroup등 학자들이나 전문인들이 사용하는등 소규모로 사용되었으나 **CERN의 팀 버너스리**에 의해 **WWW(World Wide Web) 서비스가 개발되어** 일반인들도 쉽게 멀티미디어 정보를 제공할 수 있고 이용할수 있게 되자 폭발적인 성장을 이루게 됨
 - 1993년 : **최초의 웹브라우저 모자이크(Mosaic)개발**; 넷스케이프(Netscape:1994)의 등장, 인터넷 상업화(1995)
- 현재** : 중앙통제방식이 아닌 사용자들의 규약(Protocol)으로 네트워크가 연결되어 있으며, **정보의 흐름에 있어서 양방향성(Full duplex)을 가지고 있기 때문에 전세계 수많은 호스트 컴퓨터가 인터넷에 연결되어 거대한 네트워크**를 형성.



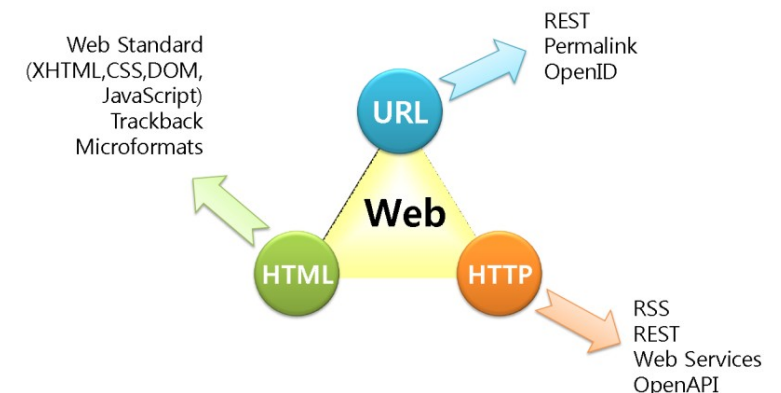
웹 (World Wide Web:WWW, WEB)이란?

□ World Wide Web(Web)

- 정보자원들의 네트워크(Network)이다.
- 인터넷 상의 다양한 자원 또는 정보에 쉽게 접근할 수 있는 기능을 제공해 주는 그래픽 사용자 인터페이스 환경의 **하이퍼텍스트 기반 정보 시스템**.
- 인터넷을 통해 전세계에 거미줄처럼 연결되어 있는 정보를 누구나 쉽게 접근할 수 있도록 하는 시스템 (**URL을 통한 다양한 프로토콜 지원**)

□ 인터넷에 있는 정보자원을 쉽게 접속할수 있게 하는 웹의 3가지 기능

- 규격화되고 통일된 웹자원의 위치지정 방법 (예:URI)
- 웹의 자원이름에 접근하는 프로토콜(예:HTTP)
- 자원들 사이에 쉽게 접근할수 있는 언어(**Hypertext**) (예:HTML)



웹의 구성요소

□ URI (Universal Resource Identifier) : 정보의 위치를 가르킴

- HTML문서, 이미지, 비디오자료, 프로그램등 웹상의 각 자원들을 URI로 인코드(encode) 될 수 있는 주소
- URI의 구성
 - 자원에 접근하는 방법이름. (Protocol)
 - 자원을 가지고 있는 호스트이름.
 - 경로(Path)로 주어진 자원의 이름.

location.href =

```

location.protocol
+ location.host (hostname + port)
+ location.port
+ location.pathname
+ location.search
+ location.hash
  
```

□ 상대URI

- 상대URL의 위치정보를 저장하지 않음.

기준 URI : <http://www.test.com/suport/index.html>

```

<A href="next.html">Next Page</A>
<IMG src="../icons/logo.gif" >
  
```

완전한 URI :

```

http://www.test.com/supoort/next.html
http://www.test.com/icons/logo.gif
  
```

■ HTML에서의URI의 사용

- 다른 문서나 자원에 연결(Link) : <A>, **<LINK>**
- 외부스타일시트(StyleSheet)나 스크립트(Script)에 연결 : <LINK>,<SCRIPT>,<STYLE>
- 이미지,오브젝트(Object),Applet을 문서에 포함시킴 : ,<OBJECT>,<APPLET>
- 이미지맵을 설정 : <MAP>,<AREA>
- 입력을 위한 폼(FORM), 프레임(FRAME)형성 : <FORM>,<FRAME>,<IFRAME>

□ URL(Uniform Resource Locator)

- URI의 일반적인 서브셋(subset)을 형성.
- 웹상에서 **서비스를 제공하는 각 서버들에 있는 파일들의 위치를 명시**하기 위한것으로 접속해야 할 서비스의 종류, 서버의 위치(Domain Name),파일의 위치를 포함.
- 표현 형식
 - protocol://host.domain.first-level-domain/path/filename.ext#hash
 - protocol://host.domain.first-level-domain
- 지원되는 프로토콜

<ul style="list-style-type: none"> • http:// 웹 서버에 존재하는 HTML 문서를 지정 • https:// HTML 문서 지정시 보안을 요구할 때 사용 • file:// 하드 디스크에 있는 화일 지정 • ftp:// FTP 서비스 사용시 • telnet:// TELNET 서비스 사용시 • news:// UseNet 뉴스 그룹 사용시 • mailto: E-mail 전송시 (예:) • ws:// HTML5 WebSocket Streaming 	<pre>location.href = location.protocol + location.host (hostname + port) + location.port + location.pathname + location.search + location.hash</pre>
--	--

□ 마케팅 및 홍보효과로서의 웹

- 인터넷이라는 엄청난 시장 앞에서 전세계 누구나 쉽게 웹에 접속하여 홍보효과의 세계화(Globalization)를 가능하게 함.

□ 정보교환의 수단

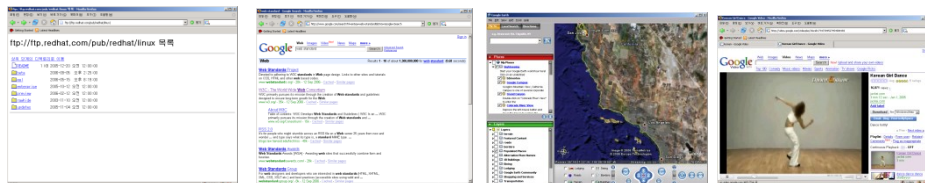
- 시간과 공간을 초월한 정보의 습득, 공유 및 교환이 가능함.
- 사람과 컴퓨터뿐 아니라 사람과 사람 사이의 중요한 커뮤니케이션의 장으로 발전.

□ 웹은 인터넷 문화자체

- 대부분의 인터넷서비스가 웹으로 구현 가능.
- 최근 상당수의 언어가 CGI나 ActiveX의 형태로 웹을 지원**

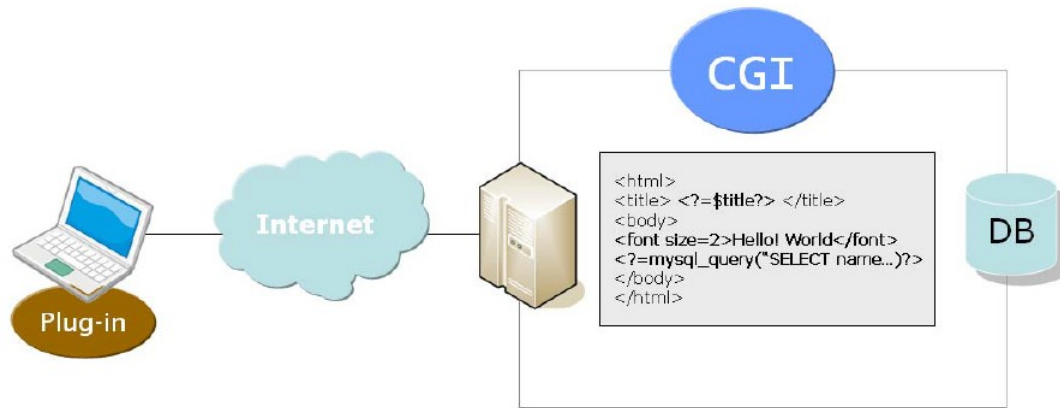
□ 미래가치기술로서의 웹서비스

- 개발단계의 생산성보다 점점 다양해지는 환경에서의 유연성 제고를 통해 운영과정에서 더욱 효과를 발휘
- 아주 빠르게 형성되고 소멸하는 인터넷의 단점에 대한 유연성을 확보



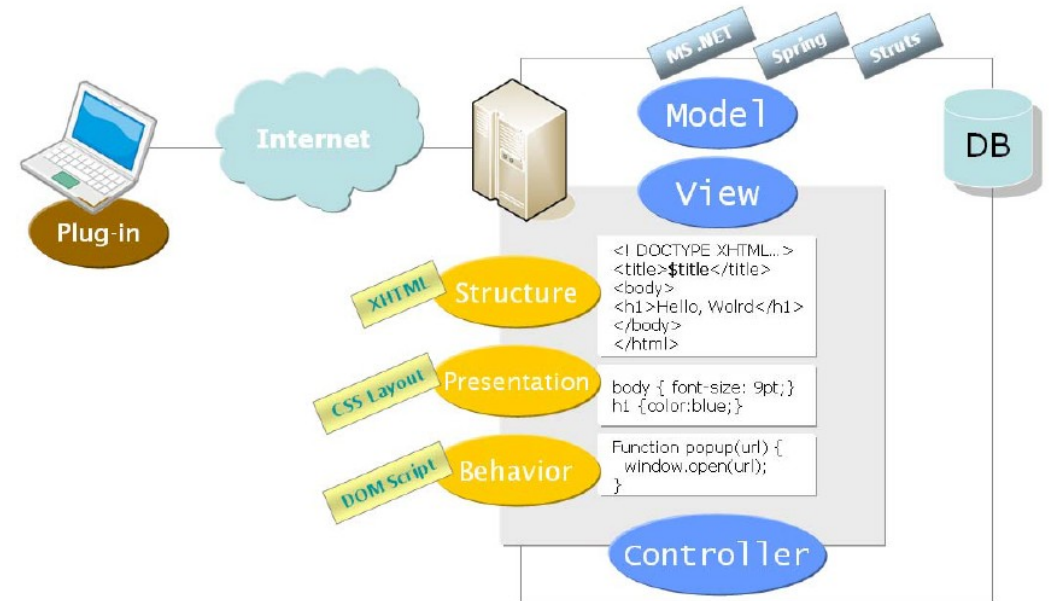
HTML5와 웹개발방법론의 진화 (1/2)

웹문서(Web Document) 시대 (1990년대)



- Web Server ↔ 웹브라우저 간 정적 HTML문서 전달
- 주로 CGI를 이용하여 개발 → 마크업과 프로그램 코드 혼재
- 개발 직군 간의 업무 분담이 전혀 이루어 지지 않음

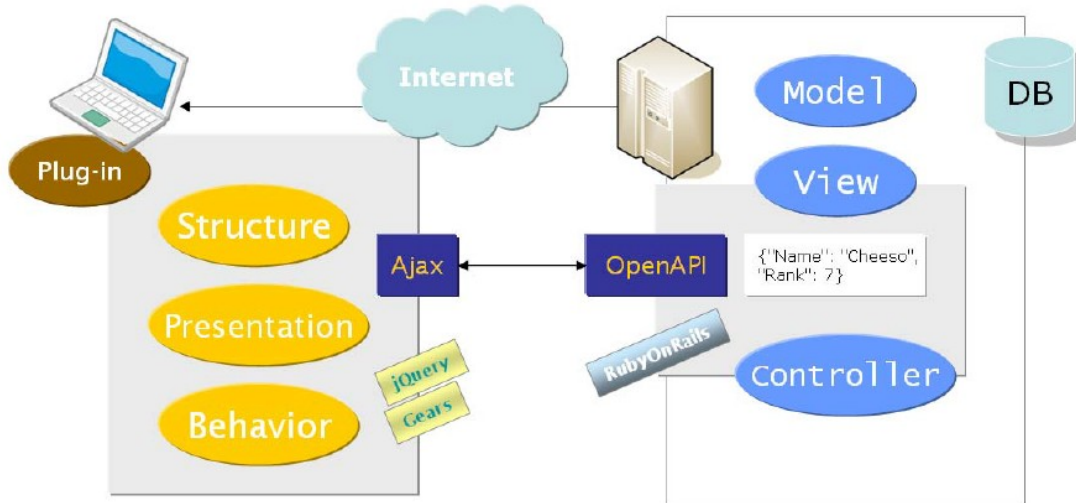
웹표준(Web Standard) 시대 (2000년대 초반)



- 데이터모델, 템플릿, 비즈니스로직이 분리된 코드를 통한 생산성과 높은 효율성
- 프론트엔드 구조(Model : HTML), 표현(View: CSS), 동작(Controllor: Javascript) 분리
- 국내 2004년도 도입 민간/공공 웹사이트에 웹접근성과 더불어 확산됨
- 국내의 경우 웹퍼블리셔(Web Publisher)직군 생겨남

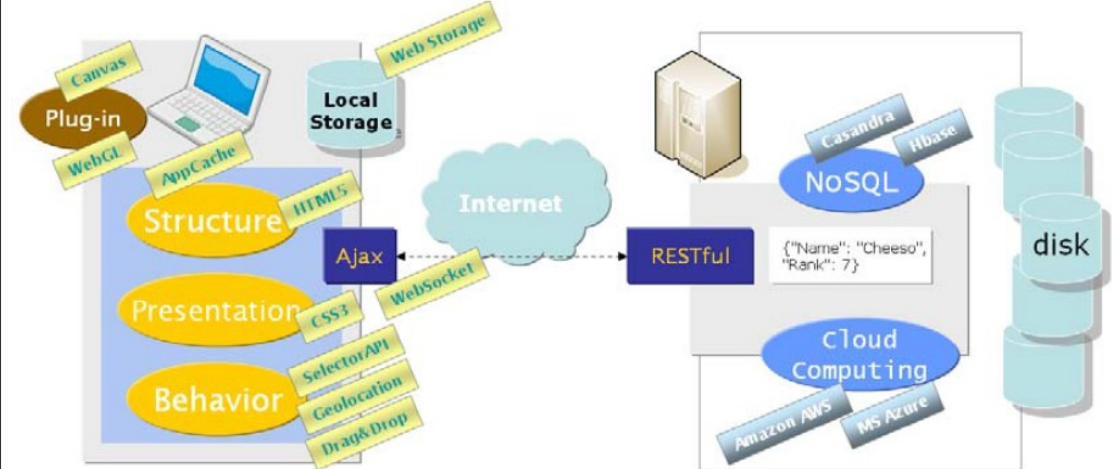
HTML5와 웹개발방법론의 진화 (2/2)

Ajax 시대 (2000년대 후반)



- 2004년 Gmail 과 Google Maps로 Front-end 웹기술 혁신
- 제임스가렛의 웹개발 설계패턴이자 아키텍처인 Ajax 확산
- 백엔드 개발자 : JSON과 같은 데이터기반 응답만 처리로 간단한 웹개발
- 프론트엔드 개발자 : 다양하고 풍부한 UX를 제공하는 웹애플리케이션 개발 용이
- 여전히 웹 서버에 종속적이며 독립적인 웹 애플리케이션 개발은 안됨

HTML5(Web Application) 시대 (2010년대 초반~)

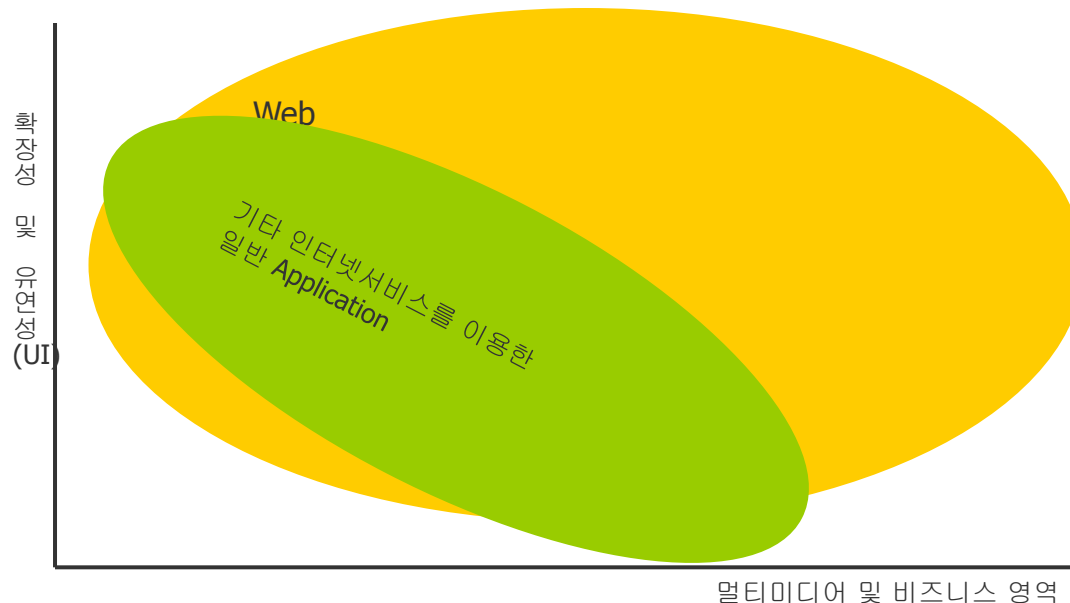


- HTML5 기술로 서버와 독립적인 웹애플리케이션 개발이 가능해짐
- 특히 모바일 환경에서 오프라인 기능과 local DB로 독립적인 개발이 더 가능해짐
- 웹TV 및 Mobile환경에서의 HTML5 기능 최대활용

인터넷속의 웹(WWW, World Wide Web)

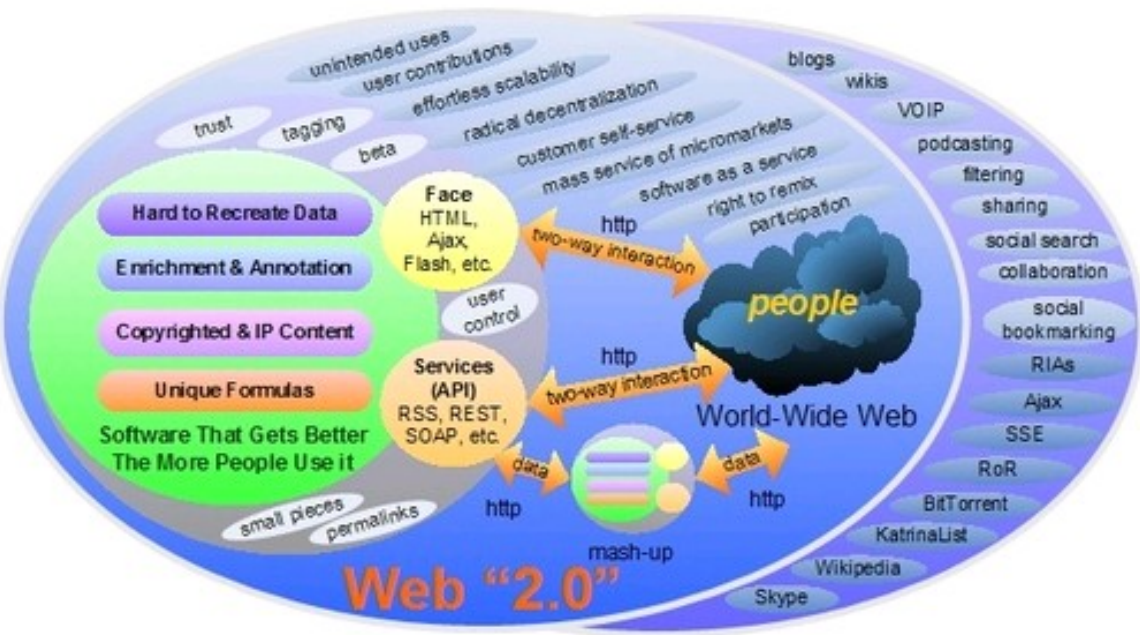
□ 웹(Web 1.0)

- 1990년대 초 인터넷상에 흩어져 있는 정보간의 링크기능을 통한 상호작용을 제공하기 위하여 웹이라는 멀티미디어 인터넷활용 방식으로 빠르게 확산됨.
- 표면적으로 대부분의 인터넷서비스를 웹에서 가능하게 됨으로써 폭발적으로 이용증가.
(웹이 곧 인터넷 : 모든 인터넷은 웹으로 통한다)
- **사용자의 증가 및 정보량의 증가와 더불어 그 한계에 직면**하고 있음.
- **기능의 한계와 정적문서의 한계를 극복하기 위해 문서의 양방향성이 요구됨.**



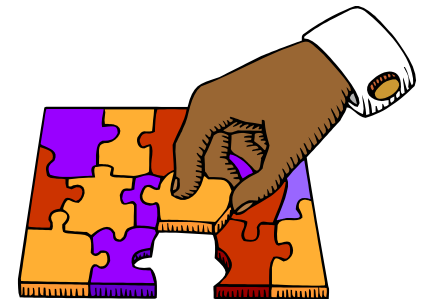
- 웹(Web 2.0) : 차세대 웹, 시맨틱웹

- 사용자의 정보검색 및 상호작용을 용이케 하는 방향으로 진화된 정보접근 환경을 제공.
- **비즈니스모델을 가진 플랫폼(Platform)으로서의 웹.**
- 목적에 맞는 정보를 수집,가공,응용이 가능 (인간의 언어와 의도를 이해하는 검색엔진의 변화)
- 웹 환경보다 정보의 생성, 공유, 소비가 자유로운 인터넷 환경을 통칭하는 의미로 쓰임.



□ 웹3.0으로의 과도기

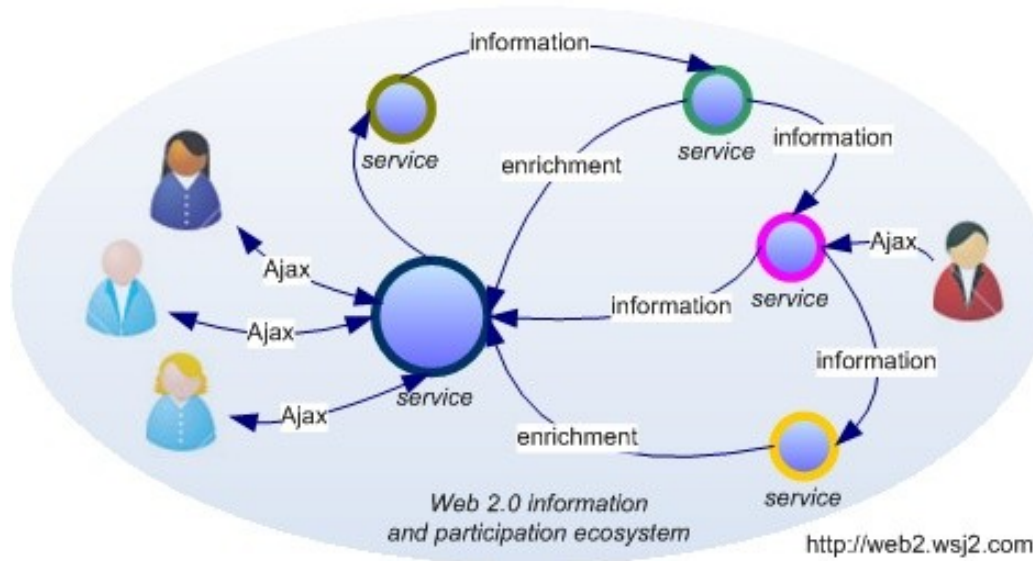
- 웹을 통해 개인과 기업용 SW를 서비스형태로 제공: **웹을 SW의 플랫폼으로 보는 시각 대두**
- 웹의 플랫폼화는 서비스를 제공하고 제공받는 **유통공간**의 의미가 강함
 - 다양한 비즈니스 애플리케이션들이 공유될 수 있는 온라인 마켓플레이스
- **SaaS**(Software as a Service) :
 - '소프트웨어'에서 '서비스웨어'로의 변화를 의미
- Open API, 매쉬업, 위젯
 - 새로운 솔루션의 추가/개발/시험이 가능 & 수많은 협력사와 개발자들이 활동 중
 - 구글 Apps for Domain, 구글 웹 오피스등 기업 대상의 SW를 서비스로 제공 중



웹플랫폼의 등장배경

- ❑ 웹 2.0 서비스들과 Open API의 확산으로 인해 등장
- ❑ 웹 2.0 서비스들을 소프트웨어의 컴포넌트로 이용
- ❑ **기존의 운영체제, 하드웨어 플랫폼에 종속되지 않는 크로스 플랫폼 성격을 가짐**
 - **유기적, 분산적, 사회적 특성을 가짐 (Social Network, Web3.0)**

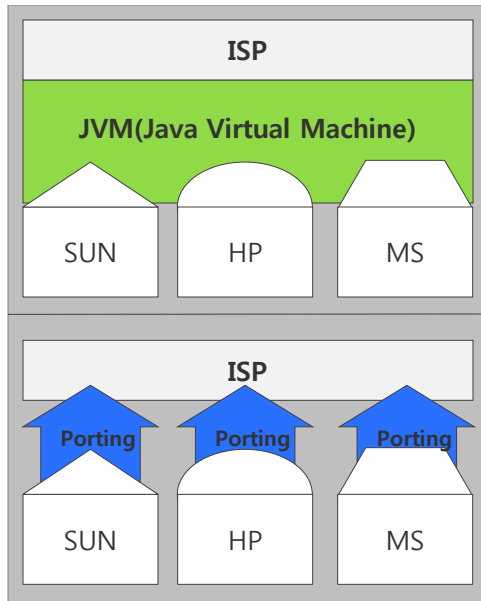
The Web As Platform Organic, Decentralized, Social Software



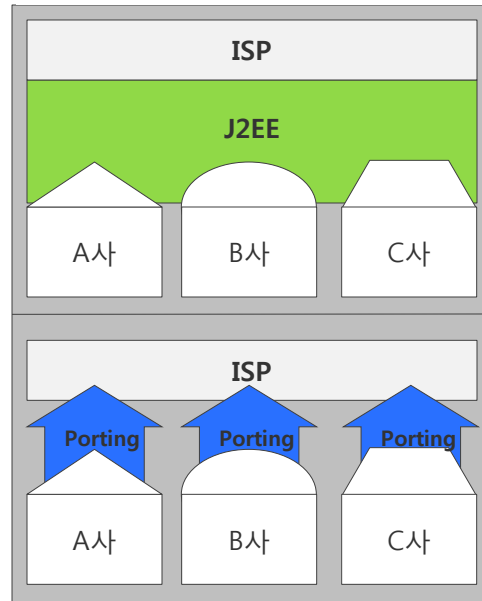
웹플랫폼의 변화

- ❑ 웹플랫폼이란? **플랫폼은 이기종간의 통합을 말한다.**
- ❑ 소프트웨어기술의 발전은 이기종성의 극복에서부터 시작한다.
- ❑ **이기종성을 극복하기 위해 웹표준을 따른다.**

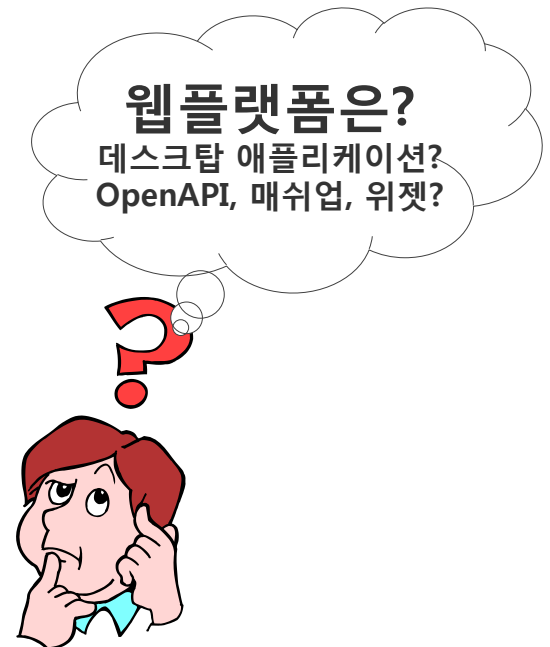
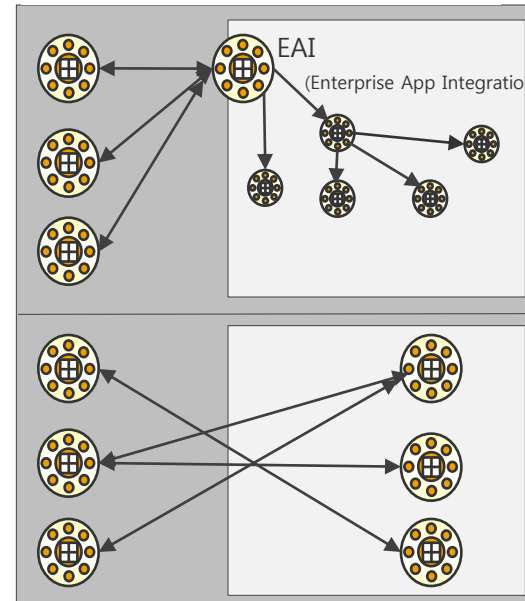
H/W, OS간의 이기종성



기업형 SW 이기종성



Application의 이기종성



웹플랫폼의 구성요소

클라이언트기술

- RIA기술의 등장
- 동적인 기능과 화려한 UI를 웹에서 구현
- Ajax, Flex, XUL, Widget등

컨텐츠기술

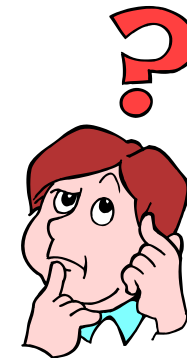
- 기존의 웹서비스는 내부 DB에 국한됨
- 많은 웹2.0서비스들은 자신의 데이터를 공개Open API
- **웹전체를 하나의 데이터로 볼 수 있게 됨**

서버기술

- 다양한 웹개발 언어 및 프레임웍 등장
- LAMP와 같은 오픈소스 등장
- LAMP, ROR, OSS Framework

표준기술

- **웹표준 준수의 확산**
- (x)HTML, CSS, DOM, ECMAScript



웹표준과 Ajax?
기반기술을 바탕으로 이기종성을
극복하기 위함!

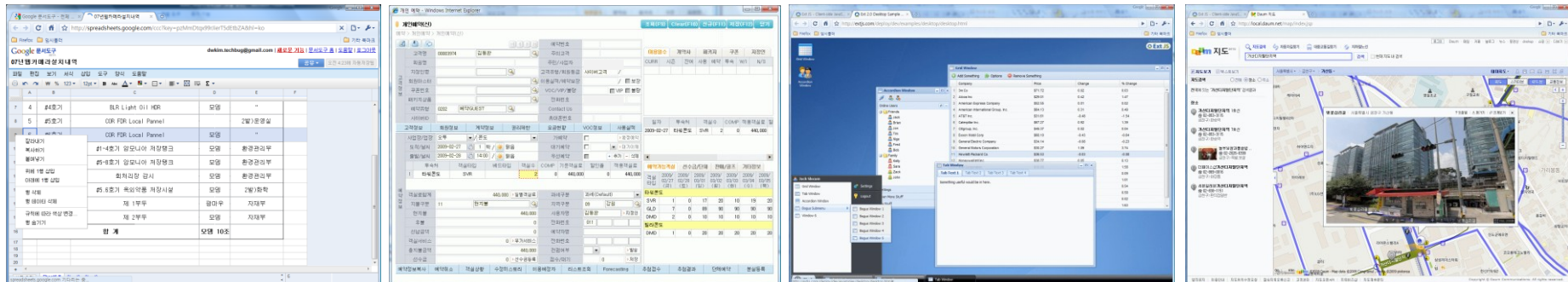
웹애플리케이션

□ 웹애플리케이션이란?

- 기존 C/S환경의 애플리케이션을 웹 브라우저를 통해서 구현
- 웹에 연결되어 작업 지향적(Task-oriented) 소프트웨어

□ 최근 웹기술의 특징

- HTML → Rich UI
- Windows Only → Cross Platform
- Desktop & Office Application → Extended Web Application & Device

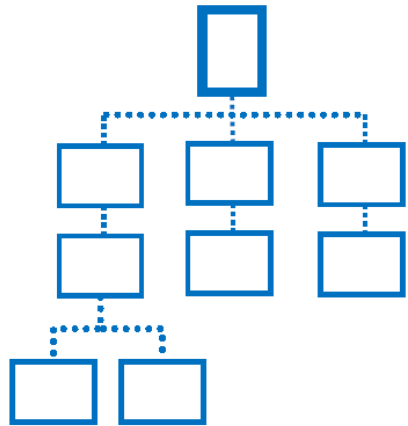


구글 오피스, 구글맵, 웹메신저(<http://www.meebo.com>), 터미널 에뮬레이터(<http://anyterm.org>), Ruby프로그래밍환경(<http://tryruby.hobix.com>), 웹데스크탑(<http://www.extjs.com>)

웹애플리케이션 패러다임의 변화

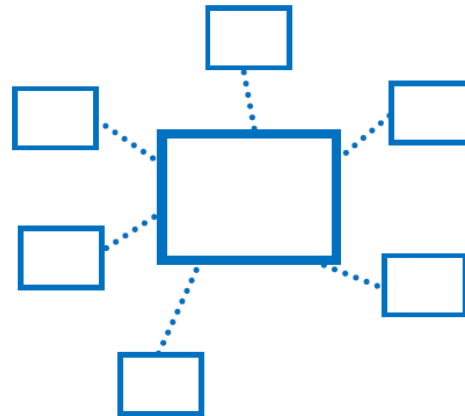
레이아웃 Interaction의 변화

Web Site: User navigates to information



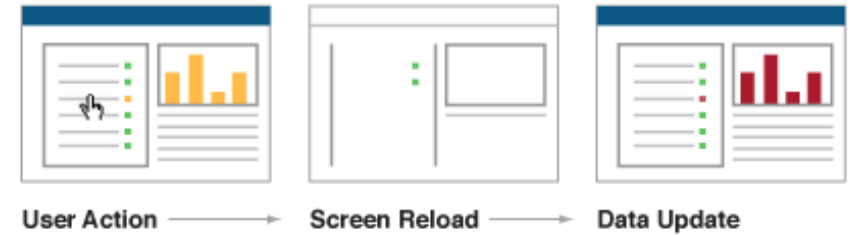
This is about wayfinding...

Web App: Functionality comes to the user



This is about working in one place...

Traditional Web Interaction

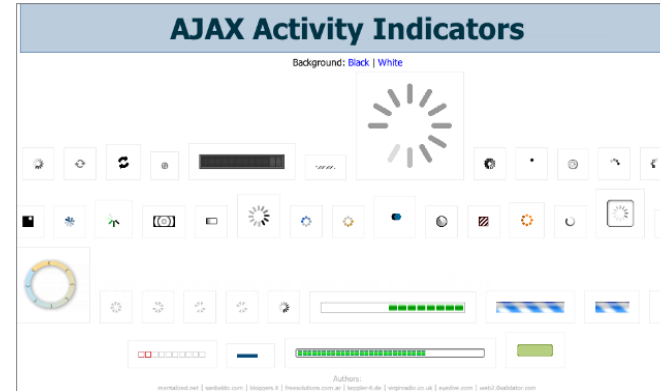


AJAX Interaction

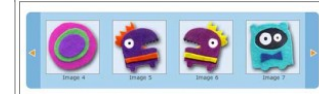


웹애플리케이션 패러다임의 변화

□ 화면 컨트롤(Widget Control)의 변화



Google's Auto-Complete



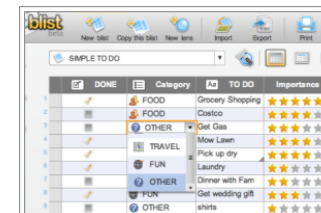
Carousel Prototype carousel widget



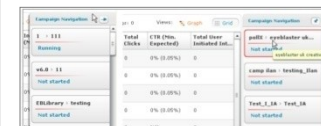
Coverflow MediaEvent Service's Slideflow



Fisheye Apslab Fish-eye Lense



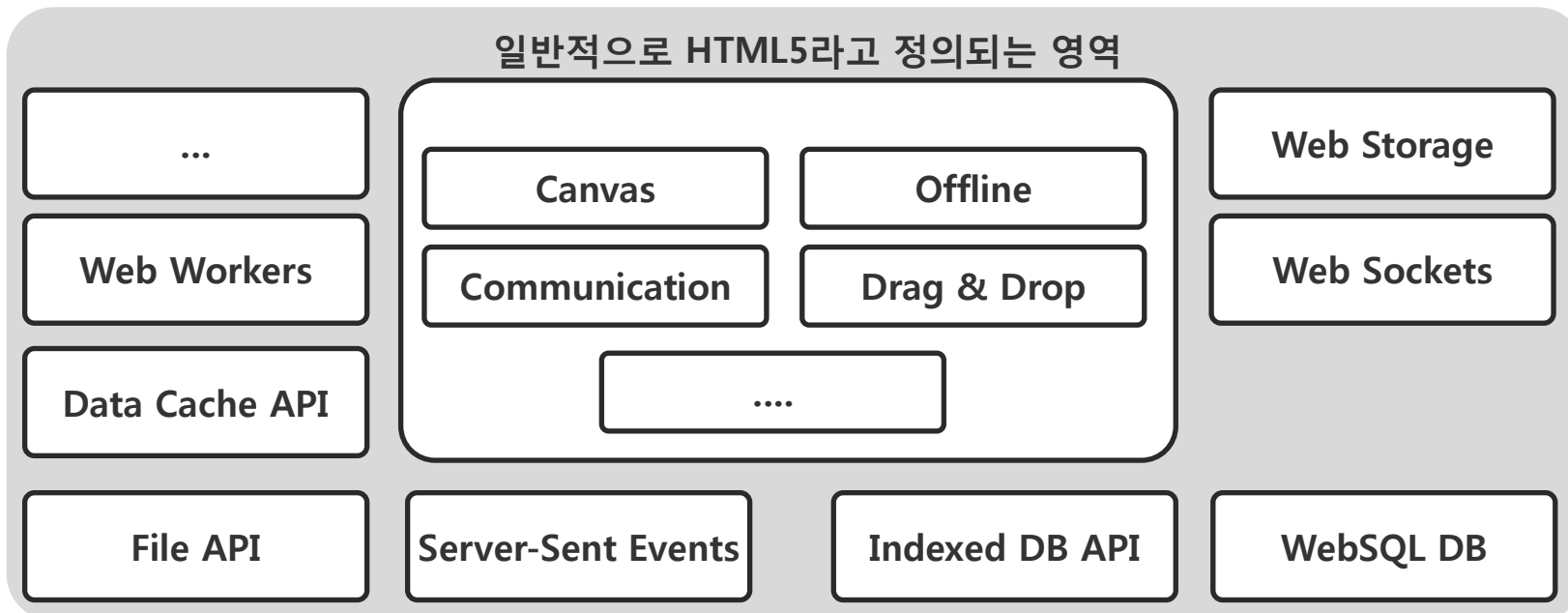
Progress indicator Picnik loading indicator



HTML5의 등장 : 웹애플리케이션의 역할

- ❑ Google(Gmail, Google Maps)의 등장 → 웹표준과 Ajax로 웹에대한 새로운 시각
- ❑ 웹표준
 - 개발자와 디자이너간의 역할 분담
 - 유지보수 및 높은 생산성
 - 뛰어난 웹접근성
- ❑ 웹표준을 기준으로 다양한 웹서비스를 선보이기 시작
- ❑ Ajax의 등장으로 Desktop Platform 밴더(AIR, Flex, Silverlight)들의 웹애플리케이션 접목 시도
- ❑ 구글크롬 및 모질라커뮤니티, 오페라, 사파리진영등 다양한 브라우저 붐물
- ❑ 유럽 : 비IE점유율 50% 육박
- ❑ W3C의 xHTML2.0 실패 → 새로운 마크업 필요성 대두
- ❑ 플러그인을 걷어내고 순수 마크업과 API만으로 웹애플리케이션을 만들자
Of the Web Developer, by the Web Developer and for the Web Developer
→ HTML5의 등장

- ❑ **Web Document** 에서 **Web Application**을 위한 플랫폼으로 진화
- ❑ **HTML5(Open Web Platform)** = HTML5.0 + CSS3.0 + Javascript API
- ❑ **W3C (HTML W/G, WebApps W/G), WHATWG**에서 각각 표준화 진행
 - HTML W/G : HTML자체 표준안, 렌더링 호환성제공 방법, 마크업과 속성을 제공
 - WebApps W/G : Javascript API 기능 제공
 - W3C DAP, OMTP Bondi,JIL, WAC <device>
- ❑ **WHATWG** : W3C에 참여하지 못하는 일반 웹개발자 대상, 샘플코드, 구현권고 등 제공



HTML5 표준문서 : HTML5인것

□ HTML W/G :

HTML5 자체 표준안 제정, 렌더링 호환성 제공방법, 마크업과 속성에 대한 용법 정리, 콘텐츠의 의미적 표현 및 문서간 데이터 알림 등의 스펙 제공

- HTML 5 Specification : A vocabulary and associated APIs for HTML and XHTML
 - 마크업 속성 정의 및 용법에 대한 상세설명 기술
- HTML5 differences from HTML4
- HTML : The Markup Language
 - 웹 개발자(Web Publisher) 및 웹브라우저 개발 회사를 위해 기술된 표준
- HTML + RDFa 1.1 : XHTML2워킹그룹 데이터 표준
- HTML Microdata : 마이크로포맷 (item, itemprop)
- HTML Canvas 2D Context
- HTML5 : Techniques for providing useful text alternatives
 - Alt를 통한 콘텐츠 특성 작성 방법 기술
- Polyglot Markup : HTML-Compatible XHTML Documents
 - HTML5를 XML 로 구문 오류없이 함께 쓰는 방법 제공

HTML5 표준문서 : HTML5가 아닌것

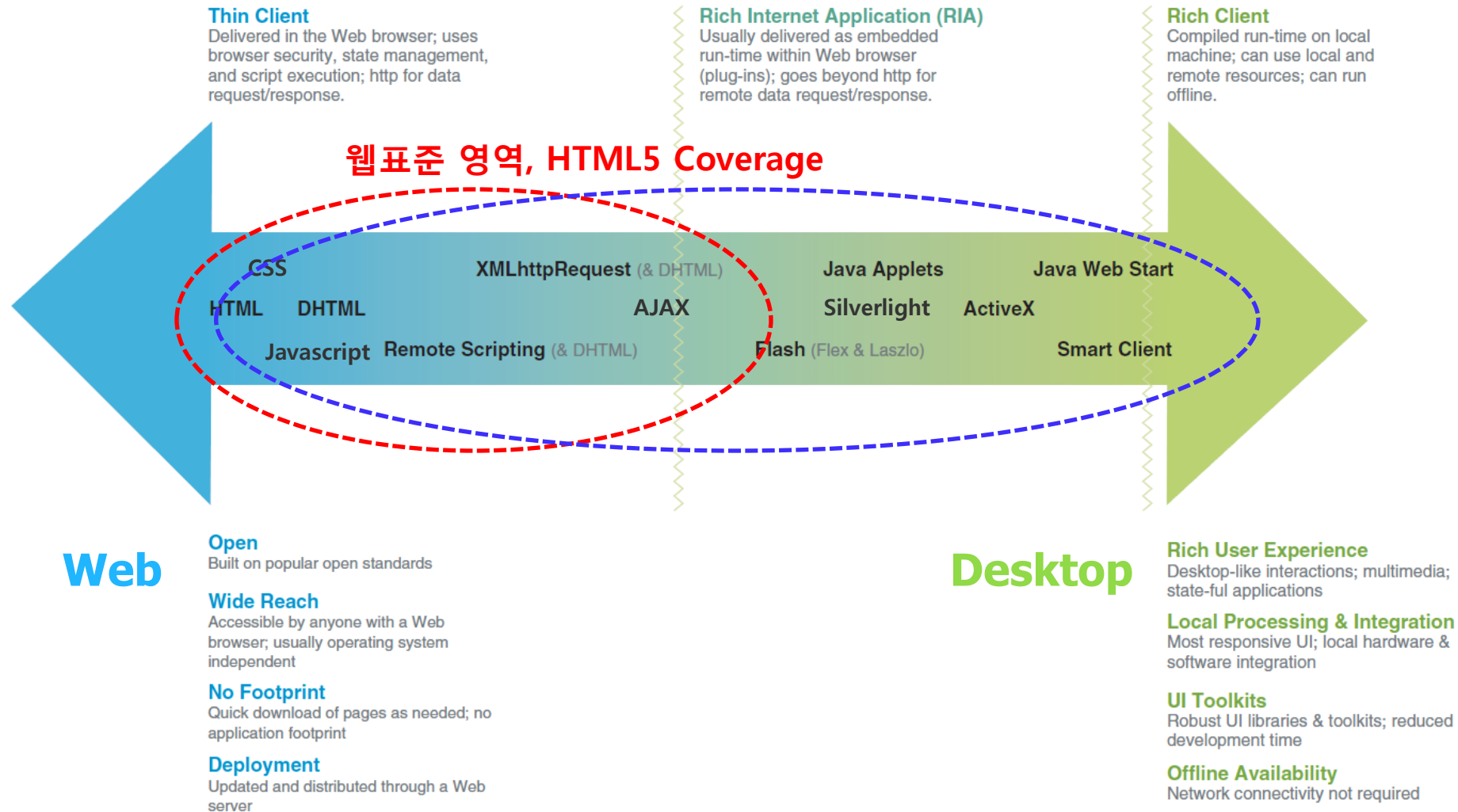
□ WebApps W/G

HTML5 주요 Spec에 포함되어 있거나 차후 분리된 표준안으로서 웹 애플리케이션 개발을 지원하기 위해 만들어진 것으로 HTML5 범주안에 포함된다고 간주되는 것들

- **Server-Sent Events** : Server Push 데이터 정의, HTML5 표준안에서 분리중
- **Communications** : 크로스도메인 XHR
- **Web SQL Database**
- **Web Socket API**
- **Web Workers**
- **그 밖의 표준**

Content-Type, Processing Model, Geolocation API, SVG, MathML 등
WHATWG에서 추진중인 WebSRT (자막), <device>요소를 통한 Camera,USB제어 기능

웹애플리케이션 Front-end 기술



출처 : WEB APPLICATION SOLUTIONS : A Designers Guide : RamirezDesign.com & lukew.com

<http://techbug.tistory.com/151>

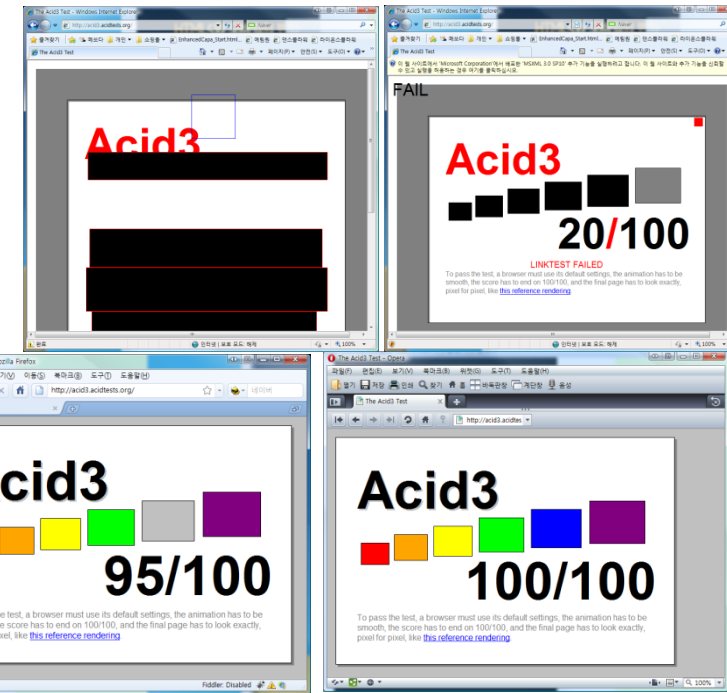
웹 표준

□ W3C의 토론을 통해 나온 권고안(Recommendation)

- 웹의 창시자 팀버너스리 :
" 웹기술발전을 장려하고 관련기술을 잘 운영하기 위한 규약(Standard)을 개발해 WWW(World Wide Web)의 잠재성을 최대한 발휘시키기 위함"
- 웹브라우저 이기종 간의 호환성 필요 :
오래된 브라우저 사용자의 불편함을 고려, 기능이라도 호환 가능하도록 해주는 공통된 표현(Cross Browsing)필요.

□ 웹표준을 지킨다 = W3C의 권고안을 지킨다.

- ACID2(HTML,CSS,PNG),
ACID3(HTML,CSS,DOM,SVG)테스트 (<http://acid3.acidtests.org/>)
- 웹표준 Validate Test
- HTML5 Test
<http://html5.validator.nu/>
<http://html5test.com/>
<http://html5readiness.com/>
<http://www.findmebyip.com/litmus>



웹표준 관리기관

□ World Wide Web Consortium(W3C) : <http://www.w3c.org>

□ 운영현황

- 시작:웹을 발명한 Tim Berners-Lee에 의해 1994년 시작
- 700개 회원사
- MIT(US),ERCIM(France),게이오대(Japan)외 15개 지역사무국 운영

□ 운영방법

- 강력한 표준안 제정과정
- 특허 무료정책
- 공감대, 개방성, 상호운용성 중시
- 40개 이상의 다른 표준단체와 연계작업
- 국제 표준에 맞는 합리적인 운영



□ 주요목표

- 웹문서에서 하나의 '웹'을 지향
- 데이터서비스 : XML, 웹서비스(Web Services), 시맨틱웹(Semantic Web)

W3C의 주요목표

- **XML** : 다수의 플랫폼이 지원, 무상특허로 이용가능
 - 기계 및 인간 모두에 대한 접근성 가능, 국제화 용이
 - XML은 읽기 쉽고, 검증하기 용이하며 애플리케이션이 아닌 데이터로 관리
 - 주요이슈
 - 텍스트 메시징의 문제(바이너리 XML)
 - 웹에서 XML전환이 사실상 실패
- **웹서비스** : 기계-기계 상호교환 (사람-기계 상호교환)
 - XML기반 메시지 처리: 플랫폼 독립적인 모델
 - 확장성을 고려한 프레임워크 : 상호운용성 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 의미있는 웹으로 전환하는것이 목표**



□ 모든 것 위의 하나의 웹

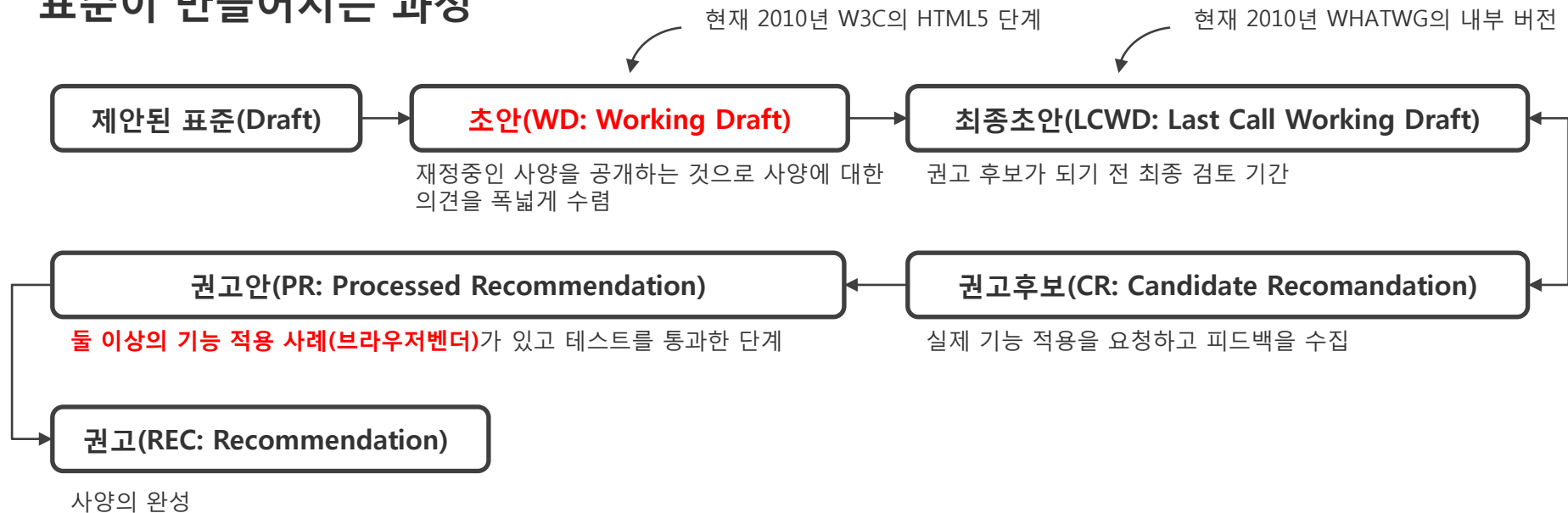
▪ 하나의 웹(One Web)

- 웹기술은 모든 디바이스를 통해 상호교환 가능해야함. 컴퓨터, 임베디드, 사람, 집, 직장, 교통, 산업, 의료분야 전 시스템
- 상호교환 기술 : XHTML, Xforms, CSS, DOM, SVG, SMIL, Voice
- 리치 웹 기술 : Compound Doc Formats, Web Applications
- 모바일 웹기술 : Mobile Web Initiative
- 유비쿼터스 기술 : Embedded Systems, Ubiquitous Web

□ 웹표준의 종류 : (Standard가 존재하지 않으며 Recommendation이 최상위개념)

- 제안된 표준(Draft)
- **작업하는 표준(Working Draft, WD)**
- 확정될 권고안(Candidate Recommendation, CR)
- 확정된 권고안(Recommendation)

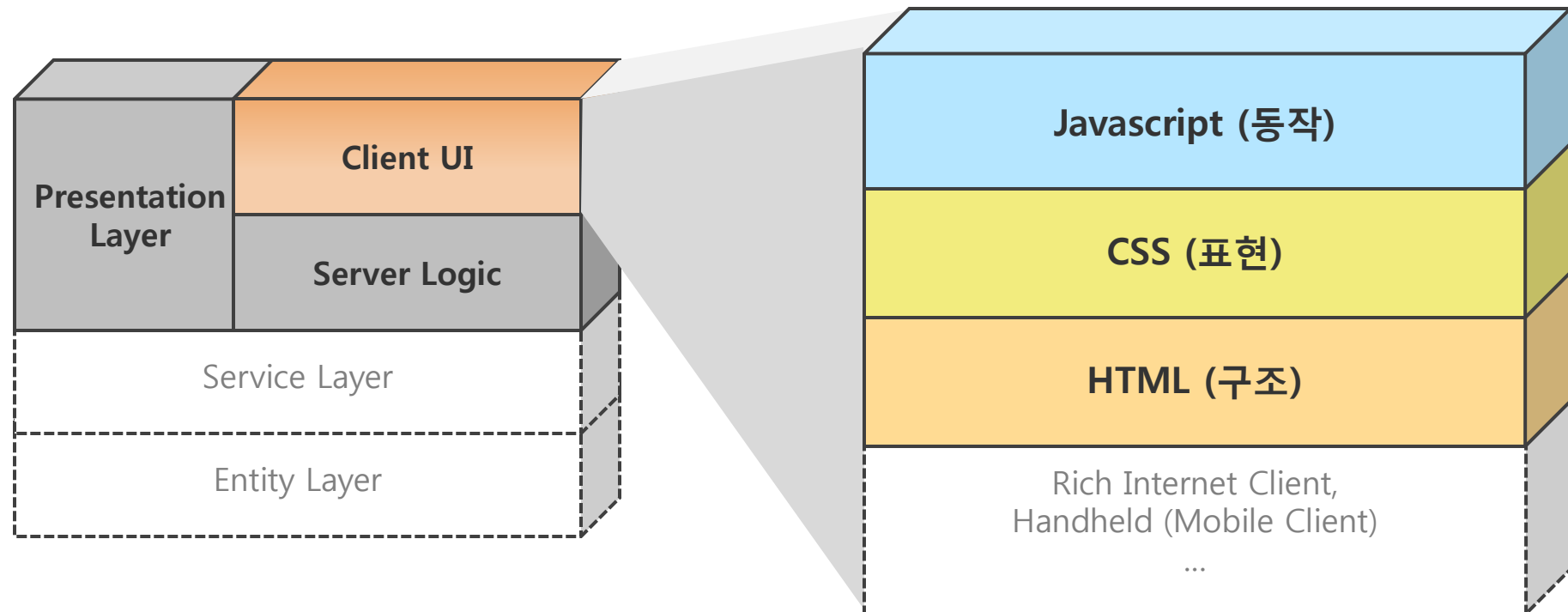
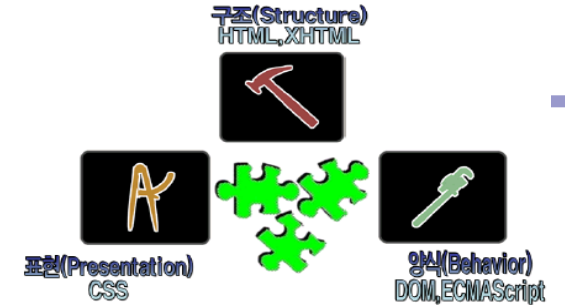
□ 표준이 만들어지는 과정



- ❑ **(X)HTML(eXtensible Hypertext Markup Lanuguage)**
 - 웹페이지를 구성하고 표현하는 기본언어
- ❑ **CSS(Cascading Style Sheets)**
 - 웹페이의 호환성 유지 및 다양한 액세스 기술을 사용
- ❑ **XML(eXtensible Markup Language)**
 - HTML이나 CSS로서 표현되지 못하는 영역을 DTD를 이용하여 정의하여 사용자 정의 태그를 생성하여 제작할수 있는 메타 마크업언어.
 - 이기종간의 통신에 이용.
- ❑ **DOM (Document Object Model)**
 - 콘텐츠, 구조, 문서 스타일을 프로그램과 스크립트가 동적으로 접근하고 수정할 수 있는 플랫폼이며 언어 중립적인 인터페이스
 - W3C DOM, MS DOM
- ❑ **ECMAScript(Javascript)**
 - W3C표준으로 제정된 것이 아님.
 - 넷스케이프사가 ECMA 표준기구로 채택

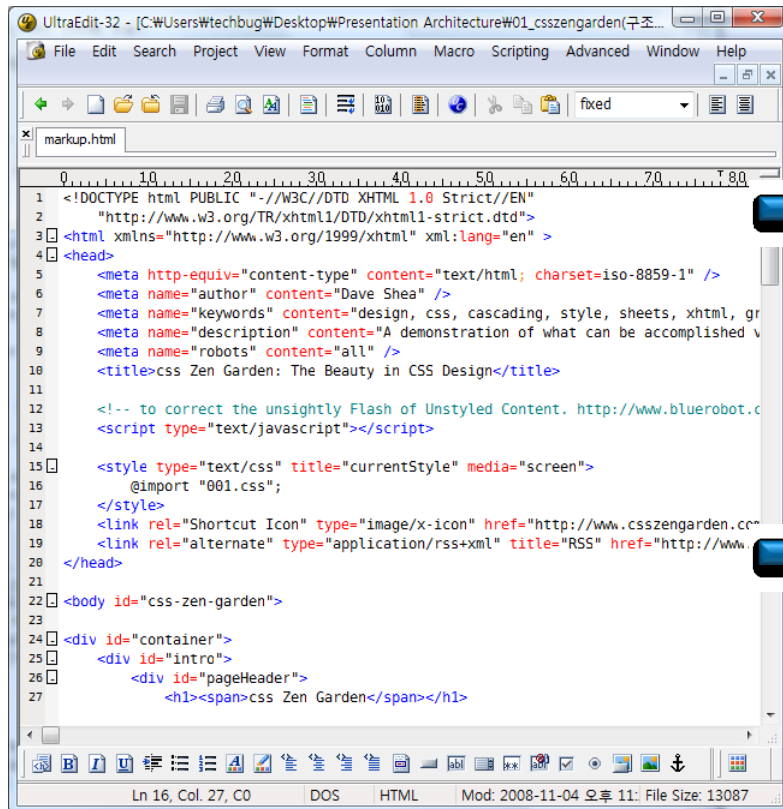
웹표준이란 : 구조,표현,동작의 분리

- AJAX구성요소 == 웹표준구성요소
- 웹표준 개념 : 구조,표현,동작의 분리



구조와 표현의 분리 (xHTML + CSS)

- Standard-based Presentation using : 웹표준의 기준?
- 문서구조(Markup) 과 디자인요소(CSS)의 분리
 - 예제) <http://www.csszengarden.com>



```

1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
2 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
3 <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" >
4 <head>
5 <meta http-equiv="content-type" content="text/html; charset=iso-8859-1" />
6 <meta name="author" content="Dave Shea" />
7 <meta name="keywords" content="design, css, cascading, style, sheets, xhtml, gr
8 <meta name="description" content="A demonstration of what can be accomplished v
9 <meta name="robots" content="all" />
10 <title>css Zen Garden: The Beauty in CSS Design</title>
11
12 <!-- to correct the unsightly Flash of Unstyled Content. http://www.bluerobot.c
13 <script type="text/javascript"></script>
14
15 <style type="text/css" title="currentStyle" media="screen">
16 @import "001.css";
17 </style>
18 <link rel="Shortcut Icon" type="image/x-icon" href="http://www.csszengarden.co
19 <link rel="alternate" type="application/rss+xml" title="RSS" href="http://www.
20 </head>
21
22 <body id="css-zen-garden">
23
24 <div id="container">
25 <div id="intro">
26 <div id="pageHeader">
27 <h1>css Zen Garden</span></h1>
    
```



001.css



001.css



003.css



004.css

□ 겉모양은 같으나 HTML은 완전히 다름

Figure 1-1 shows two side-by-side web browser windows. Both windows display a list of items: 1. 서론 (Introduction), 2. 본론 (Main Body), and 3. 맺음말 (Conclusion). The left window has a red arrow pointing from the '1. 서론' item to the '1. 서론' item in the right window, indicating a selection or focus change.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=euc-kr">
<title>그냥 표현한 것</title>
</head>
<body>
<br>
<font size="4" color="darkred"><b>1. 서론</b></font><br>
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~<font size="2" color="black">1. 개념정의</font><br>
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~<font size="2" color="black">2. 필요성</font><br>
<font size="4" color="darkred"><b>2. 본론</b></font><br>
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~<font size="2" color="black">2.1 보편적 디자인원칙</font><br>
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~<font size="2" color="black">2.2 접근성원칙</font><br>
<font size="4" color="darkred"><b>3. 맺음말</b></font><br>
</body>
</html>
```

구조와 동작의 분리 (xHTML + Javascript) 1/2

□ 문서(Markup)과 이벤트를 이용한 동작(Javascript)의 분리 (unobstructive)

- 예) ``이 아닌 DOM 객체를 직접 핸들링

□ Javascript Library(Ajax 프레임웍) 지원

- HTML DOM객체에 동적으로 이벤트 생성하여 동작시킴
- 버튼 클릭에 의한 통신부분처리
- DataGrid, Menu, Tree, Accordion, Rich Form, Drag & Drop

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
<head>
<script type="text/javascript">
//
    window.onload = function(){
        var btnObj = document.getElementById("iambutton");
        if(btnObj != null) {
            btnObj.onclick = function(){
                alert("클릭이벤트입니다.");
            }
        }
    }
//]]&gt;
&lt;/script&gt;
&lt;/head&gt;
&lt;body&gt;
    &lt;input type="button" id="iambutton" value="버튼입니다" /&gt;
&lt;/body&gt;
&lt;/html&gt;</pre>
</div>
<div data-bbox="554 511 829 908" data-label="Image">
<img alt="Screenshot of a web browser showing a button click event alert." data-bbox="554 511 829 908"/>
  A screenshot of a Windows Internet Explorer browser window. The address bar shows a local file path. The main content area displays a button labeled "버튼입니다". An alert dialog box is open in the center of the screen, displaying a yellow warning icon and the text "클릭이벤트입니다." (Click event occurred). The dialog has a blue "확인" (OK) button at the bottom. The browser's status bar at the bottom indicates "컴퓨터 | 보호 모드: 해제" and "100%" zoom.
</div>
<div data-bbox="3 939 236 981" data-label="Page-Footer">
<p>과정명: HTML5<br/>Copyright © 2011 NEXTREE SOFT, all rights reserved</p>
</div>
```

구조와 동작의 분리 (xHTML + Javascript) 2/2

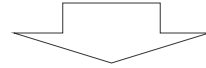
□ 뷰(xHTML)에서 로직(javascript)분리하기

```
<div class="importButton" onclick="importData('datafeed3.xml', mytextbox.value);" /> </div>
```

view

controller

model



```
<div class="importButton" id="dataFeedBtn" > </div>
```

```
function importFeedData(event){
    importData("datafeed3.xml", mytextbox.value);
}
```

```
var dfBtn=document.getElementById("dataFeedBtn");
dfBtn.onclick = importFeedData;
```

다이나믹한 표현처리 (CSS + Javascript)

□ CSS 제어를 통한 다이나믹한 표현처리

- 객체.style.[css property명] = "값"
- offsetTop, innerHTML 등을 이용한 표현 변경

□ Javascript Library(Ajax 프레임웍) 지원

- DataGrid, Menu, Tree, Accordion, Rich Form, Drag & Drop
- Effect : Animation (Style의 변화)
- Drag & Drop
- Selector (CSS Selector : Dom Query)

Javascript와 웹표준

□ 뷰에서 로직 분리하기

- 디자이너는 HTML/CSS만, 개발자는 javascript만 관리 (단, 협업이 중요)
- 화면의 추가, 배치등이 용이

```
<html>
<head> <title>HTML안 로직<title>
<link rel="stylesheet" type="text/css" href="musical.css" />
<script type="text/javascript" src="musical.js"> </script>
</head>
<body>
<div id="keyboard" class="musicalKeys">
  <div class="do musicalButton"></div>
  <div class="re musicalButton"></div>
  <div class="mi musicalButton"></div>
  <div class="fa musicalButton"></div>
  <div class="so musicalButton"></div>
  <div class="la musicalButton"></div>
  <div class="ti musicalButton"></div>
  <div class="do musicalButton"></div>
</div>
<div id="console" class="console"> </div>
</body>
</html>
```



```
var notes= new Array("do","re","mi","fa","so","ra","ti","do");

function makeKeyboard(el){
  for(var i=0, iLen=notes.length; i < iLen; i++){
    var key= document.createElement("div");
    key.className=notes[i]+" musicalButton";
    ....
    el.appendChild(key);
  }
}

<html>
<head> <title>HTML안 로직<title>
<link rel="stylesheet" type="text/css" href="musical.css" />
<script type="text/javascript" src="musical.js"> </script>
</head>
<body>
<div id="keyboard" class="musicalKeys"> </div>
<div id="console" class="console"> </div>
</body>
</html>
```


웹표준의 장점

□ 마크업(Markup) 용량이 줄어듦

- 코드량이 줄어드는것은 페이지 로딩시간이 빨라짐을 의미.
- 서버의 성능을 효율적으로 관리, 처리능력을 극대화하여 비용절감.

□ 콘텐츠와 문서모양의 분리

- CSS를 사용하여 문서의 유지보수가 쉬워짐.

□ 높은 웹접근성

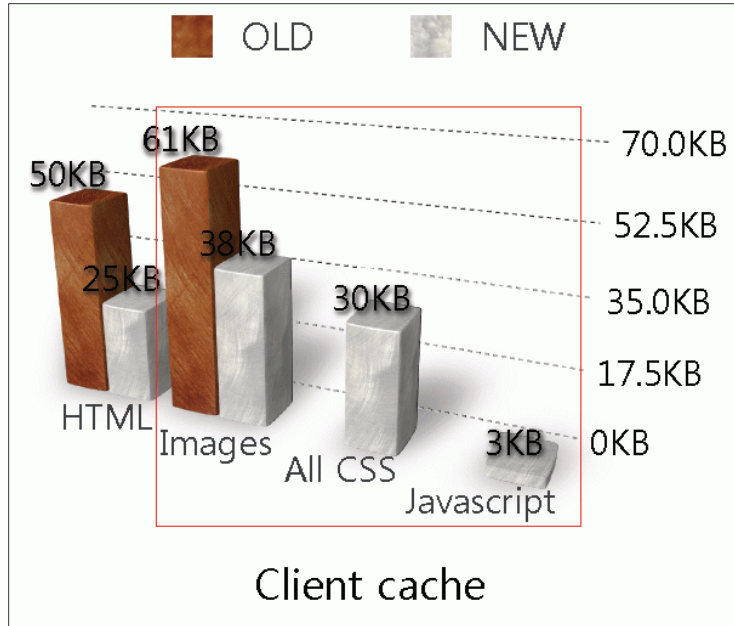
- 대부분의 웹브라우저, 휴대폰, PDA, 장애인지원용 소프트웨어등 여러 개의 단말기 적용
`<link href='css/480.css' media='only screen and (max-width: 480px), only screen and (max-device-width: 480px)' rel='stylesheet' type='text/css'>`
`@media only screen and (max-width: 480px), only screen and (max-device-width: 480px) {`
`body { background: pink; }`
`}`

□ 향후 호환성을 보장받음 : 상호호환성(Cross Browsing)

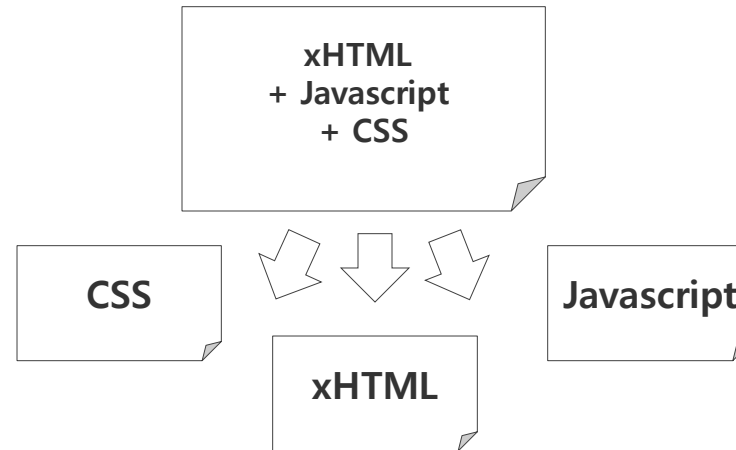
- 오래된 브라우저 사용자들의 불편함을 고려
- 다른 기종 혹은 플랫폼에 따라 달리 구현되는 기술을 비슷하게 만듦과 동시에 어느 한쪽에 최적화되어 치우치지 않도록 공통요소를 사용하여 웹페이지를 제작하는 기법.

분리의 효과

□ 구조와 표현, 동작의 분리



OLD	NEW	절감율
112KB	62.9KB	44%



출처 : 정보통신 접근성 향상 표준화 포럼 (<http://www.iabf.or.kr>)

웹표준 권고안에 따른 웹사이트 개발방법론

□ 웹표준을 지켜라

- XHTML1.x나 HTML4.x 표준에 맞추어진 문서는 99% 접근성이 좋은 사이트
- table구조는 div로, font,b등의 태그는 CSS로 사용하라.

□ 구조와 표현, 동작을 분리하라

- 빠른 사이트의 로딩속도.
- 코딩과 유지보수의 효율성이 높아짐.
- 구조화된 마크업에 따른 다양한 디자인 지원

□ 최소한의 디버깅을 거쳐라

- HTML과 XHTML, CSS, DOM,JAVASCRIPT가 표준문법을 사용했는지 확인
- <http://validator.w3.org> - 브라우저 유효성 검사
- <http://jigsaw.w3.org/css-validator> - CSS 유효성 확인
- <http://www.stg.brown.edu/service/xmlvalid> - XML 유효성 확인
- <http://www.iabf.or.kr/web/kadowah.asp> - 한국형 접근성 평가도구

□ 효율적인 웹개발 방법론을 가져라

- HTML에 표현과 내용을 분리하면 기획자와 디자이너, 개발자가 같은 시간내에 같은 일을 하는것이 가능해 짐.

표준화 참여방법

□ 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/>

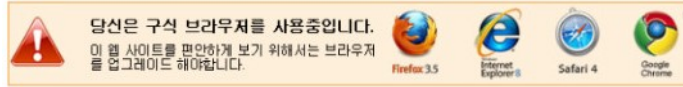
□ 웹표준화 커뮤니티

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

쉬어가며 : 최근 웹에선 어떤 일이?

개발자 좀
살려주세요 **제발!** **no more**

브라우저 업그레이드가 한 사람을 살립니다.



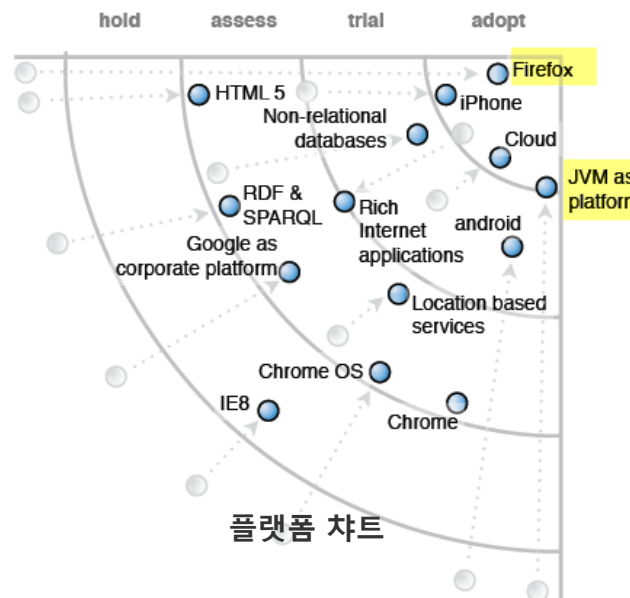
새로운 웹 전쟁

- 제 1브라우저 전쟁 = 벤더간 경쟁 (Netscape vs. Internet Explorer)
- 제 2브라우저전쟁 = 표준안 경쟁 (WHATWG vs. W3C)

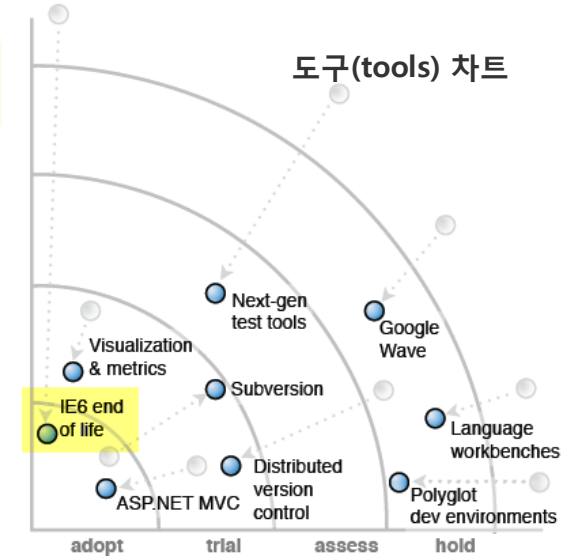


스티브 잡스 어도비 플래시 맹비난

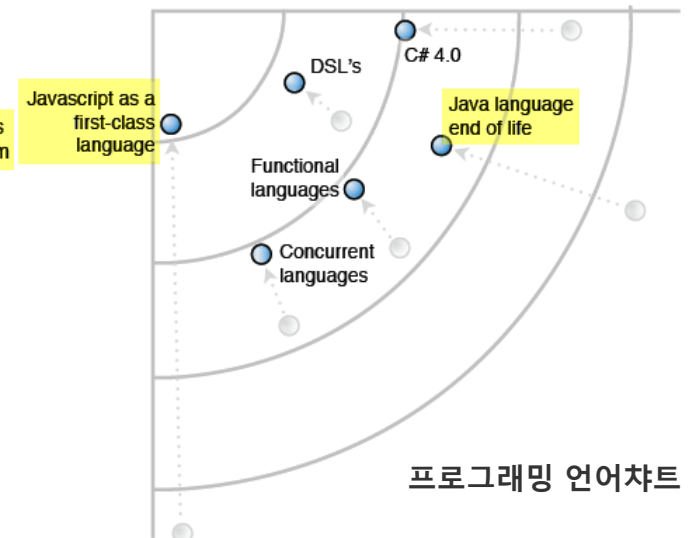
그는 앞으로 HTML5가 대세이며, 이에 따라 플래시없이도 브라우저 상에서 멀티 미디어를 구현할 수 있게 될 것이라고 주장했다.



플랫폼 차트



도구(tools) 차트



프로그래밍 언어차트