IE, 넷스케이프, 오페라, 사파리와 함께하는

# Cross Browsing 가이드

# 한국소프트웨어진흥원 공개SW지원센터

# 목 차

#### 1. 들면서

# 2. Cross Browsing 이란?

- 2.1 Cross Browsing의 정의
- 2.2 웹브라우저간 특이성
- 2.3 접근 호환성
- 2.4 표준이란 무엇인가
- 2.5 W3C 표준의 내용

# 3. Cross Browsing 방법론

- 3.1 HTML
- 3.2 CSS
- 3.3 DOM과 Javascript
- 3.4 Plugin

#### 4. 고급 디버깅 기법

- 4.1 웹브라우저 알아내기
- 4.2 레이아웃 방식의 변경
- 4.3 비IE 사용자를 위한 가이드
- 4.4 웹페이지 디버깅 방법

#### 5. 맺음말

#### Appendix.

- 1. 최신 웹브라우저별 기능 비교 차트
- 2. 브라우저별 HTML 표준 지원 비교표
- 3. 브라우저별 CSS 표준 지원 비교표

본 가이드의 모든 내용은 공개 및 인용을 할 수 있습니다. 그러나, 저작물의 출처(URL)를 밝히지 않거나 허락 없이 상업적 인 목적으로 사용해서는 안됩니다. 내용에 대한 권한은 저자와 한국소프트웨어진흥원에 있습니다.

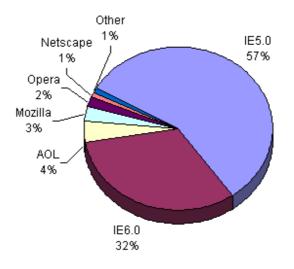
## 글 / 윤 석 찬 (다음커뮤니케이션)

- 1999 부산대 및 동대학원 졸업 (M.S)
- 1994~1998 웹코리아(WWW-KR) Member
- 1996~1997 W3C SVG W/G Member
- 1999~2003 나인포유 부설 연구소
- 2004~ 다음커뮤니케이션 부설 미디어연구소
- 2002~ 한글 Mozilla 프로젝트 리더

# 1. 들면서

1993년 4월 22일 미국 일리노이 대학에서 일단의 학생들이 개발한 모자이크(Mosaic)라는 작은 공개 소프트웨어 웹브라우저는 오늘날 웹이 전 세계에 영향을 끼치게 하는 혁명적인 첫 출발이었다. 이 웹브라우저의 근본 아이디어를 기초로 마이크로소프트(이하, MS)와 넷스케이프사 등에서 개발한 유수의 웹브라우저 들이 나와 각축을 벌이고, 이제는 넷스케이프 네비게이터(이하, NN)이 시장 선점에 실패하면서 MS사의 인터넷 익스플로러(이하 IE)가 시장 지배적인 위치에 들어서 있다. 브라우저 시장점유 전쟁 동안 서로간의 웹브라우저에 배타적인 기술을 도입하던 나머지 똑 같은 웹페이지가 다르게 보이고 서로에서 구현하지 못하는 기술 때문에 많은 혼란을 겪어 왔다. 이러한 상호 호환 브라우징 (Cross-Browsing)의 미성숙으로 말미암아 웹기술이 혼란 상태에 있어 왔던 것이 사실이다.

현재에는 MS가 IE로 시장 지배력을 넓히고 있는 이 시점은 더 이상 웹브라우저가 신기한 도구이지 않기 때문에, 다양한 인터넷 환경에서 어떠한 웹브라우저가 가장 최적의 구현을 제공하느냐가 관건이 되었다. <그림1>은 현재의 웹브라우저 시장 점유율을 보여 주는 것으로 90% 정도를 IE가 차지하고 있다.



<그림 1. 전 세계 웹브라우저 점유율 분표도, 2003. 10>

그러나 넷스케이프사가 자사의 웹브라우저 소스를 공개 소프트웨어로 전환시키면서 탄생한 모질라(Mozilla)는 전 세계 개발자들의 노력에 힘입어 1.5버전을 내놓았으며 넷스케이프산는 이를 토대로 Netscape 7.1 (이하, NS)을 출시하는 성과를 거두기도 했다. 또한, 노르웨이의 Opera 브라우저는 가볍고, 각종 OS 플랫폼과 표준 호환성이 뛰어난 기능을 무기로 시장을 개척해 나가고 있다. 뿐만 아니라 유닉스 기반 오픈소스 프로젝트 KDE(K Desktop Environment)에서 개발한 KHTML 브라우징 엔진은 애플이 사파리 브라우저에 채택되었고

#### Cross Browsing 가이드

PDA 및 Embedded Linux 등 소형 기기에 탑재될 수 있는 가능성으로 웹 브라우저의 다양한 엔진 전쟁이 예고 되고 있다.

특히, 전 세계적인 웹 기술 표준을 주도하고 있는 W3C의 HTML4.0, XHTML, CSS1/2 등의 구현 스펙이 매우 상세하고 이를 지원하는 브라우저들이 계속 늘어 남에 따라 더 이상 웹페이지가 다르게 보이거나 동작되지 않는 현상은 거의 사라지게 되었다. 다만 웹브라우저간 이종 기능이 아직은 상존하고 있고 예전에 개발되어 사용된 오래된 브라우저 사용자들의 불편함을 고려해 같은 기능이라도 호환 가능하도록 해주는 Cross Browsing을 위한 웹페이지 제작이 요구되고 있는 것이다.

이러한 Cross Browsing이 흔히 버전 호환성 유지(backward compatibility)와 혼돈하는 경우가 있다. 일반적으로 프로그램을 개발할 때, 버전이 올라가면 갈수록 새로운 기능을 추가하고 이전 기능은 폐기하게 된다. 그러나 사용자의 측면에서는 예전 기능을 계속 유지해 주어 개발 호환성을 유지해 줄 필요성이 생긴다. 버전 호환성 유지는 바로 이러한 경우에 필요한 것으로 웹브라우저의 경우, 예전에 사용되는 기능이나 태그를 표준 태그로 치환해 주는 것이다. 이를통해 대부분의 웹개발자들은 이전 지식에 따라 웹페이지를 코딩해 주어도 그대로 구현되는 것으로 생각하게 되는 것이다. 그러나, 버전 호환성 유지는 웹브라우저의 벤더에 따라 지원 가능정도가 약해 질 수가 있다. 따라서, 표준에 따라 브라우저와 관계없이 웹페이지 기능을 구현하려는 Cross browsing과는 다른 개념이다.

# 2. Cross Browsing 이란?

#### 2.1 Cross Browsing의 정의

웹페이지의 상호 호환성(Cross Browsing) 구축에 대해 이야기하면 기본적인 오해가 있다. 그 것은 바로 이것이 모든 웹브라우저에서 100% 똑 같이 보이도록 만드는 것이라는 생각이다. 작게는 1990년대 후반 Netscape사와 Microsoft사의 Browser War 기간 동안 일어난 브라우저의 비호환성을 억지로 끼워 맞추려는 기법 정도로 치부되는 것이다.



<그림2. 역사 속의 다양한 웹브라우저>

그러나 OS가 다르고 글꼴이 다르며, HTML을 렌더링(Rendering)하는 엔진이 다르기 때문에 모든 웹브라우저에서 100% 똑같이 보이게 하는 것은 가능하지 않다. Cross Browsing이란 적어도 표준 웹기술을 채용하여 다른 기종 혹은 플랫폼에 따라 달리 구현되는 기술을 비슷하게 만듦과 동시에 어느 한쪽에 최적화되어 치우지지 않도록 공통 요소를 사용하여 웹페이지를 제작하는 기법을 말하는 것이다. 또한, 지원할 수 없는 다른 웹브라우저를 위한 장치를 구현하여 모든 웹브라우저 사용자가 방문했을 때 정보로서의 소외감을 느끼지 않도록 하는 방법론적 가이드를 의미하는 것이다. 이는 인터넷 환경 자체가 일반 테스크톱 웹브라우저 뿐만 아니라 모바일, 임베디드 기기, 홈네트워킹용 장비 등 아주 다양한 인터넷 환경이 존재하며, 일반인, 장애자, 노약자, 어린이 등 다양한 사용자가 존재하기 때문이다.

즉 웹 서비스의 모든 잠재 사용자들이 사이트에 접근할 수 있어야 한다는 것은 매우 중요한 요소라는 것이다. 웹 개발자나 디자이너는 이러한 권고를 적용하지 않을 필요가 있는지 신중히 고려해야 한다. 적용하지 않는 경우에는 포괄적인 접근을 허용할 수 있도록 디스플레이 기능 옵션을 제공해야 한다.

#### 2.2 접근 호환성

Cross Browisng의 범주에는 레이아웃 및 기술적 공통성을 추구하는 면이 있는 가 하면, 일반

적이지 않는 웹사용자에 대한 지원이라는 포괄적인 의미도 함축하고 있다. 예를 들어, 청각 장애자나 시각 장애자가 웹페이지를 보기 위해 필요한 것들이나 어린이 노약자를 위한 배려 같은 것들이 그것이다.

이러한 기능 옵션에 대한 중요한 사항은 웹 사이트를 기획 운영하는 웹마스터와 웹디자이너 및 개발자들이 http://www.w3.org/TR/WAI-WEBCONTENT/ 에 제시되어 있는 'W3C web accessibility initiatives'의 규정한 지침에 유의해야 한다.

## 1) 텍스트

- ▶ 핵심 정보는 반드시 텍스트/HTML 포맷으로 제공되어야 한다. 특히 Flash 같은 것으로 전체화면을 구성하거나 메뉴를 구성하는 것은 피해야 한다. 만약 꼭 사용해야 한다면 비 Flash 버전을 만들어야 한다.
- ▶ 텍스트는 반드시 사용된 배경색에 대해 뚜렷이 대비되는 색으로 표시되어야 한다. (다양한 환경의 256 COLOR 지원 그래픽 카드에서 식별 가능여부가 테스트 되어야 한다)
- ▶ 텍스트 색상은 텍스트를 표시하는 곳에서 사용자가 원하는 색상을 선택할 수 있으므로, 색상별로 별도의 의미를 함축하지는 않는다.

#### 2) 폰트 설정(Font)

- ➤ 글자에 대한 형식은 <font> Tag를 사용하기 보다는 CSS을 통해 지정해서 사용한다. HTML4.0에서는 FONT를 사용하는 것을 추천하고 있지 않다.
- ➤ CSS에는 일반적으로 사용 가능한 글자꼴을 Face 속성에서 지정해야 한다. 예를 들면, 굴림, 굴림체, 돋움, 돋움체 등 Arial, Helvetica, Times New Roman등이 있다. 또한, 가변폭과 고정폭의 글꼴 선택에 있어 글자의 크기를 사용자가 임의로 조정할 수 있도록 가변폭 글꼴을 우선한다.
- 영문의 경우 모두 대문자로 표기하거나 이탤릭체를 과도하게 사용하는 것은 피해야 한다.민줄 친 글자는 하이퍼링크와 혼동될 우려가 있으므로 사용을 피한다.
- ▶ 색상 속성은 인쇄 시 나타나지 않으므로 흰색이나 지나치게 밝은색으로 설정하지 않으며, 쉽게 읽을 수 있도록 배경색과 대비가 되어야 한다. 특히, 'Color'가 특정 의미 부여의 유일한 방법이어서는 안 된다.
- ▶ 어떤 정보가 특정 글꼴로 표현되어야 한다면, 해당 정보는 이미지로 표현되어야 하고 텍스트 형식의 ALT 값을 제공해야 한다. 정보를 표현하는데 이미지를 사용하는 것은 최소화해야 한다.

#### 3) 테이블 (TABLE)

▶ 브라우저에 따라, 특정 사용자의 경우에는, 복잡한 테이블을 생성하는 것이 어렵거나

레이아웃이 틀려 보이는 경우가 매우 많다. 따라서, <TABLE> 태그 방식의 레이아웃 보다는 <DIV>와 CSS과 접목된 레이아웃 방식으로 변경하도록 노력한다.

- ▶ 웹페이지 내에 테이블을 아예 사용하지 않는다는 정책을 고집하는 것은 현실적으로 불 가능하므로, 최소한 디자이너는 복잡한 테이블 사용 시 일어날 수 있는 문제에 대해 인식하고 있어야 한다.
- ▶ 컬럼 수는 최소로 효과적으로 유지해야 하고, 중첩 테이블은 가능한 한 피하고, 다른 대안이 없는 경우 사용한다
- ▶ 테이블 내의 정보는 가능하면 수평으로 읽혀져야 하며, 서로 다른 웹 브라우저에 따라 가능한 동일하게 표현되어야 하고 호환성이 확인되어야 한다.
- ➤ Ending 태그는 절대 생략해서는 안 되며, 셀 내의 배경 이미지는 구버전의 브라우저 에서는 지원되지 않으므로 피해야 한다.

#### 4) 대안 TAG의 정의

- <img>, <applet>, <input> ,<object>, <applet> 태그 등에는 이미지를 보지 않 거나 볼 수 없는 사용자나 검색 엔진 위치설정에 매우 유용한 ALT나 LONGDESC, TITLE 같은 텍스트 정의를 반드시 삽입한다.
- ▶ 웹사이트는 그래픽을 연결시키지 않은 상태로도 사용이 가능해야 하며, 이미지를 볼수 없는 사람들의 비용과 이익간의 균형을 반드시 고려해야 한다.
- ▶ 대안 태그는 항상 포함되어야 하며, 이미지의 외관뿐만 아니라 기능을 설명해야 한다. 내용은 100 문자를 초과하지 않아야 한다.
- ➢ 중요한 로고가 처음으로 사용되는 곳에는(예를 들면 웹사이트 상에), 완전한 공식적인 설명("X 정보컨텐츠 팀 로고 : ....을 나타내는 로고" 등)을 제공하는 것이 권장된다. 이후 로고가 반복될 때는 ALT 텍스트 내에 "X 정보컨텐츠 팀 로고"로 명명할 수 있다.
- 때때로 중요 정보(예를 들면, 차트, 테이블 또는 다이어그램)를 나타내는 어떤 이미지의 내용에 대해 자세한 설명이 제공될 필요가 있다. 이 설명은 주요 웹 페이지 내에텍스트로 포함되거나, IMG 요소의 LONGDESC 속성에 의해 링크된 웹 페이지에 위치시킬 수도 있다.
- ▶ IMG와 A에서 사용 실례

<img src="access.gif" alt="[Description]" longdesc="imgdesc\_a.html"><a
href="w.htm" title="Description of Accessbility">[D]</a>

▶ OBJECT에서의 사용 실례

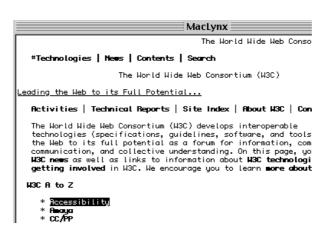
<object data="accessbrdlogo.gif" type="image/gif"> The Access Board's
<a href="projected.html">projected budget</a> for Fiscal 2005 is ...
</object>

▶ IMAGE MAP에서의 사용 실례

```
<OBJECT data="navigation.gif" type="image/gif" usemap="#mapnav">
   <MAP name="map1"><P>Navigate the Access Board site.
       href="guidelines.html"
                                shape="rect" coords="0,0,118,28">Access
   [<A
   Guide</A>]
   [<A href="news.html" shape="rect" coords="118,0,184,28">Go</A>]
   [<A href="search.html" shape="circle" coords="184.200,60">Search</A>]
   </MAP>
   </OBJECT>
▶ TABLE에서 사용 실례
   <TABLE border="1" summary="This table charts the number of web pages
   analyzed by each agency head, what kind of media the pages contain, and
   whether or not the page is part of the Executive Branch.">
   <CAPTION>Web pages Analyzed by Agency Heads</CAPTION>
   <TR>
   <TH id="header1">Agency Head</TH>
   </TR>
   </TABLE>
```

#### 5) 접근성 시험

▶ 웹사이트 접근성은 Website garage (http://www.websitegarage.com) 혹은 Bobby (http://www.cast.org/bobby) 등에서 테스트해 볼 수 있다. 좋은 방법은 텍스트 브라우저인 Lynx를 활용하여 웹사이트를 시범적으로 조사해 보는 방법도 있다.



<그림3. > 텍스트 기반 Lynx 브라우저의 실행 모습 (http://www.lynx.org)

#### 2.3 웹브라우저간 특이성

끊임없이 새로운 브라우저가 출현함에 따라 표준과 호환성에 대한 중요도가 점점 증대되고 있다. 그러나 많은 웹브라우저들은 여전히, HTML, CSS, 자바스크립트 등의 표준안을 충분히 지원하지 못하고 있다. 그러나 표준을 지키려는 노력들이 진행되고 있기 때문에 보다 빠른 속도록 브라우저간 웹 호환성이 이루어 지고 있다.

모든 웹브라우저가 같은 형태의 페이지를 출력할 것으로 예상하지만 〈표 〉에서 보이는 것 처럼 1999년 이전에 나온 오래된 웹브라우저들은 자바스크립트, CSS, HTML4의 기능들을 구현하지 못하고 있다. 따라서 Cross Browsing의 목표는 완벽한 호환성에 두는 것이 아니라 이종 웹브라우저에서 사용되는 비호환 및 비표준 구현 방식과 기법들을 가능한 표준안에서 수용할수 있는 방법을 찾는 것이다.

	프레임	테이블	플러 그인	font 크기	font 색상	자바스 크립트	자바_	CSS	gif89	DHTML	IFRAME	테이블 색 -	VML
IE6.0	V	V	V	٧	V	V		V	V	V	V	V	٧
IE5.5	V	V	V	V	V	V	V	V	V	V	V	V	٧
IE5.0	V	V	V	V	V	V	V	V	V	V	V	V	
IE4.0	V	V	V	٧	V	V	٧	V	V	V	V	V	
IE3.0	V	V	V	V	V	V	V	V	V		V	V	
IE2.0		V		V	V								
IE1.0		V		٧	V								
NS7.0	V	V	V	V	V	V	V	V	V	V	V	V	V
NS6.1	V	V	V	V	V	V	V	V	V	V	V	V	V
NS6.0	V	V	V	V	V	V	V	V	V	V	V	V	V
NN4.7	V	V	V	V	V	V	V	V	V	V		V	
NN4.5	V	V	V	V	V	V	V	V	V	V		V	
NN3.0	V	V	V	V	V	V	V		V			V	
NN2.0	V	V	V	V	V		V		V				
NN1.1		V		V									
MZ1.3	V	V	V	V	V	V	V	V	V	V	V	V	V
MZ1.0	V	V	V	V	V	V	V	V	V	V	V	V	V
OP7.V	V	V	V	V	V	V	V	V	V	V	V	V	V
OP6.0	V	V	V	٧	V	V	٧	V	V	V	V	V	٧
OP5	V	V	V	٧	V	V	٧	V	V	V	V	V	V
OP4	V	V		V	V	V	V	V	V		V	V	V

OP3.6	V	V	V	٧	V	V	V	V
OP3.5	V	V	V	V	٧		V	V

<표1. 윈도우에서 주요 웹브라우저 별 지원 웹기술 내역>

대부분의 사용자들은 NN4와 IE5 이상의 브라우저를 사용하고 있다. 그러나 IE가 다른 웹브라우저와 확연히 다른 점은 HTML태그의 시작과 끝이 명확하지 않을 때도 적당히 해석해서 표시를 해 준다는 것이다. 마크업(Markup) 언어의 특징은 시작과 끝의 태그가 명확하여야 하는데도, 다른 브라우저와 달리 IE가 적당히 해석해 줌으로서 개발자나 웹디자이너들이 표준적인코딩에 비해 실수할 가능성을 높다. 또한, 이것들이 다른 브라우저에서 다르게 보이게 하는 결정적인 요인이 되기도 한다.

HTML 표준은 아니나 지원이 다른 경우를 살펴보면, <body> marginwidth, marginheight는 NS에서 지원하고 topmargin, leftmargin은 IE에서 각각 다르게 지원하므로 같이 설정해 주어야 한다. <hr>
이야 한다. <hr>의 color와 <frame>의 framespacing, 의 bordercolor-light, bordercolordark 그리고 배경음악을 들려주는 <br/>
당명sound>, <script>의 id는 IE만 지원되는 것이므로 가급적 사용하지 않거나 대체되는 자바스크립트 기능을 이용하는 것이 팔요하다. 특히 <br/>
<br/>
<br/>
>bgsound>의 경우 <object>를 사용하여 활용 가능하다.

NS와 IE의 브라우저 시장경쟁 당시에 각 사의 대표적인 비교 태그인 <bli>chlink>, <marquee>는 글자를 깜박이거나 흐르는 기능을 해주는 것들로 이들은 요즘 자바스크립트로 충분히 구현 가능하다. IE만 지원하는 기능 중에는 <comment>, <xml>등의 태그가 있으며, <spacer>은 NS에서만 지원하는 등 차이가 있다. 이러한 웹브라우저 사이의 특이성은 테이블(table)에서 표시하는 방식에서 확연히 다른 결과를 보여준다. <그림4>은 중 하지 않았을 때 보여주는 결과의 차이이다



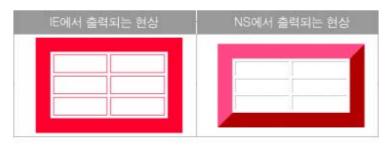
<그림4. > table의 td 태그를 완결하지 않은 경우

또한, <그림5>는 테이블에 background 이미지 파일을 설정하였을 때 달리 나타나는 현상을 표현한 것이다



<그림5.> table의 background 이미지를 지정한 경우

다음 <그림6>는 테이블에서 cellspacing과 cellpadding을 설정한 경우 달리 나타나는 현상을 표현한 것이다.



<그림6.> table의 cell 설정을 한 경우

테이블 뿐만 아니라 <form>의 경우도 input type이나 submit 버튼 형식들이 브라우저에 따라 다른 모양으로 나타나는 것을 발견할 수 있다.



<그림7.> form 속성들의 다른 형태의 표시

브라우저들 간에 나타나는 이러한 차이점은 HTML 태그의 속성을 각기 다른 렌더링 엔진을 통해 표현하기 때문에 생기는 문제들로 디자인을 달리 보이게 하는 요소들로서 지적되고 있다. 이러한 속성을 같게 보이게 하기 위해서는 요즘에는 CSS를 사용하는 방법이 추천되고 있다. 브라우저의 특이성은 당연히 존재하는 것이며 이러한 특징들을 알아두고 확인하여 최대한 표현하고자 하는 컨텐츠 형태로 출력하는 것이 중요하다.

#### 2.4 표준이란 무엇인가

웹사이트에 적용하는 HTML, CSS, Javascript 같은 것은 어디에서 정해져서 사용되는 것일까? 이 같은 승인된 개방형 인터넷 표준은 즉 World Wide Web Consortium (W3C)에서 만들어 진다. W3C는 1994년 10월 미국의 MIT 컴퓨터 과학 연구소(MIT LCS), 유럽의 정보수학유럽연구컨소시움(ERCIM), 그리고 일본의 게이오 대학이 연합하여 만들어진 국제적인 웹 기술 표준 기구이다.

언뜻 보기에는 연구 기관으로만 이루어진 것 같으나 웹과 관련된 510여개의 국제적인 다국적 IT 기업체가 참여하여 자사의 하드웨어와 소프트웨어에 웹 표준기술을 탑재하거나 자사의 기술을 표준화 하고자 하는 치열한 전투장 이기도 하다. 또한, W3C는 각국의 W3C 사무국인 지역조직과의 파트너 프로그램을 가지고 있다. W3C의 역할은 정보, 의견 교환, 아이디어 창출, 독립적 사고, 그리고 공동의 이해를 위하여 명세, 가이드 라인, 소프트웨어, 그리고 도구 및 규칙등의 표준안을 제정함으로써 웹의 모든 잠재력을 이끌어 내는 것이다. 즉, 개발자, 설계자, 그리고 표준 전문가 들에게 W3C 권고안을 채택을 촉진하게 하고, 향후 권고안을 만들 수 있도록장려해 주는 강제 기관이기도 하다.

웹에 관련한 표준에는 우리가 흔히 말하는 표준(Standard)은 존재하지 않으며, W3C의 토론을 통해 나온 권고안(Recomendation)이 가장 최상위 이다. 표준의 종류에는 제안된 표준(Draft), 작업하는 표준(Working Draft, WD), 확정될 권고안(Candidate Recommendation, RC), 확정된 권고안(Recommendation)이 있다.

권고안을 제정하는 방법은 1) 어떤 기능을 Draft로서 제안하고, 2) 드래프트를 실제로 적용할수 있게 기술적인 작업을 하고(Working Draft), 3) 이를 다시 논리 오류가 없는지, 실제 하드웨어에서 지원이 가능한지를 살피고, 4) 정식 권고안이 되기 전에 기업체에 공개하여 토론을 거친 후(Recommendate Candidation), 5) 마지막으로 권고안(Recommendation)을 확정한다.

웹브라우저 중 Mozilla, Netscape 등은 문서로서 확정된 추천안(Recomendation)이나 표준의 수정본 가운데 역시 확정된 표준을 지원한다. 다시 말해 모질라는 HTML 4.0도 지원하지만,

HTML4.01을 더 잘 지원해 준다. 특히 W3C 전용 브라우저인 Amaya 브라우저가 했던 표준의 기술 지원 시험을 요즘에는 모질라에서 하고 있어 더욱 빠르게 표준 기술이 적용되고 있다.

오페라도 비슷한 정도로 표준을 지원해 주지만, 빠른 속도를 유지하기 위해 MathML 등을 제대로 해석하지 못하기도 한다. IE는 지원하는 표준의 종류만을 보자면, 다른 두 가지 브라우저보다 더 뛰어납니다. 그러나, 지원하는 표준이 Recomendation이 아니라는 데 문제가 있다. IE는 Microsoft가 제안했던 내용만을 지원하는데, Recomendation에 자신들이 제안했던 내용이 적고, RC나 WD 또는 Draft에 자신들이 제안한 내용이 많다면 그것을 지원한다. 대표적인에가 HTML 4.0과 XHTML 1.0/1.1이다. HTML 4.0은 거의 모든 부분에서 MS의 의견이 반영되어 있다. 그럼에도 불구하고 HTML4.0의 후속 버전인 XHTML 1.0/1.1은 제대로 지원하지 않고 있다. 왜냐 하면, HTML을 모듈화하면서 MS의 의견이 상당 부분 표준에 채택되지 못했기 때문이다. 또한 CSS Level 2(흔히 CSS2) 지원도 미흡하다. 브라우저가 W3C의 권고안을 지원하는 데는 이러한 복잡한 관계가 얽혀 있다.

#### 2.5 W3C 표준의 내용

지금부터는 W3C에서 제공하는 웹사이트 제작을 위한 각종 권고안에 대한 특성을 살펴본다.

HTML(Hypertext Markup Language)는 웹페이지를 표시하는데 기본 언어로서 사용된다. 웹 컨텐츠의 내용은 표준 HTML 포맷으로 적용해야 하며 정보가 독점적인 고유 포맷으로 제공되는 경우, HTML 포맷도 제공되어야 한다. 브라우저 호환성은 모든 경우에 있어 고려되어야 하며, 웹사이트는 단일 웹 브라우저에 맞추어 제작되어서는 안되며, 클라이언트 그룹에 의해 빈 번하게 사용되는 웹 브라우저에서 올바르게 작동해야 한다. 최신의 HTML 표준은 4.01이며, HTML을 XML과 결합한 XHTML이 권고안으로 나와 있다.

CSS(Cascading Style Sheets)는 템플리트를 생성함으로써, 비 전문 설계자도 웹 페이지를 손쉽게 제작 관리 할 수 있게 한다. CSS는 HTML을 비롯하여 사용자 정의의 디자인 속성, 즉글꼴, 크기, 색상, 이벤트 등을 지정할 수 있으며 CSS를 사용한 모든 페이지는 기존 버전과의 호환성 되게 어떤 브라우저에서도 내용을 열람할 수 있다. CSS를 이용하여 설계자는 서로 다른 화면 해상도와 브라우저 상에서, 테이블 없이도 동일하게 보여질 수 있는 페이지를 생성할 수 있다. 단 IE4.0 이하와 NN4이하의 오래된 웹브라우저에서는 CSS를 지원하지 못한다. CSS를 사용하여 생성한 페이지와 템플리트는 다양한 브라우저, 화면 해상도 및 액세스 기술을 사용하여 테스트하여야 하며, 최신 시스템 사용자가 아니더라도 적합한 접근이 보장되어야 한다. CSS의 최신 표준안은 CSS2 이다.

XML(eXtensible Markup Language)은 HTML이나 CSS로서 표현되지 못하는 영역을

DTD를 이용하여 정의하여 사용자 정의의 태그를 생성하여 제작할 수 있는 메타 마크업 언어이다. XML 사용 분야를 검토하여 적절한 용도에 맞게 사용하여야 한다. XML이 고려되는 애플리케이션은 사용자가 필요한 정보를 얻기 위해 하나 이상의 데이터베이스와 상호 작용할 필요가 있는 경우, 작업이 사용자에게 전달되어 사용자가 자신의 기록 혹은 문서에 액세스할 것이예상되는 경우, 서로 다른 세트의 데이터가 서로 다른 사용자에게 디스플레이 되어야 하는 경우, 정보 검색 및 디스플레이와 관련하여 사용자 선호 프로파일을 구축해야 할 필요가 있는 경우, 각 개인이 스타일 시트를 사용하여 다양한 포맷으로 문서를 갱신해야 할 필요가 있는 경우에 사용 가능하다. XML은 다양한 인터넷 비즈니스 환경에 손쉽게 적응 가능하여 웹 표준 분야에서 가장 활발한 표준 제정 활동이 이루어 지고 있다.

DOM(Document Object Model)은 웹페이지에 표현되는 모든 속성에 대해 객체화 하여 이를 자유 자재로 사용할 수 있도록 만든 것이다. document, from, window 등 각각의 속성을 객체화 하여 트리 구조로 형상화 하여 이에 대한 이벤트 처리 같은 것이 가능하다. DOM에는 크게 W3C DOM과 MS DOM이 있는데, IE6.0은 아직 MS DOM을 따른다. Mozilla나 Netscape6 이상 최신 버전들이 W3C DOM을 지원하고 있다. 이로 인해 생기는 몇 가지 문제점이 Javascript 운영 상에 존재한다.

Javascript는 표준으로 제정된 것은 아니다. 또한, 모든 웹브라우저 사용자가 JavaScript를 볼 수 있는 것은 아니다. 특정 방화벽은 JavaScript가 통과하는 것을 허용하지 않는다. 그럼에 도 Javascript는 DOM이 표준화 되면서 웹브라우저에 널리 쓰이고 있다. 주의할 점은 클라이 언트측 스크립트는 여러 브라우저에서 폭 넓게 검사되어야 한다. 핵심 기능은 JavaScript에 의해서만 제공되어서는 안 된다. 또 JavaScript는 주석 코드를 사용하여 비호환성의 웹 브라우저 로부터 숨겨져야 한다. JavaScript는 HTML 문서의 Head 내에 위치해야 제대로 동작한다 따라서 문서의 Body 내에 JavaScript를 위치시키는 것은 피해야 한다. Mouseover는 브라우저의 검출 옵션과 함께 사용될 수 있을 뿐이다. JavaScript 변수는 부적절하지 않는 한 논리적으로 이름 붙여져야 한다.

아래는 웹사이트 개발에 필요한 표준안들에 대한 목록이다.

- ▶ [CSS1] "CSS, level 1 Recommendation", B. Bos, H. Wium Lie, eds., 17 December 1996, revised 11 January 1999. CSS1 권고안: <a href="http://www.w3.org/TR/1999/REC-CSS1-19990111">http://www.w3.org/TR/1999/REC-CSS1-19990111</a>. CSS1 최신 버전: <a href="http://www.w3.org/TR/REC-CSS1">http://www.w3.org/TR/REC-CSS1</a>.
- ➤ [CSS2] "CSS, level 2 Recommendation", B. Bos, H. Wium Lie, C. Lilley, and I. Jacobs, eds., 12 May 1998. CSS2 권고안: <a href="http://www.w3.org/TR/1998/REC-CSS2-19980512">http://www.w3.org/TR/1998/REC-CSS2-19980512</a>. CSS2 최신 버전: http://www.w3.org/TR/REC-CSS2.

- > [DOM1] "Document Object Model (DOM) Level 1 Specification", V. Apparao, S. Byrne, M. Champion, S. Isaacs, I. Jacobs, A. Le Hors, G. Nicol, J. Robie, R. Sutor, C. Wilson, and L. Wood, eds., 1 October 1998. DOM1 권고안: http://www.w3.org/TR/1998/REC-DOM-Level-1-19981001. DOM1의 최신 버전: http://www.w3.org/TR/REC-DOM-Level-1
- ▶ [HTML40] "HTML 4.0 Recommendation", D. Raggett, A. Le Hors, and I. Jacobs, eds., 17 December 1997, revised 24 April 1998. HTML 4.0 권고안: http://www.w3.org/TR/1998/REC-html40-19980424 HTML 4.0 최신 버전: http://www.w3.org/TR/REC-html40.
- ➤ [HTML32] "HTML 3.2 Recommendation", D. Raggett, ed., 14 January 1997. HTML 3.2 최신 버전: <a href="http://www.w3.org/TR/REC-html32">http://www.w3.org/TR/REC-html32</a>.
- > [RDF] "Resource Description Framework (RDF) Model and Syntax Specification", O. Lassila, R. Swick, eds., 22 February 1999. RDF 권고안: http://www.w3.org/TR/1999/REC-rdf-syntax-19990222 RDF 1.0 최신 버전: http://www.w3.org/TR/REC-rdf-syntax
- ▶ [XML] "Extensible Markup Language (XML) 1.0.", T. Bray, J. Paoli, C.M. Sperberg-McQueen, eds., 10 February 1998. XML 1.0 권교안: http://www.w3.org/TR/1998/REC-xml-19980210. XML 최신버전: http://www.w3.org/TR/REC-xml
- 아래는 웹사이트의 접근성을 높이기 위한 표준안들 이다.
  - ➤ [TECHNIQUES] "Techniques for Web Content Accessibility Guidelines 1.0", W. Chisholm, G. Vanderheiden, I. Jacobs, eds. 이 문서에서는 "웹 콘텐츠 접근성 지침 1.0"에서 정의한 체크포인트를 어떻게 구현하는지에 대해 설명하고 있다. 이 기술 문서의 최신 버전: http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/
  - ➤ [WAI-AUTOOLS] "Authoring Tool Accessibility Guidelines", J. Treviranus, J. Richards, I. Jacobs, C. McCathieNevile, eds. 저작 도구 접근성 지침에 대한 가장 최신 작업 초안(Working Draft): <a href="http://www.w3.org/TR/WAI-AUTOOLS/">http://www.w3.org/TR/WAI-AUTOOLS/</a>
  - ➤ [WAI-UA-SUPPORT] 이 문서에서는 (보조 기술을 포함하여) 웹 표시 장치들이 여기에서 언급된 접근성 관련 기능을 얼마나 지원하는지에 대해 언급하고 있다. 문서 있는 곳: http://www.w3.org/WAI/Resources/WAI-UA-Support
  - ➤ [WAI-USERAGENT] "User Agent Accessibility Guidelines", J. Gunderson and I. Jacobs, eds. 접근성이 높은 웹 표시 장치를 설계하기 위한 이 지침에 대한 최신 작업 초안: http://www.w3.org/TR/WAI-USERAGENT/
  - ➤ [WCAG-ICONS] 적합성 아이콘들에 대한 정보와 그것의 사용법: http://www.w3.org/WAI/WCAG1-Conformance.html

➤ [UWSAG] "The Unified Web Site Accessibility Guidelines", G. Vanderheiden, W. Chisholm, eds. 통합된 웹 사이트 지침은 위스콘신 대학교의 Trace R & D Center에서 미 연방 교육부 산하 국립 장애 재활 연구소(National Institute on Disability and Rehabilitation Research, NIDRR)의 지원을 받아 만들었다. 문서 있는 곳: http://www.tracecenter.org/docs/html\_guidelines/version8.htm

W3C에서는 웹페이지가 표준안에 따라 만들어 졌는지, 접근성에 대한 고려가 이루어 졌는지 유효성 검사(Validation)에 대한 정보를 제공하고 있다. 개발의 맨 첫 단계에서부터 여러 가지검사를 시작하면, 초기에 식별한 접근성 관련 문제점은 더 수정하기 쉽고, 피해 갈 수 있다.

아래는 몇 개의 중요한 유효성 검사 방법으로 제시되는 것이다. 먼저 자동화된 접근성 검사 도구와 브라우저 유효성 검사 도구(http://validation.w3c.org)를 사용한다. 소프트웨어 도구가모든 접근성 관련 문제점을 다룰 수는 없다는 점을 유의한다. 예를 들면 링크 텍스트의 의미가적절한지 여부나, 대체 텍스트(text equivalent)의 적용 가능성(applicability) 등은 다룰 수 없다.

기초 적인 문법 검사를 하고 (예를 들면, HTML, XML 등), 스타일 시트의 문법 검사를 한다 (예를 들면, CSS). 텍스트만 나오는 브라우저 또는 에뮬레이터로 시험하여 페이지의 레이아웃이 올바른지 살펴 보고 여러 개의 그래픽 브라우저를 써서, 소리와 그래픽을 모두 받는 설정, 그래픽을 받지 는 설정 , 소리를 받지 않는 설정, 마우스를 쓰지 않는 설정, 프레임, 스크립트, 스타일 시트, 애플릿을 사용하지 않는 설정 등을 통해 얼마나 접근도가 좋은지 체크해 볼 필요가 있다. 또한, 최근에 나온 것 뿐 아니라 오래된 브라우저를 포함하여 여러 개의 브라우저로 시험해 본다면 더 좋을 것이다.

만약 가능하면 음성 브라우저(self-voicing browser), 화면 음성 변환기, 화면 확대 장치, 낮은 해상도의 화면 등을 써보면 자신의 웹페이지 접근도에서 문제되는 점을 고칠 수 있다. 마지막으로 철자법과 문법 검사기를 사용한다. 음성 합성기를 통해 페이지를 읽는 사람들은 철자법이 틀린 단어에 대해서 합성기가 읽어주는 것으로는 무슨 단어인지 추측할 수가 없을 것이다. 문법적인 오류도 없어야 이해하기가 쉽다.

기본적인 접근도 검사가 수행되었다면 문서가 간단 명료하게 작성되었는지 다시 점검한다. 일부워드 프로세서가 생성해 주는 가독성 통계치같은 것들이 명확성이나 간결성에 대한 좋은 지표로 쓰일 수 있다. 그보다 더 나은 방법은 경험 있는 편집자에게 명료성을 검토해 주도록 부탁하는 것이다. 또한 경험있는 편집자는 특정 언어(단어나 표현)나 아이콘 사용이 야기할 수도 있는 잠재적으로 민감한 문화적인 문제점을 가려내어 문서의 사용자 편의성(usability)을 높일 수도 있다.

# 3. 올바른 코딩 방법

#### **3.1 HTML**

표준 HTML을 담은 웹페이지를 만든 다는 것은 어떤 의미인가? 먼저는 비표준적인 HTML과 자바스크립트 코드를 표준 코드로 바꾸는 것을 의미한다. 또한 웹페이지를 완벽한 형태(wellformed)를 갖춘 코딩 방식을 사용하는 것이다. 즉, HTML의 기본인 열기와 닫기를 완벽하게 구성하는 것이 중요하다. 특히 <img>. <br/>
하다>. 같은 태그들을 예외적으로 제외하고 모든 태그들은 이 규정을 지켜야 한다. 은 이것들을 빼먹기 쉽다. 특히 열고 닫는 위치는 정확해야 한다. 다음은 틀린 방식이다.

<b><i>These tags are not nested correctly.</b></i>
대신 아래의 표현이 정확한 방식이다.

<b><i>This text will be bolded and italicized.</i></b> 또는,

<i><b>This text will also be bolded and italicized.</b></i>

또한, <!DOCTYPE> 태그 역시 제대로 알지 못하고 사용하는 경우가 많다. 이것은 HTML 파일처음에 적는 것으로 이 페이지에서 사용하는 HTML 버전을 브라우저에게 알려주는 역할을 한다. HTML 버전에 따라 해석되는 방식이 브라우저에 따라서도 다르기 때문에 이를 지정해 주는 것은 매우 중요하다. 브라우저가 정확히 HTML을 해석하고 표시할 수 있도록 이 태그의 여러 가지 사용 방법을 알아야 할 필요가 있다. HTML4.01에서는 <font>. <b>, <frame>와 같은 태그를 사용하지 못하도록 하기 때문에 DOCTYPE에 대한 사용상 이해를 가져야 한다.

가장 흔히 사용되는 코드는 일반 형식과 엄격한 형식으로 나누어 지게 된다. <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">은 HTML 4.01에서 예전에 있었거나 없어진 태그도 지원하며, <font>에 지정된 스타일도 제대로 표현하여 준다. <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Strict//EN">은 HTML 4.01을 엄격하게 적용한다. <font> 태그에 적용된 스타일 보다는 CSS파일에서 지정된 스타일을 지켜 표현한다.

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN">은 HTML 4.01에서 Frames을 포함하는 웹 페이지에서 사용한다.

이러한 표현 방식에는 W3C <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01

Transitional//EN" "http://www.w3.org/TR/html401/loose.dtd">처럼 W3C에서 제공하는 DTD파일을 포함하기도 한다. 이는 HTML 4.01을 일반적으로 적용하는 경우이고, 엄격하게 적용하는 경우, <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Strict//EN" "http://www.w3.org/TR/html401/strict.dtd">, 그리고 앞서 언급한 대로 HTML4.01에서 프레임을 사용 하는 경우<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html401/frameset.dtd">를 사용한다.

이러한 DOCTYPE 선언에 차이는 다음과 같다. 만약 웹사이트에 BODY {font-family: Helvetica, sans-serif; font-size: 200%;}와 같은 스타일을 지정한 다음 서로 다른 DOCTYPE을 지정한 경우 아래와 같이 표현된다.

Here's some normal text in a paragraph which is in a sans-serif font, is green in color, and is 200% the default size. This is all thanks to the styles set on the BODY element.

This text is found inside a table, and so it will look different based on the DOCTYPE.

| In "strict" mode, the text will be the same as the paragraph above. In "loose" mode, the text will be the default size.

Here's another paragraph outside the table.

Here's some normal text in a paragraph which is in a sans-serif font, is green in color, and is 200% the default size. This is all thanks to the styles set on the BODY element.

This text is found inside a table, and so it will look different based on the DOCTYPE.

In "strict" mode, the text will be the same as the paragraph above. In "loose" mode, the text will be the default size.

Here's another paragraph outside the table.

(a) Transitional 형식으로 규정한 경우

(b) Strict 형식으로 규정한 경우

<그림8.> DOCTYPE에 따른 HTML 표시 방법

유사한 예로서, font-size에 대해서도 IE와 NN은 다르게 표현한다. 이러한 차이는 옛날 브라우저일수록 더욱 심하며, 최근에는 브라우저 개발자들이 표준을 지키고 있기 때문에 많이 사라졌다. DOCTYPE을 규정하는 가장 좋은 방법은 Strict 형식을 사용하는 것이다. 이것은 CSS를 통해 모든 HTML 태그의 속성을 자유자재로 규정할 수 있기 때문이다. 즉, b {font-weight: normal;} 라고 적는 다면 더 이상 〈b〉는 굵은체로 표시되지 않는다. 이것은 CSS를 통해 웹브라우저에 통일된 스타일 적용을 가능하게 한다.

통상적인 웹브라우저는 meta 태그를 통해 선언되어 있지 않더라도 문자코딩을 자동적으로 감별하여 표시하지만 Mozilla는 그렇게 하지 않는다. 따라서 meta태그의 charset을 euc-kr 등으로 아래와 같이 지정해 주어야 한다. 만약 이것을 빠뜨리면 모질라에서 한글이 깨지는 상황이생겨 마치 오류처럼 보일 수 있는 가장 첫번째 요소이다.

<meta http-equiv="Content-Type" content="text/html; charset=euc-kr">

또한, form에서 post 후에 통상적으로 이전 문자셋 코드를 그대로 사용하는 데 모질라는 데이

터를POST로 송신한 후 페이지에 대해 자동 판별을 하지 않는다. 단, POST후에 표시되는 페이지에 문자 코드 선언이 있는 경우 (HTTP헤더 혹은HTML안에서의 meta) Mozilla는 그 지정된 문자 코드로 페이지를 표시한다.

을 사용할 때도 흔히 , 로만 행과 열을 표현하고 있는데, 제목행에 대해서는
를 데이터행에 대해서는 를 사용하는 것이 필요하다. 두 개 이상의 논리적인 제목행이나 제목열을 갖는 데이터가 들어있는 표에서는 데이터 칸끼리 또는 제목 칸끼리 관련 짓는 마크업을 사용한다. 예를 들면, HTML에서 행을 모둠 지으려면 THEAD, TFOOT, TBODY와 같은 마크업을 사용하고, 열을 모둠 지으려면 COL, COLGROUP을 사용한다. IE에서는 table내에 bordercolorlight 속성과 bordercolordark 속성을 통해 음영을 주고 있지만, 다른 브라우저는 이를 사용하지 않기 때문에 사용하지 않는 것이 좋다.

<표2>는 HTML에서 필수적으로 바꾸거나 사용하지 말아야 하는 표현들을 정리하였다. 비 권장 요소들은 대체할 수 있는 제안들이 이미 나와 있는 경우이다.

바르지 않는 표현	추천 하는 표현
NN4 <layer> </layer>	HTML 4.0 <div> </div>
NN4 <nolayer> </nolayer>	N/A
NN4 <ilayer> </ilayer>	HTML 4.0 <iframe> </iframe>
Nav1/2/3/4 <bli>k&gt; </bli>	CSS1에서 제공하는 문자 표시 요소 사용
	text-decoration: blink;
NN4 <div>, <layer> 에서 src=</layer></div>	HTML 4.0 <iframe> 의 src=</iframe>
IE3/4/5 <marquee></marquee>	HTML 4.0 <div> DOM1의 Javascript로 표현.</div>
IE3/4/5 <bgsound/>	HTML 3.2 NN4-에서는 <embed/> 기타 웹 브라우저에
	서는 <object> 를 사용.</object>

<표2. HTML에서 필수 변경항목>

NN4에서는 iframe이나 div가 지원되지 않는다. 따라서 iframe이나 div를 사용할 때 아래와 같이 layer나 nolayer를 같이 사용하면 NN4를 지원 할 수 있다. 단 div에서 src를 지정하여 외부 파일을 읽어 오는 것은 표준적인 방법이 아니므로 이때는 iframe을 사용해야 한다.

<IFRAME ID="foo" SRC=foo.html> <LAYER NAME="foo" SRC=foo.html>
</LAYER> </IFRAME>

<LAYER NAME="NN4content" SRC=foo.html></LAYER>
<NOLAYER><DIV ID="nav6content">.... </DIV></NOLAYER>

HTML4.01에서 권장되어 있지 않으면서 가장 널리 쓰이는 것이 <font>이다. 이것은 이미 CSS로 옮겨지고 있으므로 이는 <div>나 <span>으로 옮기는 것이 좋다.

```
<FONT color="blue" face="Helvetica">Real
<FONT size="+1">Fun</FONT> Ficture</FONT>
는 다음과 같이 표현한다. SPAN에 사용하는 id를 통해 CSS를 지정할 수도 있다.
<SPAN style="color:blue; font:Helvetica">Real
<SPAN style="font:larger">Fun </SPAN> Books</SPAN>
```

Frame은 이미 HTML4.01에서 권장하지 않는 방법이다. 이것은 iframe이나 div의 대체안들이 이미 나와 있기 때문이다. 될 수 있으면 frame을 쓰지 않는 것이 좋으나 만약 써야 한다면, frame 비 지원 브라우저를 위해 noframes 항목을 통해 설명을 넣어 주는 것이 좋다.

```
<FRAMESET ROWS="30,*">
  <FRAME SRC="foo.html">
  <FRAME SRC="bar.html">
  <NOFRAMES>
  This page requires frames. See <a href="noframes.html">sitemap</a>.
  </NOFRAMES>
</FRAMESET>
```

<img>요소의 alt속성에 값을 설정해 두면 그림 위에 마우스 커서를 놓았을때, pop-up에 의해 alt의 설명이 나온다고 생각하는 것은 잘못된 것이다. 그러한 기능을 하는 것은 title이다. title속성은 모든 HTML tag에 공통적으로 사용하여 설명을 나타내는 것이다. alt가 하는 역할은 단지 그림에 대한 설명뿐임을 기억해야 한다. 아래 예제는 그것을 설명해 준다.



만약 Mozilla에 버그를 찾아냈을 경우 <a href="http://www.mozilla.or.kr/" title="한글 모질라"> 한글 모질라</a>에 보고하세요. <img src="?.gif" alt="(?)" title="질문하기">

<form>은 HTML사양에서 내부 요소이지 자식 요소가 아니다. IE는 <form>태그를 닫지 않은 경우, 자동으로 닫아 주는 경향이 있는데 이것도 잘못된 것이다. 따라서 <form>은 정확히 열고, 닫기를 하는 것이 중요하다.

#### 3.2 CSS

CSS을 사용할 때 우리가 가장 잊어버리기 쉬운 사실은 CSS를 지원하지 않는 예전 웹브라우저가 있다는 점이다. 대개 IE3.0과 NN4등이 CSS를 지원하지 못하며, 이들 버전은 전체 웹브라우저의 1%도 채 되지 않는다. 그러나, 이들 웹브라우저를 위해서 아래와 같이 각 HTML 태그에 대해서 간단한 속성을 부여하거나, BODY의 속성을 미리 선언해 주는 것이 좋다.

```
<BODY alink="#xxxxxx" bgcolor="#xxxxxx" link="#xxxxxx"

text="#xxxxxx" vlink="#xxxxxx">
<H1><FONT face="Courier" size=+4>Test Title</FONT></H1>
```

그러나, 대부분의 웹브라우저가 CSS를 지원해 주기 때문에 다양한 웹브라우저에서 잘 작동할 수 있도록 하는 CSS에 대한 방법을 알아보기로 한다. 먼저 CSS를 특정 레이아웃을 만들어 별도로 사용한다는 생각을 버리고 기존의 HTML TAG에 스타일을 선언하여 잘 활용할 필요가 있다. 대부분의 웹디자이너들은 CSS를 새로운 class로 선언하고 사용하는 경우가 많다.

```
<STYLE type="text/css">
<!-- .title { font-size:12pt, font-color: red } -->
</STYLE>
<font face=title>Red Title</font>
```

이러한 CSS 사용은 별로 바람직하지 못하다. 기존의 <h1>, <b>, 등의 HTML Tag에 CSS를 적용하면 HTML코드도 간단하고, 웹브라우저 간 다르게 지정된 HTML 태그에 대한 표현도 통일될 수 있다.

```
<STYLE type="text/css">
<!- h2 { font-size:12pt, font-color: red } -->
</STYLE>
<h1>Red Title</h1>
```

margin-top은 될 수 있는 한 사용하지 않고, 폰트 크기나 사이즈 단위에서 %, em 등을 사용하지 않고 pixel만 사용하는 것이 좋다. line-height는 쓰지 않는 것이 좋으며, class명에 일

반 키워드 같은 것은 혼돈되므로 쓰지 않는 것이 좋다. BODY에는 margin-left. margin-right 만 설정하고 LI, DD, DT 등에는 스타일을 설정하지 않는 것이 좋다. 또한, <hl>, <hl> 같은 Tag를 사용하여 문서의 구조의 역할을 부여할 필요가 있다. 단순히 글꼴이나 글자의 크기를 배치하기 위해 사용하는 것이 아니라 HTML 태그의 쓰임새에 따라 CSS를 설정해야 한다는 의미이다.

스타일 시트를 표시하는 방식은 HTML상에 고정으로 표시하는 방법고 link속성을 통해 스타일 시트 파일을 읽어 들이는 방법이 있다. 만약 최신 브라우저에서 지원하는 @import라는 속성으로 스타일 시트를 읽어온다면 NN4를 위해 다음과 같이 고정 스타일 시트를 지정해 줄 필요가 있다.

```
<STYLE type="text/css">
/* Navigator 4 CSS rules */
</STYLE>

<STYLE type="text/css">
/* Advanced CSS rules ignored by Navigator 4 */
@import "advanced.css";
</STYLE>
```

link속성에서 스타일 시트를 읽어 오는 경우, 아래와 같이 title을 통해 여러 대체 스타일 시트를 사용할 수 있다. title이 없는 스타일 시트가 표시의 기본 형식이 되고 title속성에 따라 또는 rel의 속성에 따라 대체 스타일 시트를 표시할 수 있다. 그러나, IE의 경우 모든 스타일 시트를 다 읽어 오기 때문에 주위를 요한다. Mozilla의 경우 각 스타일 시트를 통해 텍스트의 레이아웃을 변경할 수 있기 때문에, 기본모드, 텍스트모드, 인쇄모드 등으로 레이아웃을 간편하게 바꿀 수 있는 장점이 있다. 모질라의 방식이 표준에 부합하는 지는 논쟁의 여지가 있다.

```
rel="stylesheet" href="foo.css" type="text/css">
k rel="stylesheet" href="bar.css" type="text/css" title="그룹1">
k rel="stylesheet" href="hge.css" type="text/css" title="그룹2">
k rel="alternate stylesheet" href="alt.css" type="text/css" title="그룹2">
```

link를 통해 style파일을 불러오는 경우, 이미 불러온 HTML과 다른 파일이므로 문자코드가 선언되지 않았다면, 스타일 시트에 문자코드를 선언해 주어야 한다. 이는 font-family에서 한 글 글꼴을 사용한다면 더더욱 필요한 것이다.

```
@charset "euc-kr"; /* charset */
```

아래는 HTML에서 표현되고 있는 것들 중 CSS로 옮겨서 표현 해야 되는 내용들에 대해 정리한 것이므로 숙지할 필요가 있다.

바르지 않는 표현	추천 하는 표현
HTML3 <center></center>	CSS1 text-align:center
HTML3 <strike></strike>	CSS1 text-decoration:line-through
HTML3 <u></u>	CSS1 text-decoration:underline
HTML3 <dir>, <menu></menu></dir>	HTML4 <ul></ul>

<표3. CSS에서 필수 변경항목>

CSS에 의한 Length값은 0의 경우를 제외하고 항상 단위를 적어주어야 한다. border-width 나 width, height등에 0 이외의 값에 단위가 없으면 표준 웹브라우저에서는 부정한 구문이되어 속성 설정이 무효화 된다. 그러나, IE에서는 이러한 값을 px로서 보완하게 되어 있으므로 가능하면 px로서 단위를 꼭 사용하는 것이 좋다.

색상을 표시할 때도 #3399ff과 같에 번호 앞에 #을 꼭 붙이도록 한다. 이를 붙이지 않아도 색상을 표현하는 브라우저가 있으나 이것은 바람직한 렌더링 결과가 아니며, 이러한 차이로 다른 브라우저 간에 다른 색이 표현되기도 한다.

또한, 스타일을 선언할 때 어떤 것은 순서에 유의해야 하는 것이 있다. :hover의 경우, 마우스를 특정 링크에 올렸을 때 나타나는 색상을 지정할 때 사용하는 데, 아래의 경우는 IE와 Mozilla가 다른 결과를 나타낸다.

a:hover{ color:red; } a:link{ color:black;} a:visited{ color:green;}

위의 CSS를 사용한 경우, 마우스를 올렸을 때 visited 속성에는 붉은색이 표시되지 않고 link에만 붉은색이 표시된다. 이것은 CSS 단계화 규칙이라는 것이 있어 어느 조건에 대해서 값을 여러 차례 지정하면 마지막에 지정된 것이 유효하게 되기 때문이다. 반면, hover를 제일 나중에 두면 제대로 동작하게 된다. 따라서, link, visited, hover, active, focus의 순서 대로 CSS를 정해야 한다.

우리가 문장의 스타일을 제어할 때 사용하는 text-decoration의 경우에도 브라우저에 따라 많은 차이가 난다. CSS속성에도 트리 구조가 있어 상위에서 실행된 내용이 하위에서 취소되거나 할 수 없게 되어 있다. IE에서는 확대해석을 통해 허용해 주고 있다.

IE6_0	Mozilla
검은 텍스트 <mark>붉은 텍스트</mark> 검은 텍스트	검은 텍스트붉은 텍스트검은 텍스트

<span style="color:black;text-decoration:underline;">

검은 텍스트 <span style="color:red;text-decoration:inherit;">붉은 텍스트</span> 검은 텍스트 </span>

text-decoration은 표준 사양에 비하면 Opera와 IE가 버그가 있는 것으로 보이므로, 이것을 사용할 때는 최대한 여러 브라우저를 통해 살펴 볼 필요 가 있다.

Windows, Mac 그리고 Unix에 따라 사용되는 글꼴은 모두 다르다, 따라서 같은 내용도 글꼴에 따라 매우 다른 모양을 나타낼 수 있다. font-family 속성을 정할 때 고려해야 할 점은 각이 OS플랫폼에서 유사한 글꼴을 집합하여 보여 줌으로서 최대한 통일 성을 유지 하게 하는 것이다. 〈표4〉는 맥킨토시와 윈도우에서 유사한 글꼴 형태를 나열한 것이다.

Generic Family	Common Macintosh Fonts	Common Windows Fonts		
Serif Times, New Century Schoolbook,		Times New Roman, Georgia		
Sans-serif	Helvetica, Arial, Verdana*	Arial, Tahoma, Verdana*		
Monospace	Courier	Courier New		
Cursive				
Fantasy	WingDings, Symbol	WingDings, Symbol		

〈표4〉 매킨토시와 윈도우에서 유사 글꼴

border 속성에서 border-color의 경우는 IE, Mozilla와 Opera 모두 다른 결과를 낼 수 있는 속성이므로 주의해서 사용해야 한다. 특히 Opera는 거의 이 속성을 지원하지 못한다. 또한 아래의 경우 처럼, 세로 사이즈가 내용보다 적은 경우, IE6.0는 전체를 포함하고 그 밖에 브라우저는 넣을 수 있는 단락까지만 넣는 오류가 있다. 표준 적인 방법으로는 후자가 옳다.



<div style="font-size:12px;height:18px;border:1px red solid;">

이것은 12px 테스트입니다. <br>

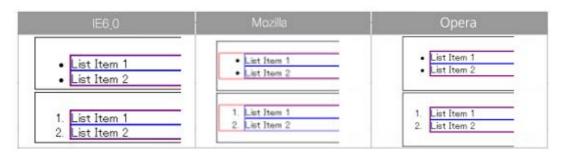
이행은 18px행에 들어갈까. </div>

한가지 예를 더 들면, 두 개의 박스간의 상하의 거리는 margin특성에 의해 결정되는데 내용이 초과한 구역이 다음에 계속되는 요소와 배치 하게 될 때 후속 내용과 초과해 버리는 상태가 되어 레이아웃에 문제가 생길 가능성이 있다.



```
   Text.<br>   Text.<br>   Text.<br>   Text.<br>   style="color:red;font-size:12px;border:red solid 1px;">
   Text.<br>   Text.<br>   Text.<br>   Text.<br>   Text.<br>   Text.<br</p>
```

따라서 상당히 이유가 없으며 텍스트를 내용에 가지는 구역에 대해서 height 값을 지정하는 것은 피하고, 브라우저의 해석에 맡기는 것이 기본적이다. 그러나 이것도 overflow라는 CSS2 사양에 따라 IE의 버그로 해석되는 경향이 있다.



```
ul,ol{ margin: 18px 0 18px 40px; padding: 0; }
li{ margin: 0; }
```

CSS에서 height: n%에 대해서 살펴 보면, 흔히 블록의 height: 300px;의 중간에 height: 50%;인 경우 산출치는150px로 생각된다. table이나 iframe같은 요소의 경우 height는 모두 auto로 지정되어 텍스트의 양만큼 표시하게 된다. 그러나 <doctype>이 transitional인 경우, 이를 전체 요소로 살펴 화면 모두를 채우는 결과가 나타날 수 있다. 이런 경우를 대비하여 가

장 상위인 body나 html에 height:100%를 지정해 줄 필요가 있다.

마지막으로 가로선을 표시하는 <hr>의 스타일 지정의 경우 좀 복잡한 경향이 있다. <hr>의 경우 브라우저 마다 모두 다른 표시 양식을 가지고 있어 이것을 정하기 어렵다. 아래와 같이 세가지 방식으로 color를 지정했을 때 각 브라우저 모두 다른 결과를 내기 때문이다.

```
<hr style="color:red;">
<hr style="background-color:red;">
<hr style="background-color:red;border:none;">
```

```
<hr style="color:red; background-color:red; border:none;
width:100%; height:0.3em;">
```

CSS는 웹브라우저에 따라 구현되어 있거나, 또는 덜 구현되어 있거나 지원하지 않는 경우가 있다. 또한, 구현되어 있더라도 문법에 따라 버그로서 존재하는 것이 있기 때문에 부록에 나타 난 표를 참고 하여 그 구현 정도를 확인해 볼 필요가 있다. (이 차트도 수시로 업데이트 되므로 참고문헌의 링크를 참조하여야 한다.)

#### 3.3 DOM과 Javascript

Javascript는 HTML내에서 여러 가지 기능을 실현할 수 있도록하는 스크립트 언어로서 Microsoft의 VBScript에 대별 된다. VBscript는 윈도우 환경의 IE에서만 동작하므로 대부분의 스크립트 기능을 사용할 때는 Javascript를 사용한다. Javascript는 어떤 표준안이 아니므로, 사용에 대한 강제적인 요소는 없다. 그러나 Netscape과 Microsoft의 Javascript 실행 방법과 코드 방법이 약간 달라 Cross browsing에 문제를 일으키기도 한다.

W3C 문서 객체 모델(DOM)은 HTML내의 거의 모든 요소를 스크립트 처리 가능한 객체로서 활용하여 만들어 Javascript 같은 스크립트에서 활용하거나 CSS에서 스타일을 지정해 줄 수도 있다. 일반적으로 DOM의 메소드와 모든 객체 모델은 DHTML 프로그래밍으로 사용되고 있던 독자적인 객체 모델보다 꽤 세련 되어 있다. 예를 들어 document 객체는 브라우저에 따라서는 요소의 배열이나 각종의 요소에 액세스하기 위한 속성을 가지고 있다. 예를 들어 NN4에서는 객체 모델 안의 특정의 요소에 액세스 하는데 document.tags 가 사용되며 IE문서 객체 모델(DOM)에서는 document.all[]이 사용된다. 이러한 차이로 인해 상당수가 W3C의 표준 DOM 의 사양에 포함되지 않아 Mozilla나 Netscape6/7 와 같은 표준에 준거한 브라우저

에서는 JavaScript 에러의 요인이 된다.

자바스크립트를 사용할 때는 script 태그에 LANGUAGE="JavaScript"를 선언해 주거나 type="text/javascript"를 사용해 준다. JScript, VBscript 등은 IE에서만 사용하므로 사용하지 않도록 한다. 또한, 텍스트 브라우저나 비 스크립트 브라우저를 위해 NOSCRIPT라는 요소를 사용하여 대체 텍스트나 링크를 제공하거나, 클라이언트측 스크립트 대신에 서버측 스크립트 라용해 호환성을 높여 주는 것이 좋다.

또한, 내용은 꼭 코멘트를 사용하여 텍스트 브라우저에서도 잘 표현 되도록 해야한다. 코멘트를 사용할 때는 <!----Comment-----> 로 쓰는 것은 잘못된 방법으로 <!--로 시작하여 -->로 끝내고, 주석 내용 안에는 하이폰(-)이 두개 이상 들어가지 않도록 한다.

즉, <!--=Comment==-->, <!-- Comment --> 방식이 바른 표현이다.

스크립트나 애플릿, 또는 다른 프로그램 객체를 사용하지 않거나 지원하지 않는 경우에도 페이지의 내용을 이해할 수 있어야 한다. 그것이 불가능하다면, 대안적으로 접근 가능한 페이지에 그들을 대체할만한 정보를 제공하는 것이 좋다. 예를 들면, 스크립트 기능이 꺼져 있거나 지원되지 않을 경우에도 스크립트를 활성화하는 링크가 작동하도록 해야 한다. (예를 들어, 링크의목적지 로 "javascript:"를 쓰지 않아야 한다. href 속성의 값으로 "javascript:"를 쓰는 것은 접근성 지침 위반일 뿐 아니라 HTML 표준 위반이기도 하다. 이런 경우, onClick 등을 사용해야 한다.

어떤 사용자 form에서 action을 받은 후 나온 결과에 자바스크립트 만을 제공해 자동 전환하는 결과는 될 수 있으면 사용하지 않는다. <script> href.location='test.html'; </script> 같은 내용만 담는 것은 권장하지 않으며 만약 한다면, window.href.location 이라고 정확하게 표현하거나, <meta> 태그의 refresh를 사용하거나 하는 것이 옳다.

특정 브라우저에서만 지원되는 태그를 사용하는 것은 추천하고 싶지는 않으나, 써야 한다면 그와 유사한 기능을 Javascript로 구성하여 제공하는 것이 좋다. 예를 들어, 글자를 흐르게 하

는 IE에서만 사용되는 <marquee> 태그를 지원하기 위해서는 다음과 같이 한다.

```
<script type="text/javascript">
var msg = "이 텍스트가 MARQUEE태그 처럼 스크롤 합니다. ";
function JsMarquee(id){
   if(document.getElementById){ // IE5, Mozilla
     var msgarea = document.getElementById(id);
       if (msgarea.firstChild && msgarea.firstChild.nodeName == "#text") {
        msgarea.firstChild.nodeValue = msg;
       } else if (document.all) {
        msgarea.innerHTML = msg;
     }
    } else if(document.all) { // IE4
        msgarea.innerHTML = msg;
    } else if(document.layers){ //NN4
     msg2=msg.substring(0,40);
      document.layers[id].visibility='hidden';
      document.layers[id].document.open();
      document.layers[id].document.write(msg2);
      document.layers[id].document.close();
      document.layers[id].visibility='show';
  msq=msq.substring(2,msq.length)+msq.substring(0,2);
  var cmd = "JsMarquee('" + id + "')";
   setTimeout(cmd,500);
</script>
```

IE와 다른 브라우저 사이에 흔하게 발생되는 에러는 W3C DOM이 아닌 MS DOM을 사용하는 경우 때문이다. 특히, 객체를 구별할 때 쓰이는 document.layers[] (Netscape4 사용), document.elementName, (예를 들면, 요소 에의 참조를 취득하는 경우 document.yooneek), document.all[](IE에서 사용)하는 방법은 W3C DOM이 모두 지원하지 않는다. 표준 방식은 document.getElementById("yoone다")을 사용하여 구별한다.

아래 표는 웹페이지의 다양한 HTML 요소에 정의된 스타일 규정을 변경할 때, 표준에 의거한

메소드를 알려주고 있다. W3C의 DOM2 권고의 CSS2 확장 인터페이스에 제시되어 있다. DOM 레벨2 에서는 요소의 CSS 속성에 새로운 값을 설정하기 위해 element.style 객체의 참조를 사용한다. 스타일에 대응하는 요소는 DOM 의 getElementById나 위의 DOM 액세스시에로 설명한 메소드에 의해 얻을 수 있다.

바르지 않는 표현	추천 하는 표현
NN4 document.layers[]	DOM1 document.getElementById()
document.tags[]	<span id="test">Test</span>
document.ids[]	에서 test 요소를 읽어오는 방법은
document.classes[]	<pre>element = document.getElementById("test")</pre>
document.elementName	
IE4/5 document.all	
NN4 element.visibility = Value;	DOM2 element.style.visibility = value;
NN4 element.left	DOM2 parseInt (element.style.left)
IE4/5 element.style.pixelLeft	
NN4 element.top	DOM2 parseInt (element.style.top)
IE4/5 element.style.pixelTop	
NN4 element.moveTo(X, y);	DOM2 element.style.left = value + "px";
<pre>IE4/5 element.style.pixelLeft = X;</pre>	DOM2 element.style.top = Value + "px";
<pre>IE4/5 element.style.pixelTop = y;</pre>	
NN4 handleEvent()	DOM2 dispatchEvent()

W3C DOM2 의 것 style.left 이나 style.top 속성이 돌려주는 값은 CSS의 단위("px"등)를 포함한다. 그러나, NN4의 element.left 나 IE4/5 의 element.style.pixelLeft는 정수치를 돌려준다. 요소의 왼쪽 혹은 위의 내부 스타일 설정을 정수값으로 읽기 위해서는 parseInt()을 사용해 문자 라인에서 정수값을 받는다. 반대로 설정하고자 하면 px과 같은 단위를 꼭 설정해야 한다.

<form id=testfrm><input type=text name="userid"></form>

흔히 위와 같은 form의 객체를 사용할 때 document.forms("userid")으로 표현하는 실수를 할 때가 있다. document.form은 메소드가 아니라 배열이기 때문에 document.forms[0]로 표현하는 것이 정확하다. 또는 document.form.userid 로 표현한다.

특히, 우리가 흔히 사용하는 요소 사용 방법중 중에 구성요소의 내용을 바꾸거나 수정하는 element.innerText, element.innerHTML, element.outerText, element.outerHTML을 사용

하는 것도 잘못된 것이다. 아래와 같이 지정된 dynatext라는 요소를 변경하는 작업에 대한 예를 보면,

```
<P>Papa's got <SPAN id="dynatext">a lot of nerve</SPAN>!</P>
```

자바스크립트의 dynatext의 요소를 먼저 span\_el에 담는다.

```
<script type="text/javascript">
var span_el = document.getElementById("dynatext");
```

각 요소의 컨텐트는 일련의 자식 노드(child note)로 분할되어 있으며, 각 노드는 단문과 그 자식요소로 구성되어 있다. 즉, 텍스트를 변경하고자 요소의 노드를 조정하는 것이 표준적인 방법이다. (노드의 구조 및 지원 메소드는 W3C DOM 레벨1 권고로 정해져 있다.)

자식 노드의 이하 요소를 가지지 않고 텍스트 뿐이라면 (통상) element.childNodes[0] 으로서 접근할 수 있는 1차 노드를 가진다. 즉, 우리가 element.innerText을 대체할 방법으로서는 element.childNodes[0].nodeValue 가 사용할 수 있다.

즉, span\_el.innerText = "a brand new bag"을 실현하려면, 다음과 같다.

```
var new_txt = document.createTextNode("a brand new bag");
span_el.replaceChild(new_txt, span_el.childNodes[0]);
```

span\_el.innerHTML = "a brand <b>new</b> bag" 처럼 HTML 태그가 포함된 문장을 표준적으로 수용하는 방법은 새로운 요소를 만든 후, 이에 3개의 노드를 추가한다. 처음의 텍스트노드, 자신의 텍스트 노드를 가지는 B 요소 , 마지막 텍스트노드에 각각 내용을 넣은 다음 원래 요소에 치환하는 방법이다

```
var new_el = document.createElement(span_el.nodeName);
new_el.appendChild(document.createTextNode("a brand "));
var bold_el = document.createElement("B");
bold_el.appendChild(document.createTextNode("new"));
new_el.appendChild(bold_el);
new_el.appendChild(document.createTextNode(" bag"));
span_el.parentNode.replaceChild(new_el, span_el);
</script>
```

문서의 컨텐트를 변경하는 스크립트에 대해 표준적인 방법이 더 어렵다고 느낄 지 모르겠다. 그러나 이를 통해 W3C DOM을 지원하는 Netscape6/7뿐 아니라 앞으로 제공될 브라우저에 동

작한다고 생각한다면 꼭 필요하다고 생각하는 것이 좋다.

#### 3.4 Plugin

Cross browsing에서 가장 난감한 문제에 봉착했다. 바로 Plugin이라는 문제이다. Plugin이란 HTML상에 특정 어플리케이션의 기능을 실행할 수 있도록 해주는 기술을 말하는 것으로 매크로미디어사의 Flash, 어도비의 Acrobat Reader, 리얼네트웍스의 Real Player, 마이크로소프트의 Windows Media Player 등이 여기에 속한다. 이들은 간단한 실행형 파일만으로도 웹브라우저 내에서 응용 프로그램을 실행할 수 있다.

이는 브라우저 전쟁 당시에 넷스케이프와 마이크로소프트가 각각 NSplugin과 ActiveX라는 상호 배타적인 기술을 브라우저에 탑재하면서부터 시작되었다. NSplugin은 OS와 관계없이 제작실행될 수 있고 Opera나 Sapari 같은 웹브라우저에 채용되는 반면, ActiveX는 IE가 설치된 윈도우즈 환경에서만 실행된다. 당시 넷스케이프는 <embed>, <applet>을 마이크로소프트는 <object>라는 별도의 태그를 만들면서 이 기능을 지원해 왔다. HTML 4.01에서는 object가 표준으로 제정되었기 때문에, object만 사용하면 되지만 예전 NN4버전과 표준 사용방법에 대해 여전히 논란이 제기되고 있다.

위의 코드는 IE에서 ActiveX를 불러오기 위한 HTML로서 classid는 clsid로 Flash의 고유프로그램 식별자이고, codebase는 Flash ActiveX가 설치되지 않은 경우 설치 파일이 위치한 경로를 지정한다. param은 문서 객체 모델에서 자식 노드로서 참고 할수 있는 값들을 의미한다. Netscape6이나 Opera4이상 부터는 object를 표준으로 지원하기는 하나 윈도우 미디어플레이어 등 몇 가지에 국한된다. 또한, classid 같은 식별자를 사용하는 것이 아니라 mimetype을 기반으로 하기 때문에 응용프로그램을 인식하는 방법이 IE의 object 기술방법과 크게다르다.

```
<object type="application/x-shockwave-flash"
data="test.swf" width="366" height="142" id="myFlash">
```

```
<param name="movie" value="test.swf">
  <param name="quality" value="high">
  <param name="swliveconnect" value="true">
     You need Flash -- get the latest version from
     <a href="http://www.macromedia.com/downloads/">here.</a>
</object>
```

위의 예제는 Mozilla 기반의 웹브라우저에서 flash를 찾아 실행하기 위한 object 의 사용 방법이다. mime-type을 통해 프로그램을 인지하고, IE와 같이 codebase를 통해 자동 설치하는 기능을 넣고 있지 않다. 특히, param을 무시하는 웹브라우저가 있기 때문에 data="...swf" 파일 형식으로도 설정해 주고 있다. object를 사용하는데 있어, 모질라 기반 브라우저들이 더 표준에 가깝게 기술되고 있으나 워낙 논란이 많고 Plugin이 IE 기반이 많기 때문에 모두 같이 쓰는 것이 유리하다. 그러나 두 가지 코드를 동시에 사용하는 것은 불가능 하므로 위의 두가지 코드를 지원 여부에 따라 나누어 쓰거나 아예 두가지 정보를 함께 담는 방법도 있다.

```
if (window.ActiveXObject) {

// IE ActiveX Code

document.write("<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000">
} else {

// Mozilla based Plugin Code

document.write("<object type="application/x-shockwave-flash"..">
}
```

<embed>와 <applet>은 지금까지 널리 쓰이긴 하였으나, HTML4.01에서 권고하지 않는 방법으로 되어 될 수 있으면 사용하지 않는 것이 좋다. 그러나 NN4와 같은 브라우저의 버전 호환성 유지 차원에서 필요할 수도 있다. 따라서 자바 애플릿을 불러올 경우에도 object 태그를 사용해야 하며, 아래와 같은 예제를 사용하여 불러온다.

```
<object classid="java:NervousText.class" width="534" height="50">
<param name="text" value="Java 2 SDK, Standard Edition v1.4">
You need the Java Plugin. Get it from
<a href="http://java.sun.com/products/plugin/index.html">here.</a>>>
</object>
```

최근 object와 plugin 사용과 관련하여 IE가 다른 업체의 특허를 침해했다는 법원 판결로 인터넷 사용 환경상 변화가 예상되는 가운데 인터넷 표준 언어인 HTML도 특허 논란에 휘말렸

다. 2003년 9월 미국 시카고 법원은 지난달 "MS의 IE가 에올라스의 플러그인 기술 관련 특허를 침해한 것이 인정된다"며 5억2000만달러를 배상하라고 판결했다. 이 소송의 핵심은 HTML 플러그인 사용에 대한 특허를 UC가 92년도에 받아서 발명만 전담으로 하는 회사인 이올라스에 94년에 팔았고, 이에 대해 소송을 제기해 W3C HTML 4.01의 object의 구동 방법에 대한 심각한 타격을 입힌 것으로 보인다. MS는 즉각 항소 의지를 밝히는 한편, W3C에 IE를 일부 수정할 뜻을 밝혔다. 이 문제는 IE 뿐만 아니라 모든 웹브라우저에 관련된 중요한 사안으로 특허를 피해 나가는 방법으로 아래 제안된 방법을 사용하도록 권고 하고 있다.

Objec사용을 HTML내에 넣지 않고 JS파일에 넣어 실행하는 방법으로 아래와 같다.

-->main.html 파일

```
<script src="aa.js" ...></script>
<script ...>test();</scirpt>
....
--> test.js 과일
function test() {
  document.write("<object classid=xxx...");
  document.write("</object>");
}
```

그리고 PARAM값을 DATA 객체로 사용할 때는 BASE64로 인코딩 하거나 js파일에 넣어 document.write를 해야 된다. 이와 관련된 사항은 http://msdn.microsoft.com/ieupdate를 참고하면 된다.

위와 같은 방법으로 object 태그의 호환성을 높일 수는 있으나 기본적으로 NSplugin과 ActiveX의 기술적인 호환성 때문에 제일 중요한 목표인 응용프로그램 실행에는 문제가 있다. 이를 해결하기 위해서는 ActiveX 개발자가 NSPlugin API를 참조하여 같은 기능의 Plugin을 개발해 줄 필요가 있다. 또한, 윈도우 환경에서는 ActiveX for Plugin, Plugin for ActiveX 등의 양쪽을 호환할 수 있는 plugin(http://www.iol.ie/~locka/mozilla/mozilla.htm)을 설치하거나, 리눅스 데스크탑 환경에서는 wine (http://www.winehq.org)같은 윈도우 에뮬레이터를 설치하여 상호 기술을 사용 가능하다.

# 4. 고급 디버깅 기법

#### 4.1 웹브라우저 알아내기

Cross Browsing 이란 웹브라우저와 관계 없이 통일된 웹페이지를 제공하는 데 목표가 있지만 실제로 웹브라우저에 따라 달리 표현하는 부분이 있기 때문에 사용자의 웹브라우저의 벤더와 버전을 확인하여 이에 따라 적절하게 웹페이지를 표시하거나 대응할 필요가 있다. 이는 오래된 웹브라우저를 사용하거나 특정 브라우저에서만 동작하는 기능을 제공할 때 특히 그렇다.

1994~2000년도 사이에 나온 브라우저들은 브라우저 시장 경쟁에서 이기기 위한 목적으로 출시된 것들이어서 W3C에서 제정하는 표준을 지키는 브라우저는 아니었다. 브라우저간 비호환성은 웹서비스 발전에 가장 중대한 도전이기 때문에 이를 표준적으로 지원하는 브라우저의 출현은 필수 불가결한 것이었다. 현재 NS6 이상, IE5.5이상 버전의 브라우저들은 W3C의 웹 페이지 표현에 대한 표준인 HTML4.0, CSS1/2, W3C DOM 시 방식을 지원하고 있다.

Cross browsing을 통해 웹페이지를 완벽하게 개발을 하기 위해서는 브라우저의 기능을 동작시에 판별할 수 있어야 한다. 즉, 에러를 일으키지 않고 다양한 방문자들이 폭 넓게 사용해 주기 위한 것이다. 일반적으로 사용되는 방법은 번거럽지만 브라우저를 식별하여 설계 시에 브라우저의 능력에 따라 웹페이지를 만드는 것이다. 그렇지만, 다양한 브라우저의 다른 기능을 개발자가 알아서 판단하고 제공한다는 것은 쉬운 일은 아니다. 그러나, 지금까지 나열된 웹브라우저 차이점을 숙지하여 브라우저에 따라 판별 해 준다면 매우 유용할 것이다.

다음은 브라우저를 판별하는데 사용하는 몇 가지 방법들이다.

```
if (navigator.appName == "Microsoft Internet Explorer") {
          document.all(id).style.visibility = "visible";
} else if (navigator.appName == "Netscape") {
    if (parseInt(navigator.appVersion) < 5) {
        document.layers[id].visibility = "show";
    } else {
        document.getElementById(id).style.visibility = "visible";
    }
}</pre>
```

위의 예에서는 navigator 객체의 appName 이라고 하는 속성 값을 따라 "Microsoft Internet Explorer" 혹은 "Netscape"를 판별하여 대응하는 코드를 실행하게 된다. 그러나, Opera와 같이 navigator.appName나 navigator.appVersion의 값을 간단하게 변경할 수 있는 브라우

저도 있고 개개의 브라우저를 하나하나 판별해야 하기 때문에 좋은 방법이라 할 수 없다.

그래서 대부분 객체 기반의 브라우저 판별법을 사용한다. 지원하는 브라우저에 객체모델이 존재 하는지 여부를 통해 간단하게 구현 기능을 확인하는 것이다.

이 예는 document.getElementById이라고 하는 객체를 가지고 있는 브라우저에 대해서는 같은 코드를 실행한다. document.getElementById 객체는 W3C이 규정되어 있는DOM의 표준으로 최근 웹브라우저는 대부분 지원하므로 통상 이 방법을 사용해야 한다. 따라서 W3C DOM을 사용하는 표준 웹브라우저에서 다음과 같이 <div id=xxx>...</div>로 규정된 영역을 이동하는 간단한 스크립트를 생성할 수 있다.

```
function moveElement(id, x, y){    // W3C DOM Browser
  var elm = document.getElementById(id);
  if (elm) {
    elm.style.left = x + 'px';
    elm.style.top = y + 'px';
  }
}
```

Browser sniffing으로 불리는 이러한 방법은 흔히 ECMAScript 함수에 의해 다루어져 아래와 같은 스크립트로 브라우저의 버전과 제품 벤더를 확인할 수도 있다.

```
// convert all characters to lowercase to simplify testing
var agt=navigator.userAgent.toLowerCase();

// *** BROWSER VERSION ***
```

```
// Note: On IE5, these return 4, so use is_ie5up to detect IE5.
var is_major = parseInt(navigator.appVersion);
var is_minor = parseFloat(navigator.appVersion);
// Note: Opera and WebTV spoof Navigator
var is_nav = ((agt.indexOf('mozilla')!=-1) && (agt.indexOf('spoofer')==-1)
&& (agt.indexOf('compatible') == -1) && (agt.indexOf('opera')==-1)
&& (agt.indexOf('webtv')==-1) && (agt.indexOf('hotjava')==-1));
var is_nav2 = (is_nav && (is_major == 2));
var is_nav3 = (is_nav && (is_major == 3));
var is_nav4 = (is_nav && (is_major == 4));
var is_nav4up = (is_nav && (is_major >= 4));
var is_navonly = (is_nav && ((agt.indexOf(";nav") != -1) ||
(agt.indexOf("; nav") != -1)) );
var is_nav6 = (is_nav && (is_major == 5));
var is_nav6up = (is_nav && (is_major >= 5));
var is_gecko = (agt.indexOf('gecko') != -1);
var is_ie = ((agt.indexOf("msie") != -1) && (agt.indexOf("opera") == -1));
var is_ie3 = (is_ie && (is_major < 4));</pre>
var is_ie4 = (is_ie && (is_major == 4) && (agt.indexOf("msie 4")!=-1) );
var is_ie4up = (is_ie && (is_major >= 4));
var is_ie5 = (is_ie && (is_major == 4) && (agt.indexOf("msie 5.0")!=-1) );
var is_ie5_5 = (is_ie && (is_major == 4) && (agt.indexOf("msie 5.5") !=-1));
var is_ie5up = (is_ie && !is_ie3 && !is_ie4);
var is_ie5_5up =(is_ie && !is_ie3 && !is_ie4 && !is_ie5);
var is_ie6 = (is_ie && (is_major == 4) && (agt.indexOf("msie 6.")!=-1) );
var is_ie6up = (is_ie && !is_ie3 && !is_ie4 && !is_ie5 && !is_ie5_5);
// KNOWN BUG: On AOL4, returns false if IE3 is embedded browser
var is_aol = (agt.indexOf("aol") != -1);
var is_aol3 = (is_aol && is_ie3);
var is_aol4 = (is_aol && is_ie4);
var is_aol5 = (agt.indexOf("aol 5") != -1);
var is_aol6 = (agt.indexOf("aol 6") != -1);
```

```
var is_opera = (agt.indexOf("opera") != -1);
var is_opera2=(agt.indexOf("opera 2") != -1 || agt.indexOf("opera/2") != -1);
var is_opera3=(agt.indexOf("opera 3") != -1 || agt.indexOf("opera/3") != -1);
var is_opera4=(agt.indexOf("opera 4") != -1 || agt.indexOf("opera/4") != -1);
var is_opera5=(agt.indexOf("opera 5") != -1 || agt.indexOf("opera/5") != -1);
var is_opera5up=(is_opera && !is_opera2 && !is_opera3 && !is_opera4);

var is_webtv = (agt.indexOf("webtv") != -1);

var is_TVNavigator = ((agt.indexOf("navio") != -1)
|| (agt.indexOf("navio_aoltv") != -1));
var is_AOLTV = is_TVNavigator;

var is_hotjava = (agt.indexOf("hotjava") != -1);
var is_hotjava3 = (is_hotjava && (is_major == 3));
var is_hotjava3up = (is_hotjava && (is_major >= 3));
```

### 4.2 레이아웃 방식의 변경

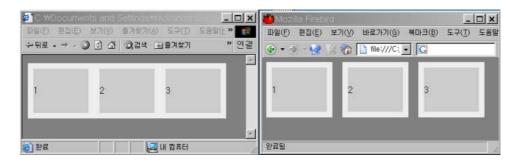
대부분의 웹 디자이너나 개발자들은 TABLE을 기초로 한 레이아웃을 제작한다. TABLE은 다양한 행과 열로 구성되어 있기 때문에, 격자형 문서구조를 구성하는데 아주 효과적이므로 많이 사용해 왔다. 그러나, 문서 구조가 복잡해 지면 TABLE의 복잡성 또한 증가하여 유지 보수 하거나 갱신 시 구조를 이해하는 데 매우 어려움을 겪게 된다. 이미 앞서 살펴 본 대로 TABLE은 각 웹 브라우저 마다 달리 표현되는 속성 때문에 Cross Browsing에 어려움을 주기도 한다. 그래서 TABLE형 레이아웃에 CSS를 결합하여 표현하기도 하는데 이것은 궁극적인 해결 방식이 될 수 없다.

TABLE형 레이아웃 대신에 대안으로 떠 오르는 것인 DIV형 레이아웃이다. DIV는 DHTML의 태그로서 소개되었다가 HTML4.0에 표준으로 채택되었다. DIV와 비슷하게 SPAN이나 LAYER등의 태그가 있었으나 DIV로 통일 되어 사용되는 것이 요즘 추세이다. DIV는 문서의 일정 영역을 구분하여 조각으로 만든 다음 각 조각(Component)을 자유롭게 활용하거나 배치시킬 수 있다. TABLE형은 행과 열에 따라 순차적으로 문서가 배치되는 반면 DIV형은 선언된 영역을 자유롭게 배치할 수 있다. CSS를 이용하면 HTML4.01과 CSS1/2를 지원하는 브라우저에서는 완벽한 Cross Browsing이 가능하다.

만약 세가지 테이블로 구성된 레이아웃을 표현해야 한다고 생각하여 보자.

### <table을 이용한 레이아웃 예제>

```
1
```



table을 통한 가장 일반적인 레이아웃을 한 경우이다. 그러나 이는 웹브라우저 마다 table을 표시하는 속성이 달라 전혀 다르게 보이게 된다.

### <div을 이용한 레이아웃 예제>

```
<STYLE type="text/css">
@import "layout.css";
</STYLE>
<div id="left"> 1 </div>
<div id="middle">2</div>
<div id="right">3</div>
```



위의 경우는 div와 CSS를 조합하여 만든 레이아웃 구성도로서 이를 통해 간단하고 명확하게 레이아웃을 표시할 수 있을 뿐 아니라, 웹브라우저에서 유사한 스타일로 표현 가능하다.

```
- layout.css 파일 예제
 #left {
       position: absolute;
       top: Opx; left: Opx; margin: 20px; padding: 10px;
       border: 5px solid #ccc; background: #666; width: 120px;
       width: 150px; /* ie5win fudge begins */
       voice-family: "\"}\""; voice-family:inherit;
       }
 html>body #left {
       width: 120px; /* ie5win fudge ends */
 #middle {
       margin: 20px 190px 20px 190px; padding: 10px;
       border: 5px solid #ccc; background: #666;
 #right {
       position: absolute;
       top: Opx; margin: 20px; padding: 10px; border: 5px solid #ccc;
       right: 0px; /* Opera5.02 */
       background: #666; width: 120px;
       width: 150px; /* ie5win fudge begins */
       voice-family: "\"}\""; voice-family:inherit;
 html>body right {
       width: 120px; /* ie5win fudge ends */
       }
```

이러한 레이아웃 방법은 말로만 듣는 것보다 실제로 웹사이트에서 구현해 보는 것이 제일 중요하다. 이러한 Box형 레이아웃을 배우기 위해서는 다음 웹사이트를 방문하기를 권장한다. http://www.thenoodleincident.com/tutorials/box\_lesson/boxes.html

### 4.4 비 IE 사용자를 위한 가이드

국내의 대부분 웹사이트들은 IE를 기준으로 만들어졌다는 비난 아닌 비난을 받고 있다. 대부분의 사용자들이 IE를 쓰는 만큼 3~4% 내외인 비 IE 사용자를 위해 웹페이지나 서비스를 바꾼다는 것은 쉽지 않다는 것은 사실이다. 특히, 상호 소통적이고 커뮤니케이션을 중요시 하는 국

내 웹서비스의 현실로 볼 때, Javascript와 같은 스크립트 언어와 Flash, Shockwave 그리고 다양한 ActiveX 플러그인 기술을 사용하는 것 자체를 막을 수는 없다고 생각한다. 그러나, 이러한 의존적인 기술을 사용하면서도 비 IE 사용자 혹은 비 Windows OS 사용자 들이 어느 정도 서비스를 접근할 수 있는 가이드 조차 마련되어 있지 않고 접근 자체가 거부 된다는 것은 문제라고 볼 수 있다. 여기서는 비 IE 사용자를 위한 몇 가지 사용 방법에 대해 설명하여, 이를 웹사이트에 소개함으로서 비 IE 사용자들에게 이해를 줄 수 있는 가이드를 제공하고자 한다.

#### 1) 모질라(Mozilla) 계열 웹브라우저

비 IE계열의 대표적인 웹브라우저가 Netscape이다. Netscape는 4.58버전일 끝으로 Navigaor라고 불리는 브라우저 시대를 끝내고, 소스를 공개함으로서 공개 소프트웨어로 전환하였다. 이 공개 소프트웨어 프로젝트를 모질라(Mozilla)라고 명명하고, Gegko라는 브라우저 엔진을 통해 웹브라우저를 발전시켜 왔다. Netscape6/7 버전은 Mozilla의 1.0.2, 1.4를 기반으로 한 것이며 모질라 기반 브라우저라고 할 수 있다. Mozilla, Netscape, Mozilla Firebird, Kameleon 등은 모두 모질라 기반 브라우저로서 모질라에 대한 정보만 제공하여도 비 IE 사용자의 상당수를 지원할 수 있다.

모질라에서 브라우저 설정을 어떻게 하며 어떤 폴더에 저장되는지 궁금해 하는 경우가 있다. 과거에 Netscape Communicator를 설치 해본 사람이라면, 프로필이라고 하는 개념은 친숙할지도 모른다. 모질라 기반 브라우저에서 프로필은 북마크 주소장, 캐시 메일 사용자 정의 설정등의 개인적인 데이터를 정리해 보존해 두는 폴더이다. Mozilla 는 복수의 프로필을 사용할 수있게 되어 있어 프로필의 변환에 프로필 관리자라고 하는 전용의 컴퍼넌트를 사용한다. 프로필관리자는 Mozilla 프로그램의 실행전 혹은 종료후가 아니면 실행할 수 없다.

Mozilla 의 프로필 관리자는 Netscape 6/7 의 프로필도 인식한다. 또 프로필의 데이터는 프로그램의 제거에서는 삭제되지 않는다. NS6/7 의 프로필을 사용하려면 불편이 생기기 때문에 이전에 NS6/7 를 설치 한 일이 있는 경우 반드시 신규의 프로필을 작성해야 한다. 신규의 프로필을 작성하려면 Mozilla 의 프로필 관리자로 "Create Profile..."을 통해 한다.

모질라 기반 브라우저에서 음악을 못 듣거나 플래시 및 자바 애플릿이 작동하지 않는 경우는 MIDI 플레이어나 Macromedia Flash Java Plugin 등의 플러그 인을 Mozilla에 설치하고 재구동하지 않은 경우로 생기는 경우가 많다.

모질라나 Netscape의 한글 버전을 다운로드 하여 설치하기 위해서는 한글 모질라 (http://www.mozilla.or.kr)를 이용하거나 모질라 영문 배포사이트 에서 OS에 맞는 모질라 영문 버전을 받아 설치한 후, 모질라를 통해 모질라 한글사이트에서 Version (mac/unix/win) 형

식의 언어팩을 설치한다. Edit->Preference...->Appearance->Language->Korean(KR)을 선택한 후 모질라를 재시작한다. 1.0버전 부터 Windows와 Linux용 설치 프로그램이 제공되고 있다. 부분적으로만 한글이 보일 경우, 모질라는 기본적으로 다른 버전에서 사용하던 프로필을 사용할 수 없기 때문에 예를 들어, 1.0버전에서 만든 프로필을 1.1 버전에서 사용할 경우이러한 문제가 발생한다. 브라우저가 업그레이드 된 경우, 이전 버전에서 만든 프로필을 사용하지 말고 새로운 프로필을 만든 후 한글 언어팩을 설치 하면 이 문제를 해결할 수 있다.

웹사이트에서 IE와 Netscape사이의 cross-browsing이 되지 않아 발생하는 문제에 대해서는 HTML 코딩 중에는 IE에서만 인식되는 비 표준 문법들이 있는데, 이것은 사이트가 IE 위주로 HTML이 작성되어 있거나 Netscape나 모질리를 지원하지 않는 경우일 수 있다. 개발자들이 브라우저에 따라 모두 테스트를 해보기 전까지는 문제를 발견해 내기가 쉽지 않고, 국내의 대부분 웹사용자들이 IE를 쓰므로 모질라나 Netscape 에서 Cross-browsing이 안될 수 있다. 사이트 운영자에게 Netscape나 모질라에서 발생되는 문제에 대해 메일을 보내셔서 문제점을 고칠 수 있도록 피드백을 마련해 주는 것이 좋다.

모질라에서 Object 태그를 쓰는 ActiveX콘트롤이 실행되지 않는 경우가 있다. 이것은 ActiveX 콘트롤이 Microsoft에서 개발한 기술로 Netscape의 Plug-in 기술과 대치되는 기술이기 때문이다. 국내 대부분의 웹사이트들이 ActiveX콘트롤을 즐겨 쓰고 있으나, 모질라나 Netscape에서는 기본적으로 ActiveX를 지원하지 않는다. Netscape7.0이나 모질라1.1 이상에서 ActiveX콘트롤을 지원 받을려면 윈도우 환경에서 ActiveX for Plugin, Plugin for ActiveX Plugin(http://www.iol.ie/~locka/mozilla/mozilla.htm)을 설치하거나, 리눅스 데스크탑 환경에서는 wine (http://www.winehq.org)같은 윈도우 에뮬레이터를 설치하여 상호기술을 사용 가능하도록 알려주어야 한다.

모질라의 북마크 파일을 백업하려면, bookmark.html 형식의 HTML파일을 프로필(profile) 디렉토리 아래의 사용자 디렉토리에서 가져온다. 프로필 디렉토리의 위치는 Win9x 같으면 Program Files 파일 아래의 mozilla.org(설치디렉토리) profile 아래에 있으며, 2000/XP라 면 'Document and Setting' 아래 개인 디렉토리의 'Local Setting' 아래 있다.

### 2) 오페라 브라우저

오페라(Opera)는 1990년 초, 노르웨이 통신회사, 텔레너의 연구소에서 근무하던 3명의 로부터 시작하여 1995년에 오페라소프트웨어가 설립되었다. 그리고 1996년에 윈도우 오페라 2.1의 오페라 최초 버전이 발표되었다. 1998년에, 오페라는 윈도우를 넘어 다른 플랫폼으로 오페라 브라우저를 확대하여 2000년과 2001년에 Linux, Macintosh, BeOS, Symbian OS (EPOC) 및 QNX와 같은 대중적인 플랫폼에 대해 출시되었다. 2000년 12월에 윈도우 오페라 5 버전이

광고가 지원되는 무료 버전으로 출시되었다. 첫달에 무료 오페라 5 버전은 2백만개가 다운로드되어 설치되었다. 오페라는 개발 초기부터 W3C의 표준 사양을 준수하고, 브라우징 속도를 가장 빠르게 한 특징을 가지고 사용자 층을 이끌어 나왔다.

오페라 한글판이 출시되기 앞서 이 한글 언어팩을 이용하여 한글로 오페라를 사용할 수 있다. 한글 언어팩을 클릭하여 다운로드 받은 후 오페라가 설치된 Opera 디렉토리에 압축을 풀면 된다. 다음에 오페라를 실행하여 "파일(File)/환경설정(Preferences)/언어(Language)"에서 사용자 인터페이스 언어 설정에서 옆에 버튼을 누르고 Opera 디렉토리 안에 owxxx\_xxxxko.lng파일을 찾아 선택하고 확인 후 설정한다. 그리고 오페라를 다시 시작하면 한글로 오페라를 사용할 수 있다.

오페라 사용시 웹페이지 내에서 한글을 제대로 표현하려면 다음과 같이 한다. 메뉴바 파일 (File) -> 환경설정(Preferences) -> 글꼴 및 색(Fonts and colors)를 선택하면 오른쪽 화면 나의 글꼴 및 색(my fonts and color)에서 먼저 일반글자(normal) 선택하고 변경을 누르면 대화창이 나타나는데 거기서 글꼴은 TT굴림 또는 TT돋음을 선택하고 유형에서는 보통, 크기는 10 또는 11(최적) 로 선택하고 확인을 선택한다. 그리고 다시 환경설정 창에서 아래 적용을 눌러 설정한다. 다른(css font-family 를 제외한) 항목들도 같은 방법으로 설정한다. 다음에 [환경설정-언어]에서 인코딩유지 항목에서 html 을 euc-kr 로 선택하고 설정하여 사용한다. 메뉴바, 도구모음 글꼴도 변경할 수 있다. 환경설정〉 브라우저 모양에서 "글꼴 및 색" 항목에서 '시스템 기본설정 사용'에 체크 지우고, "일반 텍스트", "비 활성텍스트", "북마크창 글꼴 "를 각각 누르고 차례로 글꼴을 바꾸면 된다. 크기는 동일하게 "9" 로 하면 되며, 글꼴 색상도 변경 가능하다.

오페라 브라우저를 사용할 때, 어떤 페이지는 다른 브라우저와 다르게 표시되는 것을 볼 수 있다. 대부분의 경우에서 그 차이는 그 표시되는 페이지에서 표준을 지원하지 않는 오류에 의해나타난다. 어느정도까지 오페라는 넷스케이프 및 인터넷 익스플로러로 표시되는 오류를 그대로복제하도록 시도하지만, 오페라는 우선적으로 표준 체계를 그대로 적용한다. 오페라와 넷스케이프/인터넷익스플로러 사이에 표시되는 차이에 대해서는 아래와 같다.

- ➤ 오페라에서 <HR>의 색은 배경 특징 이다, 그래서 오페라는 생성하는 내용 뿐만 아니라 모든 배경스타일을 승인한다. NN4 및 IE에서 그것은 전경 특징이다 (색). NN6은 역시 바 르게 한다 (전에 내용을 생성하고 사용된 후라면 오페라와 NN6 사이에는 차이가 있다).
- ▶ 링크 밑줄은 오페라 및 NN6 이전 버전에서와는 다르다 (하나의 색으로 밑줄이 사용되고, 텍스트는 다른 색으로 사용된다). 이것은 텍스트에 대한 CSS2의 결과이다.
- ▶ IE5/Windows는 오페라 ( 및 NN6)에서 상자를 보다 크게 보이게 하는 높이와 폭에서 오류가 있다. 이것은 표준 양식으로 IE6에서 수정되었다.

- ➤ IE5는 또한 위치에 대해 문제를 갖고 있다. 위치한 요소는 포함하는 요소가 아닌 가장 가까이 위치한 요소에 위치되어야 한다. 배경 이미지의 위치는 창이 아닌 요소 상자와 관계한다. 이것은 오페라에서 body 와 함께 위치한 이미지 (background-position: center)는 창의 중앙에서가 아닌 페이지의 중앙에서 보기 좋지 않게 나타난다는 것을 의미한다.
- ▶ 보통 padding은 그 body 요소에 적용된다. 그리고 여백을 두지 않는다 (body 및 head/html 요소 사이의 여백).

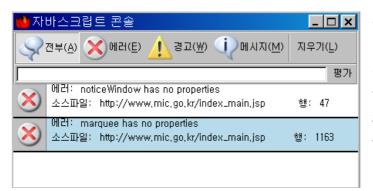
오페라에 대한 소개 및 한글 지원 페이지는 http://opera114.pe.kr를 참고하면 된다.

### 4.4 웹페이지 디버깅 방법

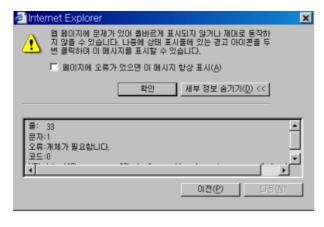
지금까지 Cross browsing을 위해 브라우저의 특성과 표준 지원 정도. 올바른 웹페이지 코딩 방법 등을 살펴보았다. 그러나, 이러한 가이드를 충분히 숙지하고 있어도 오류가 나는 것이 웹페이지이다. 가이드를 잘 익히는 것도 중요하지만, 결국 웹 개발자가 최종 작업을 마치고 여러 웹브라우저에서 기능을 구현하여 동작 여부를 체크하는 것이 필요하다. IE4.0, IE5.5, IE6.0, Mozilla 1.5, Nescape7, Opera7, Safari1.0, Lynx2.8 등의 브라우저에서 확인해 볼 것을 권장한다. 아래 링크를 따라가면, 각 웹브라우저의 예전 버전까지 제공해 준다.

- 인터넷 익스플로러: http://browsers.evolt.org/?ie/
- 모질라: http://browsers.evolt.org/?mozilla/
- 오페라: http://browsers.evolt.org/?opera/
- 넷스케이프: http://browsers.evolt.org/?navigator/
- 사파리: http://browsers.evolt.org/?safari/
- 링스(Lynx): http://browsers.evolt.org/?lynx/

웹페이지의 간단한 스크립트 오류를 알아 내기 위해서는 모질라나 파이어버드에 있는 자바스크립트 콘솔을 이용하는 방법이 있다. 이 콘솔을 이용하면, 표준안에 근접한 방법으로 웹페이지를 디버깅 할 수 있는 장점이 있다. 파이어버드는 가벼운 웹브라우저 이므로, 파이어버드와 IE 그리고 오페라 정도로도 웹페이지 디버깅을 할 수 있다.



이러한 스크립트 디버거로 알 수 있는 것은 DOM 요소와 속성 사용 에 대한 에러 처리 같은 것이다. 만약 Javascript 문법에 대해서도 확인하고자 한다면, Strict로 처리 한다. var response = true; var response = false; 라는 코드를 Strict모드로 한 경우 흔히 나타나는 "redeclaration of var response" 에러의 경우 위의 문법을 아래와 같이 수정해 야 에러가 없어진다. var response = true; response = false;



유사한 기능을 하는 IE에서도 MS 스크립트 디버거라는 프로그램이 있다. 기본적으로 자바스크립트에 에러가 나면 아래와 같은 경고창이 나온다. 여기에는 에러가 생긴 곳(Breakpoint)의 행과 문자 위치만 나오며 특별한 에러 메시지가 표시되지 않기때문에 오류를 찾아내는 것이 쉽지는 않다. MS에서 제공하는 최신의 스크립트 디버거는 아래에서 다운로드 받을 수 있다.

http://www.microsoft.com/downloads/details.aspx?FamilyID=2f465be0-94fd-4569-b3c4-dffdf19ccd99

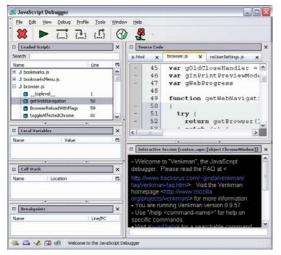
요즘 대부분의 웹사이트에서 다양한 방면의 쿠키 (Cookie)를 사용한다. IE에서는 웹사이트에 설정된 쿠키 정보를 볼 수 없으나, 모질라에서는 쿠키 관리자를 통해 설정된 쿠기 정보와 내용을 확인 할 수 있어, 웹사이트 문제 해결에 도움을 준다.

쿠키를 차단할 수도 있고, 확인할 수도 있기 때문에 관리하거나 자시의 개인정보가 빠져 나갈 수 있는 여 지가 있는지 확인 가능하다는 장점도 포함되어 있다.



그리고, HTML의 표현상 오류는 소스를 간단히 살펴 봄으로서 해결이 되는 경우가 많다. 그러나 자바스크립트 문법과 DOM의 사용상의 오류는 쉽게 알아내기 힘들다. 따라서 이러한 경우를 대비하여 디버거를 사용할 수 있다.

Mozilla와 Netscape에는 자바스크립트 콘솔 및 벤크맨(Venkman)으로 불리는 JavaScript 디버거가 내장 되어, 스크립트 개발자들에게 이용되고 있다. 이것은 화면 표시와 콘솔 양쪽에서 조작할 수 있는 디버거이다. 스크립트에서 잘못된 코드로 인해 만들어 지는 종료점(beakpoint) 괸리, Call Stack 감시 변수/객체 감시라고 하는 기능을 화면 콘솔 커멘드로 이용 가능하며, 대화형 콘솔에서는 임의의 JavaScript 코드를 실행시킬 수도 있다. 키보드 쇼트 컷은 기존의 비

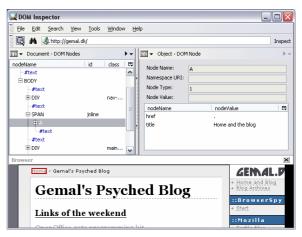


주얼 디버그 환경에 맞추고 있어 gdb 의 사용자이면 벤크맨의 break, step, next, finish, frame 및 where 커멘드를 자연스럽게 사용할수 있다.

이 JavaScript 디버거는 Windows 환경에서의 비주얼 상호 개발 환경이나 다른 대규모 웹 개발 도구보다 뛰어나 Mac OS 나 Unix 를 포함해 다른 플랫폼에 대해서는 비주얼 디버그 환경에서 이 정도까지 포괄적으로 적용 가능하다. 왼쪽의 스크린샷은 벤크맨의 실행 모습이다.

W3C의 표준 권고안인 DOM에 대한 체계적인 구조도와 웹페이지 상의 잘못된 사용 방법을 알

려주는 도구인 DOM Inspector가 모질라에 역시 내장되어 있다. 임의의 웹 문서나 XUL 어플리케이션으로 이용 중 DOM 을 정밀 조사 하거나 수정하거나 하는데 사용할 수가 있는 도구로사, 문서 및 내부의 노드를 다종 다양한 시점에서 보는 볼 수 있는 윈도우를 이용해 DOM 계층을 탐색할 수 있다. 오른쪽의 스크린샷은 전형적인 DOM 정밀 조사 작업의모습이다.



그 밖에 모질라에서는 IE와 달리 페이지 소스 보기에서 문법을 하이라이트 처리하여 별도로 확인 할 수 있으며, 캐쉬 관리자, HTTP 헤더 보기를 통해 웹서버와의 통신 과정에서 일어나는 일련의 과정을 모두 디버깅 해 볼 수 있다. 이러한 다양한 디버깅 방법들을 활용하여 보다 더 Cross browsing에 가까운 웹페이지 구현이 가능하다.

### 5. 맺음말

우리나라에서 어느 정부기관이 조사한 바에 따르면, 각 운영체제별 웹 브라우저에 따른 정부 및 공공기관, 금융기관의 정보접속성 현황은 대부분의 웹 사이트가 윈도우즈 환경 하에 익스플로러에 최적화 되어 리눅스 및 맥 OS 사용자는 정보접근에 제약이 따르는 것으로 나타났다. 특히인터넷뱅킹과 관련한 문제는 윈도우즈 기반의 공인인증서만을 현재 대부분 금융기관에서 사용하고 있어, 다른 운영체제를 사용하는 사용자들은 인터넷뱅킹을 위해서 윈도우즈 운영체제로 다시 접속해야 하는 번거로움이 있다.

이것은 표준 기술에 대한 이해 없이 시장 기술에 따라 인터넷 산업이 이끌려 옴에 따라 생긴 부작용이라고 할 수 있다. 웹 개발자들이 자신도 모르는 사이에 표준에 어긋나는 개발을 하게 되는데, 이것은 표준안에 대한 재교육과 학습과정이 결여되어 있었던 이유이기도 하다. 이 가이 드는 표준안에 대한 완벽한 설명을 담고 있지는 않지만 적어도 각 웹브라우저의 차이로 인해 야기되는 문제를 거의 대부분 다루고 있으며 이를 해결할 수 있는 방법을 제시하고 있기 때문에 이런 점들을 꼭 숙지한다면 보다 접근성이 향상된 웹사이트가 제작될 수 있을 것이다.

웹개발자가 표준 환경에 맞는 웹사이트를 구축할 수 있으려면 기본적인 마인드의 전환이 필요하다. 먼저 내용(Content : html/xhtml/xml)과 그 표현 방법(Presentation Method : CSS/XSL) 엄격하게 분리하여야 한다. 이것이 명확해야만 웹 사이트의 유지, 관리, 보수가 용이하고, 장치 독립성, 플랫폼 독립성, 접속 방법 독립성, 장애 정도와 무관한 내용에 대한 접근가능성이 보장된다.

Tim Berners-Lee가 제창한 웹의 기본 정신은 내용에 대한 '보편적 접근 가능성'(상호 운용성, 플랫폼/장치 독립성 등을 포함해서)이다. 보편적 접근 가능성(Universal Accessibility)는 결코 글자 모양, 페이지 폭, 색깔 등이 언제 어디서나 다 똑같이 보여야 한다는 것을 의미하는 것이 아니다. XML/RDF 등을 이용한 Symantic Web의 구현에도 보편적 접근 가능성은 중요하게 적용 되고 있다.

웹은 계속해서 발전하고 있다. 그러나, 한국의 웹은 상업성과 화려함에 가려져 웹이 처음 만들어 졌던 기본 정신을 외면하고, 세계적인 표준 동향을 바로 찾아가지 못한 채 한국 내부의 웹으로 전략하고 있다. 이 가이드가 국내 웹 환경의 접근성과 브라우저 호환성을 좀 더 높이는 계기가 되기를 희망한다.

# 참 고 문 헌

22.

1.	Browser News, <a href="http://www.upsdell.com/BrowserNews/stat_trends.htm">http://www.upsdell.com/BrowserNews/stat_trends.htm</a>
2.	Browser Archive, <a href="http://browsers.evolt.org">http://browsers.evolt.org</a>
3.	W3C, <a href="http://www.w3.org/TR">http://www.w3.org/TR</a>
4.	W3C WAI, http://www.w3.org/TR/WAI-WEBCONTENT
5.	US Web accessibility, <a href="http://www.section508.gov">http://www.section508.gov</a>
6.	BBC Website Guide, <a href="http://www.bbc.co.uk/guidelines/webdev/">http://www.bbc.co.uk/guidelines/webdev/</a>
7.	Netscape, <a href="http://wp.netscape.com/browsers/future/standards.html">http://wp.netscape.com/browsers/future/standards.html</a>
8.	Netscape Devedge, <a href="http://devedge.netscape.com/viewsource">http://devedge.netscape.com/viewsource</a>
9.	Mozilla Web Developers, <a href="http://www.mozilla.org/docs/web-developer">http://www.mozilla.org/docs/web-developer</a>
10.	Mozilla User Guide, <a href="http://www.gerbilbox.org/newzilla/">http://www.gerbilbox.org/newzilla/</a>
11.	Mozilla Korean Community, <a href="http://www.mozilla.or.kr">http://www.mozilla.or.kr</a>
12.	Opera Developer, <a href="http://www.opera.com/doc/">http://www.opera.com/doc/</a>
13.	Opera Korean Community, <a href="http://www.opera114.pe.kr">http://www.opera114.pe.kr</a>
14.	Apple Internet Developers, <a href="http://developer.apple.com/internet">http://developer.apple.com/internet</a>
15.	Web standards, $\underline{\text{http://www.webstandards.org/learn/resources/dom/index.html}}$
16.	Any Browser Campaign, <a href="http://www.anybrowser.org/campaign/">http://www.anybrowser.org/campaign/</a>
17.	HTML Reference, <a href="http://www.blooberry.com/indexdot/html/index.html">http://www.blooberry.com/indexdot/html/index.html</a>
18.	Javascript Reference, <a href="http://www.dannyg.com/ref/jsquickref.html">http://www.dannyg.com/ref/jsquickref.html</a>
19.	Javascript Strict warnings, <a href="http://wsabstract.com/javatutors/serror.shtml">http://wsabstract.com/javatutors/serror.shtml</a>
20.	CSS Browser Bugs, <a href="http://www.richinstyle.com/masterclass/crossbrowser.html">http://www.richinstyle.com/masterclass/crossbrowser.html</a>
21.	CSS Box Layout, http://www.thenoodleincident.com/tutorials

Quirksmode, http://www.quirksmode.org/viewport/compatibility.html

## Appendix.

## 1.최신 웹브라우저별 기능 비교 차트

	MF.7	IE 6.0	OP7	MO1.5	NN4	NS7
사용자 기반 UI 및 디자인	NO	NO	YES	YES	NO	YES
자동 팝업 차단 효과	YES	NO	YES	YES	NO	YES
윈도우내 탭 브라우징	YES	NO	YES	YES	NO	YES
강력한 마우스 기능 제공	NO	NO	YES	NO	NO	NO
간편한 개인 정보 설정 기능	YES	NO	YES	NO	NO	NO
키보드 및 마우스 설정	NO	NO	YES	NO	NO	NO
통합 검색 도구 제공	YES	NO	YES	NO	NO	NO
연구용 정보 제공	NO	NO	YES	NO	NO	NO
자동 폼 완성 기능	YES	YES	NO	NO	NO	NO
빠른 앞/뒤 네비게인션 기능	NO	NO	YES	NO	NO	NO
자동 다운로드 기능	YES	NO	YES	NO	NO	NO
자동 스타일 변경 기능	YES	NO	YES	NO	NO	NO
형식대로 찾기	YES	NO	YES	YES	NO	YES
페이지 읽어오기 정보	NO	NO	YES	NO	NO	NO
테마	YES	NO	YES	YES	NO	YES
확장기능	YES	NO	YES	YES	NO	YES
전체화면 표시	YES	YES	YES	YES	NO	YES
북마크 패널	YES	NO	YES	NO	NO	NO
도구모음 설정	YES	YES	YES	NO	NO	NO
간편한 책갈피 관리	YES	NO	YES	YES	YES	YES
암호 관리자	YES	NO	YES	YES	NO	YES
쿠키 관리자	YES	YES	YES	YES	NO	YES
페이지 확대/축소	NO	NO	YES	NO	NO	NO
표준안 지원 정도	YES	NO	YES	YES	NO	YES
광고 팝업 막기	YES	YES	NO	YES	NO	NO
스파이웨어 막기	YES	YES	YES	YES	NO	NO
개발 진척 정도	YES	NO	YES	YES	NO	NO

MF7 : Mozilla Firebird 0.7

IE 6: Internet Explorer 6.0

OP7: Opera 7.21
MO1.5: Mozilla 1.5
NN4: Netscape 4.79
NS7: Netscape 7.1

# 2. 브라우저별 HTML 표준 지원 비교표

Element	NN4	NS6	IE4	IE5	HTML3	HTML4	비고	세부 속성 내역
Α	Υ	Υ	Υ	Υ	Υ	Υ		
ABBR		Υ				Υ		
ACRONYM		Υ	Υ	Υ		Υ		
ADDRESS	Υ	Υ	Υ	Υ	Υ	Υ		
APPLET	Υ	Υ	Υ	Υ	Υ	Υ	HTML4.0에서	ALIGN, ARCHIVE, CODE,
							거부됨.	CODEBASE, HEIGHT, NAME,
								OBJECT, WIDTH
AREA	Υ	Υ	Υ	Υ	Υ	Υ		
В	Υ	Υ	Υ	Υ	Υ	Υ		
BASE	Υ	Υ	Υ	Υ	Υ	Υ		
BASEFONT	Υ		Υ	Υ	Υ	Υ	HTML4.0에서	COLOR, FACE, SIZE
							거부됨.	
BDO				Υ		Υ	NS6에서 지원 안됨	
BGSOUND			Υ	Υ				
BIG	Υ	Υ	Υ	Υ	Υ	Υ		
BLINK	Υ							
BLOCKQUOTE	Υ	Υ	Υ	Υ	Υ	Υ		
BODY	Υ	Υ	Υ	Υ	Υ	Υ		ALINK, BACKGROUND,
								BGCOLOR, LINK, TEXT, VLINK
BR	Υ	Υ	Υ	Υ	Υ	Υ		CLEAR
BUTTON		Υ	Υ	Υ		Υ		
CAPTION	Υ	Υ	Υ	Υ	Υ	Υ		ALIGN
CENTER	Υ	Υ	Υ	Υ	Υ	Υ		
CITE	Υ	Υ	Υ	Υ	Υ	Υ		
CODE	Υ	Υ	Υ	Υ	Υ	Υ		
COL		Υ	Υ	Υ		Υ		
COLGROUP		Υ	Υ	Υ		Υ		
COMMENT			Υ	Υ				
DD	Υ	Υ	Υ	Υ	Υ	Υ		
DEL		Υ	Υ	Υ		Υ		
DFN		Υ	Υ	Υ	Υ	Υ		
DIR	Υ	Υ	Υ	Υ	Υ	Υ	HTML4.0에서	COMPACT
							거부됨.	
DIV	Υ	Υ	Υ	Υ	Υ	Υ		
DL	Υ	Υ	Υ	Υ	Υ	Υ		COMPACT
DT	Υ	Υ	Υ	Υ	Υ	Υ		
EM	Υ	Υ	Υ	Υ	Υ	Υ		
EMBED	Υ	Υ	Υ	Υ				
FIELDSET		Υ	Υ	Υ		Υ		
FONT	Υ	Υ	Υ	Υ	Υ	Υ		COLOR, FACE, SIZE

OPTIONGROUP		Υ				Υ		
OL	Υ	Υ	Υ	Υ	Υ	Y		COMPACT, START, TYPE
OBJECT	Υ	Υ	Y	ľ		Y		ALIGN, BORDER, HSPACE, VSPACE
NOSCRIPT	Y	Υ 	Y Y	Y Y		Y Y		ALICN DODDED LICDAGE
NOLAYER	Y		V	V		V	세기苷	
NOFRAMES	Y	Y	Y	Y		Y	제거됨	
NOEMBED	Y							
NOBR	Y	Y	Υ	Y	Y	Y		
MULTICOL	Y							
META	Y	Y	Y	Y	Y	Y		
							거부됨.	
MENU	Υ	Υ	Υ	Υ	Υ	Y	HTML4.0에서	COMPACT
MARQUEE			Υ	Υ				
MAP	Υ	Υ	Υ	Υ	Υ	Υ		
LISTING	Υ		Υ	Υ	Υ		HTML4.0에서 제외된	======================================
LINK	Υ	Υ	Υ	Υ	Υ	Υ		
LI	Υ	Υ	Υ	Υ	Υ	Υ		TYPE, VALUE
LEGEND		Υ	Υ	Υ		Υ		ALIGN
LAYER	Υ						DIV로 대체됨	
LABEL		Υ	Υ	Υ		Υ		
KEYGEN	Υ							
KBD	Υ	Υ	Υ	Υ	Υ	Y		
IOIIIDEA	•	•	'	'	į	ı	거부됨.	, INOINI I
ISINDEX	Υ	Y	<u>.</u> Ү	<u>.</u> Ү	Υ	Y	HTML4.0에서	PROMPT
INS		<u>'</u> Ү	Y	Y	•	Y		7121011
INPUT	Υ	Υ	Y	Υ	Y	Y		VSPACE ALIGN
IMG	Υ	Υ	Υ	Υ	Υ	Υ		ALIGN, BORDER, HSPACE,
ILAYER	Y		.,	.,	.,	.,	DIV로 대체됨	ALION 20222
IFRAME		Y	Υ	Υ		Y	DIV = vil vi ri	ALIGN
<u> </u>	Y	Y	Y	Y	Y	Y		
HTML	Y	Υ	Υ	Υ	Υ	Y		VERSION
HR	Υ	Υ	Υ	Υ	Υ	Y		ALIGN, NOSHADE, SIZE, WIDTH
HEAD	Υ	Υ	Υ	Υ	Υ	Y		
H6	Υ	Υ	Υ	Υ	Υ	Y		ALIGN
H5	Υ	Υ	Υ	Υ	Υ	Y		ALIGN
H4	Υ	Υ	Υ	Υ	Υ	Y		ALIGN
H3	Υ	Υ	Υ	Υ	Υ	Y		ALIGN
H2	Υ	Υ	Υ	Υ	Υ	Y		ALIGN
H1	Υ	Υ	Υ	Υ	Υ	Y		ALIGN
FRAMESET	Υ	Υ	Υ	Υ		Υ		
FRAME	Υ	Υ	Υ	Υ		Υ		
FORM	Υ	Υ	Υ	Υ	Υ	Y		

OPTION	Υ	Υ	Υ	Υ	Υ	Y		
P	<u>.</u> Ү	Y	Y	Y	Y	Y		ALIGN
PARAM	Y	Y	Υ	Υ	Υ	Y		
PLAINTEXT	Υ		Υ	Υ	Υ		HTML4.0에서	
							제외됨.	
PRE	Υ	Υ	Υ	Υ	Υ	Υ		
Q		Υ	Υ	Υ		Υ		
RT				Υ				
RUBY				Υ				
S	Υ	Y	Υ	Υ	Υ	Υ	HTML4.0에서 거부됨.	
SAMP	Υ	Υ	Υ	Υ	Υ	Υ		
SCRIPT	Υ	Υ	Υ	Υ		Υ		
SELECT	Υ	Υ	Υ	Υ	Υ	Υ		
SERVER	Υ							
SMALL	Υ	Υ	Υ	Υ	Υ	Y		
SPACER	Υ							
SPAN	Υ	Υ	Υ	Υ		Υ		
STRIKE	Υ	Υ	Υ	Υ	Υ	Υ	HTML4.0에서 거부됨.	
STRONG	Υ	Υ	Υ	Υ	Υ	Y	711 B	
STYLE	Y	Y	Y	Y	<u>'</u>	Y		
SUB	Y	Y	Y	Y	Υ	Y		
SUP	Υ	Y	Υ	Υ	Υ	Y		
TABLE	Υ	Υ	Υ	Υ	Υ	Y		ALIGN, BGCOLOR
TBODY	Υ	Υ	Υ	Υ		Y		- ,
TD	Y	Y	Υ	Y	Y	Y		BGCOLOR, HEIGHT, NOWRAP, WIDTH
TEXTAREA	Υ	Υ	Υ	Υ	Υ	Υ		
TFOOT		Υ	Υ	Υ		Y		
TH	Υ	Y	Y	Υ	Y	Υ		BGCOLOR, HEIGHT, NOWRAP, WIDTH
THEAD		Υ	Υ	Υ		Υ		
TITLE	Υ	Υ	Υ	Υ	Υ	Υ		
TR	Υ	Υ	Υ	Υ	Υ	Υ		BGCOLOR
TT	Υ	Υ	Υ	Υ	Υ	Y		
U	Y	Y	Y	Υ	Y	Y	HTML4.0에서 거부됨.	
UL	Υ	Υ	Υ	Υ	Υ	Υ		COMPACT, TYPE
VAR	Υ	Υ	Υ	Υ	Υ	Υ		
WBR	Υ	Υ	Υ	Υ				
XML				Υ				
XMP	Υ	Υ	Υ	Υ	Υ		HTML4.0에서	
							제거됨.	

# 3. 브라우저별 CSS1/2 표준 지원 비교표

	CSS1/2			윈도	우			매킨토시				
세부 형	)목	NN4	NS6	IE5	IE6	Op4	Op5	NN4	NS6	IE4	IE5	
Basic	Concepts											
1.1	Containment in HTML	Р	Υ	Q	Q	Υ	Υ	Р	Υ	Υ	Υ	
	LINK	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	
	<style></style>	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	
	@import	N	Υ	Q	Q	Υ	Υ	N	Υ	Υ	Υ	
	<x style="dec;"></x>	В	Υ	Υ	Υ	Υ	Υ	В	Υ	Υ	Υ	
1.2	Grouping	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	
	x, y, z {dec;}	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	
1.3	<u>Inheritance</u>	В	Υ	Υ	Υ	Υ	Υ	В	Υ	Υ	Υ	
	(inherited values)	В	Υ	Υ	Υ	Υ	Υ	В	Υ	Υ	Υ	
1.4	Class selector	Υ	Υ	Q	Q	Υ	Υ	Υ	Υ	Υ	Υ	
	.class	Υ	Υ	Q	Q	Υ	Υ	Υ	Υ	Υ	Υ	
1.5	ID selector	В	Υ	В	В	Υ	Υ	В	Υ	В	Υ	
	#ID	В	Υ	В	В	Υ	Υ	В	Υ	В	Υ	
1.6	Contextual selectors	В	Υ	Υ	Υ	Υ	Υ	В	Υ	Υ	Υ	
	x y z {dec;}	В	Υ	Υ	Υ	Υ	Υ	В	Υ	Υ	Υ	
1.7	<u>Comments</u>	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	
	/* comment */	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	
Pseud	o-Classes and Pset	ıdo-Ele	ements	S								
2.1	<u>anchor</u>	Р	Υ	Υ	Υ	Р	Υ	Р	Υ	Υ	Υ	
	A:link	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	
	A:active	N	Υ	Υ	Υ	N	Υ	N	Υ	Υ	Υ	
	A:visited	N	Υ	Υ	Υ	Υ	Υ	N	Υ	Υ	Υ	
2.3	<u>first-line</u>	N	Υ	N	Υ	Υ	Υ	N	Υ	N	Υ	
	:first-line	N	Υ	N	Υ	Υ	Υ	N	Υ	N	Υ	
2.4	<u>first-letter</u>	N	Υ	N	Υ	Υ	Υ	N	Υ	N	Υ	
	:first-letter	N	Υ	N	Υ	Υ	Υ	N	Υ	N	Υ	
The C	ascade											
3.1	important	N	Υ	Υ	Υ	Υ	Υ	N	Υ	N	Υ	
	!important	N	Υ	Υ	Υ	Υ	Υ	N	Υ	N	Υ	
3.2	Cascading Order	В	Υ	Υ	Υ	Υ	Υ	В	Υ	Υ	Υ	
	Weight sorting	В	Υ	Υ	Υ	Υ	Υ	В	Υ	Υ	Υ	
	Origin sorting	В	Υ	Υ	Υ	Υ	Υ	В	Υ	Υ	Υ	
	Specificity sorting	В	Υ	Υ	Υ	Υ	Υ	В	Υ	Υ	Υ	
	Order sorting	В	Υ	Υ	Υ	Υ	Υ	В	Υ	Υ	Υ	
Font I	Properties											
5.2.2	font-family	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	
	<family-name></family-name>	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	
	<pre><generic-family></generic-family></pre>	P	Y	Υ	Y	Υ	Υ	Υ	Y	Υ	Y	

	.,		.,	.,	.,	.,	.,	.,	.,	.,	
	serif	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
	sans-serif	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
	cursive	N	Υ	Υ	Υ	Υ	Y	Υ	Υ	Y	Υ
	fantasy	N	Υ	Υ	Υ	Υ	Y	Υ	Υ	Y	Υ
	monospace	Υ	Υ	Υ	Y	Υ	Υ	Υ	Υ	Υ	Υ
5.2.3	font-style	Р	Υ	Υ	Υ	Υ	Υ	Р	Υ	Υ	Υ
	normal	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ
	italic	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ
	oblique	N	Υ	Υ	Υ	Υ	Υ	N	Υ	Υ	Υ
5.2.4	font-variant	N	Υ	Q	Q	Υ	Υ	N	Υ	Q	Υ
	normal	N	Υ	Υ	Υ	Υ	Υ	N	Υ	Υ	Υ
	small-caps	N	Υ	Q	Q	Υ	Υ	N	Υ	Q	Υ
5.2.5	font-weight	Р	Υ	Υ	Υ	Υ	Υ	Р	Υ	Υ	Υ
	normal	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ
	bold	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ
	bolder	N	Υ	Υ	Υ	Υ	Υ	N	Υ	Υ	Υ
	lighter	Υ	Υ	Υ	Υ	Υ	Υ	N	Υ	Υ	Υ
	100 – 900	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ
5.2.6	font-size	Υ	Υ	Р	Р	Υ	Υ	Υ	Υ	Q	Υ
	<absolute-size></absolute-size>	Υ	Υ	Q	Q	Υ	Υ	Υ	Υ	Q	Υ
	xx-small - xx-large	Υ	Υ	Q	Q	Υ	Q	Υ	Υ	Q	Υ
	<relative-size></relative-size>	Υ	Υ	Υ	В	Υ	Υ	Υ	Υ	Y	Υ
	larger	Υ	Υ	Υ	В	Υ	Υ	Υ	Υ	Υ	Υ
	smaller	Υ	Υ	Υ	В	Υ	Υ	Υ	Υ	Υ	Υ
	<length></length>	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ
	<percentage></percentage>	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ
5.2.7	<u>font</u>	Р	Υ	Р	Υ	Υ	Υ	Р	Υ	Q	Υ
	<font-family></font-family>	Р	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ
	<font-style></font-style>	Р	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ
	<font-variant></font-variant>	N	Υ	Q	Q	Υ	Υ	N	Υ	Q	Υ
	<font-weight></font-weight>	Р	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ
	<font-size></font-size>	Υ	Υ	Q	Р	Υ	Υ	Υ	Υ	Υ	Υ
	<li><li>e-height&gt;</li></li>	В	Υ	Υ	Υ	Υ	Υ	В	Υ	Υ	Υ
Color	and Background Pro	operti	ies								
5.3.1	<u>color</u>	Y	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ
	<color></color>	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ
5.3.2	background-color	В	Υ	Υ	Υ	Υ	Υ	В	Υ	Υ	Υ
	<color></color>	В	Υ	Υ	Υ	Υ	Υ	В	Υ	Υ	Υ
	transparent	В	Y	Y	Y	В	Y	В	Y	Υ	Y
5.3.3	background-image	Y	Y	Y	Υ	Y	Y	Y	Y	Υ	Υ
	<url></url>	Y	Y	Y	Υ	Y	Y	Y	Y	Υ	Υ
	none	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
5.3.4	background-repeat	В	Y	Y	Y	Ү	<u>.</u> Ү	<u>.</u> В	Y	Ү	Y
0.1	repeat	Y	Ү	<u>'</u> Ү	Y	Ү	<u>'</u> Ү	<u>У</u>	<u>'</u> Ү	<u>'</u> Ү	<u>'</u> Ү
	repeat-x	<u>'</u> Р	<u>'</u> Ү	Y	Y	<u>'</u> Ү	Y	P	Y	Y	Y
	ισροάι λ	•	'	'	į.	'	'	ı	'	1	'

	repeat-y	Р	Υ	Υ	Υ	Υ	Υ	Р	Υ	Υ	Υ
	no-repeat	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ
5.3.5	<u>background-</u> attachment	N	Υ	Υ	Y	Υ	Y	N	Υ	Υ	Υ
	scroll	N	Υ	Υ	Υ	Υ	Υ	N	Υ	Υ	Υ
	fixed	N	Υ	Υ	Υ	Υ	Υ	N	Υ	Υ	Υ
5.3.6	background-position	N	Υ	Υ	Υ	Υ	Υ	N	Υ	Υ	Υ
	<percentage></percentage>	N	Υ	Υ	Υ	Υ	Υ	N	Υ	Υ	Υ
	<length></length>	N	Υ	Υ	Υ	Υ	Υ	N	Υ	Υ	Υ
	top	N	Υ	Υ	Υ	Υ	Υ	N	Υ	Υ	Υ
	center	N	Υ	Υ	Υ	Υ	Υ	N	Υ	Υ	Υ
	bottom	N	Υ	Υ	Υ	Υ	Υ	N	Υ	Υ	Υ
	left	N	Υ	Υ	Υ	Υ	Υ	N	Υ	Υ	Υ
	right	N	Υ	Υ	Υ	Υ	Υ	N	Υ	Υ	Υ
5.3.7	background	Р	Υ	Υ	Υ	Р	Υ	Р	Υ	Υ	Υ
	<background-color></background-color>	В	Υ	Υ	Υ	Υ	Υ	В	Υ	Υ	Υ
	<background-image></background-image>	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ
	<background-repeat></background-repeat>	Р	Υ	Υ	Υ	Υ	Υ	Р	Υ	Υ	Υ
	<background- attachment&gt;</background- 	N	Y	Y	Y	Υ	Y	N	Υ	Υ	Y
	<background- position=""></background->	N	Υ	Υ	Υ	Υ	Υ	N	Y	Υ	Υ
Text F	Properties										
5.4.1	word-spacing	N	Υ	N	N	Υ	Υ	N	Υ	Υ	Υ
	normal	N	Υ	N	N	Υ	Υ	N	Υ	Υ	Υ
	<length></length>	N	Υ	N	N	Υ	Υ	N	Υ	Υ	Υ
5.4.2	letter-spacing	N	Υ	Υ	Υ	Υ	Υ	N	Υ	Υ	Υ
	normal	N	Υ	Υ	Υ	Υ	Υ	N	Υ	Υ	Υ
	<length></length>	N	Υ	Υ	Υ	Υ	Υ	N	Υ	Υ	Υ
5.4.3	text-decoration	Q	Υ	Р	Р	Р	Р	Q	Υ	Р	Р
	none	Q	Υ	Υ	Υ	Υ	Υ	Q	Υ	Q	Υ
	underline	Q	Υ	Υ	Υ	Υ	Υ	Q	Υ	Q	Υ
	overline	N	Υ	Υ	Υ	Υ	Υ	N	Υ	Υ	Υ
	line-through	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ
	blink	Υ	Υ	N	N	Υ	Υ	Υ	Υ	N	N
5.4.4	vertical-align	N	Υ	Р	Р	Υ	Υ	N	Υ	Р	Υ
	baseline	N	Υ	Υ	Υ	Υ	Υ	N	Υ	Υ	Υ
	sub	N	Υ	Υ	Υ	Υ	Υ	N	Υ	Υ	Υ
	super	N	Υ	Υ	Υ	Υ	Υ	N	Υ	Υ	Υ
	top	N	Υ	N	Υ	Υ	Υ	N	Υ	Υ	Υ
	text-top	N	Υ	N	Υ	Υ	Υ	N	Υ	Υ	Υ
	middle	N	Υ	N	Υ	Υ	Υ	N	Υ	Υ	Υ
	bottom	N	Υ	N	Υ	Υ	Υ	N	Υ	В	Υ
	text-bottom	N	Υ	N	Υ	Υ	Υ	N	Υ	В	Υ
	<percentage></percentage>	N	Υ	N	N	Υ	Υ	N	Υ	В	Υ
5.4.5	text-transform	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ

	uppercase	Y	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ
	lowercase	Y	Y	Y	Ү	Y	Y	Y	Ү	Y	Y
	none	Y	Υ	Υ	Y	Υ	Y	Υ	Y	Υ	Y
5.4.6	text-align	P	Υ	Υ	Υ	Υ	Υ	Р	Υ	Р	Υ
	left	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ
	right	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ
	center	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ
	justify	В	Υ	Υ	Υ	Υ	Υ	В	Υ	N	Υ
5.4.7	text-indent	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ
	<length></length>	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ
·	<percentage></percentage>	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ
5.4.8	line-height	Р	Υ	Υ	Υ	Υ	Υ	Р	Υ	Υ	Υ
	normal	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ
	<number></number>	Р	Υ	Υ	Υ	Υ	Υ	Р	Υ	Υ	Υ
	<length></length>	В	Υ	Υ	Υ	Υ	Υ	В	Υ	Υ	Υ
	<percentage></percentage>	Р	Υ	Υ	Υ	Υ	Υ	Р	Υ	Υ	Υ
Box Pı	roperties										
5.5.01	margin-top	Р	Υ	Р	Υ	Υ	Υ	Р	Υ	Р	Υ
	<length></length>	Р	Υ	Р	Υ	Υ	Υ	Р	Υ	Р	Υ
	<pre><percentage></percentage></pre>	Р	Υ	Р	Υ	Υ	Υ	Р	Υ	Р	Υ
	auto	Р	Υ	Р	Υ	Υ	Υ	Р	Υ	Р	Υ
5.5.02	margin-right	В	В	Р	Q	В	Υ	В	В	Р	Υ
	<length></length>	В	Υ	Р	Υ	Υ	Υ	В	Υ	Р	Υ
	<pre><percentage></percentage></pre>	В	Υ	Р	Υ	Υ	Υ	В	Υ	Р	Υ
	auto	N	Υ	N	Υ	Υ	Υ	N	Υ	Р	Υ
5.5.03	margin-bottom	N	Υ	Р	Υ	Υ	Υ	N	Υ	Р	Υ
	<length></length>	N	Υ	Р	Υ	Υ	Υ	N	Υ	Р	Υ
	<percentage></percentage>	N	Υ	Р	Υ	Υ	Υ	N	Υ	Р	Υ
	auto	N	Υ	Р	Υ	Υ	Υ	N	Υ	Р	Υ
5.5.04	margin-left	В	Υ	Р	Υ	В	Υ	В	Υ	Р	Υ
	<length></length>	В	Υ	Р	Υ	Υ	Υ	В	Υ	Р	Υ
	<percentage></percentage>	В	Υ	Р	Υ	Υ	Υ	В	Υ	Р	Υ
	auto	В	Υ	N	Υ	Υ	Υ	В	Υ	Р	Υ
5.5.05	<u>margin</u>	В	В	Р	Υ	В	Υ	В	В	Р	Υ
	<length></length>	В	Υ	Р	Υ	Υ	Υ	В	Υ	Р	Υ
	<percentage></percentage>	В	Υ	Р	Υ	Υ	Υ	В	Υ	Р	Υ
	auto	N	Υ	Р	Υ	Υ	Υ	N	Υ	Р	Υ
5.5.06	padding-top	В	Υ	Р	Q	Υ	Υ	В	Υ	Р	Υ
	<length></length>	В	Υ	Р	Υ	Υ	Υ	В	Υ	Р	Υ
	<percentage></percentage>	В	Υ	Р	Υ	Υ	Υ	В	Υ	Р	Y
5.5.07	padding-right	В	Υ	Р	Q	Υ	Υ	В	Υ	Р	Υ
	<length></length>	В	Υ	Р	Υ	Υ	Υ	В	Υ	Р	Υ
	<percentage></percentage>	В	Υ	Р	Υ	Υ	Υ	В	Υ	Р	Y
5.5.08	padding-bottom	В	Y	Р	Υ	Υ	Y	В	Υ	Р	Υ
	<length></length>	В	Υ	Р	Υ	Υ	Υ	В	Υ	Р	Υ

	<percentage></percentage>	В	Υ	Р	Υ	Υ	Υ	В	Υ	Р	Υ
5.5.09	padding-left_	В	Υ	Р	Υ	Υ	Υ	В	Υ	Р	Υ
	<length></length>	В	Υ	Р	Υ	Υ	Υ	В	Υ	Р	Υ
	<pre><percentage></percentage></pre>	В	Υ	Р	Υ	Υ	Υ	В	Υ	Р	Υ
5.5.10	padding	В	Υ	P	Q	Υ	Υ	В	Υ	Р	Υ
	<length></length>	В	Υ	Р	Υ	Υ	Υ	В	Υ	Р	Υ
	<pre><percentage></percentage></pre>	В	Υ	Р	Υ	Υ	Υ	В	Υ	Р	Υ
5.5.11	border-top-width	В	Υ	Р	Υ	Υ	Υ	В	Υ	Р	Υ
	thin	Υ	Υ	Р	Υ	Υ	Υ	Υ	Υ	Р	Υ
	medium	Υ	Υ	Р	Υ	Υ	Υ	Υ	Υ	Р	Υ
	thick	Υ	Υ	Р	Υ	Υ	Υ	Υ	Υ	Р	Υ
	<length></length>	Υ	Υ	Р	Υ	Υ	Υ	Υ	Υ	Р	Υ
5.5.12	border-right-width	В	Υ	Р	Υ	Υ	Υ	В	Υ	Р	Υ
	thin	Υ	Υ	Р	Υ	Υ	Υ	Υ	Υ	Р	Υ
	medium	Υ	Υ	Р	Υ	Υ	Υ	Υ	Υ	Р	Υ
	thick	Υ	Υ	Р	Υ	Υ	Υ	Υ	Υ	Р	Υ
	<length></length>	Υ	Υ	Р	Υ	Υ	Υ	Υ	Υ	Р	Υ
5.5.13	border-bottom-width	В	Υ	Р	Υ	Υ	Υ	В	Υ	Р	Υ
	thin	В	Υ	Р	Υ	Υ	Υ	В	Υ	Р	Υ
	medium	В	Υ	Р	Υ	Υ	Υ	В	Υ	Р	Υ
	thick	В	Υ	Р	Υ	Υ	Υ	В	Υ	Р	Υ
	<length></length>	В	Υ	Р	Υ	Υ	Υ	В	Υ	Р	Υ
5.5.14	border-left-width	В	Υ	Р	Υ	Υ	Υ	В	Υ	Р	Υ
	thin	Υ	Υ	Р	Υ	Υ	Υ	Υ	Υ	Р	Υ
	medium	Υ	Υ	Р	Υ	Υ	Υ	Υ	Υ	Р	Υ
	thick	Υ	Υ	Р	Υ	Υ	Υ	Υ	Υ	Р	Υ
	<length></length>	Υ	Υ	Р	Υ	Υ	Υ	Υ	Υ	Р	Υ
5.5.15	border-width	В	Υ	Р	Υ	Υ	Υ	В	Υ	Р	Υ
	thin	Υ	Υ	Р	Υ	Υ	Υ	Υ	Υ	Р	Υ
	medium	Υ	Υ	Р	Υ	Υ	Υ	Υ	Υ	Р	Υ
	thick	Υ	Υ	Р	Υ	Υ	Υ	Υ	Υ	Р	Υ
	<length></length>	Υ	Υ	Р	Υ	Υ	Υ	Υ	Υ	Р	Υ
5.5.16	border-color	Р	Υ	Υ	Υ	Υ	Υ	Р	Υ	Υ	Υ
	<color></color>	Р	Υ	Υ	Υ	Υ	Υ	Р	Υ	Υ	Υ
5.5.17	border-style	Р	Υ	Р	Υ	Υ	Υ	Р	Υ	Υ	Υ
	none	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ
	dotted	N	Υ	N	Υ	Υ	Υ	N	Υ	Υ	Υ
	dashed	N	Υ	N	Υ	Υ	Υ	N	Υ	Υ	Υ
	solid	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ
	double	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ
	groove	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ
	ridge	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ
	inset	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ
	outset	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ
5.5.18	border-top	N	Υ	Р	Υ	Υ	Υ	N	Υ	Р	Υ

	Zh	N.I.	Υ	Р	Y	Υ	Υ	N.	Υ	Р	Υ
	<pre><border-top-width></border-top-width></pre>	N	·		Y			N		-	
	   	N	Y	P P		Y	Y	N	Y	P	Y
5.5.10	<color></color>	N	Y	•	Y	Y	Y	N	Y	P	Y
5.5.19	border-right	N	Y	P	Y	Y	Y	N	Y	P	Y
	  der-right-width>	N	Y	Р	Y	Y	Y	N	Y	P	Y
	   	N	Y	Р	Y	Y	Y	N	Y	P	Y
	<color></color>	N	Y	P	Y	Y	Y	N	Y	P	Y
5.5.20	border-bottom	N	Y	P	Y	Y	Y	N	Y	P	Y
	<border-bottom- width&gt;</border-bottom- 	N	Y	Р	Y	Υ	Υ	N	Y	Р	Υ
	 border-style>	N	Υ	Р	Υ	Υ	Υ	N	Υ	Р	Υ
	<color></color>	N	Υ	Р	Υ	Υ	Υ	N	Υ	Р	Υ
5.5.21	border-left	N	Υ	Р	Υ	Υ	Υ	N	Υ	Р	Υ
	 border-left-width>	N	Υ	Р	Υ	Υ	Υ	N	Υ	Р	Υ
	 border-style>	N	Υ	Р	Υ	Υ	Υ	N	Υ	Р	Υ
	<color></color>	N	Υ	Р	Υ	Υ	Υ	N	Υ	Р	Υ
5.5.22	<u>border</u>	Р	Υ	Р	Υ	Υ	В	Р	Υ	Р	Υ
	 border-width>	В	Υ	Р	Υ	Υ	Υ	В	Υ	Р	Υ
	 border-style>	Р	Υ	Р	Υ	Υ	Υ	Р	Υ	Р	Υ
	<color></color>	Υ	Υ	Р	Υ	Υ	Υ	Υ	Υ	Р	Υ
5.5.23	<u>width</u>	Р	Υ	Р	Υ	Υ	Υ	Р	Υ	Υ	Υ
	<length></length>	Р	Υ	Р	Υ	Υ	Υ	Р	Υ	Υ	Υ
	<pre><percentage></percentage></pre>	Р	Υ	Р	Υ	Υ	Υ	Р	Υ	Υ	Υ
	auto	Р	Υ	Р	Υ	Υ	Υ	Р	Υ	Υ	Υ
5.5.24	<u>height</u>	N	Υ	Υ	Υ	Υ	Υ	N	Υ	Υ	Υ
	<length></length>	N	Υ	Υ	Υ	Υ	Υ	N	Υ	Υ	Υ
	auto	N	Υ	Υ	Υ	Υ	Υ	N	Υ	Υ	Υ
5.5.25	float	Р	Υ	Р	Q	Q	Υ	Р	Υ	В	Q
	left	В	Υ	В	Υ	Υ	Υ	В	Υ	Υ	Υ
	right	В	Υ	В	Υ	Υ	Υ	В	Υ	Υ	Υ
	none	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ
5.5.26	<u>clear</u>	Р	Υ	Р	Υ	Υ	Υ	Р	Υ	Υ	Υ
	none	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ
	left	В	Υ	В	Υ	Υ	Υ	В	Υ	Υ	Υ
	right	В	Υ	В	Υ	Υ	Υ	В	Υ	Υ	Υ
	both	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Y	Υ	Υ
Classif	ication Properties										
5.6.1	display	Р	Υ	Р	Р	Υ	Υ	Р	Υ	Р	Υ
	block	В	Υ	Υ	Υ	Υ	Υ	В	Υ	Р	Υ
	inline	N	Υ	Υ	Υ	Υ	Υ	N	Υ	N	Υ
	list-item	Р	Υ	N	N	Υ	Υ	Р	Υ	Р	Υ
	none	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ
5.6.2	white-space	Р	Υ	N	Р	Υ	Υ	Р	Υ	N	Υ
	normal	Υ	Υ	N	Υ	Υ	Υ	Υ	Υ	N	Υ
	pre	Υ	Υ	N	N	Υ	Υ	Υ	Υ	N	Υ
	nowrap	N	Υ	N	Υ	Υ	Υ	N	Υ	N	Υ
	помгар	IN	Y	IN	Y	Y	Y	IN	Y	IN	Y

## Cross Browsing 가이드

5.6.3	list-style-type	Р	Υ	Υ	Υ	Υ	Υ	Р	Υ	Υ	Υ
	disc	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ
	circle	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ
	square	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ
	decimal	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ
	lower-roman	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ
	upper-roman	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ
	lower-alpha	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ
	upper-alpha	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ
	none	Υ	Υ	Υ	Υ	Υ	Υ	В	Υ	Υ	Υ
5.6.4	list-style-image	N	Υ	Υ	Υ	Υ	Υ	N	Υ	Υ	Υ
	<url></url>	N	Υ	Υ	Υ	Υ	Υ	N	Υ	Υ	Υ
	none	N	Υ	Υ	Υ	Υ	Υ	N	Υ	Υ	Υ
5.6.5	list-style-position	N	Υ	Υ	Υ	Υ	Υ	N	Υ	Υ	Υ
	inside	N	Υ	Υ	Υ	Υ	Υ	N	Υ	Q	Υ
	outside	N	Υ	Υ	Υ	Υ	Υ	N	Υ	Υ	Υ
5.6.6	<u>list-style</u>	Р	Υ	Υ	Υ	Υ	Υ	Р	Υ	Р	Υ
	<keyword></keyword>	Υ	Υ	Υ	Υ	Υ	Υ	Р	Υ	Υ	Υ
	<position></position>	N	Υ	Q	Υ	Υ	Υ	N	Υ	Q	Υ
	<url></url>	N	Υ	Υ	Υ	Υ	Υ	N	Υ	Υ	Υ
Units											
6.1	Length Units	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ
	em	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ
	ex	Q	Υ	Q	Υ	Υ	Υ	Q	Υ	Q	Υ
	рх	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ
	in	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ
	cm	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ
	mm	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ
	pt	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ
	рс	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ
6.2	Percentage Units	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ
	<percentage></percentage>	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ
6.3	Color Units	Р	Υ	Υ	Υ	Υ	Υ	Р	Υ	Υ	Υ
	#000	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ
	#000000	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ
	(RRR,GGG,BBB)	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ
	(R%,G%,B%)	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ	Υ
	<keyword></keyword>	В	Υ	Υ	Υ	Υ	Υ	В	Υ	Υ	Υ
6.4	<u>URLs</u>	В	Υ	Υ	Υ	Υ	Υ	В	Υ	Υ	Υ
	<url></url>	В	Υ	Υ	Υ	Υ	Υ	В	Υ	Υ	Υ