

Web Standards & CSS Design

신현석, Opera Software

Purpose

◎ 목표

- 여러 브라우저에서 호환되는 CSS를 작성한다.
- 유지관리가 용이한 CSS 파일 구조를 작성한다.
- 자주 사용되는 디자인 컴포넌트들을 효율적인 코드로 작성한다.

◎ 대상

- 작업이 끝난 후 IE6에서 사이트를 열어보고 좌절한 적이 있다.
- 다른 작업자와 CSS 코드 충돌 때문에 골치 아팠던 적이 있다.
- 내가 짜고 있는 CSS 코드가 정말 맞게 짜고 있는 것인지 궁금하다.

Index

Introduction

Structure

Layout I, II

Design Components I, II

Extension

CSS Design

Why CSS Design?

Browser bugs

Tools

CSS Hack

Validation

Why use CSS?

Ideal

- ◎ 구조와 표현의 분리를 통해 HTML을 의미에 맞게 사용할 수 있다.
- ◎ 간소화, 규격화된 코드로 가독성이 높아진다.
- ◎ 코드가 간결해져서 서버 전송량이 감소된다.
- ◎ 디자인을 효과적으로 관리할 수 있다.
- ◎ 단일한 코드로 크로스 브라우징을 할 수 있다.

Why use CSS?

Reality

- ◎ 웹표준이라고 하니까.
- ◎ 웹 2.0 코딩 가능자 우대.
- ◎ IE6에서 보니까 다 깨져요.
- ◎ OOO 용 CSS 핵이 뭐예요?
- ◎ _width, *width, #width ...
- ◎ 웹 퍼블리셔가 자바스크립트도 해야 하나요?

Why use CSS?

Why?

- ◎ HTML을 의미에 맞게 작성할 수 있다.
 - CSS 보다는 HTML이 훨씬 더 중요하다.
- ◎ 디자인을 보다 정교하게 제어할 수 있다.
 - 배경이미지 위치 제어, 레이어 위치 제어 등
- ◎ 브라우저 관련 이슈 발생 시 수정이 훨씬 쉽다.
 - 물론, CSS가 구조적으로 잘 만들어졌다는 가정 하에...
- ◎ 자바스크립트와 연동이 용이하다.

Browser bugs

Browser compatibility

◎ 브라우저 버그

- 표준을 100% 구현한 브라우저는 없다.
- 표준에 모든 상황이 기술되어 있지 않다.
- 브라우저 제작사들의 해석에 차이가 있다.

◎ 비표준 코드 사용

- 특정 브라우저에서만 작동되는 코드를 사용한다.
- 브라우저 호환성을 확보하기 위해서 validation은 매우 중요하다.

Browser bugs

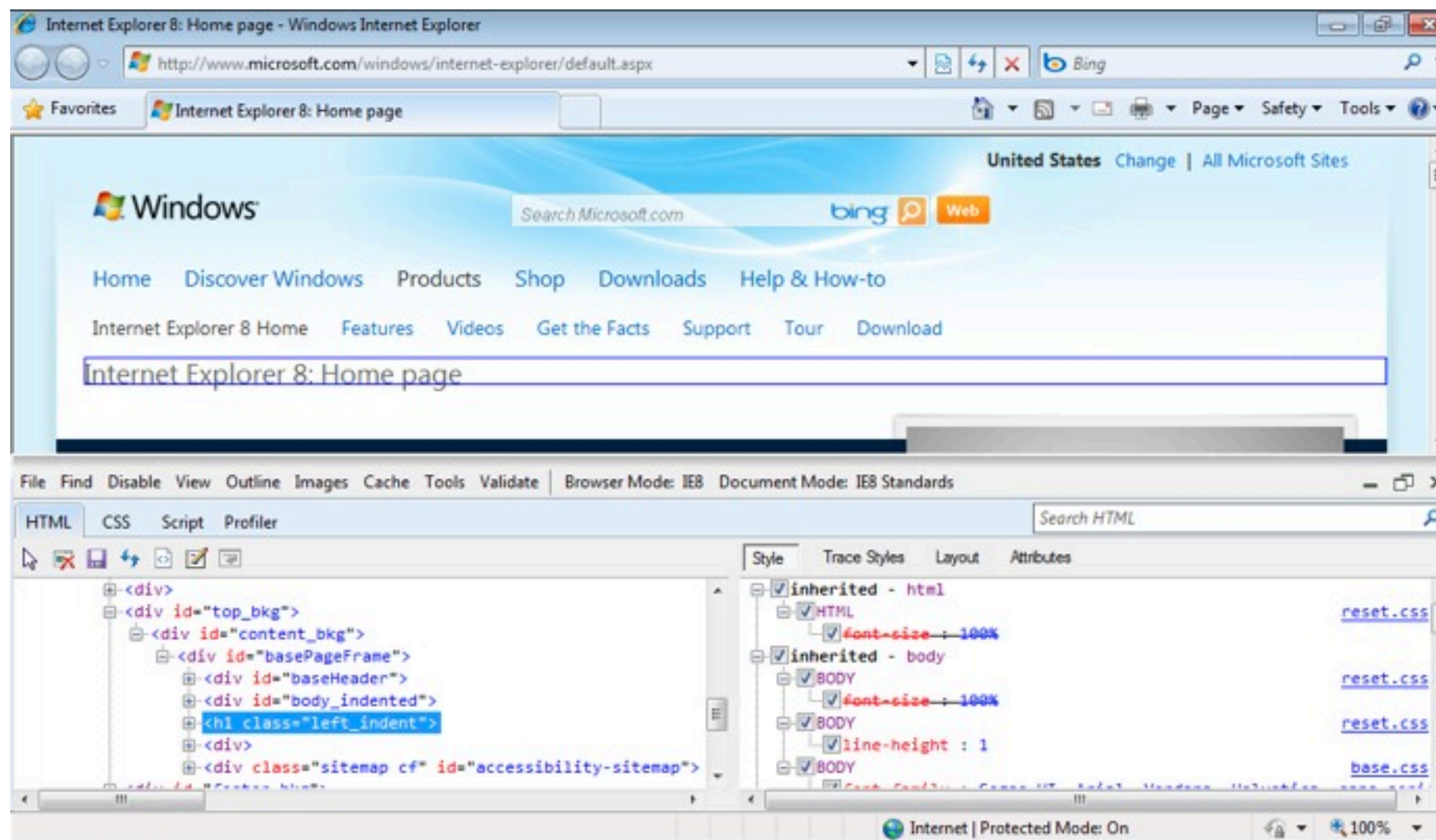
Browser compatibility

- ◎ 버그가 발생하는 부분을 최소화한다.
 - 코드 최소화(minimize)를 통해 버그 범위를 좁힌다.
- ◎ 분석툴을 통해 원인을 파악한다.
 - 분석툴에서 브라우저별로 해석된 코드를 확인 할 수 있다.
- ◎ 버그를 회피하고 다른 방법으로 구현한다.
 - CSS 핵(hack) 사용을 최소화 한다.

Tools

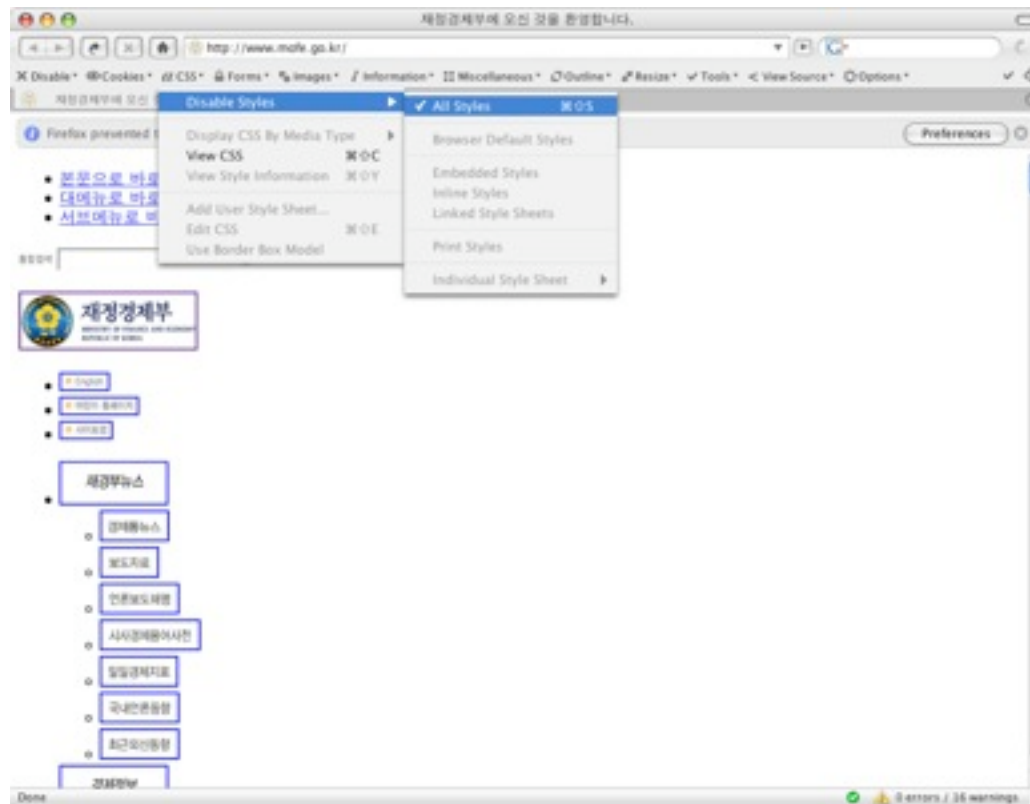
IE: Developer Tools

- Tools > Developer Tools



Firefox: Web Developer Extension

© <https://addons.mozilla.org/firefox/60/> by Chris Pederick



Disable all CSS

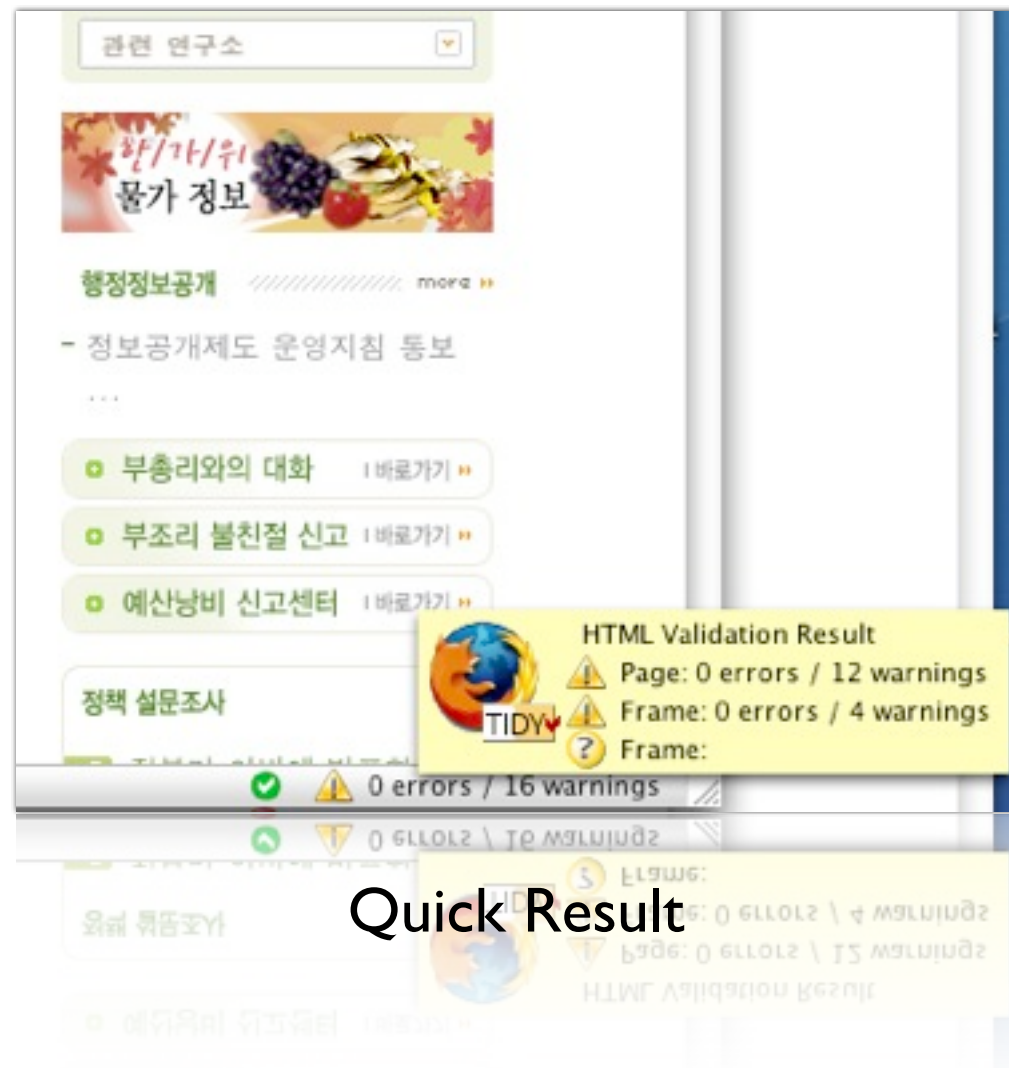


Outline block level elements

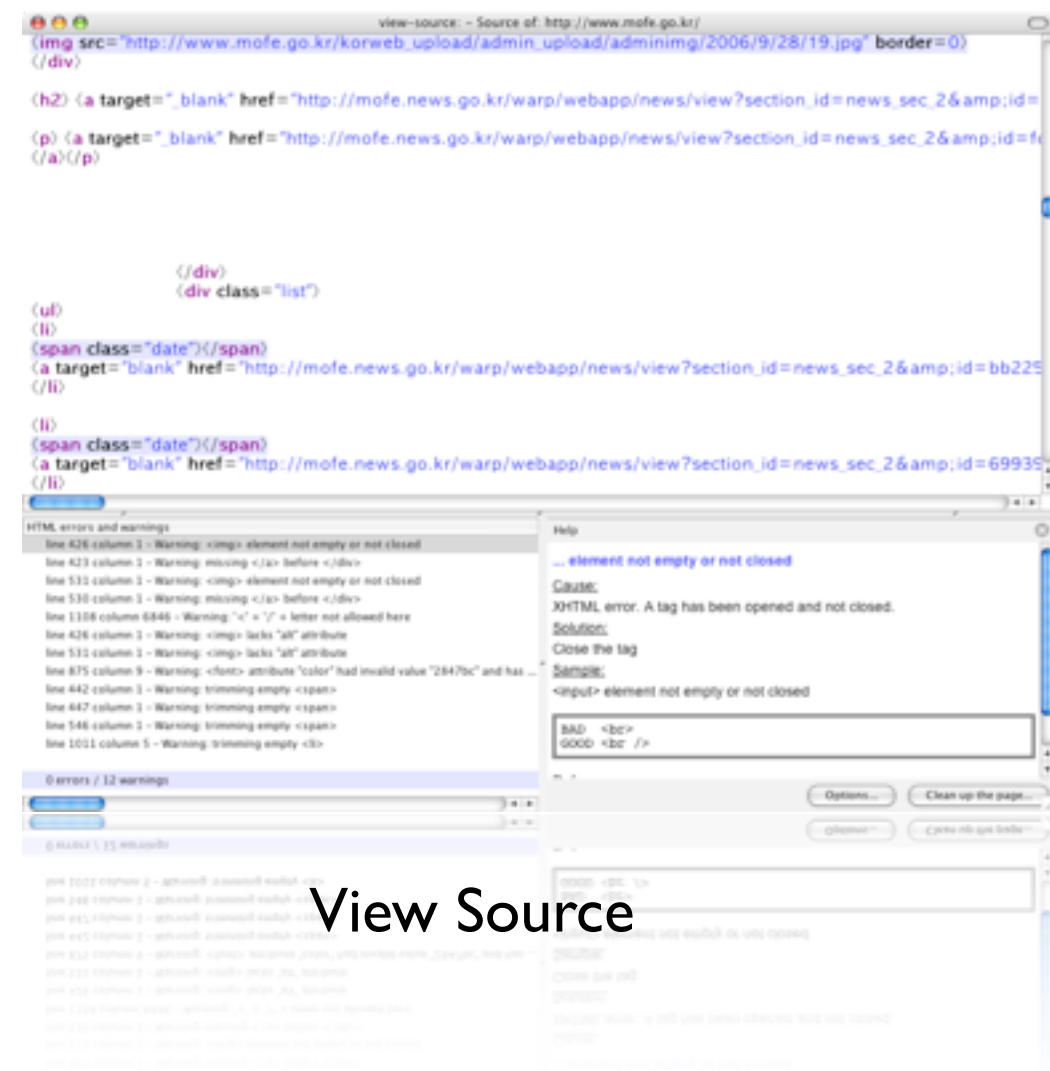
Tools

Firefox: Html Validator

- ◎ <https://addons.mozilla.org/firefox/addon/249> by Marc Gueury



Quick Result



View Source

Tools

Firefox: Firebug

◎ <http://getfirebug.com/> by Parakey, Inc.



Console

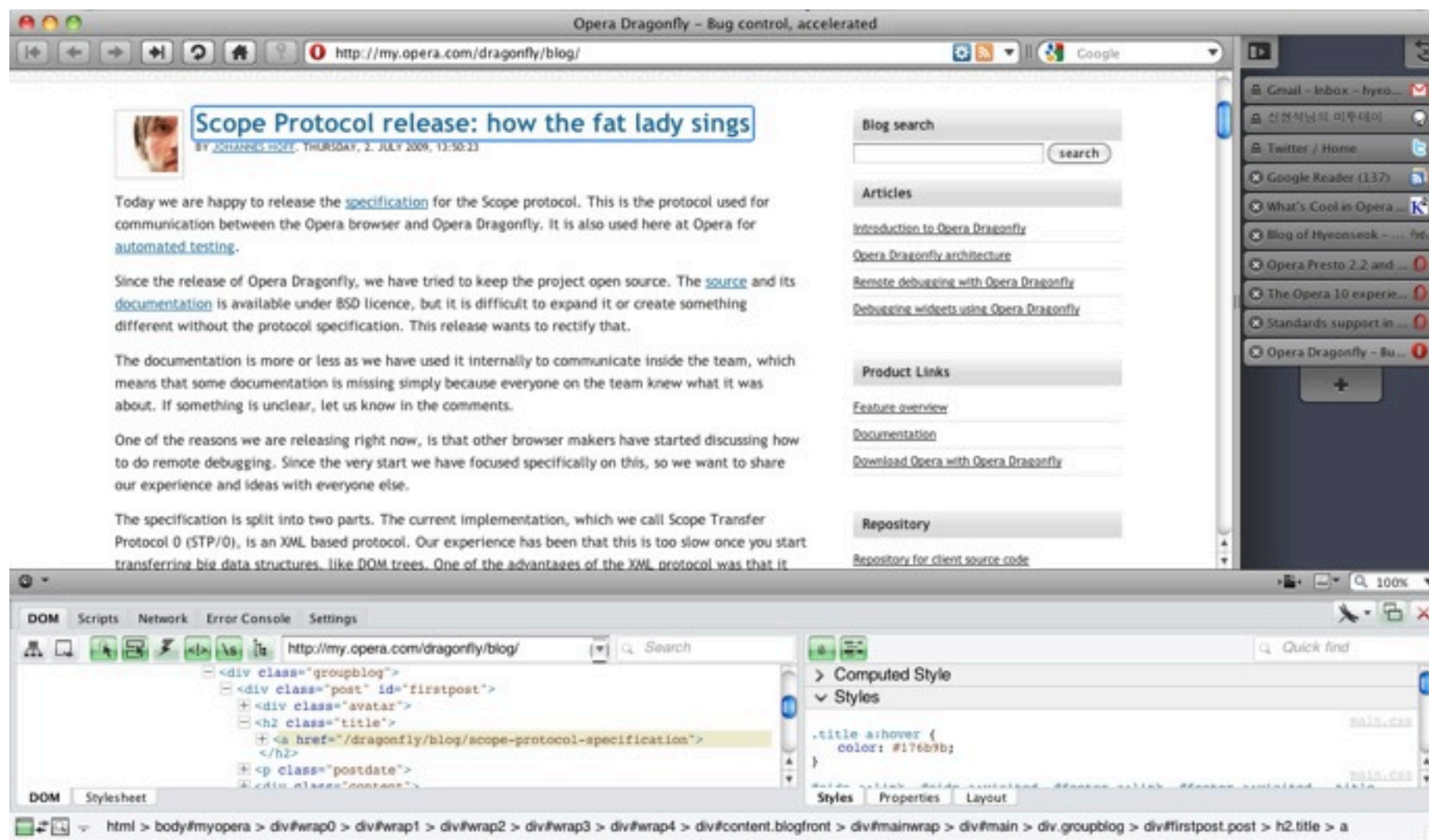


Inspector

Tools

Opera: Dragonfly

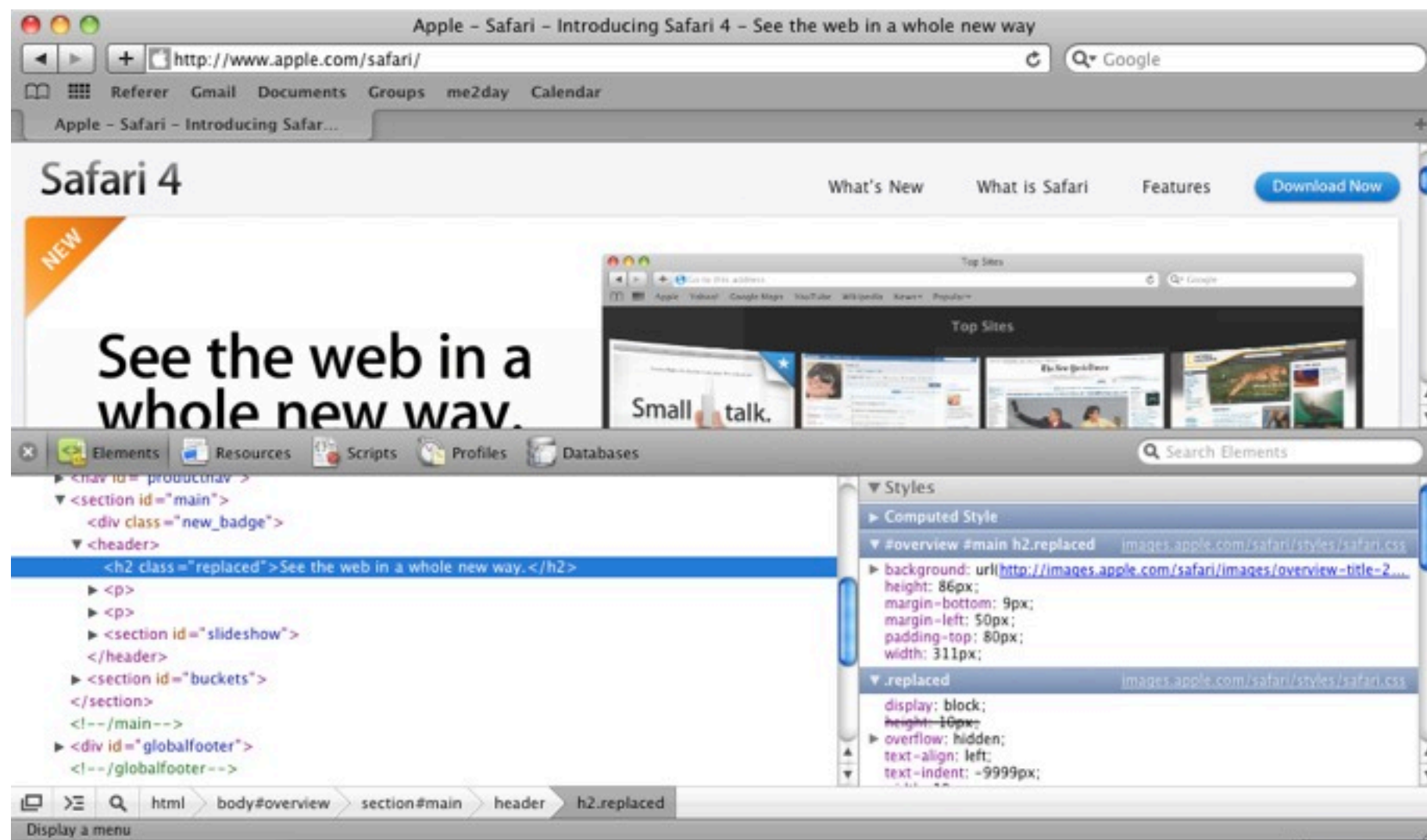
- Tools > Advanced > Developer Tools



Tools

Safari: Web Inspector

- Develop > Show Web Inspector



CSS Hack

왜 CSS hack을 사용하는가?

- ◎ 브라우저 버그, 표준의 구현정도 차이 때문에 브라우저마다 차이가 있기 때문에 사용한다.
- ◎ CSS hack은 브라우저 별로 특화된 코드를 사용해야 하기 때문에 웹 표준에서 말하는 단일 코드로 다양한 브라우저를 호환시키는 개념에 완전히 배치된다.
- ◎ 반드시 필요한 부분에만 사용하고 최대한 피한다.
- ◎ CSS hack 대신에 IE 조건부 주석(conditional comment)을 사용하는 경우: CSS hack과 크게 다르지 않기 때문에 피한다.

CSS Hack

CSS 해킹을 피하는 방법

◎ 대상 브라우저 선정

- 모든 브라우저를 대상으로 테스트 할 수는 없다.
- 호환성을 유지해야 하는 브라우저 범위를 선정한다.

◎ 기본 브라우저 선정

- 대상 브라우저 중, 가장 표준을 잘 지키는 브라우저와 표준을 지키기 않은 브라우저를 선정하여 제작과 테스트를 병행한다.

◎ 브라우저 호환성 문제를 제작 초기 단계부터 심도있게 고민하고 접근한다.

CSS Hack

CSS 핵을 피하는 방법



IE6 기반으로 제작



Firefox



Safari



Opera

브라우저 **별** 테스트



Firefox



Safari



Opera



표준으로 제작,
안전한 속성 사용

주 사용 브라우저 튜닝

Validation

W3C Validation Service

- ◎ <http://validator.w3.org/>, <http://jigsaw.w3.org/css-validator/>



W3C Markup Validation Service



W3C CSS Validation Service

Basis

Markup Language

- ◎ HTML & XHTML
- ◎ Syntax
- ◎ Document Structure

Style Sheets

- ◎ CSS Scope

Markup Language

HTML & XHTML

- ◎ 차이점과 문법적인 특징

Syntax

- ◎ 자주 틀리는 문법 오류

Document Structure

- ◎ 기본 문서 형식
- ◎ 유지보수 하기 쉬운 파일 인클루드 방식

HTML & XHTML

HTML & XHTML

- ◎ eXtensible HyperText Markup Language
- ◎ XHTML = HTML + XML
- ◎ mime-type: application/xhtml+xml
- ◎ xml 문법 기준을 따름
- ◎ 태그나 속성 등은 html을 따름

Syntax

Differences

◎ 태그의 열고 닫음이 명확해야 한다.

- Bad Case

<p>웹 표준을 생각할 때에는 사용자의 다양성을 이해하는 것이 무엇보다도 중요합니다.</p>

<p>문단을 나눌 때에는 p태그를 사용합니다.
<p>p태그를 사용할때에 닫는 태그를 생략하는 경우가 있는데 잘못된 경우 입니다.

- Good Case

<p>웹 표준을 생각할 때에는 사용자의 다양성을 이해하는 것이 무엇보다도 중요합니다.</p>

<p>문단을 나눌 때에는 p태그를 사용합니다.</p>
<p>p태그를 사용할때에 닫는 태그를 생략하는 경우가 있는데 잘못된 경우 입니다.</p>

Syntax

Differences

◎ 단독으로 사용되는 태그도 닫는 표기를 해야 한다.

- Bad Case

줄바꿈을 표하기 위해서는 `
` 태그를 사용합니다.`
`

- Good Case

줄바꿈을 표하기 위해서는 `
` 태그를 사용합니다.`
`

Syntax

Differences

◎ 속성의 값에 따옴표를 사용해야 한다.

- Bad Case

```
<table cellpadding=0 cellspacing=0 border=0>
```

- Good Case

```
<table cellpadding="0" cellspacing="0" border="0">
```

Syntax

Differences

- ◎ 대소문자를 엄격하게 구분한다.

- Bad Case

```

```

- Good Case

```

```

Syntax

Differences

- ◎ 축약형 속성을 사용하지 않는다.

- Bad Case

```
<input type="checkbox" name="hobby1" checked />
```

- Good Case

```
<input type="checkbox" name="hobby1" checked="checked" />
```

Syntax

Differences

- ◎ 스크립트와 스타일은 CDATA로 표기한다.
 - Good Case

```
<script type="text/javascript">
//<![CDATA[
if (a > b && a < 0) {      // wrong : if (a &gt; b &amp;&amp; a &lt; 0) {
    return true;
} else {
    return false;
}
//]]>
</script>
```

Syntax

HTML Entities

<	<
>	>
&	&
"	"

◎ uri안에서 &의 사용

- Bad Case

```
<a href="board_list.php?id=free&page=1&keyword=html">
```

- Good Case

```
<a href="board_list.php?id=free&amp;page=1&amp;keyword=html">
```

Syntax

HTML Entities

- ◎ value 안에서 따옴표의 사용

- Bad Case

```
<input type="text" name="keyword" value=""검색어"">
```

- Good Case

```
<input type="text" name="keyword" value="&quot;검색어&quot;">
```

Syntax

HTML Entities

- ◎ inline script안에서 따옴표의 사용

- Good Case

```

```

- ◎ escape 따옴표의 사용

- Good Case

```

```



Document Structure

General HTML Structure

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ko">

<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>기본 XHTML 문서 구조</title>
</head>

<body>
<p>기본 XHTML 문서 구조입니다.</p>
</body>

</html>
```


Document Structure

Document Type Definition

- ◎ 마크업 문서에서 사용하는 태그이름과 속성, 속성 값 등을 정의한 정보이다.

- XHTML 1.0 Strict

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

- XHTML 1.0 Transitional

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

- XHTML 1.0 Frameset

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

Document Structure

Document Type Definition

◎ 웹표준은 XHTML이다?

- HTML도 최신 권고 표준이므로 HTML도 웹표준이다.
- 문서 버전간의 우위는 없으며, 상황에 맞는 DTD를 선택해야 한다.

◎ 최신 권고 DTD

- HTML 4.01 Strict, Transitional, Frameset
- XHTML 1.0 Strict, Transitional, Frameset
- XHTML 1.1(application/xhtml+xml)
- HTML5(Working Draft)

Document Structure

HEAD & BODY

- ◎ HEAD: 문서의 정보를 나타내는 요소이다.
 - TITLE: 문서의 제목을 나타내는 요소이다.
- ◎ BODY: 문서의 내용을 담고 있는 요소이다.
 - id, class: 지시자, 연결자 역할을 한다.
 - DIV, SPAN: id, span과 함께 문서의 구조를 기술한다.
 - H1~H6: 내용 단위의 주제(제목)을 기술한다.
 - ADDRESS: 저자의 연락처를 표시한다.

Document Structure

HEAD & BODY

- ◎ 하나의 문서에는 하나의 H1만 존재할 수 있다?
 - HTML 권고안에는 H1의 갯수에 대한 언급은 없다.
 - 보통의 웹페이지는 하나의 주제를 가지고 있기 때문에 하나, 또는 2~3개의 H1이 존재하는 것이 적절하다.
- ◎ 헤딩의 단계는 순차적이어야 한다.
 - H1, H2, H1은 적절하지만, H1, H3, H1는 중간에 H2가 건너뛰어졌기 때문에 좋은 구성이 아니다.

Document Structure

Include shared code

- ◎ 사이트 전반적으로 공통적으로 사용되는 파일을 공용파일로 분류하여 사용할 때, 페이지 구조와 확장을 잘 적용할 수 있게 구성하여야 한다.
- ◎ 매 페이지마다 바뀌는 부분
 - 페이지 콘텐츠, HEAD안의 STYLE, SCRIPT 선언
- ◎ 매 페이지마다 바뀌지 않는 부분
 - DTD, HEAD안의 공용파일인클루드, 페이지 레이아웃

Document Structure

Include shared code

```
<!doctype>
<html>
<head>
<title>page title</title>
<!--#include virtual="/pagehead.html" -->
<script type="text/javascript">
// JavaScript code
</script>
<style type="text/css">
/* page specific CSS rules */
</style>
</head>
<body>
<!--#include virtual="/header.html" -->
<h1>page topic</h1>
<div id="content">
    page content
</div>
<!--#include virtual="/footer.html" -->
</body>
</html>
```

CSS Scope

CSS Selector

- ◎ CSS 셀렉터의 종류와 호환성

CSS file Structure

- ◎ 효과적인 CSS 파일 구조 작성

Priority

- ◎ CSS 우선순위를 통해 충돌을 최소화

CSS Selector

Universal selector

◎ 모든 요소 선택

```
* {  
    color: #000;  
}
```

```
<div id="body">  
  <div id="notice">  
    <ul>  
      <li>  
        <span class="head">content head</span>  
        <a href="list.html">content title</a>  
      </li>  
      <li>  
        <span class="head">content head</span>  
        <a href="list.html">content title</a>  
      </li>  
      <li>  
        <span class="head">content head</span>  
        <span>content  
          <span>title</span>  
        </span>  
      </li>  
    </ul>  
  </div>  
</div>
```


CSS Selector

Type selectors

◎ 태그명이 같은 요소 선택

```
div {  
    color: #000;  
}
```

```
<div id="body">  
  <div id="notice">  
    <ul>  
      <li>  
        <span class="head">content head</span>  
        <a href="list.html">content title</a>  
      </li>  
      <li>  
        <span class="head">content head</span>  
        <a href="list.html">content title</a>  
      </li>  
      <li>  
        <span class="head">content head</span>  
        <span>content  
          <span>title</span>  
        </span>  
      </li>  
    </ul>  
  </div>  
</div>
```

CSS Selector

Descendant selectors

◎ 하위 요소 선택

```
ul span {  
    color: #000;  
}
```

```
<div id="body">  
  <div id="notice">  
    <ul>  
      <li>  
        <span class="head">content head</span>  
        <a href="list.html">content title</a>  
      </li>  
      <li>  
        <span class="head">content head</span>  
        <a href="list.html">content title</a>  
      </li>  
      <li>  
        <span class="head">content head</span>  
        <span>content  
          <span>title</span>  
        </span>  
      </li>  
    </ul>  
  </div>  
</div>
```

CSS Selector

Child selectors

◎ 한단계 아래의 요소 선택

```
li > span {  
    color: #000;  
}
```

IE6 지원 안함

```
<div id="body">  
  <div id="notice">  
    <ul>  
      <li>  
        <span class="head">content head</span>  
        <a href="list.html">content title</a>  
      </li>  
      <li>  
        <span class="head">content head</span>  
        <a href="list.html">content title</a>  
      </li>  
      <li>  
        <span class="head">content head</span>  
        <span>content  
          <span>title</span>  
        </span>  
      </li>  
    </ul>  
  </div>  
</div>
```

CSS Selector

First-child pseudo class

◎ 첫번째 요소 선택

```
li:first-child {  
    color: #000;  
}
```

IE6 지원 안함

```
<div id="body">  
  <div id="notice">  
    <ul>  
      <li>  
        <span class="head">content head</span>  
        <a href="list.html">content title</a>  
      </li>  
      <li>  
        <span class="head">content head</span>  
        <a href="list.html">content title</a>  
      </li>  
      <li>  
        <span class="head">content head</span>  
        <span>content  
          <span>title</span>  
        </span>  
      </li>  
    </ul>  
  </div>  
</div>
```

CSS Selector

Link pseudo class

◎ 링크 요소의 상태 선택

```
a:link {
    color: #000;
}
a:visited {
    color: #000;
}
a:hover {
    color: #000;
}
a:active {
    color: #000;
}
a:focus {
    color: #000;
}
```

IE6 지원 안함

```
<div id="body">
  <div id="notice">
    <ul>
      <li>
        <span class="head">content head</span>
        <a href="list.html">content title</a>
      </li>
      <li>
        <span class="head">content head</span>
        <a href="list.html">content title</a>
      </li>
      <li>
        <span class="head">content head</span>
        <span>content
          <span>title</span>
        </span>
      </li>
    </ul>
  </div>
</div>
```

CSS Selector

Adjacent selectors

◎ 인접한 요소 선택

```
h3 + p {  
    color: #000;  
}
```

IE6 지원 안함

```
<h3>heading</h3>  
<p>This is a paragraph.</p>  
<p>This is a paragraph.</p>  
<p>This is a paragraph.</p>  
<p>This is a paragraph.</p>
```

```
<h3>heading</h3>  
<p>This is a paragraph.</p>  
<p>This is a paragraph.</p>  
<p>This is a paragraph.</p>  
<p>This is a paragraph.</p>
```

```
<h3>heading</h3>  
<p>This is a paragraph.</p>  
<p>This is a paragraph.</p>
```

CSS Selector

Attribute selectors

◎ 속성으로 요소 선택

```
input[type=text] {  
    border: 1px solid #ccc;  
}
```

IE6 지원 안함

```
<input type="text">  
<input type="password">  
<input type="radio">  
<input type="radio">  
<input type="radio">  
<input type="checkbox">  
<input type="checkbox">  
<input type="checkbox">  
<input type="text">  
<input type="text">
```

CSS Selector

Class selectors

◎ 클래스 네임으로 요소 선택

```
.head {  
    color: #000;  
}
```

```
<div id="body">  
  <div id="notice">  
    <ul>  
      <li>  
        <span class="head">content head</span>  
        <a href="list.html">content title</a>  
      </li>  
      <li>  
        <span class="head">content head</span>  
        <a href="list.html">content title</a>  
      </li>  
      <li>  
        <span class="head">content head</span>  
        <span>content  
          <span>title</span>  
        </span>  
      </li>  
    </ul>  
  </div>  
</div>
```


CSS Selector

ID selectors

◎ ID로 요소 선택

```
#notice {  
    color: #000;  
}
```

```
<div id="body">  
    <div id="notice">  
        <ul>  
            <li>  
                <span class="head">content head</span>  
                <a href="list.html">content title</a>  
            </li>  
            <li>  
                <span class="head">content head</span>  
                <a href="list.html">content title</a>  
            </li>  
            <li>  
                <span class="head">content head</span>  
                <span>content  
                    <span>title</span>  
                </span>  
            </li>  
        </ul>  
    </div>  
</div>
```

CSS Selector

:first-line pseudo element

◎ 첫번째 줄 선택

```
p:first-line {  
    color: #000;  
}
```

<p>**Littering a dark and dreary road lay the past relics** of browser-specific tags, incompatible DOMs, and broken CSS support.</p>

<p>**Today, we must clear the mind of past practices. Web** enlightenment has been achieved thanks to the tireless efforts of folk like the W3C, WaSP and the major browser creators.</p>

<p>**The css Zen Garden invites you to relax and meditate** on the important lessons of the masters. Begin to see with clarity. Learn to use the (yet to be) time-honored techniques in new and invigorating fashion. Become one with the web.</p>

CSS Selector

:first-letter pseudo element

◎ 첫번째 글자 선택

```
p:first-letter {  
    color: #000;  
}
```

<p>**L**ittering a dark and dreary road lay the past relics of browser-specific tags, incompatible DOMs, and broken CSS support.</p>

<p>**T**oday, we must clear the mind of past practices. Web enlightenment has been achieved thanks to the tireless efforts of folk like the W3C, WaSP and the major browser creators.</p>

<p>**T**he css Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the (yet to be) time-honored techniques in new and invigorating fashion. Become one with the web.</p>

CSS Selector

:first-letter pseudo element

◎ 첫번째 글자 선택

```
p:first-letter {  
  font-size: 2em;  
  float: left;  
  margin-right: 0.1em;  
  font-family: 바탕, serif;  
}
```

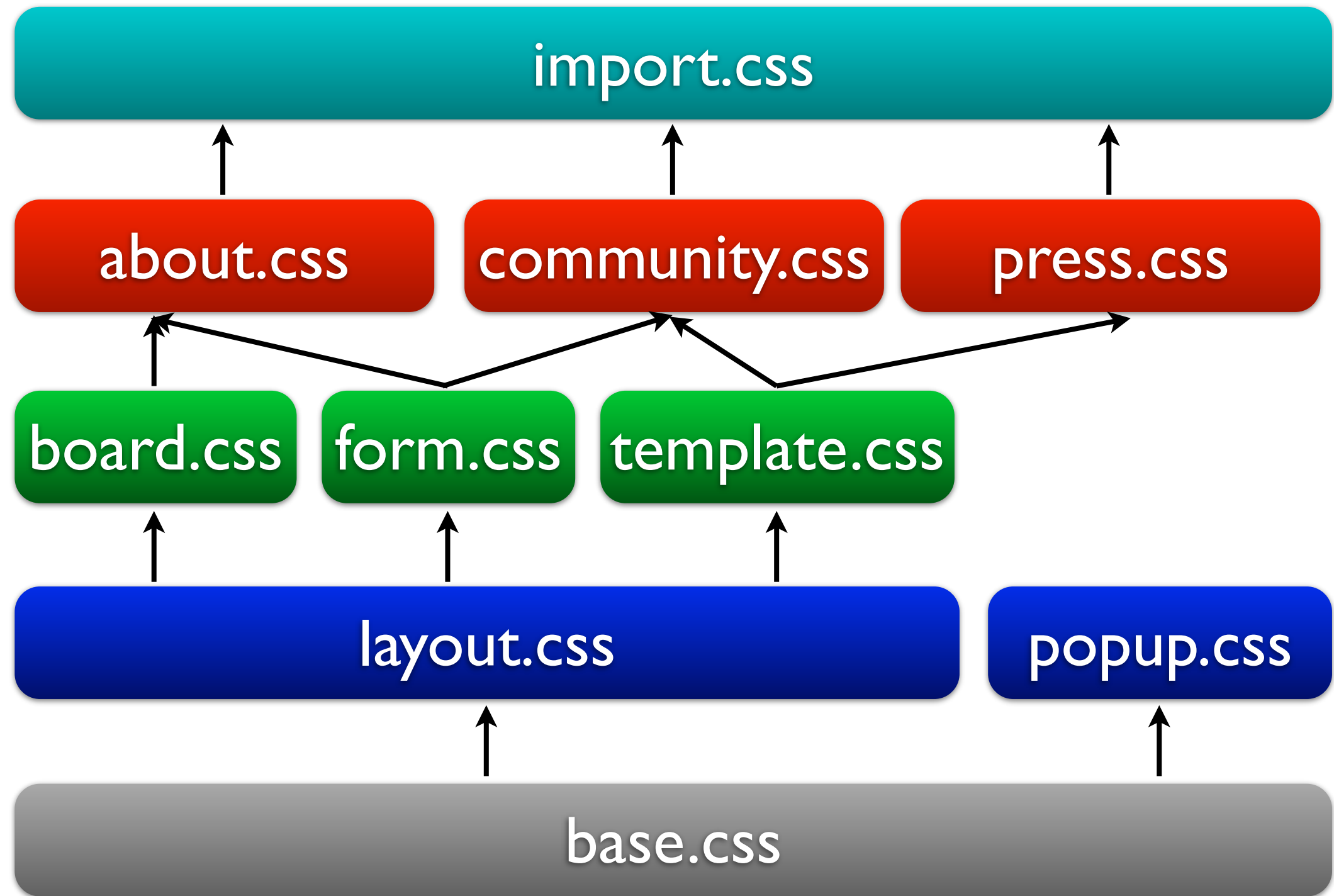


CSS file Structure

효과적인 CSS 파일 작성

- ◎ 파일의 용도를 파악할 수 있는 파일명을 사용한다.
 - 연번 보다는 섹션 이름이나 디자인 모듈로 구분한다.
- ◎ 선언된 CSS 룰의 위치를 빠르게 파악할 수 있는 순서를 사용한다.
 - 페이지 순서와 HTML 코드 안에서의 순서를 따르면 좋다.
- ◎ 충돌을 피하기 위해서 캐스캐이드 순서를 고려한다.

CSS file Structure



CSS file Structure

@import rule

```
/* layout */
@import "base.css";
@import "layout.css";

/* module */
@import "board.css";
@import "form.css";
@import "template.css";

/* section */
@import "about.css";
@import "education.css";
@import "community.css";
@import "member.css";
@import "guide.css";
```

CSS file Structure

body.class

```
body.intro {
    background-image: url (/img/06com/bg.gif);
}
body.intro #head {
    background-image: url (/img/01intro/top_menu1.jpg);
}
body.intro #body {
    background-image: url (/img/06com/sub_bodytable_img.gif);
}
body.works {
    background-image: url (/img/02work/bg.gif);
}
body.works #head {
    background-image: url (/img/02work/top_menu1.jpg);
}
body.works #body {
    background-image: url (/img/02work/sub_bodytable_img.gif);
}
body.notice {
    background-image: url (/img/03notic/bg.gif);
}
body.notice #head {
    background-image: url (/img/03notic/top_menu1.jpg);
}
body.notice #body {
    background-image: url (/img/03notic/sub_bodytable_img.gif);
}
```



CSS file Structure

section.css

◎ news.CSS

```
@charset "euc-kr";
@import url(layout.css);
/*
 * 재경부뉴스
 */
#head {
    background-image: url(/images/nav/theme/headNews.gif);
}
#top-navigation {
    background-image: url(/images/nav/theme/topNavNews.gif);
}
...
...
...
```



◎ service.CSS

```
@charset "euc-kr";
@import url(layout.css);
/*
 * 종합민원실
 */
#head {
    background-image: url(/images/nav/theme/headService.gif);
}
#top-navigation {
    background-image: url(/images/nav/theme/topNavService.gif);
}
...
...
...
```



Priority

Cascading order

◎ a, b, c, d 순으로 우선순위 결정

- style attribute: a
- ID attribute: b
- other attribute, pseudo-class: c
- element name, pseudo-element: d

```
*          {} /* a=0 b=0 c=0 d=0 -> specificity = 0,0,0,0 */
li         {} /* a=0 b=0 c=0 d=1 -> specificity = 0,0,0,1 */
li:first-line {} /* a=0 b=0 c=0 d=2 -> specificity = 0,0,0,2 */
ul li      {} /* a=0 b=0 c=0 d=2 -> specificity = 0,0,0,2 */
ul ol+li   {} /* a=0 b=0 c=0 d=3 -> specificity = 0,0,0,3 */
h1 + *[rel=up] {} /* a=0 b=0 c=1 d=1 -> specificity = 0,0,1,1 */
ul ol li.red {} /* a=0 b=0 c=1 d=3 -> specificity = 0,0,1,3 */
li.red.level {} /* a=0 b=0 c=2 d=1 -> specificity = 0,0,2,1 */
#x34y      {} /* a=0 b=1 c=0 d=0 -> specificity = 0,1,0,0 */
style=""    {} /* a=1 b=0 c=0 d=0 -> specificity = 1,0,0,0 */
```

2009년 9월 현재 working draft

Priority

ID selector + descendant selector

- ◎ ID selector와 descendant selector를 결합하여 충돌을 최소화 하는 것이 좋다.
- ◎ ID selector는 웹페이지의 가장 특징적인 모듈, 예를 들어 header, footer, sidebar 등을 정의하고,
- ◎ 각 콘텐츠별로 ID를 부여하여 충돌을 피할 수 있다.
- ◎ class는 공유되는 모듈이나 페이지 안에서 공유되는 요소에 부여함으로써, ID와 차별화 하여 사용하는 것이 좋다.
- ◎ 디자인 모듈에서 사용하는 ID와 서식 모듈에서 사용하는 ID를 사전 협의를 거쳐 충돌을 피해야 한다.

Priority

ID selector + descendant selector

```
#greeting #content-area {
    width: 670px;
    background: url(/images/forum/GreetingBg.gif) no-repeat;
}
#greeting #content-area p {
    margin: 1.5em 0 1.5em 37px;
}
/* 중략 */
#greeting #content-area p.toptext {
    color: #069;
    margin: 0 0 76px 36px;
}
/* 중략 */
#purpose #content-area h2 {
    margin: 58px 0 0 234px;
    padding-left: 35px;
    background: url(/images/forum/PurposeBullet.gif) no-repeat;
}
#purpose #content-area h2 img {
    vertical-align: middle;
}
#purpose #content-area p {
    margin: 10px 0 10px 269px;
    width: 350px;
}
/* 중략 */
#organization #content-area h2 {
    margin: 45px 0 12px 65px;
}
/* 중략 */
#activity #content-area {
    padding-top: 46px;
}
#activity #content-area h2 {
    margin: 0 0 0 182px;
    background: #DDEEF8;
    color: #036;
    font-size: 1.15em;
    padding: 0.6em 8px 0.35em 8px;
}
```

Layout Design

DTD & Rendering Mode

Float & Clear

Position

2-column layout(PR site)

3-column layout(Portal site)

DTD & Rendering Mode

Document Type Definition

- ◎ 마크업 문서에 사용되는 요소(element)와 속성(attribute)을 정의한 문서
 - ex) HTML 4.01 Strict DTD: <http://www.w3.org/TR/html4/strict.dtd>
- ◎ 모든 마크업 문서의 상단에 DTD를 선언한다.
 - ex) HTML 4.01

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">  
<html lang="en">  
<head>
```

- ex) XHTML 1.1

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ko">  
<head>
```

DTD & Rendering Mode

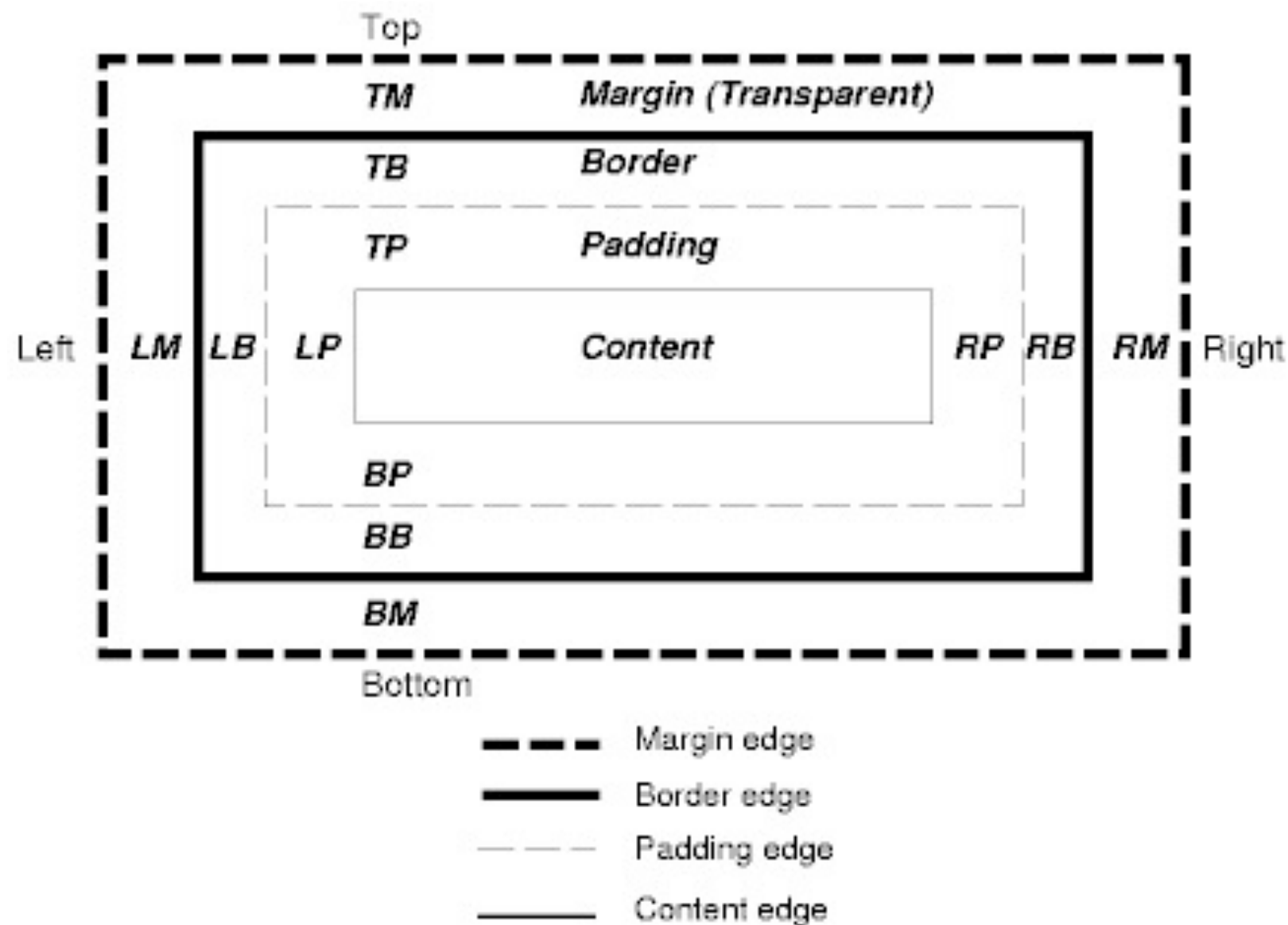
Doctype Switching

- ◎ 브라우저들은 표준의 준수와 하위 호환성 유지 사이의 차이를 극복하기 위해 여러개의 랜더링 모드(Standards mode, Almost Standards mode, Quirks mode)를 가지고 있다.
- ◎ 이 랜더링 모드는 DTD 선언에 의해서 달라진다.
- ◎ 브라우저별 독타입 스위칭 규칙
 - <http://www.webdevout.net/doctype-switching>
- ◎ IE의 경우 랜더링 모드간의 차이가 매우 크기 때문에 주의해야 한다.

DTD & Rendering Mode

Doctype Switching

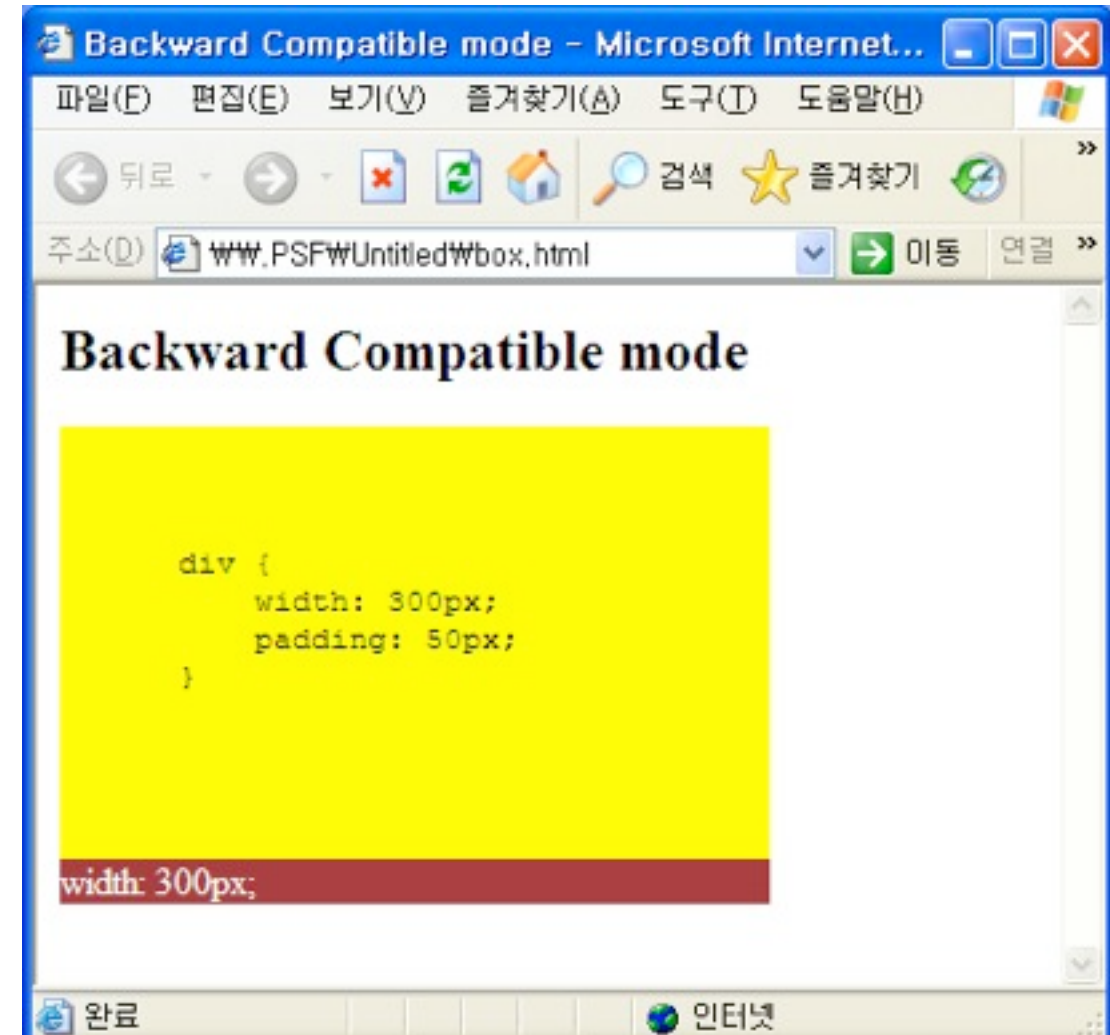
◎ CSS Box Model



DTD & Rendering Mode

Doctype Switching

- ◎ IE6 Standards mode vs. Quirks mode



Float & Clear

Float

- ◎ 박스를 좌측이나 우측으로 정렬한다.
 - float: left;
 - 좌측으로 정렬함
 - float: right;
 - 우측으로 정렬함
 - float: none;
 - float을 사용하지 않음
- ◎ 코드의 순서가 레이아웃 구성에 영향을 많이 준다.

Float & Clear

Floating



휴대전화가 롤렉스 시계처럼 자신의 부(富)와 명예, 지위를 대변하는 '욕망의 대상'으로 자리잡고 있다. 이는 최근의 '저가폰' 바람과는 정반대의 움직임으로, 휴대전화 시장도 손목시계처럼 '저가와 고가'라는 양극화를 향해 발전해 나갈 것으로 보인다.

27일 영국의 일간 가디언에 따르면 노키아는 고급 휴대전화 개발 자회사인 베르투를 통해 다이아몬드로 장식된 수제 최고급 모델 200대를 이번 주부터 선주문한 고객들에게 전달할 예정이다.



휴대전화가 롤렉스 시계처럼 자신의 부(富)와 명예, 지위를 대변하는 '욕망의 대상'으로 자리잡고 있다. 이는 최근의 '저가폰' 바람과는 정반대의 움직임으로, 휴대전화 시장도 손목시계처럼 '저가와 고가'라는 양극화를 향해 발전해 나갈 것으로 보인다.

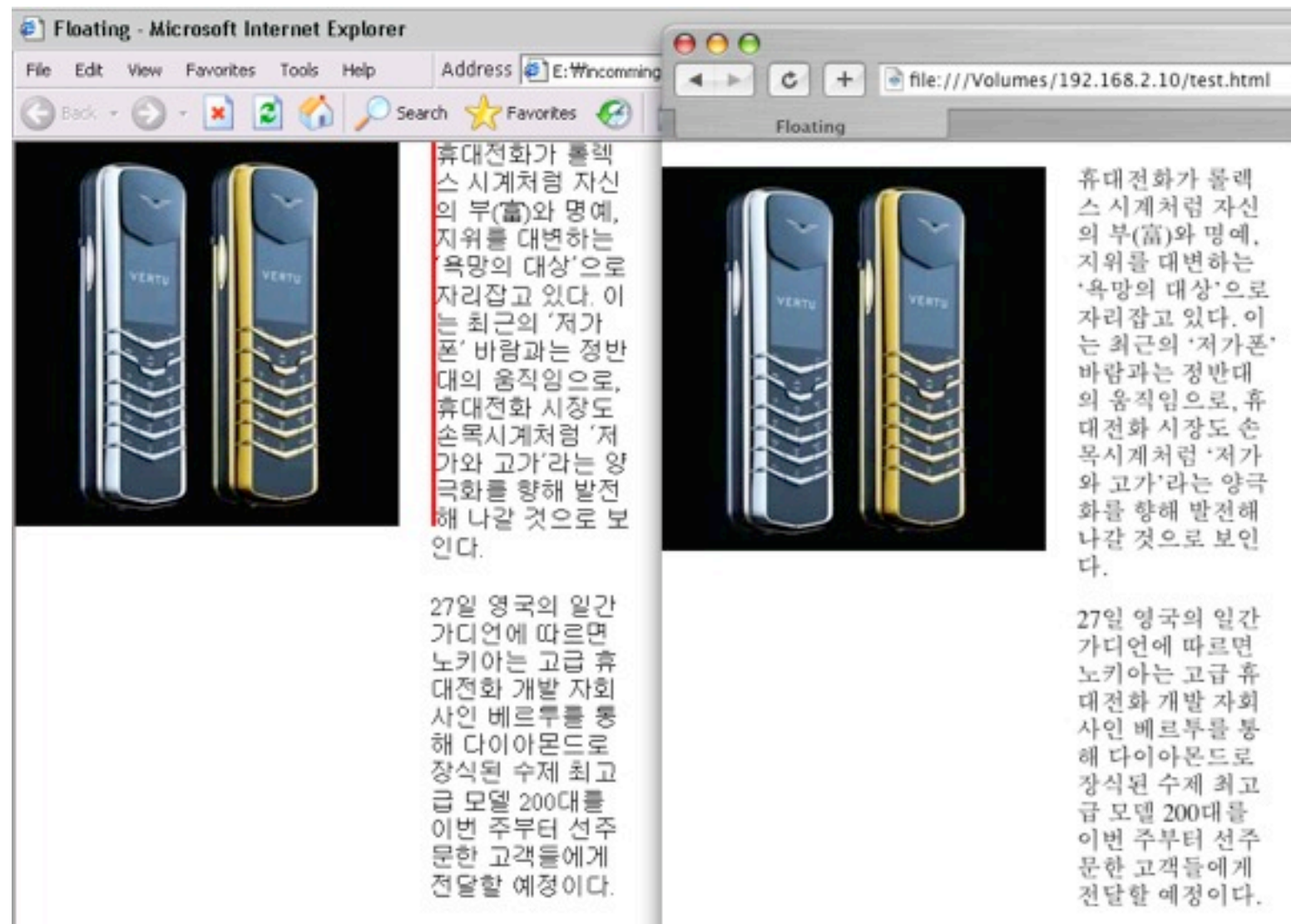
27일 영국의 일간 가디언에 따르면 노키아는 고급 휴대전화 개발 자회사인 베르투를 통해 다이아몬드로 장식된 수제 최고급 모델 200대를 이번 주부터 선주문한 고객들에게 전달할 예정이다.

```
#floating img {  
    float: left;  
    margin: 0 1em 1em 0;  
}
```

Float & Clear

IE Floating 3-pixel jog

- 텍스트를 플로트 시키지 않을 경우 IE6에서 원인을 알 수 없는 3 픽셀 여백이 생긴다.



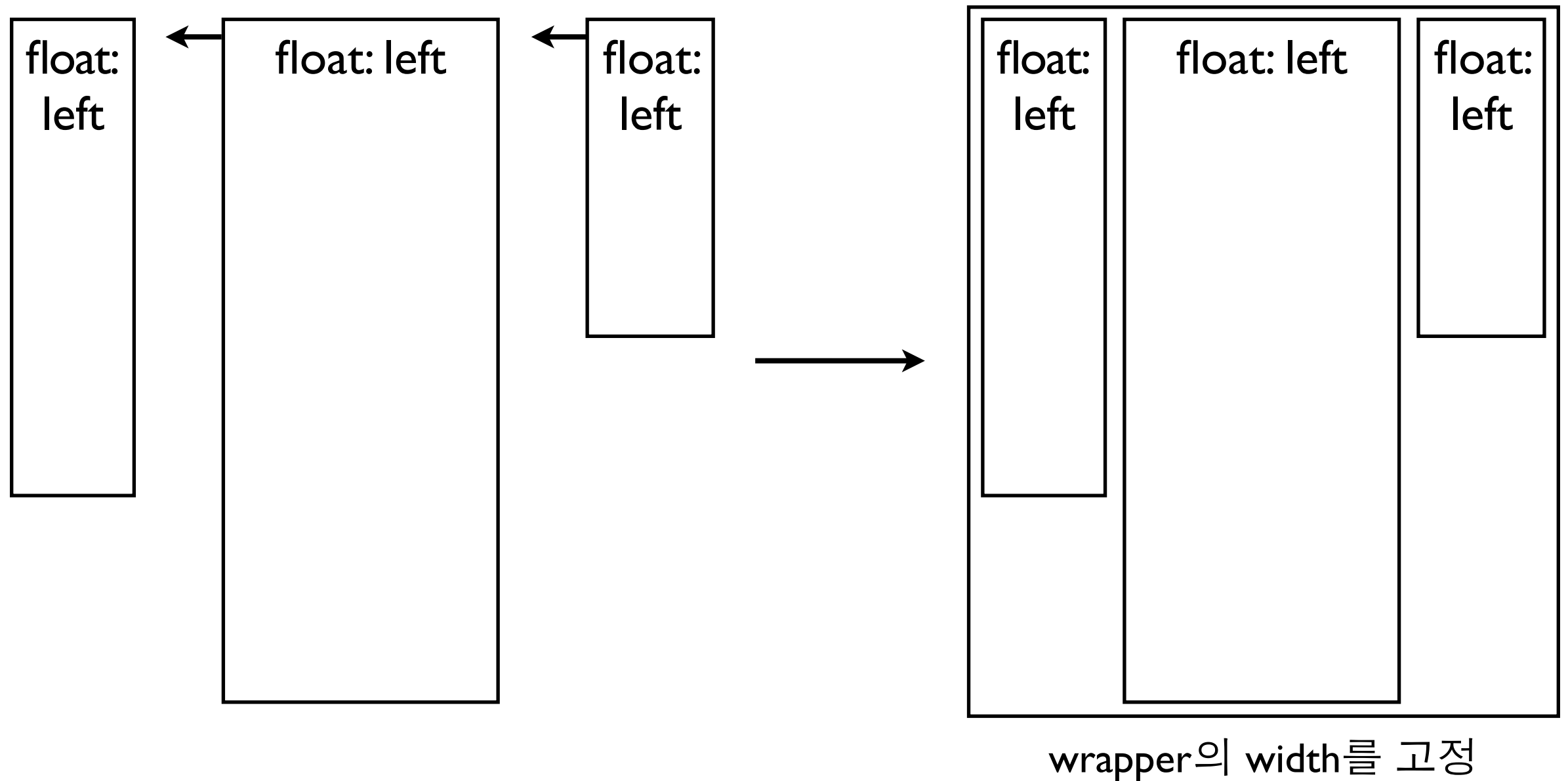
Float & Clear

Float layout

- ◎ 원래 플로트의 목적은 이미지나 작은 박스를 정렬 하는 것이었지만 레이아웃 구성용으로도 많이 쓰인다.
- ◎ 브라우저 호환성이 비교적 좋고 낮은 버전의 브라우저에서도 잘 지원된다.

Float & Clear

Float layout



Float & Clear

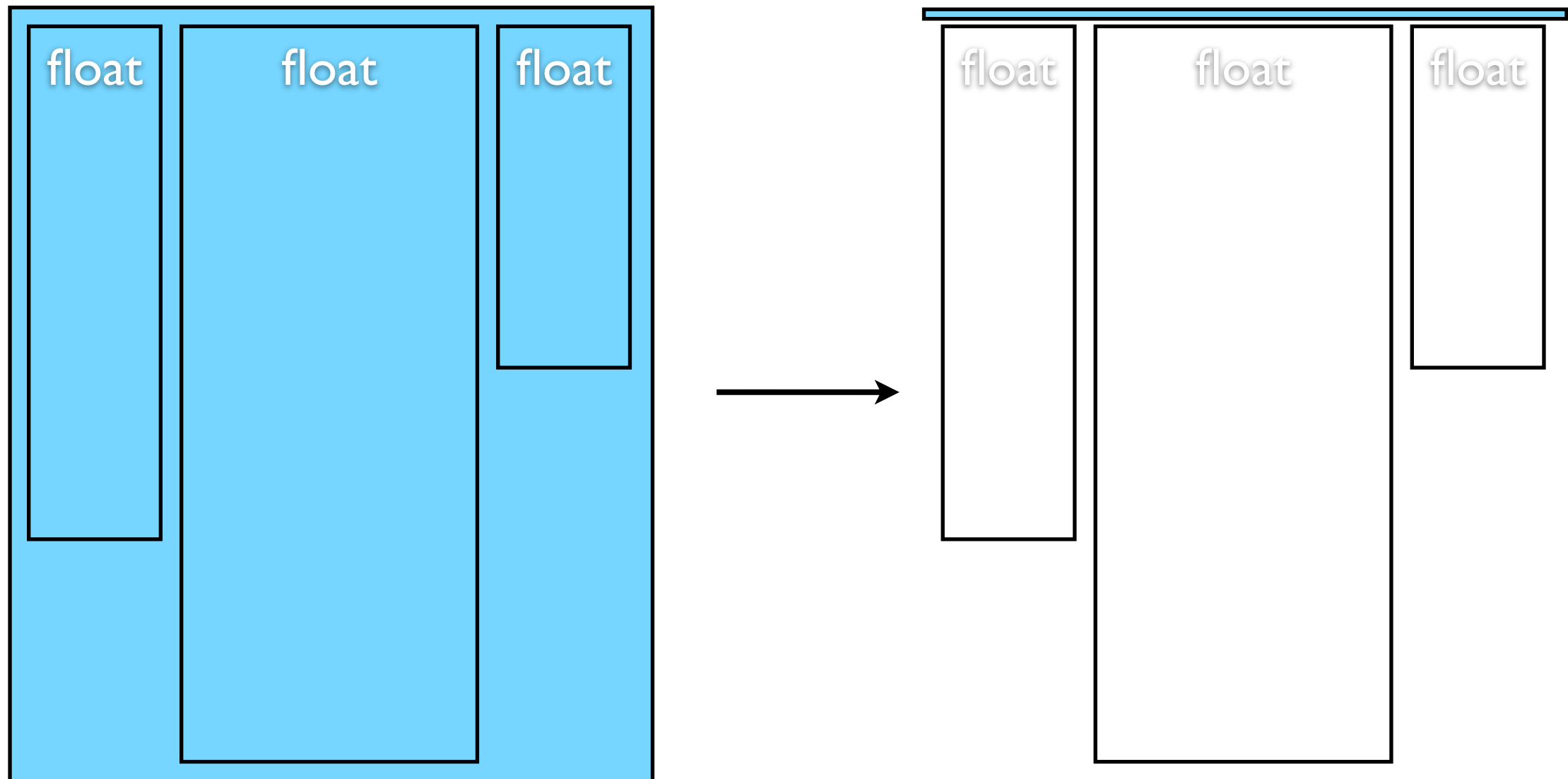
Clear

- ◎ 플로트 된 요소의 다음에서 플로트의 효과를 없애는 역할을 하는 속성
 - clear: left;
 - 좌측 플로트 효과를 없앴
 - clear: right;
 - 우측 플로트 효과를 없앴
 - clear: both;
 - 좌측과 우측 플로트 효과를 없앴

Float & Clear

Clearing

- ◎ float된 박스들은 상위 박스의 높이에 영향을 미치지 않기 때문에 배경이나 경계선 등이 의도한대로 나오지 않음

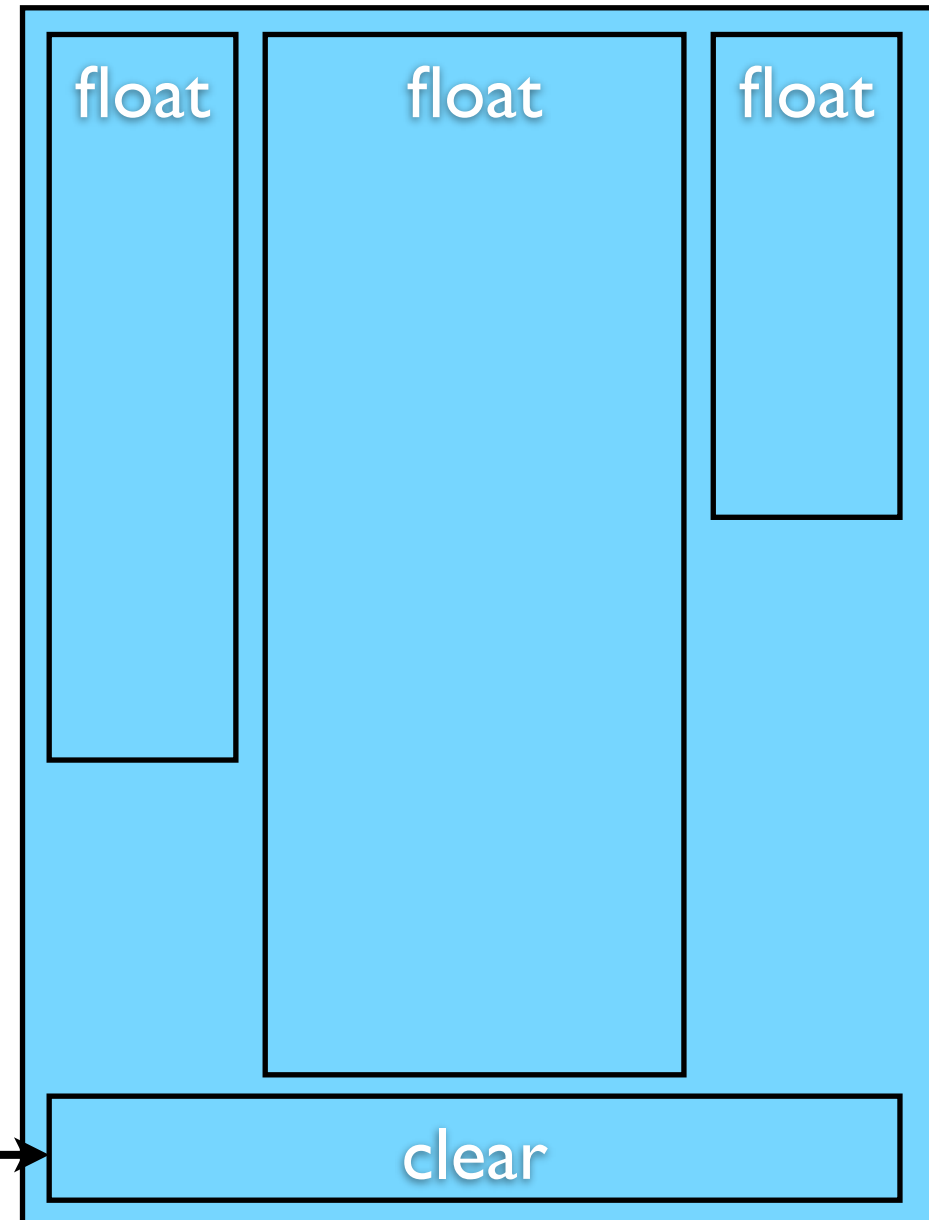


Float & Clear

Clearing

- ◎ 클리어 할 수 있는 요소를 추가하는 방법
 - 가장 효과적이지만 불필요한 요소를 추가해야 하는 단점이 있다.
 - HTML이 복잡해진다.

```
<div style="clear: both; font-size: 1px; line-height: 1px;">&nbsp;</div>
```

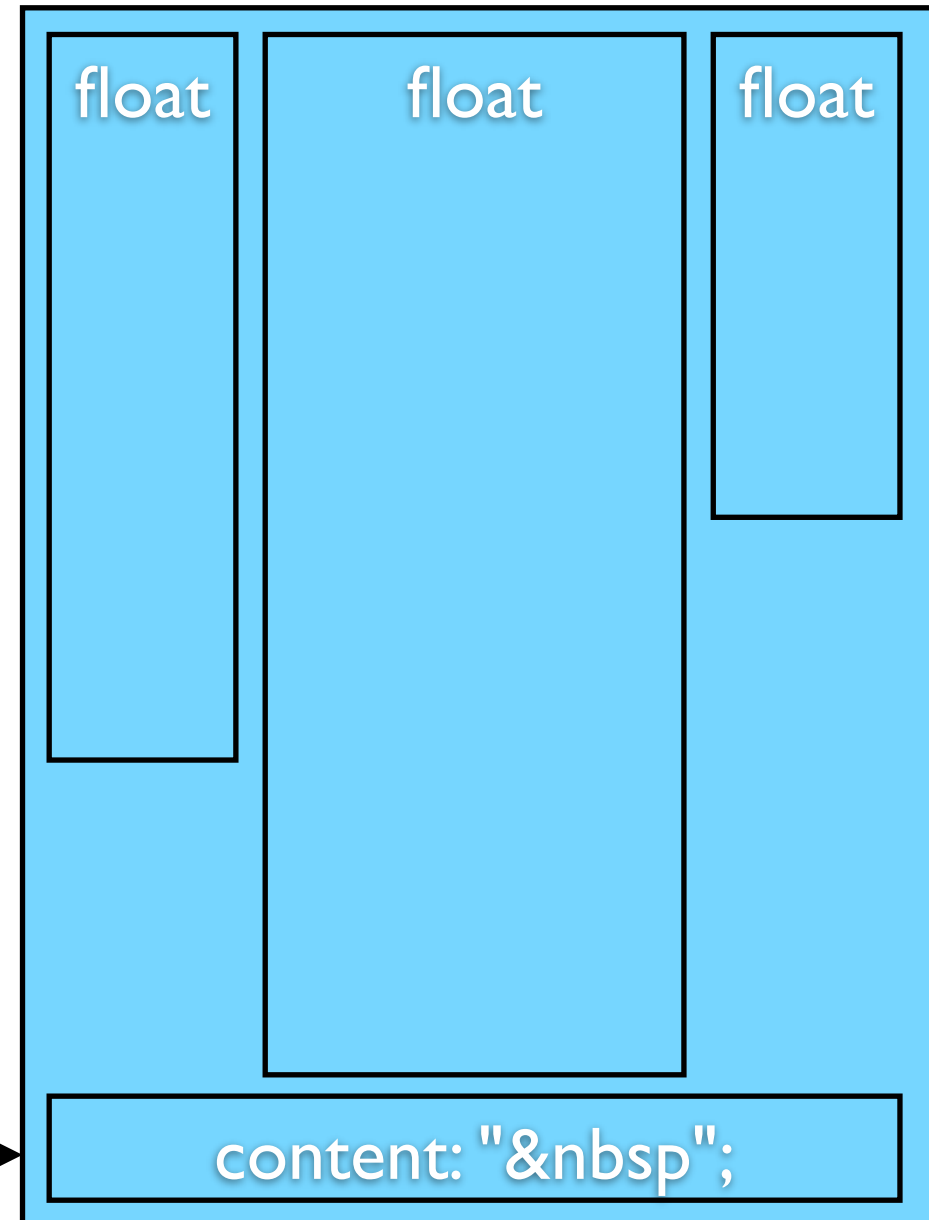


Float & Clear

Clearing

- ◎ CSS의 content 속성을 사용하는 방법
 - CSS 만으로 제어가 되기 때문에 의미상 좋은 방법이다.
 - IE가 content 속성을 지원하지 않아서 hack을 써야 한다.

```
.clearfix:after {  
  content: ".";  
  display: block;  
  height: 0;  
  clear: both;  
  visibility: hidden;  
}  
  
/* Hides from IE-mac */  
* html .clearfix {height: 1%;}  
/* End hide from IE-mac */
```

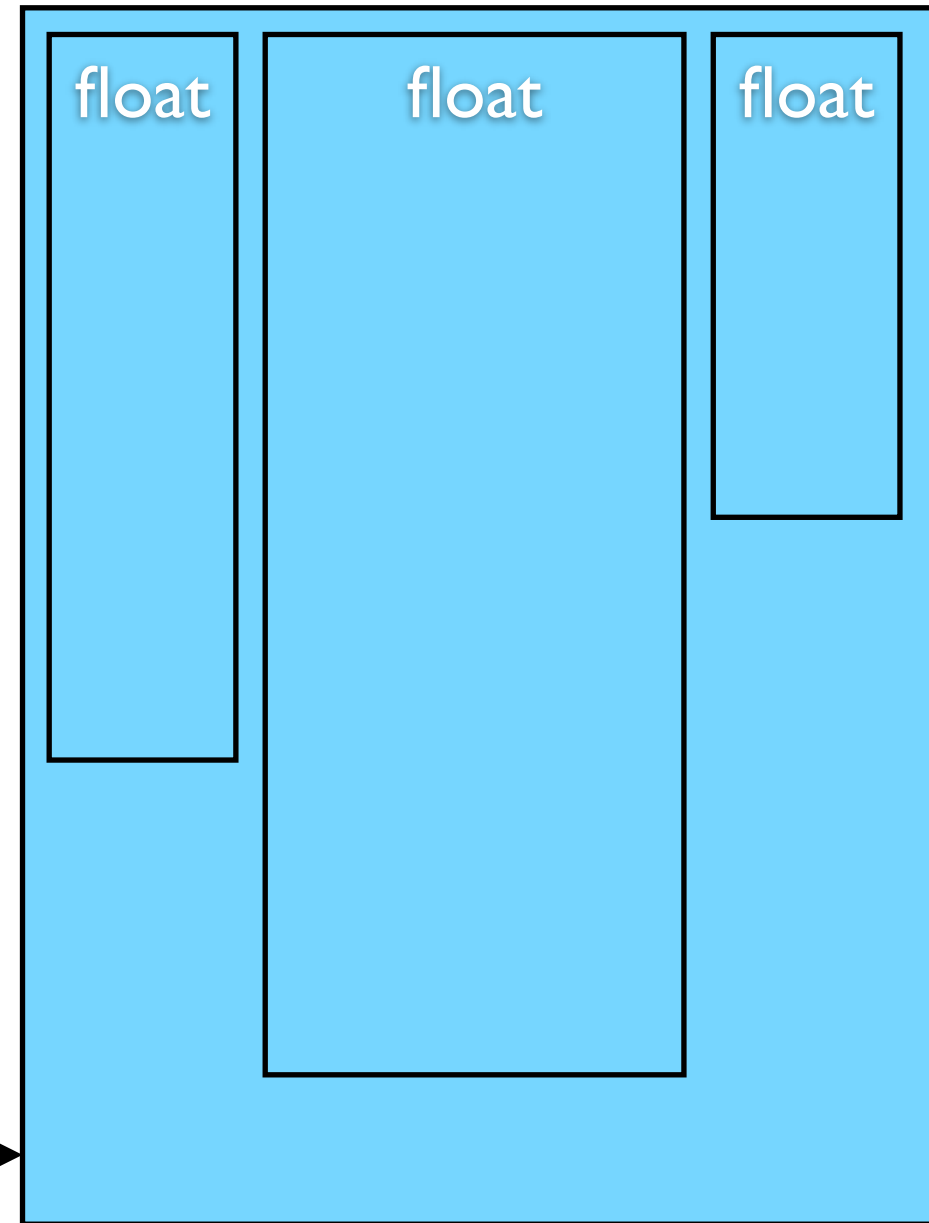


Float & Clear

Clearing

- ◎ CSS의 overflow 속성을 사용하는 방법
 - 가장 간편한 방법이다.
 - 포함하는 내용의 크기가 커질 경우 스크롤이 생기게 된다.

```
div {  
  overflow: auto;  
  width: 100%;  
}
```



Position

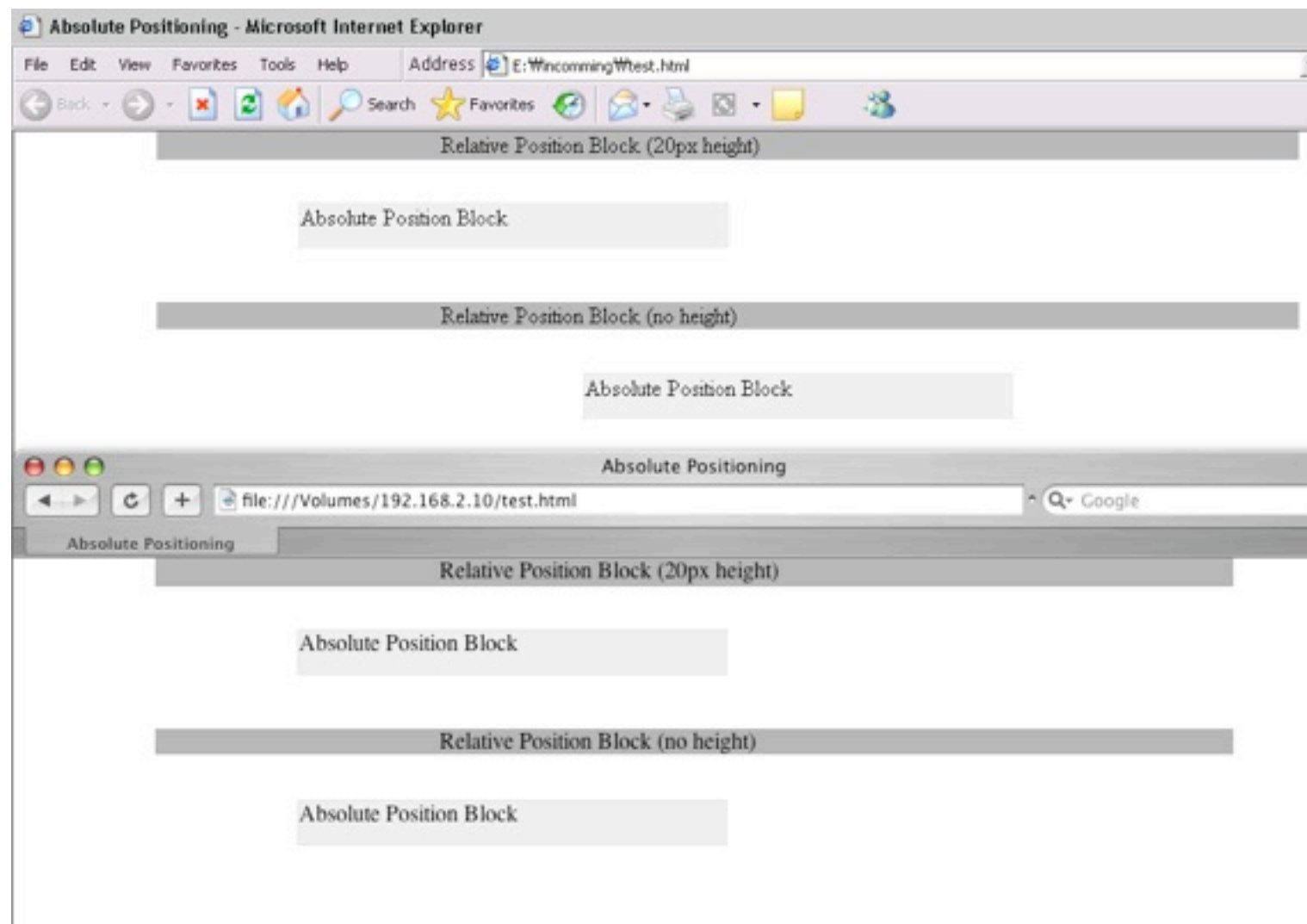
Position

- ◎ 기준으로 부터의 offset으로 element의 위치를 지정한다.
- ◎ 픽셀 단위로 자유롭게 위치 지정이 가능하다.
- ◎ 코드의 위치와 상관 없이 자유롭게 위치지정이 가능하다.
- ◎ static : 기본값, 보통의 블록들이 이 값을 가지고 있다.
- ◎ relative : static과 같지만 offset을 지정할 수 있다.
- ◎ absolute : 공간을 차지하고 있지 않으면서 offset에 의해서 위치가 지정된다. 일반적으로 레이어라고 불린다.

Position

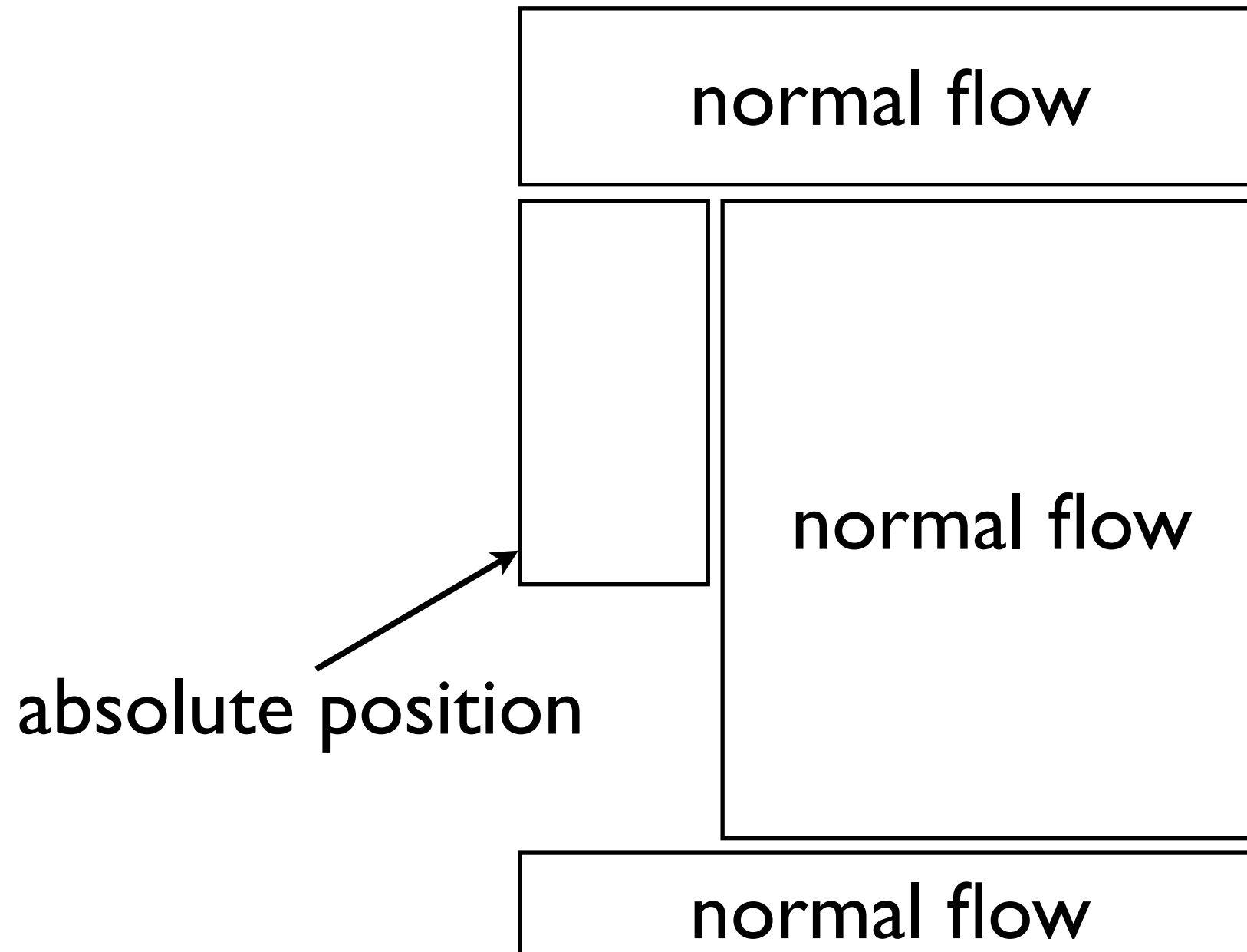
hasLayout

- IE에서 포지션 속성을 사용할 때에는 반드시 너비나 높이 속성을 주어서 박스의 형태(dimension)를 갖게 해야 한다.



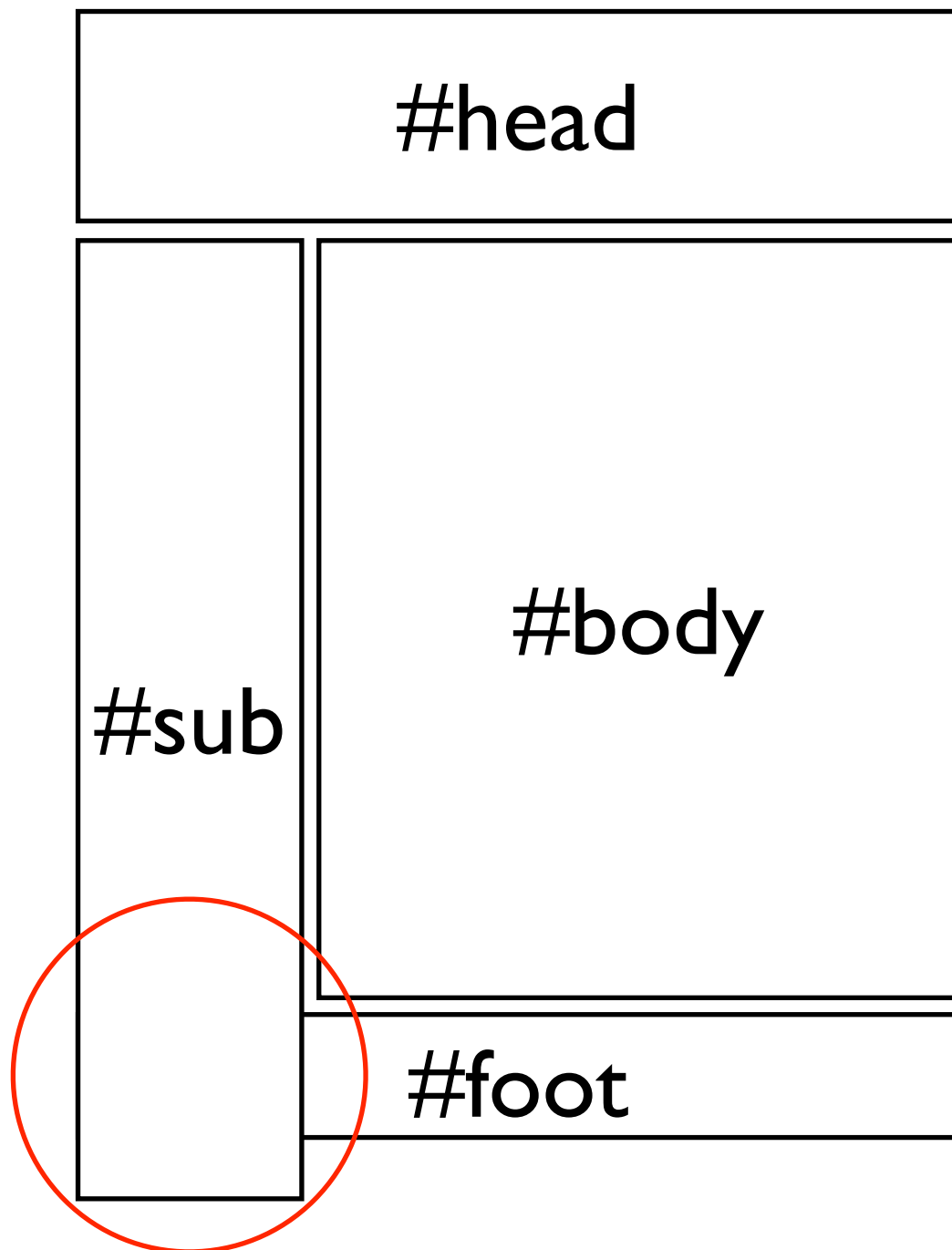
Position

Position layout



Position

Absolute position layout



```
<script type="text/javascript">
//
bodyEl = document.getElementById("body");
subEl = document.getElementById("sub");
if (bodyEl.offsetHeight &lt; subEl.offsetHeight) {
    bodyEl.style.height = subEl.offsetHeight + "px";
}
//]]&gt;
&lt;/script&gt;</pre></div><div data-bbox="838 932 991 958" data-label="Page-Footer"><p><img alt="Logo" data-bbox="845 938 875 958"/> <a href="http://hyeonseok.com">http://hyeonseok.com</a></p></div><div data-bbox="977 975 997 992" data-label="Page-Footer"><p>79</p></div>
```

Layout grid

Table grid vs. CSS box model

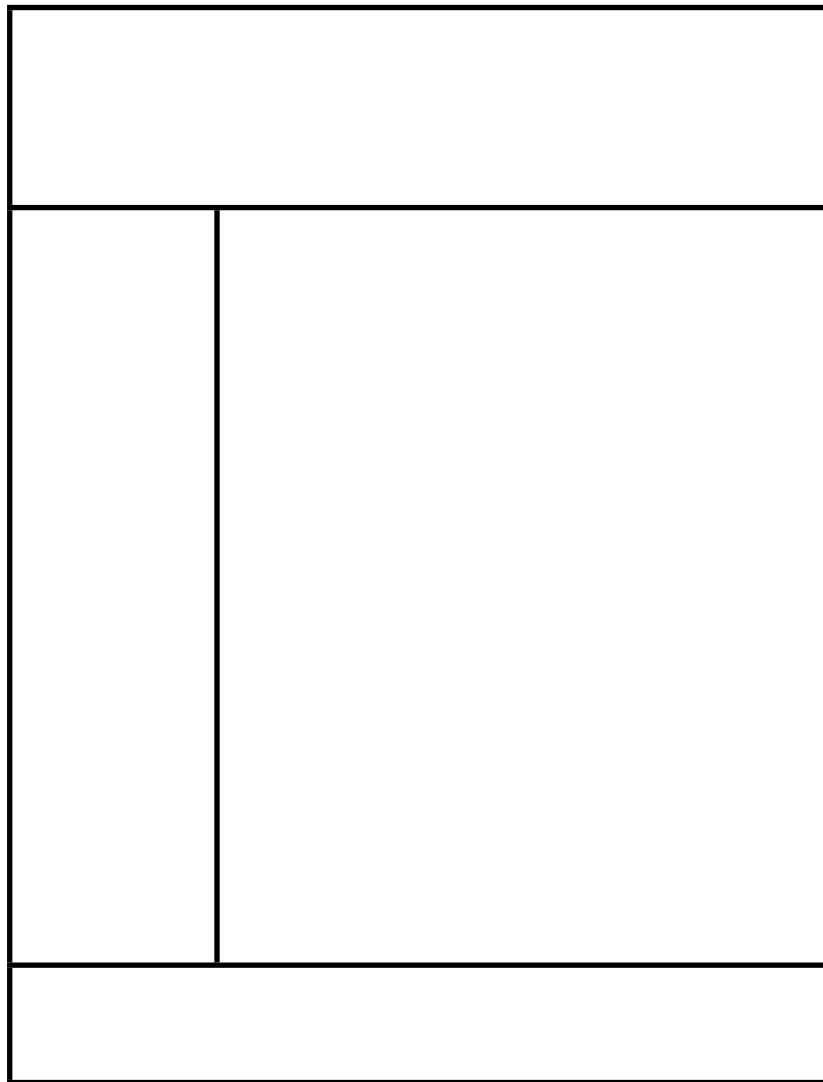
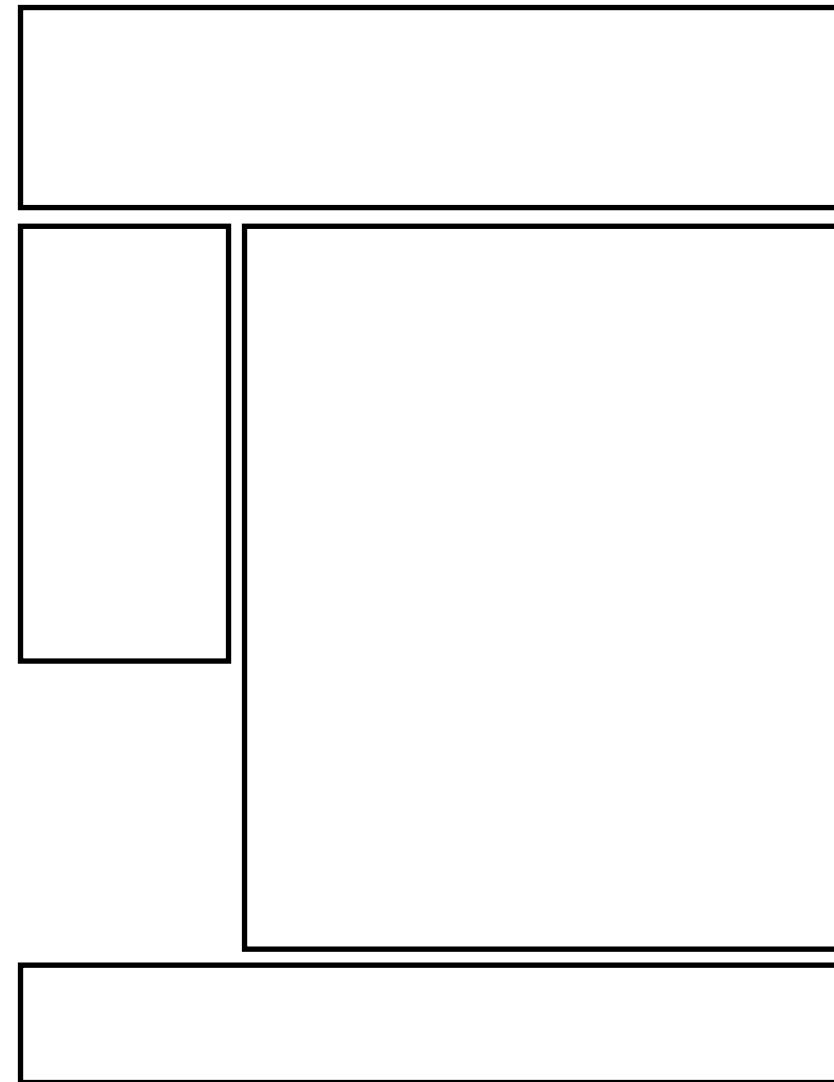


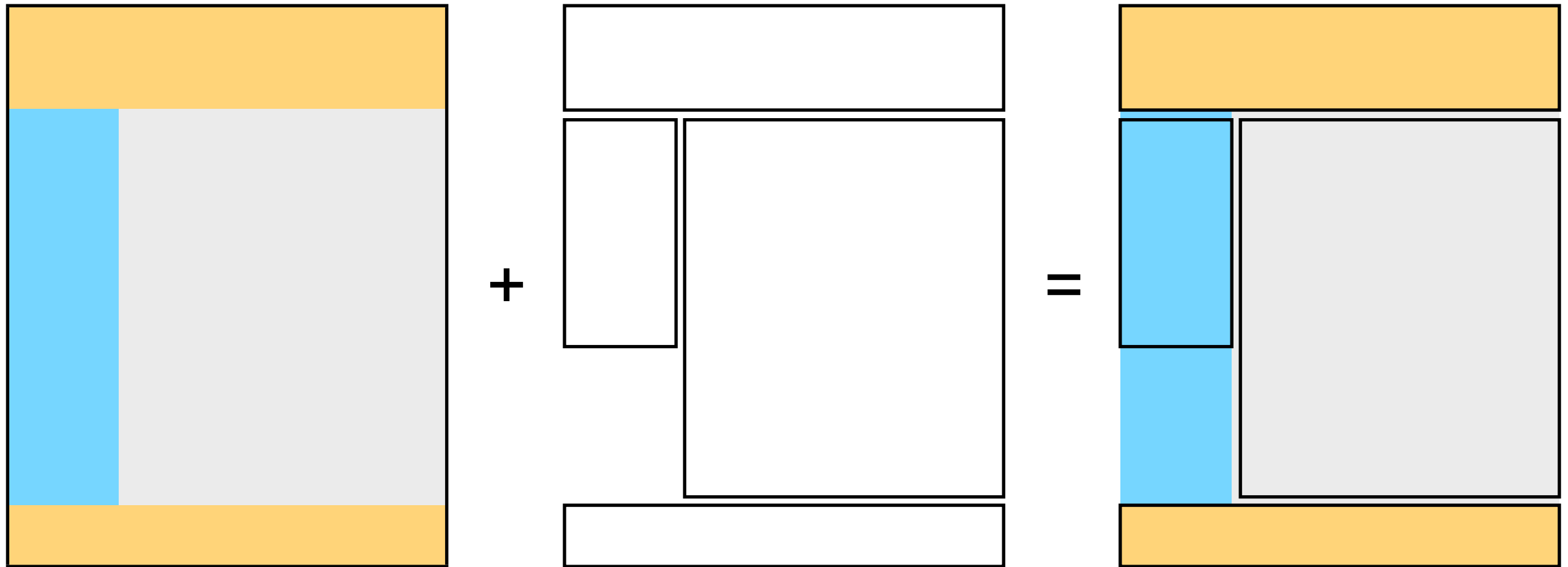
table 사용시



CSS 사용시

Layout grid

Table grid vs. CSS box model



배경이미지

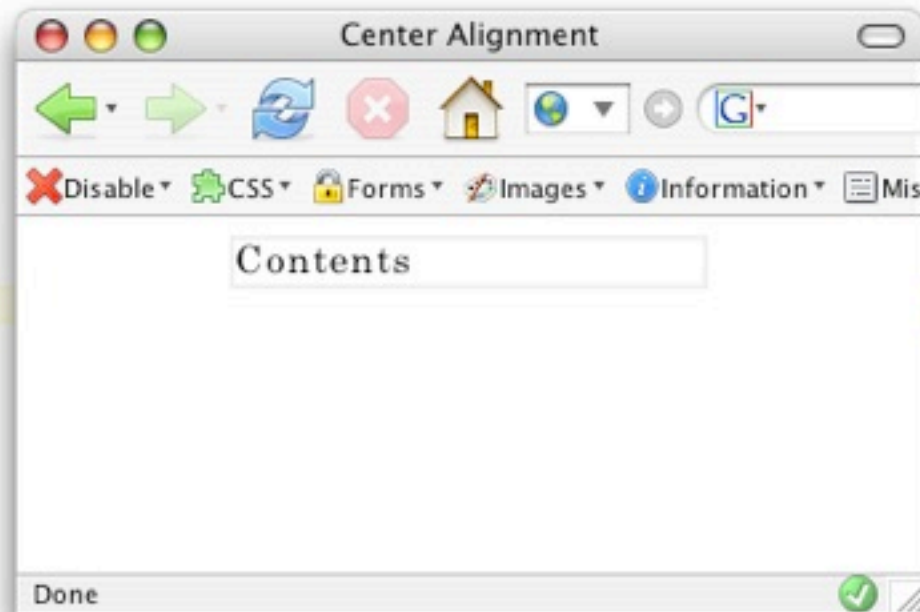
구성요소

Block alignment

Center Alignment

- IE5.5 이하 버전(Quirks mode)은 body에 text-align: center; 가 있어야 같은 효과가 나옴

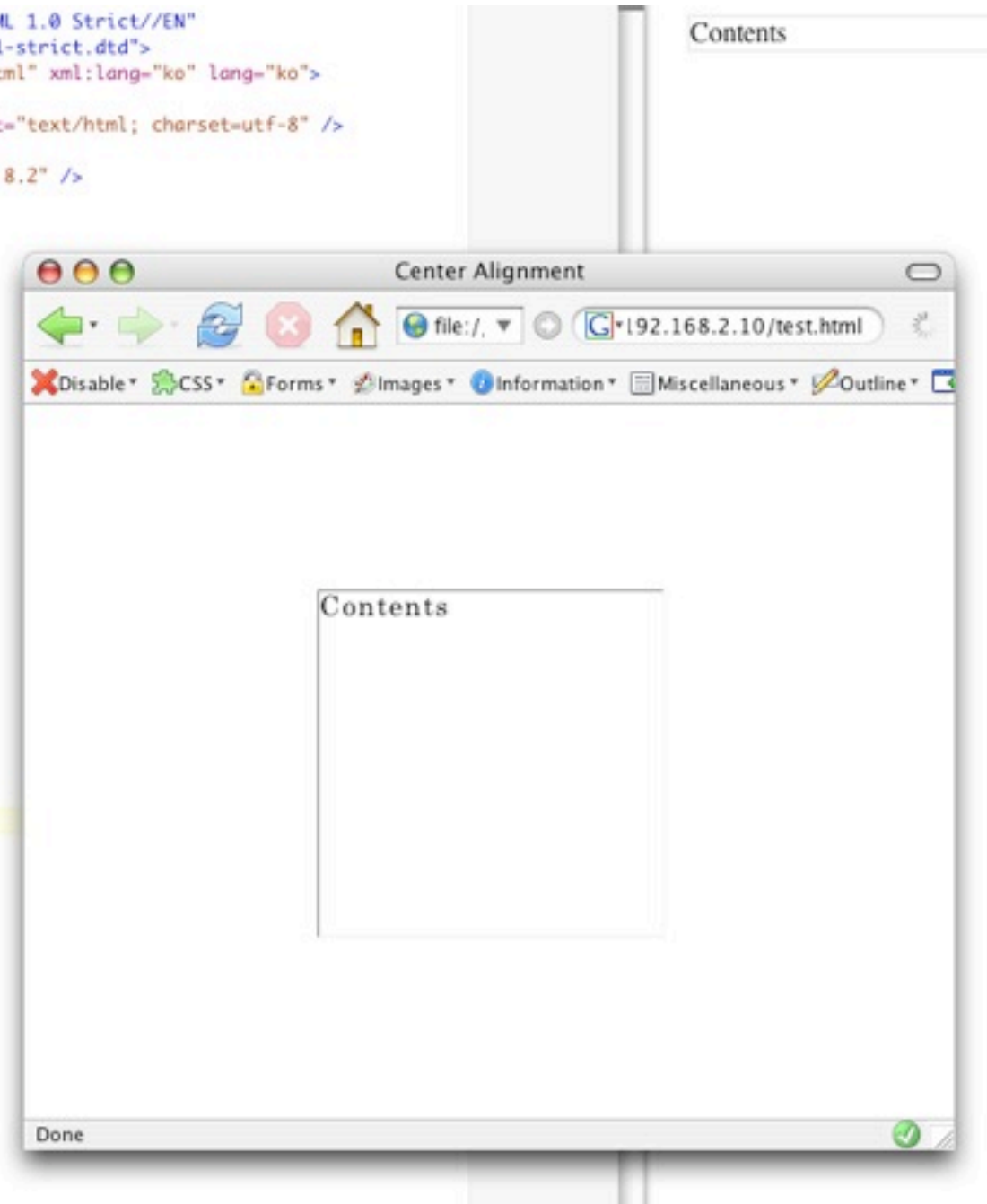
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ko" lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>Center Alignment</title>
<meta name="generator" content="BBEdit 8.2" />
<style type="text/css">
#center {
    width: 200px;
    border: 2px solid #eee;
    margin: 0 auto;
}
</style>
</head>
<body>
<div id="center">
    Contents
</div>
</body>
</html>
```



Block alignment

Vertical center alignment

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ko" lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>Center Alignment</title>
<meta name="generator" content="BBEdit 8.2" />
<style type="text/css">
html,
body {
    height: 100%;
    margin: 0;
    padding: 0;
}
#center {
    width: 200px;
    height: 200px;
    position: absolute;
    top: 50%;
    left: 50%;
    margin: -100px 0 0 -100px;
    border: 1px inset #ddd;
}
</style>
</head>
<body>
<div id="center">
    Contents
</div>
</body>
</html>
```



2-column layout

PR Site layout

DEMO

3-column layout

Portal style layout

DEMO

Design Components

List

Menu

Content Tab

Table

Board

Form

Plug-ing

List

UL, OL, DL

◎ Unordered List

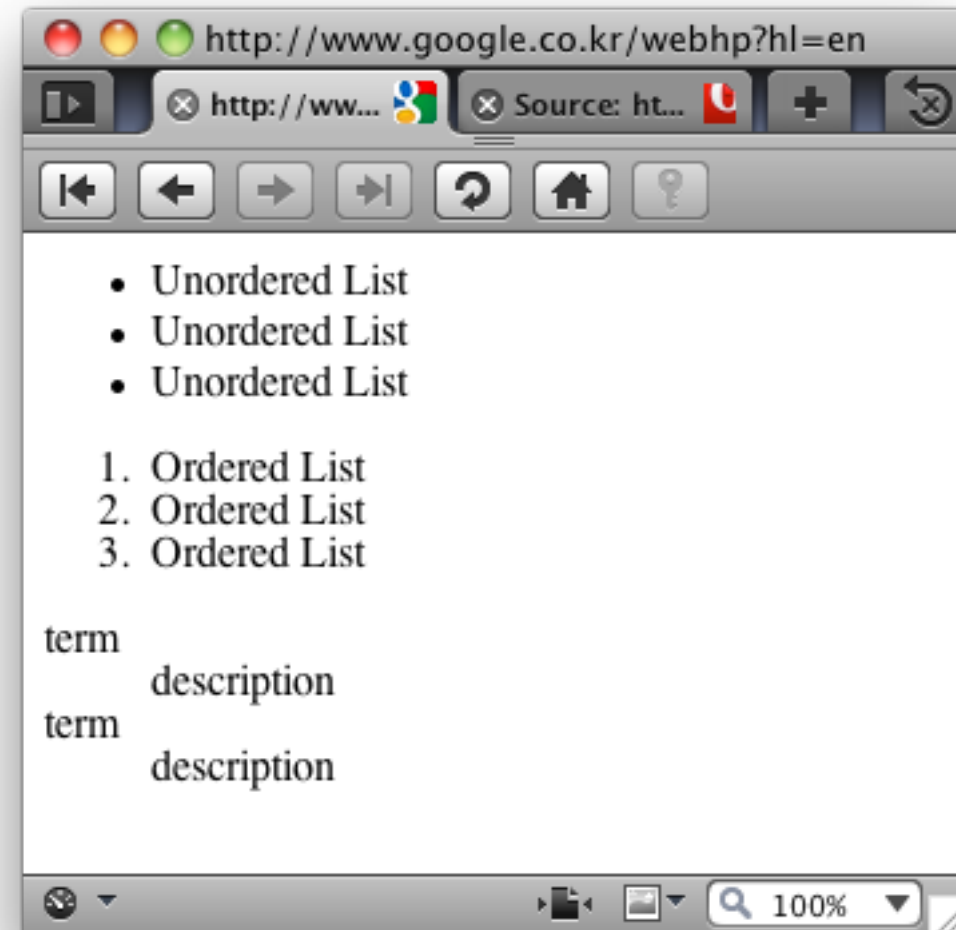
- 순서가 없는 목록

◎ Ordered List

- 순서가 있는 목록

◎ Definition List

- 정의 목록
- 용어(term) - 설명(definition)과 같은 관계를 나타낼 때 사용



List

잘못 사용된 Lists

- ◎ 서로 다른 종류의 항목을 같은 수준의 ``로 묶지 않아야 합니다.
 - `<td>`를 ``로 바꾼 것 같은 구현
 - 페이지 전체를 하나의 ``로 묶은 경우(특별한 경우 제외)
 - 게시판 리스트를 ``이나 ``로 표현한 경우
 - 테이블이 더 적절하고 의미에 맞음

List

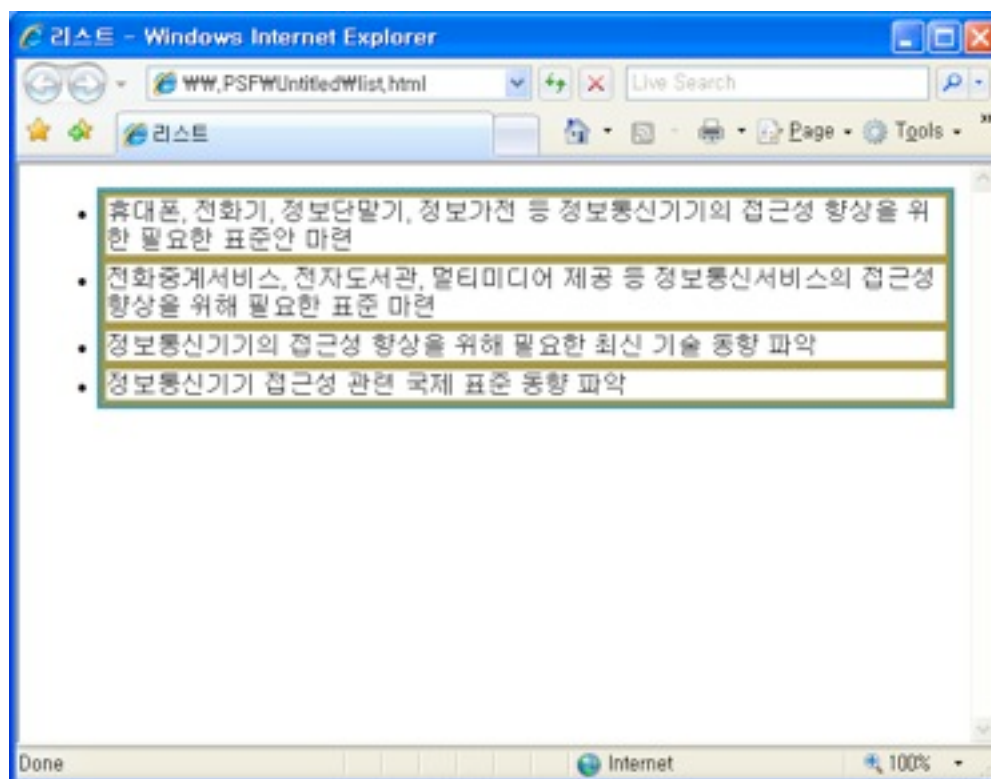
잘못 사용된 Lists

- ◎ <dt>와 <dd>의 관계가 의미에 맞아야 합니다.
 - <dt>나 <dd>가 없거나 서로 연관이 없는 경우
 - <dd>가 <dt>보다 위에 나오는 경우
 - 서식을 <dl>로 표현한 경우
 - 레이블과 서식의 관계를 <dl>로 보는 것은 무리가 있음

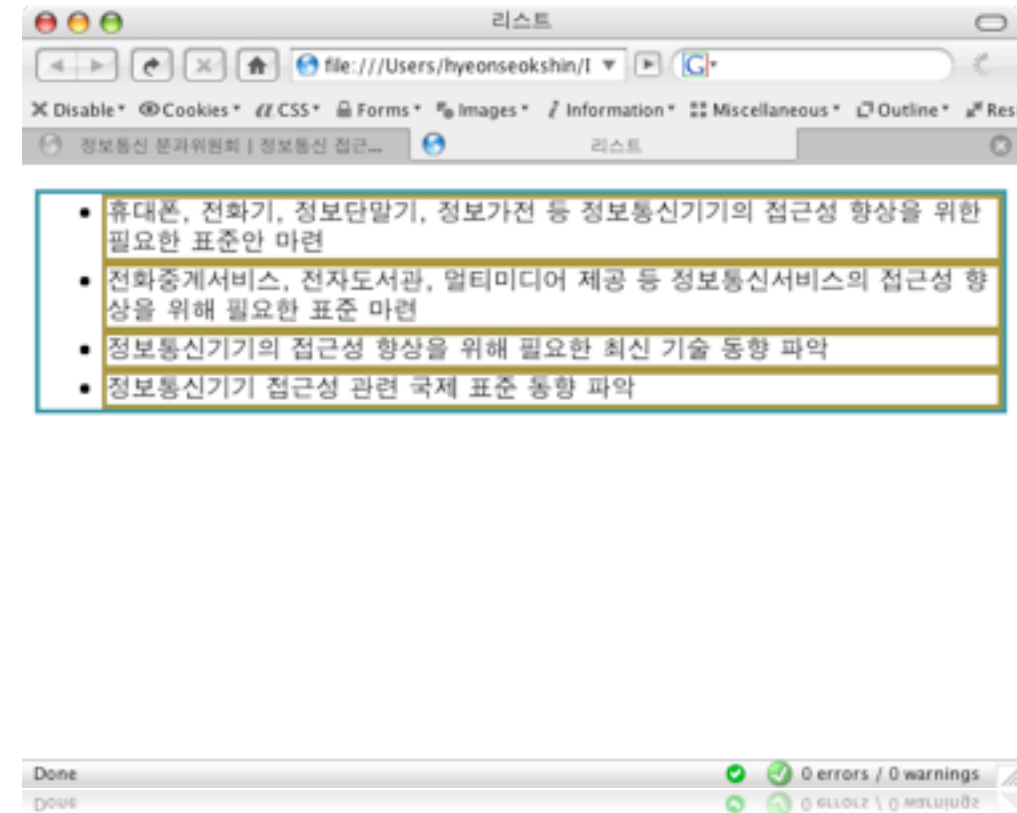
List

브라우저 호환성

- ◎ IE 계열은 불릿 영역을 margin으로 구현
 - margin, padding을 모두 리셋 해야 한다.



인터넷 익스플로러



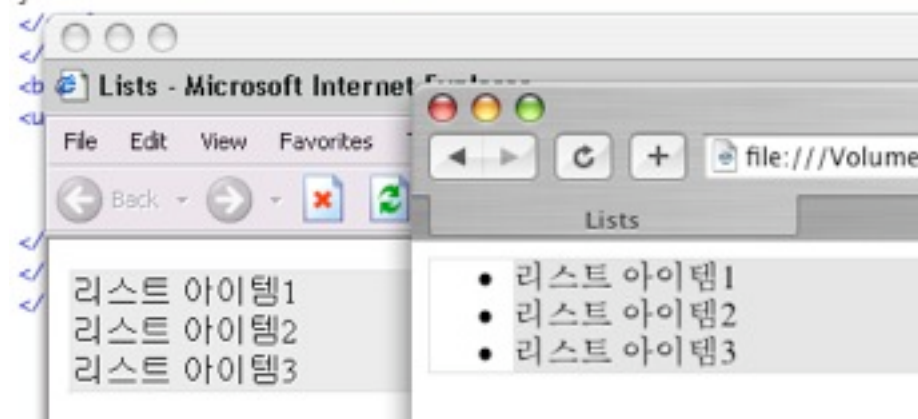
다른 브라우저

List

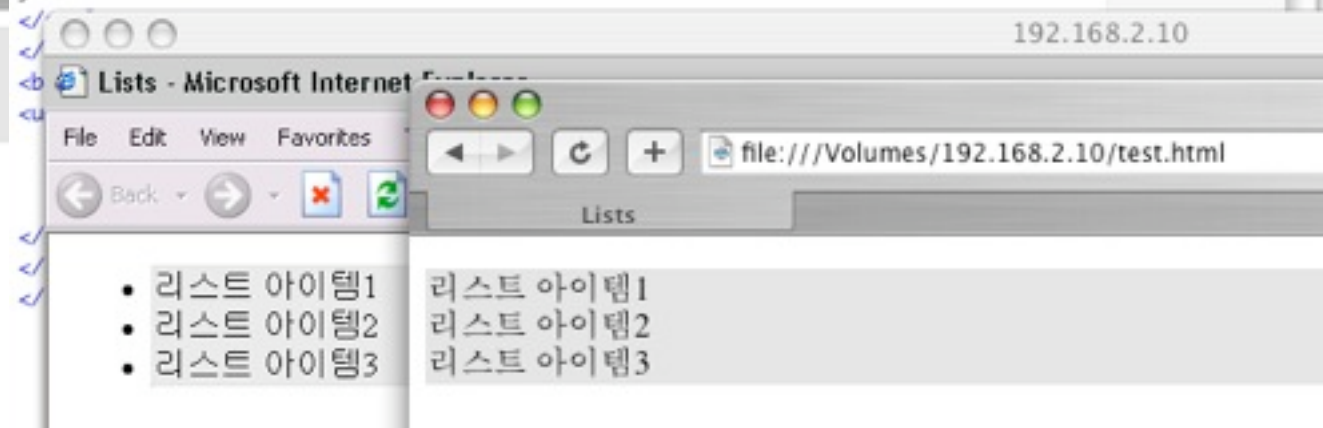
브라우저 호환성

◎ margin vs. padding

```
<style type="text/css">
ul {
    margin: 0;
    border: 1px solid #ddd;
}
ul li {
    background: #e0e0e0;
}
```



```
<style type="text/css">
ul {
    padding: 0;
    border: 1px solid #ddd;
}
ul li {
    background: #e0e0e0;
}
```



List

Background bullet

- ◎ 배경이미지를 활용하여 표현한다.

활동분야

- 휴대폰, 전화기, 정보단말기, 정보가전 등 정보통신기기의 접근성 향상을 위한 필요한 표준 마련
- 전화중계서비스, 전자도서관, 멀티미디어 제공 등 정보통신서비스의 접근성 향상을 위해 필요한 표준 마련
- 정보통신기기의 접근성 향상을 위해 필요한 최신 기술 동향 파악
- 정보통신기기 접근성 관련 국제 표준 동향 파악

- 외국을위하기의 활동위 마다 국제 표준을 파악

```
body.committee div.committee ul {  
    margin: 10px 0 0 27px;  
    padding: 0;  
    list-style: none;  
}  
body.committee div.committee ul li {  
    background: url(/images/committee/ListBullet.gif) no-repeat 0 0.5em;  
    padding-left: 9px;  
}
```

Menu

세로 형태 메뉴

- ◎ 링크 요소(<a>)에 display: block을 이용하여 메뉴 항목의 클릭 영역을 최대화 할 수 있음

자료실

동향자료

법/연구보고서

관련표준

정보통신접근성표
준화

회원사 제품 정보

회원사 제품 정보

동향

법/연구보고서

```
#submenu {
    margin: 8px 0 0 16px;
    padding: 0;
    list-style: none;
    width: 168px;
}
#submenu li {
    border-bottom: 1px solid #ddd;
}
#submenu li a {
    display: block;
    width: 140px;
    padding: 0.5em 10px 0.2em 18px;
    color: #333;
    text-decoration: none;
}
#submenu li a:hover,
#submenu li a:active,
#submenu li a.on {
    background: #F5F5F5;
    color: #069;
}
```

Menu

가로 형태 메뉴

동호회 분류

동호회 소식

아이템 샵

라운드 친구 찾기

동호회 가이드

전체 | CC | 친목 | 동창 | 직장 | 지역 | 연습장 | 기타

◎ float를 이용하여 요소를 가로로 배열

- 반드시 width를 지정해 주어야 함

```
body.club #topmenu {
    height: 72px;
    background: url(/images/club/nav/topmenu.gif) no-repeat 0 2px;
    padding: 0;
    overflow: hidden;
}
body.club #topmenu ul {
    margin: 0;
    padding: 0;
    list-style: none;
}
body.club #topmenu ul li {
    float: left;
    margin-right: -5px;
}
body.club #topmenu ul ul {
    position: absolute;
    top: 97px;
    left: 0;
    display: none;
}
```

```
body.club #topmenu ul ul li {
    margin: 0;
    padding: 8px 7px 9px 9px;
    background: url(/images/club/nav/topmenu_submenu.gif) no-repeat 0 10px;
}
body.club #topmenu ul ul li.first {
    background: none;
    padding-left: 0;
}
body.club #topmenu li.menu-1-1 ul {
    left: 20px;
    width: 40em;
}
/* 중략 */
body.club #topmenu ul ul li a:link,
body.club #topmenu ul ul li a:visited {
    color: #FFFEF3;
    font-weight: bold;
}
body.club #topmenu ul ul li a:hover,
body.club #topmenu ul ul li a:active {
    color: #fff;
    font-weight: bold;
}
```

Table

Row group tags

- ◎ <thead>, <tbody>, <tfoot>

Column group tags

- ◎ <colgroup>, <col>

Table Title

- ◎ <caption>

Data Cell

- ◎ <th>, <td>

Table

Row group

전쟁별 야스쿠니 신사 합사자 수					
번호	전쟁의 명칭	합사자수 1972.10)	합사자수 2000.10)	합사자수 2004.10)	백분율 (%)
1	명치유신	7,399	7,751	7,751	0.31%
2	서남역전후	7,292	6,971	6,971	0.28%
3	청일전쟁	13,619	13,619	13,619	0.55%
4	대만 및조선침략 사건	1,375	1,130	1,130	0.05%
5	북청사변	1,256	1,256	1,256	0.05%
6	러일전쟁	88,243	88,429	88,429	3.59%
7	대정3~9년전쟁	4,850	4,850	4,850	0.20%
8	제남사변	157	185	185	0.01%
9	만주사변	17,137	17,175	17,176	0.70%
10	지나사변	187,288	191,218	191,250	7.75%
11	태평양전쟁	2,115,569	2,133,760	2,133,915	86.52%
합		2,444,185	2,466,344	2,466,532	100.00%

11	태평양전쟁	2,115,569	2,133,760	2,133,915	86.52%
10	지나사변	187,288	191,218	191,250	7.75%
9	만주사변	17,137	17,175	17,176	0.70%
8	제남사변	157	185	185	0.01%

```

<table class="data">
  <caption>전쟁별 야스쿠니 신사 합사자 수</caption>
  <thead>
    <tr>
      <th>번호</th>
      <th>전쟁의 명칭</th>
      <th>합사자수<br>1972.10) </th>
      <th>합사자수<br>2000.10) </th>
      <th>합사자수<br>2004.10) </th>
      <th>백분율<br>(%)</th>
    </tr>
  </thead>
  <tfoot>
    <tr>
      <td colspan="2">합</td>
      <td>2,444,185</td>
      <td>2,466,344</td>
      <td>2,466,532</td>
      <td>100.00%</td>
    </tr>
  </tfoot>
  <tbody>
    <tr>
      <td>1</td>
      <td>명치유신</td>
      <td>7,399</td>
      <td>7,751</td>
      <td>7,751</td>
      <td>0.31%</td>
    </tr>
    <!-- 생략 -->
  </tbody>
</table>

```


Table

Column group

음식	밥	된장찌개	장조림	버섯 스프	치킨 샐러드	어니언 링	초밥	생선회	우동
열량	100	100	100	100	100	100	100	100	100
특유	100	100	100	100	100	100	100	100	100

```
table.data col.korean {  
  background: #fcc;  
}  
table.data col.western {  
  background: #cfc;  
}  
table.data col.japanese {  
  background: #ccf;  
}
```

```
<table class="data">  
  <caption>음식물 열량 표</caption>  
  <colgroup>  
    <col class="food" />  
  </colgroup>  
  <colgroup>  
    <col class="korean" />  
    <col class="korean" />  
    <col class="korean" />  
  </colgroup>  
  <colgroup>  
    <col class="western" />  
    <col class="western" />  
    <col class="western" />  
  </colgroup>  
  <colgroup>  
    <col class="japanese" />  
    <col class="japanese" />  
    <col class="japanese" />  
  </colgroup>  
  <thead>  
    <tr>  
      <th>음식</th>  
      <th>밥</th>  
      <th>된장찌개</th>  
      <th>장조림</th>  
      <th>버섯 스프</th>  
      <th>치킨 샐러드</th>  
      <th>어니언 링</th>  
      <th>초밥</th>  
      <th>생선회</th>  
      <th>우동</th>  
    </tr>  
  </thead>  
</table>
```

Table

cellpadding, cellspacing

- ◎ border-collapse, padding 속성을 이용해서 CSS로 대체

```
table {  
    border-collapse: collapse;  
}  
th, td {  
    padding: 0;  
}
```

Table

table-layout: fixed

- table width calculation algorithm을 거치지 않고 테이블 레이아웃을 고정하여 랜더링 속도를 향상시켜준다.
- 테이블의 첫번째 행의 너비를 기준으로 각 셀의 너비를 고정하여 디자인이 틀어지지 않게 한다.
 - 셀 안 내용의 크기와 상관 없이 내용이 셀 너비보다 클 경우에도 셀이 좌우로 늘어나지 않게 해 준다.
- 첫번째 행에 colspan을 사용했을 경우 <col>을 이용하여 너비를 지정할 수 있다.
 - 첫번째 행에 셀이 없을 경우 균등 분할 된다.

Table

table-layout: fixed

```
<table border="1" style="table-layout: fixed; width: 100%;">
  <colgroup>
    <col width="23%">
    <col width="23%">
    <col width="23%">
    <col>
  </colgroup>
  <thead>
    <tr>
      <th colspan="3">heading1</th>
      <th rowspan="2">heading2</th>
    </tr>
    <tr>
      <th>sub-heading1</th>
      <th>sub-heading2</th>
      <th>sub-heading3</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>data1</td>
      <td>data2</td>
      <td>data3</td>
      <td>data4</td>
    </tr>
    <!-- 생략 -->
  </tbody>
</table>
```

Table

기본 테이블

- ◎ thead, th를 활용한 요소 선택
- ◎ table, th를 활용한 셀 경계선 그리기

Board

Table? vs. List?

DEMO

Form

Grouping

- ◎ `<fieldset>`, `<legend>`

Label

- ◎ `<label>`, `<label for="">`

Submit

- ◎ `<input type="submit||image">`, `<button type="submit">`
- ◎ `<form action="">`

Form

Fieldset, legend

- ◎ 폼 컨트롤 들을 서로 묶어주는 역할
- ◎ 하나의 <fieldset>에는 하나의 <legend>가 필요
- ◎ 브라우저 간의 랜더링 차이로 사용에 주의 필요



```
<form action="/admin/login.php" method="post">
  <fieldset>
    <legend>로그인</legend>
    <label for="user-password">비밀번호</label>
    <input id="user-password" type="password" name="password" />
    <input type="submit" value="로그인" />
  </fieldset>
</form>
```


Form

Label

- ◎ 레이블과 서식을 연결
- ◎ 명시적 연결

```
<label for="user-id">아이디</label>  
<input type="text" id="user-id" />
```

- ◎ 암시적 연결

```
<label>  
  아이디  
  <input type="text" />  
</label>
```

- ◎ 브라우저에서는 레이블 클릭시 연결된 서식으로 포커스를 이동시켜줌

Form

Label

Post a comment

Name :

신현석

신현석

않습니다. Gravata를 표시 합니다.)

hyeonseok@cidow.com

URL :

http://hyeonseok.com

http://hyeonseok.com

URL :

http://hyeonseok.com

```
<div id="write-comment">
  <h3>Post a comment</h3>
  <label for="blog-comment-rejname">Name</label> : <br />
  <input type="text" id="blog-comment-rejname" name="rejname" value="" /><br />
  <label for="blog-comment-email">Email Adress</label> : (공개 되지 않습니다. <a href="http://
www.gravatar.com/">Gravata</a>를 표시 합니다.)<br />
  <input type="text" id="blog-comment-email" name="email" value="" /><br />
  <label for="blog-comment-url">URL</label> : <br />
  <input type="text" id="blog-comment-url" name="url" value="http://" /><br />
  <label for="blog-comment-body">Comments</label> : (html은 지원되지 않습니다)<br />
  <textarea id="blog-comment-body" name="body" rows="5" cols="40"></textarea><br />
  <input type="hidden" name="blogid" value="285" />
  <input type="submit" value="Post" />
</div>
```

Form

Label

이용약관

동북아역사재단(이하 "재단")은 동북아역사재단 홈페이지
(<http://www.historyfoundation.or.kr>, 이하 "재단 홈페이지")를 통해 인터넷 관련 서비스를
제공하고 이용자가 이 서비스를 이용하는 데 필요한 조건과 절차 및 기타 필요한 사항을 아래와 같
이 규정합니다.

- 제1조 정 의

- ① 재단 홈페이지는 재단이 사업정보를 제공하기 위하여 운영하는 인터넷 서비스입니다.
- ② "이용자"란 재단 홈페이지에 접속하여 이 약관에 따라 재단 홈페이지를 이용하는 개인을 말함

☐ 회원약관에 동의합니다.

```
<P><input type="checkbox" id="chk1"> <label for="chk1"><strong>회원약관</strong>에 동의합니다.</label></p>
```

Form

Label

◎ 불필요한 구분으로 레이블을 사용하기 힘든 서식

- 주소

주소★ -

- 주민등록번호

이름 주민등록번호 -

- 이메일

주로 사용하는 Email★ @
비밀번호 분실 시에 비밀번호를 찾으시려면, 네이트메일 이외에 자주 사용하시는 메일

- 전화번호

휴대폰번호 - -

Plug-ins

Object element

- ◎ 외부 콘텐츠를 웹페이지에 포함
 - 플러그인 콘텐츠 외에 text, html, image 등도 사용할 수 있음
- ◎ 표준 구문
 - data : 포함할 파일의 경로, type : 포함할 파일의 형식

```
<!-- 플래시 무비 -->
<object data="mymovie.swf" type="application/x-shockwave-
flash"></object>

<!-- png 이미지 -->
<object data="mypicture.png" type="image/png"></object>

<!-- html 파일 : iframe과 유사 -->
<object data="content.html" type="text/html"></object>

<!-- txt 파일 -->
<object data="log.txt" type="text/plain"></object>
```

Plug-ins

Object element

- ◎ 대체 콘텐츠(fallback contents)
 - 플러그인을 사용할 수 없는 환경을 대비해서 제공하는 콘텐츠
 - object의 하위 콘텐츠를 대체 콘텐츠로 인식

```
<!-- 플래시 플러그인 -->
<object data="hello.swf" type="application/x-shockwave-flash">
  <!-- PNG 이미지 -->
  <object data="hello.png" type="images/png">
    <!-- GIF 이미지 -->
    <object data="hello.gif" type="images/gif">
      <!-- 일반 text -->
      <p>Hello!</p>
    </object>
  </object>
</object>
```

Plug-ins

Object element

◎ 플래시 삽입

- 어도비 방식

- 크로스 브라우징을 위해서 폐기된 embed를 사용

```
<object classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/
flash/swflash.cab#version=7,0,0,0" width="550" height="400"
id="flash_movie" align="middle">
    <param name="allowScriptAccess" value="sameDomain">
    <param name="movie" value="flash_movie.swf">
    <param name="quality" value="high">
    <param name="bgcolor" value="#ffffff">
    <embed src="flash_movie.swf" quality="high"
bgcolor="#ffffff" width="550" height="400" name="Untitled-1"
align="middle" allowScriptAccess="sameDomain" type="application/
x-shockwave-flash" pluginspage="http://www.macromedia.com/go/
getflashplayer"></embed>
</object>
```

Plug-ins

Object element

◎ W3C 방식

- IE에서 플래시 무비를 인식하지 못하기 때문에 param을 사용하여 플래시 무비 경로를 제공해 주어야 함

```
<object type="application/x-shockwave-flash" data="images/  
banner.swf" width="140" height="40">  
    <param name="movie" value="images/banner.swf" />  
</object>
```

- IE에서 스트리밍 기능이 작동하지 않음
 - Flash Satay: 아주 작은 크기의 컨테이너 무비를 만들고 이 무비가 본래의 콘텐츠 로드(loadmovie)로 불러들이는 방법
 - 랜더링 오류(IE가 개체를 판별할 때 까지 화면에 박스가 나옴)는 여전히 발생

Plug-ins

Object element

◎ Hixie method

- IE의 conditional comment를 이용하여 IE에서의 object의 랜더링 오류를 보완

```
<object classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/
flash/swflash.cab#version=7,0,0,0" width="550" height="400"
id="flash_movie" align="middle">
    <param name="movie" value="flash_movie.swf">
    <!--[if !IE]> <-->
        <object type="application/x-shockwave-flash"
data="flash_movie.swf" width="550" height="400">
            <p></p>
        </object>
    <!--> <![endif]-->
</object>
```

- IE8은 표준 방법을 사용하기 때문에 이 방법을 사용하지 않아도 됨

Plug-ins

IE Eolas patch

- ◎ eolas가 IE에서 외부 객체를 웹 페이지에 포함 시킬 때 자동으로 활성화되는 방식이 자사의 특허를 침해 했다고 주장
 - 클릭을 한번 해야 활성화 되도록 IE를 수정
 - 이를 방지하기 위해 MS는 외부 자바스크립트로 넣는 방법 권장
- ◎ 2007년 12월경 MS가 라이선싱을 하면서 분쟁 해결
 - 더이상 사용할 필요가 없음
 - 브라우저 보안 업데이트를 권장
 - 플래시를 넣을 때 외부 스크립트로 제작하지 않아도 됨

Plug-ins

IE Eolas patch

◎ IE_HtmlRewrite.js: IE Eolas patch 대응 스크립트

- 대체 콘텐츠를 따로 제공하지 않아도 되어 접근성이 높음

```
function IE_HtmlRewrite(objParent) {  
    if (window.ActiveXObject && objParent) {  
        objParent.innerHTML = objParent.innerHTML;  
    }  
}
```

```
<div id="flash-movie">  
    <object type="application/x-shockwave-flash"  
data="movie.swf">  
    </object>  
</div>  
<script type="text/javascript">IE_HtmlRewrite  
(document.getElementById("flash-movie"))</script>
```

- flashvar param을 인식 못하기 때문에 get으로 적용해야 함

```
<object type="application/x-shockwave-flash" data="movie.swf?  
myvar=myvalue">  
</object>
```

Extensions

IE6

HTML5 & CSS3

DOM Scripting

IE6

IE6 ban

- ◎ 개발자들 사이에 IE6의 버그로 인해 작업 능률을 떨어뜨린다는 의견이 많다.
 - Save the developer
 - No more IE6
- ◎ 사용자들에게 브라우저 업그레이드를 권장하는 것은 바람직하지만, 개발자들이 사이트 만들기 힘들다고 브라우저를 차별하는 행위는 웹표준 이념에 배치된다.

IE6

Market share (by Internet Trend)

웹브라우저버전 - 2009/08/25 ~ 2009.09.22



인터넷트렌드

<http://internettrend.co.kr>

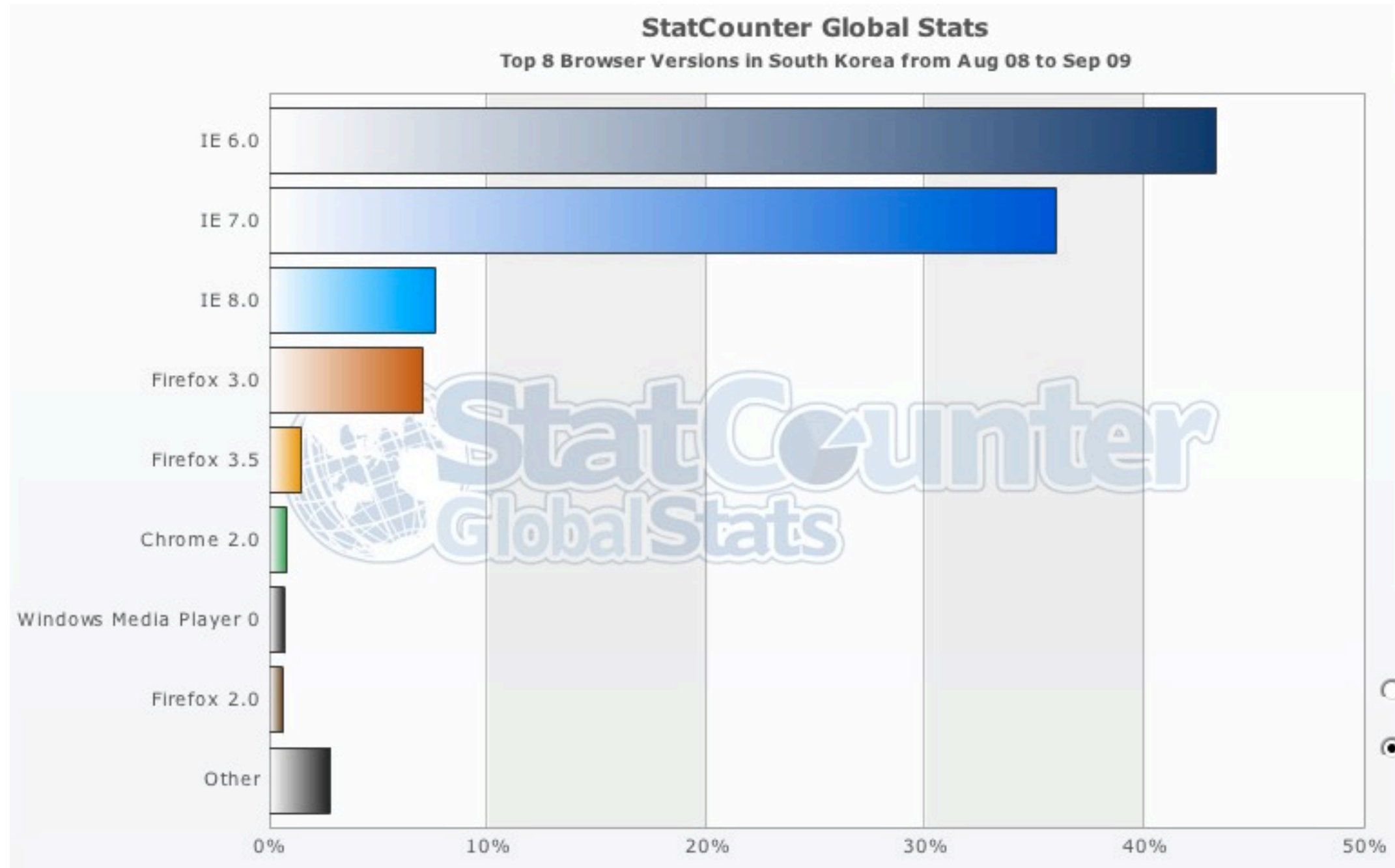


Microsoft Internet Explorer 6.0 (55.06%)
Microsoft Internet Explorer 7.0 (31.57%)
Microsoft Internet Explorer 8.0 (11.86%)
Firefox 3.5 (0.52%)
Firefox 3.0 (0.35%)
Chrome 2.0 (0.18%)
사파리 531 (0.12%)

순위	구분	기간내평균	기간시작(2009-08-25)	기간끝(2009-09-22)	기간내변화율
1	Microsoft Internet Explorer 6.0	55.06 %	56.05 %	54.57 %	1.48 % ▼
2	Microsoft Internet Explorer 7.0	31.57 %	31.78 %	31.25 %	0.53 % ▼
3	Microsoft Internet Explorer 8.0	11.86 %	10.67 %	12.65 %	1.98 % ▲
4	Firefox 3.5	0.52 %	0.49 %	0.56 %	0.07 % ▲
5	Firefox 3.0	0.35 %	0.38 %	0.32 %	0.06 % ▼
6	Chrome 2.0	0.18 %	0.19 %	0.05 %	0.14 % ▼
7	Safari 531	0.12 %	0.08 %	0.15 %	0.07 % ▲

IE6

Market share (by Stat Counter)



HTML5 & CSS3

CSS3 DEMO

- ◎ 11 Classic CSS Techniques Made Simple with CSS3 (<http://net.tutsplus.com/tutorials/html-css-techniques/11-classic-css-techniques-made-simple-with-css3/>)

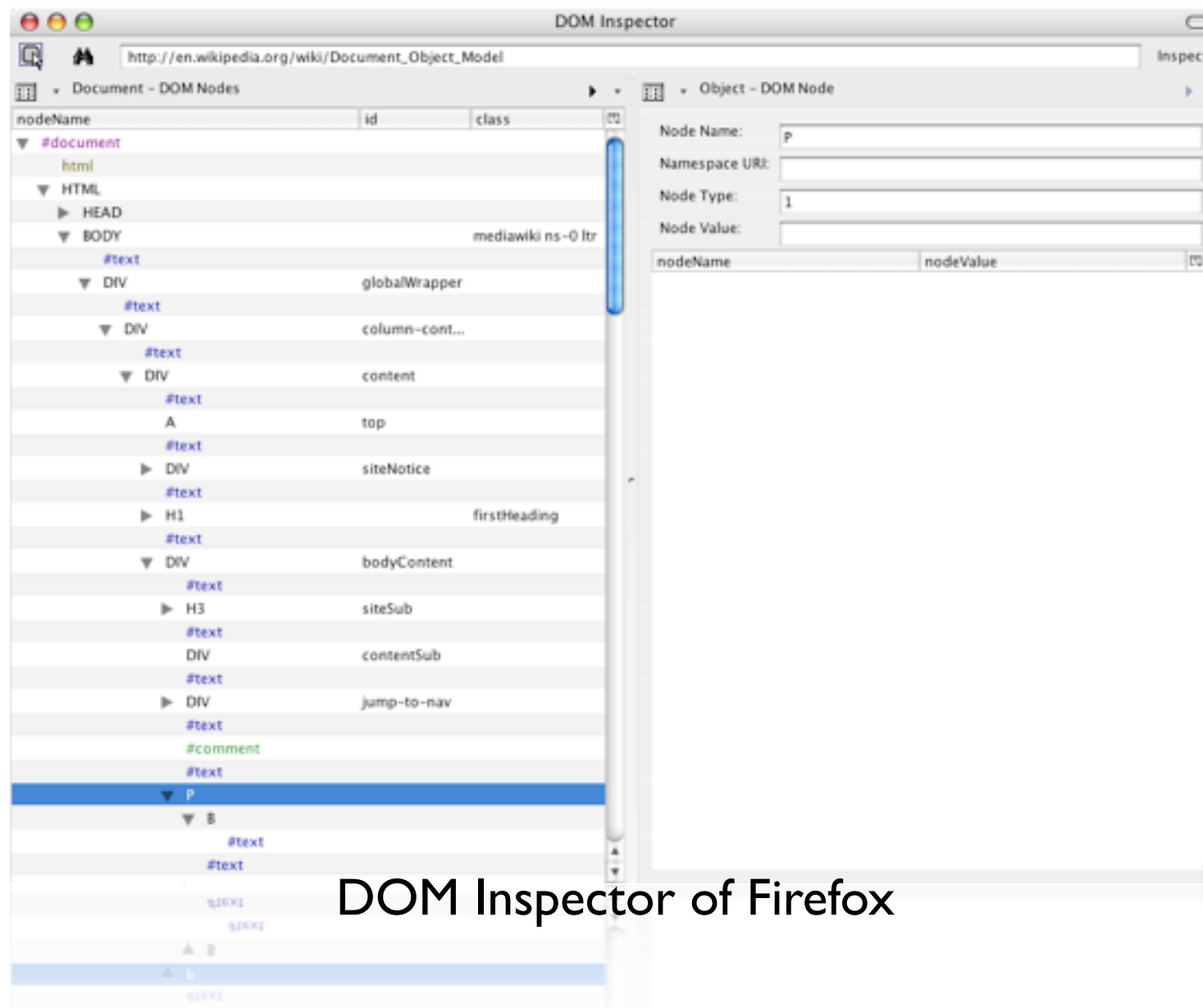
HTML5 DEMO

- ◎ HTML 5 Forms Demo (<http://brucelawson.co.uk/tests/html5-forms-demo.html>)
- ◎ Very Impressive! HTML5 demo using Canvas and Audio. (<http://9elements.com/io/projects/html5/canvas/>)

Document Object Model

Wikipedia definition

- Document Object Model (DOM) is a description of how an HTML or XML document is represented in a tree structure.



DOM Inspector of Firefox

Document Object Model

Compatibility

- ◎ IE4, NN4가 주 사용 브라우저였던 2000년 까지만 해도 자바스크립트 기능이 상호 운용성(호환성)을 갖기 위해서는 브라우저 별로 스크립트를 작성해야만 했다.
- ◎ 2005년 현재, IE6, Gecko 기반 브라우저(Firefox, Camino 등), Opera, Safari, Konquerer 등의 브라우저에서는 DOM을 충분히 지원하고 있어서 단일 코드로 상호 운용성을 확보할 수 있다.
- ◎ 스크립트 없는 환경에 대한 대비를 하여 기능을 구현 하는 것도 중요하다.

DOM tutorial

My First DOM Script

DEMO

DOM Scripting Accessibility

href : hypertext reference

◎ href 속성의 값으로 유효한 URL을 사용한다.

- Bad Case

```
<a href="#" onclick="myfunction()">
```

```
<a href="javascript:myfunction()">
```

- Good Case

```
<a href="#notice-list" onclick="return myfunction();">
```

DOM Scripting Accessibility

window.open

- 팝업창의 경로를 href에 명시한다.

- Bad Case

```

```

```
<a href="#"></a>
```

- Good Case

```
<a href="popup.html" onclick="window.open(this.href, 'popupName',  
'width=300,height=200'); return false;"></a>
```

DOM Scripting Accessibility

redirection

- 페이지 이동은 스크립트에 독립적으로 제공한다.

- Bad Case

```
<script type="text/  
javascript">document.location.href="redirection.html";</script>
```

```
<form method="post" name="sg_form" action="http://www.qubi.com/"  
target="_top"></form>  
<script> sg_form.submit(); //3</script>
```

- Good Case

```
<?php  
header("Location: end_process.html");  
?>
```

DOM Scripting Accessibility

form submit

- ◎ 자바스크립트로 서식을 전송하지 않는다.
 - Bad Case

```
<script type="text/javascript">
function submitForm() {
    loginForm.submit() ;
}
</script>
<form id="loginForm" name="loginForm" action="">
    User Id <input type="text" name="loginId">
    User Password <input type="password" name="loginPassword"><br>
    
</form>
```

DOM Scripting Accessibility

form submit

- ◎ HTML의 전송기능을 사용한다.
 - Good Case

```
<form id="loginForm" name="loginForm" action="">
  <p>
    <label for="loginId">User Id</label>
    <input type="text" id="loginId" name="loginId" /><br /
  >
    <label for="loginPassword">User Password</label>
    <input type="password" id="loginPassword"
name="loginPassword" />
  </p>
  <p>
    <input type="image" src="login.gif" alt="Login" />
  </p>
</form>
```


DOM Scripting Accessibility

form validation

- ◎ onsubmit 이벤트 핸들러를 사용한다.



DOM Scripting Accessibility

form validation

- ◎ HTML의 전송기능을 사용한다.
 - Good Case

```
<script type="text/javascript">
function submitForm(formEl) {
    //TrimAll(formEl);

    var errorMessage = null;
    var objFocus = null;

    if (formEl.loginId.value.length == 0) {
        errorMessage = "아이디를 넣어주세요.";
        objFocus = formEl.loginId;
    } else if (formEl.loginPassword.value.length == 0) {
        errorMessage = "비밀번호를 넣어주세요.";
        objFocus = formEl.loginPassword;
    }

    if(errorMessage !=null) {
        alert(errorMessage);
        objFocus.focus();
        return false;
    }
    return true;
}
</script>

<form id="loginForm" name="loginForm" action="" onsubmit="return submitForm(this)">
    <label for="loginId">아이디</label> <input type="text" id="loginId" name="loginId" />
    <label for="loginPassword">비밀번호</label> <input type="password" id="loginPassword"
name="loginPassword" /><br />
    <input type="image" src="login.gif" alt="Login" />
</form>
```

Additional informations

Useful sites

- ◎ CSS Design Korea: <http://forum.standardmag.org/>
- ◎ 웹 접근성 연구소: <http://www.wah.or.kr/>
- ◎ 오페라 웹표준 강좌 한글화 문서: http://www.clearboth.org/wiki/doku.php?id=clearboth:document:web_standards_curriculum
- ◎ UI 개발자를 위한 북마크: http://naradesign.net/wiki/UI_개발자를_위한_북마크
- ◎ Archive on hyeonseok.com: <http://hyeonseok.com/pmwiki/>