

CSS (Cascading Style Sheet)

ONE STEP AHEAD

1. CSS 개요
2. CSS Selector, Cascading, Font
3. CSS Table, Positioning
4. CSS Pseudo Class
5. CSS Print
6. CSS Layout

CSS

CSS (Cascading Style Sheet) 개요

□ CSS(Cascading Style Sheet)

- 구조적으로 짜여진 문서(HTML,XML)에 Style(글자,여백,레이아웃)을 적용하기 위해 사용하는 언어(Language).
- 운영체제나 프로그램에 관계없이 누구나 같은 형식으로 보기 위해 만들어진 W3C(World Wide Web Consortium)에서 공인한 문서형식의 표준
- 기존 웹문서를 다양하게 설계하고 수시 변경에 따른 한계점을 보완하기 위함.
 - 기존 HTML의 element의 크기(1~7)을 CSS를 이용 제한없이 사용가능.

□ HTML과의 연계적 비교

- HTML : 웹의 기본요소(HTML element)인 태그(Tag)의 모음.
- CSS : 속성(Property)의 모음
 - 기본요소(Element)의 특성(Property 또는 Attribute)이 무엇인지, 이용(Value또는 Keyword)은 어떻게 하며, 이용방법(Grammer:문법)은 무엇인지에 대한 분야.

CSS (Cascading Style Sheet) 장점

□ 기능확장성 : HTML의 기능을 확장

- HTML 대부분의 요소(element)들의 단순한 기능에 다양한 기능 추가 가능
- HTML 요소들의 기능 변경이 가능하여 독특한 문서형식을 설계가능

□ 양식의 모듈화 : 흐름이 같은 문서양식으로 전체를 구성할수 있음

- HTML는 세부항목까지 일일이 지정하나 CSS로 정의된 문서를 한번만 불러 사용가능.
- 구조화된 페이지 설계 가능
- 코드 간결성에 의한 간단한 유지보수 가능

□ 간편성 : 다양한 형식의 문서를 구성

- HTML의 <TABLE>로 구성된 레이아웃을 간단하게 조작,구성.
- 문서의 일부를 원하는 곳에 자유롭게 배치가 가능
- 효과적인 레이아웃(Layout)에 의한 풍부한 디자인 설계가 가능
 - 참고 사이트 : <http://www.csszengarden.com>

□ 일관성 : 사용환경(다른 브라우저)의 영향을 받지 않음

□ 웹 접근성 향상 [참고 사이트 : [01.01 웹접근성예제파일.html](#)]

- 각기 다른 MediaType설정 가능

CSS (Cascading Style Sheet) 단점

□ 웹브라우저(Web Browser)마다 다른 방식으로 지원

- 사용자환경(User Agent : 특히, 웹브라우저)에 따라 적용이 되지 않은 속성(property)이 있음. (Microsoft IE의 경우 가장 표준을 지키지 않음)

CSS 표기법

□ 사용방법

- `<H3><I>녹색 이탤릭</I></H3>`
- `H3 {font-style:italic; color:green}`
정의된 페이지내의 모든 H3 요소(element)에 같은 속성을 적용



- HTML의 속성(attribute)은 정해진 요소(element)에서만 사용가능하나, 스타일시트의 속성은 거의 모든 요소에 붙일수 있음.

CSS 표기법

□ 문법(Syntax)와 데이터 형식(DataType)

- 속성범위(Property Block)
Inline Style을 제외한 모든 Style Property와 Value,Keyword는 중괄호 {} 안에 들어감.

```
{CSS Property : value | keyword }
```

- 1개의 Property를 사용할 경우
CSS Property와 사용값 사이를 콜론(:)으로 구분

```
{CSS Property : (사용값) value | keyword }
```

- 여러 개의 속성을 사용할 경우

```
{ CSS Property : (사용값) value | keyword ; CSS Property : (사용값) value | keyword ; ..... }
```

CSS 표기법

□ 속성(Property)값에 사용되는 데이터의 종류 및 형태 (Data Type)표기법

- 사용단위 (Length Units)표기법
 - HTML에서는 붙이지 않으나 CSS에서는 비율(%),Pixel(px)와 같은 단위사용이 원칙
 - Strict 형식(Doctype)의 문서에서는 반드시 단위사용

```
{width : 10px} /* 올바른 표기 */  
{width : "10px"} /* 틀린 표기 */  
{width : '10px'} /* 틀린 표기 */
```

사용하지 않음 (사용시 문자열로 인식)

상대 길이 단위(Relative length units)	
em	The height of the element's font. (지정되거나 상속 받은 font 크기의 상대 크기) 예) {font:12px/1.5em;}이면 line-height = 12px × 1.5 = 18px
ex	The height of the letter "x". (소문자 "x"의 높이에 대한 상대 크기)
px	Pixels. (화면 해상도에 대한 상대 크기) 예) 해상도(resolution) 800 × 600 에서 1px 은 화면을 수평으로 800등분, 수직으로 600등분으로 나눈 화소(pixel) 1개의 단위.
%	Percentage. (자기가 속해 있는 상위 Tag의 크기 또는 상속받은 값에 대한 100분을 상대 단위)
절대 길이 단위(Absolute length units)	
in	Inches (1 inch = 2.54 centimeters).
cm	Centimeters.
mm	Millimeters.
pt	Points (1 point = 1/72 inches : 1인치를 72등분한 단위).
pc	Picas (1 pica = 12 points).

CSS 표기법

□ 속성(Property)값에 사용되는 데이터의 종류 및 형태 (Data Type) 표기법

- Keyword : 일종의 브라우저 예약어로 숫자와 단위 대신 사용하는 관용적인 단어
 - Thin, medium, thick, red, blue 등
 - 따옴표(") 사용하지 않음 (사용시 문자열로 인식)

```
{border-width : thin} /* 올바른 표기 */  
{border-width : "thin"} /* 틀린 표기 */  
{border-width : 'thin'} /* 틀린 표기 */
```

- 문자열(String)표기

```
body {background-image : url(http://www.webdesigner.co.kr/bgpicture.gif)}  
li {list-style-image : url("http://www.cadvance.org/icon/simple.gif")}
```

- 주석(Comment)표기

```
.sample{  
color:red;  
background-color:#c0c0c0;  
width:100%;  
border:1px solid #808080;  
padding:3px 4px; /* 상하 3px, 우좌 4px 의 테두리와의 미격을 준다. */  
}
```

CSS 특징 및 기능

□ 유효하지 못한(Invalid) 속성(Property)나 값의 처리(Parsing)

- 무시(Ignore)하거나 오류(Error)를 발생.
- CSS규격에 없는 속성을 사용한 경우

```
{woodplate : 300px} /* woodplate 라는 Property는 CSS에 없으므로 무시됩니다. */
```

- 유효하지 않은 사용값 입력한 경우

```
{color : redish} /* redish 는 named color 가 아니므로 무시합니다. */  
{color : #1234} /* #1234 는 Hex 3digit 이나 6digit 이 아니므로 무시합니다. */
```

- 입력값이 범위를 벗어난 경우

```
{color : rgb(-20, 128, 128)} /* {color : rgb(0, 128, 128)} 과 같습니다. */  
{color : rgb(300, 128, 128)} /* {color : rgb(255, 128, 128)} 과 같습니다. */
```

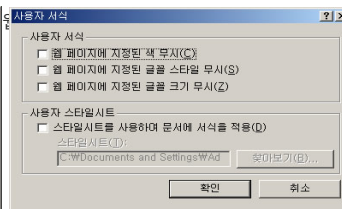
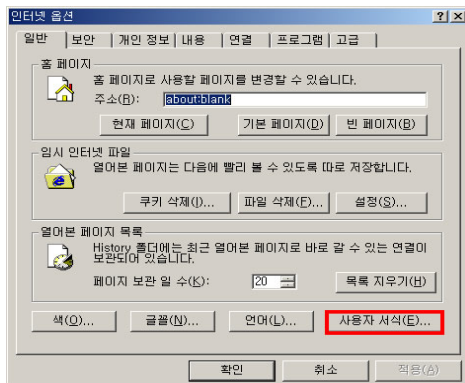
□ 대소문자의 처리

- 대소문자 구분이 없으나 Javascript나 DHTML scripting 할때 정확하게 구분함 (className, Javascript Property)

CSS 특징 및 기능

□ 계층화(Cascading)

- 제작자(Publisher),사용자(User),사용도구(User agent)에 따라 스타일시트의 중요도 변경
 1. 제작자 : 문서원문(Source) 스타일 지정 (@import)
 2. 사용자 : 사용자 나름대로 스타일시트를 포함하고 있는 문서를 지정할 수 있음.
 3. 사용도구 : 웹브라우저의 환경셋팅에 의한 스타일지정
- 위의 세가지 원천에 따라 범위가 중복될 경우 계층순서(Cascading Order)에 따라 해석됨



02_cascading브라우저예제.css

```
1 A {text-decoration: none; color:#4169e1; }
2 A:hover {text-decoration: none; color:#ff9933;}
3 body {font:12px/1.5em Verdana; color:brown; background-color:#EFEFEF;}
4
```

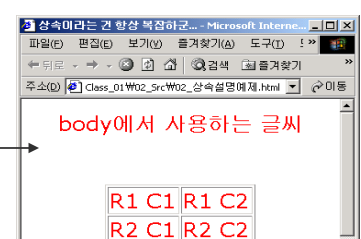
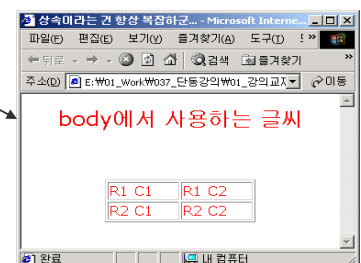
CSS 특징 및 기능

□ 속성값의 상속(Inherit)

- 하위 태그가 상위 태그의 사용값을 물려받음.(Cascading)
- CSS 속성값은 상속되지 않는값도 있음.
 - 예제 : Body의 속성값이 Table에 상속이 안됨.

```
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
2 <HTML>
3 <HEAD>
4 <TITLE>상속이라는 건 항상 복잡하군...</TITLE>
5 </HEAD>
6 <BODY style="font:24px Verdana; color:red;">
7 <CENTER>
8   body에서 사용하는 글씨</CENTER><BR><BR>
9   <TABLE align="center" border="1" width="50%">
10    <TR>
11      <TD>R1 C1</TD>
12      <TD>R1 C2</TD>
13    </TR>
14    <TR>
15      <TD>R2 C1</TD>
16      <TD>R2 C2</TD>
17    </TR>
18  </TABLE>
19 </BODY>
20 </HTML>
```

- 단 문서형식이 엄격모드(Strick)일때는 상속됨



CSS 특징 및 기능

□ 미디어(Media)타입의 표기

- 스타일시트의 가장 중요한 기능중 하나로서 하나의 문서를 모니터(screen), 인쇄(print), 음성합성장치(aural), 점자장치(braille)등 다른 미디어에 표현하는 방법을 지정할수 있음
- 각각의 속성에 따라 공통적으로 사용되는 속성과 그렇지 않은 속성이 있음
 - 예1) 'cue-before'속성은 소리(aural) 미디어타입에서만 사용됨.
 - 예2) 'font-size'속성은 모니터(screen)이나 인쇄(print)에서 동시에 사용됨.
- 표기방법

- @media 나 @import 사용

```
<style type="text/css" alt="currentStyle" media="all">@import "css/global.css";</style>
@import url("loudvoice.css") aural;
@media print { /* 인쇄를 위한 스타일시트가 이곳에 위치한다. */ }

@media print { BODY { font-size: 10pt } }
@media screen { BODY { font-size: 12pt } }
@media screen, print { BODY { line-height: 1.2 } }
```

- LINK사용

```
<LINK rel="stylesheet" type="text/css" media="print, handheld" href="foo.css">
```

CSS 특징 및 기능

□ 미디어(Media)타입의 표기

- CSS의 미디어타입은 일련의 CSS속성관련, 미디어타입에 맞는 속성과 값을 사용해야 함.
- 알려진 미디어 타입
 - all : 모든 장치에 사용하는 타입
 - sceen : 컴퓨터 모니터에서 사용하는 타입
 - print : 인쇄장치(Printer)를 위한 타입
 - aural : 음성장치(speech synthesizer)를 위한 타입
 - braille : 브레일(braille,점자)표현장치를 위한 타입
 - embossed : 페이지에 인쇄된 점자(braille)을 위한 타입
 - handheld : 전형적인 작은스크린, 단색상, 한정된 대역대(bandwidth)의 손에 들고다니는(handheld)장치를 위한 타입
 - projection : 프로젝터를 위한 타입
 - tty : 디스플레이 능력이 한정된 텍렉스(teletype), 터미널, 고정피치(fixed-pitch)미디어 위한 타입
 - tv : 텔레비전(television-type)장치를 위한 의도
(* W3C CSS2권고안에는 all, screen,print,aural에 관한 속성만 표기되어 있으며 가장 많이 사용하는 screen,print에 대한 내용만 다룸)



CSS Selector

(DomQuery, CssQuery, JQuery)

스타일시트(StyleSheet) 사용법

□ 스타일 사용법 4가지

- Inline Style(Tag속에 직접 지정)
- Embedded style sheet 와 Style block (Style block속에 일괄지정)
- Linked style sheet(외부파일 Link로 여러파일에 일괄지정)
- Imported Style sheet

□ Inline style속성 이용하기

- HTML Tag속에 style속성을 사용하여 직접 지정

```
1 <HTML>
2 <HEAD>
3 <TITLE>Inline style sheet</TITLE>
4 </HEAD>
5 <BODY>
6 <P style="color:red;">Red text</P>
7 <P style="color:green;">Green text</P>
8 <P style="color:blue;">Blue text</P>
9 </BODY>
10 </HTML>
```

Red text
Green text
Blue text

스타일시트(Style Sheet) 사용방법

□ 문서에 내포된 스타일시트(Embedded style sheet)와 스타일블럭(Style block)

- <HEAD>~</HEAD>사이에 위치
- 스타일시트를 지원하지 않는 브라우저를 위한 주석문 필요

```
1 <HTML>
2 <HEAD>
3 <TITLE>Embedded style sheet 와 Style block</TITLE>
4
5 <STYLE type="text/css">
6 <!--
7 h1 { color:red; font-style:italic;}
8 .maintext { margin-left:5%;}
9 -->
10 </STYLE>
11
12 </HEAD>
13 <BODY>
14 <H1>Linked style sheet</H1>
15 <P class="maintext">이젠 모든게 내 맘대로!</P>
16 </BODY>
17 </HTML>
18
19
```

스타일블럭

스타일시트(Style Sheet) 사용방법

□ 연결된 외부스타일 시트 (Linked style sheet)

- HEAD부분에 LINK element를 이용하여 외부 css파일 연결
 - REL="stylesheet" : 연결하는 문서가 스타일시트문서 정의
 - 많은 문서작업을 할때 모든 스타일을 "01_CSS삽입03.css"에 모두 정의해주고 문서들에는 <LINK>element를 이용하여 넣어주면 되며, 해당 스타일시트파일을 연결한 모든 문서를 수정할 때는 "01_CSS삽입03.css"파일만 수정하면 됨.
- 장점 :
 - 여러 개의 HTML문서를 동일한 스타일 적용시 사용
 - 연결된(Linked) 스타일시트 하나만 수정하여 여러 개의 HTML문서양식을 동시에 수정가능

```
1 H3 { font-style: italic; color: green; }
2
```

```
1 <HTML>
2 <HEAD>
3 <LINK REL="stylesheet" TYPE="text/css" HREF="01_CSS삽입03.css">
4 </HEAD>
5 <BODY>
6 <H3>여기는 녹색이고 이탤릭체인 H3이다.</H3>
7 <P>
8 <H3>여기도 그렇다.</H3>
9 </BODY>
10 </HTML>
11
```

스타일시트(Style Sheet) 사용방법

□ 임포트된 외부스타일 시트 (Imported style sheet)

- 결과적으로 링크된 외부스타일시트(Linked style sheet)와 같음
- 스타일시트 파일안에서 다른 스타일 시트를 가져와 사용할 수 있음.
- 보여지는 미디어타입(Media Type)을 설정할 수 있음
- 사용법 :

```
@import "mystyle.css";  
@import url("mystyle.css");
```

```
1 <HTML>  
2 <HEAD>  
3 <style type="text/css" media="all">@import "01_CSS삽입03.css";</style>  
4 </HEAD>  
5 <BODY>  
6 <H3>여기는 녹색이고 미텔릭체인 H3이다.</H3>  
7 <P>  
8 <H3>여기도 그렇다.</H3>  
9 </BODY>  
10 </HTML>
```

```
1 @import url("layout.css"); /* Page Layout */  
2  
3 /* Elements */  
4 @import url("font.css"); /* Font */  
5 @import url("form.css"); /* Form */  
6 @import url("link.css"); /* Link */  
7  
8 /* Special Pages */  
9 @import url("main.css"); /* main page */  
10 @import url("sub.css"); /* sub page */  
11
```

스타일시트(Style Sheet) 사용방법에 따른 스타일 우선순위

□ 스타일 적용의 우선순위의 (Cascading Order)

- 스타일 사용방법에 따른 우선순위
 1. Inline Style (직접작성 스타일) : 최우선 적용
 2. Embedded style sheet (문서에 내포된 스타일)
 3. Linked style sheet (링크된 외부스타일)
 4. Imported Style sheet (임포트된 외부스타일)

```
1 <HTML>  
2 <HEAD>  
3 <TITLE>Cascading order</TITLE>  
4 <STYLE type="text/css">  
5 <!--  
6 H1 {color:red;}  
7 -->  
8 </STYLE>  
9 </HEAD>  
10 <BODY>  
11 <H1 style="color:green;">Linked style sheet</H1>  
12 <P>이젠 모든게 내 맘대로!</P>  
13 </BODY>  
14 </HTML>  
15
```

Linked style sheet

이젠 모든게 내 맘대로!

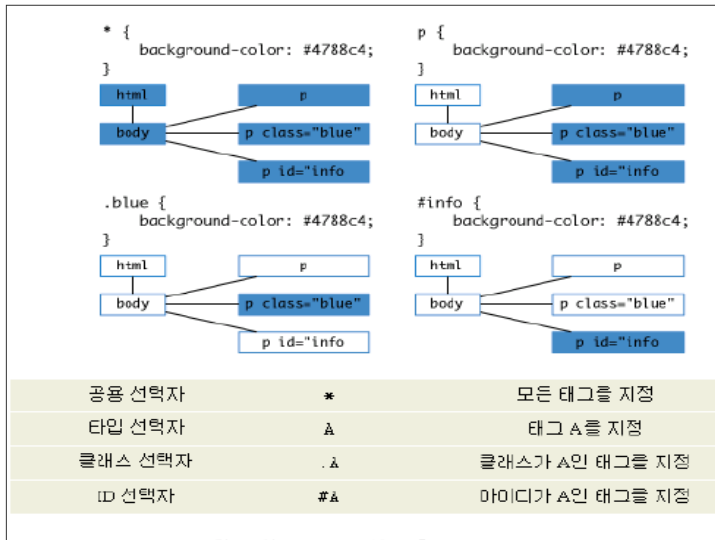
선택자(Selector)

[예제파일: 01_CSS삼일01.1]

□ 선택자 (Selector)

- CSS 속성(Property) 1개 또는 그 이상의 속성들을 속성블럭(Property block: { })속에 넣어 놓는것

□ 일반 선택자의 종류



우선순위 낮음

우선순위 높음

일반 선택자 (Selector) 요소

[예제파일: 02_01_선택자]

□ 공용선택자(Universal Selector)

- HTML문서내의 모든 요소(Element)에 같은 CSS속성을 지정.
- 사용법 : Asterisk(*)뒤에 스타일블럭(Styleblock) 사용

```
1 <STYLE type="text/css">
2 <!--
3 * {font-size : 12px;}
4 -->
5 </STYLE>
```

□ 타입선택자(Type Selector)

- HTML 요소(Element)에 직접 CSS속성(Property)을 지정.
- HTML 요소에 별도의 CSS 속성을 지정하지 않아도 속성값이 적용됨.

```
1 <HTML>
2 <HEAD>
3 <TITLE>Type Selector</TITLE>
4 <STYLE type="text/css">
5 <!--
6 H1 {color:green; text-align:center;}
7 -->
8 </STYLE>
9 </HEAD>
10 <BODY>
11 <H1>Green and bold text</H1>
12 </BODY>
13 </HTML>
```

Green and bold text

일반 선택자 (Selector) 요소

[예제파일: 02_03_클래스선택자]

□ 클래스 선택자(Class Selector)

- 원하는 HTML Element에만 선택적으로 CSS속성을 지정
- 가장 많이 사용되는 Selector
- 사용방법 :
 - Period(. :點)를 찍고 공백없이 영문 대소문자 또는 Hyphen(-), Underscore(_)사용
 - 기호나 숫자로 시작하면 안됨 (MS Internet Explorer경우 작동)**

```
1 <HTML>
2   <HEAD>
3     <TITLE>Class selector</TITLE>
4     <STYLE type="text/css">
5       <!--
6       .title {font-size:18px; color:red; text-align:center; font-weight:bold;}
7       -->
8     </STYLE>
9   </HEAD>
10  <BODY>
11    <P class="title">Class selector</P>
12  </BODY>
13 </HTML>
```

Class selector

일반 선택자 (Selector) 요소

[예제파일: 02_03_특정클래스선택자]

□ 특정 요소(Element)에만 적용되는 Class Selector

- 원하는 HTML Element에만 CSS속성을 지정

```
1 <HTML>
2   <HEAD>
3     <TITLE>특정 Element 에만 적용되는 Class</TITLE>
4     <STYLE type="text/css">
5       <!--
6       p {color:red; font-weight:bold;}
7       span {color:red; font-weight:bold;}
8       p.normal {color:black; font-weight:normal;}
9       -->
10    </STYLE>
11  </HEAD>
12  <BODY>
13    <P>Red & bold text 1</P>
14    <P>Red & bold text 2</P>
15    <P class="normal">Normal text</P>
16    <SPAN class="normal">Normal text</SPAN>
17  </BODY>
18 </HTML>
```

Red & bold text 1

Red & bold text 2

Normal text

Normal text

일반 선택자 (Selector) 요소

[예제파일: 02_03_ID선택자]

□ ID Selector

- Class Selector와 마찬가지로 사용
- DHTML이나 Javascript등 동적인 웹페이지를 만들거나 데이터를 다루는데 이용
- 사용법 : Hash(#)을 붙임

```
1 <HTML>
2 <HEAD>
3 <TITLE>특정 Element 에만 적용되는 Class</TITLE>
4 <STYLE type="text/css">
5 <!--
6 #myId1 {text-align:center; color:green;}
7 #myId2 {text-align:right; color:royalblue;}
8 -->
9 </STYLE>
10 </HEAD>
11 <BODY>
12 <P id="myId1">ID Selector 1</P>
13 <P id="myId2">ID Selector 2</P>
14 </BODY>
15 </HTML>
16
```

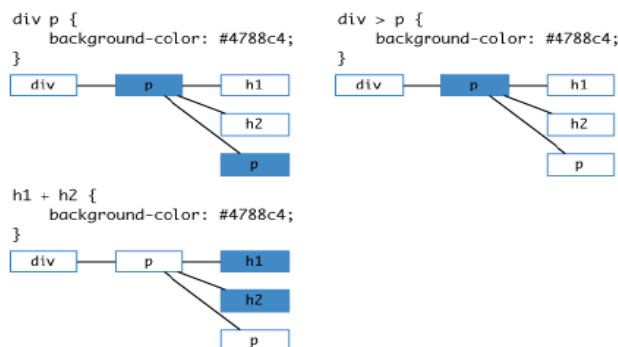
ID Selector 1

ID Selector 2

복합 선택자 (Selector)

[예제파일: 02_02_자손문맥선택자]

□ 복합선택자의 종류



하위 선택자	A B	태그 A로 감싸져 있는 태그 B를 지정
자식 선택자	A > B	태그 A로 감싸져 있는 태그 B중 한 단계 밑에 것을 지정
인접 선택자	A + B	태그 A와 B가 연속으로 나와 있는 것을 지정

그림 12 복합 선택자의 종류 (출처: <http://andsite.net/>)

복합선택자 (Selector) 요소

[예제파일: 02_02_자식문맥선택자]

□ 하위선택자(Descendant Selector ,Contextual Selector)

- HTML Element를 2개 이상 원하는 순서대로 나열하여 표기
- 순서상 뒤에 표기한 선택자에 해당하는 요소를 앞에 표기한 선택자에 해당하는 요소속에서 사용할때만 유효함.
- 1단계 하위요소(child element), 2단계이상 하위요소(decendant element)또한 같이적용

```
1 <html>
2 <head>
3   <title>Contextual Selector</title>
4   <style type="text/css">
5     <!--
6     h2 i {color:green; text-align:center;}
7     -->
8   </style>
9 </head>
10 <body>
11 <h2><i>Green and bold text</i></h2>
12 <!-- 아래와 같이 i 가 h2의 직속 하위(child)가 아니어도 된다. -->
13 <h2><div><i>Green and bold text</i></div></h2>
14 <h1><i>Green and bold text</i></h1>
15 </body>
16 </html>
```

Green and bold text

Green and bold text

Green and bold text

복합선택자 (Selector) 요소

[예제파일: Selector 종류]

□ 자식선택자 (Child Selector)

- Child element(어떤 요소의 한단계 하위요소)에만 적용되는 선택자
- IE6.0은 지원안함 (NN, FF, Opera는 지원)

```
E > F
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">

<html>
<head>
<title>Child Selector</title>
<style type="text/css">
<!--
p > span {color:red;}
-->
</style>
</head>
<body>
<p><span>red</span></p>
<p><div><span>not red</span></div></p>
</body>
</html>
```

red

not red

복합선택자 (Selector) 요소

[예제파일: [Selector 종류](#)]

□ 인접선택자 (:Adjacent Selector)

- 같은 부모(parent)를 가지면서 인접한(sibling)관계인 element에만 적용
- IE6.0 지원안됨

E + F

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

<html>
<head>
<title>first-child Selector</title>
<style type="text/css">
<!--
h1 + h2{font-style:italic;}
-->
</style>
</head>
<body>
<div>
<h1>normal</h1>
<h2>italic</h2>
<h2>normal</h2>
</div>
</body>
</html>
```

normal

italic

normal

가상클래스 선택자 (Selector) 요소

[예제파일: [Selector 종류](#)]

□ 가상클래스 선택자 종류

p:first-child {
background-color: #4788c4;
}

p:lang(ko-KR) {
background-color: #4788c4;
}

first-child 선택자	A:first-child	태그 A로 감싸져 있는 가장 처음 태그를 지정
언어 선택자	A:lang(B)	태그 A중 언어가 B로 설정된 것을 지정
링크 선택자	A:link	태그 A중 링크가 걸려있으면 지정
	A:visited	태그 A중 링크가 걸렸고 사용자가 이미 클릭한 태그를 지정
동적 선택자	A:active	태그 A중 사용자가 마우스를 누르고 있는 태그를 지정
	A:hover	태그 A중 사용자가 마우스 포인터를 위에 올려 놓은 태그를 지정
	A:focus	태그 A중 사용자의 키보드로 입력을 받는 태그를 지정

그림 14. 가상 클래스 선택자 종류 (출처: <http://andsite.net/>)

가상클래스 선택자 (Selector) 요소

[예제파일: [Selector 종류](#)]

□ First-child선택자 (:first-child pseudo class)

- 어떤 요소(element)의 자식요소(child element)가 여러 개 있을 경우 첫번째 자식요소에만 적용됨.
- IE6.0은 지원안함 (NN, FF, Opera는 지원)

E:first-child

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

<html>
<head>
<title>first-child Selector</title>
<style type="text/css">
<!--
div > p:first-child{font-weight:bold;}
-->
</style>
</head>
<body>
<div class="text">
<p>bold text</p>
<p>normal text</p>
<p>normal text</p>
</div>
</body>
</html>
```

bold text

normal text

normal text

가상클래스 선택자 (Selector) 요소

[예제파일: [Selector 종류](#)]

□ 언어선택자 (:lang pseudo class)

- Lang attribute값이 일치하는 element에만 적용
- IE6.0 지원안됨

E:lang(c)

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

<html>
<head>
<title>first-child Selector</title>
<style type="text/css">
<!--
div:lang(ko){color:blue;font-size:12px;}
-->
</style>
</head>
<body>
<div lang="ko">
나는 한글을 사용하고 있습니다.
</div>
<div lang="en">
나는 한글 아니거든...
</div>
</body>
</html>
```

나는 한글을 사용하고 있습니다.

나는 한글 아니거든...

가상클래스 선택자 (Selector) 요소

[예제파일: [Selector 종류](#)]

□ 동적선택자 (Dynamic pseudo class)

- block level과 text level 요소에 적용.
 - hover(마우스올렸을때), active(활성일때), visited(방문한것일때), link(일반링크), focus등
- Microsoft Internet Explorer는 <A>element에만 적용

```
Dynamic pseudo class
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">

<html>
<head>
<title>Dynamic pseudo class</title>
<style type="text/css">
<!--
div.dyna:hover{background-color:gold;}
div.dyna:active{background-color:blue;}
div.dyna:focus{background-color:red;}
-->
</style>
</head>
<body>
<div class="dyna" tabindex="10">Div 1</div>
<div class="dyna" tabindex="20">Div 2</div>
<div class="dyna" tabindex="30">Div 3</div>
</div>
</body>
</html>
```

Div 1
Div 2
Div 3

가상클래스 선택자 (Selector) 요소

[예제파일: [Selector 종류](#)]

□ 동적선택자 (Dynamic pseudo class)

- IE는 <A>element에만 적용안됨

```
p:first-line {
  text-transform: uppercase
}
p:first-letter {
  font-size: 2em;
}

p:before {
  text-content: "[";
}
p:after {
  text-content: "]";
}

<p>The :before, :after pseudo-
element can used to insert</p>

[ The :before, :after pseudo-element ca
used to insert ]
```

:first-line 선택자	A:first-line	태그 A의 문단중 첫번째 줄을 지정
:first-letter 선택자	A:first-letter	태그 A의 문단중 첫번째 글자를 지정
:before 선택자	A:before	태그 A의 문단 앞을 지정
:after 선택자	A:after	태그 A의 문단 뒤를 지정

그림 16 동적 선택자 종류 (출처: <http://andsite.net/>)

그외 선택자 (Selector) 요소

[예제파일: [Selector 종류](#)]

□ 그외의 Selector (:Attribute Selector) – E[attr]

- Element에 사용된 속성(attribute)과 속성값등이 selector로 지정하는 조건과 일치할때 적용

```
E[attr]

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">

<html>
<head>
<title>Attribute Selector 01</title>
<style type="text/css">
<!--
*[title]{background-color:gold;} /* title attribute를 사용하는 문서내의 모든 element(*)에 적용. */
-->
</style>
</head>
<body>
<div>
<h1 title="큰 제목">Gold background</h1>
<h2>나는 gold 아니거든...</h2>
<h3>나도 아니거든...</h3>
</div>
</body>
</html>
```

Gold background

나는 **gold** 아니거든...

나도 아니거든...

그외 선택자 (Selector) 요소

[예제파일: [Selector 종류](#)]

□ 그외의 Selector (:Attribute Selector) – E[attr="attVal"]

- 같은 속성(attribute)의 속성값이 해당 조건과 일치할때 적용
- IE6.0 지원안됨 (역시 안됩니다. —, —ㅋ)

```
E[attr="attVal"]

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">

<html>
<head>
<title>Attribute Selector 02</title>
<style type="text/css">
<!--
a[href="http://www.cadvance.org/"]{font-weight:bold;}
-->
</style>
</head>
<body>
<div>
<a href="http://www.cadvance.org/">나는 굵은 글씨</a><br />
<a href="http://www.cadvance.org/">나는 굵은 글씨</a><br />
<a href="http://www.w3.org/">나는 가는 글씨</a><br />
</div>
</body>
</html>
```

나는 굵은 글씨
나는 굵은 글씨
나는 가는 글씨

□ 그외의 Selector (:Attribute Selector) – E[attr~="attVal"]

- 속성값이 공백으로 분리되어 여러 개일 경우, 값들중에 해당 조건과 일치하는경우 적용
- IE6.0 지원안됨

E[attr~="attVal"]

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

<html>
<head>
<title>Attribute Selector 03</title>
<style type="text/css">
<!--
.a{font-size:12px;}
.b{text-decoration:underline;}
.c{font-family:Arial;}
p[class~="a"]{color:blue;}
-->
</style>
</head>
<body>
<div>
<p class="a b">12px, underline, blue color</p>
<p class="b c">underline, arial</p>
<p class="a c">12px, arial, blue color</p>
</div>
</body>
</html>
```

12px, underline, blue color

underline, arial

12px, arial, blue color

선택자 (Selector) 그룹화

□ Selector Grouping

- 같은 속성(Property)와 같은 속성값을 사용하는 Selector가 여러개일때 Group 만드는것
- 사용법: comma(,)로 구분

```
<STYLE type="text/css">
<!--
H1 {color:green;}
H2 {color:green;}
H3 {color:green;}
-->
</STYLE>

<STYLE type="text/css">
<!--
H1, H2, H3 {color:green;}
-->
</STYLE>
```

```
<STYLE type="text/css">
<!--
H1 {color:red;}
H1 {background-color:silver;}
H1 {font-family:Tahoma;}
H1 {font-size:24px;}
H1 {border:1px solid gray;}
H1 {text-decoration:underline;}
-->
</STYLE>

<STYLE type="text/css">
<!--
H1 {
color:red;
background-color:silver;
font-family:Tahoma;
font-size:24px;
border:1px solid gray;
text-decoration:underline;
}
-->
</STYLE>
```

선택자 (Selector) 우선순위 (Cascading Order)

□ Selector의 우선순위(Cascading Order)

▪ 우선순위 계산법

1. ID Selector 의 갯수를 세어서 개당 100 으로 계산합니다. (=a)
2. Class Selector 의 갯수를 세어서 개당 10 으로 계산합니다. (=b)
3. Type Selector 의 갯수를 세어서 개당 1 로 계산합니다. (=c)

LI .red .level {...} /* a=0 b=2 c=1 -> specificity(명세서) = 21 */

ID Selector는 없고 .red 과 .level 이라는 Class Selector 가 2개 이므로 20을 부여, LI 라는 Type Selector 가 1개 이므로 1을 부여하여 20 + 1 = 21 (참고 : Universal Selector 는 값 '0')

01. Inline StyleSheet가 우선되는 경우 :

Inline StyleSheet는 모든 Selector에 우선

```
1 <HTML>
2 <HEAD>
3 <TITLE>Selector 의 Cascading order</TITLE>
4 <STYLE type="text/css">
5 <!--
6 #green {color:green;}
7 -->
8 </STYLE>
9 </HEAD>
10 <BODY>
11 <P id="green" style="color:red">무슨 색이 될까?</P>
12 </BODY>
13 </HTML>
```

무슨 색이 될까?

02. ID Selector와 Class Select가 동시에 사용된 경우

ID Selector가 우선

```
1 <HTML>
2 <HEAD>
3 <TITLE>id selector vs class selector</TITLE>
4 <STYLE type="text/css">
5 <!--
6 #green {color:green;}
7 .myclass {color:red; font-weight:bold;}
8 -->
9 </STYLE>
10 </HEAD>
11 <BODY>
12 <P id="green" class="myclass">
13 ID 와 Class 에 같은 Property를 쓰고 값이 다를 경우는?
14 </P>
15 </BODY>
16 </HTML>
```

ID 와 Class 에 같은 Property를 쓰고 값이 다를 경우는?

글꼴(FONT)관련 CSS 속성(Property)

□ 태그 관련 속성은 CSS2에서 추천하지 않는 기능

- 다른 Block, Inline관련 element들에 CSS property로 대체 가능하기 때문

□ 관련속성

- font-family:글꼴지정 속성(font name)
- font-size : 글자 크기
- font-style : 글자의 스타일 (보통, 이태릭)
- font-variant : 소문자를 작은 대문자(small-caps)로 변형
- font-weight : 글꼴의 굵기에 관한 속성
- line-height: 행간격을 지정하는 속성
- color : (글자색)
- font : font에 관한 속성을 한꺼번에 지정하는 단축형(short hand) 속성

글꼴(FONT)관련 CSS 속성(Property)

□ 글꼴(font-family) 속성

- 사용법 : { font-family:글꼴이름,글꼴이름 }
- CSS Parser는 앞의 글꼴부터 읽으며 글꼴이 사용자PC에 없을 경우 다음 글꼴을 읽음

```
1 <HTML>
2 <HEAD>
3 <STYLE type="text/css">
4 H1 { font-family : Flubber, "Times New Roman", serif }
5 </STYLE>
6 <BODY>
7 <H1>Flubber라는 글꼴이 없으면 Times New Roman으로 나온다.</H1>
8 <BODY>
9 </HTML>
```

□ 글자모양(font-style) 속성

- 사용법 : { font-style: normal(정상) | italic(이태릭) | oblique(기운것) }
- 기본값은 normal

```
1 <HTML>
2 <HEAD>
3 <STYLE type="text/css">
4 SPAN {font-style : italic}
5 </STYLE>
6 <BODY>
7 여기는 보통 글자입니다.<BR>
8 <SPAN>여기는 기울임꼴이구요.</SPAN>
9 여기는 또 보통입니다.
10 <BODY>
11 </HTML>
```

글꼴(FONT)관련 CSS 속성(Property)

□ font-variant 속성

- 사용법 : { font-variant : normal | small-caps(작은대문자형태) }
- 초기값 : normal
- 한글은 변화없음

```
1 <HTML>
2 <HEAD>
3 <STYLE type="text/css">
4 SPAN {font-variant : small-caps}
5 </STYLE>
6 <BODY>
7 I can't help falling in <SPAN>love</SPAN> with you.
8 <BODY>
9 </HTML>
```

□ font-weight 속성 : 글자의 굵기를 조정

- 사용법 : { font-weight : normal | bold | bolder | lighter | 100 | 200 | 300 | 400(normal) | 500 | 600 | 700(bold) | 800 | 900 }
- 초기값 : normal

```
1 <HTML>
2 <HEAD>
3 <STYLE type="text/css">
4 SPAN {font-weight : 800}
5 </STYLE>
6 <BODY>
7 I can't help falling in <SPAN>love</SPAN> with you.
8 <BODY>
9 </HTML>
```

글꼴(FONT)관련 CSS 속성(Property)

□ font-size 속성

- 사용값 : {font-size : 속성값}
 - 절대사이즈 = xx-small | x-small | small | medium | large | x-large | xx-large
Font 요소의 size 속성값이 1~7 : 한단계 커지거나 작아질때마다 1.2배율(Scaling factor)가 적용
 - 상대사이즈 = larger | smaller
 - 비율(percentage) : 상위가 없을 경우 기본크기(12pt)를 기준으로 100분율 상대크기

상대 길이 단위(Relative length units)	
em	The height of the element's font. (지정되거나 상속 받은 font 크기의 상대 크기) 예) <code>{font:12px/1.5em;}</code> 이면 <code>line-height = 12px × 1.5 = 18px</code>
ex	The height of the letter "x". (소문자 "x"의 높이에 대한 상대 크기)
px	Pixels. (화면 해상도에 대한 상대 크기) 예) 해상도(resolution) 800 x 600 에서 1px 은 화면을 수평으로 800등분, 수직으로 600등분으로 나눈 화소(pixel) 1개의 단위.
%	Percentage. (자기가 속해 있는 상위 Tag의 크기 또는 상속받은 값에 대한 100분율 상대 단위)
절대 길이 단위(Absolute length units)	
in	Inches (1 inch = 2.54 centimeters).
cm	Centimeters.
mm	Millimeters.
pt	Points (1 point = 1/72 inches : 1인치를 72등분한 단위).

글꼴(FONT)관련 CSS 속성(Property)

□ line-height 속성

- 행(줄)간격을 다루는 속성
- 사용법 : { line-height : 상대값 | 절대값 | 비율 }
- 기본값 : 12pt

```
1 DIV { line-height: 1.2; font-size: 10pt } /* number */
2 DIV { line-height: 1.2em; font-size: 10pt } /* length */
3 DIV { line-height: 120%; font-size: 10pt } /* percentage */
```

□ color 속성

- 글자색을 다루는 속성
- 사용법 : { color : 색이름 | RGB값 }
- 초기값 : 브라우저마다 다름.

```
1 DIV { color:red } /* 글자이름 */
2 DIV { color:rgb(255,0,0) } /* rgb값 */
3 DIV { color:#FF0000 } /* rgb코드 */
4 DIV { color:#F00 } /* rgb코드 */
```

글꼴(FONT)관련 CSS 속성(Property)

[예제파일: 03 글꼴!]

□ font 속성

- Font-style,font-variant,font-weight,font-size,line-height,font-family등을 한꺼번에 지정할 수 있는 단축형 속성
- 속성값 : (반드시 1,2,3 순서대로 작성)
 1. font-style,font-variant,font-weight 중 하나 또는 전부
 2. font-size/line-height (font-size만 쓸 경우 / 생략가능)
 3. font-family

```
{ font : bold 12px/1.3em Verdana } /*bold체 12px 글씨 크기에 행 간격 1.3em 글꼴 Verdana*/  
{ font : italic 12px Tahoma } /*기울임체 12px 글씨 크기, 글꼴 Tahoma*/  
{ font : italic small-caps bold 12px/1.3em Arial } /*Full option*/  
{ font : 12px/1.5em bold sans-serif } /*bold가 순서에 맞지 않아서 이후의 sans-serif 글꼴은 무시*/
```

(글)꼴에~ style 모자(small-caps) 쓴 굵은(bold)것이 " 12/간격 가쪽" 이다.

TEXT(단어나 문장)에 관한 CSS속성

□ 글자들,공간들, 단어들과 문단들의 보여지는 표현속성을 정의

□ 관련속성

- text-align :정렬 방식에 대한 속성
- text-decoration : 텍스트의 장식에 대한 속성
- text-indent : 들여쓰기
- text-transform :대문자화
- white-space : 문자사이의 공백처리
- vertical-align : 수직정렬
- letter-spacing : 글자간의 간격
- word-spacing : 단어간의 간격
- line-height : 줄간격

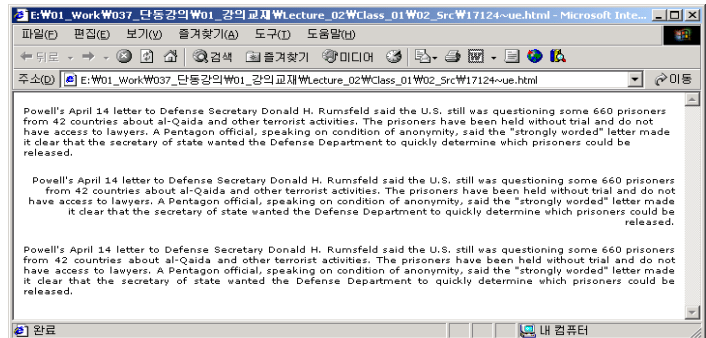
TEXT(단어나 문장)에 관한 CSS속성

[예제파일: 231 텍스트 정렬!]

□ text-align : 정렬 방식에 대한 속성

- 사용법 : { text-align : left | right | justify }
- 초기값 : 사용자 환경(User Agent:Web browser)에 따라 틀림

```
1 <HTML>
2 <HEAD>
3 <STYLE type="text/css">
4 P.left { text-align: left; }
5 P.right { text-align: right; }
6 P.justify { text-align: justify; }
7 </STYLE>
8 <BODY>
9   <p class="left">여기는 왼쪽 정렬이다.</p>
10  <p class="right">여기는 가운데 정렬이다.</p>
11  <p class="justify">여기는 왼쪽 정렬이다.</p>
12 </BODY>
13 </HTML>
```



TEXT(단어나 문장)에 관한 CSS속성

[예제파일: 231 텍스트 장식!]

□ text-decoration : 텍스트장식(Decoration)에 대한 속성

- 사용법 : { text-decoration : none | underline | overline | line-through | blink }

```
1 A#sample00:link {color:#0079C5;text-decoration:none}
2 A#sample00:visited {color:#809BAD;text-decoration:none}
3 A#sample00:active {color:#0079C5;text-decoration:none}
4 A#sample00:hover {color:RED;text-decoration:none;}
5
6 SPAN#sample01 { text-decoration:none}
7 SPAN#sample02 { text-decoration:underline }
8 SPAN#sample03 { text-decoration:overline }
9 SPAN#sample04 { text-decoration:underline overline}
10 SPAN#sample05 { text-decoration:line-through }
11 SPAN#sample06 { text-decoration:blink ;color:orange}
12 SPAN#sample07 { text-decoration:underline overline line-through }
```

기본 링크이다.

밑줄이 없는 링크이다.

Powell's April 14 letter to Defense Secretary Donald H. Rumsfeld said the U.S. still was questioning some 660 prisoners from 42 countries about al-Qaida and other terrorist activities. The prisoners have been held without trial and do not have access to lawyers. A Pentagon official, speaking on condition of anonymity, said the "strongly worded" letter made it clear that the secretary of state wanted the Defense Department to quickly determine which prisoners could be released.

TEXT(단어나 문장)에 관한 CSS속성

[예제파일: 231 텍스트 들여쓰기.1]

□ text-indent : 들여쓰기에 대한 속성

- 사용법 : { text-indent : 절대값(px,pt,em) | 배율 (%) }

```
1 <HTML>
2 <HEAD>
3 <STYLE type="text/css">
4 P { text-indent : 2em }
5 </STYLE>
6 <BODY>
7 <P>
8 Powell's April 14 letter to Defense Secretary Donald H. Rumsfeld said the U.S. still was
9 questioning some 660 prisoners from 42 countries about al-Qaida and other terrorist activities.
10 </P>
11 <P>
12 The prisoners have been held without trial and do not have access to lawyers. A Pentagon official,
13 speaking on condition of anonymity, said the "strongly worded" letter made it clear that the secretary
14 of state wanted the Defense Department to quickly determine which prisoners could be released.
15 </P>
16 <BODY>
17 </HTML>
```

Powell's April 14 letter to Defense Secretary Donald H. Rumsfeld said the U.S. still was questioning some 660 prisoners from 42 countries about al-Qaida and other terrorist activities.

The prisoners have been held without trial and do not have access to lawyers. A Pentagon official, speaking on condition of anonymity, said the "strongly worded" letter made it clear that the secretary of state wanted the Defense Department to quickly determine which prisoners could be released.

TEXT(단어나 문장)에 관한 CSS속성

[예제파일: 231 텍스트 대소문자.1]

□ text-transform : 대소문자에 대한 속성

- 사용법 : { text-transform : capitalize | uppercase | lowercase | none }
- 속성값
 - capitalize : 첫글자를 대문자로
 - uppercase : 글자전체를 대문자로
 - lowercase : 글자전체를 소문자로

```
1 <HTML>
2 <HEAD>
3 <STYLE type="text/css">
4 DIV.cap {text-transform : capitalize}
5 DIV.up {text-transform : uppercase}
6 DIV.low {text-transform : lowercase}
7 </STYLE>
8 <BODY>
9 <DIV class=cap>uppercases the first character of each word</DIV><br>
10 <DIV class=up>uppercases all letters of the element</DIV><br>
11 <DIV class=low>LOWERCASES ALL LETTERS OF THE ELEMENT</DIV><br>
12 <BODY>
13 </HTML>
```

Uppercases The First Character Of Each Word
UPPERCASES ALL LETTERS OF THE ELEMENT
lowercases all letters of the element

TEXT(단어나 문장)에 관한 CSS속성

[예제파일: 231 텍스트 공백처리]

□ white-space : 여러 개의 공백(space)처리 속성

- HTML에서의 공백처리 : (non-breaking space)
- 사용법 : { white-space : normal | pre | nowrap }
- 속성값
 - normal : 공백(space)를 1개밖에 인정안함
 - pre(pre-formatted) : 공백(space)개수 만큼 보여줌 (DTD가 Strict일때만 작동)
<PRE> element 같이 사용자가 입력한 모습 그대로 공백을 화면에 출력
 - nowrap : 줄바꿈 금지

 element 처럼 강제로 줄을 바꾸기 전에는 계속 같은 줄에 출력

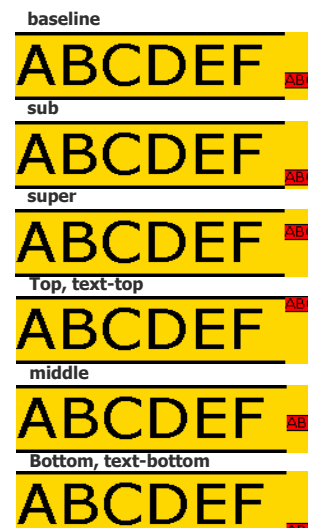
```
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Strict//EN">
2 <HTML>
3 <HEAD><TITLE>white-space Property</TITLE>
4 <STYLE type="text/css">
5   .white1 { white-space : normal; }
6   .white2 { white-space : pre; }
7   .white3 { white-space : nowrap }
8 </STYLE>
9 </HEAD>
10 <BODY>
11 <P class="white1"
12   onMouseOver="this.className='white2';"
13   onMouseDown="this.className='white3';"
14   onMouseOut="this.className='white1';">
15
16 powell's April 14 letter to Defense Secretary Donald H. Rumsfeld said
17
18 the U.S. still was questioning some 660 prisoners from 42 countries
19 about al-Qaida and other terrorist activities.
20 </P>
21 </BODY>
```

TEXT(단어나 문장)에 관한 CSS속성

[예제파일: 231 텍스트 수직정렬]

□ vertical-align : Box속의 내용물의 수직정렬을 지정

- 표<Table>의 셀(Cell)과 같은 Box속에 들어있는 내용물의 수직정렬을 지정하는 속성(Property)
- 사용법 : { vertical-align: baseline | sub | super | top | text-top | middle | bottom | text-bottom | 비율(%) | 길이 값(length) }
- 속성
 - Baseline(기초라인) : 박스의 기초라인을 모체박스의 기초라인에 정렬
 - Sub(아래첨부기능) : 박스의 기초라인을 모체박스의 아래첨자 (subscript)에 맞춤
 - Super(위 첨부위치) : 박스의 기초라인을 모체박스의 위첨자 (superscript)에 맞춤
 - Top(맨위) : 박스의 맨위를 라인박스의 맨위에 정렬
 - Text-top(텍스트 맨위) : 박스의 맨위를 모체 엘리먼트 글꼴의 맨위에 맞춤
 - Middle(중앙)** : 박스의 수직적 중간점을 그 모체박스중간에 정렬
 - Bottom(바닥) : 박스의 바닥을 라인박스의 바닥에 정렬
 - Text-bottom(텍스트맨아래) : 박스의 바닥을 모체 엘리먼트 글꼴의 바닥에 정렬
 - 비율(%) : 박스를 "line-height"값의 백분율거리만큼 올리거나(양수) 내린다(음수). 값이 '0%'면 baseline과 같음
 - 길이(length) : 박스를 거리길이 만큼 올리거나(양수) 내림(음수)



TEXT(단어나 문장)에 관한 CSS속성

[예제파일: 231 텍스트 자간]

□ letter-spacing : 자간(글자간의 간격)을 조정

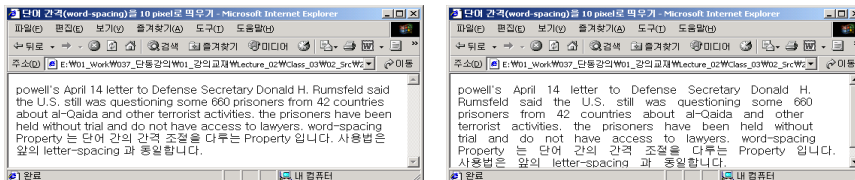
- 자간을 늘리거나(양수) 줄 일때(음수) 사용
- 영어의 경우 자간이 좁아서 혼동 우려하여 만들어짐 (예 : class => dass)
- 사용법 : { letter-spacing : normal | 길이값(length,) }

```
1 <HTML>
2   <HEAD>
3     <TITLE>특정 부분의 자간 조절하기</TITLE>
4     <STYLE type="text/css">
5     <!--
6     .letter1 { letter-spacing : 20px; }
7     -->
8   </STYLE>
9   </HEAD>
10  <BODY>
11    class <br><!-- letter-spacing을 조절하지 않은 상태 -->
12    <SPAN class="letter1">class</SPAN> <!-- letter-spacing 을 조절한 상태 -->
13  </BODY>
14 </HTML>
```

class
c | a s s

□ word-spacing : 단어간의 간격을 조절

- 사용법 : { word-spacing : normal | 길이값(length,) }



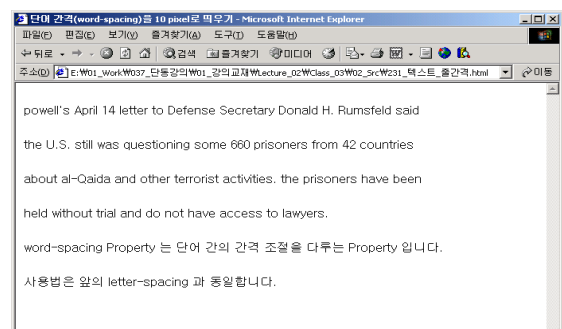
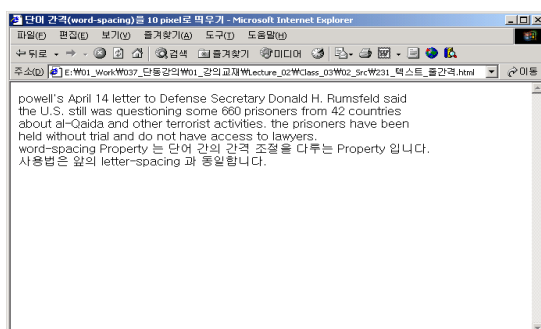
TEXT(단어나 문장)에 관한 CSS속성

[예제파일: 231 텍스트 줄간격]

□ line-height : 줄간의 간격을 조정

- 사용법 : { line-height : normal | 수치 | 길이값(length) | 백분율 }
- 최초값 : normal
- 속성값
 - Normal : 사용자환경(User Agent)에 따라 틀림.
 - 수치 : 해당 element의 글꼴 크기에 곱한 수치 (양수값만 사용)
 - 길이값 : 박스높이는 길이로 설정(양수값만 사용)
 - 백분율(%) : 해당 element의 글꼴크기를 곱한 백분율

```
1 DIV { line-height: 1.2; font-size: 10pt } /* 수치 */
2 DIV { line-height: 1.2em; font-size: 10pt } /* 길이 */
3 DIV { line-height: 120%; font-size: 10pt } /* 백분율 */
```



사용자 인터페이스 (User Interface)에 관한 속성

□ 화면에 출력될 element들에 디자인요소를 추가하는 속성

- 커서의 모양을 변경.
- 리스트의 형태를 변경 및 이미지 삽입,
- 문서의 배경색, 배경이미지
- 스크롤바(Scroll bar)의 색상 변경
- 보임, 안보임 조정

□ 관련속성














- cursor
- Scrollbar
- Classification(목록) : List-style-type, list-style-image, list-style-position, list-style
- Display
- Background-color, background-image, background-attachment, background-repeat, background

사용자 인터페이스 -마우스커서(Cursor)에 관한 속성

[예제파일: 232_UI 커서]

□ cursor : 사용자환경(UserAgent)의 마우스모양을 바꾸는 역할

- 클라이언트(Client PC)의 사용자환경에 따라 다름.
- 해당 클라이언트의 Cursor를 확인하는 법 : C:\Windows\Cursors
- 사용법 : { cursor : auto | crosshair | default | pointer | move | e-resize | ne-resize | nw-resize | n-resize | se-resize | sw-resize | s-resize | w-resize | text | wait | help | url("주소") }
- 최초값 : auto

| Cell 위에 마우스를 올리면 cursor 형태가 나옵니다. | | | |
|---|---|---|---|
|  auto |  crosshair |  default |  pointer |
|  move |  text |  wait |  help |
|  e-resize |  w-resize |  s-resize |  n-resize |
|  se-resize |  sw-resize |  ne-resize |  nw-resize |

Custom Cursor : {Cursor :url("http://testdomain.com/curosr/pen_m.cur") }

| Custom cursor sample | | | |
|---|---|--|---|
| pen_m.cur  | 3dgarro.cur  | barber.ani  | horse.ani  |

| IE 6.0 Only cursor | | | |
|--------------------|------------|-------------|------------|
| hand | progress | not-allowed | no-drop |
| vertical-text | all-scroll | col-resize | row-resize |

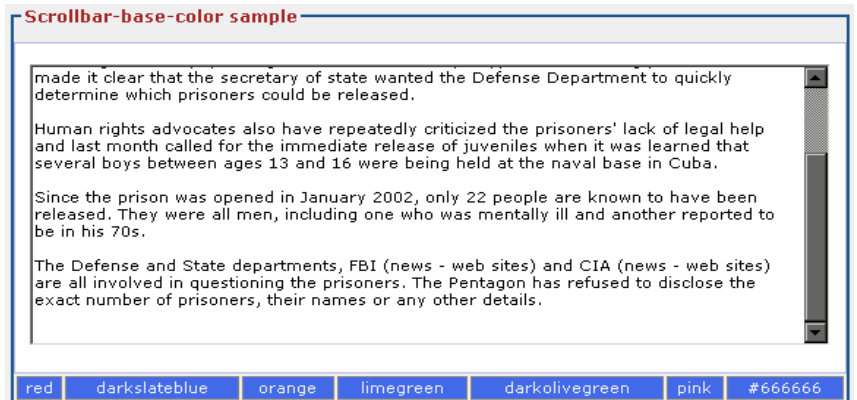
사용자 인터페이스 -스크롤바(Scrollbar)에 관한 속성

[예제파일: 232 UI 스크롤바]

□ scrollbar : 사용자환경(UserAgent)의 스크롤바모양을 바꾸는 역할

- W3C 표준권고안에는 없는 속성(property)
- MS Internet Explorer만 지원
- 사용법 :

```
body {  
  scrollbar-3dlight-color: 칼라값;  
  scrollbar-highlight-color: 칼라값;  
  scrollbar-face-color: 칼라값;  
  scrollbar-shadow-color: 칼라값;  
  scrollbar-darkshadow-color: 칼라값;  
  scrollbar-track-color: 칼라값;  
  scrollbar-arrow-color: 칼라값;  
}
```



사용자 인터페이스 -목록(Classification)속성

□ 목록(Classification) :

- ,,만으로 할수 없는 글머리 기호의 위치 지정, 글머리 기호대신에 이미지를 넣는 속성

□ 속성(Property) 목록 :

- **List-style-type** : 글머리 번호, 기호의 형태를 지정
- **List-style-image** : 글머리 번호, 기호에 이미지를 넣는 속성
- **List-style-position** : 글머리 번호, 기호의 문장에서의 위치를 지정
- **List-style** : list관련 CSS 속성의 단축형(short hand) 속성
- **display** : SPAN(Inline element:선요소), DIV(Block element:면적요소)의 element 속성을 변경

사용자 인터페이스 -목록(Classification)속성

[예제파일: 232 UI 목록 글머리기호]

□ List-style-type : 글머리 번호, 기호의 형태를 지정

- 순서가 없는 목록(, ●, ○, ■) 이나 순서가 있는 목록(, 1, a, A, I, i)의 글머리 기호나 번호의 형태를 지정하는 속성.
- 반드시 , 에 지정
- 사용법 : {list-style-type : disc | circle | square | decimal | lower-roman | upper-roman | lower-alpha 등}
- 기본값 : disc

```
1 <ul>
2   <li style="list-style-type: disc;"><b>disc</b></li>
3   <li style="list-style-type: circle;"><b>circle</b></li>
4   <li style="list-style-type: square;"><b>square</b></li>
5   <li style="list-style-type: none;"><b>none</b></li>
6   <li style="list-style-type: decimal;"><b>decimal</b></li>
7   <li style="list-style-type: lower-alpha;"><b>lower-alpha</b></li>
8   <li style="list-style-type: upper-alpha;"><b>upper-alpha</b></li>
9   <li style="list-style-type: lower-roman;"><b>lower-roman</b></li>
10  <li style="list-style-type: upper-roman;"><b>upper-roman</b></li>
11  <li style="list-style-type: decimal-leading-zero;"><b>decimal-leading-zero</b></li>
12  <li style="list-style-type: lower-greek;"><b>lower-greek</b></li>
13  <li style="list-style-type: lower-latin;"><b>lower-latin</b></li>
14  <li style="list-style-type: upper-latin;"><b>upper-latin</b></li>
15  <li style="list-style-type: hebrew;"><b>hebrew</b></li>
16  <li style="list-style-type: armenian;"><b>armenian</b></li>
17  <li style="list-style-type: georgian;"><b>georgian</b></li>
18  <li style="list-style-type: cjk-ideographic;"><b>cjk-ideographic</b></li>
19  <li style="list-style-type: hiragana;"><b>hiragana</b></li>
20  <li style="list-style-type: katakana;"><b>katakana</b></li>
21  <li style="list-style-type: hiragana-iroha;"><b>hiragana-iroha</b></li>
22  <li style="list-style-type: katakana-iroha;"><b>katakana-iroha</b></li>
23 </ul>
```

- disc
- circle
- square
- none
- 5. decimal
- f. lower-alpha
- G. upper-alpha
- viii. lower-roman
- IX. upper-roman
- decimal-leading-zero
- lower-greek
- lower-latin
- upper-latin
- hebrew
- armenian
- georgian
- cjk-ideographic
- hiragana
- katakana
- hiragana-iroha
- katakana-iroha

사용자 인터페이스 -목록(Classification)속성

[예제파일: 232 UI 목록 글머리기호]

□ List-style-image : 글머리 번호, 기호에 이미지를 넣는 속성

- 사용법 : {list-style-image : none | url("이미지주소")}
- 초기값 : none
- List-style-image를 고려하지 않고 글앞에 이미지 넣기

```
1 <IMG src="/img/arrow.gif" width="12" height="10" alt=""> 움직이지 않는 화살표...<BR>
2 <IMG src="/img/arrow.gif" width="12" height="10" alt=""> 움직이지 않는 화살표...<BR>
3 <IMG src="/img/arrow.gif" width="12" height="10" alt=""> 움직이지 않는 화살표...<BR>
```

- ➡ 움직이지 않는 화살표...
- ➡ 움직이지 않는 화살표
- ➡ 움직이지 않는 화살표...

- List-style-image를 고려한 글앞에 이미지 넣기

```
1 <STYLE type="text/css">
2 <!--
3 .arrow { list-style-image : url(/img/arrow.gif) }
4 .arrow_ani { list-style-image : url(/img/arrow_ani.gif) }
5 -->
6 </STYLE>
7
8 <UL class="arrow">
9   <LI>움직이지 않는 화살표 - 1</LI>
10  <LI>움직이지 않는 화살표 - 2</LI>
11  <LI class="arrow_ani">움직이는 화살표</LI>
12  <LI>움직이지 않는 화살표 - 3</LI>
13 </UL>
```

- ➡ 움직이지 않는 화살표 - 1
- ➡ 움직이지 않는 화살표 - 2
- ➡ 움직이는 화살표
- ➡ 움직이지 않는 화살표 - 3

사용자 인터페이스 -목록(Classification)속성

[예제파일: 232_UI_목록_글머리위치]

□ List-style-position : 글머리 번호,기호의 문장에서의 위치를 지정

- 글머리 기호나 문자가 문장속에 들어가거나(inside), 문장왼쪽 밖으로 따로 나와있거나(outside)의 여부를 다루는 속성.
- 초기값 : outside
- 사용법 : {list-style-position : inside | outside}

```
1 <STYLE type="text/css">
2   UL { list-style: outside }
3   UL.compact { list-style: inside }
4 </STYLE>
5
6
7 <UL>
8   <LI>first list item comes first
9   <LI>second list item comes second
10 </UL>
11
12 <UL class="compact">
13   <LI>first list item comes first
14   <LI>second list item comes second
15 </UL>
```

- first list item comes first
 - second list item comes second
- ↑
- first list item comes first
 - second list item comes second
- ↑

사용자 인터페이스 -목록(Classification)속성

[예제파일: 232_UI_목록_단축형]

□ List-style : 리스트관련 CSS속성의 단축형(short hand)속성

- 사용법 : {list-style : none | list-style-type list-style-position listy-style-image}
- 단축형에는 초기값이 없음.

```
1 <HTML>
2 <HEAD>
3   <TITLE>경고: 카스케이드 때문에 기대하지 않았던 효과가 발생</TITLE>
4   <STYLE type="text/css">
5     OL.alpha LI { list-style: lower-alpha }
6     UL LI { list-style: disc }
7   </STYLE>
8 </HEAD>
9 <BODY>
10  <OL class="alpha">
11    <LI>level 1
12    <UL>
13      <LI>level 2
14    </UL>
15  </OL>
16 </BODY>
17 </HTML>
```

a. level 1
a. level 2

```
1 <HTML>
2 <HEAD>
3   <TITLE>경고: 카스케이드 때문에 기대하지 않았던 효과가 발생</TITLE>
4   <STYLE type="text/css">
5     OL.alpha > LI { list-style: lower-alpha }
6     UL LI { list-style: disc }
7   </STYLE>
8 </HEAD>
9 <BODY>
10  <OL class="alpha">
11    <LI>level 1
12    <UL>
13      <LI>level 2
14    </UL>
15  </OL>
16 </BODY>
17 </HTML>
```

a. level 1
♦ level 2

UL { list-style: upper-roman inside } /* 어떤 UL */
UL > UL { list-style: circle outside } /* UL의 어떤 UL 자식(child) */

OL.alpha LI { list-style: lower-alpha } /* OL의 어떤 LI 하위(descendant) */
OL.alpha > LI { list-style: lower-alpha } /* OL의 어떤 LI 자식(child) */

사용자 인터페이스 -DISPLAY속성

□ display property 특징

- Inline element(선요소)나 Block element(면요소)가 화면상에 출력(Render)되는 방식을 다루는 속성.
 - 화면에 나오는 모든 element는 Inline, Block 두가지로 구성되어 있음.
 - Inline-Level element : 같은 줄에서 줄바꿈없이 연속적으로 이어지는 element , SPAN, FONT)
 - Block-Level element : DIV, P, H1~H6과 같이 줄바꿈이 생기는 element : 위아래로 줄바꿈이 됨.
 - Inline-level element ↔ Block-level element로 속성을 변경할수 있게 해줌
 - CSS로 화면 레이아웃(Layout)을 잡을때 상당히 중요한 역할을 함.
- 화면에 나타나지 않게(none)함
- 사용법 : {display: inline | block | list-item | table-header-group | table-footer-group | none | inline-block(* IE Only) }

사용자 인터페이스 -DISPLAY속성

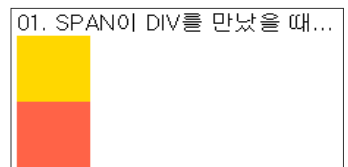
[예제파일: 232 UI 목록 DISPLAY01.]

예시 1) SPAN(Inline요소)이 DIV(Block요소)를 만났을때

```
1 01. SPAN이 DIV를 만났을 때...<br>
2 <SPAN style="width:50px; height:50px; background-color:gold; "></SPAN>
3 <DIV style="width:50px; height:50px; background-color:tomato; "></DIV>
```

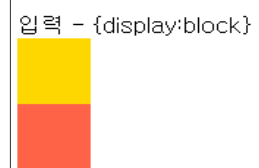
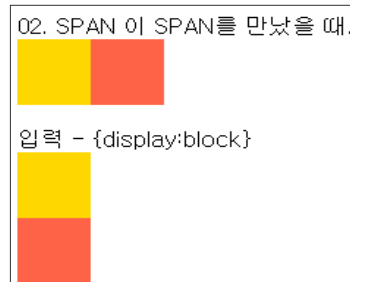
DIV를 {display:inline}사용하여 inline요소로 바꿈

```
1 <br>입력 - {display:inline} <br>
2 <SPAN style="width:50px; height:50px; background-color:gold; "></SPAN>
3 <DIV style="width:50px; height:50px; background-color:tomato; display:inline; "></DIV>
```



예시 2) SPAN(Inline요소)이 SPAN(Inline요소)를 만났을때

```
1 <br>02. SPAN 이 SPAN를 만났을 때...<br>
2 <SPAN style="width:50px; height:50px; background-color:gold; "></SPAN>
3 <SPAN style="width:50px; height:50px; background-color:tomato; "></SPAN>
4
5 <br><br>입력 - {display:block} <br>
6 <SPAN style="width:50px; height:50px; background-color:gold; "></SPAN>
7 <SPAN style="width:50px; height:50px; background-color:tomato; display:block; "></SPAN>
8 <br><br>
```



사용자 인터페이스 -DISPLAY속성

[예제파일: 232 UI 목록 DISPLAY02_list_item]

□ display :list-item속성

- 이나과 같이 화면에 출력시킴.
- SPAN같은 Inline요소를 사용해도 LI 와 같이 block요소로 화면에 render됨.

```
1 <html>
2 <head>
3 <title>display : list-item</title>
4 <style type="text/css">
5 <!--
6 span.symbol{display:list-item; list-style-type:square;}
7 span.deca{display:list-item; list-style-type:decimal;}
8 span.roman_U{display:list-item; list-style-type:upper-roman;}
9 span.alpah_U{display:list-item; list-style-type:upper-alpha;}
10 -->
11 </style>
12 </head>
13 <body>
14 <div style="padding-left:50px;">
15 <span class="symbol">Square 1</span>
16 <span class="symbol">Square 2</span>
17 <span class="symbol">Square 3</span>
18 <br /><br />
19 <span class="deca">Number 1</span>
20 <span class="deca">Number 2</span>
21 <span class="deca">Number 3</span>
22 <br /><br />
23 <span class="roman_U">Roman 1</span>
24 <span class="roman_U">Roman 2</span>
25 <span class="roman_U">Roman 3</span>
26 <br /><br />
27 <span class="alpah_U">Alphabet 1</span>
28 <span class="alpah_U">Alphabet 2</span>
29 <span class="alpah_U">Alphabet 3</span>
30 </div>
31 </body>
```

▪ Square 1
▪ Square 2
▪ Square 3

6. Number 1
7. Number 2
8. Number 3

XI. Roman 1
XII. Roman 2
XIII. Roman 3

P. Alphabet 1
Q. Alphabet 2
R. Alphabet 3

사용자 인터페이스 -DISPLAY속성

[예제파일: 232 UI 목록 DISPLAY03_TABLE_HEADER_GROUP]

□ display :table-header-group, table-footer-group속성

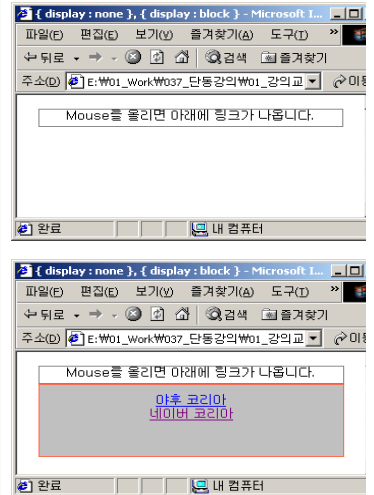
- <TABLE>의 <THEAD>와<TFOOT>에서 작동.
- 이 속성은 컴퓨터 모니터상에서 출력할 목적으로 만든 속성이 아니고 한페이지가 넘는 여러페이지 분량의 Table을 프린터로 출력할때 매 페이지마다 테이블 머리말(THEAD)과 꼬리말(TFOOT)을 넣어주기 위해서 사용.

```
1 <HTML>
2 <HEAD>
3 <TITLE>Table HEADER와 FOOTER를 매 페이지마다 프린트 하기</TITLE>
4 </HEAD>
5 <BODY>
6 <TABLE border="1" cellpadding="0" cellspacing="0" width="100%">
7 <THEAD align="center" style="display:table-header-group;">
8 <TR><TD colspan="5">Table Header</td></TR>
9 </THEAD>
10 <TFOOT style="display:table-footer-group;">
11 <TR><TD colspan="5">Table Footer</td></TR>
12 </TFOOT>
13 <TBODY bgcolor="gold" align="center">
14 <TR>
15 <TD>Cell</TD>
16 <TD>Cell</TD>
17 <TD>Cell</TD>
18 <TD>Cell</TD>
19 <TD>Cell</TD>
20 </TR>
21 .
22 .
23 .
24 .
25 .
26 .
27 .
28 </TABLE>
```

□ display :none, block

- "보이기(block)","감추기(none)" 기능을 수행.

```
1 <HTML>
2 <HEAD>
3 <TITLE>{ display : none }, { display : block }</TITLE>
4 <STYLE type="text/css">
5 <!--
6 .menuTitle { width:300px;height:30px;border:1px solid #808080;cursor:hand; }
7 .blk {display:inline-block;width:300px;height:300px;border:1px solid tomato;
8 background-color:#c0c0c0;padding:10px;}
9 .non {display:none;position:relative;top:-19px;left:0px;}
10 -->
11 </STYLE>
12 </HEAD>
13 <BODY>
14 <CENTER>
15 <DIV>
16 <DIV class="menuTitle" onMouseOver="div_1.className='blk';"
17 onMouseout="div_1.className='non';">
18 Mouse를 올리면 아래에 링크가 나옵니다.
19 </DIV>
20 <DIV id="div_1" class="non" onClick="div_1.className='non';"
21 onMouseOver="div_1.className='blk';" onMouseout="div_1.className='non';">
22 <A href="http://kr.yahoo.com" target="_blank">야후 코리아</A><BR>
23 <A href="http://www.naver.com" target="_blank">네이버 코리아</A>
24 </DIV>
25 </DIV>
26 </CENTER>
27 </BODY>
28 </HTML>
```



사용자 인터페이스 -배경색(Background)속성

□ 배경색(Background) :

- 문서나 박스형태의 표의 배경색, 배경이미지등에 관한 속성

□ 속성(Property) 목록 :

- Background-color : 문서의 배경색을 지정하는 속성
- Background-image : 문서의 배경을 이미지로 지정하는 속성
- Background-attachment : 문서의 배경이미지를 고정하거나 scroll여부를 지정하는 속성
- Background-repeat : 배경 그림의 반복 여부를 지정하는 속성
- Background-position: 배경그림의 위치를 지정하는 속성
- Background : 위의 모든 속성을 한꺼번에 지정하는 단축형(short hand) 속성

[예제파일: 232 UI 목록 배경색01 색깔,

- Inline요소나 Block요소의 배경색을 지정하는 속성
- 의 color 속성과 같음.
- 사용법 : {background-color: 칼라색 | transparent}
- 기본값 : transparent
- 속성값 :
 - Transparent color: 투명색, 상위 element의 배경색이 비취짐.(Cascading)
 - color : 색이름 | RGB값

A screenshot of a Microsoft Internet Explorer browser window. The title bar reads "새 창 - Microsoft Internet Explorer". The address bar shows the URL "E:\W01_Work\W037_단통강의W01_강의교과기". The main content area is a solid red rectangle, indicating that the page content has been redacted. The status bar at the bottom shows "한글" and "내 컴퓨터".

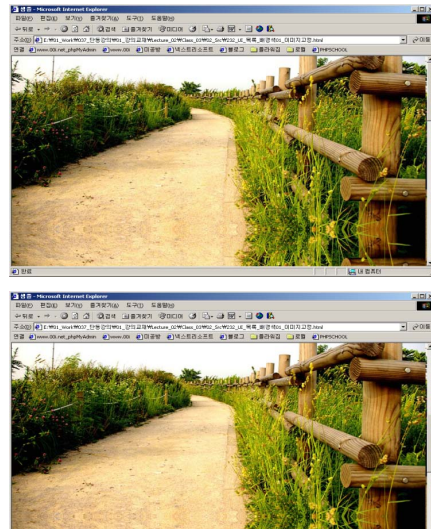
[예제파일: 232 UI 목록 배경색01 이미지.]

사용자 인터페이스 -배경색(Background)속성

□ Background-attachment :

- 배경이미지를 고정(fixed)시키거나 scroll 시킴
- 사용법 { background-attachment : scroll | fixed }
- 기본값 : scroll

```
1 <HTML>
2 <HEAD>
3   <TITLE>샘플</TITLE>
4 </HEAD>
5 <BODY style="background-image : url(/img/ultari1.jpg);
6   background-attachment:fixed">
7   <SPAN style="color:#FFFFFF;">BODY에 배경 그림 넣기</SPAN>
8   <BR><BR><BR><BR><BR><BR><BR><BR>
9   <BR><BR><BR><BR><BR><BR><BR><BR>
10  <BR><BR><BR><BR><BR><BR><BR><BR>
11  <BR><BR><BR><BR><BR><BR><BR><BR>
12  <BR><BR><BR><BR><BR><BR><BR><BR>
13  <BR><BR><BR><BR><BR><BR><BR><BR>
14  <BR><BR><BR><BR><BR><BR><BR><BR>
15  <BR><BR><BR><BR><BR><BR><BR><BR>
16 </BODY>
17 </HTML>
```

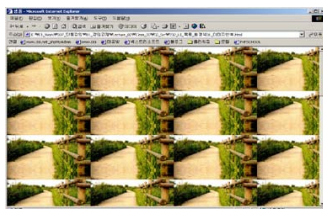


사용자 인터페이스 -배경색(Background)속성

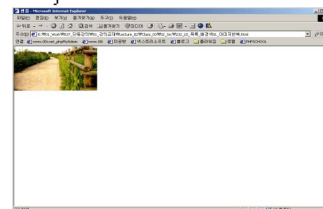
□ Background-repeat :

- 지정한 배경 그림을 반복(repeat)시키거나 시키지 않거나(no-repeat), 수평으로 반복(repeat-x), 수직으로 반복(repeat-y)하는것을 지정

```
body {
  background-image : url("/img/ultari2.jpg")
}
```



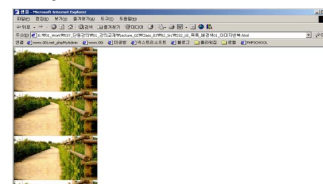
```
body {
  background-image : url("/img/ultari2.jpg");
  background-repeat: no-repeat;
}
```



```
body {
  background-image : url("/img/ultari2.jpg");
  background-repeat : repeat-x;
}
```



```
body {
  background-image : url("/img/ultari2.jpg");
  background-repeat : repeat-y;
}
```

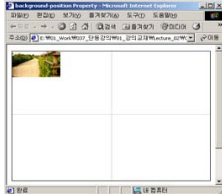


사용자 인터페이스 -배경색(Background)속성

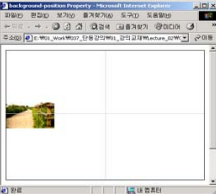
□ Background-position: 배경으로 지정한 그림의 시작지점을 지정하는 속성

- 사용법 : {background-position : 백분율(%) | 길이값(px,em..) | 수평값(left,center,right) 수직값(top,center,bottom)}
- 사용예 : { background-position : left bottom}

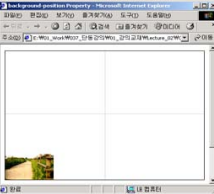
지정하지 않음



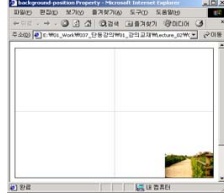
left center



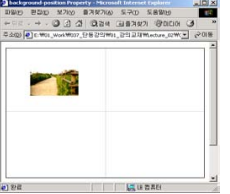
left bottom



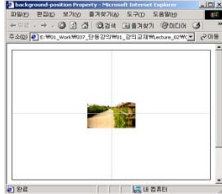
right bottom



50px 50px



50% 50%



Keyword를 Percentage로 환산한 표

| | | | |
|---------------|-----------|---------------|----------------------------|
| left top | 0% 0% | center top | 50% 0% |
| right top | 100% 0% | left center | 0% 50% |
| center center | 50% 50% | right center | 100% 50% |
| left bottom | 0% 100% | center bottom | 50% 100% |
| right bottom | 100% 100% | 값을 안 주면 | left top (0% 0% : Default) |

사용자 인터페이스 -배경색(Background)속성

[예제파일: 232 UI 목록_배경색01 단축형속성]

□ Background : 모든 속성을 한꺼번에 지정하는 단축형(short hand) 속성

- 사용법 : {background : background-color background-image background-repeat background-attachment background-position }
- Background속성은 속성값의 순서에 구애를 받지 않음.

```
body {
    background : gold url("../images/background_position.jpg") center center no-repeat fixed;
}
```

```
body {
    background-color: gold;
    background-image : url("../images/background_position.jpg");
    background-position: center center;
    background-repeat: no-repeat;
    background-attachment:fixed;
}
```

표(Table) 관련속성

□ <TABLE> element에 관련된 속성

- 테이블의 너비기능 : width, table-layout
- 테이블의 높이기능 : height속성으로 지정
- 셀(Cell) 박스안에서의 내용 수평정렬 : text-align (left, center, right)
- 셀(Cell)박스안에서의 내용 수직정렬 : vertical-align (baseline,top,bottom,middle등)

□ 테두리(border)관련 속성

- 테두리 모델 : 분리된 (seperated) 테두리모델, 통합된(collapsing) 테두리 모델
 - 테두리 모델의 구분 속성 : border-collapse
 - 분리된(seperated)테두리 모델 속성들
 - 분리하는 인접 셀 테두리들의 거리 지정 속성 : Border-spacing
 - 빈 셀 주위의 테두리 속성 : empty-cells
 - 통합(collapsing)테두리 모델 속성들
- 테두리 스타일 : border-style
- 테두리 너비 : border-width
- 테두리 색상 : border-color
- 테두리 약식속성 : border-top,border-bottom, border-right, border-left, border

표(Table) 관련속성- TABLE 너비관련 속성

□ table-layout 속성: 테이블 셀(Cell)의 width, height를 고정하거나 않는것을 지정하는 속성

- 사용법 : {table-layout : auto(default) | fixed }
- 테이블의 길이가 길어지면 화면상에 출력하기 위해 계산하는 시간이 길어짐
(참조사이트 : <http://msdn.microsoft.com/workshop/samples/author/dhtml/refs/tables3.htm>)
- **Table-layout:fixed사용**
 - 계산이 빨라짐
 - 계산된 앞부분 부터 화면에 출력(render)됨.
 - Cell의 내용물이 넘쳐도 Cell의 크기에 영향을 주지 않음
- **Table-layout:fixed 미사용**
 - Cell size를 계산하므로 느림.
 - 계산이 끝난후에 화면에 출력
 - Cell속의 내용물(data)이 Cell의 가로,세로 크기에 영향을 준다.

표(Table) 관련속성- TABLE 너비관련 속성

❑ table-layout 사용하는 방법

- <Table>Tag의 첫번째 <TR>의 <TD>에 지정하는 방법
- <Colgroup>과 <Col>요소를 사용하는 방법
- Cell의 width를 지정하지 않은 경우와 내용물(Data)이 Cell을 넘치는(overflow) 경우

```
<table border="1" style="table-layout:fixed;">
<tr height="30">
<td width="10%">R1 C1</td>
<td width="15%">R1 C2</td>
<td width="20%">R1 C3</td>
<td width="25%">R1 C4</td>
<td width="30%">R1 C5</td>
</tr>
<tr height="45">
<td>R2 C1</td>
<td>R2 C2</td>
<td>R2 C3</td>
<td>R2 C4</td>
<td>R2 C5</td>
</tr>
</table>
```

```
<table border="1" style="table-layout:fixed;">
<COLGROUP span="5" style="text-align:center; background:silver;">
<COL span="1" style="width:30%; color:tomato;">
<COL span="2" style="width:15%; color:gold;">
<COL span="2" style="width:20%; color:royalblue;">
</COLGROUP>
<TBODY style="height:30px;">
<tr>
<td>R1 C1</td>
<td>R1 C2</td>
<td>R1 C3</td>
<td>R1 C4</td>
<td>R1 C5</td>
</tr>
<tr>
<td>R2 C1</td>
<td>R2 C2</td>
<td>R2 C3</td>
<td>R2 C4</td>
<td>R2 C5</td>
</tr>
</TBODY>
```

표(Table) 관련속성- TABLE 너비관련 속성

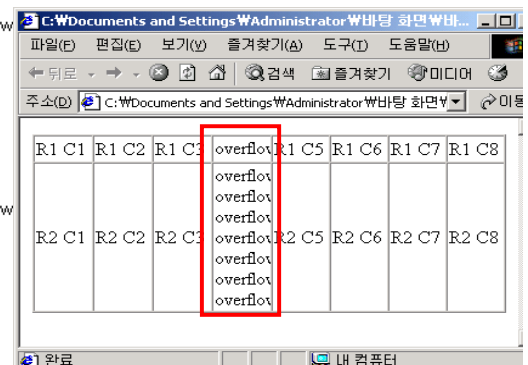
❑ table-layout 사용하는 방법

- 내용물(Data)이 Cell을 넘치는(overflow) 경우 처리

```

1 <HTML>
2 <HEAD>
3 <TITLE></TITLE>
4 </HEAD>
5 <BODY>
6 <table border="1" cellspacing="0" width="100%" style="table-layout:fixed;">
7 <tr height="25">
8 <td>R1 C1</td>
9 <td>R1 C2</td>
10 <td>R1 C3</td>
11 <td>overflow overflow overflow overflow overflow overflow overflow</td>
12 <td>R1 C5</td>
13 <td>R1 C6</td>
14 <td>R1 C7</td>
15 <td>R1 C8</td>
16 </tr>
17 <tr <!-- height를 지정하지 않음 -->
18 <td>R2 C1</td>
19 <td>R2 C2</td>
20 <td>R2 C3</td>
21 <td>overflow overflow overflow overflow overflow overflow overflow</td>
22 <td>R2 C5</td>
23 <td>R2 C6</td>
24 <td>R2 C7</td>
25 <td>R2 C8</td>
26 </tr>
27 </table>
28 </BODY>
29 </HTML>

```



표(Table) 관련속성- 테두리 (Border) 속성

□ 테두리(border)

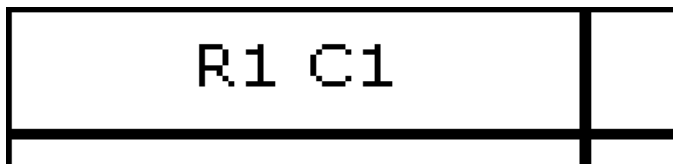
- 테두리 모델 : 분리된 (seperated) 테두리모델, 통합된(collpasing)테두리 모델

□ 테두리(border)모델 구분 속성 : border-collapse

- <Table>element의 Cellspacing속성을 주지 않을시 기본 cellspacing은 2pixel (인접한 Cell의 border가 겹침)
- 사용법 : {border-collapse : separate | collapse }
- 기본값 : seperate

R1 C1	R1 C2	R1 C3
R2 C1	R2 C2	R2 C3
R3 C1	R3 C2	R3 C3
R4 C1	R4 C2	R4 C3

Cellpadding=0 Border:1px solid #000



{border-collapse : collapse}

R1 C1	R1 C2	R1 C3
R2 C1	R2 C2	R2 C3
R3 C1	R3 C2	R3 C3
R4 C1	R4 C2	R4 C3

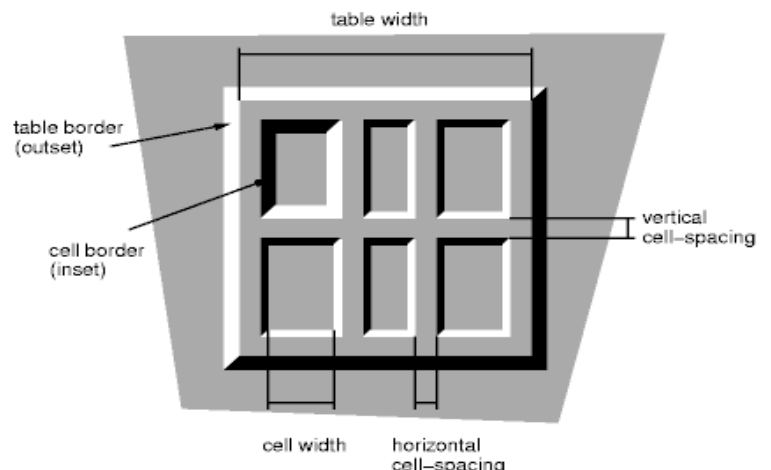
표(Table) 관련속성- 테두리 (Border) 속성

□ 분리된(Seperated) 테두리모델 속성들

- Border-spacing 속성
 - 셀(Cell)은 개별 테두리를 갖는데 border-spacing속성은 인접 셀들사이의 거리를 지정하는 속성.
 - 사용법 : { border-spacing : 수평길이(horizen length) 수직길이(vertical length) }
길이(length)값이 한 개일 경우 수평과 수직길이 : 길이값은 반드시 양수

```
TABLE {
  border: outset 10pt;
  border-collapse: separate;
  border-spacing: 15pt
}
TD { border: inset 5pt }
TD.special { border: inset 10pt } /* 위-왼쪽 셀 */
```

```
1
2 /* CSS에 cellspacing, cellpadding이 없으므로 */
3 Table {
4   border-spacing:0px;
5   padding:0px;
6   border-collapse : collapse;
7 }
8 TD, TH {padding:0px;}
```



표(Table) 관련속성- 테두리 (Border) 속성

[예제파일: 233 테두리 분리 empty cells]

□ Empty-cells 속성

- 분리된 테두리모델에서 빈 셀들과 visibility(가시적)속성을 지정하는 속성
 - 사용법 : { empty-cells : hidden | show }
- 속성값 'show' : 빈 셀의 주위에 정상셀 (내용을 가진셀)처럼 테두리가 그려짐.
* MS IE6.0 적용 안됨

able class="cl2">
r>
<td>[cl3]</td>
<td class="r1c2">r1
<td>r1c3</td>
tr>
<td class="r2c1">r2c1</td>
<td></td>
<td></td>
tr>
<td>r3c1</td>
<td>r3c2</td>
<td class="r3c3">r3c3</td>
tr>
table>

```

<style type="text/css">
table {background: gray; width: 300px; border-collapse: separate;}
table td {background: #FFF; color: black; border: 1px solid green;}
.cl1 {empty-cells: show;}
.cl2, .cl3 {empty-cells: hide;}
</style>

<table class="cl1">
<tr>
<td>[cl1]</td>
<td class="r1c2">r1c2</td>
<td>r1c3</td>
</tr>
<tr>
<td class="r2c1">r2c1</td>
<td></td>
<td>r2c3</td>
</tr>
<tr>
<td>r3c1</td>
<td>r3c2</td>
<td class="r3c3">r3c3</td>
</tr>
</table>

<table class="cl2">
<tr>
<td>[cl2]</td>
<td class="r1c2">r1c2</td>
<td>r1c3</td>
</tr>
<tr>
<td class="r2c1">r2c1</td>
<td></td>
<td>r2c3</td>
</tr>
<tr>
<td>r3c1</td>
<td>r3c2</td>
<td class="r3c3">r3c3</td>
</tr>
</table>

```

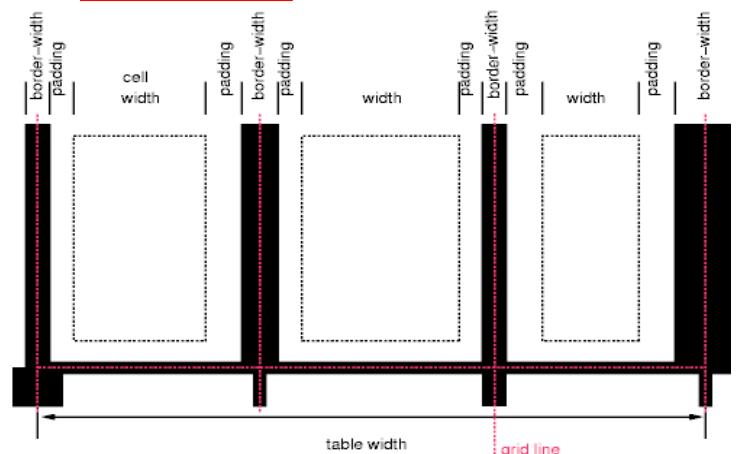
표(Table) 관련속성- 테두리 (Border) 속성

□ 통합된(Seperated) 테두리모델

- 통합테두리모델에서는 셀(cell), 줄(row), 줄그룹, 컬럼(column), 컬럼그룹의 테두리 지정하는것이 가능.: (HTML "rule" 속성이 통합된 테두리모델방식으로 지정됨)
- 테이블의 너비 구하는 공식 :

$$\text{row-width} = (0.5 * \text{border-width}_0) + \text{padding-left}_1 + \text{width}_1 + \text{padding-right}_1 + \text{border-width}_1 + \text{padding-left}_2 + \dots + \text{padding-right}_n + (0.5 * \text{border-width}_n)$$

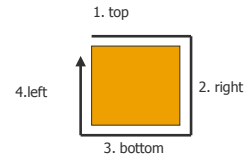
두 외부(exterior)테두리의 반만 테이블 너비로 계산되고,
다른 셀들의 양쪽 두 테두리들은 마진(margin)지역에 들어감.



표(Table) 관련속성- 테두리 (Border) 속성

□ 테두리너비(border-width) 속성

- 종류 : border-top-width, border-right-width, border-bottom-width, border-left-width, border-width(단축형속성)
- 속성값 : thin, medium, thick
- border-width(단축형 속성) 사용법:
 - 시계방향으로 설정 (맨위→오른쪽→바닥→왼쪽)
 - 값이 1개 : 모든 면에 적용
 - 값이 2개 : 1번째{top, bottom},2번째(right,left)
 - 값이 3개 : 1번째{top},2번째(right,left),3번째{bottom}
 - 값이 4개 : 1번째{top},2번째(right), 3번째(left),4번째{bottom}

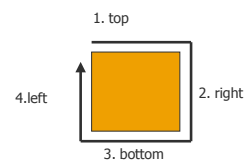


```
H1 { border-width: thin } /* thin thin thin thin */  
H1 { border-width: thin thick } /* thin thick thin thick */  
H1 { border-width: thin thick medium } /* thin thick medium thick */
```

표(Table) 관련속성- 테두리 (Border) 속성

□ 테두리색상(border-color) 속성

- 종류 : border-top-color, border-right-color, border-bottom-color, border-left-color, border-color(단축형속성)
- 속성값 : 색상값(#000000, rgb(255.0.0), red, transparent(투명색))
- border-color(단축형 속성) 사용법:
 - 시계방향으로 설정 (맨위→오른쪽→바닥→왼쪽)
 - 값이 1개 : 모든 면에 적용
 - 값이 2개 : 1번째{top, bottom},2번째(right,left)
 - 값이 3개 : 1번째{top},2번째(right,left),3번째{bottom}
 - 값이 4개 : 1번째{top},2번째(right), 3번째(left),4번째{bottom}
- 엘리먼트의 테두리 색상이 테두리속성에서 지정되지 않았으면 테두리 색상은 해당 엘리먼트의 'color'속성을 사용해야 함.



표(Table) 관련속성- 테두리 (Border) 속성

[예제파일: 233 테두리 스타일]

□ 테두리스타일(border-style) 속성 : 테두리의 형태를 지정

- 사용법 : {border-style: none | hidden | dotted}
- 속성값 :
 - None(없음) : 테두리 없음. (자신의 테두리만 억제)
 - Hidden(감춤) : 'none'과 같으나 해당 셀들의 모든 테두리를 억제
 - Dotted(점선)
 - Dashed(대쉬점선) : 짧은 선(shortline)
 - Solid(실선) : 단일선(line)
 - Double(이중실선) : 두개의 단일선
 - Goove(파인조각) : 테두리가 창(canvas)에 조각된것 처럼 보임. (음각)
 - Ridge(튀어나온 조각) : 양각
 - Inset(파일조각) : 분리된(sperated)테두리모델에서 창(canvas)에 깔린(embed)같이 전체 박스를 만듦. 통합테두리 모델에서 goove와 같음
 - Outset(튀어나온조각) : Inset의 반대
- border-color(단축형 속성) 사용법:
 - 시계방향으로 설정 (맨위→오른쪽→바닥→왼쪽)

표(Table) 관련속성- 테두리 (Border) 속성

□ 테두리약식표현 속성 { border: }

- 박스모델의 맨위→오른쪽→바닥→왼쪽 테두리의 너비→스타일→색상을 설정속성
- 종류 : border-top, border-right, border-bottom, border-left, border (단축형속성)
- 속성값 : border-width , border-style, border-color식으로 속성지정

#H1 { border-bottom: thick solid red } /*너비 스타일 색상*/

□ border (단축형 속성) 사용법:

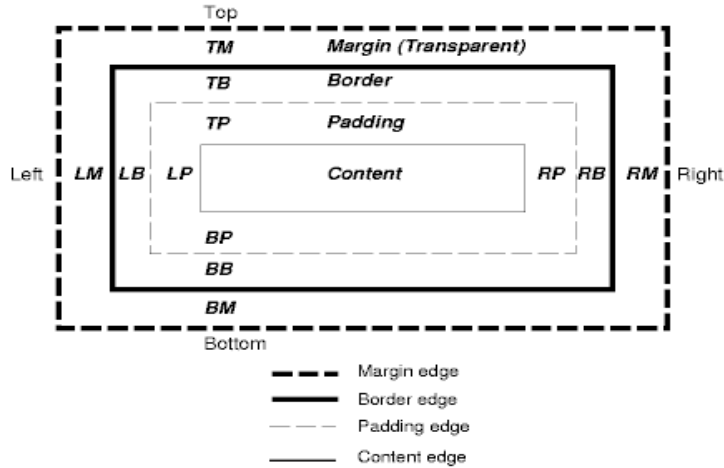
- 모든 네 테두리들에 같은 너비,스타일,색상을 설정하는 약식속성
- margin,padding의 약식속성과 달리 4테두리의 다른값을 설정 못함.

P { border: solid red } = P {
border-top: solid red;
border-right: solid red;
border-bottom: solid red;
border-left: solid red
}

박스모델(BOX MODEL)

□ 박스모델(BOX)개요

- CSS2.0 에서는 모든 element는 여러겹의 상자로 둘러쌓여 있다고 가정

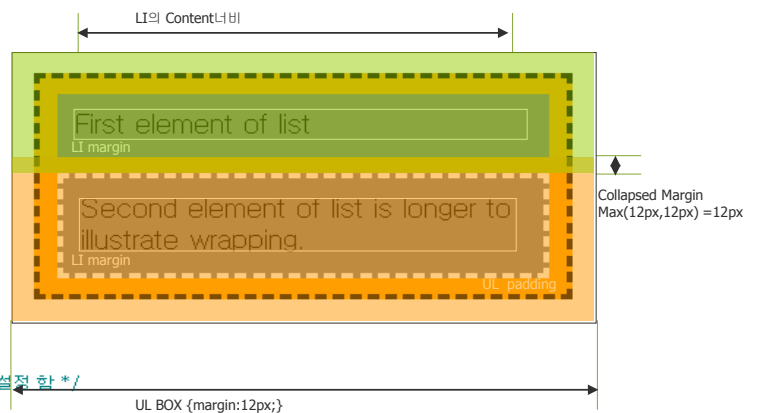


박스모델(BOX MODEL)

[예제파일: 234 박스모델_마진패딩테두리예제]

□ 마진(margin),패딩(padding),테두리(border)예제 1

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">
<HTML>
<HEAD>
<TITLE>마진, 패딩, 테두리들의 예제</TITLE>
<STYLE type="text/css">
  UL {
    background: orange;
    margin: 12px;
    border-style: dashed;
    border-width: medium;
    border-color: black;
    padding: 3px;
    /* 테두리들이 설정되지 않음 */
  }
  LI {
    color: black; /* 텍스트 색상은 검정 */
    background: gray; /* 내용, 패딩(padding)은 회색 */
    margin: 12px;
    padding: 12px; /* 오른쪽은 패딩 0px 임 */
    list-style: none /* 목록 항목 앞에 그림문자들이 없음 */
    /* 테두리들이 설정되지 않음 */
  }
  LI.withborder {
    border-style: dashed;
    border-width: medium; /* 모든 면들에 테두리 너비를 설정 함 */
    border-color: white;
  }
</STYLE>
</HEAD>
<BODY>
<UL>
<LI>First element of list
<LI class="withborder">Second element of list is longer
to illustrate wrapping.
</UL>
</BODY>
```

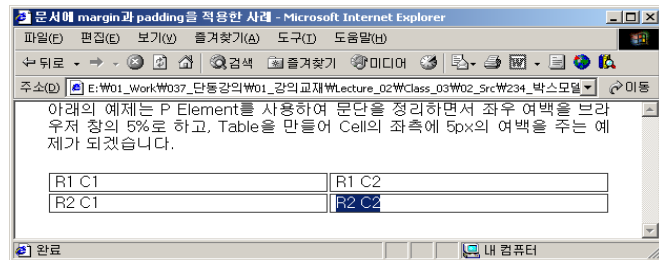


박스모델(BOX MODEL)

[예제파일: 234 박스모델_마진]

□ 마진(margin),패딩(padding),테두리(border)예제 2

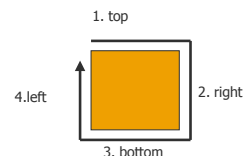
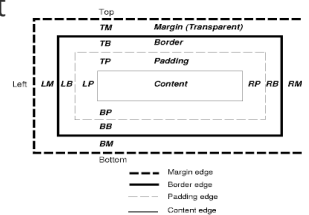
```
<HTML>
<HEAD>
  <TITLE>문서에 margin과 padding을 적용한 사례</TITLE>
  <STYLE type="text/css">
    <!--
    .horMargin{margin:0% 5%;}
    .cellPadding{padding-left:5px;border:1px solid #333333}
    .Para{text-align:justify;}
    -->
  </STYLE>
</HEAD>
<BODY class="horMargin">
  <P class="Para">
    아래의 예제는 P Element를 사용하여 문단을 정리하면서 좌우 여백을 브라우저 창의 5%로 하고,
    Table을 만들어 Cell의 좌측에 5px의 여백을 주는 예제가 되겠습니다.
  </P>
  <table border="0" cellpadding="0" cellspacing="2" width="100%">
    <tr>
      <td class="cellPadding">R1 C1</td>
      <td class="cellPadding">R1 C2</td>
    </tr>
    <tr>
      <td class="cellPadding">R2 C1</td>
      <td class="cellPadding">R2 C2</td>
    </tr>
  </table>
</BODY>
</HTML>
```



박스모델(BOX MODEL)

□ 마진(margin)속성 : box의 마진영역(margin area)너비를 지정

- 관련속성 : margin-top, margin-right, margin-bottom, margin-left
- 사용법 : { margin-top : 길이값(length value : 20px,-10px) }
- margin(단축형 속성) 사용법:
 - 시계방향으로 설정 (맨위→오른쪽→바닥→왼쪽)
 - 값이 1개 : 모든 면에 적용
 - 값이 2개 : 1번째(top, bottom),2번째(right,left)
 - 값이 3개 : 1번째(top),2번째(right,left),3번째(bottom)
 - 값이 4개 : 1번째(top),2번째(right), 3번째(left),4번째(bottom)



```
BODY { margin: 2em } /* 모든 마진들을 2em으로 설정 */
BODY { margin: 1em 2em } /* 맨위 & 바닥 = 1em, 오른쪽 & 왼쪽 = 2em */
BODY { margin: 1em 2em 3em } /* top=1em, right=2em, bottom=3em, left=2em */
```

```
BODY {
  margin-top: 1em;
  margin-right: 2em;
  margin-bottom: 3em;
  margin-left: 2em; /* 반대쪽(right)에서 복사 */
}
```

박스모델(BOX MODEL)

□ 마진(margin) 통합(collapsing)

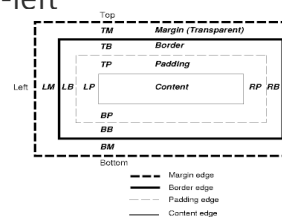
- CSS2.0 부터 인접된 두 개 이상의 박스들의 인접마진이 통합되어 단일마진을 형성
- 수평마진은 절대로 통합되지 않음
- 수직마진의 통합
 - 두개 이상의 인접 수직마진들은 통합(Table의 collapsing)
 - 유동된(floated)박스와 다른 박스의 수직 마진은 통합되지 않음
 - 위치값(position)이 absolute와 relative 로 위치된 박스의 마진들은 통합되지 않음

박스모델(BOX MODEL)

[예제파일: 234 박스모델 패딩]

□ 패딩(padding)속성 : box의 패딩영역(padding area)너비를 지정

- 관련속성 : padding-top, padding-right, padding-bottom, padding-left
padding(단축형속성)
- 사용법 : { padding-top : 길이값(length value : 20px(양수)) }
- padding(단축형 속성) 사용법:
 - 시계방향으로 설정 (맨위→오른쪽→바닥→왼쪽)
 - 값이 1개 : 모든 면에 적용
 - 값이 2개 : 1번째(top, bottom), 2번째(right, left)
 - 값이 3개 : 1번째(top), 2번째(right, left), 3번째(bottom)
 - 값이 4개 : 1번째(top), 2번째(right), 3번째(left), 4번째(bottom)

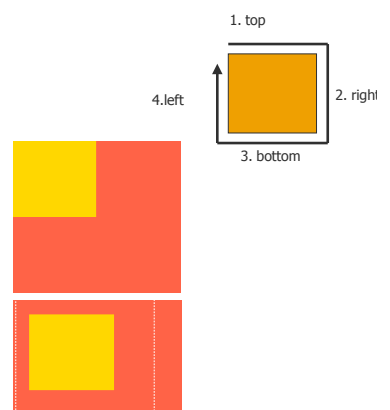


01. 패딩을 주지 않았을 경우

```
<DIV style="width:200px; height:200px; background-color:tomato;">
  <DIV style="width:100px; height:100px; background-color:gold;"></DIV>
</DIV>
```

02. 패딩을 줬을 경우

```
<DIV style="width:200px; height:200px; background-color:tomato;padding:20px;">
  <DIV style="width:100px; height:100px; background-color:gold;"></DIV>
</DIV>
```



포지셔닝(Positioning, 내용의 위치)

□ 포지셔닝(Positioning)

- **시각적인 측면의 HTML에 있어서 가장 중요한 요소**
- HTML문서안의 부분문서의 위치를 지정하거나 객체(Object)의 보임과 안보임(visiblity)을 다룸.
- 정적인 HTML을 Javascript를 이용하여 동적인(Dynamic)하게 만들기 위한 가장 기본적인 속성

□ 포지셔닝관련 속성(Positioning Property)종류

- width, height : Box의 width와 height를 지정
- position : static, relative, absolute 등의 위치를 지정
- top, right, bottom, left : 데이터를 포함하는 container box속에서 객체와 box border 사이의 간격을 지정
- overflow : 내용물이 box를 넘쳤을때 처리하는 방법을 지정
- float : 이미지나 container box와 문장과의 상대적인 위치를 지정
- clear : float로 지정된 문장의 흐름을 원상태로 만드는 속성
- clip : box의 일부분을 가려서 보이지 않게 하는 속성
- visibility : 객체(Object)를 보이거나 보이지 않게 지정하는 속성
- z-index : z좌표를 사용하여 레이어를 만드는 속성

포지셔닝(Positioning, 내용의 위치)-width,height

[예제파일: 0241포지셔닝 widthheight]

□ width, height속성(property)

- 컨테이너박스(Container Box)의 가로폭, 세로폭을 지정하는 속성
- 사용법 {width : 길이값(length: px, cm, em 등) | 백분율(%) | auto}
- 속성값 :
 - length(길이값) : px, pt, cm, mm, in 등의 길이단위
 - 백분율(%) : Block 자신이 담겨있는 상위Block의 width, height에 대한 백분율 단위
상위 Block의 크기가 바뀌면 자신의 크기도 자동으로 변경
 - auto (width의 경우) : **100%**, 자신의 상위 Block이 허용하는 width를 꽉 채우게 됨
 - auto (height의 경우) : **0%**, 이 경우 height를 결정하는 요인은 block box속의 내용물의 크기
(예 : text가 줄바꿈을 하면서 내용이 많아지면 자동으로 height가 내용에 맞게 늘어나거나 줄어듦)

포지셔닝(Positioning, 내용의 위치)-position

[예제파일: 0241포지셔닝_position]

□ position (위치)속성

- 동적인 웹페이지를 구성하기 위한 가장 핵심적인 속성중 하나.
- 문서에서의 위치를 결정:정적위치(static),절대위치(absolute),상대위치(relative)
- 사용법 : { position:static | relative | absolute | fixed}
- 속성값
 - static: 기본(default)값으로 HTML문서의 문단에서처럼 일반적인 내용물의 흐름 상단(top)과 좌측(left)에서의 거리를 지정할 수 없음.
 - relative : HTML문서에서의 일반적인 내용물의 흐름을 말하지만 top, left거리를 지정함.
 - absolute : 자신의 상위 box속에서의 top,left,right,bottom등의 절대적인 위치 변하지 않는 위치를 지정
 - fixed : 스크롤(scroll)이 일어나도 항상 화면상의 지정된 위치에 있음 (단 IE에서 작동하지 않음)

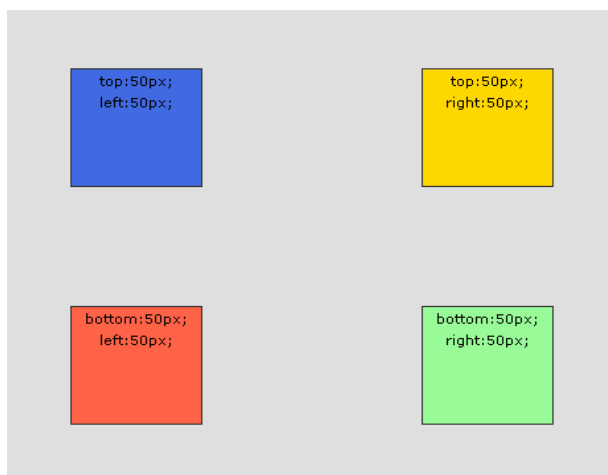
□ 포지션닝(positioning)을 할수 있는 조건

- top과 left를 지정할 수 있는 {position:relative},{position:absolute}의 두가지 경우
- 어떤 Object(DIV, SPAN,P)에 top과 left를 지정할수 있는 경우
 - 최상위 element인 BODY의 바로 한단계 하위 element인 경우(offset:0,0)
 - body가 한단계 상위가 아닐경우 자신의 한단계 상위 element에 포지셔닝이 되어 있는 경우 (offset:상위개체의 offset)

포지셔닝(Positioning, 내용의 위치)-top,left,bottom,right속성

□ top,left,bottom,right 속성

- 자신이 담겨있는 box속에서의 top,left,bottom,right에서의 거리를 지정하는 속성
- 사용법 : {top : 길이값(length:px,cm) | 백분율(%) | auto}



포지셔닝(Positioning, 내용의 위치)-overflow속성

[예제파일: 0241포지셔닝_overflow]

□ overflow 속성

- <DIV>element와 같은 box container속에서 내용물이 가로(X방향),세로(Y방향)으로 넘쳤을때 어떻게 처리하느냐를 다루는 속성
- 사용법 : {overflow : visible(기본값) | hidden | scroll | auto}
- 속성값 :
 - visible : box속의 내용물을 모두 보여줌. 내용물의 크기에 따라 box의 가로,세로폭이 늘어남
 - hidden : box의 width,height를 지정했을 경우, 지정된 범위를 넘치는 내용물은 보이지 않음.
 - auto : box의 width,height를 지정했을 경우, 내용물이 box의 크기를 넘치는 순간 scroll bar가 생겨서 범위를 넘는 내용물을 scroll하여 보여줌
 - scroll : box의 width,height를 지정했을 경우, 내용물이 box를 넘치지 않든 scroll bar의 면적을 차지하며, 내용물이 넘치는 순간 scroll bar가 생김

포지셔닝(Positioning, 내용의 위치)-float, clear속성

[예제파일: 0241포지셔닝_float]

□ float속성

- Block Box나 이미지를 좌측(left)나 우측(right)에 위치시키고 block box나 이미지 주변으로 text를 배열하는 속성 (IE와 다른브라우저가 조금씩 틀림 : float , style-float(IE))
- 사용법 : {float : left | right | none}
- 속성값 :
 - left : 그림이나 box가 왼쪽에 배치되고, 글씨는 box의 오른쪽으로 흐름
 - right : 그림이나 box가 오른쪽에 배치, 글씨는 box의 왼쪽으로 흐름
 - none : 그림이나 box가 왼쪽으로 배치되고, 글씨는 첫 줄만 box의 오른쪽으로 흐름.

□ clear 속성 : float속성과 관련

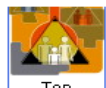
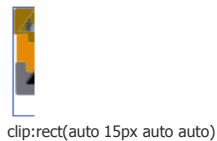
- float속성으로 위치한 이미지나 box주위의 text들이 이미지나 box주변으로 흐르게 하거나, 흐름을 멈추고 이미지나 box아래로 내려와서 줄을 바꿀것인지 지정하는 속성
- 사용법 : {clear : none | left | right | both}
- 속성값 :
 - none : 이미지나 box아래로 줄을 바꾸지 않고 주변으로 흐름.
 - left : text가 이미지나 box오른쪽으로 흐르지 않고 이미지나 box아래로 줄바꿈
 - right : text가 이미지나 box왼쪽으로 흐르지 않고 이미지나 box아래로 줄바꿈
 - both : text가 이미지나 box사이로 흐르지 않고 이미지나 box아래로 줄바꿈

포지셔닝(Positioning, 내용의 위치)-clip속성

[예제파일: 0241포지셔닝_대]

□ clip속성

- 4각형 block box를 지정하는 수치만큼 가려서 안보이도록 해주는 속성
- clip되는 box는 반드시 {position:absolute}로 지정되어 있어야 함.
- 사용법 : {float : auto | 잘리는형태(rect)}
- 속성값 :
 - auto : 초기값, 잘려서 가려지는 부분이 없이 모두 보여줌.
 - 잘리는형태 (rect : top right bottom left)
 - top : 지정한 길이만큼 위에서 아래로 가려서 가려줌.
 - right : 지정한 길이만큼 왼쪽에서 오른쪽으로 보여줌.
 - bottom : 지정한 길이만큼 위에서 아래로 보여줌.
 - left : 지정한 길이만큼 왼쪽에서 오른쪽으로 가려줌.



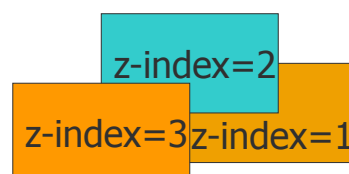
포지셔닝(Positioning, 내용의 위치)-visibility, z-index속성

□ visibility속성

- 객체(Object)를 보이게(visible)하거나 안보이게(hidden)하는 속성 {display:none}과 비슷한 효과
 - {display : none} 브라우저창의 면적을 차지하지 않음.
 - {visibility : hidden} 브라우저창의 면적을 객체(Object)의 면적만큼 차지하면서 보이지 않음
- 사용법 : {visibility : visible | hidden | collapse }
- 속성값
 - visible : 객체를 브라우저창에 보이게 함
 - hidden : 객체를 브라우저창에 보이지않게 하고, 면적을 차지
 - collapse : table의 border-collapse와 같은 효과 (*CSS2.0에 명시는 되어 있으나 작동안함)

□ z-index속성

- 레이어 기능(입체) x-index, y-index, z-index
- box(inline, block)위에 box를 얹는 기능
- 사용법 : {z-index : 정수값(양수,음수가능) }
- 기본값 : auto



가상클래스(Pseudo class, Pseudo element)

□ Psudeo class :

- HTML element나 attribute에 없는 속성을 만들어 주는 역할을 하는 CSS속성

□ 가상클래스(Pseudo class) 관련속성

- !important : 중요도를 선언, 문서에 지정된 모든 CSS속성들을 우선하는 지정.
- :first-letter : 문장에서 첫글자에 대한 지정
- :first-line : 문장에서 첫줄에 대한 지정
- :link, :visited, :hover, :active : Link걸린 글자(HyperText)에 관한 속성지정
- @Font-face Rule : Web Font사용시 Web font를 embodied 시키는 CSS Rule
- @Charset Rule : 문자세트를 지정하는 CSS Rule
- @Import : 확장자가 .css 인 외부 stylesheet file을 HTML file에 연결시켜서 사용할수 있게 하는 rule
- 이외의 많은 관련속성이 있으나 대부분 현재지원을 하지 않음.

가상클래스(!important)

[예제파일: 0242가상클래스 !important

□ !important :

- 중요도(!important Declaration)를 선언, 문서에 지정된 모든 CSS속성값보다 우선시 됨.
- inline style, id selector, class selector또한 !important로 선언한 속성값으로 적용
- 자주 적용되는 부분 : 인쇄시
- 사용법 : 속성값뒤에 공백뒤 사용, 또는 붙여사용

```
<HTML>
<HEAD>
<TITLE>!important 선언</TITLE>
<STYLE type="text/css">
<!--
*{
  color:#000000 !important;
  background-color: #FFFFFF !important;
  border:1px solid #808080 !important;
}
-->
</STYLE>
</HEAD>
<BODY>

<P style="color:red">
  Inline style로 글씨색을 red로 지정해도 style Block에서 !important 선언으로 검정색 글씨가 된다.
</P>

<DIV style="background-color:gold;border:1px solid red;width:100%;height:50px">
  Inline style로 배경색을 gold로 지정해도 style Block에서 !important 선언으로 흰색 배경색이 된다.
</DIV>

</BODY>
</HTML>
```

Inline style로 글씨색을 red로 지정해도 style Block에서 !important 선언으로 검정색 글씨가 된다.

Inline style로 배경색을 gold로 지정해도 style Block에서 !important 선언으로 흰색 배경색이 된다.

Inline style로 글씨색을 red로 지정해도 style Block에서 !important 선언으로 검정색 글씨가 된다.

Inline style로 배경색을 gold로 지정해도 style Block에서 !important 선언으로 흰색 배경색이 된다.

가상클래스(: first-letter, :first-line)


[예제파일: 0242가상클래스 firstlett


□ : first-letter

- HTML element가 지원하지 못하는 기능을 지원하는 class
- 문장의 첫글자의 속성을 지정

□ : first-line

- 문장의 첫줄(line)의 속성을 지정

 장 중에서 첫 번째 글자를 다루는 가상 element입니다. 첫 번째 글자에 가상 클래스를 적용해서 CSS Property 들을 지정합니다. 문장의 첫 번째 글자만 영향을 받고 나머지 글자들은 영향을 받지 않습니다.

 장 중에서 첫 번째 글자를 다루는 가상 element입니다. 첫 번째 글자에 가상 클래스를 적용해서 CSS Property 들을 지정합니다. 문장의 첫 번째 글자만 영향을 받고 나머지 글자들은 영향을 받지 않습니다.

```
<HTML>
<HEAD>
<TITLE>first-line pseudo class</TITLE>
<STYLE type="text/css">
<!--
p.double:first-line{font-weight:bold;}
p.double:first-letter {font:bold 300%; color:tomato;}
/* p Tag에만 적용되는 double이라는 클래스를 지정하고, :first-line 가상 element 사용*/
-->
</STYLE>
</HEAD>
<BODY>
<P class="double">
문장 중에서 첫 번째 글자를 다루는 가상 element입니다.
첫 번째 글자에 가상 클래스를 적용해서 CSS Property 들을 지정합니다.
문장의 첫 번째 글자만 영향을 받고 나머지 글자들은 영향을 받지 않습니다.
</P>
</BODY>
</HTML>
```

가상클래스(링크관련)

[예제파일: 0242가상클래스 링크관

□ Link관련 가상 rule

- <body>태그의 link(링크걸린 글자색), vlink(방문했던 링크의 글자색), alink(링크를 클릭했을때의 글자색)
- 단순히 글자의 색만 다루는것이 아니라 글자의 여러가지 속성을 줄 수 있음.
- 속성값
 - :link (방문하지 않은 링크에 대한 속성 지정)
 - :visited (방문했던 링크에 대한 속성 지정)
 - :hover (마우스를 올렸을때 나오는 글자에 대한 속성 지정)
 - :active (링크를 클릭했을때 글자에 대한 속성 지정)

```
/*<a href="링크주소">링크할곳</a> */
A:link{color:royalblue;}
A:visited{color:gray;}
A:hover{color:orange;text-decoration:underline;}
A:active{color:tomato;}
```

```
/*<a href="링크주소" class="new">링크할곳</a> */
A.new:link{color:red; font-weight:bold;}
A.new:visited{color:red; font-weight:bold;}
A.new:hover{color:orange;text-decoration:underline; font-weight:bold;}
A.new:active{color:tomato; font-weight:bold;}
```

가상클래스(@font-face)

□ @font-face rule

- Web에서 사용하는 Open Type font를 지정하는 규칙
- HTML문서에 font, font-family로 작성한 폰트가 실제 문서를 보는 사용자환경(User Agent)에 없는 글꼴을 사용할 경우 화면에 제대로 출력되지 않음.
- @font-face를 이용하여 서로 다른 사용자환경에서도 같은 글꼴로 볼수 있게 만들어 줌.
- 파일확장자 .eot(Embedding Open type)
- 사용법 : @font-face { font-family:폰트명; src:url('경로/폰트파일이름.eot')}
- 웹서버를 경유하지 않고는 작동하지 않음.

```
<STYLE TYPE="text/css">
@font-face {
  font-family: 나눔고딕;
  font-style: normal;
  font-weight: normal;
  src: url(nanumgothic.eot);
}
</STYLE>
```

사용방법 :
<P style="font-family:나눔고딕;">@font-face Rule</P>

나의 사랑은 어느 누구라 해도
움직일 수가 없습니다.
내사랑은 절대로 움직이지 못합니다.
왜냐하면 나는 당신에게만 뿌리를
박고 살 수 있는 한 그루의 나무니까요.

Microsoft Web프로그래밍 다운로드 : <http://www.microsoft.com/typography/web/embedding/weft3/default.htm?fname=%20&fsize>

가상클래스(@charset, @import)

□ @charset rule : 문서에서 사용되는 문자세트(character set)을 지정.

- <STYLE TYPE="text/css">@charset "euc-kr";</STYLE>

□ @import rule :

확장자가 .css 인 외부 stylesheet file을 HTML file에 연결시켜서 사용할수 있게 하는 rule

- link element의 href attribute를 사용하여 스타일시트를 연결하는것과 같음.

- <STYLE TYPE="text/css">
@import url("외부style sheet의 URL");
@import "외부Style sheet의 URL";
</STYLE>
- <link rel="stylesheet" type="text/css" href="외부stylesheet의 URL" />

- 사용시 주의사항

- style block이나 외부 stylesheet file의 가장 첫번째 줄에 @import Rule을 선언해야 함.
- 선언된 줄 끝에 세미콜론(;)을 반드시 찍어야 함.

인쇄(Printing) 관련 속성

- **page-break-after** : 지정하는 앞부분에서 page바꿈
 - 페이지넘김을 지정한 block-level element와 다음 block-level element사이를 경계로 페이지 넘김
- **page-break-before** : 지정하는 뒷 부분에서 page바꿈
 - 페이지넘김을 지정한 block-level element와 이전 block-level element사이를 경계로 페이지 넘김
- **사용법** : { **page-break-before** : auto | always | avoid | left | right }
- **속성값** :
 - **auto** : 기본값, 사용자의 프린터에 지정된 기본값으로 출력
 - **always** : page-break를 지정한 객체(Object) 이전(page-break-before)또는 이후(page-break-after)에서 항상 페이지를 넘김
 - **avoid** : 페이지를 넘김을 하지 못하게 함.
 - **left, right** : 서로 마주보는 페이지가 되도록 제본하기 위해서 한 두페이지를 넘김. (*always처럼 동작)

필터(Filter) 관련 속성

[예제파일: 0244필터관

- **Filter 속성**
 - 별도의 이미지 처리 없이 CSS Scripting만으로 훌륭한 그래픽(Graphic)효과를 낼 수 있는 속성들
 - CSS2.0에서는 없어진 기능 (IE만 적용됨)

CSS를 이용한 DHTML

□ CSS를 이용한 동적인(Dynamic) HTML

- HTML문서 : 화면에 표시하기 위한 정적인 문서
- DHTML : Dynamic HTML
 - 마우스, 키보드나 기타 다른 요소에 의해 일어나는 어떤상황(Event)를 javascript로 표현
 - 이벤트(Event)의 종류 : Mouse Event, Windows Event, Keyboard Event, Form Element Event 등

Mouse의 동작에 따른 Event의 구분		
onMouseOver		마우스가 지정된 영역안으로 들어가는 순간
onMouseOut		마우스가 지정된 영역안에서 빠져 나오는 순간
onMouseDown		마우스 버튼을 누르는 순간(왼쪽이나 오른쪽 모두)
onMouseUp		마우스 버튼을 누른 상태에서 버튼을 놓는 순간(왼쪽이나 오른쪽 모두)
onMouseMove		마우스가 영역에서 움직이기 시작하는 순간
onClick		mousedown + mouseup 즉 눌렀다 놓는 순간
onDoubleClick		onClick 을 연달아 두 번 하는 순간

Window Events		
Attribute	Value	설명
onload	Scripting	문서가 다 읽혀 졌을 때
onunload	Scripting	보고 있던 문서에서 빠져 나갈 때

Keyboard Events		
Attribute	Value	설명
onkeydown	Scripting	Key를 눌렀을 때
onkeypress	Scripting	Key를 눌렀다 떼을 때

CSS를 이용한 DHTML

□ DHTML을 하기 위한 사전지식

- DOM (Document Object Model) :
 - HTML element들을 객체(Object)화 해서 동적인 효과를 낼수 있는 방법을 제시하기 위해 제안됨.
 - 기존의 웹페이지 작성자들이 보다 쉬운 방법으로 DHTML이나 Javascript로 접근하게 해줌.
 - DOM의 Object : HTML element를 사용한 태그
- this : 객체(HTML Tag) 자신을 말함.
- name : 객체의 이름
 - HTML element의 속성(attribute)중 하나
 - Form, iframe, frame, IMG, MAP element가 가지고 있는 속성으로 Scripting(Javascript)에서 호출될 이름.
- id : name속성이 없는 HTML에서 scripting할때 호출될 이름
- className : 사용자가 CSS를 지정할때 붙여준 class Name ('N'은 반드시 대문자)

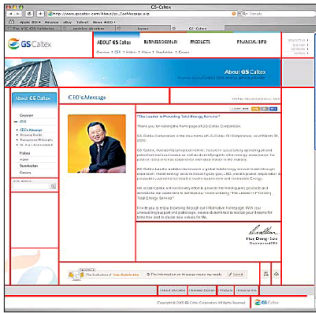
□ CSS Scripting을 이용한 DHTML

Object.style.styleProperty = stylevalue

↑
태그이름

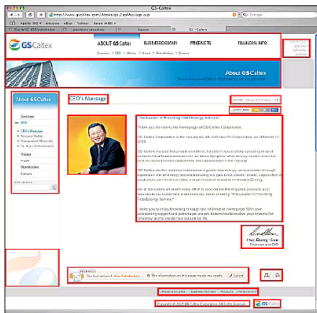
↑
CSS Scripting을 말하는 식별자(Identifier)

CSS 레이아웃(Layout)



□ 테이블(Table)레이아웃

- 격자(Grid)모양의 표(Table)
- 각 Cell의 종속성
- 기존 HTML의 경우 레이아웃을 모두 TABLE로 작성 한후Cell에 데이터를 넣는 방식
 - 전체적인 데이터량이 증가
 - 복잡해진 구조
 - 전송시간과 페이지 렌더링 시간 길어짐
 - 유지보수의 어려움.



□ CSS 레이아웃

- 그리드(Grid)가 아닌 구성요소의 집합(Set)
- 각 구성요소의 집합에 보다 쉽게 접근
- 디자인 변경이 용이
- Ajax Programming 용이

CSS 레이아웃(Layout)

□ 레이아웃에 사용되는 두가지 속성

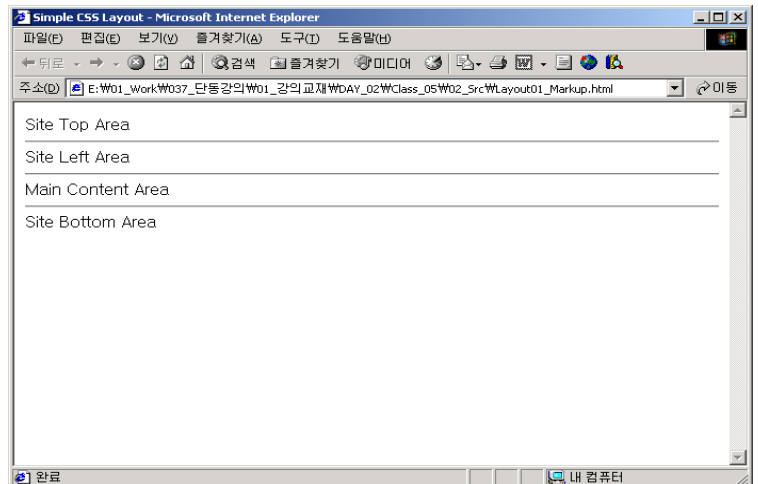
- position : 위치를 지정하여 원하는 위치에 엘리먼트를 배치
 - static : 기본값
 - relative : static과 같지만 offset(top='0', left='0') 지정.
하위 element의 offset의 기준점이 됨.
 - absolute : 화면상에서 다른 콘텐츠에 영향을 미치지 않고 위치지정 가능.
- float : 지정된 위치에서의 left, right 배치
 - left : 엘리먼트를 좌측으로 배치
 - right : 엘리먼트를 우측으로 배치
 - none : float시키지 않음

position을 이용한 기본 레이아웃(Layout) 01: 마크업(Mark up)작성

□ 마크업(Markup)작성 : 디자인이나 CSS를 고려하지 않고 제작.

- 실제 마크업(markup)은 div, id로만 구성.

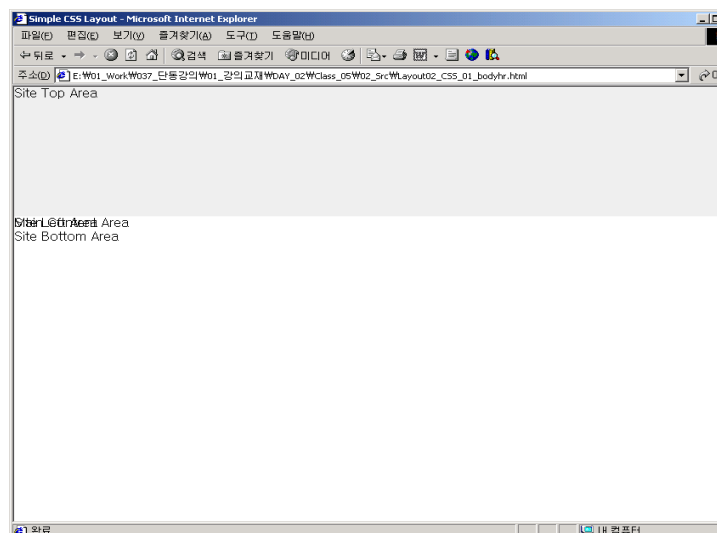
```
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
2 "http://www.w3.org/TR/html4/loose.dtd">
3 <html>
4 <head>
5 <meta http-equiv="Content-Type" content="text/html; charset=euc-kr">
6 <title>Simple CSS Layout</title>
7 </head>
8 <body>
9
10 <div id="head">Site Top Area</div>
11 <hr>
12 <div id="sub">Site Left Area</div>
13 <hr>
14 <div id="body">Main Content Area
15 </div>
16 <hr>
17 <div id="foot">Site Bottom Area</div>
18
19 </body>
20 </html>
```



position을 이용한 CSS 기본 레이아웃(Layout) 02: 스타일(CSS)작성

□ CSS로 디자인 적용

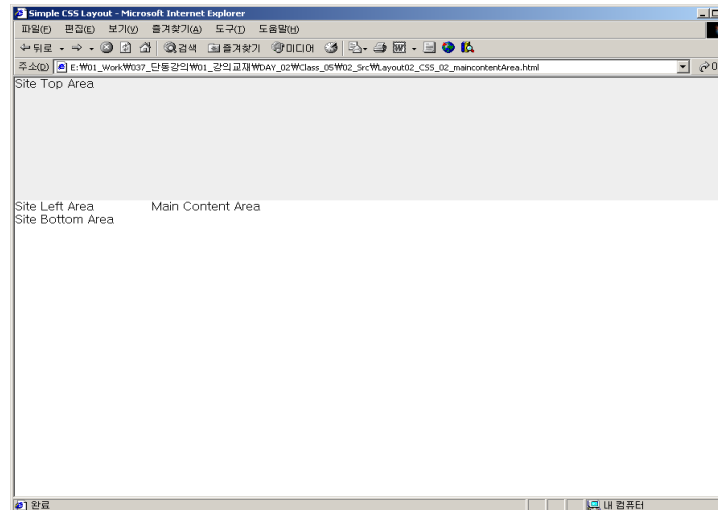
```
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
2 "http://www.w3.org/TR/html4/loose.dtd">
3 <html>
4 <head>
5 <meta http-equiv="Content-Type" content="text/html; charset=euc-kr">
6 <title>Simple CSS Layout</title>
7 <style type="text/css">
8 <!--
9 body {
10     margin: 0;
11     padding: 0;
12 }
13
14 hr { display:none; }
15
16 #head {
17     height: 170px;
18     background: #eee;
19 }
20
21 #sub {
22     position: absolute;
23     top: 170px;
24     left: 0;
25     width: 160px;
26 }
27
28 /-->
29 </style>
30 </head>
31 <body>
```



position을 이용한 CSS 기본 레이아웃(Layout) 02: 스타일(CSS)작성

□ CSS로 디자인 적용 : Main Content Area분리

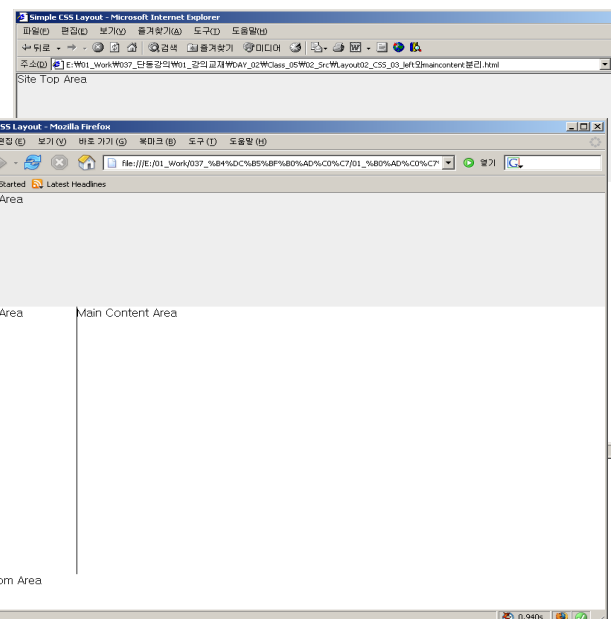
```
7 <style type="text/css">
8 <!--
9 body {
10 margin: 0;
11 padding: 0;
12 }
13
14 hr { display:none; }
15
16 #head {
17 height: 170px;
18 background: #eee;
19 }
20
21 #sub {
22 position: absolute;
23 top: 170px;
24 left: 0;
25 width: 160px;
26 }
27
28 #body {
29 padding-left: 170px;
30 width: 700px;
31 }
32
33 //-->
34 </style>
```



position을 이용한 CSS 기본 레이아웃(Layout) 02: 스타일(CSS)작성

□ CSS로 디자인 적용 : Main Content Area분리

```
7 <style type="text/css">
8 <!--
9 body {
10 margin: 0;
11 padding: 0;
12 }
13
14 hr { display:none; }
15
16 #head {
17 height: 170px;
18 background: #eee;
19 }
20
21 #sub {
22 position: absolute;
23 top: 170px;
24 left: 0;
25 width: 160px;
26 }
27
28 #body {
29 padding-left: 170px;
30 width: 700px;
31 }
32
33 //-->
34 </style>
```



```
28 #body {
29 padding-left: 170px;
30 width: 700px;
31 background: url('./img/body.gif') repeat-y 170px 0;
32 min-height: 400px;
33 }
```

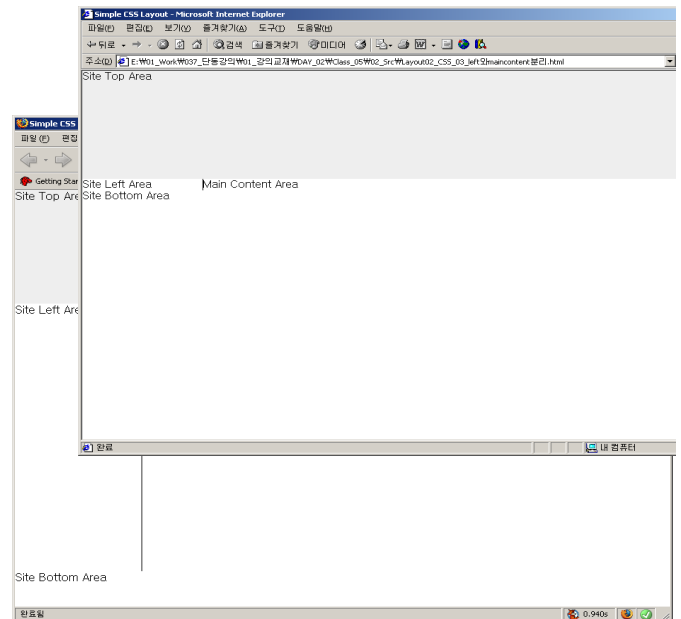
position을 이용한 CSS 기본 레이아웃(Layout) 02: 스타일(CSS)작성

□ CSS로 디자인 적용 : Main Content Area분리 (* IE-hack)

```
28 #body {
29     padding-left: 170px;
30     width: 700px;
31     background: url('./img/body.gif') repeat-y 170px 0;
32     min-height: 400px;
33 }

28 #body {
29     padding-left: 170px;
30     width: 700px;
31     background: url('./img/body.gif') repeat-y 170px 0;
32     min-height: 400px;
33 }

35 * html #body{
36     height:400px;
37 }
```



CSS Hack

□ CSS Hack

- 표준은 아니고 브라우저(Browser)들 간의 CSS해석 오류나 차이를 이용, 특정 브라우저만을 위한 CSS를 만드는 방법
- 일차적으로 표준을 준수하여 CSS작성 → 브라우저별로 테스트해보고 CSS를 해석(Render)하는 차이를 각 브라우저별로 hack을 사용하여 없앴
- Microsoft Internet Explorer용 CSS hack (IE-hack)사용법 : “ *** html** Element ”

```
div.ie-hack {
    width: 100px;
    padding: 20px;
}
* html div.ie-hack {
    width: 140px; /* IE width */
}
```

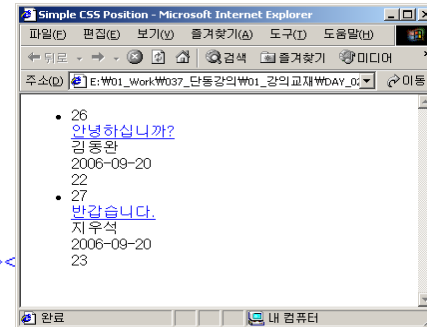
position의 offset(기준점) - 게시판 목록형태 - 연습과제

502_position의offset설

□ position:absolute 의 기준점(offset) : top,right,bottom,left 의 속성으로 offset 지정

- absolute position 블록의 상위에 아무것도 없을때 : offset이 브라우저 좌상단
- absolute position 블록의 상위에 relative position블록이 있을때 : offset은 상위 relative position블록의 좌상단

```
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
2 "http://www.w3.org/TR/html4/loose.dtd">
3 <html>
4 <head>
5 <meta http-equiv="Content-Type" content="text/html; charset=euc-kr">
6 <title>Simple CSS Position</title>
7 </head>
8 <body>
9
10 <div id="board-list" class="freeboard-item">
11 <ul id="board-list-item">
12 <li>
13 <div class="number">26</div>
14 <div class="title"><a href="view.jsp?articleSeq=271">안녕하십니까? </a></div>
15 <div class="name">김동환</div>
16 <div class="date">2006-09-20</div>
17 <div class="hits">22</div>
18 </li>
19 <li>
20 <div class="number">27</div>
21 <div class="title"><a href="view.jsp?articleSeq=271">반갑습니다.</a></div>
22 <div class="name">지우석</div>
23 <div class="date">2006-09-20</div>
24 <div class="hits">23</div>
25 </li>
26 </ul>
27 </div>
28
29 </body>
```

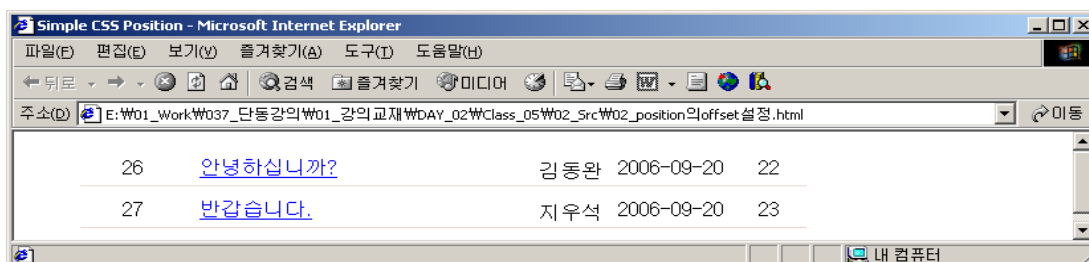


position의 offset(기준점) - 게시판 목록형태 - 연습과제

502_position의offset설

```
8 #board-list-item li {
9     position: relative;
10    width: 548px;
11    border-bottom: 1px solid #EBDDD4;
12    color: inherit;
13    list-style: none;
14 }
15 #board-list li div.number,
16 #board-list li div.date,
17 #board-list li div.hits { top: 7px; }
18 #board-list li div.title { padding-top: 7px; padding-bottom: 5px; }
19 #board-list div.number {
20     position: absolute;
21     left: 0;
22     width: 79px;
23     text-align: center;
24 }
25 #board-list div.title {
26     margin-left: 90px;
27     width: 297px;
28 }
```

```
29 #board-list div.date {
30     position: absolute;
31     right: 57px;
32     width: 92px;
33     text-align: center;
34 }
35 #board-list div.hits {
36     position: absolute;
37     right: 0;
38     width: 57px;
39     text-align: center;
40 }
41 #board-list div.name {
42     position: absolute;
43     top: 10px;
44     right: 150px;
45     width: 55px;
46     height: 1.5em;
47     text-align: center;
48     overflow: hidden;
49 }
```



float을 이용한 레이아웃 - 3단 컬럼형 레이아웃 - 연습과제

□ Markup작성

```
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
2 "http://www.w3.org/TR/html4/loose.dtd">
3 <html>
4 <head>
5 <meta http-equiv="Content-Type" content="text/html; charset=euc-kr">
6 <title>float을 이용한 컬럼형 레이아웃</title>
7 </head>
8 <body>
9 <div id="wrapper">
10 <div id="head">Site Top Area</div>
11 <hr>
12 <div id="sub">Site Left Area</div>
13 <hr>
14 <div id="body">Main Content Area</div>
15 <hr>
16 <div id="sidebar">Side Bar</div>
17 <hr>
18 <div id="foot">Site Bottom Area</div>
19 </div>
20 </body>
21 </html>
```



float을 이용한 레이아웃 - 3단 컬럼형 레이아웃 - 연습과제

503 float을 이용한 컬럼형 레이아웃

□ CSS 작성 : float속성을 이용하여 레이아웃 작성할때

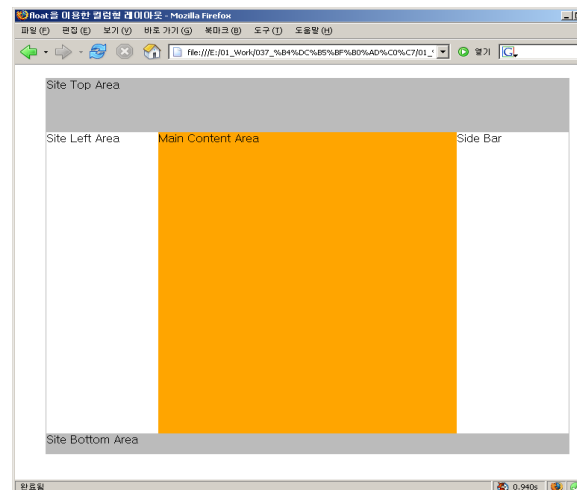
- 단지 폭을 정해주고 좌측, 우측정렬을 해주면 하나의 블록이 하나의 컬럼을 형성

```
8 <style type="text/css">
9 #wrapper {
10 width: 700px;
11 border: 1px solid #bbb;
12 margin: 20px auto;
13 }
14 #head {
15 height: 80px;
16 background: #bbb;
17 }
18 #foot {
19 height: 30px;
20 background: #bbb;
21 }
22 #sub,
23 #body,
24 #sidebar {
25 float: left;
26 }
27 #sub,
28 #sidebar {
29 width: 150px;
30 }
31 #body {
32 width: 400px; /* 700px - 150px - 150px */
33 height: 450px;
34 background: orange;
35 }
36 #foot {
37 clear: both;
38 }
39 hr {display:none}
40 </style>
```

컬럼전체의 폭을 고정하기 위해서 바깥쪽을 감싸는 블럭

block-level element들이 연속적으로 있어 float시켜주면 한 줄(line)에 순차적으로 배치

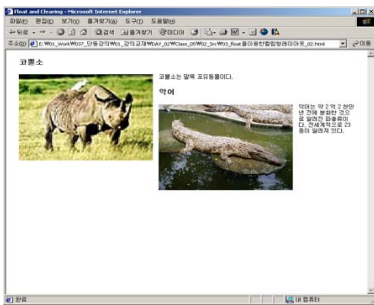
float이 끝난 위치 바로 다음에 오는 엘리먼트에서는 clear시켜주어 전체모양이 일그러지지 않게 함



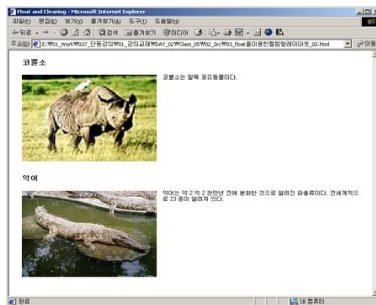
float와 clear

503 float을 이용한 컬럼 레이아웃

- float속성 : 하위블록에 영향을 끼치므로 clear속성과 같이 사용해야 함.



clear 속성 없을때



h2 { clear : both }



p { clear : both }

```

7 <style type="text/css">
8 body {
9     margin: 0;
10    padding: 0;
11 }
12 #wrapper {
13     width: 700px;
14     margin: 20px auto;
15 }
16 img {
17     width: 280px;
18     height: 200px;
19     float: left;
20     margin: 0 1em 1em 0;
21 }
22 h2 { clear : both }
23 </style>

```

```

26 <div id="wrapper">
27     <h2>코볼소</h2>
28     
29     <p>코볼소는 말목 포유동물이다.</p>
30
31     <h2>악어</h2>
32     
33     <p>악어는 약 2억 2천만년 전에 분화한 것으로 알려진 파충류이다. 전세계적으로 23종이 알려져 있다.</p>
34 </div>

```

float의 숨겨진 기능

503 float을 이용했을때overflow

- float된

```

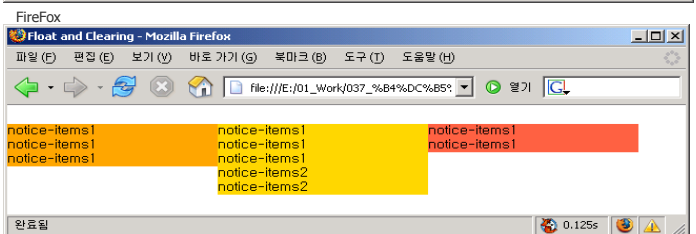
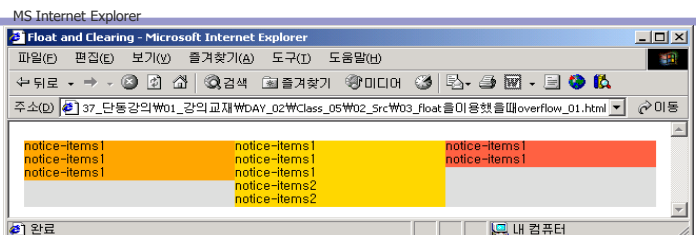
29 <div id="articles">
30     <ul id="notice">
31     </ul>
32     <div id="notice-items">notice-items1</div>
33     <div id="notice-items">notice-items1</div>
34     <div id="notice-items">notice-items1</div>
35
36     </li>
37 </ul>
38 <ul id="news">
39 </ul>
40     <div id="news-items">notice-items1</div>
41     <div id="news-items">notice-items1</div>
42     <div id="news-items">notice-items1</div>
43
44     </li>
45     <div id="notice-items">notice-items2</div>
46     <div id="notice-items">notice-items2</div>
47
48 </ul>
49 <ul id="stats">
50 </ul>
51     <div id="stats-items">notice-items1</div>
52     <div id="stats-items">notice-items1</div>
53
54 </ul>
55 </div>

```

[실제 웹사이트 개발시 많이 접하는 문제]

float된 의 상위블록인 #articles에 Background속성으로 색을 지정해주면 바탕색이 나타나지 않는 문제 (* MS IE 제외)

해결책 { overflow:auto; }



```

8 body {
9     margin: 0;
10    padding: 0;
11 }
12 #articles {
13     width:600px;
14     margin:20px auto;
15     background:#ddd;
16     overflow:auto; /* 지정하지 않으면 바탕색이 나오지 않음 */
17 }
18 ul { float:left; width:200px; margin:0;padding:0; }
19 li { list-style-type:none; }
20 #notice { background:orange; }
21 #news { background:gold; }
22 #stats { background:tomato; }

```

목록(List)

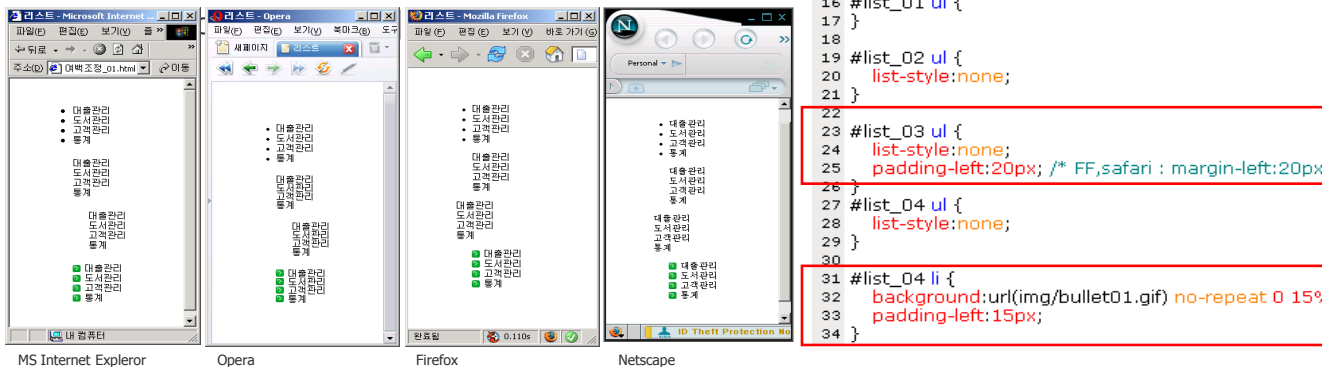
504 List를사용한원쪽여백조정

□ 3가지 목록(List) :

- ul (unordered list) : 순서가 없는 리스트
- ol (ordered list) : 순서가 있는 리스트
- dl (definition list) : term (<dt>), definition(<dd>) 쌍으로 이루어진 리스트

□ List element의 왼쪽 마진 특성 :

- 왼쪽여백의 조정이 각 브라우저마다 다르게 구현되어 있음.
 - Firefox, Safari : margin 이용.
 - Internet Explorer, Opera : padding 이용



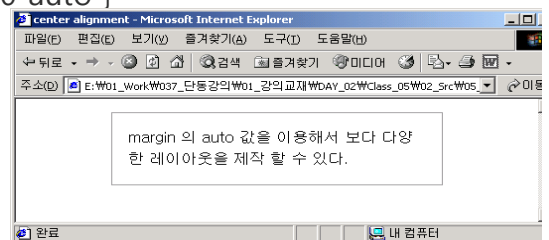
박스모델(Box Model) – 중앙정렬(aligned=center) - 연습과제

□ 웹사이트가 화면의 중앙에 정렬되어야 하는 레이아웃

- 기존 HTML : <table align="center">, <center>
- CSS에서의 사용법 : margin속성 이용 { margin : 0 auto }

```

1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
2 "http://www.w3.org/TR/html4/loose.dtd">
3 <html>
4 <head>
5 <meta http-equiv="Content-Type" content="text/html; charset=euc-kr">
6 <title>center alignment</title>
7 <style type="text/css">
8 #content {
9     width: 300px;
10    padding: 1em;
11    border: 1px solid #999;
12    margin: 0 auto;
13    line-height: 1.5em;
14 }
15 </style>
16 </head>
17 <body>
18 <div id="content">
19 <p>margin의 auto 값을 이용해서 보다 다양한 레이아웃을 제작 할 수 있다.</p>
20 </div>
21 </body>
22 </html>
    
```



[MS Internet Explorer 5.0 위한 CSS Tuning]

```

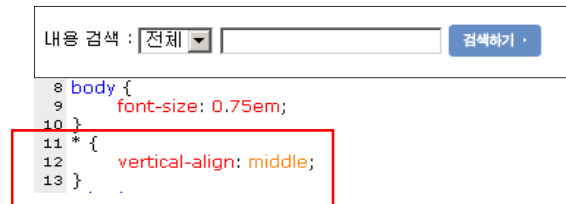
8 body { /* MS IES Tuning */
9     text-align: center;
10 }
11 #content {
12     width: 300px;
13     padding: 1em;
14     border: 1px solid #999;
15     margin: 0 auto;
16     line-height: 1.5em;
17     text-align:left;
18 }
19 </style>
    
```

body에서 상속된 정렬을 원상대로 복귀시킴 →

박스모델(Box Model) – 화면정중앙정렬(align=center valign=middle)

□ 웹사이트가 화면의 중앙에 정렬되어야 하는 레이아웃

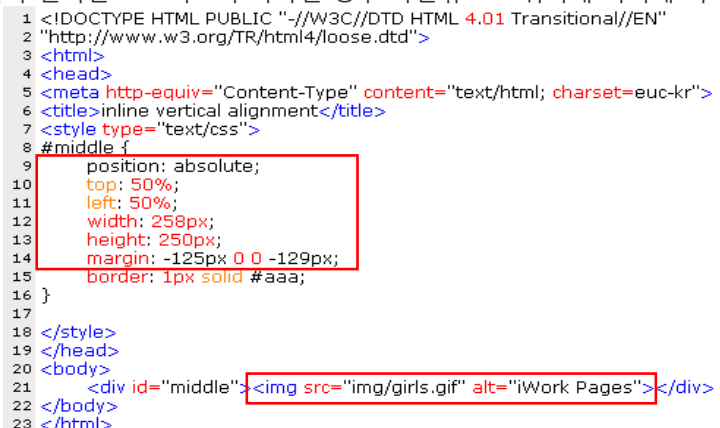
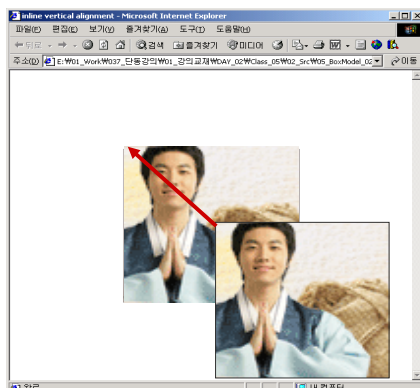
- 기존 HTML사용법 :
 - inline엘리먼트의 경우 : valign 속성이용
 - 그외의 경우 : <table width="100%" height="100%"><tr><td valign="middle" align="center">
- CSS에서의 사용법 :
 - inline 엘리먼트일 경우 : vertical-속성 이용 { margin : 0 auto }
 - 그외의 경우(inline + block 또는 block) : position속성과 margin음수값(negative)



박스모델(Box Model) – 화면정중앙정렬(position과 negative margin) - 연습과제

□ 웹사이트가 화면의 중앙에 정렬되어야 하는 레이아웃

- position속성과 margin(음수값) 사용법
 - 화면중앙에 위치 : position:absolute
 - offset을 50% : 엘리먼트의 좌상단이 화면의 정중앙에 오게됨
 - 블록의 크기의 절반만큼 margin으로 적용: 엘리먼트의 width,height의 반만큼씩 좌상단 이동
- 단점 : 음수값 사용으로 브라우저크기가 엘리먼트 크기보다 작아질 경우 화면밖으로 위치해 버리게 되어 엘리먼트 일부분이 보이지 않게 됨

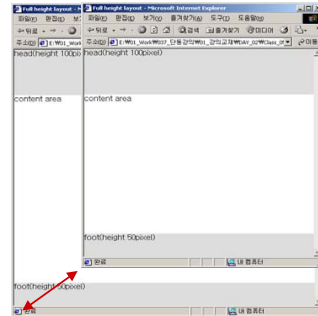


박스모델(Box Model) – 100% 높이를 유지하는 레이아웃

□ height, min-height의 사용:

<html> <body> 엘리먼트 하위에 100% 높이를 유지하는 <div> 엘리먼트 사용

```
<!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">
<head><title>Full height layout</title>
<style type="text/css">
html, body {
    height: 100%;
    margin: 0; padding: 0;
}
#head {height: 100px; background: #ddd; position: relative; }
#body { min-height: 100%; margin: -100px 0 -50px; }
#content-area { padding: 100px 0 50px; height: 300px; }
#foot { height: 50px; background: #ddd;}
</style>
<!--[if IE]>
<style type="text/css">
#body {
height: 100%;
}
</style>
<![endif]-->
</head>
<body>
<div id="head">head(height 100pixel)</div>
<div id="body"><div id="content-area">content area</div></div>
<div id="foot">foot(height 50pixel)</div>
```



박스모델(Box Model) – 테이블

□ 테이블의 CSS적용

- cellpadding="0" 제어방법 : <th> <td> 엘리먼트의 padding으로 제어
- cellspacing="0" 제어방법 : border-collapse

```
<table cellpadding="0" cellspacing="0" border="0">
<tr><td>example</td></tr>
</table>
```



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

- table-layout : fixed로 width계산과정을 줄여줌.

CSS 적용의 체크포인트

□ CSS를 이용해서 페이지에 디자인을 적용하기 전에 반드시 지켜야 하는 사항

- (X)HTML이 표준문법이어야 함.
 - 실제 디자인을 입히기 전에 먼저 작성한 Markup과 CSS의 문법을 조사한다.
W3C의 Markup Validation Service (<http://validator.w3.org>)
- (X)HTML 문서가 의미와 구조적으로 구성되어야 함.
 - 예)문서 전체의 내용을 표시할때는 <title>을 사용
- CSS가 표준문법이어야 함.
 - 예)주석문의 사용 : /* comment */ 만 사용
- CSS를 표준대로 잘 구현한 브라우저를 이용해야 함.
 - 개발시 : 표준을 잘 준수하는 브라우저를 이용
 - 개발후 : 가장 많이 사용하는 브라우저에 대한 튜닝을 진행

CSS를 효과적으로 사용하는 방법

□ 계획수립

- 웹사이트를 위한 CSS 스타일 관리계획.
- 기존 CSS와의 충돌 위험요소 예방

□ CSS스타일의 조직화

- CSS파일을 여러 섹션(section)으로 분리하여 관리.
- 문서의 재정의에 편리함

□ Cascade와 Inherit의 적절한 이용

- CSS의 특성중 하나인 계층화(cascading)를 이용한 스타일의 상속을 최대한 활용, 중복 사용 방지

□ 클래스의 남용 금지

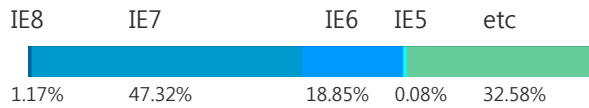
- 한 개의 HTML tag attribute에 여러 개의 클래스를 선언할수 있음.

□ 표준코드 사용

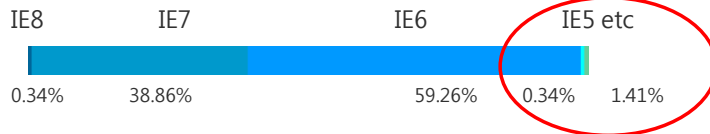
- 각 웹브라우저(Web Browser)에 따라 적용이 되지 않은 속성(property)이 있음.
Microsoft IE의 경우 가장 표준을 지키지 않음.

Internet Explorer 대처법

□ 지구촌 IE 버전별 점유율 현황



□ 국내 IE 버전별 점유율 현황 (2009년 2월 Internet Trend보고서 기준)



0.1%의 브라우저 지원은 현실적인 대안점은?
0.1%의 장애인은?

□ 따라서, IE6,7,8 모두 대응할 필요가 있다.

Internet Explorer 대처법

□ 브라우저측

- `<meta http-equiv="X-UA-Compatible" content="IE=7" />`
- `<meta http-equiv="X-UA-Compatible" content="IE=EmulateIE7 " />`

□ IIS Meta Switch

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <system.webServer>
    <httpProtocol>
      <customHeaders>
        <clear />
        <add name="X-UA-Compatible" value="IE=EmulateIE7" />
      </customHeaders>
    </httpProtocol>
  </system.webServer>
</configuration>
```

□ Apache Meta Switch

X-UA-Compatible:IE=EmulateIE7

Internet Explorer 대처법

□ UI개발자의 경우 CSS 를 별도 관리한다.

- default.css
ie7.css
ie6.css
ie5.css

□ 주석문으로 CSS 별도처리

- ```
<link rel="stylesheet" type="text/css" href="default.css" />
<!--[if IE 7]>
<link rel="stylesheet" type="text/css" href="ie7.css" />
<[endif]-->
<!--[if IE 6]>
<link rel="stylesheet" type="text/css" href="ie6.css" />
<[endif]-->
<!--[if IE 5]>
<link rel="stylesheet" type="text/css" href="ie5.css" />
<[endif]-->
```