

CSS2.0

(Cascading Style Sheet)

Park Yong Joon

(RedPlus)

<http://www.dotnetkorea.com/>

CSS Level 2 개요

- CSS란 'Cascading Style Sheets'를 말한다.
- HTML의 한계성을 극복하기 위한 방안의 하나 → DHTML
- 스타일시트를 사용하면 TABLE이나 FRAME에 의존하던 문서의 레이아웃을 효과적으로 할 수 있고, 글자 크기나 글자체, 줄 간격, 배경 색상 등도 자유롭게 선택할 수 있다. 또한, 스타일을 정의해 놓고 여러 개의 문서에서 불러 쓸 수 있어, 일관된 외양을 요구하는 여러 개의 문서를 만들 때 매우 편리하다.
- HTML과 CSS, Javascript, DOM 등을 함께 일컬어 Dynamic HTML 혹은 HTML4.0이라고 부르기도 한다.

미리보는 CSS 문법

■ HTML문서에 CSS적용하는 5가지 모습

형식:

```
<style type="text/css"> ~CSS코드~ </style>
```

```
<link
```

```
    rel="stylesheet"
```

```
    type="text/css"
```

```
    href="~CSS파일경로~"
```

```
>
```

```
<태그명 style="~CSS코드~"> ~ </태그명>
```

```
<태그명 class=".으로선언된스타일명"> ~ </태그명>
```

```
<태그명 id="#으로선언된스타일명"> ~ </태그명>
```

미리보는 CSS 문법

■ CSS 적용하는 방법

형식:

태그명 { 스타일속성명: 스타일속성값; ... }

태그명, ...

{

스타일속성명: 스타일속성값; ...

}

. 나만의스타일명 {}

#나만의스타일명 {}

HTML문서에 스타일시트 적용하기 시작

■ 처음으로 작성하는 스타일시트(아래 예제 타이핑)

```
<HTML>
<HEAD>
  <TITLE> CSS2.0 시작 </TITLE>
  <style type="text/css">
    /* CSS를 배워보자. */
    body {color: blue; }
    b {color: red; font-size: 9pt; }
  </style>
</HEAD>
<BODY>
안녕하세요... <br>
<b>CSS2.0</b><br>
반갑습니다. <br>
</BODY>
</HTML>
```

스타일시트 삽입하기 1(<head>내에 선언)

■ <STYLE type="text/css"> ~ </STYLE>

```
<HTML>
<HEAD>
  <TITLE> CSS2.0 시작 </TITLE>
  <style type="text/css">
    body {color:blue;}
    b, p, body
    {
      color:red;font-size:9pt;
    }
  </style>
</HEAD>
<BODY>
  안녕하세요... <br>
  <b>CSS2.0</b><br>
  반갑습니다. <br>
  <p>HTML의 보완적인 요소 : CSS</p>
</BODY>
</HTML>
```

스타일시트 삽입하기 2(외부파일에 스타일시트 선언)

■ <LINK rel="stylesheet" href="URL" type="text/css">

```
<HTML>
<HEAD>
  <TITLE> CSS2.0 시작 </TITLE>
  <link rel="stylesheet" type="text/css"
    href=". /3. 외부에스타일정의.css">
</HEAD>
<BODY>
안녕하세요... <br>
<b>CSS2.0</b><br>
반갑습니다. <br>
<p>HTML의 보완적인 요소 : CSS</p>
</BODY>
</HTML>
```

```
body {color: blue;}
b, p, body
{
  color: red; font-size: 9pt;
}
```

스타일시트 삽입하기 3(태그안에 스타일 선언)

■ <태그명 style="스타일"> : 인라인 스타일시트

```
<HTML>
<HEAD>
  <TITLE> CSS2.0 시작 </TITLE>
</HEAD>
<BODY style="color: blue; color: red; font-size: 9pt; ">
  안녕하세요. . . <br>
  <b style="color: red; font-size: 9pt; ">CSS2.0</b><br>
  반갑습니다. <br>
  <p style="color: red; font-size: 9pt; ">HTML의 보완적인 요소 : CSS</p>
</BODY>
</HTML>
```


사용자 정의 스타일 시트

■ <태그명 style="스타일"> : 인라인 스타일시트

```
<HTML>
<HEAD>
  <TITLE> CSS2.0 시작 </TITLE>
  <style type="text/css">
    body {color:red;}
    .myFontStyle{
      color: navy;      /*폰트 색상*/
      font-size: 9pt;   /*폰트 크기*/
      font-family: verdana;
      /*폰트 모양*/
    }
    #myBodyStyle{
      text-decoration: underline;
      /*언더라인*/
    }
  </style>
</HEAD>
```

```
<BODY class="myFontStyle">
안녕하세요. . . <br>
<b id="myBodyStyle">CSS2.0</b><br>
반갑습니다. <br>
<p style="color:red;">HTML의 보완적
인 요소 : CSS</p>
</BODY>
</HTML>
```

스타일 우선순위

■ CSS우선순위 (STYLE > ID > CLASS > TAG > Tag고유속성)

```
<HTML>
<HEAD>
  <TITLE> CSS우선순위 (STYLE>ID>CLASS>TAG>Tag고유속성) </TITLE>
  <style type="text/css">
    p {color: black; font-size: 20pt; text-decoration: overli ne; }
    .myFontStyle1{
      color: yel low; font-size: 30pt; font-fami l y: "굴림";
    }
    #myFontStyle2{
      font-size: 40pt; font-fami l y: @굴림;
    }
  </style>
</HEAD>
<BODY>
<p class="myFontStyle1" id="myFontStyle2" style="font-fami l y: ' 궁서체' ; ">안
  녕하세요. </p>
</BODY>
</HTML>
```

Selector 묶어서 사용하기

```
H1 {font-family: Helvetica}
```

```
H2 {font-family: Helvetica}
```

```
H3 {font-family: Helvetica}
```

```
H1, H2, H3 {font-family: Helvetica}
```

```
H1 {font-weight: bold; font-size: 12pt; line-height: 14pt;  
    font-family: Helvetica; font-variant: normal; font-style: normal; }
```

미리 정의된 스타일 사용

- #ID명 { 스타일 } → <태그명 id="ID명">
- .class명 { 스타일 } → <태그명 class="class명">

Class attribute selector

<STYLE TYPE="TEXT/CSS">

. KEYWORD { 속성:VALUE;...}

. KEYWORD { 속성:VALUE;...}

. KEYWORD { 속성:VALUE;...}

</STYLE>

<TAG CLASS="KEYWORD">

<TAG CLASS="KEYWORD">

<TAG CLASS="KEYWORD">

.test color : blue

H1.red color : red

<STRONG class = test>파란색

<EM class = pastoral>마찬가지

<H1 class = red>빨강색</H1>

<H1>검은색</H1>

Id attribute selector

Id 속성은 class와 비슷하지만, 한 문서에서 같은 id는 하나만 존재해야 한다는 것이 다르다.

Id attribute selector를 쓸 때는 #표시를 사용한다.

```
H1#test {letter-spacing: 5em }
```

```
<H1 id=test>Wide text</H1>
```

Contextual selectors

H1 안에 들어있는 EM element에만 파란색 글자를 주고 싶다"와 같이 특정 element에 포함된 element에만 스타일을 적용시키고자 할 때 contextual selector를 사용한다. Element와 Element 사이에 빈 칸을 주고 나란히 써 주면 된다.

H1 EM {color : blue }

<H1>검은색</H1>

검은색

<H1>검은색 푸른색 다시 검은색</H1>

Parent-child selectors

Parent-child selector는 하나의 element가 다른 element에 포함되었을 때 스타일을 정의하도록 해준다. Contextual selector와 비슷한데, contextual selector가 2대나 3대 자손이 되어도 되는 반면, parent-child selector는 이름에 나타나듯이 포함되는 첫 번째 element에만 해당된다.

Parent-child selector를 나타낼 때는 ~ 표시를 해 준다.

BODY font-size : 10pt ; line-height : 12pt

BODY ~ P line-height: 16pt

완벽하게 구현되지 않음

<P>여기는 줄 간격이 16pt이다.
</P>

<P>여기도 줄 간격이 12pt이다.
</P>

링크 스타일 지정

- A: link {스타일} /* 처음 링크 색상 */
- A: visited {스타일} /* 방문 링크 색상 */
- A: active {스타일} /* 움직일 때 링크 색상 */
- A: hover {스타일} /* 링크에 마우스 포인터 올라갈 때 링크 색상 */

Anchor pseudo-classes : :link, :visited, :active

```
A:link {color: red} /* :link라고만 써도 된다. */
```

```
A:active {color: lime}
```

```
A:visited {color: blue}
```

```
A:link, A:active, A:visited {text-decoration : none }
```

```
<A href = "http://www.naver.com">네이버</A>
```

Cascading order

- **id** 속성을 쓰면 중요도가 가장 높다.
- **class** 등 속성을 쓰면 중요도가 높다.
- **Selector**로 사용한 **element**의 수가 많으면 중요도가 높다.
- **STYLE element** 안에 정의한 것보다 **style** 속성으로 정의한 것이 중요도가 높다.

스타일 속성

- Color : 문자색
- Background-color : 배경색
- background-image:url("URL") : 배경그림
- Background-repeat : 배경 그림 배열
 - - repeat, repeat-x, repeat-y, no-repeat
- Background-position: 표시 위치
(left,right,center,top,bottom)
- Background-attachment : 배경그림 고정(fixed, scroll)
- Line-height : 줄간격(%)

스타일 속성(정렬 방법)

- Text-align : 정렬방법
 - - left, right, center
- Vertical-align : 정렬위치
 - - top, middle bottom, baseline, text-top, text-bottom, super, sub

글꼴 스타일 지정

- Font-style : 스타일
 - - normal, italic, oblique
- Text-decoration : 장식
 - - none, underline, overline, line-through, blink

글꼴 굵기/크기/종류 지정

- Font-weight : 굵기
 - - normal, bold, lighter, bolder
 - - 100, 200, 300, ... 900
- Font-size : 글꼴 크기
 - - 단위를 붙인수치(pt), smaller, lager
- Font-family : 글꼴이름, 글꼴이름, ...

문자 간격, 단어 간격

- Letter-spacing : 문자간격
 - - letter-spacing : 0.5em
- Word-spacing : 단어 간격
 - - word-spacing : 0.5em

머리 기호나 번호의 형식 변경

■ List-style-type : 형식

포인터 모양 지정

- Cursor : 포인터의 모양
- - auto, crosshair, default, pointer, move, text, wait, help, e-resize, no-resize, nw-resize, n-resize, se-resize, sw-resize, w-resize

Box dimension

모든 element는 여러 겹의 상자로 둘러 쌓여 있다고 가정하는 것이 CSS2의 box model이다.

Margin은 투명하다. 따라서 배경색이나 무늬가 비쳐 보이게 된다. Margin, border, padding 속성을 이용해 각각의 폭이나 색을 조정할 수 있다.

세로로 마진이 만날 때는 겹쳐진다.

Content → padding → border → margin

Margin 속성

마진 부분의 폭을 지정하는 속성으로, `margin-left`, `margin-right`, `margin-top`, `margin-bottom`을 이용하면 4 곳의 마진 값을 각각 줄 수 있고, `margin` 속성을 이용하면 4곳의 마진 값을 동시에 줄 수 있다.

`margin-left` : 길이 | 퍼센트 | `auto`

Left 대신 `top` / `right` / `bottom` 을 넣을 수 있다.

Margin 속성 1

길이 : 상대길이와 절대길이 있다.

상대길이 : **em**(현재 글자의 크기), **ex**(현재 글자 중 x자의 크기), **px**(픽셀)

절대길이 : **in**(인치), **cm**(센티미터), **mm**(밀리미터), **pt**(포인트 : $1\text{pt}=1/72\text{in}$),
pc(피카 : $1\text{pc}=12\text{pt}$)

퍼센트 : 부모가 되는(현재의 **element**가 포함된) **element**의 폭을 기준으로 한다.

auto : 브라우저가 자동으로 정한다.

Margin값의 통합 사용

BODY {margin: 2em}

BODY {margin: 1em 2em}

BODY {margin: 1em 2em 3em}

스타일 속성(여백 설정)

- Margin-top : 위쪽 여백
- Margin-bottom : 아래여백
- Margin-left : 왼쪽 여백
- Margin-right : 오른쪽 여백
- Margin: 여백 → 상하좌우여백 동시줄때

Padding 속성

패딩 부분의 폭을 지정하는 속성으로, padding-left, padding-right, padding-top, padding-bottom을 이용하면 4곳의 패딩 값을 각각 줄 수 있다.

padding 속성을 이용하면 4곳의 패딩 값을 동시에 줄 수 있다.

padding-left : 길이 | 퍼센트

```
H1 {background: white; padding: 1em 2em;}
```


외곽선과 내용간의 간격설정

- Padding-top : 위
- Padding-bottom : 아래
- Padding-left : 왼쪽
- Padding-right : 오른쪽
- Padding : 상하좌우 여백
- - padding : 20px

Border 속성

보더 부분의 폭을 지정하는 속성

`border-width` /*외곽선의 굵기 : 1px */

`border-color` /* 외곽선의 색상 : 색상 */

`border-style` /* 외곽선의 스타일 : none, solid, dotted*/

Border

로 나뉘어짐.

외곽선 굵기 지정

- Border-top-width : 위
- Border-bottom-width : 아래
- Border-left-width : 왼쪽
- Border-right-width : 오른쪽
- Border-width : 상하좌우 외곽선 굵기

외곽선 색 정의

- Border-top-color : 위
- Border-bottom-color : 아래
- Border-left-color : 왼쪽
- Border-right-color : 오른쪽
- Border-color : 상하좌우 외곽선 색상

외곽선 스타일 정의

- Border-top-style : 위
- Border-bottom-style : 아래
- Border-left-style : 왼쪽
- Border-right-style : 오른쪽
- Border-style : 상하좌우 외곽선 굵기
 - - none, hidden, **dotted**, dashed, **solid**, double, groove, ridge, inset, outset

Border 속성 1

border-top-width : thin or medium or thick or 길이

H1 {border-width: thin}

H1 {border-width: thin thick}

H1 {border-width: thin thick medium}

Border 속성 2

`border-top-color` : 색이름 | RGB값

H1 {`border-color` : black}

H1 {`border-color` : black yellow}

H1 {`border-color` : black yellow pink}

Position, direction 속성

하나의 **element**에 할당되는 **box**가 브라우저 화면에서 어떻게 배열될 것 인지를 정한다.

{position : absolute or relative or static or fixed}

{direction : ltr or rtl}

Absolute positioning

Absolute positioning은 reference box를 기준으로 상하좌우로 위치를 이동시킨다. Reference box란, 현재 만들고 있는 문서의 가장 상위 element가 차지하는 영역을 말한다. HTML에서는 HTML element의 영역을 말하는데, 보통 브라우저 화면 크기라고 생각해도 된다.

만약, 부모 element가 absolute positioning에 의해 위치가 정해졌다면, 자식 element에 absolute positioning을 적용할 때는 부모 element의 box를 기준으로 위치를 잡게 된다.

Absolute positioning을 하게 되면 margin은 없어지고, padding과 border만 남게 된다.

top : 길이 or 퍼센트 or auto
Left / right / bottom

Float, clear 속성

float 속성은 보통 이미지에 많이 사용한다. 이미지를 왼쪽이나 오른쪽에 떠다니도록 하면, 이미지 다음에 나오는 문자열이 이미지를 둘러싸고 나오게 된다. **HTML**에서는 그림 옆에 문자열을 넣을 경우, 항상 한 줄만 나오고, 나머지는 그림 아래로 내려가게 된다.

float : left / right / none

지정한 방향에는 떠다니는 **box**가 자리할 수 없음을 나타낸다. 즉, **clear**의 값을 **both**로 하면, 해당 **element box**의 좌우에는 떠다니는 **box**가 올 수 없다. 따라서, 떠다니는 **box**의 다음 줄로 내려가게 된다.

clear : none / left / right / both

이미지 좌우 배치 지정

- Float : 배치할 위치
 - - left, right, none
- Clear : 해제 방향
 - - left, right, both, none

Relative positioning

Block-level이나 inline layout에 의해 위치가 결정되면, 이 위치를 기준으로 box를 상하좌우로 이동시킬 수 있다. 이것을 상대적 배열(relative positioning)이라고 한다.

이동시키는 정도는 top, left, bottom, right 속성으로 지정할 수 있다. Box의 위치를 이동시킨다고 해서 다음에 나오는 형제자매 element의 위치가 함께 이동되는 것은 아니다.

따라서, 상대적 배열을 이용하면, element box들이 서로 겹쳐지는 효과를 가질 수 있다.

Z-index 속성

여러 개의 **element**가 서로 상하로 겹치게 만들 수 있는 속성이다.

Position 속성의 값이 **absolute**나 **relative**인 **element**에 적용된다.
초기값은 **auto**이다.

auto : 쓴 순서대로 쌓인다. **z-index**가 0인 것과 같다.

숫자 : 숫자가 낮은 **element**가 밑으로 간다.

단 나누기

잡지처럼 여러 개의 단으로 나뉘어진 문서를 만들 수 있도록 한다.

`columns` : 길이 or 숫자 or 퍼센트

`column-gap` : 길이 or 퍼센트

Width 속성

하나의 element에 의해 그려지는 box의 폭은 width 속성의 값으로 결정된다.

width : 길이 / 퍼센트 / auto

max-width : 길이 / 퍼센트

min-width : 길이 / 퍼센트

Height 속성

하나의 element에 의해 그려지는 box의 높이는 height 속성의 값으로 결정된다.

너비와 높이 지정

- Width : 너비
- Height : 높이

Line-height, vertical-align 속성

문자열의 아랫선에서 다음 줄에 있는 문자열의 아랫선까지의 높이를 line-height라고 한다.

한 줄 안에서의 box 간의 수직 정렬 방식을 정한다. Table에서의 vertical-align 속성

Overflow 속성

Box를 벗어나는 content가 있을 때

Position 속성의 값이 absolute인 element에 적용된다.

visible : 크기에 관계없이 내용을 모두 보여준다.

hidden : 해당 element에 할당된 공간에 들어가는 내용만 보여주고, 나머지는 자른다. Padding과 border는 할당된 공간의 크기에 맞춰서 나타난다.

auto : 할당된 공간의 크기 이상이 되면 스크롤바를 나타낸다.

scroll : 언제나 스크롤바를 나타낸다.

Visible속성 / Color속성

Visible 속성

Element box가 실제로는 존재하지만 화면에는 보이지 않도록 할 수 있다.

visible : inherit / visible / hidden

Color 속성

Element의 글자색을 나타낸다.

Background 속성

Element의 배경색이나 무늬를 지정할 수 있다.

`background-color` : 색 이름 / RGB 값

`background-image` : url / none

`background-repeat` : repeat / repeat-x / repeat-y / no-repeat