

16장. CSS3-2

표 4-1 크기 단위

단위	설명
%	백분율 단위
em	배수 단위
px	픽셀

표 4-2 색상 단위

단위 형태	설명
#000000	HEX 코드 단위
rgb(red, green, blue)	RGB 색상 단위
rgba(red, green, blue, alpha)	RGBA 색상 단위
hsl(hue, saturation, lightness)	HSL 색상 단위
hsla(hue, saturation, lightness, alpha)	HSLA 색상 단위

코드 4-8 키워드를 사용한 색상 적용

```
<style>
  h1 { background-color: red; }
  h2 { background-color: orange; }
  h3 { background-color: blue; }
  h4 { background-color: green; }
  h5 { background-color: brown; }
  h6 { background-color: purple; }
</style>
```

코드 4-14 URL 경로 표시 방법

```
/* 현재 폴더의 Desert.jpg */
background-image: url('Desert.jpg');

/* 현재 폴더 내부의 Other 폴더의 Desert.jpg */
background-image: url('Other/Desert.jpg');

/* 루트 폴더의 Desert.jpg */
/* 루트 폴더의 개념은 서버를 알아야 합니다. */
background-image: url('/Desert.jpg');
```

* 파일을 지정할 때는 url단위 사용

표 4-3 display 속성에 적용 가능한 키워드

스타일 이름	설명
none	태그를 화면에서 보이지 않게 만듭니다.
block	태그를 block 형식으로 지정합니다.
inline	태그를 inline 형식으로 지정합니다.
inline-block	태그를 inline-block 형식으로 지정합니다.

* display 속성은 태그의 영역 표현 방식을 지정하는 속성

코드 4-19 inline 키워드와 inline-block 키워드

<pre> <style> #box { display: inline; background-color: red; width: 300px; height: 50px; margin: 10px; } </style> </pre>	<pre> <style> #box { display: inline-block; background-color: red; width: 30px; height: 50px; margin: 10px; } </style> </pre>
---	--

- none 키워드를 사용하면 태그가 완전히 화면에서 제거된다.

그림 4-12 none 키워드

Dummy Dummy

- block 키워드를 사용하면 태그가 한 줄을 차지한다.

그림 4-13 block 키워드 적용

Dummy
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Dummy

- inline 키워드 또는 inline-block 키워드를 사용하면 한 줄에 들어간다.

그림 4-14 inline 키워드와 inline-block 키워드 적용

Dummy	Lorem ipsum dolor sit amet, consectetur adipiscing elit.	Dummy
-------	--	-------

- inline 키워드를 사용할 때와 inline-block 키워드를 사용할 때의 차이점은 margin 속성과 padding 속성을 적용할 때에 알 수 있다.
- inline 키워드를 사용하면 margin 속성과 padding 속성이 좌우로만 적용된다.

그림 4-15 inline 키워드 적용

Dummy	Lorem ipsum dolor sit amet, consectetur adipiscing elit.	Dummy
-------	--	-------

- inline-block 키워드를 사용하면 margin 속성과 padding 속성이 상하 좌우로 적용된다.

그림 4-16 inline-block 키워드 적용

Dummy	<div> Lorem ipsum dolor sit amet, consectetur adipiscing elit. </div>	Dummy
-------	---	-------

표 4-4 visibility 속성에 적용 가능한 키워드

스타일 이름	설명
visible	태그를 보이게 만듭니다.
hidden	태그를 보이지 않게 만듭니다.
collapse	table 태그를 보이지 않게 만듭니다.

* 대상을 보이거나 보이지 않게 지정하는 속성이다.

그림 4-18 visibility 속성에 hidden 키워드 적용



❖ opacity 속성

- 대상의 투명도를 지정하는 속성이다.
- 0부터 1 사이의 숫자를 입력한다. (0은 투명이고 1은 불투명이다.)
- 다음과 같은 방법으로 사용한다.

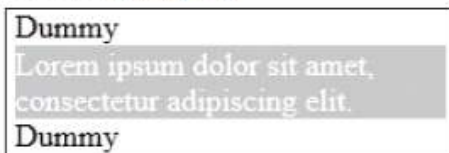
코드 4-22 opacity 속성

```
<style>
  #box {
    background-color: black;
    color: white;

    opacity: 0.2;
  }
</style>
```

- 코드를 실행하면 다음과 같이 출력한다.

그림 4-21 opacity 속성

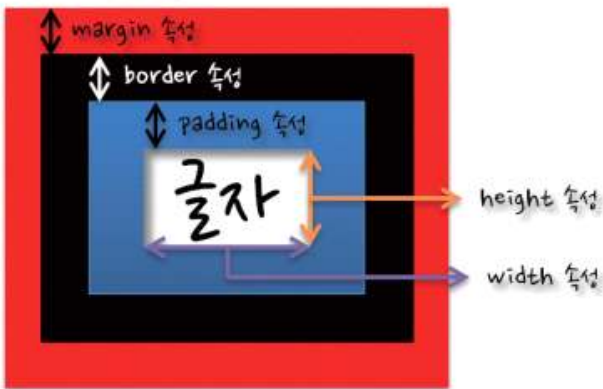


- * **none** : 블록이 생성되지 않는다 (보이지 않음)
- * **block** : 블록 또는 문단처럼 취급. 이 경우 요소의 옆에 다른 요소가 표시될 수 없다.
예외 : float 속성을 다른 요소에 대입할 수 있다.
- * **inline** : 현재 블록의 선에 맞춰진다.
- * **inline-block** : 박스를 블록 처리하고 그 안의 아이템들을 인라인 박스 형태로 처리

float : 흐르는 방법 및 방향을 지정한다. 이 속성은 절대 위치를 갖는 요소에서는 적용되지 않는다.

- ❖ 박스 속성은 웹 페이지의 레이아웃을 구성할 때 가장 중요한 속성이다.
- ❖ 다음 속성을 모두 합쳐 박스 속성이라고 이야기한다.

그림 4-22 박스 속성



코드 4-23 width 속성과 height 속성

```
<!DOCTYPE html>
<html>
<head>
  <title>CSS Property Basic</title>
  <style>
    div {
      width: 100px; height: 100px;
      background-color: red;
    }
  </style>
</head>
<body>
  <div></div>
</body>
</html>
```

그림 4-23 width 속성과 height 속성



* Width속성과 height 속성은 글자를 감싸는 영역의 크기를 지정하는 속성

코드 4-24 margin 속성과 padding 속성

```
<!DOCTYPE html>
<html>
<head>
  <title>CSS Property Basic</title>
  <style>
    div {
      width: 100px; height: 100px;
      background-color: red;

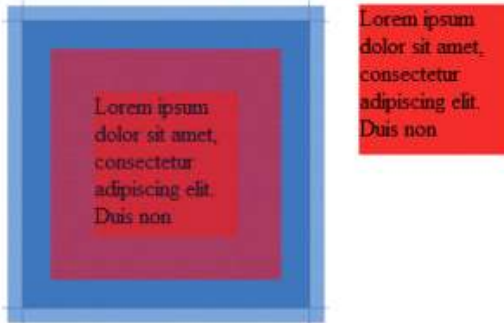
      border: 20px solid black;
      margin: 10px; padding: 30px;
    }
  </style>
</head>
<body>
  <div></div>
</body>
</html>
```

그림 4-24 margin 속성과 padding 속성



- margin 속성과 padding 속성은 width 속성과 height 속성과 별도로 적용된다.

그림 4-25 width 속성과 height 속성의 적용 범위



- 박스 크기는 다음과 같은 공식을 갖는다.

전체 너비 = width + 2 × (margin + border + padding)

전체 높이 = height + 2 × (margin + border + padding)

❖ border-width, border-style, border-color 속성을 사용해야 테두리를 생성할 수 있다.

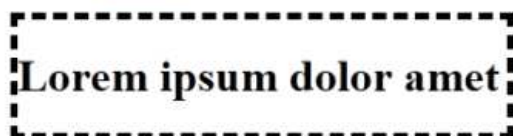
- 각각의 속성은 다음과 같은 방법으로 사용한다.

코드 4-28 테두리 속성

```
<style>
  .box {
    border-width: thick;
    border-style: dashed;
    border-color: black;
  }
</style>
```

- 코드를 실행하면 다음과 같이 출력한다.

그림 4-34 테두리 속성



❖ 방금 살펴본 3가지 테두리 속성은 border 속성으로 한번에 입력할 수 있다.

코드 4-29 border 속성

```
<style>
  .box {
    border: thick dashed black;
  }
</style>
```

❖ border-radius 속성

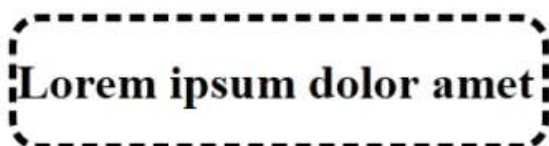
- border-radius 속성을 사용하면 테두리가 둥근 사각형 또는 원을 만들 수 있다.
- 다음과 같은 방식으로 사용한다.

코드 4-30 border-radius 속성

```
<style>
  .box {
    border: thick dashed black;
    border-radius: 20px;
  }
</style>
```

- 코드를 실행하면 다음 그림처럼 출력한다.

그림 4-38 border-radius 속성



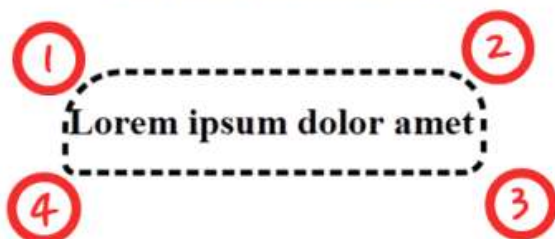
- 각각의 테두리의 둥글기를 설정할 수도 있다.

코드 4-31 각각의 모서리에 다른 border-radius 속성 적용

```
<style>
  .box {
    border: thick dashed black;

    /* border-radius: 왼쪽위 오른쪽위 오른쪽아래 왼쪽아래 */
    border-radius: 50px 40px 20px 10px;
  }
</style>
```

그림 4-39 각각의 모서리에 다른 border-radius 속성 적용



코드 4-32 background-image 속성

```

<!DOCTYPE html>
<html>
<head>
  <title>CSS3 Property Basic</title>
  <style>
    body {
      background-image: url('BackgroundFront.png');
    }
  </style>
</head>
<body>

</body>
</html>

```

그림 4-41 폴더 구성

BackgroundBack.png

BackgroundFront.png



*배경에 넣을 그림을 지정하는 속성

그림 4-42 background-image 속성



코드 4-33 이미지 중첩 사용

```

<style>
  body {
    background-image: url('BackgroundFront.png'), url('BackgroundBack.png');
  }
</style>

```

그림 4-43 이미지 중첩



* 왼쪽에 위치하는 그림이 앞에 위치한다.

코드 4-34 background-size 속성의 너비

```
<style>
  body {
    background-image: url('BackgroundFront.png'), url('BackgroundBack.png');
    background-size: 100%;
  }
</style>
```

- * 그림 크기를 조절할 때 사용
- * css3에서 추가된 속성이므로 인터넷 익스플로러 8 이하에서는 사용할 수 없다.

코드 4-34 background-size 속성의 너비

```
<style>
  body {
    background-image: url('BackgroundFront.png'), url('BackgroundBack.png');
    background-size: 100%;
  }
</style>
```

그림 4-45 background-size 속성의 너비



코드 4-35 background-size 속성의 높이

```
<style>
  body {
    background-image: url('BackgroundFront.png'), url('BackgroundBack.png');
    background-size: 100% 250px;
  }
</style>
```

- * 크기 단위를 2개 입력하면 두 번째 크기 단위는 높이를 의미한다.

코드 4-36 중첩 이미지의 크기 조정

```
<style>
  body {
    background-image: url('BackgroundFront.png'), url('BackgroundBack.png');
    background-size: 100%, 200%;
  }
</style>
```


코드 4-37 background-repeat 속성에 no-repeat 키워드 적용

```
<style>
  body {
    background-image: url('BackgroundFront.png'), url('BackgroundBack.png');
    background-size: 100%;
    background-repeat: no-repeat;
  }
</style>
```

* 배경 패턴 방식을 지정하는 속성이다.

그림 4-50 background-repeat 속성에 no-repeat 키워드 적용



코드 4-39 background-attachment 속성에 fixed 키워드 적용

```
<style>
  body {
    background-color: #E7E7E8;
    background-image: url('BackgroundFront.png'), url('BackgroundBack.png');
    background-size: 100%;
    background-repeat: no-repeat;
    background-attachment: fixed;
  }
</style>
```

* 배경 이미지를 어떠한 방식으로 화면에 붙일 것인지를 지정하는 속성

그림 4-52 background-attachment 속성에 fixed 키워드 적용



그림 4-54 background-attachment 속성에 fixed 키워드 적용



코드 4-40 background-position 속성에 bottom 키워드 적용

```
<style>
  body {
    background-color: #E7E7E8;
    background-image: url('BackgroundFront.png'), url('BackgroundBack.png');
    background-size: 100%;
    background-repeat: no-repeat;
    background-attachment: fixed;
    background-position: bottom;
  }
</style>
```

*배경의 위치를 지정하는 속성

* 형식 값

- background-position : 키워드;
- background-position : X축 크기;
- background-position : X축 크기 Y축 크기;

그림 4-56 background-position 속성에 bottom 키워드 적용



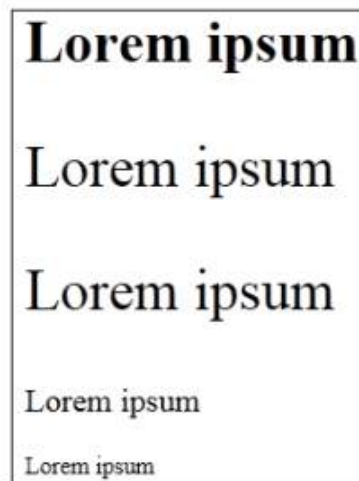
코드 4-44 font-size 속성

```

<!DOCTYPE html>
<html>
<head>
  <title>CSS3 Font Property</title>
  <style>
    .a { font-size: 32px; }
    .b { font-size: 2em; }
    .c { font-size: large; }
    .d { font-size: small; }
  </style>
</head>
<body>
  <h1>Lorem ipsum</h1>
  <p class="a">Lorem ipsum</p>
  <p class="b">Lorem ipsum</p>
  <p class="c">Lorem ipsum</p>
  <p class="d">Lorem ipsum</p>
</body>
</html>

```

그림 4-59 font-size 속성 적용



* font-size속성

코드 4-45 font-family 속성

```

<!DOCTYPE html>
<html>
<head>
  <title>CSS3 Font Property</title>
  <style>
    .font_arial { font-family: Arial }
    .font_roman { font-family: 'Times New Roman' }
  </style>
</head>
<body>
  <h1 class="font_arial">Lorem ipsum</h1>
  <p class="font_roman">Lorem ipsum</p>
</body>
</html>

```

- * 사용자의 컴퓨터에 폰트가 없으면 폰트가 적용되지 않는다.
- * 만약을 대비해 여러 개의 폰트를 연속적으로 입력
- * 하지만 다국적 웹 페이지를 제공할 경우 사용자에게 무슨 폰트가 있는지 확인불가
- * 이러한 경우 generic-family폰트 사용
- * generic-family폰트는 웹 브라우저에서 미리 지정하고 있는 폰트

- Serif 폰트(명조체), Sans-serif 폰트(고딕체), Mono space 폰트(고정 폭 글꼴)를 사용할 수 있다.
- 다음과 같은 방법으로 사용한다.

코드 4-47 generic-family 폰트

```
<!DOCTYPE html>
<html>
<head>
  <title>CSS3 Font Property</title>
  <style>
    .font_arial { font-family: '없는 폰트', sans-serif }
    .font_roman { font-family: '없는 폰트', serif }
  </style>
</head>
<body>
  <h1 class="font_arial">Lorem ipsum</h1>
  <p class="font_roman">Lorem ipsum</p>
</body>
</html>
```

인터넷 익스플로러의 경우
sans-serif 폰트와 serif 폰
트에 따옴표를 적용하면 안
됩니다.

그림 4-65 generic-family 폰트 사용

Lorem ipsum

Lorem ipsum

코드 4-48 font-style 속성과 font-weight 속성

```
<!DOCTYPE html>
<html>
<head>
  <title>CSS3 Font Property</title>
  <style>
    .font_big { font-size: 2em }
    .font_italic { font-style: italic }
    .font_bold { font-weight: bold }
  </style>
</head>
<body>
  <p class="font_big font_italic font_bold">Lorem ipsum dolor amet</p>
</body>
</html>
```

그림 4-68 font-style 속성과 font-weight 속성

Lorem ipsum dolor amet

CSS Web Safe Font Combinations

Commonly Used Font Combinations (흔히 사용하는 글꼴 조합)

The font-**family** property should hold several font names as a "fallback" system, to ensure maximum compatibility between browsers/operating systems. If the browser does not support the first font, it tries the next font.

font-**family** 속성은 일종의 보험으로 몇 개의 글꼴 이름을 가지고 있어야 한다. 왜냐하면 브라우저나 O/S에 따라 글꼴이 다르기 때문이다. 브라우저가 첫 글꼴을 지원하지 않을 때는 그 다음 글꼴을 사용하게 된다.

Start with the font you want, and end with a **generic family**, to let the browser pick a similar font in the **generic family**, if no other fonts are available:

원하는 글꼴로 시작하고, 글꼴 가족의 총칭으로 끝을 낸다. 그러면 브라우저가 원하는 글꼴이 없을 때 글꼴 가족 중에서 가장 비슷한 것을 골라 사용한다.

Example

```
p{font-family:"Times New Roman", Times, serif}
```

Below are some commonly used font combinations, organized by **generic family**.

아래는 흔히 사용하는 글꼴 조합이다. (물론 서양권에서만)

아래 표에서 Serif 글꼴이란 획의 끝에 약간 꺾인 꾸미기 형태가 있는 것을 말하고, 우리 나라 글꼴에선 명조, 굴림, 바탕 등에 해당한다. Windows, Mac OS, Linux 등에 따라 비슷한 글자 형태를 다르게 부른다.

Sans-Serif란 Serif가 없다는 뜻이다. 획이 직선 형태인 것을 말하고, 우리 나라 글꼴에선 고딕, 굴림, 돋움 등에 해당한다. Windows, Mac OS, Linux 등에 따라 비슷한 글자 형태를 다르게 부른다.

Monospace란 고정폭 글꼴을 말한다. 글자의 폭이 모두 동일한 것을 뜻하며 우리 나라 글꼴에선 Windows의 굴림체, 바탕체, 굴림체, 돋움체 같이 글꼴 이름 뒤에 "~체"라는 것이 붙어 있다. 이런 글꼴을 사용하면 세로 줄 맞추기가 편하다. 서양에선 타자기 글꼴이나 시스템 부팅 할 때 보여주는 모니터 글꼴이 이런 것에 해당된다.

프로그램 코드를 작성할 때는 숫자 1(일), 소문자 l(엘), 대문자 I(아이), 숫자 0(영), 대문자 O(오) 등이 명확히 구분되는 글꼴을 사용하는 것이 좋다. 한글 글꼴이 영문 글꼴에 비해서 예쁘지 않은 이유는 한글이 영문처럼 옆으로 음소를 풀어 쓰지 않고 조립하여 음절 형태로 표시 되기 때문이다. 그런데 글꼴의 크기를 매우 작게 사용하기 때문에 한글의 표현할 공간이 좁아서 획이 예쁘게 보이지 않는 것이다. 모니터에서 한글이 제대로 표시 되려면 최소 11포인트 크기여야 하고 이 정도 크기면 글꼴의 차이를 나타내기 어렵다. 여러 공짜 글꼴이 인터넷에 제공되고 있어서 웹 폰트 형태로 다운로드 해서 사용할 수 있으나 한글의 경우는 배보다 배꼽이 큰 경우로 전송할 문서의 내용보다 글꼴의 용량이 더 크다. 고로 어떤 시스템에서 동일하게 보이길 원한다면 차라리 이미지 파일로 만들어 붙이는 것이 낫다. 권장하는 포맷은 GIF 형태이다. 파일 용량이 매우 적다.

Serif Fonts

font-family	Example text
Georgia, serif	This is a heading This is a paragraph
"Palatino Linotype", "Book Antiqua", Palatino, serif	This is a heading This is a paragraph
"Times New Roman", Times, serif	This is a heading This is a paragraph

Sans-Serif Fonts

font-family	Example text
Arial, Helvetica, sans-serif	This is a heading This is a paragraph
"Arial Black", Gadget, sans-serif	This is a heading This is a paragraph
"Comic Sans MS", cursive, sans-serif	This is a heading This is a paragraph
Impact, Charcoal, sans-serif	This is a heading This is a paragraph
"Lucida Sans Unicode", "Lucida Grande", sans-serif	This is a heading This is a paragraph
Tahoma, Geneva, sans-serif	This is a heading This is a paragraph
"Trebuchet MS", Helvetica, sans-serif	This is a heading This is a paragraph
Verdana, Geneva, sans-serif	This is a heading This is a paragraph

Monospace Fonts

font-family	Example text
"Courier New", Courier, monospace	This is a heading This is a paragraph
"Lucida Console", Monaco, monospace	This is a heading This is a paragraph

❖ text-decoration 속성

- a 태그에 href 속성이 지정되면 다음과 같이 밑줄이 표시된다.

그림 4-76 링크 표시

[Lorem ipsum dolor amet](#)

- 이를 제거하려면 다음과 같이 text-decoration 속성에 none 키워드를 지정한다.

코드 4-53 text-decoration 속성

```
<!DOCTYPE html>
<html>
<head>
  <title>CSS3 Font Property</title>
  <style>
    a { text-decoration: none }
  </style>
</head>
<body>
  <h1>
    <a href="#">Lorem ipsum dolor amet</a>
  </h1>
</body>
</html>
```

그림 4-77 text-decoration 속성

[Lorem ipsum dolor amet](#)

❖ position 속성

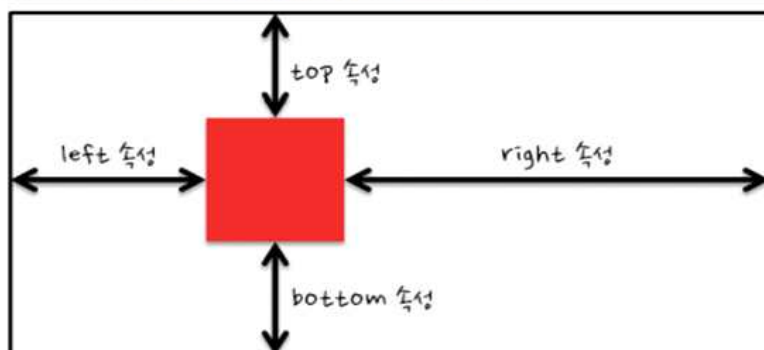
- 태그의 위치 설정 방법을 변경할 때 사용한다

표 4-5 position 속성에 적용할 수 있는 키워드

키워드	설명
static	태그가 위에서 아래로 순서대로 배치됩니다.
relative	초기 위치 상태에서 상하좌우로 위치를 이동합니다.
absolute	절대적 위치 좌표를 설정합니다.
fixed	화면을 기준으로 절대적 위치 좌표를 설정합니다.

- position 속성은 반드시 다음 스타일 속성과 함께 사용한다.

그림 4-80 position 속성과 함께 사용하는 스타일 속성



코드 4-55 position 속성과 함께 사용하는 스타일 속성

```
<style>
  .box {
    width: 100px; height: 100px;
    position: absolute;
  }
  .box:nth-child(1) {
    background-color: red;
    left: 10px; top: 10px;
  }
  .box:nth-child(2) {
    background-color: green;
    left: 50px; top: 50px;
  }
  .box:nth-child(3) {
    background-color: blue;
    left: 90px; top: 90px;
  }
</style>
```

그림 4-81 position 속성을 사용한 위치 지정



❖ z-index 속성

- HTML 태그는 아래 입력한 태그가 위로 올라온다.
- 이러한 순서를 변경할 때에 z-index 속성을 사용한다.
- 큰 값을 입력할 수록 위로 올라온다.
- 다음과 같은 방법으로 사용한다.

코드 4-56 z-index 속성

```
<style>
  .box {
    width: 100px; height: 100px;
    position: fixed;
  }
  .box:nth-child(1) {
    background-color: red;
    left: 10px; top: 10px;

    z-index: 100;
  }
  .box:nth-child(2) {
    background-color: green;
    left: 50px; top: 50px;

    z-index: 10;
  }
  .box:nth-child(3) {
    background-color: blue;
    left: 90px; top: 90px;

    z-index: 1;
  }
</style>
```

그림 4-82 z-index 속성



❖ 위치 속성과 관련된 공식

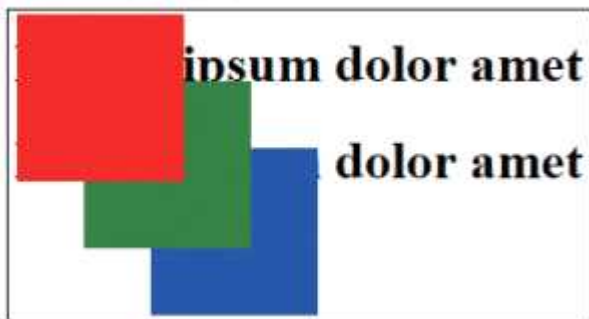
- 현재 만들고 있는 예제에서 다음과 같은 코드의 실행 결과를 예측해보자

코드 4-57 body 태그 구성

```
<body>
  <h1>Lorem ipsum dolor amet</h1>
  <div>
    <div class="box"></div>
    <div class="box"></div>
    <div class="box"></div>
  </div>
  <h1>Lorem ipsum dolor amet</h1>
</body>
```

- 코드를 실행하면 다음과 같이 출력한다.

그림 4-83 실행 결과



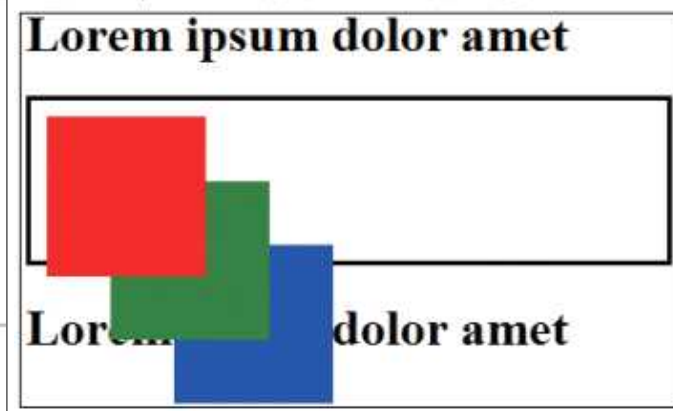
- 다음과 같은 문제를 발견할 수 있다.
 - h1 태그 두 개가 붙어 있다(div 태그가 영역을 차지하지 않는다).
 - 색상이 적용된 상자가 자신의 부모를 기준으로 위치를 잡지 않는다.
- 이를 해결할 때는 다음과 같은 공식을 사용한다.
 - 자손에게 position 속성을 absolute 키워드로 적용하면 부모에게 height 속성을 입력한다.
 - 자손의 position 속성을 absolute 키워드로 적용하면 부모의 position 속성을 relative 키워드로 적용한다.
- 이를 사용하면 다음과 같이 코드를 적용할 수 있다.

코드 4-59 position 속성에 relative 키워드 적용

```
body > div {
  width: 400px; height: 100px;
  border: 3px solid black;

  position: relative;
}
```

그림 4-84 position 속성에 relative 키워드 적용



❖ overflow 속성

- 내부의 요소가 부모의 범위를 벗어날 때 어떻게 처리할지 지정하는 속성이다.
- 다음과 같은 키워드를 입력한다.

표 4-6 overflow 속성에 적용 가능한 키워드

스타일 이름	설명
hidden	영역을 벗어나는 부분을 보이지 않게 만듭니다.
scroll	영역을 벗어나는 부분을 스크롤로 만듭니다.

- 다음과 같은 방식으로 사용한다.

코드 4-60 overflow 속성에 hidden 키워드 적용

```
body > div {
  width: 400px; height: 100px;
  border: 3px solid black;

  position: relative;
  overflow: hidden;
}
```

- 각각의 키워드를 적용할 때에 다음과 같이 실행된다.

그림 4-85 overflow 속성에 hidden 키워드 적용

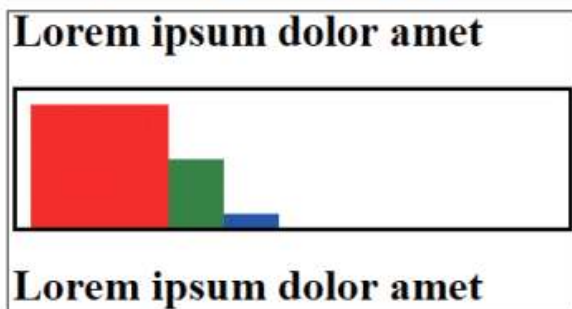
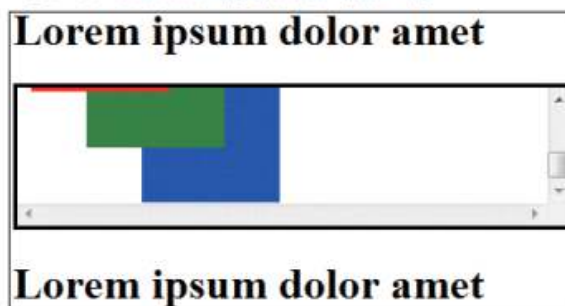


그림 4-86 overflow 속성에 scroll 키워드 적용



❖ float 속성을 사용한 수평 정렬

- 동위 위치에 있는 태그에 모두 float 속성을 적용하면 수평 정렬된다.
- 다음 코드에서 div.box 태그는 동위 위치에 있다.

코드 4-65 body 태그 구성

```
<body>
  <div class="box">1</div>
  <div class="box">2</div>
</body>
```

- 이러한 태그에 다음과 같은 스타일을 적용한다.

코드 4-66 float 속성에 적용 가능한 키워드

```
<style>
  .box {
    width: 100px; height: 100px;
    background-color: red;
    margin: 10px; padding: 10px;

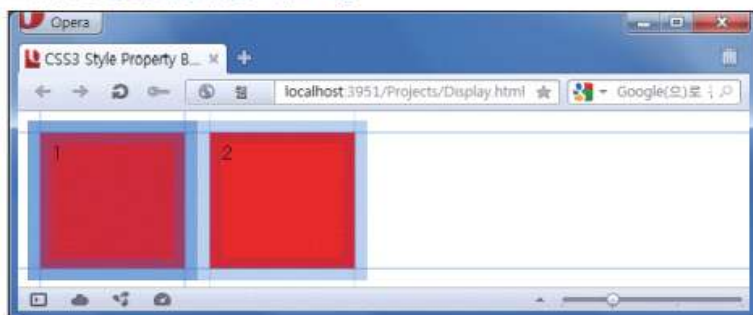
    /* 태그를 왼쪽으로 붙입니다. */
    float: left;
  }
</style>
```

```
<style>
  .box {
    width: 100px; height: 100px;
    background-color: red;
    margin: 10px; padding: 10px;

    /* 태그를 오른쪽으로 붙입니다. */
    float: right;
  }
</style>
```

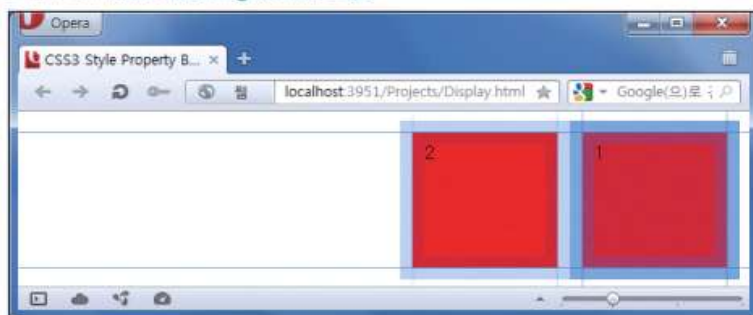
- 코드를 실행하면 다음과 같이 수평 정렬된다.

그림 4-90 float 속성에 left 키워드 적용



- right 키워드를 적용했을 경우에는 위에 위치한 태그가 오른쪽에 붙는다는 것을 주의한다.

그림 4-91 float 속성에 right 키워드 적용



❖ text-shadow 속성

- 글자에 그림자를 부여하는 속성이다.
- 다음과 같은 형태로 값을 입력한다.

그림 4-97 text-shadow 속성

text-shadow: 5px 5px 5px black

오른쪽 아래 흐림도 색상

- 다음과 같이 사용한다.

```
<style>
  h1 {
    text-shadow: 5px 5px 5px black;
  }
</style>
```

그림 4-98 text-shadow 속성 적용

Lorem ipsum dolor amet

❖ box-shadow 속성

- 박스에 그림자를 부여하는 속성이다.
- 다음과 같은 형태로 값을 입력한다.

그림 4-99 box-shadow 속성

box-shadow: 5px 5px 5px black

오른쪽 아래 흐림도 색상

- 다음과 같이 사용한다.

```
<style>
  div {
    border: 3px solid black;
    box-shadow: 10px 10px 30px black;
    text-shadow: 5px 5px 5px black;
  }
</style>
```

그림 4-100 box-shadow 속성 적용

Lorem ipsum dolor amet

❖ 벤더 프리픽스 개요

- 벤더 프리픽스는 웹 브라우저 공급 업체(마이크로소프트, 구글, 모질라, 애플, 오페라)에서 제공하는 실험적인 기능을 사용할 때 사용한다.
- 예를 들어 변환 속성은 CSS3 표준에 있지만 아직 완벽하게 제정된 상태가 아니다. 따라서 웹 브라우저 업체가 무턱대고 속성을 추가할 수 없다.
- 하지만 다른 웹 브라우저를 이기려면 새로운 기능을 모두 제공해야 하므로 벤더 프리픽스를 사용해 지원한다.
- 각 웹 브라우저마다 다음과 같은 벤더 프리픽스를 갖는다.

그림 4-103 벤더 프리픽스



❖ 벤더 프리픽스 사용방법

코드 4-75 벤더 프리픽스

```
<style>
  input[type=checkbox]:checked + div {
    height: 0px;
  }

  div {
    overflow: hidden;
    width: 650px; height: 300px;

    /* 변환 효과를 적용합니다. */
    -ms-transition-duration: 1s;
    -webkit-transition-duration: 1s;
    -moz-transition-duration: 1s;
    -o-transition-duration: 1s;
    transition-duration: 1s;
  }
</style>
```

* @import 규칙

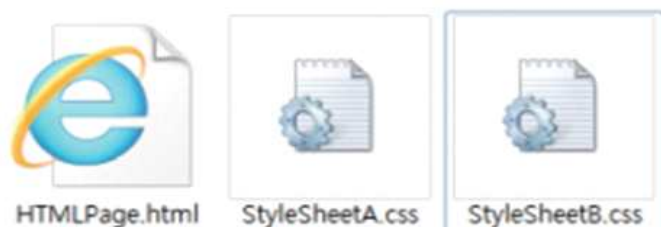
❖ 규칙

- 스타일시트 내부에서 @가 붙은 단어를 규칙(@-rule)이라고 부른다.

❖ @import 규칙

- 스타일시트 내부에서 스타일시트를 추가할 때에 사용하는 규칙
- 다음과 같은 폴더를 구성한다.

그림 11-2 폴더 구성



- 각각의 스타일시트는 다음 형태

코드 11-2 스타일시트 구성

StyleSheetA.css 파일	StyleSheetB.css 파일
<pre>body { color: red; }</pre>	<pre>html { background: black; }</pre>

- 일반적으로는 link 태그를 사용해 다음과 같이 외부 스타일시트를 추가한다.

코드 11-4 link 태그를 사용한 스타일시트 추가

```
<head>
  <title>Import Query Basic</title>
  <link rel="stylesheet" href="StyleSheetA.css" />
  <link rel="stylesheet" href="StyleSheetB.css" />
</head>
```

- @import 규칙을 사용하면 다음과 같이 스타일시트를 추가한다.

코드 11-5 @import 규칙을 사용한 스타일시트 추가

```
<head>
  <title>Import Query Basic</title>
  <style>
    @import url(StyleSheetA.css);
    @import url(StyleSheetB.css);
  </style>
</head>
```

❖ @font-face 규칙

- @font-face 규칙은 웹 폰트를 생성할 때 사용하는 규칙이다.
- 다음과 같은 폴더를 구성한다.

그림 11-4 폴더 구성



- 다음과 같이 사용한다.

코드 11-10 src 속성

```
<style>
  @font-face {
    font-family: 'AAA';
    src: local('NanumGothic'),
        url('NanumGothic.eot'),
        url('NanumGothic.ttf'),
        url('NanumGothic.woff');
  }

  * {
    font-family: 'AAA';
  }
</style>
```

생성한 폰트의 이름을 지정합니다.
아무렇게나 입력해도 됩니다.

생성한 폰트의 이름을 넣어줍니다.

- 웹 브라우저에 따라서 지원하는 폰트 형식이 다르므로 주의한다.

표 11-1 웹 브라우저가 지원하는 폰트 형식

eot	○				
ttf		○	○	○	○
otf		○	○	○	○
svg					○
woff	○	○	○		

❖ @media 규칙

- 다양한 장치에서 HTML 문서가 적절한 형태를 갖추게 만들어주는 규칙이다. 최근 HTML 페이지가 다양한 장치에서 실행되면서 중요하게 부각된 규칙이다.
- 다음과 같은 폴더를 구성한다.
그림 11-7 폴더 구성



❖ 미디어 쿼리

- link 태그는 다음과 같이 media 속성을 사용할 수 있다.

코드 11-12 media 속성

```
<head>
  <title>Media Query Basic</title>
  <link rel="stylesheet" href="desktop.css" media="screen" />
  <link rel="stylesheet" href="print.css" media="print" />
</head>
```

❖ 이때 media 속성의 내부에 입력하는 문자열을 미디어 쿼리라고 부른다.

코드 11-13 Desktop.css 파일

```
html {
  height: 100%;
  background: black;
}

body {
  color: white; font-family: serif;
}
```

코드 11-14 Print.css 파일

```
h1 {
  text-align: center;
  color: red; font-family: sans-serif;
}
```

- 코드를 실행하면 데스크톱에 적용되는 스타일시트와 프린트에서 적용되는 스타일시트가 다름을 알 수 있다.

그림 11-10 데스크톱 웹 브라우저 실행 결과



그림 11-11 프린트 결과



- ❖ 일반적으로 link 태그의 media 속성을 사용하기보다는 @media 규칙을 사용한다. media 규칙은 다음과 같이 사용한다..

코드 11-15 @media 규칙

```
<!DOCTYPE html>
<html>
<head>
  <title>Media Query Basic</title>
  <style>
    @media screen {
      html {
        height: 100%;
        background: black;
      }
      body {
        color: white; font-family: serif;
      }
    }

    @media print {
      h1 {
        text-align: center;
        color: red; font-family: sans-serif;
      }
    }
  </style>
</head>
<body>
  <h1>Lorem ipsum</h1>
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
  <p>Aenean luctus congue scelerisque. Maecenas aliquet ante.</p>
</body>
</html>
```