

# CSS3 속성

---

- ◆ 박스 모델
- ◆ 테두리 속성
- ◆ **박스 속성**
- ◆ 레이아웃 속성
- ◆ 표 속성

# 둥근 모서리

## ◆ border-radius 속성

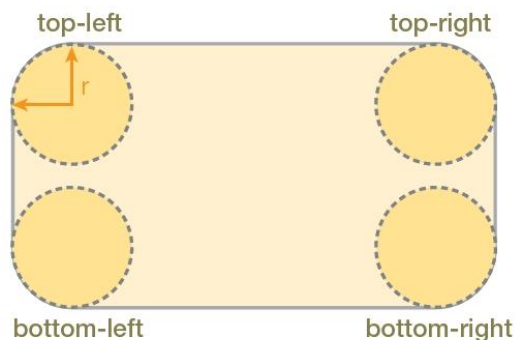
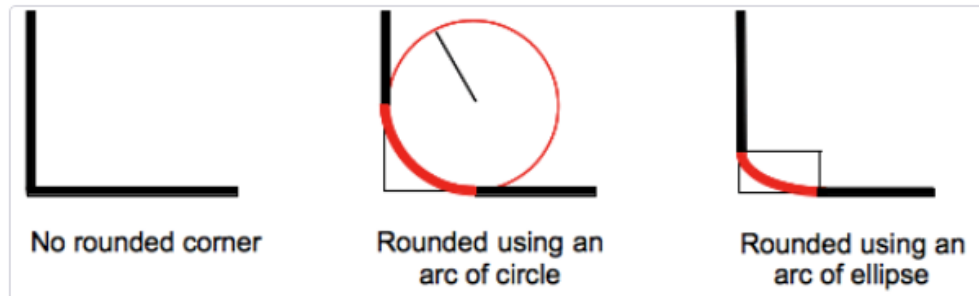


그림 7-4 둥근 모서리 속성값 설정 방법

```
<style>
#example1 {
  border: 2px solid red;
  padding: 10px;
  border-radius: 25px;
}

#example2 {
  border: 2px solid red;
  padding: 10px;
  border-radius: 50px 20px;
}
</style>
```

radius	
top-left-and-bottom-right	
top-right-and-bottom-left	
top-left	
top-right	
bottom-right	
bottom-left	



```
/* circle if box is a square or ellipse if box is a rectangle */
border-bottom-left-radius: 20%;

/* same as above: 20% of horizontal(width) and vertical(height) */
border-bottom-left-radius: 20% 20%;

/* 20% of horizontal(width) and 10% of vertical(height) */
border-bottom-left-radius: 20% 10%;
```

## 예제 7-7 박스에 둥근 모서리 적용하기

```
<head>
  <style>
    body {
      font-weight: bold;
      font-size: 12pt;
    }
    .br1 {
      background-color: lime;
      color: red;
      border-style: solid;
      border-width: thick;
      border-color: green;
      border-radius: 30%;
    }
    .br2 {
      background-color: maroon;
      color: yellow;
      border-style: dotted;
      border-width: 2px;
      border-color: white;
      border-radius: 15px 35px;
    }
    .br3 {
      background-color: teal;
      color: aqua;
      border-style: dashed;
      border-width: 5px;
      border-color: red;
      border-radius: 5px 15px 25px 35px;
    }
  </style>
</head>
<body>
  <h1 class="br1">둥근 모서리 스타일링 예제 1</h1>
  <h1 class="br2">둥근 모서리 스타일링 예제 2</h1>
  <h1 class="br3">둥근 모서리 스타일링 예제 3</h1>
  <h1 class="br4">둥근 모서리 스타일링 예제 4</h1>
</body>
```

```
.br4 {
  border: 3px solid red;
  border-top-left-radius: 30px;
}
</style>
</head>
<body>
  <h1 class="br1">둥근 모서리 스타일링 예제 1</h1>
  <h1 class="br2">둥근 모서리 스타일링 예제 2</h1>
  <h1 class="br3">둥근 모서리 스타일링 예제 3</h1>
  <h1 class="br4">둥근 모서리 스타일링 예제 4</h1>
</body>
```

# CSS box-shadow

## ◆ 구문:

```
{ box-shadow: 수평 그림자(필수) | 수직 그림자(필수) | 그림자 흐림 | 그림자 번짐 | 그림자 색상 | 삼  
입 효과; }
```

박스 그림자 스타일링 예제 1

박스 그림자 스타일링 예제 2

박스 그림자 스타일링 예제 3

박스 그림자 스타일링 예제 4

## 예제 7-8 박스에 그림자 효과 주기

```
<head>
  <style>
    body {
      font-weight: bold;
      font-size: 12pt;
    }
    .boxshadow1 {
      background-color: yellow;
      border: 5px solid blue;
      box-shadow: 10px 10px 0px teal;
    }
    .boxshadow2 {
      background-color: orange;
      border: 5px solid red;
      box-shadow: 20px 20px 50px red;
    }
    .boxshadow3 {
      background-color: silver;
      border: 5px solid black;
      box-shadow: 20px 20px 30px -20px
                  maroon;
    }
    .boxshadow4 {
      background-color: lime;
      border: 5px solid olive;
      box-shadow: 10px 10px 0px 10px
                  fuchsia inset;
    }
  </style>
</head>
```

```
<body>
  <h1 class="boxshadow1">박스 그림자 스타일링 예제 1</h1>
  <h1 class="boxshadow2">박스 그림자 스타일링 예제 2</h1>
  <h1 class="boxshadow3">박스 그림자 스타일링 예제 3</h1>
  <h1 class="boxshadow4">박스 그림자 스타일링 예제 4</h1>
</body>
```

박스 그림자 스타일링 예제 1

박스 그림자 스타일링 예제 2

박스 그림자 스타일링 예제 3

박스 그림자 스타일링 예제 4

{ box-shadow: 수평 그림자(필수) | 수직 그림자(필수) | 그림자 흐림 | 그림자 번짐 | 그림자 색상 | 삼  
입 효과; }

## 예제 7-8 박스에 그림자 효과 주기

```
<head>
  <style>
    img {
      padding: 20px;
      margin: 20px;
    }
    .shadow1 { /* Bottom right coner */
      box-shadow: 5px 5px 10px #000;
    }
    .shadow2 { /* Up right coner */
      box-shadow: 5px -5px 10px #000;
    }
  </style>
</head>
<body>
  
  
</body>
```



## ◆ Box multiple shadows:

```
#example1 {  
  border: 1px solid;  
  padding: 10px;  
  box-shadow: 5px 5px blue, 10px 10px red, 15px 15px green;  
}  
  
#example2 {  
  border: 1px solid;  
  padding: 10px;  
  box-shadow: 5px 5px 8px blue, 10px 10px 8px red, 15px 15px 8px green;  
}
```

**box-shadow: 5px 5px blue, 10px 10px red, 15px 15px green:**

Define multiple shadows.

**box-shadow: 5px 5px 8px blue, 10px 10px 8px red, 15px 15px 8px green:**

Define multiple shadows with blur effect.

# CSS3 속성

---

- ◆ 박스 모델
- ◆ 테두리 속성
- ◆ 박스 속성
- ◆ 레이아웃 속성
- ◆ 표 속성



# CSS Layout - position

## ◆ position 속성

표 7-8 position에 사용되는 속성값

구분	속성값	설명
정적 위치 설정	position: static;	각종 요소를 웹 문서의 흐름에 따라 배치한다.
상대 위치 설정	position: relative;	웹 문서의 정상적인 위치에서 상대적으로 얼마나 떨어져 있는지 표시하여 배치하는 방법이다.
절대 위치 설정	position: absolute;	전체 페이지를 기준으로 top, right, bottom, left의 속성을 이용하여 원하는 위치에 배치하는 방법이다.
고정 위치 설정	position: fixed;	요소의 위치를 '절대 위치 설정'과 똑같은 방법으로 배치하되, 창의 스크롤을 움직여도 사라지지 않고 고정된 위치에 그대로 있다.

## 예제 7-10 정적 위치 vs 상대 위치 설정으로 요소 배치하기

```
<head>
  <style>
    body {
      font-weight: bold;
      font-size: 12pt;
    }
    .sp {
      position: static;
      left: 30px; /* 적용되지 않음 */

      background-color: cyan;
      width: 400px;
      height: 50px;
    }
    .rp1 {
      position: relative;
      left: 30px;
      top: -10px;
      background-color: orange;
      width: 400px;
      height: 50px;
    }
    .rp2 {
      position: relative;
      left: 60px;
      top: 20px;
    }
  </style>
</head>
<body>
  <h1>positioning style2</h1>
  <p class="sp">정적 위치 설정 적용</p>
  <div class="rp1">상대 위치 설정 적용 -
    left 30px, top -10px</div>
  <p class="rp2">상대 위치 설정 적용 -
    left 60px, top 20px</p>
</body>
```

```
background-color: lightgreen;
width: 400px;
height: 50px;
}
</style>
</head>
<body>
  <h1>positioning style2</h1>
  <p class="sp">정적 위치 설정 적용</p>
  <div class="rp1">상대 위치 설정 적용 -
    left 30px, top -10px</div>
  <p class="rp2">상대 위치 설정 적용 -
    left 60px, top 20px</p>
</body>
```

### positioning style2

정적 위치 설정 적용

상대 위치 설정 적용 - left 30px, top -10px

상대 위치 설정 적용 - left 60px, top 20px

## 예제 7-11 절대 위치 설정으로 요소 배치하기

```
<head>
  <style>
    body {
      font-weight: bold;
      font-size: 12pt;
    }
    .ap1 {
      position: absolute;
      left: 30px;
      top: 70px;
      background-color: yellow;
      width: 400px;
      height: 50px;
    }
    .ap2 {
      position: absolute;
      left: 40px;
      top: 90px;
      background-color:
        lightgreen;

      width: 400px;
      height: 50px;
    }
  </style>
</head>
<body>
  <h1>positioning style3</h1>
  <div class="ap1">절대 위치 설정 적용 - left 30px,
    top 70px</div>
  <div class="ap2">절대 위치 설정 적용 - left 40px,
    top 90px</div>
  <div class="rp">상대 위치 설정 적용 - left 50px,
    top 80px</div>
</body>
```

```
.rp {
  position: relative;
  left: 50px;
  top: 80px;
  background-color: cyan;
  width: 400px;
  height: 50px;
}
</style>
</head>
<body>
  <h1>positioning style3</h1>
  <div class="ap1">절대 위치 설정 적용 - left 30px,
    top 70px</div>
  <div class="ap2">절대 위치 설정 적용 - left 40px,
    top 90px</div>
  <div class="rp">상대 위치 설정 적용 - left 50px,
    top 80px</div>
</body>
```

### positioning style3

절대 위치 설정 적용 - left 30px, top 70px

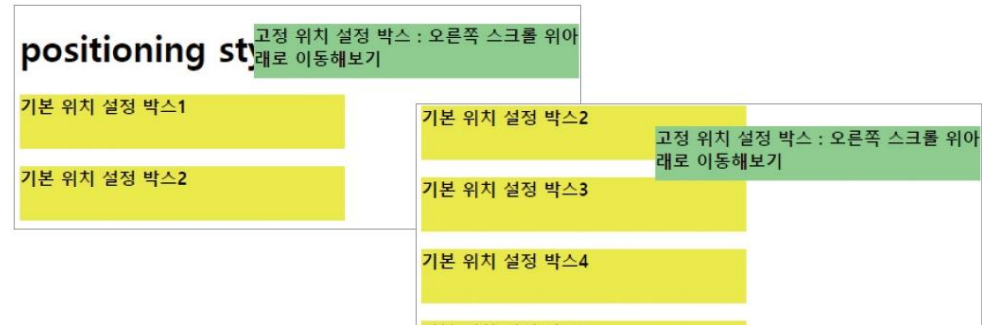
절대 위치 설정 적용 - left 40px, top 90px

상대 위치 설정 적용 - left 50px, top 80px

## 예제 7-12 고정 위치 설정으로 요소 배치하기

```
<head>
  <style>
    body {
      font-weight: bold;
      font-size: 12pt;
    }
    .p {
      background-color: yellow;
      width: 300px;
      height: 50px;
    }
    .fp {
      position: fixed;
      right: 5px;
      top: 5px;
      background-color: lightgreen;
      width: 300px;
      height: 50px;
    }
  </style>
</head>
```

```
<body>
  <h1>positioning style4</h1>
  <p class="p">기본 위치 설정 박스1</p>
  <p class="p">기본 위치 설정 박스2</p>
  <p class="p">기본 위치 설정 박스3</p>
  <p class="p">기본 위치 설정 박스4</p>
  <p class="p">기본 위치 설정 박스5</p>
  <p class="fp">고정 위치 설정 박스 :
    오른쪽 스크롤 위아래로
    이동해보기</p>
</body>
```

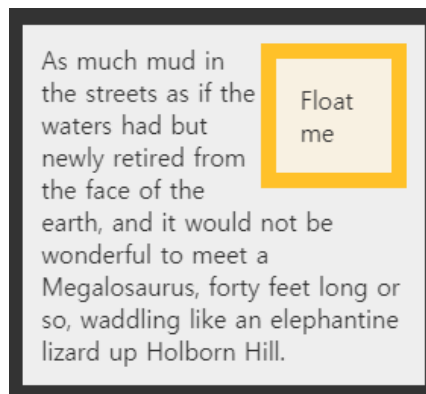
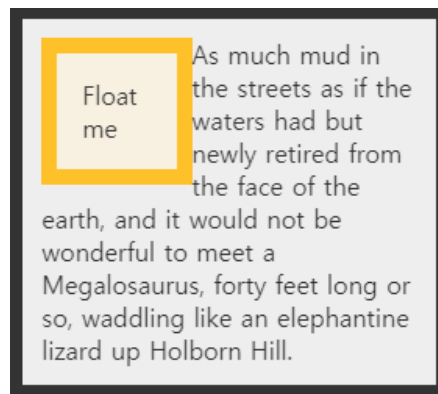
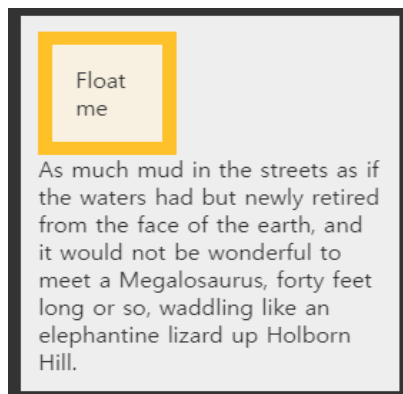


# CSS Layout - float

## ◆ float 속성

표 7-9 float에 사용되는 속성값

속성값	설명
inherit	요소를 감싸는 부모 요소의 float 속성을 상속받는다.
left	요소를 왼쪽으로 떠 있는 상태로 만든다.
right	요소를 오른쪽으로 떠 있는 상태로 만든다.
none	float 속성을 적용하지 않는다(요소를 떠 있지 않게 한다).



### 예제 7-13 float 속성 사용하기

```
<head>
  <style>
    img {
      float: right;
      margin: 0 0 10px 10px;
    }
  </style>
</head>
<body>
  <p>float 속성은 웹 문서의 레이아웃을 설계하는 과정에서 많이 사용하는 속성입니다.</p>
  <p>
    float 속성은 특정 요소를 떠 있게 해줍니다. 여기서 '떠 있다'라는 말의 의미는 특정 요소가 기본
    레이아웃에서 벗어나 웹 문서의 왼쪽이나 오른쪽에 이동하는 것을 말합니다. float 속성은 복잡한 형태의
    레이아웃을 구성하는 데 필요한 핵심 속성으로, 특정 요소가 주변 요소와 자연스럽게 어울리도록 해줍니다.
    주의할 점은 float 속성을 사용할 때 요소의 위치가 고정되면 안 되기 때문에 position 속성의 absolute를
    사용하면 안 됩니다.</p>
</body>
```

float 속성은 웹 문서의 레이아웃을 설계하는 과정에서 많이 사용하는 속성입니다.

float 속성은 특정 요소를 떠 있게 해줍니다. 여기서 '떠 있다'라는 말의 의미는 특정 요소가 기본 레이아웃에서 벗어나 웹 문서의 왼쪽이나 오른쪽에 이동하는 것을 말합니다. float 속성은 복잡한 형태의 레이아웃을 구성하는 데 필요한 핵심 속성으로, 특정 요소가 주변 요소와 자연스럽게 어울리도록 해줍니다. 주의할 점은 float 속성을 사용할 때 요소의 위치가 고정되면 안 되기 때문에 position 속성의 absolute를 사용하면 안 됩니다.



# CSS Layout - clear

## ◆ clear 속성

- float 속성을 적용되지 않도록 할때 사용  
(after you have used a float property on an element)

```
.div3 {  
  float: left;  
  width: 100px;  
  height: 50px;  
  margin: 10px;  
  border: 3px solid #73AD21;  
}  
  
.div4 {  
  border: 1px solid red;  
  clear: left;  
}
```

The `clear` property can have one of the following values:

- none - Allows floating elements on both sides. This is default
- left - No floating elements allowed on the left side
- right - No floating elements allowed on the right side
- both - No floating elements allowed on either the left or the right side
- inherit - The element inherits the clear value of its parent

```
<head>
  <style>
    .div1 {
      float: left;
      width: 100px;
      height: 50px;
      margin: 10px;
      border: 3px solid #73AD21;
    }
    .div2 {
      border: 1px solid red;
    }
    .div3 {
      float: left;
      width: 100px;
      height: 50px;
      margin: 10px;
      border: 3px solid #73AD21;
    }
    .div4 {
      border: 1px solid red;
      clear: left;
    }
  </style>
</head>
```

## float 속성 사용

div1

div2 - float 속성을 사용하여 대상 요소를 웹 문서에 배치하면 그 다음 요소에도 똑같은 속성이 적용됩니다. 하지만 float 속성이 사용되는 것을 원하지 않을 때도 있습니다. 이때 clear 속성을 사용합니다. 다양한 레이아웃 설계할 때에는 float 속성과 clear 속성을 적절히 잘 사용해야 합니다.

## clear 속성 사용

div3

div4 - clear 속성은 float 속성이 적용되는 것을 원하지 않는 요소에 사용하여 float 속성을 초기화시킵니다. float: left;를 사용했다면 clear: left;로, float: right;를 사용했다면 clear: right;로 초기화합니다. float 속성 값이 left 인지 right 인지 상관없이 무조건 초기화하고 싶다면 clear: both;를 사용합니다. 보통 clear: both;를 많이 사용합니다.



<body>

<h2>float 속성 사용</h2>

<div class="div1">div1</div>

<div class="div2">div2 - float 속성을 사용하여 대상 요소를 웹 문서에 배치하면 그 다음 요소에도 똑같은 속성이 적용됩니다. 하지만 float 속성이 사용되는 것을 원하지 않을 때도 있습니다. 이때 clear 속성을 사용합니다. 다양한 레이아웃 설계할 때에는 float 속성과 clear 속성을 적절히 잘 사용해야 합니다.</div>

<h2>clear 속성 사용</h2>

<div class="div3">div3</div>

<div class="div4">div4 - clear 속성은 float 속성이 적용되는 것을 원하지 않는 요소에 사용하여 float 속성을 초기화 시킵니다. float: left;를 사용했다면 clear: left;로, float: right;를 사용했다면 clear: right;로 초기화합니다. float 속성 값이 left 인지 right 인지 상관없이 무조건 초기화하고 싶다면 clear: both;를 사용합니다. 보통 clear: both;를 많이 사용합니다.</div>

</body>

### float 속성 사용

div1

div2 - float 속성을 사용하여 대상 요소를 웹 문서에 배치하면 그 다음 요소에도 똑같은 속성이 적용됩니다. 하지만 float 속성이 사용되는 것을 원하지 않을 때도 있습니다. 이때 clear 속성을 사용합니다. 다양한 레이아웃 설계할 때에는 float 속성과 clear 속성을 적절히 잘 사용해야 합니다.

### clear 속성 사용

div3

div4 - clear 속성은 float 속성이 적용되는 것을 원하지 않는 요소에 사용하여 float 속성을 초기화시킵니다. float: left;를 사용했다면 clear: left;로, float: right;를 사용했다면 clear: right;로 초기화합니다. float 속성 값이 left 인지 right 인지 상관없이 무조건 초기화하고 싶다면 clear: both;를 사용합니다. 보통 clear: both;를 많이 사용합니다.

# CSS Layout - Overflow

## ◆ overflow 속성

이미지가 박스 영역을 벗어남

이미지가 오른쪽 정렬로 되어 있는데, 박스 영역을 벗어났습니다.



overflow: auto; 속성을 사용하여 해결

이미지가 박스 영역을 벗어날 경우에는 overflow 속성을 auto로 설정하여 해결합니다.



## ◆ 속성값 : visible | hidden | scroll | auto

## 예제 7-15 overflow 속성 사용하기

```
<head>
  <style>
    div {
      border: 3px solid #73AD21;
    }
    .img1 {
      float: right;
    }
    .fix {
      overflow: auto;
    }
    .img2 {
      float: right;
    }
  </style>
</head>
<body>
```

<p>이미지가 박스 영역을 벗어남</p>

<div>  
이미지가 오른쪽 정렬로 되어 있는데, 박스 영역을 벗어났습니다.</div>

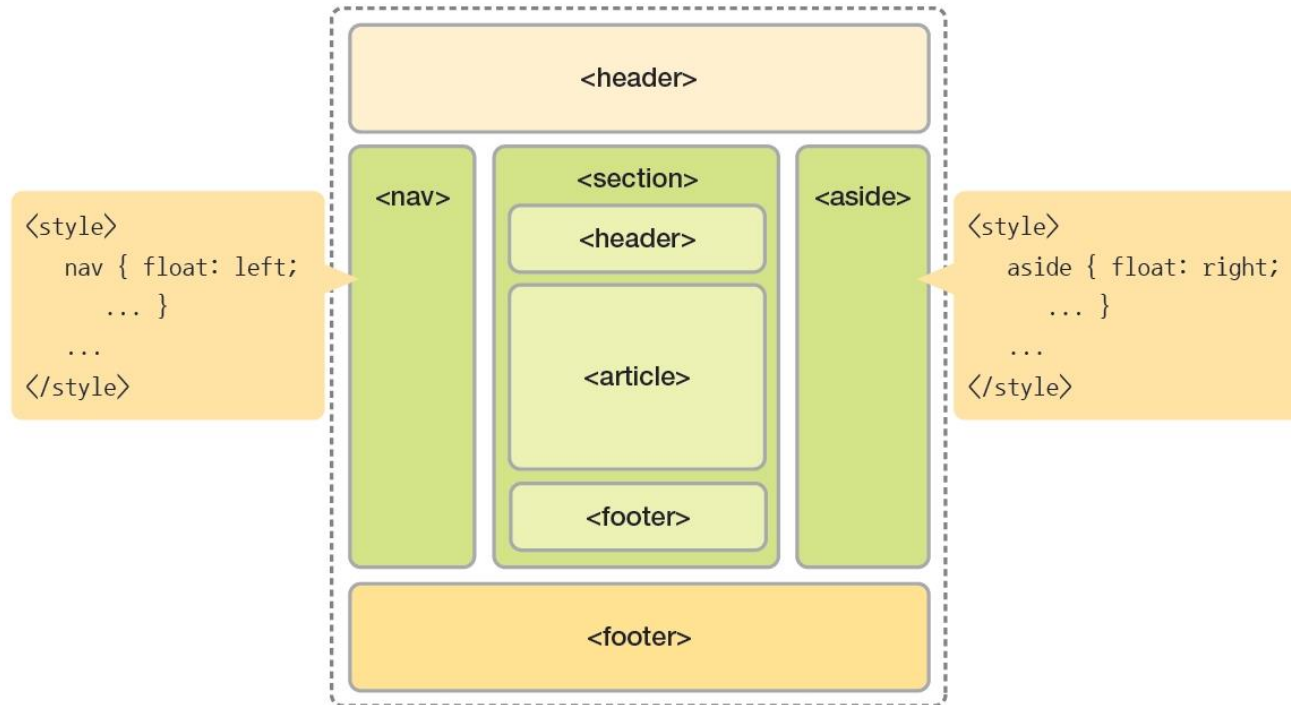
<p style="clear:right">overflow: auto; 속성을 사용하여 해결</p>

<div class="fix">  
이미지가 박스 영역을 벗어날 경우에는 overflow 속성을 auto로 설정하여 해결합니다.</div>

</body>



## ◆ 시멘틱 문서 구조에서 float 사용



## 예제 7-16 시맨틱 문서 구조에 float 속성 사용하기

```
<head>
  <style>
    div { border: 3px solid blue; }
    .fix { overflow: auto; }
    nav { float: left; width: 200px; border: 3px solid #73AD21; }
    section { margin-left: 206px; border: 3px solid red; }
  </style>
</head>
<body>
  <div class="fix">
    <nav>
      <span>목차</span>
      <ul>
        <li><a target="_blank" href="http://www.google.com">Google</a></li>
        <li><a target="_blank" href="http://www.apple.com">Apple</a></li>
        <li><a target="_blank" href="http://www.w3.org">W3C</a></li>
        <li><a target="_blank" href="http://www.oracle.com">Oracle</a></li>
        <li><a target="_blank" href="http://www.adobe.com">Adobe</a></li>
        <li><a target="_blank" href="http://www.amazon.com">Amazon</a></li>
        <li><a target="_blank" href="http://www.Mysql.com">Mysql</a></li>
      </ul>
    </nav>
    <section>
      <span>section 1</span>
      <p>float 속성은 시맨틱 문서 구조에 유용하게 사용할 수 있습니다. 예를 들면 nav나 aside에 float 속성을 추가하면 떠다니는 내비게이션 또는 사이드 바를 만들 수 있습니다.</p>
    </section>
    <section>
      <span>section 2</span>
      <p>시맨틱 문서 구조에 float 속성을 적용할 때는 footer 부분에 적용되지 않도록 해야 합니다.</p>
    </section>
  </div>
</body>
```

목차	section 1
<ul style="list-style-type: none"><li>• <a href="#">Google</a></li><li>• <a href="#">Apple</a></li><li>• <a href="#">W3C</a></li><li>• <a href="#">Oracle</a></li><li>• <a href="#">Adobe</a></li><li>• <a href="#">Amazon</a></li><li>• <a href="#">Mysql</a></li></ul>	float 속성은 시맨틱 문서 구조에 유용하게 사용할 수 있습니다. 예를 들면 nav나 aside에 float 속성을 추가하면 떠다니는 내비게이션 또는 사이드 바를 만들 수 있습니다.
	section 2
	시맨틱 문서 구조에 float 속성을 적용할 때는 footer 부분에 적용되지 않도록 해야 합니다.

# CSS z-index Property

## ◆ z-index 속성

- Specifies the **stack order** of an element (z-index 속성값이 작을수록 아래에 쌓임)
- Only works on **positioned elements**  
(position: absolute, position: relative, position: fixed, or position: sticky)

## ◆ 속성값: auto | *number* | initial | inherit;

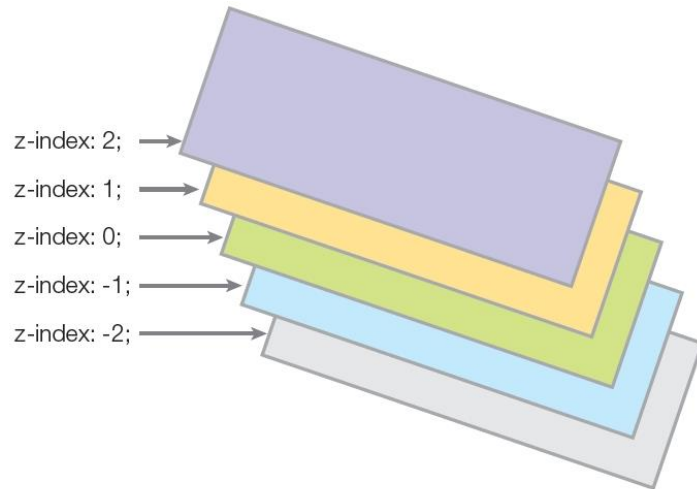
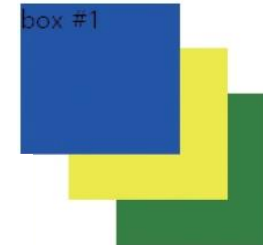


그림 7-6 z-index 속성값에 따른 배치

### 예제 7-17 z-index 속성값에 따라 요소들이 쌓이는 순서 확인하기

```
<head>
  <style>
    #box1 {
      position: absolute;
      top: 0px;
      left: 0px;
      width: 100px;
      height: 100px;
      background: blue;
      z-index: 3;
    }
    #box2 {
      position: absolute;
      top: 30px;
      left: 30px;
      width: 100px;
      height: 100px;
      background: yellow;
      z-index: 2;
    }
    #box3 {
      position: absolute;
      top: 60px;
      left: 60px;
      width: 100px;
      height: 100px;
      background: green;
      z-index: 1;
    }
  </style>
</head>
```

```
<body>
  <div id="box1">box #1</div>
  <div id="box2">box #2</div>
  <div id="box3">box #3</div>
</body>
```



# CSS3 속성

---

- ◆ 박스 모델
- ◆ 테두리 속성
- ◆ 박스 속성
- ◆ 레이아웃 속성
- ◆ **표 속성**



# CSS table-layout Property

## ◆ table-layout 속성

표 7-10 table-layout에 사용되는 속성값

속성값	설명
table-layout: auto;	내용 분량에 따라 셀 너비가 자동으로 조절된다. (기본값)
table-layout : fixed;	내용 분량과 관계없이 셀 너비를 고정한다.
table-layout: initial;	변경된 테이블 레이아웃을 기본값 상태(auto)로 설정한다.
Table-layout: inherit;	부모 요소의 값을 상속 받아 셀 너비를 결정한다.

## 예제 7-18 표의 레이아웃 설정하기

```
<head>
  <style>
    td, th {
      border: 1px solid black;
    }
    #tb1 {
      border: 2px solid red;
      table-layout: auto;
    }
    #tb2 {
      border: 3px dotted teal;
      background-color: yellow;
      table-layout: fixed;
    }
  </style>
</head>
<body>
  <h2>table layout auto 예제</h2>
  <table id="tb1">
    <tr>
      <th>table layout auto</th>
      <td>내용 분량에 따라서 자동으로 조절</td>
    </tr>
  </table>
```

```
<h2>table layout fixed 예제</h2>
  <table id="tb2" width="250px">
    <tr>
      <th>table layout fixed</th>
      <td>내용 분량과 상관 없이 고정</td>
    </tr>
  </table>
</body>
```

### table layout auto 예제

table layout auto	내용 분량에 따라서 자동으로 조절
-------------------	--------------------

### table layout fixed 예제

table layout fixed	내용 분량과 상관 없이 고정
--------------------	-----------------

## 예제 7-20 바깥 테두리와 셀 테두리 사이 간격 조정하기

```
<head>
  <style>
    td, th {
      border: 1px solid red;
    }
    #tb1 {
      border: 2px solid green;
      border-spacing: 10px; /* 상하좌우 */
    }
    #tb2 {
      border: 3px solid maroon;
      background-color: aqua;
      border-spacing: 20px 40px; /* 첫번째 값: 좌우, 두번째 값: 상하 */
    }
  </style>
</head>
<body>
  <table style id="tb1" width="350px">
    <tr>
      <th>table border-spacing</th>
      <td>10px</td>
    </tr>
  </table>
  <p></p>
  <table style id="tb2" width="350px">
    <tr>
      <th>table border-spacing</th>
      <td>20px 40px</td>
    </tr>
  </table>
</body>
```

table border-spacing	10px
table border-spacing	20px 40px

### 예제 7-21 빈 셀을 보이게 하거나 숨기기

```
<head>
  <style>
    td, th {
      border: 1px solid blue;
    }
    #tb1 {
      border-collapse: separate;
      empty-cells: hide;
    }
    #tb2 {
      border-collapse: separate;
      empty-cells: show;
    }
  </style>
</head>
<body>
  <table id="tb1" border="1" width="300px">
    <tr>
      <td>국어</td>
      <td>영어</td>
      <td></td>
    </tr>
    <tr>
      <td>수학</td>
      <td></td>
    </tr>
  </table>
```

```
<p></p>
<table id="tb2" border="1" width="300px">
  <tr>
    <td>국어</td>
    <td>영어</td>
    <td></td>
  </tr>
  <tr>
    <td>수학</td>
    <td></td>
  </tr>
</table>
</body>
```

국어	영어	
수학		

국어	영어	
수학		

## 예제 7-22 캡션 삽입하기

```
<head>
  <style>
    td, th {
      border: 2px solid black;
    }
    #c1 {
      border: 3px solid blue;
      caption-side: top;
      border-collapse: collapse;
    }
    #c2 {
      border: 3px solid red;
      caption-side: bottom;
      border-collapse: collapse;
    }
  </style>
</head>
<body>
  <table id="c1" border="1" width="300px">
    <caption>[table 1-1] Korea University</caption>
    <tr>
      <th>University</th>
      <th>Contact</th>
      <th>Country</th>
    </tr>
```

```

    <tr>
      <td>서울대학교</td>
      <td>홍현성</td>
      <td>KOREA</td>
    </tr>
  </table>
  <p></p>
  <table id="c2" border="1" width="300px">
    <caption>[table 1-2] USA University</caption>
    <tr>
      <th>University</th>
      <th>Contact</th>
      <th>Country</th>
    </tr>
    <tr>
      <td>Havard</td>
      <td>Jackie</td>
      <td>USA</td>
    </tr>
  </table>
</body>
```

[table 1-1] Korea University

University	Contact	Country
서울대학교	홍현성	KOREA

University	Contact	Country
Havard	Jackie	USA

[table 1-2] USA University

# CSS3 속성

---

- ◆ 박스 모델
  - 박스 모델 개념, 박스 크기 설정
- ◆ 테두리 속성
  - 두께, 색상, 스타일
- ◆ 박스 속성
  - 둥근 모서리, 박스 그림자
- ◆ 레이아웃 속성
  - position, float, z-index
- ◆ 표 속성
  - 레이아웃, 테두리, 빈 셀 처리, 캡션