

CSS3 속성

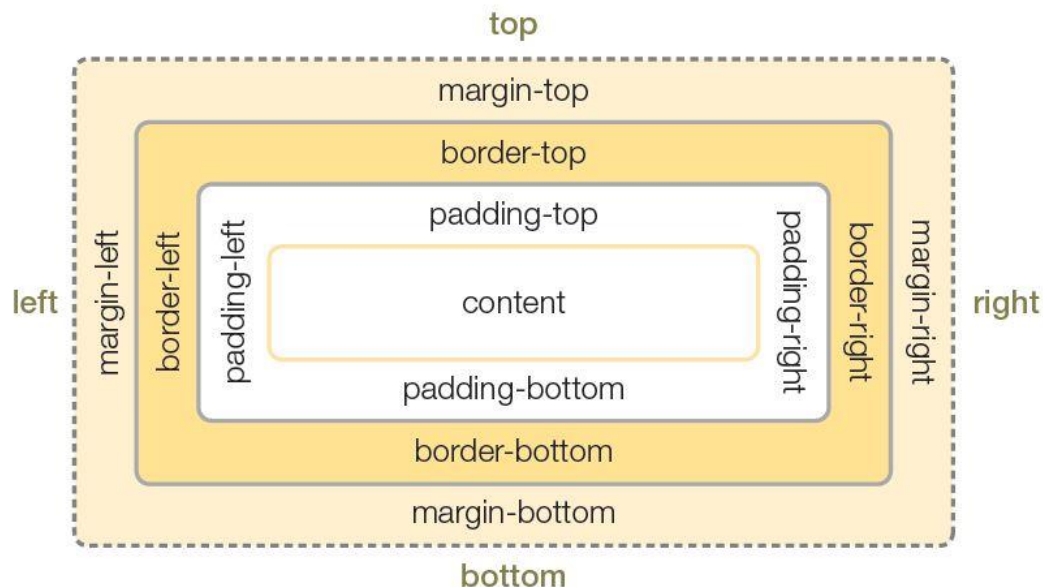
박스 모델

❖ 박스 모델

- 웹 문서에 텍스트, 이미지, 테이블 등의 요소를 배치하기 위해 사용
- 웹 문서의 전체 레이아웃을 정의
- 각종 요소들을 원하는 위치에 배치

❖ 박스의 속성

- content : 실제 내용이 표현되는 곳
- padding : 콘텐츠와 테두리 사이의 여백
- border : 박스의 테두리 두께
- margin : 테두리와 박스의 최종 경계 사이의 여백



박스 모델

❖ 박스 모델

```
<head>
  <style>
    div {
      background-color: yellow;
      width: 300px;
      padding: 25px;
      border: 15px solid navy;
      margin: 25px;
    }
  </style>
</head>
<body>
  <p>CSS3 박스 모델은 content, padding, border, margin으로 구성되어 있다.</p>
  <div>박스 모델의 padding, border, margin 속성의 기본 값은 0이며, 상하좌우 네 가지
  방향을 top, bottom, left,
  right를 이용하여 지정할 수 있습니다.</div>
</body>
```

CSS3 박스 모델은 content, padding, border, margin으로 구성되어 있다.

박스 모델의 padding, border, margin 속성의 기본 값은 0이며, 상하좌우 네 가지 방향을 top, bottom, left, right를 이용하여 지정할 수 있습니다.

박스 모델

❖ 박스의 크기 설정

- Content
 - 실제 내용이 표현되는 곳
 - 속성: width(너비 지정), height(높이 지정)

속성	설명
value	실제로 측정된 데이터 값이다.
min, max	데이터가 인식하는 최솟값과 최댓값이다. 기본값은 0~1이다.
low, high	허용되는 범위의 최솟값과 최댓값이다. low~high 값은 항상 min~max 값 범위 내에 있다.

박스 모델

❖ 박스의 내용 영역 크기 지정

```
<head>
<style>
  p {
    background-color: yellow;
    color: red;
    font-weight: bold;
    font-size: 16pt;
  }
  p.c1 {
    width: 700px;
    height: 80px;
    color: green;
  }
  p.c2 {
    width: 50%;
    height: 150%;
    color: purple;
  }
  p.c3 {
    width: 10cm;
    height: 3cm;
    color: blue;
  }
</style>
</head>
```

박스 모델

❖ 박스의 내용 영역 크기 지정

```
</head>
<body>
  <p>박스 모델의 내용 영역 크기 지정</p>
  <p class="c1">(1) 박스 모델의 크기를 픽셀(px) 단위로 지정</p>
  <p class="c2">(2) 박스 모델의 크기를 퍼센트(%) 단위로 지정</p>
  <p class="c3">(3) 박스 모델의 크기를 센티미터(cm) 단위로 지정</p>
</body>
```

박스 모델의 내용 영역 크기 지정

(1) 박스 모델의 크기를 픽셀(px) 단위로 지정

(2) 박스 모델의 크기를 퍼센트(%) 단
위로 지정

(3) 박스 모델의 크기를 센티미터(cm)
단위로 지정

박스 모델

❖ 박스의 크기 설정

- padding, margin
 - padding : 박스의 안쪽 여백 설정
 - Margin : 박스의 바깥쪽 여백 설정

속성값	설명
수치	여백을 픽셀(px), 포인트(pt), 센티미터(cm) 같은 수치 단위로 지정한다.
백분율	여백을 부모 요소를 기준으로 하여 백분율로 지정한다.
auto	여백을 웹 브라우저가 자동으로 지정한다. 기본값이다.

박스 모델

❖ 박스의 안쪽 여백과 바깥쪽 여백 지정

```
<head>
  <style>
    p {
      background-color: yellow;
      color: red;
      font-weight: bold;
      font-size: 16pt;
    }
    p.pad {
      color: purple;
      padding: 20px;
    }
    p.mar {
      color: green;
      margin: 30px;
    }
    p.mp {
      color: blue;
      padding: 5%;
      margin: 5%;
    }
  </style>
</head>
```


박스 모델

❖ 박스의 안쪽 여백과 바깥쪽 여백 지정

```
<body>  
  <p>박스의 안쪽 여백과 바깥쪽 여백 지정</p>  
  <p class="pad">(1) 안쪽 여백 지정 - padding 20px</p>  
  <p class="mar">(2) 바깥쪽 여백 지정 - margin 30px</p>  
  <p class="mp">(3) 안쪽, 바깥쪽 여백 지정 - padding 5%, margin 5%</p>  
</body>
```

박스의 안쪽 여백과 바깥쪽 여백 지정

(1) 안쪽 여백 지정 - padding 20px

(2) 바깥쪽 여백 지정 - margin 30px

(3) 안쪽, 바깥쪽 여백 지정 - padding 5%, margin 5%

박스 모델

❖ 박스의 크기 설정

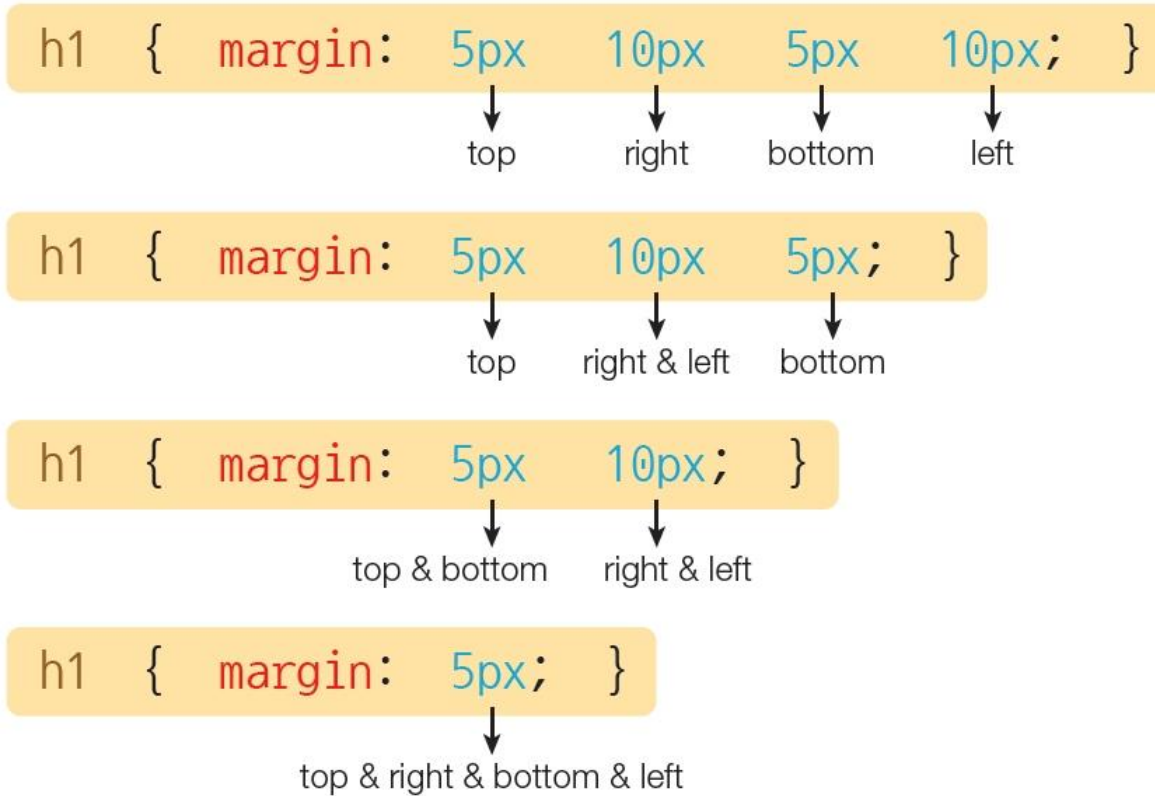
- 네 방향 여백 크기 설정

형식	설명
padding-top: 속성값;	위쪽 패딩
padding-bottom: 속성값;	아래쪽 패딩
padding-right: 속성값;	오른쪽 패딩
padding-left: 속성값;	왼쪽 패딩
margin-top: 속성값;	위쪽 마진
margin-bottom: 속성값;	아래쪽 마진
margin-right: 속성값;	오른쪽 마진
margin-left: 속성값;	왼쪽 마진

박스 모델

❖ 박스의 크기 설정

- 속성값의 개수에 따라 적용되는 위치



박스 모델

❖ 박스의 여백을 네 방향으로 지정하기

```
<head>
  <style>
    body {
      font-weight: bold;
      font-size: 16pt;
    }
    .mp1 {
      background-color: aqua;
      color: red;
      margin-bottom: 40px;
      padding-left: 50px;
    }
    .mp2 {
      background-color: silver;
      color: green;
      margin: 20px;
      padding: 10px, 20px;
    }
    .mp3 {
      background-color: gray;
      color: purple;
      margin: 50px, 15px, 20px;
      padding: 20px;
    }
  </style>
</head>
```

박스 모델

❖ 박스의 여백을 네 방향으로 지정하기

```
<body style="background-color: lightyellow;">  
  <p>박스 모델의 네 방향 여백 지정</p>  
  <p class="mp1">mp1</p>  
  <p class="mp2">mp2</p>  
  <p class="mp3">mp3</p>  
</body>
```

박스 모델의 네 방향 여백 지정

mp1

mp2

mp3

테두리 속성

❖ 테두리 두께

- border-width
 - 테두리 두께 설정
 - top, bottom, left, right를 이용하여 네 방향으로 설정 가능

속성값	설명
수치	테두리의 두께를 픽셀(px), 포인트(pt), 센티미터(cm) 같은 단위로 지정한다.
thin	얇은(1px) 두께의 테두리를 지정한다.
medium	중간(3px) 두께의 테두리를 지정한다. 기본값이다.
thick	굵은(5px) 두께의 테두리를 지정한다.

테두리 속성

❖ 테두리 색상

- border-color
 - 테두리 색상 지정

색상	색상 이름	16진수 값	10진수 값
	black	#000000	0, 0, 0
	silver	#C0C0C0	192, 192, 192
	gray	#808080	128, 128, 128
	white	#FFFFFF	255, 255, 255
	maroon	#800000	128, 0, 0
	red	#FF0000	255, 0, 0
	purple	#800080	128, 0, 128
	fuchsia	#FF00FF	255, 0, 255
	green	#008000	0, 128, 0
	lime	#00FF00	0, 255, 0
	olive	#808000	128, 128, 0
	yellow	#FFFF00	255, 255, 0
	navy	#000080	0, 0, 128
	blue	#0000FF	0, 0, 255
	teal	#008080	0, 128, 128
	aqua	#00FFFF	0, 255, 255

테두리 속성

❖ 테두리 스타일

- border-style
 - 테두리 스타일 설정

속성값	설명
none	테두리가 나타나지 않는다. 기본값이다.
hidden	테두리를 감춘다.
dotted	테두리를 점선으로 지정한다.
dashed	테두리를 파선으로 지정한다.
solid	테두리를 실선으로 지정한다.

속성값	설명
double	테두리를 이중선으로 지정한다.
groove	테두리를 오목한 선으로(홈이 파인 듯 입체적으로) 지정한다.
ridge	테두리를 볼록한 선으로(튀어나온 듯 입체적으로) 지정한다.
inset	테두리의 안쪽이 오목한 선으로 지정한다.
outset	테두리의 안쪽이 볼록한 선으로 지정한다.

테두리 속성

❖ 테두리 스타일

```
<head>
  <style>
    body { font-family: consolas; }
    h3.none { border-style: none; }
    h3.hidden { border-style: hidden; }
    h3.dotted { border-style: dotted; }
    h3.dashed { border-style: dashed; }
    h3.solid { border-style: solid; }
    h3.double { border-style: double; }
    h3.groove { border-style: groove; }
    h3.ridge { border-style: ridge; }
    h3.inset { border-style: inset; }
    h3.outset { border-style: outset; }
    h3.mix { border-style: dotted dashed solid double; }
  </style>
</head>
```

테두리 속성

❖ 테두리 스타일

```
<body>
  <h3 class="none">no border</h3>
  <h3 class="hidden">hidden border</h3>
  <h3 class="dotted">dotted border</h3>
  <h3 class="dashed">dashed border</h3>
  <h3 class="solid">solid border</h3>
  <h3 class="double">double border</h3>
  <h3 class="groove">groove border</h3>
  <h3 class="ridge">ridge border</h3>
  <h3 class="inset">inset border</h3>
  <h3 class="outset">outset border</h3>
  <h3 class="mix">mix border</h3>
</body>
```

no border

hidden border

dotted border

dashed border

solid border

double border

groove border

ridge border

inset border

outset border

mix border

테두리 속성

❖ 테두리 스타일

```
<head>
  <style>
    body {
      font-weight: bold;
      font-size: 12pt;
      font-family: consolas;
    }
    .bt1 {
      background-color: aqua;
      color: red;
      margin-bottom: 15px;
      padding-left: 50px;
      border-style: dotted;
      border-width: thick;
      border-color: green;
    }
    .bt2 {
      background-color: silver;
      color: green;
      margin: 20px;
      padding: 5px 10px;
      border-style: solid;
      border-width: 1px 3px 5px 3px;
      border-color: navy red fuchsia black;
    }
  </style>
</head>
```

테두리 속성

❖ 테두리 스타일

```
.bt3 {
    background-color: gray;
    color: purple;
    margin: 50px 15px 20px;
    padding: 20px;
    border-style: dashed double;
    border-width: 5px 0px;
    border-color: red;
}
.bt4 {
    border-top: 5px solid red;
    border-bottom: 5px solid red;
}
</style>
</head>
<body>
    <p class="bt1">박스 테두리 지정 - 1</p>
    <p class="bt2">박스 테두리 지정 - 2</p>
    <p class="bt3">박스 테두리 지정 - 3</p>
    <p class="bt4">박스의 상하좌우 테두리는 border-top,
    border-right, border-bottom, border-left라는 속성을
    이용해 지정할 수 있습니다.</p>
</body>
```

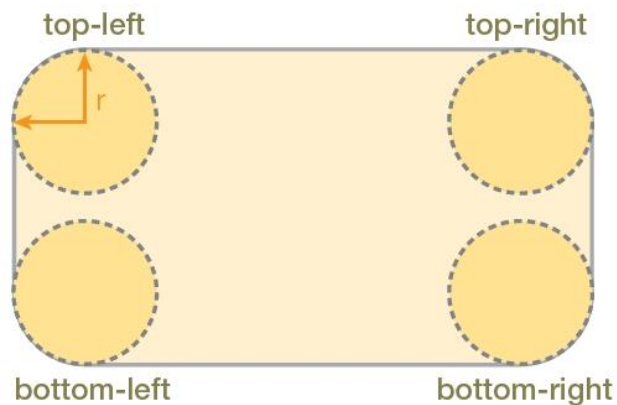


박스 속성

❖ 둥근 모서리

- border-radius
- 테두리의 모서리를 둥글게 설정

형식	설명
border-radius: 속성값;	네 개의 모서리 모두 둥글게 한다.
border-top-left-radius: 속성값;	상단 왼쪽 모서리를 둥글게 한다.
border-top-right-radius: 속성값;	상단 오른쪽 모서리를 둥글게 한다.
border-bottom-right-radius: 속성값;	하단 오른쪽 모서리를 둥글게 한다.
border-bottom-left-radius: 속성값;	하단 왼쪽 모서리를 둥글게 한다.



박스 속성

❖ 둥근 모서리

```
<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;
    }
  </style>
</head>
```

박스 속성

❖ 둥근 모서리

```
.br3 {  
    background-color: teal;  
    color: aqua;  
    border-style: dashed;  
    border-width: 5px;  
    border-color: red;  
    border-radius: 5px 15px 25px 35px;  
}  
.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>
```

둥근 모서리 스타일링 예제 1

둥근 모서리 스타일링 예제 2

둥근 모서리 스타일링 예제 3

둥근 모서리 스타일링 예제 4

박스 속성

❖ 박스 그림자

- box-shadow
 - 박스에 그림자 효과 적용

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

- 수평 그림자(h-shadow): 그림자의 수평 거리 지정
- 수직 그림자(v-shadow): 그림자의 수직 거리 지정
- 그림자 흐림(blur): 그림자의 흐림 정도 지정
- 그림자 번짐(spread): 그림자의 번짐 정도 지정
- 그림자 색상(color): 그림자의 색상 지정
- 삽입 효과(inset): 박스 외부로 표현되는 그림자를 박스 안쪽으로 표현하는 효과

박스 속성

❖ 박스 그림자

```
<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>
```

박스 속성

❖ 박스 그림자

```
<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

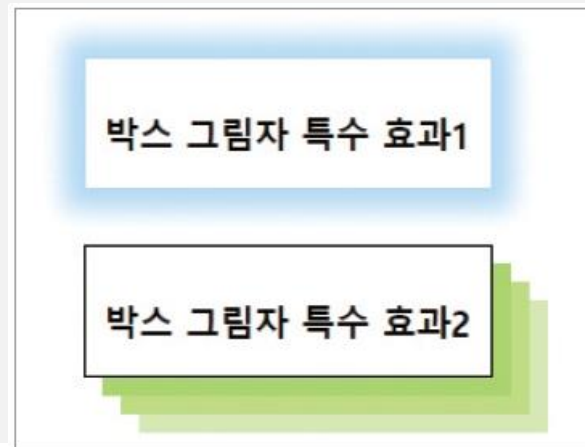
박스 그림자 스타일링 예제 3

박스 그림자 스타일링 예제 4

박스 속성

❖ 박스 그림자

```
<head>
  <style>
    div {
      width: 200px;
      height: 50px;
      padding: 10px;
      margin: 30px;
    }
    #bs1 {
      border: 1px solid rgba(102,186,255,0.4);
      box-shadow: 0px 0px 20px 10px rgba(102,186,255,0.4);
    }
    #bs2 {
      border: 1px solid black;
      box-shadow: 10px 10px #BCE55C, 20px 20px #CEF279, 30px 30px #E4F7BA;
    }
  </style>
</head>
<body>
  <div id="bs1">
    <h3>박스 그림자 특수 효과1</h3>
  </div>
  <div id="bs2">
    <h3>박스 그림자 특수 효과2</h3>
  </div>
</body>
```



박스 속성

❖ 박스 그림자

```
<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>
```



레이아웃 속성

❖ position 속성

- position
 - 텍스트, 이미지, 표 등의 요소를 웹 문서에 배치할 때 사용하는 속성

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.

레이아웃 속성

❖ position 속성

- 정적 위치 설정
 - 텍스트, 이미지, 표 등을 웹 문서의 흐름에 따라 배치하는 방법
 - 블록 요소는 위에서 아래로 쌓이고, 인라인 요소는 같은 줄에 순서대로 배치

레이아웃 속성

❖ position 속성

```
<head>
  <style>
    body {
      font-weight: bold;
      font-size: 12pt;
    }
    .sp1 {
      position: static;
      top: 100px;    /* 적용되지 않음 */
      background-color: cyan;
      width: 400px;
      height: 50px;
    }
    .sp2 {
      position: static;
      left: 30px;    /* 적용되지 않음 */
      background-color: orange;
      width: 400px;
      height: 50px;
    }
  </style>
</head>
```

레이아웃 속성

❖ position 속성

```
.sp3 {  
    background-color: lightgreen;  
    width: 400px;  
    height: 50px;  
}  
</style>  
</head>  
<body>  
    <h1>positioning style1</h1>  
    <p class="sp1">정적 위치 설정 적용1</p>  
    <div class="sp2">정적 위치 설정 적용2</div>  
    <p class="sp3">기본 위치 설정</p>  
</body>
```

positioning style1

정적 위치 설정 적용1

정적 위치 설정 적용2

기본 위치 설정

레이아웃 속성

❖ position 속성

- 상대 위치 설정
 - 각종 요소가 웹 문서의 정적 위치값에서 상대적으로 얼마나 떨어져 있는지 표시하여 배치하는 방법

레이아웃 속성

❖ 상대 위치 설정

```
<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;
    }
  </style>
</head>
```

레이아웃 속성

❖ 상대 위치 설정

```
.rp2 {  
    position: relative;  
    left: 60px;  
    top: 20px;  
    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

레이아웃 속성

❖ position 속성

- 절대 위치 설정
 - 웹 문서의 흐름과는 상관없이 전체 페이지를 기준으로 top, right, bottom, left의 속성을 이용하여 원하는 위치에 배치시키는 방법

레이아웃 속성

❖ 절대 위치 설정

```
<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>
```

레이아웃 속성

❖ 절대 위치 설정

```
.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

레이아웃 속성

❖ position 속성

- 고정 위치 설정
 - 창의 스크롤을 움직여도 사라지지 않고 고정된 위치에 그대로 있음

레이아웃 속성

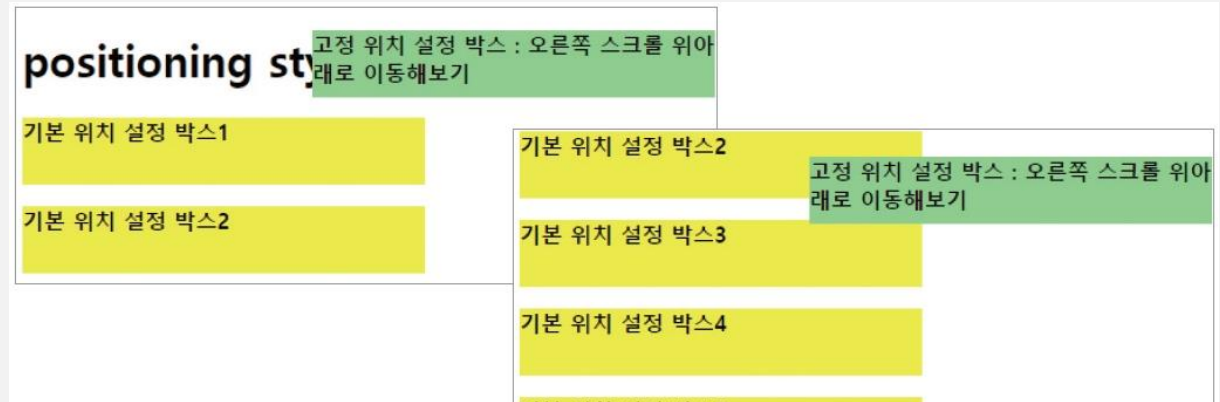
❖ 고정 위치 설정

```
<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>
```


레이아웃 속성

❖ 고정 위치 설정

```
<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>
```



레이아웃 속성

❖ float 속성

- float
 - 화면을 구성하는 요소 간의 관계를 고려하여 각 요소를 배치하는 방법

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).

레이아웃 속성

❖ float 속성

```
<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.



레이아웃 속성

❖ float 속성

- clear
 - float 속성이 적용되지 않도록 할 때 사용

레이아웃 속성

❖ float 속성

```
<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.

레이아웃 속성

❖ float 속성

```
<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 때
도 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 요소
에 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.</div>
</body>
```

레이아웃 속성

❖ float 속성

- overflow
 - auto로 설정하면 이미지가 박스 영역을 벗어나는 현상을 해결할 수 있음

레이아웃 속성

❖ float 속성

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


레이아웃 속성

❖ float 속성

```
<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>
```

이미지가 박스 영역을 벗어남

이미지가 오른쪽 정렬로 되어 있는데, 박스 영역을 벗어났습니다.



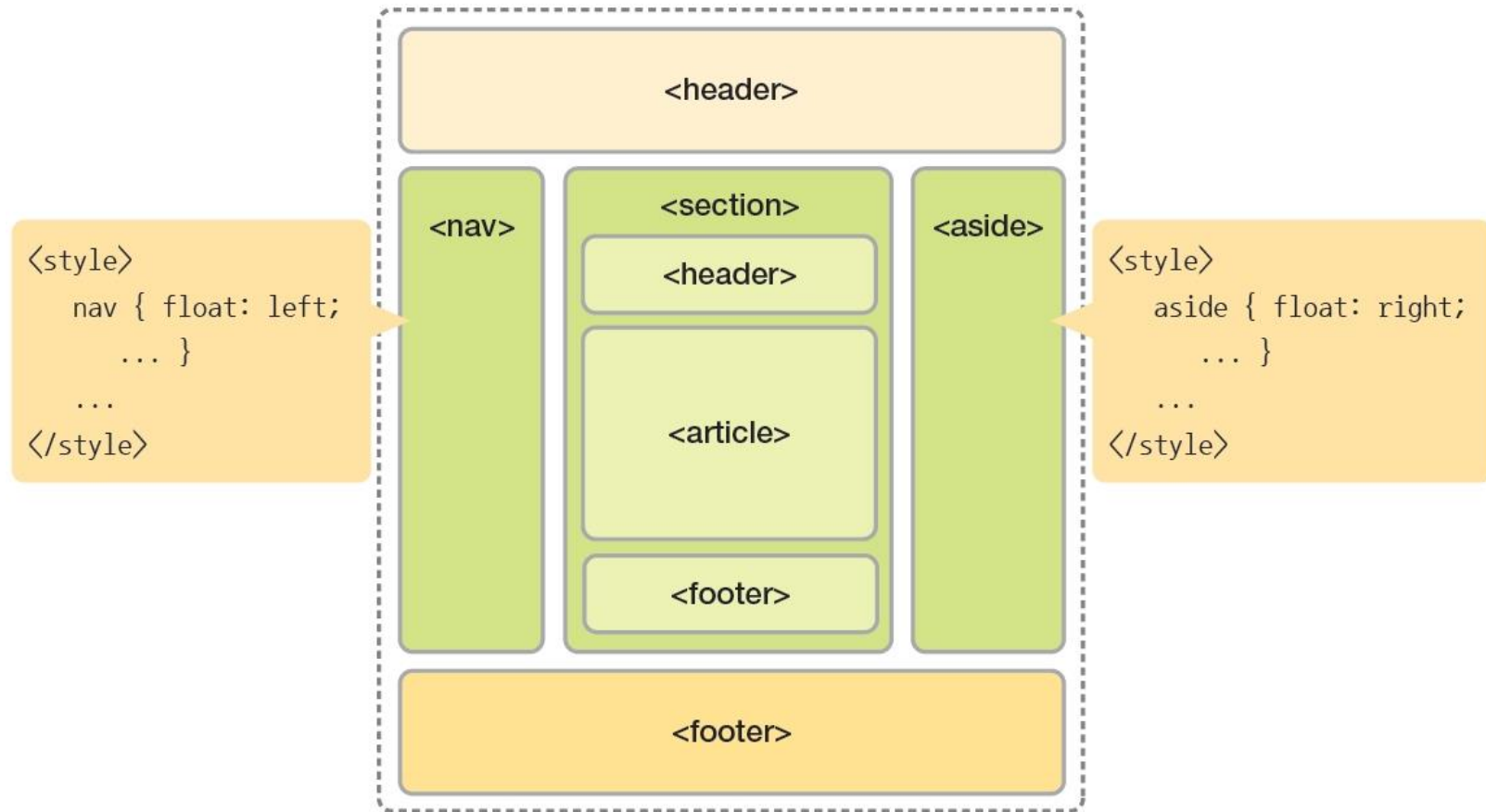
overflow: auto; 속성을 사용하여 해결

이미지가 박스 영역을 벗어날 경우에는 overflow 속성을 auto로 설정하여 해결합니다.



레이아웃 속성

❖ 시맨틱 문서 구조에서 float 속성의 사용



레이아웃 속성

❖ 시맨틱 문서 구조에서 float 속성의 사용

```
<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>
```

레이아웃 속성

❖ 시맨틱 문서 구조에서 float 속성의 사용

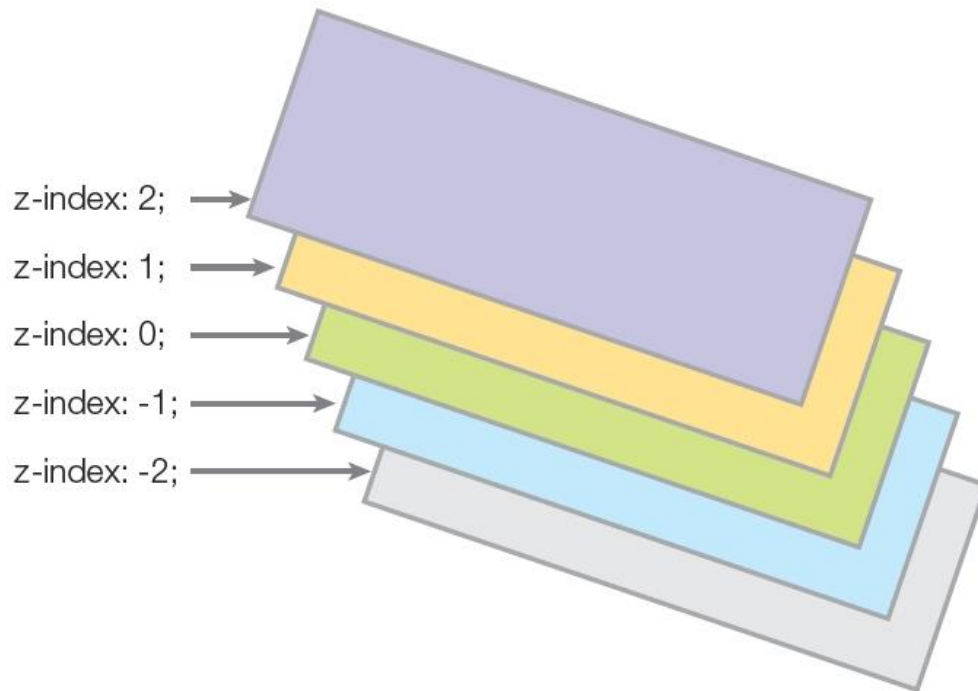
```
<section>
  <span>section 1</span>
  <p>float 속성은 시맨틱 문서 구조에 유용하게 사용할 수 있습니다.
  예를 들면 nav나 aside에 float 속성을 추가하면 떠다니는 내비게이션 또는
  사이드 바를 만들 수 있습니다.</p>
</section>
<section>
  <span>section 2</span>
  <p>시맨틱 문서 구조에 float 속성을 적용할 때는 footer 부분에 적용되지 않도록
  해야 합니다.</p>
</section>
</div>
</body>
```

목차	section 1
<ul style="list-style-type: none">• Google• Apple• W3C• Oracle• Adobe• Amazon• Mysql	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.

레이아웃 속성

❖ z-index

- 한 요소 위에 다른 요소를 쌓을 때 사용
- z-index 속성값이 작을수록 아래에 쌓임



❖ table-layout

- 셀 안 내용의 양에 따라 셀 너비를 조절

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.

❖ table-layout

```
<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>
```

❖ table-layout

```
<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 상 관 없이 고정
-----------------------	---------------------

표 속성

❖ 표 테두리

```
<head>
  <style>
    td, th {
      border: 2px solid black;
    }
    #tb1 {
      border: 3px solid red;
      background-color: yellow;
      border-collapse: separate;
      table-layout: auto;
    }
    #tb2 {
      border: 3px solid red;
      background-color: yellow;
      border-collapse: collapse;
      table-layout: auto;
    }
  </style>
</head>
```

표 속성

❖ 표 테두리

```
<body>
  <table style id="tb1" width="350px">
    <tr>
      <th>table border-collapse</th>
      <td>separate 적용</td>
    </tr>
  </table>
  <p></p>
  <table style id="tb2" width="350px">
    <tr>
      <th>table border-collapse</th>
      <td>collapse 적용</td>
    </tr>
  </table>
</body>
```

table border-collapse	separate 적용
table border-collapse	collapse 적용

표 속성

❖ 표 테두리

```
<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>
```

표 속성

❖ 표 테두리

```
<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>
```

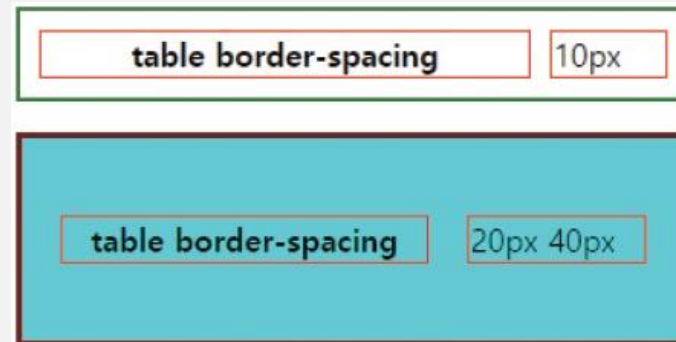


표 속성

❖ 내용 정렬, 빈 셀 처리, 캡션 위치

```
<head>
  <style>
    td, th {
      border: 1px solid blue;
    }
    #tb1 {
      border-collapse: separate;
      empty-cells: hide;
    }
    #tb2 {
      border-collapse: separate;
      empty-cells: show;
    }
  </style>
</head>
```

표 속성

❖ 내용 정렬, 빈 셀 처리, 캡션 위치

```
<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		

표 속성

❖ 내용 정렬, 빈 셀 처리, 캡션 위치

```
<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>
```

표 속성

❖ 내용 정렬, 빈 셀 처리, 캡션 위치

```
<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

[table 1-2] USA University		
University	Contact	Country
Havard	Jackie	USA