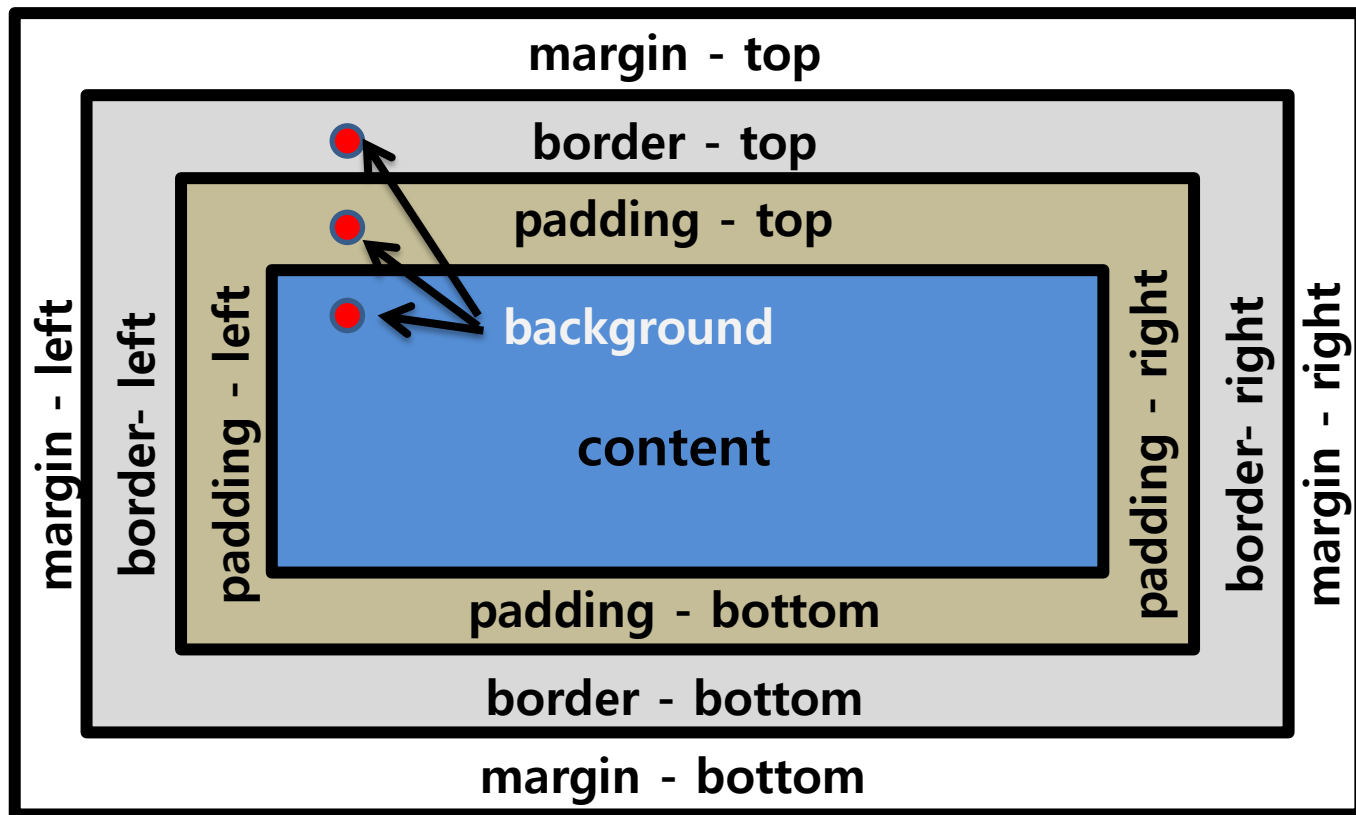


BOX MODEL

1 박스 모델

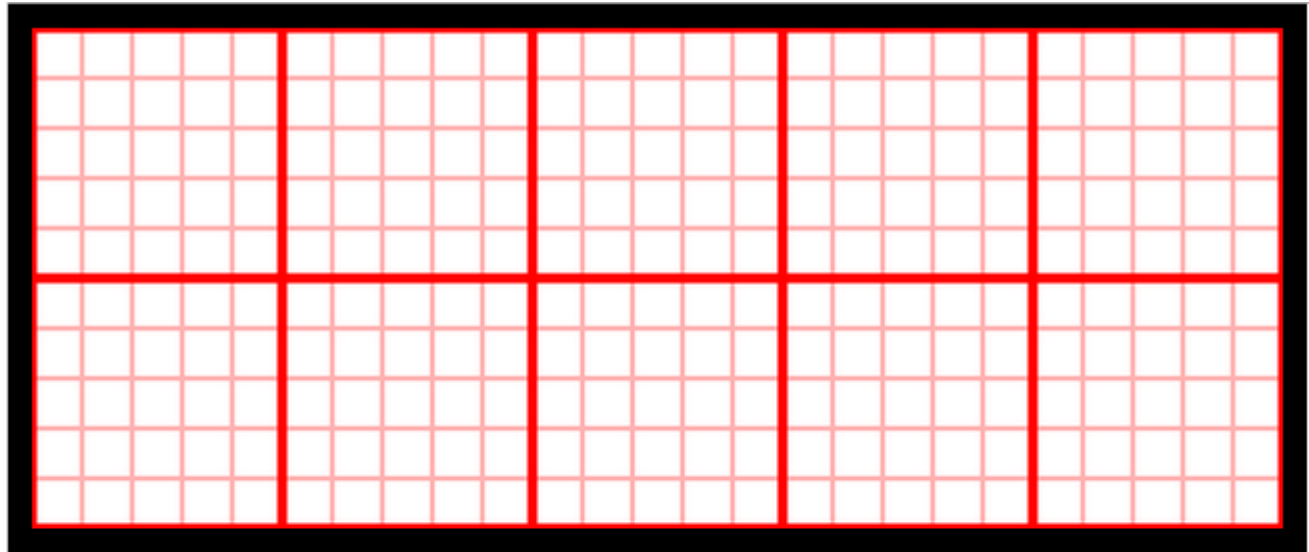


p{width:500px} 500px은 content 사이즈
이미지 배경은 border까지 표시.

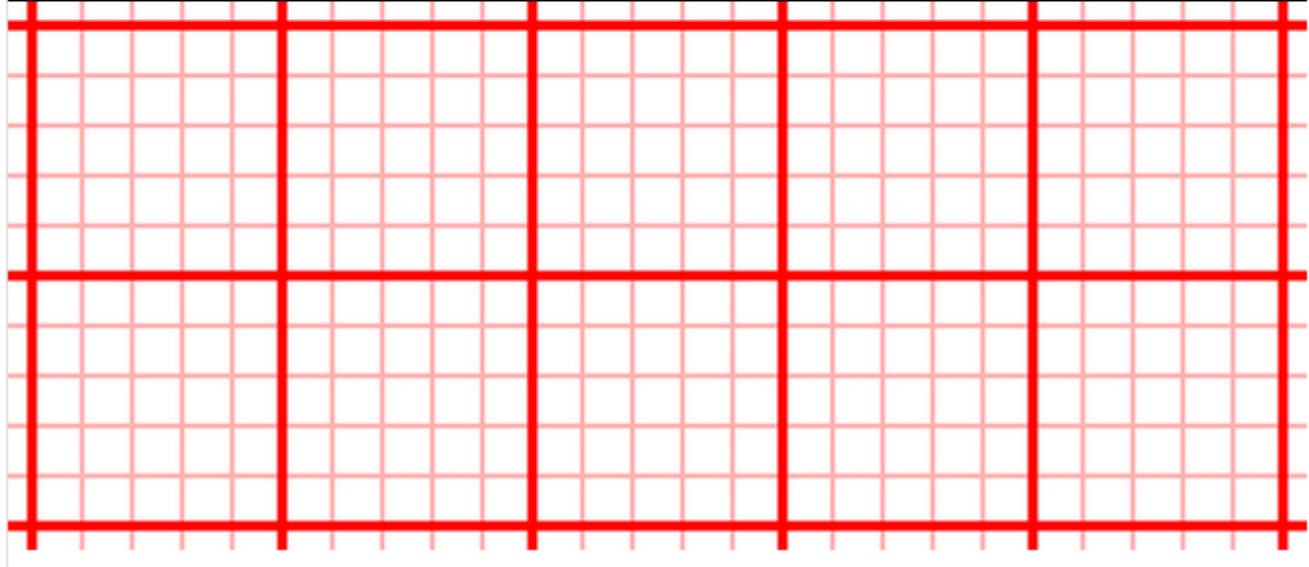
```
.bg{
    width: 500px;
    height: 200px;
    border: 10px solid;
    background: url(img/bg.png);
}
.bg_border{
    width: 500px;
    height: 200px;
    border: 10px solid transparent;
    background: url(img/bg.png);
}
```

```
<body>
  <div class="bg">
  </div>
  <div class="bg_border">
  </div>
</body>
```

.bg



.bg_border

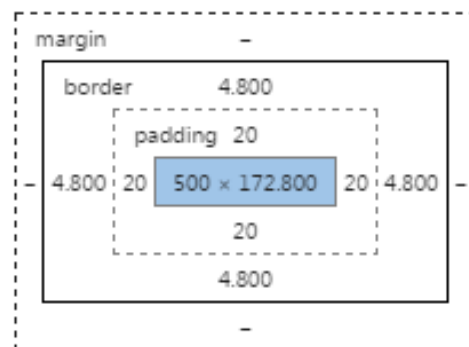
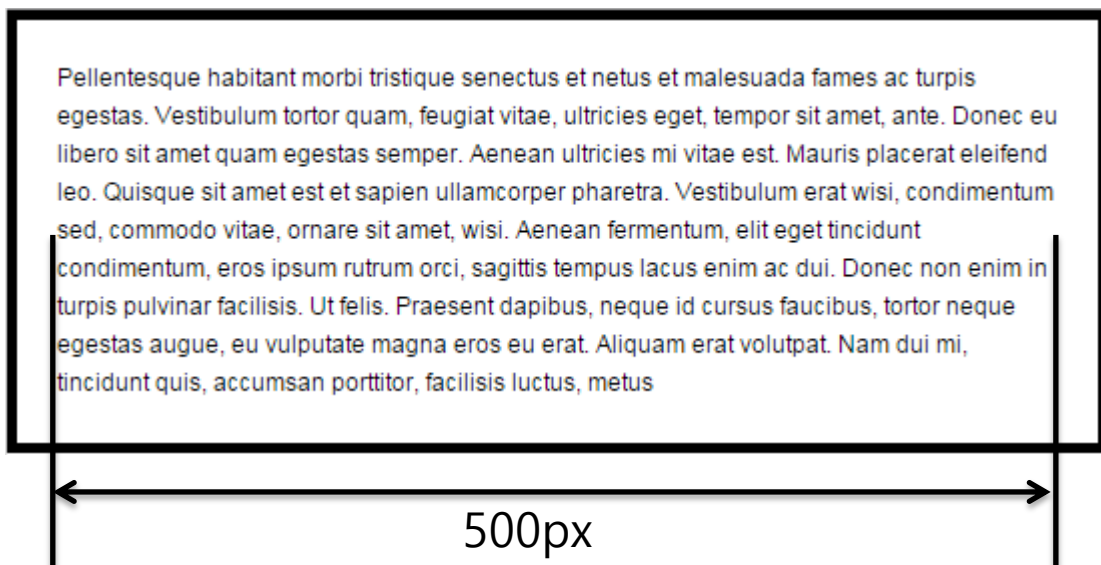


1 크기를 지정하는 width와 height

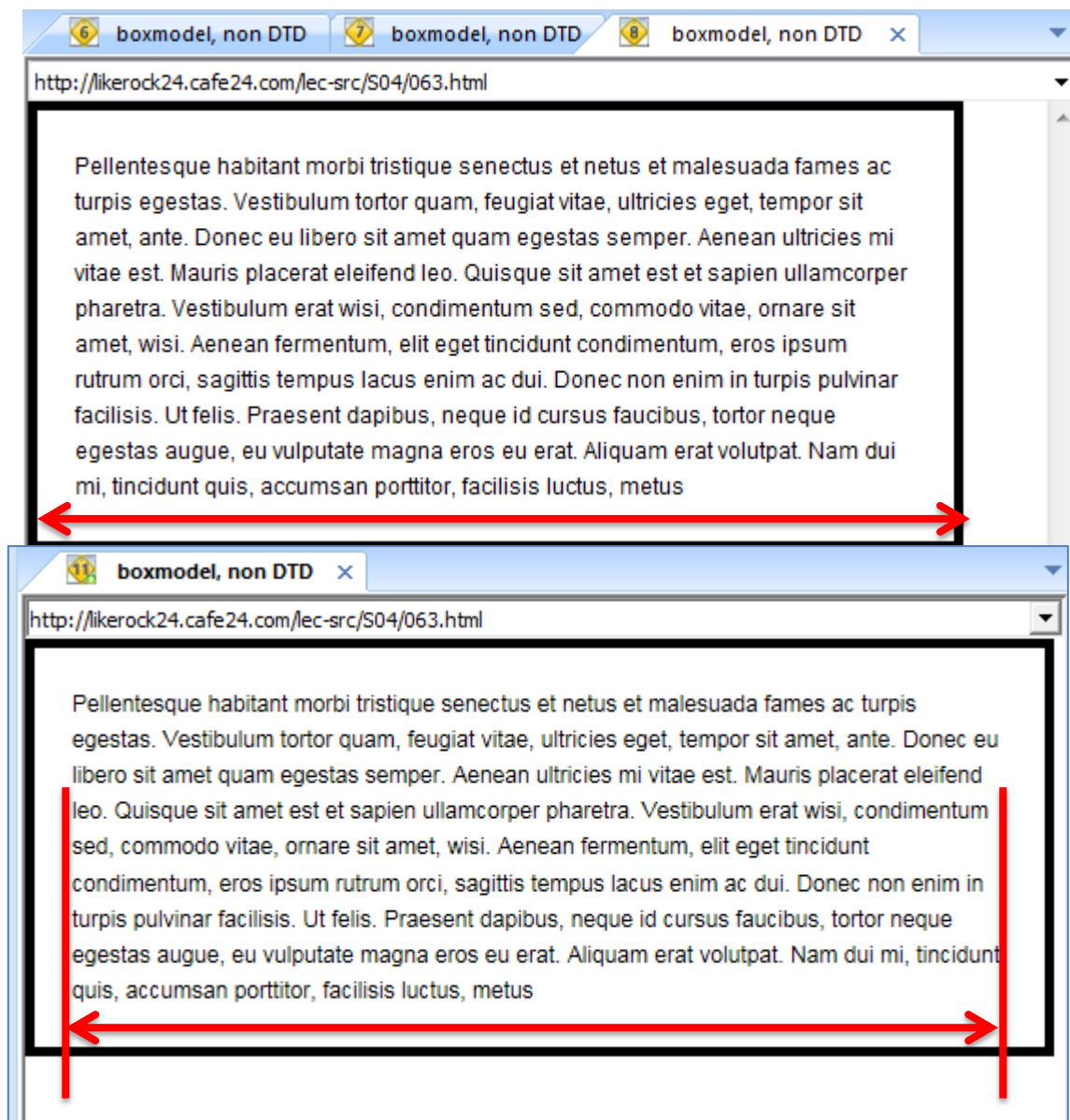
'width'와 'height' 속성이 지정하는 크기는 'content'의 크기

```
<style type="text/css">
*{
    margin: 0;
    padding: 0;
}
body{
    font: 12px/1.6 arial,sans-serif;
}
.boxmodel{
    border: 5px solid;
    padding: 20px;
    width: 500px;
}
</style>
</head>
<body>

<div class="boxmodel">
    <p>Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum tortor quam, feugiat vitae, ultricies eget, tempor sit amet, ante. Donec eu libero sit amet quam egestas semper. Aenean ultricies mi vitae est. Mauris placerat eleifend leo. Quisque sit amet est et sapien ullamcorper pharetra. Vestibulum erat wisi, condimentum sed, commodo vitae, ornare sit amet, wisi. Aenean fermentum, elit eget tincidunt condimentum, eros ipsum rutrum orci, sagittis tempus lacus enim ac dui. Donec non enim in turpis pulvinar facilisis. Ut felis. Praesent dapibus, neque id cursus faucibus, tortor neque egestas augue, eu vulputate magna eros eu erat. Aliquam erat volutpat. Nam dui mi, tincidunt quis, accumsan porttitor, facilisis luctus, metus</p>
</div>
```



8. 박스 모델 스타일



```
<!-- <!DOCTYPE html> -->  
<html lang="ko">  
<head>
```

DOCTYPE이 지정되지 않은 상태
에서의 ie6,7,8의 화면
500px 안에 padding, border를
포함해서 계산

```
<!-- <!DOCTYPE html> -->  
<html lang="ko">  
<head>
```

DOCTYPE 이 지정되지 않은 상태에
서의 ie11의 화면
정상적으로 content만 500px 적용

영역의 height 값을 지정하는 경우 주의 필요. 대체로 height 값을 지정하지 않고 **content가 많아지면 자동으로 늘어나도록** 하는 것이 좋다.
버튼처럼 고정된 높이 값이 있는 것은 예외.

```
<style type="text/css">
*{
    margin: 0;
    padding: 0;
}
body{
    font: 12px/1.6 arial,sans-serif;
}
.boxmodel{
    width: 500px;
    height: 100px;
    border: 5px solid;
}
</style>
</head>
<body>

<div class="boxmodel">
    <div>
        <p>Pellentesque habitant
tortor quam, feugiat vitae, ultricies eg
ultricies mi vitae est. Mauris placerat
wisi, condimentum sed, commodo vitae, orn
rutrum orci, sagittis tempus lacus enim
neque id cursus faucibus, tortor neque e
tincidunt quis, accumsan porttitor, faci
    </div>
    </div>
</div>
```

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum tortor quam, feugiat vitae, ultricies eget, tempor sit amet, ante. Donec eu libero sit amet quam egestas semper. Aenean ultricies mi vitae est. Mauris placerat eleifend leo. Quisque sit amet est et sapien ullamcorper pharetra. Vestibulum erat wisi, condimentum sed, commodo vitae, ornare sit amet, wisi. Aenean fermentum, elit eget tincidunt condimentum, eros ipsum rutrum orci, sagittis tempus lacus enim ac dui. Donec non enim in turpis pulvinar facilisis. Ut felis. Praesent dapibus, neque id cursus faucibus, tortor neque egestas augue, eu vulputate magna eros eu erat. Aliquam erat volutpat. Nam dui mi, tincidunt quis, accumsan porttitor, facilisis luctus, metus

1 border 테두리

'border' 속성에는 **width** (두께), **color** (색), **style**(형태) 지정가능
색상을 지정하지 않을 경우 해당 요소의 글자 색과 같은색으로 지정됨.

```
border: 1px solid;           color(글자색)과 동일한 테두리색  
border: 1px solid black;
```

<1> 원하는 방향에 지정하는 테두리

테두리 방향은 'top, right, bottom, left'

```
border-top: 1px solid black;  
border-right: 1px solid black;  
border-bottom: 1px solid black;  
border-left: 1px solid black;
```


<2> 방향과 속성을 구체적으로 지정

border – 방향 - 속성

border-top-width: 5px;	영역의 위쪽 테두리 두께
border-right-color: black;	영역의 오른쪽 테두리 색상
border-bottom-style: solid;	영역의 아래쪽 테두리 스타일

<3> 테두리 두께

border – width

border-width: 1px;	top, right, bottom, left, 모두 1px
border-width: 1px 5px;	top, bottom 1px / right, left 5px
border-width: 1px 5px 3px;	top 1px / right, left 5px / bottom 3px
border-width: 1px 5px 3px 2px;	top 1px, right 5px, bottom 3px, left 2px

값의 시작은 '**top**', 모두 입력하면 top 부터, right, bottom, left의 **시계방향**으로 설정.
padding, margin 도 동일한 방식으로 설정.

<4> 테두리 색상

border-color

'border-width' 와 마찬가지로 값으로 몇 개를 입력하느냐에 따라 지정되는 위치가 달라짐. 색상값은 , **색상명, hex code, rgb, hsl, raga, hsala** 모두 활용가능.

<5> 테두리 형태

border-style

대략 8가지 스타일, 실무에서는 'solid, dashed, dotted' 정도.
(브라우저 마다 조금씩 달리 보일 수 있다)

```
<style type="text/css">
*{
    margin: 0;
    padding: 0;
}
body{
    font: 12px/1.6 arial,sans
}
.border{
    border: 3px solid blue;
    margin: 5px;
    background-color: #ddd;
}
.border_style_01{
    border-style: solid;
}
.border_style_02{
    border-style: dashed;
}
.border_style_03{
    border-style: dotted;
}
.border_style_04{
    border-style: double;
}
.border_style_05{
    border-style: groove;
}
.border_style_06{
    border-style: ridge;
}
.border_style_07{
    border-style: inset;
}
.border_style_08{
    border-style: outset;
}
```

border-style: solid

border-style: dashed

border-style: dotted

border-style: double

border-style: groove

border-style: ridge

border-style: inset

border-style: outset

groove: 홈(물체에 오목하게 팬줄),
ridge: 솟아 오른, inset: 들어가 보이는, outset: 돌출되어 보
이는

4 테두리의 모서리 둥글게 표현하기

<1> CSS3 'border-radius' 추가됨

border-radius: 10px;	네 방향 모두 10px 둥근 모서리
border-radius: 10px 5px;	좌측상단, 우측하단 10px / 우측상단, 좌측하단 5px (X 자형)
border-radius: 10px 5px 0;	좌측상단 10px / 우측상단, 좌측하단 5px / 우측하단 0
border-radius: 10px 5px 0 8px;	좌측상단부터 시계방향

각각의 방향을 구체적으로 작성 가능

border-top-left-radius: 10px; 좌측상단 모서리 표현.
'top'과 left 자리에 'top, right, bottom, left' 중 원하는 값 지정

```
<style type="text/css">
*{
    margin: 0;
    padding: 0;
}
body{
    font: 12px/1.6 arial,sans-serif;
}
.border{
    border: 3px solid blue;
    margin: 5px;
    background-color: #ddd;
}
.border_radius_01{
    border-radius: 10px;
}
.border_radius_02{
    border-radius: 10px 5px;
}
.border_radius_03{
    border-radius: 10px 5px 0;
}
.border_radius_04{
    border-radius: 10px 5px 0 8px;
}
.border_radius_05{
    border-top-left-radius: 10px;
}
</style>
```

border-radius

border-radius

border-radius

border-radius

border-top-left-radius

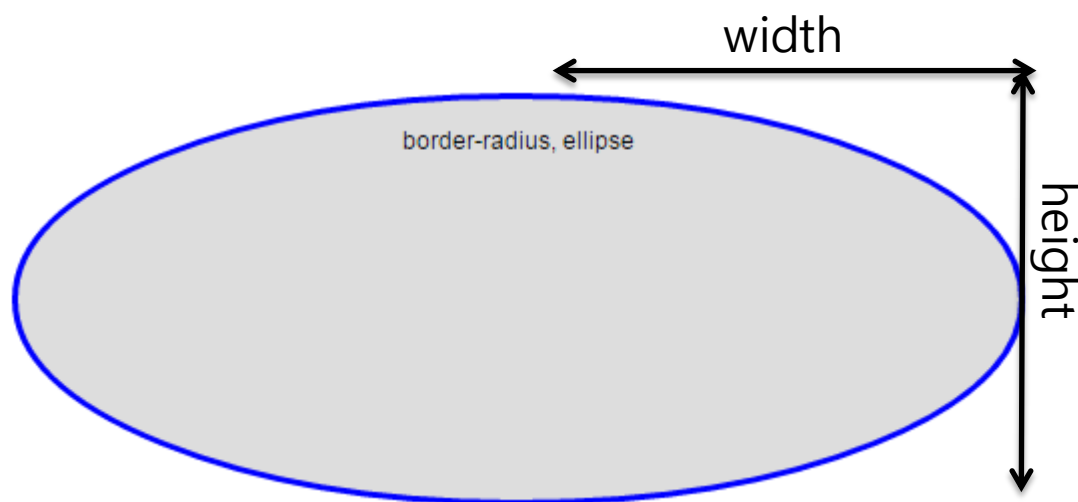
ie,6,7,8 에서는 적용 안됨.

<2> 타원형으로 만들기

```
.border{  
    width: 500px;  
    height: 200px;  
    border: 3px solid blue;  
    border-radius: 50%;  
    background-color: #ddd;  
    text-align: center;  
}
```

```
<style type="text/css">
*{
    /*margin: 0;
    padding: 0;*/
}
body{
    font: 12px/1.6 arial,sans-ser
}
.border{
    width: 500px;
    height: 200px;
    border: 3px solid blue;
    border-radius: 50%;
    background-color: #ddd;
    text-align: center;
}
</style>
</head>
<body>

<div class="border">
    <p>border-radius, ellipse</p>
</div>
```



<3> 타원형 구체적으로 지정

'/'를 통해 '/' 이전은 모서리의 width, '/'이후의 모서리의 height 지정.
다음코드에서는 모서리의 width는 100px, 모서리의 height는 50px 지정.

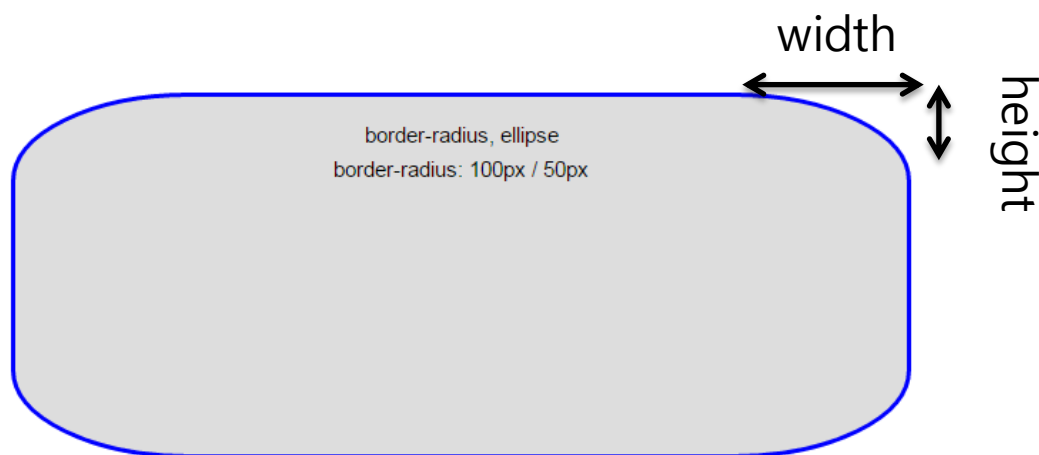
```
.border{  
    width: 500px;  
    height: 200px;  
    border: 3px solid blue;  
    border-radius: 100px / 50px;  
    background-color: #ddd;  
    text-align: center;  
}
```

<4> 타원형 크기 따로 표현

'/'를 통해 '/' 이전은 모서리의 좌측상단, 우측하단 크기 , '/' 이후는 우측상단, 좌측하단 크기

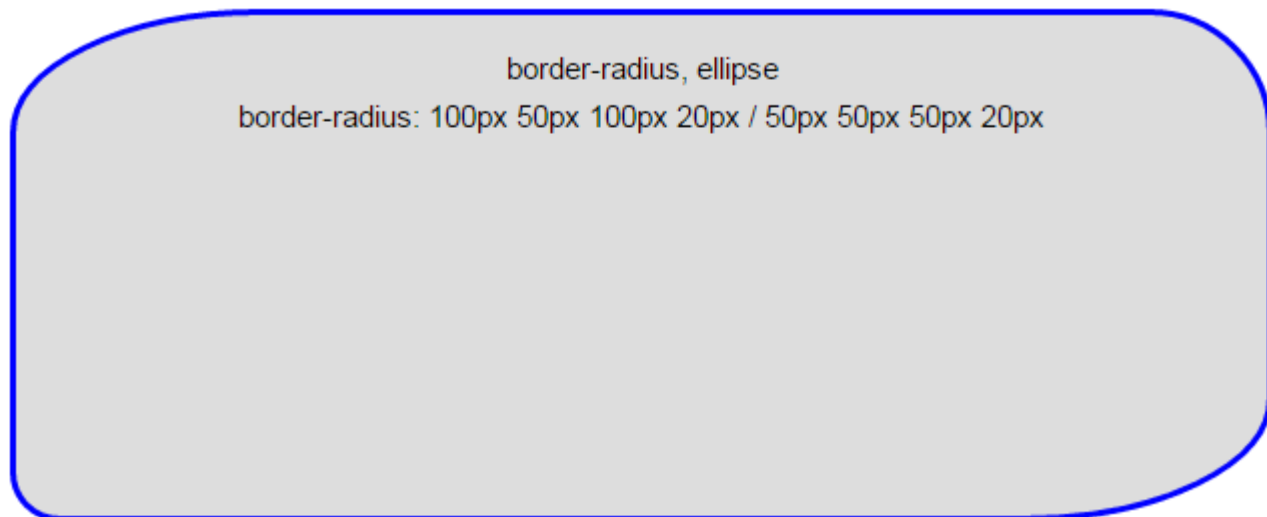

```
<style type="text/css">
*{
    /*margin: 0;
    padding: 0;*/
}
body{
    font: 12px/1.6 arial,sans-se
}
.border{
    width: 500px;
    height: 200px;
    border: 3px solid blue;
    border-radius: 100px / 50px;
    background-color: #ddd;
    text-align: center;
}
</style>
</head>
<body>

<div class="border">
    <p>border-radius, ellipse<br>
    border-radius: 100px / 50px<
</div>
```



```
<style type="text/css">
*{
    /*margin: 0;
    padding: 0;*/
}
body{
    font: 12px/1.6 arial,sans-serif;
}
.border{
    width: 500px;
    height: 200px;
    border: 3px solid blue;
    border-radius: 100px 50px 100px 20px / 50px 50px 50px 20px;
    background-color: #ddd;
    text-align: center;
}
</style>
```

'/' 이전은 모서리의 좌측상단, 우측하단 크기 ,
'/' 이후는 우측상단, 좌측하단 크기



5 margin 설정

해당 요소의 테두리를 기점으로 바깥쪽 여백이 'margin'
요소가 연속적으로 배치될 때는 요소와 요소 사이의 margin이 모두 적용되지는 않고,
두요소중 큰 값이 표현됨.

<1> 방향별 margin 설정

```
margin-top: 10px;  
margin-right: 10px;  
margin-bottom: 10px;  
margin-left: 10px;
```

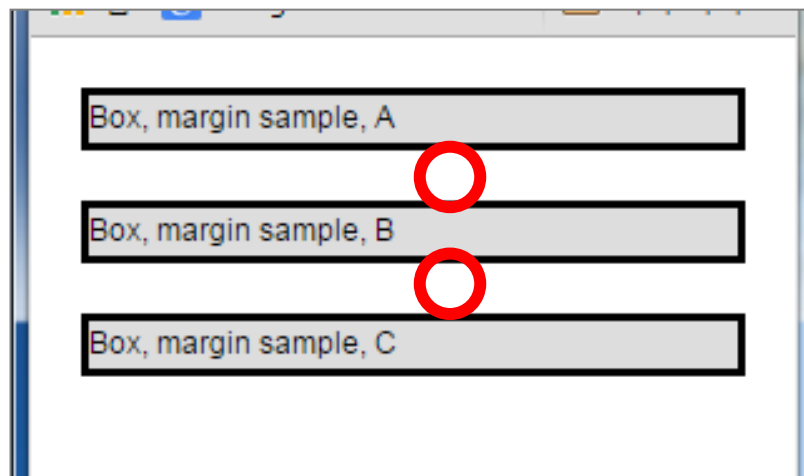
margin의 값을 지정할 때 지정되는 방향

margin: 10px	모든방향 10px
margin: 10px 0;	top,bottom 10px, right, left 0
margin: 10px 0 5px;	top 10, right left 0, bottom 5px
margin: 10px 0 5px 20px;	top부터 시계방향

<2> 연속된 요소의 margin은 중첩되어 나타난다.

```
<style type="text/css">
*{
    margin: 0;
    padding: 0;
}
body{
    font: 12px/1.6 arial,sans-ser
}
.margin{
    width: auto;
    border: 3px solid;
    margin: 20px;
    background-color: #ddd;
}
</style>
</head>
<body>

<div class="margin">
    <p>Box, margin sample, A</p>
</div>
<div class="margin">
    <p>Box, margin sample, B</p>
</div>
<div class="margin">
    <p>Box, margin sample, C</p>
</div>
```

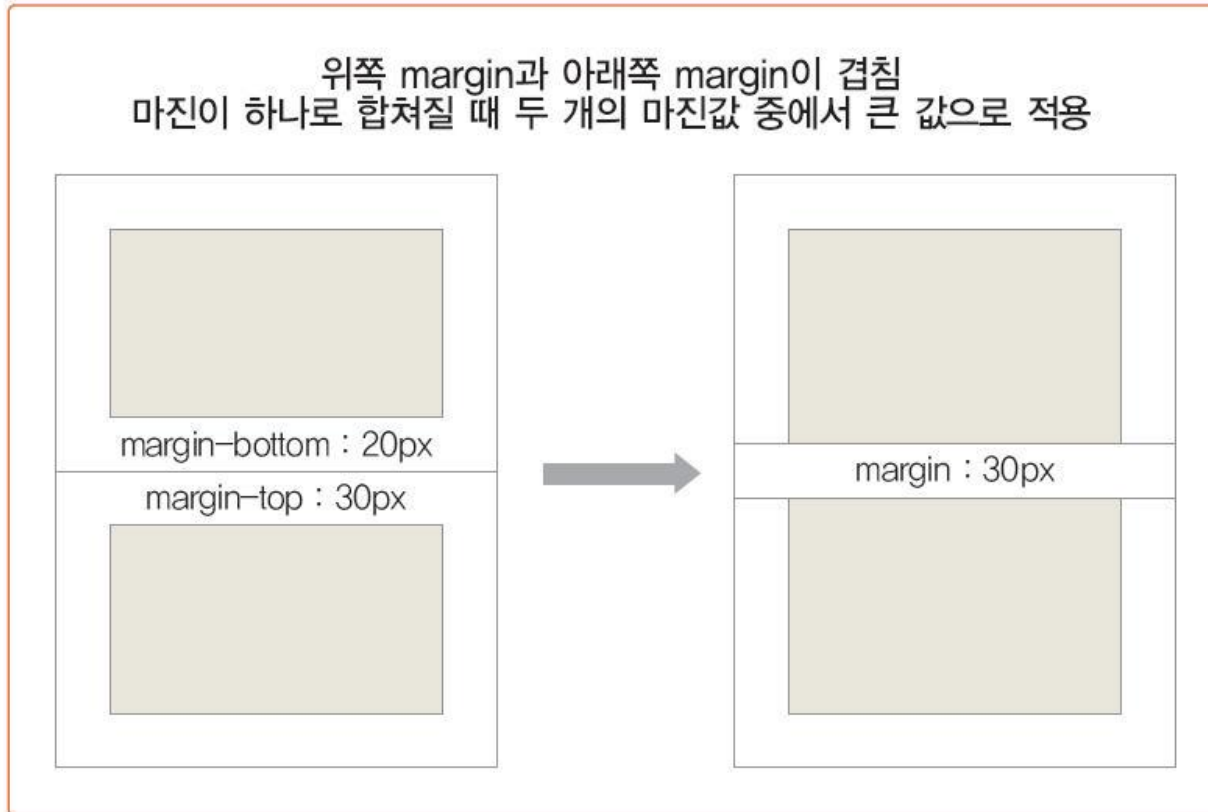


3개 블록에 margin 적용했는데,
top, bottom 방향의 마진값이 중첩되어 한방향 마진
만 표현됨. 두값중 큰값이 출력.

margin 겹침현상

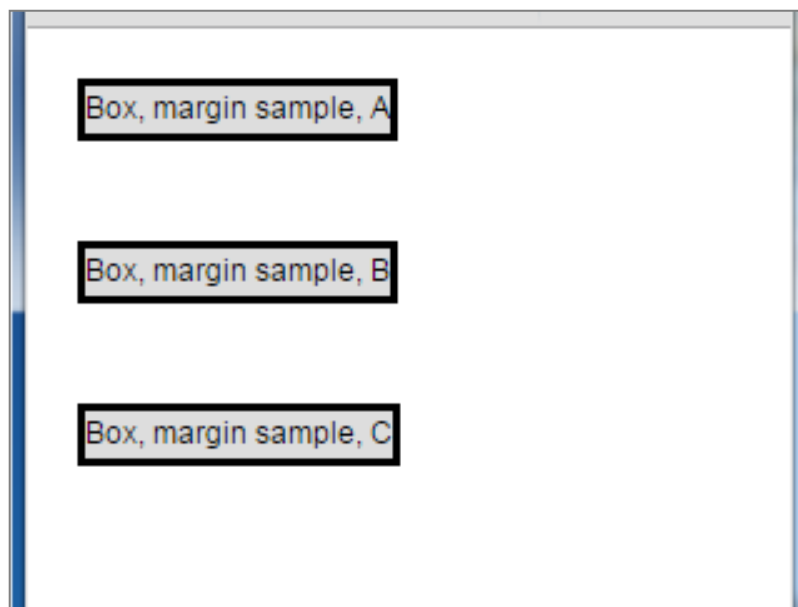
margin collapsing (마진 겹침현상)

두개 이상의 margin값이 세로 방향으로 만났을 때 하나의 margin으로 합쳐지는 현상



```
<style type="text/css">
*{
    margin: 0;
    padding: 0;
}
body{
    font: 12px/1.6 arial,sans-serif;
}
.margin{
    float: left;
    width: auto;
    border: 3px solid;
    margin: 20px;
    background-color: #ddd;
}
</style>
</head>
<body>

<div class="margin">
    <p>Box, margin sample, A</p>
</div>
<div class="margin">
    <p>Box, margin sample, B</p>
</div>
<div class="margin">
    <p>Box, margin sample, C</p>
</div>
```



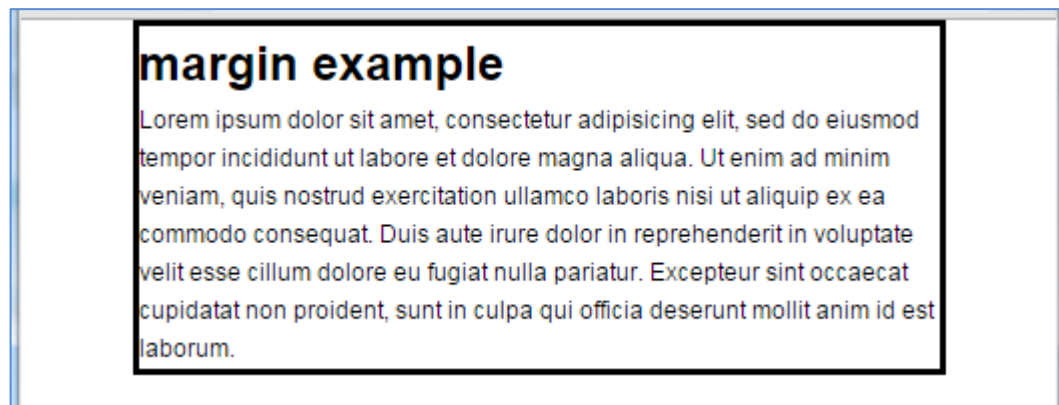
'float' 을 추가하면 margin이 중첩되지 않고 표현됨. 단, float은 내용의 크기가 지정되지 않은 경우 내용이 가진 영역까지만 감싸게 되어 width가 달라 보이게 된다.

<3> margin을 활용한 가운데 정렬

right, left 값을 'auto'로 설정하면 가운데 정렬
top, bottom은 'auto'를 지정해도 '0'과 같이 출력되므로
'margin:auto;'라고 해도 같은 결과 표현.
주의할 점은 이때 해당 영역에는 **'width' 값이 지정되어** 있어야 한다.

```
<style type="text/css">
*{
    margin: 0;
    padding: 0;
}
body{
    font: 12px/1.6 arial,sans-serif;
}
.margin{
    width: 400px;
    border: 3px solid;
    margin: 0 auto;
}
</style>
</head>
<body>

<div class="margin">
    <h1>margin example</h1>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod
    adipiscing elit, sed do eiusmod
```



6 padding으로 안쪽 공간 지정하기

해당 요소의 테두리를 기점으로 안쪽 공간이 'padding'
padding 영역에는 지정된 배경이 출력되며, 해당영역에 지정된 가로, 세로 값이 있다면,
padding은 그 크기의 바깥쪽이면서 테두리의 안쪽이 된다.

※ ie 의 경우 ie8까지는 문서형 선언(doctype)이 지정되지 않을 경우 width, height의 값은 padding과 border 까지 포함.

어떤 브라우저에서도(문서형선언 이 지정되지 않은 익스플로러 포함) padding 값이 문제 없
이 지정되게 하려면, width, height가 지정된 영역에는 padding, border를 쓰지 않는 것이
좋다.


```

<style type="text/css">
*{
    margin: 0;
    padding: 0;
}
body{
    font: 12px/1.6 arial,sans-se
}
.area{
    width: 480px;
    margin: 0 auto;
    text-align: justify;
}
.area .padding{
    padding: 10px 20px;
    border: 3px solid;
}
</style>
</head>
<body>

<div class="area">
    <div class="padding">
        <h1>padding example</h1>
        <p>Lorem ipsum dolor sit
sed do eiusmod
tempor incididunt ut lab
ad minim veniam,
quis nostrud exercitatio
ea commodo
consequat. Duis aute iru
voluptate velit esse
cillum dolore eu fugiat
occaecat cupidatat non
    
```

padding example

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor
 incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud
 exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute
 irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla
 pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia
 deserunt mollit anim id est laborum

7 table 요소와 관련된 속성들

① 테이블의 셀은 텍스트의 길이에 따라 자동으로 조절되는 것이 기본.

table-layout 속성은 auto, fixed 값을 지정할 수 있고, **auto**가 기본값이고 fixed는 cell 의 너비를 고정한다.

```

<style type="text/css">
body,h1,h2,h3{
    font-family: tahoma,sans-serif;
    font-size: 12px;
}
.fixed{
    table-layout: fixed;
}
</style>
</head>
<body>

<h1>지정하지 않음</h1>
<table width="400" border="1">
    <tbody>
        <tr>
            <td>Lorem ipsum dolor sit amet.</td>
            <td>Lorem ipsum.</td>
        </tr>
    </tbody>
</table>

<h1>fixed 지정</h1>
<table width="400" border="1" class="fixed">
    <tbody>
        <tr>

```

지정하지 않음

Lorem ipsum dolor sit amet.

Lorem ipsum.

fixed 지정

Lorem ipsum dolor sit amet.

Lorem ipsum.

fixed 설정 시에는 텍스트 길이와 상관없이 두 셀의 너비는 같은 크기로 맞춰짐.

- ② '**border-collapse**' 속성은 테이블 작성 시 cell과 cell 사이의 공백을 제거하는 데 효과적. 셀 사이를 분리하는 값은 '**separate**' 공백을 없애고 셀의 경계를 합치려면 '**collapse**'

```
<style type="text/css">
body,h1,h2,h3{
    font-family: tahoma,sans-serif;
    font-size: 12px;
}
.separate{
    border-collapse: separate;
}
.collapse{
    border-collapse: collapse;
}
</style>
</head>
<body>

<h1>separate 결과</h1>
<table width="400" border="1" class="separate">
    <tbody>
        <tr>
            <td>Lorem ipsum dolor sit amet.</td>
            <td>Lorem ipsum.</td>
        </tr>
    </tbody>
</table>

<h1>collapse 결과</h1>
<table width="400" border="1" class="collapse">
    <tbody>
        <tr>
```

separate 결과

Lorem ipsum dolor sit amet.

Lorem ipsum.

collapse 결과

Lorem ipsum dolor sit amet.

Lorem ipsum.

③ '**border-spacing**' 속성은 테이블에서 'border-collapse:collapse'가 지정되지 않아야 설정 가능.

셀사이 공백은 하나의 값 또는 두개의 값을 지정할 수 있다.

```
<style type="text/css">
body,h1,h2,h3{
    font-family: tahoma,sans-serif;
    font-size: 12px;
}
.spacing1{
    border-spacing: 5px;
}
.spacing2{
    border-spacing: 1px 10px;
}
</style>
</head>
<body>

<h1>하나의 값</h1>
<table width="400" border="1" class="spacing1">
    <tbody>
        <tr>
            <td>Lorem ipsum dolor s
            <td>Lorem ipsum.</td>
        </tr>
    </tbody>
</table>

<h1>두 개의 값</h1>
<table width="400" border="1" class="spacing2">
    <tbody>
```

하나의 값

Lorem ipsum dolor sit amet.

Lorem ipsum.

두 개의 값

Lorem ipsum dolor sit amet.

Lorem ipsum.

※ **border-spacing**의 경우
값이 하나일때는 상하좌우 전체 방향에 적용되지만,
값이 두개일때는 다른 여백설정과 달리
좌우 / 상하의 순서로 적용된다.

- ④ 'empty-cell' 속성은 테이블에서 비어 있는 셀을 표시할지의 여부 지정.

```
<style type="text/css">
body,h1,h2,h3{
    font-family: tahoma,sans-serif;
    font-size: 12px;
}
.empty1{
    empty-cells: show;
}
.empty2{
    empty-cells: hide;
}
</style>
</head>
<body>

<table width="400" border="1">
    <tbody>
        <tr>
            <td>Lorem ipsum dolor sit
            <td>Lorem ipsum dolor.</td>
        </tr>
        <tr>
            <td>Lorem ipsum dolor sit
            <td class="empty1"></td>
        </tr>
        <tr>
            <td>Lorem ipsum dolor sit
            <td class="empty2"></td>
```

olor sit amet, consectetur.	Lorem ipsum dolor.
olor sit amet, consectetur.	
olor sit amet, consectetur.	

- ⑤ 'caption' 요소를 지정할때, 캡션의 위치를 지정하는 방법으로 'caption-side' 활용,
값은 top, bottom

```
<style type="text/css">
body,h1,h2,h3{
    font-family: tahoma,sans-serif;
    font-size: 12px;
}
.caption{
    caption-side: bottom;
}
</style>
</head>
<body>

<table width="400" border="1" class="caption">
    <caption>Lorem ipsum dolor sit amet.</caption>
    <tbody>
        <tr>
            <td>Lorem ipsum dolor sit amet, consectetur.</td>
            <td>Lorem ipsum dolor.</td>
        </tr>
    </tbody>
</table>
```

Lorem ipsum dolor sit amet, consectetur.

Lorem ipsum dolor.

Lorem ipsum dolor sit amet.

※ 값으로 left, right를 지정할 수 있으나,
표준방식은 아니다.

9. 포지셔닝(positioning) 스타일

1 플롯(float)

문서 레이아웃 작성시 가장 많이 활용하는 '플롯(float)' 값은 left, right

<1> float을 지정하는 다음에 오는 콘텐츠가 float를 지정한 블록 주위를 감싼다.

※ html 요소 중에서 인라인 요소와 블록 요소는 그 결과가 다르게 나타나는데, float을 지정하면 두 요소는 같은 결과를 보여준다.

크기를 지정할 수 있고(인라인 요소는 크기지정이 안되지만, float 지정 후에는 가능), padding과 margin을 지정한 결과도 동일하게 표현된다.


```
<style type="text/css">
*{
    margin: 0;
    padding: 0;
}
body{
    font: 12px/1.6 arial,sans-serif;
}
.content{
    width: 500px;
    margin: 20px auto;
}
.thum{
    float: left;
    margin: 0 15px 15px 0;
}
.text{
    text-align: justify;
}
/* .thum{
    float: left;
}
.text{
    margin-left: 115px;
    text-align: justify;
} */

</style>
</head>
<body>

<div class="content">
    
    <p class="text">Pellentesque habitant morbi tristique senect
    Pellentesque habitant morbi tristique senect et netus et malesuada
    fames ac turpis egestas. Vestibulum tortor quam, feugiat vitae, ultricies
    eget, tempor sit amet, ante. Donec eu libero sit amet quam egestas
    semper. Aenean ultricies mi vitae est. Mauris placerat eleifend leo.
    Quisque sit amet est et sapien ullamcorper pharetra. Vestibulum erat wisi,
    condimentum sed, commodo vitae, ornare sit amet, wisi. Aenean
    fermentum, elit eget tincidunt condimentum, eros ipsum rutrum orci,
    sagittis tempus lacus enim ac dui. Donec non enim in turpis pulvinar
    facilisis. Ut felis. Praesent dapibus, neque id cursus faucibus, tortor neque egestas augue, eu
    vulputate magna eros eu erat. Aliquam erat volutpat. Nam dui mi, tincidunt quis, accumsan
    porttitor, facilisis luctus, metus
```



Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum tortor quam, feugiat vitae, ultricies eget, tempor sit amet, ante. Donec eu libero sit amet quam egestas semper. Aenean ultricies mi vitae est. Mauris placerat eleifend leo. Quisque sit amet est et sapien ullamcorper pharetra. Vestibulum erat wisi, condimentum sed, commodo vitae, ornare sit amet, wisi. Aenean fermentum, elit eget tincidunt condimentum, eros ipsum rutrum orci, sagittis tempus lacus enim ac dui. Donec non enim in turpis pulvinar facilisis. Ut felis. Praesent dapibus, neque id cursus faucibus, tortor neque egestas augue, eu vulputate magna eros eu erat. Aliquam erat volutpat. Nam dui mi, tincidunt quis, accumsan porttitor, facilisis luctus, metus

이미지에 **float**을 지정해서 다음에 오는 텍스트가 이 이미지를 감싸도록 설정.

- dummy img 삽입하기

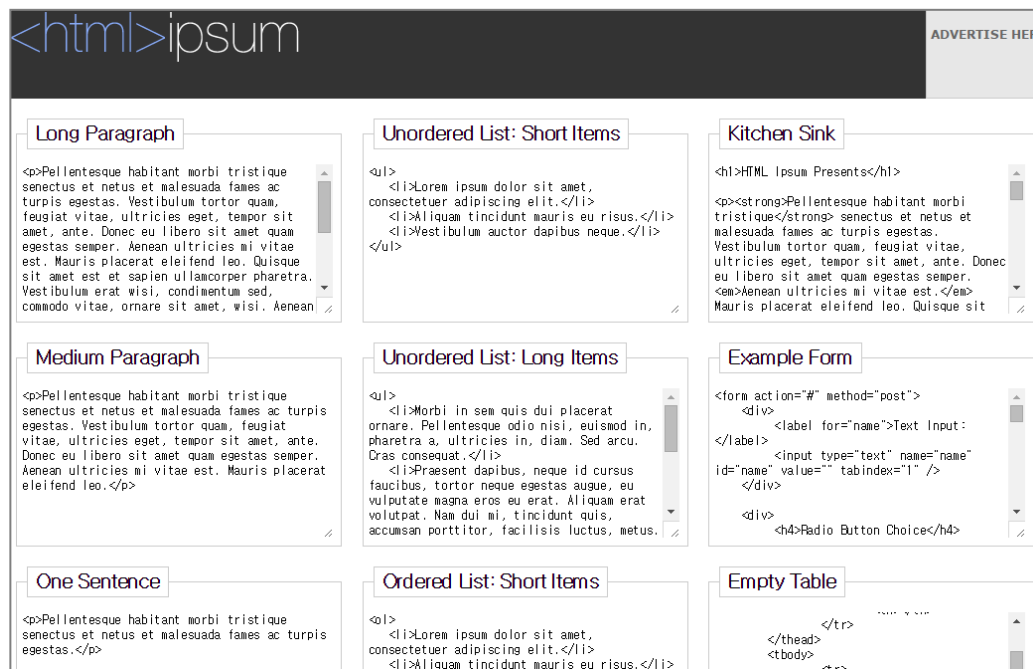
테스트용 이미지 삽입하고자 할때 경로에서 'http://placeholder.it/이
미지의 가로크기 x 세로크기' 를 지정하면 원하는 크기의 이미지 지
정 가능

```

```

- dummy text 삽입하기

<http://html-ipsu.com>



텍스트가 이미지 옆으로 별도의 새로운 칼럼으로 만든 것처럼 정렬. margin 활용

```
<style type="text/css">
*{
    margin: 0;
    padding: 0;
}
body{
    font: 12px/1.6 arial,sans-serif;
}
.content{
    width: 500px;
    margin: 20px auto;
}
.thum{
    float: left;
    margin: 0 15px 15px 0;
}
.text{
    margin-left: 115px;
    text-align: justify;
}
```



Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum tortor quam, feugiat vitae, ultricies eget, tempor sit amet, ante. Donec eu libero sit amet quam egestas semper. Aenean ultricies mi vitae est. Mauris placerat eleifend leo. Quisque sit amet est et sapien ullamcorper pharetra. Vestibulum erat wisi, condimentum sed, commodo vitae, ornare sit amet, wisi. Aenean fermentum, elit eget tincidunt condimentum, eros ipsum rutrum orci, sagittis tempus lacus enim ac dui. Donec non enim in turpis pulvinar facilisis. Ut felis. Praesent dapibus, neque id cursus faucibus, tortor neque egestas augue, eu vulputate magna eros eu erat. Aliquam erat volutpat. Nam dui mi, tincidunt quis, accumsan porttitor, facilisis luctus, metus

margin은 float를 지정한 블록부터가 아니라 바깥에 감싸고 있는 영역(여기서는 content)에서 그 값을 계산한다.

<2> float의 영향을 받지 않으려면 'clear' 속성으로 float를 해지해야 한다.

clear 의 값으로는 'left, right, both'

```
<style type="text/css">
*{
    margin: 0;
    padding: 0;
}
body{
    font: 12px/1.6 arial,sans-serif;
}
.content{
    width: 500px;
    margin: 20px auto;
}
.thum{
    float: left;
}
.text{
    margin-left: 115px;
    text-align: justify;
}
.break{
    clear: left;
}
</style>
```

```
<div class="content">
    
    <p class="text">Pellentesque habitant morbi tristique senectus
et netus et malesuada fames ac turpis egestas. <br class="break">
    Vestibulum tortor quam, feugiat vitae, ultricies eget, tempor
sit amet, ante. Donec eu libero sit amet quam egestas semper. Aenean
ultricies mi vitae est. Mauris placerat eleifend leo. Quisque sit amet
est et sapien ullamcorper pharetra. Vestibulum erat wisi, condimentum
sed, commodo vitae, ornare sit amet, wisi. Aenean fermentum, elit eget
tincidunt condimentum, eros ipsum rutrum orci, sagittis tempus lacus
enim ac dui. Praesent dapibus, neque id cursus faucibus, tortor neque egestas augue, eu vulputate
mi, tincidunt quis, accumsan porttitor, facilisis luctus, metus</div>
```



Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

Vestibulum tortor quam, feugiat vitae, ultricies eget, tempor sit amet, ante. Donec eu libero sit amet quam egestas semper. Aenean ultricies mi vitae est. Mauris placerat eleifend leo. Quisque sit amet est et sapien ullamcorper pharetra. Vestibulum erat wisi, condimentum sed, commodo vitae, ornare sit amet, wisi. Aenean fermentum, elit eget tincidunt condimentum, eros ipsum rutrum orci, sagittis tempus lacus enim ac dui. Praesent dapibus, neque id cursus faucibus, tortor neque egestas augue, eu vulputate magna eros eu erat. Aliquam erat volutpat. Nam dui mi, tincidunt quis, accumsan porttitor, facilisis luctus, metus

clear를 지정하면서 이전의 float를 해지해서 그 영역 밖으로 벗어난다.

CLEAR

- float 를 해제할 때 사용
- 이전 엘리먼트의 float값을 상속받아 같은 방향으로 정렬되는 현상을 막기 위해 사용

Clear 속성

clear : inherit ; (기본)

clear : left ;

clear : right ;

clear : both

2 플롯을 활용한 레이아웃

```
<style type="text/css">
*{
    margin: 0;
    padding: 0;
}
img{
    vertical-align: top;
}
.content{
    width: 500px;
    margin: 20px auto;
    background-color: lime;

```

```
<!-- *zoom: 1; -->
```

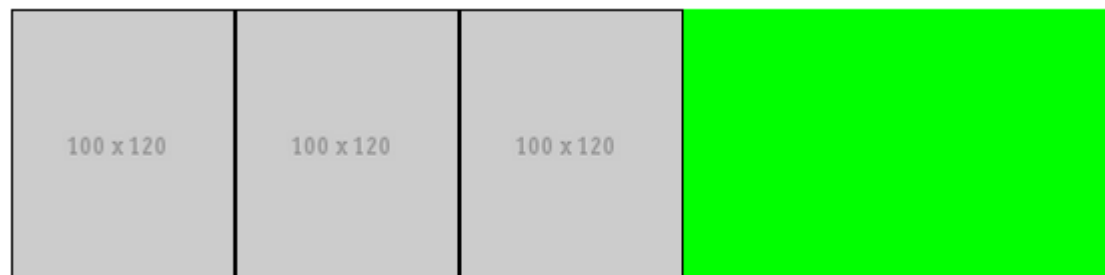
```

}
.content:after{
    content: "";
    display: block;
    clear: both;
}
.content p{
    float: left;
    border: 1px solid;
}

```

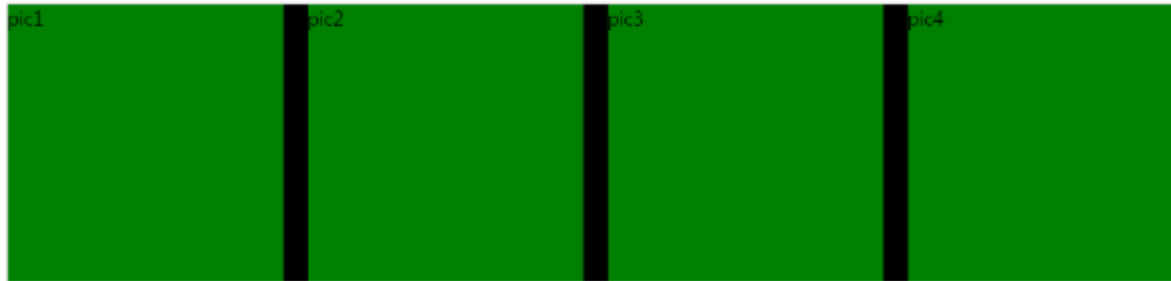
```
</style>
</head>
<body>
```

```
<div class="content">
    <p></p>
    <p></p>
    <p></p>
</div>
```

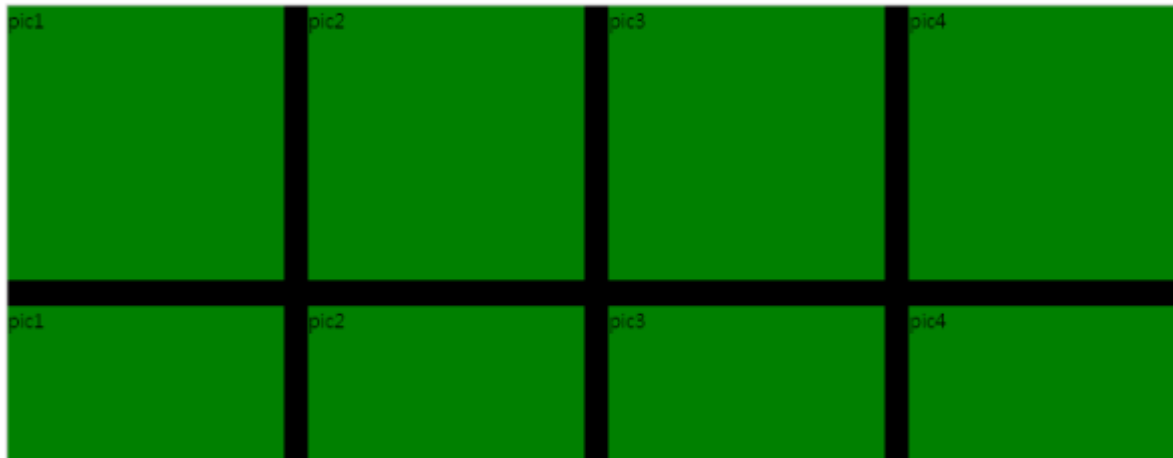


'content' 속성은 ':after' 위치에 clear를 지정할 수 있는 요소(빈요소)를 생성하기 위해 작성.

CSS_Layout_05 <http://codepen.io/alikerock/pen/MyPWOY>



CSS_Layout_06 <http://codepen.io/alikerock/pen/aNRbqp>



CSS_Layout_07 <http://codepen.io/alikerock/pen/grB0mz>

Logo

Login Member Join Site map Location

084_test.html

wrap 800px silver / header read / container green
nav 180 px lime / content 600px lime/ footer green

wrap은 브라우저 중앙 배치
container와 footer 사이가 벌어져 있도록.
nav와 content는 좌우로 배치되도록 float 지정.

LOGO

This website is...

Navigation

[Menu 1](#)

[Menu 2](#)

[Menu 3](#)

Content Title

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum tortor quam, feugiat vitae, ultricies eget, tempor sit amet, ante. Donec eu libero sit amet quam egestas semper. Aenean ultricies mi vitae est. Mauris placerat eleifend leo. Quisque sit amet est et sapien ullamcorper pharetra. Vestibulum erat wisi, condimentum sed, commodo vitae, ornare sit amet, wisi. Aenean fermentum, elit eget tincidunt condimentum, eros ipsum rutrum orci, sagittis tempus lacus enim ac dui. Donec non enim in turpis pulvinar facilisis. Ut felis. Praesent dapibus, neque id cursus faucibus, tortor neque egestas augue, eu vulputate magna eros eu erat. Aliquam erat volutpat. Nam dui mi, tincidunt quis, accumsan porttitor, facilisis luctus, metus

Seoul, Korea

3 position의 특징

position의 값은 'absolute, relative, fixed.

position 설정 후에 left, right, top, bottom 속성으로 그 위치를 지정.

① html에서 2개 이상의 요소가 나열되면, 뒤에 배치되는 요소는 앞에 놓인 요소를 기준으로 배치되는데,
position의 값을 '**absolute**'로 지정하면 다음요소가 이 블록을 **인지하지 못한 채** 주어진 '절대값'을 기준으로 배치된다.

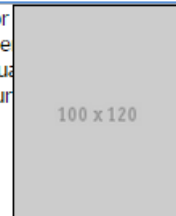
position : static ;	(기본, offset을 가지지 않음)
position : relative ;	(상대적, 원래 위치를 기준)
position : absolute ;	(절대적, 부모위치를 기준)
position : fixed ;	(화면에서 고정)

```
<style type="text/css">
*{margin: 0; padding: 0;}
body{font: 12px Tahoma, sans-serif;}
img{vertical-align: top;}
.a{
    position: absolute;
    left: 100px;
    border: 1px solid;
}
</style>
</head>
<body>

<div class="a"></div>

<p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit, sed do eiusmod
tempor incididunt ut labore et dolore magna aliqua. U
enim ad minim veniam,
```

ur adipiscing elit, sed do eiusmod tempor in
m, quis nostrud exercitation ullamco laboris
olor in reprehenderit in voluptate velit esse
atat non proident, sunt in culpa qui officia de
laborum.

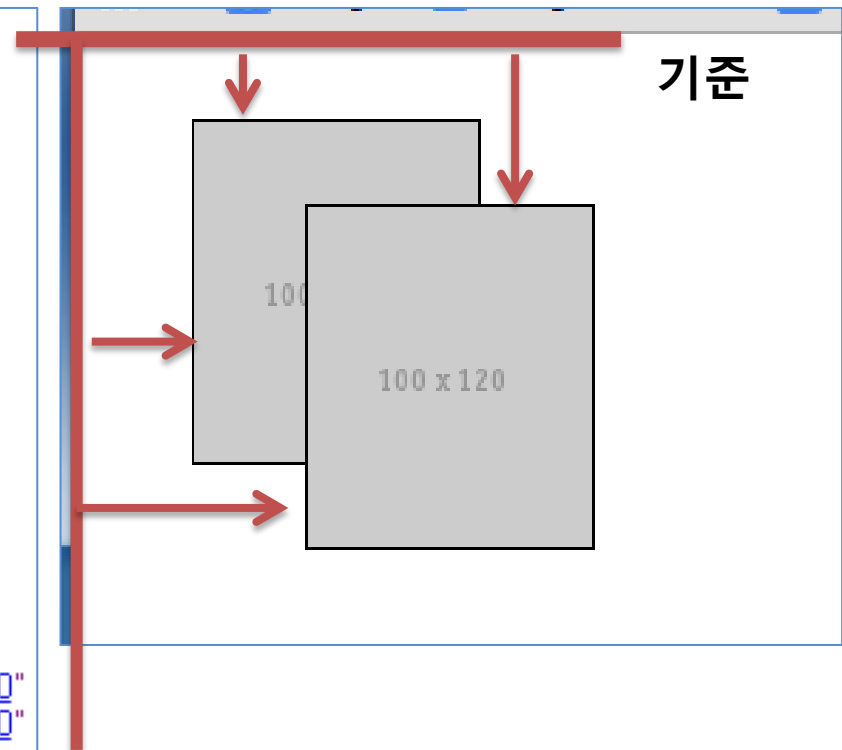


position의 값을 '**absolute**'로 지정하면 다음요소가 이 블록을 인지하지 못한 채 주어진 '절대값'을 기준으로 배치된다.

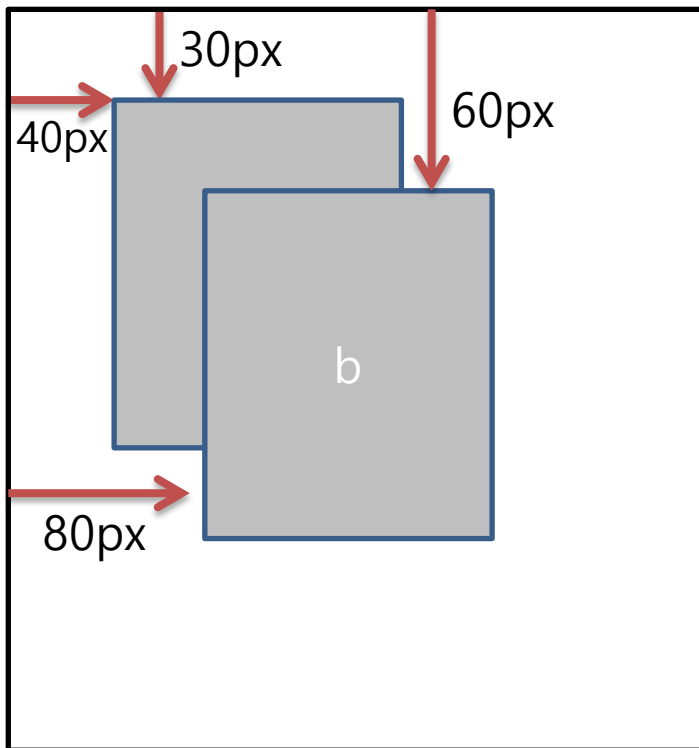
- ② '**absolute**' 지정 후에 '**left, top, right, bottom**' 속성으로 위치를 지정.
absolute 적용 요소를 감싸는 블록에 position이 지정되어 있다면 그 블록이 기준이 되고, 지정이 없으면, <**body**> 기준이 된다.

```
<style type="text/css">
*{margin: 0; padding: 0;}
img{vertical-align: top;}
.a, .b{border: 1px solid;}
.a{
    position: absolute;
    left: 40px;
    top: 30px;
}
.b{
    position: absolute;
    left: 80px;
    top: 60px;
}
</style>
</head>
<body>

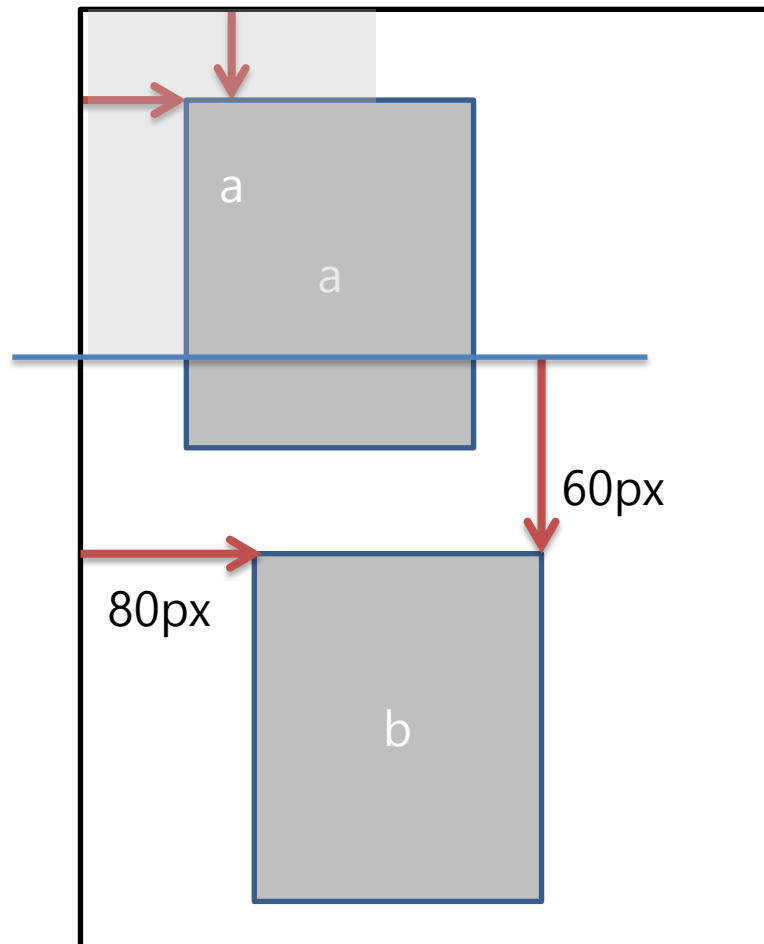
<div class="a">
</html>
```



```
.a {position:absolute; left:40px; top:30px;}  
.b {position:absolute; left:80px; top:60px;}
```



```
.a {position:relative; left:40px; top:30px;}  
.b {position:relative; left:80px; top:60px;}
```



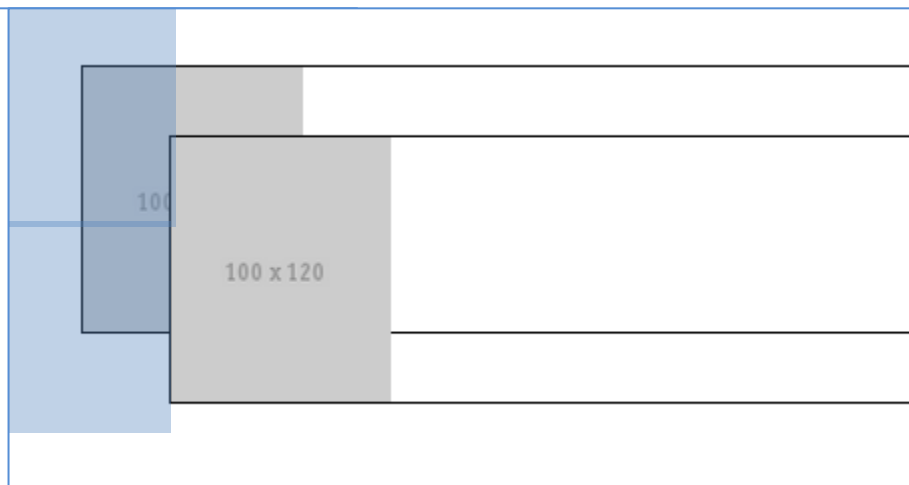
③ position 값으로 '**relative**'를 지정하면 'left, top, right, bottom'의 위치값들은 relative가 지정된 블록 그 자신의 위치 (2번째 놓이는 요소는 첫번째 요소의 다음 위치)에서 지정한 값만큼 이동한다.

```
<style type="text/css">
*{margin: 0; padding: 0;}
img{vertical-align: top;}
.a, .b{border: 1px solid;}
.a{
    position: relative;
    left: 40px;
    top: 30px;
}
.b{
    position: relative;
    left: 80px;
    top: -60px;
}
</style>
```

```
</head>
<body>

<div class="a"></div>
<div class="b"></div>

</body>
</html>
```



④ position 속성값으로 '**absolute**'를 지정한 블록에 '**left**'와 '**right**', '**top**'과 '**bottom**'을 동시에 적용할 수도 있다.

이 경우에는 가로(width), 세로(height)값을 따로 적용하지 않아야 정확한 결과를 확인할 수 있다.

```
<style type="text/css">
*{margin: 0; padding: 0;}
img{vertical-align: top;}
.box{
    position: absolute;
    left: 20px;
    right: 20px;
    top: 10px;
    bottom: 10px;
    background-color: lime;
}
</style>
</head>
<body>

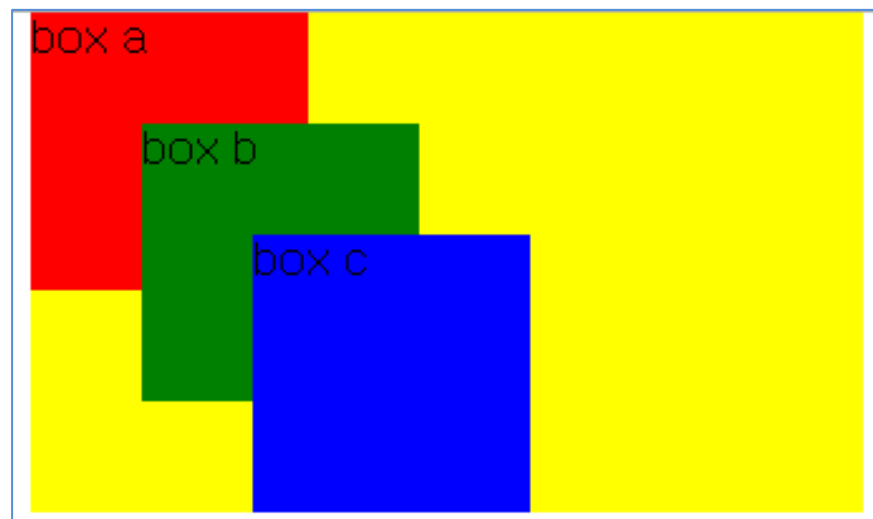
<div class="box">position: absolute</div>

</body>
</html>
```



- ⑤ 'position: **absolute**'를 지정하면 이를 감싸는(부모) 블록에는 '**relative**'를 지정

```
<style type="text/css">
*{margin: 0; padding: 0;}
img{vertical-align: top;}
.area{
  position: relative;
  width: 300px;
  height: 180px;
  margin: 0 auto;
  background-color: yellow;
}
.box{
  position: absolute;
  width: 100px;
  height: 100px;
}
.a{
  left: 0;
  top: 0;
  background-color: red;
}
.b{
  left: 40px;
  top: 40px;
  background-color: green;
}
.c{
  left: 80px;
  top: 80px;
  background-color: blue;
}
</style>
```



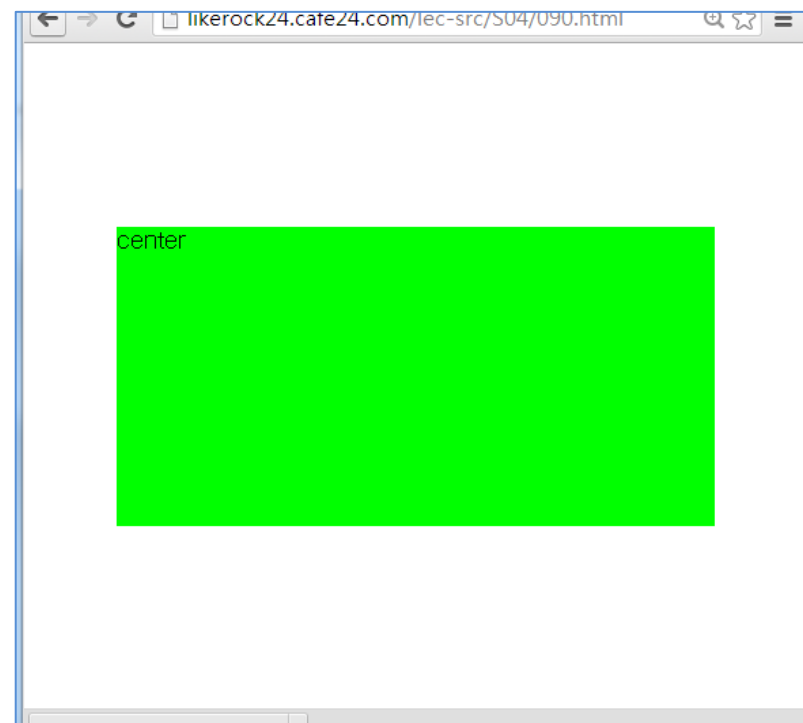
```
<div class="area">
  <div class="box a">box a</div>
  <div class="box b">box b</div>
  <div class="box c">box c</div>
</div>
```

⑥ margin활용 위치 지정.

absolute를 지정한 블록을 웹페이지 **중앙**에 배치하는 방법.

```
<style type="text/css">
*{margin: 0; padding: 0;}
.area{
    position: absolute;
    left: 50%;
    top: 50%;
    margin-left: -200px;
    margin-top: -100px;
    width: 400px;
    height: 200px;
    background-color: lime;
}
</style>
</head>
<body>

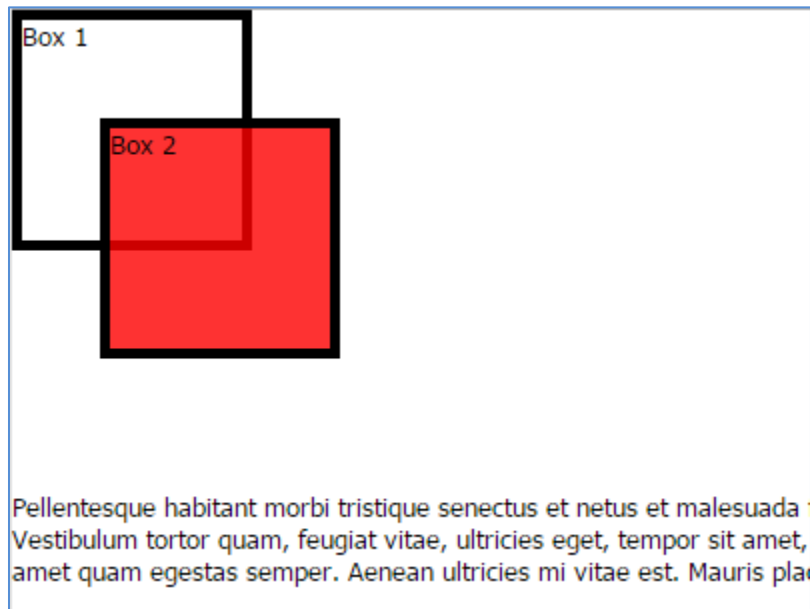
<div class="area">
    <p>center</p>
</div>
```



⑦ 보통 position을 지정하고 left, top, right, bottom등 위치 값을 지정하는 것이 일반적이지만, 그대신 '**margin**'을 활용하는 것이 효과적일 때가 많다.

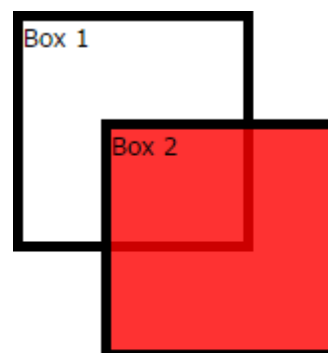
```
<style type="text/css">
*{margin: 0; padding: 0;}
body{font: 12px Tahoma,sans-serif;}
.box{
    position: relative;
    width: 100px;
    height: 100px;
    border: 5px solid;
}
.a{
    left: 0;
    top: 0;
}
.b{
    left: 40px;
    top: -60px;
    /*margin-top: -60px;*/
    background-color: rgba(255,0,0,0.8);
}
</style>
</head>
<body>

<div class="box a">Box 1</div>
<div class="box b">Box 2</div>
<p>Pellentesque habitant morbi tristique senectus et netus et malesuada
ac turpis egestas. Vestibulum tortor quam, feugiat vitae,
tempor sit amet, ante. Donec eu libero sit amet quam egest
ultricies mi vitae est. Mauris placerat eleifend leo.</p>
```



```
<style type="text/css">
*{margin: 0; padding: 0;}
body{font: 12px Tahoma,sans-serif;}
.box{
    position: relative;
    width: 100px;
    height: 100px;
    border: 5px solid;
}
.a{
    left: 0;
    top: 0;
}
.b{
    left: 40px;
    margin-top: -60px;
    background-color: rgba(255,0,0,0.8);
}
</style>
</head>
<body>

<div class="box a">Box 1</div>
<div class="box b">Box 2</div>
<p>Pellentesque habitant morbi tristique senectus et netus et malesuada
fames ac turpis egestas. Vestibulum tortor quam, feugiat vitae, ultricies
eget, tempor sit amet, ante. amet quam egestas semper. Aenean ultricies
mi placerat eleifend leo.</p>
```



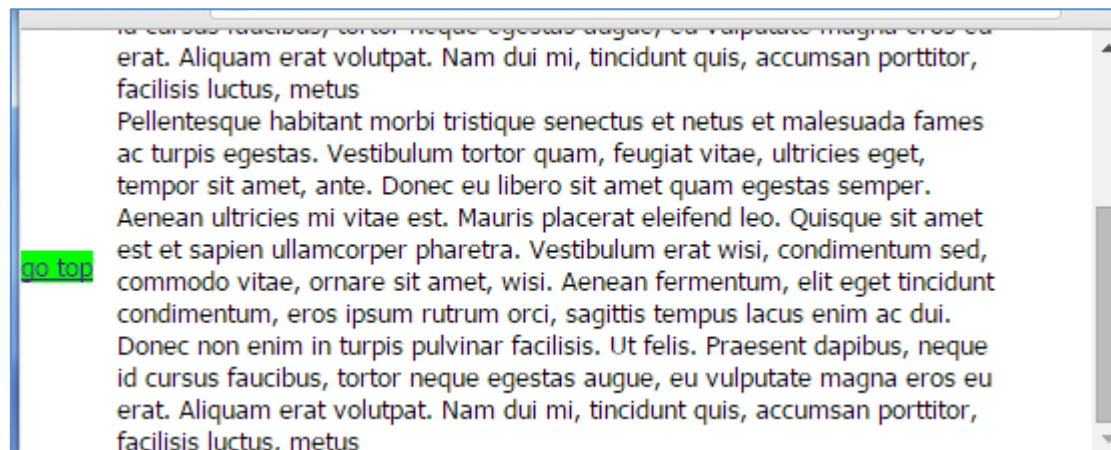
Pellentesque habitant morbi tristique senectus et netus et malesuada
Vestibulum tortor quam, feugiat vitae, ultricies eget, tempor sit an
amet quam egestas semper. Aenean ultricies mi vitae est. Mauris

⑧ position의 값이 'fixed' 라면.이 박스의 위치는 항상<body>를 기준으로 정렬된다. ※ ie6적용되지 않는다.

```
<style type="text/css">
*{margin: 0; padding: 0;}
body{font: 12px Tahoma, sans-serif;}
#wrap{
    width: 400px;
    margin: 0 auto;
}
.btn{
    position: fixed;
    left: 0;
    top: 100px;
    background-color: lime;
}
</style>
</head>
<body>

<p class="btn">
    <a href="#">go top</a>
</p>

<div id="wrap">
    <p>Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum tortor quam, feugiat vitae, ultricies eget, tempor sit amet, ante. Donec eu libero sit amet quam egestas semper. Aenean ultricies mi vitae est. Mauris placerat eleifend leo. Quisque sit amet est et sapien ullamcorper pharetra. Vestibulum erat wisi, condimentum sed, commodo vitae, ornare sit amet, wisi. Aenean fermentum, elit eget tincidunt condimentum, eros ipsum rutrum orci, sagittis tempus lacus enim ac dui. Donec non enim in turpis pulvinar facilisis. Ut felis. Praesent dapibus, neque id cursus faucibus, tortor neque egestas augue, eu vulputate magna eros eu erat. Aliquam erat volutpat. Nam dui mi, tincidunt quis, accumsan porttitor, facilisis luctus, metus</p>
    <p>Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum tortor quam, feugiat vitae, ultricies eget, tempor sit amet, ante. Donec eu libero sit amet quam egestas semper. Aenean ultricies mi vitae est. Mauris placerat eleifend leo. Quisque sit amet est et sapien ullamcorper pharetra. Vestibulum erat wisi, condimentum sed, commodo vitae, ornare sit amet, wisi. Aenean fermentum, elit eget tincidunt condimentum, eros ipsum rutrum orci, sagittis tempus lacus enim ac dui. Donec non enim in turpis pulvinar facilisis. Ut felis. Praesent dapibus, neque id cursus faucibus, tortor neque egestas augue, eu vulputate magna eros eu erat. Aliquam erat volutpat. Nam dui mi, tincidunt quis, accumsan porttitor, facilisis luctus, metus</p>
</div>
```



4 Position과 z-index

position을 지정한 여러 블록이 겹쳐지게되면, 나중에 작성한 블록이 위로 올라가게 된다.

이 경우 **z-index** 속성으로 위아래 위치를 조정할 수 있으며, 큰 값이 위로 올라가게 된다.

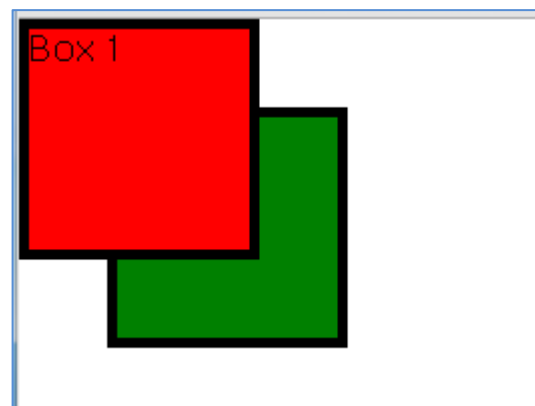
z-index에는 0, 1, 2, 3, 또는 -1, -2, -3 등의 정수만 가능.

일부 브라우저는 음수 인식이 안되므로 음수는 쓰지 않는 것이 안전하다.

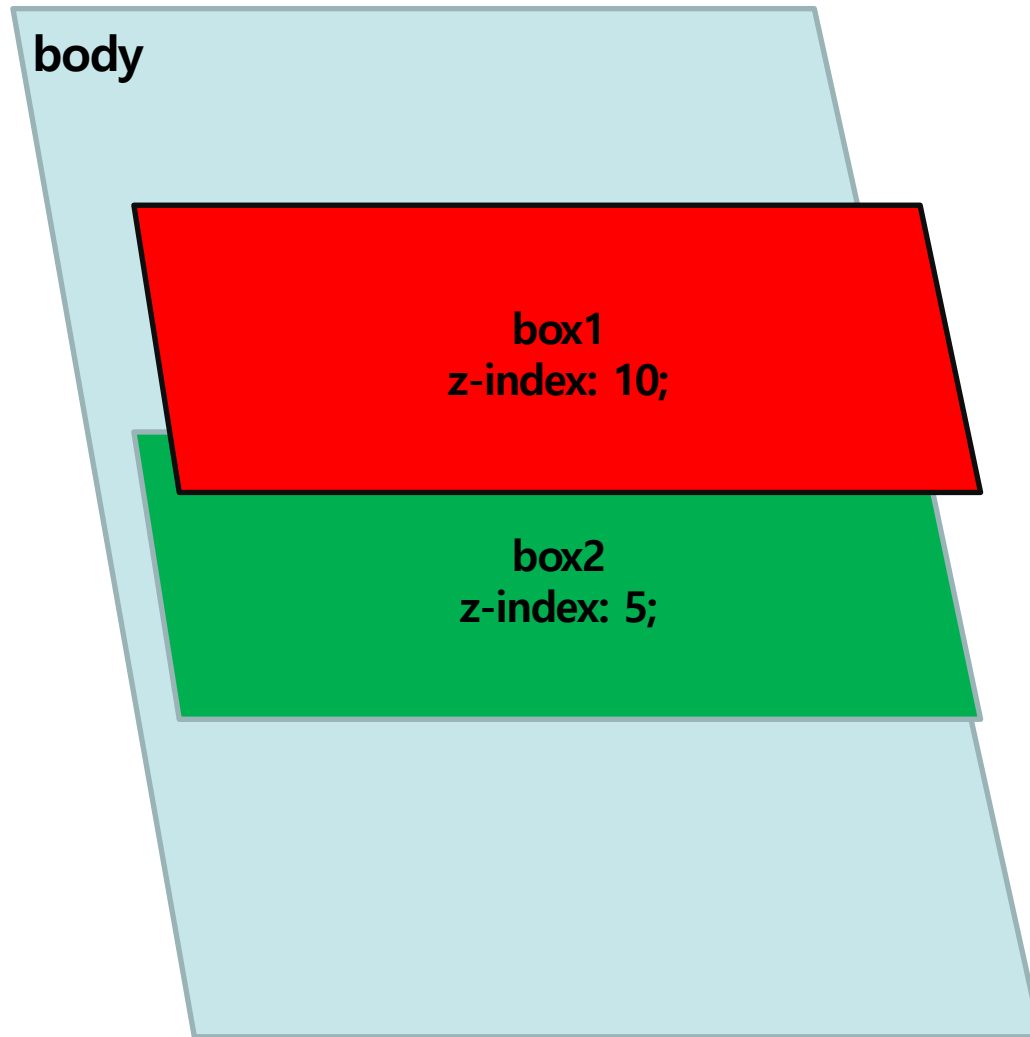
z-index 값을 지정할 경우 비교대상을 정확히 이해 해야한다. 간혹 특정 브라우저에서 버튼이나 메뉴가
클릭되지 않는다면 z-index의 값을 조정해 줄 필요가 있다.

```
<style type="text/css">
*{margin: 0; padding: 0;}
.box{
    position: absolute;
    border: 5px solid;
    width: 100px;
    height: 100px;
}
.a{
    left: 0;
    top: 0;
    z-index: 10;
    background-color: red;
}
.b{
    left: 40px;
    top: 40px;
    z-index: 5;
    background-color: green;
}
</style>
</head>
<body>

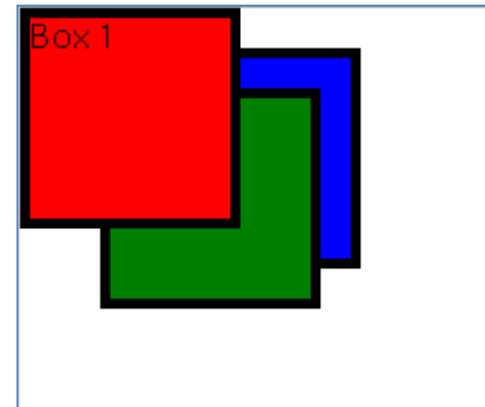
<div class="box a">Box 1</div>
<div class="box b">Box 2</div>
```



Z-INDEX



```
*{margin: 0; padding: 0;}
.box{
  position: absolute;
  border: 5px solid;
  width: 100px;
  height: 100px;
}
.area{
  position: relative;
  z-index: 110;
}
.a{
  left: 0;
  top: 0;
  z-index: 10;
  background-color: red;
}
.b{
  left: 40px;
  top: 40px;
  z-index: 5;
  background-color: green;
}
.c{
  left: 60px;
  top: 20px;
  z-index: 100;
  background-color: blue;
}
</style>
```



```
<div class="area">
  <div class="box a">Box 1</div>
  <div class="box b">Box 2</div>
</div>
<div class="box c">Box 2</div>
```

z-index는 인접한 요소끼리 비교해서 위치를 지정.
박스c가 아무리 z-index가 크다고 해서 위로 올라오진 않는다.

Review TEST

position_01

<http://codepen.io/alikerock/pen/zqMYXK>

Read more

- 리스트 1
- 리스트 2
- 리스트 3
- 리스트 4
- 리스트 5

CSS SUB-MENU

<http://codepen.io/alikerock/pen/QKeWNB>

About Us

Works

Services

Location

ceo

greeting

history

vision

position_02

<http://codepen.io/alikerock/pen/KzGwdz>

Prev

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s

Prev

9. 포지셔닝(positioning) 스타일

footer를 항상 밑에 위치하도록 수정.



1. 문서 내용이 전체 영역을 인식할 수 있도록 코드 추가

```
html,body{  
    width: 100%;  
    height: 100%;  
}
```

2. #wrap 영역에는 'min-height' 100%추가해서, footer를 body 아래쪽으로 밀어냅니다.
(ie6용으로는 '_height:100%' 필요함)

```
#wrap{  
    width: 800px;  
    min-height: 100%;  
    _height: 100%;  
    margin: 0 auto;  
    background-color: silver;  
}
```

3. 'container' footer 사이를 띄우기 위해서 'margin-top:30px 지정했지만, padding이 부드럽게 동작하므로 다음과 같이 수정.

```
#container{  
    padding-bottom: 30px;  
    *zoom: 1;  
    background-color: green;  
}
```

4. Footer를 항상 <body> 아래쪽에 붙어 있도록 해야 하는데, 이때 'absolute'가 아닌 'relative'를 지정해야 한다. 본문이 footer와 겹칠 때, absolute로는 footer를 밀어낼 수 없다. 위치는 margin-top으로 footer 높이 만큼을 끌어올려준다.

```
#footer{  
    position: relative;  
    margin-top: -50px;  
    height: 50px;  
    background-color: blue;  
}
```

5. 'margin-top으로 끌어 올린 부분은 본문과 겹쳐지게 된다. Container에서도 겹쳐져도 될 만큼의 padding을 확보한다.

```
#container{  
    /*padding-bottom: 30px;*/  
    padding-bottom: 80px;  
    *zoom: 1;  
    background-color: green;  
}
```

6. 본문의 내용이 적거나 많거나 상관없이 스크롤링이 자연스럽도록 세로 스크롤바의 위치를 항상 나타나게 한다.

```
html{  
    overflow-y: scroll;  
}
```