



HTML5 3강2- CSS 스타일 속성 기본

양 명 속

[now4ever7@gmail.com]



목차

- 위치 속성



7. 위치 속성

- 프로그램을 개발할 때는 요소의 위치를 2가지 방법으로 설정함
 - **절대적 위치 좌표** : 요소의 x 좌표와 y좌표를 설정해 절대적 위치를 지정
 - **상대적 위치 좌표** : 요소를 입력한 순서대로 상대적으로 위치를 지정
 - 지금까지 상대적 위치 좌표를 사용해 HTML 페이지를 만들었다



CSS에서 위치와 관련된 Positioning 방식

- 일반흐름(normal flow)
 - 사용자가 기본적으로 가지고 있는 positioning
 - positioning 방식을 지정하지 않으면 일반 흐름방식 적용
- 절대위치(absolute positioning)
 - absolute/relative
- float 방식



위치 지정 속성

- top, bottom, left, right(사이즈 값을 속성 값으로 사용)
 - top : 위에서부터 떨어진 거리를 나타내는데 사용
 - 예) top:20px => 부모 엘리먼트의 윗변을 기준으로 위에서부터 아래로 20px 떨어진 위치에 해당 엘리먼트의 윗변이 위치함
- 위치지정 속성들은 절대, 상대 위치 지정과 함께 사용

예제

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

```
*{margin:0;}
```

```
div{width:150px;height:120px;text-align:center;position:absolute;}
```

```
.box1{background:pink;top:30px; }
```

```
.box2{background:skyblue;left:40px;bottom:20px; }
```

```
.box3{background:yellow;right:0px;top:50%;}
```

```
</style>
```

```
</HEAD>
```

```
<BODY>
```

```
<div class="box1">
```

```
top:30px
```

```
</div>
```

```
<div class="box2">
```

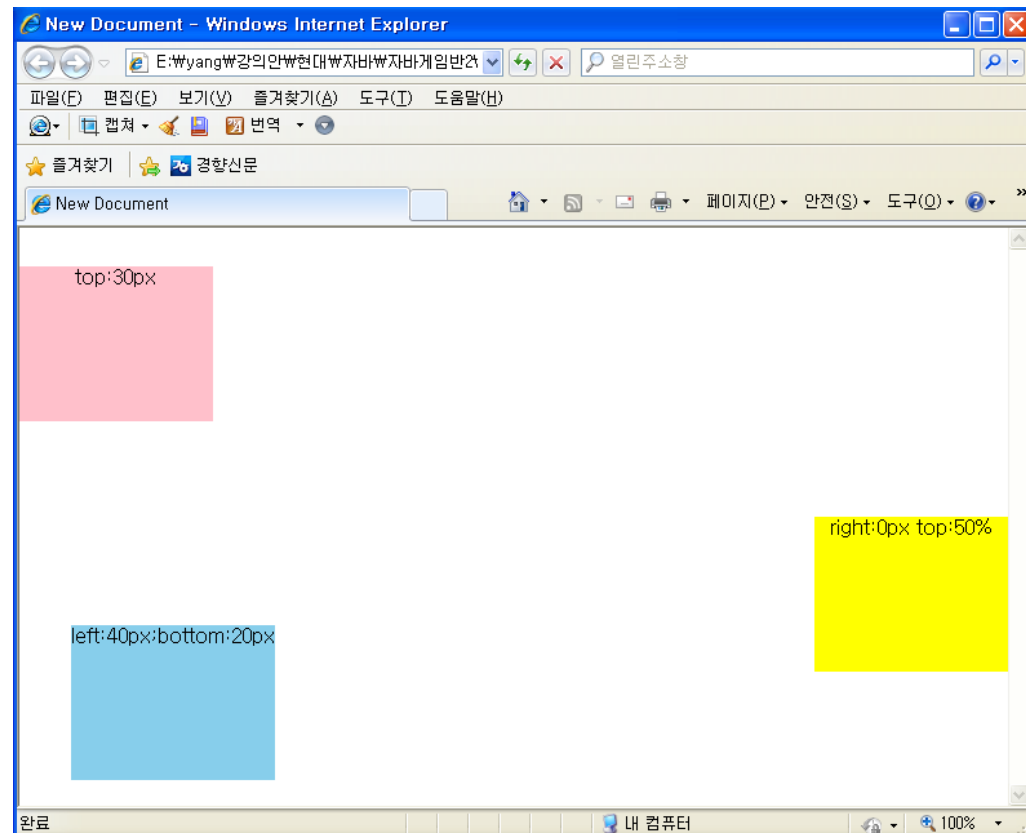
```
left:40px;bottom:20px
```

```
</div>
```

```
<div class="box3">
```

```
right:0px top:50%
```

```
</div>
```



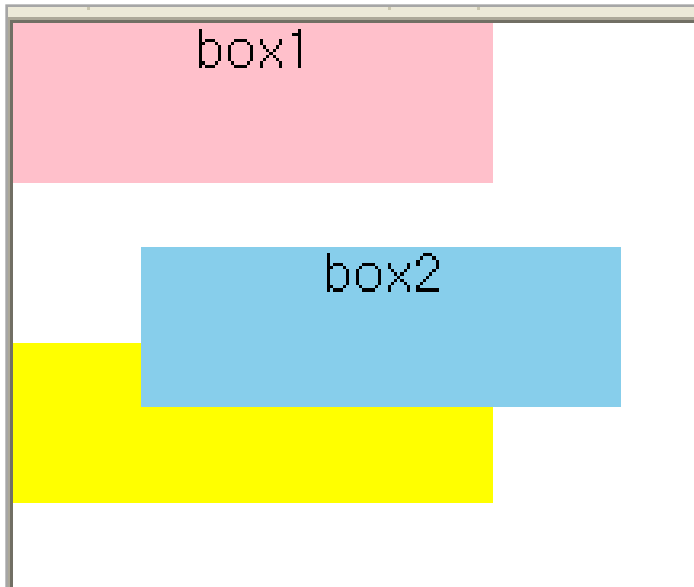


position 관련 스타일 속성

- position : 엘리먼트의 위치 지정 방식, 흐름의 종류를 정함
 - static, relative, absolute
- [1] static
 - position 의 기본값
 - 엘리먼트가 본래 가지고 있는 흐름을 가짐
 - 여러 가지 좌표값을 지정하여 위치를 지정하기 위해서는 속성값을 변경
- [2] relative
 - static 속성값과 기본적으로 같은 흐름을 가짐
 - 위치 조절을 위한 좌표값을 가질 수 있음
 - 위치의 기준점을 상대적으로 잡게 됨
 - 기본값인 static일 때의 위치를 기준으로 이동한 위치값을 가지게 됨
 - 엘리먼트의 원래 위치를 기준으로 위치 이동

예제

- box2를 위에서 20px, 왼쪽에서 40px으로 이동시키기





예제

```
<style type="text/css">
  *{margin:0;}
  div{width:150px;height:50px;text-align:center;}
  .box1{background:pink; }
  .box2{background:skyblue;left:40px;top:20px;position:relative; }
  .box3{background:yellow;}
</style>
</HEAD>

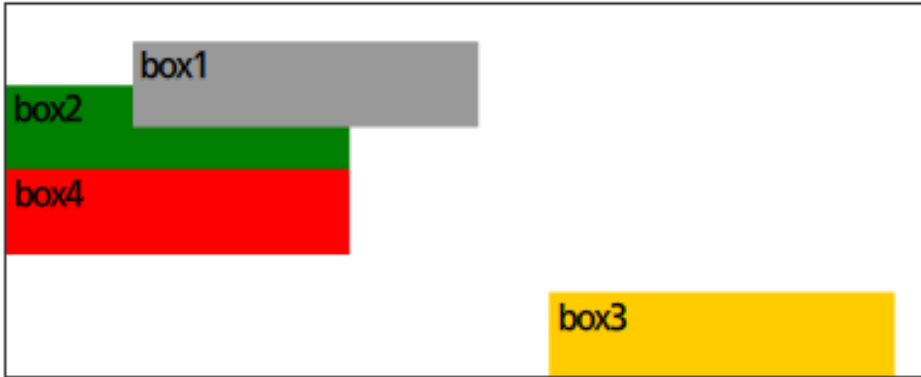
<BODY>
<div class="box1">
  box1 - static
</div>
<div class="box2">
  box2 - relative
</div>
<div class="box3">
  box3 - static
</div>
```



position 관련 스타일 속성

- [3]absolute
 - 엘리먼트의 순서에 따른 흐름을 무시하고 절대적인 위치를 가짐
 - 다른 엘리먼트들은 절대 위치를 가진 엘리먼트를 인식하지 않고 겹쳐 나옴
 - 부모 엘리먼트를 기준으로 위치값을 가짐
- relative/absolute 비교
 - relative
 - 엘리먼트의 원래 위치를 기준으로 위치 이동
 - 엘리먼트의 원래의 높이와 넓이 값이 적용
 - absolute
 - 부모 엘리먼트를 기준으로 위치가 이동
 - 넓이와 높이 값이 사라짐(원래 영역이 사라짐)

relative/absolute 비교



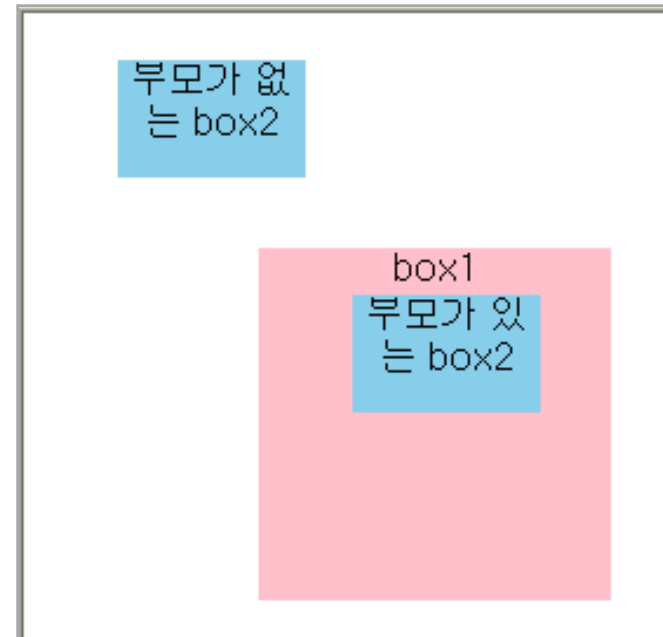
- box1 : relative – 원래 영역이 남아있음
- box3 : absolute – 원래 영역이 사라짐

absolute의 경우 부모가 **absolute**나 **relative**여야 부모 기준으로 움직임
부모가 **static**이면 부모 무시하고, **body**기준으로 움직임

예제

```
<style type="text/css">
  *{margin:0;}
  div{ width:150px;height:150px;text-align:center;}
  .box1{background:pink; position:absolute; top:100px;left:100px; }
  .box2{background:skyblue; left:40px;top:20px; width:80px;height:50px;
        position:absolute;
  }
</style>
</HEAD>

<BODY>
<div class="box1">
  box1
  <div class="box2">
    부모가 있는 box2
  </div>
</div>
<div class="box2">
  부모가 없는 box2
</div>
```





position 관련 스타일 속성

- z-index
 - 엘리먼트의 z-좌표 값을 처리
 - position 값이 absolute, relative 인 경우 적용
 - 정수 및 음수값을 가지게 되며, 숫자가 커질수록 위에 보여짐
 - 예) z-index값이 10인 엘리먼트보다 z-index값이 100인 엘리먼트가 위로 보여짐

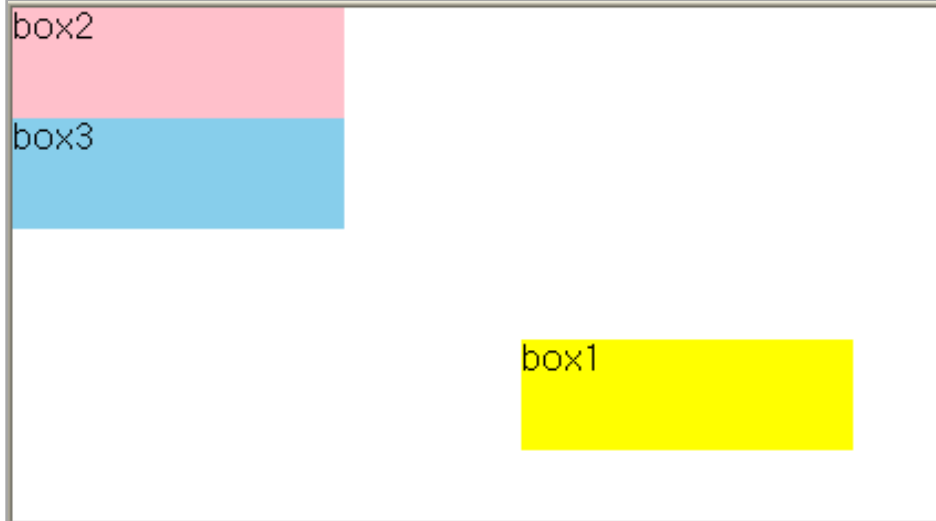


position 관련 스타일 속성 정리

- 레이어의 위치 : css스타일 속성 중 position, z-index 속성을 이용하여 조정

속성	속성 값	설명
position	absolute	-부모가 없는 경우 : 레이어의 위치를 html 문서의 왼쪽 위를 기준으로 함 - 부모가 있는 경우 : 레이어의 위치를 레이어를 포함하는 부모 태그의 왼쪽 위를 기준으로 함
	relative	엘리먼트의 원래 위치를 기준으로 위치 이동
	static	항상 레이어가 만들어진 위치에 나타나게 함
z-index	숫자	레이어의 수직 위치를 지정 수직 위치 : 레이어가 보이는 순서. z-index 값이 클수록 가장 위에 위치

레이어의 위치-absolute

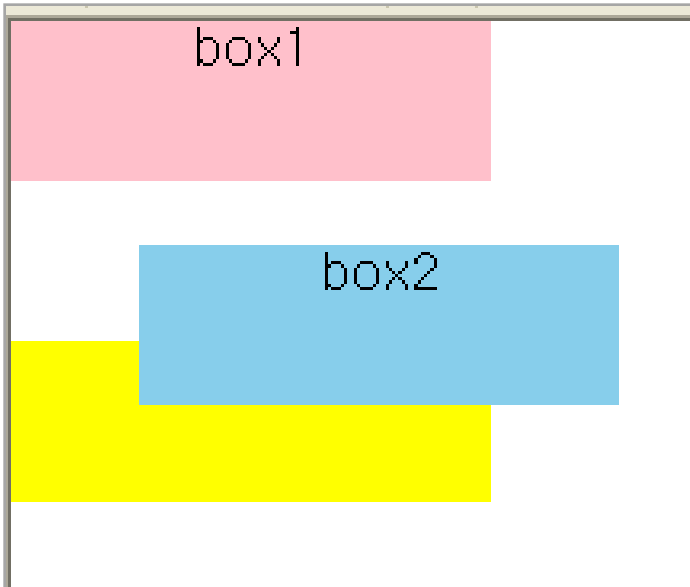


box1 => absolute

```
position: absolute;  
left: 230px; top: 150px;
```

원래 영역이 사라지고,
부모 엘리먼트의 위치를 기준으로 위
치 이동

레이어의 위치-relative



box2 => relative

원래 영역이 남아있고,
엘리먼트의 원래 위치를 기준으로 위치 이동

7. 위치 속성

- absolute

- 가장 가까운 부모 요소의 위치를 기준으로 상하좌우로 이동
- 부모 요소의 위치가 정해져 있지 않다면 브라우저 창의 왼쪽 위 모서리를 기준으로 위치를 계산함

- [1] position 속성

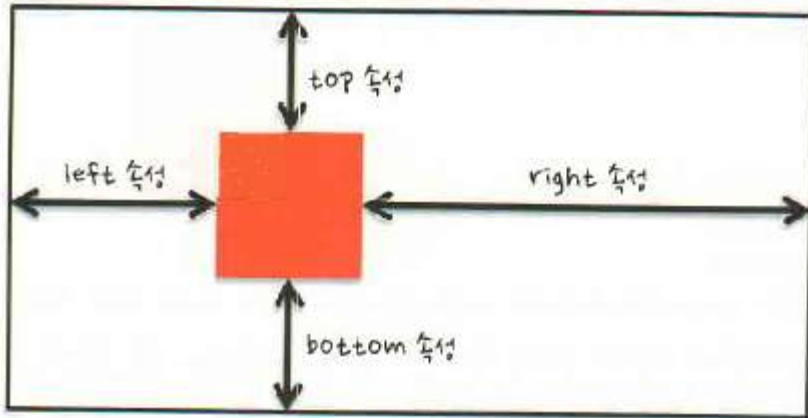
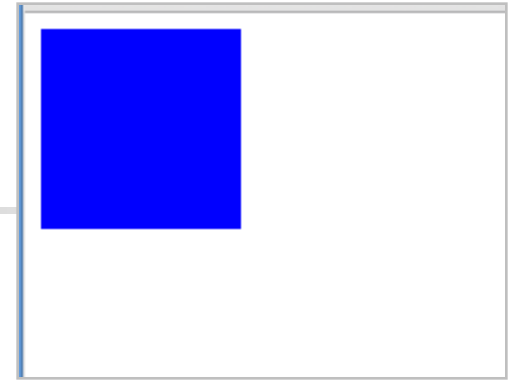
- HTML 태그의 위치 설정 방법을 변경할 때는 position 속성을 사용함
- 키워드 : absolute, relative, static
- 상대적 위치 좌표를 사용할 때는 position 스타일 속성에 static 또는 relative 키워드를 입력
 - static - 태그가 "위에서 아래로" 와 "왼쪽에서 오른쪽으로" 순서에 맞게 배치됨 (direction 속성을 사용해 "오른쪽에서 왼쪽"으로 변경할 수 있음)
 - relative - static 키워드로 초기 위치가 지정된 상태에서 상하좌우로 이동할 수 있음
- 절대적 위치 좌표를 사용할 때는 absolute 키워드를 입력

키워드	설명
static	태그가 위에서 아래로 순서대로 배치됨
relative	초기 위치 상태에서 상하좌우로 위치를 이동함
absolute	절대적 위치 좌표를 설정함

```

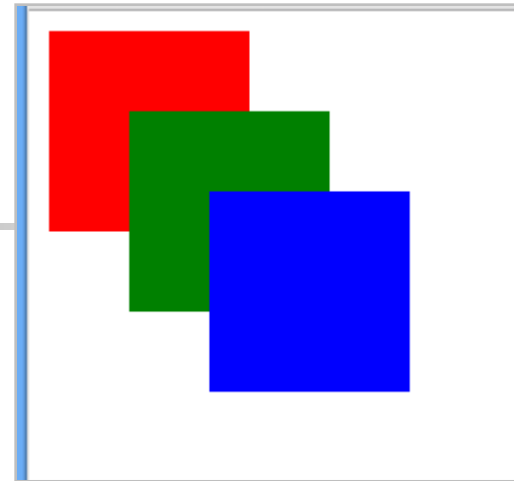
<head>
  <title>CSS3 Property Basic</title>
  <style>
    .box {
      width: 100px; height: 100px;
      position: absolute;
    }
    .box:nth-child(1) { background-color: red; }
    .box:nth-child(2) { background-color: green; }
    .box:nth-child(3) { background-color: blue; }
  </style>
</head>
<body>
  <div class="box"></div>
  <div class="box"></div>
  <div class="box"></div>
</body>

```



- **relative** - 원래 그 요소가 표시되어야 할 위치가 기준이 됨
- **absolute** - 부모 요소의 왼쪽 위 모서리가 기준이 됨

```
<head>
  <title>CSS3 Property Basic</title>
  <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>
</head>
<body>
  <div class="box"></div>
  <div class="box"></div>
  <div class="box"></div>
</body>
```



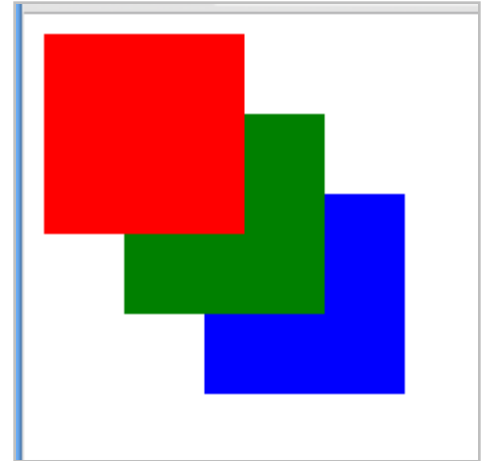
빨간색 상자가 가장 아래에 있고, 파란색 상자가 가장 위에 있다
기본으로HTML 페이지의 뒤에 입력한 태그가 상위에 올라감

7. 위치 속성

■ [2] z-index 속성

- 앞 예제에서 뒤에 입력한 파란색 사각형이 위로 올라왔는데, 이러한 순서를 변경하고 싶을 때는 z-index 속성을 사용함
- z-index 속성 - 숫자를 입력하며 숫자가 클수록 앞에(위에) 위치함

```
<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>
```



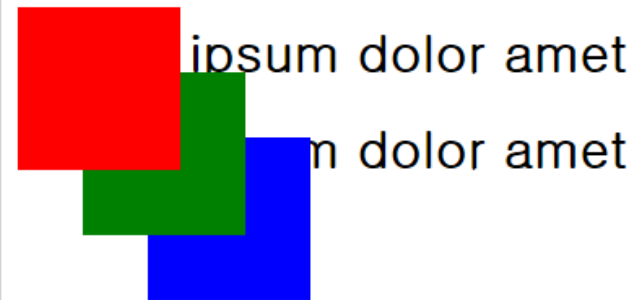
z-index 속성이 큰 태그가 앞으로 올라옴

7. 위치 속성

■ [3] 위치 속성과 관련된 공식

```
<head>
<title>CSS3 Property Basic</title>
<style>
.box {
  width: 100px; height: 100px;
  position: absolute;
}
.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>
</head>
```

```
<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>
```



두 가지 문제

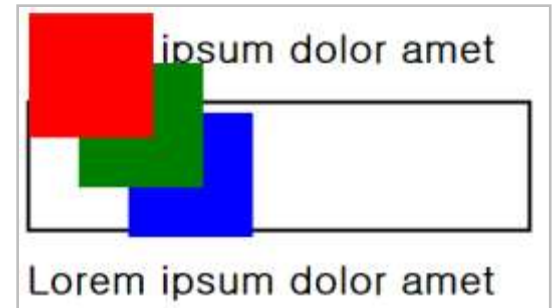
1. h1 태그 두 개가 붙어있다(div 태그가 영역을 차지하지 않는다)
2. 색상이 적용된 상자가 자신의 부모를 기준으로 위치를 잡지 않는다

7. 위치 속성

- position 속성에 absolute 키워드를 적용하면 부모 태그가 영역을 차지하지 않음
 - 자손의 position 속성에 absolute 키워드를 적용할 경우는 부모 태그에 몇 가지 처리를 해야 함
- 1) 자손의 position 속성을 absolute 키워드로 적용하면 부모에게 height 속성을 입력함

- 부모 태그가 영역을 차지하게 만들 수 있음

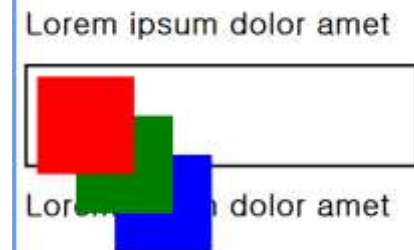
```
body > div {  
  width: 400px; height: 100px;  
  border: 3px solid black;  
}
```



- 2) 자손의 position을 absolute 로 적용하면 부모의 position을 relative 로 적용함

- 자손 태그가 부모의 위치를 기준으로 절대 좌표를 설정함

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



7. 위치 속성

visible - 기본값, 내용이 영역을 바깥으로 벗어나 보이게 됨
=> 다른 내용과 겹칠 수 있음
hidden - 초과된 내용을 숨김
scroll - 영역에 스크롤을 부여하여 내용을 스크롤하게 함
(스크롤이 무조건 나타남)
auto - 초과되는 내용이 있을 때만 스크롤을 보여줌

■ [4] overflow 속성

- 내부의 요소가 부모의 범위를 벗어나는 때 어떻게 처리할지 지정하는 속성
- 적용 가능한 키워드

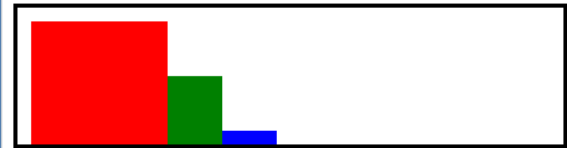
스타일 이름	설명
hidden	영역을 벗어나는 부분을 보이지 않게 만든다
scroll	영역을 벗어나는 부분을 스크롤로 만든다

- 앞의 예제를 보면 색상이 적용된 사각형이 부모의 범위를 벗어나 있음
- 이러한 div 태그의 overflow 속성에 hidden 키워드를 입력함

```
body > div {  
  width: 400px; height: 100px;  
  border: 3px solid black;  
  position: relative;  
  overflow: hidden;  
}
```

부모의 영역을 벗어난 부분이 모두 숨겨짐

Lorem ipsum dolor amet



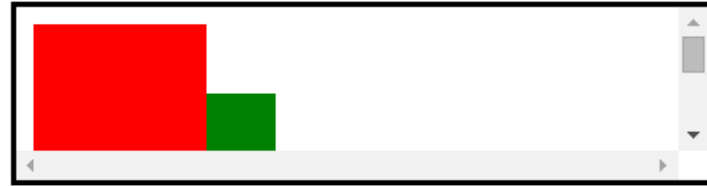
Lorem ipsum dolor amet

7. 위치 속성

```
body > div {  
  width: 400px; height: 100px;  
  border: 3px solid black;  
  position: relative;  
  overflow: scroll;  
}
```

div 태그에 스크롤이 생김
범위를 벗어나는 부분은 스크롤을 사용해 볼
수 있음

Lorem ipsum dolor amet

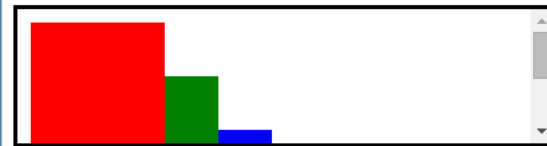


Lorem ipsum dolor amet

- overflow 속성에 scroll 키워드를 적용하면 무조건 모든 축에 스크롤이 생성됨
- 특정한 방향으로만 스크롤을 생성할 때는 overflow-x 속성과 overflow-y 속성을 사용함

```
body > div {  
  width: 400px; height: 100px;  
  border: 3px solid black;  
  position: relative;  
  overflow-y: scroll;  
}
```

Lorem ipsum dolor amet



Lorem ipsum dolor amet

8. float 속성

- **float** - 웹 요소를 문서 위에 떠 있게 하는 것
- '떠 있다'는 의미 - 왼쪽 구석이나 오른쪽 구석에 요소가 배치된다는 것
- 문서의 레이아웃을 조절하려고 할 때 많이 사용

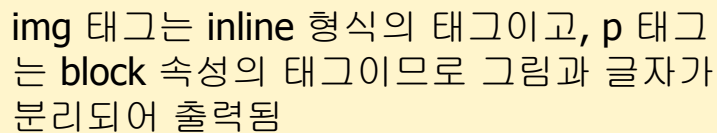
■ float 속성의 키워드

키워드	설명
left	태그를 왼쪽에 붙인다. (문서의 왼쪽으로 배치)
right	태그를 오른쪽에 붙인다.

■ [1] float 속성 개요

■ 부유하는 대상을 만들 때 사용하는 스타일 속성

```
<head>
  <title>Float Style Property</title>
  <style>
  </style>
</head>
<body>
  
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
  <p>In hac habitasse platea dictumst. Donec lobortis augue a metus.</p>
</body>
```



In hac habitasse platea dictumst. Donec lobortis augue a metus.

img 태그에 float 속성을 적용하면 이미지가 글자 위에 부유하고 있게 됨

float 속성을 사용하면 그림을 글자 위에 띄울 수 있다

원래 **float** 속성은 이러한 목적으로 만들었지만, 요즘에는 웹 페이지의 레이아웃을 만들 때 더 많이 사용함



In hac habitasse platea dictumst. Donec lobortis augue a metus. In hac habitasse platea dictumst. Donec lobortis augue a metus. In hac habitasse platea dictumst. Donec lobortis augue a metus. In hac habitasse platea dictumst. Donec lobortis augue a metus. In hac habitasse platea dictumst. Donec lobortis augue a metus.

8. float 속성

- [2] float 속성을 사용한 수평 정렬
 - float 속성을 사용하면 태그를 수평으로 정렬할 수 있음
 - float 속성을 사용해 수평 정렬할 때는 주의할 점이 있다

```
<head>
  <title>Float Style Property</title>
  <style>
    .box {
      width: 100px; height: 100px;
      background-color: red;
      margin: 10px; padding: 10px;
      /* 태그를 왼쪽으로 붙입니다. */
      float: left;
    }
  </style>
</head>
<body>
  <div class="box">1</div>
  <div class="box">2</div>
</body>
```

float: right 의 경우 첫번째에 위치한 **div** 태그가 오른쪽에 붙고, 두 번째에 위치한 **div** 태그가 오른쪽에 이어서 붙는다

```
<style>
  .box {
    width: 100px; height: 100px;
    background-color: red;
    margin: 10px; padding: 10px;
    /* 태그를 오른쪽으로 붙입니다. */
    float: right;
  }
</style>
```



`float: left;`



`float: right;`



clear 속성

- clear 속성 - float 속성 무효로 만들기
 - float 속성을 이용하여 웹 페이지 요소를 왼쪽이나 오른쪽에 배치하면 그 다음에 넣는 다른 요소들에도 똑같은 속성이 전달됨

clear : 속성값;

키워드	설명
left	왼쪽으로 떠 있지 않도록 함
right	오른쪽으로 떠 있지 않도록 함
both	왼쪽이나 오른쪽 어디에도 떠 있지 않도록 함
none	왼쪽과 오른쪽 모두에 뜰 수 있음. 기본 값



float

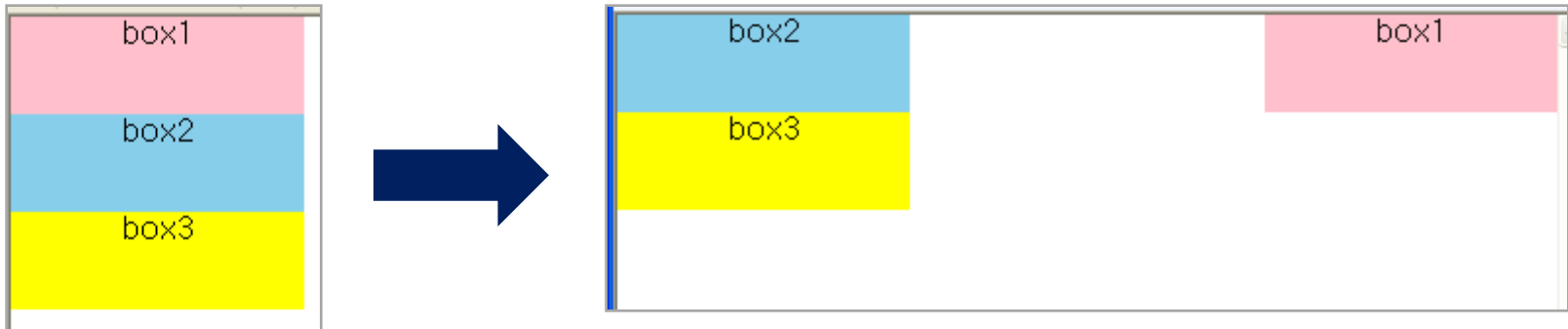
- (1) float

- none (기본값)
- left (왼쪽으로 띄움)
- right (오른쪽으로 띄움)

- 그 박스를 둘러싸고 있는 박스나 float가 지정된 다른 박스의 왼쪽이나 오른쪽 끝 면에 닿도록 이동
- float로 띄운 박스는 문서의 일반 흐름을 타지 않음
=> 문서의 기본 흐름 안에 있는 다른 블록 엘리먼트들은 float가 적용된 박스가 없는 것처럼 동작

예제

- box1 이 float:right된 경우





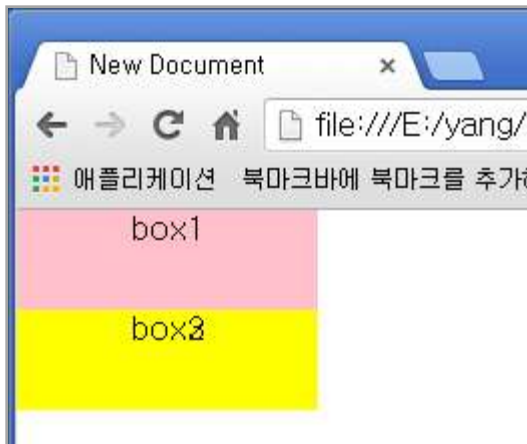
예제

```
<style type="text/css">
  *{margin:0;}
  div{ width:150px;height:50px;text-align:center; }
  .box1{ background:pink; float:right; }
  .box2{ background:skyblue; }
  .box3{ background:yellow; }
</style>
</HEAD>
```

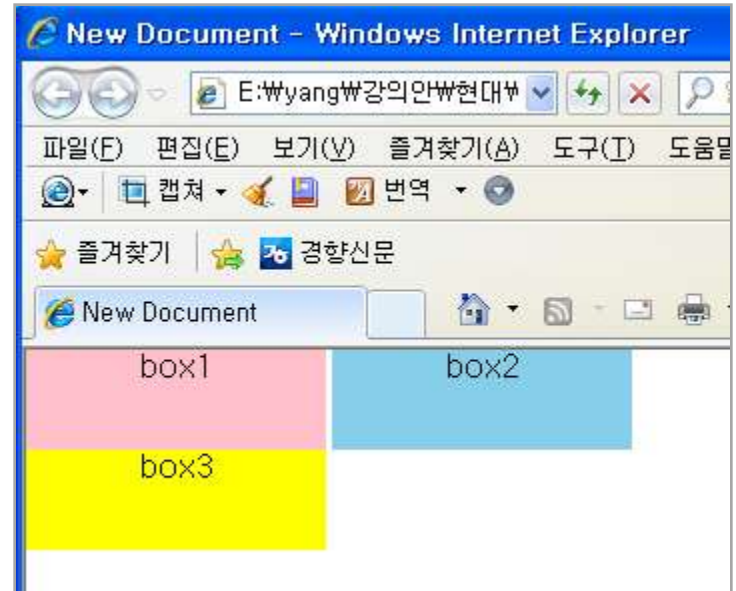
```
<BODY>
<div class="box1">
  box1
</div>
<div class="box2">
  box2
</div>
<div class="box3">
  box3
</div>
```


예제

- box1이 float:left된 경우



크롬, 파이어폭스 화면 - 정상



익스플로러 화면 - 비정상

box1에만 float:left를 준 상태지만
box1, box2에 float:left를 준 것과 같은 화면

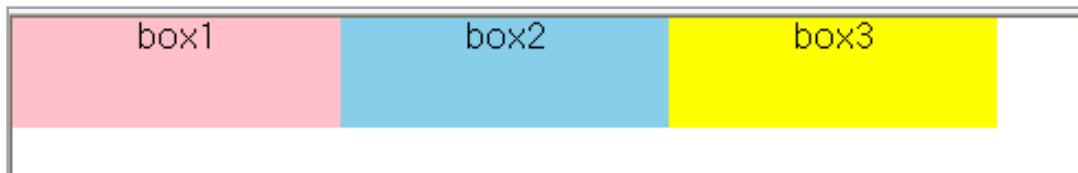
엘리먼트가 float 속성을 가지는 경우
바로 따라 오는 엘리먼트가 바로 전 엘리먼트의
float 속성을 따라가는 버그

예제

- 여러 개의 엘리먼트에 float 지정
 - 세 개의 박스를 모두 왼쪽으로 float 한 경우

```
.box1{  
    background:pink;  
    float:left;  
}  
.box2{  
    background:skyblue;  
    float:left;  
}  
.box3{  
    background:yellow;  
    float:left;  
}
```

- 부모 엘리먼트의 가로 폭이 충분하지 않을 경우
 - 충분한 공간이 있는 밑으로 떨어짐
 - float된 박스들의 높이가 다르다면 float된 다른 엘리먼트에 달라붙을 수 있음





예제

■ inline(텍스트)와의 조합

box4 float된 박스 옆에 인라인 엘리먼트가 오면 float된 박스 자리만큼 짧아지면서 그 박스 옆으로 흘러가게 됩니다. 이처럼 float와 inline을 활용하면 텍스트 문서를 그럴 듯 하게 꾸며 줄 수 있습니다.

```
.box4{  
    width:50px;height:50px;font-weight:bold;  
    background:pink;  
    float:left;  
}
```

```
<div class="box4">  
    box4  
</div>
```

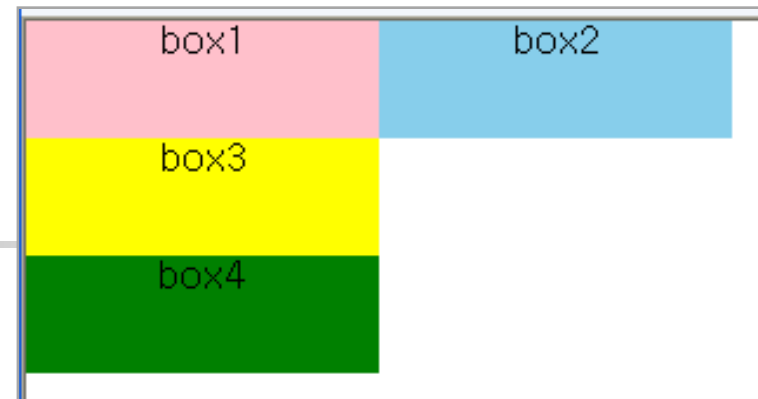
float된 박스 옆에 인라인 엘리먼트가 오면 float된 박스 자리만큼 짧아지면서 그 박스 옆으로 흘러가게 됩니다. 이처럼 float와 inline을 활용하면 텍스트 문서를 그럴 듯 하게 꾸며 줄 수 있습니다.



clear

- (2) float 해제 (clear)
- clear
 - float로 생긴 흐름을 초기화시키는 역할
 - 키워드 : left, right, **both**, none
 - 초기값 : none
 - left는 float:left에, right는 float:right 에 대응
 - both는 left와 right에 모두 대응

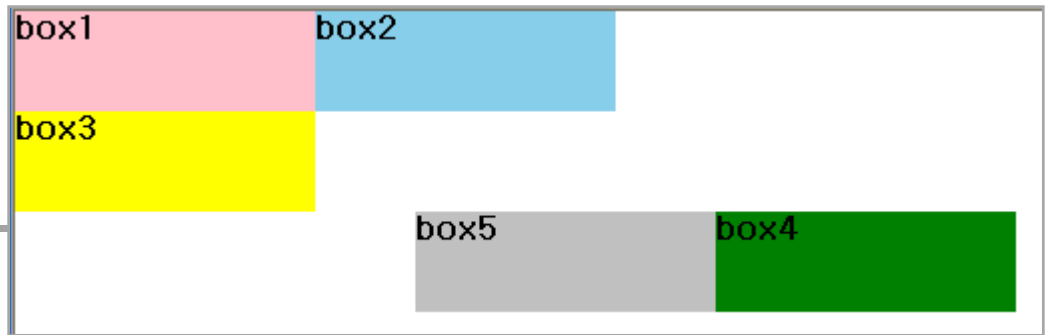
예제



```
<style type="text/css">
  *{margin:0;}
  div{width:150px;height:50px;text-align:center }
  .box1{ background:pink;float:left;    }
  .box2{ background:skyblue;float:left;  }
  .box3{ background:yellow;clear:both;   }
  .box4{ background:green;    }
</style>
```

```
<div class="box1">
  box1
</div>
<div class="box2">
  box2
</div>
<div class="box3">
  box3
</div>
<div class="box4">
  box4
</div>
```

예제

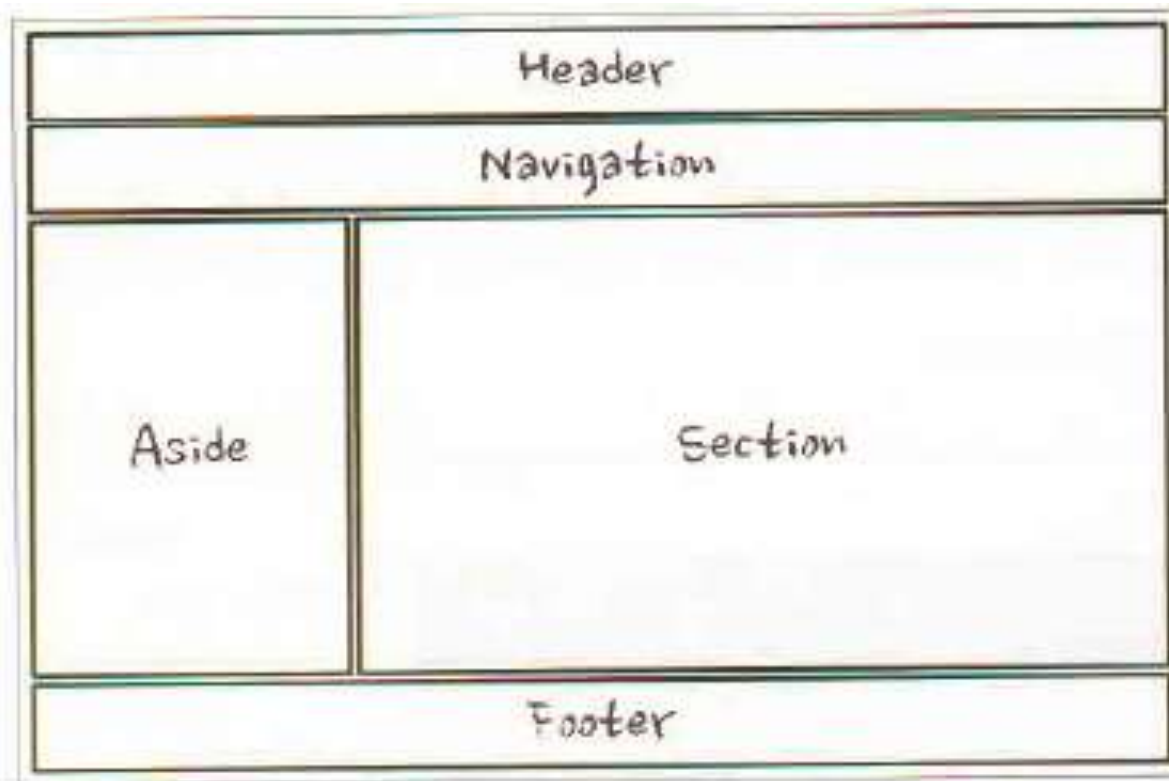


```
div#wrap{width:500px;}
*{margin:0;}
div{width:150px;height:50px;font-weight:bold;}
.box1{background:pink;float:left;}
.box2{background:skyblue;float:left;}
.box3{background:yellow;clear:both;}
.box4{background:green;float:right;}
.box5{background:silver;float:right;}
```

```
<div id="wrap">
  <div class="box1">
    box1
  </div>
  <div class="box2">
    box2
  </div>
  <div class="box3">
    box3
  </div>
  <div class="box4">
    box4
  </div>
  <div class="box5">
    box5
  </div>
</div>
```

8. float 속성

- [3] float 속성을 사용한 레이아웃 구성
 - 자손에 float 속성을 적용하면 부모의 overflow 속성에 hidden 키워드를 적용함
 - 레이아웃을 구성할 때 가장 많이 사용하는 공식



```

<head>
  <title>CSS3 Property Basic</title>
  <style>
  </style>
</head>
<body>
  <div id="header"></div>
  <div id="navigation"></div>
  <div id="wrap">
    <div id="aside"></div>
    <div id="section"></div>
  </div>
  <div id="footer"></div>
</body>

```

HTML 페이지 구성

Header

Navigation

Aside

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Section

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Footer

```

<head>
  <title>CSS3 Property Basic</title>
</head>
<body>
  <div id="header"><h1>Header</h1></div>
  <div id="navigation"><h1>Navigation</h1></div>
  <div id="wrap">
    <div id="aside">
      <h1>Aside</h1>
      <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
    </div>
    <div id="section">
      <h1>Section</h1>
      <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
    </div>
  </div>
  <div id="footer"><h1>Footer</h1></div>
</body>

```

레이아웃의 내용 입력

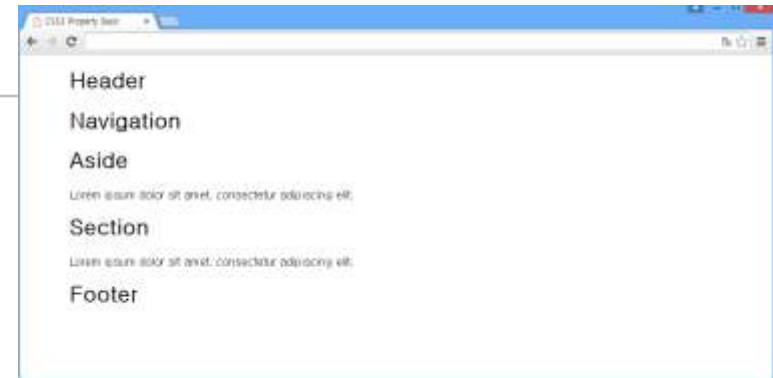

```
<style>
/* body 태그를 중앙 정렬 */
body {
  width: 960px;
  margin: 0 auto;
}
</style>
```

스타일 적용

태그에 **width** 속성 적용,
margin-left 속성과 **margin-right** 속성을 **auto** 로 적용
하면 자동으로 중앙 정렬됨

```
<style>
/* body 태그를 중앙 정렬 */
body {
  width: 960px;
  margin: 0 auto;
}
#aside {
  width: 200px;
  float: left;
}

#section {
  width: 760px;
  float: left;
}
</style>
```



#wrap 태그 내부의 영역은 왼쪽과 오른쪽으로 구분
되므로 각각의 태그에 **width** 속성을 적용하고 **float**
속성을 적용

```

<style>
/* body 태그를 중앙 정렬합니다. */
body {
  width: 960px;
  margin: 0 auto;
}
#aside {
  width: 200px;
  float: left;
}
#section {
  width: 760px;
  float: left;
}
#wrap { overflow: hidden; }
</style>

```

#footer 태그가 #section 태그 아래에 붙어 있다
#wrap 태그 내부의 요소에 float 속성을 적용했는데,
img 태그에 float 속성을 적용해 이미지가 부유했던
것처럼 #footer 태그의 글자 위에 #wrap 태그 내부의
요소가 부유하는 것임

=> 이러한 부유를 막을 때는 float 속성을 적용한 태
그의 부모에 overflow 속성으로 hidden 을 적용

#footer 태그의 clear 스타일 속성에 both 키워드를
적용해도 마찬가지로 기능을 수행
-overflow 속성을 사용하는 방법을 더 많이 사용함
-이 방법을 One True Layout 방식이라고 함

Header

Navigation

Aside

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

Section

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

Footer



화면의 레이아웃 구성

- 화면의 레이아웃을 구성하는 방법
 - [1] position 값을 활용한 레이아웃 구성
 - [2] float를 이용한 상대 위치를 사용한 방법
- CSS 초기화 ← 화면 레이아웃
- 엘리먼트의 기본 스타일값
 - <h1> 엘리먼트
 - 큰 폰트 크기
 - font-weight:bold값
 - list 엘리먼트 -
 - margin값
 - padding값
 - 리스트 스타일값
- 엘리먼트의 기본 스타일값은 브라우저의 종류나 버전별로 미묘한 차이가 있기 때문에 제대로 된 화면출력을 하기 위해서는 반드시 초기화가 필요함



clear.css

```
body,div,dl,dt,dd,ul,ol,li,h1,h2,h3,h4,h5,h6,pre,form,input,textarea,p,blockquote,th,td {
    margin:0;
    padding:0;
}
table {
    border-collapse:collapse;
}
img {
    border:0;
}
ol,ul {
    list-style:none;
}
caption,em,strong,th {
    font-style:normal;
    font-weight:normal;
}
caption,th {
    text-align:left;
}
```

- 공통 선택자(*)를 사용하여 처리하기도 함
- 프로젝트에 따라 **headings**의 **font-weight** 속성을 그대로 사용하기도 함
- 여러 가지 스타일값을 사용하는 경우에는 초기화 **CSS**에서 해당 사항을 빼고 사용
- 초기화 **CSS**는 항상 가장 처음 읽어들이게 해서 실제로 사용될 스타일들이 적용될 수 있게 해줌

```
h1,h2,h3,h4,h5,h6 {
    font-size:100%;
    font-weight:normal;
}
```



화면 레이아웃

■ float layout

- float 과 clear 속성을 활용하여 엘리먼트의 상대적 위치를 조절하여 화면 레이아웃을 잡아주는 방법
- 콘텐츠가 유동적이라도 크게 문제가 없어 가장 일반적으로 사용되는 레이아웃 기법
- 특정한 좌표값을 사용하는 것이 아니기 때문에 동적으로 위치를 변화시키거나 엘리먼트의 겹침을 사용하기는 어려움

■ position layout

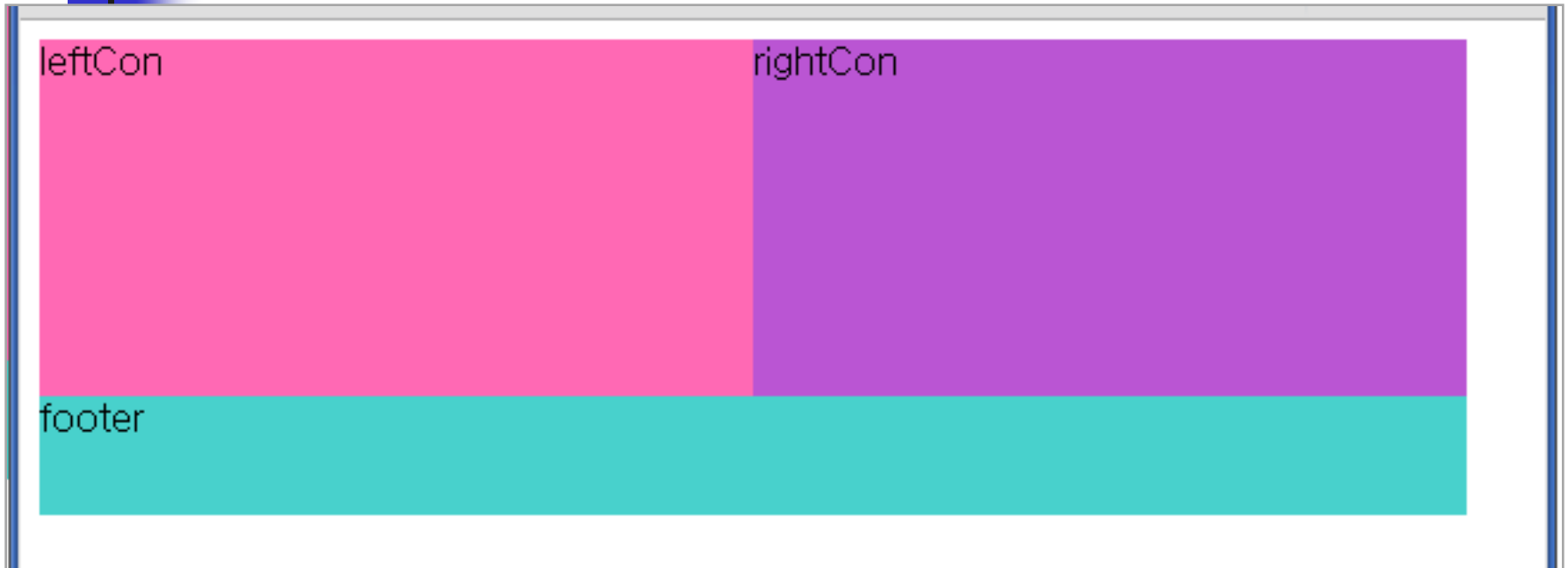
- position값과 top, left, right, bottom 값들을 활용하여 엘리먼트의 위치를 조절하여 화면 레이아웃을 잡아주는 방법
- 동적으로 변화하는 디자인이나 특정 위치의 좌표값을 지정해주고자 할 때 주로 이용됨
- 자바스크립트 등을 이용하여 동적으로 엘리먼트의 위치가 변화하거나 특정 엘리먼트가 다른 엘리먼트의 간섭없이 화면의 일정 위치에 자리잡아야 할 때 사용
- position:absolute 의 경우 다른 엘리먼트의 간섭이 없기 때문에 화면 레이아웃을 디자인하는 경우 자유롭게 엘리먼트를 배치



화면 레이아웃

- 브라우저 화면의 넓이가 작을 경우에는?
 - 부모 element의 넓이가 충분하지 못하다면 float된 element가 밑으로 떨어지게 됨
 - => float될 엘리먼트들을 감싸주는 엘리먼트가 필요함
 - container 나 wrapper
 - 안쪽에 들어간 엘리먼트들의 넓이의 합보다 전체를 감싸주는 엘리먼트가 큰 값을 가지면 됨

중앙의 2단 구조와 footer 로 이루어진 화면

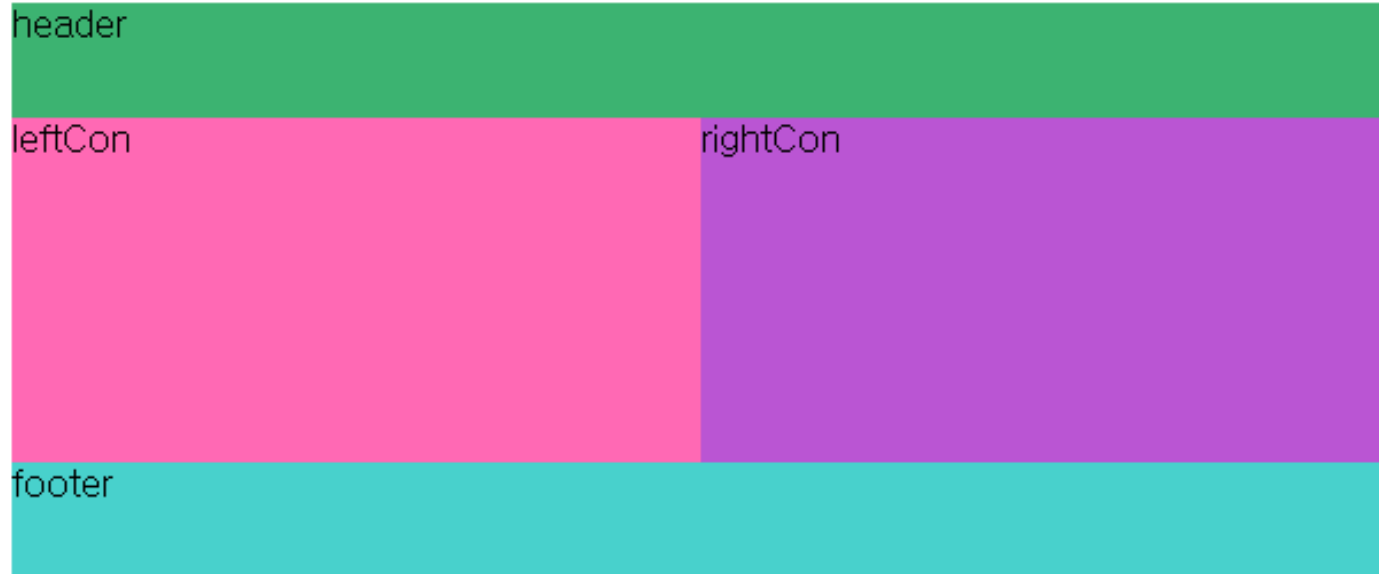


- 세 개의 block element
- #leftCon - 넓이 300px, 높이 150px, float:left
- #rightCon- 넓이 300px, 높이 150px, float:left
- #footer - clear: both

```
#wrap {
  width: 600px;
}
#leftCon {
  background: #ff69b4;
  float: left;
  width: 300px;
  height: 150px;
}
#rightCon {
  background: #ba55d3;
  float: left;
  width: 300px;
  height: 150px;
}
#footer {
  background: #48d1cc;
  /*clear: both;*/
  height: 50px;
}
#container{
  overflow:hidden;
}
```

```
<div id="wrap">
  <div id="container">
    <div id="leftCon">
      leftCon
    </div>
    <div id="rightCon">
      rightCon
    </div>
  </div>
  <div id="footer">
    footer
  </div>
</div>
```


중앙의 2단 구조, header, footer



- 세 개의 block element

#leftCon - 넓이 300px, 높이 150px, float:left

#rightCon- 넓이 300px, 높이 150px, float:left

#footer – clear: both

#wrap – 넓이 600px, 화면 중앙에 나타나게 함 margin:0 auto

```

#wrap {
  width: 600px;
  margin:0 auto;
}
#header {
  background: #3cb371;
  height: 50px;
}
#leftCon {
  background: #ff69b4;
  float: left;
  width: 300px;
  height: 150px;
}
#rightCon {
  background: #ba55d3;
  float: left;
  width: 300px;
  height: 150px;
}
#footer {
  background: #48d1cc;
  clear: both;
  height: 50px;
}
#container{
  overflow:hidden;
}

```

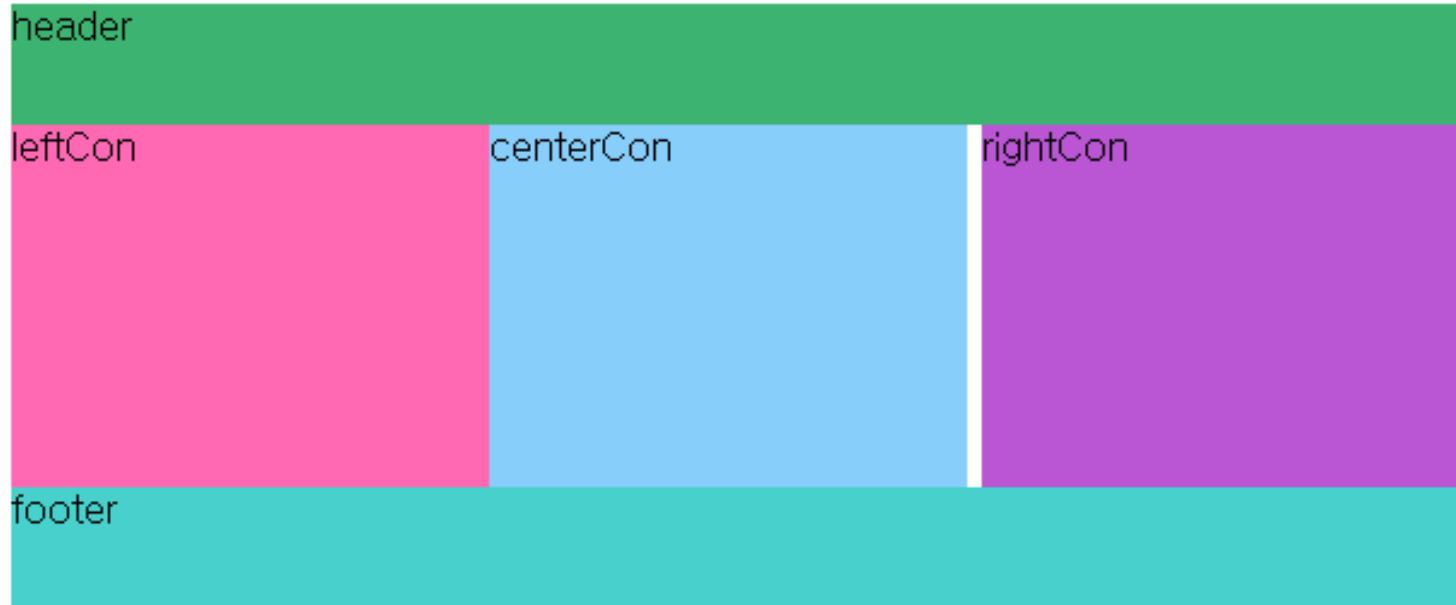
```

<div id="wrap">
  <div id="header">
    header
  </div>
  <div id="container">
    <div id="leftCon">
      leftCon
    </div>
    <div id="rightCon">
      rightCon
    </div>
  </div>
  <div id="footer">
    footer
  </div>
</div>

```

- block element의 가운데 정렬
 - 모든 브라우저에서 적용 가능한 정상적인 가운데 정렬
 - => 해당 block element의 마진값을 조절
- margin:0 auto => 가로 정렬을 가운데로 처리
- margin auto의 경우 해당 엘리먼트의 넓이 값이 없는 경우에는 제대로 동작하지 않을 수 있음

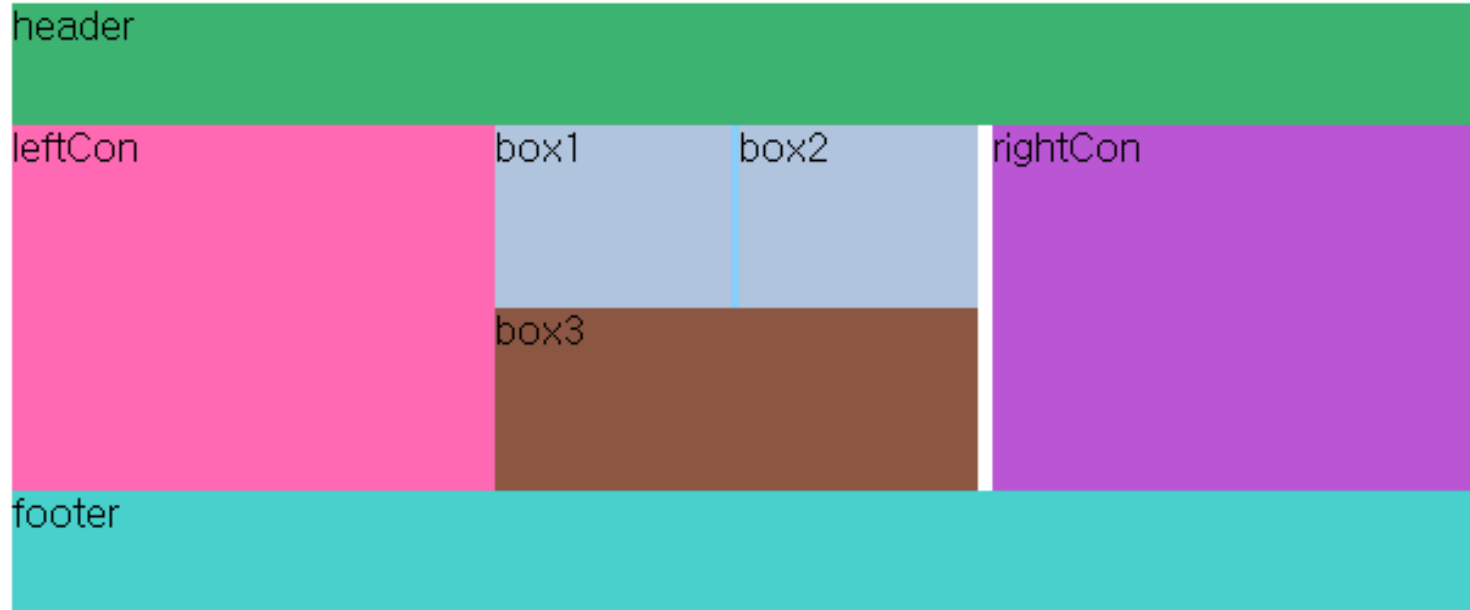
실습-중앙 3단 레이아웃



`#leftCon , #centerCon, #rightCon - width: 33%;`

`#wrap - 넓이 600px, 화면 중앙에 나타나게 함`

실습-복합구조 레이아웃



- 기존의 centerCon에 box 들을 위치시킴
 - #centerCon의 자식 element로 3개의 블록 element 추가
 - #box1 - 높이 50%, 넓이 49%
 - #box2 - 높이 50%, 넓이 49%
 - #box3 - 높이 50%
 - box1 과 box2 사이의 공백 - 마진이 부여되었거나 box2가 오른쪽으로 float
- #wrap을 화면 중앙에 나타나게 함

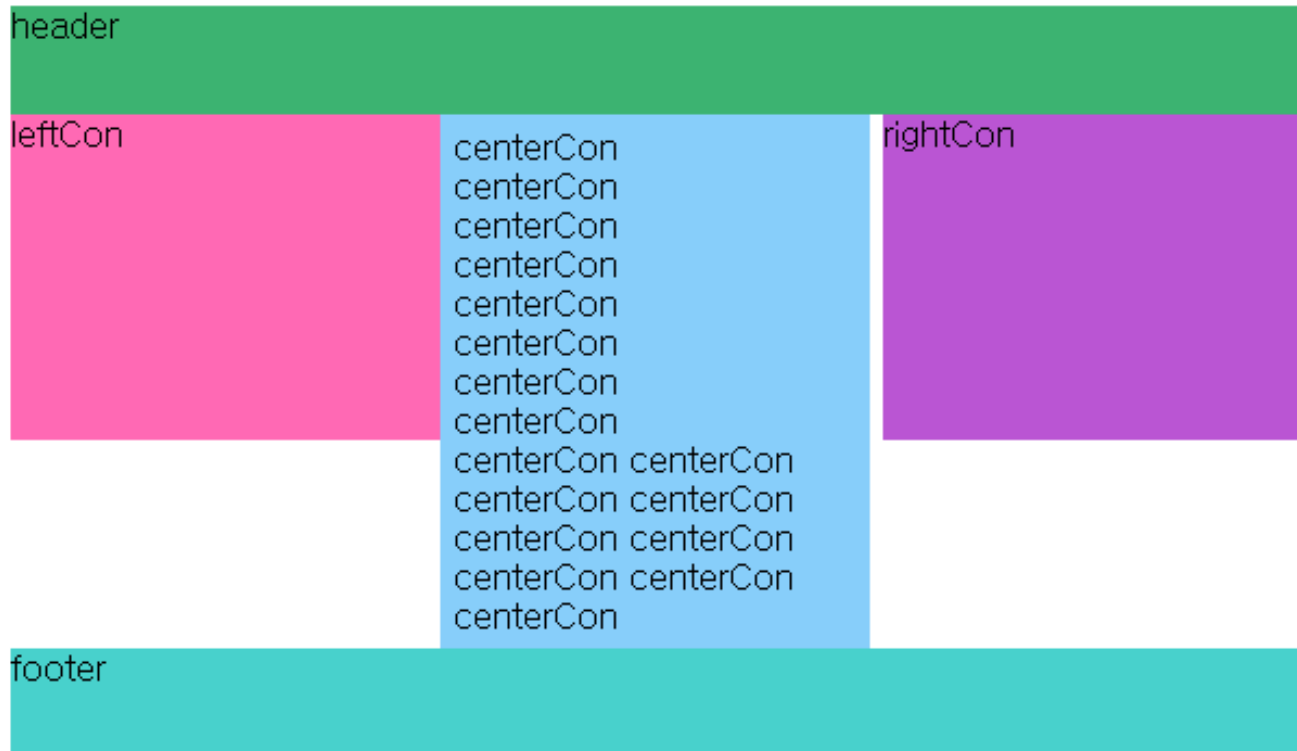
float과 position의 혼합

- float과 position의 혼합

[1] float layout

[2] position layout

=> 조화롭게 어울릴 때 한결 간결한 코드와 화면이 완성됨



float과 position의 혼합

```
#container{ position: relative; }
```

좌표 기준을 잡아주기 위해 container(부모 엘리먼트)은 **relative**를 가짐

```
#leftCon {  
    width: 33%;  
    position: absolute;  
    top: 0;  
    left: 0;  
}  
#centerCon {  
    width: 31%;  
    margin: 0 0 0 33%;  
    padding: 1%;  
}  
#rightCon {  
    width: 33%;  
    position: absolute;  
    top: 0;  
    right: 0;  
}
```

leftCon의 경우 left:0

centerCon의 경우

- leftCon의 넓이 만큼 margin 값 조절

rightCon의 경우 right:0

position을 이용한 레이아웃

-높이값이 정확하게 지정되어 있어야 제대로 된 모양을 가질 수 있음

- **position**을 이용하는 경우 각각의 엘리먼트의 높이, 넓이 값을 정확하게 숙지
- 엘리먼트를 감싸고 있는 부모의 높이값이 있어야 제대로 된 결과물을 얻을 수 있음

```

#wrap {
width: 600px;margin: 0 auto;
}
#header {
background: #3CB371;height: 50px;
}
#container {
position: relative; overflow:auto;
}
#leftCon {
background: #FF69B4;width: 33%;
height: 150px;position: absolute;
top: 0;left: 0;
}
#centerCon {
background: #87CEFA;width: 31%;
margin: 0 0 0 33%; padding: 1%;
}
#rightCon {
background: #BA55D3;
width: 33%;
height: 150px;
position: absolute;
top: 0;
right: 0;
}
#footer {
background: #48D1CC;
height: 50px;
clear: both;
}

```

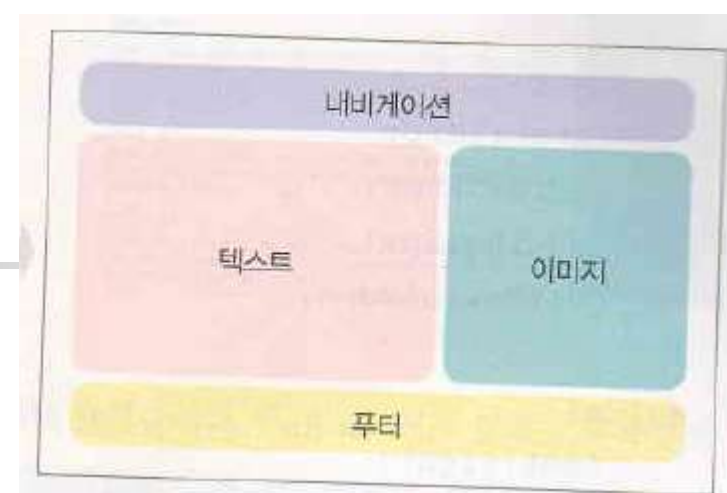
```

<div id="wrap">
  <div id="header">
    header
  </div>
  <div id="container">
    <div id="leftCon">
      leftCon
    </div>
    <div id="centerCon">
      centerCon<br />
      centerCon<br />
      centerCon<br />centerCon<br />
      centerCon<br />centerCon<br />
      centerCon<br />centerCon<br />
      centerCon centerCon centerCon
      centerCon centerCon centerCon
      centerCon centerCon centerCon
    </div>
    <div id="rightCon">
      rightCon
    </div>
  </div>
  <div id="footer">
    footer
  </div>
</div>

```

실습1

- layout1.html 이용하여 일렬로 배치되어 있는 콘텐츠를 타일 형태로 배치하시오.



강아지 용품 준비하기

강아지 집

강아지가 편히 쉴 수 있는 포근한 집이 필요합니다. 강아지의 집은 강아지가 다 큰 후에도 계속 쓸 수 있는 집으로 구입하세요. 집을 구입하실 때는 박음질이 잘 되어 있는지, 세탁이 간편한 저불인지 꼭 확인하시고 고르시는 것이 좋습니다.

강아지 먹이

강아지의 먹이는 꼭 어린 강아지용으로 나와있는 사료를 선택하세요. 강아지들은 사람에 비해 성장속도가 8배정도 빠릅니다. 따라서 강아지에게는 성장속도에 맞는 사료를 급여하여야 합니다. 사람이 먹는 음식을 먹게 되면 양념과 향신료에 입맛이 익숙해지고, 비만이 될 가능성이 매우 높아집니다. 강아지를 사르는 생후 12개월까지 급여하여야 합니다.

밥그릇, 물병

밥그릇은 쉽게 넘어지지 않도록 바닥이 넓은 것이 좋습니다. 물병은 대롱이 달린 것으로 선택하세요. 밥그릇에 물을 주게 되면 입 주변에 털이 모두 젖기 때문에 비위생적하므로 대롱을 통해서 물을 먹을 수 있는 물병을 마련하시는 것이 좋습니다.

이름표, 목줄

강아지를 잃어버릴 염려가 있으니 산책할 무렵이 되면 이름표를 목 목에 걸어주도록 하세요. 그리고 방울이 달린 목걸이를 하고자 하실 때는 신중하셔야 합니다. 울직일 때마다 방울이 딸랑 거리면 신경이 예민한 강아지들에게는 좋지 않은 영향을 끼칠 수 있기 때문입니다.



강아지 용품 준비하기

강아지 집

강아지가 편히 쉴 수 있는 포근한 집이 필요합니다. 강아지의 집은 강아지가 다 큰 후에도 계속 쓸 수 있는 집으로 구입하세요. 집을 구입하실 때는 박음질이 잘 되어 있는지, 세탁이 간편한 저불인지 꼭 확인하시고 고르시는 것이 좋습니다.

밥그릇, 물병

밥그릇은 쉽게 넘어지지 않도록 바닥이 넓은 것이 좋습니다. 물병은 대롱이 달린 것으로 선택하세요. 밥그릇에 물을 주게 되면 입 주변에 털이 모두 젖기 때문에 비위생적하므로 대롱을 통해서 물을 먹을 수 있는 물병을 마련하시는 것이 좋습니다.

강아지 먹이

강아지의 먹이는 꼭 어린 강아지용으로 나와있는 사료를 선택하세요. 강아지들은 사람에 비해 성장속도가 8배정도 빠릅니다. 따라서 강아지에게는 성장속도에 맞는 사료를 급여하여야 합니다. 사람이 먹는 음식을 먹게 되면 양념과 향신료에 입맛이 익숙해지고, 비만이 될 가능성이 매우 높아집니다. 강아지를 사르는 생후 12개월까지 급여하여야 합니다.

이름표, 목줄

강아지를 잃어버릴 염려가 있으니 산책할 무렵이 되면 이름표를 목 목에 걸어주도록 하세요. 그리고 방울이 달린 목걸이를 하고자 하실 때는 신중하셔야 합니다. 울직일 때마다 방울이 딸랑 거리면 신경이 예민한 강아지들에게는 좋지 않은 영향을 끼칠 수 있기 때문입니다.



9. 그림자 속성

- 그림자 속성 - 태그에 그림자를 부여해주는 스타일 속성
- [1] text-shadow 속성
 - 글자에 그림자를 부여하는 스타일 속성

text-shadow: 5px 5px 5px black

오른쪽 아래 흐림도 색상

그림자가 번지는 정도

```
<head>
  <title>CSS3 Property Basic</title>
  <style>
    h1 {
      text-shadow: 5px 5px 5px black;
    }
  </style>
</head>
<body>
  <h1>Lorem ipsum dolor amet</h1>
</body>
```

Lorem ipsum dolor amet

```
<style>
body {
  font-family:Arial;
  background:#808080;
}
h1 {
  font-size:5em;
  color:white;
}
.shadow1 {
  text-shadow:1px 1px 3px #000;
}
.shadow2 {
  color:black;
  text-shadow:3px -3px 5px #fff;
}
</style>
</head>

<body>
<h1>Joandora</h1>
<h1 class="shadow1">Joandora</h1>
<h1 class="shadow2">Joandora</h1>
</body>
```

Joandora

Joandora

Joandora

9. 그림자 속성

- [2] box-shadow 속성
 - 박스에 그림자를 부여하는 속성

box-shadow: 5px 5px 5px black

오른쪽 아래 흐림도 색상

- 흐림도와 색상 사이에 inset 키워드를 넣으면 그림자가 태그 안쪽에 생성됨

```
<head>
  <title>CSS3 Property Basic</title>
  <style>
    div {
      border: 3px solid black;
      box-shadow: 10px 10px 30px black;
      text-shadow: 5px 5px 5px black;
    }
  </style>
</head>
<body>
  <div>
    <h1>Lorem ipsum dolor amet</h1>
  </div>
</body>
```

Lorem ipsum dolor amet

중첩 그림자

- 그림자 속성은 쉼표를 사용해 여러 개의 그림자를 적용할 수 있음
 - 예제) 각각의 글자와 박스에 그림자를 3개씩 적용

```
<head>
<title>CSS3 Property Basic</title>
<style>
  .box {
    border: 3px solid black;
    box-shadow: 10px 10px 10px black, 10px 10px 20px orange, 10px 10px 30px red;
    text-shadow: 10px 10px 10px black, 10px 10px 20px orange, 10px 10px 30px red;
  }
</style>
</head>
<body>
  <div class="box">
    <h1>Lorem ipsum dolor amet</h1>
  </div>
</body>
```

3중 그림자가 생성됨

Lorem ipsum dolor amet