



CHAPTER 6.

CSS 레이아웃과 애니메이션





레이아웃이란?

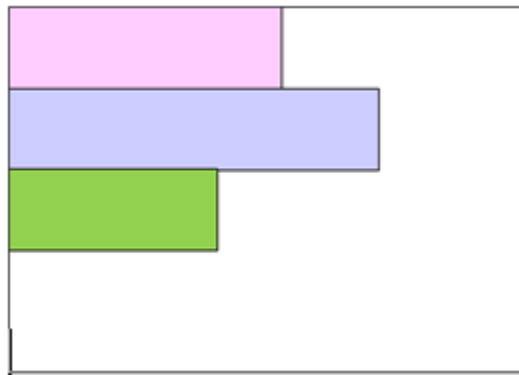
- 웹페이지에서 HTML 요소의 위치, 크기 등을 결정하는 것
 - 집안에서의 가구 배치와 비슷하다.



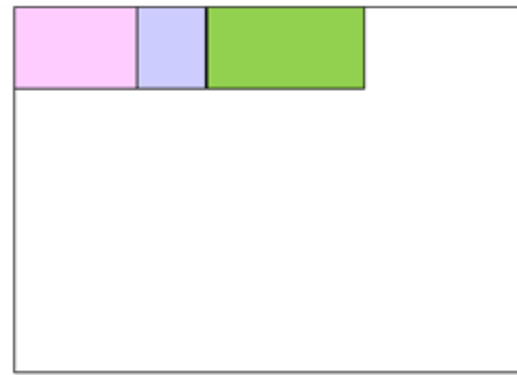


블록요소와 인라인 요소

- 블록(block) 요소 - 화면의 한 줄을 전부 차지한다.
- 인라인(inline) 요소 - 한 줄에 차례대로 배치된다. 현재 줄에서 필요한 만큼의 너비만을 차지한다.



블록(block)



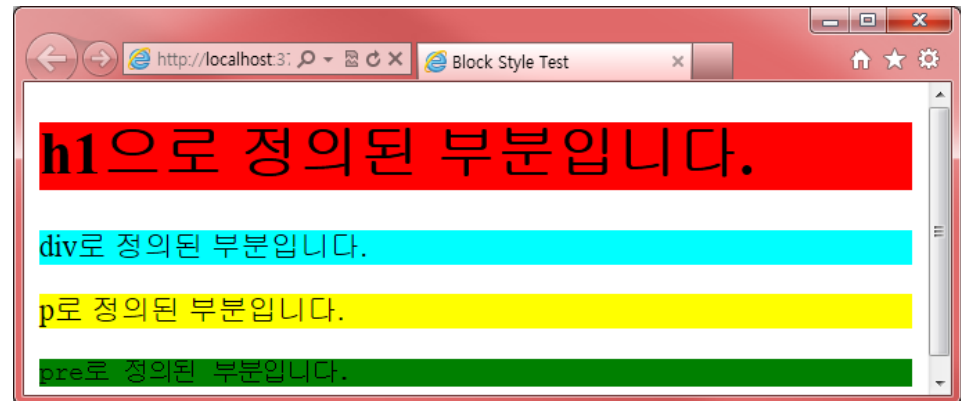
인라인(inline)



블록요소

- 한 줄을 전부 차지
- <h1>, <p>, , , <table>, <blockquote>, <pre>, <div> <form>, <header>, <nav> 요소
- 예제 실행과 소스보기

```
<body>  
  <h1 style="background-color: red">h1으로 정의된 부분입니다.</h1>  
  <div style="background-color: aqua">div로 정의된 부분입니다.</div>  
  <p style="background-color: yellow">p로 정의된 부분입니다.</p>  
  <pre style="background-color: green">pre로 정의된 부분입니다.</pre>  
</body>
```





인라인요소

- 인라인 요소들은 한 줄 안에 차례대로 배치
- `<a>`, ``, ``, ``, `
`, `<input>`, `` 요소

```
<body>  
  <em style="background-color: red">em 요소</em>  
  <span style="background-color: aqua">span 요소</span>  
    
  <a href="http://www.w3c.org">a 요소</a>  
</body>
```

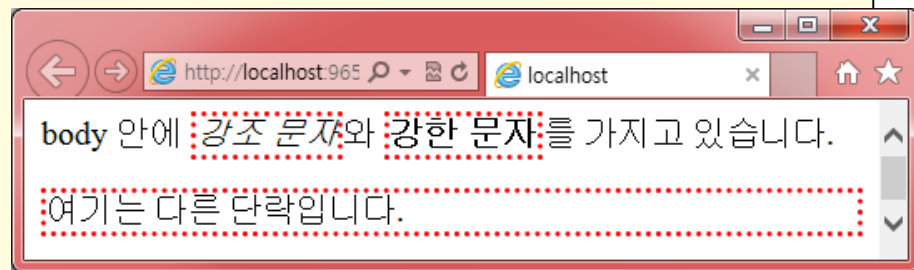




블록 요소와 인라인 요소의 혼합

```
<!DOCTYPE html>

<html>
<head>
  <style>
    p, em, strong {
      border: dotted 3px red;
    }
  </style>
</head>
<body>
  body 안에
  <em>강조 문자</em>와 <strong>강한 문자</strong>를 가지고 있습니다.
  <p>여기는 다른 단락입니다. </p>
</body>
</html>
```





CSS의 display 속성

- 속성 display를 block으로 설정하면 -> 블록 요소처럼 배치
- display를 inline으로 설정-> 인라인 요소처럼 배치
 - display:block : 블록(block)
 - display:inline : 인라인(inline)
 - display:none : 없는 것으로 간주됨, 공간을 차지하지 않음
 - visibility:hidden : 화면에서 감춰짐, 공간을 차지함

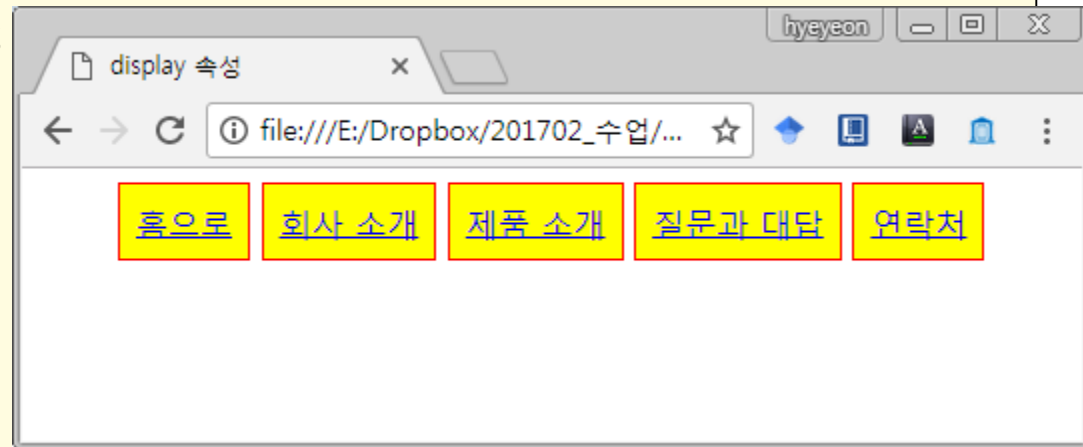


예제

```
<!DOCTYPE html>
<html>
<head>
  <title>display 속성</title>
  <style>

    .menubar li {
      display: inline;
      background-color: yellow;
      border: 1px solid;
      border-color: red;
      margin: 0;
      padding: .5em;
    }
  </style>
</head>

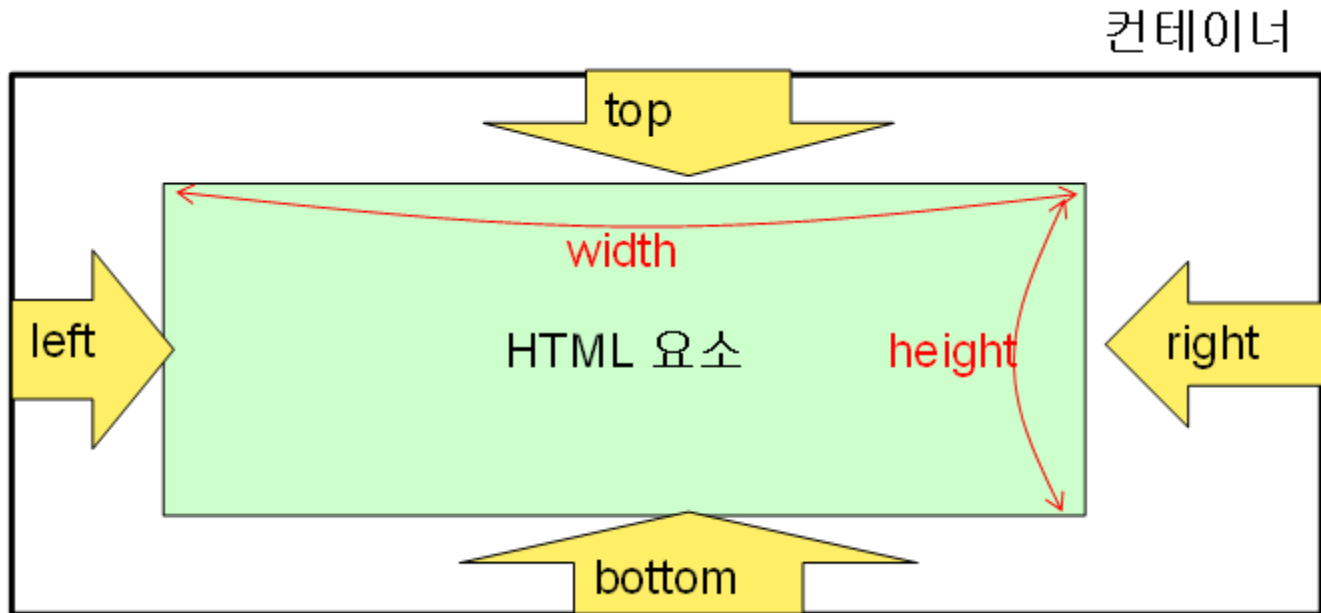
<body>
  <ul class="menubar">
    <li><a href="#">홈으로</a></li>
    <li><a href="#">회사 소개</a></li>
    <li><a href="#">제품 소개</a></li>
    <li><a href="#">질문과 대답</a></li>
    <li><a href="#">연락처</a></li>
  </ul>
</body>
</html>
```





요소의 위치

- top, bottom, left, right 속성으로 결정
- 해당하는 속성의 위치에 여백





위치 설정 방법

- 정적 위치 설정(static) - 정상적인 흐름에 따른 배치, 기본 속성
- 상대 위치 설정(relative) – 정상적인 위치가 기준점이 된다.
기본 속성 static 일 때의 위치를 기준으로 위치에 대한 효과를 적용
- 절대 위치 설정(absolute positioning) – 컨테이너의 원점이 기준점
정적 위치를 무시하고 컨테이너 위치를 기준으로 적용
- 고정 위치 설정(fixed positioning) - 윈도우의 원점이 기준점이 된다.
정적 위치를 무시하고 브라우저 페이지 위치를 기준으로 적용
스크롤이 이동되어도 위치를 유지한다.



정적 위치 설정

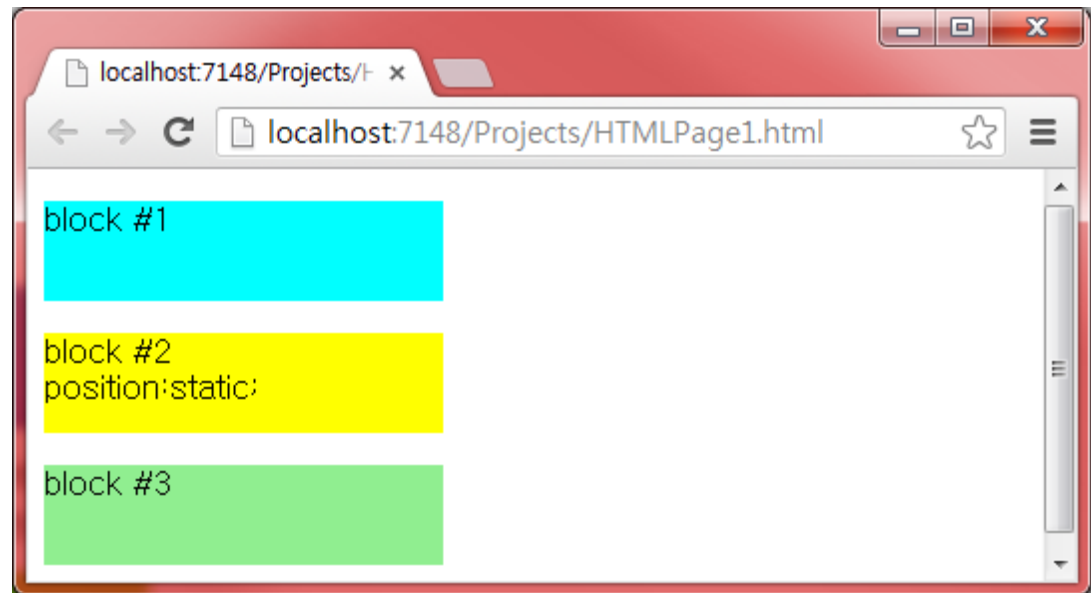
- 정적 위치 설정(static positioning)
 - 블록 요소들은 박스처럼 상하로 쌓이게 되고 인라인 요소들은 한 줄에 차례대로 배치

```
<!DOCTYPE html>
<html>
<head>
  <style>
    #one {
      background-color: cyan;
      width: 200px;
      height: 50px;
    }
    #two {
      position: static;
      background-color: yellow;
      width: 200px;
      height: 50px;
    }
    #three {
      background-color: lightgreen;
      width: 200px;
      height: 50px;
    }
  </style>
```



예제

```
<body>
  <p id="one">block #1</p>
  <div id="two">
    block #2<br />
    position:static;<br />
  </div>
  <p id="three">block #3</p>
</body>
</html>
```





상대 위치 설정

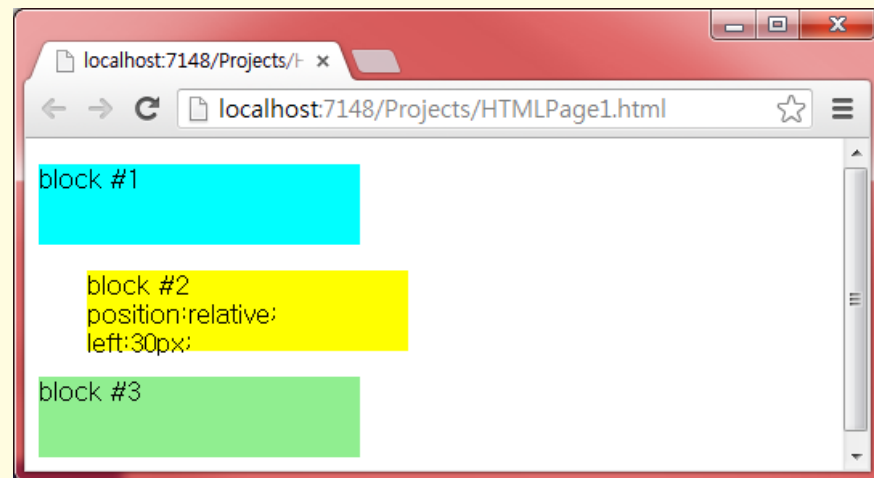
- 상대 위치 설정(relative positioning)
 - 정상적인 위치에서 상대적으로 요소가 배치

```
<style>
  #one {
    background-color: cyan;
    width: 200px;
    height: 50px;
  }

  #two {

    position: relative;
    left: 30px;
    background-color: yellow;
    width: 200px;
    height: 50px;
  }

  #three {
    background-color: lightgreen;
    width: 200px;
    height: 50px;
  }
</style>
```

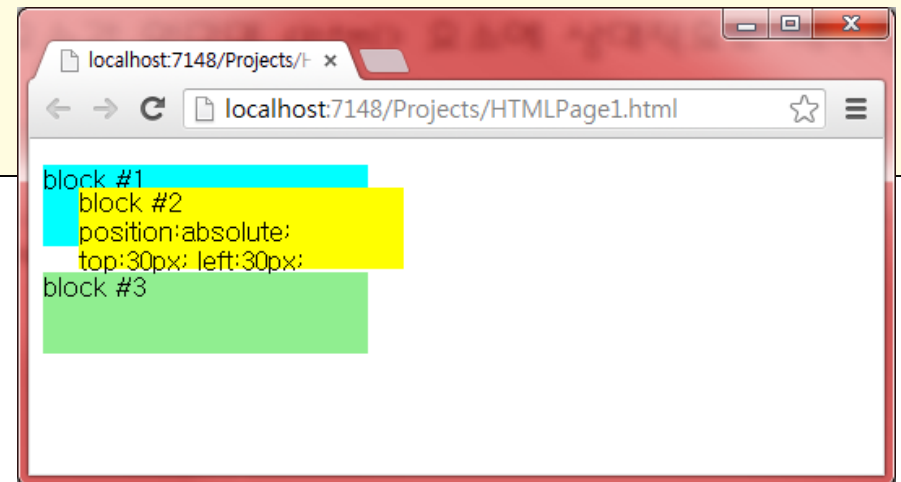




절대 위치 설정

- 절대 위치(absolute positioning)
 - 전체 페이지를 기준으로 시작 위치에서 top, left, bottom, right 만큼 떨어진 위치에 배치

```
...  
#two {  
  position: absolute;  
  
  top: 30px;  
  left: 30px;  
  
  background-color: yellow;  
  width: 200px;  
  height: 50px;  
}  
...
```

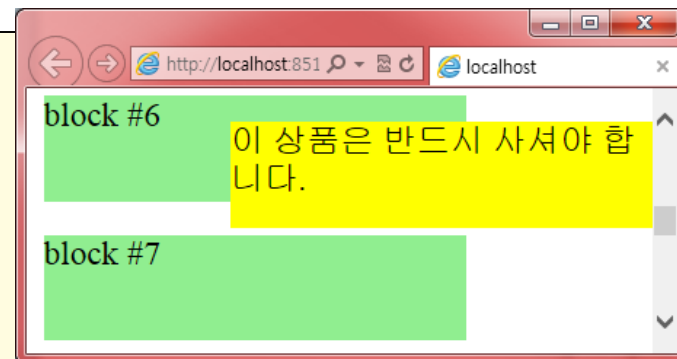




고정 위치 설정

- 고정 위치 설정(fixed positioning)
 - 브라우저 윈도우에 상대적으로 요소의 위치를 잡는 것

```
<!DOCTYPE html>
<html>
<head>
  <style>
    p {
      background-color: lightgreen;
      width: 200px;
      height: 50px;
    }
    #two {
      background-color: yellow;
      position: fixed;
      top: 0px;
      right: 0px;
    }
  </style>
</head>
```

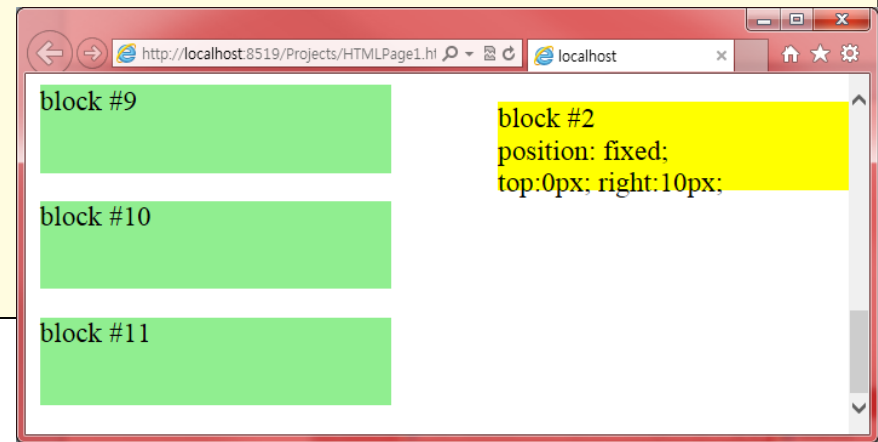
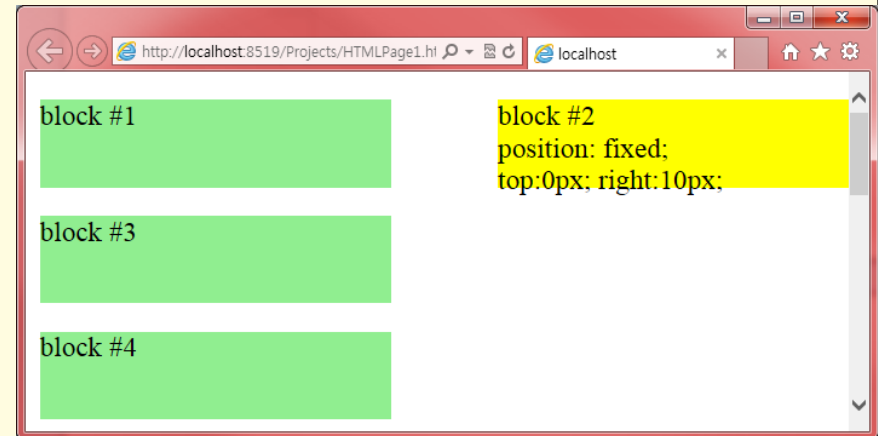




고정 위치 설정

```
<body>
  <p>block #1</p>
  <p id="two">
    block #2<br />
    position: fixed;<br />
    top:0px; right:10px;
  </p>

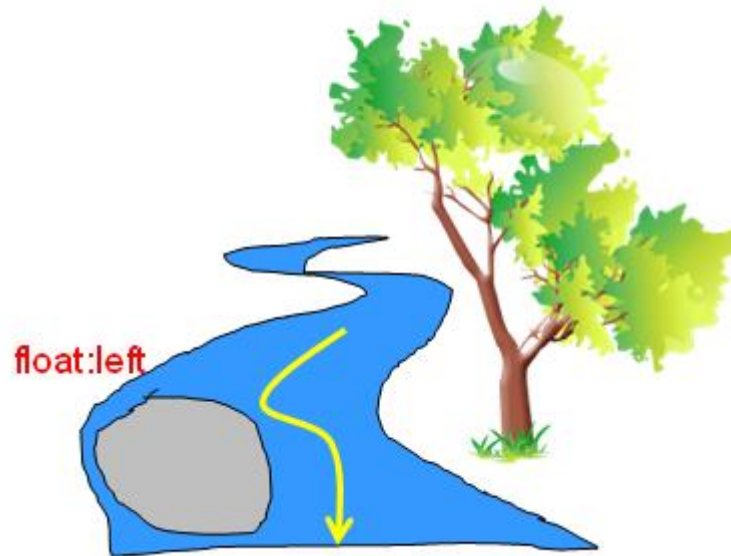
  <p>block #3</p>
  <p>block #4</p>
  <p>block #5</p>
  <p>block #6</p>
  <p>block #7</p>
  <p>block #8</p>
  <p>block #9</p>
  <p>block #10</p>
  <p>block #11</p>
</body>
</html>
```





float 속성

- 하나의 콘텐츠 주위로 다른 콘텐츠들이 물처럼 흘러가는 스타일 지정





예제

```
<!DOCTYPE html>
<html>
<head>
  <style>
    img.a {      float: left      }
  </style>
</head>

<body>
  
  <p>
    생활이 그대를 속일지라도
    슬퍼하거나 노여워 말라.
    ...
  </p>
</body>
</html>
```

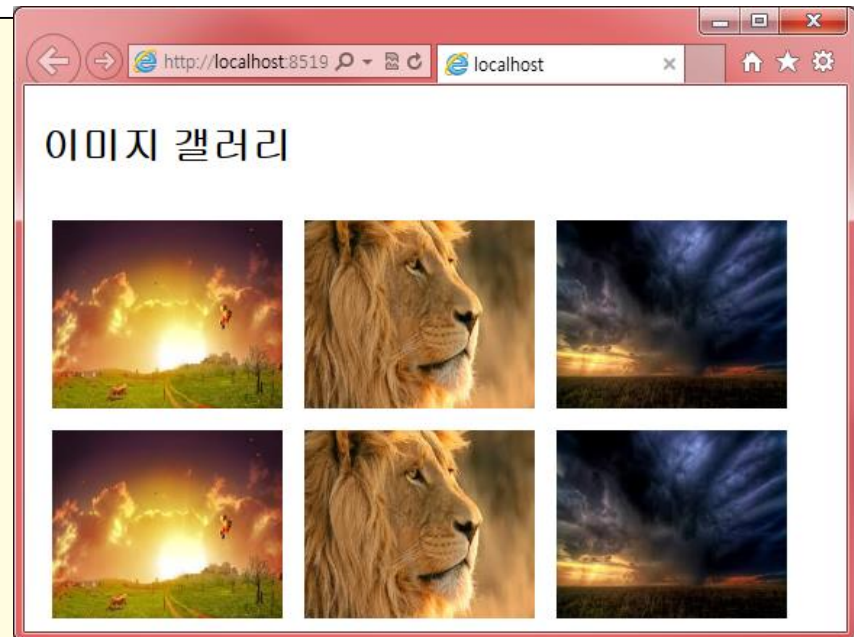




예제

```
<!DOCTYPE html>
<html>
<head>
  <style>
    img {
      float: left;
      width: 110px;
      height: 90px;
      margin: 5px;
    }
  </style>
</head>

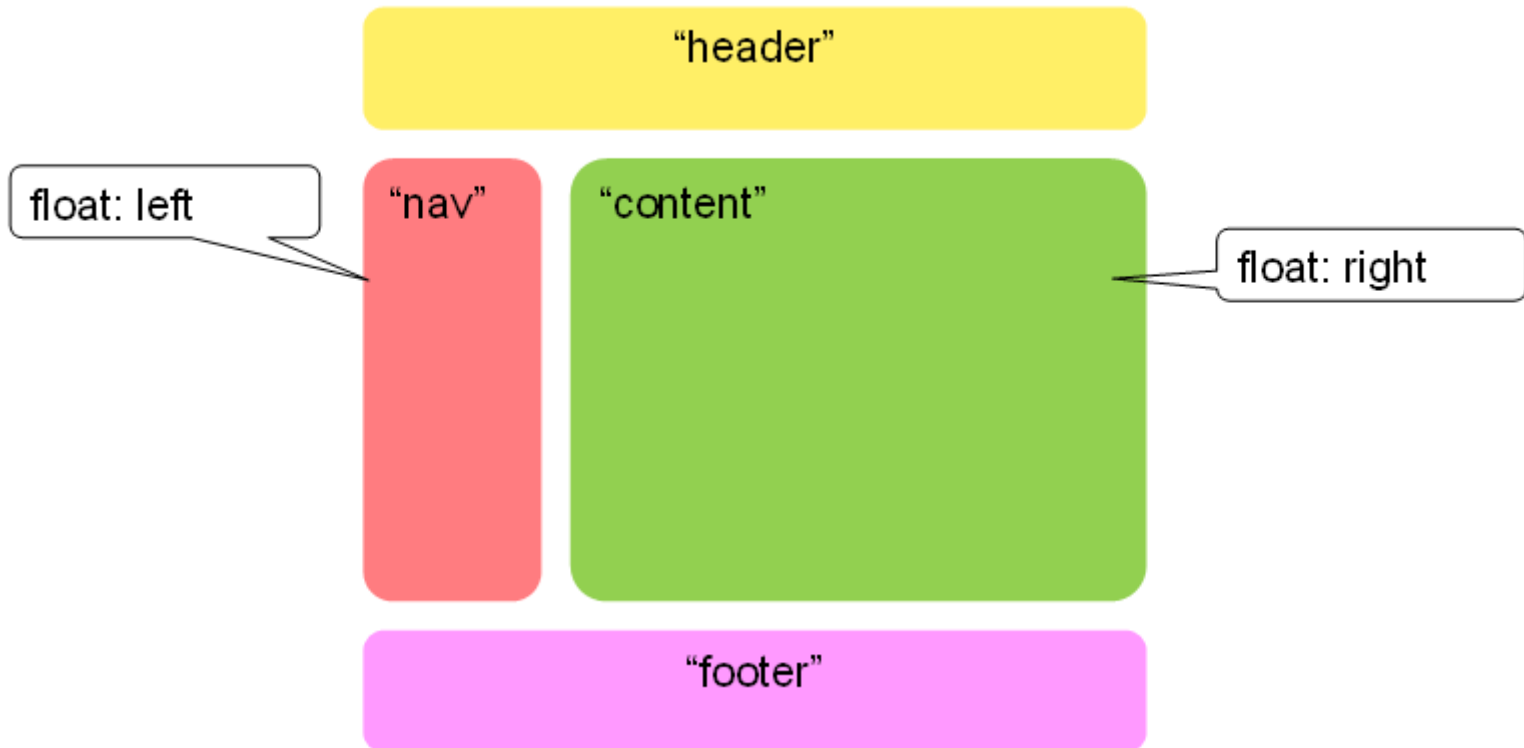
<body>
  <h3>이미지 갤러리</h3>
  
  
  
  
  
  
</body>
</html>
```





float의 용도

- 레이아웃에 많이 사용된다.





clear 속성

- float 속성을 중단할 때 사용된다.

```
#footer {  
  clear: both;  
}
```

float: left

"content"

"nav"

float: right

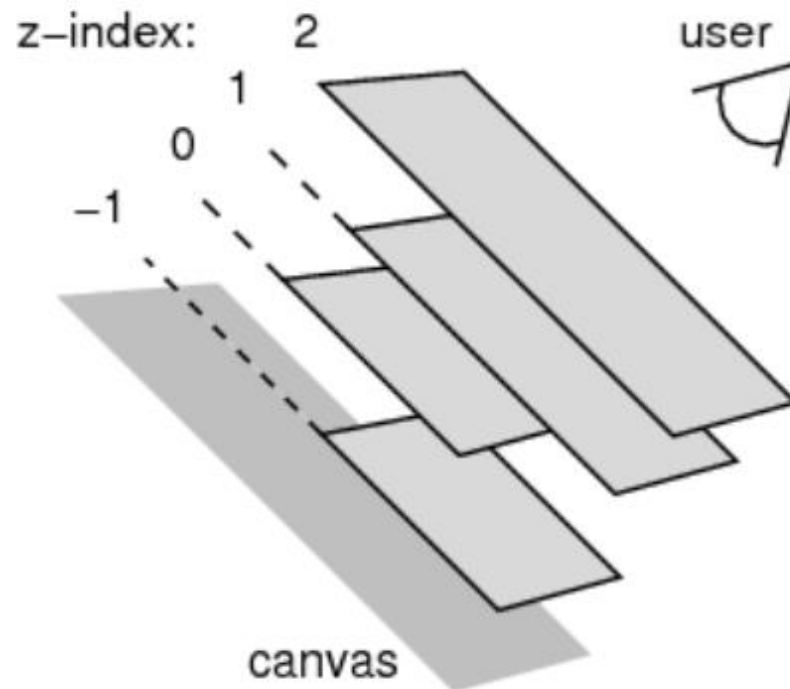
"footer"

clear: both로
지정하면 빈칸을
채우지 않고
정상적으로
배치된다.



z-index

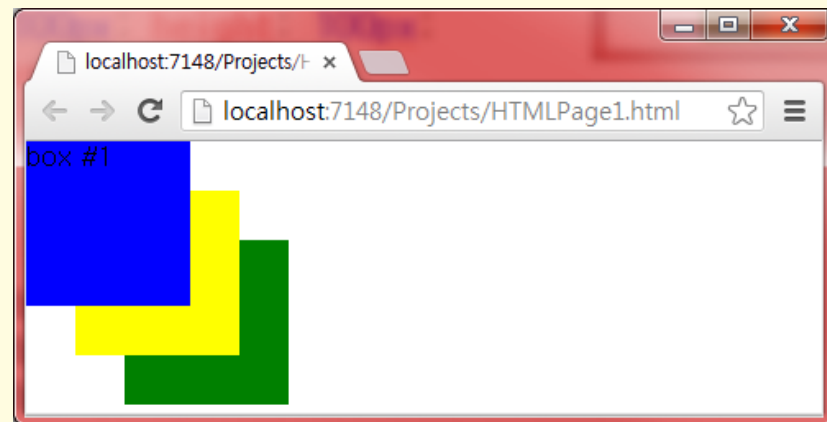
- 요소의 스택 순서를 지정
 - z-index 요소의 position 속성이 absolute 또는 fixed로 설정
 - z-index의 값이 클수록 앞에 놓이게 된다. (기본값 0)





예제

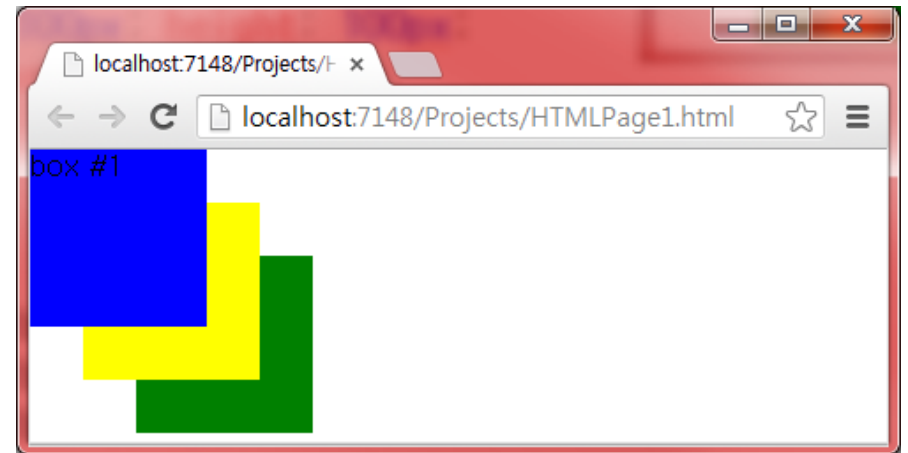
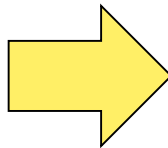
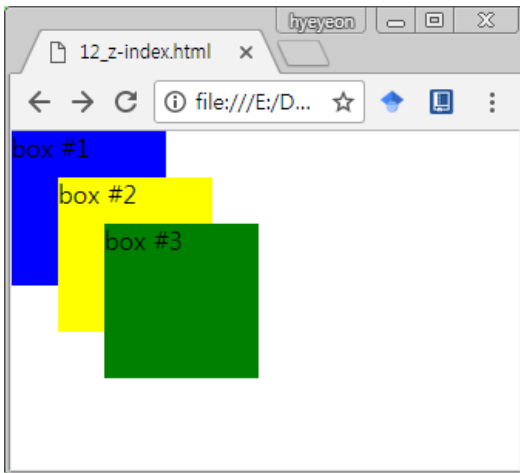
```
...  
<style>  
  #box1 {  
    position: absolute;  
    top: 0px; left: 0px;  
    width: 100px; height: 100px;  
    background: blue;  
    z-index: 200;  
  }  
  
  #box2 {  
    position: absolute;  
    top: 30px; left: 30px;  
    width: 100px; height: 100px;  
    background: yellow;  
    z-index: 100;  
  }  
  
  #box3 {  
    position: absolute;  
    top: 60px; left: 60px;  
    width: 100px; height: 100px;  
    background: green;  
    z-index: 0;  
  }  
</style>
```





예제

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

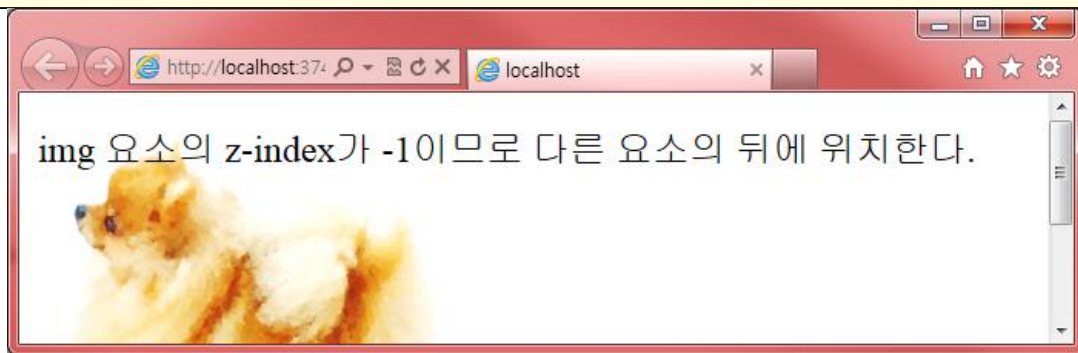




예제

```
<!DOCTYPE html>
<html>
<head>
  <style>
    img {
      position: absolute;
      left: 0px;
      top: 0px;
      z-index: -1;
    }
  </style>
</head>

<body>
  
  <p>img 요소의 z-index가 -1이므로 다른 요소의 뒤에 위치한다. </p>
</body>
</html>
```





요소의 크기 지정

- width, height – 요소의 크기
- min-width, min-height: 요소의 최소 크기
- max-width, max-height: 요소의 최대 크기



예제

```
<!DOCTYPE html>

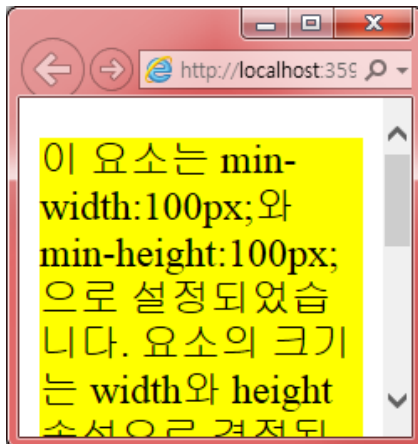
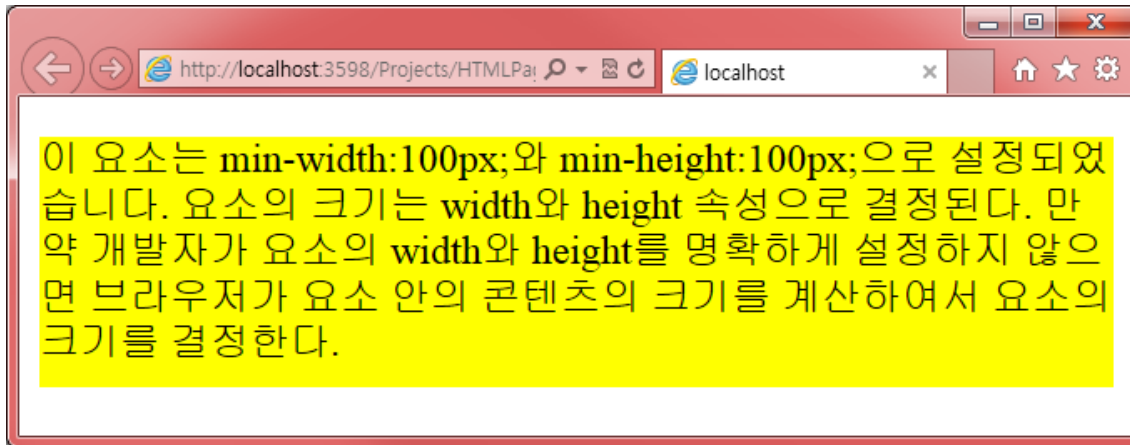
<html>
<head>
  <style>
    p {
      min-width: 100px;
      min-height: 100px;
      background-color: yellow;
    }
  </style>
</head>

<body>
  <p>
    이 요소는 min-width:100px;와 min-height:100px;으로 설정되었습니다.
    요소의 크기는 width와 height 속성으로 결정된다.
    만약 개발자가 요소의 width와 height를 명확하게 설정하지 않으면 브라우저가
    요소 안의 콘텐츠의 크기를 계산하여서 요소의 크기를 결정한다.
  </p>
</body>
</html>
```



실행 결과

- 실행 결과(클릭)





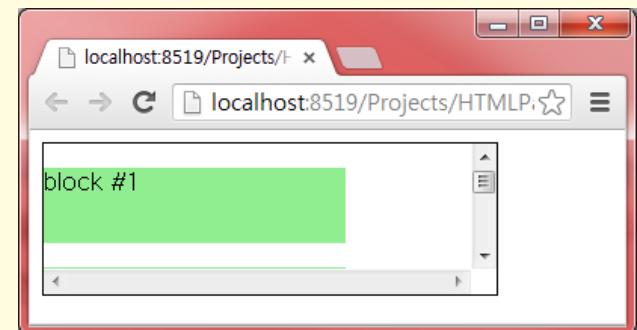
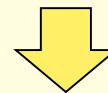
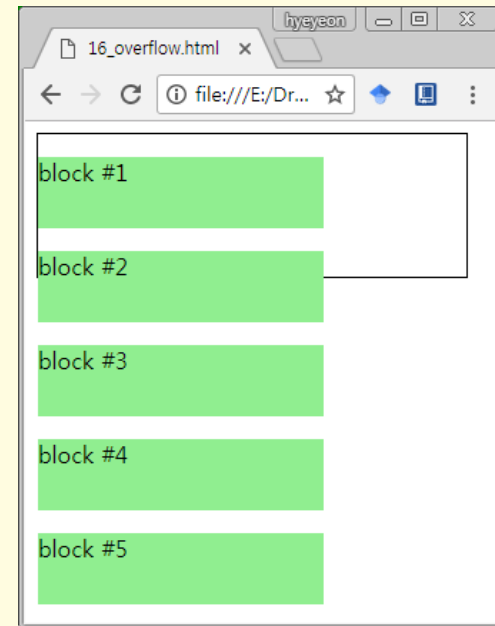
overflow 속성

- overflow 속성: 자식 요소가 부모 요소의 범위를 벗어났을 때, 어떻게 처리할 것인지를 지정
 - hidden – 부모 영역을 벗어나는 부분을 보이지 않게 한다.
 - scroll – 부모 영역을 벗어나는 부분을 스크롤 할 수 있도록 한다.
 - auto – 자동으로 스크롤바가 나타난다.



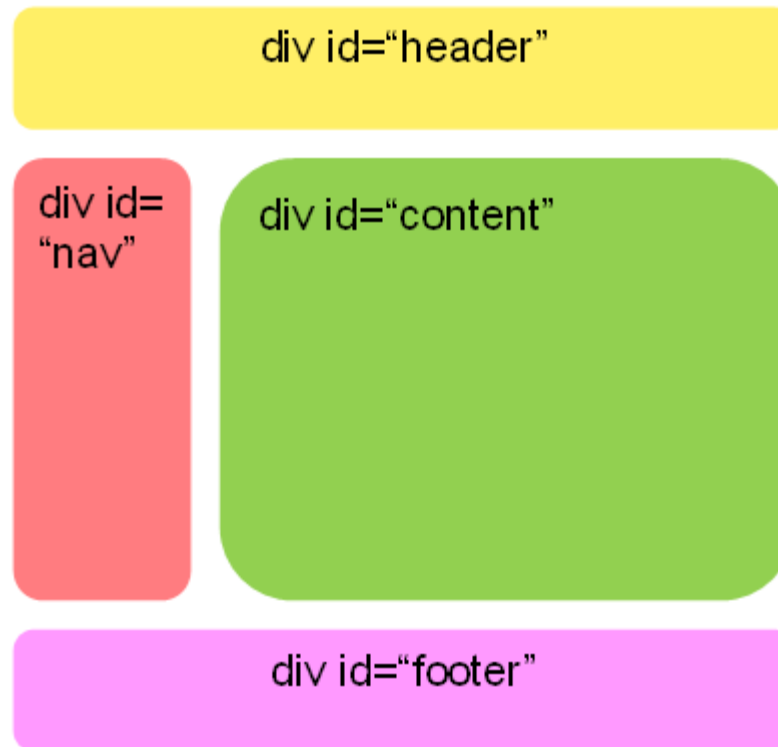
예제

```
<!DOCTYPE html>
<html>
<head>
  <style>
    p {
      background-color: lightgreen;
      width: 200px;
      height: 50px;
    }
    #target {
      border: 1px solid black;
      width: 300px;
      height: 100px;
      overflow: scroll;
    }
  </style>
</head>
<body>
  <div id=target>
    <p>block #1</p>
    <p>block #2</p>
    <p>block #3</p>
    <p>block #4</p>
    <p>block #5</p>
  </div>
</body>
</html>
```





<div>를 이용한 레이아웃





예제

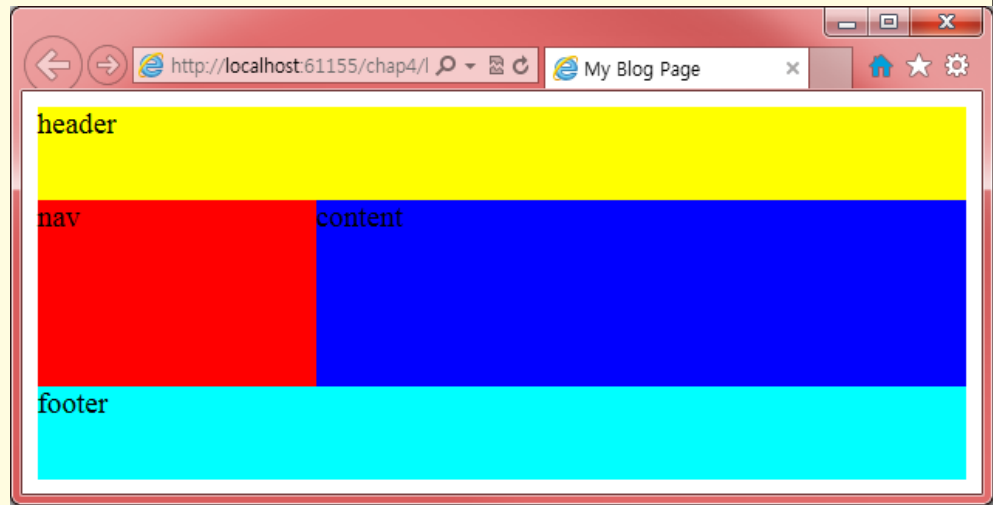
```
<!DOCTYPE html>
<html>
<head>
  <title>My Blog Page</title>
  <style>
#header {
  background-color: yellow;
  width: 100%;
  height: 50px;
}

#nav {
  width: 30%;
  background-color: red;
  height: 100px;
  float: left;
}
```



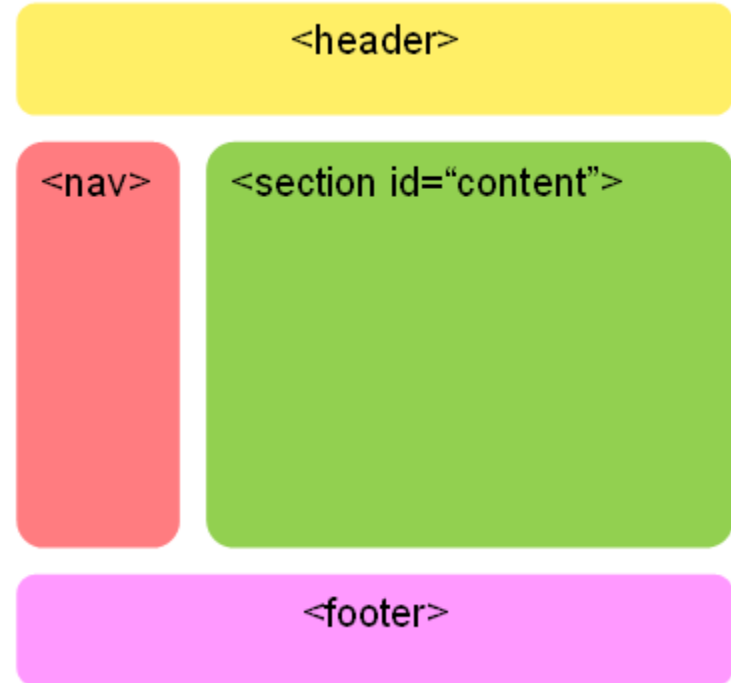
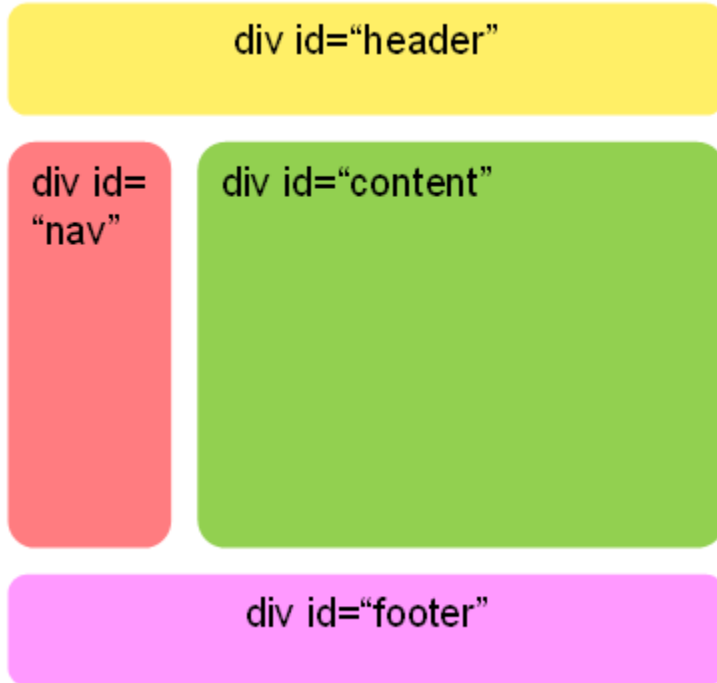

예제

```
#content {  
  width: 70%;  
  background-color: blue;  
  float: right;  
  height: 100px;  
}  
#footer {  
  background-color: aqua;  
  width: 100%;  
  height: 50px;  
  clear: both;  
}  
</style>  
</head>  
<body>  
  <div id="wrapper">  
    <div id="header"> header </div>  
    <div id="nav"> nav </div>  
    <div id="content"> content </div>  
    <div id="footer"> footer </div>  
  </div>  
</body>  
</html>
```





시맨틱 요소 레이아웃





시맨틱 요소

태그	설명
<i><header></i>	문서의 머리말(header)
<i><hgroup></i>	<h1>에서 <h6>요소들의 그룹
<i><nav></i>	내비게이션 링크
<i><article></i>	문서의 내용이나 블로그의 포스트
<i><section></i>	문서의 섹션을 의미한다.
<i><aside></i>	사이드바와 같이 옆에 위치하는 내용
<i><footer></i>	문서의 꼬리말/footer)
<i><figure></i>	그림이나 도표
<i><time></i>	날짜와 시간을 표시



예제





예제

```
h1 {  
  margin: 0px;  
}  
  
section#main {  
  display: table-cell;  
  background-color: yellow;  
  padding: 15px;  
}  
  
nav {  
  display: table-cell;  
  background-color: #ffd800;  
  padding: 15px;  
}  
  
footer {  
  background-color: #954b4b;  
  color: #efe5d0;  
  text-align: center;  
  padding: 10px;  
  margin: 0px 0px 0px 0px;  
  font-size: 90%;  
}  
</style>
```



예제

```
<body>
  <header>
    <h1>My Blog Page</h1>
  </header>
  <nav>
    <h1>Links</h1>
    <ul>
      <li><a href="http://www.w3c.org/">W3C</a></li>
      <li><a href="http://developer.mozilla.org/">MOZILLA</a></li>
      <li><a href="http://htmldog.com/guides/">HTML Dogs</a></li>
    </ul>
    <figure>
      
      <figcaption>홍길동</figcaption>
    </figure>
  </nav>
```



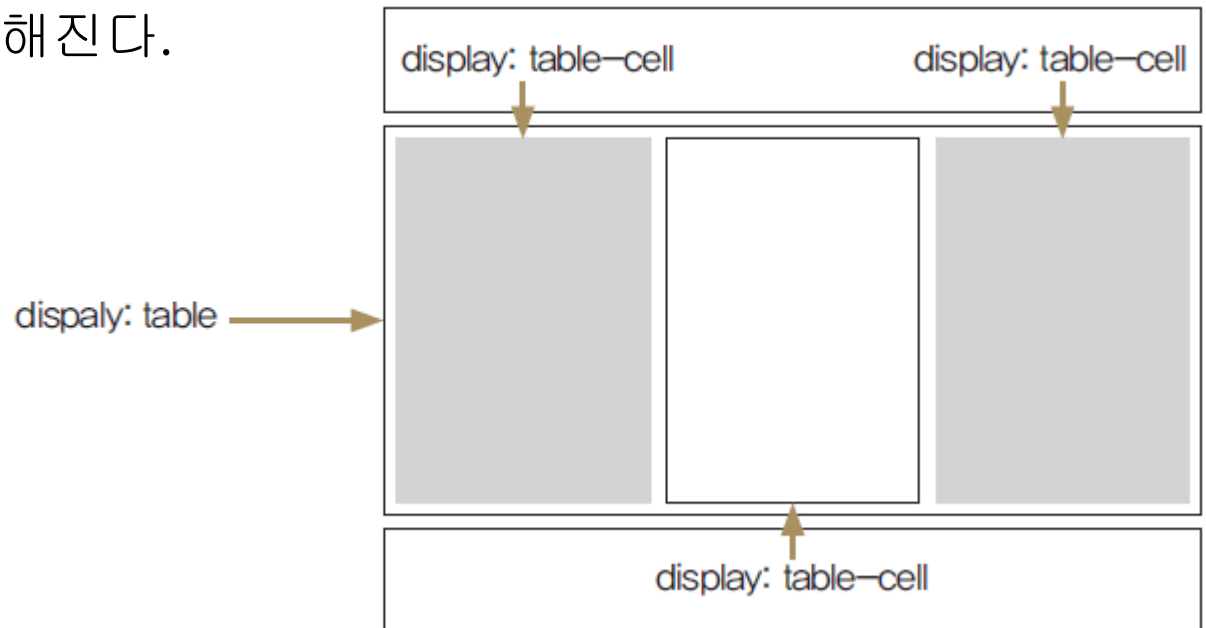
예제

```
<section id="main">
  <article>
    <h1>Semantic Tags</h1>
    <p>
      시멘틱 요소(Semantic elements)들은 브라우저에게 요소의 의미나 목적을
      명확하게 알려주는 요소이다.
    </p>
  </article>
  <article>
    <h1>div와 span</h1>
    <p>
      div은 “divide“의 약자로서 페이지를 논리적인 섹션으로 분리하는데 사용되는
      태그이다.
      span 요소는 인라인 요소로서 텍스트를 위한 컨테이너로 사용할 수 있다.
    </p>
  </article>
</section>
<footer>Copyright (c) 2013 Hong</footer>
</body>
</html>
```




table-cell 속성

- display 속성에 table-cell을 하면 자식 요소들을 테이블의 셀처럼 배치하라는 의미가 된다.
 - float나 절대위치 설정이 필요없다
 - 테이블 형태로 레이아웃 된다.
 - 컬럼의 높이가 달라도 문제되지 않는다.
 - CSS가 간결해진다.





CSS3 효과: 투명도

```
<!DOCTYPE html>

<html>
<head>
  <style>
    img { opacity: 0.4; }
    img:hover { opacity: 1.0; }
  </style>
</head>
<body>

  <h1>Opacity 속성</h1>
  
  
</body>
</html>
```





CSS3 효과: 가시성

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
  <style>
```

```
    #a {
```

```
      visibility: hidden;
```

```
      border: 1px dotted red;
```

```
    }
```

```
    #b {
```

```
      visibility: visible;
```

```
      border: 1px dotted red;
```

```
    }
```

```
  </style>
```

```
</head>
```

```
<body>
```

```
  <h1>Visibility 속성</h1>
```

```
  
```

```
  
```

```
</body>
```

```
</html>
```

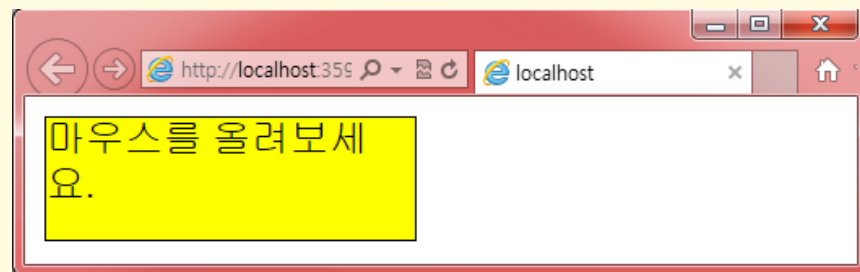




CSS3: 전환

```
<!DOCTYPE html>
<html>
<head>
  <style>
    div {
      width: 100px;
      height: 50px;
      border: 1px solid black;
      background: yellow;
      transition: width 5s;
    }

    div:hover {
      width: 200px;
    }
  </style>
</head>
<body>
  <div>마우스를 올려보세요.</div>
</body>
</html>
```



실행결과 보기



CSS3: 전환

```
<!DOCTYPE html>
<html>

<head>
  <style>
    p {
      width: 100px;
      height: 50px;
      border: 1px solid black;
      background: yellow;
      transition: width 5s height 5s border 5s, transform 5s;
      -webkit-transition: width 5s, height 5s, border 5s, -webkit-transform 5s;
    }

    p:hover {
      width: 200px;
      height: 100px;
      border: 10px solid red;
      transform: rotate(180deg);
      -webkit-transform: rotate(180deg);
    }
  </style>
</head>
<body>
  <p>마우스를 올려보세요.</p>
</body>
</html>
```

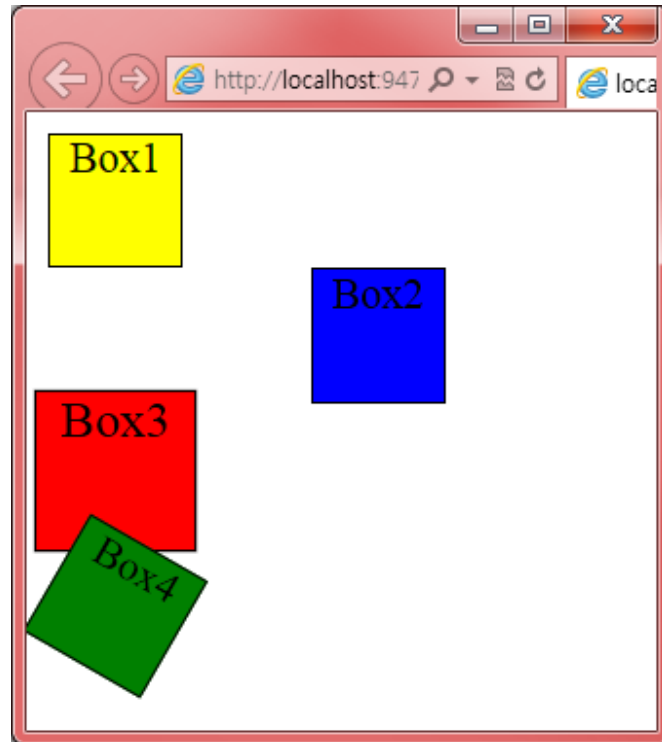


실행결과 보기



CSS3 변환

- 도형을 이동, 크기 변환, 회전
- 도형의 크기나 형태, 위치를 변환
- 2차원 또는 3차원적으로 변환





transform 속성

- `transform: translate(10px, 10px)` : 평행이동
- `transform: rotate(45deg)` : 회전
- `transform: scale(2, 1.2)` : 크기변환
- `transform: skew(20deg, 10deg)` : 비틀기 변환
- `transform: matrix()` : 일반적인 변환



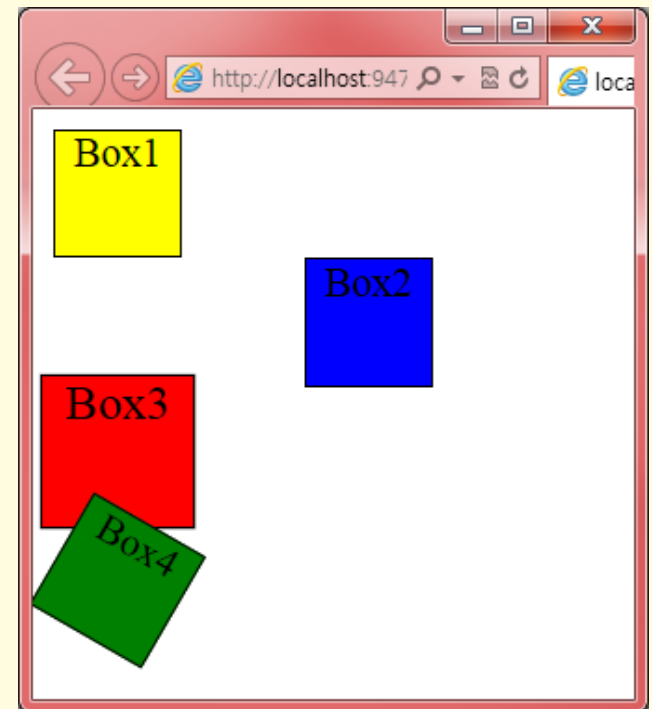
CSS3: 전환

```
<!DOCTYPE html>
<html>
<head>
  <style>
    div {
      width: 50px;
      height: 50px;
      background-color: yellow;
      border: 1px solid black;
      text-align: center;
    }

    div#box2 {
      transform: translate(100px, 0px);
      background-color: blue;
    }

    div#box3 {
      transform: scale(1.2, 1.2);
      background-color: red;
    }

    div#box4 {
      transform: rotate(30deg);
      background-color: green;
    }
  </style>
</head>
```





CSS3: 전환

```
<body>
```

```
  <div id="box1">Box1</div>
```

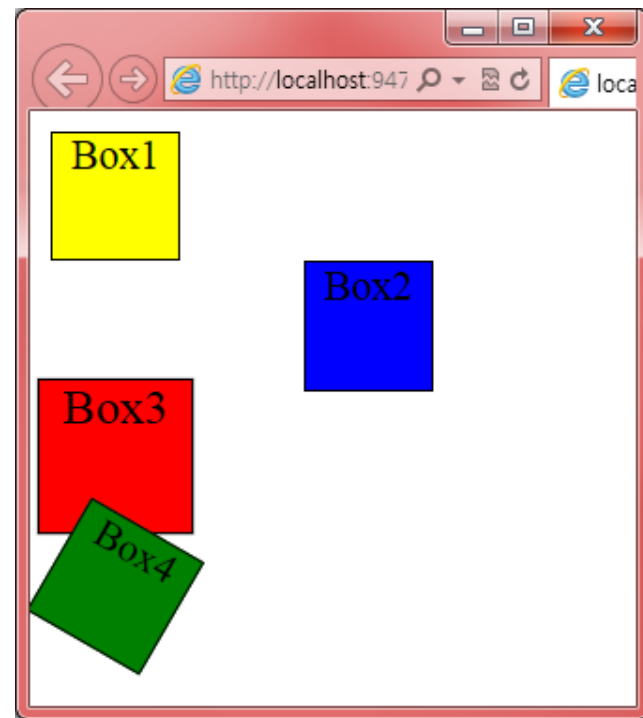
```
  <div id="box2">Box2</div>
```

```
  <div id="box3">Box3</div>
```

```
  <div id="box4">Box4</div>
```

```
</body>
```

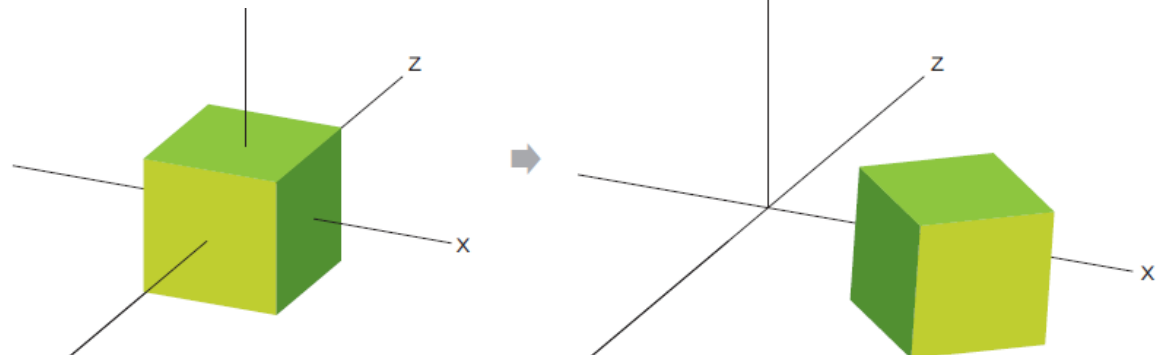
```
</html>
```





CSS3: 3차원 변환

메서드	설명
<code>translate3d(x,y,z)</code>	3차원 평행 이동
<code>translateX(x)</code>	3차원 평행 이동(x축)
<code>translateY(y)</code>	3차원 평행 이동(y축)
<code>translateZ(z)</code>	3차원 평행 이동(z축)
<code>scale3d(x,y,z)</code>	3차원 크기 변환
<code>scaleX(x)</code>	3차원 크기 변환(x축)
<code>scaleY(y)</code>	3차원 크기 변환(y축)
<code>scaleZ(z)</code>	3차원 크기 변환(z축)
<code>rotate3d(x,y,z,angle)</code>	3차원 회전 변환
<code>rotateX(angle)</code>	3차원 회전 변환(x축)
<code>rotateY(angle)</code>	3차원 회전 변환(y축)
<code>rotateZ(angle)</code>	3차원 회전 변환(z축)
<code>perspective(n)</code>	원근 변환

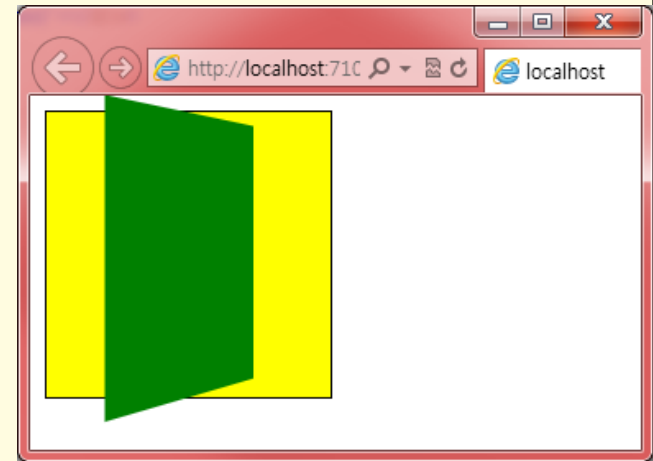




예제

```
<!DOCTYPE html>
<html>
<head>
  <style>
    div {
      background-color: green;
      height: 150px;

      width: 150px;
    }
    .container {
      background-color: yellow;
      border: 1px solid black;
    }
    .transformed {
      backface-visibility: visible;
      transform-origin: 50% 42%;
      transform: perspective(500px) rotateY(59deg) rotateX(0deg);
    }
  </style>
</head>
<body>
  <div class="container">
    <div class="transformed"></div>
  </div>
</body>
</html>
```



실행결과 보기



CSS3 애니메이션

키프레임을 지정한다. 애니메이션의 이름을 지정한다.

`@keyframes myanim`

{

0% {left:0px; top:0px;}

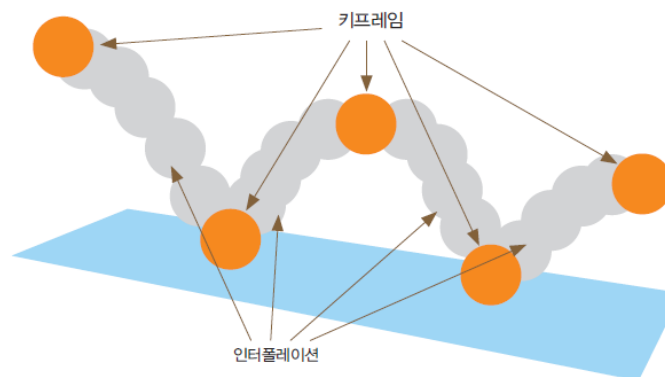
25% {left:100px; top:0px;} ← 25%일 때의 키프레임 지정

50% {left:200px; top:0px;}

75% {left:100px; top:0px;}

100% {left:0px; top:0px;}

}

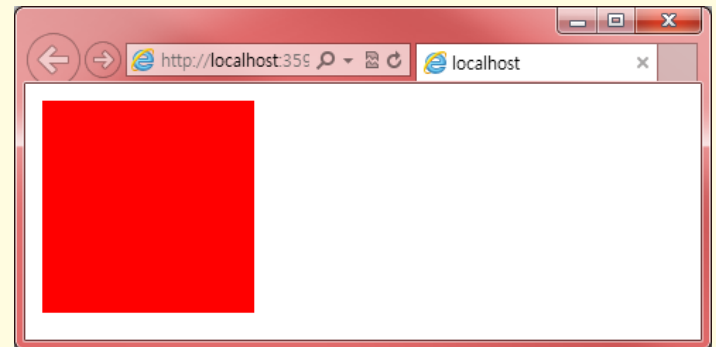




예제

```
<!DOCTYPE html>

<html>
<head>
  <style>
    div {
      width: 100px;
      height: 100px;
      background: red;
      position: relative;
      animation: 2s myanim;
      animation-iteration-count: 10;
    }
    @keyframes myanim
    {
      0%   {left:0px; top:0px;}
      25%  {left:100px; top:0px;}
      50%  {left:200px; top:0px;}
      75%  {left:100px; top:0px;}
      100% {left:0px; top:0px;}
    }
  </style>
</head>
<body>
  <div></div>
</body>
</html>
```



실행결과 보기



예제

```
<html>
<head>
  <title>튀어오르는 공</title>
  <style>
    @-webkit-keyframes bounce {
      from, to {
        bottom: 0px;
        -webkit-animation-timing-function: ease-out;
      }
      50% {
        bottom: 200px;
        -webkit-animation-timing-function: ease-in;
      }
    }
    #ball {
      position: absolute;
      width: 20px;
      height: 20px;
      background: red;
      border-radius: 10px;
      -webkit-animation-name: bounce;
      -webkit-animation-iteration-count: infinite;
      -webkit-animation-duration: 5s;
    }
  </style>
</head>
<body>  <div id="ball"> </div> </body>
</html>
```



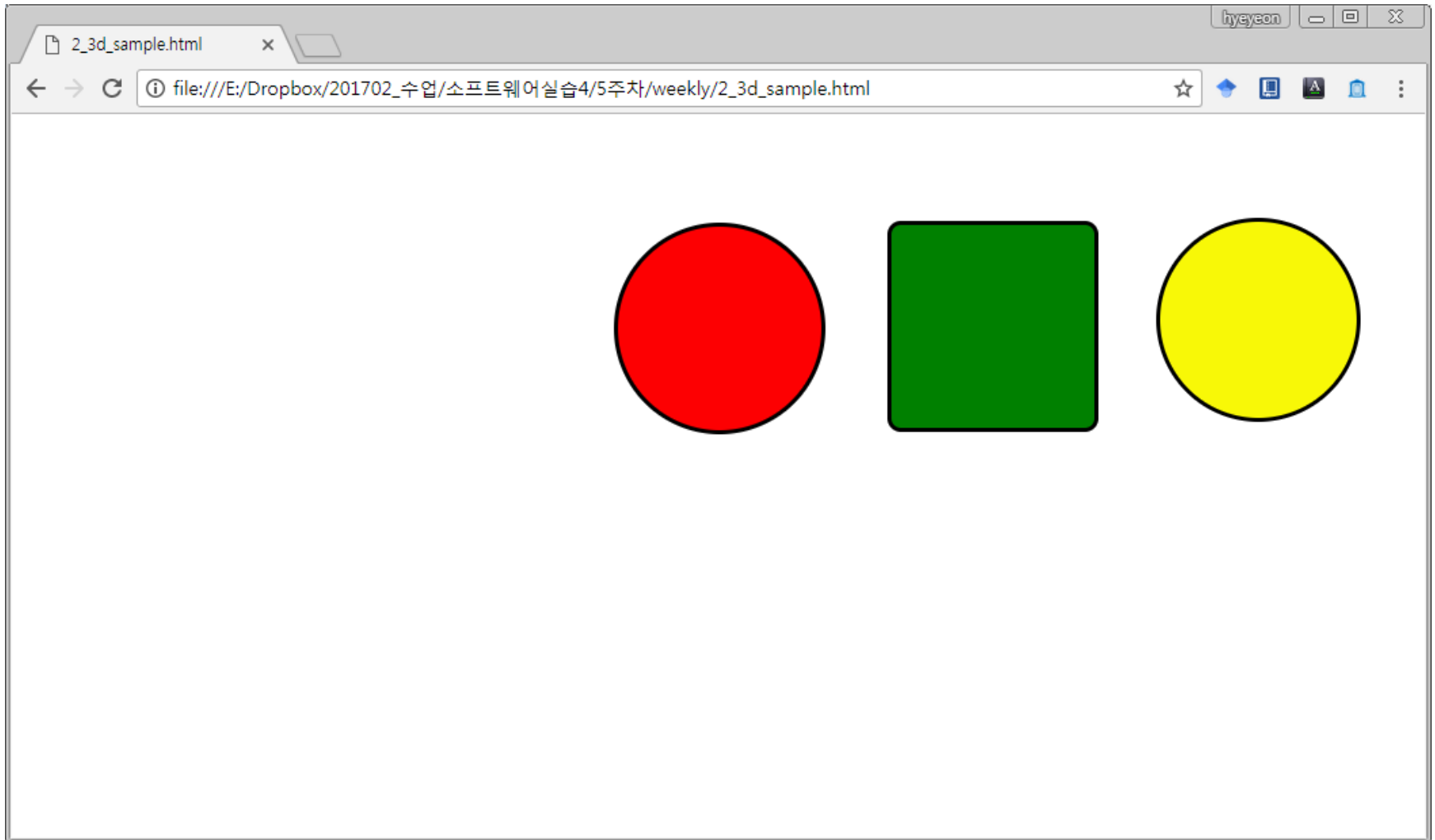
weekly - 1





weekly - 2

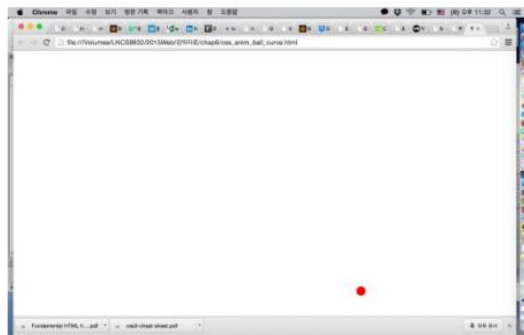
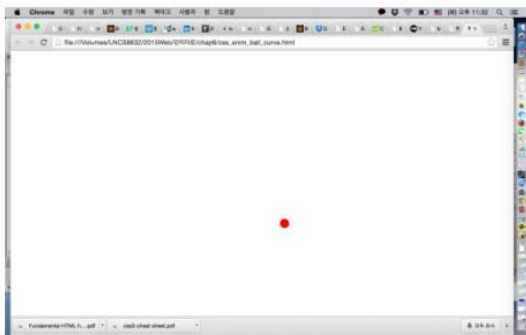
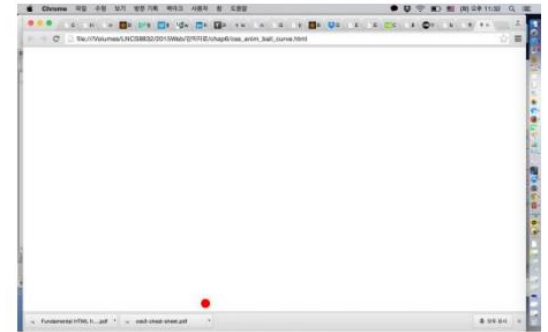
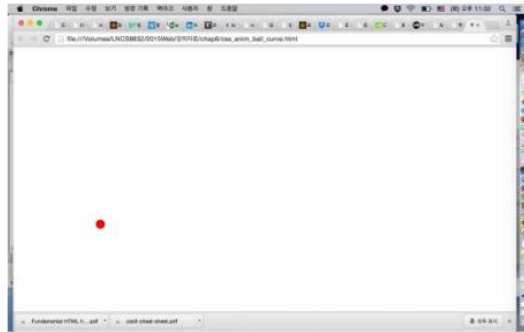
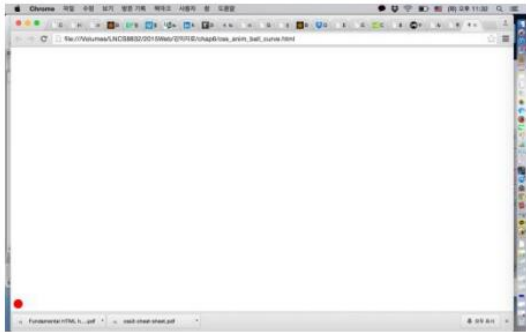
- change square





weekly - 3

- bounce ball





Q & A

