

# 06 CSS 레이아웃

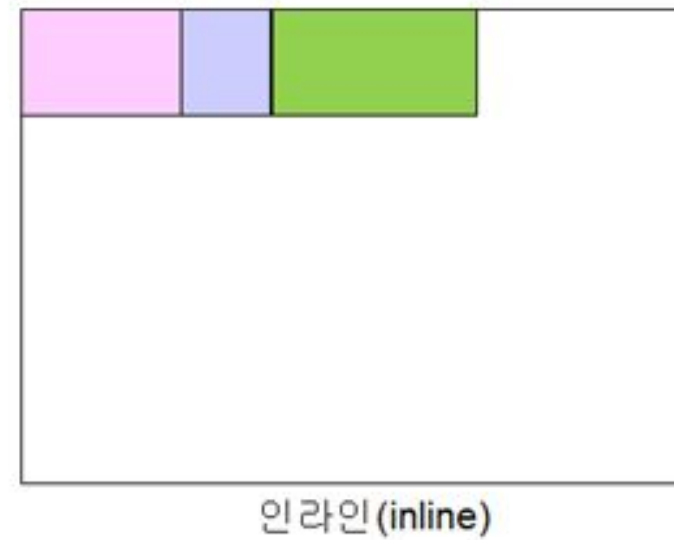
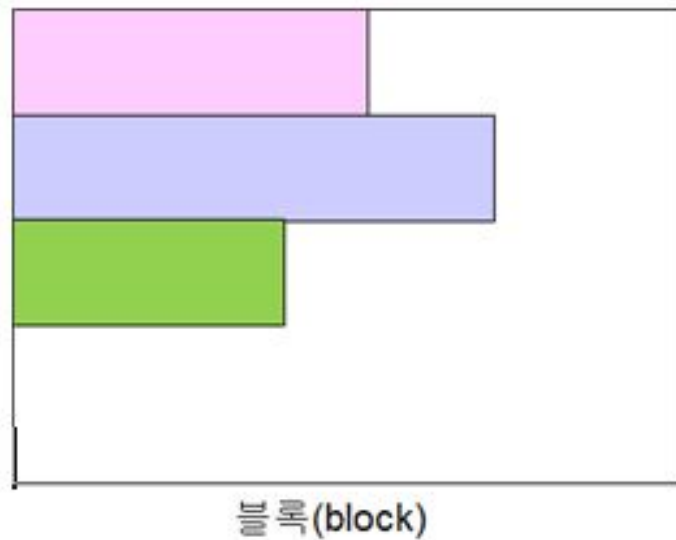
# 레이아웃이란?

- 웹 페이지에서 **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.



# 블록요소

- 한 줄을 전부 차지
- `<h1>`, `<p>`, `<ul>`, `<li>`, `<table>`,  
`<form>`, `<header>`, `<nav>`
- 예제 실행과 소스보기



div>

**`<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>, <img>, <strong>, <em>, <br>, <input>, <span> 요소

```
<body>
```

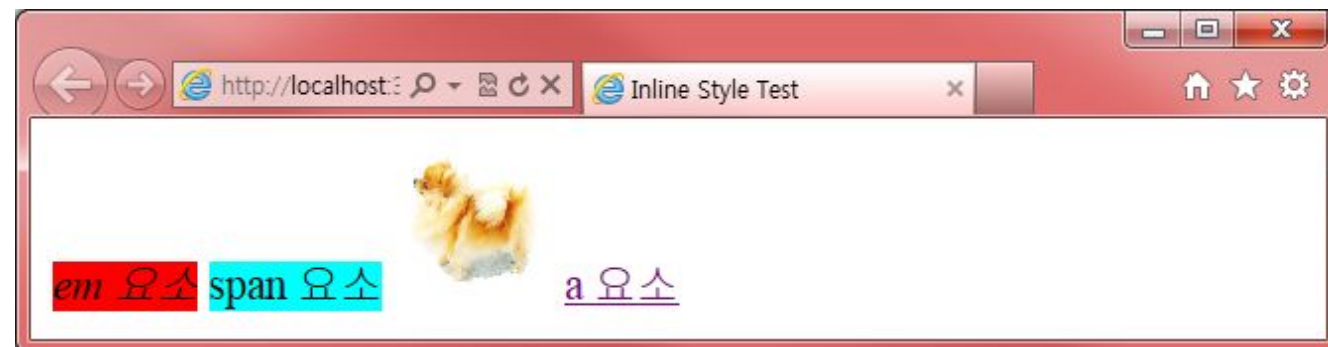
```
  <em style="background-color: red">em 요소</em>
```

```
  <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>
```

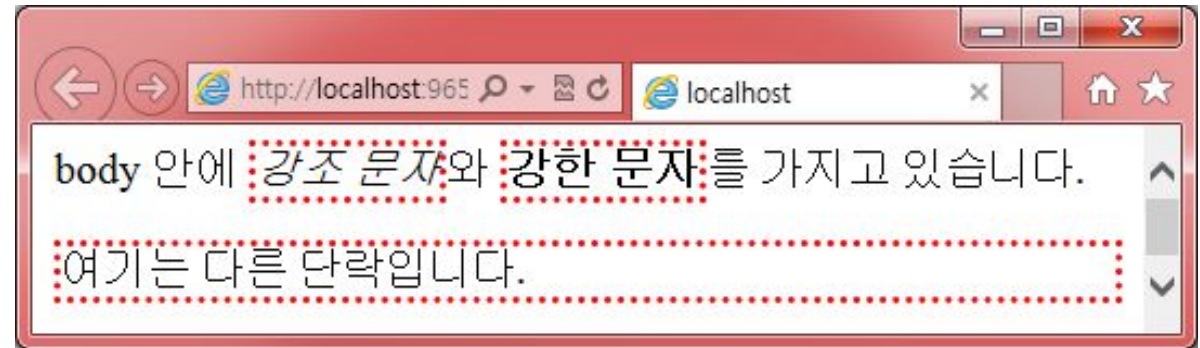
```
  body 안에
```

```
  <em>강조 문자</em>와 <strong>강한 문자</strong>를 가지고 있습니다.
```

```
  <p>여기는 다른 단락입니다. </p>
```

```
</body>
```

```
</html>
```

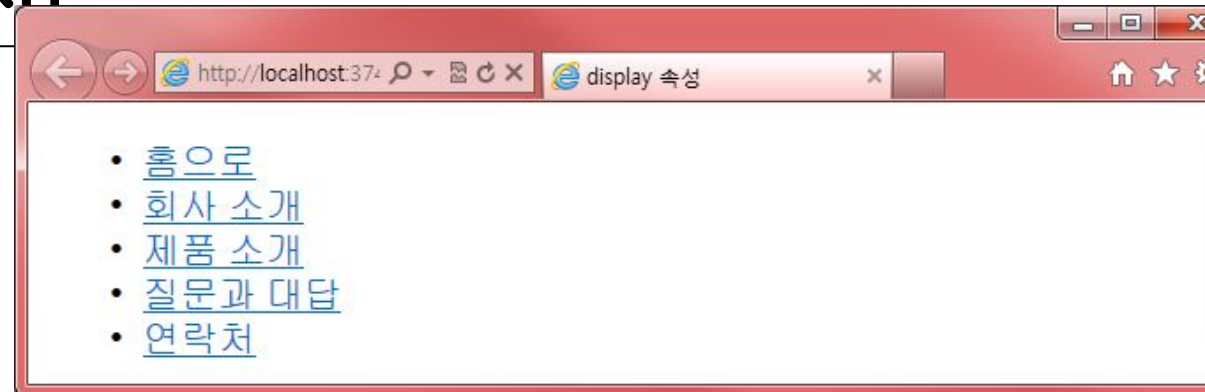


# CSS의 display 속성

- 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 : 화면에 나타나지 않음
  - visibility:hidden : 화면에서 감춰짐

# 예제

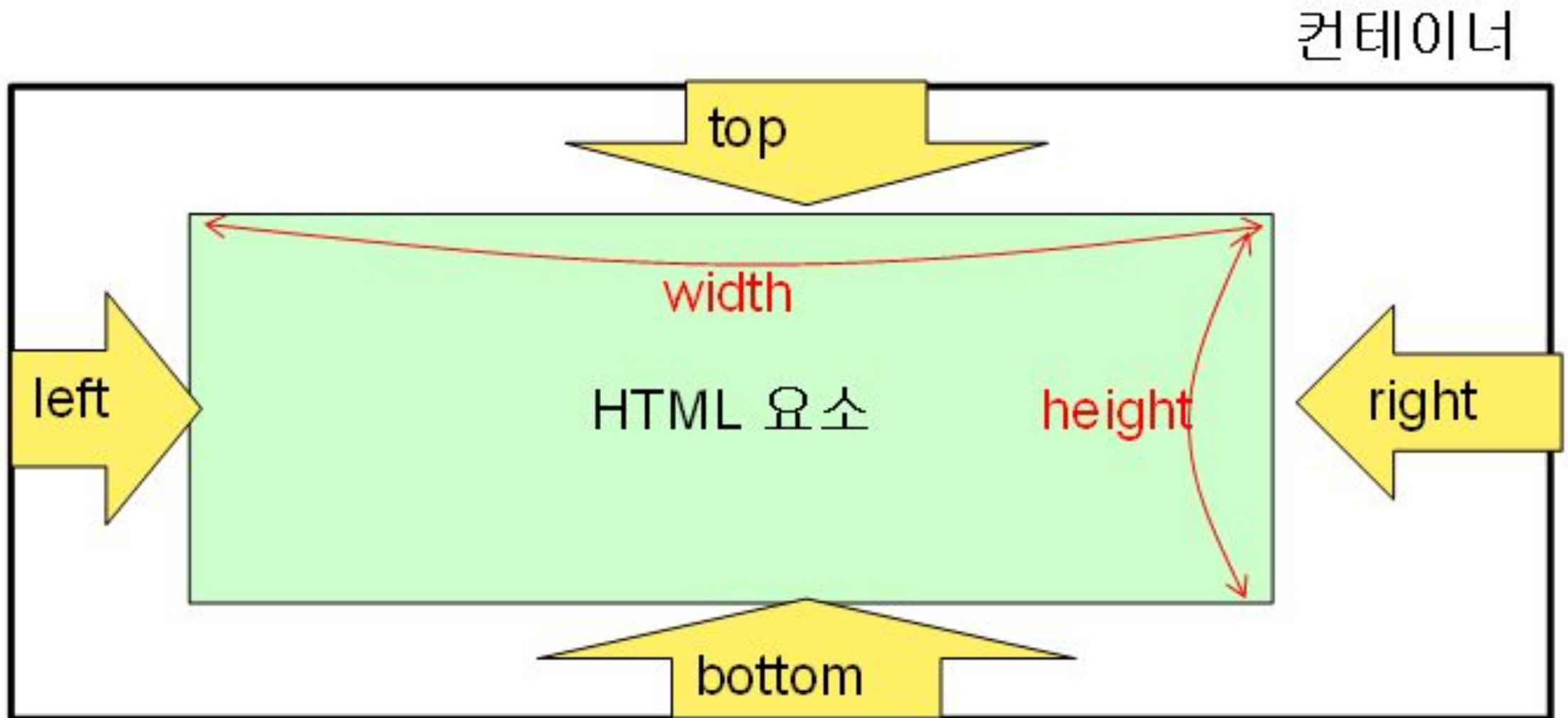
```
<!DOCTYPE html>
<html>
<head>
<title>display 속성</title>
<style>
.menu li {display: inline; background-color: yellow; margin: 0;
         border: 1px solid; border-color: red; padding: .5em;}
a { text-decoration : none; }
</style>
</head>
<body>
  <ul class="menubar">
    <li><a href="#">홈으로</a></li>
    <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



# 위치 설정 방법

- 정적 위치 설정(static positioning) - 정상적인 흐름에 따른 배치
- 상대 위치 설정(relative positioning) - 정상적인 위치가 기준점이 된다.
- 절대 위치 설정(absolute positioning) - 컨테이너의 원점이 기준점이 된다.
- 고정 위치 설정(fixed positioning) - 윈도우의 원점이 기준점이 된다.

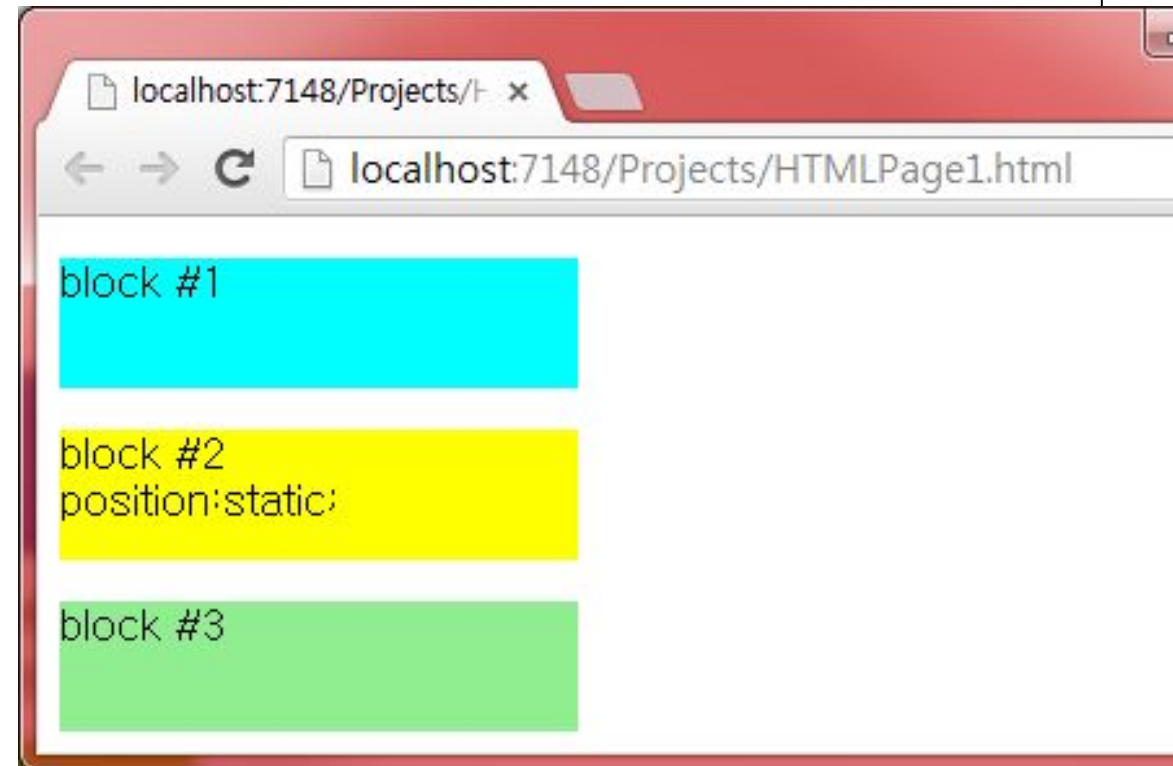
# 정적 위치 설정

- 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 상대적

```
<style>
```

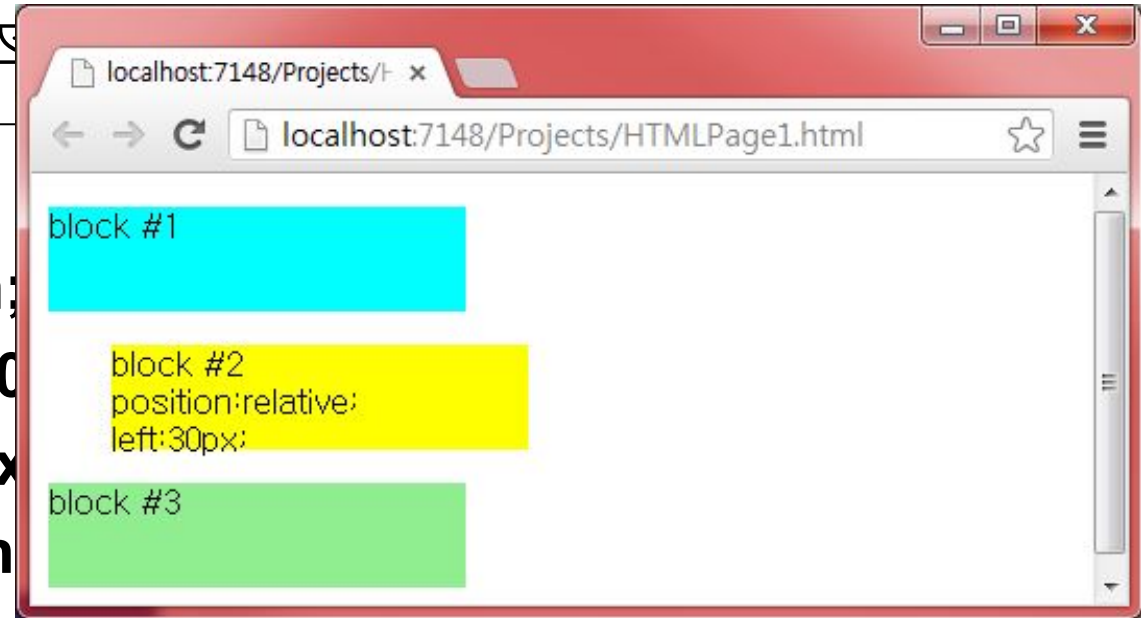
```
#one {background-color: cyan;
```

```
#two {position: relative; left: 30px;
```

```
width: 200px; height: 50px;
```

```
#three {background-color: lightgreen;
```

```
</style>
```



# 절대 위치 설정

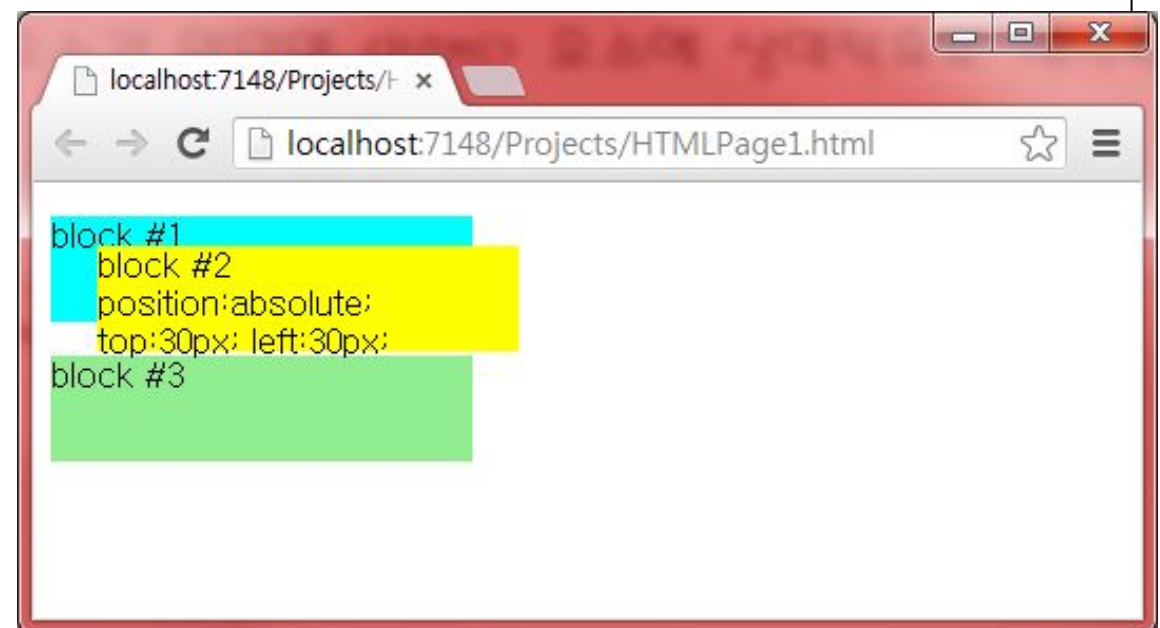
- 절대 위치(**absolute positioning**)

- 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;  
}
```

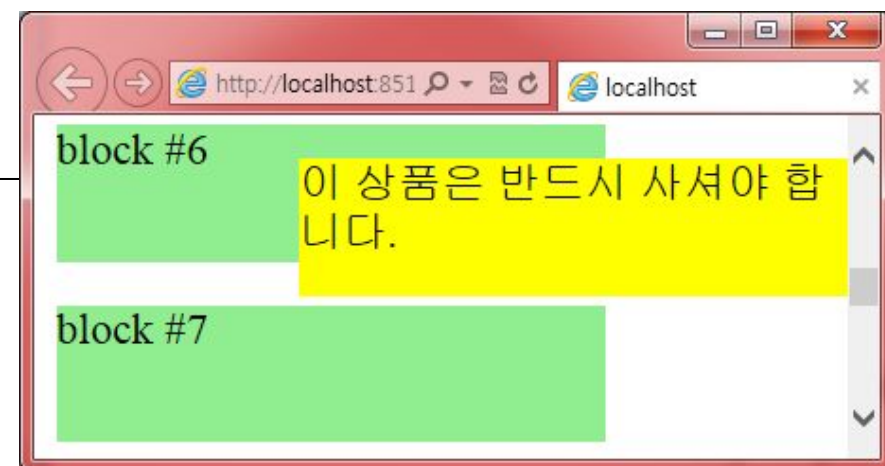
...



# 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>**

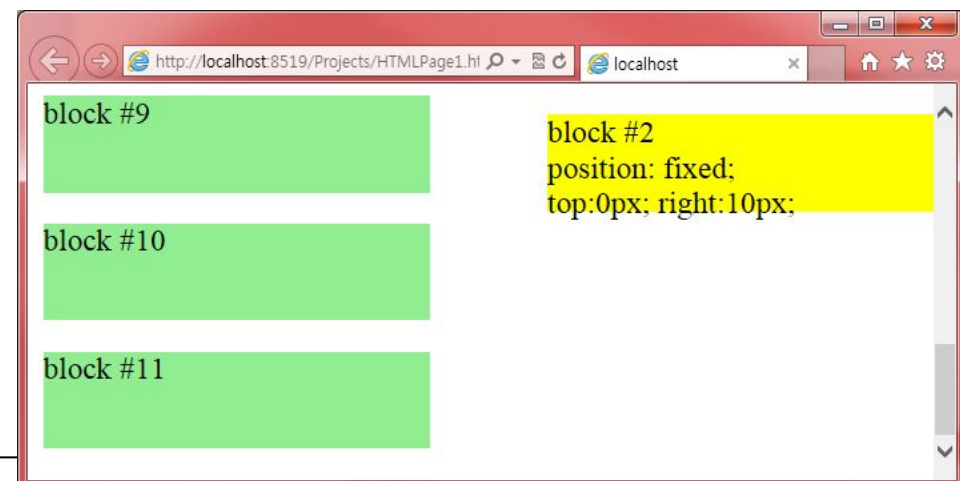
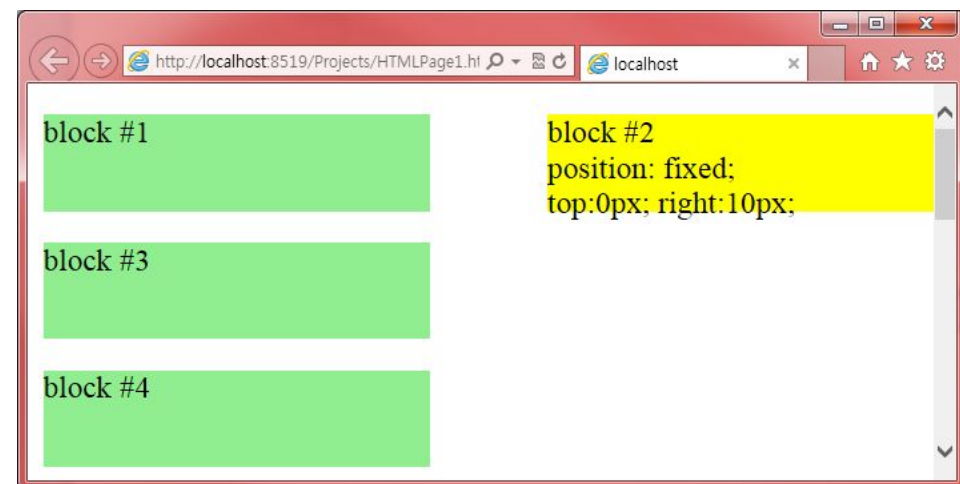
**<p>block #9</p>**

**<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>
```

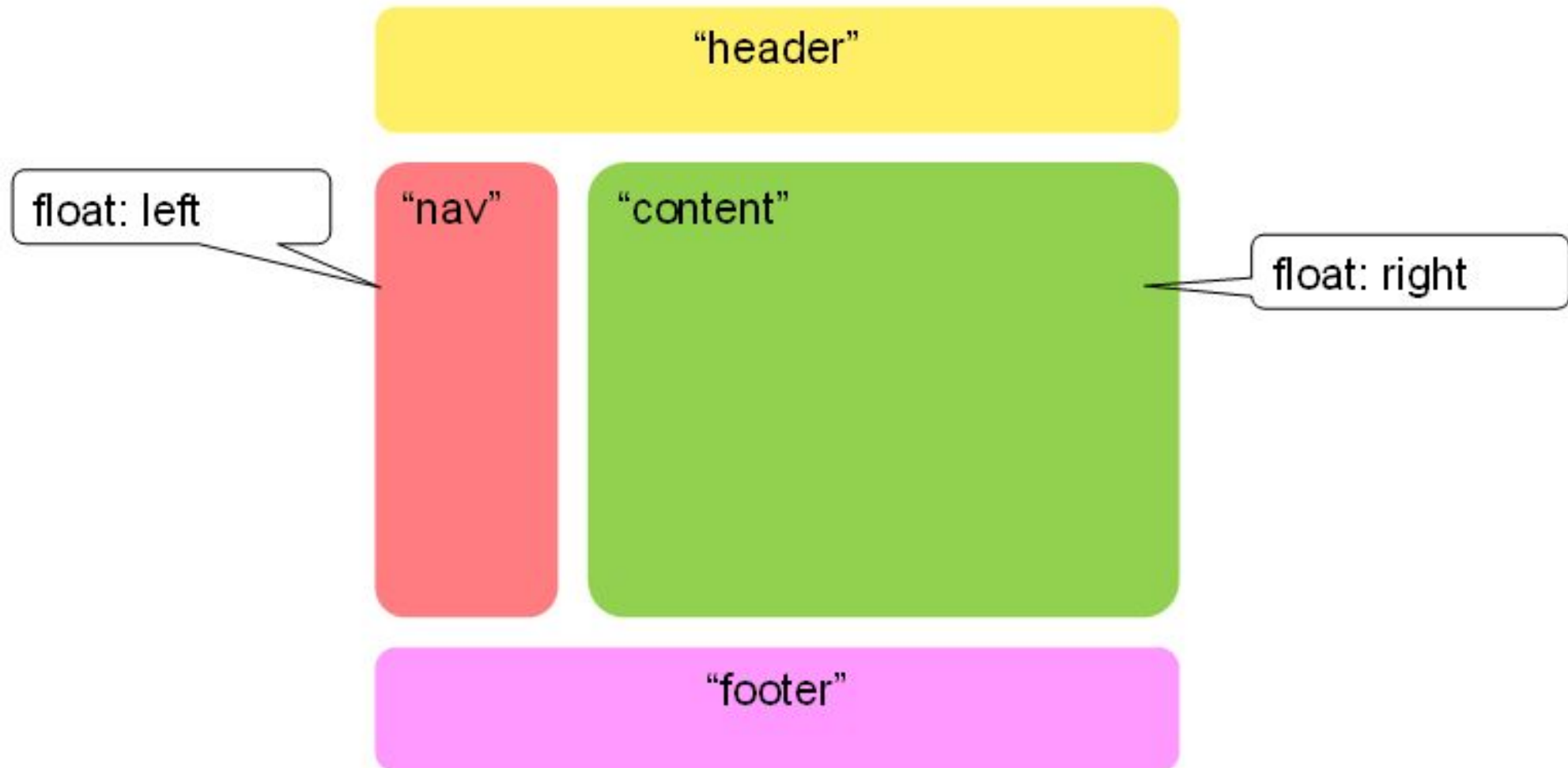


# 예제

```
<!DOCTYPE html>
<html>
<head>
<style>
img {float:left; width: 110px; height: 90px; margin: 5px; }
</style>
</head>
<body>
  <h3>이미지 갤러리</h3>
  
  
  
  
  
  
</body>
</html>
```

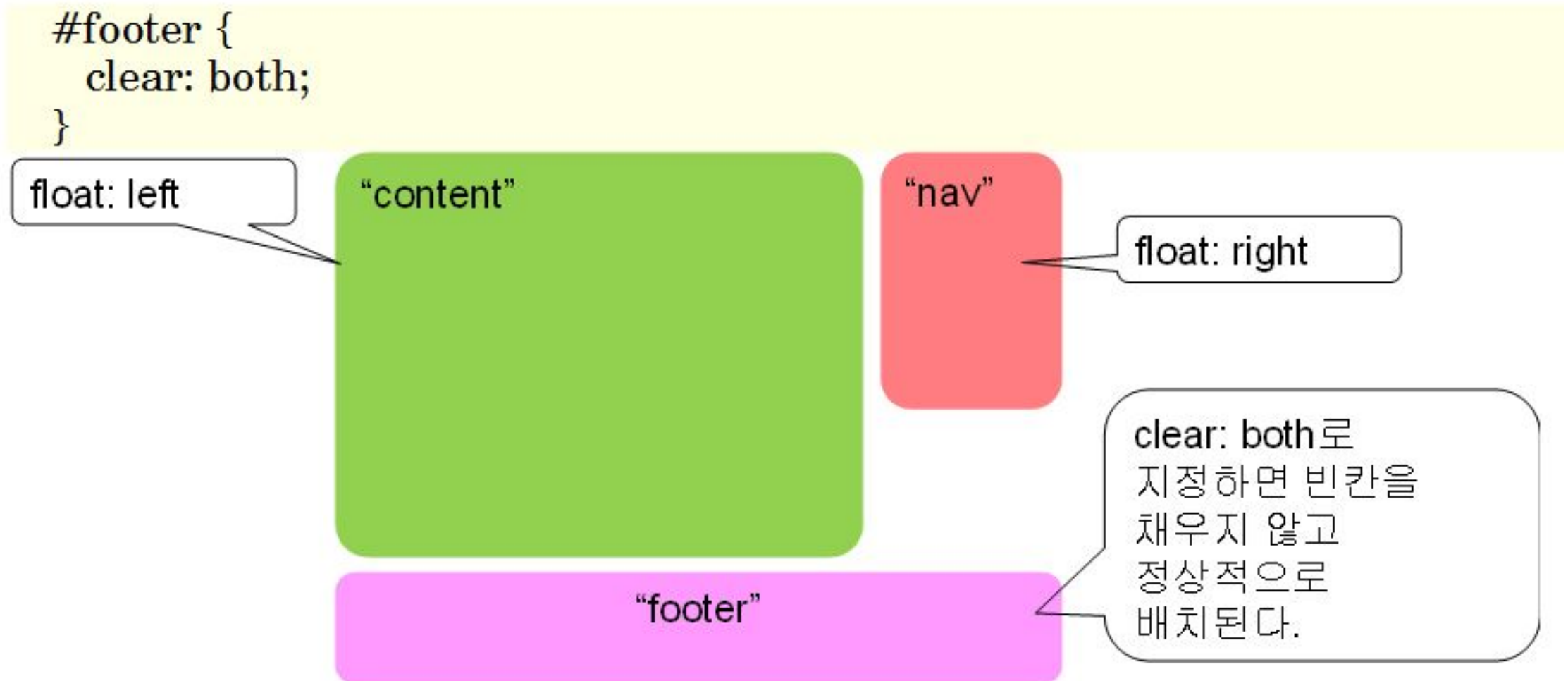
# float의 용도

- 레이아웃에 많이 사용된다.



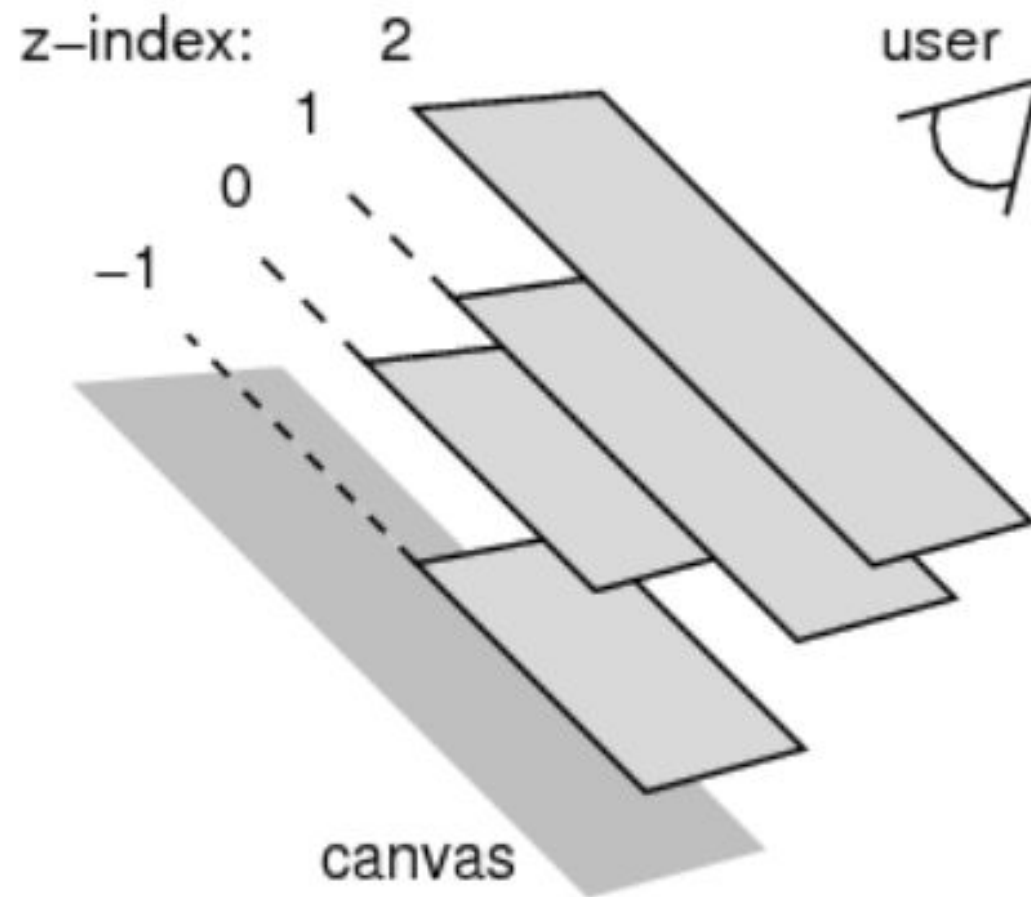
# clear 속성

- float 속성을 중단할 때 사용된다.



# z-index

- 요소의 스택 순서를 지정



# 예제

...

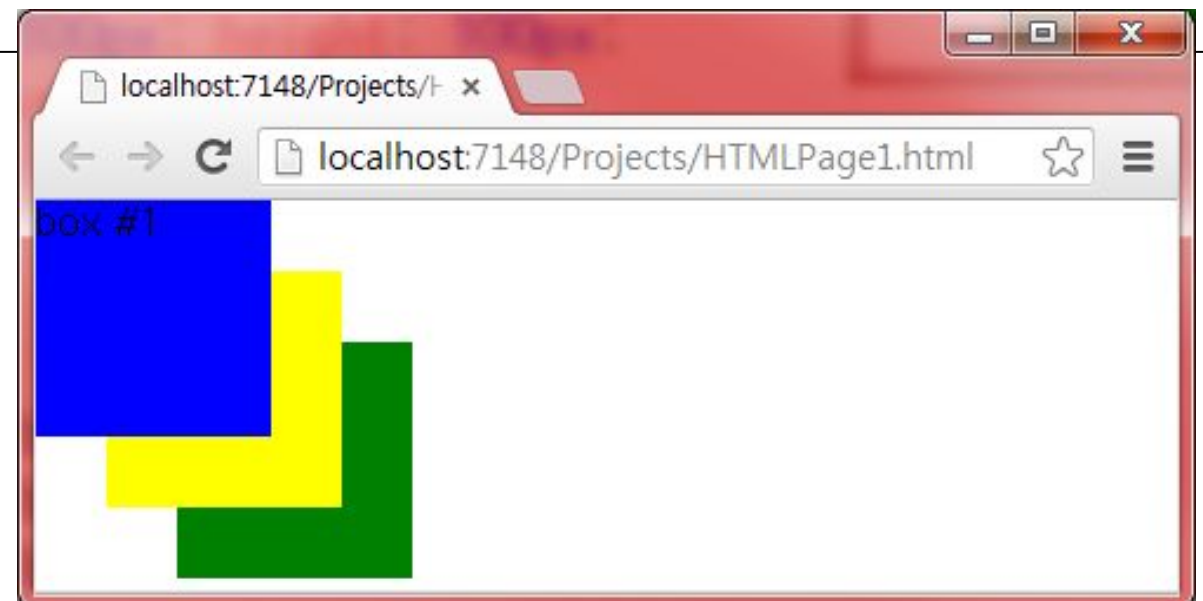
**<style>**

**#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; }**

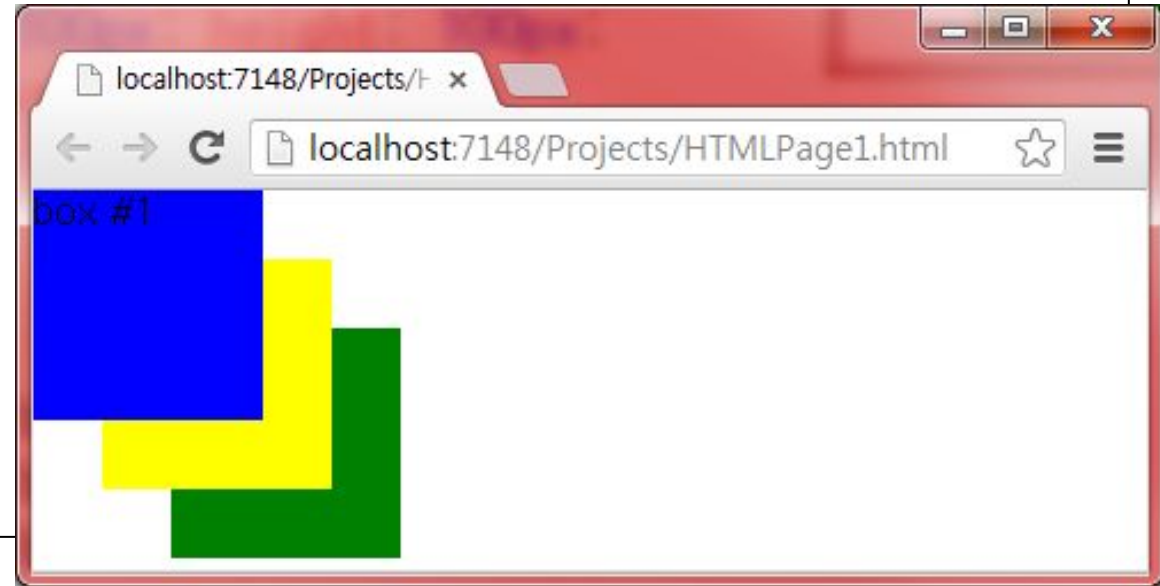
**#box3 {position: absolute; top: 60px; left: 60px;  
width: 100px; height: 100px; background: green; z-index: 0; }**

**</style>**



# 예제

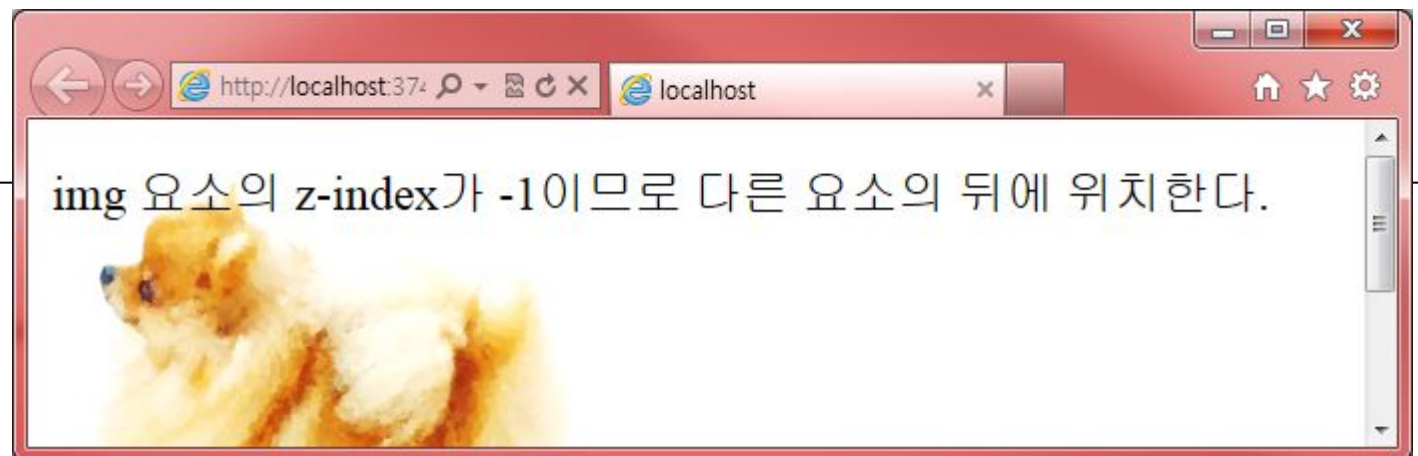
```
</head>  
<body>  
  <div id="box1">box #1 </div>  
  <div id="box2">box #2 </div>  
  <div id="box3">box #3 </div>  
</body>  
</html>
```





# 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>
```



# 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./안한다
  - **auto** – 자동으로 스크롤 바가 나타난다.  
–생길수도 있고 안생길수 도 있다

# 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>
```

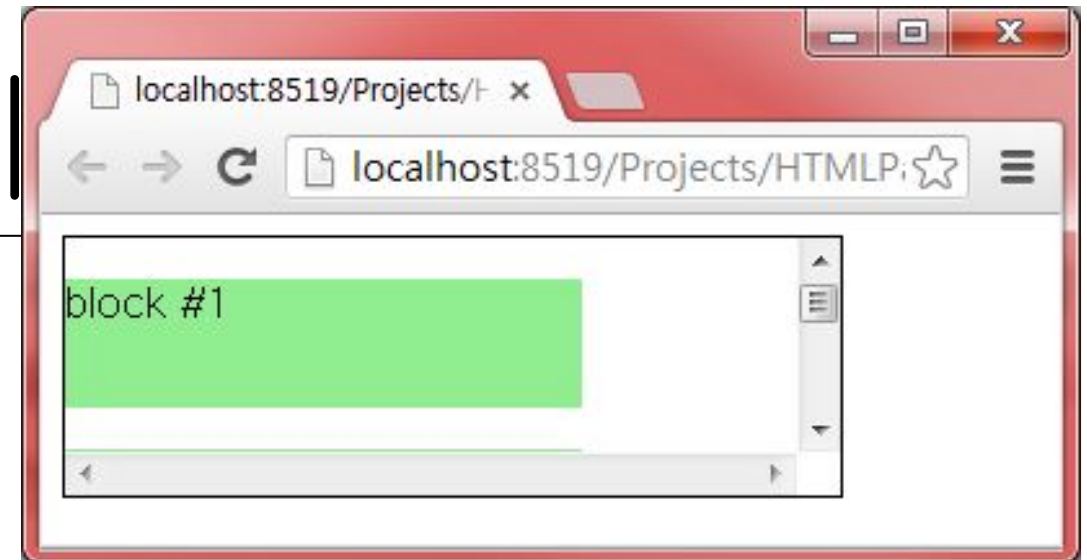
```
  <p>block #4</p>
```

```
  <p>block #5</p>
```

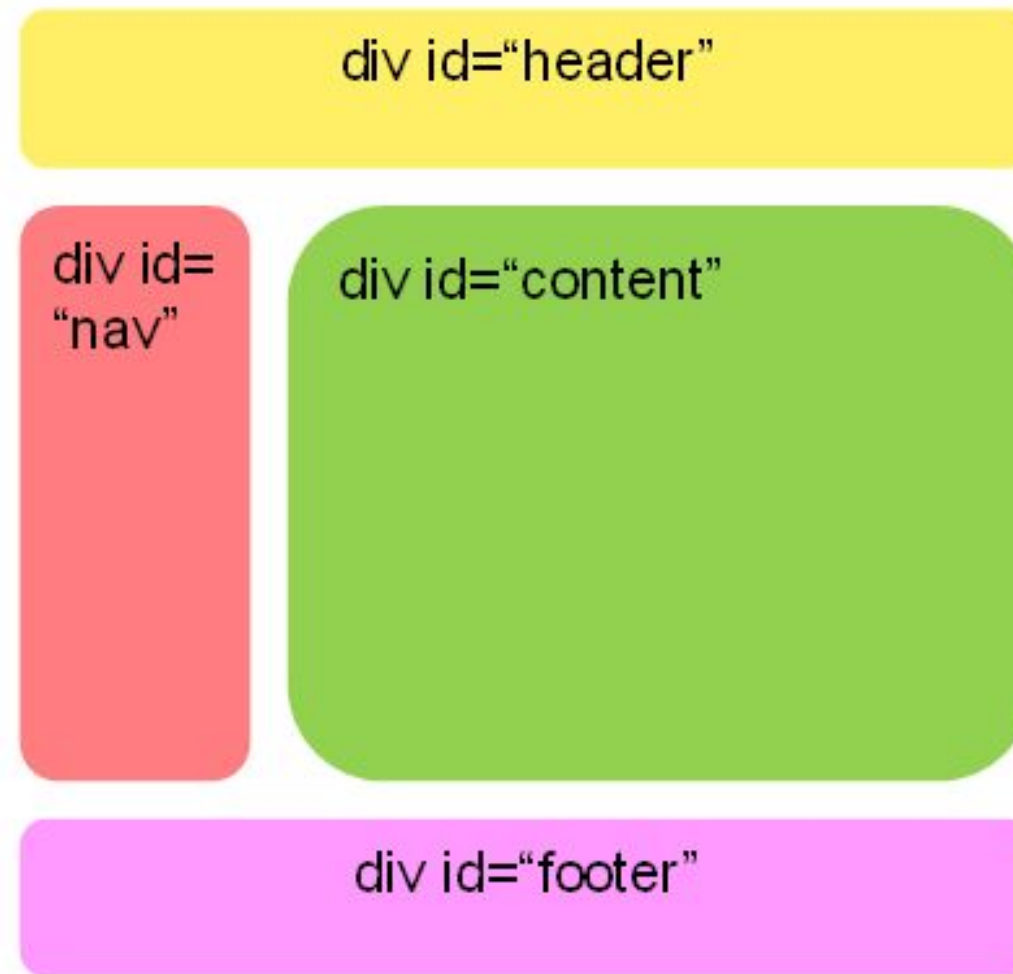
```
</div>
```

```
</body>
```

```
</html>
```



# <div>를 이용한 레이아웃

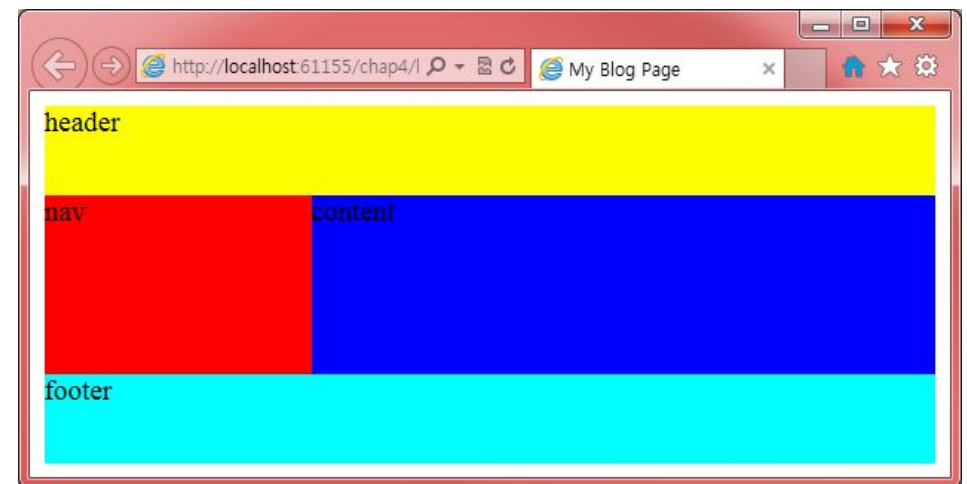


# 예제

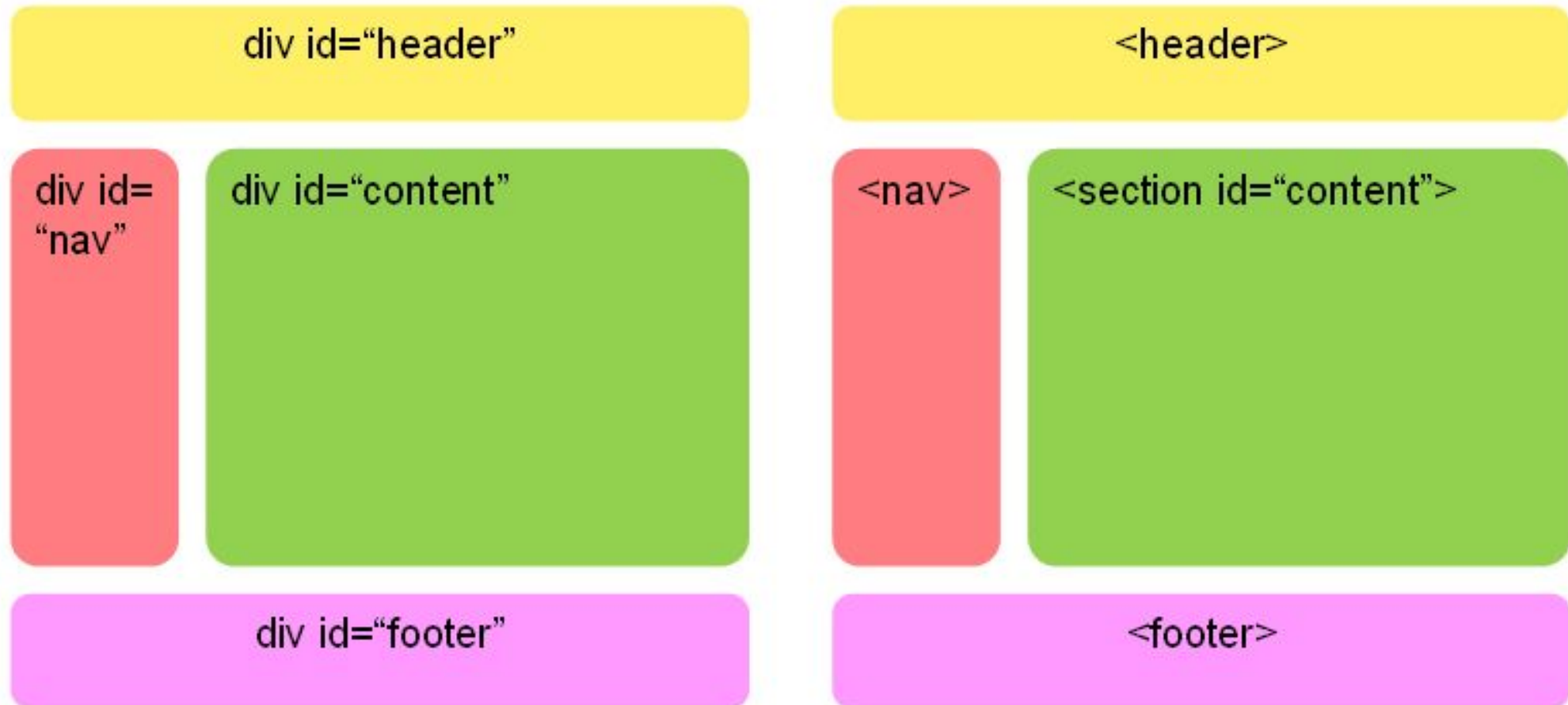
```
<!DOCTYPE html>
<html>
<head>
  <title>My Blog Page</title>
  <style>
#header {
  background-color: yellow;
  width: 100%;
  height: 50px;
}
#nav {
  width: 30%;
  background-color: red;
  height: 100px;
  float: left;
}
```

# 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
<header>	문서의 머리말(header)
<hgroup>	<h1>에서 <h6>요소들의 그룹
<nav>	내비게이션 링크
<article>	문서의 내용이나 블로그의 포스트
<section>	문서의 섹션을 의미한다.
<aside>	사이드바와 같이 옆에 위치하는 내용
<footer>	문서의 꼬리말/footer)
<figure>	그림이나 도표 <figcaption>홍길동</figcaption> </figure>



# 연습(Layout2)



# Layout

## My Blog Page

시멘틱 태그로 만들어진 페이지 입니다.

### Links

- [w3schools](#)
- [Naver](#)
- [Daum](#)



스폰지밥

### Semantic Tags

시멘틱 요소(Semantic elements)들은 브라우저에게 요소의 의미나 목적을 명확하게 알려주는 요소이다. 레이아웃을 만들때 쓰는 것. 의미가 있는 요소라면 div보다는 시멘틱 태그를 사용해준다.

### div와 span

div은 "divide"의 약자로서 페이지를 논리적인 섹션으로 분리하는데 사용되는 태그이다. span요소는 인라인 요소로서 텍스트를 위한 컨테이너로 사용할 수 있다.

Copyright (c)2019 Hong