

06 CSS 레이아웃

레이아웃이란?

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



레이아웃의 표시(1/3)

- 디스플레이(display) 속성

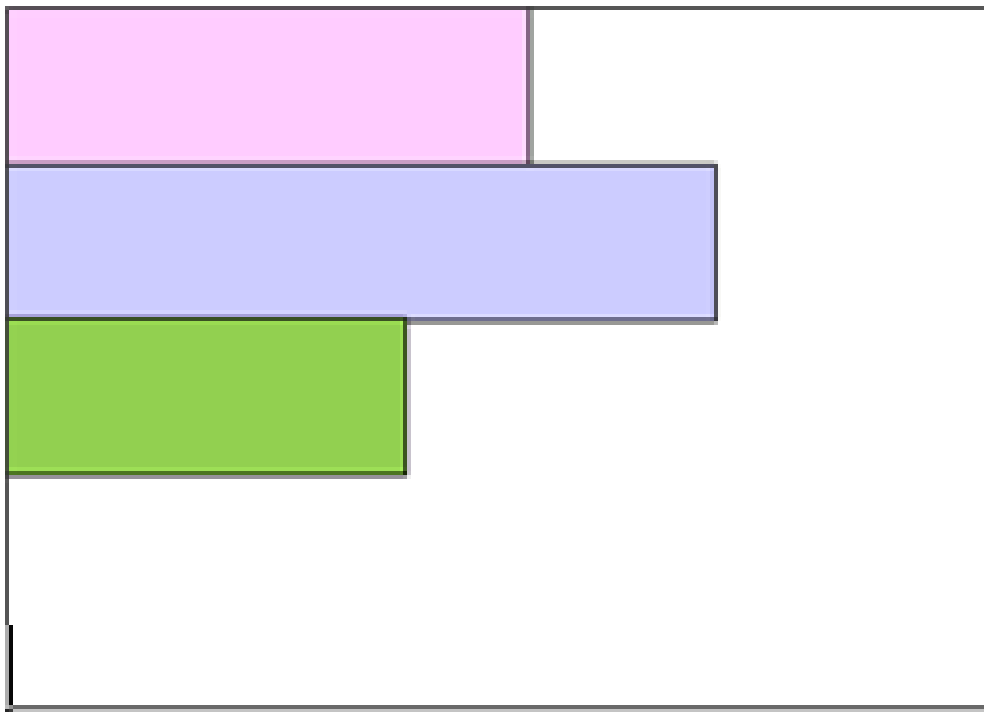
속성값	사용예	설명
block	<i>display: block;</i>	블록 레벨 표시
inline	<i>display: inline;</i>	인라인 레벨로 표시
inline-block	<i>display: inline-block;</i>	인라인 블록 레벨로 표시 인라인 레벨 + 높이
none	<i>display: none;</i>	화면에서 없는 것으로 간주 화면에 나타나지 않음

- display와 visibility의 차이

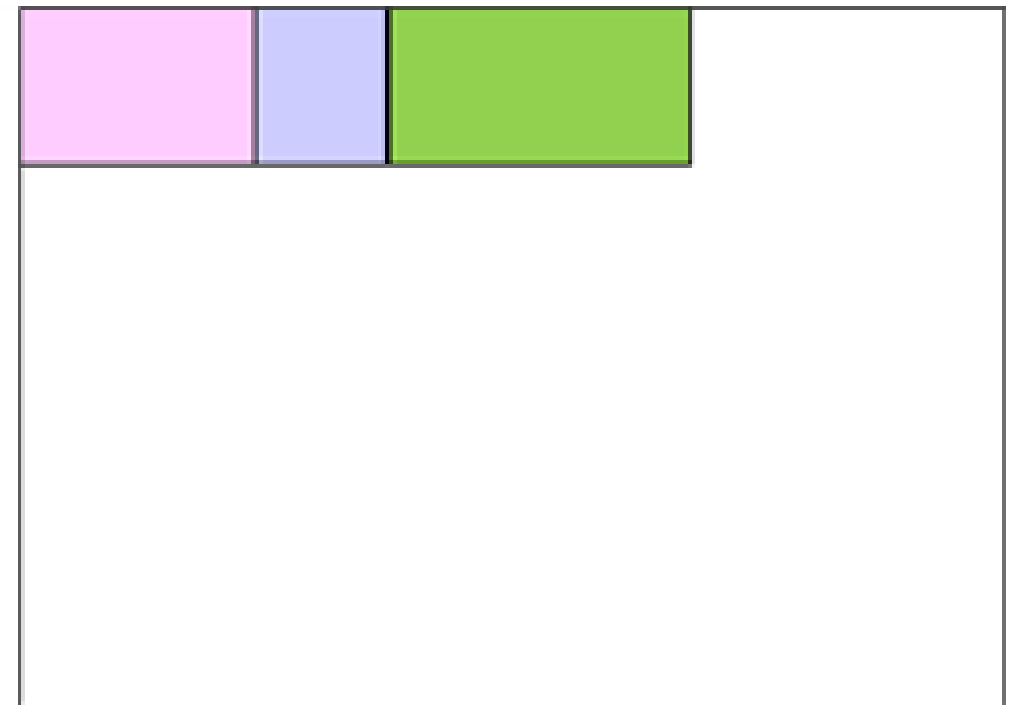
- { display: none; } - 요소가 숨겨지고 페이지에 해당 요소가 없는 것처럼 표시됨
- { visibility: hidden; } - 요소가 숨겨지지만 공간은 차지하고 있음

레이아웃의 표시 (2/3)

- 블록(block) 레벨 요소 - 항상 새 줄에서 시작. 한 줄 전부 차지.
 - <h1>, <p>, , , <table>, <blockquote>, <pre>, <div> <form>, <header>, <nav> 등 요소
- 인라인(inline) 레벨 요소 - 현재 줄에서 필요한 만큼의 너비만을 차지
 - <a>, , , , <input>, 등 요소



블록(block)



인라인(inline)

레이아웃의 표시 (3/3)

```
<body>
```

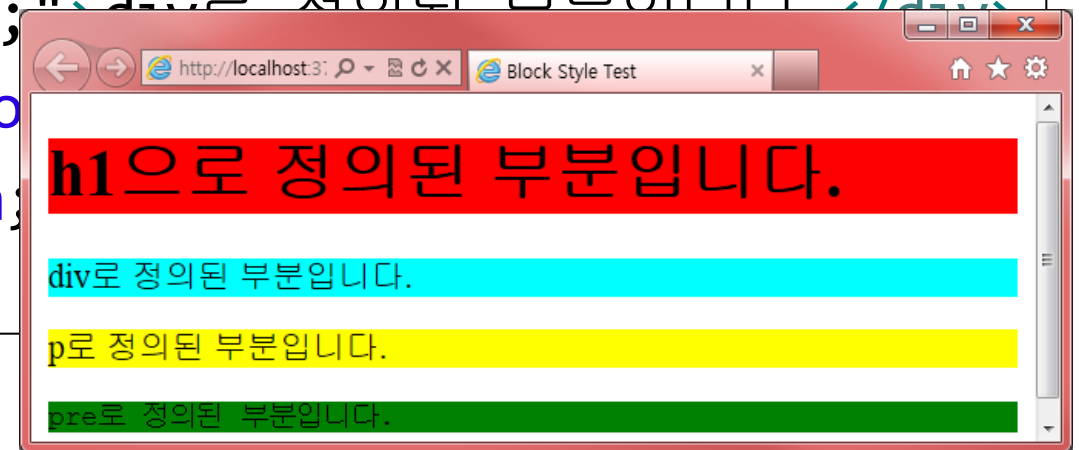
```
<h1 style="background-color: red;">h1으로 정의된 부분입니다.</h1>
```

```
<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>
```



```
<body>
```

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

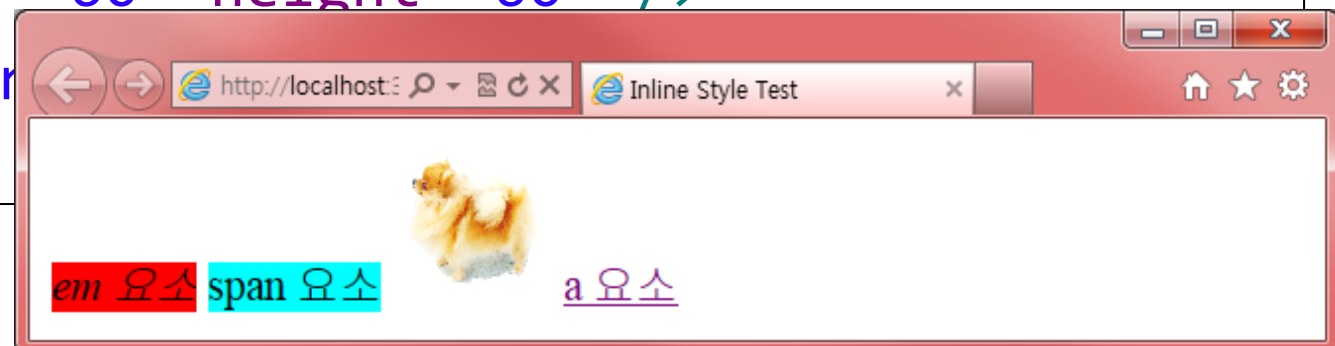
```
<span style="background-color: aqua;">span 요소</span>
```

```

```

```
<a href="http://www.w3c.org">a 요소</a>
```

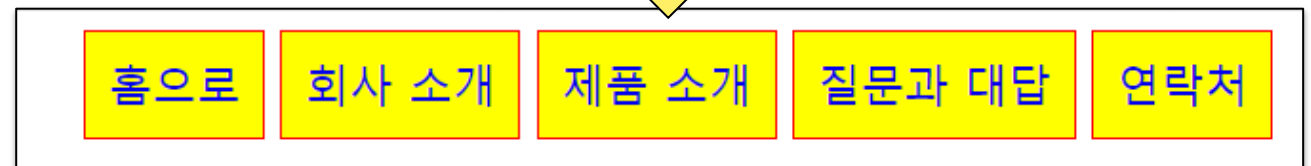
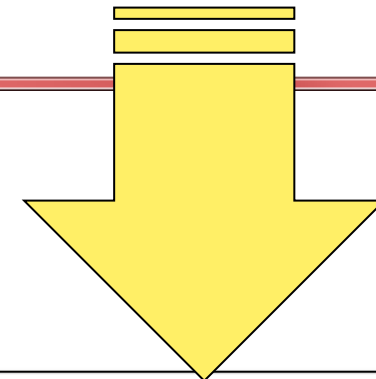
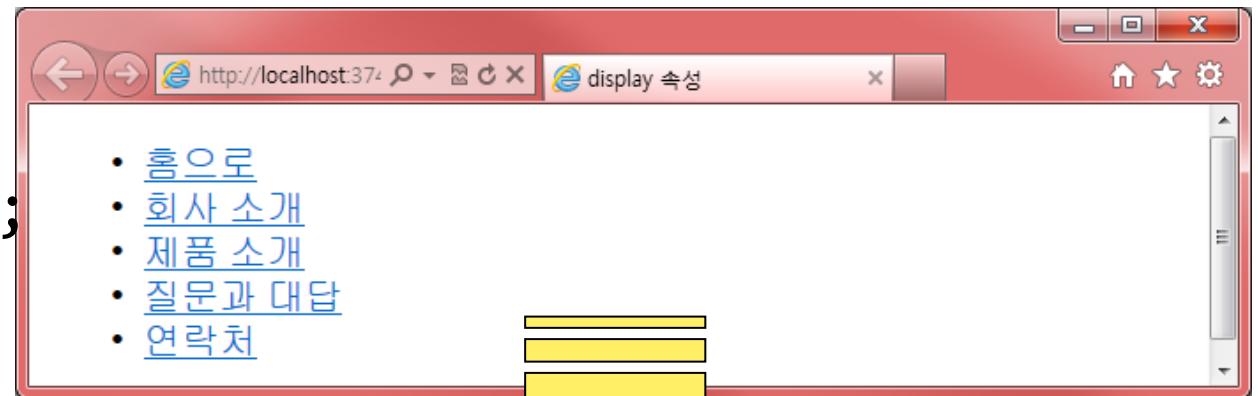
```
</body>
```



예제 1

```
<style>
.menubar li {
    display: inline;
    background-color: yellow;
    border: 1px solid red;
    padding: 0.5em;
    margin: 0;
}
a { text-decoration : none; }
</style>
```

```
...
<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>
```



예제2(1/2)

display: none;

동해물과 백두산이 마르고
닳도록 대안으로 길이보전
하세

display: inline;

동해물과 백두산이 마르고
닳도록 **대한사람** 대안으
로 길이보전하세

display: block;

동해물과 백두산이 마르고
닳도록

대한사람

대안으로 길이보전하세

display: inline-block;

동해물과 백두산이 마르고
닳도록 **대한사람** 대안으

로 길이보전하세

```
<body>
  <h2>display: none;</h2>
  <div>
    동해물과 백두산이 마르고 닳도록
    <p class="ex1">대한사람</p> 대안으로 길이보전하세
  </div>

  <h2>display: inline;</h2>
  <div>
    동해물과 백두산이 마르고 닳도록
    <p class="ex2">대한사람</p> 대안으로 길이보전하세
  </div>

  <h2>display: block;</h2>
  <div>
    동해물과 백두산이 마르고 닳도록
    <p class="ex3">대한사람</p> 대안으로 길이보전하세
  </div>

  <h2>display: inline-block;</h2>
  <div>
    동해물과 백두산이 마르고 닳도록
    <p class="ex4">대한사람</p> 대안으로 길이보전하세
  </div>
</body>
```

예제2 (2/2)

display: none;
동해물과 백두산이 마르고
닳도록 대한으로 길이보전
하세

display: inline;
동해물과 백두산이 마르고
닳도록 **대한사람** 대한으
로 길이보전하세

display: block;
동해물과 백두산이 마르고
닳도록
대한사람
대한으로 길이보전하세

display: inline-block;
동해물과 백두산이 마르고
닳도록 **대한사람** 대한으
로 길이보전하세

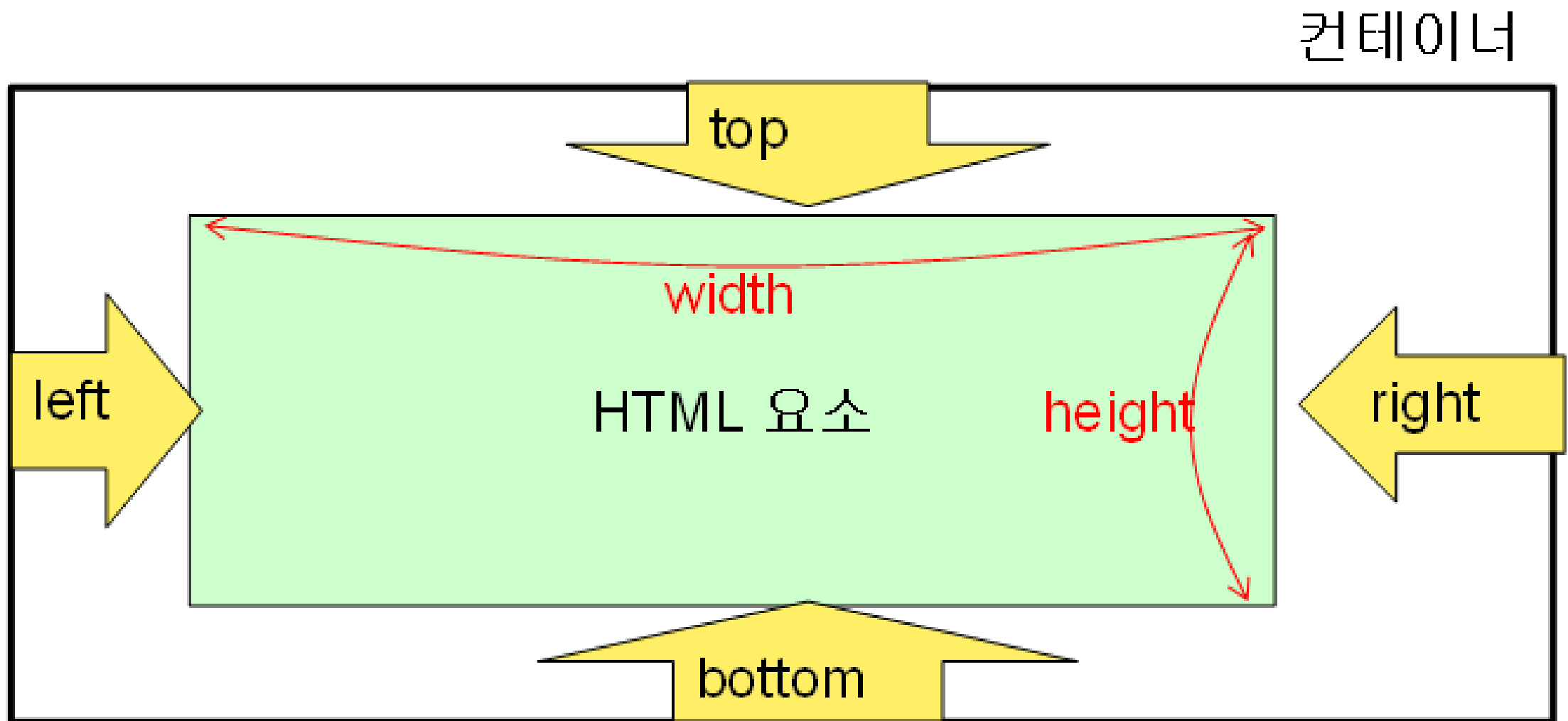
```
<style>
  h2 {margin: 0px;}

  div {width: 200px; border:1px dashed;}
  p {
    color: red; border:10px solid;
    height:100px; margin:0px
  }

  p.ex1 {display: none;}
  p.ex2 {display: inline;}
  p.ex3 {display: block;}
  p.ex4 {display: inline-block;}
</style>
```


레이아웃의 위치 (1/7)

- 위치 설정 방법에 따라
top, bottom, left, right 속성으로 결정



레이아웃의 위치 (2/7)

- 위치 설정 방법
 - static (정적 위치) - 정상적인 흐름에 따른 배치
 - relative (상대적 위치) - 정상적인 위치가 기준점이 된다.
 - absolute (절대적 위치) - 컨테이너의 원점이 기준점이 된다.
 - fixed (고정 위치) - 윈도우의 원점이 기준점이 된다.

레이아웃의 위치 (3/7)

- 정적 위치 설정(static positioning)

- 블록 요소들은 박스처럼 상하로 쌓이게 되고 인라인 요소들은 한 줄에 차례대로 배치
- top, bottom, left, right 영향 받지 않음

```
<style>
  p, div {width: 200px; height: 50px;}
  #ex1 {background-color: cyan; }
  #ex2 {background-color: yellow;
       position: static;}
  #ex3 {background-color: lightgreen;}
</style>
</head>
<body>
  <p id="ex1">블럭1</p>
  <div id="ex2">블럭2 (position:static;)
  </div>
  <p id="ex3">블럭3</p>
</body>
```

블럭1

블럭2 (position:static;)

블럭3

레이아웃의 위치 (4/7)

- 상대 위치 설정(relative positioning)
 - 정상적인 위치를 기준으로 입력된 위치에 해당 요소가 배치

```
<style>  
p, div {width: 200px; height: 50px;}  
#ex1 {background-color: cyan; }  
#ex2 {background-color: yellow;  
      position: relative;  
      left: 30px;}  
#ex3 {background-color: lightgreen;}  
</style>
```

...

```
<p id="ex1">블럭1</p>  
<div id="ex2">블럭2 (position:relative;)</div>  
<p id="ex3">블럭3</p>
```

블럭1

블럭2 (position:relative;)

블럭3

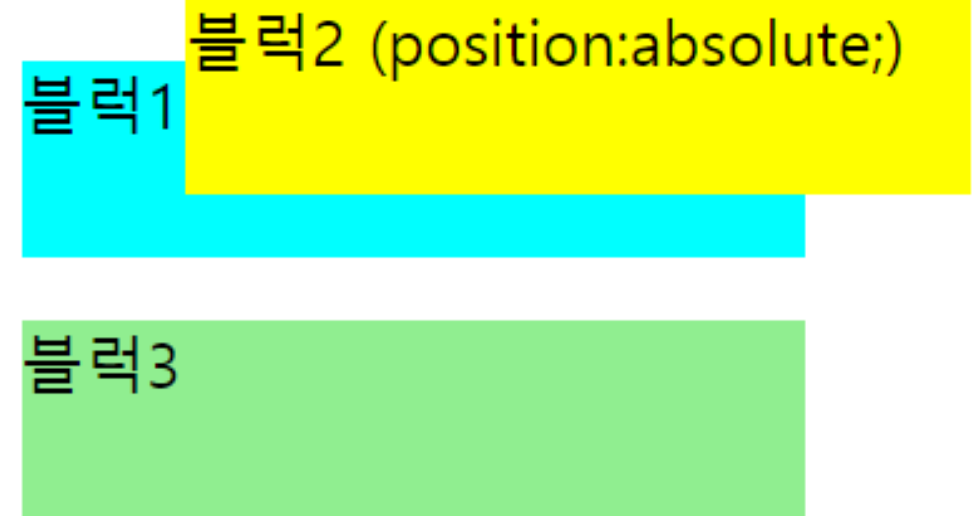
레이아웃의 위치 (5/7)

- 절대 위치(absolute positioning)

- '위치 지정'이 된 가까운 상위 요소를 기준으로 입력된 위치에 해당 요소가 배치 (기준 요소가 없으면 전체 화면을 기준으로 배치)

```
<style>
  p, div {width: 200px; height: 50px;
  #ex1 {background-color: cyan; }
  #ex2 {background-color: yellow;
        position: absolute;
        top: 30px;
        left: 30px;}
  #ex3 {background-color: lightgreen;}
</style>
```

```
...
  <p id="ex1">블럭1</p>
  <div id="ex2">블럭2 (position:absolute;)
  </div>
  <p id="ex3">블럭3</p>
```

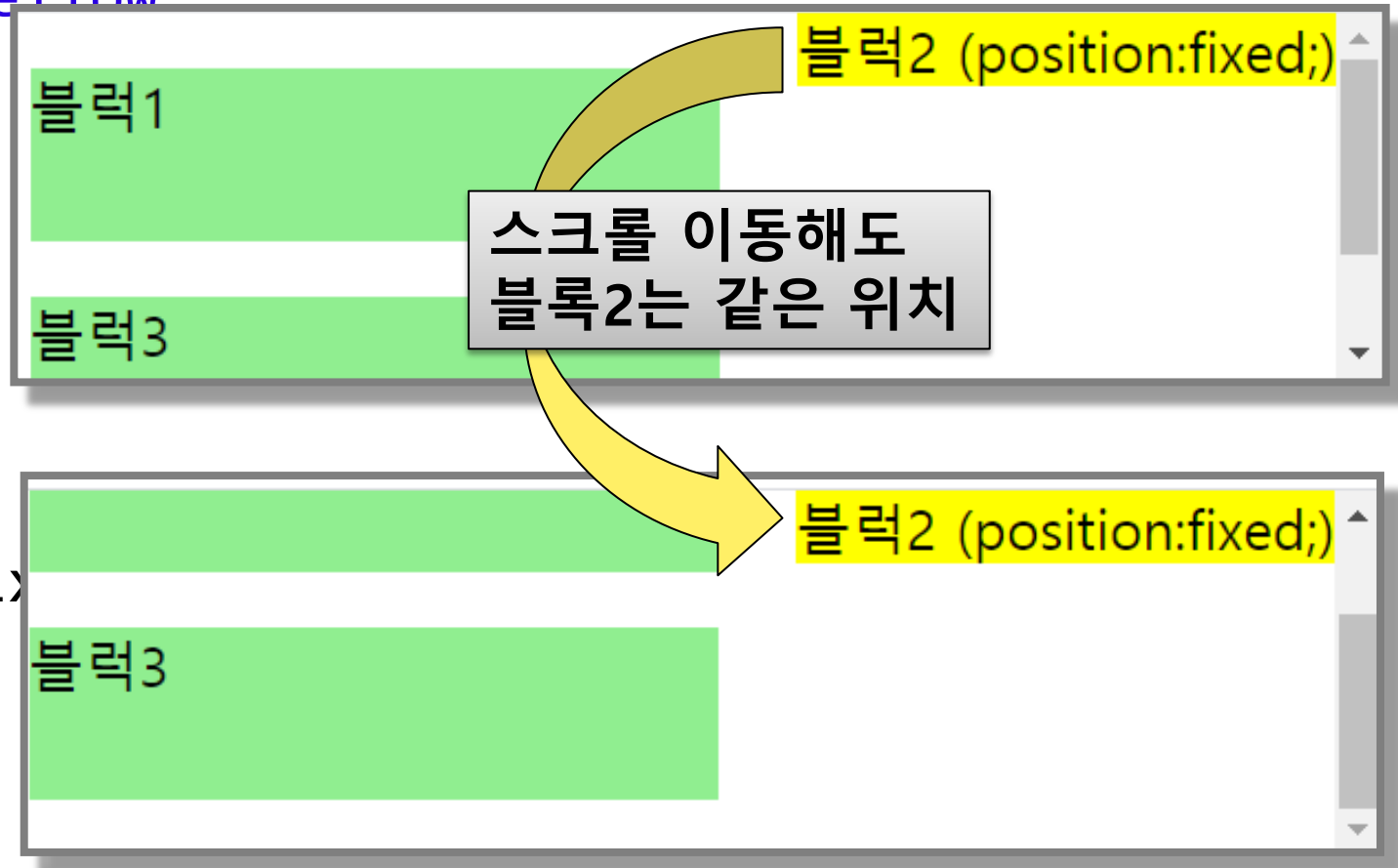


레이아웃의 위치 (6/7)

- 고정 위치 설정(fixed positioning)
 - 브라우저 윈도우를 기준으로 요소의 위치를 잡는 것

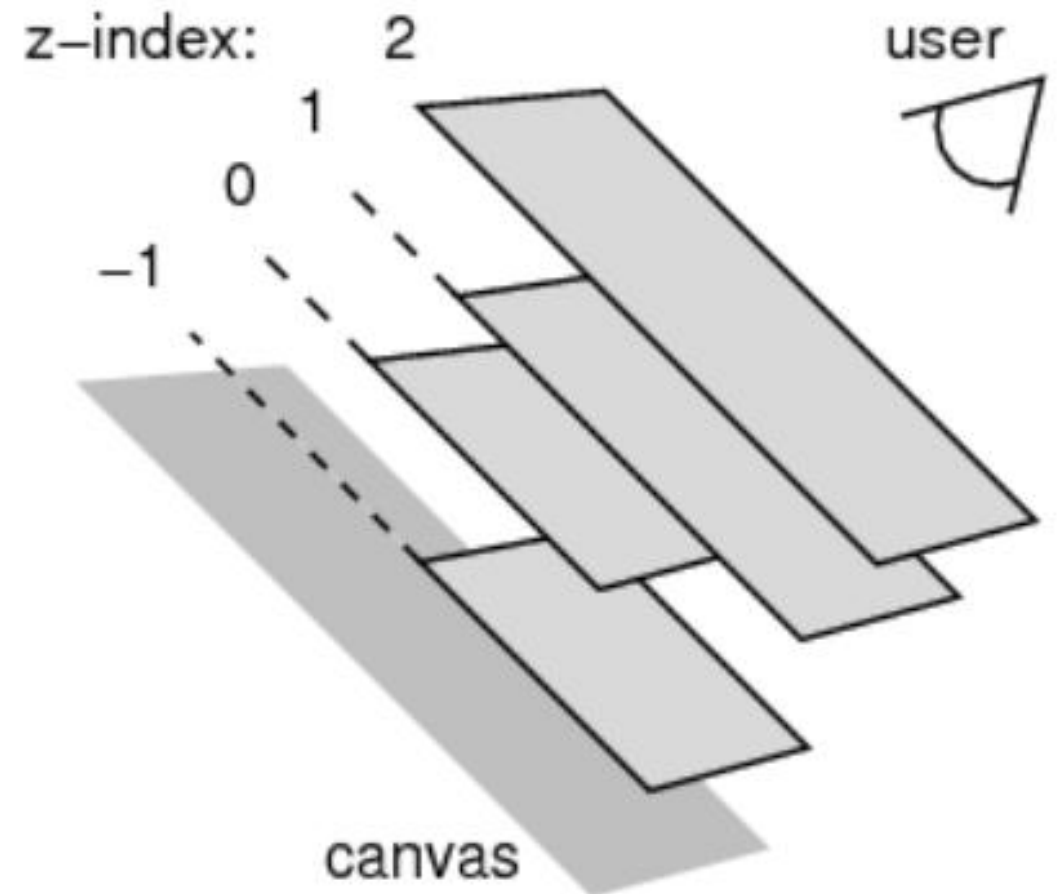
```
<style>
  p {background-color: lightgreen; width: 200px; height: 50px;}
  #ex2 {
    background-color: yellow;
    position: fixed;
    top: 0px; right: 0px;
  }
</style>
```

```
...
<p id="ex1">블럭1</p>
<div id="ex2">
  블럭2 (position:fixed)
</div>
<p id="ex3">블럭3</p>
</body>
```



레이아웃의 위치 (7/7)

- 겹치는 요소 발생 시
 - 요소가 배치되면서 다른 요소와 겹치는 경우, z-index 속성으로 요소의 스택 순서를 지정
 - 양수 또는 음수 가능
 - 스택 순서가 큰 요소가 낮은 요소 앞에 있음
 - z-index 지정 되지 않고 겹치면, HTML코드에서 마지막에 배치된 요소가 맨 위에 표시



예제2

```
<head>
```

```
  <style>
```

```
    img {
```

```
      position: absolute;
```

```
      left: 0px; top: 0px; z-index: -1;
```

```
    }
```

```
  </style>
```

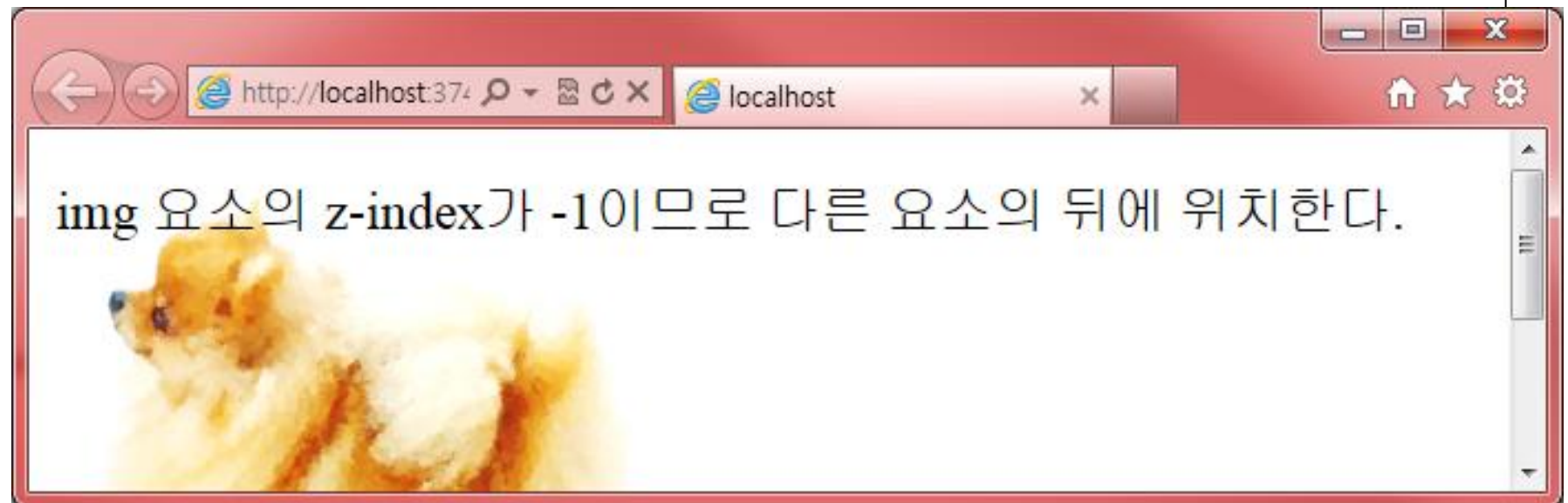
```
</head>
```

```
<body>
```

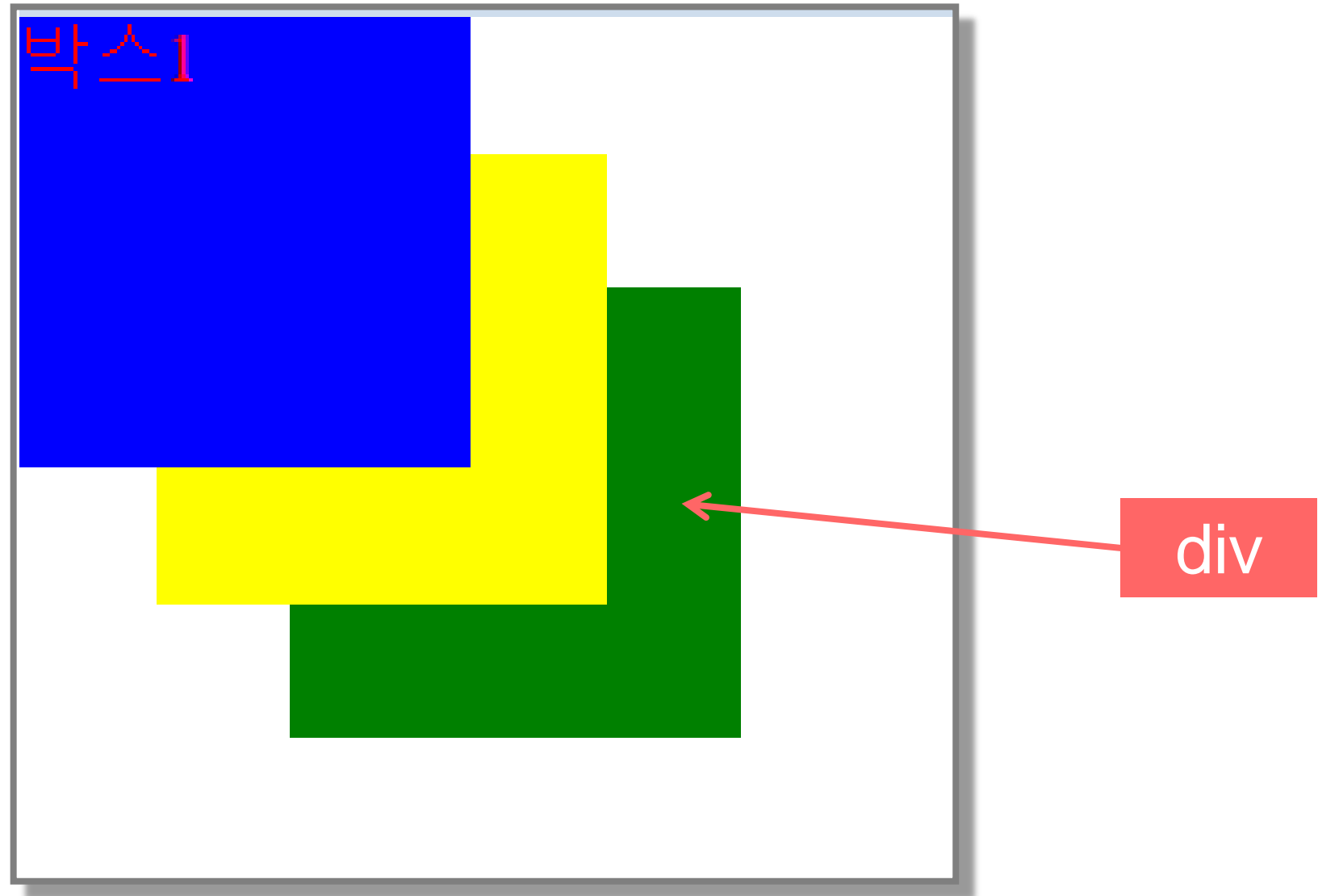
```
  
```

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

```
</body>
```



예제3 (1 / 2)



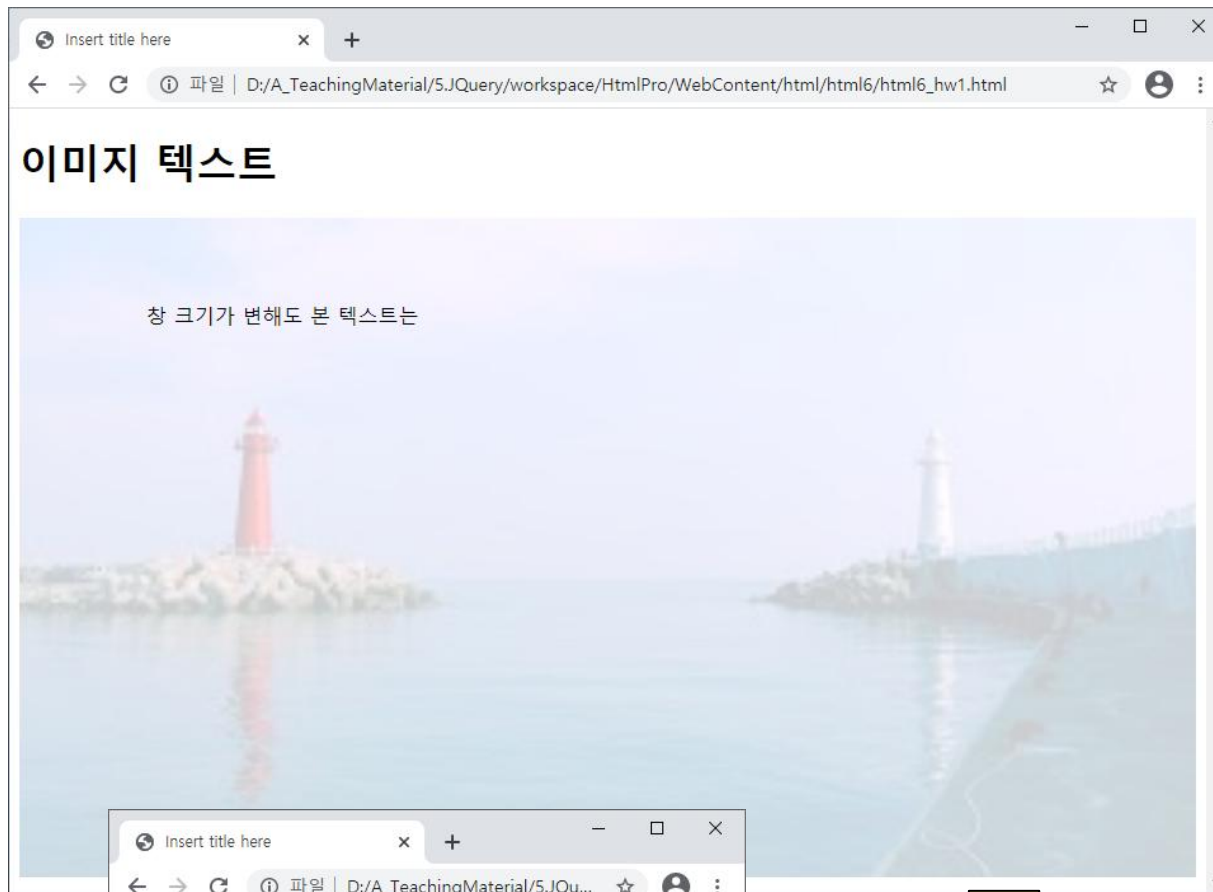
힌트

사용속성 : position, top, left, z-index

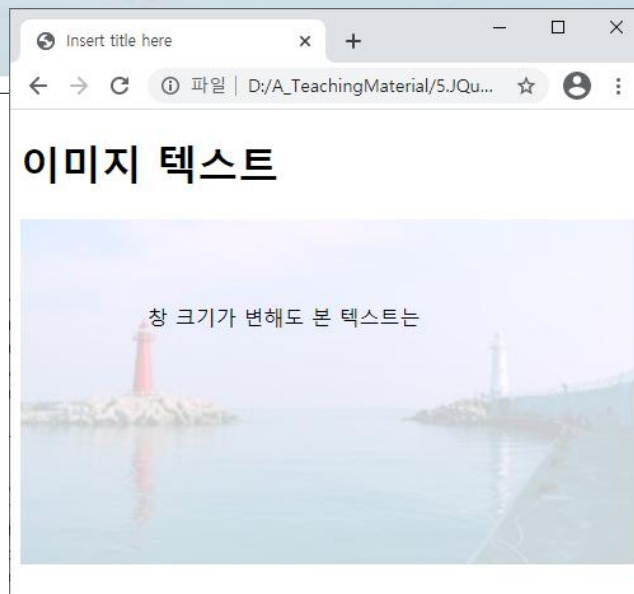
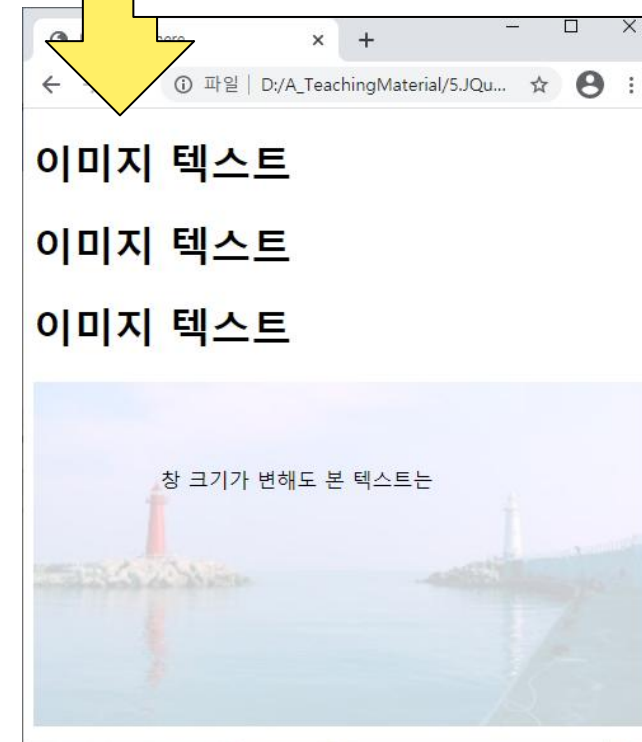
예제3 (2/2)

```
...
<style>
  div {width: 100px; height: 100px; color:red;}
  #ex1 {background: blue;
        position: absolute; top: 0px; left: 0px; z-index: 2;}
  #ex2 {background: yellow;
        position: absolute; top: 30px; left: 30px; z-index: 1;}
  #ex3 {background: green;
        position: absolute; top: 60px; left: 60px; z-index: 0;}
</style>
...
<body>
  <div id="ex1">박스1</div>
  <div id="ex2">박스2</div>
  <div id="ex3">박스3</div>
</body>
```

예제4(1/2)



이미지 영역 상단에
다른 요소가 추가되어도,
이미지 위의 글자는
이미지 위 동일 위치에 존재



브라우저 창 크기가 변경되면
이미지 크기도 변경됨

예제4(2/2)

```
<style type="text/css">
  #divContainer {
    position: relative;
  }
  img {
    width: 100%;
    height: auto;
    opacity: 0.3;
  } #divText {
    position: absolute;
    top: 50px;
    left: 100px;
  }
</style>
```

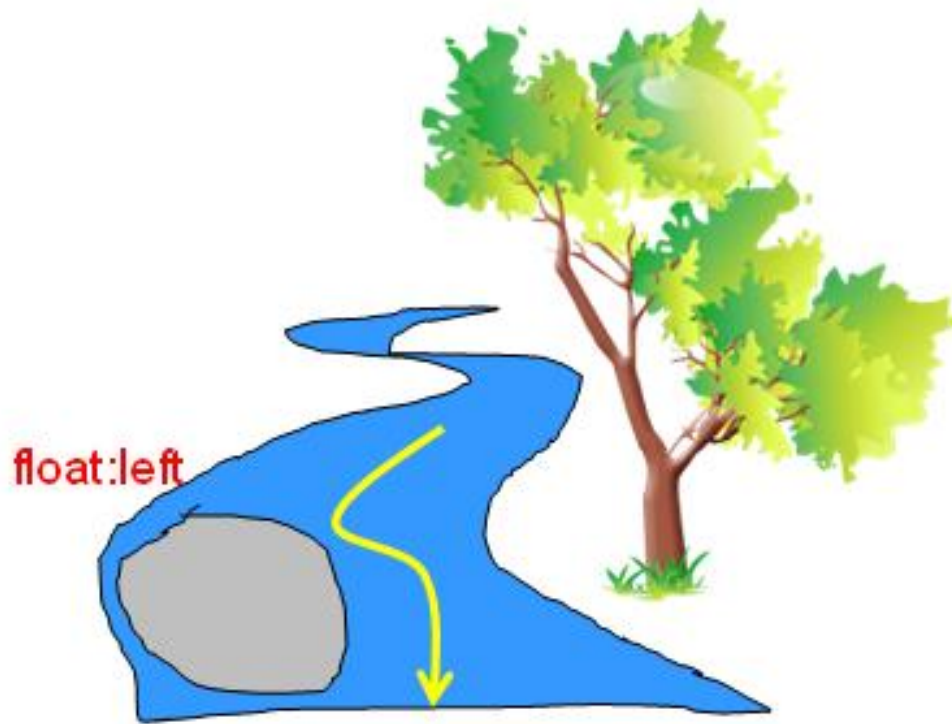
```
<body>
  <h2>Image Text</h2>
  <div id="divContainer">
    
    <div id="divText">
      창 크기가 변해도
      본 텍스트는
    </div>
  </div>
</body>
```

레이아웃의 overflow

- overflow 속성: 자식 요소가 부모 요소의 범위를 벗어났을 때, 어떻게 처리할 것인지를 지정
 - visible – default값.
 - hidden – 부모 영역을 벗어나는 부분을 보이지 않게 한다.
 - scroll – 스크롤을 만들어 진다.
 - auto – 자동으로 스크롤을 생성하거나 생성하지 않는다.
- 높이가 지정된 블록 요소에서만 작동
- overflow, overflow-x, overflow-y

float과 clear(1/5)

- 하나의 콘텐츠 주위로 다른 콘텐츠들이 물처럼 흘러가는 스타일 지정



In this example, the image will float to the right in the paragraph, and the paragraph will wrap around the image.

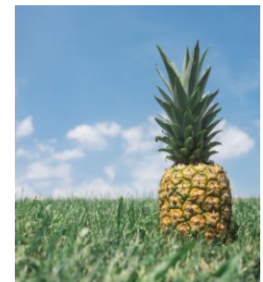
float: left;



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac. In at libero sed nunc venenatis imperdiet sed ornare turpis. Donec vitae dui eget tellus gravida venenatis. Integer fringilla congue eros non fermentum. Sed dapibus pulvinar nibh tempor porta. Cras ac leo purus. Mauris quis diam velit.

float: right;

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac...



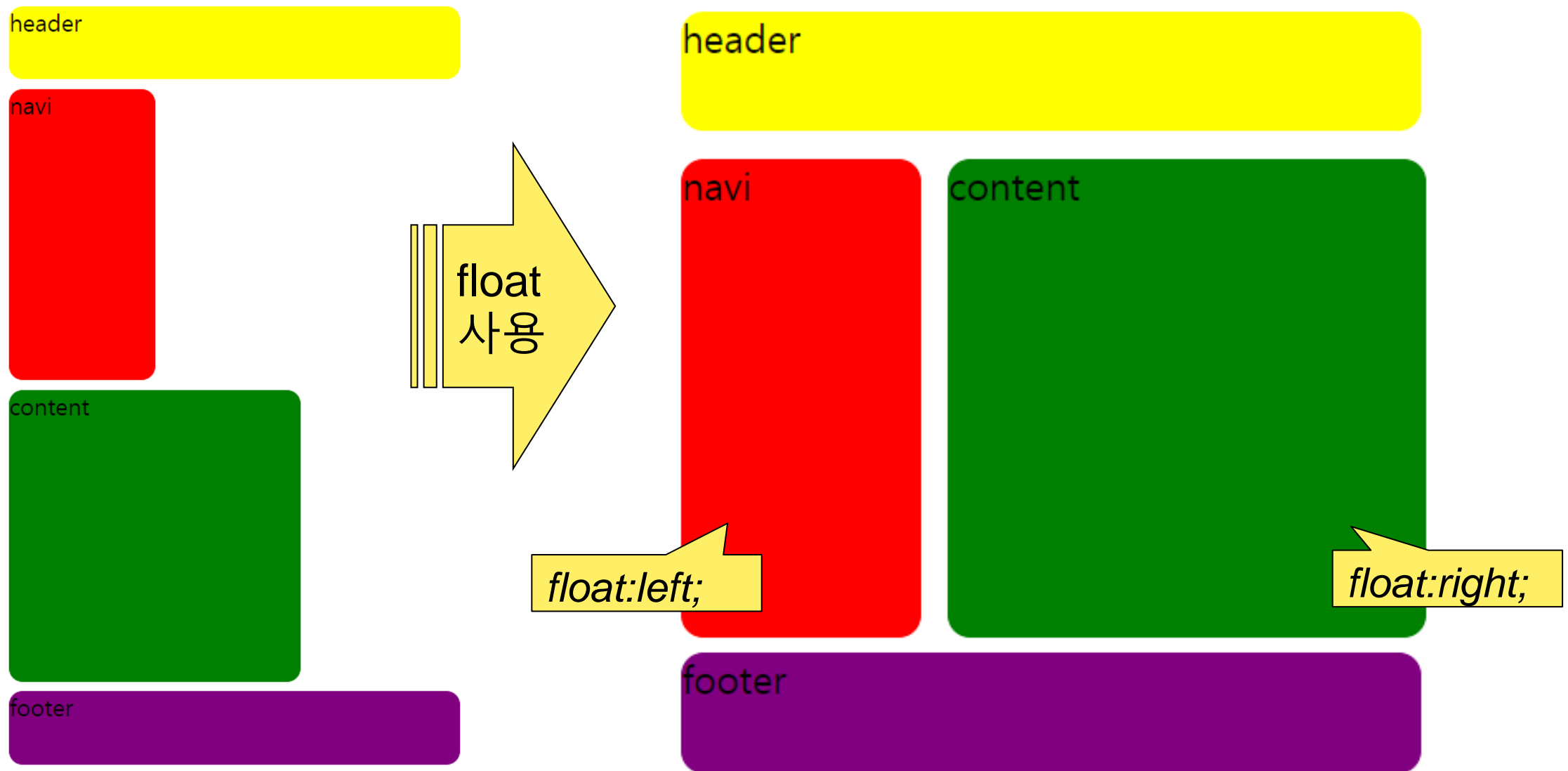
float과 clear(2/5)

```
<!DOCTYPE html>
<html>
<head>
  <style>
    img.a {float: left;}
  </style>
</head>
<body>
  
  <p>
    생활이 그대를 속일지라도
    슬퍼하거나 노여워 말라.
    ...
  </p>
</body>
</html>
```



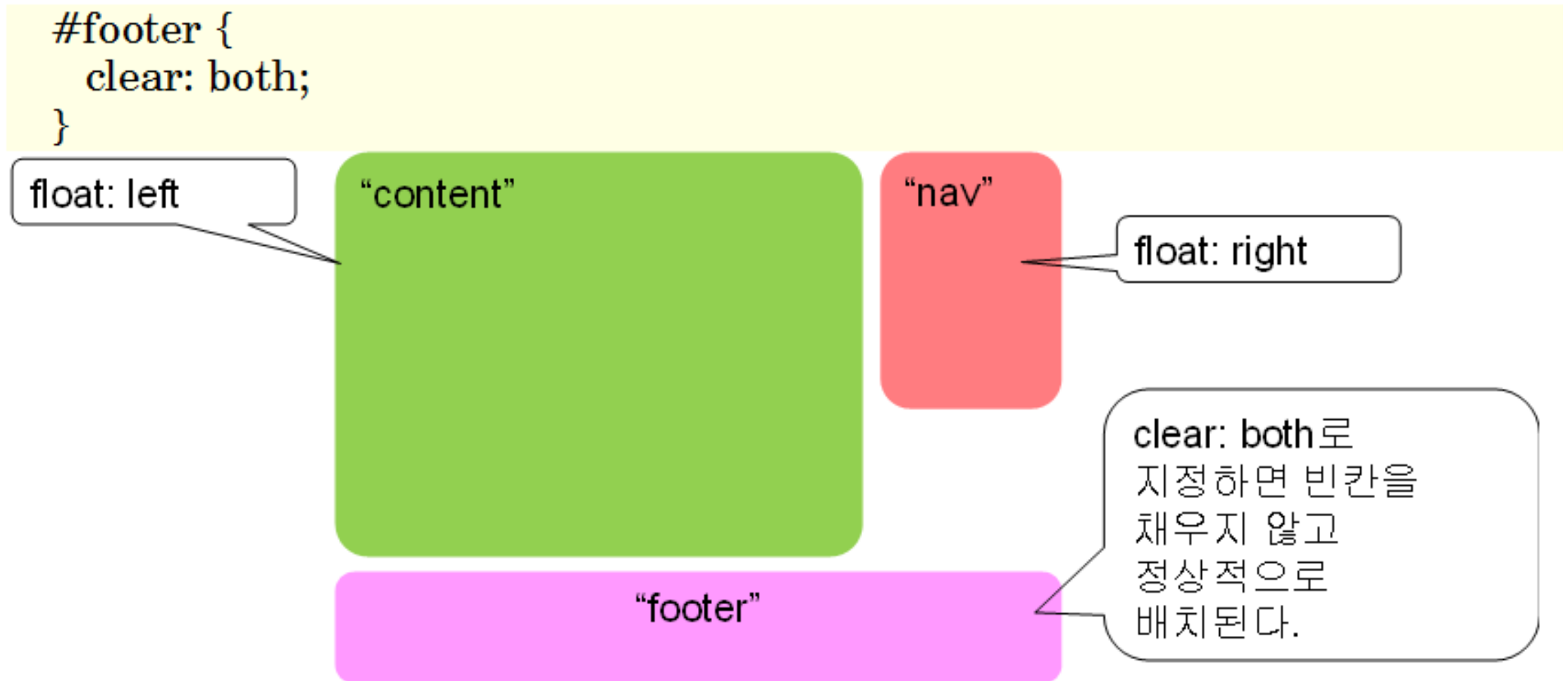
float과 clear(3/5)

- float의 용도 - 레이아웃에 많이 사용된다.



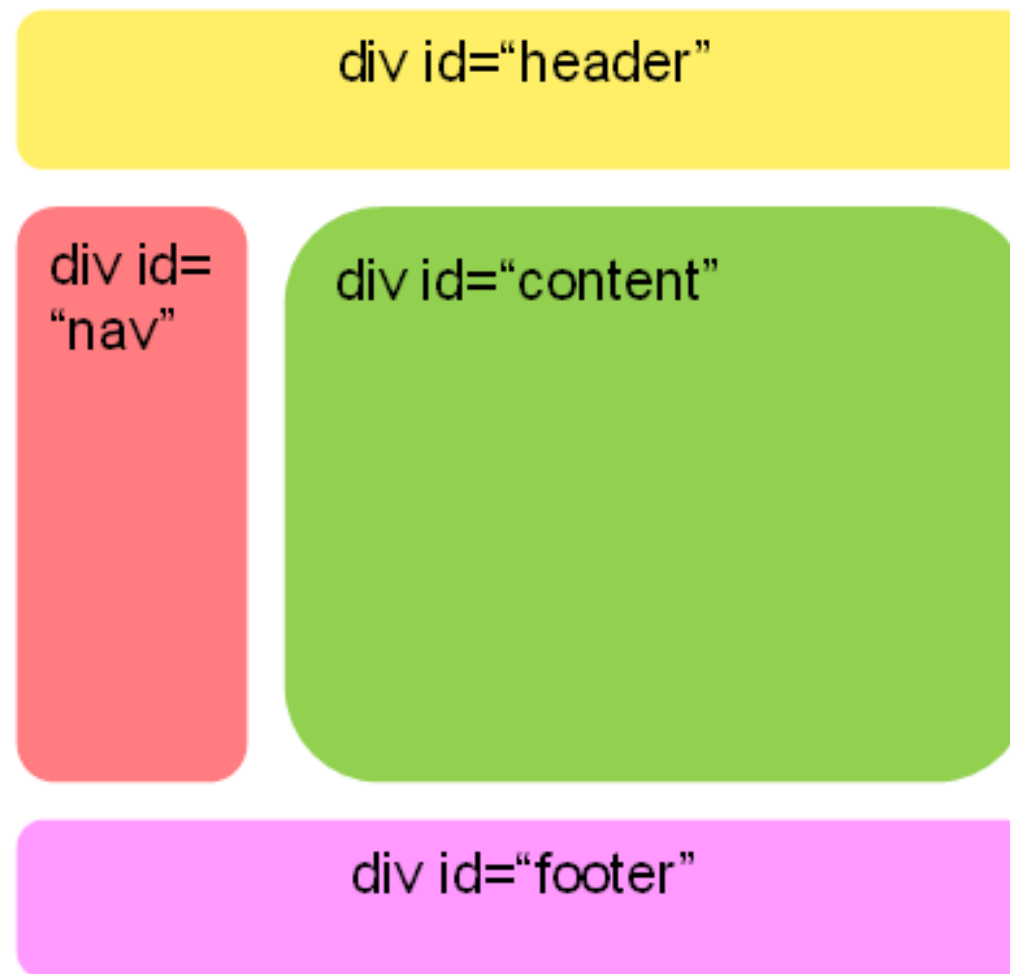
float과 clear(4/5)

- clear - float 속성을 중단할 때 사용된다.

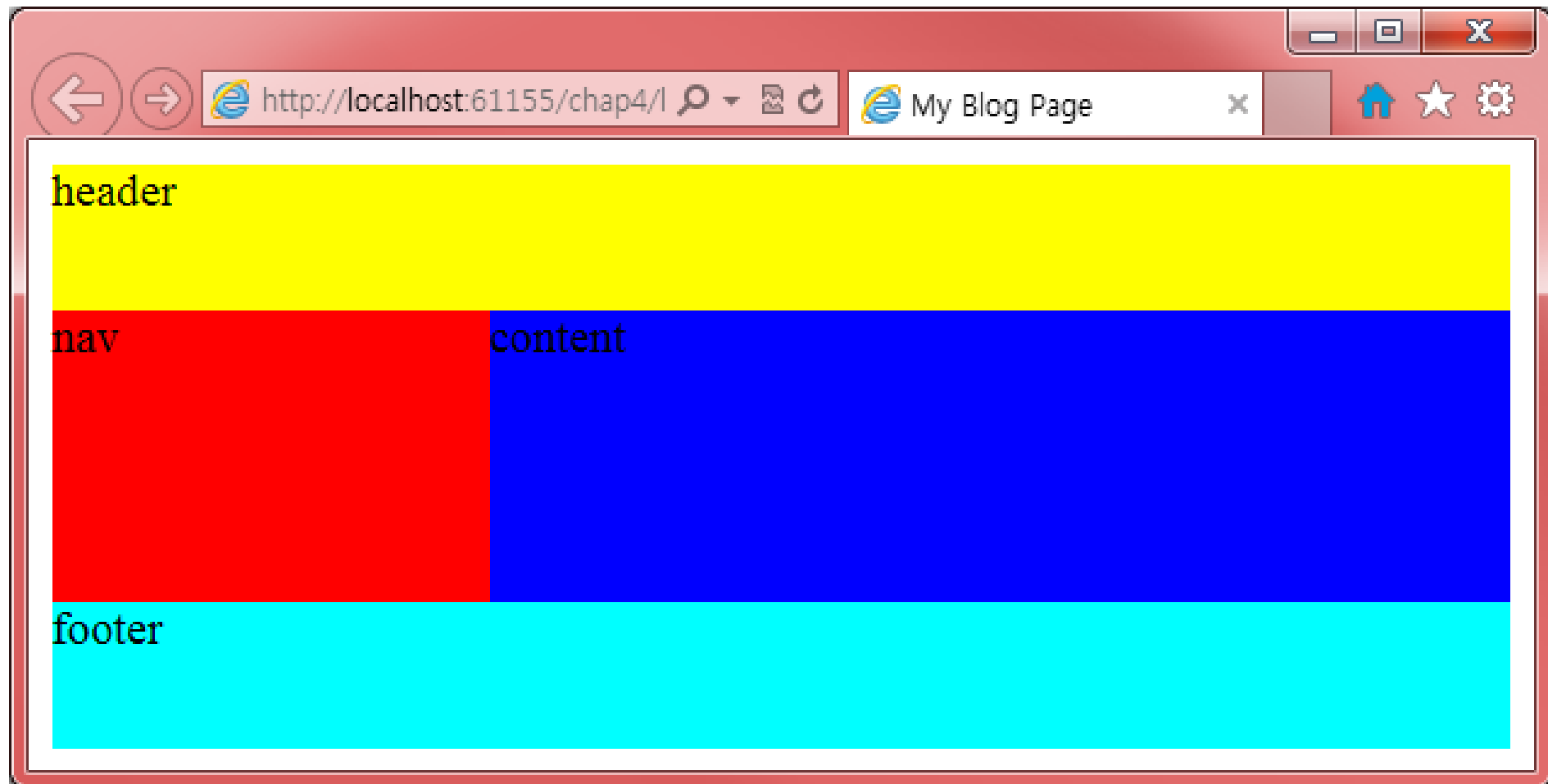


float과 clear(5/5)

- <div>를 이용한 레이아웃



예제5(1/2)



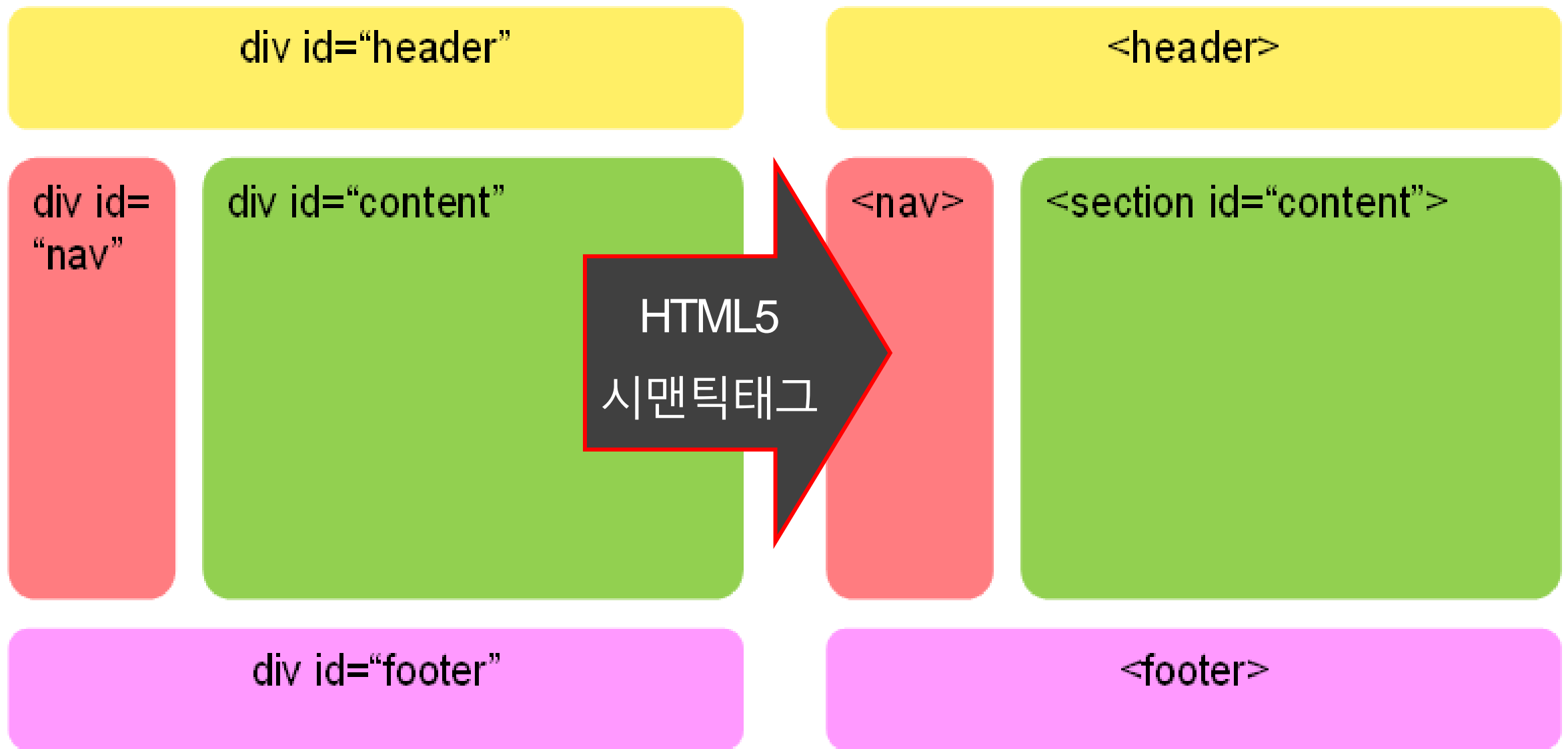
예제5 (2/2)

```
...
<style>
#header {
    background-color: yellow;
    width: 100%; height: 50px;
}
#nav {
    background-color: red;
    width: 30%; height: 100px;
    float: left;
}
#content {
    background-color: blue;
    width: 70%; height: 100px;
    float: right;
    color: white;
}
```

```
#footer {
    background-color: aqua;
    width: 100%; height: 50px;
    clear: both;
}
</style>
...
<body>
    <div id="header">header</div>
    <div id="nav">navigation</div>
    <div id="content">content</div>
    <div id="footer">footer</div>
</body>
...
```

웹 사이트 레이아웃(1/2)

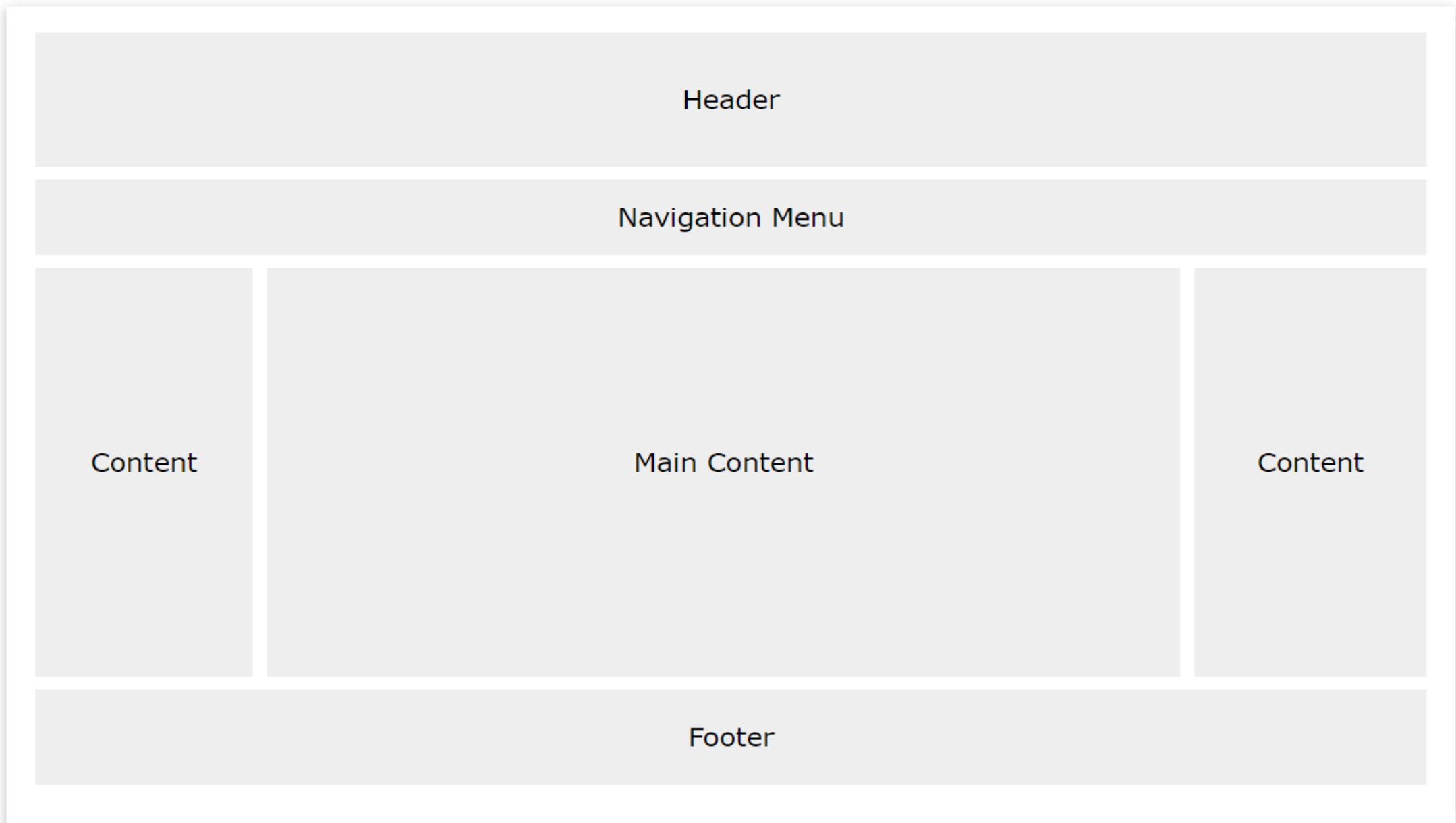
- 웹 사이트의 구조 요소 변화 – Semantic Elements 이용



*semantic element : 의미를 가진 요소

웹 사이트 레이아웃(2/2)

- 시맨틱 요소로 구성된 웹 사이트



시맨틱 요소(1/2)

- Semantic Elements : 웹페이지의 구조를 쉽게 이해할 수 있도록 정의된 태그. 단순 컨테이너 역할
 - 개발자 간 혼란 없음. 검색 엔진 등에서 소스 쉽게 판독.

태그	설명
<header>	문서의 머리말(header) 보통 홈페이지 상단의 로고나 메인 메뉴를 포함. <h1>~<h6>사이의 하나를 가지고 있어야 함.
<footer>	문서의 꼬리말(바닥글)(footer) 연락처 정보, 저작권정보 등 포함. 한 페이지에 한 번만 작성.
<section>	제목이 있는 주제별 콘텐츠 영역 <h1>~<h6>사이의 하나를 가지고 있어야 함.
<article>	웹사이트와 독립적인 콘텐츠 정의 시 사용. 블로그의 포스트, 신문기사 등 에 사용

시맨틱 요소 (2/2)

태그	설명
<code><nav></code>	내비게이션 링크. 메뉴, 목차, 인덱스 등 표현할 때 사용.
<code><aside></code>	사이드 바와 같이 옆에 위치하는 내용 외곽의 콘텐츠 표시할 때 사용. 퀵메뉴 등 표현할 때 사용.
<code><figure></code>	그림이나 도표, 사진 같은 태그를 지정할 때 묶는 태그 <code><figure></figure></code>
<code><figcaption></code>	<code><figure></code> 에 대한 제목이나 설명을 설명하는 요소 <code><figure><figcaption>...</figcaption></figure></code>

연습1

사자



사자(獅子, lion)는 식육목 고양이과 표범속으로 분류되는 포유동물이다. 현존하는 고양이과 동물 중에서 호랑이에 이어 두 번째로 큰 종이다. 사자는 흔히 **백수(百獸)의 왕**으로 불리며, 고양이과 동물 가운데 사자만이 무리를 지어 산다. 사자무리를 '프라이드'라고 하는데[1] 한 마리의 수컷이 여러 암컷들을 거느리고 다니는 습성으로 인하여 수사자는 동서고금 가리지 않고 **왕권의 상징**으로 사용되어 왔다.

호랑이



호랑이(虎狼-) 또는 범은 **고양이과에 속하는 맹수**이다. 어린 개체는 개호주라 부른다.[2] 고양이과 동물중 그 크기가 가장 크며, 재규어를 제외하면 유일하게 수영을 할 수 있다. (단, 재규어같은 경우에는 단지 사냥을 목적으로 수영을 할뿐 호랑이처럼 주기적으로 물에 들어가지는 않는다). 호랑이의 크기는 아종에 따라 다양한데, 발리호랑이는 대개 100kg 미만이며, 아무르호랑이는 수컷의 경우 최대 360kg에 이르기도 한다. 대개 황갈색 바탕에 검은색 줄무늬가 있다. 드물게 흰색을 띤 백호가 있는데, 이는 백색증과 무관한 열성유전자가 발현되어 나타나는 것이다. 뼈를 비롯한 신체의 거의 대부분의 부위가 한약재로 쓰이고 있으며, 그로인해 심각한 생존의 위협을 받고있다.

스크롤로
내려서
본문 내용
확인

힌트

사용속성 : float, overflow

연습2



연습3

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.