

웹 기초

CSS3 레이아웃

목차

- 레이아웃의 기초
- 요소 위치 정하기
- <div> 요소를 이용한 레이아웃
- 의미적 요소를 이용한 레이아웃

레이아웃이란?

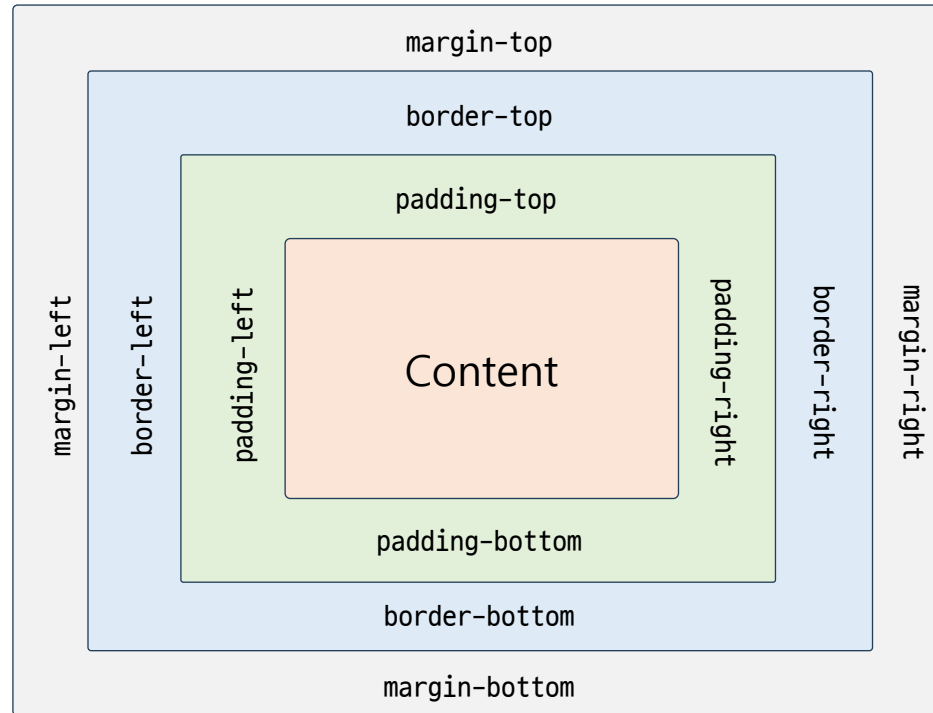
- 웹 페이지에서 HTML 요소의 위치, 크기 등을 결정하는 것
 - 집안에서의 가구 배치와 비슷하다.



박스 모델

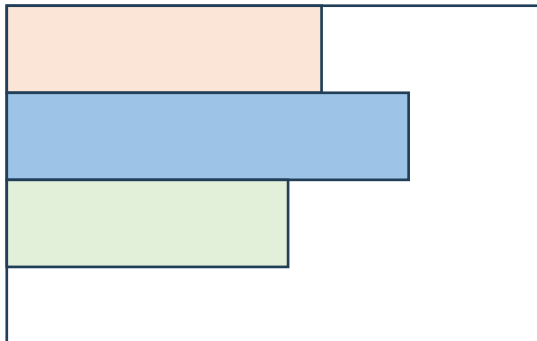
■ 박스 모델

- 웹 브라우저가 각 요소들을 화면에 그릴 때는 사각형으로 간주함
- 사각형 구성
 - padding, border, margin

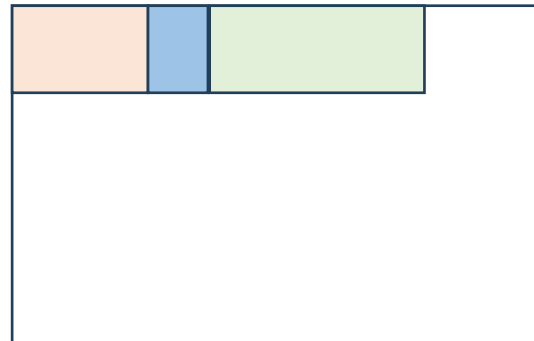


블록 요소와 인라인 요소

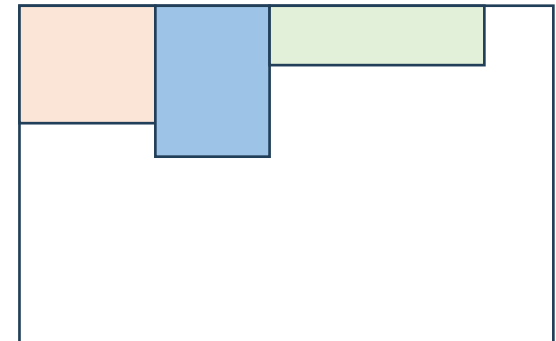
- 블록(block) 요소
 - 화면의 한 줄을 전부 차지
 - 위치와 크기 지정이 가능
- 인라인(inline) 요소
 - 한 줄에 차례대로 배치
 - 현재 줄에서 필요한 만큼의 너비만을 차지
- 인라인 블록(inline-block) 요소
 - 인라인 요소처럼 줄 바꿈 없이 한 줄에 다른 요소들과 배치
 - 블록 요소처럼 위치와 크기 지정이 가능



블록(block)



인라인(inline)

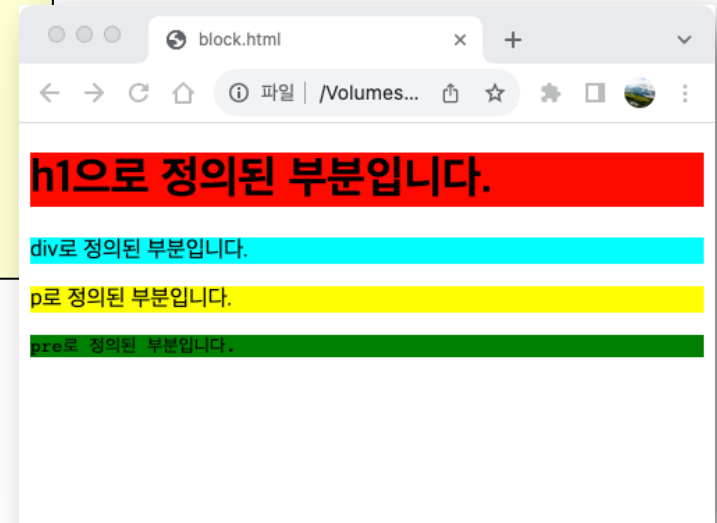


인라인 블록(inline-block)

블록 요소

- 한 줄을 전부 차지
- <h1>, <p>, , , <table>, <blockquote>, <pre>, <div> <form> , <header>, <nav> 요소
 - <nav> 태그: 네비게이션 링크
 - 다른 페이지 또는 현재 페이지의 다른 부분과 연결되는 링크들의 집합

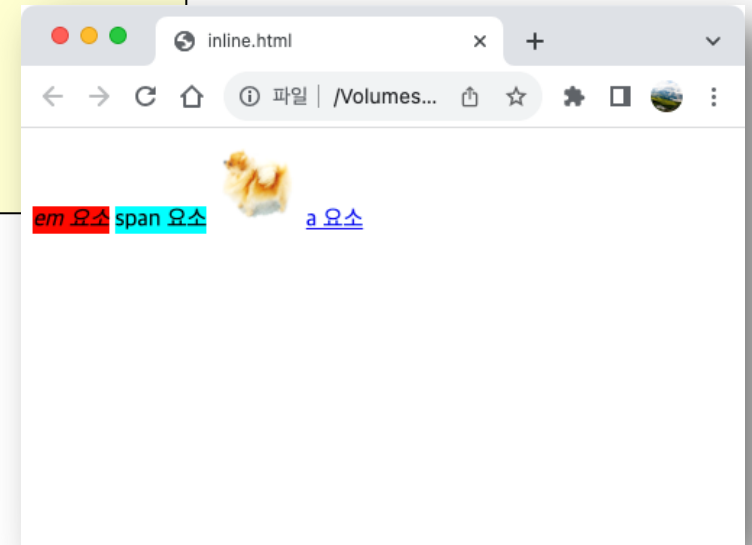
```
<!DOCTYPE html>
<html>
  <body>
    <h1 style="background-color: red">h1으로 정의된 부분입니다.</h1>
    <div style="background-color: aqua">div로 정의된 부분입니다.</div>
    <p style="background-color: yellow">p로 정의된 부분입니다.</p>
    <pre style="background-color: green">pre로 정의된 부분입니다.</pre>
  </body>
</html>
```



인라인 요소

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

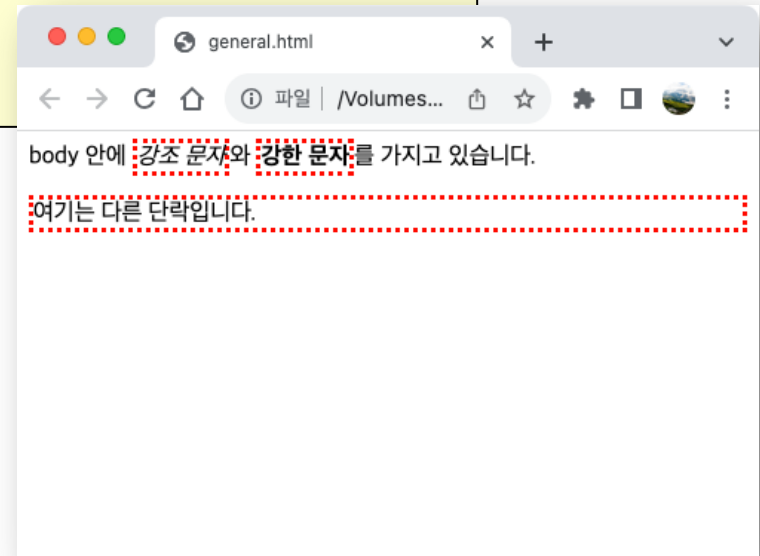
```
<!DOCTYPE html>
<html>
  <body>
    <em style="background-color: red">em 요소</em>
    <span style="background-color: aqua">span 요소</span>
    
    <a href="http://www.w3c.org">a 요소</a>
  </body>
</html>
```



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>
```

블록 요소가 나타날 때까지 인라인 요소는 같은 줄에 배치됨



CSS의 display 속성

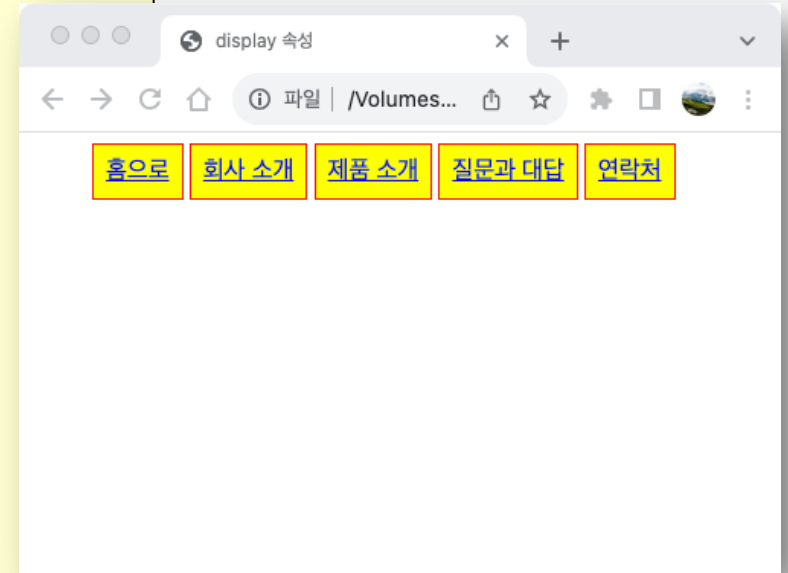
- display 속성
 - block으로 설정: 블록 요소처럼 배치
 - inline으로 설정: 인라인 요소처럼 배치
- display 속성 값
 - `display:block` : 블록(block)
 - `display:inline` : 인라인(inline)
 - `display:none` : 없는 것으로 간주됨
 - `display:hidden` : 화면에서 감춰짐

```
div {display: inline}
```

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>
```



<display.html>

href 속성

- "#": 아무 것도 실행하지 않고, 페이지 최상단으로 이동

인라인 블록 요소 예제

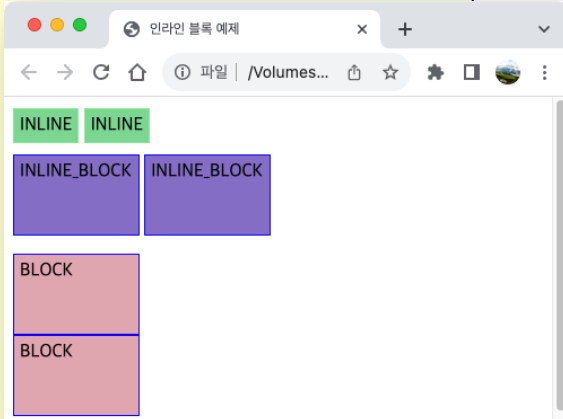
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>인라인 블록 예제</title>
  <style>
    span.a {
      display: inline;
      width: 100px;
      height: 60px;
      padding: 5px;
      border: 1px solid rgb(136, 230, 152);
      background-color: rgb(123, 214, 146);
    }

    span.b {
      display: inline-block;
      width: 100px;
      height: 60px;
      padding: 5px;
      border: 1px solid blue;
      background-color: rgb(132, 109, 196);
    }
  </style>
</head>
<body>
  <p>
    <span class="a">INLINE</span>
    <span class="a">INLINE</span>
  </p>
  <p>
    <span class="b">INLINE_BLOCK</span>
    <span class="b">INLINE_BLOCK</span>
  </p>
  <p>
    <span class="c">BLOCK</span>
    <span class="c">BLOCK</span>
  </p>
</body>
</html>
```

크기를 지정하였지만, 무시됨

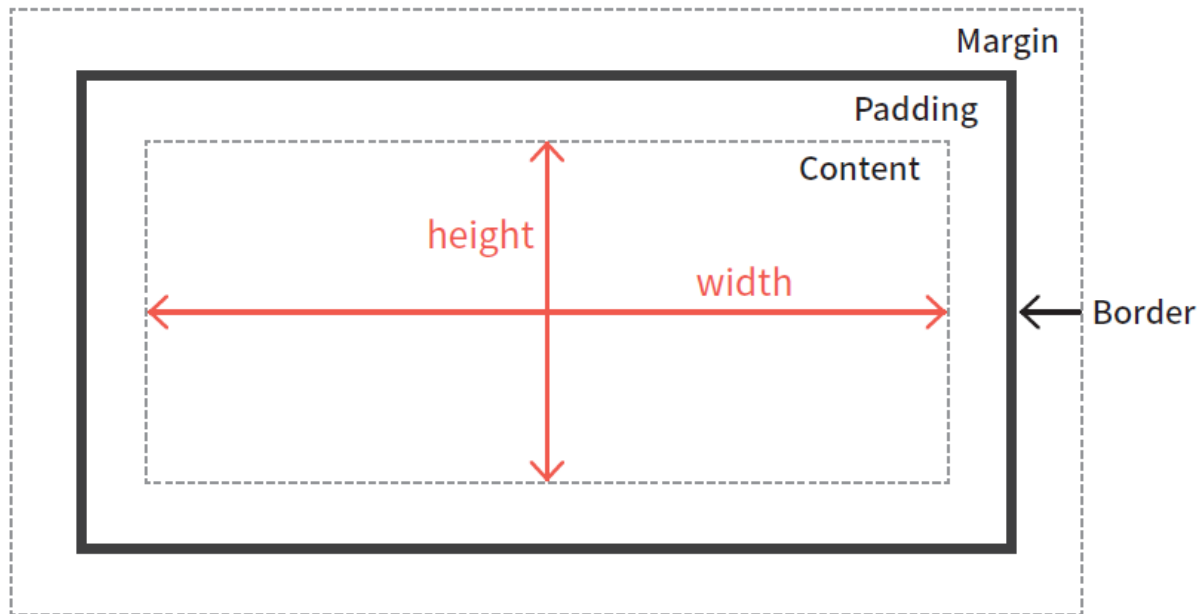
inline처럼 배치되지만, 크기 지정이 가능함

```
span.c {
  display: block;
  width: 100px;
  height: 60px;
  padding: 5px;
  border: 1px solid blue;
  background-color: rgb(224, 166, 176);
}
</style>
</head>
<body>
  <p>
    <span class="a">INLINE</span>
    <span class="a">INLINE</span>
  </p>
  <p>
    <span class="b">INLINE_BLOCK</span>
    <span class="b">INLINE_BLOCK</span>
  </p>
  <p>
    <span class="c">BLOCK</span>
    <span class="c">BLOCK</span>
  </p>
</body>
</html>
```



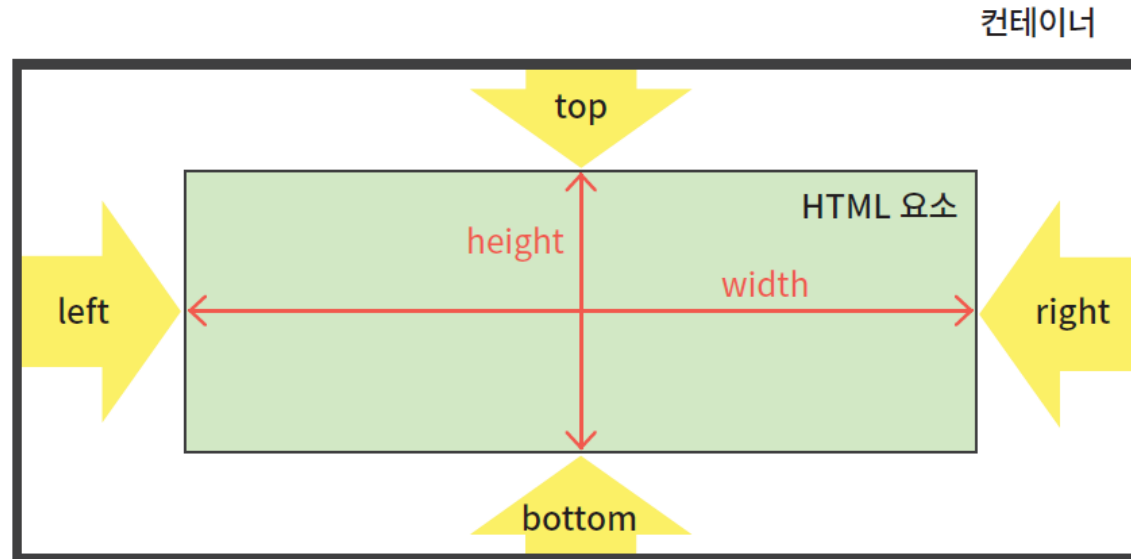
요소의 크기 설정

- 요소의 크기: width, height 속성으로 결정
 - width, height: 요소의 크기
 - min-width, min-height: 요소의 최소 크기
 - max-width, max-height: 요소의 최대 크기



요소의 위치

- top, bottom, left, right 속성으로 결정
 - 컨테이너의 경계선에서 오프셋(offset, 차이)

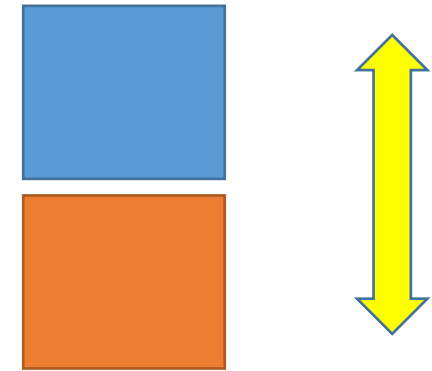


```
#target {top: 100px; left: 200px;}
```

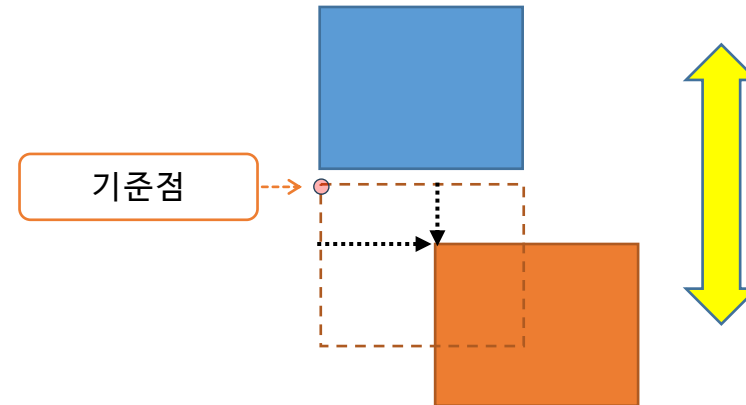
기준위치에서(100,
200)위치에 배치

기준 위치 설정 방법: position 속성

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



static positioning



relative positioning

정적 위치 설정

- 정적 위치 설정(static positioning)
 - 블록 요소들은 박스처럼 상하로 쌓이게 되고 인라인 요소들은 한 줄에 차례대로 배치
 - 정적 위치 설정을 사용하면 요소의 위치는 top, bottom, left, right 속성의 영향을 받지 않음

```
<!DOCTYPE html>
<html>
<head>
  <style>
```

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

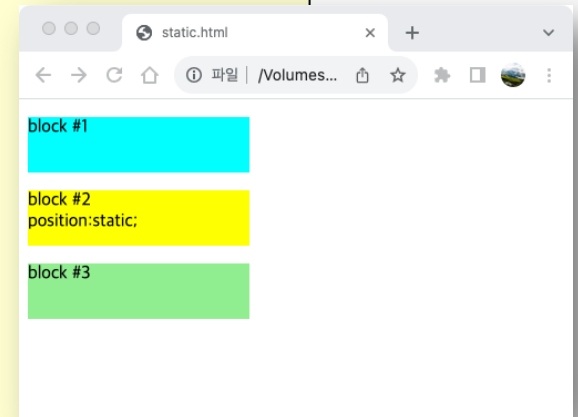
```
    #two {
      position: static;
      background-color: yellow;
      width: 200px;
      height: 50px;
    }
```

문서의 정상적 흐름에 따라 배치,
static으로 선언하지 않은 것과
결과는 같음

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

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

```
<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>
```



<static.html>

상대 위치 설정

■ 상대 위치 설정(relative positioning)

- 정상적인 위치에서 상대적으로 요소가 배치
- 상대 위치로 설정된 요소는 다른 요소 위에 겹쳐질 수 있음

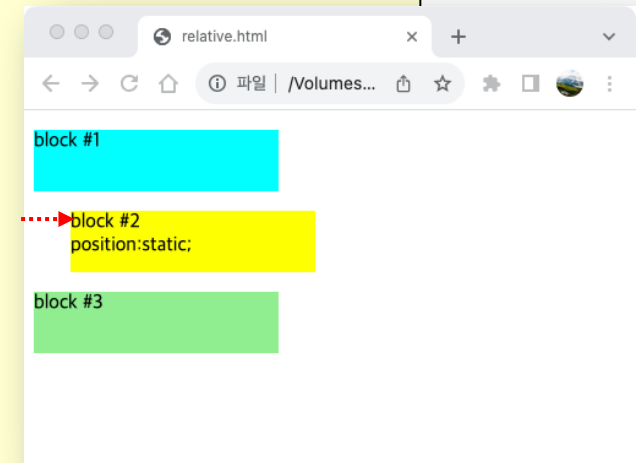
```
<!DOCTYPE html>
<html>
<head>
  <style>

    #one {
      background-color: cyan;
      width: 200px;
      height: 50px;
    }
    #two {
      position: relative;
      left: 30px;
      background-color: yellow;
      width: 200px;
      height: 50px;
    }
  }
```

정상적인 위치에서
왼쪽으로 30 픽셀

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

<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>
```

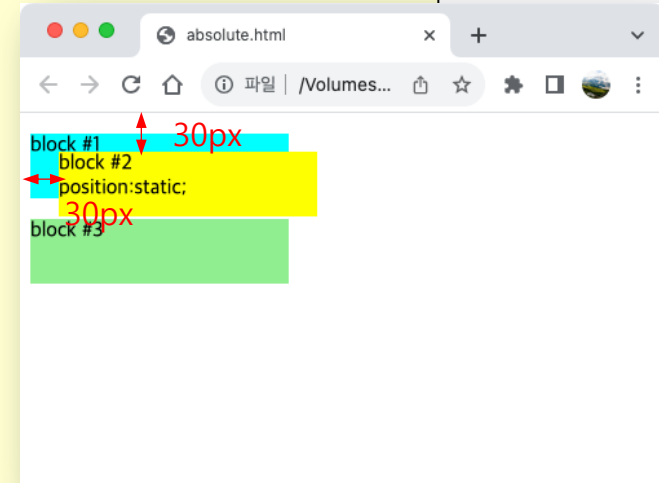


절대 위치 설정

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

```
<!DOCTYPE html>
<html>
<head>
  <style>
    #one {
      background-color: cyan;
      width: 200px;
      height: 50px;
    }
    #two {
      position: absolute;
      top: 30px;
      left: 30px;
      background-color: yellow;
      width: 200px;
      height: 50px;
    }
  </style>
</head>
<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>
```

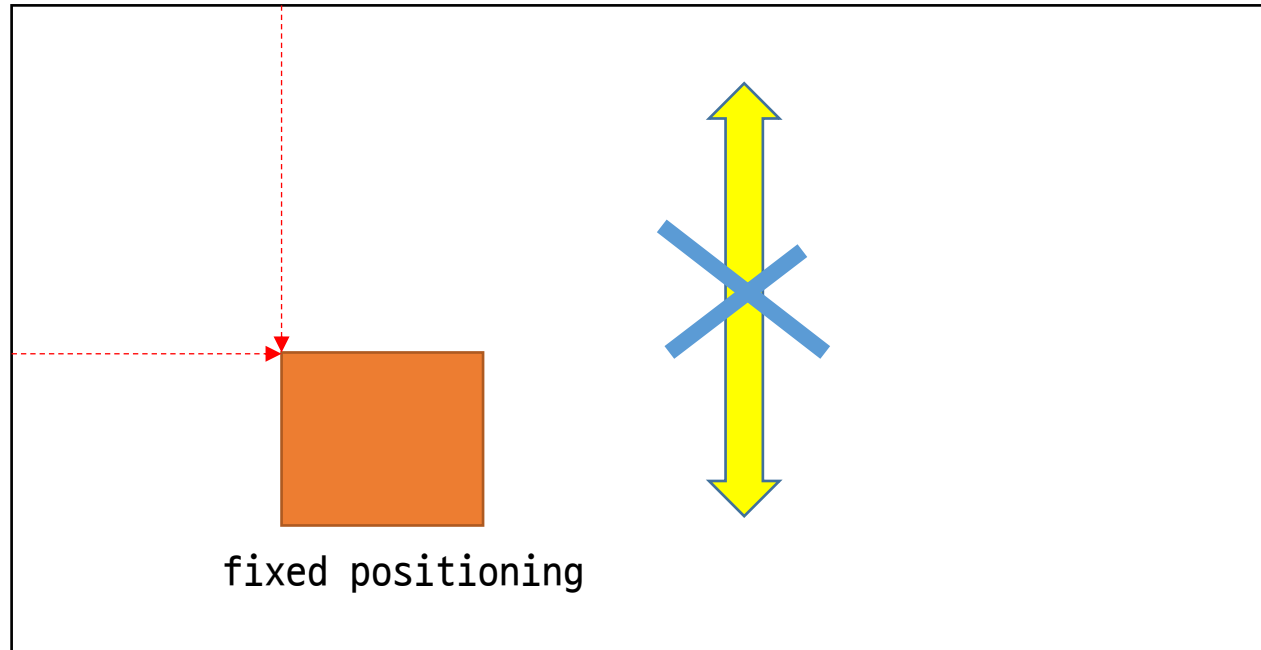
```
#three {
  background-color: lightgreen;
  width: 200px;
  height: 50px;
}
</style>
</head>
<body>
  <p id="one">block #1</p>
  <div id="two">
    block #2<br />
    position:static;<br />
  </div>
  <p id="three">block #3</p>
</body>
</html>
```



고정 위치 설정

- 고정 위치 설정(fixed positioning)

- 브라우저 윈도우에 상대적으로 요소의 위치를 잡는 것
- 윈도우가 스크롤 되어도 화면에서 요소는 움직이지 않음: 동일한 위치에 표시됨



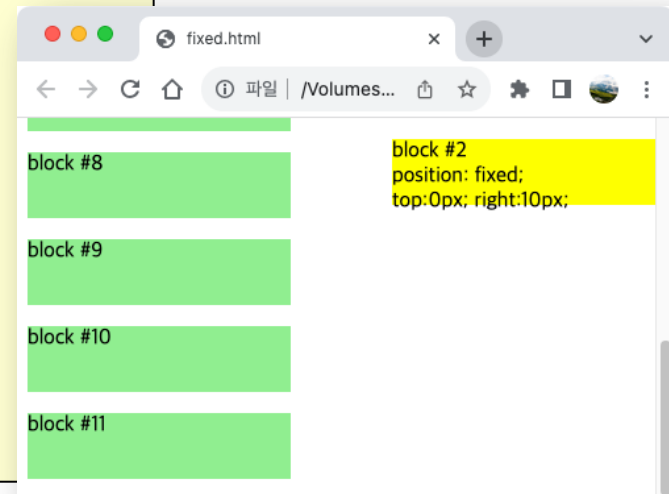
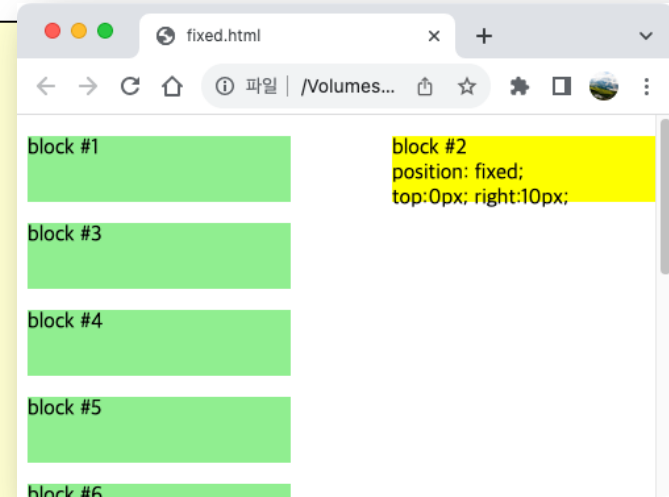
고정 위치 설정 예제

```
<!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>
```

```
<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>
```



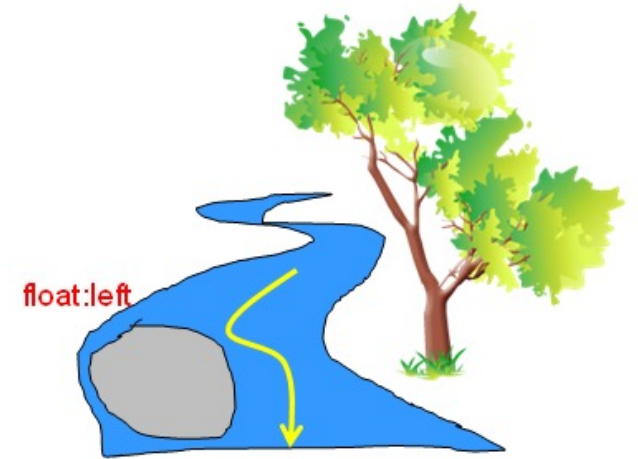
float 속성

■ float 속성

- 하나의 콘텐츠 주위로 다른 콘텐츠들이 물처럼 흘러가는 스타일 지정
- 요소를 컨테이너의 왼쪽이나 오른쪽으로만 배치 가능함
 - `float: left`, `float: right`

■ float 속성을 사용하는 경우

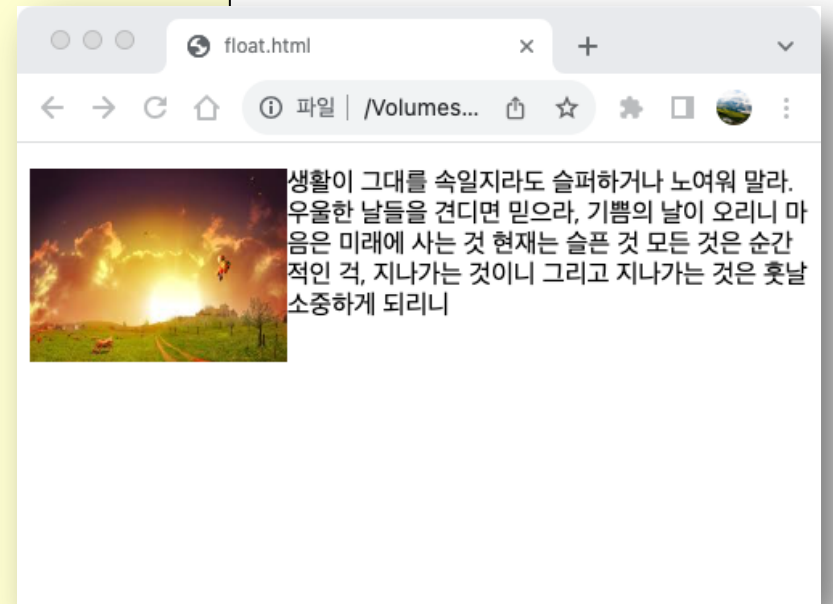
- 이미지와 텍스트를 단순히 배치
 - 이미지는 인라인 요소이고 문단은 블록 요소여서 아래와 같이 배치



float 예제

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

<body>
  
  <p>
    생활이 그대를 속일지라도
    슬퍼하거나 노여워 말라.
    우울한 날들을 견디면
    믿으라, 기쁨의 날이 오리니
    마음은 미래에 사는 것
    현재는 슬픈 것
    모든 것은 순간적인 격, 지나가는 것이니
    그리고 지나가는 것은 훗날 소중하게 되리니
  </p>
</body>
</html>
```



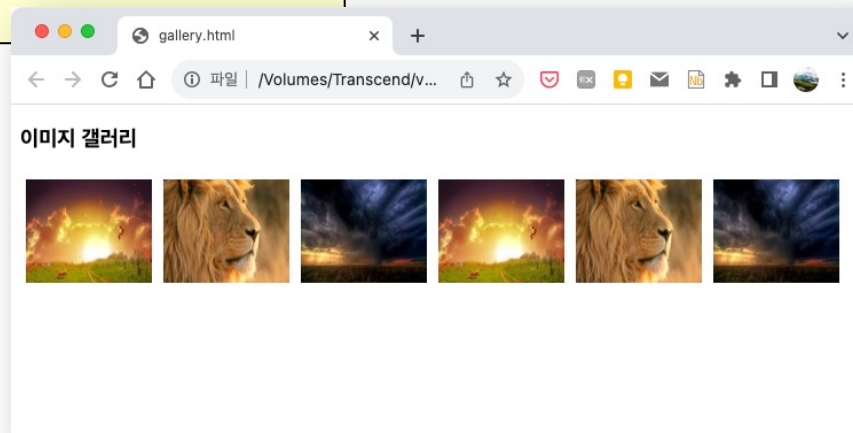
float를 여러 번 적용

- float left를 여러 번 적용하면
 - 화면에 여유가 있다면 이전 요소에 이어서 두 번째 요소가 왼쪽에 배치됨

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

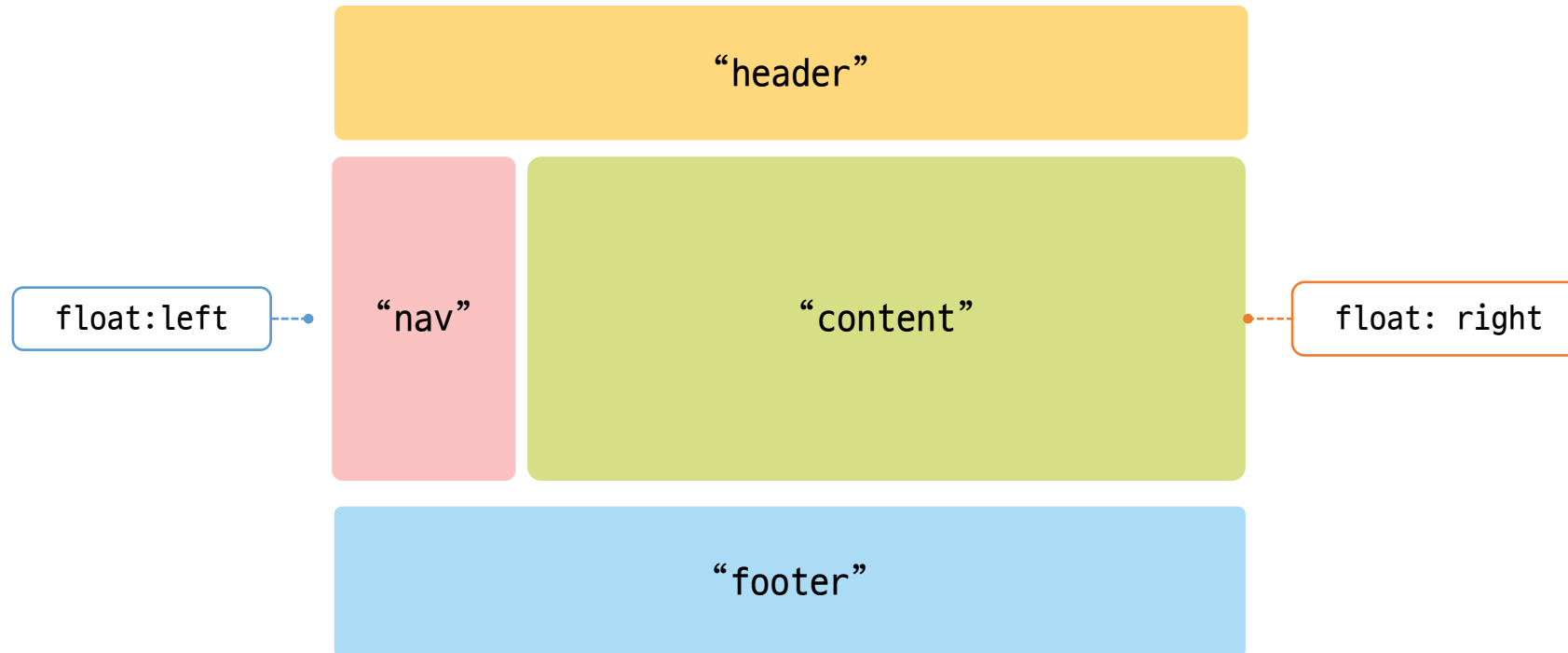
모든 이미지에
float:left 적용

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



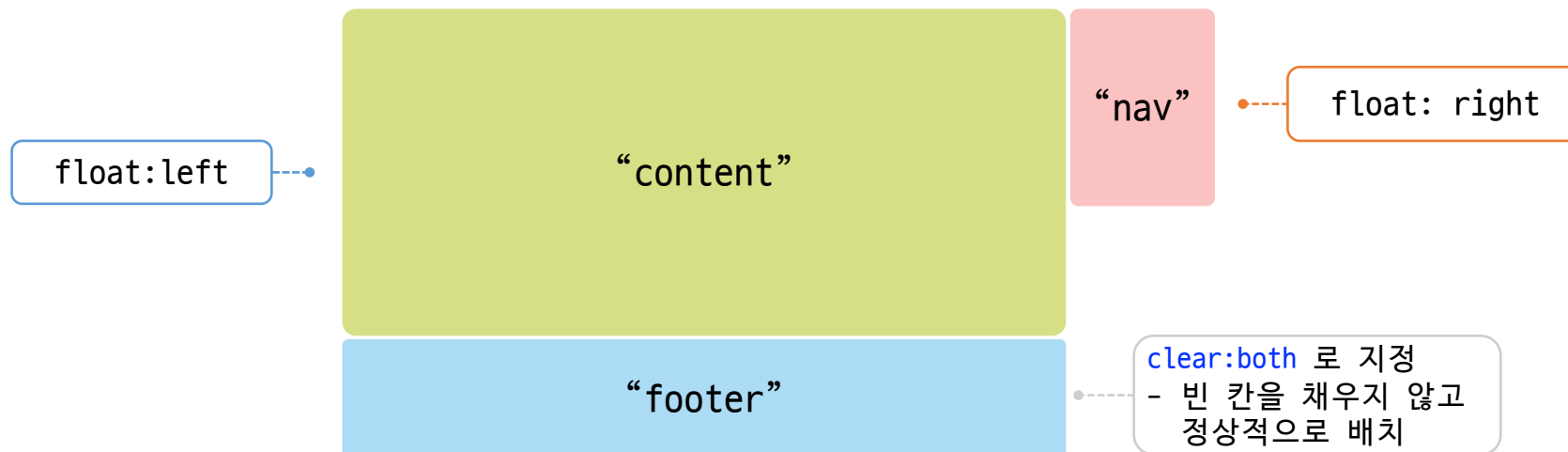
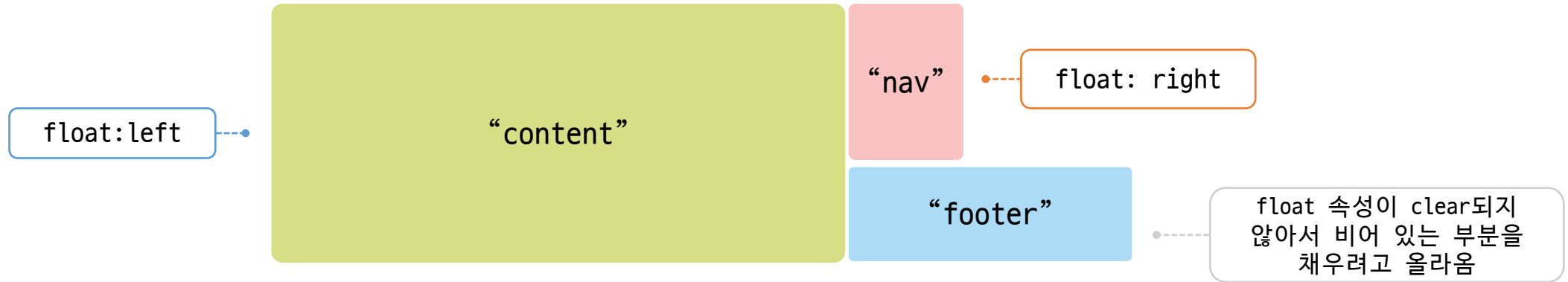
float 용도

- 웹 페이지의 레이아웃을 잡을 때 사용됨
 - nav와 content는 한 줄에 같이 배치



clear 속성

- float 속성을 중단할 때 사용된다.

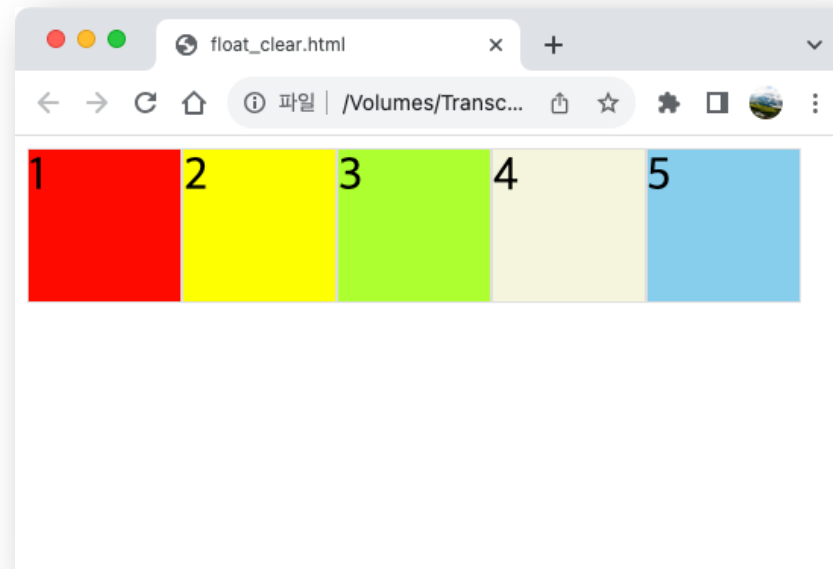


float: left 적용

```
<!DOCTYPE html>
<html>
<head>
  <style>
    .box {
      width: 100px;
      height: 100px;
      border: 1px solid #dedede;
      font-size: 30px;
    }
    .float { float: left; }
    .right { float: right; }
    .no1 { background-color: red; }
    .no2 { background-color: yellow; }
    .no3 { background-color: greenyellow; }
    .no4 { background-color: beige; }
    .no5 { background-color: skyblue; }
    .clear { clear: both; }
  </style>
</head>
```

<https://goddino.tistory.com/66>

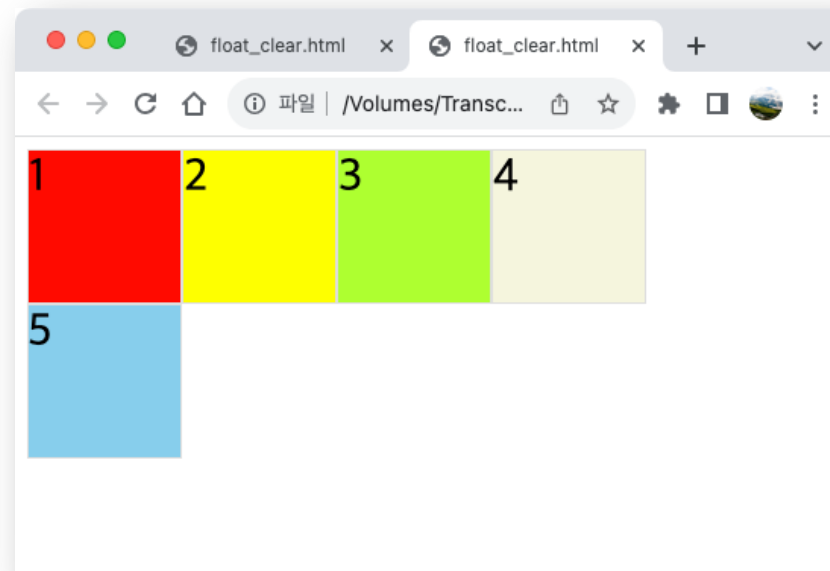
```
<body>
  <div>
    <div class="box no1 float">1</div>
    <div class="box no2 float">2</div>
    <div class="box no3 float">3</div>
    <div class="box no4 float">4</div>
    <div class="box no5 float">5</div>
  </div>
</body>
</html>
```



clear:both 적용

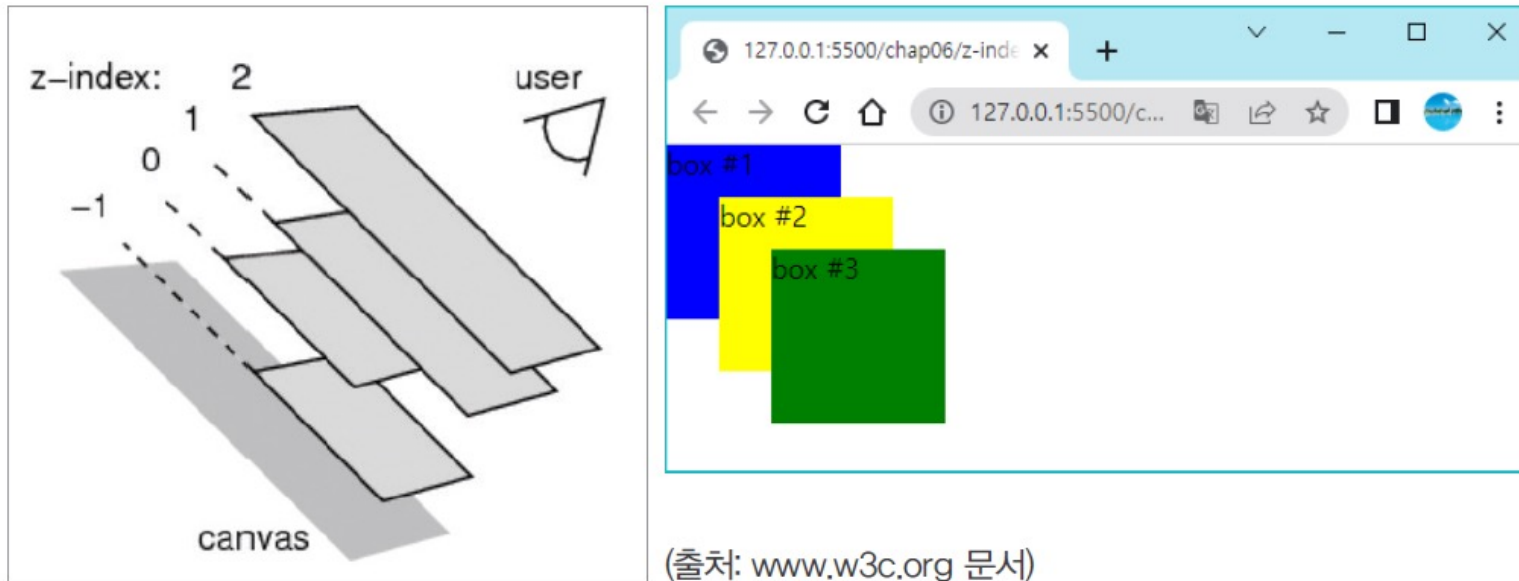
```
<!DOCTYPE html>
<html>
<head>
  <style>
    .box {
      width: 100px;
      height: 100px;
      border: 1px solid #dedede;
      font-size: 30px;
    }
    .float { float: left; }
    .right { float: right; }
    .no1 { background-color: red; }
    .no2 { background-color: yellow; }
    .no3 { background-color: greenyellow; }
    .no4 { background-color: beige; }
    .no5 { background-color: skyblue; }
    .clear { clear: both; }
  </style>
</head>
```

```
<body>
  <div>
    <div class="box no1 float">1</div>
    <div class="box no2 float">2</div>
    <div class="box no3 float">3</div>
    <div class="box no4 float">4</div>
    <div class="box no5 clear">5</div>
  </div>
</body>
</html>
```



z-index

- 요소의 스택 순서를 지정
 - 요소가 다른 요소의 앞에 위치하거나, 뒤에 위치하느냐를 지정



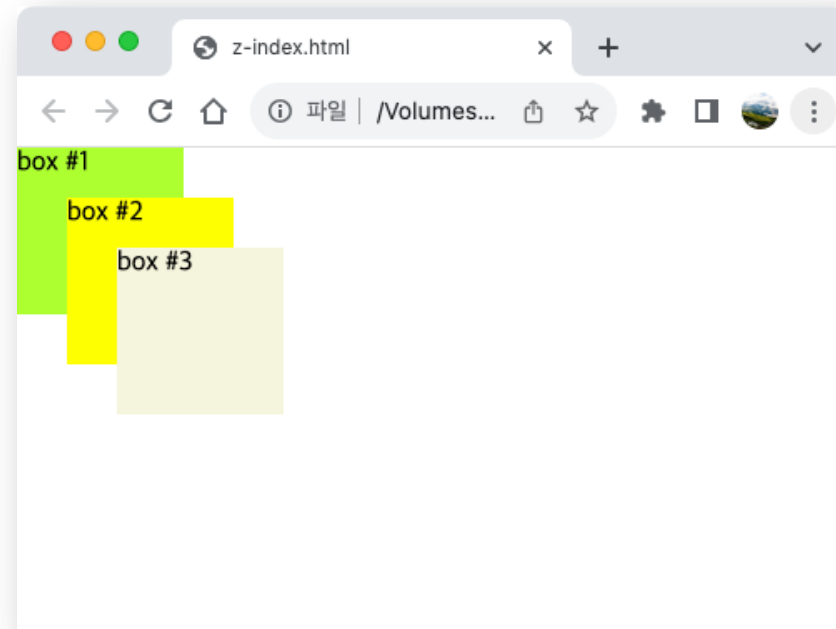
스태킹 순서 예제 (z-index 사용 안함)

```
<!DOCTYPE html>
<html>
<head>
  <style>
    #box1 {
      position: absolute;
      top: 0px; left: 0px;
      width: 100px; height: 100px;
      background: greenyellow;
    }

    #box2 {
      position: absolute;
      top: 30px; left: 30px;
      width: 100px; height: 100px;
      background: yellow;
    }

    #box3 {
      position: absolute;
      top: 60px; left: 60px;
      width: 100px; height: 100px;
      background: beige;
    }
  </style>
</head>
```

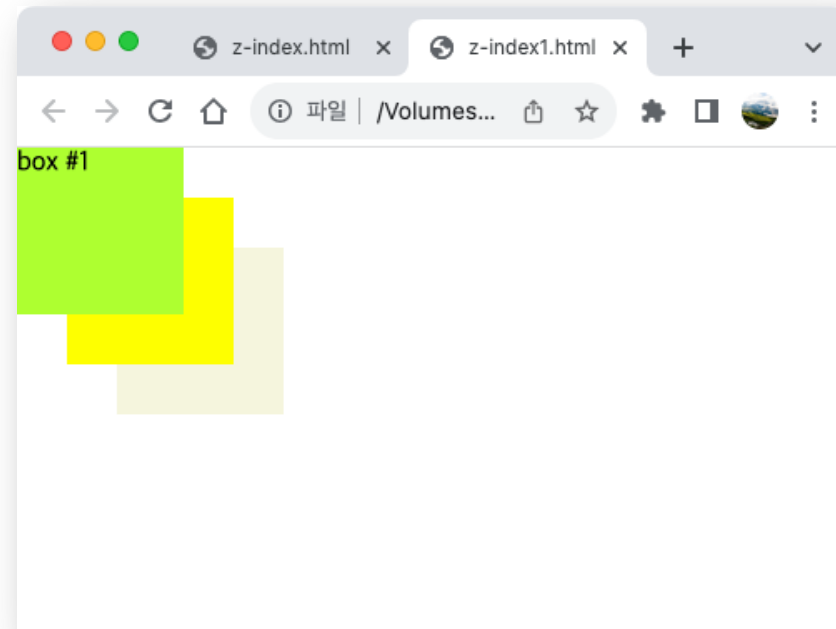
```
<body>
  <div id="box1">box #1 </div>
  <div id="box2">box #2 </div>
  <div id="box3">box #3 </div>
</body>
</html>
```



z-index 사용 (z-index1.html)

```
<!DOCTYPE html>
<html>
<head>
  <style>
    #box1 {
      position: absolute;
      top: 0px; left: 0px;
      width: 100px; height: 100px;
      background: greenyellow;
      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: beige;
      z-index: 0;
    }
  </style>
</head>
```

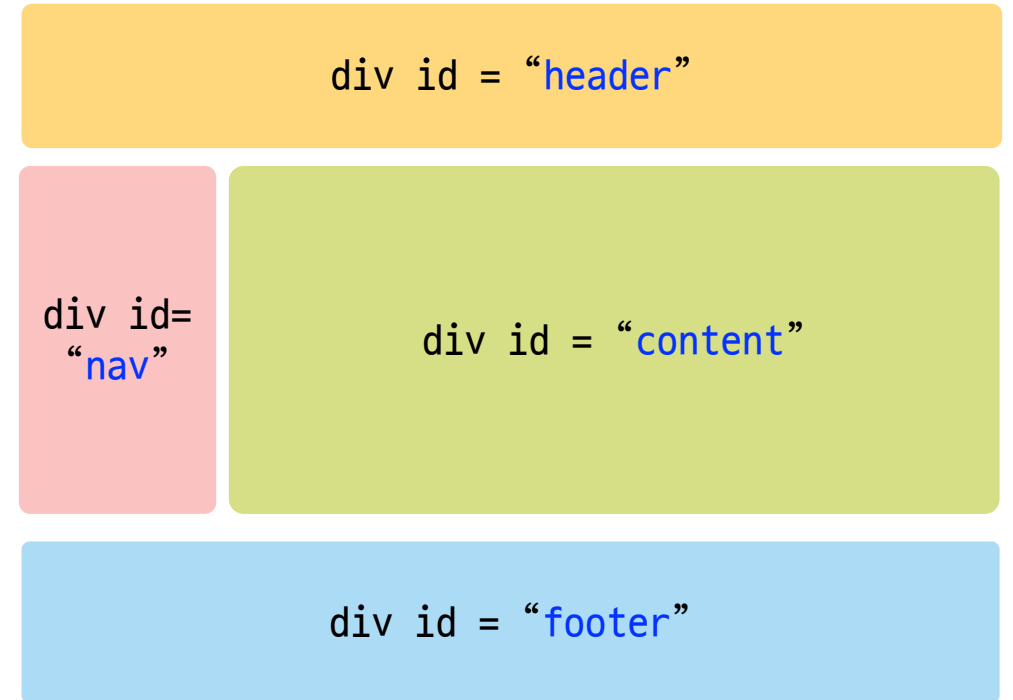
```
<body>
  <div id="box1">box #1 </div>
  <div id="box2">box #2 </div>
  <div id="box3">box #3 </div>
</body>
</html>
```



z-index 값이 클수록 앞에 놓여짐
position 속성이 absolute나 fixed로 설정되어야 함

<div> 요소를 이용한 레이아웃

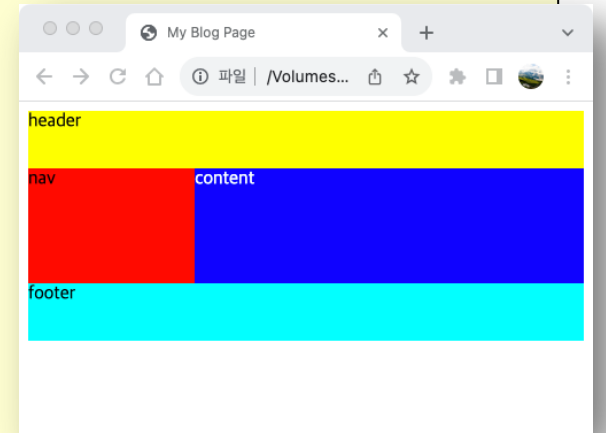
- 레이아웃 만드는 방법
 - <table> 사용한 레이아웃
 - <div>를 사용한 레이아웃
 - <nav>, <section>, <aside>등의 시멘틱 요소를 이용한 레이아웃
- 시멘틱(semantic) 요소
 - semantic 사전적 의미
 - 의미의, 의미론적인
 - HTML5에서 semantic 의미
 - 그 자체에 의미를 가지고 있는 요소
 - 시멘틱 요소를 사용하는 이유
 - 사용된 의미를 명확히 전달하기 위함
 - 요소 이름을 통해 HTML 구조 파악이 쉬워짐



예제

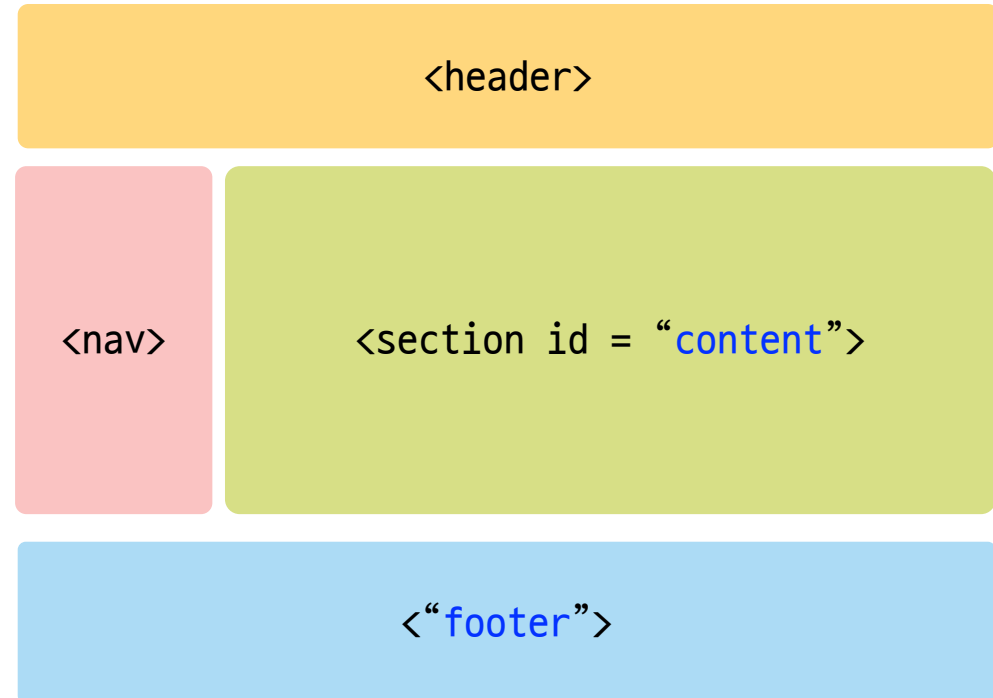
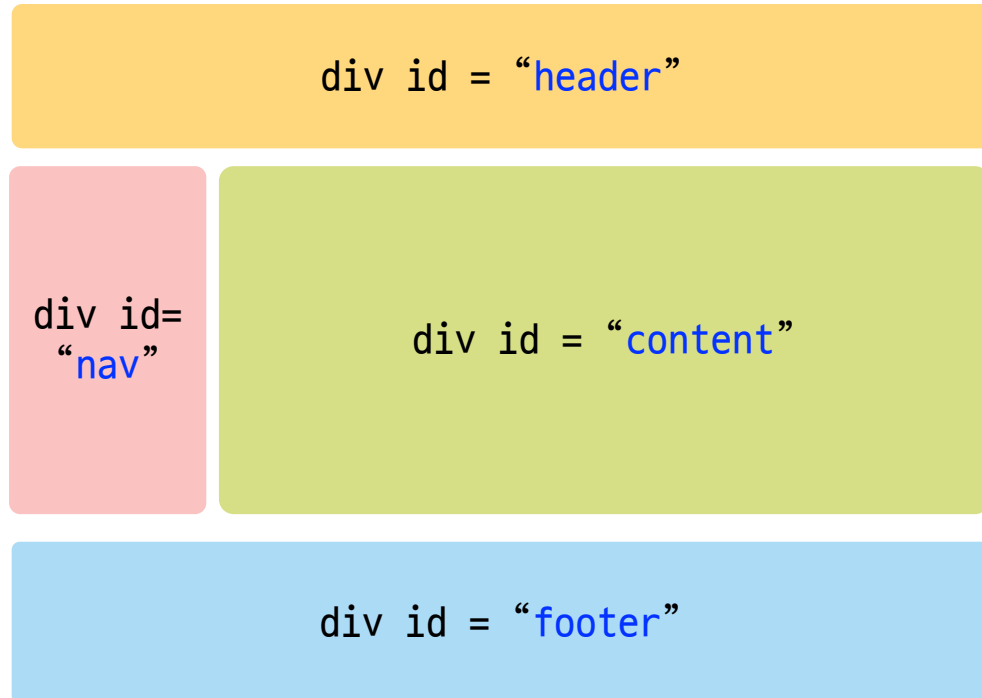
```
<!DOCTYPE html>
<html>
<head>
  <title>My Blog Page</title>
  <style>
    #header {
      background-color: yellow;
      width: 100%;
      height: 50px;
    }
    #nav {
      width: 30%;
      background-color: red;
      height: 100px;
      float: left;
    }
    #content {
      width: 70%;
      background-color: blue;
      color: white;
      float: right;
      height: 100px;
    }
  </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>
```

```
#header {
  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>
```



의미적 요소를 이용한 레이아웃

- 의미적 요소(semantic elements)
 - 브라우저에게 **요소의 의미나 목적을 명확하게 알리는 요소**
 - <div> 자체는 아무 의미가 없음

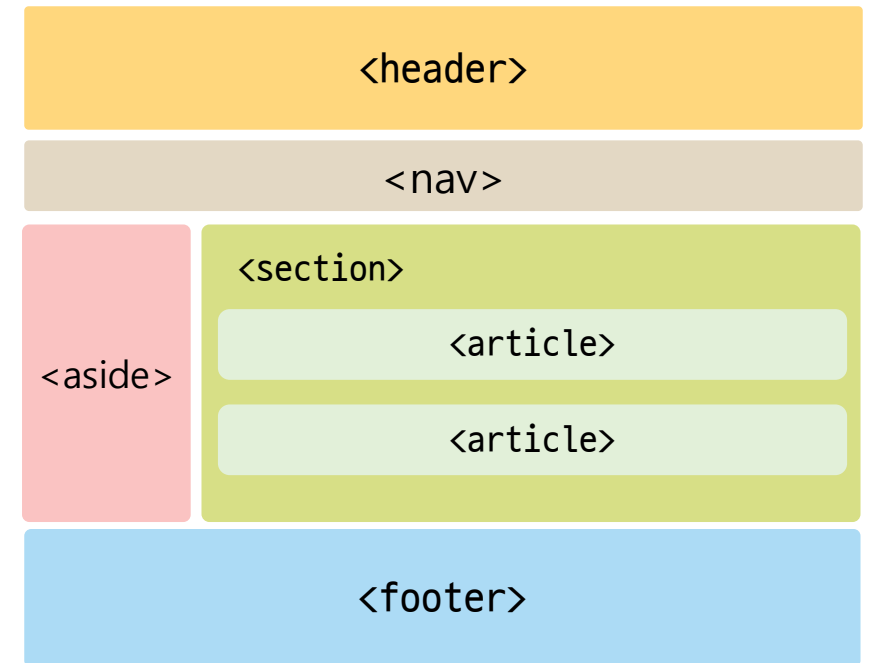


시맨틱 요소

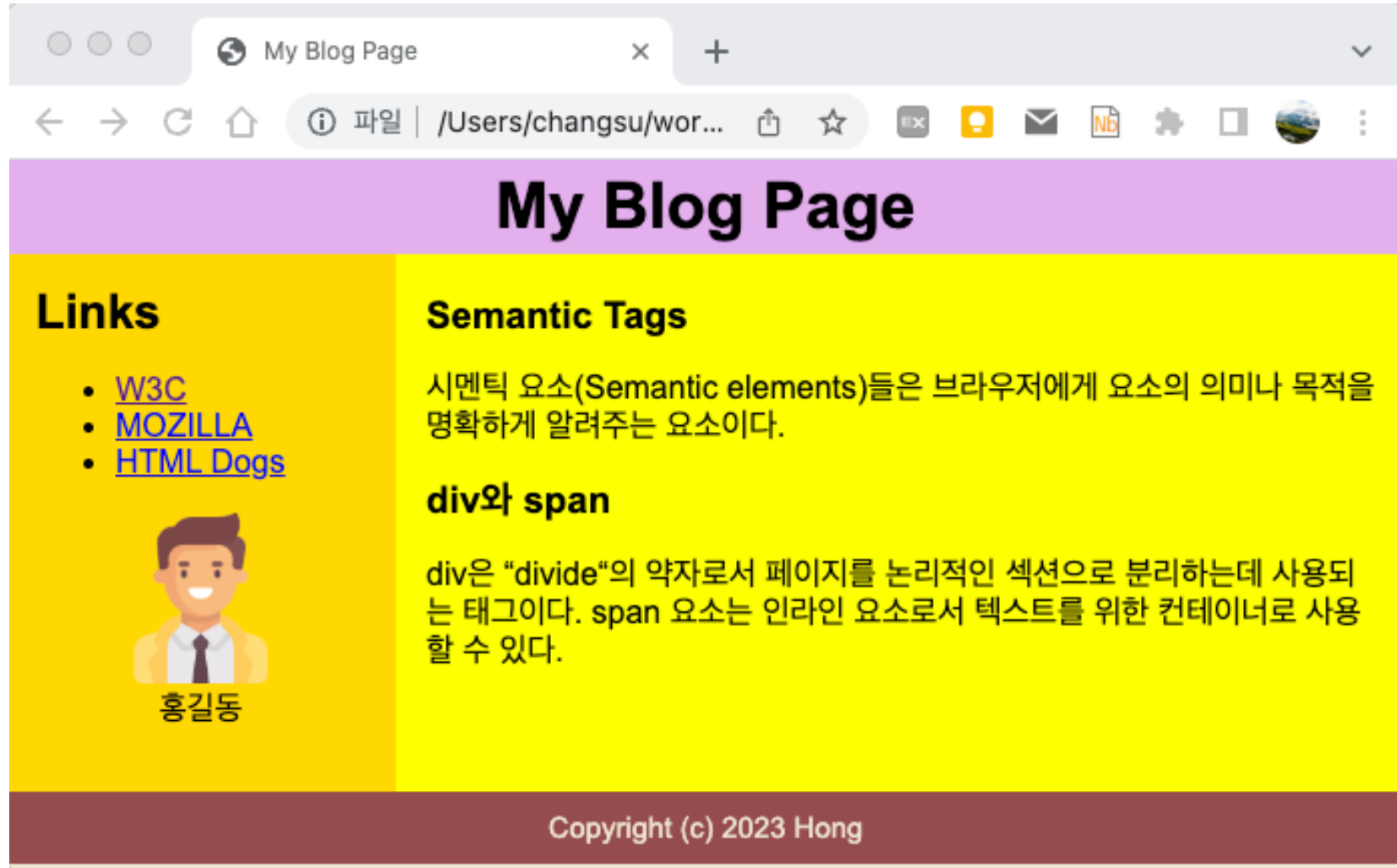
■ 시맨틱 요소

- W3C에서 많은 웹 페이지를 분석하여 개발자들이 많이 사용하고 있는 id와 class 이름을 추출
 - 이것을 표준 태그로 만들었음

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



시멘틱 요소 예제



시맨틱 요소 예제: My Blog Page #1

```
<!DOCTYPE html>
<html>
<head>
  <title>My Blog Page</title>
  <style>
    body {
      background-color: #efe5d0;
      font-family: Arial, "Trebuchet MS", sans-serif;
      margin: 0px;
    }

    header {
      background-color: #e3afed;
      color: #000000;
      margin: 0px;
      text-align: center;
      padding: 5px;
    }

    h1 {
      margin: 0px;
    }
  </style>
</head>
```

```
    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>
</head>
```

시맨틱 요소 예제: My Blog Page #2

```
<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 align="center">홍길동</figcaption>
    </figure>
  </nav>
```

시맨틱 요소 예제: My Blog Page #3

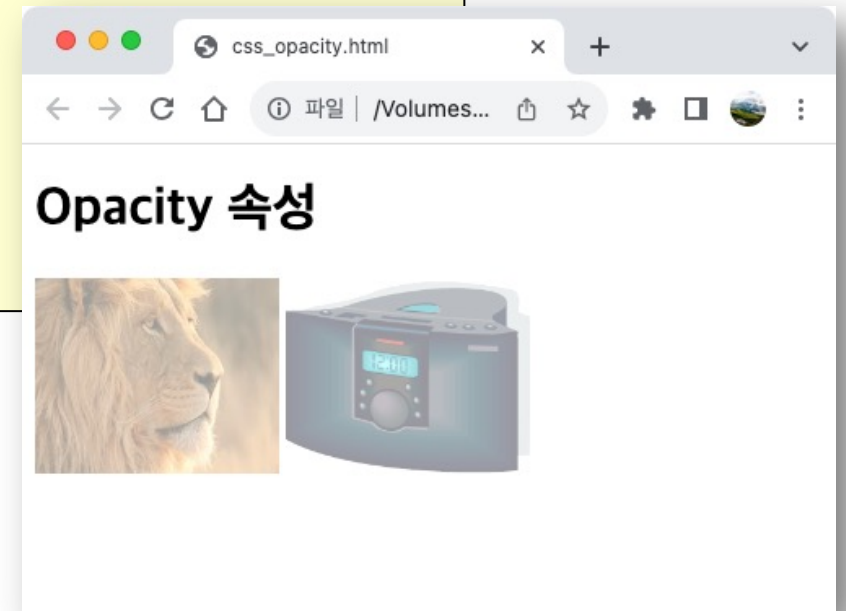
```
<section id="main">
  <article>
    <h1>Semantic Tags</h1>
    <p>
      시맨틱 요소(Semantic elements)들은 브라우저에게 요소의 의미나
      목적을 명확하게 알려주는 요소이다.
    </p>
  </article>
  <article>
    <h1>div와 span</h1>
    <p>
      div은 “divide“의 약자로서 페이지를 논리적인 섹션으로 분리하는데
      사용되는 태그이다. span 요소는 인라인 요소로서 텍스트를 위한
      컨테이너로 사용할 수 있다.
    </p>
  </article>
</section>
<footer>Copyright (c) 2023 Hong</footer>
</body>
</html>
```

효과: 투명도(opacity 속성)

```
<!DOCTYPE html>
<html>
<head>
  <style>
    img { opacity: 0.4; }
    img:hover { opacity: 1.0; }
  </style>
</head>

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

opacity 속성
- 실수 0.0 ~ 1.0의 값



가시성(visibility)

```
<!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>
```

요소를 숨김

