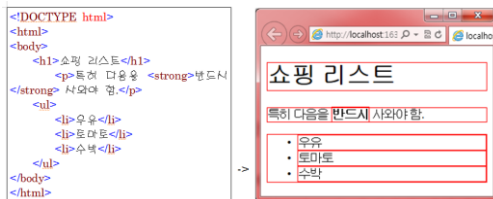


## CSS 박스모델과 응용

### Box Model

- HTML element는 여러 겹의 상자로 둘러 쌓여 있다고 가정하는 것이 css의 box model
- HTML 요소들을 박스(사각형) 형태로 그리는 것
- 박스는 배치, 색상, 경계 등의 속성을 가짐

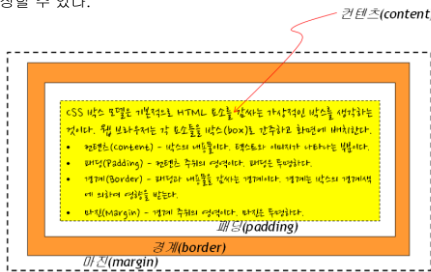


### ...Cont'd

- CSS 레이아웃을 잡을 때, 가장 기본이 되는 박스 모델
- 웹 브라우저는 각 HTML 요소를 사각형으로 간주하고 웹 페이지 작성하며 요소를 배치할 때도 사각형으로 간주
- 이와 같이 각 요소를 박스(사각형) 형태로 그리는 것을 box model이라 함
- css는 각 box의 특징(배치, 색상, 경계 등)을 결정하게 됨
- 즉 css box model은 기본적으로 HTML 요소를 감싸는 가상적인 box를 생각
- 웹 브라우저는 각 용소를 box로 간주하고 화면에 배치
- 각 box는 margin, border, padding을 가짐
- 이것을 이해해야만 깔끔한 디자인을 할 수 있음

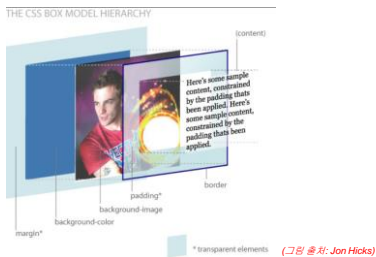
## 박스모델의 속성

- **Margin/padding**은 투명하여 배경색이나 무늬가 비쳐 보이게 된다.
- **Margin, border, padding** 속성을 이용해 각각의 폭이나 색을 조정할 수 있다.



## 배경색과 배경 이미지

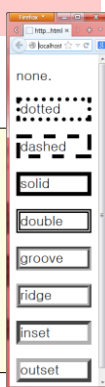
- HTML 요소에 **배경색**과 **배경 이미지**가 설정되어 있는 경우에, **패딩**은 투명하므로 배경 이미지와 배경색이 보이게 된다



## 경계선 스타일

- CSS border 속성을 이용하면 HTML 요소 경계선의 스타일과 색상을 지정
- Border 속성을 지정하려면 먼저 **border-style** 속성을 설정

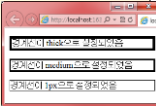
```
<!DOCTYPE html>
<html>
<body>
  <p style="border-style: none">none.</p>
  <p style="border-style: dotted">dotted.</p>
  <p style="border-style: dashed">dashed.</p>
  <p style="border-style: solid">solid.</p>
  <p style="border-style: double">double.</p>
  <p style="border-style: groove">groove.</p>
  <p style="border-style: ridge">ridge.</p>
  <p style="border-style: inset">inset.</p>
  <p style="border-style: outset">outset.</p>
</body>
</html>
```



## 경계선의 폭

- border-width 속성은 경계선의 폭을 지정
- 속성 값은 **px**, **thin**, **medium**, **thick** 중 하나
- border-style이 설정 되어 있어야 border-width가 동작

```
...
<head>
<style>
p.thick {
border-style: solid;      border-width: thick; }
p.medium {
border-style: solid;      border-width: medium; }
p.thin {
border-style: solid;      border-width: 1px; }
</style>
</head>
<body>
<p class="thick">경계선이 thick로 설정되었음</p>
<p class="medium">경계선이 medium으로 설정되었음</p>
<p class="thin">경계선이 1px으로 설정되었음</p>
</body></html>
```



## 경계선의 색상

- border-color 속성 값 : **"green"**, **rgb(0, 255, 0)**, **"#00ff00"**, **transparent** 중 하나

```
<!DOCTYPE html>
<html>
<head>
<style>
p.green {
border-style: solid;
border-color: green;
}
</style>
</head>
<body>
<p class="green">경계선의 색상: green</p>
</body>
</html>
```



## Border shorthand property

- 경계선의 스타일을 한 줄에 한꺼번에 설정 가능
  - 이를 축약 속성(shorthand property)이라고 함
- 반드시 **border-width**, **border-style(필수)**, **border-color** 순서로 작성
- 예)

**border: 5px solid red;**

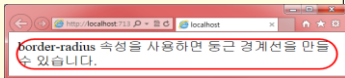
↑        ↑        ↑

border-width border-style border-color

## 둥근 경계선

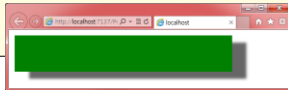
- css3에서는 둥근 경계선, 그림자 경계선, 경계 이미지 같은 고급 기능을 Photoshop 프로그램 없이도 생성 가능

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
border: 2px solid red;
border-radius: 25px;
}
</style>
</head>
<body>
<div>border-radius 속성을 사용하면 둥근 경계선을 만들 수 있습니다.</div>
</body>
</html>
```



## 경계선 그림자

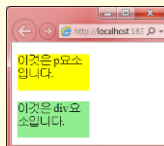
```
<!DOCTYPE html>
<html>
<head>
<style>
div {
width: 300px;
height: 50px;
background-color: green;
box-shadow: 20px 10px 5px #666666;
}
</style>
</head>
<body>
<div></div>
</body>
</html>
```



## 요소 크기 설정

- 지금 까지는 HTML 요소의 크기를 설정하지 않고 콘텐츠의 자연스러운 크기를 사용
- css에서는 모든 요소의 크기를 width와 height를 이용해 설정 가능

```
<!DOCTYPE html>
<html><head>
<style>
#target1 {
width: 100px;
height: 50px;
background-color: yellow; }
#target2 {
width: 100px;
height: 50px;
background-color: lightgreen; }
</style></head>
<body>
<p id="target1">이것은 p요소입니다.</p>
<div id="target2">이것은 div요소입니다.</div>
</body></html>
```



## 마진과 패딩 설정하기

값	설명
auto	브라우저가 마진을 계산한다.
length	마진을 px, pt, cm 단위로 지정할 수 있다. 디폴트는 0px이다.
%	마진을 요소 폭의 퍼센트로 지정한다.
inherit	마진이 부모 요소로부터 상속된다.

지정하는 2가지 방법



- 대칭의 각 변을 별도로 지정하는 방법.

```
margin-top:10px;
margin-right:20px;
margin-bottom:30px;
margin-left:40px;
```

- 한 줄에서 모든 변에 대한 값을 지정하는 방법.

```
margin: 10px 20px 30px 40px;
```

top right bottom left

```
margin: 10px /* 모든 margin이 10px로 설정*/
```

## ...Cont'd

- padding 설정하기 - padding은 콘텐츠와 border 사이의 간격
- padding을 지정하는 방법도 margin과 마찬가지로 2가지 방법이 있다.
- Padding의 각 변을 별도로 지정하는 방법:
 

```
padding-top: 10px;
padding-right: 20px;
padding-bottom: 30px;
padding-left: 40px;
```
- 한줄에서 모든 변에 대한 값을 지정하는 방법:
 

```
padding: 10px 20px 30px 40px;
```

```
padding: 10px /* 모든 padding이 10px로 설정*/
```

## 마진과 패딩 예제

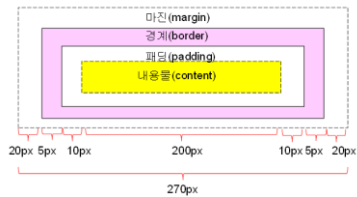
```

<style>
  body {
    margin: 0px;
    padding: 0px;
  }
  p {
    margin: 0px;
    padding: 0px;
    background-color: yellow; border: 1px solid red;
  }
  #target {
    margin: 10px;
    padding: 20px;
    background-color: lightgreen; border: 1px solid red;
  }
</style>
</head>
<body>
  <p>margin: 0px, padding: 0px인 단락입니다.</p>
  <p id="target">margin: 10px, padding: 20px인 단락입니다.</p>
</body>
</html>

```

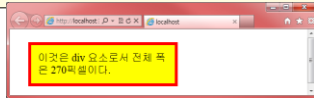
## 박스의 크기 계산

- css를 통하여 요소의 width와 height 속성을 설정하면 요소 안에서 콘텐츠 영역의 크기가 설정
- 요소의 전체 크기를 계산하려면 여기에 padding, border, margin을 더해야 한다.



## 실습

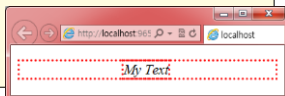
```
<!DOCTYPE html>
<html>
<head>
<style>
div.test {
background-color: yellow;
width: 200px;
padding: 10px;
border: 5px solid red;
margin: 20px;
}
</style>
</head>
<body>
<div class="test">이것은 div 요소로서 전체 폭은 270픽셀이다.</div>
</body>
</html>
```



## 수평정렬(인라인 요소)

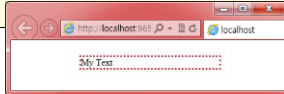
- HTML 요소를 화면의 중앙 정렬을 위해 align 속성을 사용
- css만을 사용하여 중앙 정렬 방법:

```
<!DOCTYPE html>
<html>
<head>
<style>
p, em, strong, div
{
border: dotted 3px red;
}
</style>
</head>
<body>
<p style="text-align: center">
<em>My Text </em>
</p>
</body>
</html>
```



## 수평정렬(블록 요소)

```
<!DOCTYPE html>
<html>
<head>
<style>
  p, h1
  {
    border: dotted 3px red;
  }
</style>
</head>
<body>
<p style="margin-left: auto; margin-right: auto; width: 50%">
  My Text
</p>
</body>
</html>
```



## 배경 설정하기

- css를 이용하여 배경과 관련된 속성은 다음과 같다.

속성	설명
<a href="#">background</a>	한줄에서 모든 배경 속성을 정의한다.
<a href="#">background-attachment</a>	배경 이미지가 고정되어 있는지 스크롤되는지를 지정한다.
<a href="#">background-color</a>	배경색을 정의한다.
<a href="#">background-image</a>	배경 이미지를 정의한다.
<a href="#">background-position</a>	배경 이미지의 시작 위치를 지정한다.
<a href="#">background-repeat</a>	배경 이미지의 반복 여부를 지정한다.

- 다음과 같이 16진수나 rgb 값, 색상 이름을 사용하여 설정

```
body { background-color: red; } /* 배경을 빨간색으로 설정한다 */
body { background-color: rgb(255,0,0); } /* 배경을 빨간색으로 설정한다 */
body { background-color: #ff0000; } /* 배경을 빨간색으로 설정한다 */
```

## 배경 이미지 설정

- [background-image](#) 속성을 사용해 배경으로 사용할 이미지 지정
- 배경 이미지는 수평이나 수직으로 반복하면서 칠함
- 이미지에 따라서 수평이나 수직으로만 반복이 필요한데, 이 경우에는 [background-repeat](#) 특성을 설정

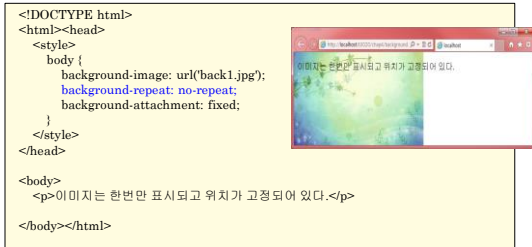
```
<!DOCTYPE html>
<html><head>
<style>
  body {
    background-image: url('back1.jpg')
  }
</style>
</head>

<body>
<h1>삶이 그대를 속일지라도</h1>
<p>삶이 그대를 속일지라도
슬퍼하거나 노하지 말아라.
...
지나가 버린 것 그리움이 되리니.
</p>
</body></html>
```



## 고정된 배경 이미지

- 만약 이미지를 한번만 표시하고 반복하지 않는 경우에는 다음과 같이 **no-repeat**로 지정
- **background-attachment** 속성: **scroll**(default), **fixed**(고정), **local** 중 하나
  - **local**은 요소의 콘텐츠와 같이 **scroll**된다.




---

---

---

---

---

---

---

---

---

---

## 배경 이미지 위치 설정

- **background-position** 속성은 배경 이미지 위치를 설정
- 위치는 다음과 같이 **px**, **%** 같은 단위로 지정 가능
  - **background-position: 100px 200px**
    - 이 경우 100px(x축/가로/수평)과 200px(y축/세로/수직)에 위치
- 다음과 같은 키워드를 사용:
  - **left top, left center, left bottom**
  - **right top, right center, right bottom**
  - **center top, center center, center bottom**
- 배경에 대한 속성을 한 줄로 설정하기
  - **body { background:#ffffff url('back.gif') no-repeat right top; }**

---

---

---

---

---

---

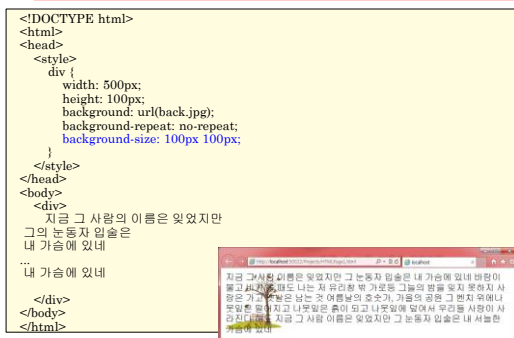
---

---

---

---

## 배경 이미지 크기




---

---

---

---

---

---

---

---

---

---



## CSS 레이아웃

### 레이아웃이란?

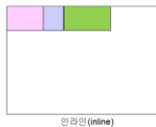
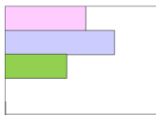
- 웹페이지에서 HTML 요소의 위치, 크기 등을 결정하는 것
  - 집안에서의 가구 배치와 비슷하다.



- 앞서 학습한 **box model**이란 웹브라우저가 각 요소를 화면에 그릴 때 무조건 사각형으로 간주한다는 것
- 이 사각형에는 **padding, border, margin** 이 붙어 있음
- 이와 같이 요소를 배치할 때도 브라우저는 사각형으로 간주

### 블록 요소와 인라인 요소

- HTML 요소는 **block** 요소이거나 **inline** 요소 중의 하나에 속한다.
- **블록(block) 요소** - 화면의 한 줄을 전부 차지한다.
- **인라인(inline) 요소** - 한 줄에 차례대로 배치된다. 현재 줄에서 필요한 만큼의 너비만을 차지한다.



## 블록 요소

- 한 줄을 전부 차지
- <h1>, <p>, <ul>, <li>, <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>
```



## 인라인 요소

- 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>

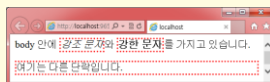
<a href="http://www.w3c.org">a 요소</a>
</body>
```



## 블록 요소와 인라인 요소의 혼합

- 일반적인 웹 페이지에는 블록 요소와 인라인 요소가 혼합되어 있다.
- 블록 요소가 나타날 때까지 인라인 요소는 같은 줄에 배치된다.
- 블록 요소가 나타나면 줄이 분리 되고 블록 요소는 새로운 줄에 배치된다.

```
<!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

- HTML 요소가 자동적으로 블록 요소와 인라인 요소로 나누어 지지만 이 속성은 css를 통하여 변경 가능
- 속성 **display**를 **block**으로 설정하면 -> 블록 요소처럼 배치
- display**를 **inline**으로 설정-> 인라인 요소처럼 배치
  - display: block** : 블록(block)
  - display: inline** : 인라인(inline)
  - display: none** : 없는 것으로 간주됨
  - display: hidden** : 화면에서 감춰짐
- 예) 블록 요소인 <div>요소를 다음과 같이 인라인으로 설정  

```
div { display: inline; }
```
- 주의할 점은 인라인 요소 안에 블록 요소를 넣으면 문제가 발생 할 수 있다.

---

---

---

---

---

---

---

---

## 실습

```

<!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>
                
```

---

---

---

---

---

---

---

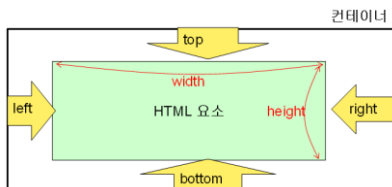
---

## 요소의 위치

- 기본적으로 요소의 위치는 다음과 같은 속성으로 지정
  - top, bottom, left, right** 속성으로 결정
- 예) 다음과 같이 지정하면 기준 위치에서 (100, 200) 만큼 떨어진 곳에 요소를 배치:
 

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

  - 즉 **left: 200px**은 컨테이너 왼쪽 경계에서 200px 만큼 떨어진 곳을 의미




---

---

---

---

---

---

---

---

## 위치 설정 방법

- 요소의 크기는 `width`, `height` 속성으로 요소의 너비와 높이를 설정
- 그러나 기준 위치는 `position` 속성으로 결정하며, `top`, `right`, `bottom`, `left` 속성은 `position` 속성이 먼저 설정되어야만 동작한다.
- `css`는 기본적으로 다음 4가지 위치 설정 방법이 존재:
  - 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 – 실습 1/2

- 정적 위치 설정(`static positioning`)
  - 블록 요소들은 박스처럼 상하로 쌓이게 되고 인라인 요소들은 한 줄에 차례대로 배치
  - 다음 예는 3개의 블록을 생성하고 중간 블록을 `static`으로 지정

```
<!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>
```

---

---

---

---

---

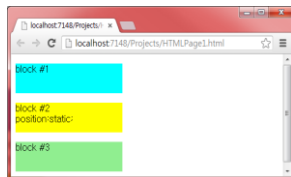
---

---

---

## 정적 위치 설정 – 실습 2/2

```
<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>
```



---

---

---

---

---

---

---

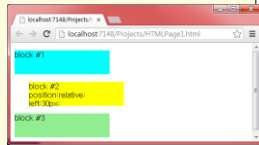
---

## 상대 위치 설정

- 상대 위치 설정(relative positioning)

- 정상적인 위치에서 상대적으로 요소가 배치
- 즉 요소를 위해 할당된 정상적인 공간은 없어지지 않는다.

```
<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>
```

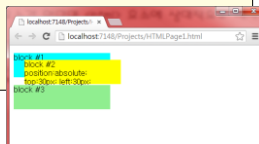


## 절대 위치 설정

- 절대 위치(absolute positioning)

- 전체 페이지를 기준으로 **페이지의 시작 (현 컨테이너)** 위치에서 top, left, bottom, right 만큼 떨어진 위치에 배치
- 예) right: 2px
  - 오른쪽으로 2px 만큼 이동하라는 의미가 아니고 페이지의 오른쪽 경계선에서 2px 만큼 떨어진 위치를 의미

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



## 고정 위치 설정 - 실습 1/2

- 고정 위치 설정(fixed positioning)

- 브라우저 윈도우에 상대적으로 요소의 위치를 잡는 것
- 만약 윈도우가 스크롤된다고 하여도 화면에서 요소는 움직이지 않는다.
  - 예) 페이지를 스크롤하여도 같은 위치에서 표시되는 쪽지 같은 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 - 실습 2/2

```

<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>
        
```

block #1

block #3

block #4

block #5

block #6

block #7

block #8

block #9

block #10

block #11

block #2  
position: fixed;  
top:0px; right:10px;

---

---

---

---

---

---

---

---

---

---

---

---

## float 속성

- 하나의 콘텐츠 주위로 다른 콘텐츠들이 물처럼 흘러가는 스타일 지정
- 다음과 같은 이미지에서 강물 위에 바위가 하나 있고 바위 주위로 물이 흘러가는 장면을 상상해 보자.
  - 이 때 바위를 왼쪽에 배치하고 싶다면 **float:left**라고 설정한다.
  - 이는 바위가 왼쪽에 떠있으라고 설정하는 것
- 이미지와 텍스트를 화면에 함께 표시할 때 이미지를 화면의 **오른쪽**이나 **왼쪽**에 띄우는 유용한 속성
- 속성:
  - float:right;**
  - float:left;**




---

---

---

---

---

---

---

---

---

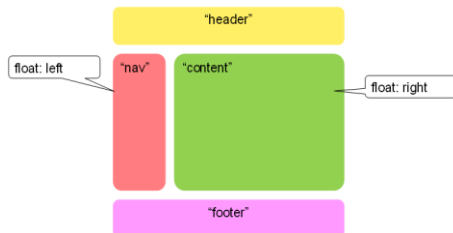
---

---

---

## float의 용도

- 웹 페이지의 레이아웃에 많이 사용된다.
- 다음은 4개의 <div> 요소로 레이아웃을 잡은 화면
- <div> 요소는 블록 요소로 한 줄 전체를 차지하지만 **"nav"**와 **"content"**는 한 줄에 같이 배치하고자 할 때 유용한 속성




---

---

---

---

---

---

---

---

---

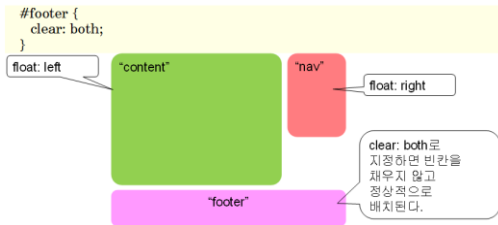
---

---

---

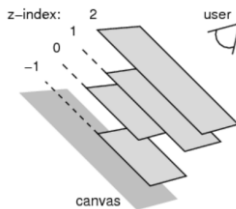
## clear 속성

- float 속성을 종단할 때 사용된다.
- 즉, clear는 float의 흐름을 제거하는 속성



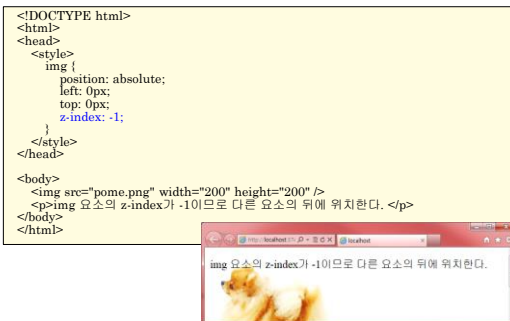
## z-index

- 요소의 스택 순서를 지정
- 즉, 요소가 다른 요소의 앞에 위치하느냐, 뒤에 위치하느냐를 지정
- 요소는 양수 혹은 음수의 스택 순서를 가질 수 있다.



## 예제

- Z-index 값을 조정하여 텍스트 뒤에 배치



## 요소의 크기 지정

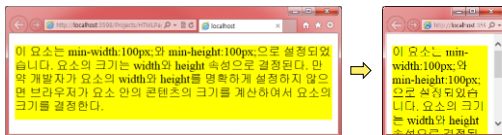
- width, height – 요소의 크기
- min-width, min-height: 요소의 최소 크기
- max-width, max-height: 요소의 최대 크기
- 만약 개발자가 요소의 크기를 정확하게 설정하지 않으면 브라우저가 요소 안의 콘텐츠 크기를 계산 해서 요소의 크기를 결정

## 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



- 윈도우를 움직여 보면 우측과 같이 100px이하로는 줄어들지 않는다.



## overflow 속성

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

---

---

---

---

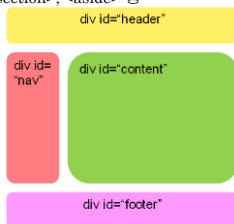
---

---

---

## <div>를 이용한 레이아웃

- 웹 페이지 레이아웃을 만드는데 현재 가장 널리 사용되는 방법
- **<div>** 요소는 블록 수준의 요소로서 다른 요소를 그룹핑하는데 사용
  - **css** 를 사용해 요소의 위치를 잡고 배경이나 색상을 지정
- 최근 **HTML 5**에서 다음 기능을 추가하여 여러가지 시맨틱 요소를 이용하여 레이아웃을 잡는 방법도 시도 되고 있음
  - **<nav>**, **<section>**, **<aside>** 등



---

---

---

---

---

---

---

## 시맨틱 요소 레이아웃

- 즉 의미적 요소(semantic elements)란 브라우저에게 요소의 의미나 목적을 명확하게 알려 주기 위한 요소
- 지금까지 학습한 **<div>** 요소는 HTML 문서 중에서 논리적인 **section**을 나타내고 컨테이너의 역할만 하는 것이지 그 자체로는 의미를 전달하지 못함
  - 물론 **id**나 **class** 이름과 함께 사용해서 의미를 전달할 수 있지만 브라우저는 인식하지 못함
- **HTML 5**에서는 이 문제를 해결하기 위해 시맨틱 요소를 정의하여 브라우저와 개발자에게 명확하게 그 의미를 설명하도록 함



---

---

---

---

---

---

---

## 시맨틱 요소

- W3c에서는 웹페이지를 분석해서 개발자가 많이 사용하고 있는 id와 class를 추출해 이것을 표준 태그로 다음과 같이 만들었다.

태그	설명
<code>&lt;header&gt;</code>	문서의 머리말(header)
<code>&lt;hgroup&gt;</code>	<code>&lt;h1&gt;</code> 에서 <code>&lt;h6&gt;</code> 요소들의 그룹
<code>&lt;nav&gt;</code>	내비게이션 링크
<code>&lt;article&gt;</code>	문서의 내용이나 블로그의 포스트 (문서 내용을 그룹핑하는데 사용)
<code>&lt;section&gt;</code>	문서의 섹션을 의미 (문서 내용을 그룹핑하는데 사용)
<code>&lt;aside&gt;</code>	사이드바와 같이 옆에 위치하는 내용 (보조적인 내용을 표시하는데 사용)
<code>&lt;footer&gt;</code>	문서의 꼬리말/바닥글(footer)
<code>&lt;figure&gt;</code>	그림이나 도표
<code>&lt;time&gt;</code>	날짜와 시간을 표시

## table-cell 속성

- 전 실습에서 중간 두개 section을 float:left와 float:right를 사용해도 되지만 컬럼의 높이가 다를 경우에는 배경색이 있는 경우 겹치게 나타나 어색하게 표현된다.
- display 속성에 table-cell을 하면 자식 요소들을 테이블의 셀처럼 배치하라는 의미가 된다.

