



CSS3+JavaScript와 함께 하는 HTML5 웹 프로그래밍

5장 CSS3 고급 활용

CHAPTER

05

HTML5+CSS3+JavaScript



CSS3 고급 활용

5.1 CSS3 박스모델

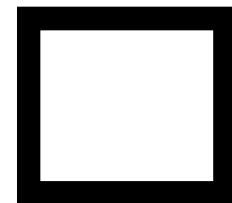
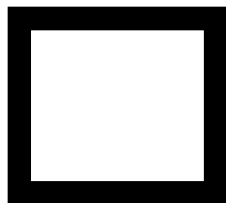
5.2 레이아웃 관련 속성들

5.3 CSS3의 고급 기능들

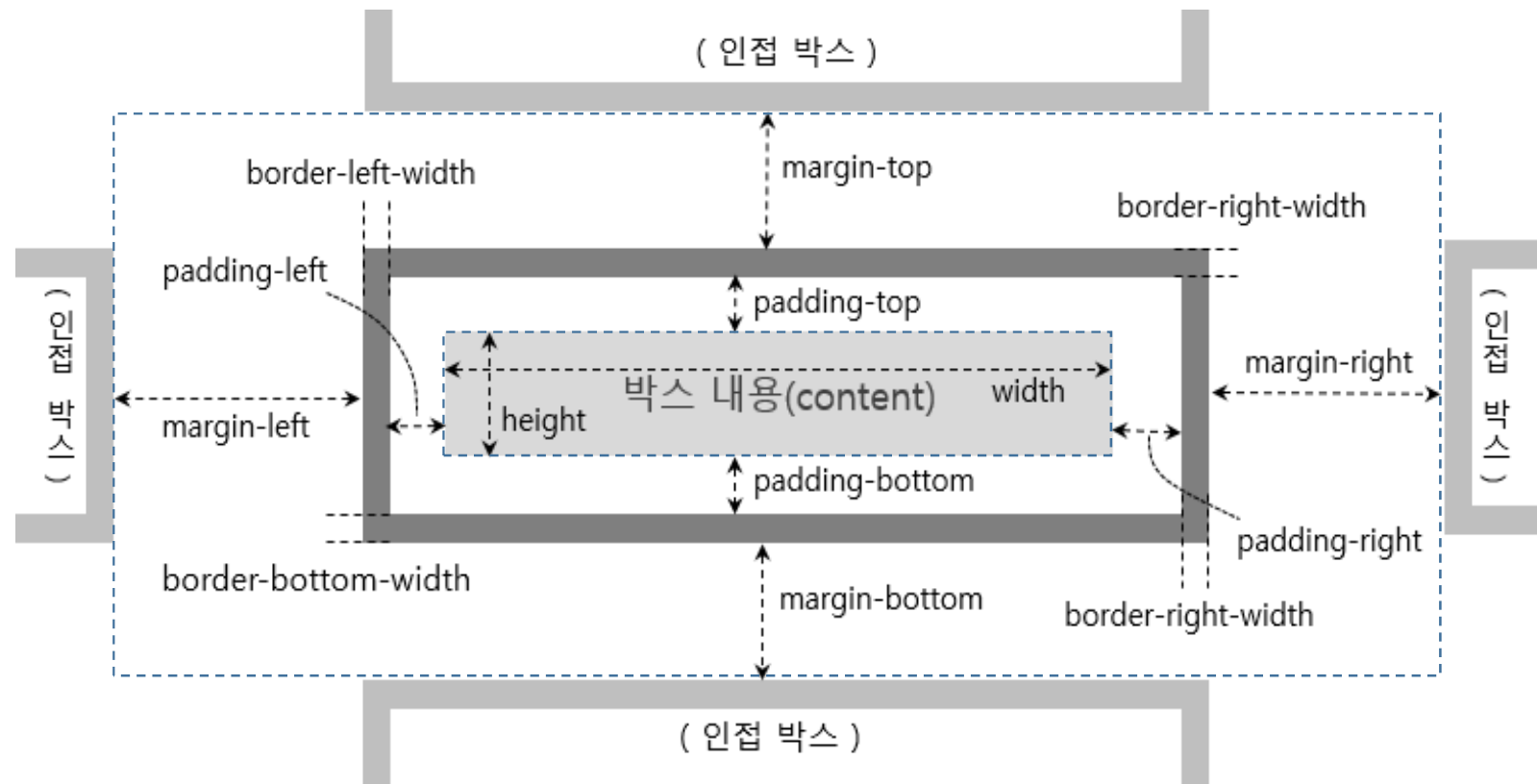
5.4 CSS3 응용 : 웹 페이지 레이아웃 구성

5.1 CSS3의 박스모델

- CSS3 박스 모델(Box Model)
 - ✓ CSS3 관점의 HTML 요소(태그)들의 기본 출력형태 및 출력스타일 지정 속성들
 - ✓ 즉, CSS3는 **HTML 요소를 웹 브라우저 화면에 나타낼 하나의 박스(box)로 생각하고**, 이 박스의 출력스타일을 나타내는데 필요한 속성들을 제공함
 - ✓ HTML 요소(태그) = 요소박스 = 태그박스 = 박스
- 웹브라우저 화면에서 한 박스가 차지하는(표시되는) 공간의 구성
 - ✓ 박스 테두리 선(border),
 - ✓ 테두리 선 외부여백(margin)
 - ✓ 내부여백(padding)
 - ✓ 박스내부에서 태그내용들이 차지하는 공간(width, height)



CSS3 박스 모델 : 박스 크기 지정



[표 5.1] 박스 크기 관련 속성들

속성 이름	속성 설명 및 속성값 종류 (관련 세부 속성 이름)	
margin	길이값, 박스 외부의 상하좌우 여백 (margin-top, margin-bottom, margin-left, margin-right : 상하좌우 외부 여백)	
padding	길이값, 박스내용과 테두리 선 사이의 상하좌우 내부 여백 (padding-top, padding-bottom, padding-left, padding-right : 상하좌우 내부 여백)	
border	박스의 상하좌우 테두리 선의 width, style, color (border-width : 테두리 선 두께, thin , medium, thick 또는 길이값(선 두께)) (border-style : 테두리 선 형태, solid, dotted, double, inset, none) (border-color : 테두리 선 색, 색상값) (border-top-width, border-bottom-width, border-left-width, border-right-width)	
width	길이값, 박스내용 표시 공간의 가로 길이 또는 테두리 선 포함 박스의 가로 길이	
height	길이값, 박스내용 표시 공간의 세로 길이 또는 테두리 선 포함 박스의 세로 길이	
box-sizing	width와 height 속성의 의미를 지정함(기본값 : content-box)	
	content-box	박스내용(content)이 차지하는 공간 크기
	border-box	테두리 선 포함 박스 공간 크기

(1) 박스의 마진 공간 지정 : margin 속성

- 박스 테두리 선의 바깥쪽 경계에서 인접해 있는 다른 박스들의 테두리 선까지의 외부 여백
 - 마진공간 지정 방법
 - 상하좌우 마진공간의 개별 지정(상하좌우 세부속성들 이용)
 - margin-top, margin-bottom, margin-left, margin-right
 - 상하좌우 마진공간의 동시 지정 : margin 속성 이용(단축표현)
 - margin : 값1 값2 값3 값4; /* 각각 박스의 상, 우, 하, 좌 마진공간 크기를 지정함 */
 - margin : 값1 값2; /* 각각 박스의 상하, 좌우 마진공간 크기를 지정함 */
 - margin : 값1; /* 박스의 상하좌우 마진공간을 동일한 크기(값1)로 지정함 */
-
- ① margin-top: 10px; margin-right: 20px; margin-bottom: 30px; margin-left: 40px;
⇔ margin: 10px 20px 30px 40px ;
 - ② margin: 10px 20px ; ⇔ margin 10px 20px 10px 20px ;
 - ③ margin: 10px ; ⇔ margin: 10px 10px;

박스의 위쪽부터 시계방향으로 돌아가면서
각 방향의 마진공간 크기를 지정함

```

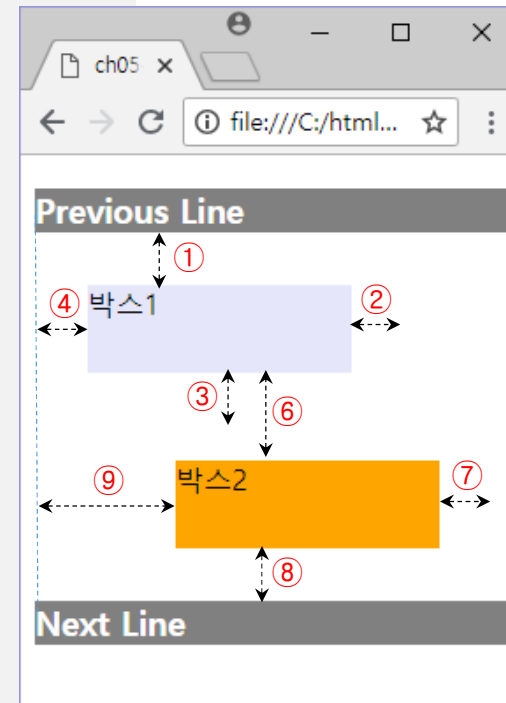
<style>
  h3 { background: gray; color: white; }

  #b1 { background-color: Lavender;
        width: 150px; height: 50px;
        ① margin-top: 30px;
        ② margin-right: 30px;
        ③ margin-bottom: 30px;
        ④ margin-left: 30px;
      }

  #b2 { background-color: orange;
        width: 150px; height: 50px;
        ⑤ margin : 50px 30px 30px 80px ;
      }
</style>

<h3> Previous Line </h3>
<div id="b1">박스1</div>
<div id="b2">박스2</div>
<h3> Next Line </h3>

```



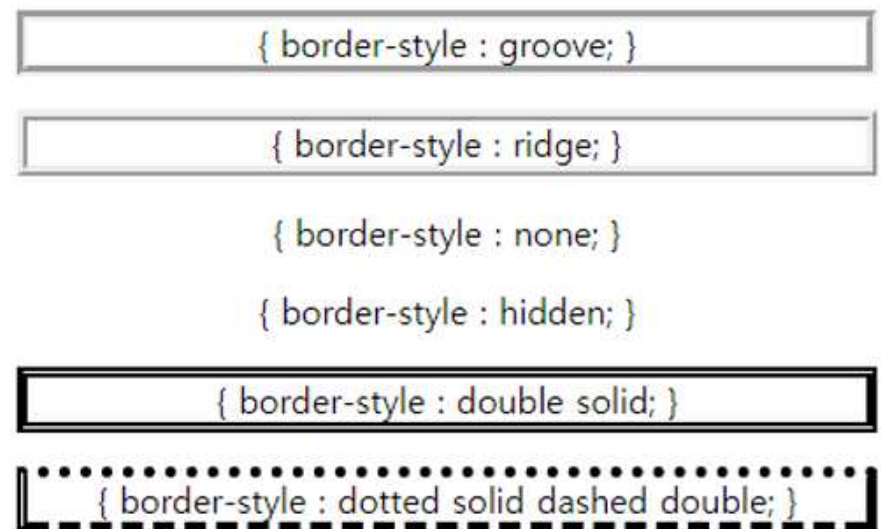
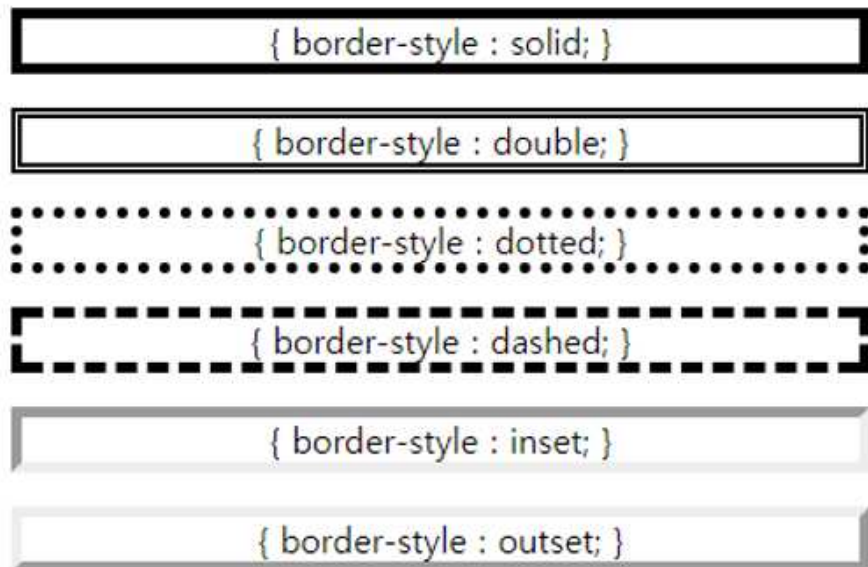
(2) 박스의 테두리 선 지정 : border 속성

박스의 테두리 선을 지정한다

테두리 선의 두께와 색상, 종류 지정

개별 지정 : border-width, border-color, border-style

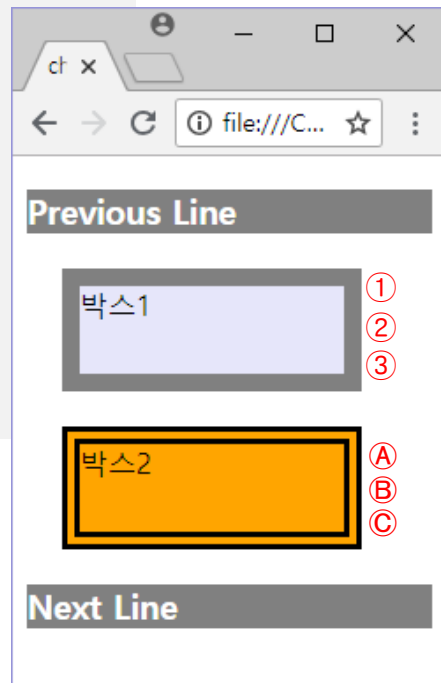
단축 표현 : border : 두께값 선-색상 선스타일값 ;



[예 5.2] 박스의 테두리 선

```
#b1 { background-color: Lavender;
      width:150px; height: 50px;
      margin: 20px;
      ① border-width: 10px;
      ② border-color: gray;
      ③ border-style: solid;
    }

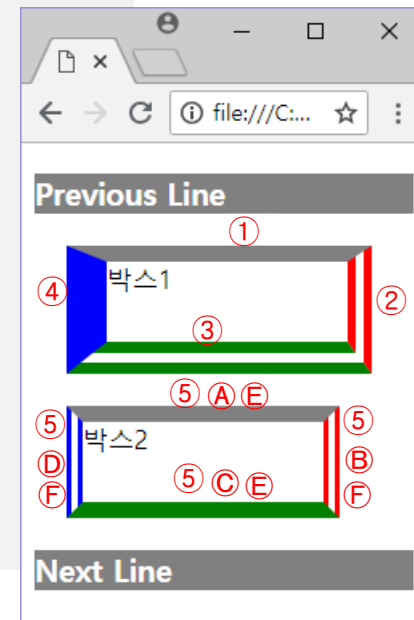
#b2 { background-color: orange;
      width: 150px; height: 50px;
      margin: 20px;
      ④ border: 10px black double;
    }
      (A) (B) (C)
```



[예 5.3] 박스의 border 속성

```
#b1 { background-color: Lavender;
      width:150px; height: 50px;
      margin: 20px;
      ① border-top: 10px gray solid;
      ② border-right: 15px red double;
      ③ border-bottom: 20px green double;
      ④ border-left: 25px blue solid;
    }

#b2 { background-color: orange;
      width: 150px; height: 50px;
      margin: 20px;
      ⑤ border-width: 10px;
      ⑥ border-color: gray red green blue;
      ⑦ border-style: solid double;
    }
      (A) (B) (C) (D) (E) (F)
```

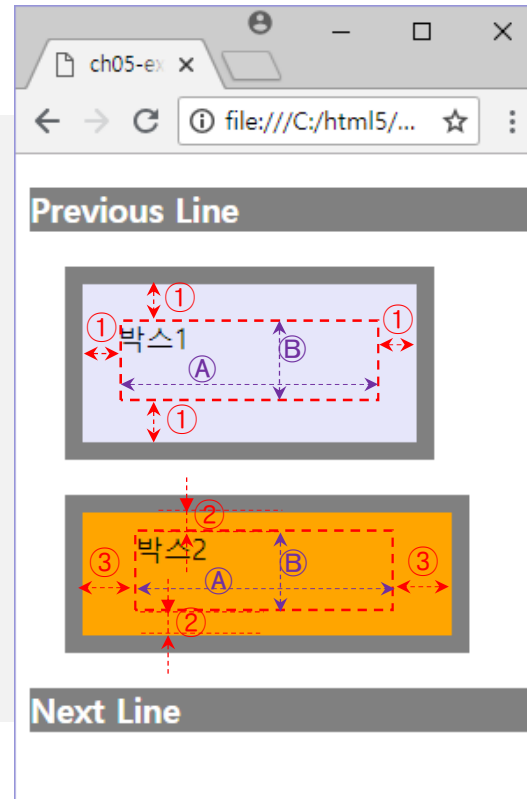


(3) 박스의 패딩공간 지정 : padding 속성

- 박스내부에서 태그내용이 차지하는 공간과 테두리 선의 안쪽 경계 사이의 내부여백
- 패딩공간 지정 방법
 - 상하좌우 마진공간의 개별 지정(상하좌우 세부속성들 이용)
 - ✓ padding-top, -bottom, -left, -right
 - 상하좌우 패딩공간의 동시 지정 : padding 속성 이용(단축표현)
 - ✓ padding : 값1 값2 값3 값4; /* 각각 박스의 상, 우, 하, 좌 패딩공간 크기를 지정함 */
 - ✓ padding : 값1 값2; /* 각각 박스의 상하, 좌우 패딩공간 크기를 지정함 */
 - ✓ padding : 값1; /* 박스의 상하좌우 패딩공간을 동일한 크기(값1)로 지정함 */
- 패딩공간을 지정하면, 태그내용들은 박스내부에서 위쪽과 왼쪽 패딩공간을 침범하지 않고 출력됨

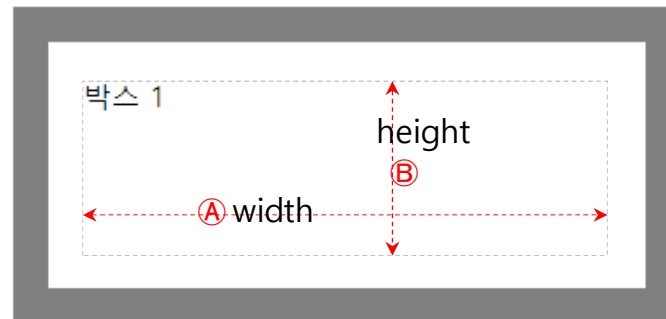
[예 5.4] 박스의 패딩 공간

```
div { margin: 20px;  
  Ⓐ width:150px;  
  Ⓑ height: 50px;  
  border: 10px gray solid;  
}  
  
#b1 { background-color: Lavender;  
  ① padding: 20px;  
}  
  
#b2 { background-color: orange;  
  padding: 10px 30px;  
}      ②      ③
```



(4) 박스 크기 지정 : width, height 속성

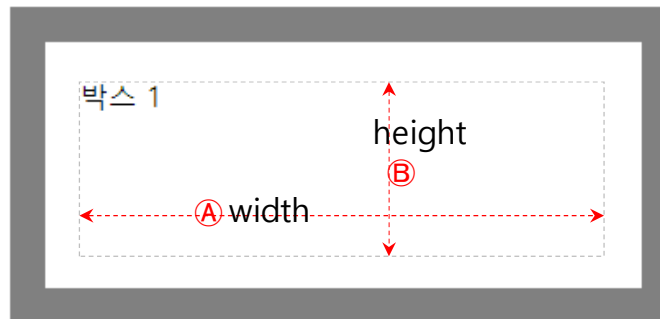
- width : 박스 내부에서 태그내용이 출력되는 공간의 가로 길이 지정
- height : 박스 내부에서 태그내용이 출력되는 공간의 세로 길이 지정
- 패딩 공간을 지정하면, 패딩 공간을 제외한 박스의 내부 공간의 크기를 지정함
 - 박스공간의 전체 가로 크기 = $2 * (\text{margin} + \text{border-width} + \text{padding}) + \text{width}$
 - 박스공간의 전체 세로 크기 = $2 * (\text{margin} + \text{border-width} + \text{padding}) + \text{height}$



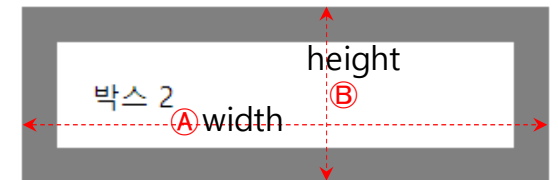
(5) 보더박스과 콘텐츠박스 지정 : box-sizing 속성

- box-sizing 속성
 - 테두리 선 두께와 패딩 공간 크기의 지정에 따른 width와 height 속성의 사용혼란을 방지하고, width와 height 속성이 나타내는 크기를 직관적으로 파악할 수 있음
- 콘텐츠 박스(content-box) : box-sizing 을 content-box 로 지정한 박스(예 : ①)
- 보더박스(border-box) : box-sizing을 border-box로 지정한 박스 (예 : ②)

```
<style>
div {
  ① width : 300px;
  ② height : 100px;
  margin : 20px; padding : 20px;
  border : 20px gray solid ;
}
① #b2 { box-sizing : content-box; }
② #b2 { box-sizing : border-box; }
</style>
```



① box-sizing="content-box" 인 경우



② box-sizing="border-box" 인 경우

5.1.2 CSS3 박스 유형

- 박스 유형 : 웹 브라우저 화면에 출력되는 특성에 따라 구분함
 - 블록 박스(block box) : <h1>, <p>, <div> 등의 태그 박스들
 - 인라인 박스 (inline box) : <a>, 등의 태그 박스들
 - width, height, margin-top, margin-bottom 속성 무시
 - 인라인-블록 박스(inline-block box) : 인라인 박스 기본 + 블록박스 특성 추가

특성 \ 종류	블록 박스	인라인 박스	인라인-블록 박스
출력 시작 위치	새로운 라인	현재 라인의 현재 위치	인라인 박스와 동일함
타 박스들과의 관계	같은 라인에 다른 블록 박스들과 연속해서 표시 불가능	같은 라인에서 다른 인라인 박스들과 연속해서 표시 됨	인라인 박스와 동일함
박스 속성들의 사용가능 여부	모든 박스 속성 지정 가능	박스 크기, 상하 마진 지정 불가능 (width, height, margin-top, margin-bottom 속성은 무시함)	블록 박스와 동일함

- HTML 태그들은 기본 박스유형이 정해져 있음(display 속성으로 박스유형 변경 가능함)

[예 5.5] 박스 유형의 비교 : 블록/인라인/인라인-블록 박스

```
<style>
  div, img { margin: 20px 20px; padding: 10px;
    width : 200px; height: 70px;
  }
  p { background: aqua; }
  div { background: skyblue; }
  span { background: orange; border: 5px solid red;
    padding: 20px; margin: 150px 150px;
    width : 200px; height: 70px;
  }
  img { background : green; }
</style>
```

```
<body>
이 페이지는 각 박스 유형의 출력 예를 나타낸다. <div> div 태그와 p 태그</div> 박스는 블록 유형이다.
<p>그리고 <span> span 태그</span> 박스는 인라인 유형이다.</p><p>img 태그로 표현하는 이미지  박스는 인라인-블록 유형이다. </p>
</body>
```



[CSS3 팁-1] 박스 내부의 텍스트 중앙정렬

- ❶ 좌우방향 중앙정렬 : text-align 속성 지정
즉, text-align : center;
- ❷ 상하방향으로 중앙정렬 : height 속성 지정, line-height 속성값=height 속성값 지정
즉, height: 값 ; line-height: height-값;

```
<style>
  div { margin: 20px; padding: 0px;
        width: 350px; height: 80px;
      }
  #b1 { background: orange; }
  #b2 { background: green; text-align: center; }
  #b3 { background: skyblue; text-align: center;
        line-height : 80px;
      }
</style>

<div id="b1">CSS3 스타일시트(기본출력)</div>
<div id="b2">CSS3 스타일시트(좌우 중앙정렬)</div>
<div id="b3">CSS3 스타일시트(박스내부 중앙출력)</div>
```

예 5.6 박스내용의 중앙정렬



[CSS3 팁-2] HTML 문서의 중앙정렬 레이아웃

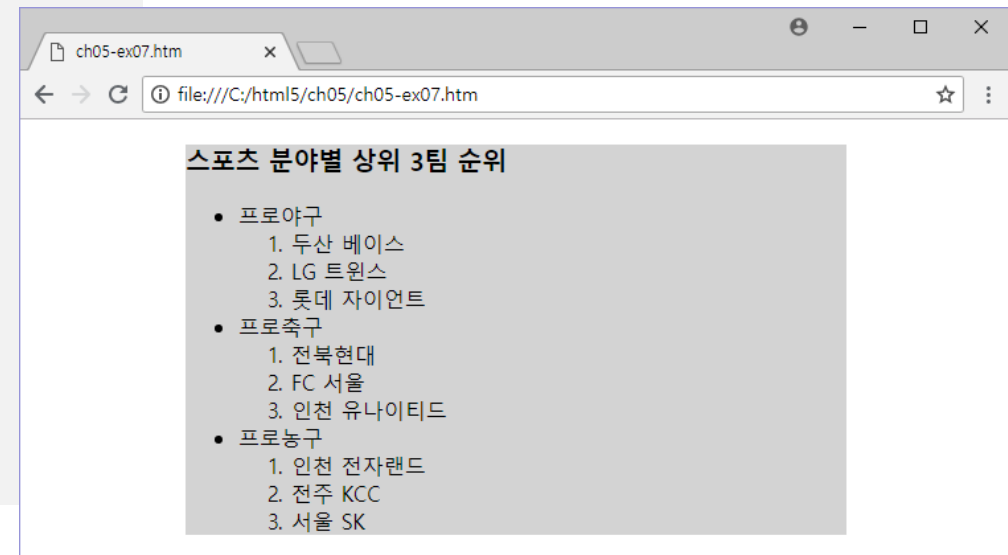
- ❶ 전체 문서내용(즉, <body>의 태그 내용)를 블록박스로 지정함 : <div> 태그 이용
즉, <body> <div> 문서 내용 </div> </body>
- ❷ div 박스의 width 속성 지정, margin 속성값은 "0 auto"로 지정함
즉, width : 500px; margin: 0 auto;

```
<style>
  div { width: 500px; margin: 0 auto;
        background: lightgray;
  }
</style>

<body>
<div>
<h3>스포츠 분야별 상위 3팀 순위</h3>
<ul>
  <li>프로야구</li>
  <ol>
    <li>두산 베이스</li>
    <li>LG 트윈스</li>
    <li>롯데 자이언츠</li>
  </ol>
  <li>프로축구</li>
  <ol>
    <li>전북현대</li>
    <li>FC 서울</li>
    <li>인천 유나이티드</li>
  </ol>
  <li>프로농구</li>
  <ol>
    <li>인천 전자랜드</li>
    <li>전주 KCC</li>
    <li>서울 SK</li>
  </ol>
  </ul>
</div>
</body>
```

```
<li>프로야구</li>
<ol>
  <li>전북현대</li>
  <li>FC 서울</li>
  <li>인천 유나이티드</li>
</ol>
<li>프로농구</li>
<ol>
  <li>인천 전자랜드</li>
  <li>전주 KCC</li>
  <li>서울 SK</li>
</ol>
</ul>
</div>
</body>
```

예 5.7 문서 내용의 화면 중앙정렬

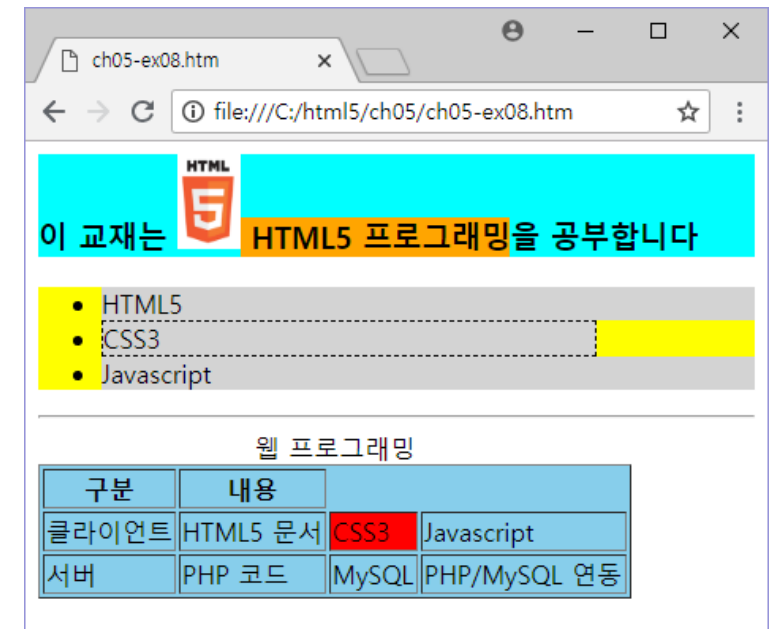


5.2.1 박스 유형 변경 : display 속성

- HTML 태그들은 출력할 때 기본적으로 블록, 인라인, 인라인-블록 중 한 박스유형으로 출력됨
 - <h1>, 등 : 블록 박스 출력
 - 등 : 인라인 박스 출력
 - : 인라인-블록 박스 출력
 - 그외, , <table>, <td> : 각각 리스트항목, 테이블, 테이블-셀 박스 출력
- display 속성으로 기본 박스유형은 변경 가능함

예 5.8 HTML 태그들의 기본 박스 유형

```
<style>
  h3 { background : aqua ; }
  span { background : orange; }
  img { background : green; width: 40px; height:60px; }
  ul { background : yellow ; }
  li { background : lightgray ; }
  table { background : skyblue ; }
  li#a { display: none; margin-right: 100px; }
  td#b { background : red ; }
</style>
```



[표 5.3] display 속성의 주요값

속성값	의미
none	박스를 출력하지 않음
block	라인 단위 박스를 구성함
inline	한 라인 내의 일부분을 좌우 외부 여백과 내부 여백만 지정 가능한 박스로 구성함 즉, 박스 속성들 중 margin-top/bottom, width, height 속성은 무시함
inline-block	한 라인 내의 일부분을 블록 박스로 구성함(즉, margin, padding, border, width, height 등 박스모델에서 설명한 모든 속성들을 지정할 수 있음)
table, table-cell	테이블 박스를 구성함
list-item	리스트 항목 박스를 구성함

[예 5.9] 박스 유형 변경을 이용한 메뉴 리스트 작성

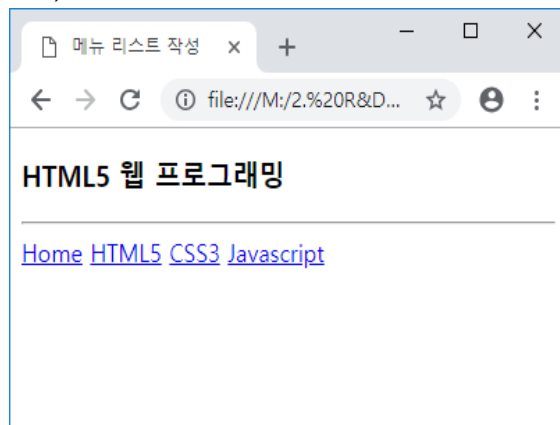
```
<!DOCTYPE html>
<html>
<head>
  <title> 메뉴리스트 작성 </title>
</head>
<body>
  <h3> HTML5 웹 프로그래밍 </h3> <hr>
  <div>
    <a href="#home">Home</a>
    <a href="#news">HTML5</a>
    <a href="#contact">CSS3</a>
    <a href="#about">Javascript</a>
  </div>
</body>
</html>
```

[예제문서]

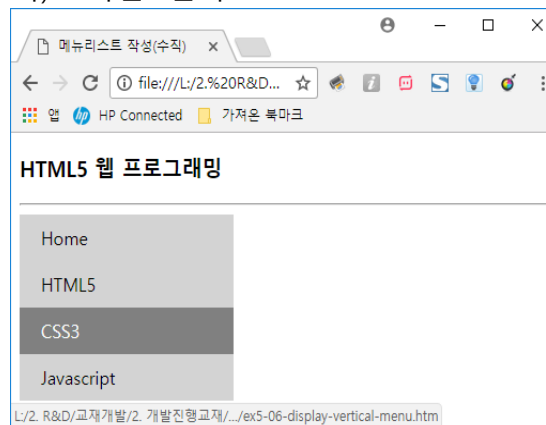
```
<style> /* 출력스타일 1 : 메뉴 리스트 수직출력 */      [스타일 1]
div { background: lightgray;
      margin: 0; padding: 0; width: 200px; }
a { display: block;
    color: black; padding: 10px 20px; text-decoration: none; }
a:hover { background: gray; color: white; }
</style>
```

```
<style> /* 출력 스타일 2 : 메뉴 리스트 수평출력 */      [스타일 2]
ul { background: lightgray; padding: 6px;}
a { color: black; text-decoration: none; padding: 5px 10px; }
a:hover { background: gray; color: white; }
</style>
```

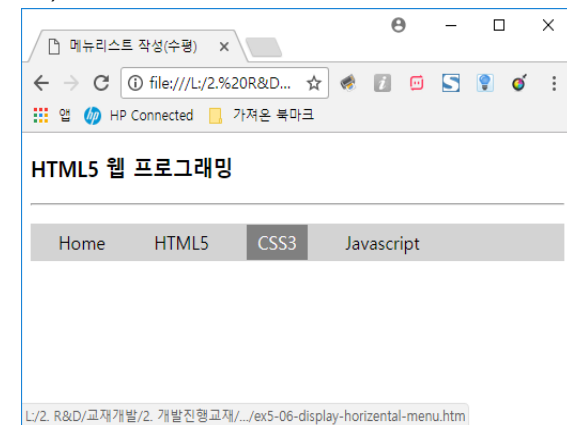
가) 기본출력



나) 스타일1 출력



다) 스타일2 출력



```

<!DOCTYPE html>
<html>
<head>
  <title> 메뉴리스트 작성 </title>
  <style> a { width: 150px; height: 50px;
    margin: 50px 20px;
    background: gray; color: black;
    padding: 5px 10px; } </style>
</head>
<body>
  <h3> HTML5 웹 프로그래밍 </h3> <hr>
  <div>
    <a href="#home">Home</a>
    <a href="#news">HTML5</a>
    <a href="#contact">CSS3</a>
    <a href="#about">Javascript</a>
  </div>
</body>
</html>

```

[예제문서]

```

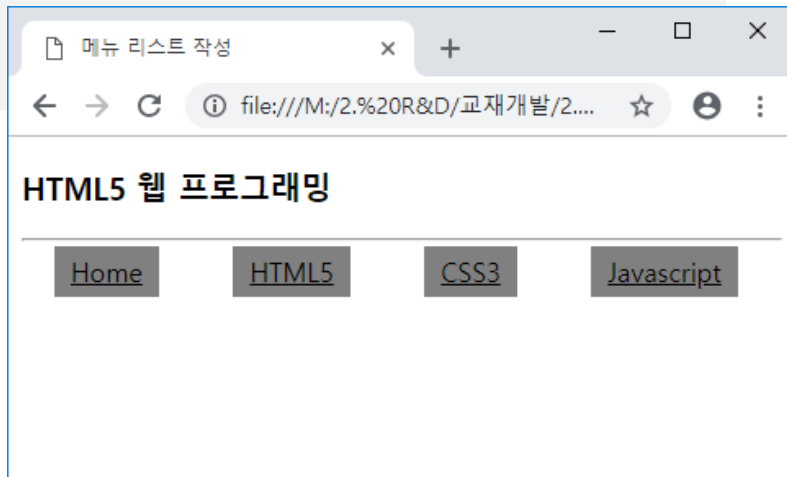
<style> /* 출력스타일 1 : 메뉴 리스트 수직출력 */      [스타일 1]
div { background: lightgray;
    margin: 0; padding: 0; width: 200px; }
a { display: block;
    color: black; padding: 10px 20px; text-decoration: none; }
a:hover { background: gray; color: white; }
</style>

```

```

<style> /* 출력 스타일 2 : 메뉴 리스트 수평출력 */      [스타일 2]
ul { background: lightgray; padding: 6px;}
a { color: black; text-decoration: none; padding: 5px 10px; }
a:hover { background: gray; color: white; }
</style>

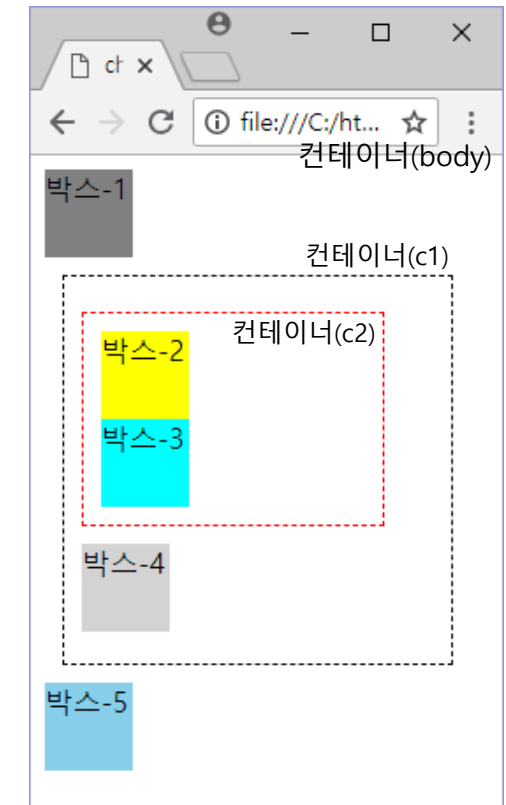
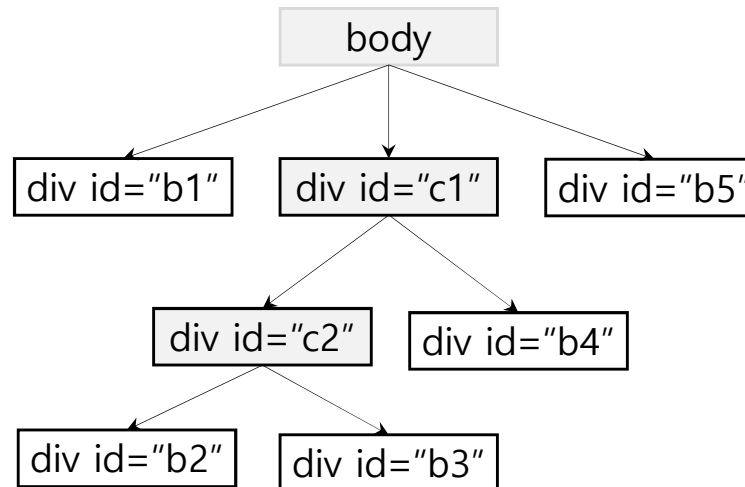
```



● 박스들의 포함관계

- 컨테이너 박스(container box, 간단히 컨테이너) : 다른 박스들을 포함하는 박스
 - 부모컨테이너(parent container, 또는 간단히 부모박스) : 특정 박스를 직접 포함하는 컨테이너 박스
- <body> 박스는 모든 HTML 태그박스들의 최상위 컨테이너임

```
<body>
<div id="b1"> 박스-1 </div>
<div id="c1">
  <div id="c2" >
    <div id="b2"> 박스-2 </div>
    <div id="b3"> 박스-3 </div>
  </div>
  <div id="b4"> 박스-4 </div>
</div>
<div id="b5"> 박스-5 </div>
</body>
```



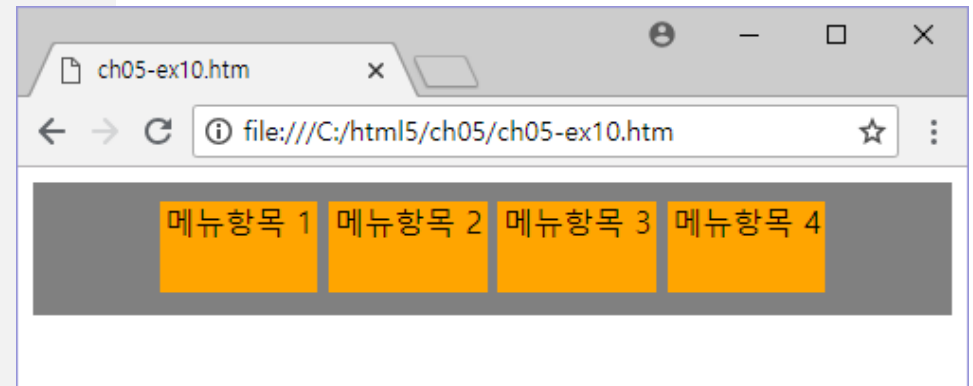
[그림 5.4] 박스들의 포함관계와 컨테이너 개념

[예 5.10] 자식박스들의 수평배열

- 부모박스에 포함되는 자식박스들을 수평방향으로 배열 : display 속성 이용
 - 자식박스들의 display 속성을 inline-block으로 지정함

```
<style>
  #parent { height: 100px; height: 50px; padding : 10px;
    background: gray; text-align: center;
  }
  .child { background: orange ; margin: 0px auto;
    display : inline-block ;
    width: 18%; height: 96%;
  }
</style>

<div id="parent" >
  <div class="child"> 메뉴항목 1 </div>
  <div class="child"> 메뉴항목 2 </div>
  <div class="child"> 메뉴항목 3 </div>
  <div class="child"> 메뉴항목 4</div>
</div>
```



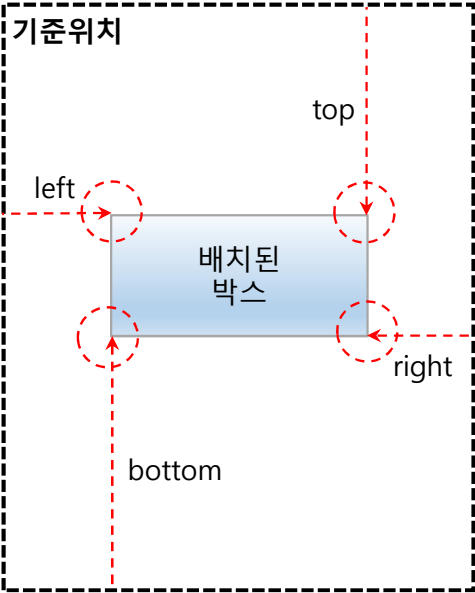
5.2.2 박스의 위치 지정 : position과 top, left, right, bottom 속성

- 웹브라우저 화면에서 박스들의 기본적인 배치
HTML 문서에서 작성된 순서에 따라 순차적으로 배치해서 출력함
- CSS3 스타일시트 이용 : 박스들의 기본적인 배치 변경가능
- CSS3의 박스의 배치 위치 : 배치공간과 배치공간에서의 그 기준위치 선택방법에 따라 구분됨
 - 정적(static)-기본, 상대(relative), 절대(absolute), 고정(fixed) 위치
 - 박스의 기본 배치 : 정적 배치
 - 배치컨테이너 : 부모박스,
 - 기준위치 : 부모박스 공간에서 작성 순서에 따라 순차적으로 배치될 때의 박스위치
 - 박스의 배치위치 변경 : position, top, bottom, left, right 속성 지정
 - position : 박스의 배치 방법 지정
 - 속성값 : static(기본값), none , relative, absolute, fixed,
 - top, bottom, left, right : 배치 컨테이너에서의 박스 위치 지정
 - 배치컨테이너에서 기준위치에 대한 박스의 상대적인 위치를 나타냄
 - 각각 기준위치(박스)의 상하좌우 경계에서 배치 박스의 상하좌우 경계까지의 거리(offset)
 - 일반적으로, top과 left (즉, 박스의 좌측 상단점의 위치) 으로 지정함
 - 박스들이 기본적으로 배치될 때는 이 속성값들은 지정하더라도 무시됨
(top와 left 속성값은 항상 0, 0 임)

[표 5.4] position 속성의 주요값

속성값	배치컨테이너	배치 기준 위치 및 위치 지정 방법
static (기본값)	부모컨테이너	- 부모컨테이너에서 해당 태그박스 순서에 따른 순차적 위치 지정 - top, bottom, left, right 속성을 무시함
relative	부모컨테이너	- 배치컨테이너에서 해당 박스의 정적 위치 공간 기준의 상대적 위치 지정 - 해당 박스의 정적 위치 공간은 계속 유지됨(즉, 후속 박스들은 정적 위치로 배치할 때 이 공간을 침범할 수 없음)
absolute	정적 위치가 아닌 가장 가까운 상위컨테이너	- 정적 위치가 아닌 가장 가까운 상위컨테이너 공간 기준의 상대적 위치 지정 - 해당 박스의 정적 위치 공간은 무시됨
fixed	브라우저 현재 화면	- 현재 웹 브라우저 화면 공간 기준의 고정 위치 지정 - 해당 박스의 정적 위치 공간은 무시됨(absolute와 동일함)
none	부모컨테이너	- 박스를 보이지 않게 함

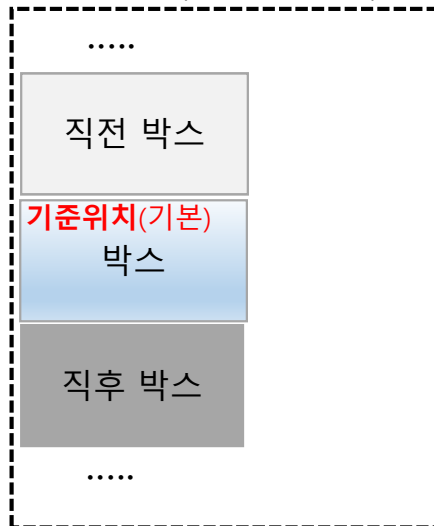
[그림 5.6]
박스 위치 속성들



[그림 5.5] 박스의 위치 지정에 따른 배치위치

```
<div id="container">
  ....
  <div>직전 박스</div>
  <div>박스</div>
  <div>직후 박스</div>
  ....
</div>
```

배치컨테이너(부모박스, 기본)



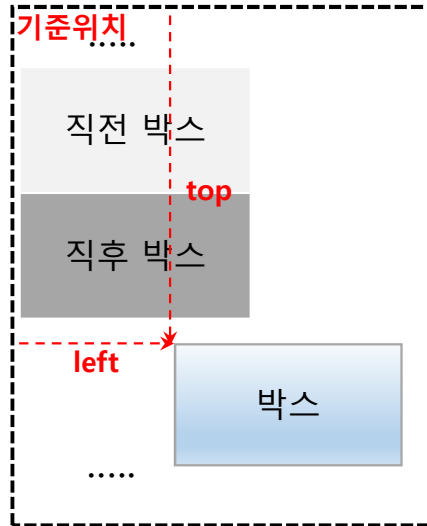
가) 기본위치(static)

배치컨테이너(부모박스)



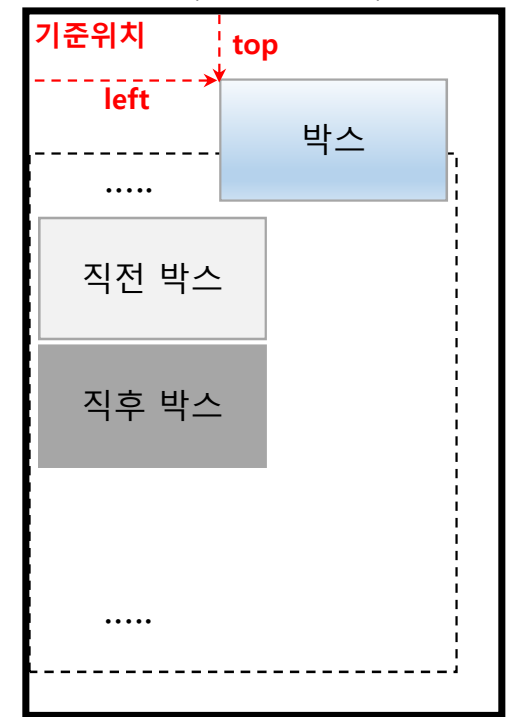
나) 상대 위치(relative)

배치컨테이너(부모/조상 박스)



다) 절대 위치(absolute)

배치컨테이너(웹브라우저 창)

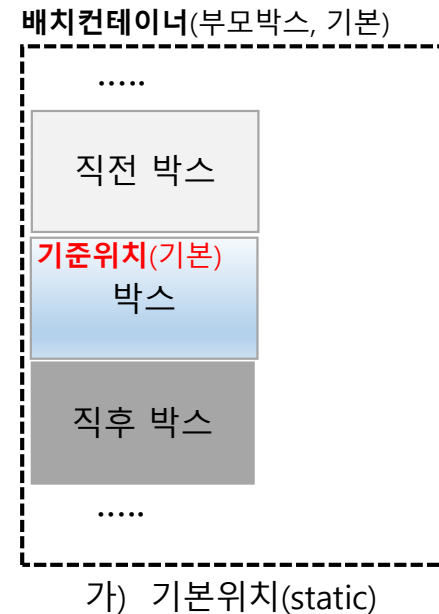


라) 고정위치(fixed)

(1) 기본 배치(정적 위치) 지정 : { position: "static";}

- 정적 위치
 - 부모박스(배치컨테이너) 공간에서 절대 위치로 배치된 박스들을 제외하고 내부 박스들을 작성된 순서대로 나타낼 때 배치되는 위치
- position 속성의 지정을 생략하거나 속성값을 static로 지정함
- 정적 위치의 박스에 대한 top, left 속성의 지정은 무시됨

```
<div id="container">  
  ....  
  <div>직전 박스</div>  
  <div>박스</div>  
  <div>직후 박스</div>  
  ....  
</div>
```



[예 5.11] 기본 배치 : 정적 위치 지정

```
<style>
  div { width : 150px; height : 50px; }
  #b1 { background : gray; }
  #b2 { background : green; margin: 0px ; }
  #b3 { background : blue; margin: 0px 10px; }
  #b4 { background : orange; margin: 0px 20px;}
  #b5 { background : darkgray;}

  #container { width: 300px; height: 200px;
    margin: 0px 10px; border: 2px black dashed;
  }
</style>
```

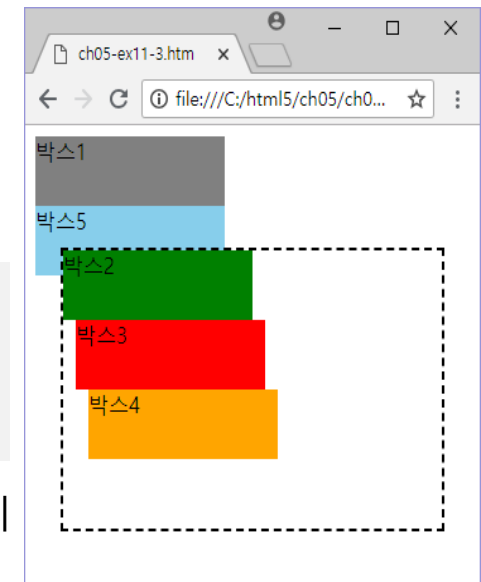
```
<body>
  <div id="b1">박스1</div>
  <div id="container" >
    <div id="b2">박스2</div>
    <div id="b3">박스3</div>
    <div id="b4">박스4</div>
  </div>
  <div id="b5">박스5</div>
</body>
```

가) 정적위치 지정(기본배치)



```
#container { height: 130px; }
```

나) container 박스:오버플로우



```
#container {
  position: absolute;
  margin: 30px 20px;
}
```

다) container 박스:절대배치

(2) 상대 배치(상대 위치) 지정 : { position: relative; top: 길이값; left: 길이값; }

- position 속성값을 relative로 지정하고, top과 left 속성을 이용해서 위치를 지정함
 - 배치컨테이너 : 기본 배치일 때와 동일하게 부모박스 공간
 - 원래의 정적위치 공간을 기준으로 하는 상대적 위치 지정(top, left 속성값)
 - top과 left는 부모박스의 기본 배치공간을 기준으로 상대적 위치를 나타냄
 - 부모박스를 배치가 변경되더라도 속성값은 변하지 않음
- 원래의 기본 배치 공간도 유지함, 뒤따라오는 형제박스들의 배치에는 영향을 주지 않음
- 부모박스 내의 다른 박스들의 배치에 영향을 주지 않고 단지 기본 배치 공간을 기준으로 이동 배치해서 나타낼 때 사용함

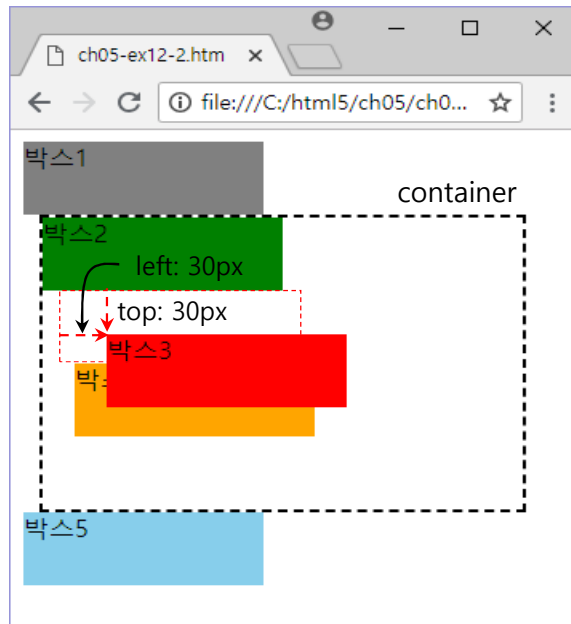
```
<div id="container">
  .....
  <div> 직전 박스</div>
  <div> 박스</div>
  <div> 직후 박스</div>
  .....
</div>
```



나) 상대 위치(relative)

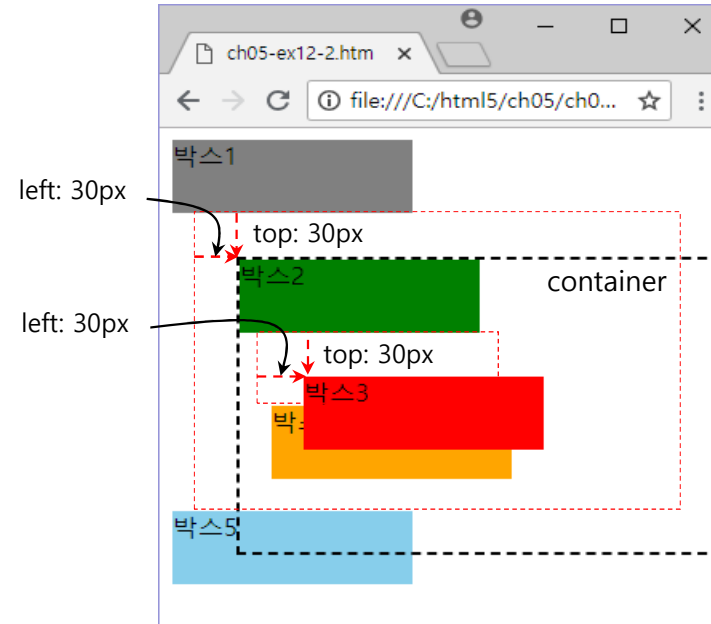
[예 5.12] 상대 배치 : 상대 위치 지정

```
#b3 { position : relative;  
      top: 30px; left: 30px;  
    }
```



가) 박스3 : 상대배치

```
#b3 { position : relative;  
      top: 30px; left: 30px; }  
#container { position: relative;  
             top: 30px; left: 30px; }
```



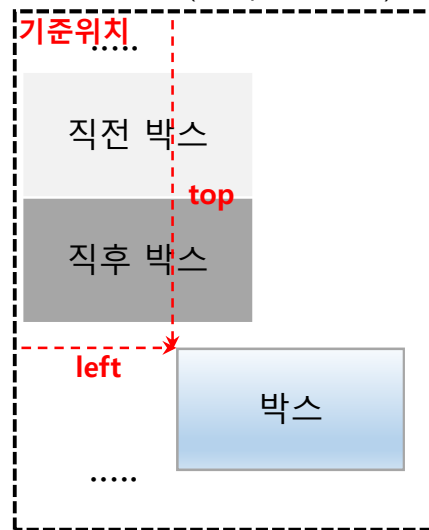
나) container 박스, 박스3 : 상대배치

(3) 절대 배치(절대 위치) 지정 : { position: absolute; top: 길이값; left: 길이값; }

- position 속성값을 absolute로 지정하고, top과 left 속성을 이용해서 위치를 지정함
- 배치 공간 및 기준 위치 : **상대 위치 또는 절대 위치로 지정된 가장 가까운 상위의 박스(배치 컨테이너)**
 - 위치를 top와 left 속성값 : 기준위치에 대한 상대적 나타냄
- 상대위치 또는 절대위치로 지정된 가장 가까운 상위의 박스가 없다면 <body> 박스가 배치 컨테이너 및 기준 위치가 됨
- 원래의 정적 위치 공간은 무시함. 따라서, 뒤따르는 형제박스들의 기본 배치에 영향을 줌

```
<div id="container">
  .....
  <div>직전 박스</div>
  <div>박스</div>
  <div>직후 박스</div>
  .....
</div>
```

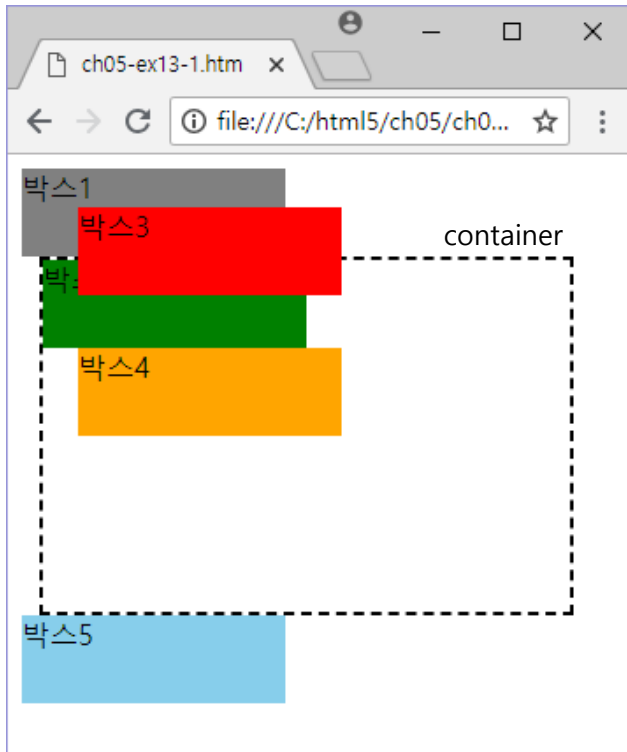
배치컨테이너(부모/조상 박스)



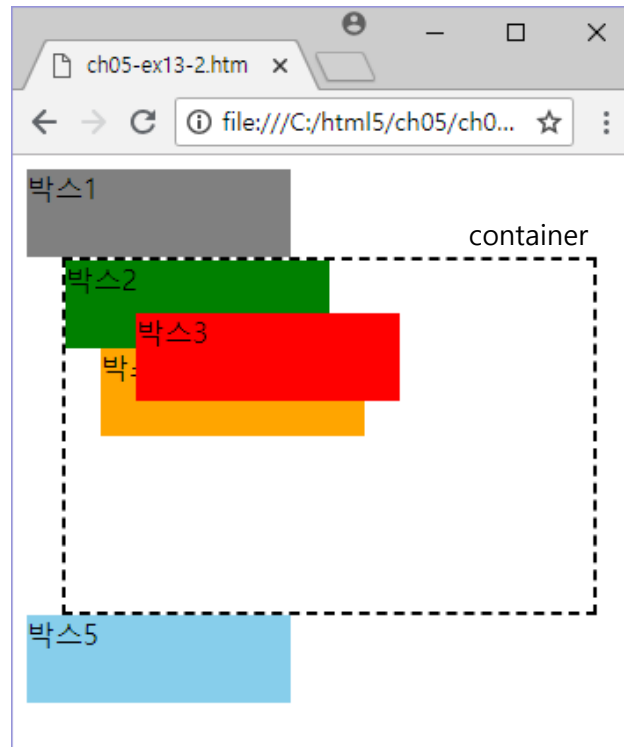
다) 절대 위치(absolute)

[예 5.13] 절대 배치 : 절대 위치 지정

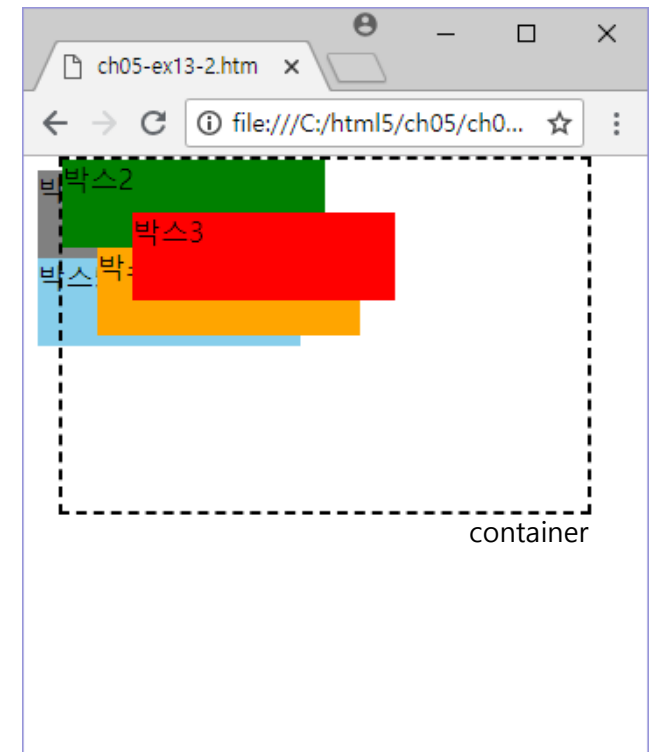
```
#b3 { position : absolute;  
        top: 30px; left: 30px; }
```



```
#b3 { position : absolute;  
        top: 30px; left: 30px; }  
#container { position: relative;  
        top : 0px; left: 10px; }
```



```
#b3 { position : absolute;  
        top: 30px; left: 30px; }  
#container { position: absolute;  
        top : 0px; left: 10px; }
```

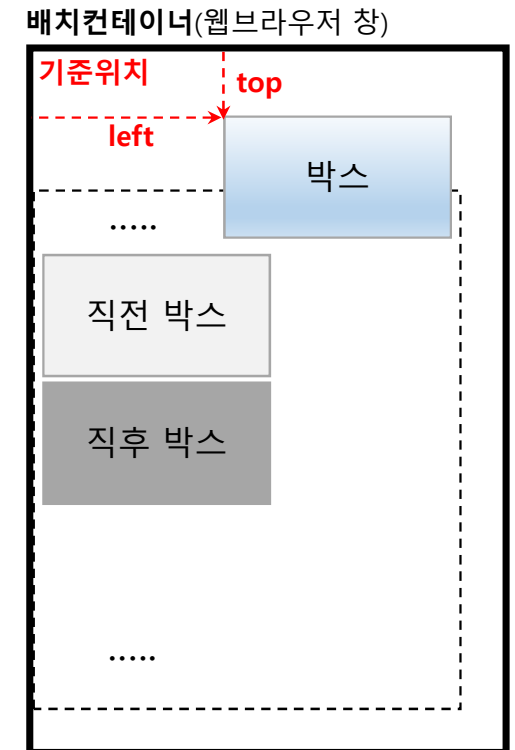


(4) 고정 배치(고정 위치) 지정 : { position: fixed; top: 길이값; left: 길이값; }

- position 속성값을 fixed로 지정하고, top, left 속성을 이용해서 위치를 지정함
 - 배치 공간 및 기준 위치 : 웹 브라우저의 현재 화면
 - top, left 속성값 : 기준위치에 대한 상대적 위치를 나타냄
- 원래의 기본 배치공간은 무시됨. 따라서, 뒤따르는 형제박스들의 기본 배치에 영향을 줌
- 박스를 고정위치로 배치하면, 화면 크기에 관계없이 항상 고정된 위치에 출력됨
- top, left 속성의 지정을 생략하면, 항상 화면의 좌측상단에 출력됨

```
<div id="container">  
  ....  
  <div>직전 박스</div>  
  <div>박스</div>  
  <div>직후 박스</div>  
  ....  
</div>
```

라) 고정위치(fixed)



[예 5.14] 고정위치 지정

```
#b3 { position : fixed; top: 30px; right:10px; }
```

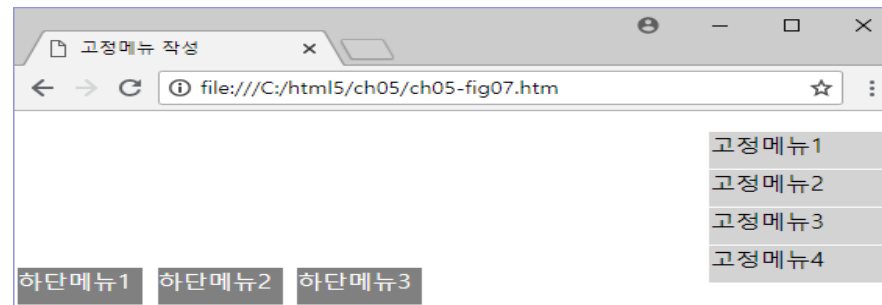
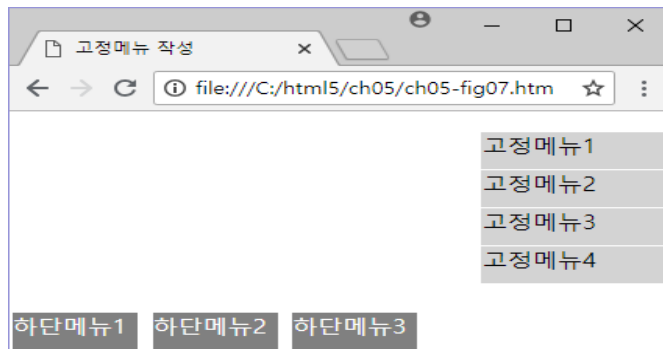


가) 화면크기를 작게 한 경우



나) 화면크기를 크게 한 경우

[그림 5.7] 고정 메뉴 표시



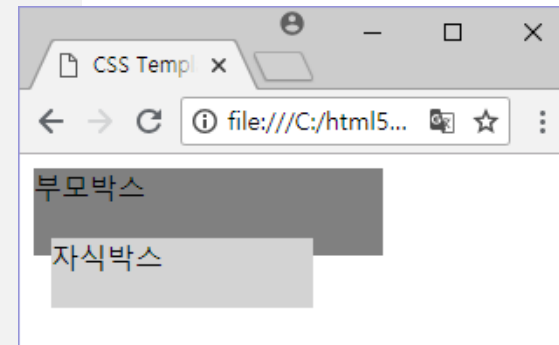
▪ 부모박스를 기준으로 자식박스를 배치하기

- ① 부모박스의 position을 relative로 지정함
- ② 자식박스를 절대 위치로 지정함. 그러면 자식박스는 부모박스를 기준으로 top, left 위치에 배치됨

(예 5.15)

```
<style>
  #parent { background: gray; width: 200px; height: 50px;
    position : relative;
  }
  #child { background: lightgray; width: 150px; height: 40px;
    position: absolute; top: 40px; left: 10px;
  }
</style>

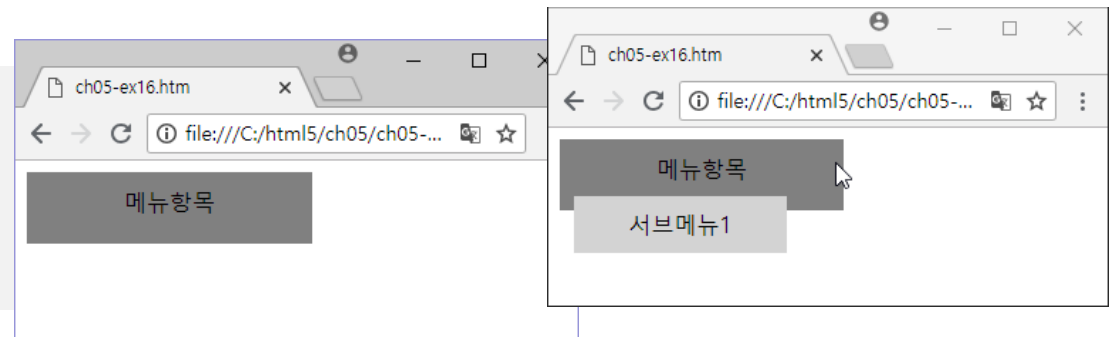
<div id="parent" > 부모박스
  <div id="child"> 자식박스 </div>
</div>
```



가) 부모박스를 기준으로 자식박스를 배치하기

```
<style>
  .... <!-- 가)의 CSS3 명령 생략 -->
  #child { display: none; }
  #parent:hover #child { display: block; }
</style>
```

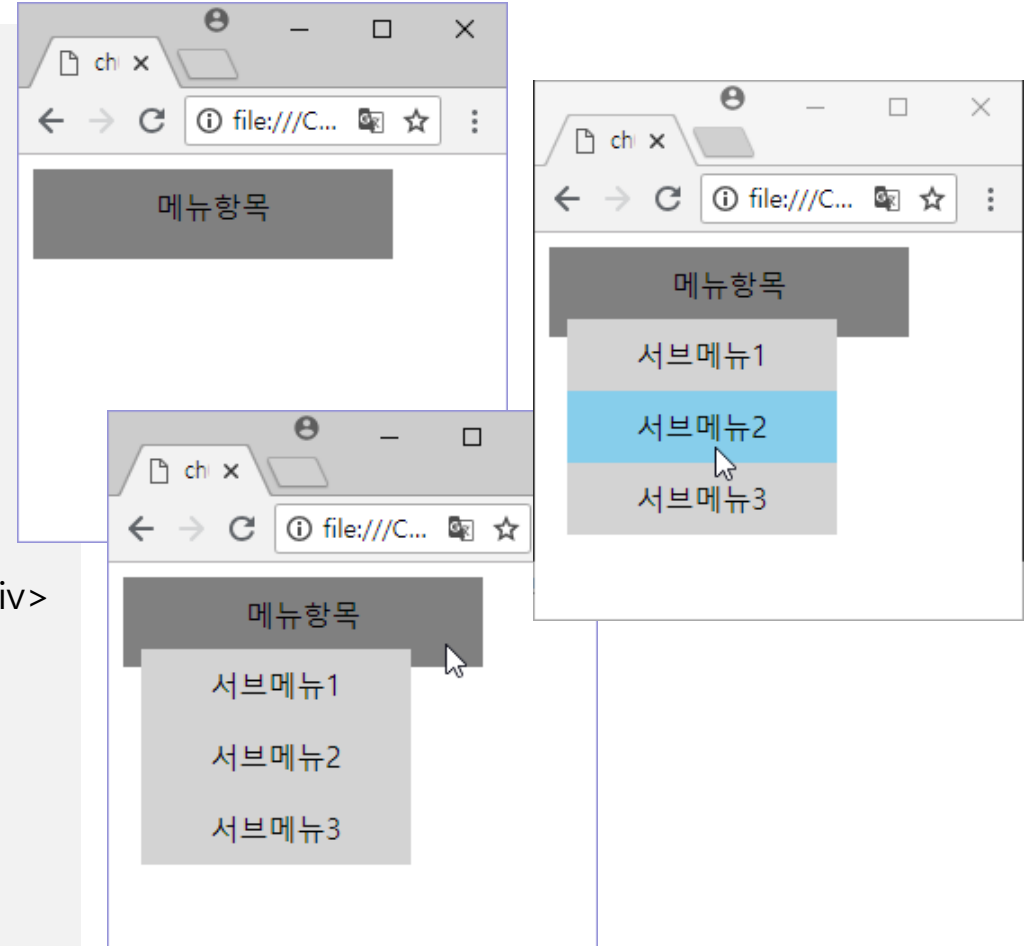
나) 자식박스를 마우스 위치에 따라 드롭다운 방식으로 나타내기



(그림 5.15-계속)

```
<style>
.....
.menu { background: lightgray;
        width:150px; height: 40px;
        text-align: center; line-height: 40px;
    }
    #child .menu:hover { background: skyblue; }
</style>

</head>
<body>
    <div id="parent" >
        <div style="text-align: center; padding: 10px;">메뉴항목</div>
        <div id="child">
            <div class="menu">서브메뉴1</div>
            <div class="menu">서브메뉴2</div>
            <div class="menu">서브메뉴3</div>
        </div>
    </div>
</body>
```

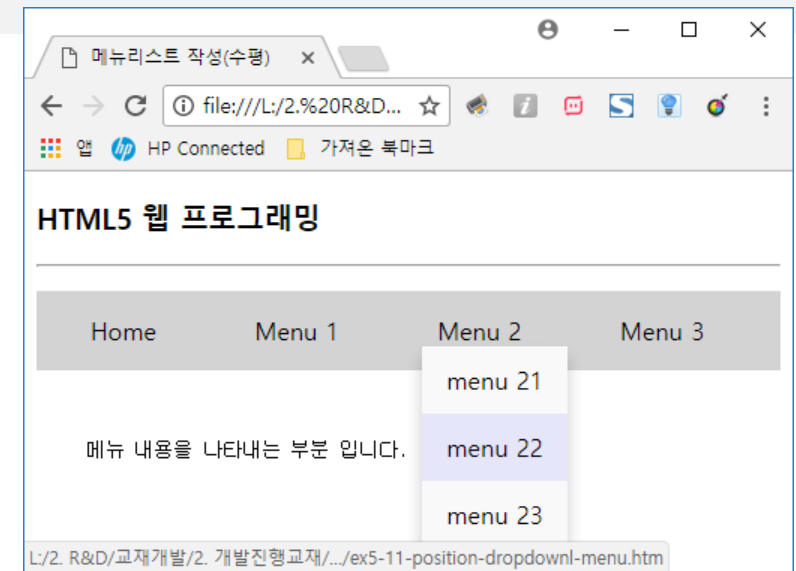


다) 드롭다운 방식 서브메뉴 리스트 작성

[예 5.16] 웹 페이지의 드롭다운 메뉴 리스트 작성

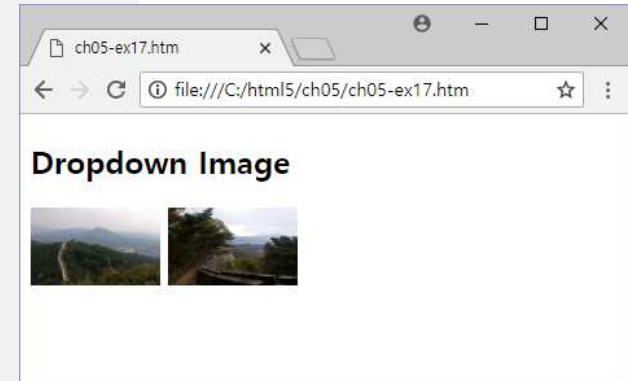
```
<body>
<h3> HTML5 웹 프로그래밍 </h3><hr>
<ul>
  <li class="dropdown">
    <a href="#home " >Home</a>
  </li>
  <li class="dropdown">
    ... 생략 ...
  </li>
  <li class="dropdown">
    <a href="" >Menu 2</a>
    <div class="dropdown-content">
      <a href="#">menu 21</a>
      <a href="#">menu 22</a>
      <a href="#">menu 23</a>
    </div>
  </li>
  <li class="dropdown">
    ... 생략 ...
  </li>
</ul>
<div style="height: auto;">
  <pre> ... 메뉴 내용을 나타내는 부분 입니다.
...</pre>
</div>
</body>
```

```
<style>
ul { background: lightgray; padding: 6px; }
li { display: inline; padding: 10px 20px; }
li a { padding: 5px 10px; { color: black; text-decoration: none; }
li a:hover { background: darkgray; }
li.dropdown { display: inline-block; }
.dropdown-content { display: none; position: absolute;
  background: ghostwhite; box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2); }
.dropdown-content a { display: block; padding: 12px 16px;
  text-decoration: none; text-align: left; color: black; }
.dropdown-content a:hover { background: lavender; }
.dropdown:hover .dropdown-content { display: block; }
</style>
```

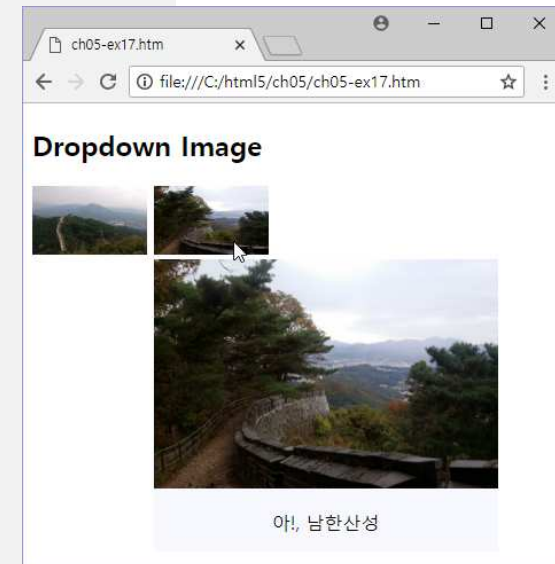


[예 5.17] 드롭다운 이미지 표현

```
<style>
.dropdown { position: relative; display: inline-block; }
.dropdown-content { display: none; position: absolute;
    background-color: ghostwhite;
}
#dropdown1:hover .dropdown-content { display: block; }
#dropdown2:hover .dropdown-content { display: block; }
.desc { padding: 15px; text-align: center; }
</style>
```



```
<h2>Dropdown Image</h2>
<div class="dropdown" id="dropdown1">
  
  <div class="dropdown-content">
    
    <div class="desc">아름다운 인왕산 길</div>
  </div>
</div>
<div class="dropdown" id="dropdown2">
  
  <div class="dropdown-content">
    
    <div class="desc">아!, 남한산성</div>
  </div>
</div>
```



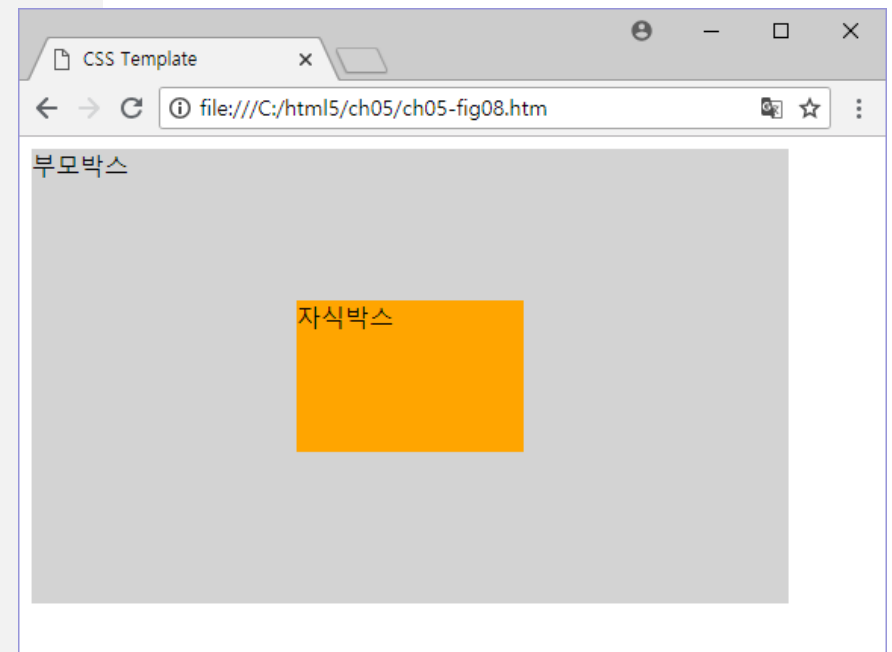
▪ 자식박스를 항상 부모박스의 중앙에 배치하기

- ❶ 부모박스의 width, height 속성을 지정하고, position 속성을 relative로 지정함
- ❷ 자식박스의 position 속성은 absolute로, left, top 속성은 각각 부모박스의 width, height 속성값의 50%로, margin-left, margin-top과 속성은 자신의 width, height 속성값의 -50%로 지정함

(그림 5.18)

```
<style>
  #parent { background: lightgray;
             width: 500px; height: 300px;
             position : relative;
          }
  #child { background: orange;
            width:150px; height: 100px;
            position: absolute;
            left: 250px; top: 150px;
            margin-left: -75px; margin-top: -50px;
          }
</style>

<div id="parent" > 부모박스
  <div id="child"> 자식박스 </div>
</div>
```

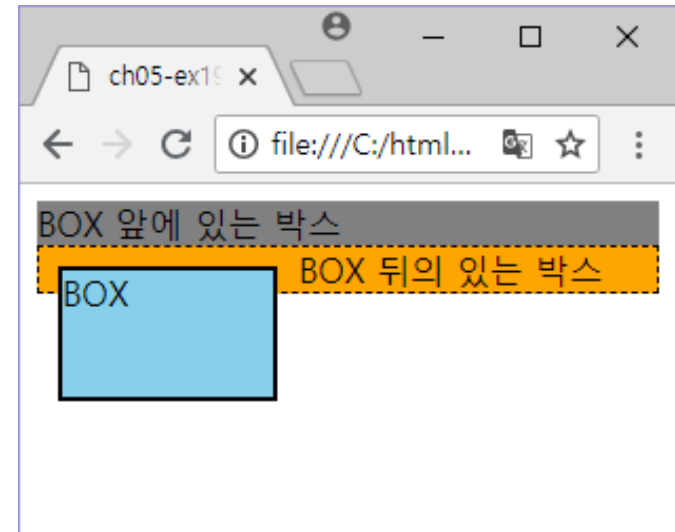


5.2.3 플로팅박스 : float, clear 속성

- float 속성 : 박스를 마치 공중에 떠 있는 것처럼 배치하는 것
 - ✓ 뒤따르는 박스들은 공중에 떠있는 플로팅 박스의 아래 부분으로 배치되고, 웹 브라우저 화면에는 마치 이것들을 위에서 본 것처럼 중복해서 출력됨
 - ✓ 플로팅박스 아래에 중복해서 출력되는 박스의 텍스트들은 중복 공간을 피해서 플로팅 박스의 바깥 부분에 출력됨
 - ✓ float : left; 플로팅박스는 화면 왼쪽에 배치되고, 아래의 텍스트들은 플로팅박스의 오른쪽 및 아래쪽 주변에 배치됨

```
<style>
#b1 { background: gray; }
#box { background: skyblue; border: 2px solid black;
      width: 100px; height: 60px; margin: 10px;
      float: left;
}
#b2 { background: orange; border: 1px dashed black;}
</style>

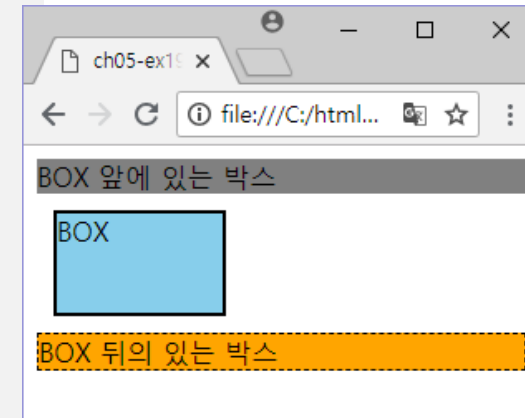
<div id="b1">BOX 앞에 있는 박스</div>
<div id="box"> BOX </div>
<div id="b2">BOX 뒤의 있는 박스</div>
```



[예 5.19] 플로팅박스의 특성

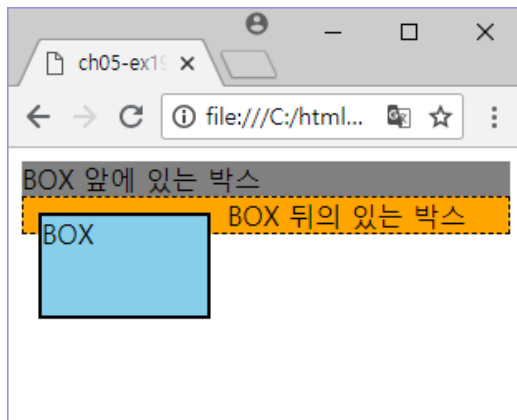
```
<style>
#b1 { background: gray; }
#box { background: skyblue; border: 2px solid black;
      width: 100px; height: 60px; margin: 10px; }
#b2 { background: orange; border: 1px dashed black;}
</style>
```

```
<div id="b1">BOX 앞에 있는 박스</div>
<div id="box"> BOX </div>
<div id="b2">BOX 뒤의 있는 박스</div>
```



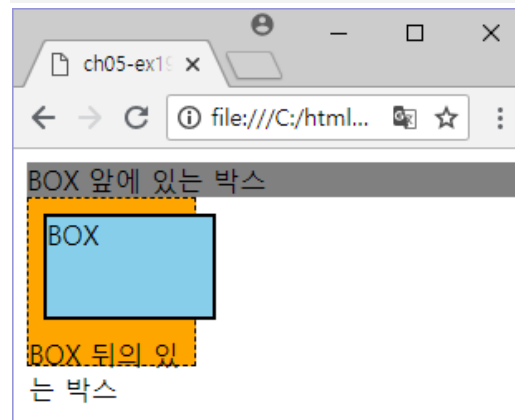
가) 문서 기본출력

```
#box { float: left; }
```



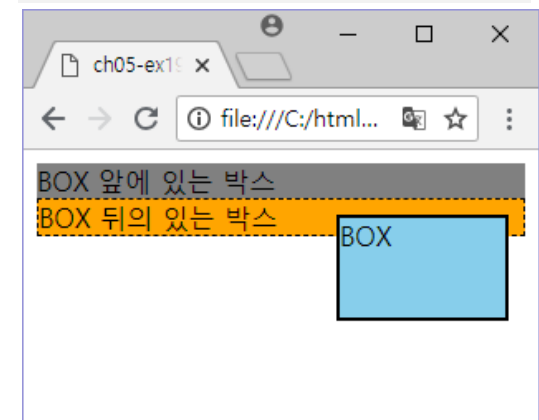
나) 왼쪽 플로팅박스출력(1)

```
#b2 { width:100px; height:100px;}
#box { float: left; }
```



다) 왼쪽 플로팅박스출력(2)

```
#box { float: right; }
```



라) 오른쪽 플로팅박스출력

- clear 속성
 - 플로팅박스 뒤의 형제박스들이 앞서 지정된 플로팅박스들과 중복되지 않게 배치할 때 사용함
 - { clear : left ; } // 선행하는 왼쪽 플로팅박스들과 중복되지 않게 배치됨
 - { clear : right ; } // 선행하는 오른쪽 플로팅박스들과 중복되지 않게 배치됨
 - { clear : both ; } // 선행하는 모든 플로팅박스들과 중복되지 않게 배치됨

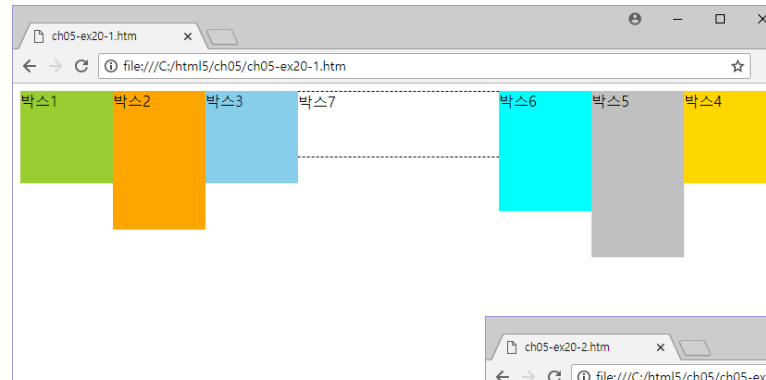
[표 5.5] float 속성의 주요값

속 성	속성값	의 미
float	left	플로팅박스는 화면 왼쪽에 배치됨
	right	플로팅박스는 화면 오른쪽에 배치됨
	none	(기본값) 플로팅박스로 지정하지 않음
clear	left	left로 지정된 앞의 플로팅박스들과 중복되지 않게 배치됨
	right	right로 지정된 앞의 플로팅박스들과 중복되지 않게 배치됨
	both	앞의 모든 플로팅박스들과 중복되지 않게 배치됨

[예 5.20] clear 속성

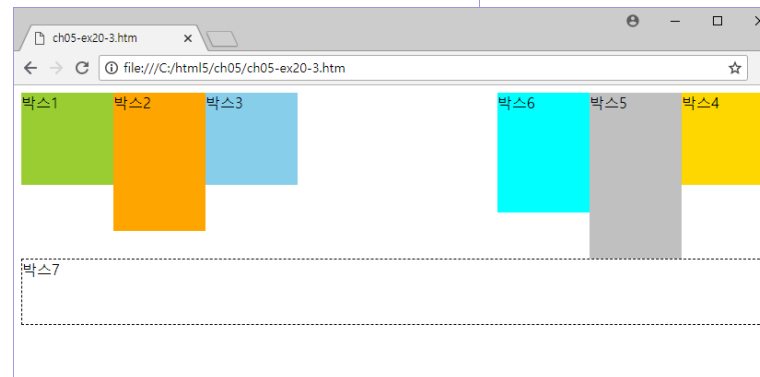
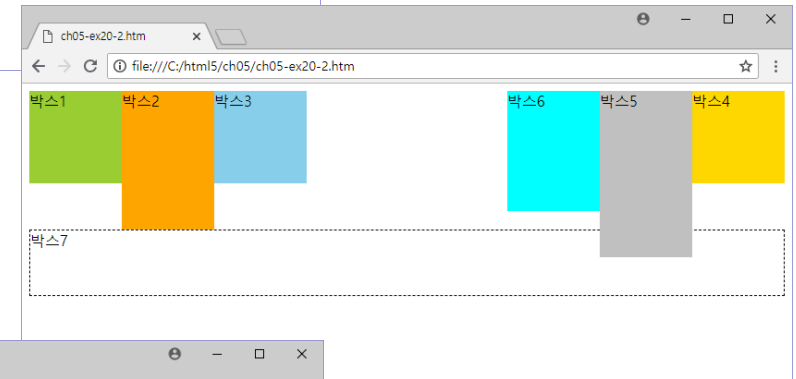
```
<style>
.float-left { float: left; width: 100px; }
.float-right { float: right; width: 100px; }
div#b1 { background: yellowgreen; height:100px; }
div#b2 { background: orange; height:150px; }
div#b3 { background: skyblue; height:100px; }
div#b4 { background: gold; height:100px; }
div#b5 { background: silver; height:180px; }
div#b6 { background: aqua; height:130px; }
div#b7 { height:70px; border: 1px dashed black; }
</style>
```

```
<div class="float-left" id="b1"> 박스1 </div>
<div class="float-left" id="b2"> 박스2 </div>
<div class="float-left" id="b3"> 박스3 </div>
<div class="float-right" id="b4"> 박스4 </div>
<div class="float-right" id="b5"> 박스5 </div>
<div class="float-right" id="b6"> 박스6 </div>
<div id="b7"> 박스 7 </div>
```



가) 박스1~3: {float: left} ,
박스 4~6: {float: right} }

나) #b7 { clear : left; }



다) #b7 { clear : right; }

[예 5.21] 플로팅 이미지를 이용한 문서 출력

```
<style> img { margin: 10px 10px; } </style>
```

```
<body>
```

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

```
<p>  Lorem ipsum dolor sit amet,  
..... Cras ac leo purus. Mauris quis diam velit.</p>
```

```
</body>
```



가) img { float : left } 인 경우

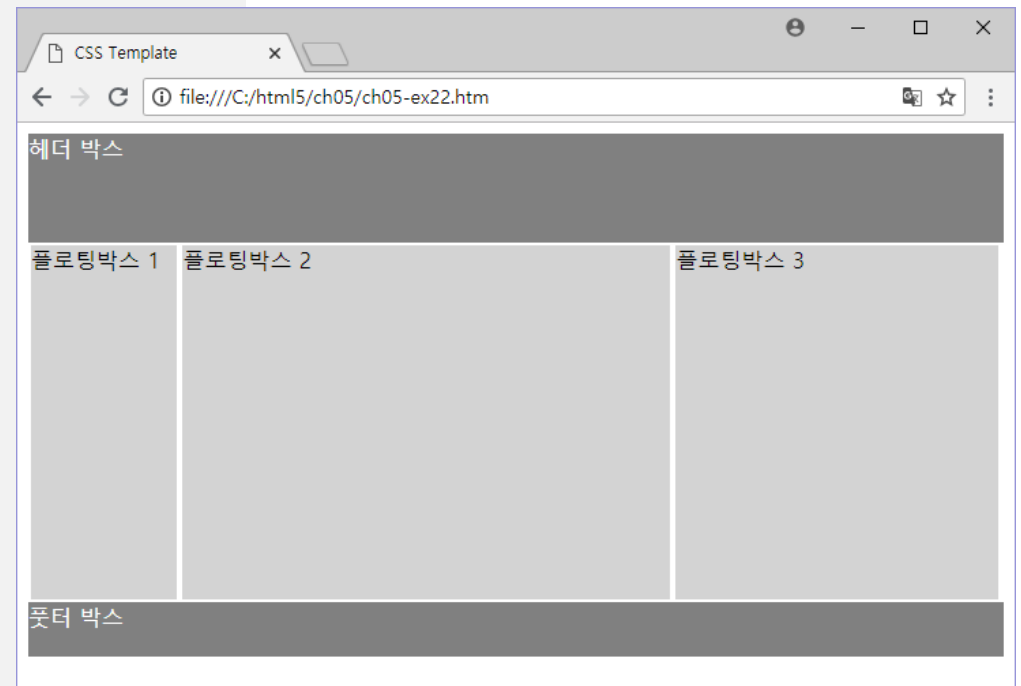


나) img { float : right } 인 경우

[예 5.22] float 속성을 이용한 웹 페이지 레이아웃 구성

```
<style>
.box { background: gray; color : white; }
.floating { background: lightgray; margin : 2px; float : left ; }
body { height : 400px; }
#b1 { height: 20%; }
#b2 { height: 65%; }
#f1 { width: 15%; height: 100%; }
#f2 { width: 50%; height: 100%; }
#f3 { width: 33%; height: 100%; }
#b3 { clear: left; height: 10%; }
</style>

<div class="box" id="b1" >헤더 박스</div>
<div id="b2">
  <div class="floating" id="f1">플로팅박스 1</div>
  <div class="floating" id="f2">플로팅박스 2</div>
  <div class="floating" id="f3">플로팅박스 3</div>
</div>
<div class="box" id="b3" >풋터 박스</div>
```



5.2.4 박스 표시 관련 속성들

- 박스들의 화면 출력할 때 배치 방법(배치위치)
 - display : 박스의 출력 유형 지정
 - position : 박스의 화면 배치 방법(top, left 속성)
 - float : 박스의 공중 배치 (뒤따라는 박스들과 중복 배치 효과)
 - clear : float 박스의 영향 배제 지정
- 배치된 박스들의 화면 표시 방법 : overflow, z-index, opacity, visibility 속성
 - overflow : 박스 내용의 출력 방법
 - z-index : 중복 배치된 박스들의 화면 표시 순서 지정
 - opacity : 박스 배경색 또는 배경 이미지의 표시 투명도 지정
 - visibility : 박스의 브라우저 화면 표시 여부 지정

(1) overflow 속성

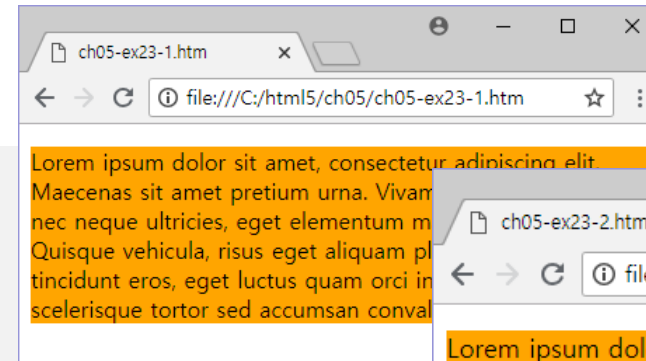
- 크기(width, height 속성)를 지정한 박스의 내용 출력 방법을 지정함

[표 5.6] overflow 속성의 주요 값

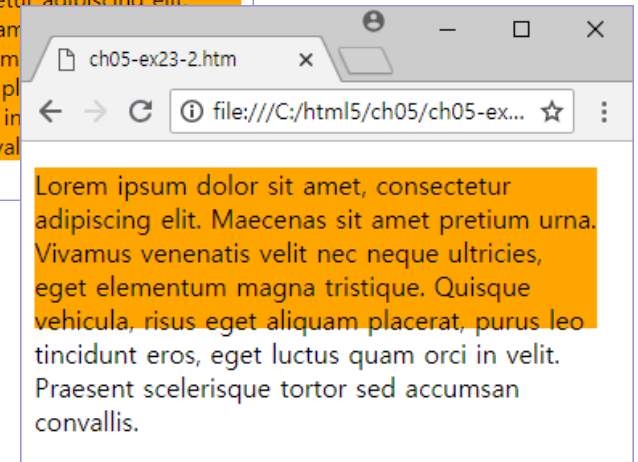
속성값	의 미
visible	(기본값) 박스의 아래쪽으로 초과된 내용들이 나타냄
hidden	박스 크기를 초과하는 내용들은 나타내지 않음
scroll	박스 스크롤-바를 표시해서, 초과되는 내용들을 볼 수 있게 함
auto	내용이 박스 크기를 초과되는 경우만 스크롤-바를 제공함

[예 5.23] overflow 속성 지정에 따른 출력 비교

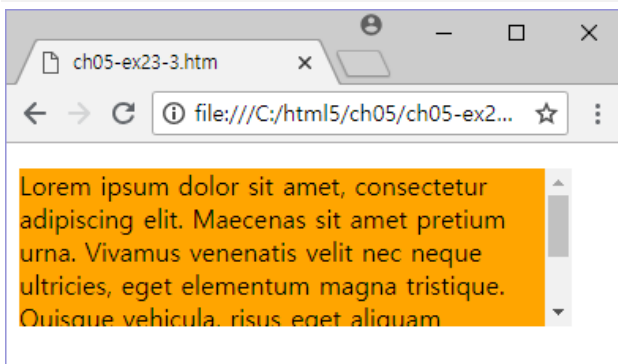
```
<style>
  p { background : orange; }
</style>
...
<body>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Maecenas sit amet pretium urna. Vivamus venenatis velit nec
neque ultricies, eget elementum magna tristique. Quisque
vehicula, risus eget aliquam placerat, purus leo tincidunt eros,
eget luctus quam orci in velit. Praesent scelerisque tortor sed
accumsan convallis.</p>
</body>
```



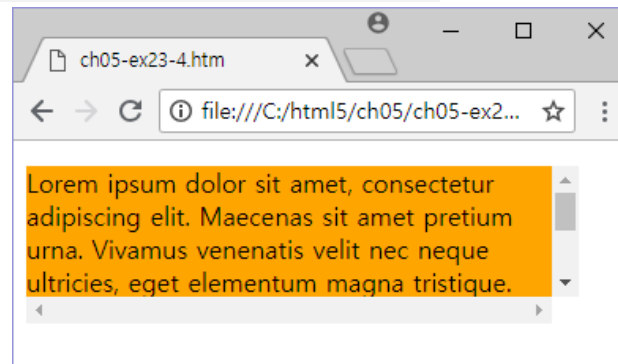
가) 기본출력



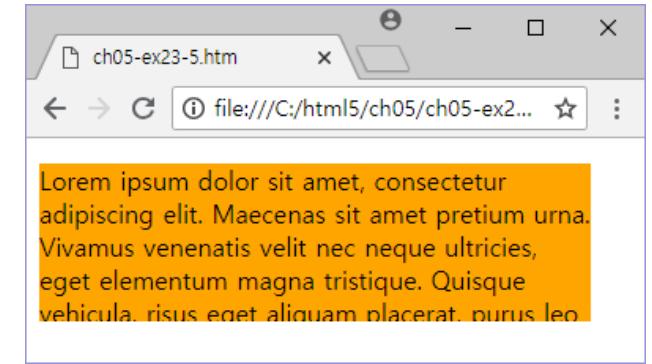
나) p { width: 350px; height: 100px; }



다) p { width: 350px; height: 100px; **overflow: auto;** }



라) p { width: 350px; height: 100px; **overflow: scroll;** }

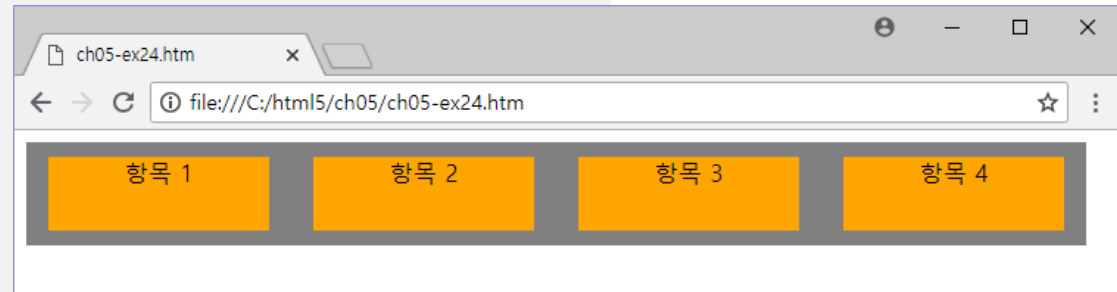


마) p { width: 350px; height: 100px; **overflow: hidden;** }

[예 5.24] float와 overflow 속성을 이용한 자식박스들의 수평 배치

```
<style>
  #parent { background: gray; text-align: center; overflow: hidden;
            box-sizing: border-box; width: 720px; height: 70px;
  }
  .child { background: orange ; float: left; box-sizing: border-box;
           width: 150px; height: 50px; margin: 10px 15px;
  }
</style>

<div id="parent" >
  <div class="child" >항목 1</div>
  <div class="child" >항목 2</div>
  <div class="child" >항목 3</div>
  <div class="child" >항목 4</div>
</div>
```



(2) z-index 속성

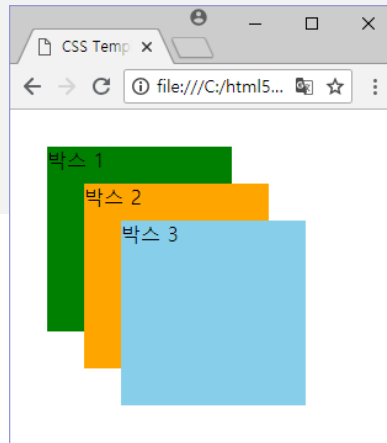
- 박스들이 같은 위치에 중복해서 배치될 때 웹 브라우저 화면에 나타나는 순서를 지정함
- z-index 속성값(정수)이 클수록 화면 출력 우선순위가 높다

[예 5.25]

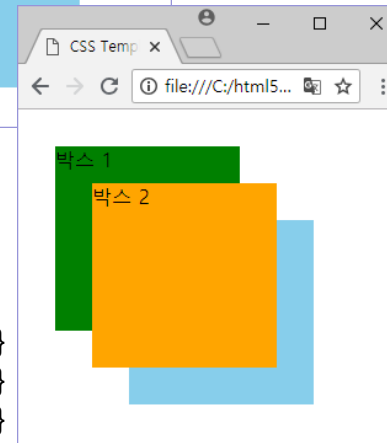
```
<style>
  div { width:150px ; height: 150px; position : absolute; }
  #b1 { background: green ; top: 30px; left : 30px; }
  #b2 { background: orange ; top: 60px; left : 60px; }
  #b3 { background: skyblue ; top: 90px; left : 90px; }
</style>
```

```
<body>
  <div id="b1">박스 1</div>
  <div id="b2">박스 2</div>
  <div id="b3">박스 3</div>
</body>
```

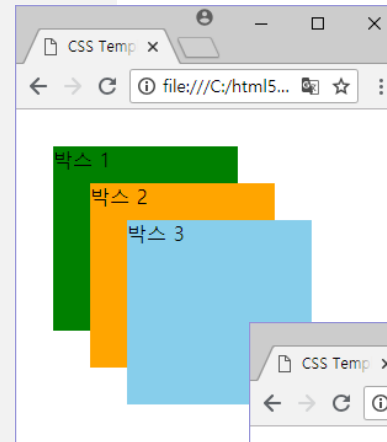
```
#b1 { z-index: 1; }
#b2 { z-index: 3; }
#b3 { z-index: 5; }
```



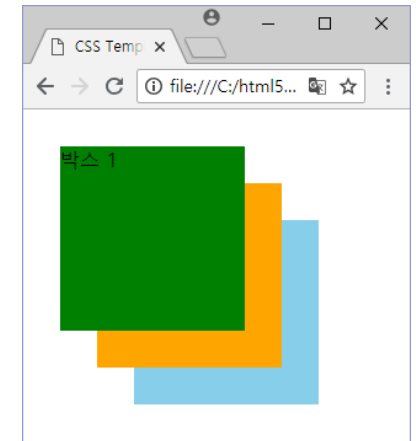
```
#b1 { z-index: 1; }
#b2 { z-index: 5; }
#b3 { z-index: 2; }
```



기본 출력



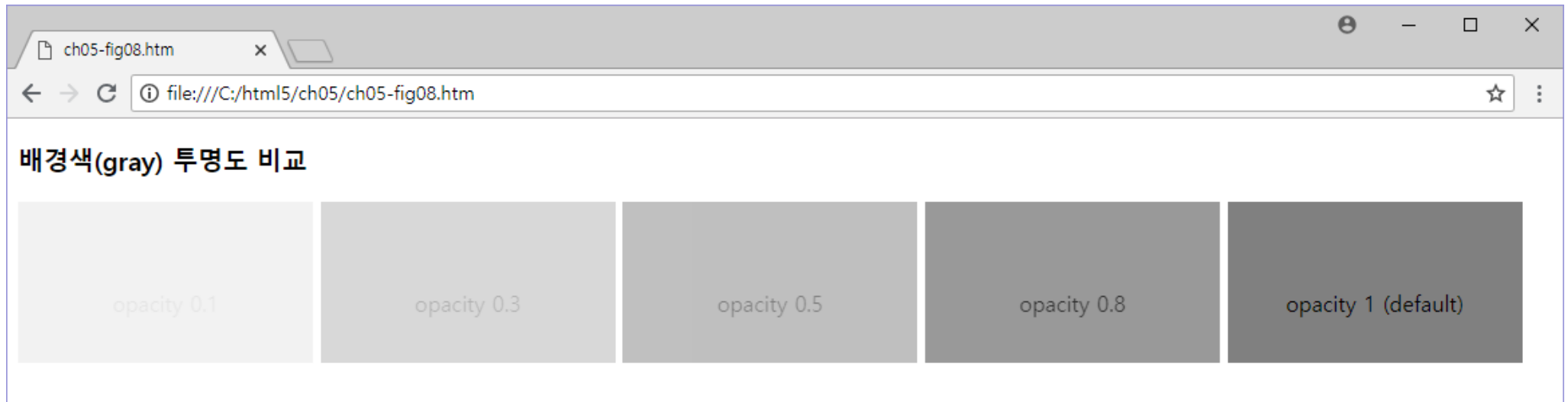
```
#b1 { z-index: 50; }
#b2 { z-index: 30; }
#b3 { z-index: 10; }
```



(3) opacity 속성

- 배경색 또는 배경 이미지를 가진 박스의 투명도를 지정함
- 박스들이 중복 배치될 때, 아래쪽 박스의 화면 선명도는 위쪽 박스의 투명도에 영향을 받음
- opacity 속성값 : 0.0 (완전투명) ~ 1.0 (완전불투명)

[그림 5.8] 회색(gray) 박스의 투명도 비교



[그림 5.9] 이미지 박스의 투명도 비교



(4) visibility 속성

- 박스의 브라우저 화면 표시 여부를 지정함
- visibility 속성값 : visible(기본값), hidden, collapse (표 작성에서 사용함)
- {display: none;} // 박스를 화면에 배치하지 않음(박스의 배치공간 자체가 없음)
- {visibility: hidden;} // 박스를 화면에 배치하지만 출력은 하지 않음

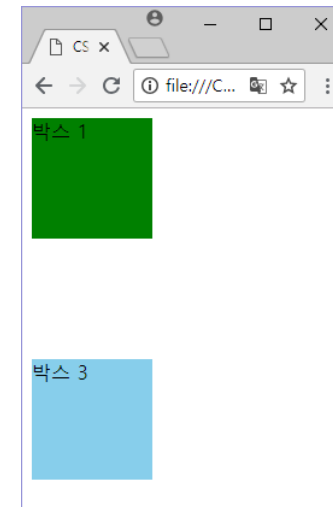
[예 5.26]

```
<style>
  div { width:100px ; height: 100px; }
  #b1 { background: green ; top: 30px; left : 30px; }
  #b2 { background: orange ; top: 60px; left : 60px; }
  #b3 { background: skyblue ; top: 90px; left : 90px; }
</style>
```

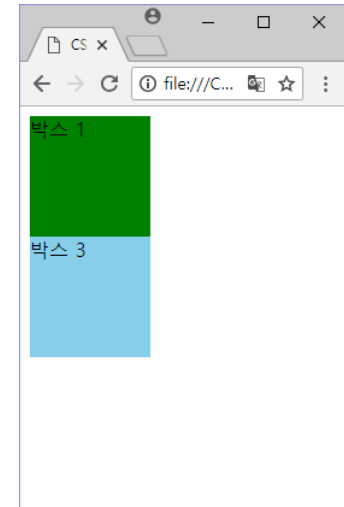
```
<div id="b1">박스 1</div>
<div id="b2">박스 2</div>
<div id="b3">박스 3</div>
```



기본 출력



#b2 { visibility: hidden; }

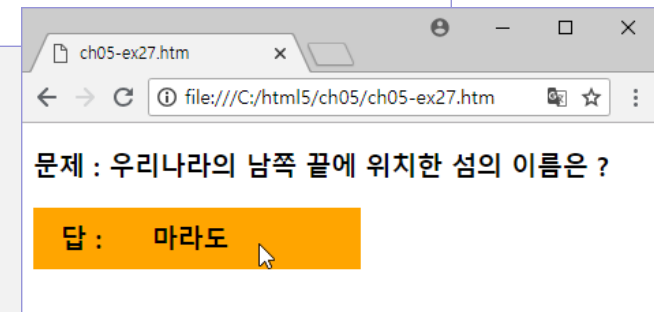
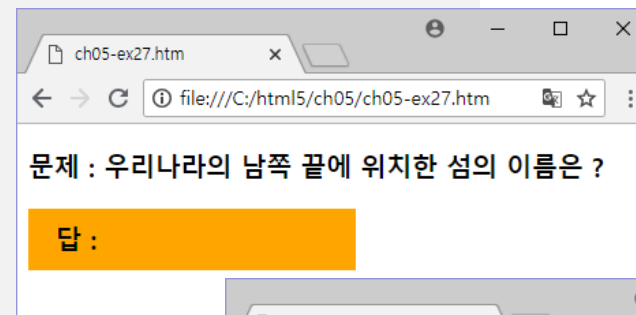


#b2 { display : none; }

[예 5.27] visibility 속성의 활용

```
<style>
#in { display : inline-block; padding : 0px 30px;
      visibility : hidden;
}
#out { background : orange;
      width : 200px; padding: 10px 20px;
}
#out:hover #in { visibility: visible; }
</style>
```

```
<h3> 문제 : 우리나라의 남쪽 끝에 위치한 섬의 이름은 ? </h3>
<h3 id="out"> 답 : <span id="in"> 마라도 </span> </h3>
```

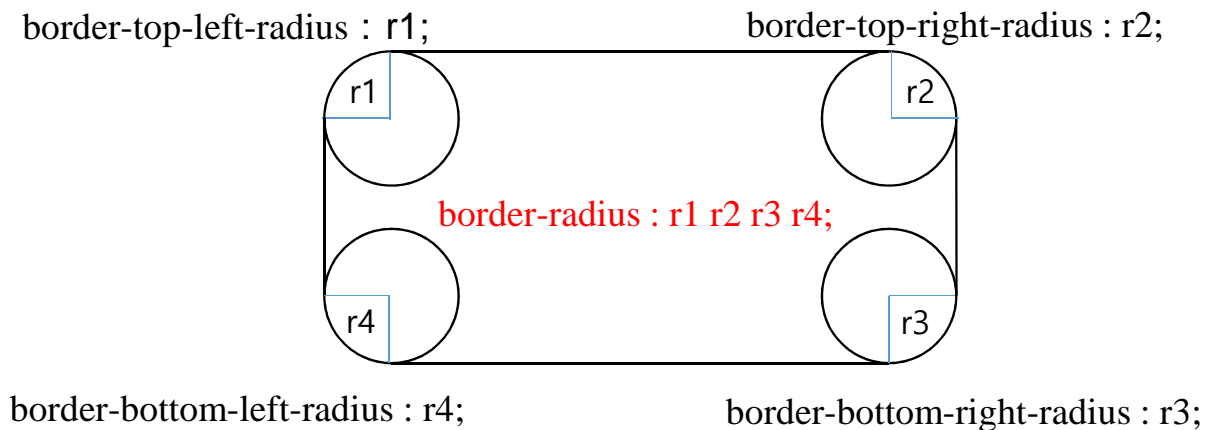


5.3 CSS3의 고급 기능들

기능	관련 속성	설명
모서리 라운딩 효과	border-radius	박스 모서리의 라운딩 정도를 지정함
그림자 효과	box-shadow, text-shadow	박스, 텍스트의 그림자 효과를 지정함
박스 변환 효과	transform	박스의 이동/회전/확대축소/왜곡변환을 지정함 translate(), scale(), skew(), rotate() 등 속성값 함수 제공
속성 전환 효과	transition	속성값 전환 효과(애니메이션)를 지정함
키 프레임 애니메이션	animation-name/ -duration, -iteration/ -count	@keyframes으로 지정한 애니메이션 효과 지정함
그라디언트 효과	background - color	색상의 점진적 변화 효과를 지정함 linear-gradient(), radial-gradient() 등 속성값 함수 제공

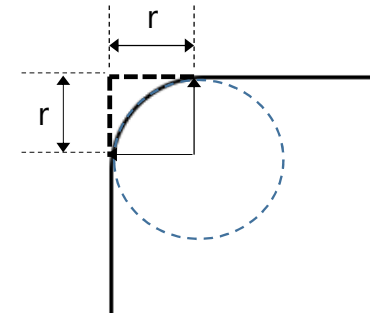
5.3.1 박스의 라운딩 효과 : border-radius 속성

- 박스 모서리의 라운딩 효과 지정가능(다양한 모양 표시)
- 모서리별 세부속성들 또는 border-radius 단축속성 이용 지정

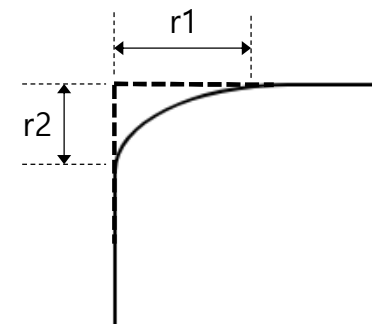


- ❶ border-radius: 20px 40px 60px 80px;
- ❷ border-radius: 30px; ("border-radius: 30px 30px 30px 30px;"과 같은 표현임)
- ❸ border-radius: 20px 50px;("border-radius: 20px 50px 20px 50px;"과 같은 표현임)
- ❹ border-top-left-radius: 20px 20px;
- ❺ border-radius: 20px/50px 20px 20px 20px ;
 ⇔ border-top-left-radius: 20px 50px ; border-top-right-radius: 20px ;
 border-bottom-right-radius: 20px ; border-bottom-left-radius: 20px ;

[그림 5.11]
모서리 라운딩 정도 지정



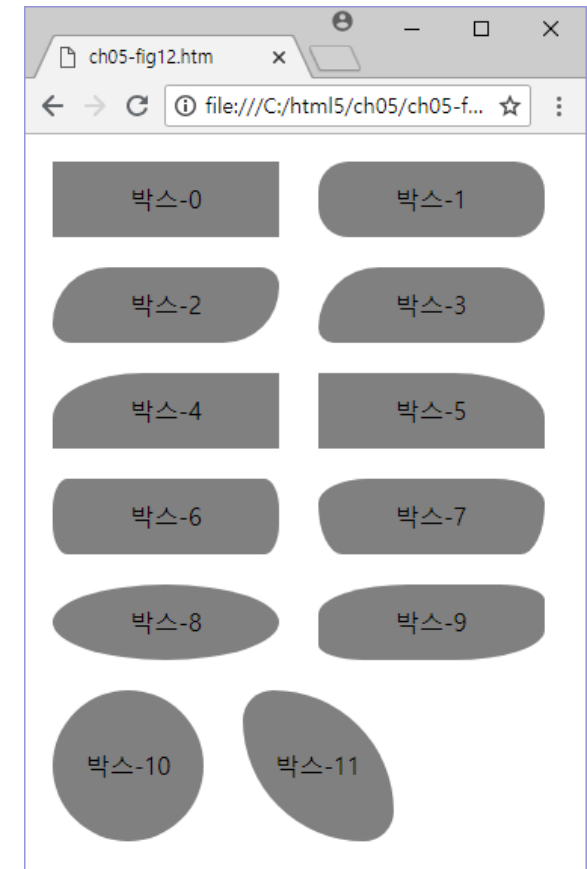
가) border-top-left-radius: r ;



나) border-top-left-radius: r1, r2 ;

[예 5.28] 라운딩 효과를 이용한 박스의 다양한 출력

```
<style>
div { width:150px; height: 50px; background: gray;
      text-align: center; line-height: 50px; margin: 10px;
      display: inline-block;
    }
#b1 { border-radius: 20px; }      #b2 { border-radius: 60px 20px; }
#b3 { border-radius: 80px 60px 40px 20px; }
#b4 { border-top-left-radius: 60px 30px; }
#b5 { border-top-right-radius: 60px 30px; }
#b6 { border-radius: 10px/20px; }
#b7 { border-top-left-radius: 60px 30px;
      border-bottom-right-radius: 30px 60px;
      border-top-right-radius: 60px 30px;
      border-bottom-left-radius: 30px 60px;
    }
#b8 { border-radius: 50%; }      #b9 { border-radius: 40% 20%; }
#b10 { width: 100px; height: 100px; line-height: 100px;
       border-radius: 50%;
    }
#b11 { width: 100px; height: 100px; line-height: 100px;
       border-radius: 20% 80%;
    }
</style>
```



```
<div> 박스-0 </div> <div id="b1"> 박스-1 </div>
<div id="b2"> 박스-2 </div> <div id="b3"> 박스-3 </div>
<div id="b4"> 박스-4 </div> <div id="b5"> 박스-5 </div>
<div id="b6"> 박스-6 </div> <div id="b7"> 박스-7 </div>
<div id="b8"> 박스-8 </div> <div id="b9"> 박스-9 </div>
<div id="b10"> 박스-10 </div> <div id="b11"> 박스-11 </div>
```

[그림 5.12] [예 5.21] 가)의 원형 이미지 출력(`img {border-radius: 50%;}`)



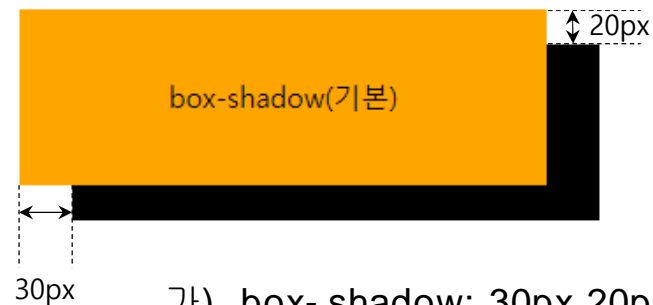
5.3.2 그림자 효과 : box-shadow/text-shadow 속성

- box-shadow/text-shadow : 각각 박스와 텍스트의 그림자 효과를 나타냄
- 그림자 효과 : 4개 속성값("수평-오프셋값, 수직-오프셋값, 번짐정도, 색") 지정

[표 5.8] box-shadow/text-shadow 속성값

속성 의미	필수 여부 및 예	속성값 설명
수평-오프셋, 수직-오프셋	(필수) 5px 5px	그림자의 수평, 수직 시작 위치
번짐 정도, 크기	(선택) 10px	그림자의 번짐 정도와 그림자 크기
색	(선택) skyblue	그림자 색상

[그림 5.13]



가) box-shadow: 30px 20px ;



나) text-shadow: 10px 5px;

[예 5.29] 박스의 다양한 그림자 효과

```
<style>
div { background: orange; width: 200px; height: 70px;
margin: 30px; line-height: 70px; text-align: center;
display: inline-block;
}
#b1 { box-shadow: 30px 20px ; }
#b2 { box-shadow: 10px 10px ; }
#b3 { box-shadow: -10px 10px ; }
#b4 { box-shadow: 10px -10px ; }
#b5 { box-shadow: -10px -10px ; }
#b6 { box-shadow: 10px 10px skyblue ; }
#b7 { box-shadow: 10px 10px 5px ; }
#b8 { box-shadow: 10px 10px 5px skyblue }
#b9 { box-shadow: 5px 5px 10px black,
0 0 30px blue, 0 0 10px skyblue;
}
</style>
```



```
<div id="b1"> 그림자 박스(기본) </div>
<div id="b2"> 그림자 박스(기본:우하) </div>
<div id="b3"> 그림자 박스(방향:좌하) </div>
<div id="b4"> 그림자 박스(방향:우상) </div>
<div id="b5"> 그림자 박스(방향:좌상) </div>
<div id="b6"> 그림자 박스(색상지정) </div>
<div id="b7"> 그림자 박스(번짐지정) </div>
<div id="b8"> 그림자 박스(색상/번짐) </div>
<div id="b9"> 그림자 박스(다중그림자) </div>
```

[예 5.30] 텍스트의 다양한 그림자 효과

```
<style>
  h1 { width: 250px; height: 50px; margin:10px; line-height: 50px; text-align: center; display:
inline-block; }
  #b1 { text-shadow: 2px 2px; }
  #b2 { text-shadow: 2px 2px blue; }
  #b3 { text-shadow: 2px 2px 5px blue; }
  #b4 { text-shadow: 0 0 3px #0000FF, 0 0 5px #FF0000; }
  #b5 { text-shadow: 1px 1px 2px black, 0 0 25px blue, 0 0 5px darkblue; color: white; }
  #b6 { text-shadow: -2px 0 black, 0 2px black, 2px 0 black, 0 -2px black; color: orange; }
  #b7 { box-shadow: -10px -10px; }
  #b8 { box-shadow: -10px -10px;
      text-shadow: 2px 2px 5px blue;
    }
</style>

<h1 id="b1"> Text-Shadow </h1>
<h1 id="b2"> Text-Shadow </h1>
<h1 id="b3"> Text-Shadow </h1>
<h1 id="b4"> Text-Shadow </h1>
<h1 id="b5"> Text-Shadow </h1>
<h1 id="b6"> Text-Shadow </h1>
<h1 id="b7"> Text-Shadow </h1>
<h1 id="b8"> Text-Shadow </h1>
```



5.3.3 박스의 전환과 변환 효과 : transition, transform 속성

(1) 박스 전환 효과 : transition 속성

- 속성값이 점진적으로 변하는 과정을 나타내서 간단한 애니메이션 효과를 나타냄
- 속성값 지정 : 세부속성별 또는 transition 단축속성 지정
 - 대상속성(transition-property) : width, background-color 등
 - 전환시간(transition-duration) : 기본값(0), second 단위
 - 전환시간 함수(transition-timing-function) : 전환 과정의 진행속도를 나타냄
- 가상클래스 ":hover"와 함께 지정하면, 마우스가 해당 박스 위에 있을 때 애니메이션 효과를 지정할 수 있음

시간 커브 예약어	의 미
ease	처음은 천천히, 중간은 빠르게, 마지막은 천천히, cubic-bezier(0.25,0.1,0.25,1)
linear	처음부터 끝까지 같은 속도, cubic-bezier(0,0,1,1)
ease-in	처음에만 천천히, cubic-bezier(0.42,0,1,1)
ease-out	마지막에만 천천히, cubic-bezier(0,0,0.58,1)
ease-in-out	처음과 마지막에만 천천히, cubic-bezier(0.42,0,0.58,1)

transition 단축표현 예

```
div { width: 200px;
      height: 50px;
      background: skyblue;
      transition: width 5s background 5s ease-in-out;
}
div:hover { width : 500px; background : green; }
```

[표 5.9] 전환 효과의 전환시간함수 예약어

[예 5.31] width 속성을 이용한 박스전환 효과

- 박스 Level에 따라 가로 길이(width)가 다르게 늘어남

style>

```
div { background: gray; margin: 5px; width:100px;  
      height: 30px; line-height: 30px;  
      text-align: center;  
      transition: width 10s;  
      transition-delay: 1s;
```

```
}  
#b1:hover { width: 150px; }  
#b2:hover { width: 200px; }  
#b3:hover { width: 300px; }  
#b4:hover { width: 400px; }  
#b5:hover { width: 500px; }
```

</style>

마우스를 박스로 가져가면, 길이가 늘어납니다.

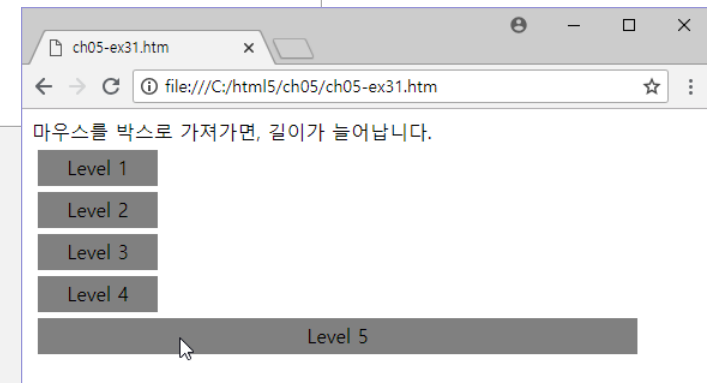
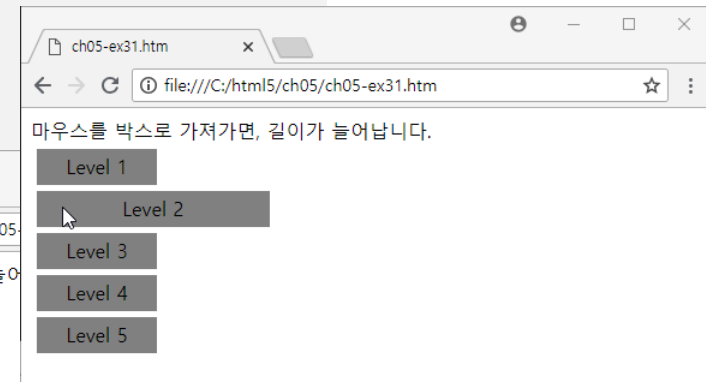
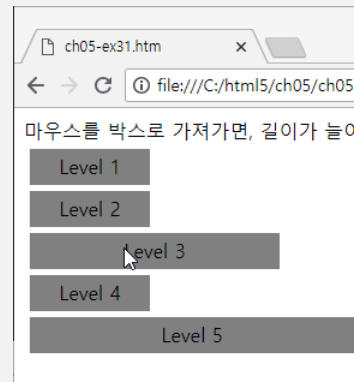
<div id="b1">Level 1</div>

<div id="b2">Level 2</div>

<div id="b3">Level 3</div>

<div id="b4">Level 4</div>

<div id="b5">Level 5</div>



(2) 박스 변환 효과 : transform 속성

- 박스의 회전(rotation), 왜곡 (skewing), 확대/축소(scaling), 평행이동(translation) 효과 등을 나타냄
- CSS3 표준 : 2차원/ 3차원의 다양한 변환함수들 제공함

변환 효과	변환 함수	설명
이동	translate(x,y)	x축, y축 방향으로 각각 x, y 만큼 이동함
	translateX(n), translateY(n)	각각 x축, y축 방향으로 n 만큼 이동함
확대/축소	scale(w,h)	폭, 높이를 각각 w, h 배 변경함
	scaleX(n), scaleY(n)	각각 폭(width), 높이(height)를 n 배 변경함
왜곡	skew(xdeg, ydeg)	x축, y축 기준으로 각각 xdeg, ydeg 기울임
	skewX(xdeg), skewY(ydeg)	각각 x축, y축 기준으로 xdeg, ydeg 기울임
회전	rotate(rdeg)	rdeg 만큼 회전함(- : 시계방향, + : 반시계방향)

```
❶ img { transform: translate (50px, 100px); }  
❷ img { transform: skew(-20deg); } // 시계방향 10도 왜곡시킴  
❸ img { transform: rotate(30deg); } // 반시계방향 20도 회전  
❹ img { transform: translate (50px, 100px) skew(-20deg) rotate(30deg); }
```

- 가상클래스 ":hover"와 함께 지정하면, 마우스가 해당 박스 위에 있을 때 애니메이션 효과를 지정할 수 있음

[예 5.32] 이미지 이동, 회전, 왜곡, 확대 축소 표현

```
<style>
#img1 { margin : 10px; transform : translate(10px, 50px) ; }
#img2 { margin : 10px; transform : rotate(-20deg); }
#img3 { margin : 10px; transform : skew(-10deg, -10deg); }
#img4 { margin : 10px; transform : scale(0.7, 0.7); }
#img5 { margin : 10px; transform : scaleX(1.5); }
</style>
```

기본 :
이동 :
회전 :

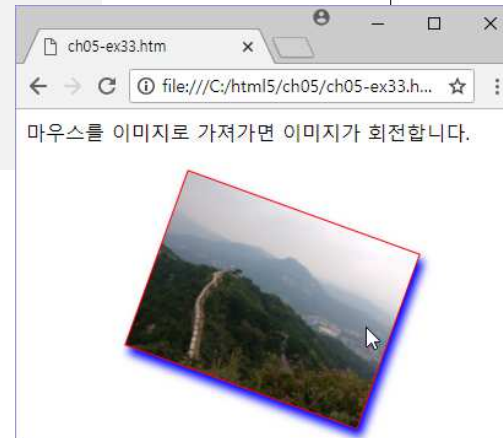
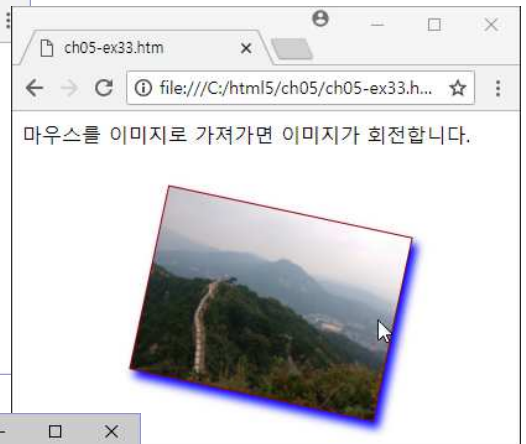
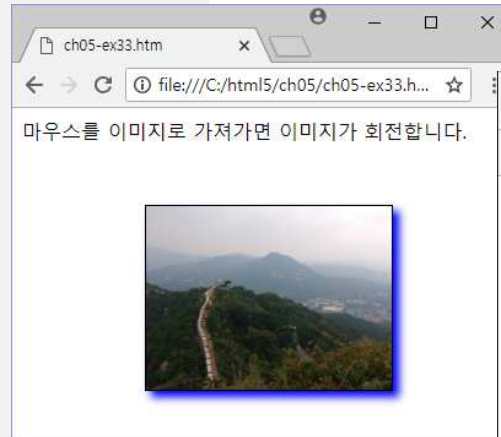
왜곡 :
축소 :
확대 :



[예 5.33] 가상 클래스 :hover와 rotate()를 이용한 이미지 변환

```
<style>
#img1 { margin-left : 100px; margin-top: 50px ;
border : 1px solid black;
box-shadow : 5px 5px 10px blue ;
transition : all 10s ease ;
}
#img1:hover {
border : 1px solid red;
box-shadow : 5px 5px 10px blue ;
transform : rotate(20deg);
}
</style>
```

마우스를 이미지로 가져가면 이미지가 회전합니다.



5.3.4 애니메이션 효과 : @keyframes

- 키 프레임(key frame)
애니메이션의 시작장면(상태)과 중간 주요 장면들, 마지막 장면 등과 같이 애니메이션 진행 과정에서 나타나는 중요 장면(key frame)들을 의미함
- 키 프레임 애니메이션
미리 정의한 키 프레임들을 자연스럽게 연결해서 애니메이션 기능을 나타냄

시점별로 CSS 스타일들을 지정해서 키 프레임들을 정의함

- ✓ 0%(from) : 시작시점
- ✓ 1~99 % : 중간시점
- ✓ 100%(to) : 종료시점

```
@frames 키 프레임명 {  
  0% { CSS3 스타일 } /* 애니메이션 시작 시점의 키 프레임 지정 */  
  30% { CSS3 스타일 } /* 애니메이션 시간의 30% 진행 시점의 키 프레임 지정 */  
  60% { CSS3 스타일 } /* 애니메이션 시간의 60% 진행 시점의 키 프레임 지정 */  
  100% { CSS3 스타일 } /* 애니메이션 종료 시점의 키 프레임 지정 */  
}
```

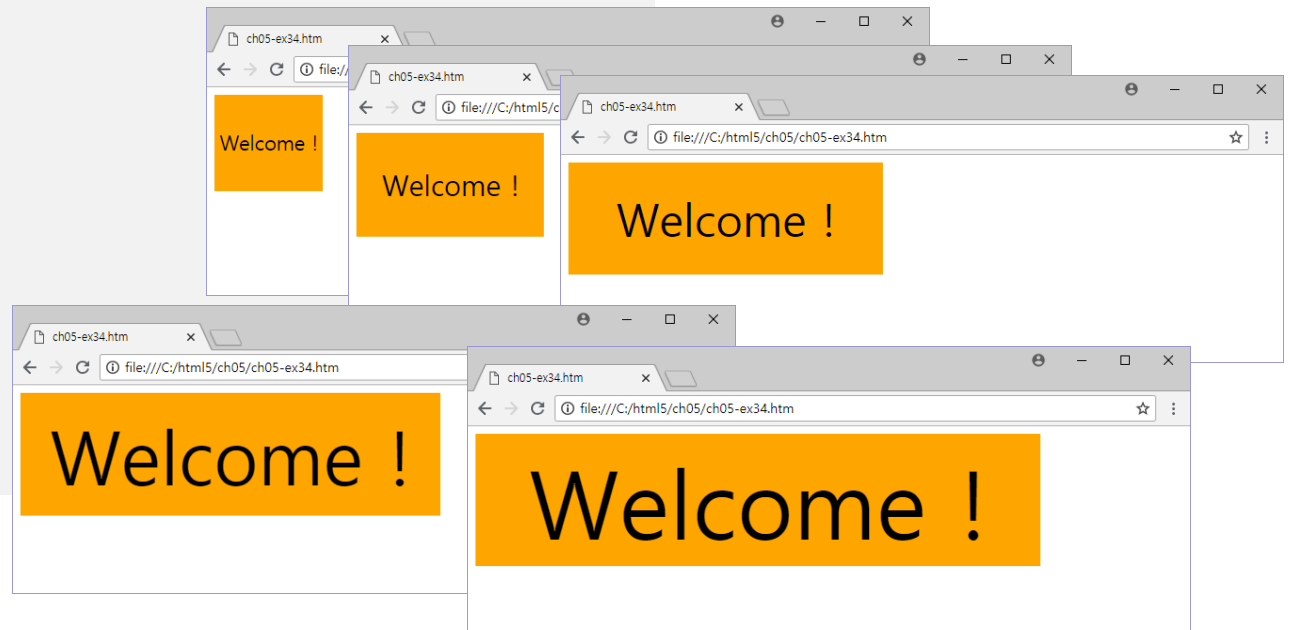
```
선택자 {  
  animation-name : 키 프레임명;  
  animation-duration : 애니메이션의 1회 지속시간;  
  animation-iteration-count : 애니메이션의 반복 횟수 또는 infinite;  
  animation-delay : 애니메이션 시작 지연시간;  
}
```

[예 5.34] 키 프레임 애니메이션 예

- "Welcome !"의 글자크기를 20px에서 100px로 단계적으로 변경함
- 20px → 30px → 50px → 80px → 100px의 크기순서로 계속 변경됨

```
<style>
  @keyframes bounding {
    0% { width: 100px; height: 100px; line-height: 100px; font-size: 20px; }
    25% { width: 200px; height: 110px; line-height: 110px; font-size: 30px; }
    50% { width: 350px; height: 120px; line-height: 120px; font-size: 50px; }
    75% { width: 450px; height: 130px; line-height: 130px; font-size: 80px; }
    100% { width: 600px; height: 140px; line-height: 140px; font-size: 100px; }
  }
  div {
    background-color: orange;
    text-align:center;
    animation-name: bounding;
    animation-duration: 10s;
    animation-iteration-count : infinite;
  }
</style>

<div> Welcome ! </div>
```



5.3.5 그레디언트 효과

- 그레디언트(gradient) : 두 가지 이상의 색상들 사이의 점진적인 변화 과정을 나타내는 것
- CSS3는 다양한 그레디언트 함수들을 제공함
- 색상값을 가지는 속성들의 속성값으로 사용가능함

[표 5.11] 그레디언트 함수 종류

함수 종류	함수 기능
linear-gradient();	아래/위/왼쪽/오른쪽/대각선 방향의 선형적 색상 변화
repeating-linear-gradient();	선형 그레디언트의 반복
radial-gradient();	박스 중심에서 바깥 방향으로 원형적 색상 변화
repeating-radial-gradient();	원형 그레디언트의 반복

[그림 5.14] 문서 배경색으로 그레디언트를 사용한 예



```
{ background: linear-gradient(orange,  
    blanchedalmond); }
```

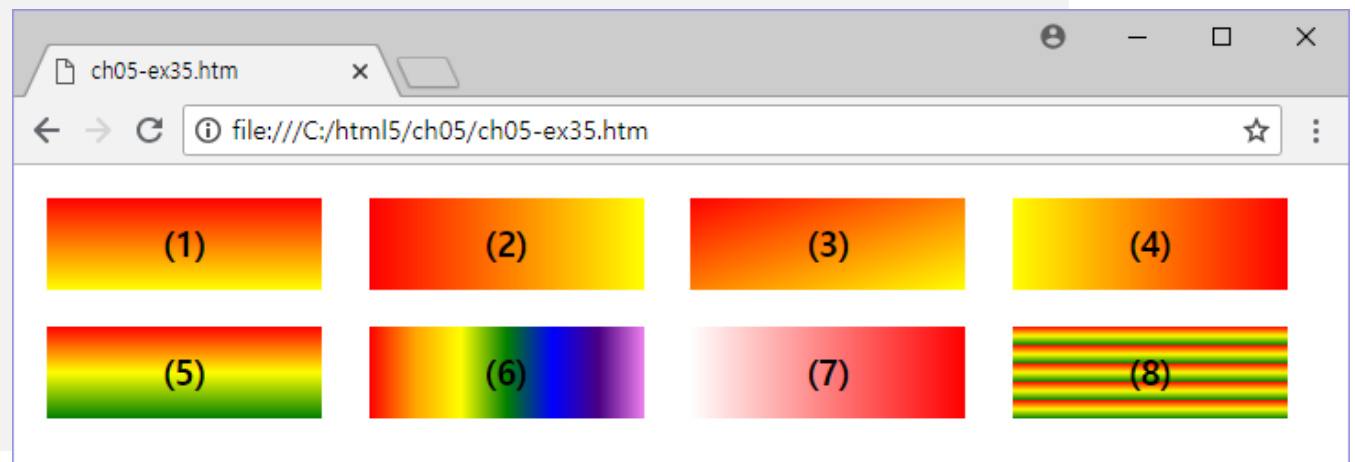


```
{ background: linear-gradient(to right,  
    blanchalmond, orange); }
```

[예 5.35] linear-gradient() 함수의 출력 결과

```
<style>
  h1 { width: 150px; height: 50px; margin:10px; line-height: 50px; text-align: center; display:
inline-block; }
  #b1 { background: linear-gradient(red, yellow); }
  #b2 { background: linear-gradient(to right, red, yellow); }
  #b3 { background: linear-gradient(to bottom right, red, yellow); }
  #b4 { background: linear-gradient(-90deg, red, yellow); }
  #b5 { background: linear-gradient(red, yellow, green); }
  #b6 { background: linear-gradient(to right, red,orange,yellow,green,blue,indigo,violet); }
  #b7 { background: linear-gradient(to right, rgba(255,0,0,0), rgba(255,0,0,1)); }
  #b8 { background: repeating-linear-gradient(red, yellow 10%, green 20%); }
</style>
```

```
<h3 id="b1"> (1) </h3>
<h3 id="b2"> (2) </h3>
<h3 id="b3"> (3) </h3>
<h3 id="b4"> (4) </h3>
<h3 id="b5"> (5) </h3>
<h3 id="b6"> (6) </h3>
<h3 id="b7"> (7) </h3>
<h3 id="b8"> (8) </h3>
```

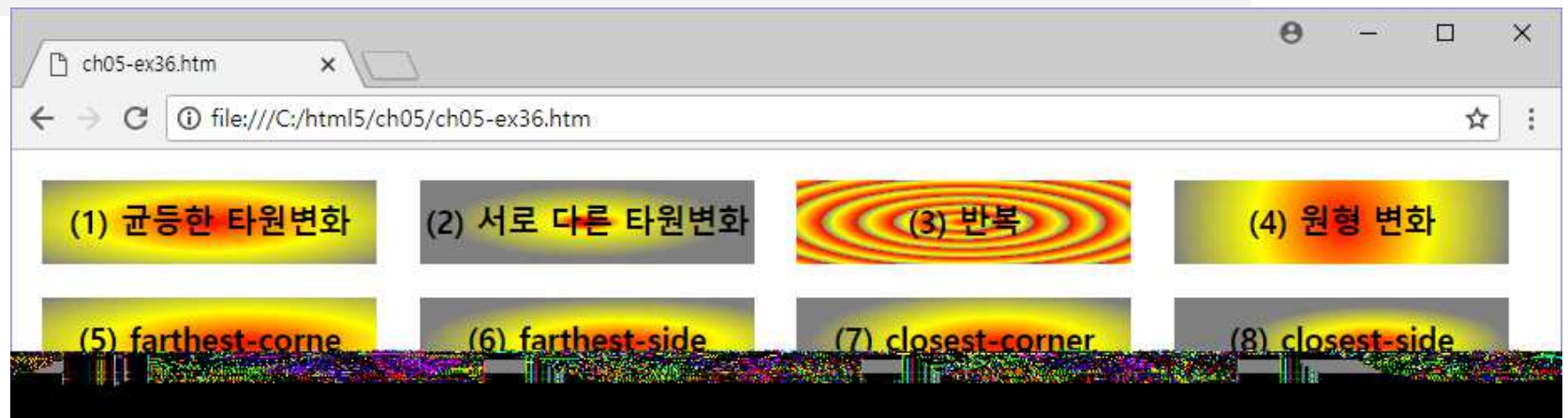



```

<style>
  h3 { width: 200px; height: 50px; margin:10px; line-height: 50px; text-align: center; display: inline-block; }
  #b1 { background: radial-gradient(red, yellow, gray); }
  #b2 { background: radial-gradient(red 5%, yellow 15%, gray 60%); }
  #b3 { background: repeating-radial-gradient(red, yellow 10%, gray 15%); }
  #b4 { background: radial-gradient(circle, red, yellow, gray); }
  #b5 { background: radial-gradient(farthest-corner at 60% 55%, red, yellow, gray); }
  #b6 { background: radial-gradient(farthest-side at 60% 55%, red, yellow, gray); }
  #b7 { background: radial-gradient(closest-corner at 60% 55%, red, yellow, gray); }
  #b8 { background: radial-gradient(closest-side at 60% 55%, red, yellow, gray); }
</style>

```

<h3 id="b1">(1) 균등한 타원변화</h3> <h3 id="b2">(2) 서로 다른 타원변화</h3>
 <h3 id="b3">(3) 반복</h3> <h3 id="b4">(4) 원형 변화</h3>
 <h3 id="b5">(5) farthest-corne</h3> <h3 id="b6">(6) farthest-side</h3>
 <h3 id="b7">(7) closest-corner</h3> <h3 id="b8">(8) closest-side</h3>

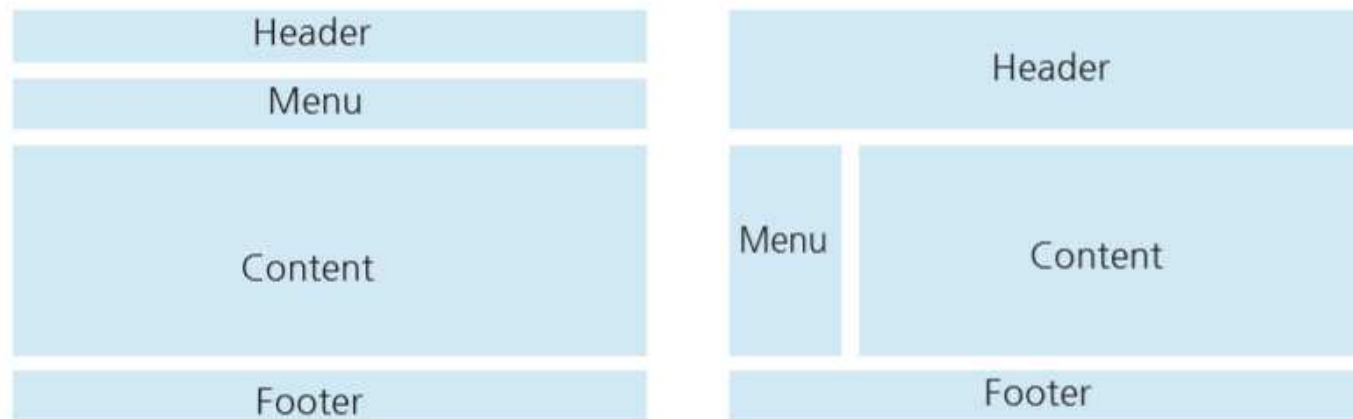


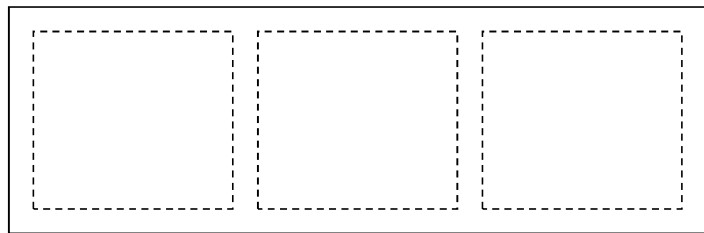
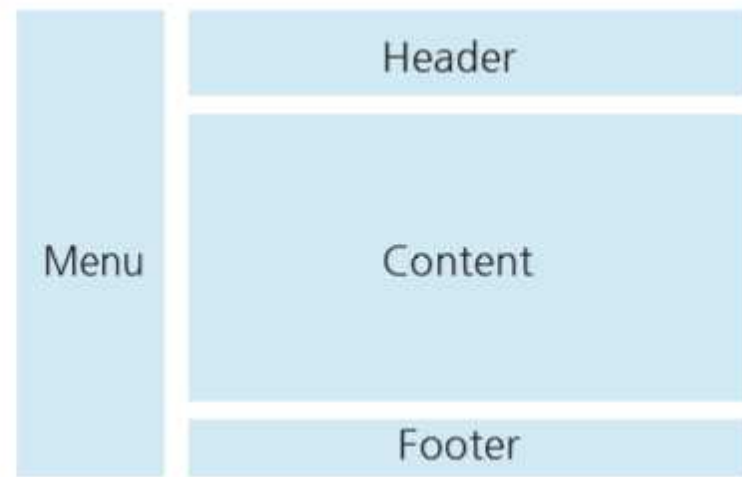
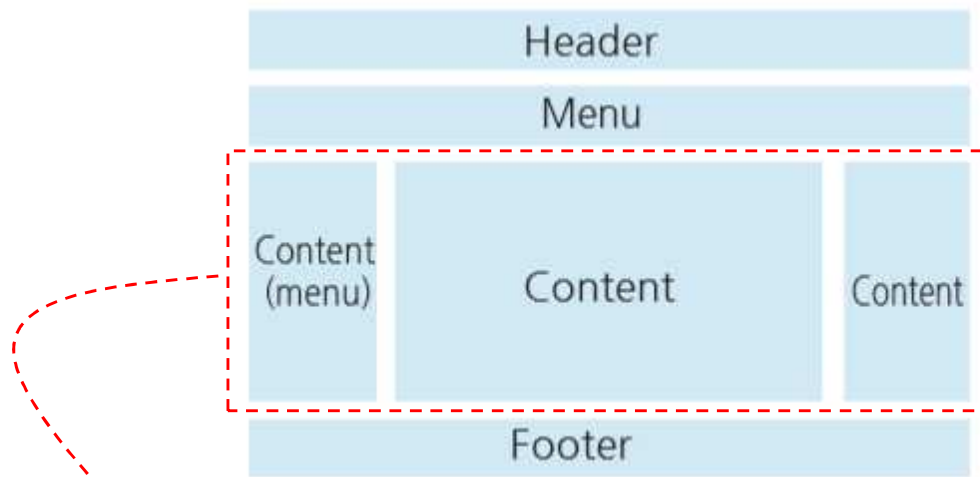
[예 5.36]
 radial-gradient()
 함수의 출력 결과

5.4 CSS3 응용 : 웹 페이지 레이아웃 구성

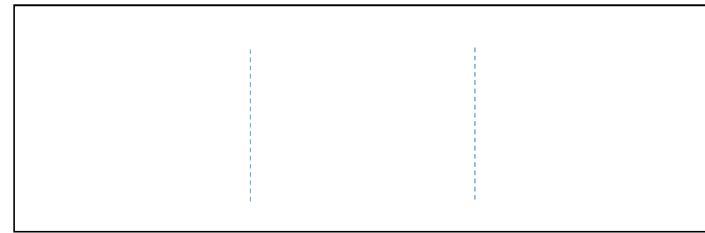
5.4.1 웹 페이지의 구성

- 사용자들이 방문하는 첫 페이지(홈페이지)의 구성의 중요성
 - 웹 사이트에 대한 첫 인상
 - 방문한 사용자들이 가능하면 오래 머무를 수 있도록 문서 구조를 구성해야 함
- 많은 그래픽과 이미지 사용 & 다양한 구조의 화면 구성 고려
- 웹 사이트의 일반적인 첫 페이지 구성
 - 헤더(Header), 메뉴(Menu), 그리고 내용(Content), 풋터 (Footer) 부분 구성





가) 부모박스과 자식박스들 이용

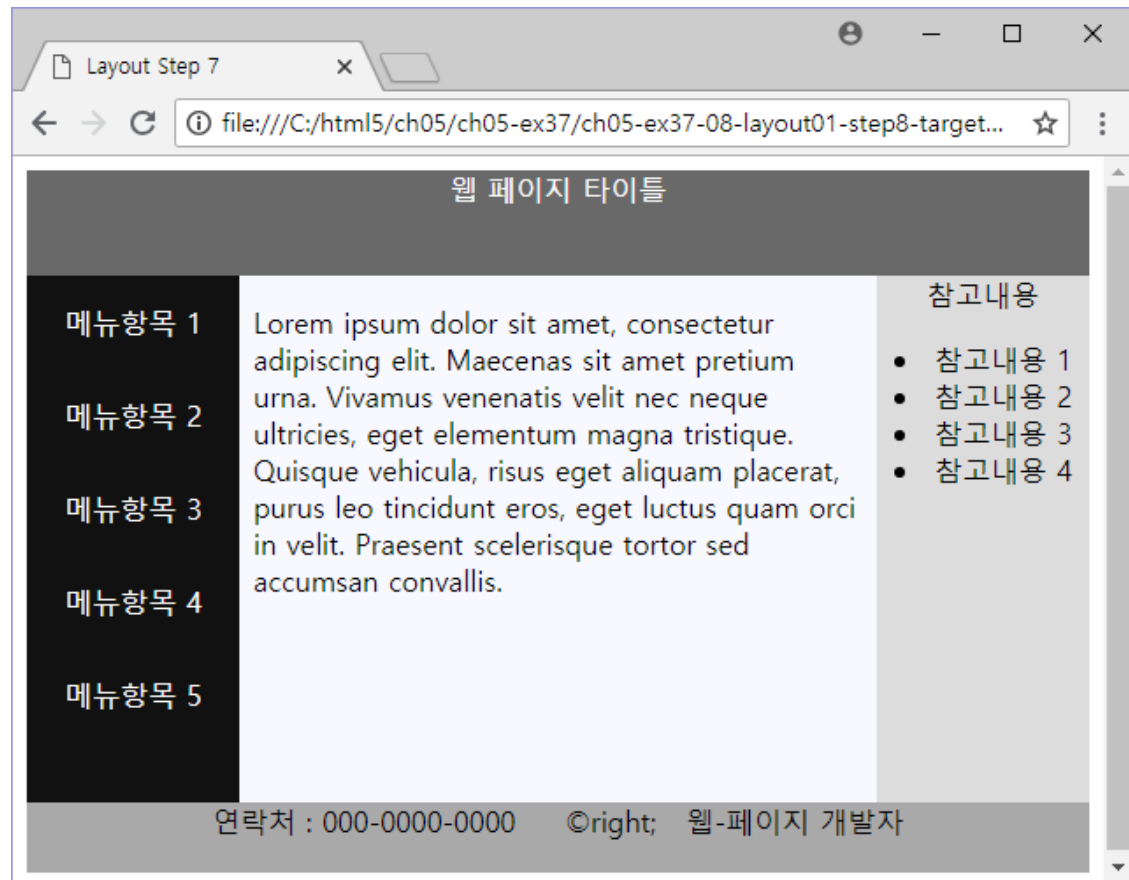


나) column-count 속성 이용

[그림 5.16] 다중 컬럼의 구성 방법

5.4.2 실습 : 웹 페이지 레이아웃의 단계적 구성

[그림 5.17] 수직 메뉴 페이지 구성



(1) 웹 페이지 레이아웃 구성 단계-1

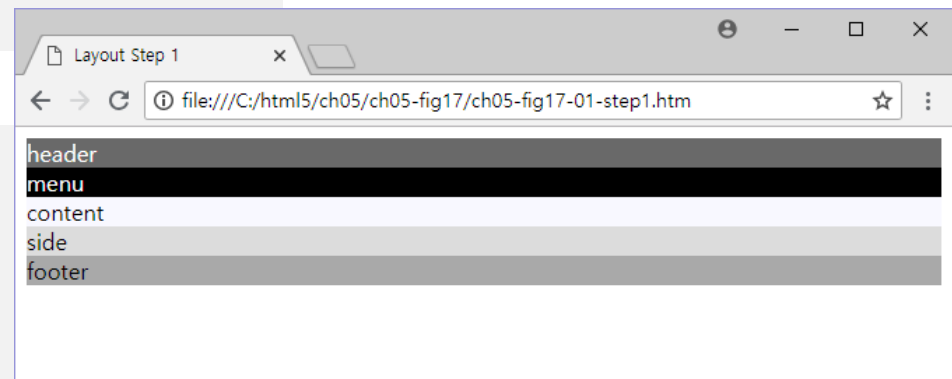
- 5개의 박스들을 <div> 태그로 나타내고, 배경색을 지정해서 구분함
- 각 박스의 id 이름을 각각 header, menu, content, side, footer로 지정함

박스 스타일 지정 (1)

```
<style>
* { box-sizing: border-box; }
#header { background: dimgray; color: white; }
#menu { background: #333; color: white; }
#content { background: ghostwhite; }
#side-right { background: Gainsboro; }
#footer { background: darkgray; }
</style>
```

HTML 문서내용 (1)

```
<body>
  <div id="header"> header </div>
  <div id="menu"> menu </div>
  <div id="content"> content </div>
  <div id="side"> side </div>
  <div id="footer"> footer </div>
</body>
```

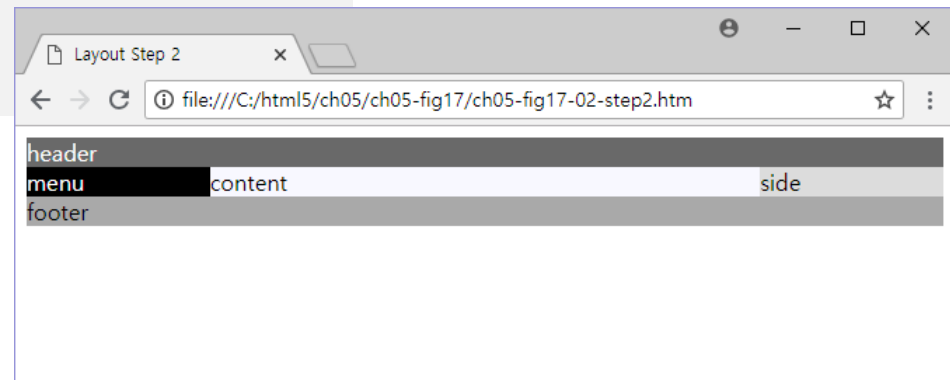


(2) 웹 페이지 레이아웃 구성 단계-2

- menu, content, side 박스는 플로팅박스로 지정해서 같은 행에 나타냄
- 세 박스의 width 속성은 부모박스(<body>)의 크기를 기준으로 상대적 크기(%)로 지정함
- 세 박스들의 width 속성의 합은 100%가 되도록 함(부모박스의 width 속성 크기와 같게 함)

스타일 추가 지정 (2)

```
<style>
  #menu { float: left; width: 20%; }
  #content { float: left; width: 60%; }
  #side { float: left; width: 20%; }
</style>
```



(3) 웹 페이지 레이아웃 구성 단계-3

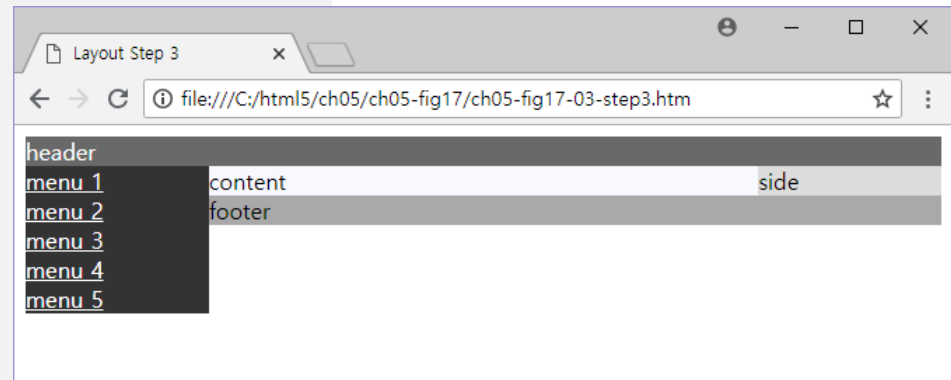
- menu 박스에 나타낼 메뉴 항목들을 작성함
- 메뉴항목들의 박스 유형은 블록박스로 지정함

스타일 추가 지정 (3)

```
<style>
  #menu a { display: block; color: white; }
</style>
```

HTML 문서내용 (3)

```
<body>
  <div id="header"> header </div>
  <div id="menu">
    <a href=""> menu item 1 </a>
    <a href=""> menu item 2 </a>
    <a href=""> menu item 3 </a>
    <a href=""> menu item 4 </a>
    <a href=""> menu item 5 </a>
  </div>
  <div id="content"> content </div>
  <div id="side"> side </div>
  <div id="footer"> footer </div>
</body>
```

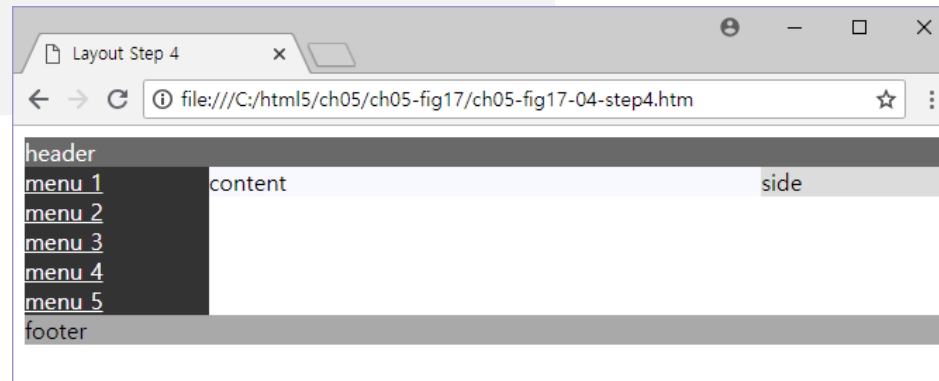


(4) 웹 페이지 레이아웃 구성 단계-4

- footer 박스의 clear 속성을 both(또는 left)로 지정해서, 앞의 모든 플로팅박스들과 중복되지 않게 배치함

스타일 추가 지정 (4)

```
<style>  
  #footer { clear : both; }  
</style>
```



(5) 웹 페이지 레이아웃 구성 단계-5

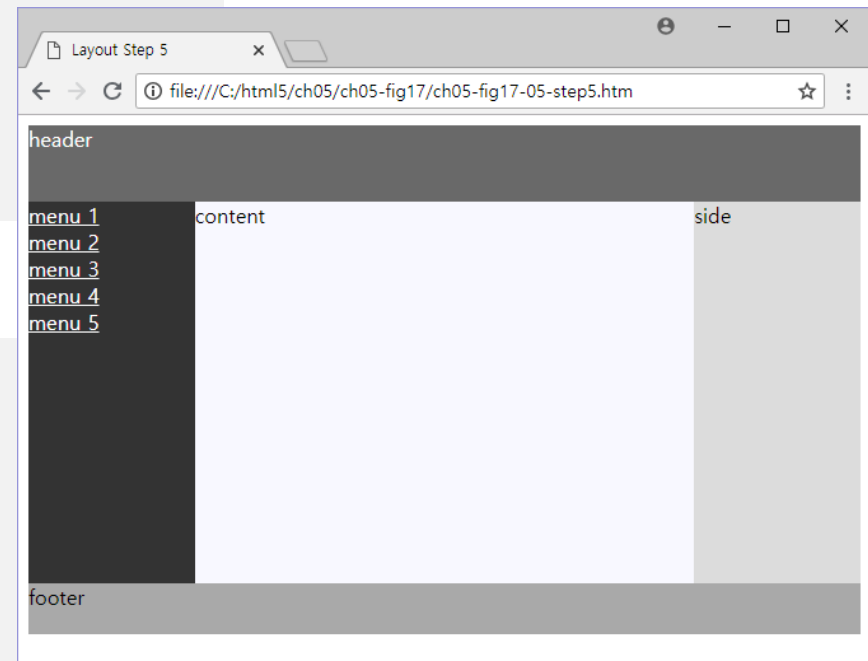
- menu와 content, side 박스의 높이 (height)를 동일하게 지정함

스타일 추가 지정 (5)

```
<style>
  #container { height: 400px; }
  #header { height: 15%; }
  #menu, #content, #side-right { height: 75%; }
  #footer { height: 10%; }
</style>
```

HTML 문서내용 (5)

```
<body>
  <div id="container" >
    <div id="header"> header </div>
    ... 생략 ..
    <div id="footer"> footer </div>
  </div>
</body>
```



(6) 웹 페이지 레이아웃 구성 단계-6

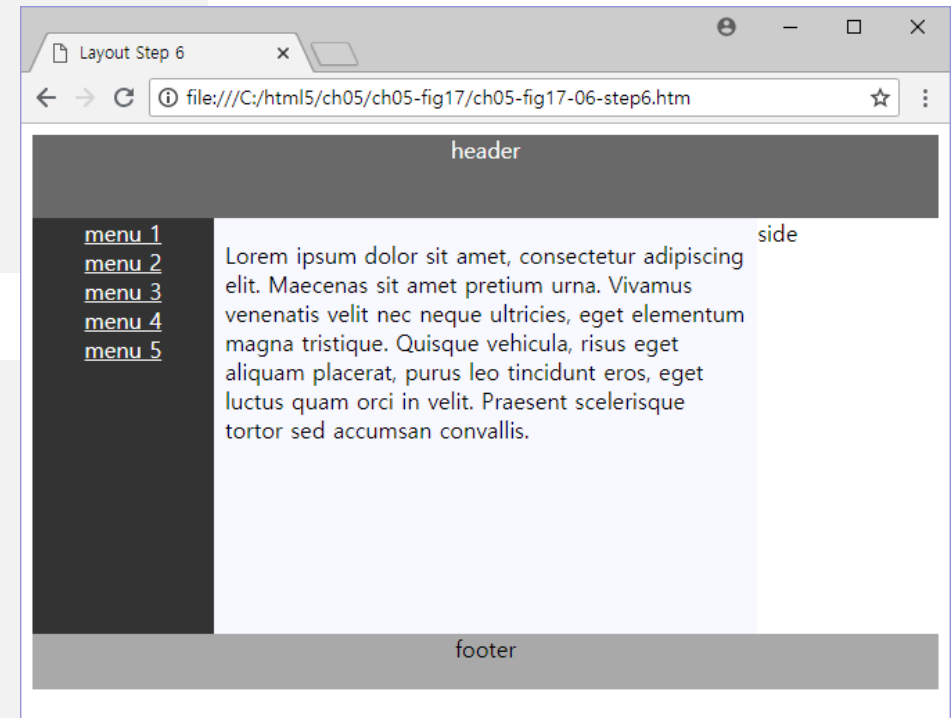
- menu 박스의 메뉴항목들을 선택하면 연결문서를 content 박스 영역에 나타나게 해야 함
 - content 박스의 자식박스로 <iframe> 박스를 포함시킴
 - 또는 content 박스를 <iframe> 박스로 대체함

스타일 추가 지정 (6)

```
<style>
  #f1 { margin: 0px 0px; padding: 0px 0px;
        border: 0px;
        width: 100%; height: 100%;
      }
  #header, #menu, #footer { text-align : center; }
```

HTML 문서내용 (6)

```
<body>
....
<div id="content">
  <iframe src="sample.htm"
    name="f1" seamless/>
</div>
....
</body>
```

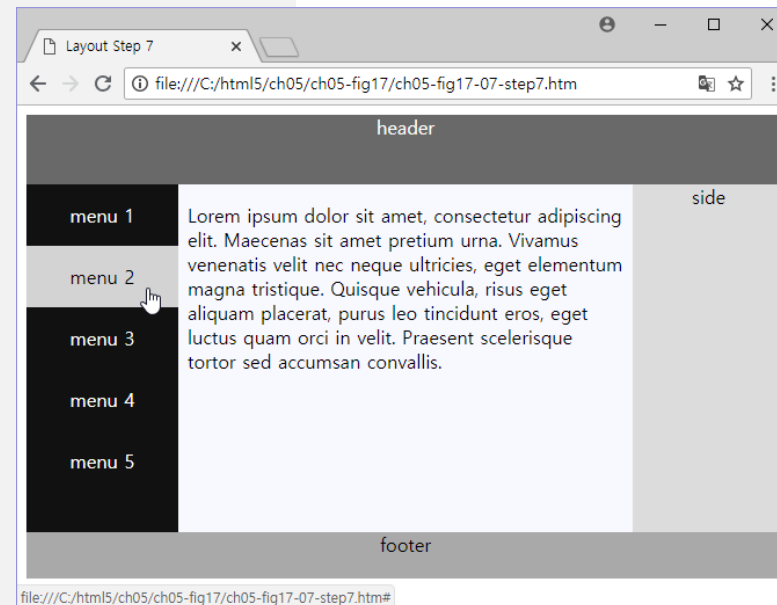


(7) 웹 페이지 레이아웃 구성 단계-7

- menu 박스에 나타나는 메뉴 항목들(<a> 태그들)의 박스스타일을 지정함
- 가상 클래스 :hover를 이용해서 마우스가 박스 위로 있을 때의 배경색을 다르게(예: lightgray 색) 지정함

스타일 추가 지정 (7)

```
<style>
  #menu {
    z-index: 1;
    background: black;
    overflow-x: hidden;
  }
  #menu a {
    color: white;
    padding: 16px;
    text-decoration: none;
  }
  #menu a:hover {
    background-color: lightgray;
    color: black;
  }
</style>
```

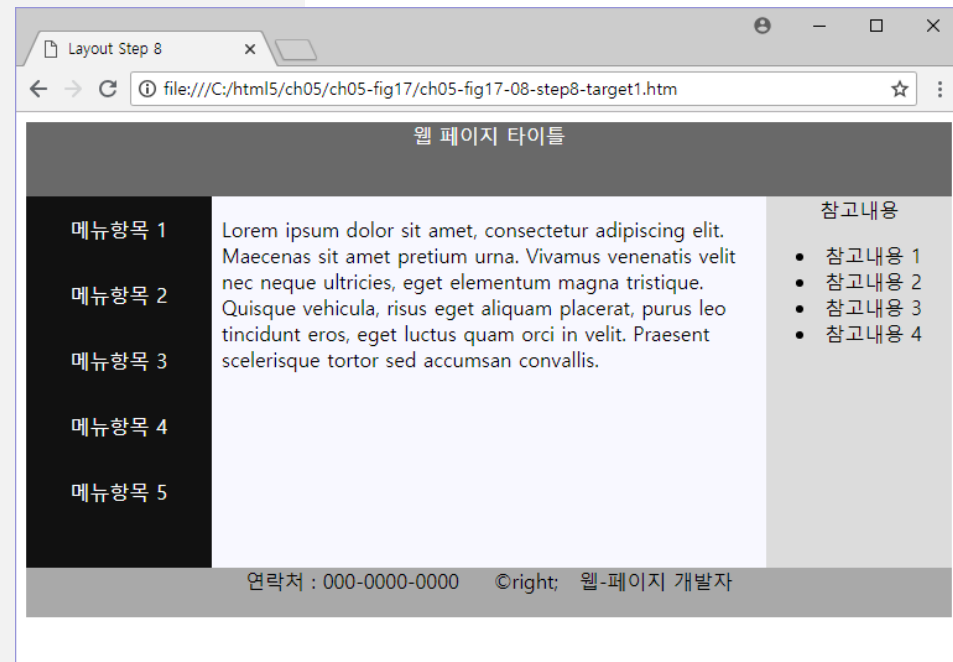


(8) 웹 페이지 레이아웃 구성 단계-8

- header, menu, side, footer 박스의 내용을 필요한 내용으로 변경함
- [그림 5.17-8]의 목표 웹 페이지 완성

HTML 문서내용 (8)

```
<body>
  <div id="header"> Header </div>
  ...
  <div id="side"> 참고내용
    <ul>
      <li> 참고내용 1 </li>
      <li> 참고내용 2 </li>
      <li> 참고내용 3 </li>
      <li> 참고내용 4 </li>
    </ul>
  </div>
  <div id="footer">
    연락처 : 000-0000-0000 ...
  </div>
</body>
```

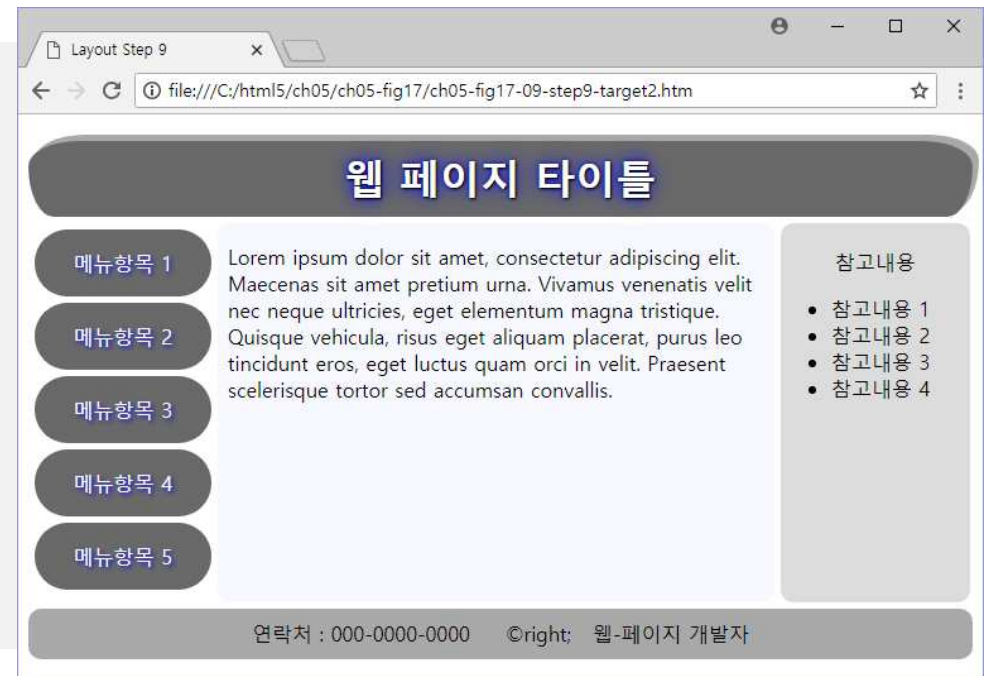


[그림 5.17-9] 목표 웹 페이지 완성(2)

- 모서리 라운딩, 그림자 효과 등의 출력스타일 지정 가능

```
/* step-3 */
#menu a { display: block;
           background: dimgray;
           margin : 5px;
           border-radius : 40px;
           color: white; padding: 16px; text-decoration: none;
           text-shadow : 2px 2px 5px blue;
        }

/* step 8 */
h1 { text-shadow : 1px 1px 2px black, 0 0 25px blue,
              0 0 5px darkblue;
    }
```



[그림 5.18] 수평 메뉴 페이지 구성

```
<style>
/* step 1 */
* { box-sizing: border-box; }
#header { background: dimgray; color: white; }
#menu { background: #333; color: white; }
#content { background: ghostwhite; }
#side { background: Gainsboro; }
#footer { background: darkgray; }
```

```
/* step 2 */
#content { float: left; width: 80%; }
```

```
/* step 3 */
#menu a { color: white; }
```

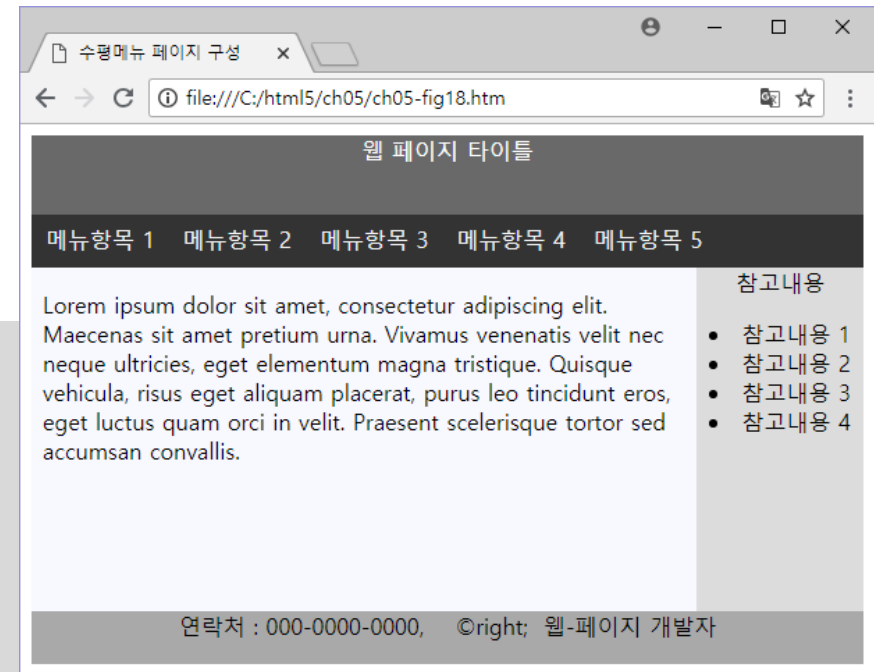
```
/* step 4 */
#footer { clear : both; }
```

```
/* step 5 */
#container { height: 380px; }
#header { height: 15% }
#menu { height: 10% }
#content, #side { height: 65% }
#footer { height: 10% }
```

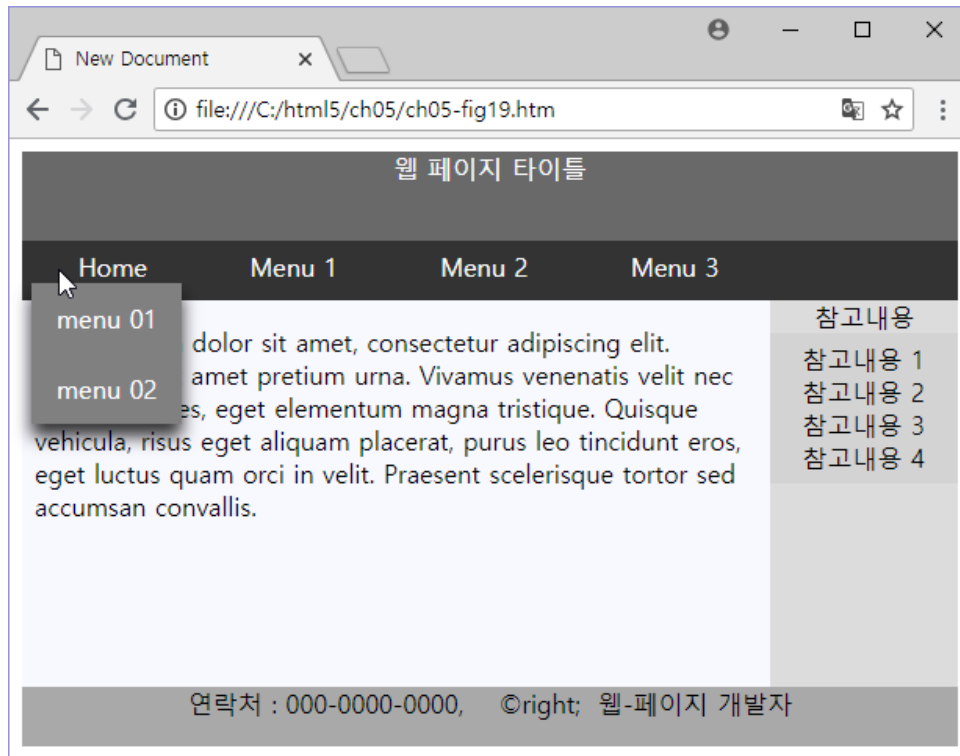
```
/* step 6 */
iframe { margin: 0px 0px; padding: 0px 0px;
border: 0px; float: right;
width: 100%; height: 100%; }
#header, #side, #footer { text-align: center; }
```

```
/* step 7 */
#menu { overflow: hidden; }
#menu a { float:left; display: block;
color: #f2f2f2; text-align: center;
padding: 7px 10px; text-decoration: none; }
#menu a:hover { background-color: #bbb;
color: black; }
```

```
/* step 8 */
li { list-style-position : inside; }
</style>
```



[그림 5.19] 드롭다운 메뉴 제공 웹 페이지



```
/* 드롭다운 서브메뉴 */
#menu { padding: 6px; }
ul { background: lightgray; padding: 6px; margin: 0; }
li { display: inline; padding: 10px 20px; }
li a { color: black; text-decoration: none; padding: 5px 10px; }
li a:hover { background: darkgray; color: black; }
li.mainmenu { display: inline-block; }
.submenu { display: none; position: absolute;
           background-color: gray;
           box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.8); }

.submenu a { display: block; color: black; padding: 12px 16px;
            text-decoration: none; text-align: left; }

.submenu a:hover { background: lavender; color: black; }
.dropdown:hover .submenu { display: block; color: black; }
```

Next 6장



자바스크립트 기본