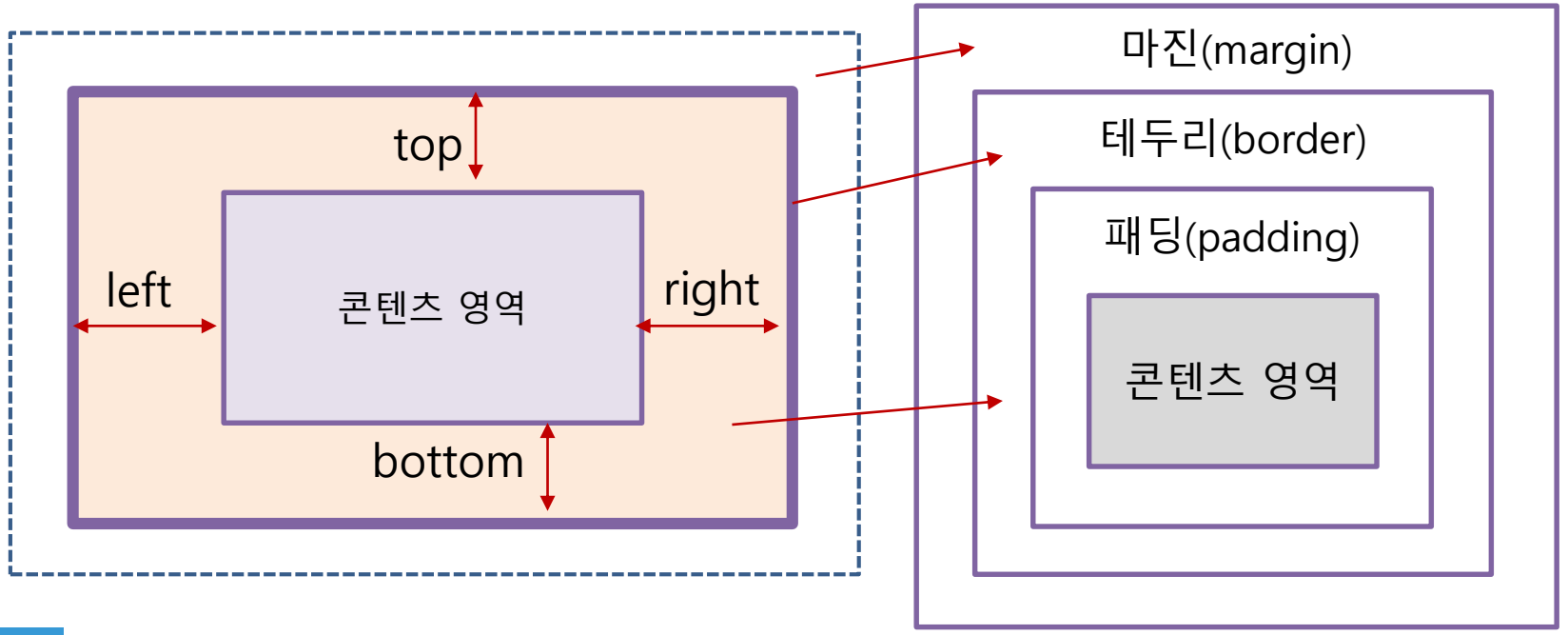
3강. 박스 모델 & 배치

CSS 박스 모델

CSS 박스모델

웹 문서에서 내용을 배치할 때는 각 요소를 박스 형태로 구성한다.

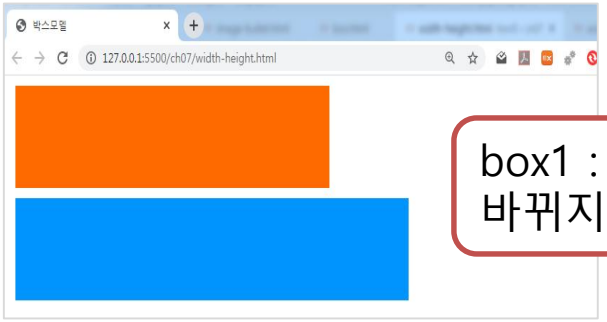
박스 모델을 실제 내용이 들어가는 콘텐츠 영역과 테두리(border), 여백(margin, padding)들로 구성된다.



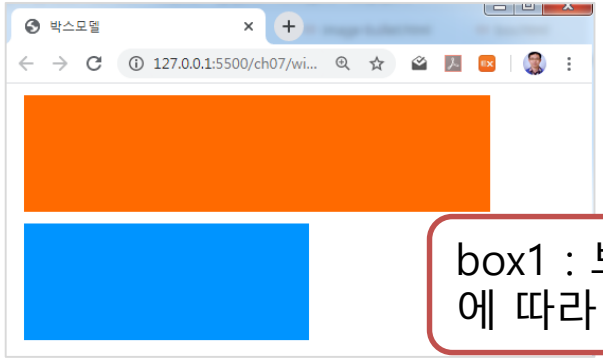
박스 모델

width, height 속성 – 콘텐츠 영역의 크기

width-height.html



box1 : 너비와 높이가 바뀌지 않음



box1 : 브라우저 창의 비율에 따라 크기가 달라짐

```css
.box1{
    width: 400px;
    height: 100px;
    background: orange;
    margin: 20px 20px 20px 20px;
}
.box2{
    width: 50%;
    height: 100px;
    background: #0594ff;
    margin: 20px;
}
```

바깥 여백 – margin 속성

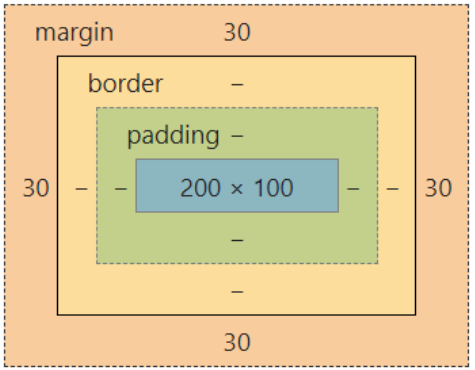
margin 속성

- 마진(margin)은 현재 요소 주변의 여백이다.
- 한 요소와 다른 요소 사이의 간격을 조절할 수 있다.

margin: <크기> | <두께> | <색상>
margin-top: <크기> | <두께> | <색상>
margin-right: <크기> | <두께> | <색상>
margin-bottom: <크기> | <두께> | <색상>
margin-left: <크기> | <두께> | <색상>

바깥 여백 – margin 속성

margin 속성 – 요소 주변 여백 설정하기

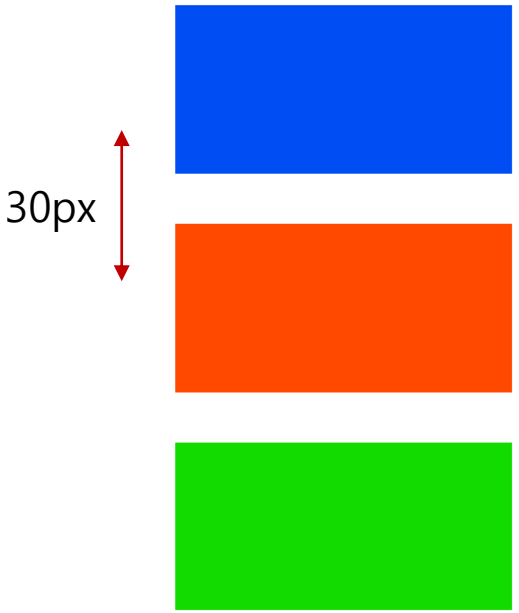



```
<style>
  div {
    width:200px;  /* 너비 */
    height:100px;  /* 높이 */
    margin:30px;  /* 마진 */
    display:inline-block;  /* 가로로 배치 */
  }
  #box1 {
    background:rgb(0, 77, 243);
  }
  #box2 {
    background:rgb(255, 72, 0);
  }
  #box3 {
    background:rgb(18, 219, 0);
  }
</style>
```

바깥 여백 – margin 속성

margin 속성 – 요소 주변 여백 설정하기

마진 중첩 – 위 마진과 아래 마진이 서로 만날때 큰 마진값으로 합쳐지는 것을 말하는데, 가로 마진은 중첩 안됨




```
<style>
  div {
    width:200px;  /* 너비 */
    height:100px;  /* 높이 */
    margin:30px;  /* 마진 */
  }
  #box1 {
    background: rgb(0, 77, 243);
  }
  #box2 {
    background: rgb(255, 72, 0);
  }
  #box3 {
    background: rgb(18, 219, 0);
  }
</style>
```

바깥 여백 – margin 속성

margin 속성 – 가운데 정렬하기

margin-center.html

```
.box {
    width:200px;  /* 너비 */
    height:300px;  /* 높이 */
    background:#ff6a00;  /* 배경색 */
    margin:0 auto; /* 마진 - 0 auto 0 auto */
}
```

margin:0 auto

요소의 너비 값을 뺀 나머지 공간의 좌우 마진을 똑같이 맞춘다.

안쪽 여백 – padding 속성

padding 속성

- 패딩(padding)은 콘텐츠 영역과 테두리 사이의 여백을 말한다.

padding: <크기> | <두께> | <색상>

padding-top: <크기> | <두께> | <색상>

padding-right: <크기> | <두께> | <색상>

padding-bottom: <크기> | <두께> | <색상>

padding-left: <크기> | <두께> | <색상>

안쪽 여백 – padding 속성

padding 속성 – 콘텐츠 영역과 테두리 사이 여백 설정하기

패딩(padding)이란 콘텐츠 영역과 테두리 사이의 여백을 말합니다.

패딩(padding)이란 콘텐츠 영역과 테두리 사이의 여백을 말합니다.

패딩(padding)이란 콘텐츠 영역과 테두리 사이의 여백을 말합니다.

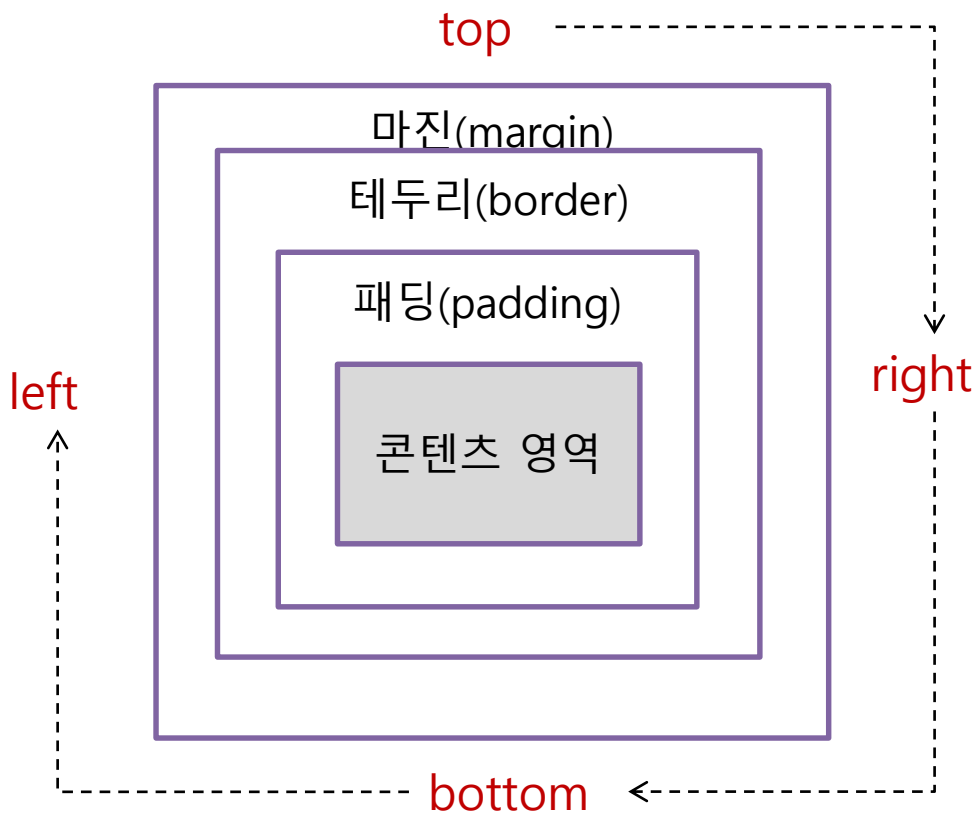
```
<body>
    <div class="box1">패딩(padding)이란 콘텐츠영역과 테두리사이의 여백을 말합니다.</div>
    <div class="box2">패딩(padding)이란 콘텐츠영역과 테두리사이의 여백을 말합니다.</div>
    <div class="box3">패딩(padding)이란 콘텐츠영역과 테두리사이의 여백을 말합니다.</div>
</body>
```

```
<style>
    div{
        width:200px;
        background-color: #0094ff;
        display: inline-block;
        margin: 15px;
    }
    .box1{padding:10px;}
    .box2{padding:10px 30px 10px 30px}
    .box3{padding:10px 30px}
</style>
```

테두리 관련 속성들

테두리(Border)

- 4개 방향의 값을 한꺼번에 지정할 때는 방향 순서를 지켜야 함
- border-width: top -> right -> bottom -> left



박스 모델(Box Model)

border 색상, 두께, 색상 등을 한꺼번에 묶어 표기

border: <크기> | <두께> | <색상>

border-top: <크기> | <두께> | <색상>

border-right: <크기> | <두께> | <색상>

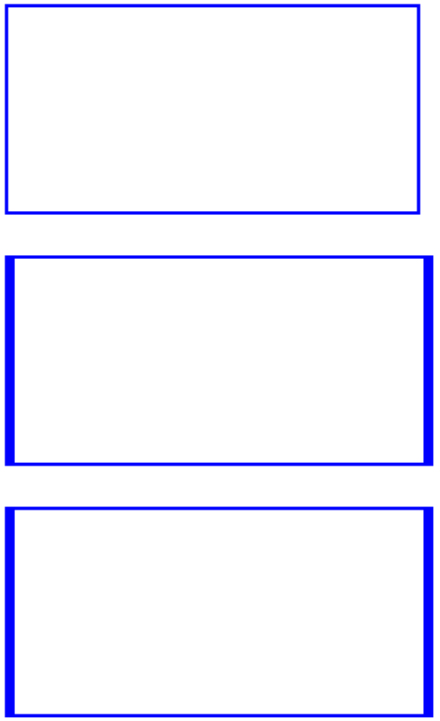
border-bottom: <크기> | <두께> | <색상>

border-left: <크기> | <두께> | <색상>

테두리 관련 속성들

border-style/ border-width / border-color

border.html

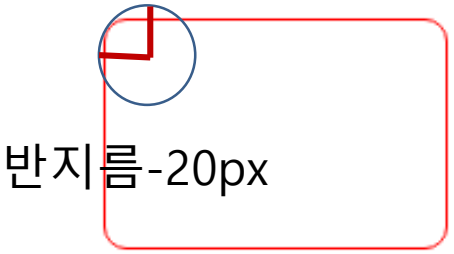


```
div{
    width: 200px;
    height: 100px;
    border-style: solid;
    border-color: blue;
    margin-bottom: 20px;
}
#box1{
    border-width: 2px;
}
#box2{
    border-width: 2px 5px;
}
#box3{
    border-width: 2px 5px 2px 5px;
}
```

테두리 관련 속성들

border-radius 속성 – 박스 모서리 둥글게 만들기

- 박스 모서리 부분을 다양한 형태로 처리할 수 있다.



```
    <style>
        img{
            border-radius: 150px;
            /*border-radius: 50%;*/
            /* px - 반지름의 크기, %는 현재 요소의 크기 기준*/
        }
    </style>
</head>
<body>
    <img src="images/photo.jpg" alt="손">
</body>
```

박스 모델(Box Model)

박스 모델

CSS 박스모델이란 웹 문서의 내용을 박스 형태로 정의하는 방법이다. 실제 컨텐츠 영역, 박스와 컨텐츠 영역사이의 여백인 패딩(padding), 테두리 영역과 박스 모델간의 여백인 마진(margin)으로 구성되어 있다.

```css
h1{
    width: 400px;
    margin-left: 20px;
    padding: 4px 16px;
    border-bottom: 5px solid #ccc;
}
p{
    width: 400px;
    border: 3px dotted blue;
    padding: 8px 16px;
    margin-left: 20px;
}
```

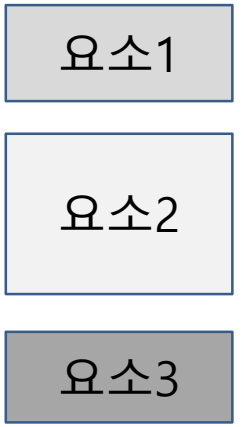
box_model.html

```html
<h1>박스 모델</h1>

<p>CSS 박스모델이란 웹 문서의 내용을 박스 형태로 정의하는 방법이다.
   실제 콘텐츠 영역, 박스와 콘텐츠 영역사이의 여백인 패딩(padding),
   테두리 영역과 박스 모델간의 여백인 마진(margin)으로 구성되어 있다.
</p>
```

블록 레벨 요소와 인라인 레벨 요소

블록 레벨 요소

- 요소를 삽입했을 때 **혼자 한 줄**을 차지하는 요소이다.
- 요소의 너비가 100%라는 의미
- ** , <p>** 태그 등

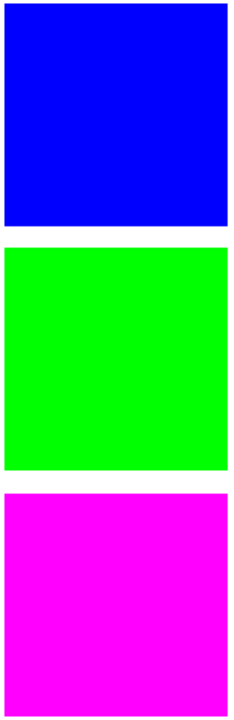


인라인 레벨 요소

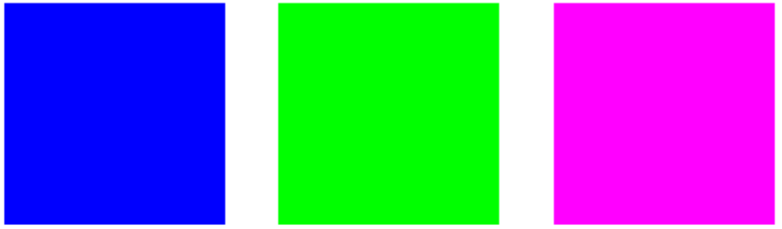
- 줄을 차지하지 않는 요소
- 화면에 표시되는 콘텐츠 만큼만 영역을 차지하고 나머지 공간에는 다른 요소가 들어올 수 있다.
- **, <img>** 태그 등

display 속성

display 속성



display: block



display: inline-block

display 속성

display 속성

```
div{
    width: 100px;
    height: 100px;
    margin: 10px;
    /* display: block; */
    display: inline-block;
}
.box1{background: #00f;}
.box2{background: #0f0;}
.box3{background: #f0f;}
```

display 속성

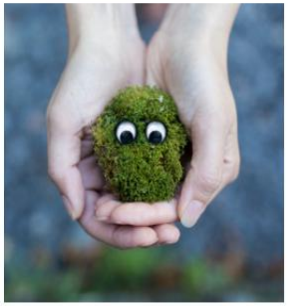
display 속성 – 이미지



이미지는 기본이 inline(수평정렬)


display: none





display: block

display 속성

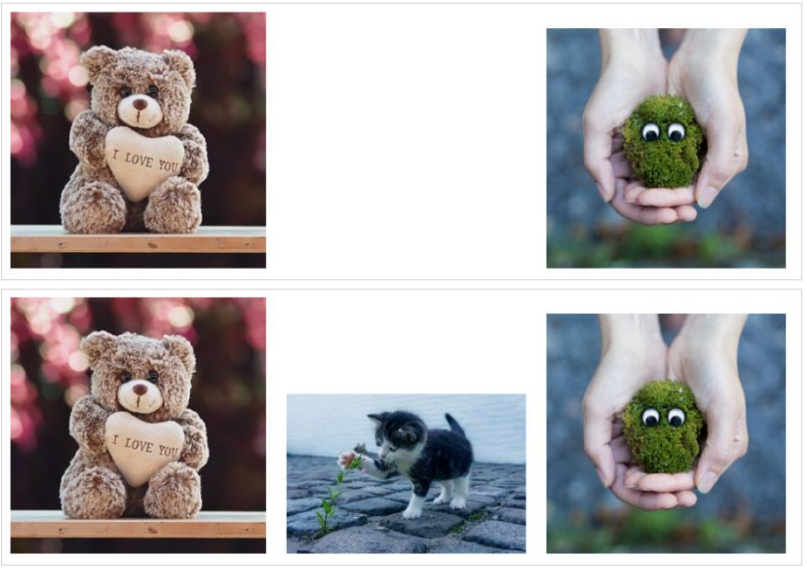
```css
.pic{
    /* img 태그는 기본 수평 배치가 됨 */
    /* 표시하지 않고 숨기기 할때 - none 사용 */
    /* 이미지의 공간도 사라짐 */
    /*display: none;*/

    /* 수직 배치 */
    display: block;
}
```

display1.html

```html
<img src="images/bear.jpg" alt="곰인형">
<img src="images/cat.jpg" alt="고양이" class="pic">
<img src="images/photo.jpg" alt="손">
```

visibility 속성

visibility 속성 – 이미지 숨기기/보이기(공간 유지)



display2.html

```
.pic{
    /* 표시하지 않고 숨기기 할때 - hidden 사용 */
    /* 이미지의 공간은 유지됨 */
    visibility: hidden;

    /* 보이기 할때 - visible 사용 */
    /* visibility: visible; */
}
```

블록 레벨 요소

- 좌측 메뉴 만들기

ABC 주식회사

회사 소개
도서
자료실
동영상 강의

블록 레벨 요소

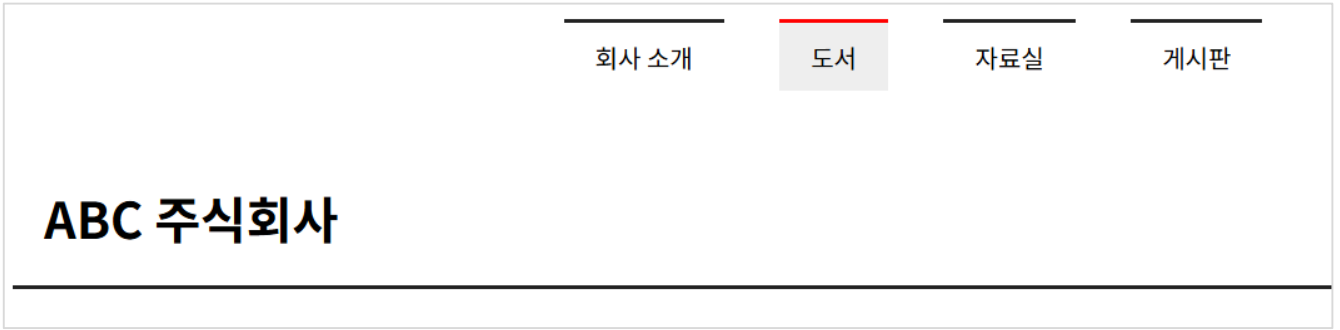
navbar.html

```html
<h1>ABC 주식회사</h1>
<nav>
  <ul>
    <li><a href="#">회사 소개</li>
    <li><a href="#">도서</li>
    <li><a href="#">자료실</li>
    <li><a href="#">동영상 강의</li>
  </ul>
</nav>
```

```css
<style>
    nav{width: 300px;}
    ul li{
        list-style: none;
        border: 1px solid #222;
        padding: 20px 40px;
        cursor: pointer;  /* 마우스 커서 포인터 모양 */
    }
    ul li a{
        text-decoration: none;
    }
    ul li:hover{background-color: #eee;}
</style>
```

display:inline-block

- 상단 메뉴 만들기



navbar2.html

```
<header>
    <nav>
        <ul>
            <li><a href="#">회사 소개</a></li>
            <li><a href="#">도서</a></li>
            <li><a href="#">자료실</a></li>
            <li><a href="#">게시판</a></li>
        </ul>
    </nav>
</header>

<section>
   <h1>ABC 주식회사</h1>
</section>
```

display:inline-block

style.css

```css
body{margin: 0;}/* body의 기본 margin 제거 */
nav ul{
    list-style: none;
    margin: 0; /* ul의 기본 margin 제거 */
    padding: 0; /* ul의 기본 padding 제거 */
    text-align: center;
}
nav ul li{
    display: inline-block; /* li를 가로로 나열 */
    margin: 16px; /* li 사이의 간격 */
    border-top: 3px solid #222;
    padding: 10px;
}
nav ul li a{
   text-decoration: none;
   color: #000;
   padding: 10px;
}
```

display:inline-block

style.css

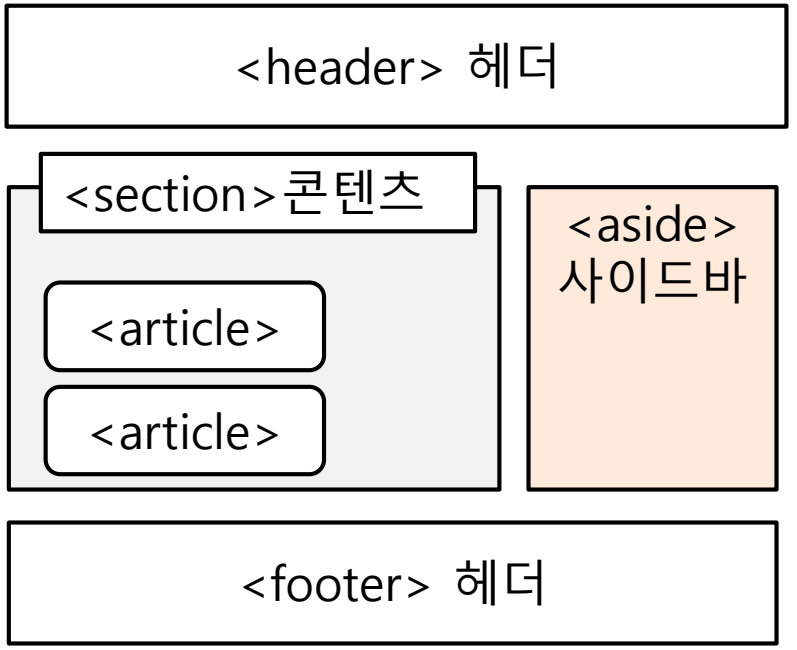
```css
nav ul li:hover{
    /* 마우스 오버 시 변경 */
    background-color: #eee;
    border-top: 3px solid #f00;
}

/* section 스타일 */
section{margin: 20px;}
section h1{
    padding: 20px;
    border-bottom: 3px solid #222;
}
```

HTML5 문서

HTML5 문서

웹 문서의 표준화를 위해 레이아웃을 만들 때 사용하는 주요 태그를 미리 약속.

태그만 봐도 직관적으로 어느 부분이 헤더이고 실제 내용인지 알 수 있음.
즉, 문서구조를 정확히 이해할 수 있음.



- 2008년 1월 HTML5 초안 공개
- 2014년 11월 공식 표준 발표
- 2022년 현재 W3C에서 명세를 관리함

HTML5 문서

<header> 태그 – 머리말 지정하기

- 사이트 전체의 제목 부분이 될 수도 있고, 본문의 제목 부분이 될 수도 있다.
- <nav> 태그를 사용해 사이트 메뉴를 넣을 수 있다.

<section>, <article>태그 – 주제별 컨텐츠 영역

- 문서에서 주제별로 컨텐츠를 묶을 때 사용
- 섹션 제목을 나타내는 <hn> 태그가 함께 사용됨

<aside> 태그 – 본문 이외의 내용

- 본문 내용 외에 주변에 표시되는 기타 내용들
- 광고나 링크 모음 등 문서의 메인 내용에 영향을 미치지 않는 내용

<footer> 태그 – 연락처정보, 주소

- 사이트 제작자의 연락처 정보와 저작권 정보
- 주소는 **<address>** 태그가 사용됨

HTML5 문서


```
<header>
  <div class="nav_head">…</div>
  <nav>
    <div class="wrap" id="gnb">…</div> scroll
  </nav>
</header>
<main>…</main>
<!-- 푸터 영역 -->
<footer>
  <div class="wrap">…</div>
  <div class="graybox">…</div>
```

HTML5 문서

```
▼<main>
  <!-- 사이드바 -->
  <div class="sidebar_background" id="sidebar_wrap" onclick="w3_clo
  ▶<div class="sidebar" id="sidebar">…</div>
  <!-- //사이드바 -->
  <!-- 9등급 -->
  ▼<section class="lank9">
    ▶<div class="wrap">…</div> grid
  </section>
  <!-- //9등급 -->
  <!-- 특집 -->
```

CSS 포지셔닝

CSS 포지셔닝이란?

- 브라우저 화면 안에 각 콘텐츠 영역을 어떻게 배치할지 결정하는 것
- float 속성과 position 속성이 있다.
- 박스 모델의 패딩이나 마진, 테두리 속성까지 포함해 전체적인 레이아웃이 완성 된다.

float 속성

- 요소를 왼쪽이나 오른쪽에 떠 있게 만듬

- float 속성을 사용하면 그 다음에 넣는 다른 요소들에도 똑같은 속성이 적용

float: left | right

속성 값	설명
left	해당 요소를 문서의 왼쪽으로 배치
right	해당 요소를 문서의 오른쪽으로 배치

float 속성

float 속성 예제.

float1.html

```
<div class="box1">박스1</div>
<div class="box2">박스2</div>
<div class="box3">박스3</div>
<div class="box4">박스4</div>
```

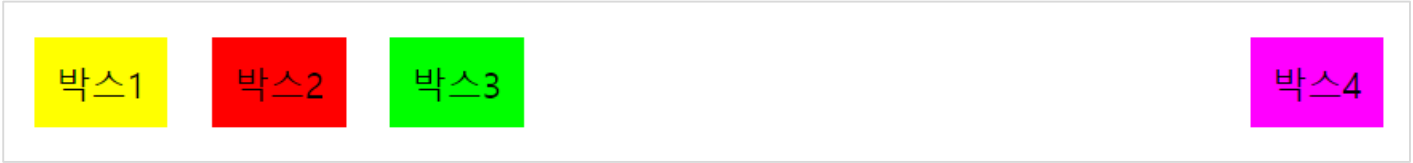
float 속성

float 속성 예제.

```
div{
    padding: 10px;
    margin: 10px;
    float: left; /* 모든 박스에 float 적용 */
    /*width: 100px; /* 박스 크기 지정 */
    /*display: inline-block; /* float 대신 inline-block 사용 */
}
.box1{
    background: #f0f;
    /* box1의 위치 left가 되고 다음 개체는 그 옆에 위치함 */
    /* float: left;  */
}
.box2{background: #0f0;}
.box3{background: #ff0;}
.box4{background: #0ff;}
```

float 속성

float 속성 예제.




```
div{
    margin: 10px;
    padding: 10px;
}
.box1{background: #ffff00; float:left;}
.box2{background: #ff0000; float:left;}
.box3{background: #00ff00; float:left;}
.box4{background: #ff00ff; float:right;}
```

float 속성

clear 속성 - 해제하기

float 속성을 사용하면 그 다음에 넣는 다른 요소들에도 똑같은 속성이 적용되므로 해제하고 싶을 때 clear 속성을 사용한다.

clear : left | right | both


```
div{
    margin: 10px;
    padding: 10px;
}
.box1{background: #ffff00; float:left;}
.box2{background: #ff0000; float:left;}
.box3{background: #00ff00; float:right;}
.box4{background: #ff00ff; clear:both;}
```

float 속성

텍스트와 이미지 배치하기

왼쪽이나 오른쪽으로 배치하는 float 속성

웹 문서를 만들다 보면 문단과 이미지를 표시해야 할 경우가 있다. <P> 태그는 블록 레벨 요소이므로 이미지와 나란히 배치할 수 없다. 이럴때 float 속성을 사용하여 이미지를 표시한다.

float-text.html

```
<div id="container">
  <img src="images/cat.jpg">
  <h2>왼쪽이나 오른쪽으로 배치하는 float 속성</h2>
  <p>웹 문서를 만들다 보면 문단과 이미지를 나란히 표시해야 할 경우가 있다.
     &lt;P&gt; 태그는 블록 레벨 요소이므로 이미지와 나란히 배치할 수 없다.
     이럴때는 float 속성을사용하여 이미지를 표시한다.
  </p>
</div>
```

float 속성

텍스트와 이미지 배치하기

```
#container{
    width: 700px;
    height: 200px;
    margin: 0 auto;  /* 가로 가운데 정렬 */
    border: 1px solid #ccc;
    padding: 10px;
}
img{
    float: left;
    margin-right: 10px;
}
p{
    line-height: 2;
   text-align: justify;
}
```

2단 레이아웃 만들기

float 속성을 활용하여 레이아웃 만들기

2단 레이아웃 만들기

```
<body>
    <div id="container">
        <header>
            <h1>사이트 제목</h1>
        </header>

        <section>
            <h1>본문</h1>
        </section>

        <aside>
            <h1>사이드바</h1>
        </aside>

        <footer>
            <h1>푸터</h1>
        </footer>
    </div>
</body>
```

float-layout.html

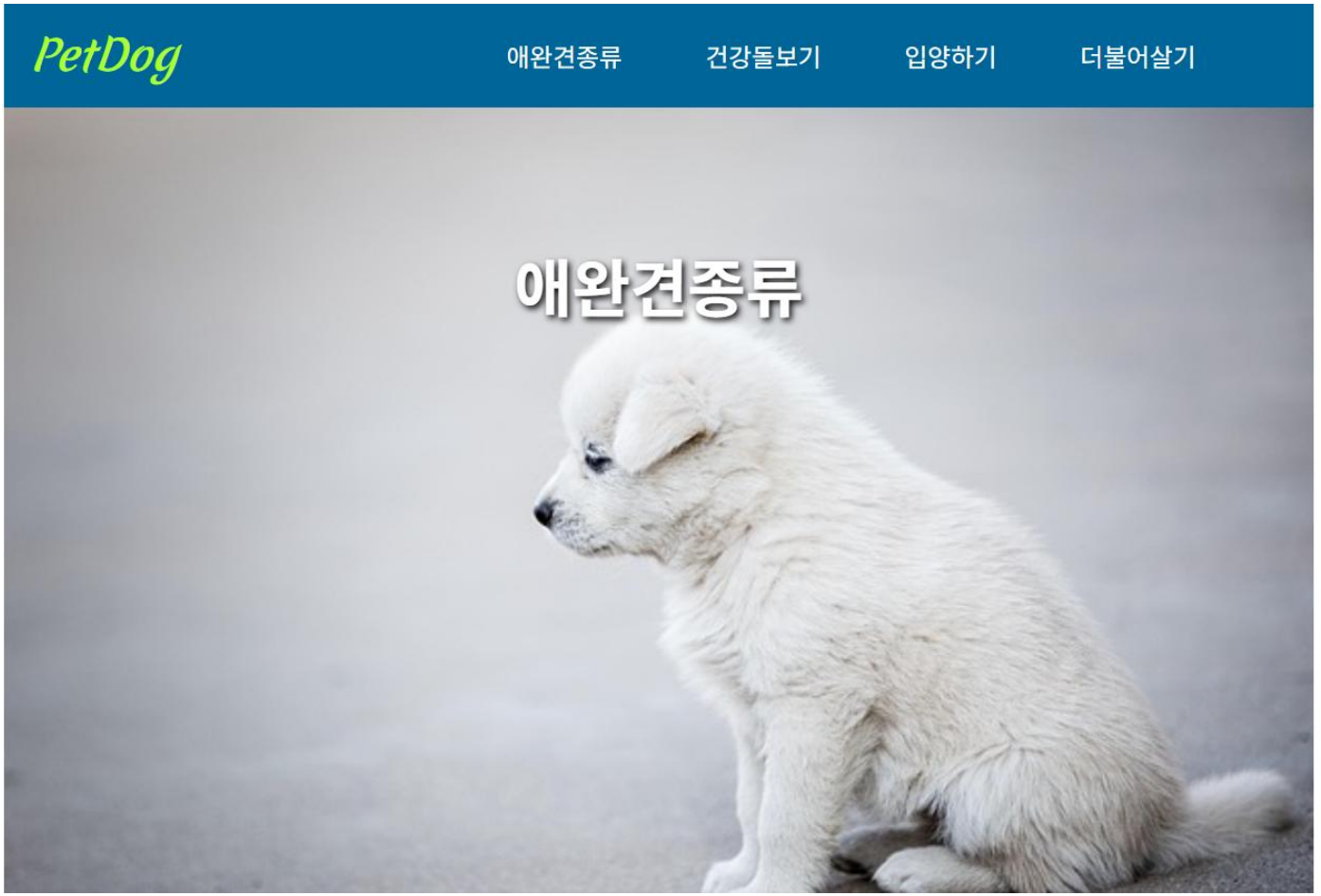
2단 레이아웃 만들기

```css
*{
    /* 전체 여백 초기화 */
    margin: 0; padding: 0;
}
#container{
    width: 1000px;
    margin: 0 auto;
}
h1{color: blue; padding: 10px 20px;}
header{
    /* container 1000px이므로 998+2(테두리) */
    width: 998px;
    height: 100px;
    border: 1px solid #ccc;
}
section{
    width: 660px;
    height: 600px;
    border: 1px solid #ccc;
    background: skyblue;
    float: left;
}
```

layout.css

```css
aside{
    width: 320px;
    height: 600px;
    border: 1px solid #ccc;
    background: greenyellow;
    float: right;
}
footer{
    width: 998px;
    height: 100px;
    border: 1px solid #ccc;
    clear: both;
}
```

navbar 메뉴



navbar 메뉴

```
<div id="container">
    <header>
        <div id="logo">
            <h1>PetDog</h1>
        </div>
        <nav>
            <ul>
                <li><a href="#">애완견종류</a></li>
                <li><a href="#">건강돌보기</a></li>
                <li><a href="#">입양하기</a></li>
                <li><a href="#">더불어살기</a></li>
            </ul>
        </nav>
    </header>
    <section>
        <h1>애완견종류</h1>
    </section>
</div>
```

nav-pet.html

navbar 메뉴

```css
/* 웹 폰트 임포트 경로는 맨 위에 위치시킴 */
@import url('https://fonts.googleapis.com/css2?family=Merienda:
*{
    margin: 0;
    padding: 0;
}
#container{
    width: 1000px;
    margin: 0 auto;
}
/* header 스타일 */
header{
    width: 100%;
    height: 80px;
    background: #069;
}
header #logo{
    width: 30%;  /* header 크기의 30% */
    float: left;
    line-height: 80px; /* 수직 정렬 - 가운데 */
}
```

pet.css

navbar 메뉴

```css
#logo h1{
    color: greenyellow;
    padding-left: 20px;
    font-style: italic;
    /* font-family: Nanum Gothic Coding; */
    font-family: Merienda;
}
header nav{
    width: 70%;
    float: right;
}
nav ul{
    list-style: none;
    text-align: right; /* 오른쪽 정렬 */
    margin-right: 60px;
}
nav ul li{
    display: inline-block;
    margin: 0 15px;
    line-height: 80px;
}
```

navbar 메뉴

```css
nav ul li a{
    color: white;
    text-decoration: none;
    padding: 10px;
    font-size: 1.2em;
}
ul li a:hover{color: greenyellow;}

/* section 스타일 */
section{
    width: 100%;
    height: 600px;
    background: url('../images/dog.jpg') no-repeat center;
    background-size: cover; /* 이미지 크기 조절 */
}
section h1{
    text-align: center;
    padding-top: 100px;
    font-size: 3em;
    color: white;
    /* text-shadow: 가로, 세로, 번짐정도, 색상 */
    text-shadow: 2px 2px 5px black;
}
```

box-sizing

box-sizing 속성 – 박스 너비 기준 정하기

- **content-box** : *width 속성 값을 콘텐츠 영역* 너비 값으로만 사용한다.

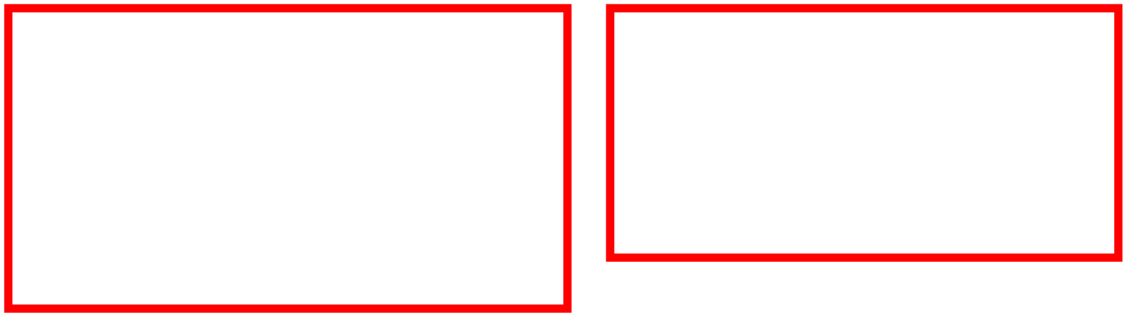
  예. { box-sizing: content-box }

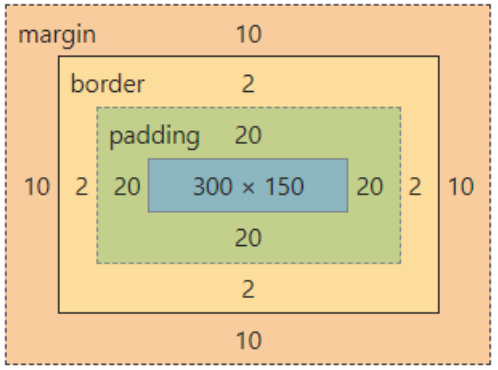
- **border-box** : *width 속성 값을 콘텐츠+패딩+테두리 영역까지 포함*한 전체 너비 값으로 사용한다

  예. { box-sizing: border-box }

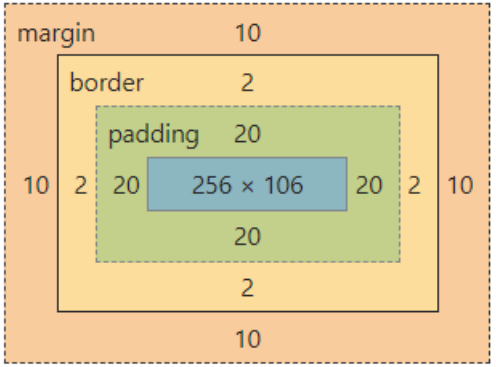
box-sizing

box-sizing 속성 – 박스 너비 기준 정하기





box-sizing: content-box;



box-sizing: border-box;

box-sizing

box-sizing.html

```
div{
    width: 300px;
    height: 150px;
    border: 5px solid red;
    margin: 10px;
    padding: 10px;
    float: left;
}
.box1{
    box-sizing: content-box;
}
.box2{
    box-sizing: border-box;
}
```

2단 레이아웃 만들기

box-sizing 적용

2단 레이아웃 만들기

```
<body>
    <div id="container">
        <header>
            <h1>사이트 제목</h1>
        </header>

        <section>
            <h1>본문</h1>
        </section>

        <aside>
            <h1>사이드바</h1>
        </aside>

        <footer>
            <h1>푸터</h1>
        </footer>
    </div>
</body>
```

layout2.html

2단 레이아웃 만들기

layout2.css

```css
*{
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}
#container{
  width: 1000px;
  margin: 0 auto;
}
header{
  /* width는 container와 같으면 생략 가능함 */
  /* width: 1000px; */
  height: 100px;
  border: 1px solid #ccc;
  padding: 10px;
}
```

```css
section{
  width: 640px;
  height: 600px;
  border: 1px solid #ccc;
  padding: 10px;
  background-color: beige;
  float: left;
}
aside{
  width: 360px;
  height: 600px;
  border: 1px solid #ccc;
  padding: 10px;
  background-color: aliceblue;
  float: right;
}
footer{
  height: 100px;
  border: 1px solid #ccc;
  padding: 10px;
  clear: both;
}
```

position 속성

position 속성

웹 문서 안에 요소들을 자유 자재로 배치하기 위한 속성

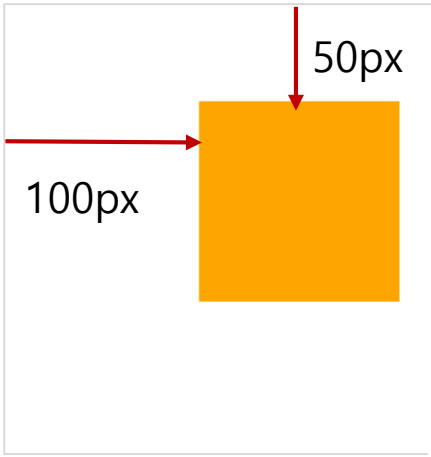
좌표를 이용해 각 요소를 배치할 수 있고, top, right, bottom, left로 지정

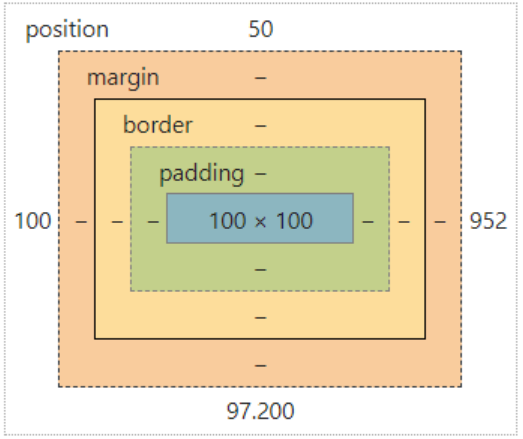
position: static | relative | absolute | fixed

속성 값	설명
releative	이전 요소에 자연스럽게 연결해 배치하며 위치 지정 가능
absolute	원하는 위치를 지정해 배치 relative 값을 사용한 상위 요소를 기준으로 위치를 지정해 배치함
fixed	지정한 위치에 고정해 배치

position 속성

- **absolute 속성**
  - 단독으로 사용하면 브라우저 창 기준
  - 부모 요소가 relative이면 부모요소 기준



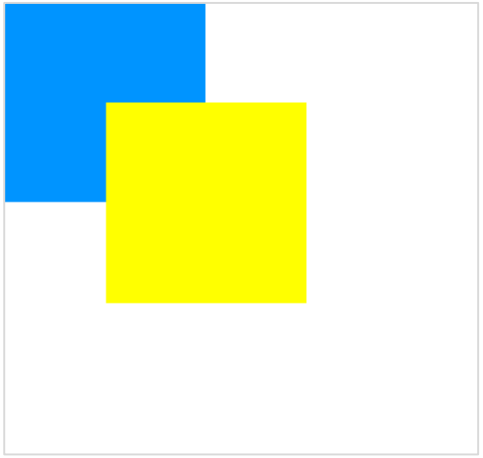


position-abs.html

```
<style>
    div{
        width:100px;
        height: 100px;
        background: orange;
        position: absolute;
        left: 100px;
        top: 50px;
    }
</style>
```

position 속성

- **relative 속성**



relative1.html

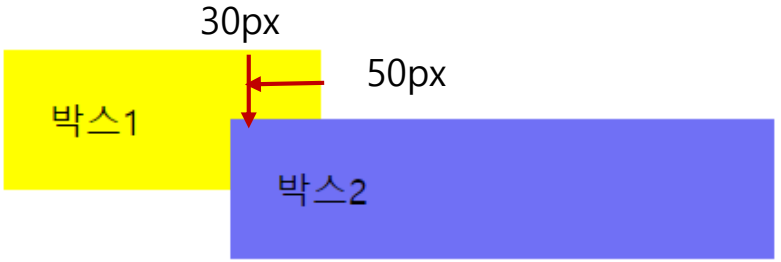
```
<style>
    *{margin: 0; padding: 0;}
    .box1{
        width: 100px;
        height: 100px;
        background: #0094ff;
    }
    .box2{
        width: 100px;
        height: 100px;
        background: #ffff00;
        position: relative;
        top: -50px;
        left: 50px;
    }
</style>
```

position 속성

- **relative 속성**

박스2의 원래 위치

박스2 위치 이동

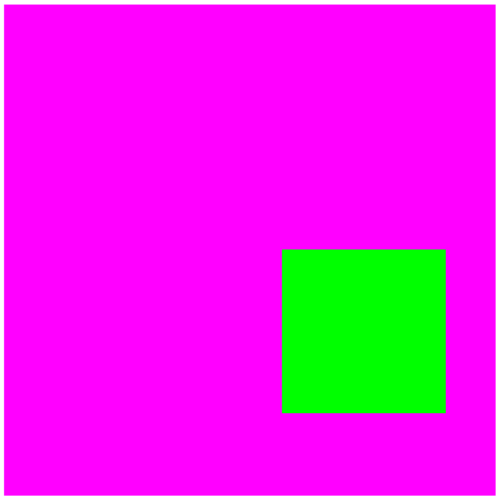


relative-2.html

```
.box1{
    width: 100px;
    background: #ff0;
    padding: 20px;
    margin-right: 10px;
    float: left;
}
.box2{
    width: 200px;
    background: rgb(112, 112, 245);
    padding: 20px;
    float: left;
    position: relative;
    top: 30px;
    left: -50px;
}
```

position 속성

- absolute 속성(부모 요소가 relative인 경우)



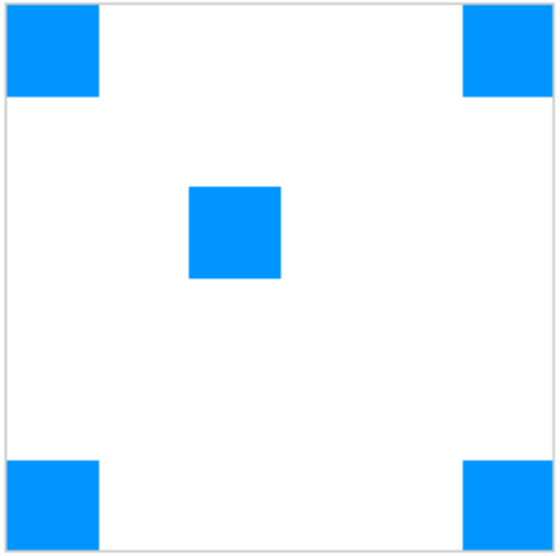
position-rel_abs.html

```
<div id="rec1">
    <div id="rec2"></div>
</div>
```

```
#rec1{
    width: 300px;
    height: 300px;
    background: #f0f;
    margin: 50px;
    /* 부모 요소가 relative */
    position: relative;
}
#rec2{
    width: 100px;
    height: 100px;
    background: #0f0;
    /* 절대 위치 */
    position: absolute;
    bottom: 50px;
    right: 30px;
}
```

position 속성

- absoluet 속성(부모 요소가 relative인 경우)



position-rel_abs2.html

```
<div id="wrap">
    <div class="box" id="coord1"></div>
    <div class="box" id="coord2"></div>
    <div class="box" id="coord3"></div>
    <div class="box" id="coord4"></div>
    <div class="box" id="coord5"></div>
</div>
```

position 속성

- absoluet 속성(부모 요소가 relative인 경우)

```
<style>
    #wrap{
        width: 300px;
        height: 300px;
        border: 1px solid #ccc;
        position: relative;
    }
    .box{
        width:50px;
        height: 50px;
        position: absolute;
        background: #0094ff;
    }
    #coord1{ top: 0; left: 0;}
    #coord2{ top: 0; right: 0;}
    #coord3{ bottom: 0; left: 0;}
    #coord4{ bottom: 0; right: 0;}
    #coord5{ left: 100px; top: 100px;}
</style>
```

기준이 되는 부모 요소

position 속성

fixed 속성

- 문서의 흐름과 상관없이 원하는 위치에 요소를 배치
- 부모 요소가 아닌 **브라우저 창이 기준**이 됨
- 브라우저 창 화면을 스크롤 하더라도 계속 같은 위치에 고정

fixed 값을 사용하는 요소 역시 absolute값을 사용하는 요소처럼 문서 흐름과는 상관없이 좌료로 위치를 결정하지만, 기존이 되는 요소가 부모요소가 아니라 브라우저 창이 기준이 됩니다 브라우저 창을 스크롤 하더라도 계속 고정되어 표시됩니다.

fixed 값을 사용하는 요소 역시 absolute값을 사용하는 요소처럼 문서 흐름과는 상관없이 좌료로 위치를 결정하지만, 기존이 되는 요소가 부모요소가 아니라 브라우저 창이 기준이 됩니다 브라우저 창을 스크롤 하더라도 계속 고정되어 표시됩니다.

fixed 값을 사용하는 요소 역시 absolute값을 사용하는 요소처럼 문서 흐름과는 상관없이 좌료로 위치를 결정하지만, 기존이 되는 요소가 부모요소가 아니라 브라우저 창이 기준이 됩니다 브라우저 창을 스크롤 하더라도 계속 고정되어 표시됩니다.

fixed 속성

position 속성

fixed 속성

```
<div id="container">
    <div id="content">
        <p>fixed 값을 사용하는 요소 역시 absolute값을 사용하는 요소처럼 문서 흐름과는 상관없이
        좌표로 위치를 결정하지만, 기존이 되는 요소가 부모요소가 아니라 브라우저 창이 기준이 됩니다
        브라우저 창을 스크롤하더라도 계속 고정되어 표시됩니다.</p>
        <p>fixed 값을 사용하는 요소 역시 absolute값을 사용하는 요소처럼 문서 흐름과는 상관없이
        좌표로 위치를 결정하지만, 기존이 되는 요소가 부모요소가 아니라 브라우저 창이 기준이 됩니다
        브라우저 창을 스크롤하더라도 계속 고정되어 표시됩니다.</p>
        <p>fixed 값을 사용하는 요소 역시 absolute값을 사용하는 요소처럼 문서 흐름과는 상관없이
        좌표로 위치를 결정하지만, 기존이 되는 요소가 부모요소가 아니라 브라우저 창이 기준이 됩니다
        브라우저 창을 스크롤하더라도 계속 고정되어 표시됩니다.</p>
        <p>fixed 값을 사용하는 요소 역시 absolute값을 사용하는 요소처럼 문서 흐름과는 상관없이
            좌표로 위치를 결정하지만, 기존이 되는 요소가 부모요소가 아니라 브라우저 창이 기준이 됩니다
            브라우저 창을 스크롤하더라도 계속 고정되어 표시됩니다.</p>
            <p>fixed 값을 사용하는 요소 역시 absolute값을 사용하는 요소처럼 문서 흐름과는 상관없이
                좌표로 위치를 결정하지만, 기존이 되는 요소가 부모요소가 아니라 브라우저 창이 기준이 됩니다
                브라우저 창을 스크롤하더라도 계속 고정되어 표시됩니다.</p>
    </div>
    <div id="fix"></div>
</div>
```

position 속성

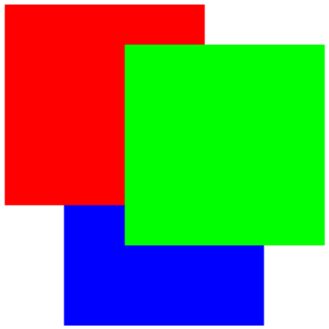
fixed 속성

```
#container{width: 1000px; margin: 0 auto;}
#fix{
    width: 100px;
    height: 160px;
    /* background: pink; */
    background: url(./images/bg3.png);
    position: fixed;
    top: 20px;
    right: 20px;
}
#content{width: 500px;}
```

position 속성

z-index 속성

- 요소 쌓는 순서 정하기
- z-index 값이 크면 작은 요소보다 위에 쌓인다.
- z-index 값을 명시하지 않으면 1부터 시작 1씩 커진다.



```
<body>
    <div class="box1"></div>
    <div class="box2"></div>
    <div class="box3"></div>
</body>
```

position 속성

```css
/* index 숫자가 클수록 위로 올라옴 */
div{
    width: 100px;
    height: 100px;
    font-size: 2em;
    position: absolute;
}
.box1{
    background: #ff0000;
    top: 50px;
    left: 50px;
    z-index: 2;
}
.box2{
    background: #00ff00;
    top: 70px;
    left: 110px;
    z-index: 3;
}
.box3{
    background: #0000ff;
    top: 110px;
    left: 80px;
    z-index: 1;
}
```

z-index 속성

z-index 속성

이미지의 z-index 속성값이 -1이므로 제목 뒤에 위치함

```
img{
    position: absolute;
    top: 10px;
    left: 20px;
    z-index: -1;
}
p{margin-top: 100px;}
```

```
<h1>z-index 속성</h1>

<img src="images/bg3.png" alt="열기구" width="80px" height="120px">

<p>이미지의 z-index 속성값이 -1이므로 제목 뒤에 위치함</p>
```