

# Position 속성

---

요소의 배치 방법을 결정하는 속성

`relative`(상대좌표), `absolute`(절대좌표), `fixed`(고정좌표) 방식이 있다.

각 방식에 따라 좌표가 설정되는 기준이 달라진다.

## #01. 좌표설정

`position`이 부여된 상태에서 요소의 좌표를 설정할 수 있다.

사용 가능한 속성으로는 `left`, `top`, `right`, `bottom`이 있다.

각 속성에 px이나 %단위로 수치값 부여한다.

`relative`, `absolute`에 따라 좌표의 기준 선정 방식이 변경된다.

### 1) relative

요소의 원래 위치를 기준으로 좌표가 설정된다.

주변 요소들은 해당 요소가 이동했다는 사실을 모른다.

### 2) absolute

브라우저의 끝을 기준으로 좌표가 설정된다.

주변 요소들은 이 요소의 존재 사실을 모르기 때문에 이 요소의 원래 위치를 매우기 위해 위치를 이동한다.

**부모요소에게 `position`속성이 `relative`나 `absolute`로 지정된 경우 부모를 기준으로 좌표가 설정된다.**

### 3) fixed의 특성

기본적으로 `absolute`와 동일하다.

스크롤이 이동하더라도 요소의 위치가 화면상에 고정되어 있다.

## #02. z-index

`position`속성으로 인해 요소가 화면상에서 겹치는 경우 `z-index` 속성에 부여된 값의 크기를 비교해서 순서가 결정된다.

값이 클 수록 위에 배치된다.

단순히 크기만을 비교할 뿐 반드시 순차적인 값을 사용할 필요는 없다.

## #03. HTML 요소 배치

로고, 메뉴, 사이드요소의 세 영역을 상위 요소 안에서 배치하는 방법.

아래 내용으로 HTML 파일을 실습하고 결과물 제출

## 1) 공통항목(common.css)

자주 사용될 속성들은 별도의 클래스로 미리 만들어둔다.(코드의 재사용)

```
* { padding: 0; margin: 0; }
.text-center { text-align: center; }
.text-right { text-align: right; }
.text-left { text-align: left; }
.inline { display: inline; }
.inline-block { display: inline-block; }
.pull-left { float: left; }
.pull-right { float: right; }
.clearfix:after { content: ''; display: block; clear: both; float: none; }
```

## 2) 기본 HTML 구성

모든 예제가 아래의 HTML을 기본 구조로 갖는다.

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <link rel="stylesheet" type="text/css" href="common.css" />
  <style>
    .container {
      width: 1000px;
      background-color: #eee;
      margin: auto;
    }

    .logo {
      width: 300px;
      height: 100px;
      background-color: #ff6600;
    }

    .side1 {
      width: 80px;
      height: 80px;
      background-color: #ff00ff;
    }

    .side2 {
      width: 80px;
      height: 80px;
      background-color: #00ff00;
    }
  </style>
</head>
```

```
<body>
  <div class="container">
    <div class="logo">logo</div>
    <div class="side1">side1</div>
    <div class="side2">side2</div>
  </div>
</body>
</html>
```

## 2) case1 : 모두 왼쪽정렬

미리 만들어 둔 클래스만으로 float 처리

```
<div class="container clearfix">
  <div class="logo pull-left">logo</div>
  <div class="side1 pull-left">side1</div>
  <div class="side2 pull-left">side2</div>
</div>
```

## 3) case2 : 로고는 왼쪽, 메뉴 요소들은 오른쪽

```
<div class="container clearfix">
  <div class="logo pull-left">logo</div>
  <div class="side1 pull-right">side1</div>
  <div class="side2 pull-right">side2</div>
</div>
```

## 4) case3 : 로고는 중앙, 메뉴 요소들은 양 끝을 기준으로 간격 조절

```
.logo {
  width: 300px;
  height: 100px;
  background-color: #ff6600;
  /* (추가) 절대좌표 방식 */
  position: absolute;
  /* (추가) box의 좌측 상단 꼭지점을 부모 중앙에 맞춤 */
  left: 50%;
  top: 50%;
  /* (추가) 스스로의 반만큼 반대로 이동 */
  margin-left: -150px;
  margin-top: -50px;
}
```

```
<div class="container clearfix">
  <div class="logo">logo</div>
  <div class="side1 pull-left">side1</div>
  <div class="side2 pull-right">side2</div>
</div>
```

## 5) case4 : 모두 가운데에서 출발. 간격조절

모든 요소를 부모의 중앙으로 모은 후, margin으로 간격을 조절.

```
.container {
  width: 1000px;
  background-color: #eee;
  margin: auto;

  /* (추가) .logo의 좌표 기준점 설정 */
  position: relative;
  height: 150px;
}

.logo {
  width: 300px;
  height: 100px;
  background-color: #ff6600;
  position: absolute;
  left: 50%;
  top: 50%;
  margin-left: -150px;
  margin-top: -50px;
}

.side1 {
  width: 80px;
  height: 80px;
  background-color: #ff00ff;

  /* (추가) */
  position: absolute;
  left: 50%;
  top: 50%;
  margin-left: -240px;
  margin-top: -40px;
}

.side2 {
  width: 80px;
  height: 80px;
  background-color: #00ff00;

  /* (추가) */
```

```
position: absolute;
left: 50%;
top: 50%;
margin-left: 160px;
margin-top: -40px;
}
```

```
<!-- float 사용 안함 -->
<div class="container">
  <div class="logo">logo</div>
  <div class="side1">side1</div>
  <div class="side2">side2</div>
</div>
```