



요소 정렬

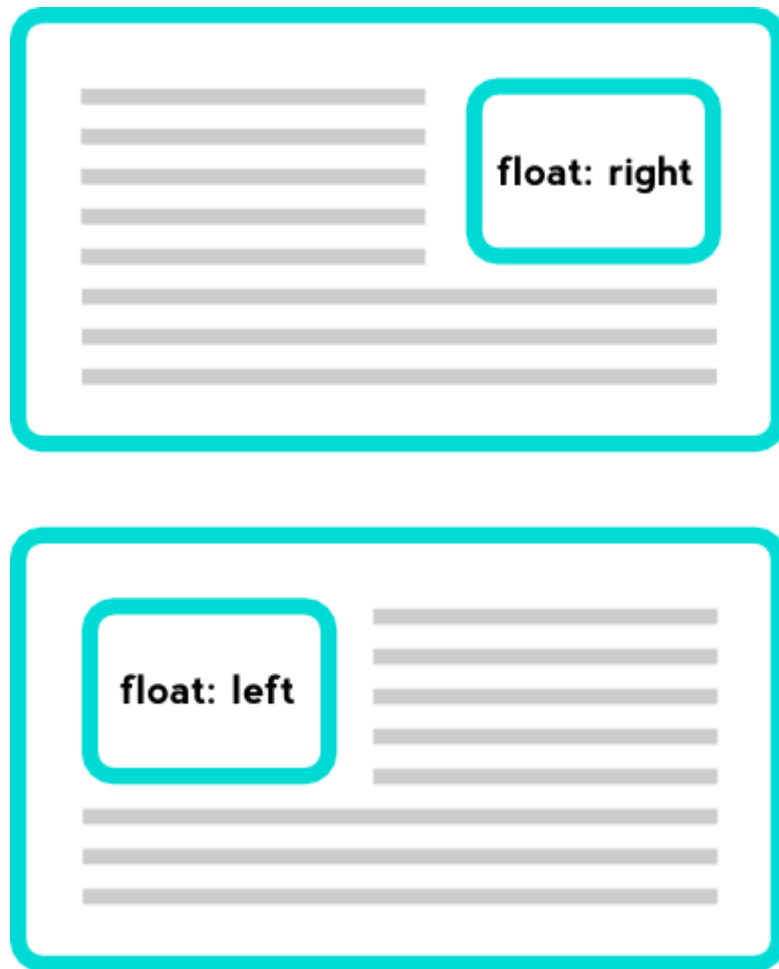
float 프로퍼티는 주로 레이아웃을 구성할 때 블록 레벨 요소를 가로 정렬하기 위해 사용되는 중요한 기법이다. flexbox 레이아웃을 사용한다면 더욱 간단하게 정렬을 구현할 수도 있지만 flexbox 레이아웃을 지원하지 않는 IE를 고려해야 한다면 float 프로퍼티를 사용해야 한다.

float 프로퍼티는 본래 아래 예제와 같이 이미지와 텍스트가 있을 때, 이미지 주위를 텍스트로 감싸기 위해 만들어진 것이다.

```
<!DOCTYPE html>
<html>
<head>
  <style>
    img {
      float: left;
      margin-right: 10px;
    }
  </style>
</head>
<body>
  
  <div>Lorem ipsum dolor sit amet, consectetur adipisicing el
</body>
</html>
```

float 프로퍼티는 해당 요소를 다음 요소 위에 떠 있게(부유하게) 한다. 여기서 떠 있다 (float)는 의미는 요소가 기본 레이아웃 흐름에서 벗어나 요소의 모서리가 페이지의 왼쪽이나 오른쪽에 이동하는 것이다. float 프로퍼티를 사용할 때 요소의 위치를 고정시키는 position 프로퍼티의 absolute를 사용하면 안된다.

프로퍼티값	Description
none	요소를 떠 있게 하지 않는다. (기본값)
right	요소를 오른쪽으로 이동시킨다
left	요소를 왼쪽으로 이동시킨다.



정렬

float 프로퍼티를 사용하지 않은 블록 요소들은 수직으로 정렬된다. `float:left;` 프로퍼티를 사용하면 왼쪽부터 가로 정렬되고, `float:right;` 프로퍼티를 사용하면 오른쪽부터 가로 정렬된다.

오른쪽 가로 정렬의 경우, 먼저 기술된 요소가 가장 오른쪽에 출력되므로 출력 순서가 역순이 된다.

```

<!DOCTYPE html>
<html>
<head>
  <style>
    .box {
      color: white;
      font-weight: bold;
      font-size: 50px;
      border-radius: 6px;
      width: 100px;
      height: 100px;
      margin: 10px;
      padding: 10px;
    }
    .d1, .d2 {
      float: left;
    }
    .d1 {
      background: red;
    }
    .d2 {
      background: orange;
    }
    .d3, .d4 {
      float: right;
    }
    .d3 {
      background: red;
    }
    .d4 {
      background: orange;
    }
  </style>
</head>
<body>
  <div class="container">
    <div class="box d1"> 1 </div>
    <div class="box d2"> 2 </div>
  </div>

```

```

    <div class="box d3"> 3 </div>
    <div class="box d4"> 4 </div>
  </div>
</body>
</html>

```

float 프로퍼티는 좌측, 우측 가로 정렬만 할 수 있다. 중앙 가로 정렬은 margin 프로퍼티를 사용해야 한다.

```

div {
  width: 960px;
  margin: 0 auto;
}

```

width

width 프로퍼티의 기본값은 100%이므로 width 프로퍼티값을 지정하지 않은 block 요소는 부모 요소의 가로폭을 가득 채운다.

```

<!DOCTYPE html>
<html>
<head>
  <style>
    .box {
      color: white;
      font-weight: bold;
      font-size: 30px;
      line-height: 50px;
      height: 50px;
      margin: 0 10px;
      padding: 10px;
    }
    .d1 {

```

```

        background: red;
    }
    .d2 {
        background: orange;
    }
</style>
</head>
<body>
    <div class="box d1"> div </div>
    <div class="box d2"> div </div>
</body>
</html>

```

width 프로퍼티를 선언하지 않은 block 레벨 요소에 float 프로퍼티가 선언되면 width가 inline 요소와 같이 content에 맞게 최소화되고 다음 요소 위에 떠 있게(부유하게) 된다.

```

<!DOCTYPE html>
<html>
<head>
    <style>
        .box {
            color: white;
            font-weight: bold;
            font-size: 30px;
            line-height: 50px;
            height: 50px;
            margin: 0 10px;
            padding: 10px;
        }
        .d1 {
            float: left;
            background: red;
        }
        .d2 {
            background: orange;
        }
    </style>
</head>
<body>
    <div class="box d1"> div </div>
    <div class="box d2"> div </div>
</body>
</html>

```

```

    </style>
</head>
<body>
  <div class="box d1"> float: left; </div>
  <div class="box d2"> div </div>
</body>
</html>

```

위 예제를 살펴보면 d1 클래스 요소에는 `float: left;` 를 선언하였고 d2 클래스 요소에는 `float`를 선언하지 않았다. 이때 d1 클래스 요소는 width가 inline 요소와 같이 content에 맞게 최소화되고 다음 요소인 d2 클래스 요소 위에 떠 있게(부유하게) 된다.

주의할 것은 d1 클래스 요소가 d2 클래스 요소 위에 떠 있게 되어도 d2 클래스 요소의 width는 d1 클래스 요소가 차이한 width만큼 줄어들지 않고 100%를 유지한다는 것이다. 이는 d2 클래스 요소는 `float`를 선언하지 않았기 때문에 본래의 width를 유지하기 때문이다. 따라서 d2 클래스 요소는 본래의 width(100%)를 유지한 상태에서 d1 클래스 요소가 그 위에 위치한다.

float 프로퍼티 관련 문제 해결 방법

float 프로퍼티가 선언된 요소와 float 프로퍼티가 선언되지 않은 요소간 margin이 사라지는 문제

위 예제를 보면 두 요소는 차례대로 정렬된 것처럼 보이지만 사실은 `float` 프로퍼티가 선언된 요소가 다음 요소 위에 떠 있는(부유하고 있는) 상태이다. 따라서 두 요소간의 `margin`은 제대로 표현되지 않는다.

이것은 두번째 요소에 `float` 프로퍼티를 선언하지 않았기 때문에 발생하는 박스 모델 상의 문제이다. 이 문제를 해결하는 가장 쉬운 방법은 `float` 프로퍼티를 선언하지 않은 요소(.d2)에 `overflow: hidden` 프로퍼티를 선언하는 것이다.

`overflow: hidden` 프로퍼티는 자식 요소가 부모 요소의 영역보다 클 경우 넘치는 부분을 안 보이게 해주는 역할을 하는데 여기서는 `float` 프로퍼티가 없어서 제대로 표현되지 못하는 요소를 제대로 출력해준다.

```

<!DOCTYPE html>
<html>

```

```

<head>
  <style>
    .box {
      color: white;
      font-weight: bold;
      font-size: 30px;
      line-height: 50px;
      height: 50px;
      margin: 0 10px;
      padding: 10px;
    }
    .d1 {
      float: left;
      background: red;
    }
    .d2 {
      overflow: hidden;
      background: orange;
    }
  </style>
</head>
<body>
  <div class="box d1"> float: left; </div>
  <div class="box d2"> div </div>
</body>
</html>

```

두번째 요소에도 float 프로퍼티를 선언하면 `overflow: hidden` 프로퍼티는 선언하지 않아도 되지만 너비가 최소화된다.

float 프로퍼티가 선언된 자식 요소를 포함하는 부모 요소의 높이가 정상적으로 반영되지 않는 문제

아래 예제를 보면 float 프로퍼티가 선언된 두개의 자식 요소를 포함하는 부모 요소의 높이가 정상적인 값을 가지지 못하는 문제가 발생한다. float 요소는 일반적인 흐름 상에 존재하지 않기 때문에 float 요소의 높이를 알 수 없기 때문인데 이 문제는 부모 요소 이후에 위치하는 요소의 정렬에 문제를 발생시킨다.

```

<!DOCTYPE html>
<html>
<head>
  <style>
    .container {
      color: white;
      text-align: center;
      padding: 10px;
      background-color: #def0c2;
    }
    .d1, .d2 {
      float: left;
      width: 50%;
      padding: 20px 0;
    }
    .d1 {
      background-color: #59b1f6;
    }
    .d2 {
      background-color: #ffb5b4;
    }
  </style>
</head>
<body>
  <div class="container">
    <div class="d1">1</div>
    <div class="d2">2</div>
  </div>
  <div style="background:red;padding:10px;color:white;">3</div>
</body>
</html>

```

이 문제를 해결하는 가장 쉬운 방법은 float 프로퍼티가 선언된 자식 요소의 부모 요소 (.container)에 `overflow: hidden` 프로퍼티를 선언하는 것이다.


```
.container {
  ...
  overflow: hidden;
}
```

다른 방법으로 부모 요소(.container)에 float 프로퍼티를 선언하는 방법도 있다. 하지만 부모 요소의 너비는 float된 두개의 자식요소의 콘텐츠를 표현할 수 있는 만큼만으로 작게 줄어들게 된다. 권장할 수 있는 방법은 아니다.

container 영역이 끝나기 직전 빈 요소를 만들고 `clear:both` 를 설정하는 방법도 가능하다. 하지만 의미 없는 빈 요소를 사용하여야 하기 때문에 이 방법 역시 권장할 수 있는 방법은 아니다.

```
<!DOCTYPE html>
<html>
<head>
  <style>
    .container {
      color: white;
      text-align: center;
      padding: 10px;
      background-color: #def0c2;
      /*overflow: hidden;*/
    }
    .d1, .d2 {
      float: left;
      width: 50%;
      padding: 20px 0;
    }
    .d1 {
      background-color: #59b1f6;
    }
    .d2 {
      background-color: #ffb5b4;
    }
    .clear {
```

```

        height: 0;
        clear: both;
    }
</style>
</head>
<body>
    <div class="container">
        <div class="d1">1</div>
        <div class="d2">2</div>
        <div class="clear"></div>
    </div>
    <div style="background:red; padding:10px; color:white;">3</div>
</body>
</html>

```

`overflow: hidden;` 과 함께 많이 사용되는 방법은 `::after` 가상 요소 선택자를 이용하는 것이다. 가상 요소 선택자는 CSS2 문법(`:after`)과 CSS3 문법(`::after`)이 있는데 IE8까지 지원하는 CSS2 문법을 사용하는 편이 좋다.

```

<!DOCTYPE html>
<html>
<head>
<style>
    .container {
        color: white;
        text-align: center;
        padding: 10px;
        background-color: #def0c2;
        /*overflow: hidden;*/
    }
    .clearfix:after {
        content: "";
        display: block;
        clear: both;
    }
    .d1, .d2 {

```

```

        float: left;
        width: 50%;
        padding: 20px 0;
    }
    .d1 {
        background-color: #59b1f6;
    }
    .d2 {
        background-color: #ffb5b4;
    }
</style>
</head>
<body>
    <div class="container clearfix">
        <div class="d1">1</div>
        <div class="d2">2</div>
    </div>
    <div style="background:red;padding:10px;color:white;">3</div>
</body>
</html>

```

부모 요소에 위 예제와 같이 사전에 작성한 clearfix 클래스만 추가하거나, 해당 요소를 선택하여 클리어 문법을 선언하면 되기 때문에 가장 깔끔하고 간편하다. 이 방법을 사용하는 것을 추천한다.

```

.clearfix:after {
    content: "";
    display: block;
    clear: both;
}

/* or */

selector:after {
    content: "";
    display: block;

```

```
clear: both;
}
```

또 다른 방법은 float 프로퍼티 대신 `display: inline-block;` 을 선언하는 것이다. 주의해야 할 점은 inline-block 프로퍼티 요소를 연속 사용하는 경우, 두 요소 사이에 정의하지 않은 공백(4px)가 자동 지정되는 것이다.

```
<!DOCTYPE html>
<html>
<head>
  <style>
    .container {
      color: white;
      text-align: center;
      padding: 10px;
      background-color: #def0c2;
    }
    .d1, .d2 {
      display: inline-block;
      width: 50%;
      padding: 20px 0;
    }
    .d1 {
      background-color: #59b1f6;
    }
    .d2 {
      background-color: #ffb5b4;
    }
  </style>
</head>
<body>
  <div class="container">
    <div class="d1">1</div>
    <div class="d2">2</div>
  </div>
  <div style="background:red;padding:10px;color:white;">3</div>
```

```
</body>
</html>
```

위 예제를 보면 .d1, .d2 요소에 `display: inline-block;` 을 선언하여 텍스트와 같이 배치되도록 하였지만 두 요소 사이에 정의하지 않은 공백(4px)이 자동 지정되어 부모 요소의 width를 초과하여 가로 정렬이 되지 않았다.

이 현상을 회피하기 위해 부모 요소에 `font-size: 0;` 을 선언하여 두 요소 사이에 정의하지 않은 공백을 제거한다.

```
<!DOCTYPE html>
<html>
<head>
  <style>
    .container {
      color: white;
      text-align: center;
      padding: 10px;
      background-color: #def0c2;
      /* 폰트 사이즈를 0으로 지정하여 두 요소 사이에 정의하지 않은 공백을
      font-size: 0;
    }
    .d1, .d2 {
      display: inline-block;
      width: 50%;
      padding: 20px 0;
      /* 폰트 사이즈를 재지정 */
      font-size: 1rem;
    }
    .d1 {
      background-color: #59b1f6;
    }
    .d2 {
      background-color: #ffb5b4;
    }
  </style>
</head>
<body>
```

```
</style>
</head>
<body>
  <div class="container">
    <div class="d1">1</div>
    <div class="d2">2</div>
  </div>
  <div style="background:red;padding:10px;color:white;">3</div>
</body>
</html>
```

좀 더 자세한 회피 방법은 [Fighting the Space Between Inline Block Elements](#)를 참조하기 바란다.