

## CSS Position 속성 값(value)의 종류

CSS Position 속성 값은 5 가지가 존재한다. **static**, absolute, relative, fixed, inherit 가 각각 그것이다. 각각에 대해 간단히 살펴보면

1. static: Position 속성 값으로 아무 것도 지정하지 않는 경우 default 값으로 사용된다. CSS 의 Normal-flow(자세한 건 뒤에서 설명)에 따라 배치된다.
2. absolute: 부모 요소(태그) 위치의 상대값에 따라 배치된다. 단, 부모 요소가 static 인 경우에는 조상(ancestor) 요소의 position 속성이 static 이 아닐 때까지 거슬러 올라가 배치된다. absolute 로 정의된 요소는 CSS Normal-flow 의 흐름에서 배제된다.
3. relative: 자신이 원래 있어야 할 위치의 상대값(offset)에 따라 배치된다. offset 은 top,left,right,bottom 속성값을 활용한다. 즉 "left:20px"이라고 정의되면 원래 있어야 할 좌측 위치에서 20px 떨어진 곳에 위치하는 것이다.offset 을 지정하지 않으면 static 과 동일한 기능을 수행한다.
4. fixed: 브라우저의 창에 상대적인 위치(offset)에 보이도록 한다. 즉, 화면 스크롤과 상관없이 항상 화면의 정해진 위치에 정보가 나타난다.
5. inherit: 부모 element 의 position 속성을 물려받는다. **IE7 까지는 지원하지 않으므로 해당브라우저를 고려해야 하는 경우는 사용하지 않도록 한다.**

### Normal-flow 란?

**Normal flow** 라하면 일반적인 상황에서 각각의 요소들이 배치되는 순서(흐름)를 뜻하는 말이다. HTML 의 element 들은 block element 와 inline element 로 구분할 수 있다. 여기에서는 position 속성에 집중할 것이기 때문에 block element 에 대해서만 언급하도록 한다. block element 에 해당하는 html element 들에는 자주 사용되는 항목기준으로 h1~h6, div, p, ul, li, table 등이 있다. 이러한 블락요소(element)들은 내부에 또 다른 block-element 나 inline-element 를 포함할 수 있다. 블록 요소들은 기본적으로 상하로 배치된다. 따라서

```
1 <style>
2   div {width:200px;height:200px;}
3   div.testdiv1 { background-color:red}
4   div.testdiv2 { background-color:yellow}
5 </style>
```

```
6    <div class="testdiv1"> </div>
```

```
7    <div class="testdiv2"> </div>
```

위와 같은 코드가 있는 경우 각각의 div 는 어떻게 배치될까? 가로로 나란히 배치될까? 아니면 위아래로 배치될까? 앞서 이야기했듯이 블록요소들은 기본적으로 상하로 배치된다. 이렇게 각각의 요소가 정해진 흐름에 따라 배치되는 방식을 normal-flow 라고 한다.

## static

static 은 position 속성 값의 default 값이다. 따라서 static 을 지정하는 경우에는 별도로 스타일을 지정하지 않는다. 일반적인 normal-flow 에 따라서 각각의 element 들을 배치하는 것만으로도 layout 지정이 가능한 경우 주로 사용된다. static 을 사용하게 되면 자식 element 에 absolute 를 적용한다 하더라도 원하는대로 배치되지 않는 것을 명심하자. static 을 사용하는 경우에는 relative, fixed, absolute 처럼 top, left, right, bottom 과 같은 offset 속성을 적용할 수 없다.

```
1    <style>
```

```
2    div {width:200px;height:200px;}
```

```
3    div.testdiv1 { background-color:red;top:30px;}
```

```
4    div.testdiv2 { background-color:yellow}
```

```
5    </style>
```

```
6    <div class="testdiv1"> </div>
```

```
7    <div class="testdiv2"> </div>
```

위의 코드를 브라우저에서 확인하면 top 속성을 적용한다 하더라도 해당 값이 적용되지 않는 것을 확인할 수 있다.

## absolute

absolute 는 그 말의 의미처럼 "절대적"이라는 뜻과 유사하게 동작하기는 하지만 엄밀히 말해서 같지는 않으므로 그 말의 의미를 정확하게 파악해야 한다. position="absoulte" 값의 의미를 다음 사례를 통해 좀 더 면밀히 알아보도록 하자.

```

1    <style>
2    div {width:200px;height:200px;}
3    div.testdiv1 { background-color:red;position:absolute;top:30px;}
4    div.testdiv2 { background-color:yellow}
5    </style>
6    <div class="testdiv1"> </div>
7    <div class="testdiv2"> </div>

```

div.testdiv1 은 position 값이 absolute 이므로 상대적인 offset 지정이 가능하다. 따라서 top:30px; 부분이 제대로 동작한다. div.testdiv1 은 색상이 붉은 색이면서 상단에서 30px 떨어진 박스가 된다. 그렇다면 div.testdiv2 는 왜 div.testdiv1 아래에 배치되지 않고 서로 겹쳐서 표현되는 것일까? 아까 앞에서 언급했던 것을 기억하는가? position 속성 값이 absolute 로 표현되면 normal-flow 에서 제외된다는 것을 ... 따라서 div.testdiv2 는 마치 div.testdiv1 이 없는 것처럼 normal-flow 에 따라 배치되는 것이다.

다른 사례)

```

1    <style>
2    div {width:200px;height:200px;}
3    div.testdiv1 { background-color:red;position:absolute;top:500px;}
4    div.testdiv2 { background-color:yellow;}
5    div.testdiv3 { background-color:blue;}
6    p { position:absolute; top:50px; left:30px; }
7    </style>
8    <div class="testdiv1">
9        <p> hello </p>
10    </div>
11    <div class="testdiv2">
12        <p> Good Morning! </p>
13    </div>
14    <div class="testdiv3">
15        <p> How are you? </p>
16    </div>

```

1~7 라인은 Style 을 지정하는 부분이다. 명확한 구분을 위해 각 div 별로 색상을 주었다. absolute 값을 지정한 부분을 유심히 보도록 한다.

8~10 라인코드먼저 보자.

div.testdiv1 은 우선 position 이 absolute 이다. 따라서 위로부터 500px 떨어진 곳에 자리잡게 된다. 그리고 그 안에 있는 p 태그는 마찬가지로 position 속성 값이 absolute 이면서 top 50px, left 30px 값을 가지고 있기 때문에 div.testdiv1 안에서 해당 offset 만큼 떨어진 곳에 위치한다.

11~13 라인코드를 살펴보면

div.testdiv2 는 normal-flow 에 따라 상단에 노란색박스로 위치한다.(div.testdiv1 은 position 속성 값이 absolute 이므로 normal-flow 에서 제외됨), 그 안에 있는 p 태그 역시 offset 만큼 떨어져 배치된다. 이 부분도 이해하는데 무리 없을 것이다.

그럼 마지막으로 14~16 라인을 보자, Normal-flow 에 따라 div.testdiv3 파란색 박스는 노란색 박스 하단에 위치한다. 그렇다면 div.testdiv3 태그안의 p 태그에 있는 **How are you?**는 어디로 갔을까? 결과를 보면 알 수 있듯이 div.testdiv2 태그안의 good morning 과 동일한 위치에 출력된다. 왜 이런 현상이 발생할까? 부모요소의 position 속성이 static 인 경우에는 absolute 속성의 offset 값은 position 속성이 static 이 아닌 가장가까운 조상요소를 기준으로 배치된다. 위의 예제에서는 div.testdiv3 의 position 속성은 static(default 값)이기 때문에 그보다 상위요소(body 태그)를 기준으로 "How are you"가 출력되는 것이다.

## relative

absolute 에 대한 개념을 명확히 이해했다면 relative 는 크게 어렵지 않다. relative 가 absolute 와의 차이는 다음 두 가지를 명확히 이해하면 된다.

1. relative 는 normal-flow 를 따른다.(단 offset 이 지정된 경우에는 offset 이 지정되기 전 기준으로 다음 element 가 배치된다)
2. relative 는 부모의 position 속성과 상관없이 설정된 offset 을 따른다.

다음 코드를 통해 좀 더 확실히 알아보도록 하자.

```

1    <style>
2    div {width:200px;height:200px;}
3    div.testdiv1 { background-color:red;position:relative;top:400px;}
4    div.testdiv2 { background-color:yellow;}
5    p { position:relative; top:50px; left:30px; }
6    </style>
7    <div class="testdiv1">
8        <p> hello </p>
9    </div>
10   <div class="testdiv2">
11       <p> Good Morning! </p>
12   </div>

```

간단히 설명하자면, div.testdiv1 은 relative 이면서 top 이 400px 이기 때문에 상단으로부터 400px 떨어진 곳에 위치한다. 그리고 div.testdiv2 는 원래 div.testdiv1 이 있어야 하는 위치(offset 값이 없다는 가정하에)에서 normal-flow 에 따라 그 아래에 배치된다.

## fixed

fixed 는 어떻게 보면 가장 이해하기 쉽다. 화면의 특정 위치에 항상 보여주고 싶은 경우 사용하면 유용하다.

```

1    <style>
2    div {width:200px;height:200px;}
3    div.testdiv1 { background-color:red;position:relative;}
4    div.testdiv2 { background-color:yellow;}
5    p { position:fixed; top:150px; left:30px; }
6    </style>
7    <div class="testdiv1">
8        <p> hello </p>
9        <p> 동해물과 백두산이</p>
10   </div>
11   <div class="testdiv2">
12       <p> Good Morning! </p>

```

13 </div>

p 태그의 style 은 position 속성이 fixed 이다. 부모의 position 속성값이 뭐이든지간에 p 태그는 항상 동일한 곳에 위치한다. 브라우저의 창을 줄여 위아래로 스크롤해보면 더욱 분명하게 알 수 있을 것이다.

---

## float 플롯

블럭 요소들을 가로 정렬 할때 사용 하는 속성입니다.

float 은 요소를 공중으로 붕 띄운다고 생각하시면 됩니다.

float:left 는 왼쪽에 띄우고

float:right 는 오른쪽에 띄웁니다.

예를 들어 2 개의 블럭 요소가 있다고 가정하고

첫번째 요소에 float 을 적용하면 그 요소는 실제로 그렇게 보이진 않지만 공중으로 붕 띄게 됩니다.

두번째 요소는 첫번째 요소가 있던 곳으로 올라가게 됩니다.

```
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
2 "http://www.w3.org/TR/html4/loose.dtd">
3 <html xmlns="http://www.w3.org/1999/xhtml">
4   <head>
5     <meta http-equiv="Content-Type" content="text/html; charset=utf
6     <title>New Web Project</title>
7     <style type="text/css">
8       *{margin:0;padding:0}
9       div{border:1px solid red;}
10      #box1 {float:left;}
11    </style>
12  </head>
13  <body>
14    <div id="box1">내용1</div>
15    <div id="box2">내용2</div>
16  </body>
17 </html>
```



그런데

위의 예제를 보면 첫번째 요소에 float 적용 되었음에도 두번째 요소가 설명한 것 처럼 첫번째 위치에 있지 않고 첫번째 요소 뒤에 있는 것 처럼 보입니다.

실제 두번째 요소는 첫번째 요소의 위치에 있는 것이지만 첫번째 요소안의 내용 때문에 두번째 요소가 옆으로 밀리는 것 처럼 보이는 것 뿐입니다.

```

1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
2 "http://www.w3.org/TR/html4/loose.dtd">
3 <html xmlns="http://www.w3.org/1999/xhtml">
4   <head>
5     <meta http-equiv="Content-Type" content="text/html; charset=utf-
6     <title>New Web Project</title>
7     <style type="text/css">
8       *{margin:0;padding:0}
9       div{border:1px solid red;}
10      #box1{float:left;}
11      #box2{margin-left:20px;border-color: blue;}
12    </style>
13  </head>
14  <body>
15    <div id="box1">내용1</div>
16    <div id="box2">내용2</div>
17  </body>
18 </html>

```



box2 가 margin-left 를 적용 했을때 왼쪽에서 떨어지는 기준이 첫번째 요소의 위치인것을 알수 있습니다.

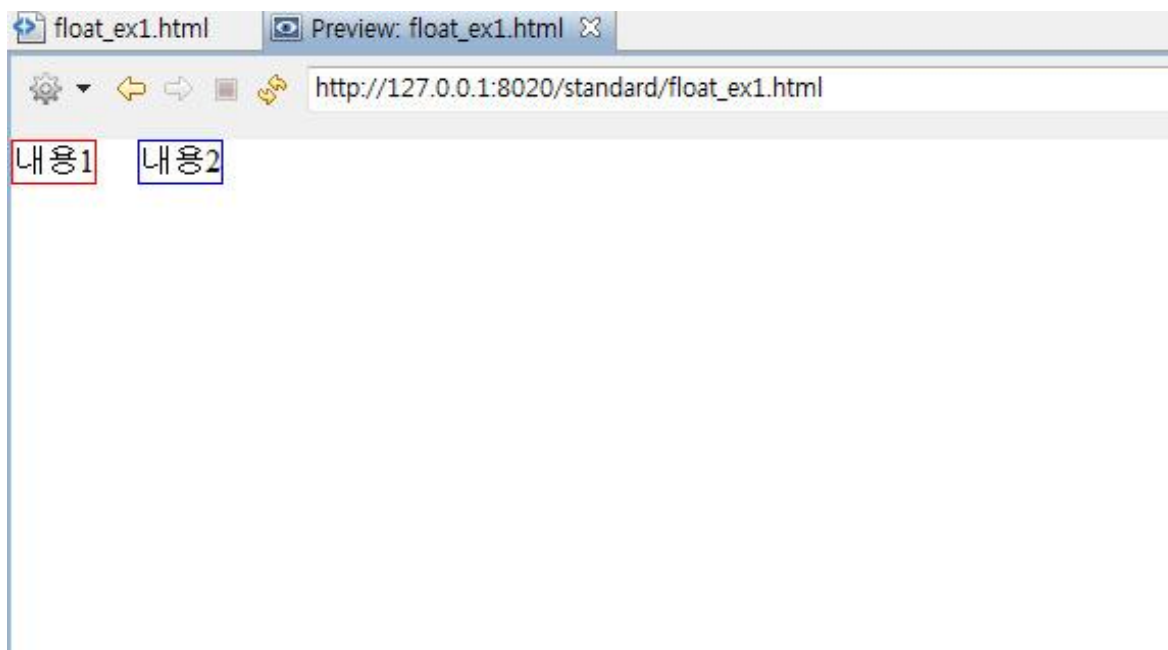
즉,box2 가 box1 옆에 있는것 처럼 보이지만 실제로는 box1 위치에 있다는 것입니다.



```

1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
2 "http://www.w3.org/TR/html4/loose.dtd">
3 <html xmlns="http://www.w3.org/1999/xhtml">
4   <head>
5     <meta http-equiv="Content-Type" content="text/html; charset=utf-
6     <title>New Web Project</title>
7     <style type="text/css">
8       *{margin:0;padding:0}
9       div{border:1px solid red;}
10      #box1{float:left;}
11      #box2{float:left;margin-left:20px;border-color: blue;}
12    </style>
13  </head>
14  <body>
15    <div id="box1">내용1</div>
16    <div id="box2">내용2</div>
17  </body>
18 </html>

```



box2 를 box1 의 옆으로 20px 띄어정렬시키려면

box2 도 float 을 적용해야 합니다.

결과를 보면 box2 의 왼쪽에서 떨어지는 기준이 box1 의 오른쪽 입니다.

\*- 참고 -\*

---

float 을 적용하고

요소에 넓이 높이가 적용되어 있지 않다면  
요소의 길이만큼 넓이와 높이가 보입니다.

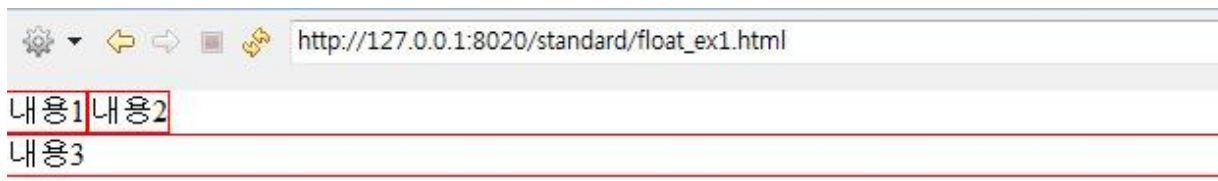
---

float 가 적용된 요소는 띄워진 요소로 인식되기 때문에 아래에 요소가 float  
적용된 요소의 위치로 올라오게 됩니다.

이러한 현상을 방지 하기 위한 속성이 바로 clear 속성입니다.

내용 3 이 위의 두 요소에 적용된 float 정렬에서 풀렸습니다.

```
5 <meta http-equiv="Content-Type" content="text/html; charset=u
6 <title>New Web Project</title>
7 <style type="text/css">
8     *{margin:0;padding:0}
9     div{border:1px solid red;}
10    #box1{float:left;}
11    #box2{float:left;}
12    #box3{clear:both;}
13 </style>
14 </head>
15 <body>
16     <div id="box1">내용1 </div>
17     <div id="box2">내용2 </div>
18     <div id="box3">내용3 </div>
19 </body>
20 </html>
21
22
```

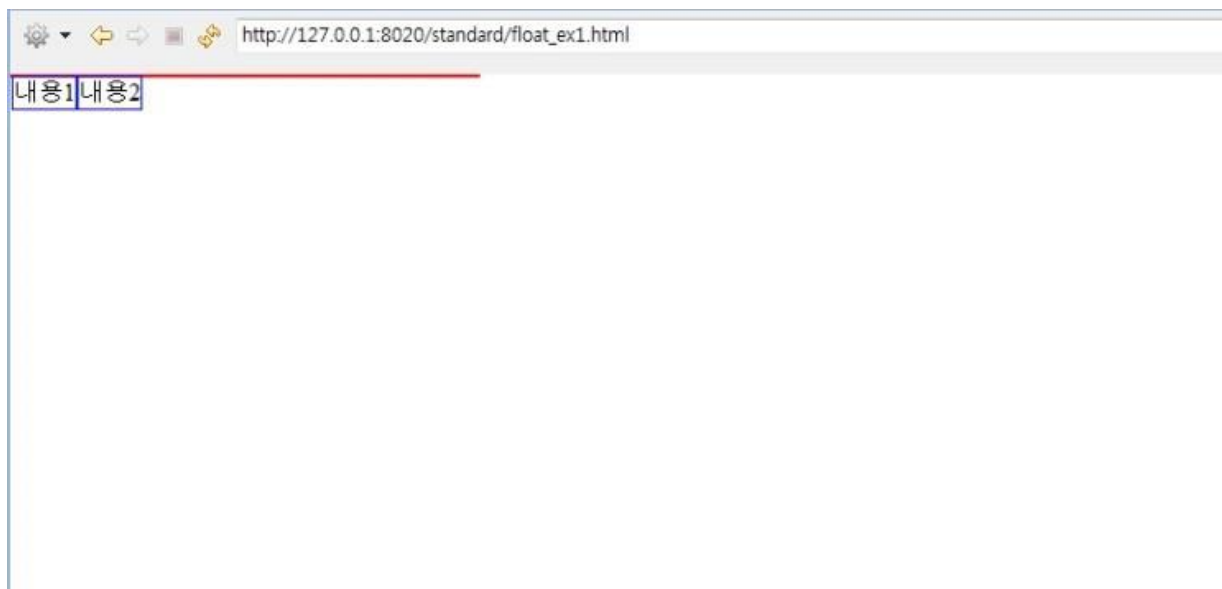


float 은 일반적으로 상위 요소 안에 하위 요소들이 float 되는 경우가 많습니다.  
(엄마 안에 자식을 넣는 경우)

```

1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
2 "http://www.w3.org/TR/html4/loose.dtd">
3 <html xmlns="http://www.w3.org/1999/xhtml">
4   <head>
5     <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
6     <title>New Web Project</title>
7     <style type="text/css">
8       * {margin: 0;padding: 0;}
9       #box {width: 300px;border: 1px solid red;}
10      #box .sub1 {float:left;border: 1px solid blue;}
11      #box .sub2 {float:left;border: 1px solid blue;}
12    </style>
13  </head>
14  <body>
15    <div id="box">
16      <div class="sub1">내용1</div>
17      <div class="sub2">내용2</div>
18    </div>
19
20  </body>
21 </html>

```



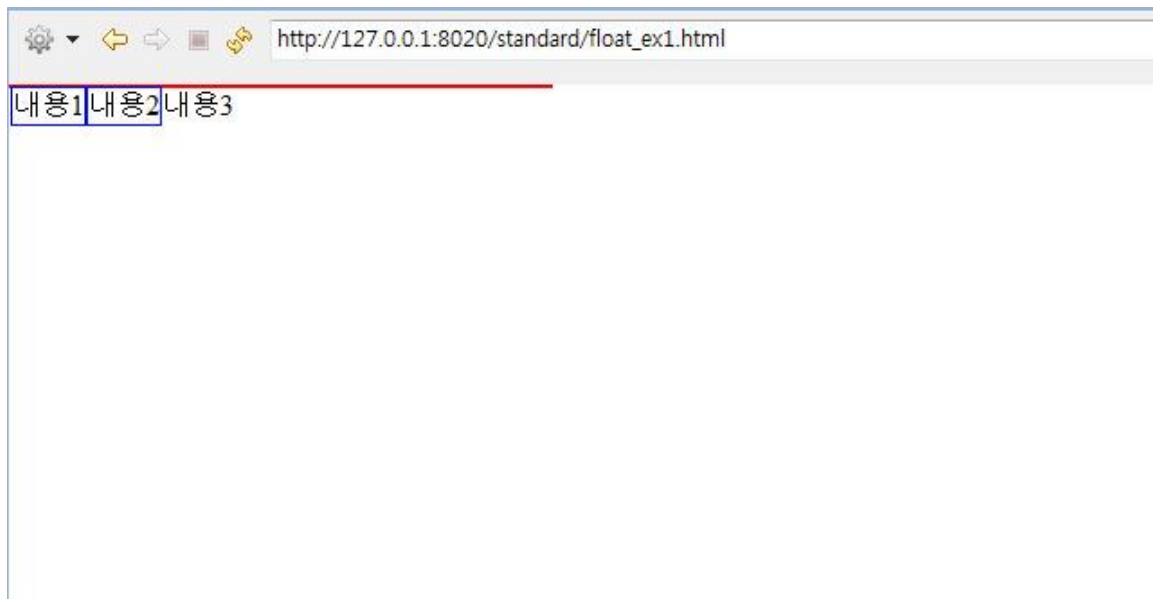
위 예제를 보면

자식 요소 sub1 과 sub2 는 float 에 의해 가로 정렬 되었으나 부모인 box 의 높이 값이 사라지는 것을 볼 수 있습니다.

보통 엄마요소에 높이 값을 지정해 주지 않으면 자식 요소의 높이 값만큼 벌어지는 것이 일반적이거나 float 이 적용되면 **float 적용된 요소는 위로 뜨는 것으로 인식하기 때문에 자식요소들이 엄마 요소 안에 채워져 있는 것으로 인식하지 않기 때문에 엄마 요소의 높이 값이 사라지게 됩니다.**

이렇게 되면 엄마 요소인 box 에 배경색이나 배경 이미지를 넣을 수 없다는 것과 float 적용된 마지막 요소 다음에 오는 요소라 따라 올라 온다는 것입니다. (아래 예제사진에서 확인 )

```
3 <html xmlns="http://www.w3.org/1999/xhtml">
4   <head>
5     <meta http-equiv="Content-Type" content="text/html; charset=u
6     <title>New Web Project</title>
7     <style type="text/css">
8       * {margin: 0;padding: 0;}
9       #box {width: 300px;border: 1px solid red;background:green;}
10      #box .sub1 {float:left;border: 1px solid blue;}
11      #box .sub2 {float:left;border: 1px solid blue;}
12    </style>
13  </head>
14  <body>
15    <div id="box">
16      <div class="sub1">내용1</div>
17      <div class="sub2">내용2</div>
18    </div>
19
20    <div>내용3</div>
21
22  </body>
23 </html>
```

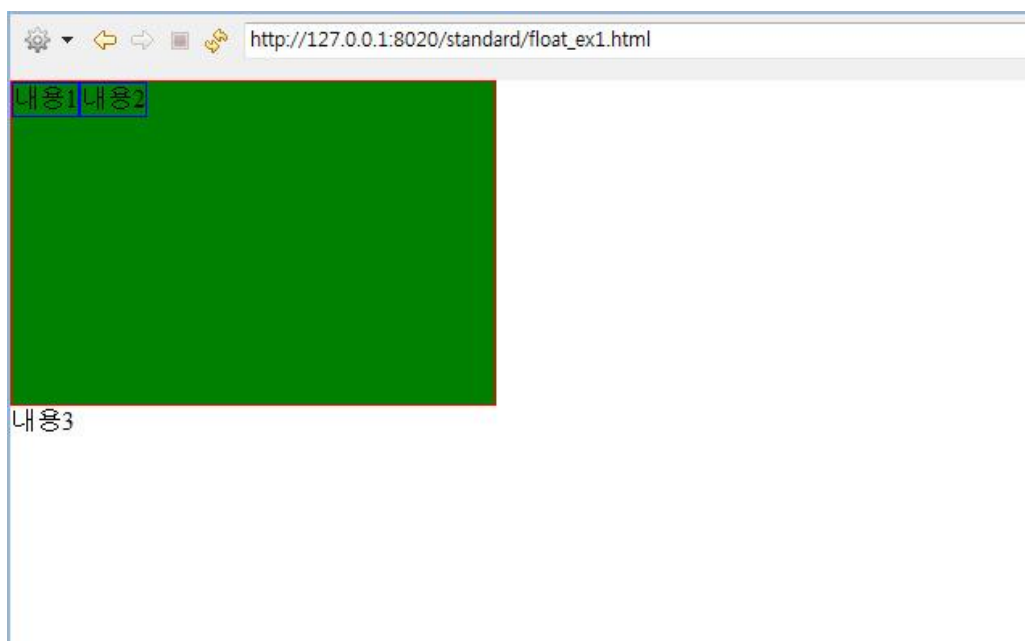


이런 문제들은 엄마 요소에 높이 값이 없어진것이 원인이므로  
엄마 요소에 높이 값을 주면 해결 됩니다. (아래 예제 확인 )

```

3 <html xmlns="http://www.w3.org/1999/xhtml">
4 <head>
5   <meta http-equiv="Content-Type" content="text/html; charset=utf
6   <title>New Web Project</title>
7   <style type="text/css">
8     * {margin: 0;padding: 0;}
9     #box {height:200px;width: 300px;border: 1px solid red;background
10    #box .sub1 {float:left;border: 1px solid blue;}
11    #box .sub2 {float:left;border: 1px solid blue;}
12  </style>
13 </head>
14 <body>
15   <div id="box">
16     <div class="sub1">내용1</div>
17     <div class="sub2">내용2</div>
18   </div>
19
20   <div>내용3</div>
21
22 </body>
23 </html>
24
25

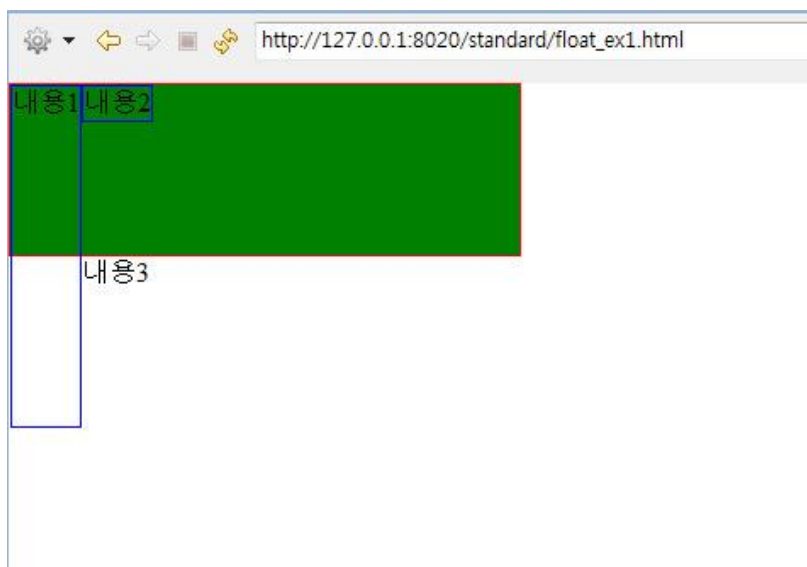
```





그런데 만약 자식 요소의 높이 값이 엄마 요소의 높이 값 보다 큰경우 자식 요소가 엄마 요소의 범위를 벗어날 뿐만 아니라 자식 요소가 엄마 요소의 바로 아래 있는 요소의 영역을 침범 해 버리는 문제가 발생합니다. (아래 예제 확인 )

```
3 <html xmlns="http://www.w3.org/1999/xhtml">
4   <head>
5     <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
6     <title>New Web Project</title>
7     <style type="text/css">
8       * {margin: 0;padding: 0;}
9       #box {height:100px;width: 300px;border: 1px solid red;background-color: #00FF00;}
10      #box .sub1 {float:left;border: 1px solid blue;height:200px;}
11      #box .sub2 {float:left;border: 1px solid blue;}
12    </style>
13  </head>
14  <body>
15    <div id="box">
16      <div class="sub1">내용1</div>
17      <div class="sub2">내용2</div>
18    </div>
19
20    <div>내용3</div>
21
22  </body>
23 </html>
```





이문제를 해결하기 위해서는

### 1. 부모요소에 overflow 속성을 적용합니다.

```
2 <!DOCTYPE html>
3 <html xmlns="http://www.w3.org/1999/xhtml">
4 <head>
5 <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
6 <title>New Web Project</title>
7 <style type="text/css">
8 * {margin: 0;padding: 0;}
9 #box {overflow:hidden;width: 300px;border: 1px solid red;}
10 #box:after{content: "";clear: both;display: block;}
11 #box .sub1 {float:left;border: 1px solid blue;height:200px;}
12 #box .sub2 {float:left;border: 1px solid blue;}
13 </style>
14 </head>
15 <body>
16 <div id="box">
17 <div class="sub1">내용1</div>
18 <div class="sub2">내용2</div>
19 </div>
20
21 <div>내용3</div>
22
23 </body>
24 </html>
```



## 2. :after 속성을 적용합니다.

after 는 선택한 요소 다음에 새로운 요소를 생성하는 속성입니다. :after 로 생성된 요소는 인라인 요소로 인식하기 때문에 블록 요소로 바꿔줘야 clear 속성을 적용할 수 있습니다.

```
2 <?xml:namespace prefix="http://www.w3.org/TR/html4/loose.dtd" ?>
3 <html xmlns="http://www.w3.org/1999/xhtml">
4   <head>
5     <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
6     <title>New Web Project</title>
7     <style type="text/css">
8       * {margin: 0;padding: 0;}
9       #box {width: 300px;border: 1px solid red;background:green;}
10      #box:after{content: "";clear: both;display: block;}
11      #box .sub1 {float:left;border: 1px solid blue;height:200px;}
12      #box .sub2 {float:left;border: 1px solid blue;}
13    </style>
14  </head>
15  <body>
16    <div id="box">
17      <div class="sub1">내용1</div>
18      <div class="sub2">내용2</div>
19    </div>
20
21    <div>내용3</div>
22
23  </body>
24 </html>
```



### 3. 부모요소에 float 속성이 적용 되는 경우에는 부모요소에 높이 값이 적용 됩니다.

대신 이런경우 부모요소가 float 된 상태이고 내용 3 은 그 밖에 있기 때문에 위치가 위의 1.2 번의 경우와는 다르게 나타납니다.

```

2  "http://www.w3.org/TR/html4/loose.dtd">
3  <html xmlns="http://www.w3.org/1999/xhtml">
4    <head>
5      <meta http-equiv="Content-Type" content="text/html; charset=
6      <title>New Web Project</title>
7      <style type="text/css">
8        * {margin: 0;padding: 0;}
9        #box {float:left;width: 300px;border: 1px solid red;background
10       #box:after{content: "";clear: both;display: block;}
11       #box .sub1 {float:left;border: 1px solid blue;height:200px;}
12       #box .sub2 {float:left;border: 1px solid blue;}
13     </style>
14   </head>
15   <body>
16     <div id="box">
17       <div class="sub1">내용1</div>
18       <div class="sub2">내용2</div>
19     </div>
20
21     <div>내용3</div>
22
23   </body>

```

