

# float 추가내용

≡ 구분 Web

## CSS 원칙 1



## Normal Flow

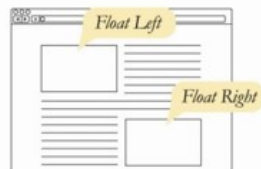
## CSS 원칙 1

모든 요소는 네모(박스모델)이고,  
위에서부터 아래로, 왼쪽에서 오른쪽으로 쌓인다.  
(좌측 상단에 배치)

# 어떤 요소를 감싸는 형태로 배치는? 혹은 좌/우측에 배치는?

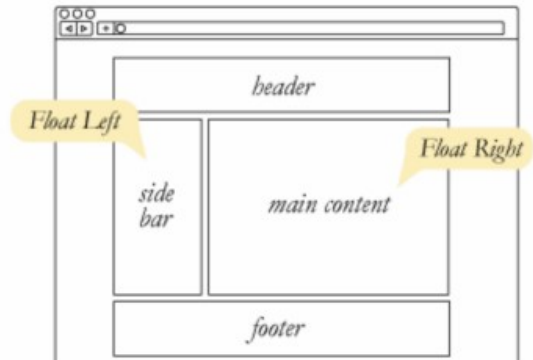
### Float

- 박스를 왼쪽 혹은 오른쪽으로 이동시켜 텍스트를 포함 인라인요소들이 주변을 wrapping 하도록 함
- 요소가 Normal flow를 벗어나도록 함



## Float 속성

- none : 기본값
- left : 요소를 왼쪽으로 띄움
- right : 요소를 오른쪽으로 띄움



float 를 실습하기 위해서 셋팅부터 해보자.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>

<style>
  .box{
    width: 150px;
    height: 150px;
    border: 1px solid black;
    background-color: crimson;
    color: white; font-size: 50px;
    margin: 50px;
  }

  .c{
    float: right;
  }
</style>

<body>

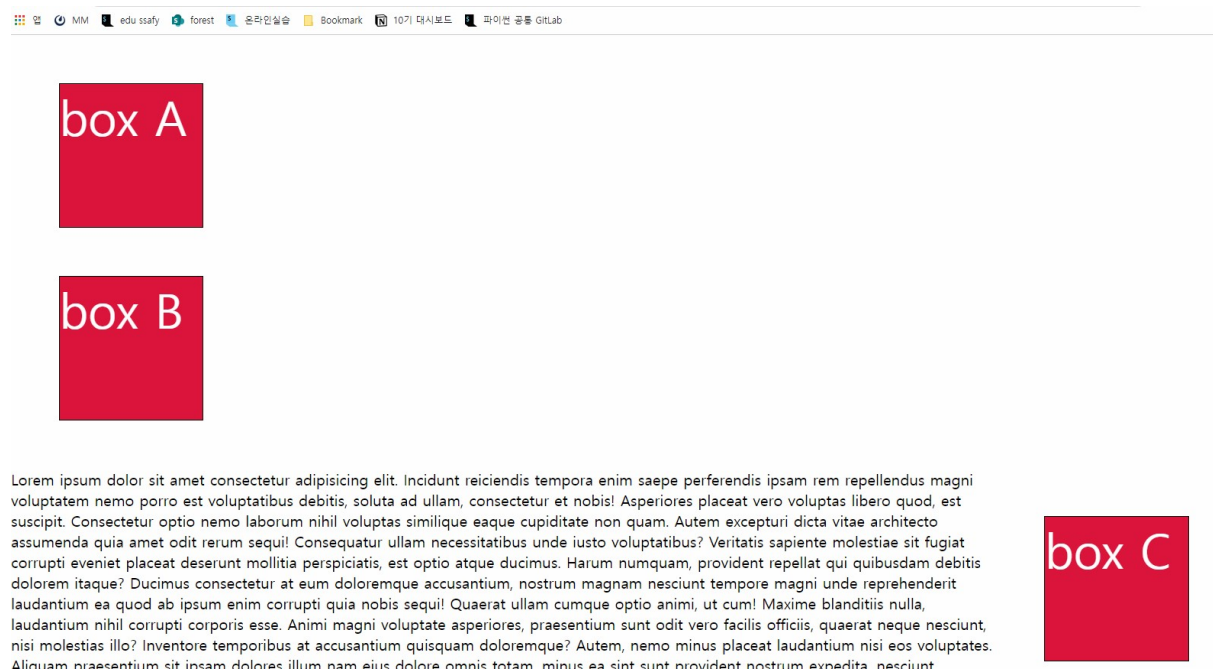
  <!-- div.boxa*3+p -->
  <div class="box a">box A</div>
  <div class="box b">box B</div>
  <div class="box c">box C</div>

  <p>Lorem*300</p>

</body>
</html>
```

코드 작성 후 확인해 보면 box C 밑에 lorem 단어들이 있는것이 아니라  
box C가 lorem 단어들 옆으로 붓 뗐다.

CSS 원칙인 노말 플로우에서 벗어난 것이다.



Float를 사용 할때 나타나는 대표적인 문제점 또는 에로사항에 대해서 먼저 살펴보자.

세가지를 명시해 보았다.

1. normal flow를 따르지 않으며 경우에 따라 요소끼리 겹쳐지는 현상이 발생하거나 (보통흐름-위에서 아래로 좌에서 우로 박스가 쌓인다)
2. float 한 모든 요소들은 inline 요소의 display 속성이 block요소로 바뀌고 렌더링은 inline-block 처럼 렌더링이 된다는 특이한 점이 있다.
3. 부모요소의 공간이 무시되는 현상이 발생하기도 함

이러한 단점? 문제들 때문에 css3부터는 웹페이지 레이아웃 잡을때 float 사용해서 잡는것을 권장하지는 않는다.

글로 보지 말고 위의 예를 직접 하나씩 확인만 해보자  
먼저 float 사용시 요소 겹침 현상부터 보자

1. normal flow 를 따르지 않으며 경우에 따라 요소끼리 겹치는 현상이 일어난다고 했다.

박스 C에 CSS 적용시킨 것을 지우자 그리고 아래 코드를 추가해 보자.

```
/* .c css먹인것을 지우고  
   박스 b를 left에 float 해보자 */  
  
.b{  
    float: left;  
}
```

박스 B와 박스 C가 겹침 현상이 일어난다. 이는 float속성이 있는 박스B가 float속성이 없는 박스C를 인식할 못하는 문제가 발생하면서 요소가 겹치는 문제가 발생하는 것이다.

간단하게 b 밑에 있는 박스 c에 float right를 해줘서 겹친것 없이 p 태그의 왼쪽 오른쪽에 floatong 되도록 해 보자.

```
.b{
  float: left;
}
.c{
  float: right;
}
```

Float 사용하다가 나오는 두번째 문제점.

2. float 한 모든 요소들은 inline 요소의 display 속성이 block요소로 바뀌고 렌더링은 inline-block 처럼 렌더링이 된다는 특이한 점이 있다.

```
<body>

<!--
  <div class="box a">box A</div>
  <div class="box b">box B</div>
  <div class="box c">box C</div> -->

  <span style="float:right">심우석 학생은 이서진 님이었다.</span>

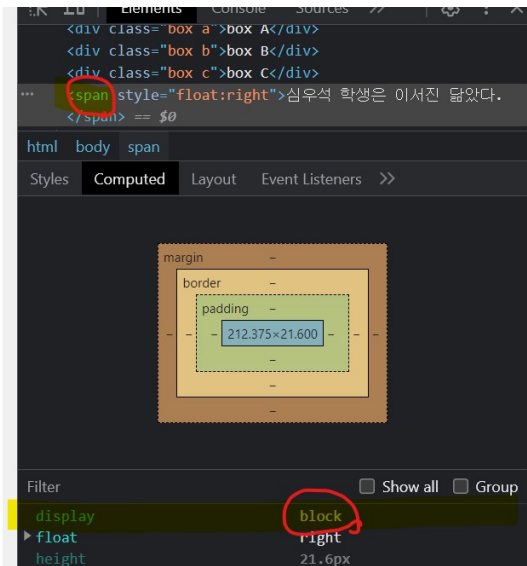
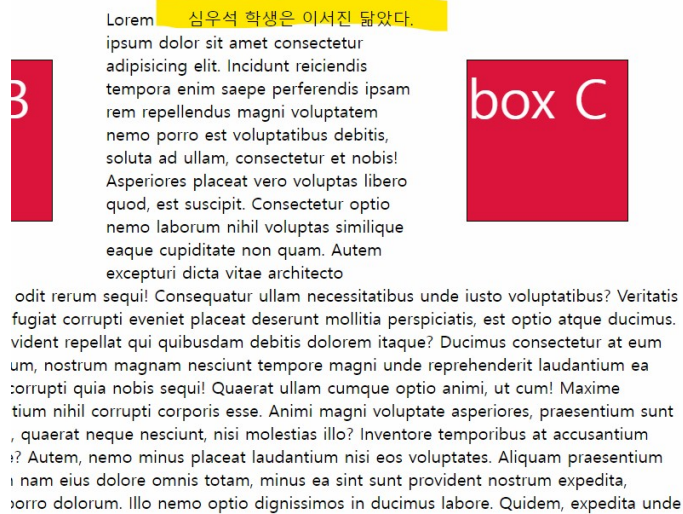
<!--
  <p>Lorem ipsum dolor sit amet consec... 생략 -->
```

기존에 있는 코드에서

body > box c 밑에 inline 요소인 span 태그하나를 추가했다.

그리고 저장 후 라이브서버로 확인해 보자.

개발자도구 (F12)를 켜서 span 태그를 확인해 보자.

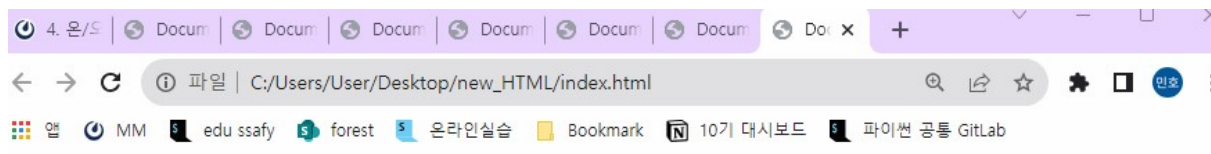


개발자 도구에 들어가서 [계산됨] 확인하면 display가 block으로 바뀐것을 확인할 수 있다. (참고로 span태그는 inline 속성이다!!)

그리고 또 신기한 것은 display가 block이지만 렌더링 할때는 inline block으로 되는 이상한 특징이 있다.

3. 세번째로 float를 사용하다 보면 부모요소의 공간이 무시되는 현상이 발생하기도 한다고 했는데 이게 무슨말인지 코드로 살펴보자.

여기서 두 박스를 나란히 놓아볼 것이다. 아래 그림과 같이.



## section 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aut officiis inventore sunt eligendi dolore veniam expedita tenetur? Dolores reprehenderit molestiae, totam labore, id vitae quo suscipit ullam delectus neque corporis.

## section 2

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aut officiis inventore sunt eligendi dolore veniam expedita tenetur? Dolores reprehenderit molestiae, totam labore, id vitae quo suscipit ullam delectus neque corporis.

```
<style>

  .box{
    display: inline-block;
    float: left;
  }
  .a {
    background-color: yellow;
    width: 30%;
  }
  .b {
    background-color: yellowgreen;
    width: 50%;
  }
  .box{
    display: inline-block;
```

```

    }

</style>
<body>
  <div class="container">

    <section class="box a">
      <h1>section 1</h1>
      <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Aut officiis inventore sunt eligendi dolore
        veniam expedita tenetur? Dolores reprehenderit molestiae, totam labore, id vitae quo suscipit ullam
        delectus neque corporis.</p>
    </section>
    <section class="box b">
      <h1>section 2</h1>
      <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Aut officiis inventore sunt eligendi dolore
        veniam expedita tenetur? Dolores reprehenderit molestiae, totam labore, id vitae quo suscipit ullam
        delectus neque corporis.</p>
    </section>

  </div>
</body>

```

여기까지 셋팅이 되었다면 이제 footer도 하나 만들어 보겠다.

body 맨 밑에 footer도 추가해 보자.

<footer>풏터</footer>

그리고 렌더링이 어떻게 되었는지 확인해 보라. .box에 float 먹인것이 있어서 footer가 가장 밑으로 내려가 있지 않았을 것이다. 이때

```

footer{
  clear: both;
}

```

를 추가해서 float의 영향을 벗어버리자. 풏터가 가장 밑으로 내려갔을 것이다.

그 다음에

컨테이너에 배경색을 넣어 볼 것이다.

```

<style>
  .container{
    background-color: aqua;
  }
  .box{
    display: inline-block;
    float: left;
  }
  .a {
    background-color: yellow;
    width: 30%;
  }
  .b {
    background-color: yellowgreen;
    width: 50%;
  }
  .box{
    display: inline-block;
  }
  footer{
    clear: both;
  }

</style>

```



분명히 box들의 부모요소인 container의 배경을 아쿠아로 했는데 확인해 보면 적용이 안되어 있다.  
이것이 아까 말한 부모요소의 공간이 무시되는 현상이 발생한 것이다.

## 해결책

section의 마지막 태그 밑에 아래 코드를 추가

```
<div style="clear:both"></div>
```

확인해 보면 컨테이너의 배경색으로 아쿠아가 적용될 것이다.

결론은 한 태그의 자식들을(여기서는 box들) float를 통해서 배치를 했다면  
부모요소가 (container) 끝나는 시점에서 clear를 해줘야 부모요소의 공간이 무시되는  
현상을 막을수가 있다.

마지막으로 <footer>풋터 주소 사업자등록번호 전화번호 대표자명</footer> 넣어보는 것으로  
마무리 짓자.