

float속성

박스를 부모의 왼쪽이나 오른쪽에 고정시키고 문장을 그 옆으로 흐르도록 하기 위한 속성.

#01. 적용가능한 값

- left(왼쪽), right(오른쪽), none(해제), inherit(부모값 상속)

#02. 제약사항

- 새로운 문단을 시작하기 위해서는 이전 float를 off시켜야 한다.
- 부모요소가 float가 적용된 자식요소를 감싸지 못한다.

해결책

이전 float 속성을 해제.

- 부모요소가 끝나기 전(혹은 새로운 문단이 시작하기 전) `float: none; clear: both`가 적용된 Block-Level요소를 배치.

```
float: none;
clear: both;
```

:after 가상 클래스 사용

부모의 :after 가상 클래스를 사용해서 아래의 코드 적용

```
content: '';
display: block;
float: none;
clear: both
```

#03. 특성

두 개 이상의 Block-Level요소에게 float를 연속적으로 적용하면 박스의 배치 효과

1) 고전적인 레이아웃 구성 방법

사이트의 전체 넓이와 정렬을 구성하기 위한 `<div>`태그 안에 각 영역을 구성하기 위한 `<div>`태그를 배치하고, float 속성을 사용하여 각 영역을 배치하여 화면 레이아웃을 구성할 수 있다.

3단 레이아웃

```
<div id="container">
  <div id="header">상단영역</div>
  <div id="content">내용영역</div>
  <div id="footer">하단영역</div>
</div>
```

- #container에 넓이와 margin속성을 사용하여 정렬을 구성
- #header, #footer에 각 영역에 대한 height 지정 (고정높이)
- #content 영역은 내용에 따라 높이가 변경되어야 하므로 min-height 지정(가변높이)

사이드바 구성

```
<div id="container">
  <div id="header">상단영역</div>
  <div id="content">
    <div id="sidebar">사이드바</div>
    <div id="body">내용영역</div>
  </div>
  <div id="footer">하단영역</div>
</div>
```

- 3단 레이아웃 상태에서 #content에 사이드바와 본문영역을 위한 <div>를 추가
- #sidebar와 #body를 float 속성을 사용하여 가로 배치

최근에는 이 방식외에도 Flex나 Grid 기반 레이아웃 구성도 많이 활용되고 있습니다.