

11월 24일까지 CSS도 HTML 가로배치하기 (float, overflow, clear, inline-block) ~  
CSS 포지셔닝 속성 완벽하게 이해하기 (3편) - 부모도 밖으로 위치시키기.

### [CSS로 HTML 가로 배치하기(float, overflow, clear, inline-block)]

CSS 포지셔닝 - float. -> 요소가 부모요소 기준으로 왼쪽 또는 오른쪽에 배치할지 지정.

(none, left, right, margin: auto -> 중앙. 인라인 요소는 적용되지 않음. 블록 요소만 적용.)

HTML 요소가 왼쪽에 있다고 해서 float:left 가 기본값이 아님!! 기본값은 float: none

<div class="parent"> <div class="child"> </div></div> 의 경우

```
.parent{
  border: 5px solid red;
  width: 600px;
  /* height: 200px; */
  overflow: hidden;
}
.child {
  background-color: gold;
  width: 200px;
  height: 200px;
  float: left; /*-> 원래 부모 요소는 자식 요소의 값에 따라 높이가 정해지지만, 자식 요소에 float 속성이 사용
되면 부모요소는 자식요소가 없다고 판단해 높이값을 잃음.
이 경우 1. 부모요소에 height 값을 직접 지정하는 방법, 2. overflow: hidden; 을 하는 방법이 있음.
하지만 1번을 할 경우 자식요소의 높이가 늘어나거나 줄어들면 그때그때 다시 지정해줘야 함. -> 번거로움.
2번의 경우 알아서 자식요소의 높이값에 따라 변화함. */
}
```

<div class="parent"> <span class="child"> </span></div> 의 경우

```
.parent{
  border: 5px solid red;
  width: 600px;
  /* height: 200px; */
  overflow: hidden;
  text-align: center; /*inline, inline-block 요소는 이 방법으로 배치해야 함. */
}
.child {
  background-color: gold;
  width: 200px;
  height: 200px;
  display: block; /*display: inline-block으로 하면 inline 성질 때문에 margin: auto로 원하는대로 할 수 없음
*/
  margin: auto; /*inline 요소, inline-block 요소는 이 방법으로 중앙으로 배치할 수 없음. */
}
```

###inline, inline-block 요소를 중앙으로 보내려면 -> 부모요소의 text-align: center;

block 요소를 중앙으로 보내려면 -> 본인의 margin: auto;

clear 속성 -> float 속성이 적용되면 다음 요소가 float 속성을 상속받는데 이것을 해제시킴. (left, right, both\*\*)

both를 자주 사용.

section, article 요소 -> div랑 큰 차이 없음. section 안에 article

가로배치하는 두 가지 방법.

1. float 와 overflow 를 사용.

```
<section>
```

```
  <article>article</article>
```

```
  <article>article</article>
```

```
  <article>article</article>
```

```
</section>
```

```
section {
```

```
  border: 5px solid black;
```

```
  width: 600px;
```

```
  overflow: hidden;
```

```
}
```

```
section article{
```

```
  background-color: gold;
```

```
  width: 200px;
```

```
  height: 100px;
```

```
  float: left;
```

```
  border: 1px solid red;
```

```
  box-sizing: border-box; /*border, padding 값을 width에 포함 시키게. */
```

```
}
```

2. display: inline-block으로 배치.

```
section{
```

```
  border: 5px solid black;
```

```
  text-align: center;
```

```
}
```

```
section article {
```

```
  background-color: gold;
```

```
  width: 200px;
```

```
  height: 100px;
```

```
  border: 1px solid red;
```

```
  display: inline-block; /*inline은 width, height 값을 갖지 못함. */
```

```
}
```

1px도 틀리지 않아야 한다 -> 1번째 방법 이용. 여유있다 -> 2번째 방법 이용.

**[HTML5 시멘틱 태그로 CSS 레이아웃 설계하기(부모요소, 자식요소, 형제요소)]**

HTML4까지는 레이아웃 -> div로 클래스 이름을 줘서 만들었음.

가장 상위의 컨테이너 -> [.container](#) 또는 wrapper

문서의 주요 내용 지정 -> main (꼭 필요한 것은 아님)

주제별 콘텐츠 영역 -> [section](#)

헤더 영역 (로그, 메뉴, 로그, 검색 등) -> [header](#)

제작 정보 및 저작권 정보 표시 -> [footer](#)

콘텐츠 내용 넣기 -> [article](#)

문서를 링크하는 탐색 영역 -> nav header 안에 nav 안에 article, div 이런식!!

세부 사항 요소 -> summary / 추가 세부 정보를 정의: details. (잘 사용하지 않음)

계층 구조 1. [.container](#) 2. [section](#), [header](#), [footer](#) 3. [article](#) 4. [div](#)(상세하게!!)

```

<div class="container">
  <header>
    <article class="logo"></article>
    <article class="navi"></article>
  </header>
  <section class="box1"></section> /*의미없게 box1으로 하지말고 대표적인 단어 쓸것. */
  <section class="box2">
    <article class="sub1"></article>
    <article class="sub2"></article>
    <article class="sub3"></article>
  </section>
  <footer></footer>
</div>

```

## CSS

```

.container{
  border: 1px solid red;
  width: 1200px;
  margin: auto;
}
header{
  background-color: skyblue;
  height: 100px; /*overflow: hidden; 해도 되긴 함. */
}
header article{ /* header article 은 logo와 navi 의 공통 특성. */
  background-color: gray;
  width: 300px;
  height: 100px;
}
.logo{
  float:left; /*logo와 navi 중 하나만 float 값이 있고 하나는 없으면 둘이 높이상 위치가 다름. */
}
.navi{
  float:right;
}
.box1 { /*width는 100%, 자식요소가 없으니까 height 값을 줌. */
  background-color: yellowgreen;
  height: 300px;
}
.box2 { /*자식 요소가 있으므로 height 값을 주지 않아도 됨. */
  background-color: greenyellow;
  overflow: hidden;
}
.box2 article{ /*공통적인 요소들. */
  width: 33.33333333%; /*보통 개별로 넣게 되어있음. */
  height: 300px;
  float: left;
}
.sub1 -> background-color: darkgray;
.sub2 -> background-color: lightgray;
.sub3 -> background-color: lightslategray;
footer -> background-color: skyblue; height:100px;

```

## [CSS 포지션 속성 완벽하게 이해하기(1편) - 부모요소 vs 자식요소]

position 속성: relative & absolute

Relative -> 부모요소!!! (상대적인) 감싸고 있는 요소

Absolute -> 자식요소!!! (절대적인) 감싸져 있는, 감쌌을 당하는 요소

```
<div class="parent">
  <div class="child"></div>
</div>
```

CSS

```
.parent{
  background-color: dodgerblue;
  width: 600px;
  height: 300px;
  position: relative; /*이걸 하지 않으면 child는 부모요소를 parent가 아닌, 브라우저 자체라고 생각. */
}
.child {
  background-color: crimson;
  width: 200px;
  height: 100px;
  position: absolute;
  /* left, right, top, bottom이라는 속성은 반드시 position 속성과 함께 써야함. */
  right: 0;
  bottom: 0;
}
```

position으로 부모-자식 관계를 맺어주는 것!!

## [CSS 포지션 속성 완벽하게 이해하기(2편) - 자동으로 수평 수직 중앙 위치시키기]

원하는 위치에 요소 위치 시키기.

좌표 관련 속성은 꼭! position 속성과 함께 이용해야함.

부모요소에는 position: relative; 자식요소에는 position: absolute;

top:0, right:0을 하면 오른쪽위 코너에 요소 위치.

브라우저 크기에 관계없이 항상 수직 수평 중앙 정렬하기

```
position: absolute;
top: 50%; /*퍼센트 단위!! */
left: 50%;
```

transform: translate(-50%, -50%) /\*그냥 top, left 만 하면 요소의 좌측상단이 정중앙에 위치. 이를 본인 크기 절반만큼 이동!! 해야 요소가 정확히 중앙에 위치하는것! 반대로 가야하므로 -50%!! \*/

HTML은 좌측상단이 좌표의 기준!!

## [CSS 포지션 속성 완벽하게 이해하기(3편) - 부모요소 밖으로 위치시키기]

원하는 위치로 정확히 보내기.

+값 -> 위에서 아래로, 왼쪽에서 오른쪽으로

-값 -> 아래에서 위로, 오른쪽에서 왼쪽으로

%를 써서 표현해야 부모요소의 값이 바뀌어도 알아서 바뀜. 직접 값을 입력해서 하는 경우 -> 매번 부모요소의 값이 바뀔때마다 자식요소의 값도 바뀌어야함!