

Bootcamp Full Stack

CSS avanzado

Encolumnado: float

Encolumnado

float

Esta propiedad fue la primera que permitió **encolumnar elementos**. Pero, **por su complejidad, rápidamente fue reemplazada por grid y flex**.

Sin embargo, cualquier desarrollador debe conocerla porque muchos elementos la incluyen. Por esta razón, se hace necesario saber cómo se utiliza correctamente.

Los elementos de bloque se ubican de manera predeterminada uno debajo del otro, como vemos en el siguiente slide.



A la derecha, observamos elementos de bloque ubicados naturalmente uno debajo del otro.

Para cambiar esa disposición podemos indicar que esos elementos estén ***flotados***:

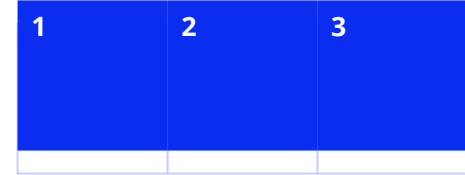
```
div { width: 100px;  
      height: 100px;  
      background-color: blue;  
      float:left; }
```



El valor de **float** puede indicar que flote hacia la izquierda: **left** o hacia la derecha: **right**.

```
div { width: 100px;  
      height: 100px;  
      background-color: blue;  
      float:right; }
```

Resultado



Es posible **combinar elementos flotados**, como se ve en la siguiente estructura:

```
<div id="uno">1</div>  
<div id="dos">2</div>  
<div id="tres">3</div>
```

Generamos la siguiente regla de estilo:

```
div { width: 100px;  
      height: 100px;  
      background-color: blue;  
      float:right; }  
#uno { float: left;}
```

clear

También podemos indicar que algunos elementos **no acepten flotado** a través de la propiedad **clear**.

Clear es una propiedad complementaria de **float**, y posee valores similares:

- left
- right
- both
- none

```
#tres { clear: right;}
```

Resultado



Un uso frecuente de **clear** con el **valor: both** es que **sirve para generar *footers*** o elementos que no deseamos que sean afectados por el float de elementos anteriores:

```
div { width: 100px;
      height: 100px;
      background-color: blue;}
#uno { float: left;}
#dos { float: right;}
#tres { clear: both;}
```

Resultado



¿Sabías que?

Una complejidad que tiene **float es que si flotamos elementos** dentro de un contenedor sin altura explicitada, **éste desaparece**:

```
✓ div { width: 100px;
        height: 100px;
        background-color: blue; float:left}
main { width: 100%; background:red;}
```

Resultado



No podemos observar el **fondo rojo del main**.

Para resolverlo tenemos dos opciones:

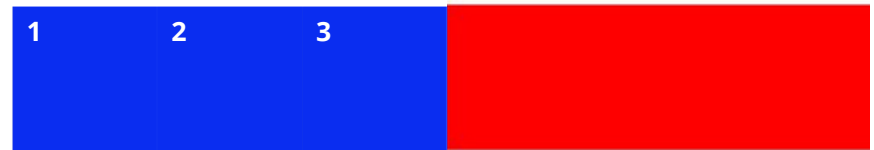
1. Ubicar un elemento con **clear:both al final.**

```
div { width: 100px;
      height: 100px;
      background-color: blue; float:left}
main { width: 100%; background: red;}
footer { clear:both;}
```

Se cambiará la estructura:

```
<main>
  <div id="uno">1</div>
  <div id="dos">2</div>
  <div id="tres">3</div>
  <footer></footer>
</main>
```

Resultado



2.Podemos utilizar la propiedad **overflow** que indica cómo debe comportarse el contenido **excedente** con el valor **hidden**.

Es decir el valor excedente se ocultará:

```
<main>
  <div id="uno">1</div>
  <div id="dos">2</div>
  <div id="tres">3</div>
</main>
```

```
div { width: 100px;
      height: 100px;
      background-color: blue; float:left}
main { width: 100%; background: red; overflow: hidden;}
```

Si bien esta propiedad en la actualidad puede no estar contemplada en muchos maquetados, dado el gran poder de trabajo y simplificación que trajo el encolumnado con **flex**, sigue siendo útil conocerla, dado que en ciertas interfaces podemos encontrarla y debemos conocer cómo modificarla, así como en frameworks como *Bootstrap* aún sigue vigente en algunos ejemplos, por caso cuando queremos colocar una imagen hacia la izquierda o derecha de un texto.



Revisión

- Repasar los conceptos vistos de **float**, **clear** y **overflow**.
- Implementar estas nuevas propiedades.
- Ver todos los videos y materiales necesarios antes de continuar.

**¡Sigamos
trabajando!**

