



## Tabla de contenido

<b>Usando tablas .....</b>	<b>1</b>
<b>Usando float.....</b>	<b>1</b>
Desventajas .....	2
Ventajas.....	3
<b>Usando display.....</b>	<b>3</b>
Desventajas .....	4
Ventajas.....	4
<b>Usando table .....</b>	<b>4</b>

Para crear columnas en nuestras páginas web tenemos diferentes modos que vamos a estudiar a continuación.

## Usando tablas

Es el método más antiguo y que ya no deberíamos usar hoy día, principalmente por dos motivos:

- es muy verboso, es decir, hay que escribir demasiadas etiquetas para ellos (table, un tr por cada fila y un td o th por cada celda dentro de una fila).
- no es adecuado para diseños responsivos, pues no podemos cambiar con CSS el número de filas, columnas, cambiar una columna por una fila, ...

Por lo tanto no veremos ningún ejemplo, además de los que vimos al principio del curso.

## Usando float

Mediante la propiedad CSS float podemos flotar un elemento a la izquierda o derecha, pudiendo crear así columnas, siempre que tengan establecido un ancho, pues si no ocupará todo el ancho.

Ejemplo:

```
.caja1 {
  background-color: rgb(43, 122, 226);
  width: 50%;
  height: 200px;
  float: left;
}
.caja2 {
  background-color: rgb(35, 220, 158);
  width: 50%;
  height: 100px;
  float: left;
}
</style>
</head>
<body>
  <div class="caja1">Lorem ipsum, dolor sit amet consectetur adipisicing elit. Voluptatum, eaque.</div>
  <div class="caja2">Lorem ipsum dolor sit amet.</div>
```



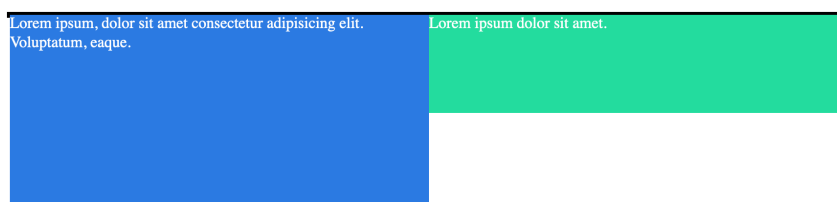
Sin darle ancho a la primera caja, ocuparía todo el ancho, con lo que la segunda columna se pondría debajo por mucha propiedad float que usemos.

Tened **MUCHO CUIDADO** si usamos borde o padding en una de las columnas, pues por defecto suman al ancho de un elemento, con lo que en el caso anterior nos pasaríamos del 50% y la segunda columna no cabría. Recordemos que podemos usar la propiedad **box-sizing** con el valor border-box para que no se sumen.

## Desventajas

Cuando un elemento se pone a flotar, deja de considerarse dentro de su contenedor, con lo que, si ponemos borde o color de fondo a este, no tendrá en cuenta esas columnas.

Por ejemplo, si en el código anterior ponemos los dos div dentro de otro div, al que asignamos un borde y color de fondo, este será el resultado:



No hay color de fondo y el borde solo es una línea, pues se considera que no hay nada en su interior.

La solución es crear una clase, que se suele llamar clearfix, y aplicarla al contenedor de las dos columnas. Existen diferentes variantes de esta clase, pero hoy día esta nos vale:

```
.clearfix::after {  
  content: "";  
  clear: both;  
  display: table;  
}
```

Otro problema que podemos encontrarnos es si queremos que las dos columnas tengan la misma altura; en nuestro ejemplo la columna 2 es más corta. Una posibilidad es darle al contenedor el mismo color de fondo que la columna más corta.

Otra opción es establecer position: relative en el contenedor y luego en la columna, en nuestro ejemplo la segunda, añadir estas propiedades:

```
position: absolute;  
top: 0;
```

```
right: 0;  
height: 100%;
```

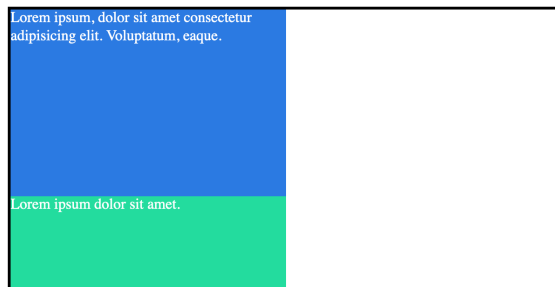
## Ventajas

Este modo es fácil de usar, clearfix aparte, y podemos poner un elemento a la izquierda y otro a la derecha fácilmente usando float: left y float: right respectivamente.

## Usando display

Con la propiedad display al valor inline-block podemos convertir un elemento de tipo bloque a en línea, pero conservando sus propiedades como anchos, altos, márgenes y espaciados superiores e inferiores. Al igual que con float, las columnas deben tener un ancho que permita que quepan en la misma línea, sino irán en filas diferentes.

Ejemplo:



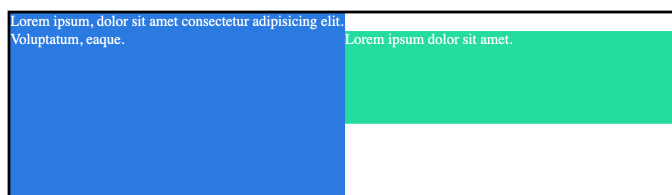
¿Por qué no ha funcionado? Ambas columnas tienen un 50% de ancho y no tienen bordes ni espaciados establecidos. El problema es el salto de línea entre el div de cierre de la primera columna y el de apertura en la segunda, que en los elementos en línea se tiene en cuenta aunque no se vea en pantalla, pues se considera un espacio entre palabras.

Tenemos dos soluciones:

- la primera, simple pero poco recomendable es poner ambas etiquetas en la misma línea y sin espacios

```
</div><div class="caja2">Lorem ipsum dolor sit amet.</div>
```

- no usar un 50% en ambas columnas, sino algo menos, aunque quedará un espacio entre columnas que puede que no queramos.
- usar un pequeño margen izquierdo negativo en la segunda columna; en nuestro ejemplo valdría margin-left: -4px; aunque pueden quedar un pequeño espacio vacío por la derecha
- usar font-size:0 en el contenedor, debiendo tener que poner de nuevo la deseada en las columnas





## Desventajas

- uno es el problema de ese salto de línea que hemos comentado y que debemos solucionar
- podemos ver en la imagen anterior que tenemos el mismo problema que con float si queremos que ambas columnas tengan la misma altura. Podemos resolverlo de modo similar que vimos con float:
  - o usar position: relative en el contenedor
  - o usar height: 100%; position: absolute; y bottom:0; en la columna
- además, según el ancho del contenedor, la columna más corta puede no quedar siempre en la parte superior. Esto lo solucionamos fácilmente usando la propiedad vertical-align: top en esa columna

## Ventajas

- no tenemos que usar la clase clearfix como cuando usamos float
- podemos usar vertical-align con las propiedades bottom (abajo), middle (medio), ... para colocar las columnas verticalmente

## Usando table

CSS permite crear tablas mediante estilos usando los siguientes valores en la propiedad display:

- table: para indicar que un elemento es una tabla
- table-row: para indicar que un elemento es una fila
- table-column: para indicar que un elemento es una columna
- table-row: para indicar que un elemento es una fila
- table-cell: para indicar que un elemento es una celda
- ...

Ejemplo:

```
.caja1 {
  background-color: rgb(43, 122, 226);
  width: 50%;
  height: 200px;
  display: table-cell;
}
.caja2 {
  background-color: rgb(35, 220, 158);
  width: 50%;
  height: 100px;
  display: table-cell;
}
</style>
</head>
<body>
  <div class="caja1">
    Lorem ipsum, dolor sit amet consectetur adipisicing elit. Voluptatum,
    eaque.
  </div>
  <div class="caja2">Lorem ipsum dolor sit amet.</div>
```

Lorem ipsum, dolor sit amet consectetur  
adipiscing elit. Voluptatum, eaque.

Lorem ipsum dolor sit amet.

#### Notas:

- Podemos ver que no hemos usado contenedor, aunque podríamos hacerlo e indicarle display: table
- Las dos columnas tienen la misma altura, cogiéndose la mayor de ellas
- Podemos usar vertical-align para alinear el contenido de la celda

#### Ventajas:

- es muy fácil de usar
- no tenemos que indicar anchos si no queremos y se asignarán según el contenido

¿Y por qué no usarla siempre? La primera es que veremos métodos más modernos como Flexbox y Grid posteriormente. La segunda es más semántica, pues se supone que son propiedades para diseñar tablas.

## Consejos

- Comprobad siempre los width de las columnas si una se coloca debajo de la otra
- A este width hay que sumarle el borde y el padding, con lo que es más fácil si usáis la propiedad box-sizing: border-box en las columnas, con lo que ya no se sumarán
- Si queremos centrar las columnas en pantalla, meterlas dentro de un contenedor con un width y usar margin: auto (o margin-left: auto; y margin-right: auto)
- También puede ser útil crear alguna clase css reset para quitar bordes, márgenes, espaciados, ... por defecto. Por ejemplo:

```
* {  
    margin: 0;  
    padding: 0;  
    font-size: 100%; /* todo con el mismo tipo de letra */  
    border: 0;  
}
```