

Uso de float

La propiedad **float** sirve para alinear cajas o <divs> a la izquierda o a la derecha de otras cajas, utilizando el valor **float:left** o **float:right**.

Al usar **float** hay que tener claro una serie de cuestiones:

1. Para alinear varias cajas (o <divs>) una a la derecha de la otra, **todas las cajas** tienen que tener un **width** determinado y un **float**.
2. La suma de todos los width de todas las cajas deben caber en el ancho del <body>. Si no caben se pondrá una debajo de la otra.
3. Por lo general, en todas las cajas se utiliza el mismo tipo de alineación (o todas left o todas right).
4. Aunque se quiera colocar una caja a la derecha de otra, se utiliza **float:left**, ya que el valor **right** indica que las cajas se alinearán **desde la** parte derecha de la ventana del navegador.
5. La siguiente caja que venga después de la última caja que utiliza un **float**, (aunque no tenga ningún valor float), se alineará en el espacio vacío que deje la alineación anterior. Para evitar esto, a esta nueva caja se le debe añadir la declaración: **clear:both**. De esta manera no aprovechará el espacio vacío que pudiera quedar y se colocará correctamente debajo de la última caja.

Ejemplo práctico del uso de float

Tenemos dos cajas (o <divs>), etiquetadas con los id="caja1" e id="caja2". La primera (#caja1) contiene una imagen de 200 x 133 píxeles, mientras que #caja2 contiene un texto centrado verticalmente.

Para que ambas cajas no se sitúen una debajo de la otra (como es lo habitual), tenemos que añadir un **width** y un **float** a cada caja:

```
#caja1{
  width:200px;
  float:left;
  margin-right:15px;
}
#caja2{
  width:200px;
  float:left;
}
```

El resultado es:



Dentista económico

Grandes ofertas
Sin listas de espera
Quirófano disponible

El problema que tenemos es que ahora queda un espacio vacío muy grande a la derecha de #caja2 (justamente en el espacio que ocupa este mismo texto). Así, cualquier cosa que pongamos en el código html DETRÁS de #caja2, **aunque no le asignemos ningún float** se colocará en este espacio vacío.

Para que este <div> se coloque **debajo**, le tendríamos que aplicar la declaración CSS **float:both**.

clear:both

La mayoría de problemas de colocación que se producen al utilizar la propiedad float se pueden solucionar de dos maneras:

1. Utilizando **clear:both** es la solución socorrida, aunque tremendamente efectiva. Si a la primera caja que viene detrás de la última caja que tiene la propiedad **float** se le aplica **clear:both**, NO aprovechará el espacio que queda libre y dicho <div> se colocará debajo de la caja anterior.

```
#caja_descolocada{
  clear:both;
}
```

2. Utilizando <div> **contenedores** es la forma más elegante y correcta, aunque también la menos utilizada. Consiste en meter #caja1 y #caja2 dentro de otro <div>, al que le daremos -al menos- una altura (**height**) determinada o un **display:inline-block**; para que automáticamente la altura se adapte al contenido.

```
<div id="cajas">
  <div id="caja1">
    </div>

  <div id="caja2">
    </div>
</div>
```

A esta estructura HTML le tenemos que añadir los correspondientes selectores CSS:

```
#cajas{
  height:131px;
}
#caja1{
  width:395px;
  float:left;
  margin-right:15px;
}
#caja2{
  width:300px;
  float:left;
}
```

Posicionamiento de las cajas

Existen tres opciones de posicionamiento de las cajas dentro del <body>.

relative:

Este es el tipo de posicionamiento por defecto.

Las cajas o <divs> se posicionan siempre en relación al resto de <divs>. Si se mueven o se modifica el tamaño de las cajas anteriores, las cajas que hayan debajo se moverán.

Con este posicionamiento NO es posible colocar una caja encima de otra.



absolute:

Al indicar un posicionamiento absoluto es posible colocar una caja en una coordenada concreta sin importar el posicionamiento del resto de cajas.

Es posible que una caja esté encima de otra.

Un ejemplo es el caso de la chincheta lila (que hay en la parte superior de esta misma página), que se coloca por encima de otras cajas, pero que al mover el scroll de la página se mueve como el resto de elementos.

fixed:

Tiene las mismas características que el posicionamiento absoluto, pero con la diferencia de que al mover el scroll del navegador la caja con posicionamiento fijo NO se moverá del lugar donde ha sido ubicada.

Un ejemplo es la bandera (ubicada en la parte superior derecha de la pantalla), que está por encima del resto de elementos y se queda fija en la esquina superior derecha. Así, aunque movamos la barra de scroll o hagamos más pequeña la ventana del navegador no se mueve de esa posición.

Otro ejemplo es el <footer> inferior de esta página, que está colocado fijo en la parte inferior de la ventana del navegador.

Se utilizaría únicamente una de los tres valores que siguen:

```
#nombre_del_div{
  position:relative;
  position:absolute;
  position:fixed;
}
```

Al definir el posicionamiento como **absolute** o como **fixed** es necesario indicar la posición o coordenada exacta en la que se desea ubicar la caja (o <div>).

Para ello podemos utilizar los siguientes valores:

- **left**: Especifica la coordenada horizontal, empezando a contar desde el extremo izquierdo.
- **right**: Especifica la coordenada horizontal, empezando a contar desde el extremo derecho.
- **top**: Especifica la coordenada vertical, empezando a contar desde el extremo superior.
- **bottom**: Especifica la coordenada vertical, empezando a contar desde el extremo inferior.

Por ejemplo, en el siguiente código se especifica un posicionamiento fijo del <footer> (como es el caso de este html) en el que éste siempre estará colocado en la parte inferior de la pantalla y en el extremo izquierdo. Así, aunque hagamos más alto o más bajo la ventana del navegador (puedes intentarlo) o exista más o menos contenido, el <footer> siempre estará fijo en la parte inferior de la misma.

```
footer{
  position:fixed;
  bottom:0px;
  left:0px;
}
```

Lo realmente importante

Hay una cosa **realmente importante** para evitar problemas de última hora.

Si ubicamos un <div> en una posición absoluta de la pantalla sin indicar nada más, será muy probable que cuando cambiemos la anchura del navegador o -todavía más grave- según tenga el usuario una anchura u otra del navegador, el <div> posicionado con valor absolute esté en un lugar muy diferente.

Para ver un ejemplo real, sube al principio de esta página, redimensiona la ventana del navegador y verás como la chincheta lila se mueve horizontalmente mientras modificas el ancho del navegador ([Subir](#)).

Para evitar este grave problema debemos introducir el <div> con posición absoluta dentro de otro <div> (contenedor) con posición relativa.

```
#contenedor{
  position:relative;
}

#div_absoluto{
  position:absolute;
}
```