



APRENDERAPROGRAMAR.COM

CLEAR CSS: LEFT, RIGHT, BOTH. ¿FLOAT CSS NO FUNCIONA? ¿EL TEXTO NO ENVUELVE UNA IMAGEN HTML? EXPLICACIÓN DE PROBLEMAS TÍPICOS CON FLOAT (CU01035D)

Sección: Cursos

Categoría: Tutorial básico del programador web: CSS desde cero

Fecha revisión: 2029

**Resumen:** Entrega nº35 del Tutorial básico: "CSS desde cero".

Autor: César Krall

## POSICIONAMIENTO FLOTANTE

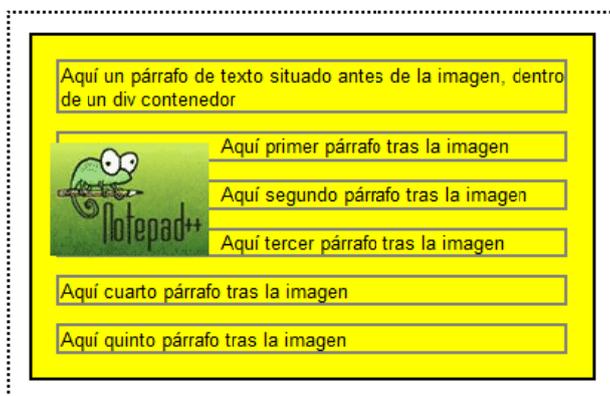
Hay varias cuestiones sobre el posicionamiento flotante que aún debemos resolver. Por ejemplo, veremos cómo usando la propiedad clear podemos definir que un elemento y aquellos que vengan después de este deben dejar de envolver al elemento flotante. También veremos qué ocurre cuando varios elementos consecutivos se flotan.



Recordaremos en primer lugar la propiedad float:

PROPIEDAD CSS float	
<b>Función de la propiedad</b>	Permite establecer un comportamiento especial para un elemento, que es desplazado tan a la derecha o izquierda como sea posible y admite ser envuelto por otros elementos.
<b>Valor por defecto</b>	none
<b>Aplicable a</b>	Elementos con un ancho, especificado o implícito como en imágenes.
<b>Valores posibles para esta propiedad</b>	<b>none</b> (el elemento tendrá comportamiento normal)
	<b>right</b> (el elemento se desplaza a la derecha tanto como es posible y admite ser envuelto).
	<b>left</b> (el elemento se desplaza a la izquierda tanto como es posible y admite ser envuelto).
	<b>inherit</b> (se heredan las características del elemento padre).
<b>Ejemplos aprenderaprogramar.com</b>	#content1 {float: left;} .elementoMonter {float: right;}

En la entrega anterior vimos cómo si a un elemento como una imagen se le aplicaba un valor de float distinto de none, los elementos a continuación de él (en nuestro ejemplo párrafos) lo envolvían.



## PROPIEDAD CLEAR

La propiedad clear indica si un elemento y los que le siguen pueden envolver a un elemento flotante precedente y cómo (por la izquierda, por la derecha o por ninguno de los dos lados).

La sintaxis a emplear es del tipo:

```
selectorElemento {clear: valorAsignado; }
```

PROPIEDAD CSS clear	
<b>Función de la propiedad</b>	Permite establecer si un elemento y los que le siguen debe envolver (wrap) a un elemento flotante precedente y cómo.
<b>Valor por defecto</b>	none
<b>Aplicable a</b>	Elementos tipo block.
<b>Valores posibles para esta propiedad</b>	<b>none</b> (el elemento y los que le siguen envolverá al elemento flotante anterior)
	<b>left</b> (el elemento y los que le siguen no envolverán a un elemento flotante anterior cuya propiedad float sea left; no habrá elementos a la derecha del flotante).
	<b>right</b> (el elemento y los que le siguen no envolverán a un elemento flotante anterior cuya propiedad float sea right; no habrá elementos a la izquierda del flotante).
	<b>both</b> (el elemento y los que le siguen no envolverán a un elemento flotante anterior por ninguno de los dos lados; reestablece el flujo "normal")
<b>inherit</b> (se heredan las características del elemento padre).	
<b>Ejemplos aprenderaprogramar.com</b>	#content1 {clear: left;} .elementoMonter {float: right;}

Usaremos este código para probar la propiedad:

```
<html>
<head> <title>Portal web - aprenderaprogramar.com</title> <meta charset="utf-8">
<link rel="stylesheet" type="text/css" href="estilosCU01035DA.css">
</head>
<body> <div>
<p>Aquí un párrafo de texto situado antes de la imagen, dentro de un div contenedor</p>

<p>Aquí primer párrafo tras la imagen</p> <p>Aquí segundo párrafo tras la imagen</p>
<p id="tercero">Aquí tercer párrafo tras la imagen</p>
<p>Aquí cuarto párrafo tras la imagen</p> <p>Aquí quinto párrafo tras la imagen</p>
</div> </body> </html>
```

Y el código CSS lo dejaremos así (pon el nombre adecuado al archivo css):

```

/* Curso CSS estilos aprenderaprogramar.com*/
*{font-family: arial; }
body {width: 510px; border-style: dotted; }
div {border-style: solid; margin: 15px; padding: 5px; background-color: yellow;}
p {text-align: justify; margin:15px; border-style:solid; border-color:grey;}
img {margin:10px; float:left;}

```

Ahora probaremos a añadir estas reglas y comprobar el resultado:

- #tercero {clear: both;} : comprobaremos cómo el párrafo con id="tercero" que corresponde al párrafo cuyo texto es "Aquí tercer párrafo..." ya no flota a la derecha de la imagen, sino que se sitúa debajo de ella restableciendo el comportamiento normal.
- #tercero {clear: right;} : comprobaremos cómo el párrafo con id="tercero" se mantiene flotando en torno a la imagen. ¿Por qué? Porque clear: right; da lugar a que los elementos no floten a la izquierda de un elemento con valor de float right. Como en este caso no tenemos un elemento con valor de float right, este código no genera ningún efecto.
- #tercero {clear: left;} : comprobaremos cómo el párrafo con id="tercero" que corresponde al párrafo cuyo texto es "Aquí tercer párrafo..." ya no flota a la derecha de la imagen, sino que se sitúa debajo de ella. ¿Por qué? Porque clear: left; da lugar a que un elemento y los que le siguen dejen de envolver a un elemento anterior cuyo valor de float sea left, como es el caso que nos ocupa.

Es fácil intuir que clear es una propiedad que se usa en combinación con la propiedad float para crear diseños atractivos.

## ¿QUÉ ENVUELVE A UN ELEMENTO FLOTANTE Y CUÁNDO LO ENVUELVE?

El posicionamiento flotante tiene diversas peculiaridades que debemos ir conociendo poco a poco.

Vamos a emplear este código HTML:

```

<html> <head>
<title>Portal web - aprenderaprogramar.com</title>
<meta charset="utf-8">
<link rel="stylesheet" type="text/css" href="estilosCU01035DB.css">
</head>
<body>
<div>
<p>Un texto antes de las cajas</p>
<div id="caja1">Caja 1</div>
<div id="caja2">Caja 2</div>
<div id="caja3">Caja 3</div>
<div id="caja4">Caja 4</div>
<p>Un texto después de las cajas</p>
<div class="limpiador"></div>
</div> </body>
</html>

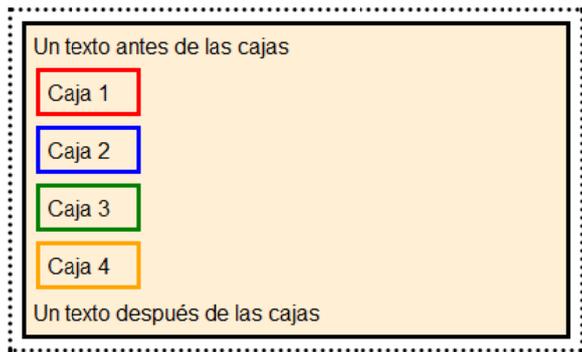
```

Y el siguiente código css de partida (ponle el nombre de archivo adecuado):

```

/* Curso CSS estilos aprenderaprogramar.com */
*{font-family: arial;}
body {width:410px; border-style: dotted;}
div {border-style: solid; margin:7px; background-color: #FFEFD5; }
div div {padding: 5px; width: 60px;}
p {margin:5px;}
.limpiador{padding:0; border-style:none;}
#caja1{ border-color:red; } #caja2{ border-color:blue;} #caja3{ border-color:green;} #caja4{border-color:orange;}
    
```

La situación de partida es esta:



Ahora vamos a realizar este cambio en el código CSS: #caja1{ border-color:red; float:right; }

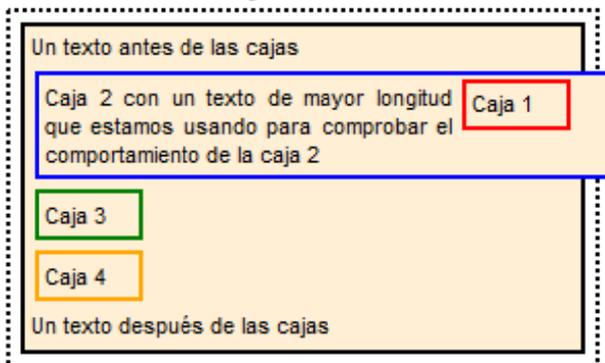
Al establecer el valor de float right comprobamos que la caja 1 se va hacia la derecha tanto como puede saliendo del flujo normal del documento. La caja 2 se sitúa en la línea en que se encontraba la caja 1, ocupando la posición "natural" que le correspondería si no existiera la caja 1, de hecho pasa algo similar a esto porque la caja 1 ha salido del flujo normal y la caja 2 "ignora a la caja 1". Ahora bien, la caja 2 no se ve que envuelva a la caja 1 ¿Por qué? En realidad la caja 2 sí es envolvente de la caja 1, lo que ocurre es que no hay texto para apreciar el efecto. Alarga el texto correspondiente al div de la caja 2 para comprobar el efecto. ¿Qué ocurre? Que el texto se alarga hacia abajo tratándose de adaptar al width definido para la caja, con lo cual tampoco comprobamos que haya comportamiento envolvente. Modifica así el código: #caja2{ border-color:blue; width:390px;}

Ahora sí podemos comprobar que el texto de la caja 2 que llega a confluir con la caja 1 tiene un comportamiento envolvente.

¿La caja 2 no envuelve a la caja 1?



Sí, pero para verlo hace falta texto y anchura



Conclusión: el comportamiento envolvente no se aprecia si no existe texto y anchura suficiente para que se pueda apreciar. En caso de no existir texto y anchura suficiente, el documento se muestra como si el flujo normal continuara ignorando al elemento flotante.

A la pregunta ¿Qué envuelve a un elemento flotante? Podemos responder que **lo que envuelve a un elemento flotante es el texto** de los elementos que vengan a continuación de él. Repetimos: lo que envuelve es el texto, no envuelven las cajas ni imágenes ni otro tipo de elementos, sólo texto.

¿Cuándo envuelve el texto de un elemento posterior a un elemento flotante? Cuando existe ancho (width) suficiente para que el siguiente elemento alcance el borde del elemento flotante y cuando el texto tiene longitud suficiente como para poder apreciar el efecto.

### ¿QUÉ OCURRE SI NO HAY ESPACIO SUFICIENTE PARA ENVOLVER?

Sobre el código inicial que habíamos planteado, vamos ahora a realizar este cambio en el código CSS: `#caja1{ border-color:red; float:left; }`

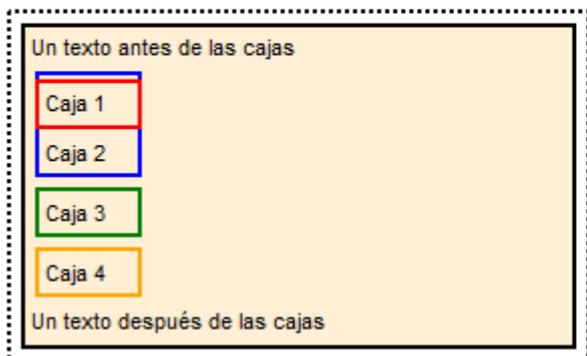
Observaremos algo extraño: la caja 2 parece que no envuelve a la caja 1. El texto de la caja 2 aparece debajo del texto de la caja 1. ¿Qué ocurre? Que el texto de la caja 2 para poder situarse a la derecha del texto de la caja 1 necesita disponer de un ancho suficiente. Si no dispone de ancho, pasa a la siguiente línea (agrandando el tamaño de su caja). Basta darle el ancho suficiente para que se vea cómo el texto de la caja 2 envuelve al texto de la caja 1 por su derecha.

Modificaremos el código así: `#caja2{ border-color:blue; width:250px; }`

Con ello comprobamos cómo el texto de la caja 2 envuelve a la caja 1.

Muchas veces un programador o diseñador web nos ha planteado por qué el texto no envuelve a una imagen HTML u otro tipo de elemento. Aquí está la respuesta. **Cuando no hay espacio suficiente para envolver el texto se sitúa debajo del elemento flotante.**

¿La caja 2 no envuelve a la caja 1?



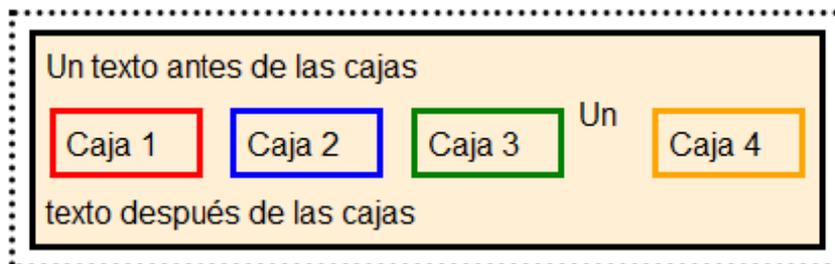
Sí, pero para verlo hace falta texto y anchura



### ¿QUÉ OCURRE SI APLICAMOS FLOAT A VARIOS ELEMENTOS CONSECUTIVOS?

Si aplicamos float a varios elementos consecutivos, los elementos se van arrimando unos a otros mientras exista espacio suficiente. Cuando ya no queda espacio, pasarán a la siguiente línea.

Modifica el código inicial que hemos usado así: #caja1{ border-color:red; float:left; } #caja2{ border-color:blue; float: left;} #caja3{ border-color:green; float: left;} #caja4{border-color:orange; float: right;}

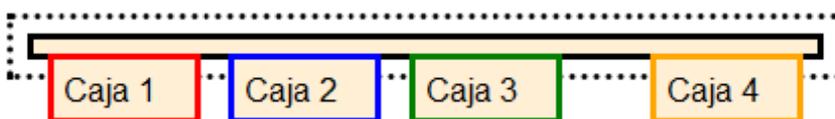


El resultado lo interpretamos de la siguiente manera. La caja 1 se sitúa tan a la izquierda como le es posible, es decir, pegada al límite del contenedor. La caja 2 trata de situarse tan a la izquierda como le es posible, pero al encontrar un elemento flotante ya posicionado no puede pegarse al borde del contenedor, sino que se alinea junto al elemento flotante que se ha encontrado. La caja 3 se comporta igual. La caja 4, en cambio, se sitúa tan a la derecha como le es posible.

El párrafo "Un texto después de las cajas" trata de envolver al último elemento flotante (la caja 4). Su texto comienza en la vertical línea que sería su posición natural si ninguna de las 4 cajas existieran (ya que han salido del flujo normal) y en la horizontal comienza desde que un elemento float deja espacio suficiente para poder insertarse. Trata de avanzar para envolver a la caja 4, pero al no haber espacio disponible salta a la siguiente línea.

Elimina ahora los párrafos "Un texto antes de las cajas" y "Un texto después de las cajas" y comprueba el resultado (manteniendo el posicionamiento flotante de las cuatro cajas, tres a la izquierda y la cuarta a la derecha).

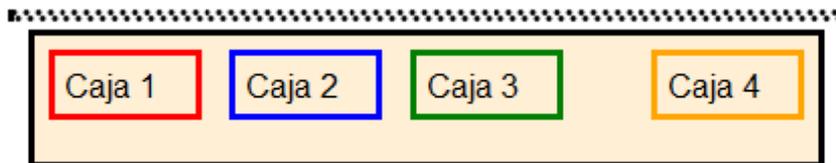
Te podrás encontrar con algo así:



¿Qué está ocurriendo? Que al salir del flujo normal los elementos flotantes dejan de ocupar espacio en su caja contenedora de forma que el valor height de la caja contenedora puede llegar a hacerse nulo. Esto en alguna ocasión puede que ni siquiera lo notemos. No obstante, en algunos casos como aquellos donde tenemos un borde, color de fondo o imagen de fondo para la caja contenedora puede ser un problema.

¿Cómo evitar que la caja contenedora no incluya a las cajas flotantes hijas y hacer que se dibujen sus fondos correctamente englobando a los elementos flotantes? Hay varias maneras para hacer esto.

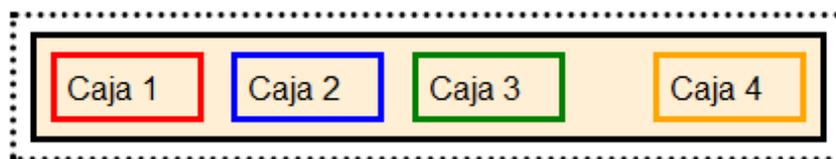
Una forma de hacerlo es convertir en flotante también a la caja madre. Una caja flotante amplía su tamaño para incluir a sus cajas hijas, de forma que declarando a la madre como flotante ya se ampliará automáticamente. Si hacemos esta modificación: div {border-style: solid; margin:7px; background-color: #FFefd5; float:left; } el resultado será similar a este:



Vemos cómo el elemento body aparece “comprimido” sin altura en la parte superior, porque al ser tanto la caja madre como las hijas flotantes, están fuera del flujo del documento y el elemento body no adquiere altura. Tener en cuenta que en este caso estamos viendo el elemento body porque hemos aplicado la propiedad border-style:dotted; pero esto lo hacemos en el marco de un curso de aprendizaje, en la práctica profesional el elemento body no lleva borde prácticamente nunca.

Ahora veamos otra alternativa distinta a convertir en flotante a la caja madre. Vamos a dejar la caja madre como `div {border-style: solid; margin: 7px; background-color: #FFefd5;}` y por el contrario vamos a hacer esta modificación: `.limpiador{padding:0; border-style:none; clear:both; }`

El resultado será similar a este:



¿Por qué ocurre esto? Porque al incluir un elemento, en este caso un div, con valor `clear:both;` que indica que debe mostrarse sin flotar, la caja se comporta como si debajo de los elementos flotantes estuviera el elemento con flujo normal, de modo que se muestra conteniendo tanto a los elementos flotantes como a este elemento. Si el elemento “limpiador” está vacío realmente no se muestra, pero como el navegador interpreta que hay algo, adapta las dimensiones de la caja.

Hay más formas de lograr un resultado similar a este (por ejemplo aplicando la propiedad `overflow: hidden;` a la caja madre), pero no vamos a entrar a explicarlas ahora.

### EJERCICIO

Define un documento HTML donde a través del uso de las propiedades float y clear y de las anteriores propiedades que hemos visto a lo largo del curso crees un diseño con este aspecto:

- a) En primer lugar se deben mostrar 8 cajas div de 50x50 píxeles, con margin-right de 5 píxeles para cada una de ellas, y cada una de ellas con distinto color de fondo, alineadas en horizontal hacia la izquierda gracias al uso de float left.
- b) En segundo lugar se debe mostrar un div con un texto y color de fondo amarillo, con margen superior e inferior de 20 píxeles, abarcando todo el ancho disponible.
- c) En tercer lugar se deben mostrar 3 cajas div de 200x50 píxeles, con margin-right de 5 píxeles para cada una de ellas, y cada una de ellas con distinto color de fondo, alineadas en horizontal hacia la derecha gracias al uso de float right.

d) En cuarto lugar un div de fondo de color #DEB887 que ocupe todo el ancho disponible y contenga el texto <<Curso CSS aprenderaprogramar.com>>

Nota: si es necesario, usa los div contenedores auxiliares que te sean necesarios.

Para comprobar si tu código y respuestas son correctas puedes consultar en los foros aprenderaprogramar.com.

**Próxima entrega:** CU01036D

**Acceso al curso completo** en aprenderaprogramar.com -- > Cursos, o en la dirección siguiente:  
[http://aprenderaprogramar.com/index.php?option=com\\_content&view=category&id=75&Itemid=203](http://aprenderaprogramar.com/index.php?option=com_content&view=category&id=75&Itemid=203)