

5.4. Posicionamiento float

El **posicionamiento float** es uno de los más utilizados, sobre todo al definir la estructura de las páginas, como se verá más adelante. Una caja posicionada mediante la propiedad `float`, se desplaza hasta la zona más a la izquierda o más a la derecha de la línea en la que se debería mostrar si no se desplazara.

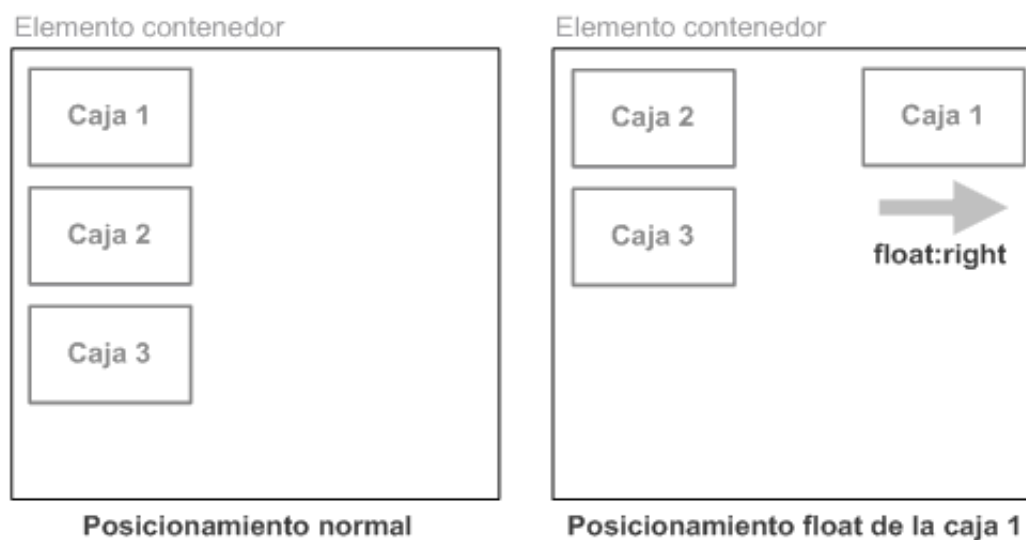


Figura 5.7. Ejemplo de posicionamiento float de una caja

Una caja desplazada mediante `float` no pertenece al posicionamiento normal de un documento, por lo que los elementos de bloque anteriores y posteriores se visualizan como si la caja desplazada no existiera.

Si en el anterior ejemplo la "Caja 1" se posiciona mediante un `float: left` el resultado sería el que se muestra en la imagen:

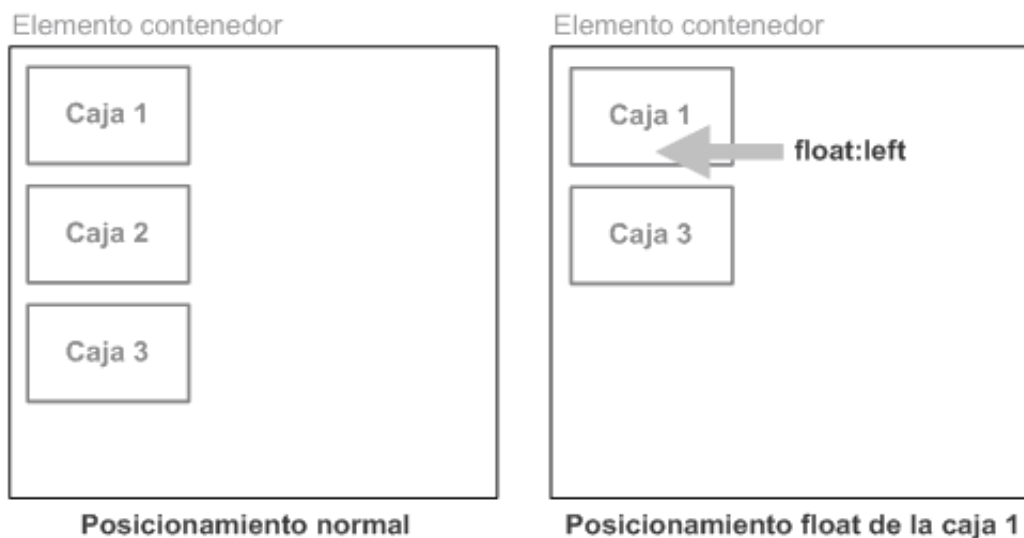


Figura 5.8. Ejemplo de posicionamiento float de una caja

En las imágenes anteriores, el resto de cajas ocupan la posición libre dejada por la "Caja 1" original. Como la "Caja 1" está desplazada a la izquierda, se coloca por encima de la nueva posición que ocupa la "Caja 2", a la que cubre por completo.

Si existen otras cajas desplazadas hacia la izquierda o derecha, la nueva caja desplazada se coloca al lado de las demás cajas. El siguiente ejemplo desplaza las tres cajas:

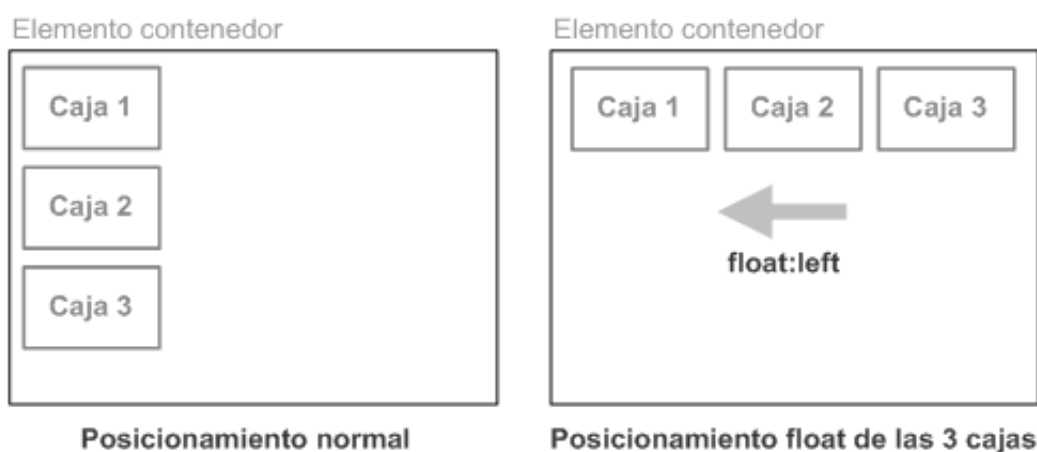


Figura 5.9. Ejemplo de posicionamiento float de varias cajas

Si no existe sitio en la línea actual, la caja pasa a la siguiente línea hasta que encuentra el sitio necesario para mostrarse a la izquierda o derecha de la línea.

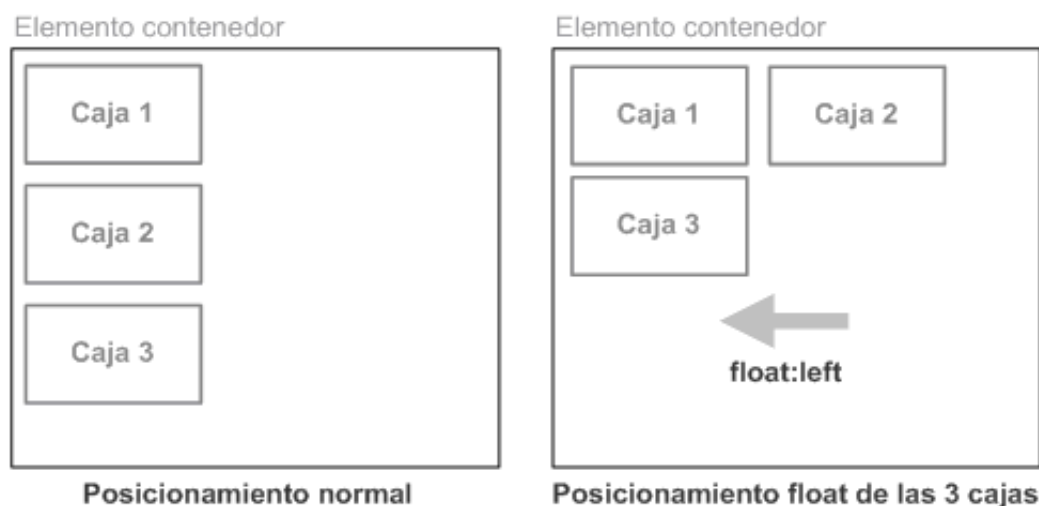


Figura 5.10. Ejemplo de posicionamiento float cuando no existe sitio suficiente

Una caja posicionada mediante `float` influye en la disposición de todas las demás cajas. CSS permite definir si el resto de los elementos fluyen alrededor de la caja desplazada o no lo hacen.

Los elementos en línea que se encuentran al lado de las cajas desplazadas mediante `float` adaptan su anchura al espacio libre dejado por la caja desplazada. Si en la línea donde se encuentra la caja desplazada no existe sitio necesario para los contenidos de los elementos en línea, estos se visualizan en la línea inmediatamente inferior.

float	Posicionamiento float
Valores	<code>left</code> <code>right</code> <code>none</code> <code>inherit</code>
Se aplica a	Todos los elementos
Valor inicial	<code>none</code>
Descripción	Establece el tipo de posicionamiento float del elemento

Los posibles valores de la propiedad determinan el posicionamiento del elemento y el comportamiento de los elementos adyacentes. Si se indica un valor `left`, el elemento se desplaza hasta el punto más a la izquierda posible en esa misma línea (si no existe sitio en esa línea, el elemento baja una línea y se muestra lo más a la izquierda posible en esa nueva línea). El resto de elementos adyacentes se adaptan y fluyen alrededor del elemento desplazado.

El valor `right` tiene un funcionamiento idéntico, salvo que en este caso el elemento se desplaza hacia la derecha. El valor `none` permite eliminar el posicionamiento y que el elemento se muestre en su posición original.

Los elementos que se encuentran alrededor de un elemento que ha sido posicionado mediante `float`, adaptan sus contenidos para que fluyan alrededor del elemento posicionado:

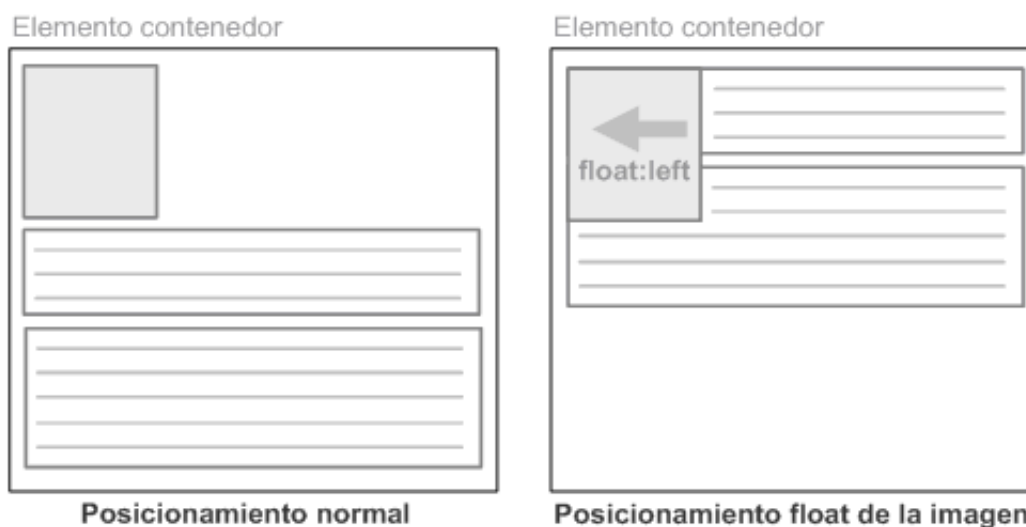


Figura 5.11. Elementos que fluyen alrededor de un elemento posicionado mediante `float`

La regla CSS que se aplica en la imagen del ejemplo anterior es:

Uno de los principales motivos para la creación del posicionamiento `float` fue precisamente la posibilidad de colocar imágenes alrededor de las cuales fluye el texto.

La flexibilidad de CSS permite controlar la forma en la que los contenidos fluyen alrededor de los contenidos posicionados mediante `float`. De hecho, en muchas ocasiones es admisible que el texto del primer párrafo fluya alrededor de una imagen, pero el resto de los párrafos deberían mostrarse en su totalidad y no fluyendo alrededor de la imagen:

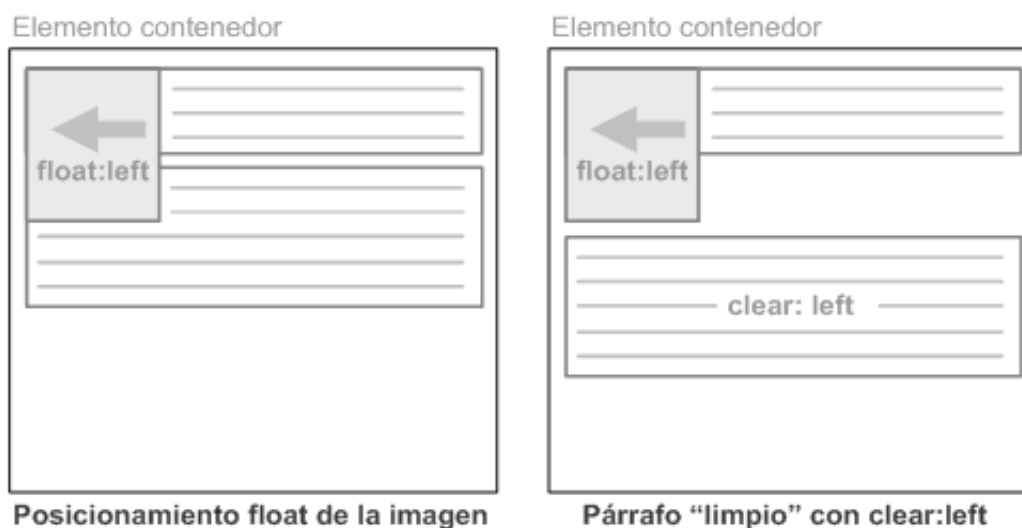


Figura 5.12. Forzando a que un elemento no fluya alrededor de otro elemento posicionado mediante float

La propiedad `clear` permite contrarrestar el comportamiento por defecto de los `float` y permite forzar a un elemento a mostrarse debajo de cualquier elemento posicionado con `float`.

La regla CSS que se aplica al segundo párrafo del ejemplo anterior sería la siguiente:

```
<p style="clear: left;">...</p>
```

clear Despejar los elementos adyacentes

Valores none | left | right | both | inherit

Se aplica a Todos los elementos de bloque

Valor inicial none

Descripción Permite limpiar el lado derecho, izquierdo o ambos de un elemento, para que no se muestren elementos adyacentes

La propiedad `clear` indica el lado del elemento HTML que no debe ser adyacente a otros elementos desplazados mediante `float`.

Si se indica el valor `left`, el elemento se desplaza de forma descendente hasta que pueda colocarse en una línea en la que no haya ningún elemento en el lado izquierdo. La especificación oficial de CSS explica este comportamiento como *"un desplazamiento descendente hasta que el borde superior del elemento esté por debajo del borde inferior de cualquier elemento desplazado hacia la izquierda"*.

Si se indica el valor `right`, el comportamiento es el mismo, salvo que en este caso se tienen en cuenta los elementos desplazados hacia la derecha.

El valor `both` despeja los lados izquierdo y derecho del elemento, ya que desplaza el elemento de forma descendente hasta que el borde superior se encuentre por debajo de cualquier borde inferior de los elementos desplazados hacia la izquierda y hacia la derecha.

Como se verá más adelante, la propiedad `clear` es imprescindible para la creación de estructuras y layouts de páginas complejas. En el ejercicio anterior, se utilizaba `float` para desplazar los dos elementos:

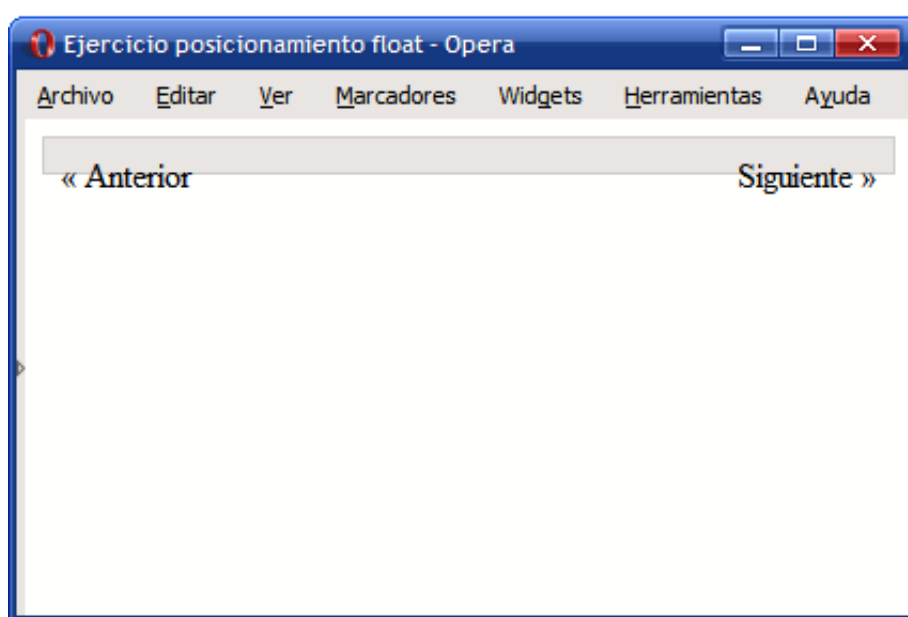


Figura 5.13. Visualización incorrecta de dos elementos posicionados mediante `float`

Como los dos elementos `` creados dentro del elemento `<div>` se han posicionado mediante `float`, los dos han salido del flujo normal del documento. Así, el elemento `<div>` no tiene contenidos y por eso no llega a cubrir el texto de los dos elementos ``:

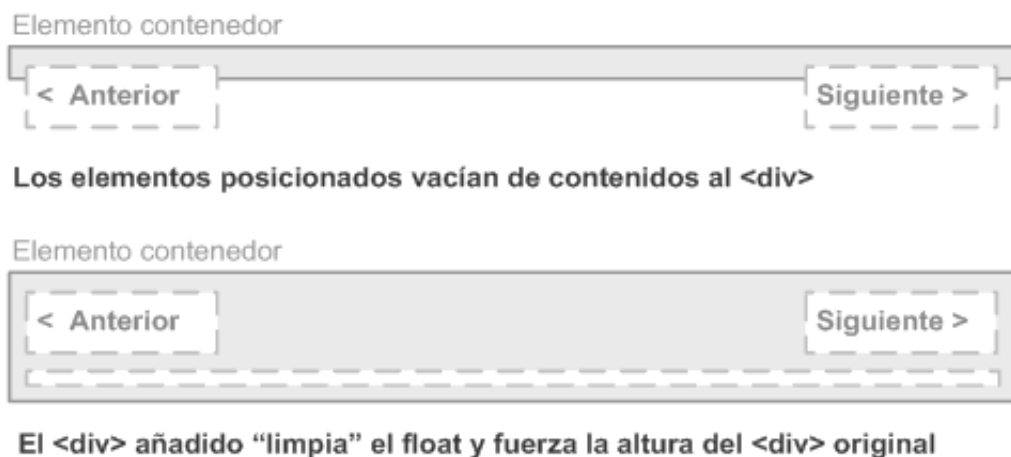


Figura 5.14. Esquema del problema y solución de la visualización incorrecta de 2 elementos posicionados mediante float

La solución consiste en añadir un elemento adicional invisible que *limpie* el float forzando a que el `<div>` original cubra completamente los dos elementos ``. El código HTML y CSS final se muestra a continuación:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1" />
<title>Ejercicio posicionamiento float</title>
<style type="text/css">
div#paginacion {
    border: 1px solid #CCC;
    background-color: #E0E0E0;
    padding: .5em;
}

.derecha {
    float: right;
}

.izquierda {
    float: left;
```

```

}

div.clear {
  clear: both;
}
</style>
</head>

<body>
<div id="paginacion">
  <span class="izquierda">&laquo; Anterior</span>
  <span class="derecha">Siguiete &raquo;</span>
  <div class="clear"></div>
</div>
</body>
</html>

```

Al añadir un `<div>` con la propiedad `clear: both`, se tiene la seguridad de que el `<div>` añadido se va a mostrar debajo de cualquier elemento posicionado con `float` y por tanto, se asegura que el `<div>` original tenga la altura necesaria como para encerrar a todos sus contenidos posicionados con `float`.

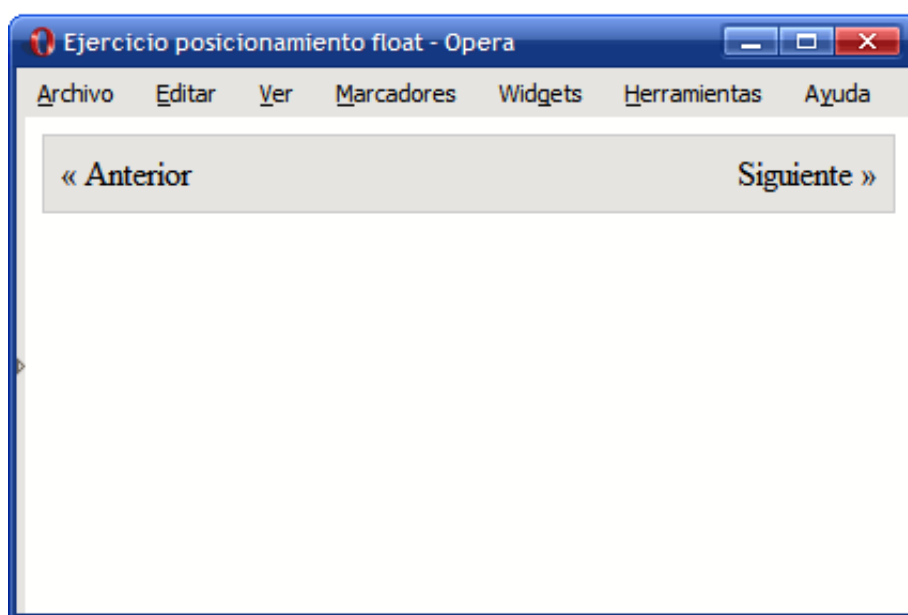


Figura 5.15. Visualización correcta de 2 elementos posicionados mediante float

Además de un elemento `<div>` invisible, también se puede utilizar un `<p>` invisible

o un `<hr />` invisible.