

Cajas

Los navegadores crean una caja alrededor de cada elemento para determinar el área que ocupan.

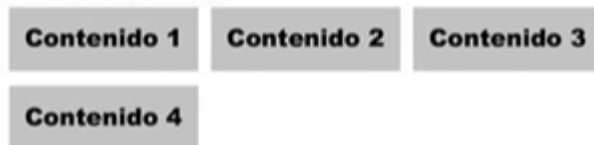
Para organizar estas cajas en pantalla, los elementos se clasifican en Block (bloque) e Inline (en línea). Su principal diferencia es que los elementos Block tienen un tamaño personalizado y generan saltos de línea, mientras que los Inline tienen un tamaño determinado por su contenido y no generan saltos de línea.

Por sus características, los elementos Block se colocan de uno en uno en las distintas líneas y los elementos Inline se colocan uno al lado del otro en la misma línea (a menos que no haya espacio horizontal disponible).

Elementos Block



Elementos Inline



Los elementos Block son apropiados para crear columnas y secciones en una página web, y los Inline son adecuados para representar contenido.

Por esta razón, elementos que definen la estructura de un documento como `<section>`, `<nav>`, `<header>`, `<footer>` o `<div>` se declaran como elementos Block por defecto, y otros que representan el contenido de esos elementos como ``, `` o `` se declaran como elementos Inline.

Display

El navegador determina que un elemento sea tipo Block o Inline, pero podemos cambiar esa condición desde CSS con la propiedad `Display`.

La propiedad Display define el tipo de caja usado para representar el elemento en pantalla. Existen varios valores disponibles para esta propiedad:

- **none**: elimina el elemento
- **block**: muestra el elemento en una nueva línea y con un tamaño personalizado
- **inline**: muestra el elemento en la misma línea
- **inline-block**: muestra el elemento en la misma línea y con un tamaño personalizado

Los elementos estructurales se configuran con el valor block y los elementos que representan el contenido se configuran como inline. Modificando el tipo de elemento con la propiedad display, las antiguas versiones de navegadores no reconocen los elementos incorporados por HTML 5 y los consideran inline por defecto.

Si asignamos a algún elemento la propiedad display con el elemento none, el documento se presenta como si el elemento no existiera. Es útil cuando queremos cambiar el documento dinámicamente desde JavaScript o cuando usamos diseño web adaptable para diseñar nuestro sitio web.

Los elementos Block tienen dos características: producen un salto de línea y pueden adoptar un tamaño personalizado. Con la propiedad inline-block, los elementos se posicionarán uno al lado del otro en la misma fila, pero con el tamaño que queramos.

Elementos Inline-Block



Modelo de caja tradicional

El modelo de caja tradicional establece que los elementos pueden flotar a cada lado de la ventana y compartir espacio en la misma línea con otros elementos, sin importar su tipo. Por ejemplo, si tenemos dos elementos Block que representan columnas, podemos posicionar una columna a la izquierda y otra a la derecha.

Las propiedades que ofrece CSS son:

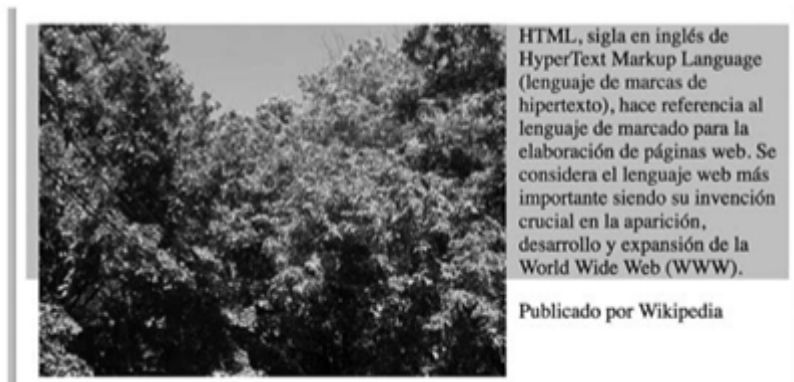
- **float**: Permite a un elemento flotar hacia un lado u otro y ocupar el espacio disponible incluso cuando tiene que compartir línea con otro elemento. Los valores disponibles son:
 - none: el elemento no flota
 - left: el elemento flota a la izquierda
 - right: el elemento flota a la derecha
- **clear**: Restaura el flujo normal del documento y no permite que el elemento siga flotando. Los valores disponibles son:
 - none
 - left
 - right
 - both

Contenido flotante

Las propiedades `float` y `clear` se usaron originalmente para que el contenido flote alrededor de un elemento. Por ejemplo, si queremos que un texto se muestre al lado de una imagen, podemos hacerla flotar hacia un lado y el texto compartirá con la imagen el espacio disponible.

Cuando un elemento flota hacia un lado, los siguientes elementos flotan a su alrededor ocupando el espacio disponible.

Los navegadores no pueden calcular el tamaño de un contenedor a partir del tamaño de elementos flotantes, por lo que si el elemento afectado por `float` es más alto que el resto de los elementos de la misma línea, desbordará el contenedor.

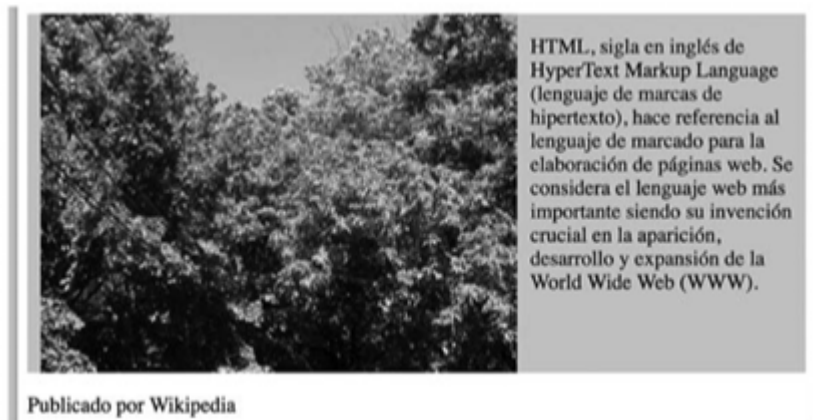


El navegador estima el tamaño del contenedor según el tamaño del texto, por lo que la imagen se sitúa fuera de los límites del contenedor.

Una forma de asegurarnos de que el contenedor es lo suficientemente alto como para contener los elementos flotantes es asignándole la propiedad `overflow` con el valor `auto`. Esto fuerza al navegador a calcular el tamaño del contenedor considerando todos los elementos en su interior.

```
section {  
  background-color: #CCCCCC;  
  overflow: auto;  
}  
section img {  
  float: left;  
  margin: 0px 10px;  
}
```

La propiedad overflow no deja que el contenido desborde a su contenedor extendiendo el tamaño de este o incluyendo barras de desplazamiento.



Otra alternativa para normalizar el flujo del documento es declarar la propiedad clear en el elemento que esté a continuación del elemento flotante dentro del contenedor.

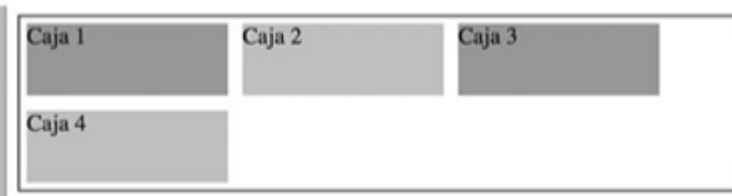
Como no siempre tenemos un elemento hermano después de un elemento flotante, está técnica requiere que agreguemos un elemento adicional al documento.

Cajas flotantes

La propiedad clear no afecta a otros aspectos del elemento y no agrega barras de desplazamiento como la propiedad overflow. Por eso, clear es la propiedad que se implementa en el modelo de caja tradicional para organizar la estructura de un documento. Con float y clear podemos controlar de forma precisa dónde se mostrarán los elementos en pantalla.

Cuando el contenido de un elemento flota, el elemento padre no puede calcular su propia altura desde la altura de su contenido. Esta es la razón por la que, cada vez que tenemos elementos que flotan, debemos recuperar el flujo normal en el elemento con la propiedad clear.

En el caso de que el contenedor no tenga espacio suficiente, los elementos que no se puedan ubicar en la misma línea se moverán a una nueva.



Si tenemos más espacio disponible en el contenedor del que necesitan las cajas, el espacio restante se ubica a los lados (izquierda o derecha, dependiendo el valor de float). Podemos ubicar el espacio en medio haciendo flotar algunas cajas a la izquierda y otras a la derecha.



La caja 4 aparece antes que la 3 por el orden en el que se han declarado en CSS.

Posicionamiento absoluto

La posición de un elemento puede ser relativa o absoluta. Con una posición relativa, las cajas se colocan detrás de la otra en el espacio designado por el contenedor y, si no hay espacio suficiente o los elementos no son flotantes, se colocan en una nueva línea. Este es el modo de posicionamiento por defecto.

El posicionamiento absoluto nos permite especificar las coordenadas exactas en las que queremos posicionar cada elemento.

Las siguientes propiedades están disponibles para este propósito:

- **position**: define el tipo de posicionamiento
 - static (se posiciona siguiendo el flujo normal del documento)
 - relative (se posiciona según la posición original del elemento)
 - absolute (se posiciona con una posición absoluta relativa al contenedor del elemento)
 - fixed (se posiciona con una posición absoluta relativa a la ventana del navegador)
- **top**: esta propiedad especifica la distancia entre el margen superior del elemento y el margen superior de su contenedor
- **bottom**: esta propiedad especifica la distancia entre el margen inferior del elemento y el margen inferior de su contenedor
- **left**: esta propiedad especifica la distancia entre el margen izquierdo del elemento y el margen izquierdo de su contenedor
- **right**: esta propiedad especifica la distancia entre el margen derecho del elemento y el margen derecho de su contenedor

Las propiedades top, bottom, left y right se aplican en ambos tipos de posicionamiento, pero trabajan de formas diferentes.

Cuando el elemento se ubica con posicionamiento relativo el elemento se desplaza pero el diseño no se modifica.



Con posicionamiento absoluto el elemento se limita del diseño, por lo que el resto de los elementos también se desplazan para ocupar el nuevo espacio libre.



Los índices negativos se consideran más bajos que los asignados por defecto. Si asignamos el valor -1 a “Caja 1”, esta se moverá debajo de “Caja 2”.



Al usar posicionamiento relativo y absoluto el diseño del documento se modifica, por lo que esta técnica no se usa para organizar los elementos en pantalla, sino para crear efectos en los que los elementos ocultos se muestran respondiendo a acciones del usuario.

Columnas

CSS incluye un grupo de propiedades para facilitar la creación de columnas:

- **column-count**: Especifica el número de columnas que el navegador tiene que generar para distribuir el contenido del elemento
- **column-width**: Declara el ancho de las columnas. Su valor se puede declarar como:
 - auto (por defecto)
 - Cualquiera de las unidades de CSS, como píxeles o %
- **column-span**: Se aplica a elementos dentro de la caja para determinar si el elemento se ubicará en una columna o repartido entre varias. Sus valores son:
 - all (todas las columnas)
 - none (por defecto)
- **column-fill**: Determina cómo se repartirá el contenido entre las columnas. Sus valores son:
 - auto (columnas completadas de forma secuencial)
 - balance (el contenido se divide en partes iguales entre las columnas)
- **column-gap**: Define el tamaño del espacio entre las columnas. Acepta cualquier valor de las unidades de CSS, como píxeles o %
- **columns**: Nos permite declarar los valores de column-count y column-width a la vez

Las propiedades para generar las columnas se deben aplicar al contenedor.

La propiedad column-gap define el tamaño del espacio entre columnas, pero esta separación es un espacio vacío. CSS ofrece propiedades para generar una línea que ayude al usuario a visualizar la división:

- **column-rule-style**: Define el estilo de línea usada para representar la división.
 - hidden (por defecto)
 - dotted
 - dashed
 - solid
 - doubled
 - groove
 - ridge
 - inset
 - outset
- **column-rule-color**: Especifica el color de la línea
- **column-rule-width**: Especifica el ancho de la línea
- **column-rule**: Permite definir todos los valores de la línea a la vez

Modelo de caja flexible

El objetivo principal de un modelo de caja es ofrecer un mecanismo con el que dividir el espacio disponible, y crear filas y columnas que son parte del diseño de una página web. Sin embargo, las herramientas que ofrece el modelo de caja tradicional no cumplen este objetivo, ya que no se puede definir cómo distribuir las cajas ni especificar su tamaño.

Aunque el modelo de caja flexible tiene sus ventajas frente al modelo de caja tradicional, aún está en fase experimental y algunos navegadores no pueden procesar sus propiedades.

Contenedor flexible

El modelo de caja flexible aprovecha las herramientas que usa el modelo de caja tradicional, como el posicionamiento absoluto y las columnas, pero en lugar de hacer flotar los elementos organiza las cajas usando contenedores flexibles.

En este modelo, cada grupo de cajas debe estar incluido dentro de otra caja que se encarga de configurar sus características.

Elementos flexibles

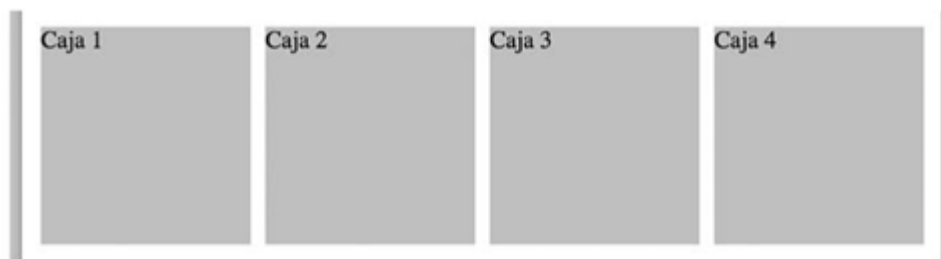
Para que un elemento dentro de un contenedor flexible se vuelva flexible, debemos declararlo como tal con las siguientes propiedades:

- **flex-grow**: Declara la proporción en la cual el elemento se va a expandir o encoger. La proporción se determina considerando los valores asignados a los demás elementos de la caja (elementos hermanos)
- **flex-shrink**: Declara la proporción en la que el elemento se va a reducir. La proporción se determina a partir de los valores asignados a los demás elementos de la caja (elementos hermanos)
- **flex-basis**: Declara un tamaño inicial para el elemento
- **flex**: Permite declarar los valores de flex-grow, flex-shrink y flex-basis a la vez.

Las cajas flexibles se expanden o encogen para ocupar el espacio libre dentro de la caja padre. La distribución del espacio depende de las propiedades del resto de las cajas.

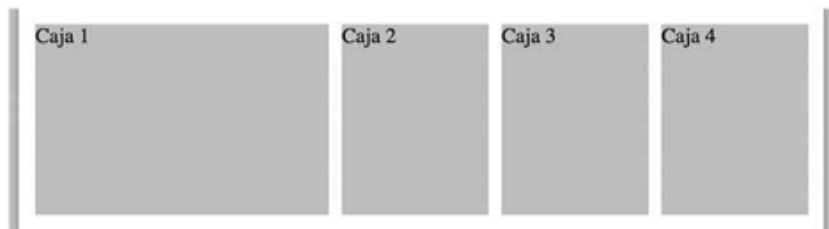
Utilizando el siguiente código, conseguiremos que las cuatro cajas tengan el mismo tamaño:

```
#cajapadre {
  display: flex;
  width: 600px;
}
#cajapadre > div {
  height: 145px;
  margin: 5px;
  background-color: #CCCCCC;
}
#caja-1 {
  flex: 1;
}
#caja-2 {
  flex: 1;
}
#caja-3 {
  flex: 1;
}
#caja-4 {
  flex: 1;
}
```



Si al código anterior le aumentamos el valor a la propiedad flex de alguna caja, esta crecerá y las demás perderán tamaño.

```
#cajapadre {
  display: flex;
  width: 600px;
}
#cajapadre > div {
  height: 145px;
  margin: 5px;
  background-color: #CCCCCC;
}
#caja-1 {
  flex: 2;
}
#caja-2 {
  flex: 1;
}
#caja-3 {
  flex: 1;
}
#caja-4 {
  flex: 1;
}
```



La propiedad flex-basis nos permite definir un tamaño inicial para influenciar al navegador a la hora de distribuir el espacio disponible entre las cajas, pero las cajas aún se expanden o reducen más allá de ese valor. CSS ofrece propiedades para poner limitaciones al tamaño de una caja:

- **max-width**: Especifica el ancho máximo permitido para el elemento.
- **min-width**: Especifica el ancho mínimo permitido para el elemento.
- **max-height**: Especifica la altura máxima permitida para el elemento.
- **min-height**: Especifica la altura mínima permitida para el elemento.

Todas estas propiedades aceptan valores en todas las unidades disponibles en CSS, como píxeles o %

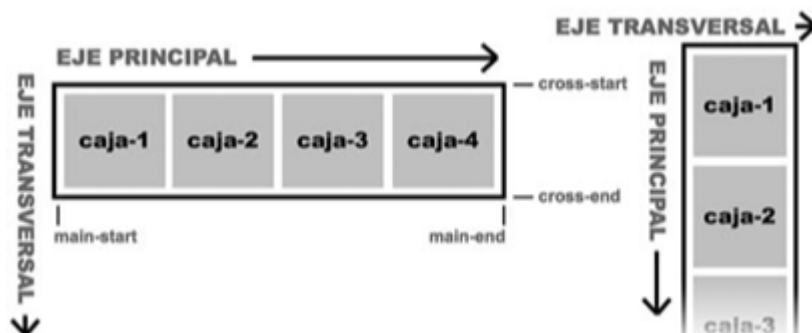
Organizando elementos flexibles

De forma predeterminada, los elementos dentro de un contenedor flexible se muestran en la misma línea de forma horizontal, pero no se organizan con una orientación estándar.

Un contenedor flexible usa ejes para describir la orientación de su contenido:

- Eje principal: Es aquel en el que se presenta el contenido (normalmente equivale a la orientación horizontal)
- Eje transversal: Es el perpendicular al eje principal (normalmente equivale a la orientación vertical).

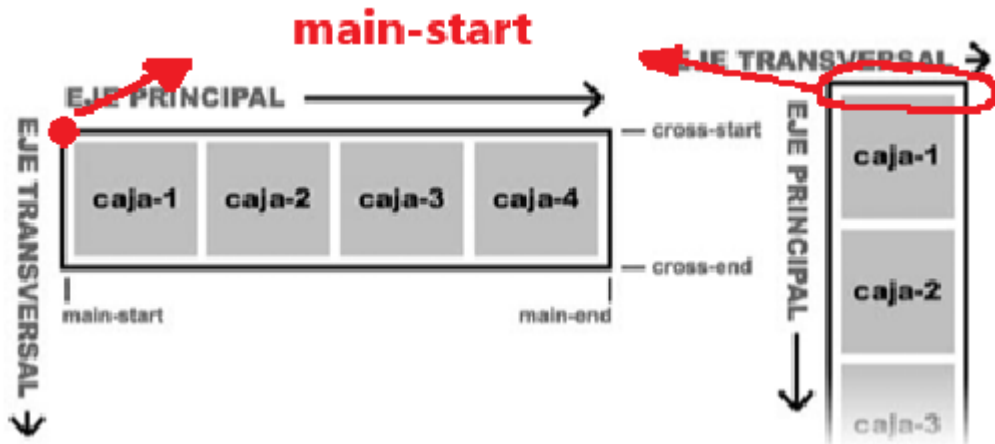
Si la orientación cambia, los ejes se desplazan junto con el contenido.



Las propiedades definidas para este modelo trabajan con estos ejes y organizan los elementos desde sus extremos:

- **main-start**
- **main-end**
- **cross-start**
- **cross-end**

En la imagen de la izquierda, main-start hace referencia a la esquina superior izquierda, mientras que en la imagen de la derecha main-start hace referencia a la línea superior completa.



Podemos cambiar la organización de las cajas. CSS ofrece las siguientes propiedades:

- **flex-direction:** Define el orden y la orientación de las cajas en un contenedor flexible
 - row (por defecto)
 - row-reverse
 - column
 - column-reverse
- **order:** Especifica el orden de las cajas. Acepta números enteros que determinan la ubicación de cada caja
- **justify-content:** Determina cómo se va a distribuir el espacio libre
 - flex-start
 - flex-end
 - center
 - space-between
 - space-around
- **align-items:** Alinea las cajas en el eje transversal
 - flex-start
 - flex-end
 - center
 - baseline
 - stretch
- **flex-wrap:** Determina si se permiten crear múltiples líneas de cajas
 - nowrap
 - wrap
 - wrap-reverse
- **align-content:** Alinea las líneas de cajas en el eje vertical
 - flex-start
 - flex-end
 - center
 - space-between
 - space-around
 - stretch