

Esta página ha sido traducida del inglés por la comunidad. Aprende más y únete a la comunidad de MDN Web Docs.

## El modelo de caja

Todo en CSS tiene una caja alrededor, y comprender estas cajas es clave para poder crear diseños con CSS o para alinear elementos con otros elementos. En este artículo, echaremos un vistazo más de cerca al *modelo de cajas* en CSS con el que vas a poder crear diseños de compaginación más complejos con una comprensión de cómo funciona y la terminología relacionada.

<b>Prerrequisitos:</b>	Conocimientos básicos de informática, tener el <a href="#">software básico</a> instalado, conocimientos básicos de cómo <a href="#">trabajar con archivos</a> , conocimientos básicos de HTML (véase <a href="#">Introducción al HTML</a> ) y nociones de CSS (véase <a href="#">Primeros pasos con CSS</a> ).
<b>Objetivo:</b>	Aprender sobre el modelo de cajas en CSS, en qué consiste el modelo de cajas y cómo cambiar al modelo alternativo.

## Cajas en bloque y en línea

En CSS, en general, hay dos tipos de cajas: **cajas en bloque** y **cajas en línea**. Estas características se refieren al modo como se comporta la caja en términos de flujo de página y en relación con otras cajas de la página:

Si una caja se define como un bloque, se comportará de las maneras siguientes:

- La caja fuerza un salto de línea al llegar al final de la línea.
- La caja se extenderá en la dirección de la línea para llenar todo el espacio disponible que haya en su contenedor. En la mayoría de los casos, esto significa que la caja será tan ancha como su contenedor, y llenará el 100% del espacio disponible.
- Se respetan las propiedades [width](#) y [height](#).
- El relleno, el margen y el borde mantienen a los otros elementos alejados de la caja.

A menos que decidamos cambiar el tipo de visualización a en línea, elementos como los encabezados (por ejemplo, `<h1>`) y todos los elementos `<p>` usan por defecto `block` como tipo de visualización externa.

Si una caja tiene una visualización externa de tipo `inline`, entonces:

- La caja no fuerza ningún salto de línea al llegar al final de la línea.
- Las propiedades [width](#) y [height](#) no se aplican.
- Se aplican relleno, margen y bordes verticales, pero no mantienen alejadas otras cajas en línea.
- Se aplican relleno, margen y bordes horizontales, y mantienen alejadas otras cajas en línea.

El elemento `<a>`, que se utiliza para los enlaces, y los elementos `<span>`, `<em>` y `<strong>` son ejemplos de elementos que se muestran en línea por defecto.

El tipo de caja que se aplica a un elemento está definido por los valores de propiedad `display`, como `block` y `inline`, y se relaciona con el valor **externo** (*outer*) de visualización (`display`).

## Aparte: tipos de visualización interna y externa

En este punto, será mejor que también expliquemos los tipos de visualización **interna** y **externa**. Como se mencionó anteriormente, las cajas en CSS tienen un tipo de visualización *externa*, que define si se trata de una caja en bloque o en línea.

Sin embargo, las cajas también tienen un tipo de visualización *interna*, que determina cómo se disponen los elementos dentro de esa caja. De forma predeterminada, los elementos dentro de una caja se presentan en [flujo normal](#), lo que significa que se comportan como otros elementos de tipo en bloque o en línea (como se explicó anteriormente).

Sin embargo, podemos cambiar el tipo de visualización interna utilizando valores de `display`, como `flex`. Si en un elemento establecemos `display: flex`, el tipo de visualización externa es de tipo bloque (`block`), pero el tipo de visualización interna cambia a flexible (`flex`). Cualquier elemento que sea hijo directo de esta caja pasará a comportarse como un elemento de tipo flex, de acuerdo con las reglas que se establecen en la especificación de [Flexbox](#), tema que veremos más adelante.

**Nota:** Para obtener más información acerca de los valores de visualización y el modo como funcionan las cajas en las disposiciones en bloque y en línea, echa un vistazo a la guía [Disposiciones en bloque y en línea](#) <sup>(inglés)</sup> de MDN.

A medida que vayas aprendiendo más detalles sobre el diseño CSS, te irás encontrando con el valor `flex` y con otros valores internos que puedan presentar tus cajas, por ejemplo, `grid`.

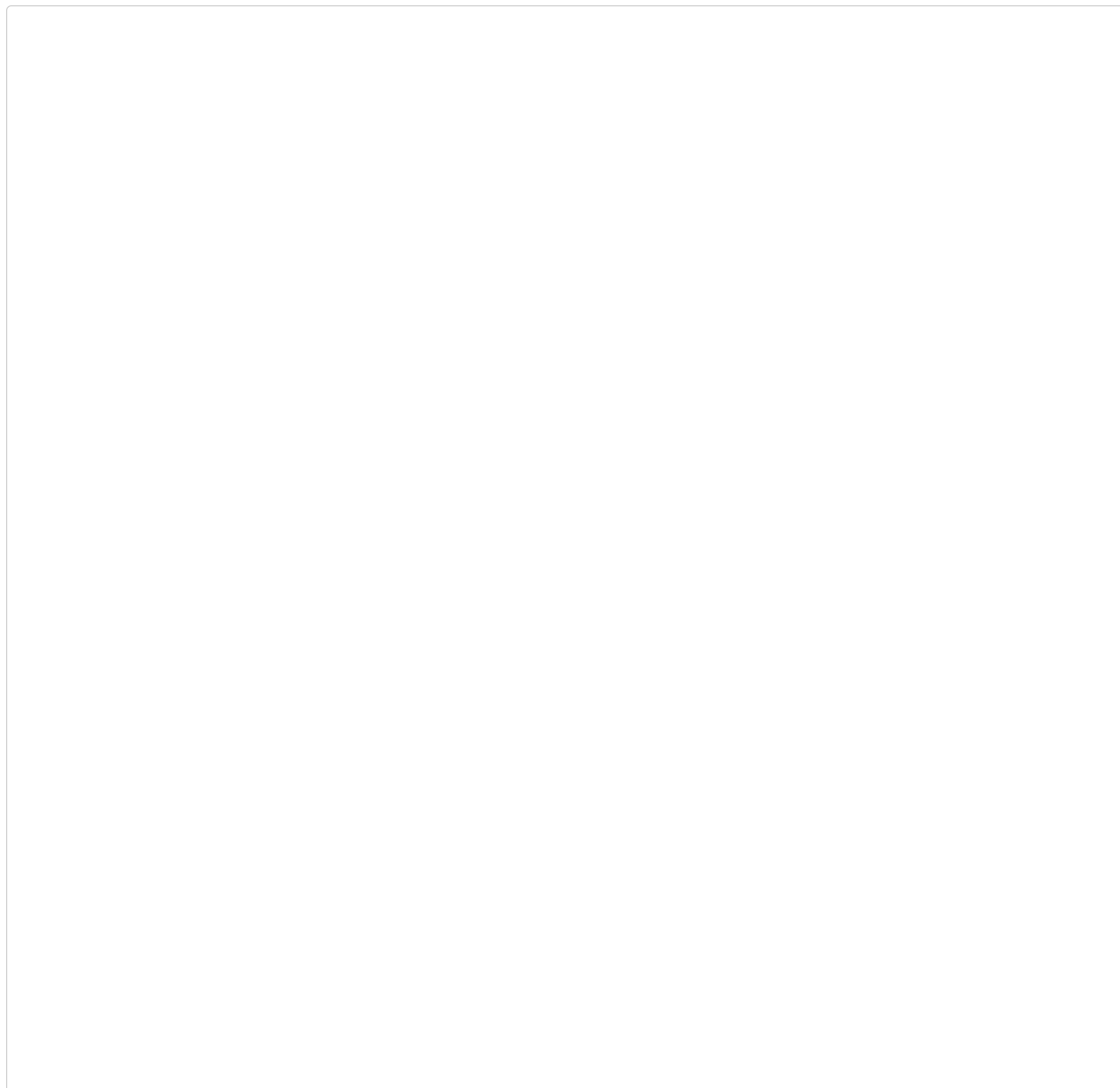
Sin embargo, la disposición en bloque y en línea es la forma predeterminada cómo se comportan las cosas en la web; como ya dijimos, a veces esto se conoce como *flujo normal*, porque nuestras cajas se dispondrán en bloque o en línea, si no reciben ninguna otra instrucción.

## Ejemplos de diferentes tipos de visualización

Sigamos adelante y veamos algunos ejemplos. A continuación tenemos tres elementos HTML diferentes, todos con visualización externa de tipo `block`. El primero es un párrafo, que tiene un borde añadido con CSS. El navegador representa esto como una caja en bloque, por lo que el párrafo comienza en una línea nueva y se expande por todo el ancho disponible.

El segundo es una lista, que se presenta usando `display: flex`. Esto establece una disposición flexible para los elementos que están dentro del contenedor; sin embargo, la lista en sí misma es una caja que se comporta en bloque y, como el párrafo, se expande por todo el ancho del contenedor y fuerza un salto de línea al llegar al final de línea.

Debajo hay un párrafo a nivel de bloque, dentro del cual hay dos elementos `<span>`. Estos elementos normalmente serían de tipo `inline`; sin embargo, uno de los elementos tiene una clase de bloque, y lo hemos establecido como `display: block`.



Podemos ver cómo se comportan los elementos `inline` en el ejemplo siguiente. Los elementos `<span>` del primer párrafo están en línea de manera predeterminada y, por lo tanto, no fuerzan ningún salto de línea.

También hay un elemento `<ul>` que se establece como `display: inline-flex`, que crea una caja con un comportamiento de tipo en línea alrededor de algunos elementos de tipo `flex`.

Finalmente, hay dos párrafos configurados con `display: inline`. El contenedor flexible en línea y los párrafos fluyen todos juntos en línea, en lugar de dividirse en líneas nuevas como lo harían si se mostraran como elementos de bloque.

**En el ejemplo puedes cambiar `display: inline` por `display: block` o `display: inline-flex` y por `display: flex` para alternar entre estos modos de visualización.**

En artículos posteriores encontrarás cosas como el diseño flexible. El aspecto clave a recordar aquí es que cambiar el valor de la propiedad `display` puede cambiar entre el modo de visualización exterior en bloque y en línea de una caja, que cambia la forma en que se presenta junto con otros elementos en la disposición en pantalla.

En el resto de este artículo, nos concentraremos en el tipo de visualización externa.

## ¿Qué es el modelo de cajas CSS?

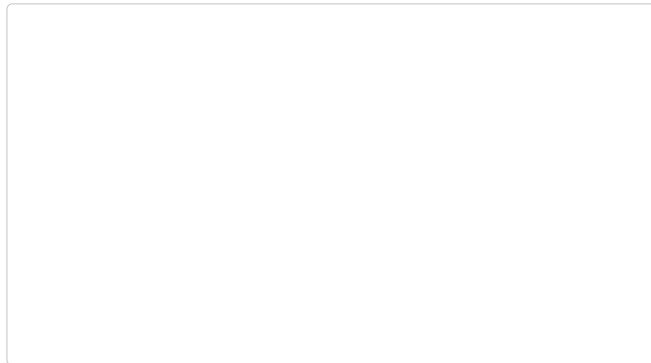
El modelo de cajas CSS completo se aplica a cajas que presentan comportamiento en bloque; las cajas con comportamiento en línea solo usan una parte del comportamiento definido en el modelo de cajas. El modelo define cómo funcionan juntas las diferentes partes de una caja (margen, borde, relleno y contenido) para crear una caja que puedas ver en tu página. Para complicarlo un poco más, hay un modelo de cajas estándar y un modelo de cajas alternativo.

### Partes de una caja

Al hacer una caja de tipo bloque en CSS tenemos los elementos siguientes:

- El **contenido de la caja** (o *content box*): El área donde se muestra el contenido, cuyo tamaño puede cambiarse utilizando propiedades como [width](#) y [height](#).
- El **relleno de la caja** (o *padding box*): El relleno es espacio en blanco alrededor del contenido; es posible controlar su tamaño usando la propiedad [padding](#) y otras propiedades relacionadas.
- El **borde de la caja** (o *border box*): El borde de la caja envuelve el contenido y el de relleno. Es posible controlar su tamaño y estilo utilizando la propiedad [border](#) y otras propiedades relacionadas.
- El **margen de la caja** (o *margin box*): El margen es la capa más externa. Envuelve el contenido, el relleno y el borde como espacio en blanco entre la caja y otros elementos. Es posible controlar su tamaño usando la propiedad [margin](#) y otras propiedades relacionadas.

El diagrama siguiente muestra estas capas:



## El modelo de cajas CSS estándar

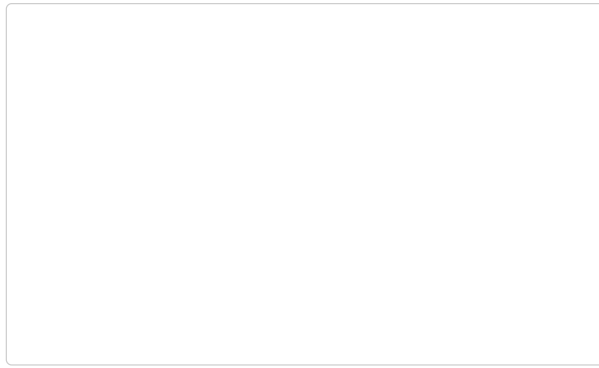
En el modelo de cajas estándar, cuando estableces los atributos `width` y `height` para una caja, defines el ancho y el alto del *contenido de la caja*. Cualquier área de relleno y borde se añade a ese ancho y alto para obtener el tamaño total que ocupa la caja. Esto se muestra en la imagen que encontrarás a continuación.

Si suponemos que la caja tiene el CSS siguiente, que establece los valores para las propiedades `width`, `height`, `margin`, `border`, y `padding`:

CSS

```
.box {  
  width: 350px;  
  height: 150px;  
  margin: 10px;  
  padding: 25px;  
  border: 5px solid black;  
}
```

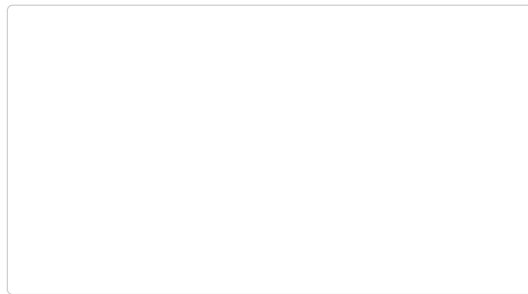
El espacio que ocupa nuestra caja usando el modelo de cajas estándar será en realidad de 410 px ( $350 + 25 + 25 + 5 + 5$ ); y su altura, de 210 px ( $150 + 25 + 25 + 5 + 5$ ), porque el área de relleno y el borde se añaden al ancho que se utiliza para el contenido de la caja.



**Nota:** El margen no se cuenta para el tamaño real de la caja; por supuesto, afecta al espacio total que la caja ocupa en la página, pero solo al espacio de fuera de la caja. El área de la caja se termina en el borde, no se extiende hasta el margen.

## El modelo de cajas CSS alternativo

Podrías pensar que es más bien incómodo tener que sumar el borde y el área de relleno para obtener el tamaño real de la caja, ¡y tienes razón! Por este motivo, CSS introdujo un modelo de caja alternativo algún tiempo después del modelo de cajas estándar. Con este modelo, cualquier ancho es el ancho de la caja visible en la página, por lo tanto, el ancho del área de contenido es ese ancho menos el ancho para el relleno y el borde. El mismo CSS que hemos usado antes daría entonces el resultado siguiente (ancho = 350 px, altura = 150 px).



Por defecto, los navegadores usan el modelo de cajas estándar. Si deseas activar el modelo de cajas alternativo para un elemento, hazlo configurando `box-sizing: border-box`. Con ello, le dices al navegador que tome como el borde de la caja el área definida por cualquier tamaño que establezcas.

CSS

```
.box {  
  box-sizing: border-box;  
}
```

Si quieres que todos tus elementos usen el modelo de cajas alternativo (opción común entre los desarrolladores) debes establecer la propiedad `box-sizing` en el elemento `<html>`. Luego debes configurar todos los demás elementos para que hereden ese valor, como se ve en el fragmento de código siguiente. Si deseas comprender qué hay detrás, consulta el [artículo de CSS-Tricks sobre el tamaño de las cajas](https://developer.mozilla.org/es/docs/Learn_web_development/Core/Styling_basics/Box_model).

CSS

```
html {  
  box-sizing: border-box;  
}  
*,
```

```
*::before,  
*::after {  
  box-sizing: inherit;  
}
```

**Nota:** Un dato curioso es que Internet Explorer usaba por defecto el modelo de cajas alternativo, y no disponía de ningún mecanismo para cambiarlo.

## Jugar con los modelos de cajas

En el ejemplo siguiente puedes ver dos cajas. Ambas tienen una clase `.box`, lo que les da los mismos atributos `width`, `height`, `margin`, `border` y `padding`. La única diferencia es que la segunda caja se ha configurado para utilizar el modelo de cajas alternativo.

**¿Puedes cambiar el tamaño de la segunda caja (añadiendo CSS a la clase `.alternate`) para que su anchura y altura coincidan con las de la primera caja?**

**Nota:** Puedes encontrar la solución [aquí](#) .

## Utilizar las DevTools del navegador para ver el modelo de cajas

Las [herramientas del desarrollador de tu navegador](#) pueden facilitar la comprensión del modelo de cajas. Si inspeccionas un elemento con las DevTools de Firefox, puedes ver el tamaño del elemento más su margen, área de relleno y borde. Inspeccionar un elemento de esta manera es un modo excelente de descubrir si tu caja es en realidad del tamaño que crees que es.



## Márgenes, relleno y bordes

Ya has visto las propiedades [margin](#) , [padding](#) y [border](#) que usamos en el ejemplo anterior. Las propiedades que hemos usado en ese ejemplo son **propiedades abreviadas** y nos permiten establecer los cuatro lados de la caja a la vez. Estas propiedades abreviadas también tienen propiedades sin abreviar equivalentes, que permiten tener control sobre los diferentes lados de la caja de forma individual.

Vamos a explorar estas propiedades más detalladamente.

### Margen

El margen es un espacio invisible que hay alrededor de la caja. Aleja el resto de elementos de la caja. Los márgenes pueden tener valores positivos o negativos. Establecer un margen negativo para un lado de tu caja puede hacer que se superponga con otros elementos de la página. Tanto si utilizas el modelo de cajas estándar como el alternativo, el margen siempre se añade después de haber calculado el tamaño de la caja que se ve.

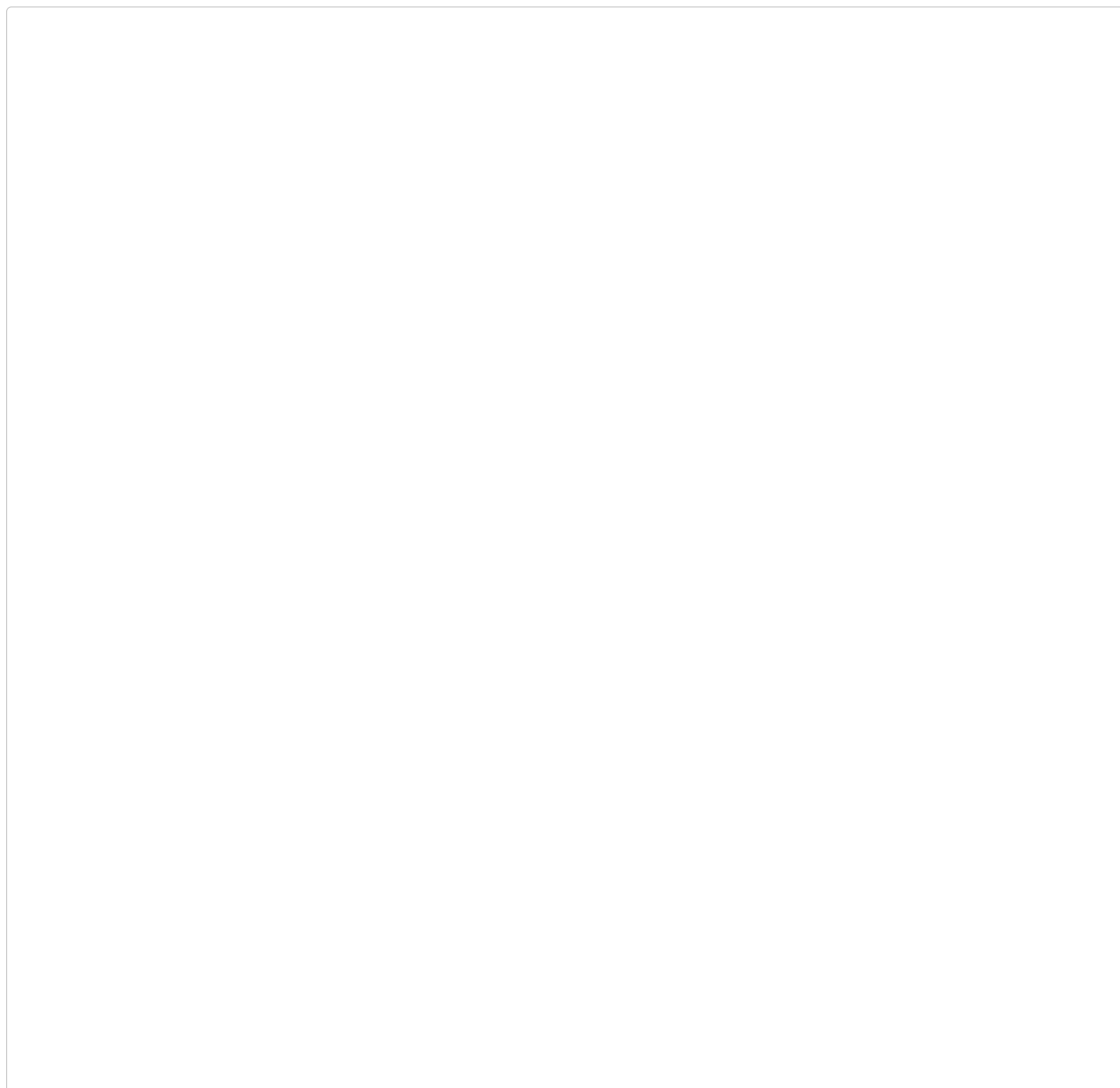
Podemos controlar todos los márgenes de un elemento a la vez usando la propiedad [margin](#) , o cada lado individualmente usando las propiedades equivalentes sin abreviar:

- [margin-top](#) <sup>(inglés)</sup>



- [margin-right](#)
- [margin-bottom](#)
- [margin-left](#) (inglés)

En el ejemplo siguiente, cambia los valores de margen para ver cómo se empuja la caja debido al espacio que el margen crea o se elimina (si es un margen negativo) entre este elemento y el elemento que lo contiene.



## Colapso del margen

Un punto clave a la hora de entender los márgenes es el concepto de colapso del margen. Si tienes dos elementos cuyos márgenes se tocan, esos márgenes se combinan para convertirse en un solo margen, cuyo tamaño es el del margen más grande.

En el ejemplo siguiente hay dos párrafos. El párrafo superior tiene un atributo `margin-bottom` de 50 píxeles. El segundo párrafo tiene un atributo `margin-top` de 30 píxeles. Los márgenes colapsan, por lo que el margen real entre las cajas es de 50 píxeles, y no el total de ambos márgenes.

**Pruébalo ajustando el atributo `margin-top` del segundo párrafo a 0. El margen visible entre los dos párrafos no cambiará, sino que conservará los 50 píxeles fijados en el atributo `bottom-margin` del primer párrafo.**



Hay una serie de reglas que establecen cuándo los márgenes colapsan y cuándo no. Para obtener más información, consulta la página web sobre [entender el colapso de márgenes](#). Por ahora solo debes recordar que el colapso de los márgenes es algo que puede suceder. Si creas un espacio con márgenes y no obtienes el espacio que esperas, probablemente es que se haya producido algún colapso de márgenes.

## Bordes

El borde se dibuja entre el margen y el área de relleno de una caja. Si utilizas el modelo de cajas estándar, el tamaño del borde se añade a los elementos `width` y `height` que establecen el alto y el ancho de la caja. Si utilizas el modelo de cajas alternativo, el tamaño del borde reduce el tamaño de la caja de contenido, porque ocupa una parte del alto y el ancho disponibles.

Hay una gran cantidad de propiedades que sirven para aplicar estilo a los bordes: hay cuatro bordes y cada borde tiene un estilo, un ancho y un color que podemos modificar.

Puedes establecer el ancho, el estilo o el color de los cuatro bordes a la vez utilizando la propiedad [border](#).

Para establecer las propiedades de cada lado de forma individual, puedes utilizar:

- [border-top](#)
- [border-right](#)
- [border-bottom](#)
- [border-left](#)

Para establecer el ancho, el estilo o el color de todos los lados, usa lo siguiente:

- [border-width](#)
- [border-style](#)
- [border-color](#)

Para establecer el ancho, el estilo o el color de un solo lado, puedes usar una de las propiedades no abreviadas:

- [border-top-width](#) (inglés)
- [border-top-style](#) (inglés)
- [border-top-color](#)
- [border-right-width](#) (inglés)
- [border-right-style](#) (inglés)
- [border-right-color](#) (inglés)
- [border-bottom-width](#)
- [border-bottom-style](#)
- [border-bottom-color](#)
- [border-left-width](#) (inglés)
- [border-left-style](#) (inglés)
- [border-left-color](#)

En el ejemplo siguiente, hemos utilizado varios ejemplos de la lista anterior para crear bordes. Juega con las diferentes propiedades para comprobar que entiendes cómo funcionan. Las páginas de MDN sobre las propiedades de los bordes te proporcionan información sobre los diferentes estilos entre los que puedes elegir para los bordes.



## Relleno

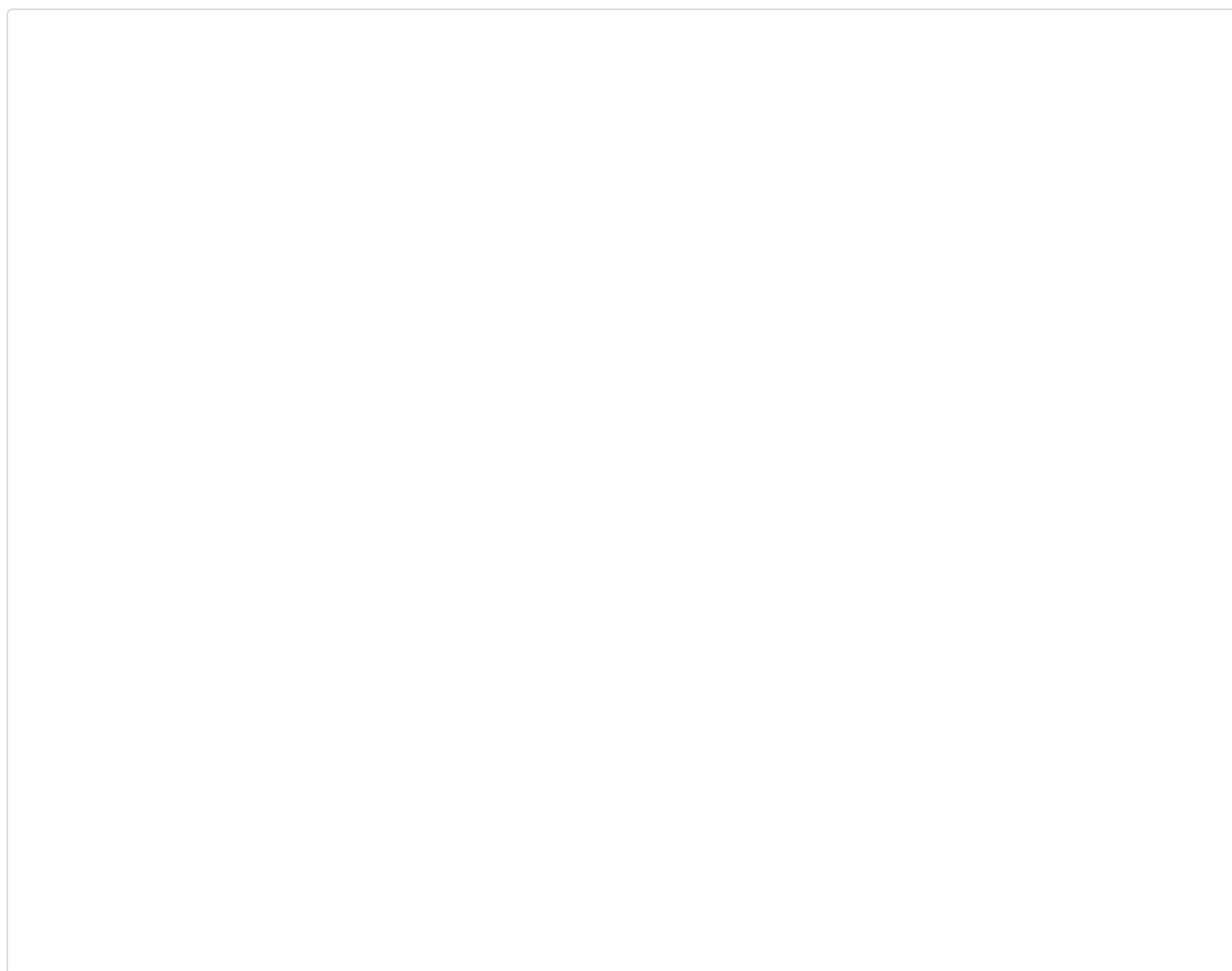
El relleno se encuentra entre el borde y el área de contenido. A diferencia de los márgenes, el relleno no puede tomar valores negativos, por lo que el valor debe ser 0 o positivo. Cualquier fondo aplicado a tu elemento se mostrará detrás del área de relleno y, generalmente, se usa para mantener el contenido alejado del borde.

Podemos controlar el área de relleno para todos los lados de un mismo elemento usando la propiedad [padding](#), o para cada uno de los lados usando las propiedades equivalentes:

- [padding-top](#)
- [padding-right](#) (inglés)
- [padding-bottom](#)
- [padding-left](#) (inglés)

Si cambias los valores para el relleno en la clase `.box` del ejemplo siguiente, puedes ver que cambia dónde comienza el texto en relación con la caja.

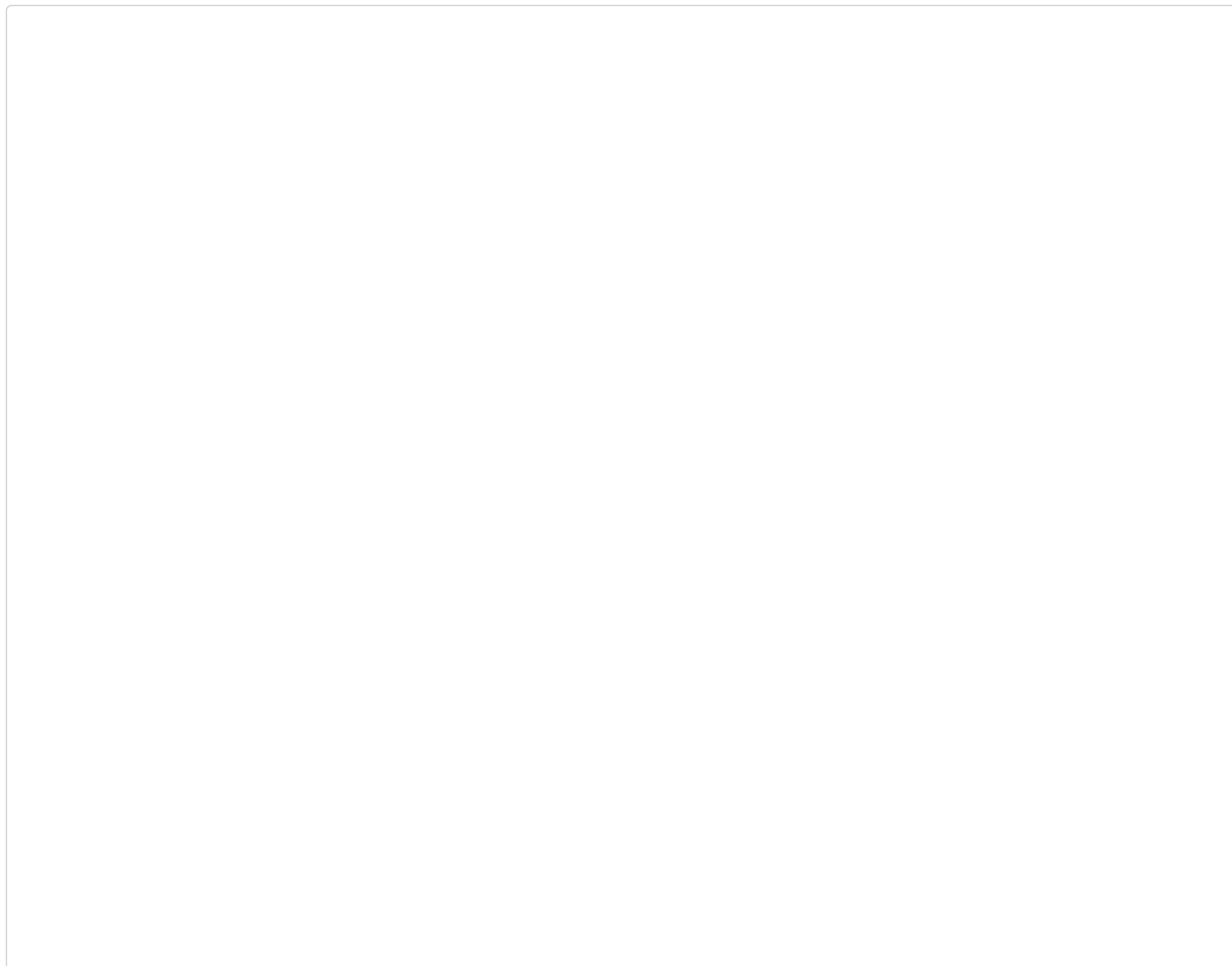
También puedes cambiar el relleno en la clase `.container`, que abrirá el espacio entre el contenedor y la caja. El área de relleno se puede cambiar para cualquier elemento y abrirá espacio entre su borde y lo que esté dentro del elemento.



## El modelo de cajas y las cajas en línea

Todo lo anterior se aplica por completo a las cajas en bloque. Algunas de las propiedades también pueden aplicarse a las cajas en línea, como las que crea un elemento `<span>`.

En el ejemplo siguiente hay un elemento `<span>` dentro de un párrafo al que hemos aplicado las propiedades `width`, `height`, `margin`, `border`, y `padding`. Puedes ver que la anchura y la altura se ignoran. Se respetan el margen, el relleno y el borde, pero no cambian la relación de otro contenido con respecto a nuestra caja en línea, por lo que el relleno y el borde se superponen a otras palabras en el párrafo.



## El uso de display: inline-block

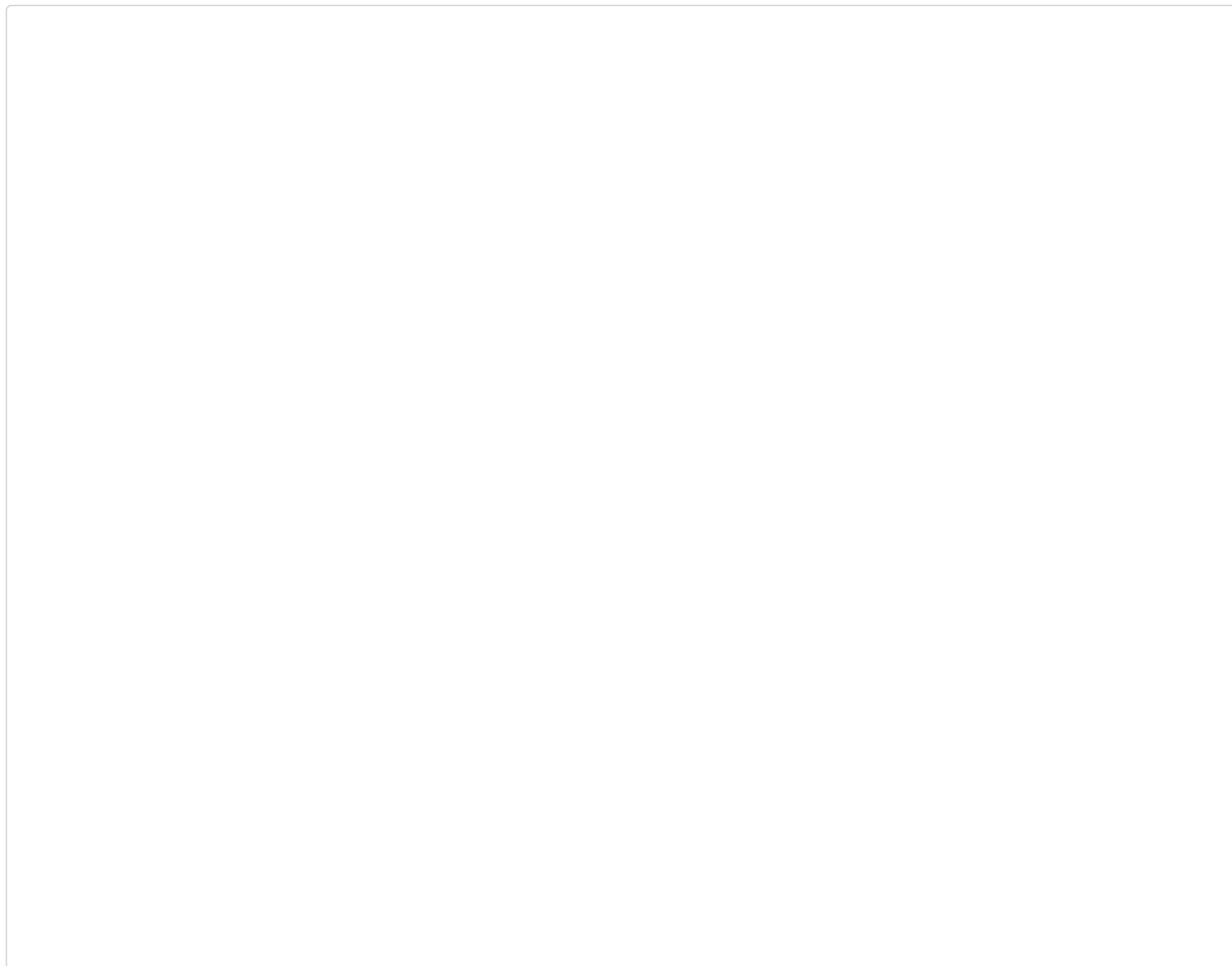
Hay un valor especial de `display` que proporciona un punto medio entre `inline` y `block`. Esto es útil para situaciones en las que no deseas que un elemento fuerce un salto de línea, pero sí deseas que se respeten las propiedades `width` y `height` para evitar superposiciones como la que se ve arriba.

Un elemento con `display: inline-block` conforma un subconjunto de los elementos en bloque que ya conocemos:

- Se respetan las propiedades de ancho y alto.
- El relleno, el margen y el borde mantienen los otros elementos alejados de la caja.

Sin embargo, no se fuerza un salto de línea, y solo se hace más grande que su contenido si añades las propiedades `width` y `height` explícitamente.

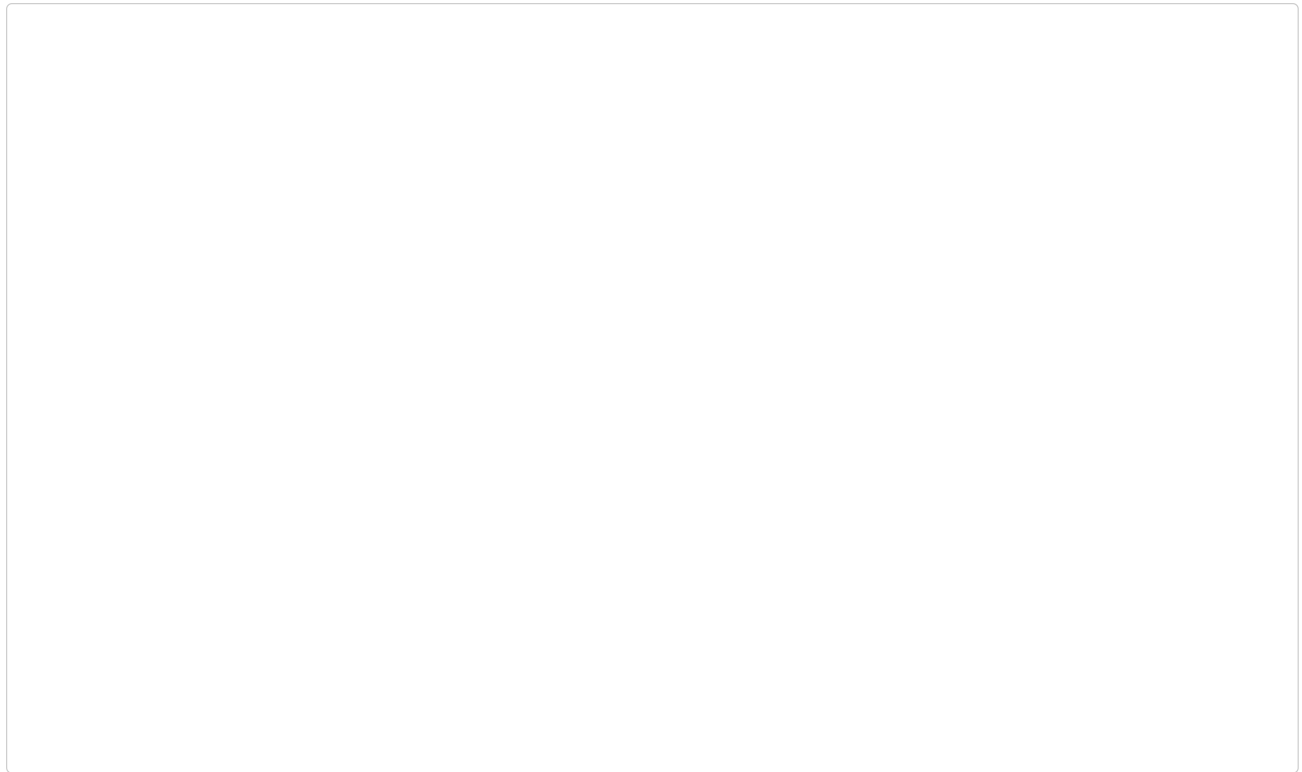
**En el ejemplo siguiente hemos añadido `display: inline-block` a nuestro elemento `<span>`. Cámbialo por `display: block` o elimina la línea para ver la diferencia entre ambos modelos de visualización.**



Esto puede ser útil cuando deseas dar a un enlace un área de impacto más grande añadiendo `padding`. `<a>` es un elemento en línea como `<span>`; puedes usar `display: inline-block` para configurar el área de relleno para facilitar al usuario hacer clic en el enlace.

Esto se ve con bastante frecuencia en las barras de navegación. La navegación siguiente se muestra en una fila usando `flexbox` y hemos añadido una área de relleno al elemento `<a>` porque queremos poder cambiar su color de fondo ( `background-color` ) cuando se pasa el ratón por encima de `<a>` . El área de relleno parece superponerse al borde del elemento `<ul>` . Esto se debe a que `<a>` es un elemento en línea.

**Añade `display: inline-block` a la regla con el selector `.links-list a` y verás cómo se soluciona este problema, al hacer que otros elementos respeten el área de relleno.**



## Pon a prueba tus habilidades

Hemos cubierto mucho terreno en este artículo. ¿Recuerdas la información más relevante? Encontrarás más pruebas para verificar que has retenido esa información en [Test your skills: The Box Model](#) <sup>(inglés)</sup>.

## Resumen

Eso es lo que hay que entender sobre el modelo de cajas. Es posible que en el futuro desees volver a este artículo si alguna vez te lías con los tamaños de las cajas en la disposición de tu página web.

En el artículo siguiente veremos cómo se pueden usar los [fondos y bordes](#) para hacer que tus simples cajas presenten un aspecto más interesante.

### Help improve MDN

Was this page helpful to you?

[Learn how to contribute.](#)

This page was last modified on 21 dic 2024 by [MDN contributors](#).

