

# Modelo de Caja / Display

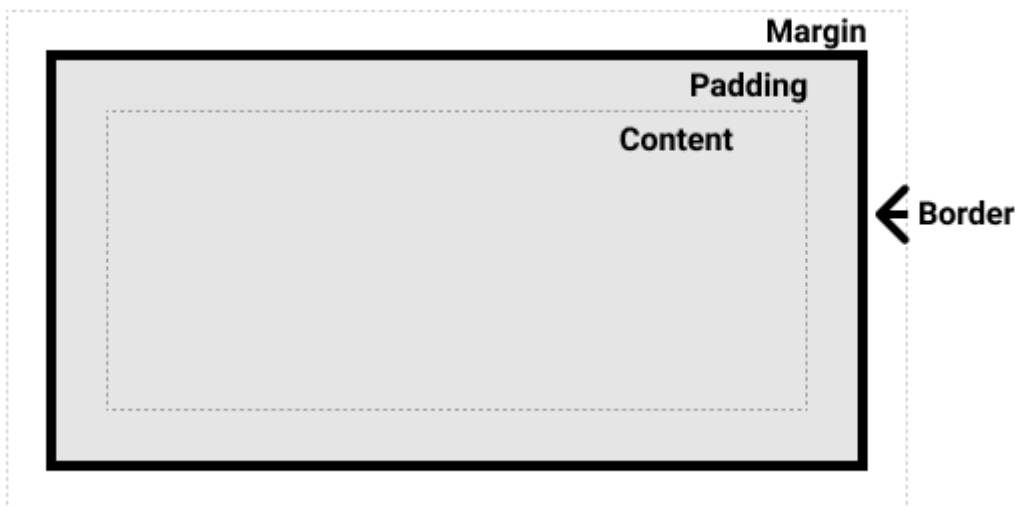
---

El Modelo de Caja en HTML es un concepto fundamental en la representación y el diseño de elementos en una página web. Este concepto se adentra en la manera en que se estructuran y presentan los elementos HTML en el entorno del navegador web. Piensa en cada elemento HTML como si fuera un objeto contenido en una "caja" rectangular dentro del Modelo de Caja, y esta caja tiene múltiples facetas que desempeñan un papel crucial en la experiencia de diseño.

## Partes de la Caja en el Modelo de Caja de HTML

En el Modelo de Caja de HTML, una caja típica tiene las siguientes partes:

1. **Contenido (Content):** Esta es la parte interna de la caja que contiene el contenido real del elemento, como texto, imágenes u otros elementos HTML anidados.
2. **Padding:** El padding es un espacio transparente que rodea el contenido del elemento. Se puede especificar el valor del padding en CSS para controlar la distancia entre el contenido y el borde de la caja.
3. **Borde (Border):** El borde es una línea que rodea el padding y el contenido. Puede tener un grosor, un estilo y un color específicos definidos en CSS.
4. **Margen (Margin):** El margen es un espacio transparente que rodea el borde de la caja. Se utiliza para controlar la distancia entre esta caja y otros elementos circundantes en la página.



## Propiedad display

La propiedad display en CSS nos ayuda a controlar dónde se va a ver un elemento HTML dentro de la pantalla para estructurar nuestra página web. Esta propiedad se basa en la lógica del modelo de caja en CSS. En resumen, el modelo de caja en CSS dice que cada elemento HTML es realmente un bloque con una serie de características de borde, margen y padding. Cómo se ven estas características en nuestra página web dependerá de la opción de la propiedad display en CSS que elijamos para el elemento.

1. **block**: Los elementos con `display: block` se representan como bloques que ocupan todo el ancho disponible y comienzan en una nueva línea. Ejemplos de elementos bloque son `<div>`, `<p>`, y `<h1>`.
2. **inline**: Los elementos con `display: inline` se representan en línea con el contenido circundante y solo ocupan el ancho necesario. Ejemplos de elementos en línea son `<span>`, `<a>`, y `<strong>`.
3. **inline-block**: Los elementos con `display: inline-block` se comportan como elementos en línea, pero pueden tener dimensiones definidas y márgenes como los elementos de bloque. Esto permite que se muestren en línea pero tengan cierto control de diseño.
4. **none**: Los elementos con `display: none` no se muestran en la página y se consideran ocultos. Son útiles para controlar la visibilidad de elementos mediante JavaScript o CSS.
5. **flex**: Los elementos con `display: flex` se utilizan para crear diseños flexibles y pueden alinear y distribuir elementos dentro de un contenedor de manera flexible.
6. **grid**: Los elementos con `display: grid` se utilizan para crear diseños de cuadrícula, donde se pueden definir filas y columnas para organizar elementos.
7. **table**: Los elementos con `display: table` se tratan como elementos de tabla y pueden contener filas (`table-row`) y celdas (`table-cell`).
8. **table-cell**: Los elementos con `display: table-cell` se utilizan para representar celdas de una tabla.
9. **table-row**: Los elementos con `display: table-row` se utilizan para representar filas de una tabla.
10. **table-column**: Los elementos con `display: table-column` se utilizan para representar columnas de una tabla.
11. **table-row-group**: Los elementos con `display: table-row-group` se utilizan para agrupar filas de una tabla.
12. **list-item**: Los elementos con `display: list-item` se comportan como elementos de lista, como los elementos `<li>` en listas ordenadas o no ordenadas.
13. **initial**: La propiedad `display` se establece en su valor inicial predeterminado definido por el navegador.
14. **inherit**: La propiedad `display` hereda el valor de `display` de su elemento contenedor.
15. **unset**: La propiedad `display` se restablece a su valor inicial si está configurado, de lo contrario, se comporta como `inherit`.