

Programación Web Desde Cero

Modelo de cajas

¡Hola!

El **Box Model** (“modelo de caja”) es uno de los conceptos más importantes en CSS, y es fundamental para comprender cómo se estructuran y diseñan los elementos en una página web.

Describe cómo se componen los elementos HTML en una página.

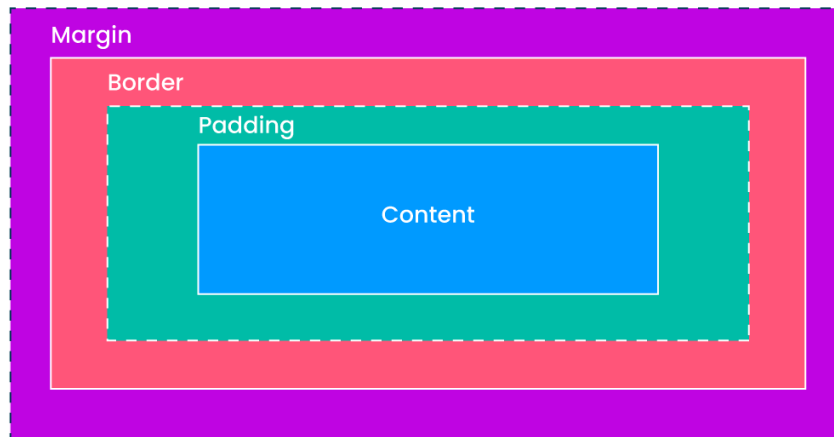
¡Comencemos! 

Box Model

Para CSS cada elemento es considerado como una "caja" que consta de cuatro partes:

- Content
- Padding
- Border
- Margin

Box Model CSS



Veamos en más detalle de qué trata cada parte:

Content → Es el área donde se muestra el contenido real del elemento (como el texto o las imágenes).

Padding → La propiedad de padding ("relleno") se usa para definir el espacio entre el contenido de un elemento y su borde. Los valores que se pueden utilizar incluyen:

- Un valor numérico, que define el espacio en píxeles.
- Un valor porcentual, que define el espacio en relación al ancho del elemento.
- La palabra clave "auto", que centra el contenido horizontalmente.

Border → La propiedad de border se usa para definir el borde de un elemento, es decir, es la línea que rodea el contenido y el padding. Los valores que se pueden utilizar incluyen:

- El ancho del borde en píxeles, como "1px".
- El tipo de borde, como "solid" para un borde sólido o "dashed" para un borde discontinuo.
- El color del borde, como "black" para un borde negro o "#cccccc" para un color hexadecimal.

👉 Puedes ver más efectos de borde en el siguiente link [Borde CSS](#)

Margin → La propiedad de margin ("margen") se usa para definir el espacio entre el borde y el elemento adyacente (como otro elemento o el borde del navegador). Los valores que se pueden utilizar incluyen:

- Un valor numérico, que define el espacio en píxeles.
- Un valor porcentual, que define el espacio en relación al ancho del elemento padre.
- La palabra clave "auto", que centra el elemento horizontalmente.

Veamos un ejemplo de cómo se aplica el box model a un elemento HTML simple:

```
<div class="ejemplo">
  | Este es un ejemplo de contenido
</div>
```

En este caso, el elemento es un "div" con la clase "ejemplo". El contenido es simplemente el texto: "Este es un ejemplo de contenido".

El *padding*, *border* y *margin* no se han definido explícitamente, por lo que serían 0.

💡 De todas maneras, el navegador por defecto agrega estilos. Por lo que, si quisiéramos resetearlos, vamos a poder hacerlo con el siguiente código CSS:

```
* {
  margin: 0px;
  padding: 0px;
}
```

Ahora, si aplicamos algunos estilos CSS para agregar padding, border y margin, se podría declarar así:

```
.ejemplo {  
  padding: 10px;  
  border: 1px solid black;  
  margin: 20px;  
}
```

💡 Como podemos observar en la imagen, **la propiedad de border tiene asignados varios valores en una misma línea** (el ancho, tipo y color). Se escribe de esa manera para simplificar el código y que sea más legible y eficiente.

Y el resultado final se vería así:



Como se puede ver, el contenido está rodeado por un espacio de 10px de padding, luego por un borde negro sólido de 1px, y finalmente por un margen de 20px. Todo esto compone la "caja" del elemento.

Ahora bien, si quisiéramos darle diferentes valores a los espacios de cada uno de los cuatro lados del elemento, las propiedades de padding y margin se pueden declarar junto con la palabra: top, right, bottom o left, para especificar la cantidad de espacio que se desea agregar en una dirección específica. Por ejemplo:

Para agregar 200px de margen en la parte superior y a la parte derecha de un elemento, pero no en la parte izquierda y en la parte inferior, podríamos utilizar la siguiente regla CSS:

```
.ejemplo {  
  padding: 10px;  
  border: 1px solid black;  
  margin-top: 200px;  
  margin-right: 200px;  
}
```

Y el resultado final se vería así:



Este es un ejemplo de contenido

Otra forma de declarar los margins que vimos anteriormente sería la siguiente:

```
.ejemplo {  
  padding: 10px;  
  border: 1px solid black;  
  margin: 200px 200px 0px 0px;  
}
```

💡 Cuando el `margin` se declara de manera abreviada se divide en cuatro valores separados por espacios. El primer valor corresponde al margen superior (*top*), el segundo al margen derecho (*right*), el tercero al margen inferior (*bottom*) y el cuarto al margen izquierdo (*left*).

De esta manera, al asignar `"200px 200px 0px 0px"` se está indicando un margen superior y derecho de 200px y un margen inferior e izquierdo de 0px, simplificando la escritura del código y haciendo que sea más fácil de leer. Y el resultado que obtenemos será el mismo:



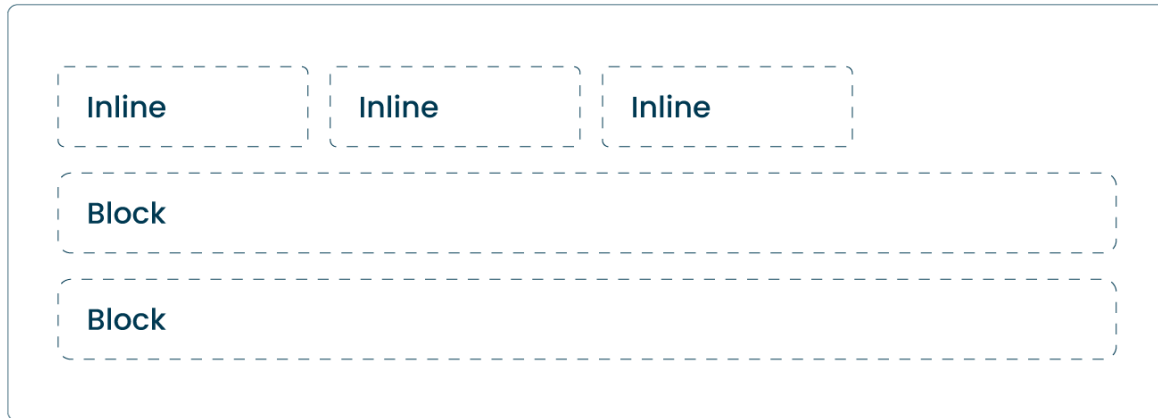
Este es un ejemplo de contenido

Elementos en bloque vs. elementos en línea

En HTML existen elementos en **bloque (block)** y elementos en **línea (inline)**. Los **elementos en bloque ocupan el total del ancho que se establece** (esto significa que los elementos en bloque comienzan en una nueva línea y empujan a otros elementos hacia abajo). Mientras que **los elementos en línea ocupan solo el**

espacio necesario dentro de una línea de texto, permitiendo que otros elementos se sitúen junto a ellos en la misma línea.

Veamos un ejemplo:



Algunos ejemplos de elementos en bloque son:

- `<div>`: Es un contenedor genérico que se utiliza para agrupar otros elementos y aplicarles estilos CSS o realizar manipulaciones con JavaScript.
- `<p>`: Representa un párrafo de texto.
- `<h1>`, `<h2>`, ..., `<h6>`: Son los encabezados, siendo `<h1>` el de mayor jerarquía y `<h6>` el de menor.
- ``, ``, ``: Se utilizan para crear listas, ya sean desordenadas (``) o ordenadas (``), y cada elemento de la lista se representa con ``.
- `<table>`, `<tr>`, `<td>`, `<th>`: Se utilizan para crear tablas.

Algunos ejemplos de elementos en línea son:

- ``: Similar a `<div>`, pero es un contenedor en línea. Se utiliza para aplicar estilos o JavaScript a una porción de texto.
- `<a>`: Define un enlace (hipervínculo) a otra página o recurso.
- ``: Se utiliza para incrustar imágenes.

- `` y ``: Se utilizan para dar énfasis al texto, haciéndolo negrita. Aunque visualmente son similares, `` tiene un significado semántico de mayor importancia o seriedad.
 - `` y `<i>`: Se utilizan para dar énfasis al texto, haciéndolo cursiva. Al igual que con `` y ``, `` tiene un significado semántico de énfasis, mientras que `<i>` es simplemente una representación visual.
-

Resumen

¡Felicidades!

Esperamos que esta información te haya sido de utilidad y que te lleves para repasar:

- Qué es el modelo de cajas y por qué es importante aprenderlo.
- Qué son los elementos en bloque y en línea en HTML