

# Propiedades para cajas

Para el CSS, todas las etiquetas html son consideradas como cajas. Tanto una etiqueta em como una h2, o una img, son rectangulares (o cuadradas dependiendo de su contenido). Si las etiquetas son EN LÍNEA, serán una caja AL LADO DE LA OTRA y, si son de bloque, UNA DEBAJO DE LA OTRA.

De esta manera “maquetar” se tratará de hacer “encajar” las etiquetas de una determinada manera, como si fuera un rompecabezas o un Tetris. Ese concepto de que “todo es una caja” da lugar al BOX MODEL o MODELO DE CAJA CSS. No importa si se trata de una etiqueta en línea o una de bloque, TODAS presentan las siguientes propiedades...

## Box Model – Propiedades

Suponiendo que tenemos una etiqueta que contiene texto (contenido de la caja), sin importar si es en línea o de bloque (puede ser un span o un párrafo, da igual), vemos el texto sobre fondo blanco y no nos da mucha idea de que realmente es una caja. Sin embargo, veremos realmente es una caja rectangular que asume valores para dos propiedades del Box Model: width (ancho) y height (alto).

Contenido de la caja

**width:** espacio declarado para el ancho del contenido de la caja.

**height:** espacio declarado para el alto del contenido de la caja.

Por fuera del espacio ocupado por estas propiedades se encontrará el contorno o borde que establece el límite de la caja, representado por la propiedad border.



**border:** Tipo de línea que envolverá la caja. Por defecto esta propiedad no tiene un valor visible aplicado, pero TODAS las etiquetas cuentan con él.

En principio el borde queda “pegado” al contenido, es decir al width y height de la caja, pero esto, habitualmente y sobre todo si el contenido es texto, no es una situación deseable ya que un texto a corte de un borde (o de la ventana del navegador) hace que se genere tensión en su lectura por parte del usuario. Por lo tanto, CSS nos ofrece una propiedad que “juega” entre el contenido de la etiqueta (width y height) y su borde, estableciendo una separación interior, también llamado relleno o margen interno (aunque no hay que confundirla con la siguiente propiedad).

Se trata del padding.



**padding:** separación entre el borde y el contenido de la caja (width y height). Aclaramos en este punto que, por razones pedagógicas, se está incluyendo cada propiedad con un color diferente para evidenciar el espacio que ocupa dentro del modelo de caja.

Luego veremos que, al darle un color de fondo a la caja, se aplicará tanto al espacio del contenido (width y height) como al del padding (el border sí, tendrá la posibilidad de manejar su color independiente).

Como ahora estas propiedades pueden hacer “visible” a las cajas, en el sentido de que veremos que podemos aplicarle colores tanto en los fondos como en el borde y evidenciar desde dónde hasta dónde llega cada una, nos surgirá la necesidad de no mostrar todas las cajas “pegoteadas” unas contra otras. En muchos casos necesitaremos “que el aire pase” entre ellas para no

empastarlas visualmente. Para eso tenemos la última propiedad QUE FORMA PARTE de la caja que se denomina margin.



**margin:** es el margen externo, la separación entre el borde y el “afuera” de la caja, es decir que separa la caja de los demás elementos (o cajas) que la circundan. Una característica de esta propiedad es que SIEMPRE es TRANSPARENTE, si bien forma parte de la caja, como su misión es separarla de los elementos que tiene a su alrededor, no se le puede aplicar color ya que queremos que “se note” que es una separación, “que el aire pasa”.

Por lo tanto, en el espacio que ocupa el margen veremos el color de fondo del CONTENEDOR del elemento ya que al ser transparente nos permite ver el color que está por detrás de la caja. Supongamos que tenemos un párrafo con color de fondo rojo contenido directamente por el body con color de fondo blanco, el espacio que ocupa el margen se verá de color blanco.

Algo MUY IMPORTANTE para tener en cuenta es que LAS CINCO PROPIEDADES SON NUMÉRICAS y NECESITAN UNA UNIDAD DE MEDIDA que acompañe a ese valor numérico, ya que estamos determinando el espacio que ocupa cada una. Como se dijo anteriormente, si a alguna de estas propiedades que necesitan una unidad de medida, no se la incluye, NO SE APLICA.

A medida que agreguemos valores para estas propiedades, la caja ocupará más espacio dentro de la página. Es lógico pensar que, si una caja no tiene valores para el padding o el borde por defecto y nosotros se lo aplicamos desde CSS, esa caja ocupará un mayor espacio que antes.

Entonces, podemos concluir que TODAS estas propiedades SON ADITIVAS en cuanto al espacio que va a terminar ocupando una caja (se suman).

Esto determina dos conceptos muy importantes que tenemos que tener en cuenta a la hora de maquetar para poder controlar perfectamente la distribución de los elementos en la página:

## Espacio VISIBLE que ocupa una caja

Como se dijo anteriormente, el margen de una caja es transparente y siempre se verá el espacio que ocupa con el color de fondo de su contenedor.

Por lo tanto, podemos definir el espacio VISIBLE como el que está ocupado por las propiedades a las que le podemos ajustar un color y que, de esta manera, quede perfectamente evidenciado el espacio que ocupan.

Entonces, el espacio VISIBLE que ocupa la caja resultará de la SUMA de las primeras CUATRO PROPIEDADES del box model. De este modo, el ancho visible de la caja será la suma del ancho del borde (izquierdo y derecho), el padding (izquierdo y derecho) y el valor de width.

El alto visible de la caja será la suma del ancho del borde (superior e inferior), el padding (superior e inferior) y el valor de height.

## Espacio TOTAL que ocupa una caja

Cuando hablamos del espacio TOTAL que ocupa la caja, nos referimos exactamente a eso, TODO el espacio en el cual NO SE PUEDE UBICAR otro elemento, ni en línea ni de bloque, ya que está ocupado por esta caja en cuestión.

Entonces tenemos que recordar que, si bien el margen es transparente, FORMA PARTE DE LA CAJA, es una propiedad más de la misma y la que hace que se "separe" del resto de los elementos, ya que ESE espacio le pertenece.

El margen NO ES APARTE de la caja. En consecuencia, el espacio TOTAL que ocupa una caja resultará de la SUMA DE LAS CINCO PROPIEDADES que la afectan.

Así, el ancho TOTAL que ocupa una caja, será la suma del margen (izquierdo y derecho), el ancho del borde (izquierdo y derecho), el padding (izquierdo y derecho) y el valor de width.

Y el alto TOTAL que ocupa, será la suma del margen (superior e inferior), el ancho del borde (superior e inferior), el padding (superior e inferior) y el valor de height.

Pongamos esto en números...

```
1 div{  
2   width: 200px;  
3   height: 300px;  
4   padding: 50px 20px;  
5   border: 2px solid black;  
6   margin: 30px;  
7 }
```

En este ejemplo tenemos una regla que afecta a los divs de un documento. Vamos a calcular cada uno de los conceptos descriptos:

**Ancho Visible:** width + padding (izquierdo y derecho) + border (izquierdo y derecho) = 200px + 20px + 20px (el segundo valor de la propiedad padding que afecta a la izquierda y a la derecha, se suma dos veces justamente por esta situación) + 2px + 2px (se suma dos veces por estar aplicado a la izquierda y a la derecha de la caja) = **244px**

**Ancho Total:** ancho visible + margin (izquierdo y derecho) = 244px + 30px + 30px = **304px**

**Alto Visible:** height + padding (top y bottom) + border (top y bottom) = 300px + 50px + 50px (el primer valor de la propiedad padding que se aplica arriba y abajo) + 2px + 2px = **404px**

**Alto Total:** alto visible + margin (top y bottom) = 404px + 30px + 30px = **464px**

## Formato de las cajas

Como podemos ver en el esquema del box model, el width se aplica solo en el eje horizontal (de izquierda a derecha) y el height solo en el vertical (de arriba a abajo).

Pero el resto de las propiedades PUEDEN estar presente en los CUATRO COSTADOS de la caja. Así podemos tener margen, borde y padding tanto arriba, como abajo, como a la izquierda o a la derecha de la caja.

Esto hace que las propiedades margin, border y padding ACPTEN 1, 2 o 4 VALORES separados por espacios. Si recibe:

- **1 solo valor:** se aplica el mismo valor a los 4 costados de la caja
- **2 valores:** el primero se aplica en el top y bottom (arriba y abajo) y el segundo en right y left (derecha e izquierda)
- **4 valores:** empezando por el top se aplican a cada lado de la caja en el sentido de las agujas del reloj: top, right, bottom y left (empezando por el 12 del reloj la aguja se mueve y recorre los lados en el sentido: 12 – 3 – 6 – 9)

NO ACEPTA 3 VALORES, son 1, 2 o 4 con las características descritas. A estas propiedades que pueden aceptar más de un valor las denominamos propiedades SHORTHAND o acortadas. En realidad, se denominan así porque nos permiten aplicarles valores a varias propiedades a la vez.

Esto es porque existen versiones de estas propiedades para cada costado de la caja a la que afectan. Esas propiedades llevan el nombre de la propiedad más un guion medio y el cotado de la caja a la que afectan, por ejemplo: margin-top, margin-bottom, margin-right, margin-left. Lo mismo para padding y border.

Estas propiedades AFECTA SOLO un costado de la caja y, por lo tanto, RECIBEN UN SOLO VALOR, que es el que se le aplica a ese costado.

Entonces, podemos afectar los cuatro costados de una caja con valores iguales o diferentes para cualquiera de estas propiedades, utilizando las versiones individuales para cada lado o la propiedad shorthand que afecta a los cuatro lados, DA LO MISMO.

```
1 p{  
2   padding: 20px 5px;  
3 }
```

Se le aplica un padding de 20px arriba y abajo (primer valor) y de 5px a la derecha e izquierda (segundo valor) a los párrafos del documento (seleccionados por una regla de etiqueta).

Esto sería lo mismo que haberlo aplicado con las propiedades individuales por lado de la caja:

```
1 p{  
2   padding-top: 20px;  
3   padding-right: 5px;  
4   padding-bottom: 20px;  
5   padding-left: 5px;  
6 }
```

Asignamos los mismos valores, pero con las propiedades individuales, por supuesto que da lo mismo el orden en el cual se aplica cada propiedad individual (cuál escribo primero y cuál después) ya que, en este caso, afectan cada una un lado distinto de la caja.

También sucede lo mismo con las propiedades margin y border, aunque para el borde vamos a ver con detalle las propiedades que lo afectan.

## Propiedad border

Como se dijo, TODAS las etiquetas html aceptan un borde. Pero esta es una propiedad especial ya que, si queremos "dibujar" un borde a cualquier elemento, tendremos varios aspectos a considerar: el ancho de la línea (puede ser delgada o gruesa), su estilo (puede ser punteada o continua) y su color. En CSS sucede lo mismo.

En realidad, la propiedad border está formada por la conjunción de tres propiedades que cubren esos aspectos:

- **border-width:** determina el grosor de la línea del borde. Recibe un valor numérico acompañado de su respectiva unidad de medida.
- **border-style:** determina el tipo o estilo de línea del borde. Hay varios valores posibles:
  - solid: línea continua.
  - dashed: línea a rayas separadas por espacios.
  - dotted: línea punteada.
  - double: doble línea.
  - inset: define un borde hundido en 3D, el efecto depende del color del borde.
  - outset: define un borde salido o en relieve en 3D, el efecto depende del color del borde.

- ridge: define un borde estriado en 3D, el efecto depende del color del borde.
- groove: define un borde acanalado en 3D, el efecto depende del color del borde.
- none: sin borde (por default).
- hidden: borde oculto (igual que none excepto en bordes en conflicto para celdas de una tabla).
- **border-color:** color de la línea del borde. Como cualquier otra propiedad de color, recibe su valor a partir del nombre en inglés, la función rgb() o el sistema hexadecimal.

En el ejemplo se realiza una regla de etiqueta que afecta a todos los párrafos y luego se le aplica a cada uno una clase con los distintos tipos de estilos de borde (como las propiedades que manejan las reglas que afectan a una misma etiqueta son distintas, ya que con una aplicamos el ancho de la línea y el color y con la otra el estilo, se aplican todas por el concepto de cascada CSS).

```

1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <meta charset="UTF-8">
5      <title>Propiedades para Cajas</title>
6      <!-- Linkeamos el CSS externo -->
7      <link rel="stylesheet" href="estilos.css">
8  </head>
9  <body>
10     <p class="solido">Soy un párrafo</p>
11     <p class="rayado">Soy un párrafo</p>
12     <p class="punteado">Soy un párrafo</p>
13     <p class="doble">Soy un párrafo</p>
14     <p class="hundido">Soy un párrafo</p>
15     <p class="salido">Soy un párrafo</p>
16     <p class="estriado">Soy un párrafo</p>
17     <p class="acanalado">Soy un párrafo</p>
18     <p class="ninguno">Soy un párrafo</p>
19     <p class="oculto">Soy un párrafo</p>
20 </body>
21 </html>

```

```

1  p{
2      border-width: 10px;
3      border-color: lightgreen;
4  }
5  .solido{border-style: solid;}
6  .rayado{border-style: dashed;}
7  .punteado{border-style: dotted;}
8  .doble{border-style: double;}
9  .hundido{border-style: inset;}
10 .salido{border-style: outset;}
11 .estriado{border-style: ridge;}
12 .acanalado{border-style: groove;}
13 .ninguno{border-style: none;}
14 .oculto{border-style: hidden;}

```





Estas tres propiedades aceptan 1, 2 o 4 valores ya que definen el ancho, estilo y color de los 4 bordes de la caja. Si bien habitualmente se define un borde homogéneo su alrededor, es posible aplicar un borde completamente diferente en cada lado de la caja.

También existe una propiedad por lado, con todas las combinaciones posibles de propiedades (ancho, color y estilo) y costados de la caja (top, bottom, right y left). En estos casos el lado de la caja se ubica en el medio del nombre de la propiedad rodeado por guiones medio:

Por ejemplo: border-top-width, border-top-style, border-top-color, border-right-width... Cada una de estas propiedades ACEPTAN UN ÚNICO VALOR ya que son individuales.

Pero aún tenemos más propiedades shorthand que consideran el lado de la caja al que se le aplica: border-top, border-bottom, border-right y border-left. Estas propiedades, al igual que la MADRE DE TODAS: border, aceptan los tres valores necesarios para generar un borde: ancho, estilo y color, separados por un espacio:

```

9 <body>
10   <p class="todos">Soy un párrafo</p>
11   <p class="superior">Soy un párrafo</p>
12   <p class="derecho">Soy un párrafo</p>
13   <p class="inferior">Soy un párrafo</p>
14   <p class="izquierdo">Soy un párrafo</p>
15   <p class="individuales">Soy un párrafo</p>
16 </body>
17 </html>

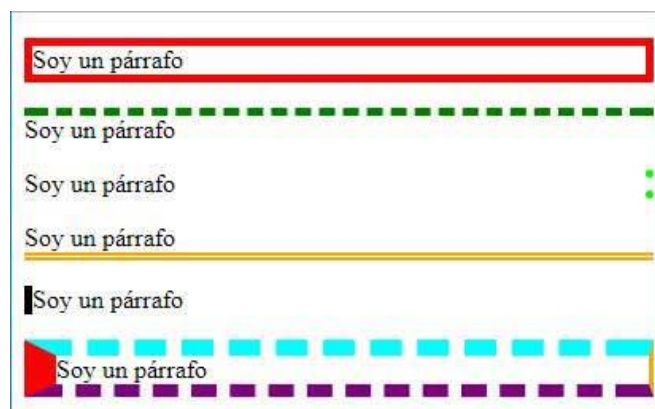
```

```

1 .todos{ border: 5px solid #ff0000; }
2 .superior{ border-top: 5px dashed green; }
3 .derecho{ border-right: 5px dotted rgb(0, 255, 0); }
4 .inferior{ border-bottom: 5px double orange; }
5 .izquierdo{ border-left: 5px solid black; }
6 .individuales{
7   border-top-color: aqua;
8   border-top-width: 10px;
9   border-top-style: dashed;
10  border-right-color: orange;
11  border-right-width: 3px;
12  border-right-style: solid;
13  border-bottom-color: purple;
14  border-bottom-width: 8px;
15  border-bottom-style: dashed;
16  border-left-color: red;
17  border-left-width: 20px;
18  border-left-style: solid;
19 }
20

```

Al primer párrafo le asignamos la clase que le aplica un borde con la propiedad shorthand border, del segundo al quinto párrafo fueron afectados por clases que manejan la propiedad shorthand por lado y al último, le aplicamos propiedades individuales para cada borde.



# Selector Universal

Antes de seguir avanzando con las propiedades para cajas, vamos a detenernos un momento a analizar la siguiente situación. Maquetaremos un simple párrafo al que le aplicaremos un borde con la propiedad shorthand para observar el resultado obtenido...


```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <title>Propiedades para Cajas</title>
6   <!-- Linkeamos el CSS externo -->
7   <link rel="stylesheet" href="estilos.css">
8 </head>
9 <body>
10  <p>Contenido del párrafo.</p>
11 </body>
12 </html>
```

```
1 p{
2   border: 2px solid red;
3 }
4
```



Ahora analicemos... El ancho del párrafo asume el 100% del ancho de su contenedor, ya que es una caja de bloque y no le ajustamos la propiedad width, el alto es el del contenido también por ser una caja de bloque, crece automáticamente hasta donde llega su contenido, por no haberle aplicado ningún valor a la propiedad height.

Tampoco le aplicamos padding, por eso vemos que el borde, que es la única propiedad que asignamos, se muestra "pegado" a su contenido (el texto). Pero, si observamos bien, veremos un espacio entre el borde de la caja y los bordes de la ventana del navegador, top, left y right ya que al bottom no llega por tener poco contenido (en la siguiente imagen ese espacio se muestra coloreado de rosa).



De acuerdo con el modelo de caja CSS que vimos anteriormente, ese espacio corresponde al MARGEN de la caja ya que es una separación con el “afuera” de la misma. El razonamiento es perfecto, ahora... ¿dónde o cuándo le dijimos al párrafo que debía tener un margen?

```
1 p{  
2   border: 2px solid red;  
3 }  
4
```

¡¡Solo le dimos borde!! Entonces... ¿de dónde salió ese valor de margen? La explicación es simple. Así como el navegador les aplica un determinado formato a ciertas etiquetas, como los encabezados, los refuerzos semánticos o los vínculos, también les aplica a muchas de ellas, valores de margen y/o de padding para que el contenido no se presente “todo pegoteado” por defecto.

Si recordamos las primeras clases habíamos visto que, si ponemos varios párrafos consecutivos, el espacio entre ellos es mayor que el interlineado que presenta cada uno, es decir, que les está aplicando un margen inferior o superior para separarlos.

Varias veces hemos dicho que no nos importa el pre formato que le aplique el navegador a las etiquetas, ya que desde CSS lo podríamos modificar y este caso NO ES LA EXCEPCIÓN.

Mucho menos nos interesa estos espacios que agrega entre etiquetas, sobre todo pensando en las maquetaciones que vamos a encarar, ya que no sabemos cuánto es ese valor, ni si todos los navegadores aplican el mismo valor de separación.

Pensando en que en algún momento tendremos que maquetar dos columnas, por ejemplo, debemos tener perfectamente controlado el espacio

que ocupe cada una de ellas, ya que cualquier agregado "desconocido" puede hacer que ya no entren una al lado de la otra y "caiga" para abajo.

Para evitar esta situación indeseable, vamos a usar una regla CSS que resetee los valores por default de margin y padding que afecte a TODAS las etiquetas html. Resetear los valores simplemente significa asignarles valor cero a esas propiedades.

Si agregamos esa regla al comienzo del CSS, luego cualquier otra regla puede "pisar" ese valor para una etiqueta en particular a la que sí le queremos aplicar un padding o margen.

Para afectar a TODAS las etiquetas, vamos a hacer uso del selector universal (se llama así porque precisamente selecciona a todas las etiquetas html) que se escribe simplemente con un asterisco (\*).

Más adelante veremos algunas otras maneras más eficientes de resetear estilos por default, ya que esta forma solo se encarga de los valores de márgenes y paddings.

```
1  *{  
2    margin: 0;  
3    padding: 0;  
4  }  
5  p{  
6    border: 2px solid red;  
7  }  
8
```

Y el resultado es...



Ahora vemos lo que hubiéramos esperado desde el principio, la caja ocupa el 100% del ancho de su contenedor (la ventana del browser) por ser de bloque, sin dejar espacios a su alrededor.

Por lo tanto, a partir de este momento, TODOS nuestros trabajos comenzarán incluyendo esta regla con el selector universal para resetear márgenes y paddings, para poder tener perfecto control del tamaño de las cajas, sobre todo, como se dijo, de los anchos, ya que es el eje que más nos interesa al momento de maquetar nuestras páginas.