

# Modelo de caja

**DigitalHouse** >  
Coding School



**Certified Tech  
Developer**  
The Ultimate Degree

# Índice

1. [Modelo de caja](#)
2. [width y height](#)
3. [padding](#)
4. [Propiedades abreviadas](#)
5. [border](#)
6. [margin](#)
7. [box-sizing](#)

# 1 | Modelo de caja

“

En **HTML** todos los elementos se representan mediante **cajas**, eso se conoce como el **modelo de caja**. Cada caja se compone de contenido, relleno, bordes y márgenes.



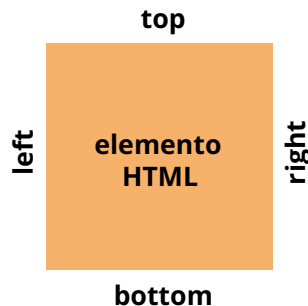
”

# Modelo de caja

Cada elemento en HTML es una caja, y esas cajas se componen de márgenes (margin), bordes (border), relleno (padding) y finalmente el contenido (content).

Mediante CSS podemos manipular todas estas propiedades para cambiar la apariencia de cada elemento. A su vez, estas propiedades pueden aplicarse de manera diferente a los cuatro lados de cada caja (top, right, bottom y left).

Muchas de las propiedades solo aplican a elementos de bloque o semibloque.



**2** | **width y height**

# Propiedad width

Si un elemento de bloque no tiene declarada la propiedad width, el ancho será igual al 100% de su padre contenedor. Podemos asignarle un valor a esta propiedad usando cualquier unidad de medida, como porcentajes (%) o píxeles (px).

css

```
div {  
    width: 120px;  
}
```

# Propiedad height

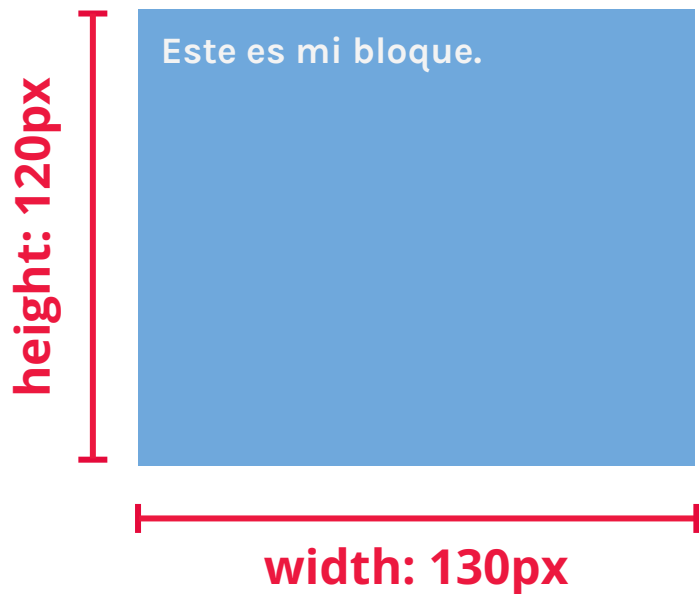
Si un elemento no tiene declarado la propiedad height, el alto será igual a la altura que le proporcione su contenido. Sea un elemento de bloque o de línea. Podemos asignarle un valor a esta propiedad usando cualquier unidad de medida, como pixeles (px). Para la altura de los elementos no se recomienda usar porcentaje.

css

```
div {  
    height: 130px;  
}
```



# width y height en un elemento



```
css

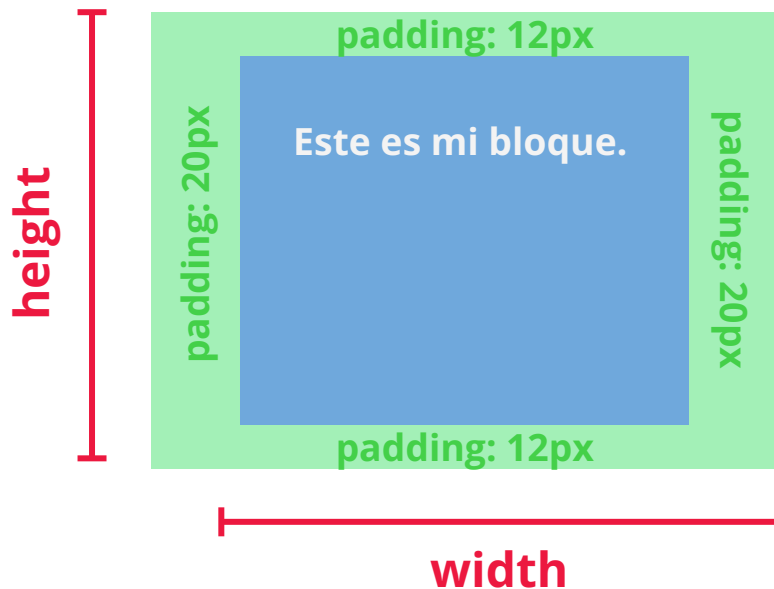
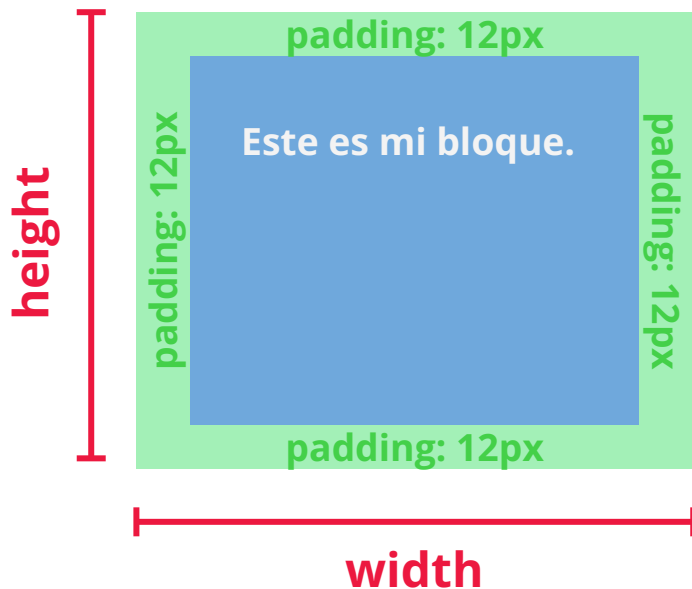
{width: 130px; height: 120px;}


```

# 3 | padding

# Propiedad padding

Es el espacio de **relleno** que podemos agregar **entre** el **contenido** del elemento y su **borde**.



Podemos asignarle un valor a esta propiedad usando cualquier unidad de medida, como píxeles (px), indicando **1 valor** para los 4 lados de la caja.

```
css
```

```
div { padding: 12px; }
```



### Único valor

Esto quiere decir que el relleno de **12px** se aplicará a todos los lados.

También podemos hacerlo con **2 valores**. El primero va a indicar el padding de **arriba y abajo**, y el segundo el de la **izquierda y la derecha**.

css

```
div { padding: 12px 20px; }
```



### Dos valores

**12px** de relleno para arriba y abajo.

**20px** de relleno para izquierda y derecha.

También podemos hacerlo con **3 valores**. El primero va a indicar el padding de **arriba**, el segundo el de la **izquierda y la derecha** y el tercero será el de **abajo**.

css

```
div { padding: 12px 20px 18px; }
```



### Tres valores

**12px** de relleno para arriba.

**20px** de relleno para izquierda y derecha.

**18px** de relleno para abajo.

Por último, podemos utilizar **4 valores** que representarán los cuatro costados individualmente. Empezaremos por el valor de arriba y seguiremos (en el sentido de las agujas del reloj) por derecha, abajo e izquierda.

css

```
div { padding: 12px 15px 18px 21px; }
```



### Cuatro valores

**12px** de relleno para arriba.

**15px** de relleno para la derecha.

**18px** de relleno para abajo.

**21px** de relleno para la izquierda.

# 4 | Propiedades abreviadas



# Propiedades abreviadas

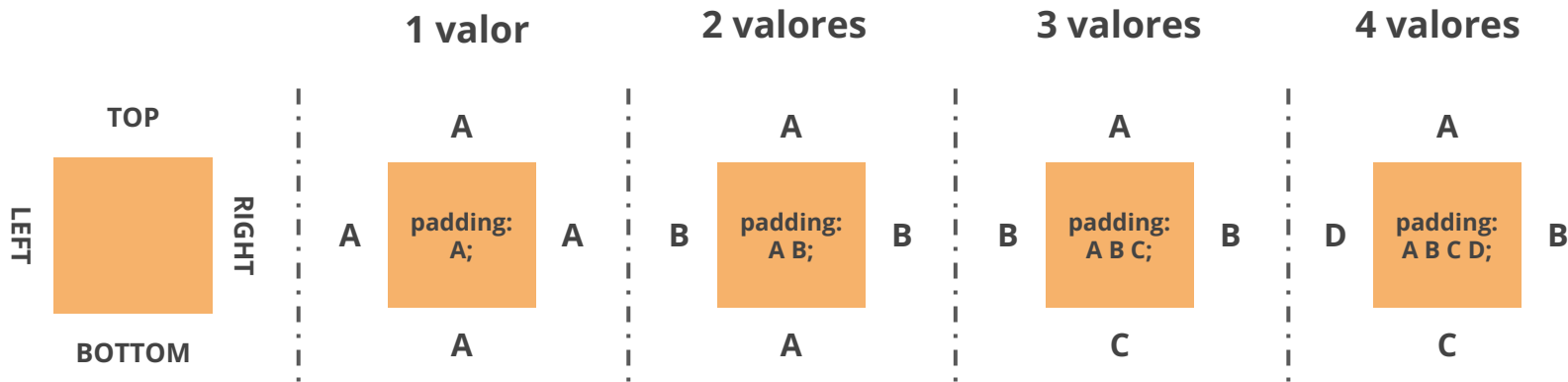
Como dijimos antes, muchas de las propiedades que vamos a ver se pueden aplicar de manera específica a uno de los costados de la caja. Cuando escribimos un solo valor para padding, por ejemplo:

```
css  div { padding: 15px }
```

Lo que en realidad está pasando, sin que nos demos cuenta, es esto:

```
css  div {  
      padding-top: 15px; padding-right: 15px;  
      padding-bottom: 15px; padding-left: 15px;  
  }
```

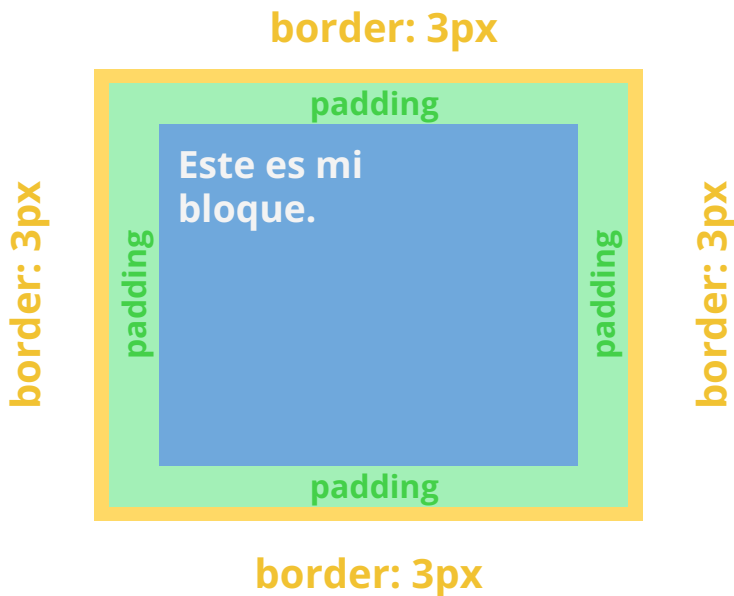
En todas las propiedades de este estilo, como margin y padding, podemos escribir 1, 2, 3 o 4 valores, y según cuántos escribamos será a qué lados de la caja se apliquen los valores.



# 5 | border

# Propiedad border

Hace referencia al borde del elemento. Se ubica entre el contenido y el margen.



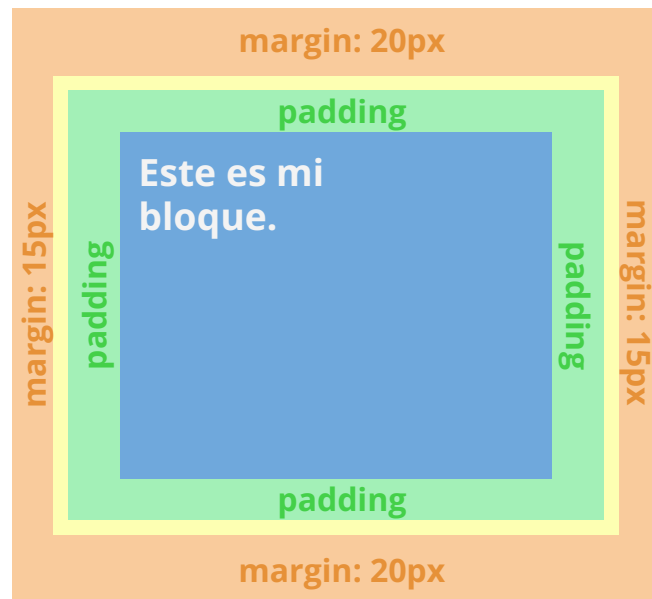
Podemos asignarle un valor a esta propiedad definiendo el **estilo de línea**, su **espesor** y su **color**. El estilo de línea puede ser `solid`, `dotted`, `dashed` o `double`. El espesor de línea puede ser cualquier unidad de medida de CSS. El color puede ser cualquier color válido de CSS.



# 6 | margin

# Propiedad margin

Hace referencia al **margen exterior** del elemento. Sirve para separar una caja de la otra.



Podemos asignarle valor a esta propiedad usando cualquier unidad de medida, como los píxeles (px), indicando 1 valor para los 4 lados de la caja.

```
css
```

```
div { margin: 15px }
```



### Único valor

Esto quiere decir que el margen exterior de **15px** se aplicará a todos los lados: arriba, derecha, abajo e izquierda.



De la misma manera que con padding, podemos asignar también 2, 3 y hasta 4 valores para la propiedad.

```
css  div { margin: 20px 15px }
```

```
css  div { margin: 20px 15px 30px }
```

```
css  div { margin: 20px 15px 30px 25px }
```

# 7 | box-sizing

# Calculando el ancho y el alto

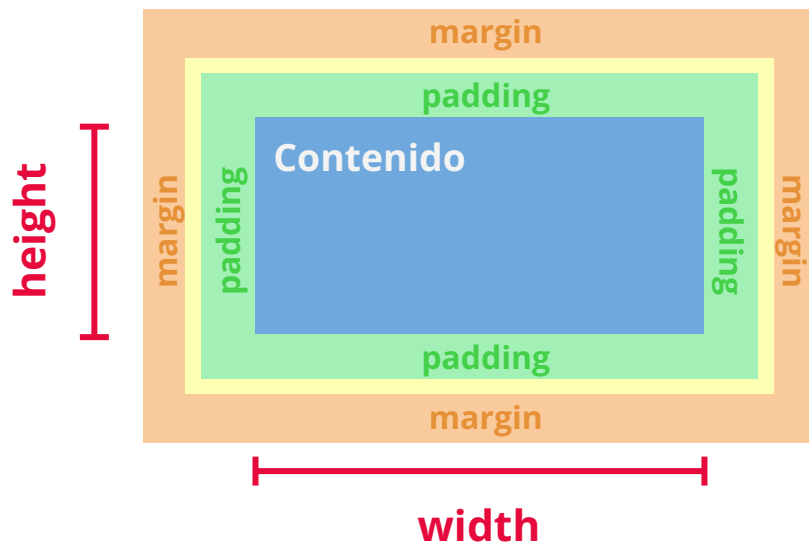
Para poder calcular el ancho o el alto total de un elemento, tenemos que sumar todas las propiedades que vimos antes.

Como se pueden imaginar, hacer este cálculo para saber cuánto va a ocupar un elemento finalmente es bastante trabajoso. La propiedad `box-sizing` nos ayuda con este cálculo.



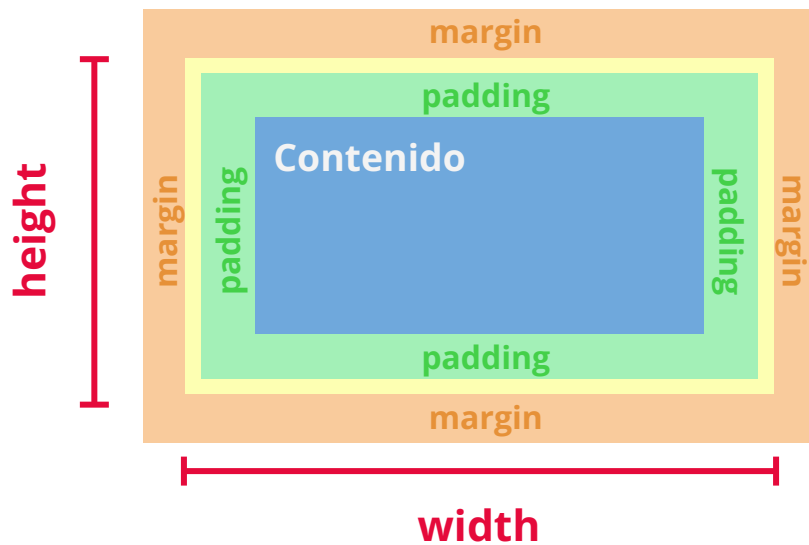
# box-sizing: content-box

Por defecto su valor será `content-box`, y el comportamiento será el de aplicarle el ancho y alto que definamos al contenido del elemento.



# box-sizing: border-box

Si le asignamos el valor `border-box`, el ancho y alto que indiquemos tomará en cuenta no solo el contenido del elemento, sino también el padding y el borde, dejando solo el margen por fuera.



# box-sizing: border-box



15px + 150px + 15px

**ancho total = 180px**

```
div {  
  width: 150px;  
  height: 130px;  
  padding: 12px;  
  border: solid 3px yellow;  
  margin: 15px;  
  box-sizing: border-box;  
}
```

“

Es una práctica muy común asignarle **box-sizing: border-box** a todos los elementos del sitio con la siguiente línea de código:

```
* { box-sizing: border-box }
```

”



DigitalHouse>  
Coding School