

Clase 05

Diseño y Programación Web

Materia: Programación Web I

Docentes contenidistas: GARCIA, Mabel y PEREZ, Jorge

Revisión: ROLDÁN, Hernán

Contenido

Box Model [Cajas]	3
Propiedades de caja.....	3
Dimensiones de las cajas (Width y Height)	4
Border	5
Bordes redondeados	7
Contorno (outline)	9
Diferencia entre border y outline	10
Sintaxis de Margin y Padding	10
Datos sobre el margin (centrar elementos de bloque)	12
Box Model (propiedad box-sizing).....	13
Color de fondo	15
Para ampliar la información	17

CLASE 5



¡Te damos la bienvenida a la clase 5 de la materia Programación Web I!

En esta clase vamos a ver los siguientes temas:

- Box Model (Cajas).
- Propiedades de Caja (margin, border, padding, content).
- Dimensiones de cajas (width, height).
- Propiedad border (style, width, color).
- Propiedad border-radius.
- Sintaxis de Margin y Padding.
- Diferencia entre border y Outline.
- Datos sobre el margin (centrar elementos de bloque).
- Box model (propiedad box-sizing).
- Color de fondo.

Box Model [Cajas]

El Box Model (Modelo de Caja) es un concepto en CSS que describe cómo se representa visualmente un elemento en una página web. En esencia, el Modelo de Caja define que cada elemento de HTML es visto como una caja rectangular que está compuesto por cuatro componentes, content, padding, border y margin (contenido, relleno, borde y margen).

Propiedades de caja

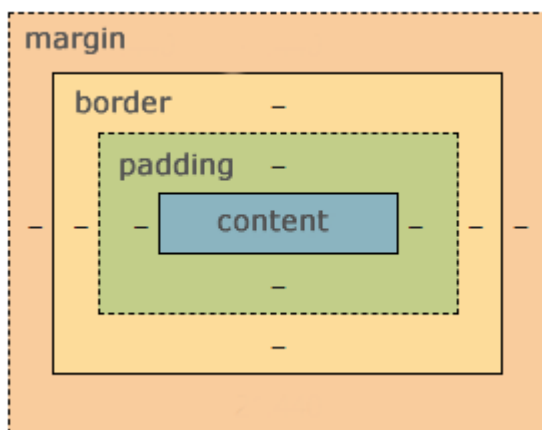
Todos los elementos del HTML son cajas y son rectangulares, más allá de su comportamiento por defecto.

En los elementos de **línea**, se verá uno al lado del otro.

En los elementos de **bloque**, uno debajo del otro.

Ese concepto de que "todo es una caja" da lugar a lo que en web llamamos **box model**.

Sin importar si son de línea o de bloque, todas las etiquetas tienen las siguientes propiedades en común:



margin:

Separación entre el borde y el afuera de la caja.

border:

Es el límite que separa el interior del exterior del elemento.

padding:

Es el relleno entre el borde y el contenido de la caja.

content:

Es el contenido "real" del elemento. Puede ser texto, una imagen, etc.

Cada una de estas propiedades del **modelo de cajas** de CSS se puede modificar, cambiando sus dimensiones, colores, etc. De esta forma, cada elemento HTML tendrá su propio borde, margen, padding y contenido.

Dimensiones de las cajas (Width y Height)

Para entender cuánto mide una caja, sin haberla modificado todavía desde CSS hay que saber a qué tipo de grupo pertenece la etiqueta:

Si es un elemento de **bloque**, su **ancho será el total del espacio disponible** y su **alto será el alto de su contenido**.

Si es un elemento de **línea**, su **ancho y alto será lo que posea su contenido**.

Texto del párrafo

Texto que posee 2 elementos de línea. **Soy un strong!** *Soy un em!*

Si queremos modificar sus medidas usaremos la propiedad **width** (ancho) y en caso de establecer un alto usaremos la propiedad **height**.

Hay que tener en cuenta que debemos estar seguros de la altura y del ancho de un elemento.



Por ejemplo, si digo que la altura de una caja es de 100px y su contenido es superior a esta medida, el contenido "saldrá" por fuera de la caja y se superpondrá con el contenido que venga abajo, en el orden de lectura.

Otra cuestión a tener presente es que **los elementos de línea no entienden ni width, ni height**. Por ahora, sólo podremos modificar elementos que sean de **bloque** (div, p, h1, h2, etc).

Border

El borde de las cajas se define por el uso de 3 propiedades:

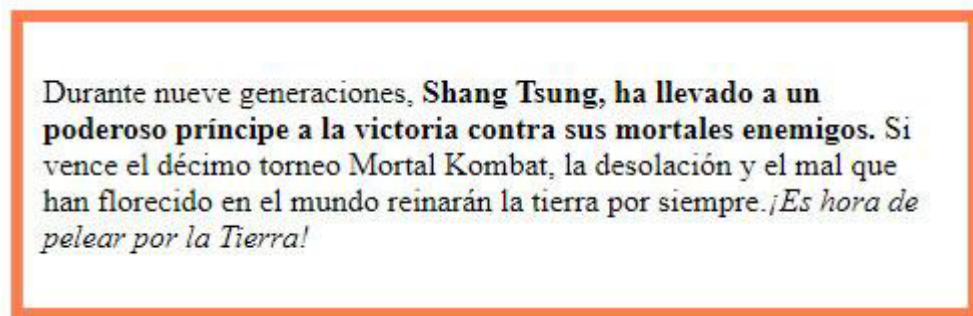
- **border-style:** Estilo de borde a aplicar – **obligatorio** para que sea visible. Hay varios tipos de valores: **dotted** (basado en puntos), **dashed** (basado en rayas -línea discontinua), **solid** (borde sólido - línea continua), **double** (borde doble -dos líneas continuas), **groove** (borde biselado con luz desde arriba), **ridge** (biselado con luz desde abajo), **inset** (borde con profundidad "hacia dentro"), **outset** (borde con profundidad "hacia fuera").
- **border-width:** Permite definir el ancho del borde de la caja. El valor debe ser un número y su unidad.
- **border-color:** Permite definir el color del borde (nombre, hexadecimal o rgb).

Estas propiedades permiten distintas combinaciones y definir estilos particulares para los lados de la caja, para el top, right, bottom y left.

Es decir, existe **border-top-style**, **border-right-style**, **border-bottom-style** y **border-left-style**. Las combinaciones para border-width: **border-top-width**, **border-right-width**, etc.) y **border-top-color**, **border-right-color**, etc.

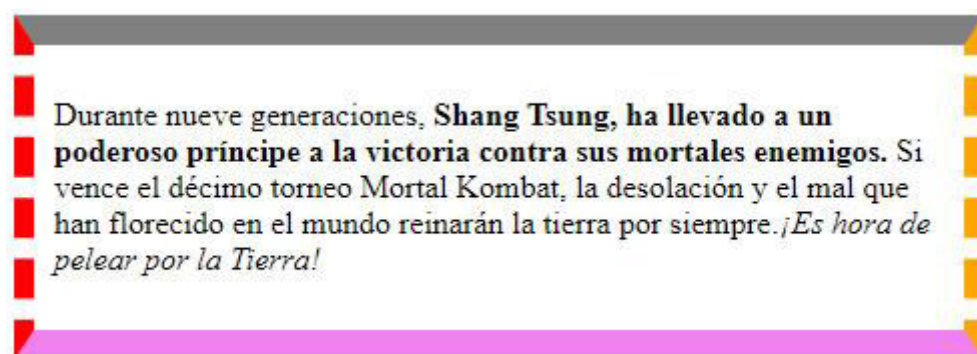
También existe el atajo para estas propiedades que se definen con la propiedad **border**, sus **valores pueden escribirse en cualquier orden y con espacios, y se aplican a los 4 lados de la caja (como mínimo requiere declarar el style)**.

```
div {border:solid 6px coral;}
```



También existen atajos para estas propiedades de bordes en zonas concretas: **border-top**, **border-right**, **border-bottom** y **border-left**

```
div {  
border-top: 15px solid grey;  
border-right: 10px dashed orange;  
border-bottom: 15px solid violet;  
border-left: 10px dashed red; }
```



Bordes redondeados

CSS permite redondear los bordes de las cajas (para que no todo se vea cuadrado en la web).

La idea es especificar un radio para el borde de las esquinas. Por defecto, este borde es de tamaño 0, es decir, no hay borde redondeado. A medida que se aumenta este valor, el borde se redondea más.

Una vez llegado a su máximo, no se apreciará ningún cambio.

Para lograr esto se usa la propiedad **border-radius**, que recibe valores numéricos y sus respectivas unidades.

Según la cantidad de valores, afectará diferentes esquinas.

- **1 valor:**
Asume que son los 4 lados por igual.
- **2 valores:**
El primero, esquinas arriba-izquierda + abajo-derecha.
El segundo, esquinas arriba-derecha + abajo-izquierda.
- **4 valores:**
Las 4 esquinas en sentido de las agujas del reloj.
Empieza a aplicar desde la esquina arriba-izquierda

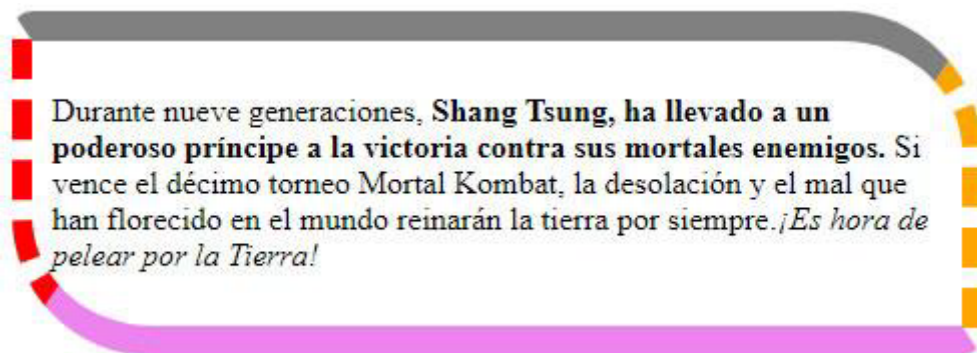
Cada valor puede ser mixto (dos números) **10px/20px**

- El primero es la **curvatura horizontal** de esa esquina.
- El segundo es la **curvatura vertical** de esa esquina.
- Esto sale del círculo perfecto, y permite hacer un óvalo.

Es posible especificar los valores de cada esquina mediante propiedades por separado:

- **border-top-left-radius:**
Afecta al radio de la esquina top-left.
- **border-top-right-radius:**
Afecta al radio de la esquina top-right.
- **border-bottom-left-radius:**
Afecta al radio de la esquina bottom-left.
- **border-bottom-right-radius:**
Afecta al radio de la esquina bottom-right.

```
div {  
border-top: 15px solid grey;  
border-right: 10px dashed orange;  
border-bottom: 15px solid violet;  
border-left: 10px dashed red;  
border-radius: 10px 70px;  
}
```



Contorno (outline)

Es una línea divisoria que rodea el contenido externo del propio elemento. A diferencia de los bordes, esta línea divisoria, no ocupa espacio y se ubica fuera de los límites del borde, para resaltar un elemento.

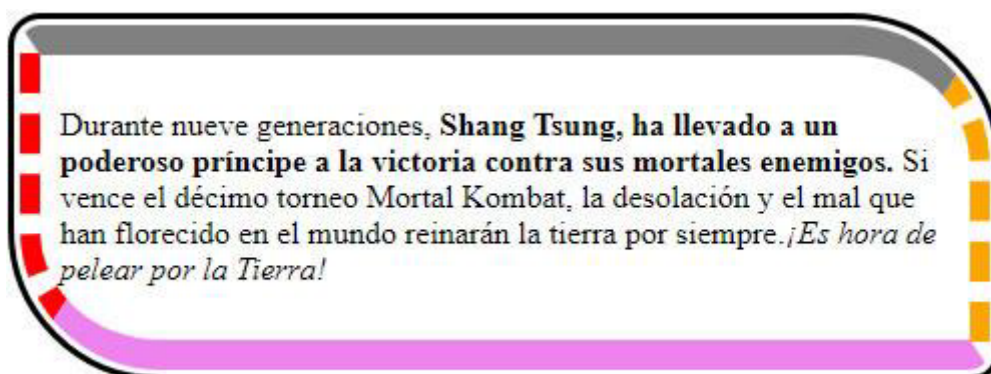
Tenemos distintas propiedades para manejar el contorno:

- **outline-style:** Su valor es como el de los border (**solid, dashed, etc**).
- **outline-color:** Su valor será un color (**hexadecimal, rgb, etc**).
- **outline-width:** Tamaño de los bordes, puede ser **predefinido o específico**. Valores: **thin** (aprox. 1px), **medium** (aprox. 3px) o **thick** (aprox. 5px).
- **outline-offset:** su valor será el espacio entre el contorno y el borde de un elemento.

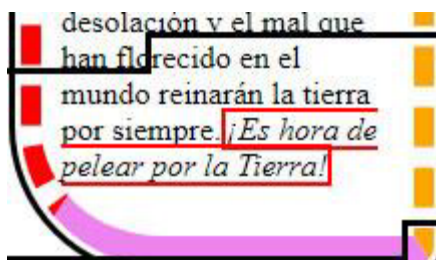
Como vimos con varias propiedades, también tenemos la posibilidad de **abreviar** la propiedad **outline-style, outline-color y outline-width**, usando propiedad **outline** que reúne sus valores:

```
div {  
border-top: 15px solid grey;  
border-right: 10px dashed orange;  
border-bottom: 15px solid violet;  
border-left: 10px dashed red;  
border-radius: 10px 70px;  
outline: black solid medium;  
outline-offset: 3px;  
}
```

Agregando el outline-offset a 3px de distancia podemos ver que se ubica por fuera del borde que ya tenía la caja.



Diferencia entre border y outline



Como dijimos antes, los **outline** **no ocupan espacio**, se dibujan por encima del elemento. Por lo tanto, su aparición u ocultamiento **no causa un redibujado de las cajas**. **No suma en el tamaño de la caja** y no hay movimiento o desplazamiento de los elementos como con **border**.

Los **outline** pueden **no ser rectangulares**.

Si el elemento se distribuye en varias líneas, el contorno no forma un rectángulo para encerrar a todo el elemento.

Sintaxis de Margin y Padding

Para manipular el **espacio externo de una caja (margin)**, como el **relleno interno (padding)**, CSS nos ofrece las propiedades individuales para afectar a cada lado de la misma.

En el caso de manipular **el margen**, tenemos las propiedades: **margin-top** (arriba), **margin-right** (derecha), **margin-bottom** (abajo) y **margin-left** (izquierda).

Si ese elemento fuese un elemento de tipo inline, no reconocerá los margin-top, ni los margin-bottom.

CSS también tiene una propiedad abreviada llamada **margin**, que reúne los valores para poder manipular los lados, dependiendo de la cantidad de valores que le pasemos a la propiedad.

- **1 solo valor:** Aplica el mismo valor a los 4 lados.
- **2 valores:** El primero afecta al top y bottom. El segundo al left y right.
- **3 valores:** El primero afecta al top. El segundo al right y left y el tercero como bottom.
- **4 valores:** Empezando desde el top, es cada lado en sentido de las agujas del reloj (top, right, bottom, left).

```
P{
margin:10px 15px 30px 50px;
}
```

Los 4 valores equivalen a las propiedades explícitas de la columna derecha, en cuanto al orden.

```
p {
margin-top:10px;
margin-right:15px;
margin-bottom:30px;
margin-left:50px;
}
```

En el caso del **padding**, CSS también nos da propiedades para afectar a cada uno de los lados de las cajas.

Tenemos las propiedades: **padding-top**, **padding-right**, **padding-bottom** y **padding-left**.

Si esa caja fuese un elemento de tipo inline, aplicará el padding, pero no reconocerá la altura o límite del elemento.

También nos ofrece la propiedad **padding** para abreviar la sintaxis y sus valores dependerán de la cantidad de lados que queramos afectar (Espera los mismos valores que el margin, 1-2-3 o 4 valores).

- **1 solo valor:** Aplica el mismo valor a los 4 lados.
- **2 valores:** El primero afecta al top y bottom. El segundo al left y right.
- **3 valores:** El primero afecta al top. El segundo al right y left y el tercero como bottom.
- **4 valores:** Empezando desde el top, es cada lado en sentido de las agujas del reloj (top, right, bottom, left).

```
P{
padding:10px 15px 30px 50px;
}
```

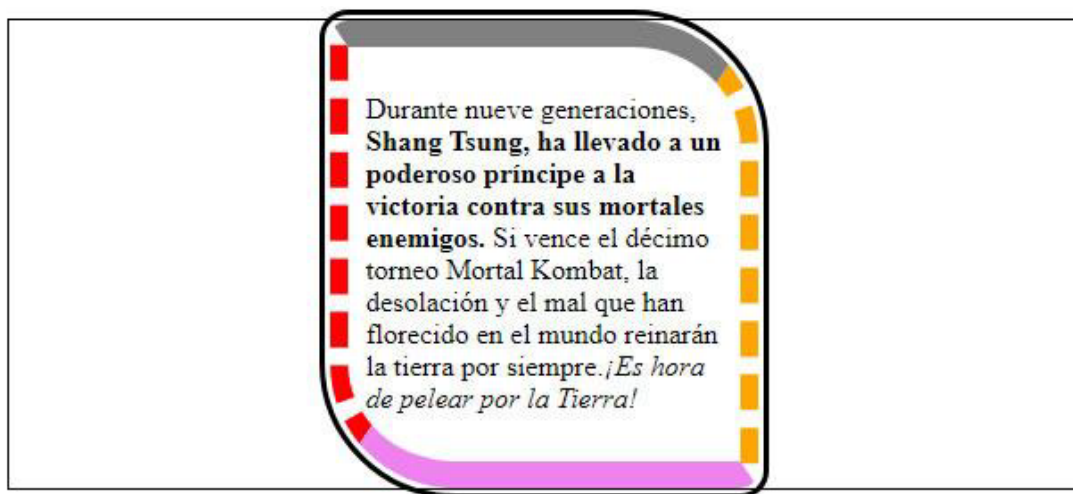
Los 4 valores equivalen a las propiedades explícitas de la columna derecha, en cuanto al orden.

```
p {
padding-top:10px;
padding-right:15px;
padding-bottom:30px;
padding-left:50px;
}
```

Datos sobre el margin (centrar elementos de bloque)

Es muy común que tengamos algunos contenedores que debemos centrar dentro del layout cuando diseñamos nuestro sitio.

Para lograr esto, hay una técnica muy sencilla, en donde sólo necesitamos definir un ancho en el elemento (debe ser de bloque) y asignarle un **margin:auto**: así, se logra repartir el espacio que tiene el elemento tanto a la derecha como a la izquierda, quedando **centrado**.

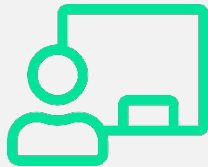


```
div {  
  width:300px;  
  margin:auto;  
  border-top: 15px solid grey;  
  border-right: 10px dashed orange;  
  border-bottom:15px solid violet;  
  border-left:10px dashed red;  
  border-radius: 10px 70px;  
  outline:black solid medium;  
  outline-offset: 3px;  
}
```

Box Model (propiedad box-sizing)

Ahora que vimos cómo manejar las propiedades de las cajas es importante saber cuánto miden estos elementos.

Hay que tener presente qué sucede con los **anchos** del **padding** y el **borde** cuando nuestras cajas tienen un tamaño declarado.



Veamos lo siguiente:

Si tenemos una caja de 400px de alto y 100px de ancho y le agregamos un borde de 20px y un padding de 20px ¿Cuánto mide realmente la caja?

Nuestro CSS tiene las reglas muy claras con respecto a las medidas que fijamos para este diseño:

```
div{  
  width:400px;  
  height:100px;  
  padding:20px;  
  border:solid red 20px;  
}
```

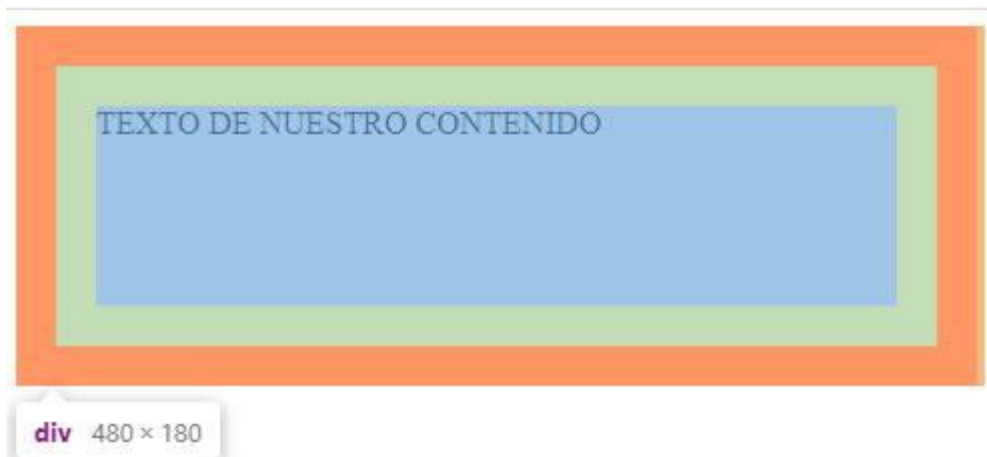
Esta pregunta puede tener dos respuestas y cuál de las dos se aplica depende de la propiedad **box-sizing**.

Si el valor de esta propiedad es **content-box**, entonces nuestra caja mide **480px de ancho y 180px de alto**:

De **ancho** le dimos unos **400px** que será lo que ocupe nuestro **contenido** + **40px de padding total** (tenemos **20px** del lado **izquierdo** y **20px** del lado **derecho**) y **40px de borde total** (**20px** del borde **izquierdo** y **20px** del borde **derecho**).

Por lo tanto, **los anchos de cada borde y de cada padding se suman al ancho y al alto declarado de la caja**.

El **valor content-box** es el valor por defecto que **CSS aplica a todas las cajas**, por lo tanto, si no lo modificamos, éste será el comportamiento que obtendremos. Es lo que se conoce como **box model tradicional**



```
div {box-sizing: content-box;}
```

Entonces, esos 400px están siendo reservados para el contenido, pero los tamaños de borde y padding hacen que el tamaño real de la caja sea mayor al declarado.

Si modificamos el valor de la propiedad **box-sizing a border-box**, entonces el tamaño de la caja será exactamente el declarado (400px x 100px). ¿Qué pasó con nuestra caja, y con los valores del padding y el borde?

Fueron restados del tamaño disponible para el contenido. Este modelo se lo conoce como **box model alternativo**.



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

Ahora, nuestra caja tiene solo **400px de ancho** y **100px alto** para mostrar su contenido, el resto del espacio es utilizado por los bordes y el padding. (de nuevo, dos en cada dirección.)

En ambos casos, el margen no entra en la ecuación, ya que éste existe por fuera de la caja.

Es importante tener en mente esta propiedad al armar la estructura de nuestro sitio ya que afectará la cantidad de contenido podemos mostrar. Estas reglas afectan a todos los elementos de nuestra página, ya que para CSS todo es una caja.

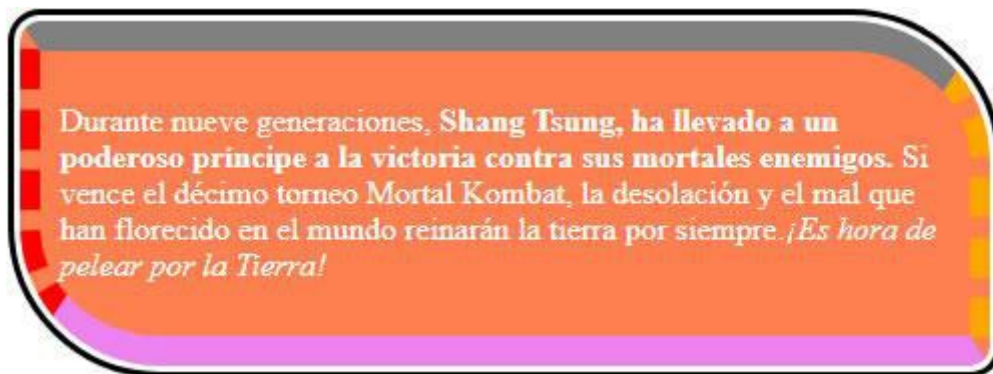
Color de fondo

Todos los elementos aceptan color de fondo mediante la propiedad **background-color**.

Aceptan como valor cualquier representación de color (nombre en inglés, hexadecimal, rgba, etc). Ese color ocupará toda la superficie (ancho y alto) del objeto con ese color sólido.

Hay que tener muy presente que el color seleccionado sea armónico con la estética de la web y contrastar con el color de texto.

Le agregaremos un color de fondo y buscaremos un color para el contraste de la tipografía.



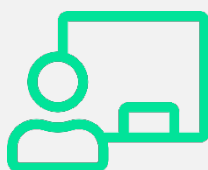
```
div {  
border-top: 15px solid grey;  
border-right: 10px dashed orange;  
border-bottom: 15px solid violet;  
border-left: 10px dashed red;  
border-radius: 10px 70px;  
outline: black solid medium;  
outline-offset: 3px;  
background-color: coral;  
color: white;  
}
```

¡El tema del background lo seguiremos desarrollando en las próximas clases!



Llegamos así al final de esta primera clase en la que vimos:

- Box Model (Cajas).
- Propiedades de Caja (margin, border, padding, content).
- Dimensiones de cajas (width, height).
- Propiedad border (style, width, color).
- Propiedad border-radius.
- Sintaxis de Margin y Padding.
- Diferencia entre border y Outline.
- Datos sobre el margin.
- Box model (propiedad box-sizing).
- Color de fondo.



Recuerden visualizar el **desafío semanal** que encontrarán en el aula y conectarse a la clase en vivo de esta semana.

¡Hasta la próxima clase!

Para ampliar la información:

Mozilla Developers Network, Documentación sobre CSS:

https://developer.mozilla.org/es/docs/Web/CSS/CSS_Box_Model/Introduction_to_the_CSS_box_model/