

Clase 07

- El padding, o relleno, es el espacio que hay entre el contenido y el borde de la caja.
- Esta propiedad consta de 4 valores posibles:
 - Padding-top, padding-right, padding-left, padding-bottom.
- O un atajo llamado “padding” que nos permite setear los valores de los cuatro costados.

- Si utilizamos el atajo, es importante que sepamos qué:
 - Un solo valor: se aplica a los cuatro lados.
 - Dos valores: el primero se aplica arriba y abajo, y el segundo a los costados.
 - Tres valores: el primero va arriba, el segundo a los costados y el tercero abajo.
 - Cuatro valores: Arriba, derecha, abajo, izquierda. Siempre en ese orden.

-

- Una vez que termina el padding, nos encontramos con el borde. Que literalmente es el borde de la caja.
- Aunque no lo veamos, todas las cajas tienen un borde y nosotros podemos trabajar con ellos.
- Los bordes tienen un montón de propiedades, que nos permiten modificar tres cosas:
 - Ancho.
 - Tipo.
 - Color.

- Para trabajar con el ancho de un borde tenemos la propiedad **border-width**.
- Recibe un valor numérico y mientras más grande sea, mayor será el grosor del borde.

Box Model – Border

- Si queremos modificar el color, utilizamos **border-color**.
- Esta recibe un color en cualquiera de sus variantes: palabra clave, hexadecimal, rgb, etc.



Box Model – Border

- El estilo del borde es lo más particular.
- Spoiler: **SON HORRIBLES.**
- La propiedad para modificarlo es border-style, y recibe alguno de los siguientes valores:
 - Solid.
 - Dotted.
 - Dashed.
 - Double.
 - Inset.



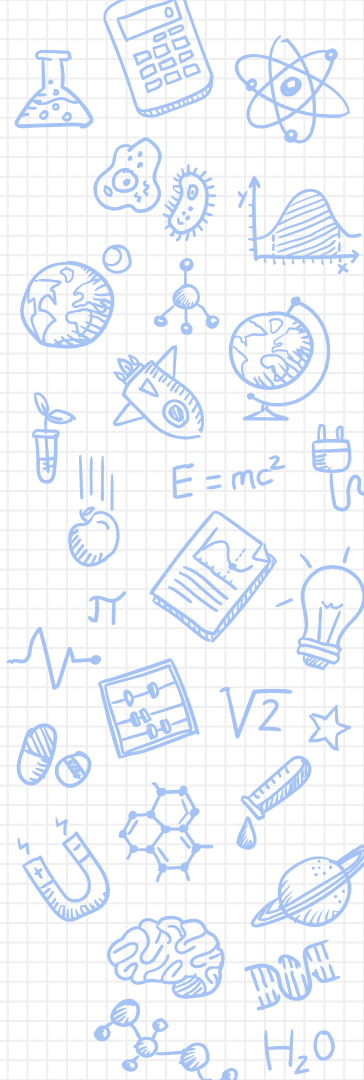
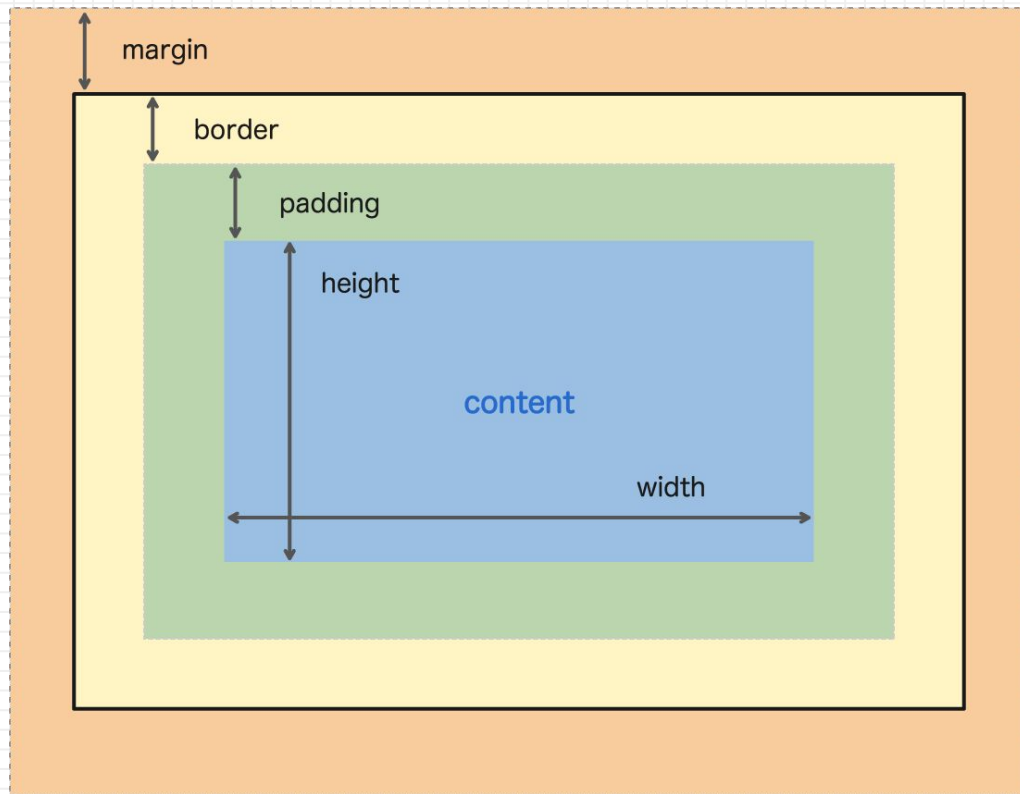
-

- Por último. Una de las cosas más interesantes que tiene border, es la posibilidad de trabajar con su redondeado.
- Existe una propiedad llamada **border-radius**, que recibe un valor en px o %, y se utiliza para redondear los bordes.
- Con esta propiedad, por ejemplo, podemos generar un círculo perfecto a partir de un cuadrado.

- El **margin** es el espacio que hay luego del borde de un elemento, para separarlo de los que estén a su alrededor.
- Al igual que el **padding**, podemos especificar un **margin-left**, **margin-top**. O utilizar la propiedad **margin** y pasarle diferentes valores.

-

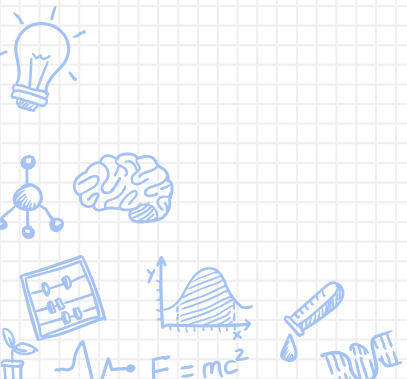
Box Model



- Lo que indica la teoría es que el tamaño final de un elemento es:
 - contenido + padding + border.
- Entonces, si nosotros tenemos una caja de 100x100px. Y le agregamos 20px de padding y 2px de borde. ¿Cuánto mide ahora?



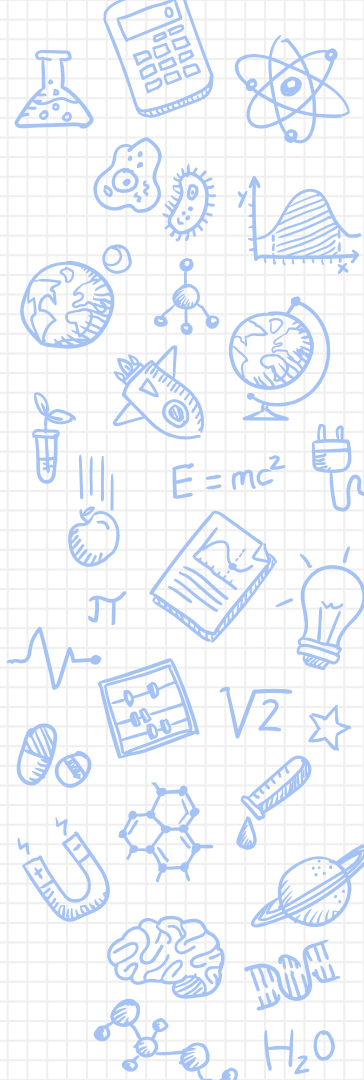
i144 x 144!



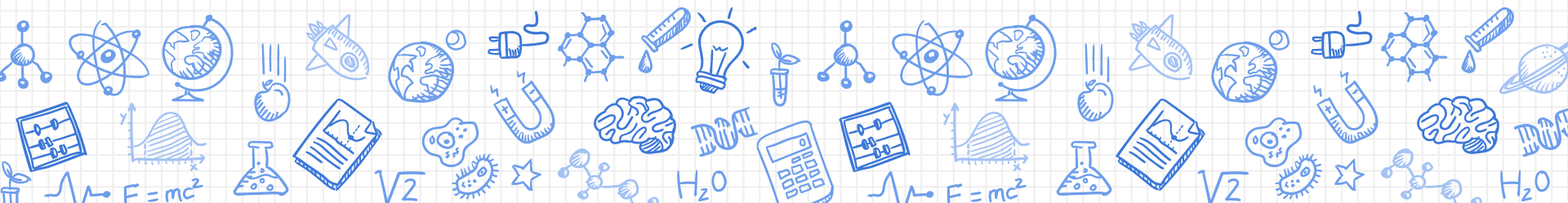
- Cuando nosotros agregamos 20px de padding, estamos agregando 20px **a cada lado**. Lo mismo sucede con el borde.
- Si nosotros hubiéramos especificado diferentes valores para cada lado, la cuenta sería distinta.

En la clase de hoy:

- ## - CSS - Parte 3



Display



- Display es una propiedad que se encarga de definir cómo se va a ver cada uno de los elementos.
- Dijimos que existían los elementos de línea y de bloque. Display tiene la capacidad de alterar esa naturaleza.
- Los valores que puede recibir son:
 - Block.
 - Inline.
 - Inline-block.

- Con el valor `block`, `display` se encarga de tomar un elemento de línea y convertirlo a bloque para que se comporte como tal.

- En cuanto al valor inline, se utiliza para lo opuesto a block.
- Con este valor, tomamos un elemento de bloque y lo pasamos a línea.

Display – Inline-block

- Sin dudas, el valor más interesante es inline-block.
- Ya que va a permitir que los elementos se comporten como si fueran de línea, pero acepten todas las propiedades de un elemento de bloque.
- Esta variante es sin dudas la más utilizada.



- Existe un cuarto valor, que se utiliza para ocultar elementos.
- Por ahora no vamos a estar usando esta opción, pero si lo vamos a hacer cuando trabajemos en responsive.
- Este valor es el **display: none;**

- Paralelamente, existe una propiedad llamada visibility que recibe dos posibles valores:
 - Visible.
 - Hidden.
- A diferencia del display, si ocultamos el elemento, no se va a ver pero su espacio va a seguir reservado.
- Con display lo quitamos y otro elemento se acomoda en su lugar.

- Las flotaciones son probablemente la forma más antigua de mover elementos en un sitio web.
- Esta propiedad float, solo recibe tres valores posibles:
 - Left.
 - Right.
 - None.

Float

- El float none, lo que indica es que ese elemento debe dejar de flotar.
- Pero, qué es flotar? Flotar un elemento significa llevarlo lo más a la derecha o a la izquierda y arriba posible.



Float

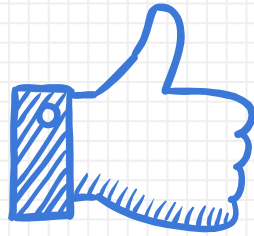
- La primer particularidad que encontramos es que deja de pertenecer al flujo natural de nuestra página.
- Esto quiere decir que el HTML deja de reconocerlo por lo cual todos los elementos que vengan después van a tratar de re-acomodarse.
- Para solucionar este problema, existe la propiedad **clear**.



- Clear es una propiedad que nos permite forzar el comportamiento de una caja, más allá de que se encuentre debajo de una flotante.
- Esta propiedad recibe tres valores posibles:
 - Left.
 - Right.
 - Both.

- Cuando utilizamos clear, de alguna manera le estamos indicando al elemento que no le dé importancia a un elemento flotado que pueda tener antes o después, y se posicione como vendría por defecto.
- Los valores left y right dependen específicamente de dónde esté el elemento flotado.
- Both es una especie de comodín, que afecta a ambos lados al mismo tiempo.

- El segundo inconveniente, es que un elemento flotado deja de ser reconocido por su contenedor para determinar la altura del mismo.
- Este problema es muy común pero se puede solucionar con la propiedad overflow y el valor auto, aplicado al contenedor. Para que incluso si tiene un elemento flotado, lo reconozca.



SE ACABÓ!

Dudas?

Escriban a:

✕ federico.mirandaa@gmail.com