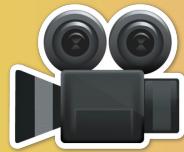


**RECUERDA PONER A GRABAR LA
CLASE**





¿DUDAS DEL ON-BOARDING?

[MIRALO AQUI](#)



Clase 04. DESARROLLO WEB

CSS
MEDIDAS, COLORES, FUENTES
+ **BEM**



OBJETIVOS DE LA CLASE

- Conocer el uso de medidas, fondos y fuentes en CSS.
- Conocer la metodología BEM
- Aplicar los atributos al Proyecto Final.

GLOSARIO:

Clase 3

Etiqueta IFRAME: es un elemento HTML que permite insertar o incrustar un documento HTML dentro de un documento HTML principal.

CSS (Cascading Style Sheets): es un lenguaje web para aplicar formato visual (color, tamaño, separación y ubicación) al HTML. Con él puedes cambiar por completo el aspecto de cualquier etiqueta HTML.

Padre e hijos: es un concepto que se aplica cuando tienes una etiqueta “dentro” de otra. Esto te habilita a agregar atributos específicos a “hijos”, sin alterar los del “padre”.

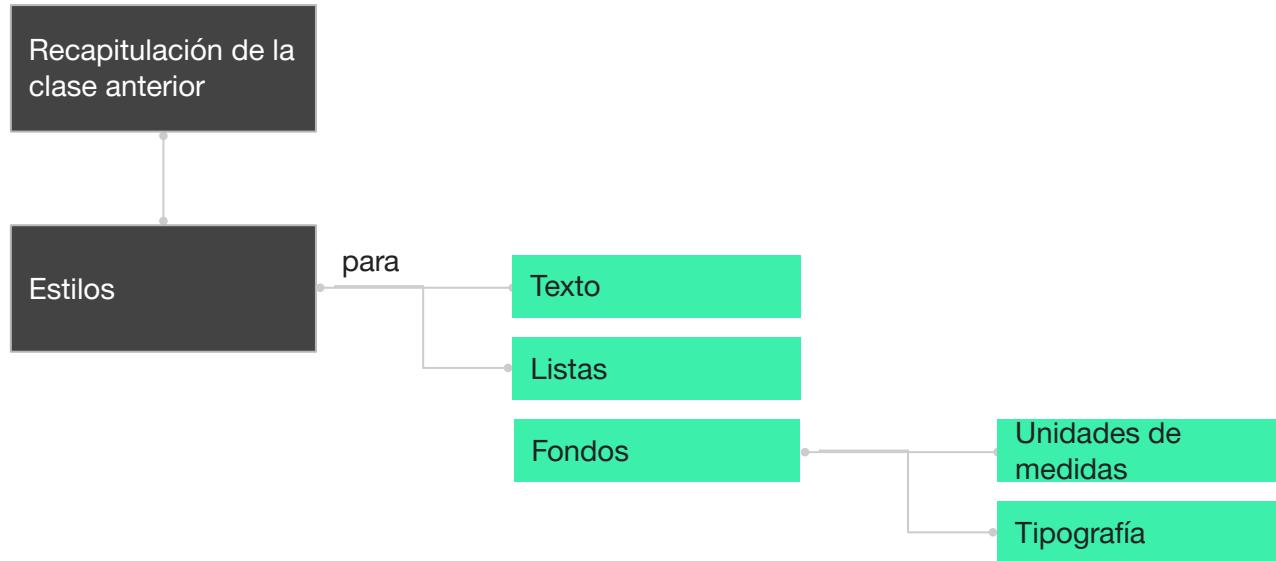
Class: generalmente se utiliza para darle estilos a cierta parte del código.

Atributo ID: suele usarse para nombrar porciones de código y sectores, como por ejemplo cuando quieres nombrar distintas secciones.

MAPA DE CONCEPTOS

MAPA DE CONCEPTOS CLASE 4

¡Para recordar!



CRONOGRAMA DEL CURSO

Clase 3



Incluyendo CSS a nuestro proyecto



PRÁCTICAS DE LO VISTO EN CLASE



ATRIBUTOS



AGREGANDO CSS A NUESTRO HTML

Clase 4



CSS - Medidas, colores, fuentes



PRÁCTICAS DE LO VISTO EN CLASE



ESTILOS



TIPOGRAFÍA



ASIGNANDO ESTILOS



PRIMERA ENTREGA DEL PROYECTO FINAL

Clase 5



Box Modeling



PRÁCTICAS DE LO VISTO EN CLASE



BOX MODEL



GUIÓN DE LA CLASE

Accede al material complementario [aquí](#).



Momento de exposición teórica

RECAPITULACIÓN DE LA CLASE ANTERIOR

CONCEPTOS PREVIOS

Sabemos que existen tres maneras de aplicar CSS a un documento HTML:

- Hacerlo sobre la etiqueta con el atributo `style=""`
- En el head, insertar la etiqueta `<style>`
- Buscar un archivo externo con un `<link />`

(Es de las etiquetas que se cierran solas. Requiere el= "stylesheet" para funcionar. Además un href= "" con la ruta al archivo.)

SELECCIÓN DE HTML MEDIANTE CSS

3 formas

Por etiqueta



```
h1 {  
    propiedad: valor;  
}
```

Por clase
(anteponiendo el ".")



```
.clase {  
    propiedad: valor;  
}
```

Por ID
(anteponiendo el "#")



```
#id {  
    propiedad: valor;  
}
```

SELECCIÓN DE HTML MEDIANTE CSS

Ejemplos



p a

```
<p>  
Párrafo con <a href="">enlace</a>  
</p>
```



a.foo

```
<a href="" class="foo">Enlace</a>
```



a#foo

```
<a href="" id="foo">Enlace</a>
```



.foo .bar

```
<etiqueta class="foo">  
  <etiqueta class="bar"></etiqueta>  
</etiqueta>
```

SELECCIÓN DE HTML MEDIANTE CSS

Ejemplos



.foo .foo #foo



p.foo a.bar

```
<etiqueta class="foo">
<etiqueta class="foo">
    <etiqueta id="foo"></etiqueta>
</etiqueta>
</etiqueta>
```

```
<p class="foo">
    <a href="" class="bar">Enlace</a>
</p>
```

LOS NOMBRES DE LAS CLASES E IDS

No es posible crear nombres separados por espacios.

La “*joroba de camello*”, permite que se puedan leer de forma más simple palabras compuestas.

claseDeMaquetacion

conBorde

productosMasVendidos



BEM

CODER HOUSE

BEM

Todos queremos hacer que nuestro código sea más fácil de leer. Esto nos ayuda a trabajar más rápidamente y de manera eficiente, y cuando otros trabajen con nosotros podremos mantener claridad y coherencia.

Hoy vamos a cubrir la metodología BEM, que nos ayudará a entender estructuras de CSS, y a mejorar las nuestras.

BEM

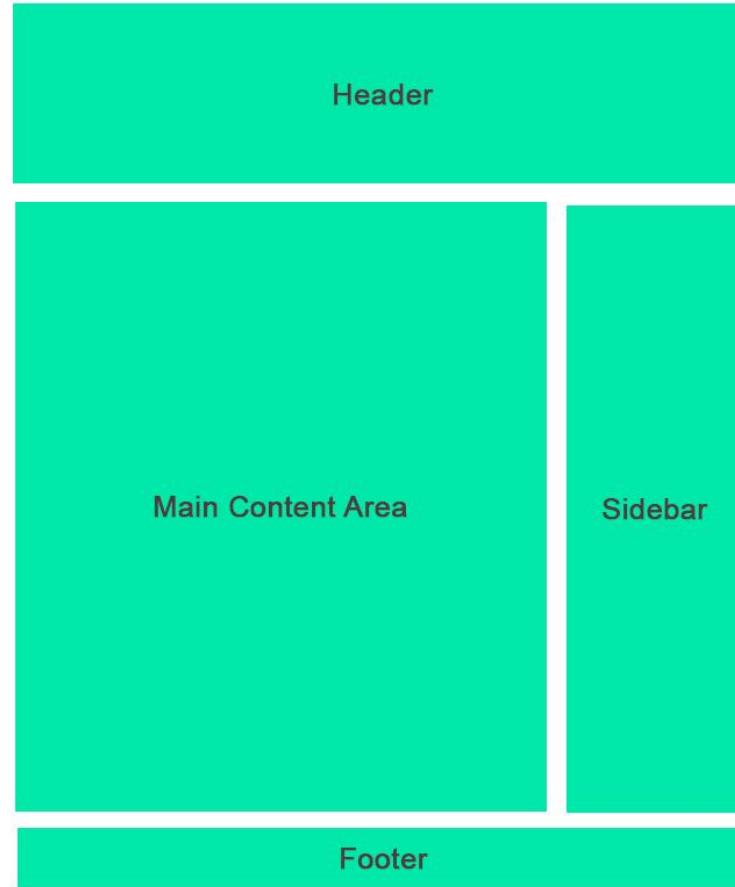
BEM significa *Modificador de Bloques de Elementos (Block Element Modifier)* por sus siglas en inglés. Sugiere una manera estructurada de nombrar tus clases, basada en las propiedades del elemento en cuestión.

BEM tiene como horizonte modularizar lo máximo posible cada uno de los bloques de código dispuesto. Se centra en tres parámetros o variables posibles: bloques (div, section, article, ul, ol, etc.), elementos (a, button, li, span, etc.) y modificadores. Estos últimos se definen de acuerdo a la posterior utilización que haga el desarrollador a cargo.

BLOQUE

El *bloque* es un contenedor o contexto donde el elemento se encuentra presente. Piensa como si fueran partes estructurales de código más grandes. Puede que tengas un encabezado, pie de página, una barra lateral y un área de contenido principal; cada uno de estos sería considerado como un bloque. Miremos la imagen a continuación:

BLOQUE



BLOQUE

El bloque de elementos forma la raíz de la clase y siempre irá primero. Solo debes saber que una vez que has definido tu bloque, estarás listo para comenzar a nombrar tus elementos.

```
.block__element {  
    background-color: #FFFFFF;  
}
```

BLOQUE

La doble barra baja te permite visualizar, navegar rápidamente y manipular tu código. Aquí hay algunos ejemplos de cómo funciona la metodología de elementos:

```
.header__logo {}  
.header__tagline {}  
.header__searchbar {}  
.header__navigation {}
```

BLOQUE

HTML

```
11    <section>
12        <article class="noticia">
13            <!--Bloque contenedor-->
14        </article>
15    </section>
```

CSS

```
1 .noticia{
2     background: #lightgray;
3 }
```

El punto es mantener los nombres simples, claros, y precisos.
No lo pienses demasiado.

Actualizar el nombre de las clases no debería ser un problema cuando encuentras una mejor semántica que funcione (sólo debes tratar de ser consistente, y apegarte a ella).

MODIFICADORES

Cuando nombras una clase, la intención es ayudar a que ese elemento pueda ser repetido, para que no tengas que escribir nuevas clases en otras áreas del sitio si los elementos de estilo son los mismos.

Cuando necesitas modificar el estilo de un elemento específico, puedes usar un modificador. Para lograr esto, añade un doble guión -- luego del elemento (o bloque). Aquí tenemos un corto ejemplo:

```
.block--modifier {}  
.block__element--modifier {}
```

ELEMENTOS

El elemento es una de las piezas que compondrán la estructura de un bloque. **El bloque es el todo, y los elementos son las piezas de este bloque.**

De acuerdo a la metodología BEM, cada elemento se escribe después del bloque padre, usando dos guiones bajos.

ELEMENTOS

CSS

```
5  .noticia__titulo{  
6      font-family: 'Times New Roman', serif;  
7  }
```

HTML

```
11 <section>  
12   <article class="noticia">  
13     <h1 class="noticia__titulo">Título de la noticia</h1>  
14   </article>  
15 </section>
```



¡NUNCA USES SÓLO MODIFICADORES!

Las clases *modificador* siempre deben acompañar a una clase *bloque* o una clase *elemento*, no tiene sentido que aparezcan solas.

Esto está mal 

```
<button class="button-primary"></button>
```

```
<button class="menu__button-primary"></button>
```

Esto está bien 

```
<button class="button button-primary"></button>
```

```
<button class="menu__button menu__button-primary"></button>
```

MODIFICADORES

CSS

```
 9  .noticia--destacada{  
10    background: #dimgray;  
11  }  
12  .noticia__titulo--uppercase{  
13    text-transform: uppercase;  
14  }
```

HTML

```
11  <section>  
12    <article class="noticia--destacada">  
13      <h1 class="noticia__titulo--uppercase">Título de la noticia</h1>  
14    </article>  
15  </section>
```

BEM



Ventajas

- Añade especificidad.
- Aumenta la independencia.
- Mejora la herencia múltiple.
- Permite la reutilización.
- Entrega simplicidad.



Desventajas

- Las convenciones pueden ser muy largas.
- A algunas personas les puede tomar tiempo aprender la metodología.
- El sistema de organización puede ser difícil de implementar en proyectos pequeños.

BEM: ¿PARA QUÉ LO USARÍAS?

- Para simplificar nuestro CSS y conseguir un estilo consistente, por lo que nuestro código será mucho más legible y fácil de mantener.
- Si estamos usando Bootstrap y queremos modificar ciertas clases.
- Cuando trabajamos en equipo y cada miembro tiene una manera distinta de escribir CSS

PROPIEDADES CSS

RESETEO CSS

Los reset CSS contienen en su código fuente definiciones para propiedades problemáticas, que los diseñadores necesitan unificar desde un principio.

Por ejemplo, la mayoría de navegadores establece un margen por defecto entre el contenido de la página web y su propia ventana, cuyo valor varía de un navegador a otro.

Para subsanar esa diferencia, los diseñadores y las diseñadoras de sitios webs suelen declarar la siguiente línea al comienzo de sus hojas de estilo:

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

RESETEO CSS

Esa única línea indica, con el selector universal de CSS representado por un asterisco, que todos los elementos contenidos en el HTML a los que se aplique, carecerán de márgenes. De esa manera, el diseñador o la diseñadora se verán obligados a declarar luego los márgenes necesarios en el diseño de su página web, en cada uno de los lugares donde se requiera, sin tener que dejar ese aspecto a decisión de ningún navegador, y minimizando las diferencias visuales entre los mismos.

Atención: los reset CSS pueden contener esa y otras muchas líneas de código que, en su conjunto, servirán al diseñador/a web para unificar su visualización entre navegadores.

ESTILO PARA LISTA

LIST-STYLE-TYPE

CSS

```
ol {  
    list-style-type: none;  
}  
  
ul {  
    list-style-type: none;  
}
```

Aplicando esta propiedad y este valor, vamos a poder eliminar las bullets y los números.

Valores posibles: ([ver aquí](#))

ESTILOS PARA TEKTO

FONT-STYLE

CSS

```
.normal {  
    font-style: normal;  
}  
  
.italica {  
    font-style: italic;  
}
```

Se ve así

Texto normal

Texto en italica

Valores comunes: normal | italic

FONT-WEIGHT

CSS

```
.negrita {  
    font-weight: bold;  
}  
  
.normal {  
    font-weight: normal;  
}
```

Se ve así

Texto en negrita

Texto normal

Valores comunes: normal | bold (luego verán, que puede tener otros valores, en números)

FONTSIZE

CSS

```
.textoGrande {  
    font-size: 20px;  
}  
  
.textoRelativo {  
    font-size: 200%;  
}
```

Se ve así

Texto en 20 px

Texto en 200%

Valores posibles: <medida de longitud> | <porcentaje>

FONT-FAMILY

CSS

```
.impact {  
    font-family:Impact, sans-serif;  
}  
  
.comicSans {  
    font-family: "Comic Sans MS",  
    sans-serif;  
}
```

Se ve así

Familia Impact

Familia Comic

Valores posibles: <familia o nombre genérico>

FONT-FAMILY

Cada sistema operativo y navegador interpretan de distinta forma las fuentes predeterminadas.

- **Serif:** «Times New Roman» en Windows, y «Times» en Macintosh (diferente a la de Windows).
- **Sans serif:** «Arial» en Windows, y «Helvetica» en Macintosh.
- **Monospace:** «Courier New» en Windows, «Courier» en Macintosh, y por lo general «VeraSans» o «DejaVuSans» en Linux.

Nota: te recomendamos visitar el sitio <https://www.cssfontstack.com/>, para conocer más acerca de cómo funciona cada fuente, en los distintos sistemas operativos.

FONT-FAMILY



Las **tipografías Serif** son aquellas que llevan remates, es decir, detalles adicionales en los bordes de las letras. El ejemplo por excelencia de Serif es la **Times New Roman**. Son muy usadas en los periódicos impresos, puesto que los detalles de las letras ayudan a seguir la lectura. Úsalas si quieras transmitir clasicismo, formalidad, precisión, tradición, delicadeza y/o refinamiento.

Por el contrario, las **tipografías Sans Serif**, como su nombre indica –sans es sin en francés-, carecen de estos detalles y también son denominadas de palo seco. Algunas de las más conocidas son la **Arial** o la **Calibri**. Se utilizan mucho en entornos digitales, puesto que los detalles son difíciles de plasmar en píxeles. Transmiten fuerza, modernidad, vanguardia, elegancia y actualidad, a los diseños y textos en los que se incluyen.

Nota: este texto está escrito en Arial que es una tipografía Sans Serif.

TEXT-ALIGN

CSS

```
.centrar {  
    text-align: center;  
}  
 
```

```
.aLaDerecha {  
    text-align: right;  
}
```

Se ve así

texto

texto

Valores posibles: left | right | center | justify

LINE-HEIGHT

CSS

```
.interlineado {  
    line-height: 1.6;  
}
```

Se ve así

texto
ejemplo

Valores posibles: none | <número> | <longitud> | <porcentaje>

TEXT-DECORATION

CSS

```
.subrayado {  
    text-decoration: none;  
}  
  
.tachado {  
    text-decoration: line-through;  
}
```

Se ve así

Enlace

Parrafo

Valores posibles: none | underline | overline | line-through

ESTILOS PARA FONDOS

BACKGROUND-COLOR

CSS

```
.fondoFuerte {  
    background-color: yellow;  
}
```

Valores posibles: [color]

Se ve así

Parrafo fondo amarillo

BACKGROUND-IMAGE

CSS

```
.catsandstars {  
    background-image:  
    url("https://mdn.mozilla.org/  
files/11991/startransparent.gif"),  
    url("https://mdn.mozilla.org/  
files/7693/catfront.png");  
}
```

Valores posibles: url | none

Se ve así



Párrafo con fondo.

Párrafo sin fondo.

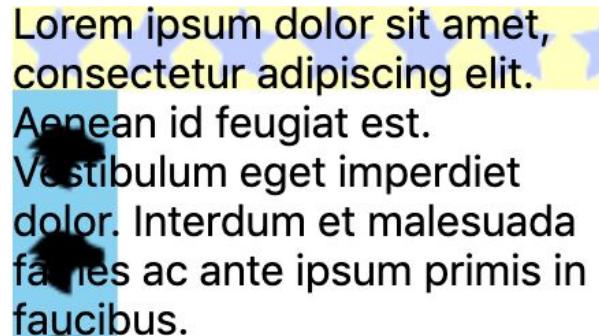
BACKGROUND-REPEAT

CSS

```
.ejemplo {  
    background-image:  
url(https://mdn.mozilla.org/files/12005/starsolid.gif),  
url(https://developer.mozilla.net/media/redesign/img/favicon32.png);  
    background-repeat: repeat-x,  
                    repeat-y;  
}
```

Valores posibles: repeat | repeat-x | repeat-y | no-repeat | space | round ([ver ejemplos](#))

Se ve así



Lorem ipsum dolor sit amet,
consectetur adipiscing elit.
Aenean id feugiat est.
Vestibulum eget imperdiet
dolor. Interdum et malesuada
faucibus ac ante ipsum primis in
faucibus.

BACKGROUND-POSITION

CSS

```
.ejemplo {  
    background-image:  
url("https://mdn.mozilla.org/fil  
es/12005/starsolid.gif");  
    background-repeat: no-repeat;  
    background-position: right center;  
}
```

Valores posibles: posicionX posicionY ([ver ejemplos](#))

Se ve así

Lorem ipsum dolor sit amet,
consectetur adipiscing elit.
Aenean id feugiat est.
Vestibulum eget imperdiet  dolor. Interdum et malesuada
fames ac ante ipsum primis in
faucibus.

BACKGROUND-SIZE

CSS

```
.ejemplo {  
    background-image:  
url("https://mdn.mozilla.org/fil  
es/12005/starsolid.gif");  
    background-repeat: no-repeat;  
    background-size: cover;  
}
```

Se ve así

Lore ipsum dolor sit amet,
consectetur adipiscing elit.
Aenean id feugiat est.
Vestibulum eget imperdiet
dolor. Interdum et malesuada
fames ac ante ipsum primis in
faucibus.

Valores posibles: [ancho] | [alto] | cover | contain ([ver ejemplos](#))

UNIDADES DE MEDIDAS

UNIDADES DE MEDIDAS

Hay una amplia variedad de absolutas y relativas, pero nos centraremos en:

Absolutas

- **Px (pixels):** es la unidad que usan las pantallas.

Relativas

- **Rem:** relativa a la configuración de tamaño de la raíz (etiqueta html).
- **Porcentaje:** tomando en cuenta que 16px es 100%.
- **Viewport:** se utilizan para layouts responsivos (más adelante).

UNIDADES DE MEDIDAS

El problema con los píxeles.

The image shows a split-screen comparison. On the left is a screenshot of the Santander mobile banking app. At the top is the Santander logo. Below it is a large text area containing the message "Hola :) ¿en qué te podemos ayudar?". Below this are navigation links: "do", "Tarjetas", "Cuentas", "Seguros", "AAdvantage", and "SuperClub". At the bottom is a dark grid with columns labeled A through I and rows labeled 1 through 4, containing numerical values. On the right is a screenshot of a browser's "Configuración" (Configuration) menu. The "Diseño" (Design) section is visible, featuring a "Temas" (Themes) link that points to the "Abrir Chrome Web Store" (Open Chrome Web Store). Below this are two other configuration options: "Mostrar el botón de la Página principal" (Show Home button) which is set to "Inhabilitada" (Disabled), and "Mostrar barra de favoritos" (Show bookmarks bar). The "Tamaño de fuente" (Font size) option is highlighted with a red border and a dropdown menu showing "Muy pequeño" (Very small). At the bottom is a "Personalizar fuentes" (Customize fonts) link.

	A	B	C	D	E	F	G	H	I
1	07	45	77	85	23	23	24	23	24
2	94	78	83	68	75	75	44	75	44
3	75	09	93	16	16	16	51	16	51
4	22	40	65	22	22	22	22	22	22

UNIDADES DE MEDIDAS

¿Encuentras las diferencias? Exacto, no cambió nada

The screenshot shows a portion of the Santander website's homepage on the left and a 'Configuración' (Configuration) overlay on the right. The configuration overlay has a blue header with a search icon. It displays the word 'Diseño' (Design) and a section titled 'Temas' (Themes) with a link to 'Abrir Chrome Web Store'. Below this, there are two settings: 'Mostrar el botón de la Página principal' (Show the Home button) which is set to 'Inhabilitada' (Disabled), and 'Mostrar barra de favoritos' (Show the Favorites bar). At the bottom, there are two buttons: 'Tamaño de fuente' (Font size) with a red border around it, and 'Muy grande' (Very large) with a blue border.

Santander

= Configuración

Diseño

Temas

Abrir Chrome Web Store

Mostrar el botón de la Página principal

Inhabilitada

Mostrar barra de favoritos

Tamaño de fuente

Muy grande

UNIDADES DE MEDIDAS

Ejemplo que tiene la solución:

ELECCIONES EN URUGUAY

Las encuestadoras dan una leve ventaja

Luis Lacalle Pou (Partido Nacional) es el favorito, mientras que dar la sorpresa



Temas

Abrir Chrome Web Store

Mostrar el botón de la Página principal

Inhabilitada

Mostrar barra de favoritos

Tamaño de fuente

Muy pequeño

Personalizar fuentes

UNIDADES DE MEDIDAS

¿Ahora sí está diferente, no?

ELECCIONES EN URUGUAY

Las encuestadoras da Pou

Luis Lacalle Pou (Partido Nacional) es el favorito. El candidato del Frente Amplio, espera dar la sorpresa



DISEÑO

Temas

Abrir Chrome Web Store

Mostrar el botón de la Página principal

Inhabilitada

Mostrar barra de favoritos

Tamaño de fuente

Muy grande

UNIDADES DE MEDIDAS

Ahora veamos qué medida es más conveniente para los textos.

```
html { /* etiqueta raíz */  
    font-size: 62.5%;  
}  
  
p {  
    font-size: 2rem; /* 20px */  
}
```

**Texto simulando
20px**

62.5%, hace que en vez de que 16px sea el valor a tomar en cuenta para calcular las unidades relativas, se use 10px.

TIPOGRAFÍA

TIPOGRAFÍA WEB

Habíamos visto que usando “*font-family*”, es posible agregar algunas limitadas fuentes, pero... podemos usar muchísimas opciones de fuentes con “**Google Fonts**”.



Viewing 877 of 877 font families

Roboto

Christian Robertson (12 styles)



Black Han Sans

Zess Type (1 style)



All their equipment
and instruments
are alive.

**A red flare
silhouetted
the jagged
edge of a wing.**

Gugi

TAE System & Typefaces Co. (1 style)



Open Sans

Steve Matteson (10 styles)



I watched the
storm, so
beautiful yet

Almost before we
knew it, we had
left the ground.

Categories

- Serif
- Sans Serif
- Display
- Handwriting
- Monospace

Sorting

Trending ▾

Languages

All Languages ▾

Number of styles

Thickness

Slant

TIPOGRAFÍA WEB

CSS

```
h1 {  
    font-family: 'Roboto',  
    sans-serif;  
}
```

Ver [Google Fonts](#)

HTML

```
<head>  
    <link  
        href="https://fonts.googleapis.com  
        /css?family=Roboto&display=swap"  
        rel="stylesheet">  
    <title>Document</title>  
</head>
```

iBONUS!

[EMBED](#)[CUSTOMIZE](#)

Load Time: Moderate

Roboto

- thin 100
- thin 100 Italic
- light 300
- light 300 Italic
- regular 400
- regular 400 Italic
- medium 500
- medium 500 Italic
- bold 700

¡BONUS!

CSS

```
h1 {  
    font-family: 'Roboto',  
    sans-serif;  
    font-weight: 100;  
}
```

HTML

```
<link  
    href="https://fonts.googleapis.com  
    /css?family=Roboto:100,300,400,500  
    ,700&display=swap"  
    rel="stylesheet">
```

Ver [Google Fonts](#)

¡BONUS!

Se ve así porque 100 es lo más delgado posible:

Titulo



BREAK

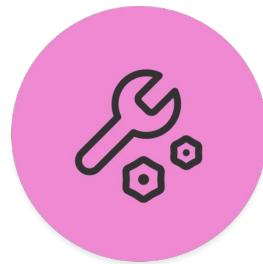
¡5/10 MINUTOS Y VOLVEMOS!

Ejemplo
en vivo



¡VAMOS A PRACTICAR LO VISTO!

Momento de consolidación de aprendizajes



ESTILOS

- Crea un archivo HTML y otro CSS.
- Agrega un estilo de tipografía a un archivo.



¡A PRACTICAR!

Crea un archivo HTML con un elemento párrafo dentro de él, con texto Lorem Ipsum. Además, crea un archivo CSS y agrégale estilo al párrafo con los siguientes valores:

- Color de texto: #0033ff
- Indexar una google font a elección.
- Espacio entre renglones: 15px
- Tamaño de texto: 12px

Cuentas con **15 minutos** para completar la actividad.



ASIGNANDO ESTILOS

Agrega nuevas propiedades en el CSS de tu Proyecto.

ASIGNANDO ESTILOS

Formato: archivo HTML y CSS. Debe tener el nombre “Idea+Apellido”.

Sugerencia: carpeta en formato zip o rar con el/los archivos html y css.

Desafío
Complementario



>> Consigna: asigna estilos vistos en esta clase al proyecto final, en función de lo que tenías en mente gracias al boceto.

>>Aspectos a incluir en el entregable:

Incluir de acuerdo a tu proyecto algunas de las propiedades vistas:

- Texto: font-style, font-weight, font-size, font-family, text-align, line-height y text-decoration.
- Listas: list-style-type, list-style-position y/o list-style.
- Fondos: background-color, background-image, background-repeat, background-position y/o background-size.
- Tipografía de [Google Fonts](#).

>>Ejemplo:

[Carpeta comprimida con los archivos](#)

SUGERENCIA: ¡HAZ ESTE DESAFÍO COMPLEMENTARIO!

Hacer el desafío "***Asignando estilos***" es importante porque podrás desarrollar código. Además, te ayudará a ver cómo traducir todo tu diseño al lenguaje de las computadoras.

De esta manera, aprenderás a poner en práctica propiedades que se usan todo el tiempo en el mundo del trabajo, y a la hora de la pre-entrega y posteriormente la entrega de tu proyecto final, aportarán valor agregado si las puedes manejar de manera correcta.



PRIMERA ENTREGA DEL PROYECTO FINAL

PRIMERA ENTREGA DEL PROYECTO FINAL

Formato: aclarar tipo de archivo en que se espera el desafío. Debe tener el nombre “Idea+Apellido”.

Sugerencia: activar comentarios en el archivo.

Proyecto
Final



2

>>Objetivos Generales:

1. Lograr armar un primer prototipo de la web, su estructura en HTML y estilo inicial.

>>Objetivos Específicos:

1. **Prototipar la web:** para tener una idea clara del resultado al que quiere llegar.
2. **Maquetar la web:** El estudiante deberá utilizar los tags, en especial los semánticos, para desde el código describir la estructura de la web.
3. **Crear un estilo inicial:** El estudiante deberá comenzar a darle estilo básico a su web definiendo colores. El documento debe estar linkeado en las páginas del proyecto.

>>Se debe entregar:

- Sketch.
- Estructura de la web en HTML: página principal (index.html), cada una de las secciones y enlaces entre secciones.
- Estilo básico: agregar al CSS colores pertinentes al diseño inicial.

*¿*PREGUNTAS?

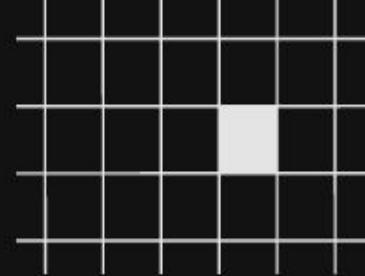




**¿QUIERES SABER MÁS? TE DEJAMOS
MATERIAL AMPLIADO DE LA CLASE**



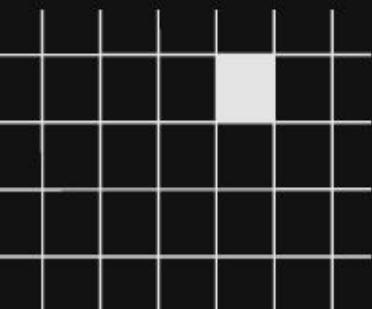
- [Referencias de reglas tipográficas](#) | **CSS Reference**
- [Aplicación para generar paletas de colores](#) | **Palette App**
- [Aplicación para generar paletas de colores](#) | **Lyft Design**



¡MUCHAS GRACIAS!

Resumen de lo visto en clase hoy:

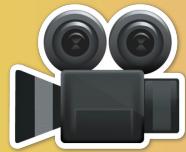
- Tipografia, tipo de fuentes, listas.
 - Fondos.





OPINA Y VALORA ESTA CLASE

**RECUERDA PONER A GRABAR LA
CLASE**





¿DUDAS DEL ON-BOARDING?

[MIRALO AQUI](#)



Clase 05. DESARROLLO WEB

BOX MODELING



OBJETIVOS DE LA CLASE

- Comprender las cajas y sus propiedades.
- Modelar la página web.
- Conocer las posiciones de un elemento.

GLOSARIO:

Clase 4

Joroba de camello: permite que se puedan leer de forma más simple palabras compuestas.

Reset CSS: contienen en su código fuente definiciones para propiedades problemáticas, que los diseñadores necesitan unificar desde un principio.

Unidades de medidas

Absolutas

- **Px (pixels):** es la unidad que usan las pantallas.

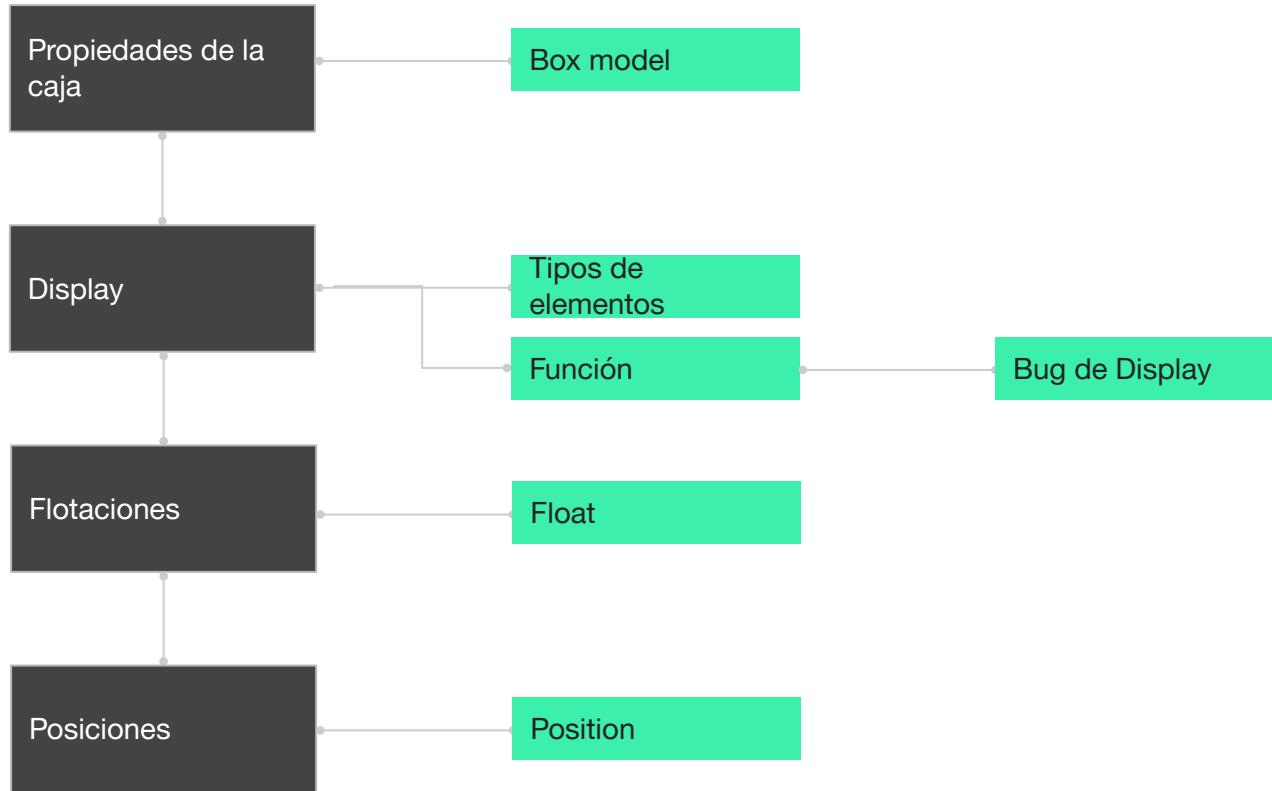
Relativas

- **Rem:** relativa a la configuración de tamaño de la raíz (etiqueta html).
- **Porcentaje:** tomando en cuenta que 16px es 100%.
- **Viewport:** se utilizan para layouts responsivos (más adelante).

MAPA DE CONCEPTOS

MAPA DE CONCEPTOS CLASE 5

¡Para recordar!



CRONOGRAMA DEL CURSO

Clase 4



CSS - Medidas, colores, fuentes



PRÁCTICAS DE LO VISTO EN CLASE



ESTILOS



TIPOGRAFÍA



ASIGNANDO ESTILOS



PRIMERA ENTREGA DEL PROYECTO FINAL

Clase 5



Box Modeling



PRÁCTICAS DE LO VISTO EN CLASE



BOX MODEL

Clase 6



Flexbox



PRÁCTICAS DE LO VISTO EN CLASE



APLICAR FLEXBOX



GUIÓN DE LA CLASE

Accede al material complementario [aquí](#).

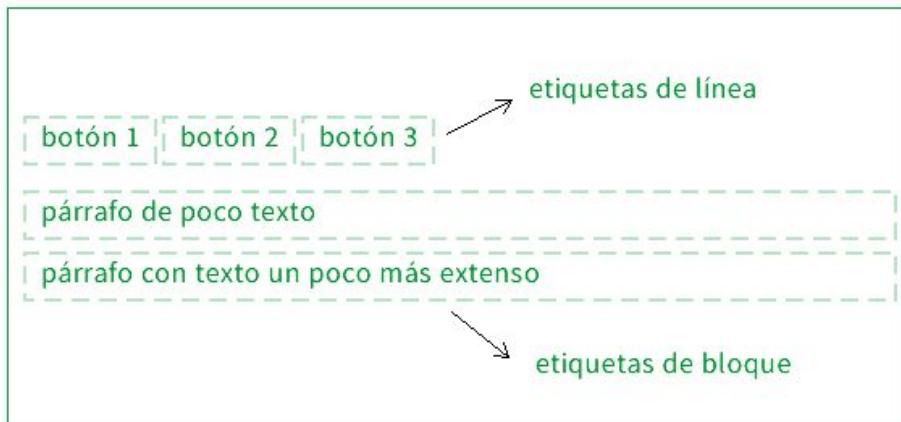


Momento de exposición teórica

PROPIEDADES DE LA CAJA

PROPIEDADES DE LA CAJA

Todos los elementos del HTML son cajas. Un ``, un `<h2>` y demás, son rectangulares:



- En los elementos de **línea**, se verá uno al lado del otro.
- En cambio en los de **bloque**, uno debajo del otro.

DIV

La etiqueta `<div>` **funciona y sirve de contenedor de bloque**, es decir puede contener uno o varios elementos HTML dentro de ella, es bastante utilizada para agrupar varios elementos que luego serán apuntados con CSS con la finalidad de dar un estilo adecuado a dicho grupo, o también puede servir para organizar mejor nuestra página web.

SPAN

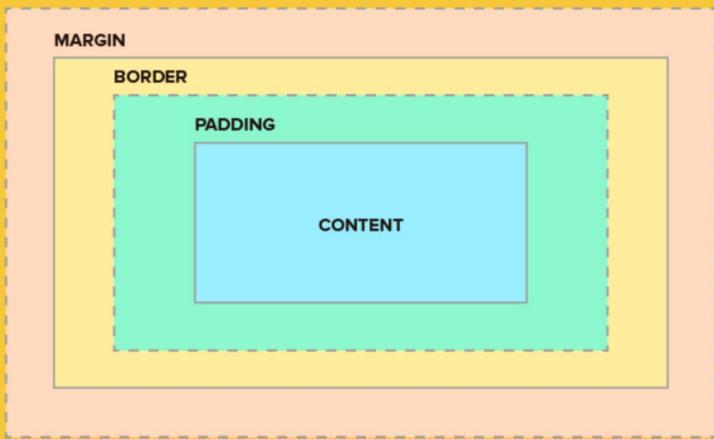
La etiqueta `` también funciona como un contenedor pero **en este caso será un contenedor en línea**. Por ejemplo para destacar una palabra o expresión dentro de un párrafo utilizaremos la etiqueta `` para encerrar esa palabra o expresión y luego ponerle un estilo mediante un atributo. O también podemos ponerle un identificador, ya sea id o class para luego apuntarlo y maquetarlo con CSS.

BOX MODEL

Ese concepto de que “todo es una caja”, da lugar a algo llamado en web como **box model**. Sin importar si son de línea o de bloque (pero tienen su incidencia en lo que sean), todas las etiquetas tienen propiedades en común.

PROPIEDADES EN COMÚN

Box Model Css



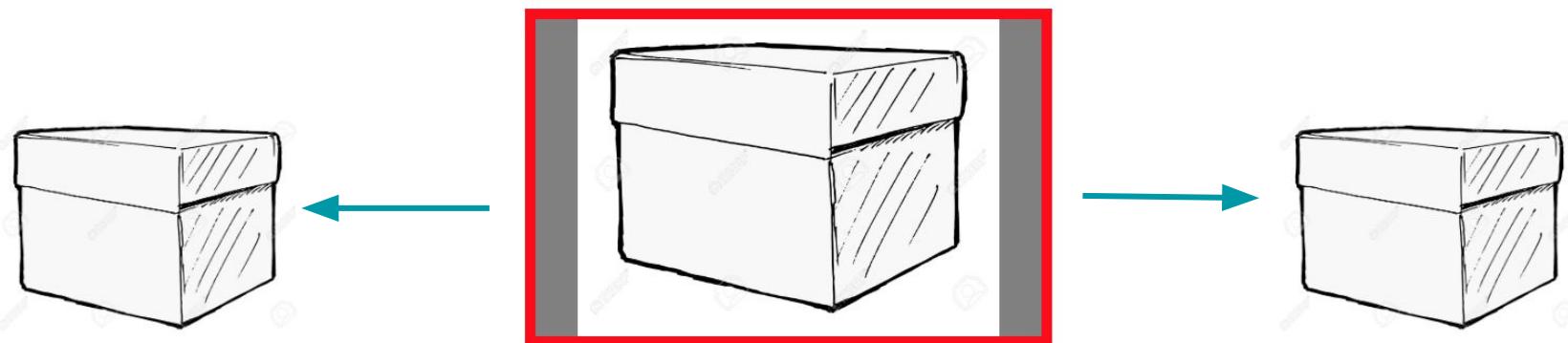
CONTENT: el espacio para el texto o imagen.

BORDER: el límite entre el elemento y el espacio externo.

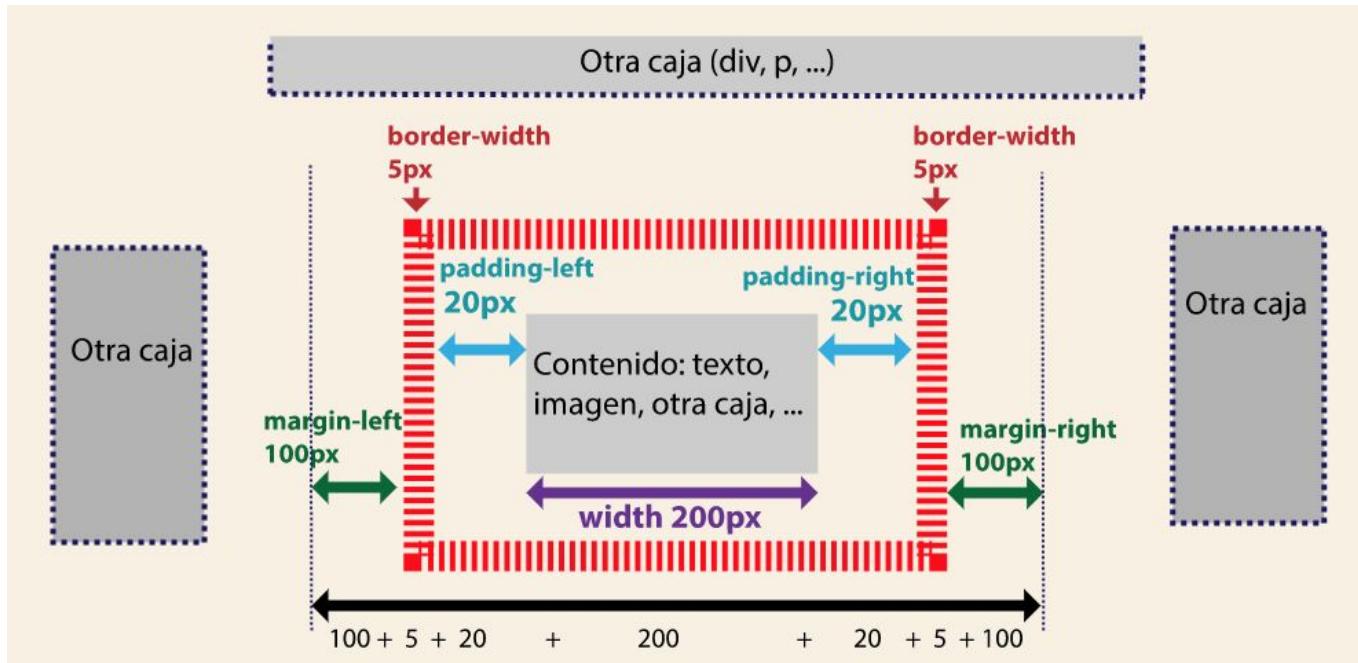
PADDING: separación entre el borde y el contenido de la caja. Es un espacio interior.

MARGIN: separación entre el borde y el afuera de la caja. Es un espacio exterior.

EJEMPLO



EJEMPLO



ALTO Y ANCHO

(de los elementos)

Ancho

Se denomina **width** a la propiedad CSS que controla la anchura de la caja de los elementos.

Dicha propiedad no admite valores negativos, y aquellos en porcentaje se calculan a partir de la anchura de su elemento padre.

Alto

La propiedad CSS que controla la altura de la caja de los elementos se denomina **height**.

No admite valores negativos, y aquellos en porcentaje se calculan a partir de la altura de su elemento padre.

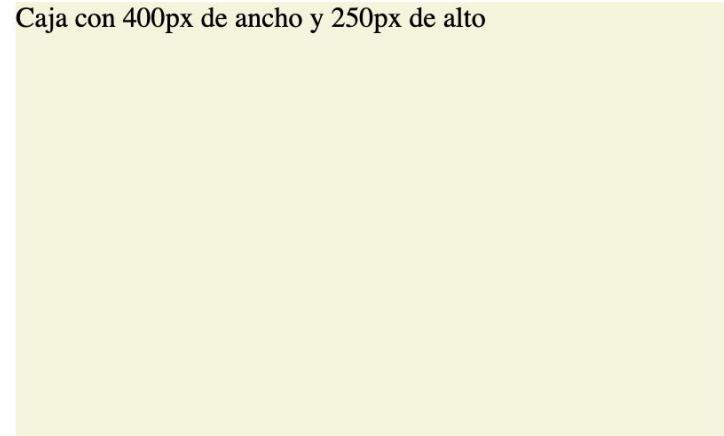
ALTO Y ANCHO

CSS

```
div {  
    background-color: beige;  
  
    width: 400px; /* ancho */  
    height: 250px; /* alto */  
}
```

Se ve así

Caja con 400px de ancho y 250px de alto



Valores comunes: unidad (px, porcentaje, rem, viewport) | [Ejemplos y más información](#).

PROBLEMA COMÚN

CSS

```
div {  
    background-color: beige;  
  
    /* no va a funcionar poner la  
    altura así, si su padre es body */  
    height: 100%;  
}
```

Como el padre es BODY, y no tiene una altura definida no podrá aplicarse un 100% ([ver más](#)).

Se ve así

 Lorem



???

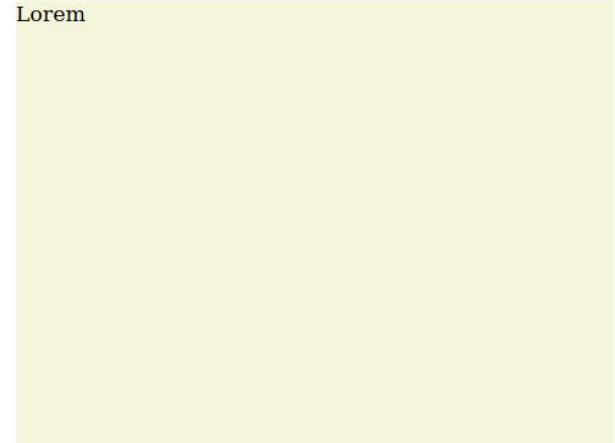
SOLUCIÓN

CSS

```
div {  
    background-color: beige;  
  
    /* al usar medida 'vh'  
     * (viewport height) */  
    height: 100vh;  
  
}
```

Se ve así

Lorem



Usando medidas “viewport” (tamaño de la ventana del navegador) es posible solucionarlo ([ver más](#)).

ALGO MÁS PARA ACLARAR

Cuando un elemento tiene un alto o ancho fijos, cualquier contenido que exceda la caja será visible. El inconveniente que esto genera es que, si luego se suma otro contenido, los mismos se van a superponer.



EJEMPLO

HTML

```
<div>  
    CSS IS <strong>AWESOME</strong>  
</div>
```



CSS

```
div {  
    /* propiedades decorativas */  
    border: solid 1px black;  
    padding: 5px;  
    display: inline-block;  
    font-size: 32px;  
    font-family: Arial;  
    /* propiedades que hacen el "problema" */  
    width: 100px;  
    height: 110px;  
}
```

OVERFLOW

Propiedad: **overflow**

Tiene 4 valores posibles:

- **visible**: valor por defecto. El excedente es visible.
- **hidden**: el excedente no se muestra (lo corta) → **recomendado**.
- **scroll**: genera una barra de scroll en los dos ejes (x/y) del elemento, aunque no se necesite.
- **auto**: genera el scroll solo en el eje necesario.

Veamos cómo se ve aplicando el **overflow: hidden**.

SOLUCIÓN

HTML

```
<div>  
    CSS IS <strong>AWESOME</strong>  
</div>
```



CSS

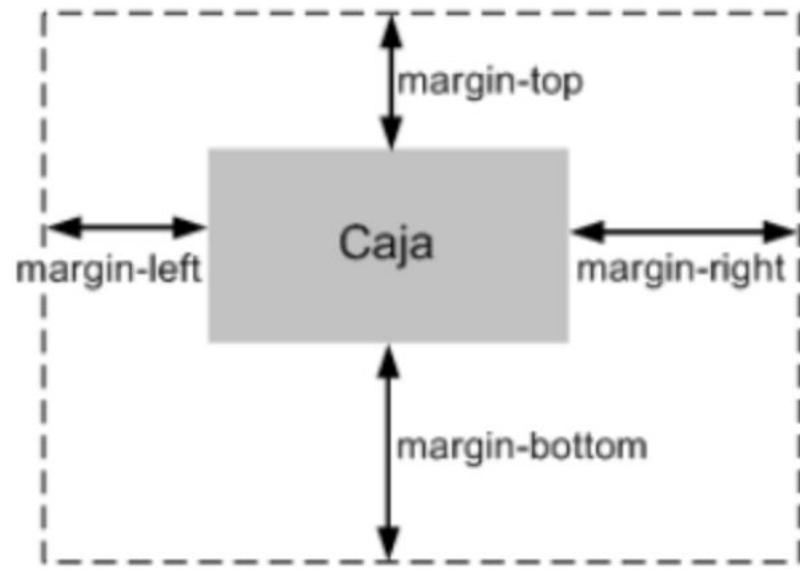
```
div {  
    /* propiedades decorativas */  
    border: solid 1px black;  
    padding: 5px;  
    display: inline-block;  
    font-size: 32px;  
    font-family: Arial;  
    /* propiedades que hacen el "problema" */  
    width: 100px;  
    height: 110px;  
    /* solucion */  
    overflow: hidden;  
}
```

ESPACIO EXTERIOR

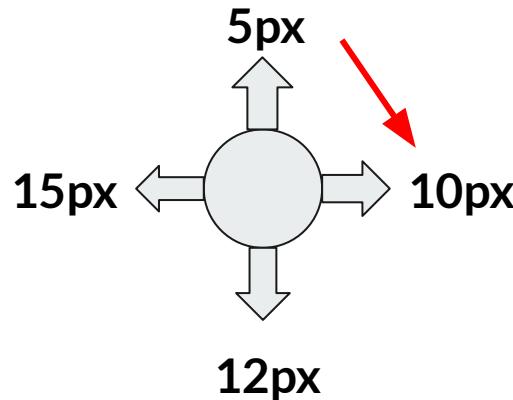
Margin (márgenes)

Las propiedades `margin-top`, `margin-right`, `margin-bottom` y `margin-left`, se utilizan para definir los márgenes de cada uno de los lados del elemento por separado.

Puedes definir los 4 lados (forma abreviada “`margin`”) o sólo aquellos que necesites.



CÓDIGO EJEMPLO



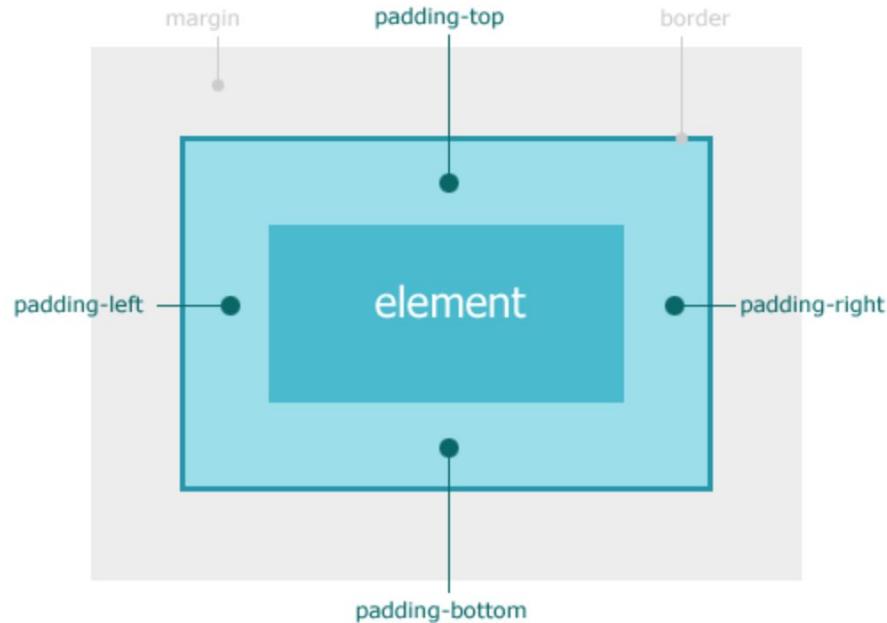
```
div {  
    margin-top: 5px;  
    margin-right: 10px;  
    margin-bottom: 12px;  
    margin-left: 15px;  
}  
  
/* forma abreviada pone en top,  
right, bottom, left */  
div {  
    margin: 5px 10px 12px 15px;  
}
```

ESPACIO INTERIOR

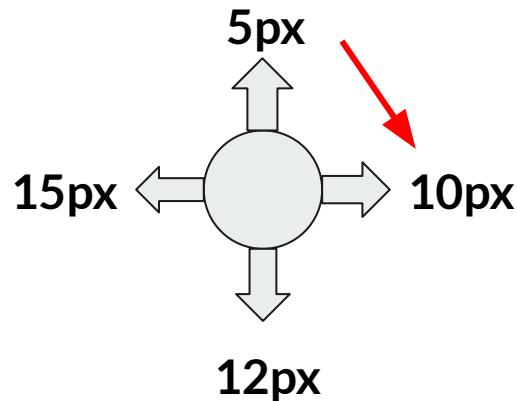
Padding (relleno)

Las propiedades `padding-top`, `padding-right`, `padding-bottom` y `padding-left`, se utilizan para definir los espacios internos de cada uno de los lados del elemento, por separado.

Puedes definir los 4 lados (forma abreviada “`padding`”) o sólo aquellos que necesites.



CÓDIGO EJEMPLO



```
div {  
    padding-top: 5px;  
    padding-right: 10px;  
    padding-bottom: 12px;  
    padding-left: 15px;  
}
```

/* forma abreviada */

```
div {  
    padding: 5px 10px 12px 15px;  
}
```

BORDES

Border

Las propiedades `border-top`, `border-right`, `border-bottom`, y `border-left`, se utilizan para definir los bordes de cada lado del elemento por separado.

Puedes definir los 4 lados (forma abreviada “`border`”) o sólo aquellos que necesites.

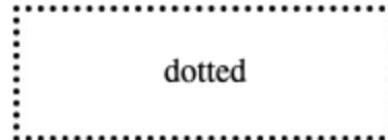


BORDES

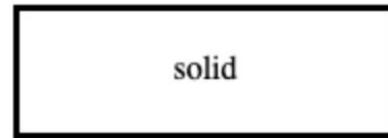
Nota

A diferencia de los márgenes y padding, los bordes se forman con 3 valores:

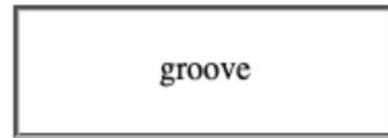
- Tipo de borde (border-style).
- Grosor (-width).
- Color (-color).



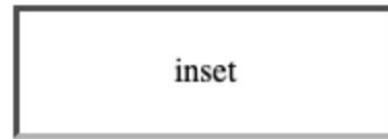
dashed



double



ridge



outset



none

hidden

BORDES

CSS

```
div {  
    border-top:solid 5px red;  
    border-right:solid 10px cyan;  
    border-bottom:solid 7px green;  
    border-left:solid 12px yellow;  
}
```

Se ve así



Valores comunes: estilo grosor color| [Ejemplos y más información](#)

DISPLAY

CODER HOUSE

TIPOS DE ELEMENTOS

- El estándar HTML clasifica a todos sus elementos en dos grandes grupos: elementos **en línea (*inline*)** y **de bloque (*block*)**.
- Los elementos de bloque siempre empiezan en una nueva línea, y ocupan todo el espacio disponible hasta el final de la misma (100%).
- Por otra parte, los elementos en línea no empiezan necesariamente en nueva línea y sólo ocupan el espacio necesario para mostrar sus contenidos.

TIPOS DE ELEMENTOS

BLOCK VS INLINE



display: block;

Block elements stack, regardless of their width.

display: inline;

Inline elements flow from one line to the next.

TIPOS DE ELEMENTOS

- Los elementos **en línea** definidos por HTML son aquellos que se usan para marcar texto, imágenes y formularios.
[Ver listado de etiquetas de “en línea”.](#)
- Los elementos **de bloque** definidos por HTML se utilizan para marcar estructura (división de información/código)
[Ver listado de etiquetas de en bloque](#)

DISPLAY

Se encarga de definir **cómo se ve un elemento HTML**. Los dos comportamientos más importantes son:

- Pasar un elemento de bloque a uno de línea.
- Pasar un elemento de línea a uno de bloque.

Eso se hace con los valores ***block*** e ***inline*** respectivamente:

- ***block*:** convierte el elemento en uno de bloque.
- ***Inline*:** transforma el elemento en uno de línea.

DISPLAY

HTM
L

```
<p>Lorem ipsum dolor sit  
amet, consectetur  
adipisicing elit.  
<span>Laudantium </span>  
perspiciatis itaque  
veritatis ea fugit qui.  
</p>
```

CSS

```
p /*es un elemento en bloque que  
convierzo en línea*/  
    display: inline;  
    background-color: yellow;}  
span /*es un elemento en línea que  
convierzo en bloque*/  
    display: block;  
    background-color: grey;}
```

Con este ejemplo podemos verificar cómo modifíco el display de las etiquetas, puedes probar más [acá](#).

DISPLAY

Inline-block

Hay una propiedad que permite tomar lo mejor de ambos grupos, llamada “*inline-block*”. Brinda la posibilidad tener “*padding*” y “*margin*” hacia arriba y abajo.

```
li {  
    display: inline-block;  
}
```

Haz clic [aquí](#) para ver más ejemplos.

TABLA COMPARATIVA

Dependiendo de si la etiqueta de HTML es “**de bloque**” o “**en línea**”, algunas propiedades serán omitidas ([más información](#)).

	Width	Height	Padding	Margin
Bloque	SI	SI	SI	SI
En línea	NO	NO	Solo costados	Solo costados
En línea y bloque	SI	SI	SI	SI

QUITAR UN ELEMENTO

El display tiene también un valor para quitar un elemento del layout

display: none; Lo oculta, y además lo quita (no ocupa su lugar).

```
div {  
    display: none;  
}
```

POSICIONES

POSITION

Es una propiedad CSS pensada para ubicar un elemento, con una libertad muy flexible. Algunos ejemplos de uso:

- Superponer elementos.
- Crear publicidades que te sigan con el scroll o un menú.
- Hacer un menú con submenú adentro.

Valores posibles: relative, absolute, fixed, o sticky (cualquiera excepto static).

¿CÓMO UBICAR UN ELEMENTO?



Define qué tipo de posición quieras usar.



Indica desde dónde calcular la distancia (si será desde arriba, derecha, abajo o izquierda).



Determina un valor numérico para las propiedades ***top***, ***bottom***, ***left***, ***right***.

¿Recuerdas cuando ubicábamos los fondos? Similar, tiene algunas particularidades.

POSITION

Al aplicar esta propiedad, puedes usar 4 propiedades para posicionar los elementos, y debes darles un valor numérico. Ellas son:

- **top:** calcula desde el borde superior (ej: top: 100px).
- **right:** calcula desde el borde derecho (ej: right: 50px).
- **bottom:** calcula desde el borde inferior (ej: bottom: 100px).
- **left:** calcula desde el borde izquierdo (ej: left: 50%).

Haz clic [aquí](#) para acceder a más información.

POSITION: RELATIVE

El elemento es posicionado de acuerdo al flujo normal del documento, y luego es **desplazado en relación a sí mismo**.

El desplazamiento no afecta la posición de ningún otro elemento, provocando que se pueda superponer sobre otro.

POSITION: RELATIVE

CSS

```
div {  
    width: 100px;  
    height: 100px;  
  
    position: relative;  
    top: 40px;  
    left: 40px;  
}
```

Se ve así



Lorem ipsum dolor sit, amet,
consectetur adipiscing elit. Sed
tortor leo, ultrices eget libero eu.

POSITION: ABSOLUTE

El elemento es removido del flujo normal del documento, sin crearse espacio alguno para el mismo en el esquema de la página.

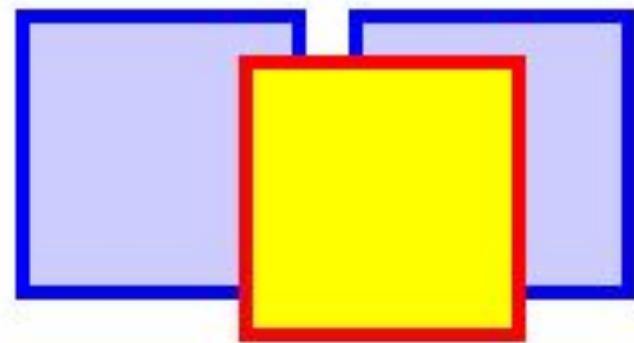
Es posicionado relativo a su padre, siempre y cuando su padre tenga “position:relative”. De lo contrario, se ubica relativo al body. Se recomienda establecer un ancho y alto (width, height).

POSITION: ABSOLUTE

CSS

```
div {  
    width: 100px;  
    height: 100px;  
  
    position: absolute;  
    top: 40px;  
    left: 40px;  
}
```

Se ve así



POSITION: FIXED Y STICKY

Ambos métodos permiten que el elemento se mantenga visible, aunque se haga scroll.

FIXED

Esta posición es similar a la absoluta, con la excepción de que el elemento contenedor es el “viewport”, es decir, la ventana del navegador.

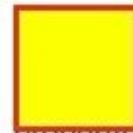
Puede ser usada para crear elementos que floten, y que queden en la misma posición aunque se haga scroll.

FIXED

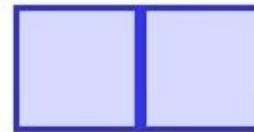
CSS

```
div {  
    width: 300px;  
    background-color: yellow;  
  
    position: fixed;  
    top: 0;  
    left: 0;  
}
```

Se ve así



mo you can control the position property for the yellow box.



To see the effect of sticky positioning, select the position: sticky option and scroll this container.

STICKY

El elemento es posicionado en el “flow” natural del documento, podría decirse que es un valor que funciona de forma híbrida, es decir, como “relative” y también “fixed”.

Esto es, cuando llega el “viewport” (la ventana del navegador) hasta donde se encuentra, se “pegará” sobre el borde superior.

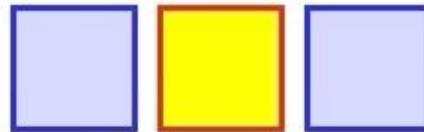
STICKY

CSS

```
div {  
    position: sticky;  
    top: 20px;  
}
```

Se ve así

In this demo you can control the `position` property for the yellow box.



To see the effect of sticky positioning, select the `position: sticky` option and scroll this container.

PROPIEDAD Z-INDEX

(para el orden de superposición)

El z-index entra en juego cuando dos elementos que tienen *position* se superponen. Esta propiedad acepta como valor un número (sin ninguna unidad, ni px, ni cm, ni nada); a valor más alto, se mostrará por encima de los demás elementos.

Por defecto, todos los objetos tienen z-index:1. Si dos objetos tienen el mismo valor de z-index y se superponen, el que fue creado después en el HTML se verá encima del otro.



BREAK

¡5/10 MINUTOS Y VOLVEMOS!

Ejemplo
en vivo



¡VAMOS A PRACTICAR LO VISTO EN LOS BREAKOUT ROOMS!

Momento de consolidación de aprendizajes

EXTRAS

CODER HOUSE

BUG DEL DISPLAY: INLINE-BLOCK

El display *inline-block* con ancho fijo pone las cosas una al lado de la otra (sí, lo dijimos recién). Pero si tienes los anchos milimétricamente calculados, puede ser que el último aparezca abajo (y no al lado).

Si entre los elementos de línea (inline o inline-block) hay “aire” (sea uno o 500 enter, espacios o tab), el mismo se muestra como un espacio de barra espaciadora. Esa es la razón por la cual no entran, y se muestra uno debajo de los demás.

MENÚ CON DISPLAY

HTML

```
<ul>
  <li><a href="#home"
    class="activo">
    Home</a></li>
  <li><a href="#nosotros">
    Nosotros</a></li>
  <li><a href="#contacto">
    Contacto</a></li>
</ul>
```

CSS

```
ul {
  list-style-type: none;
  overflow: hidden;
  background-color: #333;}
li {
  float: left; }
li a {
  display: inline-block;
}

.activo {
  background-color: blue;}
```

MENÚ CON SUBMENÚ

- El position (tanto *relative* como *absolute*) se usa, entre otras, para hacer un menú que tenga un submenú emergente. Los ítems del primero son relativos, sirven como borde de cualquier hijo.
- La lista dentro de un list-item es absoluta. Por defecto, la sublista tiene *display: none*. Recién cuando un list-item detecte el *:hover*, si adentro tiene una lista, dale *display: block* (*no te preocupes, esto quedará quedará más claro con el ejemplo que veremos a continuación*).

MENÚ CON SUBMENÚ



Por defecto la lista es display none
Cuando se pasa el mouse por arriba
del li se muestra la sub-lista con
display block

```
ul li:hover ul{ display: block }
```

Este list-item es relative

La lista es
absolute.

Como está en
el relative,
lo usará como
límites del top,
bottom, left y
right

HTML

```
<ul>
  <li><a href="">Item</a></li>
  <li><a href="">Item</a></li>
  <li><a href="">Item</a>
    <ul>
      <li><a href="">Subitem</a></li>
      <li><a href="">Subitem</a></li>
    </ul>
  </li>
  <li><a href="">Item</a></li>
</ul>
```

CSS

```
ul {
  list-style: none;
  font-size: 0 /* truco por el uso de inline-block*/
}
li {
  display: inline-block;
  width: 25%;
  position: relative;
  font-size: 14px
}
ul ul {
  position: absolute;
  display: none;
}
ul ul li {
  display: block;
}
ul li:hover ul {
  display: block;
}
```



BOX MODEL

Incluye box model en tu proyecto final.

BOX MODEL

Formato: archivo HTML y CSS. Debe tener el nombre “Idea+Apellido”.

Sugerencia: carpeta en formato ZIP o RAR, con el o los archivos HTML y CSS.

Desafío
Complementario



>> Consigna: agrega al CSS del Proyecto Final (¿Cuál necesitan: espacio exterior, interior, bordes?).

- Márgenes.
- Rellenos.
- Bordes.
- Menú.

>>Aspectos a incluir en el entregable: modifica el menú con las propiedades de Display. Sumar márgenes, rellenos y bordes a las secciones dentro de las páginas. No necesariamente tienen que estar las tres propiedades a la misma sección, pueden estar aplicadas a diferentes secciones diferentes propiedades.

>>Ejemplo: [Carpeta comprimida con los archivos de box modeling](#)

¿PREGUNTAS?

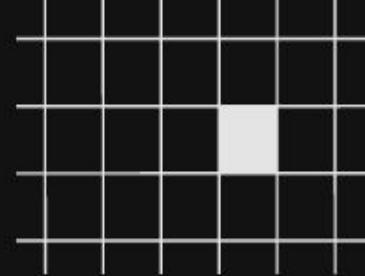




**¿QUIERES SABER MÁS? TE DEJAMOS
MATERIAL AMPLIADO DE LA CLASE**



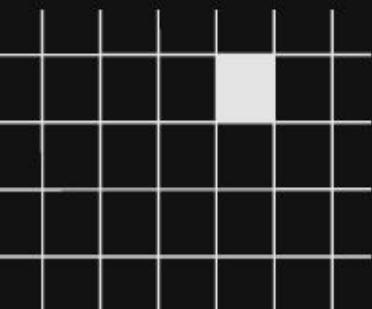
- [Imágenes de relleno](#) | ***placekitten***
- [Imágenes de relleno](#) | ***placedog.net***
- [The CSS Box Model](#) | **CSS-TRICKS**



¡MUCHAS GRACIAS!

Resumen de lo visto en clase hoy:

- Cajas y sus propiedades.
- Modelación de la página web.
- Posiciones de un elemento.





OPINA Y VALORA ESTA CLASE