

HTML y CSS

Contenidos de la Clase

- Divs y Spans
- Imágenes
- Unidades de Medida
- Width y Height
- Box-modeling
- Box-sizing
- Contenedores Límites
- Técnicas de Centrado

Elemento DIV

Se utiliza como un contenedor genérico para agrupar y organizar otros elementos en bloques.

Se utiliza para dividir y agrupar secciones del contenido de una manera flexible y se puede aplicar estilos a través de selectores CSS.

```
<div class="claseDiv">  
  <h2>Título del Contenido</h2>  
  <p>Texto descriptivo...</p>  
  <ul>  
    <li>Elemento 1</li>  
    <li>Elemento 2</li>  
  </ul>  
</div>
```

<https://developer.mozilla.org/es/docs/Web/HTML/Element/div>

Elemento SPAN

Se utiliza para aplicar estilos o realizar manipulaciones específicas en partes individuales de un texto o contenido dentro de un elemento más grande. A diferencia de <div>, no genera un salto de línea y se considera un elemento de línea.

Se utiliza para aplicar estilos o manipular partes específicas del texto, como cambiar el color, aplicar un resaltado o agregar interactividad.

```
<div class="claseDiv">  
  <p>El <span class="resaltado">texto destacado</span>  
  es importante en este contexto.</p>  
</div>
```

<https://developer.mozilla.org/es/docs/Web/HTML/Element/span>

Imágenes

Para poner una imagen simple en una página web, utilizamos el elemento ``. Se trata de un elemento vacío (lo que significa que no contiene texto o etiqueta de cierre) que requiere del atributo `src` que contiene una ruta que apunta a la imagen que querés poner en la página. Esta URL puede ser relativa o absoluta.

Otro atributo útil para las imágenes es `alt`. Su valor debe ser una descripción textual de la imagen para usarla en situaciones en que la imagen no puede ser vista/mostrada o tarde demasiado en mostrarse por una conexión lenta a internet.

Imágenes

El atributo **width** establece la anchura de visualización de la imagen, y el atributo **height** establece la altura de visualización de la imagen. No son obligatorios, si no aparecen la imagen tendrá su altura y anchura original. Si aparecen, modificarán el aspecto de la imagen pero no afectarán al archivo de imagen, es decir, sólo afectarán a la visualización que realiza el navegador de la imagen.

```

```

Unidades de Medida

Tenemos unidades absolutas y relativas.

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.

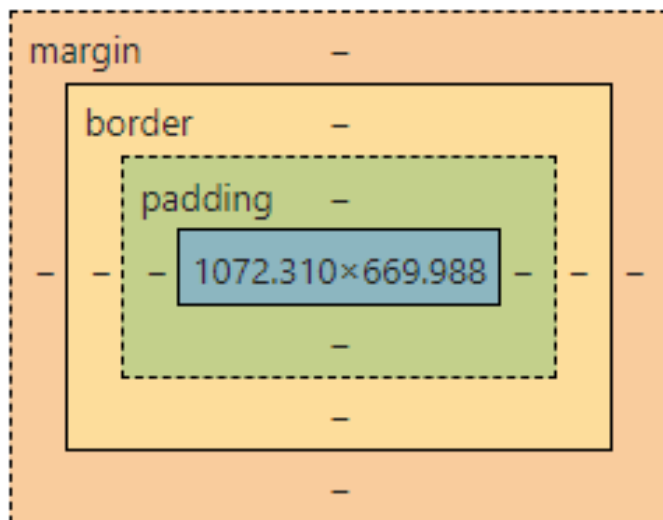
Las más convenientes para los textos son las unidades relativas, ya que al cambiar el tamaño de la ventana, cambiarán los tamaños de las letras.

Box Modeling

Todos los elementos del HTML son cajas rectangulares.

Recordemos que tenemos elementos...

- ❑ De línea: se verán uno al lado del otro, ocupan el espacio necesario.
- ❑ De bloque: se verán uno debajo del otro y ocupan todo el espacio disponible.



Box model: Todas las etiquetas tienen propiedades en común.

- Content: espacio para el texto o imágenes.
- Border: límite entre el elemento y el espacio externo.
- Padding: separación entre el borde y el contenido de la caja, espacio interior.
- Margin: separación entre el borde y el exterior de la caja, espacio exterior.

Box-sizing - content-box

Por defecto el box modeling calcula el tamaño total de un elemento sumando el ancho o alto de contenido coma el padding y el borde. Podemos cambiar esto utilizando la propiedad “**box-sizing**” para especificar cómo se debe calcular el tamaño total del elemento. Para esto necesitamos 2 valores principales:

- “**content-box**”: Este es el comportamiento por defecto. El tamaño total del elemento se calcula sumando el ancho o alto del contenido, el padding y el borde.

```
.box {  
  box-sizing: content-box;  
  width: 300px;  
  padding: 10px;  
  border: 1px solid black #000;  
}
```

Box-sizing – border-box

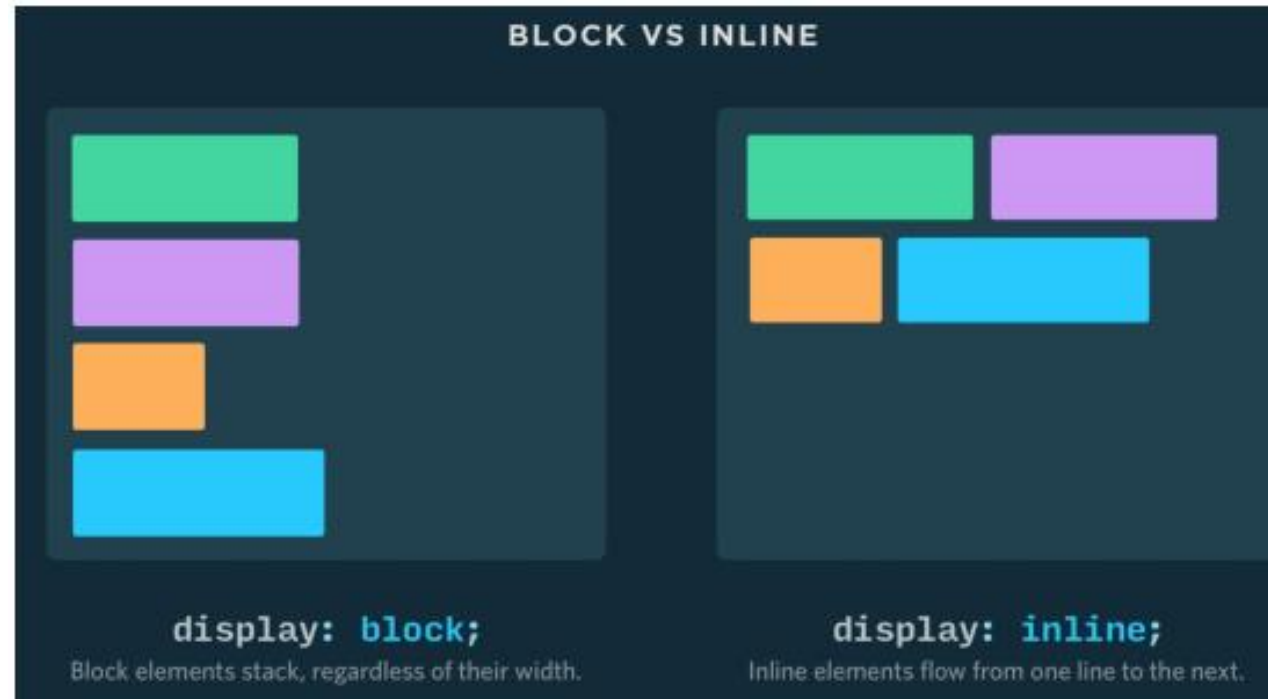
- “border-box”: Con este valor, el tamaño total del elemento se calcula incluyendo el contenido, el padding y el borde.

El contenido se ajustará automáticamente para que se ajuste al espacio disponible.

```
.box {  
    box-sizing: border-box;  
    width: 300px;  
    padding: 10px;  
    border: 1px solid black;  
}
```

<https://developer.mozilla.org/es/docs/Web/CSS/box-sizing>

Block vs Inline



Se usan para marcar texto, imágenes y formularios.

[Listado](#)

Se usan para marcar estructura, es decir, una división de información o código.

[Listado](#)

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 → “**inline**”
- Pasar un elemento de línea a uno de bloque → “**block**”

“**inline-block**”: permite tomar lo mejor de ambos grupos, tenemos padding y margin hacia arriba y abajo.

	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

“**display: none**” oculta y quita un elemento del layout, es decir, no ocupa su lugar.

Para profundizar: [“position”](#)

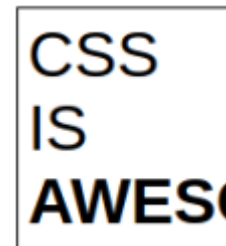
Overflow

Tiene 4 valores posibles:

- ❖ **visible**: valor por defecto (el excedente se ve).
- ❖ **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 solamente en el eje necesario.

```
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;
}
```

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



Contenedores Límites

Son elementos utilizados para restringir el ancho máximo de su contenido y **evitar que se extienda más allá de un límite establecido**.

Estos contenedores son útiles en diseños responsivos, donde se desea controlar el ancho máximo de ciertos elementos para garantizar una experiencia de usuario adecuada en diferentes dispositivos y tamaños de pantalla (*más adelante veremos responsive*).

Ejemplo de contenedor límite:

```
.container {  
  max-width: 800px;  
  margin-left: auto;  
  margin-right: auto;  
}
```

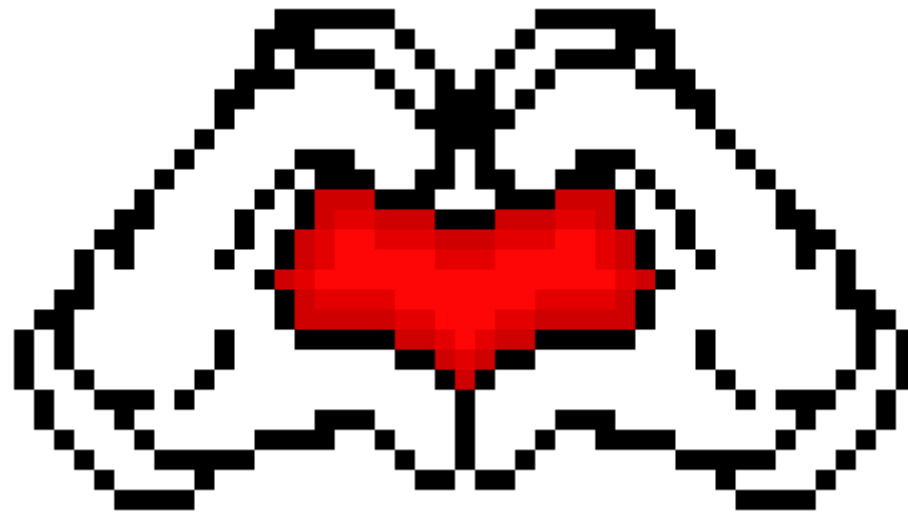
```
<div class="container">  
  <!-- Contenido dentro del contenedor -->  
</div>
```

Veamos algunas técnicas de centrado en vsCode...

¿Preguntas?



¡Nos vemos la próxima clase!



BA MULTIPLICA 2.0

jóvenes  jóvenes



UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

