

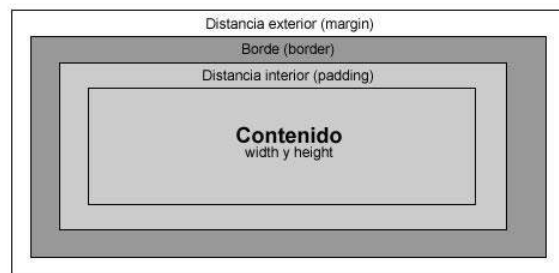
1.1 Modelo de cajas

Por defecto Las propiedades **width** y **height** de la caja solo afectan al contenido, al tamaño total hay que sumarle el padding, el border y el margin. Esto es porque por defecto la propiedad **box-sizing:** tiene el valor **content-box**.

Para incluirlos hay que utilizar la propiedad **box-sizing: border-box;**

1.2 Float y Clear

La propiedad **float** se usa, para dejar que una imagen flote hacia la izquierda en el texto en un contenedor.



- **left:** el elemento flota a la izquierda de su contenedor
- **right:** el elemento flota a la derecha de su contenedor

- Dejan su posición que es ocupada por los siguientes
- Entre elementos flotantes si se respetan su espacio

La propiedad **clear** indica que el elemento no debe tener a su izquierda (**left**) o derecha, elementos flotantes.

- **clear:left** (no quiero elementos flotantes a mi izquierda, lo mismo **clear:right**).
- **clear: both:** (ambos lados despejados) Esto obliga si tiene elementos flotantes a pasarse debajo

Los molinos de viento

float: left



Los molinos de viento

float: left float: left



1.3 Position

static	<ul style="list-style-type: none"> • Valor por defecto. Comportamiento normal (de izquierda a derecha y de arriba abajo)
relative	<ul style="list-style-type: none"> • Mueve un elemento (div, h1, p etc.) en relación a donde se dispondría en el flujo normal. • Se mueve con las propiedades top, left, bottom, right • El espacio que ocupaba antes de desplazarse no es ocupado por otro
absolute	<ul style="list-style-type: none"> • El posicionamiento es siempre relativo el primer elemento contenedor padre con posicionamiento NO STATIC. En otro caso es relativo la esquina superior izquierda del documento (body) • El espacio que ocupaba antes de desplazarse SÍ es ocupado por otro • Si se superponen se pueden ordenar con la propiedad z-index
fixed	<ul style="list-style-type: none"> • Ajusta el elemento en una coordenada fija con respecto a la ventana del navegador. (si se hace scroll no se mueve) • No reserva el espacio del elemento. (es como existirá y es ocupado por otros) • Ocupa por defecto solo el tamaño de su contenido
sticky	<ul style="list-style-type: none"> • Ajusta el elemento en una coordenada fija con respecto a la ventana del navegador. (si se hace scroll no se mueve) IGUAL QUE EL fixed • SÍ reserva el espacio del elemento. (s ocupado por si no otros)

1.4 Display:block y display:inline. visibility

display: Cada elemento HTML tiene un valor de visualización predeterminado según el tipo de elemento que sea. El valor de la propiedad **display** predeterminado para la mayoría de los elementos es **block** o **inline**.

- **block:** siempre comienza en una nueva línea y ocupa todo el ancho disponible.
- **inline;** no comienza en una nueva línea y solo ocupa el ancho que sea necesario.

- Con **display: block** o **display: inline** podemos cambiar el comportamiento a uno u otro
- **none**; se usa comúnmente con JavaScript para ocultar y mostrar elementos sin eliminarlos. A diferencia con **visibility: hidden**, no guarda el espacio mientras que **visibility: hidden** sí.

1.5 Display: inline-block

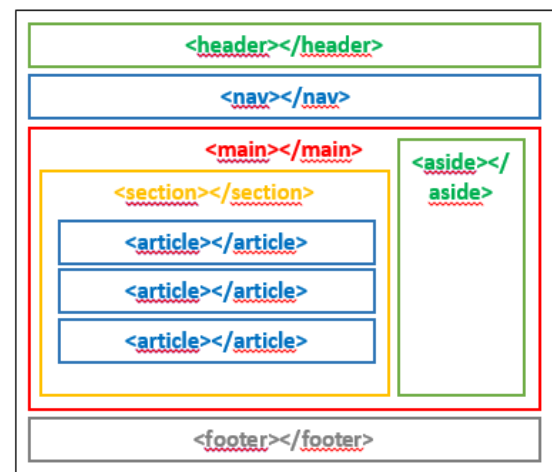
En comparación con **display: inline**, la diferencia principal es que **display: inline-block** permite establecer una anchura y altura en el elemento.

Además, con **display: inline-block** **margin** y **padding** son respetados, pero con **display: inline** no lo son.

1.5.1 Usar display: inline-block para mostrar los elementos de una lista horizontalmente

HTML	
<pre><div id="menu"> Quienes somos Qué hacemos Donde estamos </div></pre>	<pre>#menu { background-color: yellow; list-style-type: none; padding: 0; margin: 0; } #menu li { display: inline-block; font-size: 20px; padding: 20px; }</pre>

1.6 Etiquetas semánticas HTML5 de maquetación



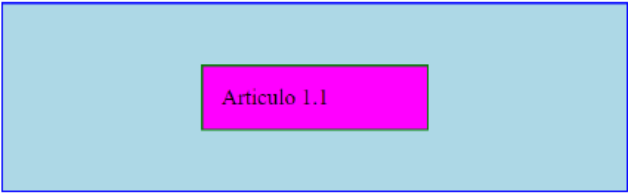

Hasta ahora las divisiones de un sitio web se realizaba mediante la etiqueta **table** o mediante **div**, pero estas etiquetas no proveen ninguna información sobre el contenido de dicha división. HTML 5 cambia la forma en que se hace la maquetación e incorpora las etiquetas semánticas:

header:
footer:
main:
aside:
section:
article:
nav:
figure:

1.7 Flexbox

HTML	CSS	Propiedad justify-content : Alineación horizontal (por defecto)
<pre><section> <article id="art1"> Artículo 1.1 </article> <article id="art12"> Artículo 1.2 </article> <article id="art13"> Artículo 1.3 </article> </section></pre>	<pre>section {display: flex; background-color: lightblue; border: 2px solid blue; justify-content: flex-start;} article { border: 2px solid green; background-color: red; padding: 15px;}</pre>	<p>justify-content: flex-start</p>
	justify-content: flex-end	
	justify-content: center	
	justify-content: space-around	
	justify-content: space-between	

		Propiedad align-items : Alineacion vertical (por defecto)
<pre><section> <article id="art11"> Artículo 1.1 </article> <article id="art12"> Artículo 1.2 </article> <article id="art13"> Artículo 1.3 </article> </section></pre>	<pre>section {display:flex; background-color: lightblue; border:2px solid blue; height: 100px; justify-content: center; align-items: flex-start }</pre>	
	align-items: flex-end	
	align-items: center	
	align-items: stretch	
Propiedad align-self : Alineacion vertical INDIVIDUAL de cada item		
	<pre>article#art11 { align-self: flex- end } article#art12 { align-self: center } article#art13 { align-self: flex- start}</pre>	
Propiedad flex-wrap : por defecto el valor es nowrap (todo en la misma línea. si el valor es wrap : se crean las filas necesarias para mantener el ancho indicado		
	<pre>section {display:flex; background-color: lightblue; border:2px solid blue; justify-content: center; align-items: baseline; width: 100px; flex-wrap: wrap; align-content : center</pre>	flex-wrap:wrap; align-content:center 
	<pre>flex-wrap: wrap; align-content : space-around</pre>	flex-wrap: wrap; align-content : space-around 
	<pre>section {display:flex; background-color: lightblue; border:2px solid blue; justify-content: center; align-items: baseline; height: 100px; flex-wrap: wrap; align-content : flex-start</pre>	flex-wrap:wrap; align-content:flex-start 

<p>Con una caja</p> <pre><section> <article> Artículo 1.1 </article> </section></pre>	<pre>section {display:flex; background-color: lightblue; border:2px solid blue; justify-content: center; align-items: center; height: 100px;</pre>	<p>Centrar horizontal y verticalment un objeto</p>  <p>(no hace falta que esté dentro de una capa, funciona con textos e imágenes)</p> 
<p>Con un párrafo:</p> <pre><section> <p>Esto es un párrafo</p> </section></pre>		

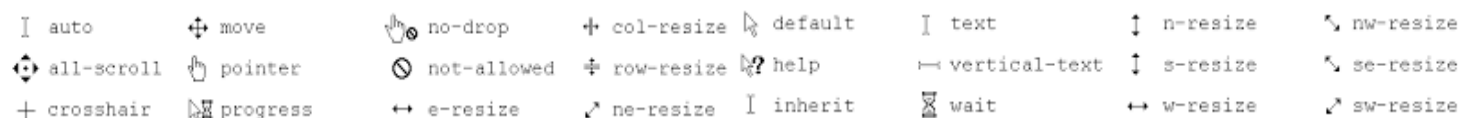
1.8 Overflow

overflow especifica si se debe recortar el contenido o agregar barras de desplazamiento cuando el contenido de un elemento es demasiado grande para caber en el área especificada. tiene los siguientes valores:

- **visible**- Defecto. El contenido se representa fuera del cuadro del elemento.
- **hidden** - El desbordamiento se recorta y el resto del contenido será invisible.
- **scroll** - El desbordamiento se recorta y se agrega una barra de desplazamiento para ver el resto del contenido
- **auto**- Similar a **scroll**, pero agrega barras de desplazamiento solo cuando es necesario

1.9 Punteros del ratón (cursor)

cursor especifica el cursor del mouse que se mostrará al apuntar sobre un elemento ej: **cursor: pointer;**



1.1 Transiciones

transition-property: (qué propiedad css es la que va a cambiar, width, color....)

transition-duration; duración en segundos de la transición

transition: (propiedad abreviada) *propiedad duración*

Normalmente la propiedad se cambia con la pseudoclase **:hover**

```
div {
width: 100px;
height: 100px;
background: red;
-webkit-transition: width 2s;
/* For Safari 3.1 to 6.0 */
transition: width 2s;
/* o también
transition-property:width;|
transition-duration:2s */
}
div:hover {
width: 300px;
}
```

1.2 Sombras

a) Sombras de texto(propiedad **text-shadow**):

```
h1 { color: white;
text-shadow: 2px 2px 4px black;}
```

Desplazamiento-x Color de la sombra

Desplazamiento-y Desenfoque

efecto de sombra de texto!

b)Sombras de caja(propiedad **box-shadow**):

```
div { box-shadow: 10px 10px 5px grey;}
```

Este es un elemento de color amarillo <div> con una borrosa, gris box-shadow

1.3 Opacidad (opacity)

La propiedad **opacity** puede tener valor comprendido entre 0,0 - 1,0. El valor más bajo, el más transparente. Se suele variar con la pseudoclase hover:

```
img { opacity: 0.5; } img:hover {
opacity: 1.0; } /* la imágenes recuperan el color
al pasar el ratón */
```



IES JUAN BOSCO
www.iesjuanbosco.es

I.E.S. "JUAN BOSCO"

Familia Profesional de Informática



UT1: Guión CSS MEDIO. V1.0

Página 5 de 5