

16. TABLAS EN HTML

Una tabla en un conjunto de celdas organizadas dentro de las cuales podemos alojar distintos contenidos.

Las tablas son definidas por las etiquetas `<table>` y `</table>`.

Dentro de estas dos etiquetas colocaremos todas las otras etiquetas, textos e imágenes que darán forma y contenido a la tabla.

Las tablas son descritas por líneas de izquierda a derecha. Cada una fila es definida por otra etiqueta y su cierre: `<tr>` y `</tr>`

Dentro de cada fila, habrá diferentes celdas. Cada una de estas celdas será definida por otro par de etiquetas: `<td>` y `</td>`. Dentro de estas etiquetas será donde coloquemos nuestro contenido.

Ejemplo de estructura de tabla:

```
<table>
<tr>
<td>Celda 1, linea 1</td>
<td>Celda 2, linea 1</td>
</tr>
<tr>
<td>Celda 1, linea 2</td>
<td>Celda 2, linea 2</td>
</tr>
</table>
```

El resultado:

Celda 1, linea 1 Celda 2, linea 1

Celda 1, linea 2 Celda 2, linea 2

Tablas en HTML. Atributos para tablas, filas y celdas.

Bordes	<code>border</code>	<code>1px solid black;</code>
	<code>border-color</code>	<code>black</code>
	<code>border-width</code>	<code>1px</code>
	<code>border-style</code>	<code>solid, dashed, dotted, double, groove, ridge, inset, outset, hidden, none</code>
Bordes sencillos	<code>border-collapse</code>	<code>collapse</code> (para table. Borde sencillo)
Ancho	<code>width</code>	<code>100% 400px</code>
Alto	<code>height</code>	<code>50px</code>

Otros atributos que pueden ser únicamente asignados a una celda (`<td>` o `<th>`)

Alineación horizontal	<code>text-align</code>	<code>left right center</code>
Alineación vertical	<code>vertical-align</code>	<code>bottom top middle</code>
Espacio entre el borde y el contenido	<code>padding</code>	<code>15px</code>
Color de fondo	<code>background-color</code>	<code>pink</code>
Espacio entre las celdas	<code>border-spacing</code>	<code>10px</code>

ejemplotabla

17. LISTAS

Las listas, que sirven para enumerar y definir elementos, los textos preformateados y las cabeceras o títulos.

Las listas son utilizadas para citar, numerar y definir objetos. También son utilizadas corrientemente para desplazar el comienzo de línea hacia la derecha.

Son delimitadas por las etiquetas `` y ``

Cada uno de los elementos de la lista es citado por medio de una etiqueta ``

```
<p>Países del mundo</p>
```

```
<ul>
```

```
  <li>Argentina</li>
```

```
  <li>Perú</li>
```

```
  <li>Chile</li>
```

```
</ul>
```

El resultado:

Países del mundo

- Argentina
- Perú
- Chile

Podemos definir el tipo de viñeta empleada para cada elemento.

list-style-type: disc, circle, square, decimal, lower-alpha, lower-roman, upper-alpha, upper-roman

list-style-image: url(imagen.gif), none

list-style-position: outside, inside

Ejemplo

```
<style>
```

```
  ul {
```

```
    list-style-type:square;
```

```
</style>
```

```
<ul>
```

```
<li>Elemento 1 </li>
```

```
<li>Elemento 2 </li>
```

```
<li>Elemento 3 </li>
```

```
<li>Elemento 4 </li>
```

```
</ul>
```

Otro ejemplo:

```
<style>
```

```
  u {
```

```
    list-style-type:decimal;
```

```
</style>
```

```
<p>Bebida a elegir:</p>
```

```
<ul>
```

```
  <li>Café</li>
```

```
  <li>Tea</li>
```

```
  <li>Colacao</li>
```

```
</ul>
```

ejemplotablalista

18.POSICIONAMIENTO

display: inline, block, none

inline: muestra los elementos en línea

block: muestra los elementos en bloque añadiendo un salto de línea

inline-block: muestra los elementos en bloque en la misma línea

none: oculta

Un elemento de bloque es un elemento que ocupa todo el ancho completo disponible, y tiene un salto de línea antes y después de ella.

Ejemplos de elementos de bloque:

- `<h1>`
- `<p>`
- `<div>`

Un elemento en línea sólo se ocupa el mismo ancho que sea necesario, y no fuerza saltos de línea.

Ejemplos de elementos en línea:

- ``
- `<a>`

Ej6display

float: left, right

left: la caja se mueve a la posición más a la izquierda posible y el resto de elementos fluyen a su derecha.

right: igual.

Ej6menufloat

clear: left, right, both

Rompe el float left, right o ambos.

overflow: visible, hidden, scroll

Indica lo que se hace con el contenido que excede de las dimensiones de una caja.

visible: Los contenidos se salen del elemento y son visibles.

hidden: Oculta los elementos que se salen de la caja.

scroll: Muestra las barras de desplazamiento cuando los contenidos no caben en la caja.

auto: Cada navegador decide si es necesario mostrar las barras de scroll.

z-index: número

Profundidad de las cajas. A mayor número, la caja se muestra por encima.

Posicionamiento

Los navegadores crean y posicionan de forma automática todas las cajas que forman cada página HTML. No obstante, CSS permite al diseñador modificar la posición en la que se muestra cada caja.

Utilizando las propiedades que proporciona CSS para alterar la posición de las cajas es posible realizar efectos muy avanzados y diseñar estructuras de páginas que de otra forma no serían posibles.

El estándar de CSS define cinco modelos diferentes para posicionar una caja:

- **Posicionamiento normal o estático (static)**
Se trata del posicionamiento que utilizan los navegadores si no se indica lo contrario.
Si se utiliza este valor, se ignoran los valores de las propiedades top, right, bottom y left.
- **Posicionamiento relativo (relative)**
Variante del posicionamiento normal que consiste en posicionar una caja según el posicionamiento normal y después desplazarla respecto de su posición original.
El desplazamiento de la caja se controla con las propiedades top, right, bottom y left.
- **Posicionamiento absoluto (absolute)**
La posición de una caja se establece de forma absoluta respecto de su elemento contenedor y el resto de elementos de la página ignoran la nueva posición del elemento.
El desplazamiento de la caja también se controla con las propiedades top, right, bottom y left, pero su interpretación es mucho más compleja, ya que el origen de coordenadas del desplazamiento depende del posicionamiento de su elemento contenedor.
- **Posicionamiento fijo (fixed)**
Variante del posicionamiento absoluto que convierte una caja en un elemento inamovible, de forma que su posición en la pantalla siempre es la misma independientemente del resto de elementos e independientemente de si el usuario sube o baja la página en la ventana del navegador.
El desplazamiento se establece de la misma forma que en el posicionamiento absoluto, pero en este caso el elemento permanece inamovible en la pantalla.

El posicionamiento de una caja se establece mediante la propiedad position:

position	Posicionamiento
Valores	static relative absolute fixed
Se aplica a	Todos los elementos
Valor inicial	static
Descripción	Selecciona el posicionamiento con el que se mostrará el elemento

Normalmente, cuando se posiciona una caja también es necesario desplazarla respecto de su posición original o respecto de otro origen de coordenadas. CSS define cuatro propiedades llamadas top, right, bottom y left para controlar el desplazamiento de las cajas posicionadas:

top right bottom left	Desplazamiento superior Desplazamiento lateral derecho Desplazamiento inferior Desplazamiento lateral izquierdo
--------------------------------	--

Valores	<medida> <porcentaje> auto inherit
Se aplica a	Todos los elementos posicionados
Valor inicial	auto
Descripción	Indican el desplazamiento horizontal y vertical del elemento respecto de su posición original

En el caso del posicionamiento relativo, cada una de estas propiedades indica el desplazamiento del elemento desde la posición original de su borde superior/derecho/inferior/izquierdo. Si el posicionamiento es absoluto, las propiedades indican el desplazamiento del elemento respecto del borde superior/derecho/inferior/izquierdo de su primer elemento padre posicionado.

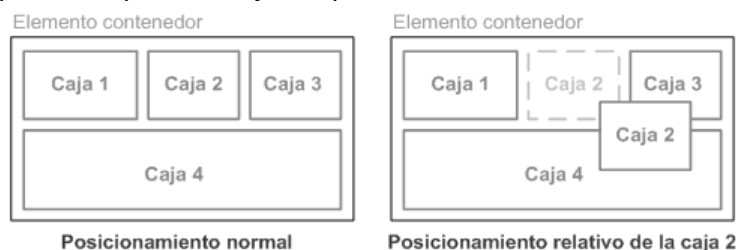
En cualquiera de los dos casos, si el desplazamiento se indica en forma de porcentaje, se refiere al porcentaje sobre la anchura (propiedades right y left) o altura (propiedades top y bottom) del elemento.

Posicionamiento relativo

El estándar CSS considera que el posicionamiento relativo es un caso particular del posicionamiento normal, aunque en realidad presenta muchas diferencias.

El posicionamiento relativo permite desplazar una caja respecto de su posición original establecida mediante el posicionamiento normal. El desplazamiento de la caja se controla con las propiedades top, right, bottom y left.

El desplazamiento de una caja no afecta al resto de cajas adyacentes, que se muestran en la misma posición que si la caja desplazada no se hubiera movido de su posición original.

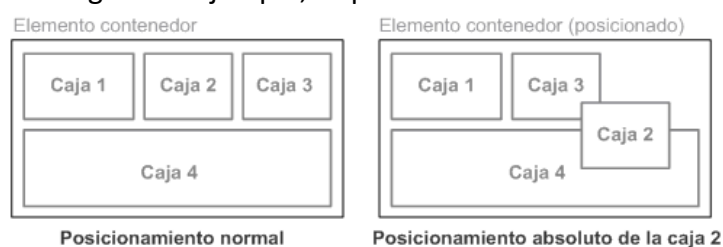


Posicionamiento absoluto

El posicionamiento absoluto se emplea para establecer de forma precisa la posición en la que se muestra la caja de un elemento. La nueva posición de la caja se indica mediante las propiedades top, right, bottom y left. La interpretación de los valores de estas propiedades es mucho más compleja que en el posicionamiento relativo, ya que en este caso dependen del posicionamiento del elemento contenedor.

Cuando una caja se posiciona de forma absoluta, el resto de elementos de la página la ignoran y ocupan el lugar original ocupado por la caja posicionada. Al igual que en el posicionamiento relativo, cuando se posiciona de forma absoluta una caja es probable que se produzcan solapamientos con otras cajas.

En el siguiente ejemplo, se posiciona de forma absoluta la caja 2:



19.MENÚ DE OPCIONES

Diferentes ejemplos de cómo hacer un menú de opciones en una página web.

Ej6menus

20.HERENCIA DE ESTILOS

En el conjunto de las etiquetas HTML podemos establecer una jerarquía de etiquetas que contienen a otras, para darnos una relación de herencia. En primer lugar, tendríamos la etiqueta `<body> ... </body>`, que hace referencia a todo el documento, y podemos considerarla como la etiqueta "padre" de todas las demás etiquetas de formato, puesto que todas ellas se encuentran contenidas en el cuerpo (body) del documento.

Después, tenemos las etiquetas de párrafo (`<p>...</p>`, `<div>...</div>`, cabeceras, ...) y etiquetas de elementos insertados en línea (`...`, `<i>...</i>`, `...`, ...). Las etiquetas de párrafo serán contenedoras de las etiquetas de elementos insertados en línea (en el sentido que les estamos dando), estableciéndose así una nueva relación "padre-hijo".

Esto es interesante porque la mayoría de los estilos que se definen se heredan, es decir, si definimos un cierto estilo para una etiqueta, este estilo será heredado por las etiquetas "hijas", con lo que no tendremos que volver a definirlo para ellas. Por ejemplo, si definimos un tipo de letra y un color para la fuente para la etiqueta `<body> ... </body>`, este estilo será heredado por todas las etiquetas del documento y no tendremos que definirlo para las otras etiquetas.

Sin embargo, si tenemos definido un estilo para una etiqueta "padre", podremos definir un estilo distinto para una etiqueta "hija", es decir, un estilo heredable se hereda a no ser que especifiquemos lo contrario. Se heredarán aquellas características que no pongamos, y se aplicarán aquellas que definamos para la etiqueta que no tenga la etiqueta "padre".

Para el resto de propiedades CSS que por defecto no se heredan, se puede utilizar el valor **inherit** para forzar a que se produzca la herencia automática.