

HTML5

Introducción

Es la última revisión del estándar HTML. Se incluyen algunas etiquetas nuevas cuyo significado se comentará a continuación.

Etiquetas estructurales

Crean la estructura para el resto de la página:

- doctype: identifica el estándar.
- todo documento debe ir entre las marcas `<html>` y `</html>`.
- Todo html tiene dos partes: head y body. El primero incluye otros elementos estructurales como `<title>` que indica el título de dicha página. Dentro del body se incluye el contenido real de la página.
- Dentro del body pueden incluirse otras etiquetas que estructuran el contenido de la página:
 - La etiqueta `<section>` permite marcar contenido de una página relacionado con un tema concreto.
 - `<article>` es una unidad de contenido sobre un tema específico el cual puede ser independiente de otros «artículos».
 - `<header>` se utiliza para indicar cual es la cabecera de un artículo o sección.
 - `<footer>` permite definir un «pie de página», normalmente con indicación de derechos de autor, fecha o datos similares.
 - `<address>` se usa para marcar información de contacto.
 - `<aside>` se usa para definir contenido con «una relación vaga con el resto de la página» (definición tomada del estándar).
 - `<hgroup>` permite agrupar un conjunto de encabezados y marcarlos como pertenecientes al mismo contenido.
- `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>` y `<h6>` establecen encabezados: trozos de texto que identifican la importancia del siguiente trozo de texto.

- Cualquier etiqueta puede ir comentada. Los comentarios no se muestran, son solo de interés para el programador en un futuro. Un comentario se abre con `<!--` y se cierra con `-->`
- La etiqueta `<a>` permite construir enlaces.
- La etiqueta `<nav>` se utilizará para crear barras de navegación.
- La etiqueta `<aside>` se utiliza para indicar información relacionada con el artículo o texto pero que no tiene porque ser parte del mismo. El ejemplo más común es utilizarlo para publicidad relacionada o texto del tipo «artículos relacionados con este».
- Las etiquetas `<script>` y `<noscript>` se utilizan para marcar pequeños programas o la ausencia de ellos.
- El elemento `<main>` indica **el contenido principal de una página**
- La etiqueta `<div>` permite crear «divisiones» en el documento y nos serán muy útiles cuando hagamos posicionamiento. Una etiqueta muy similar, pero con un comportamiento distinto es `` que permite dividir trozos de texto en un párrafo.
- La etiqueta `<details>` junto con la etiqueta `<summary>` permite crear controles que ofrecen al usuario la posibilidad de «ampliar información» a voluntad.

```
<body>
  <details>
    <summary>HTML5</summary>
    <p>Es la última revisión del lenguaje</p>
  </details>
</body>
```

- La etiqueta `<hr>` produce una línea horizontal.

! Nota

Una de las grandes preguntas es si escribir la etiqueta `
` o `
`. La respuesta corta es que da igual. La respuesta larga es que el W3C permite que «*aquellos elementos que nunca lleven nada dentro pueden indistintamente escribirse de manera autocerrada o simplemente sin cerrar*». Así, en la cabecera podremos poner cosas como `<meta...>` o `<meta... />`

- La etiqueta `<abbr>` permite usar abreviaturas. Ejemplo: El `<abbr title="World Wide Web Consortium">W3C</abbr>` .
- La etiqueta `<acronym>` se ha llegado a usar pero está **obsoleta** .
- La etiqueta `<blockquote>` se usa para indicar citas largas, por ejemplo.

`<blockquote>`

`<p>En un lugar de la Mancha de cuyo nombre...</p>`

`</blockquote>`

`uote>`

En un lugar de la Mancha de cuyo nombre...</p>

`quote>`

- La etiqueta `<cite>` permite indicar una cita breve, como en `<p> Es habitual decir <cite>Alea jacta est</cite></p>`
- La etiqueta `` permite mostrar texto tachado.
- La etiqueta `<dfn>` permite mostrar una definición. Los navegadores suelen mostrar esta etiqueta igual que `<abbr>` .
- La etiqueta `<ins>` permite indicar texto que haya sido insertado a posteriori. Suele mostrarse en cursiva.
- La etiqueta `<mark>` permite destacar texto de una manera muy llamativa.
- La etiqueta `<output>` es como el control de un formulario pero se introdujo para recalcar que contiene el resultado de un cálculo.
- La etiqueta `<samp>` está pensada para muestras (samples) de programas o resultado de ejecuciones.
- La etiqueta `<tt>` imita los teletipos (estilo máquina de escribir).
- La etiqueta `<wbr>` está pensada para ayudar al navegador a decidir donde «romper» una palabra y poner el guión.

Gestión de espacios

Los navegadores web manejan el espacio de una forma un poco especial:

- Si se pone uno o varios espacios en blanco o si se pulsa la tecla ENTER muchas veces el navegador mostrará *un solo espacio en blanco*
- Para poner un espacio en blanco horizontal se puede usar la entidad ` `.
- Para hacer un salto de línea se puede usar la etiqueta `
` (esta etiqueta no lleva asociada una etiqueta de cierre, es *autocerrada*)
- Se puede indicar el comienzo y el final de un párrafo con `<p>` y `</p>`.

Una pregunta habitual es «¿Cuándo se debe usar `<p>` y cuándo `
`?». La respuesta es «depende». Una posible respuesta es que si se escriben varios párrafos relacionados es bastante habitual separarlos con `
` mientras que si se ponen varios párrafos que hablan de distintas cosas es habitual usar `<p>` con cada uno de ellos, sin embargo no hay una respuesta universal.,

Entidades

Las entidades HTML permiten escribir determinados símbolos especiales que podrían confundir al navegador, así como otros símbolos que no aparecen directamente en los teclados:

- `<` y `>` representan los símbolos `<` y `>`.
- `©`
- `™`
- `®`
- `€` y `¥`
- `&`

Texto preformateado

Algunas marcas, como `<pre>` permiten obligar al navegador a que respete los espacios en blanco tal y como aparecen en la página original.

Si se desea indicar que algo debe ser teclado por el usuario se usa la marca `<kbd>`.

Si se desea indicar que algo es una variable se puede usar la marca `<var>`.

La etiqueta `<code>` permite indicar que un determinado es código en un lenguaje de programación.

Listas

Es una secuencia de elementos relacionados en torno a un mismo concepto Para abrir una lista de elementos se utilizan dos posibles marcas:

- `` Para crear una lista ordenada (numerada)
- `` Para crear una lista desordenada (no numerada)

Una vez creadas hay que etiquetar cada elemento de la lista con la etiqueta ``.

En un plano distinto se pueden encontrar las **listas de definiciones**. Con estos elementos se puede especificar una secuencia de términos para los cuales proporcionamos una definición. Su estructura es la siguiente:

- `<dl>` y `</dl>` marcan el inicio y el final de la lista de definiciones. Dentro de estas etiquetas pondremos las dos siguientes.
- `<dt>` y `</dt>` especifican el *término* que vamos a definir.
- `<dd>` y `</dd>` indican la definición asociada al término anterior.

Tablas

Una tabla muestra un conjunto de elementos relacionados en forma de matriz. No deberían usarse para maquetar la posición de los elementos. Todo el contenido de la tabla debe ir entre las etiquetas `<table>` y `</table>`. Las tablas se construyen de izquierda a derecha (por columnas) y de arriba a abajo (filas).

Una tabla puede tener una cabecera, un cuerpo y un pie, especificados por `<thead>`, `<tbody>` y `<tfoot>`. La primera etiqueta dentro de `<tbody>`, solo puede ser `<tr>`.

Cuidado al crear tablas, todo dato, o subtablas debe ir dentro de `<td>`, es absolutamente obligatorio

Para ser exactos una tabla puede llevar estas etiquetas:

- `caption`: permite asociar a una tabla un texto.
- `colgroup`: permite asociar información a columnas usando dentro la etiqueta `<col>`.
- `thead`: dentro de ella a su vez pondremos una fila (`<tr>`) con celdas en las que la etiqueta es `<th>`
- `tbody`: utiliza las filas y columnas normales.
- `tfoot`: también usa `<tr>` y `<td>` de la forma habitual, sin embargo permite describir mejor el contenido de la tabla. Se utiliza para celdas con los valores acumulados o similares.

Formularios

Un formulario permite que el usuario interactúe con la página por medio de una serie de controles

Campo de texto

Permite crear una zona donde el usuario puede escribir y se muestra un ejemplo a continuación. Tiene algunos atributos que se usan muy a menudo:

- **type** indica el tipo de control
- **name** será el nombre de la variable en JS (no lo usaremos por ahora)
- **id** permitirá procesar el control de JS (no se usará por ahora)
- **value** permite indicar un valor por defecto
- **size** indica la anchura por defecto

Un campo de texto puede llevar asociada una etiqueta **label** que indique al navegador que texto va con ese campo. Esto es de utilidad para programas lectores

Si el type de este elemento se sustituye por **password** se obtiene un control igual, pero que reemplaza el texto por símbolos que ocultan el texto.

Es posible ofrecer un conjunto de posibles opciones para un campo de texto usando un control llamado **<datalist>**. Así, supongamos que pedimos el idioma al usuario pero deseamos ofrecer la posibilidad de autocompletar mostrando algunos idiomas comunes. Se puede usar este código:

Campo email

Si sabemos con seguridad que en un campo se va a introducir el email se puede usar este código:

```
<br/>
```

Introduzca su email:<input type="email" name="sexo">

El navegador hará **automáticamente** la comprobación de que lo que introduce el usuario es realmente un email.

Selector único (radio-button)

Permite elegir una sola opción de entre muchas, se necesita usar un **input** de tipo **radio**.

```
<br/>
```

```
<input type="radio" name="sexo">Masculino
```

```
<br/>
```

```
<input type="radio" name="sexo">Femenino
```

Si se desea que un control de tipo **checkbox** o **radio** aparezca marcado por defecto se debe añadir el atributo **multiple="multiple"**.

Selector múltiple (checkbox)

Permite elegir múltiples combinaciones de opciones. El nombre del control utilizará los corchetes para crear un vector que se procesará desde Javascript.

Lista desplegable

Permite elegir valores de una lista.

```
<select name="provincia">
```

```
  <option value="AB">Albacete</option>
```

```
  <option value="CR">Ciudad R.</option>
```

```
  <option value="CU">Cuenca</option>
```

```
</select>
```

Se debe recordar que el texto que ven los usuarios es lo que va entre las etiquetas option. El valor que comprobarán los programadores es lo que va en value En una lista desplegable se pueden elegir muchos valores usando el atributo **multiple**.

Si se desea que una opción (o varias si usamos el selector múltiple) aparezca marcada se debe usar **selected="selected"**.

Textareas

Permiten introducir textos muy largos:

```
<textarea rows="10" cols="15">
```

Valor por defecto

```
</textarea>
```

Otros elementos HTML: contenido embebido y multimedia

En HTML existen otras etiquetas que permiten insertar contenido dentro del HTML que no tiene por qué ser HTML

Contenido embebido en general

Podemos usar la etiqueta **object** para insertar contenido de otro tipo, como archivos de vídeo, de audio, documentos PDF etc... Así, por ejemplo, el siguiente HTML inserta una imagen (cosa que en realidad ya se podía hacer con la etiqueta **img**

```
<object data="low_res.png" width="550px" height="150px">Imagen </object>
```

El problema principal con la etiqueta **object** es que su soporte dentro de los navegadores es menos amplio que el de las etiquetas que veremos ahora. Sin embargo, su versatilidad es mayor.

La etiqueta **audio** permite insertar audios dentro del documento, ofreciendo además un interfaz de control del audio con los elementos típicos: reproducción, parada, control de volumen, etc...

```
<audio controls="controls" src="media/cancion.mp3">
```

Su navegador no ofrece soporte para audios embebidos

```
</audio>
```

La etiqueta **video** funciona de manera similar a **audio** permitiendo insertar en este caso vídeos dentro de una página.

```
<video controls="controls" src="videos/video.mp4">
```

Su navegador no ofrece soporte para vídeos embebidos.

```
</video>
```