[twitter](https://twitter.com/jorgesancheznet) [FaceBook](https://facebook.com/jorgesancheznet) [You Tube](https://youtube.com/jorgesancheznet) [Git Hub](https://github.com/jorgesancheznet) [Enviar email](https://jorgesanchez.net/manuales/html/elementos-estructurales-html.html)

[JorgeSanchez.net logo](https://jorgesanchez.net/)

**Manual de HTML5 y CSS3**

**[5]  
Elementos estructurales**

**PUBLICIDAD**

**[5.1] elemento *div***

Se trata de un elemento clásico, ya estaba presente en las primeras versiones de HTML, que se utiliza como contenedor de otros elementos a fin de poder manipularlos todos a la vez. Dentro de **div** se pueden colocar todo tipo de etiquetas (tablas, párrafos, imágenes,…).

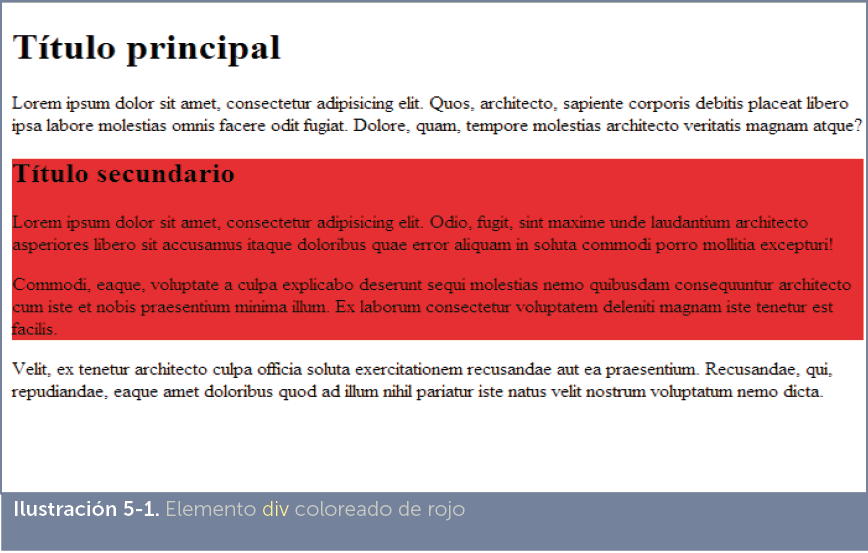
Desde hace años se ha convertido en el elemento utilizado, combinado con CSS, para crear **capas**. Las capas son elementos visuales que se pueden posicionar a voluntad en la página web. De esta forma, a través de las capas se consiguen maquetaciones muy artísticas.

Por defecto, los navegadores no dan formato al contenido de un elemento *div*, con lo que visualmente no se diferencia del resto de elementos, salvo por el hecho de que un elemento div tiene un display de tipo block. Es decir, el contenido de estos elementos, por defecto, se muestra de forma separada al resto.

En el ejemplo siguiente se destaca el elemento div coloreándolo de rojo:

|  |
| --- |
| **<h1>**Título principal**</h1>**  **<p>**Lorem ipsum dolor sit amet, consectetur adipisicing  elit. Quos, architecto, sapiente corporis debitis  placeat libero ipsa labore molestias omnis facere  odit fugiat. Dolore, quam, tempore molestias  architecto veritatis magnam atque?  **</p>**  **<div** **style=**”background-color:red”;**>**  **<h2>**Título secundario**</h2>**  **<p>**Lorem ipsum dolor sit amet, consectetur  adipisicing elit. Odio, fugit, sint maxime unde  laudantium architecto asperiores libero sit  accusamus itaque doloribus quae error aliquam in  soluta commodi porro mollitia excepturi!  **</p>**  **<p>**Commodi, eaque, voluptate a culpa explicabo  deserunt sequi molestias nemo quibusdam  consequuntur architecto cum iste et nobis  praesentium minima illum. Ex laborum consectetur  voluptatem deleniti magnam iste tenetur est  facilis.  **</p>**  **</div>**  **<p>**Velit, ex tenetur architecto culpa officia  soluta exercitationem recusandae aut ea  praesentium. Recusandae, qui, repudiandae, eaque  amet doloribus quod ad illum nihil pariatur iste  natus velit nostrum voluptatum nemo dicta.  **</p>** |

El resultado en el navegador sería:

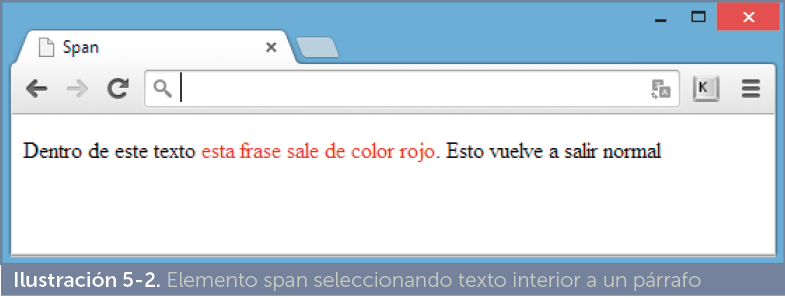


**[5.1.1]elemento span**

Es muy similar al anterior, pero como los navegadores no agregan nada (ni siquiera espacios) a esta etiqueta cuando se muestra por pantalla, en lugar de para definir capas (aunque se podría) se usa para marcar contenido interior a un párrafo, a fin de que a ese contenido se le pueda dar un formato especial mediante CSS. Es decir, con **div** marcamos capas (contenedores grandes) y con **span** marcamos bloques pequeños de texto.

Ejemplo:

|  |
| --- |
| **<p>**Dentro de este texto **<span** **style=**”color:red”**>**esta frase sale de color rojo**</span>**. Esto vuelve a salir normal**</p>** |



**[5.2] elementos estructurales semánticos**

La idea en HTML5 (a diferencia de las versiones anteriores) es que los elementos HTML aporten valor semántico al contenido. Es decir, se trata de que el código HTML indique qué tipo de contenido es el que está dentro de cada elemento. Eso vale para casi todos los elementos ya conocidos como **p** (párrafo), **abbr** (abreviatura), **strong** (remarcado fuerte), etc.

Sin embargo los estándares anteriores a la versión 5 tenían una carencia muy importante. El único elemento para marcar contenedores grandes era **div**. Y este elemento no aporta ninguna semántica a su contenido.

En HTML 5 hay una serie muy importante de elementos que no dan ningún formato al texto, salvo el hecho de que aparezcan con un display de bloque, pero permiten darle un significado. Será CSS el encargado de que esos elementos aparezcan con un formato apropiado a su significado en la página final. La idea final es que estos elementos sustituyan el sobreuso que, tradicionalmente, se hacía de la etiqueta **div**.

No es fácil esta idea, puesto que existen numerosas plantillas y patrones de trabajo profesionales (como ocurría por ejemplo con **Bootstrap**) que se basan en *div* para trabajar con ellas. La mayoría de plantillas actuales (incluida la propia Bootstrap) se han adaptado a esta idea.

**[5.2.1]elemento header**

Permite marcar contenido como cabecera del documento. Hay que señalar que este elemento no tiene nada que ver con el elemento **head**. Header marca una serie párrafos para indicar que pertenecen a la cabecera de la página.

La cabecera de la página suele contener el logotipo de la página, el menú, el título principal, etc.

Realmente una página puede tener varios elementos **header**. Si está al nivel de la etiqueta **body** indica que su contenido es la cabecera de la página completa. Pero dentro de un elemento de sección, por ejemplo, **article,** indicaría que su contenido es la cabecera de ese artículo.

No se debe indicar un elemento header dentro de otro header ni dentro de un elemento **footer** o **address**.

**[5.2.2]elemento footer**

Similar al anterior, pero sirve para marcar el pie de una página, sección, artículo etc. En el caso de que un elemento footer se coloque al nivel del elemento **body**, servirá para indicar el pie general de la página que suele contener información sobre el autor, copyright, términos de uso de la página, contacto, etc.

Al igual que ocurría con header, no se puede colocar un footer dentro de otro ni puede estar dentro de header o address.

**[5.2.3]elemento main**

Es incluso más moderno que los anteriores. Permite indicar una sección (que podrá incluir artículos, cabeceras, títulos, párrafos, secciones de imagen,…) de forma que se entenderá que esta es la sección principal del documento.

Solo puede haber un elemento main en una página web. Se usa para diferenciar de las secciones de menú, publicidad o cualquier tipo de información en la página considerada como secundaria.

**main**no puede colocarse dentro de secciones de tipo **article**, **footer**, **header**, **aside**o **nav.**Sí se admite dentro de **section**, para marcar elementos principales dentro de una sección.

**[5.2.4]elemento section**

Es un elemento que permite dividir en diferentes partes o secciones un documento. Ejemplo de página dividida en secciones:

|  |
| --- |
| **<!DOCTYPE html>** **<html lang=**”es-ES”**>**  **<head>**  **<meta charset=**”UTF-8”**>**  **<title></title>**  **</head>**  **<body>**  **<header>**  **<h1>**Historia de HTML**</h1>**  **<p>**Desde 1989 hasta nuestros días**</p>**  **</header>**  **<section id=**”inicio”**>**  **<h2>**Inicios en HTML. Tim Bernes Lee**</h2>**  **<p>**.....**</p>**  **</section>**  **<section id=**”creacion”**>**  **<h2>**Creación de la web. Primeros navegadores  **</h2>**  **<p>**.....**</p>**  **</section>**  **<section id=**”guerra”**>**  **<h2>**La guerra de los navegadores**</h2>**  **<p>**.....**</p>**  **</section>** |
| **<section id=**”versiones”**>**  **<h2>**De HTML 4 a HTML 5 pasando por XHTML**</h2>**  **<p>**...**</p>**  **</section>**  **<footer>**  **<p>**Realizado por Jorge Sánchez**</p>**  **</footer>**  **</body>** **</html>** |

El atributo **id** no es obligatorio, pero a veces se usa con la finalidad de identificar de modo único a cada sección, por ejemplo para darle un formato especial a una sección concreta.

**[5.2.5]elemento nav**

Se trata de un elemento que marca a su contenido como una sección de enlaces o menú, es decir una barra de navegación. Más adelante con CSS se puede dar un formato especial a dichos enlaces. **nav** se puede escribir dentro de cualquier elemento HTML estructural (**section, article, header, footer**,..), pero tampoco se deben marcar todos los enlaces de navegación como parte de un elemento nav, solo los que formen un menú real y completo de navegación.

Los elementos marcados por **nav** pueden ser omitidos por los lectores digitales de páginas web que utilizan, por ejemplo, las personas invidentes. Lo cual facilita la comprensión del texto) y así que dicho contenido quede marcado solo para utilizar los enlaces interiores a **nav**.

Lógicamente dentro de **nav** se suelen incluir numerosos elementos de tipo **a**.

**[5.2.6]elemento article**

Si se observan los elementos descritos anteriormente parece claro que HTML 5 utiliza, como metáfora, la forma de distribuir contenidos de un periódico. Así hay cabeceras, pies, secciones y, con este elemento, artículos.

La idea es colocar dentro de este elemento, que tiene sentido que aparezca dentro del elemento **section** o incluso aparecer de forma independiente, contenido que pueda ser entendido como un *todo* que describa un tema de forma íntegra.

Es decir dentro de **article** se colocan los elementos que sirvan para describir un mismo tema.

Es posible (aunque poco habitual) colocar elementos article dentro de otros elementos article, pero solo cuando queremos indicar artículos externos relacionados con el artículo principal.

La autoría del artículo (nombre del autor y créditos) se puede indicar con el elemento **address**.

**[5.2.7]elemento figure**

Sirve para agrupar los elementos relativos a una imagen como la propia imagen (elemento **img**), el título de la misma, el pie, los párrafos relativos, etc. Contenido relacionado con el que tiene alrededor, pero que se podría utilizar de forma independiente en otro documento.

**[5.2.8]elemento figcaption**

Permite indicar el título de una imagen, dentro de un elemento **figure**, haciendo el efecto de un pie de imagen. Por defecto la imagen y el texto quedarán alineados de la misma forma.

Ejemplo de uso:

|  |  |
| --- | --- |
| **<article>  <hgroup>  <h2>**Palencia**</h2>**  **<h3>**Paisajes**</h3>**  **</hgroup>**  **<figure>**   **<img src=**”montanaPalentina.jpg”  **alt=**”Vista de la Montaña Palentina”  **width=**”350” **height=**”400”/>   |  | | --- | | **<figcaption>**  Vista de la Montaña Palentina desde  Camporredondo  **</figcaption>** |   **</figure>** **<p>**  Los paisajes en la provincia de Palencia.......  **</p>**  ....   **</article>** |

**[5.2.9]elemento aside**

Permite marcar texto dentro de un artículo para que no se tenga en cuenta como parte del texto del artículo, sino como un texto aparte que permite realizar aclaraciones al artículo, referencias, resúmenes remarcados y sobre todo cuadros de texto de estilo periodístico para destacar partes del artículo. Para que este texto aparezca de manera especial, debe dársele formato con CSS.

La idea es simular el típico recuadro que aparece en los periódicos en los que se expresan ideas interesantes o que resumen el artículo.

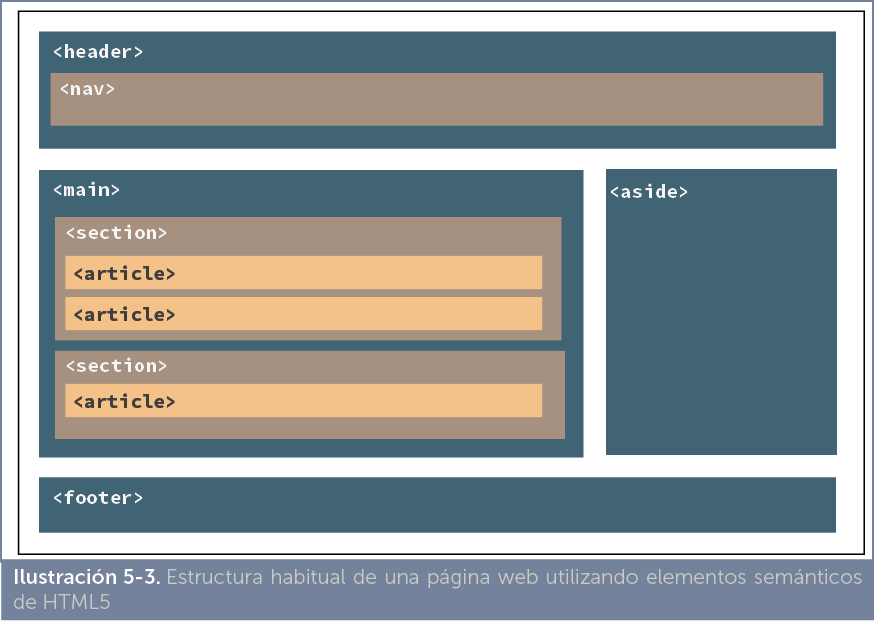
En definitiva, aside define texto que no forma parte del flujo principal de la página sino que posee contenido que debe de fluir aparte.

**[5.3] títulos en las elementos de sección**

Podemos entender que hay contenedores que permiten estructurar un contenido de forma completa y que este contenido es, a su vez, parte de un todo más grande. Es decir, una página actual se puede entender como un conjunto de secciones cada sección, a su vez, se puede dividir en más secciones.

De este modo, los títulos h1, h2, h3, etc., permiten establecer una jerarquía extra. Es decir: un elemento h1 dentro directamente del elemento body se consideraba un título principal de toda la página web. Ahora, normalmente, se colocará h1 dentro de una sección concreta, lo que significará que ese h1 es el título principal de esa sección. Incluso esa sección se puede dividir en subsecciones, cada una de las cuales tendrán sus propios elementos h1 que serán títulos principales pero de cada subsección, y por lo tanto de menos nivel que los h1 de la sección.

Esto suena a galimatías, pero lo cierto es que da una mayor flexibilidad a la hora de estructurar el contenido de las páginas y, también, a la hora de darla formato.



**[5.4] elementos para acciones interactivas**

**[5.4.1]elementos para indicar secciones de tipo mostrar/ocultar**

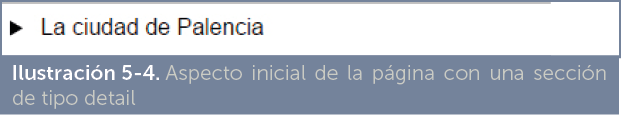
Se trata de secciones que están dentro de un elemento llamado **details**. Dentro de ese elemento se puede poner una sección de tipo **summary**la cual muestra un texto de resumen de la sección. Al hacer clic en el apartado summary, se muestra todo el contenido de la sección.

Es decir en definitiva consigue crear en un documento HTML un apartado de tipo resumen/detalle. Ejemplo:

Ejemplo:

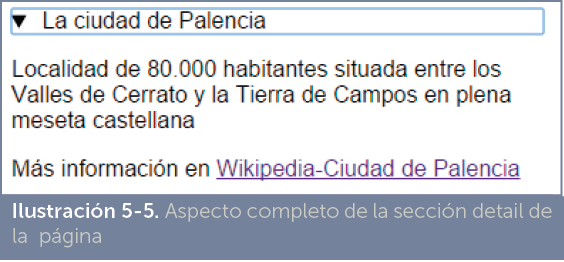
|  |
| --- |
| **<body>**  **<details>**  **<summary>**  La ciudad de Palencia  **</summary>**  **<p>**Localidad de 80.000 habitantes situada entre  los Valles de Cerrato y la Tierra de Campos  en plena meseta castellana**</p>**  **<p>Más información en**  **<a href=”https://es.wikipedia.org/wiki/Palencia”>**  Wikipedia-Ciudad de Palencia  **</a>**  **</p>**  **</details>**  **</body>** |

Al abrir esa página, se mostraría:



Como se observa, solo se muestra el contenido del apartado **summary**. Tras hacer clic en el triángulo o en el texto, se mostraría el contenido del elemento *details*.

Ejemplo:



**summary**y **details**forman parte de la especificación HTML 5.2, aunque ya todos los navegadores (salvo los de Microsoft: Edge y Explorer) los reconocen.

**element summary**

Se coloca dentro del anterior. Es un texto que se puede poner en los elementos de tipo **detail**. Ese texto aparece en lugar del texto *Detalles* que aparece por defecto en una sección de tipo *detail*.

**[5.4.2]elementos de cuadro de diálogo**

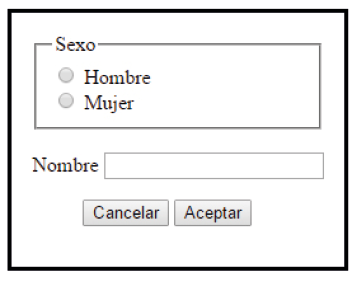
Se suelen llamar cuadros de diálogo a las ventanas que los sistemas muestran a los usuarios para darles mensajes, o para permitirles configurar una determinada respuesta u operación.

En HTML 5 se ha incorporado una nueva etiqueta llamada **dialog**, precisamente con esta finalidad. Todavía está en fase de pruebas. El contenido del elemento dialog se muestra en un recuadro. La gracia es que desde JavaScript tiene muchas opciones de manipulación.

Hay un atributo llamado **open** (no se le pone valor alguno cuando se usa) que hace que se muestre el cuadro de diálogo (sin ese atributo, el cuadro no se muestra). Ejemplo:

|  |
| --- |
| **<dialog id=”favDialog” open>**  **<form method=”dialog”>**  **<fieldset>**  **<legend>**Sexo**</legend>**  **<input type=”radio”** **id=”hombre”**   **name=”sexo”>**  **<label for=”hombre”>**Hombre**</label><br>**  **<input type=”radio” id=”mujer”   name=”sexo”>**  **<label for=”mujer”>**Mujer**</label>**  **</fieldset>**  **<br>** |
| **<label for=”nombre”>**Nombre**</label>**  **<input type=”text”** **name=”nombre”   id=”hombre”>**  **<button id=”cancel”** **type=”reset”**>  Cancelar  **</button>**  **<button type=”submit”>**Aceptar**</button>**  **</form>**  **</dialog>** |

Resultado:



Realmente sus principales posibilidades se consiguen combinando su uso con el lenguaje JavaScript.

El elemento **dialog**es parte de la recomendación HTML 5.1 y todavía está en proceso de adopción por parte de la mayoría de navegadores. En el momento de escribir este texto, solo Chrome y Opera reconocen este elemento.