

Guía HTML

Sitio: [Instituto Superior Politécnico Córdoba](#)

Curso: Programador Full Stack - TSDWAD - 2022

Libro: Guía HTML

Imprimido por: Ezequiel Maximiliano GIAMPAOLI

Día: jueves, 8 septiembre 2022, 4:13

Tabla de contenidos

1. Introducción

2. HTML

3. Estructura básica

4. Elementos y Etiquetas HTML

4.1. Organizar el texto

4.2. Listas

4.3. Resaltar

4.4. Enlaces

4.5. Imágenes

4.6. Tablas

4.7. Formularios

5. HTML 5

5.1. Header

5.2. Footer

5.3. Nav

5.4. Section

5.5. Aside

5.6. Article

5.7. Figuras

5.8. Resumen

1. Introducción



Podríamos decir que el lenguaje HTML es parte fundamental de lo que conocemos como www o world wide web, en su traducción al español, telaraña mundial o malla mundial. El lenguaje HTML fue creado y puesto a pruebas por Timothy Jhon Berners-Lee y su equipo del C.E.R.N., entre los años 1989 y 1991 (<https://www.w3.org/People/Berners-Lee/>) (<https://www.w3.org/History/1989/proposal.html>).

El mismo Tim Berners-Lee, ante la necesidad de distribuir e intercambiar información acerca de sus investigaciones de una manera más efectiva, desarrolló las ideas fundamentales que estructuran la www. Él y su grupo crearon lo que por sus siglas en inglés se denominan HTML (HyperText Markup Language), HTTP (HyperText Transfer Protocol) y URL (Uniform Resource Locator). En este módulo nos centraremos exclusivamente en el lenguaje HTML.

Entonces, desde el año 1991, el HTML se utiliza como lenguaje de marcado para el desarrollo de páginas web y es el encargado de presentar en un explorador web los contenidos de una manera estructurada jerárquicamente.

El HTML se compone de texto y una serie de etiquetas, y nos ofrece una gran adaptabilidad, una robusta estructuración lógica y es fácil de interpretar tanto por humanos como por máquinas.

Objetivos:

- Crear páginas web básicas respetando los últimos estándares HTML.
- Reconocer las etiquetas semánticas HTML 5 a fin de construir esquemas simples con encabezado, cuerpo y pie de página.
- Incorporar conocimientos básicos del lenguaje HTML para tener una visión amplia y poder complementarlo con otros lenguajes de front-end tales como CSS y Javascript.

2. HTML

¿Qué es HTML?

Por sus siglas en inglés HTML (Hypertext Markup Language) y se lo define como un lenguaje de marcado estándar que nos permite crear y presentar páginas web de una manera versátil y eficiente.

El HTML describe la *ESTRUCTURA* y el *CONTENIDO* de una página web y es un lenguaje que consiste en etiquetas agrupadas o estructuradas de una manera lógica en función de lo que necesitamos como vista. Estas etiquetas le dicen al "[navegador web](https://www.euskadi.eus/navegadores-web/web01-a2wz/es/)" (<https://www.euskadi.eus/navegadores-web/web01-a2wz/es/>) cómo debe mostrar el contenido.

HTML es un lenguaje y como tal tiene su propio vocabulario (palabras) y su propia gramática (reglas).

En la [actualidad y desde hace varios años](https://www.w3c.es/Consorcio/historia) (<https://www.w3c.es/Consorcio/historia>), el HTML está definido por el [W3C](https://www.w3.org). "El Consorcio World Wide Web (W3C) es una comunidad internacional que desarrolla estándares abiertos para asegurar el crecimiento a largo plazo de la Web" (<https://www.w3.org>).

¿Qué NO es HTML?

El HTML *no es un lenguaje de programación*, por lo tanto, no define el *COMPORTAMIENTO* (lógica) de las páginas web.

Cuando hablamos de lógica, hablamos de la capacidad del navegador web para realizar cálculos y la capacidad del navegador web para tomar decisiones en función de posibles entradas, como pueden ser un clic del usuario o un cálculo numérico de ciertos valores de entrada.

El HTML *no es un lenguaje de estilos*, por lo tanto, no define la estética (presentación visual) de las páginas web.

Es decir, con el lenguaje HTML en sí mismo, sólo podremos presentar texto e imágenes básicas en el navegador web. Para la lógica y la presentación estética de una página web, utilizaremos otros lenguajes complementarios al HTML, tales como *Javascript* y *CSS*.

Reglas gramaticales

- Las etiquetas siempre se tienen que cerrar. (E. pares y huérfanas)
- Los documentos anidados deben tener el correcto orden de apertura y cierre. (último en abrir, primero en cerrar)
- Si bien HTML5 permite el uso de mayúsculas, es recomendable que los nombres de las etiquetas estén en minúsculas.
- Los valores de los atributos siempre tienen que estar entre comillas (es indistinto si son simples o dobles).
- Los nombres de los archivos html deben contener caracteres alfanuméricos, - y _ (no ñ, ni acentos).
- La extensión de las páginas web debe ser .html

3. Estructura básica

La estructura básica de una página web se divide en una cabecera <head> ... </head> y un cuerpo <body> ... </body>. El esqueleto básico de una página web es:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8" />
5     <title>Título</title>
6   </head>
7
8   <body>
9
10  </body>
11 </html>
```

Doctype

La primera línea se denomina el **doctype**. Es fundamental porque indica que esto es efectivamente una página web HTML.

No es realmente una etiqueta como las otras (empieza con un signo de exclamación) y se puede considerar como una excepción que confirma la regla.

El elemento </html>

Este es el elemento principal del código. Incluye el contenido completo de la página. ¡Como ven la etiqueta </html> de cierre está situada a la derecha del final del código!

La cabecera <head> y el cuerpo <body>

Una página web se compone de dos partes:

- La cabecera <head>: esta sección proporciona información general sobre la página como su título, la codificación (para la gestión de caracteres especiales), etc. Esta sección suele ser bastante corta. La información que contiene la cabecera no se muestra en la página y es simplemente información general concebida para el ordenador. ¡Sin embargo, es muy importante!
- El cuerpo <body>: es aquí donde se encuentra la parte principal de la página. Todo lo que tecleamos aquí se mostrará en la pantalla. La mayor parte de nuestro código se escribirá en el cuerpo.

La codificación (charset)

```
1 <meta charset="utf-8" />
```

Esta etiqueta indica la codificación que se usa en el archivo .html.

Es lo que determina la forma en la que se van a mostrar los caracteres especiales (acentos, caracteres japoneses y chinos, caracteres arábigos, etc.)

taskDesafío: Investiga ¿Qué son las etiquetas meta? ¿Para qué sirven? Enumera aquellas que consideres más importantes. Las compartimos en el foro!

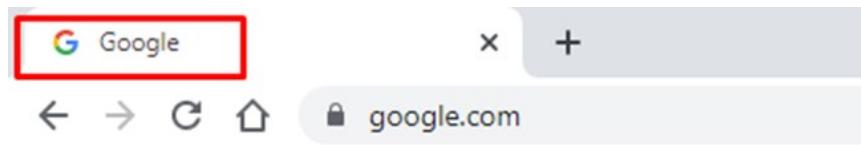
El título principal de la página

```
1 <title>
```

Este es el título de la página, **¡lo cual es probablemente el elemento más importante!** Todas las páginas tienen que tener un título que describa lo que contienen.

Es aconsejable que el título se mantenga bastante corto (normalmente menos de 100 caracteres).

El título no se muestra en la página sino en su parte superior (habitualmente en la pestaña del navegador). Guardar la página web y abrirla en el navegador. Verán que el título se muestra en la pestaña como se ve en la figura siguiente.



El título de la página figura en la parte superior del navegador

Consejos:

- Prestar atención al cerrar etiquetas porque éstas tienen un orden. Ej. No se puede cerrar la etiqueta de </html> antes que la de </body>
- Poner títulos a todos los documentos web procurando que sean lo más descriptivos posibles dado que si algún usuario decide incluir dicha página en su lista de marcadores, será el título el que quede almacenado en dicha lista (junto a la URL)
- En cualquier parte del html, se pueden incluir comentarios de la siguiente manera: <!-- Comentario -->. Los comentarios no se procesan y no producen salida visible de la página.
- Los saltos de línea y espacios en blanco son irrelevantes para el navegador.

4. Elementos y Etiquetas HTML

Las etiquetas HTML, las cuales están rodeadas por los signos menor que < y mayor que >, estas etiquetas también son llamadas o conocidas por el nombre tag (inglés) y son palabras clave para el navegador web. Por medio de las etiquetas podremos estructurar la información a presentar en pantalla.

Generalmente las etiquetas vienen de a pares, como la de párrafo <p> y </p>

La primera etiqueta del par se llama etiqueta de apertura <etiqueta>. El tag o etiqueta de cierre se escribe igual que el de apertura pero con la barra de división "</etiqueta>".

Elementos HTML

En la siguiente figura se observan los componentes de un elemento HTML. Los elementos HTML están formados por etiquetas de apertura <etiqueta>, etiqueta de cierre </etiqueta> y el texto contenido entre ellas.

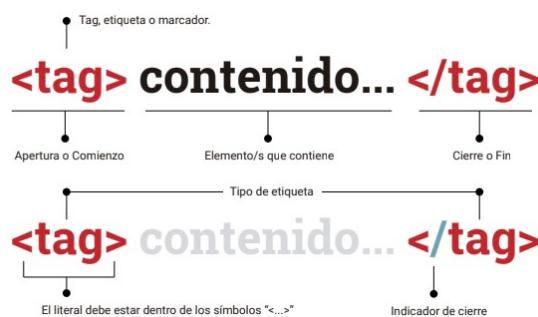


Figura Elemento HTML

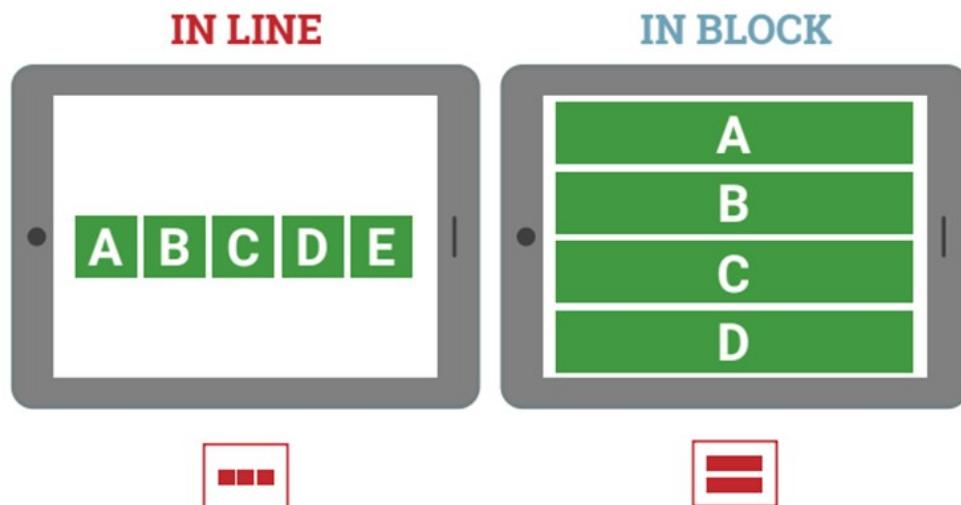
En el ejemplo que sigue a continuación se observan 4 elementos, el elemento <body>, el elemento <h1> y dos elementos <p>. Vale destacar que los elementos <h1>, <p> y <p>, están contenidos dentro del elemento <body>, de esta manera podemos observar cierta estructura y podríamos hacer una analogía con una estructura de cajas de cartón, donde hay ciertas cajas que van dentro de otras y ciertas cajas que van una al lado de otra. Con la totalidad de las cajas, analógicamente armaremos la totalidad de información que se va a presentar en pantalla.

En este punto es importante aclarar que, los elementos y las etiquetas no son las mismas cosas. Las etiquetas comienzan o terminan un elemento en el código fuente, mientras que los elementos son parte del DOM (Document Object Model), el modelo de documento para mostrar la página en el explorador web (<https://developer.mozilla.org/es/docs/Glossary/Element>).

Elementos de bloque y de línea

Todos los elementos HTML tienen un valor de display por defecto dependiendo del tipo de elemento que sea.

La mayoría de los elementos tienen valor de display de bloque (block) o de línea (inline).



Desafío: Investiga ¿Cuál es la diferencia entre los elementos *inline* y *block*? Enumera ejemplos de cada uno.

Anidar elementos

Se puede colocar elementos dentro de otros elementos, esto se denomina **anidamiento**. Si quisieramos resaltar una palabra dentro de un párrafo, la podríamos encerrar dentro de un elemento , que significa que dicha palabra se debe enfatizar.

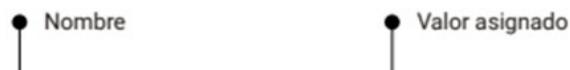


A continuación, te invitamos a mirar un video de anidamiento. Enfatizamos una palabra o frases dentro de un párrafo:



Atributos

Como vemos en el siguiente ejemplo, el atributo class contiene información adicional acerca del elemento <p> pero esta información no aparecerá dentro del contenido del elemento. En este caso "class" es el nombre del atributo e "importante" el valor del atributo. El atributo "class" permitirá darle al elemento un nombre identificativo, que se puede utilizar luego para aportar a ese elemento información de estilo. En otras unidades aprenderemos sobre CSS.



<tag atributo='valor'>

contenido...

</tag>

Ejemplo de uso:

link a google



Un atributo debe tener las siguientes características:

1. Un espacio entre este y el nombre del elemento (o del atributo previo, si el elemento ya posee uno o más atributos).
2. Un atributo debe estar definido por el par nombre/valor, como por ejemplo nombre = "valor".
3. Comillas de apertura y de cierre, encerrando el valor del atributo.
4. Un atributo siempre se especifica en la etiqueta de apertura del elemento HTML.

4.1. Organizar el texto

Para organizar el texto HTML nos provee:

Párrafos

El texto en una página web se escribe habitualmente en párrafos. En el lenguaje HTML la etiqueta <p> se usa para delimitar los párrafos.

Títulos

En HTML podemos seleccionar seis niveles de títulos diferentes. Niveles que le pueden decir al navegador: «este es un título muy importante», «este es un título un poco menos importante», «este es un título mucho menos importante», etc. Por consiguiente, tienes seis etiquetas de título diferentes:

- <h1> </h1>: significa «título muy importante». En general se usa para mostrar el título de la página al comienzo de la página.
- <h2> </h2>: significa "título importante".
- <h3> </h3>: asimismo, esto significa un título un poco menos importante (podéis decir «subtítulo» si queréis).
- <h4> </h4>: el título es incluso menos importante.
- <h5> </h5>: el título no es importante.
- <h6> </h6>: el título no es nada importante.

En HTML:

Ejemplo de uso - elementos h1..h6:

taskDesafío: ¿Te animas a crear tu primera página web? Puedes comenzar con el ejemplo demostrado arriba o bien, crear una nueva página web a partir de una temática que te guste.

4.2. Listas

Las listas normalmente nos permiten organizar el texto y ordenar la información.

Vamos a descubrir dos tipos de listas aquí:

- listas no ordenadas o listas con viñetas;
- listas ordenadas o listas numeradas.

Listas no ordenadas

Una lista no ordenada tiene esta apariencia:

- Fresas
- Frambuesas
- Cerezas

Es un sistema que nos permite crear una lista sin ningún concepto de orden (no hay «primero» o «último»). Crear una lista no ordenada es muy sencillo. Simplemente usar la etiqueta `` cerrada posteriormente con ``.

Entonces, empezar introduciendo esto:

Y ahora esto es lo que vamos a hacer: vamos a escribir cada elemento de la lista dentro de las etiquetas ``. Cada una de estas etiquetas tienen que estar situadas entre `` y ``.

Van a entender esto de inmediato con este ejemplo:

En HTML:

En el Navegador:

Listas ordenadas

Una lista ordenada funciona de la misma forma únicamente cambia la etiqueta: tienes que sustituir `` por ``.

No cambies nada dentro de la lista: siempre tienes que usar las etiquetas `` para delimitar los elementos de la lista.

Cómo es especialmente intuitivo este ejemplo (los resultados se muestran en la figura de abajo):

videoA continuación, te invitamos a mirar el siguiente video que explica cómo trabajar listas:

HTML Listas



taskDesafío: Accede al siguiente enlace:<https://stackblitz.com/edit/web-platform-cschrw?file=index.html> e intenta modificar la lista "Mí día" a fin de:

- ordenar en base a letras en orden alfabético
- ordenar en base a letras pero en orden inverso al alfabético
- ordenar en base a letras pero comenzando por la letra c
- convertir en lista no ordenada.
- convertir en lista no ordenada con una viñeta diferente a la predefinida "disc"

4.3. Resaltar

Algunas palabras en los párrafos son algunas veces más importantes que otras y por lo que, es posible que deseemos resaltarlas. HTML proporciona varias formas para resaltar el texto de la página.

Enfatizar

Para **enfatizar** el texto, debemos que usar el elemento ** **.

Es muy simple de usar, **¡todo lo que tienes que hacer es encerrar las palabras a enfatizar dentro de estas etiquetas!**

Ejemplo:

Enfatizar considerablemente

Para **enfatizar** considerablemente un texto, puedes usar la etiqueta **** que significa «énfasis fuerte» o si prefieres «importante». Se usa exactamente de la misma forma que ****:

Sin lugar a dudas verás que el texto aparecerá en negrita. También en este caso la negrita es solo una *consecuencia*. El navegador eligió mostrar las palabras importantes en negrita para hacer que destaque más.

La etiqueta no significa «negrita» sino «importante»

Marcado de Texto

La etiqueta **<mark>** se usa para resaltar visualmente una parte del texto. El extracto no se tiene que considerar necesariamente importante, pero queréis que destaque del resto del texto. Esto puede ser útil para resaltar texto que sea relevante, por ejemplo después de buscar en vuestra página web.

Por definición **<mark>** tiene el efecto de resaltar el texto.

4.4. Enlaces

Como sabemos un sitio web está formado por varias páginas. ¿Cómo se va de una página a otra? **¡Usando enlaces por supuesto!**

Los enlaces se reconocen fácilmente en una página dado que están escritos de una forma diferente (de forma predeterminada en azul y subrayados) y un cursor que se parece a una mano aparece cuando se pone el mouse sobre ellos.

El elemento HTML `<a>` permite crear un enlace a otras páginas de la web, archivos o ubicaciones dentro de la misma página, direcciones de correo, o cualquier otra URL que especifiquemos en sus atributos.

Ejemplo:

```
1 <a href="http://www.google.com">Google</a>
```

Por defecto el enlace se resalta en azul. Si ya abrieron la página el enlace se muestra en morado. Veremos cómo cambiar este aspecto cuando avancemos en CSS.

videoA continuación, te invitamos a ver un video explicativo sobre enlaces:



Dentro del atributo href la URL puede escribirse de forma absoluta (incluyendo el dominio) o relativa (sin incluir el dominio) solo para enlaces dentro del mismo dominio. Tanto de una forma u otra, la ruta de carpetas (path) debe especificarse.

taskDesafío: En HTML, existen a grandes rasgos 3 formas de trabajar los enlaces: enlaces internos, externos y marcadores o hipervínculos. ¿Cuál es la diferencia entre ellos? ¿Cómo se implementan? Luego de responder las preguntas, confecciona una página web a modo de ejemplo que incluya los tres tipos.

4.5. Imágenes

En los sitios web se pueden utilizar varios **formatos** de imágenes diferentes y no debemos elegirlos al azar. De hecho, a veces las imágenes son grandes y tardan mucho en descargarse, lo que ralentiza el tiempo de carga de la página (¡mucho más que el texto!).

Entonces, debes asegurarte de que tus páginas son legibles y rápidas de descargar.



Desafío: Investiga ¿Cuáles son los formatos de imágenes recomendados para la web? ¿Por qué?

Insertar una imagen

¿Cuál es la famosa etiqueta que utilizamos para insertar una imagen? ¡Es...!

Es una etiqueta **huérfana** (como
). Esto quiere decir que no tienes que introducir dos etiquetas como en el caso de la mayoría de etiquetas que hemos visto hasta ahora. De hecho, no tenemos que definir una porción de texto; lo único que queremos es colocar una imagen en un punto específico.

La etiqueta debe ir acompañada de dos atributos obligatorios:

- **src:** que indica la ubicación en la que quieras insertar la imagen. Puedes incluir una ruta absoluta (e.g.<http://www.website.com/flor.png>) o una ruta relativa (que es lo que suele hacerse). Así que si la imagen está en la subcarpeta imágenes, tienes que introducir: src="imágenes/flor.png"
- **alt:** que significa "texto alternativo". Un texto alternativo a la imagen, en otras palabras, un texto corto que describe lo que contiene la imagen, debe indicarse *siempre*. Este texto se mostrará en lugar de la imagen si esta última no se puede descargar (esto ocurre), o en los navegadores de personas con discapacidad visual que lamentablemente no pueden "ver" la imagen. También ayuda a los robots de motores de búsqueda a buscar imágenes. Para la flor, por ejemplo, se debería escribir lo siguiente: alt="Una flor"..

Añadir una descripción emergente

El atributo utilizado para mostrar una descripción emergente es el mismo que para los enlaces: title. Este atributo es opcional (al contrario que alt).

Ejemplo:

```
1 <p>
2 Esta es una foto que hice durante mis pasadas vacaciones en las montañas:<br />
3 
4 </p>
```

Nota: Coloca el puntero del mouse sobre la foto para que aparezca la descripción emergente.

videoA continuación, te invitamos a ver un video explicativo:

HTML Imágenes



4.6. Tablas

Una tabla simple

Uno de los elementos HTML que también nos interesa es <table> </table>.

La etiqueta <table> es de bloque, por lo que deberá que situarse fuera de un párrafo.

Ejemplo:

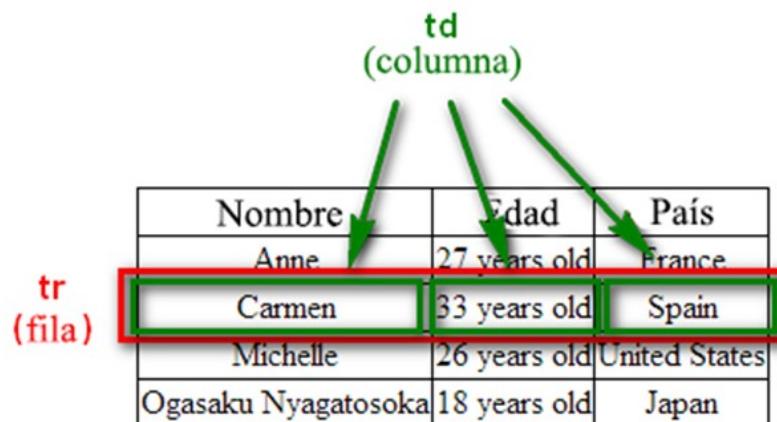
```
1 <p>Esto es un párrafo anterior a la tabla.</p>
2
3 <table>
4     <!-- Inserta aquí el contenido de la tabla --&gt;
5 &lt;/table&gt;
6
7 &lt;p&gt;Esto es un párrafo posterior a la tabla.&lt;/p&gt;
8</pre>
```

A esta, se suman otras dos muy importantes:

- <tr> </tr>: define el comienzo y el final de una fila de la tabla.
- <td> </td>: define el comienzo y el final del contenido de una celda.

En HTML, las tablas se crean fila por fila. En cada fila (<tr>) se especifican los contenidos de las celdas (<td>).

Una tabla se construye básicamente tal y como se muestra en la siguiente ilustración.



Contiene una etiqueta de fila (<tr>) que incluye un grupo de celdas (<td>)

Por ejemplo, si deseo crear una tabla con dos filas y tres celdas por fila (tres columnas, por tanto) deberé escribir lo siguiente:

```
1 <table>
2   <tr>
3     <td>Carmen</td>
4     <td>33 years old</td>
5     <td>Spain</td>
6   </tr>
7   <tr>
8     <td>Michelle</td>
9     <td>26 years old</td>
10    <td>United States</td>
11  </tr>
12 </table>
```

El resultado es un poco lúgubre (siguiente ilustración).

Carmen 33 years old Spain
Michelle 26 years old United States

La fila de encabezado

Ahora que tenemos lo que buscábamos, vamos a añadir la fila de encabezado a la tabla. En el siguiente ejemplo, los encabezados son «Nombre», «Edad» y «País».

La fila de encabezado se ha creado con una etiqueta `<tr>` como hemos visto hasta ahora, pero las celdas que contiene están esta vez enmarcadas por etiquetas `<th>` y no `<td>`.

```
1 <table>
2   <tr>
3     <th>Nombre</th>
4     <th>Edad</th>
5     <th>País</th>
6   </tr>
7
8   <tr>
9     <td>Carmen</td>
10    <td>33 años</td>
11    <td>España</td>
12  </tr>
13  <tr>
14    <td>Michelle</td>
15    <td>26 años</td>
16    <td>Estados Unidos</td>
17  </tr>
18 </table>
```

La fila de encabezado es fácil de reconocer por dos razones:

- las celdas contienen etiquetas `<th>`, en lugar de las habituales etiquetas `<td>`;
- es la primera fila de la tabla (hay que remarcarlo, aunque pueda parecer muy obvio).

Título de tabla

Normalmente, todas las tablas deben tener título. El título informa rápidamente al visitante sobre el contenido de la tabla.

Nuestro ejemplo es tan solo una lista de gente.... pero *¿y qué? ¿Qué representa?* Sin un título para la tabla, estamos un poco perdidos, como puedes ver.

Existe la etiqueta `<caption>`

Esta etiqueta se coloca al comienzo de la tabla, justo antes del encabezado. Es la etiqueta que contiene el título de la tabla (imagen siguiente)

```
1 <table>
2   <caption>Passengers of flight 377</caption>
3
4   <tr>
5     <th>Name</th>
6     <th>Age</th>
7     <th>Country</th>
8   </tr>
9
10  <tr>
11    <td>Carmen</td>
12    <td>33 years old</td>
13    <td>Spain</td>
14  </tr>
15  <tr>
16    <td>Michelle</td>
17    <td>26 years old</td>
18    <td>United States</td>
19  </tr>
20 </table>
```

Passengers of flight 377

Name	Age	Country
Carmen	33 years old	Spain
Michelle	26 years old	United States

Una tabla con título

Una tabla estructurada

Vamos a descubrir dos técnicas especiales:

Las tablas grandes pueden **dividirse** en tres partes:

- Encabezado;
 - Cuerpo de la tabla;
 - Pie de tabla.
- En algunas tablas, puede que necesites **combinar** celdas.

Dividir una tabla grande

Si la tabla es bastante grande, lo mejor es dividirla en varias partes. Para esto, las etiquetas HTML se utilizan para definir los tres «campos» de la tabla:

- **el encabezado (al principio)**: se define con las etiquetas `<thead></thead>`;
- **el cuerpo (en el centro)**: se define con las etiquetas `<tbody></tbody>`;
- **el pie de tabla (al final)**: se define con las etiquetas `<tfoot></tfoot>`.

¿Qué deberías poner en el pie de tabla? Si es una tabla larga, normalmente copiarás los encabezados. Esto te permite ver a lo que se refiere cada columna, incluso al final de la tabla. Fundamentalmente, una tabla se divide en tres partes como en el ejemplo siguiente.

Passengers of flight 377		
Name	Age	Country
Carmen	33 years old	Spain
Michelle	26 years old	United States
Francois	43 years old	France
Martine	34 years old	France
Jonathan	13 years old	Australia
Xu	19 years old	China
Name	Age	Country

Una tabla dividida en varias partes

Es un poco confuso, pero es recomendable escribir las etiquetas en el orden siguiente:

1. <head>
2. <tbody>
3. <tfoot>

Por lo tanto, en el código, primero debemos definir la parte superior, luego la parte final y, por último, la parte principal (<tbody>). No te preocunes, el navegador se ocupará de mostrar cada elemento en el lugar adecuado.

Por lo tanto, este es el código que tienes que introducir para hacer una tabla en tres partes:

```
1 <table>
2   <caption>Pasajeros del vuelo 377</caption>
3
4   <thead> <!-- Pasajeros del vuelo 377 -->
5     <tr>
6       <th>Nombre</th>
7       <th>Edad</th>
8       <th>País</th>
9     </tr>
10    </thead>
11
12  <tfoot> <!-- Pie de tabla -->
13    <tr>
14      <th>Nombre</th>
15      <th>Edad</th>
16      <th>País</th>
17    </tr>
18  </tfoot>
19
20  <tbody> <!-- Cuerpo de la tabla -->
21    <tr>
22      <td>Carmen</td>
23      <td>33 años</td>
24      <td>España</td>
25    </tr>
26    <tr>
27      <td>Michelle</td>
28      <td>26 años</td>
29      <td>Estados Unidos</td>
30    </tr>
31    <tr>
32      <td>François</td>
33      <td>43 años</td>
34      <td>Francia</td>
35    </tr>
36    <tr>
37      <td>Martine</td>
38      <td>34 años</td>
39      <td>Francia</td>
40    </tr>
41    <tr>
42      <td>Jonathan</td>
43      <td>13 años</td>
44      <td>Australia</td>
45    </tr>
46    <tr>
47      <td>Xu</td>
48      <td>19 años</td>
49      <td>China</td>
50    </tr>
51  </tbody>
52 </table>
```

IMPORTANTE

No tienes que utilizar necesariamente estas tres etiquetas (`<thead>`, `<tbody>`, `<tfoot>`) en todas las tablas. De hecho, las utilizarás principalmente si la tabla es muy grande y necesitas organizarla de forma más clara.

En tablas «pequeñas», puedes mantener fácilmente la organización más simple que vimos al principio.

[Combinar](#)

En algunas tablas complejas, necesitarás «combinar» celdas.

¿Un ejemplo de combinar? El ejemplo de la tabla a continuación proporciona una lista de películas y su audiencia prevista.

Movie title	For children?	For teenagers?
Chainsaw massacre	No, too violent	Yes
The care bears go skiing	Yes, suitable	Not violent enough...
Lucky Luke, goes it alone	For the whole family!	

Una tabla que contiene títulos de películas y su audiencia

En la última película, puedes ver que las celdas se han combinado en una. Es exactamente el efecto que estamos intentando conseguir.

Para combinar, se añade un atributo a la etiqueta <td>. Hay dos tipos de combinar:

- **Combinar columnas:** es lo que acabamos de hacer en este ejemplo. Las celdas se combinan horizontalmente.

Se utiliza el atributo colspan.

- **Combinar filas:** en este caso, dos filas se combinan en una. Las celdas se combinan verticalmente.

Se utiliza el atributo rowspan.

Como sabes, tienes que darle un valor al atributo (ya sea colspan o rowspan). Tienes que especificar el número de celdas que quieras combinar. En nuestro ejemplo, hemos combinado dos celdas: la celda de la columna «¿Para niños?» y la celda de la columna «¿Para adolescentes?». De esta manera, tenemos que introducir:

```
1 <td colspan="2">
```

Esto significa: «Esta celda es la combinación de dos celdas». Puedes combinar varias celdas a la vez (tres, cuatro, cinco, etc. tantas como quieras).

Aquí está el código HTML que combina las celdas correspondientes a la tabla de arriba

```
1 <table>
2   <tr>
3     <th>Título de la película</th>
4     <th>¿Para niños?</th>
5     <th>¿Para adolescentes?</th>
6   </tr>
7   <tr>
8     <td>La matanza de Texas</td>
9     <td>No, demasiado violenta</td>
10    <td>Sí</td>
11  </tr>
12  <tr>
13    <td>Los osos amorosos se van a esquiar</td>
14    <td>Sí, apropiada</td>
15    <td>No hay violencia suficiente...</td>
16  </tr>
17  <tr>
18    <td>Lucky Luke el intrépido</td>
19    <td colspan="2">¡Para toda la familia!</td>
20  </tr>
21 </table>
```

Nota importante: puedes ver que la línea 19 solo contiene dos celdas en lugar de tres (solo hay dos etiquetas <td>). Es bastante normal, ya que he combinado las últimas dos celdas. <td colspan="2"> especifica que esta celda ocupa el lugar de

dos celdas a la vez.

¿Y cómo se procede para combinar verticalmente con rowspan?

Es un poco más complicado. En nuestro ejemplo, vamos a «revertir» el orden de nuestra tabla: en lugar de poner los títulos de las películas a la izquierda, los pondremos arriba.

Es otra forma de ver la tabla: en lugar de hacerla en altura, la hacemos en anchura.

En este caso colspan deja de ser adecuado y se debería usar rowspan:

```
1 <table>
2   <tr>
3     <th>Título de la película</th>
4     <td>La matanza de Texas</td>
5     <td>Los osos amorosos se van a esquiar</td>
6     <td>Lucky Luke el intrépido</td>
7   </tr>
8   <tr>
9     <th>¿Para niños?</th>
10    <td>No, demasiado violenta</td>
11    <td>Sí, apropiada</td>
12    <td rowspan="2">¡Para toda la familia!</td>
13  </tr>
14  <tr>
15    <th>¿Para adolescentes?</th>
16    <td>Sí</td>
17    <td>No hay violencia suficiente...</td>
18  </tr>
19 </table>
```

Resultado: ¡las celdas se combinan verticalmente (ejemplo debajo)!

Film title	Chainsaw massacre	The care bears go skiing	Lucky Luke, goes it alone
For children?	No, too violent	Yes, suitable	
For adolescents?	Yes	Not violent enough...	For the whole family!

Las celdas se han combinado verticalmente

task**Desafío:** Compartimos un enlace: <https://creatuweb.espaciolatino.com/ejercicios/tablas/index.html> con ejercicios en relación a tablas. Intenta resolver.

En resumen

- Una tabla se inserta con la etiqueta <table> y se define fila por fila utilizando la etiqueta <tr>.
- Cada fila contiene celdas <td> (celdas normales) o <th> (celdas de encabezado).
- El título de la tabla se define mediante <caption>.
- Puedes añadir un borde a las celdas de la tabla mediante border. Para combinar bordes, se usa la propiedad de CSS border-collapse.
- Una tabla puede dividirse en tres secciones: <thead> (encabezado), <tbody> (cuerpo) y <tfoot> (pie de la tabla). El uso de estas etiquetas no es obligatorio.
- Puedes combinar celdas horizontalmente mediante el atributo colspan o verticalmente con rowspan. Tienes que especificar cuántas celdas tienen que combinarse.

4.7. Formularios

Cualquier página HTML puede mejorarse con formularios interactivos que piden al visitante que rellene información: introducir texto, seleccionar opciones, confirmar con un botón... ¡cualquier cosa es posible!

Sin embargo, estamos alcanzando los límites del lenguaje HTML, ya que después deberemos analizar la información que el visitante ha introducido... y eso no se puede hacer en HTML. Como vamos a ver, los resultados tienen que procesarse en otro lenguaje, como PHP, por ejemplo.

Mientras tanto, tenemos un gran número de etiquetas HTML por descubrir. **Bienvenido al maravilloso mundo de los formularios.** *Un mundo en el que los botones, las casillas de verificación y las listas desplegables coexisten en armonía.*

Crear un formulario

Cuando tienes la repentina necesidad de insertar un formulario en tu página HTML, necesitas empezar escribiendo una etiqueta <form> </form>. Esta es la etiqueta principal de los formularios, y especifica su comienzo y su fin.

```
1 <p>Texto antes del formulario</p>
2
3 <form>
4   <p>Texto dentro del formulario</p>
5 </form>
6
7 <p>Texto después del formulario</p>
```

Debes tener en cuenta que debes introducir etiquetas de bloque (como <p> </p>) dentro del formulario si quieres incluir texto en él.

Veamos un ejemplo para clarificar las cosas. Supón que un visitante acaba de incluir un comentario en el formulario, como, por ejemplo, un mensaje que les gustaría publicar en el foro. Este mensaje tiene que *enviarse* para que puedas recibirla (lógico, ¿verdad?) y mostrarlo a otros visitantes.

Bien, ese es el problema o, más bien, esos son los problemas, que vamos a abordar:

- **Problema n.º 1:** ¿Cómo enviar el texto introducido por el visitante? ¿De qué manera?
- **Problema n.º 2:** una vez que los datos se han enviado, ¿Cómo los procesas? ¿quieres recibir el mensaje automáticamente mediante correo electrónico, o prefieres un programa que lo guarde en algún sitio y después lo muestre en una página que todo el mundo lo pueda ver?

Para aportar las soluciones a estos dos problemas, tienes que añadir dos atributos a la etiqueta <form>:

- **method:** este atributo especifica cómo se van a enviar los datos (solución al **problema n.º 1**). Hay dos maneras de enviar datos a la Web:
 - **method="get":** este método normalmente es menos adecuado, ya que se limita a 255 caracteres. Esta particularidad proviene del hecho de que la información se enviará a la dirección de la página (<http://...>, aunque este detalle en realidad no tiene importancia por el momento. Te recomiendo que utilices la mayoría de las veces el otro método: post).
 - **method="post":** es el método más usado en formularios, puesto que permite que se envíe un montón de información. Los datos introducidos en el formulario no van a través de la barra de direcciones.
- **action:** es la dirección de la página o programa que va a procesar la información (solución al **problema n.º 2**). Esta página te enviará un correo electrónico con el mensaje si eso es lo que quieras o, por otro lado, guardará el mensaje con todos los demás mensajes en una base de datos. Esto no se puede hacer en HTML y CSS, así que normalmente utilizamos otro lenguaje. Ej. java, php, python, etc.

Por lo tanto, ahora vamos a completar la etiqueta <form> con los dos atributos que acabamos de ver.

Para method, lo has adivinado, vamos a introducir el valor post.

Para action, vamos a introducir el nombre de una página ficticia en PHP (tratamiento.php). Esta es la página que será solicitada cuando el visitante haga clic en el botón de envío en el formulario.

```
1 <p>Texto antes del formulario</p>
2
3 <form method="post" action="tratamiento.php">
4   <p>Texto dentro del formulario</p>
5 </form>
6
7 <p>Texto después del formulario</p>
```

Por el momento, no abordaremos lo que pasa dentro en tratamiento.php. Asumimos que esta página está lista y funcionando.

Nuestra prioridad, por ahora, es descubrir en HTML/CSS cómo insertar campos de texto, botones y casillas de verificación en nuestra página web. Y eso es lo que vamos a ver ahora.

Campos de entrada básicos

Por lo tanto, regresemos a algo tangible.

Vamos a revisar varias etiquetas HTML que necesitamos para introducir texto en un formulario.

Deberías saber que hay dos campos de texto diferentes.

- **El campo de texto de una línea:** como su nombre sugiere, solo puedes introducir una única línea en él. Se utiliza para textos cortos, por ejemplo, un nombre de usuario.
- **El campo de texto de líneas múltiples:** este campo de texto permite introducir un gran cantidad de texto en varias líneas, por ejemplo, una tesis sobre el uso de HTML en el desarrollo de países del Sudeste de Asia (tan solo una sugerencia...).

Campo de texto de una línea

Esta imagen a continuación muestra el aspecto de un campo de texto de una línea.



Un campo de texto de una línea

Para insertar un campo de texto en una línea, vamos a utilizar la etiqueta <input type="text" />.

Para insertar un campo de texto en una línea, vamos a utilizar la etiqueta <input type="text" />.

Encontraremos esta etiqueta varias veces más adelante del capítulo. En cada caso, el que cambia es el valor de type.

Para crear un campo de texto de una línea, tienes que introducir:

```
1 <input type="text" ></code>
```

Para dar un nombre a un elemento de formulario, se utiliza el atributo name. En este caso, vamos a asumir que al visitante se le ha pedido que vuelva a introducir su nombre de usuario:

```
1 <input type="text" name="nombre" ></code>
```

Así que vamos a intentar crear un formulario muy básico con nuestro campo de texto:

```
1 <form method="post" action="tratamiento.php">
2   <p><input type="text" name="username" /></p>
3 </form>
```

Etiquetas

Este campo de texto está muy bien, pero si tu visitante se lo encuentra, no sabrá lo que poner. Esta es exactamente la labor de la etiqueta <label>:

```
1 <form method="post" action="tratamiento.php">
2   <p>
3     <label>Nombre usuario</label> : <input type="text" name="username" />
4   </p>
5 </form>
```

Este código proporciona exactamente el mismo resultado que has visto en la imagen anterior.

Pero no es suficiente. Hay que enlazar la etiqueta con el campo de texto.

¿Un name y un id en el campo? ¿No sería redundante?

Sí, de cierta manera pero no supone ningún problema.

Para enlazar la etiqueta con el campo, se le tiene que dar un atributo for, que tiene el mismo valor que el campo id... Lo mejor es verlo con un ejemplo:

```
1 <form method="post" action="tratamiento.php">
2   <p>
3     <label for="username">Nombre usuario</label> : <input type="text" name="username" id="username"
4   />
5   </p>
6 </form>
```

Algunos atributos adicionales

Puedes añadir varios atributos adicionales a la etiqueta <input/> para personalizar su funcionamiento:

- Puedes aumentar el tamaño del campo con size.
- Puedes limitar el número de caracteres que pueden introducirse mediante maxlength.
- Puedes prellenar el campo con un valor por defecto con value.
- Puedes dar algún indicio de los contenidos del campo mediante placeholder. Esta indicación desaparecerá tan pronto como el visitante haga clic dentro del campo.

En el siguiente ejemplo, el campo de texto contiene una indicación para entender lo que debería escribirse; aunque el campo mide 30 caracteres, no puedes escribir más de 10 dentro:

```
1 <form method="post" action="tratamiento.php">
2   <p>
3     <label for="username">Nombre usuario:</label>
4     <input type="text" name="username" id="username" placeholder="Zozor" size="30" maxlength="10"
5   />
6   </p>
7 </form>
```

En el navegador:

Nombre usuario

Un campo de texto con un indicador (marcador de posición)

Campo de contraseña

Puedes hacer fácilmente que el campo de texto se comporte como un «campo de contraseña», es decir, un campo donde no puedes ver en la pantalla los caracteres que se escriban. Para crear este tipo de campo de entrada, utiliza el atributo type="password".

Completaré mi formulario. Ahora le pide al visitante que escriba su nombre de usuario y su contraseña:

```
1 <form method="post" action="tratamiento.php">
2   <p>
3     <label for="username">Tu nombre de usuario:</label>
4     <input type="text" name="username" id="username" />
5
6     <br />
7     <label for="pass">Tu contraseña:</label>
8     <input type="password" name="pass" id="pass" />
9
10  </p>
11 </form>
```

Verás que los caracteres de la contraseña no se visualizan en pantalla, como se muestra en la siguiente imagen.

Nombre usuario	Zozor
Contraseña	*****
Un campo de entrada de contraseña	

Campo de texto de líneas múltiples

Para crear un campo de texto de líneas múltiples, cambia la etiqueta: vamos a utilizar <textarea> </textarea>.

Como cualquier otro elemento de formulario, tienes que proporcionarle un nombre mediante name y utilizar label para explicar qué es.

```
1 <form method="post" action="tratamiento.php">
2   <p>
3     <label for="mejora">¿Cómo crees que podría mejorar mi página web?</label><br />
4     <textarea name="mejora" id="mejora"></textarea>
5   </p>
6 </form>
```

¡Y aquí está la imagen con el resultado (siguiente captura)!

¿Cómo crees que podría mejorar mi página web?

Un campo de entrada pequeño de líneas múltiples

Como puedes ver, ¡es un poco pequeño! Afortunadamente, puedes cambiar el tamaño de <textarea> de dos maneras distintas.

- **En CSS:** tan solo aplica las propiedades CSS width y height a <textarea>.
- **Con atributos:** puedes añadir los atributos rows y cols a la etiqueta <textarea>. El primero especifica el número de líneas de texto que pueden mostrarse simultáneamente y el segundo, el número de columnas.

¿Por qué abres la etiqueta <textarea> y la cierras de nuevo inmediatamente después?

Puedes prerrellenar <textarea> con un valor por defecto. En este caso, no utilizamos el atributo value: simplemente escribimos el texto por defecto entre la etiqueta de apertura y la de cierre.

```
1 <form method="post" action="tratamiento.php">
2   <p>
3     <label for="mejora">
4       ¿Cómo crees que podría mejorar mi página web?
5     </label>
6     <br />
7
8     <textarea name="mejora" id="mejora" rows="10" cols="50">
9       ¡Mejorar tu página web!<br/>
10      ¡Venga! ¡Es tan perfecta que no necesita ninguna mejora!
11    </textarea>
12  </p>
13 </form>
```

La siguiente imagen muestra el resultado.

¿Cómo crees que podría mejorar mi página web?

¡Mejorar tu página web!
¡Venga! ¡Es tan perfecta que no necesita ninguna mejora!

Un campo de entrada de líneas múltiples completado previamente

Campos de entrada mejorados

HTML5 incluye muchas características nuevas en los formularios. De hecho, han aflorado nuevos tipos de campos con esta versión. Tan solo dale al atributo type de la etiqueta <input/> uno de los nuevos valores disponibles. ¡Echemos un vistazo rápido!

Todos los navegadores todavía no reconocen estos campos de entrada mejorados. En su lugar, los navegadores antiguos mostrarán un campo de texto simple de una sola línea (como si hubieras escrito type="text"). Que quede entre nosotros, no pasa nada: los nuevos navegadores pueden disfrutar de las últimas funciones, mientras que los navegadores antiguos muestran un campo de texto de sustitución que encaja igual de bien.

¡Así que te recomendamos usar estos nuevos campos de entrada desde ahora! En el mejor de los casos, tus visitantes se beneficiarán de estas nuevas funciones y, en el peor, no verán ningún problema.

Correo electrónico

Puedes pedir a los visitantes que introduzcan una dirección de correo electrónico:

```
1 <input type="email" ></code>
```

El campo *normalmente* te parecerá el mismo, pero el navegador ahora sabe que el usuario tiene que escribir una dirección de correo electrónico. Puede dar una indicación si la dirección no es un correo electrónico, que es lo que hace Firefox (ejemplo a continuación).

dsqfdssdfsp

Un correo electrónico escrito incorrectamente aparece enmarcado en rojo.

Una URL

Con el tipo url, puedes pedirle al visitante que escriba una dirección absoluta (normalmente comienza con http://):

```
1 <input type="url" ></code>
```

Mismo principio: si el campo no parece diferente en tu ordenador, ten en cuenta que ha entendido eficazmente que se supone que el visitante debe escribir una dirección.

Número de teléfono

Este campo se usa para escribir números de teléfono.

Número

Este campo se utiliza para escribir un número entero:

```
1 <input type="number" ></code>
```

El campo normalmente se muestra con unas pequeñas flechas para cambiar el valor (imagen siguiente).



Puedes personalizar la manera en la que funciona el campo con los siguientes atributos:

- min: valor mínimo permitido.
- max: valor máximo permitido.
- step: es el desplazamiento. Si especificas un «step» de 2, el campo solo aceptará valores de 2 en 2 (por ejemplo: 0, 2, 4, 6, etc.).

Fecha

Hay varios tipos de campos de selección de fecha:

- date: para la fecha (05/08/1985, por ejemplo);
- time: para la hora (13:37, por ejemplo);
- week: para la semana;
- month: para el mes;
- datetime: para la fecha y la hora (con administración de zona horaria);
- datetime-local para la fecha y la hora (sin administración de zona horaria).

```
1 <input type="date" ></code>
```

Manejo de Opciones

HTML incluye un montón de opciones que pueden usarse en el formulario. Son elementos que requieren que el visitante elija a partir de una lista de posibilidades. Vamos a repasar:

- casillas de verificación;
- campos de opciones;
- listas desplegables.

Casillas de verificación

¿Cómo se crea una casilla de verificación? ¡No podría ser más fácil! Vamos a utilizar la etiqueta <input />, pero esta vez especificando el tipo checkbox:

```
1 <input type="checkbox" name="opciones" ></code>
```

¡Añade <label> en la posición correcta, ¡y ya lo tienes!

```
1 <form method="post" action="tratamiento.php">
2   <p>
3     Marca las comidas que te gustan:<br/>
4     <input type="checkbox" name="patatas fritas" id="patatas fritas" /> <label for="patatas
      fritas">Patatas fritas</label><br />
5     <input type="checkbox" name="hamburguesa" id="hamburguesa" /> <label
      for="hamburguesa">Hamburguesa</label><br />
6     <input type="checkbox" name="espinacas" id="espinacas" /> <label
      for="espinacas">Espinacas</label><br />
7     <input type="checkbox" name="ostras" id="ostras" /> <label for="ostras">Ostras</label>
8   </p>
9 </form>
```

La siguiente imagen muestra el resultado.

Marca las comidas que te gustan:

- Patatas Fritas
- Hamburguesa
- Espinacas
- Ostras

Casillas de verificación

No olvides dar un nombre diferente a cada casilla de verificación. Más tarde, te permitirá identificar las que ha seleccionado el visitante.

Por último, recuerda que puedes tener una casilla marcada por defecto con el atributo checked:

```
1 <input type="checkbox" name="opciones" checked ></code>
```

Todos los atributos normalmente tienen un valor. Sin embargo, en ese caso, añadir un valor no es necesario: la presencia del atributo es suficiente para hacer que el ordenador entienda que la casilla tiene que estar marcada.

Si te molesta, recuerda que puedes asignar cualquier valor al atributo (algunos administradores de páginas algunas veces escriben checked="checked", pero es un poco redundante). La casilla se marcará sin orden.

Botones de radio

Los campos de opciones te permiten elegir una (y solamente una) opción de una lista de posibilidades. De alguna manera, parecen casillas de verificación pero hay una ligera dificultad adicional: tienen que estar organizadas como grupos. Aunque las opciones en el mismo grupo tienen el mismo (nombre), cada opción tienen que tener un (valor) diferente.

La etiqueta que se utiliza siempre es <input />, esta vez con el valor radio en el atributo type.

El siguiente ejemplo lo aclarará:

```
1 <form method="post" action="tratamiento.php">
2   <p>
3     Indica el grupo de edad al que perteneces:<br />
4     <input type="radio" name="edad" value="menos15" id="menos15" /> <label for="menos15">Menos de
      15 años</label><br />
5     <input type="radio" name="edad" value="entre15-25" id="entre15-25" /> <label for="entre15-
      25">Entre 15 y 25 años</label><br />
6     <input type="radio" name="edad" value="entre25-40" id="entre25-40" /> <label for="entre25-
      40">Entre 25 y 40 años</label><br />
7     <input type="radio" name="edad" value="mas40" id="mas40" /> <label for="mas40">Incluso mayor que
      eso?</label>
8   </p>
9 </form>
```

Lo que nos da el resultado de la siguiente imagen.

Indica el grupo de edad al que perteneces:

- Menos de 15 años
- Entre 15 y 25 años
- Entre 25 y 40 años
- Incluso mayor que eso?

Botones de radio

¿Por qué pusiste el mismo nombre para cada opción? Solo para que el navegador sepa a qué «grupo» pertenece el botón.

Intenta eliminar los atributos name y verás que entonces puedes seleccionar todas las opciones. Sin embargo, no es lo que queremos, por esa razón los «enlazamos» al darles el mismo nombre.

Te darás cuenta de que esta vez hemos elegido una id diferente al name. De hecho, dado que los valores de name son los mismos, no podríamos haber diferenciado las ids (y sabes muy bien que una id tiene que ser única!). Por esta razón hemos elegido dar a id el mismo valor que a value.

Si tienes dos campos de opciones diferentes, tienes que dar un name único a cada grupo, de esta manera:

```
1 <form method="post" action="tratamiento.php">
2   <p>
3     Indica el grupo de edad al que perteneces:<br />
4     <input type="radio" name="edad" value="menos15" id="menos15" /> <label for="menos15">Menos de
15 años</label><br />
5     <input type="radio" name="edad" value="entre15-25" id="entre15-25" /> <label for="entre15-
25">Entre 15 y 25 años</label><br />
6     <input type="radio" name="edad" value="entre25-40" id="entre25-40" /> <label for="entre25-
40">Entre 25 y 40 años</label><br />
7     <input type="radio" name="edad" value="mas40" id="mas40" /> <label for="mas40">Incluso mayor?
</label>
8   </p>
9   <p>
10    ¿En qué continente vives?<br />
11    <input type="radio" name="continente" value="europa" id="europa" /> <label
for="europa">Europa</label><br />
12    <input type="radio" name="continente" value="africa" id="africa" /> <label
for="africa">África</label><br />
13    <input type="radio" name="continente" value="asia" id="asia" /> <label for="asia">Asia</label>
<br />
14    <input type="radio" name="continente" value="america" id="america" /> <label
for="america">América</label><br />
15    <input type="radio" name="continente" value="australia" id="australia" /> <label
for="australia">Australia</label>
16   </p>
17 </form>
```

Listas desplegables

Las listas desplegables son otra manera elegante de seleccionar una opción a partir de varias posibilidades. Funcionan de una manera ligeramente distinta. Vamos a utilizar la etiqueta `<select> </select>`, que especifica el comienzo y el final de la lista desplegable. Añadimos el atributo name a la etiqueta para darle un nombre a la lista.

Después, dentro de la etiqueta `<select> </select>`, vamos a insertar varias etiquetas `<option> </option>` (una para cada opción posible). Añadimos un atributo value a cada uno de ellas para identificar lo que el visitante ha elegido.

Aquí podéis ver un ejemplo de uso:

```
1 <form method="post" action="tratamiento.php">
2   <p>
3     <label for="pais">¿En qué país vives?</label><br />
4     <select name="pais" id="pais">
5       <option value="estadosunidos">Estados Unidos</option>
6       <option value="canada">Canadá</option>
7       <option value="reinounido">Reino Unido</option>
8       <option value="francia">Francia</option>
9       <option value="espana">España</option>
10      <option value="italia">Italia</option>
11      <option value="china">China</option>
12      <option value="japon">Japón</option>
13    </select>
14  </p>
15 </form>
```

El resultado se muestra en la siguiente imagen.

¿En qué país vives?



Lista desplegable

Si quieres que una opción esté seleccionada por defecto, esta vez utiliza el atributo selected:

```
1 <option value="canada" selected>Canadá</option>
```

También puedes agrupar las opciones con la etiqueta <optgroup> </optgroup>. En nuestro ejemplo, *¿por qué no separas países según su continente?*

```
1 <form method="post" action="tratamiento.php">
2   <p>
3     <label for="pais">¿En qué país vives?</label><br />
4     <select name="pais" id="pais">
5       <optgroup label="Europa">
6         <option value="reinounido">Reino Unido</option>
7         <option value="francia">Francia</option>
8         <option value="espana">España</option>
9         <option value="italia">Italia</option>
10        </optgroup>
11        <optgroup label="América">
12          <option value="estadosunidos">Estados Unidos</option>
13          <option value="canada">Canadá</option>
14        </optgroup>
15        <optgroup label="Asia">
16          <option value="china">China</option>
17          <option value="japon">Japón</option>
18        </optgroup>
19      </select>
20  </p>
21 </form>
```

El resultado se muestra en la siguiente imagen.

¿En qué país vives?

United Kingdom ▾

Europe

- United Kingdom
- France
- Spain
- Italy

America

- United States
- Canada

Asia

- China
- Japan

Opciones agrupadas por continente

Los grupos no pueden seleccionarse. Así que no podemos elegir «Europa» en nuestro ejemplo: solo los nombres de países pueden seleccionarse.

```
1 <form method="post" action="tratamiento.php">
2
3     <fieldset>
4         <legend>Tus datos de contacto</legend> <!-- Título del conjunto de campos -->
5
6         <label for="apellidos">¿Cuáles son tus apellidos?</label><br>
7         <input type="text" name="apellido" id="apellido" /><br>
8
9         <label for="nombre">¿Cuál es tu nombre?</label><br>
10        <input type="text" name="nombre" id="nombre" /><br>
11
12        <label for="email">¿Cuál es tu dirección de correo electrónico?</label><br>
13        <input type="email" name="email" id="email" /><br>
14
15    </fieldset>
16
17    <fieldset>
18        <legend>Tu deseo</legend> <!-- Título del conjunto de campos -->
19
20        <p>
21            Pide un deseo que quieras que se haga realidad:<br>
22
23            <input type="radio" name="deseo" value="rico" id="rico" /> <label for="rico">Ser
24            rico</label><br>
25            <input type="radio" name="deseo" value="famoso" id="famoso" /> <label for="famoso">Ser
26            famoso</label><br>
27            <input type="radio" name="deseo" value="inteligente" id="inteligente" /> <label
28            for="inteligente">Ser aún más inteligente</label><br>
29            <input type="radio" name="deseo" value="otro" id="otro" /> <label for="otro">Otro...
30        </label><br>
31        </p>
32
33        <p>
34            <label for="specs">Si has elegido «Otro», especifica cuál:</label><br>
35            <textarea name="specs" id="specs" cols="40" rows="4"></textarea>
36        </p>
37
38    </fieldset>
39
40 </form>
```

El resultado se muestra en la siguiente imagen.

Tus datos de contacto

¿Cuales son tus apellidos?

¿Cual es tu nombre?

¿Cual es tu correo electrónico?

Tu deseo

Pide un deseo que quieras que se haga realidad:

Ser rico
 Ser famoso

Los campos están agrupados

Seleccionar un campo automáticamente

Puedes situar el cursor automáticamente en uno de los campos del formulario con el atributo autofocus. Una vez que el visitante cargue la página, el cursor aparece en este campo.

Por ejemplo, para situar el cursor por defecto en el campo name:

```
1 <input type="text" name="nombre" id="nombre" autofocus></code>
```

Hacer un campo obligatorio

Puedes hacer un campo obligatorio al asignarle el atributo required.

```
1 <input type="text" name="nombre" id="nombre" required></code>
```

Entonces el navegador le indicará al usuario, si el campo está vacío cuando el formulario sea enviado, que necesita completarlo.

El botón de envío

Así que ahora lo que nos queda por hacer es crear el botón de envío. De nuevo, la etiqueta `<input />` viene a nuestro rescate. Hay cuatro versiones:

- `type="submit"`: el botón de envío principal del formulario. Este es el que usarás más a menudo. El usuario será redirigido a la página especificada en el atributo action del formulario.
- `type="reset"`: restablece el formulario
- `type="image"`: equivalente al botón submit, pero esta vez mostrado como una imagen. Añade el atributo src para especificar la URL de la imagen.
- `type="button"`: botón genérico, que no tendrá efecto alguno (por defecto). En general, este botón se maneja en JavaScript para desarrollar acciones en la página. No lo utilizaremos aquí.

Puedes cambiar el texto mostrado dentro de los botones con el atributo value.

Para crear un botón de envío, entonces escribimos, por ejemplo:

Lo que nos da el resultado de la siguiente imagen.

Botón

Un botón de envío

Cuando haces clic en el botón «Enviar», entonces el formulario te lleva a la página especificada en el atributo action. Recuerda, imaginamos una página ficticia: tratamiento.php.

El problema es que no puedes crear esta página exclusivamente en HTML. Tienes que aprender un nuevo lenguaje, como PHP, para ser capaz de «recuperar» la información introducida y decidir lo que hacer con ella.

videoPara finalizar, te invitamos a mirar un video explicativo:



task**Desafío:** En base a lo visto en esta sección, intenta crear el siguiente formulario:

Registro de Programador

Nombre y Apellido

Email

Teléfono

Lenguajes que conoce

Angular

Java

NodeJS

Python

Otro

Nivel de Inglés

Bajo

Medio

Alto

5. HTML 5

Bien, hasta ahora hemos leído y visto ejemplos sobre HTML básico, pero, en la actualidad existe en plena vigencia la versión HTML 5.

Es decir, HTML5 es la versión más reciente de HTML homologada por la W3C.

Una de las principales ventajas de HTML5 es la inclusión de elementos semánticos, o marcadores semánticos, que nos ayudan a definir las distintas divisiones de una página web (<https://www.eniun.com/html5-estructura-basica-elementos-semanticos>). Es decir, los elementos semánticos de HTML5, nos ayudan a identificar cada sección del documento y organizar el cuerpo de una página web de una manera eficiente y estandarizada.

En la siguiente figura observamos el esquema básico de los elementos semánticos introducidos por la última y actual versión HTML5.

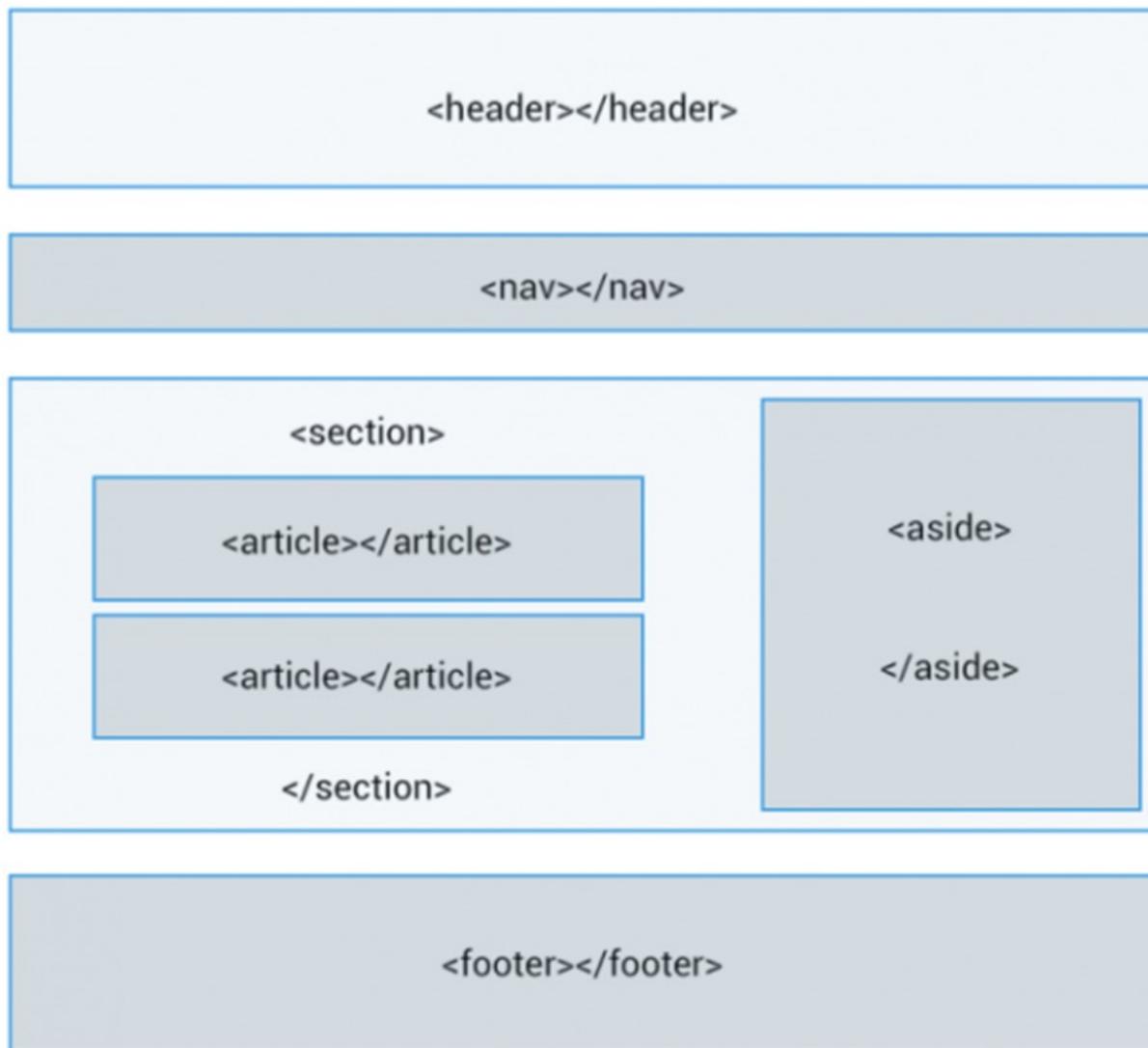


Figura extraída de: <https://www.eniun.com/html5-estructura-basica-elementos-semanticos/>

Según el W3C, una Web Semántica:

"Permite que los datos sean compartidos y reutilizados en todas las aplicaciones, las empresas y las comunidades."

¿Por qué utilizar elementos semánticos?

En HTML4 los desarrolladores usaban sus propios nombres para identificación de los elementos, tales como: cabecera, superior, inferior, pie de página, menú, navegación, principal, contenedor, contenido, artículo, barra lateral, etc. Cómo estos

nombres no estaban estandarizados, se hacía imposible que los motores de búsqueda identificaran el contenido correcto de la página web.

Elementos semánticos = elementos con significado

Un elemento semántico describe claramente su significado tanto para el navegador web como para el desarrollador (https://www.w3schools.com/html/html5_semantic_elements.asp)

Ejemplos de elementos no semánticos: <div>y - No dice nada sobre su contenido.

Ejemplos de elementos semánticos: <form>, <table>y <article>- Definen claramente su contenido.

Descripción de elementos semánticos HTML5

Elemento HTML <section>

El elemento <section> define una sección en un documento. Según la documentación HTML del W3C: "Una sección es una agrupación temática de contenido, normalmente con un encabezado.

Una página web normalmente se puede dividir en secciones para la introducción, el contenido y la información de contacto.

Elemento HTML <article>

El elemento <article> especifica contenido autónomo e independiente.

Un artículo debe tener sentido por sí solo y debe ser posible distribuirlo independientemente del resto del sitio web.

Ejemplos de dónde se puede utilizar un elemento <article>:

- Publicación en el foro
- Entrada en el blog
- Artículo de periódico

Elemento HTML <header>

El elemento <header> sirve para especificar contenido de tipo introductorio o un conjunto de enlaces de navegación.

Un elemento <header> normalmente contiene:

- Uno o más elementos de encabezado (<h1> - <h6>).
- Logo o ícono.
- Información de autoría.

Nota: puede tener varios elementos <header> en un documento HTML. Sin embargo, el elemento <header> no puede ser colocado dentro de elementos <footer>, <address> o dentro de otro elemento <header>.

Elemento HTML <footer>

El elemento <footer> define un pie de página para un documento o sección.

Un elemento <footer> normalmente contiene:

- Información de autoría.
- Información registrada.
- Información de contacto.
- Mapa del sitio.
- Volver a los enlaces superiores.
- Documentos relacionados.

Puede tener varios elementos <footer> en un documento.

Elemento HTML <nav>

El elemento HTML <nav> representa una sección de una página cuyo propósito es proporcionar enlaces de navegación, ya sea dentro del documento actual o a otros documentos. Ejemplos comunes de secciones de navegación son menús, tablas de contenido e índices.

Debemos tener en cuenta que NO todos los enlaces de un documento deben estar dentro de un elemento <nav>. El elemento <nav> está destinado sólo para el bloque principal de enlaces de navegación.

Los navegadores, como los lectores de pantalla para usuarios con capacidades diferentes, pueden utilizar este elemento para

determinar si se debe omitir la representación inicial de este contenido.

Elemento HTML <aside>

El elemento HTML <aside> representa una sección de una página que consiste en contenido que está indirectamente relacionado con el contenido principal del documento. Estas secciones son a menudo representadas como barras laterales o como inserciones y contienen una explicación al margen como una definición de glosario, elementos relacionados indirectamente, como publicidad, la biografía del autor, o en aplicaciones web, la información de perfil o enlaces a blogs relacionados.



Desafío: Investiga *¿Qué es la web semántica? ¿por qué será importante utilizar elementos semánticos?*

5.1. Header

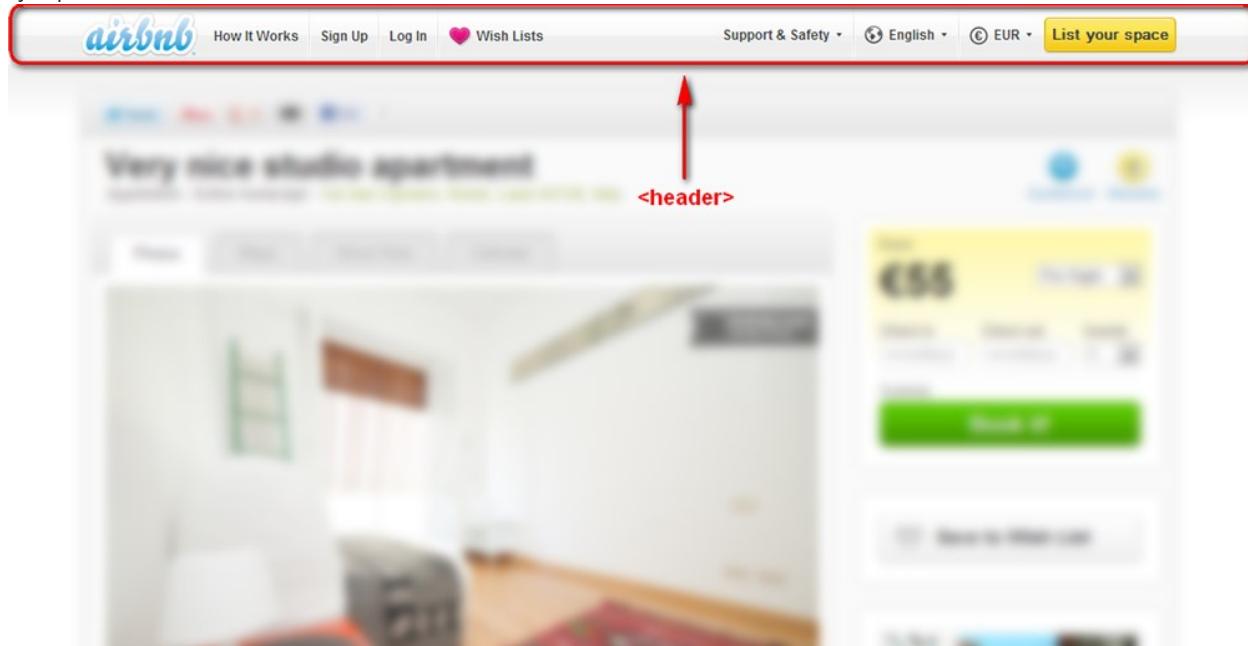
Al igual que las páginas de un libro, la mayoría de las páginas web normalmente tienen un *encabezado*. Dicho encabezado suele contener un logo, un banner, el eslogan de tu web, etc.

El elemento <header> sirve entonces, para especificar contenido de tipo introductorio o un conjunto de enlaces de navegación.

```
1 <header>
2     <!-- Pon tu encabezado aquí -->
3 </header>
```

Nota: puede tener varios elementos <header> en un documento HTML. Sin embargo, el elemento <header> no puede ser colocado dentro de elementos <footer>, <address> o dentro de otro elemento <header>.

Ejemplo:



5.2. Footer

A diferencia del encabezado, el pie de página normalmente está en la parte inferior derecha del documento. Contiene información como enlaces de contacto, nombre del autor, informes legales, etc.

El elemento <footer> define un pie de página para un documento o sección.

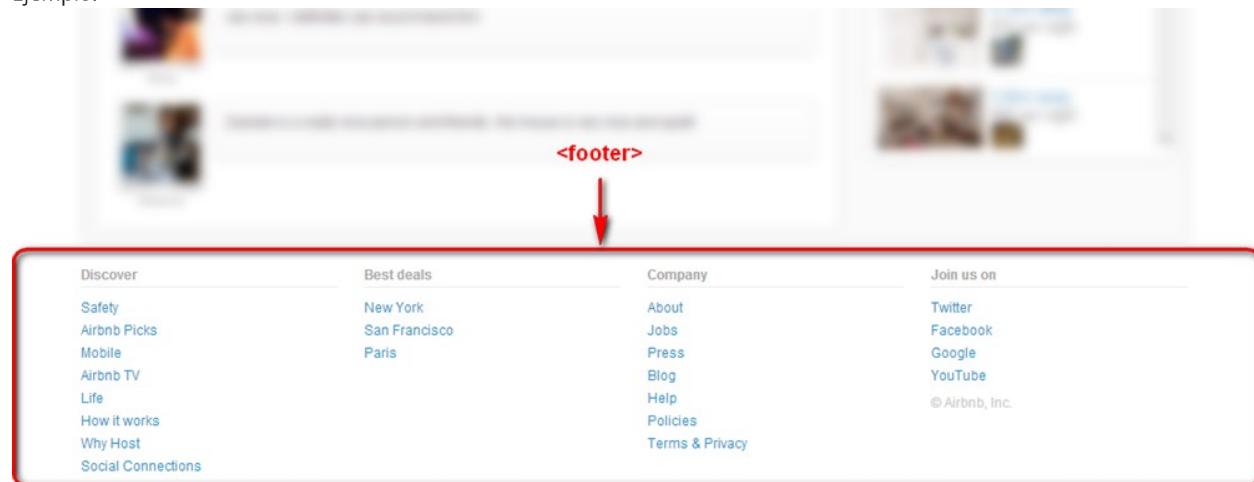
```
1 <footer>
2     
3 </footer>
```

Un elemento <footer> normalmente contiene:

- Información de autoría.
- Información registrada.
- Información de contacto.
- Mapa del sitio.
- Volver a los enlaces superiores.
- Documentos relacionados.

Puede tener varios elementos <footer> en un documento.

Ejemplo:



5.3. Nav

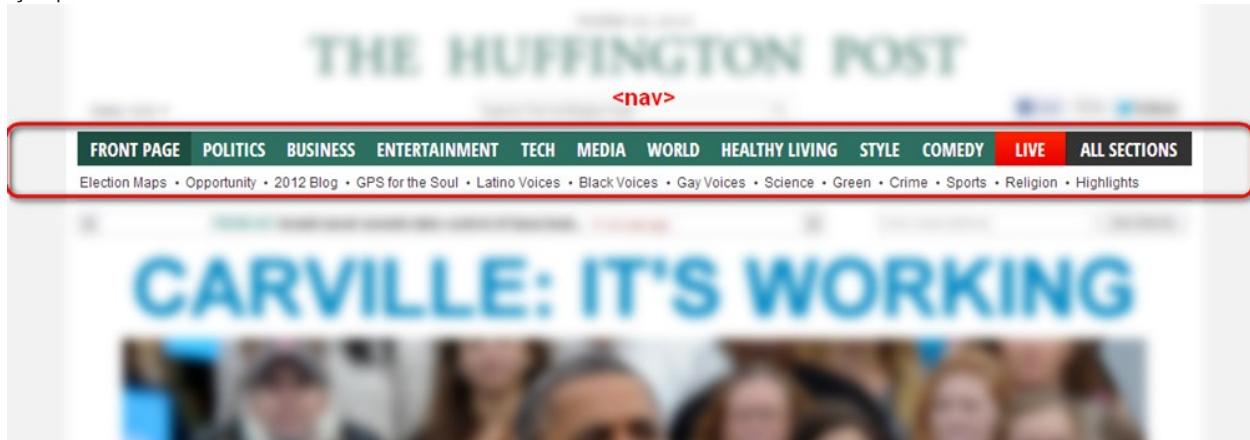
El elemento HTML <nav> representa una sección de una página cuyo propósito es proporcionar enlaces de navegación, ya sea dentro del documento actual o a otros documentos. Ejemplos comunes de secciones de navegación son menús, tablas de contenido e índices.

Debemos tener en cuenta que NO todos los enlaces de un documento deben estar dentro de un elemento <nav>. El elemento <nav> está destinado sólo para el bloque principal de enlaces de navegación.

```
1 <nav>
2   <ul>
3     <li><a href="index.html">Inicio</a></li>
4     <li><a href="forum.html">Foro</a></li>
5     <li><a href="contact.html">Contacto</a></li>
6   </ul>
7 </nav>
```

Los navegadores, como los lectores de pantalla para usuarios con capacidades diferentes, pueden utilizar este elemento para determinar si se debe omitir la representación inicial de este contenido.

Ejemplo:



5.4. Section

El elemento `<section>` define una sección en un documento. Según la documentación HTML del W3C: "Una sección es una agrupación temática de contenido, normalmente con un encabezado.

Una página web normalmente se puede dividir en secciones para la introducción, el contenido y la información de contacto.

```

1 <section>
2   <h1>Sección de mi página</h1>
3   <p>Bla bla bla bla</p>
4 </section>

```

Ejemplo:

En la página de inicio de Wikipedia, hay varios bloques que podrían considerarse secciones de página (figura siguiente).

The screenshot shows the main page of Wikipedia with several sections highlighted by red boxes and labeled with '<section>' at the top right of each box. The sections include:

- Today's featured article**: A box featuring a portrait of Andjar Asmara and a detailed text about his life as a dramatist and filmmaker.
- In the news**: A box listing recent news items such as the discovery of an Earth-sized planet, Hilary Mantel winning the Man Booker Prize, and the death of Norodom Sihanouk.
- On this day...**: A box listing historical events on October 20, such as the birth of the Báb and the first armed test of a submarine-launched ballistic missile.
- Did you know...**: A box listing interesting facts, such as the use of lapis lazuli in ancient artifacts and the first armed test of a submarine-launched ballistic missile.

Each section includes links to 'Archive', 'Email', and 'More' for further reading.

Cada sección puede tener su nivel 1 título (`<h1>`), y el encabezado puede también incluir el título `<h1>`. Cada uno de estos bloques es independiente de los demás, y es habitual encontrar varios títulos `<h1>` en el código de la página web. Así «El título `<h1>` del `<header>`», «El título `<h1>` de esta `<section>`», etc.

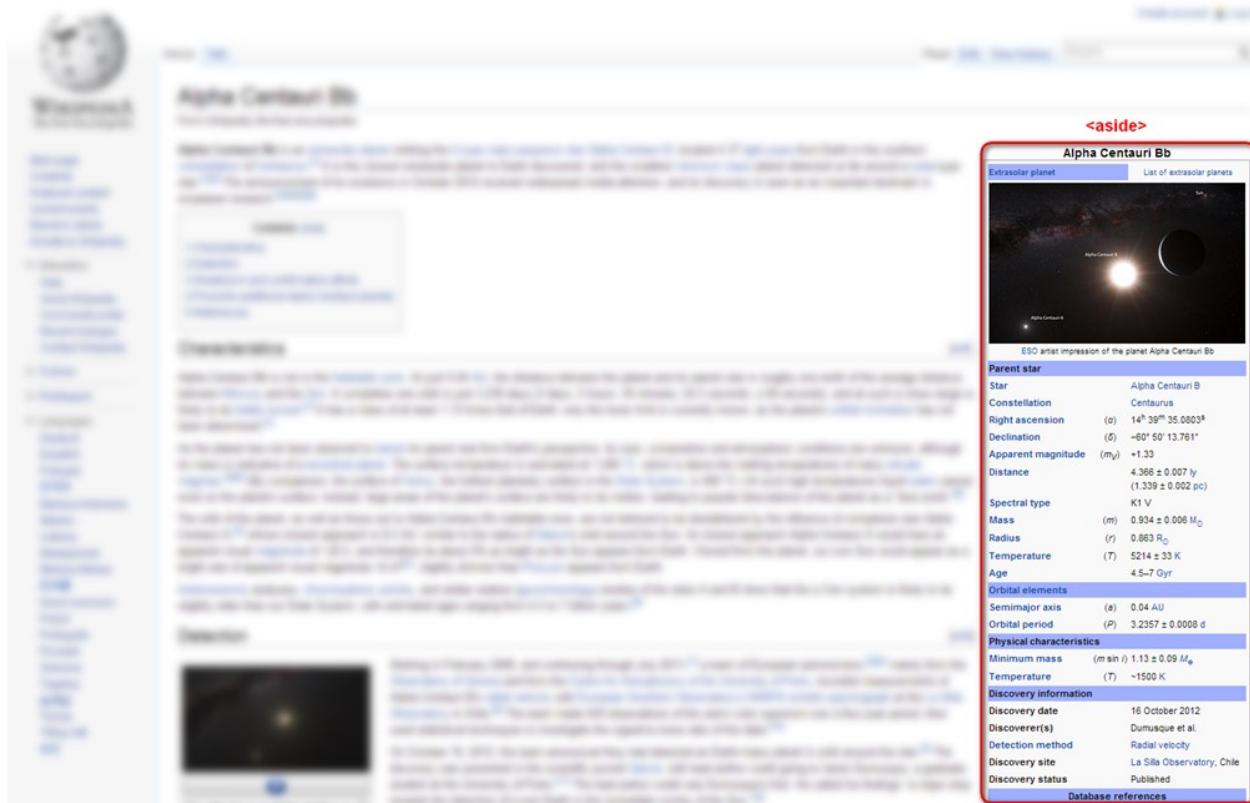
5.5. Aside

El elemento HTML <aside> representa una sección de una página que consiste en contenido que está indirectamente relacionado con el contenido principal del documento. Estas secciones son a menudo representadas como barras laterales o como inserciones y contienen una explicación al margen como una definición de glosario, elementos relacionados indirectamente, como publicidad, la biografía del autor, o en aplicaciones web, la información de perfil o enlaces a blogs relacionados.

```
1 <aside>
2     <!-- Pon aquí más información sobre tu contenido -->
3 </aside>
```

Ejemplo:

En Wikipedia, por ejemplo, el artículo que se está viendo a menudo tiene un bloque de información adicional a la derecha. Así, la página que describe una estrella (siguiente figura) contiene un bloque dando las características de la estrella (dimensiones, peso, etc.).



5.6. Article

El elemento <article> especifica contenido autónomo e independiente.

Un artículo debe tener sentido por sí solo y debe ser posible distribuirlo independientemente del resto del sitio web.

Ejemplos de dónde se puede utilizar un elemento <article>:

- Publicación en el foro
- Entrada en el blog
- Artículo de periódico

```
1 <article>
2   <h1>Mi artículo</h1>
3   <p>Bla bla bla bla</p>
4 </article>
```

Ejemplo:

SCIENCE

<article>

Mars' Shiny Particles: Curiosity Rover Digs Up Unidentified Material, NASA Scientists Say

Posted: 10/18/2012 3:52 pm Updated: 10/18/2012 3:52 pm

Like 1,021 people like this. Sign Up to see what your friends like.



NASA/JPL-Caltech/MSSS

402 51 28 94 1172

GET SCIENCE ALERTS:
Enter email SIGN UP

f share tweet +1 email comment

REACT: Amazing Inspiring Funny Scary Hot Crazy Important Weird

FOLLOW: Video, Planetary Science, Curiosity, Curiosity Shiny Particles, Mars, Mars Rocks, Mars Shiny Particles, Shiny Particles Mars, Space, Science News

By: SPACE.com Staff
Published: 10/18/2012 10:14 AM EDT on SPACE.com

NASA's Mars rover Curiosity has found some more bright stuff on the Red Planet, scientists say.

The Curiosity rover had dumped out a recent sample of Mars dirt after spotting shiny particles inside the hole dug by the shovel-like scoop on its robotic arm. Mission scientists were concerned that the bright material might be debris from the rover itself, so they ordered Curiosity to toss the Mars sample as a safety precaution, mission managers said in a statement.

As it turns out, the bright particles are actual Martian in origin, they added.

Curiosity is scooping up Mars dirt in a spot called "Rocknest" in preparation to deliver the first samples to a pair of instruments on its car-size body. The rover scooped its first Mars sample on Oct. 7, but further work was delayed when the rover's cameras spotted a bright object —likely a piece of plastic —on the ground in the area.

Scientists had cleared Curiosity to start scooping Mars dirt again ahead of the first sample study when they spotted the new bright material. Curiosity's initial Mars dirt scoops are designed to clean the rover's sampling system. [Mars Rover Curiosity's



5.7. Figuras

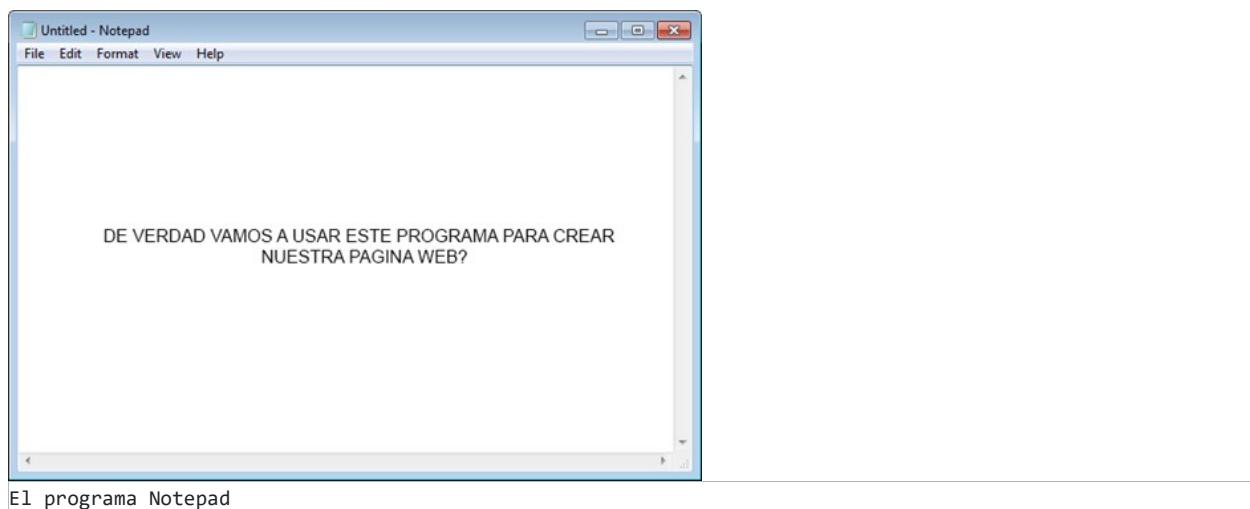
Son elementos que realzan el texto para complementar la información de la página.

Las figuras pueden ser de diferentes tipos:

- imágenes
 - código fuente;
 - comillas;
 - ...

Crear una figura

Tomemos de nuevo el ejemplo de la captura de pantalla que vimos en el primer capítulo, que se muestra en la siguiente ilustración:



En HTML5, el elemento es <figure>. Aquí tenemos la forma de usarlo:

```
1 <figure>
2 
3 </figure>
```

Una figura suele ir acompañada la mayoría de las veces de una leyenda. Para añadir una leyenda, utiliza la etiqueta `<figcaption>` dentro de la etiqueta `<figure>`, así:

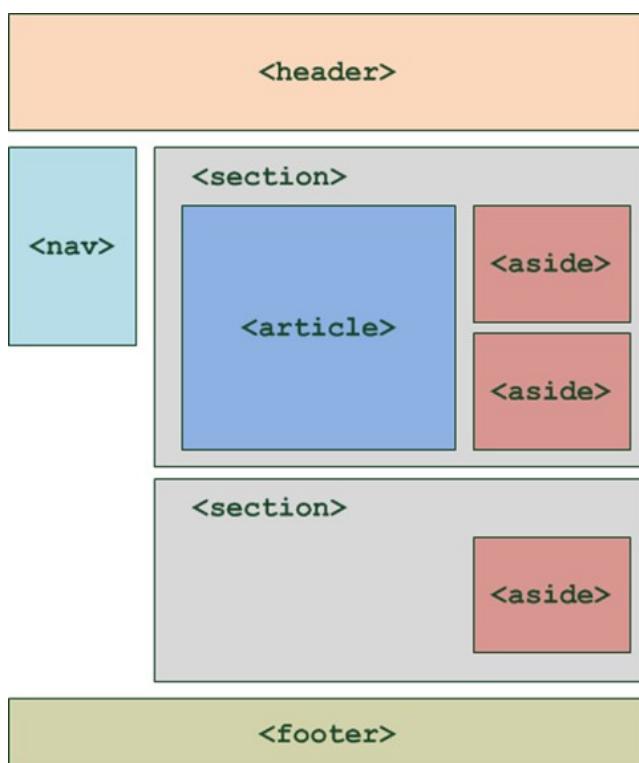
```
1 <figure>
2 
3 <figcaption>El programa Notepad</figcaption>
4 </figure>
```

El elemento `<figure>` tiene un papel principalmente **semántico**. Esto significa que le dice a la computadora que la imagen tiene un significado y que es importante para la comprensión del texto. Esto puede permitir a un programa recuperar todas las figuras del texto para mencionarlas en una tabla de figuras, por ejemplo.

Por último, ten en cuenta que una figura puede tener varias imágenes. Este es un caso que lo justifica:

```
1 <figure>
2 
3 
4 
5 <figcaption>Logos de varios navegadores</figcaption>
6 </figure>
```

5.8. Resumen



No cometas errores: este diagrama te ofrece un ejemplo de organización de página. Tú decides si pones tu menú de navegación en la derecha, o en la parte de arriba, o tus etiquetas `<aside>` arriba, etc.

Podías imaginar un segundo `<header>`, esta vez ubicado dentro de una `<section>`. En este caso, se consideraría la sección encabezada.

Finalmente, una sección no tiene que contener necesariamente una etiqueta de `<article>` y unas etiquetas `<aside>`. Usa solo estas etiquetas si las necesitas. Nada te impide crear secciones que contengan sólo párrafos, por ejemplo.

A continuación, compartimos algunas convenciones HTML5 que debemos tener en cuenta: <https://wwwexasoluciones.com/inicio/tutoriales/html/guia-de-estilos-y-convenciones-de-codificacion-html5>