CBT No.1 Dr. Jorge Jiménez Cantú, Tecámac

Técnico en Informática.

**Profr.** Ramiro Feregrino López Estudiante**:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Módulo Profesional III: Desarrolla aplicaciones web y móviles**  **Grado**: Segundo de Informática 5. **Núm. de Lista**: \_\_

**Submódulo I**: Desarrolla aplicaciones Web. **Calificación:** \_\_\_\_\_\_\_\_\_\_\_\_\_

ETIQUETAS DE LENGUAJE HTML

# 1. Introducción a HTML

## Introducción y sintaxis básica

Los archivos HTML son los que conforman las páginas del sitio web y tienen una extensión .html o .htm. Es indiferente utilizar una u otra extensión pero se recomienda utilizar la misma para todo el sitio por motivos de consistencia.

Como hemos visto, HTML es un lenguaje de marcado que nos sirve para describir la estructura y el contenido del documento.

Las etiquetas HTML están definidas en las especificaciones del W3C (El World Wide Web Consortium, es un consorcio internacional que produce recomendaciones para la World Wide Web) y a cada una de ellas se le asocia un significado.

## ¿QUÉ ES UNA ETIQUETA HTML?

Antes de entrar en materia, conviene explicar de que vamos a hablar. Las etiquetas HTML son pequeños bloques de código, que indican al navegador como debe interpretar el contenido recogido entre dichas etiquetas. Por ejemplo, si queremos «pintar» un párrafo de texto, hay una etiqueta específica para que el navegador interprete ese texto como un párrafo. Estas etiquetas cuentan además con **atributos**que podemos añadir y que, en sí, pueden determinar cómo será el comportamiento específico que tendrá la etiqueta. Te dejo por aquí un [listado de atributos HTML](https://carontestudio.com/blog/atributos-html/) para complementar esta entrada.

Las etiquetas HTML comienzan siempre con el símbolo «<» y finalizan con el símbolo «>». Entre medias de estos dos símbolos irá el nombre de la etiqueta que queremos que el navegador interprete.

Todas las etiquetas HTML están compuestas por una **etiqueta de apertura**, y una **etiqueta de cierre**. Aunque hay excepciones, como por ejemplo la etiqueta para cargar una imagen que son llamadas etiquetas huérfanas, ya que solo tienen etiqueta de apertura.

Las etiquetas de cierre se construyen con una barra «/» justo antes de la propia etiqueta y los símbolos «<» y «>»

## Reglas básicas

* Las etiquetas están encerradas entre los signos "**<**" y "**>**".
* Generalmente vienen en pares, por ejemplo **<p>** y **<⁄p>**.
* La primera es de **apertura** y la segunda de **cierre**.
* El texto que se encuentra entre dos etiquetas es el **contenido** del elemento.
* Las etiquetas no son sensibles a las mayúsculas y minúsculas, o sea **<b>** es lo mismo que **<B>**.

Por ejemplo: <h1>El título de la página</h1>

En el caso del ejemplo hemos utilizado la etiqueta de título de primer nivel (h1) para marcar que el texto que contiene corresponde al titular de la página.

### ETIQUETAS INICIALES O DE RAÍZ

* **<!DOCTYPE html>** Indica al navegador que el documento está basado en el estándar HTML5
* **<html> </html>** Representa la raíz de un documento HTML. Todos los demás elementos de la estructura HTML deben ser recogidos dentro de estas etiquetas.

### METADATOS DEL DOCUMENTO

* **<head> </head>** Representa una colección de metadatos acerca del documento, incluyendo enlaces a, o definiciones de, scripts y hojas de estilo. El resto de etiquetas de metadatos, **irán recogidas dentro de las etiquetas de apertura y cierre del head**. Importante explicar que estos metadatos del documento, es información para el navegador y no contenido que será visible en la página web. A excepción de la etiqueta <title> que veremos a continuación.
* **<title> </title>** Etiqueta usada para definir el título de la página web.
* **<link>** Se usa para enlazar recursos externos al documento HTML. El ejemplo más común son **las hojas de estilos CSS.**
* **<meta>** Etiqueta usada para definir otros metadatos que no se pueden definir con una etiqueta HTML especifica. Por ejemplo, para definir el autor del sitio, o la descripción del mismo.
* **<style> </style>** Etiquetas usadas para introducir código CSS en línea, es decir, en el propio documento HTML.

**La sección <head>**

Los elementos que se encuentran en la sección **head** incluyen el título de la página web (**title**), las metaetiquetas (**meta tags**) que describen el documento (como la codificación de caracteres utilizada y la información a la que pueden acceder los motores de búsqueda) y las referencias a scripts y estilos (style, script, base y link). Muchas de estas características no se muestran directamente en la página web.

El elemento head comienza con la etiqueta <head> y termina con la etiqueta </head>. Siempre codificarás al menos otros dos elementos en la sección head: un elemento title y un elemento meta.

* El elemento **title** configura el texto que aparecerá en la sección barra de título de la ventana del navegador. El texto entre las etiquetas <title> y </title> se llama título de la página web.
* El elemento **meta** describe una característica de una página web, como la codificación de caracteres. La codificación de caracteres es la representación interna de letras, números y símbolos en un archivo como una página web u otro archivo que se almacena en una computadora y puede transmitirse a través de Internet.

Hay muchos conjuntos de codificación de caracteres diferentes. Sin embargo, es una práctica común utilizar un conjunto de codificación de caracteres ampliamente soportado, como **utf-8**, que es una forma de Unicode (http://www.unicode.org). La etiqueta meta no se utiliza como un par de etiquetas de apertura y cierre.

Se considera una etiqueta independiente o autocontenida (denominada elemento *void* en HTML5). La etiqueta meta utiliza el atributo **charset** para indicar la codificación de caracteres.

En cualquier documento HTML encontraremos etiquetas anidadas, es decir, etiquetas que marcan elementos que se encuentran dentro de otras etiquetas. En este caso es muy importante mantener la simetría de las etiquetas.

Forma correcta:

<p>Un texto que contiene una parte a la que interesa <strong>dar más fuerza </strong></p>

Forma incorrecta:

<p>Un texto que contiene una parte a la que interesa <strong>dar más fuerza </p></strong>

En este caso, dentro de un párrafo (etiqueta <p>) hay unas palabras a las que se les quiere dar más fuerza (etiqueta <strong>).

Las etiquetas pueden tener atributos, como por ejemplo las imágenes, donde se indica la ruta del archivo:

<img src="img/miimagen.jpg"  />

La estructura para marcar los atributos siempre es la misma: atributo= “valor”

## Estructura de un documento HTML

El primer elemento que encontramos en un archivo HTML es la Definición del Tipo de Documento (DTD), donde se especifica qué versión de la especificación del W3C estamos utilizando y el vínculo a ésta.

<!DOCTYPE html">

<HTML>

<HEAD>

<TITLE>Ejemplo 1</TITLE>

</HEAD>   
 <BODY>   
 Hola mundo   
 </BODY>   
</HTML>

<HTML>   
<HEAD>   
<TITLE>Ejemplo 2</TITLE>   
</HEAD>   
<BODY>   
   <H1>Mi primera página</H1>   
      
   <!-- Aquí va un comentario que no es   
      interpretado por el navegador -->   
      
<P>Hola mundo, esta es un página con titular,   
que tiene también un párrafo y unos cuantos   
saltos de línea.</P>   
  
Uno<br>   
Dos<br>   
Tres<br>   
</BODY>   
</HTML>

De este modo, estamos indicando la referencia sobre cómo descifrar el código HTML, ya que variará en función de la especificación que estemos utilizando.

Un documento HTML está marcado por la etiqueta <html> y tiene dos partes principales:

<head>: Es el encabezado del documento y en él encontramos información relacionada con la página. Son elementos que no visualizamos directamente en el navegador:

<title> : título de la página

<meta>: información adicional sobre la página, como descripción, palabras clave, autores y distintas informaciones para motores de búsqueda.

También se pueden incluir en la cabecera los scripts y códigos CSS o vínculos a ellos.

<body>: Es donde encontramos los elementos que sí vamos a visualizar, es decir,  el contenido y su estructura: texto, imágenes, enlaces…

La estructura básica es la siguiente:

<!-- Aquí comienza la página html -->

<html>

<head>

<title>Aquí se pone el título de la página HTML</title>

</head>

<body>

Aquí va el contenido de la página.

</body>

<!-- Aquí termina la página html -->

</html>

**Párrafos y saltos de línea**

Son dos de las etiquetas más utilizadas, ya que nos permiten formatear el texto.

La etiqueta de párrafo es **<p>**:

<p>El texto de un párrafo</p>

La etiqueta de salto de línea es un poco especial ya que se abre y cierra en sí misma:

Salto de línea simple <br />

Otra línea de texto

Al ser visualizados en un navegador, la diferencia principal entre un párrafo y un salto de línea es que el párrafo tiene un margen inferior que lo separa claramente del elemento siguiente mientras que el salto de línea no contempla esta separación. Como veremos más adelante, los márgenes se pueden modificar con la hoja de estilos.

En Dreamweaver sólo será necesario pulsar la tecla **Enter** para que automáticamente inserte las marcas de párrafo necesarias. Para conseguir un salto de línea debernos pulsar **Mayúsculas + Enter**

**Encabezados**

Los encabezados sirven para marcar los títulos de los distintos apartados de la página. Podemos establecer una jerarquía de hasta 6 niveles de titular por importancia.

El encabezado de nivel 1 será el más importante de la página y el 6 el de menos relevancia.

Utilizaremos las etiquetas:

<h1>Encabezado de mayor importancia</h1>

<h2>Encabezado de segundo nivel</h2>

<h3>Encabezado de tercer nivel</h3>

<h4>Encabezado de cuarto nivel</h4>

<h5>Encabezado de quinto nivel</h5>

<h6>Encabezado de menor importancia</h6>

Marcar los titulares no sólo sirve para mostrarlos de manera distinta al visualizar la página sino que estamos indicando a los agentes de la red, como buscadores, la importancia de estos elementos y su jerarquía, facilitando la indexación eficiente de la información.

En Dreamweaver podemos aplicar de manera sencilla los encabezados: seleccionamos el texto y, desde la ventana propiedades, le asignamos el tipo de encabezado dentro del desplegable de formato.

**Énfasis**

Podemos remarcar ciertas palabras o frases a través de las etiquetas de énfasis. Podemos utilizar dos niveles de énfasis:

A través de la etiqueta **<em>** daremos un énfasis normal. Con la etiqueta **<strong>** daremos un énfasis acentuado.

A la práctica, estas etiquetas se utilizan para mostrar en cursiva **(em)** o negrita **(strong)** el texto, pero tenemos que recordar que ésta es la visualización por defecto que muestran los navegadores. Podemos modificar a través de la hoja de estilos cómo mostrar los elementos a los que hemos dado énfasis.

Ejemplo:

<em> un elemento al que damos énfasis </em>

<strong> un elemento al que damos mucho énfasis </strong>

En Dreamweaver utilizaremos los iconos habituales de negrita y cursiva para aplicar estas etiquetas al texto seleccionado.

**Enlaces**

Los enlaces son la base de la World Wide Web, ya que son los que permiten conectar páginas entre sí y, por lo tanto, hacen posible la navegación.

La etiqueta de enlace **<a>** está compuesta por varios atributos, siendo esencial la referencia:

<a href = ”otrapagina.html”>Enlace a otra página</a>

La referencia puede ser a:

* Una página del mismo sitio, utilizando una ruta relativa respecto al documento que enlaza, como es el caso anterior donde se hace referencia a un archivo que se encuentra en el mismo directorio.
* Una página de otro sitio usando su URL, por ejemplo:

<a href=”http://www.mipagina.com”>Enlace a otro sitio web</a>

* Otro punto de la misma página. En este caso es necesario poner además un ancla en el destino del enlace con un atributo de nombre para poder realizar el enlace, o bien referirse a un identificador ya existente:  
  El ancla:

<a name="seccion"></a>

El enlace:

<a href=”#seccion”>Enlace a una sección de la misma página</a>

* Un archivo, de modo que se podrá descargar o ejecutar el programa idóneo para abrirlo.  
  Por ejemplo:

<a href=”uninforme.pdf”>Enlace un archivo PDF</a>

Otro atributo habitual es el destino (target), con el que podemos definir en qué ventana se va a mostrar la página enlazada.

<a href=”unapagina.html” target=”\_blank”>Enlace un archivo PDF</a>

Los destinos son:

* **\_self** : se abre en la propia ventana, es el comportamiento por defecto.
* **\_blank** : abre una nueva ventana del navegador.

Otros en el caso de páginas construidas por marcos (frames, páginas compuestas por distintos documentos HTML, pero que no se recomienda su uso): **\_parent** **\_top** o indicando el nombre del marco de destino.

En Dreamweaver crearemos los enlaces desde la ventana **Propiedades**, donde podremos buscar el archivo a enlazar entre los ya existentes en el sitio o escribiendo directamente la URL de destino.

**Imágenes**

La forma de incluir imágenes en una página guarda cierta relación con los enlaces, ya que también debemos incluir un atributo con la ruta de la imagen:

<img src=”img/miimagen.jpg” />

Vemos, además, que es una etiqueta que se cierra en sí misma: debemos añadir la barra inclinada de cierre al final.

Entre los atributos de la imagen encontramos:

* **alt**: un texto alternativo para describir la imagen en caso de no poder ser visualizada. Es importante añadir esta información por numerosos motivos: accesibilidad, buscadores, problemas en la descarga de la imagen…
* **width**: indica la anchura de la imagen.
* **height**: indica la altura de la imagen.

<img src=”img/miimagen.jpg” alt=”Vista aérea de la población” width=”400” height=”300” />

En Dreamweaver insertaremos las imágenes desde la ventana **Insertar > común**. Al seleccionar el icono de imagen se nos abrirá una ventana emergente para buscar el archivo. Los atributos para facilitar la accesibilidad también los indicaremos en el mismo proceso.

**Listas**

Las listas enumeran un conjunto de elementos.

Una lista, por ejemplo **<ul>**, está formada por un conjunto de items de lista **<li>**.

El navegador nos mostrará los elementos precedidos de una viñeta circular. A través de la hoja estilos podremos personalizar la viñeta.

Los menús de navegación son listas de enlaces, y  los debemos marcar como tales.

Las listas pueden ser de tres tipos:

**Listas ordenadas:** implica que el orden de los elementos es necesario. El navegador nos mostrará los elementos precedidos del número de orden. A través de la hoja estilos podremos cambiar el tipo de numeración.

<ol>

<li>Item 1</li>

<li>Item 2</li>

<li>Item 3</li>

</ol>

**Listas desordenadas:** enumera elementos que no necesariamente han de estar en el orden que aparecen.

La única diferencia en el código respecto las listas ordenadas está en la etiqueta que define la lista.

<ul>

<li>Item 1</li>

<li>Item 2</li>

<li>Item 3</li>

</ul>

**Listas de definición:** contienen un listado de definiciones, a modo de glosario. Las etiquetas en este caso difieren de las listas vistas anteriormente, ya que encontramos la definición de la lista **<dl>**, el término **<dt>** y la descripción **<dd>**:

<dl>

<dt>Palabra 1</dt>

<dd>Definición de la palabra 1</dd>

<dt>Palabra 2</dt>

<dd>Definición de la palabra 2</dd>

</dl>

En Dreamweaver marcaremos un elemento como lista desde la ventana **propiedades** a través de los iconos de lista, de manera similar a un procesador de textos.

### ETIQUETAS DE SECCIONES O PARA ESTRUCTURAR EL HTML

* **<body> </body>** Al contrario que la etiqueta de metadatos <head>, todo lo que quieras mostrar en la página web debe ir recogido dentro de las etiquetas de apertura y cierre de <body>. **Este contenido será el que se muestre en la web.**
* **<nav> </nav>** Usadas para definir el contenido que será la sección de navegación de la web.
* **<main> </main>** Se usa para definir el contenido principal del documento. Solamente puede existir uno por documento.
* **<section> </section>** Define una sección del documento
* **<article> </article>** Define contenido independiente de la web.
* **<aside> </aside>** Dentro de estas etiquetas suele alojarse el contenido adicional de la web. Suele ser contenido relacionado con la web pero de poca importancia
* **<h1>,<h2>,<h3>,<h4>,<h5>,<h6>** Son etiquetas HTML muy importantes, ya que son usadas para jerarquizar el contenido de la web. Las etiquetas se usan para explicar brevemente el contenido que irá a continuación.
* **<header> </header>** Se usan para definir la cabecera la página web. Suele contener el logotipo, menú de navegación, etc.
* **<footer> </footer>** Usadas para definir el pie de página.

**Tablas**

A pesar de que las tablas han sido usadas mayoritariamente para maquetar las páginas, este uso está totalmente desaconsejado hoy en día. Por lo tanto, la función de las tablas estará reservada exclusivamente a mostrar información tabulada.

Al igual que las listas, las tablas están constituídas por distintos elementos, por lo que tendremos que marcar la tabla **<table>**, que está formada por filas **<tr>** y a su vez, las filas formadas por celdas **<td>**. Su estructura es:

<table>

<tr>

<td>Contenido de la celda</td>

<td>Contenido de la celda</td>

<td>Contenido de la celda</td>

</tr>

<tr>

<td>Contenido de la celda</td>

<td>Contenido de la celda</td>

<td>Contenido de la celda</td>

</tr>

</table>

En Dreamweaver insertaremos las tablas desde el menú **Insertar > común**, indicando las filas y columnas necesarias.

### ETIQUETAS PARA LA AGRUPACIÓN DE CONTENIDO

* **<p> </p>** Etiqueta usada para escribir párrafos de texto.
* **<hr>** Etiqueta utilizada para «romper» entre dos secciones de una web. Usada comúnmente como separador.
* **<pre> </pre>** Usada para pegar texto manteniendo el pre formato propio del texto.
* **<blockquote> </blockquote>** Se usan para indicar que el contenido es texto citado.
* **<ol> </ol>** Etiquetas para crear una lista ordenada
* **<ul> </ul>** Etiquetas para crear una lista des-ordenada
* **<li> </li>** Etiquetas que recogen el contenido de un elemento de una lista, sea ordenada o no.
* **<dl> </dl>** Usada para crear una lista de definiciones.
* **<dt> </dt>** Representa un término definido por la siguiente etiqueta <dd>
* **<dd> </dd>** Se usa para definir los términos listados antes que él.
* **<figure> </figure>** Indica una figura ilustrada como parte del documento HTML5.
* **<figcaption> </figcaption>** Utilizada para definir la leyenda de una figura.
* **<div> </div>** Etiqueta común utilizada para crear un contenedor genérico

# 3. Otros elementos de HTML

### <body> </body>

Indica la parte del cuerpo del contenido de un documento HTML. Es una etiqueta esencial para cualquier documento ya que indica donde empieza el contenido visible del documento.

### <head></head>

La parte superior del documento HTML, es donde podremos indicar los metadatos: título del documento, hojas de estilos, [javaScript](https://www.iebschool.com/blog/como-mejorar-el-seo-del-contenido-javascript-de-una-web-seo-sem/" \t "_blank), CSS…

### <div> </div>

Un elemento que es usado mayoritariamente para agrupar otros elementos y actuar como plantilla de otros controles. La etiqueta <div> nos ayuda a estructurar el documento en secciones.

### <a> </a>

Es una etiqueta que nos ayuda a poder crear un enlace a una página web. El atributo principal de la etiqueta HTML es href, donde pondremos el enlace al que queremos conectar. Otro atributo muy usado es target, el cual nos sirve para indicar si el enlace se abrirá en una nueva ventana o en la misma.

Ejemplo HTML:

Pulsa <a href=”https://www.nombredelaweb.com/” target=”\_blank”>aquí</a> para visitar DonDominio.

### <strong> </strong>

Si tienes mucho texto, es importante poder dar énfasis a una parte en concreto, con la etiqueta strong lo podemos hacer.

Ejemplo HTML:

Quiero destacar solo <strong>esta palabra</strong>.

### <br>

Con esta etiqueta HTML le podemos decir al navegador que viene un salto de línea. Nos sirve para hacer el texto más leíble.

### <H1> </H1> …. <H6> </H6>

Hay diferentes niveles de títulos, del 1 al 6. Las etiquetas <H + número> nos permiten indicar la importancia del título y para estructurar el contenido, de esta forma ayudamos a los bots a entender la importancia del contenido.

### <IMG> </IMG>

Usamos la etiqueta IMG para mostrar imágenes dentro del contenido. Necesita el atributo src para funcionar, ya que será donde indicaremos desde donde tiene que mostrar la imagen.

### <OL> <li></li> <OL> | <UL> <li></li> <UL>

Las etiquetas OL y LI nos sirven para crear listas, OL para listas ordenadas y UL para listas sin orden. Dentro de las listas, los elementos se identifican con la etiqueta LI.

Ejemplo HTML:

<ul>

<li>Primer elemento</li>

<li>Segundo elemento</li>

<ul>

### <P> </P>

Etiqueta que nos sirve para agrupar texto dentro de un parágrafo. El propósito es poder hacer el contenido más fácil de leer y organizado.

### <SPAN> </SPAN>

Con la etiqueta podemos personalizar el estilo de solamente una parte del texto.

Ejemplo HTML:

Solo <span style=”color: red;”>esta palabra</span> en rojo.

## Ventajas y desventajas del HTML

Como todo lenguaje informático, el HTML tiene sus ventajas y sus desventajas. Entre las ventajas, podríamos destacar que es apto para principiantes, que tiene una curva de aprendizaje poco profunda y que es accesible. Además, es de **código abierto y completamente gratuito** y se ejecuta de forma nativa en todos los navegadores web.

Por otro lado, entre las desventajas, se encuentra que para una funcionalidad dinámica es posible que haya que utilizar JavaScript o un lenguaje back-end como PHP. Además, los usuarios deben crear [páginas web](https://www.iebschool.com/blog/crear-sitio-web-marketing-digital/) individuales para HTML, incluso si los elementos son los mismos y es posible que los navegadores más antiguos no muestren las etiquetas más nuevas.

## Elementos de bloque y elementos en línea

Todos los elementos HTML los podemos clasificar en dos grande grupos:

* **Elementos de bloque (block):** Los elementos de bloque son aquellos que ocupan todo el ancho que tienen a su disposición. Así pues, ocuparán todo el ancho de la página dejando un salto de línea antes y después. Si se encuentran dentro de un elemento con un ancho definido ocuparán todo el ancho de este elemento. Son elementos de bloque los párrafos, encabezados, tablas, formularios, listas y items de lista.
* **Elementos en línea (inline):** Son elementos en línea los enlaces, las imágenes y los elementos que componen los formularios (input, textarea, select…)

## Etiquetas div y span

Hemos dejado para el final las etiquetas **<div>** y **<span>** porque tiene características especiales, ya que no aportan un significado al elemento que contienen y su función es muy distinta a la de las etiquetas vistas anteriormente. Su misión principal es dividir en partes el documento HTML, ya sea creando bloques que agrupen varios elementos con el objetivo de aplicarles estilos como conjunto, o añadiendo una subdivisión dentro de una etiqueta para poder referirnos a una parte de ésta:

### Etiqueta <div>

Es un elemento de bloque, lo que significa que ocupará todo el ancho disponible. Nos podemos referir a ella como división, contenedor e, incluso, capa. Su uso mayoritario está orientado a la maquetación de la página, ya que al agrupar elementos podemos asignarle a través de la hoja de estilos una posición, medidas, color de fondo, bordes etc., que englobará a todas las etiquetas dentro del bloque. Podemos entender cada división como una caja que contiene varios elementos HTML y que la podemos mover, colocar delante o detrás de otras cajas y asignarle unas características como conjunto.

### Etiqueta <span>

Es un elemento en línea, y por lo tanto sólo ocupará el ancho del elemento que contenga. Se utiliza sobre todo para añadir estilos a parte del contenido de una etiqueta. Por ejemplo, si queremos resaltar unas palabras dentro de un párrafo necesitaremos crear esta subdivisión sin afectar al resto del párrafo.

Ya en el momento de diseñar la estructura de la página es importante tener en cuenta qué bloques serán necesarios, y mantener los mismos bloques principales entre las distintas páginas de un sitio web para simplificar la fase de desarrollo.

En Dreamweaver podemos crear las etiquetas div desde la ventana **Propiedades** aunque en ciertas ocasiones será aconsejable insertarlas directamente desde la vista de código, ya que precisaremos con exactitud el abasto de éstas.

## Atributos id y class

Los atributos id y class nos permiten identificar etiquetas HTML a través de un nombre. Es decir, podemos otorgar un nombre libremente a una etiqueta específica para podernos referir a ella, como por ejemplo aplicarle un estilo específico sin que éste afecte l resto de etiquetas del mismo tipo.

Podemos añadir estos atributos a cualquier etiqueta HTML que necesitemos singularizar.

La función de ambas etiquetas, **id** y **class**, es la misma y sólo se diferencian por la cantidad de veces que las podemos usar en cada página:

* **id**: es un identificador único, por lo que sólo lo podremos utilizar una vez por página
* **class**: son clases que podemos utilizar tantas veces como sea necesario en una misma página

<ul id=”menu”>

<li class=”tipo1”>Elemento de tipo 1</li>

<li class=”tipo2”>Elemento de tipo 2</li>

<li class=”tipo1”>Elemento de tipo 1</li>

<li class=”tipo2”>Elemento de tipo 2</li>

</ul>

En este ejemplo tenemos una lista a la que hemos dado un nombre único a través de un identificador. Esta lista contiene elementos de dos tipos: hemos utilizado el atributo class porque hay más de un elemento de este tipo.

De este modo podemos tener varias listas desordenadas en la misma página y dar un estilo especial a la lista que corresponde al menú. El resto de listas no se verán afectadas por esta modificación.

A un mismo elemento le podemos otorgar tantas clases como sea necesario. Para indicarlo deberemos separar los nombres por un espacio en blanco:

<li class=”tipo1 destacado primero”>

## Insertar comentarios en el código

A menudo necesitaremos insertar comentarios entre el código, ya sea para hacer algunas aclaraciones o bien para facilitar el trabajo de otras personas.

Los comentarios no tienen una etiqueta específica, sino que se marcan de la siguiente manera:

Para empezar el comentario: **<!–**  
Para cerrarlo: **–>**

Ejemplo:

<!-- Esto es un comentario -->

# 5. Introducción a CSS

CSS es el acrónimo de **Cascading Style Sheets**, la traducción del cual es **Hojas de Estilo en Cascada**.

Como hemos visto, la hoja de estilos es la responsable de la apariencia de cada página: en ella se indica cómo deben visualizarse los distintos elementos de un documento HTML.

Al ser independiente del contenido, resulta muy fácil cambiar completamente la apariencia de un sitio web: sólo será necesario modificar el archivo CSS para que todos los documentos HTML adopten las nuevas reglas de presentación.

## Aplicación de estilos a documentos HTML

Hay tres maneras de aplicar estilos a los elementos de un documento HTML:

### Hoja de estilos externa

Es el caso más habitual y recomendado. Se trata de un archivo independiente con extensión **.css**

En el documento HTML encontramos el vínculo a este archivo a través de una instrucción en el head:

<link rel="stylesheet" type="text/css" href="css/miestilo.css" />

Es el sistema más limpio para aplicar la hoja de estilos, ya que toda la información se encuentra en un archivo externo y puede ser modificada en cualquier momento sin necesidad de editar el documento HTML.

Además, todas las páginas de un sitio pueden estar vinculadas al mismo archivo CSS, de modo que sólo es necesario cambiar este archivo único para que todo el sitio cambie de apariencia.

Es posible, además, vincular más de un archivo CSS a cada página.

Esto tiene sentido en sitios de gran tamaño en que, por ejemplo, se pueden definir unos estilos generales para todo el sitio en una hoja y en otra estilos particulares para los distintos apartados.

En Dreamweaver vincularemos los archivos CSS desde el archivo HTML: en la ventana **propiedades**, en el apartado **CSS / Clase** encontraremos la opción de vincular la hoja de estilos. También los podemos vincular **desde la ventana CSS**, utilizando el **icono de enlace**.

### CSS interno

La información de los estilos está incrustada dentro del head del documento HTML.

Sólo utilizaremos esta opción en casos puntuales, ya que en este caso sí debemos modificar el archivo HTML para acceder a los estilos. En un sitio tendríamos que abrir cada archivo HTML para poder modificar los estilos.

<style type="text/css">

body { font-family: Georgia, "Times New Roman", serif; background-color: #ffffff;  }

</style>

Podría justificarse utilizar un estilo interno en una página especial con un diseño completamente distinto al del resto del sitio.

### CSS en línea (inline)

Es el caso menos aconsejado para incrustar estilos CSS y sólo debe ser utilizado ocasionalmente, ya que éstos se colocan como atributos de la etiqueta HTML del siguiente modo:

<p style=”color: #333333”>Párrafo con el texto en gris</p>

La ventaja es que garantizará que se aplicará este estilo al elemento, ya que tiene prioridad sobre cualquier otra especificación que venga de la hoja de estilo interna o externa.

El inconveniente es que se pierde la noción de separación de presentación y contenido y, por lo tanto, sus numerosas ventajas.

## Reglas CSS y sintaxis básica

Las reglas CSS están formadas por un selector, y una declaración formada por propiedades y valores del siguiente modo:

selector { propiedad: valor; }

* El selector es el elemento al que queremos aplicar el estilo, y tiene relación con las etiquetas HTML y los identificadores y clases utilizados e los archivos HTML del sitio.
* Las propiedades son los distintos atributos que podemos modificar de las páginas, a los que otorgamos un valor.
* Los atributos los escribimos entre llaves y, en el caso de modificar más de un atributo en una regla CSS al final de cada uno es necesario colocar un punto y coma para separarlos.

Por ejemplo:

h1 {  color: #333333; font-family: verdana, arial, sans-serif; }

En este caso estamos definiendo todos los encabezados de primer nivel, asignándoles una tipografía verdana de color gris.

Encontramos varias maneras de referirnos a los elementos de un documento HTML: desde redefinir el estilo de todas las etiquetas de un tipo a aplicar un estilo a una parte de un elemento en particular.

**Para quitar el espacio después del encabezado, se selecciona este, se redefine H1, H2, etc. y se va a la cuarta opción: cuadro, en la sección margin desactivar igual para todos. Después en Bottom dar –20 y listo**.

**Para poder agregar espacios al texto.**

**Te vas a vista de código, agregas la etiqueta <pre> el texto a espaciar </pre>**

**¡Nos ponemos en marcha!**

**¿Qué se necesita?**

Lo más probable es que ya tengas todo lo que necesitas.

Ya dispones de un "navegador". El navegador es el programa que te permite navegar y abrir sitios web. En este preciso momento estás viendo esta página en el navegador.

No importa qué navegador uses. El más habitual es Microsoft Internet Explorer, pero existen otros como Opera y Mozilla Firefox, y todos ellos se pueden usar indistintamente.

**¿Es necesario estar conectado a Internet?**

No es necesario conectarse a Internet, ni cuando estés leyendo este tutorial ni al crear tus sitios web.

Si quieres evitar tener que conectarte mientras lees este tutorial, o bien puedes imprimirlo o desconectarte de Internet mientras lo lees en la pantalla; para crear el sitio web puedes hacerlo en el disco duro de tu ordenador y subirlo después a Internet una vez terminado.

**¿Qué es HTML?**

**HTML es la "lengua materna" de tu navegador.**

Para abreviar una larga historia, podemos decir que un científico llamado Tim Berners-Lee inventó HTML allá por 1990. El objetivo era facilitar a científicos de diferentes universidades el acceso a los documentos de investigación de cada uno de ellos. El proyecto obtuvo un éxito mucho mayor del que Tim Berners-Lee nunca hubiese llegado a imaginar. Al inventar HTML, este científico sentó las bases de la web tal y como la conocemos hoy día.

HTML es un lenguaje que hace posible presentar información (por ejemplo, investigaciones científicas) en Internet. Lo que ves al visualizar una página en Internet es la interpretación que hace el navegador del código HTML. Para ver el código HTML de una página sólo tienes que pinchar en la opción "Ver" de la barra de menús y elegir "Código fuente" (en Internet Explorer).

Para el ojo inexperto, el código HTML parece complicado, pero este tutorial te ayudará a que le encuentres todo su sentido.

**¿Para qué puedo usar HTML?**

Si quieres crear sitios web, no hay otra solución que aprender HTML. Incluso si usas un programa como Dreamweaver, por ejemplo, para la creación de sitios web, poseer unos conocimientos básicos de HTML hacen la vida mucho más fácil y tus sitios web mucho mejores. La buena noticia es que HTML es fácil de aprender y de usar. En las dos próximas lecciones ya habrás aprendido cómo crear tu primer sitio web.

**HTML se usa para crear sitios web. ¡Así de sencillo!**

**Muy bien, pero... ¿qué representan las siglas H-T-M-L?**

HTML es la abreviatura de "HyperText Mark-up Language", es decir, "Lenguaje de marcado hipertextual", y esto ya es más de lo que necesitas saber a estas alturas. En todo caso, para mantener un buen orden, vamos a explicar qué significa con mayor detalle.

* **Hiper** es lo contrario de lineal. En los buenos viejos tiempos -cuando un ratón era un animalillo que perseguía un gato- los programas de ordenador se ejecutaban de forma lineal: cuando el programa había ejecutado una acción seguía hasta la siguiente línea, y después de ésta a la siguiente, y a la siguiente, ... HTML, sin embargo, es diferente: se puede ir donde uno quiera cuando uno quiera. Por ejemplo, no es necesario visitar MSN.com antes de visitar HTML.net.
* **Texto** se explica por sí solo.
* **Marcado** es lo que haces con el texto. Se marca el texto del mismo modo que en un programa de edición de textos con encabezados, viñetas, negrita, etc.
* **Lenguaje** es lo que es HTML. Este lenguaje hace uso de muchos términos en inglés.

Por medio de este tutorial aprenderás lo que se ha dado en llamar XHTML (Extensible HyperText Mark-up Language, es decir, Lenguaje de marcado hipertextual extensible) que, en pocas palabras, es una forma de escribir HTML nueva y mejor estructurada.

Ahora que ya sabes qué representan las siglas HTML (y XHTML), nos ponemos en marcha y empezamos a crear sitios web.

**"¿Etiquetas?"**

Las etiquetas son marcas que se usan para señalar el inicio y el fin de un elemento.

Todas las etiquetas comparten el mismo formato: empiezan con el signo menor que "<" y terminan con el signo mayor que ">".

Por lo general, hay dos tipos de etiquetas: la etiquetas de inicio, por ejemplo, <html>, y las etiquetas de cierre, por ejemplo, </html>. La única diferencia entre la etiqueta de inicio y la de cierre es la barra oblicua "/". El contenido queda etiquetado al colocarlo entre una etiqueta de inicio y una etiqueta de cierre.

HTML trata esencialmente de elementos. Aprender HTML consiste en aprender a usar diferentes etiquetas.

**¿Y si vemos algunos ejemplos?**

Vale. El elemento em se usa para dar énfasis al texto. Todo el texto entre la etiqueta de inicio <em> y la etiqueta de cierre </em> aparecerá enfatizado en el navegador. ("em" es la abreviatura de "emphasis").

**Ejemplo 1:**

<em>Texto con énfasis.</b>

que se verá así en el navegador:

*Texto con énfasis.*

Los elementos h1, h2, h3, h4, h5 and h6 se utilizan para crear encabezados ("h" es la abreviatura de "heading"), donde h1 representa el primer nivel y, normalmente, el texto más grande, h2 es el segundo nivel y, normalmente, es un texto algo más pequeño, y h6 es el sexto y último nivel en la jerarquía de los encabezados y es, normalmente, el texto más pequeño.

**Ejemplo 2:**

<h1>Este es un encabezado de primer nivel</h1>

<h2>Este es un encabezado de segundo nivel</h2>

que se verá así en el navegador:

**Este es un encabezado de primer nivel**

**Este es un encabezado de segundo nivel**

**Así pues, ¿siempre son necesarias las etiquetas de inicio y de cierre?**

Como se dice, siempre hay una excepción que confirma la regla, y en HTML la excepción son unos cuantos elementos que se abren y cierran con la misma etiqueta. Estos elementos que se denominan elementos vacíos no se asocian a un pasaje específico del texto sino que son más bien etiquetas aisladas; por ejemplo, un salto de línea tiene este formato: <br />.

**¿Las etiquetas hay que escribirlas en mayúscula o en minúscula?**

A la mayoría de los navegadores les da igual si se escriben en mayúscula, minúscula o una mezcla de las dos. <HTML>, <html> or <HtMl> normalmente tendrán el mismo resultado. Sin embargo, el modo **correcto** es escribir las etiquetas en minúscula. Así que **acostúmbrate a escribir las etiquetas en minúscula**.

**¿Y dónde coloco todas estas etiquetas?**

Las etiquetas se escriben dentro de un documento HTML. Todo sitio web contiene uno o más documentos HTML. Cuando navegas por la web, no haces otra cosa que abrir diferentes documentos HTML.

**Crea tu primer sitio web**

Con lo que ya has aprendido en las lecciones anteriores, sólo te quedan unos minutos para crear tu primer sitio web.

**¿Cómo?**

En la lección 1 vimos qué se necesita para crear un sitio web: un navegador y el Bloc de notas (o un editor de texto similar). Puesto que estás leyendo esto, lo más probable es que ya tengas abierto el navegador. Lo único que necesitas es abrir otra ventana del navegador de forma que puedas leer este tutorial y veas cómo va quedando tu nuevo sitio web al mismo tiempo.

Abre también el Bloc de notas (que se encuentra en el menú de Inicio, Todos los programas, en la sección Accesorios):

¡Y ya estamos listos!

**¿Y qué puedo hacer?**

Empecemos con algo sencillo. Que te parece una página que diga: "¡Hurra! Esta es mi primera página web." Sigue leyendo y verás qué sencillo es.

HTML is sencillo y lógico. **El navegador lee el código HTML igual que lo haces tú: de arriba abajo y de izquierda a derecha**. Así pues, todo documento HTML empieza con lo que debería aparecer en primer lugar en la página y termina con lo que debería ir en último lugar.

Lo primero que tienes que hacer es decirle al navegador que le "hablarás" usando el lenguaje HTML. Esto se consigue con la etiqueta <html> (sin sorpresas, ¿no?). Así que antes de hacer cualquier otra cosa, escribe "<html>" en la primera línea del documento creado con el Bloc de notas.

Como recordarás de las lecciones anteriores, <html> es una etiqueta de inicio y tiene que cerrarse con su correspondiente etiqueta de cierre cuando hayas terminado de escribir el código HTML. Así que para no olvidar la etiqueta de cierre del elemento HTML escribe "</html>" un par de líneas más abajo, y entre ambas etiquetas <html> y </html> escribe el resto del documento.

Lo siguiente que necesita el documento es una "cabecera", que proporcionará información relativa a este mismo documento, y un "cuerpo" donde se incluirá el contenido del documento. Puesto que HTML es de lo más lógico, la cabecera (es decir, las etiquetas <head> y </head>) se encuentra encima del cuerpo (o sea, encima de las etiquetas <body> y </body>).

El documento debería presentar ahora este aspecto:

<html>

<head>

</head>

<body>

</body>

</html>

Presta atención a cómo hemos estructurado las etiquetas introduciendo nuevas líneas (usando la tecla Enter) al igual que hemos usado sangrado (utilizando la tecla Tab). En principio, da igual cómo estructures el documento HTML; sin embargo, para que te sirva de ayuda tanto para ti y como para otros a la hora de interpretar el código fuente, y con el fin tener una buena visión de conjunto, **es recomendable estructurar el código HTML de una forma nítida, con saltos de línea y sangrados**, como en el ejemplo anterior.

Si el documento se parece al del ejemplo anterior, ya has creado tu primer sitio web: un sitio web especialmente aburrido y en modo alguno parecido a lo que soñabas cuando empezaste este tutorial, pero, con todo y con eso, ya es algo que parece un sitio web. El código que acabas de crear te servirá como plantilla básica para todos tus futuros documentos HTML.

**Hasta ahora todo va bien, pero ¿cómo añado contenido a mi sitio web?**

Como aprendiste anteriormente, todo documento HTML tiene dos partes: un encabezado y un cuerpo. **En la sección del encabezado se incluye información sobre la propia página, mientras que el cuerpo contiene la información que constituye la página en sí.**.

Por ejemplo, si quieres poner a la página un título que aparecerá en la barra de títulos del navegador, deberías hacerlo en la sección del encabezado, es decir dentro de las etiquetas "head". El elemento que se usa para el título es title, o sea, escribe el título de la página entre la etiqueta de inicio <title> y la etiqueta de cierre </title>:

<title>Mi primer sitio web</title>

Ten en cuenta que este título no aparecerá en la página en sí. Cualquier cosa que quieras que aparezca en la página es contenido y debe, por la tanto, añadirse entre las etiquetas <body> y </body>.

Como lo prometido es deuda, vamos a conseguir que la página nos diga: "¡Hurra! Este es mi primer sitio web." Este es el texto que queremos transmitir, y, por lo tanto, pertenece a la sección del cuerpo. Así pues, en la sección del cuerpo, escribe lo siguiente:

<p>¡Hurra! Este es mi primer sitio web.</p>

La p de <p> es la abreviatura de "párrafo", que es exactamente lo que es: un párrafo de texto.

El documento HTML debería tener ahora este aspecto:

<html>

<head>

<title>Mi primer sitio web</title>

</head>

<body>

<p>¡Hurra! Este es mi sitio web.</p>

</body>

</html>

¡Hecho! ¡Acabas de crear ahora mismo tu primer sitio web de verdad!

Lo siguiente que hay que hacer es guardarlo en el disco duro y luego abrirlo en el navegador:

* En el Bloc de notas elige "Guardar como..." del menú "Archivo" en el menú principal.
* Elige "Todos los archivos" en la caja "Tipo". Esto es muy importante, pues, de lo contrario, se guardará como documento de texto y no como un documento HTML.
* Ahora guarda el documento como "pagina1.htm" (la terminación ".htm" indica que es un documento HTML; con la terminación ".html" se consigue el mismo resultado. Personalmente, siempre uso ".htm"; tú puedes elegir cualquiera de las dos extensiones, según prefieras). Da igual dónde guardes el documento en el disco duro, siempre que recuerdes dónde lo hiciste para poder recuperarlo con posterioridad.

Ahora pasamos al navegador:

* En el menú principal elige "Abrir" del menú "Archivo" (CTRL+O)
* Haz clic en el botón "Examinar" de la caja que aparece.
* Ahora localiza el documento HTML y haz clic en "Abrir".

En el navegador debería poder leerse ahora: "¡Hurra! Este es mi primer sitio web." ¡Enhorabuena!

**Empieza siempre con la plantilla básica que creamos en la lección anterior:**

<html>

<head>

<title></title>

</head>

<body>

</body>

</html>

En la sección del encabezado, incluye siempre un título: <title> El título de la página </title>. Fíjate cómo aparece el título en la esquina superior izquierda del navegador:

El título es especialmente importante porque los motores de búsqueda (por ejemplo, Google) lo utilizan para indexar tu sitio y porque aparece reflejado en los resultados de la búsqueda.

En la sección del cuerpo, se escribe el contenido propiamente dicho de la página. Ya conoces algunos de los elementos más importantes:

<p>Se usa para escribir párrafos</p>

<em>Muestra el texto enfatizado</em>

<h1>Encabezado de nivel 1</h1>

<h2>Encabezado de nivel 2</h2>

<h3>Encabezado de nivel 3</h3>

Recuerda: el único modo de aprender HTML es utilizando el método de la prueba y el error. Pero no te preocupes, en modo alguno vas a destrozar tu ordenador o Internet. Así que sigue experimentando; es el mejor modo de adquirir experiencia.

**¿Y esto qué se supone que significa?**

Nadie se convierte en un buen desarrollador web sólo con aprender los ejemplos de este tutorial. Lo que este tutorial te proporciona son sencillamente unos conocimientos básicos de los elementos esenciales; para llegar a ser bueno tienes que usar estos elementos esenciales de formas nuevas y creativas.

Así pues, lánzate de lleno y empieza a volar por ti mismo... Vale, puede que aún no sea el momento. Pero, de todos modos, inténtalo y experimenta con lo que has aprendido.

**¿Y ahora qué sigue?**

Intenta crear unas cuantas páginas tú mismo. Por ejemplo, crea una página con un título, un encabezado, algo de texto, un encabezado de segundo nivel y algo más de texto. No pasa nada por echarle un vistazo al tutorial mientras preparas estas primeras páginas. Pero, más adelante, intenta hacerlo tú solo... sin mirar.

**Más elementos**

¿Has conseguido crear algunas páginas tú solo? Si no es así, aquí tienes un ejemplo:

<html>

<head>

<title>Mi sitio web</title>

</head>

<body>

<h1>Encabezado de nivel 1</h1>

<p>texto, texto, texto, texto</p>

<h2>Encabezado de nivel 2</h2>

<p>texto, texto, texto, texot</p>

</body>

</html>

**¿Y ahora qué?**

Pues ha llegado el momento de aprender siete nuevos elementos.

Del mismo modo que enfatizamos el texto al incluirlo entre la etiqueta de inicio <em> y la etiqueta de cierre </em>, se puede acentuar más ese énfasis usando la etiqueta de inicio <strong> y la etiqueta de cierre </strong>.

**Ejemplo 1:**

<strong>Énfasis acentuado</strong>

y se verá así en el navegador:

**Énfasis acentuado.**

Asimismo, puedes hacer que el texto sea más pequeño usando small:

**Ejemplo 2:**

<small>Este texto tendría que ser un poco más pequeño.</small>

y se visualizará así en el navegador:

Este texto tendría que ser un poco más pequeño.

**¿Puedo usar varios elementos a la misma vez?**

Sí, se pueden usar fácilmente varios elementos a la vez siempre que **evitemos elementos superpuestos**. Esto lo entenderemos mejor con un ejemplo:

**Ejemplo 3:**

Si quieres que el texto aparezca enfatizado y más pequeño, se debe hacer de la siguiente manera:

<em><small>Texto enfatizado más pequeño</small></em>

y NO de la siguiente manera:

<em><small>Texto enfatizado más pequeño</em></small>

La diferencia estriba en que, en el primer ejemplo, cerramos en último lugar el elemento que abrimos primero. De este modo evitamos confundir al navegador y a nosotros mismos.

**¡Más elementos!**

Como mencionábamos en la lección 3, **existen una serie de elementos que se abren y cierran con la misma etiqueta**. Estos elementos conocidos como elementos vacíos no están relacionados con un pasaje específico del texto sino que son más bien etiquetas aisladas. Un ejemplo de dichas etiquetas es <br /> que crea un salto de línea forzado.

**Ejemplo 4:**

Algo de texto <br />y algo más de texto en una nueva línea que se visualizará así en el navegador:

Algo de texto  
y algo más texto en una nueva línea

Fíjate en que la etiqueta parece una forma contraída de la etiqueta de inicio y de la etiqueta de cierre con un espacio en blanco y una barra al final: <br />.

Otro elemento que se abre y cierra con la misma etiqueta es <hr />, que se usa para trazar una línea horizontal ("hr" es la abreviatura de "horizontal rule"):

**Ejemplo 5:**

<hr />

Que se verá así en el navegador:

Otros ejemplos de elementos que necesitan tanto la etiqueta de inicio como de cierre -la mayoría de elementos cumplen esta norma- son: ul, ol y li. Estos tres elementos se usan para crear listas.

**ul** is la forma abreviada de "unordered list" (es decir, lista no ordenada) e inserta viñetas para cada elemento.

**ol** es la abreviatura de "ordered list" (es decir, lista ordenada) y numera cada elemento de la lista.

Para crear elementos de lista use la etiqueta **li** ("list item", o sea, elemento de lista). Parece algo confuso, ¿no? Veamos, entonces, algunos ejemplos:

**Ejemplo 7:**

<ul>

<li>Un elemento de lista</li>

<li>Otro elemento de lista</li>

</ul>

que se verá así en el navegador:

* Un elemento de lista
* Otro elemento de lista

**Ejemplo 8:**

<ol>

<li>Primer elemento de lista</li>

<li>Segundo elemento de lista</li>

</ol>

que se visualizará así en el navegador:

1. Primer elemento de lista
2. Segundo elemento de lista

**¡Uf! ¿Eso es todo?**

De momento, sí. Una vez más, experimenta y crea tus propias páginas usando alguno de los siete nuevos elementos que acabas de aprender en esta lección:

<strong>Énfasis más fuerte</strong>

<small>Texto pequeño</small>

<br />Salto de línea

<hr />Línea horizontal

<ul>Lista sin ordenar</ul>

<ol>Lista ordenada</ol>

<li>Elemento de lista</li>

**Atributos**

Es posible añadir atributos a determinados elementos.

**¿Y qué es un atributo?**

Como recordarás, los elementos sirven para estructurar el contenido de todo el documento HTML e indican al navegador cómo debería presentarse el sitio web (por ejemplo, <br /> informa al navegador para que introduzca un salto de línea). En algunos elementos se puede añadir más información. Esa información extra se denomina atributo.

**Ejemplo 1:**

<h2 style="background-color:#ff0000;">Mi amistad con HTML</h2>

Los atributos se incluyen siempre dentro de la etiqueta de inicio y llevan inmediatamente detrás el signo de igual ("=") con los valores de dichos atributos entre comillas (""). El punto y coma que sigue al atributo se usa para separar los diferentes comandos de estilo. Más tarde volveremos a tratar sobre este tema.

Existen muchos atributos diferentes. El primero que vas a aprender es style. **Con el atributo style se puede añadir efectos de presentación a tu sitio web**. Por ejemplo, un color de fondo:

**Ejemplo 2:**

<html>

<head>

</head>

<body style="background-color:#ff0000;">

</body>

</html>

El código anterior mostrara un página con fondo rojo en el navegador; adelante, pruébalo tú mismo. En unos minutos explicaremos como más detalle cómo funciona el sistema de colores en HTML.

Ten en cuenta que algunas etiquetas y atributos usan ortografía estadounidense, es decir, "color" en vez de "colour". Es importante que utilices la misma ortografía que usamos en los ejemplos de este tutorial; de otro modo, los navegadores no serán capaces de interpretar el código. Además, no olvides cerrar siempre las comillas detrás del atributo.

**¿Qué ocurrió para que la página apareciera en color rojo?**

En el ejemplo anterior, hicimos referencia al color de fondo usando el código "#ff0000". Este es el código de color para el rojo en el sistema numérico hexadecimal (HEX). **Cada color tiene su propio número hexadecimal.**. Y estos son algunos ejemplos:

Blanco: #ffffff  
Negro: #000000 (ceros)  
Rojo: #ff0000  
Azul: #0000ff  
Verde: #00ff00  
Amarillo: #ffff00

El código de color hexadecimal se compone de una almohadilla, "#", y seis dígitos (del 0 al 9) o letras (de la A a la F). Existen más de 1.000 códigos hexadecimales y no es fácil calcular qué código hexadecimal corresponde a un color específico. Para facilitar las cosas hemos preparado una tabla con los 216 colores más usados: [Tabla de colores seguros para la web](http://es.html.net/tutorials/html/lesson7_216websafecolourchart.php).

También se pueden usar los nombres en inglés para los colores más comunues (white: blanco, black: negro, red: rojo, blue: azul, green: verde y yellow: amarillo).

**Ejemplo 3:**

<body style="background-color: red;">

De momento, ya vale de colores. Volvamos a los atributos.

**¿Qué elementos pueden usar atributos?**

**Son varios los atributos diferentes que se pueden aplicar a la mayoría de los elementos**.

Por ejemplo, usarás con frecuencia atributos en la etiqueta body, mientras que, por ejemplo, raras veces usarás atributos en la etiqueta br, ya que un salto de línea es un salto de línea que no tiene parámetros que ajustar.

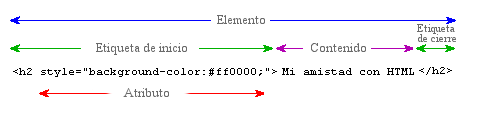
Al igual que existen muchos elementos, existen también muchos y diferentes atributos. Algunos atributos están hechos a la medida para un elemento concreto, mientras que otros se pueden usar para muchos elementos diferentes. Y viceversa: algunos elementos sólo pueden contener un tipo de atributo mientras otros pueden contener muchos.

Puede sonar algo confuso, pero una vez que te acostumbres a los diferentes atributos, es en realidad de lo más lógico y pronto verás qué fácil es usarlos y cuántas posibilidades brindan.

Este tutorial te presentará los atributos más importantes.

**¿Y de qué partes se compone exactamente un elemento?**

Por lo general, todo elemento se compone de una etiqueta de inicio, sin atributos o con uno o más atributos, el contenido correspondiente y una etiqueta de cierre. Así de sencillo. Observa la imagen inferior.



# Enlaces

En esta lección aprenderás a crear enlaces entre páginas.

## ¿Qué necesito para crear un enlace?

Para crear enlaces se usa lo que siempre usamos al codificar HTML: un elemento. **Un sencillo elemento con un atributo y podrás enlazar con cualquier cosa y con todo**. Vas a ver ahora mismo un ejemplo de cómo se codificaría un enlace a HTML.net:

### Ejemplo 1:

<a href="http://www.html.net/">Desde aquí enlazamos con HTML.net</a>

que se visualizará así en el navegador:

El elemento a es la abreviatura de "anchor" (ancla, en español). Y el atributo href es la abreviatura de "hypertext reference" (es decir, referencia hipertextual), que especifica hacia dónde nos dirige el enlace; normalmente, una dirección de internet o un nombre de fichero.

En el ejemplo anterior el atributo href contiene el valor "http://www.html.net", que es la dirección completa de HTML.net y que se denomina URL (Uniform Resource Locator; en español, localizador de recursos uniforme). Ten en cuenta que "http://" siempre debe incluirse en las URLs. El texto "Desde aquí enlazamos con HTML.net" es el que aparece en el navegador como enlace. Recuerda cerrar el elemento con la etiqueta de cierre </a>.

## ¿Y se pueden crear enlaces entre mis propias páginas?

Si quieres crear enlaces entre las páginas de un mismo sitio web, no es necesario incluir la dirección completa (el URL) del documento. Por ejemplo, si has creado dos páginas (llamémoslas pagina1.htm y pagina2.htm) y las has guardado en la misma carpeta, puedes enlazar una página con la otra con sólo escribir el nombre del fichero en el enlace. Si este es el caso, enlazar la pagina1.htm con la pagina2.htm tendría el siguiente aspecto:

### Ejemplo 2:

<a href="pagina2.htm">Haz clic para ir a la página 2</a>

Si la página 2 estuviera en una subcarpeta (llamada "subcarpeta"), el enlace tendría el siguiente aspecto:

### Ejemplo 3:

<a href="subcarpeta/pagina2.htm">Haz clic para ir a la página 2</a>

En sentido inverso, enlazar desde la página 2 (en la subcarpeta) a la página 1 se haría usando con el siguiente código:

### Ejemplo 4:

<a href="../pagina1.htm">Enlace a la pagina1</a>

Los dos puntos con la barra, "../", hacen referencia a una carpeta que se encuentra un nivel por encima de donde se encuentra el fichero desde el que se crea el enlace. Siguiendo este mismo sistema, se puede señalar a dos (o más) carpetas en niveles superiores escribiendo "../../".

¿Has captado cómo funciona el sistema? De forma alternativa, siempre puedes escribir la dirección completa del fichero (el URL).

## ¿Y se pueden crear enlaces internos dentro de una misma página?

También puedes crear enlaces internos dentro de una misma página; por ejemplo, una tabla de contenidos en la parte superior de la página con enlaces a cada capítulo que se encuentra más abajo. Todo lo que necesitas es usar un atributo muy útil llamado id (identification, en inglés; es decir, identificador) y el símbolo "#", conocido como almohadilla.

Usa el atributo id para marcar el elemento con el que quieras enlazar. Por ejemplo:

<h1 id="encabezado1">Primer encabezado</h1>

Ahora ya puedes crear un enlace a dicho elemento usando la almohadilla "#" en el atributo "href" del elemento a. El símbolo "#" debe ir seguido del id de la etiqueta con la que quieres enlazar. Por ejemplo:

<a href="#encabezado1">Enlace con el primer encabezado</a>

Vamos a ver un ejemplo para que todo quede claro:

### Ejemplo 5:

<html>

<head>

</head>

<body>

<p><a href="#encabezado1">Enlace con el primer encabezado</a></p>

<p><a href="#encabezado2">Enlace con el segundo encabezado</a></p>

<h1 id="encabezado1">Primer encabezado</h1>

<p>Texto texto texto texto</p>

<h1 id="encabezado2">Segundo encabezado</h1>

<p>Texto texto texto texto</p>

</body>

</html>

que se visualizará así en el navegador (haz clic en los dos enlaces):

[Enlace con el primer encabezado](http://es.html.net/tutorials/html/lesson8.php#encabezado1)

[Enlace con el segundo encabezado](http://es.html.net/tutorials/html/lesson8.php#encabezado2)

# Primer encabezado

Texto texto texto texto

# Segundo encabezado

Texto texto texto texto

(Nota: El atributo "id" debe empezar con una letra)

# ¿Puedo crear enlaces a otras cosas?

También puedes crear enlaces a una dirección de correo electrónico. Se hace casi igual que cuando se crea un enlace a un documento:

### Ejemplo 6:

<a href="mailto:nobody@html.net">Envíe un mensaje a nadie en HTML.net</a>

que se visualizará en el navegador así:

[Envíe un mensaje a nadie en HTML.net](mailto:nobody@html.net)

La única diferencia entre un enlace a una dirección de correo electrónico y un enlace a un fichero consiste en que en lugar de escribir la dirección del documento, se escribe: mailto: seguido de dicha dirección de correo electrónico. Al hacer clic sobre el enlace se abre el programa de correo por defecto con un nuevo mensaje en blanco dirigido a la dirección de correo especificada. Téngase en cuenta que esta función sólo funcionará si tienes instalado un programa de correo electrónico en tu ordenador. ¡Pruébalo!

## ¿Hay más atributos que debería conocer?

Para crear un enlace, siempre tienes que usar el atributo href. Además, también puedes poner un título (atributo title) al enlace:

### Ejemplo 7:

<a href="http://www.html.net/" title="Visita HTML.net y aprende HTML">HTML.net</a>

que se verá así en el navegador:

[HTML.net](http://www.html.net/)

El atributo title se usa para incluir una breve descripción del enlace. Si se coloca el cursor sobre el enlace -sin pulsarlo- aparecerá el texto "Visita HTML.net y aprende HTML".

[Home](http://es.html.net/) » [Tutoriales](http://es.html.net/tutorials/) » [HTML](http://es.html.net/tutorials/html/) » Lección 9: Imágenes

**Imágenes**

¿No sería genial poder tener la imagen del actor y leyenda musical David Hasselhoff en el mismísimo centro de la página?

**Eso suena como un gran desafío...**

Puede que sí, pero es bastante sencillo hacerlo. Todo lo que necesitas es un nuevo elemento:

**Ejemplo 1:**

<img src="david.jpg" alt="David" />

que se visualizará así en el navegador:



Lo que hay que hacer es indicar al navegador que quieres insertar una imagen, usando el elemento img y, luego, dónde está localizada por medio del atributo src (abreviatura de "source", es decir, fuente). ¿Lo has pillado?

Fíjate cómo el elemento img se abre y cierra usando la misma etiqueta. Al igual que la etiqueta <br />, no está relacionado con un pasaje específico de texto.

"david.jpg" es el nombre del fichero de imagen que quieres insertar en la página. ".jpg" es el tipo de fichero de la imagen. Al igual que la extensión ".htm" muestra que un fichero es un documento HTML, la extensión ".jpg" indica al navegador que dicho fichero es una imagen. Hay tres clases diferentes de tipos de fichero de imagen que puedes insertar en tus páginas:

* GIF (Graphics Interchange Format)
* JPG o JPEG (Joint Photographic Experts Group)
* PNG (Portable Network Graphics)

**Normalmente, las imágenes GIF son mejores para gráficos y dibujos, mientras que las imágenes JPEG son mejores para fotografías**. Esto es así por dos razones; la primera es que las imágenes GIF sólo constan de 256 colores, mientras que las imágenes JPEG constan de millones de colores, y la segunda es que el formato GIF es mejor que el formato JPEG a la hora de comprimir imágenes sencillas, ya que éste último está optimizado para imágenes más complejas. Cuanto mejor sea la compresión, y más pequeño el tamaño del fichero de imagen, más rápida se cargará la página. Como sabrás por experiencia propia, las páginas innecesariamente "pesadas" pueden ser extremadamente irritantes para el usuario.

Tradicionalmente, los formatos GIF y JPEG han sido los dos tipos de imagen dominantes, pero últimamente el formato PNG ha ganado cada vez más popularidad (principalmente a expensas del formato GIF). **El formato PNG contiene, en muchos aspectos, lo mejor de ambos formatos: millones de colores y una compresión efectiva**.

**¿Dónde puedo conseguir mis imágenes?**

Para crear tus propias imágenes, necesitas un programa de edición de imágenes. **Un programa de edición de imágenes es una de las herramientas esenciales que necesitarás para crear bonitos sitios web.**.

Por desgracia, Windows, al igual que otros sistemas operativos, no incluye un buen programa de edición de imágenes. Así pues, podrías pensar en invertir algo de dinero en uno de estos programas: Paint Shop Pro, Photoshop o Macromedia Fireworks, que son tres de los mejores programas de edición de imágenes del mercado en la actualidad.

Sin embargo, como ya dijimos anteriormente, no será necesario comprar programas caros para completar este tutorial. Por ahora, puedes descargar el excelente programa de edición de imágenes [IrfanView](http://www.irfanview.com/) que es un programa que se denomina "freeware" y, por lo tanto, no cuesta nada.

O bien puedes tomar prestadas imágenes de otros sitios descargándolas. Pero ten cuidado de no infringir derechos de autor al descargar dichas imágenes. Con todo, es útil saber cómo descargar las imágenes, así que aquí te mostramos cómo hacerlo:

1. Haz clic con el botón derecho sobre la imagen que quieras descargar de internet.
2. Elige "Guardar imagen como..." del menú emergente.
3. Elige la carpeta de tu ordenador donde quieres guardar la imagen y pulsa en "Guardar".

Haz esto con la imagen de más abajo y guárdala en tu ordenador en el mismo lugar que los documentos HTML. (Observa que el logo se guarda como un archivo PNG: logo.png):

logo de HTML.net

Ahora ya puedes insertar la imagen en una de tus propias páginas. Pruébalo.

**¿Eso es todo lo que necesito saber sobre imágenes?**

Hay unas cuantas cosas más que deberías saber respecto a las imágenes.

En primer lugar, puedes insertar fácilmente imágenes localizadas en otras carpetas, o incluso imágenes localizadas en otros sitios web:

**Ejemplo 2:**

<img src="imagenes/logo.png" />

**Ejemplo 3:**

<img src="http://www.html.net/logo.png" />

En segundo lugar, las imágenes se pueden usar como enlaces:

**Ejemplo 4:**

<a href="http://www.html.net"> <img src="logo.png" /></a>

que se visualizará así en el navegador (intenta hacer clic sobre la imagen):

[Logo de HTML.net](http://www.html.net/)

**¿Hay más atributos que tenga que conocer?**

Siempre hay que usar el atributo src, que indica al navegador dónde está localizada la imagen. Aparte de éste, hay otros atributos que pueden ser útiles a la hora de insertar imágenes.

El atributo alt se usa para ofrecer una descripción alternativa de la imagen si, por alguna razón, la misma no se puede mostrar al usuario. Esto es especialmente importante para las personas con discapacidades visuales, o si la página se carga de manera muy lenta. Por lo tanto, usa siempre el atributo alt:

**Ejemplo 5:**

<img src="logo.gif" alt="Logo de HTML.net" />

Algunos navegadores muestran el texto del atributo alt como una pequeña caja desplegable cuando el usuario coloca el cursor encima de la imagen. Ten en cuenta que el objetivo de usar el atributo alt es proporcionar una descripción alternativa de la imagen. El atributo alt no debería usarse para crear mensajes desplegables especiales dirigidos al usuario, ya que, en ese caso, las personas con discapacidades visuales oirán el mensaje sin saber cómo es la imagen.

El atributo title se puede usar para añadir información a la imagen:

**Ejemplo 6:**

<img src="logo.gif" title="Aprende HTML con HTML.net" />

que se verá de la siguiente manera en el navegador:

Logo de HTML.net

Si colocas el cursor sobre la imagen -sin llegar a hacer clic- verás aparecer el texto "Aprende HTML con HTML.net" en forma de caja desplegable.

Otros dos atributos importantes son width y height:

**Ejemplo 7:**

<img src="logo.png" width="141" height="32" />

que se verá así en el navegador:

HTML.net logo

Los atributos width y height se pueden usar para establecer la altura y la anchura de una imagen. El valor usado para establecer la altura y la anchura se expresa en píxeles. Los píxeles son unidades de medida utilizadas para medir la resolución de la pantalla. (La resolución de pantalla más habitual es 1024 x 768 píxeles). A diferencia de los centimetros, los píxeles son unidades de medida relativas que dependen de la resolución de la pantalla. Para un usuario con una resolución de pantalla alta, 25 píxeles pueden corresponder a 1 centímetro, mientras que los mismos 25 píxeles en una resolución de pantalla baja pueden corresponder a 1,5 centímetros sobre la pantalla.

Si no se establece la anchura y la altura, la imagen se insertará con su tamaño real. Sin embargo, con la anchura y la altura se puede manipular el tamaño:

**Ejemplo 8:**

<img src="logo.gif" width="32" height="32" />

que se visualizará así en el navegador:

HTML.net logo

Sin embargo, hay que tener en cuenta que el tamaño real en kilobytes de la imagen seguirá siendo el mismo, por lo que la imagen tardará en cargarse el mismo tiempo que antes, aunque aparezca más pequeña en la pantalla. Por lo tanto, **nunca deberías disminuir el tamaño de la imagen usando los atributos width y height**. En vez de hacer esto, siempre deberías redimensionar las imágenes usando un programa de edición de imágenes con el fin de que las páginas sean más ligeras y rápidas de cargar.

Dicho esto, sigue siendo buena idea usar los atributos width y height porque, de este modo, el navegador podrá detectar cuánto espacio necesitará la imagen en la presentación final de la página antes de que la imagen se haya descargado totalmente. Esto permite que el navegador componga correctamente la página con mayor rapidez.

Y con esto, de momento, ya hemos tenido suficiente respecto a David Hasselhoff y a las imágenes.

**Lección 10: Tablas**

**Las tablas se usan cuando necesitamos mostrar "datos en formato de tabla"**, o sea, información que se presenta de forma lógica en filas y columnas.

**¿Es difícil?**

Construir tablas con HTML puede parecer, al principio, complicado, pero si no pierdes la calma y andas con ojo, en realidad es de lo más lógico: como todo en HTML.

**Ejemplo 1:**

<table>

<tr>

<td>Celda 1</td>

<td>Celda 2</td>

</tr>

<tr>

<td>Celda 3</td>

<td>Celda 4</td>

</tr>

</table>

que se verá asi en el navegador:

|  |  |
| --- | --- |
| Celda 1 | Celda 2 |
| Celda 3 | Celda 4 |

**¿Qué diferencia hay entre <tr> y <td>?**

Como observarás en el ejemplo anterior, éste es probablemente el ejemplo más complicado de código HTML que hemos escrito hasta el momento. Vamos a desglosarlo y a explicar las diferentes etiquetas:

**Para insertar tablas se usan 3 elementos diferentes:**

* La etiqueta de inicio <table> y la etiqueta de cierre </table> inician y finalizan la tabla. Lógico.
* <tr> es la abreviatura de "**t**able **r**ow" (es decir, fila de la tabla) e inicia y finaliza las filas horizontales. Lógico también.
* <td> es la abreviatura de "**t**able **d**ata" (es decir, datos de la tabla). Esta etiqueta inicia y finaliza cada una de las celdas que componen las filas de la tabla. Todo sencillo y lógico.

Y esto es lo que ocurre en el ejemplo 1: la tabla empieza con <table>, seguida de <tr>, que indica el inicio de una nueva fila. Se insertan dos celdas en esta fila: <td>Celda 1</td> y <td>Celda 2</td>. La fila se cierra a continuación con </tr> e inmediatamente después se inicia una nueva fila con <tr>. La nueva fila contiene dos celdas. Y por fin la tabla se cierra con </table>.

Vamos a dejarlo claro: **las filas son líneas horizontales de celdas y las columnas son líneas verticales de celdas**:

|  |  |
| --- | --- |
| Celda 1 | Celda 2 |
| Celda 3 | Celda 4 |

La celda 1 y la Celda 2 forman una fila, mientras que la Celda 1 y la Celda 3 forma una columna.

En el ejemplo anterior, la tabla consta de dos filas y dos columnas. Sin embargo, una tabla puede contener una cantidad ilimitada de filas y columnas.

**Ejemplo 2:**

<table>

<tr>

<td>Celda 1</td>

<td>Celda 2</td>

<td>Celda 3</td>

<td>Celda 4</td>

</tr>

<tr>

<td>Celda 5</td>

<td>Celda 6</td>

<td>Celda 7</td>

<td>Celda 8</td>

</tr>

<tr>

<td>Celda 9</td>

<td>Celda 10</td>

<td>Celda 11</td>

<td>Celda 12</td>

</tr>

</table>

que se visualizará así en el navegador:

|  |  |  |  |
| --- | --- | --- | --- |
| Celda 1 | Celda 2 | Celda 3 | Celda 4 |
| Celda 5 | Celda 6 | Celda 7 | Celda 8 |
| Celda 9 | Celda 10 | Celda 11 | Celda 12 |

**¿Existe algún atributo para el elemento table?**

Por supuesto, hay atributos. Por ejemplo, el atributo border se usa para especificar el grosor del borde de la tabla:

**Ejemplo 3:**

<table border="1">

<tr>

<td>Celda 1</td>

<td>Celda 2</td>

</tr>

<tr>

<td>Celda 3</td>

<td>Celda 4</td>

</tr>

</table>

Que se verá así en el navegador:

|  |  |
| --- | --- |
| Celda 1 | Celda 2 |
| Celda 3 | Celda 4 |

Al igual que con las imágenes, se puede establecer el ancho de la tabla en píxeles, o, de modo alternativo, por medio de un porcentaje del ancho de la pantalla:

**Ejemplo 4:**

<table border="1" width="30%">

La tabla del anterior ejemplo se mostrará en el navegador como una tabla con un ancho equivalente al 30 % de la pantalla. Pruébalo.

**¿Hay algún otro atributo?**

Para las tablas existen muchos atributos. Aquí mostramos dos más:

* align: especifica la alineación horizontal del contenido de toda la tabla, de una fila o de una sola celda. Por ejemplo, left, center o right.
* valign: especifica la alineación vertical del contenido de una celda. Por ejemplo, top, middle o bottom.

**Ejemplo 5:**

<td align="right" valign="top">Celda 1</td>

**¿Qué puedo insertar en las tablas?**

En teoría, se puede colocar cualquier cosa dentro de una tabla: texto, enlaces, imágenes, ... **Sin embargo, las tablas se usan para presentar datos en forma de tabla** (es decir, datos que se pueden presentar de forma coherente en columnas y filas), así que evita colocar cosas dentro de una tabla porque quieras colocarlas una al lado de otra.

En los viejos tiempos de Internet -es decir, hace unos años-, las tablas se usaban con frecuencia como herramienta de presentación. Sin embargo, si quieres controlar la presentación de los textos y las imágenes existe un modo más guay de hacerlo (pista: CSS). Pero de eso ya trataremos más adelante.

Ahora, empieza a poner en práctica lo que acabas de aprender y diseña varias tablas de diferentes tamaños, con diferentes atributos y contenido. Esto debería de darte para unas cuantas horas de trabajo

**y más tablas**

El título "... y más tablas" puede sonar algo aburrido. Pero míralo por el lado bueno; cuando domines las tablas, nada de nada en HTML te podrá dejar K.O.

**Entonces, ¿qué más queda por conocer?**

Los atributos colspan y rowspan se usan cuando queremos crear tablas elaboradas.

Colspan es la abreviatura de "column span" (expandir columna). **Colspan se usa con la etiqueta <td> para especificar cuántas columnas ocupará la celda**:

**Ejemplo 1:**

<table border="1">

<tr>

<td colspan="3">Celda 1</td>

</tr>

<tr>

<td>Celda 2</td>

<td>Celda 3</td>

<td>Celda 4</td>

</tr>

</table>

que se visualizará así en el navegador:

|  |  |  |
| --- | --- | --- |
| Celda 1 | | |
| Celda 2 | Celda 3 | Celda 4 |

Al establecer el atributo colspan a "3", la celda de la primera fila ocupa tres columnas. Si hubiésemos establecido colspan a "2", la celda sólo habría ocupado dos columnas y habría sido necesario insertar una celda adicional en la primera fila para que la cantidad de columnas sea igual en las dos filas.

**Ejemplo 2:**

<table border="1">

<tr>

<td colspan="2">Celda 1</td>

<td>Celda 2</td>

</tr>

<tr>

<td>Celda 3</td>

<td>Celda 4</td>

<td>Celda 5</td>

</tr>

</table>

que se verá así en el navegador:

|  |  |  |
| --- | --- | --- |
| Celda 1 | | Celda 2 |
| Celda 3 | Celda 4 | Celda 5 |

**¿Y el atributo rowspan?**

Como ya habrás adivinado, **rowspan especifica cuántas filas debería ocupar una celda**:

**Ejemplo 3:**

<table border="1">

<tr>

<td rowspan="3">Celda 1</td>

<td>Celda 2</td>

</tr>

<tr>

<td>Celda 3</td>

</tr>

<tr>

<td>Celda 4</td>

</tr>

</table>

que se verá así en el navegador:

|  |  |
| --- | --- |
| Celda 1 | Celda 2 |
| Celda 3 |
| Celda 4 |

En el ejemplo anterior el atributo rowspan se establece con el valor "3" en la Celda 1. Esto especifica que la celda debe expandirse a lo largo de tres filas (o sea, ocupa la suya propia y dos más). La Celda 1 y la Celda 2 , por tanto, están en la misma fila, mientras que la Celda 3 y la Celda 4 forman dos filas independientes.

¿Confundido? Bueno, la verdad es que no es sencillo, y es fácil perderse. Por lo tanto, sería una buena idea dibujar la tabla sobre un trozo de papel antes de empezar a escribir el código HTML.

¿Que no estas confuso? Pues sigue adelante y crea un par de tablas tú solo usando los atributos colspan y rowspan

**Presentación (CSS)**

¿No sería genial poder dar a tus páginas la presentación que se merecen?

**Pues claro que sí; pero... ¿cómo?**

**Para la presentación de tu sitio web usa Cascading Style Sheets (CSS), es decir, hojas de estilo en cascada.** En esta lección te presentaremos una breve introducción a CSS. Sin embargo, más adelante, podrás aprender todo lo relacionado con CSS, desde cero, en nuestro [Tutorial sobre CSS](http://es.html.net/tutorials/css/). Así pues, considera esta lección como un pequeño aperitivo.

CSS se puede considerar la media naranja de HTML. Y a la hora de escribir código, no hay punto de comparación en cuanto a su situación: **HTML se encarga del contenido en bruto de la página (la estructura), mientras que CSS le proporciona ese toque elegante (la presentación)**.

Como vimos en la [Lección 7](http://es.html.net/tutorials/html/lesson7.php), **se pueden añadir estilos con CSS recurriendo al atributo style**. Por ejemplo, por medio de dicho atributo se puede establecer el tamaño y tipo de fuente de un párrafo:

**Ejemplo 1:**

<p **style="font-size:20px;"**>El tamaño de este párrafo es de 20 píxeles</p>

<p **style="font-family:courier;"**>El tipo de letra usado en este párrafo es Courier</p>

<p **style="font-size:20px; font-family:courier;"**>Este párrafo combina tipo de letra

Courier con un tamaño de 20 píxeles</p>

y se verá así en el navegador:

El tamaño de este párrafo es de 20 píxeles

El tipo de letra usado en este párrafo es Courier

Este párrafo combina tipo de letra Courier con un tamaño de 20 píxeles

En el ejemplo anterior hemos usado el atributo style para especificar el tipo de fuente que vamos a usar (por medio de la propiedad font-family) y el tamaño de la fuente (por medio de la propiedad font-size). Observa cómo en el último párrafo establecemos tanto el tipo de fuente como el tamaño separando ambas propiedades con un punto y coma.

**¿Y no parece un trabajo hercúleo?**

**Una de la características "elegantes" de CSS es la posibilidad de gestionar la presentación de manera centralizada.** En vez de usar el atributo style en cada etiqueta de inicio, se puede indicar al navegador, de una sola vez, cómo tiene que presentar el texto en la página:

**Ejemplo 2:**

<html>

<head>

<title>Mi primera página con estilos CSS</title>

**<style type="text/css">**

**h1 {font-size: 30px; font-family: arial;}**

**h2 {font-size: 15px; font-family: courier;}**

**p {font-size: 8px; font-family: "times new roman";}**

**</style>**

</head>

<body>

<h1>Mi primera página con estilos CSS</h1>

<h2>Bienvenidos a mi primera página con estilos CSS</h2>

<p>Aquí podéis ver cómo funciona CSS</p>

</body>

</html>

En el ejemplo anterior, **el código CSS se ha insertado en la sección de encabezado (head), aplicándose, por tanto, a toda la página.**. Para hacer esto, utiliza la etiqueta style añadiéndole el atributo type y como valor de éste text/css, es decir: <style type="text/css">, con lo que le estamos indicando al navegador que lo que estamos escribiendo es código CSS.

En el ejemplo todos los encabezados de nivel 1 aparecerán con letra de tipo Arial y tamaño de 30 píxeles; todos los encabezados de nivel 2 se mostrarán con tipo de letra Courier y tamaño de 15 píxeles; y el texto de un párrafo normal usará el tipo de letra Times New Roman con tamaño de 8 píxeles.

Otra opción es crear el código CSS en un documento independiente. **Con un documento CSS independiente se puede gestionar la presentación de muchas páginas a la misma vez.**. Esta opción es muy inteligente si se quiere cambiar el tipo de fuente o el tamaño a un sitio web con cientos o miles de páginas. Ahora no profundizaremos en este aspecto, pero lo puedes aprender más tarde en nuestro tutorial sobre CSS.

**¿Qué más puedo hacer con CSS?**

CSS se puede usar para muchas más cosas que especificar tipos y tamaños de fuente. Por ejemplo, se puede añadir colores y fondos. Aquí te presentamos algunos ejemplos para que experimentes con ellos:

<p **style="color:green;"**>Texto de color verde</p>

<h1 **style="background-color: blue;"**>Encabezado de nivel 1 con fondo azul</h1>

<body **style="background-image: url ('http://www.html.net/logo.png');"**>

Prueba a insertar los ejemplos anteriores en alguna de tus páginas, tal como aparecen anteriormente y bien como código CSS insertado en la sección head del documento.

**¿CSS sólo trata de colores y tipos de fuente?**

Aparte de añadir presentación relativa a colores, tipos de fuente, etc., **CSS también se puede usar para controlar la configuración de la página y la presentación de la misma** (márgenes, elementos flotantes, alineación, anchura, altura, etc.). Al establecer los diferentes elementos con CSS podrás presentar tus páginas de forma elegante y precisa.

**Ejemplo 3:**

<p **style="padding:25px;border:1px solid red;"**>Me encanta CSS</p>

que se visualizará así en el navegador:

Me encanta CSS

Con la propiedad float se puede hacer flotar un elemento bien a la derecha o a la izquierda. El siguiente ejemplo muestra este principio:

**Ejemplo 4:**

<img src="bill.jpg" alt="Bill Gates" **style= "float:left;"** />

<p>Lorem ipsum dolor sit amet, consectetuer adipiscing elit,

sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam

erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exercitation

ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat...</p>

que se verá así en el navegador:



Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat...

En este ejemplo un elemento (la imagen) flota a la izquierda y el otro elemento (el texto) rellena el hueco.

Con la propiedad position se puede colocar un elemento en el lugar preciso que se quiera de la página:

**Ejemplo 5:**

<img src="bill.jpg" alt="Bill Gates" **style="position:absolute;bottom:50px;right:10px;"** />

En el ejemplo anterior la imagen se coloca a 50 píxeles del borde inferior y a 10 píxeles del borde derecho del navegador. Pero se puede colocar exactamente donde se quiera. Pruébalo. Muy fácil y llamativo, ¿verdad?

**Llamativo, sí. Pero... ¿fácil?**

**No se aprende CSS en diez minutos**.

De momento, concéntrate en HTML, y pasa a la siguiente lección donde aprenderás cómo publicar tu sitio web en Internet para que todo el mundo pueda admirarlo.

**Sube tus páginas a Internet**

Hasta ahora, sólo tú has tenido la satisfacción de ver tus páginas. Este es el momento de que el resto del mundo pueda ver tu obra maestra.

**¿Está el mundo preparado para eso?**

El mundo lo está... y tú también lo estarás bien pronto. **Para que tu sitio web forme parte de Internet sólo necesitas espacio en un servidor y un programa FTP gratuito.**

Si tienes acceso a Internet, podría ser que ya tuvieras espacio en algún servidor gratuito para tu sitio web. Si es así, el espacio en el servidor se llamará algo así: http://home.provider.com/~usernumber. Pero antes de nada es probable que tuvieras que activarlo. Sigue leyendo sobre esto en la información que te haya proporcionado tu servidor de Internet o en sus páginas de asistencia al cliente.

Otra opción es conseguir espacio en algún servidor gratuito de Internet. Del mismo modo que puedes crear una cuenta de correo electrónico (por ejemplo, en Hotmail), **te puedes registrar para conseguir espacio en un servidor gratuito de Internet**. Son varias las compañías que ofrecen este servicio, entre ellas [000webhost.com](http://www.000webhost.com/) (hac clic en "Order" ("Registrarse") y elige "free membership" ("Socio gratuito"); no tardarás más de un par de minutos en registrarte.

Para acceder al servidor es necesario saber el nombre del servidor ("Host Name", por ejemplo, ftp.htmlnet.site50.net) y tener a mano tu nombre de usuario ("username") y contraseña ("password").

**¿Es eso todo lo que necesito?**

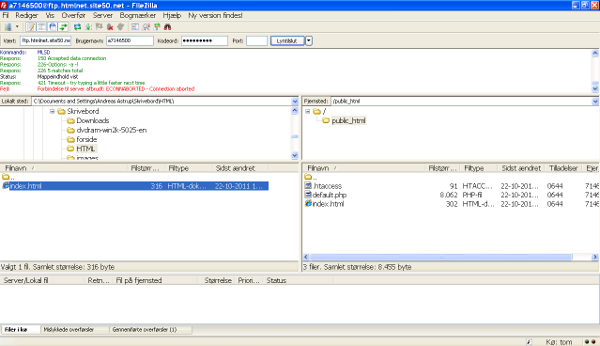
Para acceder al servidor y subir las páginas también necesitarás un programa FTP. FTP es la abreviatura de File Transfer Protocol (protocolo de transferencia de archivos). El programa FTP se usa para conectar dos ordenadores en Internet, de modo que puedas transferir ficheros desde tu ordenador a otro ordenador (en nuestro caso, el servidor). Puede que aún no tengas este tipo de programa, pero, por suerte, algunos de ellos se pueden descargar de forma gratuita.

**Existen muchos programas FTP diferentes. Uno de los mejores es FileZilla, que es totalmente gratuito**. Así que ya puedes ir descargándote FileZilla de la página [filezilla.sourceforge.net](http://filezilla.sourceforge.net/).

**¿Y cómo subo las páginas?**

Más abajo describimos cómo subir tus páginas con FileZilla a una cuenta gratuita en 000webhost.com. Pero el procedimiento es igual, poco más o menos, para todos los proveedores y programas ftp.

Abre el programa FTP cuando estés conectado a Internet. Inserta "Nombre del servidor" ("Host Name", "ftp.htmlnet.site50.net") en el campo "Dirección" ("Address"), "nombre de usuario" (en el campo "User") y "contraseña" (en el campo "Password") y haz clic en "Conectar" ("Connect"). Ahora mismo ya deberías tener acceso al servidor. En un lado del programa puedes ver los contenidos de tu ordenador ("Local Site" o "sitio local"), y en el otro lado puedes ver el contenido del servidor ("Remote Site" o "sitio remoto"):



Localiza los documentos HTML y las imágenes en tu ordenador (en "Local Site") y transfiérelos al servidor (en "Remote Site") haciendo doble clic sobre ellos. ¡Ahora ya puede verlos todo el mundo! (Por ejemplo, en la dirección http://htmlnet.site50.net/pagina1.htm).

Dale a una de las páginas el nombre "index.htm" (o "index.html") y de forma automática se convertirá en la página de inicio. Es decir, si escribes http://htmlnet.site50.net (sin ningún nombre de fichero) la página que abrirás realmente será http://htmlnet.site50.net/index.htm.

**A la larga, podría ser una buena idea comprar tu propio dominio** (por ejemplo, www.tu-nombre.com o www.tu-nombre.net) y evitar las largas y complicadas direcciones que te asignará tu proveedor de Internet o los proveedores de servidores de espacio gratuito. Puedes encontrar y comprar dominios, por ejemplo, en las direcciones [Speednames](http://www.speednames.com/) o [NetworkSolutions](http://www.networksolutions.com/).

En esta lección, adquirirás nociones algo más teóricos sobre HTML.

**¿Qué más hay que saber sobre HTML?**

HTML se puede codificar de muchas maneras diferentes. Y los navegadores pueden interpretar HTML de todas esas mismas maneras. Se podría decir que **HTML tiene muchos dialectos**. Esta es la razón de que algunos sitios web se visualizan de forma diferente en navegadores diferentes.

**Por parte del** [**Consorcio del World Wide Web (W3C)**](http://www.w3.org/), fundado por Tim Berners-Lee (¡sí, ese gran tipo que inventó HTML!), se han producido intentos para crear un estándar común de HTML. Pero éste ha sido un camino largo y difícil.

En los viejos tiempos, cuando para tener un navegador había que pagarlo, Netscape era el navegador dominante. Por aquel entonces, el estándar HTML con mayor soporte eran la versión 2.0 y, más tarde, la versión 3.2. Sin embargo, con una cuota de mercado de más del 90%, Netscape no tenía por qué preocuparse, ni se preocupaba, de seguir unos estándares comunes. Todo lo contrario, Netscape inventó sus propios elementos extraños, que no funcionaban en otros navegadores.

Durante muchos años Microsoft dejó de lado Internet casi por completo. Pasado un tiempo, reanudó la competencia con Netscape y presentó un navegador. Las primeras versiones del navegador de Microsoft, el Internet Explorer, no eran muchos mejores que las de Netscape a la hora de soportar los estándares HTML. Pero Microsoft se decantó por ofrecer su navegador de forma gratuita (algo siempre muy popular) e Internet Explorer pronto pasó a ser el navegador más popular.

A partir de las versiones 4 y 5 de Internet Explorer, Microsoft se propuso soportar cada vez más los estándares del W3C. Netscape, por su parte, no consiguió desarrollar un nueva versión de su navegador y siguió distribuyendo la desfasada versión 4.

El resto es historia. Hoy día los estándares HTML son la versión 4.01 y XHTML. Ahora es Internet Explorer el que acapara más del 90% de cuota de mercado; sigue teniendo sus propios elementos extraños, pero también soporta los estándares HTML del W3C. Al igual que lo hacen los demás navegadores, como, por ejemplo, Mozilla, Opera and Netscape.

Así pues, **cuando se escribe código HTML conforme con los estándares del W3C, creamos sitios web que se pueden ver en todos los navegadores**... tanto en los actuales como en los futuros. Y, por suerte, **lo que has aprendido en este tutorial es una nueva versión más estricta y limpia de HTML llamada XHTML**.

**¿Cómo indico qué versión uso?**

Teniendo en cuenta los diferentes tipos de dialectos HTML existentes, es necesario indicar al navegador qué "dialecto" utiliza tu código HTML, en este caso XHTML. Para hacer esto, usa una DTD ("Document Type Declaration", declaración de tipo de documento) que debe aparecer siempre al inicio del documento:

**Ejemplo 1:**

**<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"**

**"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">**

**<html xmlns="http://www.w3.org/1999/xhtml" lang="en">**

<head>

<title>Título</title>

</head>

<body>

<p>texto, texto</p>

</body>

</html>

Aparte del DTD (la primera línea del ejemplo anterior), que indica al navegador que vas a usar código escrito en XHTML, también es necesario insertar información adicional en la etiqueta html por medio de los atributos xmlns y lang.

xmlns is la abreviatura de "XML-Name-Space" (espacio de nombre XML) y siempre debería tener el valor **http://www.w3.org/1999/xhtml**. Eso es todo lo que necesitas saber respecto al atributo xmlns; pero si lo que tienes es una sed desmesurada de conocimientos sobre temas complicados, te invitamos a que sigas leyendo sobre espacios de nombre en el sitio web del [W3C](http://www.w3.org/TR/1999/REC-xml-names-19990114/).

Con el atributo lang se especifica en qué idioma se ha escrito el documento. Para esto se usa el estándar [ISO 639](http://www.w3.org/WAI/ER/IG/ert/iso639.htm#2letter), que presenta un listado de códigos de todos los idiomas del mundo. En el ejemplo anterior el idioma usado es el inglés ("en").

Con la DTD, el navegador sabe con exactitud cómo interpretar y mostrar el código HTML. Así pues, usa el ejemplo anterior como plantilla para todos los documentos futuros que vayas a crear.

La DTD también es importante a la hora de validar tus páginas web.

**¿Validar? ¿Por qué y cómo debería hacer eso?**

Al insertar la DTD en tus páginas siempre podrás comprobar el código HTML para revisar posibles errores usando el [validador gratuito del W3C](http://validator.w3.org/).

Para probarlo, crea una página y súbela a Internet. Ahora abre la página del [validator.w3.org](http://validator.w3.org/) y escibe la dirección (el URL) de tu página y valídala. Si el código HTML es correcto recibirás un mensaje de enhorabuena; si no, recibirás un informe de errores que te indicará qué error se ha producido y dónde se localiza el mismo. Comete algún error que otro a propósito para ver qué ocurre.

El validador no es sólo útil para localizar errores. Algunos navegadores intentan compensar la falta de destreza entre los desarrolladores web solucionando errores en el código HTML y mostrando las páginas como suponen que deberían visualizarse. Con este tipo de navegadores, podría ser que nunca llegaras a ver un error en tu propio navegador. Sin embargo, otros navegadores podrían intentar mostrarlas de forma diferente o no mostrarlas en absoluto. El validador puede servirte de gran ayuda para encontrar errores que ni siquiera sabías que existían.

Valida siempre tus páginas para asegurarte de que se mostrarán de forma correcta.

**Últimos consejos**

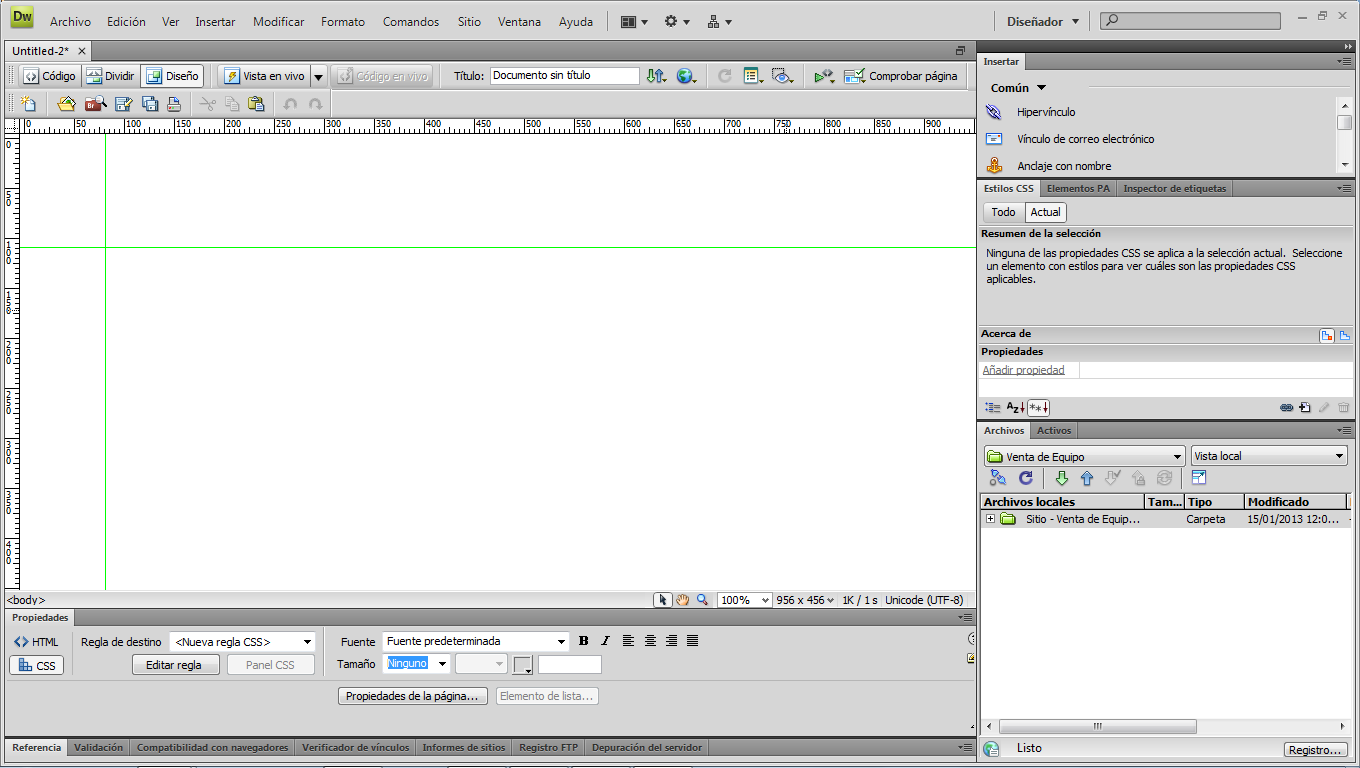
Enhorabuena, ya has llegado a la última lección.

**Y ahora, ¿ya sé todo lo que hay que saber?**

¡Has aprendido mucho, y ya eres capaz de crear tus propios sitios web! Sin embargo, lo que has aprendido son los principios básicos y aún te queda mucho para dominar HTML. sin embargo, ya tienes una buena base a partir de la cual puedes seguir avanzando.

En esta última lección te vamos a proporcionar algunos consejos de última hora:

* En primer lugar, es una buena idea mantener un cierto orden y estructura en tus documentos HTML. Al publicar documentos bien estructurados no sólo mostrarás a los demás tu dominio de HTML, sino que, además, te será considerablemente más facil tener una visión de conjunto.
* Cíñete a los estándares y valida tus páginas. Esto hay que subrayarlo una y otra vez; escribe siempre XHTML claro, usa la DTD adecuada y valida tus páginas en [validator.w3c.org](http://validator.w3.org).
* Llena tus páginas de contenido. Recuerda que HTML es una herramienta que permite presentar información en Internet, así que asegúrate de presentar dicha información. Las páginas bonitas pueden parecer agradables, pero la mayor parte de la gente usa Internet para buscar información.
* Evita sobrecargar tus páginas con imágenes pesadas y otras cosas divertidas que hayas podido encontrar en Internet, ya que el tiempo de carga de la página aumenta y esto podría confundir a los visitantes. Por ejemplo, las páginas que tardan más de 20 segundos en cargarse pueden llegar a perder hasta un 50 % de sus visitantes.
* Recuerda añadir tu sitio web a directorios y motores de búsqueda de modo que la gente, en general, y no sólo tus familiares, pueda encontarla y disfrutar de ella. En la página principal de todos los motores de búsqueda encontrarás un enlace para añadir nuevas páginas (el más importante es [Google](http://www.google.com), pero también existen otros como [DMOZ](http://www.dmoz.org), [Yahoo](http://www.yahoo.com), [AltaVista](http://www.altavista.com), [AlltheWeb](http://www.alltheweb.com) y [Lycos](http://www.lycos.com)).
* En este tutorial has aprendido a usar el Bloc de notas, que es un editor sencillo y muy fácil de usar, pero a lo mejor te puede ser de ayuda usar un editor más avanzado que proporcione una mejor visión de conjunto y más posibilidades. Puedes encontrar un resumen de diferentes editores y revisiones de los mismos en [Download.com](http://www.download.com).

Entorno Dreamweaver

## 

## LISTADO ETIQUETAS HTML

A continuación, voy a indicarte cuales son las etiquetas HTML más usadas y también te indicaré su finalidad. Voy a centrarme en las principales etiquetas y sobre todo las que están soportadas en **HTML 5**. Ahí van:

### ETIQUETAS INICIALES O DE RAÍZ

* **<!DOCTYPE html>** Indica al navegador que el documento está basado en el estándar HTML5
* **<html> </html>** Representa la raíz de un documento HTML. Todos los demás elementos de la estructura HTML deben ser recogidos dentro de estas etiquetas.

### METADATOS DEL DOCUMENTO

* **<head> </head>** Representa una colección de metadatos acerca del documento, incluyendo enlaces a, o definiciones de, scripts y hojas de estilo. El resto de etiquetas de metadatos, **irán recogidas dentro de las etiquetas de apertura y cierre del head**. Importante explicar que estos metadatos del documento, es información para el navegador y no contenido que será visible en la página web. A excepción de la etiqueta <title> que veremos a continuación.
* **<title> </title>** Etiqueta usada para definir el título de la página web.
* **<link>** Se usa para enlazar recursos externos al documento HTML. El ejemplo más común son **las hojas de estilos CSS.**
* **<meta>** Etiqueta usada para definir otros metadatos que no se pueden definir con una etiqueta HTML especifica. Por ejemplo para definir el autor del sitio, o la descripción del mismo.
* **<style> </style>** Etiquetas usadas para introducir código CSS en línea, es decir, en el propio documento HTML.

### ETIQUETAS DE SECCIONES O PARA ESTRUCTURAR EL HTML

* **<body> </body>** Al contrario que la etiqueta de metadatos <head>, todo lo que quieras mostrar en la página web debe ir recogido dentro de las etiquetas de apertura y cierre de <body>. **Este contenido será el que se muestre en la web.**
* **<nav> </nav>** Usadas para definir el contenido que será la sección de navegación de la web.
* **<main> </main>** Se usa para definir el contenido principal del documento. Solamente puede existir uno por documento.
* **<section> </section>** Define una sección del documento
* **<article> </article>** Define contenido independiente de la web.
* **<aside> </aside>** Dentro de estas etiquetas suele alojarse el contenido adicional de la web. Suele ser contenido relacionado con la web pero de poca importancia
* **<h1>,<h2>,<h3>,<h4>,<h5>,<h6>** Son etiquetas HTML muy importantes, ya que son usadas para jerarquizar el contenido de la web. Las etiquetas se usan para explicar brevemente el contenido que irá a continuación.
* **<header> </header>** Se usan para definir la cabecera la página web. Suele contener el logotipo, menú de navegación, etc.
* **<footer> </footer>** Usadas para definir el pie de página.

¿Sabías que también puedes [añadir un video de fondo HTML a tu página web](https://carontestudio.com/blog/como-poner-un-video-de-fondo-en-html/)?

¡Descubre cómo hacerlo!

### ETIQUETAS PARA LA AGRUPACIÓN DE CONTENIDO

* **<p> </p>** Etiqueta usada para escribir párrafos de texto.
* **<hr>** Etiqueta utilizada para «romper» entre dos secciones de una web. Usada comúnmente como separador.
* **<pre> </pre>** Usada para pegar texto manteniendo el pre formato propio del texto.
* **<blockquote> </blockquote>** Se usan para indicar que el contenido es texto citado.
* **<ol> </ol>** Etiquetas para crear una lista ordenada
* **<ul> </ul>** Etiquetas para crear una lista des-ordenada
* **<li> </li>** Etiquetas que recogen el contenido de un elemento de una lista, sea ordenada o no.
* **<dl> </dl>** Usada para crear una lista de definiciones.
* **<dt> </dt>** Representa un término definido por la siguiente etiqueta <dd>
* **<dd> </dd>** Se usa para definir los términos listados antes que él.
* **<figure> </figure>** Indica una figura ilustrada como parte del documento HTML5.
* **<figcaption> </figcaption>** Utilizada para definir la leyenda de una figura.
* **<div> </div>** Etiqueta común utilizada para crear un contenedor genérico.

*Complementa este listado de etiquetas con nuestro*[*listado de todos los atributos HTML*](https://carontestudio.com/blog/atributos-html/)*que existen.*

### ETIQUETAS SEMÁNTICAS PARA TEXTO

* **<a> </a>** Etiqueta utilizada para crear hiperenlaces en el documento HTML
* **<strong> </strong>** Etiqueta para definir una palabra o conjunto de ellas como importantes. **Tiene una fuerte**[**importancia en el SEO de la página**](https://carontestudio.com/blog/15-mitos-del-seo-en-2022/)**.**
* **<small> </small>** Utilizada para dejar un comentario aparte, del tipo una nota de derechos de autoría, u otros textos que no son esenciales para la comprensión del documento.
* **<cite> </cite>** Para indicar el título de una obra
* **<sub> </sub>** y **<sup> </sup>** Etiquetas utilizadas para representar un subíndice o superíndice.
* **<mark> </mark>** Usada para resaltar texto
* **<span> </span>** Etiqueta HTML sin ningún significado específico. Se usa conjuntamente con los atributos «class» o «id» para atribuirle ciertas características.
* **<br>** Etiqueta utilizada para crear un salto de línea

### ETIQUETAS PARA INCRUSTAR CONTENIDO

* **<img>** Etiqueta para «pintar» una imagen en la página web.
* **<iframe> </iframe>** Es una etiqueta que sirve para anidar otro documento HTML dentro del documento principal.
* **<embed>** Usada para integrar una aplicación o contenido interactivo externo que no suele ser HTML.
* **<object> </object>** Utilizada llamar a un recurso externo de la web. Este recurso será tratado como una imagen, o un recurso externo para ser procesado por un plugin.
* **<video> </video>** Se usa para reproducir [video en la página web](https://carontestudio.com/blog/como-poner-un-video-de-fondo-en-html/) junto a sus archivos de audio y capciones asociadas.
* **<audio> </audio>** Usada para[cargar en una web un archivo de audio](https://carontestudio.com/blog/como-anadir-un-reproductor-de-audio-html/) o stream de audio.
* **<source>** Permite a autores especificar recursos multimedia alternativos para las etiquetas de <video> o <audio>
* **<svg> </svg>** Se usa para llamar a una imagen vectorizada.

### ETIQUETAS PARA LA CREACIÓN DE TABLAS

* **<table> </table>** Etiquetas de apertura y cierre de una tabla. El resto de etiquetas de la tabla han de ir siempre recogidas entre estas dos etiquetas.
* **<caption> </caption>** Usada para indicar el título de la tabla.
* **<colgroup> </colgroup>** Etiqueta utilizada para agrupar dos o más columnas de una tabla.
* **<tbody> </tbody>** Usada para describir los datos concretos de una tabla.
* **<thead> </thead>** Indica el bloque de filas que describen las etiquetas de las columnas de la tabla.
* **<tfoot> </tfoot>** Indica los bloques de filas que describen los resúmenes, o datos totales de una columna de una tabla.
* **<tr> </tr>** Se usa para indicar una fila de celdas de una tabla.
* **<td> </td>** Usada para definir una celda de una tabla.
* **<th> </th>** Etiqueta que se usa para definir el encabezado de una celda

### ETIQUETAS PARA LA CREACIÓN DE FORMULARIOS

* **<form> </form>** Etiqueta de apertura y cierre de un formulario de una página web. El resto de etiquetas de formulario deben ir siempre recogidas entre estas etiquetas de apertura y cierre de formulario.
* **<fieldset> </fieldset>** Etiqueta que representa un conjunto o agrupación de elementos de un formulario. «Pinta» un recuadro alrededor de las etiquetas que estén contenidas dentro del <fieldset>
* **<legend> </legend>** Etiqueta ligada a <fieldset>. Indica el título del <fieldset>
* **<label> </label>** Se usa para definir el nombre o título de un control del formulario.
* **<input>** Pinta un campo de introducción de datos para el usuario. Es de las principales etiquetas de un formulario.
* **<button> </button>** Etiqueta utilizada para representar un botón en el formulario.
* **<select> </select>** Input que permite una selección entre un conjunto de opciones.
* **<option> </option>** Etiqueta ligada a <select>. Permite añadir diferentes opciones al <select>
* **<textarea> </textarea>** Añade un campo al usuario para que pueda introducir texto en unas líneas máximas de texto que el desarrollador puede definir.

<http://es.html.net/tutorials/css/>

<https://carontestudio.com/blog/atributos-html/>

http://html.ne