# Introducción

Antes de HTML, trabajábamos con ficheros de texto plano sin formato. HTML, es un lenguaje de marcas que nos permitirá elaborar páginas web. Actualmente nos encontramos ante la versión 5 de HTML o como también es conocida HTML5.

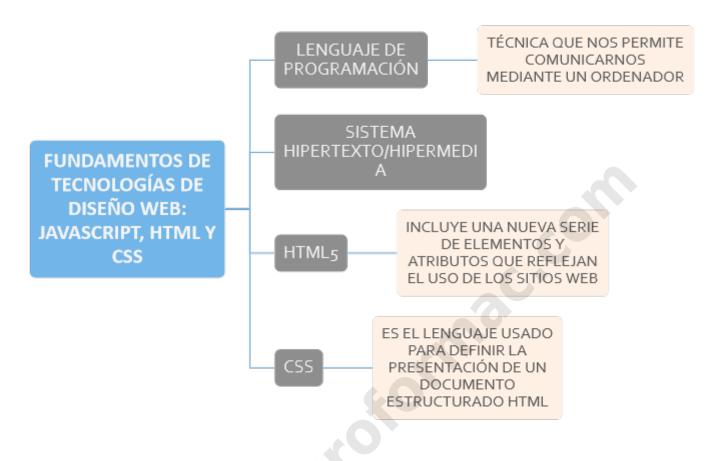
La evolución de estás versiones de HTML junto a la combinación de otras tecnologías (capas) han ido produciendo páginas webs más interactivas, más visuales, etc.

En la actualidad HTML se utiliza principalmente para otorgar una estructura básica y contenido a nuestra web. Posteriormente, lo combinaremos con otras capas (lenguajes) como CSS (mediante a hojas de estilo) que nos proporcionarán la parte estética y JavaScript (JS) que otorgará la inteligencia, el dinamismo, la acción, es decir, la funcionalidad.

# **Objetivos**

- Conocer la evolución de HTML.
- campus.euroformac.com

# **Mapa Conceptual**



# Introducción a lenguajes de programación Web

#### Introducción

Antes de adentrarnos en el mundo de la programación y la complejidad de los diversos lenguajes que existen debemos de conocer los conceptos más básicos.

Programación es el concepto que se utiliza para referirnos al proceso de diseño, creación, mantenimiento del código fuente de un programa computacional.

Un lenguaje de programación es una técnica que nos permite comunicarnos mediante una serie de expresiones con un ordenador.

El **código fuente** que se utiliza en la programación debe ser escrito en algún lenguaje de programación.

El ordenador es el encargado de traducir las expresiones que recibe a código máquina y así procesar una respuesta o solución.

Un lenguaje de programación web permite crear un documento web que puede ser visualizado por un usuario y entendido por un servidor.

Para entender mejor este concepto, vamos a ver cómo está formado la web y que sistema usa para crear los documentos que se encuentran en ella.

# Sistema hipertexto/hipermedia

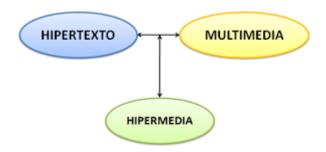
La web es un servicio que se encuentra en internet y en el que podemos encontrar todo tipo de páginas conectadas entre sí.

Está formada por un entramado de páginas de texto conectadas con enlaces, vínculos hacia otras páginas, construido con el sistema llamado hipertexto.

Los sistemas de hipertexto son muy sencillos de utilizar y con una gran utilidad.

La web no solo permite presentar páginas con textos e imágenes, también permite el contenido multimedia.

El termino hipermedia hace referencia al contenido de hipertexto que está formado con contenido multimedia.



Esquema hipermedia.

Dependiendo de la funcionalidad que tenga una página web, se pueden dividir en varios tipos: páginas web estáticas y dinámicas.

#### Tipos de páginas web

En la web existen dos tipos de páginas web, las páginas estáticas y dinámicas.

Las páginas estáticas son las páginas no presentan movimiento y no permiten una funcionalidad más allá de los enlaces hacia otras páginas.

Son más sencillas de crear y permiten ofrecer texto plano y contenido multimedia a los visitantes de la página web.

Las páginas dinámicas son otro tipo de páginas web mucho más versátiles que nos permiten interactuar con ellas.

A estas páginas web le podemos de dotar de funcionalidad mediante el uso de un lenguaje de programación.

Normalmente, estas páginas son más "atractivas" para el usuario, ya que le permite realizar acciones dentro de la misma página.

Nosotros nos centraremos en las páginas web dinámicas, ya que aprenderemos a dotar con funcionalidad a nuestra página web.

[AFO02874B] IFCD05 Desarrollo de Aplicaciones Móviles Java ME [MOD0246CS] IFCD05 Desarrollo de Aplicaciones Móviles Java ME [UDI15091V] FUNDAMENTOS DE TECNOLOGÍAS DE DISEÑO WEB: JAVASCRIPT, HTML Y CSS

Páginas web dinámicas

Como hemos explicado en el punto anterior, estas páginas web implementan alguna funcionalidad o

interactividad que las hace más atractivas.

Las ventajas más conocidas de las páginas web dinámicas son:

• Creación de aplicaciones dentro de la misma web.

• Posibilidad de dotar a nuestra página con un aspecto más profesional.

• Facilita la actualización de los datos de nuestra página web.

• Mayor sencillez para el usuario.

Para dotar de funcionalidad a este tipo de páginas, deberemos de usar algún tipo de lenguaje de

programación aparte del simple HTML.

Aun así, no debemos de olvidar el lenguaje HTML, ya que muchos lenguajes de programación de los

que hablaremos en esta unidad, se incluye embebido dentro de las etiquetas y código HTML.

Esquema modelo cliente/servidor.

El modelo cliente/servidor es una arquitectura distribuida que divide el trabajo en dos partes, el

cliente y el servidor.

Los clientes obtienen acceso a la información que está almacenada en el servidor.

Los servidores tienen que facilitar la información que pide el cliente de una forma transparente que

este pueda entender.

Las páginas web dinámicas existen en un entorno cliente/servidor donde se cumplen los siguientes

puntos:

• El cliente es el encargado de solicitar una página web.

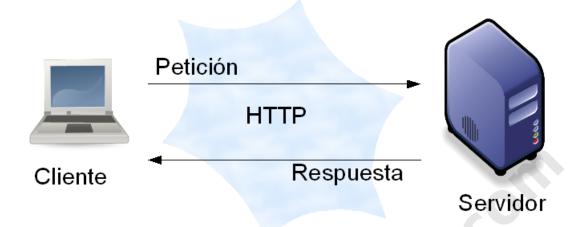
• El servidor se encarga de recibir las peticiones que hacen los clientes, procesar la petición y

devolver la página web construida al cliente.

• El cliente recibe la respuesta del servidor y muestra la página web al usuario.

campus.euroformac.com

6 / 50



# Esquema cliente/servidor

#### Lenguajes de programación web: cliente.

Los lenguajes de programación web del lado del cliente son aquellos lenguajes que son directamente asimilados por nuestro navegador haciendo que no necesiten un pre tratamiento.

El servidor no tiene que intervenir para construir la página web que solicita el usuario, se ejecuta en la aplicación cliente, principalmente en el navegador.

La programación web del lado del cliente nos permite construir páginas web interactivas, superando así las limitaciones de HTML.

Los lenguajes de programación del lado del cliente más conocidos son:

### • HTML/CSS.

HTML (Hyper Text Markup Language) es el lenguaje de programación más utilizado para escribir la estructura de las páginas web.

Tiene origen en el año 1980, cuando un trabajador de la CERN, propuso un sistema de hipertexto que nos permitiese compartir documentos.

Hubo muchos cambios, hasta que en 1995, se estandarizó y se puso en manos del organismo W3C que no ha dejado de publicar mejoras para este lenguaje.

Es un lenguaje basado en hipertexto, lo que nos permite escribir texto de forma estructurada y crear elementos con etiquetas.

HTML es un estándar reconocido en todo el mundo permitiendo así que se visualice de una forma muy parecía en cualquier navegador.

# Ejemplo de código HTML

Cuando se hizo el estándar HTML, no tenía ningún tipo de presentación, era texto sin formato, con las nuevas actualizaciones de este lenguaje se ha ido cambiado el aspecto por ejemplo la letra, colores, márgenes, imágenes...

Con estos nuevos cambios se ha creado un mecanismo que permite separar el contenido de la presentación.

Este mecanismo se llama CSS (cascading style sheets).

CSS nos permite desarrollar una presentación visual y así crear nuevas páginas con aspectos más "bonitos".

# Ejemplo de código CSS

• JavaScript.

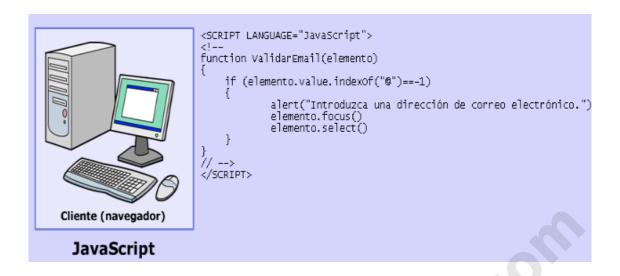
JavaScript es un lenguaje de programación que surgió a principio de los años 90.

En estos años, la velocidad de conexión de los usuarios que se conectaban a internet era muy lenta y las aplicaciones web iban evolucionando a aplicaciones más complejas lo que requería mucho tiempo de espera.

Fue creado por Netscape para solventar el problema de la época, creando un lenguaje de programación con la capacidad de poder ejecutarse directamente en el navegador.

JavaScript es un lenguaje de programación del lado del cliente que se utiliza principalmente para dotar a una página web de dinamismo.

Es un lenguaje de programación interpretado, esto quiere decir que no es necesario compilarlo para que pueda ser ejecutado. En otras palabras, el código escrito en JavaScript es interpretado por el navegador sin necesidad de procesos secundarios.



Ejemplo de código JavaScript

• VBScript.

VBScript es un lenguaje de programación creado por Microsoft.

Es un lenguaje interpretado solamente por el navegador de esta compañía, Internet Explorer.

Su sintaxis está basada en el popular lenguaje de programación de script que permite crear aplicaciones en el sistema operativo Windows, Visual Basic.

Es un competidor directo a JavaScript, pero su utilización es menor debido a que solo es compatible con un solo navegador.

```
<!DOCTYPE html>
<html>
    <head>
        <title>Ejemplo código VBScript</title>
    </head>
    <body>
        <script type="text/vbscript">
           dim e name(2)
            e name(0)="Mai"
            e name(1)="Kale"
            e name(2)="Lu"
            for i=0 to 2
               document.write(e name(i) & "<br />")
           next
        </script>
        <div>
            Ejemplo VBScript
        </div>
    </body>
</html>
```

Ejemplo de código VBScript.

Las páginas web que creamos con estos lenguajes de programación son enviadas a nuestro navegador. Este se encarga de interpretar el código y mostrar la página al usuario.

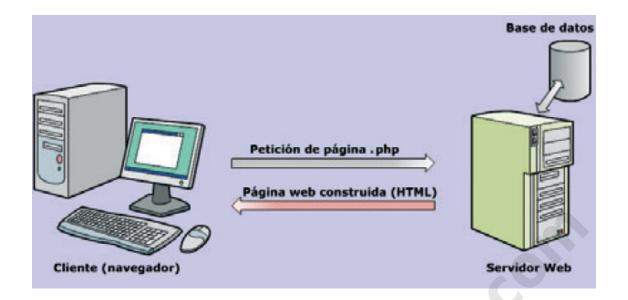
A la hora de elegir un lenguaje de programación del lado del cliente, hay que tener en cuenta que lo que funciona en un navegador web, puede ser que no funcione en el otro.

#### Lenguajes de programación web: servidor.

Los lenguajes de programación web del lado del servidor son aquellos lenguajes que son asimilados, reconocidos, interpretados y ejecutados en el propio servidor donde se encuentra la página web.

Este tipo de lenguaje nos permite crear aplicaciones web que puedan acceder a datos que se encuentren solamente en el servidor como consultas a una base de datos, usuarios, documentación, etc.

Las páginas web que utilizan lenguajes de programación del lado del servidor son enviadas al servidor antes de mostrar el resultado al usuario para que el servidor interprete este código y mande al cliente la página web comprensible para él.



Ejemplo lenguaje de programación (servidor).

Los lenguajes de programación del lado del servidor más conocidos son:

• PHP.

PHP es un lenguaje de programación del lado del servidor de código abierto creado para el desarrollo web de contenido dinámico.

Permite ser incrustado en el propio HTML el código PHP. Este código es interpretado y generado nuevamente por un servidor web compatible con esta tecnología.

Es compatible con la mayoría de sistemas operativos y servidores web que existen actualmente.

Aunque es un lenguaje de programación que incorpora una gran lista de características para nuestras aplicaciones, lo mejor es su simplicidad a la hora de implementar el código.

```
o
     <?php
2
     function Saludo() {
3 🖃
          $hoy = localtime(time(), 1);
4
 5
          $hora = $hoy[tm hour];
          if (($hora >= 7) && ($hora < 13)) {
 6
7
              return "Buenos días";
          } elseif (($hora >= 13) && ($hora < 20)) {</pre>
8
              return "Buenas tardes";
9
          } else {
10
11
              return "Buenas noches";
12
     }
13
     ?>
14
15 🖃
     <html>
16 🗀
          <head>
              <title>Mi primera página PHP</title>
17
              <meta http-equiv="Content-Type" content="text/html;</pre>
18 🗐
                    charset=iso-8859-1">
19
          </head>
20
          <body>
21 🗀
              <?php echo Saludo();
22
23
          </body>
24
     </html>
```

Ejemplo de código PHP.

• JSP.

JSP es una extensión del lenguaje de programación Java que nos permite crear páginas web dinámicas.

Para interpretar el código escrito con el lenguaje JSP es necesario tener instalado y funcionando un servidor web compatible con esta tecnología.

El servidor web se encarga de compilar el código JSP que le enviamos y lo procesa en tiempo real para devolver al cliente una página web interpretada que pueda ser capaz de visualizar.

```
<%
   String author = request.getParameter("author");
   if (author != null && !author.equals(""))) {
%>
        You have choose author <%= author %>
<%
   }
}</p>
```

Ejemplo de código JSP.

• ASP.

ASP es un lenguaje de programación creado por Microsoft que permite crear aplicaciones web del lado del servidor.

Permite generar páginas web mezclando el código HTML para crear la estructura y el código ASP para crear la funcionalidad.

La principal desventaja de este lenguaje es que solamente puede ser utilizado en el sistema operativo Windows pues necesita un servidor web IIS de Microsoft para interpretar el código.

El código resultante ASP puede ser visualizado en cualquier navegador que utilice el cliente.

Ejemplo de código ASP.

Llegará el momento en el que debamos decidirnos por una de estas tecnologías, pero hay que tener cuidado ya que de esto dependerá la base de datos utilizada, el servidor web y muchos otros factores. No todas las combinaciones son posibles.

# HTML5

HTML5 es la quinta versión del lenguaje de marcado para la elaboración de páginas web. Esta versión está todavía en desarrollo, aunque algunas empresas ya lo están usando para crear sus páginas Web.

HTML5 incluye una nueva serie de elementos y atributos que reflejan el uso convencional de los sitios web modernos. Algunos elementos tienen las mismas características técnicamente hablando que las etiquetas

 $\boldsymbol{y}$  , pero con un significado semántico. Por ejemplo

, para la cabecera,

para el pie de página. Otros elementos como y pretenden ofrecer nuevas funcionalidades a través de una interfaz estandarizada. También es importante las mejores en el elemento , que ahora es capaz de renderizar elementos 3D en algunos navegadores. A continuación vamos a ver con más detalle estos nuevos elementos.

Algunos elementos han pasado a estar obsoletos en esta nueva versión ya que se puede conseguir el mismo efecto con CSS3.

#### ¿Cuáles son las mejoras?

- Nuevas etiquetas para representar semánticamente el contenido de las páginas Web.
- Eliminación de etiquetas y atributos que ya no tienen sentido.
- Nuevos elementos para crear animaciones o reproducir vídeo y audio sin necesidad de utilizar ningún complemento externo al navegador.
- Etiquetas para manejar grandes conjuntos de datos: Datagrid, Details, Menu y Command.

  Permiten generar tablas dinámicas que pueden filtrar, ordenar y ocultar contenido.
- Formularios mejor estructurados y con funcionalidad para validar los datos nativamente.
   Nuevos tipos de datos (eMail, number, url, datetime...).
- Visores de fórmulas matemáticas (MathML) y gráficos vectoriales (SVG)
- Arrastrar objetos, Drag & Drop.

#### **Etiquetas Obsoletas**

Con la llegada de HTML5 y CSS3 hay varios elementos que se han quedado obsoletos, por lo tanto estos se han omitido, son:

- : usado para definir los acrónimos, se usa ahora
- : se usaba para meter applet embebidos, ahora se utiliza la etiqueta
- : Usado para definir el color, tamaño y tipo de letra para todo el documento. Actualmente se usa CSS.
- : Usado para hacer el texto más grande. Actualmente se usa CSS.
- : Usado para centrar el texto, actualmente se usa CSS.
- Usado para definir una lista de directorios. Ahora se usa
  - Especificaba el color, tamaño y fuente de un texto determinado, actualmente se usa
     CSS para cambiarlo.
  - y usados para definir una determinada ventana, se han dejado de usar debido a los efectos negativos a la hora de usarlo. : es usado para mostrar el texto por el navegador que no manejan los frames
  - : manda información al servidor.
  - : usado para definir texto tachado. Ahora se usa
  - : usado para definir el tipo de texto.

#### Etiquetas modificadas

Hay algunos elementos que han sido modificados, vamos a compararlos con la versión antigua de html.

• **Etiquetas :** antiguamente esta etiqueta podía ser un hipervínculo o un ancla, ahora la etiqueta siempre es un vínculo, si no lleva el atributo href, es solo un marcador de posición para un hipervínculo. Hay que tener cuidado porque muchos de sus atributos se han cambiado.

# • Etiqueta

: se puede usar varias veces en un mismo documento, indica una forma de contacto referida al documento, a la sección o a un artículo. Pasa a ser un nuevo concepto de sección.

• Etiqueta : representa un trozo de texto a ser estilizado sin ninguna importancia.

- **Etiqueta** : aparte del elemento al que puede hacer referencia también puede tener definido el formulario al que puede hacer referencia.
- **Etiqueta**: se usa para crear los menús actuales. En las versiones antiguas se desechó esta etiqueta, pero ahora se ha vuelto a reutilizar.
- **Etiqueta** : antes se definía para representar el texto más pequeño. Ahora sirve para definir texto pequeño y también para los comentarios secundarios.
- Etiqueta : en versiones antiguas definía el texto con énfasis, en la nueva versión lo define como un texto importante.

#### Lista de etiquetas HTML5

HTML provee de interoperabilidad y reduce los costos a la hora de desarrollar creando nuevas reglas más precisas que mejoran el manejo de elementos/etiquetas HTML y haciendo así que no se produzcan tantos errores.

Crea nuevas funciones como embeber audio, video, gráficos, nuevos almacenamientos de datos y documentos interactivos. A continuación vemos la lista de etiquetas nuevas que introduce HTML5 y de las etiquetas que mantiene de versiones anteriores

Etiqueta		Descripción
		Define los comentarios
		Define el tipo de documento
<a></a>		Define los enlaces
<abbr></abbr>		Define abreviaciones
<address></address>		Define información sobre el autor o creador del documento o articulo
<area/>		Define una imagen de mapas
<article></article>	¡NUEVA!	Define un artículo
<aside></aside>	¡NUEVA!	Define un contenido nuevo de la página
<audio></audio>	¡NUEVA!	Define un contenido de sonido
<b></b>		Define el texto en negrita
<base/>		Especifica la base URL de todas las URL del documento
<bdi>&gt;</bdi>	¡NUEVA!	Indica una parte del texto que esta formateada
<bdo></bdo>		Cambia la dirección del texto
<blookquote></blookquote>		Define una cita copiada de otro documento
<body></body>		Define el cuerpo del documento
		Define saltos de línea
<button></button>		Define un botón para hacer clic
<canvas></canvas>	¡NUEVA!	Usado para dibujar gráficos ( usando javascript)
<caption></caption>		Define un titulo de una tabla
<cite></cite>		Define un título de un trabajo
<code></code>		Define una pieza del código
<col/>		Especifica una columna de propiedades de un <colgroup></colgroup>
<colgroup></colgroup>		Especifica un grupo de una o más columnas de una tabla
<command/>	¡NUEVA!	Define un botón de comandos que el usuario invoca
<datalist></datalist>	¡NUEVA!	Especifica una lista predefinida de opciones para un input
<dd></dd>		Define una descripción de un elemento de una lista de definición

<del></del>		Define un texto que ha sido borrado de un documento
<details></details>	¡NUEVA!	Muestra detalles que el usuario puede ver u ocultar
<dfn></dfn>		Define una definición de un termino
<div></div>		Define una sección del documento
<dl></dl>		Define una definición de lista
<dd></dd>		Define un termino de una definición de lista
<em></em>		Define un texto enfatizado
<embed/>	¡NUEVA!	Define un contenedor que tendrá una aplicación externa o un contenido interactivo
<fieldset< td=""><td></td><td>Grupo de elementos de un formulario</td></fieldset<>		Grupo de elementos de un formulario
<figcaption></figcaption>	¡NUEVA!	Define el título de un elemento <figure></figure>
<figure></figure>	¡NUEVA!	Define un contenedor
<footer></footer>	¡NUEVA!	Define el pie de nuestro documento o de la sección
<h1>- <h6></h6></h1>		Define los títulos
<head></head>		Define la información de nuestro documento
<header></header>	¡NUEVA!	Define la cabecera de un documento o sección
<hgroup></hgroup>	¡NUEVA!	Grupo de títulos ( <h1<-<h6>)</h1<-<h6>
<hr/>		Define una línea
<html></html>		Define donde empieza el documento HTML
<i>&gt;</i>		Define una parte del texto debe ser destacado
<iframe></iframe>		Define un frame en línea
<img/>		Define una imagen
<input/>		Define un cuadro de texto
<ins></ins>		Define un texto que va a ser insertado en el documento
<keygen/>	¡NUEVA!	Define un campo con una clave generada (para formularios)
<kdb></kdb>		Define un input de teclado
<label></label>		Define una etiqueta para el input
<legend></legend>		Define el titulo de <fieldset>,<figure> y<details></details></figure></fieldset>
<li>&lt;</li>		Define un listado de términos
<li>k&gt;</li>		Define una relación con un documento externo
<map></map>		Define una imagen de mapas
<mark></mark>	¡NUEVA!	Define un nuevo marcado de texto
<menu></menu>		Define una lista de menu
<meta/>		Define el metadata del documento html
<meter></meter>	¡NUEVA!	Define un a medición de un rango

<nav></nav>	¡NUEVA!	Define el menu de navegación
<noscript></noscript>		Define un contenido que no soporta en el cliente
<object></object>		Define un objeto embebido
<ol></ol>		Define una lista ordenada
<optgroutp></optgroutp>		Define un grupo de relaciones de una lista
<option></option>		Define una opción de una lista desplegable
<ouput></ouput>	¡NUEVA!	Define un resultado de un calculo
		Define un párrafo
<param/>		Define un parámetro de un objeto
<pre></pre>		Define un texto preformateado
<pre><pre><pre><pre><pre><pre><pre><pre></pre></pre></pre></pre></pre></pre></pre></pre>	¡NUEVA!	Representa el progreso de un a tarea
< <b>q&gt;</b>		Define un cita corta
<rp></rp>	¡NUEVA!	Define texto por si el navegador no soporta <ruby></ruby>
<rt></rt>	¡NUEVA!	Define la explicación/ pronunciación de caracteres
<ruby></ruby>	¡NUEVA!	Define la anotación ruby ( para la tipografía asiática)
<s></s>		Define el texto que no es correcto, lo tacha
<script></td><td></td><td>define donde empieza el código javascript</td></tr><tr><td><section></td><td>¡NUEVA!</td><td>Define una sección del documento</td></tr><tr><td><select></td><td></td><td>Define una lista desplegable</td></tr><tr><td><source></td><td>¡NUEVA!</td><td>Define el destino del fichero multimedia que se desea cargar</td></tr><tr><td><span></td><td></td><td>Define una sección del documento</td></tr><tr><td><strong></td><td></td><td>Define la importancia del texto, pone en negrita</td></tr><tr><td><style></td><td></td><td>Define información sobre el estilo del documento</td></tr><tr><td><sub></td><td></td><td>Indica el subíndice del texto</td></tr><tr><td><summary></td><td>¡NUEVA!</td><td>Define una cabecera para el elemento details</td></tr><tr><td><sup></td><td></td><td>Define un superíndice</td></tr><tr><td></td><td></td><td>Define las tablas</td></tr><tr><td></td><td></td><td>Define el cuerpo de una tabla</td></tr><tr><td></td><td></td><td>Indica una celda de una tabla</td></tr><tr><td><textarea></td><td></td><td>Define un campo de múltiples líneas</td></tr><tr><td><tfoot></td><td></td><td>Define el pie de la tabla</td></tr><tr><td></td><td></td><td>Define la celda cabecera de la tabla</td></tr><tr><td><thead></td><td></td><td>Define la cabecera de la tabla</td></tr><tr><td><time></td><td>¡NUEVA!</td><td>Define una fecha</td></tr><tr><td></td><td></td><td></td></tr></tbody></table></script>		

<title>&lt;/th&gt;&lt;th&gt;&lt;/th&gt;&lt;th&gt;Define el titulo del documento&lt;/th&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;&lt;/td&gt;&lt;td&gt;&lt;/td&gt;&lt;td&gt;Define las filas de una tabla&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;&lt;u&gt;&gt;&lt;/td&gt;&lt;td&gt;&lt;/td&gt;&lt;td&gt;Define el texto que debe ser diferente al normal&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;&lt;ul&gt;&lt;li&gt;&lt;ul&gt;&lt;/li&gt;&lt;/ul&gt;&lt;/td&gt;&lt;td&gt;&lt;/td&gt;&lt;td&gt;Define una lista desordenada&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;&lt;video&gt;&lt;/td&gt;&lt;td&gt;¡NUEVA!&lt;/td&gt;&lt;td&gt;Define un video o película&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;&lt;wbr&gt;&lt;/td&gt;&lt;td&gt;¡NUEVA!&lt;/td&gt;&lt;td&gt;Define un espacio roto, seria lo contrario a&lt;/td&gt;&lt;/tr&gt;&lt;/tbody&gt;&lt;/table&gt;</title>
---

#### Atributos eliminados

Los atributos que hacen referencia a la presentación han sido eliminados, menos style que ahora sirve para indicar la presentación de cualquier elemento. La siguiente lista contiene todos los atributos que ya no se van a usar en HTML5:

- Atributo **align** en todos los elementos
- Atributo alink, link, text y vlink en el elemento body
- Atributo background en el elemento body
- Atributo **bgcolor** en los elementos table, tr, td, th y body
- Atributos **border** en todos los elementos
- Atributo **cellpadding** y **cellspacing** en el elemento table
- Atributos **char** y **charoff** en los elementos col, colgroup, tbody, td, tfoot, th, thead y tr.
- Atributo **clear** en el elemento br
- Atributo **compact** en los elementos dl, menu, ol y ul.
- Atributo **frame** en el elemento table
- Atributo frameborder en el elemento iframe
- Atributo **height** en los elementos th y td
- Atributo **hspace** y **vspace** en los elementos ima y object
- Atributos **marginheight** y **marginwidth** en el elemento iframe
- Atributo **noshade** en el elemento hr
- Atributo **nowrap** en los elementos td y th.
- Atributo **rule** en el elemento tabla
- Atributo **scrolling** en el elemento iframe

- Atributo **size** en el elemento hr
- Atributo type en los elementos li, ol y ul
- Atributo **vlagin** en los elementos col, colgroup, tbody, td, tfoot, th, thead y tr
- Atributo **width** en los elementos hr, table, td, th, col, colgroup y pre

Existen otros atributos que se quitan por redundancia, por evitar confusiones, por su bajo uso o porque simplemente se han quedado obsoletos.

- Atributos **rev** y **charset** en los elementos link y a
- Atributos **shape** y **coords** en el elemento a
- Atributo **longdesc** en los elementos ima y iframe
- Atributo target en el elemento link
- Atributo nohref en el elemento area
- Atributo **profile** en el elemento head
- Atributo **version** en el elemento html
- Atributo **name** en los elementos ima y a. Se recomienda usar id.
- Atributo scheme en el elemento meta
- Atributo archive, classid, codebase, codetype, declare y standby en el atributo object
- Atributos valuetype y type en el elemento param
- Atributos **axis** y **abbr** en los elementos td y th.
- Atributo **scope** en el elemento td.
- Atributo **summary** en el elemento table

#### Otros cambios

Hay elementos que no se eliminan por su elevado uso, como los elementos <u>e frente a y .</u>

Estos elementos indican negrita y cursiva respectivamente y pasan a definirse de una manera vaga para indicar un texto diferente de alguna manera al texto normal. Otro elemento es small, se mantiene, al contrario de big, que se elimina.

También es importante resaltar un cambio en la estructura de las tablas y es que en versiones antiguas de HTML, los elementos que organizaban las tablas debían ir en el

orden: , y . Ahora no es necesario que la cabecera y el píe estén definidos juntos, sino que puede estar definido en el orden lógico por el que se visualiza (cabecera, cuerpo y pie).

#### Estructura HTML5

El DOCTYPE de HTML5 es sencillo, breve y conciso, a diferencia de las otras versiones antiguas, se representa como:

<!DOCTYPE HTML>

Cuando se incluye esa línea en un fichero estamos diciendo al navegador que esa página está creada con HTML5. Si no se incluye esa línea, el navegador no interpretará el código fuente de forma adecuada.

Lo siguiente que encontraremos en nuestra Web son las típicas etiquetas html, head y body. Es importante definir la codificación de caracteres dentro de la cabecera, para evitar posibles desajustes.

```
<!DOCTYPE html>
<html>
<head>
<title>Primera página HTML5</title>
<meta http-equiv="Content-Type" content="text/html;
charset=utf-8">
</head>
<body>
</body>
</html>
```

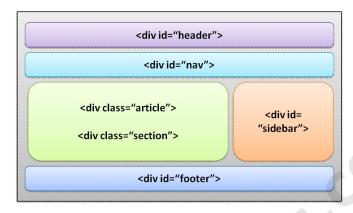
La estructura es prácticamente la misma, salvo el DOCTYPE, con esto ya tenemos lo básico de nuestro primer fichero HTML5. La extensión del fichero que contiene el código, es la misma de siempre: .html.

#### Nueva estructuración del body

Una diferencia bastante significativa de esta nueva versión es que añade nueva semántica y accesibilidad. HTML 5 implementa una nueva estructura, debido al aumento del uso de la etiqueta div para estructurar la Web en bloques. Esta versión nos proporciona varios

elementos que establecerán la estructuración principal, sin necesidad de utilizar div.

# HTML



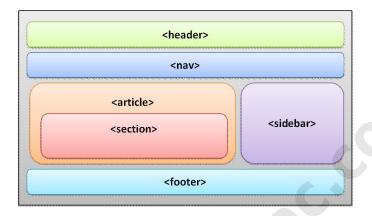
Este cambio en la semántica hace que sea más coherente y fácil de entender. Además brinda a los navegadores más facilidad a la hora de buscar y a la hora de dar más importancia a una determinada sección.

Las Web en HTML5, su body se dividirá en:

- : representa una sección general dentro de un documento o aplicación. Puede contener otra subsección y se acompaña normalmente de h1-h6 para facilitar la estructura, creando jerarquías.
- : representa un componente de la página que consiste en una composición autónoma de un documento, página, aplicación o sitio Web. Podría ser: mensaje en un foro, artículo de una revista o un periódico, entrada de Blog, comentario de usuario, widget interactivo o gadget...
- : representa una sección de la página que contiene normalmente un contenido independiente de lo que le rodea, aunque también puede estar relacionado. Por ejemplo: publicidad, barras laterales, efectos tipográficos, información del autor...
- : representa la sección de encabezado. Donde se suele encontrar el título del sitio Web, el logotipo o estructuras de navegación.
- : representa una sección de una página que es un link a otras páginas o a partes dentro de la misma.

• : representa el píe de página, con información acerca de la página, como el autor, el copyright o el año.





De esta manera, si tenemos una página html convencional...

```
<body>
<div id="cabecera">
   <!-- aqui iba el logotipo y las búsquedas-->
</div>
<div id="menu">
   <!-- aqui iba el menú de navegación>-->
</div>
<div id="contenedor">
    <div id="contenido">
        <!-- aqui iba el contenido de la página, articulos ....->
   </div≻
    <div id="lateral">
       <!-- aqui iba contenido adicional o publicidad-->
   </div>
</div>
<div id="pie">
   <!-- aqui iba quien había diseñado la web o copyright-->
</div>
</body>
```

En HTML5 quedaría de la siguiente manera.

```
<body>
<header>
   <!-- aqui iba el logotipo y las búsquedas-->
</header>
<nav≻
   <!-- aqui iba el menú de navegación>-->
<div id="contenedor">
   <article">
        <!-- aqui iba el contenido de la página, articulos ....->
    </article>
    <aside>
       <!-- aqui iba contenido adicional o publicidad-->
    </aside>
</div>
<footer>
   <!-- aqui iba quien había diseñado la web o copyright
</footer>
</body>
```

#### Otros elementos nuevos de estructuración

Además de los descritos arriba, también se han introducido otros elementos de la estructura como hgroup y figure. También se creó una etiqueta llamada dialog aunque esta finalmente quedo suprimida.

Sirve para agrupar un conjunto de uno o más elementos de encabezado (h1- h6), nos permite agrupar en un mismo bloque un título y un subtítulo. Antes podíamos poner los títulos sin problema, pero ahora se pueden usar estructuras para asignar prioridades de los elementos. Aunque esto solo son ejemplos, se podrían estructurar:

- H1 para el título de nuestra Web
- Div para el slogan de la cabecera
- H2 para el título de la noticia
- H3 para el subtítulo de la noticia
- H3 para el título de una sección con información adicional en la columna de la derecha.

En resumen esta nueva etiqueta englobaría títulos y subtítulos para ordenarlos y esquematizarlos.

Representa varios contenidos, opcionalmente con un título que lo identifica. Está relacionado con el flujo principal pero su posición es independiente a este.

Se suele tratar de una imagen, ilustración, diagrama, fragmento de código o un esquema al que se hace referencia en el texto principal. Se puede mover a otra página o a un apéndice sin que afecte al resto del documento.

```
<figure>
<figure>
<figure>
<figure>
<figure>
<figure>
<figure>
<figure>
<figure>
<figure-
</figure-
</pre>
```

Representa el título del elemento figure.

```
<figure>
'Twas brillig, and the slithy toves<br>
Did gyre and gimble in the wabe; <br>
All mimsy were the borogoves, <br>
And the mome raths outgrabe.
<figcaption><cite>Jabberwocky</cite> (first verse). Lewis Carroll, 1832-98</figcaption>
</figure>
```

# • CSS3

CSS es el lenguaje usado para definir la presentación de un documento estructurado HTML. CSS3 es la nueva versión de CSS. Esta versión es totalmente compatible con las anteriores. ¿Y qué aporta de nuevo CSS3?

- Nuevos selectores.
- Técnicas de diseño avanzado: esquinas redondeadas, gradientes, múltiples imágenes de fondo, etc.
- Color HSL y transparencia.
- Incrustación de fuentes y nuevo formato WOFF.
- Transiciones, transformaciones y animaciones

#### Unidades, colores y fuentes

Las unidades y colores no han cambiado mucho respecto a las antiguas versiones de CSS ya que se siguen definiendo igual, pero a continuación vamos a hacer un breve repaso para recordarlas.

#### Unidades

Aquí vemos una tabla con las unidades que se utilizan en CSS para las medidas:

Unidad	Descripción
%	Porcentaje
inch	Pulgada
cm	Centímetros
mm	Milímetros
em	1em es igual al tamaño de la letra actual. 2em es dos veces el tamaño de la letra.
ex	Un ex es el alto te la letra x de una fuente
pt	Puntos ( 1 pt es lo mismo que 1/72 inch)
рс	Pica( 1pc es igual a 12 pt)
рх	Pixel ( un punto en la pantalla del ordenador)

[AFO02874B] IFCD05 Desarrollo de Aplicaciones Móviles Java ME [MOD0246CS] IFCD05 Desarrollo de Aplicaciones Móviles Java ME [UDI15091V] FUNDAMENTOS DE TECNOLOGÍAS DE DISEÑO WEB: JAVASCRIPT, HTML Y CSS

**Colores** 

Los colores en CSS se definen a partir de los siguientes métodos:

• Colores en Hexadecimal: colores especificados a partir de #RRGGBB, R (red),

G(green) y B(blue), "background-color:#ff0000;"

• Colores RGB: se definen mediante rgb(red, green,blue) y pueden ser definidos

mediante (0%-100% o 0-255) background-color: rgb (255,0,0);

• Colores RGBA: Igual que el anterior, pero al final se le añade un número más,

con él indicamos el nivel de transparencia que deseamos, "0,0" totalmente

transparente – "1.0" totalmente opaco. background-color: rgba (255,0,0,0.5);

• Colores HSL: Significa matiz, saturación y luminosidad. El valor puede ser de

0-360 o también 0%-100%. background-color: hsl (120,65%,75%);

• Colores HSLA: Igual que el anterior pero con la diferencia de que tiene un

cuarto valor que es para la transparencia, puede tomar valores que estén entre

0.0 (totalmente transparente) y 1.0 (totalmente opaco). background-color: hsla

(120,65%,75%,0.3);

• Colores predefinidos. Aquí tenemos una lista con los nombres de los colores

en el navegador:

http://www.w3schools.com/cssref/css\_colornames.asp

**Fuentes** 

Las fuentes recordamos que se definían a partir de la propiedad font-family. Esta

propiedad define las fuentes por defecto para asegurar la compatibilidad entre los

navegadores. A continuación vamos a ver las fuentes más comunes organizadas por

familias:

Serif Fonts

• Georgia, serif

Ricardo:

¿Has encontrado a Lopez?

campus.euroformac.com

29 / 50

• "Palatino Linotype", "Book Antiqua", Palatino, serif

Ricardo:

¿Has encontrado a Lopez?

• "Times New Roman", Times, serif

Ricardo:

¿Has encontrado a Lopez?

# Sans-Serif Fonts

• Arial, Helvetica, sans-serif

Ricardo:

¿Has encontrado a Lopez?

• Arial Black, Gadget, sans-serif

#### Ricardo:

¿Has encontrado a Lopez?

• "Comic Sans MS", cursive, sans-serif

Ricardo:

¿Has encontrado a Lopez?

• Impact, Charcoal, sans-serif

Ricardo:

¿Has encontrado a Lopez?

• "Lucida Sans Unicode", "Lucida Grande", sans-serif

# Ricardo: ¿Has encontrado a Lopez?

• Tahoma, Geneva, sans-serif

Ricardo:

¿Has encontrado a Lopez?

• "Trebuchet MS", Helvetica, sans-serif

Ricardo:

¿Has encontrado a Lopez?

· Verdana, Geneva, sans-serif

Ricardo:

¿Has encontrado a Lopez?

# **Monospace Fonts**

• "Courier New", Courier, monospace

Ricardo:

¿Has encontrado a Lopez?

"Lucida Console", Monaco, monospace

Ricardo:

¿Has encontrado a Lopez?

Nuevos Selectores y pseudo-clases

[attribute^=value]

Selecciona todos los elementos "x" cuyo valor de atributo empiece por el valor indicado.

Por ejemplo la clase del div empiece por "test".

# [attribute\$=value]

Selecciona todos los elementos "x" cuyo valor del atributo acabe por el valor indicado.

```
<style type="text/css">
div[class^="test"]{
background: #ffff00;
div[class$="test"]{
background:grey;
</style>
</head>
<body>
    <div class="testnumerouno">
         Este es el div numero uno.
     <div class="segundonumerotest">
         Este es el segundo div
    </div>
      ( 🔷 ) 🔀 | localhost/pruebas/html/ejercicio2.ht
     🚱 톥 횓 Más visitados 🔀 El primer docume
      Este es el div numero uno.
      Este es el segundo div
```

# [attribute\*=value]

Selecciona todos los elementos "x" cuyo valor del atributo contenga el valor indicado.

```
div[class*="test"]
{
font-size:20px;
}

② S ② Más visitados ② El primer documen

Este es el div numero uno.
Este es el segundo div
```

## :first-of-type

Selecciona el primer elemento cuyo selector hayamos declarado.

# :last-of-type

Selecciona el último elemento cuyo selector hayamos declarado.

```
Este es p1 el div numero uno.

Este es p2 el div numero uno.

Este es p1 el segundo div

Este es p2 el segundo div

Este es p3 el segundo div
```

# :only-of-type

Especifica el estilo que se le aplicará a todos los elementos (etiquetas), que sean los únicos de su tipo.



## :only-child

Especifica el estilo que se le aplicará a todos los elementos (etiquetas), que son hijos únicos de su padre.

```
p:only-child{
font-family:'Arial Black', Gadget, sans-serif;
</style>
</head>
<body>
   <div class="nivelcero">
       Kste es el nivel 0
   </div>
   <div class="testnumerouno">
       Este es p1 el div numero uno.
       Este es p2 el div numero uno.
   </div>
   <div class="segundonumerotest">
       Este es p1 el segundo div
       Este es p2 el segundo div
       Kste es p3 el segundo div
    </div>
```

# Este es el nivel 0

Este es p1 el div numero uno.

Este es p2 el div numero uno.

Este es p1 el segundo div

Este es p2 el segundo div

Fete es n3 el semindo div

# :nth-child(n)

Selecciona el elemento (etiqueta) que sea el hijo número "n" de su padre.

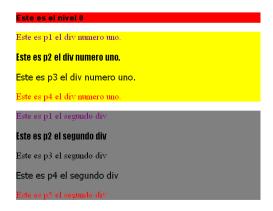
```
p:nth-child(2){
font-family:Impact, Charcoal, sans-serif;
}
</style>
</head>
<body>
   <div class="nivelcero">
       Este es el nivel 0
   </div>
   <div class="testnumerouno">
       <hl>prueba</hl>
      Este es p1 el div numero uno.
       Este es p2 el div numero uno.
   <div class="segundonumerotest">
       Este es p1 el segundo div
      Este es p2 el segundo div
       Este es p3 el segundo div
   </div>
```



#### :nth-last-child(n)

Selecciona el elemento (etiqueta) que sea el hijo número "n" de su padre empezando por el final.

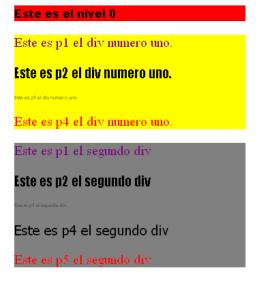
```
p:nth-last-child(2){
font-family:Tahoma, Geneva, sans-serif;
</style>
</head>
<body>
   <div class="nivelcero">
       Este es el nivel 0
   </div>
   <div class="testnumerouno">
       Este es p1 el div numero uno.
       Este es p2 el div numero uno.
      Este es p3 el div numero uno.
       Kste es p4 el div numero uno.
       </div>
   <div class="segundonumerotest">
       Este es p1 el segundo div
       Kste es p2 el segundo div
       Este es p3 el segundo div
     Este es p4 el segundo div
       Este es p5 el segundo div
   </div>
```



#### :nth-of-type(n)

Este selector aplica un estilo al hijo del tipo especificado número n de su padre. Por ejemplo p:nth-of-type(3) afectará a los párrafos que sean tercer hijo de su padre.

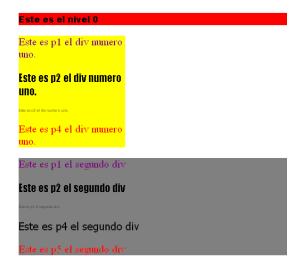
```
p:nth-of-type(3){
font-size:7px;
</style>
</head>
<body>
   <div class="nivelcero">
       Este es el nivel 0
   </div>
   <div class="testnumerouno">
       Este es p1 el div numero uno.
       Este es p2 el div numero uno.
      ≤p>Este es p3 el div numero uno.
       Este es p4 el div numero uno.
       </div>
    <div class="segundonumerotest">
       Este es p1 el segundo div
       Este es p2 el segundo div
      rste es p3 el segundo div
       Este es p4 el segundo div
       Este es p5 el segundo div
   </div>
```



#### :nth-last-of-type(n)

Este selector aplica un estilo al hijo del tipo especificado numero n de su padre, pero empezando por el final. Por ejemplo div:nth-of-type(2) afectará a los div que sean el segundo hijo empezando por el final, en este caso, el penúltimo.

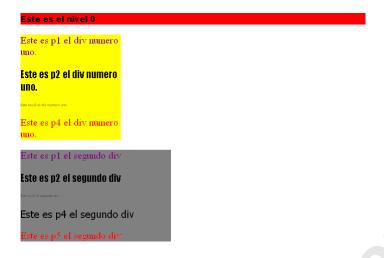
```
div:nth-last-of-type(2){
width: 200px;
}
</style>
</head>
<body>
   <div class="nivelcero">
       Kste es el nivel 0
   </div≥
   <div class="testnumerouno">
       Este es p1 el div numero uno.
       Este es p2 el div numero uno.
       Kste es p3 el div numero uno.
       Este es p4 el div numero uno.
       </div>
   <div class="segundonumerotest">
       Este es p1 el segundo div
       Este es p2 el segundo div
       Este es p3 el segundo div
       Este es p4 el segundo div
       Este es p5 el segundo div
   </div>
```



#### :last-child

Selecciona el último "hijo" del elemento que seleccionemos, en nuestro caso seleccionará el último div del body. Google Chrome no admite esta pseudo-clase.

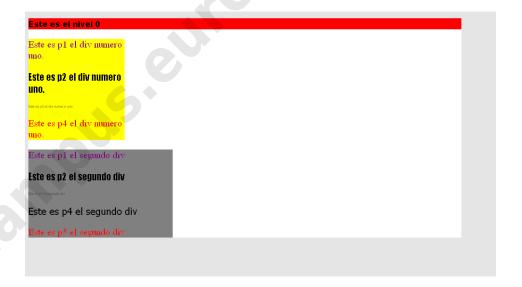
```
div:last-child
{
width:300px;
</style>
</head>
<body>
   <div class="nivelcero">
       Este es el nivel 0
   </div>
   <div class="testnumerouno">
       Este es p1 el div numero uno.
       Este es p2 el div numero uno.
       Este es p3 el div numero uno.
       Este es p4 el div numero uno.
       </div>
   <div class="segundonumerotest">
       Este es p1 el segundo div
       Este es p2 el segundo div
       Este es p3 el segundo div
       Este es p4 el segundo div
       Este es p5 el segundo div
   </div
</body>
```



#### :root

Aplica el formato a todo el documento, parecido a un selector sobre la etiqueta, pero en este caso se engloba todo el documento.

:root{background:#e5e5e5;}
body{background:white; width:900px;}



#### :empty

Específica el estilo que se le aplicará a un elemento (etiqueta) vacía.

```
p:empty{
   background-color:orange;
   height:10px;
</style>
</head>
<body>
   <div class="nivelcero">
       Este es el nivel 0
   </div>
   <div class="testnumerouno">
       Este es p1 el div numero uno.
       Este es p2 el div numero uno.
       <p≻Este es p3 el div numero uno.</p>
       </div>
   <div class="segundonumerotest">
       Este es p1 el segundo div
       Este es p2 el segundo div
       Este es p3 el segundo div
       Este es p4 el segundo div
       </div>
```



#### :target

Esta pseudo-clase sirve para referirnos al elemento destino de un enlace o ancla. En el ejemplo, esta propiedad nos va a servir para insertar una imagen delante del enlace.

```
<style type="text/css">
:target:before
{
content: url(pp.png);
}
</style>
</head>
</head>
</body>
</hl>

<hl>Bienvenidos</hl>

<a href="#cambio1">Cambio 1</a>
<a href="#cambio2">Cambio 2 </a>

<a name="cambio1"></a><b>cambio realizado 1...
<a name="cambio2"></a><b>cambio realizado 2...

<a name="cambio2"></a><b/p>
```

# Bienvenidos

## Bienvenidos



#### :enabled

Sirve para cambiar el estilo de un campo de texto cuando este está habilitado.

```
<style type="text/css">
input[type="text"]:enabled
{
background:orange;
}
</style>
</style>
</head>
<body>
<form action="">
Nombre: <input type="text" value="Nombre" /><br />
Rpellidos: <input type="text" value="Rpellidos" /><br />
</form>
```

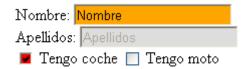
Nombre: Nombre
Apellidos: Apellidos

#### :disabled

Esta pseudo-clase sirve para cambiar el formato de una etiqueta que esté deshabilitada. Casi siempre se usa con elementos de formulario.

#### :checked

Sirve para cambiar el formato (estilo) de una casilla que está marcada. Aplicable a los elementos input tipo radio, button o checkboxes. Esta pseudo-clase de momento solo es soportada por el navegador Opera.



```
input:checked
{
background:#ff0000;
}
</style>
</head>
</head>
</hody>
<form action="">
Nombre: <input type="text" value="Nombre" /><br />
Rpellidos: <input type="text" disabled="disabled" value="Rpellidos" /><br />
<input type="checkbox" checked="checked" value="coche" /> Tengo coche
<input type="checkbox" value="moto" /> Tengo moto
```

#### :not(selector)

Aplica el estilo que definamos a los elementos que no coincidan con lo especificado. Es decir si indicamos ":not(p)", se aplicara el estilo a todos los elementos que no sean párrafos.

```
p{
    color:green;
}
:not(p){
    color:orange;
}
```

# primeros parrafos

```
este es el parrafo primero
este es el parrafo segundo
este es el parrafo tercero
```

#### enlace

Vamos a google?

#### div!

esto es un div

#### ::selection

Esta pseudo-clase se usa para editar el estilo de los elementos cuando son seleccionados

con el ratón. Por defecto si no indicamos nada, la selección es parecida a esto.

<mark>e</mark>ste es el parrafo segundo

Al modificarlo podemos conseguir prácticamente lo que nos propongamos.

```
::selection
{
color:#ff0000;
}
::-moz-selection
{
color:#ff0000;
}
```

# primeros parrafos

este es el parrafo primero este es el parrafo segundo

La pseudo-clase "::selection" es compatible con todos los navegadores a excepción de Mozilla Firefox. Para hacerla funcionar necesitaremos indicarla así: "::-moz-selection". De esta manera si funcionará sin problemas.

En resumen, todas estas propiedades son soportadas por los navegadores **Google Chrome**, **Firefox**, **Safari** y **Opera**. La pseudo-clase ":checked" solo es soportada por

Opera e Internet Explorer sólo es capaz de soportar los selectores ([attribute^=value] y

[attribute\$=value]).

### Javascript

Como ya sabemos, la estructura de un HTML es del tipo:

```
<HTML>

<HEAD>

</HEAD>

<BODY>

</BODY>

</HTML>
```

Pues bien, nosotros podremos introducir el código de JavaScript o en la cabecera (**HEAD**) o en el cuerpo (**BODY**) depende de la función que le vayamos a dar.

Este código de JavaScript se encierra siempre entre y tiene la estructura:

```
<html>
    <head>
3
   <!--
                  codigo
8
    </script>
9
10
11
    </head>
12
   ⊟<body>
13
14
15
16
    </body>
17
     </html>
18
```

Donde podremos observar varias cosas:

• El código se puede escribir en varios lenguajes, por tanto, hay que indicar al SCRIPT en qué tipo de lenguaje lo vamos a escribir, esto se hace mediante

type= "text/javascript", antiguamente se utilizaba language = "JavaScript", pero hoy en día, ha quedado en desuso.

- Por otra parte, debemos ocultar el código a los navegadores que no sepan interpretarlo (en la actualidad, todos los navegadores interpretan JavaScript), y para ello se utilizan los comentarios (comentario de inicio "").
- Para indicar un comentario de una sola línea, JavaScript utiliza "//".

Veamos qué pasaría si el navegador no reconociera el código JavaScript:

• Como no reconoce la etiqueta , pasa a la línea siguiente.

Ahora veamos qué pasaría con un navegador que sí lo reconociera:

• Reconoce la etiqueta, de forma que sabe que lo siguiente ya no es JavaScript.

Una cosa a tener en cuenta es que JavaScript a diferencia de HTML, distingue entre minúsculas y mayúsculas. Tenga cuidado con ello.

#### Contenido Alternativo

Aunque en la actualidad, todos los navegadores están preparados para ejecutar JavaScript, podríamos crear código alternativo para aquellos navegadores que no lo estuviesen.

¿Cómo?, muy fácil:

- Incluya la etiqueta tras la etiqueta de fin de guión.
- Incluya el código **HTML** que desee para esta sección de la página.
- Finalice con .

Cuando un navegador interpreta la etiqueta

#### Recuerda

- Las páginas estáticas son las páginas no presentan movimiento y no permiten una funcionalidad más allá de los enlaces hacia otras páginas.
- Las páginas dinámicas son otro tipo de páginas web mucho más versátiles que nos permiten interactuar con ellas.
- Los lenguajes de programación web del lado del servidor son aquellos lenguajes que son asimilados, reconocidos, interpretados y ejecutados en el propio servidor donde se encuentra la página web.
- HTML5 es la quinta versión del lenguaje de marcado para la elaboración de páginas web. Esta versión está todavía en desarrollo, aunque algunas empresas ya lo están usando para crear sus páginas Web.
- CSS es el lenguaje usado para definir la presentación de un documento estructurado HTMI

# Autoevaluación

1. Indica si la siguiente afirmación es verdadera o falsa : "La web es un servicio que se encuentra en internet y en el que podemos encontrar todo tipo de páginas conectadas entre sí"			
	Verdadero.		
	Falso.		
pág	inas estáticas son	nte afirmación es verdade n otro tipo de páginas web eractuar con ellas".	-
	Verdadero.		
	Falso.		
(Ну	per Text Markup l	nte afirmación es verdade Language) es el lenguaje ir la estructura de las pág	de programación más
	Verdadero.		

	Falso.
<b>4. E</b> l	l modelo cliente/servidor
	es una arquitectura distribuida que divide el trabajo en dos partes, el cliente y el servidor
	no es una arquitectura distribuida que divide el trabajo en dos partes, el cliente y el servidor
	es una arquitectura distribuida que une el trabajo en una partea, el cliente y el servidor
leng leng	ndica si la siguiente afirmación es verdadera o falsa : "Los quajes de programación web del lado del servidor son aquellos quajes que son asimilados, reconocidos, interpretados y utados en el propio servidor donde se encuentra la página web."
	Verdadero.