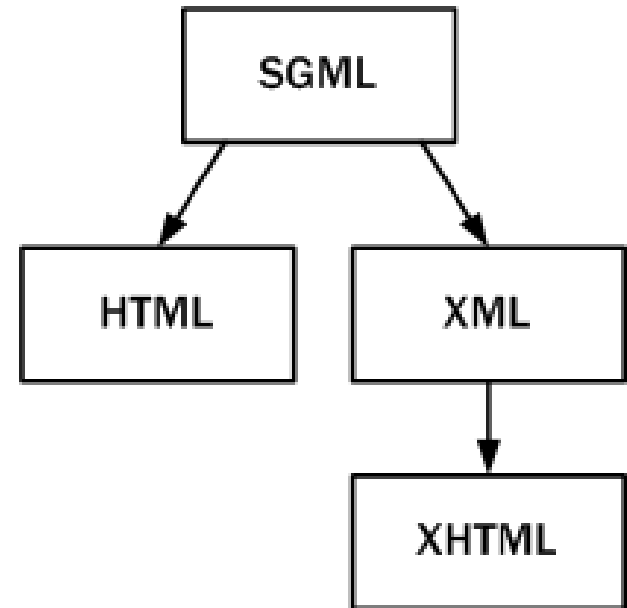


# HTML 4 y 5

Antonio Espín Herranz

# Introducción a HTML

- **HTML** (HyperText Markup Language) es un lenguaje muy sencillo que permite describir hipertexto, es decir, texto presentado de forma estructurada y agradable, con *enlaces* (hyperlinks) que conducen a otros documentos o fuentes de información relacionadas, y con *inserciones* multimedia (gráficos, sonido...)



- Especificación oficial **HTML 4**:
  - <http://www.w3.org/TR/html401/>
- Especificación oficial **HTML 5**:
  - <http://www.w3.org/TR/html5/>

# Estructura de un fichero HTML

```
<!DOCTYPE ...>
```

```
<html>
```

```
<head>
```

```
  <title> El titulo de la pagina </title>
```

```
  <!-- Metainformación -->
```

```
</head>
```

```
<body>
```

```
  <!-- Contenidos de la página -->
```

```
</body>
```

```
</html>
```

# Estructura de un fichero HTML

- El **DocType**, conviene ponerlo ayuda al navegador a pintar la página mas rápida.
- Se utiliza para validar la estructura de la página y comprobar si la sintaxis es correcta para el doctype que hemos elegido.
- Se valida contra una **DTD**. De la especificación oficial.
- Podemos utilizar un validador en la red para comprobarlo:

<http://validator.w3.org/>

# DocType

New HTML File (4.01 frameset)	html 4.01 frameset
New HTML File (4.01 strict)	html 4.01 strict
New HTML File (4.01 transitional)	html 4.01 transitional
New HTML File (5)	html 5
New XHTML File (1.0 frameset)	xhtml 1.0 frameset
New XHTML File (1.0 strict)	xhtml 1.0 strict
New XHTML File (1.0 transitional)	xhtml 1.0 transitional

- Cabeceras distintas que nos podemos encontrar en una página. Para indicar el tipo de documento.
- XHTML → XML + HTML
- Tanto para HTML y XHTML tenemos (3 posibilidades):

**frameset**

**strict**

**transitional**

# DocType

- **transitional**
  - El documento utiliza un abanico más amplio de etiquetas HTML y atributos, incluso algunos que puedan haberse quedado obsoletos.
- **strict**
  - Indica que el código es HTML o XHTML estricto, sin incluir atributos y etiquetas en deshuso.
- **frameset**
  - Se utiliza cuando estamos realizando una declaración de frames o frameset.

# DocType

- **Doctype XHTML 1.0 strict**

- `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">`

- **Doctype XHTML 1.0 transitional**

- `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">`

- **Doctype XHTML 1.0 frameset**

- `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">`

- **Doctype HTML 4.01 strict**

- `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">`

- **Doctype HTML 4.01 transitional**

- `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">`

- **Doctype HTML 4.01 frameset**

- `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd">`

**FRAMES YA NO SOPORTADOS EN HTML5** <sup>7</sup>

# DocType

- En el caso de HTML5 se simplifica bastante y se reduce a una única etiqueta:
- **<!DOCTYPE html>**



# Estructura de un fichero HTML

- El head, está destinado a la metainformación, se compone:
  - Datos como el autor, fecha de la página.
  - **Metainformación** para los robots de búsqueda.
  - Se pueden indicar estilos **CSS**, que normalmente se hará una referencia al fichero de estilos.
  - Y declaraciones de código en **JavaScript**.

# Ejemplo (HTML4)

```
<!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">
<head>
  <title> TITULO DE LA PÁGINA</title>
  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
  <meta name="keywords" content="lista de palabras clave" />
  <meta name="description" content="Descripción del Sitio" />

  <!-- HOJAS DE ESTILO y JAVASCRIPT - - >
  <link rel="stylesheet" type="text/css" href="util/style.css" media="screen" />
  <script type="text/javascript" src="util/capas.js"></script>
</head>
<body>
....
</body>
</html>
```

# Ejemplo (HTML5)

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
  <head>
```

```
    <meta charset="iso-8859-1">
```

```
    <meta name="keywords" content="....">
```

```
    <title>Título de la Web</title>
```

```
    <link rel="stylesheet" href="css/estilos.css">
```

```
  </head>
```

```
  <body>
```

```
</body>
```

```
</html>
```

Las etiquetas **meta**, **link** no es necesario Cerrarlas.

# Tipos de etiquetas

- Es importante saber que hay dos tipos de etiquetas: **inline** y **block** y el tratamiento que les da el navegador a estas.
- **inline**: El navegador las pinta en fila india, una detrás de otra. Cuando completa una fila pasa a la siguiente. Ejemplos: a, input.
- **block**: Cada etiqueta se pinta en una nueva línea. Ejemplos: div, form, p, h1 ... h6

# Etiquetas

- La etiquetas (o Tags) se componen de un nombre, encerrado de `<>` donde pueden aparecer atributos (encerrados **entre comillas**).
  - `<etiqueta>`
  - `<etiqueta atributo1="valor1">`
  - Por ejemplo: `<p>`, `<h1>`
    - `<p class="rojo">`
- Casi todas las etiquetas tienen que cerrarse y otras no es necesario:  

`<p>texto</p>``<br />`
- Las etiquetas se van anidando de acuerdo a su especificación (su DTD)
- En los **atributos** se pueden indicar también eventos de javascript, estilos, tamaños (por ejemplo en imágenes).

# Etiquetas

- El estilo del código:
  - En el caso de **XHTML** es case-sensitive pero **HTML4** y **5** no es case-sensitive.
  - Por convención se suele escribir en minúsculas.
  - En el caso del DOCTYPE se mantiene así:  
`<!DOCTYPE html>`

# Etiquetas

- 91 Etiquetas definidas en HTML:
  - a, abbr, acronym, address, applet, area, b, base, basefont, bdo, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, dd, del, dfn, dir, div, dl, dt, em, fieldset, font, form, frame, frameset, h1, h2, h3, h4, h5, h6, head, hr, html, i, iframe, img, input, ins, isindex, kbd, label, legend, li, link, map, menu, meta, noframes, noscript, object, ol, optgroup, option, p, param, pre, q, s, samp, script, select, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var.
- Etiquetas obsoletas:
  - applet, basefont, center, dir, font, isindex, menu, s, strike, u

# Clasificación Etiquetas

- **Básicas:** Estructura principal de la página, párrafos, encabezados.
- **Enlaces**
- **Listas:** ordenadas, desordenadas, de definición.
- **Formatear:**
  - Tamaños de texto, negrita, cursiva. **Se complementan sobretodo con CSS.**
- **Tablas**
- **Imágenes**
- **Formularios** y campos de entrada: pantallas de usuario.
- **Frames:** En HTML5 solo disponible **iframe**.
- **Estilos** y etiquetas **semánticas:**
  - La nuevas etiquetas de HTML5.
- **Audio y Video.**
- Etiquetas **Meta.**
- **Programación:** JavaScript, object.



# Etiquetas básicas

- `<!DOCTYPE>` El tipo del documento (ya comentado).
- `<html>` Inicio del documento HTML
- `<title>` Título de la página. Va dentro del **head**.
- `<body>` El cuerpo del documento.
- `<h1>` to `<h6>` Encabezados.
- `<p>` Párrafos.
- `<br>` Salto de línea.
- `<hr>` Separador horizontal. Línea
- `<!--...-->` Comentarios

# Etiquetas básicas

- Párrafos de texto (**<p> ... </p>**)

- Permiten estructurar el contenido, uniendo frases relacionadas entre sí.

- Saltos de línea (**<br />**)

- Se utilizan cuando se desea forzar un salto de línea; se recomienda que no se usen de forma indiscriminada y estructurar el contenido en párrafos.

- Títulos (**<hX>...</hX>**)

- Permiten fijar títulos para distintas secciones del documento; para ello se ofrecen las marcas <h1> hasta <h6>, siendo <h1> la marca de más alto nivel.  
<h2>Subtítulo</h2>

- Separadores horizontales (**<hr />**)

- Permiten crear una línea horizontal, prácticamente no se usan ya que se pueden conseguir efectos muy atractivos con las CSS.

# Etiquetas básicas

- **<html>**
  - Representa la raíz del documento y va inmediatamente después de la etiqueta **DOCTYPE**.
  - Se puede especificar el atributo **lang** con el idioma.
  - `<html lang="es">`

# Etiquetas básicas (Ejemplo)

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="iso-8859-1">
<title>Título de la página XHTML</title>
</head>
<body>
<h1>Este es un título de cabecera de primer nivel</h1>
<p> Algo de contenido para el documento. Este documento no será válido
al menos que BODY contenga un párrafo de texto u otro elemento hijo</p>
<p> En este segundo párrafo se está utilizando la marca &lt;br /&gt;
para separar líneas. <br /> Es algo no demasiado recomendable y de lo
que no se ha de abusar. </p>
<p>Si se desea citar, hemos de utilizar la marca &lt;cite&gt;;, por ejemplo:
<cite>esto es una cita a Cervantes</cite></p>
</body>
</html>
```

# Etiquetas para enlaces

- **<a>**
  - Se utiliza para definir enlaces a otras páginas: externas o internas al sitio web.
- **<link>** Para definir enlaces a hojas de estilo.
- **<nav>**
  - Barra de navegación de la página. Se comenta dentro de las etiquetas semánticas.
  - Su cometido es colocar en ella el menú de navegación.

# Enlaces

- `<a href="...">Texto</a>`
- Con los enlaces podemos referenciar páginas dentro de nuestra aplicación (utilizar rutas relativas), podemos enviar correos (teniendo instalado un cliente de correo) y acceder a URL externas (en este caso utilizar el protocolo: http:
- **Ejemplos:**
  - `<a href="otraPagina.html">Texto</a>`
  - `<a href="mailto:webmaster@hotmail.es">Texto</a>`
  - `<a href="http://www.google.es"> Buscador</a>`

# Enlaces

- Los enlaces también se pueden especificar dentro del propio documento.
- Para realizar saltos a otras secciones más abajo o más arriba.
- Útil cuando el documento es muy grande.

# Enlaces

- Primero indicar las zonas destino:

```
<a name="arriba"></a>
```

```
<a name="abajo"></a>
```

- Y posteriormente los enlaces:

```
<a href="#arriba" title="Ir Arriba">Ir arriba</a>
```

```
<a href="#abajo" title="Ir Abajo">Ir abajo</a>
```



# link

- Para referenciar las hojas de estilo:

<head>

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

```
<link rel="stylesheet" type="text/css" href="print.css" media="print">
```

</head>

- El atributo **media** por defecto toma el valor **all**, pero se pueden indicar otros como screen, print (son los más habituales).
- [http://www.w3schools.com/tags/att\\_link\\_media.asp](http://www.w3schools.com/tags/att_link_media.asp)

# link

- También nos sirve para colocar un icono dentro de la pestaña del navegador:
- Dentro del **head**:
  - `<link rel="icon" href="ruta al icono" type="image/gif" size="16x16">`



# Etiquetas para Listas

- **<ul>** Define una lista desordenada: viñetas.
- **<ol>** Define una lista ordenada: numerada.
- **<li>** Define un elemento de una lista: ol / ul.
- **<dl>** Define una lista de términos.
- **<dt>** Define un término dentro de una lista.
- **<dd>** Define la descripción del término dentro de la lista.
- **<menu>** Define una lista de opciones de menú.
- **<menuitem>** Define una opción de menú dentro de la etiqueta menú.

# Listas

Existen tres tipos de listas: *Ordenadas*, *sin ordenar* y *listas de definición*.

– Sea cual sea el tipo de lista, no es necesario que aparezcan dentro de los párrafos de texto.

## – Listas ordenadas

- Muestran sus elementos numerados de forma numérica.
- La marca principal es **<ol>**, utilizando **<li>** para cada ítem.

```
<p>Mis grupos musicales preferidos  
son:</p>  
<ol>  
<li>Pizzicato Five</li>  
<li>Tino Casal</li>  
<li>Mecano</li>  
</ol>
```

Mis grupos musicales preferidos son:

1. Pizzicato Five
2. Tino Casal
3. Mecano

# Listas

## Listas no ordenadas

- No se enumeran los elementos.
- Se utiliza la marca `<ul>` en lugar de `<ol>`.

## – Listas de definición

- En ellas, cada ítem se encuentra compuesto por dos elementos, un *término* (`<dt>`) y una *definición del mismo* (`<dd>`). La marca principal es `<dl>`:

```
<p>Mis grupos musicales preferidos son:</p>
<ul>
<li>Pizzicato Five</li>
<li>Tino Casal</li>
<li>Mecano</li>
<p>Mis grupos musicales preferidos son:</p>
<dl>
<dt>Pizzicato Five</dt>
<dd>Se trata de un grupo Japonés, que se hizo popular a mediados de los años 80</dd>
<dt>Tino Casal</dt>
<dd>José Celestino Casal Álvarez, más conocido como Tino Casal, fue un cantante y
compositor clave de la Movida madrileña de los años 1980.</dd>
</dl>
```

Mis grupos musicales preferidos son:

- Pizzicato Five
- Tino Casal
- Mecano

# menu / menuitem

- **La etiqueta menú se puede anidar para tener submenús:**

```
<menu type="context" id="mymenu">
  <menuitem label="Refresh" onclick="window.location.reload();"
    icon="ico_reload.png"></menuitem>
  <menu label="Share on...">
    <menuitem label="Twitter" icon="ico_twitter.png"
      onclick="window.open('//twitter.com/intent/tweet?text=' +
        window.location.href);">
    </menuitem>
    <menuitem label="Facebook" icon="ico_facebook.png"
      onclick="window.open('//facebook.com/sharer/sharer.php?u=' +
        window.location.href);">
    </menuitem>
  </menu>
  <menuitem label="Email This Page"
    onclick="window.location='mailto:?body='+window.location.href;"></menuite
    m>
</menu>
```

- **Ejemplo (solo para Firefox y menú popup)**

– [http://www.w3schools.com/tags/tryit.asp?filename=tryhtml5\\_menuitem](http://www.w3schools.com/tags/tryit.asp?filename=tryhtml5_menuitem)

# Formatear

- **<address>** Definir información de un artículo o de un documento.
- **<b>** Definir texto en negrita.
- **<cite>** Para definir el título de un libro o de un trabajo.
- **<del>** Definir un texto que se ha eliminado de un documento.
- **<em>** Para dar énfasis a un texto.
- **<i>** Texto con letra itálica.
- **<ins>** Definir un texto que se ha insertado en un documento.
- **<mark>** Marcar el texto con color llamativo.

# Formatear

- **<q>** Texto entrecomillado.
- **<s>** Representar texto que no es correcto. Tachado.
- **<small>** Texto más pequeño.
- **<strong>** Definir texto con importancia.
- **<sub>** Texto con subíndice.
- **<sup>** Texto con superíndice.
- **<time>** Definir horas y fechas.
- **<u>** Texto subrayado.
- **<pre>** Definir un texto preformateado.



# mark / small

- **<mark>**
  - Esta etiqueta es para resaltar parte de un texto que originalmente no era importante pero ahora lo es por las acciones del usuario.
  - `<span>Curso de <mark>HTML5</mark></span>`
  - En algunos navegadores esta palabra se marca con fondo amarillo.
  - Otras etiquetas parecidas pueden ser:
    - **<em>**                      Para dar énfasis a un texto.
    - **<strong>**                      Es para dar importancia a un texto.
- **<small>**
  - El nuevo propósito de small es presentar la llamada letra pequeña: impresiones legales, etc.
  - `<small>Todos los derechos reservados &copy; 2012</small>`

# address

- **<address>**

- Se utiliza para presentar información de contacto relacionada con el contenido <article> o el cuerpo completo.

<article>

<header><h1>Título del mensaje</h1></header>

Contenido del mensaje

<footer>

    <address>

        <a href="http:// ...">Contacto</a>

    </address>

</footer>

</article>

# cite

- Citas (**<cite>...</cite>**)
- Permite realizar una cita **dentro de un párrafo de texto.**
- Útiles para explicar el código.
  - `<p>`Si se desea citar, hemos de utilizar la marca
  - `&lt;cite&gt;`, por ejemplo:
  - `<cite>`esto es una cita a Cervantes`</cite></p>`
- Ahora se utiliza para indicar el título de un libro, de una película, una canción.
  - `<span>`
    - Consulta el libro **`<cite>Programación Java</cite>`**
  - `</span>`

# time

- **<time>**

- Representa una fecha comprensible para humanos y navegadores.
- El atributo `datetime` es para el navegador.

`<article>`

`<header>`

`<h1>Titulo del mensaje</h1>`

`<time datetime="2011-10-12" pubdate>Publicado 12-10-2012</time>`

`</header>`

Este es el contenido del mensaje

`</article>`

- El atributo **pubdate** indica que el valor del atributo `datetime` coincide con la fecha de publicación.

# time

- **Ejemplos de time:**

`<time datetime="2012-10-08T12:00">12h del 8 de octubre de 2012</time>`

`<time datetime="12:00">Las 12 horas</time>`

`<time datetime="2012">Año 2012</time>`

`<time>8 de octubre de 2012</time>`

# time

- En HTML5 podemos implementar duraciones. Utilizando el prefijo P para períodos, D para días, H para horas, M para minutos, y S para segundos.
  - `<time datetime="P4D">` es una duración de 4 días.
- Si usamos una T después de P nos permite todavía ser mas precisos:
  - `<time datetime="PT23H 9M 2.343S">`
    - Es una duración de 23 horas, 9 minutos y 2343 segundos.

# pre

- La etiqueta **<pre>** le indica a los navegadores que se trata de un texto preformateado.  
Los textos preformateados tienen las siguientes características:
  - Se representan con un tipo de fuente de ancho fijo.
  - Respetan los saltos de línea.
  - Respetan los espacios en blanco.
  - Se pueden utilizar para mostrar código.

# Etiquetas para imágenes

- **<img>** Mostrar imágenes.
- **<map>** Define una imagen que representa un mapa.
- **<area>** Define una región dentro de un mapa.
- **<canvas>** Define una región para dibujar en modo gráfico: Círculos, líneas, ...
- **<figcaption>** Indicar un título a una imagen, dentro de un elemento figure.
- **<figure>** Para especificar el contenido de una imagen.



# Imágenes

- Es posible utilizar tres formatos de imágenes: **PNG, JPEG y GIF**.
- El formato **PNG es el estándar que recomienda el W3C**, totalmente libre. Soporta canal *alpha*, para crear efectos con transparencias de diferente opacidad.
  - El formato **GIF sólo puede almacenar 256 colores**, aunque uno de ellos es el transparente.
  - El formato **JPEG es el adecuado para imágenes con muchos colores**, como fotografías.
- La sintaxis es:
  - ``
  - **La descripción es obligatoria. El ancho y el alto son muy recomendables**, ya que agilizan el renderizado de la página.

# Imágenes

- Para referenciar imágenes se puede hacer a rutas relativas o mediante URLs.
- `** y **<figcaption>**
  - Con **figure** podemos definir el contenido del documento para hacer referencia a imágenes.

**<article>**

**<header>...</header>**

Contenido en texto del primer artículo

**<figure>**

****

**<figcaption>Una descripción de la imagen</figcaption>**

**</figure>**

**<footer>**

**<p>Comentarios (0)</p>**

**</footer>**

**</article>**

- Con **figcaption** podemos poner una descripción a la imagen del primer artículo.

# canvas

- Permite dibujar dentro de la pantalla del Navegador en modo gráfico.
- Es un área rectangular, en el que el programador controla todos y cada uno de los píxeles que lo forman, creando, modificando y añadiendo efectos para conseguir el dibujo requerido.
- Es necesario el uso de **JavaScript** para dibujar.

```
<canvas id="myCanvas" width="200" height="100"></canvas>
```

# map / area

```
<!DOCTYPE html>
<html>
<body>

<map name="mapa">
    <area href="cuadrado.html" shape="rect" coords="18,124,111,219">
    <area href="circulo.html" shape="circle" coords="128,60,50">
    <area href="triangulo.html" shape="poly" coords="139,200, 193,110, 236,199">
</map>
</body>
</html>
```

# Etiquetas para Tablas

- **<table>** Define una tabla.
- **<caption>** El título de la tabla.
- **<th>** Define una celda de cabecera.
- **<tr>** Define una fila de una tabla.
- **<td>** Define una celda de una fila.
- **<thead>** Grupos de cabeceras.
- **<tbody>** Grupos en el cuerpo de la tabla.
- **<tfoot>** Grupos en el pie de página.
- **<col>** Especificar las propiedades de una columna dentro de un colgroup
- **<colgroup>** Especificar un grupo de una o más columnas en una tabla.

# Tablas

En HTML podemos utilizar las tablas para presentar información de forma tabulada; **nunca para maquetar una página.**

- Las etiquetas HTML necesarias son:
  - **<table>: Permite crear una tabla.**
  - **<caption>: Fija el título de la tabla.**
  - **<tr>: Crea una nueva fila en la tabla.**
  - **<td>: Crea una celda dentro de una fila.**
  - **<th>: Crea una celda de encabezamiento.**

# colgroup / col

```
<table>
  <colgroup>
    <col span="2" style="background-color:red">
    <col style="background-color:yellow">
  </colgroup>
  <tr>
    <th>ISBN</th>
    <th>Title</th>
    <th>Price</th>
  </tr>
  <tr>
    <td>3476896</td>
    <td>My first HTML</td>
    <td>$53</td>
  </tr>
  <tr>
    <td>5869207</td>
    <td>My first CSS</td>
    <td>$49</td>
  </tr>
</table>
```

ISBN	Title	Price
3476896	My first HTML	\$53
5869207	My first CSS	\$49



# thead / tbody / tfoot

```
<head>
<style>
thead {color:green;}
tbody {color:blue;}
tfoot {color:red;}

table, th, td {
  border: 1px solid black;
}
</style>
</head>

<body>
<table>
  <thead>
    <tr>
      <th>Month</th>
      <th>Savings</th>
    </tr>
  </thead>
```

```
<tfoot>
  <tr>
    <td>Sum</td>
    <td>$180</td>
  </tr>
</tfoot>
<tbody>
  <tr>
    <td>January</td>
    <td>$100</td>
  </tr>
  <tr>
    <td>February</td>
    <td>$80</td>
  </tr>
</tbody>
</table>
```

# Etiquetas para formularios

- **<form>** Define un formulario de HTML para introducir datos.
- **<input>** Define un control de entrada (de distintos tipos).
- **<textarea>** Define un campo de observaciones con varias líneas y columnas.
- **<button>** Un botón.
- **<select>** Desplegable para 1 o n opciones.
- **<option>** Defines an option in a drop-down list
- **<label>** Etiqueta para un campo de entrada input.
- **<fieldset>** Agrupa los elementos de un form.
- **<legend>** Define un título para un <fieldset>.
- **<datalist>** Especificar una lista de valores predefinidos.
- **<output>** Define the resultado de un cálculo.
- **<keygen>** Define un par de claves (pública y privada).
- **<optgroup>** Agrupar las opciones de un combo <select>

# Formularios

- La etiqueta principal es **<form>**
  - Se indicará como atributo el script de interpretación del formulario y el método de envío:
  - **action**="script" → podemos llamar a un Servlet / JSP / PHP / ...
  - **method**="post | get"
    - Si se omite el atributo method, por defecto es get.

# Formularios

- Etiqueta **`<input type="file" ... />`**
  - Para subir ficheros al servidor.
- Envío de ficheros al servidor:
  - `<form name="form_producto"`  
**`enctype="multipart/form-data"`**  
`action="grabar_productos.php" method="post">`
    - `<input type='file' id='foto' name='foto'`  
`class='campo_formulario' size='40' />`
    - ...
  - `</form>`

# Formularios

- **Campos input de texto:** `<input type="text" ... />`
  - Se le ha de indicar el ancho en caracteres mediante el atributo **size**
  - Se recomienda usar el atributo **id** para identificarlo, útil para las CSS
  - El atributo **name** indica el nombre del campo, es muy recomendable que sea igual al id.
  - El atributo **maxsize** indica el número máximo de caracteres.
  - El atributo **value** le fija un texto por defecto.
- **Campos de contraseña** `<input type="password" ".../>`
  - Es idéntico al anterior, salvo que oculta lo que se escribe.

# Formularios

## Etiquetar campos: <label for="nombreCampo">

- Se encarga de asociar un texto con su campo correspondiente. El atributo **for** recibe el id del campo de texto al que se desea asociar.

- ```
<p>
<label for="nombre">Nombre:</label>
<br />
<input type="text" name="nombre" id="nombre" />
</p>
```

## Áreas de texto: <textarea>

- Permite que se introduzca texto en varias líneas.
- Dispone de los atributos **id**, **name** y **title**.
- El atributo **cols** fija el ancho en caracteres, así como **rows** lo hace para el alto.

```
<p>
<label for="comentario">¿Algún comentario?</label>
<br />
<textarea name="comentario" id="comentario"
cols="30" rows="5">
Bla bla bla
</textarea>
```

# Formularios

## Casillas de verificación: `<input type="checkbox">`

- Dispone de los atributos **id**, **name** y **title** que funcionan de manera normal.
- El atributo **value** indica el valor que se pasará al script que interprete el formulario cuando el *checkbox* esté activado.
- Para activar una casilla por defecto se usa el atributo **checked="checked"**

```
<p>Has jugado a...<br />
<input type="checkbox" name="monkey1" id="monkey1"
value="si" />
<label for="monkey1">Monkey Island I</label>
</p>
```

# Formularios

## Botones de selección (radiobuttons): `<input type="radio">`

- Se trata de casillas circulares agrupadas, sólo una podrá estar activa en un momento determinado. El atributo **value** especifica el valor a enviar al *script*.
- Para agruparlos, los elementos del mismo grupo han de tener el mismo valor para el atributo **name**. Los **id** han de seguir siendo distintos.

```
<input type="radio" name="peli" id="sw_hope"
value="hope" checked="checked" />
<label for="sw_hope">A New Hope</label>
```

```
<input type="radio" name="peli" id="sw_empire"
value="empire" />
<label for="sw_empire">The Empire Strikes
Back</label>
```

```
<input type="radio" name="peli" id="sw_jedi"
value="jedi" />
<label for="sw_jedi">
The Return of the Jedi</label>
```



# Formularios

## Listas de selección (combobox): <select...>

- Presentan una lista con una serie de opciones.
- La etiqueta que las crea es **select**, entre la apertura y el cierre se utiliza la etiqueta **option** para añadir una entrada a la lista.

Utilizar los atributos  
**Size y múltiple**  
Para convertir el combo  
En una lista.

```
<p>
<label for="peli">Película preferida:</label>
<br />

<select name="peli" id="peli">
  <option value="hope">A New Hope</option>

  <option value="empire">The Empire
    Strikes Back</option>

  <option value="jedi" selected="selected">
    The Return of the Jedi</option>

</select>
</p>
```

# Formularios

## Botones de envío y reestablecimiento.

- `<input type="submit" name="nombre" value="valor" />`
  - Se encarga de enviar el formulario.
- `<input type="reset" name="Nombre" value="valor" />`
  - Se encarga de resetear los valores del formulario.
  - Cuidado con su mal uso, puede ser molesto.

## OTROS BOTONES:

- `<input type="button" name="Nombre" value="valor" />`
- A todos los botones se le pueden asociar eventos, y estos llamar a funciones de java script.
  - Ejemplo: `<input type="button" name="boton" value="pulsa aquí" onclick = "javascript:funcion()" />`
- `<input type="file" name = "nombre" value="valor" />`
  - Muestra una caja y un botón para poder seleccionar un fichero.

# Formularios

## **Campos ocultos:**

```
<input type='hidden' name='nombre' value='valor' />
```

No son visibles en el formulario, se utilizan para pasar información entre las páginas.

# Formularios Web

- El elemento **form** no cambia mucho, añade los siguientes atributos:
  - **autocomplete**: valores posibles on / off. Con el valor off los input tendrán la función autocompletar desactivada.
  - **novalidate**: Por defecto los formularios son automáticamente validados. Se puede desactivar con este atributo:
    - `<form name="miFormulario" novalidate="novalidate">`

# input

- *En HTML5 no es necesario cerrar la etiqueta input pero se aconseja por compatibilidad entre navegadores.*
- **DATOS DE IDENTIFICACIÓN:**
  - Una interesante novedad en HTML5 es la ampliación de la etiqueta *input* para simplificar el ingreso y la validación de variados elementos de identificación y localización. Los nuevos tipos de input relacionados con esta característica son:
  - **tel**, para el ingreso de números de teléfono.
    - `<input type="tel" name="mitelefono">`
  - **url**, para ingresar una dirección web. **Validado**
    - `<input type="url" name="url">`
  - **email**, para la dirección de correo electrónico: **Validado**
    - `<input type="email" name="email">`

# input

- **Fechas y Horas:** (depende del navegador nos puede construir un calendario o no).
  - Soporte para introducir:
    - Date:** `<input type="date" id="fecha" name="fecha" />`
    - Week: `<input type="week" id="fecha2" name="fecha2" />`
    - Month: `<input type="month" id="fecha3" name="fecha3" />`
    - Time:** `<input type="time" id="fecha4" name="fecha4" />`
    - Datetime: `<input type="datetime" id="fecha5" name="fecha5" />`
    - Datetime-local: `<input type="datetime-local" id="fecha6" name="fecha6" />`

# input

- **date** : selecciona la fecha completa, día-mes-año.
- **datetime**: selecciona la hora completa más el día, mes, año y zona horaria (timezone).
- **datetime- local**: incluye la hora actual, día, mes y año (no incluye zona horaria).
- **time**: selecciona la hora actual en horas y minutos.
- **month**: selecciona el mes y el año.
- **week** : indica la semana (un número de 1 a 52) y el año.

# input

- Trabajo con números (sólo en algunos navegadores):
  - **Range**: dibuja en pantalla un control y permite al usuario seleccionar un valor a partir de un deslizador.
    - Att:
      - Min, max, step: valores mínimos, máximos y el paso.
      - `<input type="range" name="deslizador" min="1" max="100" step="1">`
  - **Number**: Para introducir números, permite al usuario seleccionar los valores. Mediante dos botones:
    - `<input type="number" name="numero" id="numero" min="5" max="100" step="5" />`
    - Mismos atributos que el anterior.



# input

- **search:**

- La etiqueta *search* no es más que un típico campo de input de texto, sólo que con un formato similar al muy conocido de búsqueda de contenido. Algunos navegadores agregan a la derecha del casillero una “x” para eliminar el texto ingresado.
- `<input type="search" name="buscador">`

# input

- **Color:**
  - Seleccionar un color. Se espera un número hexadecimal del estilo: #00FF90.
  - Algunos navegadores muestran un desplegable para seleccionar el color.
  - `<input type="color" name="color" id="color" />`

# Nuevos atributos

- **Autofocus** fuerza el foco (la posición del cursor) dentro de un campo, una vez que se carga la página.
  - `<input type="tel" id="telefono" name="telefono" autofocus="autofocus" />`
- **Autocomplete** permite el rellenado automático del input en base al texto que vaya escribiendo el usuario, en base a los textos introducidos anteriormente. Puede utilizarse como atributo de un campo de entrada o del formulario completo.
  - `<input type="email" id="correo" name="correo" autocomplete="on" />`
- **Placeholder** muestra un texto por defecto en el input hasta que este tiene el foco, en ese momento el texto desaparece para que el usuario ingrese el texto que desee.
  - `<input type="url" id="url" name="url" placeholder="http://" />`
- **Required** es un atributo que se aplica a un campo de entrada e impide que se envíe el formulario si el campo no se a rellenado.
  - `<input type="text" id="nombre" name="nombre" required="required" />`

# Nuevos atributos

- **Multiple:** permite introducir varios valores en el mismo campo, tienen que ir separados por comas (y sin blancos).
  - Se permite en elementos como email y file.
    - `<input type="email" id="correo" name="correo" multiple="multiple"/>`
  - Se puede ver también así:
    - `<input type="email" id="correo" name="correo" multiple/>`

# Nuevos atributos

- **Pattern:**

- Permite establecer un patrón con el que se valida el campo.
- Utiliza expresiones regulares.
  - `<input type="text" pattern="[0-9]{5}" name="cp" id="cp" />`

- **Form:**

- Permite declarar elementos del formulario que están fuera de las etiquetas form.
- Hace referencia al **nombre** del form.

```
<nav>
```

```
  <p>
```

```
    <input type="search" name="buscador" form="miFormulario" />
```

```
  </p>
```

```
</nav>
```

# Nuevos elementos para formularios

- **Datalist:**

- Permite crear una lista de sugerencias que luego se utilizará dentro de un campo de formulario.
- Creamos una lista con datalist, y luego se referencia desde otro control con el atributo list.

```
<datalist id="info">
```

```
  <option value="112">Emergencias</option>
```

```
  <option value="091">Policía</option>
```

```
  <option value="060">Ambulancias</option>
```

```
</datalist>
```

```
<p>Telefono de asistencia: <input type="tel"  
  name="tno" id="tno" list="info" /></p>
```

# Nuevos elementos para formularios

- **Progress**

- Muestra una barra de progreso. Indica el progreso o estado de una tarea, indicamos un valor y un máximo.
  - `<progress id="progreso" value="5" max="10"></progress>`

- **Meter**

- Muestra una escala para representar una medida. Atributos: min, max (límites de la escala), value: determina el valor medio y low, high y optimum para segmentar la escala.
- Por ejemplo: para representar la seguridad de una clave.

- **Output**

- Representa el resultado de un cálculo. Normalmente se utiliza para mostrar el resultado de un cálculo en javascript.
- `<output>valor</output>`
- **Atributos:**
  - name: el nombre,
  - for: hace referencia a los ids de los controles que influyen en el cálculo.

# Ejemplo: output

```
<form onsubmit="return false" oninput="o.value =  
parseInt(a.value) + parseInt(b.value)">  
  <input name="a" type="number" step="any"> +  
  <input name="b" type="number" step="any"> =  
  <output name="o"></output>  
</form>
```

The diagram shows a visual representation of the HTML form's output. It consists of a horizontal row containing a text input field with the value '2', followed by a plus sign, another text input field with the value '3', an equals sign, and the number '5'. Below the first input field is a blue arrow pointing upwards to it, with the text '<input type="number">' written in blue. Below the second input field is a blue arrow pointing upwards to it. Below the equals sign and the number '5' is a blue arrow pointing upwards to them, with the text '<output>' written in blue.



# Ejemplo: output

- También se puede utilizar la propiedad: **valueAsNumber** de **number** para convertir directamente a número.

```
<form onsubmit="return false" oninput="o.value =  
a.valueAsNumber + b.valueAsNumber">  
  <input name="a" id="a" type="number" step="any"> +  
  <input name="b" id="b" type="number" step="any"> =  
  <output name="o" for="a b"></output>  
</form>
```

# fieldset / leyend

<form>

<fieldset> <!-- Agrupar campos de un formulario -->

<legend>Personalia:</legend> <!-- Título -->

Name: <input type="text"><br>

Email: <input type="text"><br>

Date of birth: <input type="text">

</fieldset>

</form>



# optgroup

- Para agrupar las opciones de un combo:

`<select>`

`<optgroup label="Swedish Cars">`

`<option value="volvo">Volvo</option>`

`<option value="saab">Saab</option>`

`</optgroup>`

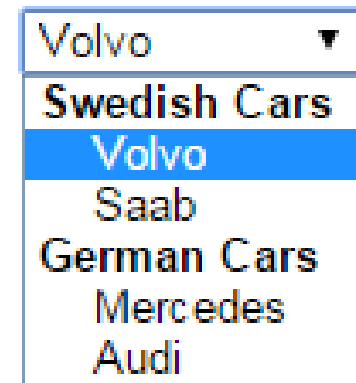
`<optgroup label="German Cars">`

`<option value="mercedes">Mercedes</option>`

`<option value="audi">Audi</option>`

`</optgroup>`

`</select>`



# keygen

- Esta etiqueta sirve para **enviar claves en el autenticado web**. Al enviarse el formulario se crean 2 claves, una pública que se envía al servidor y que posteriormente se usará para seguir autenticándonos en el sistema, y otra privada que se almacena localmente en el navegador.

```
<?php
if ($_POST['clave'])
echo "clave pública enviada: " . $_POST['clave'];
?>
```

```
<form action="formu.php" method="post">
  User: <input type="text" name="usuario">
  <!-- añadimos el campo de tipo keygen -->
  Pass: <keygen name="clave">
  <input type="submit" value="Enviar">
</form>
```

# Etiquetas de estilo y semánticas

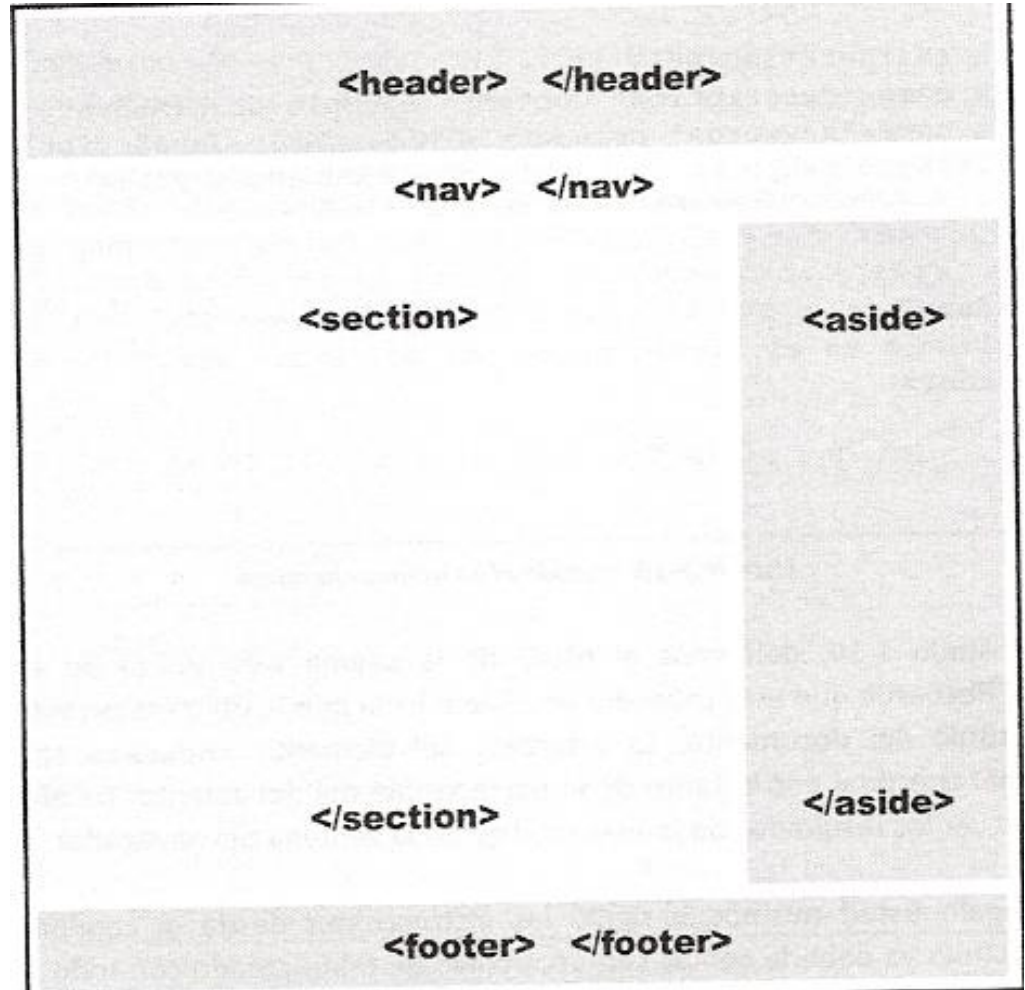
- Todas las etiquetas de contenido semántico se añaden en la especificación de HTML5. Surgen con la necesidad del abuso de etiquetas DIV para la maquetación de las páginas.
- **<style>** Definir estilos dentro del documento. Sin CSS.
- **<div>** Sección en bloque de un documento.
- **<span>** Sección de un documento.
- **<header>** Define cabecera del documento o de una sección.
- **<footer>** Define pie de página del documento o de una sección.
- **<section>** Define una sección de un documento,
- **<article>** Define un artículo.
- **<aside>** Define el contenido lateral de una página.
- **<details>** Define detalles que se pueden mostrar u ocultar.
- **<summary>** Define la cabecera visible de un grupo de detalles.

# Estructura del cuerpo

- Dentro del elemento body, se agregan nuevas secciones:
  - <header>
  - <nav>
  - <section>
  - <aside>
  - <footer>
- La colocación de estas partes del body se realiza mediante estilos css.

# Estructura del cuerpo

```
<body>  
  <header>  
  </header>  
  
  <nav>  
  </nav>  
  
  <section>  
  </section>  
  
  <aside>  
  </aside>  
  
  <footer>  
  </footer>  
</body>
```



# Estructura del cuerpo

- **<header>**

- Representa la cabecera y está destinado a la colocación de encabezados: <h1> ... <h6>, también para logos, imágenes.

- <body>

- <header><h1>Titulo de la página</h1></header>

- **<nav>**

- Representa la barra de navegación.

- Dentro de este podemos utilizar listas de enlaces.



# Estructura del cuerpo

- **<section>**
  - El elemento central de la página.
  - Información de interés.
  - Sección principal.
- **<aside>**
  - Representa la barra lateral de navegación o información para seleccionar contenidos a mostrar en la section.
  - Puede ubicarse al lado derecho o izquierdo de la section.
  - No tiene ubicación predefinida.

# Estructura del cuerpo

- **<footer>**
  - Información de tipo institucional.
  - El copyright, nombre de la empresa, enlaces de contacto, toda la información relevante a mostrar.

<footer>

Derechos reservados &copy; 2010-2011

</footer>

# Dentro del cuerpo

- Dentro de la sección **section**, la información a mostrar se puede organizar en **article**.

<section>

<article>

Primer artículo ...

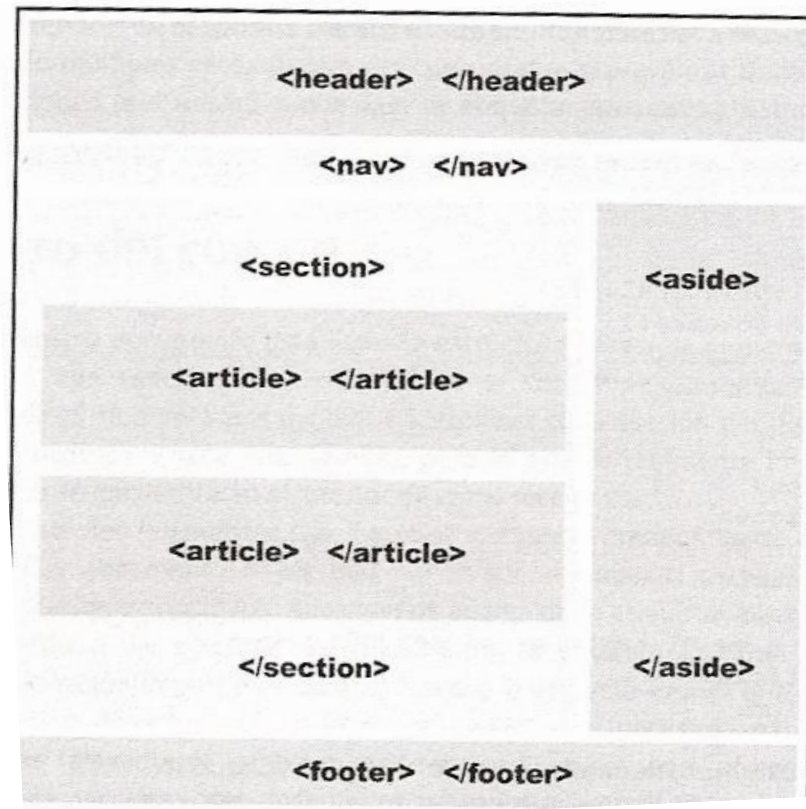
</article>

<article>

Segundo artículo ...

<article>

</section>



# Dentro del cuerpo

- Dentro de un article podemos tener una estructura formada por header y footer.
- Estas dos etiquetas se pueden usar para la estructura del documento y otros elementos como la etiqueta article.

```
<article>
  <header>
    <h1>Título del artículo</h1>
  </header>
  Contenido del primer artículo
  <footer>
    <p>Comentarios(0)</p>
  </footer>
</article>
```

# style

- ```
<html>
<head>
<style>
h1 {color:red;}
p {color:blue;}
</style>
</head>
<body>

<h1>A heading</h1>
<p>A paragraph.</p>

</body>
</html>
```

# details / summary

- <details>
  - <summary>Copyright 1999-2014.</summary>
  - <p> - by Refsnes Data. All Rights Reserved.</p>
  - <p>All content and graphics on this web site are the property of the company Refsnes Data.</p>
- </details>

▼ Copyright 1999-2014.

- by Refsnes Data. All Rights Reserved.

All content and graphics on this web site are the property of the company Refsnes Data.

# Etiquetas DIV / SPAN

- Las etiquetas **div** (llamadas capas) son etiquetas contenedoras. Se utilizan para organizar la información que se muestra en la página. Se utiliza para la maquetación de páginas.
- **span**: Se utiliza para dar estilo a un texto dentro de nuestra página.

# Maquetación de Páginas con capas



```
<div id="contenedor">  
  <div id="banner"></div>  
  <div id="centro">  
    <div id="menu"></div>  
    <div id="derecha">  
      <div id="body"></div>  
      <div id="footer"></div>  
    </div>  
  </div>  
</div>
```



# Maquetación de Páginas con capas

- **Estilos asociados:**

```
#contenedor {  
    margin: 0px auto;  
    width: 90% (admiten tb. 800 px)  
    text-align: center;  
}
```

```
#cabecera {  
    width: 100%;  
    height: 100px;  
    background-color: orange;  
}
```

```
#centro {  
    width: 100%;  
}
```

```
#menu {  
    width: 30%;  
    float: left;  
}
```

```
#derecha {  
    width: 65%;  
    float: right;  
}
```

```
#body {  
    width: 100%;  
}
```

```
#footer {  
    width: 100%;  
}
```

# Audio y Video

- **<audio>** Define un contenido de sonido.
- **<video>** Define un video o película.

# Video

- De una forma simple se puede mostrar un video con la etiqueta video:

```
<video width="640" height="360"  
src="http://www.youtube.com/demo/google_main.mp4" controls  
autobuffer><p> Try this page in Safari 4! Or you can <a  
href="http://www.youtube.com/demo/google_main.mp4">download  
the video</a> instead.</p>  
</video>
```

```
<video src="movie.ogv" controls="controls">  
</video>
```

El atributo "controls" proporciona los controles nativos que ofrece HTML5 para la etiqueta video

# Video

- El elemento vídeo permite añadir múltiples elementos, y con éstos elementos direccionar a distintos archivos de video.
- El navegador usará el primer formato reconocido de entre los formatos que le introduzcamos en la etiqueta **<source>** por lo que usaría el que aceptase y el resto los ignoraría:

```
<video width="320" height="240" controls="controls">  
  <source src="movie.ogg" type="video/ogg" />  
  <source src="movie.mp4" type="video/mp4" />  
  Your browser does not support the video tag.  
</video>
```

# Video: Formatos Disponibles

- **Ogg:** archivos Ogg con códec de vídeo Theodora y códec de audio Vorbis.
- **MPEG4:** archivos MPEG4 con códec de vídeo H.264 y códec de audio ACC.
- **WebM:** archivos WebM con códec de vídeo VP8 y códec de audio Vorbis.

# Video: Atributos

- **Autoplay:** Especifica que el vídeo comenzará a reproducirse tan pronto como esté listo
- **Controls:** Especifica que los controles se mostrará como un botón de reproducción.
- **Height(píxeles):** *Especifica la altura del reproductor de vídeo*
- **Loop:** Especifica que el archivo multimedia se iniciará de nuevo, una vez que haya terminado.
- **Preload:** Especifica que el vídeo se cargará en la carga de la página, y listo para reproducir. Se ignora si "autoplay" está activado.
- **Src:** Especifica la dirección URL del vídeo a reproducir.
- **Width(píxeles):** *Especifica el ancho del reproductor de vídeo.*

# Audio

- Etiqueta similar al Video.
- Formatos:
  - Ogg
  - MP3
  - WAV.

```
<audio src="song.ogg" controls="controls">  
</audio>
```

# Audio: Atributos

- **Autoplay:** Especifica que el audio comenzará a reproducirse tan pronto como esté listo
- **Controls:** Especifica que los controles se mostrará como un botón de reproducción.
- **Loop:** Especifica que el archivo multimedia se iniciará de nuevo, una vez que haya terminado.
- **Preload:** Especifica que el audio se cargará en la carga de la página, y listo para reproducir. Se ignora si "autoplay" está activado.
- **Src:** Especifica la dirección URL del audio a reproducir.



# Etiquetas meta

- **<head>** Información y meta-información del documento.
- **<meta>** Metadatos.
- **<base>** Especificar rutas base para referenciar a los objetos de forma local.

# base

- `<head>`  
`<base href="http://www.w3schools.com/images/" target="_blank">`  
`</head>`  
`<body>`  
``  
`<a href="http://www.w3schools.com">W3Schools`  
`</a>`  
`</body>`

# Etiquetas de programación

- **<script>** Añadir código o adjuntar ficheros javascript.
- **<noscript>** Contenido alternativo si el cliente no soporta javascript.
- **<embed>** Definir un contenedor para elementos no HTML.
- **<object>** Para incrustar un objeto.
- **<param>** Definir parámetros que pasamos al objeto.

# embed

- Para incrustar una animación de flash:
- `<embed src="helloworld.swf">`

# object / param

- Ejemplo para incrustar un **PDF** en una página:
  - `<object data='miPDF.pdf' type='application/pdf' width='100%' height='800px'></object>`
- También se pueden incrustar **applet** y pasar **parámetros**:
  - `<object type="application/x-java-applet" height="300" width="550">`
    - `<param name="code" value="Sample" />`
    - `<param name="archive" value="Sample.jar" />`
    - `<param name="java_arguments" value="-Djnlp.packEnabled=true"/>` Applet failed to run. No Java plug-in was found.
  - `</object>`

# iframe

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<iframe src="http://www.w3schools.com">
```

```
<p>Your browser does not support iframes.</p>
```

```
</iframe>
```

```
</body>
```

```
</html>
```

Incrustar contenido

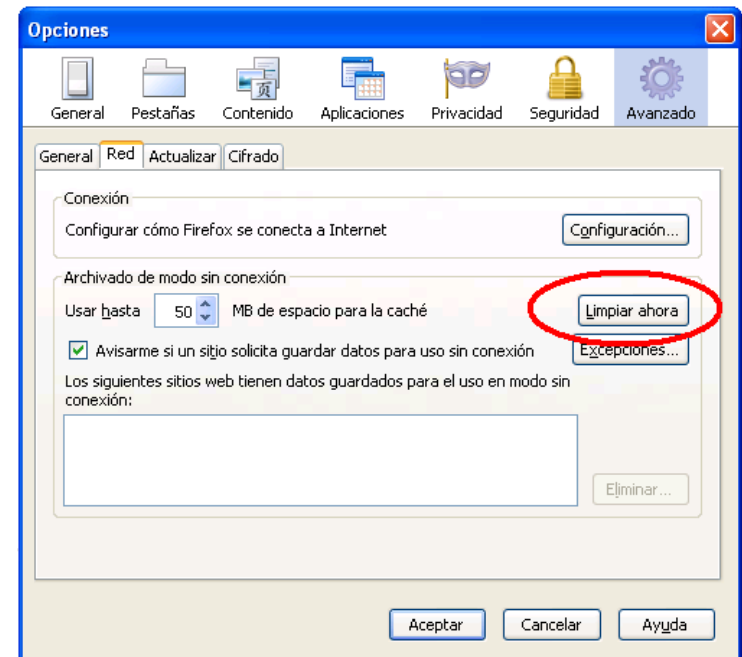
# Validación de formularios HTML con JavaScript (en el cliente)

```
!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
<script type="text/javascript">
    function valida(){
        if (document.miForm.nombre.value=="")
            alert("Rellenar el campo");

        else
            document.submit();
    }
</script>
</head>
<body>
<form name="miForm">
Su nombre es:
<input type="text" name="nombre" value="nombre"></input>
<input type="button" name="enviar" value="enviar" onclick="valida()"></input>
</form>
</body>
</html>
```

# FireFox

- Instalar las plugins:
  - MeasureIt.
  - ColorZilla.
  - Firebug.
  - Web Developer.
- Limpiar la caché:
  - Menú Herramientas
    - Opciones.





# Ejemplos

- [http://www.w3schools.com/html/html\\_examples.asp](http://www.w3schools.com/html/html_examples.asp)