

# HTML

LABORATORIO DE COMPUTACIÓN III

UTN-FRA

# HTML

Es un lenguaje de marcado que utiliza etiquetas e hipertexto para definir la estructura de una página web.

Es interpretado por el navegador.

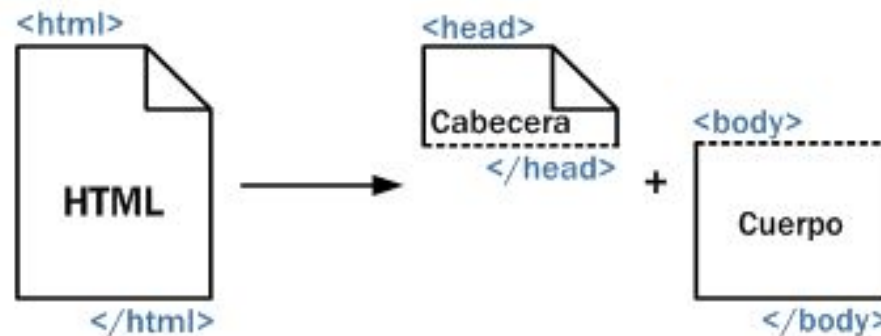
Para el desarrollo del curso utilizaremos un editor de código. Aunque HTML es texto plano, es decir se podría escribir en el block de notas. Utilizaremos un editor ya que trae una serie de herramientas que aumentan la productividad.

# Estructura Interna de una Página HTML

Las páginas HTML se dividen en dos partes: la cabecera y el cuerpo.

La cabecera incluye información sobre la propia página, como por ejemplo su título y su idioma.

El cuerpo de la página incluye todos sus contenidos, como párrafos de texto e imágenes.



El cuerpo (*body* en inglés) contiene todo lo que el usuario ve en su pantalla y la cabecera (*head* en inglés) contiene todo lo que no se ve (con la única excepción del título de la página, que los navegadores muestran como título de sus ventanas).

# Estructura Interna de una Página HTML

```
index.html x
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <meta http-equiv="X-UA-Compatible" content="ie=edge">
7      <title>Document</title>
8  </head>
9  <body>
10
11
12
13  </body>
14  </html>
```

# Etiquetas y Atributos

HTML define 81 etiquetas que los diseñadores pueden utilizar para *marcar* los diferentes elementos que componen una página:

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

Las etiquetas se pueden personalizar a través de **atributos**, cada etiqueta define su propia lista de atributos disponibles. Además, cada atributo también indica el tipo de valor que se le puede asignar. Si el valor de un atributo no es válido, el navegador ignora ese atributo.

Aunque cada una de las etiquetas HTML define sus propios atributos, algunos de estos son comunes a muchas o casi todas las etiquetas.

# Atributos Comunes

Los atributos comunes se dividen en cuatro grupos según su funcionalidad:

**Atributos básicos:** se pueden utilizar prácticamente en todas las etiquetas HTML.

**Atributos para internacionalización:** los utilizan las páginas que muestran sus contenidos en varios idiomas.

**Atributos de eventos:** sólo se utilizan en las páginas web dinámicas creadas con JavaScript.

**Atributos de foco:** relacionados principalmente con la accesibilidad de los sitios web.



# Atributos Básicos

Atributo	Descripción
<b>id</b> = "texto"	Establece un identificador único a cada elemento dentro de una página HTML
<b>class</b> = "texto"	Establece la clase CSS que se aplica a los estilos del elemento
<b>style</b> = "texto"	Establece de forma directa los estilos CSS de un elemento
<b>title</b> = "texto"	Establece el título a un elemento (mejora la accesibilidad y los navegadores lo muestran cuando el usuario pasa el ratón por encima del elemento)

# Atributos de Internacionalización

Estos atributos son útiles para aquellas páginas que muestran sus contenidos en varios idiomas y para aquellas que quieren indicar de forma explícita el idioma de sus contenidos:

Atributo	Descripción
<b>Lang</b> = "código de idioma"	Indica el idioma del elemento mediante un código predefinido
<b>xml:lang</b> = "código de idioma"	Indica el idioma del elemento mediante un código predefinido
<b>dir</b>	Indica la dirección del texto (útil para los idiomas que escriben de derecha a izquierda)



# Atributos de Eventos

Atributo	Descripción	Elementos que pueden usarlo
<b>onblur</b>	Deseleccionar el elemento	<button>, <input>, <label>, <select>, <textarea>, <body>
<b>onchange</b>	Deseleccionar un elemento que se ha modificado	<input>, <select>, <textarea>
<b>onclick</b>	Pinchar y soltar el ratón	Todos los elementos
<b>ondblclick</b>	Pinchar dos veces seguidas con el ratón	Todos los elementos
<b>onfocus</b>	Seleccionar un elemento	<button>, <input>, <label>, <select>, <textarea>, <body>
<b>onkeydown</b>	Pulsar una tecla (sin soltar)	Elementos de formulario y <body>
<b>onkeypress</b>	Pulsar una tecla	Elementos de formulario y <body>
<b>onkeyup</b>	Soltar una tecla pulsada	Elementos de formulario y <body>
<b>onload</b>	La página se ha cargado completamente	<body>

# Atributos de Eventos II

<b>onmousedown</b>	Pulsar (sin soltar) un botón del ratón	Todos los elementos
<b>onmousemove</b>	Mover el ratón	Todos los elementos
<b>onmouseout</b>	El ratón " <i>sale</i> " del elemento (pasa por encima de otro elemento)	Todos los elementos
<b>onmouseover</b>	El ratón " <i>entra</i> " en el elemento (pasa por encima del elemento)	Todos los elementos
<b>onmouseup</b>	Soltar el botón que estaba pulsado en el ratón	Todos los elementos
<b>onreset</b>	Inicializar el formulario (borrar todos sus datos)	<form>
<b>onresize</b>	Se ha modificado el tamaño de la ventana del navegador	<body>
<b>onselect</b>	Seleccionar un texto	<input>, <textarea>
<b>onsubmit</b>	Enviar el formulario	<form>
<b>onunload</b>	Se abandona la página (por ejemplo al cerrar el navegador)	<body>

# Atributos de Foco

Cuando el usuario selecciona un elemento en una aplicación, se dice que *"el elemento tiene el foco del programa"*. Si por ejemplo un usuario pincha con su ratón sobre un cuadro de texto y comienza a escribir, ese cuadro de texto tiene el foco del programa, llamado *"focus"* en inglés. Si el usuario selecciona después otro elemento, el elemento original pierde el foco y el nuevo elemento es el que tiene el foco del programa.

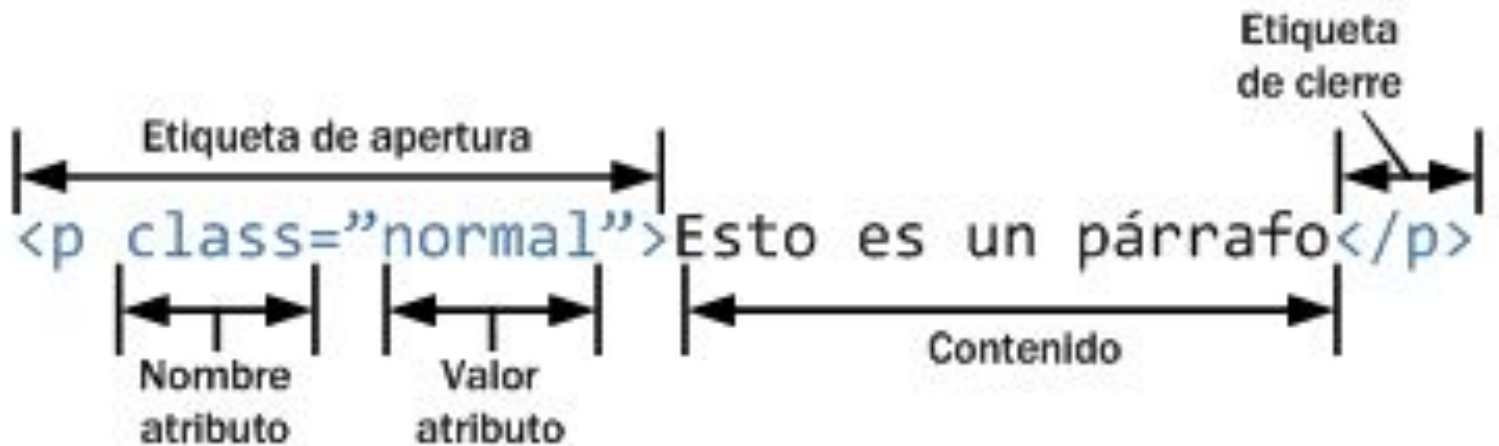
Atributo	Descripción
<b>accesskey</b> = "letra"	Establece una tecla de acceso rápido a un elemento HTML
<b>tabindex</b> = "numero"	Establece la posición del elemento en el orden de tabulación de la página. Su valor debe estar comprendido entre 0 y 32.767
<b>onfocus</b> , <b>onblur</b>	Controlan los eventos JavaScript que se ejecutan cuando el elemento obtiene o pierde el foco

# Elementos HTML

Además de etiquetas y atributos, HTML define el término **elemento** para referirse a las partes que componen los documentos HTML.

Aunque en ocasiones se habla de forma indistinta de "elementos" y "etiquetas", en realidad un elemento HTML es mucho más que una etiqueta, ya que está formado por:

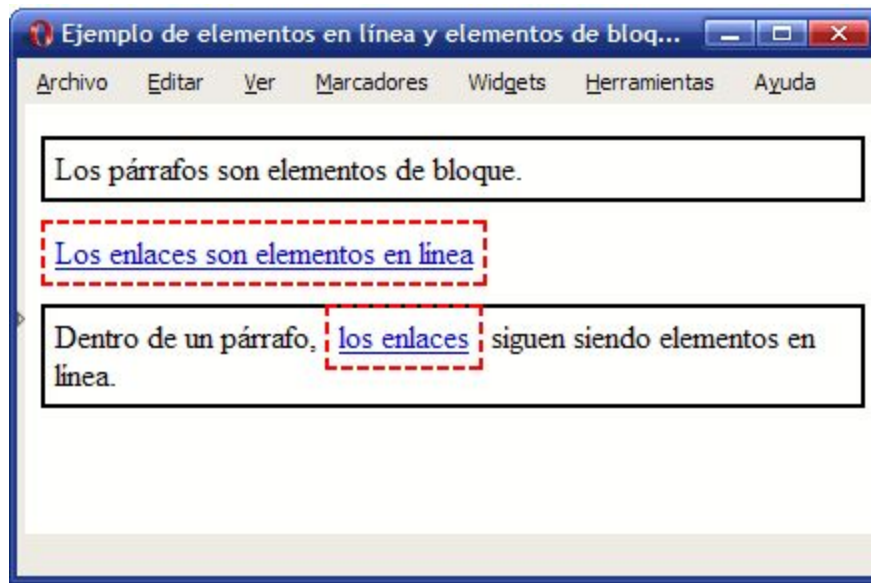
- Una etiqueta de apertura.
- Cero o más atributos.
- Texto encerrado por la etiqueta.
- Una etiqueta de cierre.



# Elementos en línea y en bloque

El lenguaje HTML clasifica a todos los elementos en dos grupos: elementos **en línea** (*inline elements* en inglés) y elementos de **bloque** (*block elements* en inglés).

La principal diferencia entre los dos tipos de elementos es la forma en la que ocupan el espacio disponible en la página. Los elementos de bloque siempre empiezan en una nueva línea y ocupan todo el espacio disponible hasta el final de la línea, aunque sus contenidos no lleguen hasta el final de la línea. Por su parte, los elementos en línea sólo ocupan el espacio necesario para mostrar sus contenidos.



El primer enlace del ejemplo anterior también tiene un texto corto que ocupa solamente la mitad de la anchura de la ventana del navegador. En este caso, el navegador sólo reserva para el enlace el sitio necesario para mostrar sus contenidos. Si se añade otro enlace en esa misma línea, se mostraría a continuación del primer enlace. Por tanto, los elementos `<a>` son elementos en línea.

Por último, el segundo párrafo sigue ocupando todo el espacio disponible hasta el final de cada línea (por ser un elemento de bloque) y el enlace que se encuentra dentro del párrafo sólo ocupa el sitio necesario para mostrar sus contenidos (por ser un elemento en línea).

La mayoría de elementos de bloque pueden contener en su interior elementos en línea y otros elementos de bloque. Los elementos en línea sólo pueden contener texto u otros elementos en línea. En otras palabras, un elemento de bloque no puede aparecer dentro de un elemento en línea. En cambio, un elemento en línea puede aparecer dentro de un elemento de bloque y dentro de otro elemento en línea.



# Sintaxis de las etiquetas HTML

- 1) Las etiquetas se tienen que cerrar de acuerdo a como se abren**
- 2) Los nombres de las etiquetas y atributos siempre se escriben en minúsculas**
- 3) El valor de los atributos siempre se encierra con comillas**
- 4) Los atributos no se pueden comprimir**
- 5) Todas las etiquetas deben cerrarse siempre**

# Texto

El lenguaje HTML incorpora al tratamiento del texto muchas de las ideas y normas establecidas en otros entornos de publicación de contenidos. De esta forma, HTML define etiquetas para **estructurar** el contenido en secciones y párrafos y define otras etiquetas para **marcar** elementos importantes dentro del texto.



# Estructurar Texto      Párrafos

La forma más sencilla de estructurar un texto consiste en separarlo por párrafos. Además, HTML permite incluir títulos que delimitan cada una de las secciones. Para delimitar el texto de un párrafo, se encierra ese texto con la etiqueta `<p>`

Los párrafos creados con HTML son elementos de bloque, por lo que siempre ocupan toda la anchura de la ventana del navegador. Además, no tienen atributos específicos, pero sí que se les pueden asignar los atributos comunes de HTML básicos, de internacionalización y de eventos.

# Estructurar Texto

# Encabezados

Los títulos de sección se utilizan para delimitar el comienzo de cada sección de la página. HTML permite crear secciones de hasta seis niveles de importancia. De esta forma, aunque una página puede definir cualquier número de secciones, sólo puede incluir seis niveles jerárquicos.

Las etiquetas que definen los títulos de sección son `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>` y `<h6>`. La etiqueta `<h1>` es la de mayor importancia y por tanto se utiliza para definir los titulares de la página. La importancia del resto de etiquetas es descendiente, de forma que la etiqueta `<h6>` es la que se utiliza para delimitar las secciones menos importantes de la página.

Al igual que la etiqueta `<p>`, las etiquetas de título de sección son elementos de bloque y no tienen atributos específicos.

# Marcado básico de texto

Los textos habituales están formados por elementos como palabras en negrita o cursiva, anotaciones y correcciones, citas a otros documentos externos, etc. HTML proporciona varias etiquetas para *marcar* cada uno de los diferentes tipos de texto.

La etiqueta `<em>` marca un texto indicando que su importancia es mayor que la del resto del texto. La etiqueta `<strong>` indica que un determinado texto es de la mayor importancia dentro de la página.

Por defecto, los navegadores muestran los elementos `<em>` en cursiva para hacer evidente su importancia y muestran los elementos `<strong>` en negrita, para indicar que son los más importantes.

Por defecto, el texto eliminado (marcado con la etiqueta `<del>`) se muestra tachado.

El texto insertado (marcado con la etiqueta `<ins>`) se muestra subrayado.

HTML define la etiqueta `<blockquote>` para incluir citas textuales en las páginas web.

Para indicar de forma clara que el texto es una cita externa, los navegadores muestran por defecto el texto del elemento `<blockquote>` con un gran margen en la parte izquierda.

# Marcado avanzado de textos

Las páginas y documentos más avanzados suelen incluir otros elementos importantes que se deben marcar de forma adecuada. Por ello, HTML incluye muchas otras etiquetas que permiten marcar más elementos del texto.

La etiqueta `<abbr>` marca las abreviaturas de un texto y la etiqueta `<acronym>` se emplea para marcar las siglas o acrónimos del texto.

En ambos casos, el atributo `title` se puede utilizar para incluir el significado completo de la abreviatura o sigla.

La mayoría de navegadores muestran por defecto un borde inferior punteado para todos los elementos `<abbr>` y `<acronym>`. Al posicionar el puntero del mouse sobre la palabra subrayada, el navegador muestra un pequeño recuadro (llamado *tooltip* en inglés) con el valor del atributo **title**



# Marcado genérico de texto

El estándar HTML incluye una etiqueta llamada `<span>` que se emplea para marcar cualquier elemento que no se puede marcar con las otras etiquetas definidas.

La etiqueta `<span>` se visualiza por defecto con el mismo aspecto que el texto normal. Por tanto es habitual utilizar esta etiqueta junto con los atributos `id` y `class` para modificar posteriormente su aspecto con CSS.

La etiqueta `<span>` se visualiza por defecto con el mismo aspecto que el texto normal. Por tanto es habitual utilizar esta etiqueta junto con los atributos `id` y `class` para modificar posteriormente su aspecto con CSS.

# Espacios en blanco y nuevas líneas

HTML considera *espacio en blanco* a los espacios en blanco, los tabuladores, los retornos de carro y el carácter de nueva línea (ENTER)

Para incluir una nueva línea en un punto y forzar a que el texto que sigue se muestre en la línea inferior, se utiliza la etiqueta `<br>`. La particularidad de `<br>` es que es una etiqueta vacía, es decir, no encierra ningún texto. HTML permite utilizar un atajo para indicar que una etiqueta se está abriendo y cerrando de forma consecutiva: `<br/>`

Para incluir espacios en blanco adicionales, se debe sustituir cada nuevo espacio en blanco por el texto `&nbsp;`;

En ocasiones, es necesario mostrar los espacios en blanco de un texto que no se puede modificar.

En estos casos, se puede utilizar la etiqueta `<pre>`, que muestra el texto tal y como se ha escrito, respetando todos los espacios en blanco y todas las nuevas líneas.

# Caracteres especiales

Entidad	Carácter	Descripción	Traducción
<b>&amp;lt;</b>	<	less than	signo de menor que
<b>&amp;gt;</b>	>	more than	signo de mayor que
<b>&amp;amp;</b>	&	ampersand	ampersand
<b>&amp;quot;</b>	"	quotation mark	comillas
<b>&amp;nbsp;</b>	(espacio en blanco)	non-breaking space	espacio en blanco
<b>&amp;apos;</b>	'	apostrophe	apóstrofo

# Caracteres especiales II

Entidad	Carácter	Descripción oficial
<b>&amp;ntilde;</b>	ñ	latin letter n with tilde
<b>&amp;Ntilde;</b>	Ñ	latin capital n letter with tilde
<b>&amp;aacute;</b>	á	a acute
<b>&amp;eacute;</b>	é	e acute
<b>&amp;iacute;</b>	í	i acute
<b>&amp;oacute;</b>	ó	o acute
<b>&amp;uacute;</b>	ú	u acute
<b>&amp;Aacute;</b>	Á	A acute
<b>&amp;Eacute;</b>	É	E acute
<b>&amp;Iacute;</b>	Í	I acute
<b>&amp;Oacute;</b>	Ó	O acute
<b>&amp;Uacute;</b>	Ú	U acute
<b>&amp;euro;</b>	€	euro

# Enlaces

El lenguaje de marcado HTML se definió teniendo en cuenta algunas de las características que existían en ese momento para la publicación digital de contenidos. Entre los conceptos utilizados en su creación, se encuentra el mecanismo de "*hipertexto*".

De hecho, las letras "HT" de la sigla HTML significan "*hipertexto*" (*hypertext* en inglés), por lo que el significado completo de HTML podría traducirse como "lenguaje de marcado para hipertexto". La incorporación del *hipertexto* fue una de las claves del éxito del lenguaje HTML, ya que permitió crear documentos interactivos que proporcionan información adicional cuando se solicita.

Los enlaces se utilizan para establecer relaciones entre dos recursos. Aunque la mayoría de enlaces relacionan páginas web, también es posible enlazar otros recursos como imágenes, documentos y archivos. El enlace comienza en un recurso y apunta hacia otro recurso. Cada uno de los dos extremos se llaman "*anchors*" en inglés, "*anclas*".

# URL

El acrónimo URL (del inglés *Uniform Resource Locator*) hace referencia al identificador único de cada recurso disponible en Internet.

La URL de un recurso tiene dos objetivos principales:

- Identificar de forma única a ese recurso
- Permitir localizar de forma eficiente ese recurso

De esta forma es posible crear enlaces que apunten de forma inequívoca a una determinada página.

Si se accede a la página principal de Google, la dirección que muestra el navegador es:

<https://www.google.com.ar>



# URL II

Una URL sencilla siempre está formada por las mismas tres partes. Si por ejemplo se considera la siguiente URL:

<http://www.sistemas-utnfra.com.ar/#/pages/carrera/tecnico-programacion/resumen.html>

Las partes que componen la URL anterior son:

- **Protocolo** (<http://>): el mecanismo que debe utilizar el navegador para acceder a ese recurso. Todas las páginas web utilizan <http://>. Las páginas web *seguras* (por ejemplo las de los bancos y las de los servicios de email) utilizan <https://> (se añade una letra S).
- **Servidor** ([www.sistemas-utnfra.com.ar](http://www.sistemas-utnfra.com.ar)): simplificando mucho su explicación, se trata del ordenador en el que se encuentra guardada la página que se quiere acceder. Los navegadores son capaces de obtener la dirección de cada servidor a partir de su nombre.
- **Ruta** ([/#/pages/carrera/tecnico-programacion/resumen.html](http://www.sistemas-utnfra.com.ar/#/pages/carrera/tecnico-programacion/resumen.html)): *camino* que se debe seguir, una vez que se ha llegado al servidor, para localizar el recurso específico que se quiere acceder.

# Enlaces a otra página del mismo sitio

El elemento más importante que tiene una página de internet es el hipervínculo, estos nos permiten cargar otra página en el navegador. Esto es lo que hace diferente la página de un libro con la página de un sitio en internet. Normalmente un libro lo recorremos en forma secuencial, pero un sitio de internet podemos disponer estos enlaces entre un conjunto de páginas y luego tener distintas alternativas de recorrido.

```
<a href="./pagina2.html">Ir a página 2</a>
```

La propiedad href del elemento "a" hace referencia a la página que debe mostrar el navegador si el visitante hace clic sobre el hipervínculo.

# Enlace a otro sitio de Internet

La sintaxis para disponer un hipervínculo a otro sitio de internet es:

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

Ahora la propiedad href la inicializamos con el nombre del dominio del otro sitio.

Algo importante que hay que anteceder al nombre del dominio es el tipo de protocolo a utilizar. Cuando se trata de una página de internet, el protocolo es el http.

Resumiendo a la propiedad href la inicializamos con el nombre del protocolo (http) seguida de dos puntos (:) y dos barras (//) luego la cadena (www.) y finalmente el nombre de dominio del sitio a enlazar.

El elemento "a" tiene una propiedad target que nos permite indicar que la referencia del recurso sea abierto en otra pestaña.

Esta propiedad se llama target y debemos asignarle el valor "\_blank" para indicar que la página sea abierta en otra pestaña.

# Anclas llamadas desde la misma página

HTML nos permite disponer una referencia dentro de la página para poder posteriormente disponer un hipervínculo a dicha marca.

Es una práctica común cuando queremos desplazarnos dentro de una página de gran tamaño. La sintaxis para definir un ancla es:

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

No debemos confundir un ancla con un hipervínculo, más allá que se utiliza el mismo elemento a. Para un ancla inicializamos la propiedad name con el nombre del ancla.

El ancla es definida en la parte de la página que queremos que el operador llegue a partir de un hipervínculo.

```
<a href="#nombreancla">Ir a Sección III</a>
```

La definición del ancla se hace inmediatamente anterior al título donde queremos que el navegador se sitúe

# Anclas llamadas desde otra página

También es perfectamente válido la llamada a anclas desde otra página (no importa si se encuentra en el mismo sitio o en otro)

Debemos conocer el nombre de la página a llamar y el nombre del ancla, luego la sintaxis para la llamada al ancla es:

```
<a href="pagina2.html#introduccion">Introducción</a>
```

Es decir luego del nombre de la página que llamamos disponemos el carácter # y seguidamente el nombre del ancla.

# Listas

En ocasiones, es posible agrupar determinadas palabras o frases en un conjunto de elementos que tienen más significado de forma conjunta. El menú de navegación de un sitio web por ejemplo está formado por un grupo de palabras. Aunque cada palabra por separado tiene sentido, de forma conjunta constituyen el menú de navegación de la página, por lo que su significado conjunto es mayor que por separado.

El lenguaje HTML define tres tipos diferentes de listas para agrupar los elementos:

**Listas no ordenadas:** Se trata de una colección simple de elementos en la que no importa su orden.

**Listas ordenadas:** Similar a la anterior, pero los elementos están numerados y por tanto, importa su orden.

**Listas de definición:** Un conjunto de términos y definiciones similar a un diccionario.



# Listas no ordenadas

Las listas no ordenadas son las más sencillas y las que más se utilizan. Una lista no ordenada es un conjunto de elementos relacionados entre sí pero para los que no se indica un orden o secuencia determinados. La etiqueta `<ul>` encierra todos los elementos de la lista y la etiqueta `<li>` cada uno de sus elementos.

```
<h2>Lista No Ordenada</h2>
<ul>
  <li>Inicio</li>
  <li>Noticias</li>
  <li>Articulos</li>
  <li>Contacto</li>
</ul>
```

## Lista No Ordenada

- Inicio
- Noticias
- Articulos
- Contacto

# Listas ordenadas

La lista ordenada, que se define mediante la etiqueta `<ol>`. Los elementos de la lista se definen mediante la etiqueta `<li>`, la misma que se utiliza en las listas no ordenadas.

```
<h2>Lista Ordenada</h2>
<ol>
  <li>Enchufar correctamente</li>
  <li>Comprobar conexiones</li>
  <li>Encender el aparato</li>
</ol>
```

## Lista Ordenada

- Enchufar correctamente
- Comprobar conexiones
- Encender el aparato

El navegador muestra la lista de forma muy parecida a las listas no ordenadas, salvo que en este caso no se emplean viñetas gráficas en los elementos, sino que se numeran de forma consecutiva. El tipo de numeración empleada también se puede modificar aplicando hojas de estilos CSS a los elementos de la lista.

# Listas de definición

Las listas de definición apenas se utilizan en la mayoría de páginas HTML. Su funcionamiento es similar al de un diccionario, ya que cada elemento de la lista está formado por términos y definiciones. La etiqueta `<dl>` crea la lista de definición y las etiquetas `<dt>` y `<dd>` definen respectivamente el término y la descripción de cada elemento de la lista.

```
<dl>
  <dt>C++</dt>
  <dd>
    Es un lenguaje de programación, diseñado a mediados de los años 1980,
    por Bjarne Stroustrup, como extensión del lenguaje de programación C.
  </dd>
  <dt>Java</dt>
  <dd>
    Es un lenguaje de programación orientado a objetos desarrollado por Sun
    Microsystems a principios de los 90.
  </dd>
  <dt>JavaScript</dt>
  <dd>
    Es un lenguaje interpretado, es decir, que no requiere compilación,
    utilizado principalmente en páginas web, con una sintaxis semejante a la
    del lenguaje C.
  </dd>
</dl>
```

C++

Es un lenguaje de programación, diseñado a mediados de los años 1980, por Bjarne Stroustrup, como extensión del lenguaje de programación C.

Java

Es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los 90.

JavaScript

Es un lenguaje interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web, con una sintaxis semejante a la del lenguaje C.

# Imágenes

Para insertar una imagen dentro de una página debemos utilizar el elemento HTML **<img>**, la misma no tiene una marca de finalización (similar a la etiqueta **<br>**).

Generalmente, la imagen se encuentra en el mismo servidor donde se almacenan nuestras páginas HTML. Los formatos clásicos son los archivos con extensiones gif, jpg y png.

```

```

**<img>** viene de la palabra **image**

**src** viene de la palabra **source**

**alt** viene de la palabra **alternative**



# Tablas

Desde sus primeras versiones, HTML incluyó el soporte para crear tablas de datos en las páginas web. Además de ser sencillo, el modelo definido por HTML es muy flexible y bastante completo. Las tablas en HTML utilizan los mismos conceptos de filas, columnas, cabeceras y títulos que los que se utilizan en cualquier otro entorno de publicación de documentos

The diagram illustrates the components of an HTML table. It features a table titled "Cursos de diseño gráfico" with four columns: "Nombre", "Horas", "Plazas", and "Horario". The first row is the header row, and the following three rows are data rows. Labels with arrows point to various parts of the table: "título de tabla" points to the title, "cabecera de columna" points to the header row, "cabecera de tabla" points to the first column, "fila" points to a data row, "cabecera de fila" points to the first cell of a data row, and "columna" points to a data column.

Nombre	Horas	Plazas	Horario
Introducción a XHTML	20	20	09:00 – 13:00
CSS avanzado	40	15	16:00 – 20:00
Taller de usabilidad	40	10	16:00 – 20:00
Introducción a AJAX	60	20	08:30 – 12:30

# Tablas básicas

Las tablas más sencillas de HTML se definen con tres etiquetas:

**<table>** para crear la tabla.

**<tr>** para crear cada fila .

**<td>** para crear cada columna.

País	Habitantes
China	1300 millones
India	1080 millones
EEUU	295 millones

La etiqueta <table> encierra todas las filas y columnas de la tabla.

Las etiquetas <tr> (del inglés *"table row"*) definen cada fila de la tabla y encierran todas las columnas. Por último, la etiqueta <td> (del inglés *"table data cell"*) define cada una de las columnas de las filas, aunque realmente HTML no define columnas sino *celdas de datos*.

```
<table border="1px">
  <tr>
    <th>País</th>
    <th>Habitantes</th>
  </tr>
  <tr>
    <td>China</td>
    <td>1300 millones</td>
  </tr>
  <tr>
    <td>India</td>
    <td>1080 millones</td>
  </tr>
  <tr>
    <td>EEUU</td>
    <td>295 millones</td>
  </tr>
</table>
```



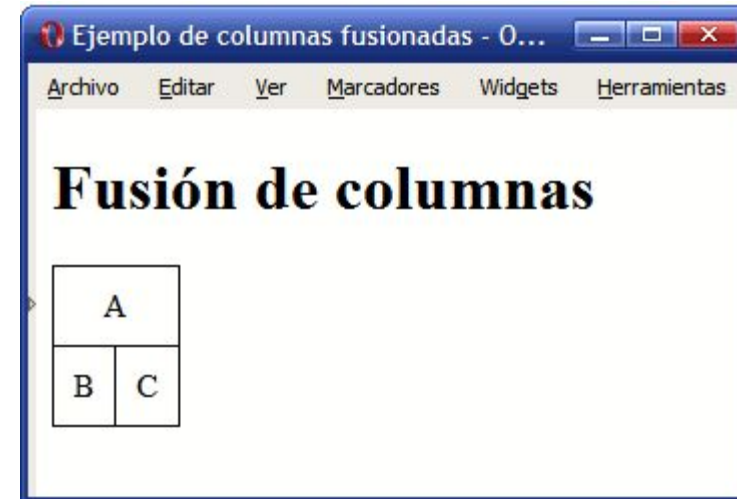
# Tablas complejas

Las tablas complejas suelen disponer de una estructura irregular que junta varias columnas para formar una columna ancha o une varias filas para formar una fila más alta que las demás. Para fusionar filas o columnas, se utilizan los atributos `rowspan` y `colspan` respectivamente.



The screenshot shows a web browser window with the title "Ejemplo de filas fusionadas - Opera". The browser's menu bar includes "Archivo", "Editar", "Ver", "Marcadores", "Widgets", and "Herramientas". The main content area displays the heading "Fusión de filas" above a table. The table has two columns and two rows. The first column contains the letters "A" and "C". The second column contains the letter "B". This visualizes a table where the second column spans two rows.

A	B
C	



The screenshot shows a web browser window with the title "Ejemplo de columnas fusionadas - O...". The browser's menu bar includes "Archivo", "Editar", "Ver", "Marcadores", "Widgets", and "Herramientas". The main content area displays the heading "Fusión de columnas" above a table. The table has two rows and two columns. The first row contains the letter "A". The second row contains the letters "B" and "C". This visualizes a table where the first row spans two columns.

A	
B	C

# Tablas complejas II

Como la segunda columna de la tabla ocupa el espacio de las dos filas, el código HTML debe indicar claramente que esa celda va a ocupar dos filas, de manera que todas las columnas de la tabla cuenten con el mismo número de filas.

```
<table>
  <tr>
    <td>A</td>
    <td rowspan="2">B</td>
  </tr>

  <tr>
    <td>C</td>
  </tr>
</table>
```

La primera fila de la tabla está formada sólo por una columna, mientras que la segunda fila está formada por dos columnas.

```
<table>
  <tr>
    <td colspan="2">A</td>
  </tr>

  <tr>
    <td>B</td>
    <td>C</td>
  </tr>
</table>
```

# Ejercicio 1

Ejemplo de columnas fusionadas - Opera

Archivo Editar Ver Marcadores Widgets Herramientas Ayuda

## Fusión de columnas

A		B	
C	D	E	
F			
G	H	I	J

# Ejercicio 2

Ejemplo de filas fusionadas - Opera

Archivo Editar Ver Marcadores Widgets Herramientas Ayuda

## Fusión de filas

A	B		D
E	F	C	G
	H		
I	J	K	

# Tablas Semánticas

Las partes que componen las tablas complejas se definen mediante las etiquetas **<thead>**, **<tbody>** y **<tfoot>**. La cabecera de la tabla se define con la etiqueta **<thead>**, el pie de la tabla se define mediante **<tfoot>** y cada sección de datos se define con una etiqueta **<tbody>**.

# Formularios

HTML es un lenguaje de marcado cuyo propósito principal consiste en estructurar los contenidos de los documentos y páginas web. Sin embargo, HTML también incluye elementos para crear aplicaciones web. El estándar HTML permite crear formularios para que los usuarios interactúen con las aplicaciones web. Los formularios más sencillos se pueden crear utilizando solamente dos etiquetas: **<form>** y **<input>**. Si se considera el formulario que muestra la siguiente imagen:

## Formulario Básico

La etiqueta **<form>** encierra todos los contenidos del formulario (botones, cuadros de texto, listas desplegables) y la etiqueta **<input>** permite definir varios tipos diferentes de elementos (botones y cuadros de texto).



# Formularios atributos

La mayoría de formularios utilizan sólo los atributos **action** y **method**. El atributo **action** indica la URL de la aplicación del servidor que se encarga de procesar los datos introducidos por los usuarios. Esta aplicación también se encarga de generar la respuesta que muestra el navegador.

El atributo **method** establece la forma en la que se envían los datos del formulario al servidor. Este atributo hace referencia al método HTTP, por lo que no es algo propio de HTML. Los dos valores que se utilizan en los formularios son **GET** y **POST**. De esta forma, casi todos los formularios incluyen el atributo **method="get"** o el atributo **method="post"**.

```
<h2>Formulario Básico</h2>

<form action="#" method="GET">

  <input type="text" placeholder="Ingrese nombre" /><br><br>
  <input type="button" value="Enviar" />

</form>
```

# Controles

Los elementos de formulario como botones y cuadros de texto también se denominan "*campos de formulario*" y "*controles de formulario*". El más común es el control input

- **type** = "text | password | checkbox | radio | submit | reset | file | hidden | image | button" - Indica el tipo de control que se incluye en el formulario
- **name** = "texto" - Asigna un nombre al control (es imprescindible para que el servidor pueda procesar el formulario)
- **value** = "texto" - Valor inicial del control
- **size** = "unidad\_de\_medida" - Tamaño inicial del control (para los campos de texto y de password se refiere al número de caracteres, en el resto de controles se refiere a su tamaño en píxel)
- **maxlength** = "numero" - Máximo número de caracteres para los controles de texto y de password
- **checked** = "checked" - Para los controles checkbox y radiobutton permite indicar qué opción aparece preseleccionada
- **disabled** = "disabled" - El control aparece deshabilitado y su valor no se envía al servidor junto con el resto de datos
- **readonly** = "readonly" - El contenido del control no se puede modificar
- **src** = "url" - Para el control que permite crear botones con imágenes, indica la URL de la imagen que se emplea como botón de formulario
- **alt** = "texto" - Descripción del control

# Cuadro de Texto

Se trata del elemento más utilizado en los formularios. En el caso más sencillo, se muestra un cuadro de texto vacío en el que el usuario puede escribir cualquier texto:

Nombre

A continuación se muestra el código HTML correspondiente al ejemplo anterior:

Nombre `<br/> <input type="text" name="nombre" value="" />`

El atributo `type` diferencia a cada uno de los diez controles que se pueden crear con la etiqueta `<input>`. Para los cuadros de texto, su valor es `text`. El atributo `name` es el más importante en los campos del formulario. De hecho, si un campo no incluye el atributo `name`, sus datos no se envían al servidor. El valor que se indica en el atributo `name` es el nombre que utiliza la aplicación del servidor para obtener el valor de este campo de formulario.

# Cuadro de Contraseña

La única diferencia entre este control y el cuadro de texto normal es que el texto que el usuario escribe en un cuadro de contraseña no se ve en la pantalla. En su lugar, los navegadores ocultan el texto utilizando asteriscos o círculos, por lo que es ideal para escribir contraseñas y otros datos sensibles.



Contraseña `<br/> <input type="password" name="contrasena" value="" />`

Cambiando el valor del atributo type por password se transforma el cuadro de texto normal en un cuadro de contraseña. Todos los demás atributos se utilizan de la misma forma y tienen el mismo significado.

# Checkbox

Los checkbox o "*casillas de verificación*" son controles de formulario que permiten al usuario seleccionar y deseleccionar opciones individualmente. Aunque en ocasiones se muestran varios checkbox juntos, cada uno de ellos es completamente independiente del resto. Por este motivo, se utilizan cuando el usuario puede activar y desactivar varias opciones relacionadas pero no excluyentes.

Puestos de trabajo buscados <br/>

<input name="puesto\_directivo" type="checkbox" value="direccion"/> Dirección

<input name="puesto\_tecnico" type="checkbox" value="tecnico"/> Técnico

<input name="puesto\_empleado" type="checkbox" value="empleado"/>

Empleado

Si se quiere mostrar un *checkbox* seleccionado por defecto, se utiliza el atributo checked. Si el valor del atributo es checked, el *checkbox* se muestra seleccionado.

Puestos de trabajo buscados

☐ Dirección

☐ Técnico

☐ Empleado

<input type="checkbox" checked="checked" ... /> Checkbox seleccionado por defecto



# Radiobutton

Los controles de tipo radiobutton son similares a los controles de tipo checkbox, pero presentan una diferencia muy importante: son mutuamente excluyentes. Los radiobutton se utilizan cuando el usuario solamente puede escoger una opción entre las distintas opciones relacionadas que se le presentan. Cada vez que se selecciona una opción, automáticamente se deselecta la otra opción que estaba seleccionada.

Sexo <br/>

```
<input type="radio" name="sexo" value="hombre" checked="checked" />
```

Hombre

```
<input type="radio" name="sexo" value="mujer" /> Mujer
```

Sexo

☒ Hombre

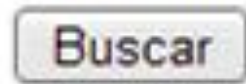
☐ Mujer

El valor del atributo type para estos controles de formulario es radio. El atributo name se emplea para indicar los radiobutton que están relacionados. Por lo tanto, cuando varios *radiobutton* tienen el mismo valor en su atributo name, el navegador sabe que están relacionados y puede deselectar una opción del grupo de *radiobutton* cuando se seleccione otra opción.



# Botón de envío de formulario

La mayoría de formularios dispone de un botón para enviar al servidor los datos introducidos por el usuario:

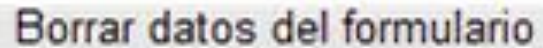


```
<input type="submit" name="buscar" value="Buscar" />
```

El valor del atributo type para este control de formulario es submit. El navegador se encarga de enviar automáticamente los datos cuando el usuario pincha sobre este tipo de botón. El valor del atributo value es el texto que muestra el botón. Si no se establece el atributo value, el navegador muestra el texto predefinido Enviar consulta.

# Botón de reseteo del formulario

Aunque su uso era muy popular hace unos años, la mayoría de formularios modernos ya no utilizan este tipo de botón. Se trata de un botón especial que borra todos los datos introducidos por el usuario y devuelve el formulario a su estado original:

A rectangular button with rounded corners, a light gray background, and a thin gray border. The text "Borrar datos del formulario" is centered on the button in a dark gray, sans-serif font.

El valor del atributo `type` para este control de formulario es `reset`. Cuando el usuario pulsa este botón, el navegador borra toda la información introducida y muestra el formulario en su estado original. Si el formulario no contenía originalmente ningún valor, el botón de reset lo vuelve a mostrar vacío. Si el formulario contenía información, el botón reset vuelve a mostrar la misma información original.

Como es habitual en los botones de formulario, el atributo `value` permite establecer el texto que muestra el botón. Si no se utiliza este atributo, el navegador muestra el texto predefinido del botón, que en este caso es `Restablecer`.

# Ficheros adjuntos

Los formularios también permiten adjuntar archivos para subirlos al servidor. Aunque desde el punto de vista de HTML y del navegador no existe ninguna limitación sobre el número, tipo o tamaño total de los archivos que se pueden adjuntar, todos los servidores añaden restricciones por motivos de seguridad.

Fichero adjunto

```
<input type="file" name="adjunto" />
```

Fichero adjunto



El valor del atributo type para este control de formulario es file. El navegador se encarga de mostrar un cuadro de texto donde aparece el nombre del archivo seleccionado y un botón que permite navegar por los directorios y archivos del ordenador del usuario.

Si se incluye un control para adjuntar archivos, es obligatorio añadir el atributo enctype en la etiqueta <form> del formulario. El valor del atributo enctype debe ser multipart/form-data, por lo que la etiqueta <form> de los formularios que permiten adjuntar archivos siempre es:

```
<form action="..." method="post" enctype="multipart/form-data"> ... </form>
```

# Campos ocultos

Los campos ocultos se emplean para añadir información oculta en el formulario:

```
<input type="hidden" name="url_previa" value="/articulo/primer.html" />
```

El valor del atributo type para este control de formulario es hidden. Los campos ocultos no se muestran por pantalla, de forma que el usuario desconoce que el formulario los incluye. Normalmente los campos ocultos se utilizan para incluir información que necesita el servidor pero que no es necesario o no es posible que la establezca el usuario.

# Botón

Algunos formularios complejos necesitan botones más avanzados que los de enviar datos (`type="submit"`) y resetear el formulario (`type="reset"`). Por ese motivo, el estándar HTML/XHTML define un botón de tipo genérico:



```
<input type="button" name="guardar" value="Guardar Cambios" />
```

El valor del atributo `type` para este control de formulario es `button`. Si pruebas a pulsar un botón de este tipo, verás que el navegador no hace nada: no envía los datos al servidor y no borra los datos introducidos. Este tipo de botones sólo son útiles si se utilizan junto con el lenguaje de programación JavaScript. Si la página incluye código JavaScript, los botones de este tipo se pueden programar para que realicen cualquier tarea compleja cuando se pulsa sobre ellos.



# Select

El elemento **select** es un cuadro de selección.

Este elemento HTML nos permite seleccionar una opción entre un conjunto.

```
<h2>Lenguajes</h2>
<select name="lenguaje">
  <option value="1">Javascript</option>
  <option value="2">PHP</option>
  <option value="3">Ruby</option>
  <option value="4">Java</option>
  <option value="5">Python</option>
</select>
```

Lenguajes

Javascript ▼

Lenguajes

Javascript  
PHP  
Ruby  
Java  
Python

El elemento **option** define el texto a mostrar y en la propiedad **value** indica el valor a enviar al servidor en caso de estar seleccionada dicha opción.



# Selección múltiple y grupos

Si se necesitan seleccionar varias opciones se puede utilizar el atributo **multiple** y en la propiedad name se debe usar un array

```
<select name="lenguaje">
  <optgroup label="Frontend">
    <option value="1">Html</option>
    <option value="2">CSS</option>
    <option value="3">Javascript</option>
  </optgroup>
  <optgroup label="Backend">
    <option value="4">PHP</option>
    <option value="5">Ruby</option>
    <option value="6">Python</option>
  </optgroup>
</select>
```

Frontend
Html
CSS
Javascript
Backend
PHP
Ruby
Python

Podemos agrupar las opciones que tiene el cuadro de selección, esto tiene sentido si el cuadro de selección tiene muchos items.

Se cuenta con un nuevo elemento llamado **optgroup** que agrupa un conjunto de elementos option.

La propiedad label del elemento optgroup aparece dentro del control select pero no se puede seleccionar, es un título.

# Textarea

El elemento de tipo **textarea** nos permite el ingreso de varias líneas a diferencia del cuadro de texto (input/text)

Es muy utilizado cuando queremos ingresar un comentario de una longitud de caracteres grande.

La sintaxis para definir un área de texto para el ingreso de múltiples líneas es:

```
<textarea name="comentarios" rows="5" cols="60"></textarea>
```

Además de tener la propiedad **name** similar a los otros elementos relacionados a formularios tiene dos propiedades llamadas **rows** y **cols**. Estas dos propiedades indican la cantidad de filas y columnas que visualiza el área de texto.

# Agrupamiento de controles

El HTML dispone de un elemento llamado **fieldset** que solo tiene el objetivo de recuadrar y agrupar un conjunto de controles de un formulario.

Debemos encerrar todos los controles a agrupar entre las marcas **<fieldset>** y **</fieldset>**. Además para agregar un título a este recuadro debemos agregar otro elemento HTML llamado **legend**.

Datos Personales

Nombre:

Apellido:

```
<fieldset>
  <legend>Datos Personales</legend>
  <label>Nombre:<input type="text" id="txtNombre"/></label><br /><br />
  <label>Apellido:<input type="text" id="txtApellido"/></label><br />
</fieldset>
```

# Label

Hasta este momento siempre que queríamos disponer un mensaje antes o después de un control de formulario lo escribíamos sin más.

Existe en HTML un elemento que permite asociar un texto con un control de formulario.

Veamos como lo hacíamos hasta ahora:

Ingrese su nombre: `<input type="text" name="nombre" size="20">`

Utilizando el elemento label podemos hacer una referencia entre el texto y el control de entrada de datos:

```
<label for="nombre">Ingrese su nombre:</label>  
<input type="text" name="nombre" size="20" id="nombre">
```

La propiedad for de la label hace referencia al id del control y no al name.

# Div y Span

Estas marcas se utilizan en conjunción con las hojas de estilo. Sin estas tiene poco sentido el empleo de estos elementos HTML.

Estos elementos "div" y "span" nos permiten agrupar un conjunto de elementos y aplicar reglas de estilo. La diferencia entre estos dos elementos es que cuando utilizamos el elemento div produce un salto de línea previo y uno al final, es decir es una marca de bloque como lo son h1,h2,p etc. En cambio el elemento span no produce un salto de línea porque se trata de un elemento en línea como lo son a,em,strong,input etc.

div es un elemento en bloque.

span es un elemento en línea.



# Audio

El comité de estandarización W3C deja abierto a cada empresa que desarrolla navegadores los formatos que quieran soportar (así tenemos que algunos soportan mp3, wav, ogg, au)

```
<audio src="sonido.ogg" autoplay controls loop></audio>
```

Las propiedades que podemos utilizar con la marca audio son:

- src**: La URL donde se almacena el archivo de audio. Si no definimos la URL la busca en el mismo directorio donde se almacena la página.
- autoplay**: En caso de estar presente el archivo se ejecuta automáticamente luego de cargarse la página sin requerir la intervención del visitante.
- loop**: El archivo de audio se ejecuta una y otra vez.
- controls**: Indica que se deben mostrar la interface visual del control en la página (este control permite al visitante arrancar el audio, detenerlo, desplazarse etc.)
- autobuffer**: En caso de estar presente indica que primero debe descargarse el archivo en el cliente antes de comenzar a ejecutarse.



# Audio II

Como no hay un formato de audio universalmente adoptado por todos los navegadores el elemento audio nos permite agregarle distintas fuentes:

```
<audio controls autoplay loop>  
<source src="sonido.ogg">  
<source src="sonido.mp3">  
<source src="sonido.wav">  
<source src="sonido.au">  
</audio>
```

El elemento source indica a través de la propiedad src la ubicación del archivo de audio respectivo. El orden en que disponemos estas fuentes es importante. Primero el navegador busca la primera fuente y verifica que puede reproducir dicho archivo, en caso negativo pasa a la siguiente fuente.

En el caso que solo necesitemos reproducir un único formato de archivo podemos evitar los elementos HTML "source" y disponer la propiedad src directamente en la etiqueta "audio"

# Video

El elemento VIDEO permite mostrar un video sin la necesidad de plugin (Flash). En este momento los navegadores permiten mostrar formatos como el mp4, webm y ogv.

FireFox permite mostrar videos en formato ogv (formato de vídeo de código abierto Ogg/Theora).

Luego para visualizar un video con este formato en FireFox tenemos:

```
<video width="640"height="360"src="http://videos.mozilla.org/firefox/3.5/overview/overview.ogv" controls>
```

Este navegador no permite tag video

```
</video>
```

# Video II

Las propiedades más importantes de la marca video son:

- src**: Dirección donde se almacena el video.
- controls**: Se visualiza el panel de control del video: botón de inicio, barra de avance del video etc.
- autoplay**: El video se inicia inmediatamente luego que la página se carga en el navegador.
- width**: Ancho en píxeles del video.
- height**: Alto en píxeles del video.

Como no hay un formato de video universalmente adoptado por todos los navegadores el elemento video nos permite agregarle distintas fuentes:

```
<video width="640" height="360" controls>  
<source src="http://videos.mozilla.org/firefox/3.5/overview/overview.ogv">  
<source src="http://videos.mozilla.org/firefox/3.5/overview/overview.mp4">  
</video>
```

Esto es similar al elemento AUDIO visto anteriormente.