

HTML5

UTN FRA

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 contains the headers for each column. The subsequent three rows list specific courses: "Introducción a XHTML", "CSS avanzado", "Taller de usabilidad", and "Introducción a AJAX". Arrows point to various parts of the table with labels: "título de tabla" points to the title, "cabecera de columna" points to a column header, "cabecera de tabla" points to the header row, "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>
```

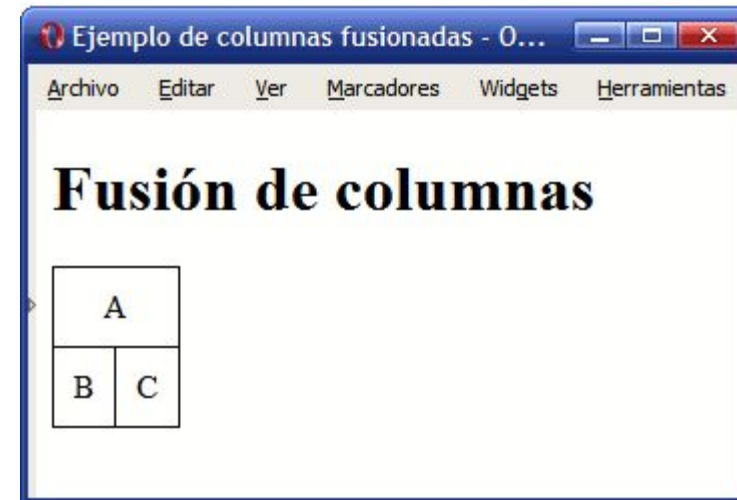
Tablas complejas

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" followed by 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". The "B" in the second column is vertically aligned with the "A" in the first column, indicating it 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" followed by a table. The table has two rows and one column. The first row contains the letter "A". The second row contains the letters "B" and "C". The "A" in the first row is horizontally aligned with the "B" and "C" in the second row, indicating it 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	C	D
E	F		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 `
 <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

Contraseña `
 <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

<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 deselectiona la otra opción que estaba seleccionaba.

Sexo


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

Hombre

```
<input type="radio" name="sexo" value="mujer" /> 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 deselectionar una opción del grupo de *radiobutton* cuando se seleccione otra opción.

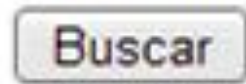
Sexo

☒ Hombre

☐ Mujer

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 a light gray background and a thin 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>
```

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.

Lenguajes

Javascript ▼

Lenguajes

Javascript
PHP
Ruby
Java
Python

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.