Clase 02 - Primeros pasos con HTML

# Enlaces

## Arquitectura de un sitio

Los enlaces, también conocidos como links o anchors, se utilizan para relacionar partes del mismo documento. Por defecto, se visualizan azules y subrayados.

Para crear uno, es necesario utilizar la etiqueta de ancla <a> con el atributo “href”, que establecerá el destino al que apunta. Por ejemplo:

|  |
| --- |
| <a href="productos.html">Productos</a> |

## Enlaces absolutos, relativos e internos

### Enlaces absolutos

Los enlaces absolutos son aquellos cuyo destino apunta a un documento que está fuera del sitio, y debe ser especificado utilizando la URL completa:

|  |
| --- |
| <a href="http://www.coderhouse.com/frontend">Curso de Frontend</a> |

### Enlaces relativos

Los enlaces relativos son aquellos que apuntan a páginas ubicadas dentro del mismo proyecto. Si la página referenciada se encuentra en el mismo directorio, alcanza con mencionar el nombre de la misma para generar el enlace.

|  |
| --- |
| <a href="contacto.html">Contacto</a> |

En caso de que el archivo se encuentre en un directorio específico, el mismo deberá ser mencionado.

|  |
| --- |
| <a href="imagenes/mapa.jpg">ver mapa</a> |

También puedes usar como destino una sección específica una página distinta:

|  |
| --- |
| <a href="contacto.html#formulario">Formulario de contacto</a> |

En el ejemplo anterior, el enlace apunta a la sección que tiene el id formulario dentro de la página “contacto.html”. No sólo es posible agregar enlaces a texto, también puedes hacerlo con otros elementos. Por lo general, se usan textos o imágenes. Veamos un ejemplo de enlaces con una imagen:

|  |
| --- |
| <a href="http://www.coderhouse.com/cursos.html#frontend">  <img src="img/logo\_coderhouse.png" alt="coderhouse"/>  </a> |

### Enlaces internos

Los enlaces internos permiten referenciar secciones de la misma página, para lo cual se utiliza el id:

|  |
| --- |
| <a href="#pie">Ir al pie de página</a>  ...  <footer id="pie"></footer> |

# Multimedia en HTML

## Imágenes

Enriquecen el HTML: las imágenes son elementos que, bien utilizados, mejoran la experiencia de los usuarios.

### Insertar imágenes

📌 Se insertan con la etiqueta <img/>, que pertenece al grupo de las etiquetas que se cierran a sí mismas (con la barra al final).

📍 Para funcionar requiere, como mínimo, indicar en dónde está el archivo a mostrar. Eso se hace con el atributo “src” (el source o fuente), que respeta todas las reglas de ruteo vistas en los links.

👥Se comportan como elementos de línea, esto significa que se verán una al lado de la otra.

### Atributo *“alt”*

El “alt” es un texto que debe representar la foto que se está visualizando. Tiene que ser conciso y breve, pero dejar en claro de qué se trata la imagen.

|  |
| --- |
| <img src="smiley.gif" alt="Smiley Cara" /> |

### Favicon

El favicon es un pequeño ícono que identifica un sitio web cuando está abierto en una pestaña o cuando es guardado como favorito.

HTML recomendado para agregar dentro de tu etiqueta <head>:

|  |
| --- |
| <link rel="shortcut icon" href="favicon.png"> |

## Etiqueta iframe

Es un elemento HTML que permite insertar o incrustar un documento HTML dentro de un documento HTML principal.

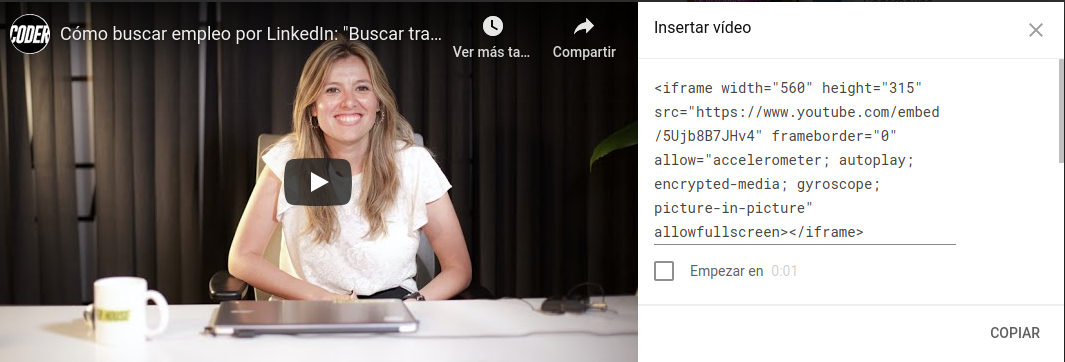
|  |
| --- |
| <iframe src="pagina\_fuente.html" width=290 height=250>Texto para cuando el navegador no conoce la etiqueta iframe</iframe> |

Los frames (frame en inglés significa marco) son unas herramientas que han tenido una historia dilatada en el desarrollo de páginas web con HTML. De ser una etiqueta no estándar ha pasado a ser soportada por todos los navegadores y formar parte de las especificaciones de HTML, para luego retirarse de nuevo del estándar en HTML5. No obstante, ha permanecido en uso y dentro del estándar una etiqueta hermana IFRAME que vamos a ver en este artículo, y que todavía hoy tiene mucha utilidad.

En concreto, iframe sirve para crear un espacio dentro de la página donde se puede incrustar otra web. Es un cuadrado cuyas dimensiones debe especificar el desarrollador en la propia página, incluidas por los atributos width y height en la propia etiqueta IFRAME.

El iframe tiene asociada una página web, que se carga en el espacio y operará de manera totalmente independiente. Esa página web tendrá sus propios contenidos y estilos. Además será perfectamente funcional: si tiene enlaces se mostrarán en ese mismo espacio y si tiene scripts o aplicaciones dentro se ejecutarán también de manera autónoma en el espacio reservado al iframe.

**Ejemplo**: Si deseo insertar un video de YouTube, me voy a la sección de **Compartir** y en **Insertar**, YouTube me proporciona el **iframe** a usar.



## Audio y video

HTML5 soporta contenido multimedia gracias a los elementos **<audio>** y **<video>**  
Insertar contenido multimedia en tus documentos HTML es muy sencillo:

**Para insertar un video video usaremos la siguiente sintaxis:**

|  |
| --- |
| <video src="videos/tutorial.ogg" controls>  Tu navegador no implementa el elemento <code>video</code>.  </video> |

**De la misma forma insertamos audio con el siguiente código:**

|  |
| --- |
| <audio src="audio.ogg">  <p>Tu navegador no implementa el elemento audio.</p>  </audio> |

El atributo src puede ser una URL del archivo de audio o la ruta al archivo en el sistema local.

### Atributos de la etiqueta <audio>

* Controls: muestra los controles estándar de HTML5 para audio en una página web.
* Autoplay: hace que el audio se reproduzca automáticamente.
* Loop: hace que el audio se repita automáticamente.
* Preload: es usado en el elemento audio para almacenar temporalmente (buffering) archivos de gran tamaño. Este puede tomar uno de 3 valores:  
  *"none"* no almacena temporalmente el archivo  
  *"auto"* almacena temporalmente el archivo multimedia  
  *"metadata"* almacena temporalmente sólo los metadatos del archivo

Se pueden especificar múltiples fuentes de archivos usando el elemento **<source>** con el fin de proporcionar vídeo o audio codificados en formatos diferentes para diferentes navegadores. Por ejemplo:

|  |
| --- |
| <video controls>  <source src="foo.ogg" type="video/ogg">  <source src="foo.mp4" type="video/mp4">  Tu navegador no implementa el elemento <code>video</code>.  </video> |

Esto reproduce el archivo Ogg en navegadores que admiten el formato Ogg. Si el navegador no admite Ogg, el navegador usará el archivo MPEG-4.

Listas

HTML permite agrupar 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. Esto se denomina lista.

## Tipos de listas

* Listas no ordenadas
* Listas ordenadas
* Listas de definición

**Ejemplos:** paso a paso de un procedimiento (como una receta de cocina), características de una persona, galería de imágenes, el menú de una página web, entre otros.

### ¿Viñetas o números?

* Las listas numéricas establecen un orden en la lectura de sus ítems.
* Las listas de viñetas no representan ningún orden o importancia entre sus ítems. Son elementos compuestos.

**<ol>** define una lista ordenada de artículos (numéricas).

**<ul>** define una lista de artículos sin orden (viñetas).

**<li>** define un artículo de una lista.

### Veamos un ejemplo

Ambas listas se deben insertar mediante la etiqueta <li></li> (list-item). Ejemplo de servicios de una empresa (lista de viñetas/sin orden):

|  |
| --- |
| <ul>  <li>Empresa</li>  <li>Producto</li>  <li>Servicios</li>  <li>Contacto</li>  </ul> |

## Anidar listas

**Es probable que te veas en la necesidad de crear una estructura de sublistas como la siguiente:**

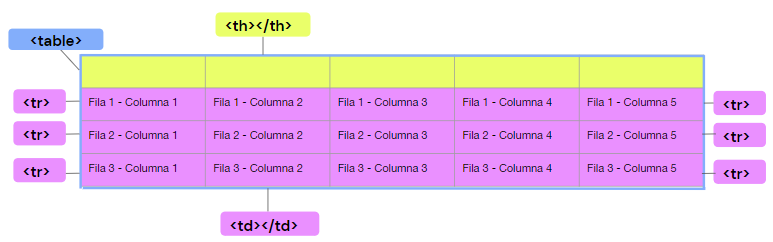
* Computadoras portátiles:
* Procesador I4.
* Procesador I5.
* Computadoras de escritorio:
* Procesador Pentium.
* Procesador Celeron.

# Tablas

Son un conjunto de celdas organizadas, dentro de las cuales es posible alojar distintos contenidos. HTML dispone de una gran variedad de etiquetas y atributos para crear tablas. Sirven para representar información tabulada, en filas y columnas.

En HTML4 las tablas se usaban para maquetar. Cuando CSS creció y se hizo más fuerte, nacieron los detractores de las tablas.

## Etiquetas básicas para tablas en HTML

****

|  |
| --- |
| <table>  <tr><!-- inicio de fila-->  <td>Fila 1 - Columna 1</td>  <td>Fila 1 - Columna 2</td>  <td>Fila 1 - Columna 3</td>  </tr><!-- cierre de fila -->  <tr><!-- inicio de otra fila-->  <td>Fila 2 - Columna 1</td>  <td>Fila 2 - Columna 2</td>  <td>Fila 2 - Columna 3</td>  </tr><!-- cierre de la segunda fila -->  </table> |

**Acepta 3 atributos de “diseño”:**

* **Borde**r: bordes de la tabla.
* **Cellpadding:** especifica el espacio, en píxeles, entre la pared de la celda y su contenido.
* **Cellspacing**: indica la distancia entre las celdas y el margen exterior de la tabla

# Formularios

## Definición

Son etiquetas donde el usuario ingresará o seleccionará valores, que serán enviados a un archivo encargado de procesar la información.

## Etiqueta <form>

Para insertar un formulario se usa la etiqueta <form>, que dentro lleva todos los controles que vayan al mismo destino. Un formulario requiere 3 atributos para funcionar:

* **Action**: documento que se encarga de recibir los datos y procesarlos.
* **Method:** la forma en que será enviada la información. Existen dos métodos de envío, que son GET y POST.
* **Enctype:** cómo se codificarán los contenidos.

### Action

En este atributo se indicará cuál es el archivo que recibe y procesa los datos. Debe ser de un lenguaje de los llamados “del lado del servidor” (PHP / ASP / JSP). Si no se indica un valor, el Action será por defecto el mismo archivo donde está el formulario**.**

### Method

Forma en la que se recopilan y envían los datos. Existen dos métodos comunes en el HTML:

* GET: la información viajará por la barra de direcciones a continuación del nombre del archivo.
* POST: la información viajará junto a los encabezados del HTML (será “invisible”).

Si el method no se indica, por defecto será GET.

### Enctype

Cuando el valor del atributo method es post, el mismo es el tipo MIME del contenido, que es usado para enviar el formulario al servidor. Los posibles valores son:

* application/x-www-form-urlencoded: será el valor por defecto si un atributo no está especificado.
* multipart/form-data: usar este valor si se está usando el elemento input con el atributo type ajustado a "file".
* text/plain (HTML5)

Normalmente, se utiliza para permitir el envío de archivos a través de un formulario.

## Ingreso de texto

Existen tres controles generales para el ingreso de texto:

* Cajas de texto de una sola línea (no acepta el uso de la tecla Enter).
* Cajas para el ingreso de contraseñas (el contenido no será visible).
* Cajas para contenido multilínea. Puede ser una o muchas líneas de texto.

Atributo “name”

Control de formulario: <input>: Text, Email, Password.

Control de formulario: <textarea></textarea>

## Botones

Los botones disparan las acciones del formulario. Hay 3 tipos:

* El que envía los datos al archivo indicado como Action.
* El que vacía todo lo ingresado y resetea los campos.
* El que “no hace nada”, pensado para usarse con Javascript.

Todos los botones son etiquetas <input>, con distintos tipos de “Type”. El botón debe de estar dentro del <form> que afectará.

### Atributo “value”

Representa la etiqueta del botón, la cual es normalmente mostrada por los navegadores dentro de éste.

* Input de tipo “submit”: envía el formulario.
* Input de tipo “reset”: resetea el formulario.
* Input de tipo “button”: no tiene acciones por defecto.

|  |
| --- |
| <form>  <input type="submit" value="Enviar formulario"/>  <input type="reset" value="Limpiar formulario"/>  <input type="button" value="Sin acciones"/>  </form> |

## Controles de selección

En estos casos, el usuario no puede ingresar libremente un texto, sino que el programador le da una lista predefinida. El dato que llega al elegir una opción se define desde el atributo “value”. Existen 3 grupos de controles de selección:

* Botones de radio: sólo se puede elegir una opción.
* Casillas de chequeo: de toda la lista de opciones, el usuario puede optar por una, todas o ninguna opción.
* Menú desplegable: sólo es posible seleccionar una opción.

### Atributo “value”

En este caso es el valor que se enviará al enviarse el formulario.

* Botones de radio:

|  |
| --- |
| <form>  <div>hombre</div>  <input type="radio" name="sexo" value="hombre" />  <div>mujer</div>  <input type="radio" name="sexo" value="mujer" />  </form> |

* Casillas de chequeo:

|  |
| --- |
| <form>  <div>Acepta términos y condiciones</div>  <input type="checkbox" name="acepta" value="1" />  </form> |

## Etiqueta <label>

Define formalmente a cada elemento de un formulario. Esta etiqueta es de mucha ayuda para generar un formulario accesible.

Su principal atributo es “for”, que va a referenciar a “label” con su elemento del formulario. El valor del atributo “for” debe ser igual al valor del atributo “id” o “name” del elemento**.**

|  |
| --- |
| <form>  <label for="nombre\_apellido">Nombre:</label>  <input type="text" name="nombre\_apellido" />  </form> |

## Menú desplegable

Es el llamado combo-box, selector o menú. De toda la lista, se puede elegir una opción (aunque tiene un atributo que permite cambiarlo). Lo ideal es que sean al menos dos elementos distintos para observar el select:

|  |
| --- |
| <form>  <select name="talles">  <option value="L">Large</option>  <option value="M">Medium</option>  <option value="S">Small</option>  </select>  </form> |

### Conjunto de campo

Las etiquetas <fieldset> y <legend> se utilizan en conjunto. La primera, tiene como objetivo crear grupos de elementos del formulario que posean un mismo propósito; mientras que la segunda, define formalmente el propósito del elemento fieldset. Se estructuran de la siguiente manera:

|  |
| --- |
| <form>  <fieldset>  <legend>Talle de remera</legend>  <!-- Aquí irán los elementos de formulario -->  </fieldset>  </form> |