

Para empezar

Supongamos que quiere crear una página web que tenga un encabezado que diga "Mi sitio web" y un párrafo que diga "Bienvenidos a mi sitio web". Para hacer esto, necesitamos usar etiquetas HTML para "marcar" el contenido. Aquí está el código HTML:

```
<!DOCTYPE html>
<html>
<head>
  <title>Mi sitio web</title>
</head>
<body>
  <h1>Mi sitio web</h1>
  <p>Bienvenidos a mi sitio web</p>
</body>
</html>
```

En este ejemplo, las etiquetas **<html>**, **<head>**, y **<body>** se usan para marcar el inicio y el final de la página web, mientras que las etiquetas **<title>**, **<h1>**, y **<p>** se usan para marcar el contenido de la página. El encabezado se marca con la etiqueta **<h1>** y el párrafo se marca con la etiqueta **<p>**.

HTML utiliza etiquetas para proporcionar un significado semántico y legibilidad automática al contenido de la página. Asimismo aprenderemos sobre las jerarquías en HTML: la **cabecera (head)** y el **cuerpo (body)**, y que estas secciones pueden contener subsecciones. Además, se presenta la etiqueta **<!DOCTYPE>**, que define el tipo de documento y permite al navegador interpretar correctamente el código HTML. Por otra parte, conoceremos la estructura básica de un documento HTML5 y la estructura de una etiqueta, diferenciando así entre elemento y etiqueta.

Finalmente, se presentarán algunas excepciones en las etiquetas que no requieren una etiqueta de cierre.

¿Qué significa marcado?

Significa que con HTML podemos declarar lo **que** se presentará a un espectador, no **cómo** se presentará.

HTML cuenta con etiquetas que nos proporcionan un significado semántico y legibilidad automática al contenido de la página.

Para que la información que está en el documento HTML sea interpretada correctamente, el navegador espera que el documento siga un formato estándar, estructurado jerárquicamente por dos secciones principales: la cabecera (head) y el cuerpo (body).

Tanto el encabezado como el cuerpo, permiten anidar más secciones. Lo cual da forma a un tipo de árbol de contenido, que es la forma como el navegador lo ve.

Etiqueta `<!DOCTYPE>`

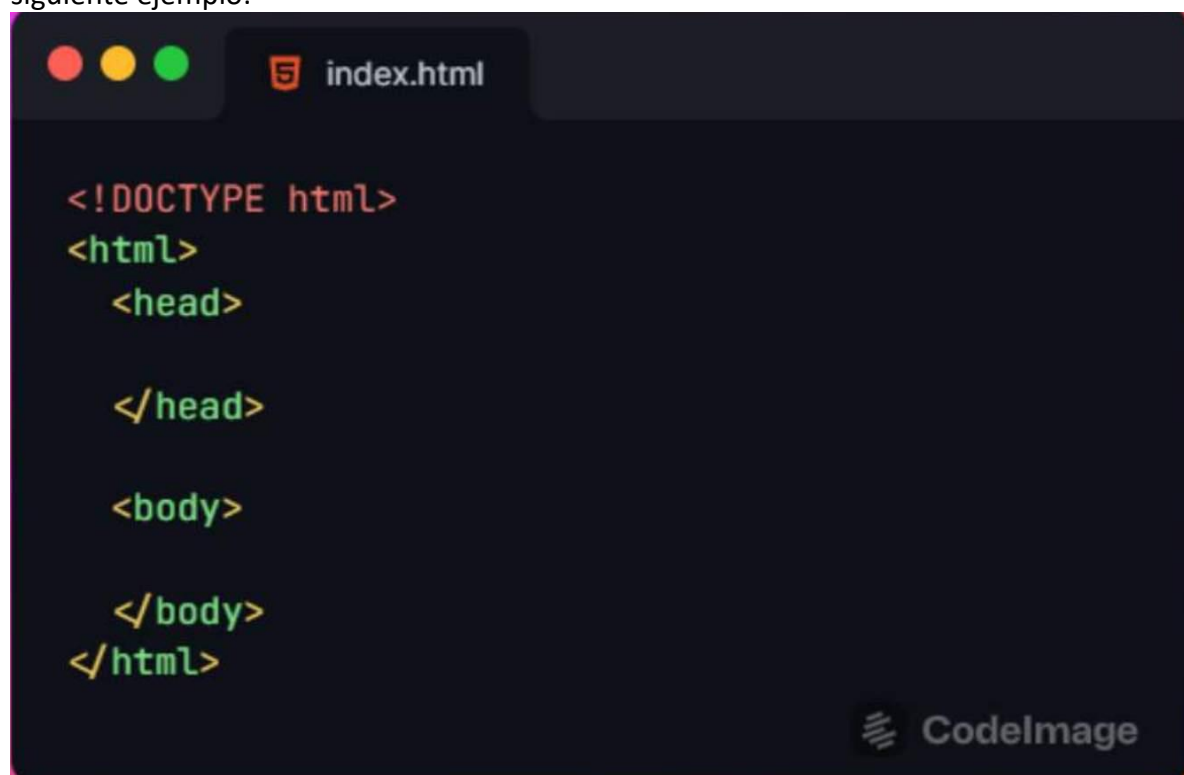
HTML, como todos los lenguajes, ha pasado por diferentes versiones con el paso del tiempo y su implementación. Por esto mismo, es necesario indicar al navegador la versión de HTML en la que se encuentran escritos los documentos que estamos intentando renderizar (mostrar en la pantalla del navegador).

Para ello, existe la etiqueta `<!DOCTYPE>` que, como su nombre lo indica, define el tipo de documento e indica cómo el navegador debe interpretar el código HTML. Para el objetivo de nuestro curso solo usaremos la etiqueta que define el tipo de documento como HTML5. La definimos de la siguiente manera:

`<!DOCTYPE>`

Estructura básica de un documento HTML5

Para definir que el documento creado es un documento con la versión 5 de HTML se debe incluir la etiqueta `<!DOCTYPE>` antes de la etiqueta de apertura `<html>`, como en el siguiente ejemplo:

A screenshot of a code editor window titled 'index.html'. The code is written in a dark theme with syntax highlighting. It shows the basic structure of an HTML5 document:

```
<!DOCTYPE html>
<html>
  <head>

  </head>

  <body>

  </body>
</html>
```

 The 'CodeImage' logo is visible in the bottom right corner of the editor.

Estructura de una etiqueta

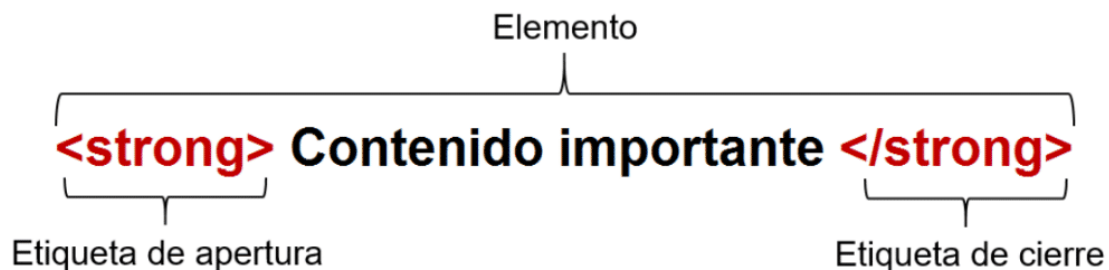
Una etiqueta es una marca o identificador que se utiliza para definir el inicio y el final de un elemento dentro de una página web.

Cada etiqueta se compone de corchetes angulares (`<` y `>`), seguidos del nombre de la etiqueta, y puede tener atributos y valores que proporcionan información adicional sobre el elemento. Las etiquetas se utilizan para definir la estructura y el contenido de una

página web, y permiten al navegador interpretar y renderizar correctamente la información que se muestra al usuario. Ejemplos de etiquetas en HTML incluyen `<html>` , `<head>` , `<body>` , `<p>` , `` , `<a>`, entre otras.

Podemos referirnos a las etiquetas como elementos, sin embargo hay una pequeña diferencia entre estos conceptos:

- Elemento : Es el bloque de HTML que ya ha sido renderizado, generalmente está conformado por dos etiquetas separadas, una de apertura y otra de cierre.
- Etiqueta : La etiqueta está escrita de una forma que el navegador puede encontrarla, rodeada de los símbolos menores que `<` y mayor que `>`.



La estructura del ejemplo anterior aplica para casi todas las etiquetas, sin embargo existen excepciones para el caso en que una etiqueta no tenga contenido para trabajar, por lo que se omite la etiqueta de cierre. Te dejamos algunas excepciones:

- `<hr>`
- `
`
- ``
- `<entrada>`
- `<enlace>`
- `<meta>`

¿Qué hemos aprendido?

En conclusión, la estructura básica de un documento HTML se divide en dos secciones principales: la cabecera y el cuerpo, las cuales pueden anidar más secciones para conformar un árbol de contenido. Para indicar al navegador la versión de HTML que estamos utilizando, utilizamos la etiqueta `<!DOCTYPE>`, la cual define el tipo de documento y cómo el navegador debe interpretar el código HTML.

Las etiquetas son palabras clave con un formato especial que rodea el contenido y aplican significado semántico y legibilidad automática al mismo. Es importante recordar que las etiquetas tienen una etiqueta de apertura y una de cierre, aunque existen excepciones cuando una etiqueta no tiene contenido para trabajar.

Con esta base, podemos comenzar a crear documentos HTML y utilizar las etiquetas necesarias para maquetar nuestra página web.

En la lección anterior vimos cómo se estructura una etiqueta y algunas excepciones. En HTML, existen muchas etiquetas que se utilizan para crear y organizar el contenido dentro del cuerpo de la página. Estas etiquetas de maquetado incluyen encabezados (`<h1>` a

<h6>), párrafos (<p>), listas ordenadas () y listas no ordenadas (), enlaces (<a>), imágenes (), tablas (<table>), formularios (<form>) y muchos más.

El uso adecuado de las etiquetas de encabezado y cuerpo es fundamental para crear páginas web bien estructuradas y semánticamente correctas. Además, el uso adecuado de las etiquetas de maquetado ayuda a mejorar la accesibilidad y la usabilidad de la página web para los usuarios, así como a mejorar el rendimiento y la optimización del motor de búsqueda.

En esta lección hablaremos sobre las etiquetas de encabezado (<head>) y cuerpo (<body>), además de algunas etiquetas de maquetado.

Etiqueta <cabeza>

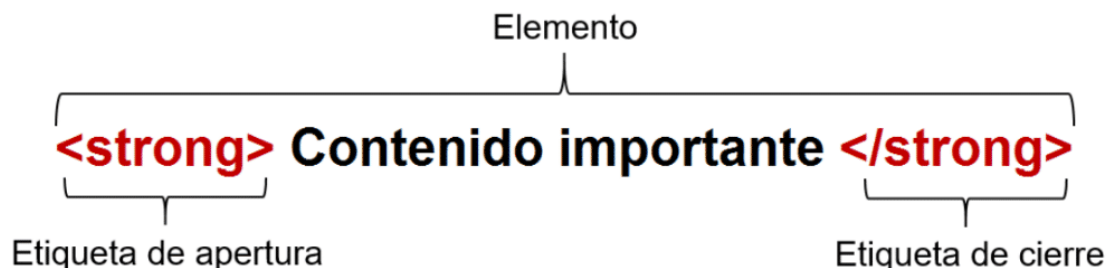
El encabezado (head) de un documento de HTML contiene información que no tiene una representación visual dentro del contenido de la página web, excepto por el título. Esto quiere decir que todos los elementos y etiquetas que ingresemos dentro de la etiqueta <head>, no se verán en nuestra página, excepto el título.

Metainformación

El encabezado puede contener información adicional sobre el documento (lo que llamamos metainformación), que puede declararse haciendo uso de la etiqueta <meta>. Como esta etiqueta no renderiza contenido en el navegador, no lleva etiqueta de cierre.

La información que tenga esta etiqueta puede ser usada por navegadores para renderizar mejor el contenido o para los robots de los buscadores, lo cual mejora el posicionamiento en las búsquedas de los navegadores.

Se puede definir diferente información en la etiqueta <meta>, por ejemplo:



• Vista

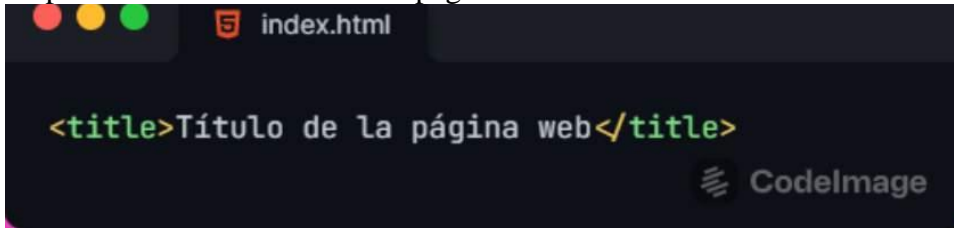
Esto es importante para el diseño web, ya que gracias a esta etiqueta, el diseño web responsive puede funcionar (esto lo veremos dentro del bootcamp). A esta información debemos acompañarla de dos más:

width=device-width: establecemos que el ancho de la página sea igual al dispositivo.

initial-scale=1.0: establecemos que la página tendrá un tamaño inicial de 1.

- **Título**

Es posible definir el título de la página web:

A screenshot of a code editor window. The title bar at the top shows a red, yellow, and green window control button on the left, followed by a file icon and the text "index.html". The editor area has a dark background and shows the HTML code `<title>Título de la página web</title>` in a light green monospace font. In the bottom right corner of the editor, there is a small icon of three horizontal lines and the text "CodeImage".

```
<title>Título de la página web</title>
```

Etiqueta <cuerpo>

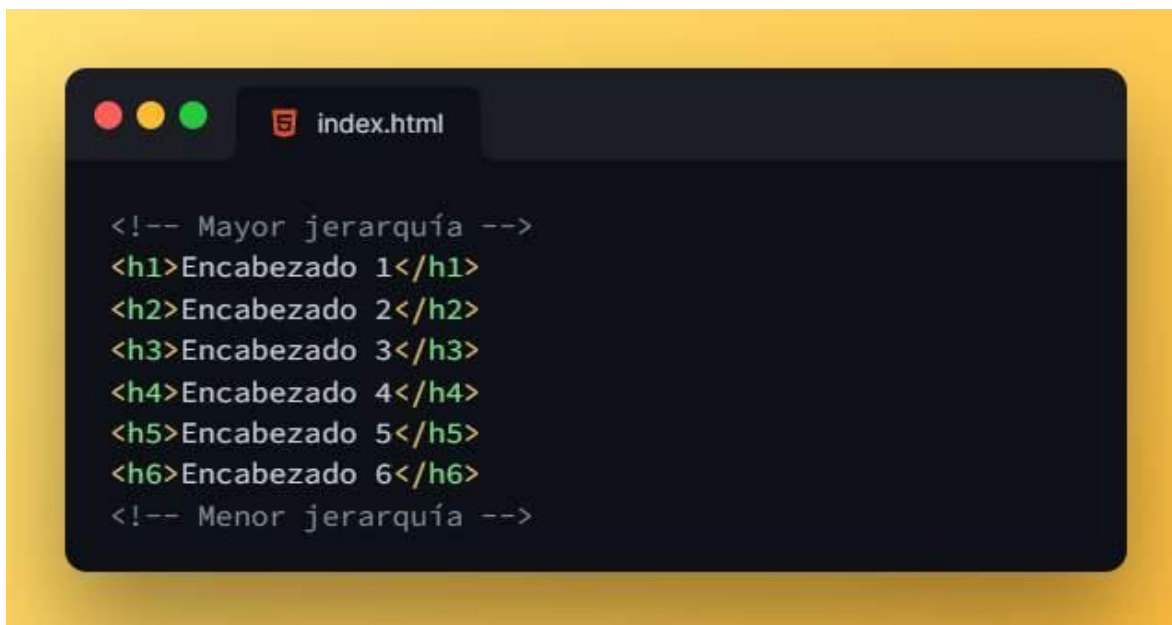
La etiqueta de contenido o cuerpo (body), permite incluir todo el contenido que se muestra en la página web. Todas las etiquetas que ingresemos dentro del cuerpo se verán reflejadas en el navegador. Veamos algunas:

Estos son utilizados para indexar la estructura y contenido de la página. Son muy importantes ya que los usuarios tienden a realizar una vista rápida de la página de acuerdo a los encabezados.

El encabezado `<h1>` lo usaremos para los de mayor importancia o para el principal. Luego continuamos con el `<h2>` hasta llegar al `<h6>`.

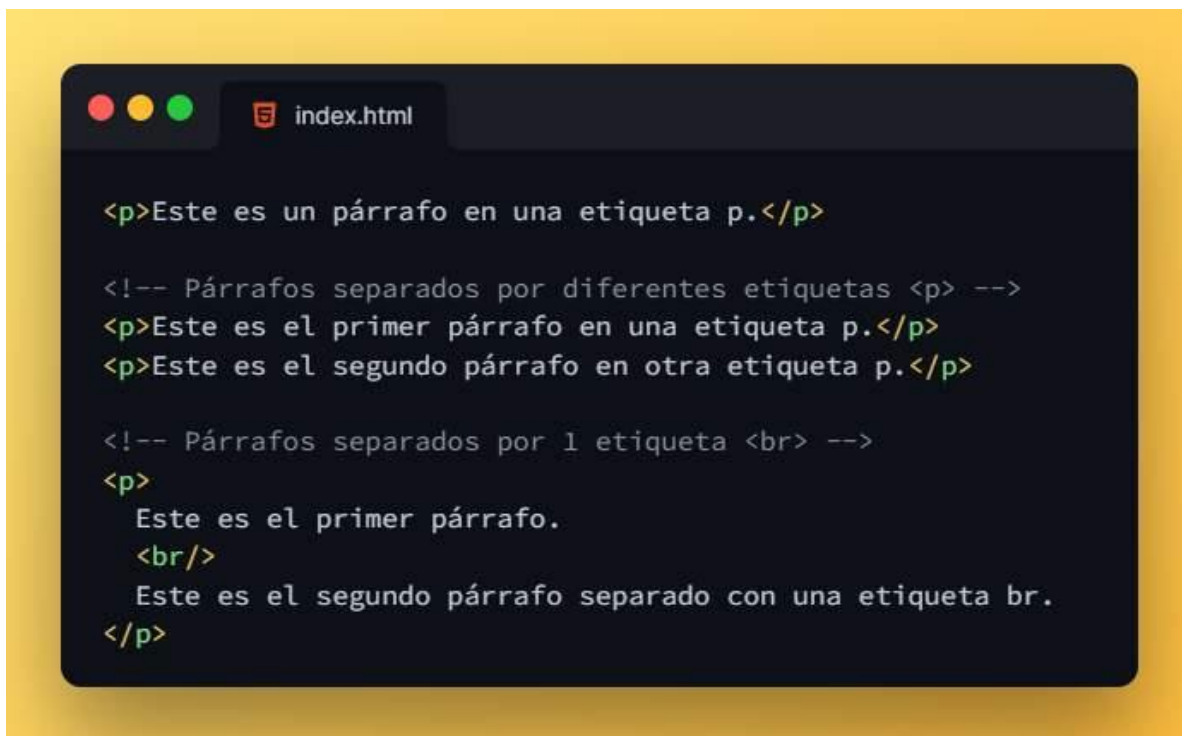
Estos encabezados son usados de acuerdo al orden y estructura de la página para que esta tenga una mejor ubicación. Por ejemplo, Google sólo permite un <h1> por página.

NOTA : Nunca deben usar para mostrar textos más grandes, para esto utilizamos CSS3 (lo cual abordaremos más adelante).

A screenshot of a code editor window titled 'index.html'. The editor has a dark background with syntax-highlighted HTML code. The code consists of a series of header tags from <h1> to <h6>, each followed by a closing tag. The tags are color-coded: <h1> is green, <h2> through <h6> are yellow. The code is enclosed in a comment block between <!-- and -->. The text 'Mayor jerarquía' is above the first tag and 'Menor jerarquía' is below the last tag.

```
<!-- Mayor jerarquía -->  
<h1>Encabezado 1</h1>  
<h2>Encabezado 2</h2>  
<h3>Encabezado 3</h3>  
<h4>Encabezado 4</h4>  
<h5>Encabezado 5</h5>  
<h6>Encabezado 6</h6>  
<!-- Menor jerarquía -->
```

Recomendamos utilizar la etiqueta <p> para escribir párrafos y no colocar texto plano. Además, para realizar saltos de línea podemos hacer uso de la etiqueta
.

A screenshot of a code editor window with a dark theme. The window has a title bar with three colored circles (red, yellow, green) and a tab labeled 'index.html'. The code is written in a light green font on a dark background. It shows two examples of paragraph creation in HTML. The first example uses a single paragraph tag. The second example uses two separate paragraph tags. The third example uses a paragraph tag with a line break tag to create two lines of text within a single paragraph.

```
<p>Este es un párrafo en una etiqueta p.</p>

<!-- Párrafos separados por diferentes etiquetas <p> -->
<p>Este es el primer párrafo en una etiqueta p.</p>
<p>Este es el segundo párrafo en otra etiqueta p.</p>

<!-- Párrafos separados por 1 etiqueta <br> -->
<p>
  Este es el primer párrafo.
  <br/>
  Este es el segundo párrafo separado con una etiqueta br.
</p>
```

En conclusión, la etiqueta `<head>` en HTML se utiliza para incluir información que no tiene representación visual en la página web, como el título y la metainformación. La etiqueta `<meta>` se utiliza para definir información adicional, como el juego de caracteres y la vista previa de la página en los motores de búsqueda. La etiqueta `<body>` se utiliza para incluir el contenido visible de la página web, como los encabezados y párrafos. Es importante utilizar correctamente estas etiquetas para mejorar la estructura y posicionamiento de la página en los motores de búsqueda y proporcionar una mejor experiencia de usuario.

En esta lección hemos aprendido sobre las principales etiquetas, su utilidad e implementación. Sumaremos estos conocimientos a la tarea anterior

En nuestro siguiente tem a conoceremos más etiquetas y sus usos

Etiqueta	Ejemplo	Uso
	Bold	No tiene importancia semántica, sólo le da estilo negrita al texto.
	Strong	Tiene importancia semántica ya que nos permite subrayar texto importante
<i>	<i>Italic</i>	No tiene importancia semántica, sólo permite cambiar el formato a Italic
	<i>Énfasis</i>	Es Italic con importancia semántica, permite hacer énfasis en una parte del texto
<small>	small	No tiene importancia semántica, permite agregar texto pequeño
<mark>	Marked	Sin importancia semántica, permite marcar o resaltar texto
	Deleted	Indicar que hemos borrado un texto
<ins>	Inserted	Indica que hemos insertado un texto
<sub>	Otro <u>textoSubscript</u>	Texto pequeño inferior
<sup>	 Otro Texto 1	Texto pequeño superior

< citar en Bloque >

Lorem ipsum dolor sit amet consectetur adipisicing elit. Quod ipsum, voluptatem accusantium ut ea nulla ullam vitae voluptatibus sapiente est officiis non ipsam itaque velit laborum tempora architecto quo pariatur.

<cite>

Lorem ipsum dolor sit amet consectetur adipisicing elit. Quod ipsum, voluptatem accusantium ut ea nulla ullam vitae voluptatibus sapiente est officiis non ipsam itaque velit laborum tempora architecto quo pariatur.

<hr>

Lorem ipsum dolor sit amet consectetur adipisicing elit. Quod ipsum, voluptatem accusantium ut ea nulla ullam vitae voluptatibus sapiente est officiis non ipsam itaque velit laborum tempora architecto quo pariatur.

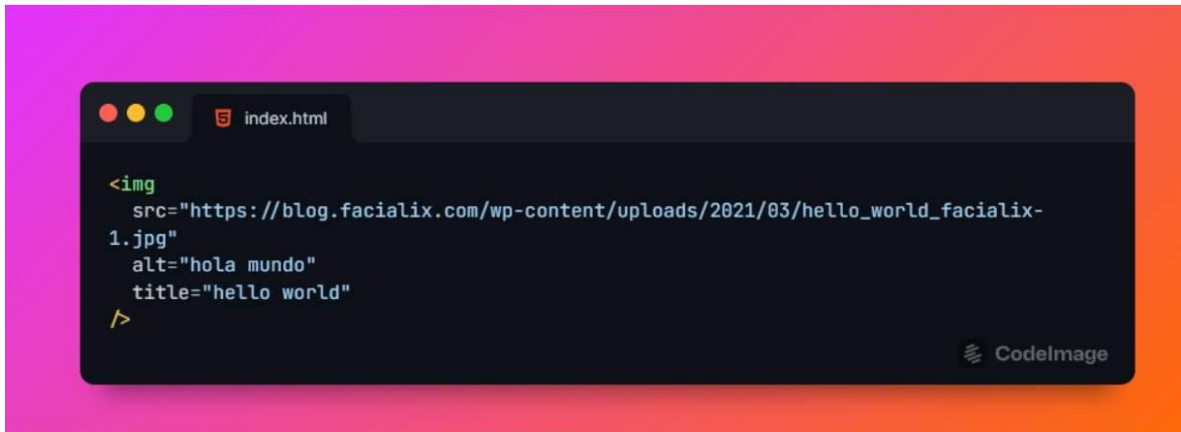
Etiqueta imagen

La etiqueta que utilizaremos para insertar imágenes en nuestro sitio web se llama . Es una etiqueta abierta, es decir que no necesita una etiqueta de cierre, y no permite renderizar imágenes de distintos formatos en nuestra página.

Los formatos admitidos por esta etiqueta son: jpeg, gif, png, apng, svg, bmp, bmp ico, png ico.

Recibe tres súper importantes atributos que debemos incluir siempre en nuestro código:

- Origen: Indica la ruta donde está ubicada la imagen que queremos mostrar.
- Alternativa Especifica un texto alternativo a la imagen, generalmente por accesibilidad y cuando no cargan las imágenes este es el texto que se muestra en su lugar.
- Título Permite definir un título para la imagen que se muestra cuando el cursor se ubique por encima de ella.



Una vez que aprendimos a utilizar la etiqueta correctamente, es muy importante que tengamos en cuenta que los recursos multimedia de los proyectos web son quienes ocupan más espacio en memoria y suelen hacer más lenta la carga de las mismas. A continuación profundizaremos en ello. Te brindaremos algunos tips para que optimices al máximo tu proyecto.

Las ventajas de formatear y comprimir tus imágenes

Hay numerosas ventajas de optimizar los archivos multimedia de tu proyecto para mejorar su rendimiento. Según HTTP Archive, las imágenes constituyen en promedio el 21% del peso total de la página web, y los videos hasta un 60%. Así que en este apartado, te daremos unos tips para mejorar el rendimiento de tu sitio web, en un principio teniendo en cuenta el tamaño y formato de las imágenes.

Debes tener en cuenta que, la optimización de la imagen es una de las cosas más fáciles de implementar, no es un proceso para nada complejo de realizar, sin embargo, muchos de los propietarios de sitios web pasan por alto este hecho, pagan una suma extra a su servicio de hosting para poder alojar su proyecto.

Te presentamos algunos de los principales beneficios de comprimir tus imágenes:

- bala

Se mejorará la velocidad de carga de tus páginas. Si tu página tarda demasiado en cargarse, los usuarios que ingresen podrían cansarse de esperar y pasar a otra página web.

- bala

Mejora tu SEO (posicionamiento en buscadores). Tu sitio clasifica mas alto en los resultados del motor de búsqueda. Los archivos grandes ralentizan el sitio y los motores de búsqueda lo dejarán en sus últimas páginas de resultados.

- bala

Crear copias de seguridad va a ser más rápido.

- bala

Los tamaños de los archivos de imagen más pequeños utilizan menos ancho de banda.

- bala

Requiere menos espacio de almacenamiento en el servidor.

¿Cómo optimizar mis imágenes?

El objetivo principal de formatear tus imágenes es encontrar el equilibrio entre el tamaño de archivo menor y una calidad aceptable.

Hay más de una manera de realizar estas optimizaciones. Una de las formas más populares es simplemente comprimirlas antes de cargarlas. Normalmente, esto se puede hacer mediante herramientas como Adobe Photoshop, Affinity Photo, entre otras.

Los dos aspectos principales a considerar son el formato de archivo y el tipo de compresión que utilizan. Elegir la combinación adecuada de formato de archivo y el tipo de compresión puede reducir el tamaño de la imagen. Tendrás que experimentar con cada imagen o formato de archivo para ver cuál funciona mejor, pero a continuación te dejaremos algunos aspectos a tener en cuenta.

Formato de archivo

Antes de empezar a modificar tus imágenes, debes asegurarte de que hayas elegido el mejor tipo de archivo. Existen varios tipos de archivos, los más populares y utilizados son: PNG, JPG y GIF.

Compresión con o sin pérdida

También es importante entender que hay dos tipos de compresión que se pueden utilizar, con pérdida y sin pérdida.

- Con pérdida: Es un filtro que elimina algunos de los datos. Esto degradará la imagen, así que deben tener cuidado al utilizarlo, ya que puede reducir demasiado la calidad de la imagen.
- Sin pérdida: Es un filtro que comprende los datos. Esto no reduce la calidad pero va a requerir que las imágenes sean descomprimidas antes de ser renderizadas.

Formato SVG

Otra recomendación que les podemos dar, es utilizar SVG junto con el resto de sus imágenes. SVG es una imagen vectorial en formato XML que funciona muy bien para los logotipos, iconos, texto e imágenes sencillas.

Una de las principales razones por la cual son tan recomendadas para la web, es que son automáticamente escalables tanto en navegadores como herramientas de edición de fotos, y nunca pierden calidad. Además, son tradicionalmente más pequeños que el tamaño del archivo PNG o JPG, por lo cual, puede resultar en tiempos de carga más rápidos.

Buenas Prácticas

- Usar JPG para fotos y capturas de pantalla generales.
- Usar PNG si necesita alto nivel de detalle y alta resolución o fondos transparentes.
- Usar GIFs sólo si necesitan utilizar animaciones (no olviden comprimirlas).

- Utilice imágenes vectoriales, siempre que sea posible, junto con los PNG y JPG.
- Usar compresión con pérdida cuando sea posible.
- Eliminar datos y metadatos de imagen necesarios.
- Guarde las imágenes como «optimizadas para web» en herramientas como Photoshop.
- Usar fuentes web en lugar de colocar el texto dentro de las imágenes, se ven mejor cuando se escalan y ocupan menos espacio.
- Recortar el espacio blanco y recrearlo usando CSS para proporcionar el relleno.
- Usar efectos CSS3 tanto como sea posible (ya lo tratarán más adelante en el módulo CSS).

Tabla HTML

Una tabla HTML es un conjunto de celdas organizadas dentro de las cuales podemos presentar distintos contenidos. Dicho lenguaje dispone de una gran variedad de etiquetas y atributos para crear tablas. Sirven principalmente para representar información tabulada, en filas y columnas.

En HTML4 las tablas se usan para maquetar. Cuando CSS creció y se hizo más fuerte, nacieron herramientas más potentes que las tablas. Pero para crear una grilla tabular, maquetar un calendario o utilizar simplemente filas y columnas, seguiremos haciendo uso de ellas. Además, aún se utiliza para el desarrollo de boletines.

Etiquetas básicas para tablas en HTML

Las tablas están definidas por las etiquetas `<table>` y `</table>`. Están compuestas por líneas que van de arriba a abajo que representan cada una de sus filas, y luego por columnas que van de izquierda a derecha conforme cada una de sus celdas.

Cada una de estas líneas o filas, está definida por la etiqueta: `<tr>` y `</tr>`, y dentro de cada línea, habrá diferentes celdas definidas por otra par de etiquetas: `<td>` y `</td>`. Dentro de estas etiquetas es donde colocaremos el contenido. Para definir los encabezados de columna, se utilizan las etiquetas `<th>` y `</th>`.



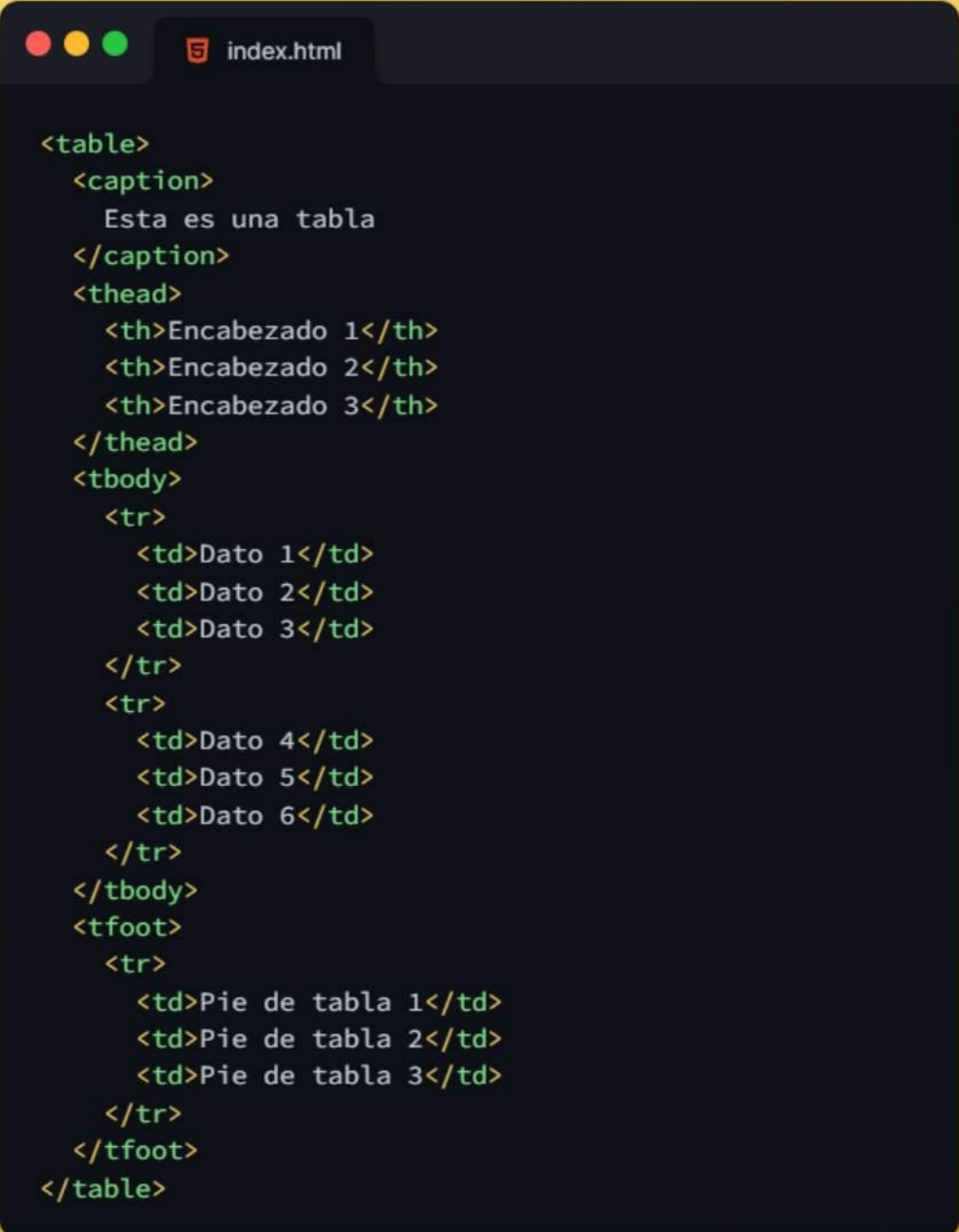
```

<!-- Definimos una tabla -->
<table>
  <!-- Definimos encabezados -->
  <th>Encabezado 1</th>
  ...
  <!-- Definimos las filas -->
  <tr>
    <!-- Y dentro de las filas, definimos las celdas de cada columna -->
    <td>Dato 1</td>
    ...
  </tr>
  ...
</table>

```

Además, podemos definir las diferentes secciones de una tabla, como el header o encabezado de tabla (<thead>), el body o cuerpo de la tabla (<tbody>) o el footer o pie de la tabla (<tfoot>).

Por último, también podemos utilizar la etiqueta <caption> que sirve para darle un título a la tabla creada. A continuación veremos un ejemplo del código de una tabla básica:



```
<table>
  <caption>
    Esta es una tabla
  </caption>
  <thead>
    <th>Encabezado 1</th>
    <th>Encabezado 2</th>
    <th>Encabezado 3</th>
  </thead>
  <tbody>
    <tr>
      <td>Dato 1</td>
      <td>Dato 2</td>
      <td>Dato 3</td>
    </tr>
    <tr>
      <td>Dato 4</td>
      <td>Dato 5</td>
      <td>Dato 6</td>
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <td>Pie de tabla 1</td>
      <td>Pie de tabla 2</td>
      <td>Pie de tabla 3</td>
    </tr>
  </tfoot>
</table>
```

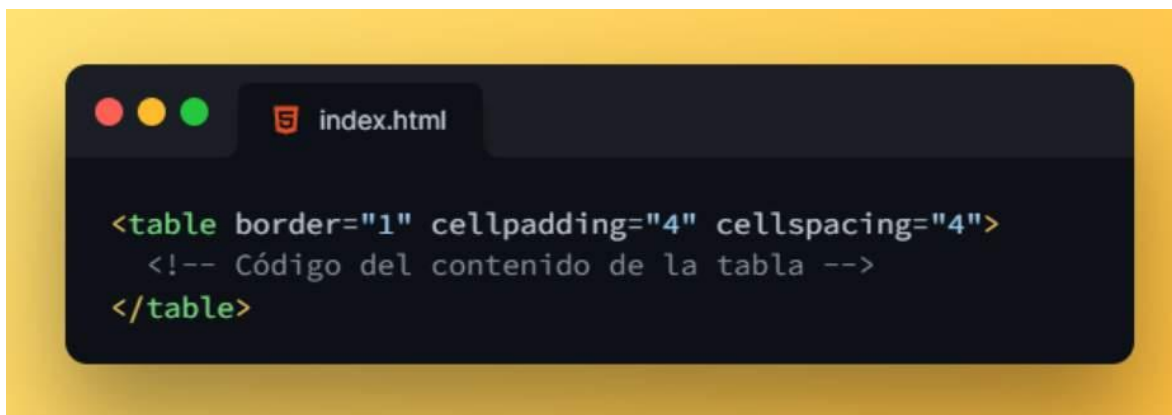
Si observamos este código en el navegador, veremos una tabla en la cual se nos dificulta entender cómo están divididos los datos en filas y columnas, por lo cual, les presentaremos algunas características que pueden utilizar hasta que aprendamos CSS.

Atributos de diseño

Como dijimos antes, hasta que aprendamos a dar estilos visuales a los elementos HTML, la etiqueta `<table>` acepta 3 atributos de diseño, estos son:

- Borde: Número que representa el tamaño del borde de todas las filas y columnas (por defecto es 0).
- Cellpadding: Número que representa el espacio interno en píxeles de una celda (por defecto es 3).
- Cellspacing: Número que representa el espacio hacia afuera entre celda y celda (por defecto es 3).

La forma de utilizarlos seria la siguiente:



El valor indicado en cada caso se aplica a los 4 lados por igual. Si quieres algo más específico tendrás que utilizar CSS.

Haciendo uso del código de ejemplo para crear una tabla que mostramos más arriba, e implementando los atributos de diseño que nombramos, deberíamos obtener como resultado una tabla como esta:

Esta es una tabla

Encabezado 1	Encabezado 2	Encabezado 3
Dato 1	Dato 2	Dato 3
Dato 4	Dato 5	Dato 6
Pie de tabla 1	Pie de tabla 2	Pie de tabla 3

Listas

HTML no 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, y aunque cada palabra por separado tiene sentido, de forma conjunta constituye el menú de navegación de la página, por lo que su significado es mayor en grupo que por separado. Esto se denomina como “listas”.

Les dejamos algunos casos de uso para saber en qué situaciones utilizar dichas listas:

Toda la enumeración de pasos o secuencias a seguir en orden.

Cualquier enumeración de propiedades o características técnicas de una persona o producto.

Las galerías de imágenes (al tener todas las fotos la misma información –supongamos título, imagen y descripción– y por formar todas parte del mismo álbum).

Las botoneras de todo sitio web (ya que cada botón forma parte del listado de opciones que tenemos para navegar nuestra página).

Tipos de listas

Cuando queramos hacer uso de las listas, debemos tener en cuenta que no todas tienen que verso de la misma forma, existen diferentes tipos de listas que podemos utilizar, y cada una tiene su particularidad. Las más conocidas son:

listas ordenadas

Listas no ordenadas

Listas de definición

¿Cómo creamos una lista?

En primer lugar, debemos saber que son elementos compuestos. Dado que por un lado tenemos la inserción de la lista, y por otro lado los elementos (ítems) del listado en cuestión.

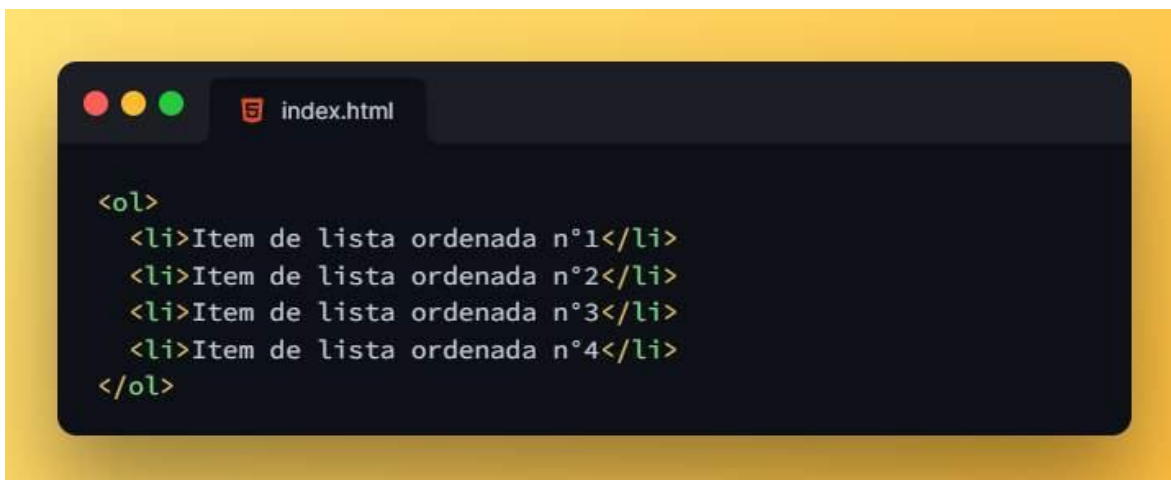
Entonces, para insertar una lista deberíamos usar las etiquetas:

``: Definir una lista ordenada.

``: Definir una lista no ordenada.

``: Define un nuevo ítem de una lista, por lo cual, cada uno de los ítems que forman parte de cualquiera de ambas listas se debe insertar mediante la etiqueta `` (list-item).

Profundizando en el tema, las listas numéricas u ordenadas fundamentando un orden en la lectura de sus ítems (indicando qué elemento va antes del próximo). También define una jerarquía entre los elementos, cuál de los ítems es más importante que el siguiente. Utilice símbolos numéricos para enumerar cada uno de sus artículos.

A screenshot of a code editor window titled 'index.html'. The editor has a dark background and shows the following HTML code for an ordered list:

```
<ol>
  <li>Item de lista ordenada nº1</li>
  <li>Item de lista ordenada nº2</li>
  <li>Item de lista ordenada nº3</li>
  <li>Item de lista ordenada nº4</li>
</ol>
```

Las listas de viñetas o no ordenadas no representan ningún orden o importancia entre sus artículos, se asumen todos igual de importantes. Es indistinto en qué orden se leen, el conjunto de los elementos representa un todo. Utilice viñetas para enumerar cada uno de sus artículos.

Anidar listas

Es probable que nos veamos en la necesidad de crear una estructura de sub-listas en algún momento, es decir, introducir una lista dentro de otra. Por ejemplo, cuando en el índice de un libro podemos tener no solo los títulos de los capítulos, sino también los subtítulos de los temas que va tratando. Para esto se puede recurrir a anidar listas.

Para poder hacerlo, basta con definir una nueva lista dentro de un `` de la lista en cuestión. Es importante saber que dentro de una lista sólo se aceptan list-items, y dentro de un list-item se acepta cualquier elemento (incluso otras listas). Te mostramos un ejemplo:

A screenshot of a code editor window titled 'index.html'. The editor has a dark background and shows the following HTML code for a nested unordered list:

```
<ul>
  <li>Item de lista no ordenada nº1</li>
  <li>Item de lista no ordenada nº2</li>
  <li>
    Item de lista no ordenada nº3
    <ul>
      <li>Sub-Item de lista no ordenada nº1</li>
      <li>Sub-Item de lista no ordenada nº2</li>
      <li>Sub-Item de lista no ordenada nº3</li>
    </ul>
  </li>
  <li>Item de lista no ordenada nº4</li>
</ul>
```


- Item de lista no ordenada n°1
- Item de lista no ordenada n°2
- Item de lista no ordenada n°3
 - Sub-Item de lista no ordenada n°1
 - Sub-Item de lista no ordenada n°2
 - Sub-Item de lista no ordenada n°3
- Item de lista no ordenada n°4

Listas de definición

Las listas de definición no forman parte del conjunto anterior, dado que tienen un objetivo diferente que las listas ordenadas o desordenadas y sus elementos son diferentes también.

En primera instancia, una lista de definición no está formada por elementos, sino por significados. Y para lograrlo, se requieren dos elementos distintos:

- El término para explicar,
- El significado que corresponde a dicho término.

Para insertar, entonces, una lista de definición deberemos trabajar con tres elementos distintos:

- `<dl>`: Define una lista de definición, es decir, una lista de términos y sus definiciones asociadas.
- `<dt>`: Representa un término a definir.
- `<dd>`: Representa la definición o significado de los términos listados antes que él.

Les dejamos un ejemplo con la sintaxis de dicho tipo de lista:

A screenshot of a code editor window titled 'index.html'. The editor has a dark background with light-colored text. It displays the HTML code for a definition list. The code is as follows:

```
<dl>
  <dt>Término 1</dt>
  <dd>Significado o definición del término 1</dd>
  <dt>Término 2</dt>
  <dd>Significado o definición del término 2</dd>
  <dt>Término 3</dt>
  <dd>Significado o definición del término 3</dd>
</dl>
```

Término 1

Significado o definición del término 1

Término 2

Significado o definición del término 2

Término 3

Significado o definición del término 3

¿Qué hemos aprendido?

En esta lección hemos conocido las **listas en HTML**, elementos muy útiles para organizar información de manera jerárquica y estructurada en nuestros documentos web. Conociendo las etiquetas ``, ``, y ``, podemos crear diferentes tipos de listas y luego podremos personalizar su apariencia según nuestras necesidades.

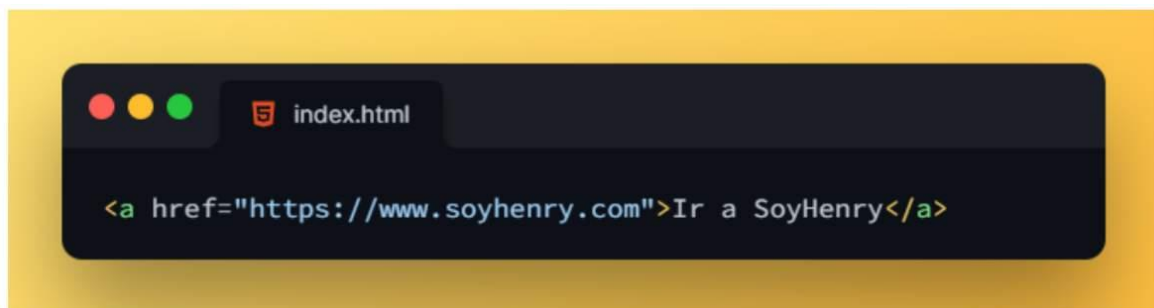
En la siguiente lección veremos los enlaces en HTML que nos permitirán conectar diferentes páginas web y recursos multimedia entre sí, brindando a los usuarios una experiencia de navegación más completa y enriquecedora.

Enlaces

Como dijimos anteriormente, los enlaces (también conocidos como links o anchors) se utilizan para relacionar partes del documento con otros documentos o con partes del mismo documento. Por defecto, los enlaces se visualizan azules y subrayados, aspecto que luego podremos cambiar con CSS.

Para crear un enlace es necesario utilizar la etiqueta de ancla `<a>` con el atributo `"href"`. En este atributo es donde indicamos la URL a la que apuntaremos con el enlace. O sea, la página o la sección de la página a dónde irá el navegador al hacer clic en el enlace.

Esta es la sintaxis de un enlace en HTML:



El texto "Ir a SoyHenry", que figura entre las etiquetas <a>, es el que se visualizará en el navegador y que servirá como enlace a la url especificada en el atributo "href".

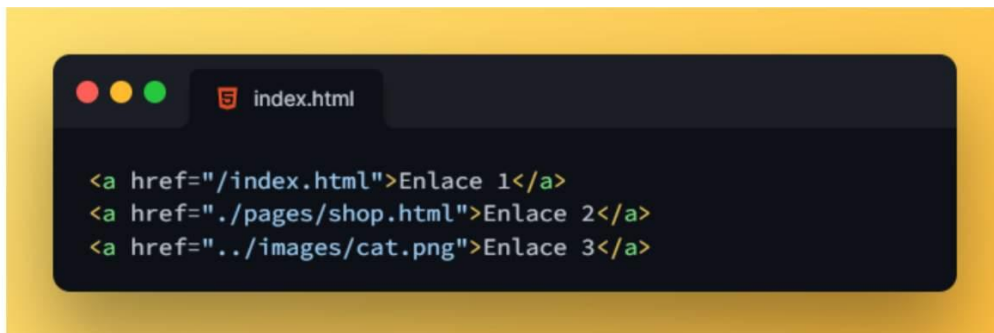
Tipos de enlaces

Existen 3 tipos de enlaces que nosotros podemos utilizar dentro de nuestras webs, y nos van a servir para realizar diferentes acciones, ellos son:

- Enlaces relativos
- Enlaces absolutos
- Enlaces internos

Enlaces relativos

Los enlaces relativos son aquellos que apuntan a páginas ubicadas dentro del mismo proyecto. Los tres casos que mostraremos a continuación, son ejemplos de este tipo de rutas, rutas relativas a la página en la que estoy ahora mismo.

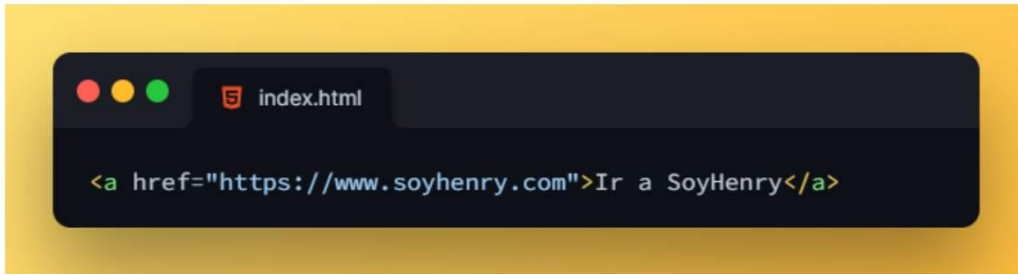


- En el primer caso, la dirección empieza por un slash "/". Esto significa que empezará a buscar la ruta a partir de la raíz del dominio.
- En el segundo caso empieza por "./". Esto significa que empezará a buscar la ruta a partir del nivel de la carpeta actual.
- Y en el tercero, empieza por "../". Esto significa que empezará a buscar la ruta a partir del nivel inmediatamente superior al de la carpeta actual. Podrías anidar varios "../" para subir varios niveles.

Este tipo de rutas tienen la ventaja de que son independientes del dominio. Si el día de mañana le cambias el dominio a tu página web, seguirán funcionando igual de bien.

Enlaces absolutos

Los enlaces absolutos son aquellos cuyo destino apunta a un documento por fuera del sitio y debe ser especificado utilizando la URL completa.



Enlaces internos

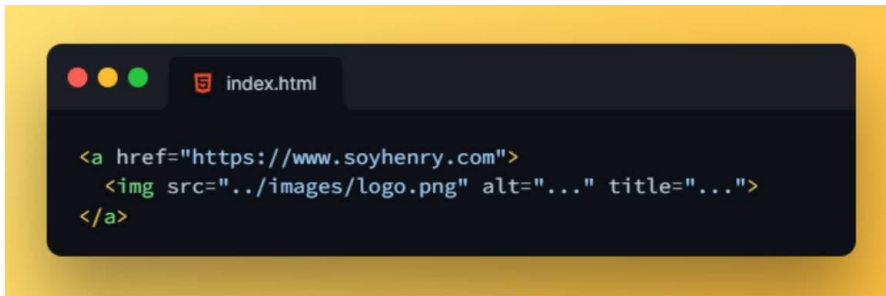
Los enlaces internos permiten referenciar secciones de nuestra página, para hacer esto se utiliza el atributo `id` del elemento a referenciar. Este tipo de enlace nos permite generar un efecto de scroll automático hacia el elemento especificado. Por ejemplo, es un recurso muy utilizado en las tablas de contenido de los artículos de Wikipedia.



También podemos usar como destino una sección específica una página distinta, indicando la página a visitar y la respectiva sección a buscar.



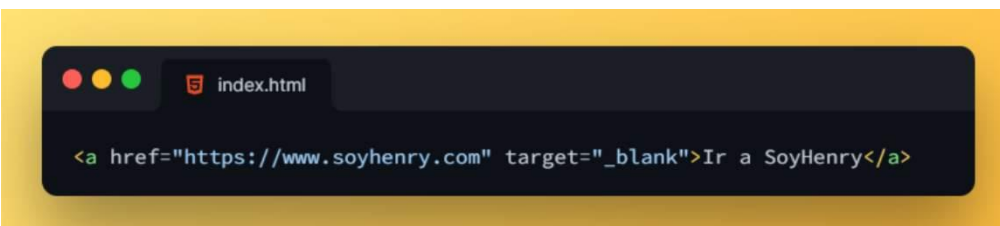
No sólo podemos agregar enlaces a texto, también podemos hacerlo con otros elementos. Por lo general se usan texto o imágenes. Ejemplo de enlace con una imagen:



En este caso la imagen será clickeable y enlazará al destino especificado en el atributo “href”.

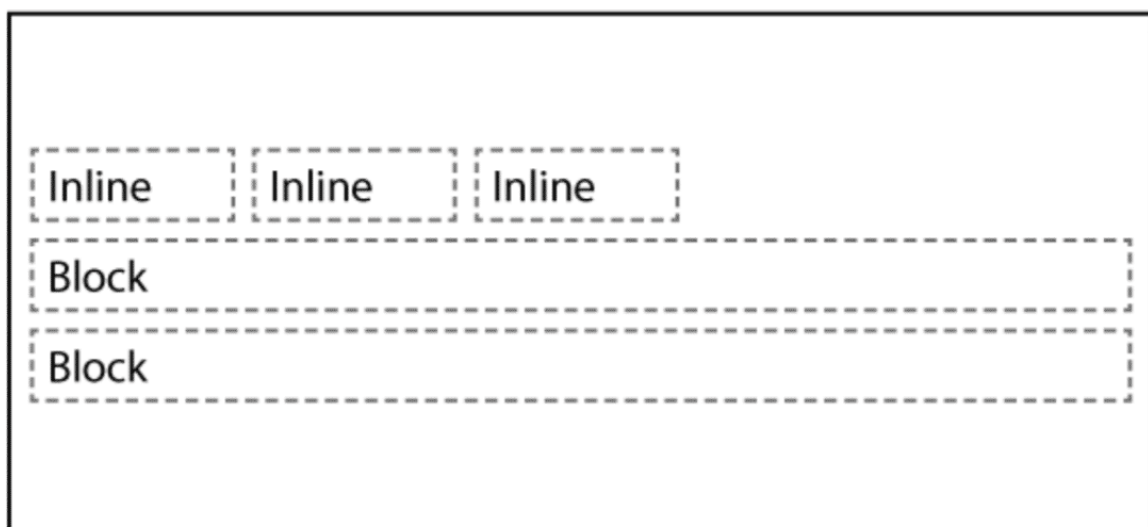
¿Y en dónde se abre el enlace?

El atributo target se utiliza para especificar dónde vamos a abrir la URL enlazada. Parte del ámbito en el que estamos (una ventana) y, en función de este, se designa dónde abrimos el recurso clickeado. Su valor más utilizado es “_blank”, y sirve para abrir el recurso en una pestaña nueva. Te dejamos un ejemplo de uso:



Elementos en bloque y en línea

Cuando hablamos de la forma que tienen los elementos HTML, debemos tener en cuenta que todos son “cajas”. Estas cajas, rectangulares por defecto, ocupan diferentes tamaños según el tipo de elemento. El estándar HTML clasifica a todos sus elementos en dos grandes grupos: elementos en línea (inline) y en bloque (block).



Debemos entender que los bloques jamás van a permitir que haya nada a su lado aunque haya espacio. Nunca podremos colocar nada junto a un elemento en bloque. Por ello, en una web sin formato todos los elementos suelen ordenarse de forma vertical. Los elementos en bloque crean saltos de línea y esto es debido a que siempre ocuparán el 100% del ancho, normalmente son utilizados como contenedores. Si le dices a un bloque que ocupe un ancho determinado en píxeles (width: 200px; por ejemplo), dará lo mismo, creará un margen a su lado que impedirá que cualquier elemento se ponga junto a él.

Por otro lado, tenemos los elementos en línea. Si los elementos en bloque eran el contenedor, debemos tratar a los elementos en línea como el contenido. Cuando colocamos un elemento en línea junto a otro se colocará a su lado siempre que el ancho de pantalla lo permita, es decir, siempre que haya espacio a su lado. Podemos tratar a los elementos en línea como si fuera texto. No están pensados para ser contenedores, sino que son contenidos por ellos. Entonces, a diferencia de los elementos en bloque, un elemento en línea sólo ocupa lo estrictamente necesario, no el 100% del ancho disponible.

Debemos entender las diferencias entre ambos elementos para poder crear nuestras estructuras luego con CSS. No podemos tratar por igual a un elemento en bloque y en línea, es uno de los errores más comunes cuando se comienza a maquetar. Y muchas veces, esta es la parte más difícil, diferenciarlos.

Etiquetas en línea tenemos muchas. Y etiquetas en bloque, aún más. Te dejamos una lista de las etiquetas HTML más utilizadas y su clasificación para ayudarte un poco con tu trabajo a la hora de maquetar en HTML:

Elementos en Línea	Elementos en Bloque
a, abbr, acronym, b, bdo, big, br, button, cite, code, dfn, em, i, img, input, kbd, label, map, object, q, samp, script, select, small, span, strong, sub, sup, textarea, time, tt, var	address, article, aside, audio, blockquote, canvas, dd, div, dl, fieldset, figcaption, figure, footer, form, h1, h2, h3, h4, h5, h6, header, hgroup, hr, li, main, nav, noscript, ol, output, p, pre, section, table, tfoot, ul, video

Este concepto de que “todo es una caja”, da lugar a algo llamado en la web como “box model” o “modelo de caja”. Sin importar si son de línea o de bloque (pero tienen su incidencia en lo que sean), todas las etiquetas tienen ciertas propiedades en común que luego trataremos en el módulo de CSS.

Formularios

Como dijimos anteriormente, si se necesita recopilar información ingresada por un usuario, HTML ofrece componentes de formulario, que son básicamente etiquetas donde el usuario ingresará o seleccionará valores que serán enviados a un archivo encargado de procesar la información.

Los formularios se usan bastante seguidos en la web, los pueden encontrar cuando:

1. Ingresan a un sistema de usuario/clave.
2. Publican un nuevo contenido en redes sociales.
3. Envían un mensaje desde un sitio web.

4. Confirman una compra o entrada al cine.

Para crear un formulario, debemos hacer uso de un conjunto específico de etiquetas y atributos que te enseñaremos a continuación.

Etiqueta

Para insertar un formulario se usa la etiqueta `<form>` que dentro lleva todos los componentes que van al mismo destino.

Un formulario requiere tres atributos para funcionar:

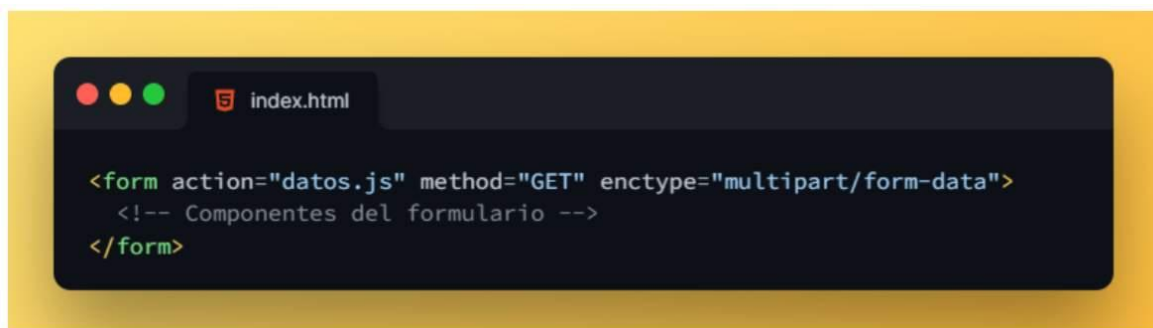
- **Acción:** Representa el enlace al documento que se encarga de recibir los datos y procesarlos. Debe estar escrito en un lenguaje de los llamados “del lado del servidor” (PHP / ASP / JS). Si no se indica un valor, por defecto la Acción es el mismo archivo donde está el formulario.
- **Método:** Representa la forma en que será enviada la información. Existen principalmente dos métodos de envío:
- **GET**, donde la información viajará por la barra de direcciones a continuación del nombre del archivo (URL), y
- **POST**, donde la información viajará junto a los encabezados del HTML (quedará oculta, será “invisible”).

Por defecto, si no se indica el Método, es GET.

Enctype: Indica cómo se codificarán los contenidos a enviar. Sus posibles valores son:

- **application/x-www-form-urlencoded:** Será el valor por defecto si el atributo no está especificado.
- **multipart/form-data:** Debe usar este valor si se está usando el elemento `input` con el atributo `type` ajustado a "file", es decir, si se necesita enviar archivos en el formulario.
- **text/plain:** Es un formato ambiguo, envía contenido legible por humanos que no puede interpretarse de manera confiable por una computadora.

Teramos como utilizar esta etiqueta con sus atributos:

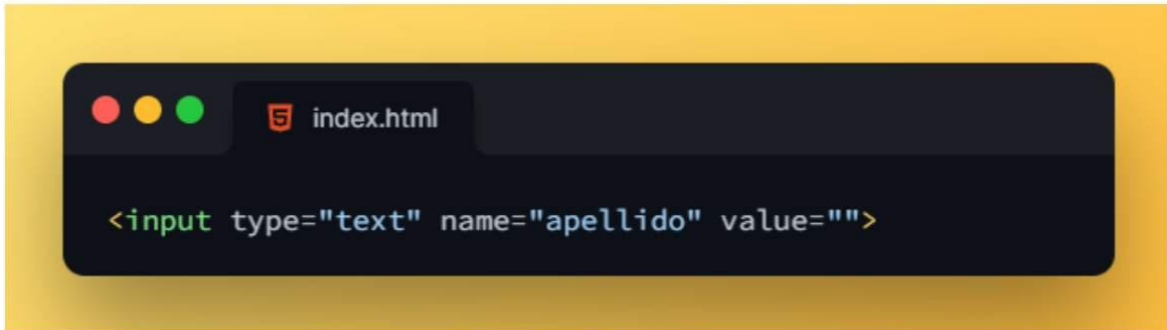


Etiqueta `<entrada>`

Este elemento tiene múltiples propósitos según el valor que tome el atributo “tipo”. Principalmente se dedica a capturar texto escrito por el usuario, y es una etiqueta abierta, por lo que no requiere etiqueta de cierre. A continuación te enseñamos la sintaxis básica de una entrada:

Etiqueta <entrada>

Este elemento tiene múltiples propósitos según el valor que tome el atributo “tipo”. Principalmente se dedica a capturar texto escrito por el usuario, y es una etiqueta abierta, por lo que no requiere etiqueta de cierre. A continuación te enseñamos la sintaxis básica de una entrada:



Atributos comunes entre componentes de formularios

Todos los insumos que utilizaremos en los formularios permiten la utilización de diferentes atributos. Algunos de ellos son propios del tipo de entrada específico, pero también tenemos algunos atributos que son comunes a todos los insumos que insertamos en nuestros formularios, los más utilizados son:

- tipo
- nombre
- valor

Nombre del atributo

Todos los componentes de un formulario deben tener un nombre, que se indica con el atributo “name”. Con ese nombre después se accede a su valor en el archivo indicado en la acción. Un input sin nombre no se envía.

Valor de atributo

En atributo value se comporta de forma diferente dependiendo del tipo de entrada.

En el caso de los input que permiten la entrada de texto, el valor del atributo sirve para dar un valor predeterminado al input, puede ser intervenido por el usuario e incluso enviado al servidor en caso de que no se modifique.

Por otro lado, en el caso de los inputs de selección de valores, en especial en los menús desplegables, botones de radio y checkbox, el atributo value es el valor que se envió al enviarse el formulario, que puede ser independiente al valor que es mostrando por pantalla. El navegador nunca muestra el valor al usuario.

Y por último, en el caso de los input que se muestran como botones, el valor del atributo representa la etiqueta del botón, es decir, el texto que será visible dentro del botón en los navegadores.

Tipo de atributo

Este elemento tiene múltiples propósitos según el valor que tome el atributo "tipo". Principalmente se dedica a capturar texto escrito por el usuario. Los posibles valores del atributo type son:

Operaciones aritméticas	
Types	Descripción / Uso
text	Campo de texto de línea simple. Los saltos de línea son eliminados automáticamente del valor introducido.
password	Campo de texto de línea simple cuyo valor permanece oculto, mostrando un símbolo por cada carácter escrito.
radio	Botón radio (circular) de selección única. Se debe usar el atributo value para definir el valor que se enviará por este elemento. Se usa el atributo checked para indicar si el elemento está seleccionado de forma predeterminada.
checkbox	Casilla de selección múltiple o verificación. Se debe usar el atributo value para definir el valor que se enviará por este elemento. Se usa el atributo checked para indicar si el elemento está seleccionado.
color	Input para especificar un color. Despliega una interfaz de selección de color, no requiere más funcionalidad que la de aceptar colores simples como texto.
email	Campo de texto para introducir una dirección de correo electrónico. El valor introducido se valida para que contenga una cadena vacía o una dirección de correo válida antes de enviarse.
file	Botón que permite al usuario seleccionar un archivo. Se puede usar el atributo accept para definir los tipos de archivo que el input podrá seleccionar.

Operaciones aritméticas	
hidden	Input que no es mostrado en pantalla, pero cuyo valor es enviado al servidor.
image	Botón de envío de formulario con gráfico/imagen. Su funcionamiento es muy similar al input de tipo file.
number	Input para introducir un número.
range	Input para introducir un número cuyo valor exacto no es importante. Representa un rango numérico.
search	Campo de texto de línea simple para introducir textos de búsqueda. Los saltos de línea son eliminados automáticamente del valor introducido.
tel	Input para introducir un número telefónico. Los saltos de línea son eliminados automáticamente del valor introducido, pero no hay otra sintaxis forzada.
time	Input para introducir un valor de tiempo sin zona horaria específica.
date	Input para introducir una fecha (año, mes y día, sin hora).

datetime	Input para introducir una fecha y hora (horas, minutos, segundos y fracción de segundo), basado en la zona horaria UTC. Hoy en día, se encuentra obsoleto.
datetime-local	Input para introducir fecha y hora, sin zona horaria específica.
week	Input para introducir una fecha que consiste en número de semana del año sin zona horaria específica.
month	Input para introducir un mes y año, sin zona horaria específica.
url	Input para escribir una URL. El valor introducido se valida para que contenga una cadena vacía o una ruta URL absoluta antes de enviarse. Los saltos de línea y espacios en blanco al principio o al final del valor son eliminados automáticamente.
submit	Input de tipo botón que permite el envío de datos al archivo que los procesará, declarado en el atributo action.
reset	Input de tipo botón que restaura los contenidos de un formulario a sus valores predeterminados.
button	Input de tipo botón sin un comportamiento específico, está pensado para usarse luego con scripts de lenguajes como Javascript.

Etiqueta <área de texto>

Es un campo que nos permite insertar múltiples líneas de texto, y acepta los siguientes atributos:

- cols: Indica la cantidad de caracteres que podrán tener el campo a lo ancho.
- filas: Indica la cantidad de líneas de texto que aceptará.

Entre sus etiquetas de apertura y cierre, se puede dejar un texto predeterminado que podrá ser modificado por el usuario.



```

<textarea name="descripcion" rows="5" cols="33">
  Texto modificable...
</textarea>

```

Menú desplegable de selección: <seleccionar>

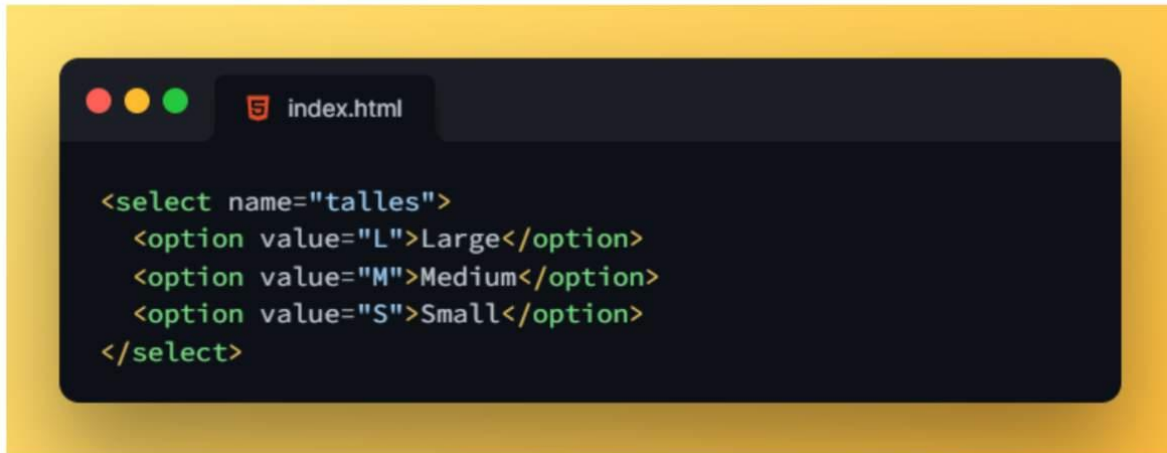
Es el llamado combo-box, selector o menú. Se le proporciona al usuario una lista de valores, y de toda la lista podrá elegir una opción (aunque tiene un atributo que permite cambiarlo).

Se necesitan dos elementos distintos para crearlo:

- Para insertar el menú se usa la etiqueta <select>, que abre y cierra la lista.
- Dentro de <select> va una etiqueta <option> por cada opción a mostrar. Dentro de ellas va el texto a mostrar.

Algunas cuestiones a tener en cuenta para su uso:

- El `<select>` es quien lleva el atributo “name”.
- Los `<option>` son quienes llevan el atributo “value”.
- Si no hay “valor”, se usa el texto interior

A screenshot of a code editor window titled 'index.html' with a dark background. The code is written in a light green font and defines a select dropdown menu with three options: Large, Medium, and Small.

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

Etiqueta `<etiqueta>`

La etiqueta `<label>` define formalmente cada elemento de un formulario, esta etiqueta es de mucha ayuda para generar un formulario accesible. El principal atributo de la etiqueta `label` es “for”, quien va a referenciar al “label” con su elemento del formulario. El valor del atributo debe ser igual al valor del atributo `id` o `name` del elemento, dependiendo de la utilidad que se le desee dar.

A screenshot of a code editor window titled 'index.html' with a dark background. The code is written in a light green font and defines a label for a text input field.

```
<label for="apellido">Apellido:</label>
<input type="text" name="apellido" value="" />
```

Definición de grupos de radio o checkbox

Un grupo de opción de radio o checkbox se define dando a cada uno de los botones de opción del grupo el mismo valor al atributo `name`. Una vez que se establece un grupo de radio, al seleccionar cualquier botón de radio en ese grupo automáticamente se deselecta cualquier botón de radio seleccionado actualmente en el mismo grupo. En el caso de los grupos de checkbox no sucede lo mismo, ya que este tipo de entrada nos permite la selección de múltiples valores.

Además, cada uno de los elementos de radio o checkbox, debe tener un valor único que lo identifique de manera única dentro del grupo, utilizando el atributo `id`. Esta identificación única, será utilizada luego por el atributo del elemento `<label>` para asociar las etiquetas con los botones de opción y generar un formulario accesible.

Te dejamos un ejemplo de grupo de radio:

A screenshot of a code editor window titled 'index.html' with a dark theme. The code defines a group of radio buttons. It starts with a <div> tag, followed by a <label for="grupo">Grupo de opciones:</label>. Then, two <input type="radio" name="grupo" id="opcion1" value="op1" /> and <input type="radio" name="grupo" id="opcion2" value="op2" /> are defined, each followed by its corresponding <label for="opcion1">Opción 1</label> and <label for="opcion2">Opción 2</label>. The code ends with </div>.

```
<div>

  <label for="grupo">Grupo de opciones:</label>

  <input type="radio" name="grupo" id="opcion1" value="op1" />
  <label for="opcion1">Opción 1</label>

  <input type="radio" name="grupo" id="opcion2" value="op2" />
  <label for="opcion2">Opción 2</label>

</div>
```

Y un ejemplo de grupo de checkbox:

A screenshot of a code editor window titled 'index.html' with a dark theme. The code defines a group of checkboxes. It starts with a <div> tag, followed by a <label for="grupo">Grupo de opciones:</label>. Then, two <input type="checkbox" name="grupo" id="opcion1" value="op1" /> and <input type="checkbox" name="grupo" id="opcion2" value="op2" /> are defined, each followed by its corresponding <label for="opcion1">Opción 1</label> and <label for="opcion2">Opción 2</label>. The code ends with </div>.

```
<div>

  <label for="grupo">Grupo de opciones:</label>

  <input type="checkbox" name="grupo" id="opcion1" value="op1" />
  <label for="opcion1">Opción 1</label>

  <input type="checkbox" name="grupo" id="opcion2" value="op2" />
  <label for="opcion2">Opción 2</label>

</div>
```

Un último paso...

Te dejamos un último ejemplo, en donde puedes observar el código de un formulario de contacto, y como se ve en el navegador:

Nombre Completo:

Correo Electrónico:

Género: ☐ Femenino ☐ Masculino

Talle:

Consulta:

Enviar

Borrar

CODIGO HTML5

```
<form action="datos.js" method="GET" enctype="multipart/form-data">

  <label for="nombre">Nombre Completo:</label>
  <input type="text" name="nombre" value="" />

  <label for="email">Correo Electrónico:</label>
  <input type="email" name="email" value="" />

  <label for="genero">Género:</label>
  <input type="radio" name="genero" id="femenino" value="f" />
  <label for="femenino">Femenino</label>
  <input type="radio" name="genero" id="masculino" value="m" />
  <label for="masculino">Masculino</label>

  <label for="talles">Talle:</label>
  <select name="talles">
    <option value="L">Large</option>
    <option value="M">Medium</option>
    <option value="S">Small</option>
  </select>

  <label for="consulta">Consulta:</label>
  <textarea name="consulta" rows="5" cols="33"></textarea>

  <input type="submit" value="Enviar" />
  <input type="reset" value="Borrar" />

</form>
```