



Módulo 2: Fundamentos de Desarrollo Front-End

**Desarrollador de aplicaciones
Full Stack**

Python Trainee



MÓDULO 2 – FUNDAMENTOS DE DESARROLLO FRONT-END

2.2.- El Lenguaje HTML

2.2.1.- El Lenguaje HTML

2.2.1.1.- Características de HTML5

Es un lenguaje de etiquetas usado para presentar y estructurar el contenido en un sitio web [1]. Es la quinta revisión del estándar HTML, el cual fue creado en 1990. La W3C recomendó su utilización para transformarse en el estándar a ser usado en proyectos venideros. Está relacionado también con la entrada en decadencia del anterior estándar HTML 4, que se combinaba con otros lenguajes para producir los sitios que se pueden ver en la actualidad. Con HTML5 existen otras posibilidades para explotar sitios usando menos recursos, entrando conjuntamente en desuso el formato XHTML, dado que ya no sería necesaria su implementación.

Se trata de un sistema para formatear el “layout” o disposición de las páginas web, así como hacer algunos ajustes a su aspecto. Con HTML5 los navegadores como Firefox, Chrome, Explorer, Safari y más pueden saber cómo mostrar una determinada página web, saber dónde están los elementos, dónde poner las imágenes, o bien dónde ubicar el texto. En este sentido, el HTML5 no se diferencia demasiado de su predecesor. La diferencia principal, sin embargo, es el nivel de sofisticación del código que se podrá construir usando HTML5. Las etiquetas que se indicarán más adelante son una combinación del estándar original con HTML5.

Entre sus características se puede destacar:

- **Semántica:** nuevas etiquetas con el fin de indicar principalmente a los motores de búsqueda cuál es el contenido principal, de navegación y contextual. Ayuda a mejorar la indexación en caso de que los motores de búsqueda no detecten la importancia de cada texto en una página web.
- **Conectividad:** Permite comunicarse con el servidor de formas nuevas e innovadoras. Entre estas formas de conexión se encuentran los WebSockets, eventos de servidor enviados y WebRTC.
- **Sin conexión y almacenamiento:** Permite a las páginas web almacenar datos de forma local del lado del cliente y operar offline de una manera más eficiente.
- **Multimedia:** Se añade un mejor soporte sobre contenido multimedia permitiendo la reproducción de audio y video sin necesidad de componentes o plugins adicionales haciendo que la página reproduzca de

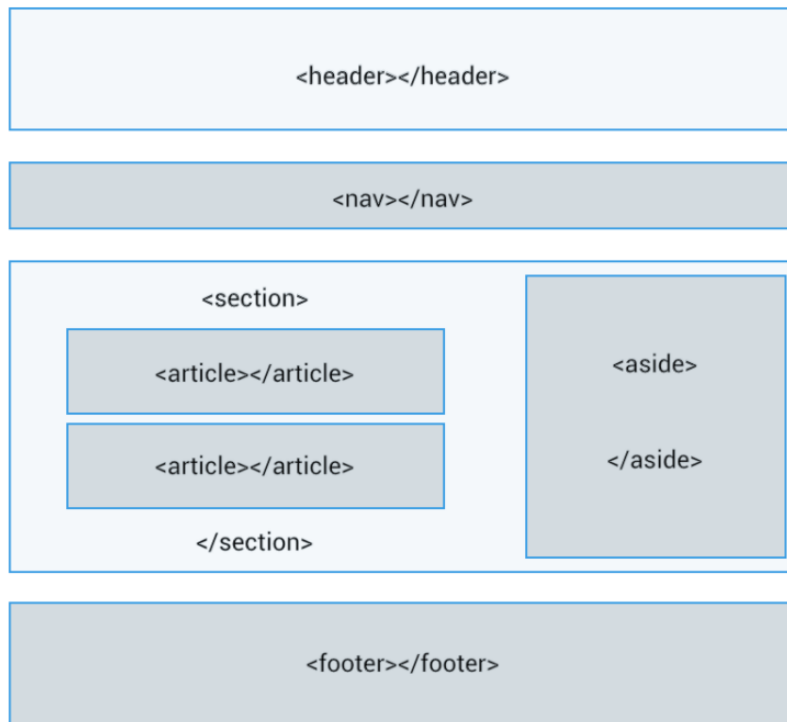


forma nativa dicho contenido, De esta forma la página tarda menos en cargar y su navegación es más rápida.

- **Gráficos y efectos 2D/3D:** Proporciona una amplia gama de nuevas características que se ocupan de los gráficos en la web como lo son canvas 2D, WebGL, SVG, etc.
- **Rendimiento e Integración:** En general, se puede prescindir de muchos plugins y mejorar el rendimiento y velocidad optimizando los recursos de hardware. En esta categoría se han implementado: Web Workers, XMLHttpRequest, Motores JIT compilación de JavaScript, History API, el atributo contentEditable, Arrastrar y soltar, Gestión del foco en HTML, Manejadores de protocolo basados en web, requestAnimationFrame, Fullscreen API, API Pointer Lock, Eventos en línea y fuera de línea.
- **Acceso al dispositivo:** Proporciona APIs para el uso de la cámara, eventos táctiles, geolocalización, detección de la orientación del hardware y Pointer Lock.
- **CSS3:** HTML5 con CSS3 se hace una mancuerna estupenda porque se podría hacer una gran variedad de animaciones y efectos para realizar diseños sofisticados y sorprendentes con mínimos recursos de procesamiento. Se ha ampliado para ser capaz de manejar elementos de estilo de una manera mucho más compleja. Se pueden agregar sombras, bordes redondeados, transiciones, uso de tipografías externas, mejoras en el manejo de columnas y cuadros de distribución flexible.
- **Compatibilidad:** El W3C es una comunidad internacional donde los miembros trabajan en conjunto para la desarrollo de los estándares web. Estas implementaciones aún se siguen desarrollando y podemos ver los avances aquí, cada navegador implementa estas especificaciones a su propio ritmo.

2.2.1.2.- Elementos, etiquetas y atributos

Una de las principales ventajas de HTML5 es la inclusión de elementos semánticos, o marcados semánticos, que ayudan a definir las distintas divisiones de una página web. En las versiones anteriores a HTML5 se solía utilizar la etiqueta <div> para realizar las divisiones de una página web, pero actualmente es recomendado utilizar los elementos semánticos ya que describen claramente su propósito. Como se puede observar en el esquema de la figura siguiente, a través de marcadores semánticos se puede diferenciar claramente el contenido que referencia cada etiqueta.



Cada uno de los elementos o etiquetas de HTML5 tiene sus propias propiedades. En la tabla siguiente se muestran las etiquetas más importantes que son parte de este estandar.

Etiqueta	Función
<!--...-->	Define un comentario
<!DOCTYPE>	Define el tipo de documento
<a>	Define un hipervínculo
<address>	Define la información de contacto del autor / propietario del documento
<article>	Define un artículo
<aside>	Define el contenido lateral del contenedor de una página
<audio>	Define contenido de sonido
	Define texto en negrita
<blockquote>	Define una sección que tiene otra fuente
<body>	Define el cuerpo del documento
 	Define un salto de línea
<button>	Define un botón clickeable
<canvas>	Se usa para dibujar gráficos en pantalla
<caption>	Define el título de una tabla
<code>	Define un trozo de código de programación
<datalist>	Especifica en un input una lista predefinida de opciones
<div>	Define una sección en un documento
<embed>	Define el contenedor de una aplicación externa (no HTML)
<fieldset>	Grupo de elementos relacionados en un formulario



<figure>	Especifica autocontenido
<footer>	Define el pie de página de un documento
<form>	Define un formulario html
<h1> a <h6>	Define encabezados o títulos
<head>	Define información acerca del documento
<header>	Define la sección de encabezado del documento
<hr>	Define un cambio de temática a partir de una línea dibujada
<html>	Define la raíz del documento
<i>	Define una parte del texto de modo alternativo
<iframe>	Define un frame en línea
	Define una imagen
<input>	Define un control de entrada de texto
	Define un ítem de una lista
<link>	Define la relación entre un documento y un recurso externo (generalmente con hojas de estilo)
<meta>	Define un metadato de un documento
<nav>	Define un link de navegación
	Define una lista ordenada
<optgroup>	Define un grupo de opciones relacionadas en una lista desplegable
<option>	Define una opción en una lista desplegable
<p>	Define un párrafo
<param>	Define un parámetro para un objeto
<script>	Define un script del lado cliente
<section>	Define una sección de un documento
<select>	Define un drop-down list
	Define una pequeña sección de un documento
	Define un texto en negrita
<style>	Define un estilo para la información de un documento
<table>	Define una tabla
<tbody>	Define el cuerpo de una tabla
<td>	Define una celda en una tabla
<textarea>	Define un control de entrada de múltiples líneas
<tfoot>	Agrupar los footer contenidos en una tabla
<th>	Define una celda de encabezado en una tabla
<thead>	Agrupar los encabezados de una tabla
<time>	Define fecha / hora
<title>	Define un título para el documento
<tr>	Define una fila en una tabla
	Define una lista desordenada
<video>	Define un vídeo o película

Estas etiquetas o tags se componen y contienen otras propiedades, como son los atributos y el contenido.



HTML define un total de 91 etiquetas, de las cuales 10 se consideran obsoletas. Sin embargo, una etiqueta por sí sola a veces no contiene la suficiente información para estar completamente definida. Para ello se cuenta con los atributos: pares nombre-valor separados por "=" y escritos en la etiqueta inicial de un elemento después del nombre del elemento. El valor puede estar encerrado entre "comillas dobles" o 'simples'. Existen, también, algunos atributos que afectan al elemento por su presencia en la etiqueta de inicio, como puede ser el atributo ismap para el elemento .

Esta sería la estructura general de una línea de código en lenguaje HTML:

```
<tag attribute1="value1" attribute2="value2">content</tag>
```

O lo que es lo mismo, con un ejemplo:

```
<a href="http://www.enlace.com" target="_blank">Ejemplo de enlace</a>
```

Donde:

- <a> es la etiqueta o tag inicial y la etiqueta de cierre.
- href y target son los atributos.
- http://www.enlace.com y _blank son las variables.
- Ejemplo de enlace es el contenido.

Aunque también existen elementos vacíos que no necesitan tag de cierre, cuya estructura sería ésta:

```
<tag attribute1="value1" attribute2="value2" />
```

Estos elementos vacíos no constan de contenido, como por ejemplo, los tags
 o .

2.2.1.3.- Estructura básica de un documento: html, head, body, meta, title, link

Debido a que los navegadores son capaces de procesar diferentes tipos de archivos, lo primero que se debe hacer en la construcción de un documento HTML es indicar su tipo. Para tener seguridad de que el contenido de los documentos sea interpretado correctamente como código HTML, se debe agregar la declaración <!DOCTYPE> al comienzo del archivo. Esta declaración, similar en



formato a las etiquetas HTML, se requiere al comienzo de cada documento para ayudar al navegador a decidir cómo debe generar la página web.

Los elementos HTML conforman una estructura de tipo árbol con el elemento `<html>` como su raíz. Esta estructura presenta múltiples niveles de organización, con algunos elementos a cargo de definir secciones generales del documento y otros encargados de representar secciones menores o contenido. Los siguientes son los elementos disponibles para definir la columna vertebral de la estructura y facilitar la información que el navegador necesita para mostrar la página en la pantalla.

<code><html></code>	Este elemento delimita el código HTML. Puede incluir el atributo <code>lang</code> para definir el idioma del contenido del documento.
<code><head></code>	Este elemento se usa para definir la información necesaria para configurar la página web, como el título, el tipo de codificación de caracteres y los archivos externos requeridos por el documento.
<code><body></code>	Este elemento delimita el contenido del documento (la parte visible de la página).

El código HTML insertado entre las etiquetas `<html>` se tiene que dividir en dos secciones principales: la cabecera y el cuerpo. Por supuesto, la cabecera va primero y, al igual que el resto de los elementos estructurales, está compuesta por etiquetas de apertura y cierre.

Entre las etiquetas `<head>` es necesario definir el título de la página web, declarar el tipo de codificación de caracteres, facilitar información general acerca del documento, e incorporar los archivos externos con estilos y códigos necesarios para generar la página. Excepto por el título e iconos, el resto de la información insertada en medio de estas etiquetas no es visible para el usuario. Entre las etiquetas que son parte de la cabecera están descritas en la tabla siguiente.

<code><title></code>	Este elemento define el título de la página.
<code><base></code>	Este elemento define la URL usada por el navegador para establecer la ubicación real de las URL relativas. El elemento debe incluir el atributo <code>href</code> para declarar la URL base. Cuando se declara este elemento, en lugar de la URL actual, el navegador usa la URL asignada al atributo



	href para completar las URL relativas.
<meta>	Este elemento representa metadatos asociados con el documento, como la descripción del documento, palabras claves, el tipo de codificación de caracteres, etc. El elemento puede incluir los atributos name para describir el tipo de metadata , content para especificar el valor, y charset para declarar el tipo de codificación de caracteres a utilizar para procesar el contenido.
<link>	Este elemento especifica la relación entre el documento y un recurso externo (generalmente usado para cargar archivos CSS). El elemento puede incluir los atributos href para declarar la ubicación del recurso, rel para definir el tipo de relación, media para especificar el medio al que el recurso está asociado (pantalla, impresora, etc.), y type y sizes para declarar el tipo de recurso y su tamaño (usado a menudo para cargar iconos).
<style>	Este elemento se usa para declarar estilos CSS dentro del documento
<script>	Este elemento se usa para cargar o declarar código JavaScript

2.2.1.4.- Estructura del cuerpo: header, nav, section, aside, footer

HTML siempre ha ofrecido diferentes maneras de construir y organizar la información en el cuerpo del documento. Uno de los primeros elementos utilizados con este propósito fue `<table>` (tabla). Este elemento permitía a los desarrolladores organizar datos, textos, imágenes, así como herramientas en filas y columnas de celdas.

Con la introducción de CSS, la estructura generada por estas tablas ya no resultaba práctica, por lo que los desarrolladores comenzaron a implementar un elemento más flexible llamado `<div>` (división). Pero `<div>`, así como `<table>`, no facilita demasiada información acerca de las partes del cuerpo que representa. Cualquier cosa, desde imágenes hasta menús, texto, enlaces, códigos o formularios, se puede insertar entre las etiquetas de apertura y cierre de un elemento `<div>`.



En otras palabras, el nombre `div` solo especifica una división en el cuerpo, como una celda en una tabla, pero no ofrece ninguna pista acerca del tipo de división que está creando, cuál es su propósito o qué contiene. Esta es la razón por la que HTML5 introdujo nuevos elementos con nombres más descriptivos que permiten a los desarrolladores identificar cada parte del documento. Estos elementos no solo ayudan a los desarrolladores a crear el documento, sino que además informan al navegador sobre el propósito de cada sección. La siguiente lista incluye todos los elementos disponibles para definir la estructura del cuerpo.

<code><div></code>	Este elemento define una división genérica. Se usa cuando no se puede aplicar ningún otro elemento.
<code><main></code>	Este elemento define una división que contiene el contenido principal del documento (el contenido que representa el tema central de la página).
<code><nav></code>	Este elemento define una división que contiene ayuda para la navegación, como el menú principal de la página o bloques de enlaces necesarios para navegar en el sitio web.
<code><section></code>	Este elemento define una sección genérica. Se usa frecuentemente para separar contenido temático, o para generar columnas o bloques que ayudan a organizar el contenido principal. <small>[SEP]</small>
<code><aside></code>	Este elemento define una división que contiene información relacionada con el contenido principal pero que no es parte del mismo, como referencias a artículos o enlaces que apuntan a publicaciones anteriores.
<code><article></code>	Este elemento representa un artículo independiente, como un mensaje de foro, el artículo de una revista, una entrada de un blog, un comentario, etc.
<code><header></code>	Este elemento define la cabecera del cuerpo o de secciones dentro del cuerpo. <small>[SEP]</small>
<code><footer></code>	Este elemento define el pie del cuerpo o de secciones dentro del cuerpo.

Estos elementos han sido definidos con el propósito de representar secciones específicas de una página web. Aunque son flexibles y se pueden implementar en

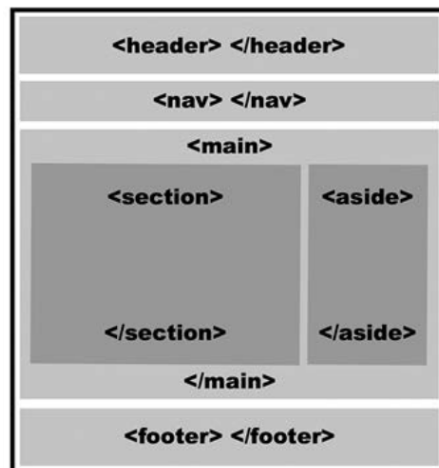


diferentes partes del diseño, todos siguen un patrón que se encuentra comúnmente en la mayoría de los sitios web. La figura que se muestra a continuación ilustra este tipo de diseño.



A pesar de que cada desarrollador crea sus propios diseños, en general se puede describir todo sitio web considerando estas secciones. En la barra superior, descrita como cabecera en la figura ubicamos el logo, el nombre del sitio, los subtítulos y una descripción breve del sitio o página web en cuestión. En la barra de navegación situada debajo es donde la mayoría de los desarrolladores ofrecen un menú o una lista de enlaces para navegar en el sitio. El contenido relevante de la página se ubica en el medio del diseño, donde generalmente existen artículos o noticias, y también enlaces a documentos relacionados o recursos. En el ejemplo esta sección se ha dividido en dos columnas, información principal y barra lateral, pero los diseñadores la adaptan a sus necesidades insertando columnas adicionales o dividiendo las columnas en bloques más pequeños. En la parte inferior de un diseño tradicional, existe otra barra llamada barra institucional. Se llama de este modo porque en este área es donde se muestra la información general acerca del sitio web, el autor, la compañía, los enlaces relacionados con reglas de uso, términos y condiciones, el mapa del sitio, etc.

Como se mencionó anteriormente, los elementos de HTML5 se han diseñado siguiendo este patrón. En la figura siguiente se aplican los elementos introducidos anteriormente para definir el diseño de la figura anterior.



2.2.1.5.- Encabezados: H1...H6

Las etiquetas `<h1>` a `<h6>` se utilizan para definir encabezados HTML. En el caso de `<h1>`, ésta define el encabezado más importante, mientras que `<h6>` define el encabezado menos importante.

Se debe utilizar solo un `<h1>` por página; esto debe representar el encabezado / tema principal de toda la página. Además, no se deben omitir los niveles de encabezado: se recomienda comenzar con `<h1>`, luego usar `<h2>` y así sucesivamente.

2.2.1.6.- Enlaces o hipervínculos

Los enlaces o hipervínculos, también llamados hipertextos son los textos o los objetos sobre los que es posible hacer clic para que redirijan al usuario a otra parte del documento, a otra página web en el mismo sitio o a otra página de Internet, entre otras funciones.

Cuando un visitante hace clic en el hipervínculo, el destino se mostrará en su navegador web, se abrirá o se ejecutará, en función del tipo de destino. El destino es con frecuencia otra página web, pero también puede ser una imagen, un archivo multimedia, un documento de Microsoft Office, un documento PDF, una dirección de correo electrónico o un programa.

Por tanto, con los hipervínculos se puede establecer “saltos” desde una página a otra, incluso a puntos concretos dentro de páginas con contenidos extensos. Los hipervínculos pueden asociarse a imágenes o a textos de modo que haciendo clic



sobre ellos con el botón izquierdo del ratón se accede automáticamente al destino asociado a ellos.

Lo más habitual es que el cursor tome la forma de una pequeña mano cuando pasa sobre un hipervínculo.

Existen distintos tipos de enlaces:

- **Hipervínculos locales:** Los hipervínculos locales o marcadores son enlaces dentro de la misma página. Es decir, al hacer clic en uno de ellos llevará al usuario a una posición distinta dentro de la misma página que está visualizando.
- **Hipervínculos externos:** Un hipervínculo externo es un vínculo a otro sitio web en Internet (sitio externo). Es un vínculo a cualquier otro lugar fuera del sitio actual. Al agregar un vínculo externo, se escribe la dirección completa de la página incluido `http://www....`. Estas rutas que incluyen `http://...` se denominan rutas absolutas.
- **Imágenes como enlaces:** Para poner una imagen como enlace basta con crear un enlace y añadir en su interior en lugar de texto, una imagen.

2.2.1.7.- Imágenes

Las imágenes pueden ser consideradas el segundo medio más importante en la Web. HTML incluye los siguientes elementos para introducir imágenes en los documentos que conforman un portal.

	Este elemento inserta una imagen en el documento. El elemento requiere del atributo src para especificar la URL del archivo con la imagen que queremos incorporar.
<picture>	Este elemento inserta una imagen en el documento. Trabaja junto con el elemento <source> para ofrecer múltiples imágenes en diferentes resoluciones. Es útil para crear sitios web adaptables.
<figure>	Este elemento representa contenido asociado con el contenido principal, pero que se puede eliminar sin que se vea afectado, como fotos, vídeos, etc.
<figcaption>	Este elemento introduce un título para el elemento <figure> . <small>[1] [SEP]</small>



2.2.1.8.- Listas anidadas

A menudo la información se debe representar como una lista de ítems. Por ejemplo, muchos sitios web incluyen listados de libros, películas, o términos y descripciones. Para crear estos listados, HTML ofrece los siguientes elementos.

	Este elemento crea una lista de ítems sin orden. Está compuesto por etiquetas de apertura y cierre para agrupar los ítems (y) y trabaja junto con el elemento para definir cada uno de los ítems de la lista.
	Este elemento crea una lista ordenada de ítems. Está compuesto por etiquetas de apertura y cierre para agrupar los ítems (y) y trabaja junto con el elemento para definir los ítems de la lista. Este elemento puede incluir los atributos reversed para invertir el orden de los indicadores, start para determinar el valor desde el cual los indicadores tienen que comenzar a contar y type para determinar el tipo de indicador que queremos usar. Los valores disponibles para el atributo type son 1 (números), a (letras minúsculas), A (letras mayúsculas), i (números romanos en minúsculas) e I (números romanos en mayúsculas).
<dl>	Este elemento crea una lista de términos y descripciones. El elemento trabaja junto con los elementos <dt> y <dd> para definir los ítems de la lista. El elemento <dl> define la lista, el elemento <dt> define los términos y el elemento <dd> define las descripciones.

Los siguientes elementos se han diseñado con propósitos diferentes, pero también se utilizan frecuentemente para construir listas de ítems.

<blockquote>	Este elemento representa un bloque de texto que incluye una cita tomada de otro texto en el documento.
<details>	Este elemento crea una herramienta que se expande cuando se hace clic en ella para mostrar información adicional. La parte visible se define con el elemento <summary> , y se pueden usar elementos comunes como <p> para definir el contenido.



2.2.1.9.- Tablas

Las tablas organizan información en filas y columnas. Debido a sus características, se usaron durante mucho tiempo para estructurar documentos HTML, pero con la introducción de CSS, los desarrolladores pudieron lograr el mismo efecto implementando otros elementos.

Aunque ya no se recomienda usar tablas para definir la estructura de un documento, todavía se utilizan para presentar información tabular, como estadísticas o especificaciones técnicas, por ejemplo. HTML incluye varios elementos para crear una tabla. Los siguientes son los más utilizados.

<table>	Este elemento define una tabla. Incluye etiquetas de apertura y cierre para agrupar el resto de los elementos que definen la tabla.
<tr>	Este elemento define una fila de celdas. Incluye etiquetas de apertura y cierre para agrupar las celdas.
<td>	Este elemento define una celda. Incluye etiquetas de apertura y cierre para delimitar el contenido de la celda y puede incluir los atributos colspan y rowspan para indicar cuántas columnas y filas ocupa la celda.
<th>	Este elemento define una celda para la cabecera de la tabla. Incluye etiquetas de apertura y cierre para delimitar el contenido de la celda y puede incluir los atributos colspan y rowspan para indicar cuántas columnas y filas ocupa la celda. <small>[L] [SEP]</small>

2.1.1.10.- Formularios

Los formularios son herramientas que es posible incluir en un documento para permitir a los usuarios insertar información, tomar decisiones, comunicar datos y cambiar el comportamiento de una aplicación. El propósito principal de los formularios es permitir al usuario seleccionar o insertar información y enviarla al servidor para ser procesada.

Los formularios pueden presentar varias herramientas que permiten al usuario interactuar con el documento, incluidos campos de texto, casillas de control,




menús desplegables y botones. Cada una de estas herramientas se representa por un elemento y el formulario queda definido por el elemento `<form>`, que incluye etiquetas de apertura y cierre para agrupar al resto de los elementos y requiere de algunos atributos para determinar cómo se envía la información al servidor. Los atributos posibles se indican en la tabla a continuación.

name	Este atributo especifica el nombre del formulario. También se encuentra disponible para otros elementos, pero es particularmente útil para elementos de formulario, como veremos más adelante.
method	Este atributo determina el método a utilizar para enviar la información al servidor. Existen dos valores disponibles: GET y POST. El método GET se usa para enviar una cantidad limitada de información de forma pública (los datos son incluidos en la URL, la cual no puede contener más de 255 caracteres). Por otro lado, el método POST se utiliza para enviar una cantidad ilimitada de información de forma privada (los datos no son visibles al usuario y pueden tener la longitud que necesitemos).
action	Este atributo declara la URL del archivo en el servidor que va a procesar la información enviada por el formulario.
target	Este atributo determina dónde se mostrará la respuesta recibida desde el servidor. Los valores disponibles son <code>_blank</code> (nueva ventana), <code>_self</code> (mismo recuadro), <code>_parent</code> (recuadro padre), y <code>_top</code> (la ventana que contiene el recuadro). El valor <code>_self</code> se declara por defecto, lo que significa que la respuesta recibida desde el servidor se mostrará en la misma ventana. <code><input type="button" value="SEP" /></code>
enctype	Este atributo declara la codificación aplicada a los datos que envía el formulario. Puede tomar tres valores: <code>application/x-www-form-urlencoded</code> (los caracteres son codificados), <code>multipart/form-data</code> (los caracteres no son codificados), <code>text/plain</code> (solo los espacios son codificados). El primer valor se asigna por defecto.
accept-charset	Este atributo declara el tipo de codificación aplicada al texto del formulario. Los valores más comunes son UTF-8 e ISO-8859-1. El valor por defecto se asigna al documento con el elemento <code><meta></code> .



Un formulario puede incluir diferentes herramientas para permitir al usuario seleccionar o insertar información. HTML incluye múltiples elementos para crear estas herramientas. Los siguientes son los más utilizados.

<input>	Este elemento crea un campo de entrada. Puede recibir diferentes tipos de entradas, dependiendo del valor del atributo type.
<textarea>	Este elemento crea un campo de entrada para insertar múltiples líneas de texto. El tamaño se puede declarar en números enteros usando los atributos rows y cols, o en píxeles con estilos CSS.
<select>	Este elemento crea una lista de opciones que el usuario puede elegir. Trabaja junto con el elemento <option> para definir cada opción y el elemento <optgroup> para organizar las opciones en grupos.
<button>	Este elemento crea un botón. Incluye el atributo type para definir el propósito del botón. Los valores disponibles son submit para enviar el formulario (por defecto), reset para reiniciar el formulario, y button para realizar tareas personalizadas. 
<output>	Este elemento representa un resultado producido por el formulario. Se implementa por medio de código JavaScript para mostrar el resultado de una operación al usuario.
<meter>	Este elemento representa una medida o el valor actual de un rango.
<progress>	Este elemento representa el progreso de una operación.
<datalist>	Este elemento crea un listado de valores disponibles para otros controles. Trabaja junto con el elemento <option> para definir cada valor.
<label>	Este elemento crea una etiqueta para identificar un elemento de formulario.
<fieldset>	Este elemento agrupa otros elementos de formulario. Se usa para crear secciones dentro de formularios extensos. El elemento puede contener un elemento <legend> para



	definir el título de la sección.
--	----------------------------------

El elemento `<input>` es el más versátil de todos. Este elemento genera un campo de entrada en el que el usuario puede seleccionar o insertar información, pero puede adoptar diferentes características y aceptar varios tipos de valores dependiendo del valor de su atributo `type`. Los siguientes son los valores disponibles para este atributo.

- **text:** Este valor genera un campo de entrada para insertar texto genérico.
- **email:** Este valor genera un campo de entrada para insertar cuentas de correo.
- **search:** Este valor genera un campo de entrada para insertar términos de búsqueda.
- **url:** Este valor genera un campo de entrada para insertar URL.
- **tel:** Este valor genera un campo de entrada para insertar números de teléfono.
- **number:** Este valor genera un campo de entrada para insertar números.
- **range:** Este valor genera un campo de entrada para insertar un rango de números.
- **date:** Este valor genera un campo de entrada para insertar una fecha.
- **datetime-local:** Este valor genera un campo de entrada para insertar fecha y hora.
- **week:** Este valor genera un campo de entrada para insertar el número de la semana (dentro del año).
- **month:** Este valor genera un campo de entrada para insertar el número del mes.
- **time:** Este valor genera un campo de entrada para insertar una hora (horas y minutos).
- **hidden:** Este valor oculta el campo de entrada. Se usa para enviar información complementaria al servidor.
- **password:** Este valor genera un campo de entrada para insertar una clave. Reemplaza los caracteres insertados con estrellas o puntos para ocultar información sensible.
- **color:** Este valor genera un campo de entrada para insertar un color.
- **checkbox:** Este valor genera una casilla de control que permite al usuario activar o desactivar una opción.
- **radio:** Este valor genera un botón de opción para seleccionar una opción de varias posibles.
- **file:** Este valor genera un campo de entrada para seleccionar un archivo en el ordenador del usuario.



- **button:** Este valor genera un botón. El botón trabaja como el elemento `<button>` de tipo `button`. No realiza ninguna acción por defecto; la acción debe ser definida desde JavaScript, como veremos en próximos capítulos.
- **submit:** Este valor genera un botón para enviar el formulario.
- **reset:** Este valor genera un botón para reiniciar el formulario.
- **image:** Este valor carga una imagen que se usa como botón para enviar el formulario. Un elemento `<input>` de este tipo debe incluir el atributo `src` para especificar la URL de la imagen.

2.2.1.11.- Elemento DIV

La etiqueta `<div>` define una división. Esta etiqueta permite agrupar varios elementos de bloque (párrafos, encabezados, listas, tablas, divisiones, etc). En principio, los navegadores no muestran nada especial cuando se crea una división, salvo que se dé formato a la división con la hoja de estilo.

En secciones anteriores se habló de las etiquetas semánticas. Es importante recordar que todas ellas, por naturaleza, son `div`. Esto implica que su comportamiento será el mismo en la práctica.



Anexo: Referencias

[1] ¿Qué es el HTML5?

Referencia: <http://www.elpadawan.com/html/definicion-caracteristicas-y-nuevas-etiquetas-de-html5>

[2] Etiquetas, atributos y elementos

Referencia: <https://www.arkaitzgarro.com/xhtml/capitulo-3.html>

[3] HTML5: Estructura básica y elementos semánticos

Referencia: <https://www.eniun.com/html5-estructura-basica-elementos-semanticos/>

[4] Etiquetas semánticas del HTML5

Referencia: <https://desarrolloweb.com/articulos/etiquetas-semanticas-html5.html>

[5] Hipervínculos, links o enlaces HTML

Referencia:

https://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=508