

Unidad 2

HTML (HyperText Markup Language)

Introducción a HTML

La creación de una página web es como construir un mundo digital, donde cada elemento tiene su lugar y propósito. En este universo virtual, **HTML** (HyperText Markup Language) sirve como el lenguaje fundamental que define la estructura y el contenido de cada página web.

HTML es el cimiento sobre el cual se construyen las páginas web modernas. Es un lenguaje de marcado que permite a los desarrolladores web organizar y presentar información en forma de documentos hipertextuales, los cuales pueden contener texto, imágenes, enlaces, formularios, videos y mucho más. Desde los primeros días de la World Wide Web, HTML ha sido el lenguaje primario utilizado para crear y dar forma al contenido que vemos en nuestros navegadores.

En su esencia, HTML proporciona una forma estructurada de organizar la información, utilizando etiquetas que definen el significado y la función de cada elemento en una página web. Estas etiquetas permiten al navegador interpretar y mostrar el contenido de manera coherente y legible para los usuarios. Desde los encabezados que definen el título de una página hasta los párrafos que presentan información, HTML proporciona las herramientas necesarias para crear una experiencia de navegación coherente y accesible.

A lo largo de los años, HTML ha evolucionado para adaptarse a las necesidades cambiantes de la web. Con la introducción de nuevas especificaciones y estándares, como **HTML5**, el lenguaje ha ganado capacidades adicionales para la creación de contenido multimedia, la accesibilidad mejorada y la interactividad avanzada. Hoy en día, HTML sigue siendo el lenguaje fundamental utilizado en la construcción de páginas web, sirviendo como la base sobre la cual se construyen otras tecnologías como CSS (Cascading Style Sheets) y JavaScript.

La **evolución de HTML** ha sido un viaje fascinante que ha transformado la forma en que interactuamos con la web y cómo se presenta la información en línea. Desde sus humildes comienzos hasta su estado actual como uno de los lenguajes de marcado más poderosos y versátiles, la historia de HTML es un testimonio del constante progreso y la innovación en el mundo de la tecnología web.

HTML 1.0 - Los Primeros Pasos:

HTML (HyperText Markup Language) hizo su debut en 1991 como una especificación simple que permitía a los usuarios crear documentos hipertextuales. En esta primera versión, HTML proporcionaba un conjunto básico de etiquetas para

definir la estructura y el contenido de una página web, incluyendo títulos, párrafos, enlaces y listas.

HTML 2.0 - La Estandarización:

HTML 2.0, lanzado en 1995, marcó un importante hito en la evolución de HTML al establecer un conjunto de estándares y convenciones para la creación de páginas web. Esta versión introdujo nuevas etiquetas y atributos, así como la especificación de formularios para recopilar datos de los usuarios.

HTML 3.2 - La Era de las Tablas y Frames:

HTML 3.2, publicado en 1997, introdujo características avanzadas como tablas, marcos (frames) y formularios mejorados. Estas adiciones permitieron una mayor flexibilidad en el diseño y la presentación de contenido en línea, lo que llevó a la creación de sitios web más complejos y visualmente atractivos.

HTML 4.01 - La Madurez:

HTML 4.01, lanzado en 1999, consolidó las especificaciones anteriores y añadió nuevas características como estilos en línea, hojas de estilo en cascada (CSS) y compatibilidad con XML. Esta versión también introdujo la noción de elementos semánticos, que permitían a los desarrolladores etiquetar el contenido de manera más significativa.

XHTML - El Puente hacia el Futuro:

XHTML (Extensible HyperText Markup Language) surgió a principios de los años 2000 como una versión más estricta y bien formada de HTML, basada en XML. XHTML combinó la familiaridad de HTML con la sintaxis rigurosa de XML, sentando las bases para la transición hacia tecnologías web más avanzadas.

HTML5 - La Revolución Web:

HTML5, lanzado en 2014, representa la última evolución de HTML hasta la fecha. Esta versión introdujo una amplia gama de nuevas características y API, incluyendo elementos multimedia nativos (como audio y video), gráficos vectoriales (SVG), almacenamiento local (localStorage) y mucho más. HTML5 también impulsó el desarrollo de aplicaciones web modernas y responsivas, estableciendo un nuevo estándar para la creación de contenido web interactivo y dinámico.

Estructura básica de un documento HTML

```
<!DOCTYPE html>
<html>
<head>
<title>Titulo</title>
</head>
<body>

</body>
</html>
```

La estructura básica de un documento HTML sirve como el armazón fundamental sobre el cual se construye cualquier página web. Es como el esqueleto que sostiene todo el contenido y define cómo se presenta en un navegador web.

A continuación, se describen los elementos principales de esta estructura:

- **Etiqueta <!DOCTYPE html>:** Esta etiqueta define el tipo de documento y la versión de HTML que se está utilizando. En HTML5, esta etiqueta es simple y concisa: <!DOCTYPE html>. Indica al navegador que el documento se está escribiendo en HTML5.
- **Etiqueta <html>:** La etiqueta <html> envuelve todo el contenido de la página web. Es el elemento principal que indica al navegador que el documento es un documento HTML. Se utiliza para definir el inicio y el final del documento HTML.
- **Etiqueta <head>:** La etiqueta <head> contiene información meta sobre el documento, como el título de la página, enlaces a hojas de estilo CSS, referencias a scripts JavaScript, metadatos para SEO, etc. Esta sección no se muestra en la ventana del navegador, pero proporciona información importante sobre la página.
- **Etiqueta <title>:** La etiqueta <title> se encuentra dentro del elemento <head> y define el título de la página que se muestra en la barra de título del navegador o en la pestaña del navegador. Es importante para el SEO y para proporcionar una descripción clara de la página a los usuarios.
- **Etiqueta <body>:** La etiqueta <body> contiene todo el contenido visible de la página web, como texto, imágenes, enlaces, formularios, etc. Todo lo que se muestra en la ventana del navegador se coloca dentro de esta etiqueta. Es el lugar donde se presenta el contenido principal de la página.

Nota: Intenta ir armando un archivo de texto con extensión html e ir probando los ejemplos que se irán detallando.

¿Qué es un elemento HTML?

Un elemento HTML se define mediante una etiqueta de inicio, algo de contenido y una etiqueta de cierre:

`<nombredeetiqueta>`El contenido va aquí...`</nombredeetiqueta>`

El elemento HTML es todo, desde la etiqueta de inicio hasta la etiqueta de cierre:

`<h1>`Mi primer encabezado`</h1>`

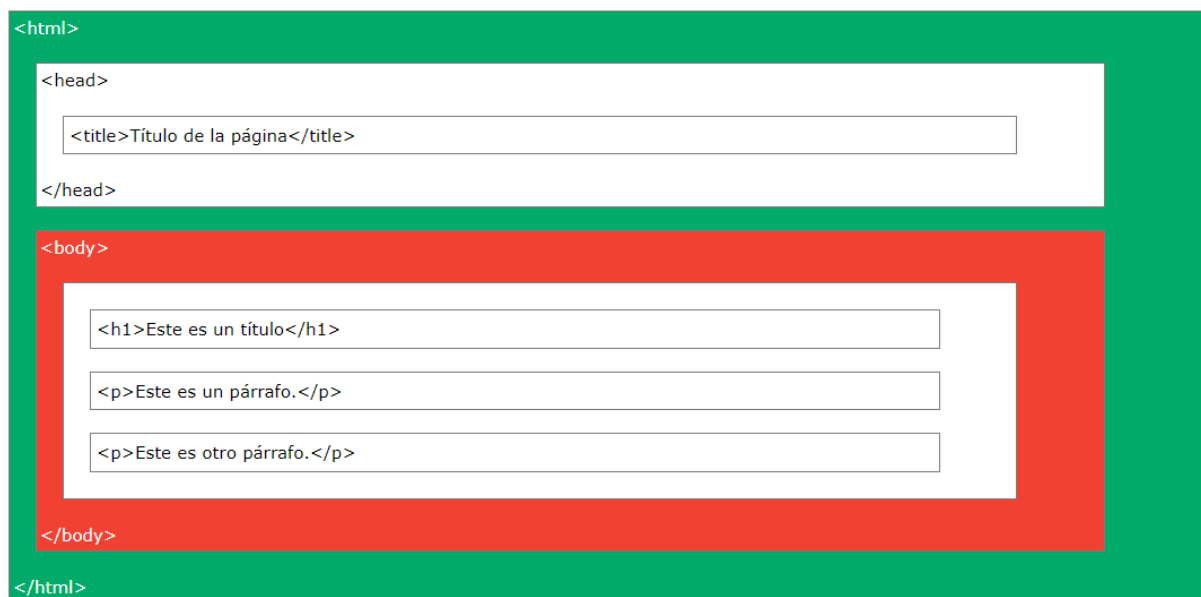
`<p>`Mi primer párrafo.`</p>`

Recuerda que para poder leer un documento html necesitas un navegador web (Chrome, Edge, Firefox, Safari, etc).

Un navegador no muestra las etiquetas HTML, pero las utiliza para determinar cómo mostrar el documento.

HTML Estructura de la página

Una estructura básica de un documento html podría ser la siguiente:



Etiquetas HTML fundamentales

La declaración `<!DOCTYPE>`

La declaración `<!DOCTYPE>` es una parte fundamental de un documento HTML ya que define el tipo de documento y su versión. Esta declaración informa al navegador web sobre cómo interpretar y renderizar el contenido HTML del documento.

En HTML5, la declaración <!DOCTYPE> es bastante simple y concisa:

```
<!DOCTYPE html>
```

En HTML 4, la declaración <!DOCTYPE> solía hacer referencia a un Documento de Tipo de Documento (DTD) externo para especificar la versión de HTML que se estaba utilizando.

Por ej: ***<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">***

HTML5 se diseñó para ser más simple y fácil de usar, eliminando la necesidad de referenciar un DTD externo. En otras palabras, en HTML5, la declaración <!DOCTYPE> se simplificó a <!DOCTYPE html>, y no requiere un DTD externo.

El propósito principal de la declaración <!DOCTYPE> es garantizar que el navegador interprete correctamente el contenido HTML del documento de acuerdo con los estándares establecidos. Sin esta declaración, el navegador podría entrar en modo de "cuasi-estándar" o "modo de compatibilidad", lo que puede llevar a resultados inesperados o inconsistentes en la representación del contenido.

HTML Headings

Los encabezados en HTML (también conocidos como headings en inglés) son etiquetas que se utilizan para definir la estructura jerárquica y el contenido de una página web. Los encabezados van desde el nivel 1 hasta el nivel 6, donde el nivel 1 (<h1>) es el más importante y el nivel 6 (<h6>) es el menos importante.

```
<h1>Esto se dirige 1</h1>
<h2>Esto se dirige 2</h2>
<h3>Esto se dirige 3</h3>
```

HTML Párrafos

Los párrafos en HTML se definen utilizando la etiqueta <p>. Esta etiqueta se utiliza para indicar que un bloque de texto forma un párrafo dentro de un documento HTML. Los párrafos son una forma fundamental de organizar y presentar contenido textual en una página web, ya que ayudan a estructurar y dividir el contenido en unidades significativas.

Es importante tener en cuenta que los párrafos son elementos de bloque en HTML, lo que significa que ocupan todo el ancho disponible de su contenedor principal y se muestran en una línea separada. Esto ayuda a mejorar la legibilidad y la organización del contenido en la página.

```
<p>Este es un párrafo.</p>
<p>Este es otro párrafo.</p>
```

HTML Enlaces

Los enlaces en HTML se crean utilizando la etiqueta `<a>`. Los enlaces permiten a los usuarios navegar de una página a otra, o dentro de la misma página, haciendo clic en el texto o en una imagen asociada al enlace.

Aquí tienes un ejemplo de cómo crear un enlace básico en HTML:

```
<a href="https://www.ejemplo.com">Visitar Ejemplo.com</a>
```

En este ejemplo, `<a>` es la etiqueta de inicio del enlace, y `` es la etiqueta de cierre del enlace. El atributo `href` especifica la dirección URL a la que el enlace debe dirigir al usuario cuando se hace clic en él. El texto entre las etiquetas de inicio y cierre del enlace es el texto del enlace que se muestra al usuario.

Además de enlazar a páginas web externas, los enlaces también pueden apuntar a ubicaciones dentro de la misma página utilizando el atributo `href` con un identificador. Por ejemplo:

```
<a href="#seccion1">Ir a la Sección 1</a>
```

HTML Imágenes

Las imágenes en HTML se insertan utilizando la etiqueta ``. Esta etiqueta se utiliza para mostrar imágenes en una página web, lo que permite a los desarrolladores integrar gráficos, fotografías y otros elementos visuales en el contenido de la página.

```

```

En este ejemplo:

- **src** es un atributo obligatorio que especifica la ruta de la imagen. Puede ser una ruta relativa (a la imagen dentro del mismo sitio web) o una URL absoluta (a una imagen en otro sitio web).
- **alt** es un atributo opcional que proporciona un texto alternativo para la imagen. Este texto se muestra si la imagen no se carga correctamente o si el usuario utiliza un lector de pantalla. Proporcionar un texto alternativo es importante para la accesibilidad y la usabilidad del sitio web.
- Además de estos dos atributos, la etiqueta `` puede tener otros atributos opcionales, como **width (ancho)** y **height (alto)**, que especifican las dimensiones de la imagen en píxeles.

Ej:

```

```

Estructura semántica: encabezados, párrafos, listas, etc.

HTML los encabezados se definen con las etiquetas <h1> to <h6> tags.

<h1> define el título más importante. <h6> define el encabezado menos importante.

Los títulos son importantes. Los motores de búsqueda utilizan los encabezados para indexar la estructura y el contenido de sus páginas web.

Los usuarios suelen hojear una página por sus títulos. Es importante utilizar títulos para mostrar la estructura del documento.

Los <h1> encabezados deben usarse para los encabezados principales, seguidos de los encabezados <h2> y luego los menos importantes <h3>, y así sucesivamente.

Cada encabezado HTML tiene un tamaño predeterminado. Sin embargo, puede especificar el tamaño de cualquier encabezado con el atributo style, utilizando la propiedad **CSS** font-size:

```
<h1 style="font-size:60px;">Heading 1</h1>
```

El elemento HTML <p> define un párrafo.

Un párrafo siempre comienza en una nueva línea y los navegadores agregan automáticamente algunos espacios en blanco (un margen) antes y después de un párrafo.

No puedes estar seguro de cómo se mostrará HTML. Las pantallas grandes o pequeñas y las ventanas redimensionadas crearán resultados diferentes.

Con HTML, no puedes cambiar la visualización agregando espacios o líneas adicionales en tu código HTML.

El navegador eliminará automáticamente los espacios y líneas adicionales cuando se muestre la página.

La etiqueta <hr> define una ruptura temática en una página HTML y suele mostrarse como una regla horizontal.

El elemento <hr> se utiliza para separar contenido (o definir un cambio) en una página HTML:

```
<p>Este es un texto.</p>
<hr>
<h2>Esto se dirige 2</h2>
```

El elemento HTML **
** define un salto de línea.

Utilice **
** si desea un salto de línea (una nueva línea) sin comenzar un nuevo párrafo:

```
<p>Este es<br>un párrafo<br>con saltos de línea.</p>
```

El elemento HTML **<pre>** define texto preformateado.

El texto dentro de un elemento **<pre>** se muestra en una fuente de ancho fijo (normalmente Courier) y conserva espacios y saltos de línea:

```
<pre>
My Bonnie lies over the ocean.

Oh, bring back my Bonnie to me.
</pre>
```

HTML Elementos de formato

Los elementos de formato se diseñaron para mostrar tipos especiales de texto:

- ****: Esta etiqueta se utiliza para aplicar negrita al texto. Sin embargo, se considera obsoleta en HTML5 y se recomienda usarla solo cuando se quiere dar énfasis visual al texto y no hay un elemento semántico más apropiado disponible.
- ****: Esta etiqueta se utiliza para denotar que el texto es importante o relevante dentro de un contexto específico. Además de aplicar negrita visualmente, también enfatiza el significado semántico del texto.
- **<i>**: Esta etiqueta se utiliza para aplicar cursiva al texto. Similar a ****, se considera obsoleta en HTML5 y se recomienda usarla solo cuando se quiere dar énfasis visual al texto y no hay un elemento semántico más apropiado disponible.
- ****: Esta etiqueta se utiliza para enfatizar el texto, generalmente se muestra en cursiva por defecto. A diferencia de **<i>**, **** enfatiza el significado semántico del texto, no solo su aspecto visual.
- **<mark>**: Esta etiqueta se utiliza para resaltar o marcar un fragmento de texto, generalmente se muestra con un color de fondo amarillo para hacerlo más visible.
- **<small>**: Esta etiqueta se utiliza para hacer que el texto sea más pequeño que el tamaño de fuente predeterminado.
- ****: Esta etiqueta se utiliza para indicar que el texto ha sido eliminado o marcado para su eliminación. Por lo general, se muestra con una línea a través del texto.
- **<ins>**: Esta etiqueta se utiliza para indicar que el texto ha sido insertado o añadido recientemente. Por lo general, se muestra subrayado.
- **<sub>**: Esta etiqueta se utiliza para representar texto en formato de subíndice, que se muestra ligeramente más abajo que el texto circundante.
- **<sup>**: Esta etiqueta se utiliza para representar texto en formato de superíndice, que se muestra ligeramente más arriba que el texto circundante.

HTML <blockquote>

La etiqueta **<blockquote>** en HTML se utiliza para representar citas o bloques de texto que se han tomado directamente de otra fuente, como un libro, un artículo o una entrevista. Esta etiqueta indica que el texto está siendo citado de otra fuente y a menudo se muestra con sangría en ambos lados para distinguirlo visualmente del resto del contenido.

Aquí tienes un ejemplo de cómo se utiliza la etiqueta **<blockquote>** en HTML:

<blockquote>

```
<p>La vida es lo que pasa mientras estás ocupado haciendo otros planes.</p>  
<footer>John Lennon</footer>
```

</blockquote>

En este ejemplo, el texto dentro de la etiqueta **<blockquote>** es una cita atribuida a John Lennon. La etiqueta **<footer>** se utiliza para proporcionar información adicional sobre la cita, como el autor.

La etiqueta **<blockquote>** puede contener cualquier tipo de contenido, como texto, imágenes, enlaces, etc. Se utiliza principalmente para citas largas que ocupan múltiples líneas, pero también puede usarse para citas cortas si se desea.

Es importante tener en cuenta que la etiqueta **<blockquote>** se utiliza para citar contenido de otras fuentes y no para resaltar o enfatizar texto. Para resaltar o enfatizar texto, se deben utilizar otras etiquetas como **** o ****, dependiendo del contexto semántico.

La etiqueta HTML **<q>** define una cita breve.

Los navegadores normalmente insertan comillas alrededor de la cita.

```
<p>El objetivo de WWF es: <q>Construir un futuro donde las  
personas vivan en armonía con la naturaleza.</q></p>
```

La etiqueta HTML **<abbr>** define una abreviatura o un acrónimo, como "HTML", "CSS", "Mr.", "Dr.", "ASAP", "ATM".

Marcar abreviaturas puede brindar información útil a navegadores, sistemas de traducción y motores de búsqueda.

Consejo: Utilice el atributo de título global para mostrar la descripción de la abreviatura/acrónimo cuando pase el mouse sobre el elemento.

```
<p>The <abbr title="World Health Organization">WHO</abbr> was  
founded in 1948.</p>
```

La etiqueta HTML **<address>** define la información de contacto del autor/propietario de un documento o artículo.

La información de contacto puede ser una dirección de correo electrónico, URL, dirección física, número de teléfono, identificador de redes sociales, etc.

El texto del elemento **<address>** normalmente se muestra en cursiva, y los navegadores siempre agregan un salto de línea antes y después del elemento **<address>**.

Ej:

```
<address>
Written by John Doe.<br>
Visit us at:<br>
Ejemplo.com<br>
Box 564, Disneyland<br>
USA
</address>
```

La etiqueta HTML **<cite>** define el título de una obra creativa (por ejemplo, un libro, un poema, una canción, una película, una pintura, una escultura, etc.) .

Nota: El nombre de una persona no es el título de una obra.

El texto del elemento **<cite>** normalmente se muestra en cursiva.

```
<p><cite>The Scream</cite> by Edvard Munch. Painted in 1893.</p>
```

Etiqueta de comentario HTML

La etiqueta de comentario en HTML se utiliza para insertar comentarios en el código fuente de una página web. Los comentarios son textos que **no se muestran** en la página web cuando se visualiza en un navegador web, pero que sirven para proporcionar información adicional o anotaciones dentro del código HTML.

En HTML, los comentarios se escriben entre **<!--** y **-->**. Todo el texto que se encuentra dentro de estos delimitadores se trata como un comentario y se ignora al renderizar la página en un navegador.

```
<!-- Este es un comentario -->
```

```
<p>Este es un párrafo.</p>
```

```
<!-- Recuerda agregar más información aquí →
```

Enlaces, tablas, listas, elementos de bloque, clases, identificador y multimedia.

HTML Enlaces

Recordemos que la etiqueta HTML **<a>** define un hipervínculo.

De forma predeterminada, la página vinculada se mostrará en la ventana actual del navegador. Para cambiar esto, debe especificar otro destino para el enlace.

El atributo **target** especifica dónde abrir el documento vinculado.

El atributo **target** puede tener uno de los siguientes valores:

- **_self** - Por defecto. Abre el documento en la misma pestaña en la que se hizo clic
- **_blank** - Abre el documento en una nueva ventana o pestaña
- **_parent** - Abre el documento en el marco principal
- **_top** - Abre el documento en el cuerpo completo de la ventana

```
<a href="https://www.w3schools.com/" target="_blank">¡Prueba!</a>
```

Utilice **mailto:** dentro del atributo **href** para crear un vínculo que abra el programa de correo electrónico del usuario (para permitirle enviar un nuevo correo electrónico)

```
<a href="mailto:someone@example.com">Enviar correo electrónico</a>
```

HTML Tablas

Las tablas en HTML son elementos fundamentales para organizar y presentar datos de manera estructurada y visualmente atractiva en una página web. Permiten mostrar información en filas y columnas, lo que facilita la comparación y la comprensión de los datos.

Para crear una tabla en HTML, utilizamos las etiquetas **<table>**, **<tr>**, **<td>** y **<th>**. Aquí tienes una explicación de cada una de estas etiquetas:

- **<table>**: Define una tabla en HTML y delimita su contenido. Todo el contenido de la tabla se coloca entre las etiquetas de inicio y cierre de **<table>**.
- **<tr>**: Define una fila en una tabla. Cada fila de la tabla se crea utilizando la etiqueta **<tr>**, y dentro de cada fila, se definen las celdas utilizando las etiquetas **<td>** o **<th>**.
- **<td>**: Define una celda de datos en una fila de la tabla. El contenido de la celda se coloca entre las etiquetas de inicio y cierre de **<td>**. Estas celdas se utilizan para mostrar datos regulares en la tabla.

- **<th>**: Define una celda de encabezado en una fila de la tabla. El contenido de la celda se coloca entre las etiquetas de inicio y cierre de **<th>**. Estas celdas se utilizan típicamente para encabezados de columna o fila, y su contenido se muestra en negrita y centrado por defecto.

Ejemplo:

```
<table>
  <tr>
    <th>Company</th>
    <th>Contact</th>
    <th>Country</th>
  </tr>
  <tr>
    <td>Alfreds Futterkiste</td>
    <td>Maria Anders</td>
    <td>Germany</td>
  </tr>
  <tr>
    <td>Centro comercial Moctezuma</td>
    <td>Francisco Chang</td>
    <td>Mexico</td>
  </tr>
</table>
```

Los atributos **colspan** y **rowspan** son herramientas en HTML que permiten fusionar celdas en una tabla para crear diseños más complejos y mejorar la presentación de los datos. Aquí tienes una explicación de cada uno:

NAME		

APRIL		

2025		
FIESTA		

- **colspan**: Este atributo se utiliza para fusionar celdas horizontalmente en una tabla. Permite que una celda se extienda a través de múltiples columnas. Cuando se aplica colspan a una celda, esta celda abarca el número especificado de columnas, combinando visualmente las celdas contiguas en una sola.
- **rowspan**: Este atributo se utiliza para fusionar celdas verticalmente en una tabla. Permite que una celda se extienda a través de múltiples filas. Cuando se aplica rowspan a una celda, esta celda abarca el número especificado de filas, combinando visualmente las celdas contiguas en una sola.

Aquí tienes un ejemplo de cómo se ve el código HTML para utilizar colspan y rowspan:

```

<table border="1">
  <tr>
    <td colspan="2">Encabezado fusionado</td>
    <td>Encabezado 3</td>
  </tr>
  <tr>
    <td rowspan="2">Celda 1</td>
    <td>Celda 2</td>
    <td>Celda 3</td>
  </tr>
  <tr>
    <td>Celda 4</td>
    <td>Celda 5</td>
  </tr>
</table>

```

En este ejemplo:

- La primera fila tiene una celda que abarca dos columnas (**colspan="2"**) y luego una celda regular.
- En la segunda fila, la primera celda abarca dos filas (**rowspan="2"**), lo que hace que se extienda verticalmente a través de dos filas. Las otras celdas en esta fila se colocan normalmente, pero la celda que abarca dos filas "Celda 1" ocupa el espacio de las dos filas.

HTML Listas

Las listas en HTML son elementos que permiten organizar y presentar información de manera estructurada y fácilmente comprensible. Hay tres tipos principales de listas en HTML: **listas desordenadas** (), **listas ordenadas** () y **listas de definición** (<dl>). Aquí tienes una explicación de cada uno:

Listas Desordenadas ():

Se utilizan para mostrar elementos en un orden no específico o sin importar el orden. Cada elemento de la lista se representa con un marcador, como un punto, un guión o un círculo. Se crean utilizando la etiqueta y cada elemento de la lista se define con la etiqueta .

```

<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>

```

Listas Ordenadas ():

Se utilizan para mostrar elementos en un orden específico o secuencia.

Cada elemento de la lista se representa con un número, en orden ascendente.

Se crean utilizando la etiqueta y cada elemento de la lista se define con la etiqueta .

```
<ol>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

Listas de Definición (<dl>):

Se utilizan para mostrar términos junto con sus definiciones.

Cada término se representa con la etiqueta <dt> (término de definición) y cada definición se representa con la etiqueta <dd> (definición de término).

Se crean utilizando la etiqueta <dl> y cada término y definición se define dentro de ella.

```
<dl>
  <dt>Coffee</dt>
  <dd>- black hot drink</dd>
  <dt>Milk</dt>
  <dd>- white cold drink</dd>
</dl>
```

Las listas en HTML son muy flexibles y permiten anidar listas dentro de otras listas para crear estructuras más complejas. Además, se pueden aplicar estilos CSS para personalizar la apariencia de las listas, como cambiar el tipo de marcador, el color o la sangría.

HTML Elemento div

El elemento <div> en HTML es un contenedor genérico que se utiliza para agrupar y organizar otros elementos en bloques separados en una página web. La etiqueta <div> no tiene un significado semántico específico por sí sola, pero se utiliza como un contenedor de propósito general para ayudar en la estructuración y estilización del contenido de una página.

Algunas características clave del elemento <div>:

- **Contenedor Genérico:** El elemento <div> se utiliza principalmente como un contenedor de propósito general para agrupar otros elementos y crear bloques de contenido en una página web.
- **Flexibilidad:** Dado que el elemento <div> no tiene un significado semántico específico, es muy flexible y se puede utilizar de diversas formas para organizar y estructurar el contenido de una página web.

- Estilización con CSS: Los elementos <div> se pueden estilizar fácilmente utilizando CSS para cambiar su apariencia, tamaño, posición y otros estilos visuales. Esto permite crear diseños complejos y personalizados en una página web.
- Anidamiento: Los elementos <div> se pueden anidar dentro de otros elementos <div>, lo que permite crear estructuras de contenido jerárquicas y más complejas en una página web.

Ej:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Ejemplo de Div</title>
  <style>
    /* Estilos CSS para el contenedor div */
    .contenedor {
      border: 1px solid #000;
      padding: 20px;
      background-color: #f0f0f0;
    }
  </style>
</head>
<body>
  <!-- Contenedor div -->
  <div class="contenedor">
    <h1>Título de la Página</h1>
    <p>Este es un párrafo de ejemplo dentro del contenedor div.</p>
    <ul>
      <li>Elemento de lista 1</li>
      <li>Elemento de lista 2</li>
      <li>Elemento de lista 3</li>
    </ul>
  </div>
</body>
</html>
```

- Luego veremos que significa el bloque style.

HTML Atributo de clase

El atributo de clase en HTML se utiliza para asignar una o más clases a un elemento HTML. Las clases son nombres o identificadores que se utilizan para aplicar estilos CSS y para seleccionar elementos específicos en un documento HTML utilizando JavaScript u otras técnicas de manipulación del DOM.

Algunas características clave del atributo de clase en HTML:

- **Asignación de Estilos:** El atributo de clase se utiliza principalmente para aplicar estilos CSS a uno o más elementos HTML. Al asignar una clase a un elemento, se pueden aplicar reglas de estilo específicas definidas en la hoja de estilos CSS correspondiente.
- **Reutilización de Estilos:** Las clases permiten reutilizar estilos CSS en múltiples elementos HTML en una página web. Esto facilita la aplicación de un diseño consistente y coherente en todo el sitio web.
- **Separación de Responsabilidades:** El uso de clases separa la estructura del documento HTML de su presentación visual. Esto promueve una mejor práctica de desarrollo web conocida como "separación de preocupaciones", donde el HTML se utiliza para estructurar el contenido y el CSS se utiliza para definir su apariencia.
- **Selección y Manipulación con JavaScript:** Las clases también se pueden utilizar para seleccionar elementos específicos en el DOM utilizando JavaScript. Esto facilita la manipulación dinámica de la página web, como agregar, quitar o modificar clases en respuesta a eventos o acciones del usuario.

Ejemplo:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Ejemplo de Clase en HTML</title>
  <style>
    /* Estilos CSS para la clase "destacado" */
    .destacado {
      font-weight: bold;
      color: red;
    }
  </style>
</head>
<body>
  <!-- Utilización del atributo de clase -->
  <p class="destacado">Este párrafo está destacado.</p>
```



```
<p>Este es un párrafo normal.</p>
</body>
</html>
```

En este ejemplo, se define una clase llamada "destacado" en la hoja de estilos CSS, que aplica estilos de negrita y color rojo al texto. Esta clase se asigna al primer párrafo utilizando el atributo de clase **<p class="destacado">**. Como resultado, el texto dentro del primer párrafo se mostrará en negrita y en color rojo.

HTML Atributo de identificación

El atributo de identificación en HTML se utiliza para proporcionar un identificador único a un elemento HTML dentro de un documento. Este identificador, conocido como ID, permite seleccionar y manipular el elemento de forma única mediante CSS y JavaScript.

Algunas características clave del atributo de identificación en HTML:

- **Unicidad:** Cada elemento en un documento HTML puede tener un ID único. Este ID debe ser único dentro del documento para garantizar que no haya conflictos al seleccionar el elemento utilizando CSS o JavaScript.
- **Selección de Estilos con CSS:** Los ID se utilizan principalmente para aplicar estilos CSS específicos a un elemento en particular. Puedes definir reglas de estilo en tu hoja de estilos CSS y luego aplicar esas reglas a un elemento utilizando su ID como selector.
- **Manipulación con JavaScript:** Los ID también se utilizan para seleccionar y manipular elementos HTML utilizando JavaScript. Puedes utilizar métodos como `getElementById()` para obtener una referencia al elemento y luego realizar acciones como cambiar su contenido, estilo, atributos, etc.
- **Anclas y Enlaces Internos:** Los ID se utilizan frecuentemente en combinación con anclas (`<a>`) para crear enlaces internos dentro de una página web. Al proporcionar un ID único a un elemento, puedes enlazar directamente a ese elemento desde otra parte de la página o desde otra página web.

Ejemplo:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Ejemplo de ID en HTML</title>
  <style>
    /* Estilos CSS para el elemento con ID "titulo" */
    #titulo {
      font-size: 24px;
```

```
        color: blue;
    }
</style>
</head>
<body>
    <!-- Utilización del atributo de identificación -->
    <h1 id="titulo">Este es un título con ID</h1>
    <p>Este es un párrafo normal.</p>
</body>
</html>
```

En este ejemplo, se define un elemento `<h1>` con el ID **"titulo"**. Se define un estilo CSS específico para el ID "titulo", que cambia el tamaño de la fuente a 24px y el color a azul. Este estilo se aplica solo al elemento con el ID "titulo", lo que le da una apariencia diferente al resto de los elementos en la página.

Marcadores HTML con ID y enlaces

Los marcadores HTML se utilizan para permitir a los lectores saltar a partes específicas de una página web.

Los marcadores pueden resultar útiles si su página es muy larga.

Para utilizar un marcador, primero debe crearlo y luego agregarle un enlace.

Luego, cuando se hace clic en el enlace, la página se desplazará hasta la ubicación con el marcador.

Ejemplo:

Primero, cree un marcador con el atributo id:

```
<h2 id="C4">Chapter 4</h2>
```

Luego, agregue un enlace al marcador ("Saltar al Capítulo 4"), desde la misma página:

```
<a href="#C4">Saltar al Capítulo 4</a>
```

O agregue un enlace al marcador ("Saltar al Capítulo 4"), desde otra página:

```
<a href="html_demo.html#C4">Saltar al Capítulo 4</a>
```

HTML Iframe

El elemento **<iframe>** en HTML se utiliza para incrustar otro documento HTML dentro de una página web. Este elemento permite mostrar contenido de otra fuente, como una página web externa, un documento PDF, un video de YouTube, etc., dentro de un marco dentro de la página actual.

Algunas características clave del elemento **<iframe>** en HTML:

- **Incorporación de Contenido Externo:** El elemento **<iframe>** permite incrustar contenido de otras fuentes dentro de una página web. Esto puede incluir páginas web externas, documentos PDF, videos, mapas, widgets y más.
- **Separación de Contenido:** Al utilizar **<iframe>**, puedes mostrar contenido de otras fuentes mientras mantienes la separación entre ese contenido y el resto de tu página web. Esto es útil para integrar funcionalidades externas sin comprometer la integridad de tu propio sitio.
- **Personalización del Tamaño y Aspecto:** Puedes definir el tamaño y aspecto del **<iframe>** utilizando atributos como **width** y **height**, lo que te permite controlar cómo se muestra el contenido incrustado dentro de tu página.
- **Navegación Segura:** El contenido dentro de un **<iframe>** se carga en un marco separado, lo que proporciona una capa adicional de seguridad. Esto significa que el contenido incrustado no tiene acceso directo al DOM de la página principal y viceversa, lo que ayuda a prevenir ataques de scripting cruzado (XSS).

Ejemplo:

```
<iframe src="demo_iframe.htm" style="height:200px;width:300px;"  
title="Iframe Ejemplo"></iframe>
```

HTML Rutas de archivos

Una ruta de archivo describe la ubicación de un archivo en la estructura de carpetas de un sitio web.

Ejemplos de ruta de archivo

Camino	Descripción
<code></code>	El archivo "picture.jpg" se encuentra en la misma carpeta que la página actual.
<code></code>	El archivo "picture.jpg" se encuentra en la carpeta de imágenes de la carpeta actual
<code></code>	El archivo "picture.jpg" se encuentra en la carpeta de imágenes en la raíz de la página web actual
<code></code>	El archivo "picture.jpg" se encuentra en la carpeta un nivel arriba de la carpeta actual

HTML Entidades

Las entidades HTML son secuencias de caracteres predefinidas que representan caracteres especiales o reservados en HTML. Estas entidades se utilizan para asegurar que los navegadores web interpreten correctamente los caracteres y no los confundan con marcadores HTML o códigos de formato. Aquí tienes algunos ejemplos de entidades HTML comunes:

Entidades de Caracteres Especiales:

<; Representa el símbolo "<" (menor que).
>; Representa el símbolo ">" (mayor que).
&; Representa el símbolo "&" (y comercial).
"; Representa el símbolo "\"" (comillas dobles).
' o '; Representa el símbolo "'" (comillas simples).

Entidades de Caracteres Acentuados:

Á; Á
É; É
Í; Í
Ó; Ó
Ú; Ú
á; á
é; é
í; í
ó; ó
ú; ú

Entidades de Puntuación y Símbolos:

 ; Representa un espacio en blanco no rompible.
—; Representa un guión largo — (em dash).
–; Representa un guión corto – (en dash).
©; Representa el símbolo de derechos de autor ©.
®; Representa el símbolo de marca registrada ®.
™; Representa el símbolo de marca comercial ™.

Estas entidades se utilizan dentro del código HTML, reemplazando el carácter especial o reservado que representan. Por ejemplo, si deseas mostrar el símbolo "<" en tu página web sin que sea interpretado como una etiqueta HTML, debes escribir <. De esta manera, el navegador mostrará "<" en lugar de interpretarlo como una etiqueta HTML de inicio.

Muchos símbolos matemáticos, técnicos y monetarios no están presentes en un teclado normal.

Para agregar dichos símbolos a una página HTML, puede usar el nombre de la entidad o el número de la entidad (una referencia decimal o hexadecimal) para el símbolo.










Algunos símbolos matemáticos soportados por HTML

Carácter	Número	Entidad	Descripción
∀	∀	∀	FOR ALL
∂	∂	∂	PARTIAL DIFFERENTIAL
∃	∃	∃	THERE EXISTS
∅	∅	∅	EMPTY SETS
∇	∇	∇	NABLA
∈	∈	∈	ELEMENT OF
∉	∉	∉	NOT AN ELEMENT OF
∋	∋	∋	CONTAINS AS MEMBER
∏	∏	∏	N-ARY PRODUCT
Σ	∑	∑	N-ARY SUMMATION

Los emojis parecen imágenes o iconos, pero no lo son.

Son letras (caracteres) del juego de caracteres UTF-8 (Unicode).

Algunos símbolos emoji en UTF-8

Emoji	Valor
	🗻
	🗼
	🗽
	🗾
	🗿
	😀
	😁
	😂
	😃
	😄
	😅

HTML Multimedia

Se refiere a la capacidad de incorporar diferentes tipos de contenido multimedia, como imágenes, audio y video, dentro de una página web. Estos elementos multimedia permiten enriquecer la experiencia del usuario al proporcionar contenido visual y auditivo interactivo. Aquí tienes una descripción de los principales elementos multimedia en HTML:

Imágenes: Las imágenes son elementos visuales que se pueden incrustar en una página web para mostrar gráficos, fotografías, ilustraciones, iconos y más. En HTML, las imágenes se insertan utilizando el elemento ``, que incluye el atributo `src` para especificar la URL de la imagen.

```

```

Audio: El elemento `<audio>` se utiliza para incrustar archivos de audio en una página web. Permite a los usuarios reproducir pistas de audio directamente en el navegador. Se pueden proporcionar múltiples fuentes de audio para garantizar la compatibilidad con diferentes navegadores.

```
<audio controls>
  <source src="audio.mp3" type="audio/mpeg">
  Tu navegador no soporta el elemento de audio.
</audio>
```

Video: El elemento `<video>` se utiliza para incrustar archivos de video en una página web. Permite a los usuarios reproducir videos directamente en el navegador. Al igual que con el audio, se pueden proporcionar múltiples fuentes de vídeo para garantizar la compatibilidad con diferentes navegadores.

```
<video controls width="400" height="300">
  <source src="video.mp4" type="video/mp4">
  Tu navegador no soporta el elemento de video.
</video>
```

En el caso de Videos YouTube, la mejor forma de hacerlo es usando un `iframe`:

```
<iframe width="420" height="315"
src="https://www.youtube.com/embed/tgbNymZ7vqY">
</iframe>
```

- **Autoplay y Muted:**
`src="https://www.youtube.com/embed/tgbNymZ7vqY?autoplay=1&muted=1">`
- **Loop:**
`src="https://www.youtube.com/embed/tgbNymZ7vqY?playlist=tgbNymZ7vqY&loop=1">`

- Controles:

`src="https://www.youtube.com/embed/tgbNymZ7vqY?controls=0">`

Formularios y controles de entrada

Los formularios HTML son elementos para la interacción del usuario en la web, ya que permiten recopilar datos y enviarlos al servidor para su procesamiento. Un formulario puede contener diversos elementos, como campos de texto, botones de opción, casillas de verificación, botones de envío y más.

Elemento **<form>**: El elemento `<form>` es la envoltura principal de un formulario HTML. Define el inicio y el final del formulario y proporciona atributos para controlar cómo se envían y procesan los datos del formulario.

```
<form>
.
form elements
.
</form>
```

El atributo `action` en los formularios HTML especifica la URL a la cual se enviarán los datos del formulario cuando este sea enviado. Indica el script o punto final del lado del servidor que procesa los datos introducidos por el usuario.

```
<form action="procesar.php" method="post">
  <!-- Aquí van los elementos del formulario -->
</form>
```

En este ejemplo, `procesar.php` es la URL a la que se enviarán los datos del formulario cuando este sea enviado. Esta URL podría apuntar a un script del lado del servidor escrito en PHP, Python, Node.js o cualquier otro lenguaje del lado del servidor. El atributo `method` especifica el método HTTP que se utilizará al enviar los datos del formulario. En este caso, se utiliza **post**, que envía los datos del formulario como una solicitud **POST HTTP**. Alternativamente, se puede usar **get como método**, lo que envía los datos del formulario como una cadena de consulta agregada a la URL.

Notas sobre GET:

- Añade los datos del formulario a la URL, en pares de nombre/valor.
- ¡NUNCA uses GET para enviar datos sensibles! (los datos del formulario enviados son visibles en la URL).
- La longitud de una URL está limitada (2048 caracteres).
- Útil para envíos de formularios donde un usuario quiere marcar el resultado.
- GET es útil para datos no seguros, como cadenas de consulta en Google.

Notas sobre POST:

- Añade los datos del formulario dentro del cuerpo de la solicitud HTTP (los datos del formulario enviados no se muestran en la URL).
- POST no tiene limitaciones de tamaño y puede ser utilizado para enviar grandes cantidades de datos.
- Los envíos de formularios con POST no pueden ser marcados.

Autocomplete Attribute

El atributo autocomplete es un atributo de los elementos de formulario HTML que controla si el navegador debe completar automáticamente los campos del formulario con información previamente ingresada por el usuario. Este atributo puede tener los siguientes valores:

- on: Indica que el navegador puede completar automáticamente los campos del formulario.
- off: Indica que el navegador no debe completar automáticamente los campos del formulario.

```
<form action="/action_page.php" autocomplete="on">
```

El uso del atributo autocomplete puede mejorar la experiencia del usuario al ahorrarle tiempo al rellenar formularios, especialmente en sitios web que recopilan información similar en múltiples ocasiones, como nombres de usuario y contraseñas. Sin embargo, en formularios que contienen información sensible, como datos financieros o de tarjetas de crédito, es común desactivar la función de autocompletar para mejorar la seguridad y privacidad del usuario.

Novalidate Attribute

```
<form action="/action_page.php" novalidate>
```

El atributo novalidate es un atributo booleano que se puede agregar a un elemento <form> en HTML para indicar al navegador que no valide los campos del formulario antes de enviar los datos al servidor. Cuando se utiliza este atributo, el navegador no ejecutará las validaciones predeterminadas que normalmente realiza, como verificar si se han completado los campos requeridos o si los datos ingresados cumplen con ciertos criterios de formato.

Este atributo es útil en situaciones donde se necesita realizar la validación de forma personalizada mediante JavaScript o cuando se está realizando la validación en el lado del servidor y no se requiere una validación adicional en el navegador. Al agregar el atributo novalidate, se evita que el navegador muestre mensajes de error automáticos y se permite un mayor control sobre el proceso de validación del formulario.

Elementos de Entrada: Los elementos de entrada son los campos donde los usuarios pueden introducir datos. Algunos de los elementos de entrada más comunes son:

Algunos ejemplos de elementos de entrada:

<input type="button">: Un botón sin ninguna acción predeterminada.

`<input type="button" value="Click Me">`

<input type="checkbox">: Una casilla de verificación que permite al usuario seleccionar múltiples opciones.

`<input type="checkbox" id="opcion1" name="opcion1" value="opcion1">`

`<label for="opcion1">Opción 1</label>`

<input type="color">: Un control para seleccionar un color.

`<input type="color" value="#ff0000">`

<input type="date">: Un campo para introducir una fecha.

`<input type="date" value="2022-12-31">`

<input type="datetime-local">: Un campo para introducir una fecha y hora local.

`<input type="datetime-local" value="2022-12-31T12:00">`

<input type="email">: Un campo para introducir una dirección de correo electrónico.

`<input type="email" placeholder="correo@example.com">`

<input type="file">: Un control para seleccionar archivos del dispositivo del usuario.

`<input type="file" accept=".jpg, .png">`

<input type="hidden">: Un campo oculto que no es visible para el usuario pero que se envía con el formulario.

`<input type="hidden" name="id" value="123">`

<input type="image">: Una imagen que actúa como un botón de envío de formulario.

`<input type="image" src="boton.png" alt="Enviar">`

<input type="month">: Un campo para introducir un mes y un año.

`<input type="month" value="2022-12">`

HTML Input Attributes

Los atributos de entrada en HTML son propiedades adicionales que se pueden agregar a los elementos `<input>` para modificar su comportamiento o apariencia. Aquí tienes una descripción de algunos de los atributos más comunes que se pueden usar con los elementos `<input>`:

- **type:** Especifica el tipo de entrada. Los valores posibles incluyen text, password, checkbox, radio, submit, button, file, email, date, number, color, entre otros.

- **value:** Define el valor predeterminado del campo de entrada. Para campos de tipo checkbox o radio, este atributo define el valor que se enviará al servidor si el campo está marcado.
- **placeholder:** Muestra un texto de marcador de posición dentro del campo de entrada que proporciona una pista sobre el tipo de información que se espera.
- **name:** Especifica el nombre del campo de entrada, que se utiliza para identificar el valor del campo cuando se envía el formulario.
- **id:** Proporciona un identificador único para el campo de entrada, que se puede utilizar para vincularlo con etiquetas de <label> o para manipularlo mediante JavaScript.
- **required:** Indica que el campo de entrada es obligatorio y no se puede enviar el formulario sin completarlo.
- **disabled:** Deshabilita el campo de entrada, impidiendo que el usuario interactúe con él o envíe su valor con el formulario.
- **readonly:** Hace que el campo de entrada sea de solo lectura, lo que significa que el usuario puede ver su valor pero no puede editarlo.
- **autocomplete:** Controla si el navegador debe completar automáticamente el campo de entrada basándose en entradas anteriores del usuario.
- **autofocus:** Hace que el campo de entrada reciba automáticamente el foco cuando la página se carga, lo que permite al usuario comenzar a escribir inmediatamente.
- Estos son solo algunos de los atributos disponibles para los elementos <input>. Combinar diferentes atributos puede proporcionar una amplia gama de funcionalidades y personalizaciones para los formularios HTML en tus páginas web.

Ejemplo de un formulario:

```
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Ejemplo de Formulario HTML</title>
</head>
<body>

<h2>Formulario de Registro</h2>
```

```
<form action="/procesar.php" method="post">
  <div>
    <label for="nombre">Nombre:</label>
    <input type="text" id="nombre" name="nombre" placeholder="Ingrese
su nombre" required autofocus>
  </div>

  <div>
    <label for="email">Email:</label>
    <input type="email" id="email" name="email"
placeholder="correo@example.com" required>
  </div>

  <div>
    <label for="password">Contraseña:</label>
    <input type="password" id="password" name="password" required>
  </div>

  <div>
    <label for="archivo">Archivo:</label>
    <input type="file" id="archivo" name="archivo" accept=".jpg, .png"
required>
  </div>

  <div>
    <label for="fecha">Fecha de Nacimiento:</label>
    <input type="date" id="fecha" name="fecha">
  </div>

  <div>
    <label for="color">Color Favorito:</label>
    <input type="color" id="color" name="color" value="#ff0000">
  </div>

  <div>
    <label for="telefono">Teléfono:</label>
    <input type="tel" id="telefono" name="telefono"
pattern="[0-9]{3}-[0-9]{3}-[0-9]{4}" title="Formato: 123-456-7890">
  </div>

  <div>
    <label for="submit"></label>
    <input type="submit" id="submit" value="Enviar">
    <input type="reset" value="Limpiar">
  </div>
</form>

</body>
</html>
```

HTML Pattern Attribute

El atributo `pattern` es utilizado en elementos de entrada HTML para especificar un patrón que debe cumplir el valor introducido por el usuario. Este patrón se define utilizando una expresión regular, y el navegador verificará que el valor ingresado coincida con el patrón antes de permitir que se envíe el formulario.

- **Expresión Regular:** El patrón especificado en el atributo `pattern` debe ser una expresión regular válida. Esta expresión regular define el formato que deben tener los datos ingresados por el usuario.
- **Validación del lado del cliente:** La validación del patrón se realiza en el lado del cliente, es decir, en el navegador del usuario. Esto significa que el navegador verificará el valor del campo de entrada tan pronto como el usuario intente enviar el formulario.
- **Mensajes de Validación Personalizados:** Puedes utilizar el atributo `title` junto con `pattern` para proporcionar un mensaje de validación personalizado que se mostrará al usuario si el valor introducido no coincide con el patrón especificado.
- **Uso Común:** El atributo `pattern` es comúnmente utilizado para validar campos como números de teléfono, códigos postales, direcciones de correo electrónico, números de tarjeta de crédito, entre otros.

Ejemplo:

```
<input type="tel" id="telefono" name="telefono" pattern="[0-9]{3}-[0-9]{3}-[0-9]{4}" title="Formato: 123-456-7890">
```

En este caso, el patrón `[0-9]{3}-[0-9]{3}-[0-9]{4}` especifica que el campo debe contener un número de teléfono en el formato XXX-XXX-XXXX, donde X representa un dígito del 0 al 9. Si el usuario intenta enviar el formulario con un número de teléfono que no cumple con este formato, se mostrará el mensaje de validación "Formato: 123-456-7890".