



El elemento **<code>** también trabaja junto con otros elementos para presentar contenido, pero tiene una relación particular con el elemento **<pre>**. Estos elementos se implementan juntos para presentar código de programación. El elemento **<code>** indica que el contenido es código de programación y el elemento **<pre>** formatea ese contenido para mostrarlo en pantalla como se ha declarado originalmente en el documento (respetando los espacios y los saltos de línea).

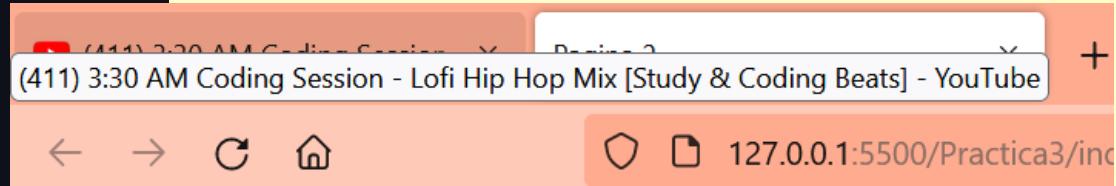
**<article>** muestra código programado en JavaScript. Debido a que este código no se encuentra dentro de un elemento **<script>**, el navegador no lo ejecuta y, debido a que se incluye dentro de un elemento **<pre>**, se presenta con los saltos de línea y los espacios declarados en el documento.

# Desarrollo Web



GOBIERNO DE LA  
CIUDAD DE MÉXICO

```
9 <body>
10 <article>
11   <pre>
12     <code>
13       function cambiarColor() {
14         document.body.style.backgroundColor = "#0000FF";
15       }
16       document.addEventListener("click", cambiarColor);
17     </code>
18   </pre>
19 </article>
20 <footer>
21   <address>Ciudad de México, México</address>
22   <small>© Derechos Reservados 2022</small>
23 </footer>
24 </body>
```



```
function cambiarColor() {
  document.body.style.backgroundColor = "#0000FF";
}
document.addEventListener("click", cambiarColor);
```

Ciudad de México, México  
© Derechos Reservados 2022

95



Para agregar un comentario, tenemos que escribir el texto entre las etiquetas `<!-- y -->`, como en `<!-- Este es un comentario -->`.

## Enlaces

Conectar documentos con otros documentos mediante enlaces es lo que hace posible la Web. Un enlace es contenido asociado a una URL que indica la ubicación de un recurso. Cuando el usuario hace clic en el contenido (texto o imagen), el navegador descarga el recurso. HTML incluye el siguiente elemento para crear enlaces.



**< a >**—Este elemento crea un enlace. El texto o la imagen que representa el enlace se incluye entre las etiquetas de apertura y cierre. El elemento incluye el atributo **href** para especificar la URL del enlace.

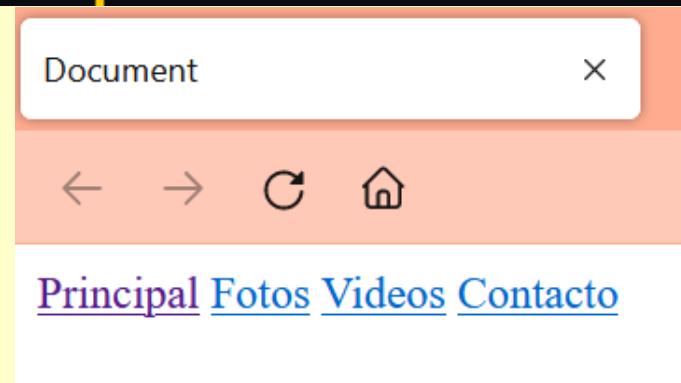
Los enlaces se pueden crear para conectar el documento actual con otros documentos en el mismo sitio web o en otros sitios. Por ejemplo, podemos enlazar las opciones en el menú de nuestra página web a otros documentos en nuestro servidor.

# Desarrollo Web



GOBIERNO DE LA  
CIUDAD DE MÉXICO

```
9   <body>
10     <nav>
11       <a href="index.html">Principal</a>
12       <a href="fotos.html">Fotos</a>
13       <a href="videos.html">Videos</a>
14       <a href="contacto.html">Contacto</a>
15     </nav>
16   </body>
```





El elemento **<nav>** incluye cuatro elementos **<a>** por cada opción del menú. Los elementos incluyen el atributo **href** para indicar al navegador el documento que tiene que abrir cuando el usuario hace clic en el enlace. Por defecto, los enlaces se muestran subrayados y en color azul (o violeta si el usuario ya ha hecho clic en ellos).

Cuando el usuario hace clic en cualquiera de estos enlaces, el navegador descarga el documento indicado por el atributo href y muestra su contenido en pantalla.



Los documentos enlazados en el menú del ejemplo anterior pertenecen al mismo sitio web y esa es la razón por la que usamos URL relativas para especificar su ubicación, pero si lo que necesitamos es crear enlaces a documentos que no están almacenados en nuestro servidor, tenemos que usar URL absolutas, como en el siguiente ejemplo.

El código en el ejemplo siguiente agrega un enlace al pie de página que apunta al sitio web <https://pilares.cdmx.gob.mx/inicio>. El enlace trabaja como cualquier otro, pero ahora el navegador tiene la URL completa para acceder al documento (en este caso, el archivo index del sitio web con el dominio <https://pilares.cdmx.gob.mx/inicio>). Practica4

# Desarrollo Web



GOBIERNO DE LA  
CIUDAD DE MÉXICO

```
<body>
    <nav>
        <a href="index.html">Principal</a>
        <a href="fotos.html">Fotos</a>
        <a href="videos.html">Videos</a>
        <a href="contacto.html">Contacto</a>

        <p>Esta es la pagina Principal</p>
    </nav>

    <footer>
        <address>Ciudad de México</address>
        <small>&copy; 2022 <a href="https://pilares.cdmx.gob.mx/inicio"
            target="_blank">Pilares CDMX</a></small>
    </footer>
</body>
```

# Desarrollo Web



GOBIERNO DE LA  
CIUDAD DE MÉXICO



127.0.0.1:5500/Practica4/

[Principal](#) [Fotos](#) [Videos](#) [Contacto](#)

Esta es la pagina Principal

*Ciudad de México*

© 2022 [Pilares CDMX](#)

https://pilares.cdmx.gob.mx/inicio

PILARES EN LÍNEA ENCUENTRA TU PILARES COMUNIDAD CONVOCATORIAS

ESCUELA



El elemento `<a>` puede incluir el atributo **target** para especificar el destino en el cual el documento será abierto. El valor `_self` se asigna por defecto, lo que significa que el documento se abre en la misma ubicación que el documento actual (el mismo recuadro o ventana). Otros valores son `_blank` (el documento se abre en una nueva ventana), `_parent` (el documento se abre en el recuadro padre), y `_top` (el documento se abre en la ventana actual).

El siguiente ejemplo implementa el último valor para acceder al dominio <https://pilares.cdmx.gob.mx/inicio> desde una nueva ventana, de modo que el usuario nunca abandona nuestro sitio web.

# Desarrollo Web



GOBIERNO DE LA  
CIUDAD DE MÉXICO

```
<body>
  <nav>
    <a href="index.html">Principal</a>
    <a href="fotos.html">Fotos</a>
    <a href="videos.html">Videos</a>
    <a href="contacto.html">Contacto</a>

    <p>Esta es la pagina Principal</p>
  </nav>

  <footer>
    <address>Ciudad de México</address>
    <small>&copy; 2022 <a href="https://pilares.cdmx.gob.mx/inicio"
      target="_blank">Pilares CDMX</a></small>
  </footer>
</body>
```



Además de conectar un documento con otro, los enlaces también se pueden crear hacia otros elementos dentro del mismo documento. Esto es particularmente útil cuando el documento genera una página extensa que el usuario debe desplazar para poder ver todo su contenido. Aprovechando esta característica, podemos crear enlaces hacia diferentes partes de una página. Cuando el usuario quiere ver algo que no es visible al momento, puede hacer clic en estos enlaces y el navegador desplaza la página hasta que el elemento apuntado por el enlace aparece en la pantalla. El elemento que queremos enlazar tiene que ser identificado con el atributo **id**. Para crear un enlace a un elemento, debemos incluir el valor asignado a este atributo precedido por el carácter #, igual que ilustra el siguiente ejemplo.



## VER PRACTICA 5

The screenshot shows a web browser window with the following details:

- Address Bar:** 127.0.0.1:5500/Practica5/index.html#titulo
- Toolbar:** Includes back, forward, search, and refresh buttons.
- Content Area:**
  - ### Blog
  - [Post 1](#)
  - [Post 2](#)
  - [Post 3](#)
  - [Post 4](#)
  - [Post 5](#)
  - ### Post 1

Lore ipsum dolor sit amet consectetur adipisicing elit. Beatae facilis numquam ullam maiores quaerat, reprehenderit eos aspernatur voluptas magnam cupiditate! Est placeat quia ipsa quo, tempore voluptas ipsum debitis aliquam.

Lore ipsum dolor sit amet consectetur adipisicing elit. Beatae facilis numquam ullam maiores quaerat, reprehenderit eos aspernatur voluptas magnam cupiditate! Est placeat quia ipsa quo, tempore voluptas ipsum debitis aliquam.

Lore ipsum dolor sit amet consectetur adipisicing elit. Beatae facilis numquam ullam maiores quaerat, reprehenderit eos aspernatur voluptas magnam cupiditate! Est placeat quia ipsa quo, tempore voluptas ipsum debitis aliquam.

Lore ipsum dolor sit amet consectetur adipisicing elit. Beatae facilis numquam ullam maiores quaerat, reprehenderit eos aspernatur voluptas magnam cupiditate! Est placeat quia ipsa quo, tempore voluptas ipsum debitis aliquam.

Lore ipsum dolor sit amet consectetur adipisicing elit. Beatae facilis numquam ullam maiores quaerat, reprehenderit eos aspernatur voluptas magnam cupiditate! Est placeat quia ipsa quo, tempore voluptas ipsum debitis aliquam.

Lore ipsum dolor sit amet consectetur adipisicing elit. Beatae facilis numquam ullam maiores quaerat, reprehenderit eos aspernatur voluptas magnam cupiditate! Est placeat quia ipsa quo, tempore voluptas ipsum debitis aliquam.

Lore ipsum dolor sit amet consectetur adipisicing elit. Beatae facilis numquam ullam maiores quaerat, reprehenderit eos aspernatur voluptas magnam cupiditate! Est placeat quia ipsa quo, tempore voluptas ipsum debitis aliquam.



En el ejemplo anterior, usamos el atributo **id** con el valor "titulo" para identificar el elemento **<header>**. Usando este valor y el carácter #, creamos un enlace al final del contenido que lleva al usuario hacia la parte superior de la página. Cuando el usuario hace clic en el enlace, en lugar de abrir un documento, el navegador desplaza la página hasta que el contenido del elemento **<header>** se vuelve visible.



El elemento `<a>` también se puede usar para crear enlaces a aplicaciones. HTML ofrece las palabras clave **mailto** y **tel** para especificar una cuenta de correo o un número de teléfono. Cuando se hace clic en un enlace de estas características, el sistema abre el programa encargado de responder a este tipo de solicitudes (enviar un correo o hacer una llamada telefónica) y le envía los datos especificados en el enlace. El siguiente ejemplo implementa la palabra clave **mailto** para enviar un correo electrónico.

# Desarrollo Web



GOBIERNO DE LA  
CIUDAD DE MÉXICO

Formato Insertar Dibujar ▾

N K S ▾ | Título 1

De: gaussluis@hotmail.com

Para: [gaussluiskof@gmail.com](mailto:gaussluiskof@gmail.com) |

Asunto

Enviado desde [Correo](#) para Windows

## PRACTICA 5

```
<footer>
    <address>Ciudad de México</address>
    <small>&copy; 2022 <a href="mailto:gaussluiskof@gmail.com">email</a></small>
</footer>
```



Los documentos a los que se accede a través de un enlace creado por el elemento `<a>` son descargados por el navegador y mostrados en pantalla, pero a veces los usuarios no necesitan que el navegador abra el documento, sino que el archivo se almacene en sus discos duros para usarlo más adelante. HTML ofrece dos atributos para este propósito:

**download**—Este es un atributo booleano que, cuando se incluye, indica que en lugar de leer el archivo el navegador debería descargarlo.

**ping**—Este atributo declara la ruta del archivo que se debe abrir en el servidor cuando el usuario hace clic en el enlace. El valor puede ser una o más URL separadas por un espacio.



Cuando el atributo **download** se encuentra presente dentro de un elemento **<a>**, el archivo especificado por el atributo **href** se descarga y almacena en el disco duro del usuario. Por otro lado, el archivo que indica el atributo **ping** no se descarga, sino que se ejecuta en el servidor. Este archivo se puede usar para ejecutar código que almacena información en el servidor cada vez que el archivo principal se descarga o para llevar un control de las veces que esta acción ocurre.

En el siguiente ejemplo, implementamos ambos atributos para permitir al usuario descargar un archivo PDF.



```
<body>
  <article>
    <p>La muerte es una quimera: porque mientras yo existo, no existe la muerte;<br>y cuando existe la muerte, ya no existo yo.<br>Epicuro de Samos</p>
    <footer>
      <a href="http://www.formasterminds.com/content/miarchivo.pdf" ping="http://www.formasterminds.com/control.php" download>Clic aquí para descargar</a>
    </footer>
  </article>
</body>
```

# Desarrollo Web



GOBIERNO DE LA  
CIUDAD DE MÉXICO

A screenshot of a web browser window. The tabs include: 'Recibidos (52) - gauss' (highlighted in orange), 'Correo: José Luis Mald' (blue icon), 'Correo: Jose Luis Lopez' (blue icon), '(486) 3:30 AM Coding' (YouTube icon, playing), 'Página 6' (white tab), and 'usuario.pdf'. The address bar shows '127.0.0.1:5500/Practica6/index.html'. The navigation bar includes icons for back, forward, refresh, and home.

La muerte es una quimera: porque mientras yo existo, no existe la muerte;  
y cuando existe la muerte, ya no existo yo.  
Epicuro de Samos

[Clic aquí para descargar](#)

A screenshot of a PDF viewer window. The title bar shows 'file:///C:/Users/gauss/Downloads/usuario.pdf'. The page content is as follows:

**¡AVISO IMPORTANTE!**  
Estimados Usuarios:  
Los días Martes, Miércoles, Jueves y Viernes.  
En el horario de 2:00 a 4:00 pm. Esta área está  
asignada exclusivamente para los usuarios del  
curso “Escuela de Código”.

¡Gracias por su compresión!

114



En el ejemplo anterior, agregamos un pie de página al artículo con un enlace a un archivo PDF. En circunstancias normales, un navegador moderno mostraría el contenido del archivo en pantalla, pero en este caso el atributo **download** obliga al navegador a descargar el archivo y almacenarlo en el disco duro. Este ejemplo incluye un atributo **ping** que apunta a un archivo llamado control.php. Como resultado, cada vez que el usuario hace clic en el enlace, se descarga el archivo PDF y el código PHP se ejecuta en el servidor, lo que permite al desarrollador hacer un seguimiento de las veces que esta acción ocurre (almacenando información acerca del usuario en una base de datos, por ejemplo).



## Imágenes

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

**<img>**—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, videos, etc.

**<figcaption>**—Este elemento introduce un título para el elemento **<figure>**.



Para incluir una imagen en el documento, solo necesitamos declarar el elemento **<img>** y asignar la URL del archivo al atributo **src**.

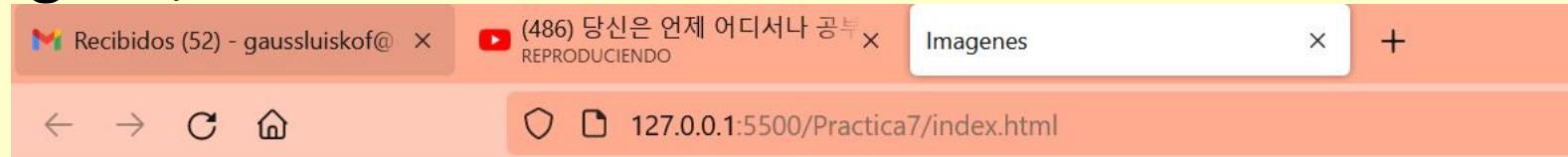
```
<body>
  <article>
    <p>
      Lorem ipsum dolor sit amet consectetur adipisicing elit.
      Soluta sunt officiis consectetur modi libero, repellat aperiam,
      velit consequatur quas saepe ipsam atque mollitia labore voluptate
      necessitatibus rerum. Eligendi, accusamus corporis.
    </p>
    
  </article>
</body>
```

# Desarrollo Web



GOBIERNO DE LA  
CIUDAD DE MÉXICO

El código carga la imagen del archivo css.jpg y la muestra en pantalla en su tamaño original, como lo ilustra.



Recibidos (52) - gaussluiskof@ ... X (486) 당신은 언제 어디서나 공부하세요 X REPRODUCIENDO Imagenes x +  
← → ⌂ ⌂ 127.0.0.1:5500/Practica7/index.html

Lorem ipsum dolor sit amet consectetur adipisicing elit. Soluta sunt officiis consectetur modi libero, repellat aperiam, velit necessitatibus rerum. Eligendi, accusamus corporis.



119



La imagen se representa en su tamaño original, pero podemos definir un tamaño personalizado y algunos parámetros de configuración usando el resto de los atributos disponibles para el elemento <img>.

**width**—Este atributo declara el ancho de la imagen.

**height**—Este atributo declara la altura de la imagen.

**alt**—Este atributo especifica el texto que se muestra cuando la imagen no se puede cargar.



**srcset**—Este atributo nos permite especificar una lista de imágenes de diferentes resoluciones que el navegador puede cargar para el mismo elemento.

**sizes**—Este atributo especifica una lista de *media queries* (consulta de medios) junto con distintos tamaños de imágenes para que el navegador decida qué mostrar según la resolución de la pantalla.

**crossorigin**—Este atributo establece las credenciales para imágenes de origen cruzado (múltiples orígenes). Los valores posibles son **anonymous** (sin credenciales) y **usecredentials** (requiere credenciales).

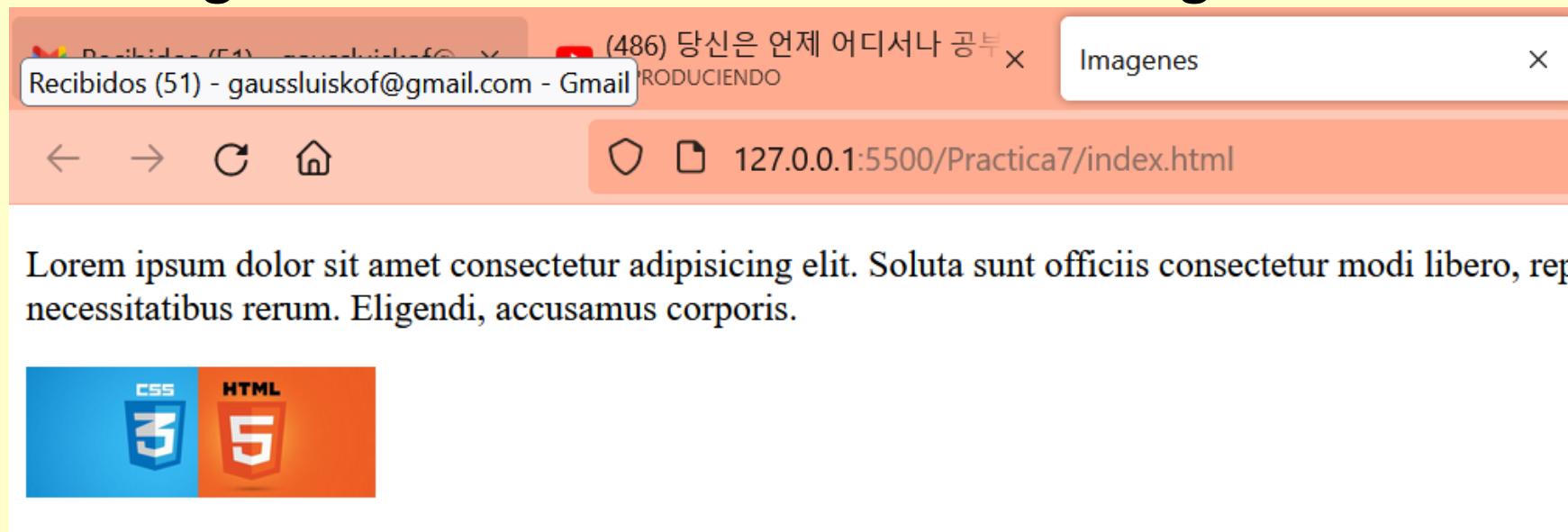


Los atributos `width` y `height` determinan las dimensiones de la imagen, pero no tienen en cuenta la relación. Si declaramos ambos valores sin considerar la proporción original de la imagen, el navegador deberá estirar o achatar la imagen para adaptarla a las dimensiones definidas. Para reducir la imagen sin cambiar la proporción original, podemos especificar uno solo de los atributos y dejar que el navegador calcule el otro.

```
<article>
  <p>
    Lorem ipsum dolor sit amet consectetur adipisicing elit.
    Soluta sunt officiis consectetur modi libero, repellat aperiam,
    velit consequatur quas saepe ipsam atque mollitia labore voluptate
    necessitatibus rerum. Eligendi, accusamus corporis.
  </p>
  
</article>
```



El código agrega el atributo **width** con el valor 150 al elemento **<img>** introducido en el ejemplo anterior. Esto reduce el ancho de la imagen a 150 píxeles, pero como el atributo **height** no se ha declarado, la altura de la imagen se calcula automáticamente considerando las proporciones originales de la imagen. El resultado se muestra en la Figura.





Algunas imágenes, como los iconos, son importantes porque otorgan un significado al resto del contenido, pero otras, como la imagen de estos ejemplos, actúan como complemento y se pueden eliminar sin que afecten al flujo de información. Cuando esta clase de información se encuentra presente, se puede utilizar el elemento `<figure>` para identificarla. Este elemento se suele implementar junto con el elemento `<figcaption>` para incluir texto descriptivo. En el siguiente ejemplo usamos estos dos elementos para identificar nuestra imagen y mostrar su título al usuario.

# Desarrollo Web



GOBIERNO DE LA  
CIUDAD DE MÉXICO

```
<body>
  <article>
    <p>
      Lorem ipsum dolor sit amet consectetur adipisicing elit.
      Soluta sunt officiis consectetur modi libero, repellat aperiam,
      velit consequatur quas saepe ipsam atque mollitia labore voluptate
      necessitatibus rerum. Eligendi, accusamus corporis.
    </p>
    <figure>
      
      <figcaption>HTML Y CSS JUNTOS</figcaption>
    </figure>
  </article>
</body>
```

# Desarrollo Web



GOBIERNO DE LA  
CIUDAD DE MÉXICO

M Recibidos (51) - gaussluiskof@g X

YouTube (486) 당신은 언제 어디서나 공부 X  
REPRODUCIENDO

Imagenes X +

← → ⌂ ⌂ 127.0.0.1:5500/Practica7/index.html

Lorem ipsum dolor sit amet consectetur adipisicing elit. Soluta sunt officiis consectetur modi libero, repellat a necessitatibus rerum. Eligendi, accusamus corporis.



HTML Y CSS JUNTOS

126



## Listados

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.

**<ul>**—Este elemento crea una lista de ítems sin orden. Está compuesto por etiquetas de apertura y cierre para agrupar los ítems (**<ul>** y **</ul>**) y trabaja junto con el elemento **<li>** para definir cada uno de los ítems de la lista.



**<ol>**—Este elemento crea una lista ordenada de ítems. Está compuesto por etiquetas de apertura y cierre para agrupar los ítems (**<ol>** y **</ol>**) y trabaja junto con el elemento **<li>** 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.

El elemento que usamos para crear la lista depende de las características del contenido. Por ejemplo, si el orden de los ítems no es importante, podemos usar el elemento **<ul>**. En esta clase de listas, los ítems se declaran entre las etiquetas **<ul>** con el elemento **<li>**, como muestra el siguiente ejemplo.

# Desarrollo Web



GOBIERNO DE LA  
CIUDAD DE MÉXICO

```
<body>
  <aside>
    <ul>
      <li>IT, Stephen King</li>
      <li>Carrie, Stephen King</li>
      <li>El Resplandor, Stephen King</li>
      <li>Misery, Stephen King</li>
    </ul>
  </aside>
</body>
```

The screenshot shows a web browser window with the following interface elements:

- Top bar: "Recibidos (51) - gaussluiskof@" (highlighted in orange), "YouTube (486) 당신은 언제 어디서나 공부 X REPRODUCIENDO" (highlighted in orange), and "Listas X".
- Address bar: "127.0.0.1:5500/Practica8/index.html".
- Content area:
  - A list of Stephen King books: IT, Stephen King; Carrie, Stephen King; El Resplandor, Stephen King; Misery, Stephen King.

130



El elemento **<ul>** presenta los ítems en el orden en el que se han declarado en el código y los identifica con un punto del lado izquierdo.

Si necesitamos declarar la posición de cada ítem, podemos crear la lista con el elemento **<ol>**. Este elemento crea una lista de ítems en el orden en el que se han declarado en el código, pero en lugar de usar puntos para identificarlos, les asigna un valor. Por defecto, los indicadores se crean con números, pero podemos cambiarlos con el atributo **type**. En el siguiente ejemplo, usamos letras mayúsculas.

# Desarrollo Web



GOBIERNO DE LA  
CIUDAD DE MÉXICO

```
<body>
  <aside>
    <ol type="A">
      <li>IT, Stephen King</li>
      <li>Carrie, Stephen King</li>
      <li>El Resplandor, Stephen King</li>
      <li>Misery, Stephen King</li>
    </ol>
  </aside>
</body>
```

The screenshot shows a web browser window with the following details:

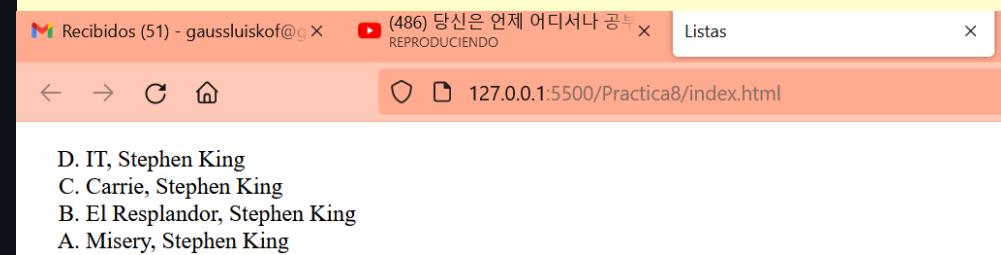
- Title bar: Recibidos (51) - gaussluiskof@gmail.com - Gmail
- Address bar: 127.0.0.1:5500/Practica8/index.html
- Content area:
  - A. IT, Stephen King
  - B. Carrie, Stephen King
  - C. El Resplandor, Stephen King
  - D. Misery, Stephen King

132



Aunque los ítems se muestran siempre en el orden en el que se han declarado en el código, podemos utilizar otros atributos del elemento `<ol>` para cambiar el orden de los indicadores. Por ejemplo, agregando el atributo `reversed` logramos que los indicadores se muestren en orden invertido.

```
<body>
  <aside>
    <ol type="A" reversed>
      <li>IT, Stephen King</li>
      <li>Carrie, Stephen King</li>
      <li>El Resplandor, Stephen King</li>
      <li>Misery, Stephen King</li>
    </ol>
  </aside>
</body>
```





Los elementos **<dl>**, **<dt>**, y **<dd>** trabajan de forma similar al elemento **<ul>**, pero su propósito es mostrar una lista de términos y descripciones. Los términos se representan por el elemento **<dt>** y las descripciones por el elemento **<dd>**. En el siguiente ejemplo, los usamos para agregar la descripción de los libros listados anteriormente.

# Desarrollo Web



GOBIERNO DE LA  
CIUDAD DE MÉXICO

```
<body>
  <aside>
    <dl>
      <dt>IT</dt>
      <dd>Tras lustros de tranquilidad y lejanía una antigua promesa
infantil les hace volver al lugar en el que vivieron su infancia y
juventud como una terrible pesadilla.</dd>
      <dt>Carrie</dt>
      <dd>Sus compañeros se burlan de ella, pero Carrie tiene un don.
Puede mover cosas con su mente. Este es su poder y su gran
problema.</dd>
      <dt>El Resplandor</dt>
      <dd>Al escritor Jack Torrance le es ofrecido un empleo como cuidador
del hotel Overlook durante el invierno junto a su familia.</dd>
      <dt>Misery</dt>
      <dd>Paul Sheldon es un famoso escritor de novelas románticas
ambientadas en la época victoriana, cuyo personaje principal se llama
Misery Chastain.</dd>
    </dl>
  </aside>
</body>
```

# Desarrollo Web



GOBIERNO DE LA  
CIUDAD DE MÉXICO

Recibidos (51) - gaussluiskof@g Recibidos (51) - gaussluiskof@g x (1) WhatsApp (1) WhatsApp x (486) 3:30 AM Coding Session - REPRODUCIENDO x Listas +

← → C ⌂ 127.0.0.1:5500/Practica8/index.html

IT

Tras lustros de tranquilidad y lejanía una antigua promesa infantil les hace volver al lugar en el que vivieron su infancia y juventud como una terrible pesadilla.  
Carrie

Sus compañeros se burlan de ella, pero Carrie tiene un don. Puede mover cosas con su mente. Este es su poder y su gran problema.

El Resplandor

Al escritor Jack Torrance le es ofrecido un empleo como cuidador del hotel Overlook durante el invierno junto a su familia.

Misery

Paul Sheldon es un famoso escritor de novelas románticas ambientadas en la época victoriana, cuyo personaje principal se llama Misery Chastain.

136



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.

# Desarrollo Web



GOBIERNO DE LA  
CIUDAD DE MÉXICO

El elemento **<blockquote>** es similar al elemento **<p>**, pero como también incluye márgenes a los costados, se ha usado tradicionalmente para presentar listas de valores, como lo demuestra el siguiente ejemplo.

```
<body>
  <aside>
    <blockquote>IT</blockquote>
    <blockquote>Carrie</blockquote>
    <blockquote>El Resplandor</blockquote>
    <blockquote>Misery</blockquote>
  </aside>
</body>
```

IT  
Carrie  
El Resplandor  
Misery

El listado generado por el elemento **<blockquote>** se muestra igual que otras listas, pero no incluye puntos o números para identificar cada ítem.

138



En sitios web modernos son comunes las herramientas que revelan información adicional cuando el usuario lo requiere. Para ofrecer esta posibilidad, HTML incluye el elemento **<details>**. Este elemento muestra un título, especificado por el elemento **<summary>**, y contenido que se puede representar por elementos comunes como **<p>** o **<blockquote>**.

Debido a esto, el elemento **<details>** se puede utilizar para revelar una lista de valores, como lo hacemos en el siguiente ejemplo.

# Desarrollo Web



GOBIERNO DE LA  
CIUDAD DE MÉXICO

```
<body>
  <details>
    <summary>My Books</summary>
    <p>IT</p>
    <p>Carrie</p>
    <p>El Resplendor</p>
    <p>Misery</p>
  </details>
</body>
```

Recibidos (52) - gaussluiskof@g... X

(486) 3:30 AM Coding Session - REPRODUCIENDO

Listas X

← → ⏪ ⏪ 127.0.0.1:5500/Practica8/index.html

▼ My Books

IT

Carrie

El Resplendor

Misery

140



## 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.

# Desarrollo Web



GOBIERNO DE LA  
CIUDAD DE MÉXICO

Para incluir una tabla en el documento, primero tenemos que declarar el elemento **<table>** y luego describir las filas una por una con los elementos **<tr>** y **<td>**, como muestra el siguiente ejemplo.

```
<body>
  <article>
    <table>
      <tr>
        <td>IT</td>
        <td>Stephen King</td>
        <td>1986</td>
      </tr>
      <tr>
        <td>Carrie</td>
        <td>Stephen King</td>
        <td>1974</td>
      </tr>
      <tr>
        <td>El Resplandor</td>
        <td>Stephen King</td>
        <td>1977</td>
      </tr>
    </table>
  </article>
</body>
```

IT	Stephen King	1986
Carrie	Stephen King	1974
El Resplandor	Stephen King	1977



Debido a que el navegador interpreta el documento de forma secuencial desde la parte superior a la inferior, cada vez que declaramos una fila, tenemos que declarar las celdas que corresponden a esa fila y su contenido.

La primer celda de cada fila representa el título del libro, la segunda celda representa el autor, y la tercera celda el año de publicación. Cuando el navegador abre este documento, muestra la información en el orden en el que se ha declarado en el código y con el tamaño determinado por el contenido de las celdas.

# Desarrollo Web



GOBIERNO DE LA  
CIUDAD DE MÉXICO

Si queremos incluir una cabecera para describir el contenido de cada columna, podemos crear una fila de celdas adicional representadas con el elemento **<th>**.

Título	Autor	Año
IT	Stephen King	1986
Carrie	Stephen King	1974
El Resplandor	Stephen King	1977

```
<body>
  <article>
    <table>
      <tr>
        <th>Título</th>
        <th>Autor</th>
        <th>Año</th>
      </tr>
      <tr>
        <td>IT</td>
        <td>Stephen King</td>
        <td>1986</td>
      </tr>
      <tr>
        <td>Carrie</td>
        <td>Stephen King</td>
        <td>1974</td>
      </tr>
      <tr>
        <td>El Resplandor</td>
        <td>Stephen King</td>
        <td>1977</td>
      </tr>
    </table>
  </article>
</body>
```



Por defecto, los navegadores muestran las cabeceras con el texto en negrita y centrado.

Las celdas se pueden estirar para que ocupen más de una columna con los atributos colspan y rowspan. Por ejemplo, podemos usar solo una celda de cabecera para identificar el título y el autor del libro.

El ejemplo siguiente incluye una celda de cabecera con el título **Libro** para las primeras dos columnas. Debido al valor asignado al atributo colspan, esta celda se estira para que ocupe el espacio de dos. El resultado se muestra a continuación.

# Desarrollo Web



GOBIERNO DE LA  
CIUDAD DE MÉXICO

```
<body>
  <article>
    <table>
      <tr>
        <th colspan="2">Libro</th>
        <th>Año</th>
      </tr>
      <tr>
        <td>IT</td>
        <td>Stephen King</td>
        <td>1986</td>
      </tr>
      <tr>
        <td>Carrie</td>
        <td>Stephen King</td>
        <td>1974</td>
      </tr>
      <tr>
        <td>El Resplandor</td>
        <td>Stephen King</td>
        <td>1977</td>
      </tr>
    </table>
  </article>
</body>
```

Nueva pestaña Tablas

Libro	Año
IT	Stephen King 1986
Carrie	Stephen King 1974
El Resplandor	Stephen King 1977

147



## Atributos Globales

La mayoría de los navegadores actuales automáticamente traducen el contenido del documento cuando detectan que se ha escrito en un idioma diferente al del usuario, pero en algunos casos la página puede incluir frases o párrafos enteros que no se deben alterar, como nombres de personas o títulos de películas. Para controlar el proceso de traducción, HTML ofrece un atributo global llamado **translate**. Este atributo puede tomar dos valores: **yes** y **no**. Por defecto, el valor es **yes (si)**. En el siguiente ejemplo, usamos un elemento `<span>` para especificar la parte del texto que no se debería traducir.

# Desarrollo Web



GOBIERNO DE LA  
CIUDAD DE MÉXICO

```
<body>
  <p>My favorite movie is <span translate="no">Two Roads</span></p>
</body>
```

The screenshot shows a web browser window with two tabs. The active tab is titled "Atributos Globales" and displays the text "My favorite movie is Two Roads". The address bar shows the URL "127.0.0.1:5500/Practica". The browser interface includes standard navigation buttons (back, forward, search, home) and a shield icon indicating a secure connection.

149



## Formularios

Los formularios son herramientas que podemos 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. La Figura siguiente muestra algunas de las herramientas facilitadas este fin.

Texto

Botones de Opción  1  2  3

Clave .....

Casilla de Control



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.

**name**—Este atributo especifica el nombre del formulario. También se encuentra disponible para otros elementos, pero es particularmente útil para elementos de formulario.



**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 **\_blank** (nueva ventana), **\_self** (mismo recuadro), **\_parent** (recuadro padre), y **\_top** (la ventana que contiene el recuadro). El valor **\_self** se declara por defecto, lo que significa que la respuesta recibida desde el servidor se mostrará en la misma ventana.

**enctype**—Este atributo declara la codificación aplicada a los datos que envía el formulario. Puede tomar tres valores: **application/x-www-form-urlencoded** (los caracteres son codificados), **multipart/form-data** (los caracteres no son codificados), **text/plain** (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 **<meta>**.

El siguiente ejemplo define un formulario básico. El atributo **name** identifica el formulario con el nombre "formulario", el atributo **method** determina que los datos se incluirán en la URL (GET), y el atributo **action** declara que procesar.php es el archivo que se ejecutará en el servidor para procesar la información y devolver el resultado.

# Desarrollo Web



GOBIERNO DE LA  
CIUDAD DE MÉXICO

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Formularios</title>
</head>
<body>
    <section>
        <form name="formulario" method="get" action="procesar.php">
            </form>
    </section>
</body>
</html>
```



Abra el documento en su navegador. En este momento no verá nada en la pantalla porque el formulario se ha declarado vacío. A continuación, desarrollaremos su contenido.

## Elementos

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.

**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.



Para incluir un formulario en nuestro documento, tenemos que declararlo con el elemento **<form>**, como hemos hecho en el ejemplo anterior, y luego incorporar en su interior todos los elementos que el usuario necesitará para insertar la información y enviarla al servidor. Por ejemplo, si queremos que el usuario inserte su nombre y edad, tenemos que incluir dos campos de entrada para texto y un tercer elemento para crear el botón con el que enviar el formulario.

La información insertada en el formulario se envía al servidor para ser procesada. Para que el servidor pueda identificar cada valor, los elementos deben incluir el atributo **name**.

# Desarrollo Web



GOBIERNO DE LA  
CIUDAD DE MÉXICO

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Formularios</title>
</head>
<body>
    <section>
        <form name="formulario" method="get" action="procesar.php">
            <p><input type="text" name="nombre"></p>
            <p><input type="text" name="edad"></p>
            <p><input type="submit"></p>
        </form>
    </section>
</body>
</html>
```

# Desarrollo Web



GOBIERNO DE LA  
CIUDAD DE MÉXICO

A screenshot of a web browser window. The title bar says "Formularios". The address bar shows the URL "127.0.0.1:5500/Practica". The page content area contains two empty input fields and a button labeled "Enviar consulta".

Nueva pestaña

Formularios

← → C ⌄ 127.0.0.1:5500/Practica

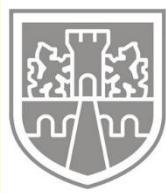
Enviar consulta

166



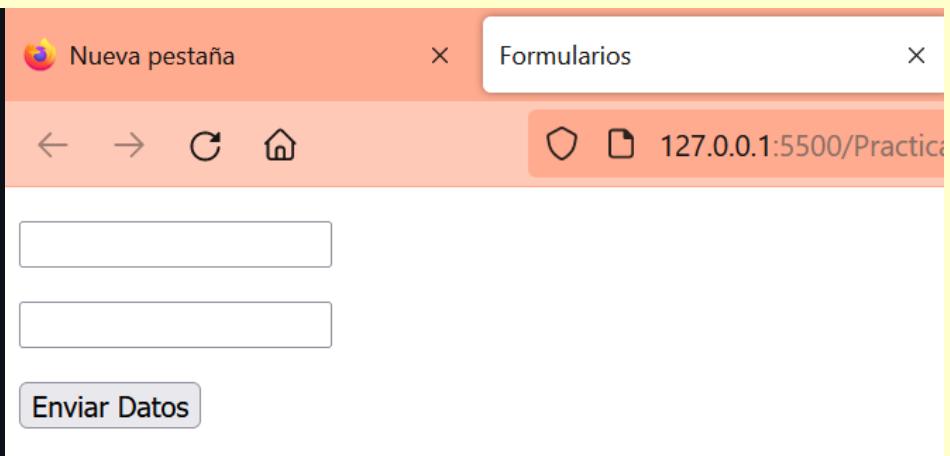
En el ejemplo anterior, llamamos a los campos de entrada "nombre" y "edad" (el elemento **<input>** que crea el botón para enviar el formulario no necesita un nombre porque no envía ningún dato al servidor).

Los elementos de formulario no generan un salto de línea; se muestran en la pantalla uno detrás del otro. Si queremos que el navegador muestre un elemento en cada línea, tenemos que modificar el diseño nosotros mismos. En el ejemplo, usamos elementos **<p>** para separar los elementos del formulario, pero este diseño normalmente se logra a través de estilos CSS.



Otro atributo que podemos usar en este ejemplo es value. El tipo de entrada **submit** crea un botón para enviar el formulario. Por defecto, los navegadores le dan al botón el título **Enviar (Submit)**, pero podemos usar el atributo **value** para modificarlo.

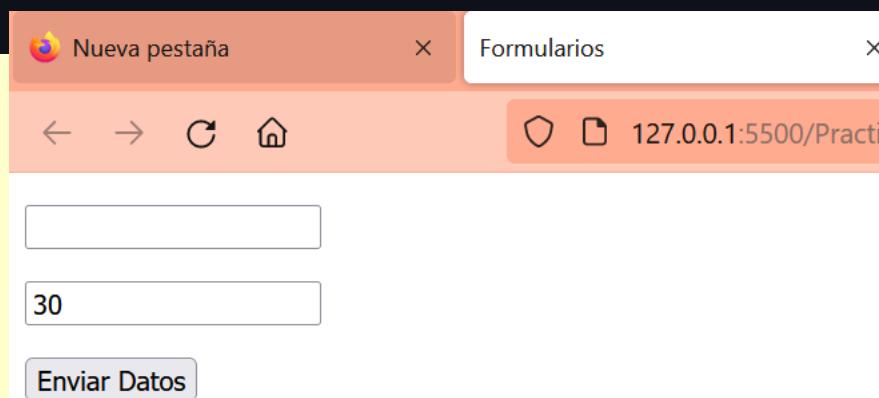
```
<body>
  <section>
    <form name="formulario" method="get" action="procesar.php">
      <p><input type="text" name="nombre"></p>
      <p><input type="text" name="edad"></p>
      <p><input type="submit" value="Enviar Datos"></p>
    </form>
  </section>
</body>
```





El atributo **value** también se puede usar para declarar el valor inicial de un elemento. Por ejemplo, podemos insertar la edad del usuario en el campo **edad** si ya la conocemos.

```
<body>
  <section>
    <form name="formulario" method="get" action="procesar.php">
      <p><input type="text" name="nombre"></p>
      <p><input type="text" name="edad" value="30"></p>
      <p><input type="submit" value="Enviar Datos"></p>
    </form>
  </section>
</body>
```





Los formularios necesitan incluir descripciones que le indiquen al usuario lo datos que debe introducir. Por esta razón, HTML incluye el elemento **<label>** tienen que asociarse al elemento al que están describiendo. Para asociar un elemento **<label>** con el elemento de formulario correspondiente, podemos incluir el elemento de formulario dentro del elemento **<label>**, como muestra el siguiente ejemplo.



```
<body>
  <section>
    <form name="formulario" method="get" action="procesar.php">
      <p>
        <label>Nombre: <input type="text" name="nombre"></label>
      </p>
      <p>
        <label>Edad: <input type="text" name="edad"></label>
      </p>
      <p><input type="submit" value="Enviar"></p>
    </form>
  </section>
</body>
```

# Desarrollo Web



GOBIERNO DE LA  
CIUDAD DE MÉXICO

Nueva pestaña × Formularios ×

← → ⏪ ⏪ 127.0.0.1:5500/Practica

Nombre:

Edad:

**Enviar**

172



Otra alternativa para asociar un elemento <label> con su elemento de formulario es implementando el atributo for. El atributo for conecta el elemento <label> con el elemento de formulario por medio del valor del atributo id, según ilustra el siguiente ejemplo

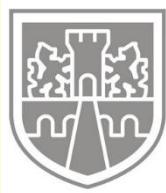
```
<body>
  <section>
    <form name="formulario" method="get" action="procesar.php">
      <p>
        <label for="nombre">Nombre: </label>
        <input type="text" name="nombre" id="nombre"></label>
      </p>
      <p>
        <label for="edad">Edad: </label>
        <input type="text" name="edad" id="edad"></label>
      </p>
      <p><input type="submit" value="Enviar"></p>
    </form>
  </section>
</body>
```



Los campos de entrada usados eran de tipo text, lo que significa que los usuarios pueden introducir cualquier clase de texto que deseen, pero esto no es lo que necesitamos para este formulario. El primer campo espera un nombre, por lo que no debería permitir que se introduzcan números o textos muy extensos, y el segundo campo espera la edad del usuario, por lo que no debería aceptar ningún tipo de carácter excepto números. Para determinar cuántos caracteres se pueden introducir, el elemento **<input>** debe incluir los siguientes atributos.

**maxlength**—Este atributo especifica el máximo número de caracteres que se permite introducir en el campo.

**minlength**—Este atributo especifica el mínimo número de caracteres que se permite introducir en el campo.



El siguiente ejemplo limita el nombre a un máximo de 15 caracteres.

```
<body>
  <section>
    <form name="formulario" method="get" action="procesar.php">
      <p><label>Nombre: <input type="text" name="nombre" maxlength="15"></label></p>
      <p><label>Edad: <input type="text" name="edad"></label></p>
      <p><input type="submit" value="Enviar"></p>
    </form>
  </section>
</body>
```

Nueva pestaña × Formularios ×

← → ⏪ ⏪ 127.0.0.1:5500/Practica

Nombre: jose luis lopez

Edad:

Enviar



Si el usuario escribe un número en el campo nombre o letras en el campo edad, el navegador considerará la entrada válida. Para controlar lo que el usuario puede introducir, tenemos que declarar un tipo de campo diferente con el atributo **type**.

El siguiente ejemplo declara el tipo **number** para el campo **edad** para permitir que solo se introduzcan números, e incluye otros campos para que el usuario pueda declarar su cuenta de correo, número de teléfono y sitio web.

# Desarrollo Web



GOBIERNO DE LA  
CIUDAD DE MÉXICO

```
<body>
  <section>
    <form name="formulario" method="get" action="procesar.php">
      <p><label>Nombre: <input type="text" name="nombre" maxlength="15"></label></p>
      <p><label>Edad: <input type="number" name="edad"></label></p>
      <p><label>Correo: <input type="email" name="correo"></label></p>
      <p><label>Teléfono: <input type="tel" name="telefono"></label></p>
      <p><label>Sitio Web: <input type="url" name="sitioweb"></label></p>
      <p><input type="submit" value="Enviar"></p>
    </form>
  </section>
</body>
```

# Desarrollo Web



GOBIERNO DE LA  
CIUDAD DE MÉXICO

A screenshot of a web browser window. The title bar says "Nueva pestaña" and "Formularios". The address bar shows a shield icon, a document icon, and the URL "127.0.0.1:5500/Practica". The main content area contains a form with fields for Nombre, Edad, Correo, Teléfono, and Sitio Web, each with an associated input field. At the bottom is a button labeled "Enviar".

Nueva pestaña × Formularios ×

← → ⌛ ⌄ 127.0.0.1:5500/Practica

Nombre:

Edad:  ⌄

Correo:

Teléfono:

Sitio Web:

Enviar



Por defecto los navegadores incluyen flechas en el lado derecho de un campo de tipo number con las que podemos seleccionar un número. Para establecer restricciones en los números que el usuario puede seleccionar con estas flechas o controlar los números permitidos, los elementos <input> de este tipo pueden incluir los siguientes atributos.

**min**—El valor de este atributo determina el valor mínimo que acepta el campo.

**max**—El valor de este atributo determina el valor máximo que acepta el campo.



**step**—El valor de este atributo determina el número por el cual el valor del campo se puede incrementar o reducir. Por ejemplo, si declaramos el valor 5 para este atributo y un valor mínimo de 0 y uno máximo de 10 para el campo, el navegador no nos dejará introducir valores entre 0 y 5 o 5 y 10.

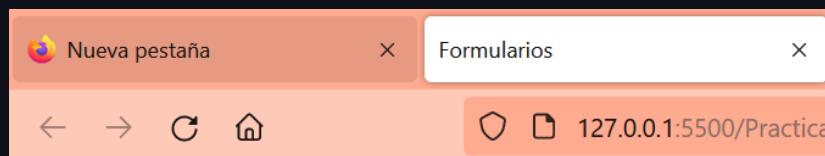
El siguiente ejemplo restringe el valor insertado en el campo edad de nuestro formulario a un mínimo de 13 y un máximo de 100.

# Desarrollo Web



GOBIERNO DE LA  
CIUDAD DE MÉXICO

```
<body>
  <section>
    <form name="formulario" method="get" action="procesar.php">
      <p><label>Nombre: <input type="text" name="nombre" maxlength="15"></label></p>
      <p><label>Edad: <input type="number" name="edad" min="13" max="100"></label></p>
      <p><label>Correo: <input type="email" name="correo"></label></p>
      <p><label>Teléfono: <input type="tel" name="telefono"></label></p>
      <p><label>Sitio Web: <input type="url" name="sitioweb"></label></p>
      <p><input type="submit" value="Enviar"></p>
    </form>
  </section>
</body>
```



Nombre:

Edad:

Correo:

Teléfono:

Sitio Web:



Otro tipo de entrada que implementa estos mismo atributos es **range**. El tipo range crea un campo que nos permite seleccionar un número desde un rango de valores. El valor inicial se establece de acuerdo a los valores de los atributos **min** y **max**, pero podemos declarar un valor específico con el atributo **value**, como muestra el siguiente ejemplo.

Los navegadores muestran un campo de entrada de tipo range como un control que el usuario puede deslizar para seleccionar un valor.

# Desarrollo Web



GOBIERNO DE LA  
CIUDAD DE MÉXICO

```
<body>
  <section>
    <form name="formulario" method="get" action="procesar.php">
      <p><label>Nombre: <input type="text" name="nombre" maxlength="15"></label></p>
      <p><label>Edad: <input type="range" name="edad" min="13" max="100"></label></p>
      <p><label>Correo: <input type="email" name="correo"></label></p>
      <p><label>Teléfono: <input type="tel" name="telefono"></label></p>
      <p><label>Sitio Web: <input type="url" name="sitioweb"></label></p>
      <p><input type="submit" value="Enviar"></p>
    </form>
  </section>
</body>
```

Nombre:

Edad:

Correo:

Teléfono:

Sitio Web:

183



El tipo radio crea un botón circular que se resalta cuando se selecciona. Esto nos permite crear una lista de valores que el usuario puede seleccionar con solo hacer clic en el botón correspondiente. Para ofrecer al usuario todas las opciones disponibles, tenemos que insertar un elemento **<input>** por cada opción. Los elementos **<input>** se asocian entre ellos por medio del valor del atributo **name**, y el valor de cada opción se define por el atributo **value**, como muestra el siguiente ejemplo.

En el formulario declaramos cuatro elementos **<input>** de tipo **radio** para ofrecer distintas edades que el usuario puede elegir. Como todos los elementos tienen el mismo nombre (**edad**), se consideran parte del mismo grupo y, por lo tanto, solo una de las opciones puede ser seleccionada a la vez.

# Desarrollo Web



GOBIERNO DE LA  
CIUDAD DE MÉXICO

```
<body>
  <form name="formulario" method="get" action="procesar.php">
    <p><label>Nombre: <input type="text" name="nombre" maxlength="15"></label></p>
    <p><label><input type="radio" name="edad" value="15" checked> 15
      Años</label></p>
    <p><label><input type="radio" name="edad" value="30"> 30
      Años</label></p>
    <p><label><input type="radio" name="edad" value="45"> 45
      Años</label></p>
    <p><label><input type="radio" name="edad" value="60"> 60
      Años</label></p>
    <p><input type="submit" value="Enviar"></p>
  </form>
</body>
```

YouTube (489) 3 AM Coding Session - Loi x REPRODUCIENDO

Formularios

127.0.0.1:5500/Práctica

Nombre:

15 Años

30 Años

45 Años

60 Años

Enviar

185



Además del atributo **name**, también implementamos un atributo booleano llamado **checked**. Este atributo le dice al navegador que seleccione el botón cuando se carga el documento, lo cual determina la edad de 15 años como el valor por defecto.

El tipo **checkbox** genera un tipo de entrada similar. En este caso, el usuario puede seleccionar múltiples valores haciendo clic en las casillas correspondientes. Cuando el usuario selecciona una o más opciones, los valores de todos esos elementos se envían al servidor. Esta clase de campo de entrada también puede incluir el atributo **checked** para seleccionar opciones por defecto. En nuestro ejemplo seleccionamos dos valores: 15 y 30.

# Desarrollo Web



GOBIERNO DE LA  
CIUDAD DE MÉXICO

```
<body>
<form name="formulario" method="get" action="procesar.php">
    <p><label>Nombre: <input type="text" name="nombre" maxlength="15"></label></p>
    <p><label><input type="checkbox" name="edad15" value="15" checked> 15
        Años</label></p>
    <p><label><input type="checkbox" name="edad30" value="30" checked> 30
        Años</label></p>
    <p><label><input type="checkbox" name="edad45" value="45"> 45
        Años</label></p>
    <p><label><input type="checkbox" name="edad60" value="60"> 60
        Años</label></p>
    <p><input type="submit" value="Enviar"></p>
</form>
</body>
```

Reproductor de video: (489) 3 AM Coding Session - Lo X  
REPRODUCIENDO

Formularios

← → ⏪ ⏹ 127.0.0.1:5500/

Nombre:

15 Años

30 Años

45 Años

60 Años

187



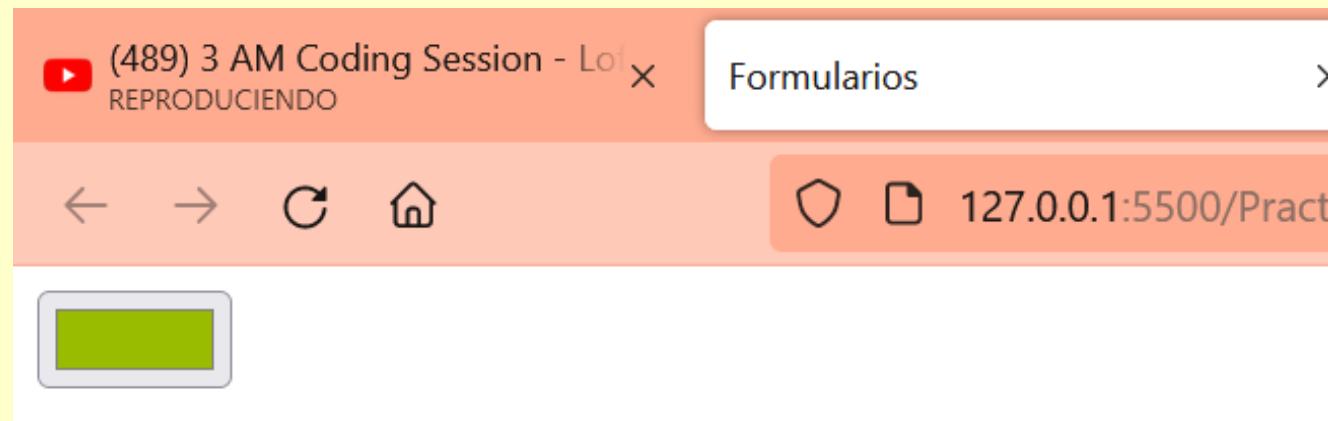
Otro tipo de entrada que genera una herramienta visual es date. Este control le ayuda al usuario a seleccionar una fecha. Algunos navegadores lo implementan como un calendario que se muestra cada vez que el usuario hace clic en el campo. El valor enviado al servidor por este tipo de campos tiene la sintaxis año-mes-día, por lo que si queremos especificar un valor inicial o el navegador no facilita una herramienta para seleccionarlo, debemos declararlo en este formato.

```
✓ <body>
  |   <input type="date" name="fecha" value="2012-10-29">
</body>
```



Además de los tipos de campos de entrada disponibles para introducir fechas, existe un tipo de campo llamado color que ofrece una interfaz predefinida para seleccionar un valor. Generalmente, el valor esperado por estos campos es un número hexadecimal, como #00FF00.

```
<body>
  <input type="color" name="micolor" value="#99BB00">
</body>
```





Hasta el momento hemos usado un elemento `<input>` de tipo **submit** para crear el botón que el usuario tiene que pulsar para enviar el formulario. Este tipo de entrada crea un botón estándar identificado con un título. Si queremos mejorar el diseño, podemos implementar el tipo de entrada **image** que crea un botón con una imagen. Estos tipos de campos requieren el atributo **src** con la URL del archivo que contiene la imagen que queremos usar para el botón, y pueden incluir los atributos **width** y **height** para definir su tamaño.

El formulario del ejemplo crea el botón para enviar el formulario con un elemento `<input>` de tipo **image**. El elemento carga la imagen del archivo `enviar.png` y la muestra en pantalla.



Cuando el usuario hace clic en la imagen, el formulario se envía del mismo modo que con los botones que hemos usado anteriormente.

```
<body>
  <form name="formulario" method="get" action="procesar.php">
    <p><label>Nombre: <input type="text" name="nombre"></label></p>
    <p><label>Edad: <input type="text" name="edad"></label></p>
    <p><input type="image" src="enviar.png" width="100"></p>
  </form>
</body>
```

YouTube (489) [ Playlist ] 코딩할때 듣기 X REPRODUCIENDO Formularios

← → ⏪ ⏩ 127.0.0.1:5500

Nombre:

Edad:

Enviar



HTML ofrece un elemento más versátil para crear botones llamado **<button>**. Este elemento incluye el atributo **type** para determinar el tipo de botón que queremos generar. Por ejemplo, si queremos crear un botón para enviar el formulario, debemos declarar el valor **submit**.

El elemento **<button>** crea un botón estándar con las mismas características que el botón creado por el elemento **<input>**. La diferencia es que el título de estos botones se define entre las etiquetas de apertura y cierre, lo cual nos permite usar otros elementos HTML e incluso imágenes para declararlo. Por esta razón, el elemento **<button>** es el que se prefiere cuando queremos personalizarlo usando estilos CSS o cuando queremos usarlo para ejecutar códigos JavaScript.

# Desarrollo Web



GOBIERNO DE LA  
CIUDAD DE MÉXICO

```
<body>
  <form name="formulario" method="get" action="procesar.php">
    <p><label>Nombre: <input type="text" name="nombre"></label></p>
    <p><label>Edad: <input type="text" name="edad"></label></p>
    <p><button type="submit">Enviar Formulario</button></p>
  </form>
</body>
```

The screenshot shows a web browser window with the following details:

- Address Bar:** 127.0.0.1:5500/
- Content Area:**
  - Nombre:**
  - Edad:**
  - Submit Button:**



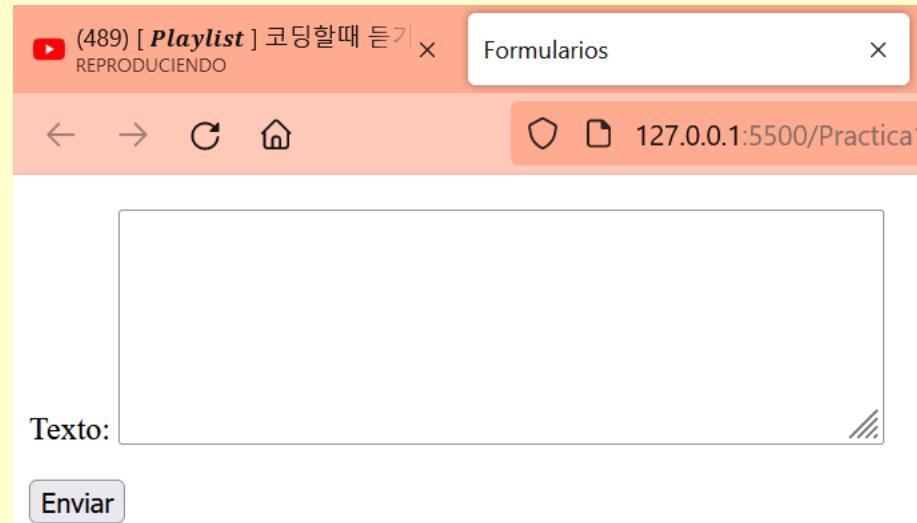
El elemento **<input>** permite al usuario insertar o seleccionar varios tipos de valores, pero los campos de entrada generados por este elemento nos dejan introducir una sola línea de texto. HTML ofrece el elemento **<textarea>** para insertar múltiples líneas de texto. El elemento está compuesto por etiquetas de apertura y cierre, y puede incluir los atributos **rows** y **cols** para definir el ancho y la altura del área en caracteres.

# Desarrollo Web



GOBIERNO DE LA  
CIUDAD DE MÉXICO

```
<body>
  <form name="formulario" method="get" action="procesar.php">
    <p><label>Texto: <textarea name="texto" cols="50"
      rows="6"></textarea></label></p>
    <p><input type="submit" value="Enviar"></p>
  </form>
</body>
```



195



HTML ofrece el elemento **<select>** para presentar una lista de valores al usuario. Cuando el usuario hace clic en este elemento, la lista se muestra en una ventana desplegable, y luego el valor seleccionado por el usuario se inserta en el campo. Debido a que el elemento **<select>** no genera un campo de entrada, el usuario no puede insertar valores distintos de los incluidos en la lista. El elemento **<select>** trabaja junto con el elemento **<option>** para definir las opciones.

El elemento **<select>** debe incluir el atributo name para identificar el valor, y cada elemento **<option>** debe incluir el atributo value para definir el valor que representa.

# Desarrollo Web



GOBIERNO DE LA  
CIUDAD DE MÉXICO

```
<body>
  <form name="formulario" method="get" action="procesar.php">
    <p>
      <label for="listado">Libros: </label>
      <select name="libro" id="listado">
        <option value="1">IT</option>
        <option value="2">Carrie</option>
        <option value="3">El Resplandor</option>
        <option value="4">Misery</option>
      </select>
    </p>
    <p><input type="submit" value="Enviar"></p>
  </form>
</body>
```

La captura de pantalla muestra una interfaz de usuario en un navegador web. En la parte superior, hay una barra de título que dice "YouTube (489) [ Playlist ] 코딩할때 듣기 X REPRODUCIENDO Formularios" y una barra de dirección que dice "127.0.0.1:5500/Formularios". El contenido principal es un formulario con un campo de selección desplegable titulado "Libros:" que contiene las siguientes opciones:

- IT
- Carrie
- El Resplandor
- Misery

A continuación del campo de selección, hay un botón "Enviar".



Otra manera de crear una lista predefinida es con el elemento **<datalist>**. Este elemento define una lista de ítems que, con la ayuda del atributo **list**, se puede usar como sugerencia en un campo de entrada. Al igual que las opciones para el elemento **<select>**, las opciones para este elemento se definen por elementos **<option>**, pero en este caso deben incluir el atributo **label** con una descripción del valor. El siguiente ejemplo crea un formulario con un campo de entrada para insertar un número telefónico. El código incluye un elemento **<datalist>** con dos elementos **<option>** que definen los valores que queremos sugerir al usuario.

# Desarrollo Web



GOBIERNO DE LA  
CIUDAD DE MÉXICO

```
<body>
  <form name="formulario" method="get" action="procesar.php">
    <datalist id="datos">
      <option value="5541353383" label="Teléfono 1">
      <option value="5512286478" label="Teléfono 2">
    </datalist>
    <p><label>Teléfono: <input type="tel" name="telefono" list="datos"></label></p>
    <p><input type="submit" value="Enviar"></p>
  </form>
</body>
```

La captura de pantalla muestra una interfaz de usuario para un formulario de datos telefónicos. En la parte superior, hay un encabezado que indica que se está reproduciendo un video de YouTube con el título "(489) [ Playlist ] 코딩할때 듣기". El formulario se titula "Formularios". La URL en la barra de direcciones es "127.0.0.1:5500/Prac". El formulario contiene un campo de texto para el teléfono con el valor "5541353383" y una lista desplegable que incluye "Teléfono 1" y "Teléfono 2".

Teléfono:

5541353383
Teléfono 1
Teléfono 2

Enviar

199



Para conectar un campo de entrada con un elemento **<datalist>**, tenemos que incluir el atributo **list** en el elemento **<input>** con el mismo valor que usamos para identificar el elemento **<datalist>**. Para este propósito, en el formulario incluimos el atributo **id** en el elemento **<datalist>** con el valor "datos" y luego asignamos este mismo valor al atributo **list** del elemento **<input>**. En consecuencia, el campo de entrada muestra una flecha que despliega una lista de valores predefinidos que el usuario puede seleccionar para completar el formulario.



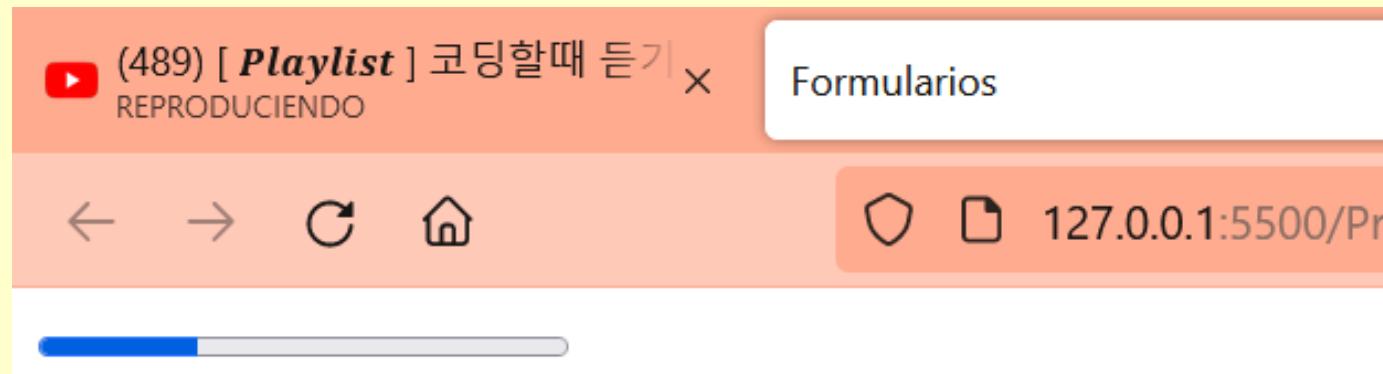
HTML incluye otros dos elementos que podemos usar en un formulario: **<progress>** y **<meter>**, que no se consideran elementos de formulario, pero debido a que representan medidas, son realmente útiles cuando nuestro formulario produce esta clase de información.

El elemento **<progress>** se usa para informar del progreso en la ejecución de una tarea. Requiere dos atributos que determinan el valor actual y el máximo. El atributo **value** indica el progreso logrado hasta el momento, y el atributo **max** declara el valor que necesitamos alcanzar para dar por finalizada la tarea.



Los navegadores representan este elemento con una barra de dos colores. Por defecto, la porción de la barra que representa el progreso se muestra en color azul. Si el navegador no reconoce el elemento, el valor entre las etiquetas de apertura y cierre se muestra en su lugar.

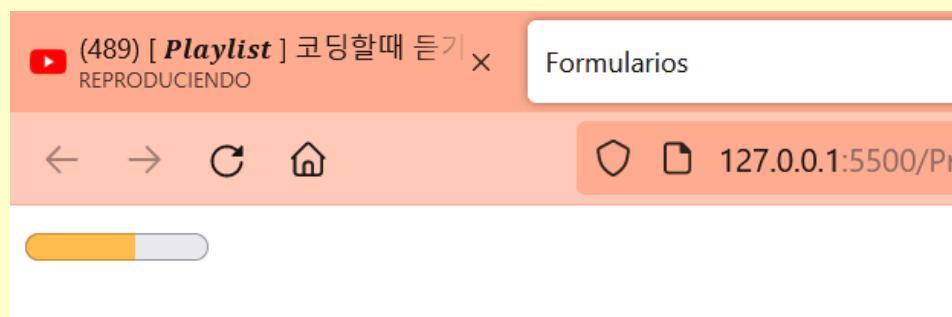
```
<body>
  <progress value="30" max="100">0%</progress>
</body>
```





El elemento **<meter>** se usa para mostrar una escala, pero no representa progreso. Su propósito es representar un rango de valores predefinidos (por ejemplo, el espacio ocupado en un disco duro). Este elemento cuenta con varios atributos asociados. Los atributos **min** y **max** determinan los límites del rango, **value** determina el valor medido, y **low**, **high** y **optimum** se usan para segmentar el rango en secciones diferenciadas y declarar la posición óptima.

```
<body>
  <meter value="60" min="0" max="100" low="40" high="80" optimum="100">60</meter>
</body>
```





El código del ejemplo genera una barra en la pantalla que muestra un nivel de 60 en una escala de 0 a 100 (de acuerdo con los valores declarados por los atributos **value**, **min** y **max**). El color de la barra generada por el elemento depende de los niveles determinados por los atributos **low**, **high** y **optimum**. Debido a que el valor actual en nuestro ejemplo se encuentra entre los valores de los atributos **low** y **high**, la barra se muestra en color amarillo.

**Enviando el formulario.** Los pares nombre/valor producidos por un formulario se envían al servidor dentro de estas solicitudes HTTP.



Si el método se ha declarado como **GET**, los pares nombre/valor se agregan al final de la URL, pero si el método se ha declarado como **POST**, los valores se incluyen en el cuerpo de la solicitud. Esto significa que la información enviada con el método **GET** es visible para el usuario (el usuario puede ver la URL con todos los pares nombre/valor en la barra de navegación del navegador), pero la información enviada con el método **POST** se oculta dentro de la solicitud. En consecuencia, si la información es sensible o privada, debemos usar el método **POST**, pero si la información no es sensible, como valores de búsqueda insertados por el usuario, podemos usar el método **GET**.



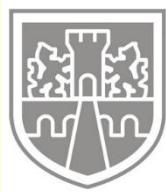
Así mismo, tenemos que considerar que el método **POST** se puede usar para enviar una cantidad ilimitada de información, pero el método **GET** tiene que adaptarse a las limitaciones presentadas por las URL. Esto se debe a que el largo de una URL es limitado. Si la información insertada en el formulario es muy extensa, se podría perder.

El siguiente ejemplo presenta un formulario con un único campo de entrada para ilustrar cómo funciona este proceso. El elemento **<input>** se identifica con el nombre "val", y el método utilizado para enviar la información se declara como **GET**, lo que significa que el valor insertado por el usuario se agregará a la URL.



Así mismo, tenemos que considerar que el método **POST** se puede usar para enviar una cantidad ilimitada de información, pero el método **GET** tiene que adaptarse a las limitaciones presentadas por las URL. Esto se debe a que el largo de una URL es limitado. Si la información insertada en el formulario es muy extensa, se podría perder.

El siguiente ejemplo presenta un formulario con un único campo de entrada para ilustrar cómo funciona este proceso. El elemento **<input>** se identifica con el nombre "val", y el método utilizado para enviar la información se declara como **GET**, lo que significa que el valor insertado por el usuario se agregará a la URL.



En el ejemplo un archivo llamado procesar.php se declara a cargo de procesar la información. Cuando el usuario pulsa el botón **Enviar**, el navegador crea una solicitud HTTP que incluye la URL que apunta a este archivo. Debido a que el método del formulario se ha declarado como GET, el nombre del campo de entrada y el valor insertado por el usuario se agregan a la URL, tal como se muestra a continuación.

```
<body>
  <section>
    <form name="formulario" method="get" action="procesar.php">
      <p><input type="text" name="val"></p>
      <p><input type="submit" value="Enviar"></p>
    </form>
  </section>
</body>
```

# Desarrollo Web



GOBIERNO DE LA  
CIUDAD DE MÉXICO

Cuando la información se envía con el método GET, los pares nombre/valor se agregan al final de la URL separados por el carácter =, y el primer par es precedido por el carácter ?. Si existe más de un par de valores, los restantes se agregan a la URL separados por el carácter &, como en [www.ejemplo.com/procesar.php?val1=10&val2=20](http://www.ejemplo.com/procesar.php?val1=10&val2=20).

The screenshot shows two browser tabs. The top tab is titled "Envío de Formulario" and has the URL "127.0.0.1:5500/Practica12/index.html". It contains a text input field with the value "10" and a button labeled "Enviar". The bottom tab is titled "Error" and has the URL "127.0.0.1:5500/Practica12/procesar.php?val=10". The error message "Cannot GET /Practica12/procesar.php" is displayed.

209



Al otro lado, el servidor recibe esta solicitud, lee la URL, extrae los valores y ejecuta el código en el archivo procesar.php. Este código debe procesar la información recibida y producir una respuesta. La forma en la que se realiza esta tarea depende del lenguaje de programación que utilizamos y lo que queremos lograr. Por ejemplo, para leer los valores enviados con el método GET, PHP los ofrece en un listado llamado `$_GET`. La sintaxis requerida para obtener el valor incluye su nombre entre corchetes.

```
<?php  
    print('El valor es: '.$_GET['val']);  
?>
```



Las etiquetas `<?php` y `?>` indican al servidor que este es código PHP y que tiene que ser ejecutado como tal. El código puede ser extenso o estar compuesto por solo unas pocas instrucciones, dependiendo de lo que necesitamos.

Nuestro ejemplo incluye una sola instrucción para ilustrar cómo se procesa la información. Esta instrucción, llamada `print()`, toma los valores entre paréntesis y los incluye en un archivo que se va a devolver al navegador como respuesta. En este caso, agregamos el valor recibido por medio de la solicitud al texto "El valor es: ". Si enviamos el valor 10, el servidor genera un archivo con el texto "El valor es: 10" y lo envía de regreso al navegador.



El código PHP de nuestro ejemplo utiliza el nombre `$_GET` para capturar la información recibida desde el navegador porque el método del formulario se ha declarado como `GET`, pero si cambiamos el método a `POST`, debemos utilizar el nombre `$_POST`. El siguiente ejemplo inserta código PHP dentro de un elemento `<p>`.

Cuando el servidor abre el archivo procesar.php con este documento, ejecuta el código PHP, inserta el resultado dentro de las etiquetas `<p>`, y devuelve el archivo al servidor. El resultado es el mismo que si hubiéramos creado un documento estático con el texto "El valor es: 10", pero el texto se genera dinámicamente en el servidor con los valores recibidos desde el formulario.

# Desarrollo Web



GOBIERNO DE LA  
CIUDAD DE MÉXICO

Practica12  
index.html  
procesar.php

```
<body>
  <section>
    <form name="formulario" method="get" action="procesar.php">
      <p><input type="text" name="val"></p>
      <p><input type="submit" value="Enviar"></p>
    </form>
  </section>
</body>

<body>
  <section>
    <p>
      <?php
      print('El valor es: '.$_GET['val']);
      ?>
    </p>
  </section>
```

213



## Atributos globales

HTML define atributos globales que son exclusivos de elementos de formulario. Los siguientes son los más utilizados.

**disabled**—Este es un atributo booleano que desactiva el elemento. Cuando el atributo está presente, el usuario no puede introducir valores o interactuar con el elemento.

**readonly**—Este atributo indica que el valor del elemento no se puede modificar.

**placeholder**—Este atributo muestra un texto en el fondo del elemento que indica al usuario el valor que debe introducir.



**autocomplete**—Este atributo activa o desactiva la función de autocompletar. Los valores disponibles son **on** y **off**.

**novalidate**—Este es un atributo booleano para el elemento `<form>` que indica que el formulario no debería ser validado.

**formnovalidate**—Este es un atributo booleano para los elementos `<button>` e `<input>` de tipo **submit** e **image** que indica que el formulario al que pertenecen no debería ser validado.



**required**—Este es un atributo booleano que indica al navegador que el usuario debe seleccionar o insertar un valor en el elemento para validar el formulario.

**multiple**—Este es un atributo booleano que indica al navegador que se pueden insertar múltiples valores en el campo (se aplica a elementos `<input>` de tipo **email** y **file**).

**autofocus**—Este atributo booleano solicita al navegador que mueva el foco al elemento tan pronto como se carga el documento.



**pattern**—Este atributo define una expresión regular que el navegador debe usar para validar el valor insertado en el campo.

**form**—Este atributo asocia el elemento con un formulario. Se usa para conectar un elemento con un formulario cuando el elemento no se define entre las etiquetas **<form>**. El valor asignado a este atributo debe ser el mismo asignado al atributo **id** del elemento **<form>**.

**spellcheck**—Este atributo solicita al navegador que compruebe la ortografía y gramática del valor introducido en el campo. Los valores disponibles son **true** (verdadero) y **false** (falso).



Los atributos **disabled** y **readonly** tienen un propósito similar, no permitir al usuario interactuar con el elemento, pero se aplican en diferentes circunstancias. El atributo **disabled** normalmente se implementa cuando queremos mostrar al usuario que el elemento puede estar disponible en otras condiciones, como cuando el control no es aplicable en el país del usuario, por ejemplo. Por otro lado, el atributo **readonly** se implementa cuando solo existe un valor posible y no queremos que el usuario lo cambie. Por ejemplo, en el siguiente formulario, el usuario no puede introducir la edad.

# Desarrollo Web



GOBIERNO DE LA  
CIUDAD DE MÉXICO

```
<body>
  <section>
    <form name="formulario" method="get" action="procesar.php">
      <p><label>Nombre: <input type="text" name="nombre"></label></p>
      <p><label>Edad: <input type="text" name="edad" disabled></label></p>
      <p><input type="submit" value="Enviar"></p>
    </form>
  </section>
</body>
```

The screenshot shows a web browser window with the following details:

- Title Bar:** (493) 3:30 AM Coding Session - REPRODUCIENDO | Formularios | X
- Address Bar:** 127.0.0.1:5500/Practic
- Content Area:**
  - Nombre:
  - Edad:
  -

219



Otro atributo que afecta la apariencia de un elemento es **placeholder**. Este se usa en campos de entrada para ofrecer una pista (una palabra o frase) que ayude al usuario a introducir el valor correcto. El siguiente ejemplo inserta esta ayuda en un campo de búsqueda.

El valor de este atributo lo muestran los navegadores dentro del campo hasta que el usuario inserta un valor.

# Desarrollo Web



GOBIERNO DE LA  
CIUDAD DE MÉXICO

```
<body>
  <section>
    <form name="formulario" method="get" action="procesar.php">
      <p><label>Buscar: <input type="search" name="buscar" placeholder="Término a buscar"></label></p>
      <p><input type="submit" value="Buscar"></p>
    </form>
  </section>
</body>
```

The screenshot shows a web browser window with the following details:

- Top Bar:** A red notification bar with a speaker icon and the text "(493) 3:30 AM Coding Session REPRODUCIENDO".
- Title Bar:** The title "Formularios" is visible.
- Address Bar:** The URL "127.0.0.1:5500/Practica" is displayed.
- Content Area:** A search form with the following elements:
  - A label "Buscar:" followed by a placeholder "Término a buscar".
  - A "Buscar" button.

221



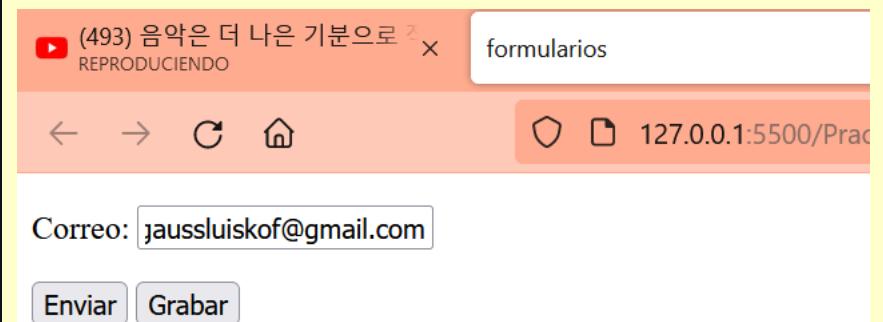
# Desarrollo Web

Una de las características de los formularios es que tienen la capacidad de validar los datos introducidos. Por defecto, los formularios validan los datos a menos que el atributo **novalidate** sea declarado. Este atributo booleano es específico de elementos **<form>**. Cuando es incluido, el formulario se envía sin validar. La presencia de este atributo afecta al formulario de forma permanente, pero a veces el proceso de validación es requerido solo en ciertas circunstancias. Por ejemplo, cuando la información insertada debe ser grabada para permitir al usuario continuar con el trabajo más adelante. En casos como este, podemos implementar el atributo **formnovalidate**.



Este atributo está disponible para los elementos que crean los botones para enviar el formulario. Cuando los datos se envían con un botón que contiene este atributo, el formulario no es validado.

```
<body>
  <section>
    <form name="formulario" method="get" action="procesar.php">
      <p><label>Correo: <input type="email" name="correo"></label></p>
      <p>
        <input type="submit" value="Enviar">
        <input type="submit" value="Grabar" formnovalidate>
      </p>
    </form>
  </section>
```





En el ejemplo el formulario será validado en circunstancias normales, pero incluimos un segundo botón con el atributo **formnovalidate** para poder enviar el formulario sin pasar por el proceso de validación. El botón **Enviar** requiere que el usuario introduzca una cuenta de correo válida, pero el botón **Grabar** no incluye este requisito.

Cuando usamos el tipo **email** para recibir una cuenta de correo, como en el ejemplo anterior, el navegador controla si el valor introducido es una cuenta de correo, pero valida el campo cuando se encuentra vacío. Esto se debe a que el campo no es obligatorio.



HTML ofrece el atributo **required** para cambiar esta condición. Cuando se incluye el atributo **required**, el campo solo será válido si el usuario introduce un valor y este valor cumple con los requisitos de su tipo. El siguiente ejemplo implementa este atributo para forzar al usuario a introducir un correo electrónico.

```
<body>
  <section>
    <form name="formulario" method="get" action="procesar.php">
      <p><label>Correo: <input type="email" name="correo" required></label></p>
      <p>
        <input type="submit" value="Enviar">
        <input type="submit" value="Grabar" formnovalidate>
      </p>
    </form>
  </section>
</body>
```

# Desarrollo Web



GOBIERNO DE LA  
CIUDAD DE MÉXICO

A screenshot of a web browser showing a contact form. The 'Correo' field contains 'lif'. The 'Enviar' button has a tooltip that says: 'Por favor, introduce una dirección de correo electrónico.'

Pulse el botón enviar. Debería recibir un mensaje de error en el campo **correo** indicando que debe insertar un valor. Pulse el botón **Grabar**. Como este botón incluye el atributo **formnovalidate**, el error ya no se muestra y se envía el formulario.

226



Otro atributo que se usa para validación es **pattern**. Algunos tipos de campos de entrada validan cadenas de caracteres específicas, pero no pueden hacer nada cuando el valor no es estándar, como en el caso de los códigos postales. No existe un tipo de campo predeterminado para esta clase de valores. El atributo **pattern** nos permite crear un filtro personalizado usando expresiones regulares.

Las expresiones regulares son textos compuestos por una serie de caracteres que definen un patrón de concordancia. Por ejemplo, los caracteres 0-9 determinan que solo se aceptan los números entre 0 y 9.



Utilizando esta clase de expresiones, podemos crear un filtro personalizado para validar cualquier valor que necesitemos. El siguiente formulario incluye un campo de entrada que solo acepta números con cinco dígitos.

```
<body>
  <section>
    <form name="formulario" method="get" action="procesar.php">
      <p><label>Código Postal: <input pattern="[0-9]{5}" name="cp" title="Inserte su código postal"></label></p>
      <p><input type="submit" value="Enviar"></p>
    </form>
  </section>
</body>
```

Formularios +

Código Postal:

Enviar

Por favor, coincide con el formato: Inserte su código postal.



El elemento **<input>** en el ejemplo incluye el atributo **pattern** con el valor **[0-9]{5}**. Esta expresión regular determina que el valor debe tener exactamente cinco caracteres y que esos caracteres deben ser números entre 0 y 9. En consecuencia, solo podemos insertar números de cinco dígitos; cualquier otro carácter o tamaño devolverá un error.

Estos tipos de entrada también pueden incluir el atributo **title** para explicar al usuario cuál es el valor esperado. Este mensaje complementa el mensaje de error estándar que muestra el navegador



Otro atributo que puede ayudar al usuario a decidir qué insertar es **autofocus**. En este caso, el atributo establece el foco en el elemento cuando se carga el documento, sugiriendo al usuario qué valor insertar primero. El siguiente ejemplo incluye dos campos de entrada para insertar el nombre y la edad del usuario, pero el campo para la edad incluye el atributo autofocus y, por lo tanto, el navegador posicionará el cursor en este campo por defecto.

# Desarrollo Web



GOBIERNO DE LA  
CIUDAD DE MÉXICO

```
<body>
  <section>
    <form name="formulario" method="get" action="procesar.php">
      <p><label>Nombre: <input type="text" name="nombre"></label></p>
      <p><label>Edad: <input type="text" name="edad" autofocus></label></p>
      <p><input type="submit" value="Enviar"></p>
    </form>
  </section>
</body>
```

The screenshot shows a web browser window with two tabs: 'Formularios' and 'formularios'. The active tab is 'formularios'. The page content includes a form with fields for 'Nombre' (containing 'LUIS') and 'Edad' (an empty input field), and a 'Enviar' (Send) button.

Formularios

formularios

← → C ⌂ 127.0.0.1:5500/Practica

Nombre: LUIS

Edad:

Enviar

231