



Etiquetas en HTML

Etiqueta `` en HTML

Concepto de ``

La etiqueta [img en HTML](#) se utiliza para insertar imágenes en una página web. Es una de las herramientas fundamentales para añadir contenido visual y mejorar la estética y la experiencia del usuario en un sitio web.

Características de la Etiqueta ``

1. **Elemento Vacío:** La etiqueta `` no tiene etiqueta de cierre; es un elemento vacío.
2. **Atributos Requeridos:**
 - **src (source):** Indica la ruta de la imagen que se va a mostrar. Puede ser una ruta relativa o absoluta.
 - **alt (alternative text):** Proporciona un texto alternativo para la imagen si no se puede mostrar. Es crucial para la accesibilidad y el SEO.
3. **Atributos Opcionales:**
 - **title:** Ofrece información adicional sobre la imagen cuando el usuario pasa el cursor sobre ella.
 - **width y height:** Especifican el ancho y el alto de la imagen en píxeles.

Ejemplo de Uso

```
Unset

```

Importancia del Atributo `alt`

- **Accesibilidad:** Permite a los lectores de pantalla describir imágenes a usuarios con discapacidad visual.
- **SEO:** Mejora la indexación de la imagen en motores de búsqueda.

- **Funcionalidad:** Muestra un texto descriptivo si la imagen no se carga.

Buenas Prácticas

- **Calidad y Tamaño de Imagen:** Usa imágenes de alta calidad pero optimizadas para no afectar negativamente los tiempos de carga.
- **Texto Alternativo Descriptivo:** El atributo `alt` debe describir con precisión el contenido de la imagen.
- **Uso Responsable de Imágenes:** Evita el uso excesivo de imágenes que puedan distraer o abrumar al usuario.

Responsividad de Imágenes con la Etiqueta `` en HTML

Concepto de Imágenes Responsivas

Las imágenes responsivas se adaptan a diferentes tamaños de pantalla y resoluciones, garantizando una buena experiencia de usuario en dispositivos como smartphones, tabletas y computadoras de escritorio. La meta es que las imágenes se vean bien y carguen eficientemente en todos los dispositivos.

Implementación de Imágenes Responsivas

1. Uso de CSS:

- **Porcentaje de Ancho:** Establecer el `width` de la imagen en un porcentaje (comúnmente `width: 100%;`) y el `height` en `auto` para mantener la proporción de la imagen.

Unset

```
img {  
  max-width: 100%;  
  height: auto;  
}
```

- **Media Queries:** Utilizar media queries en CSS para aplicar diferentes estilos a las imágenes según el tamaño de la pantalla. Este tema lo veremos en profundidad más adelante.

2. Atributo `srcset` en ``:

- Permite especificar múltiples versiones de una imagen para diferentes tamaños de pantalla.

- El navegador elige automáticamente la imagen más adecuada según la resolución de pantalla del dispositivo.

Unset

```

```

- **sizes:** Define un conjunto de condiciones de medios y anchos de imagen para ayudar al navegador a seleccionar la imagen adecuada.

3. Atributo **loading**:

- El atributo **loading="lazy"** permite la carga diferida (lazy loading) de imágenes, mejorando la velocidad de carga de la página.

Unset

```

```

Importancia de la Responsividad

- **Mejora la Experiencia del Usuario:** Las imágenes responsivas se adaptan a la pantalla del usuario, evitando problemas de sobreescala o resoluciones inapropiadas.
- **Optimización del Rendimiento:** Al cargar versiones de imagen adecuadas para el dispositivo del usuario, se ahorra ancho de banda y se mejora la velocidad de carga.
- **SEO Mejorado:** Los sitios web con imágenes responsivas son más amigables para los motores de búsqueda, especialmente para búsquedas realizadas en dispositivos móviles.

Buenas Prácticas

- **Pruebas en Diversos Dispositivos:** Asegúrate de probar cómo se ven tus imágenes en diferentes dispositivos y orientaciones.
- **Optimización de Imágenes:** Utiliza imágenes optimizadas para reducir el tiempo de carga sin sacrificar la calidad.
- **Alternativas de Alta Resolución:** Proporciona imágenes de alta resolución para pantallas con alta densidad de píxeles.

Al implementar imágenes responsivas en tu sitio web, garantizas que los usuarios disfruten de una experiencia visual óptima, independientemente del dispositivo que utilicen.

Etiqueta **<form>** en HTML

Concepto de **<form>**

La etiqueta [form en HTML](#) es utilizada para crear un formulario en una página web.

Los formularios son esas famosas cajas de texto y botones que podemos encontrar en muchas páginas web. Son muy utilizados para realizar búsquedas o bien para introducir datos personales por ejemplo en sitios de comercio electrónico. Los datos que el usuario introduce en estos campos son enviados al correo electrónico del administrador del formulario o bien a un programa que se encarga de procesarlo automáticamente.

Capacidades y Limitaciones de los Formularios en HTML

Los formularios creados con HTML son herramientas fundamentales para recopilar datos de los usuarios en una página web. Con HTML, puedes estructurar un formulario con una variedad de campos: cajas de texto, botones de radio, menús desplegables, entre otros, diseñados para capturar diferentes tipos de información.

Sin embargo, las capacidades de HTML en lo que respecta a los formularios tienen ciertas limitaciones. Aunque puedes diseñar el formulario y recoger los datos, HTML por sí solo no puede procesar esta información. Por ejemplo, no es posible enviar directamente los datos del formulario a un correo electrónico o almacenarlos en una base de datos solo con HTML.

Para estas tareas, es necesario combinar HTML con un lenguaje de programación del lado del servidor, conocido como backend. Lenguajes como Java, PHP, Python o Node.js son comúnmente utilizados para manejar y procesar los datos recopilados a través de formularios HTML. Estos lenguajes permiten no solo el envío de datos a un correo electrónico sino también su almacenamiento en bases de datos, validación de la información, y más.

En resumen, mientras que HTML es eficaz para crear la estructura y recoger datos de los formularios, para gestionar y utilizar efectivamente esta información se requiere de la integración con tecnologías del lado del servidor.

Estructura y Atributos Básicos de **<form>**

- **action:** La URL del servidor donde se enviarán los datos del formulario.
- **method:** El método HTTP para enviar los datos. Los más comunes son **GET** (datos enviados en la URL) y **POST** (datos enviados en el cuerpo de la solicitud).

- **enctype:** Especifica cómo se codifican los datos del formulario al enviarlos. Importante cuando el formulario incluye archivos para cargar.

Ejemplo de Uso Básico

Unset

```
<form action="/submit-form" method="post">
  <!-- Campos del formulario aquí -->
</form>
```

Campos Comunes en un Formulario

- **<input>**: Campo de texto, contraseña, radio, checkbox, etc.
- **<textarea>**: Área de texto para entradas más largas.
- **<select>**: Menú desplegable con opciones.
- **<button>**: Botón para enviar o controlar elementos del formulario.
- **<label>**: Etiqueta para mejorar la accesibilidad y la usabilidad de los campos del formulario.
- **fieldset**: Permite organizar en grupos los campos de un formulario.

Manejo de Datos del Formulario

- **En el lado del servidor:** Los datos enviados desde un formulario se procesan en el servidor. Dependiendo del lenguaje de programación y del framework, los datos se manejan de diferentes maneras.
- **Validación:** Es importante validar los datos tanto en el lado del cliente (para la experiencia del usuario) como en el servidor (por seguridad).

Ejemplo de Formulario Completo

Unset

```
<form action="/submit-form" method="post">
  <label for="name">Nombre:</label>
  <input type="text" id="name" name="name" />

  <label for="email">Email:</label>
  <input type="email" id="email" name="email" />

  <label for="message">Mensaje:</label>
  <textarea id="message" name="message"></textarea>
```

```

<fieldset>
  <legend>Edad</legend>
  <div class="fieldset">
    <input type="checkbox" tabindex="20" name="edad" value="20-39" />
    20-39
  </div>
  <div class="fieldset">
    <input type="checkbox" tabindex="21" name="edad" value="40-59" />
    40-59
  </div>
  <div class="fieldset">
    <input type="checkbox" tabindex="22" name="edad" value="60-79" />
    60-79
  </div>
</fieldset>

<button type="submit">Enviar</button>
</form>

```

Resultado

Nombre:

Email:

Mensaje:

Edad

☐ 20-39
☐ 40-59
☐ 60-79

Enviar

Consideraciones de Accesibilidad


- Usa la etiqueta `<label>` con el atributo `for` correspondiente al `id` del campo. Esto mejora la accesibilidad para lectores de pantalla y facilita la interacción para los usuarios.

- Asegúrate de que todos los campos tengan etiquetas claras y que los campos requeridos estén marcados.

Buenas Prácticas

- Mantén los formularios lo más simples y claros posible para no abrumar al usuario.
- Utiliza la validación en el lado del cliente para retroalimentación inmediata, pero siempre valida en el servidor como medida de seguridad.
- Asegúrate de que el formulario sea accesible y fácil de usar en dispositivos móviles.

Los formularios son una parte integral de la web, permitiendo la recolección de datos de los usuarios y la interacción dinámica. Un buen diseño y una implementación cuidadosa son clave para una buena experiencia del usuario.

 **Nota Importante:** Aunque hemos explorado en profundidad la etiqueta `<form>`, en nuestras próximas actividades nos centraremos en aplicarla de una manera más básica y fundamental. Esta aproximación inicial nos ayudará a afianzar nuestro entendimiento y prepararnos para usos más avanzados en el futuro. ¡Estén atentos a los ejemplos prácticos que se vienen!

Etiqueta `<input>` en HTML

Concepto General de `<input>`

La etiqueta [input en HTML](#) es uno de los controles de formulario más versátiles y esenciales, utilizada para recopilar entradas del usuario. Ofrece una variedad de tipos para manejar diferentes tipos de datos, desde texto hasta selecciones y rangos.

Atributos Comunes de `<input>`

- **type:** Define el tipo de control de entrada y su comportamiento (ej., `text`, `password`, `radio`, `checkbox`, `submit`, etc.).
- **name:** Nombre del control que se utiliza para identificar los datos del formulario cuando se envían al servidor.
- **value:** Valor predeterminado o valor del campo (especialmente relevante para botones, radio, y checkboxes).
- **placeholder:** Texto de sugerencia que se muestra en el campo hasta que el usuario comienza a escribir.
- **required:** Especifica que el campo debe ser completado antes de enviar el formulario.
- **disabled:** Desactiva el campo de entrada.

Tipos Comunes de `<input>`

- **Texto** (**type="text"**): Campo estándar para introducir texto.
- **Contraseña** (**type="password"**): Campo para contraseñas, oculta el texto ingresado.
- **Radio** (**type="radio"**): Botones de opción para seleccionar una de las opciones disponibles.
- **Checkbox** (**type="checkbox"**): Para seleccionar o deseleccionar una opción.
- **Submit** (**type="submit"**): Botón para enviar el formulario.
- **Email** (**type="email"**): Campo para introducir un correo electrónico, con validación de formato.
- **Number** (**type="number"**): Campo para introducir números, con controles para aumentar o disminuir el valor.

Ejemplos de Uso

Unset

```
<!-- Campo de texto -->
<input type="text" name="username" placeholder="Nombre de usuario"
/>

<!-- Contraseña -->
<input type="password" name="password" placeholder="Contraseña" />

<!-- Radio Button -->
<input type="radio" name="gender" value="male" /> Male
<input type="radio" name="gender" value="female" /> Female

<!-- Checkbox -->
<input type="checkbox" name="subscribe" value="newsletter" />
Suscribirse al
boletín

<!-- Botón de Envío -->
<input type="submit" value="Enviar" />
```

Validación y Seguridad

- La validación del lado del cliente se puede hacer usando atributos como **required**, **min**, **max**, y **pattern**.
- La validación del lado del servidor siempre es necesaria para la seguridad, ya que la validación del lado del cliente puede ser eludida.

Buenas Prácticas

- Usa el atributo **name** correctamente para asegurar que los datos del formulario se envíen de manera efectiva.
- Utiliza **placeholder** para proporcionar ejemplos de entrada o instrucciones breves.
- Asegúrate de que los campos **required** sean realmente esenciales para no sobrecargar al usuario.

Los campos `<input>` son fundamentales en cualquier formulario web, permitiendo recopilar una amplia gama de datos de los usuarios de manera eficiente y efectiva. Su correcta utilización y configuración son clave para una buena experiencia de usuario y una recolección de datos efectiva.

Listas en HTML

Las [listas en HTML](#) son una herramienta clave para organizar datos y contenido de manera estructurada y legible. Hay tres tipos principales de listas que se pueden crear con HTML:

1. Listas Ordenadas (``)

- **Uso:** Para una secuencia o lista de elementos donde el orden importa.
- **Estructura:**
 - Se inicia con la etiqueta `` (Ordered List).
 - Cada ítem de la lista se representa con la etiqueta `` (List Item).
- **Numeración Automática:** Los ítems en una lista ordenada se numeran automáticamente por defecto, comenzando desde 1.
- **Personalización:** La numeración puede ser personalizada usando el atributo **type** y **start** en la etiqueta ``.

2. Listas Desordenadas (``)

- **Uso:** Para listas donde el orden no es relevante.
- **Estructura:**
 - Se inicia con la etiqueta `` (Unordered List).
 - Los ítems también se marcan con ``.
- **Marcadores por Defecto:** Por lo general, los ítems en una lista desordenada se muestran con viñetas.

3. Listas de Definición (`<dl>`)

- **Uso:** Ideal para listas de términos y sus definiciones correspondientes.
- **Estructura:**

- `<dl>` (Description List) envuelve la lista completa.
- `<dt>` (Description Term) se usa para el término que se va a definir.
- `<dd>` (Description Details) se usa para la definición del término.

Ejemplos de Código

Lista Ordenada

Unset

```
<ol>
  <li>Primer Elemento</li>
  <li>Segundo Elemento</li>
  <li>Tercer Elemento</li>
</ol>
```

1. Primer Elemento
2. Segundo Elemento
3. Tercer Elemento

Lista Desordenada

Unset

```
<ul>
  <li>Manzana</li>
  <li>Banana</li>
  <li>Naranja</li>
</ul>
```

- Manzana
- Banana
- Naranja

Lista de Definición

Unset

```
<dl>
  <dt>HTML</dt>
```

```
<dd>Lenguaje de Marcado de Hipertexto</dd>  
<dt>CSS</dt>  
<dd>Hojas de Estilo en Cascada</dd>  
</dl>
```

HTML

Lenguaje de Marcado de Hipertexto

CSS

Hojas de Estilo en Cascada

Buenas Prácticas y Consideraciones

- **Legibilidad:** Las listas mejoran la legibilidad del contenido, especialmente cuando se trata de enumerar o categorizar información.
- **SEO:** Las listas son beneficiosas para el SEO, ya que los motores de búsqueda valoran la estructura y organización del contenido.
- **Estilización con CSS:** Puedes personalizar la apariencia de las listas (como el tipo de viñetas o la numeración) utilizando CSS.
- **Anidación de Listas:** Las listas pueden anidarse dentro de otras listas para crear subniveles.

Las listas en HTML son fundamentales para estructurar información de manera clara y coherente, mejorando tanto la experiencia del usuario como la eficacia del contenido en términos de SEO.