

19 EL ATRIBUTO ARIA-INVALID

El atributo `aria-invalid` se utiliza para indicar si un campo de entrada tiene un valor inválido. Este atributo es especialmente útil para formularios, ya que permite a las tecnologías de asistencia, como los lectores de pantalla, informar a los usuarios sobre errores en los campos de entrada.

EJEMPLO DE USO DE ARIA-INVALID

A continuación, te proporciono un ejemplo de cómo usar `aria-invalid` en un formulario para indicar campos con errores y cómo proporcionar una explicación de los errores.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Ejemplo de aria-invalid</title>
  <style>
    .error {
      border: 2px solid red;
    }
    .error-message {
      color: red;
      font-size: 0.9em;
    }
  </style>
</head>
<body>

<form id="exampleForm">
  <div>
    <label for="name">Nombre:</label>
    <input type="text" id="name" name="name" aria-
required="true" aria-invalid="false">
    <span id="nameError" class="error-message" aria-
live="polite"></span>
  </div>
  <div>
    <label for="email">Email:</label>
    <input type="email" id="email" name="email" aria-
required="true" aria-invalid="false">
    <span id="emailError" class="error-message" aria-
live="polite"></span>
  </div>
  <div>
    <label for="password">Contraseña:</label>
    <input type="password" id="password" name="password"
aria-required="true" aria-invalid="false">
    <span id="passwordError" class="error-message" aria-
live="polite"></span>
  </div>
  <button type="submit">Enviar</button>
```

```
</form>

<script>

document.getElementById('exampleForm').addEventListener('submit',
function(event) {
    event.preventDefault();
    let isValid = true;

    // Validar nombre
    const nameInput = document.getElementById('name');
    const nameError = document.getElementById('nameError');
    if (nameInput.value.trim() === '') {
        nameInput.setAttribute('aria-invalid', 'true');
        nameInput.classList.add('error');
        nameError.textContent = 'El nombre es requerido.';
        isValid = false;
    } else {
        nameInput.setAttribute('aria-invalid', 'false');
        nameInput.classList.remove('error');
        nameError.textContent = '';
    }

    // Validar email
    const emailInput = document.getElementById('email');
    const emailError = document.getElementById('emailError');
    if (emailInput.value.trim() === '' ||
!emailInput.validity.valid) {
        emailInput.setAttribute('aria-invalid', 'true');
        emailInput.classList.add('error');
        emailError.textContent = 'El email es requerido y
debe ser válido.';
        isValid = false;
    } else {
        emailInput.setAttribute('aria-invalid', 'false');
        emailInput.classList.remove('error');
        emailError.textContent = '';
    }

    // Validar contraseña
    const passwordInput =
document.getElementById('password');
    const passwordError =
document.getElementById('passwordError');
    if (passwordInput.value.trim() === '') {
        passwordInput.setAttribute('aria-invalid', 'true');
        passwordInput.classList.add('error');
        passwordError.textContent = 'La contraseña es
requerida.';
        isValid = false;
    } else {
        passwordInput.setAttribute('aria-invalid', 'false');
        passwordInput.classList.remove('error');
        passwordError.textContent = '';
    }
}
```

```

        if (isValid) {
            alert('Formulario enviado con éxito');
        }
    });
</script>

</body>
</html>

```

EXPLICACIÓN DEL CÓDIGO

1. HTML:

- **Formulario:** El formulario contiene tres campos de entrada: nombre, email y contraseña. Cada campo tiene un atributo `aria-required="true"` para indicar que es obligatorio y un atributo `aria-invalid` que se actualizará dinámicamente.
- **Mensajes de Error:** Cada campo de entrada tiene un elemento `` asociado para mostrar mensajes de error. El atributo `aria-live="polite"` en estos elementos asegura que los lectores de pantalla anuncien los mensajes de error cuando se actualicen.

2. CSS:

- **Estilos de Error:** Se definen estilos para los campos de entrada con errores (`error`) y para los mensajes de error (`error-message`).

3. JavaScript:

- **Validación del Formulario:** Se añade un evento `submit` al formulario para validar los campos de entrada. Si un campo es inválido, se actualiza el atributo `aria-invalid` a `true`, se añade la clase `error` al campo y se muestra un mensaje de error. Si el campo es válido, se actualiza el atributo `aria-invalid` a `false`, se elimina la clase `error` y se oculta el mensaje de error.
- **Notificación de Envío:** Si todos los campos son válidos, se muestra una alerta indicando que el formulario ha sido enviado con éxito.

EXPLICACIÓN PASO A PASO DEL JAVASCRIPT

1. Añadir un Evento de Enviar (Submit) al Formulario:

```
document.getElementById('exampleForm').addEventListener('submit',
function(event) {
```

- Esto añade un evento de envío (`submit`) al formulario con el ID `exampleForm`. Cada vez que se intente enviar el formulario, se ejecutará la función especificada.

2. Prevenir el Envío por Defecto del Formulario:

```
event.preventDefault();
```

- `event.preventDefault()` previene el envío por defecto del formulario, lo que permite realizar validaciones antes de enviar los datos.

3. Inicializar una Variable de Estado de Validación:

```
let isValid = true;
```

- Se inicializa una variable isValid a true. Esta variable se utilizará para rastrear si todos los campos del formulario son válidos.

4. Validar el Campo de Nombre:

```
const nameInput = document.getElementById('name');
const nameError = document.getElementById('nameError');
if (nameInput.value.trim() === '') {
  nameInput.setAttribute('aria-invalid', 'true');
  nameInput.classList.add('error');
  nameError.textContent = 'El nombre es requerido.';
  isValid = false;
} else {
  nameInput.setAttribute('aria-invalid', 'false');
  nameInput.classList.remove('error');
  nameError.textContent = '';
}
```

- Se obtiene el campo de entrada para el nombre y el elemento para mostrar el mensaje de error.
- Si el campo de nombre está vacío (nameInput.value.trim() === ''), se actualiza el atributo aria-invalid a true, se añade la clase error para estilizar el campo como inválido y se muestra un mensaje de error. También se actualiza isValid a false.
- Si el campo de nombre no está vacío, se actualiza el atributo aria-invalid a false, se elimina la clase error y se oculta el mensaje de error.

5. Validar el Campo de Email:

```
const emailInput = document.getElementById('email');
const emailError = document.getElementById('emailError');
if (emailInput.value.trim() === '' || !emailInput.validity.valid) {
  emailInput.setAttribute('aria-invalid', 'true');
  emailInput.classList.add('error');
  emailError.textContent = 'El email es requerido y debe ser válido.';
  isValid = false;
} else {
  emailInput.setAttribute('aria-invalid', 'false');
  emailInput.classList.remove('error');
  emailError.textContent = '';
}
```

- Se obtiene el campo de entrada para el email y el elemento para mostrar el mensaje de error.
- Si el campo de email está vacío o no es válido (!emailInput.validity.valid), se actualiza el atributo aria-invalid a true, se añade la clase error y se muestra un mensaje de error. También se actualiza isValid a false.
- Si el campo de email no está vacío y es válido, se actualiza el atributo aria-invalid a false, se elimina la clase error y se oculta el mensaje de error.

6. Validar el Campo de Contraseña:

```
const passwordInput = document.getElementById('password');
const passwordError = document.getElementById('passwordError');
if (passwordInput.value.trim() === '') {
    passwordInput.setAttribute('aria-invalid', 'true');
    passwordInput.classList.add('error');
    passwordError.textContent = 'La contraseña es requerida.';
    isValid = false;
} else {
    passwordInput.setAttribute('aria-invalid', 'false');
    passwordInput.classList.remove('error');
    passwordError.textContent = '';
}
```

- Se obtiene el campo de entrada para la contraseña y el elemento para mostrar el mensaje de error.
- Si el campo de contraseña está vacío, se actualiza el atributo aria-invalid a true, se añade la clase error y se muestra un mensaje de error. También se actualiza isValid a false.
- Si el campo de contraseña no está vacío, se actualiza el atributo aria-invalid a false, se elimina la clase error y se oculta el mensaje de error.

7. Enviar el Formulario si es Válido:

```
if (isValid) {
    alert('Formulario enviado con éxito');
}
```

- Si todos los campos son válidos (isValid es true), se muestra una alerta indicando que el formulario ha sido enviado con éxito.

RESUMEN

El código JavaScript realiza las siguientes tareas:

1. Previene el envío por defecto del formulario.
2. Valida cada campo de entrada (nombre, email y contraseña).
3. Actualiza los atributos ARIA (aria-invalid) y los mensajes de error en función de la validez de cada campo.
4. Si todos los campos son válidos, muestra una alerta indicando que el formulario ha sido enviado con éxito.

Este enfoque asegura que los usuarios reciban retroalimentación clara sobre los errores en los campos del formulario, mejorando la accesibilidad y la usabilidad del formulario.