05 USO DE VARIOS ATRIBUTOS **ARIA**

Ejemplo en HTML y JavaScript que demuestra el uso de varios atributos **ARIA** para mejorar la accesibilidad de un componente interactivo: un botón que expande y contrae un panel de información. Este ejemplo usa los siguientes atributos:

- **aria-label**: Proporciona una etiqueta accesible para un elemento sin texto visible.
- **aria-labelledby**: Asocia el elemento con otro que contiene su etiqueta accesible.
- **aria-describedby**: Vincula el elemento con una descripción adicional.
- **aria-expanded**: Indica si un elemento colapsable está expandido o contraído.
- **aria-hidden**: Oculta elementos del lector de pantalla cuando no son relevantes.

Código de ejemplo:

```
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Ejemplo de ARIA</title>
    <style>
        .hidden {
            display: none;
        }
    </style>
</head>
<body>

<button id="toggleButton" aria-expanded="false" aria-controls="hiddenContent">
    Mostrar/Ocultar Contenido
</button>

<div id="hiddenContent" class="hidden" aria-hidden="true">
    <p id="description">Este es un contenido oculto que se puede mostrar o ocultar.</p>
    <button aria-label="Cerrar contenido" onclick="toggleContent()">Cerrar</button>
</div>
```

```
<script>
    const toggleButton = document.getElementById('toggleButton');
    const hiddenContent =
document.getElementById('hiddenContent');

    toggleButton.addEventListener('click', () => {
        const isExpanded = toggleButton.getAttribute('aria-
expanded') === 'true';
        toggleButton.setAttribute('aria-expanded', !isExpanded);
        hiddenContent.setAttribute('aria-hidden', isExpanded);
        hiddenContent.classList.toggle('hidden', isExpanded);
    });

    function toggleContent() {
        toggleButton.click();
    }
</script>

</body>
</html>
```

EXPLICACIÓN DEL CÓDIGO:

1. **aria-label:**
    - Uso: Proporciona una etiqueta de texto para un elemento que no tiene texto visible.
    - Ejemplo: <button aria-label="Cerrar contenido" onclick="toggleContent()">Cerrar</button>
    - Práctica: Utilízalo cuando un elemento no tiene texto visible, pero necesita una descripción para los lectores de pantalla.
2. **aria-labelledby:**
    - Uso: Asocia un elemento con otro elemento que actúa como su etiqueta.
    - Ejemplo: No utilizado en este ejemplo, pero se podría usar así: ...
    - Práctica: Úsalo cuando un elemento necesita una etiqueta que ya existe en otro lugar del documento.
3. **aria-describedby:**
    - Uso: Proporciona una descripción adicional para un elemento.

- Ejemplo: No utilizado en este ejemplo, pero se podría usar así: <button aria-describedby="description">Mostrar/Ocultar Contenido</button>
- Práctica: Úsalo para proporcionar información adicional que complementa la etiqueta principal.

4. **aria-expanded:**
    - Uso: Indica si un elemento, o un grupo de elementos, está expandido o colapsado.
    - Ejemplo: <button id="toggleButton" aria-expanded="false" aria-controls="hiddenContent">Mostrar/Ocultar Contenido</button>
    - Práctica: Úsalo en elementos que controlan la visibilidad de otros elementos, como botones de acordeón o menús desplegables.
5. **aria-hidden:**
    - Uso: Oculta un elemento y su contenido de la tecnología de asistencia.
    - Ejemplo: ...
    - Práctica: Úsalo para ocultar elementos que no deben ser leídos por los lectores de pantalla, como contenido temporalmente oculto.
6. **aria-controls:**
    - Uso: Indica que un elemento controla otro elemento.
    - Ejemplo: <button id="toggleButton" aria-expanded="false" aria-controls="hiddenContent">Mostrar/Ocultar Contenido</button>
    - Práctica: Úsalo para establecer una relación de control entre dos elementos, como un botón que controla un panel desplegable.

Prácticas Recomendadas

- Usa atributos ARIA solo cuando sea necesario: No añadas atributos ARIA a elementos que ya son accesibles por defecto.
- Mantén la coherencia: Asegúrate de que los atributos ARIA sean coherentes con el contenido visible y la funcionalidad del elemento.
- Actualiza dinámicamente: Si un elemento cambia su estado (por ejemplo, de oculto a visible), asegúrate de actualizar los atributos ARIA correspondientes.
- Prueba con tecnologías de asistencia: Verifica que tu implementación funcione correctamente con lectores de pantalla y otras tecnologías de asistencia.

ELEMENTOS QUE SON ACCESIBLES POR DEFECTO

En HTML, hay varios elementos que son accesibles por defecto, lo que significa que los navegadores y las tecnologías de asistencia pueden interpretarlos y proporcionar una experiencia accesible sin necesidad de atributos ARIA adicionales. Aquí tienes una lista de algunos de estos elementos y una breve explicación de por qué son accesibles por defecto:

1. <a> (Ancla):
    - Por qué es accesible: Los enlaces son naturalmente accesibles ya que los navegadores y los lectores de pantalla pueden identificarlos y anunciarlos como enlaces.
    - Ejemplo: Ir a Ejemplo
2. <button> (Botón):
    - Por qué es accesible: Los botones son interactivos y los navegadores y los lectores de pantalla pueden identificarlos y anunciarlos como botones.
    - Ejemplo: <button>Enviar</button>
3. <input> (Entrada):
    - Por qué es accesible: Los campos de entrada son naturalmente accesibles y los navegadores y los lectores de pantalla pueden identificarlos y anunciarlos como campos de entrada.
    - Ejemplo: <input type="text" name="nombre" id="nombre">
4. <label> (Etiqueta):
    - Por qué es accesible: Las etiquetas están asociadas con campos de entrada y proporcionan una descripción accesible para ellos.
    - Ejemplo: <label for="nombre">Nombre:</label>
5. <select> (Selección):
    - Por qué es accesible: Los menús desplegables son naturalmente accesibles y los navegadores y los lectores de pantalla pueden identificarlos y anunciarlos como menús desplegables.
    - Ejemplo: <select name="opciones" id="opciones"><option value="1">Opción 1</option><option value="2">Opción 2</option></select>
6. <textarea> (Área de texto):
    - Por qué es accesible: Las áreas de texto son naturalmente accesibles y los navegadores y los lectores de pantalla pueden identificarlas y anunciarlas como áreas de texto.
    - Ejemplo: <textarea name="comentarios" id="comentarios"></textarea>
7. <form> (Formulario):
    - Por qué es accesible: Los formularios son naturalmente accesibles y los navegadores y los lectores de pantalla pueden identificarlos y anunciarlos como formularios.
    - Ejemplo: <form action="/submit" method="post"><label for="nombre">Nombre:</label><input type="text" name="nombre" id="nombre"><button type="submit">Enviar</button></form>
8. <img> (Imagen):

- Por qué es accesible: Las imágenes pueden ser accesibles si se proporciona un atributo alt descriptivo.

- Ejemplo: <img src="imagen.jpg" alt="Descripción de la imagen">

9. , <ol>, <li> (Listas):

    - Por qué son accesibles: Las listas y los elementos de lista son naturalmente accesibles y los navegadores y los lectores de pantalla pueden identificarlos y anunciarlos como listas y elementos de lista.

    - Ejemplo: Elemento 1
Elemento 2

10. <table>, <tr>, <td>, <th> (Tablas):

    - Por qué son accesibles: Las tablas y sus elementos son naturalmente accesibles y los navegadores y los lectores de pantalla pueden identificarlos y anunciarlos como tablas y elementos de tabla.

    - Ejemplo: <table><tr><th>Encabezado</th><td>Dato</td></tr></table>