SUBMENÚ UTILIZANDO SOLO EL TECLADO

Para abrir un submenú utilizando solo el teclado en un sitio web accesible, es importante seguir las prácticas de accesibilidad y utilizar HTML y CSS de manera adecuada. Aquí te explico cómo hacerlo paso a paso:

1. ESTRUCTURA HTML

Primero, asegúrate de que la estructura HTML del menú esté correctamente construida. Utiliza elementos semánticos como <nav>, <ul>, <li>, y <a> para crear el menú y los submenús.

```
<nav aria-label="Menú principal">
  <ul>
    <li>
      <a href="#" aria-haspopup="true" aria-expanded="false">Menú
1</a>
      <ul>
        <li><a href="#">Submenú 1.1</a></li>
        <li><a href="#">Submenú 1.2</a></li>
      </ul>
    </li>
    <li><a href="#">Menú 2</a></li>
  </ul>
</nav>
```

- aria-haspopup="true": Indica que el elemento tiene un submenú.
- aria-expanded="false": Indica que el submenú está inicialmente cerrado.

2. ESTILOS CSS

Asegúrate de que el submenú esté oculto inicialmente y se muestre solo cuando sea necesario. Puedes usar CSS para controlar la visibilidad.

```
nav ul ul {
  display: none;
}

nav ul li:hover > ul,
nav ul li:focus-within > ul {
  display: block;
}
```

- display: none;: Oculta el submenú inicialmente.
- :hover y :focus-within: Muestran el submenú cuando el usuario pasa el ratón o enfoca el elemento.

3. INTERACCIÓN CON EL TECLADO

Para que el submenú se abra con el teclado, es necesario agregar un poco de JavaScript. Esto permitirá que el submenú se abra cuando el usuario presione la tecla "Enter" o la barra espaciadora.

```
document.addEventListener('DOMContentLoaded', function() {
  const menuItems = document.querySelectorAll('nav ul li a');
```

```
  menuItems.forEach(item => {
    item.addEventListener('keydown', function(event) {
      if (event.key === 'Enter' || event.key === ' ') {
        event.preventDefault();
        const submenu = this.nextElementSibling;
        if (submenu && submenu.tagName === 'UL') {
          const isExpanded = this.getAttribute('aria-expanded')
=== 'true';
          this.setAttribute('aria-expanded', !isExpanded);
          submenu.style.display = isExpanded ? 'none' : 'block';
        }
      }
    });
  });
});
```

- keydown: Escucha el evento de teclado.
- event.key === 'Enter' || event.key === ' ': Detecta si se presionó "Enter" o la barra espaciadora.
- aria-expanded: Cambia el estado del atributo para indicar si el submenú está abierto o cerrado.
- submenu.style.display: Muestra u oculta el submenú.

4. NAVEGACIÓN CON EL TECLADO

Asegúrate de que los usuarios puedan navegar por el submenú con el teclado. Esto se puede lograr utilizando las teclas de flecha (ArrowUp, ArrowDown, ArrowLeft, ArrowRight) para moverse entre los elementos del menú y el submenú.

```
document.addEventListener('keydown', function(event) {
  const focusedElement = document.activeElement;

  if (focusedElement.tagName === 'A' &&
focusedElement.parentElement.tagName === 'LI') {
    const parentLi = focusedElement.parentElement;
    const submenu = parentLi.querySelector('ul');

    if (submenu) {
      const firstSubmenuItem = submenu.querySelector('a');
      const lastSubmenuItem = submenu.querySelector('a:last-
child');

      if (event.key === 'ArrowDown') {
        event.preventDefault();
        firstSubmenuItem.focus();
      } else if (event.key === 'ArrowUp') {
        event.preventDefault();
        lastSubmenuItem.focus();
      }
    }
  }
});
```

- ArrowDown: Mueve el foco al primer elemento del submenú.
- ArrowUp: Mueve el foco al último elemento del submenú.

5. PRUEBAS DE ACCESIBILIDAD

Finalmente, asegúrate de probar la accesibilidad del menú utilizando herramientas como WAVE o NVDA para verificar que todo funcione correctamente.

RESUMEN

- **HTML**: Utiliza elementos semánticos y atributos ARIA.
- **CSS**: Controla la visibilidad del submenú.
- **JavaScript**: Maneja la interacción con el teclado.
- **Pruebas**: Verifica la accesibilidad con herramientas especializadas.