

INDICE

INTRODUCCIÓN	1
EJERCICIOS	3
Acceso y navegación por elementos	3
Código:	3
Explicación:	3
Function recorrerAdelante	4
Código:	4
Explicación:	4
Function recorrerHermanos	4
Código:	4
Explicación:	4
Function recorrerHijos	5
Código:	5
Explicación:	5
BIBLIOGRAFIA	5

INTRODUCCIÓN

Acceso y Navegación por elementos.

Empecemos creando un fichero HTML con títulos H2 donde pongamos 4 Comunidades Autónomas de España (visualiza el siguiente enlace

http://www.ordiecole.com/voyages/esp_com_aut.html y elige una que sea del NORTE, otra del SUR, ESTE y OESTE) y debajo de cada uno de ellas, deberás añadir un con sus respectivas provincias ().

Intenta ordenar estas listas con 4 DIV cuyos id sean "norte" y "sur", "este", "oeste". El objetivo será, que cuando "pichemos" al elemento H2, podamos extraer, por un lado, el identificador del elemento DIV superior, y por otro lado, todos los elementos LI que contiene.

Nota: Suponemos que NO sabemos el nombre de los ids que tenemos en nuestro HTML, por lo que NO podremos acceder mediante los mismos.

Function recorrerAdelante:

Deberá tener un parámetro que capture la posición donde el usuario ha hecho click. Piensa en un parámetro predefinido que hemos estado utilizando en ejercicios anteriores para conseguir captar esta información.

Guarda en una variable llamada "texto" la Comunidad que ha elegido el usuario de la siguiente manera: "Has elegido " + texto del H2 seleccionado. Busca cómo podemos capturar el contenido del texto en el que el usuario ha hecho click.

Acto seguido, busca como podemos referenciar al padre de dicho elemento H2. Una vez referenciado, deberás añadir a tu variable TEXTO el siguiente mensaje: " que está situada en el " + norte/sur/ este/oeste , dependiendo del ID del DIV padre seleccionado.

Function recorrerHermanos:

Accede al hermano del H2 seleccionado. Una vez encontrado, deberemos añadir a nuestra variable TEXTO, lo siguiente: "El número de provincias es: " + Encuentra cómo podemos contabilizar el número de hijos de un elemento.

Function recorrerHijos:

Ahora vamos a ir recorriendo todos los hijos del elemento donde haya hecho click el usuario. Para ello, crea una variable donde guardemos la información del primer hijo, y simplemente mientras ese hijo sea distinto de null, añadiremos a nuestro TEXTO el nombre de cada una de las provincias.

EJERCICIOS

Acceso y navegación por elementos

Código:

```
<body>
  <div class="container">
    <div id="este">
      <h2>Cataluña</h2>
      <ul>
        <li>Barcelona</li>
        <li>Gerona</li>
        <li>Lerida</li>
        <li>Tarragona</li>
      </ul>
    </div>
    <div id="norte">
      <h2>Galicia</h2>
      <ul>
        <li>La Coruña</li>
        <li>Lugo</li>
        <li>Orense</li>
        <li>Pontevedra</li>
      </ul>
    </div>
    <div id="sur">
      <h2>Andalucía</h2>
      <ul>
        <li>Almería</li>
        <li>Cádiz</li>
        <li>Córdoba</li>
        <li>Granada</li>
        <li>Huelva</li>
        <li>Jaén</li>
        <li>Málaga</li>
        <li>Sevilla</li>
      </ul>
    </div>
    <div id="oeste">
      <h2>Extremadura</h2>
      <ul>
```

Explicación:

La estructura principal del sitio web la organizo dentro de un div contenedor ("container"). A través de bloques div con identificadores concretos ("norte", "este", etc.), establezco las secciones geográficas de España (Andalucía, Galicia, Cataluña). En cada bloque, empleo un h2 para el nombre de la región y una lista desordenada (ul) que incluye las provincias correspondientes (li), lo cual permite que el contenido esté categorizado con claridad.

Function recorrerAdelante

Código:

```
function recorrerAdelante(event) {  
    const h2Pulsado = event.currentTarget;  
    let texto = `Has elegido la comunidad autonoma de ${h2Pulsado.textContent}.`;   
    texto += "Se encuentra en el " + h2Pulsado.parentNode.id;  
    texto += " El numero de provicias es: " + recorrerHermanos(h2Pulsado)[1] + ".";  
    texto += " Las provincias son: " + recorrerHijos(recorrerHermanos(h2Pulsado)[0]) + ".";  
    const div = document.createElement("div");  
    const p = document.createElement("p");  
    p.textContent = texto;  
    div.append(p);  
    document.body.append(div);  
    console.log(texto);  
}
```

Explicación:

Cuando se hace clic en un h2, se activa esta función: `recorrerAdelante(event)`. Consigo el elemento presionado y elaboro una cadena de texto que describe la comunidad autónoma seleccionada, su localización geográfica (`parentNode.id`) y sus provincias. Para ello, empleo otras funciones de apoyo (`recorrerHijos` y `recorrerHermanos`) que me permiten obtener la información necesaria. Por último, genero un nuevo `div` que incluye el resumen completo y lo añado al cuerpo del documento. Además, muestro el texto en la consola del navegador.

Function recorrerHermanos

Código:

```
function recorrerHermanos(h2) {  
    const ul = h2.nextElementSibling;  
    const li = ul.getElementsByTagName("li");  
    return [ul, li.length];  
}
```

Explicación:

Defino la función `recorrerHermanos(h2)`, que recibe un elemento `h2` como parámetro. Utilizando `nextElementSibling`, localizo el elemento `ul` (lista desordenada) que es el hermano adyacente siguiente al `h2`. Después, obtengo la longitud de los elementos `li` (provincias) contenidos en la lista `ul` mediante `getElementsByTagName("li")`. Para concluir, retorno un array que incluye el elemento `"ul"` y la cantidad total de provincias (`li.length`).

Function recorrerHijos

Código:

```
function recorrerHijos(ul) {  
    ul.style.display = "block";  
    const li = ul.getElementsByTagName("li");  
    const liContent = Array.from(li).map((l) => l.textContent);  
    return liContent.join(", ");  
}
```

Explicación:

La función recorrerHijos(ul), que recibe un elemento ul como parámetro, la defino. En primer lugar, garantizo que la lista sea visible al establecer su estilo display en "block". Después, consigo todos los elementos li que tiene esa lista. Transformo este conjunto de li en un array y lo recorro para obtener el contenido textual de cada elemento. Al final, devuelvo una cadena de texto que reúne todos los nombres de las provincias, separados por una coma y un espacio.

RESULTADOS

Cataluña	Galicia	Andalucía	Extremadura
Barcelona Gerona Lerida Tarragona	La Coruña Lugo Orense Pontevedra	Almeria Cadiz Cordoba Granada Huelva Jaen Malaga Sevilla	Badajoz Caceres
Has elegido la comunidad autonoma de Cataluña.Se encuentra en el este El numero de provicias es: 4. Las provincias son: Barcelona, Gerona, Lerida, Tarragona.			
Has elegido la comunidad autonoma de Extremadura.Se encuentra en el oeste El numero de provicias es: 2. Las provincias son: Badajoz, Caceres.			
Has elegido la comunidad autonoma de Andalucía.Se encuentra en el sur El numero de provicias es: 8. Las provincias son: Almeria, Cadiz, Cordoba, Granada, Huelva, Jaen, Malaga, Sevilla.			
Has elegido la comunidad autonoma de Galicia.Se encuentra en el norte El numero de provicias es: 4. Las provincias son: La Coruña, Lugo, Orense, Pontevedra.			

BIBLIOGRAFIA

Uso de MDN Web Docs (Mozilla Developer Network): Para comprender cómo utilizar los métodos modernos de JavaScript, como Array.from() y map(), que me facilitan la manipulación de conjuntos de elementos del DOM, así como para corroborar las API que permiten acceder a los nodos del DOM (nextElementSibling, parentElement), reviso la documentación oficial.

Enlace: https://developer.mozilla.org/es/docs/Web/API/Document_Object_Model

Uso de W3Schools: Para fortalecer la sintaxis y el funcionamiento de los selectores que empleo para encontrar los encabezados (h2), los elementos de lista (li) y los contenedores (div), examino los tutoriales sobre Manipulación del DOM y Navegación entre Nodos.

Enlace: https://www.w3schools.com/js/js_htmlDOM.asp

Referencias para el uso avanzado de JavaScript: Es fundamental para el procesamiento y la extracción eficiente del contenido de las provincias que compruebe las funciones de las colecciones de datos y la conversión de elementos del DOM a arreglos manejables.

Enlace: <https://developer.mozilla.org/es/docs/Web/JavaScript>