

INDICE

INTRODUCCIÓN	1
EJERCICIO.....	2
Código:.....	2
Documento de datos:	2
Explicación:	3
RESULTADOS.....	3
BIBLIOGRAFIA.....	4

INTRODUCCIÓN

Selección de productos de venta:

Crea un fichero de texto con la siguiente información:

```
<nombre_producto1>;<precio_producto_1>  
<nombre_producto_2>;<precio_producto_2>  
---  
<nombre_producto_N>;<precio_producto_N>
```

A continuación, visualiza y elige a través de un elemento html de selección desplegable el nombre de un producto, para que aparezca como resultado su precio.

Mediante la API File carga la información en un array de objetos llamado productos.

EJERCICIO

Código:

```
1 document.getElementById("archivo").addEventListener("change", leerFichero);
2 document.getElementById("elegir").addEventListener("click", mostrarPrecio);
3 let productos = [];
4 let nombreProductos = [];
5 let precioProductos = [];
6
7 function leerFichero(e) {
8     const archivo = e.target.files[0];
9
10    if (!archivo) return;
11
12    const lector = new FileReader();
13
14    lector.readAsText(archivo);
15    lector.onload = function(e) {
16        const datos = e.target.result;
17        productos = datos.split(/[:,\n]/);
18        mostrarProductos();
19    }
20 }
21
22 function mostrarProductos() {
23     document.getElementById("contenedorArchivo").style.display = "none";
24     document.getElementById("contenedorProductos").style.display = "block";
25     normalizarDatos();
26 }
27
28 function normalizarDatos() {
29     productos.forEach(element => {
30         if (isNaN(element)) nombreProductos.push(element);
31         else precioProductos.push(element);
32     });
33     rellenarDesplegable();
34 }
35
36 function rellenarDesplegable() {
37     const desplegable = document.getElementById("productos");
38     for(let i = 0; i < nombreProductos.length; i++) {
39         const option = document.createElement("option");
40         option.setAttribute("value", nombreProductos[i]);
41         option.textContent = nombreProductos[i];
42         desplegable.appendChild(option);
43     }
44 }
45
```

Documento de datos:

```
Laptop Ultraligera;1250.99
Smartphone X Pro;899.00
Auriculares Inalámbricos;149.95
Monitor Curvo 27";345.50
Teclado Mecánico RGB;95.00
```

Explicación:

Cuando se carga el documento, establezco dos oyentes: uno para el botón "elegir", que muestra el precio al hacer clic con `mostrarPrecio()`; y otro para el input "archivo", que comienza la lectura con `leerFichero()` al detectar un evento de cambio.

La función `leerFichero()` emplea `FileReader` para leer en formato de texto el archivo elegido. Cuando se carga (`onload`), separo el contenido del archivo utilizando los caracteres de nueva línea (`\n`) para formar un arreglo de productos y luego hago una llamada a la función `mostrarProductos()`. Esta función hace que el contenedor del archivo no se pueda ver, pero sí el de los productos, mediante la invocación de `normalizarDatos()`.

En la función `normalizarDatos()`, itero a través del arreglo de productos para dividir los nombres (cadenas de texto) en `nombreProductos` y los precios (números) en `precioProductos`, y luego invoco a `rellenarDesplegable()`.

Por último, `rellenarDesplegable()` genera elementos `<option>` para un desplegable ("productos") de manera dinámica a partir de los nombres; mientras que `mostrarPrecio()` obtiene el precio del producto elegido al buscar su índice en `nombreProductos` y acceder a la misma posición en `precioProductos`.

RESULTADOS

Selecciona el archivo de datos: Ningún archivo seleccionado

Elige un producto: ▼

Elige un producto: ▼

El precio es de: 1250.99

BIBLIOGRAFIA

Uso de MDN Web Docs (Mozilla Developer Network): Para poner en práctica la lectura del archivo de texto local (readAsText y evento onload), reviso la documentación de la API File y la clase FileReader. Además, utilizo la manipulación de componentes del DOM (como appendChild o createElement) para crear las opciones (<option>) del menú desplegable de manera dinámica.

Enlaces:

https://developer.mozilla.org/es/docs/Web/API/File_API

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

W3Schools: para los valores seleccionados de los inputs y selects (element.value) para la lógica que busco en precios, valiéndome de esta fuente para fortalecer el manejo de eventos (change, click, addEventListener) y la navegación básica en el DOM.

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

Referencias de JavaScript y lógica de datos: cómo se emplean: Confirmando la aplicación de la función split() para dividir las líneas del archivo y los métodos de array, como forEach e indexOf, para segmentar los datos (nombres/precios) en diferentes arrays y determinar el índice que establece una relación entre el producto y su precio.

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