

## INDICE

INTRODUCCIÓN .....	1
EJERCICIOS .....	2
EJERCICIO 1 .....	2
Código: .....	2
Explicación:.....	3
Vista:.....	3
Ejercicio 2.....	3
Código: .....	4
Explicación:.....	5
Vista:.....	5
BIBLIOGRAFÍA.....	6

## INTRODUCCIÓN

### **Ejercicio 1:**

Crea un formulario HTML con dos elementos: uno select y otro datalist con los mismos datos.

A continuación, mediante código javascript, visualiza el contenido de los datos elegidos de cada elemento, comentando las diferencias que encuentres a la hora de referenciarlos (obtener sus valores).

### **Ejercicio 2:**

Crea un formulario HTML con dos controles o elementos: un select con opciones y el otro un select con agrupaciones de opciones (optgroup) de manera que el primero permita deshabilitar un grupo o varios grupos de opciones del segundo.

## EJERCICIOS

### EJERCICIO 1

Código:

```
<!DOCTYPE html>
<html lang="es">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="styles.css">
    <script src="script.js" defer></script>
    <title>Ejercicio 1</title>
</head>

<body>
    <div>
        <h3>Select HTML</h3>
        <label for="selectFrutas">Elige una fruta:</label>
        <select id="selectFrutas" name="frutaSeleccionada">
            <option value="" disabled selected>-- Selecciona una opción --</option>
            <option value="manzana">Manzana</option>
            <option value="platano">Plátano</option>
            <option value="fresa">Fresa</option>
            <option value="uva">Uva</option>
            <option value="kiwi">Kiwi</option>
        </select>
    </div>
    <div>
        <h3>DataList HTML</h3>
        <label for="inputFrutas">Escribe una fruta (o elige de la sugerencia):</label>
        <input list="listaFrutas" id="inputFrutas" name="frutaSugerida">
        <datalist id="listaFrutas">
            <option value="Manzana">
            <option value="Plátano">
            <option value="Fresa">
            <option value="Uva">
            <option value="Kiwi">
        </datalist>
    </div>
    <div id="ultimoContainer">
        <p id="texto"></p>
    </div>
</body>
</html>
```

```
document.getElementById("selectFrutas").addEventListener("change", verDatos);
document.getElementById("inputFrutas").addEventListener("change", verDatos);
const p = document.getElementById("texto");
const div = document.getElementById("ultimoContainer");
const input = document.getElementById("inputFrutas");

function verDatos(e) {
    p.textContent = e.target.value;
    p.textContent = e.target.value;

    comprobarContainer();
}

function comprobarContainer() {
    if (p.textContent === "") {
        div.style.display = "none";
    } else {
        div.style.display = "block";
    }
}
```

### Explicación:

Construyo una página HTML básica que presenta dos formas de selección de datos: un select tradicional y un input con una datalist.

El select (selectFrutas) permite elegir una fruta de una lista de opciones definidas con <option>. La datalist (listaFrutas) ofrece sugerencias a un campo de texto (inputFrutas) mientras el usuario escribe, combinando la libertad del input con la ayuda de la lista de opciones.

En JavaScript, asigno la función verDatos() al evento change de ambos selectores. La función verDatos() captura el valor seleccionado y lo muestra en un párrafo (p.textContent). Inmediatamente, llamo a comprobarContainer() que verifica si el párrafo tiene contenido; si el párrafo está vacío (por ejemplo, al seleccionar la opción deshabilitada), oculto el contenedor (div.style.display = "none"), y si hay valor, lo hago visible ("block").

### Vista:

The screenshot displays two instances of a 'Select HTML' form. The top instance shows a dropdown menu with the options 'Plátano' and 'Manzana'. The bottom instance shows the same dropdown menu, but the option 'Plátano' is highlighted or selected. Both instances include a label 'Elige una fruta:' and a dropdown menu with the same two options.

The screenshot displays two instances of a 'Datalist HTML' form. The top instance shows an input field with the placeholder text 'Escribe una fruta (o elige de la sugerencia)'. The bottom instance shows the same input field with the text 'platano' entered into it.

The screenshot displays two instances of a 'Select HTML' form. The top instance shows a dropdown menu with the options 'Plátano' and 'Manzana'. The bottom instance shows the same dropdown menu, but the option 'Manzana' is highlighted or selected. Both instances include a label 'Elige una fruta:' and a dropdown menu with the same two options.

## Ejercicio 2

Código:

```
<html lang="es">
<head>
    <title>Selectores Flexibles</title>
    <link rel="stylesheet" href="styles.css">
    <script src="script.js" defer></script>
</head>

<body>

    <div class="container">
        <label for="selectSimple">Elige una Mascota:</label>
        <select id="selectSimple" name="frutaSeleccionada" size="2">
            <option value="Frutas">Frutas</option>
            <option value="Verduras">Verduras</option>
        </select>

        <label for="selectAgrupado">Elige un Alimento:</label>
        <select id="selectAgrupado" name="frutaSugerida" size="4">
            <optgroup id="frutas" label="Frutas" disabled>
                <option value="manzana">🍎 Manzana</option>
                <option value="uva">🍇 Uva</option>
            </optgroup>
            <optgroup id="verduras" label="Verduras" disabled>
                <option value="brocoli">🥦 Brócoli</option>
                <option value="zanahoria">🥕 Zanahoria</option>
            </optgroup>
        </select>
    </div>

</body>

</html>
```

```
document.getElementById("selectSimple").addEventListener("change", desactivar);
const frutas = document.getElementById("frutas");
const verduras = document.getElementById("verduras");
function desactivar(e) {
    console.log(e.target.value);
    if (e.target.value === "Frutas") {
        frutas.removeAttribute("disabled");
        verduras.setAttribute("disabled", true);
    } else {
        verduras.removeAttribute("disabled");
        frutas.setAttribute("disabled", true);
    }
}
```

### Explicación:

Construyo dos estructuras de selección de datos en HTML. La primera es un select simple (selectSimple) con opciones agrupadas por optgroup (Frutas/Verduras), y asocio la función desactivar() a su evento change.

desactivar() obtiene el valor seleccionado del select simple; si es "Frutas", activo el grupo de frutas (removeAttribute("disabled")) y desactivo el de verduras (setAttribute("disabled", true)) en el select agrupado (selectAgrupado), y viceversa si se selecciona "Verduras".

De esta manera, logro una interdependencia entre selectores, activando o desactivando dinámicamente las opciones del segundo select en función de la elección realizada en el primero.

### Vista:

The screenshot shows a user interface with two main sections. On the left, a dropdown menu titled "Elige una Mascota" contains the options "Frutas" and "Verduras". On the right, another dropdown menu titled "Elige un Alimento" contains two groups: "Frutas" with items "Manzana" (apple icon) and "Uva" (grape icon), and "Verduras" with items "Papa" (potato icon) and "Cebolla" (onion icon). The "Verduras" option in the first dropdown is highlighted with a blue selection bar.

### Elige una Mascota:

A close-up view of the "Elige una Mascota" dropdown menu. The "Frutas" option is selected, indicated by a blue horizontal bar above it. The other option, "Verduras", is below it.

### Elige un Alimento:

A close-up view of the "Elige un Alimento" dropdown menu. The "Frutas" group is expanded, showing "Manzana" (apple icon) and "Uva" (grape icon). The "Verduras" group is collapsed, showing its name and a small icon.

## BIBLIOGRAFÍA

MDN Web Docs (Mozilla Developer Network) Uso: Consulto la documentación esencial sobre manipulación del DOM, específicamente para verificar y aplicar los métodos para modificar atributos de los elementos HTML (removeAttribute y setAttribute), que me permiten activar o desactivar los grupos de opciones del selector (optgroup).

Enlace:

[https://developer.mozilla.org/es/docs/Web/API/Document\\_Object\\_Model/Introduccion](https://developer.mozilla.org/es/docs/Web/API/Document_Object_Model/Introduccion)

W3Schools Uso: Reviso los tutoriales sobre Eventos del DOM para reforzar la sintaxis del evento change que utilizo para detectar la selección del usuario, y la navegación básica para obtener el valor seleccionado (e.target.value) que uso en la lógica de interdependencia.

Enlace: [https://www.w3schools.com/js/js\\_htmldom\\_events.asp](https://www.w3schools.com/js/js_htmldom_events.asp)

Referencias de JavaScript y Selectores HTML Uso: Verifico el uso correcto de los selectores HTML anidados (select con optgroup) y la manera de obtener referencias a los elementos mediante su id (getElementById) para controlar su comportamiento dinámicamente con JavaScript.

Enlace: <https://developer.mozilla.org/es/docs/Web/HTML/Element/select>