

PRÁCTICA 4.04 Formularios y eventos

Normas de entrega

- En cuanto al **código**:
 - en la **presentación interna**, importan los **comentarios**, la claridad del código, la significación de los nombres elegidos; todo esto debe permitir considerar al programa como **autodocumentado**. No será necesario explicar que es un **if** un **for** pero sí su funcionalidad. Hay que comentar las cosas destacadas y, sobre todo, las **funciones** y **clases** empleadas. La ausencia de comentarios será penalizada,
 - en la **presentación externa**, importan las leyendas aclaratorias, información en pantalla y avisos de ejecución que favorezcan el uso de la aplicación,
 - todo el código debe estar situado dentro del evento `window.onload = () => {};` o a través del evento `document.addEventListener("DOMContentLoaded", () => {});`,
 - si no se especifica lo contrario, la información resultante del programa debe aparecer en la consola del navegador `console.log(información)`,
 - los ejercicios deben realizarse usando **JavaScript ES6** y usar el modo estricto (`use strict`) No se podrá utilizar *jQuery* ni cualquier otra biblioteca (si no se especifica lo contrario en el enunciado),
 - para el nombre de **variables**, **constantes** y **funciones** se utilizará *lowerCamelCase*,
 - para la asignación de eventos se utilizará `addEventListener()` indicando sus tres parámetros en su definición,
 - se usarán las funcionalidades **import** y **export** para crear bibliotecas de funciones temáticas (no debe haber declaración de funciones ni objetos en el documento principal).
- En cuanto a la **entrega** de los archivos que componen los ejercicios:
 - todos los ejercicios en **una carpeta** (creando las **subcarpetas** necesarias para documentación anexa como imágenes o estilos) cuyo nombre queda a discreción del discente,
 - el nombre de los ficheros necesarios para resolver el ejercicio será el número de ejercicio que contenga,
 - el código contendrá ejemplos de ejecución, si procede, y
 - la carpeta será comprimida en formato **ZIP** y será subida a **Aules** de forma puntual.

Ejercicio 1 - Más números

Haz un programa que cree cien elementos `<checkbox>` con números aleatorios entre el 100 y el 1.000. Además, la página tendrá un botón **Marcar todos los pares** y un botón **Desmarcar todos** que realizarán las acciones que describen su texto.

Ejercicio 2 - Creador párrafos

Crea un formulario que tenga un `<textarea>`, un `<select>` con tres tipos de formato (inventate tres estilos de texto en clases `CSS`) y un botón con el texto **Crear párrafo**. Cuando el usuario haga clic sobre el botón se creará un nuevo párrafo `<p>` dentro de un `<div id="resultado">` situado debajo del formulario. Este nuevo `<p>` contendrá el texto del `<textarea>` con el estilo `CSS` elegido en el `<select>`. En caso de que cualquiera de los dos campos esté vacío, mostrará un mensaje de error **insultando** informando al usuario y no creará ningún párrafo. El mensaje de error deberá aparecer en un `div` para ello embebido en el código HTML, no se deberá usar `alert`.

Ejercicio 3 - Discos

Prepara un formulario diseñado para almacenar los discos con los datos listados a continuación:

- nombre del disco,
- grupo de música o intérprete,
- año de publicación,
- tipo de música (podrá ser "rock", "progressive", "punk" o "trash"),
- localización, almacenará un número de estantería (inventado pero que contiene números y letras),
- prestado, almacenará un valor booleano. Por defecto será **false**.

Tendrá un botón **Guardar** que añadirá el disco a un listado almacenado en un objeto (que tendrás que diseñar) **JSON** y otro **Mostrar** que mostrará el listado de discos debidamente formateado obteniendo la información del **JSON** (usa `CSS`).

Ejercicio 4 - Pueblos

Realiza un formulario con dos elementos `<select>` (provincia y población). Al cambiar el primero se actualizará el segundo. Al enviar el formulario (evento `submit`) se comprobará que ambos han sido marcados a una opción válida. Cuando no tengan ninguna selección previa los campos mostrarán **No seleccionado** que será la situación inicial. Los valores del primer campo serán **Alicante**, **Castellón** y **Valencia**. Los valores para el segundo campo son:

- cuando el primer campo sea **Alicante**: **Alicante**, **Elche** y **Petrer**,
- cuando el primer campo sea **Castellón**: **Castellón**, **Oropesa** y **Vinaroz**,
- cuando el primer campo sea **Valencia**: **Valencia**, **Xàtiva** y **Torrent**.

Ejercicio 5 - Discos II

A partir del formulario creado en la actividad **Discos**, realiza las funciones necesarias para validarlo teniendo en cuenta:

- nombre del disco al menos cinco caracteres y obligatorio
- grupo de música o intérprete al menos cinco caracteres y obligatorio
- año de publicación cuatro caracteres numéricos
- tipo de música comprobará si se ha seleccionado alguno
- localización formato **ES-001AA** donde **001** es el número de la estantería y **AA** la balda (combinación de dos letras mayúsculas)
- prestado no tiene comprobación

La validación se asignará al formulario mediante un **addEventListener** y los campos nombre del disco y grupo de música se validarán en la **misma función**.

En caso de que se produzca un error en la validación, el campo implicado tendrá el borde rojo de tamaño **3 px** y la etiqueta que lo acompaña también aparezca de color rojo. Crea las clases **CSS** que estimes oportuno en un fichero aparte. Si se corrigen los datos una vez cambiado el estilo del **<input>** deberá volver a su color habitual.