

## Ejercicios Unidad 9 – Formularios y controles

---

### EJERCICIO: u09e01\_formulario

Prepara un formulario diseñado para almacenar los discos con los que trabajamos en los ejercicios de una unidad anterior. Los campos serán los siguientes:

- Nombre del disco.
- Grupo de música o cantante.
- Año de publicación.
- Tipo de música (podrá ser “rock”, “pop”, “punk” o “indie”);
- Localización: almacenará un número de estantería.
- Prestado: almacenará un valor booleano. Por defecto será false.

Realiza las funciones necesarias para validarlo con JavaScript teniendo en cuenta:

- Nombre del disco: 20 caracteres, obligatorio.
- Grupo de música o cantante: 20 caracteres, obligatorio.
- Año de publicación: 4 caracteres numéricos.
- Tipo de música (podrá ser “rock”, “pop”, “punk” o “indie”): sin comprobación.
- Localización: almacenará un número de estantería: vacío o numérico.
- Prestado: sin comprobación.

Deberás tener en cuenta, además:

- La validación de cada campo se realizará al cambiar en cada campo (no todas las validaciones al final).
- Al enviar el formulario, se validará sólo que los campos obligatorios se han rellenado.
- Se utilizará el evento “DOMContentLoaded”
- Los campos nombre del disco y grupo de música se validarán en la misma función (campo20).
- Toda la validación se realizará con JavaScript (no con HTML)

Ampliación:

- Modifica las funciones de tal manera que, en caso de que se produzca un error en la validación, se marque de alguna manera visible, atractiva y cómoda para el usuario.
- Para ello deberás crear las clases CSS necesarias (puedes incluirlas en el propio HTML si lo deseas) para que se visualice correctamente.
- Ten en cuenta que, si el usuario pone correctamente los datos, el campo deberá volver a su formato original.

### EJERCICIO: u09e02\_test

Vas a crear un pequeño test sobre la serie de televisión que tú quieras. El test tendrá 5 preguntas con 3 respuestas cada una a elegir con botones de radio. El test deberá hacer lo siguiente:

- Cuando el usuario conteste las 5 preguntas obtendrá el resultado final en un cuadro de texto.
- Además, el usuario podrá ver qué preguntas ha fallado, porque al enviar el formulario le aparecerá un pequeño icono con un tic verde en las preguntas correctas y una cruz roja en las preguntas incorrectas.
- En caso de que el usuario deje alguna pregunta sin contestar, no mostrará el resultado e indicará con un mensaje “No has respondido a todas las preguntas”. Y se marcará en color rojo la pregunta que no haya sido respondida.

## EJERCICIO: u09e03\_expresionesRegulares

**Breaking Bad.** Walter White y Jesse Pinkman tienen que controlar las bolsas de producto azul que cocinan para distribuir en Albuquerque. Para ello tienen que crear un formulario que almacene la información de cada bolsa y validarlo teniendo en cuenta lo siguiente:

- Fecha de creación: obligatorio, fecha válida y con formato dd/mm/aaaa.
- Cocinero: será un nombre en clave formado por dos letras en mayúscula, un símbolo y cuatro dígitos (ej. WW\$1234)
- Destinatario: estará formado por dos o tres letras mayúsculas correspondientes al estado, un guion bajo, el nombre de la ciudad en minúsculas, dos puntos, y el código de distrito de 4 dígitos (ej. NM\_albuquerque:1234).
- Gramos: será un número del 100 al 5000.
- Composición: estará formado por una cantidad en gramos seguida de dos conjuntos de una o dos letras seguidas o no de un número. (ej. 200gC3OH7)
- Número de cuenta de EEUU: supongamos que un número de cuenta estadounidense tiene el siguiente formato:
  - Dos letras: suponemos que el valor de cada letra es del 1 al 26 (no hay ñ ni ll).
  - Dos dígitos: debe corresponderse con la suma de la primera letra y la segunda: en caso de que sea menor que 10 se pone el 0 delante.
  - Un guion.
  - Doce dígitos de cuenta.
  - Un guion.
  - Dos dígitos de control: Cómo se obtiene cada uno:
    - Primer dígito: Sumar los 6 primeros dígitos de la cuenta, dividir entre 6 y extrayendo parte entera del cociente.
    - Segundo dígito: exactamente igual, pero con los 6 siguientes.
  - Si el número está correcto se colocará en un campo de texto al lado del anterior, pero sin guiones: solamente los números y las letras.

## EJERCICIO: u09e04\_test\_cookies

Crea una nueva versión del ejercicio del test, donde:

- Cada vez que el usuario trate de enviar el formulario y haya algún error, una variable contador (almacenada en una cookie) se incrementará.
- El resultado del número de intentos se reflejará en un campo de texto que se encontrará al final del formulario.
- Si el usuario sale del programa y vuelve a entrar, el campo de texto mostrará el número almacenado en la cookie.
- Junto al campo de texto habrá un botón que, al pulsarlo, permitirá reiniciar el valor de la cookie a 0.

\*\* Probarlo con Firefox, porque Chrome no almacena cookies para archivos locales.

## EJERCICIO: u09e05\_test\_Storage

Crea una nueva versión del ejercicio anterior, pero implementada con WebStorage. Decide si debes emplear localStorage o sessionStorage y por qué.

## EJERCICIO: u09e06\_repaso

Crea una página web con HTML y JavaScript que permita la reserva online de salas para reuniones de una empresa.

En un formulario se deben recoger y validar los siguientes datos, todos obligatorios excepto el número de socio:

- Nombre y apellidos (texto).
- DNI.
- Número de personas.
- Sala con cocina: pueden ser con cocina o sin cocina (se debe hacer mediante una selección bien mediante `checkbox` o `radiobutton`).
- Número de socio. El número de socio está compuesto por dos letras al principio seguido de 3 dígitos.

Las salas con cocina por día cuestan 150 €, y las que no tienen cocina cuestan 100 €. Además, si es socio se le hará un descuento del 10 %.

Se podrán realizar las siguientes operaciones:

- **Reservar:** permitirá reservar una sala solicitando la información anterior. No permitirá que un DNI reserve dos veces. Además, se ha de tener en cuenta que existen para alquilar 4 salas con cocina y 3 sin cocina. Se ha de tener en cuenta que la capacidad máxima de las salas es de 25.
- **Anular:** introduciendo el DNI me permitirá anular una reserva.
- **Mostrar:** mostrará todas las reservas realizadas.
- **Salir:** sale de la aplicación.

Aclaraciones:

- Para poder almacenar la información se utilizará un array de objetos.
- Si los datos introducidos en el formulario no son correctos, se informará sobre el error y no se realizará la reserva hasta que se corrijan.