

Ejercicios Unidad 4 - Interacción con el usuario: eventos y formularios.

Eventos

EJERCICIO: u4e01_bombillaTradicional y u4e01_bombillaW3C

Crear los siguientes scripts, partiendo del ejemplo **01_EventosEnLinea**:

1. **u4e01_bombillaTradicional**: que contendrá el resultado de transformar el modelo de eventos en línea al modelo de eventos tradicional.
2. **u4e01_bombillaW3C**: que contendrá el resultado de transformar el modelo de eventos en línea o el modelo de eventos tradicional, al modelo de eventos del W3C.

EJERCICIO: u4e02_calculadora

Partiendo del ejemplo **06_EventosCalculadoraW3CBubbling**, añadir a nuestra calculadora dos botones más x^3 y $1/x$. Habrá que realizar la gestión de eventos correspondiente para que funcionen correctamente las dos operaciones nuevas.

También se añadirá un **campo informativo** mediante la etiqueta `<h3>` con contenido inicial "Información sobre el número". El contenido del campo se debe actualizar cada vez que se hace un cálculo en la calculadora. Su contenido debe ser "Información: el resultado de la operación $x^2 \mid x^3 \mid 1/x$ es *número*."

EJERCICIO: u4e03_mostrarInfo

A partir de la página web proporcionada, completar el código JavaScript para que se comporte de la siguiente forma:

1. Cuando se haga clic sobre el primer enlace, se oculte su sección relacionada.
2. Cuando se vuelva a hacer clic sobre el mismo enlace, se muestre otra vez esa sección de contenidos.
3. Completar el ejercicio con el resto de enlaces de la página para que su comportamiento sea idéntico al del primer enlace
4. Cuando una sección se oculte, debe cambiar el mensaje del enlace asociado

Si es necesario, modifica el HTML.

EJERCICIO: u4e04_zonasPulsar

A partir del archivo HTML proporcionado, crear un script que informe al usuario en qué zona de la pantalla ha hecho clic el ratón. Las zonas posibles son las siguientes: izquierda arriba, izquierda abajo, derecha arriba y derecha abajo, todas relativas al tamaño de la ventana del navegador.

EJERCICIO: u4e05_informacionEventos

Partiendo del archivo HTML proporcionado, crear un script que haga lo siguiente:

- Al mover el ratón en cualquier punto de la ventana del navegador, se muestre la posición del puntero respecto del navegador y respecto de la pantalla:



- Al pulsar cualquier tecla, se debe mostrar el valor de la tecla pulsada y su código asociado:

Teclado
 Carácter [s]
 Código [KeyS]

- Añadir la siguiente característica al script: cuando se pulsa un botón del ratón, el color de fondo del cuadro de mensaje debe ser amarillo (#FFFFCC) y cuando se pulsa una tecla del teclado, el color de fondo debe ser azul (#CCE6FF). Al volver a mover el ratón, el color de fondo vuelve a ser blanco.

Ratón
 Navegador [324, 244]
 Pantalla [324, 347]

Teclado
 Carácter [d]
 Código [KeyD]

EJERCICIO: u4e06_parejas

Vas a crear un juego de memorizar parejas. Consiste en encontrar parejas en un tablero de 12 cartas, que contiene 6 cartas repetidas de los personajes de los Simpson. El juego consistirá en lo siguiente:

- La aplicación deberá tener una tabla con 3 filas y cuatro columnas de un color. Además, habrá un cuadro de texto con el valor 0 pero no modificable y un botón que permita reiniciar el juego.
- Cuando el usuario haga clic sobre una celda, se mostrará una imagen.
- Cuando el usuario haga clic sobre otra celda, se mostrará otra imagen.
- Si las dos imágenes son iguales, se mantendrá visibles y se añadirá 1 al cuadro de texto.
- Si las dos imágenes son diferentes, se ocultarán mostrando nuevamente el color inicial.
- Cuando se hayan completado las 6 parejas, se mostrará el mensaje: ¡¡GANASTE!!
- El botón de reiniciar el juego, cambia el orden de las cartas para volver a jugar. El siguiente método te puede ayudar, permite ordenar un array aleatoriamente:

```
array.sort(() => 0.5 - Math.random());
```

Validar formularios y almacenar información en el navegador

EJERCICIO: u4e07_formulario

Prepara un formulario diseñado para almacenar los discos con los que trabajamos en los ejercicios de la 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:

- Se utilizará el evento “DOMContentLoaded”
- La validación se asignará al formulario mediante el método `.addEventListener()`.
- Los campos nombre del disco y grupo de música se validarán en la misma función (campo20).

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: u4e08_test

Vas a crear un pequeño test sobre la serie de televisión que tú quieras. El test tendrá 10 preguntas con 3 respuestas cada una a elegir con botones de radio. El test deberá hacer lo siguiente:

- Cuando el usuario conteste las 10 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: u4e9_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: los dos primeros deben ser la suma de los 6 primeros dígitos de la cuenta dividido entre 6 y extrayendo solamente su parte entera; y los dos últimos 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: u4e10_cookiesExpresionesRegulares

Crea una nueva versión del ejercicio anterior, 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: u4e11_localStorageExpresionesRegulares

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

EJERCICIO: u4e12_test

Modifica el ejercicio **u4e08_test** para que el usuario pueda salir de la página y continuar en otro momento.

EJERCICIO: u4e13_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 `radio` button).
- 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.