

# EJERCICIOS UNIDAD 5

## EJERCICIO 5.1

Haz un programa que mediante eventos y el uso del objeto event, te muestre en todo momento la posición actual del ratón en pantalla. Para mostrarlo modificaremos de forma dinámica un elemento HTML para que nos muestre la posición actual del ratón.

## EJERCICIO 5.2

Realizar un programa con dos botones “Comenzar Saludos” y “Parar Saludos”. Al hacer click en “Comenzar Saludos”, lanzará un setInterval que cada 3 segundos muestre un alert con “Hola”. El botón “Parar Saludos” parará esa secuencia.

## EJERCICIO 5.3

Necesitamos un **formulario con dos campos**: en el primero se introduce los **dígitos del NIF** y en el segundo debería aparecer **automáticamente, sin necesidad de hacer click en ningún botón, la letra** correspondiente.

## EJERCICIO 5.4

Observa el funcionamiento de la página siguiente.

<https://www.loteriasypuestas.es/es/resultados/euromillones>

Al colocarte encima de la frase “**ver por orden de aparición**” los números se ordenan en el orden en el que salieron en el sorteo. Dispones de un array con los números del sorteo en el orden de aparición.

Crea una página web que realice esa funcionalidad.

## EJERCICIO 5.5

Prepara un formulario diseñado para introducir los datos de los discos con los que trabajamos la U4. Recuerda qué elementos se almacenaban, y elige el componente de formulario más adecuado:

- 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: máximo 20 caracteres, obligatorio.
- Grupo de música o cantante: máximo 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.

Tendrás que tener en cuenta, además:

- Que 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 tenga el reborde rojo, y la etiqueta que lo acompaña también aparezca de color rojo.
- 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 mete correctamente el nombre, deberá volver a su color habitual.

## EJERCICIO 5.6

Vas a crear un juego que consiste en encontrar parejas en 12 cartas con 6 parejas de imágenes. El juego consistirá en lo siguiente:

- La aplicación deberá tener una tabla con 3 filas y cuatro columnas de un color (puede ser también una imagen con un color) . Además se mostrará un cuadro de texto de aciertos con el valor inicial 0.
- 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 quedarán a la vista y se añadirá 1 al cuadro de texto de aciertos.
- Si las dos imágenes son diferentes, se ocultarán mostrando nuevamente el color inicial.

Además al entrar en la página al usuario se le pedirá un nombre la primera vez que entre, ese nombre será almacenado en una **cookie** además de un contador de visitas a la página.

Si vuelve a entrar en la página se le mostrará un mensaje en la misma página diciendo Hola “nombre” y el número de veces que visitó la página.

Puede ser de utilidad el atributo **src** del elemento `<img>`. Podremos acceder desde `event.target.src` o `this.src`

### EJERCICIO 5.7

Crear un formulario que compruebe la información de los envíos de una empresa y validarlo por medio de expresiones regulares teniendo en cuenta lo siguiente:

- Fecha de creación: obligatorio y con formato dd/mm/aaaa.
- Empleado: será un nombre en clave formado por dos letras en mayúscula, un símbolo (algo que no sea ni letras ni números) y cuatro dígitos (ej. WW\$1234)
- Destinatario: estará formado por dos o tres letras mayúsculas correspondientes al estado, un guión bajo, el nombre de la ciudad con la primera letra Mayúscula y el resto en minúsculas, dos puntos, y el código de distrito de 4 dígitos (ej. NM\_Madrid:1234).
- Peso en gramos: será un número del 100 al 5000.
- Número de cuenta: Formato habitual de los números de cuenta con ( IBAN + 20 dígitos)

### EJERCICIO 5.8

Crea un formulario con un pequeño test sobre una temática que tu cedidas. El test tendrá 4 preguntas con 3 respuestas cada una a elegir con radio buttons.

El programa deberá hacer lo siguiente:

- Cuando el usuario conteste las 4 preguntas y envíe los datos obtendrá el resultado final (“Has acertado x preguntas”)

- Además, el usuario podrá ver qué preguntas ha fallado, ya que al enviar el formulario le aparecerá un pequeño icono con un tick 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 5.9

Simula la pantalla de creación de nuevo usuario de Amazon, utilizando las características de HTML5 y Javascript.:

### [Amaz3n crear usuario](#)

The screenshot shows the Amazon.es 'Crear cuenta' (Create account) form. At the top is the Amazon.es logo. The form title is 'Crear cuenta'. It contains the following fields and labels:

- Nombre**: A text input field with a red border and a small red exclamation mark icon to its left. Below it is the text 'Introduce tu nombre'.
- N3mero de m3vil o direcci3n de correo electr3nico**: A text input field with a red border and a small red exclamation mark icon to its left. Below it is the text 'Indica tu direcci3n de e-mail o tel3fono m3vil'.
- Contrase1a**: A text input field with a red border and a small red exclamation mark icon to its left. Below it is the text 'Al menos 6 caracteres' and 'M3nimo de 6 caracteres obligatorios'.
- Confirma tu contrase1a**: A text input field with a red border.

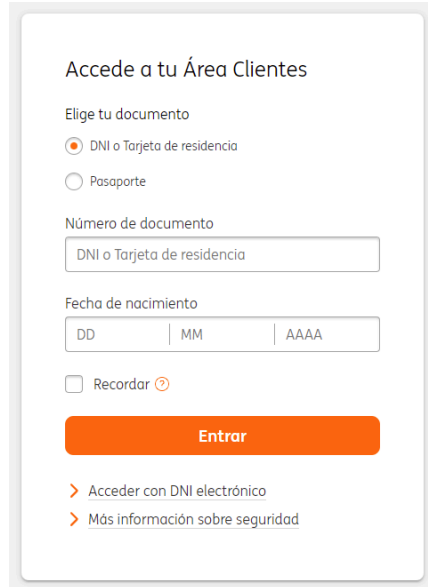
Below the fields is a yellow 'Continuar' button. Underneath the button, there is a paragraph: 'Al identificarte aceptas nuestras Condiciones de uso y venta. Consulta nuestro Aviso de privacidad, nuestro Aviso de Cookies y nuestro Aviso sobre publicidad basada en los intereses del usuario.' Below this paragraph are two links: '¿Ya tiene una cuenta? Iniciar sesi3n' and '¿Compras para tu empresa? Crear una cuenta de empresa gratuita'.

At the bottom of the form, there is a footer with links: 'Condiciones de uso', 'Aviso de privacidad', 'Ayuda', 'Cookies', and 'Publicidad basada en intereses'. Below these links is the copyright notice: '© 1996-2021, Amazon.com, Inc. o afiliados. Todos los derechos reservados.'

Utiliza LocalStorage para almacenar el nombre del usuario la primera vez que se registre, cuando vuelva a entrar deben de aparecer los datos ya colocados.

## EJERCICIO 5.10

Simula la página de bienvenida de un banco (ING en este caso):



Accede a tu Área Clientes

Elige tu documento

☒ DNI o Tarjeta de residencia

☐ Pasaporte

Número de documento

DNI o Tarjeta de residencia

Fecha de nacimiento

DD MM AAAA

☐ Recordar

Entrar

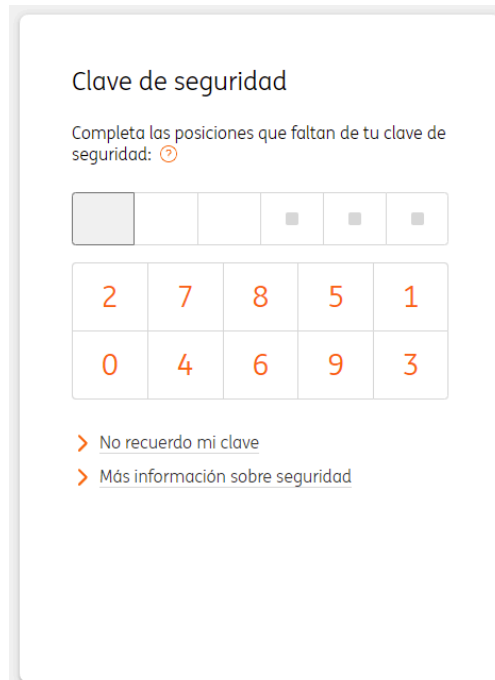
> Acceder con DNI electrónico

> Más información sobre seguridad

Los datos de DNI-Tarjeta de residencia-Pasaporte, fecha de nacimiento y clave están almacenados en un array de objetos de tipo Cliente.

El checkBox permite almacenar los datos de un cliente por medio de WebStorage, para que la próxima vez que acceda a la página le aparezca el DNI y la fecha de nacimiento ya colocados.

Cliente ha de ser una clase que almacene la información del cliente necesaria para que al pulsar en entrar acceda a la siguiente ventana:



Clave de seguridad

Completa las posiciones que faltan de tu clave de seguridad:

--	--	--	--	--

2	7	8	5	1
0	4	6	9	3

> No recuerdo mi clave

> Más información sobre seguridad

En esta ventana se mostrarán en una tabla los números del 0 al 9 en posiciones aleatorias.

El usuario completará los números de su clave de seguridad que faltan por rellenar pulsando sobre ellos y validando la clave al pulsar en el último. (los huecos a rellenar también serán aleatorios).

En caso de introducir la clave correcta saldrá un mensaje de bienvenida.

### **EJERCICIO 5.11**

Realiza un aplicación en javascript en la que aparezca un botón en pantalla y que cada vez que intente hacer click en él, cambie su posición para no poder pulsarlo.  
(puede ser de ayuda las propiedades `style.position` `style.top` `style.left`)