

Ejercicios Unidad 2

EJERCICIO 1

Crea un programa en el que crees 5 variables numéricas (entero, decimal, científico, octal y hexadecimal).

A las variables les asignarás los siguientes números: 1357, 135.7, 135e7, 01357 y 0x1357.

Muestra con 5 alerts su valor, escribiendo la siguiente sentencia:

```
alert ("Número tipo" + entero);
```

Comenta el código con los comentarios que estimes necesarios.

EJERCICIO 2

Crea un programa en el que crees 4 variables, 2 cadenas y 2 números, con los siguientes valores: tu nombre, tu apellido, tu edad y tu año de nacimiento.

- Muestra en un alert una frase que incluya comillas simples.
- Muestra en un alert tu nombre y apellidos separados por un salto de línea.
- Muestra en un alert la suma de las variables edad y año de nacimiento.
- Muestra en un alert la suma de todas las variables.

Comenta el código con los comentarios que estimes necesarios.

EJERCICIO 3

Crea un programa en el que muestres el resultado de varias operaciones mediante alert, mostrando el texto exacto de la operación realizada y su resultado.

Ej:

- `var operacion1 = (10 == 10);`
- `alert ("La operación 10==10 es" + operacion1);`

Las operaciones a realizar son:

- `10 == 10`
- `10 === 10`
- `10 === 10.0`
- `"Laura" == "laura"`
- `"Laura" > "laura"`
- `"Laura" < "laura"`
- `"123" == 123`
- `"123" === 123`
- `parseInt("123") === 123`

Comenta el código con los comentarios que estimes necesarios.

EJERCICIO 4

Crea un programa que pida al usuario que introduzca una edad y muestre el siguiente mensaje en función del número introducido:

- 0-12: Niño
- 13-26: Joven
- 26-60: Adulto
- >60: Jubilado

Para guardar un valor del usuario utilizamos el siguiente código:

- `var respuesta = prompt ("Mensaje");`

También comprobará que no se meta un número menor de 0. Comenta el código con los comentarios que estimes necesarios.

EJERCICIO 5

Crea un programa que genere un listado de horas que vayan desde las 9 hasta las 21:30 de 30 minutos en 30 minutos.

En lugar de mostrar un alert, utiliza la instrucción `document.write("Mensaje")` que muestra los datos en el propio HTML.

Ten en cuenta que tendrás que utilizar un for para las horas y (por ejemplo) una variable booleana que irá cambiando de true a false si es hora en punto u hora media.

Comenta el código con los comentarios que estimes necesarios.

EJERCICIO 6

Crea un programa que genere un listado de horas que vayan desde las 9 hasta las 21 de 5 minutos en 5 minutos.

En lugar de mostrar un alert, utiliza la instrucción `document.write("Mensaje")` que muestra los datos en el propio HTML.

Comenta el código con los comentarios que estimes necesarios.

EJERCICIO 7

Crea un control parental para que no se pueda entrar en una página.

Se pedirá al usuario que escriba el apellido del primer presidente de la democracia. (si no sabes quién fue, busca en Google)

Para guardar un valor del usuario utilizamos el siguiente código:

`var respuesta = prompt ("Mensaje");`

Mientras el usuario no introduzca el valor correcto, el mensaje seguirá saliendo.

Comenta el código con los comentarios que estimes necesarios.

EJERCICIO 8

Modifica el ejercicio anterior de la siguiente forma:

- En primer lugar, se carga un prompt donde pregunta: “¿Cuál fue el primer presidente de la democracia española?”
- En caso de que el usuario introduzca mal nombre y apellido, muestre como mensaje: “ERROR. Inténtelo de nuevo. ¿Cuál fue el primer presidente de la democracia española?”
- En caso de que el usuario introduzca el nombre del presidente (solamente) el prompt muestre como mensaje “Te falta el apellido. ¿Cuál fue el primer presidente de la democracia española?”.
- En caso de que el usuario introduzca el apellido del presidente (solamente) el prompt muestre como mensaje “Te falta el nombre. ¿Cuál fue el primer presidente de la democracia española?”.

Comenta el código con los comentarios que estimes necesarios.

EJERCICIO 9

Modifica el programa 4 que utilizaba ifelse para que realice la misma operación pero utilizando un switch

Ejercicio 10

Realiza un programa que solicite, un nombre de un jugadora de fútbol de la selección y que responda si está convocada o no para el siguiente partido. La lista de convocadas es: Luisa, Maria, Carlota, Ana, Martina y Claudia.

Resuélvelo de dos maneras distintas.

Ejercicios Unidad 3

EJERCICIO 1 (objeto Date)

Crea un programa que muestre el número de días que quedan desde hoy hasta el fin de curso (por ejemplo, el 22 de junio).

EJERCICIO 2 (objeto Date)

Crea un programa que pida el día y el mes de tu cumpleaños y muestre los años en que tu cumpleaños va a caer en domingo desde el año actual hasta el año 2100.

EJERCICIO 3 (objeto Math)

Crea un programa que pida al usuario que elija una opción del siguiente menú:

1. Potencia.
2. Raíz.
3. Redondeo.
4. Trigonometría.

Si el usuario introduce 1, se le deberá pedir una base y un exponente y se mostrará el resultado en pantalla (La potencia de X elevado a Y es:)

Si el usuario introduce 2, se le pedirá un número (no negativo) y se mostrará el resultado en pantalla (La raíz de X es:)

Si el usuario introduce 3, se le pedirá un decimal por pantalla y se mostrará el redondeo al entero más próximo, al alta y a la baja.

Si el usuario introduce 4, se le pedirá un ángulo (entre 0 y 360) y se le mostrarán por pantalla los valores trigonométricos del seno, coseno y tangente.

EJERCICIO 4 (objeto Math)

Crea un programa que pida al usuario el valor del radio y muestre por pantalla:

1. El valor del radio.
2. El valor del diámetro.

3. El valor del perímetro de la circunferencia.
4. El valor del área del círculo.
5. El valor del área de la esfera.
6. El valor del volumen de la esfera.

- El valor de Pi debes obtenerlo del objeto Math, no introducirlo manualmente.
- Debes escribir al lado si son cm, o cm², o cm³.

Por ejemplo, si metes 5, deberías obtener: 5 / 10 / 31,41 / 78,54 / 314,15 / 523,59.

EJERCICIO 5 (objeto Number)

Crea un programa que pida al usuario un número entero por pantalla y muestre:

1. Su valor exponencial.
 2. El número con 4 decimales.
 3. El número en binario.
 4. El número en octal.
 5. El número en hexadecimal.
- Utiliza para ello los métodos del objeto Number.
 - Por ejemplo si metes 50, deberías obtener: 5e1 / 50.0000 / 00110010 / 62 / 0x32

EJERCICIO 6 (objeto String)

Crea un programa que pida al usuario su nombre y apellidos y muestre:

1. El tamaño del nombre más los apellidos (sin contar espacios).
2. La cadena en minúsculas y en mayúsculas.
3. Que divida el nombre y los apellidos y los muestre en 3 líneas, donde ponga

Nombre:

Apellido 1:

Apellido 2:

4. Una propuesta de nombre de usuario, compuesto por la inicial del nombre, el primer apellido y la inicial del segundo apellido. Ej. Para Marcos Prado Fernández sería mpradof
5. Una propuesta de nombre de usuario compuesto por las tres primeras letras del nombre y de los dos apellidos: ej. MarPraFer

EJERCICIO 7 (Objeto Window, método setInterval())

Realizar un programa que, pasados 20 segundos, nos muestre una vez la fecha actual del sistema.

EJERCICIO 8 (objeto String)

Haz un programa que reciba cadenas del tipo:

“nombre:apellidos:teléfono:email:codigopostal” y que te muestre:

- El código postal.
- Los apellidos
- El email.
- Suponiendo un formato de email “direccion@servidor” te muestre el servidor asociado.
- El teléfono.

EJERCICIO 9 (objeto String)

Estamos implementando una funcionalidad en nuestra aplicación de creación de NIFs y NIE y necesitamos una calculadora de la letra de control..

– Los NIFs están formados por **8 dígitos y una letra de control**.

-Los NIEs están formados por una letra (X, Y o Z), **7 dígitos y una letra de control**.

[enlace donde se explica cómo calcular ese dígito de control.](#)

Necesitamos un **formulario con dos campos**: en el primero se introducen los **dígitos del NIF o NIE** y en el segundo debería aparecer **la letra** correspondiente.

EJERCICIOS UNIDAD 4

EJERCICIO 1

Vamos a gestionar una lista de países haciendo uso de Arrays.

Cuando el usuario cargue la página, se cargarán las opciones:

- Mostrar el número de países.
- Mostrar listado de países (y le preguntará si quiere mostrarlos en el orden que se encuentran en el array, del revés u ordenados alfabéticamente).
- Añadir un país (y le preguntará si quiere añadir al principio o al final).
- Borrar un país (y le preguntará si quiere borrar al principio o al final).
- Consultar un país (y le preguntará si quiere consultar por posición o por nombre).

Para ello necesitarás crear un archivo `arrays.js` que incluya las siguientes funciones, recibirán un array por parametro:

- Mostrar el número de elementos del array.
- Mostrar todos los elementos del array.
- Muestra los elementos del array en sentido inverso.
- Muestra los elementos del array ordenados alfabéticamente.
- Añadir un elemento solicitado al usuario al principio del array.
- Añadir un elemento solicitado al usuario al final del array.
- Borrar un elemento al principio del array (y decir cuál se ha borrado).
- Borrar un elemento al final del array (y decir cuál se ha borrado).
- Muestra el elemento que se encuentra en una posición que el usuario indica.
- Muestra la posición en la que se encuentra un elemento que le indica el usuario.

EJERCICIO 2

Haz un programa que pregunte una letra de la A a la Z. Tras ello te dirá cuantos DNIs de 3 cifras (del 001 al 999) tienen esa letra y los meterá en un array, tras ello te mostrará el listado de todos los DNIs que tienen esa letra.

EJERCICIO 3

Crea una clase Cliente y define un array de clientes con los siguientes datos:

Nombre;Localidad;Cuota
Laura;Santander;50
Álvaro;Castro;50
Igor;Castro;60
Ivan;Santander;40
Mónica;Zamora;30
Javi;Bilbao;30
David;Bilbao;50

A partir del mismo, el usuario podrá elegir del menú:

1. Todos los clientes: se mostrará una tabla con los valores que están en la variable anterior.
2. Usuarios de una provincia: y a partir de la provincia introducida por el usuario se mostrarán en una tabla los nombres y cuotas de las personas que viven en esa provincia.
3. Usuarios que tengan una cuota mayor o menor que un valor: y se mostrarán en una tabla los nombres de usuario, provincias y cuotas de aquellos que tienen una cuota superior o inferior al valor introducido por el usuario (valora cuál es el mejor modo de hacerlo).

EJERCICIO 4

Necesitamos almacenar en un programa todos los discos de música que tenemos en casa.

Crea una clase “Disco” que almacene la siguiente información:

- 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.
- Carátula: nombre del archivo de imagen de la carátula (p.ej: imagen.png)

Además tendrá los siguientes métodos:

- Un constructor sin parámetros (las 4 primeras propiedades serán cadenas vacías, la localización será 0 por defecto y prestado estará a false).
- Un método que permitirá cambiar el número de estantería en la localización.
- Un método que permitirá cambiar la propiedad Prestado.
- Un método que muestre toda la información de un disco.

Guarda todo el código en un archivo llamado disco.js y reutiliza en tu página el archivo de arrays que hicimos en la práctica 1.

Crea un array vacío para almacenar los discos.

Cuando el usuario cargue la página, se mostrarán las opciones:

- Mostrar número de discos.
- Mostrar tabla de discos con todos los datos de cada disco(y le preguntará si quiere mostrarlos en el orden que se encuentran en el array, del revés u ordenados alfabéticamente).
- Mostrar un intervalo de discos(y le pedirá que introduzca el intervalo en formato inicio-fin; luego deberás extraer el valor inicio y fin).
- Añadir un disco (y le preguntará si quiere añadir al principio o al final).
- Borrar un disco (y le preguntará si quiere borrar al principio o al final).
- Consultar un disco (y le preguntará si quiere consultar por posición o por nombre).

EJERCICIO 5

Estamos desarrollando un juego y necesitamos crear un objeto jugador, que tenga una propiedad fuerza que inicialmente vale 1, tendremos un método incrementarFuerza que aumentará la fuerza del jugador en 1 y un método consultar fuerza que mostrará un mensaje: “Tu fuerza es x” donde x es la fuerza actual del jugador.

Mostrar 2 botones que sirvan para incrementar la fuerza y para mostrar la fuerza.

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 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 elije 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: 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.

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. Además habrá un cuadro de texto con el valor 0 pero no modificable.
- 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 cambiará el color de la celda y se añadirá 1 al cuadro de texto.
- 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 ``. Podremos acceder desde `event.target` 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 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 en un cuadro de texto.
- 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.:

[Amazon crear usuario](#)

amazon.es

Crear cuenta

Nombre

/ Introduce tu nombre

Número de móvil o dirección de correo electrónico

/ Indica tu dirección de e-mail o teléfono móvil

Contraseña

Al menos 6 caracteres
/ Mínimo de 6 caracteres obligatorios

Confirma tu contraseña

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.

¿Ya tiene una cuenta? [Iniciar sesión](#)
¿Compras para tu empresa? [Crear una cuenta de empresa gratuita](#)

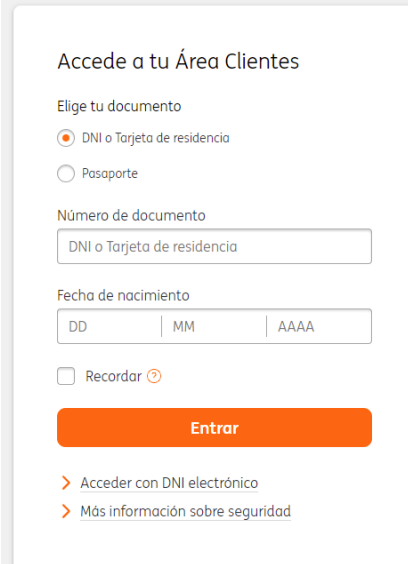
[Condiciones de uso](#) [Aviso de privacidad](#) [Ayuda](#) [Cookies](#) [Publicidad basada en intereses](#)

© 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):



Formulario de acceso a Área Clientes:

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

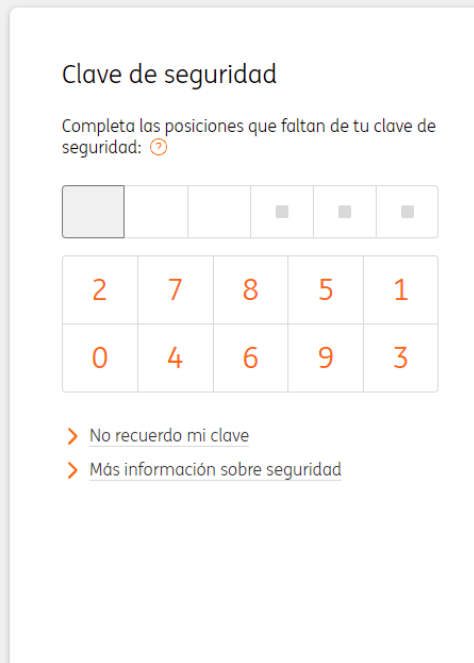
[Acceder con DNI electrónico](#)

[Más información sobre seguridad](#)

Los datos de DNI-Tarjeta de residencia-Pasaporte y fecha de nacimiento 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:



Formulario de clave de seguridad:

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.