**Ejercicios UD 8: Javascript**

1. Crear un script en un fichero externo que muestre una ventana con un mensaje de texto. No olvides añadir comentarios al código.
2. Modifica el ejercicio anterior, de manera que el mensaje que se muestra al usuario se almacene en una variable llamada mensaje y el funcionamiento del script sea el mismo.
3. Crear un array llamado meses y que almacene el nombre de los doce meses del año. Mostrar por pantalla los doce nombres utilizando la función alert().
4. A partir del siguiente array que se proporciona:

**var valores = [true, 5, false, "hola", "adios", 2];**

* 1. Determinar cuál de los dos elementos de texto es mayor
  2. Utilizando exclusivamente los dos valores booleanos del array, determinar los operadores necesarios para obtener un resultado true y otro resultado false
  3. Determinar el resultado de las cinco operaciones matemáticas realizadas con los dos elementos numéricos

1. Completar las condiciones de los if del siguiente script para que los mensajes de los alert() se muestren siempre de forma correcta.



1. El cálculo de la letra del Documento Nacional de Identidad (DNI) es un proceso matemático sencillo que se basa en obtener el resto de la división entera del número de DNI y el número 23. A partir del resto de la división, se obtiene la letra seleccionándola dentro de un array de letras. El array de letras es:

**var letras = ['T', 'R', 'W', 'A', 'G', 'M', 'Y', 'F', 'P', 'D', 'X', 'B', 'N', 'J', 'Z', 'S', 'Q', 'V', 'H', 'L', 'C', 'K', 'E', 'T'];**

Por tanto si el resto de la división es 0, la letra del DNI es la T y si el resto es 3 la letra es la A. Con estos datos, elaborar un pequeño script que:

* 1. Almacene en una variable el número de DNI indicado por el usuario y en otra variable la letra del DNI que se ha indicado. (Pista: si se quiere pedir directamente al usuario que indique su número y su letra, se puede utilizar la función prompt())
  2. En primer lugar (y en una sola instrucción) se debe comprobar si el número es menor que 0 o mayor que 99999999. Si ese es el caso, se muestra un mensaje al usuario indicando que el número proporcionado no es válido y el programa no muestra más mensajes.
  3. Si el número es válido, se calcula la letra que le corresponde según el método explicado anteriormente.
  4. Una vez calculada la letra, se debe comparar con la letra indicada por el usuario. Si no coinciden, se muestra un mensaje al usuario diciéndole que la letra que ha indicado no es correcta. En otro caso, se muestra un mensaje indicando que el número y la letra de DNI son correctos.

1. El factorial de un número entero n es una operación matemática que consiste en multiplicar todos los factores:

Así, el factorial de 5 (escrito como 5!) es igual a: Utilizando la estructura for, crear un script que calcule el factorial de un número entero.

1. Escribir el código de una función a la que se pasa como parámetro un número entero y devuelve como resultado una cadena de texto que indica si el número es par o impar. Mostrar por pantalla el resultado devuelto por la función.
2. Definir una función que muestre información sobre una cadena de texto que se le pasa como argumento. A partir de la cadena que se le pasa, la función determina si esa cadena está formada sólo por mayúsculas, sólo por minúsculas o por una mezcla de ambas.
3. Definir una función que determine si la cadena de texto que se le pasa como parámetro es un palíndromo, es decir, si se lee de la misma forma desde la izquierda y desde la derecha. Ejemplo de palíndromo complejo: "La ruta nos aporto otro paso natural"
4. A partir de la página web proporcionada y utilizando las funciones DOM, mostrar por pantalla la siguiente información:
   1. Número de enlaces de la página
   2. Dirección a la que enlaza el penúltimo enlace
   3. Numero de enlaces que enlazan a *http://prueba*
   4. Número de enlaces del tercer párrafo
5. Completar el código JavaScript proporcionado para que cuando se pinche sobre el enlace se muestre completo el contenido de texto. Además, el enlace debe dejar de mostrarse después de pulsarlo por primera vez. En este ejercicio, sólo se debe saber que al pinchar sobre el enlace, se ejecuta la función llamada *muestra()*.
6. Completar el código JavaScript proporcionado para que se añadan nuevos elementos a la lista cada vez que se pulsa sobre el botón. Utilizar las funciones DOM para crear nuevos nodos y añadirlos a la lista existente. En este ejercicio, sólo se debe saber que al pinchar sobre el botón, se ejecuta la función llamada *anade()*.
7. A partir de la página web proporcionada, completar el código JavaScript para que:
   1. Cuando se pinche sobre el primer enlace, se oculte su sección relacionada
   2. Cuando se vuelva a pinchar sobre el mismo enlace, se muestre otra vez esa sección de contenidos
   3. Completar 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 (pista: propiedad innerHTML)
8. Completar el código JavaScript proporcionado para que:
   1. 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 página. Para mostrar los mensajes, utilizar la función *muestraInformacion()* deduciendo su funcionamiento a partir de su código fuente.
   2. Al pulsar cualquier tecla, el mensaje mostrado debe cambiar para indicar el nuevo evento y su información asociada
   3. 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, el color de fondo debe ser azul (#CCE6FF). Al volver a mover el ratón, el color de fondo vuelve a ser blanco.



1. Crear un script que informe al usuario en que zona de la pantalla ha pulsado el ratón. Las zonas definidas son las siguientes: izquierda arriba, izquierda abajo, derecha arriba y derecha abajo. Para determinar el tamaño de la ventana del navegador, utilizar la función *tamanoVentanaNavegador()* proporcionada.
2. Crear una página web (con Bootstrap4) que contenga una cabecera, un pie de página y un formulario de registro que solicite, al menos, los siguientes datos:
   * + - 1. nombre
         2. apellidos
         3. DNI (8 cifras y 1 letra, juntos en el mismo campo)
         4. correo electrónico
         5. contraseña (con petición por duplicado)
         6. fecha de nacimiento
3. Para la página creada en el ejercicio 17, coloca junto los campos de contraseña un icono con un ojo. Crea una función Javascript que muestre el valor del campo de contraseña cuando se pase el ratón sobre dicho icono.
4. Para la página creada en el ejercicio 17, implementar un método que será llamado cuando se pincha en el botón de enviar en el formulario. Este método realizará un procesamiento previo de los datos del formulario antes de que se envíen al servidor. En el cuerpo de este método solamente deben ir las llamadas a los métodos que se van a implementar en los siguientes ejercicios.

A partir de aquí, vamos a comprobar todos los campos del formulario, por lo que en todos los ejercicios, cuando un dato del formulario sea incorrecto, se debe cambiar la apariencia del campo (por ejemplo, ponerle el borde más grueso y rojo, y que la fuente sea también roja) y colocar un aviso del error junto al campo en cuestión.

1. Crea el método pasarNombreApellidosAMayusculas() que, como su propio nombre indica, debe tomar los datos de los campos nombre y apellidos y convertirlos a una cadena entera en mayúsculas.
2. Crea el método comprobarDNI(), que comprobará que la letra del DNI es la correcta. Además, dejará la letra en mayúscula, aunque se haya escrito en minúscula. Puedes ayudarte del código realizado en el ejercicio 6 del boletín anterior.
3. Crea el método comprobarCorreo(), que comprobará que el correo introducido contiene una @ y que, a continuación de ella, hay al menos 1 letra o número, seguida de 1 punto y seguido de 3 letras.
4. Crea el método comprobarContrasena(), que comprueba que las dos contraseñas coinciden, tienen al menos 8 caracteres y combinan letras y números.

(*EXTRA: si quieres, puedes hacer el típico indicador de complejidad de contraseña)*

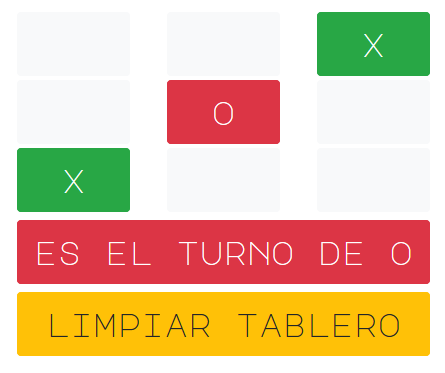
1. Crea el método comprobarEdad(), que comprueba que la persona es mayor de edad en función de la fecha de nacimiento introducida.

*EXTRA: si quieres, puedes hacer que todos estos métodos se lancen también cada vez que se retira el focus de cada campo del formulario. En este caso, solo habría que llamar al método que corresponda, no a todos.*

1. Se trata de iniciar el desarrollo de un juego de 3 en rayas, aunque por suerte solamente nos vamos a ocupar de colocar las fichas y cambiar el turno. No vamos a calcular si se gana o no se gana la partida.

* Hay dos jugadores: el jugador verde con la X y el jugador rojo con la O.
* Cuando se pincha sobre una casilla:
* si la casilla está vacía:
  + se debe cambiar la casilla para ponerla roja o verde (usando los colores danger o success de bootstrap) y rellenarla con X o con O, según corresponda
  + se debe cambiar el mensaje bajo el tablero, para indicar que le toca al siguiente jugador, cambiando el mensaje y el color de fondo del mismo
  + se debe cambiar el turno
* si la casilla está ocupada:
  + no se rellena la casilla
  + no se cambia el turno
  + se modifica el mensaje bajo el tablero para decir "Casilla ocupada". En este caso no se cambia el color del mensaje
* Cuando se pincha en el botón de "Limpiar Tablero":
* se deben limpiar todas las casillas, para que vuelvan a estar grises y sin letra
* se debe reiniciar el turno, para que le toque al verde, indicándolo en el mensaje bajo el tablero.

Esta es una captura de cómo estaría el tablero a mitad de la partida:



*EXTRA: si quieres, puedes implementar el control de la partida, comprobando tras cada jugada si existe un ganador.*

1. Se trata de crear un generador de colores aleatorios. Cada vez que se pulsa en el botón de "Generar Color" se debe:

* Generar un color RGB aleatorio
* Cambiar el color de fondo del body, para asignarle el color generado
* En la zona de mensajes, mostrar un mensaje del tipo badge-pill de color verde que indique el código RGB generado y el número de colores que se llevan generados.

Cuando se pincha en el botón "Limpiar Color", se debe:

* devolver el body a color blanco
* eliminar el mensaje
* reiniciar el contador de colores.

*Ayuda: recuerda que para generar un número aleatorio comprendido entre un mínimo y un máximo (ambos incluidos) puedes utilizar la siguiente instrucción: Math.floor(Math.random() \* (max - min + 1) ) + min*

Aquí tienes una captura de un momento de la ejecución:

