

Práctica 1. Practicando JS. Curso 23/24

En este documento se indica una batería de ejercicios a realizar para superar la parte de prácticas asociada a JavaScript. Se han de implementar y proporcionar en el formato siguiente:

Los datos que deban ser introducidos se solicitarán con el método **prompt()**. Los datos serán devueltos por pantalla en la página HTML (**document.write**) o en la consola (**console.log**). **IMPORTANTE:** Si no puedo visualizar de una de estas dos últimas formas los resultados de cada ejercicio, el ejercicio se dará como incorrecto.

La idea es que comentando y eliminando la marca de comentarios en las líneas que cargan los ficheros **XX.js** dentro del fichero **index.html**, se pueda ejecutar cada uno de los ejercicios de forma independiente.

Se requiere que cada fichero js tenga en las primeras líneas una descripción de cómo se va a resolver el ejercicio, así como un comentario en todas, o casi todas, indicando que es lo que se está realizando.

En alguno de los ejercicios se proporcionan ficheros sobre los que se ha de realizar la práctica, estos han de conservarse en la misma ubicación dentro de la estructura de archivos cuando se entreguen. (Caso del ejercicio 10).

La entrega se realizará en un archivo zip cuyo nombre es el DNI del alumn@ y conteniendo todos los ficheros dentro de la carpeta que actualmente se llama DNI_Alumn@ que será renombrada CON el DNI del alumn@.

Por favor, cualquier duda me la comentáis por el aula virtual o correo electrónico. Aquellas que vea que se repiten o son interesantes para el resto de los alumnos las subiré al foro para que todos tengáis la respuesta y además se tendrá en cuenta para la nota final, me refiero a las buenas preguntas no a preguntar por preguntar. 😊

Fecha de entrega de este primer boletín de prácticas hasta el 29 de Octubre de 2023. Obligatorio realizar hasta el ejercicio 11, es resto son opcionales para subir nota.

Ejercicio 01:

Investiga sobre los métodos de la clase **Date** y extrae 5 que te parezcan curiosos en su funcionamiento.

En el fichero 01.js indica que realiza cada uno de ellos junto con un ejemplo de su funcionamiento.

Ejercicio 02:

Crear una función edad, que dada la fecha de nacimiento como entrada calcule los meses que han transcurrido hasta hoy. También las semanas, días, horas, minutos y segundos.

Ejercicio 03:

Convertir la siguiente función en una Arrows Function



Ejercicio 04:

Explicar el funcionamiento del siguiente código e indicar el resultado mostrado por cada línea:

1. var contador =10;
2. console.log(contador++);
3. console.log(contador++);
4. console.log(contador+=10);
5. console.log(contador-=5);

Ejercicio 05:

Estudiar el uso de expresiones regulares en JavaScript y crear un formulario simple en el que indicando un DNI y pulsando un botón verificar, muestre un mensaje indicando si el formato del DNI es correcto o no.

No se busca que el DNI sea correcto, sino que cumpla el formato 99999999X.

Referencia: uso de Regex y match().

Ejercicio 06:

Dado el siguiente array de valores y haciendo uso de los comandos **forEach** y **map** muestre en pantalla un mensaje como el siguiente para cada uno de los elementos del array. (Realizar con ambos comandos)

Mensaje que mostrar: "El precio de la fruta FFFFFFFF es de EEEEE € por kilo"

```
const frutas = [
  { nombre: 'Manzana', precio: 2.500 },
  { nombre: 'Pera', precio: 1.700 },
  { nombre: 'Ciruela', precio: 4.300 },
  { nombre: 'Uva', precio: 2.200 },
  { nombre: 'Melocotón', precio: 3.50 },
  { nombre: 'Sandia', precio: 0.500},
  { nombre: 'Tomate', precio: 0.900},
  { nombre: 'Melón', precio: 1.800}
];
```

Ejercicio 07:

Ampliar el ejemplo visto en clase de sumar y restar (**12.js**) para desarrollar una función que se llame Calcular en vez de Operar y la cual implemente además de la suma y la resta, las operaciones de multiplicar (**multi**), dividir (**divi**), raíz cuadrada(**racu**), potencia(**pote**) y factorial(**fact**).

Nota: En el caso de raíz cuadrada y factorial, realizar el cálculo para ambos números y mostrar ambos resultados o solicitar por pantalla la operación y dependiendo la que sea entonces pedir uno o dos valores.

Ejercicio 08:

Realizar una función que calcule la letra de DNI. El único requisito para cumplir es que la cadena de entrada esté compuesta de 1 a 8 caracteres numéricos. Si son menos de 8 se rellenará por la izquierda de ceros hasta completar la longitud de 8 caracteres y se añadirá la letra, teniendo el DNI un total de 9 caracteres.

Ejercicio 09:

Hacer que se muestre cada 4 segundos la hora en formato HH:MM:SS:mmmm

Indicación: Hacer uso de setTimeout.

Ejercicio 10:

Uso de **form**. En carpeta Ejercicio_10 se detalla el código a utilizar.

Para este ejercicio se solicita que implementemos una función que realiza el siguiente calculo:

$\text{CalcuPeso} = 50 + ((\text{altura} - 150) / 4) * 3 + (\text{Edad} - 20) / 4$, este resultado se ha de multiplicar por 0.9 si es mujer.

Nota: Existen algunas etiquetas en el código HTML que no están bien puestas así que tenéis que arreglarlas.

Ejercicio 11: Jugar a pares y nones.

Implementar un formulario (página HTML) tomando como base el ejercicio anterior, en el que se solicite por pantalla un campo **Importe de apuesta** y un campo **par o non** (puede ser un texto o un radio button).

La funcionalidad consiste en:

- Partimos de un saldo que se mostrará en pantalla y que viene precargado en la página HTML inicialmente, por ejemplo 100€.
- Realizamos apuestas indicando el importe a apostar (nunca puede ser superior al saldo) y si la suma de los dedos extendidos de dos manos (suponemos 5 dedos en cada una de ellas) será par o impar (non).
- Si coincide el valor predicho, con la paridad de los dedos extendidos, se suma al saldo que tengamos el importe apostado por 3, en caso contrario se resta el importe apostado del saldo.
- Siempre que realicemos una apuesta el valor de saldo ha de actualizarse con las ganancias o pérdidas generadas.
- Se limpian los campos y queda preparado para volver a apostar.

Ejercicio 12:

Se han de solicitar por pantalla dos números del 1 a 10 para devolver todos los productos del primer número y sus dos posteriores por los números desde 1 hasta el segundo número.

Es decir, si informamos 2 y 5 ha de devolver los siguientes productos con sus resultados: $2*1$, $2*2$, $2*3$, $2*4$, $2*5$, $3*1$, $3*2$, $3*3$, $3*4$, $3*5$, $4*1$, $4*2$, $4*3$, $4*4$ y $4*5$.

Ejercicio 13:

Al comienzo del programa crear de forma automática un array con todos los números pares del 2 a 30. Solicitar un número por pantalla entre 2 y 30, asegurarse que está en ese intervalo, y devolver en un array los valores que son igual o superior a ese número y en otro los valores inferiores.

Ejercicio 14:

Haciendo uso de todos los métodos de la clase array, se ha de componer la siguiente frase:

Los condicionantes son que los siguientes métodos se deben emplear al menos 2 veces (`pop()`, `push()`, `unshift()`, `shift()`) y el método `reverse()` solo una vez.

El array final ha de ser idéntico al siguiente:

`["In", "a", "place", "in", "La", "Mancha", "whose", "name", "I do", "not", "want to", "remember"]`

Ejercicio 15:

Crear un array de dos dimensiones en base a dos datos que se solicitan por pantalla. Rellenar el array de forma descendente desde la posición (0,0) hasta la posición (m,n). Ejemplo:

9	8	7
6	5	4
3	2	1