

Los ejercicios son para que practiquéis vosotros. **Solo se entregarán las prácticas** indicando en la tarea la URL donde encontrar el desarrollo en GitHub. Es obligatorio realizarlos con el entorno de desarrollo que se ha definido al inicio del curso.

Para organizarlo mejor, vamos a crear una página home (index.html) y ahí añadiremos enlaces a cada una de las prácticas de forma que se pueda navegar desde esa página a cada una de las prácticas que hacemos.

1. Ejercicios sencillos:

- 1.1. Crea una función tradicional que reciba un número y retorne su cuadrado. Luego, usa esa función para mostrar el cuadrado de 5 en la consola.
- 1.2. Crea una función anónima autoinvocada que defina una variable llamada contador y la incremente en cada invocación. Luego, intenta acceder a la variable contador fuera de la función y observa lo que sucede.
- 1.3. Crea una función autoinvocada que reciba dos parámetros: una cadena de texto y un número. Dentro de la función, concatena la cadena con el número y muestra el resultado en consola.
- 1.4. Crea una función flecha que sirva para devolver verdadero si un número es par o falso si no lo es. Asigna esa función a una variable y prueba la función para ver que responde correctamente.

2. Ejercicios de uso avanzado:

- 2.1 Crea un bucle que recorra un array de números y calcule la suma de todos ellos de forma iterativa y recursiva. ¿Qué solución crees que es mejor?
- 2.2 Tienes un array de objetos que representan productos. Usa `forEach()` para incrementar en un 10% el precio de cada producto.

`const productos = [{ nombre: "Camiseta", precio: 20 }, { nombre: "Pantalón", precio: 30 }, { nombre: "Zapatos", precio: 50 }];`
- 2.3. Usa una combinación de los métodos `map()`, `reduce()` y `filter()` para procesar un array de objetos. Tienes un array de personas con sus nombres y edades, filtra a las personas mayores de 18, multiplica su edad por 2, y luego suma todas las edades.

```
const personas = [ { nombre: "Juan", edad: 17 }, { nombre: "Ana", edad: 22 }, { nombre: "Pedro", edad: 19 }, { nombre: "Laura", edad: 16 } ];
```

Práctica 1 (Obligatoria)

Crea una aplicación web que pida al usuario palabras continuamente hasta que el popup se deje vacío o se cancele.

A continuación, se eliminarán todas las palabras repetidas y además se ordenarán en español, pero en orden inverso (de la Z a la A) y se mostrará el resultado por pantalla.

Utilizar funciones flecha si se considera apropiado.

Ayuda: [cancelar el popup devuelve null](#).

Práctica 2 (Obligatoria)

Crea una aplicación web que pida al usuario palabras continuamente hasta que el popup se deje vacío o se cancele.

Crear una función que recibirá todas las palabras y devolverá un mapa que contenga como clave cada palabra y cuyo valor será el número de veces que aparece esa palabra en el array. Mostrar los resultados por pantalla.

Ayuda: Como en las 2 prácticas se pide introducir las palabras y devolver el resultado de la misma forma, podéis sacar esa parte del código a un script común que utilicen las 2 prácticas (main.js).

Práctica 3 (Opcional)

El método filter de los arrays permite indicar una función callback, para aplicar un filtro a los elementos del array. [Esta parte es sólo a efectos informativos, no hace falta que hagamos nada con el método filtro de los arrays].

Es una buena práctica para aprender a implementar funciones callback tratar de crear nuestras propias funciones.

Crea una función que se llame filtro y que reciba un array y una función callback. La función callback se entenderá que sólo tiene un parámetro. Esta función se aplicará a cada elemento del array y el resultado volverá a guardarse en el mismo array.

Por ejemplo: Array [1,2,3,4]

función = multiplicar por 2

Resultado: [2, 4, 6, 8]

Crear otros dos casos:

- Dado un array de palabras en minúsculas, pasarlas a mayúsculas.
- Dado un array de números, calcular el factorial de cada uno.
(<https://es.wikipedia.org/wiki/Factorial>)