

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.

### **Ejercicios sencillos:**

**Ej1:** Crea una página HTML que permita al usuario ingresar su nombre en un campo de texto. Al hacer clic en un botón "Guardar", guarda el nombre en una cookie con una duración de 7 días. Al cargar la página, si ya existe la cookie, muestra un mensaje de bienvenida con el nombre del usuario en lugar del campo de texto.

*Nota: Los ejercicios de cookies solo funcionarán si los ejecutáis con liveserver. Si los ejecutáis desde local se usa el protocolo file en vez de http, y no funcionará.*

**Ej2:** Crea una función divide(a, b) que tome dos números y devuelva el resultado de dividir a por b. La función debe lanzar un error si b es 0 con el mensaje "No se puede dividir por cero". Luego, llama a la función dentro de un bloque try-catch y muestra un mensaje en la consola si ocurre un error.

**Ej3:** Crea una función simulaProceso(exito) que devuelva una promesa. La promesa debe resolverse con el mensaje "Proceso exitoso" si exito es true, y rechazarse con el mensaje "Hubo un error" si exito es false. Prueba la función llamándola con un bloque then-catch y con async-await usando tanto casos exitosos como casos fallidos.

**Ej4:** Crea un módulo calculadora.js que exporte funciones básicas de una calculadora: suma, resta, multiplica y divide. Importa las funciones en otro archivo y usa cada una con valores de prueba.

**Ej5:** Crea tres funciones que devuelvan promesas:

- primeraTarea: devuelve el valor 10.
- segundaTarea: recibe el valor de primeraTarea, lo multiplica por 2 y devuelve el nuevo valor.
- terceraTarea: recibe el valor de segundaTarea, le suma 5 y devuelve el nuevo valor.

Encadena estas promesas usando then y muestra el resultado final en la consola.

### **Práctica 1 (Obligatoria)**

Crea una página que muestre el número de visitas que has realizado en la página. La cookie que creas caducará en un mes. Si se visita la página más de 10 veces, el contador se pondrá a cero y se borrará la cookie existente.

## **Práctica 2 (Obligatoria)**

Crea una función que reciba un número en milisegundos y genere un temporizador que escriba el texto "Tiempo concluido" cuando pasen esos milisegundos.

Realmente la función no escribe, genera una promesa cuyo resultado es el texto nombrado.

En caso de que no se cumpla la promesa, el resultado será el texto: "El tiempo no va bien".

Se decidirá que la promesa no se ha cumplido cuando pase el doble de tiempo que el indicado.

Crea un programa que invoque esta función (con 5 segundos) y escriba por pantalla (elemento HTML) los resultados de la promesa.

Nota: para el temporizador podéis usar **setTimeout**:

El método **setTimeout** tiene dos parámetros: el primero es la función que se ejecutará cuando se cumpla el tiempo, el segundo es un valor numérico que indica el tiempo a esperar en milisegundos antes de ejecutar el código de la función. Ejemplo:

```
setTimeout(()=>alert("Hola"),5000);
```

El método **setTimeout** devuelve un número que identifica al temporizador en sí. Ese número se puede usar para cancelar el temporizador mediante **clearTimeout**. Ejemplo:

```
var templ=setTimeout(()=>alert("Hola"),5000);
```

```
clearTimeout(templ);
```

Tal cual está escrito el código anterior, jamás aparecerá el mensaje **Hola** porque se cancela el temporizador inmediatamente (no habrán pasado los 5 segundos).

## **Práctica 3 (Opcional)**

**IMPORTANTE- En esta práctica se debe hacer el programa principal dentro de la página HTML, en una etiqueta Javascript. El objetivo simplemente es para que poner en práctica el EXPORT / IMPORT de módulos javascript.**

Crea un archivo llamado temporizador.js, el cual será un módulo que podremos importar a nuestro código javascript principal (en la página HTML)

Este módulo constará de dos funciones:

- La función **temporizador**, creada en la práctica anterior
- La función **cuenta** (es obligatorio que esta función sea ASYNC). Esta función será capaz de escribir una cuenta atrás y mostrarla como elemento HTML. Los parámetros son:
  - El número con el que se inicia la cuenta atrás

- El elemento en el que escribiremos la cuenta atrás. Por defecto se utilizará el body.
  - El intervalo en milisegundos en el que cambia cada número. Por defecto valdrá 1 segundo.
  - Una función callback, que se ejecutará cuando la cuenta finalice.
- Crea, además, una aplicación web que cree 2 párrafos. En el primero se contará hacia atrás desde el 6, de segundo en segundo. En el segundo se contará desde el 60 pero cada número se moverá de décima en décima.

Además, al llegar a cero en el segundo párrafo se escribirá la palabra **"Fin"**. Este será el resultado final:

0

Fin