

Counter

Explicación del Ejercicio

¿Qué vamos a hacer?

Vamos a implementar un contador interactivo en JavaScript que permitirá aumentar, disminuir y reiniciar un número en pantalla. Para ello, utilizaremos tres botones:

1. Uno que disminuirá el valor del contador.
2. Uno que reiniciará el contador a cero.
3. Uno que aumentará el valor del contador.

Deberás hacer el fork del siguiente link:

¿Dónde lo vamos a hacer?

El proyecto será forkeado desde GitHub.

<https://github.com/Andreabcm/js-counter.git>

Trabajaremos en dos archivos de JavaScript ubicados en la carpeta src/js:

1. Archivo `counter.js`, para la lógica que manejará el estado del contador (aumentar, disminuir y reiniciar).
2. Archivo `events.js` para disparar los eventos cuando se haga clic en los botones.

El HTML y CSS ya están creados.

¿Cómo lo hacemos?

El contador comenzará con un número 0. Tendrás tres botones: uno para disminuir el número, otro para reiniciarlo a 0, y otro para aumentarlo. La lógica del contador debe escribirse en el archivo `counter.js`, y este archivo debe controlar la interacción de los botones mediante eventos. Los eventos los escribimos en el archivo `events.js`.

En la **Parte I**, se implementarán las funciones que permiten aumentar, disminuir y reiniciar el valor del contador. Cada botón tendrá un evento asociado que activará la función correspondiente cuando sea clicado.

En la **Parte II**, se refactorizará la función de "disminuir" para evitar que el contador baje de 0, es decir, no habrá valores negativos.

¿Con qué lo hacemos?

Utilizaremos JavaScript y un editor de texto como Visual Studio Code.

Objetivo del Ejercicio

El objetivo principal es que logres **interactuar con el DOM** y accionar **Eventos** para crear un contador que responda a la interacción con tres botones, utilizando JavaScript. Además, se busca **refactorizar** la función de disminución para que el contador no baje de cero en la segunda parte.

Objetivos Específicos

1. **Parte I:**
 - Implementar las funciones necesarias en `counter.js` para aumentar, disminuir y reiniciar el contador.
 - Vincular los eventos de clic a los botones correspondientes en `events.js`.
 2. **Parte II:**
 - Refactorizar la función de disminución en `counter.js` para que no permita que el contador tenga valores negativos.
 - Asegurarse de que el comportamiento de los botones se mantenga correctamente tras la refactorización.
-

Criterios de Evaluación

1. **Implementación de la lógica:** El estudiante debe demostrar una comprensión clara sobre cómo gestionar eventos y actualizar la UI.
 2. **Funcionalidad completa en `counter.js` y `events.js`:** Todo el código JS debe estar concentrado en el archivo `counter.js` y `events.js`, sin modificar otros archivos.
 3. **Organización del código:** El código debe estar organizado de manera que sea fácil de leer y entender.
 4. **Refactorización correcta:** Se evaluará si la función `decrease` impide correctamente los valores negativos.
-

Paso a Paso

Parte I: Crear el Proyecto

- Forkea el repositorio proporcionado y clónalo en tu máquina local.

Parte II: Crear la lógica del contador

Paso 1: Visualizar el contador en la página web

1. El archivo `index.html` y el `style.css` ya están creados y contiene tres botones: uno para aumentar, otro para disminuir, y otro para reiniciar el contador. También incluye un elemento para mostrar el valor actual del contador.
 - No es necesario modificar el HTML ni el CSS.
 - Visualiza la página web con Live Server y observa el contador, de momento no tiene aún la lógica implementada ni está funcional (al hacer click en los botones no debe ocurrir nada).

Paso 2: Implementar la lógica y el evento en `counter.js`

1. Abrir el archivo `counter.js`
 - Este archivo estará ubicado dentro de la carpeta `src/js/`.
2. Escribir las funciones para controlar el contador:

Implementa tres funciones:

- Implementa la lógica para **aumentar** el valor del contador en la función **`add()`**.
- Implementa la lógica para **reiniciar** el valor del contador en la función **`reset()`**.
- Implementa la lógica para **disminuir** el valor del contador en la función **`decrease()`**.
- Cada función debe actualizar el valor del contador en el DOM, de modo que el número se muestre en tiempo real en la pantalla.

Paso 3: Vincular los eventos de click en `events.js`

1. Abrir el archivo `events.js`
 - Este archivo también estará ubicado dentro de la carpeta `src/js/`.
2. Agregar eventos a los botones:
 - Usa `document.getElementById` para seleccionar los botones.
 - Utiliza `addEventListener` para vincular el evento de clic a cada botón, y así ejecutar las funciones de aumentar, disminuir o reiniciar según corresponda.

Paso 4: Guardar y Probar

- Guarda todos los cambios y asegúrate de que el contador funcione correctamente.

Parte III: Refactorización para evitar valores negativos

Paso 1: Refactorizar la función de disminución en `counter.js`

1. Abrir el archivo `counter.js`
 - Nos enfocaremos en la función que disminuye el valor del contador.
2. Modificar la función de disminuir

- Agrega una condición que verifique si el valor actual del contador es mayor que 0 antes de restar. Si el valor ya es 0, no se reducirá más.
- Esto evitará que el contador muestre valores negativos.

Paso 2: Verificar el funcionamiento

- Asegúrate de que el botón de disminución no permita reducir el contador por debajo de 0.
- El resto de los botones (aumentar y reiniciar) deben seguir funcionando correctamente tras la refactorización.

Resumen:

- La refactorización se realizará únicamente en `counter.js`, asegurando que el valor del contador no pueda ser negativo.
-

Recursos y Herramientas

- **Editor de Texto:** [Visual Studio Code](#)
 - **Documentación de JavaScript:**
 - [JavaScript Events](#)
 - [JavaScript Functions](#)
-

Entrega del Ejercicio

- **Formato de entrega:**
 - La entrega deberá realizarse a través de **Discord**, enviando a las formadoras el **enlace al repositorio de GitHub**.
-

Este ejercicio te permitirá desarrollar una mejor comprensión de cómo manipular el DOM, manejar eventos y aplicar la lógica de programación para interactuar con los elementos de una página web.