

T2Actividad6: Suma de números ingresados

Ejercicio: Sumar números introducidos por el usuario

Enunciado

El objetivo del ejercicio es:

1. Crear una función `sumInput()` que:
 - Solicite valores al usuario mediante un cuadro de entrada (`prompt`) y los almacene en un array.
 - Detenga la solicitud de valores si el usuario:
 - Introduce una cadena vacía.
 - Introduce un valor no numérico.
 - Presiona "Escape".
 - Asegurarse de que el número `0` sea considerado un valor válido.
 2. Calcular y devolver la suma de los valores introducidos.
 3. Mostrar el resultado de la suma en un mensaje emergente (`alert`).
-

Análisis del Problema

Para resolver este problema, seguimos los siguientes pasos:

1. **Solicitar valores:**
 - Usar un bucle `while (true)` para solicitar números al usuario hasta que se cumpla una condición de salida.
 - Validar cada entrada para asegurarse de que sea un número o determinar si es momento de finalizar la solicitud.
 - Convertir la entrada a número y almacenarla en un array si es válida.
 2. **Calcular la suma:**
 - Usar el método `reduce()` del array para sumar los valores acumulados.
 3. **Mostrar el resultado:**
 - Retornar la suma calculada y mostrarla al usuario con un mensaje de `alert`.
-

Solución Propuesta

Se implementó una solución en JavaScript que utiliza:

- La función `prompt` para solicitar valores al usuario.
 - Un array para almacenar los números válidos introducidos.
 - El método `reduce()` para calcular la suma de los valores.
 - Validaciones para detener la solicitud de valores cuando corresponda.
-

Código Fuente

HTML

El código puede ejecutarse directamente en un archivo `.html` sin elementos adicionales.

JavaScript

El código JavaScript es el siguiente:

```

javascript
Copiar código
function sumInput() {
    let numeros = [];
    let entrada;

    // Pedir valores al usuario hasta que introduzca un valor no numérico o vacío
    while (true) {
        entrada = prompt("Introduce un número (o deja en blanco/Escape para termina
r):");

        // Salir si se presiona "Escape", se deja vacío o no es un número
        if (entrada === null || entrada.trim() === "" || isNaN(entrada)) break;

        // Convertir la entrada a número y agregarla al array
        numeros.push(parseFloat(entrada));
    }

    // Calcular la suma de los números en el array
    const suma = numeros.reduce((acumulado, num) => acumulado + num, 0);

    // Retornar la suma
    return suma;
}

// Ejecutar la función y mostrar el resultado
const resultado = sumInput();
alert(`La suma de los números introducidos es: ${resultado}`);

```

Pruebas Realizadas

1. Caso con números válidos:

- Entrada: 10, 20, 30 .
- Salida esperada: La suma de los números introducidos es: 60 .

2. Caso con un número 0 :

- Entrada: 10, 0, 20 .
- Salida esperada: La suma de los números introducidos es: 30 .

3. Caso con valores no numéricos:

- Entrada: 10, abc, 20 .
- Salida esperada: La suma de los números introducidos es: 10 .

4. Caso con cadena vacía:

- Entrada: 10, (vacío) .
- Salida esperada: La suma de los números introducidos es: 10 .

5. Caso de Escape:

- Entrada: 10, 20, (Escape) .
- Salida esperada: La suma de los números introducidos es: 30 .

Conclusión

El programa cumple con los requisitos del enunciado:

- Solicita valores válidos y los almacena en un array.
- Detiene la solicitud de valores cuando el usuario introduce un valor no numérico, vacío o presiona "Escape".
- Calcula y muestra correctamente la suma de los valores introducidos.