

Examen DAW UT1

Enunciado

Al cargar la página:

1. Pedir al usuario una lista de números separados por comas (ej. `4, 5, 10, 2, 7, 8, 5`).
2. Validar la entrada: todos los valores deben ser números. Si no, mostrar un mensaje de error en la página.
3. Procesar los números para mostrar en la página:
 - **Lista original** (como array).
 - **Números pares** y **números impares** por separado.
 - **Números únicos** (eliminar duplicados).
 - **Lista ordenada de mayor a menor**.
 - **Suma** de todos los números.

Requisitos técnicos

1. Implementar la función:
 - **function name:** `analyzeNumbers(nums)`
 - `nums` es un array de números.
 - Devuelve un objeto con las propiedades:

```
{  
  original: [...],  
  pares: [...],  
  impares: [...],  
  únicos: [...],  
  ordenDesc: [...],  
  suma: Number  
}
```

2. Entrada:

- Pedir los números con `prompt()`.
- Convertir a enteros (ignorar espacios).
- Mostrar mensaje de error en DOM si hay algún valor no numérico o array vacío.

3. Salida:

- Crear nodos DOM dinámicamente para mostrar en formato lista
 - Lista original
 - Pares
 - Impares
 - Únicos
 - Lista ordenada descendente
 - Suma total.
- No usar innerHTML.

Ejemplo de ejecución

Entrada del usuario:

4,5,10,2,7,8,5

Salida en la página:

Lista original: [4,5,10,2,7,8,5]

Números pares: [4,10,2,8]

Números impares: [5,7,5]

Números únicos: [4,5,10,2,7,8]

Lista ordenada (desc): [10,8,7,5,5,4,2]

Suma total: 41

Entrada inválida:

4,x,2

Mostrar mensaje en la página: `Entrada inválida: 'x' no es un número.`

Criterios de corrección

- Funcionalidad completa (`analyzeNumbers`): **60%**
- Validación de entrada y mensajes claros en DOM: **20%**
- Uso correcto de DOM: **15%**
- Código legible, comentado y use strict: **5%**

Entregable

Archivos a entregar, un zip con: `main.js`, `main.html`