CONTENIDO

INTRODUCCIÓN

Ejercicio 1

- Crea un script que pida introducir tres números y que los ordene de mayor a menor.
- Realiza las validaciones pertinentes evitando datos no numéricos o vacíos.

- Investiga algún algoritmo de ordenación que permita optimizar el tiempo de ejecución (minimizar el número de comparaciones) y desarrolla el script basándote en él.

- Recomendable: utilización de mensajes por consola (console.log, info, etc)

Ejercicio 2:

- En este ejercicio se pretende practicar el uso de switch…case y bucles, para realizar cálculos aritméticos.
- Dado un número y una operación aritmética introducida por teclado (a través de prompt) se quiere hacer la operación introducida sucesivamente sobre dicho número-1, hasta llegar a 0 (0 no incluido).
- Valores válidos de la operación: “SUMA”, “RESTA”, “MULTIPLICACIÓN”, “DIVISÓN”.

Ejemplo:

  - Número introducido = 9

  - Operación = SUMA

  - Deberá realizar 9 + 8 + 7 + 6 + 5 + 4 + 3 + 2 + 1

EJERCICIO 3.
Cálculo de la letra del DNI.
El proceso sería dividir la parte numérica entre 23 y obtener su resto.
Dicho resto tendríamos que identificarlo con la posición de la siguiente lista de letras (empezando en 0). En obtener la letra.

[‘T’, ‘R’, ‘W’, ‘A’, ‘G’, ‘M’, ‘Y’, ‘F’, ‘P’, ‘D’, ‘X’, ‘B’, ‘N’, ‘J’, ‘Z’, ‘S’, ‘Q’, ‘V’, ‘H’, ‘L’, ‘C’, ‘K’, ‘E’, ‘T’];

Nota: El objetivo de este ejercicio es utilizar switch...case con el resto de la división y obtener los distintos casos de letras.

No deben usurarse métodos del objeto array ni elementos arrays. (Se estudiarán más adelante)

EJERCICIO 1

Crea un script que pida introducir tres números y que los ordene de mayor a menor. Realiza las validaciones pertinentes evitando datos no numéricos o vacíos.

Investiga algún algoritmo de ordenación que permita optimizar el tiempo de ejecución (minimizar el número de comparaciones) y desarrolla el script basándote en él.

Recomendable: utilización de mensajes por consola (console.log, info, etc.)

Algoritmo de ordenación:

```
function ordenacion() {
    let exp;
    if (num1 < num2) {
        exp = num2;
        num2 = num1;
        num1 = exp;
    }
    if (num2 < num3) {
        exp = num3;
        num3 = num2;
        num2 = exp;
    }
    if (num1 < num2) {
        exp = num2;
        num2 = num1;
        num1 = exp;
    }
    return `${num1} ${num2} ${num3}`
};
```

Validar entrada de datos:

```
let num1, num2, num3;

function comprobarNumeros() {
    while (true) {
        num1 = Number(prompt('Dame el primer numero: '));
        if (num1 === null || isNaN(num1) || num1 === 0) {
            alert('Introduce un numero valido...');
        } else break;
    }
    while (true) {
        num2 = Number(prompt('Dame el segundo numero: '));
        if (num2 === null || isNaN(num2) || num2 === 0) {
            alert('Introduce un numero valido...');
        } else break;
    }
    while (true) {
        num3 = Number(prompt('Dame el tercer numero: '));
        if (num3 === null || isNaN(num3) || num3 === 0) {
            alert('Introduce un numero valido...');
        } else break;
    }
}
```

Explicacion:
El algoritmo que utilizamos es el ordenamiento burbuja, que consiste en tener una variable como de reserva para que no se pierdan datos a la hora de intercambiarlos. Primero comprobamos el num1 y el num2, si el 1 es más pequeño, se mete el 2 en la variable de reserva para no perder el dato el 1 se pasa al 2 y el 1 al de reserva que seria el 2. Después comprobamos el 2 y el 3, es el mismo algoritmo. Finalmente repetimos el algoritmo nuevamente con el 1 y el 2 por si cuando hicimos el 2 y 3 cambió algo.

Hemos implementado una validación para la entrada de datos, para evitar cualquier cosa que no sea un numero ni un elemento vacío. Lo que hacemos es en un bucle while comprobamos cada dato individualmente, diciéndole que si es null o no es un numero o no es 0 que saque una alerta de 'número no valido' y continue para volver a preguntar ese dato

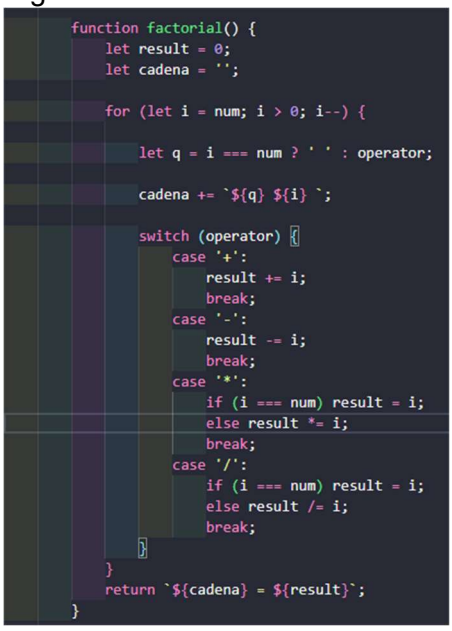
EJERCICIO 2

En este ejercicio se pretende practicar el uso de switch…case y bucles, para realizar cálculos aritméticos.
Dado un número y una operación aritmética introducida por teclado (a través de prompt) se quiere hacer la operación introducida sucesivamente sobre dicho número-1, hasta llegar a 0 (0 no incluido).
Valores válidos de la operación: "SUMA", "RESTA", "MULTIPLICACIÓN", "DIVISÓN".

Ejemplo:

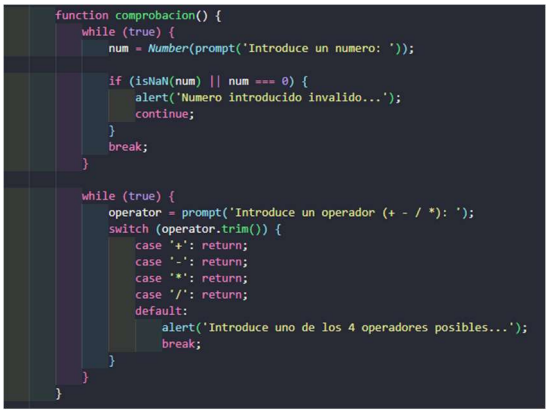
- Número introducido = 9
- Operación = SUMA
- Deberá realizar 9 + 8 + 7 + 6 + 5 + 4 + 3 + 2 + 1

Algoritmo de cálculo:



```
function factorial() {
    let result = 0;
    let cadena = '';

    for (let i = num; i > 0; i--) {

        let q = i === num ? ' ' : operator;

        cadena += `${q} ${i} `;

        switch (operator) {
            case '+':
                result += i;
                break;
            case '-':
                result -= i;
                break;
            case '*':
                if (i === num) result = i;
                else result *= i;
                break;
            case '/':
                if (i === num) result = i;
                else result /= i;
                break;
        }
    }
    return `${cadena} = ${result}`;
}
```

Validador de entrada de datos:



```
function comprobacion() {
    while (true) {
        num = Number(prompt('Introduce un numero: '));

        if (isNaN(num) || num === 0) {
            alert('Numero introducido invalido...');
            continue;
        }
        break;
    }

    while (true) {
        operator = prompt('Introduce un operador (+ - / *): ');
        switch (operator.trim()) {
            case '+': return;
            case '-': return;
            case '*': return;
            case '/': return;
            default:
                alert('Introduce uno de los 4 operadores posibles...');
                break;
        }
    }
}
```

Resultado:

$$20 + 19 + 18 + 17 + 16 + 15 + 14 + 13 + 12 + 11 + 10 + 9 + 8 + 7 + 6 + 5 + 4 + 3 + 2 + 1 = 210$$

Explicación:
Este código solicita al usuario un operador (+, -, *, /) y un número.
Primero verifica que el operador esté dentro de los permitidos y que el número no sea inválido. Luego, realiza un recorrido en un ciclo decreciente desde ese número hasta 1.
En cada etapa va formando una cadena con la operación entera
(por ejemplo: 5 * 4 * 3 * 2 * 1).
Paralelamente, va determinando el resultado utilizando el operador seleccionado.
Para los cálculos de * y /, la operación se inicia desde el primer número para que la inicialización a 0 de la variable no afecte al resultado.
Por último, muestra el resultado y la operación total en la pantalla por medio de la etiqueta ‘p’.

EJERCICIO 3

Cálculo de la letra del DNI.
El proceso sería dividir la parte numérica entre 23 y obtener su resto.
Dicho resto tendríamos que identificarlo con la posición de la siguiente lista de letras (empezando en 0). En obtener la letra.
[‘T’, ‘R’, ‘W’, ‘A’, ‘G’, ‘M’, ‘Y’, ‘F’, ‘P’, ‘D’, ‘X’, ‘B’, ‘N’, ‘J’, ‘Z’, ‘S’, ‘Q’, ‘V’, ‘H’, ‘L’, ‘C’, ‘K’, ‘E’, ‘T’];

Nota: El objetivo de este ejercicio es utilizar switch...case con el resto de la división y obtener los distintos casos de letras.

Validar entrada:

```
const posLetra = ['T', 'R', 'W', 'A', 'G', 'M', 'Y', 'F', 'P', 'D', 'X', 'B', 'N', 'J', 'Z', 'S', 'Q', 'V', 'H', 'L', 'C', 'K', 'E', 'T'];

let numDNI;
function comprobarEntrada() {
    while (true) {
        numDNI = prompt('Introduce los numeros del DNI: ');
        if (numDNI.length !== 8 || isNaN(numDNI)) {
            alert('Numero de DNI invalido...');
        } else break;
    }
}
```

Algoritmo:

```
function calculoDNI() {
    let result = posLetra[Number(numDNI % 23)];
    return result;
}

comprobarEntrada();
const unique = document.getElementById('unique');
unique.innerText = `Tu DNI es: ${numDNI}${calculoDNI()}`;
```

Resultado:

Tu DNI es: 51221474Z

Explicación:
Este código determina la letra del DNI. Primero, almaceno todas las letras posibles en un array. ComprobarEntrada() me permite garantizar que el usuario ingrese 8 números válidos. En la función calculoDNI(), realizo el módulo 23 del número y empleo el resultado como índice dentro del arreglo.De esta manera, obtengo la letra correspondiente. Finalmente, en el párrafo con ID "unique" muestro el DNI completo (número y letra).

BIBLIOGRAFIA

La documentación de MDN Web Docs, en la que encontré ejemplos prácticos y explicaciones claras acerca del funcionamiento de JavaScript.
https://developer.mozilla.org/es/docs/Web/JavaScript

El sitio web W3Schools, que utilicé para revisar ideas elementales sobre validaciones, bucles y estructuras como switch-case.
https://www.w3schools.com/js/

Una descripción simple del algoritmo de ordenación burbuja que encontré en EcuRed, que me sirvió para refrescar la memoria sobre como funcionaba este método.
https://www.ecured.cu/Ordenamiento_de_burbuja

Algunos ejemplos de algoritmos en JavaScript que hallé en GeeksforGeeks, pero en español.
https://www.geeksforgeeks.org/es/

Un artículo de Genbeta Dev que trata sobre cómo calcular la letra del DNI con un código, me inspiró para realizar el ejercicio 3.
https://www.genbeta.com/desarrollo