**Universidad Tecnológica Metropolitana**

**4F**

**ACTIVIDAD #2 PRACTICAS 2-4**

**Nombre alumno:**

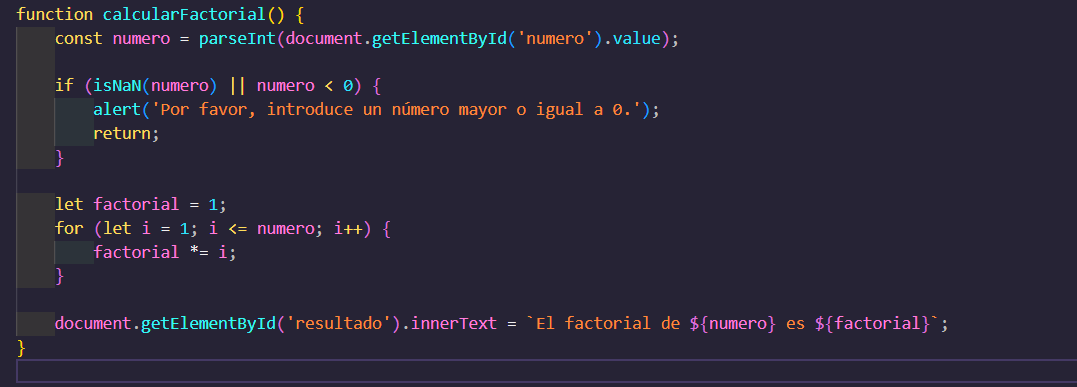
**Fernando David Sanchez Sacnhez**

**Nombre del profesor(a): Ruth Dominguez**

**Fecha de entrega: Mérida, Yucatán a martes 10 de septiembre de 2024**

En este documento veremos las actividades de la 2 a la 4 en la cual nos solicitaron lo siguiente:

1. Primero nos encargamos de desarrollar el script que nos solicitan en el cual nos solicitan calcular el valor Factorial de un numero mayor o igual a cero utilizando una función o método recursivo.



1. Utilizamos la función calcularfactorial (), Esta función se encarga de calcular la factorial de un número que el usuario introduce en un campo de entrada en la página web.



1. Para obtener el valor numérico utilizamos la siguiente función: Aquí, se utiliza document.getElementById('numero').value para obtener el valor que el usuario ha ingresado en el campo de entrada con el id número.

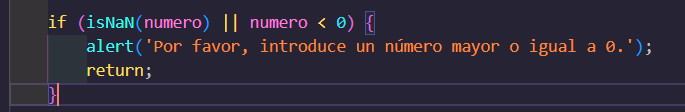
parseInt() convierte ese valor de tipo string a un número entero para que podamos realizar cálculos con él.



1. Para la Validación de numero utilizamos las siguientes funciones: isNaN(numero) verifica si el valor ingresado no es un número válido.

numero < 0 verifica si el número es negativo.

Si cualquiera de estas condiciones es verdadero, el programa muestra un mensaje de error en el div con el id resultado y se detiene la ejecución (return).



1. Inicializar factorial, Se inicializa una variable factorial en 1. Esta variable almacenará el resultado del cálculo de la factorial.

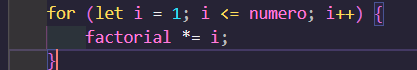


1. Se hace el cálculo factorial usando en bucle for:

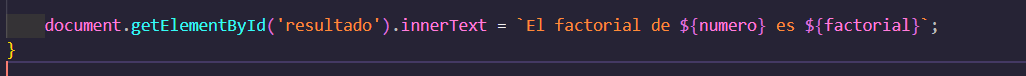
Este bucle for comienza en 1 y se ejecuta hasta que alcanza el valor del número ingresado.

En cada iteración, se multiplica el valor actual de factorial por el valor de i y se actualiza la variable factorial.

Esto sigue la lógica de que la factorial de un número n es el producto de todos los enteros desde 1 hasta n.



1. Mostrar el resultado: Finalmente, el resultado se muestra en el div con el id resultado. Se usa la interpolación de cadenas (con ${}) para incluir el número y su factorial en el mensaje.



Practica 3.

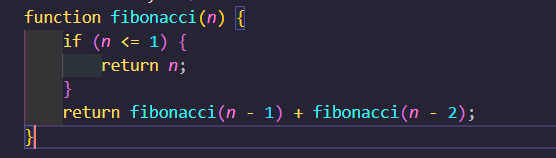
1. En esta actividad veremo como hacer una función de Fibonacci:

Función Fibonacci(n): Esta función es recursiva y se encarga de calcular el valor en la serie de Fibonacci para la posición n. La serie de Fibonacci se define como:

F(0) = 0

F(1) = 1

F(n) = F(n-1) + F(n-2) para n > 1.



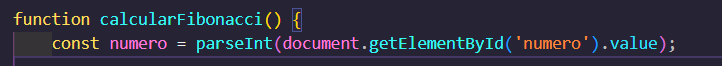
Base de la Recursión: if (n <= 1) { return n; }

Si n es 0 o 1, la función simplemente devuelve n. Esto es importante porque define los casos base de la recursión.

Recursividad: return fibonacci(n - 1) + fibonacci(n - 2);

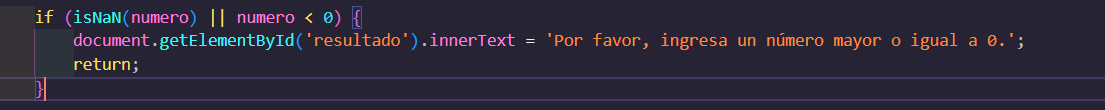
Para cualquier otro valor de n, la función se llama a sí misma dos veces: una vez con n-1 y otra vez con n-2. Los resultados de estas llamadas se suman para calcular el valor de F(n).

1. Función calcularFibonacci(): Esta función se encarga de interactuar con el usuario. Toma el número ingresado, valida que sea un número no negativo, y utiliza la función fibonacci(n) para calcular el resultado.



Aquí, document.getElementById('numero').value obtiene el valor del campo de entrada con el id numero. parseInt() convierte ese valor en un número entero.

1. Validación de numero



Se verifica si el valor ingresado es un número válido (no NaN) y si es mayor o igual a 0. Si no cumple con estos criterios, muestra un mensaje de error en el div con el id resultado y se detiene la ejecución.

1. Cálculo del número de Fibonacci:



i el número es válido, se llama a la función fibonacci(n) con el número ingresado, y el resultado se almacena en la variable resultado.

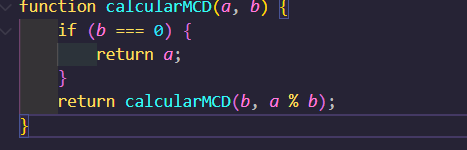
1. Mostrar resultado:



Finalmente, el resultado se muestra en el div con el id resultado. Se usa la interpolación de cadenas para construir el mensaje.

Actividad 4.

1. **Función calcularMCD(a, b)**: Esta función es recursiva y se encarga de calcular el Máximo Común Divisor (MCD) de dos números enteros a y b utilizando el algoritmo de Euclides. El MCD es el número más grande que divide exactamente a ambos números.



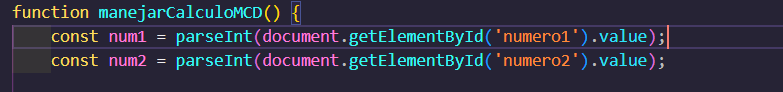
Base de la Recursión: if (b === 0) { return a; }

El algoritmo de Euclides establece que, si b es 0, el MCD es a. Esto se debe a que cualquier número dividido por 0 no tiene un resto definido, por lo que a es el divisor común más grande.

Recursividad: return calcularMCD(b, a % b);

Si b no es 0, la función se llama a sí misma con b y a % b (donde % es el operador de módulo que devuelve el resto de la división de a entre b). Este proceso se repite hasta que el segundo número (b) se convierte en 0, momento en el cual se devuelve a como el MCD.

1. **Función manejarCalculoMCD()**: Esta función interactúa con el usuario. Recoge los dos números introducidos, valida que sean números enteros, y utiliza calcularMCD para encontrar el MCD.

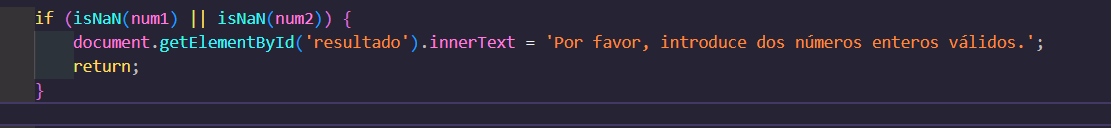


Obtener los valores:

Se utiliza document.getElementById('numero1').value y document.getElementById('numero2').value para obtener los valores de los campos de entrada con los ids numero1 y numero2.

parseInt() convierte esos valores de tipo string a números enteros para poder realizar cálculos con ellos.

1. Validación de los números



Se verifica si alguno de los valores ingresados no es un número (isNaN) y, si es así, se muestra un mensaje de error en el div con el id resultado y se detiene la ejecución (return).

1. Cálculo del MCD:



Si ambos números son válidos, se llama a calcularMCD(num1, num2) para obtener el MCD, y el resultado se almacena en la variable mcd.

1. Mostrar el resultado:



Finalmente, el resultado se muestra en el div con el id resultado. El mensaje utiliza interpolación de cadenas para incluir los números originales y el MCD calculado.