

Informe Breve: Diseño e Implementación de la Calculadora JavaScript

Contenido

1. Introducción	1
2. Decisiones de Diseño	2
1. Estructura del Código.....	2
Decidí usar un objeto llamado Operación para estructurar cada cálculo. De esta forma, pude definir las propiedades esenciales, como num1 y el operador, desde el inicio.	2
Utilicé prototype para agregar num2 después, ya que algunas operaciones, como la raíz cuadrada, solo necesitan un número. También agregué los métodos realizarOperacion y toString en el prototype para que todas las instancias pudieran compartirlos.....	2
2. Validación de Números	2
3. Manejo de Errores	2
4. Interfaz de Usuario	2
3. Desafíos Encontrados	3
1. Validación de Entradas unido con el Historial de Operaciones	3
2. Utilización de diferentes formas de crear funciones.	3
4. Conclusión	3

1. Introducción

En este proyecto, desarrollamos una calculadora en JavaScript capaz de realizar operaciones matemáticas básicas como suma, resta, multiplicación, división y cálculo de la raíz cuadrada. La meta principal fue crear algo funcional, fácil de usar y que manejara los errores de manera adecuada. También había que asegurarse de que el historial de operaciones solo guardara cálculos válidos, sin operaciones problemáticas.

2. Decisiones de Diseño

1. Estructura del Código

Decidí usar un objeto llamado Operación para estructurar cada cálculo. De esta forma, pude definir las propiedades esenciales, como num1 y el operador, desde el inicio.

Utilicé prototype para agregar num2 después, ya que algunas operaciones, como la raíz cuadrada, solo necesitan un número. También agregué los métodos realizarOperacion y toString en el prototype para que todas las instancias pudieran compartirlos.

2. Validación de Números

Era imprescindible evitar que los usuarios ingresaran datos no numéricos, así que implementé validarInput. Esta función sigue pidiendo un número hasta que se ingresa algo válido, entrando en un bucle hasta que el usuario ingresa un número.

3. Manejo de Errores

Cuando se trata de errores como la división por cero o la raíz cuadrada de un número negativo, me aseguré de que el programa mostrara un mensaje en la consola y no guardara estas operaciones en el historial. La función manejarError centraliza este manejo, asegurando que las respuestas sean consistentes. La forma de mostrarlo al usuario es a través de mensajes claros y concisos por consola.

4. Interfaz de Usuario

La intención desde el principio fue lograr que la interacción fuera sencilla y directa. Para mostrar el historial de manera clara, aproveché el método toString de Operacion, que convierte cada cálculo en una cadena legible, lo que hace que las operaciones válidas se vean organizadas y comprensibles en la consola. Además, diseñé un menú que guía al usuario a través de las opciones de manera intuitiva, asegurando que la calculadora fuera fácil de usar desde el inicio.

3. Desafíos Encontrados

1. Validación de Entradas unido con el Historial de Operaciones

Uno de los principales desafíos fue manejar el historial de operaciones. Quería asegurarme de que se mostrara de forma legible y organizada para facilitar su comprensión. Además, me aseguré de que solo las operaciones válidas se guardaran en el historial, evitando registros de cálculos incorrectos a través de la realización de validaciones.

2. Utilización de diferentes formas de crear funciones.

Un desafío que me autoimpuse fue el uso de diferentes formas de crear funciones. A lo largo del proyecto, decidí usar el prototype para los métodos como `realizarOperacion` y `toString` porque de esta manera las funciones se compartían entre todas las instancias de `Operacion`. Por otro lado, para las operaciones matemáticas más simples, utilicé funciones normales, ya que eran más fáciles de entender y de implementar. Además del uso de la función flecha para mostrar y recorrer el historial.

4. Conclusión

Desarrollar esta calculadora fue una experiencia altamente enriquecedora, en la que pude aprender y aplicar conceptos clave sobre la estructuración del código y el manejo adecuado de errores. Además, ayudo a ganar y afianzar conocimientos en el área de creación de objetos y funciones. A pesar de estos obstáculos, el resultado es una calculadora funcional, intuitiva y bien organizada. En el futuro, me gustaría agregar una interfaz gráfica más atractiva o incorporar nuevas funcionalidades para mejorar aún más su usabilidad.