Investigación de conceptos de JavaScript

Objetos JSON

Es un formato de intercambio de datos ligero y legible por humanos. Se utiliza comúnmente para transmitir datos entre un servidor y un cliente, o entre diferentes partes de una aplicación. Un ejemplo de objeto JSON:

```
{
  "Modelo": "Corolla",
  "Marca": "Toyota",
  "Año": 2023,
  "Motor": 1.6
}
```

Estructuras de control

Las estructuras de control permiten a los desarrolladores controlar el flujo de ejecución del código.

If else: Esta estructura permite ejecutar un bloque de código si una condición se evalúa como verdadera y otro bloque si la condición se evalúa como falsa.

Switch: Se utiliza cuando tienes múltiples casos posibles y quieres ejecutar diferentes bloques de código en función del valor de una expresión.

For: Esta estructura se utiliza para crear bucles que ejecutan un bloque de código un número específico de veces.

While: Permite ejecutar un bloque de código mientras una condición se evalúe como verdadera.

Do while: Garantiza que el bloque de código se ejecute al menos una vez antes de verificar la condición

For Each: Método específico para las matrices que permite ejecutar una función para cada elemento de la matriz.

For of: Se utiliza para iterar sobre elementos de objetos iterables, como matrices y cadenas.

Let - Const vs. Var

Las tres son formas de declarar variables en el lenguaje JavaScript, las diferencias son las siguientes:

Var: Las variables declaradas con var tienen un alcance de función. Esto significa que la variable está disponible en toda la función en la que se declara, incluso antes de su declaración.

var x = 10;

var x = 20; // Esto es válido y reasigna el valor de x.

Let: Las variables declaradas con let tienen un alcance de bloque (block scope). Esto significa que la variable está disponible solo dentro del bloque en el que se declara (bloque entre llaves {}).

let y = 30;

y = 40; // Esto es válido y reasigna el valor de y.

let y = 50; // Esto genera un error de re declaración.

Const: Las variables declaradas con const también tienen un alcance de bloque, pero a diferencia de let, las variables const no pueden ser reasignadas después de su declaración. No les puedes cambiar el valor una vez declaradas.

const z = 60;
z = 70; // Esto genera un error porque z no puede ser reasignada.

Funciones (normales, anónimas y funciones flecha)

Funciones Normales: Estas son funciones que tienen un nombre y se pueden utilizar en cualquier parte del código, incluso antes de su declaración debido al hoisting.

Function suma (a, b) {

return a + b;

}

Funciones Anónimas: Las funciones anónimas son funciones sin nombre. A menudo se utilizan como argumentos para otras funciones o se asignan a variables. También se pueden usar como closures.

const multiplicar = function (a, b) {

*return a * b;*

};

Funciones Flecha: Las funciones flecha son una forma más concisa de escribir funciones en JavaScript. Son especialmente útiles para funciones simples y tienen un comportamiento de enlace léxico, lo que significa que heredan el contexto de su contenedor.

```
const resta = (a, b) => a - b;

// Llamar a la función flecha
const resultado = resta (8, 3);
console.log(resultado); // Imprimirá 5
```

Las funciones flecha son especialmente útiles cuando se trabaja con funciones de orden superior (como map, filter y reduce) o cuando se necesita mantener el contexto de this en funciones dentro de objetos y clases.