

Depuración en JavaScript: debugger ; y Funciones de Consola

En JavaScript, la depuración es una parte esencial del proceso de desarrollo de aplicaciones. Permite identificar y corregir errores en el código, lo cual mejora la calidad y el rendimiento de la aplicación. En esta clase, nos enfocaremos en dos herramientas importantes para la depuración en JavaScript: `debugger ;` y las funciones de consola.

1. Puntos de interrupción: debugger ;

El statement `debugger ;` se utiliza para establecer puntos de interrupción en el código JavaScript. Al colocar esta instrucción en una línea específica, la ejecución del programa se detiene en ese punto, lo que permite inspeccionar el estado de las variables y seguir el flujo de ejecución.

```
function sumar(a, b) {  
    debugger; // Punto de interrupción  
    return a + b;  
}  
const resultado = sumar(5, 7);  
console.log(resultado);
```

Al ejecutar el código anterior en un entorno de desarrollo que admita depuración, como el navegador web o una herramienta como Node.js, la ejecución se detendrá en el punto donde se encuentra la instrucción `debugger ;`. A partir de ahí, puedes utilizar las herramientas de depuración para examinar las variables, ejecutar el código paso a paso y comprender mejor lo que está sucediendo.

2. Funciones de Consola

Las funciones de consola son un conjunto de herramientas que nos permiten mostrar información y mensajes en la consola del navegador o la terminal. Estas funciones son útiles para imprimir valores, mensajes de error y realizar un seguimiento del flujo de ejecución de nuestro código.

2.1. Imprimir en la consola: `console.log()`

La función `console.log()` se utiliza para imprimir mensajes y valores en la consola. Puede aceptar múltiples argumentos y los muestra en la consola separados por espacios.

```
const nombre = 'Juan';
const edad = 25;
console.log('Hola,', nombre + '! Tienes', edad, 'años.');
```

El código anterior imprimirá el siguiente mensaje en la consola:

```
Hola, Juan! Tienes 25 años.
```

2.2. Mensajes de error: `console.error()`

La función `console.error()` se utiliza para mostrar mensajes de error en la consola. Es especialmente útil para indicar cuando ocurren problemas o excepciones en el código.

```
const dividir = (a, b) => {
  if (b === 0) {
    console.error('Error: División por cero no permitida.');
```

```
    return null;
  }
  return a / b;
};
console.log(dividir(10, 2));
console.log(dividir(5, 0));
```

En el código anterior, cuando intentamos dividir por cero, se imprimirá el siguiente mensaje de error en la consola:

```
Error: División por cero no permitida.
```

2.3. Otras funciones de consola

Además de `console.log()` y `console.error()`, existen otras funciones útiles de consola, como `console.warn()`, `console.info()` y `console.table()`. Aquí hay un breve resumen de cada una:

- `console.warn()` : Muestra mensajes de advertencia en la consola.
- `console.info()` : Muestra mensajes informativos en la consola.
- `console.table()` : Muestra una tabla en la consola con los datos proporcionados.

A continuación, se presentan ejemplos de uso de estas funciones:

```
const usuarios = [
  { nombre: 'Pedro', edad: 20 },
  { nombre: 'Juan', edad: 25 },
  { nombre: 'María', edad: 30 }
];
console.warn('Advertencia: Se encontraron usuarios duplicados.');
```

```
console.info('Información: Los usuarios están ordenados por edad.');
```

```
console.table(usuarios);
```

En el código anterior, se imprimirá un mensaje de advertencia, uno informativo y una tabla en la consola con los datos de los usuarios:

```
Advertencia: Se encontraron usuarios duplicados.
Información: Los usuarios están ordenados por edad.
```

(indice)	nombre	edad
0	'Pedro'	20
1	'Juan'	25
2	'Maria'	30

Conclusión

En este material, hemos explorado dos herramientas esenciales para la depuración en JavaScript: `debugger`; y las funciones de consola. `debugger`; nos permite establecer puntos de interrupción en el código para examinar y seguir el flujo de ejecución. Por otro lado, las funciones de consola, como `console.log()`, `console.error()`, `console.warn()`, `console.info()` y `console.table()`, nos ayudan a imprimir mensajes y valores, mostrar errores y visualizar datos en forma de tabla. Estas herramientas son fundamentales para el proceso de depuración y nos permiten mejorar la calidad y el rendimiento de nuestras aplicaciones JavaScript. ¡Ahora estás listo para depurar con confianza y utilizar las funciones de consola de manera efectiva!