

Felipe Rueda

¿Qué es un closure?

Un cierre o closure en JavaScript es un concepto fundamental que se relaciona con la forma en que funcionan las funciones y el alcance léxico en el lenguaje. En términos simples, se produce un cierre cuando una función "recuerda" las variables de su ámbito exterior, incluso después de que la función exterior haya terminado de ejecutarse.

```
function exterior() {  
  var variableExterior = "Soy de la función exterior";  
  
  function interna() {  
    console.log(variableExterior); // Accede a variableExterior desde el c  
  }  
  
  return interna;  
}  
  
var funcionCierre = exterior();  
funcionCierre(); // Imprime "Soy de la función exterior"
```

En este ejemplo, exterior define una función interna interna que hace referencia a variableExterior. Cuando se llama a exterior y se devuelve la función interna, la variable funcionCierre contiene una referencia a la función interna junto con su cierre. Aunque la función exterior haya terminado de ejecutarse, funcionCierre aún puede acceder y usar variableExterior desde su cierre.

Los cierres son poderosos porque permiten crear datos privados y mantener el estado en JavaScript. A menudo se utilizan en situaciones como la creación de funciones de fábrica, la gestión de devoluciones de llamada y la implementación de la encapsulación de datos.

¿Qué es un hosting?

En el contexto de JavaScript, "hosting" no se refiere a la acción de proporcionar alojamiento para un sitio web, como suele entenderse en el mundo del desarrollo web. En cambio, el término "hoisting" (traducido como "elevación" o "izamiento") se refiere a un comportamiento específico de cómo JavaScript maneja las declaraciones de variables y funciones en su fase de compilación y ejecución.

El hoisting es una característica de JavaScript que puede resultar confusa para los desarrolladores, ya que puede parecer que las declaraciones de variables y funciones se mueven automáticamente al comienzo de su alcance, incluso si están definidas más abajo en el código.

Existen dos tipos de Hosting, de variables y de funciones, a continuación un ejemplo de variables.

```
var miVariable;  
console.log(miVariable); // undefined  
miVariable = 10;
```

El maximo común divisor entre a y b:

```
const gcd = (a, b) => (b === 0 ? a : gcd(b, a % b));
```