¿Cuáles son las diferencias y similitudes entre Java y JavaScript?

Diferencias:

- Java es un lenguaje de programación orientado a objetos diseñado para ser independiente de la plataforma, mientras que JavaScript es un lenguaje de secuencias de comandos principalmente utilizado en el desarrollo web.
- Java se compila en bytecode y se ejecuta en una máquina virtual Java (JVM), mientras que JavaScript es un lenguaje interpretado que se ejecuta en el navegador web o en un entorno de servidor a través de Node.js.
- Java está más orientado a la programación de aplicaciones de escritorio, servidores y dispositivos móviles, mientras que JavaScript se utiliza principalmente para crear interacciones dinámicas en sitios web.

Similitudes:

- Ambos lenguajes utilizan una sintaxis similar a C y C++, lo que hace que sea relativamente fácil para los programadores que conocen uno de los dos aprender el otro.
- Ambos lenguajes son ampliamente utilizados en la industria del desarrollo de software y tienen una amplia comunidad de desarrolladores.
- Tanto Java como JavaScript admiten la programación orientada a objetos, aunque de diferentes maneras y con diferentes características.

¿Cuáles son algunos métodos del objeto Date en JavaScript?

Algunos métodos del objeto Date en JavaScript incluyen:

- **getDate():** Obtiene el día del mes.
- **getMonth():** Obtiene el mes (0-11).
- **getFullYear():** Obtiene el año (a cuatro dígitos).
- **getHours():** Obtiene la hora.
- **toISOString()**: Devuelve una cadena en formato ISO (UTC).

¿Cuáles son algunos métodos de los arreglos en JavaScript?

Algunos métodos de los arreglos en JavaScript incluyen:

- **push():** Agrega uno o más elementos al final de un array y devuelve la nueva longitud del array.
- **pop():** Elimina el último elemento de un array y lo devuelve.
- **splice():** Cambia el contenido de un array eliminando elementos existentes y/o agregando nuevos elementos.
- **forEach():** Ejecuta una función por cada elemento del array.
- **filter():** Crea un nuevo array con todos los elementos que pasan una prueba implementada por la función proporcionada.

¿Cómo se declara una variable con alcance local dentro de una función en JavaScript?

En JavaScript, las palabras clave var, let y const se utilizan para declarar variables locales dentro de funciones, pero difieren en su alcance y capacidad de reasignación de valores.

- **var:** Es la opción tradicional, pero sus variables tienen un alcance de función, lo que puede llevar a comportamientos inesperados y redeclaraciones accidentales.

- **let:** Introducido en ES6, ofrece un alcance de bloque más estricto y no permite redeclaraciones en el mismo ámbito, lo que mejora la claridad y previene errores.

- **const:** También introducido en ES6, declara variables cuyos valores no cambian después de la asignación inicial. Proporciona una garantía adicional al evitar la reasignación, pero permite modificar el contenido de objetos y arrays referenciados por la variable.

¿Cuáles son las implicaciones de utilizar variables globales dentro de funciones en JavaScript?

Utilizar variables globales dentro de funciones puede tener varias implicaciones negativas:

- Puede hacer que el código sea menos legible y mantenible, ya que las funciones dependen de variables que no están explícitamente pasadas como parámetros.
- Puede causar conflictos de nombres si se utilizan variables globales con nombres similares en diferentes partes del código.
- Las variables globales son accesibles desde cualquier parte del código, lo que puede aumentar el riesgo de efectos secundarios no deseados o errores difíciles de rastrear.
- Puede dificultar la reutilización del código, ya que las funciones pueden depender de variables globales específicas del contexto en el que se definen.