

Glosario de 25 términos comunes en JavaScript:

1. **Variable:** Espacio de almacenamiento para datos, definido con var, let o const.
2. **Función:** Bloque de código diseñado para realizar una tarea específica. Se define con la palabra clave function.
3. **Objeto:** Colección de pares clave-valor. Se define con llaves {} y puede contener propiedades y métodos.
4. **Array:** Estructura de datos que almacena una lista ordenada de elementos. Se define con corchetes [].
5. **Evento:** Acción que ocurre en el navegador, como un clic o una tecla presionada, que puede ser manejada con addEventListener.
6. **DOM (Document Object Model):** Representación en memoria de la estructura de un documento HTML. Permite la manipulación de elementos HTML y CSS.
7. **Callback:** Función que se pasa como argumento a otra función y se ejecuta después de que la tarea de la función principal se complete.
8. **Promesa (Promise):** Objeto que representa el resultado eventual de una operación asíncrona. Puede estar en estado pendiente, cumplido o rechazado.
9. **Async/Await:** Sintaxis que simplifica el trabajo con promesas y facilita la escritura de código asíncrono.
10. **Clausura (Closure):** Función que recuerda el entorno en el que fue creada, permitiendo acceder a variables de esa función incluso después de que haya terminado su ejecución.
11. **Método:** Función asociada a un objeto. Se define como una propiedad de ese objeto que contiene una función.
12. **Propiedad:** Valor asociado a una clave dentro de un objeto.
13. **Constructor:** Función especial en una clase que se utiliza para crear e inicializar objetos.
14. **Clase:** Plantilla para crear objetos con propiedades y métodos, introducida en ES6.
15. **Herencia:** Mecanismo que permite que una clase adquiera propiedades y métodos de otra clase.
16. **Scope (Ámbito):** Contexto en el que se definen y accede a las variables y funciones.
17. **Hoisting:** Comportamiento de JavaScript que eleva las declaraciones de variables y funciones al inicio del ámbito en el que se encuentran.
18. **Temporizador (Timer):** Función que permite ejecutar código después de un cierto período de tiempo, usando setTimeout o setInterval.

19. **Iterador**: Objeto que permite recorrer elementos en una colección, como los arrays, utilizando métodos como `forEach` o `map`.
20. **Módulo**: Unidad de código que puede exportar e importar funcionalidades, introducida con ES6 para la modularidad.
21. **JSON (JavaScript Object Notation)**: Formato ligero de intercambio de datos que utiliza una sintaxis similar a la de los objetos JavaScript.
22. **Llamada a función (Function Call)**: Ejecución de una función, pasando argumentos si es necesario.
23. **Operador**: Símbolo que realiza operaciones sobre operandos, como `+`, `-`, `*`, y `&&`.
24. **Template Literal**: Cadena de texto que permite interpolar expresiones y se define usando las comillas invertidas ```.
25. **Strict Mode**: Modo que enforza una sintaxis más rigurosa y errores más estrictos en el código, activado con `"use strict";`.