

UNIVERSIDAD DE COSTA RICA

Escuela de Ciencias de la Computación e Informática

Ingeniería de Software

Laboratorio 2B

Grupo 03

Estudiante:

Andy Hernández Rojas

C23785

Profesora: Rebeca Obando

I Ciclo

2025

1. ¿Cuáles son los tipos de datos que soporta JavaScript?

JavaScript soporta varios tipos de datos primitivos, entre los que se encuentran: String (cadenas de texto), Number (números), Boolean (valores verdadero o falso), Undefined (valor no asignado), Null (ausencia intencionada de valor), Symbol (identificadores únicos) y BigInt (números enteros grandes). Además, cuenta con tipos de datos complejos como objetos y arreglos que permiten almacenar múltiples valores (MDN Web Docs, 2023).

2. ¿Cómo se puede crear un objeto en JavaScript? Dé un ejemplo.

Un objeto en JavaScript es una colección de propiedades, donde cada propiedad es un par clave-valor. Por ejemplo, un objeto que representa un coche podría tener propiedades como marca, modelo y año (W3Schools, 2023).

3. ¿Cuáles son los alcances (scope) de las variables en JavaScript?

JavaScript maneja tres tipos principales de alcance de variables: global (accesible en todo el programa), local o de función (accesible solo dentro de la función donde se declara) y de bloque (accesible solo dentro de un bloque de código específico) (MDN Web Docs, 2023).

4. ¿Cuál es la diferencia entre undefined y null?

La diferencia entre undefined y null radica en que undefined se asigna automáticamente cuando una variable ha sido declarada pero no se le ha asignado ningún valor, mientras que null es un valor asignado intencionalmente para representar la ausencia de valor (Geeks for Geeks, 2024).

5. ¿Qué es el DOM?

El Document Object Model (DOM) es una representación estructurada de un documento HTML que permite acceder y manipular sus elementos de manera programática. A través del DOM, los desarrolladores pueden modificar el contenido, la estructura y el estilo de una página web en tiempo real (MDN Web Docs, 2023).

6. Usando Javascript se puede acceder a diferentes elementos del DOM. ¿Qué hacen, que retorna y para qué funcionan las funciones getElement y querySelector? Cree un ejemplo

getElementById: Devuelve el elemento que coincide con el ID especificado.

querySelector: Devuelve el primer elemento que coincida con un selector CSS especificado. Estas funciones permiten manipular elementos específicos del DOM (W3Schools, 2023).

7. Investigue cómo se pueden crear nuevos elementos en el DOM usando Javascript. De un ejemplo

Para crear nuevos elementos en el DOM, se utiliza el método `createElement()` que permite generar un nodo de elemento. Posteriormente, se añade al DOM con métodos como `appendChild()` o `insertBefore()` (MDN Web Docs, 2023).

8. ¿Cuál es el propósito del operador this?

El operador `this` se refiere al objeto que invoca la función en la que se encuentra, permitiendo acceder a las propiedades y métodos del objeto actual (W3Schools, 2023).

9. ¿Qué es un promise en JavaScript?

Una Promise es un objeto que representa la eventual finalización (o falla) de una operación asíncrona y su valor resultante, lo que facilita la gestión de operaciones asíncronas en JavaScript (MDN Web Docs, 2023).

10. ¿Qué es Fetch en JavaScript?

La API `fetch()` proporciona una forma sencilla de realizar solicitudes HTTP, facilitando la obtención de recursos de forma asíncrona (MDN Web Docs, 2023).

11. ¿Qué es Async/Await en JavaScript?

Async/Await es una sintaxis que simplifica la gestión de Promises, haciendo que el código asíncrono se lea de manera similar al código sincrónico, mejorando la claridad y el manejo de errores (MDN Web Docs, 2023).

12. ¿Qué es un Callback?

Un Callback es una función que se pasa como argumento a otra función y se ejecuta después de que la función principal haya completado su tarea, permitiendo la ejecución de operaciones asíncronas (Geeks for Geeks, 2024).

13. ¿Qué es Closure?

Un Closure es una función que recuerda el entorno en el que fue creada, incluso después de que ese entorno haya dejado de existir, lo que permite preservar variables y acceder a ellas en diferentes momentos (MDN Web Docs, 2023).

14. ¿Cómo se puede crear un cookie usando JavaScript?

Para crear una cookie en JavaScript se asigna un valor a `document.cookie` junto con parámetros como `expires` o `path` para controlar su duración y alcance (W3Schools, 2023).

15. ¿Cuál es la diferencia entre `var`, `let` y `const`?

var: Tiene un ámbito de función y puede redeclararse.

let: Tiene un ámbito de bloque y no puede redeclararse en el mismo ámbito.

const: Tiene un ámbito de bloque y su valor no puede ser reasignado una vez declarado (MDN Web Docs, 2023).

Bibliografía:

Geeks for Geeks. (2024). *Undefined Vs Null in JavaScript*. Recuperado de <https://www.geeksforgeeks.org>

MDN Web Docs. (2023). *JavaScript Documentation*. Recuperado de <https://developer.mozilla.org>

W3Schools. (2023). *JavaScript Tutorial*. Recuperado de <https://www.w3schools.com>