**Fecha**: 16 de junio de 2026

**Nombre del aprendiz:** Jonathan David Diaz Ochoa.

**Instructor:** Daniel Martínez Payán

**TALLER DE JAVASCRIPT**

Para la resolución de este taller se necesita que lo resuelva en el Word y adjunte las ***capturas de pantalla de su código*** implementado para sus ejemplos. Recuerde que este taller usted lo puede resolver como desee (Si utiliza la IA para resolverlo es su problema, recuerde que el parcial de esto será a Lápiz y Papel, así que le recomiendo que en la medida no la utilice y lo realice a conciencia).

**Conceptos fundamentales**

1. ¿Qué es JavaScript y para qué se utiliza principalmente en el desarrollo web?
2. ¿Cómo se inserta JavaScript en un archivo HTML?
3. ¿Cuál es la diferencia entre var, let y const?
4. ¿Qué tipo de datos primitivos existen en JavaScript? (nombra al menos 4)
5. ¿Cómo se declara un array y cómo se accede al primer elemento?
6. ¿Qué es un objeto en JavaScript y cómo se declara uno?
7. ¿Para qué sirve console.log()?
8. ¿Qué resultado devuelve la expresión typeof 10?
9. ¿Cómo se concatena texto en JavaScript?
10. ¿Qué operadores aritméticos básicos conoces?

**CONDICIONALES Y BUCLES**

1. ¿Cómo se escribe una estructura if-else en JavaScript?
2. ¿Cuál es la diferencia entre == y ===?
3. ¿Qué estructura de control usarías para recorrer un array?
4. Escribe un bucle for que imprima los números del 1 al 5.
5. ¿Cómo funciona el bucle while? Da un ejemplo.
6. ¿Para qué sirve el switch?

**FUNCIONES**

1. ¿Qué es una función en JavaScript y para qué sirve?
2. ¿Cómo se declara una función llamada saludar que imprima "Hola"?
3. ¿Cuál es la diferencia entre una función declarada y una función expresada?
4. ¿Qué es un parámetro y qué es un argumento en una función?
5. ¿Cómo se retorna un valor desde una función? Da un ejemplo.
6. ¿Qué es una función flecha (arrow function)?
7. Escribe una función que reciba dos números y devuelva su suma.

**POO (Programación Orientada a Objetos)**

1. ¿Qué es un objeto en la POO de JavaScript?
2. ¿Qué es una clase?
3. ¿Cómo se define una clase y se crea una instancia de ella?
4. ¿Qué es un constructor en una clase?
5. ¿Cómo se define un método dentro de una clase?
6. ¿Qué es this y cómo funciona dentro de una clase?
7. Escribe un ejemplo de clase Persona con un constructor que reciba nombre y edad, y un método que imprima esos datos.

**Solución**:

## Conceptos fundamentales

1. JavaScript es un lenguaje de programación interpretado, dinámico y orientado a objetos que se ejecuta principalmente en los navegadores web, en el desarrollo web, JavaScript se utiliza principalmente para agregar funcionalidad dinámica e interactividad a las páginas web.

2. una vez la estructura del HTML esta insertada con el “!” o usando html:5, procedemos a insertar el comando “script” dentro del body y luego dentro de este insertamos el comando “src” referenciando el archivo de j.s previamente creado.

3. **Var: (forma antigua de declarar variables).**

-No respeta los bloques (if, for, etc.).

-Se puede re-declarar y reasignar.

-Se "eleva" al principio de la función, pero su valor queda como undefined hasta que se asigna.

**Let (forma moderna, recomendada).**

-Tiene scope de bloque (respeta {}).

- Se puede reasignar, pero no se puede volver a declarar en el mismo bloque.

- Se “eleva”, pero no se puede usar antes de declararla (error).

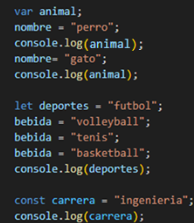
**Const: (para valores constantes).**

-Igual que let, scope de bloque.

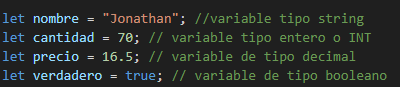
- No se puede reasignar (es constante).

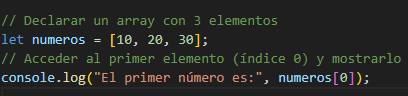
- No se puede re-declarar.

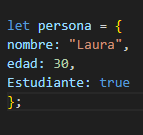
- Se puede modificar su contenido, pero no se puede cambiar la referencia.



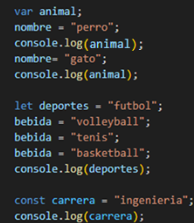
4. En JavaScript, los datos primitivos son los tipos de datos más básicos, que no son objetos y no tienen métodos. Son inmutables, lo que significa que su valor no puede ser cambiado directamente.



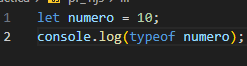
5. Un array (o arreglo) es una estructura de datos que permite almacenar múltiples valores en una sola variable, ordenados por posición (índices).

6. Un objeto es una estructura de datos que permite almacenar pares clave–valor (key–value) cada propiedad se asocia a un valor, que puede ser de cualquier tipo: número, string, función, array, otro objeto, estos permiten organizar y agrupar datos relacionados en una sola entidad.

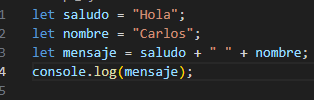
7. Se usa para imprimir mensajes en la consola del navegador o del entorno de ejecución.



8. Se utiliza para conocer el tipo de dato de un valor.



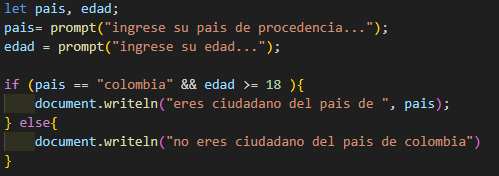
9. Se puede concatenar utilizando operadores.



10. Los operadores aritméticos son + suma, - resta, / división, \* multiplicación, % modulo, \*\* potenciación, ++ incremento y – decremento.

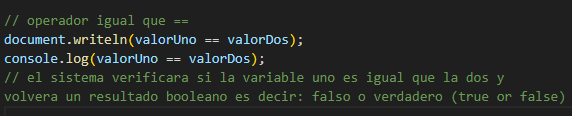
## CONDICIONALES Y BUCLES

11. La estructura if-else Se usa para tomar decisiones. Evalúa una condición, y dependiendo si es true o false, ejecuta un bloque de código u otro.

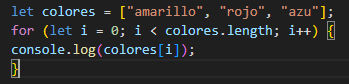
****

12. == compara los valores, pero convierte los tipos si son diferentes.

=== compara valores y tipos, sin hacer conversión de tipos.



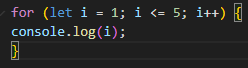
13. Utilizando el for se puede recorrer el arrays.



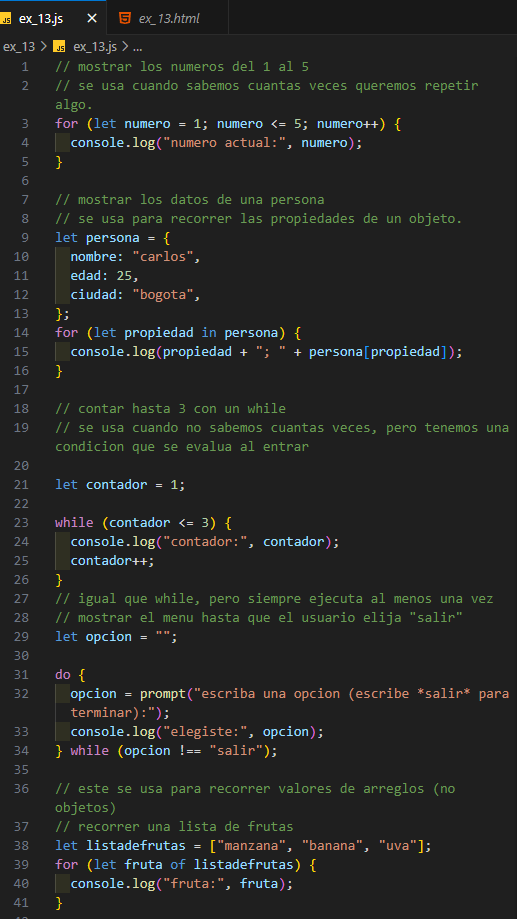
14. Se tiene en cuenta: let i = 1: empieza en 1.

i <= 5: se repite mientras i sea menor o igual a 5.

i++: incrementa i en 1 en cada vuelta.



15. While repite un bloque de código mientras una condición sea verdadera,

es útil cuando no sabes exactamente cuántas veces debe repetirse el ciclo.

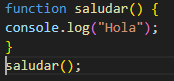
16. Switch es una estructura de control que sirve para evaluar una expresión y ejecutar diferentes bloques de código según el valor de esa expresión.

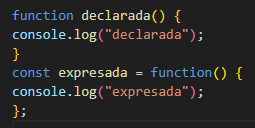


## FUNCIONES

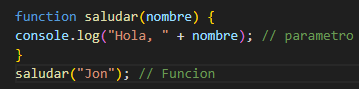
17. Una función en JavaScript es un bloque de código reutilizable que realiza una tarea o calcula un valor, funciona para organizar el código, evitar repetirlo y facilitar su mantenimiento.

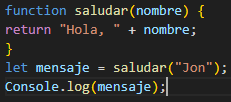


18. Función saludar:

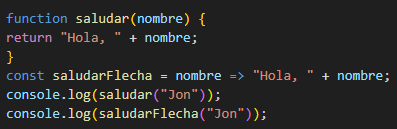
19. Funcion declarada: Se declara usando la palabra clave function con un nombre, fuera de cualquier expresión.   
Funcion expresada: Se asigna una función (anónima o nombrada) a una variable.

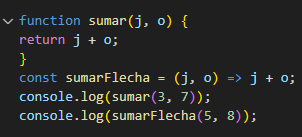
20. Parámetro: Variable definida en la función (como una "cajita" para guardar datos).

Argumento: Valor que pones dentro de la cajita cuando usas la función.

21. Para que una función se devuelva se utiliza el comando clave “return”.

22. Es una manera más fácil y corta para escribir funciones.



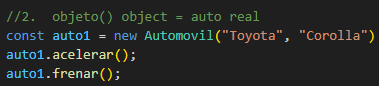
23. Usando función =>.  


## POO (Programación Orientada a Objetos)

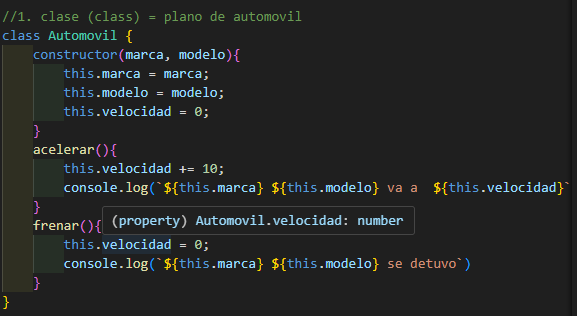
24. Objeto es una entidad que representa algo del mundo real o un concepto:

Tiene propiedades (también llamadas atributos o características) que describen sus datos o estado.

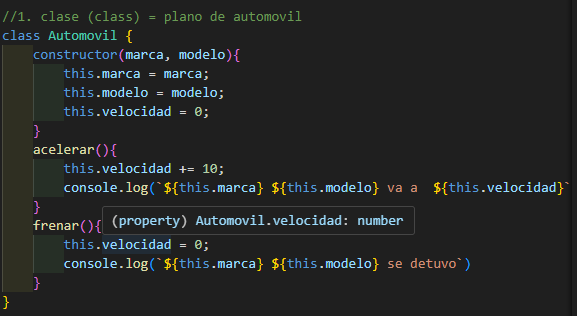
Tiene métodos (funciones) que definen su comportamiento o acciones que puede realizar.

****

25. Clase es como un molde o plantilla para crear objetos, define las propiedades (atributos) y métodos (comportamientos) que tendrán los objetos que se creen a partir de ella.

26. Definición de una clase.

27. Constructor funciona como un método especial dentro de una clase en JavaScript (y en otros lenguajes orientados a objetos). se ejecuta automáticamente cuando se crea una nueva instancia (objeto) de esa clase.

28. método es una función definida dentro de una clase, funciona para representar el comportamiento de los objetos creados a partir de esa clase.

29. This es una palabra clave especial que hace referencia al objeto actual —es decir, al objeto desde el cual se está ejecutando el método, accede a las propiedades del objeto y llama a otros métodos del mismo objeto.

30. Constructor con nombre y edad como parámetros, tambien método llamado presentarse() que imprime esos datos.

