



Clase 4

Desarrollo web Avanzado - Fullstack



Hello, Welcome!

Registra tu asistencia

<https://forms.gle/6hfdT5U5v5fF34RQ8>





Repaso de Fundamentos de Programación con JavaScript e Inducción a la Terminal o Bash.

Revisaremos los Fundamentos de JS y veremos una inducción a Bash o Terminal



¿Qué es JavaScript?

- Lenguaje de programación interpretado.
- Corre en el navegador (Frontend) y en el servidor (Node.js).
- Permite crear páginas web dinámicas e interactivas

¿Para qué se usa JS?

- Validación de formularios.
- Interacción con el DOM.
- Llamadas a APIs.
- Desarrollo de juegos y apps web.



Variables y Tipos de Datos

`let`, `const`, `var`: formas de declarar variables.

Variables Locales | Variables Globales: <https://www.geeksforgeeks.org/javascript/global-and-local-variables-in-javascript/>

- Tipo de datos: string, number, boolean, object, null, undefined.



```
let nombre = "Juan"; // Variable de tipo string
let edad = 30; // Variable de tipo number
let esMayor = true; // Variable de tipo boolean
let nulo = null; // Variable de tipo null
let indefinido; // Variable de tipo undefined (inicialmente undefined)
const PI = 3.1416; // Constante de tipo number
let frutas = ["manzana", "banana", "naranja"]; // Array (tipo object)
function saludar(nombre) { return "Hola, " + nombre; } // Función (tipo object)
```



Operadores Basicos

let suma = 2 + 2;

let comparacion = 5 > 3;

- Aritméticos: +, -, *, /
- Comparación: ==, ===, !=, >, <



Estructuras de Control

- Condicionales: `if`, `else`, `switch`
- Bucles: `for`, `while`, `do...while`



Diferencias

`if`

Condición

Para decisiones simples o múltiples.

`while`

Repetición

Para repetir algo mientras sea válido.

`switch`

Condición exacta

Para comparar una variable con varios valores fijos.



If

```
const precioPublico = 70;  
const edadCliente = 25;  
var clienteDescuento = true;
```

```
if (clienteDescuento) {  
  console.log("El cliente tiene descuento. Precio final: $" + (precioPublico * 0.8))  
} else {  
  console.log("El cliente no tiene descuento. Precio final: $" + precioPublico);  
}
```

Si `clienteDescuento` es `true`, se aplica un **20% de descuento** (puedes cambiar ese porcentaje).

Si no, se cobra el precio completo.



While

```
const precioPublico = 70;  
const edadCliente = 25;  
var clienteDescuento = true;
```

```
// Usamos while para repetir el cálculo mientras el cliente tenga descuento  
while (clienteDescuento) {  
  const precioFinal = precioPublico * 0.8;  
  console.log("El cliente tiene descuento. Precio final: $" + precioFinal);
```

```
  // Rompemos el ciclo para que no sea infinito  
  clienteDescuento = false;  
}
```


```
if (!clienteDescuento) {  
  console.log("El cliente ya no tiene descuento. Precio normal: $" + precioPublico);  
}
```



Switch

```
const precioPublico = 70;
const edadCliente = 25;
var clienteDescuento = true;

switch (clienteDescuento) {
  case true:
    console.log("El cliente tiene descuento. Precio final: $" + (precioPublico * 0.8));
    break;
  case false:
    console.log("El cliente no tiene descuento. Precio final: $" + precioPublico);
    break;
  default:
    console.log("Estado de descuento desconocido.");
}
```



Ejercicio | lógica en condicionales

```
if (calificacion >= 7) {  
  console.log("Has aprobado");  
} else {  
  console.log("Has Reprobado");  
}
```

//Actividad de participacion // Completa el código para que compile correctamente



Funciones

```
function saludar(nombre) {  
  return "Hola, " + nombre;  
}  
console.log(saludar(nombre));
```

- Declarativas y expresiones
- Arrow functions (`=>`)

<https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Functions>



Objetos

En JavaScript, la **sintaxis de objetos** permite agrupar datos y funcionalidades relacionados mediante **pares clave-valor**.

```
const persona = {  
  nombre: "Jose",  
  edad: 30,  
  esEstudiante: true  
};
```

```
console.log(persona.nombre);  
console.log(persona["edad"]);
```

https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Working_with_objects



El DOM y Eventos

- DOM: Document Object Model
- Escuchar eventos: `click`, `submit`, `change`

https://developer.mozilla.org/es/docs/Web/API/Document_Object_Model

¿Por qué es importante el DOM?



1. Interacción con la página

Permite que JavaScript pueda **leer, modificar, crear o eliminar elementos HTML** en tiempo real (sin recargar la página).

Ejemplo: cambiar el texto, mostrar mensajes, ocultar botones, validar formularios, etc.

2. Dinamicidad

Gracias al DOM, una página puede **responder a acciones del usuario** (clics, escritura, movimiento del mouse, etc.).


Ejemplo: un menú que se despliega al pasar el mouse, un botón que cambia de color al hacer clic, etc.

3. Conexión entre HTML, CSS y JavaScript

El DOM es el **punto** entre el contenido (HTML), el estilo (CSS) y el comportamiento (JavaScript).



Un ejemplo básico del uso del DOM (Document Object Model) en JavaScript implica acceder a un elemento HTML específico y modificar su contenido. Por ejemplo, se puede cambiar el texto de un párrafo o modificar el estilo de un elemento.



```
<!DOCTYPE html>
<html>
<head>
<title>Ejemplo DOM</title>
</head>
<body>
  <p id="miParrafo">Este es el texto original.</p>
  <button onclick="cambiarTexto()">Cambiar Texto</button>

  <script>
    function cambiarTexto() {
      // Acceder al elemento con el ID "miParrafo"
      const parrafo = document.getElementById("miParrafo");

      // Modificar su contenido
      parrafo.innerText = "¡El texto ha sido cambiado!";
    }
  </script>
</body>
</html>
```



En este ejemplo:

Se crea un párrafo con el ID "miParrafo" y un botón.

Al hacer clic en el botón, se llama a la función `cambiarTexto()`.

`cambiarTexto()` utiliza `document.getElementById("miParrafo")` para obtener una referencia al elemento párrafo.

Luego, `parrafo.innerText = "¡El texto ha sido cambiado!"`; cambia el contenido de texto del párrafo al nuevo texto.



```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Evento de Alerta , DOM</title>
</head>
<body>

  <button id="btn">Haz clic aquí</button>

  <script>
    document.getElementById("btn").addEventListener("click", function() {
      alert("¡Clic!");
    });
  </script>

</body>
</html>
```



En este ejemplo:

```
document.getElementById("btn");
```

Busca un elemento en el HTML que tenga el atributo `id="btn"`.

```
.addEventListener("click", function() { ... });
```

Agrega un *escuchador de eventos* para detectar cuando ese elemento (el botón) sea **clicado**.

```
alert("¡Clic!");
```

Cuando se hace clic en el botón, se ejecuta esta función anónima que muestra una ventana emergente (*alert*) con el mensaje **"¡Clic!"**.



Buenas Prácticas

Usa `const` y `let`, evita `var`.

Nombres descriptivos para variables.

Comenta tu código.

Usa linters y formateadores.



Ejercicios practico - mini proyecto

Empezaremos a desarrollar un generador de contraseñas con lo que hemos aprendido hasta ahora.

...Code Hour ;) ...Siguiente Clase



A Participar

Opiniones

Dudas y Respuestas



See you later!

Gracias por tu tiempo, Nos vemos la siguiente sesión :)