

Sintaxis básica

- **Variables:** let, const, var
- **Tipos de Datos:** number, string, boolean, null, undefined, object, symbol
- **Estructuras de Control:** if, else, switch, for, while
- **Funciones:** Declaración de función, funciones de flecha
- **Objetos:** Creación de objetos, propiedades, métodos
- **Eventos:** Manejo de eventos del DOM

Variables

JavaScript utiliza las palabras clave `let`, `const`, y `var` para declarar variables.

```
let nombre = "Juan";  
const edad = 30;  
var esActivo = true;
```

Imprimir por consola

JavaScript utiliza la función `console.log()` para imprimir datos en la consola del navegador.

```
let nombre = "Juan";  
console.log(nombre);
```

Tipos de datos

JavaScript tiene varios tipos de datos, incluyendo number, string, boolean, null, undefined y object.

```
let numero = 42;  
let texto = "Hola Mundo";  
let esVerdadero = true;  
let nulo = null;  
let indefinido = undefined;  
let objeto = { nombre: "Juan", edad: 30 };
```

Estructuras de control

```
if (edad >= 18) {  
    console.log("Es mayor de edad");  
} else {  
    console.log("Es menor de edad");  
}
```

Estructuras de control

```
for (let i = 0; i < 5; i++) {  
    console.log(i);  
}
```

```
let contador = 0;  
while (contador < 5) {  
    console.log(contador);  
    contador++;  
}
```

Funciones

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

```
let sumar = function(a, b) {  
    return a + b;  
};
```

Objetos

```
let persona = {  
  nombre: "Juan",  
  edad: 30,  
  esEstudiante: false,  
  saludar: function() {  
    console.log("Hola, soy " + this.nombre);  
  }  
};
```

```
console.log(persona.nombre); // Acceder a una propiedad  
persona.saludar(); // Llamar a un método
```


Objetos especiales

El objeto document representa el documento HTML cargado en el navegador y proporciona métodos y propiedades para interactuar con él.

```
let titulo = document.querySelector("h1");
```

```
document.write("<h2>Nuevo Título</h2>");
```

Objetos especiales

El objeto window representa la ventana del navegador y proporciona métodos y propiedades relacionadas con la ventana y su contenido.

```
window.alert("¡Hola Mundo!");
```

```
let nombre = window.prompt("Por favor, introduce tu nombre:");  
if (nombre) {  
    console.log("Hola, " + nombre);  
} else {  
    console.log("No se proporcionó un nombre");  
}
```

Objetos de tipo elemento

Los elementos son un tipo especial de nodo que representan elementos HTML en el DOM.

Propiedades y métodos básicos:

- **innerHTML, textContent**: Contenido HTML o texto dentro del elemento.
- **getAttribute(nombre)**: Obtiene el valor de un atributo del elemento.
- **setAttribute(nombre, valor)**: Establece el valor de un atributo del elemento.
- **classList**: Lista de clases CSS aplicadas al elemento.
- **addEventListener(evento, función)**: añade un listener

Objetos de tipo elemento

Propiedades básicas:

- **id**: devuelve el id del elemento.
- **src**: si es una imagen, accede al atributo src.
- **href**: si es un enlace, accede al atributo href.
- **value**: si es un input, accede al atributo value.
- **style**: accede al objeto de estilos CSS
 - `objeto.style.backgroundColor = "red";`
 - `objeto.style.color="green";`
 - `objeto.style.width="100px";`
 - `objeto.style.display="none";`

Objetos de tipo elemento

```
let elemento = document.querySelector("div"); // Obtener un elemento (el primero)
```

```
console.log(elemento.tagName); // Imprimir el nombre de la etiqueta  
console.log(elemento.innerHTML); // Imprimir el HTML interno del elemento  
console.log(elemento.getAttribute("id")); // Obtener el valor del atributo 'id'  
elemento.classList.add("nueva-clase"); // Agregar una nueva clase al elemento
```

```
let elemento = document.querySelector("p#intro"); // Obtener un elemento <p> con  
id intro
```

```
console.log(elemento.textContent); // Imprimir el texto del elemento  
elemento.textContent = "¡Hola Mundo!"; // Cambiar el texto dentro del elemento
```

Obtener elementos

La función `querySelector()` sirve seleccionar un solo elemento del DOM utilizando un selector CSS. Retorna el primer elemento que coincida con el selector especificado. Si no se encuentra ninguna coincidencia, retorna `null`.

```
// Obtener el primer párrafo del documento
let primerParrafo = document.querySelector("p");
```

```
// Obtener un elemento con un ID específico
let elementoConId = document.querySelector("#miElemento");
```

```
// Obtener un elemento con una clase específica
let elementoConClase = document.querySelector(".miClase");
```

Lista de clases

La propiedad `classList` es una propiedad que retorna una colección de las clases CSS de un elemento. Permite agregar, quitar y verificar clases de manera sencilla en un elemento HTML.

```
let elemento = document.getElementById("miElemento");
```

```
// Agregar una clase al elemento  
elemento.classList.add("nuevaClase");
```

```
// Eliminar una clase del elemento  
elemento.classList.remove("claseExistente");
```

```
// Alternar la presencia de una clase  
elemento.classList.toggle("otraClase");
```

Eventos

JavaScript permite manejar eventos del DOM, como clics, cambios, y teclas presionadas. Se deben crear listeners, indicando que hacer cuando el evento ocurra.

```
function avisar() {  
    console.log("El botón fue clickeado");  
}
```

```
let boton = document.querySelector("#miBoton");  
boton.addEventListener("click", avisar );
```


Cambiar clases

```
// Obtener referencia al elemento y al botón
let parrafo = document.getElementById("parrafo");
let boton = document.getElementById("boton");

// Definir función para cambiar la clase
function cambiarClase() {
    parrafo.classList.toggle("resaltado");
}

// Agregar listener al botón
boton.addEventListener("click", cambiarClase);
```