



# Manipulación del DOM

## Manipulación del DOM en JavaScript:

El DOM (Document Object Model) es una representación jerárquica de la estructura de un documento HTML. La manipulación del DOM en JavaScript permite interactuar y modificar dinámicamente el contenido, la estructura y el estilo de una página web.

### 1. Selección de Elementos:

Para manipular el DOM, primero necesitas seleccionar elementos. Puedes hacer esto mediante funciones como `getElementById`, `getElementsByClassName`, `getElementsByTagName`, o utilizando selectores más avanzados como `querySelector` y `querySelectorAll`.

```
// Seleccionar un elemento por su ID
```

```
let titulo = document.getElementById("tituloPrincipal");
```

```
// Seleccionar elementos por su clase
```

```
let elementosClase = document.getElementsByClassName("parrafo");
```

```
// Seleccionar elementos por su etiqueta
```

```
let elementosParrafo = document.getElementsByTagName("p");
```

```
// Seleccionar un elemento con un selector CSS
```

```
let primerParrafo = document.querySelector("p");
```

## 2. Modificación de Contenido:

Una vez que has seleccionado elementos, puedes modificar su contenido. Por ejemplo, cambiando el texto dentro de un párrafo.

```
// Modificar el contenido de un elemento

titulo.innerHTML = "Nuevo Título";

// Añadir un nuevo párrafo

let nuevoParrafo = document.createElement("p");

nuevoParrafo.innerHTML = "Este es un nuevo párrafo";

document.body.appendChild(nuevoParrafo);
```

## 3. Manipulación de Atributos:

Puedes cambiar los atributos de un elemento, como el atributo src de una imagen.

```
// Cambiar el atributo src de una imagen

let imagen = document.getElementById("miImagen");

imagen.src = "nueva-imagen.jpg";
```

## 4. Estilo Dinámico:

Puedes cambiar el estilo de un elemento dinámicamente mediante JavaScript.

```
// Cambiar el color de fondo de un elemento

titulo.style.backgroundColor = "lightblue";
```

```
// Añadir una nueva clase a un elemento

nuevoParrafo.classList.add("resaltado");
```

## 5. Eventos:

La manipulación del DOM también implica responder a eventos, como clics del mouse o pulsaciones de teclas.

```
// Agregar un evento de clic a un botón

let miBoton = document.getElementById("boton");

miBoton.addEventListener("click", function() {

    alert("¡Botón clickeado!");

});
```

La manipulación del DOM es esencial para crear aplicaciones interactivas y dinámicas en la web. Estos son solo algunos ejemplos básicos, pero hay muchas otras operaciones y métodos disponibles para manipular el DOM de manera más avanzada.