

Manipulación del DOM y CSS en JavaScript

La manipulación del DOM (Document Object Model) y el CSS (Cascading Style Sheets) en JavaScript te permite acceder y modificar los elementos HTML y sus estilos en una página web. El DOM proporciona una representación estructurada de los elementos de la página, y JavaScript te permite interactuar con ellos de diferentes maneras.

A continuación, exploraremos en detalle las diferentes formas de acceder a los elementos HTML y manipular su contenido y estilos mediante JavaScript:

Acceso a elementos HTML

1. **getElementById():** Este método permite acceder a un elemento por su identificador único (ID):

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

2. **getElementsByClassName():** Este método devuelve una lista de elementos que tienen la misma clase:

```
var elementos = document.getElementsByClassName("miClase");
```

3. **getElementsByTagName():** Este método devuelve una lista de elementos que tienen la misma etiqueta:

```
var elementos = document.getElementsByTagName("p");
```

4. **querySelector():** Este método devuelve el primer elemento que coincide con un selector CSS:

```
var elemento = document.querySelector("#miElemento");
```

5. **querySelectorAll():** Este método devuelve una lista de elementos que coinciden con un selector CSS:

```
var elementos = document.querySelectorAll(".miClase");
```

Manipulación de contenido

Una vez que has accedido a un elemento HTML, puedes manipular su contenido utilizando propiedades y métodos específicos. Algunos ejemplos comunes son:

- **innerHTML:** Esta propiedad te permite acceder o modificar el contenido HTML dentro de un elemento:

```
var elemento = document.getElementById("miElemento");  
elemento.innerHTML = "<p>Nuevo contenido</p>";
```

- **innerText:** Esta propiedad te permite acceder o modificar el texto dentro de un elemento (ignorando las etiquetas HTML):

```
var elemento = document.getElementById("miElemento");  
elemento.innerText = "Nuevo texto";
```

- **textContent:** Esta propiedad te permite acceder o modificar todo el contenido de texto, incluyendo espacios y saltos de línea:

```
var elemento = document.getElementById("miElemento");  
elemento.textContent = "Nuevo texto";
```

- **setAttribute():** Este método te permite establecer un atributo en un elemento:

```
var elemento = document.getElementById("miElemento");  
elemento.setAttribute("class", "nuevaClase");
```

Manipulación de estilos CSS

JavaScript también te permite manipular los estilos CSS de los elementos HTML. Puedes modificar directamente las propiedades de estilo utilizando la propiedad `style`, o agregar

o eliminar clases para aplicar estilos predefinidos.

- **style.property:** Puedes acceder y modificar propiedades de estilo directamente mediante la propiedad `style`:

```
var elemento = document.getElementById("miElemento");
elemento.style.backgroundColor = "red";
elemento.style.fontSize = "20px";
```

- **classList.add():** Este método te permite agregar una clase a un elemento:

```
var elemento = document.getElementById("miElemento");
elemento.classList.add("miClase");
```

- **classList.remove():** Este método te permite eliminar una clase de un elemento:

```
var elemento = document.getElementById("miElemento");
elemento.classList.remove("miClase");
```

- **classList.toggle():** Este método te permite alternar la presencia de una clase en un elemento. Si la clase está presente, se eliminará; de lo contrario, se agregará:

```
var elemento = document.getElementById("miElemento");
elemento.classList.toggle("miClase");
```

- **classList.contains():** Este método te permite verificar si un elemento tiene una clase específica:

```
var elemento = document.getElementById("miElemento");
if (elemento.classList.contains("miClase")) {
    // Realiza alguna acción si la clase está presente
}
```

Estos son solo algunos ejemplos de cómo puedes acceder y manipular los elementos HTML y sus estilos en JavaScript. La combinación de estas técnicas te permite crear páginas web interactivas y dinámicas, modificando el contenido y los estilos según sea necesario.

Recuerda que el DOM es una representación en memoria de la estructura de una página web, y mediante JavaScript puedes interactuar con los elementos y actualizarlos en tiempo

real. Experimenta con diferentes métodos y propiedades para lograr los efectos deseados en tu proyecto web.