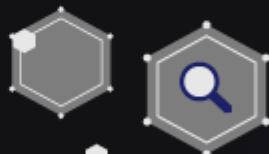


# LENGUAJES DE PROGRAMACIÓN

## JAVA SCRIPT





## DOM: Definición, Acceso y Manipulación de Elementos HTML desde JavaScript

El Modelo de Objeto de Documento (DOM) es una representación en memoria de la estructura de una página web. Esencialmente, es una interfaz de programación para documentos HTML y XML.

El DOM convierte todo el contenido de una página web en objetos, lo que permite a los programadores interactuar y manipular las páginas web. Cuando el navegador carga una página web, se crea automáticamente un Modelo de Objeto de Documento.

### Estructura del DOM

La estructura del DOM es similar a la de un árbol con varios nodos. La raíz del árbol es el documento en sí, y cada etiqueta de HTML es un nodo. Los nodos que tienen etiquetas dentro de ellos se consideran nodos padres de los nodos que representan las etiquetas internas.





## Acceso a los elementos del DOM

Para interactuar con una página web, primero debes acceder a los elementos del DOM. Existen varios métodos que te permiten hacerlo:

**`getElementById(id)`**: Este método obtiene un elemento por su atributo id. Solo puede haber un id único en un documento, por lo que este método solo retorna un solo elemento.

```
let titulo = document.getElementById('miTitulo');
```

**`getElementsByClassName(nombreClase)`**: Este método obtiene todos los elementos que tienen el nombre de clase especificado. Retorna una colección de elementos.

```
let elementosRojo = document.getElementsByClassName('rojo');
```

**`getElementsByTagName(tagName)`**: Este método obtiene todos los elementos que tienen el nombre de la etiqueta especificado. Retorna una colección de elementos.

```
let parrafos = document.getElementsByTagName('p');
```

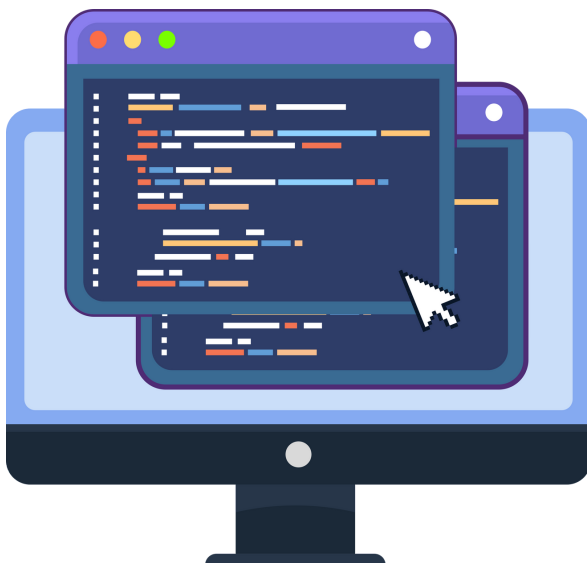


**`querySelector(selectorCSS)`**: Este método obtiene el primer elemento que coincida con el selector CSS especificado.

```
let primerElementoRojo = document.querySelector('.rojo');
```

**`querySelectorAll(selectorCSS)`**: Este método obtiene todos los elementos que coincidan con el selector CSS especificado. Retorna una colección de nodos.

```
let todosElementosRojo = document.querySelectorAll('.rojo');
```





## Manipulación de los elementos del DOM

Una vez que has accedido a los elementos del DOM, puedes modificarlos para cambiar el contenido de la página web. Algunas de las cosas que puedes hacer incluyen:

**Modificar el contenido de un elemento:** Puedes usar las propiedades ``textContent`` e ``innerHTML`` para cambiar el contenido de texto de un elemento.

```
titulo.textContent = 'Nuevo título'; // cambia el contenido de texto del elemento
```

**Modificar los atributos de un elemento:** Puedes cambiar los atributos de un elemento, como su id, clases, o estilos, usando los métodos ``setAttribute`` y ``removeAttribute``.

```
titulo.setAttribute('id', 'nuevold'); // cambia el id del elemento a 'nuevold'
```

**Modificar los estilos de un elemento:** Puedes modificar los estilos CSS de un elemento cambiando las propiedades de su propiedad ``style``.

```
titulo.style.color = 'red'; // cambia el color del texto del elemento a rojo
```

# LENGUAJES DE PROGRAMACIÓN

## JAVA SCRIPT



**Crear y agregar nuevos elementos al DOM:** Puedes crear nuevos elementos DOM y agregarlos a la página web usando los métodos ``createElement`` y ``appendChild``.

```
let nuevoParrafo = document.createElement('p'); // crea un nuevo elemento p
nuevo
```

```
Parrafo.textContent = 'Este es un nuevo párrafo'; // establece el contenido de texto del nuevo elemento
```

```
document.body.appendChild(nuevoParrafo); // añade el nuevo elemento al final del body
```

**-Eliminar elementos del DOM:** Puedes eliminar elementos existentes del DOM usando el método ``removeChild``.

```
document.body.removeChild(nuevoParrafo); // elimina el nuevo elemento del body
```





## Interacción con eventos

En JavaScript, puedes usar eventos para crear interactividad en tu página web. Los eventos son acciones del usuario que el navegador puede detectar, como hacer clic, presionar una tecla, mover el mouse, cargar una página, etc.

Para usar eventos en JavaScript, necesitas definir una función que se ejecutará cuando ocurra el evento (esta función se conoce como manejador de eventos) y luego necesitas asignar esa función al evento de un elemento específico.

Por ejemplo, podrías querer cambiar el color de un párrafo cuando el usuario haga clic en él. Aquí te muestro cómo podrías hacerlo:

```
let parrafo = document.querySelector('p');  
  
parrafo.onclick = function() {  
  parrafo.style.color = 'red';  
}
```

Aquí, la función anónima se asigna al evento `onclick` del párrafo. Cuando el usuario haga clic en el párrafo, se ejecutará la función y cambiará el color del texto del párrafo a rojo.