Manipulación del DOM con JS

Web Development

DOM con JS

Usar JavaScript para cambiar el contenido, la estructura o los estilos de una página web.

El DOM representa el documento HTML como una estructura de objetos, y mediante JavaScript podemos acceder y modificarlos.

Obtener elementos del DOM

Métodos más comunes:

- document.getElementById(id)

  Obtiene un solo elemento según su atributo id.
- document.querySelector(cssSelector)

  Obtiene el primer elemento que coincide con un selector CSS.
- document.querySelectorAll(cssSelector)

  Obtiene todos los elementos que coinciden con el selector CSS (como una lista tipo NodeList).
- document.getElementsByTagName(tagName)

  Obtiene todos los elementos con una etiqueta HTML específica (como p, div, h1…).
- document.getElementsByClassName(className)

  Obtiene todos los elementos con una clase específica.

Modificar o establecer atributos

- element.style.propiedad

  Cambia un estilo CSS directamente en línea.

  (Ejemplo: element.style.color = 'blue';)
- element.setAttribute('atributo', 'valor')

  Cambia o agrega un atributo HTML.

  (Ejemplo: img.setAttribute('src', 'foto.jpg');)
- element.textContent

  Cambia solo el texto dentro del elemento (sin etiquetas HTML).
- element.innerText

  Obtiene solo el texto visible que se muestra en pantalla.
- element.innerHTML

  Permite obtener o modificar el HTML interno del elemento (incluye etiquetas).

Modificar o establecer atributos

- element.attribute

  Se puede acceder directamente a los atributos como propiedades.

  (Ejemplo: input.value obtiene el valor del campo de texto).

- element.classList

  Permite manejar las clases CSS del elemento con:

- add('clase') = agrega una clase.
- remove('clase') = elimina una clase.
- toggle('clase') = alterna entre agregar y quitar una clase.

Agregar y eliminar elementos

- document.createElement(tagName)

  Crea un nuevo elemento HTML.

- document.createTextNode(texto)

  Crea un nodo de texto (como alternativa a textContent).

- document.createDocumentFragment()

  Crea un fragmento de documento para insertar múltiples elementos a la vez (útil en bucles).

- element.appendChild(elemento)

  Agrega un elemento al final del contenido de otro.

- element.append(nodo1, nodo2, …)

  Agrega uno o más nodos (elementos o texto) al final.

- element.prepend(nodo1, nodo2, …)

  Agrega uno o más nodos al inicio del elemento.

- element.remove()

  Elimina el elemento del DOM.

Tamaños y desplazamiento (Scrolling)

- window.innerWidth = ancho visible de la ventana del navegador.
- window.innerHeight = alto visible de la ventana del navegador.
- window.getComputedStyle(element) = obtiene los estilos aplicados (en píxeles).
- element.clientHeight = altura visible del contenido y relleno (padding).
- element.offsetHeight = altura total del elemento (incluye bordes).
- element.scrollHeight = altura total del contenido, incluyendo la parte desplazada.
- element.offsetTop = distancia entre el borde superior del elemento y su contenedor.
- element.scrollIntoView()

  Desplaza la vista para que el elemento sea visible.

- element.scrollTo({top, behavior: 'smooth'})

  Desplaza la vista a una posición específica.

El parámetro behavior: 'smooth' crea un desplazamiento suave.