

NODE JS

Clase 16: Manejo del DOM



... Clase 16

01

Conceptos
generales

02

Características
del DOM

03

Estructura del
DOM

04

JS- HTML-DOM
Métodos



Nación
Servicios

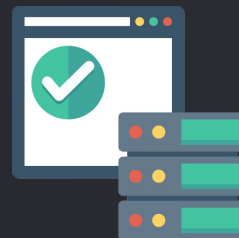


Conceptos Generales

A grayscale photograph of four people (two men and two women) sitting around a table in a library, working together. They are looking at a laptop and some papers. The background is filled with bookshelves. The text "Conceptos Generales" is overlaid in a bright orange color.

... DOM

El **modelo de objetos de documento (DOM)** del W3C es una plataforma y una interfaz de lenguaje neutral que permite que los programas y scripts accedan y actualicen dinámicamente el contenido, la estructura y el estilo de un documento.



... DOM

Representa la estructura de una página web como un árbol jerárquico de objetos, donde cada objeto corresponde a una parte del contenido o la estructura de la página, como elementos, atributos y texto.

... DOM

El **DOM HTML** representa la forma en que los documentos HTML están estructurados en memoria cuando se cargan en un navegador web o se manipulan mediante JavaScript.

... DOM

Un documento HTML está compuesto por varios elementos, como encabezados, párrafos, listas, imágenes, enlaces, formularios y más.

Estos elementos se definen utilizando etiquetas como `<h1>`, `<p>`, ``, ``, `<a>`, `<form>`

... Estructura de Árbol

Cuando se carga una página web en un navegador, este crea una representación de la estructura de la página como un árbol de objetos. Cada elemento HTML corresponde a un nodo en el árbol DOM.

... Estructura de Árbol

La **raíz del árbol** es el elemento `<html>`, y sus nodos hijos podrían ser `<head>` y `<body>`.

Dentro del elemento `<body>`, tendrás nodos adicionales que representan el contenido y la estructura de la página.

Características del DOM

... Tipos de Nodos

Los **nodos** son los elementos individuales en el DOM. Hay varios tipos de nodos, incluyendo elementos HTML, atributos, texto, comentarios.

... Tipos de Nodos

Cada nodo tiene propiedades y métodos que permiten interactuar con él.

Los nodos del DOM pueden ser de varios tipos. Los tipos principales relacionados con HTML incluyen:

... Tipos de Nodos

Nodos de Elementos

Los **nodos de elemento** representan las etiquetas HTML en el documento. Cada elemento tiene propiedades que reflejan los atributos y contenido del elemento.

... Tipos de Nodos

Nodos de Texto

Representan el contenido de texto dentro de un elemento, son elementos esenciales en la representación programática de documentos HTML o XML en el modelo de objetos del documento.

... Tipos de Nodos

Nodos de Atributo

Representan los atributos de un elemento.

Cada elemento HTML o XML puede tener atributos que proporcionan información adicional sobre el elemento. Estos atributos se representan como nodos en el árbol DOM. Cada nodo de atributo contiene un nombre y un valor.

... Tipos de Nodos

Nodos de Comentario

Representan los comentarios en el código HTML.

... Acceso y Manipulación

El DOM permite a los desarrolladores acceder y manipular los elementos y contenido de una página web usando JavaScript. Esto permite la creación de páginas interactivas y dinámicas.

... **Carga Progresiva**

Los navegadores cargan el DOM de manera progresiva a medida que se descarga y analiza el documento HTML. Esto significa que los scripts pueden acceder y manipular partes del DOM incluso antes de que se haya cargado toda la página.

... Eventos

El DOM también maneja **eventos**, como clics de mouse, pulsaciones de teclas y cambios en el estado de la página. Los eventos pueden ser detectados y manejados mediante JavaScript para ejecutar acciones específicas en respuesta a la interacción del usuario.

Estructura del DOM



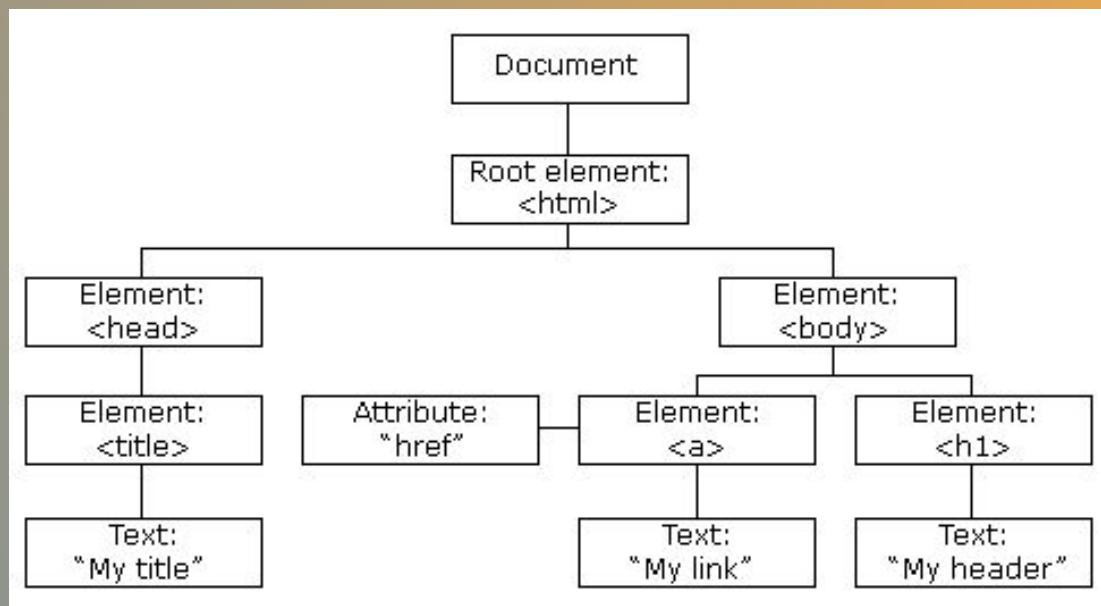
... Estructura del DOM

La **estructura del DOM** se asemeja a un árbol, donde los nodos superiores tienen nodos hijos y, a su vez, pueden tener nodos hijos adicionales.

Estructura del DOM



Nación
Servicios



... Estructura del DOM

La **manipulación del DOM** mediante JavaScript permite a los desarrolladores acceder, modificar y actualizar elementos y contenido en tiempo real, lo que facilita la creación de interacciones dinámicas en páginas web y aplicaciones.

JavaScript, HTML y DOM



... JavaScript y el DOM

Con el modelo de objetos con JavaScript es posible manejar y crear HTML dinámico.

... JavaScript y el DOM

- JavaScript puede cambiar todos los **elementos** HTML en la página.
- JavaScript puede cambiar todos los **atributos** HTML en la página.
- JavaScript puede cambiar todos los **estilos** CSS en la página.

... JavaScript y el DOM

- JavaScript puede eliminar **elementos y atributos** HTML existentes.
- JavaScript puede agregar nuevos **elementos y atributos** HTML.
- JavaScript puede reaccionar a todos los **eventos** HTML existentes en la página.
- JavaScript puede crear **nuevos eventos** HTML en la página.

Métodos HTML DOM



... Métodos

- `document.getElementById(id)`
- Busca y devuelve un elemento por su atributo id
- Uso: `var element = document.getElementById("miElemento");`

Métodos

```
// Obtener una referencia a un elemento por su ID  
const miElemento = document.getElementById('miElemento');
```

... Métodos

- `document.querySelector(selector)`
- **Descripción:** Devuelve el primer elemento que coincide con el selector CSS especificado.
- **Uso:** `var element = document.querySelector(".miClase");`

... Métodos

- `element.textContent`
- **Descripción:** Obtiene o establece el contenido de texto plano dentro de un elemento.
- **Uso:** `var texto = element.textContent;` o `element.textContent = "Nuevo texto";`

Métodos

```
const miElemento = document.getElementById('miElemento');  
  
// Cambiar el contenido de texto del elemento  
miElemento.textContent = 'Nuevo contenido';
```

... Métodos

- `element.setAttribute(name, value)`
- **Descripción:** Establece el valor de un atributo en un elemento.
- **Uso:** `element.setAttribute("class", "nuevaClase");`

... Métodos

- `element.getAttribute(name)`
- **Descripción:** Obtiene el valor de un atributo en un elemento.
- **Uso:** `var valor = element.getAttribute("href");`

... Métodos

- `element.classList.add(className)`
- **Descripción:** Agrega una clase CSS al elemento.
- **Uso:** `element.classList.add("nuevaClase");`

... Métodos

- `element.classList.remove(className)`
- **Descripción:** Elimina una clase CSS del elemento.
- **Uso:** `element.classList.remove("viejaClase");`

... Métodos

- `element.appendChild(childElement)`
- **Descripción:** Agrega un elemento hijo al final de la lista de hijos del elemento padre.
- **Uso:** `parentElement.appendChild(childElement);`

Métodos

```
// Crear un nuevo elemento
const nuevoElemento = document.createElement('div');
nuevoElemento.textContent = 'Este es un nuevo elemento';

// Agregar el nuevo elemento al documento
document.body.appendChild(nuevoElemento);
```

... Métodos

- `element.removeChild(childElement)`
- **Descripción:** Elimina un elemento hijo del elemento padre.
- **Uso:** `parentElement.removeChild(childElement);`

... Métodos

- `element.addEventListener(event, callback)`
- **Descripción:** Agrega un escuchador de eventos al elemento.
- **Uso:** `element.addEventListener("click", miFuncion);`

Métodos

```
b.addEventListener('click', () => {  
  cambiarTituloParrafo(titulos[index],parrafos[index]);  
})
```

... Métodos

- `window.onload = function() {}`
- **Descripción:** Ejecuta una función cuando todo el contenido de la página y los recursos externos se han cargado.
- **Uso:** `window.onload = function() { /* código a ejecutar */ };`

¡Muchas gracias!

