En breve iniciamos

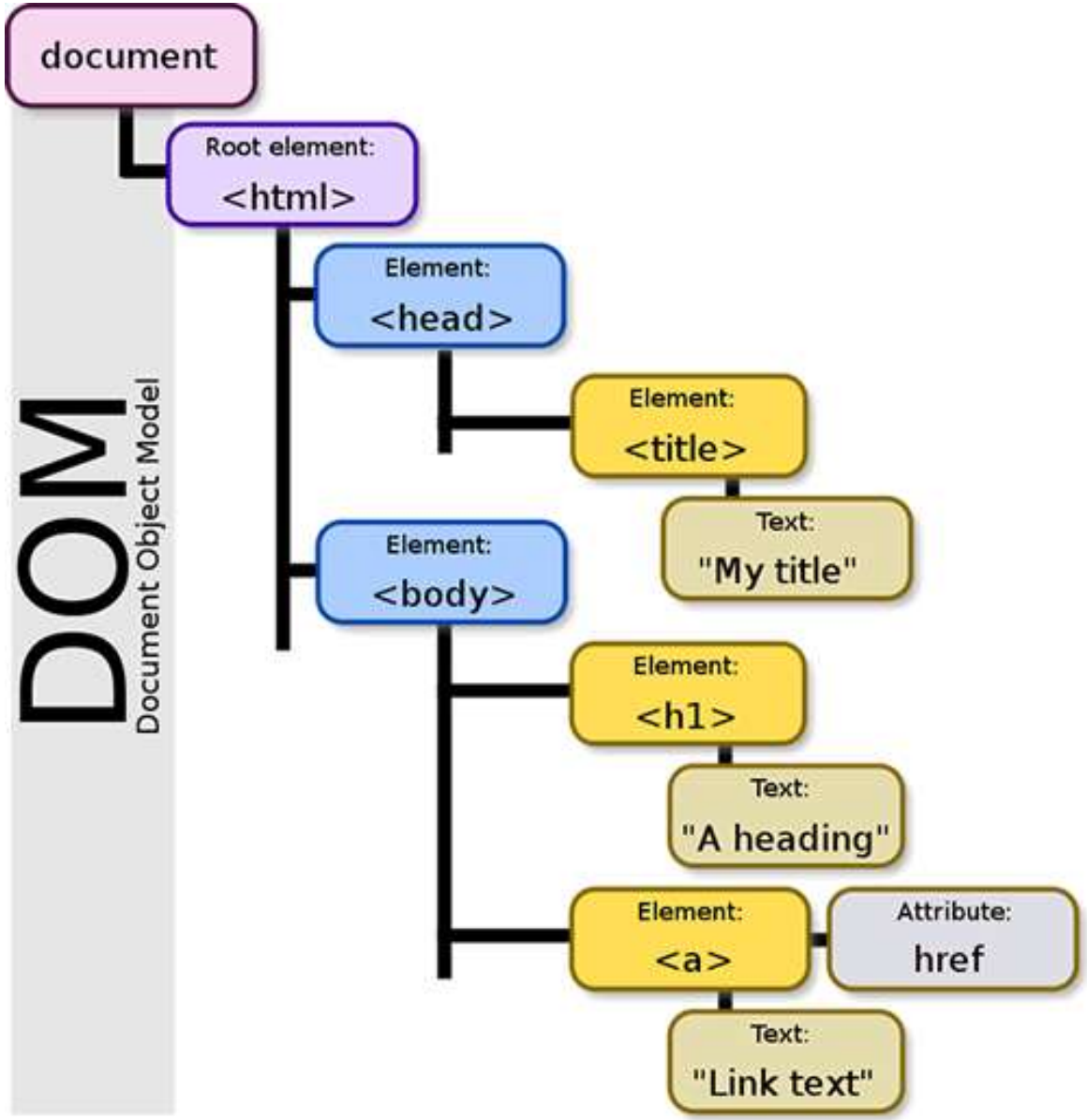
Manipulación del DOM. **JavaScript.**

@SENAComunica

www.sena.edu.co

Nelson E. Rincón C.
Instructor – Líder.
ADSO.

Que es DOM



Document Object Model se refiere a la capacidad de modificar la estructura, contenido y estilo de un documento HTML utilizando JavaScript. JavaScript proporciona métodos y propiedades para acceder y modificar esos elementos. El DOM es una representación en forma de árbol de todos los elementos HTML de una página web.

SENA
Nelson E. Rincón C.
Instructor - ADSO.

DOM en JavaScript – getElementById()

```html
<body>
    <div>
        <h1 id="titulo"> saludo </h1>
        <p id="mensaje">Mensaje al cliente</p>
        <button onclick="cambiarTexto()">Cambiar</button>
    </div>
</body>
```

```html
<script>
    function cambiarTexto(){
        var elemento=document.getElementById("titulo");
        //cambia el texto
        elemento.innerHTML="Hola Mundo.....";
        cambiarMensaje("Nelson Rincon");
    }
    function cambiarMensaje(nombreCliente){
        let elemento=document.getElementById("mensaje");
        //Cambiar el mensaje
        elemento.innerHTML="Saludo desde el mundo del javaScript";
    }
</script>
```

SENA
Nelson E. Rincón C.
Instructor - ADSO.

DOM en JavaScript – getElementsByClassName()

```html
<body>
    <div class="caja">
        <h1 id="titulo"> saludo </h1>
        <p id="mensaje">Mensaje al cliente</p>
        <button onclick="cambiarTexto()">Cambiar</button>
    </div>
    <div class="caja">caja 2</div>
    <div class="caja">caja 3</div>
    <div class="caja">caja 4</div>
</body>
```

```html
<script>
    var datoId=document.getElementById("titulo");
    console.log(datoId);
    var datoClassName=document.getElementsByClassName("caja");
    console.log(datoClassName);
    console.log(typeof(datoClassName));
    for (let i=0; i<datoClassName.length; i++){
        console.log(datoClassName[i]);
    }
</script>
```

Nelson E. Rincón C.
Instructor - ADSO.

DOM en JavaScript

Las operaciones más comunes para manipular el DOM en JavaScript son:

1. Acceder a elementos:

- document.getElementById(id): Obtiene un elemento por su ID.
- document.getElementsByClassName(className): Obtiene una lista de elementos por su clase.
- document.getElementsByTagName(tagName): Obtiene una lista de elementos por su etiqueta.
- document.querySelector(selector): Obtiene el primer elemento que coincide con el selector CSS especificado.
- document.querySelectorAll(selector): Obtiene una lista de todos los elementos que coinciden con el selector CSS especificado.

2. Modificar contenido:

- .innerHTML: Permite obtener o establecer el contenido HTML de un elemento (incluye TAG).
- .textContent: Permite obtener o establecer el contenido de texto de un elemento.
- .innerText: Permite obtener o establecer el contenido de texto de un elemento, sin incluir las etiquetas HTML internas.
- .outerHTML: Permite obtener o establecer el contenido HTML completo de un elemento, incluyendo el propio elemento.

SENA
Nelson E. Rincón C.
Instructor - ADSO.

DOM en JavaScript – *Modificar contenido*

```html
<script>
    var caja2=document.getElementById("caja2");
    //obtener los elementos del HTML
    var datoCaja2=caja2.innerHTML;
    console.log("info es: ", datoCaja2);
    console.log(typeof(datoCaja2));
    //reemplazar o asignar informacion
    caja2.innerHTML="<h1> Nuevo contenido Caja2</h1>";
    nuevoDato=caja2.innerHTML;
    console.log(nuevoDato);
    console.log(`El solo texto: ${caja2.innerText}`);
    //obtener el nuevo dato
    nuevoDato=caja2.textContent;
    console.log(nuevoDato);
    caja2.textContent="Nuevo Reemplazo solo Texto";
    nuevoContenido=caja2.textContent;
    console.log(nuevoContenido);
</script>
```

```html
<body>
    <div class="caja">
        <h1 id="titulo"> saludo </h1>
        <p id="mensaje">Mensaje al cliente</p>
        <button onclick="cambiarTexto()">Cambiar</button>
    </div>
    <div class="caja" id="caja2">caja 2</div>
    <div class="caja">caja 3</div>
    <div class="caja">caja 4</div>
</body>
```

```
info es:  caja 2
string
<h1> Nuevo contenido Caja2</h1>
El solo texto: Nuevo contenido Caja2
 Nuevo contenido Caja2
Nuevo Reemplazo solo Texto
```

SENA
Nelson E. Rincón C.
Instructor - ADSO.

DOM en JavaScript

3. Modificar atributos:

- element.getAttribute(attribute): Obtiene el valor de un atributo específico de un elemento.
- element.setAttribute(attribute, value): Establece el valor de un atributo específico de un elemento (div-> id, class, style, title, data, lang, dir, tabindex accesskey, role. Que no sean atributos de CSS
- element.removeAttribute(attribute): Elimina un atributo específico de un elemento.
- element.hasAttribute(attribute): Devuelve un valor booleano que indica si un elemento tiene el atributo especificado.

4. Modificar estilos:

- element.style.property: Permite obtener o establecer el valor de una propiedad de estilo de un elemento.
- element.classList: Proporciona métodos para agregar, eliminar y verificar clases CSS en un elemento.
- element.classList.add(className): Agrega una clase CSS al elemento.
- element.classList.remove(className): Elimina una clase CSS del elemento.
- element.classList.toggle(className): Agrega la clase CSS al elemento si no está presente, y la elimina si ya está presente.
- element.style.cssText: Permite obtener o establecer el texto completo de los estilos en línea de un elemento.

SENA
Nelson E. Rincón C.
Instructor - ADSO.

DOM en JavaScript

```html
<body>
    <div class="central">
        <div class="caja">caja </div>
        <div>
            <button onclick="cambio()">
                dar click
            </button>
        </div>
    </div>
```

```css
<style>
    .central{
        position: relative;
        display: flex;
        width: 50%;
        flex-direction: column;
    }
    .caja{
        width: 300px;
        height: 300px;
    }
    .caja_nueva{
        width: 300px;
        height: 300px;
        background-color: red;
```

```javascript
<script>
    function cambio(){
        //capturamos el DOM--> HTML
        var caja=document.getElementsByClassName("caja")[0];
        //capturamos el atriburo del HTML
        atributo_caja=caja.getAttribute("class");
        //validamos si es verdadero o falso
        dato=caja.hasAttribute("class");
        if (atributo_caja=="caja" && dato==true){
            //Cambiamos los atriburos del Class
            caja.setAttribute("class", "caja_nueva");
            //console.log(caja, dato, atributo_caja);
        }
        aux=caja.getAttribute("class");
        caja_nueva=document.getElementsByClassName("caja_nueva")[0];
        if(aux=="caja_nueva") {
            //eliminamos el atributo del HTML
            caja_nueva.removeAttribute("aux");
        }
    }
</script>
```

DOM en JavaScript

5. Crear y manipular elementos:

- document.createElement(tagName): Crea un nuevo elemento con la etiqueta especificada.
- element.appendChild(childElement): Agrega un elemento secundario al final de la lista de hijos de un elemento.
- element.removeChild(childElement): Elimina un elemento secundario de un elemento.
- element.replaceChild(newChild, oldChild): Reemplaza un elemento secundario por otro.
- parentElement.insertBefore(newElement, referenceElement): Inserta un nuevo elemento antes de otro elemento específico como su hermano.
- parentElement.cloneNode(deep): Crea una copia superficial o profunda del elemento y todos sus descendientes.
- element.setAttributeNS(namespace, attribute, value): Establece el valor de un atributo específico con un espacio de nombres en un elemento.
- element.hasChildNodes(): Devuelve un valor booleano que indica si un elemento tiene nodos hijos.

SENA
Nelson E. Rincón C.
Instructor - ADSO.

DOM en JavaScript

```
<script>
    function crearItems(){
        //Tomo el elemento DOM HTML
        var lista=document.getElementById("listas");
        // Se crea el hijo del elemento tomado
        item=document.createElement("li");
        // se anexa el texto que va contener el hijo
        item.innerHTML="<h3> hola </h3>";
        // se agrega el hijo en el DOM
        lista.appendChild(item);
    }
</script>
```

```
function eliminarItems(){
    // se captura el DOM html
    var elemento=document.getElementById("listas");
    // se toma DOM padre que sea eliminar
    // se le indica el hijo que se elimna.
    elemento.removeChild(elemento.lastChild);
}
```

```
<body>
    <button>crear</button>
    <button>eliminar</button>
    <ul>
        <li id="listas">lista de elementos.</li>
    </ul>
</body>
<script>
    var saludos="NELSON"
    var boton=document.getElementsByTagName("button");
    boton[0].addEventListener("click", eventoClick);
    boton[1].addEventListener("click", eliminarItems);
    console.log(boton);

    function eventoClick(){
        alert("Click en el boton " + saludos);
        crearItems();
    }
```

SENA
Nelson E. Rincón C.
Instructor - ADSO.

DOM en JavaScript

6. Eventos desde DOM:

- Captura de Evento: addEventListener(evento, función, [opciones])→ Permite adjuntar un evento a un elemento del DOM y especificar una función que se ejecutará cuando ocurra el evento.
- Remueve evento: removeEventListener(evento, función, [opciones])→ Elimina un evento previamente adjuntado a un elemento del DOM. Se deben proporcionar los mismos argumentos que se utilizaron al adjuntar el evento.

7. Eventos desde el atributo del elemento HTML:

- Onclick→ Es un atributo que se puede asignar a un elemento HTML del DOM para especificar la función que se ejecutará cuando se haga clic en ese elemento.
- Onchange→ Se utiliza en elementos de entrada (input, select, textarea) y se dispara cuando el valor del elemento cambia y se pierde el foco.
- onmouseover y onmouseout → Estos eventos se disparan cuando el puntero del mouse se mueve sobre un elemento (onmouseover) o se sale de un elemento (onmouseout).
- onkeydown, onkeyup y onkeypress → Estos eventos se disparan cuando se presiona (onkeydown), se suelta (onkeyup) o se mantiene presionada (onkeypress) una tecla del teclado.

SENA
Nelson E. Rincón C.
Instructor - ADSO.

Eventos desde DOM - JavaScript

Eventos son una acción que ocurre en el navegador, que puede ser ocasionado por el usuario, sistema o alguna otra aplicación, los eventos más comunes son por teclado, mouse, formularios, ventanas y /o personalizados.

```
<script>
    var saludos="NELSON"
    var boton=document.getElementsByTagName("button");
    boton[0].addEventListener("click", function(){
        alert("Click en el boton" + saludos);
    });
    console.log(boton);
</script>
```

Uso de función Anónima

Uso de función Normal

```
<script>
    var saludos="NELSON"
    var boton=document.getElementsByTagName("button");
    boton[0].addEventListener("click", eventoClick);
    console.log(boton);

    function eventoClick(){
        alert("Click en el boton " + saludos);

    }
</script>
```

SENA
Nelson E. Rincón C.
Instructor - ADSO.

Ejercicio DOM en JavaScript

- Escribe un programa que valide un formulario de registro. El formulario debe contener campos para el nombre, el correo electrónico y la contraseña. Utiliza JavaScript para capturar los valores ingresados en los campos del formulario y validarlos. Si algún campo está vacío o la contraseña es demasiado débil (por ejemplo, menos de 6 caracteres), muestra un mensaje de error. De lo contrario, almacena la información del formulario e imprimir en consola y los mensajes deben crear directamente en cajas contenedoras del html (div).

- Escribir un programa para crear una lista de tareas interactiva. El formulario debe tener un campo de texto para ingresar una tarea y un botón para agregarla a la lista. Al hacer clic en el botón, captura el valor del campo de texto y agrega la tarea a una lista de html (ul, ol, li). muestra la lista de tareas en el DOM. Además, implementa la funcionalidad para marcar una tarea como completada o eliminar una tarea de la lista.

- Implementa una función de búsqueda en una tabla HTML (listas). El formulario debe tener un campo de texto para ingresar el término de búsqueda y un botón para realizar la búsqueda. Al hacer clic en el botón, captura el valor del campo de texto y busca en la tabla las filas que coincidan con el término ingresado. Dentro del HTML debe tener un contenedor de resultado indicando la cantidad de veces que se repite la palabra dentro de la tabla diccionario.

SENA
Nelson E. Rincón C.
Instructor - ADSO.

G R A C I A S

Línea de atención al ciudadano: 01 8000 910270
Línea de atención al empresario: 01 8000 910682

@SENAComunica

www.sena.edu.co