7.7 Clase en vivo - Profesores

Status | Done

Unidad 7: DOM y Eventos en JavaScript

Ejemplo en vivo

Instrucción para el profesor:

En un documento HTML, tenemos un tag <h1> con el id “titulo”, y un tag lista desordenada con el id “listado”. Además, tenemos el siguiente array: const frutas = [‘Ananá’, ‘Banana’, ‘Durazno’, ‘Kiwi’, ‘Manzana’, ‘Papaya’, ‘Pera’]

Crearemos una función llamada cargarDOM(), la cual se conectará con ambos elementos HTML, y asignará el valor “www.mandafruta.com” en el título, mientras que generará los list ítems con cada una de las frutas, en el elemento HTML de lista desordenada.

```
// Array de frutas
const frutas = ['Ananá', 'Banana', 'Durazno', 'Kiwi', 'Manzana',

// Función que actualiza el DOM
function cargarDOM() {
    // Conectar y actualizar el título
    const tituloElemento = document.getElementById('titulo');
    tituloElemento.textContent = 'www.mandafruta.com';

    // Conectar y generar los elementos de la lista desordenada
    const listaElemento = document.getElementById('listado');

    // Limpiar cualquier contenido previo de la lista
    listaElemento.innerHTML = '';

    // Generar los ítems de la lista con cada fruta del array
```

```
    frutas.forEach(fruta => {
        // Crear un nuevo elemento de lista
        const li = document.createElement('li');
        // Asignar el texto de la fruta al elemento de lista
        li.textContent = fruta;
        // Agregar el elemento de lista a la lista desordenada
        listaElemento.appendChild(li);
    });
}

// Llamar a la función para cargar el DOM
cargarDOM();
```

Actividad: Find The Bug

Instrucción para el profesor:

```
function cargarDOM() {
const titulo = document.querySelector("titulo")
const listado = document.querySelector(".listado")
titulo.innerText = "www.mandafruta.com"
frutas.forEach(fruta => {
listado.innerHTML += '<li>{fruta}</li>`
})
}
```

Preguntas guía:

- ¿En qué se asemeja al código del ejercicio anterior? ¿En qué se diferencia?
- ¿Qué debería ocurrir? ¿Cuáles son las instrucciones que da este código?
- ¿Por qué eso no sucede?
- ¿Cómo harías para lograr que el código se ejecute con éxito?

Actividad: Modificación de Nodos con `innerHTML` e `innerText`

Posible Resolución:

```
let app = document.getElementById("app");
app.innerHTML = `
    <h1 id="titulo">Hola Mundo!</h1>
    <h2>Subtítulo agregado</h2>
    <p>Este es un nuevo párrafo</p>
`;

let titulo = document.getElementById("titulo");
titulo.innerText = "Hola Coder!";
```

Instrucciones para el Profesor:

- Explica la diferencia entre `innerHTML` e `innerText` y cuándo es más conveniente usar cada uno.
- Muestra en vivo cómo cambiar el contenido del `div` con `id="app"` usando `innerHTML`.
- Recalca la seguridad en el uso de `innerHTML`, mencionando el riesgo de inyección de código.

Actividad: Uso de `querySelector` y `querySelectorAll`

Posible Resolución:

```
let primerParrafo = document.querySelector("#contenedor .texto");
primerParrafo.innerText = "Texto modificado con querySelector";

let todosLosParrafos = document.querySelectorAll("#contenedor .texto");
```

```
todosLosParrafos.forEach(parrafo => {
    parrafo.innerText = "Texto modificado con querySelectorAl
l";
});
```

Instrucciones para el Profesor:

- Explica cómo `querySelector` selecciona el primer elemento que coincide con el selector y cómo `querySelectorAll` devuelve todos los elementos coincidentes.
- Realiza un ejemplo en vivo usando `querySelector` para seleccionar y modificar un elemento específico.
- Pregunta a los estudiantes si tienen alguna duda respecto a la sintaxis de los selectores CSS utilizados.

Actividad: Manejo de Eventos con `addEventListener`

Posible Resolución:

```
let boton = document.getElementById("btnPrincipal");

boton.addEventListener("click", () => {
    console.log("Botón clickeado");
});

boton.addEventListener("mouseover", () => {
    boton.innerText = "Mouse sobre mí";
});
```

Instrucciones para el Profesor:

- Explica cómo `addEventListener` permite asignar múltiples eventos a un mismo elemento.
- Realiza en vivo la adición del evento `click` y muestra cómo se ejecuta al hacer clic en el botón.

- Resalta la diferencia entre manejar eventos con `addEventListener` y propiedades de evento (`onclick`).

Preclase - 30 minutos previos

Repaso de Conceptos Clave (10 minutos)

- **Estructura del DOM:** Breve revisión de cómo HTML se convierte en una estructura jerárquica de nodos, incluyendo tipos de nodos como `Element`, `Text`, y `Comment`.
- **Acceso al DOM:** Explicación rápida de los métodos más comunes como `getElementById()`, `getElementsByClassName()`, `getElementsByTagName()`, y `querySelector()`.
- **Modificar Nodos:** Uso de propiedades como `innerHTML`, `innerText`, y `className` para manipular el contenido y atributos de los nodos.
- **Eventos en JavaScript:** Introducción a los eventos y su manejo con `addEventListener()` y otras formas de definir eventos, como eventos del mouse (`click`, `mouseover`) y del teclado (`keydown`, `keyup`).

Espacio para Preguntas y Respuestas (15 minutos)

- Abrir el espacio para que los estudiantes planteen preguntas sobre la manipulación del DOM y el uso de eventos.
- Resolver dudas con ejemplos en vivo si es necesario, como agregar eventos a elementos dinámicamente o modificar el contenido del DOM en respuesta a eventos.
- Si no hay preguntas, sugerir temas comunes como la diferencia entre `innerHTML` y `textContent`, o cómo prevenir el comportamiento predeterminado de un formulario con `preventDefault()`.

Demostraciones Prácticas (Opcional, dependiendo de las preguntas)

- Realizar demostraciones en vivo si las preguntas lo requieren, por ejemplo, mostrar cómo usar `querySelectorAll()` para seleccionar múltiples elementos y aplicar cambios masivos.
- Ejemplos posibles: Crear un evento `click` en un botón que añada elementos a una lista, o usar `addEventListener()` para capturar entradas de texto en tiempo

real.