

Resumen rápido

La sesión se centró en la manipulación del DOM usando JavaScript, abarcando desde la selección y modificación de elementos hasta la creación dinámica de contenido HTML. Se explicaron conceptos clave como la diferencia entre `textContent` e `innerHTML`, cómo modificar estilos y la importancia de entender el DOM como una entidad separada del documento HTML original. Finalmente, se asignaron ejercicios prácticos para que los estudiantes aplicaran lo aprendido y se anunció que la próxima clase se enfocará en eventos.

Siguientes pasos

Estudiantes: Completar los 8 ejercicios sobre manipulación del DOM utilizando JavaScript.

Estudiantes: Responder a las valoraciones de la clase en el campus.

Estudiantes: Revisar el tema de eventos para la próxima clase del miércoles.

EBIS: Preparar la clase del miércoles sobre eventos en JavaScript.

Sergio: Investigar cómo implementar un observador para solucionar el problema con la librería de imágenes 360°.

Resumen

DOM Y JavaScript Interacción

El resumen de esta parte de la reunión es el siguiente: EBIS explica el concepto del DOM (Document Object Model) y cómo interactuar con él usando JavaScript. Comienza definiendo el DOM como una representación virtual del documento HTML que permite manipularlo dinámicamente. Luego demuestra cómo seleccionar elementos del DOM usando diferentes métodos como `getElementById` y `querySelector`. EBIS muestra ejemplos prácticos de cómo acceder y modificar propiedades de los elementos seleccionados, como el `textContent`. También explica la diferencia entre `querySelector` y `querySelectorAll`, donde el segundo devuelve un array de todos los elementos que coinciden con el selector. A lo largo de la explicación, EBIS enfatiza la importancia de entender que el DOM es una entidad separada del documento HTML original y cómo JavaScript puede interactuar con él para crear páginas web dinámicas.

Inserting DOM Elements With JavaScript.

El instructor explica cómo crear y modificar elementos del DOM usando JavaScript. Se enfatiza que crear un elemento con `createElement()` no lo inserta automáticamente en el DOM, sino que debe usarse `appendChild()` para añadirlo a un elemento padre. Se muestra cómo crear un párrafo, modificar su contenido con `textContent`, y luego insertarlo en el DOM. El instructor recomienda crear una función para reutilizar el código si se necesita insertar elementos frecuentemente. También se menciona brevemente cómo insertar contenido más complejo como listas HTML dentro de otros elementos.

Diferencia Entre textContent E innerHTML.

El instructor explica la diferencia entre textContent e innerHTML al modificar el contenido de elementos HTML. Mientras textContent solo inserta texto plano sin interpretar etiquetas, innerHTML permite insertar y renderizar HTML completo, incluyendo etiquetas y scripts. Se advierte que usar innerHTML con contenido generado por usuarios puede crear vulnerabilidades de seguridad como inyección de código. Por eso, se recomienda usar textContent cuando no se tiene control total sobre el contenido a insertar, ya que este método no interpreta el código y lo inserta como texto simple.

Generando Dinámicamente Tarjetas HTML.

El instructor explica cómo generar dinámicamente tarjetas HTML a partir de un array de objetos usando JavaScript. Crea una función que recibe los datos de cada objeto (imagen, nombre y texto) y devuelve un string con el HTML de la tarjeta. Luego itera sobre el array usando forEach, llamando a la función para cada elemento y concatenando los resultados en una variable. Finalmente, inserta todo el HTML generado en el DOM usando innerHTML. Demuestra cómo esto permite crear contenido web dinámicamente a partir de datos, lo cual es útil para mostrar información que cambia, como una lista de estudiantes que se actualiza cada año.

Modificando Estilos Del DOM Con JavaScript.

El instructor explica cómo modificar los estilos de los elementos del DOM usando JavaScript. Menciona que se puede acceder a las propiedades de estilo a través del objeto "style" de cada elemento, pero que los nombres de las propiedades deben escribirse en camelCase en lugar de kebab-case. También demuestra cómo agregar estilos inline directamente al generar elementos HTML, lo cual es más eficiente que modificarlos después. Finalmente, ejemplifica esto aplicando colores de fondo a unas tarjetas generadas dinámicamente.

Manipulación Del DOM Con JavaScript.

EBIS explica cómo manipular el DOM usando JavaScript, incluyendo cómo agregar, modificar y eliminar elementos. Destaca la importancia de este tema, señalando que es la primera vez que los estudiantes están realmente programando una página web con lógica JavaScript. EBIS anuncia que en la próxima clase se abordarán los eventos. Finalmente, menciona que ha compartido ejercicios para que los estudiantes practiquen y recuerda la importancia de completar las evaluaciones de la clase. Sergio plantea una pregunta sobre observadores y watchers, lo que lleva a una breve discusión sobre su uso en diferentes contextos de desarrollo web.

JavaScript Para Crear HTML

EBIS presenta ocho ejercicios de JavaScript para crear HTML de forma programática. Los ejercicios incluyen crear títulos y párrafos, usar variables para mostrar mensajes personalizados, solicitar datos al usuario para realizar cálculos y mostrar resultados, y crear una página web completa con JavaScript. EBIS aclara que los datos del usuario deben recogerse con `window.prompt` y no con formularios HTML. Los estudiantes tienen hasta las 9 para trabajar en los ejercicios y hacer preguntas, y la próxima clase será el miércoles.