

Resumen rápido

EBIS proporcionó una introducción a JavaScript, explicando su historia, evolución y importancia en el desarrollo web moderno. También discutió el estándar ECMAScript, la evolución de JavaScript y la importancia de Node.js en el desarrollo web. Por último, cubrió conceptos básicos de JavaScript, incluyendo sintaxis, ejecución secuencial, estructuras condicionales y bucles, y enfatizó la importancia de la práctica y la documentación en el aprendizaje de la programación.

Siguientes pasos

Estudiantes: Practicar y familiarizarse con la sintaxis básica de JavaScript.

Estudiantes: Investigar y documentarse sobre conceptos de JavaScript no cubiertos en clase.

Estudiantes: Completar los ejercicios proporcionados en el PDF de la unidad 4.

Estudiantes: Evitar el uso de ChatGPT y Copilot para resolver los ejercicios.

Estudiantes: Desarrollar habilidades de búsqueda efectiva de información sobre programación.

Estudiantes: Repasar el contenido del PDF proporcionado como resumen de la clase.

Estudiantes: Prepararse para la próxima clase del lunes sobre tipos de datos en JavaScript.

EBIS: Revisar brevemente los ejercicios en la próxima clase del lunes.

EBIS: Preparar ejercicios más específicos para corregir en las próximas clases.

Estudiantes: Practicar el uso de la consola del navegador para ejecutar código JavaScript.

Estudiantes: Familiarizarse con las herramientas de desarrollo del navegador (DevTools).

Estudiantes: Repasar los conceptos de ejecución secuencial y control de flujo en JavaScript.

Estudiantes: Investigar sobre los estándares ECMAScript y su relación con JavaScript.

Resumen

Introducción a JavaScript

El resumen de la sesión es el siguiente: EBIS introduce el tema de JavaScript, explicando que es un lenguaje de programación interpretado creado para dar interactividad a las páginas web. Destaca la importancia de JavaScript en el desarrollo web moderno, siendo el único lenguaje que entienden directamente los navegadores. Se discute la diferencia entre lenguajes compilados e interpretados, y cómo JavaScript ha evolucionado desde simples animaciones hasta permitir crear aplicaciones web completas. EBIS enfatiza la necesidad de entender los fundamentos de HTML, CSS y JavaScript vanilla antes de pasar a frameworks más avanzados. La clase cubre una introducción a JavaScript, su historia, el estándar ECMAScript, su expansión al backend y nociones básicas de sintaxis.

Historia Y Evolución De JavaScript.

El presentador explica la historia y evolución de JavaScript. Creado en 1995 por Brendan Eich en solo 10 días para Netscape Navigator, JavaScript surgió como respuesta a la necesidad de interactividad en las páginas web. Inicialmente llamado Mocha y luego LiveScript, se rebautizó como JavaScript por razones de marketing, aprovechando la popularidad de Java, aunque no tienen relación. A pesar de sus problemas iniciales, JavaScript se expandió rápidamente y se convirtió en omnipresente en el desarrollo web, evolucionando hasta ser un lenguaje moderno y versátil que se utiliza en múltiples dispositivos y plataformas.

JavaScript: El Lenguaje De Desarrollo Web

El grupo discute la evolución de JavaScript y su importancia en el desarrollo web moderno. Se explica que JavaScript ha pasado de ser un lenguaje para pequeñas interacciones a sostener aplicaciones completas, permitiendo que las aplicaciones web funcionen de manera similar en diferentes plataformas. Se mencionan frameworks y bibliotecas populares como React, Angular y Vue, destacando la necesidad de aprendizaje continuo en el campo del desarrollo web debido a su constante evolución. Se señala que JavaScript se ha vuelto tan relevante debido a su sintaxis sencilla y permisividad, aunque ha evolucionado para ser más seguro y potente, convirtiéndose en el lenguaje base para la mayoría de las aplicaciones web modernas.

ECMAScript: El Estándar De JavaScript.

El grupo discute sobre ECMAScript, el estándar que define JavaScript. ECMA explica que ECMA es una institución de estandarización que estableció ECMAScript en 1997 para unificar el desarrollo de JavaScript en diferentes navegadores. Se aclara la diferencia entre ECMAScript (el estándar) y JavaScript (el lenguaje que lo implementa). ECMA también describe cómo evolucionan los estándares a través de propuestas de los programadores para cubrir nuevas necesidades, las cuales, si son aprobadas, se incorporan en nuevas versiones del estándar.

Evolución De ECMAScript Y Desafíos

El instructor explica la evolución de ECMAScript (JavaScript) y los desafíos en el desarrollo web. Describe cómo hubo un estancamiento en el desarrollo de ECMAScript 4 debido a desacuerdos entre Microsoft y otros desarrolladores, lo que llevó a una época oscura en el desarrollo web con incompatibilidades entre navegadores. Google intervino en 2008-2009, lo que resultó en una reestructuración del comité de ECMAScript y la reanudación del desarrollo con ECMAScript 5. Desde entonces, se han lanzado versiones anuales, llegando hasta ECMAScript 15 en julio de 2024. El instructor introduce la herramienta "Can I Use" para ayudar a los desarrolladores a determinar la compatibilidad de las características en diferentes navegadores, destacando la fragmentación actual en el desarrollo web.

Node.js Democratizes Backend Development.

En la reunión, EBIS discutió la importancia de Node.js, un motor de ejecución de JavaScript que permite a los desarrolladores crear aplicaciones en JavaScript sin necesidad de un navegador. Este desarrollo revolucionó el desarrollo web, democratizando el backend y haciendo que fuera más accesible y ágil. Node.js permite a los desarrolladores utilizar JavaScript para el frontend y el backend, lo que lleva a la unificación del stack tecnológico. Esto ha resultado en costos reducidos, desarrollo más rápido y la capacidad de crear servicios y aplicaciones de manera constante. Sin embargo, también ha llevado a una disminución en la calidad del código debido a la falta de programadores de backend con formación sólida.

Desarrollo Web Y SEO

La discusión se centra en las herramientas de desarrollo web y el funcionamiento de los motores de búsqueda. Se explica que la consola del navegador es una interfaz para comunicarse con el motor JavaScript. Se analizan las diferentes pestañas de las herramientas de desarrollo, como Elements, Network y Performance. Se habla sobre la importancia del SEO y cómo Google clasifica las páginas web mediante el PageRank, un algoritmo secreto que los desarrolladores intentan descifrar. También se menciona la importancia de los Core Web Vitals para el posicionamiento en buscadores y cómo las herramientas de rendimiento ayudan a optimizar las páginas web.

JavaScript Sintaxis Básica

EBIS explica la sintaxis básica de JavaScript. Destaca que la terminación de sentencias con punto y coma ya no es necesaria, aunque históricamente se usaba. Menciona que los decimales en JavaScript usan punto en lugar de coma debido a su origen estadounidense. Advierte que los espacios generalmente son ignorados por el intérprete, pero su uso descuidado puede generar errores, especialmente al escribir números. EBIS demuestra estos conceptos con ejemplos prácticos en la consola del navegador.

JavaScript Basics and Syntax

El instructor explica conceptos básicos de JavaScript, enfocándose en la sensibilidad a mayúsculas y minúsculas, y el uso de diferentes tipos de paréntesis. Se destaca que JavaScript distingue entre mayúsculas y minúsculas en variables y funciones, y que las variables deben comenzar con minúscula. También se explica el uso de llaves para definir bloques, corchetes para arrays, y paréntesis para aislar y dar prioridad a operaciones. Finalmente, se menciona que JavaScript tiene una ejecución secuencial, leyendo el código de arriba a abajo.

JavaScript Execution and Control Flow

EBIS explica el concepto de ejecución secuencial en JavaScript y cómo se pueden modificar el flujo de ejecución mediante estructuras condicionales y bucles. Demuestra cómo las variables se actualizan secuencialmente y cómo los condicionales (if, else if, else) permiten ejecutar bloques de código según se cumplan ciertas condiciones. También introduce el concepto de bucles, explicando que pueden repetir un bloque de código mientras se cumpla una condición, lo que puede llevar a situaciones de bucle infinito si no se manejan correctamente.

Desbordamiento De Pila en Programación.

EBIS explica el concepto de desbordamiento de pila (stack overflow) y cómo se puede producir fácilmente con un bucle infinito en programación. Describe cómo los bucles infinitos pueden llenar rápidamente la memoria, especialmente en lenguajes como JavaScript que no tienen recolección de basura automática. También discute los diversos niveles de protección en los sistemas informáticos modernos para prevenir daños por desbordamientos de memoria, desde el navegador hasta el sistema operativo, la BIOS y los componentes físicos. Finalmente, EBIS introduce materiales de clase, incluyendo un PDF con resúmenes de las lecciones y ejercicios para practicar.

Práctica Y Documentación en Programación

En la reunión, EBIS enfatizó la importancia de la práctica y la documentación en el aprendizaje de la programación. Explicó que la documentación es fundamental para resolver problemas y entender nuevas tecnologías. EBIS también advirtió contra la dependencia de herramientas como ChatGPT y Copilot, sugiriendo que los estudiantes deberían aprender a buscar y comprender información por sí mismos. Los estudiantes fueron asignados ejercicios para practicar, con la advertencia de que no se corregirían, pero se revisarían brevemente en la próxima clase. EBIS terminó la reunión liberando a los estudiantes para el resto del día.