

Informe tarea 1 Modulo 3

Generalidades del Lenguaje JavaScript

1 Historia de JavaScript

JavaScript es un lenguaje interpretado orientado a objetos creado por Brendan Eich de Netscape en el año 1995.

En la actualidad implementas varios paradigmas como imperativo, orientado a objetos, funcional, etc.

Se utiliza principalmente del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas.

2 Uso de JavaScript en Navegadores Web

JavaScript es un lenguaje de programación esencial en los navegadores web porque permite que las páginas sean interactivas y dinámicas. Se ejecuta en el lado del cliente, lo que significa que el código JavaScript se descarga y se ejecuta directamente en el navegador del usuario, sin necesidad de comunicarse constantemente con el servidor. Además, JavaScript puede manipular el DOM para cambiar el contenido y la estructura de la página sin recargarla.

3 Entornos Virtuales de JavaScript

Debido a la flexibilidad del lenguaje JavaScript este puede ser ejecutado en distintos entornos desde navegadores web, aplicaciones de escritorio a dispositivos móviles.

Algunas de las herramientas y entornos en donde se puede ejecutar JavaScript son los siguientes:

- Navegador web: los navegadores web pueden interpretar sin herramientas adicionales vanilla JavaScript.
- Librerías/frameworks para interfaces web: existen varias librerías/frameworks que nos permiten crear interfaces web dinámicas en el navegador, algunas de ellas son las siguientes:
 - React: librería utilizada para crear interfaces de usuario a través de componentes.
 - Angular: framework que permite crear interfaces de usuario a través de módulos, controller, providers, etc.
- Node.js: entorno de ejecución de JavaScript basado en el motor V8 de Google capaz de ejecutar JavaScript por el lado del servidor.
- Electron: framework capaz de crear aplicaciones de escritorio nativas.
- React Native: framework utilizado para crear aplicaciones móviles nativas.

4 Diferencias entre JavaScript y otros lenguajes

La diferencia principal de JavaScript con otros lenguajes es que es un lenguaje interpretado, esto quiere decir que se utiliza un programa que es capaz de leer el código fuente de nuestros archivos y realizar la traducción de instrucciones del lenguaje a lenguaje de máquina.

De entre los lenguajes interpretados es uno de los mas populares debido a su flexibilidad y rendimiento, además los paradigmas que este implementa.

5 Fortalezas y debilidades de JavaScript

Una de las fortalezas de JavaScript es su curva de aprendizaje comparado con otros leguajes es muy fácil de aprender sin tanta fricción. Otra ventaja es la flexibilidad en los entornos de ejecución que posee ya que podemos desarrollar tanto interfaces web, aplicaciones de escritorio, aplicaciones móviles y APIs.

Dentro de las debilidades que podemos encontrar es el tipeo débil que este posee, ya que al no ser un lenguaje de tipado duro no posee la declaración de tipos de datos. Operadores “complejos” para la comparación de datos, JavaScript a través de su historia creo operadores de comparación que además de comparar los valores comparan los tipos (===, !==, etc.) cosa que en otros lenguajes no existen.

6 JavaScript como lenguaje asíncrono

JavaScript permite la programación asíncrona realizando tareas “largas” y seguir respondiendo a otros eventos mientras esa tarea se ejecuta, en lugar de esperar a que la tarea haya terminado.

Esto lo implementa a través de las palabras reservadas **async** y **await**.

Evolución del Lenguaje JavaScript y el Estándar ECMAScript

1 Lenguaje Interpretado vs Compilado

Los lenguajes interpretados son aquellos que a través de un software se interpretan las instrucciones del código fuente y se traducen a un lenguaje que entiende el computador, por el contrario, los lenguajes compilados son aquellos que a través de un compilador crean archivos que son ejecutados directo por la maquina (lenguaje de maquina).

Las ventajas de los lenguajes compilados es que son capaces de detectar errores en tiempo de compilación no así los lenguajes interpretados que detectan errores en tiempo de ejecución.

JavaScript es un lenguaje interpretado orientado a objetos, esto quiere decir que necesita de un software (interprete) para ser ejecutado.

2 Evolución del Estándar ECMAScript

Las características de las versiones de ECMAScript desde ES3 a ES9 son las siguientes:

- **ES3 (1999):** Se añadieron cosas básicas como reglas para manejar errores y nuevas formas de manejar texto.
- **ES5 (2009):** Se mejoraron las reglas para que los programas fueran más seguros y se hicieron algunas tareas más fáciles.
- **ES6 (2015):** Se añadieron muchas características nuevas, como formas más fáciles de declarar variables y funciones para trabajar con listas de datos.
- **ES7 (2016):** Se añadieron algunas pequeñas mejoras, como una nueva forma de hacer cálculos y nuevas formas de buscar elementos en listas.
- **ES8 (2017):** Se añadieron nuevas maneras de trabajar con objetos y listas, y se mejoraron las funciones asíncronas (las que esperan a que algo termine para seguir).
- **ES9 (2018):** Se mejoraron más las funciones asíncronas y se añadieron formas más fáciles de manejar errores en promesas (una forma de manejar tareas asíncronas).

En resumen, cada versión ha añadido y mejorado herramientas para hacer que escribir y ejecutar código en JavaScript sea más fácil y eficiente.

3 JavaScript vs. ECMAScript

La diferencia de JavaScript con ECMAScript es que JavaScript es el lenguaje de programación que implementa las reglas que define ECMAScript mientras que ECMAScript es un conjunto de reglas que define como debe funcionar un lenguaje de programación.

4 TypeScript y sus Características

Las principales características de TypeScript son los siguientes:

- Tipos definidos: tiene la capacidad de asignar tipos de datos a variables.
- Incluye todas las funciones de JavaScript y añade nuevas herramientas y características.
- Mejores herramientas de desarrollo: Ofrece autocompletado, navegación de código y otras ayudas en entornos de desarrollo.
- Integración con JavaScript: Se puede usar junto con código JavaScript existente.

5 Ventajas y Desventajas de TypeScript

La ventaja de usar TypeScript en el proyecto del hospital es tener la capacidad de asignar tipos de datos a las variables y tener la opción de usar generics.

Una de las desventajas de usar TypeScript es que es un proyecto pequeño y el setup para la transpiración de TypeScript a JavaScript requiere demasiado trabajo.

Análisis de la Pertinencia de Integrar JavaScript Avanzado o TypeScript en el Proyecto

Ventajas de utilizar JavaScript avanzado o TypeScript

JavaScript nos permite renderizar o realizar operaciones a través del lenguaje, manipular el DOM, crear contenido dinámico, consumir APIs REST, etc.

Desventajas o posibles dificultades

Una de las desventajas es la puesta en marcha de las configuraciones para agregar JavaScript o TypeScript en el proyecto ya que se necesitan herramientas como webpack, babel, etc.

Conclusión

Es recomendable usar JavaScript si necesitamos manipular el DOM de manera dinámica para agregar, quitar, actualizar, etc. Elementos del sitio web.