

Taller Análisis de la Incorporación de JavaScript en el Proyecto del Hospital

Informe de Investigación: Generalidades del Lenguaje JavaScript

Una de las características definitorias de JavaScript es el hecho de que fue creado en solo diez días. En 1995, Brendan Eich, su creador, cambió la forma en que se interactúa con los sitios web, pues en ese momento las páginas web eran estáticas, entonces el objetivo era permitir que los usuarios pudieran interactuar de formas dinámicas, sin tener que recargar toda la página. Desde entonces JavaScript ha evolucionado hasta convertirse en el pilar fundamental del desarrollo web actual.

La lógica y ventaja de JavaScript es que se ejecuta directamente en el navegador, utilizando motores (sea V8 o SpiderMonkey). Esto permite que cualquier dispositivo con un navegador popular puede ejecutar JavaScript. Más recientemente su capacidad se ha expandido al ser capaz de funcionar también en servidores mediante Node.js. Desde el navegador es capaz de manejar eventos, el DOM y validaciones, así JavaScript se convirtió en los esteroides de HTML.

Comparado con otros lenguajes, JavaScript destaca por su simplicidad inicial, pero también por su flexibilidad. Por ejemplo, aunque Python puede ser práctico para tareas de datos, JavaScript ofrece una combinación de funcionalidad y facilidad de uso para todo lo relacionado con la web. Aunque hay que reconocer que el tipado dinámico puede ser un arma de doble filo. Es decir, permite bastante libertad, pero a veces con demasiada libertad tiende a ser difícil rastrear errores que serían evidentes en otros lenguajes.

Otro aspecto importante de JavaScript son sus funciones asíncronas, pues es una característica fundamental en las interacciones en web. Gracias a su modelo basado en el Event Loop, puede manejar múltiples tareas al mismo tiempo sin bloquear la ejecución, es decir, es posible cargar contenido nuevo sin actualizar la página completa, e incluso ir cargándola de a poco, eso se logra usando callbacks, promises y funciones async/await.

Evolución del Lenguaje JavaScript y el Estándar ECMAScript

Personalmente valoro que JavaScript sea un lenguaje interpretado, mientras que otros, como C++ o Java, son compilados. Esto implica una de las cualidades más importantes de JavaScript: no necesita una etapa de compilación, se ejecuta directamente, lo cual lo hace ágil pero también propenso a errores sin la atención necesaria.

ECMAScript, el estándar en el que se basa JavaScript. Se podría decir que ECMAScript es el plano maestro, y JavaScript es la construcción que sigue ese plano. JavaScript ha implementado nuevas herramientas con cada nueva versión del estándar. Por ejemplo, ECMAScript 2015 trajo cambios transformadores como las funciones flecha, las promesas y las clases (que fueron usados extensivamente en los próximos ejercicios de este módulo).

En cuanto a TypeScript, aunque pueda parecer una limitación agregar tipado estático a un lenguaje que originalmente no exige declarar tipos, esta característica es valiosa al ayudar a detectar y prevenir errores

antes de que ocurran.

Análisis de la Pertinencia de Integrar JavaScript Avanzado o TypeScript

Pensando en el proyecto de la página web hospital, creo que la decisión de usar JavaScript avanzado o TypeScript seguirá dependiendo de varios factores, sin ser ninguna de las opciones categóricamente mejor que la otra. Por un lado, usar JavaScript avanzado permite bosquejar y desarrollar rápidamente. Pero por otro lado, TypeScript podría ser una mejor opción a largo plazo, especialmente si el equipo necesita trabajar en funcionalidades complejas y garantizar la calidad del código.

a. Ventajas:

- Aumenta la robustez del código.
- Mejora la legibilidad y facilita la escalabilidad.

b. Inconvenientes:

- Implica asimilar los conceptos de tipado, lo que puede alargar la fase inicial de desarrollo.
- El tipado estricto puede dificultar el desarrollo si no se cuenta con un plan bien definido desde el principio.

Como elección personal, prefiero comenzar con JavaScript y luego evaluar la transición a TypeScript en la medida de que el código empiece a complejizarse. Esto dará tiempo para definir exactamente que necesitará la página web. En circunstancias profesionales externas al curso, si en un equipo de desarrollo ya hay experiencia con TypeScript, no tendría dudas en implementarlo desde el principio. En conclusión, depende del contexto del equipo y del alcance del proyecto.