

ELP : Comparaison entre ELM et Javascript

Le programme TcTurtle est une application web interactive permettant de visualiser des dessins générés à partir de commandes textuelles décrivant le déplacement d'un crayon virtuel. L'objectif de ce projet est d'implémenter un interpréteur pour ce langage en ELM, afin de transformer ces commandes en tracés graphiques dynamiques. Ce choix technologique permet d'explorer les différences entre ELM et JavaScript dans le développement d'applications web interactives.

Tout d'abord, ELM est un langage de programmation fonctionnel et, de ce fait, les fonctions sont pures. Cela implique qu'il n'y a pas d'effet de bord lors de leur exécution et que, pour une même entrée, elles produisent toujours le même résultat. En JavaScript, la non-pureté des fonctions peut engendrer des bugs et des problèmes d'exécution en raison des effets de bord imprévus. Cette pureté fonctionnelle permet une meilleure prévisibilité et simplifie le raisonnement autour du code, ce qui est un atout majeur dans un projet comme TcTurtle.

ELM est un langage déterministe, ce qui signifie que l'exécution d'un même programme avec les mêmes entrées produira toujours le même résultat. Cela garantit une prévisibilité totale du comportement du programme, ce qui est particulièrement utile pour TcTurtle. En JavaScript, les effets de bord et les variations d'exécution entre navigateurs peuvent parfois engendrer des comportements inattendus, rendant le débogage plus complexe. De plus, ce déterminisme facilite grandement l'identification et la correction des erreurs, car les mêmes entrées mènent toujours aux mêmes sorties, ce qui est rarement garanti en JavaScript.

Par ailleurs, en ELM, le typage statique permet d'éviter de nombreuses erreurs avant même d'exécuter le programme, car celles-ci sont détectées à la compilation. Cela offre une sécurité accrue par rapport à JavaScript, où le typage dynamique peut entraîner des erreurs qui ne sont détectées qu'à l'exécution. En ELM, chaque variable et chaque fonction a un type bien défini, ce qui permet au compilateur de signaler les incohérences et d'éviter des erreurs courantes comme les accès à des valeurs undefined ou les erreurs de conversion de type. Ce mécanisme contribue à améliorer la fiabilité du code et à réduire le temps passé à corriger des bugs imprévus.

Enfin, ELM a été conçu spécifiquement pour le développement d'applications web monopages, comme celle au cœur de ce projet. Son architecture, inspirée du modèle MVC, simplifie la gestion de l'état et rend le code plus structuré. De plus, ELM est interopérable avec JavaScript, ce qui permet d'utiliser ce dernier en complément si nécessaire, offrant ainsi une plus grande flexibilité d'implémentation. Contrairement à JavaScript, où la gestion de l'état peut devenir complexe en raison des modifications asynchrones et des effets de bord, ELM garantit un flux de données unidirectionnel et prévisible, ce qui facilite le développement et la maintenance.

En conclusion, ELM offre une robustesse et une sécurité accrues grâce à son typage strict, permettant un code prévisible et structuré. La maintenabilité d'ELM est également supérieure à celle de JavaScript en raison de ces caractéristiques. Opter pour ELM permet d'éviter de nombreuses erreurs et de réduire le temps consacré à la gestion des bugs et autres problèmes. Dans le cadre du projet TcTurtle, ces avantages ont permis d'implémenter un interpréteur fiable et performant, garantissant un rendu graphique précis et une expérience utilisateur fluide.