Angie Bui
2-2-31

JavaScript Language Evaluation

Efficiency:

Compilation - Unlike Java, JavaScript is not compiled. Instead, it is interpreted and executed directly by the browser or JavaScript engine.

Runtime - JavaScript is executed on a JavaScript engine in either a web browser or a standalone environment like Node.js. It offers a high-level, dynamic programming style that can simplify coding but may not be as fast as low-level languages.

Coding - JavaScript has a readable and high-level syntax, making it easier to code than low-level programming languages. It supports various programming paradigms, including object-oriented, functional, and event-driven programming, and has a dynamic type system.

Regularity:

JavaScript has an inconsistent orthogonality, meaning that its syntax and features are not always logically consistent. This can be traced back to the language's evolution over time and the addition of new features, as well as its dynamic typing system and automatic type coercion. Additionally, some aspects of JavaScript's syntax can also contribute to inconsistencies, such as in equality comparisons.

Security:

JavaScript does not have pointers, unlike languages like C or C++. It uses variables to store values and relies on the JavaScript engine's garbage collector to manage memory, which eliminates the need for manual memory management through pointers. This makes programming in JavaScript more convenient, but it also limits the control developers have over memory allocation.

Extensibility:

JavaScript is highly customizable and can be extended through various methods, such as using libraries, defining custom data types, and using macro preprocessors (such as TypeScript). This flexibility allows developers to add new features and functionality to the language, making it possible to build complex and powerful applications.