1.3 Difference between JavaScript and TypeScript

One of the fundamental differences between JavaScript and TypeScript is type system. Dynamically typed language like JavaScript enables variables to determine any data type during execution. While it offers higher flexibility, it also can cause errors during execution (Nestsiarovich, 2023). On the other hand, TypeScript builds upon JavaScript by incorporating a static typing, enabling developers to define types for variables, functions, and return values. It can ensure code robustness and check errors during development rather than execution. However, it might be challenging for beginner developer to use TypeScript due to its additional concepts and complexity (Target, 2023).

The compilation process further underscores the differences between the two languages. JavaScript is interpreted directly by browsers or other JavaScript engines, while TypeScript needs to be converted into JavaScript using TypeScript compiler. This extra step ensures adaptability with all browsers, but potentially slowing down iteration cycles (Holmes, 2023).

Beyond that, tooling and IDE support is one of the main distinctions between JavaScript and TypeScript. TypeScript's static typing enables advanced support, such as code completion, automatic imports, and type checking within IDEs (Deshpande, 2024). This increases productivity of developers and quality of code. Furthermore, by serving as self-documentation, TypeScript's static typing also improves code readability and maintainability. This is particularly beneficial when developers are engaged in large projects where they need to review a large codebase.

Aside from that, one of the disparities between these languages lies in their compatibility. Since TypeScript is a superset of JavaScript, developers can gradually integrate TypeScript into existing JavaScript without having to rebuild the code again (Interview Bit, 2024). The compatibility allows for less disruptive transition to ongoing project and enables the teams to gradually familiarise themselves with Typescript's features over time.

Besides that, another primary distinction between JavaScript and TypeScript is community and ecosystem. As a more established language with a longer history and broader adoption, JavaScript has a larger and more mature ecosystem beyond TypeScript. While TypeScript continues to grow, it still does not hold as many libraries, frameworks, and resources available as JavaScript's ecosystem has accumulated (Hartman, 2024).