RAZLIKE I SLIČNOSTI IZMEĐU PROGRAMSKIH JEZIKA JAVASCRIPT I TYPESCRIPT

Maja Gaborov[1], Mihajlo Mulić[2], Ninoslava Tihi[3], Stefan Popović[4]

Rezime: Java Script i TypeScript su moderni programski jezici koji se koriste u veb razvoju. JavaScipt je se smatra fleksibilnim jezikom i koristi se za izradu dinamičkih stranica, a TypeScript je nastao da reši nedostatke JavaScripta kao što je nedostatak statičke tipizacije. JavaScript ima široku podršku u pretraživačima, dok TypeScript uvodi statičko kucanje i dodatne objektno orijentisane funkcije kako bi mogle da se prave aplikacije velikih razmera. Svrha ovog rada je da se predstave ove dve tehnologije, odnosno da se uporede. Prikuplljene i analizirane su informacije kako bi se izdvojile ključne sličnosti i razlike između ova dva programska jezika. Razumevajući ove programske jezike, programeri mogu da donesu odluke kada biraju jezike za svoje projekte. Rad prikazuje da je TypeScript proširena i unapređena verzija JavaScripta i da je posebno korisno za velike projekte gde je važno sprečiti greške pri izvršavanju koda, tako da je cilj ovog rada da razjasni zašto je korisno da se koriste pomenuti programski jezici u zavisnosti od potrebe i složenosti projekta.

Ključne reči: JavaScript, TypeScript, programski jezici, razlike, sličnosti, veb programiranje

DIFFERENCES AND SIMILARITIES BETWEEN THE PROGRAMMING LANGUAGES JAVASCRIPT AND TYPESCRIPT

Abstract: JavaScript and TypeScript are modern programming languages used in web development. JavaScript is considered a flexible language and is used to create dynamic pages. TypeScript was created to address the shortcomings of JavaScript, such as the lack of static typing. JavaScript has broad browser support, while TypeScript introduces static typing and additional object-oriented features to enable large-scale applications to be built. The purpose of this work is to present and compare these two technologies. Information was collected and analyzed to highlight the key similarities and differences between these two programming languages. By understanding these programming languages, developers can make decisions when choosing languages for their projects. The paper shows that TypeScript is an extended and improved version of JavaScript and is especially useful for large projects where it is important to prevent errors during code execution, so the aim of this paper is to clarify why it is useful to use the mentioned programming languages depending on the need and complexity of the project.

Key words: JavaScript, TypeScript, programming languages, differences, similarities, web programming

1. INTRODUCTION

JavaScript is a popular programming language. Its flexibility and wide support in all modern browsers have made it an essential tool for web developers [1]. However, as web applications became more complex, there was a need for a language that offered more advanced functionality, such as static typing and more complex data types [2-3]. In response to these needs, TypeScript was developed [4-5].

TypeScript was introduced with the aim of improving the development of large and complex applications by providing additional functionalities such as static typing, interfaces, and abstract classes [6]. TypeScript maintains compatibility with existing JavaScript code, allowing for gradual migration and use of existing JavaScript libraries and tools [7-8].

The goal of this paper is to explore the key differences and similarities between JavaScript and TypeScript, using theoretical sources from the Internet to provide a comprehensive analysis. Understanding these differences and similarities is crucial for developers who want to choose the right language for their projects and maximize efficiency and productivity [9]. The author of this paper wants to better see the similarities and differences because he wants to decide on the appropriate programming language to use for his project.

2. METODOLOGY

In this paper, the similarities and differences between JavaScript and TypeScript are shown, where the data is collected from various online databases. Trusted and relevant sources such as the official language websites (TypeScript and JavaScript), the Mozilla Developer Network (MDN), and the works of experts such as Brendan Eich, Anders Heilsberg, and others were preferred.

The data were collected by searching the Internet and online databases. Criteria for the selection of sources included the reliability of the author and the relevance of the information to the research.

Data analysis was carried out by identifying key similarities and differences when reading the mentioned sources. The collected data was synthesized to identify and visualize the differences and similarities between JavaScript and TypeScript. The image presents the similarities and differences, enabling a simple understanding of the key aspects of both languages.

3. DIFFERENCES BETWEEN JAVA SCRIPT AND TYPE SCRIPT

Differences between JavaScript and TypeScript are:

1. Typification

JavaScript: A dynamically typed language. Types are assigned at runtime [7].

TypeScript: A statically typed language. Types are assigned during development, which allows errors to be detected before the code is executed [4].

2. Complex Types

JavaScript: No support for complex types such as unions, intersections, or tuples [7].

TypeScript: Enables the use of complex types that allow better control over data structures [6].

3. Compiling

JavaScript: Executes directly in the browser or Node.js environment [7].

TypeScript: Must be transpiled to JavaScript by the TypeScript compiler (tsc) before it can be executed [4].

4. Interfaces and Abstractions

JavaScript: No support for interfaces or abstract classes [7].

TypeScript: Enables the creation of interfaces and abstract classes, which contributes to better code organization and reusability [9].

5. Support for ES6/ES7

JavaScript: Newer versions (ES6/ES7) bring new functionalities, but older versions do not support all these functionalities [1].

TypeScript: Supports all ES6/ES7 functionality and can be configured to target different versions of JavaScript [4].

6. Tools and Ecosystem

JavaScript: Widely supported by many tools, libraries, and frameworks [7].

TypeScript: Compatible with all JavaScript tools and libraries, with additional TypeScript-specific tools such as linters and IDE plugins [6].

JavaScript is dynamically typed, while TypeScript is statically typed, allowing for error detection during development. TypeScript introduces support for complex types, while JavaScript does not. Also, TypeScript must be transpiled to JavaScript before execution, while JavaScript is executed directly. TypeScript provides the ability to use interfaces and abstract classes for better code organization, while JavaScript does not. Another important difference is that TypeScript supports all

the functionality of the ES6/ES7 standard and can be configured for different versions, while JavaScript depends on the version being used. Although both languages have a broad ecosystem, TypeScript offers additional tools that make it easier to develop and maintain larger applications.

4. SIMMILIARITIES BETWEEN JAVA SCRIPT AND TYPE SCRIPT

Simmiliarities between JavaScript and TypeScript are:

1. Syntax

Both languages: TypeScript is a superset of JavaScript, which means that any valid JavaScript code is also valid in TypeScript. The syntax for basic code structures (such as loops, conditional statements, functions) is identical [4].

2. Ecosystem

Both languages: Use the same ecosystem of libraries and tools. TypeScript code is transpiled to JavaScript, allowing it to use all existing JavaScript libraries and frameworks such as React, Angular, Vue, Node.js, etc. [6]

3. Execution

Both languages: The final output of TypeScript code is JavaScript, which is executed in the same runtime environments, whether it is a browser or Node.js [7].

4. ES6/ES7 functionalities

Both languages: Support functionality introduced in ES6/ES7 such as let/const declarations, arrow functions, classes, modules, template literals, etc. [1]

5. Asynchronous Programming

Both languages: Support asynchronous programming using promises and async/await syntax [9].

6. Support for OOP

Both languages: Support object-oriented programming (OOP) approaches, allowing the creation and use of classes, inheritance, encapsulation, etc. [4].

JavaScript and TypeScript are similar languages because TypeScript is a superset of JavaScript. They use identical syntax for basic structures like loops, conditions, and functions. They have a common ecosystem of tools and libraries, as TypeScript code is transpiled to JavaScript and can use the same frameworks as React, Angular, and Vue. They run in the same environments, like the browser and Node.js. Both technologies support functionality from the ES6/ES7 standard, asynchronous programming (using promises and async/await), as well as object-oriented programming with classes, inheritance, and encapsulation.

5. DISCUSSION

Figure 1 shows the sets representing the characteristics JavaScript and TypeScript, as well as the intersection representing the similarities. The left set in the image represents the JavaScript features, and the right set represents the TypeScript features.

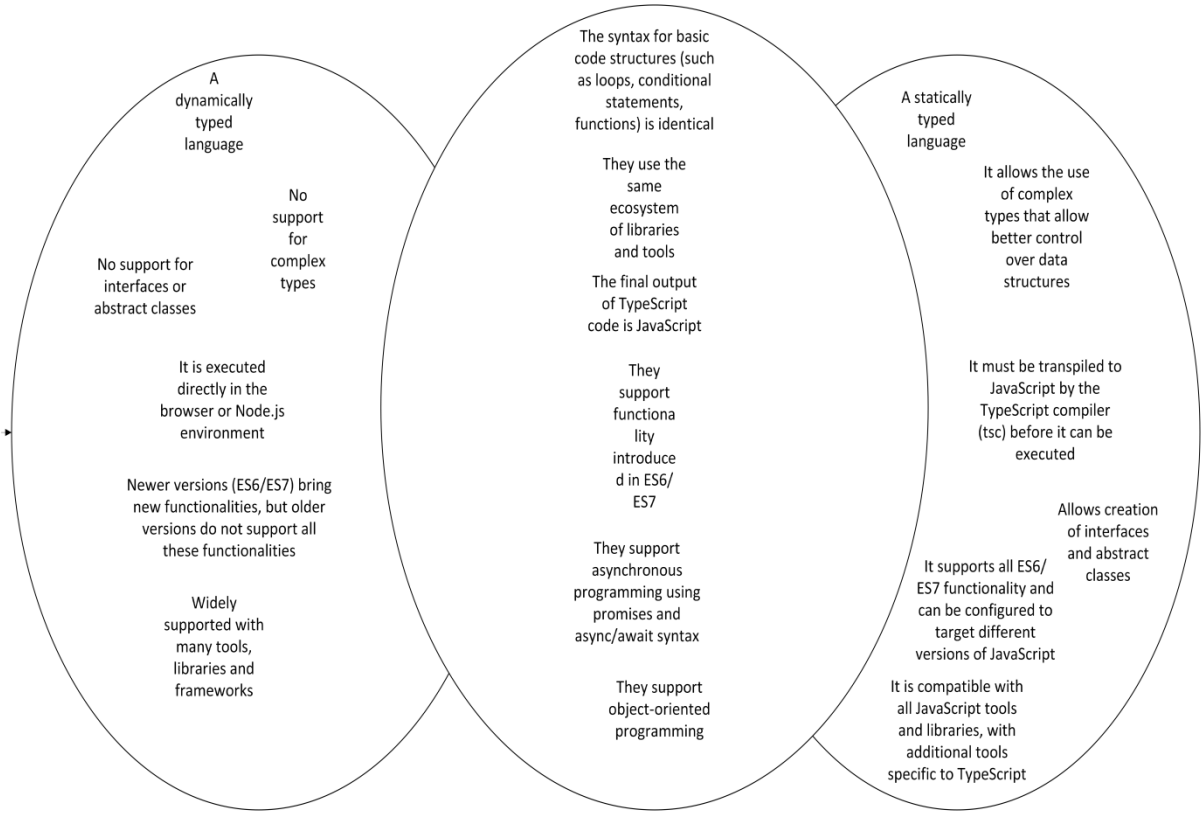

Figure 1Differences and simmiliraties of JavaScript and TypeScript

The key difference between JavaScript and TypeScript lies in typing. JavaScript is a dynamically typed language, meaning that the types of variables are not known until the code is executed, while TypeScript is statically typed, allowing errors to be detected during development before the code is executed. This allows for greater security and better control over the code in TypeScript. Also, TypeScript supports more complex types and abstract structures, such as interfaces and abstract classes, that JavaScript does not. Although TypeScript requires transpilation to JavaScript before it can be executed, both languages use the same ecosystem of tools, libraries, and frameworks

The similarity between them is in their syntax and ability to use the same functionality and support asynchronous programming and object-oriented programming. TypeScript is actually an upgrade to JavaScript that allows for better code structures and easier maintenance, while JavaScript offers simplicity and flexibility for rapid development.

JavaScript dominates as a standard for web programming, and TypeScript is gaining more and more importance in larger projects where it is important to reduce the number of errors and improve code readability [2]. Static typing in the TypeScript programming language allows for better error detection early in development, which can save time and resources. The choice between languages depends on the size of the team, the complexity of the project, and the level of knowledge of the programmer. TypeScript requirements bring numerous benefits in the maintenance and development of large applications.

6. CONCLUSION

TypeScript is a better choice for larger, more complex projects that require greater control over the code, greater security thanks to technical typing, and support for advanced code structures such as interfaces and abstract classes. It allows for faster error detection during development, which improves the maintainability and scalability of the application.

JavaScript is a simpler and more flexible language, ideal for rapid development and projects where static typing is not essential. Its versatility and broad support make it extremely suitable for many web applications, especially when rapid prototyping or working with smaller teams is required.

Essentially, the choice between JavaScript and TypeScript depends on the needs of the project, the size of the team, and the complexity of the code being developed.

7. REFERENCES

[1] Eich, B. (2008). JavaScript: The good parts. O'Reilly Media.

[2] Bogner, J., & Merkel, M. (2022). To type or not to type? a systematic comparison of the software quality of javascript and typescript applications on github. In Proceedings of the 19th International Conference on Mining Software Repositories (pp. 658-669).

[3] Jansen, R. H., Vane, V., & De Wolff, I. G. (2016). TypeScript: Modern JavaScript Development. Packt Publishing Ltd

[4] Hejlsberg, A., & Rosenwasser, D. (2024). TypeScript: JavaScript that scales. Retrieved from https://www.typescriptlang.org/

[5] Islam, M. S. (2023). JavaScript alternative (TypeScript) and its effectiveness in web development.

[6] Fenton, W., & Schenker, G. (2021). Mastering TypeScript. Packt Publishing.

[7] Mozilla Developer Network (MDN). (n.d.). JavaScript reference. Retrieved from https://developer.mozilla.org/en-US/docs/Web/JavaScript

[8] Haapalainen, A. (2024). A comparative analysis of converting JavaScript to TypeScript in advanced web applications course.

[9] Bierman, G., Abadi, M., & Torgersen, M. (2021). Understanding TypeScript. Journal of Computer Science and Technology.