TypeScript Crash Course

A hands-on guide to building safer and more reliable web applications

Daniel Cavalcante

www.bpbonline.com

First Edition 2024

ISBN: 978-93-55516-763

LIMITS OF LIABILITY AND DISCLAIMER OF WARRANTY

The information contained in this book is true to correct and the best of author's and publisher's knowledge. The author has made every effort to ensure the accuracy of these publications, but publisher cannot be held responsible for any loss or damage arising from any information in this book.

All trademarks referred to in the book are acknowledged as properties of their respective owners but BPB Publications cannot guarantee the accuracy of this information.

Dedicated to

My amazing son ***Pedro***

and

My supporting parents and family

About the Author

Daniel Cavalcante has been working as a developer for more than 15 years. He has worked on Frontend, Backend, and mobile development for companies in Brazil, United States and Europe. He graduated from journalism and always aspired to be a writer, but his interest in technology and the web led to a career as a self-taught developer. Working with advertising agencies in São Paulo he has created websites and applications for companies like Johnnie Walker, Mattel, and large Brazilian brands like Sadia. He has been a speaker at conferences like React CWB. For the past years, he has been working remotely for companies like Osmosis, an American startup that was acquired by Elsevier, world's largest scientific publisher. Currently, he provides consultancy (and talks about technology) through his company Lightcode.

About the Reviewer

Jahred Love has been a software engineer for nearly three decades. He has worked with some of the world's biggest and well-known companies, producing cutting-edge R&D. His most recent enterprise is Xirsys, which provides tools and software for real-time communication. Jahred is a leader in the field of WebRTC and currently heads the Voice Quality department for a well-known global telecommunications company.

Acknowledgement

I am truly thankful for the unwavering support and encouragement from my parents, George and Noeme, and my son Pedro, for inspiring me to be more creative and bolder. I am also thankful to my siblings Marina, Carolina and David.

I also extend my thanks to BPB Publications for their expert guidance and support in bringing this book to life. The process of textual and technical revision was a collaborative effort that enriched both the quality of this book and my technical knowledge.

I also want to highlight the invaluable lessons and feedback from colleagues and collaborators that I have met in the tech industry. A shoutout to the team at Osmosis.org, especially the founder Shiv. He remains a friend and mentor who taught me a lot about expressing gratitude.

Finally, my appreciation goes to all the readers who have shown an interest in my book. Your support makes this dream a reality and I hope it contributes to making your TypeScript journey more enjoyable!

Preface

The immense popularity of the Web has hoisted JavaScript to the top of the most popular programming languages. From small websites to large-scale enterprise applications, the dynamic nature of JavaScript has challenged developers due to the unpredictability of increasingly larger codebases. Typescript has been established as the de facto solution to this problem, a statically typed superset that introduces type safety and powerful development tools. It is a layer of abstraction ultimately converted into JavaScript itself.

This book will help you start developing enterprise-level applications using TypeScript, beginning with the basics, covering its syntax and key features such as types, interfaces, functions, and classes. Then, we explore more advanced topics including, how to work with modules, manage asynchronous code with promises and async/await, etc.

Whether you are new to TypeScript and looking to understand its application in enterprise development, or an experienced developer aiming to enhance your skills in building scalable and maintainable applications, this book offers valuable knowledge and insights. By the end of this book, you will have a solid foundation in TypeScript and the skills necessary to tackle the challenges of enterprise application development head-on.

Chapter 1: Introduction to TypeScript - This chapter serves as the starting point to learning TypeScript, a powerful extension of JavaScript that introduces types to your code. It begins with the basics, outlining TypeScript's significance, and the advantages it brings to large-scale application development.

Chapter 2: Installation and Setup – This chapter focuses on the initial steps required to integrate TypeScript into your development workflow, including the installation process across Windows, MacOS, and Linux. It covers setting up popular code editors to work seamlessly with TypeScript. We have laid a special emphasis on debugging techniques and the configuration of the TypeScript compiler for optimal development. This chapter ensures that you are well-prepared to write, debug, and compile TypeScript code efficiently.

Chapter 3: TypeScript's Fundamentals – This chapter discusses the core concepts of TypeScript, providing a comprehensive overview of its syntax, type system, and the foundational knowledge necessary for effective coding. From basic syntax and type annotations to more intricate features like enums and tuples, this chapter builds a solid base for understanding how TypeScript enhances JavaScript by adding static types.

Chapter 4: Structuring and Extending Types – This chapter explores the power of TypeScript's type system, focusing on creating and using interfaces, classes, and advanced types to build

well-organized and maintainable code. This chapter discusses how TypeScript's type system allows for defining complex data structures, implementing inheritance and encapsulation with classes, and utilizing utility types for more flexible code

Chapter 5: Working with Advanced TypeScript Features – This chapter unveils the more sophisticated aspects of TypeScript, including generics, decorators, and advanced type manipulation techniques such as mapped and conditional types. This chapter empowers readers to leverage these advanced features to write highly reusable and flexible code. Through practical examples and in-depth explanations, readers will learn how to use generics to create highly adaptable functions and classes, decorators to add metadata and behavior to code, and advanced types for precise type transformations.

Chapter 6: Migrating a JavaScript Web App to TypeScript – This chapter provides a step-by-step guide on transitioning an existing JavaScript application to TypeScript. Using a Todo List app as a practical example, this chapter walks through the process of converting JavaScript code to TypeScript, integrating third-party libraries, and addressing common challenges encountered during migration.

Chapter 7: Adding TypeScript to a React Application – This chapter takes the principles of the Todo List app and applies them to a React application, demonstrating how to enhance a React project with TypeScript's static typing. This chapter covers setting up a React project with TypeScript, refactoring components to use TypeScript features, and addressing specific type challenges inherent in React development.

Chapter 8: Using TypeScript with a Node.js Application – This chapter expands the application of TypeScript to server-side development with Node. This chapter outlines how to integrate TypeScript into a Node project, from setting up the development environment to converting an existing app to use TypeScript.

Chapter 9: Building TypeScript for Production – This chapter focuses on the practical aspects of preparing a TypeScript project for production deployment. It covers topics such as optimizing TypeScript code for performance, setting up continuous integration and continuous deployment (CI/CD) pipelines, and managing dependencies. This chapter also explores using TypeScript with modern tooling like bundlers and task runners, ensuring readers are equipped to deliver efficient, scalable, and maintainable TypeScript applications.

Chapter 10: Best Practices and Next Steps – This chapter concludes the book by summarizing best practices for using TypeScript effectively and exploring pathways for further learning and mastery. This chapter emphasizes strategies for incremental adoption of TypeScript, tips for maintaining clean and type-safe code, and resources for continuing education in TypeScript and its ecosystem. By the end of this book, you will leave with a roadmap for advancing your TypeScript skills and integrating them into future projects.

Code Bundle and Coloured Images

Please follow the link to download the
Code Bundle and the ***Coloured Images*** of the book:

https://rebrand.ly/9b2d9a

The code bundle for the book is also hosted on GitHub at
https://github.com/bpbpublications/TypeScript-Crash-Course.
In case there's an update to the code, it will be updated on the existing GitHub repository.

We have code bundles from our rich catalogue of books and videos available at **https://github.com/bpbpublications**. Check them out!

Errata

We take immense pride in our work at BPB Publications and follow best practices to ensure the accuracy of our content to provide with an indulging reading experience to our subscribers. Our readers are our mirrors, and we use their inputs to reflect and improve upon human errors, if any, that may have occurred during the publishing processes involved. To let us maintain the quality and help us reach out to any readers who might be having difficulties due to any unforeseen errors, please write to us at :

errata@bpbonline.com

Your support, suggestions and feedbacks are highly appreciated by the BPB Publications' Family.

Piracy

If you come across any illegal copies of our works in any form on the internet, we would be grateful if you would provide us with the location address or website name. Please contact us at **business@bpbonline.com** with a link to the material.

If you are interested in becoming an author

If there is a topic that you have expertise in, and you are interested in either writing or contributing to a book, please visit **www.bpbonline.com**. We have worked with thousands of developers and tech professionals, just like you, to help them share their insights with the global tech community. You can make a general application, apply for a specific hot topic that we are recruiting an author for, or submit your own idea.

Reviews

Please leave a review. Once you have read and used this book, why not leave a review on the site that you purchased it from? Potential readers can then see and use your unbiased opinion to make purchase decisions. We at BPB can understand what you think about our products, and our authors can see your feedback on their book. Thank you!

For more information about BPB, please visit **www.bpbonline.com**.

Join our book's Discord space

Join the book's Discord Workspace for Latest updates, Offers, Tech happenings around the world, New Release and Sessions with the Authors:

https://discord.bpbonline.com

Table of Contents

Chapter 1
Introduction to TypeScript

Introduction

This chapter introduces you to TypeScript, a popular programming language that extends JavaScript by adding types. You will learn about the importance of a type system, TypeScript's history, its key features and how it compares to JavaScript.

In the upcoming chapters of this crash course book, we will explore the various features of TypeScript and learn how to leverage them effectively to build robust and scalable applications. Starting from the basics and finishing with practical examples on how to integrate TypeScript with existing JavaScript and NodeJS projects.

You will learn about the **TypeScript compiler** (`tsc`), a command line tool used to check for type errors and convert TypeScript to equivalent JavaScript, which is the actual code that will be executed on the runtime of the browser or NodeJS.

Also, we are going to leverage Visual Studio Code, a popular development environment for TypeScript, that offers features like autocompletion and type checking that significantly speeds up the development process. By providing better tooling and editor support, TypeScript makes it easier for developers to navigate and refactor their code.

Structure

The chapter covers the following topics:

- Introduction to TypeScript
- What is a type system and why does it matter?
- TypeScript's key features
- Differences between TypeScript and JavaScript
- Benefits of a type system in large-scale projects

Objectives

By the end of this chapter, you will understand what TypeScript is, what type systems are, the differences between TypeScript and JavaScript, and finally the advantages TypeScript offers for building scalable web applications.

Introduction to TypeScript

TypeScript is a programming language that extends JavaScript by adding static types on top of it. Created by Microsoft developer *Anders Hejlsberg*, TypeScript was released in 2012 as an open-source project.

Brief overview of TypeScript

TypeScript provides the ability to write scalable and maintainable applications while leveraging the features and versatility of JavaScript. As a *superset* of JavaScript, TypeScript is compatible with any valid JavaScript code, allowing developers to gradually adopt it in their existing projects without the need for a complete rewrite. It enables developers to specify the type of variables, function parameters, and return types. This allows for catching errors at compile-time rather than runtime, resulting in improved code quality, maintainability, readability, and fewer bugs in production. Additionally, it can result in fewer lines of code, as values do not need to be explicitly checked for type.

TypeScript also introduces a range of advanced features such as interfaces and generics. These features enhance the development experience by providing a way to define contracts, organize code into reusable components, and enforce type safety across the application. Through this last decade, TypeScript has gained significant popularity in the web development community due to its ability to address many challenges faced by JavaScript developers. It is becoming the preferred choice for large-scale applications and has been widely adopted by notable companies like Microsoft, Google, Airbnb, and Slack. With a growing community and support from major industry players, TypeScript's future is bright for years to come!

Community-driven projects like **DefinitelyTyped** provide TypeScript definitions for popular JavaScript libraries, making it easier for developers to leverage existing code while enjoying the benefits of TypeScript's type system on the web browser, NodeJS or any JavaScript runtime.

The motivation behind creating TypeScript

TypeScript was created to address the limitations and challenges faced by JavaScript developers when working on large-scale applications. As web applications grows both in size and complexity, the flexibility of JavaScript's dynamic typing becomes a double-edged sword, leading to runtime errors, maintainability issues, and difficulty scaling codebases.

One of the primary motivations behind creating TypeScript was to introduce static typing to JavaScript, enabling developers to catch errors at compile time instead of runtime. This would result in more robust, maintainable, and scalable applications. Since TypeScript is a superset of JavaScript, it allows developers to gradually adopt its features without having to rewrite their entire codebase.

Another motivation was to improve tooling and editor support. With static typing, editors can provide better code navigation, autocompletion, and refactoring capabilities, ultimately improving developer productivity. The type information also enables better documentation and code understanding, making it easier for teams to collaborate and maintain their projects.

TypeScript also enables better code organization and scalability through the introduction of features like classes, modules, and interfaces. These language constructs allow developers to structure their code in a more modular and maintainable manner, facilitating code reuse and separation of concerns.

TypeScript's relationship with JavaScript

TypeScript is a superset of JavaScript, which means that any valid JavaScript code is also valid TypeScript code. TypeScript extends JavaScript by adding optional static typing and a host of other features that make it easier to develop, maintain, and scale large web applications.

Developers can start by writing JavaScript code and then gradually introduce type annotations to enhance the code with static typing. This allows projects to transition to TypeScript at their own pace, without the need for a complete rewrite to start using it.

One of the key benefits of TypeScript's relationship with JavaScript is its ability to leverage its vast ecosystem. TypeScript can use existing JavaScript libraries, frameworks, and tools seamlessly, enabling developers to utilize popular JavaScript libraries like React or Express in their TypeScript projects without any issues.