O'REILLY®

Deutsche Ausgabe

TypeScript

Ein praktischer Einstieg

Typsicheres JavaScript für skalierbare Webanwendungen

Josh Goldberg

Übersetzung von
Jens Olaf Koch

Leserstimmen zu *TypeScript – Ein praktischer Einstieg*

Wenn Sie schon einmal wütend die roten, geschlängelten Linien unter Ihrem Code angebrüllt haben, dann sollten Sie *TypeScript – Ein praktischer Einstieg* lesen. Goldberg ordnet alles meisterhaft in den Kontext ein, bleibt dabei immer praktisch orientiert und zeigt uns so, dass TypeScript niemals eine Einschränkung, sondern immer eine wertvolle Bereicherung ist.

– Stefan Baumgartner, Senior Product Architect, Dynatrace; Gründer, oida.dev

Josh stellt die wichtigsten Konzepte von TypeScript in den Mittelpunkt und erklärt sie anhand von klaren Beispielen – und mit einer Prise Humor. Eine Pflichtlektüre für den JavaScript-Programmierer, der TypeScript-Profi werden will.

– Andrew Branch, Software Engineer für TypeScript, Microsoft

TypeScript – Ein praktischer Einstieg ist eine hervorragende Ressource für Programmierer, die zumindest ein wenig Vorkenntnisse besitzen, aber bisher vielleicht vor typisierten Sprachen zurückgeschreckt sind. Es geht eine Ebene tiefer als das *TypeScript Handbook* und gibt Ihnen auf diese Weise die Sicherheit, TypeScript in Ihren eigenen Projekten einsetzen zu können.

– Boris Cherny, Software Engineer, Meta; Autor von »Programmieren in TypeScript« (O'Reilly)

Wir wissen nicht, was »Types Code« ist, aber wir sind sehr stolz auf Josh und sicher, dass es ein schönes Buch wird.

– Frances und Mark Goldberg

Josh ist eines dieser seltenen Exemplare von Menschen, die leidenschaftlich daran interessiert sind, sich die Grundlagen eines Themas zu eigen zu machen und Anfängern die wesentlichen Konzepte zu erklären. Ich glaube, dass sich dieses Buch schnell zu einer kanonischen Ressource für TypeScript-Neulinge und -Experten gleichermaßen entwickeln wird.

– Beyang Liu, CTO und Mitgründer, Sourcegraph

TypeScript – Ein praktischer Einstieg ist eine fantastische Einführung in und Referenz zu TypeScript. Joshs Schreibstil ist klar und informativ, und das hilft bei der Erklärung der oft verwirrenden Konzepte von TypeScript und dessen Syntax. Ein großartiger Einstieg für alle, die mit TypeScript beginnen!

– Mark Erikson, Senior Frontend Engineer, Replay; Maintainer, Redux

TypeScript – Ein praktischer Einstieg ist ein großartiges Buch, um Ihre TypeScript-Reise zu beginnen. Es gibt Ihnen die Werkzeuge an die Hand, um die Sprache, das Typsystem und die IDE-Integration zu verstehen, und zeigt Ihnen, wie Sie all dies nutzen können, um das Beste aus Ihrer TypeScript-Erfahrung herauszuholen.

– Tizian Cernicova Dragomir, Software Engineer, Bloomberg LP

Josh ist seit vielen Jahren ein wichtiger Teil der TypeScript-Community, und ich freue mich sehr, dass jetzt durch *TypeScript – Ein praktischer Einstieg* alle von seinem tiefen Verständnis und zugänglichen Lehrstil profitieren können.

– James Henry, Consultant Architect, Nrwl; 4x Microsoft MVP; Schöpfer, angular-eslint und typescript-eslint

Josh ist nicht nur ein sehr begabter Softwareentwickler, sondern auch ein hervorragender Mentor; seine Leidenschaft für Wissensvermittlung ist in diesem Buch deutlich zu spüren. *TypeScript – Ein praktischer Einstieg* ist meisterhaft strukturiert und enthält praktische, wirklichkeitsnahe Beispiele, die TypeScript-Neulinge und -Enthusiasten auf das nächste Level bringen werden. Ich kann guten Gewissens sagen, dass *TypeScript – Ein praktischer Einstieg* der ultimative Leitfaden für alle ist, die mehr über TypeScript lernen und Ihren Wissensstand vertiefen wollen.

– Remo Jansen, CEO, Wolk Software

In *TypeScript – Ein praktischer Einstieg* bricht Josh Goldberg die komplexesten Konzepte von TypeScript auf einfache Beschreibungen und gut verdauliche Beispiele herunter, sodass sein Buch auf Jahre hinaus als Lernhilfe und Nachschlagewerk dienen wird. Vom ersten Haiku bis zum letzten Witz ist *TypeScript – Ein praktischer Einstieg* eine wunderbare Einführung in die Sprache, die genau »mein Typ« ist.

– Nick Nisi, Staff Engineer, C2FO

Früher sagte man: »Setze immer auf JavaScript.« Jetzt heißt es: »Setze immer auf TypeScript«, und dieses Buch wird sich branchenweit zu einer gefragten Ressource entwickeln. Garantiert.

– Joe Previte, Open Source TypeScript Engineer

TypeScript – Ein praktischer Einstieg zu lesen, fühlt sich an, als verbringe man Zeit mit einem warmherzigen und klugen Freund, der einem voller Begeisterung faszinierende Dinge erzählt. Sie werden sich am Ende gut unterhalten und über TypeScript aufgeklärt vorkommen, unabhängig davon, wie viel Vorwissen Sie mitbringen.

– John Reilly, Group Principal Engineer, Investec;
Maintainer, ts-loader; DefinitelyTyped-Historiker

TypeScript – Ein praktischer Einstieg ist ein umfassender und dennoch leicht verständlicher Leitfaden für TypeScript und dessen Ökosystem. Es deckt den großen Funktionsumfang von TypeScript ab, gibt Ratschläge und erläutert dessen Vor- und Nachteile auf der Grundlage umfassender Erfahrung.

– Daniel Rosenwasser, Program Manager, TypeScript, Microsoft;
Beitragender bei TC39

Dieses Buch ist meine Lieblingsquelle, wenn es um das Erlernen von TypeScript geht. Von einführenden bis hin zu fortgeschrittenen Themen ist alles klar, prägnant und umfassend dargestellt. Josh ist nicht nur ein ausgezeichneter, sondern zugleich unterhaltsamer Autor.

– Loren Sands-Ramshaw, Autor, The GraphQL Guide*;*
TypeScript SDK Engineer, Temporal

Wenn Sie ein effektiver TypeScript-Entwickler werden wollen, bietet Ihnen *TypeScript – Ein praktischer Einstieg* alles vom Einstieg bis hin zu den fortgeschritteneren Konzepten.

– Basarat Ali Syed, Principal Engineer, SEEK;
Autor, Beginning NodeJS *und* TypeScript Deep Dive*;*
Youtuber (Basarat Codes); Microsoft MVP

Dieses Buch ist eine großartige Möglichkeit, die Sprache zu erlernen, und eine perfekte Ergänzung zum *TypeScript Handbook*.

– Orta Therox, ehemaliger TypeScript-Compiler-Entwickler, Puzmo

Josh ist einer der klarsten und engagiertesten TypeScript-Vermittler der Welt, und jetzt liegt sein Wissen endlich in Buchform vor! Sowohl Anfänger als auch erfahrene Entwickler werden von der sorgfältigen Zusammenstellung und Abfolge der Themen begeistert sein. Die Tipps, Hinweise und Warnungen im klassischen O'Reilly-Stil sind ihr Gewicht in Gold wert.

– Shawn »swyx« Wang, Head of DX, Airbyte

Dieses Buch wird Ihnen wirklich helfen, TypeScript zu lernen. Das Zusammenspiel von Theoriekapiteln und Praxisprojekten sorgt für eine ausgewogene Lernerfahrung, die so gut wie jeden Aspekt der Sprache abdeckt. Die Lektüre dieses Buchs hat sogar mir »altem Hund« ein paar neue Tricks beigebracht. Endlich verstehe ich die Feinheiten von Deklarationsdateien. Äußerst empfehlenswert.

– Lenz Weber-Tronic, Full-Stack-Entwickler, Mayflower Deutschland;
Maintainer, Redux

TypeScript – Ein praktischer Einstieg ist ein leicht zugängliches, fesselndes Buch, in dem Josh seine jahrelange Erfahrung bei der Entwicklung eines TypeScript-Curriculums nutzt, um Ihnen alles, was Sie wissen müssen, in genau der richtigen Reihenfolge beizubringen. Was auch immer Ihr Programmierhintergrund ist, bei Josh und *TypeScript – Ein praktischer Einstieg* sind Sie in guten Händen.

– Dan Vanderkam, Senior Staff Software Engineer, Google;
Autor von Effective TypeScript

TypeScript – Ein praktischer Einstieg ist das Buch, das ich gerne gehabt hätte, als ich mit TypeScript angefangen habe. Joshs Begeisterung für das Unterrichten neuer Anwender ist auf jeder einzelnen Seite zu spüren. Es ist durchdacht organisiert und in leicht verdauliche Häppchen unterteilt und deckt alles ab, was Sie brauchen, um ein TypeScript-Experte zu werden.

– Brad Zacher, Software Engineer, Meta;
Core-Maintainer, typescript-eslint

TypeScript – Ein praktischer Einstieg

Typsicheres JavaScript für skalierbare Webanwendungen

Josh Goldberg

Deutsche Übersetzung von Jens Olaf Koch

O'REILLY®

Josh Goldberg

Lektorat: Ariane Hesse
Übersetzung: Jens Olaf Koch
Fachliche Unterstützung: Stefan Baumgartner
Korrektorat: Claudia Lötschert, *www.richtiger-text.de*
Satz: III-satz, *www.drei-satz.de*
Herstellung: Stefanie Weidner
Umschlaggestaltung: Michael Oréal, *www.oreal.de*
Druck und Bindung: mediaprint solutions GmbH, 33100 Paderborn

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über *http://dnb.d-nb.de* abrufbar.

ISBN:
Print 978-3-96009-218-6
PDF 978-3-96010-792-7
ePub 978-3-96010-793-4
mobi 978-3-96010-794-1

1. Auflage 2023

Wieblinger Weg 17
69123 Heidelberg

Authorized German translation of the English edition of *Learning TypeScript*
ISBN 9781098110338 © 2022 Josh Goldberg.

Dieses Buch erscheint in Kooperation mit O'Reilly Media, Inc. unter dem Imprint »O'REILLY«.
O'REILLY ist ein Markenzeichen und eine eingetragene Marke von O'Reilly Media, Inc. und wird mit Einwilligung des Eigentümers verwendet.

Hinweis:
Dieses Buch wurde mit mineralölfreien Farben auf PEFC-zertifiziertem Papier aus nachhaltiger Waldwirtschaft gedruckt. Der Umwelt zuliebe verzichten wir zusätzlich auf die Einschweißfolie. Hergestellt in Deutschland.

Schreiben Sie uns:
Falls Sie Anregungen, Wünsche und Kommentare haben, lassen Sie es uns wissen: *kommentar@oreilly.de*.

5 4 3 2 1 0

Dieses Buch widme ich meiner wunderbaren Partnerin Mariah,
die mir beigebracht hat, wie bereichernd es ist,
Hinterhofkatzen zu adoptieren, und es seither bedauert.

Inhalt

Teil II Features

Vorwort Josh Goldberg

Mein Weg zu TypeScript verlief weder direkt noch schnell. Als ich in meiner Schulzeit mit dem Programmieren begann, schrieb ich hauptsächlich in Java, dann in C++, und wie viele neue Entwickler, die mit statisch typisierten Sprachen aufgewachsen sind, sah ich auf JavaScript als die schlampige kleine Skriptsprache herab, mit denen die Leute ihre Websites pimpen.

Mein erstes größeres Projekt in dieser Sprache war ein albernes Remake des originalen Videospiels *Super Mario Bros.* in reinem HTML5/CSS/JavaScript und – ganz typisch für viele erste Projekte – ein absolutes Chaos. Zu Beginn des Projekts missfiel mir instinktiv die seltsame Flexibilität von JavaScript und das Fehlen jeglicher Leitplanken. Erst gegen Ende begann ich, die Eigenschaften und Macken von JavaScript wirklich zu respektieren: die Flexibilität der Sprache, die Möglichkeit, kleine Funktionen zu kombinieren und anzupassen, und vor allem die Fähigkeit, innerhalb von Sekunden nach dem Laden einer Seite im Browser *einfach zu funktionieren*.

Als ich mit diesem ersten Projekt fertig war, hatte ich mich in JavaScript verliebt.

Statische Analyse mit Tools wie TypeScript, die Code analysieren, ohne ihn auszuführen, verursachten bei mir anfangs ein ungutes Gefühl. *JavaScript ist so luftig und geschmeidig*, dachte ich, warum sollen wir uns mit starren Strukturen und Typen quälen? Wollte ich denn in die Welt von Java und C++ zurückkehren, die ich gerade hinter mir gelassen hatte?

Als ich mich erneut mit meinen bestehenden Projekten beschäftigte, war mir nach zehn qualvollen Minuten mit meinem alten, verworrenen JavaScript-Code klar, wie unübersichtlich die Dinge ohne statische Analyse werden können. Bei der Bereinigung des Codes stieß ich auf all die Stellen, an denen ich von einer gewissen Strukturierung profitiert hätte. Von diesem Zeitpunkt an wollte ich ständig so viel statische Analyse in meine Projekte einzubauen, wie ich konnte.

Es ist schon fast ein Jahrzehnt her, dass ich zum ersten Mal mit TypeScript gearbeitet habe, und es macht mir nach wie vor viel Spaß. Die Sprache wird kontinuierlich weiterentwickelt und mit neuen Funktionen ausgestattet und ist nützlicher denn je, wenn es darum geht, JavaScript *Sicherheit* und *Struktur* zu verleihen.

Ich hoffe, dass Sie durch die Lektüre von *TypeScript – Ein praktischer Einstieg* lernen, TypeScript auf die gleiche Weise zu schätzen, wie ich es tue, also nicht nur als Mittel, um Bugs und Tippfehler zu finden – und ganz sicher nicht als wesentliche Änderung der JavaScript-Entwurfsmuster. Sondern als JavaScript *mit Typen* und damit als ein schönes System, um festzulegen, wie unser JavaScript funktionieren soll, und dabei zu helfen, uns selbst daran zu halten.

Für wen dieses Buch gedacht ist

Wenn Sie wissen, wie man JavaScript-Code schreibt, grundlegende Befehle in einem Terminal ausführen können und daran interessiert sind, TypeScript kennenzulernen, dann ist dieses Buch genau das richtige für Sie.

Vielleicht haben Sie gehört, dass TypeScript Ihnen helfen kann, eine Menge JavaScript mit weniger Fehlern zu schreiben *(stimmt!)* oder Ihren Code gut zu dokumentieren, damit andere ihn lesen können *(stimmt auch!)*. Vielleicht haben Sie bemerkt, dass Kenntnisse in TypeScript in vielen Stellenbeschreibungen erwähnt oder in einem neu begonnenen Job erwartet werden.

Was auch immer Ihre Motivation sein mag, solange Sie die Grundlagen – Variablen, Funktionen, Closures, Geltungsbereiche und Klassen – von JavaScript kennen, wird Sie dieses Buch bis zur Beherrschung der Grundlagen und wichtigsten Merkmale der Sprache führen, auch wenn Sie keinerlei TypeScript-Vorkenntnisse mitbringen. Wenn Sie dieses Buch gelesen haben, werden Sie diese Themen verstanden haben:

- Die Geschichte der Sprache und die Gründe, warum TypeScript eine hilfreiche Ergänzung von »Vanilla-JavaScript« – purem JavaScript – ist.
- Wie ein Typsystem Code modelliert.
- Wie ein Typechecker Code analysiert.
- Wie Typanmerkungen bzw. Typannotationen, die nur während der Entwicklung genutzt werden, dabei helfen, das Typsystem zu informieren.
- Wie TypeScript mit IDEs (Integrierten Entwicklungsumgebungen) zusammenarbeitet, um Werkzeuge zur Code-Exploration und für Refaktorierungen bereitzustellen.

Und Sie werden in der Lage sein:

- die Vorteile von TypeScript und die allgemeinen Merkmale seines Typsystems zu beschreiben.
- Ihrem Code an sinnvollen Stellen Typanmerkungen hinzuzufügen.
- mäßig komplexe Typen mit den in TypeScript integrierten Ableitungen und dessen Syntax darzustellen.
- TypeScript zu verwenden, um in der lokalen Entwicklung das Refactoring von Code zu unterstützen.

Warum ich dieses Buch geschrieben habe

TypeScript ist sowohl in Unternehmen als auch im Open-Source-Bereich eine äußerst beliebte Sprache:

- In den GitHub-Auswertungen State of the Octoverse 2021 und 2020 ist sie die viertwichtigste Sprache auf der Plattform, nachdem sie 2019 und 2018 auf Platz sieben und 2017 auf Platz zehn lag.
- Im Developer Survey 2021 auf StackOverflow steht sie an dritter Stelle der beliebtesten Sprachen der Welt (72,73 % der Nutzer).
- Die Ausgabe 2020 des State of JS Survey zeigt, dass TypeScript sowohl als Build-Tool als auch als JavaScript-Variante konstant hohe Zufriedenheits- und Nutzungswerte aufweist.

Für Frontend-Entwickler wird TypeScript von allen wichtigen UI-Bibliotheken und -Frameworks gut unterstützt, darunter Angular, das TypeScript ausdrücklich empfiehlt, sowie Gatsby, Next.js, React, Svelte und Vue. Für Backend-Entwickler generiert TypeScript JavaScript, das nativ in Node.js ausgeführt wird. Deno, eine ähnliche Laufzeitumgebung, die ebenfalls vom ursprünglichen Schöpfer von Node stammt, betont die eigene, direkte Unterstützung von TypeScript-Dateien.

Trotz dieser Fülle an populärer Projektunterstützung war ich in meiner anfänglichen Lernphase doch ziemlich enttäuscht über den Mangel an guten einführenden Inhalten im Internet. Viele der online verfügbaren Dokumentationsquellen konnten nicht besonders gut erklären, was genau ein »Typsystem« ist oder wie man es benutzt. Sie setzten oft ein hohes Maß an Vorkenntnissen in JavaScript und stark typisierten Sprachen voraus oder enthielten nur recht oberflächliche Codebeispiele.

Damals kein O'Reilly-Buch mit einem niedlichen Tier-Cover zur Einführung in TypeScript finden zu können, war eine ziemliche Enttäuschung. Zwar gibt es bereits andere Bücher über TypeScript, auch von O'Reilly, aber ich konnte kein Buch finden, das sich in einer Weise auf die Grundlagen der Sprache konzentriert, wie ich es wollte, und beschreibt, warum TypeScript so funktioniert, wie es funktioniert, und wie die Kernfunktionen ineinandergreifen. Also ein Buch, das mit einer grundlegenden Erklärung der Sprache beginnt und dann nach und nach weitere Features beschreibt. Ich freue mich sehr, jetzt selbst Lesern, die noch nicht mit den Prinzipien von TypeScript vertraut sind, eine klare, umfassende Einführung in die Grundlagen der Sprache geben zu können.

Der Aufbau dieses Buchs

TypeScript – Ein praktischer Einstieg verfolgt zwei Ziele:

- Sie können es »am Stück« lesen, um TypeScript als Ganzes zu verstehen.
- Später können Sie auf dieses Buch als eine praktische, einführende Sprachreferenz zurückgreifen.

Dieses Buch bewegt sich in drei allgemeinen Abschnitten von den Konzepten zur praktischen Anwendung:

- Teil I, »Konzepte«: Beschreibt, wie JavaScript entstanden ist, was durch TypeScript hinzugefügt wird, und die Grundlagen eines *Typsystems*, wie TypeScript es erzeugt.
- Teil II, »Features«: Präzisiert die Art und Weise, in der das Typsystem mit den wichtigsten Komponenten von JavaScript, mit denen man beim Schreiben von TypeScript-Code zu tun hat, interagiert.
- Teil III, »Verwendung«: Nachdem Sie die Funktionen von TypeScript verstanden haben, erfahren Sie in diesem Abschnitt, wie Sie diese in realen Situationen einsetzen können, um das Lesen und Bearbeiten von Code zu verbessern.

Ich habe am Ende des Buchs einen Abschnitt Teil IV, »Zugaben«, eingefügt, um seltener genutzte, aber dennoch gelegentlich nützliche TypeScript-Funktionen zu behandeln. Sie müssen sie nicht in der Tiefe verstehen, um sich als TypeScript-Entwickler betrachten zu dürfen. Es handelt sich jedoch um nützliche Konzepte, die beim Einsatz von TypeScript in realen Projekten wahrscheinlich immer wieder einmal eine Rolle spielen werden. Wenn Sie die ersten drei Abschnitte verstanden haben, empfehle ich Ihnen, sich auch mit diesem zusätzlichen Abschnitt zu beschäftigen.

Jedes Kapitel beginnt mit einem Haiku zur Einstimmung auf den Inhalt und endet mit einem kleinen Wortspiel. Die Webentwicklungs-Community als Ganzes und die TypeScript-Community als Teil davon sind dafür bekannt, dass sie sich Neulingen gegenüber freundlich und aufgeschlossen verhalten. Ich habe versucht, dieses Buch so zu schreiben, dass es für Lernende wie mich, die lange, trockene Texte nicht besonders schätzen, angenehm zu lesen ist.

Beispiele und Projekte

Im Gegensatz zu vielen anderen Ressourcen, die in TypeScript einführen, konzentriert sich dieses Buch bei der Einführung einzelner Spracheigenschaften absichtlich auf eigenständige Beispiele, die nur die neuen Informationen vermitteln, anstatt sich dazu auf mittelgroße oder große Projekte zu stützen. Ich bevorzuge diese Lehrmethode, weil sie vor allem TypeScript als Sprache selbst in den Mittelpunkt stellt. TypeScript ist für so viele Frameworks und Plattformen nützlich – von denen viele regelmäßig API-Updates erfahren –, dass ich in diesem Buch nichts Framework- oder Plattformspezifisches unterbringen wollte.

Dennoch ist es beim Erlernen einer Programmiersprache äußerst hilfreich, Konzepte unmittelbar nach ihrer Einführung einzuüben. Ich empfehle dringend, nach jedem Kapitel eine Lesepause einzulegen, um den Inhalt des jeweiligen Kapitels praktisch anzuwenden. Jedes Kapitel endet deshalb mit einem Vorschlag, den passenden Abschnitt auf *https://learningtypescript.com* zu besuchen und die dort aufgeführten Beispiele und Projekte durchzuarbeiten.

Konventionen, die in diesem Buch verwendet werden

In diesem Buch werden die folgenden typografischen Konventionen verwendet:

Kursiv
: Zeigt neue Begriffe, URLs, E-Mail-Adressen, Dateinamen und Dateierweiterungen an.

`Nichtproportional`
: Wird für Programmlistings verwendet, aber auch innerhalb von Absätzen, um dort auf Programmelemente wie Variablen- oder Funktionsnamen, Datentypen, Anweisungen und Schlüsselwörter zu verweisen.

Dieses Element weist auf einen Tipp oder Vorschlag hin.

Dieses Element kennzeichnet einen allgemeinen Hinweis.

Dieses Element weist auf eine Warnung hin.

Verwendung von Codebeispielen

Zusätzliches Material (Codebeispiele, Übungen usw.) steht unter folgendem Link zum Download bereit: *https://learningtypescript.com*.

Wenn Sie technische Fragen oder ein Problem mit den Codebeispielen haben, senden Sie bitte eine E-Mail an *bookquestions@oreilly.com*.

Dieses Buch soll Sie bei der Erledigung Ihrer Aufgaben unterstützen. Wenn in diesem Buch Beispielcode angeboten wird, können Sie diesen in Ihren Programmen und Ihrer Dokumentation verwenden. Sie müssen uns nicht um Erlaubnis bitten, es sei denn, Sie reproduzieren einen wesentlichen Teil des Codes. Wenn Sie zum Beispiel ein Programm schreiben, das mehrere Teile des Codes aus diesem Buch verwendet, benötigen Sie keine Genehmigung. Der Verkauf oder Vertrieb von Beispielen aus O'Reilly-Büchern erfordert dagegen eine Genehmigung. Die Beantwortung einer Frage durch das Zitieren dieses Buchs und von Beispielcode ist nicht genehmigungspflichtig. Das Einbinden einer beträchtlichen Menge an Beispielcode aus diesem Buch in die Dokumentation Ihres Produkts erfordert dagegen eine Genehmigung.

Wir freuen uns über eine Quellenangabe, verlangen sie aber im Allgemeinen nicht. Eine Quellenangabe umfasst in der Regel den Titel, den Autor, den Verlag und die ISBN. Zum Beispiel: »*Learning TypeScript* by Josh Goldberg (O'Reilly). Copyright 2022 Josh Goldberg, 978-1-098-11033-8.«

Wenn Sie der Meinung sind, dass Ihre Verwendung von Codebeispielen nicht unter die Fair-Use-Regelung oder die oben genannte Erlaubnis fällt, können Sie uns gerne unter *permissions@oreilly.com* kontaktieren.

Danksagungen

Dieses Buch ist eine Teamleistung, und ich möchte allen, die daran mitgewirkt haben, aufrichtig danken. In erster Linie meiner übermenschlichen Chefredakteurin Rita Fernando für ihre unglaubliche Geduld und ihre hervorragende Anleitung während der gesamten Entstehungszeit. Ein zusätzliches Lob an den Rest der O'Reilly-Crew: Kristen Brown, Suzanne Huston, Clare Jensen, Carol Keller, Elizabeth Kelly, Cheryl Lenser, Elizabeth Oliver und Amanda Quinn. Ihr rockt!

Ein herzliches Dankeschön an die technischen Prüfer für ihre durchweg erstklassigen pädagogischen Hinweise und ihr TypeScript-Wissen: Mike Boyle, Ryan Cavanaugh, Sara Gallagher, Michael Hoffman, Adam Reineke und Dan Vanderkam. Dieses Buch wäre ohne euch nicht dasselbe, und ich hoffe, dass es mir gelungen ist, die mit all euren großartigen Vorschlägen verbundenen Absichten einzufangen!

Ein weiteres Dankeschön geht an die verschiedenen Kollegen und Lobredner, die das Buch punktuell bewertet und mir dabei geholfen haben, die technische Genauigkeit und die Textqualität zu verbessern: Robert Blake, Andrew Branch, James Henry, Adam Kaczmarek, Loren Sands-Ramshaw, Nik Stern und Lenz Weber-Tronic. Jede Anregung hilft!

Und schließlich möchte ich meiner Familie für ihre Liebe und Unterstützung in all den Jahren danken. Meinen Eltern Frances und Mark sowie meinem Bruder Danny – danke, dass ich meine Zeit mit Lego, Büchern und Videospielen verbringen durfte. Meiner Frau Mariah Goldberg für ihre Geduld während meiner langen Redaktions- und Schreibphasen und unseren Katzen Luci, Tiny und Jerry, dass sie mir in ihrer vornehmen Kuscheligkeit Gesellschaft geleistet haben.

Vorwort zur deutschen Ausgabe

Im Herbst 2012 ging eine neue Technologie um die Welt. Auf der Goto-Konferenz in Aarhus vorgestellt, versprach die von Legende Anders Hejlsberg mit entwickelte Programmiersprache TypeScript endlich mit allen Problemen aufzuräumen, die große JavaScript-Projekte mit sich bringen. Er stieß auf offene Ohren im hauptsächlich mit C# und Java beschäftigten Enterprise-Publikum, in den einschlägigen Internetforen (*https://news.ycombinator.com/item?id=4597716*) waren die Reaktionen eher verhalten.

Trotz aller Meinungsunterschiede wurde TypeScript von Enthusiasten und Programmiererinnen durch Konferenzen und Meetups getragen - beinahe schon evangelisierend! So schlug knapp einen Monat nach der Premiere in Aarhus TypeScript im beschaulichen Linz in Österreich auf. Der Ort: das berühmt-berüchtigte *Technologieplauscherl*. Damals steckte das Programmier-Meetup noch in den Kinderschuhen. Eine Handvoll Menschen lauschten einem lokalen Entwickler bei der Vorstellung der Features und der neuen Syntax, die TypeScript JavaScript hinzufügte. Auch hier gingen die Meinungen auseinander: Während die Java-Liga begeistert war und endlich (endlich!) in ihren gewohnten Mustern weiterarbeiten konnte, war die Riege der Webentwickler:innen, die damals hauptsächlich mit jQuery ihre Arbeit bewerkstelligen musste, eher skeptisch, wenn nicht sogar angewidert. Unter dieser Gruppe: ich. Langjähriger Webentwickler, JavaScript-Anhänger, Programmiersprachen-Nerd und heute, nach langer Zeit der Weigerung, brennender TypeScript-Enthusiast.

Nun stellt sich natürlich die Frage, wie man vom Saulus zum strikt typisierten Paulus wird. Die ersten Anzeichen haben sich schon beim Meetup damals offenbart: Denn wo zunächst fremd anmutende Syntax präsentiert wurde, war der generierte JavaScript-Code vertraut und lesbar. Das Ziel des TypeScript-Teams, mit Nähe zu JavaScript zu arbeiten und nur da zu innovieren, wo es für die Produktivität nötig war, wurde strikt eingehalten. Und nach all den Jahren hat sich gezeigt, dass die Nähe zu JavaScript größer war und ist als anfangs angenommen – verdeckt durch die Marktschreierei der frühen Tage.

Genau bei dieser großen Nähe zu JavaScript setzt Josh Goldbergs Buch an. Josh versteht es wie kein Zweiter, zu erklären, wie TypeScript als Superset von Java-

Script ein striktes Typsystem zu einer dynamischen Programmiersprache hinzufügt. Joshs Erklärungen ordnen ein, helfen dabei, TypeScript zu verstehen und stellen einen Kontext für die Fragen her, die seit einer Dekade heiß diskutiert werden.

Doch Kontext ist nicht alles. Neben der Einordnung gibt es in vielen praktischen Beispielen und Übungen genug Inhalte für die geneigten Leser, um eigene Fähigkeiten zu erwerben, sie weiter zu vertiefen und vor allem dieses Wissen praktisch anzuwenden.

So ist »TypeScript – Ein praktischer Einstieg« meines Erachtens nach ein unverzichtbares Standardwerk für alle, die sich mit dieser Sprache auseinandersetzen wollen. Das Buch bleibt – wie die Sprache – treu bei deren Wurzeln und ermöglicht es uns, deutlich produktiver zu werden. Josh ist dabei jemand, der die wunderbare Dualität und Ambivalenz der Programmiersprache TypeScript voll verstanden hat und auch vermitteln kann. Dieses Wissen spricht aus jeder Seite. Noch dazu hat Josh eine ganz eigene Art von Humor, den Sie auch in dieser Übersetzung wiederfinden werden.

Ich wünsche Ihnen viel Spaß und angenehme Lektürestunden!

Stefan Baumgartner

TEIL I

Konzepte

KAPITEL 1

Von JavaScript zu TypeScript

JavaScript heute
Unterstützt Browser jahrzehntealt
Schönheit des Webs

Bevor wir über TypeScript sprechen, müssen wir zunächst verstehen, woher es stammt: JavaScript!

Die Geschichte von JavaScript

JavaScript wurde 1995 innerhalb von 10 Tagen bei Netscape von Brendan Eich mit dem Ziel entwickelt, leicht verständlich und auf Websites einfach einsetzbar zu sein. Über die Eigenheiten von JavaScript und dessen vermeintliche Unzulänglichkeiten machen sich Entwickler seitdem gerne lustig. Auf einige dieser Eigenarten werde ich im nächsten Abschnitt eingehen.

JavaScript hat sich seit 1995 jedoch enorm weiterentwickelt! Sein Lenkungsausschuss, TC39, veröffentlicht seit 2015 jährlich neue Versionen von ECMAScript – der Sprachspezifikation, auf der JavaScript basiert – mit neuen Features, die es auf Augenhöhe mit anderen modernen Sprachen bringen. Beeindruckend ist, dass JavaScript trotz dieser regelmäßigen neuen Sprachversionen über Jahrzehnte hinweg in unterschiedlichsten Umgebungen abwärtskompatibel geblieben ist, einschließlich Browsern, eingebetteten Anwendungen und Server-Runtimes.

Heute ist JavaScript eine wunderbar flexible Sprache mit vielen Stärken. Insbesondere sollte man anerkennen, dass JavaScript trotz all seiner Macken dazu beigetragen hat, das unglaubliche Wachstum von Webanwendungen und des Internets zu ermöglichen.

> Zeigen Sie mir die perfekte Programmiersprache, und ich zeige Ihnen eine Sprache ohne Benutzer.
>
> *– Anders Hejlsberg, TSConf 2019*

Die Tücken von purem JavaScript

Entwickler sprechen oft davon, dass man mit »Vanilla-JavaScript« arbeitet, wenn die Sprache ohne signifikante Erweiterungen oder Frameworks benutzt wird: Sie meinen damit im übertragenen Sinn die vertraute, ursprüngliche, pure »Geschmacksrichtung«. Ich werde gleich darauf eingehen, warum TypeScript genau das richtige Aroma hinzufügt und durch seine Möglichkeiten hilft, die ungewöhnlich großen Schwächen von JavaScript auszubügeln. Aber zuvor müssen wir verstehen, warum Letztere so schmerzhaft sein können, zumal sie umso deutlicher zu Buche schlagen, je größer und langlebiger ein Projekt ist.

Kostspielige Freiheit

Das größte Ärgernis vieler Entwickler ist leider gleichzeitig eines der Hauptmerkmale von JavaScript: Es gibt praktisch keine Einschränkungen bei der Strukturierung des eigenen Codes. Diese Freiheit macht es zu einem Riesenspaß, ein Projekt in JavaScript zu beginnen!

Doch je mehr Dateien sich anhäufen, desto deutlicher wird, wie nachteilig sich diese Freiheit auswirken kann. Schauen Sie sich den folgenden, aus dem Zusammenhang gerissenen Codeausschnitt eines fiktiven Malprogramms an:

```
function paintPainting(painter, painting) {
  return painter
    .prepare()
    .paint(painting, painter.ownMaterials)
    .finish();
}
```

Wenn man diesen Code ohne Kontext liest, kann man nur vage Vorstellungen davon entwickeln, wie man die Funktion `paintPainting` aufruft. Hätten Sie bereits in der umgebenden Codebasis gearbeitet, könnten Sie sich vielleicht daran erinnern, dass `painter` der Rückgabewert einer `getPainter`-Funktion sein soll. Vielleicht erraten Sie sogar, dass `painting` ein String (eine Zeichenkette) ist.

Aber selbst wenn diese Annahmen zutreffen, können sie durch spätere Änderungen des Codes hinfällig werden. Vielleicht wird `painting` von einem String in einen anderen Datentyp umgewandelt, oder bestimmte Methoden von `painter` werden umbenannt.

Andere Sprachen verweigern möglicherweise die Ausführung von Code, wenn der Compiler feststellt, dass dies wahrscheinlich zu einem Absturz führen würde. Nicht so bei dynamisch typisierten Sprachen wie JavaScript – also Sprachen, die Code ausführen, ohne vorher zu prüfen, ob er möglicherweise abstürzen wird.

Die unterhaltsame Freiheit beim Schreiben von JavaScript-Code wird zu einer echten Qual, wenn man eine sichere Ausführung seines Codes anstrebt.