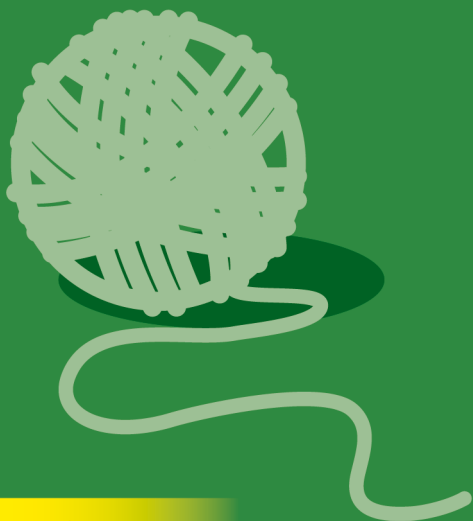


entwickler.press

3.
aktualisierte
und erweiterte
Auflage

JavaScript für Java-Entwickler

Oliver Zeigermann



Mit ECMAScript 2015
und TypeScript

schnell + kompakt

Oliver Zeigermann

JavaScript für Java-Entwickler

schnell+kompakt

entwickler.press

Oliver Zeigermann

JavaScript für Java-Entwickler (3. aktualisierte und erweiterte Auflage)
schnell+kompakt

ISBN: 978-3-86802-342-8

© 2015 entwickler.press

Ein Imprint der Software & Support Media GmbH

Bibliografische Information Der Deutschen Bibliothek

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.ddb.de> abrufbar.

Ihr Kontakt zum Verlag und Lektorat:

Software & Support Media GmbH

entwickler.press

Darmstädter Landstraße 108

60598 Frankfurt am Main

Tel.: +49 (0)69 630089-0

Fax: +49 (0)69 630089-89

lektorat@entwickler-press.de

<http://www.entwickler-press.de>

Lektorat: Corinna Neu

Korrektur: Frauke Pesch

Copy-Editor: Nicole Bechtel

Satz: Franziska Sponer

Umschlaggestaltung: Maria Rudi

Belichtung, Druck und Bindung: Media-Print Informationstechnologie GmbH, Paderborn

Alle Rechte, auch für Übersetzungen, sind vorbehalten. Reproduktion jeglicher Art (Fotokopie, Nachdruck, Mikrofilm, Erfassung auf elektronischen Datenträgern oder anderen Verfahren) nur mit schriftlicher Genehmigung des Verlags. Jegliche Haftung für die Richtigkeit des gesamten Werks kann, trotz sorgfältiger Prüfung durch Autor und Verlag, nicht übernommen werden. Die im Buch genannten Produkte, Warenzeichen und Firmennamen sind in der Regel durch deren Inhaber geschützt.

Inhaltsverzeichnis

1	Einleitung	9
2	Grundlagen	13
2.1	Hallo Welt	13
2.2	Typen	16
2.3	Variablen	24
2.4	Syntax	26
2.5	== vs. ===	29
2.6	undefined vs. null	30
2.7	Kontrollstrukturen	32
2.8	Exceptions	35
3	Funktionen	39
3.1	Bürger erster Klasse	39
3.2	Sichtbarkeitsbereiche (Scopes)	40
3.3	Strict Mode	42
3.4	Parameter	44
3.5	Funktionen höherer Ordnung	49
3.6	Hoisting	51

4	Objekte, Klassen, Prototypen und Vererbung	55
4.1	Prototypische Vererbung	55
4.2	Klassen	58
4.3	Vererbung	62
5	Closures und Module	67
5.1	Closures	67
5.2	Das Revealing Module Pattern	70
5.3	Standardformate für Module	71
6	Fortgeschrittene Themen	75
6.1	Wie wird „this“ gebunden?	75
6.2	Weitere OO-Patterns	81
6.3	JSON	84
6.4	Reguläre Ausdrücke	85
7	ECMAScript 2015 aka ES6	87
7.1	Ablaufumgebungen	87
7.2	Block Scope	88
7.3	Spracherweiterungen	91
7.4	Klassen und erweiterte Objektliterale	94
7.5	Module, Exporte und Importe	100
7.6	Erweiterungen bei Funktionen	103
7.7	Iteratoren und for..of	107
7.8	Weitere Eigenschaften	109

8	Einführung in TypeScript	117
8.1	Einführung	117
8.2	Typen und Interfaces	121
8.3	Erweiterungen für Klassen	127
8.4	Generics	128
8.5	Namespaces und Module	129
8.6	Was noch?	131
	Stichwortverzeichnis	135

Einleitung

JavaScript sieht auf den ersten Blick syntaktisch aus wie ein vereinfachtes Java. Somit haben viele Java-Entwickler das Gefühl, die Sprache JavaScript eigentlich zu kennen und sie daher nicht lernen zu müssen. Allerdings ist das Verhalten von JavaScript deutlich anders als das von Java.

Dieses Buch ist für Java-Entwickler gedacht, die mit so wenig Mühe wie möglich einen Überblick über die Sprache JavaScript erlangen wollen oder müssen. Missverständnisse werden ausgeräumt und eine Beherrschung der Muster und Grundkonzepte von JavaScript wird vermittelt.

Bibliotheken und Frameworks werden in diesem Buch ganz bewusst nicht behandelt. Hier geht es ausschließlich um die Sprache JavaScript und um Patterns, die für Java-Entwickler wichtig sind. Die Beschreibung der Sprache und der Patterns geschieht unabhängig von allen Bibliotheken und Frameworks, d. h. egal, welches Framework oder welche Bibliothek ihr einsetzen wollt, dieses Buch vermittelt euch die dazu notwendigen Grundlagen der Sprache JavaScript.

Dieses Buch ist keine Referenz und erhebt keinen Anspruch auf Vollständigkeit. Ich verzichte auf alle Details, die nicht wirklich notwendig für das Verständnis der Sprache sind. Zu jedem Thema gibt es aber Referenzen auf die ECMAScript-Spezifikation, Er-

klärungen beim Mozilla Development Network [1] oder andere passende Links.

Ich beziehe mich in den ersten Kapiteln auf die ECMAScript-Version 5.1 [2, 3]. Diese Version wird von allen modernen Browsern ab Internet Explorer 9 unterstützt. Eine kurze Erklärung zu ECMAScript: ECMAScript ist der Standard und JavaScript ist dazu die Implementierung.

In diesem Jahr (2015) ist die ECMAScript-Version 2015 [4] veröffentlicht worden. Andere Namen dafür sind ECMAScript 6 oder ES6. In Kapitel 7 werden wir uns die wichtigsten Neuerungen in ECMAScript 2015 ansehen.

Inhalt

Ihr könnt dieses Buch von vorn bis hinten durchlesen. In diesem Fall werden alle notwendigen Grundlagen der Sprache JavaScript vermittelt – zugeschnitten auf Java-Entwickler.

Je nach Interesse, Zeit und Vorkenntnissen könnt ihr aber auch nur einzelne Kapitel lesen. Wenn ihr noch wenig oder keine Erfahrung mit JavaScript gemacht habt, solltet ihr zumindest Kapitel 2 (Grundlagen) und Kapitel 3 (Funktionen) lesen. Sie bilden die Grundlage für die folgenden Kapitel. Insbesondere in Kapitel 3 sehen wir einiges, das auch für manche erfahrene JavaScript-Programmierer neu sein könnte.

In Kapitel 4 (Objekte, Prototypen und Vererbung) schauen wir uns an, wie Vererbung in JavaScript funktioniert und wie man die aus Java bekannten Mechanismen von Klassen und Vererbung auch in der JavaScript-Welt anwenden kann. Dazu nutzen wir einige Best Practices.

In Kapitel 5 (Module) schauen wir auf Modulkonzepte und Closures. Für dieses Kapitel solltet ihr das Wissen aus Kapitel 2 und 3

haben, Kapitel 4 ist dafür nicht unbedingt notwendig. Wir schließen das Kapitel mit einer Betrachtung der gängigen Modulformate AMD und CommonJS.

In Kapitel 6 (Fortgeschrittene Themen) kommen alle wichtigen Themen, die keine Grundlagen mehr sind. Besonders spannend sind dabei zusätzliche OO-Muster, die allerdings ein Verständnis der Themen aus Kapitel 4 erfordern.

Das siebte Kapitel wird, wie bereits erwähnt, alle besonders relevanten Neuerungen von ECMAScript 2015 zusammenfassen.

Das letzte Kapitel (TypeScript) behandelt eine typisierte Erweiterung von JavaScript, die die Entwicklung in großen Projekten erleichtert. Da TypeScript auf ECMAScript 2015 basiert, solltet ihr vorher auch Kapitel 7 gelesen haben.

Danksagung

Ich möchte mich bei allen bedanken, die bei diesem Buch mitgeholfen haben. Vielen Dank!

Links & Literatur

- [1] Das Mozilla Development Network für JavaScript:
<https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- [2] HTML-Version der ECMAScript-Spezifikation 5.1:
<http://www.ecma-international.org/ecma-262/5.1>
- [3] PDF-Version der ECMAScript-Spezifikation 5.1:
<http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf>
- [4] ECMAScript 2015: *<http://www.ecma-international.org/ecma-262/6.0/>*

Grundlagen

2.1 Hallo Welt

Die typische Ablaufumgebung für ein JavaScript-Programm ist der Browser. Zwar gibt es seit einiger Zeit mit Node.js [1] die Möglichkeit, JavaScript auch auf dem Server und von der Kommandozeile aufzurufen, wir werden uns hier aber auf die Ausführung im Browser beschränken. Im Browser läuft JavaScript durch Einbetten in eine HTML-Seite.

Die JavaScript-Konsole

Starten wir mit einem kurzen „Hallo-Welt“-Programm, das wir in der JavaScript-Konsole des Browsers laufen lassen. Dazu lernen wir das globale Object *console* (nicht zu verwechseln mit der JavaScript-Konsole) kennen, das die Funktion *log()* bietet. Mit dieser Funktion könnt ihr – ähnlich wie *System.out.println()* in Java – eine Ausgabe auf der Konsole erzeugen:

```
console.log("Hallo, Welt");
```

Je nachdem, welchen Browser ihr nutzt, kommt ihr unterschiedlich an die Konsole heran. Mit dem Internet Explorer geht das z. B. über F12 oder das Zahnrad oben rechts, im Chrome über die Tastenkombinationen *CTRL + Shift + J* (auf dem Mac *⌘ + J*) oder über das so genannte „Hotdog-Menü“ rechts oben [2]. Hier könnt ihr nun direkt den JavaScript-Schnipsel von oben aufrufen und

seht als Ausgabe den erwarteten „Hallo, Welt“-String. Zusätzlich erscheint eine Zeile mit der Ausgabe *undefined*, da JavaScript-Konsolen typischerweise auch die Rückgabe einer Funktion ausgeben. Die ist in diesem Fall eben *undefined*. Salopp kann man sich diese Rückgabe so vorstellen wie das, was bei einer *void*-Methode in Java zurückgeliefert wird, nämlich nichts.

JavaScript in HTML-Seiten

Für JavaScript-Code, der mehr als ein kleines Experiment ist, empfiehlt es sich, ihn aus einer HTML-Seite heraus aufzurufen. Das geht, indem man den Code direkt in die HTML-Seite einbettet wie in dem folgenden Beispiel.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Hallo, Welt!</title>
  <script>
    alert("Hallo, Welt");
  </script>
</head>
<body>
</body>
</html>
```

Wenn ihr einen Rechner zur Hand habt, tippt dieses kleine Beispiel einmal mit einem Texteditor ab, speichert es unter *index.html* und ruft es mit dem Browser auf. Es sollte sich eine Nachrichtenbox mit dem Text „Hallo, Welt“ zeigen. Im Chrome sieht das Ganze in etwa wie in Abbildung 2.1 aus.