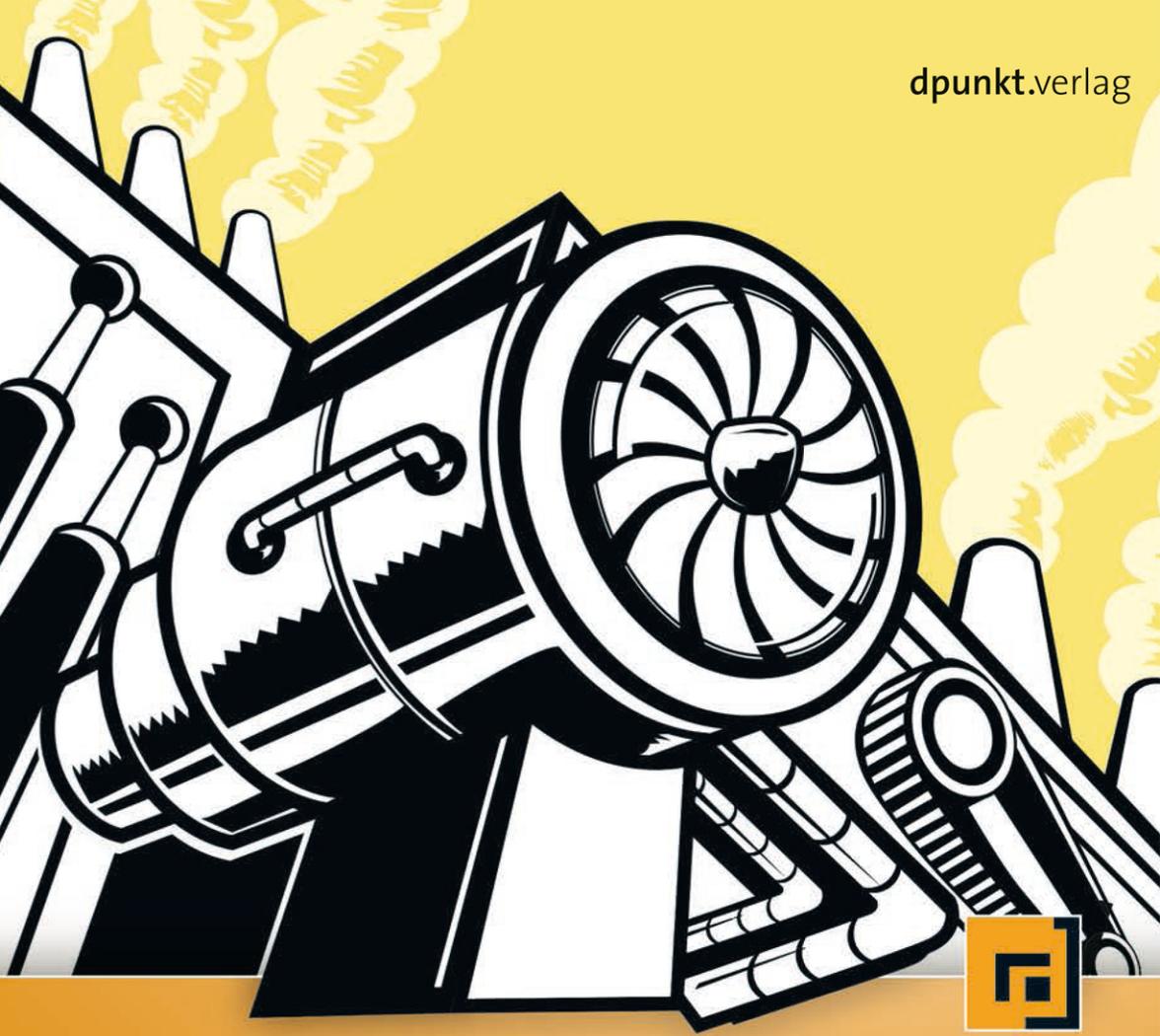


dpunkt.verlag



Nicholas C. Zakas

# JavaScript objektorientiert

Verständlicher, flexibler und effizienter programmieren



Papier  
**plus<sup>+</sup>**  
PDF.

Zu diesem Buch – sowie zu vielen weiteren dpunkt.büchern – können Sie auch das entsprechende E-Book im PDF-Format herunterladen. Werden Sie dazu einfach Mitglied bei dpunkt.plus<sup>+</sup>:

[www.dpunkt.de/plus](http://www.dpunkt.de/plus)

Nicholas Zakas

# JavaScript objektorientiert

**Verständlicher, flexibler,  
effizienter programmieren**



dpunkt.verlag

Nicholas Zakas

Übersetzung: Volkmar Gronau, Flensburg  
Lektorat: Dr. Michael Barabas  
Copy-Editing: Ursula Zimpfer, Herrenberg  
Satz: Frank Heidt  
Herstellung: Susanne Bröckelmann  
Umschlaggestaltung: Helmut Kraus, [www.exclam.de](http://www.exclam.de)  
Druck und Bindung: M.P. Media-Print Informationstechnologie 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: 978-3-86490-202-4  
ISBN-PDF: 978-3-86491-553-6  
ISBN-ePub: 978-3-86491-554-3

#### 1. Auflage

Translation Copyright für die deutschsprachige Ausgabe © 2014 dpunkt.verlag GmbH  
Wieblingerg Weg 17  
69123 Heidelberg

Copyright der amerikanischen Originalausgabe © 2014 No Starch Press.  
Title of American original: The Principles of Object-Oriented JavaScript  
ISBN: 978-1-59327-540-2

Die vorliegende Publikation ist urheberrechtlich geschützt. Alle Rechte vorbehalten. Die Verwendung der Texte und Abbildungen, auch auszugsweise, ist ohne die schriftliche Zustimmung des Verlags urheberrechtswidrig und daher strafbar. Dies gilt insbesondere für die Vervielfältigung, Übersetzung oder die Verwendung in elektronischen Systemen.

Es wird darauf hingewiesen, dass die im Buch verwendeten Soft- und Hardware-Bezeichnungen sowie Markennamen und Produktbezeichnungen der jeweiligen Firmen im Allgemeinen warenzeichen-, marken- oder patentrechtlichem Schutz unterliegen.

Alle Angaben und Programme in diesem Buch wurden mit größter Sorgfalt kontrolliert. Weder Autor noch Verlag können jedoch für Schäden haftbar gemacht werden, die in Zusammenhang mit der Verwendung dieses Buches stehen.

5 4 3 2 1 0

## Danksagungen

Ich möchte Kate Matsudaira danken, die mich davon überzeugte, dass eine Eigenpublikation als E-Book die beste Möglichkeit sei, diese Informationen zu verbreiten. Ohne ihren Rat würde ich wahrscheinlich immer noch herauszufinden versuchen, was ich mit den hier enthaltenen Informationen tun soll.

Abermals möchte ich Rob Friesel für sein hervorragendes Feedback zu einem frühen Entwurf dieses Buchs und Cody Lindley für seine Vorschläge danken. Weiterer Dank gilt Angus Croll für die fachliche Durchsicht der endgültigen Fassung – dank seiner Erbsenzählerei ist dieses Buch viel besser geworden!

Danke auch an Bill Pollock, den ich auf einer Konferenz traf und der den Ball ins Rollen brachte, um dieses Buch zu veröffentlichen.



## Der Autor

Nicholas C. Zakas ist Softwareingenieur bei Box und widmet sich als bekannter Redner und Autor den neuesten Best Practices in der JavaScript-Entwicklung. Seine Kenntnisse konnte er während seiner fünf Jahre bei Yahoo! verfeinern, wo er als leitender Frontend-Ingenieur für die Yahoo!-Homepage tätig war. Er hat mehrere Bücher geschrieben, darunter *Maintainable JavaScript* (O'Reilly Media, 2012) und *Professional JavaScript for Web Developers* (Wrox, 2012).

## Der Fachgutachter

Angus Croll stammt ursprünglich aus Großbritannien und gehört jetzt zum Webframework-Team von Twitter in San Francisco. Außerdem ist er Koautor des Open-Source-Frameworks Flight von Twitter und in leitender Position mit dessen Wartung befasst. Er ist gleichermaßen von JavaScript und Literatur besessen und tritt leidenschaftlich für eine stärkere Beteiligung von Künstlern und kreativen Köpfen in der Softwareentwicklung ein. Angus tritt häufig als Redner bei Konferenzen in aller Welt auf und arbeitet zurzeit an zwei Büchern für No Starch Press. Auf Twitter können Sie ihm unter @angustweets folgen.



Der Name Nicholas Zakas steht gewissermaßen für die JavaScript-Entwicklung. Ich könnte hier noch seitenlang seine beruflichen Auszeichnungen aufzählen, aber das werde ich nicht tun. Nicholas ist als äußerst talentierter JavaScript-Entwickler und Autor gut bekannt und braucht keine Vorstellung. Ich möchte allerdings einige persönliche Gedanken äußern, bevor ich auf den Inhalt dieses Buchs lobend eingehe.

Meine Beziehung zu Nicholas begann in den Jahren, als ich noch ein JavaScript-Schüler war und seine Bücher studierte, seine Blogbeiträge las, ihm als Redner zuhörte und seine Twitter-Meldungen verfolgte. Persönlich haben wir uns zum ersten Mal vor einigen Jahren getroffen, als ich ihn bat, bei einer jQuery-Konferenz als Redner aufzutreten. Er bedachte die jQuery-Community mit einem sehr fundierten Vortrag, und danach haben wir öffentlich und privat über das Internet miteinander gesprochen. In dieser Zeit habe ich ihn immer mehr zu bewundern gelernt, und zwar nicht nur als Entwickler und als Führungspersönlichkeit der JavaScript-Community. Was er sagt, ist immer freundlich und überlegt, sein Verhalten stets liebenswürdig. Als Entwickler, als Redner und als Autor ist er stets bestrebt, zu helfen, zu lehren und zu verbessern. Wenn er spricht, sollten Sie ihm zuhören, und zwar nicht nur, weil er ein JavaScript-Experte ist, sondern wegen seines Charakters, der über seine berufliche Qualifikation hinaus deutlich wird.

Der Titel des Buchs und die Einleitung lassen Nicholas' Absicht erkennen: Er hat dieses Buch geschrieben, um an Klassen gewöhnte Programmierer (also solche, die mit C++ oder Java vertraut sind) den Wechsel zu einer klassenlosen Sprache zu erleichtern. Er erklärt ihnen hier, wie sie Kapselung, Aggregation, Vererbung und Polymorphismus umsetzen, wenn sie JavaScript-Code schreiben. Dies ist ein ideales Lehrbuch, um

erfahrene Programmierer mit der objektorientierten JavaScript-Entwicklung vertraut zu machen. Entwicklern, die von einer anderen Sprache kommen, bietet dieses Buch eine knappe, sorgfältig formulierte Lektüre über JavaScript.

Dieses Buch ist jedoch auch für Programmierer aus der JavaScript-Community geeignet. Die Kenntnisse vieler JavaScript-Entwickler über Objekte orientieren sich nur an ECMAScript 3 (ES3), weshalb sie eine gute Einführung in die Merkmale von Objekten in ECMAScript 5 (ES5) benötigen. Dieses Buch stellt eine solche Einführung dar und bringt sie vom ES3-Kenntnistand über Objekte zu dem von ES5.

Vielleicht denken Sie jetzt: »Na und? In vielen Büchern gibt es Kapitel oder Hinweise zu den Ergänzungen, die in ES5 zu JavaScript gemacht wurden.« Das mag schon sein, aber ich glaube, dass dies zurzeit das einzige Buch ist, das sich auf die Beschaffenheit von Objekten konzentriert, indem es ES5-Objekte im gesamten Text ins Rampenlicht stellt. In diesem Buch finden Sie nicht nur eine zusammenhängende Einführung in ES5-Objekte, sondern auch Einzelheiten über ES3, die Sie kennen müssen, während Sie die Erweiterungen aus ES5 lernen.

Ich bin der festen Überzeugung, dass dieses Buch mit seinem Schwerpunkt auf objektorientierten Prinzipien und den Neuerungen bei Objekten in ES5 genau das Buch ist, das wir brauchen, während wir auf die Erweiterung unserer Skriptumgebung durch ES6 warten.

Cody Lindley ([www.codylindley.com](http://www.codylindley.com))  
Autor von *JavaScript Enlightenment*,  
*DOM Enlightenment* und *jQuery Enlightenment*

Boise, Idaho  
16. Dezember 2013

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Zielgruppe .....	2
1.2	Überblick .....	3
1.3	Hilfe und Unterstützung .....	4
<b>2</b>	<b>Primitive Typen und Referenztypen</b>	<b>5</b>
2.1	Was sind Typen? .....	6
2.2	Primitive Typen .....	6
	Primitive Typen identifizieren .....	9
	Primitive Methoden .....	10
2.3	Referenztypen .....	11
	Objekte erstellen .....	11
	Objekte dereferenzieren .....	13
	Eigenschaften hinzufügen und entfernen .....	13
2.4	Eingebaute Typen instanziiieren .....	14
	Literalformen .....	15
	Objekt- und Arraylitterale .....	15
	Funktionslitterale .....	16
	Litterale für reguläre Ausdrücke .....	17
2.5	Zugriff auf Eigenschaften .....	18
2.6	Referenztypen identifizieren .....	19
2.7	Arrays identifizieren .....	20
2.8	Wrapper-Typen für primitive Typen .....	21
2.9	Zusammenfassung .....	24

<b>3</b>	<b>Funktionen</b>	<b>25</b>
3.1	Der Unterschied zwischen Deklarationen und Ausdrücken . . .	26
3.2	Funktionen als Werte . . . . .	27
3.3	Parameter . . . . .	29
3.4	Überladung. . . . .	31
3.5	Objektmethoden. . . . .	33
	Das this-Objekt . . . . .	34
	Den Wert von this ändern . . . . .	35
3.6	Zusammenfassung . . . . .	39
<b>4</b>	<b>Objekte</b>	<b>41</b>
4.1	Eigenschaften definieren . . . . .	41
4.2	Eigenschaften ermitteln . . . . .	43
4.3	Eigenschaften entfernen . . . . .	45
4.4	Aufzählung. . . . .	46
4.5	Arten von Eigenschaften. . . . .	48
4.6	Eigenschaftsattribute . . . . .	50
	Gemeinsame Attribute . . . . .	50
	Attribute von Dateneigenschaften . . . . .	52
	Attribute von Zugriffseigenschaften . . . . .	54
	Mehrere Eigenschaften definieren . . . . .	56
	Eigenschaftsattribute abrufen . . . . .	57
4.7	Objektänderungen verhindern . . . . .	58
	Erweiterungen verhindern . . . . .	59
	Objekte versiegeln . . . . .	59
	Objekte einfrieren . . . . .	61
4.8	Zusammenfassung . . . . .	63

<b>5</b>	<b>Konstruktoren und Prototypen</b>	<b>65</b>
5.1	Konstruktoren	65
5.2	Prototypen	70
	Die Eigenschaft [[Prototype]]	71
	Prototypen in Konstruktoren verwenden	74
	Prototypen ändern	78
	Prototypen der eingebauten Objekte	80
5.3	Zusammenfassung	81
<b>6</b>	<b>Vererbung</b>	<b>83</b>
6.1	Prototypverkettung und Object.prototype	83
	Von Object.prototype geerbte Methoden	84
	Object.prototype ändern	86
6.2	Objektvererbung	88
6.3	Konstruktorvererbung	91
6.4	Konstruktordiebstahl	95
6.5	Zugriff auf die Methoden des Supertyps	97
6.6	Zusammenfassung	98
<b>7</b>	<b>Objektmuster</b>	<b>101</b>
7.1	Private und privilegierte Elemente	101
	Das Modul-Muster	101
	Private Elemente für Konstruktoren	104
7.2	Mixins	106
7.3	Bereichssichere Konstruktoren	114
7.4	Zusammenfassung	116
	<b>Index</b>	<b>117</b>

