

# JavaScript Programmierung von Kopf bis Fuß



Nehmen Sie sich vor  
typischen JavaScript-  
Fallstricken in Acht

Ein Lernbuch zur  
JavaScript-Programmierung

Starten Sie Ihre  
neue Karriere in  
einem Kapitel



Vermeiden Sie  
peinliche Fehler bei  
der Typumwandlung



Trainieren Sie Ihre  
Denkmuskulatur  
mit 120 Rätseln und  
Übungen



Erfahren Sie, warum alles,  
was Ihre Freunde zu  
Funktionen und Objekten  
wissen, vermutlich falsch ist



Eric Freeman & Elisabeth Robson

Deutsche Übersetzung von Jørgen W. Lang

Die Informationen in diesem Buch wurden mit größter Sorgfalt erarbeitet. Dennoch können Fehler nicht vollständig ausgeschlossen werden. Verlag, Autoren und Übersetzer übernehmen keine juristische Verantwortung oder irgendeine Haftung für eventuell verbliebene Fehler und deren Folgen. Das heißt, wenn Sie beispielsweise ein Kernkraftwerk unter Verwendung dieses Buchs betreiben möchten, tun Sie das auf eigene Gefahr.

Alle Warennamen werden ohne Gewährleistung der freien Verwendbarkeit benutzt und sind möglicherweise eingetragene Warenzeichen. Der Verlag richtet sich im Wesentlichen nach den Schreibweisen der Hersteller. Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Alle Rechte vorbehalten einschließlich der Vervielfältigung, Übersetzung, Mikroverfilmung sowie Einspeicherung und Verarbeitung in elektronischen Systemen.

Kommentare und Fragen können Sie gerne an uns richten:

O'Reilly Verlag  
Balthasarstr. 81  
50670 Köln  
E-Mail: [komentar@oreilly.de](mailto:komentar@oreilly.de)

Copyright der deutschen Ausgabe:  
© 2015 by O'Reilly Verlag GmbH & Co. KG



Die Originalausgabe erschien 2014 unter dem Titel  
Head First JavaScript Programming bei O'Reilly Media, Inc.

Bei der Erstellung dieses Buchs wurden keinerlei Variablen verletzt.

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

Übersetzung und deutsche Bearbeitung: Jørgen W. Lang, Hatten  
Lektorat: Inken Kiupel und Imke Hirschmann, Köln  
Korrektur: Sibylle Feldmann, Düsseldorf  
Satz: Ulrich Borstelmann, Dortmund  
Umschlaggestaltung: Randy Comer, Sebastopol & Michael Oreal, Köln  
Produktion: Andrea Miß, Köln  
Belichtung, Druck und buchbinderische Verarbeitung: Media-Print, Paderborn

ISBN 978-3-95561-866-7

Dieses Buch ist auf 100% chlorfrei gebleichtem Papier gedruckt.

Für JavaScript. Du kommst nicht aus gutem Hause –  
trotzdem hast du bisher noch jede Sprache in den Schatten  
gestellt, die dich im Browser herausgefordert hat.

## Die Autoren von JavaScript-Programmierung von Kopf bis Fuß



Eric Freeman

**Eric** wurde von Kathy Sierra, der Mitschöpferin der Von Kopf bis Fuß-Reihe, »als eine der seltenen Persönlichkeiten« bezeichnet, »die die Sprache, die Umgangsformen und das Selbstverständnis von so unterschiedlichen Leuten wie Hackern, Managern, Ingenieuren und Wissenschaftlern verstehen«.

Beruflich hat Eric kürzlich einen Schlussstrich unter ein Jahrzehnt in der Führungsetage eines Medienunternehmens gezogen – er war CTO von Disney Online und Disney.com bei der Walt Disney Company. Zurzeit kümmert er sich um WickedlySmart, das er gemeinsam mit Elisabeth gegründet hat.

Eric ist von Haus aus Informatiker. Er hat während seiner Promotion an der Yale University bei der IT-Koryphäe David Gelernter studiert. Seine Doktorarbeit gilt als bahnbrechendes Werk zu Alternativen zur sogenannten Schreibtischmetapher und stellt außerdem die erste Implementierung von Activity-Streams dar, einem Konzept, das er gemeinsam mit Dr. Gelernter entwickelt hat.

In seiner Freizeit befasst sich Eric intensiv mit Musik; sein jüngstes Projekt, eine Zusammenarbeit mit dem Ambient Music-Pionier Steve Roach, finden Sie unter dem Namen Immersion Station im iPhone App Store.

Eric lebt mit seiner Frau und seiner kleinen Tochter auf Bainbridge Island. Seine Tochter stattet Erics Studio häufig Besuche ab, wo sie mit größtem Vergnügen an den Knöpfen seiner Synthesizer und Audiogeräte herumspielt.

Schreiben Sie Eric unter [eric@wickedlysmart.com](mailto:eric@wickedlysmart.com) oder besuchen Sie seine Site unter <http://ericfreeman.com>.



Elisabeth Robson

**Elisabeth** ist Software-Ingenieurin, Autorin und Trainerin. Ihre Technologieleidenschaft rührt schon aus ihren Studententagen an der Yale University, an der sie einen Master in Informatik erwarb und eine parallelisierte, visuelle Programmiersprache und Softwarearchitektur entwarf.

Mit dem Internet befasst sich Elisabeth seit seiner Entstehung. Sie ist Mitautorin einer preisgekrönten Website, dem Ada Project, einer der ersten Websites, die Frauen in der Informatik helfen sollen, online berufsbezogene Informationen und Unterstützung durch MentorInnen zu finden.

Elisabeth ist Mitbegründerin von WickedlySmart, einem Onlinelehrportal, das sich primär mit Webtechnologien befasst, wo sie Bücher, Artikel, Videos und anderes verfasst. Zuvor hat Elisabeth als Director of Special Projects bei O'Reilly Media Workshops und Onlinekurse zu einer Vielzahl von Technologiethemen geschrieben und ihre Leidenschaft für die Konzeption von Lernerlebnissen entwickelt, die Menschen helfen, technische Themen zu verstehen. Vor ihrer Arbeit bei O'Reilly verbrachte Elisabeth ihre Zeit damit, bei der Walt Disney Company Feenstaub zu verstreuen und die Forschungs- und Entwicklungsabteilungen im Bereich digitaler Medien zu leiten.

Wenn sie nicht vor ihrem Rechner sitzt, finden Sie Elisabeth beim Wandern, Radfahren oder Kajakfahren, immer mit der Kamera in der Hand, oder bei der Zubereitung vegetarischer Mahlzeiten.

Sie können ihr unter [beth@wickedlysmart.com](mailto:beth@wickedlysmart.com) E-Mails zukommen lassen oder ihr Blog unter <http://elisabethrobson.com> besuchen.

## Über den Übersetzer dieses Buchs

**Jørgen W. Lang** lebt und arbeitet als freier Autor (»CSS Kochbuch«), Übersetzer und Frontend-Entwickler in Oldenburg/Niedersachsen. Mitte der neunziger Jahre des vergangenen Jahrhunderts begann er, sich mit dem damals noch jungen World Wide Web und seinen Möglichkeiten zu beschäftigen. Pünktlich zum Jahrtausendwendejahr erschien seine erste Übersetzung für den O'Reilly Verlag. Mittlerweile ist der Umfang auf mehr als 10.000 Seiten angewachsen.

Mit großer Energie und Ausdauer bringt Jørgen seit fast schon zwei Jahrzehnten Webseiten bei, das zu tun, was von ihnen erwartet wird – unabhängig davon, auf welchem Gerät sie betrachtet werden (elektrische Zahnbürsten ausgenommen).

Das zweite Standbein von Jørgen Lang ist die Musik. Außerhalb der Welt der semantischen Elemente, Selektoren und Objekte hat er sich einen Namen als hervorragender Gitarrist, Sänger, Komponist und Arrangeur gemacht und kann auf eine Vielzahl veröffentlichter Alben und über mehrere hundert Konzerte in aller Welt (z.B. für die UNESCO in Seoul) zurückblicken.

Nach der Eröffnung eines Webdev-Freizeitparks ist geplant, einen oder mehrere Exoplaneten zu terraformen und zu besiedeln.

## **Ergänzende Bücher von O'Reilly**

HTML5 & CSS3

HTML5 – kurz & gut

CSS Kochbuch

CSS – kurz & gut

JavaScript – Das umfassende Referenzwerk

JavaScript – kurz & gut

## **Weitere Bücher aus unserer *Von-Kopf-bis-Fuß-Reihe***

HTML5-Programmierung von Kopf bis Fuß

Mobiles Web von Kopf bis Fuß

Webdesign von Kopf bis Fuß

jQuery von Kopf bis Fuß

PHP & MySQL von Kopf bis Fuß

C von Kopf bis Fuß

Datenanalyse von Kopf bis Fuß

Entwurfsmuster von Kopf bis Fuß

Java von Kopf bis Fuß

Netzwerke von Kopf bis Fuß

Objektorientierte Analyse und Design von Kopf bis Fuß

Programmieren von Kopf bis Fuß

Python von Kopf bis Fuß

Servlets & JSP von Kopf bis Fuß

Softwareentwicklung von Kopf bis Fuß

SQL von Kopf bis Fuß

Statistik von Kopf bis Fuß

## Der Inhalt (in der Übersicht)

	Einführung	xxiii
1	Ein Sprung ins kalte JavaScript-Wasser: <i>Zeit für nasse Füße</i>	1
2	Richtigen Code schreiben: <i>Die nächsten Schritte</i>	43
3	Einführung in Funktionen: <i>Jetzt funkt's</i>	79
4	Etwas Ordnung in die Daten bringen: <i>Arrays</i>	125
5	Objekte verstehen: <i>Ein Ausflug nach Objectville</i>	173
6	Interaktion mit der Webseite: <i>Das DOM kennenlernen</i>	229
7	Typen, Gleichheit, Umwandlung und der ganze Rest: <i>Seriöse Typen</i>	265
8	Die Einzelteile zusammenfügen: <i>Eine App schreiben</i>	317
9	Asynchron programmieren: <i>Events be-handeln</i>	381
10	Funktionen erster Klasse: <i>Befreite Funktionen</i>	429
11	Anonyme Funktionen, Geltungsbereiche und Closures: <i>Seriöse Funktionen</i>	475
12	Fortgeschrittene Objektkonstruktion: <i>Objekte erstellen</i>	521
13	Prototypen verwenden: <i>Extrastarke Objekte</i>	563
A	Die Top Ten der Themen, die wir nicht behandelt haben: <i>Was übrig bleibt</i>	623
	Index	641

## Der Inhalt (jetzt ausführlich)

### Einführung

**Ihr Gehirn und JavaScript.** Sie versuchen, etwas zu lernen, und Ihr *Hirn* tut sein Bestes, damit das Gelernte nicht *hängen bleibt*. Es denkt nämlich: »Wir sollten lieber Platz für wichtigere Dinge lassen, z. B. für das Wissen darüber, welche Tiere einem gefährlich werden könnten oder dass es eine ganz schlechte Idee ist, nackt Snowboard zu fahren.« Tja, wie schaffen wir es nun, Ihr Gehirn davon zu überzeugen, dass Ihr Leben davon abhängt, etwas über JavaScript-Programmierung zu wissen?



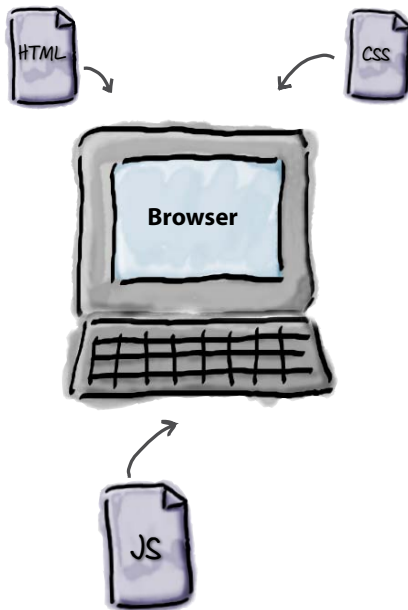
Für wen ist dieses Buch?	xxiv
Wir wissen was Sie gerade denken.	xxv
Wir stellen uns unseren Leser als einen aktiv Lernenden vor	xxvi
Metakognition: Nachdenken übers Denken	xxvii
Das haben WIR getan	xxviii
Das können SIE tun, um sich Ihr Gehirn untertan zu machen	xxix
Lies mich!	xxx
Fachgutachter	xxxiii
Danksagungen	xxxiv

# 1

## Ein Sprung ins kalte JavaScript-Wasser

### Zeit für nasse Füße

**JavaScript verleiht Superkräfte.** Es ist die **Programmiersprache** des Webs. Mit JavaScript bekommen Ihre Webseiten nämlich **Verhalten**. Das heißt: keine trockenen, langweiligen statischen Seiten mehr, die einfach nur dasitzen und Sie anstarren. Mit JavaScript erreichen Sie Ihre Benutzer direkt. Sie können auf interessante Events reagieren, Daten aus dem Web einbinden, Diagramme direkt in Ihren Seiten erstellen und vieles mehr. Haben Sie JavaScript einmal verstanden, können Sie vollkommen neue Verhaltensweisen für Ihre Benutzer erstellen.



- Wie JavaScript funktioniert 2
- Wie Sie JavaScript schreiben werden 3
- JavaScript in die Seite einbinden 4
- JavaScript, du hast einen langen Weg hinter dir ... 6
- Anweisungen definieren 10
- Variablen und Werte 11
- Finger weg von der Tastatur! 12
- Der richtige Ausdruck 15
- Dinge mehr als einmal tun 17
- Wie eine while-Schleife funktioniert 18
- Entscheidungen in JavaScript 22
- Und wenn Sie richtig VIELE Entscheidungen treffen müssen 23
- Sprechen Sie Ihre Benutzer direkt an 25
- Ein näherer Blick auf console.log 27
- Die Konsole öffnen 28
- Die erste richtige JavaScript-Applikation 29
- Wie bekomme ich den Code in meine Seite?  
(... mal die Möglichkeiten zählen ...) 32
- Wir müssen euch trennen 33





# Richtigen Code schreiben

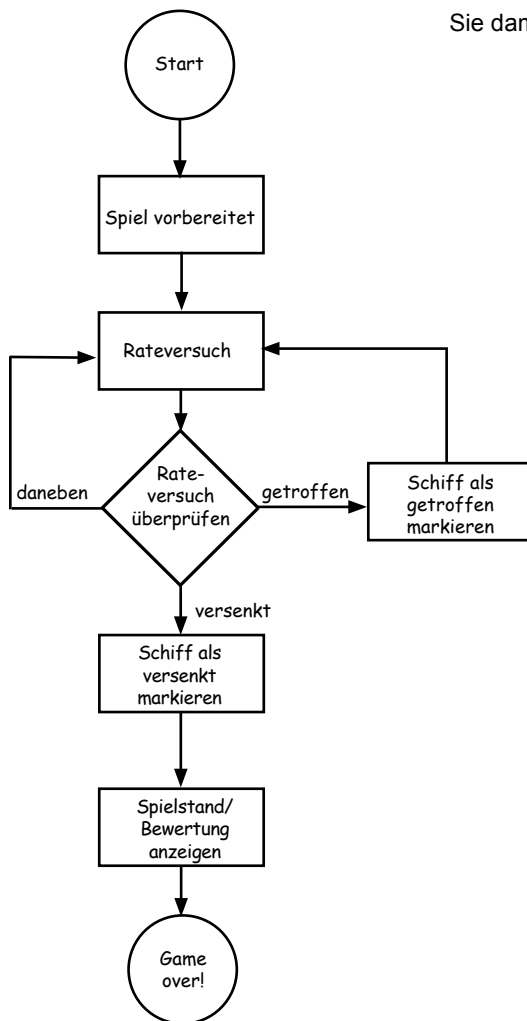


## 2

### Die nächsten Schritte

#### Mittlerweile kennen Sie schon Variablen, Typen, Ausdrücke und noch mehr.

Sie wissen bereits ein paar Dinge über JavaScript. Das heißt, Sie können schon **richtigen Code** schreiben. Code, der etwas Interessantes tut – Code, den andere Leute benutzen wollen. Jetzt fehlt Ihnen nur noch die **richtige Erfahrung** beim Schreiben von Code. Und das wollen wir hier und jetzt ändern. Wie? Indem wir kopfüber ins kalte Wasser springen und ein komplettes JavaScript-Spiel programmieren. Klingt ehrgeizig, aber wir lassen Sie nicht hängen. Auf geht's, lassen Sie uns direkt loslegen. Und wenn Sie damit eine neue Firma gründen wollen – nur zu! Der Code gehört Ihnen.



Schiffe versenken: Die Programmierung	44
Unser erster Versuch...	44
Der allgemeine Spielverlauf	45
Arbeiten wir uns durch den Pseudocode	47
Bevor Sie weiter machen sollten Sie den HTML-Code nicht vergessen!	49
Der Code für die einfache Version von »Schiffe versenken«	50
Und damit zur Logik des Spiels	51
Schritt eins: Die Schleife einrichten und Benutzereingaben auslesen	52
Wie prompt funktioniert	53
Die Benutzereingabe überprüfen	54
Und? Haben wir getroffen?	56
Der Code zum Finden von Treffern	57
Etwas Spielanalyse für den Benutzer	58
Und damit ist die Logik vollständig	60
Ein wenig Qualitätssicherung	61
Lassen Sie uns über Wortreichtum reden...	65
Das Schiffe-versenken-Spiel fertigstellen	66
Eine zufällige Position zuweisen	67
Das weltberühmte Rezept zur Erzeugung von Zufallszahlen	67
Etwas mehr QA	69
Herzlichen Glückwunsch zu Ihrem ersten richtigen JavaScript-Programm. Aus diesem Anlass ein paar Worte zum Code-Recycling.	71

# Einführung in Funktionen

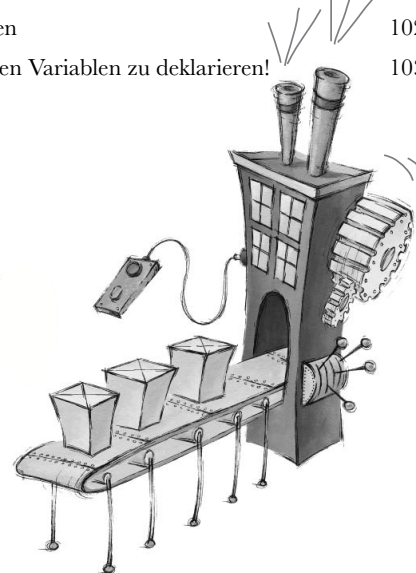
## 3

### Jetzt funk't's

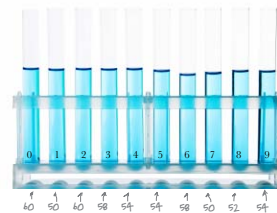
**Hier kommt Ihre erste Superkraft.** Inzwischen können Sie schon ein wenig programmieren. Daher wollen wir Ihnen jetzt das Konzept der **Funktionen** vorstellen. Funktionen geben Ihnen die Macht, Code zu schreiben, der unter verschiedensten Umständen **wiederverwendet** werden kann. Dieser Code ist deutlich besser **wartbar**, er kann **abstrahiert** und mit einem einfachen Namen versehen werden. Dadurch können Sie die Komplexität ignorieren und sich auf die wichtigen Dinge konzentrieren. Funktionen sind ein wichtiger Schritt auf Ihrem Weg vom Skriptler zum Programmierer und der Schlüssel zum Programmierstil von JavaScript. In diesem Kapitel zeigen wir Ihnen die Grundlagen: die Mechanik, die ganzen kleinen Details, die zum Verständnis von Funktionen wichtig sind. Danach werden wir Ihre Kenntnisse im Laufe dieses Buchs immer mehr erweitern und vertiefen. *Jetzt* wollen wir aber erst einmal solides Fundament schaffen.



Was stimmt denn mit dem Code nicht?	81
Haben wir übrigens schon über FUNKTIONEN gesprochen?	83
Schön, aber wie funktioniert das überhaupt?	84
Was kann einer Funktion übergeben werden?	89
JavaScript verwendet Werteparameter	92
Seltene Funktionen	94
Funktionen können auch Dinge zurück geben	95
Eine Funktion mit return-Anweisung nachvollziehen	96
Globale und lokale Variablen	99
Den Geltungsbereich globaler und lokaler Variablen verstehen	101
Das kurze Leben der Variablen	102
Vergessen Sie nicht, alle lokalen Variablen zu deklarieren!	103



## Etwas Ordnung in die Daten bringen



# 4

## Arrays

**JavaScript besteht nicht nur aus Zahlen, Strings und Booleschen Werten.** Bisher haben Sie in Ihrem JavaScript-Code nur primitive Datentypen – einfache Strings, Zahlen und Boolesche Werte wie »Fido«, 23 und true – benutzt. Damit lässt sich schon eine Menge anstellen, aber irgendwann müssen Sie mit **mehr Daten** arbeiten. Das könnten alle Artikel in einem Warenkorb sein oder alle Titel einer Playlist oder eine Liste von Sternen und ihre scheinbare Größe oder ein kompletter Produktkatalog. Dafür brauchen wir etwas mehr *Wumms*. Der Datentyp der Wahl für geordnete Daten dieser Art ist das JavaScript-**Array**. In diesem Kapitel zeigen wir Ihnen, wie Sie Ihre Daten in einem Array ablegen, es weitergeben und damit arbeiten können. Später in diesem Buch werden Sie weitere Möglichkeiten kennenlernen, Ihre **Daten zu strukturieren**; jetzt wollen wir uns aber erst mal mit Arrays befassen.

Können Sie Bubbles-R-U's helfen?	126
Mehrere Werte in JavaScript darstellen	127
Wie Arrays funktionieren	128
Wie groß ist das Array eigentlich?	130
Der Phras-O-Mat	132
Inzwischen bei Bubbles-R-U's...	135
Über ein Array iterieren	138
Moment mal! Es gibt noch eine bessere Methode über ein Array zu iterieren	140
Wir müssen mal wieder über Ihren Wortschatz reden	146
Eine Neufassung der for-Schleife mit dem Postinkrement-Operator	147
Schnelle Probefahrt	147
Ein neues Array anlegen (und Elemente hinzufügen)	151
Und die Gewinner sind ...	155
Eine schnelle Codeüberprüfung ...	157
Eine printAndGetHighScore-Funktion schreiben	158
Den Code mit der printAndGetHighScore-Funktion refaktorisieren	159
Alles zusammengenommen ...	161



## Objekte verstehen

# 5

### Ein Ausflug nach Objectville

**Bis jetzt haben Sie in Ihrem Code nur »primitive« Datentypen und Arrays benutzt.** Die Programmierung sind Sie bisher eher **prozedural** angegangen um mit einfachen Anweisungen, Bedingungen, for-/while-Schleifen und Funktionen, die nicht gerade objektorientiert waren. Eigentlich waren sie *überhaupt nicht* objektorientiert. Wir haben zwar hier und da, ohne es zu wissen, ein paar Objekte benutzt, aber eigene Objekte haben Sie noch nicht geschrieben. Jetzt ist es Zeit, diese langweilige prozedurale Stadt zu verlassen und selbst ein paar **Objekte** zu erstellen. In diesem Kapitel werden Sie lernen, wie Sie sich das Leben deutlich erleichtern können, zumindest was die **Programmierung** angeht. (Im selben Buch auch noch Ihrem modischen Geschmack auf die Sprünge zu helfen, ist dann doch ein bisschen viel verlangt.) Eine kleine Warnung: Haben Sie die Objekte einmal kennengelernt, wollen Sie nicht wieder zurück. Schicken Sie uns eine Postkarte, wenn Sie angekommen sind.



Hat gerade jemand »Objekte« gesagt?!	174
Ein paar Gedanken zu Eigenschaften...	175
Objekte anlegen	177
Was bedeutet eigentlich »objektorientiert«?	180
Wie Eigenschaften funktionieren	181
Wie kann eine Variable ein Objekt enthalten?	186
Primitive Datentypen und Objekte im Vergleich	187
Noch mehr Dinge mit Objekten...	188
Die Vorauswahl Schritt für Schritt	190
Noch ein paar Worte zur Übergabe von Objekten an Funktionen	192
Objekte mit Verhalten versehen	198
Die drive-Methode verbessern	199
Warum weiß die drive-Methode nicht von der Eigenschaft started?	202
Wie <b>this</b> funktioniert	204
Wie Verhalten einen Zustand beeinflussen kann	210
Und jetzt soll der Zustand das Verhalten beeinflussen	211
Herzlichen Glückwunsch zu Ihren ersten Objekten!	213
Objekte sind überall (und sie machen Ihnen das Leben leichter)!	214

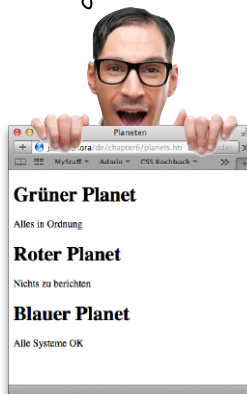
# Interaktion mit der Webseite

## 6

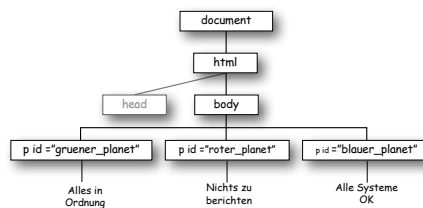
### Das DOM kennenlernen

Ihre JavaScript-Fähigkeiten haben große Fortschritte gemacht. Mittlerweile haben Sie sich von einem Anfänger zu einem Skriptler, ja sogar zu einem echten **Programmierer** entwickelt. Es fehlt aber noch etwas. Damit sich Ihre JavaScript-Kenntnisse wirklich entfalten können, brauchen Sie eine Möglichkeit, mit der Webseite, die Ihren Code enthält, zu interagieren. Nur so ist es möglich, **dynamische** Seiten zu schreiben, die reagieren, antworten und sich nach dem Laden automatisch aktualisieren. Wie funktioniert diese Interaktion? Über das **DOM**, das **Document Object Model**. Wir werden uns das DOM in diesem Kapitel Stück für Stück vornehmen und sehen, wie es zusammen mit JavaScript dazu bewegt werden kann, Ihrer Seite ein paar neue Tricks beizubringen.

Hallo, hier ist der Browser.  
Ich lese die Seite und  
erstelle daraus ein DOM.



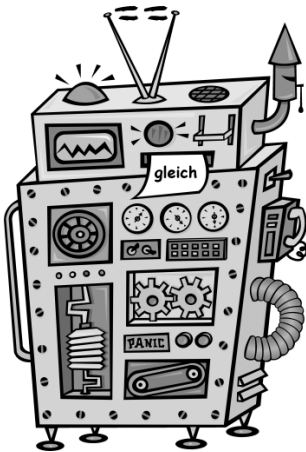
Im letzten Kapitel gab es eine »Knacken Sie den Geheimcode«-Aufgabe	230
Und was macht dieser Code genau?	231
Wie JavaScript tatsächlich mit Ihrer Seite interagiert	233
Ein Rezept für Ihr eigenes DOM	234
Ihr erster Eindruck vom DOM	235
Mit getElementById auf ein Element zugreifen	240
Was genau gibt mir das DOM zurück?	241
Finden Sie Ihr inneres HTML	242
Was passiert wenn Sie das DOM verändern?	244
Ein kleiner Probeflug	247
Denken Sie nicht mal daran, meinen Code vor dem vollständigen Laden der Seite auszuführen!	249
Sie sagen »Event-Handler«, ich sage »Callback«	250
Ein Attribut per setAttribute bearbeiten	255
Mehr Spaß mit Attributen	256
Vergessen Sie nicht, dass getElementById auch null zurückgeben kann!	256
Und wofür ist das DOM sonst noch gut?	258



# Typen, Gleichheit, Umwandlung und der ganze Rest

## 7 Seriöse Typen

**Jetzt wollen wir einmal ernsthaft über Typen reden.** Eine der guten Seiten von JavaScript ist, dass Sie auch ohne viel Detailwissen schon sehr weit kommen können. Aber um **die Sprache wirklich zu meistern**, die Gehaltserhöhung zu bekommen und es im Leben richtig zu etwas bringen, müssen Sie sich extrem gut mit **Typen** auskennen. Wissen Sie noch, was wir am Anfang über JavaScript gesagt haben? Dass es keine auf dem Silbertablett servierte, durch akademische Kreuzgutachten überprüfte Sprache ist? Das stimmt wohl, aber das akademische Leben hat Steve Jobs und Bill Gates nicht von ihrem Erfolg abgehalten, und JavaScript hält es auch nicht ab. Das bedeutet, dass JavaScript ... nun ja ... kein besonders durchdachtes Typensystem besitzt und wir noch ein paar **Eigenheiten** bemerken werden. Aber machen Sie sich keine Sorgen! In diesem Kapitel werden wir die Sache in den Griff bekommen, und Sie werden schnell in der Lage sein, den peinlichen Momenten, die Ihnen mit Typen begegnen können, aus dem Weg zu gehen.



Die Wahrheit ist irgendwo da draußen ...	266
Achtung, Sie könnten undefined begegnen wenn Sie es nicht erwarten ...	268
Die Verwendung von null	271
Mit NaN arbeiten	273
Es wird noch merkwürdiger	273
Wir müssen Ihnen etwas gestehen	275
Den Gleichheitsoperator (auch bekannt als ==) verstehen	276
Wie der Gleichheitsoperator seinen Operanden konvertiert	277
Die Gleichheit strikter verstehen	280
Noch mehr Typumwandlungen ...	286
Testen, ob zwei Objekte gleich sind	289
Die truthy ist irgendwo da draußen	291
Was JavaScript als falsey ansieht	292
Das geheime Leben der Strings	294
Strings, die gleichzeitig primitive Datentypen und Objekte sind	295
Eine Fünf-Minuten-Tour der String-Methoden (und -Eigenschaften)	297
Krieg der Stühle	301

## Die Einzelteile zusammenfügen

# 8

## Eine App schreiben

**Greifen Sie sich Ihren Werkzeugkasten.** Damit meinen wir den Werkzeugkasten mit all Ihren neuen Programmierkenntnissen, dem Wissen über das DOM und sogar mit etwas HTML und CSS. In diesem Kapitel bringen Sie alles zusammen und erstellen Ihre erste richtige **Webapplikation** – keine **albernen Kinderspiele** mit nur einem Schiff, das auf einer Zeile versteckt ist. Jetzt bauen wir **das komplette Spiel**: ein schönes Spielfeld mit mehreren Schiffen und Benutzereingaben direkt auf der Webseite. Die nötige Seitenstruktur erstellen wir mit HTML, visuelle Stile schreiben wir in CSS, und mit JavaScript programmieren wir das Verhalten für das Spiel. Machen Sie sich bereit: Dies ist ein richtiges Entwicklungskapitel, in dem wir ernsthaften Code schreiben.



A							
B							
C	Ship1						
D			Ship2				
E							
F							
G				Ship3	Hit		
	0	1	2	3	4	5	6

Diesmal wollen wir ein ECHTES »Schiffe-versenken«-Spiel erstellen	318
Ein Schritt zurück ... zu HTML und CSS	319
Die HTML-Seite erstellen: Ein Überblick	320
Etwas Stil hinzufügen	324
Die Klassen hit und miss verwenden	327
Das Spiel entwickeln	329
Den View implementieren	331
Die Funktionsweise von showMessage	331
Die Funktionsweise von displayHit und displayMiss	333
Das Model	336
Wie die Schiffe intern dargestellt werden	338
Das Model-Objekt implementieren	341
Nachdenken über die fire-Methode	342
Den Controller implementieren	349
Den Rateversuch den Spielers verarbeiten	350
Den Code planen ...	351
Die parseGuess-Methode implementieren	352
Rateversuche zählen und feuern	355
Ein Event-Handler für den Feuer!-Button	359
Die Eingabe an den Controller übergeben	360
Die Schiffe richtig platzieren	364
Die generateShip-Methode erstellen	365
Die Startposition für ein neues Schiff erzeugen	366
Die generateShip-Methode fertigstellen	367

# Asynchron programmieren

## 9

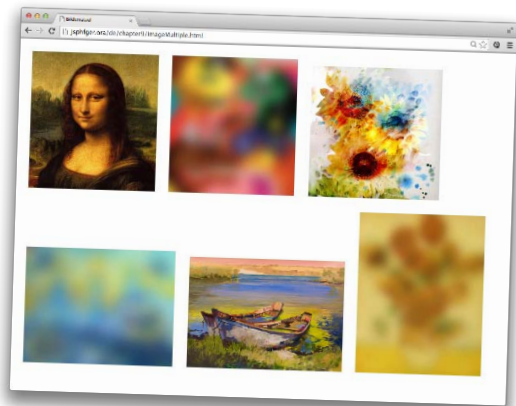
### Events be-handeln

**Nach diesem Kapitel werden Sie merken, dass Sie nicht mehr in Kansas sind.** Bis jetzt haben Sie Code geschrieben, der typischerweise von oben nach unten ausgeführt wird. Sicher, Ihr Code ist vermutlich etwas komplexer und enthält ein paar Funktionen, Objekte und Methoden, aber irgendwann wird auch dieser Code ausgeführt. Es tut uns wirklich leid, wenn wir Ihnen das erst so spät im Buch sagen, aber **so schreibt man eigentlich keinen JavaScript-Code**. Stattdessen wird

JavaScript meistens geschrieben, um auf **Ereignisse (Events) zu reagieren**. Was für Events? Alle möglichen! Das kann ein Mausclick auf Ihrer Seite, ankommende Daten aus dem Netzwerk, ein im Browser ablaufender Timer oder eine Änderung im DOM sein, um nur ein paar zu nennen. Eigentlich passieren **ständig** irgendwelche Events im Hintergrund des Browsers. In diesem Kapitel werden wir unsere Art, in JavaScript zu programmieren, neu überdenken und lernen, warum und wie Sie Code schreiben können, der auf Events reagiert.



Was sind Events?	383
Was ist ein Event-Handler?	384
Den ersten Event-Handler erstellen	385
Probefahrt für Ihr Event	386
Events verstehen... indem wir ein Spiel programmieren	388
Das Spiel implementieren	389
Probefahrt	390
Noch mehr Bilder	394
Jetzt müssen wir den onclick-Eigenschaften jedes Bilds den gleichen Handler zuweisen	395
Den gleichen Handler für alle Bilder benutzen	396
Die Funktionsweise des Event-Objekts	399
Das Event-Objekt benutzen	401
Probefahrt für das Event-Objekt und die target-Eigenschaft	402
Events und ihre Warteschlange	404
Noch mehr Events	407
Wie setTimeout funktioniert	408
Das Bilderrätsel fertigstellen	412
Probefahrt für den Timer	413







## Funktionen erster Klasse

# 10

### Befreite Funktionen

**Erst Funktionen machen Sie zum Rockstar.** Jedes Handwerk, jede Kunst besitzt ein Schlüsselprinzip, das die mittelmäßigen Spieler von den großen Virtuosen unterscheidet. Bei JavaScript ist es das Verständnis von **Funktionen**. Funktionen sind ein wesentlicher Bestandteil von JavaScript und bieten viele Techniken für das **Design und die Organisation** von Code. Die Basis hierfür ist ein solides Wissen um die Verwendung von Funktionen. Funktionen auf diesem Niveau zu lernen, ist interessant und kann bewusstseinsverändernd wirken, also machen Sie sich bereit. Es ist ein bisschen, als würde Ihnen Willy Wonka eine Führung durch die Schokoladenfabrik geben. Beim Lernen von JavaScript-Funktionen werden Ihnen einige ziemlich wilde, verrückte, aber auch wundervolle Dinge begegnen.

Das mysteriöse Doppelleben des Schlüsselworts function	430
Funktionsdeklarationen im Vergleich mit Funktionsausdrücken	431
Die Funktionsdeklaration parsen	432
Was kommt jetzt? Der Browser führt den Code aus	433
Weiter geht's ... Die Bedingung	434
Wie Funktionen auch Werte sein können	439
Haben wir schon gesagt, dass Funktionen in JavaScript als erstklassig gelten?	442
Erster Klasse fliegen	443
Den Code schreiben, um Passagiere zu verarbeiten und zu überprüfen	444
Über die Passagiere iterieren	446
Eine Funktion an eine andere übergeben	447
Funktionen aus Funktionen zurückgeben	450
Den Code für die Getränkebestellung schreiben	451
Der Flugbegleiter-Code für die Getränkebestellung: ein neuer Ansatz	452
Bestellungen mit Funktionen erster Klasse	454
Webville Cola	457
Wie die sort-Methode für Array funktioniert	459
Die Einzelteile zusammensetzen	460
Probefahrt für die Sortierung	462



## 11

## Anonyme Funktionen, Geltungsbereiche und Closures

**Seriöse Funktionen**

**Sie haben sich umfassend mit Funktionen befasst, aber es gibt noch mehr zu lernen.** In diesem Kapitel gehen wir sogar noch weiter, jetzt kommt echter Hardcore.

Wir zeigen Ihnen, wie man **wirklich** mit Funktionen **umgeht**. Dies wird kein superlanges Kapitel, aber es wird intensiv, und am Ende wird Ihr JavaScript ausdrucksstärker sein als je zuvor. Sie werden auch in der Lage sein, den Code Ihrer Mitarbeiter zu verstehen und eine Open Source-JavaScript-Bibliothek zu durchschauen. In diesem Kapitel behandeln wir nämlich einige häufige Code-Idiome und Konventionen, die bei Funktionen immer wieder vorkommen. Und wenn Sie noch nie von **anonymen Funktionen** oder **Closures** gehört haben, dann sind Sie hier mehr als richtig.



Sehen Sie Funktionen mal andersherum	476
Anonyme Funktionen benutzen	477
Wir müssen nochmal über Ihren Wortschatz reden	479
Wann wird eine Funktion definiert? Kommt drauf an ...	483
Was ist da gerade passiert? Warum war fly nicht definiert?	484
Funktionen verschachteln	485
Wie sich die Verschachtelung auf den Geltungsbereich auswirkt	486
Ein kleiner Rückblick zum lexikalischen Geltungsbereich	488
Wo der lexikalische Geltungsbereich richtig interessant wird	489
Wiedersehen mit Funktionen	491
Wiedersehen mit Funktionsaufrufen	492
Was zum Henker ist eine Closure?	495
Eine Funktion »schließen«	496
Mit Closures einen magischen Zähler implementieren	498
Ein Blick hinter die Kulissen	499
Closures durch Übergabe einer Funktionsreferenz als Argument erstellen	501
Die Closure enthält keine Kopie, sondern die tatsächliche Umgebung	502
Eine Closure per Event-Handler erstellen	503
Wie die Klick mich!-Closure funktioniert	506

## Fortgeschrittene Objektkonstruktion

# 12

### Objekte erstellen

**Bisher haben wir unsere Objekte von Hand erstellt.** Für jedes Objekt haben wir ein **Objektliteral** verwendet, um die einzelnen Eigenschaften anzugeben. Das ist in kleinem Rahmen auch kein Problem. Für ernsthaften Code brauchen wir aber etwas Besseres. Und da kommen **Objektconstructoren** ins Spiel. Mit Constructoren können wir die Objekte wesentlich einfacher erstellen, die außerdem alle nach der gleichen **Designschablone** aufgebaut sind. Das heißt, durch die Verwendung von Constructoren können wir sicherstellen, dass jedes Objekt die gleichen Eigenschaften besitzt und die gleichen Methoden enthält. Außerdem ist der mit Constructoren erstellte Objektcode wesentlich **kürzer** und deutlich weniger fehleranfällig, besonders wenn viele Objekte erstellt werden müssen. Legen wir also einfach mal los! In kürzester Zeit werden Sie so gut »konstruktorisch« sprechen, als seien Sie in Objectville groß geworden.



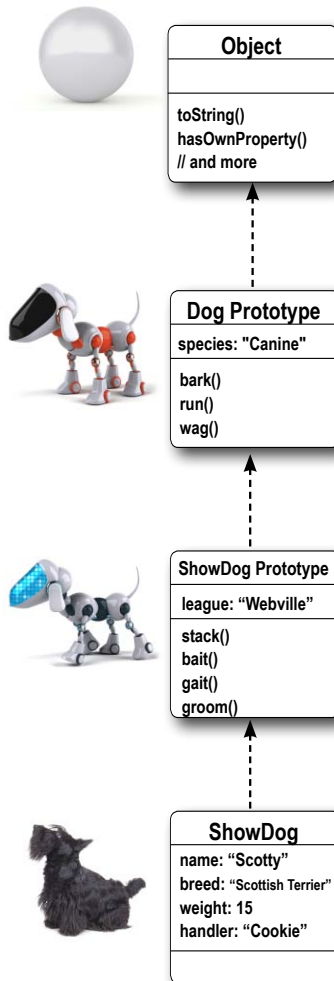
Objekte mit Objektliteralen erstellen	522
Konventionen für Objekte verwenden	523
Einführung in Objektconstructoren	525
Einen Konstruktor erstellen	526
Einen Konstruktor benutzen	527
Wie Constructoren funktionieren	528
Constructoren können auch Methoden enthalten	530
Zeit für die Produktion!	536
Jetzt wollen wir ein paar neue Autos Probe fahren	538
Verlassen Sie sich nicht einfach auf Objektliterale	539
Die Argumente als Objektliteral neu verdrahten	540
Den Car-Konstruktor umbauen	541
Objektinstanzen verstehen	543
Auch per Konstruktor erstellte Objekte können eigene unabhängige Eigenschaften haben	546
Constructoren im wahren Leben	548
Der Array-Objektconstructor	549
Noch mehr Spaß mit eingebauten Objekten	551

# 13

## Prototypen verwenden

### Extrastarke Objekte

**Zu lernen, wie man Objekte erstellt, war erst der Anfang.** Jetzt ist es Zeit, Ihren Objekten ein paar Muskeln wachsen zu lassen. Wir brauchen neue Möglichkeiten, **Beziehungen** zwischen den Objekten herzustellen und **Code** zwischen ihnen **auszutauschen**. Und wir benötigen die Fähigkeit, bestehende Objekte zu erweitern und aufzuwerten. Anders gesagt: Wir brauchen mehr Werkzeuge. In diesem Kapitel werden Sie sehen, dass JavaScript ein sehr mächtiges **Objektmodell** besitzt, aber auch, dass es sich von den üblichen objektorientierten Programmiersprachen unterscheidet. Anstelle des typischen klassenbasierten Objektmodells verwendet JavaScript ein mächtigeres **Prototypenmodell**, bei dem die Objekte voneinander erben oder das Verhalten anderer Objekte erweitern können. Wofür das gut sein soll, werden Sie gleich sehen. Auf geht's ...



Ach so, bevor wir anfangen, haben wir noch eine bessere Möglichkeit unsere Objekte in einem Diagramm darzustellen	565
Wiedersehen mit Objektkonstruktoren	566
Ist die Duplizierung der Methode wirklich ein Problem?	568
Was sind Prototypen?	569
Von einem Prototyp erben	570
Wie die Vererbung funktioniert	571
Prototypen überschreiben	573
Den Prototyp einrichten	576
Prototypen sind dynamisch	582
Eine interessantere Implementierung der sit-Methode	584
Noch einmal: die Funktionsweise der sitting-Eigenschaft	585
Die Herangehensweise für Ausstellungshunde	589
Eine Prototypenkette einrichten	591
Die Funktionsweise der Vererbung in der Prototypenkette	592
Den Prototyp für Ausstellungshunde erstellen	594
Eine Ausstellungshunde-Instanz erzeugen	598
Eine letzte Reinigung der Ausstellungshunde	602
Dog.call Schritt für Schritt	604
Die Kette endet nicht bei Dog	607
Vererbung zu Ihrem Vorteil nutzen ... durch Überschreiben von eingebautem Verhalten	608
Vererbung zum eigenen Vorteil nutzen ... durch Erweiterung eines JavaScript-eigenen Objekts	610
Die große Einheitstheorie von Allem	612
Besser leben mit Objekten	612
Die Einzelteile zusammenfügen	613
Was kommt als Nächstes?	613

## 14

## Anhang: Was übrig bleibt

**Die Top Ten der Themen,  
die wir nicht behandelt haben**

**Wir sind ganz schön weit gekommen, und Sie haben das Buch fast durch.** Wir werden Sie vermissen, aber bevor wir Sie gehen lassen, wollen wir Sie nicht ohne etwas zusätzliche Vorbereitung losschicken. Leider können wir nicht alles, was Sie noch wissen müssen, in diesem relativ kurzen Kapitel unterbringen. Ursprünglich hatten wir tatsächlich sämtliches notwendiges Wissen über JavaScript-Programmierung (das nicht bereits in den anderen Kapiteln behandelt wurde) hier untergebracht, indem wir die Schriftgröße ungefähr auf 0,00004 Punkt reduziert haben. Es passte alles rein, aber niemand konnte es lesen. Also haben wir das meiste wieder rausgeschmissen und die besten Sachen für den Top-Ten-Anhang behalten.

Das hier ist tatsächlich das Ende des Buchs. Das heißt, bis auf den Index natürlich (den Sie unbedingt lesen müssen!).



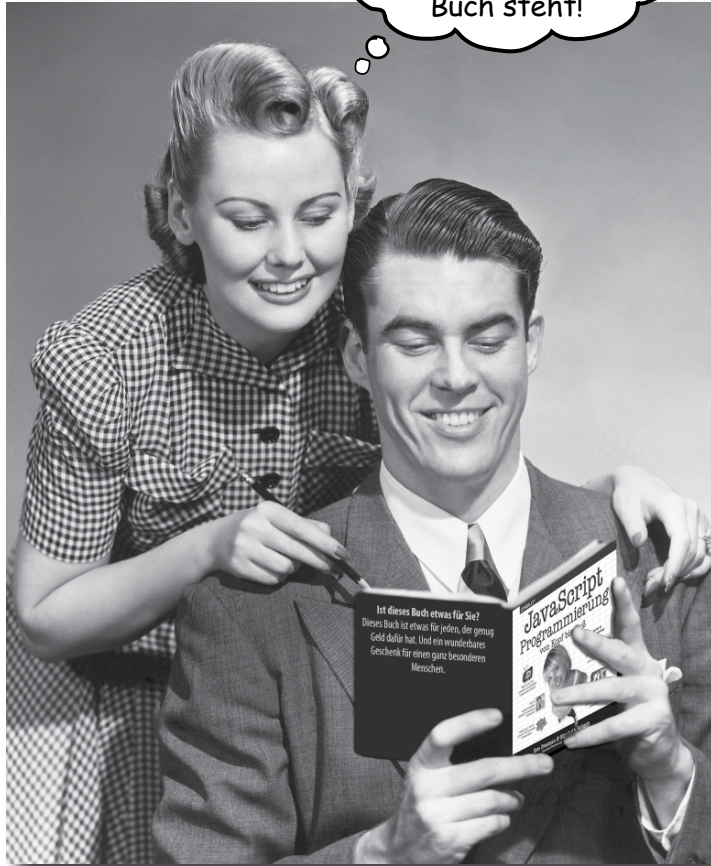
1. jQuery	624
2. Mehr mit dem DOM anstellen	626
3. Das window-Objekt	627
4. arguments	628
5. Mit Ausnahmen umgehen	629
6. Event-Handler mit addEventListener hinzufügen	630
7. Reguläre Ausdrücke	632
8. Rekursion	634
9. JSON	636
10. Serverseitiges JavaScript	637



Wie man dieses Buch benutzt

# Einführung

Ich kann einfach nicht fassen, dass **so etwas** in einem JavaScript-Buch steht!



In diesem Abschnitt beantworten wir die brennende Frage:  
»Und warum STEHT so etwas in einem JavaScript-Buch?«

## Für wen ist dieses Buch?

Wenn Sie alle folgenden Fragen mit »Ja« beantworten können:

- ① Haben Sie einen Computer mit einem **modernen Webbrowser** und einen **Texteditor**?
- ② Möchten Sie **lernen, verstehen und behalten**, unter Verwendung der besten Techniken und neuesten Standards in **JavaScript zu programmieren**?
- ③ Ziehen Sie eine **anregende Unterhaltung beim Abendessen** einer **trockenen, langweiligen Vorlesung vor**?

... dann ist dieses Buch etwas für Sie.

[Hinweis aus der Marketing-Abteilung: Dieses Buch ist etwas für jeden, der eine Kreditkarte besitzt.]

← Als modern betrachten wir eine aktualisierte Version von Safari, Chrome, Firefox oder IE in der Version 9 oder neuer.

## Wer sollte eher die Finger von diesem Buch lassen?

Wenn Sie eine der folgenden Fragen mit »Ja« beantworten müssen:

- ① **Ist die Entwicklung von Webseiten völliges Neuland für Sie?**  
Sind Ihnen HTML und CSS vollkommen fremd? Falls ja, sollten Sie wahrscheinlich mit *HTML und CSS – Von Kopf bis Fuß* anfangen, um zu verstehen, wie man Webseiten zusammenbaut, bevor Sie sich mit JavaScript beschäftigen.
- ② Sind Sie bereits Webentwickler und suchen Sie nach einem **Referenzbuch**?
- ③ Haben Sie **Angst, etwas Neues auszuprobieren**? Ist Ihnen eine Wurzelbehandlung lieber, als Streifen mit Karos zu tragen? Glauben Sie, dass ein technisches Buch nicht seriös sein kann, in dem JavaScript-Objekte vermenschlicht werden?

... dann ist dieses Buch nichts für Sie.





## Wir wissen was Sie gerade denken.

- » Kann das wirklich ein seriöses Buch sein? «
- »Was ist mit all den Abbildungen?«
- » Kann ich das auf diese Art wirklich lernen?«

## Und wir wissen, was Ihr Gehirn gerade denkt.

Ihr Gehirn lechzt nach Neuem. Es ist ständig dabei, Ihre Umgebung abzusuchen, und es *wartet* auf etwas Ungewöhnliches. So ist es nun einmal gebaut, und es hilft Ihnen zu überleben.

Heutzutage ist es weniger wahrscheinlich, dass Sie von einem Tiger verputzt werden. Aber Ihr Gehirn hält immer noch Ausschau. Man weiß ja nie.

Also, was macht Ihr Gehirn mit all den gewöhnlichen, normalen Routinesachen, denen Sie begegnen? Es tut alles in seiner Macht stehende, damit es dadurch nicht bei seiner *eigentlichen* Arbeit gestört wird: Dinge zu erfassen, die wirklich *wichtig* sind. Es gibt sich nicht damit ab, die langweiligen Sachen zu speichern, sondern lässt diese gar nicht erst durch den »Dies-ist-offensichtlich-nicht-wichtig«-Filter.

Woher *weiß* Ihr Gehirn denn, was wichtig ist? Nehmen Sie an, Sie machen einen Tagesausflug und ein Tiger springt vor Ihnen aus dem Gebüsch: Was passiert dabei in Ihrem Kopf und Ihrem Körper?

Neuronen feuern. Gefühle werden angekurbelt. *Chemische Substanzen durchfluten Sie.*

Und so weiß Ihr Gehirn:

### **Dies muss wichtig sein! Vergiss es nicht!**

Aber nun stellen Sie sich vor, Sie sind zu Hause oder in einer Bibliothek. In einer sicheren, warmen, tigerfreien Zone. Sie lernen. Bereiten sich auf eine Prüfung vor. Oder Sie versuchen, irgendein schwieriges Thema zu lernen, von dem Ihr Chef glaubt, Sie bräuchten dafür eine Woche oder höchstens zehn Tage.

Da ist nur ein Problem: Ihr Gehirn versucht Ihnen einen großen Gefallen zu tun. Es versucht dafür zu sorgen, dass diese *offensichtlich* unwichtigen Inhalte nicht knappe Ressourcen verstopfen. Ressourcen, die besser dafür verwendet würden, die wirklich *wichtigen* Dinge zu speichern. Wie Tiger. Wie die Gefahren des Feuers. Oder dass Sie nie wieder in Shorts snowboarden sollten.

Und es gibt keine einfache Möglichkeit, Ihrem Gehirn zu sagen: »Hey, Gehirn, vielen Dank, aber egal, wie langweilig dieses Buch auch ist und wie klein der Ausschlag auf meiner emotionalen Richterskala gerade ist, ich *will* wirklich, dass du diesen Kram behältst.«

Ihr Gehirn denkt DAS  
HIER ist wichtig.



Na toll. Nur noch  
672 trockene und  
langweilige Seiten.

Ihr Gehirn denkt  
DAS HIER zu  
speichern lohnt  
sich nicht.



## Wir stellen uns unseren Leser als einen aktiv Lernenden vor.

Also, was ist nötig, damit Sie etwas *lernen*? Erst einmal müssen Sie es *aufnehmen* und dann dafür sorgen, dass Sie es nicht wieder *vergessen*. Es geht nicht darum, Fakten in Ihren Kopf zu schieben. Nach den neuesten Forschungsergebnissen der Kognitionswissenschaft, der Neurobiologie und der Lernpsychologie gehört zum *Lernen* viel mehr als nur Text auf einer Seite. Wir wissen, was Ihr Gehirn anmacht.

### Einige der Lernprinzipien dieser Buchreihe:

**Bilder einsetzen.** An Bilder kann man sich viel besser erinnern als an Worte allein und lernt so viel effektiver (bis zu 89% Verbesserung bei Abrufbarkeits- und Lernttransferstudien). Außerdem werden die Dinge dadurch verständlicher. **Text in oder neben die Grafiken setzen**, auf die sie sich beziehen, anstatt darunter oder auf eine andere Seite. Die Leser werden auf den Bildinhalt bezogene Probleme dann mit *doppelt* so hoher Wahrscheinlichkeit lösen können.

Browser stellen Anfragen für HTML-Seiten und andere Ressourcen, z. B. Bilder



Es ist wirklich ätzend, wenn das <body>-Element vergessen wird



Im <head>-Element platzieren Sie Infos über Ihre Seite.

**Verwenden Sie einen gesprächsorientierten Stil mit persönlicher Ansprache.** Nach neueren Untersuchungen haben Studenten nach dem Lernen bei Tests bis zu 40% besser abgeschnitten, wenn der Inhalt den Leser direkt in der ersten Person und im lockeren Stil angesprochen hat statt in einem formalen Ton. Halten Sie keinen Vortrag, sondern erzählen Sie Geschichten. Benutzen Sie eine zwanglose Sprache. Nehmen Sie sich selbst nicht zu ernst. Würden Sie einer anregenden Unterhaltung beim Abendessen mehr Aufmerksamkeit schenken oder einem Vortrag?

**Bringen Sie den Lernenden dazu, intensiver nachzudenken.** Mit anderen Worten: Falls Sie nicht aktiv Ihre Neuronen strapazieren, passiert in Ihrem Gehirn nicht viel. Ein Leser muss motiviert, begeistert und neugierig sein und angeregt werden, Probleme zu lösen, Schlüsse zu ziehen und sich neues Wissen anzueignen. Und dafür brauchen Sie Herausforderungen, Übungen, zum Nachdenken anregende Fragen und Tätigkeiten, die beide Seiten des Gehirns und mehrere Sinne einbeziehen.

Ist es sinnvoll, eine Klasse »badewanne« für meinen Style anzulegen oder sollte ich besser das gesamte Badezimmer stylen?



**Ziehen Sie die Aufmerksamkeit des Lesers auf sich – und behalten Sie sie.** Wir alle haben schon Erfahrungen dieser Art gemacht: »Ich will das wirklich lernen, aber ich kann einfach nicht über Seite 1 hinaus wach bleiben.« Ihr Gehirn passt auf, wenn Dinge ungewöhnlich, interessant, merkwürdig, auffällig, unerwartet sind. Ein neues, schwieriges, technisches Thema zu lernen muss nicht langweilig sein. Wenn es das nicht ist, lernt Ihr Gehirn viel schneller.

**Sprechen Sie Gefühle an.** Wir wissen, dass Ihre Fähigkeit, sich an etwas zu erinnern, wesentlich von dessen emotionalem Gehalt abhängt. Sie erinnern sich an das, was Sie *bewegt*. Sie erinnern sich, wenn Sie etwas *fühlen*. Nein, wir erzählen keine herzerreißenden Geschichten über einen Jungen und seinen Hund. Was wir erzählen, ruft Überraschungs-, Neugier-, Spaß- und Was-soll-das?-Emotionen hervor und dieses Hochgefühl, das Sie beim Lösen eines Puzzles empfinden oder wenn Sie etwas lernen, was alle anderen schwierig finden. Oder wenn Sie merken, dass Sie etwas können, was dieser »Ich-bin-ein-besserer-Techniker-als-du«-Typ aus der Technikabteilung *nicht kann*.



## Metakognition: Nachdenken übers Denken

Wenn Sie wirklich lernen möchten, und zwar schneller und nachhaltiger, dann schenken Sie Ihrer Aufmerksamkeit Aufmerksamkeit. Denken Sie darüber nach, wie Sie denken. Lernen Sie, wie Sie lernen.

Die Meisten von uns haben in ihrer Jugend keine Kurse in Metakognition oder Lerntheorie gehabt. Es wurde von uns *erwartet*, dass wir lernen, aber nur selten wurde uns auch *beigebracht*, wie man lernt.

Wir nehmen aber an, dass Sie wirklich lernen möchten, wie Sie Webseiten erstellen. Und wahrscheinlich möchten Sie nicht viel Zeit aufwenden. Und Sie wollen sich an das *erinnern*, was Sie lesen, und es anwenden können. Und deshalb müssen Sie es *verstehen*. Wenn Sie so viel wie möglich von diesem Buch profitieren wollen oder von irgendeinem anderen Buch oder einer anderen Lernerfahrung, übernehmen Sie Verantwortung für Ihr Gehirn. Ihr Gehirn im Zusammenhang mit diesem Lernstoff.

Der Trick besteht darin, Ihr Gehirn dazu zu bringen, neuen Lernstoff als etwas wirklich Wichtiges anzusehen. Als entscheidend für Ihr Wohlbefinden. So wichtig wie ein Tiger. Andernfalls stecken Sie in einem dauernden Kampf, in dem Ihr Gehirn sein Bestes gibt, um die neuen Inhalte davon abzuhalten, hängen zu bleiben.

### Wie bringen Sie also Ihr Gehirn dazu, HTML & CSS für so wichtig zu halten wie einen Tiger?

Da gibt es den langsamen, ermüdenden Weg oder den schnelleren, effektiveren Weg. Der langsame Weg geht über bloße Wiederholung. Natürlich ist Ihnen klar, dass Sie lernen und sich sogar an die langweiligsten Themen erinnern *können*, wenn Sie sich die gleiche Sache immer wieder einhämmern. Wenn Sie nur oft genug wiederholen, sagt Ihr Gehirn: »Er hat zwar nicht das *Gefühl*, dass das wichtig ist, aber er sieht sich dieselbe Sache *immer und immer wieder* an – dann muss sie wohl wichtig sein.«

Der schnellere Weg besteht darin, **alles zu tun, was die Gehirnaktivität erhöht**, vor allem verschiedene Arten von Gehirnaktivität. Eine wichtige Rolle dabei spielen die auf der vorhergehenden Seite erwähnten Dinge – alles Dinge, die nachweislich helfen, dass Ihr Gehirn *für* Sie arbeitet. So hat sich z.B. in Untersuchungen gezeigt: Wenn Wörter *in* den Abbildungen stehen, die sie beschreiben (und nicht irgendwo anders auf der Seite, z.B. in einer Bildunterschrift oder im Text), versucht Ihr Gehirn herauszufinden, wie die Wörter und das Bild zusammenhängen, und dadurch feuern mehr Neuronen. Und je mehr Neuronen feuern, umso größer ist die Chance, dass Ihr Gehirn mitbekommt: Bei dieser Sache lohnt es sich aufzupassen und vielleicht auch, sich daran zu erinnern.

Ein lockerer Sprachstil hilft, denn Menschen tendieren zu höherer Aufmerksamkeit, wenn ihnen bewusst ist, dass sie ein Gespräch führen – man erwartet dann ja von ihnen, dass sie dem Gespräch folgen und sich beteiligen. Das Erstaunliche daran ist: Es ist Ihrem Gehirn ziemlich egal, dass die »Unterhaltung« zwischen Ihnen und einem Buch stattfindet! Wenn der Schreibstil dagegen formal und trocken ist, hat Ihr Gehirn den gleichen Eindruck wie bei einem Vortrag, bei dem in einem Raum passive Zuhörer sitzen. Nicht nötig, wach zu bleiben.

Aber Abbildungen und ein lockerer Sprachstil sind erst der Anfang.



## Das haben WIR getan:

Wir haben **Bilder** verwendet, weil Ihr Gehirn auf visuelle Eindrücke eingestellt ist, nicht auf Text. Soweit es Ihr Gehirn betrifft, sagt ein Bild *wirklich* mehr als 1.024 Worte. Und dort, wo Text und Abbildungen zusammenwirken, haben wir den Text *in* die Bilder eingebettet, denn Ihr Gehirn arbeitet besser, wenn der Text *innerhalb* der Sache steht, auf die er sich bezieht, und nicht in einer Bildunterschrift oder irgendwo vergraben im Text.

Wir haben **Redundanz** eingesetzt, d.h. dasselbe auf *unterschiedliche* Art und mit verschiedenen Medientypen ausgedrückt, damit Sie es über *mehrere Sinne* aufnehmen. Das erhöht die Chance, dass die Inhalte an mehr als nur einer Stelle in Ihrem Gehirn verankert werden.

Wir haben Konzepte und Bilder in **unerwarteter** Weise eingesetzt, weil Ihr Gehirn auf Neuigkeiten programmiert ist. Und wir haben Bilder und Ideen mit zumindest *etwas emotionalem* Charakter verwendet, weil Ihr Gehirn darauf eingestellt ist, auf die Biochemie von Gefühlen zu achten. An alles, was ein *Gefühl* in Ihnen auslöst, können Sie sich mit höherer Wahrscheinlichkeit erinnern, selbst wenn dieses Gefühl nicht mehr ist als ein bisschen **Belustigung, Überraschung oder Interesse**.

Wir haben einen **umgangssprachlichen Stil** mit direkter Anrede benutzt, denn Ihr Gehirn ist von Natur aus aufmerksamer, wenn es Sie in einer Unterhaltung wähnt als wenn es davon ausgeht, dass Sie passiv einer Präsentation zuhören – sogar dann, wenn Sie *lesen*.

Wir haben mehr als 100 **Aktivitäten** für Sie vorgesehen, denn Ihr Gehirn lernt und behält von Natur aus besser, wenn Sie Dinge **tun**, als wenn Sie nur darüber *lesen*. Und wir haben die Übungen zwar anspruchsvoll, aber doch lösbar gemacht, denn so ist es den meisten Lesern am liebsten.

Wir haben **mehrere unterschiedliche Lernstile** eingesetzt, denn vielleicht bevorzugen *Sie* ein Schritt-für-Schritt-Vorgehen, während jemand anders erst einmal den groben Zusammenhang verstehen und ein Dritter einfach nur ein Code-Beispiel sehen möchte. Aber ganz abgesehen von den jeweiligen Lernvorlieben profitiert *jeder* davon, wenn er die gleichen Inhalte in unterschiedlicher Form präsentiert bekommt.

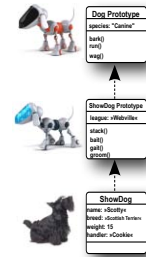
Wir liefern Inhalte für **beide Seiten Ihres Gehirns**, denn je mehr Sie von Ihrem Gehirn einsetzen, umso wahrscheinlicher werden Sie lernen und behalten und umso länger bleiben Sie konzentriert. Wenn Sie mit einer Seite des Gehirns arbeiten, bedeutet das häufig, dass sich die andere Seite des Gehirns ausruhen kann; so können Sie über einen längeren Zeitraum produktiver lernen.

Und wir haben **Geschichten** und Übungen aufgenommen, die **mehr als einen Blickwinkel repräsentieren**, denn Ihr Gehirn lernt von Natur aus intensiver, wenn es gezwungen ist, selbst zu analysieren und zu beurteilen.

Wir haben **Herausforderungen** eingefügt: in Form von Übungen und indem wir **Fragen** stellen, auf die es nicht immer eine eindeutige Antwort gibt, denn Ihr Gehirn ist darauf eingestellt, zu lernen und sich zu erinnern, wenn es an etwas *arbeiten* muss. Überlegen Sie: Ihren *Körper* bekommen Sie ja auch nicht in Form, wenn Sie nur die Leute auf dem Sportplatz *beobachten*. Aber wir haben unser Bestes getan, um dafür zu sorgen, dass Sie – wenn Sie schon hart arbeiten – an den *richtigen* Dingen arbeiten. Dass Sie **nicht einen einzigen Dendriten darauf verschwenden**, ein schwer verständliches Beispiel zu verarbeiten oder einen schwierigen, mit Fachbegriffen gespickten oder übermäßig gedrängten Text zu analysieren.

Wir haben **Menschen** eingesetzt. In Geschichten, Beispielen, Bildern usw. – denn *Sie sind* ein Mensch. Und Ihr Gehirn schenkt *Menschen* mehr Aufmerksamkeit als *Dingen*.

Wir haben einen **80/20-Ansatz** benutzt. Wir gehen davon aus, dass dies nicht Ihr einziges Buch sein wird, wenn Sie ein genialer Webentwickler werden wollen. Deshalb besprechen wir nicht *alles*. Nur das, was Sie wirklich *brauchen* werden.



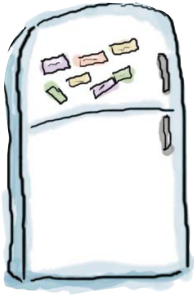
Spielen Sie Browser



Punkt für Punkt

Kreuzworträtsel





Schneiden Sie dies aus und heften Sie es an Ihren Kühlschrank.

## Und das können SIE tun, um sich Ihr Gehirn untertan zu machen

So, wir haben unseren Teil der Arbeit geleistet. Der Rest liegt bei Ihnen. Diese Tipps sind ein Anfang; hören Sie auf Ihr Gehirn und finden Sie heraus, was bei Ihnen funktioniert und was nicht. Probieren Sie neue Wege aus.

- 1 Immer langsam. Je mehr Sie verstehen, umso weniger müssen Sie auswendig lernen.**

Lesen Sie nicht nur. Halten Sie inne und denken Sie nach. Wenn das Buch Sie etwas fragt, springen Sie nicht einfach zur Antwort. Stellen Sie sich vor, dass Sie das wirklich jemand *fragt*. Je gründlicher Sie Ihr Gehirn zum Nachdenken zwingen, umso größer ist die Chance, dass Sie lernen und behalten.
- 2 Bearbeiten Sie die Übungen. Machen Sie selbst Notizen.**

Wir haben sie entworfen, aber wenn wir sie auch für Sie lösen würden, wäre das, als ob jemand anderes Ihr Training für Sie absolviert. Und sehen Sie sich die Übungen *nicht einfach nur an*. **Benutzen Sie einen Bleistift.** Es deutet vieles darauf hin, dass körperliche Aktivität *beim* Lernen den Lernerfolg erhöhen kann.
- 3 Lesen Sie die Abschnitte »Es gibt keine Dummen Fragen«.**

Und zwar alle. Das sind keine Zusatzanmerkungen – **sie gehören zum Kerninhalt!** Überspringen Sie sie nicht.
- 4 Lesen Sie dies als Letztes vor dem Schlafengehen. Oder lesen Sie danach zumindest nichts Anspruchsvolles mehr.**

Ein Teil des Lernprozesses (vor allem die Übertragung in das Langzeitgedächtnis) findet erst statt, *nachdem* Sie das Buch zur Seite gelegt haben. Ihr Gehirn braucht *Zeit* für sich, um weitere Verarbeitung zu leisten. Wenn Sie in dieser Zeit etwas Neues aufnehmen, geht ein Teil dessen, was Sie gerade gelernt haben, verloren.
- 5 Trinken Sie Wasser. Viel.**

Ihr Gehirn arbeitet am besten in einem schönen Flüssigkeitsbad. Austrocknung (zu der es schon kommen kann, bevor Sie überhaupt Durst verspüren) beeinträchtigt die kognitive Funktion.
- 6 Reden Sie drüber. Laut.**

Sprechen aktiviert einen anderen Teil des Gehirns. Wenn Sie etwas verstehen wollen oder Ihre Chancen verbessern wollen, sich später daran zu erinnern, sagen Sie es laut. Noch besser: Versuchen Sie es jemand anderem laut zu erklären. Sie lernen dann schneller und haben vielleicht Ideen, auf die Sie beim bloßen Lesen nie gekommen wären.
- 7 Hören Sie auf Ihr Gehirn.**

Achten Sie darauf, Ihr Gehirn nicht zu überladen. Wenn Sie merken, dass Sie etwas nur noch überfliegen oder dass Sie das gerade erst Gelesene vergessen haben, ist es *Zeit* für eine Pause. Ab einem bestimmten Punkt lernen Sie nicht mehr schneller, indem Sie mehr hineinzustopfen versuchen; das kann sogar den Lernprozess stören.
- 8 Aber bitte mit Gefühl!**

Ihr Gehirn muss wissen, dass es *um etwas Wichtiges geht*. Lassen Sie sich in die Geschichten hineinziehen. Erfinden Sie eigene Bildunterschriften für die Fotos. Über einen schlechten Scherz zu stöhnen ist *immer noch* besser, als gar nichts zu fühlen.
- 9 Erschaffen Sie etwas!**

Wenden Sie das hier Gelernte auf ein Design an, an dem Sie gerade arbeiten, oder gestalten Sie ein älteres Projekt damit um. Tun Sie *irgendetwas*, um neben den Übungen in diesem Buch weitere Erfahrungen zu sammeln. Sie brauchen dazu nur einen Bleistift und ein zu lösendes Problem ... ein Problem, das davon profitieren würde, das Sie HTML & CSS nutzen.

# Lies mich!

Dies ist ein Lernerlebnis, kein Nachschlagewerk. Wir haben bewusst alles herausgestrichen, was an irgendeiner Stelle des Buchs hinderlich für den Lernprozess sein könnte. Und wenn Sie das Buch zum ersten Mal durcharbeiten, müssen Sie am Anfang beginnen, denn das Buch baut stets darauf auf, was Sie schon gesehen und gelernt haben.

## **Wir zeigen Ihnen die GUTEN Seiten von JavaScript und warnen Sie vor den SCHLECHTEN.**

JavaScript ist keine Programmiersprache von einer Elitehochschule, bei der es viel Zeit für Kreuzgutachten gibt. JavaScript wurde in die Welt geworfen, weil man es brauchte, und wuchs in der Nachbarschaft der frühen Browser auf. Seien Sie also gewarnt: JavaScript hat einige großartige Seiten und einige, die nicht so toll sind. Insgesamt ist JavaScript aber brillant, wenn Sie es intelligent einsetzen.

In diesem Buch zeigen wir Ihnen, wie Sie die guten Seiten zum größtmöglichen Vorteil nutzen können. Wir weisen Sie aber auch auf die schlechten Dinge hin und zeigen Ihnen, wie Sie diese am besten umgehen können.

## **Es wird nicht jeder Aspekt der Sprache ausführlich behandelt.**

Zu JavaScript gibt es viel zu lernen. Dies ist jedoch kein Referenzbuch, sondern ein Lernbuch. Daher wird auch nicht alles behandelt, was man zu JavaScript wissen kann. Unser Ziel ist es, Ihnen die Grundlagen von JavaScript zu vermitteln, damit Sie die Details in einem beliebigen Referenzbuch nachschlagen und mit JavaScript die Dinge tun können, die Ihnen in den Sinn kommen.

## **Dieses Buch zeigt Ihnen JavaScript im Browser.**

Der Browser ist nicht nur die häufigste Umgebung, in der JavaScript läuft, sondern auch die bequemste (jeder hat einen Computer mit einem Texteditor und einem Browser – mehr brauchen Sie nicht, um mit JavaScript zu beginnen). Das Ausführen von JavaScript im Browser bedeutet sofortige Belohnung: Sie können Code schreiben und müssen die Seite nur neu laden, um direkt zu sehen, was sie tut.

## **Dieses Buch befürwortet gut strukturierten und lesbaren Code, der auf den bestmöglichen Vorgehensweisen basiert.**

Sie wollen Code schreiben, den Sie und andere lesen und verstehen können und der auch in den Browsern der folgenden Generationen noch funktioniert. Sie wollen Code möglichst ohne Schnörkel schreiben, um ihre Aufgabe zu erledigen, und sich mit den wichtigeren Dingen beschäftigen. In diesem Buch zeigen wir Ihnen, wie man klaren, gut organisierten Code schreibt, der von Anfang an auf Veränderungen vorbereitet ist. Code, auf den Sie stolz sein können, Code, den Sie rahmen und an die Wand hängen möchten (denken Sie nur daran, ihn abzunehmen, bevor Sie ein Rendezvous haben).

## **Wir möchten Sie dazu ermuntern, mehr als einen Browser mit diesem Buch zu benutzen.**

Wir zeigen Ihnen, wie man standardfreundliches JavaScript schreibt. Trotzdem werden Ihnen kleine Unterschiede in der Art begegnen, wie Webbrowser JavaScript interpretieren. Wir tun unser Bestes, um sicherzustellen, dass der Code in diesem Buch mit allen modernen Browsern funktioniert, und zeigen