

# HANSER



## Inhaltsverzeichnis

zu

## „JavaScript“

von Ralph Steyer

ISBN (Buch): 978-3-446-43942-9

ISBN (E-Book): 978-3-446-43947-4

Weitere Informationen und Bestellungen unter  
<http://www.hanser-fachbuch.de/978-3-446-43942-9>

sowie im Buchhandel

© Carl Hanser Verlag München

# Inhalt

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Über das Buch	1
1.1.1	An wen sich das Buch wendet	2
1.1.2	Wie dieses Buch organisiert ist	2
1.2	Was Sie unbedingt haben sollten	3
1.2.1	Die Hardware	3
1.2.2	Die Software	4
1.2.3	Node.js und Git	7
1.2.4	Browser als Zentrale der Webentwicklung	8
<b>2</b>	<b>Grundlagen und erste Beispiele</b>	<b>11</b>
2.1	Erste JavaScript-Beispiele	11
2.1.1	Ein einfaches Mitteilungsfenster	11
2.1.2	Schreiben eines angepassten Aktualisierungsdatums	15
2.1.3	Entgegennahme einer Benutzereingabe	16
2.2	Einige Details zum Internet und zum WWW	17
2.3	Die Besonderheit bei der Webprogrammierung	17
2.3.1	Kompilierung versus Interpretation	18
2.3.2	Unterschiedliche Plattformen und Interpretation	18
<b>3</b>	<b>Vorbereitende Basics</b>	<b>21</b>
3.1	Der Aufbau von (X)HTML-Dateien	21
3.1.1	Das Prinzip der Fehlertoleranz	22
3.2	Ein erster Blick auf Objekte und das DOM-Konzept	26
3.3	HTML5 und DOM5	27
3.3.1	Das neue Vokabular	28
3.3.2	Das neue DOM5-Konzept und die MIME-Typen von HTML5	36
3.3.3	Das Namensraumkonzept von HTML5	37
3.3.4	Die erwartete Darstellung	37
3.3.5	Der Browserkontext und die Kommunikation zwischen verschiedenen Kontexten	38

<b>4</b>	<b>Die Versionen von JavaScript und die Einbindung in Webseiten . .</b>	<b>39</b>
4.1	Die Versionszyklen von JavaScript und ECMAScript . . . . .	39
4.2	JavaScript in Webseiten einbinden . . . . .	40
4.2.1	Die Inline-Referenz . . . . .	40
4.2.2	Notation eines Skript-Containers in der Webseite . . . . .	41
4.2.3	Die Verwendung von externen JavaScript-Dateien . . . . .	44
4.3	Der gemeinsame Namensraum . . . . .	45
4.4	Der <noscript>-Container . . . . .	45
4.5	Kann man testen, ob bei einem Browser JavaScript aktiviert ist? . . . . .	46
4.5.1	Eine Browserweiche . . . . .	46
<b>5</b>	<b>Elementare JavaScript-Grundstrukturen . . . . .</b>	<b>49</b>
5.1	Token und Parser . . . . .	50
5.1.1	Token . . . . .	50
5.1.2	Parser . . . . .	50
5.2	Kommentare in JavaScript . . . . .	50
5.2.1	Einzeilige Kommentare in JavaScript . . . . .	51
5.2.2	Mehrzeilige Kommentare in JavaScript . . . . .	51
5.3	Schlüsselwörter in JavaScript . . . . .	52
5.4	Datentypen, Variablen und Literale . . . . .	54
5.4.1	Datentypen und die JavaScript Global Properties . . . . .	54
5.4.2	Literale und Steuerzeichen bei Zeichenketten . . . . .	57
5.4.3	Variablen anlegen . . . . .	59
5.5	Anweisungen . . . . .	62
5.5.1	Ausdrücke . . . . .	62
5.5.2	Blöcke . . . . .	63
5.5.3	Operatoren und Operanden . . . . .	63
5.5.4	Die Operatoren typeof und delete . . . . .	71
5.6	Kontrollstrukturen in JavaScript . . . . .	77
5.6.1	Entscheidungsanweisungen . . . . .	77
5.6.2	Iterationsanweisungen . . . . .	82
5.6.3	Sprunganweisungen . . . . .	86
5.7	Arrays . . . . .	88
5.7.1	Ein Array erzeugen . . . . .	88
5.7.2	Zugriff auf Array-Elemente und der Index . . . . .	90
5.7.3	Beispiele . . . . .	91
5.7.4	Mit Schleifen auf Array-Elemente zugreifen . . . . .	92
5.8	Funktionen, Prozeduren und Methoden . . . . .	96
5.8.1	Funktion, Prozedur, Methode . . . . .	97
5.8.2	Verwendung von Standardfunktionen . . . . .	97
5.8.3	Die Definition eigener Funktionen . . . . .	101
5.8.4	Der Aufruf einer Funktion . . . . .	103
5.8.5	Beispiele zum Nutzen von selbstdefinierten Funktionen . . . . .	103

5.8.6	Lokale Variablen in Funktionen	107
5.8.7	Flexible Anzahl an Funktionsargumenten und der Token arguments	108
5.8.8	Rekursive Funktionsaufrufe	112
5.8.9	Funktionsreferenzen	116
5.8.10	Benannte versus anonyme Funktionen	117
5.8.11	Innere Funktionen – Closures	118
5.8.12	Callbacks	119
5.9	Aufruf einer JavaScript-Funktion per Eventhandler	121
5.9.1	Was ist ein Ereignis?	121
5.9.2	Ein Mechanismus zur Reaktion auf Ereignisse	122
<b>6</b>	<b>Fehlerbehandlung, Dokumentation und Qualitätskontrolle</b>	<b>125</b>
6.1	Ein leidiges Thema – Fehler	125
6.1.1	Welche Fehler gibt es?	126
6.2	Vorbeugen statt heilen	128
6.2.1	Planung und Vorbereitung	128
6.2.2	IDEs und Tools nutzen	129
6.2.3	Kopieren statt tippen	129
6.2.4	Einhalten von Konventionen	129
6.2.5	Saubere Formatierungen	130
6.2.6	Das KISS-Prinzip	130
6.2.7	Kommentare und Dokumentation	131
6.2.8	Kodierungstechniken	132
6.3	Fehler suchen	132
6.3.1	Fehler ohne Debugger finden	132
6.3.2	Fehlersuche mit einem Debugger	134
6.3.3	Fehler beheben	138
6.3.4	Fehler zur Laufzeit abfangen	139
6.4	Erweiterte Techniken	139
6.4.1	Automatische Dokumentation des Quelltextes und Dokumentationskommentare	140
6.4.2	Automatische Kontrolle und Qualitätssicherung mit JSLint	144
<b>7</b>	<b>Objekte und JavaScript</b>	<b>147</b>
7.1	Was sind Objekte?	147
7.1.1	Wie werden Methoden und Eigenschaften in JavaScript verwendet?	148
7.1.2	Klassen, Instanzen, Prototypen und Vererbung	148
7.2	In JavaScript Objekte erzeugen	150
7.2.1	Konstruktoren	150
7.2.2	Test auf die Existenz eines Objekts	153
7.3	Klassenmethoden und Klasseigenschaften verwenden	153
7.4	Native Objekte und Klassen in JavaScript	154
7.4.1	Das globale Objekt	155
7.4.2	Nutzen von Klasselementen aus Math	155

7.4.3	Instanzen und Klasselemente von Number	157
7.4.4	Arrays als Objekte	159
7.4.5	Objekte vom Typ Boolean	161
7.4.6	Datumsoperationen mit Date	161
7.4.7	String-Operationen	170
7.4.8	Objekte vom Typ Function	172
7.4.9	Native Bildobjekte	173
7.4.10	Reguläre Ausdrücke	173
<b>8</b>	<b>Das DOM – das Mysterium der JavaScript-Welt</b>	<b>189</b>
8.1	Was ist das DOM-Konzept?	189
8.2	Grundsätzliches zum Weg durch das DOM	190
8.2.1	Knoten und Verwandtschaftsbeziehungen	190
8.2.2	Wichtige Knotentypen bei Webseiten	191
8.2.3	Objekthierarchien im DOM	191
8.3	Eine Übersicht der verfügbaren DOM-Objekte	192
8.4	Das Konzept der Objektfelder	193
8.5	Generelle Zugriffsmöglichkeiten auf DOM-Objekte	194
8.5.1	Der Zugriff über einen Namen – getElementByName()	195
8.5.2	Zugriff über eine ID – getElementById()	196
8.5.3	Zugriff über den Elementnamen – getElementsByTagName()	198
8.5.4	Zugriff über die Klasse	198
8.5.5	Der kurze Dienstweg – this	199
8.5.6	Die with()-Anweisung	199
8.6	Zugriff auf Inhalte von Elementen in der Webseite	200
8.6.1	Zugriff auf Textinhalte	201
8.6.2	Zugriff auf Formularinhalte	201
8.6.3	Zugriff auf klassische HTML-Attribute	202
8.7	Das Objekt document	204
8.7.1	Die Eigenschaften von document	204
8.7.2	Die Methoden von document	209
8.8	Das node-Objekt	214
8.8.1	Knotenarten	215
8.8.2	Die Eigenschaften eines node-Objekts	216
8.8.3	Die Methoden eines node-Objekts	220
8.9	Fenstertechniken	223
8.9.1	Die Eigenschaften des window-Objekts	224
8.9.2	Die Methoden von window	226
8.9.3	Anwendungen mit Fenstern	228
8.10	Frames, IFrames und das Objektfeld frames	243
8.10.1	Daten aus IFrames laden – ein unsichtbarer Datenspeicher	244
8.10.2	Daten in einem IFrame speichern	248
8.11	Browserauswertung und das Objekt navigator	249

8.11.1	Auflistung aller Eigenschaften von navigator	250
8.11.2	Eigenschaften und Methoden	251
8.11.3	Die Browserkennung	251
8.12	Zugriff auf Bildschirminformationen eines Besuchers	254
8.13	Ein Beispiel für die rekursive Auflistung aller Knoten einer Webseite	255
8.14	DHTML und Animationen	257
8.14.1	DHTML und Animation mit reinem JavaScript	258
8.14.2	DHTML und Animation mit reinem CSS	258
8.14.3	Verbinden von JavaScript und Style Sheets	259
8.14.4	Verändern des Inhalts der Webseite	261
8.14.5	Eine Animation mit style	261
<b>9</b>	<b>Ausnahmebehandlung, Ereignisbehandlung und das Ereignisobjekt</b>	<b>265</b>
9.1	Eventhandler in JavaScript	265
9.1.1	(X)HTML- versus JavaScript-Eventhandler	266
9.2	Verschiedene Ereignisse	267
9.2.1	Laden und Verlassen der Webseite	267
9.2.2	Mausereignisse	268
9.2.3	Tastaturereignisse	268
9.2.4	Formularereignisse	268
9.2.5	Weitere Ereignisse	268
9.3	Grundlagen zu Ereignisobjekten	269
9.3.1	Blubbern und die Bubble-Phase	269
9.3.2	Das Ereignisobjekt verwenden	270
9.4	Ausnahmebehandlung	276
9.4.1	Was ist eine Ausnahme?	277
9.4.2	Wozu ein Ausnahmekonzept?	278
9.4.3	Das Auffangen einer Ausnahme	280
9.4.4	Selbstdefinierte Ausnahmen erzeugen und verwenden	283
9.5	Universelle Ereignisbehandlung mit Ausnahmebehandlung	285
9.6	Globale Ereignisbehandlung	287
9.6.1	Eine universelle Funktion zur Registrierung eines Listeners	288
<b>10</b>	<b>AJAX und das Web 2.0</b>	<b>291</b>
10.1	Die Beziehung zwischen Webserver und Browser	292
10.1.1	Der grundsätzliche Ablauf einer Kommunikation zwischen Herr und Diener – HTTP	292
10.1.2	Die HTTP-Details und die HTTP-Header	293
10.1.3	Die Methoden der Datenübertragung per HTTP	295
10.1.4	Die URL-Kodierung	296
10.2	Das Datenformat	297
10.2.1	Klartextdatenformate	297

10.2.2	Vollständige HTML- und XHTML-Dateien	298
10.2.3	HTML-Fragmente	298
10.2.4	Reiner Klartext	298
10.2.5	Klartext mit Struktur und beliebiger Logik	299
10.3	Was ist AJAX und was bezeichnet Web 2.0?	299
10.3.1	AJAX als technische Basis des Web 2.0	299
10.4	Der grundsätzliche Ablauf einer AJAX-Anfrage und das XMLHttpRequest-Objekt (XHR)	301
10.4.1	Der grundsätzliche Ablauf	301
10.4.2	Ein XMLHttpRequest-Objekt erzeugen	302
10.5	Die Anforderung von Daten	304
10.5.1	Die Methoden eines XHR-Objekts	304
10.5.2	Die Eigenschaften eines XMLHttpRequest-Objekts	306
10.6	Verschiedene Datentypen vom Server per AJAX nachfordern	307
10.6.1	Eine reine Textdatei mit AJAX nachfordern	308
10.6.2	Übergabe von Werten an den Webserver	313
10.6.3	Nachfordern von XML-Daten	314
10.6.4	JSON-Daten anfordern und verarbeiten	318
<b>11</b>	<b>Erweiterte Tools, Hintergründe &amp; Techniken</b>	<b>321</b>
11.1	Optimierung von JavaScripts	321
11.1.1	Interpreterprinzip = langsam - JIT = schnell	323
11.1.2	Rezepte	323
11.2	Sicherheitsfragen	326
11.2.1	Welche Sicherheitslücken hat JavaScript?	326
11.2.2	Wie kann man seine Skripte schützen?	329
11.3	JUnit-Testing	331
11.3.1	Download des JUnit-Frameworks, Dokumentation und Tutorial	331
11.3.2	Wie arbeitet man grundsätzlich bei testgetriebener Entwicklung mit JUnit?	331
11.3.3	Ergebnis und Aufbau eines JUnit-Tests	333
11.3.4	Ein praktisches Beispiel	334
11.3.5	Pro und Contra TDD	337
<b>12</b>	<b>Zugriff auf Formulare</b>	<b>339</b>
12.1	Das Formularobjekt	339
12.1.1	Die Eigenschaften eines Formularobjekts	340
12.1.2	Die Methoden	340
12.2	Die Elemente in dem Webformular	341
12.2.1	Eigenschaften von elements	341
12.2.2	Die Methoden von elements	342
12.2.3	Das Objektfeld options	342
12.3	Zugriff auf ein Webformular per JavaScript	343
12.3.1	Zugriff auf den Wert eines Formularelements	343

12.4	Verschicken und Zurücksetzen von Formulardaten	350
12.4.1	Klassisches Verschicken von Formulardaten	350
12.4.2	Verschicken von Formulardaten mit AJAX	352
12.5	Plausibilisieren: Formulare	353
12.5.1	Abbruch von Aktionen	354
12.5.2	Fehlerreaktion	355
12.5.3	Ein Beispiel mit Plausibilisierung	355
<b>13</b>	<b>Erweiterte Objekttechniken</b>	<b>359</b>
13.1	Was zeichnet in JavaScript eine Methode gegenüber einer normalen Funktion aus?	359
13.2	Ein einzelnes Objekt erweitern	360
13.2.1	Ein Beispiel für die Objekterweiterung	360
13.3	Was ist Prototyping?	362
13.3.1	Prototyping von Object	363
13.3.2	Die Standardklasse Image erweitern	365
13.3.3	Date mit Prototyping erweitern	365
13.3.4	Und doch nicht ganz ein Klasselement	367
13.4	Erstellen von Konstruktormethoden	368
13.4.1	Was zeichnet eine Konstruktormethode aus?	369
13.4.2	Vererbung mit Prototyping simulieren	373
13.5	Datenkapselung	374
13.6	Ausblick auf JavaScript Version 2.0	377
13.6.1	Explizite Datentypen und der use strict-Modus	378
13.6.2	Klassendefinitionen und mehr	380
13.6.3	Namensräume	382
13.6.4	Konstanten	382
<b>14</b>	<b>JavaScript-Techniken bei HTML5 und mehr</b>	<b>383</b>
14.1	Multithreading über Web Workers	383
14.1.1	Erzeugen von Web Workers	384
14.1.2	Kommunikation mit einem Web Worker	385
14.1.3	Einen Worker mit terminate() beenden	385
14.1.4	Beispiele zu Web Worker	385
14.2	WebSockets und serverseitiges JavaScript mit Node.js	392
14.2.1	Push-Service im Web	393
14.2.2	Die WebSocket-Spezifikation	394
14.2.3	Same-Origin-Policy und das Sprengen der Grenzen	394
14.2.4	Ein WebSocket mit JavaScript erzeugen	395
14.2.5	Die Serverseite anhand von Node.js	396
14.2.6	Ein praktisches Beispiel für einen einfachen WebSocket-Zugriff	398
14.3	Daten im Browser speichern mit HTML5 und der Technik des Local Data Storage bzw. Session Data Storage	401

14.3.1	Die Objekte localStorage und sessionStorage sowie deren Methoden und Eigenschaft .....	402
14.3.2	Ein Beispiel, um einfache Daten in einem Local Storage abzulegen ...	403
14.3.3	Ein Beispiel zum Speichern von komplexen Informationen .....	404
14.4	Geodating mit HTML5 und mehr .....	408
14.4.1	Hintergründe der Geolokalisierung .....	408
14.4.2	Der Geolokalisierungsdienst von Google .....	411
14.4.3	Die Geolokalisierung mit HTML5 .....	416
14.5	Zeichnen mit Canvas, SVG und CSS3 .....	419
14.5.1	Canvas .....	419
14.5.2	Umgang mit Scalable Vector Graphics - SVG .....	444
14.5.3	CSS3-Keyframes .....	450
<b>Index .....</b>		<b>453</b>