
Inhaltsverzeichnis

1	Einleitung	1
1.1	Das Ziel des Buchs.	1
1.2	An wen sich das Buch wendet	2
1.3	Voraussetzungen.	3
1.3.1	Hardware	3
1.3.2	Die Software	4
2	Erste Beispiele – Der Sprung ins kalte Wasser	11
2.1	Der Interaktivmodus – die Kommandozeile bei JavaScript.	11
2.2	Anweisungen in (echten) Quelltext auslagern	13
2.3	Ein einfaches Mitteilungsfenster	17
2.4	Schreiben in die Webseite	18
2.5	Entgegennahme einer Benutzereingabe	19
2.6	Besonderheit bei der Webprogrammierung	19
3	Versionen von JavaScript und Einbindung in Webseiten	23
3.1	Versionszyklen von JavaScript, ECMAScript & Co	24
3.2	JavaScript in Webseiten	25
3.2.1	Die Inline-Referenz	26
3.2.2	Notation eines Skript-Containers in der Webseite	26
3.2.3	Verwendung von externen JavaScript-Dateien	29
3.3	Der gemeinsame Namensraum	31
3.4	Kann man testen, ob bei einem Browser JavaScript aktiviert ist?	33
3.4.1	Eine Browserweiche	33
4	Elementare JavaScript-Grundstrukturen	35
4.1	Token und Parser	35
4.1.1	Zerlegen von Quelltext	36
4.2	Kommentare	38
4.3	Schlüsselwörter und definierte Literale in JavaScript	39
4.4	Bezeichner und Namenskonventionen	40

4.4.1	Regeln für Bezeichner in JavaScript	40
4.4.2	Namenskonventionen.	42
4.5	Anweisungen	43
4.5.1	Ausdrücke	44
4.5.2	Blöcke	44
4.6	Datentypen, Variablen und Literale	45
4.6.1	Datentypen und die JavaScript Global Properties	45
4.6.2	JavaScript Global Properties und das globale Objekt	49
4.6.3	Variablen	52
4.6.4	Konstanten	57
4.6.5	Spezielle Schreibweisen für Literale	57
4.7	Operatoren	63
4.7.1	Die Operatoren typeof und delete	63
4.7.2	Arithmetische Operatoren und der Verknüpfungsoperator	63
4.7.3	Vergleichsoperatoren	69
4.7.4	Logische Operatoren	70
4.7.5	Bit-Operatoren	70
4.7.6	Zuweisungsoperatoren	77
4.7.7	Die Operatorenrangfolge	78
5	Kontrollflussanweisungen	79
5.1	Entscheidungsanweisungen	79
5.1.1	Die if-Entscheidungsanweisungen.	80
5.1.2	Die switch-case-Fallunterscheidung	83
5.2	Iterationsanweisungen	86
5.2.1	Die while-Schleife	86
5.2.2	Die do-while-Schleife	87
5.2.3	Die for-Schleife	88
5.2.4	Verschachtelte Schleifen	89
5.3	Sprunganweisungen	90
5.3.1	Abbruch mit break	91
5.3.2	Fortsetzen mit continue	91
5.3.3	Rückgabe mit return	92
5.3.4	Unterbrechung mit throw.	92
6	Arrays, JSON und andere iterierbare Elemente	93
6.1	Arrays.	93
6.1.1	Ein Array erzeugen	95
6.1.2	Deklaration mit einem Array-Literal	96
6.1.3	Der Datentyp eines Arrays.	96
6.1.4	Zugriff auf Array-Elemente	96
6.1.5	Verschachtelte Arrays	97
6.1.6	Die Eigenschaft length.	98

6.1.7	Textindizes	99
6.2	Die Objekt-Notation – JSON	99
6.2.1	Der Datentyp bei JSON	100
6.2.2	Zugriff auf Elemente in einem JSON-Objekt	100
6.2.3	Verschachtelte JSON-Arrays	101
6.3	Iteration über Arrays und JSON	103
6.3.1	Zugriff auf numerisch indizierte Arrays	103
6.3.2	Zugriff auf alle Elemente eines Arrays über for-in und for-of sowie forEach	104
6.3.3	Über verschachtelte Strukturen iterieren	107
6.4	Weitere iterative Datenstrukturen	112
6.4.1	Maps	112
6.4.2	Sets	113
6.4.3	Weitere Datenstrukturen	114
7	Funktionen, Prozeduren und Methoden	117
7.1	Was ist allgemein ein Unterprogramm?	117
7.2	Die Deklaration eigener Funktionen in JavaScript	118
7.3	Aufruf einer Funktion	120
7.4	Lokale Variablen in Funktionen	121
7.5	Besonderheiten bei JavaScript im Zusammenhang mit Funktionen	122
7.5.1	Funktionen redefinieren – wer zuletzt kommt	123
7.5.2	Funktionen mit Parametern	123
7.5.3	Funktionsreferenzen	128
7.5.4	Benannte versus anonyme Funktionen	131
7.5.5	Hoisting bei Funktionen	132
7.5.6	Innere Funktionen – Closures	133
7.5.7	Lambda-Ausdrücke	134
7.5.8	Callbacks	136
7.5.9	JavaScript-Generatoren	137
7.5.10	Weitere Möglichkeiten für die Deklaration von Funktionen . . .	138
7.5.11	Rekursive Funktionsaufrufe	140
7.6	Vordefinierte Funktionen in JavaScript	145
8	Module	151
8.1	Was sind Module?	151
8.2	CommonJS	152
8.3	ES6-Module	155
8.4	Module in Browsern	157
9	Objekte und JavaScript	161
9.1	Grundlagen der OOP	161
9.1.1	Was sind Objekte?	162

9.1.2	Klassen, Instanzen, Prototypen und Vererbung	162
9.1.3	Ziele der OOP – Wiederverwendbarkeit und bessere Softwarequalität.	163
9.1.4	Kernkonzepte der Objektorientierung	164
9.1.5	Wie lange gibt es Objektorientierung als Konzept?	165
9.1.6	Hintergründe der OOP.	166
9.1.7	Funktionales Paradigma.	166
9.1.8	Alternative Paradigma	167
9.2	Wie Objekte in JavaScript realisiert werden	168
9.2.1	Array = Objekt?	169
9.2.2	Klassen vs Prototypen in JavaScript	170
9.2.3	Was zeichnet in JavaScript eine Methode gegenüber einer normalen Funktion aus?	171
9.2.4	Ein einzelnes Objekt erweitern	174
9.3	Details zu Prototyping	175
9.3.1	Prototyping von Objekttypen.	176
9.3.2	Und doch nicht ganz ein Klassenelement	177
9.3.3	Erstellen von Konstruktormethoden	178
9.4	Vererbung.	182
9.4.1	Vererbung mit Prototyping.	182
9.4.2	Vererbung mit class und extends	183
9.5	Datenkapselung	184
9.6	Module bei OOP	186
9.7	Frameworks und POJO	186
9.7.1	Frameworks und Toolkits.	187
9.7.2	POJOs	188
10	Eingebaute Objektdeklarationen	189
10.1	Alles sind Objekte	189
10.2	Standardobjekte in JavaScript erzeugen.	190
10.3	Klassenmethoden und Klasseneigenschaften verwenden	191
10.4	Native Objektdeklarationen in JavaScript	192
10.4.1	Math.	192
10.4.2	Instanzen und Elemente von Number	194
10.4.3	Arrays als Objekte	194
10.4.4	Objekte vom Typ Boolean	196
10.4.5	Datumsoperationen mit Date	196
10.4.6	String-Operationen	203
10.4.7	Objekte vom Typ Function	204
10.4.8	Native Bildobjekte.	207
10.4.9	Reguläre Ausdrücke.	207

11	Der DOM – das Mysterium der JavaScript-Welt	215
11.1	Was ist das DOM-Konzept?	215
11.1.1	Grundsätzliches zum Weg durch den DOM	216
11.2	Wichtige DOM-Objekte.	218
11.2.1	Objekthierarchien im DOM.	220
11.3	Eventhandling	221
11.3.1	Verschiedene Ereignisse	222
11.3.2	Grundlagen zu Ereignisobjekten	223
11.4	Generelle Zugriffsmöglichkeiten auf DOM-Objekte.	229
11.4.1	Der Zugriff über einen Namen – getElementByName.	230
11.4.2	Zugriff über eine Id – getElementById	231
11.5	Zugriff auf Inhalte von Elementen in der Webseite	234
11.5.1	Zugriff auf Textinhalte.	234
11.5.2	Zugriff auf Formulare	235
11.5.3	Zugriff auf klassische HTML-Attribute.	247
11.6	Das Objekt document.	249
11.6.1	Ein Dokument mit Knoten dynamisch aufbauen.	250
11.7	Das node-Objekt	254
11.7.1	Knotenarten	254
11.7.2	Eigenschaften eines node-Objekts.	255
11.7.3	Methoden eines node-Objekts	255
11.8	Fenstertechniken	260
11.8.1	Member des window-Objekts	260
11.8.2	Anwendungen mit Fenstern.	261
11.8.3	Frames, IFrames und das Objektfeld frames	271
11.9	Browserauswertung und das Objekt navigator	277
11.9.1	Auflistung aller Eigenschaften von navigator	278
12	Ausnahmebehandlung	281
12.1	Was ist Ausnahmebehandlung?	281
12.1.1	Eine Ausnahme	281
12.1.2	Wozu ein Ausnahmekonzept?	283
12.1.3	Das Auffangen einer Ausnahme	284
12.1.4	Der finally-Block	287
12.2	Selbstdefinierte Ausnahmen erzeugen und verwenden	287
12.2.1	Erstellen einer selbstdefinierten Ausnahme.	287
12.2.2	Auswerfen von Ausnahmen mit throw	288
12.2.3	Ein Beispiel zum manuellen Werfen von Ausnahmen.	288
13	Asynchrone Programmierung	291
13.1	Was versteht man unter asynchronem JavaScript?	291
13.2	Die Beziehung zwischen Webserver und Browser.	292

13.2.1	Der grundsätzliche Ablauf einer Kommunikation – HTTP und HTTPS	293
13.3	Ajax & Web 2.0	300
13.3.1	Das Datenformat	301
13.3.2	Was ist Ajax und was bezeichnet Web 2.0?	303
13.3.3	Der grundsätzliche Ablauf	304
13.3.4	Ein XMLHttpRequest-Objekt erzeugen	304
13.3.5	Die Anforderung von Daten	305
13.3.6	Verschiedene Datentypen vom Server per Ajax nachfordern	308
13.4	Deferred Object und Promises	321
13.4.1	Promises	322
13.4.2	Deferred Objects	326
13.5	Web Worker	327
13.5.1	Was ist ein Web Worker?	327
13.5.2	Erzeugen von Web Workern	328
13.5.3	Die Kommunikation mit einem Web Worker	328
13.5.4	Kommunikation mit einem Web Worker	329
13.5.5	Einen Worker mit terminate beenden	329
13.5.6	Beispiele zu Web Worker	329
13.6	WebSockets	336
13.6.1	Allgemeiner Ablauf	336
13.6.2	Die WebSocket-Spezifikation	337
13.6.3	Ein WebSocket mit JavaScript erzeugen	338
13.6.4	Der WebSocket- Handshake	338
13.6.5	Senden von Daten an den Server – die Methode send	339
13.6.6	Reaktion mit Callbacks und Eventhndlern	339
13.6.7	Die Serverseite anhand von Node.js	340
13.7	Webservices	343
14	Erweiterte Techniken	347
14.1	Daten im Browser speichern	347
14.1.1	Cookies	348
14.1.2	Local Storage und Session Storage	349
14.2	Geolocation	355
14.2.1	Hintergründe der Geolokalisierung	356
14.2.2	Die konkrete Geolokalisierung	358
14.3	DHTML, Zeichnen und Animationen	361
14.3.1	DHTML und Animation mit reinem JavaScript oder CSS	362
14.3.2	Verbinden von JavaScript und Style Sheets	363
14.3.3	Zeichnen mit Canvas und SVG	370
14.3.4	Umgang mit Scalable Vector Graphics – SVG	378

15	Eine Frage der Qualität	385
15.1	Was versteht man allgemein unter Softwarequalität?	385
15.2	Allgemeine Konzepte für bessere Qualität	386
15.2.1	KISS-Prinzip	386
15.2.2	Das DRY-Prinzip	387
15.2.3	Weitere Konzepte	387
15.3	Ergänzende Regeln für JavaScript-Kodierung	388
15.4	Ein leidiges Thema – Fehler	390
15.4.1	Welche Fehler gibt es?	390
15.4.2	Vorbeugen statt heilen	392
15.4.3	Fehler suchen	392
15.4.4	Fehler beheben	394
15.4.5	Fehler zur Laufzeit abfangen	395
15.5	Erweiterte Techniken	395
15.5.1	Automatische Dokumentation des Quelltextes und Dokumentationskommentare	396
15.5.2	Automatische Kontrolle und Qualitätssicherung mit JSLint	396
15.5.3	xUnit-Testing	398
15.5.4	Cross-Engine-Probleme vermeiden	406
15.5.5	Optimierung von JavaScripts	407
	Stichwortverzeichnis	411