

Auf einen Blick

1	Grundlagen und Einführung	29
2	Erste Schritte	55
3	Sprachkern	85
4	Mit Referenztypen arbeiten	239
5	Webseiten dynamisch verändern	377
6	Ereignisse verarbeiten und auslösen	435
7	Mit Formularen arbeiten	487
8	Browser steuern und Browserinformationen auslesen	517
9	Inhalte einer Webseite dynamisch nachladen	545
10	Aufgaben vereinfachen mit jQuery	589
11	Bilder und Grafiken dynamisch erstellen	635
12	Moderne Web-APIs verwenden	667
13	Objektorientierte Programmierung	771
14	Funktionale Programmierung	813
15	Den Quelltext richtig strukturieren	827
16	Asynchrone Programmierung und weitere fortgeschrittene Features verwenden	849
17	Serverseitige Anwendungen mit Node.js erstellen	883
18	Mobile Anwendungen mit JavaScript erstellen	931
19	Desktopanwendungen mit JavaScript	951
20	Mikrocontroller mit JavaScript steuern	965
21	Einen professionellen Entwicklungsprozess aufsetzen	987

Inhalt

Materialien zum Buch	23
Vorwort	25

1 Grundlagen und Einführung 29

1.1 Grundlagen der Programmierung	29
1.1.1 Mit dem Computer kommunizieren	30
1.1.2 Programmiersprachen	31
1.1.3 Hilfsmittel für den Programmentwurf	39
1.2 Einführung in JavaScript	44
1.2.1 Historie	45
1.2.2 Anwendungsgebiete	46
1.3 Zusammenfassung	53

2 Erste Schritte 55

2.1 Einführung in JavaScript und die Webentwicklung	55
2.1.1 Der Zusammenhang zwischen HTML, CSS und JavaScript	55
2.1.2 Das richtige Werkzeug für die Entwicklung	59
2.2 JavaScript in eine Webseite einbinden	63
2.2.1 Eine geeignete Ordnerstruktur vorbereiten	63
2.2.2 Eine JavaScript-Datei erstellen	64
2.2.3 Eine JavaScript-Datei in eine HTML-Datei einbinden	65
2.2.4 JavaScript direkt innerhalb des HTML definieren	68
2.2.5 Platzierung und Ausführung der <script>-Elemente	69
2.2.6 Den Quelltext anzeigen	73
2.3 Eine Ausgabe erzeugen	76
2.3.1 Standarddialogfenster anzeigen	76
2.3.2 Auf die Konsole schreiben	78
2.3.3 Bestehende UI-Komponenten verwenden	82
2.4 Zusammenfassung	83

3 Sprachkern	85
3.1 Werte in Variablen speichern	85
3.1.1 Variablen definieren	85
3.1.2 Gültige Variablenamen verwenden	88
3.1.3 Konstanten definieren	96
3.2 Die verschiedenen Datentypen verwenden	96
3.2.1 Zahlen	97
3.2.2 Zeichenketten	100
3.2.3 Boolesche Werte	106
3.2.4 Arrays	106
3.2.5 Objekte	112
3.2.6 Besondere Datentypen	113
3.2.7 Symbole	115
3.3 Die verschiedenen Operatoren einsetzen	116
3.3.1 Operatoren für das Arbeiten mit Zahlen	117
3.3.2 Operatoren für das einfachere Zuweisen	118
3.3.3 Operatoren für das Arbeiten mit Zeichenketten	120
3.3.4 Operatoren für das Arbeiten mit booleschen Werten	121
3.3.5 Operatoren für das Arbeiten mit Bits	128
3.3.6 Operatoren für das Vergleichen von Werten	129
3.3.7 Der Optional Chaining Operator	132
3.3.8 Die Logical Assignment Operatoren	134
3.3.9 Operatoren für spezielle Operationen	136
3.4 Den Ablauf eines Programms steuern	137
3.4.1 Bedingte Anweisungen definieren	137
3.4.2 Verzweigungen definieren	139
3.4.3 Den Auswahloperator verwenden	145
3.4.4 Mehrfachverzweigungen definieren	147
3.4.5 Zählschleifen definieren	153
3.4.6 Kopfgesteuerte Schleifen definieren	161
3.4.7 Fußgesteuerte Schleifen definieren	164
3.4.8 Schleifen und Schleifeniterationen vorzeitig abbrechen	166
3.5 Wiederverwendbare Codebausteine erstellen	175
3.5.1 Funktionen definieren	175
3.5.2 Funktionen aufrufen	178
3.5.3 Funktionsparameter übergeben und auswerten	178
3.5.4 Rückgabewerte definieren	187
3.5.5 Standardwerte für Parameter definieren	189
3.5.6 Elemente aus einem Array als Parameter verwenden	191

3.5.7 Funktionen über Kurzschreibweise definieren	193
3.5.8 Zeichenketten über Funktionen verändern	195
3.5.9 Funktionen im Detail	197
3.5.10 Funktionen aufrufen durch Nutzerinteraktion	205
3.6 Auf Fehler reagieren und sie richtig behandeln	206
3.6.1 Syntaxfehler	206
3.6.2 Laufzeitfehler	207
3.6.3 Logische Fehler	208
3.6.4 Das Prinzip der Fehlerbehandlung	209
3.6.5 Fehler fangen und behandeln	211
3.6.6 Fehler auslösen	213
3.6.7 Fehler und der Funktionsaufruf-Stack	216
3.6.8 Bestimmte Anweisungen unabhängig von aufgetretenen Fehlern aufrufen	219
3.7 Den Quelltext kommentieren	225
3.8 Den Code debuggen	225
3.8.1 Einführung	226
3.8.2 Ein einfaches Codebeispiel	226
3.8.3 Haltepunkte definieren	227
3.8.4 Variablenbelegungen einsehen	229
3.8.5 Ein Programm schrittweise ausführen	230
3.8.6 Mehrere Haltepunkte definieren	232
3.8.7 Haltepunkte definieren	233
3.8.8 Den Funktionsaufruf-Stack einsehen	234
3.9 Zusammenfassung	236
4 Mit Referenztypen arbeiten	239
4.1 Unterschied zwischen primitiven Datentypen und Referenztypen	239
4.1.1 Das Prinzip von primitiven Datentypen	239
4.1.2 Das Prinzip von Referenztypen	240
4.1.3 Primitive Datentypen und Referenztypen als Funktionsargumente	242
4.1.4 Den Typ einer Variablen ermitteln	243
4.1.5 Ausblick	246
4.2 Zustand und Verhalten in Objekten kapseln	247
4.2.1 Einführung objektorientierte Programmierung	247
4.2.2 Objekte erstellen über die Literal-Schreibweise	248
4.2.3 Objekte erstellen über Konstruktorfunktionen	250
4.2.4 Objekte erstellen unter Verwendung von Klassen	253

4.2.5	Objekte erstellen über die Funktion »Object.create()«	257
4.2.6	Auf Eigenschaften zugreifen und Methoden aufrufen	260
4.2.7	Objekteigenschaften und Objektmethoden hinzufügen oder überschreiben	267
4.2.8	Objekteigenschaften und Objektmethoden löschen	271
4.2.9	Objekteigenschaften und Objektmethoden ausgeben	274
4.2.10	Symbole zur Definition eindeutiger Objekteigenschaften verwenden	277
4.2.11	Änderungen an Objekten verhindern	279
4.3	Mit Arrays arbeiten	282
4.3.1	Arrays erzeugen und initialisieren	282
4.3.2	Auf Elemente eines Arrays zugreifen	286
4.3.3	Elemente einem Array hinzufügen	287
4.3.4	Elemente aus einem Array entfernen	292
4.3.5	Einen Teil der Elemente aus einem Array kopieren	295
4.3.6	Arrays sortieren	298
4.3.7	Arrays als Stack verwenden	301
4.3.8	Arrays als Queue verwenden	302
4.3.9	Elemente in Arrays finden	304
4.3.10	Elemente innerhalb eines Arrays kopieren	307
4.3.11	Arrays in Zeichenketten umwandeln	308
4.4	Werte aus Arrays und Objekten extrahieren	308
4.4.1	Werte aus Arrays extrahieren	308
4.4.2	Werte aus Objekten extrahieren	312
4.4.3	Werte innerhalb einer Schleife extrahieren	316
4.4.4	Argumente einer Funktion extrahieren	317
4.4.5	Objekteigenschaften in ein anderes Objekt kopieren	319
4.4.6	Objekteigenschaften aus einem anderen Objekt kopieren	320
4.5	Mit Zeichenketten arbeiten	321
4.5.1	Der Aufbau einer Zeichenkette	321
4.5.2	Die Länge einer Zeichenkette ermitteln	322
4.5.3	Innerhalb einer Zeichenkette suchen	323
4.5.4	Teile einer Zeichenkette extrahieren	326
4.6	Maps verwenden	329
4.6.1	Maps erstellen	330
4.6.2	Grundlegende Operationen	331
4.6.3	Über Maps iterieren	332
4.6.4	Weak Maps verwenden	335
4.7	Sets verwenden	336
4.7.1	Sets erstellen	337
4.7.2	Grundlegende Operationen von Sets	337

4.7.3	Über Sets iterieren	339
4.7.4	Weak Sets verwenden	340
4.8	Sonstige globale Objekte	341
4.8.1	Mit Datum und Zeit arbeiten	341
4.8.2	Komplexe Berechnungen durchführen	344
4.8.3	Wrapperobjekte für primitive Datentypen	345
4.9	Mit regulären Ausdrücken arbeiten	345
4.9.1	Reguläre Ausdrücke definieren	346
4.9.2	Zeichen gegen einen regulären Ausdruck testen	346
4.9.3	Zeichenklassen verwenden	349
4.9.4	Anfang und Ende begrenzen	353
4.9.5	Quantifizierer verwenden	356
4.9.6	Nach Vorkommen suchen	361
4.9.7	Alle Vorkommen innerhalb einer Zeichenkette suchen	362
4.9.8	Auf einzelne Teile eines Vorkommens zugreifen	363
4.9.9	Nach bestimmten Zeichenketten suchen	364
4.9.10	Vorkommen innerhalb einer Zeichenkette ersetzen	365
4.9.11	Nach Vorkommen suchen	366
4.9.12	Zeichenketten zerteilen	366
4.10	Funktionen als Referenztypen	367
4.10.1	Funktionen als Argumente verwenden	367
4.10.2	Funktionen als Rückgabewert verwenden	370
4.10.3	Standardmethoden jeder Funktion	371
4.11	Zusammenfassung	375
5	Webseiten dynamisch verändern	377
5.1	Aufbau einer Webseite	377
5.1.1	Document Object Model	377
5.1.2	Die verschiedenen Knotentypen	378
5.1.3	Der Dokumentknoten	382
5.2	Elemente selektieren	383
5.2.1	Elemente per ID selektieren	385
5.2.2	Elemente per Klasse selektieren	388
5.2.3	Elemente nach Elementnamen selektieren	391
5.2.4	Elemente nach Namen selektieren	393
5.2.5	Elemente per Selektor selektieren	395
5.2.6	Das Elternelement eines Elements selektieren	401

5.2.7	Die Kindelemente eines Elements selektieren	404
5.2.8	Die Geschwisterelemente eines Elements selektieren	408
5.2.9	Selektionsmethoden auf Elementen aufrufen	411
5.2.10	Elemente nach Typ selektieren	413
5.3	Mit Textknoten arbeiten	413
5.3.1	Auf den Textinhalt eines Elements zugreifen	414
5.3.2	Den Textinhalt eines Elements verändern	415
5.3.3	Das HTML unterhalb eines Elements verändern	416
5.3.4	Textknoten erstellen und hinzufügen	417
5.4	Mit Elementen arbeiten	417
5.4.1	Elemente erstellen und hinzufügen	418
5.4.2	Elemente und Knoten entfernen	421
5.4.3	Die verschiedenen Typen von HTML-Elementen	422
5.5	Mit Attributen arbeiten	427
5.5.1	Den Wert eines Attributs auslesen	428
5.5.2	Den Wert eines Attributs ändern oder ein neues Attribut hinzufügen	429
5.5.3	Attributknoten erstellen und hinzufügen	430
5.5.4	Attribute entfernen	431
5.5.5	Auf CSS-Klassen zugreifen	431
5.6	Zusammenfassung	432
6	Ereignisse verarbeiten und auslösen	435
6.1	Das Konzept der ereignisgesteuerten Programmierung	435
6.2	Auf Ereignisse reagieren	436
6.2.1	Einen Event-Handler per HTML definieren	439
6.2.2	Einen Event-Handler per JavaScript definieren	441
6.2.3	Event-Listener definieren	443
6.2.4	Mehrere Event-Listener definieren	445
6.2.5	Argumente an Event-Listener übergeben	447
6.2.6	Event-Listener entfernen	449
6.2.7	Event-Handler und Event-Listener per Helferfunktion definieren	450
6.2.8	Auf Informationen eines Ereignisses zugreifen	451
6.3	Die verschiedenen Typen von Ereignissen	453
6.3.1	Ereignisse bei Interaktion mit der Maus	454
6.3.2	Ereignisse bei Interaktion mit Tastatur und Textfeldern	459
6.3.3	Ereignisse beim Arbeiten mit Formularen	462
6.3.4	Ereignisse bei Fokussieren von Elementen	462

6.3.5	Allgemeine Ereignisse der Nutzerschnittstelle	463
6.3.6	Ereignisse bei mobilen Endgeräten	466
6.4	Den Ereignisfluss verstehen und beeinflussen	467
6.4.1	Die Event-Phasen	467
6.4.2	Den Ereignisfluss unterbrechen	475
6.4.3	Standardaktionen von Events verhindern	480
6.5	Ereignisse programmatisch auslösen	483
6.5.1	Einfache Ereignisse auslösen	483
6.5.2	Ereignisse mit übergebenen Argumenten auslösen	484
6.5.3	Standardereignisse auslösen	484
6.6	Zusammenfassung	485
7	Mit Formularen arbeiten	487
7.1	Auf Formulare und Formularfelder zugreifen	488
7.1.1	Auf Formulare zugreifen	488
7.1.2	Auf Formularelemente zugreifen	492
7.1.3	Den Wert von Textfeldern und Passwortfeldern auslesen	494
7.1.4	Den Wert von Checkboxen auslesen	495
7.1.5	Den Wert von Radiobuttons auslesen	495
7.1.6	Den Wert von Auswahllisten auslesen	497
7.1.7	Die Werte von Mehrfachauswahllisten auslesen	499
7.1.8	Auswahllisten per JavaScript mit Werten befüllen	500
7.2	Formulare programmatisch abschicken und zurücksetzen	501
7.3	Formulareingaben validieren	504
7.4	Zusammenfassung	515
8	Browser steuern und Browserinformationen auslesen	517
8.1	Das Browser Object Model	517
8.2	Auf Fensterinformationen zugreifen	519
8.2.1	Die Größe und Position eines Browserfensters ermitteln	519
8.2.2	Die Größe und Position eines Browserfensters ändern	521
8.2.3	Auf Anzeigeeigenschaften der Browserleisten zugreifen	523
8.2.4	Allgemeine Eigenschaften ermitteln	524
8.2.5	Neue Browserfenster öffnen	525

8.2.6	Browserfenster schließen	526
8.2.7	Dialoge öffnen	527
8.2.8	Funktionen zeitgesteuert ausführen	528
8.3	Auf Navigationsinformationen der aktuellen Webseite zugreifen	530
8.3.1	Auf die einzelnen Bestandteile der URL zugreifen	530
8.3.2	Auf Querystring-Parameter zugreifen	531
8.3.3	Eine neue Webseite laden	532
8.4	Den Browserverlauf einsehen und verändern	533
8.4.1	Im Browserverlauf navigieren	533
8.4.2	Browserverlauf bei Single Page Applications	534
8.4.3	Einträge in den Browserverlauf hinzufügen	535
8.4.4	Auf Änderungen im Browserverlauf reagieren	538
8.4.5	Den aktuellen Eintrag im Browserverlauf ersetzen	538
8.5	Browser erkennen und Browserfeatures bestimmen	540
8.6	Auf Informationen des Bildschirms zugreifen	542
8.7	Zusammenfassung	543
9	Inhalte einer Webseite dynamisch nachladen	545
9.1	Das Prinzip von Ajax	545
9.1.1	Synchrone Kommunikation	545
9.1.2	Asynchrone Kommunikation	546
9.1.3	Typische Anwendungsfälle für die Verwendung von Ajax	548
9.1.4	Verwendete Datenformate	551
9.2	Das XML-Format	551
9.2.1	Der Aufbau von XML	551
9.2.2	XML und die DOM API	553
9.2.3	Zeichenketten in XML-Objekte umwandeln	554
9.2.4	XML-Objekte in Zeichenketten umwandeln	555
9.3	Das JSON-Format	557
9.3.1	Der Aufbau von JSON	557
9.3.2	Unterschied zwischen JSON und JavaScript-Objekten	559
9.3.3	Objekte in das JSON-Format umwandeln	560
9.3.4	Objekte aus dem JSON-Format umwandeln	561
9.4	Anfragen per Ajax stellen	562
9.4.1	Das »XMLHttpRequest«-Objekt	562
9.4.2	HTML-Daten per Ajax laden	568

9.4.3	XML-Daten per Ajax laden	573
9.4.4	JSON-Daten per Ajax laden	576
9.4.5	Daten per Ajax an den Server schicken	579
9.4.6	Formulare per Ajax abschicken	580
9.4.7	Daten von anderen Domains laden	581
9.4.8	Die neuere Alternative zu »XMLHttpRequest«: die Fetch API	584
9.5	Zusammenfassung	588
10	Aufgaben vereinfachen mit jQuery	589
10.1	Einführung	589
10.1.1	jQuery einbinden	590
10.1.2	jQuery über ein CDN einbinden	591
10.1.3	jQuery verwenden	592
10.1.4	Aufgaben mit jQuery vereinfachen	593
10.2	Mit dem DOM arbeiten	595
10.2.1	Elemente selektieren	595
10.2.2	Auf Inhalte zugreifen und diese verändern	600
10.2.3	Ausgewählte Elemente filtern	603
10.2.4	Auf Attribute zugreifen	606
10.2.5	Auf CSS-Eigenschaften zugreifen	607
10.2.6	Zwischen Elementen navigieren	608
10.2.7	Effekte und Animationen verwenden	610
10.3	Auf Ereignisse reagieren	612
10.3.1	Event-Listener registrieren	612
10.3.2	Auf allgemeine Ereignisse reagieren	613
10.3.3	Auf Mausereignisse reagieren	614
10.3.4	Auf Tastaturereignisse reagieren	616
10.3.5	Auf Formularereignisse reagieren	617
10.3.6	Auf Informationen von Ereignissen zugreifen	618
10.4	Ajax-Anfragen erstellen	620
10.4.1	Ajax-Anfragen erstellen	620
10.4.2	Auf Ereignisse reagieren	623
10.4.3	HTML-Daten per Ajax laden	624
10.4.4	XML-Daten per Ajax laden	625
10.4.5	JSON-Daten per Ajax laden	626
10.5	Zusammenfassung	628

11 Bilder und Grafiken dynamisch erstellen	635
11.1 Bilder zeichnen	635
11.1.1 Die Zeichenfläche	635
11.1.2 Der Rendering-Kontext	636
11.1.3 Rechtecke zeichnen	638
11.1.4 Pfade verwenden	641
11.1.5 Texte zeichnen	647
11.1.6 Farbverläufe zeichnen	648
11.1.7 Speichern und Wiederherstellen des Canvas-Zustands	650
11.1.8 Transformationen anwenden	652
11.1.9 Animationen erstellen	655
11.2 Vektorgrafiken einbinden	657
11.2.1 Das SVG-Format	657
11.2.2 SVG in HTML einbinden	659
11.2.3 Das Aussehen von SVG-Elementen mit CSS beeinflussen	662
11.2.4 Das Verhalten von SVG-Elementen mit JavaScript beeinflussen	663
11.3 Zusammenfassung	665
12 Moderne Web-APIs verwenden	667
12.1 Über JavaScript kommunizieren	669
12.1.1 Unidirektionale Kommunikation mit dem Server	669
12.1.2 Bidirektionale Kommunikation mit einem Server	671
12.1.3 Vom Server ausgehende Kommunikation	673
12.2 Nutzer wiedererkennen	678
12.2.1 Cookies verwenden	678
12.2.2 Cookies anlegen	680
12.2.3 Cookies auslesen	681
12.2.4 Ein Beispiel: Einkaufswagen auf Basis von Cookies	683
12.2.5 Nachteile von Cookies	686
12.3 Den Browserspeicher nutzen	686
12.3.1 Werte im Browserspeicher speichern	687
12.3.2 Werte aus dem Browserspeicher lesen	688
12.3.3 Werte im Browserspeicher aktualisieren	689
12.3.4 Werte aus dem Browserspeicher löschen	689
12.3.5 Auf Änderungen im Browserspeicher reagieren	689

12.3.6 Die verschiedenen Typen von Browserspeichern	690
12.3.7 Ein Beispiel: Einkaufswagen auf Basis des Browserspeichers	692
12.4 Die Browserdatenbank nutzen	693
12.4.1 Öffnen einer Datenbank	694
12.4.2 Erstellen einer Datenbank	695
12.4.3 Erstellen eines Objektspeichers	696
12.4.4 Hinzufügen von Objekten zu einem Objektspeicher	697
12.4.5 Lesen von Objekten aus einem Objektspeicher	700
12.4.6 Löschen von Objekten aus einem Objektspeicher	701
12.4.7 Aktualisieren von Objekten in einem Objektspeicher	703
12.4.8 Verwendung eines Cursors	704
12.5 Auf das Dateisystem zugreifen	705
12.5.1 Auswählen von Dateien per Dateidialog	706
12.5.2 Auswählen von Dateien per Drag-and-drop	707
12.5.3 Lesen von Dateien	709
12.5.4 Den Lesefortschritt überwachen	711
12.6 Komponenten einer Webseite verschieben	713
12.6.1 Ereignisse einer Drag-and-drop-Operation	714
12.6.2 Verschiebbare Elemente definieren	715
12.6.3 Verschieben von Elementen	717
12.7 Aufgaben parallelisieren	719
12.7.1 Das Prinzip von Web Workern	720
12.7.2 Web Worker verwenden	721
12.8 Den Standort von Nutzern ermitteln	723
12.8.1 Auf Standortinformationen zugreifen	723
12.8.2 Kontinuierlich auf Standortinformationen zugreifen	726
12.8.3 Position auf Karte anzeigen	726
12.8.4 Anfahrtsbeschreibung anzeigen	728
12.9 Den Batteriestand eines Endgeräts auslesen	729
12.9.1 Auf Batterieinformationen zugreifen	730
12.9.2 Auf Ereignisse reagieren	731
12.10 Sprache ausgeben und Sprache erkennen	733
12.10.1 Sprache ausgeben	734
12.10.2 Sprache erkennen	736
12.11 Animationen erstellen	737
12.11.1 Verwendung der API	738
12.11.2 Steuern einer Animation	741

12.12 Mit der Kommandozeile arbeiten	741
12.12.1 Auswahl und Inspektion von DOM-Elementen	742
12.12.2 Analyse von Events	744
12.12.3 Debugging, Monitoring und Profiling	747
12.13 Mehrsprachige Anwendungen entwickeln	751
12.13.1 Begriffserklärungen	752
12.13.2 Die Internationalization API	753
12.13.3 Vergleich von Zeichenketten	755
12.13.4 Formatierung von Datums- und Zeitangaben	758
12.13.5 Formatierung von Zahlenwerten	761
12.14 Übersicht über verschiedene Web-APIs	764
12.15 Zusammenfassung	768
13 Objektorientierte Programmierung	771
<hr/>	
13.1 Die Prinzipien der objektorientierten Programmierung	771
13.1.1 Klassen, Objektinstanzen und Prototypen	772
13.1.2 Prinzip 1: Abstraktes Verhalten definieren	774
13.1.3 Prinzip 2: Zustand und Verhalten kapseln	775
13.1.4 Prinzip 3: Zustand und Verhalten vererben	776
13.1.5 Prinzip 4: Verschiedene Typen annehmen	777
13.1.6 JavaScript und die Objektorientierung	778
13.2 Prototypische Objektorientierung	778
13.2.1 Das Konzept von Prototypen	778
13.2.2 Von Objekten ableiten	779
13.2.3 Methoden und Eigenschaften vererben	780
13.2.4 Methoden und Eigenschaften im erbenden Objekt definieren	780
13.2.5 Methoden überschreiben	781
13.2.6 Die Prototypenkette	783
13.2.7 Methoden des Prototyps aufrufen	784
13.2.8 Prototypische Objektorientierung und die Prinzipien der Objektorientierung	785
13.3 Pseudoklassische Objektorientierung	786
13.3.1 Konstruktorfunktionen definieren	786
13.3.2 Objektinstanzen erzeugen	786
13.3.3 Methoden definieren	786
13.3.4 Von Objekten ableiten	787
13.3.5 Konstruktor der »Oberklasse« aufrufen	791
13.3.6 Methoden überschreiben	791

13.3.7 Methoden der »Oberklasse« aufrufen	792
13.3.8 Pseudoklassische Objektorientierung und die Prinzipien der Objektorientierung	792
13.4 Objektorientierung mit Klassensyntax	792
13.4.1 Klassen definieren	793
13.4.2 Objektinstanzen erzeugen	795
13.4.3 Getter und Setter definieren	795
13.4.4 Private Eigenschaften und private Methoden definieren	797
13.4.5 Von »Klassen« ableiten	800
13.4.6 Methoden überschreiben	805
13.4.7 Methoden der »Oberklasse« aufrufen	806
13.4.8 Statische Methoden definieren	807
13.4.9 Statische Eigenschaften definieren	809
13.4.10 Klassensyntax und die Prinzipien der Objektorientierung	811
13.5 Zusammenfassung	811
14 Funktionale Programmierung	813
<hr/>	
14.1 Prinzipien der funktionalen Programmierung	813
14.1.1 Prinzip 1: Funktionen sind Objekte erster Klasse	813
14.1.2 Prinzip 2: Funktionen arbeiten mit unveränderlichen Datenstrukturen	814
14.1.3 Prinzip 3: Funktionen haben keine Nebeneffekte	814
14.1.4 Prinzip 4: Funktionale Programme sind deklarativ	814
14.2 Imperative Programmierung und funktionale Programmierung	815
14.2.1 Iterieren mit der Methode »forEach()«	816
14.2.2 Werte abbilden mit der Methode »map()«	819
14.2.3 Werte filtern mit der Methode »filter()«	820
14.2.4 Mehrere Werte zu einem Wert reduzieren mit der Methode »reduce()«	822
14.2.5 Kombination der verschiedenen Methoden	824
14.3 Zusammenfassung	825
15 Den Quelltext richtig strukturieren	827
<hr/>	
15.1 Namenskonflikte vermeiden	827
15.2 Module definieren und verwenden	831
15.2.1 Das Module-Entwurfsmuster	832

15.2.2	Das Revealing-Module-Entwurfsmuster	835
15.2.3	AMD	840
15.2.4	CommonJS	841
15.2.5	Native Module	843
15.3	Zusammenfassung	846
16	Asynchrone Programmierung und weitere fortgeschrittene Features verwenden	849
<hr/>		
16.1	Asynchrone Programmierung verstehen und anwenden	849
16.1.1	Das Callback-Entwurfsmuster verwenden	850
16.1.2	Promises verwenden	854
16.1.3	Async Functions verwenden	863
16.2	Das Iterieren über Datenstrukturen kapseln	867
16.2.1	Das Prinzip von Iteratoren	867
16.2.2	Iteratoren verwenden	867
16.2.3	Einen eigenen Iterator erstellen	868
16.2.4	Ein iterierbares Objekt erstellen	869
16.2.5	Über iterierbare Objekte iterieren	871
16.3	Funktionen anhalten und fortsetzen	871
16.3.1	Eine Generatorfunktion erstellen	871
16.3.2	Einen Generator erstellen	872
16.3.3	Über Generatoren iterieren	873
16.3.4	Unendliche Generatoren erstellen	874
16.3.5	Generatoren mit Parametern steuern	874
16.4	Den Zugriff auf Objekte abfangen	875
16.4.1	Das Prinzip von Proxies	875
16.4.2	Proxies erstellen	876
16.4.3	Handler für Proxies definieren	876
16.5	Zusammenfassung	881
17	Serverseitige Anwendungen mit Node.js erstellen	883
<hr/>		
17.1	Einführung Node.js	883
17.1.1	Die Architektur von Node.js	883
17.1.2	Installation von Node.js	885

17.1.3	Eine einfache Anwendung	886
17.2	Node.js-Packages verwalten	887
17.2.1	Den Node.js Package Manager installieren	887
17.2.2	Packages installieren	887
17.2.3	Eigene Packages erstellen	891
17.3	Ereignisse verarbeiten und auslösen	895
17.3.1	Ein Event auslösen und abfangen	895
17.3.2	Ein Event mehrfach auslösen	897
17.3.3	Ein Event genau einmal abfangen	898
17.3.4	Ein Event mehrfach abfangen	898
17.4	Auf das Dateisystem zugreifen	899
17.4.1	Dateien lesen	899
17.4.2	Dateien schreiben	900
17.4.3	Dateiinformatioen auslesen	901
17.4.4	Dateien löschen	902
17.4.5	Mit Verzeichnissen arbeiten	903
17.5	Einen Webserver erstellen	904
17.5.1	Einen Webserver starten	904
17.5.2	Dateien per Webserver zur Verfügung stellen	906
17.5.3	Einen Client für einen Webserver erstellen	906
17.5.4	Routen definieren	907
17.5.5	Das Web-Framework Express.js verwenden	908
17.6	Auf Datenbanken zugreifen	913
17.6.1	Installation von MongoDB	913
17.6.2	MongoDB-Treiber für Node.js installieren	914
17.6.3	Verbindung zur Datenbank herstellen	914
17.6.4	Eine Collection erstellen	915
17.6.5	Objekte speichern	916
17.6.6	Objekte lesen	917
17.6.7	Objekte aktualisieren	919
17.6.8	Objekte löschen	920
17.7	Mit Streams arbeiten	921
17.7.1	Einführung und Arten von Streams	921
17.7.2	Anwendungsfälle von Streams	922
17.7.3	Daten mit Streams lesen	923
17.7.4	Daten mit Streams schreiben	925
17.7.5	Streams mithilfe von Piping kombinieren	926
17.7.6	Fehlerbehandlung beim Piping	928
17.8	Zusammenfassung	929

18 Mobile Anwendungen mit JavaScript erstellen	931
18.1 Die unterschiedlichen Arten mobiler Anwendungen	931
18.1.1 Native Anwendungen	931
18.1.2 Mobile Webanwendungen	932
18.1.3 Hybridanwendungen	934
18.1.4 Vergleich der verschiedenen Ansätze	935
18.2 Mobile Anwendungen mit React Native erstellen	937
18.2.1 Das Prinzip von React Native	938
18.2.2 Installation und Projektinitialisierung	938
18.2.3 Die Anwendung starten	940
18.2.4 Das Grundgerüst einer React-Native-Anwendung	942
18.2.5 UI-Komponenten verwenden	944
18.2.6 Kommunikation mit dem Server	948
18.2.7 Bauen und Veröffentlichen von Anwendungen	950
18.3 Zusammenfassung	950
19 Desktopanwendungen mit JavaScript	951
19.1 NW.js	952
19.1.1 Installation und Erstellen einer Anwendung	953
19.1.2 Starten der Anwendung	955
19.1.3 Packaging der Anwendung	955
19.1.4 Weitere Beispielanwendungen	956
19.2 Electron	957
19.2.1 Installation und Erstellen einer Anwendung	958
19.2.2 Starten der Anwendung	960
19.2.3 Packaging	961
19.2.4 Weitere Beispielanwendungen	962
19.3 Zusammenfassung	963
20 Mikrocontroller mit JavaScript steuern	965
20.1 Espruino	966
20.1.1 Technische Informationen	966
20.1.2 Anschluss und Installation	967

20.1.3 Erstes Beispiel	967
20.1.4 LEDs ansteuern	968
20.1.5 Weitere Module	970
20.1.6 Sensoren auslesen	971
20.2 Tessel	972
20.2.1 Technische Informationen	972
20.2.2 Anschluss und Installation	973
20.2.3 LEDs ansteuern	973
20.2.4 Die Drucktaster programmieren	975
20.2.5 Den Tessel durch Module erweitern	976
20.3 BeagleBone Black	977
20.3.1 Technische Informationen	977
20.3.2 Anschluss und Installation	978
20.3.3 LEDs ansteuern	979
20.4 Arduino	980
20.4.1 Das Firmata-Protokoll	981
20.4.2 Anschluss und Installation	981
20.4.3 Das Node.js-Modul Johnny Five	982
20.5 Cylon.js	983
20.5.1 Steuern eines BeagleBone Black mit Cylon.js	984
20.5.2 Steuern eines Tessel-Boards mit Cylon.js	984
20.5.3 Steuern eines Arduino mit Cylon.js	985
20.6 Zusammenfassung	985
21 Einen professionellen Entwicklungsprozess aufsetzen	987
21.1 Aufgaben automatisieren	987
21.1.1 Aufgaben automatisieren mit Grunt	988
21.1.2 Aufgaben automatisieren mit Gulp	991
21.2 Quelltext automatisiert testen	992
21.2.1 Das Prinzip von automatisierten Tests	993
21.2.2 Das Prinzip der testgetriebenen Entwicklung	994
21.2.3 Den Quelltext automatisiert testen mit QUnit	996
21.2.4 Den Quelltext automatisiert testen mit mocha	1002
21.3 Versionsverwaltung des Quelltexts	1006
21.3.1 Einführung in die Versionsverwaltung	1006
21.3.2 Das Versionsverwaltungssystem Git installieren und konfigurieren	1010

21.3.3	Ein neues lokales Repository anlegen	1012
21.3.4	Ein bestehendes Repository klonen	1012
21.3.5	Änderungen in den Staging-Bereich übertragen	1013
21.3.6	Änderungen in das lokale Repository übertragen	1013
21.3.7	Die verschiedenen Zustände in Git	1015
21.3.8	Änderungen in das Remote Repository übertragen	1016
21.3.9	Änderungen aus dem Remote Repository übertragen	1017
21.3.10	In einem neuen Branch arbeiten	1018
21.3.11	Änderungen aus einem Branch übernehmen	1019
21.3.12	Übersicht über die wichtigsten Befehle und Begriffe	1020
21.4	Zusammenfassung	1024

Anhang 1025

A	JavaScript-Referenz	1027
B	DOM-Referenz und HTML-Erweiterungen	1087
C	BOM und Ajax	1175
D	HTML5-Web-APIs-Referenz	1209
Index		1257