

Auf einen Blick

1	Grundlagen und Einführung	29
2	Erste Schritte	55
3	Sprachkern	89
4	Mit Objekten und Referenztypen arbeiten	229
5	Webseiten dynamisch verändern	339
6	Ereignisse verarbeiten und auslösen	397
7	Mit Formularen arbeiten	449
8	Browser steuern und Browserinformationen auslesen	477
9	Inhalte einer Webseite dynamisch nachladen	505
10	Aufgaben vereinfachen mit jQuery	549
11	Bilder und Grafiken dynamisch erstellen	597
12	Moderne Web-APIs verwenden	629
13	Objektorientierte Programmierung	733
14	Funktionale Programmierung	767
15	Den Quelltext richtig strukturieren	781
16	Die seit ES2015 eingeführten Features richtig nutzen	801
17	Serverseitige Anwendungen mit Node.js erstellen	867
18	Mobile Anwendungen mit JavaScript erstellen	907
19	Desktopanwendungen mit JavaScript	965
20	Mikrocontroller mit JavaScript steuern	979
21	Einen professionellen Entwicklungsprozess aufsetzen	1001

Inhalt

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 JavaScript	44
1.2.1 Historie	45
1.2.2 Anwendungsgebiete	46
1.3 Zusammenfassung	53
2 Erste Schritte	55
2.1 Einführung JavaScript und 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	64
2.2.1 Eine geeignete Ordnerstruktur vorbereiten	64
2.2.2 Eine JavaScript-Datei erstellen	65
2.2.3 Eine JavaScript-Datei in eine HTML-Datei einbinden	66
2.2.4 JavaScript direkt innerhalb des HTML definieren	69
2.2.5 Platzierung und Ausführung der <script>-Elemente	70
2.2.6 Den Quelltext anzeigen	74
2.3 Eine Ausgabe erzeugen	77
2.3.1 Standarddialogfenster anzeigen	77
2.3.2 Auf die Konsole schreiben	78
2.3.3 Bestehende UI-Komponenten verwenden	85
2.4 Zusammenfassung	86

3 Sprachkern	89
3.1 Werte in Variablen speichern	89
3.1.1 Variablen definieren	89
3.1.2 Gültige Variablennamen verwenden	92
3.1.3 Konstanten definieren	99
3.2 Die verschiedenen Datentypen verwenden	100
3.2.1 Zahlen	101
3.2.2 Zeichenketten	103
3.2.3 Boolesche Wahrheitswerte	106
3.2.4 Arrays	106
3.2.5 Objekte	112
3.2.6 Besondere Datentypen	113
3.3 Die verschiedenen Operatoren einsetzen	115
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	119
3.3.4 Operatoren für das Arbeiten mit booleschen Werten	120
3.3.5 Operatoren für das Arbeiten mit Bits	126
3.3.6 Operatoren für das Vergleichen von Werten	127
3.3.7 Operatoren für spezielle Operationen	129
3.4 Den Ablauf eines Programms steuern	130
3.4.1 Bedingte Anweisungen definieren	131
3.4.2 Verzweigungen definieren	133
3.4.3 Den Auswahloperator verwenden	139
3.4.4 Mehrfachverzweigungen definieren	140
3.4.5 Zählschleifen definieren	147
3.4.6 Kopfgesteuerte Schleifen definieren	155
3.4.7 Fußgesteuerte Schleifen definieren	158
3.4.8 Schleifen und Schleifeniterationen vorzeitig abbrechen	159
3.5 Wiederverwendbare Codebausteine erstellen	168
3.5.1 Funktionen definieren	168
3.5.2 Funktionen aufrufen	171
3.5.3 Funktionsparameter übergeben und auswerten	171
3.5.4 Rückgabewerte definieren	180
3.5.5 Standardwerte für Parameter definieren	182
3.5.6 Elemente aus einem Array als Parameter verwenden	184
3.5.7 Funktionen über Kurzschreibweise definieren	186
3.5.8 Funktionen im Detail	188
3.5.9 Funktionen aufrufen durch Nutzerinteraktion	196

3.6 Auf Fehler reagieren und sie richtig behandeln	198
3.6.1 Syntaxfehler	198
3.6.2 Laufzeitfehler	199
3.6.3 Logische Fehler	200
3.6.4 Das Prinzip der Fehlerbehandlung	201
3.6.5 Fehler fangen und behandeln	202
3.6.6 Fehler auslösen	205
3.6.7 Fehler und der Funktionsaufruf-Stack	208
3.6.8 Bestimmte Anweisungen unabhängig von aufgetretenen Fehlern aufrufen	210
3.7 Den Quelltext kommentieren	217
3.8 Den Code debuggen	217
3.8.1 Einführung	218
3.8.2 Ein einfaches Codebeispiel	218
3.8.3 Haltepunkte definieren	219
3.8.4 Variablenbelegungen einsehen	221
3.8.5 Ein Programm schrittweise ausführen	222
3.8.6 Mehrere Haltepunkte definieren	224
3.8.7 Bedingte Haltepunkte definieren	224
3.8.8 Den Funktionsaufruf-Stack einsehen	225
3.9 Zusammenfassung	226
4 Mit Objekten und Referenztypen arbeiten	229
4.1 Unterschied zwischen primitiven Datentypen und Referenztypen	229
4.1.1 Das Prinzip von primitiven Datentypen	229
4.1.2 Das Prinzip von Referenztypen	230
4.1.3 Primitive Datentypen und Referenztypen als Funktionsargumente	232
4.1.4 Den Typ einer Variablen ermitteln	233
4.1.5 Ausblick	236
4.2 Zustand und Verhalten in Objekten kapseln	236
4.2.1 Einführung objektorientierte Programmierung	237
4.2.2 Objekte erstellen über die Literal-Schreibweise	238
4.2.3 Objekte erstellen über Konstruktorfunktionen	239
4.2.4 Objekte erstellen unter Verwendung von Klassen	242
4.2.5 Objekte erstellen über die Funktion »Object.create()«	246
4.2.6 Auf Eigenschaften zugreifen und Methoden aufrufen	250

4.2.7	Objekteigenschaften und Objektmethoden hinzufügen oder überschreiben	256
4.2.8	Objekteigenschaften und Objektmethoden löschen	260
4.2.9	Objekteigenschaften und Objektmethoden ausgeben	263
4.2.10	Änderungen an Objekten verhindern	266
4.3	Mit Arrays arbeiten	270
4.3.1	Arrays erzeugen und initialisieren	270
4.3.2	Auf Elemente eines Arrays zugreifen	273
4.3.3	Elemente einem Array hinzufügen	274
4.3.4	Elemente aus einem Array entfernen	279
4.3.5	Einen Teil der Elemente aus einem Array kopieren	282
4.3.6	Arrays sortieren	285
4.3.7	Arrays als Stack verwenden	288
4.3.8	Arrays als Queue verwenden	289
4.3.9	Elemente in Arrays finden	291
4.3.10	Elemente innerhalb eines Arrays kopieren	293
4.3.11	Arrays in Zeichenketten umwandeln	294
4.4	Mit Zeichenketten arbeiten	295
4.4.1	Der Aufbau einer Zeichenkette	295
4.4.2	Die Länge einer Zeichenkette ermitteln	296
4.4.3	Innerhalb einer Zeichenkette suchen	297
4.4.4	Teile einer Zeichenkette extrahieren	300
4.5	Sonstige globale Objekte	303
4.5.1	Mit Datum und Zeit arbeiten	303
4.5.2	Komplexe Berechnungen durchführen	306
4.5.3	Wrapperobjekte für primitive Datentypen	307
4.6	Mit regulären Ausdrücken arbeiten	307
4.6.1	Reguläre Ausdrücke definieren	308
4.6.2	Zeichen gegen einen regulären Ausdruck testen	309
4.6.3	Zeichenklassen verwenden	311
4.6.4	Anfang und Ende begrenzen	315
4.6.5	Quantifizierer verwenden	318
4.6.6	Nach Vorkommen suchen	322
4.6.7	Alle Vorkommen innerhalb einer Zeichenkette suchen	324
4.6.8	Auf einzelne Teile eines Vorkommens zugreifen	325
4.6.9	Nach bestimmten Zeichenketten suchen	326
4.6.10	Vorkommen innerhalb einer Zeichenkette ersetzen	327
4.6.11	Nach Vorkommen suchen	327
4.6.12	Zeichenketten zerteilen	328

4.7	Funktionen als Referenztypen	329
4.7.1	Funktionen als Argumente verwenden	329
4.7.2	Funktionen als Rückgabewert verwenden	332
4.7.3	Standardmethoden jeder Funktion	333
4.8	Zusammenfassung	337
5	Webseiten dynamisch verändern	339
5.1	Aufbau einer Webseite	339
5.1.1	Document Object Model	339
5.1.2	Die verschiedenen Knotentypen	340
5.1.3	Der Dokumentknoten	344
5.2	Elemente selektieren	345
5.2.1	Elemente per ID selektieren	347
5.2.2	Elemente per Klasse selektieren	350
5.2.3	Elemente nach Elementnamen selektieren	353
5.2.4	Elemente nach Namen selektieren	355
5.2.5	Elemente per Selektor selektieren	357
5.2.6	Das Elternelement eines Elements selektieren	363
5.2.7	Die Kindelemente eines Elements selektieren	365
5.2.8	Die Geschwisterelemente eines Elements selektieren	370
5.2.9	Selektionsmethoden auf Elementen aufrufen	372
5.2.10	Elemente nach Typ selektieren	375
5.3	Mit Textknoten arbeiten	375
5.3.1	Auf den Textinhalt eines Elements zugreifen	376
5.3.2	Den Textinhalt eines Elements verändern	377
5.3.3	Das HTML unterhalb eines Elements verändern	378
5.3.4	Textknoten erstellen und hinzufügen	379
5.4	Mit Elementen arbeiten	379
5.4.1	Elemente erstellen und hinzufügen	380
5.4.2	Elemente und Knoten entfernen	383
5.4.3	Die verschiedenen Typen von HTML-Elementen	384
5.5	Mit Attributen arbeiten	389
5.5.1	Den Wert eines Attributs auslesen	390
5.5.2	Den Wert eines Attributs ändern oder ein neues Attribut hinzufügen	392
5.5.3	Attributknoten erstellen und hinzufügen	392

5.5.4	Attribute entfernen	393
5.5.5	Auf CSS-Klassen zugreifen	393
5.6	Zusammenfassung	394
6	Ereignisse verarbeiten und auslösen	397
6.1	Das Konzept der ereignisgesteuerten Programmierung	397
6.2	Auf Ereignisse reagieren	398
6.2.1	Einen Event-Handler per HTML definieren	401
6.2.2	Einen Event-Handler per JavaScript definieren	403
6.2.3	Event-Listener definieren	405
6.2.4	Mehrere Event-Listener definieren	407
6.2.5	Argumente an Event-Listener übergeben	409
6.2.6	Event-Listener entfernen	411
6.2.7	Event-Handler und Event-Listener per Helferfunktion definieren	412
6.2.8	Auf Informationen eines Ereignisses zugreifen	413
6.3	Die verschiedenen Typen von Ereignissen	415
6.3.1	Ereignisse bei Interaktion mit der Maus	416
6.3.2	Ereignisse bei Interaktion mit Tastatur und Textfeldern	421
6.3.3	Ereignisse beim Arbeiten mit Formularen	424
6.3.4	Ereignisse bei Fokussieren von Elementen	424
6.3.5	Allgemeine Ereignisse der Nutzerschnittstelle	425
6.3.6	Ereignisse bei mobilen Endgeräten	428
6.4	Den Ereignisfluss verstehen und beeinflussen	429
6.4.1	Die Event-Phasen	429
6.4.2	Den Ereignisfluss unterbrechen	437
6.4.3	Standardaktionen von Events verhindern	442
6.5	Ereignisse programmatisch auslösen	445
6.5.1	Einfache Ereignisse auslösen	445
6.5.2	Ereignisse mit übergebenen Argumenten auslösen	446
6.5.3	Standardereignisse auslösen	446
6.6	Zusammenfassung	447

7	Mit Formularen arbeiten	449
7.1	Auf Formulare und Formularfelder zugreifen	450
7.1.1	Auf Formulare zugreifen	450
7.1.2	Auf Formularelemente zugreifen	453
7.1.3	Den Wert von Textfeldern und Passwortfeldern auslesen	455
7.1.4	Den Wert von Checkboxes auslesen	456
7.1.5	Den Wert von Radiobuttons auslesen	457
7.1.6	Den Wert von Auswahllisten auslesen	459
7.1.7	Die Werte von Mehrfachauswahllisten auslesen	460
7.1.8	Auswahllisten per JavaScript mit Werten befüllen	462
7.2	Formulare programmatisch abschicken und zurücksetzen	463
7.3	Formulareingaben validieren	465
7.4	Zusammenfassung	475
8	Browser steuern und Browserinformationen auslesen	477
8.1	Das Browser Object Model	477
8.2	Auf Fensterinformationen zugreifen	479
8.2.1	Die Größe und Position eines Browserfensters ermitteln	479
8.2.2	Die Größe und Position eines Browserfensters ändern	481
8.2.3	Auf Anzeigeinformationen der Browserleisten zugreifen	482
8.2.4	Allgemeine Eigenschaften ermitteln	484
8.2.5	Neue Browserfenster öffnen	484
8.2.6	Browserfenster schließen	486
8.2.7	Dialoge öffnen	487
8.2.8	Funktionen zeitgesteuert ausführen	488
8.3	Auf Navigationsinformationen der aktuellen Webseite zugreifen	490
8.3.1	Auf die einzelnen Bestandteile der URL zugreifen	490
8.3.2	Auf Querystring-Parameter zugreifen	491
8.3.3	Eine neue Webseite laden	491
8.4	Den Browserverlauf einsehen und verändern	493
8.4.1	Im Browserverlauf navigieren	493
8.4.2	Browserverlauf bei Single Page Applications	494
8.4.3	Einträge in den Browserverlauf hinzufügen	495
8.4.4	Auf Änderungen im Browserverlauf reagieren	498
8.4.5	Den aktuellen Eintrag im Browserverlauf ersetzen	498

8.5	Browser erkennen und Browserfeatures bestimmen	500
8.6	Auf Informationen des Bildschirms zugreifen	502
8.7	Zusammenfassung	504
9	Inhalte einer Webseite dynamisch nachladen	505
9.1	Das Prinzip von Ajax	505
9.1.1	Synchrone Kommunikation	505
9.1.2	Asynchrone Kommunikation	506
9.1.3	Typische Anwendungsfälle für die Verwendung von Ajax	508
9.1.4	Verwendete Datenformate	510
9.2	Das XML-Format	511
9.2.1	Der Aufbau von XML	511
9.2.2	XML und die DOM API	513
9.2.3	Zeichenketten in XML-Objekte umwandeln	514
9.2.4	XML-Objekte in Zeichenketten umwandeln	515
9.3	Das JSON-Format	516
9.3.1	Der Aufbau von JSON	517
9.3.2	Unterschied zwischen JSON und JavaScript-Objekten	519
9.3.3	Objekte in das JSON-Format umwandeln	519
9.3.4	Objekte aus dem JSON-Format umwandeln	521
9.4	Anfragen per Ajax stellen	522
9.4.1	Das »XMLHttpRequest«-Objekt	522
9.4.2	HTML-Daten per Ajax laden	529
9.4.3	XML-Daten per Ajax laden	533
9.4.4	JSON-Daten per Ajax laden	537
9.4.5	Daten per Ajax an den Server schicken	540
9.4.6	Formulare per Ajax abschicken	541
9.4.7	Daten von anderen Domains laden	542
9.4.8	Die neuere Alternative zu »XMLHttpRequest«: die Fetch API	545
9.5	Zusammenfassung	546

10	Aufgaben vereinfachen mit jQuery	549
10.1	Einführung	549
10.1.1	jQuery einbinden	550
10.1.2	jQuery über ein CDN einbinden	551
10.1.3	jQuery verwenden	552
10.1.4	Aufgaben mit jQuery vereinfachen	553
10.2	Mit dem DOM arbeiten	555
10.2.1	Elemente selektieren	555
10.2.2	Auf Inhalte zugreifen und diese verändern	560
10.2.3	Ausgewählte Elemente filtern	564
10.2.4	Auf Attribute zugreifen	566
10.2.5	Auf CSS-Eigenschaften zugreifen	568
10.2.6	Zwischen Elementen navigieren	569
10.2.7	Effekte und Animationen verwenden	571
10.3	Auf Ereignisse reagieren	573
10.3.1	Event-Listener registrieren	573
10.3.2	Auf allgemeine Ereignisse reagieren	574
10.3.3	Auf Mausereignisse reagieren	575
10.3.4	Auf Tastaturereignisse reagieren	577
10.3.5	Auf Formularereignisse reagieren	578
10.3.6	Auf Informationen von Ereignissen zugreifen	579
10.4	Ajax-Anfragen erstellen	581
10.4.1	Ajax-Anfragen erstellen	581
10.4.2	Auf Ereignisse reagieren	584
10.4.3	HTML-Daten per Ajax laden	585
10.4.4	XML-Daten per Ajax laden	587
10.4.5	JSON-Daten per Ajax laden	588
10.5	Zusammenfassung	589
11	Bilder und Grafiken dynamisch erstellen	597
11.1	Bilder zeichnen	597
11.1.1	Die Zeichenfläche	597
11.1.2	Der Rendering-Kontext	598
11.1.3	Rechtecke zeichnen	600
11.1.4	Pfade verwenden	603
11.1.5	Texte zeichnen	609

11.1.6	Farbverläufe zeichnen	610
11.1.7	Speichern und Wiederherstellen des Canvas-Zustands	612
11.1.8	Transformationen anwenden	614
11.1.9	Animationen erstellen	617
11.2	Vektorgrafiken einbinden	619
11.2.1	Das SVG-Format	619
11.2.2	SVG in HTML einbinden	621
11.2.3	Das Aussehen von SVG-Elementen mit CSS beeinflussen	624
11.2.4	Das Verhalten von SVG-Elementen mit JavaScript beeinflussen	625
11.3	Zusammenfassung	627
12	Moderne Web-APIs verwenden	629
12.1	Über JavaScript kommunizieren	631
12.1.1	Unidirektionale Kommunikation mit dem Server	631
12.1.2	Bidirektionale Kommunikation mit einem Server	633
12.1.3	Vom Server ausgehende Kommunikation	635
12.2	Nutzer wiedererkennen	640
12.2.1	Cookies verwenden	640
12.2.2	Cookies anlegen	642
12.2.3	Cookies auslesen	643
12.2.4	Ein Beispiel: Einkaufswagen auf Basis von Cookies	645
12.2.5	Nachteile von Cookies	648
12.3	Den Browserspeicher nutzen	648
12.3.1	Werte im Browserspeicher speichern	649
12.3.2	Werte aus dem Browserspeicher lesen	650
12.3.3	Werte im Browserspeicher aktualisieren	651
12.3.4	Werte aus dem Browserspeicher löschen	651
12.3.5	Auf Änderungen im Browserspeicher reagieren	652
12.3.6	Die verschiedenen Typen von Browserspeichern	653
12.3.7	Ein Beispiel: Einkaufswagen auf Basis des Browserspeichers	654
12.4	Die Browserdatenbank nutzen	655
12.4.1	Öffnen einer Datenbank	656
12.4.2	Erstellen einer Datenbank	658
12.4.3	Erstellen eines Objektspeichers	659
12.4.4	Hinzufügen von Objekten zu einem Objektspeicher	659
12.4.5	Lesen von Objekten aus einem Objektspeicher	663
12.4.6	Löschen von Objekten aus einem Objektspeicher	664

12.4.7	Aktualisieren von Objekten in einem Objektspeicher	665
12.4.8	Verwendung eines Cursors	666
12.5	Auf das Dateisystem zugreifen	668
12.5.1	Auswählen von Dateien per Dateidialog	668
12.5.2	Auswählen von Dateien per Drag & Drop	670
12.5.3	Lesen von Dateien	671
12.5.4	Den Lesefortschritt überwachen	674
12.6	Komponenten einer Webseite verschieben	676
12.6.1	Ereignisse einer Drag-and-Drop-Operation	676
12.6.2	Verschiebbare Elemente definieren	677
12.6.3	Verschieben von Elementen	680
12.7	Aufgaben parallelisieren	681
12.7.1	Das Prinzip von Web Workern	683
12.7.2	Web Worker verwenden	684
12.8	Den Standort von Nutzern ermitteln	685
12.8.1	Auf Standortinformationen zugreifen	686
12.8.2	Kontinuierlich auf Standortinformationen zugreifen	688
12.8.3	Position auf Karte anzeigen	689
12.8.4	Anfahrtsbeschreibung anzeigen	690
12.9	Den Batteriestand eines Endgeräts auslesen	692
12.9.1	Auf Batterieinformationen zugreifen	692
12.9.2	Auf Ereignisse reagieren	693
12.10	Sprache ausgeben und Sprache erkennen	695
12.10.1	Sprache ausgeben	696
12.10.2	Sprache erkennen	698
12.11	Animationen erstellen	700
12.11.1	Verwendung der API	700
12.11.2	Steuern einer Animation	703
12.12	Mit der Kommandozeile arbeiten	704
12.12.1	Auswahl und Inspektion von DOM-Elementen	705
12.12.2	Analyse von Events	707
12.12.3	Debugging, Monitoring und Profiling	710
12.13	Mehrsprachige Anwendungen entwickeln	714
12.13.1	Begriffserklärungen	715
12.13.2	Die Internationalization API	716
12.13.3	Vergleich von Zeichenketten	718
12.13.4	Formatierung von Datums- und Zeitangaben	721
12.13.5	Formatierung von Zahlenwerten	724

12.14 Übersicht über verschiedene Web-APIs	727
12.15 Zusammenfassung	732
13 Objektorientierte Programmierung	733
13.1 Die Prinzipien der objektorientierten Programmierung	733
13.1.1 Klassen, Objektinstanzen und Prototypen	734
13.1.2 Prinzip 1: Abstraktes Verhalten definieren	736
13.1.3 Prinzip 2: Zustand und Verhalten kapseln	737
13.1.4 Prinzip 3: Zustand und Verhalten vererben	738
13.1.5 Prinzip 4: Verschiedene Typen annehmen	739
13.1.6 JavaScript und die Objektorientierung	740
13.2 Prototypische Objektorientierung	740
13.2.1 Das Konzept von Prototypen	740
13.2.2 Von Objekten ableiten	741
13.2.3 Methoden und Eigenschaften vererben	742
13.2.4 Methoden und Eigenschaften im erbenden Objekt definieren	742
13.2.5 Methoden überschreiben	743
13.2.6 Die Prototypenkette	744
13.2.7 Methoden des Prototyps aufrufen	746
13.2.8 Prototypische Objektorientierung und die Prinzipien der Objektorientierung	747
13.3 Pseudoklassische Objektorientierung	747
13.3.1 Konstrukturfunktionen definieren	748
13.3.2 Objektinstanzen erzeugen	748
13.3.3 Methoden definieren	748
13.3.4 Von Objekten ableiten	749
13.3.5 Konstruktor der »Oberklasse« aufrufen	753
13.3.6 Methoden überschreiben	753
13.3.7 Methoden der »Oberklasse« aufrufen	753
13.3.8 Pseudoklassische Objektorientierung und die Prinzipien der Objektorientierung	754
13.4 Objektorientierung mit Klassensyntax	754
13.4.1 Klassen definieren	755
13.4.2 Objektinstanzen erzeugen	756
13.4.3 Getter und Setter definieren	756

13.4.4 Von »Klassen« ableiten	757
13.4.5 Methoden überschreiben	760
13.4.6 Methoden der »Oberklasse« aufrufen	762
13.4.7 Statische Methoden definieren	763
13.4.8 Statische Eigenschaften definieren	764
13.4.9 Klassensyntax und die Prinzipien der Objektorientierung	765
13.5 Zusammenfassung	766
14 Funktionale Programmierung	767
14.1 Prinzipien der funktionalen Programmierung	767
14.1.1 Prinzip 1: Funktionen sind Objekte erster Klasse	767
14.1.2 Prinzip 2: Funktionen arbeiten mit unveränderlichen Datenstrukturen	768
14.1.3 Prinzip 3: Funktionen haben keine Nebeneffekte	768
14.1.4 Prinzip 4: Funktionale Programme sind deklarativ	768
14.2 Imperative Programmierung und funktionale Programmierung	769
14.2.1 Iterieren mit der Methode »forEach()«	769
14.2.2 Werte abbilden mit der Methode »map()«	772
14.2.3 Werte filtern mit der Methode »filter()«	774
14.2.4 Mehrere Werte zu einem Wert reduzieren mit der Methode »reduce()«	776
14.2.5 Kombination der verschiedenen Methoden	778
14.3 Zusammenfassung	779
15 Den Quelltext richtig strukturieren	781
15.1 Namenskonflikte vermeiden	781
15.2 Module definieren und verwenden	785
15.2.1 Das Module-Entwurfsmuster	785
15.2.2 Das Revealing-Module-Entwurfsmuster	789
15.2.3 AMD	793
15.2.4 CommonJS	795
15.2.5 Native Module	796
15.3 Zusammenfassung	799

16 Die seit ES2015 eingeführten Features richtig nutzen	801
16.1 Maps verwenden	804
16.1.1 Maps erstellen	804
16.1.2 Grundlegende Operationen	805
16.1.3 Über Maps iterieren	807
16.1.4 Weak Maps verwenden	809
16.2 Sets verwenden	811
16.2.1 Sets erstellen	811
16.2.2 Grundlegende Operationen von Sets	812
16.2.3 Über Sets iterieren	813
16.2.4 Weak Sets verwenden	814
16.3 Das Iterieren über Datenstrukturen kapseln	816
16.3.1 Das Prinzip von Iteratoren	816
16.3.2 Iteratoren verwenden	816
16.3.3 Einen eigenen Iterator erstellen	817
16.3.4 Ein iterierbares Objekt erstellen	819
16.3.5 Über iterierbare Objekte iterieren	820
16.4 Funktionen anhalten und fortsetzen	820
16.4.1 Eine Generatorfunktion erstellen	821
16.4.2 Einen Generator erstellen	821
16.4.3 Über Generatoren iterieren	822
16.4.4 Unendliche Generatoren erstellen	823
16.4.5 Generatoren mit Parametern steuern	823
16.5 Den Zugriff auf Objekte abfangen	824
16.5.1 Das Prinzip von Proxies	824
16.5.2 Proxies erstellen	825
16.5.3 Handler für Proxies definieren	825
16.6 Asynchrone Programmierung vereinfachen	828
16.6.1 Das Prinzip der asynchronen Programmierung	828
16.6.2 Promises erstellen	833
16.6.3 Verarbeiten eines Promises	834
16.6.4 Promise-Aufrufe verketteten	835
16.6.5 Die Zustände von Promises	836
16.6.6 Async Functions	837
16.7 Vorlagen für Zeichenketten definieren	840
16.7.1 Template-Strings erstellen	841
16.7.2 Platzhalter innerhalb von Zeichenketten definieren	841
16.7.3 Ausdrücke innerhalb von Zeichenketten auswerten	841

16.7.4 Mehrzeilige Zeichenketten definieren	842
16.7.5 Zeichenketten über Funktionen verändern	843
16.8 Symbole verwenden	844
16.8.1 Symbole erstellen	844
16.8.2 Die Symbol-Registry verwenden	845
16.8.3 Symbole zur Definition eindeutiger Objekteigenschaften verwenden	846
16.8.4 Symbole zur Definition von Konstanten verwenden	848
16.9 Werte aus Arrays und Objekten extrahieren	848
16.9.1 Werte aus Arrays extrahieren	849
16.9.2 Werte aus Objekten extrahieren	852
16.9.3 Werte innerhalb einer Schleife extrahieren	856
16.9.4 Argumente einer Funktion extrahieren	857
16.9.5 Objekteigenschaften in ein anderes Objekt kopieren	859
16.9.6 Objekteigenschaften aus einem anderen Objekt kopieren	860
16.10 Neue Methoden der Standardobjekte	861
16.10.1 Neue Methoden in »Object«	861
16.10.2 Neue Methoden in »String«	861
16.10.3 Neue Methoden in »Array«	862
16.10.4 Neue Methoden in »RegExp«	863
16.10.5 Neue Methoden in »Number«	864
16.10.6 Neue Methoden in »Math«	864
16.11 Zusammenfassung	865
17 Serverseitige Anwendungen mit Node.js erstellen	867
17.1 Einführung Node.js	867
17.1.1 Die Architektur von Node.js	867
17.1.2 Installation von Node.js	869
17.1.3 Eine einfache Anwendung	870
17.2 Node.js Packages verwalten	871
17.2.1 Den Node.js Package Manager installieren	871
17.2.2 Packages installieren	871
17.2.3 Eigene Packages erstellen	875
17.3 Ereignisse verarbeiten und auslösen	879
17.3.1 Ein Event auslösen und abfangen	879
17.3.2 Ein Event mehrfach auslösen	881
17.3.3 Ein Event genau einmal abfangen	882
17.3.4 Ein Event mehrfach abfangen	882

17.4 Auf das Dateisystem zugreifen	883
17.4.1 Dateien lesen	883
17.4.2 Dateien schreiben	884
17.4.3 Dateiinformationen auslesen	885
17.4.4 Dateien löschen	886
17.4.5 Mit Verzeichnissen arbeiten	887
17.5 Einen Webserver erstellen	888
17.5.1 Einen Webserver starten	888
17.5.2 Dateien per Webserver zur Verfügung stellen	890
17.5.3 Einen Client für einen Webserver erstellen	890
17.5.4 Routen definieren	891
17.5.5 Das Webframework Express verwenden	892
17.6 Auf Datenbanken zugreifen	897
17.6.1 Installation von MongoDB	897
17.6.2 MongoDB-Treiber für Node.js installieren	898
17.6.3 Verbindung zur Datenbank herstellen	898
17.6.4 Eine Collection erstellen	899
17.6.5 Objekte speichern	900
17.6.6 Objekte lesen	901
17.6.7 Objekte aktualisieren	903
17.6.8 Objekte löschen	904
17.7 Zusammenfassung	905

18 Mobile Anwendungen mit JavaScript erstellen 907

18.1 Die unterschiedlichen Arten mobiler Anwendungen	907
18.1.1 Native Anwendungen	907
18.1.2 Mobile Webanwendungen	908
18.1.3 Hybridanwendungen	910
18.1.4 Vergleich der verschiedenen Ansätze	911
18.2 Mobile Anwendungen mit jQuery Mobile erstellen	913
18.2.1 Das Grundgerüst einer mobilen Anwendung definieren	914
18.2.2 Einzelne Seiten innerhalb einer Anwendung definieren	915
18.2.3 Übergänge zwischen den Seiten definieren	918
18.2.4 Themes verwenden	920
18.2.5 UI-Komponenten verwenden	921
18.2.6 Layout-Raster definieren	930
18.2.7 Auf Ereignisse reagieren	934

18.3 Hybride Anwendungen mit Cordova erstellen	936
18.3.1 Das Prinzip von Cordova	936
18.3.2 Eine Anwendung erstellen	937
18.3.3 Eine Anwendung starten	940
18.3.4 Plugins verwenden	942
18.3.5 Auf Geräteinformationen zugreifen	945
18.3.6 Dialoge anzeigen	947
18.3.7 Auf die Kamera zugreifen	948
18.3.8 Auf Bewegungsinformationen zugreifen	949
18.3.9 Auf Orientierungsinformationen zugreifen	950
18.3.10 Auf Geolokalisierungsinformationen zugreifen	951
18.3.11 Bild-, Audio- und Videoaufnahmen durchführen	952
18.3.12 Auf Verbindungsinformationen zugreifen	955
18.3.13 Auf Kontakte zugreifen	955
18.3.14 Dateien herunterladen und hochladen	958
18.3.15 UI-Komponenten verwenden	960
18.3.16 Auf Ereignisse reagieren	960
18.3.17 Eine Anwendung bauen	962
18.4 Zusammenfassung	963

19 Desktopanwendungen mit JavaScript 965

19.1 NW.js	966
19.1.1 Installation und Beispielanwendung	967
19.1.2 Packaging	968
19.2 Electron	969
19.2.1 Installation und Beispielanwendung	970
19.2.2 Interprozesskommunikation	971
19.2.3 Packaging	973
19.2.4 Debugging, Monitoring und Testing	974
19.3 Zusammenfassung	976

20 Mikrocontroller mit JavaScript steuern 979

20.1 Espruino	980
20.1.1 Technische Informationen	980
20.1.2 Anschluss und Installation	981

20.1.3	Erstes Beispiel	981
20.1.4	LEDs ansteuern	982
20.1.5	Weitere Module	984
20.1.6	Sensoren auslesen	985
20.2	Tessel	986
20.2.1	Technische Informationen	986
20.2.2	Anschluss und Installation	987
20.2.3	LEDs ansteuern	988
20.2.4	Die Drucktaster programmieren	989
20.2.5	Den Tessel durch Module erweitern	990
20.3	BeagleBone Black	991
20.3.1	Technische Informationen	991
20.3.2	Anschluss und Installation	992
20.3.3	LEDs ansteuern	993
20.4	Arduino	994
20.4.1	Das Firmata-Protokoll	995
20.4.2	Anschluss und Installation	995
20.4.3	Das Node.js-Modul Johnny Five	996
20.5	Cylon.js	997
20.5.1	Steuern eines BeagleBone Black mit Cylon.js	998
20.5.2	Steuern eines Tessel-Boards mit Cylon.js	998
20.5.3	Steuern eines Arduinos mit Cylon.js	999
20.6	Zusammenfassung	999
21	Einen professionellen Entwicklungsprozess aufsetzen	1001
21.1	Aufgaben automatisieren	1001
21.1.1	Aufgaben automatisieren mit Grunt	1002
21.1.2	Aufgaben automatisieren mit Gulp	1005
21.2	Quelltext automatisiert testen	1006
21.2.1	Das Prinzip von automatisierten Tests	1007
21.2.2	Das Prinzip der testgetriebenen Entwicklung	1008
21.2.3	Den Quelltext automatisiert testen mit QUnit	1009
21.2.4	Den Quelltext automatisiert testen mit mocha	1016

21.3	Versionsverwaltung des Quelltextes	1020
21.3.1	Einführung in die Versionsverwaltung	1020
21.3.2	Das Versionsverwaltungssystem Git installieren und konfigurieren	1024
21.3.3	Ein neues lokales Repository anlegen	1026
21.3.4	Ein bestehendes Repository klonen	1026
21.3.5	Änderungen in den Staging-Bereich übertragen	1027
21.3.6	Änderungen in das lokale Repository übertragen	1027
21.3.7	Die verschiedenen Zustände in Git	1029
21.3.8	Änderungen in das Remote Repository übertragen	1030
21.3.9	Änderungen aus dem Remote Repository übertragen	1031
21.3.10	In einem neuen Branch arbeiten	1032
21.3.11	Änderungen aus einem Branch übernehmen	1033
21.3.12	Übersicht über die wichtigsten Befehle und Begriffe	1034
21.4	Zusammenfassung	1038
Anhang	1005
A	JavaScript-Referenz	1041
B	DOM-Referenz und HTML-Erweiterungen	1097
C	BOM und Ajax	1189
D	HTML5-Web-APIs-Referenz	1223
Index	1275