

Inhalt

Vorwort	23
---------------	----

1 Grundlagen und Einführung

1.1 Grundlagen der Programmierung	27
1.1.1 Mit dem Computer kommunizieren	28
1.1.2 Programmiersprachen	29
1.1.3 Hilfsmittel für den Programmentwurf	37
1.2 Einführung JavaScript	42
1.2.1 Historie	43
1.2.2 Anwendungsgebiete	43
1.3 Zusammenfassung	51

2 Erste Schritte

2.1 Einführung JavaScript und Webentwicklung	53
2.1.1 Der Zusammenhang zwischen HTML, CSS und JavaScript	53
2.1.2 Das richtige Werkzeug für die Entwicklung	57
2.2 JavaScript in eine Webseite einbinden	61
2.2.1 Eine geeignete Ordnerstruktur vorbereiten	62
2.2.2 Eine JavaScript-Datei erstellen	63
2.2.3 Eine JavaScript-Datei in eine HTML-Datei einbinden	63
2.2.4 JavaScript direkt innerhalb des HTML definieren	66
2.2.5 Platzierung und Ausführung der <script>-Elemente	68
2.2.6 Den Quelltext anzeigen	72
2.3 Eine Ausgabe erzeugen	74
2.3.1 Standarddialogfenster anzeigen	74
2.3.2 Auf die Konsole schreiben	75
2.3.3 Bestehende UI-Komponenten verwenden	81
2.4 Zusammenfassung	82

3 Sprachkern	85
3.1 Werte in Variablen speichern	85
3.1.1 Variablen definieren	85
3.1.2 Gültige Variablennamen verwenden	88
3.1.3 Konstanten definieren	95
3.2 Die verschiedenen Datentypen verwenden	96
3.2.1 Zahlen	96
3.2.2 Zeichenketten	99
3.2.3 Boolesche Wahrheitswerte	102
3.2.4 Arrays	102
3.2.5 Objekte	107
3.2.6 Besondere Datentypen	109
3.3 Die verschiedenen Operatoren einsetzen	110
3.3.1 Operatoren für das Arbeiten mit Zahlen	112
3.3.2 Operatoren für das einfachere Zuweisen	113
3.3.3 Operatoren für das Arbeiten mit Zeichenketten	114
3.3.4 Operatoren für das Arbeiten mit booleschen Werten	115
3.3.5 Operatoren für das Arbeiten mit Bits	121
3.3.6 Operatoren für das Vergleichen von Werten	122
3.3.7 Operatoren für spezielle Operationen	124
3.4 Den Ablauf eines Programms steuern	125
3.4.1 Bedingte Anweisungen definieren	126
3.4.2 Verzweigungen definieren	128
3.4.3 Den Auswahloperator verwenden	134
3.4.4 Mehrfachverzweigungen definieren	135
3.4.5 Zählschleifen definieren	142
3.4.6 Kopfgesteuerte Schleifen definieren	150
3.4.7 Fußgesteuerte Schleifen definieren	153
3.4.8 Schleifen und Schleifeniterationen vorzeitig abbrechen	154
3.5 Wiederverwendbare Codebausteine erstellen	163
3.5.1 Funktionen definieren	163
3.5.2 Funktionen aufrufen	166
3.5.3 Funktionsparameter übergeben und auswerten	166
3.5.4 Rückgabewerte definieren	175
3.5.5 Standardwerte für Parameter definieren	177
3.5.6 Elemente aus einem Array als Parameter verwenden	179
3.5.7 Funktionen über Kurzschreibweise definieren	181
3.5.8 Funktionen im Detail	183
3.5.9 Funktionen aufrufen durch Nutzerinteraktion	191

3.6 Auf Fehler reagieren und sie richtig behandeln	193
3.6.1 Syntaxfehler	193
3.6.2 Laufzeitfehler	194
3.6.3 Logische Fehler	195
3.6.4 Das Prinzip der Fehlerbehandlung	196
3.6.5 Fehler fangen und behandeln	197
3.6.6 Fehler auslösen	200
3.6.7 Fehler und der Funktionsaufruf-Stack	203
3.6.8 Bestimmte Anweisungen unabhängig von aufgetretenen Fehlern aufrufen	205
3.7 Den Quelltext kommentieren	211
3.8 Den Code debuggen	212
3.8.1 Einführung	212
3.8.2 Ein einfaches Codebeispiel	213
3.8.3 Haltepunkte definieren	214
3.8.4 Variablenbelegungen einsehen	216
3.8.5 Ein Programm schrittweise ausführen	217
3.8.6 Mehrere Haltepunkte definieren	219
3.8.7 Bedingte Haltepunkte definieren	219
3.8.8 Den Funktionsaufruf-Stack einsehen	220
3.9 Zusammenfassung	221

4 Mit Objekten und Referenztypen arbeiten	225
4.1 Unterschied zwischen primitiven Datentypen und Referenztypen	225
4.1.1 Das Prinzip von primitiven Datentypen	225
4.1.2 Das Prinzip von Referenztypen	226
4.1.3 Primitive Datentypen und Referenztypen als Funktionsargumente	228
4.1.4 Den Typ einer Variablen ermitteln	229
4.1.5 Ausblick	232
4.2 Zustand und Verhalten in Objekten kapseln	232
4.2.1 Einführung objektorientierte Programmierung	233
4.2.2 Objekte erstellen über die Literal-Schreibweise	234
4.2.3 Objekte erstellen über Konstruktorfunktionen	235
4.2.4 Objekte erstellen unter Verwendung von Klassen	238
4.2.5 Objekte erstellen über die Funktion »Object.create()«	242
4.2.6 Auf Eigenschaften zugreifen und Methoden aufrufen	245
4.2.7 Objekteigenschaften und Objektmethoden hinzufügen oder überschreiben	252

4.2.8	Objekteigenschaften und Objektmethoden löschen	256
4.2.9	Objekteigenschaften und Objektmethoden ausgeben	258
4.2.10	Änderungen an Objekten verhindern	261
4.3	Mit Arrays arbeiten	265
4.3.1	Arrays erzeugen und initialisieren	265
4.3.2	Auf Elemente eines Arrays zugreifen	268
4.3.3	Elemente einem Array hinzufügen	269
4.3.4	Elemente aus einem Array entfernen	274
4.3.5	Einen Teil der Elemente aus einem Array kopieren	277
4.3.6	Arrays sortieren	280
4.3.7	Arrays als Stack verwenden	283
4.3.8	Arrays als Queue verwenden	284
4.3.9	Elemente in Arrays finden	286
4.3.10	Elemente innerhalb eines Arrays kopieren	288
4.3.11	Arrays in Zeichenketten umwandeln	289
4.4	Mit Zeichenketten arbeiten	290
4.4.1	Der Aufbau einer Zeichenkette	290
4.4.2	Die Länge einer Zeichenkette ermitteln	290
4.4.3	Innerhalb einer Zeichenkette suchen	291
4.4.4	Teile einer Zeichenkette extrahieren	294
4.5	Sonstige globale Objekte	297
4.5.1	Mit Datum und Zeit arbeiten	297
4.5.2	Komplexe Berechnungen durchführen	300
4.5.3	Wrapperobjekte für primitive Datentypen	301
4.6	Mit regulären Ausdrücken arbeiten	301
4.6.1	Reguläre Ausdrücke definieren	302
4.6.2	Zeichen gegen einen regulären Ausdruck testen	302
4.6.3	Zeichenklassen verwenden	305
4.6.4	Anfang und Ende begrenzen	308
4.6.5	Quantifizierer verwenden	311
4.6.6	Nach Vorkommen suchen	316
4.6.7	Alle Vorkommen innerhalb einer Zeichenkette suchen	317
4.6.8	Auf einzelne Teile eines Vorkommens zugreifen	318
4.6.9	Nach bestimmten Zeichenketten suchen	319
4.6.10	Vorkommen innerhalb einer Zeichenkette suchen	319
4.6.11	Nach Vorkommen suchen	320
4.6.12	Zeichenketten zerteilen	321
4.7	Funktionen als Referenztypen	322
4.7.1	Funktionen als Argumente verwenden	322

4.7.2	Funktionen als Rückgabewert verwenden	324
4.7.3	Standardmethoden jeder Funktion	326
4.8	Zusammenfassung	329

5 Webseiten dynamisch verändern

5.1	Aufbau einer Webseite	331
5.1.1	Document Object Model	331
5.1.2	Die verschiedenen Knotentypen	332
5.1.3	Der Dokumentknoten	335
5.2	Elemente selektieren	337
5.2.1	Elemente per ID selektieren	339
5.2.2	Elemente per Klasse selektieren	341
5.2.3	Elemente nach Elementnamen selektieren	345
5.2.4	Elemente nach Namen selektieren	346
5.2.5	Elemente per Selektor selektieren	348
5.2.6	Das Elternelement eines Elements selektieren	354
5.2.7	Die Kindelemente eines Elements selektieren	357
5.2.8	Die Geschwisterelemente eines Elements selektieren	361
5.2.9	Selektionsmethoden auf Elementen aufrufen	363
5.2.10	Elemente nach Typ selektieren	366
5.3	Mit Textknoten arbeiten	366
5.3.1	Auf den Textinhalt eines Elements zugreifen	367
5.3.2	Den Textinhalt eines Elements verändern	368
5.3.3	Das HTML unterhalb eines Elements verändern	369
5.3.4	Textknoten erstellen und hinzufügen	370
5.4	Mit Elementen arbeiten	370
5.4.1	Elemente erstellen und hinzufügen	371
5.4.2	Elemente und Knoten entfernen	374
5.4.3	Die verschiedenen Typen von HTML-Elementen	375
5.5	Mit Attributen arbeiten	380
5.5.1	Den Wert eines Attributs auslesen	381
5.5.2	Den Wert eines Attributs ändern	382
5.5.3	Attributknoten erstellen und hinzufügen	383
5.5.4	Attribute entfernen	384
5.5.5	Auf CSS-Klassen zugreifen	384
5.6	Zusammenfassung	385

6 Ereignisse verarbeiten und auslösen

387

6.1	Das Konzept der ereignisgesteuerten Programmierung	387
6.2	Auf Ereignisse reagieren	388
6.2.1	Einen Event-Handler per HTML definieren	391
6.2.2	Einen Event-Handler per JavaScript definieren	393
6.2.3	Event-Listener definieren	395
6.2.4	Mehrere Event-Listener definieren	397
6.2.5	Argumente an Event-Listener übergeben	399
6.2.6	Event-Listener entfernen	401
6.2.7	Event-Handler und Event-Listener per Helperfunktion definieren	402
6.2.8	Auf Informationen eines Ereignisses zugreifen	403
6.3	Die verschiedenen Typen von Ereignissen	406
6.3.1	Ereignisse bei Interaktion mit der Maus	406
6.3.2	Ereignisse bei Interaktion mit Tastatur und Textfeldern	411
6.3.3	Ereignisse beim Arbeiten mit Formularen	414
6.3.4	Ereignisse bei Fokussieren von Elementen	414
6.3.5	Allgemeine Ereignisse der Nutzerschnittstelle	415
6.3.6	Ereignisse bei mobilen Endgeräten	418
6.4	Den Ereignisfluss verstehen und beeinflussen	419
6.4.1	Die Event-Phasen	419
6.4.2	Den Ereignisfluss unterbrechen	427
6.4.3	Standardaktionen von Events verhindern	432
6.5	Ereignisse programmatisch auslösen	435
6.5.1	Einfache Ereignisse auslösen	435
6.5.2	Ereignisse mit übergebenen Argumenten auslösen	436
6.5.3	Standardereignisse auslösen	436
6.6	Zusammenfassung	437

7 Mit Formularen arbeiten

439

7.1	Auf Formulare und Formularfelder zugreifen	440
7.1.1	Auf Formulare zugreifen	440
7.1.2	Auf Formularelemente zugreifen	443
7.1.3	Den Wert von Textfeldern und Passwortfeldern auslesen	445
7.1.4	Den Wert von Checkboxen auslesen	446

7.1.5	Den Wert von Radiobuttons auslesen	447
7.1.6	Den Wert von Auswahllisten auslesen	449
7.1.7	Die Werte von MehrfachauswahlListen auslesen	450
7.1.8	AuswahlListen per JavaScript mit Werten befüllen	451
7.2	Formulare programmatisch abschicken und zurücksetzen	453
7.3	Formulareingaben validieren	455
7.4	Zusammenfassung	465

8 Browser steuern und Browserinformationen auslesen 467

8.1	Das Browser Object Model	467
8.2	Auf Fensterinformationen zugreifen	469
8.2.1	Die Größe und Position eines Browserfensters ermitteln	469
8.2.2	Die Größe und Position eines Browserfensters ändern	471
8.2.3	Auf Anzeigeelementen der Browserleisten zugreifen	472
8.2.4	Allgemeine Eigenschaften ermitteln	474
8.2.5	Neue Browserfenster öffnen	474
8.2.6	Browserfenster schließen	476
8.2.7	Dialoge öffnen	477
8.2.8	Funktionen zeitgesteuert ausführen	478
8.3	Auf Navigationsinformationen der aktuellen Webseite zugreifen	480
8.3.1	Auf die einzelnen Bestandteile der URL zugreifen	480
8.3.2	Auf Querystring-Parameter zugreifen	481
8.3.3	Eine neue Webseite laden	481
8.4	Den Browserverlauf einsehen und verändern	483
8.4.1	Im Browserverlauf navigieren	483
8.4.2	Browserverlauf bei Single Page Applications	484
8.4.3	Einträge in den Browserverlauf hinzufügen	485
8.4.4	Auf Änderungen im Browserverlauf reagieren	488
8.4.5	Den aktuellen Eintrag im Browserverlauf ersetzen	488
8.5	Browser erkennen und Browserfeatures bestimmen	490
8.6	Auf Informationen des Bildschirms zugreifen	492
8.7	Zusammenfassung	494

9 Inhalte einer Webseite dynamisch nachladen

495

9.1	Das Prinzip von Ajax	495
9.1.1	Synchrone Kommunikation	495
9.1.2	Asynchrone Kommunikation	496
9.1.3	Typische Anwendungsfälle für die Verwendung von Ajax	498
9.1.4	Verwendete Datenformate	500
9.2	Das XML-Format	501
9.2.1	Der Aufbau von XML	501
9.2.2	XML und die DOM API	503
9.2.3	Zeichenketten in XML-Objekte umwandeln	504
9.2.4	XML-Objekte in Zeichenketten umwandeln	505
9.3	Das JSON-Format	506
9.3.1	Der Aufbau von JSON	507
9.3.2	Unterschied zwischen JSON und JavaScript-Objekten	509
9.3.3	Objekte in das JSON-Format umwandeln	509
9.3.4	Objekte aus dem JSON-Format umwandeln	511
9.4	Anfragen per Ajax stellen	512
9.4.1	Das »XMLHttpRequest«-Objekt	512
9.4.2	HTML-Daten per Ajax laden	519
9.4.3	XML-Daten per Ajax laden	524
9.4.4	JSON-Daten per Ajax laden	527
9.4.5	Daten per Ajax an den Server schicken	530
9.4.6	Formulare per Ajax abschicken	531
9.4.7	Daten von anderen Domains laden	532
9.4.8	Die neuere Alternative zu »XMLHttpRequest«: die Fetch API	535
9.5	Zusammenfassung	537

10 Aufgaben vereinfachen mit jQuery

539

10.1	Einführung	539
10.1.1	jQuery einbinden	540
10.1.2	jQuery über ein CDN einbinden	541
10.1.3	jQuery verwenden	542
10.1.4	Aufgaben mit jQuery vereinfachen	543
10.2	Mit dem DOM arbeiten	545
10.2.1	Elemente selektieren	545
10.2.2	Auf Inhalte zugreifen und diese verändern	550

10.2.3	Ausgewählte Elemente filtern	554
10.2.4	Auf Attribute zugreifen	556
10.2.5	Auf CSS-Eigenschaften zugreifen	558
10.2.6	Zwischen Elementen navigieren	558
10.2.7	Effekte und Animationen verwenden	560
10.3	Auf Ereignisse reagieren	562
10.3.1	Event-Listener registrieren	562
10.3.2	Auf allgemeine Ereignisse reagieren	564
10.3.3	Auf Mausereignisse reagieren	565
10.3.4	Auf Tastaturereignisse reagieren	567
10.3.5	Auf Formularereignisse reagieren	568
10.3.6	Auf Informationen von Ereignissen zugreifen	569
10.4	Ajax-Anfragen erstellen	571
10.4.1	Ajax-Anfragen erstellen	571
10.4.2	Auf Ereignisse reagieren	574
10.4.3	HTML-Daten per Ajax laden	575
10.4.4	XML-Daten per Ajax laden	576
10.4.5	JSON-Daten per Ajax laden	577
10.5	Zusammenfassung	579

11	Bilder und Grafiken dynamisch erstellen	587
11.1	Bilder zeichnen	587
11.1.1	Die Zeichenfläche	587
11.1.2	Der Rendering-Kontext	588
11.1.3	Rechtecke zeichnen	590
11.1.4	Pfade verwenden	593
11.1.5	Texte zeichnen	599
11.1.6	Farbverläufe zeichnen	600
11.1.7	Speichern und Wiederherstellen des Canvas-Zustands	602
11.1.8	Transformationen anwenden	604
11.1.9	Animationen erstellen	607
11.2	Vektorgrafiken einbinden	609
11.2.1	Das SVG-Format	609
11.2.2	SVG in HTML einbinden	611
11.2.3	Das Aussehen von SVG-Elementen mit CSS beeinflussen	614
11.2.4	Das Verhalten von SVG-Elementen mit JavaScript beeinflussen	615
11.3	Zusammenfassung	617

12 Moderne Web-APIs verwenden

619

12.1	Über JavaScript kommunizieren	621
12.1.1	Unidirektionale Kommunikation mit dem Server	621
12.1.2	Bidirektionale Kommunikation mit einem Server	623
12.1.3	Vom Server ausgehende Kommunikation	625
12.2	Nutzer wiedererkennen	630
12.2.1	Cookies verwenden	630
12.2.2	Cookies anlegen	632
12.2.3	Cookies auslesen	633
12.2.4	Ein Beispiel: Einkaufswagen auf Basis von Cookies	635
12.2.5	Nachteile von Cookies	638
12.3	Den Browserspeicher nutzen	638
12.3.1	Werte im Browserspeicher speichern	639
12.3.2	Werte aus dem Browserspeicher lesen	640
12.3.3	Werte im Browserspeicher aktualisieren	640
12.3.4	Werte aus dem Browserspeicher löschen	641
12.3.5	Auf Änderungen im Browserspeicher reagieren	641
12.3.6	Die verschiedenen Typen von Browserspeichern	642
12.3.7	Ein Beispiel: Einkaufswagen auf Basis des Browserspeichers	644
12.4	Die Browserdatenbank nutzen	645
12.4.1	Öffnen einer Datenbank	646
12.4.2	Erstellen einer Datenbank	647
12.4.3	Erstellen eines Objektspeichers	648
12.4.4	Hinzufügen von Objekten zu einem Objektspeicher	649
12.4.5	Lesen von Objekten aus einem Objektspeicher	652
12.4.6	Löschen von Objekten aus einem Objektspeicher	653
12.4.7	Aktualisieren von Objekten in einem Objektspeicher	655
12.4.8	Verwendung eines Cursors	656
12.5	Auf das Dateisystem zugreifen	657
12.5.1	Auswählen von Dateien per Dateidialog	658
12.5.2	Auswählen von Dateien per Drag & Drop	659
12.5.3	Lesen von Dateien	660
12.5.4	Den Lesefortschritt überwachen	663
12.6	Komponenten einer Webseite verschieben	665
12.6.1	Ereignisse einer Drag-and-Drop-Operation	665
12.6.2	Verschiebbare Elemente definieren	667
12.6.3	Verschieben von Elementen	669

12.7 Aufgaben parallelisieren	670
12.7.1 Das Prinzip von Web Workern	672
12.7.2 Web Worker verwenden	673
12.8 Den Standort von Nutzern ermitteln	674
12.8.1 Auf Standortinformationen zugreifen	674
12.8.2 Kontinuierlich auf Standortinformationen zugreifen	677
12.8.3 Position auf Karte anzeigen	677
12.8.4 Anfahrtsbeschreibung anzeigen	679
12.9 Den Batteriestand eines Endgeräts auslesen	680
12.9.1 Auf Batterieinformationen zugreifen	680
12.9.2 Auf Ereignisse reagieren	682
12.10 Übersicht über verschiedene Web-APIs	683
12.11 Zusammenfassung	687

13 Objektorientierte Programmierung 689

13.1 Die Prinzipien der objektorientierten Programmierung	689
13.1.1 Klassen, Objektinstanzen und Prototypen	690
13.1.2 Prinzip 1: Abstraktes Verhalten definieren	692
13.1.3 Prinzip 2: Zustand und Verhalten kapseln	693
13.1.4 Prinzip 3: Zustand und Verhalten vererben	694
13.1.5 Prinzip 4: Verschiedene Typen annehmen	695
13.1.6 JavaScript und die Objektorientierung	696
13.2 Prototypische Objektorientierung	696
13.2.1 Das Konzept von Prototypen	696
13.2.2 Von Objekten ableiten	697
13.2.3 Methoden und Eigenschaften vererben	698
13.2.4 Methoden und Eigenschaften im erbenden Objekt definieren	698
13.2.5 Methoden überschreiben	699
13.2.6 Die Prototypenkette	700
13.2.7 Methoden des Prototyps aufrufen	702
13.2.8 Prototypische Objektorientierung und die Prinzipien der Objektorientierung	703
13.3 Pseudoklassische Objektorientierung	703
13.3.1 Konstruktorfunktionen definieren	704
13.3.2 Objektinstanzen erzeugen	704
13.3.3 Methoden und Eigenschaften definieren	704
13.3.4 Von Objekten ableiten	705
13.3.5 Konstruktor der »Oberklasse« aufrufen	709

13.3.6	Methoden überschreiben	709
13.3.7	Methoden der »Oberklasse« aufrufen	709
13.3.8	Pseudoklassische Objektorientierung und die Prinzipien der Objektorientierung	710
13.4	Objektorientierung mit Klassensyntax	710
13.4.1	Klassen definieren	711
13.4.2	Objektinstanzen erzeugen	712
13.4.3	Getter und Setter definieren	712
13.4.4	Von »Klassen« ableiten	713
13.4.5	Methoden überschreiben	716
13.4.6	Methoden der »Oberklasse« aufrufen	718
13.4.7	Statische Methoden definieren	719
13.4.8	Statische Eigenschaften definieren	720
13.4.9	Klassensyntax und die Prinzipien der Objektorientierung	721
13.5	Zusammenfassung	722

14 Funktionale Programmierung

14.1	Prinzipien der funktionalen Programmierung	723
14.1.1	Prinzip 1: Funktionen sind Objekte erster Klasse	723
14.1.2	Prinzip 2: Funktionen arbeiten mit unveränderlichen Datenstrukturen	724
14.1.3	Prinzip 3: Funktionen haben keine Nebeneffekte	724
14.1.4	Prinzip 4: Funktionale Programme sind deklarativ	724
14.2	Imperative Programmierung und funktionale Programmierung	725
14.2.1	Iterieren mit der Methode »forEach()«	725
14.2.2	Werte abbilden mit der Methode »map()«	728
14.2.3	Werte filtern mit der Methode »filter()«	729
14.2.4	Mehrere Werte zu einem Wert reduzieren mit der Methode »reduce()«	731
14.2.5	Kombination der verschiedenen Methoden	734
14.3	Zusammenfassung	735

15 Den Quelltext richtig strukturieren

15.1	Namenskonflikte vermeiden	737
15.2	Module definieren und verwenden	741
15.2.1	Das Module-Entwurfsmuster	741

15.2.2	Das Revealing-Module-Entwurfsmuster	745
15.2.3	AMD	749
15.2.4	CommonJS	751
15.2.5	Native Module	752
15.3	Zusammenfassung	754

16 Die neuen ES6-Features richtig nutzen

16.1	Maps verwenden	759
16.1.1	Maps erstellen	759
16.1.2	Grundlegende Operationen	760
16.1.3	Über Maps iterieren	762
16.1.4	Weak Maps verwenden	764
16.2	Sets verwenden	766
16.2.1	Sets erstellen	766
16.2.2	Grundlegende Operationen von Sets	767
16.2.3	Über Sets iterieren	769
16.2.4	WeakSets verwenden	770
16.3	Das Iterieren über Datenstrukturen kapseln	771
16.3.1	Das Prinzip von Iteratoren	771
16.3.2	Iteratoren verwenden	772
16.3.3	Einen eigenen Iterator erstellen	773
16.3.4	Ein iterierbares Objekt erstellen	774
16.3.5	Über iterierbare Objekte iterieren	775
16.4	Funktionen anhalten und fortsetzen	776
16.4.1	Eine Generatorfunktion erstellen	776
16.4.2	Einen Generator erstellen	777
16.4.3	Über Generatoren iterieren	778
16.4.4	Unendliche Generatoren erstellen	778
16.4.5	Generatoren mit Parametern steuern	779
16.5	Den Zugriff auf Objekte abfangen	779
16.5.1	Das Prinzip von Proxies	780
16.5.2	Proxies erstellen	780
16.5.3	Handler für Proxies definieren	780
16.6	Asynchrone Programmierung vereinfachen	783
16.6.1	Das Prinzip der asynchronen Programmierung	783
16.6.2	Promises erstellen	788
16.6.3	Verarbeiten eines Promises	789

16.6.4	Promise-Aufrufe verketten	790
16.6.5	Die Zustände von Promises	791
16.7	Vorlagen für Zeichenketten definieren	792
16.7.1	Template-Strings erstellen	792
16.7.2	Platzhalter innerhalb von Zeichenketten definieren	793
16.7.3	Ausdrücke innerhalb von Zeichenketten auswerten	793
16.7.4	Mehrzeilige Zeichenketten definieren	794
16.7.5	Zeichenketten über Funktionen verändern	795
16.8	Symbole verwenden	796
16.8.1	Symbole erstellen	796
16.8.2	Die Symbol-Registry verwenden	797
16.8.3	Symbole zur Definition eindeutiger Objekteigenschaften verwenden	798
16.8.4	Symbole zur Definition von Konstanten verwenden	799
16.9	Werte aus Arrays und Objekten extrahieren	800
16.9.1	Werte aus Arrays extrahieren	800
16.9.2	Werte aus Objekten extrahieren	804
16.9.3	Werte innerhalb einer Schleife extrahieren	807
16.9.4	Argumente einer Funktion extrahieren	809
16.10	Neue Methoden der Standardobjekte	810
16.10.1	Neue Methoden in »Object«	811
16.10.2	Neue Methoden in »String«	811
16.10.3	Neue Methoden in »Array«	812
16.10.4	Neue Methoden in »RegExp«	813
16.10.5	Neue Methoden in »Number«	814
16.10.6	Neue Methoden in »Math«	814
16.11	Zusammenfassung	815

17	Serverseitige Anwendungen mit Node.js erstellen	817
17.1	Einführung Node.js	817
17.1.1	Die Architektur von Node.js	817
17.1.2	Installation von Node.js	819
17.1.3	Eine einfache Anwendung	820
17.2	Node.js Packages verwalten	820
17.2.1	Den Node.js Package Manager installieren	821
17.2.2	Packages installieren	821
17.2.3	Eigene Packages erstellen	824

17.3 Ereignisse verarbeiten und auslösen	828
17.3.1 Ein Event auslösen und abfangen	829
17.3.2 Ein Event mehrfach auslösen	831
17.3.3 Ein Event genau einmal abfangen	832
17.3.4 Ein Event mehrfach abfangen	832
17.4 Auf das Dateisystem zugreifen	833
17.4.1 Dateien lesen	833
17.4.2 Dateien schreiben	834
17.4.3 Dateiinformationen auslesen	835
17.4.4 Dateien löschen	836
17.4.5 Mit Verzeichnissen arbeiten	837
17.5 Einen Webserver erstellen	838
17.5.1 Einen Webserver starten	838
17.5.2 Dateien per Webserver zur Verfügung stellen	839
17.5.3 Einen Client für einen Webserver erstellen	840
17.5.4 Routen definieren	841
17.5.5 Das Webframework Express verwenden	842
17.6 Auf Datenbanken zugreifen	847
17.6.1 Installation von MongoDB	847
17.6.2 MongoDB-Treiber für Node.js installieren	848
17.6.3 Verbindung zur Datenbank herstellen	848
17.6.4 Eine Collection erstellen	849
17.6.5 Objekte speichern	850
17.6.6 Objekte lesen	851
17.6.7 Objekte aktualisieren	853
17.6.8 Objekte löschen	854
17.7 Zusammenfassung	855

18 Mobile Anwendungen mit JavaScript erstellen

18.1 Die unterschiedlichen Arten mobiler Anwendungen	857
18.1.1 Native Anwendungen	857
18.1.2 Mobile Webanwendungen	858
18.1.3 Hybridanwendungen	860
18.1.4 Vergleich der verschiedenen Ansätze	861
18.2 Mobile Anwendungen mit jQuery Mobile erstellen	863
18.2.1 Das Grundgerüst einer mobilen Anwendung definieren	863
18.2.2 Einzelne Seiten innerhalb einer Anwendung definieren	865

18.2.3	Übergänge zwischen den Seiten definieren	869
18.2.4	Themes verwenden	870
18.2.5	UI-Komponenten verwenden	871
18.2.6	Layout-Raster definieren	881
18.2.7	Auf Ereignisse reagieren	885
18.3	Hybride Anwendungen mit Cordova erstellen	887
18.3.1	Das Prinzip von Cordova	887
18.3.2	Eine Anwendung erstellen	888
18.3.3	Eine Anwendung starten	891
18.3.4	Plugins verwenden	893
18.3.5	Auf Geräteinformationen zugreifen	896
18.3.6	Dialoge anzeigen	898
18.3.7	Auf die Kamera zugreifen	899
18.3.8	Auf Bewegungsinformationen zugreifen	900
18.3.9	Auf Orientierungsinformationen zugreifen	901
18.3.10	Auf Geolokalisierungsinformationen zugreifen	902
18.3.11	Bild-, Audio- und Videoaufnahmen durchführen	904
18.3.12	Auf Verbindungsinformationen zugreifen	906
18.3.13	Auf Kontakte zugreifen	906
18.3.14	Dateien herunterladen und hochladen	909
18.3.15	UI-Komponenten verwenden	911
18.3.16	Auf Ereignisse reagieren	911
18.3.17	Eine Anwendung bauen	913
18.4	Zusammenfassung	914

19 Mikrocontroller mit JavaScript steuern

19.1	Espruino	916
19.1.1	Technische Informationen	916
19.1.2	Anschluss und Installation	917
19.1.3	Erstes Beispiel	917
19.1.4	LEDs ansteuern	918
19.1.5	Weitere Module	920
19.1.6	Sensoren auslesen	921
19.2	Tessel	922
19.2.1	Technische Informationen	922
19.2.2	Anschluss und Installation	923
19.2.3	LEDs ansteuern	924

19.2.4	Die Drucktaster programmieren	925
19.2.5	Den Tessel durch Module erweitern	926
19.3	BeagleBone Black	927
19.3.1	Technische Informationen	927
19.3.2	Anschluss und Installation	928
19.3.3	LEDs ansteuern	929
19.4	Arduino	930
19.4.1	Das Firmata-Protokoll	931
19.4.2	Anschluss und Installation	931
19.4.3	Das Node.js-Modul Johnny Five	932
19.5	Cylon.js	933
19.5.1	Steuern eines BeagleBone Black mit Cylon.js	934
19.5.2	Steuern eines Tessel-Boards mit Cylon.js	934
19.5.3	Steuern eines Arduinos mit Cylon.js	935
19.6	Zusammenfassung	935

20 Einen professionellen Entwicklungsprozess aufsetzen 937

20.1	Aufgaben automatisieren	937
20.1.1	Aufgaben automatisieren mit Grunt	938
20.1.2	Aufgaben automatisieren mit Gulp	941
20.2	Quelltext automatisiert testen	942
20.2.1	Das Prinzip von automatisierten Tests	943
20.2.2	Das Prinzip der testgetriebenen Entwicklung	944
20.2.3	Den Quelltext automatisiert testen mit QUnit	945
20.2.4	Den Quelltext automatisiert testen mit mocha	952
20.3	Versionsverwaltung des Quelltextes	956
20.3.1	Einführung in die Versionsverwaltung	956
20.3.2	Das Versionsverwaltungssystem Git installieren und konfigurieren	960
20.3.3	Ein neues lokales Repository anlegen	962
20.3.4	Ein bestehendes Repository klonen	962
20.3.5	Änderungen in den Staging-Bereich übertragen	963
20.3.6	Änderungen in das lokale Repository übertragen	963
20.3.7	Die verschiedenen Zustände in Git	965
20.3.8	Änderungen in das Remote Repository übertragen	966
20.3.9	Änderungen aus dem Remote Repository übertragen	967

20.3.10 In einem neuen Branch arbeiten	968
20.3.11 Änderungen aus einem Branch übernehmen	969
20.3.12 Übersicht über die wichtigsten Befehle und Begriffe	970
20.4 Zusammenfassung	974
Anhang	975
<hr/>	
A JavaScript-Referenz	977
B DOM-Referenz und HTML-Erweiterungen	1035
C BOM und Ajax	1129
D HTML5-Web-APIs-Referenz	1161
Index	1213