

Contents

Book Resources	25
Preface	27

1 Basics and Introduction 31

1.1 Programming Basics	31
1.1.1 Communicating with the Computer	32
1.1.2 Programming Languages	33
1.1.3 Tools for Program Design	40
1.2 Introduction to JavaScript	46
1.2.1 History	46
1.2.2 Fields of Application	47
1.3 Summary	53

2 Getting Started 55

2.1 Introduction to JavaScript and Web Development	55
2.1.1 The Relationship among HTML, CSS, and JavaScript	55
2.1.2 The Right Tool for Development	59
2.2 Integrating JavaScript into a Web Page	64
2.2.1 Preparing a Suitable Folder Structure	64
2.2.2 Creating a JavaScript File	65
2.2.3 Embedding a JavaScript File in an HTML File	66
2.2.4 Defining JavaScript Directly within the HTML	69
2.2.5 Placement and Execution of the <script> Elements	70
2.2.6 Displaying the Source Code	74
2.3 Creating Output	76
2.3.1 Showing the Standard Dialog Window	76
2.3.2 Writing to the Console	78
2.3.3 Using Existing UI Components	81
2.4 Summary	83

3 Language Core 85

3.1 Storing Values in Variables	85
3.1.1 Defining Variables	85
3.1.2 Using Valid Variable Names	88
3.1.3 Defining Constants	94
3.2 Using the Different Data Types	94
3.2.1 Numbers	95
3.2.2 Strings	98
3.2.3 Boolean Values	103
3.2.4 Arrays	104
3.2.5 Objects	109
3.2.6 Special Data Types	110
3.2.7 Symbols	112
3.3 Deploying the Different Operators	112
3.3.1 Operators for Working with Numbers	114
3.3.2 Operators for Easier Assignment	115
3.3.3 Operators for Working with Strings	116
3.3.4 Operators for Working with Boolean Values	117
3.3.5 Operators for Working with Bits	124
3.3.6 Operators for Comparing Values	125
3.3.7 The Optional Chaining Operator	128
3.3.8 The Logical Assignment Operators	130
3.3.9 Operators for Special Operations	132
3.4 Controlling the Flow of a Program	132
3.4.1 Defining Conditional Statements	133
3.4.2 Defining Branches	134
3.4.3 Using the Selection Operator	140
3.4.4 Defining Multiway Branches	142
3.4.5 Defining Counting Loops	148
3.4.6 Defining Head-Controlled Loops	155
3.4.7 Defining Tail-Controlled Loops	158
3.4.8 Prematurely Terminating Loops and Loop Iterations	160
3.5 Creating Reusable Code Blocks	168
3.5.1 Defining Functions	168
3.5.2 Calling Functions	171
3.5.3 Passing and Evaluating Function Parameters	172
3.5.4 Defining Return Values	180
3.5.5 Defining Default Values for Parameters	182
3.5.6 Using Elements from an Array as Parameters	184
3.5.7 Defining Functions Using Short Notation	185

3.5.8	Modifying Strings via Functions	188
3.5.9	Functions in Detail	189
3.5.10	Calling Functions through User Interaction	197
3.6	Responding to Errors and Handling Them Correctly	198
3.6.1	Syntax Errors	198
3.6.2	Runtime Errors	199
3.6.3	Logic Errors	200
3.6.4	The Principle of Error Handling	201
3.6.5	Catching and Handling Errors	202
3.6.6	Triggering Errors	205
3.6.7	Errors and the Function Call Stack	208
3.6.8	Calling Certain Statements Regardless of Errors That Have Occurred	210
3.7	Commenting the Source Code	216
3.8	Debugging the Code	216
3.8.1	Introduction	217
3.8.2	A Simple Code Example	217
3.8.3	Defining Breakpoints	218
3.8.4	Viewing Variable Assignments	220
3.8.5	Running a Program Step by Step	221
3.8.6	Defining Multiple Breakpoints	223
3.8.7	Other Types of Breakpoints	223
3.8.8	Viewing the Function Call Stack	224
3.9	Summary	226

4 Working with Reference Types 229

4.1	Difference between Primitive Data Types and Reference Types	229
4.1.1	The Principle of Primitive Data Types	229
4.1.2	The Principle of Reference Types	230
4.1.3	Primitive Data Types and Reference Types as Function Arguments	232
4.1.4	Determining the Type of a Variable	233
4.1.5	Outlook	236
4.2	Encapsulating State and Behavior in Objects	236
4.2.1	Introduction to Object-Oriented Programming	236
4.2.2	Creating Objects Using Literal Notation	237
4.2.3	Creating Objects via Constructor Functions	239
4.2.4	Creating Objects Using Classes	242
4.2.5	Creating Objects via the Object.create() Function	246

4.2.6	Accessing Properties and Calling Methods	249
4.2.7	Adding or Overwriting Object Properties and Object Methods	256
4.2.8	Deleting Object Properties and Object Methods	260
4.2.9	Outputting Object Properties and Object Methods	262
4.2.10	Using Symbols to Define Unique Object Properties	265
4.2.11	Preventing Changes to Objects	267
4.3	Working with Arrays	270
4.3.1	Creating and Initializing Arrays	270
4.3.2	Accessing Elements of an Array	273
4.3.3	Adding Elements to an Array	274
4.3.4	Removing Elements from an Array	279
4.3.5	Copying Some of the Elements from an Array	282
4.3.6	Sorting Arrays	284
4.3.7	Using Arrays as a Stack	287
4.3.8	Using Arrays as a Queue	288
4.3.9	Finding Elements in Arrays	290
4.3.10	Copying Elements within an Array	292
4.3.11	Converting Arrays to Strings	293
4.4	Extracting Values from Arrays and Objects	294
4.4.1	Extracting Values from Arrays	294
4.4.2	Extracting Values from Objects	298
4.4.3	Extracting Values within a Loop	302
4.4.4	Extracting Arguments of a Function	303
4.4.5	Copying Object Properties to Another Object	304
4.4.6	Copying Object Properties from Another Object	305
4.5	Working with Strings	306
4.5.1	The Structure of a String	306
4.5.2	Determining the Length of a String	307
4.5.3	Searching within a String	308
4.5.4	Extracting Parts of a String	310
4.6	Using Maps	314
4.6.1	Creating Maps	314
4.6.2	Basic Operations	315
4.6.3	Iterating over Maps	317
4.6.4	Using Weak Maps	319
4.7	Using Sets	321
4.7.1	Creating Sets	321
4.7.2	Basic Operations of Sets	321
4.7.3	Iterating over Sets	323
4.7.4	Using Weak Sets	324

4.8	Other Global Objects	325
4.8.1	Working with Date and Time Information	325
4.8.2	Performing Complex Calculations	328
4.8.3	Wrapper Objects for Primitive Data Types	329
4.9	Working with Regular Expressions	329
4.9.1	Defining Regular Expressions	330
4.9.2	Testing Characters against a Regular Expression	330
4.9.3	Using Character Classes	333
4.9.4	Limiting Beginning and End	336
4.9.5	Using Quantifiers	339
4.9.6	Searching for Occurrences	343
4.9.7	Searching All Occurrences within a String	344
4.9.8	Accessing Individual Parts of an Occurrence	345
4.9.9	Searching for Specific Strings	346
4.9.10	Replacing Occurrences within a String	347
4.9.11	Searching for Occurrences	347
4.9.12	Splitting Strings	348
4.10	Functions as Reference Types	349
4.10.1	Using Functions as Arguments	349
4.10.2	Using Functions as Return Values	351
4.10.3	Standard Methods of Each Function	353
4.11	Summary	356

5 Dynamically Changing Web Pages 357

5.1	Structure of a Web Page	357
5.1.1	Document Object Model	357
5.1.2	The Different Types of Nodes	358
5.1.3	The Document Node	361
5.2	Selecting Elements	363
5.2.1	Selecting Elements by ID	364
5.2.2	Selecting Elements by Class	367
5.2.3	Selecting Elements by Element Name	370
5.2.4	Selecting Elements by Name	371
5.2.5	Selecting Elements by Selector	373
5.2.6	Selecting the Parent Element of an Element	378
5.2.7	Selecting the Child Elements of an Element	381
5.2.8	Selecting the Sibling Elements of an Element	385
5.2.9	Calling Selection Methods on Elements	387

5.2.10	Selecting Elements by Type	389
5.3	Working with Text Nodes	390
5.3.1	Accessing the Text Content of an Element	391
5.3.2	Modifying the Text Content of an Element	391
5.3.3	Modifying the HTML below an Element	392
5.3.4	Creating and Adding Text Nodes	393
5.4	Working with Elements	394
5.4.1	Creating and Adding Elements	394
5.4.2	Removing Elements and Nodes	397
5.4.3	The Different Types of HTML Elements	398
5.5	Working with Attributes	403
5.5.1	Reading the Value of an Attribute	403
5.5.2	Changing the Value of an Attribute or Adding a New Attribute	405
5.5.3	Creating and Adding Attribute Nodes	406
5.5.4	Removing Attributes	406
5.5.5	Accessing CSS classes	406
5.6	Summary	408

6 Processing and Triggering Events 409

6.1	The Concept of Event-Driven Programming	409
6.2	Responding to Events	410
6.2.1	Defining an Event Handler via HTML	412
6.2.2	Defining an Event Handler via JavaScript	415
6.2.3	Defining Event Listeners	417
6.2.4	Defining Multiple Event Listeners	418
6.2.5	Passing Arguments to Event Listeners	420
6.2.6	Removing Event Listeners	422
6.2.7	Defining Event Handlers and Event Listeners via a Helper Function	423
6.2.8	Accessing Information of an Event	424
6.3	The Different Types of Events	426
6.3.1	Events when Interacting with the Mouse	427
6.3.2	Events when Interacting with the Keyboard and with Text Fields	431
6.3.3	Events when Working with Forms	434
6.3.4	Events when Focusing Elements	435
6.3.5	General Events of the User Interface	435
6.3.6	Events on Mobile Devices	438

6.4	Understanding and Influencing the Flow of Events	439
6.4.1	The Event Phases	439
6.4.2	Interrupting the Event Flow	447
6.4.3	Preventing Default Actions of Events	452
6.5	Programmatically Triggering Events	454
6.5.1	Triggering Simple Events	454
6.5.2	Triggering Events with Passed Arguments	455
6.5.3	Triggering Default Events	456
6.6	Summary	456

7 Working with Forms 459

7.1	Accessing Forms and Form Fields	459
7.1.1	Accessing Forms	459
7.1.2	Accessing Form Elements	463
7.1.3	Reading the Value of Text Fields and Password Fields	465
7.1.4	Reading the Value of Checkboxes	467
7.1.5	Reading the Value of Radio Buttons	467
7.1.6	Reading the Value of Selection Lists	469
7.1.7	Reading the Values of Multiple Selection Lists	470
7.1.8	Populating Selection Lists with Values Using JavaScript	471
7.2	Programmatically Submitting and Resetting Forms	472
7.3	Validating Form Inputs	475
7.4	Summary	485

8 Controlling Browsers and Reading Browser Information 487

8.1	The Browser Object Model	487
8.2	Accessing Window Information	489
8.2.1	Determining the Size and Position of a Browser Window	489
8.2.2	Changing the Size and Position of a Browser Window	490
8.2.3	Accessing Display Information of the Browser Bars	492
8.2.4	Determining General Properties	493
8.2.5	Opening New Browser Windows	494
8.2.6	Closing the Browser Window	495

8.2.7	Opening Dialogs	496
8.2.8	Executing Functions in a Time-Controlled Manner	497
8.3	Accessing Navigation Information of a Currently Open Web Page	499
8.3.1	Accessing the Individual Components of the URL	499
8.3.2	Accessing Query String Parameters	500
8.3.3	Loading a New Web Page	500
8.4	Viewing and Modifying the Browsing History	502
8.4.1	Navigating in the Browsing History	502
8.4.2	Browsing History for Single-Page Applications	503
8.4.3	Adding Entries to the Browsing History	503
8.4.4	Responding to Changes in the Browsing History	506
8.4.5	Replacing the Current Entry in the Browsing History	506
8.5	Recognizing Browsers and Determining Browser Features	508
8.6	Accessing Screen Information	510
8.7	Summary	511

9 Dynamically Reloading Contents of a Web Page 513

9.1	The Principle of Ajax	513
9.1.1	Synchronous Communication	513
9.1.2	Asynchronous Communication	514
9.1.3	Typical Use Cases for Ajax	516
9.1.4	Data Formats Used	518
9.2	The XML Format	519
9.2.1	The Structure of XML	519
9.2.2	XML and the DOM API	521
9.2.3	Converting Strings to XML Objects	522
9.2.4	Converting XML Objects to Strings	523
9.3	The JSON Format	524
9.3.1	The Structure of JSON	524
9.3.2	Difference between JSON and JavaScript Objects	526
9.3.3	Converting Objects to JSON Format	527
9.3.4	Converting Objects from JSON Format	528
9.4	Making Requests via Ajax	529
9.4.1	The XMLHttpRequest Object	529
9.4.2	Loading HTML Data via Ajax	535
9.4.3	Loading XML Data via Ajax	539

9.4.4	Loading JSON Data via Ajax	543
9.4.5	Sending Data to the Server via Ajax	545
9.4.6	Submitting Forms via Ajax	546
9.4.7	Loading Data from Other Domains	547
9.4.8	The Newer Alternative to XMLHttpRequest: The Fetch API	550
9.5	Summary	554

10 Simplifying Tasks with jQuery 555

10.1	Introduction	555
10.1.1	Embedding jQuery	556
10.1.2	Embedding jQuery via a Content Delivery Network	557
10.1.3	Using jQuery	558
10.1.4	Simplifying Tasks with jQuery	559
10.2	Working with the DOM	560
10.2.1	Selecting Elements	561
10.2.2	Accessing and Modifying Content	566
10.2.3	Filtering Selected Elements	569
10.2.4	Accessing Attributes	571
10.2.5	Accessing CSS Properties	572
10.2.6	Navigating between Elements	573
10.2.7	Using Effects and Animations	575
10.3	Responding to Events	576
10.3.1	Registering Event Listeners	577
10.3.2	Responding to General Events	578
10.3.3	Responding to Mouse Events	579
10.3.4	Responding to Keyboard Events	581
10.3.5	Responding to Form Events	581
10.3.6	Accessing Information from Events	582
10.4	Creating Ajax Requests	584
10.4.1	Creating Ajax Requests	584
10.4.2	Responding to Events	587
10.4.3	Loading HTML Data via Ajax	588
10.4.4	Loading XML Data via Ajax	589
10.4.5	Loading JSON Data via Ajax	590
10.5	Summary	592

11

Dynamically Creating Images and Graphics

599

11.1	Drawing Images	599
11.1.1	The Drawing Area	599
11.1.2	The Rendering Context	600
11.1.3	Drawing Rectangles	602
11.1.4	Using Paths	604
11.1.5	Drawing Texts	610
11.1.6	Drawing Gradients	611
11.1.7	Saving and Restoring the Canvas State	613
11.1.8	Using Transformations	615
11.1.9	Creating Animations	618
11.2	Integrating Vector Graphics	620
11.2.1	The SVG Format	620
11.2.2	Integrating SVG in HTML	621
11.2.3	Changing the Appearance of SVG Elements with CSS	624
11.2.4	Manipulating the Behavior of SVG Elements via JavaScript	625
11.3	Summary	627

12

Using Modern Web APIs

629

12.1	Communicating via JavaScript	631
12.1.1	Unidirectional Communication with the Server	631
12.1.2	Bidirectional Communication with a Server	633
12.1.3	Outgoing Communication from the Server	635
12.2	Recognizing Users	639
12.2.1	Using Cookies	639
12.2.2	Creating Cookies	641
12.2.3	Reading Cookies	642
12.2.4	Example: Shopping Cart Based on Cookies	644
12.2.5	Disadvantages of Cookies	647
12.3	Using the Browser Storage	647
12.3.1	Storing Values in the Browser Storage	648
12.3.2	Reading Values from the Browser Storage	649
12.3.3	Updating Values in the Browser Storage	649
12.3.4	Deleting Values from the Browser Storage	650
12.3.5	Responding to Changes in the Browser Storage	650
12.3.6	The Different Types of Browser Storage	651
12.3.7	Example: Shopping Cart Based on the Browser Storage	653

12.4	Using the Browser Database	654
12.4.1	Opening a Database	655
12.4.2	Creating a Database	656
12.4.3	Creating an Object Store	657
12.4.4	Adding Objects to an Object Store	657
12.4.5	Reading Objects from an Object Store	661
12.4.6	Deleting Objects from an Object Store	661
12.4.7	Updating Objects in an Object Store	663
12.4.8	Using a Cursor	664
12.5	Accessing the File System	665
12.5.1	Selecting Files via File Dialog	666
12.5.2	Selecting Files via Drag and Drop	667
12.5.3	Reading Files	668
12.5.4	Monitoring the Reading Progress	671
12.6	Moving Components of a Web Page	673
12.6.1	Events of a Drag-and-Drop Operation	673
12.6.2	Defining Movable Elements	674
12.6.3	Moving Elements	676
12.7	Parallelizing Tasks	678
12.7.1	The Principle of Web Workers	679
12.7.2	Use Web Workers	680
12.8	Determining the Location of Users	682
12.8.1	Accessing Location Information	682
12.8.2	Continuously Accessing Location Information	684
12.8.3	Showing the Position on a Map	685
12.8.4	Showing Directions	686
12.9	Reading the Battery Level of an End Device	688
12.9.1	Accessing Battery Information	688
12.9.2	Responding to Events	689
12.10	Outputting Speech and Recognizing Speech	691
12.10.1	Outputting Speech	692
12.10.2	Recognizing Speech	694
12.11	Creating Animations	695
12.11.1	Using the API	696
12.11.2	Controlling an Animation	699
12.12	Working with the Command Line	699
12.12.1	Selecting and Inspecting DOM Elements	701
12.12.2	Events Analysis	704
12.12.3	Debugging, Monitoring, and Profiling	704

12.13 Developing Multilingual Applications	708
12.13.1 Explanation of Terms	709
12.13.2 The Internationalization API	710
12.13.3 Comparing Character String Expressions	712
12.13.4 Formatting Dates and Times	714
12.13.5 Formatting Numeric Values	717
12.14 Overview of Various Web APIs	720
12.15 Summary	724

13 Object-Oriented Programming 725

13.1 The Principles of Object-Oriented Programming	725
13.1.1 Classes, Object Instances, and Prototypes	726
13.1.2 Principle 1: Define Abstract Behavior	728
13.1.3 Principle 2: Encapsulate Condition and Behavior	728
13.1.4 Principle 3: Inherit Condition and Behavior	729
13.1.5 Principle 4: Accept Different Types	731
13.1.6 JavaScript and Object Orientation	731
13.2 Prototypical Object Orientation	732
13.2.1 The Concept of Prototypes	732
13.2.2 Deriving from Objects	733
13.2.3 Inheriting Methods and Properties	733
13.2.4 Defining Methods and Properties in the Inheriting Object	734
13.2.5 Overwriting Methods	735
13.2.6 The Prototype Chain	736
13.2.7 Calling Methods of the Prototype	737
13.2.8 Prototypical Object Orientation and the Principles of Object Orientation	738
13.3 Pseudoclassical Object Orientation	739
13.3.1 Defining Constructor Functions	739
13.3.2 Creating Object Instances	739
13.3.3 Defining Methods	740
13.3.4 Deriving from Objects	740
13.3.5 Calling the Constructor of the "Superclass"	744
13.3.6 Overwriting Methods	744
13.3.7 Calling Methods of the "Superclass"	745
13.3.8 Pseudoclassical Object Orientation and the Principles of Object Orientation	745

13.4	Object Orientation with Class Syntax	745
13.4.1	Defining Classes	746
13.4.2	Creating Object Instances	748
13.4.3	Defining Getters and Setters	748
13.4.4	Defining Private Properties and Private Methods	750
13.4.5	Deriving from "Classes"	753
13.4.6	Overwriting Methods	757
13.4.7	Calling Methods of the "Superclass"	759
13.4.8	Defining Static Methods	760
13.4.9	Defining Static Properties	762
13.4.10	Class Syntax and the Principles of Object Orientation	763
13.5	Summary	764

14 Functional Programming 765

14.1	Principles of Functional Programming	765
14.2	Imperative Programming and Functional Programming	767
14.2.1	Iterating with the <code>forEach()</code> Method	767
14.2.2	Mapping Values with the <code>map()</code> Method	770
14.2.3	Filtering Values with the <code>filter()</code> Method	771
14.2.4	Reducing Multiple Values to One Value with the <code>reduce()</code> Method	773
14.2.5	Combination of the Different Methods	775
14.3	Summary	776

15 Correctly Structuring the Source Code 779

15.1	Avoiding Name Conflicts	779
15.2	Defining and Using Modules	783
15.2.1	The Module Design Pattern	783
15.2.2	The Revealing Module Design Pattern	786
15.2.3	AMD	791
15.2.4	CommonJS	792
15.2.5	Native Modules	794
15.3	Summary	797

16 Using Asynchronous Programming and Other Advanced Features 799

16.1 Understanding and Using Asynchronous Programming	799
16.1.1 Using the Callback Design Pattern	800
16.1.2 Using Promises	804
16.1.3 Using Async Functions	813
16.2 Encapsulating Iteration over Data Structures	816
16.2.1 The Principle of Iterators	816
16.2.2 Using Iterators	817
16.2.3 Creating Your Own Iterator	817
16.2.4 Creating an Iterable Object	819
16.2.5 Iterating over Iterable Objects	820
16.3 Pausing and Resuming Functions	820
16.3.1 Creating a Generator Function	821
16.3.2 Creating a Generator	821
16.3.3 Iterating over Generators	822
16.3.4 Creating Infinite Generators	822
16.3.5 Controlling Generators with Parameters	823
16.4 Intercepting Access to Objects	824
16.4.1 The Principle of Proxies	824
16.4.2 Creating Proxies	825
16.4.3 Defining Handlers for Proxies	825
16.5 Summary	829

17 Creating Server-Based Applications with Node.js 831

17.1 Introduction to Node.js	831
17.1.1 The Architecture of Node.js	831
17.1.2 Installing Node.js	833
17.1.3 A Simple Application	833
17.2 Managing Node.js Packages	834
17.2.1 Installing the Node.js Package Manager	834
17.2.2 Installing Packages	835
17.2.3 Creating Your Own Packages	838
17.3 Processing and Triggering Events	841
17.3.1 Triggering and Intercepting an Event	841
17.3.2 Triggering an Event Multiple Times	844

17.3.3	Intercepting an Event Exactly Once	844
17.3.4	Intercepting an Event Multiple Times	845
17.4	Accessing the File System	846
17.4.1	Reading Files	846
17.4.2	Writing Files	847
17.4.3	Reading File Information	848
17.4.4	Deleting Files	849
17.4.5	Working with Directories	849
17.5	Creating a Web Server	851
17.5.1	Starting a Web Server	851
17.5.2	Making Files Available via Web Server	852
17.5.3	Creating a Client for a Web Server	853
17.5.4	Defining Routes	853
17.5.5	Using the Express.js Web Framework	854
17.6	Accessing Databases	859
17.6.1	MongoDB Installation	859
17.6.2	Installing a MongoDB Driver for Node.js	860
17.6.3	Establishing a Connection to the Database	860
17.6.4	Creating a Collection	861
17.6.5	Saving Objects	862
17.6.6	Reading Objects	862
17.6.7	Updating Objects	865
17.6.8	Deleting Objects	865
17.7	Working with Streams	866
17.7.1	Introduction and Types of Streams	866
17.7.2	Stream Use Cases	867
17.7.3	Reading Data with Streams	868
17.7.4	Writing Data with Streams	869
17.7.5	Combining Streams Using Piping	870
17.7.6	Error Handling during Piping	873
17.8	Summary	874

18 Creating Mobile Applications with JavaScript 877

18.1	The Different Types of Mobile Applications	877
18.1.1	Native Applications	877
18.1.2	Mobile Web Applications	878
18.1.3	Hybrid Applications	879
18.1.4	Comparison of the Different Approaches	881

18.2	Creating Mobile Applications with React Native	883
18.2.1	The Principle of React Native	883
18.2.2	Installation and Project Initialization	883
18.2.3	Starting the Application	884
18.2.4	The Basic Structure of a React Native Application	887
18.2.5	Using UI Components	889
18.2.6	Communication with the Server	894
18.2.7	Building and Publishing Applications	895
18.3	Summary	895

19 Desktop Applications with JavaScript 897

19.1	NW.js	898
19.1.1	Installing and Creating an Application	899
19.1.2	Starting the Application	900
19.1.3	Packaging of the Application	901
19.1.4	More Sample Applications	902
19.2	Electron	903
19.2.1	Installing and Creating an Application	904
19.2.2	Starting the Application	906
19.2.3	Packaging	906
19.2.4	More Sample Applications	907
19.3	Summary	908

20 Controlling Microcontrollers with JavaScript 909

20.1	Espruino	910
20.1.1	Technical Information	910
20.1.2	Connection and Installation	911
20.1.3	First Example	911
20.1.4	Controlling LEDs	912
20.1.5	More Modules	914
20.1.6	Reading Sensors	915
20.2	Tessel	916
20.2.1	Technical Information	916
20.2.2	Connection and Installation	917

20.2.3	Controlling LEDs	917
20.2.4	Programming the Push Buttons	919
20.2.5	Extending the Tessel with Modules	920
20.3	BeagleBone Black	921
20.3.1	Technical Information	921
20.3.2	Connection and Installation	922
20.3.3	Controlling LEDs	923
20.4	Arduino	924
20.4.1	The Firmata Protocol	924
20.4.2	Connection and Installation	925
20.4.3	The Johnny Five Node.js Module	925
20.5	Cylon.js	927
20.5.1	Controlling the BeagleBone Black with Cylon.js	927
20.5.2	Controlling the Tessel Board with Cylon.js	928
20.5.3	Controlling an Arduino with Cylon.js	928
20.6	Summary	929

21 Establishing a Professional Development Process 931

21.1	Automating Tasks	931
21.1.1	Automating Tasks with Grunt	931
21.1.2	Automating Tasks with Gulp	935
21.2	Automated Testing of Source Code	936
21.2.1	The Principle of Automated Tests	936
21.2.2	The Principle of Test-Driven Development	937
21.2.3	Automated Testing of Source Code with QUnit	939
21.2.4	Automated Testing of Source Code with mocha	945
21.3	Source Code Version Management	949
21.3.1	Introduction to Version Management	949
21.3.2	Installing and Configuring the Git Version Control System	953
21.3.3	Creating a New Local Repository	954
21.3.4	Cloning an Existing Repository	954
21.3.5	Transferring Changes to the Staging Area	955
21.3.6	Transferring Changes to the Local Repository	956
21.3.7	The Different States in Git	957
21.3.8	Transferring Changes to the Remote Repository	958
21.3.9	Transferring Changes from the Remote Repository	959

21.3.10 Working in a New Branch	960
21.3.11 Adopting Changes from a Branch	961
21.3.12 Overview of the Most Important Commands and Terms	962
21.4 Summary	965
The Author	967
Index	969