

LEARNING MADE EASY



# JavaScript<sup>®</sup>

ALL-IN-ONE

for  
**dummies**<sup>®</sup>  
A Wiley Brand

```
document.querySelector(
  const link of document.querySelector(
link.addEventListener('click', (e) =>
  e.preventDefault();
  selectBook(e.target.innerText).then((final
    console.log(`I learned ${finalResult}!`);
  });
});
});
}

async function selectBook(subject) {
  if (subject === 'JavaScript') {
    let { book } = await import('/modules/learn-
    return book;
  } else if (subject === 'ReactJS') {
    let { book } = await import('/modules/learn-re
    return book;
  } else if (subject === 'Vue.js') {
    let { book } = await import('/modules/learn
    return book;
  } else if (subject === 'NodeJS') {
    let { book } = await import('/modul
    return book;
  }
}
```



**Chris Minnick**

Author of *Coding with JavaScript  
For Dummies*





# JavaScript<sup>®</sup>

ALL-IN-ONE

by Chris Minnick

for  
**dummies**<sup>®</sup>  
A Wiley Brand

## JavaScript® All-in-One For Dummies®

Published by: **John Wiley & Sons, Inc.**, 111 River Street, Hoboken, NJ 07030-5774, [www.wiley.com](http://www.wiley.com)

Copyright © 2023 by John Wiley & Sons, Inc., Hoboken, New Jersey

Published simultaneously in Canada

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without the prior written permission of the Publisher. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at <http://www.wiley.com/go/permissions>.

**Trademarks:** Wiley, For Dummies, the Dummies Man logo, Dummies.com, Making Everything Easier, and related trade dress are trademarks or registered trademarks of John Wiley & Sons, Inc., and may not be used without written permission. All other trademarks are the property of their respective owners. John Wiley & Sons, Inc., is not associated with any product or vendor mentioned in this book.

LIMIT OF LIABILITY/DISCLAIMER OF WARRANTY: WHILE THE PUBLISHER AND AUTHORS HAVE USED THEIR BEST EFFORTS IN PREPARING THIS WORK, THEY MAKE NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE ACCURACY OR COMPLETENESS OF THE CONTENTS OF THIS WORK AND SPECIFICALLY DISCLAIM ALL WARRANTIES, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. NO WARRANTY MAY BE CREATED OR EXTENDED BY SALES REPRESENTATIVES, WRITTEN SALES MATERIALS OR PROMOTIONAL STATEMENTS FOR THIS WORK. THE FACT THAT AN ORGANIZATION, WEBSITE, OR PRODUCT IS REFERRED TO IN THIS WORK AS A CITATION AND/OR POTENTIAL SOURCE OF FURTHER INFORMATION DOES NOT MEAN THAT THE PUBLISHER AND AUTHORS ENDORSE THE INFORMATION OR SERVICES THE ORGANIZATION, WEBSITE, OR PRODUCT MAY PROVIDE OR RECOMMENDATIONS IT MAY MAKE. THIS WORK IS SOLD WITH THE UNDERSTANDING THAT THE PUBLISHER IS NOT ENGAGED IN RENDERING PROFESSIONAL SERVICES. THE ADVICE AND STRATEGIES CONTAINED HEREIN MAY NOT BE SUITABLE FOR YOUR SITUATION. YOU SHOULD CONSULT WITH A SPECIALIST WHERE APPROPRIATE. FURTHER, READERS SHOULD BE AWARE THAT WEBSITES LISTED IN THIS WORK MAY HAVE CHANGED OR DISAPPEARED BETWEEN WHEN THIS WORK WAS WRITTEN AND WHEN IT IS READ. NEITHER THE PUBLISHER NOR AUTHORS SHALL BE LIABLE FOR ANY LOSS OF PROFIT OR ANY OTHER COMMERCIAL DAMAGES, INCLUDING BUT NOT LIMITED TO SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR OTHER DAMAGES.

For general information on our other products and services, please contact our Customer Care Department within the U.S. at 877-762-2974, outside the U.S. at 317-572-3993, or fax 317-572-4002. For technical support, please visit <https://hub.wiley.com/community/support/dummies>.

Wiley publishes in a variety of print and electronic formats and by print-on-demand. Some material included with standard print versions of this book may not be included in e-books or in print-on-demand. If this book refers to media such as a CD or DVD that is not included in the version you purchased, you may download this material at <http://booksupport.wiley.com>. For more information about Wiley products, visit [www.wiley.com](http://www.wiley.com).

Library of Congress Control Number: 2023933800

ISBN 978-1-119-90683-4 (pbk); ISBN 978-1-119-90684-1 (ebk); ISBN 978-1-119-90685-8 (ebk)

# Contents at a Glance

<b>Introduction</b> .....	1
<b>Book 1: JavaScript Fundamentals</b> .....	7
CHAPTER 1: Jumping into JavaScript .....	9
CHAPTER 2: Filling Your JavaScript Toolbox .....	41
CHAPTER 3: Using Data .....	63
CHAPTER 4: Working with Operators and Expressions .....	81
CHAPTER 5: Controlling Flow .....	91
CHAPTER 6: Using Arrays .....	105
CHAPTER 7: Making and Using Objects .....	125
CHAPTER 8: Writing and Running Functions .....	139
CHAPTER 9: Getting Oriented with Classes .....	167
CHAPTER 10: Making Things Happen with Events .....	181
CHAPTER 11: Writing Asynchronous JavaScript .....	197
CHAPTER 12: Using JavaScript Modules .....	223
<b>Book 2: Meet Your Web Browser</b> .....	231
CHAPTER 1: What a Web Browser Does .....	233
CHAPTER 2: Programming the Browser .....	241
<b>Book 3: React</b> .....	261
CHAPTER 1: Getting Started with React .....	263
CHAPTER 2: Writing JSX .....	281
CHAPTER 3: Building React Components .....	297
CHAPTER 4: Using Data and Events in React .....	327
<b>Book 4: Vue</b> .....	341
CHAPTER 1: Getting an Overview of Vue .....	343
CHAPTER 2: Introducing Vue Components .....	357
CHAPTER 3: Making Vue Templates .....	369
CHAPTER 4: Using Data and Reactivity .....	395
CHAPTER 5: Responding to Events .....	411
<b>Book 5: Svelte</b> .....	421
CHAPTER 1: Getting Started with Svelte .....	423
CHAPTER 2: Building with Components .....	437
CHAPTER 3: Designing Templates .....	451

CHAPTER 4: Using Directives .....	463
CHAPTER 5: Using the Component Lifecycle .....	471
CHAPTER 6: Advanced Svelte Reactivity .....	483
<b>Book 6: Sharpening Your Tools</b> .....	493
CHAPTER 1: Building from Scratch .....	495
CHAPTER 2: Optimizing and Bundling .....	513
CHAPTER 3: Testing Your JavaScript .....	535
<b>Book 7: Node.js</b> .....	557
CHAPTER 1: Node.js Fundamentals .....	559
CHAPTER 2: Streaming .....	587
CHAPTER 3: Working with Buffers .....	603
CHAPTER 4: Accessing the File System .....	613
CHAPTER 5: Networking with Node .....	627
CHAPTER 6: Using Events .....	639
CHAPTER 7: Error Handling and Debugging .....	651
CHAPTER 8: Accessing Databases .....	673
CHAPTER 9: Riding on the Express Train .....	697
CHAPTER 10: Registration and Authentication .....	721
<b>Index</b> .....	755

# Table of Contents

<b>INTRODUCTION</b> .....	1
Why This Book? .....	2
JavaScript is a huge topic .....	2
How this book is different .....	3
Conventions Used in This Book.....	4
Foolish Assumptions.....	5
Icons Used in This Book .....	5
Beyond the Book .....	6
Where to Go from Here .....	6
<b>BOOK 1: JAVASCRIPT FUNDAMENTALS</b> .....	7
<b>CHAPTER 1: Jumping into JavaScript</b> .....	9
JavaScript, the Basics.....	9
JavaScript is a programming language.....	10
A short and epic history of JavaScript.....	12
Reading and Copying JavaScript Code .....	15
How the web works.....	15
Front end and back end .....	16
The front end is open, the back end is closed.....	18
The value of a service .....	21
JavaScript on the server .....	22
Starting Your Development Environment .....	22
Installing Visual Studio Code .....	23
Learning to use Visual Studio Code .....	25
Writing Your First JavaScript Program .....	29
JavaScript is made of statements .....	30
JavaScript is case-sensitive.....	31
JavaScript ignores white space .....	31
JavaScript programmers use camelCase and underscores.....	32
Running Code in the Console .....	33
Rerunning Commands in the Console.....	34
Running Code in a Browser Window .....	35
Running JavaScript from HTML event attributes.....	35
Running HTML inside script elements .....	36
Including JavaScript files in your HTML .....	37
<b>CHAPTER 2: Filling Your JavaScript Toolbox</b> .....	41
Installing Node.js .....	43
Configuring Visual Studio Code.....	43
Getting prettier .....	44
Installing Live Server .....	47

Documenting Your Code . . . . .	51
Line comments . . . . .	51
Block comments . . . . .	51
The README file. . . . .	52
The basics of Markdown. . . . .	53
Coding Responsibly with Git . . . . .	55
Introducing Git. . . . .	55
Installing Git . . . . .	56
Configuring and testing Git . . . . .	57
Learning the basics of Git. . . . .	60
Moving forward with Git and GitHub . . . . .	62
<b>CHAPTER 3: Using Data . . . . .</b>	<b>63</b>
Making Variables with let . . . . .	63
Declaring variables . . . . .	64
Initializing variables. . . . .	64
Using variables. . . . .	64
Naming variables. . . . .	66
Making Constants with const. . . . .	66
When to use constants. . . . .	67
Naming constants . . . . .	67
Taking a Look at the Data Types . . . . .	68
JavaScript is loose and dynamic . . . . .	68
Passing by value. . . . .	69
String data type . . . . .	69
Number data type . . . . .	73
BigInt data type . . . . .	74
Boolean data type . . . . .	75
NaN data type . . . . .	77
Undefined data type . . . . .	77
Symbol data type. . . . .	77
Wrangling the Object: The Complex Data Type . . . . .	78
Examining the Array — a Special Kind of Object . . . . .	79
Getting a Handle on Scope. . . . .	80
<b>CHAPTER 4: Working with Operators and Expressions . . . . .</b>	<b>81</b>
Building Expressions. . . . .	82
Operators: The Lineup . . . . .	83
Operator precedence . . . . .	83
Using parentheses. . . . .	84
Assignment operators . . . . .	85
Comparison operators . . . . .	85
Arithmetic operators. . . . .	86
Concatenation operator . . . . .	86

	Logical operators . . . . .	87
	Combining operators . . . . .	89
	Other Operators . . . . .	89
<b>CHAPTER 5:</b>	<b>Controlling Flow . . . . .</b>	<b>91</b>
	Choosing a Path . . . . .	91
	if . . . else statements . . . . .	91
	Switch statements . . . . .	93
	Making Loops . . . . .	97
	for loops . . . . .	97
	for . . . in loops . . . . .	98
	for . . . of loops . . . . .	100
	while loops . . . . .	100
	do . . . while loops . . . . .	101
	break and continue statements . . . . .	102
<b>CHAPTER 6:</b>	<b>Using Arrays . . . . .</b>	<b>105</b>
	Introducing Arrays . . . . .	106
	Creating Arrays . . . . .	108
	Using the Array() constructor . . . . .	108
	Using array literal notation . . . . .	109
	Using the split function . . . . .	109
	Accessing Array Elements . . . . .	110
	Modifying Arrays . . . . .	111
	Deleting Array Elements . . . . .	111
	Programming with Array Methods . . . . .	112
	Pushing and popping . . . . .	113
	Shifting and unshifting . . . . .	114
	Slicing an array . . . . .	116
	Splicing an array . . . . .	116
	Looping with Array Methods . . . . .	117
	Passing callback functions to array methods . . . . .	117
	Reducing an array . . . . .	118
	Mapping an array . . . . .	119
	Filtering arrays . . . . .	121
	Destructuring Arrays . . . . .	122
	Spreading Arrays . . . . .	122
<b>CHAPTER 7:</b>	<b>Making and Using Objects . . . . .</b>	<b>125</b>
	Objects: The Basics . . . . .	125
	Creating Objects . . . . .	127
	Making objects using literal notation . . . . .	127
	Making objects using a constructor function . . . . .	128
	Making objects with class . . . . .	128
	Making objects with Object.create() . . . . .	130

Modifying Objects . . . . .	130
Using dot notation . . . . .	130
Using square brackets notation . . . . .	131
Comparing and Copying Objects . . . . .	132
Understanding Prototypes . . . . .	134
Deleting Object Properties . . . . .	138
<b>CHAPTER 8: Writing and Running Functions . . . . .</b>	<b>139</b>
Functions: An Introduction . . . . .	140
Using Top-level functions . . . . .	141
Using methods of built-in objects . . . . .	142
Passing by value . . . . .	142
Passing by reference . . . . .	143
Writing Functions . . . . .	144
Naming functions . . . . .	144
Passing arguments . . . . .	145
Writing a function body . . . . .	148
Returning data . . . . .	149
Function declaration scope and hoisting . . . . .	150
Declaring Anonymous functions . . . . .	150
Defining function expressions . . . . .	150
Writing anonymous functions as arrow functions . . . . .	152
Simplifying arrow functions . . . . .	152
Knowing the limits of arrow functions . . . . .	154
Writing Methods . . . . .	154
Understanding Context and this . . . . .	155
Passing an object to a function . . . . .	156
Setting the context of a function . . . . .	157
Passing a function from one object to another . . . . .	158
Passing a function to a child to change the parent . . . . .	159
Chaining Functions . . . . .	165
<b>CHAPTER 9: Getting Oriented with Classes . . . . .</b>	<b>167</b>
Encapsulation . . . . .	168
Abstraction . . . . .	168
Inheritance . . . . .	169
Polymorphism . . . . .	169
Base Classes . . . . .	169
Recognizing that classes aren't hoisted . . . . .	170
Using class expressions . . . . .	170
Making instances of base classes . . . . .	170
Derived Classes . . . . .	170
Constructors . . . . .	171

Properties and Methods. . . . .	173
Creating methods in a class. . . . .	173
Overriding methods in a derived class. . . . .	174
Defining methods, properties, and fields . . . . .	174
Practicing and Becoming comfortable with Classes. . . . .	177
<b>CHAPTER 10: Making Things Happen with Events. . . . .</b>	<b>181</b>
Understanding the JavaScript Runtime Model . . . . .	182
Stacking function calls. . . . .	182
Heaping objects. . . . .	183
Queuing messages . . . . .	183
The Event Loop . . . . .	183
JavaScript is single-threaded . . . . .	183
Messages run until they're done. . . . .	184
Listening for Events. . . . .	184
Listening with HTML event attributes. . . . .	185
Listening with Event handler properties . . . . .	185
Using addEventListener() . . . . .	186
The Event object . . . . .	190
Listening on multiple targets. . . . .	190
Dispatching events programmatically . . . . .	192
Removing event listeners . . . . .	194
Preventing default actions . . . . .	195
<b>CHAPTER 11: Writing Asynchronous JavaScript. . . . .</b>	<b>197</b>
Understanding Asynchronous JavaScript. . . . .	198
Reading synchronous code . . . . .	198
Events to the rescue . . . . .	199
Calling you back. . . . .	200
Making Promises . . . . .	202
Writing promises . . . . .	203
Introducing async functions. . . . .	206
Using AJAX. . . . .	210
Getting data with the Fetch API . . . . .	211
Calling other Response methods . . . . .	213
Handling fetch() errors . . . . .	214
The fetch init object. . . . .	215
Introducing HTTP. . . . .	216
Making requests with CORS. . . . .	218
Working with JSON data . . . . .	220
Getting JSON data . . . . .	222
Sending JSON data. . . . .	222

<b>CHAPTER 12: Using JavaScript Modules</b> .....	223
Defining Modules .....	224
Exporting Modules .....	224
Named exports .....	225
Default exports .....	225
Importing Modules .....	227
Importing named modules .....	227
Importing default modules .....	227
Renaming Exports and Imports .....	228
Importing a Module Object .....	228
Loading Dynamic Modules .....	229
Importing Modules into HTML .....	229
 <b>BOOK 2: MEET YOUR WEB BROWSER</b> .....	 231
<b>CHAPTER 1: What a Web Browser Does</b> .....	233
Interfacing with a Browser .....	234
Introducing the Browser Engine .....	235
The Rendering Engine .....	235
The JavaScript engine .....	237
Identifying and preventing render blocking .....	237
Unblocking your code with async and defer .....	238
Networking .....	239
Data storage .....	239
 <b>CHAPTER 2: Programming the Browser</b> .....	 241
Understanding Web APIs and Interfaces .....	241
Hooking into interfaces .....	242
Built-in browser APIs .....	242
Custom APIs .....	243
Getting Around the Navigator .....	243
Inspecting the navigator's quirks .....	244
Navigator properties .....	244
Stealing a Glimpse Through the Window .....	246
Window properties .....	247
Window methods .....	247
Introducing the HTML DOM .....	249
Document properties .....	249
Document methods .....	250
Selecting element nodes .....	251
Selecting with getElementById() .....	252
Selecting using selectors .....	252
Creating and adding elements to the DOM .....	254
Element nodes .....	255
Element methods .....	256
Knowing Your History .....	257

<b>BOOK 3: REACT</b> .....	261
<b>CHAPTER 1: Getting Started with React</b> .....	263
Understanding ReactJS .....	263
Distilling “Thinking in React” .....	264
Building a React UI .....	265
React is component-based .....	270
React is declarative .....	271
React is just JavaScript .....	271
Initializing a Project with Vite .....	271
Introducing Vite .....	272
Launching the VS Code terminal .....	272
Touring the structure of a React app .....	274
Modifying a React project .....	277
Introducing ReactDOM and the Virtual DOM .....	278
<b>CHAPTER 2: Writing JSX</b> .....	281
Learning the Fundamentals of JSX .....	281
JSX is not HTML .....	282
JSX is XML .....	282
Transpiling with Babel .....	283
Writing HTML output with JSX .....	283
Using built-in components .....	284
Using JavaScript Expressions in JSX .....	285
Conditionally Rendering JSX .....	286
Conditional rendering with element variables .....	287
Conditional rendering with && .....	288
Conditional rendering with the conditional operator .....	289
Making a List .....	290
Styling React Apps and Components .....	292
Adding global styles .....	292
Using local styles .....	293
Using the style attribute .....	293
Using style objects .....	294
Making style modules .....	295
Other style strategies .....	296
<b>CHAPTER 3: Building React Components</b> .....	297
Thinking in Components .....	298
Designing your own elements .....	298
Returning valid values .....	300
Recognizing the Two Types of Data .....	300
Props .....	300
Getting reactive with state .....	301
How state enables reactivity .....	304

Function Components . . . . .	304
Functions are stateless. . . . .	305
Introducing useState(). . . . .	305
Class Components. . . . .	307
The Component Lifecycle. . . . .	309
The mounting methods . . . . .	309
The updating methods. . . . .	313
Unmounting a component. . . . .	315
Using the Lifecycle in Function Components . . . . .	316
Running effects less often . . . . .	317
Performing an effect on unmounting. . . . .	319
Composing Components . . . . .	321
How inheritance works in object-oriented programming. . . . .	321
Composition using explicit props . . . . .	322
Composition using the children prop. . . . .	323
Composition with custom hooks. . . . .	324
<b>CHAPTER 4: Using Data and Events in React . . . . .</b>	<b>327</b>
Event Handling in React . . . . .	328
Using event attributes. . . . .	328
Dispatching Synthetic Events. . . . .	328
Specifying a handler function . . . . .	329
Passing arguments to an event handler . . . . .	329
Passing functions as props . . . . .	330
Making Forms with React. . . . .	334
Using controlled inputs. . . . .	335
Using uncontrolled forms . . . . .	337
<b>BOOK 4: VUE. . . . .</b>	<b>341</b>
<b>CHAPTER 1: Getting an Overview of Vue . . . . .</b>	<b>343</b>
Comparing Vue to React. . . . .	343
Scaffolding Your First Vue.js Application . . . . .	344
Bootstrapping with vue-create . . . . .	344
Installing Volar . . . . .	346
Exploring the structure of a Vue app . . . . .	347
Going to the src . . . . .	348
Mounting a Root Component . . . . .	348
Configuring an app . . . . .	349
Mounting multiple apps . . . . .	349
Exploring Vue's Two Styles. . . . .	350
The Options API. . . . .	350
The Composition API. . . . .	352
Deciding which API to use . . . . .	352
Installing Vue DevTools. . . . .	353

<b>CHAPTER 2: Introducing Vue Components</b> .....	357
Introducing the Single-File Component .....	357
The script element .....	358
The setup() function .....	358
The setup shortcut .....	360
Naming Components .....	362
Following the Component Lifecycle .....	362
onMounted() .....	362
onUpdated() .....	365
onUnmounted() .....	365
onBeforeMount() .....	366
onBeforeUpdate() .....	366
onBeforeUnmount() .....	366
onErrorCaptured() .....	366
Handling Errors in Components .....	366
<b>CHAPTER 3: Making Vue Templates</b> .....	369
Writing HTML Templates .....	369
Using JavaScript in Templates .....	371
Calling functions .....	372
JavaScript in templates is restricted .....	372
Using globals in templates .....	373
Introducing Directives .....	373
Built-in directives .....	373
Directive shorthand names .....	375
Passing arguments to directives .....	375
Dynamic arguments .....	375
Directive modifiers .....	375
Custom directives .....	376
Conditional Rendering .....	378
Conditional rendering with JavaScript .....	378
Conditional rendering using directives .....	378
Rendering Lists .....	379
Using v-for with numbers and strings .....	380
Using v-for with objects .....	380
Using v-for with arrays .....	382
Specifying a key .....	383
Composing with Slots .....	383
Specifying fallback content .....	385
Naming slots .....	386
Adding Style to Components .....	387
Global CSS .....	387
Scoped CSS .....	388

	Multiple style blocks .....	389
	CSS modules .....	390
	v-bind in CSS .....	391
<b>CHAPTER 4:</b>	<b>Using Data and Reactivity</b> .....	395
	Passing and Using Props .....	396
	Defining props with <script setup> .....	396
	Defining props with setup() .....	397
	Binding Data to Templates .....	398
	Initializing and Changing Reactive Data .....	400
	reactive() creates a Proxy object .....	401
	Limitations of reactive() .....	402
	Introducing ref() .....	403
	Experimenting with Reactivity Transform .....	404
	Computing Properties .....	405
	Reacting to State Changes with Watch() .....	408
<b>CHAPTER 5:</b>	<b>Responding to Events</b> .....	411
	Setting Listeners with v-on .....	411
	Inline handlers .....	411
	Method handlers .....	413
	Choosing between method and inline handlers .....	414
	Using Event Modifiers .....	416
	Using key modifiers .....	417
	Detecting exact combinations .....	419
	Binding Forms to Events and Data .....	419
	Making two-way bindings with v-model .....	420
	Using v-model with various input types .....	420
	<b>BOOK 5: SVELTE</b> .....	421
<b>CHAPTER 1:</b>	<b>Getting Started with Svelte</b> .....	423
	What Makes Svelte Different? .....	423
	Building Your Scaffolding .....	425
	Getting the Svelte for VS Code Extension .....	426
	Exploring a Svelte App .....	427
	Playing with Svelte .....	431
	Building the basic look-and-feel .....	431
	Making reactive data .....	432
	Handling the event .....	435
<b>CHAPTER 2:</b>	<b>Building with Components</b> .....	437
	Writing Lean Components .....	437
	Identifying What's in a Component .....	439
	Scripting in Svelte components .....	439
	Exporting and using props .....	440

	Triggering reactivity with assignments . . . . .	442
	Recognizing that array methods don't trigger updates . . . . .	442
	Creating reactive statements . . . . .	445
	Using <code>&lt;script&gt;</code> data and functions . . . . .	446
	Adding Style to a Component . . . . .	447
<b>CHAPTER 3:</b>	<b>Designing Templates . . . . .</b>	<b>451</b>
	Elements Are the Building Blocks . . . . .	451
	Using the built-in elements . . . . .	451
	Using custom elements . . . . .	452
	Documenting Svelte with Comments . . . . .	453
	Choosing a Path . . . . .	454
	Creating Loops . . . . .	456
	Writing Text Expressions . . . . .	458
	Composing with Slots . . . . .	459
<b>CHAPTER 4:</b>	<b>Using Directives . . . . .</b>	<b>463</b>
	Listening for Events with <code>on:</code> . . . . .	463
	Basic event handling . . . . .	464
	Attaching modifiers to event listeners . . . . .	464
	Forwarding events . . . . .	465
	Handling multiple events . . . . .	466
	Creating Two-Way Bindings with <code>:bind</code> . . . . .	466
	Recognizing that number inputs create numbers . . . . .	467
	Binding select inputs . . . . .	467
	Using Transition Animations . . . . .	468
	Creating your first transition . . . . .	468
	Passing arguments to transitions . . . . .	470
	Creating unidirectional transitions . . . . .	470
<b>CHAPTER 5:</b>	<b>Using the Component Lifecycle . . . . .</b>	<b>471</b>
	The Svelte Lifecycle . . . . .	471
	Mounting . . . . .	472
	Using <code>beforeUpdate()</code> and <code>afterUpdate()</code> . . . . .	473
	Using <code>onDestroy()</code> . . . . .	473
	Getting ticks . . . . .	474
	Fetching Data in Svelte . . . . .	476
	Refreshing data . . . . .	477
	Awaiting asynchronous requests . . . . .	479
<b>CHAPTER 6:</b>	<b>Advanced Svelte Reactivity . . . . .</b>	<b>483</b>
	Constructing and Stocking the Store . . . . .	483
	Creating a writable store . . . . .	484
	Creating a readable store . . . . .	484

Subscribing to a store . . . . .	485
Unsubscribing from a store . . . . .	485
Setting and updating a store . . . . .	486
Using the reactive shortcut . . . . .	487
Store starting and stopping functions . . . . .	488
Getting and Setting Context . . . . .	490
<b>BOOK 6: SHARPENING YOUR TOOLS . . . . .</b>	<b>493</b>
<b>CHAPTER 1: Building from Scratch . . . . .</b>	<b>495</b>
Why You Need a Build Tool . . . . .	496
“Back in my day . . .” . . . . .	496
The road to dependency hell . . . . .	497
Enter package management . . . . .	497
Managing Dependencies with npm . . . . .	497
Initializing a project . . . . .	497
Learning the parts of package.json . . . . .	500
Using the node_modules folder . . . . .	502
Writing Your First Files . . . . .	503
Writing a dev Script . . . . .	505
Making Modules . . . . .	506
Refactoring index.js . . . . .	506
The moveBall() function . . . . .	507
The generateMap() function . . . . .	508
Adding style . . . . .	509
Testing for collisions . . . . .	510
Testing it out . . . . .	511
<b>CHAPTER 2: Optimizing and Bundling . . . . .</b>	<b>513</b>
Automating Your Build Script . . . . .	513
Installing and using a module bundler . . . . .	514
Configuring your dev server . . . . .	515
Building it up . . . . .	518
Copying static assets . . . . .	519
Cleaning up . . . . .	520
Converting to React . . . . .	522
Configuring Webpack for React . . . . .	522
Converting the UI to React . . . . .	524
Detecting collisions . . . . .	531
<b>CHAPTER 3: Testing Your JavaScript . . . . .</b>	<b>535</b>
Using a Linter . . . . .	536
Debugging in Chrome . . . . .	542
Getting started with the Sources panel . . . . .	542
Building a source map . . . . .	542

Editing your code in the Sources panel . . . . .	544
Setting breakpoints . . . . .	544
Using watch expressions . . . . .	546
Unit Testing . . . . .	547
Installing and configuring Jest . . . . .	547
Writing your first test . . . . .	548
Learning how Jest works . . . . .	550
Writing better code through testing . . . . .	552
Using testing-library . . . . .	553
<b>BOOK 7: NODE.JS . . . . .</b>	<b>557</b>
<b>CHAPTER 1: Node.js Fundamentals . . . . .</b>	<b>559</b>
Learning What Makes Node.js Tick . . . . .	560
Node.js is not a programming language . . . . .	560
Node.js is not a framework . . . . .	560
Node.js is a runtime environment . . . . .	560
Why developers need Node.js . . . . .	561
Learning the Parts of Node.js . . . . .	562
The V8 engine . . . . .	563
libuv . . . . .	563
Node.js bindings . . . . .	564
The Node.js standard library . . . . .	564
Introducing the Node.js Core Modules . . . . .	564
Recognizing What Node.js Is Good For . . . . .	566
Why is Node.js so fast? . . . . .	566
What is Node.js not good at? . . . . .	567
Working with Node.js . . . . .	567
Writing a Node.js program . . . . .	568
Monitoring your script . . . . .	569
Running a code on the command line . . . . .	570
Using REPL . . . . .	570
Playing with the Node.js REPL . . . . .	570
Working with REPL commands . . . . .	573
Making and Using Node.js Modules . . . . .	575
Using CommonJS . . . . .	575
Using ES modules . . . . .	577
Setting the module type in package.json . . . . .	579
Getting Data to Node Modules . . . . .	580
Environment variables . . . . .	581
Passing arguments . . . . .	584
Node's Callback Pattern . . . . .	584

<b>CHAPTER 2: Streaming</b> .....	587
Jumping into Streams .....	587
Chunking is the key .....	588
Loading without streams .....	588
Converting to streams .....	589
Viewing chunks .....	590
Identifying types of streams .....	592
Creating Readable Streams .....	592
Reading readable streams from the fs module .....	593
Distinguishing between the two read modes .....	594
Creating Writable Streams .....	596
Producing Duplex Streams .....	598
Backpressure .....	598
PassThrough .....	599
Transforming Streams .....	599
Chaining Streams .....	600
<b>CHAPTER 3: Working with Buffers</b> .....	603
Knowing Your Buffer Basics .....	603
Differentiating between encoding and decoding .....	604
Examining buffer content .....	605
Decoding Buffers .....	606
Creating Buffers .....	607
Using Other Buffer Methods .....	609
Iterating over Buffers .....	611
<b>CHAPTER 4: Accessing the File System</b> .....	613
Importing the fs module .....	613
Reading Files .....	614
Reading from a file with fs.read() .....	614
Using readFile() .....	617
Using readFileSync() .....	618
Writing Files .....	619
Writing it to disk with fs.write() .....	619
Using fs.writeFile() .....	620
Using Paths .....	621
Getting File and Directory Information .....	623
Listing the files in a directory .....	623
Finding directories .....	624
Getting file stats .....	624

<b>CHAPTER 5: Networking with Node</b> .....	627
A Note about Security .....	628
Making a Web Server .....	630
Understanding the Request object .....	632
Understanding the response object .....	633
Methods of the response object .....	636
Using implicit headers .....	637
Knowing the differences between setHeader() and writeHead() .....	637
<b>CHAPTER 6: Using Events</b> .....	639
Introducing EventEmitter .....	640
Creating custom events .....	640
Extending EventEmitter .....	641
Passing arguments to listeners .....	642
Understanding and Using maxListeners .....	644
Finding the value of defaultMaxListeners .....	644
Exceeding the maximum listeners for an emitter .....	645
Increasing the maximum number of listeners .....	646
Removing Listeners .....	647
Removing individual listeners .....	647
Removing all listeners .....	648
Emitting Once .....	649
<b>CHAPTER 7: Error Handling and Debugging</b> .....	651
Knowing the Types of Errors .....	651
Operational errors .....	652
Programmer errors .....	652
Understanding Node.js's Error Object .....	653
Reading error.stack .....	654
Reading a stack frame .....	654
Exceptions versus Errors .....	654
Handling Exceptions .....	655
Catching exceptions with promises .....	656
Catching exceptions with async functions .....	659
Debugging Node.js Programs .....	660
Using the command-line debugger .....	663
Debugging in Chrome DevTools .....	667
<b>CHAPTER 8: Accessing Databases</b> .....	673
Getting Started with MongoDB .....	674
Discerning between relational and NoSQL databases .....	674
Installing MongoDB .....	677

Starting MongoDB.....	680
Starting MongoDB on Windows .....	681
Starting MongoDB on macOS .....	681
Using Mongosh .....	681
Connecting to MongoDB and creating a database.....	681
Creating a collection .....	682
Making an id and listing documents.....	682
Finding documents .....	684
Learning MongoDB Shell commands.....	686
Working with users and roles .....	687
Using MongoDB from Node.js.....	688
Installing the Node.js driver.....	689
Connecting to a MongoDB server.....	689
Inserting documents into a collection .....	691
Getting data .....	693
Updating data .....	695
Deleting data .....	696
<b>CHAPTER 9: Riding on the Express Train.....</b>	<b>697</b>
Installing Express.....	697
Server-Side Routing with Express .....	699
Introducing routing methods .....	699
Using routing methods.....	699
Using Express Middleware.....	701
The next() function .....	702
Types of middleware.....	702
Serving static files .....	706
Analyzing a Complete Express Server .....	707
Installing the server and dependencies.....	709
Setting up a REST client .....	709
Testing the API server.....	711
Serving a View .....	713
Benefiting from a template engine.....	713
Introducing Pug.....	714
Using the Express Application Generator .....	715
<b>CHAPTER 10: Registration and Authentication .....</b>	<b>721</b>
Making and Configuring the Directory.....	722
Adding the App and Server Modules .....	723
Making Some Basic Routes .....	725
Testing Your Routes .....	728

Making a Schema with Mongoose .....	730
Using mongoose.Schema and mongoose.model .....	730
Installing Mongoose and connecting to a database.....	731
Creating the User model .....	733
Create the post model .....	734
Implementing User Registration.....	734
Understanding the basics of authentication.....	735
Programming the user sign-up route.....	735
Understanding password security .....	736
Understanding hashes .....	736
Adding salt to hashing .....	738
Hashing and saving .....	738
Testing user registration.....	740
Handling Authentication.....	741
Generating and Using Tokens.....	744
Recognizing that tokens must expire .....	744
Sending a refresh token .....	745
Finishing the Login Route.....	746
Testing the login route .....	749
Looking at an access token .....	750
Using an access token.....	751
<b>INDEX.....</b>	<b>755</b>



# Introduction

---

*“Do your work, and you shall reinforce yourself.”*

—RALPH WALDO EMERSON

**T**he following scenario plays out at some point in the career of every developer working on mobile or web apps today:

You show up at your new job (or new freelance gig). You were good enough at using JavaScript that you got hired, and now you’re feeling confident that you can start making contributions quickly.

The HR manager gives you a tour and brings you to your new desk — or maybe you work at home or at your local coffee shop. You log in to your new email account and see a message from your manager:

Welcome to the team! Normally, we’d give you a few weeks to get settled in, but we’re slammed and I’m hoping you can get working on some code right away.

This morning, please log in to Slack and clone our repo, and we’ll get you started on some low-level tasks. A couple of things you should know first are that we use a feature branch workflow, we enforce Google style with ESLint, and we unit-test with Mocha and Chai. Everything is Jamstack with MERN on the back end. The API isn’t fully documented yet, but it’s REST, and you should be able to figure out the endpoints by poking around with Postman. Here are a few small tasks where we could use your help today:

- Figure out why the refresh token isn’t getting set in some cases.
- Our code splitting is suboptimal. Take a look at the webpack config and see whether you can improve the situation.
- Speaking of suboptimal, we have some unnecessary Sass in the login module that may be slowing down our build and load times and should be converted to CSS Modules.
- Our ticker widget still relies on jQuery (ha-ha!), and we need to refactor that out before we ship so that we don’t embarrass ourselves.

Okay, that should be enough for today. We have our weekly stand-up tomorrow morning, and we can get you started on something more important then. Don’t hesitate to reach out if you have questions!

Your palms start to sweat as you reread the message. You were told you'd be working with JavaScript. What's all this stuff? Sass? You know how to deal with unnecessary sass from your dog. . . .

You grab on to the part about jQuery. You recall reading about it in one of your books, and you're pretty sure you understand it. But why does the manager think it's so funny?

You close your email and go to the cafe to order a chai mocha latte and a Jamstack sandwich and have a rest in order to think about whether you really and truly need this job.

## Why This Book?

I wrote this book because I've been in the situation just described numerous times in my career as a full-stack developer. Maybe you've already had an experience like that one and you want to make sure you're better prepared next time. Maybe you're going through this experience right now. Maybe you want to try to prevent this from happening to you as you begin your career in JavaScript programming. If you fit into any of these categories, this book is for you.

Software development is a dynamic craft. A good part of your responsibility as a professional programmer is to learn new things constantly. The world of JavaScript, however, is notoriously dynamic. Something you learn this year will likely be out of style or at least vastly different next year. There's no sense in fighting the current by sticking to your tried-and-true methods and tools. Many times, if not most of the time, new libraries and tools catch on quickly in JavaScript because they're genuinely useful and superior to the previous ones.

Learning new things can be difficult, and learning new things under pressure to begin using them on real-life projects can be particularly stressful unless you're properly prepared with the right attitude and sufficient experience.

My goal with this book is to prepare you to understand and work with JavaScript and JavaScript programmers. If you read this entire book, you'll understand much of what the manager in the scenario I described earlier said — and maybe even what to do about it. You'll also be well-equipped to continue your learning and to ask the right questions when something you haven't heard of comes up next time.

## JavaScript is a huge topic

Programmers create new ways of working with JavaScript all the time and package them as libraries and frameworks that can be downloaded and used by

other programmers. Think of libraries as tools to solve a particular problem in a different way. Think of a framework as a complete system for doing particular things.

No single person can know and remember every JavaScript library and framework that's been created. And there's really no need to. By definition, a JavaScript library or framework is just JavaScript. In theory, if you know JavaScript, you can learn every JavaScript library and framework. In fact, if you know enough about JavaScript, you can write your own JavaScript library or framework!

## How this book is different

Most books about JavaScript focus on either teaching the JavaScript language itself, or on focusing on a single library or framework. Both approaches have their merits, but both leave out a vital ingredient for becoming a JavaScript programmer.

Learning to program with JavaScript isn't fundamentally about memorizing syntax or knowing every function available in a library or framework. If you know and understand the basics, you can look up everything else easily enough. Furthermore, if you spend a lot of time learning every function and feature of a particular library from a book, you'll be frustrated when you go to use it and find that much has changed in the time (whether it's a month or several years) since the book was published.

In my experience, it's much better to learn just enough about as many different ways of doing things as possible, so that when you encounter something new, you'll have something to compare it to.

## Learn JavaScript as it's used

Knowing how to write JavaScript isn't enough to be able to develop apps. JavaScript lives in an environment, whether it's a web browser, a mobile phone, a web server, or a hardware device. Once you get to a certain level of proficiency with JavaScript, knowing how JavaScript interfaces with its environment is what's most important.

This book shows you how JavaScript is used in the real world, using a combination of real-world and simplified examples.

## Understand similarities between the most popular libraries

Learning about a single library — whether it's React or Vue.js or Svelte or Angular — is great. But without knowledge of other ways of doing things, you

may fall into the trap of thinking that every problem can be best solved by the tool you know.

If you learn a wide variety of tools, you'll understand how libraries and frameworks are constantly improving on what's been done before, and you'll gain an appreciation for why change is so important in the JavaScript world.

## Adapt to new technologies

Another benefit of learning multiple ways to do the same thing is that you'll find that the more you learn, the easier it becomes to learn additional tools. One trait of outstanding JavaScript programmers is that they're excited by opportunities to learn new libraries and tools.

# Conventions Used in This Book

This book is designed for readers who have some experience with programming or web development and who want to learn not only JavaScript but also how to apply it. Topics I describe in this book include how to

- » Write JavaScript code using the most modern and up-to-date syntax
- » Use the development tools used by professional JavaScript programmers
- » Build reactive user interfaces with ReactJS
- » Build reactive user interfaces with Vue.js
- » Build reactive user interfaces with Svelte
- » Write server-side JavaScript with Node.js
- » Connect to data sources with Node.js
- » Build a complete back-end application using Node.js
- » Connect a front-end user interface to a Node.js back end

As you read this book, keep the following information in mind:

- » The book can be read from beginning to end, but feel free to skip around, if you like. If a topic interests you, start there. You can always return to earlier chapters, if necessary.
- » At some point, you *will* get stuck and the code you write won't work as intended. Do not fear! You can find many resources to help you, including

support forums, others on the Internet, and me! You can email me directly at [chris@minnick.com](mailto:chris@minnick.com) or message me on Twitter, Mastodon, or through my website ([www.chrisminnick.com](http://www.chrisminnick.com)). Additionally, you can submit a ticket and find additional code at my GitHub repo for this book, at <https://github.com/chrisminnick/javascriptaio>.

- » Code in the book appears in a monospaced font, like this: `<h1>Hi there! </h1>`.

## Foolish Assumptions

I do not make many assumptions about you, the reader, but I do make a few.

- » I assume that you have some experience or familiarity with HTML and CSS. Many of the applications I show you how to develop make use of both fundamental languages of the web to style and structure their output. Many excellent books and tutorials have been written about both topics, and the amount of HTML and CSS knowledge you need can be learned in a day or two.
- » I assume that you have a computer running the latest version of Google Chrome. The examples in this book have been tested and optimized for the Chrome browser, which is available for free from Google. Even so, the examples also work in the latest version of Firefox, Safari, or Microsoft Edge.
- » I assume that you have access to an Internet connection. Many of the examples in this book can be completed without an Internet connection, but some require one.
- » I assume that you can download and install free software to your computer. Oftentimes, the computer you use at work has restrictions on what can be installed by the user. If you use your own computer to develop and run the applications in this book, that will generally work without a problem.

## Icons Used in This Book

Here are the icons used in the book to flag text that should be given extra attention or that you can skip.



TIP

This icon flags useful information or explains a shortcut to help you understand a concept.



TECHNICAL  
STUFF

This icon explains technical details about the concept being explained. The details might be informative or interesting but are not essential to your understanding of the concept at this stage.



REMEMBER

Try not to forget the material marked with this icon. It signals an important concept or process that you should keep in mind.



WARNING

Watch out! This icon flags common mistakes and problems that can be avoided if you heed the warning.

## Beyond the Book

A lot of extra content that you won't find in this book is available at [www.dummies.com](http://www.dummies.com). Go online to find the following:

» **The source code for the examples in this book:** You can find it at <https://www.dummies.com/go/javascriptallinonefd>.

The source code is organized by book and chapter. The best way to work with a chapter is to download all the source code for it at one time.

» **Updates:** Code and specifications are constantly changing, so the commands and syntax that work today may not work tomorrow. You can find any updates or corrections by visiting [www.dummies.com/go/javascriptallinonefd](http://www.dummies.com/go/javascriptallinonefd) or <https://github.com/chrisminnick/javascriptaio>.

» **Cheat Sheet:** The Cheat Sheet offers quick access to useful tips and shortcuts. Just go to [www.dummies.com](http://www.dummies.com) and type **JavaScript All-in-One For Dummies Cheat Sheet** in the Search box.

## Where to Go from Here

All right, now that all the administrative stuff is out of the way, it's time to get started. You can totally do this. Congratulations on taking your first step (or continuing your journey) in the exciting world of JavaScript! Feel free to jump around the book if you're interested in specific topics. If you're a total noob, start with Chapter 1 in Book 1.