

Copyrighted Material

TECH TODAY



# PROFESSIONAL JAVASCRIPT<sup>®</sup> FOR WEB DEVELOPERS

5<sup>TH</sup> EDITION



MATT FRISBIE

FOREWORD BY JOHN HUBBERTS  
Founding Principal Engineer at Roboto

WILEY

Copyrighted Material

Copyrighted Material

PROFESSIONAL  
**JavaScript® for Web Developers**

Fifth Edition

Matt Frisbie

**WILEY**

Copyrighted Material

# CONTENTS

|                                           |              |
|-------------------------------------------|--------------|
| <i>FOREWORD</i>                           | <i>xliii</i> |
| <i>INTRODUCTION</i>                       | <i>xlv</i>   |
| <b>CHAPTER 1: WHAT IS JAVASCRIPT?</b>     | <b>1</b>     |
| A Short History                           | 2            |
| Javascript Implementations                | 3            |
| ECMAScript                                | 3            |
| ECMAScript Editions                       | 4            |
| What Does ECMAScript Conformance Mean?    | 5            |
| The Document Object Model                 | 5            |
| Why the DOM Is Necessary                  | 6            |
| DOM Levels                                | 6            |
| Other DOMs                                | 7            |
| The Browser Object Model                  | 8            |
| Summary                                   | 8            |
| <b>CHAPTER 2: JAVASCRIPT IN HTML</b>      | <b>9</b>     |
| The <code>&lt;Script&gt;</code> Element   | 9            |
| Tag Placement                             | 12           |
| Deferred Scripts                          | 13           |
| Asynchronous Scripts                      | 13           |
| Dynamic Script Loading                    | 14           |
| Inline Code Versus External Files         | 14           |
| Document Modes                            | 15           |
| The <code>&lt;Noscript&gt;</code> Element | 16           |
| Summary                                   | 17           |
| <b>CHAPTER 3: LANGUAGE BASICS</b>         | <b>19</b>    |
| Syntax                                    | 20           |
| Case-Sensitivity                          | 20           |
| Identifiers                               | 20           |
| Comments                                  | 20           |
| Strict Mode                               | 21           |
| Statements                                | 21           |
| Keywords and Reserved Words               | 22           |
| Variables                                 | 22           |

---

|                                       |    |
|---------------------------------------|----|
| The var Keyword                       | 23 |
| var Declaration Scope                 | 23 |
| var Declaration Hoisting              | 24 |
| let Declarations                      | 25 |
| Temporal Dead Zone                    | 26 |
| Global Declarations                   | 26 |
| Conditional Declaration               | 26 |
| let Declaration in for Loops          | 27 |
| const Declarations                    | 28 |
| Declaration Styles and Best Practices | 29 |
| Don't Use var                         | 29 |
| Prefer const Over let                 | 29 |
| Data Types                            | 30 |
| The typeof Operator                   | 30 |
| The Undefined Type                    | 31 |
| The Null Type                         | 32 |
| The Boolean Type                      | 33 |
| The Number Type                       | 34 |
| Floating-Point Values                 | 35 |
| Numeric Separators                    | 36 |
| Range of Values                       | 36 |
| NaN                                   | 37 |
| Number Conversions                    | 38 |
| The BigInt Type                       | 40 |
| Conversion                            | 41 |
| Operators                             | 41 |
| Static Methods                        | 42 |
| JSON                                  | 42 |
| The String Type                       | 43 |
| Character Literals                    | 43 |
| The Nature of Strings                 | 44 |
| Converting to a String                | 44 |
| Template Literals                     | 45 |
| Interpolation                         | 46 |
| Template Literal Tag Functions        | 47 |
| Raw Strings                           | 49 |
| The Symbol Type                       | 50 |
| Basic Symbol Use                      | 50 |
| Using the Global Symbol Registry      | 51 |
| Using Symbols as Properties           | 52 |
| Well-Known Symbols                    | 53 |
| The Object Type                       | 63 |

|                                                 |    |
|-------------------------------------------------|----|
| Operators                                       | 64 |
| Unary Operators                                 | 64 |
| Increment/Decrement                             | 64 |
| Unary Plus and Minus                            | 66 |
| Bitwise Operators                               | 67 |
| Bitwise NOT                                     | 69 |
| Bitwise AND                                     | 69 |
| Bitwise OR                                      | 70 |
| Bitwise XOR                                     | 71 |
| Left Shift                                      | 71 |
| Signed Right Shift                              | 72 |
| Unsigned Right Shift                            | 73 |
| Boolean Operators                               | 73 |
| Logical NOT                                     | 73 |
| Logical AND                                     | 74 |
| Logical OR                                      | 75 |
| Multiplicative Operators                        | 76 |
| Multiply                                        | 76 |
| Divide                                          | 77 |
| Modulus                                         | 77 |
| Exponentiation Operator                         | 78 |
| Additive Operators                              | 78 |
| Add                                             | 78 |
| Subtract                                        | 79 |
| Relational Operators                            | 80 |
| Equality Operators                              | 82 |
| Equal and Not Equal                             | 82 |
| Identically Equal and Not Identically Equal     | 83 |
| Conditional Operator                            | 84 |
| Nullish Coalescing Operator                     | 84 |
| Assignment Operators                            | 85 |
| Comma Operator                                  | 86 |
| Statements                                      | 86 |
| The <i>if</i> Statement                         | 86 |
| The <i>do-while</i> Statement                   | 87 |
| The <i>while</i> Statement                      | 87 |
| The <i>for</i> Statement                        | 88 |
| The <i>for-in</i> Statement                     | 89 |
| The <i>for-of</i> Statement                     | 89 |
| Labeled Statements                              | 90 |
| The <i>break</i> and <i>continue</i> Statements | 90 |

---

|                                                                  |            |
|------------------------------------------------------------------|------------|
| The <i>with</i> Statement                                        | 91         |
| The <i>switch</i> Statement                                      | 92         |
| Functions                                                        | 94         |
| Summary                                                          | 95         |
| <b>CHAPTER 4: VARIABLES, SCOPE, AND MEMORY</b>                   | <b>97</b>  |
| <hr/>                                                            |            |
| Primitive and Reference Values                                   | 97         |
| Dynamic Properties                                               | 98         |
| Copying Values                                                   | 99         |
| Argument Passing                                                 | 100        |
| Determining Type                                                 | 101        |
| Execution Context and Scope                                      | 102        |
| Scope Chain Augmentation                                         | 105        |
| Variable Declaration                                             | 105        |
| Function Scope Declaration Using <i>var</i>                      | 105        |
| Block Scope Declaration Using <i>let</i>                         | 107        |
| Constant Declaration Using <i>const</i>                          | 108        |
| Identifier Lookup                                                | 109        |
| Garbage Collection                                               | 110        |
| Performance                                                      | 111        |
| Managing Memory                                                  | 111        |
| Performance Boosts with <i>const</i> and <i>let</i> Declarations | 112        |
| Hidden Classes and the <i>delete</i> Operation                   | 112        |
| Memory Leaks                                                     | 113        |
| Static Allocation and Object Pools                               | 114        |
| Summary                                                          | 116        |
| <b>CHAPTER 5: BASIC REFERENCE TYPES</b>                          | <b>119</b> |
| <hr/>                                                            |            |
| The Date Type                                                    | 120        |
| Inherited Methods                                                | 122        |
| Date-Formatting Methods                                          | 122        |
| Date/Time Component Methods                                      | 123        |
| The Regexp Type                                                  | 125        |
| RegExp Instance Properties                                       | 128        |
| RegExp Instance Methods                                          | 129        |
| Primitive Wrapper Types                                          | 131        |
| The Boolean Type                                                 | 133        |
| The Number Type                                                  | 133        |
| The <i>isInteger()</i> Method and Safe Integers                  | 135        |

---

|                                                   |            |
|---------------------------------------------------|------------|
| The String Type                                   | 136        |
| The JavaScript Character                          | 136        |
| The <i>normalize()</i> Method                     | 139        |
| String-Manipulation Methods                       | 140        |
| String Location Methods                           | 142        |
| String Inclusion Methods                          | 143        |
| The <i>trim()</i> Method                          | 144        |
| The <i>repeat()</i> Method                        | 144        |
| The <i>padStart()</i> and <i>padEnd()</i> Methods | 144        |
| String Iterators and Destructuring                | 145        |
| String Case Methods                               | 145        |
| String Pattern-Matching Methods                   | 146        |
| The <i>localeCompare()</i> Method                 | 149        |
| Singleton Built-In Objects                        | 150        |
| The Global Object                                 | 150        |
| URI-Encoding Methods                              | 150        |
| The <i>eval()</i> Method                          | 151        |
| Global Object Properties                          | 152        |
| The Window Object                                 | 153        |
| The Math Object                                   | 154        |
| Math Object Properties                            | 154        |
| The <i>min()</i> and <i>max()</i> Methods         | 155        |
| Rounding Methods                                  | 155        |
| The <i>random()</i> Method                        | 156        |
| Other Methods                                     | 157        |
| Summary                                           | 158        |
| <b>CHAPTER 6: ADVANCED REFERENCE TYPES</b>        | <b>159</b> |
| The Object Type                                   | 159        |
| The Array Type                                    | 162        |
| Creating Arrays                                   | 162        |
| Array Holes                                       | 165        |
| Indexing into Arrays                              | 166        |
| Detecting Arrays                                  | 168        |
| Iterator Methods                                  | 168        |
| Copy and Fill Methods                             | 169        |
| Spread Operator                                   | 171        |
| Rest Operator                                     | 171        |
| Conversion Methods                                | 172        |
| Stack Methods                                     | 173        |
| Queue Methods                                     | 174        |

---

|                                             |     |
|---------------------------------------------|-----|
| Reordering and Sorting Methods              | 175 |
| Manipulation Methods                        | 178 |
| Search and Location Methods                 | 180 |
| Iterative Methods                           | 182 |
| Reduction Methods                           | 183 |
| Flattening Methods                          | 184 |
| Typed Arrays                                | 186 |
| History                                     | 186 |
| WebGL                                       | 186 |
| Emergence of Typed Arrays                   | 187 |
| Using ArrayBuffers                          | 187 |
| DataViews                                   | 188 |
| ElementType                                 | 189 |
| Big-Endian and Little-Endian                | 190 |
| Corner Cases                                | 191 |
| Typed Arrays                                | 192 |
| Typed Array Behavior                        | 193 |
| Merging, Copying, and Changing Typed Arrays | 194 |
| Underflow and Overflow                      | 195 |
| The Map Type                                | 196 |
| Basic API                                   | 196 |
| Order and Iteration                         | 199 |
| Choosing Between Objects and Maps           | 201 |
| Keys                                        | 201 |
| Memory Profile                              | 201 |
| Insertion Performance                       | 201 |
| Lookup Performance                          | 201 |
| Delete Performance                          | 202 |
| The Set Type                                | 202 |
| Basic API                                   | 202 |
| Order and Iteration                         | 204 |
| Weak References                             | 205 |
| WeakRef                                     | 206 |
| FinalizationRegistry                        | 206 |
| The Weakmap Type                            | 207 |
| Basic API                                   | 207 |
| Weak Keys                                   | 209 |
| Non-Iterable Keys                           | 210 |
| Utility                                     | 210 |
| The Weakset Type                            | 210 |
| Basic API                                   | 211 |



|                                                                         |            |
|-------------------------------------------------------------------------|------------|
| Weak Keys                                                               | 212        |
| Non-Iterable Values                                                     | 213        |
| Utility                                                                 | 213        |
| Iteration and Spread Operators                                          | 213        |
| Summary                                                                 | 215        |
| <b>CHAPTER 7: ITERATORS AND GENERATORS</b>                              | <b>217</b> |
| Introduction To Iteration                                               | 217        |
| The Iterator Pattern                                                    | 218        |
| The Iterable Protocol                                                   | 219        |
| The Iterator Protocol                                                   | 221        |
| Custom Iterator Definition                                              | 223        |
| Early Termination of Iterators                                          | 225        |
| Generators                                                              | 227        |
| Generator Basics                                                        | 227        |
| Interrupting Execution with <i>yield</i>                                | 229        |
| Using a Generator Object as an Iterable                                 | 231        |
| Using <i>yield</i> for Input and Output                                 | 231        |
| Yielding an Iterable                                                    | 233        |
| Recursive Algorithms Using <i>yield*</i>                                | 235        |
| Using a Generator as the Default Iterator                               | 238        |
| Early Termination of Generators                                         | 238        |
| The <i>return()</i> Method                                              | 239        |
| The <i>throw()</i> Method                                               | 240        |
| Asynchronous Iteration                                                  | 241        |
| Creating and Using an Async Iterator                                    | 241        |
| Understanding the Async Iterator Queue                                  | 243        |
| Async Iterator <i>reject()</i> Handling                                 | 245        |
| Manual Async Iteration Using <i>next()</i>                              | 246        |
| Top-Level Async Loops                                                   | 246        |
| Implementing Observables                                                | 247        |
| Summary                                                                 | 249        |
| <b>CHAPTER 8: OBJECTS, CLASSES, AND<br/>OBJECT-ORIENTED PROGRAMMING</b> | <b>251</b> |
| Understanding Objects                                                   | 251        |
| Types of Properties                                                     | 252        |
| Data Properties                                                         | 252        |
| Accessor Properties                                                     | 254        |
| Accessing Object Properties                                             | 255        |
| Chaining Properties                                                     | 256        |

## CONTENTS

---

|                                       |     |
|---------------------------------------|-----|
| Object Static Methods                 | 257 |
| Controlling Object Mutability         | 258 |
| Freezing Objects                      | 258 |
| Sealing Objects                       | 259 |
| Controlling Extensibility             | 260 |
| Defining Multiple Properties          | 260 |
| Reading Property Attributes           | 261 |
| Merging Objects                       | 263 |
| Object Identity and Equality          | 265 |
| Enhanced Object Syntax                | 266 |
| Property Value Shorthand              | 266 |
| Computed Property Keys                | 267 |
| Concise Method Syntax                 | 268 |
| Object Destructuring                  | 269 |
| Nested Destructuring                  | 271 |
| Partial Destructuring Completion      | 273 |
| Parameter Context Matching            | 273 |
| Rest Operator                         | 274 |
| Spread Operator                       | 274 |
| Object Creation                       | 275 |
| Overview                              | 275 |
| The Function Constructor Pattern      | 276 |
| Constructors as Functions             | 278 |
| Problems with Constructors            | 279 |
| The Prototype Pattern                 | 280 |
| How Prototypes Work                   | 281 |
| Understanding the Prototype Hierarchy | 284 |
| Prototypes and the <i>in</i> Operator | 287 |
| Property Enumeration Order            | 290 |
| Object Iteration                      | 290 |
| Dynamic Nature of Prototypes          | 292 |
| Native Object Prototypes              | 293 |
| Problems with Prototypes              | 294 |
| Prototype Inheritance                 | 295 |
| Prototype Chaining                    | 295 |
| Default Prototypes                    | 296 |
| Prototype and Instance Relationships  | 297 |
| Working with Methods                  | 298 |
| Problems with Prototype Chaining      | 299 |
| Classes                               | 299 |
| Class Definition Basics               | 300 |

---

|                                                                 |            |
|-----------------------------------------------------------------|------------|
| Class Composition                                               | 300        |
| The Class Constructor                                           | 301        |
| Instantiation                                                   | 302        |
| Understanding Classes as Special Functions                      | 304        |
| Instance, Prototype, and Class Members                          | 305        |
| Instance Members                                                | 305        |
| Class Field Declarations                                        | 306        |
| Prototype Methods and Accessors                                 | 307        |
| Private Class Members                                           | 308        |
| Static Class Methods and Accessors                              | 310        |
| Static Initialization Blocks                                    | 311        |
| Iterator and Generator Methods                                  | 312        |
| Class Inheritance                                               | 313        |
| Inheritance Basics                                              | 313        |
| Constructors, <code>__proto__</code> , and <code>super()</code> | 314        |
| Abstract Base Classes                                           | 317        |
| Inheriting from Built-In Types                                  | 318        |
| Class Mixins                                                    | 319        |
| Summary                                                         | 321        |
| <b>CHAPTER 9: PROXIES AND REFLECT</b>                           | <b>323</b> |
| Proxy Fundamentals                                              | 324        |
| Creating a Passthrough Proxy                                    | 324        |
| Defining Traps                                                  | 325        |
| Trap Parameters and the Reflect API                             | 326        |
| Trap Invariants                                                 | 328        |
| Revocable Proxies                                               | 329        |
| Utility of the Reflect API                                      | 329        |
| Reflect API vs. Object API                                      | 330        |
| Status Flags                                                    | 330        |
| Supplanting Operators with First-Class Functions                | 331        |
| Safe Function Application                                       | 331        |
| Proxying a Proxy                                                | 331        |
| Proxy Considerations and Shortcomings                           | 332        |
| this Inside a Proxy                                             | 332        |
| Proxies and Internal Slots                                      | 333        |
| Proxy Traps and Reflect Methods                                 | 333        |
| get()                                                           | 334        |
| set()                                                           | 334        |
| has()                                                           | 335        |
| defineProperty()                                                | 336        |

---

|                                                          |            |
|----------------------------------------------------------|------------|
| getOwnPropertyDescriptor()                               | 337        |
| deleteProperty()                                         | 338        |
| ownKeys()                                                | 339        |
| getPrototypeOf()                                         | 339        |
| setPrototypeOf()                                         | 340        |
| isExtensible()                                           | 341        |
| preventExtensions()                                      | 341        |
| apply()                                                  | 342        |
| construct()                                              | 343        |
| Proxy Patterns                                           | 343        |
| Tracking Property Access                                 | 344        |
| Hidden Properties                                        | 344        |
| Property Validation                                      | 345        |
| Function and Constructor Parameter Validation            | 345        |
| Data Binding and Observables                             | 346        |
| Summary                                                  | 347        |
| <b>CHAPTER 10: FUNCTIONS</b>                             | <b>349</b> |
| Arrow Functions                                          | 350        |
| Function Names                                           | 351        |
| Understanding Arguments                                  | 353        |
| Arguments in Arrow Functions                             | 355        |
| No Overloading                                           | 355        |
| Default Parameter Values                                 | 356        |
| Default Parameter Scope and Temporal Dead Zone           | 358        |
| Spread Arguments and Rest Parameters                     | 359        |
| Spread Arguments                                         | 359        |
| Rest Parameter                                           | 360        |
| Function Declarations Versus Function Expressions        | 361        |
| Functions as Values                                      | 363        |
| Function Internals                                       | 364        |
| arguments                                                | 364        |
| this                                                     | 365        |
| caller                                                   | 367        |
| new.target                                               | 368        |
| Function Properties and Methods                          | 368        |
| Using <i>apply()</i> , <i>call()</i> , and <i>bind()</i> | 369        |
| Serializing Functions                                    | 371        |
| Recursion                                                | 371        |
| Tail Call Optimization                                   | 372        |

---

|                                                              |            |
|--------------------------------------------------------------|------------|
| Tail Call Optimization Requirements                          | 373        |
| Coding for Tail Call Optimization                            | 375        |
| Closures                                                     | 376        |
| The <i>this</i> Object                                       | 379        |
| Memory Leaks                                                 | 380        |
| Immediately Invoked Function Expressions                     | 381        |
| Summary                                                      | 382        |
| <b>CHAPTER 11: PROMISES AND ASYNC/AWAIT</b>                  | <b>383</b> |
| <hr/>                                                        |            |
| Introduction to Asynchronous Programming                     | 384        |
| Synchronous vs. Asynchronous JavaScript                      | 384        |
| Legacy Asynchronous Programming Patterns                     | 385        |
| Returning Asynchronous Values                                | 386        |
| Handling Failure                                             | 386        |
| Nesting Asynchronous Callbacks                               | 387        |
| Promises                                                     | 387        |
| The Promises/A+ Specification                                | 387        |
| Promise Basics                                               | 388        |
| The Promise State Machine                                    | 388        |
| Fulfilled Values, Rejection Reasons, and Utility of Promises | 389        |
| Controlling Promise State with the Executor                  | 389        |
| Promise Casting with <i>Promise.resolve()</i>                | 390        |
| Promise Rejection with <i>Promise.reject()</i>               | 391        |
| Synchronous/Asynchronous Execution Duality                   | 392        |
| Promise Instance Methods                                     | 392        |
| Implementing the thenable Interface                          | 393        |
| Using <i>Promise.prototype.then()</i>                        | 393        |
| Using <i>Promise.prototype.catch()</i>                       | 396        |
| Using <i>Promise.prototype.finally()</i>                     | 397        |
| Non-Reentrant Promise Methods                                | 398        |
| Sibling Handler Order of Execution                           | 400        |
| Fulfilled Value and Rejected Reason Passing                  | 400        |
| Rejecting Promises and Rejection Error Handling              | 401        |
| Promise Chaining and Composition                             | 403        |
| Promise Chaining                                             | 403        |
| Using <i>Promise.all()</i>                                   | 405        |
| Using <i>Promise.allSettled()</i>                            | 407        |
| Using <i>Promise.race()</i>                                  | 408        |
| Using <i>Promise.any()</i>                                   | 409        |
| Serial Promise Composition                                   | 410        |
| Promises and the Microtask Queue                             | 412        |

## CONTENTS

|                                             |            |
|---------------------------------------------|------------|
| Avoiding Unhandled Rejections               | 412        |
| Promise Extensions                          | 413        |
| Promise Canceling                           | 414        |
| Promise Progress Notifications              | 415        |
| Async Functions                             | 417        |
| Async Function Basics                       | 418        |
| The <i>async</i> keyword                    | 418        |
| The <i>await</i> keyword                    | 420        |
| Restrictions on <i>await</i>                | 422        |
| Strategies for Async Functions              | 426        |
| Implementing <i>Sleep()</i>                 | 426        |
| Maximizing Parallelization                  | 426        |
| Serial Promise Execution                    | 429        |
| Stack Traces and Memory Management          | 430        |
| Rejection-Safe Parallelization              | 431        |
| Summary                                     | 432        |
| <b>CHAPTER 12: THE BROWSER OBJECT MODEL</b> | <b>435</b> |
| The Window Object                           | 436        |
| The Global Scope                            | 436        |
| The <i>globalThis</i> property              | 437        |
| Window Relationships                        | 437        |
| Window Position and Pixel Ratio             | 438        |
| Pixel Ratios                                | 438        |
| Window Size                                 | 438        |
| Window Viewport Position                    | 439        |
| Navigating and Opening Windows              | 440        |
| Popping Up Windows                          | 440        |
| Security Restrictions                       | 443        |
| Pop-Up Blockers                             | 443        |
| Intervals and Timeouts                      | 444        |
| System Dialogs                              | 446        |
| The Location Object                         | 448        |
| Manipulating the Location                   | 449        |
| The Navigator Object                        | 451        |
| Registering Handlers                        | 453        |
| The Screen Object                           | 454        |
| The History Object                          | 454        |
| Navigation                                  | 455        |
| History State Management                    | 456        |
| Summary                                     | 457        |

---

|                                                     |            |
|-----------------------------------------------------|------------|
| <b>CHAPTER 13: THE DOCUMENT OBJECT MODEL</b>        | <b>459</b> |
| Hierarchy of Nodes                                  | 460        |
| The Node Type                                       | 461        |
| The <i>nodeName</i> and <i>nodeValue</i> Properties | 462        |
| Node Relationships                                  | 462        |
| Manipulating Nodes                                  | 464        |
| Other Methods                                       | 465        |
| The Document Type                                   | 466        |
| Document Children                                   | 466        |
| Document Information                                | 467        |
| Locating Elements                                   | 469        |
| Special Collections                                 | 471        |
| Document Writing                                    | 471        |
| The Element Type                                    | 473        |
| HTML Elements                                       | 474        |
| Getting Attributes                                  | 474        |
| Setting Attributes                                  | 476        |
| The <i>attributes</i> Property                      | 476        |
| Creating Elements                                   | 478        |
| Element Children                                    | 478        |
| The Text Type                                       | 479        |
| Creating Text Nodes                                 | 480        |
| Normalizing Text Nodes                              | 481        |
| Splitting Text Nodes                                | 482        |
| The Comment Type                                    | 482        |
| The CDATASection Type                               | 483        |
| The DocumentType Type                               | 484        |
| The DocumentFragment Type                           | 484        |
| The Attr Type                                       | 485        |
| Working with The Dom                                | 486        |
| Dynamic Scripts                                     | 487        |
| Dynamic Styles                                      | 488        |
| Using NodeLists                                     | 488        |
| Selectors API                                       | 490        |
| The <i>querySelector()</i> Method                   | 490        |
| The <i>querySelectorAll()</i> Method                | 491        |
| The <i>matches()</i> Method                         | 491        |
| Element Traversal                                   | 492        |
| Html5                                               | 493        |
| Class-Related Additions                             | 493        |

---

|                                                                         |            |
|-------------------------------------------------------------------------|------------|
| The <i>getElementsByClassName()</i> Method                              | 493        |
| The <i>classList</i> Property                                           | 494        |
| Focus Management                                                        | 495        |
| Changes to HTMLDocument                                                 | 496        |
| The <i>readyState</i> Property                                          | 496        |
| Compatibility Mode                                                      | 496        |
| The <i>head</i> Property                                                | 497        |
| Character Set Properties                                                | 497        |
| Custom Data Attributes                                                  | 497        |
| Markup Insertion                                                        | 498        |
| The <i>innerHTML</i> Property                                           | 498        |
| The <i>outerHTML</i> Property                                           | 499        |
| The <i>insertAdjacentHTML()</i> and <i>insertAdjacentText()</i> Methods | 500        |
| Memory and Performance Issues                                           | 500        |
| Cross-Site Scripting Considerations                                     | 501        |
| The <i>scrollIntoView()</i> Method                                      | 501        |
| The <i>children</i> Property                                            | 502        |
| The <i>contains()</i> Method                                            | 502        |
| Markup Insertion                                                        | 503        |
| The <i>innerText</i> Property                                           | 503        |
| The <i>outerText</i> Property                                           | 504        |
| Summary                                                                 | 504        |
| <b>CHAPTER 14: DOM EXTENSIONS</b>                                       | <b>507</b> |
| <b>Styles</b>                                                           | <b>508</b> |
| Accessing Element Styles                                                | 508        |
| DOM Style Properties and Methods                                        | 509        |
| Computed Styles                                                         | 511        |
| Working with Style Sheets                                               | 512        |
| CSS Rules                                                               | 513        |
| Creating Rules                                                          | 514        |
| Deleting Rules                                                          | 515        |
| Element Dimensions                                                      | 515        |
| Offset Dimensions                                                       | 515        |
| Client Dimensions                                                       | 517        |
| Scroll Dimensions                                                       | 518        |
| Determining Element Dimensions                                          | 519        |
| Traversals                                                              | 519        |
| NodeIterator                                                            | 521        |
| TreeWalker                                                              | 524        |
| Ranges                                                                  | 525        |



---

|                                                                         |            |
|-------------------------------------------------------------------------|------------|
| The <i>getElementsByClassName()</i> Method                              | 493        |
| The <i>classList</i> Property                                           | 494        |
| Focus Management                                                        | 495        |
| Changes to <i>HTMLDocument</i>                                          | 496        |
| The <i>readyState</i> Property                                          | 496        |
| Compatibility Mode                                                      | 496        |
| The <i>head</i> Property                                                | 497        |
| Character Set Properties                                                | 497        |
| Custom Data Attributes                                                  | 497        |
| Markup Insertion                                                        | 498        |
| The <i>innerHTML</i> Property                                           | 498        |
| The <i>outerHTML</i> Property                                           | 499        |
| The <i>insertAdjacentHTML()</i> and <i>insertAdjacentText()</i> Methods | 500        |
| Memory and Performance Issues                                           | 500        |
| Cross-Site Scripting Considerations                                     | 501        |
| The <i>scrollIntoView()</i> Method                                      | 501        |
| The <i>children</i> Property                                            | 502        |
| The <i>contains()</i> Method                                            | 502        |
| Markup Insertion                                                        | 503        |
| The <i>innerText</i> Property                                           | 503        |
| The <i>outerText</i> Property                                           | 504        |
| Summary                                                                 | 504        |
| <b>CHAPTER 14: DOM EXTENSIONS</b>                                       | <b>507</b> |
| <b>Styles</b>                                                           | <b>508</b> |
| Accessing Element Styles                                                | 508        |
| DOM Style Properties and Methods                                        | 509        |
| Computed Styles                                                         | 511        |
| Working with Style Sheets                                               | 512        |
| CSS Rules                                                               | 513        |
| Creating Rules                                                          | 514        |
| Deleting Rules                                                          | 515        |
| Element Dimensions                                                      | 515        |
| Offset Dimensions                                                       | 515        |
| Client Dimensions                                                       | 517        |
| Scroll Dimensions                                                       | 518        |
| Determining Element Dimensions                                          | 519        |
| <b>Traversals</b>                                                       | <b>519</b> |
| NodeIterator                                                            | 521        |
| TreeWalker                                                              | 524        |
| <b>Ranges</b>                                                           | <b>525</b> |

|                                                          |            |
|----------------------------------------------------------|------------|
| Ranges in the DOM                                        | 525        |
| Simple Selection in DOM Ranges                           | 526        |
| Complex Selection in DOM Ranges                          | 527        |
| Interacting with DOM Range Content                       | 529        |
| Inserting DOM Range Content                              | 531        |
| Collapsing a DOM Range                                   | 532        |
| Comparing DOM Ranges                                     | 533        |
| Cloning DOM Ranges                                       | 534        |
| Cleanup                                                  | 534        |
| Observer APIs                                            | 534        |
| Observer API Methods                                     | 535        |
| Initialization                                           | 535        |
| Callback Execution                                       | 535        |
| The <i>observe()</i> method                              | 535        |
| The <i>unobserve()</i> and <i>disconnect()</i> methods   | 536        |
| Async Callbacks and the Record Queue                     | 536        |
| The <i>takeRecords()</i> method                          | 537        |
| Observer References                                      | 537        |
| Resize Observers                                         | 537        |
| Box Model Configuration                                  | 539        |
| Callback Scheduling and Execution                        | 539        |
| ResizeObserverEntry                                      | 539        |
| Intersection Observers                                   | 541        |
| Intersection Configuration                               | 541        |
| Working with Callbacks and IntersectionObserverEntry     | 543        |
| Mutation Observers                                       | 545        |
| Working with Callbacks and MutationRecords               | 546        |
| Controlling the Observer Scope with MutationObserverInit | 548        |
| Observing Attribute Mutations                            | 549        |
| Observing Character Data Mutations                       | 550        |
| Observing Child Mutations                                | 551        |
| Observing Subtree Mutations                              | 553        |
| Observer Performance                                     | 554        |
| Summary                                                  | 555        |
| <b>CHAPTER 15: EVENTS</b>                                | <b>557</b> |
| Event Flow                                               | 558        |
| Event Bubbling                                           | 558        |
| DOM Event Flow                                           | 559        |
| Event Handlers                                           | 560        |
| HTML Event Handlers                                      | 560        |

|                                                                |            |
|----------------------------------------------------------------|------------|
| Gesture Events                                                 | 596        |
| Event Reference                                                | 598        |
| Memory and Performance                                         | 598        |
| Event Delegation                                               | 598        |
| Removing Event Handlers                                        | 600        |
| Simulating Events                                              | 601        |
| DOM Event Simulation                                           | 601        |
| Simulating Mouse Events                                        | 602        |
| Simulating Keyboard Events                                     | 603        |
| Simulating Other Events                                        | 605        |
| Custom DOM Events                                              | 606        |
| Summary                                                        | 606        |
| <b>CHAPTER 16: ANIMATION AND GRAPHICS WITH CANVAS</b>          | <b>609</b> |
| Using <code>requestAnimationFrame</code>                       | 610        |
| Early Animation Loops                                          | 610        |
| Problems with Intervals                                        | 611        |
| <code>requestAnimationFrame</code>                             | 611        |
| <code>cancelAnimationFrame</code>                              | 612        |
| Performance Throttling with <code>requestAnimationFrame</code> | 612        |
| Basic Canvas Usage                                             | 613        |
| The 2D Context                                                 | 614        |
| Fills and Strokes                                              | 614        |
| Drawing Rectangles                                             | 615        |
| Drawing Paths                                                  | 616        |
| Drawing Text                                                   | 618        |
| Transformations                                                | 620        |
| Drawing Images                                                 | 622        |
| Shadows                                                        | 623        |
| Gradients                                                      | 624        |
| Patterns                                                       | 626        |
| Working with Image Data                                        | 626        |
| Compositing                                                    | 628        |
| WebGL                                                          | 629        |
| The WebGL Context                                              | 630        |
| WebGL Basics                                                   | 630        |
| Constants                                                      | 631        |
| Method Naming                                                  | 631        |
| Getting Ready to Draw                                          | 631        |
| Viewports and Coordinates                                      | 632        |
| Buffers                                                        | 633        |

---

|                                    |            |
|------------------------------------|------------|
| Errors                             | 633        |
| Shaders                            | 634        |
| Drawing                            | 638        |
| Textures                           | 640        |
| Reading Pixels                     | 641        |
| WebGL1 versus WebGL2               | 641        |
| Summary                            | 642        |
| <b>CHAPTER 17: SCRIPTING FORMS</b> | <b>645</b> |
| <hr/>                              |            |
| Form Basics                        | 645        |
| Submitting Forms                   | 646        |
| Resetting Forms                    | 647        |
| Form Fields                        | 648        |
| Common Form-Field Properties       | 649        |
| Common Form-Field Methods          | 651        |
| Common Form-Field Events           | 652        |
| Scripting Text Boxes               | 653        |
| Text Selection                     | 654        |
| The <i>select</i> Event            | 654        |
| Retrieving Selected Text           | 655        |
| Partial Text Selection             | 655        |
| Input Filtering                    | 655        |
| Blocking Characters                | 655        |
| Dealing with the Clipboard         | 657        |
| HTML5 Constraint Validation API    | 658        |
| Required Fields                    | 658        |
| Alternate Input Types              | 658        |
| Numeric Ranges                     | 659        |
| Input Patterns                     | 660        |
| Checking Validity                  | 660        |
| Disabling Validation               | 661        |
| Scripting Select Boxes             | 662        |
| Options Selection                  | 663        |
| Adding Options                     | 665        |
| Removing Options                   | 665        |
| Moving and Reordering Options      | 666        |
| Rich Text Editing                  | 666        |
| Using <i>contenteditable</i>       | 667        |
| Interacting with Rich Text         | 668        |
| Rich Text Selections               | 671        |
| Rich Text in Forms                 | 672        |
| Summary                            | 673        |

|                                                                      |            |
|----------------------------------------------------------------------|------------|
| <b>CHAPTER 18: JAVASCRIPT APIs</b>                                   | <b>675</b> |
| <b>Atoms and SharedArraybuffer</b>                                   | <b>676</b> |
| SharedArrayBuffer                                                    | 676        |
| Atoms Basics                                                         | 677        |
| Atomic Arithmetic and Bitwise Methods                                | 678        |
| Atomic Reads and Writes                                              | 680        |
| Atomic Exchanges                                                     | 680        |
| Atoms Futex Operations and Locks                                     | 681        |
| <b>Clipboard API</b>                                                 | <b>684</b> |
| Permissions                                                          | 684        |
| Text Read and Write                                                  | 685        |
| Clipboard Events                                                     | 686        |
| Working with Non-Text Data                                           | 686        |
| <b>Cross-Context Messaging</b>                                       | <b>687</b> |
| <b>Encoding API</b>                                                  | <b>689</b> |
| Encoding Text                                                        | 689        |
| Bulk Encoding                                                        | 689        |
| Stream Encoding                                                      | 690        |
| Decoding Text                                                        | 691        |
| Bulk Decoding                                                        | 691        |
| Stream Decoding                                                      | 692        |
| <b>Blob And File APIs</b>                                            | <b>694</b> |
| The File Type                                                        | 694        |
| The FileReader Type                                                  | 695        |
| The FileReaderSync Type                                              | 697        |
| Blobs and Partial Reads                                              | 697        |
| Object URLs and Blobs                                                | 698        |
| Drag-and-Drop File Reading                                           | 699        |
| <b>Fullscreen API</b>                                                | <b>700</b> |
| <b>Geolocation API</b>                                               | <b>700</b> |
| <b>Device APIs</b>                                                   | <b>703</b> |
| Browser and Operating System Identification                          | 703        |
| The <i>navigator.oscpu</i> Property                                  | 704        |
| The <i>navigator.vendor</i> Property                                 | 704        |
| The <i>navigator.platform</i> Property                               | 704        |
| The <i>screen.colorDepth</i> and <i>screen.pixelDepth</i> Properties | 705        |
| The <i>screen.orientation</i> Property                               | 705        |
| Connection State and the NetworkInformation API                      | 705        |
| Battery Status API                                                   | 707        |
| Hardware                                                             | 708        |

## CONTENTS

|                                           |     |
|-------------------------------------------|-----|
| Processor Cores                           | 709 |
| Device Memory                             | 709 |
| Maximum Touch Points                      | 709 |
| Media Elements                            | 709 |
| Properties                                | 710 |
| Events                                    | 712 |
| Custom Media Players                      | 713 |
| Codec Support Detection                   | 714 |
| The Audio Type                            | 714 |
| Notifications API                         | 715 |
| Notification Permissions                  | 715 |
| Showing and Hiding Notification           | 716 |
| Notification Lifecycle Callbacks          | 716 |
| Page Visibility API                       | 716 |
| Streams API                               | 717 |
| Introduction to Streams                   | 718 |
| Chunks, Internal Queues, and Backpressure | 718 |
| Readable Streams                          | 719 |
| Using the ReadableStreamDefaultController | 719 |
| Using the ReadableStreamDefaultReader     | 720 |
| Writable Streams                          | 721 |
| Creating a WritableStream                 | 721 |
| Using a WritableStreamDefaultWriter       | 722 |
| Transform Streams                         | 723 |
| Piping Streams                            | 724 |
| URL APIs                                  | 726 |
| The URL Object                            | 726 |
| The URLSearchParams Object                | 727 |
| Timing APIs                               | 728 |
| High Resolution Time API                  | 729 |
| Performance Timeline API                  | 730 |
| User Timing API                           | 731 |
| Navigation Timing API                     | 732 |
| Resource Timing API                       | 732 |
| Web Components                            | 733 |
| HTML Templates                            | 733 |
| Using a DocumentFragment                  | 733 |
| Using <code>&lt;template&gt;</code> tags  | 735 |
| Template Scripts                          | 736 |
| Shadow DOM                                | 736 |
| Introduction to Shadow DOM                | 736 |

|                                                     |            |
|-----------------------------------------------------|------------|
| Creating a Shadow DOM                               | 737        |
| Using a Shadow DOM                                  | 739        |
| Composition and Shadow DOM Slots                    | 741        |
| Event Retargeting                                   | 743        |
| Custom Elements                                     | 744        |
| Defining a Custom Element                           | 744        |
| Adding Web Component Content                        | 746        |
| Using Custom Element Lifecycle Hooks                | 747        |
| Reflecting Custom Element Attributes                | 748        |
| Upgrading Custom Elements                           | 749        |
| The Web Cryptography API                            | 750        |
| Random Number Generation                            | 750        |
| Using the SubtleCrypto Object                       | 752        |
| Generating Cryptographic Digests                    | 752        |
| CryptoKeys and Algorithms                           | 754        |
| Generating CryptoKeys                               | 756        |
| Exporting and Importing Keys                        | 758        |
| Deriving Keys from Master Keys                      | 759        |
| Signing and Verifying Messages with Asymmetric Keys | 760        |
| Encrypting and Decrypting with Symmetric Keys       | 762        |
| Wrapping and Unwrapping a Key                       | 762        |
| Summary                                             | 764        |
| <b>CHAPTER 19: ERROR HANDLING AND DEBUGGING</b>     | <b>765</b> |
| Browser Error Reporting                             | 766        |
| Desktop Consoles                                    | 766        |
| Mobile Consoles                                     | 766        |
| Error Handling                                      | 766        |
| The <i>try-catch</i> Statement                      | 767        |
| The <i>finally</i> Clause                           | 768        |
| Throwing Values vs. Error Objects                   | 769        |
| Catching and Rethrowing with <i>error.cause</i>     | 769        |
| Error Types                                         | 770        |
| Throwing Errors                                     | 772        |
| When to Throw Errors                                | 773        |
| Throwing Errors versus <i>try-catch</i>             | 774        |
| The <i>error</i> Event                              | 774        |
| Error Handling Strategies                           | 775        |
| Identifying Where Errors Might Occur                | 776        |
| Common Sources of Error                             | 776        |
| Type Coercion Errors                                | 776        |

---

|                                                          |            |
|----------------------------------------------------------|------------|
| Data Type Errors                                         | 778        |
| Distinguishing Between Fatal and Nonfatal Errors         | 780        |
| Debugging Techniques                                     | 781        |
| Logging Messages to a Console                            | 781        |
| Log Levels                                               | 782        |
| Advanced Console Methods                                 | 783        |
| Understanding the Console Runtime                        | 786        |
| Using the JavaScript Debugger                            | 787        |
| Logging Messages to the Page                             | 787        |
| Shimming Console Methods                                 | 788        |
| Throwing Errors                                          | 788        |
| Summary                                                  | 789        |
| <b>CHAPTER 20: JSON</b>                                  | <b>791</b> |
| Syntax                                                   | 792        |
| Simple Values                                            | 792        |
| Objects                                                  | 792        |
| Arrays                                                   | 793        |
| Parsing and Serialization                                | 794        |
| The JSON Object                                          | 794        |
| Serialization Options                                    | 795        |
| Filtering Results                                        | 796        |
| String Indentation                                       | 797        |
| The <code>toJSON()</code> Method                         | 798        |
| Parsing Options                                          | 799        |
| Summary                                                  | 799        |
| <b>CHAPTER 21: NETWORK REQUESTS AND REMOTE RESOURCES</b> | <b>801</b> |
| The Fetch API                                            | 802        |
| Basic API Utilization                                    | 803        |
| Dispatching a Request                                    | 803        |
| Reading a Response                                       | 803        |
| Handling Status Codes and Request Failures               | 803        |
| Custom Fetch Options                                     | 805        |
| Common Fetch Patterns                                    | 810        |
| Sending JSON Data                                        | 810        |
| Sending Parameters in a Request Body                     | 810        |
| Sending Files                                            | 811        |
| Loading Files as Blobs                                   | 811        |
| Sending a Cross-Origin Request                           | 812        |



---

|                                           |            |
|-------------------------------------------|------------|
| Aborting a Request                        | 812        |
| The Headers Object                        | 812        |
| Unique Features of the Headers Object     | 813        |
| Header Guards                             | 813        |
| The Request Object                        | 814        |
| Creating Request Objects                  | 814        |
| Cloning Request Objects                   | 815        |
| Using Request Objects with <i>fetch()</i> | 816        |
| The Response Object                       | 817        |
| Creating Response Objects                 | 817        |
| Reading Response Status Information       | 819        |
| Cloning Response Objects                  | 821        |
| Requests, Responses, and the Body Mixin   | 822        |
| Body.text()                               | 823        |
| Body.json()                               | 823        |
| Body.formData()                           | 823        |
| Body.arrayBuffer()                        | 824        |
| Body.blob()                               | 824        |
| Single-Use Streams                        | 825        |
| Using a ReadableStream Body               | 826        |
| Cross-Origin Resource Sharing             | 830        |
| Preflighted Requests                      | 831        |
| Credentialed Requests                     | 832        |
| The Beacon API                            | 832        |
| Web Sockets                               | 833        |
| The API                                   | 834        |
| Sending/Receiving Data                    | 834        |
| Other Events                              | 835        |
| The Eventsource API                       | 836        |
| Summary                                   | 836        |
| <b>CHAPTER 22: CLIENT-SIDE STORAGE</b>    | <b>839</b> |
| Cookies                                   | 839        |
| Restrictions                              | 840        |
| Cookie Parts                              | 841        |
| Cookies in JavaScript                     | 842        |
| Cookie Considerations                     | 843        |
| Web Storage                               | 843        |
| The Storage Type                          | 844        |
| The <i>sessionStorage</i> Object          | 845        |
| The <i>localStorage</i> Object            | 846        |

|                                     |            |
|-------------------------------------|------------|
| The <i>storage</i> Event            | 846        |
| Limits and Restrictions             | 847        |
| Indexeddb                           | 847        |
| Databases                           | 847        |
| Object Stores                       | 848        |
| Transactions                        | 849        |
| Insertion                           | 850        |
| Querying with Cursors               | 851        |
| Key Ranges                          | 853        |
| Setting Cursor Direction            | 854        |
| Indexes                             | 855        |
| Concurrency Issues                  | 857        |
| Limits and Restrictions             | 858        |
| Wrapper Libraries                   | 858        |
| Summary                             | 859        |
| <b>CHAPTER 23: MODULES</b>          | <b>861</b> |
| Understanding The Module Pattern    | 862        |
| Module Identifiers                  | 862        |
| Module Dependencies                 | 862        |
| Module Loading                      | 862        |
| Entry Points                        | 863        |
| Asynchronous Dependencies           | 864        |
| Programmatic Dependencies           | 864        |
| Static Analysis                     | 865        |
| Circular Dependencies               | 865        |
| Working with Pre-Es6 Module Loaders | 867        |
| CommonJS                            | 868        |
| Asynchronous Module Definition      | 870        |
| Universal Module Definition         | 871        |
| Module Loader Deprecation           | 872        |
| Working with ECMAScript Modules     | 872        |
| Module Tagging and Definition       | 873        |
| Module Loading                      | 874        |
| Module Behavior                     | 874        |
| Module Exports                      | 875        |
| Module Imports                      | 878        |
| Import Metadata                     | 880        |
| Dynamic Imports                     | 881        |
| Module Side Effects                 | 881        |
| Module Passthrough Exports          | 882        |

|                                              |            |
|----------------------------------------------|------------|
| Import Maps                                  | 883        |
| Worker Modules                               | 884        |
| Backwards Compatibility                      | 884        |
| Summary                                      | 885        |
| <b>CHAPTER 24: WORKERS</b>                   | <b>887</b> |
| Introduction to Workers                      | 888        |
| Comparing Workers and Threads                | 888        |
| Types of Workers                             | 889        |
| Dedicated Web Worker                         | 889        |
| Shared Web Worker                            | 889        |
| Service Worker                               | 889        |
| The WorkerGlobalScope                        | 890        |
| WorkerGlobalScope Properties and Methods     | 890        |
| Subclasses of WorkerGlobalScope              | 891        |
| Dedicated Workers                            | 891        |
| Dedicated Worker Basics                      | 891        |
| Creating a Dedicated Worker                  | 891        |
| Worker Security Restrictions                 | 892        |
| Using the Worker Object                      | 893        |
| The DedicatedWorkerGlobalScope               | 894        |
| Dedicated Workers and Implicit MessagePorts  | 895        |
| Understanding the Dedicated Worker Lifecycle | 895        |
| Configuring Worker Options                   | 897        |
| Creating a Worker from Inline JavaScript     | 898        |
| Dynamic Script Execution Inside a Worker     | 899        |
| Delegating Tasks to Subworkers               | 900        |
| Handling Worker Errors                       | 900        |
| Communicating with a Dedicated Worker        | 901        |
| Communicating with <i>postMessage()</i>      | 901        |
| Communicating with MessageChannel            | 902        |
| Communicating with BroadcastChannel          | 905        |
| Worker Data Transfer                         | 905        |
| Structured Clone Algorithm                   | 906        |
| Transferable Objects                         | 907        |
| SharedArrayBuffer                            | 909        |
| Worker Pools                                 | 912        |
| Shared Workers                               | 915        |
| Shared Worker Basics                         | 916        |
| Creating a Shared Worker                     | 916        |
| SharedWorker Identity and Single Occupancy   | 916        |

|                                                                 |     |
|-----------------------------------------------------------------|-----|
| Using the SharedWorker Object                                   | 918 |
| The SharedWorkerGlobalScope                                     | 918 |
| Understanding the Shared Worker Lifecycle                       | 919 |
| Connecting to a Shared Worker                                   | 920 |
| Service Workers                                                 | 921 |
| Service Worker Use Cases                                        | 922 |
| Service Worker Basics                                           | 923 |
| The ServiceWorkerContainer                                      | 923 |
| Creating a Service Worker                                       | 923 |
| Using the ServiceWorkerContainer Object                         | 924 |
| Using the ServiceWorkerRegistration Object                      | 925 |
| Using the ServiceWorker Object                                  | 927 |
| Service Worker Security Restrictions                            | 927 |
| The ServiceWorkerGlobalScope                                    | 928 |
| Service Worker Scope Limitations                                | 930 |
| The Service Worker Cache                                        | 931 |
| The CacheStorage Object                                         | 932 |
| The Cache Object                                                | 934 |
| Maximum Cache Storage                                           | 937 |
| Service Worker Clients                                          | 937 |
| Service Workers and Consistency                                 | 938 |
| Understanding the Service Worker Lifecycle                      | 939 |
| The Parsed State                                                | 939 |
| The Installing State                                            | 940 |
| The Installed State                                             | 941 |
| The Activating State                                            | 941 |
| The Activated State                                             | 943 |
| The Redundant State                                             | 943 |
| Updating a Service Worker                                       | 943 |
| Inversion of Control and Service Worker Persistence             | 944 |
| Managing Service Worker File Caching with <i>updateViaCache</i> | 944 |
| Forced Service Worker Operation                                 | 945 |
| Service Worker Messaging                                        | 946 |
| Intercepting a fetch Event                                      | 948 |
| Return from Network                                             | 948 |
| Return from Cache                                               | 949 |
| Return from Network with Cache Fallback                         | 949 |
| Return from Cache with Network Fallback                         | 949 |
| Generic Fallback                                                | 949 |
| Push Notifications                                              | 950 |
| Displaying Notifications                                        | 950 |

---

|                                           |            |
|-------------------------------------------|------------|
| Handling Notification Events              | 951        |
| Subscribing to Push Events                | 951        |
| Handling Push Events                      | 952        |
| Summary                                   | 953        |
| <b>CHAPTER 25: BEST PRACTICES</b>         | <b>955</b> |
| <hr/>                                     |            |
| Maintainability                           | 955        |
| What Is Maintainable Code?                | 956        |
| Code Conventions                          | 956        |
| Readability                               | 957        |
| Variable and Function Naming              | 957        |
| Variable Type Transparency                | 958        |
| Loose Coupling                            | 959        |
| Decouple HTML/JavaScript                  | 960        |
| Decouple CSS/JavaScript                   | 961        |
| Decouple Application Logic/Event Handlers | 962        |
| Programming Practices                     | 963        |
| Respect Object Ownership                  | 963        |
| Avoid Globals                             | 964        |
| Avoid Null Comparisons                    | 966        |
| Use Constants                             | 967        |
| Avoid Using <i>var</i> Keyword            | 967        |
| Performance                               | 967        |
| Don't Over-Optimize                       | 968        |
| Be Scope-Aware                            | 968        |
| Problematic Language Features             | 969        |
| Avoid the <i>with</i> Statement           | 969        |
| Avoid the <i>delete</i> Operator          | 969        |
| Avoid HTML and JS String Parsing          | 970        |
| Choose the Right Approach                 | 970        |
| Avoid Unnecessary Property Lookup         | 970        |
| Optimize Loops                            | 971        |
| Other Performance Considerations          | 973        |
| Minimize Statement Count                  | 973        |
| Multiple Variable Declarations            | 973        |
| Insert Iterative Values                   | 974        |
| Use Array and Object Literals             | 974        |
| Optimize DOM Interactions                 | 975        |
| Minimize Live Updates                     | 975        |
| Use <i>innerHTML</i>                      | 976        |

## CONTENTS

|                                                        |            |
|--------------------------------------------------------|------------|
| Use Event Delegation                                   | 977        |
| Beware of HTMLCollections                              | 977        |
| Strong Typing                                          | 978        |
| Deployment                                             | 979        |
| Build Process                                          | 979        |
| File Structure                                         | 979        |
| Task Runners                                           | 980        |
| Tree Shaking                                           | 980        |
| Module Bundlers                                        | 981        |
| Validation                                             | 981        |
| Compression                                            | 982        |
| Code Minification                                      | 982        |
| JavaScript Compilation                                 | 983        |
| JavaScript Transpilation                               | 983        |
| HTTP Compression                                       | 983        |
| Summary                                                | 984        |
| <b>APPENDIX A: ES.NEXT</b>                             | <b>987</b> |
| Array Find From Last Methods                           | 987        |
| Hashbang/Shebang Grammar                               | 988        |
| Symbols As Weakmap Keys                                | 988        |
| Change Array By Copy                                   | 988        |
| <b>APPENDIX B: STRICT MODE</b>                         | <b>991</b> |
| Opting-In                                              | 991        |
| Classes And Modules                                    | 992        |
| Variables                                              | 992        |
| Objects                                                | 992        |
| Functions                                              | 993        |
| Function Parameters                                    | 995        |
| Using eval()                                           | 995        |
| eval AND ARGUMENTS                                     | 996        |
| Coercion Of This                                       | 996        |
| Other Changes                                          | 997        |
| <b>APPENDIX C: JAVASCRIPT LIBRARIES AND FRAMEWORKS</b> | <b>999</b> |
| Frameworks                                             | 999        |
| React                                                  | 999        |
| Angular                                                | 1000       |
| Vue                                                    | 1000       |

|                                                         |             |
|---------------------------------------------------------|-------------|
| Alpine.js                                               | 1000        |
| Ember                                                   | 1001        |
| Meteor                                                  | 1001        |
| Backbone.js                                             | 1001        |
| Useful Libraries                                        | 1001        |
| jQuery                                                  | 1001        |
| Google Closure Library                                  | 1002        |
| Underscore.js                                           | 1002        |
| Lodash                                                  | 1002        |
| D3                                                      | 1002        |
| three.js                                                | 1002        |
| Anime.js                                                | 1003        |
| Chart.js                                                | 1003        |
| Leaflet                                                 | 1003        |
| Axios                                                   | 1003        |
| Rxjs                                                    | 1004        |
| <b>APPENDIX D: JAVASCRIPT TOOLS</b>                     | <b>1005</b> |
| Package Managers                                        | 1005        |
| npm                                                     | 1005        |
| Yarn                                                    | 1006        |
| Bower                                                   | 1006        |
| Module Loaders                                          | 1006        |
| SystemJS                                                | 1007        |
| RequireJS                                               | 1007        |
| Module Bundlers                                         | 1007        |
| Webpack                                                 | 1007        |
| Parcel                                                  | 1007        |
| Rollup                                                  | 1008        |
| Compilation/Transpilation Tools And Static Type Systems | 1008        |
| Babel                                                   | 1008        |
| Google Closure Compiler                                 | 1008        |
| TypeScript                                              | 1008        |
| Flow                                                    | 1008        |
| High-Performance Script Tools                           | 1009        |
| WebAssembly                                             | 1009        |
| asm.js                                                  | 1009        |
| Emscripten and LLVM                                     | 1009        |

## CONTENTS

---

|                                                          |             |
|----------------------------------------------------------|-------------|
| <b>Editors</b>                                           | <b>1009</b> |
| Sublime Text                                             | 1010        |
| Atom                                                     | 1010        |
| Brackets                                                 | 1010        |
| Visual Studio Code                                       | 1010        |
| WebStorm                                                 | 1010        |
| <b>Build Tools, Automation Systems, And Task Runners</b> | <b>1010</b> |
| npm                                                      | 1011        |
| Grunt                                                    | 1011        |
| Gulp                                                     | 1011        |
| <b>Linters and Formatters</b>                            | <b>1011</b> |
| ESLint                                                   | 1011        |
| Google Closure Compiler                                  | 1012        |
| JSLint                                                   | 1012        |
| JSHint                                                   | 1012        |
| Clang Format                                             | 1012        |
| <b>Minifiers</b>                                         | <b>1012</b> |
| Uglify                                                   | 1013        |
| Google Closure Compiler                                  | 1013        |
| JSMIn                                                    | 1013        |
| <b>Unit Testing</b>                                      | <b>1013</b> |
| Mocha                                                    | 1013        |
| Jest                                                     | 1013        |
| Jasmine                                                  | 1014        |
| qUnit                                                    | 1014        |
| JsUnit                                                   | 1014        |
| <b>Documentation Generators</b>                          | <b>1014</b> |
| ESDoc                                                    | 1014        |
| documentation.js                                         | 1014        |
| Docco                                                    | 1015        |
| JsDoc Toolkit                                            | 1015        |
| <b>INDEX</b>                                             | <b>1017</b> |