

brief contents

PART 1	INTRODUCTION	1
1	■ HTML5: from documents to applications	3
PART 2	BROWSER-BASED APPS	35
2	■ Form creation: input widgets, data binding, and data validation	37
3	■ File editing and management: rich formatting, file storage, drag and drop	71
4	■ Messaging: communicating to and from scripts in HTML5	101
5	■ Mobile applications: client storage and offline execution	131
PART 3	INTERACTIVE GRAPHICS, MEDIA, AND GAMING	163
6	■ 2D Canvas: low-level, 2D graphics rendering	165
7	■ SVG: responsive in-browser graphics	199
8	■ Video and audio: playing media in the browser	237
9	■ WebGL: 3D application development	267

contents

foreword xi
preface xiii
acknowledgments xv
about this book xvii

PART 1 INTRODUCTION.1

- 1 HTML5: from documents to applications 3**
- 1.1 Exploring the markup: a whirlwind tour of HTML5 4
 - Creating the basic structure of an HTML5 document 5*
 - Using the new semantic elements 6* ▪ *Enhancing accessibility using ARIA roles 9* ▪ *Enabling support in Internet Explorer versions 6 to 8 10* ▪ *Introducing HTML5’s new form features 11* ▪ *Progress bars, meters, and collapsible content 13*
- 1.2 Beyond the markup: additional web standards 15
 - Microdata 16* ▪ *CSS3 18* ▪ *JavaScript and the DOM 19*
- 1.3 The HTML5 DOM APIs 20
 - Canvas 21* ▪ *Audio and video 21* ▪ *Drag and drop 22*
 - Cross-document messaging, server-sent events, and WebSockets 23*

- Document editing 25 ▪ Web storage 26
- Offline web applications 27
- 1.4 Additional APIs and specifications 28
 - Geolocation API 29 ▪ Indexed database (IndexedDB API) 29
 - File, File Reader, File Writer, and File System APIs 30
 - Scalable Vector Graphics 31 ▪ Web Graphics Library 33
- 1.5 Summary 33

PART 2 BROWSER-BASED APPS35

2 Form creation: input widgets, data binding, and data validation 37

- 2.1 Previewing the form and gathering prerequisites 38
 - Gathering the application prerequisites 39
- 2.2 Building a form's user interface 40
 - Defining a form's basic HTML document structure 40
 - Using the form input types email and tel and the input attributes autofocus, required, and placeholder 41 ▪ Using the form input attribute required 44 ▪ Building a calculator-style form using the input type number, the input attributes min/max and data-*, and the element <output> 45 ▪ Using the form input type month and input attribute pattern 49 ▪ Allowing users to choose whether to save or submit a form: using the input attributes formnovalidate and formaction 51
- 2.3 Calculating totals and displaying form output 53
 - Building calculation functions 53 ▪ Accessing values from HTML5 data-* attributes 55
- 2.4 Checking form input data with the Constraint Validation API 58
 - Creating custom validation tests and error messages with the setCustomValidity method and the validationMessage property 59 ▪ Detecting a failed form validation with the invalid event 60 ▪ Styling invalid elements using CSS3 pseudo-classes 61
- 2.5 Providing fallbacks for unsupported browsers 62
 - Detecting features and loading resources with Modernizr: an overview 63 ▪ Using polyfills and Modernizr to plug the gaps 64 ▪ Performing validation without the Constraint Validation API 65
- 2.6 Summary 68

3 *File editing and management: rich formatting, file storage, drag and drop* 71

- 3.1 The Super HTML5 Editor: application overview, prerequisites, and first steps 72
 - Defining the HTML document structure* 74
 - Implementing navigation and state management in JavaScript* 76
- 3.2 Rich-text editing and geolocation 78
 - Using designMode to make an HTML document editable* 79
 - Providing rich-text editing controls with execCommand* 81
 - Mapping a user's current location with the Geolocation API* 82
- 3.3 Managing files locally: the File System, Quota Management, File, and File Writer APIs 84
 - Creating an application filesystem* 86
 - *Getting a list of files from the filesystem* 87
 - *Loading, viewing, editing, and deleting files* 89
 - *Creating new files* 91
 - Saving files using the File Writer API* 94
- 3.4 Adding drag-and-drop interactivity 96
 - Dragging files into an application for import* 97
 - Dragging files out of an application for export* 98
- 3.5 Summary 99

4 *Messaging: communicating to and from scripts in HTML5* 101

- 4.1 Server-sent events (SSE) 102
 - A simple SSE chat application* 102
 - *When to use SSE* 112
- 4.2 Using WebSockets to build a real-time messaging web app 112
 - Application overview and prerequisites* 113
 - Creating a WebSocket with Node.js* 115
 - Building the planner application* 117
- 4.3 Messaging on the client side 125
 - Communicating across domains with postMessage* 126
 - Joining the applications with cross-document messaging* 127
- 4.4 Summary 129

5 **Mobile applications: client storage and offline execution** 131

- 5.1 My Tasks: application overview, prerequisites, and first steps 132
 - Defining the HTML document structure* 135
 - Controlling visibility of views using CSS* 137
 - Implementing navigation with JavaScript* 137
- 5.2 Managing data with the Web Storage API 139
 - Reading data from localStorage* 140 ■ *Saving data to localStorage* 141 ■ *Deleting data from localStorage* 141
- 5.3 Managing data using IndexedDB 143
 - Detecting database support on a browser* 144 ■ *Creating or connecting to an IndexedDB database, creating an object store and index* 145 ■ *Developing a dynamic list with HTML and JavaScript* 148 ■ *Searching an IndexedDB database* 150
 - Adding data to a database using IndexedDB or Web SQL* 152
 - Updating and deleting data from an IndexedDB database* 154
 - Dropping a database using IndexedDB* 155
- 5.4 Creating a web application that works offline: using the application cache manifest 156
 - Configuring a web server for an application cache manifest's MIME type* 157 ■ *Creating a cache manifest file* 158
 - Automating application updates* 160
- 5.5 Summary 162

PART 3 INTERACTIVE GRAPHICS, MEDIA, AND GAMING.... 163

6 **2D Canvas: low-level, 2D graphics rendering** 165

- 6.1 Canvas basics 166
 - Setting the Canvas context* 166
 - Generating a Canvas context* 169
- 6.2 Creating a Canvas game 170
 - Creating the main engine components* 171 ■ *Creating dynamic rectangles* 175 ■ *Creating arcs and circles* 178 ■ *Using paths to create complex shapes* 179
- 6.3 Breathing life into Canvas elements 181
 - Animating game elements* 181 ■ *Detecting overlap* 183
 - Creating keyboard, mouse, and touch controls* 185
 - Control input considerations* 188

- 6.4 Polishing Canvas games 189
 - Tracking score and levels* 189
 - *Adding opening and closing screens* 193
 - *Getting help from code libraries* 195
- 6.5 Summary 196

7 **SVG: responsive in-browser graphics** 199

- 7.1 How bitmap and vector graphics compare 200
- 7.2 Starting SVG Aliens with XML 202
 - Setting up SVG inside HTML* 204
 - *Programming simple shapes and text* 206
 - *Using XLink and advanced shapes* 208
- 7.3 Adding JavaScript for interactivity 212
 - Game engine essentials and using screens* 214
 - *Design patterns, dynamic object creation, and input* 216
 - *Creating and organizing complex shapes* 221
 - *Maintaining a complex SVG group* 227
 - *SVG vs. Canvas* 232
- 7.4 Summary 235

8 **Video and audio: playing media in the browser** 237

- 8.1 Playing video with HTML5 238
 - Application preview and prerequisites* 238
 - *Building the basic jukebox framework* 240
 - *Using the video element to add videos to web pages* 241
- 8.2 Controlling videos with the HTMLMediaElement interface 243
- 8.3 Specifying multiple formats with the <source> element 248
 - Discovering which video is playing with .currentSrc* 249
 - Converting between media formats* 252
- 8.4 Combining user input with video to build a telestrator 252
 - Playing video through the <canvas> element* 253
 - Creating custom video playback controls* 255
 - Manipulating video as it's playing* 257
 - Building the telestrator features* 262
- 8.5 Summary 265

9 **WebGL: 3D application development** 267

- 9.1 Building a WebGL engine 269
 - Setting up the engine's layout* 271
 - *Tools to create, alter, and delete objects* 277

9.2	Communicating with a graphics card	282
	<i>Graphics cards: a quick primer</i>	283
	<i>Creating shaders for 3D data</i>	284
	<i>Creating buffers for shape, color, and dimension</i>	287
	<i>Displaying shape data on a screen</i>	288
9.3	Putting it all together: creating Geometry Destroyer	293
	<i>Creating a game interface and control objects</i>	293
	<i>Creating 2D shapes in 3D</i>	296
	<i>Creating 3D shapes and particles</i>	300
9.4	Summary	309
<i>appendix A</i>	<i>HTML5 and related specifications</i>	310
<i>appendix B</i>	<i>HTML5 API reference</i>	317
<i>appendix C</i>	<i>Installing PHP and MySQL</i>	336
<i>appendix D</i>	<i>Computer networking primer</i>	354
<i>appendix E</i>	<i>Setting up Node.js</i>	364
<i>appendix F</i>	<i>Channel messaging</i>	373
<i>appendix G</i>	<i>Tools and libraries</i>	378
<i>appendix H</i>	<i>Encoding with FFmpeg</i>	385
<i>appendix I</i>	<i>HTML next</i>	390
<i>appendix J</i>	<i>Links and references</i>	425
	<i>index</i>	429