

Contents

Foreword	23
Preface	25

1 Understanding the Basics 29

1.1 Terminology	29
1.1.1 Client and Server	29
1.1.2 Relationship between URLs, Domains, and IP Addresses	30
1.2 Structure of Web Applications	32
1.2.1 Creating Web Pages Using HTML	32
1.2.2 Designing Web Pages with CSS	33
1.2.3 Making Web Pages Interactive with JavaScript	34
1.2.4 Making Web Pages Dynamic Using Server-Side Logic	35
1.3 Full Stack Development	36
1.3.1 What Are Software Stacks?	36
1.3.2 What Types of Stacks Exist?	37
1.3.3 What Is a Full Stack Developer?	38
1.3.4 Structure of This Book	40
1.4 Tools for Full Stack Developers	42
1.4.1 Editors	43
1.4.2 Development Environments	44
1.4.3 Browsers	46
1.5 Summary and Outlook	50
1.5.1 Key Points	50
1.5.2 Outlook	50

2 Structuring Web Pages with HTML 51

2.1 Introduction	51
2.1.1 Versions	52
2.1.2 Using Elements and Attributes	52
2.1.3 Saving Web Pages as HTML Documents	54

- 2.2 Using the Most Important Elements** 56
 - 2.2.1 Using Headings, Paragraphs, and Other Text Formatting 56
 - 2.2.2 Creating Lists 57
 - 2.2.3 Defining Links 59
 - 2.2.4 Including Images 64
 - 2.2.5 Structuring Data in Tables 66
 - 2.2.6 Defining Forms 72
 - 2.2.7 Further Information 76
- 2.3 Summary and Outlook** 77
 - 2.3.1 Key Points 77
 - 2.3.2 Recommended Reading 78
 - 2.3.3 Outlook 78

3 Designing Web Pages with CSS 79

- 3.1 Introduction** 79
 - 3.1.1 The Principle of CSS 80
 - 3.1.2 Including CSS in HTML 80
 - 3.1.3 Selectors 85
 - 3.1.4 Cascading and Specificity 88
 - 3.1.5 Inheritance 91
- 3.2 Applying Colors and Text Formatting** 91
 - 3.2.1 Defining the Text Color and Background Color 91
 - 3.2.2 Designing Texts 93
- 3.3 Lists and Tables** 103
 - 3.3.1 Designing Lists 103
 - 3.3.2 Designing Tables 107
- 3.4 Understanding the Different Layout Systems** 112
 - 3.4.1 Basic Principles of Positioning with CSS 113
 - 3.4.2 Float Layout 114
 - 3.4.3 Flexbox Layout 118
 - 3.4.4 Grid Layout 124
- 3.5 Summary and Outlook** 129
 - 3.5.1 Key Points 130
 - 3.5.2 Recommended Reading 130
 - 3.5.3 Outlook 131

4	Making Web Pages Interactive with JavaScript	133
4.1	Introduction	133
4.1.1	Including JavaScript	134
4.1.2	Displaying Dialog Boxes	136
4.1.3	Using the Developer Console	137
4.1.4	Introduction to Programming	139
4.2	Variables, Constants, Data Types, and Operators	140
4.2.1	Defining Variables	140
4.2.2	Defining Constants	141
4.2.3	Using Data Types	141
4.2.4	Using Operators	143
4.3	Using Control Structures	144
4.3.1	Using Conditional Statements and Branching	144
4.3.2	Using Loops	146
4.4	Functions and Error Handling	147
4.4.1	Defining and Calling Functions	147
4.4.2	Passing and Analyzing Function Parameters	148
4.4.3	Defining Return Values	149
4.4.4	Responding to Errors	149
4.5	Objects and Arrays	151
4.5.1	Using Objects	151
4.5.2	Using Arrays	152
4.6	Summary and Outlook	154
4.6.1	Key Points	154
4.6.2	Recommended Reading	155
4.6.3	Outlook	155
5	Using Web Protocols	157
5.1	Hypertext Transfer Protocol	157
5.1.1	Requests and Responses	158
5.1.2	Structure of HTTP Requests	160
5.1.3	Structure of HTTP Responses	161
5.1.4	Header	162
5.1.5	Methods	164
5.1.6	Status Codes	166
5.1.7	MIME Types	167

- 5.1.8 Cookies 170
 - 5.1.9 Executing HTTP from the Command Line 173
- 5.2 Bidirectional Communication 174
 - 5.2.1 Polling and Long Polling 174
 - 5.2.2 Server-Sent Events 175
 - 5.2.3 WebSockets 176
- 5.3 Summary and Outlook 178
 - 5.3.1 Key Points 178
 - 5.3.2 Recommended Reading 178
 - 5.3.3 Outlook 179

6 Using Web Formats 181

- 6.1 Data Formats 182
 - 6.1.1 CSV 182
 - 6.1.2 XML 182
 - 6.1.3 JSON 187
- 6.2 Image Formats 193
 - 6.2.1 Photographs in the JPG Format 193
 - 6.2.2 Graphics and Animations in the GIF Format 193
 - 6.2.3 Graphics in the PNG Format 194
 - 6.2.4 Vector Graphics in the SVG Format 195
 - 6.2.5 Everything Gets Better with the WebP Format 196
 - 6.2.6 Comparing Image Formats 196
 - 6.2.7 Programs for Image Processing 198
- 6.3 Video and Audio Formats 199
 - 6.3.1 Video Formats 199
 - 6.3.2 Audio Formats 202
- 6.4 Summary and Outlook 204
 - 6.4.1 Key Points 204
 - 6.4.2 Recommended Reading 204
 - 6.4.3 Outlook 205

7 Using Web APIs 207

- 7.1 Changing Web Pages Dynamically Using the DOM API 208
 - 7.1.1 The Document Object Model 208

7.1.2	The Different Types of Nodes	209
7.1.3	Selecting Elements	211
7.1.4	Modifying Elements	213
7.1.5	Creating, Adding, and Deleting Elements	214
7.1.6	Practical Example: Dynamic Creation of a Table	215
7.2	Loading Data Synchronously via Ajax and the Fetch API	218
7.2.1	Synchronous versus Asynchronous Communication	218
7.2.2	Loading Data via Ajax	220
7.2.3	Loading Data via the Fetch API	223
7.3	Other Web APIs	223
7.3.1	Overview of Web APIs	224
7.3.2	Browser Support for Web APIs	227
7.4	Summary and Outlook	227
7.4.1	Key Points	228
7.4.2	Recommended Reading	228
7.4.3	Outlook	228

8 Optimizing Websites for Accessibility 229

8.1	Introduction	229
8.1.1	Introduction to Accessibility	230
8.1.2	User Groups and Assistive Technologies	230
8.1.3	Web Content Accessibility Guidelines	232
8.2	Making Components of a Website Accessible	236
8.2.1	Structuring Web Pages Semantically	236
8.2.2	Using Headings Correctly	239
8.2.3	Making Forms Accessible	239
8.2.4	Making Tables Accessible	241
8.2.5	Making Images Accessible	246
8.2.6	Making Links Accessible	248
8.2.7	Accessible Rich Internet Applications	249
8.2.8	Miscellaneous	251
8.3	Testing Accessibility	254
8.3.1	Types of Tests	254
8.3.2	Tools for Testing	255
8.4	Summary and Outlook	258
8.4.1	Key Points	258
8.4.2	Recommended Reading	259
8.4.3	Outlook	259

9 Simplifying CSS with CSS Preprocessors 261

9.1 Introduction	261
9.1.1 How CSS Preprocessors Work	262
9.1.2 Features of CSS Preprocessors	262
9.1.3 Sass, Less, and Stylus	264
9.2 Using Sass	264
9.2.1 Installing Sass	264
9.2.2 Compiling Sass Files to CSS	265
9.2.3 Using Variables	266
9.2.4 Using Operators	270
9.2.5 Using Branches	271
9.2.6 Using Loops	272
9.2.7 Using Functions	276
9.2.8 Implementing Custom Functions	278
9.2.9 Nesting Rules	281
9.2.10 Using Inheritance and Mixins	282
9.3 Summary and Outlook	285
9.3.1 Key Points	285
9.3.2 Recommended Reading	286
9.3.3 Outlook	286

10 Implementing Single-Page Applications 287

10.1 Introduction	287
10.2 Setup	290
10.3 Components: The Building Blocks of a React Application	293
10.3.1 State: The Local State of a Component	295
10.3.2 The Lifecycle of a Component	296
10.4 Styling Components	298
10.4.1 Inline Styling	298
10.4.2 CSS Classes and External Stylesheets	299
10.4.3 Overview of Other Styling Options	301
10.5 Component Hierarchies	302
10.6 Forms	307
10.7 The Context API	310

10.8	Routing	314
10.9	Summary and Outlook	316
10.9.1	Key Points	317
10.9.2	Recommended Reading	317
10.9.3	Outlook	318

11 Implementing Mobile Applications 319

11.1	The Different Types of Mobile Applications	319
11.1.1	Native Applications	320
11.1.2	Mobile Web Applications	321
11.1.3	Hybrid Applications	323
11.1.4	Comparing the Different Approaches	324
11.2	Responsive Design	326
11.2.1	Introduction: What Is Responsive Design?	326
11.2.2	Viewports	328
11.2.3	Media Queries	330
11.2.4	Flexible Layouts	333
11.3	Cross-Platform Development with React Native	338
11.3.1	The Principle of React Native	338
11.3.2	Installation and Project Initialization	339
11.3.3	Starting the Application	340
11.3.4	The Basic Structure of a React Native Application	343
11.3.5	User Interface Components	344
11.3.6	Building and Publishing Applications	349
11.4	Summary and Outlook	349
11.4.1	Key Points	349
11.4.2	Recommended Reading	350
11.4.3	Outlook	350

12 Understanding and Using Web Architectures 351

12.1	Layered Architectures	352
12.1.1	Basic Structure of Layered Architectures	352
12.1.2	Client-Server Architecture (Two-Tier Architecture)	353
12.1.3	Multi-Tier Architecture	355

- 12.2 Monoliths and Distributed Architectures** 358
 - 12.2.1 Monolithic Architecture 358
 - 12.2.2 Service-Oriented Architecture 359
 - 12.2.3 Microservice Architecture 361
 - 12.2.4 Component-Based Architecture 362
 - 12.2.5 Microfrontends Architecture 363
 - 12.2.6 Messaging Architecture 364
 - 12.2.7 Web Service Architecture 366
- 12.3 MV* Architectures** 367
 - 12.3.1 Model-View-Controller 367
 - 12.3.2 Model-View-Presenter 370
 - 12.3.3 Model-View-Viewmodel 370
- 12.4 Summary and Outlook** 371
 - 12.4.1 Key Points 371
 - 12.4.2 Recommended Reading 372
 - 12.4.3 Outlook 372

13 Using Programming Languages on the Server Side

373

- 13.1 Types of Programming Languages** 374
 - 13.1.1 Programming Languages by Degree of Abstraction 374
 - 13.1.2 Compiled and Interpreted Programming Languages 375
- 13.2 Programming Paradigms** 378
 - 13.2.1 Imperative and Declarative Programming 378
 - 13.2.2 Object-Oriented Programming 379
 - 13.2.3 Functional Programming 384
- 13.3 What Are the Programming Languages?** 385
 - 13.3.1 Rankings of Programming Languages 385
 - 13.3.2 Which Programming Language Should You Learn? 388
 - 13.3.3 But Seriously Now: Which Programming Language Should You Learn? 394
- 13.4 Summary and Outlook** 395
 - 13.4.1 Key Points 395
 - 13.4.2 Recommended Reading 396
 - 13.4.3 Outlook 397

14 Using JavaScript on the Server Side 399

14.1 JavaScript on Node.js	399
14.1.1 Node.js Architecture	400
14.1.2 A First Program	403
14.1.3 Package Management	405
14.2 Using the Integrated Modules	409
14.2.1 Reading Files	411
14.2.2 Writing Files	412
14.2.3 Deleting Files	413
14.3 Implementing a Web Server	413
14.3.1 Preparations	414
14.3.2 Providing Static Files	416
14.3.3 Using the Express.js Web Framework	420
14.3.4 Processing Form Data	421
14.4 Summary and Outlook	423
14.4.1 Key Points	424
14.4.2 Recommended Reading	424
14.4.3 Outlook	424

15 Using the PHP Language 425

15.1 Introduction to the PHP Language	425
15.2 Installing PHP and the Web Server Locally	425
15.3 Variables, Data Types, and Operators	427
15.3.1 Using Variables	428
15.3.2 Using Constants	432
15.3.3 Using Operators	432
15.4 Using Control Structures	435
15.4.1 Conditional Statements	435
15.4.2 Loops	437
15.5 Functions and Error Handling	439
15.5.1 Defining Functions	439
15.5.2 Function Parameters	439
15.5.3 Defining Return Values	441
15.5.4 Using Data Types	441

15.5.5	Anonymous Functions	442
15.5.6	Declaring Variable Functions	443
15.5.7	Arrow Functions	443
15.5.8	Responding to Errors	443
15.6	Using Classes and Objects	445
15.6.1	Writing Classes	445
15.6.2	Creating Objects	445
15.6.3	Class Constants	446
15.6.4	Visibility	446
15.6.5	Inheritance	447
15.6.6	Class Abstraction	448
15.6.7	More Features	449
15.7	Developing Dynamic Websites with PHP	450
15.7.1	Creating and Preparing a Form	450
15.7.2	Receiving Form Data	452
15.7.3	Verifying Form Data	452
15.8	Summary and Outlook	460
15.8.1	Key Points	460
15.8.2	Recommended Reading	461
15.8.3	Outlook	462

16 Implementing Web Services 463

16.1	Introduction	463
16.2	SOAP	465
16.2.1	The Workflow with SOAP	466
16.2.2	Description of Web Services with WSDL	467
16.2.3	Structure of SOAP Messages	469
16.2.4	Conclusion	470
16.3	REST	471
16.3.1	The Workflow with REST	471
16.3.2	The Principles of REST	472
16.3.3	Implementing a REST API	476
16.3.4	Calling a REST API	483
16.4	GraphQL	488
16.4.1	The Disadvantages of REST	488
16.4.2	The Workflow of GraphQL	491

16.5 Summary and Outlook	493
16.5.1 Key Points	493
16.5.2 Recommended Reading	494
16.5.3 Outlook	494

17 Storing Data in Databases 495

17.1 Relational Databases	496
17.1.1 The Functionality of Relational Databases	496
17.1.2 The SQL Language	498
17.1.3 Real-Life Example: Using Relational Databases in Node.js	506
17.1.4 Object-Relational Mappings	515
17.2 Non-Relational Databases	518
17.2.1 Relational versus Non-Relational Databases	518
17.2.2 The Functionality of Non-Relational Databases	519
17.2.3 Key-Value Databases	519
17.2.4 Document-Oriented Databases	520
17.2.5 Graph Databases	522
17.2.6 Column-Oriented Databases	523
17.3 Summary and Outlook	524
17.3.1 Key Points	524
17.3.2 Recommended Reading	525
17.3.3 Outlook	526

18 Testing Web Applications 527

18.1 Automated Tests	527
18.1.1 Introduction	528
18.1.2 Types of Tests	529
18.1.3 Test-Driven Development	531
18.1.4 Running Automated Tests in JavaScript	534
18.2 Test Coverage	537
18.2.1 Introduction	537
18.2.2 Determining Test Coverage in JavaScript	538
18.3 Test Doubles	539
18.3.1 The Problem with Dependencies	540
18.3.2 Replacing Dependencies with Test Doubles	540

18.3.3	Spies	541
18.3.4	Stubs	543
18.3.5	Mock Objects	543
18.4	Summary and Outlook	544
18.4.1	Key Points	544
18.4.2	Recommended Reading	545
18.4.3	Outlook	545

19 Deploying and Hosting Web Applications

547

19.1	Introduction	547
19.1.1	Building, Deploying, and Hosting	548
19.1.2	Types of Deployment	549
19.1.3	Types of Hosting	552
19.1.4	Requirements for Servers	555
19.2	Container Management	557
19.2.1	Docker	557
19.2.2	Real-Life Example: Packaging a Web Application using Docker	559
19.2.3	Number of Docker Images	565
19.2.4	Docker Compose	567
19.3	Summary and Outlook	569
19.3.1	Key Points	569
19.3.2	Recommended Reading	570
19.3.3	Outlook	570

20 Securing Web Applications

571

20.1	Vulnerabilities	572
20.1.1	Open Web Application Security Project	572
20.1.2	Injection	572
20.1.3	Broken Authentication	574
20.1.4	Sensitive Data Exposure	574
20.1.5	XML External Entities	575
20.1.6	Broken Access Control	575
20.1.7	Security Misconfiguration	576
20.1.8	Cross-Site Scripting	577
20.1.9	Insecure Deserialization	577

20.1.10	Using Components with Known Vulnerabilities	578
20.1.11	Insufficient Logging and Monitoring	579
20.1.12	Outlook	579
20.2	Encryption and Cryptography	579
20.2.1	Symmetric Cryptography	580
20.2.2	Asymmetric Cryptography	581
20.2.3	SSL, TLS, and HTTPS	582
20.3	Same-Origin Policies, Content Security Policies, and Cross-Origin Resource Sharing	584
20.3.1	Same Origin Policy	584
20.3.2	Cross-Origin Resource Sharing	585
20.3.3	Content Security Policy	587
20.4	Authentication	593
20.4.1	Basic Authentication	593
20.4.2	Session-Based Authentication	594
20.4.3	Token-Based Authentication	595
20.5	Summary and Outlook	597
20.5.1	Key Points	597
20.5.2	Recommended Reading	598
20.5.3	Outlook	598

21 Optimizing the Performance of Web Applications 599

21.1	Introduction	599
21.1.1	What Should Be Optimized and Why?	600
21.1.2	How Can Performance Be Measured?	601
21.1.3	Which Tools Are Available for Measuring Performance?	605
21.2	Options for Optimization	609
21.2.1	Optimizing Connection Times	609
21.2.2	Using a Server-Side Cache	611
21.2.3	Optimizing Images	612
21.2.4	Using a Client-Side Cache	615
21.2.5	Minifying the Code	618
21.2.6	Compressing Files	622
21.2.7	Lazy Loading: Loading Data Only When Needed	623
21.2.8	Preloading Data	623

21.3 Summary and Outlook	627
21.3.1 Key Points	628
21.3.2 Recommended Reading	629
21.3.3 Outlook	629

22 Organizing and Managing Web Projects 631

22.1 Types of Version Control Systems	632
22.1.1 Central Version Control Systems	632
22.1.2 Decentralized Version Control Systems	633
22.2 The Git Version Control System	635
22.2.1 How Git Stores Data	635
22.2.2 The Different Areas of Git	636
22.2.3 Installation	636
22.2.4 Creating a New Git Repository	638
22.2.5 Transferring Changes to the Staging Area	640
22.2.6 Committing Changes to the Local Repository	641
22.2.7 Committing Changes to the Remote Repository	643
22.2.8 Transferring Changes from the Remote Repository	644
22.2.9 Working in a New Branch	645
22.2.10 Transferring Changes from a Branch	647
22.3 Summary and Outlook	648
22.3.1 Key Points	648
22.3.2 Recommended Reading	650
22.3.3 Outlook	650

23 Managing Web Projects 651

23.1 Classic Project Management versus Agile Project Management	651
23.1.1 Classic Project Management	652
23.1.2 Agile Project Management	653
23.2 Agile Project Management Based on Scrum	654
23.2.1 The Scrum Workflow	654
23.2.2 The Roles of Scrum	657
23.2.3 Events in Scrum	659
23.2.4 Artifacts in Scrum	663

23.3	Summary and Outlook	665
23.3.1	Key Points	665
23.3.2	Recommended Reading	666
23.3.3	Outlook	666

Appendices	667
-------------------------	------------

A	HTTP	669
B	HTML Elements	691
C	Tools and Command References	703
D	Conclusion	715
E	The Author	717

Index	719
-------------	-----