

Inhalt

Geleitwort	17
Vorwort	19

1 Die Grundlagen verstehen 23

1.1 Begrifflichkeiten	23
1.1.1 Client und Server	23
1.1.2 Zusammenhang von URLs, Domains und IP-Adressen	24
1.2 Aufbau von Webapplikationen	26
1.2.1 Webseiten erstellen mit HTML (Hypertext Markup Language)	27
1.2.2 Webseiten gestalten mit CSS (Cascading Style Sheets)	27
1.2.3 Webseiten interaktiv machen mit JavaScript	28
1.2.4 Webseiten dynamisch machen mit serverseitiger Logik	30
1.3 Fullstack-Entwicklung	31
1.3.1 Was sind Software-Stacks?	31
1.3.2 Welche Stacks gibt es?	32
1.3.3 Was ist ein Fullstack-Entwickler?	33
1.3.4 Fokus in diesem Buch	35
1.4 Tools für Fullstack-Entwickler	38
1.4.1 Editoren	39
1.4.2 Entwicklungsumgebungen	40
1.4.3 Browser	41
1.5 Zusammenfassung und Ausblick	45
1.5.1 Die wichtigsten Punkte	46
1.5.2 Ausblick	46

2 Webseiten strukturieren mit HTML 47

2.1 Einführung	47
2.1.1 Versionen	48
2.1.2 Elemente und Attribute verwenden	48
2.1.3 Webseiten als HTML-Dokumente speichern	51

2.2	Die wichtigsten Elemente verwenden	52
2.2.1	Überschriften, Absätze und sonstige Textformatierungen verwenden	52
2.2.2	Listen erstellen	53
2.2.3	Links definieren	56
2.2.4	Bilder einbinden	62
2.2.5	Daten in Tabellen strukturieren	63
2.2.6	Formulare definieren	70
2.2.7	Weitere Informationen	74
2.3	Zusammenfassung und Ausblick	76
2.3.1	Die wichtigsten Punkte	76
2.3.2	Weiterführende Literatur	76
2.3.3	Ausblick	77
3	Webseiten gestalten mit CSS	79
3.1	Einführung	79
3.1.1	Das Prinzip von CSS	80
3.1.2	CSS in HTML einbinden	81
3.1.3	Selektoren	86
3.1.4	Kaskadierung und Spezifität	89
3.1.5	Vererbung	92
3.2	Farben und Textformatierungen anwenden	92
3.2.1	Textfarbe und Hintergrundfarbe definieren	92
3.2.2	Texte gestalten	94
3.3	Listen und Tabellen	105
3.3.1	Listen gestalten	105
3.3.2	Tabellen gestalten	110
3.4	Die verschiedenen Layoutsysteme verstehen	116
3.4.1	Grundlagen der Positionierung mit CSS	116
3.4.2	Float-Layout	118
3.4.3	Flexbox-Layout	121
3.4.4	Grid-Layout	128
3.5	Zusammenfassung und Ausblick	133
3.5.1	Die wichtigsten Punkte	134
3.5.2	Weiterführende Literatur	134
3.5.3	Ausblick	135

4	Webseiten interaktiv machen mit JavaScript	137
<hr/>		
4.1	Einführung	138
4.1.1	JavaScript einbinden	138
4.1.2	Dialogfenster anzeigen	141
4.1.3	Die Entwicklerkonsole verwenden	141
4.1.4	Einführung in die Programmierung	143
4.2	Variablen, Konstanten, Datentypen und Operatoren	145
4.2.1	Variablen definieren	145
4.2.2	Konstanten definieren	146
4.2.3	Datentypen verwenden	146
4.2.4	Operatoren verwenden	148
4.3	Kontrollstrukturen	149
4.3.1	Bedingte Anweisungen und Verzweigungen verwenden	149
4.3.2	Schleifen verwenden	151
4.4	Funktionen und Fehlerbehandlung	153
4.4.1	Funktionen definieren und aufrufen	153
4.4.2	Funktionsparameter übergeben und auswerten	154
4.4.3	Rückgabewerte definieren	155
4.4.4	Auf Fehler reagieren	155
4.5	Objekte und Arrays	157
4.5.1	Objekte verwenden	157
4.5.2	Arrays verwenden	158
4.6	Zusammenfassung und Ausblick	160
4.6.1	Die wichtigsten Punkte	161
4.6.2	Weiterführende Literatur	162
4.6.3	Ausblick	162
5	Webprotokolle verwenden	163
<hr/>		
5.1	Hypertext Transfer Protocol	163
5.1.1	Request und Response	164
5.1.2	Aufbau von HTTP-Anfragen (HTTP-Requests)	166
5.1.3	Aufbau von HTTP-Antworten (HTTP-Responses)	167
5.1.4	Header	168
5.1.5	Methoden	172

- 5.1.6 Statuscodes 173
- 5.1.7 MIME-Typen 175
- 5.1.8 Cookies 177
- 5.1.9 HTTP über die Kommandozeile ausführen 180
- 5.2 Bidirektionale Kommunikation 182**
 - 5.2.1 Polling und Long Polling 182
 - 5.2.2 Server-Sent Events 184
 - 5.2.3 WebSockets 184
- 5.3 Zusammenfassung und Ausblick 186**
 - 5.3.1 Die wichtigsten Punkte 186
 - 5.3.2 Weiterführende Literatur 187
 - 5.3.3 Ausblick 187

6 Webformate verwenden 189

- 6.1 Datenformate 189**
 - 6.1.1 CSV 190
 - 6.1.2 XML 190
 - 6.1.3 JSON 196
- 6.2 Bildformate 201**
 - 6.2.1 Fotografien mit dem JPG-Format 201
 - 6.2.2 Grafiken und Animationen mit dem GIF-Format 202
 - 6.2.3 Grafiken mit dem PNG-Format 203
 - 6.2.4 Vektorgrafiken mit dem SVG-Format 204
 - 6.2.5 Alles besser mit dem WebP-Format 205
 - 6.2.6 Vergleich der Bildformate 206
 - 6.2.7 Programme für die Bildbearbeitung 207
- 6.3 Video- und Audioformate 209**
 - 6.3.1 Videoformate 209
 - 6.3.2 Audioformate 212
- 6.4 Zusammenfassung und Ausblick 214**
 - 6.4.1 Die wichtigsten Punkte 214
 - 6.4.2 Weiterführende Literatur 215
 - 6.4.3 Ausblick 215

7	Web-APIs verwenden	217
<hr/>		
7.1	Webseiten dynamisch ändern mit der DOM API	218
7.1.1	Das Document Object Model (DOM)	218
7.1.2	Die verschiedenen Knotentypen	220
7.1.3	Elemente selektieren	222
7.1.4	Elemente verändern	224
7.1.5	Elemente erstellen, hinzufügen und löschen	225
7.1.6	Praxisbeispiel: dynamisches Erstellen einer Tabelle	226
7.2	Daten asynchron laden mit Ajax und der Fetch API	229
7.2.1	Synchrone vs. asynchrone Kommunikation	229
7.2.2	Daten per Ajax laden	232
7.2.3	Daten über die Fetch API laden	235
7.3	Weitere Web-APIs	235
7.3.1	Übersicht Web-APIs	236
7.3.2	Browsersupport für Web-APIs	240
7.4	Zusammenfassung und Ausblick	240
7.4.1	Die wichtigsten Punkte	240
7.4.2	Weiterführende Literatur	241
7.4.3	Ausblick	241
8	Webseiten für Barrierefreiheit optimieren	243
<hr/>		
8.1	Einführung	243
8.1.1	Einführung Barrierefreiheit	244
8.1.2	Nutzergruppen und assistive Technologien	245
8.1.3	Web Content Accessibility Guidelines (WCAG)	246
8.2	Bestandteile einer Webseite barrierefrei machen	250
8.2.1	Webseiten semantisch strukturieren	251
8.2.2	Überschriften richtig verwenden	254
8.2.3	Formulare barrierefrei machen	254
8.2.4	Tabellen barrierefrei machen	256
8.2.5	Bilder barrierefrei machen	262
8.2.6	Links barrierefrei machen	263
8.2.7	Accessible Rich Internet Applications (ARIA)	264
8.2.8	Sonstiges	267

8.3	Testen von Barrierefreiheit	270
8.3.1	Arten von Tests	270
8.3.2	Tools für das Testen	271
8.4	Zusammenfassung und Ausblick	274
8.4.1	Die wichtigsten Punkte	274
8.4.2	Weiterführende Literatur	275
8.4.3	Ausblick	276

9 Single-Page-Applikationen implementieren 277

9.1	Einführung	277
9.2	Setup	280
9.3	Komponenten – die Bausteine einer React-Applikation	284
9.3.1	State – der lokale Zustand einer Komponente	287
9.3.2	Der Lebenszyklus einer Komponente	288
9.4	Styling von Komponenten	289
9.4.1	Inline Styling	290
9.4.2	CSS-Klassen und externe Stylesheets	291
9.4.3	Überblick über weitere Styling-Möglichkeiten	293
9.5	Komponentenhierarchien	295
9.6	Formulare	299
9.7	Die Kontext-API	303
9.8	Routing	307
9.9	Zusammenfassung	310
9.9.1	Die wichtigsten Punkte	310
9.9.2	Weiterführende Literatur	311
9.9.3	Ausblick	312

10 Webarchitekturen verstehen und einsetzen 313

10.1	Schichtenarchitekturen	314
10.1.1	Grundsätzlicher Aufbau von Schichtenarchitekturen	314
10.1.2	Client-Server-Architektur (Zweischichtenarchitektur)	316
10.1.3	Mehrschichtenarchitektur	318

10.2 Monolithen und verteilte Architekturen	321
10.2.1 Monolithische Architektur	321
10.2.2 Serviceorientierte Architektur	322
10.2.3 Microservice-Architektur	324
10.2.4 Komponentenbasierte Architektur	325
10.2.5 Microfrontends-Architektur	326
10.2.6 Messaging-Architektur	327
10.2.7 Webservice-Architektur	331
10.3 MV*-Architekturen	331
10.3.1 Model View Controller	331
10.3.2 Model View Presenter	335
10.3.3 Model View ViewModel	335
10.4 Zusammenfassung und Ausblick	336
10.4.1 Die wichtigsten Punkte	336
10.4.2 Weiterführende Literatur	337
10.4.3 Ausblick	338

11 Programmiersprachen auf der Serverseite verwenden 339

11.1 Arten von Programmiersprachen	340
11.1.1 Unterteilung nach Abstraktionsgrad	340
11.1.2 Kompilierte und interpretierte Programmiersprachen	341
11.2 Programmierparadigmen	344
11.2.1 Imperative und deklarative Programmierung	344
11.2.2 Objektorientierte Programmierung	346
11.2.3 Funktionale Programmierung	351
11.3 Welche Programmiersprachen gibt es?	352
11.3.1 Programmiersprachen-Rankings	352
11.3.2 Welche Programmiersprache sollte ich lernen?	356
11.3.3 Jetzt ernsthaft: Welche Programmiersprache sollte ich lernen?	363
11.4 Zusammenfassung und Ausblick	364
11.4.1 Die wichtigsten Punkte	364
11.4.2 Weiterführende Literatur	365
11.4.3 Ausblick	366

12	JavaScript auf der Serverseite verwenden	367
12.1	JavaScript unter Node.js	367
12.1.1	Architektur von Node.js	368
12.1.2	Ein erstes Programm	371
12.1.3	Package Management	373
12.2	Die eingebauten Module verwenden	379
12.2.1	Dateien lesen	381
12.2.2	Dateien schreiben	382
12.2.3	Dateien löschen	383
12.3	Einen Webserver implementieren	383
12.3.1	Vorbereitungen	384
12.3.2	Statische Dateien bereitstellen	386
12.3.3	Das Web-Framework express verwenden	390
12.3.4	Formulardaten verarbeiten	392
12.4	Zusammenfassung und Ausblick	394
12.4.1	Die wichtigsten Punkte	394
12.4.2	Weiterführende Literatur	395
12.4.3	Ausblick	395
13	Webservices implementieren	397
13.1	Einführung	397
13.2	SOAP	399
13.2.1	Der Workflow bei SOAP	400
13.2.2	Beschreibung von Webservices mit WSDL	401
13.2.3	Aufbau von SOAP-Nachrichten	403
13.2.4	Fazit	405
13.3	REST	405
13.3.1	Der Workflow bei REST	405
13.3.2	Die Prinzipien von REST	406
13.3.3	Eine REST-API implementieren	412
13.3.4	Eine REST-API aufrufen	419
13.4	GraphQL	425
13.4.1	Die Nachteile von REST	425
13.4.2	Der Workflow von GraphQL	427

13.5 Zusammenfassung	430
13.5.1 Die wichtigsten Punkte	430
13.5.2 Weiterführende Literatur	431
13.5.3 Ausblick	431
14 Daten in Datenbanken speichern	433
<hr/>	
14.1 Relationale Datenbanken	434
14.1.1 Die Funktionsweise von relationalen Datenbanken	434
14.1.2 Die Sprache SQL	436
14.1.3 Praxisbeispiel: Relationale Datenbanken verwenden in Node.js	444
14.1.4 Objektrelationale Mappings	454
14.2 Nicht-relationale Datenbanken	457
14.2.1 Relationale vs. nicht-relationale Datenbanken	458
14.2.2 Die Funktionsweise von nicht-relationalen Datenbanken	458
14.2.3 Key-Value-Datenbanken	458
14.2.4 Dokumentenorientierte Datenbanken	460
14.2.5 Graphdatenbanken	462
14.2.6 Spaltenorientierte Datenbanken	463
14.3 Zusammenfassung	464
14.3.1 Die wichtigsten Punkte	464
14.3.2 Weiterführende Literatur	465
14.3.3 Ausblick	466
15 Webanwendungen testen	467
<hr/>	
15.1 Automatisierte Tests	468
15.1.1 Einführung	468
15.1.2 Arten von Tests	469
15.1.3 Testgetriebene Entwicklung	472
15.1.4 Automatisierte Tests in JavaScript ausführen	475
15.2 Testabdeckung	478
15.2.1 Einführung	478
15.2.2 Die Testabdeckung in JavaScript ermitteln	479
15.3 Test-Doubles	481
15.3.1 Das Problem mit Abhängigkeiten	481
15.3.2 Abhängigkeiten mit Test-Doubles ersetzen	482

15.3.3	Spies	483
15.3.4	Stubs	484
15.3.5	Mock-Objekte	485
15.4	Zusammenfassung	486
15.4.1	Die wichtigsten Punkte	486
15.4.2	Weiterführende Literatur	487
15.4.3	Ausblick	487

16 Webanwendungen deployen und hosten 489

16.1	Einführung	489
16.1.1	Build, Deployment und Hosting	490
16.1.2	Arten von Deployment	491
16.1.3	Arten von Hosting	494
16.1.4	Anforderungen an den Server	497
16.2	Container Management	500
16.2.1	Docker	500
16.2.2	Praxisbeispiel: Eine Webanwendung mit Docker paketieren	502
16.2.3	Anzahl der Docker Images	508
16.2.4	Docker Compose	510
16.3	Zusammenfassung	513
16.3.1	Die wichtigsten Punkte	513
16.3.2	Weiterführende Literatur	514
16.3.3	Ausblick	514

17 Webanwendungen absichern 515

17.1	Sicherheitslücken	516
17.1.1	OWASP	516
17.1.2	Einschleusen von schädlichem Code	517
17.1.3	Fehlerhafte Authentifizierung	519
17.1.4	Exposition sensibler Daten	519
17.1.5	Attacken über externe XML-Entitäten (XXE)	520
17.1.6	Defekte Zugriffskontrolle	520
17.1.7	Sicherheitsfehlkonfiguration	521
17.1.8	Einschleusen von schädlichem JavaScript-Code (XSS)	522

17.1.9	Unsichere Deserialisierung	523
17.1.10	Verwendung von Komponenten mit Sicherheitslücken	524
17.1.11	Unzureichende Protokollierung und Überwachung	524
17.1.12	Ausblick	525
17.2	Verschlüsselung und Kryptografie	525
17.2.1	Symmetrische Kryptografie	526
17.2.2	Asymmetrische Kryptografie	527
17.2.3	SSL, TSL und HTTPS	528
17.3	SOP, CSP und CORS	530
17.3.1	Same Origin Policy (SOP)	530
17.3.2	Cross-Origin Resource Sharing (CORS)	532
17.3.3	Content Security Policy (CSP)	534
17.4	Authentifizierung	540
17.4.1	Basic Authentication	540
17.4.2	Session Based Authentication	542
17.4.3	Token Based Authentication	543
17.5	Zusammenfassung	544
17.5.1	Die wichtigsten Punkte	544
17.5.2	Weiterführende Literatur	546
17.5.3	Ausblick	546
18	Webprojekte organisieren und verwalten	547
18.1	Arten von Versionsverwaltungssystemen	548
18.1.1	Zentrale Versionsverwaltungssysteme	548
18.1.2	Dezentrale Versionsverwaltungssysteme	549
18.2	Das Versionsverwaltungssystem Git	551
18.2.1	Wie Git die Daten speichert	551
18.2.2	Die verschiedenen Bereiche von Git	552
18.2.3	Installation	553
18.2.4	Ein neues Git-Repository anlegen	554
18.2.5	Änderungen in den Staging-Bereich übertragen	556
18.2.6	Änderungen in das lokale Repository übertragen	557
18.2.7	Änderungen in das Remote-Repository übertragen	559
18.2.8	Änderungen aus dem Remote-Repository übertragen	561
18.2.9	In einem neuen Branch arbeiten	561
18.2.10	Änderungen aus einem Branch übernehmen	564

18.3 Zusammenfassung	565
18.3.1 Die wichtigsten Punkte	565
18.3.2 Weiterführende Literatur	567
18.3.3 Ausblick	567

19 Webprojekte managen

19.1 Klassisches Projektmanagement vs. agiles Projektmanagement	570
19.1.1 Klassisches Projektmanagement	570
19.1.2 Agiles Projektmanagement	571
19.2 Agiles Projektmanagement mit Scrum	572
19.2.1 Der Workflow von Scrum	572
19.2.2 Die Rollen von Scrum	575
19.2.3 Die Ereignisse von Scrum	578
19.2.4 Die Artefakte von Scrum	582
19.3 Zusammenfassung und Ausblick	584
19.3.1 Die wichtigsten Punkte	584
19.3.2 Weiterführende Literatur	585
19.3.3 Ausblick	586

Anhang

A HTTP	589
B HTML-Elemente	613
C Tools und Befehlsreferenzen	625

Schlusswort	639
Index	641