

Inhalt

Materialien zum Buch	21
Geleitwort	23
Vorwort	25

1 Die Grundlagen verstehen 29

1.1 Begrifflichkeiten	29
1.1.1 Client und Server	29
1.1.2 Zusammenhang von URLs, Domains und IP-Adressen	30
1.2 Aufbau von Webapplikationen	33
1.2.1 Webseiten erstellen mit HTML (Hypertext Markup Language)	33
1.2.2 Webseiten gestalten mit CSS (Cascading Style Sheets)	33
1.2.3 Webseiten interaktiv machen mit JavaScript	34
1.2.4 Webseiten dynamisch machen mit serverseitiger Logik	36
1.3 Fullstack-Entwicklung	37
1.3.1 Was sind Software-Stacks?	37
1.3.2 Welche Stacks gibt es?	38
1.3.3 Was ist ein Fullstack-Entwickler?	39
1.3.4 Aufbau des Buches	42
1.4 Tools für Fullstack-Entwickler	45
1.4.1 Editoren	45
1.4.2 Entwicklungsumgebungen	47
1.4.3 Browser	48
1.5 Zusammenfassung und Ausblick	52
1.5.1 Die wichtigsten Punkte	52
1.5.2 Ausblick	53

2 Webseiten strukturieren mit HTML 55

2.1 Einführung	55
2.1.1 Versionen	56
2.1.2 Elemente und Attribute verwenden	56
2.1.3 Webseiten als HTML-Dokumente speichern	59

2.2	Die wichtigsten Elemente verwenden	60
2.2.1	Überschriften, Absätze und sonstige Textformatierungen verwenden	60
2.2.2	Listen erstellen	61
2.2.3	Links definieren	64
2.2.4	Bilder einbinden	69
2.2.5	Daten in Tabellen strukturieren	71
2.2.6	Formulare definieren	77
2.2.7	Weitere Informationen	82
2.3	Zusammenfassung und Ausblick	83
2.3.1	Die wichtigsten Punkte	83
2.3.2	Weiterführende Literatur	84
2.3.3	Ausblick	84

3 Webseiten gestalten mit CSS

3.1	Einführung	86
3.1.1	Das Prinzip von CSS	86
3.1.2	CSS in HTML einbinden	87
3.1.3	Selektoren	92
3.1.4	Kaskadierung und Spezifität	95
3.1.5	Vererbung	98
3.2	Farben und Textformatierungen anwenden	98
3.2.1	Textfarbe und Hintergrundfarbe definieren	98
3.2.2	Texte gestalten	100
3.3	Listen und Tabellen	111
3.3.1	Listen gestalten	111
3.3.2	Tabellen gestalten	116
3.4	Die verschiedenen Layoutsysteme verstehen	122
3.4.1	Grundlagen der Positionierung mit CSS	122
3.4.2	Float-Layout	124
3.4.3	Flexbox-Layout	128
3.4.4	Grid-Layout	135
3.5	Zusammenfassung und Ausblick	140
3.5.1	Die wichtigsten Punkte	141
3.5.2	Weiterführende Literatur	141
3.5.3	Ausblick	142

4 Webseiten interaktiv machen mit JavaScript 145

4.1	Einführung	146
4.1.1	JavaScript einbinden	146
4.1.2	Dialogfenster anzeigen	149
4.1.3	Die Entwicklerkonsole verwenden	150
4.1.4	Einführung in die Programmierung	151
4.2	Variablen, Konstanten, Datentypen und Operatoren	153
4.2.1	Variablen definieren	154
4.2.2	Konstanten definieren	154
4.2.3	Datentypen verwenden	155
4.2.4	Operatoren verwenden	156
4.3	Kontrollstrukturen verwenden	157
4.3.1	Bedingte Anweisungen und Verzweigungen verwenden	158
4.3.2	Schleifen verwenden	160
4.4	Funktionen und Fehlerbehandlung	161
4.4.1	Funktionen definieren und aufrufen	161
4.4.2	Funktionsparameter übergeben und auswerten	163
4.4.3	Rückgabewerte definieren	163
4.4.4	Auf Fehler reagieren	164
4.5	Objekte und Arrays	165
4.5.1	Objekte verwenden	166
4.5.2	Arrays verwenden	167
4.6	Zusammenfassung und Ausblick	168
4.6.1	Die wichtigsten Punkte	169
4.6.2	Weiterführende Literatur	170
4.6.3	Ausblick	170

5 Webprotokolle verwenden 171

5.1	Hypertext Transfer Protocol	171
5.1.1	Request und Response	172
5.1.2	Aufbau von HTTP-Anfragen (HTTP-Requests)	174
5.1.3	Aufbau von HTTP-Antworten (HTTP-Responses)	175
5.1.4	Header	176
5.1.5	Methoden	179
5.1.6	Statuscodes	181

5.1.7	MIME-Typen	182
5.1.8	Cookies	185
5.1.9	HTTP über die Kommandozeile ausführen	188
5.2	Bidirektionale Kommunikation	189
5.2.1	Polling und Long Polling	190
5.2.2	Server-Sent Events	191
5.2.3	WebSockets	192
5.3	Zusammenfassung und Ausblick	193
5.3.1	Die wichtigsten Punkte	193
5.3.2	Weiterführende Literatur	194
5.3.3	Ausblick	194

6 Webformate verwenden 195

6.1	Datenformate	196
6.1.1	CSV	196
6.1.2	XML	197
6.1.3	JSON	202
6.2	Bildformate	207
6.2.1	Fotografien mit dem JPG-Format	208
6.2.2	Grafiken und Animationen mit dem GIF-Format	208
6.2.3	Grafiken mit dem PNG-Format	209
6.2.4	Vektorgrafiken mit dem SVG-Format	209
6.2.5	Alles besser mit dem WebP-Format	211
6.2.6	Vergleich der Bildformate	212
6.2.7	Programme für die Bildbearbeitung	213
6.3	Video- und Audioformate	215
6.3.1	Videoformate	215
6.3.2	Audioformate	217
6.4	Zusammenfassung und Ausblick	219
6.4.1	Die wichtigsten Punkte	219
6.4.2	Weiterführende Literatur	220
6.4.3	Ausblick	220

7 Web-APIs verwenden 221

7.1 Webseiten dynamisch ändern mit der DOM API	222
7.1.1 Das Document Object Model (DOM)	222
7.1.2 Die verschiedenen Knotentypen	223
7.1.3 Elemente selektieren	226
7.1.4 Elemente verändern	228
7.1.5 Elemente erstellen, hinzufügen und löschen	229
7.1.6 Praxisbeispiel: dynamisches Erstellen einer Tabelle	230
7.2 Daten asynchron laden mit Ajax und der Fetch API	233
7.2.1 Synchrone vs. asynchrone Kommunikation	233
7.2.2 Daten per Ajax laden	236
7.2.3 Daten über die Fetch API laden	239
7.3 Weitere Web-APIs	240
7.3.1 Übersicht Web-APIs	240
7.3.2 Browsersupport für Web-APIs	244
7.4 Zusammenfassung und Ausblick	244
7.4.1 Die wichtigsten Punkte	244
7.4.2 Weiterführende Literatur	245
7.4.3 Ausblick	245

8 Webseiten für Barrierefreiheit optimieren 247

8.1 Einführung	247
8.1.1 Einführung Barrierefreiheit	248
8.1.2 Nutzergruppen und assistive Technologien	249
8.1.3 Web Content Accessibility Guidelines (WCAG)	250
8.2 Bestandteile einer Webseite barrierefrei machen	254
8.2.1 Webseiten semantisch strukturieren	255
8.2.2 Überschriften richtig verwenden	258
8.2.3 Formulare barrierefrei machen	258
8.2.4 Tabellen barrierefrei machen	260
8.2.5 Bilder barrierefrei machen	266
8.2.6 Links barrierefrei machen	267
8.2.7 Accessible Rich Internet Applications (ARIA)	268
8.2.8 Sonstiges	271

8.3	Testen von Barrierefreiheit	274
8.3.1	Arten von Tests	274
8.3.2	Tools für das Testen	275
8.4	Zusammenfassung und Ausblick	278
8.4.1	Die wichtigsten Punkte	278
8.4.2	Weiterführende Literatur	279
8.4.3	Ausblick	280

9 CSS vereinfachen mit CSS-Präprozessoren 281

9.1	Einführung	282
9.1.1	Die Funktionsweise von CSS-Präprozessoren	282
9.1.2	Features von CSS-Präprozessoren	283
9.1.3	Sass, Less und Stylus	284
9.2	Sass verwenden	285
9.2.1	Sass installieren	285
9.2.2	Sass-Dateien nach CSS kompilieren	286
9.2.3	Variablen verwenden	287
9.2.4	Operatoren verwenden	291
9.2.5	Verzweigungen verwenden	292
9.2.6	Schleifen verwenden	293
9.2.7	Funktionen verwenden	297
9.2.8	Eigene Funktionen implementieren	299
9.2.9	Regeln verschachteln	303
9.2.10	Vererbung und Mixins verwenden	304
9.3	Zusammenfassung und Ausblick	307
9.3.1	Die wichtigsten Punkte	307
9.3.2	Weiterführende Literatur	308
9.3.3	Ausblick	308

10 Single-Page-Applikationen implementieren 309

10.1	Einführung	309
10.2	Setup	312
10.3	Komponenten – die Bausteine einer React-Applikation	316
10.3.1	State – der lokale Zustand einer Komponente	318
10.3.2	Der Lebenszyklus einer Komponente	320

10.4 Styling von Komponenten	321
10.4.1 Inline Styling	322
10.4.2 CSS-Klassen und externe Stylesheets	323
10.4.3 Überblick über weitere Styling-Möglichkeiten	325
10.5 Komponentenhierarchien	327
10.6 Formulare	331
10.7 Die Kontext-API	335
10.8 Routing	339
10.9 Zusammenfassung und Ausblick	342
10.9.1 Die wichtigsten Punkte	342
10.9.2 Weiterführende Literatur	343
10.9.3 Ausblick	343

11 Mobile Anwendungen implementieren 345

11.1 Die unterschiedlichen Arten mobiler Anwendungen	345
11.1.1 Native Anwendungen	346
11.1.2 Mobile Webanwendungen	347
11.1.3 Hybridanwendungen	349
11.1.4 Vergleich der verschiedenen Ansätze	351
11.2 Responsive Design	353
11.2.1 Einführung: Was ist Responsive Design?	354
11.2.2 Viewport	355
11.2.3 Media Queries	358
11.2.4 Flexible Layouts	361
11.3 Cross Platform Development mit React Native	367
11.3.1 Das Prinzip von React Native	367
11.3.2 Installation und Projektinitialisierung	367
11.3.3 Die Anwendung starten	369
11.3.4 Das Grundgerüst einer React-Native-Anwendung	372
11.3.5 UI-Komponenten verwenden	373
11.3.6 Bauen und Veröffentlichen von Anwendungen	378
11.4 Zusammenfassung und Ausblick	379
11.4.1 Die wichtigsten Punkte	379
11.4.2 Weiterführende Literatur	380
11.4.3 Ausblick	380

12 Webarchitekturen verstehen und einsetzen 381

12.1 Schichtenarchitekturen	382
12.1.1 Grundsätzlicher Aufbau von Schichtenarchitekturen	382
12.1.2 Client-Server-Architektur (Zweischichtenarchitektur)	384
12.1.3 Mehrschichtenarchitektur	386
12.2 Monolithen und verteilte Architekturen	389
12.2.1 Monolithische Architektur	389
12.2.2 Serviceorientierte Architektur	390
12.2.3 Microservice-Architektur	391
12.2.4 Komponentenbasierte Architektur	393
12.2.5 Microfrontends-Architektur	394
12.2.6 Messaging-Architektur	395
12.2.7 Webservice-Architektur	397
12.3 MV*-Architekturen	398
12.3.1 Model View Controller	398
12.3.2 Model View Presenter	402
12.3.3 Model View ViewModel	402
12.4 Zusammenfassung und Ausblick	403
12.4.1 Die wichtigsten Punkte	403
12.4.2 Weiterführende Literatur	404
12.4.3 Ausblick	405

13 Programmiersprachen auf der Serverseite verwenden 407

13.1 Arten von Programmiersprachen	408
13.1.1 Unterteilung nach Abstraktionsgrad	408
13.1.2 Kompilierte und interpretierte Programmiersprachen	409
13.2 Programmierparadigmen	412
13.2.1 Imperative und deklarative Programmierung	412
13.2.2 Objektorientierte Programmierung	414
13.2.3 Funktionale Programmierung	419
13.3 Welche Programmiersprachen gibt es?	420
13.3.1 Programmiersprachen-Rankings	420
13.3.2 Welche Programmiersprache sollte ich lernen?	424
13.3.3 Jetzt ernsthaft: Welche Programmiersprache sollte ich lernen?	431

13.4 Zusammenfassung und Ausblick	432
13.4.1 Die wichtigsten Punkte	432
13.4.2 Weiterführende Literatur	433
13.4.3 Ausblick	434

14 JavaScript auf der Serverseite verwenden 435

14.1 JavaScript unter Node.js	436
14.1.1 Architektur von Node.js	436
14.1.2 Ein erstes Programm	439
14.1.3 Package Management	441
14.2 Die eingebauten Module verwenden	447
14.2.1 Dateien lesen	449
14.2.2 Dateien schreiben	450
14.2.3 Dateien löschen	451
14.3 Einen Webserver implementieren	452
14.3.1 Vorbereitungen	452
14.3.2 Statische Dateien bereitstellen	455
14.3.3 Das Web-Framework express verwenden	458
14.3.4 Formulardaten verarbeiten	460
14.4 Zusammenfassung und Ausblick	462
14.4.1 Die wichtigsten Punkte	463
14.4.2 Weiterführende Literatur	463
14.4.3 Ausblick	463

15 Die Sprache PHP verwenden 465

15.1 Einführung in die Sprache PHP	465
15.2 PHP und Webserver lokal installieren	466
15.3 Variablen, Datentypen und Operatoren	467
15.3.1 Variablen verwenden	468
15.3.2 Konstanten verwenden	472
15.3.3 Operatoren verwenden	473
15.4 Kontrollstrukturen verwenden	476
15.4.1 Bedingte Anweisungen	476
15.4.2 Schleifen	478

15.5 Funktionen und Fehlerbehandlung	480
15.5.1 Funktionen definieren	480
15.5.2 Funktionsparameter	481
15.5.3 Rückgabewerte definieren	482
15.5.4 Arbeiten mit Datentypen	482
15.5.5 Anonyme Funktionen	483
15.5.6 Variablenfunktionen deklarieren	484
15.5.7 Pfeilfunktionen	484
15.5.8 Auf Fehler reagieren	485
15.6 Klassen und Objekte verwenden	486
15.6.1 Klassen schreiben	486
15.6.2 Objekte erstellen	487
15.6.3 Klassenkonstanten	487
15.6.4 Sichtbarkeit	488
15.6.5 Vererbung	489
15.6.6 Klassenabstraktion	490
15.6.7 Weitere Features	491
15.7 Dynamische Webseiten mit PHP entwickeln	491
15.7.1 Formular erstellen und vorbereiten	492
15.7.2 Formulardaten empfangen	493
15.7.3 Formulardaten prüfen	494
15.8 Zusammenfassung und Ausblick	503
15.8.1 Die wichtigsten Punkte	503
15.8.2 Weiterführende Literatur	504
15.8.3 Ausblick	504

16 Webservices implementieren 505

16.1 Einführung	505
16.2 SOAP	507
16.2.1 Der Workflow bei SOAP	508
16.2.2 Beschreibung von Webservices mit WSDL	509
16.2.3 Aufbau von SOAP-Nachrichten	511
16.2.4 Fazit	513
16.3 REST	513
16.3.1 Der Workflow bei REST	513
16.3.2 Die Prinzipien von REST	515

16.3.3	Eine REST-API implementieren	520
16.3.4	Eine REST-API aufrufen	527
16.4	GraphQL	533
16.4.1	Die Nachteile von REST	533
16.4.2	Der Workflow von GraphQL	535
16.5	Zusammenfassung und Ausblick	537
16.5.1	Die wichtigsten Punkte	538
16.5.2	Weiterführende Literatur	539
16.5.3	Ausblick	539

17 Daten in Datenbanken speichern 541

17.1	Relationale Datenbanken	542
17.1.1	Die Funktionsweise von relationalen Datenbanken	542
17.1.2	Die Sprache SQL	544
17.1.3	Praxisbeispiel: Relationale Datenbanken verwenden in Node.js	552
17.1.4	Objektrelationale Mappings	563
17.2	Nicht relationale Datenbanken	565
17.2.1	Relationale vs. Nicht relationale Datenbanken	566
17.2.2	Die Funktionsweise von nicht relationalen Datenbanken	566
17.2.3	Key-Value-Datenbanken	566
17.2.4	Dokumentenorientierte Datenbanken	568
17.2.5	Graphdatenbanken	570
17.2.6	Spaltenorientierte Datenbanken	571
17.3	Zusammenfassung und Ausblick	572
17.3.1	Die wichtigsten Punkte	572
17.3.2	Weiterführende Literatur	573
17.3.3	Ausblick	574

18 Webanwendungen testen 575

18.1	Automatisierte Tests	576
18.1.1	Einführung	576
18.1.2	Arten von Tests	577
18.1.3	Testgetriebene Entwicklung	580
18.1.4	Automatisierte Tests in JavaScript ausführen	583

18.2 Testabdeckung	586
18.2.1 Einführung	586
18.2.2 Die Testabdeckung in JavaScript ermitteln	587
18.3 Test-Doubles	589
18.3.1 Das Problem mit Abhängigkeiten	589
18.3.2 Abhängigkeiten mit Test-Doubles ersetzen	590
18.3.3 Spies	591
18.3.4 Stubs	592
18.3.5 Mock-Objekte	593
18.4 Zusammenfassung und Ausblick	594
18.4.1 Die wichtigsten Punkte	594
18.4.2 Weiterführende Literatur	595
18.4.3 Ausblick	595

19 Webanwendungen deployen und hosten 597

19.1 Einführung	597
19.1.1 Build, Deployment und Hosting	598
19.1.2 Arten von Deployment	600
19.1.3 Arten von Hosting	602
19.1.4 Anforderungen an den Server	605
19.2 Container Management	608
19.2.1 Docker	608
19.2.2 Praxisbeispiel: Eine Webanwendung mit Docker paketieren	609
19.2.3 Anzahl der Docker Images	616
19.2.4 Docker Compose	618
19.3 Zusammenfassung und Ausblick	620
19.3.1 Die wichtigsten Punkte	621
19.3.2 Weiterführende Literatur	621
19.3.3 Ausblick	621

20 Webanwendungen absichern 623

20.1 Sicherheitslücken	624
20.1.1 OWASP	624
20.1.2 Einschleusen von schädlichem Code	625

20.1.3	Fehlerhafte Authentifizierung	627
20.1.4	Exposition sensibler Daten	627
20.1.5	Attacken über externe XML-Entitäten (XXE)	628
20.1.6	Defekte Zugriffskontrolle	628
20.1.7	Sicherheitsfehlkonfiguration	629
20.1.8	Einschleusen von schädlichem JavaScript-Code (XSS)	630
20.1.9	Unsichere Deserialisierung	631
20.1.10	Verwendung von Komponenten mit Sicherheitslücken	631
20.1.11	Unzureichende Protokollierung und Überwachung	632
20.1.12	Ausblick	632
20.2	Verschlüsselung und Kryptografie	633
20.2.1	Symmetrische Kryptografie	633
20.2.2	Asymmetrische Kryptografie	634
20.2.3	SSL, TLS und HTTPS	635
20.3	SOP, CSP und CORS	637
20.3.1	Same Origin Policy (SOP)	638
20.3.2	Cross-Origin Resource Sharing (CORS)	639
20.3.3	Content Security Policy (CSP)	641
20.4	Authentifizierung	647
20.4.1	Basic Authentication	647
20.4.2	Session Based Authentication	649
20.4.3	Token Based Authentication	650
20.5	Zusammenfassung und Ausblick	651
20.5.1	Die wichtigsten Punkte	652
20.5.2	Weiterführende Literatur	653
20.5.3	Ausblick	653

21 Die Performance von Webanwendungen optimieren

655

21.1	Einführung	656
21.1.1	Was und warum sollte optimiert werden?	656
21.1.2	Wie kann die Performance gemessen werden?	657
21.1.3	Welche Tools für die Messung der Performance gibt es?	661
21.2	Möglichkeiten der Optimierung	665
21.2.1	Verbindungszeiten optimieren	666
21.2.2	Einen serverseitigen Cache verwenden	668

21.2.3	Bilder optimieren	669
21.2.4	Einen clientseitigen Cache verwenden	672
21.2.5	Den Code minifizieren	675
21.2.6	Dateien komprimieren	679
21.2.7	Lazy Loading: Daten erst bei Bedarf laden	680
21.2.8	Daten im Voraus laden	681
21.3	Zusammenfassung und Ausblick	684
21.3.1	Weiterführende Literatur	686
21.3.2	Ausblick	686

22 Webprojekte organisieren und verwalten 687

22.1	Arten von Versionsverwaltungssystemen	688
22.1.1	Zentrale Versionsverwaltungssysteme	688
22.1.2	Dezentrale Versionsverwaltungssysteme	689
22.2	Das Versionsverwaltungssystem Git	691
22.2.1	Wie Git die Daten speichert	691
22.2.2	Die verschiedenen Bereiche von Git	692
22.2.3	Installation	693
22.2.4	Ein neues Git-Repository anlegen	694
22.2.5	Änderungen in den Staging-Bereich übertragen	696
22.2.6	Änderungen in das lokale Repository übertragen	697
22.2.7	Änderungen in das Remote-Repository übertragen	699
22.2.8	Änderungen aus dem Remote-Repository übertragen	701
22.2.9	In einem neuen Branch arbeiten	702
22.2.10	Änderungen aus einem Branch übernehmen	704
22.3	Zusammenfassung und Ausblick	705
22.3.1	Die wichtigsten Punkte	705
22.3.2	Weiterführende Literatur	707
22.3.3	Ausblick	707

23 Webprojekte managen 709

23.1	Klassisches Projektmanagement vs. agiles Projektmanagement	710
23.1.1	Klassisches Projektmanagement	710
23.1.2	Agiles Projektmanagement	711

23.2	Agiles Projektmanagement mit Scrum	712
23.2.1	Der Workflow von Scrum	712
23.2.2	Die Rollen von Scrum	715
23.2.3	Die Ereignisse von Scrum	718
23.2.4	Die Artefakte von Scrum	722
23.3	Zusammenfassung und Ausblick	724
23.3.1	Die wichtigsten Punkte	724
23.3.2	Weiterführende Literatur	725
23.3.3	Ausblick	726

Anhang 727

A	HTTP	729
B	HTML-Elemente	753
C	Tools und Befehlsreferenzen	767
D	Schlusswort	781
Index		783