KOMPLETT IN FARBE

Dr. Thomas Pissulla

HTML UND CSS KOMPENDIUM

PROFESSIONELLES WEBDEVELOPMENT LERNEN

HTML CSS



Eine umfassende Einführung in HTML und CSS!

- Umfassende HTML und CSS Grundlagen einfach erklärt
- Responsive Webdesign, JavaScript und jQuery, HTML-APIs, u.v.m.
- Übungsaufgaben mit Musterlösungen nach jedem Kapitel

Inklusive eBook zum Download

{BMU VERLAG}

HTML und CSS Kompendium

Professionelles Webdevelopment lernen

Dr. Thomas Pissulla

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie;
detaillierte bibliografische Informationen sind im Internet über http://dnb.d-nb.de abrufbar.

©2023 BMU Media GmbH
www.bmu-verlag.de
info@bmu-verlag.de

Lektorat: Lektormeister
Einbandgestaltung: Pro ebookcovers Angie
Druck und Bindung: WirMachenDruck

Taschenbuch-ISBN: 978-3-96645-064-5
Hardcover-ISBN: 978-3-96645-065-2
eBook-ISBN: 978-3-96645-063-8

Dieses Werk ist urheberrechtlich geschützt.
Alle Rechte (Übersetzung, Nachdruck und Vervielfältigung) vorbehalten. Kein Teil des Werks darf ohne schriftliche Genehmigung des Verlags in irgendeiner Form – auch nicht für Zwecke der Unterrichtsgestaltung- reproduziert, verarbeitet, vervielfältigt oder verbreitet werden.

Dieses Buch wurde mit größter Sorgfalt erstellt, ungeachtet dessen können weder Verlag noch Autor, Herausgeber oder Übersetzer für mögliche Fehler und deren Folgen eine juristische Verantwortung oder irgendeine Haftung übernehmen. Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären.

HTML und CSS Kompendium

Inhaltsverzeichnis

1. Einführung 9

1.1 Über dieses Buch ... 9
1.2 Was ist HTML und wozu dient es? ... 10
1.3 CSS und HTML ... 11

2. Begriffe 14

3. Erste Schritte 17

3.1 Einrichtung der Arbeitsumgebung ... 19
3.1.1 Lokaler Webserver ... 20
3.1.2 Editoren ... 29
3.1.3 Browser ... 32

4. HTML-Grundlagen 36

4.1 HTML4 vs. HTML5 ... 36
4.2 Dateistruktur im Webserver ... 36
4.2.1 Starten des Apache-Webservers bei Windows ... 36
4.2.2 Starten des Apache-Webservers bei macOS ... 39
4.2.3 Weiterführende Erläuterungen ... 43
4.3 Aufbau einer HTML-Datei ... 45
4.3.1 Wie wird eine HTML-Datei geladen? ... 48
4.4 Der Dateikopf ... 49
4.4.1 Seitentitel ... 49
4.4.2 Metainformationen ... 50
4.4.3 Zeichenkodierung ... 50
4.4.4 Metadaten für Suchmaschinen ... 51
4.4.5 Weitere Metadaten für Browser ... 52
4.4.6 Viewport ... 53
4.4.7 Weitere Tags im Dateikopf ... 54
4.5 Der Dateikörper ... 54
4.5.1 Absätze und Zeilenumbrüche (<p>, <br>, <hr>) ... 56
4.5.2 Links (<a>, <base>) ... 62
4.5.3 Überschriften (<h1>, <h2>, <h3>, …, Sprungmarken) ... 69
4.5.4 Listen ... 74
4.5.5 Tabellen ... 78
4.5.6 Bilder ... 84
4.5.7 Einbindung verschiedener anderer Medien (<video>, <audio>, <iframe>) ... 89
4.5.8 Formulare ... 99
4.5.9 Favicons und Shortcut-Icons ... 132
4.5.10 Semantische Elemente ... 135

4.5.11 Sonstige Formatierungselemente ... 137
4.5.12 HTML-Sonderzeichen ... 139
4.5.13 Kommentare ... 139
4.5.14 HTML-Validierung ... 140

5. CSS-Grundlagen 142

5.1 Grundlagen der Formatierung mit CSS ... 142
5.1.1 Eine wesentliche CSS-Eigenschaft: Vererbung ... 142
5.1.2 Wohin mit dem CSS-Code? ... 143
5.1.3 Selektoren, CSS-Klassen vs. individuelle Elemente ... 151
5.1.4 Ein Hinweis zum Browser-Cache ... 154
5.1.5 CSS-Kommentare ... 155
5.2 Formatierungsmöglichkeit für HTML-Elemente ... 155
5.2.1 Text formatieren ... 156
5.2.2 Farben und Füllungen ... 164
5.2.3 Rahmen, Schattierungen ... 168
5.2.4 Abstände, Breite, Höhe ... 173
5.2.5 Breite, Höhe, Darstellungsebene und Positionierung von Elementen ... 177
5.2.6 Links formatieren, Pseudoklassen und Pseudoelemente ... 188
5.2.7 Bilder formatieren, Hintergrundbilder ... 195
5.2.8 Listen formatieren (Navigationsmenüs gestalten) ... 210
5.2.9 Tabellen formatieren ... 218
5.2.10 Formulare formatieren ... 222
5.2.11 Standardwerte (initial, inherit und unset) ... 230

6. Fortgeschrittene Techniken 237

6.1 Einbinden externer Medien und ein technischer Ausflug in die Welt der DSGVO 237
6.1.1 Einbinden externer Schriftarten mit Google Fonts ... 238
6.1.2 Einbinden externer Icons mit Font Awesome ... 249
6.2 Animationen und andere Effekte ... 256
6.2.1 Animationen ausschließlich mit CSS ... 256
6.2.2 CSS-Transitions ... 264
6.2.3 Farbverläufe, Pfeile und schräge Kanten mit CSS ... 268
6.2.4 CSS-Kombinatoren und fortgeschrittene Selektoren ... 281
6.3 Fortgeschrittene CSS-Layouts ... 298
6.3.1 Mehrspaltenumgebungen ... 298
6.3.2 Flexboxen ... 303
6.3.3 CSS-Grid ... 309
6.4 Responsives Webdesign – „Mobile First“ ... 315
6.5 Ausgabe auf Druckern ... 335
6.6 Browserkompatibilität, CSS-Validierung ... 342

7. Komplexe Webseiten 344

7.1 Code sparen mit SSIs ... 344
7.2 CSS-Frameworks ... 349
7.3 CSS-Präprozessoren (SASS, LESS) ... 354
7.3.1 Installation von Koala für MacOS ... 356
7.3.2 Installation von Koala für Windows ... 357

7.3.3 Arbeiten mit Koala ... 360
7.3.4 SASS-Variablen ... 364
7.3.5 SASS-Nesting ... 367
7.3.6 SASS-Mixins ... 374
7.3.7 SASS-Extends ... 376
7.3.8 Aufteilen von SCSS-Code auf mehrere Dateien ... 377
7.3.9 Fortgeschrittene SASS-Funktionalität ... 380
7.3.10 Anpassung von Bootstrap mittels SASS ... 380
7.3.11 Besonderheiten von LESS ... 389

8. Dynamische Webseiten 391

8.1 Exkurs: JavaScript ... 392
8.1.1 Die Syntax und wichtige Befehle ... 392
8.1.2 Wohin mit dem JavaScript-Code? ... 418
8.1.3 jQuery ... 419
8.2 HTML-APIs ... 445
8.2.1 Geolocation ... 446
8.2.2 Web Storage ... 451

9. Online gehen 455

9.1 Vor dem Online gehen – lokale Tests ... 455
9.2 Verschiedene Arten von Webhostern ... 462
9.3 Die Seite auf den Server laden (FTP) ... 468
9.3.1 FTP-Plugin für VS Code installieren ... 469
9.3.2 Welche Dateien gehören auf den Server? (Deployment) ... 474
9.4 Ladezeitoptimierung ... 481
9.4.1 Einflussfaktoren für lange Ladezeiten ... 481
9.4.2 Einbindung von JS/CSS (preload und defer) ... 482
9.4.3 Uglyfy/Minify ... 484
9.4.4 Lazy-Loading ... 487
9.4.5 Ausgabe von Bildern entsprechend der Anzeigegröße ... 487
9.4.6 Nutzung moderner Bildformate ... 494
9.4.7 Caching ... 504
9.4.8 Google Pagespeed Insights ... 504
9.5 Serverkonfiguration für Webdesigner ... 506
9.5.1 Ändern der Index-Datei ... 507
9.5.2 Anpassung der Cache-Richtlinien ... 508
9.5.3 Weiterleitungen/Rewrites (suchmaschinenfreundliche URLs) ... 510

10. Wie geht es weiter? 512

10.1 Projekt 1: Eine Single-Page-Website (Onepager) ... 513
10.1.1 Grundsätzliches Layout ... 514
10.1.2 Scrollverhalten und JavaScript des Onepagers ... 527
10.2 Projekt 2: Ein Onlineshop basierend auf WordPress ... 530
10.2.1 Das HTML-Grundgerüst in WordPress ... 532
10.2.2 Der Kopfbereich mit den Menüs ... 536
10.2.3 Der Seitenbereich ... 541
10.2.4 Bonus: E-Mails ... 547

10.3 Projekt 3: Die Verwaltungsoberfläche für einen Onlineshop 550
10.4 Projekt 4: Entwicklung einer Progressive Web App (PWA) 565
10.4.1 Seitenstruktur und Layout 567
10.4.2 Funktionalität des Taschenrechners 569
10.4.3 Von der Internetseite zur Progressive Web App 573

Index **583**

Downloadhinweis

Alle Programmcodes aus diesem Buch sind zum Download verfügbar. Dadurch müssen Sie sie nicht abtippen:
https://bmu-verlag.de/books/html-css-kompendium/



Außerdem erhalten Sie die eBook-Ausgabe zum Buch im PDF Format kostenlos auf unserer Website:

https://bmu-verlag.de/books/html-css-kompendium/
Downloadcode: siehe Kapitel 10

Kapitel 1
Einführung

1.1 Über dieses Buch

Wohin man heutzutage schaut: Fast überall finden sich Displays mit Werbung, Abfahrtszeiten, Nachrichten... nicht zu vergessen Smartphones. Es gibt praktisch keinen Ort mehr, an dem man nicht Menschen über ihrem Smartphone gebeugt sieht, beschäftigt mit sozialen Netzwerken, Büchern, aktuellen Nachrichten oder den neuesten Apps.

Wo kauft man ein? Online.

Wo liest man "Zeitung"? Im Internet...

Doch was hat das mit diesem Buch zu tun? Die meisten der oben beschriebenen Anwendungen werden heutzutage mit Hilfe von HTML und CSS realisiert. Durch die sagenhafte Verbreitung von Computern, Smartphones, Tablets und anderen Anzeigegeräten aller Art sowie den immer schneller werdenden Netzwerkanbindungen (in den meisten Gegenden Deutschlands auch mobil) ist das Internet zu einem wesentlichen Teil des modernen Lebens geworden. Und HTML ist *der* Standard, der seinerzeit speziell für die Darstellung von Informationen im Internet beziehungsweise dem World Wide Web (WWW) entwickelt wurde und noch bis heute weiterentwickelt wird. Dank der Entwicklung von CSS haben die Internetseiten und Anwendungen von heute optisch auch nichts mehr mit den Ursprüngen gemein. Allerdings: Die Grundlagen sind nach wie vor dieselben. Zudem ist es einfach zu erlernen, was sicherlich ebenfalls zur raschen Verbreitung beigetragen hat.

Dieses Buch soll Sie in die Grundlagen von HTML und CSS einführen und wird durch kreative Beispiele auf Basis der vorher eingeführten Theorie die fast unendlichen Einsatzfelder demonstrieren. Ihrer Krea-

tivität sind quasi keine Grenzen gesetzt, dennoch sollen vier konkrete Projekte dazu dienen, Sie mit den wesentlichen Konzepten vertraut zu machen. Durch einen kleinen Ausflug in die Programmiersprache JavaScript werden Sie zudem im fortgeschrittenen Teil des Buches in die Lage versetzt werden, eine dynamische Webseite und eine eigene Smartphone(-Web)-App zu entwickeln. Bei allen Beispielen wird stillschweigend davon ausgegangen, dass Sie sich ohne besondere Vorkenntnisse beruflich mit dem Thema befassen möchten. Die Projekte sind daher so formuliert, als wären Sie angehender Webdesigner. Aber auch wenn Sie lediglich aus privatem Interesse zu diesem Buch gegriffen haben, sind die Beispiele geeignet. Stellen Sie sich in diesem Fall einfach vor, ein Freund, dem Sie von Ihren Ambitionen berichtet haben, hätte Sie um Unterstützung gebeten. Ziel des vorliegenden Buches ist es daher, Ihnen die Grundlagen von HTML und CSS nach Maßgabe der aktuellen Möglichkeiten so praxisnah wie möglich beizubringen. Beispiele dienen jeweils der Veranschaulichung der Grundlagen, Übungsaufgaben helfen beim Vertiefen des Verständnisses.

1.2 Was ist HTML und wozu dient es?

Die Abkürzung HTML steht für Hypertext Markup Language, also übersetzt Hypertext-Auszeichnungssprache. Hypertext wiederum ist Text, der mit anderem Text über "Hyperlinks" (ja, die 1970er...) verknüpft ist. In der Konsequenz entsteht ein ganzes Netz verknüpfter beziehungsweise verlinkter Informationen: das Internet.

HTML dient also dazu, die Elemente von Hypertext zu gliedern und zu formatieren. Dazu bedient sich HTML sogenannter Tags, also den eigentlichen Auszeichnern, dazu später mehr. In diesem Sinne ist HTML also in seiner ursprünglichen Form keine Programmiersprache, da es nicht möglich ist, mit HTML Programme zu entwickeln, die beispielsweise Benutzereingaben verarbeiten, Berechnungen anstellen etc. Hierzu braucht man grundsätzlich eine Programmiersprache, von denen es im Bereich des Internets mittlerweile viele gibt. Die wohl populärsten sind JavaScript und PHP. JavaScript wird wie HTML und CSS im Browser ausgeführt, dem sogenannten Client. Demgegenüber stehen

Programmiersprachen, die auf dem Server ausgeführt werden, wie zum Beispiel PHP.

PHP-Programme laufen auf dem Server ab und liefern als Ausgabe in der Regel wieder HTML-Code, der dann dem Benutzer durch den Browser dargestellt wird. Eingehender werden wir das Thema im Abschnitt 8 über Dynamische Webseiten besprechen. Wie bereits erwähnt, werden Programme zum Öffnen von HTML-Dateien als Internetbrowser oder kurz Browser bezeichnet. Hiervon gibt es heutzutage eine größere Zahl, was für den Ersteller von Internetseiten beziehungsweise den Webdesigner des Öfteren zur Herausforderung werden kann. Auch dazu später mehr.

Der Browser interpretiert die jeweiligen HTML-Tags und gibt den Quellcode (idealerweise) als ansehnlichen Text aus.

1.3 CSS und HTML

Wie wir später noch sehen werden, bietet HTML über die Strukturierung von Texten in Abschnitte, Überschriften, Absätze, Listen etc. hinaus die Möglichkeit, Medien wie Bilder, Videos oder auch andere Internetseiten in den Hypertext einzubinden. Erst durch die Kombination mit CSS wurde dies alles jedoch massentauglich, da erst CSS die Möglichkeit bot, die HTML-Elemente „aufzuhübschen“ und an die verschiedenen Ausgabegeräte anzupassen, das sogenannte responsive Webdesign.

CSS steht für Cascading Style Sheets, also übersetzt etwa kaskadierende Darstellungsvorgaben. Kaskadierend deshalb, weil die Vorgaben an die hierarchisch niedrigeren Elemente – wie wir später noch sehen – „vererbt“ werden. Mit CSS wird der Entwickler in die Lage versetzt, HTML-Elemente nahezu beliebig zu gestalten. Dies betrifft sowohl deren Farbe als auch Größe und Bildschirmposition. CSS bietet aber darüber hinaus auch die Möglichkeit, Elemente zu animieren oder ganz „verschwinden“ zu lassen. Aus modernen Webseiten ist CSS nicht mehr wegzudenken. Die Möglichkeiten von HTML und CSS werden stetig fortentwickelt, was dazu führt, dass ganz moderne HTML-Tags oder CSS-Kommandos nur

von den jeweils neuesten Browsern unterstützt werden. Man spricht daher sowohl bei HTML als auch bei CSS von einem living standard.

Eine weitere Eigenschaft von CSS ist, dass damit HTML-Code nicht nur für Bildschirmgeräte, sondern für viele Ausgabemedien, wie zum Beispiel Drucker, gestaltet werden kann.

Mithin erlaubt CSS die Trennung von Information und deren Darstellung, ein allgemeines Architekturmuster, nicht nur in der Informatik.

Alle Programmcodes aus diesem Buch sind zum Download verfügbar. Dadurch müssen Sie sie nicht abtippen:
https://bmu-verlag.de/books/html-css-kompendium/



Außerdem erhalten Sie die eBook-Ausgabe zum Buch im PDF Format kostenlos auf unserer Website:

https://bmu-verlag.de/books/html-css-kompendium/
Downloadcode: siehe Kapitel 10

Kapitel 2
Begriffe

Apache

Vollständig Apache HTTP Server. Ist einer der meistgenutzten Webserver im Internet. Die Software dient dazu, Anfragen von Benutzern (Seitenanfragen) zu verarbeiten und entsprechende Inhalte auszuliefern (Webseiten, Bilder etc.).

Browser

Auch Webbrowser. Spezielle Computerprogramme zur Darstellung von Internetseiten.

Client

Auch Client-Rechner. Bezeichnet das Endgerät oder die Software, das/die Daten von einem Server abruft.

Computernetzwerk

Ist ein Zusammenschluss mehrerer Rechner, die über definierte Protokolle miteinander Daten austauschen. Ein wichtiges Computernetzwerk ist das Internet.

Dynamische Website

Ist eine Internetseite, die vom Webserver oder vom Browser, basierend auf den Eingaben des Benutzers, generiert wird.

E-Mail

Briefähnliche Nachrichten, die über Computernetzwerke (insbesondere das Internet) verteilt werden.

FTP

Akronym für File Transfer Protocol also Dateiübertragungsprotokoll. Ein Protokoll zur Übertragung von Dateien zwischen Rechnern über das Internet.

Hypertext-Dateien

Textdateien, die durch sogenannte Hyperlinks miteinander verknüpft sind und dadurch eine netzähnliche Struktur schaffen.

Internet

Ist ein weltweiter Verbund von Computernetzwerken. Es ermöglicht unter anderem Dienste wie das World Wide Web, E-Mail, FTP und so weiter.

IP-Adresse

Adresse in einem Computernetzwerk, die wie das Internet auf dem Internet-Protokoll (IP) basiert. Die IP-Adressen erlauben eine eindeutige Zuordnung eines Rechners im Netzwerk (ähnlich einer Postanschrift). Es gibt aktuell die Versionen IPv4 und IPv6. Eine IP-Adresse der Version IPv4 hat die Form `192.0.2.23`.

Statische Website

Ist eine Website, die vorgefertigt vorliegt und unabhängig von Benutzereingaben vom Webserver ausgeliefert wird. Auch Bilder etc. sind statische Elemente.

Webserver

Ist eine Software, die Dokumente an Clients übertragen kann.

Website

Ist der über eine bestimmte Adresse abrufbare Satz von Dateien, wie zum Beispiel HTML-Dateien, Bildern, CSS-Dateien etc. Sie kann mit einem Browser wiedergegeben werden.

World Wide Web

Ein über das Internet abrufbares System von Hypertext-Dokumenten.

Kapitel 3

Erste Schritte

Wie eingangs beschrieben, wird für die Übersetzung von HTML-Code in eine ansehnliche Internetseite, eine Web-App oder sonstige oben erwähnten Spielarten ein Browser benötigt. Dieser ruft die HTML-Datei auf und interpretiert sie.

Dabei ist es egal, ob sich die Datei auf Ihrem lokalen Rechner oder in einem Computernetzwerk wie dem Internet befindet. Der Browser braucht lediglich die korrekte Adresse der Datei. Hierbei spielt – anders als beispielsweise bei E-Mail-Adressen – Groß- und Kleinschreibung eine wichtige Rolle. HTML-Dateien haben üblicherweise die Dateiendung **.html** oder **.htm**.

Die erste Seite einer Webseite, die Startseite, wird nach den gängigen Konventionen **index.html** genannt. Besteht die Website aus mehreren Seiten, benennt man diese typischerweise entsprechend ihrem Inhalt, also beispielsweise **kontakt.html** für die Kontaktseite oder **impressum.html** für das Impressum der Website.

Allerdings sind die Namen nicht in Stein gemeißelt. Diese kann der Webdesigner selbst festlegen, wobei Sonder- und Leerzeichen sowie Umlaute aus Kompatibilitätsgründen vermieden werden sollten. Nur bei der Startseite ist man gut beraten, sich an die gängigen Konventionen zu halten, da Webserver oft entsprechend vorkonfiguriert sind. Siehe dazu auch Abschnitt 9.5.

Die Adresse einer solchen Datei ist dann abhängig vom Speicherort der Datei. Diese Adresse wird auch als Uniform Resource Identifier (URI) bezeichnet, der stets nach demselben Muster aufgebaut ist:

schema://authority/pfad[?abfrage][#fragment]

Bei lokal, also auf Ihrem Rechner, gespeicherten Dateien lautet die Adresse zum Beispiel

file://Users/Klaus/Desktop/index.html (beim Mac),

beziehungsweise

file://C:/Benutzer/Klaus/Desktop/index.html (Windows).

Das Schema des URI wäre in diesem Fall **file**, bezeichnet also eine lokal gespeicherte Datei, gefolgt vom Pfad, unter dem die Datei gespeichert ist. Dies kann wie im Beispiel ein lokaler Pfad auf dem Rechner sein, aber auch ein beliebiger Pfad in einem Netzwerk.

Typische Schemata bei Webseiten sind **http** beziehungsweise **https**, wobei das **s** für SSL, also eine Verschlüsselung der Datenübertragung, steht. Ein kompletter URI im Internet könnte also beispielsweise

https://www.html-css-kompendium.de/verzeichnis/index.html

lauten. Dieser beinhaltet nach der Domain (also nach URI-Schema der Authority) auch den Pfad zur Datei, in diesem Fall **/verzeichnis/index.html**.

Diese Pfadangaben sind das wesentliche Element der Hyperlinks, also werden die Querverweise im Folgenden nur noch als Links bezeichnet. Sie bestimmen, wohin ein Link führen soll. Verlinkungen funktionieren mit dem kompletten URI (dann spricht man auch von einem absoluten Link) oder dem (relativen) Pfad alleine. In letzterem Fall spricht man von einem relativen Link. Dieser geht stets von der aktuellen Datei aus.

Betrachten wir die folgende Verzeichnisstruktur:

- link.html
- verzeichnis
  - index.html

Diese ist sehr einfach und beinhaltet nur zwei HTML-Dateien: **link.html** im Stammverzeichnis und **index.html** im Verzeichnis **verzeichnis**. Relativ zur Datei **link.html** befindet sich die Datei **index.html** also hier: **verzeichnis/index.html**, während umgekehrt der relative Pfad von der **index.html** zur link.html entsprechend **../link.html** lauten würde.

Dabei steht das **../** für die jeweils höhere Verzeichnisebene. Auch Konstruktionen wie **../kontakt/kontakt.html** wären denkbar.

Übungsaufgabe: Ergänzen Sie die in der dargestellte Verzeichnisstruktur, wenn der Pfad **../kontakt/kontakt.html** relativ zur Datei `index.html` gültig sein soll.

Lösung:

Die Verzeichnisstruktur müsste wie folgt aussehen:

- link.html
- verzeichnis
  - index.html
- kontakt
  - kontakt.html

Auf die weiteren URI-Bestandteile, die **query** und das **fragment**, kommen wir später noch mal zurück.

3.1 Einrichtung der Arbeitsumgebung

HTML und CSS sind sehr genügsam, was die Entwicklungsumgebung angeht. Sie benötigen lediglich einen (oder idealerweise mehrere) Browser, die bei den meisten Computern ohnehin vorhanden sind, und einen Texteditor. Allerdings ist es gerade für Anfänger ratsam, einen Editor zu wählen, der die HTML (und CSS) Syntax versteht, also die Tags

und Befehle kennt, und diese entsprechend hervorhebt (das sogenannte Syntax-Highlighting). Dies erleichtert die Fehlersuche und die Erstellung eines sauberen Codes. Viele Editoren bieten zudem eine Autoergänzung der Befehle – ein tolles Feature, da man bei der Vielzahl der Kommandos schnell den Überblick verlieren kann. Zudem ist es auch für die (später anstehende) Entwicklung größerer Projekte mit mehreren Dateien besser, sich direkt als Anfänger mit einer professionelleren Entwicklungsumgebung zu befassen.

Da HTML-Code unmittelbar vom Browser interpretiert wird, ist im Prinzip auch kein Webserver erforderlich. Salopp gesprochen genügt es, die HTML-Datei auf dem Schreibtisch zu speichern und per Klick mit einem Browser zu öffnen. Allerdings ist es für die Entwicklung hilfreich, sich direkt an die spätere Verwendung des Codes an eine realistische Webserverumgebung zu gewöhnen. Insbesondere kann so das Verhalten realistischer URLs, aber auch die Einbindung von Medien analog zum späteren Verhalten auf dem Webserver simuliert werden, und das erleichtert wiederum die Fehlerbehebung in der späteren Produktionsumgebung.

3.1.1 Lokaler Webserver

Server nutzen zur Auslieferung der statischen und dynamisch generierten Dateien eine bestimmte Webserver-Software. Diese Software wird oft mit dem Webserver synonym gesetzt, was technisch jedoch nicht ganz korrekt ist, da der eigentliche Webserver ja „nur“ ein Computer mit Internetanschluss ist. Eines der am weitesten verbreiteten Webserver-Programme ist der Apache HTTP Server.

Die manuelle Installation, aber insbesondere die Konfiguration eines Apache HTTP Server ist vor allem für Neueinsteiger recht aufwändig. Glücklicherweise gibt es seit mehreren Jahren eine selbstinstallierende und vorkonfigurierte Version, die zusätzlich auch noch mit weiterer Software ausgestattet ist, die üblicherweise auf einem Webserver läuft (Programmiersprachen wie PHP oder Perl, Datenbank wie MySQL oder

MariaDB). Dieses Softwarepaket ist unter dem Namen XAMPP bekannt und steht für die verschiedenen Betriebssysteme unter

https://www.apachefriends.org/de/download.html

zum Herunterladen zur Verfügung.

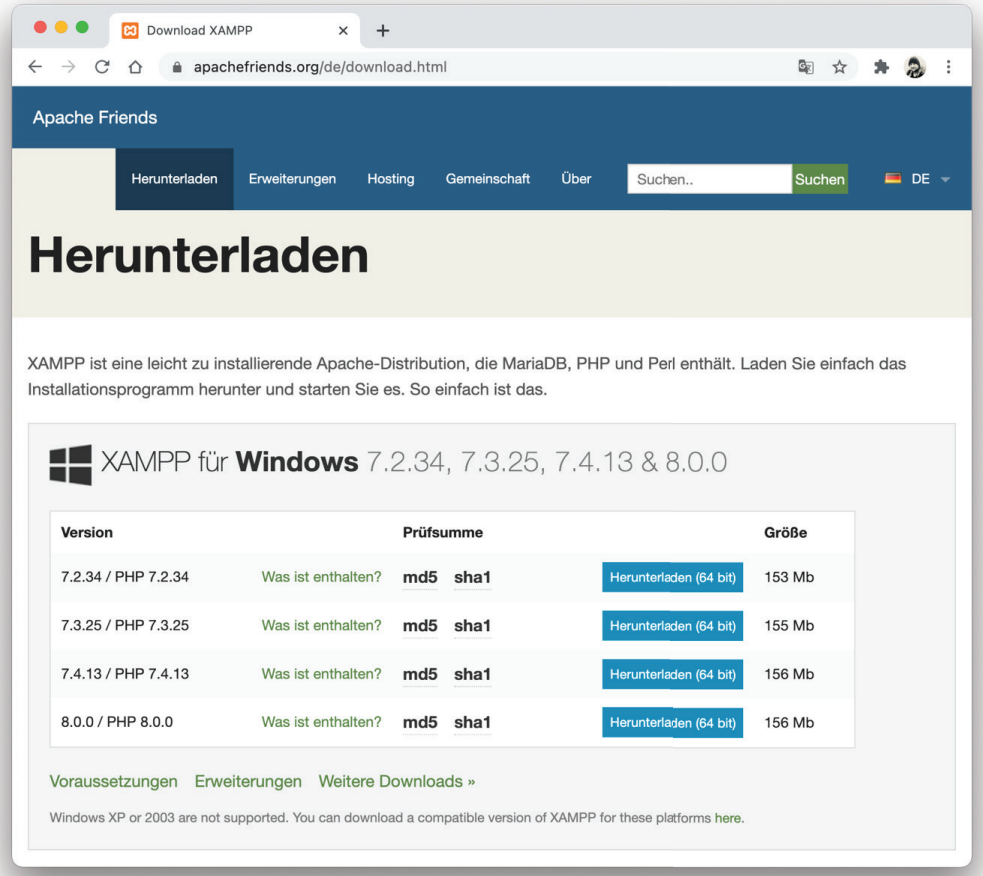


Abb. 3.1 Download-Seite des XAMPP-Pakets

Die Downloadseite des XAMPP-Pakets ist in Abb. 3.1 dargestellt. Auf dieser Seite finden sich die entsprechenden Installationspakete für die Betriebssysteme Windows, Linux und MacOS.

Es werden verschiedene Versionen angeboten, die sich im Wesentlichen in den unterschiedlichen Versionen der mitgelieferten Software (PHP, MariaDB, siehe oben) unterscheiden. Da dies für HTML und CSS

irrelevant ist, laden Sie einfach die aktuellste Fassung für Ihr Betriebssystem herunter und starten Sie den Installationsprozess (die Beispiele in diesem Buch sind mit der Version 8.2 getestet worden).

Sie benötigen dazu Administratorrechte auf dem Computer, mit dem Sie arbeiten.

3.1.1.1 Installation von XAMPP für Windows

Der Installationsprozess für Windows umfasst mehrere Schritte, die im Folgenden besprochen werden sollen.

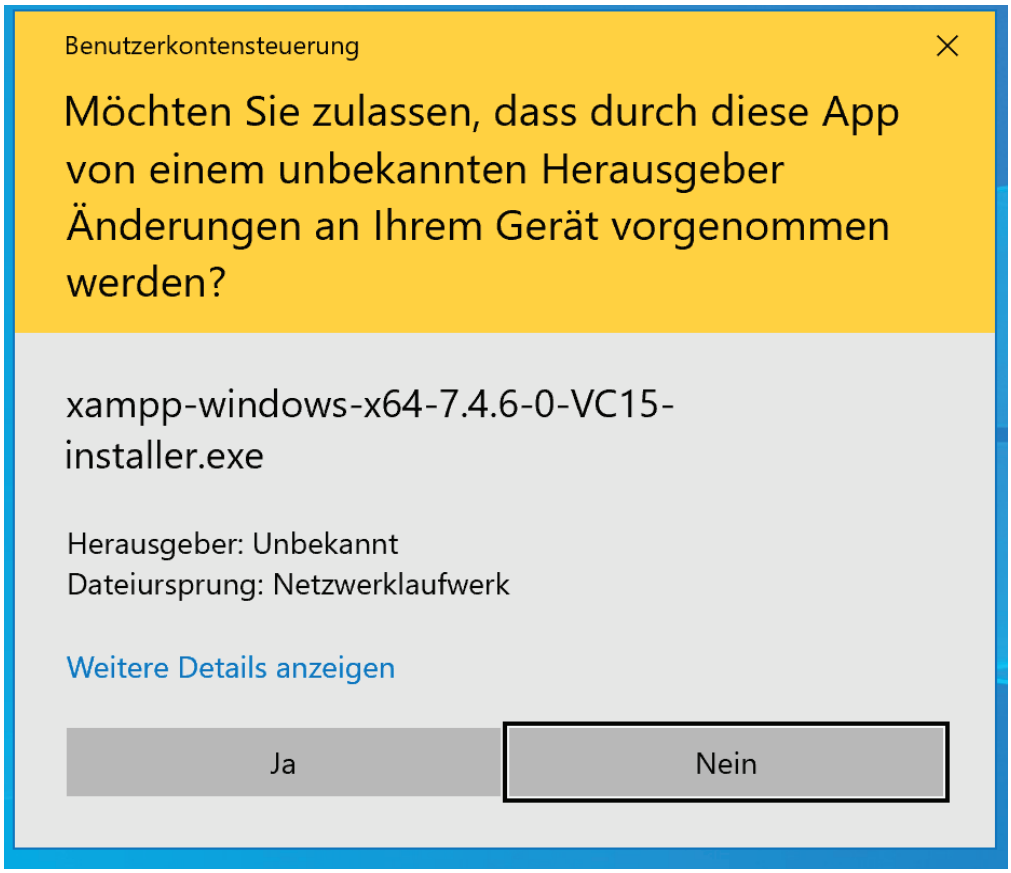


Abb. 3.2 Sicherheitswarnung bei der Installation von XAMPP

Zu Beginn wird Ihnen möglicherweise die in Abb. 3.2 dargestellte Warnung eingeblendet. Bestätigen Sie diese mit **Ja**, um die Installation zu starten.

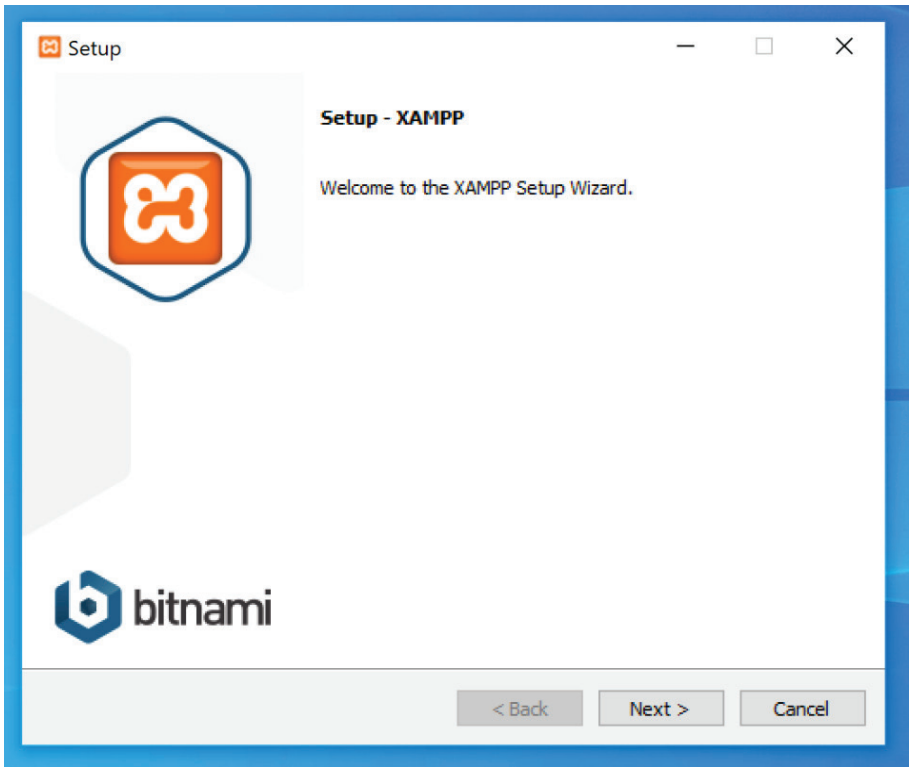


Abb. 3.3 Start der Installation XAMPP für Windows

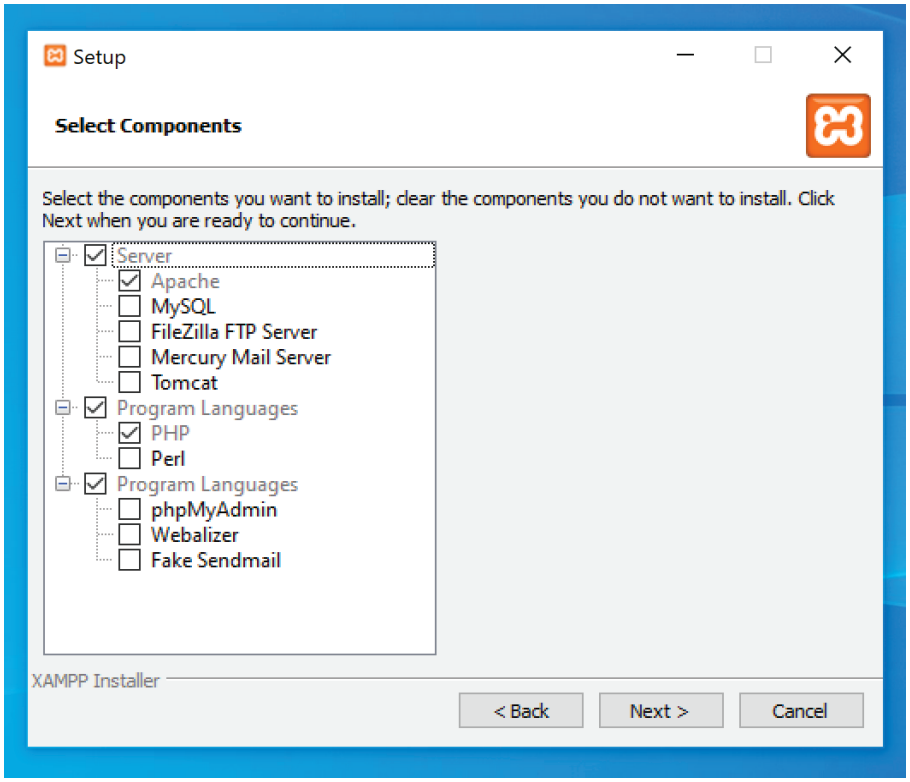


Abb. 3.4 Auswahl der zu installierenden Pakete (XAMPP für Windows)

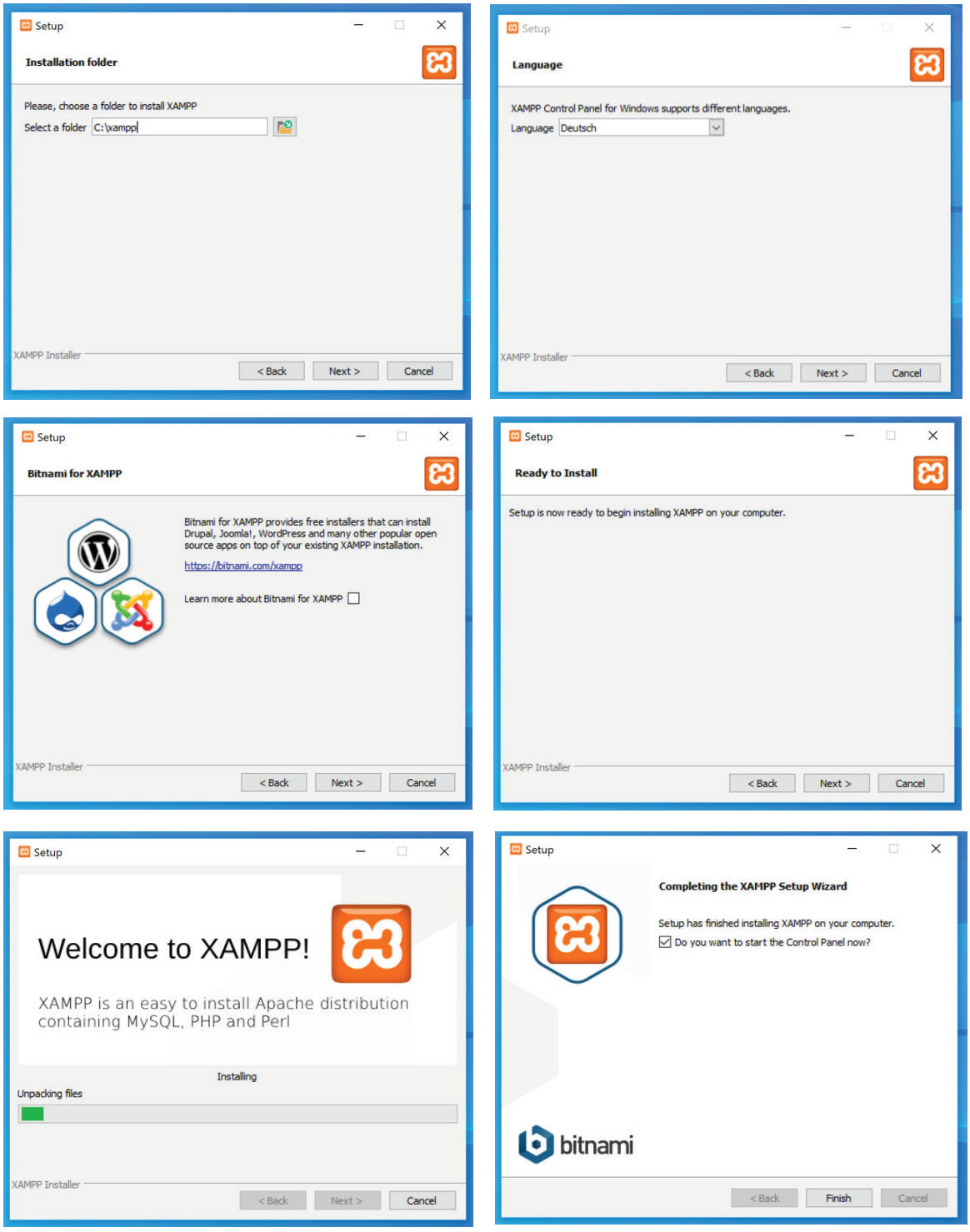


Abb. 3.5 Schritte der Installation (XAMPP für Windows)

Im Wesentlichen können bei den einzelnen Schritten die voreingestellten Werte genutzt werden. Wählen Sie **Next >** im Dialogfeld, welches in Abb. 3.3 dargestellt ist.

Im Auswahlmenü des in Abb. 3.4 wiedergegebenen Dialogfelds können Sie alle Haken entfernen. Für die Übungen in diesem Buch benötigen Sie lediglich den Apache HTTP Server. Sofern Sie später weitere Pakete benötigen sollten, da Sie beispielsweise in die Datenbankerstellung eingestiegen sind, lassen sich diese noch nachträglich installieren.

Bei den in Abbildung Abb. 3.5 dargestellten Dialogfeldern klicken Sie erneut jeweils **Next >** beziehungsweise **Finish** beim letzten Feld. Insbesondere sollten Sie den Pfad zum XAMPP-Verzeichnis beibehalten, da die weiteren Beschreibungen im Buch von dieser Standardkonfiguration ausgehen. Wenn wie im Screenshot der Haken bei **Do you want to start the control panel now?** gesetzt ist, startet XAMPP mit der Kontrolloberfläche aus Abb. 3.6.

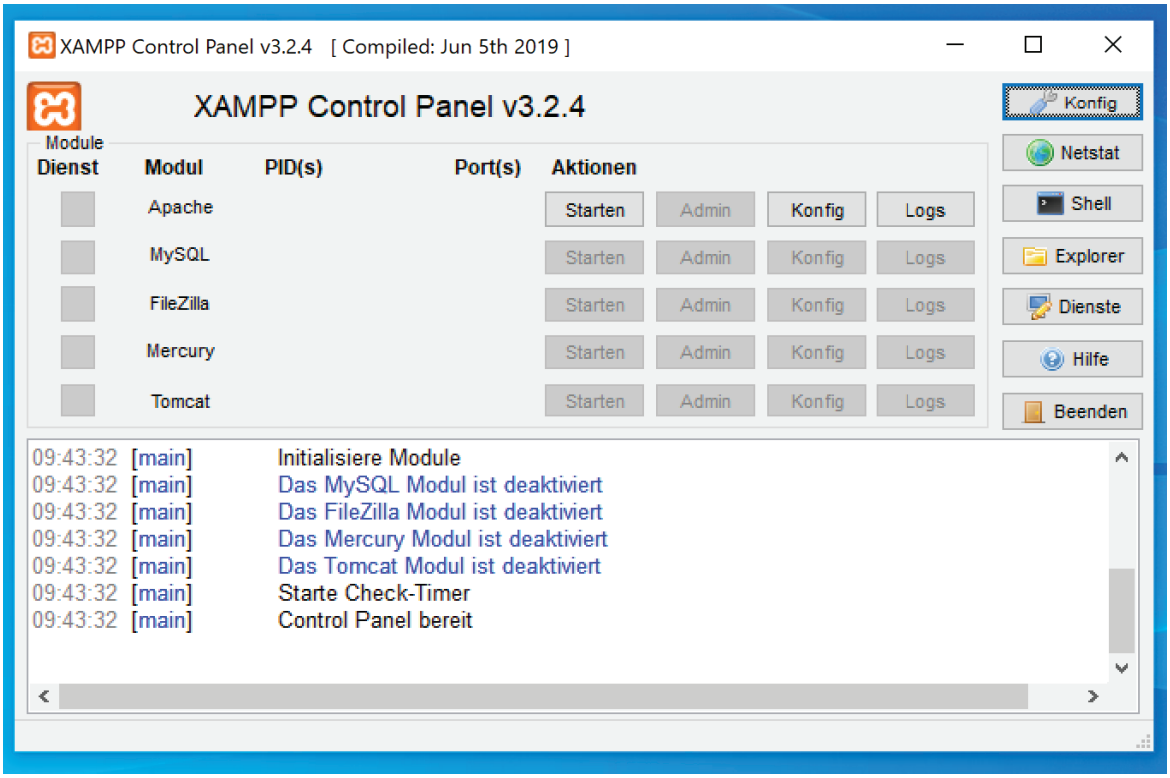


Abb. 3.6 XAMPP-Kontrolloberfläche der Windows-Version

Da bei der Installation nur Apache und PHP ausgewählt wurden, finden sich in der Konsole im unteren Teil der Kontrolloberfläche Meldungen wie **Das MySQL Modul ist deaktiviert** etc.

Die weitere Verwendung von XAMPP wird weiter unten eingehend erklärt. Lassen Sie das Kontrollpanel geöffnet, minimieren Sie es gegebenenfalls.

Installation von XAMPP für MacOS stehen jeweils zwei XAMPP-Versionen zur Verfügung, wie in Abb. 3.7 gezeigt ist.

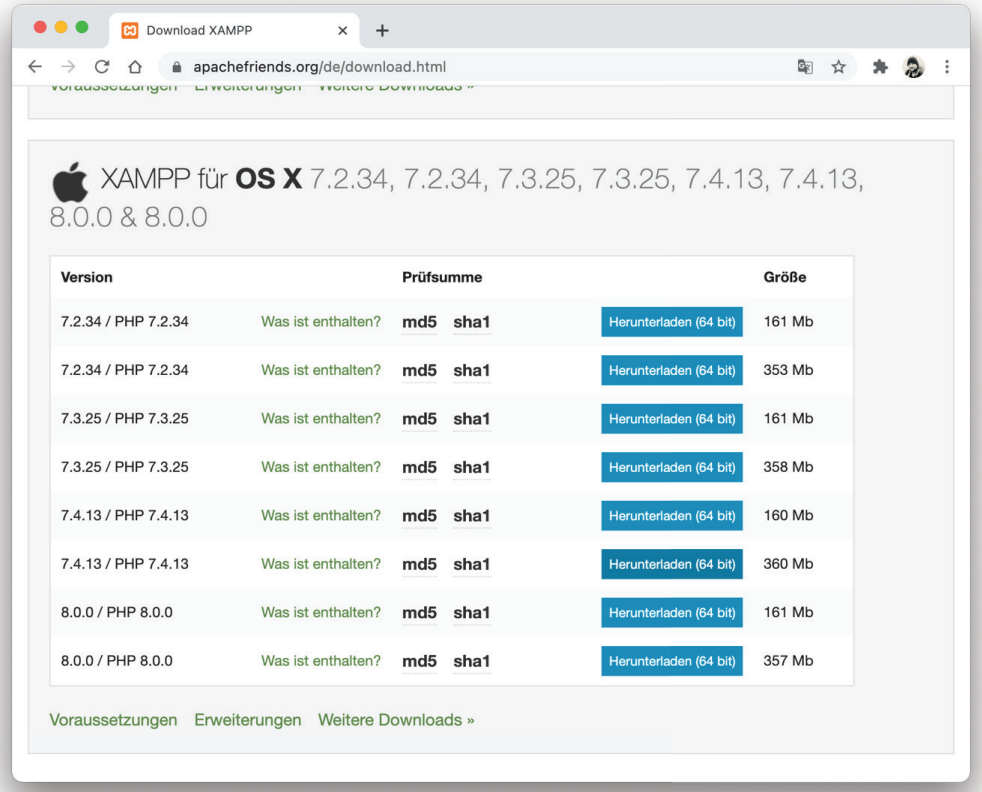


Abb. 3.7 Verschiedene Versionen von XAMPP für macOS

Es handelt sich dabei um die „normale“ Version von XAMPP, welche in den Programmen installiert ist, und um eine Variante, die mit einer sogenannten virtuellen Maschine installiert wird, die eine Linux-Umgebung auf dem Mac simuliert. Die Diskussion der Unterschiede würde uns vom wesentlichen Inhalt des Buches ablenken. Somit entscheiden

wir uns für die „normale“ Version, deren Dateiname die Form **xampp-osx-8.2.0-0-installer.dmg** hat.

Um die Installation für MacOS zu starten, öffnen Sie das heruntergeladene Image (dmg-Datei). Es öffnet sich ein Dialogfenster, welches in Abb. 3.8 wiedergegeben ist, gefolgt vom eigentlichen Installationsdialog. Öffnen Sie das Installationsprogramm, in dem Sie bei gedrückter Control-Taste das Icon anklicken und Öffnen wählen (vergleiche Abb. 3.9). Nach erneutem Klicken auf Öffnen im Warndialog (rechts in Abb. 3.9) startet der Installationsprozess. Wählen Sie Standardvorgaben während der Installation; die Installationsschritte sind in Abb. 3.10 dargestellt.

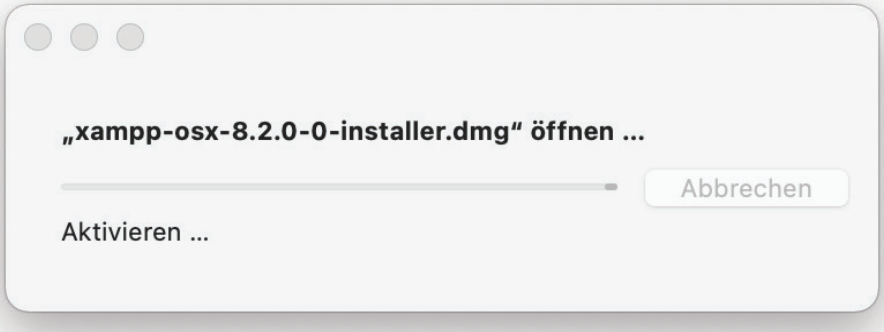


Abb. 3.8 Installationsvorbereitung (XAMPP für macOS)

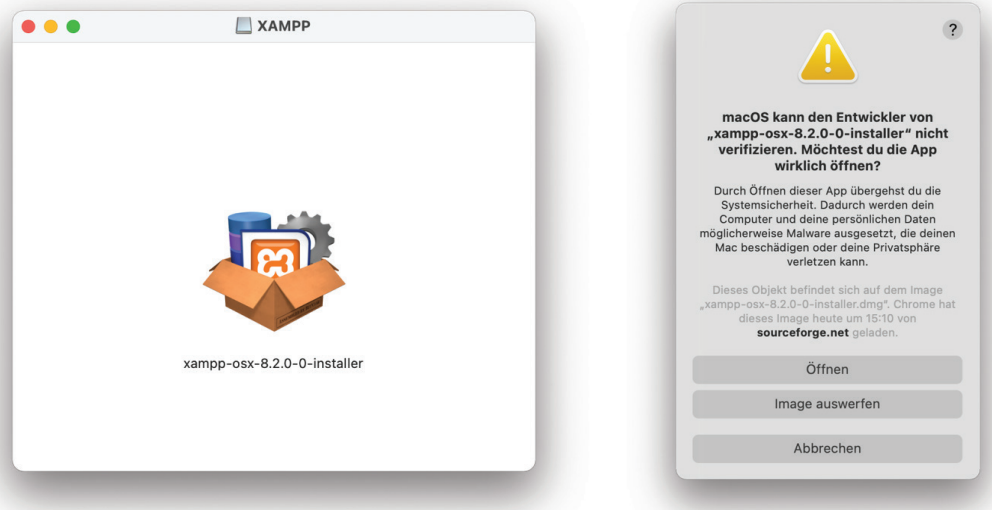


Abb. 3.9 Installationsdialog von XAMPP für macOS nach Control-Click > Öffnen

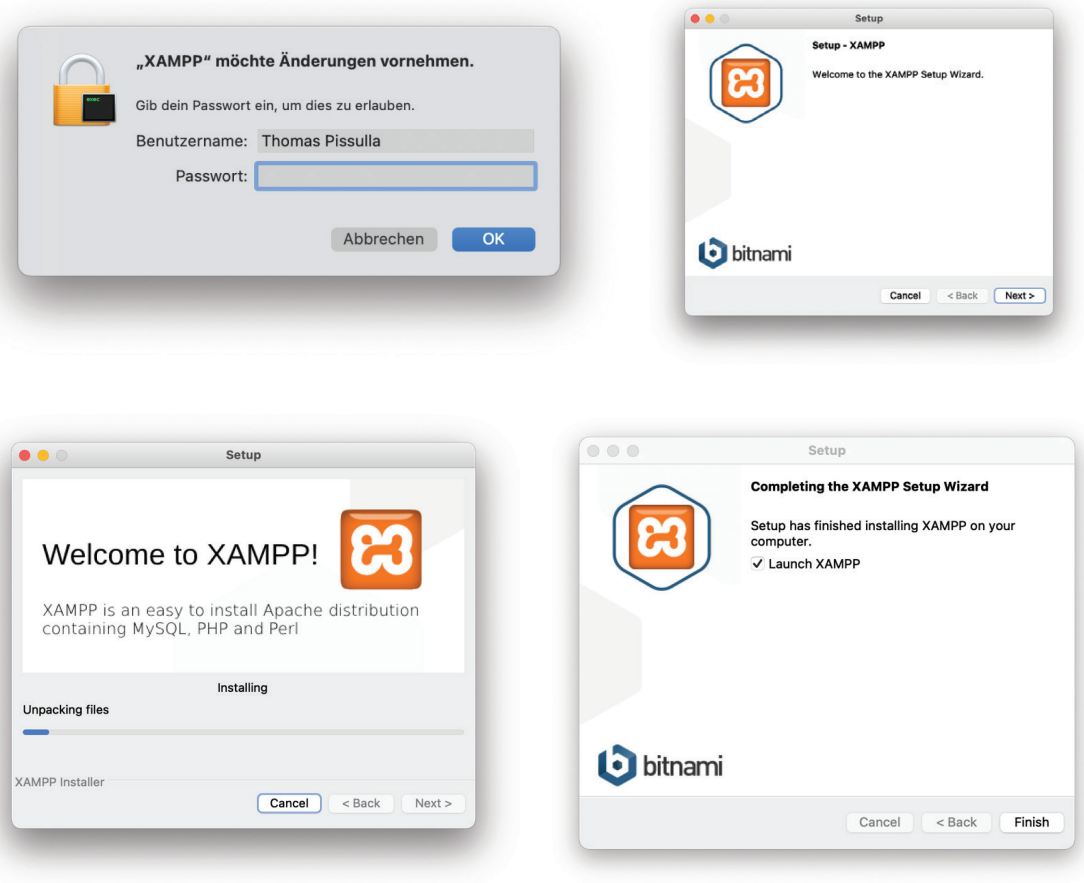


Abb. 3.10 Installationsschritte von XAMPP für MacOS

Nach Klick auf „Finish“ startet XAMPP automatisch. Alternativ können Sie den Finder nutzen, um in den Programmordner zu gelangen und XAMPP zu starten. Möglicherweise weist ein Dialog darauf hin, dass XAMPP für den erstmaligen Gebrauch initialisiert werden muss, bevor die Kontrolloberfläche in Abb. 3.11 angezeigt wird.



Abb. 3.11 Kontrolloberfläche (XAMPP für macOS)

Die weitere Verwendung von XAMPP wird weiter unten eingehend erklärt. Lassen Sie das Kontrollpanel geöffnet, minimieren Sie es gegebenenfalls.

3.1.2 Editoren

Die Wahl des Editors ist letztlich reine Geschmackssache. Es gibt eine Vielzahl kostenloser und kostenpflichtiger Editoren auf dem Markt.

Für die Beispiele im vorliegenden Buch wird der kostenlose Editor **Visual Studio Code** genutzt. Dieser Editor verfügt in seiner Basisausstattung über alle notwendigen Funktionen. Zudem ist er durch Pakete nahezu beliebig erweiterbar. Der Editor ist über folgenden Link zu beziehen:

https://code.visualstudio.com/

Nach dem Öffnen dieser Seite (vergleiche Abb. 3.12) sollte automatisch der Downloadlink für das von Ihnen verwendete Betriebssystem angezeigt werden. Laden Sie die dort vorgeschlagene Fassung herunter.

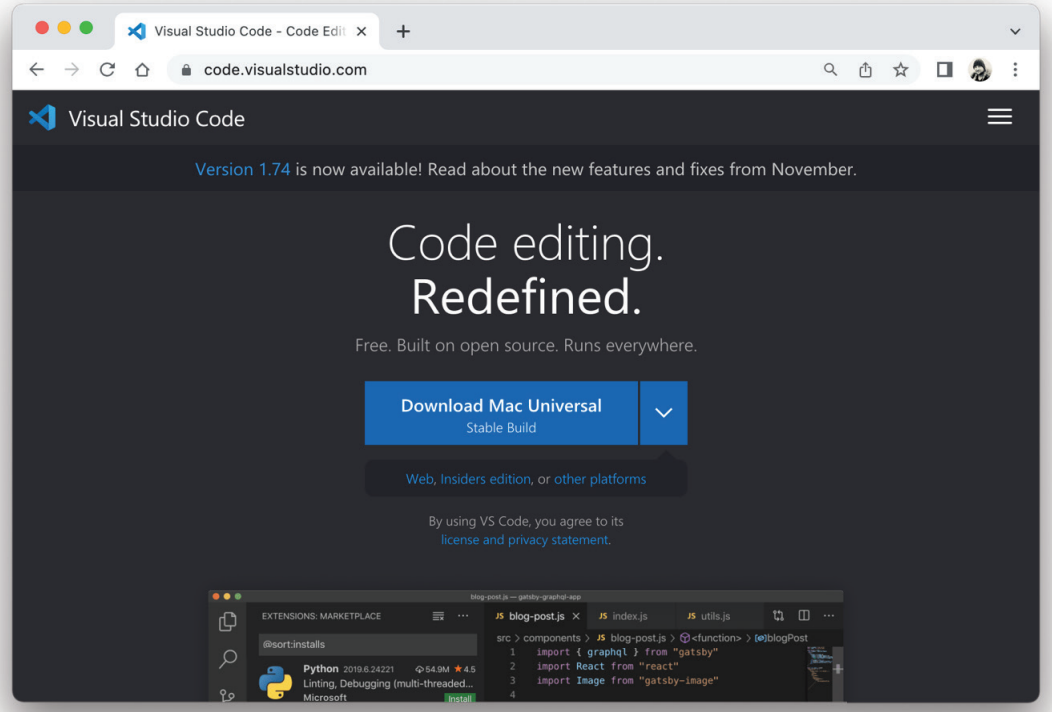


Abb. 3.12 Downloadseite des Editors Visual Studio Code

Für die Windowsversion wird der Nutzer nach Starten des Installationsprogramms durch die Installation geführt, wie in Abb. 3.13 angezeigt. Nach Abschluss der Installation steht der Editor Visual Studio Code wie jedes andere Programm zur Verfügung.

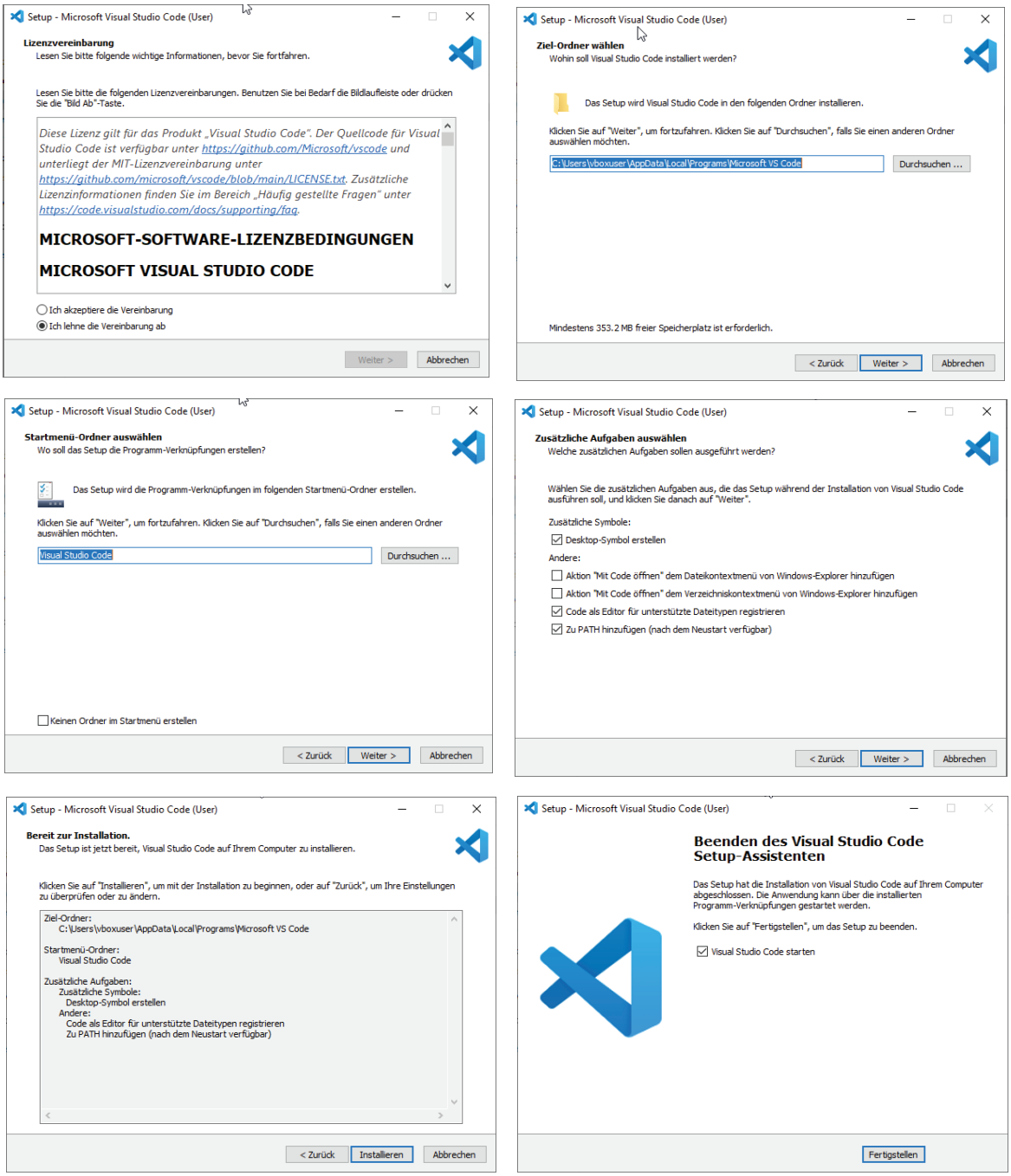


Abb. 3.13 Installationsdialog des Programms Visual Studio Code für Windows

Die macOS-Version von Visual Studio Code wird lediglich als ZIP-Archiv (vergleiche Abb. 3.14) angeboten. Nach Entpacken des Archivs liegt der ausführbare Editor im selben Ordner. Verschieben Sie diese Datei in den Programmordner, um sie zu installieren.

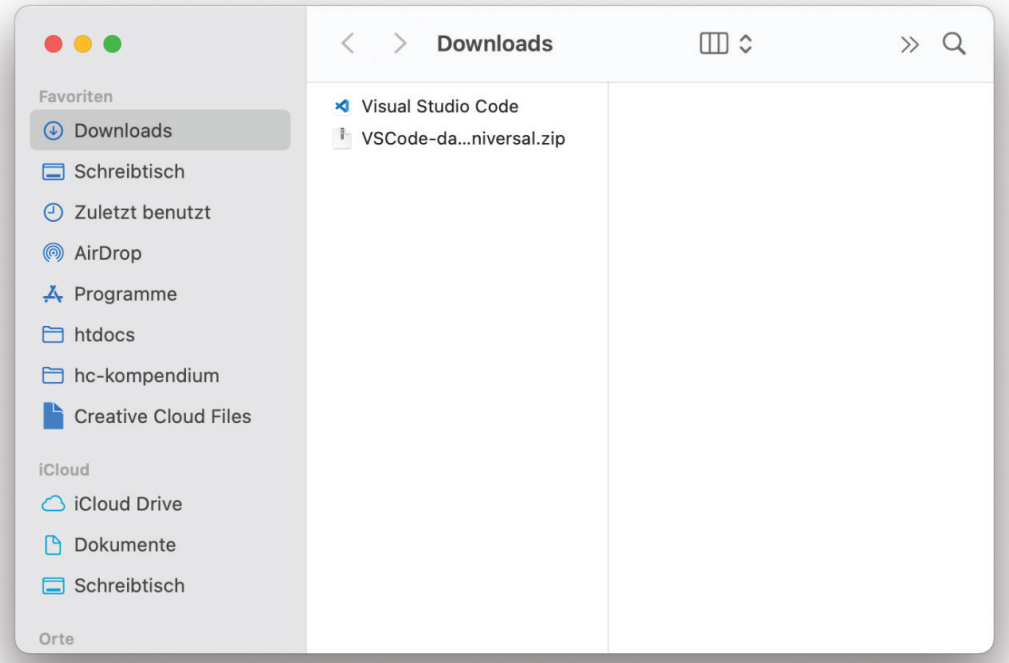


Abb. 3.14 Editor vor der Installation bei macOS

Wir werden später mit dem Editor weiterarbeiten, für den wir im Weiteren den offiziellen Kurztitel VS Code nutzen werden.

3.1.3 Browser

Der letzte, aber fast wichtigste Bestandteil der Entwicklungsumgebung sind die Browser. Zum Zeitpunkt der Drucklegung dieses Buches decken die nachfolgend aufgeführten Browser über 95% der Markanteile in Deutschland ab (Quelle statista.de):

Desktop

- Google Chrome
- Safari
- Firefox
- Edge

- Internet Explorer
- Opera

Hierbei bilden allein Chrome, Safari und Firefox deutlich über 80% der Nutzer ab.

Bei Smartphones und Tablets, also den mobilen Endgeräten, ist die Zahl der marktführenden Browser sogar noch deutlich geringer. Hier bilden folgende Browser über 95% der Marktanteile in Deutschland ab:

Mobil

- Google Chrome
- (Mobile) Safari
- Samsung Internet

Behalten Sie im Hinterkopf, dass die fertigen Seiten auf allen diesen Browsern ansehnlich beziehungsweise auf den älteren Modellen zumindest funktional sein sollten.

Ohne auf die Details, beispielsweise die jeweiligen verschiedenen Versionen, eingehen zu wollen, wird allein durch diese Auflistung klar, dass zumindest die jeweils ersten drei Browser zur Entwicklungsumgebung gehören sollten.

Wie eingangs bereits erwähnt, handelt es sich bei HTML und CSS um einen living standard. Insbesondere bei der Interpretation von CSS, aber auch – wie wir später noch sehen werden – bei der Umsetzung von JavaScript verhalten sich die verschiedenen Browser unterschiedlich. Es ist daher essentiell, die von Ihnen programmierten Seiten stets in den jeweiligen Browsern ausgiebig zu testen.

Safari und Internet Explorer stellen Entwickler regelmäßig vor Herausforderungen, da Safari nur für macOS beziehungsweise die mobile Variante iOS zur Verfügung steht und der Internet Explorer nur für Windows. Edge wiederum – der neueste Microsoft-Browser – steht auch für macOS zum Download zur Verfügung.

Chrome ist plattformunabhängig verfügbar und bildet somit eine sehr gute Basis Ihrer Entwicklungsumgebung.

Moderne Browser können nicht nur für die Darstellung der fertigen Webseite genutzt werden, sondern liefern über die jeweiligen Entwicklermenüs auch wertvolle Tools zur Analyse und zur Fehlersuche. Alle Browser haben hier mehr oder weniger vergleichbare Möglichkeiten. Im folgenden Text wird der Einfachheit halber ausschließlich die Entwicklerkonsole von Chrome zur Illustration des Geschriebenen genutzt. Wie beim Editor gilt: Es ist letztlich Geschmackssache und Sie sollten ein wenig herumprobieren, um den Browser zu finden, der am besten zu Ihrem Stil passt. Beachten Sie jedoch den jeweiligen Marktanteil – immerhin soll Ihre Seite von möglichst vielen Nutzern perfekt dargestellt und genutzt werden können.

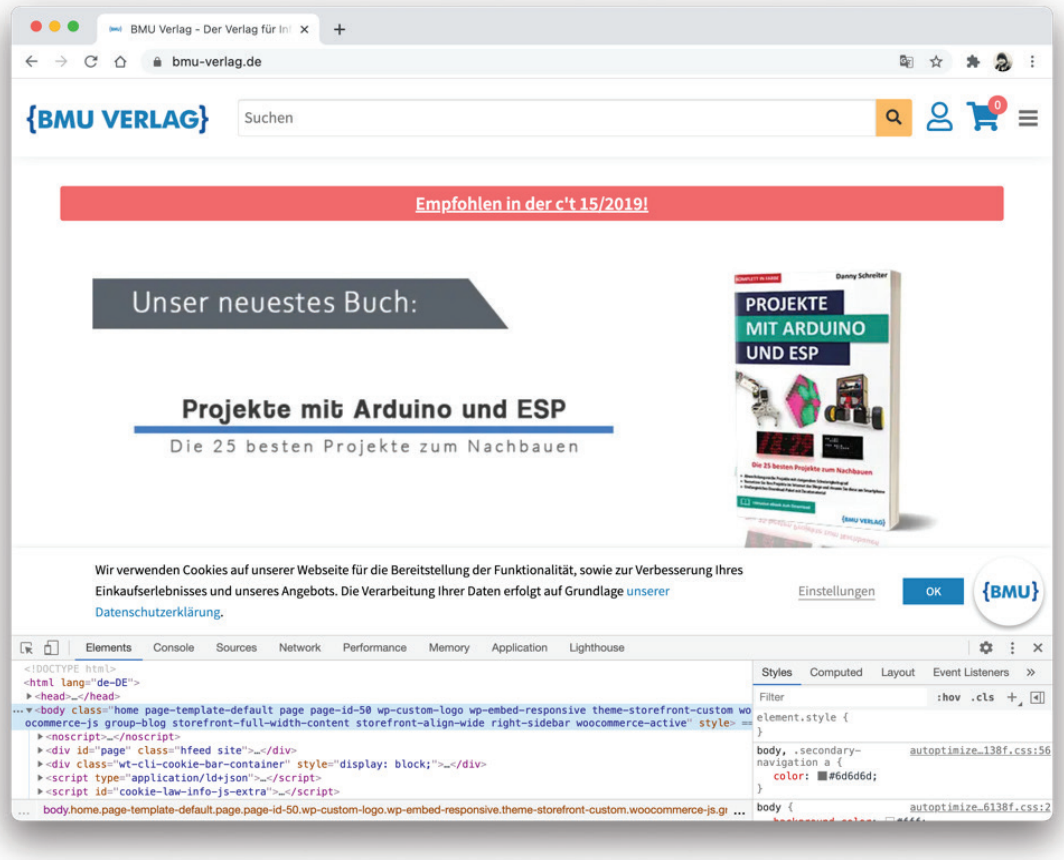


Abb. 3.15 Der Browser Chrome mit eingeblendeter Entwicklerkonsole (unterer Bereich)

In Abb. 3.15 ist ein Screenshot der eingeblendeten Entwicklertools von Google Chrome zu sehen. Eine Erklärung der verschiedenen Nutzungsszenarien erfolgt bei den jeweiligen Beispielen im weiteren Text.