PHP und MySQL für Einsteiger

Michael Bonacina

3. Auflage: Oktober 2019

© dieser Ausgabe 2019 by BMU Media GmbH

ISBN: 978-3-96645-009-6

Herausgegeben durch:
BMU Media GmbH
Hornissenweg 4
84034 Landshut

PHP und MySQL für Einsteiger

Inhaltsverzeichnis

1. Einleitung 9

1.1 PHP – was ist das überhaupt? ... 10
1.2 Statische und dynamische Webseiten ... 11
1.3 Anwendungsmöglichkeiten für PHP ... 12
1.4 SQL-Datenbanken für Internetanwendungen ... 13

2. Vorbereitungsmaßnahmen für das Programmieren mit PHP 16

2.1 HTML: eine wichtige Grundlage für Programme in PHP ... 16
2.2 Webserver-Software für die Ausführung eines PHP-Programms ... 18
2.3 Einen passenden Texteditor installieren ... 20

3. Das erste Programm mit PHP gestalten 22

3.1 PHP-Scripte im Text kenntlich machen ... 22
3.2 Das erste Programm schreiben ... 23
3.3 Das Programm zum Laufen bringen ... 25
3.4 Übung: ein einfaches Programm in PHP schreiben ... 27

4. PHP und HTML miteinander verbinden 31

4.1 HTML-Code mit PHP ausgeben ... 31
4.2 PHP-Scripte in eine HTML-Seite integrieren ... 36

5. Variablen: ein wichtiges Element der Programmierung mit PHP 42

5.1 Text mit einer Variablen erfassen ... 43
5.2 Zahlen als Variable abspeichern ... 45
5.3 Boolesche Variablen ... 46
5.4 Operationen mit Variablen ... 47

5.5 Arrays: zusammengesetzte Variablen 51
5.6 Übung: Umgang mit Variablen 56

6. Entscheidungen durch if-Abfragen 60

6.1 Der Aufbau einer if-Abfrage 60
6.2 Verschiedene Vergleichsoperatoren verwenden 61
6.3 Logische Operatoren in die Abfrage integrieren 63
6.4 else und elseif 66
6.5 Übung: Abfragen selbst erstellen 68

7. Die Funktionalität eines Programms durch Schleifen erweitern 72

7.1 Kopfgesteuerte Schleifen: while und for 72
7.2 Fußgesteuerte Schleife: do while 77
7.3 foreach-Schleifen für die Arbeit mit Arrays 79
7.4 Übung: Programme mit Schleifen gestalten 84

8. Funktionen in PHP 89

8.1 Der Aufbau einer Funktion in PHP 90
8.2 Rückgabewerte der Funktionen 93
8.3 Eine Funktion in das PHP-Programm einbinden 96
8.4 Funktionen aus der PHP-Bibliothek verwenden 97
8.5 Übung: PHP-Programme mit Funktionen erstellen 99

9. Objektorientierte Programmierung: Klassen, Objekte und Methoden 105

9.1 Die Klasse: Grundlage der objektorientierten Programmierung 106
9.2 Mit einer Klasse ein Objekt erzeugen 108
9.3 Methoden für die Arbeit mit Objekten 110
9.4 Übung: Klassen, Objekte und Methoden anwenden 112

10. Dateien für die Speicherung von Daten 117

10.1 Daten aus einer Datei einlesen 117

10.2 Daten in einer Datei speichern 120
10.3 Die Dateirechte beachten 123
10.4 Übung: Dateien für die Datenspeicherung verwenden 124

11. Datenbanken: Die effiziente Alternative für die Datenspeicherung 128

11.1 Was ist eine Datenbank? 129
11.2 Der Aufbau von Datenbanken 130
11.3 MySQL: wichtiges Verwaltungssystem für Datenbanken 132
11.4 SQL: die Datenbanksprache für die Arbeit mit MySQL 134

12. Grundlegende Operationen für den Umgang mit MySQL-Datenbanken 136

12.1 DDL, DCL und DML: drei Arten von SQL-Befehlen 136
12.2 Datenbanken und Tabellen anlegen 137
12.3 Daten eintragen 142
12.4 Daten ändern 144
12.5 Daten löschen 145
12.6 Zugriffsrechte verwalten 146
12.7 Übung: Eine Tabelle in MySQL anlegen 147

13. Weitere wichtige SQL-Funktionen 152

13.1 Teilmengen der Datensätze auswählen 152
13.2 Anzahl der Einträge feststellen 155
13.3 In Zellen gespeicherte Werte addieren 156
13.3 Einfache mathematische Operationen mit SQL ausführen 158
13.4 Übung: Informationen aus einem Warenbestand abrufen 159

14. SQL-Datenbank in PHP einbinden 161

14.1 Das PHP-Programm mit einer Datenbank verbinden 162
14.2 Die Datenbank füllen 165
14.3 Einträge löschen oder ändern 169
14.4 Daten abfragen 170
14.5 Übung: Datenbanken mit PHP bearbeiten 173

15. Anwendungsbeispiel: Kundendaten über das Internet erfassen 177

15.1 Kundendaten über ein mit PHP generiertes Formularfeld abfragen 178
15.2 Daten im Objekt Kunde ablegen 182
15.3 Werte an Datenbank übermitteln 189
15.4 Kundendaten mit PHP-Programm anzeigen lassen 192

16. PHP und Cookies: Besucher wiedererkennen und Informationen speichern 201

16.1 Was ist ein Cookie? 202
16.2 Ein Cookie setzen 203
16.3 Informationen aus dem Cookie auswerten 207
16.4 Session Cookies 210
16.5 Anwendungsbeispiel: Eine Webseite mit Cookies gestalten 213
16.6 Übungsaufgabe: Cookies selbst erstellen 227

17. Fortgeschrittene PHP-Funktionen: E-Mails automatisch versenden und PDFs erzeugen 231

17.1 Textmails mit PHP versenden 231
17.2 Anhänge versenden 236
17.3 PDFs mit PHP erzeugen 239
17.4 Anwendungsbeispiel: Eine PDF-Rechnung erstellen und als Anhang versenden 243
17.5 Übungsaufgabe: Mails verschicken und PDFs gestalten 250

18. Für mehr Sicherheit: SQL-Injection vermeiden 256

18.1 Was ist SQL-Injection? 256
18.2 Prepared Statements: So wird die Eingabe weiterer Befehle verhindert 259
18.3 Anwendungsbeispiel: Das Eingabeformular für die Kundendaten vor SQL-Injection schützen 263
18.4 Übungsaufgabe: Ein sicheres Formular erstellen 270

Alle Programmcodes aus diesem Buch sind als PDF zum Download verfügbar. Dadurch müssen Sie sie nicht abtippen:
https://bmu-verlag.de/books/php-mysql/

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

https://bmu-verlag.de/books/php-mysql/
Downloadcode: siehe Kapitel 18

Kapitel 1
Einleitung

PHP und SQL sind zwei wichtige Elemente vieler Internet-Anwendungen. PHP zählt zu den am häufigsten für die Erstellung von Webseiten genutzten Programmiersprachen. Sie bietet sich hervorragend für die Erstellung dynamischer Inhalte an. Diese lassen es zu, die Seiten individuell an den Nutzer oder an den Kontext anzupassen.

Die Daten, die für die Erstellung der Inhalte zum Einsatz kommen, werden dabei in der Regel in einer Datenbank gespeichert. Diese bietet ein sehr effizientes und gut strukturiertes System für die Datenspeicherung. Um eine Datenbank einzurichten, einzelne Daten darin zu speichern oder um sie abzurufen, kommt eine Datenbanksprache zum Einsatz. Für Webanwendungen ist dabei die Sprache SQL besonders beliebt. Die Inhalte unzähliger Webanwendungen sind in SQL-Datenbanken abgespeichert.

PHP zeichnet sich unter anderem durch eine breite Datenbankunterstützung aus. Insbesondere für SQL-Datenbanken stehen unzählige Funktionen zur Verfügung. Gerade dieses gute Zusammenspiel zwischen Programmiersprache und Datenbankanbindung macht PHP zu einem sehr mächtigen Werkzeug für viele Webanwendungen. Das führte dazu, dass PHP und SQL im Bereich der Gestaltung von Internetseiten eine herausragende Rolle spielen.

Dieses Buch gibt eine Einführung in die Nutzung von PHP und SQL. Die Leser lernen dabei zunächst, einfache Programme in PHP zu schreiben. Diese werden Stück für Stück um wichtige Funktionen erweitert. Später wird die Funktionsweise von SQL-Datenbanken erläutert und die Leser erfahren, auf welche Weise sich diese in ein PHP-Programm einbinden lassen. Am Schluss steht ein Anwendungsbeispiel, das das Zusammenspiel von PHP und SQL deutlich macht.

1.1 PHP – was ist das überhaupt?

Die erste Version von PHP wurde 1995 vorgestellt. Der dänische Programmierer Rasmus Lerdorf hatte hierfür einige Scripte in der Programmiersprache Perl erstellt. Der Ausgangspunkt für diese Arbeit bestand darin, dass er die Zugriffe auf seinen Online-Lebenslauf protokollieren wollte. Daraus entstand schließlich eine umfangreiche Scriptsammlung, die Lerdorf für die Gestaltung seiner persönlichen Homepage nutzte. Daher gab er ihr den Namen Personal Home Page Tools – PHP.

Von diesen Ursprüngen ist außer dem Namen jedoch nicht mehr viel übrig geblieben. Bereits für die zweite Version, die nur kurze Zeit später erschien, überarbeitete Lerdorf das Konzept stark. Anstatt in Perl programmierte er diese Version nun in C. Bis heute wurden alle weiteren Versionen ebenfalls in C entwickelt.

1997 änderte sich das Entwicklerteam. Lerdorf wirkte zwar nach wie vor an diesem Projekt mit, die Hauptverantwortung übernahmen nun jedoch die beiden israelischen Programmierer Andi Gutmans und Zeev Suraski. Das neue Entwicklerteam wies auch der Abkürzung PHP eine neue Bedeutung zu. Diese steht seitdem für PHP: Hypertext Preprocessor. Damit wollten die Programmierer verhindern, dass der Eindruck entsteht, dass sich die Programmiersprache nur für persönliche Projekte eignet.

Der große Durchbruch gelang der Version PHP 4. Zu dieser Zeit wuchs das World Wide Web in einem beachtlichen Tempo und für die Gestaltung dynamischer Inhalte war eine geeignete Programmiersprache erforderlich. Da PHP genau auf diese Aufgabe spezialisiert ist, setzte es sich in diesem Bereich als Standard durch und ist seitdem die Grundlage unzähliger Internetauftritte.

Mit der darauffolgenden Version machten die Entwickler PHP zu einer objektorientierten Programmiersprache. Was das genau bedeutet, wird später im Buch noch ausführlicher erläutert. An dieser Stelle sei nur erwähnt, dass es sich hierbei um eine der Grundlagen moderner Pro-

grammiertechniken handelt. Auch diese Neuerung trug dazu bei, dass PHP seine Vormachtstellung im Bereich der Programmierung von Web-Anwendungen ausbauen konnte.

Nachdem die Entwicklung der Version 6 eingestellt wurde, erschien im Dezember 2016 PHP 7. Diese Version brachte zahlreiche technische Optimierungen mit sich, sodass sie deutlich schneller arbeitet und außerdem weniger Speicherplatz benötigt.

PHP ist auf fast allen Webservern vorinstalliert. Daher eignet sich diese Sprache hervorragend für serverseitig programmierte dynamische Webseiten. Eine Studie kam zu dem Ergebnis, dass im Jahre 2017 rund 83 Prozent aller Internetauftritte diese Technik verwendeten (W3Techs: abgerufen unter https://w3techs.com/technologies/overview/programming_language/all am 14.03.2018). Dieser Wert zeigt deutlich, welche Bedeutung PHP für die Entwicklung des Internets hat.

1.2 Statische und dynamische Webseiten

In den vorherigen Abschnitten wurde bereits erwähnt, dass PHP zur Erstellung dynamischer Webseiten dient. Um eine genauere Vorstellung davon zu bekommen, was mit PHP alles möglich ist, soll nun erklärt werden, was es damit auf sich hat.

Als das World Wide Web 1989 entstand, kamen hierfür einfache HTML-Seiten zum Einsatz. Dabei handelt es sich um eine recht einfache Auszeichnungssprache, die den Inhalten einer Internetseite eine Struktur verleiht. Diese Inhalte bleiben jedoch – solange keine manuelle Anpassung vorgenommen wird – stets gleich. Aus diesem Grund werden reine HTML-Seiten als statisch bezeichnet.

Als sich das Internet jedoch weiterentwickelte, reichten diese statischen Seiten nicht mehr aus. Bei vielen Anwendungen war es notwendig, den Inhalt, der dabei ausgegeben wird, immer wieder an den Nutzer oder an die aktuellen Bedingungen anzupassen. Ein einfaches Beispiel hierfür ist die Wiedergabe von E-Mails in einem Webbrowser. Wenn hier

immer nur die gleichen Seiten angezeigt werden, ist keine sinnvolle Kommunikation möglich. Bei einem Online-Shop ist es ebenfalls erforderlich, die Darstellung an die vom Kunden ausgewählten Artikel anzupassen. Auch für Nachrichtenportale, die ihre Meldungen im Abstand von wenigen Minuten erneuern, wäre es sehr aufwendig, dafür jedes Mal eine neue HTML-Seite zu erstellen. Daher kommen hierbei Programme zum Einsatz, mit denen die Administratoren die Inhalte schnell und einfach ändern können. Mit PHP ist es möglich, all diese Funktionen zu programmieren.

Bei der Gestaltung dynamischer Webseiten ist es möglich, diese clientseitig oder serverseitig zu programmieren. Bei clientseitigen Anwendungen liefert der Server den kompletten Programmcode aus. Anschließend führt der Browser diesen aus. Damit ist es beispielsweise möglich, die aktuelle Uhrzeit in die Seite einzufügen, dem Anwender einen Rechner für eine spezifische Aufgabe bereitzustellen oder die Eingaben in ein Formular auf ihre Richtigkeit zu überprüfen. Für diese Aufgaben eignet sich beispielsweise Javascript. PHP ist jedoch eine serverseitige Sprache. Dabei führt der Server bereits das Programm aus. Das fertige Ergebnis liefert er dann dem Anwender aus. Dieses ist anschließend nicht mehr veränderbar. Diese Methode ist insbesondere dann sinnvoll, wenn für die Inhalte die Informationen aus einer Datenbank notwendig sind. Da diese in der Regel ebenfalls auf dem Server gespeichert ist, bietet sich in diesen Fällen eine serverseitige Umsetzung an.

1.3 Anwendungsmöglichkeiten für PHP

Da es sich bei PHP um die Programmiersprache handelt, die am häufigsten für die serverseitige Programmierung von Internetanwendungen zum Einsatz kommt, gibt es unzählige Programme, die diese nutzen. Dabei handelt es sich nicht nur um individuelle Software, die für eine ganz spezielle Web-Anwendung programmiert wurde und nur für diese zum Einsatz kommt. Darüber hinaus gibt es unzählige weitere Programme, die Funktionen bereitstellen, die für viele Websites sinnvoll sind. Die Nutzer können diese herunterladen und auf ihrem

Webserver installieren und ausführen. Viele davon sind unentgeltlich erhältlich, da es sich dabei um freie Software handelt. Bei anderen Produkten ist es hingegen erforderlich, eine Gebühr zu entrichten.

Sehr häufig kommt PHP beispielsweise für Blog-Software zum Einsatz. Beliebte Blog-Systeme wie Wordpress und Serendipity sind in dieser Sprache programmiert. Auch im Bereich der Content Management Systeme (CMS) ist PHP führend. Beispiele hierfür sind Joomla, TYPO3 und Drupal. Hinzu kommen Shopsysteme wie Magento, Foren-Software wie phpBB, Customer Relationship Management Systeme wie SugarCRM und viele weitere Programme.

Diese kleine Auflistung verdeutlicht nicht nur, wie viele Anwendungen in PHP geschrieben wurden. Darüber hinaus zeigen sie ein wichtiges Betätigungsfeld für PHP-Programmierer auf. In den meisten Fällen ist es erlaubt, den Quellencode an die eigenen Bedürfnisse anzupassen. Eine häufige Aufgabe besteht daher darin, kleine Details dieser Programme abzuändern, damit sie genau den Wünschen des Auftraggebers entsprechen. Außerdem ist es normalerweise möglich, den Funktionsumfang der Software durch kleine Zusatzprogramme – sogenannte Plug-ins – zu erweitern. Auch das stellt eine wichtige Aufgabe für PHP-Programmierer dar.

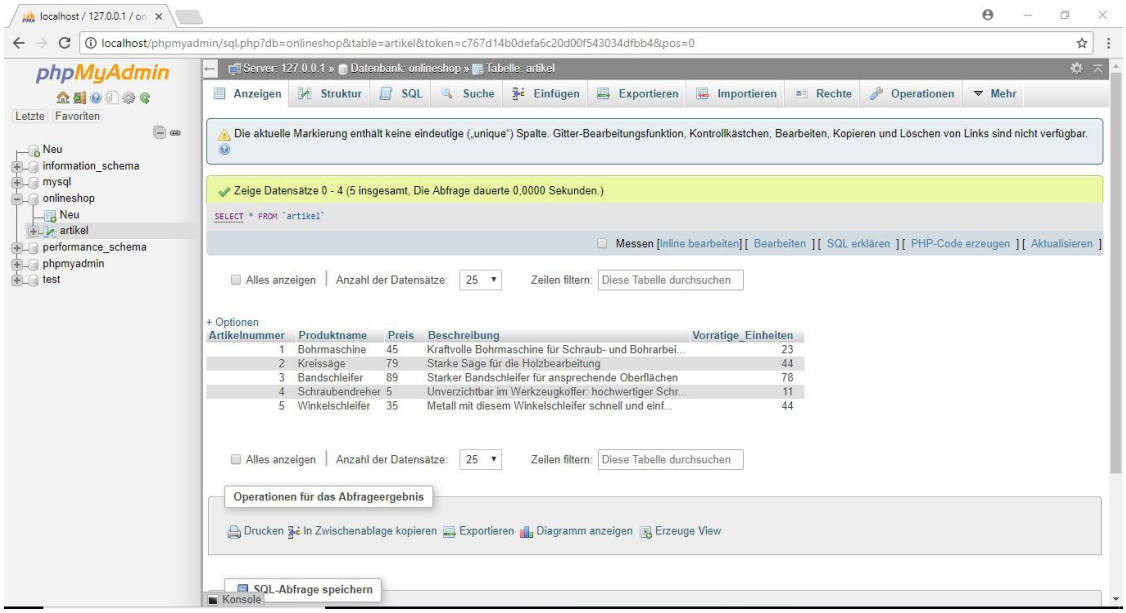
1.4 SQL-Datenbanken für Internetanwendungen

Dieses Buch behandelt neben der Programmiersprache PHP auch SQL-Datenbanken. Diese spielen eine sehr wichtige Rolle, um Daten für Internetanwendungen zu speichern. Aus diesem Grund ist es sinnvoll, kurz darauf einzugehen, was eine Datenbank ist, welche Anwendungsmöglichkeiten sie bietet und welche Rolle die Datenbanksprache SQL dabei spielt.

Eine Datenbank besteht immer aus zwei verschiedenen Teilen. Zum einen gibt es die eigentlichen Daten. Diese liegen auf einem entsprechenden Speicher und die Anwender können diese abrufen, verändern oder löschen. An dieser Stelle kommt das Datenbankmanagementsys-

tem (DBMS) ins Spiel. Hierbei handelt es sich um eine Software, die alle Aufgaben rund um die Verwaltung der entsprechenden Daten übernimmt.

Das DBMS muss die Daten strukturieren. Dafür gibt es unterschiedliche Modelle. Bei SQL-Datenbanken handelt es sich stets um relationale Datenbanken. Diese sind – vereinfacht gesagt – in Tabellen dargestellt. Wenn beispielsweise ein Online-Shop Produktdaten in einer Datenbank ablegt, dann sind für jeden einzelnen Artikel mehrere Einträge notwendig – die Artikelnummer, der Preis, eine Beschreibung, die Anzahl der vorrätigen Einheiten und einige weitere Informationen. Da sich all diese Daten auf das gleiche Produkt beziehen, besteht eine Verbindung zwischen ihnen. In der Tabelle wird das daran deutlich, dass sie in der gleichen Zeile stehen. Das relationale Datenbankmodell ist nicht die einzige Möglichkeit, um die Einträge zu strukturieren. In der Praxis kommt es jedoch bei fast allen Anwendungen zum Einsatz.



Screenshot 1 Beispiel für die tabellarische Gestaltung einer Datenbank

Das DBMS regelt außerdem den Zugriff auf die Daten. Es muss neue Tabellen erstellen, Einträge hinzufügen, löschen oder ändern. Auch bearbeitet es Abfragen, um den Nutzer über die Inhalte der einzelnen Felder zu informieren. Darüber hinaus muss es die Berechtigungen

kontrollieren. Bei jeder Anfrage ist es notwendig, zu überprüfen, ob der Anwender die entsprechende Aktion durchführen darf.

Um mit dem DBMS zu kommunizieren, kommt die Datenbanksprache SQL zum Einsatz. Offiziell handelt es sich hierbei um einen Eigennamen ohne weitere Bedeutung. Dennoch ist die Abkürzung auf den Begriff Structured Query Language – auf Deutsch strukturierte Abfragesprache –zurückzuführen. Daraus geht hervor, was die Hauptaufgabe von SQL ist: Daten aus der Datenbank abzufragen. Diese Sprache enthält alle notwendigen Befehle, um Daten anzufordern, einzufügen, zu löschen oder zu ändern. Es ist möglich, die SQL-Befehle direkt über die Benutzeroberfläche des DBMS einzugeben. In den meisten Fällen sind sie jedoch in ein PHP-Programm integriert, um auf diese Weise eine automatische Abfrage zu ermöglichen.

Kapitel 2

Vorbereitungsmaßnahmen für das Programmieren mit PHP

Das vorherige Kapitel hat die wichtigsten Themenbereiche aufgezeigt, mit denen sich dieses Buch beschäftigen wird. Nun ist es an der Zeit, mit der praktischen Arbeit zu beginnen und die ersten Programme in PHP zu schreiben. Allerdings sind hierfür zunächst einige Vorbereitungsmaßnahmen notwendig. Ohne diese ist es nicht möglich, mit dem Programmieren zu beginnen.

2.1 HTML: eine wichtige Grundlage für Programme in PHP

In der Einleitung wurde dargestellt, dass PHP für die Erstellung dynamischer Internetseiten zum Einsatz kommt. Statische Seiten verwenden hingegen HTML. Der folgende Abschnitt befasst sich nun mit genau dieser Auszeichnungssprache. Das stellt jedoch nur auf den ersten Blick einen Widerspruch dar. Denn fast jede Internetseite, die mit PHP erstellt wird, verwendet auch HTML-Code.

Der Zusammenhang besteht darin, dass die Ausgabe eines PHP-Programms normalerweise in HTML erfolgt. Wenn der Server das entsprechende Programm ausführt, erzeugt er eine gewöhnliche HTML-Seite. Diese gibt er daraufhin an den Browser des Nutzers weiter, der sie dann wie eine statische Seite anzeigt. Das dynamische Element liegt hierbei in der Ausführung auf dem Server, bei der individuell angepasste Seiten entstehen. Alle nachfolgenden Schritte sind genau gleich wie bei der Anzeige herkömmlicher Internetseiten.

Daraus geht hervor, dass jeder PHP-Programmierer über sehr gute HTML-Kenntnisse verfügen muss. Um ein Programm zu schreiben, dessen Ausgabe in HTML erfolgen soll, ist es selbstverständlich notwendig, die Strukturen und die Funktionen dieser Auszeichnungssprache

genau zu kennen. Darüber hinaus ist es üblich, nur einzelne Elemente einer Internetseite in PHP zu programmieren. Das bedeutet, dass zunächst eine gewöhnliche HTML-Seite entsteht. Lediglich an den Stellen, an denen dynamische Inhalte erforderlich sind, werden PHP-Scripte eingefügt, um die entsprechenden Informationen einzufügen.

Leser, die bislang über keine HTML-Kenntnisse verfügen, sollten sich diese daher vor der weiteren Lektüre unbedingt aneignen. In dieser Reihe ist auch zu diesem Thema ein Buch erschienen, das sich zu diesem Zweck hervorragend eignet. Dieses erklärt die Gestaltung von Internetseiten mit HTML von Grund auf. An dieser Stelle ist keine ausführliche Erklärung zu diesem Thema möglich. Dennoch sollen zur Wiederholung die wichtigsten Grundlagen kurz dargestellt werden.

Eine Internetseite besteht eigentlich aus gewöhnlichen Textbausteinen. Mithilfe von Tags werden diesen jedoch verschiedene Funktionen zugewiesen. Damit ist es beispielsweise möglich, Überschriften oder Absätze zu kennzeichnen oder bestimmte Textbereiche fett oder kursiv zu gestalten. Darüber hinaus eignen sich diese Tags für Listen und Tabellen.

Tags stehen immer in spitzen Klammern: <>. Darin befindet sich eine Buchstabenkombination, die die Funktion deutlich macht – beispielsweise `h1` für die Hauptüberschrift, `p` für einen Absatz oder `i` für Kursivschrift. Ein Beispiel für ein HTML-Tag wäre `<h1>`. Die meisten HTML-Tags müssen geöffnet und geschlossen werden. Für das Schlusstag kommt der Schrägstrich zum Einsatz. Um das oben genannte Beispiel zu schließen, wäre es demnach notwendig das Tag `</h1>` einzufügen. Das folgende Beispiel verdeutlicht den Einsatz der HTML-Tags:

```
<h1>Das ist die Überschrift</h1>
<p>Nun folgt ein kleiner Absatz. Dieser verwendet auch
<i>Kursivschrift</i>. Am Schluss ist es notwendig, den Absatz durch
ein entsprechendes Tag zu schließen.</p>
```

Die eigentlichen Seiteninhalte stehen im Body-Tag (`<body>`). Davor kommt in der Regel ein Head-Tag (`<head>`) zum Einsatz. Dieser enthält weiterführende Informationen zur verwendeten HTML-Version,

den Seitentitel, eine Kurzbeschreibung und weitere wichtige Informationen. Das gesamte Dokument steht in HTML-Tags (`<html>`). Die Struktur des HTML-Dokuments sieht demnach folgendermaßen aus:

```
<html>
<head>
Hier stehen weiterführende Informationen zum Dokument.
</head>
<body>
An dieser Stelle stehen die eigentlichen Seiteninhalte.
</body>
</html>
```

2.2 Webserver-Software für die Ausführung eines PHP-Programms

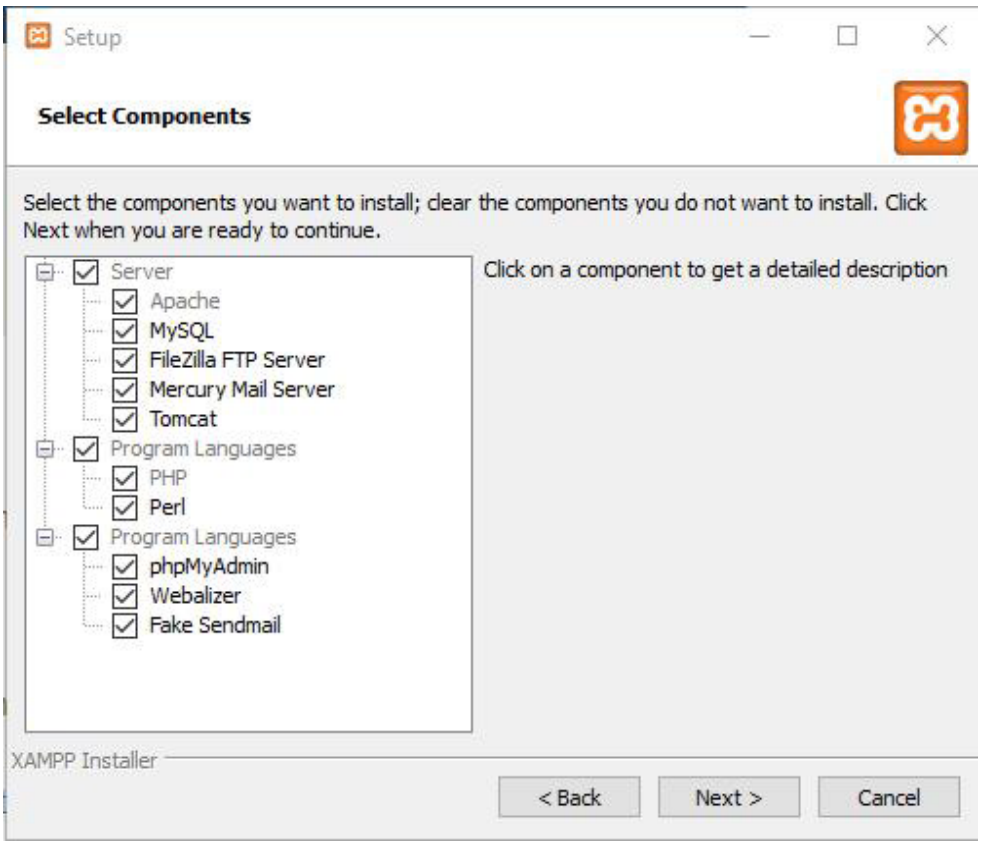
Bei PHP-Programmen handelt es sich um Scripte. Das bedeutet, dass der Code nicht kompiliert wird und daher nicht als ausführbare Datei vorliegt. Anstatt dessen ist ein Interpreter notwendig. Dieser liest das Programm, das als reiner Text vorliegt, ein und setzt die Funktionen um. Wie in den vorherigen Abschnitten bereits mehrfach erwähnt, werden PHP-Programme auf einem Webserver ausgeführt. Gängige Server-Software verfügt standardmäßig über einen PHP-Interpreter, sodass der Umgang mit diesen Programmen keinerlei Problem darstellt.

Viele Menschen, die gerade die ersten Schritte mit PHP unternehmen, haben jedoch keinen Zugriff auf einen Webserver. Aus diesem Grund ist hierfür eine andere Lösung notwendig. Es gibt frei verfügbare Programme, die einen Webserver auf dem heimischen PC installieren. Diese enthalten nicht nur einen Interpreter für PHP. Es ist damit auch möglich, Perl auszuführen. Darüber hinaus ist MySQL – eines der am häufigsten verwendeten Datenbankmanagementsysteme für Internetanwendungen – enthalten. Wenn später Datenbankanwendungen in die Programme eingefügt werden, ist die Ausführung daher ebenfalls problemlos möglich.

Die bekannteste Software in diesem Bereich trägt den Namen XAMPP. Dabei handelt es sich um freie Software, sodass für die Nutzung keinerlei Lizenzgebühren anfallen. Es gibt Versionen für Windows, Linux,

MacOS und einige weitere Betriebssysteme. Daher sollte es kein Problem darstellen, eine passende Version zu finden und diese zu installieren. Aktuelle Versionen für verschiedene Betriebssysteme stehen auf der offiziellen Website unter der Adresse https://www.apachefriends.org/de/index.html zum Download bereit.

Die Installation der Software ist ganz einfach. Der Installations-Assistent führt den Nutzer automatisch durch das Set-up. In einem der Installationsschritte fragt der Assistent den Nutzer, welche Funktionen installiert werden sollen.



Screenshot 2 Auswahlmöglichkeiten bei der Installation von XAMPP

Im Prinzip ist es möglich, die Standardeinstellungen zu übernehmen. Für dieses Buch werden jedoch lediglich die Funktionen Apache Webserver, PHP, MySQL und phpMyAdmin benötigt. Daher ist es auch möglich, auf die Installation der übrigen Funktionen zu verzichten, damit das Programm weniger Speicherplatz in Anspruch nimmt. Sobald XAMPP auf dem Rechner installiert ist, steht die notwendige Software

zur Verfügung, um die PHP-Programme, die im Laufe dieser Anleitung erstellt werden, auszuführen.

2.3 Einen passenden Texteditor installieren

Ein PHP-Programm besteht aus reinem Text. Um diesen zu schreiben und abzuspeichern, ist ein passendes Programm notwendig. Dabei ist es wichtig, zu beachten, dass die gängigen Textverarbeitungsprogramme wie Word oder LibreOffice Writer hierfür nicht geeignet sind. Diese speichern neben dem eigentlichen Text noch viele weitere Informationen zum Layout und zur Formatierung des Texts. Diese zusätzlichen Komponenten führen dazu, dass sich hiermit keine PHP-Programme erstellen lassen. Für diese Aufgabe ist ein Texteditor notwendig. Dieser speichert die Zeichen als reinen ASCII-Code ab. Auf diese Weise sind keine Zusatzinformationen enthalten, die die Ausführung des Programms stören.

Windows ist bereits mit einem Texteditor ausgestattet. Der Microsoft Editor – auch bekannt unter der Bezeichnung Notepad – bietet die notwendigen Funktionen. Allerdings handelt es sich hierbei um eine ausgesprochen einfache Ausführung, die sich nur für die ersten Programme eignet. Es gibt darüber hinaus viele weitere Editoren, die die Anwender beim Programmieren durch zahlreiche Zusatzfunktionen unterstützen. Diese kennzeichnen beispielsweise bestimmte funktionale Elemente des Programms durch unterschiedliche Farben oder sorgen für eine automatische Einrückung beim Erstellen von Funktionen. Das macht den Code deutlich übersichtlicher und erleichtert dadurch das Programmieren – insbesondere bei sehr komplexen Programmen. Manche Editoren machen außerdem auf Syntaxfehler aufmerksam. Dadurch verhindern sie Fehler bereits in der Entstehung.

Daher ist es empfehlenswert, einen Texteditor mit etwas höherem Funktionsumfang zu verwenden. Linux-Nutzer müssen sich darum in der Regel nicht weiter kümmern. Die meisten Linux-Distributionen sind bereits mit einer geeigneten Software ausgestattet. Ubuntu ver-

wendet beispielsweise gedit und KDE den Texteditor Kate. Sollte bislang kein passendes Programm installiert sein, steht eine große Auswahl an kostenfreier Software zur Auswahl. Die Seite https://de.wikipedia.org/wiki/Liste_von_Texteditoren gibt einen Überblick über die verschiedenen Möglichkeiten.

Im Prinzip ist es möglich, jeden der aufgeführten Texteditoren zu verwenden – unter der Voraussetzung, dass er für das verwendete Betriebssystem verfügbar ist und dass er die Programmierung in PHP unterstützt. An dieser Stelle soll exemplarisch das Programm Geany vorgestellt werden. Dieser Editor eignet sich hervorragend für PHP und außerdem sind Versionen für alle gängigen Betriebssysteme verfügbar.

Dieses Programm steht unter https://www.geany.org/Download/Releases zum Download bereit. Dabei ist es lediglich notwendig, die Version für das verwendete Betriebssystem herunterzuladen und anschließend den Installations-Assistenten auszuführen.

foreach3.php | array.php | array2.php | aufgabe7-2.php

```php
<?php

$sortiment = array();

$produkt[0]['Produktname'] = "Bohrmaschine";
$produkt[0]['Preis'] = 45;
$produkt[0]['Anzahl'] = 6;

$produkt[1]['Produktname'] = "Kreissäge";
$produkt[1]['Preis'] = 79;
$produkt[1]['Anzahl'] = 0;

$produkt[2]['Produktname'] = "Bandschleifer";
$produkt[2]['Preis'] = 89;
$produkt[2]['Anzahl'] = 15;

foreach ($produkt as $ebene1)
{
    foreach ($ebene1 as $feldname => $ebene2)
    {
        print $feldname.": ".$ebene2."<br>\n";
    }
    print "<br>";
}

?>
```

Screenshot 3 Farbige Hervorhebungen der Programmelemente im Texteditor Geany

Kapitel 3

Das erste Programm mit PHP gestalten

Nachdem alle Vorbereitungsmaßnahmen abgeschlossen sind, ist es nun an der Zeit, das erste eigene Programm zu schreiben. Bei dieser ersten Anwendung kommen selbstverständlich ganz einfache Strukturen zum Einsatz, um den Einstieg zu erleichtern. Dieses Kapitel dient dazu, die grundsätzliche Gestaltung eines PHP-Programms kennenzulernen und ein Gefühl für die Erstellung zu bekommen. Außerdem lernen die Leser dabei die wesentlichen Bestandteile der Syntax eines PHP-Programms kennen.

3.1 PHP-Scripte im Text kenntlich machen

Wenn ein Anwender mit einem Browser ein PHP-Dokument aufruft, muss der Webserver zunächst das entsprechende PHP-Programm ausführen, um die gewünschte Ausgabe zu erstellen. Dafür ist es jedoch notwendig, dass er den Text als PHP-Programm erkennt. Nur so ist eine sachgemäße Umsetzung möglich. Um dies dem Webserver zu vermitteln, ist zum einen eine passende Dateiendung notwendig. PHP-Programme enden normalerweise auf .php. Darüber hinaus gibt es noch weitere Möglichkeiten – beispielsweise .php3 und .phtml. Zwar unterstützen viele Webserver auch diese Endungen. Da sie nicht mehr gebräuchlich sind, ist es jedoch ratsam, sie nicht mehr zu verwenden.

Im vorherigen Kapitel wurde bereits ausgeführt, dass mit PHP erstellte Webseiten häufig zu großen Teilen aus gewöhnlichem HTML bestehen. Nur in kleinen Bereichen, in denen dynamische Funktionen erforderlich sind, kommen PHP-Scripte zum Einsatz. Diese Mischung führt dazu, dass der Webserver nicht den gesamten Code des Dokuments

als PHP-Programm interpretieren muss, sondern nur einzelne Teile davon. Um eine exakte Trennung zwischen den einzelnen Bereichen zu ermöglichen, ist es erforderlich, die PHP-Scripte im Text zu kennzeichnen. So weiß der Interpreter, an welchen Stellen er zum Einsatz kommen soll. Die übrigen Abschnitte gibt der Server unverändert wieder.

Für diese Aufgabe kommen Tags zum Einsatz. An der Stelle, an der der entsprechende Abschnitt beginnt, muss das Tag `<?php` stehen. Danach folgt das Script. Am Ende steht das Schlusstag `?>`.

Neben dieser Standardmethode gibt es noch zwei weitere Möglichkeiten. Zwar ist es als Anfänger nicht ratsam, diese zu verwenden, da hierfür oftmals die Serverkonfiguration angepasst werden muss. Das kann leicht zu Fehlern führen. Da PHP-Programmierer jedoch häufig auch mit Code in Kontakt kommen, den andere Personen erstellt haben, kann es vorkommen, dass sie dabei auch auf andere Tags treffen. Daher ist es wichtig, diese ebenfalls zu kennen. Als alternative Auszeichnungen ist es möglich, die Kombinationen `<? ?>` und `<% %>` zu verwenden.

3.2 Das erste Programm schreiben

Nun ist es an der Zeit, ein erstes eigenes Programm zu schreiben. Zu diesem Zweck ist es notwendig, zunächst den Texteditor zu öffnen, um die entsprechenden Kommandozeilen einzugeben. Da es sich hierbei um ein kleines PHP-Script handeln wird, ist es erforderlich, dieses im Code zu kennzeichnen. Zu diesem Zweck kommen die im vorherigen Abschnitt beschriebenen Tags zum Einsatz: `<?php` zum Öffnen des Scripts und `?>` am Ende. Dabei ist es sinnvoll, bereits beide Tags zu Beginn einzufügen. Das ist hilfreich, da viele Anfänger die Endung vergessen. Dieser häufige Fehler führt manchmal dazu, dass das ganze Programm nicht funktioniert. Im ersten Schritt sieht der Code daher wie folgt aus:

```
<?php

?>
```

In den Raum zwischen diesen beiden Tags sollen nun die eigentlichen Funktionen eingefügt werden. Für den Anfang ist es sinnvoll, einen einfachen Befehl zu wählen – beispielsweise `print`. Dieser erzeugt eine Ausgabe auf dem Bildschirm. Dabei ist es möglich, beliebige Wörter zu verwenden. Für das erste Programm soll die Ausgabe "Mein erstes Programm" lauten.

Der Text, der dabei erscheinen soll, muss immer in Anführungszeichen stehen. Das ist insbesondere dann wichtig, wenn Zahlen ausgeben werden sollen. Wenn beispielsweise nach dem `print`-Befehl "2+3" steht, gibt das Programm diesen Wert als Zeichenkette aus. Auf dem Bildschirm erscheinen daher genau die gleichen Zeichen: 2+3. Wenn dieser Ausdruck jedoch ohne Anführungszeichen eingefügt wird, betrachtet ihn das Programm als eine mathematische Operation. Das bedeutet, dass es zunächst den Wert dieses Terms berechnet: 2+3=5. Als Ausgabe erscheint daher lediglich die Zahl 5. Um Fehler zu vermeiden, ist es sehr wichtig, stets auf die richtige Setzung der Anführungszeichen zu achten.

Nach jedem PHP-Befehl muss ein Semikolon stehen. Dieses beendet die Funktion und ermöglicht, dass das Programm mit der nächsten Aufgabe fortfährt. Ein einfacher Zeilenumbruch ist hierfür nicht ausreichend. Sollte das Semikolon fehlen, interpretiert das Programm auch die darauffolgende Zeile als einen Teil des Befehls. Das kann zu Syntax-Fehlern führen und dafür sorgen, dass das Programm nicht lauffähig ist.

Wenn nun der vollständige Befehl zwischen die beiden bereits vorhandenen Tags eingefügt wird, ergibt sich folgender Code:

```
<?php
print "Mein erstes Programm";
?>
```

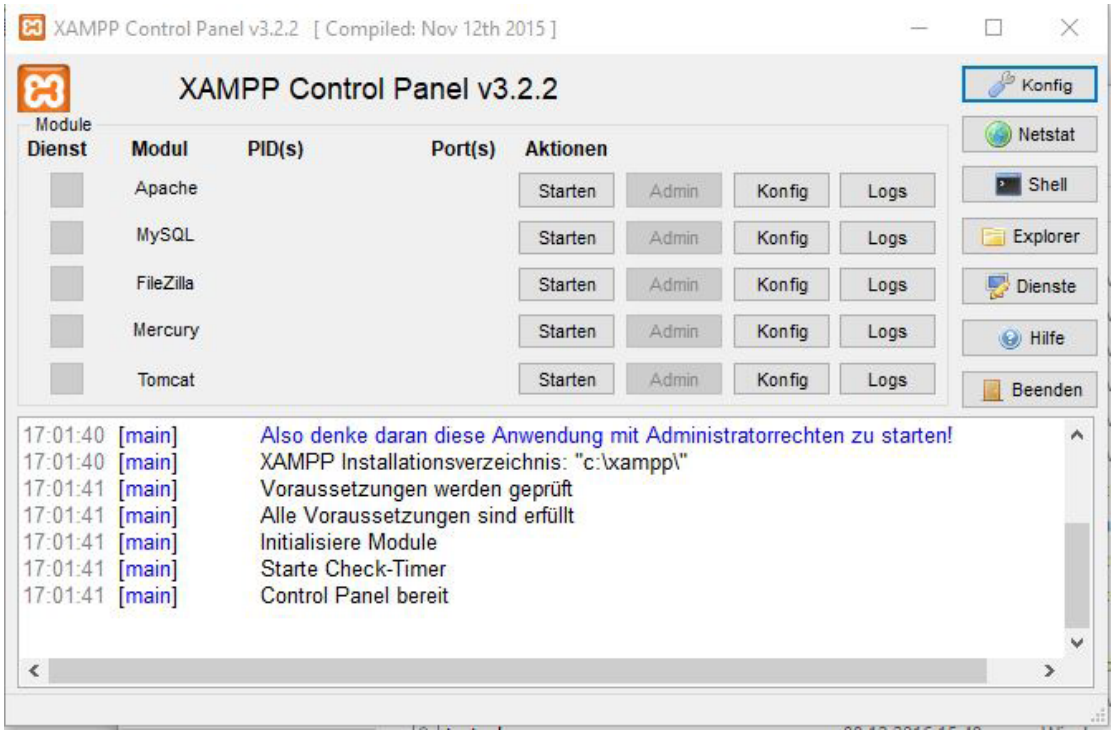
Damit ist das erste Programm bereits fertig. Dessen Funktionsweise ist ausgesprochen einfach. Wenn der Server diesen Code ausführt, erscheint lediglich der Titel "Mein erstes Programm" auf dem Bildschirm.

Nun ist es nur noch notwendig, den geschriebenen Code als Datei zu speichern. Der Dateiname darf dabei frei gewählt werden. Wichtig ist es lediglich, die Endung .php zu verwenden. Daran erkennt der Server, der das Programm ausführt, dass es sich hierbei um eine PHP-Datei handelt. Bei einigen Texteditoren ist es dafür notwendig, die Standard-Option – die Speicherung als txt.-Dokument – beim Abspeichern zu entfernen. Um dabei stets den Überblick zu behalten, empfiehlt es sich, selbsterklärende Namen auszuwählen, zum Beispiel: erstes_programm.php.

3.3 Das Programm zum Laufen bringen

Nun ist das erste Programm zwar bereits fertig, doch ist das Ergebnis bislang noch nicht auf dem Bildschirm zu erkennen. Die Anwender können lediglich den Text, den sie selbst geschrieben haben, im Texteditor betrachten. Um die Funktionsweise zu überprüfen, ist es jedoch notwendig, das Programm auszuführen. Hierfür kommt die Software XAMPP zum Einsatz. Da die Installation bereits in einem der vorherigen Kapitel erklärt wurde, sollte sich diese bereits auf dem PC des Lesers befinden.

Um die Software zu starten, ist es notwendig, die Datei xampp-control auszuführen. Diese befindet sich in dem Ordner, der bei der Installation von XAMPP gewählt wurde. Nach einem Doppelklick auf diese Datei erscheint folgendes Fenster:

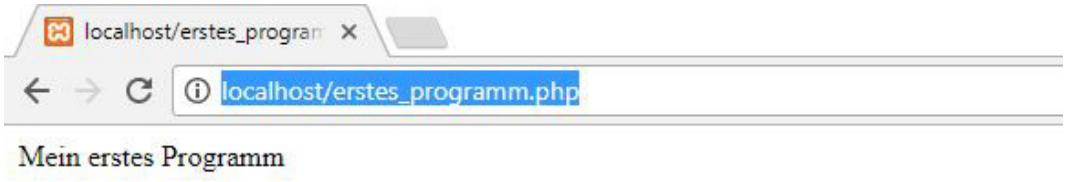


Screenshot 4: Der Start-Dialog bei XAMPP

Nun ist es notwendig, in der mit "Apache" gekennzeichneten Zeile auf den Button "Starten" zu klicken. Das führt dazu, dass der Webserver ausgeführt wird. Es ist nun möglich, das entsprechende Fenster zu schließen oder zu minimieren. Der Prozess findet weiterhin im Hintergrund statt.

Im nächsten Schritt ist es notwendig, das PHP-Programm in den Stammordner von XAMPP zu verschieben. Dabei handelt es sich um den Ordner "htdocs". Sollten später weitere Programme hinzukommen, ist es sinnvoll, Unterverzeichnisse einzufügen. Das verbessert die Übersicht deutlich.

Wenn das Programm im richtigen Verzeichnis platziert wurde, ist es erforderlich, es über einen Webbrowser aufzurufen. Dafür ist es notwendig, folgenden Pfad in die Adressleiste einzugeben: http://localhost/erstes_programm.php. Sollte der Leser die Datei in einem Unterverzeichnis abgespeichert haben, muss der entsprechende Ordner in den Pfad eingefügt werden. Daraufhin erscheint das Ergebnis des ersten Programms im Browser.



Screenshot 5 Die Ausgabe des ersten Programms im Browser

Häufig kommt es vor, dass der Programmierer einen ersten Entwurf ausführen lässt, danach das Ergebnis im Browser betrachtet und daraufhin einige Änderungen vornimmt. Wenn er nun die neue Version überprüfen will, muss er diese zunächst im Texteditor abspeichern. Dies ist wie gewohnt über die Menüleiste möglich. Da dieser Vorgang beim Programmieren mit PHP jedoch unzählige Male wiederholt werden muss, sei es dem Leser ans Herz gelegt, hierfür die Tastenkombination Strg+S zu verwenden. Auf diese Weise lässt sich viel Zeit einsparen. Danach ist es notwendig, die Seite im Browser zu aktualisieren. Das ist mit dem kreisförmigen Pfeil neben der Adressleiste oder durch die Betätigung der Taste F5 möglich.

3.4 Übung: ein einfaches Programm in PHP schreiben

Im folgenden Abschnitt soll das Gelernte in die Praxis umgesetzt werden. Daher stehen an dieser Stelle zwei Übungsaufgaben, die Sie selbstständig lösen sollten. Die notwendigen Kenntnisse wurden in den vorherigen Abschnitten vermittelt. Nach den beiden Übungsaufgaben sind die Lösungen eingefügt, um Ihre Ergebnisse zu kontrollieren. Allerdings sind bei vielen Aufgaben mehrere Lösungswege möglich. Daher ist es stets sinnvoll, das Programm zur Kontrolle im Browser aufzurufen, um selbst zu überprüfen, ob es alle Anforderungen erfüllt. Die

hier dargestellten Lösungsvorschläge dienen zum Vergleich und außerdem sind sie nützlich, wenn Sie alleine mit den Programmen nicht weiterkommen.

1. Erstellen Sie ein Programm, das die Besucher Ihrer Homepage begrüßt.
2. Zeigen Sie drei verschiedene Möglichkeiten auf, um die Zahl 8 als Ausgabe eines PHP-Programms darzustellen und erklären Sie die Unterschiede.

Lösungen:

1.

```
<?php
print "Herzlich Willkommen auf meiner Homepage!";
?>
```


Dieses Programm ist beinahe identisch mit dem Beispiel, das im Text beschrieben wurde. Der einzige Unterschied besteht darin, dass es einen anderen Text für die Wiedergabe verwendet. Dabei ist es wichtig, neben den beiden Tags für die Kennzeichnung des PHP-Scripts auf die Anführungszeichen und auf das Semikolon zu achten.

2.

```
<?php
print "8";
?>
```

In diesem Fall wird als Ausgabe der `print`-Funktion die Zahl 8 als Zeichenkette verwendet. Das liegt daran, dass sie in Anführungszeichen steht. Damit sind keine mathematischen Operationen möglich.

```
<?php
print 8;
?>
```

Im zweiten Beispiel ist die Ausgabe vollkommen identisch. Allerdings behandelt das Programm den Wert hierbei nicht als Zeichenkette, sondern als Zahl. Daher wäre es in einem weiteren Schritt hierbei möglich, weitere Berechnungen durchzuführen.

```
<?php
print 2*4;
?>
```

Im letzten Beispiel wird die Zahl 8 nicht direkt in die print-Funktion eingefügt. Sie ist das Ergebnis aus dem Term 2*4. Da dieser Ausdruck nicht in Anführungszeichen steht, betrachtet ihn PHP als mathematische Operation und berechnet automatisch das Ergebnis.

Kapitel 4

PHP und HTML miteinander verbinden

Bei der Gestaltung von Webseiten mit PHP besteht eine sehr enge Verbindung zwischen der Programmiersprache und dem HTML-Code. Das Ziel besteht in diesem Anwendungsbereich stets darin, mithilfe von PHP korrekte HTML-Seiten auszugeben. Daher ist es vor der Erstellung eines Programms mit PHP stets sinnvoll, sich zu überlegen, wie die Ausgabe aussehen soll. Daher ist es wichtig, die entsprechende Seite in HTML zu gestalten. Anschließend ist es erforderlich, die Stellen, die dynamische Inhalte verwenden, zu kennzeichnen. Anschließend müssen diese Bereiche durch ein PHP-Script ersetzt werden. Diese kleinen Programmstücke sollen nun Inhalte erzeugen, die an den jeweiligen Kontext angepasst sind. Sie sollen jedoch genau die gleiche Funktion übernehmen, wie die entsprechenden Elemente in der ursprünglichen HTML-Seite.

Für PHP-Programmierer ist es daher notwendig, diese enge Verbindung zwischen PHP und HTML zu berücksichtigen. Zum einen ist es wichtig, die PHP-Scripte sinnvoll in die HTML-Seite einzubinden. Zum anderen ist es notwendig, dass jedes einzelne dieser Scripte gültigen HTML-Code ausgibt. Die folgenden Abschnitte stellen vor, auf welche Weise es möglich ist, PHP und HTML miteinander zu verbinden.

4.1 HTML-Code mit PHP ausgeben

Im ersten Schritt soll dargestellt werden, auf welche Weise es möglich ist, HTML-Code mit PHP zu erzeugen. Diese Aufgabe ist im Prinzip ganz einfach. Hierfür kommt die gewöhnliche `print`-Funktion zum Einsatz. Dabei kommt jedoch kein einfacher Text zum Einsatz.

Dieser muss mit HTML-Tags versehen sein. Wenn beispielsweise eine Überschrift als solche gekennzeichnet werden soll, ist es lediglich notwendig, die entsprechenden `<h1>`-Tags in die print-Funktion aufzunehmen:

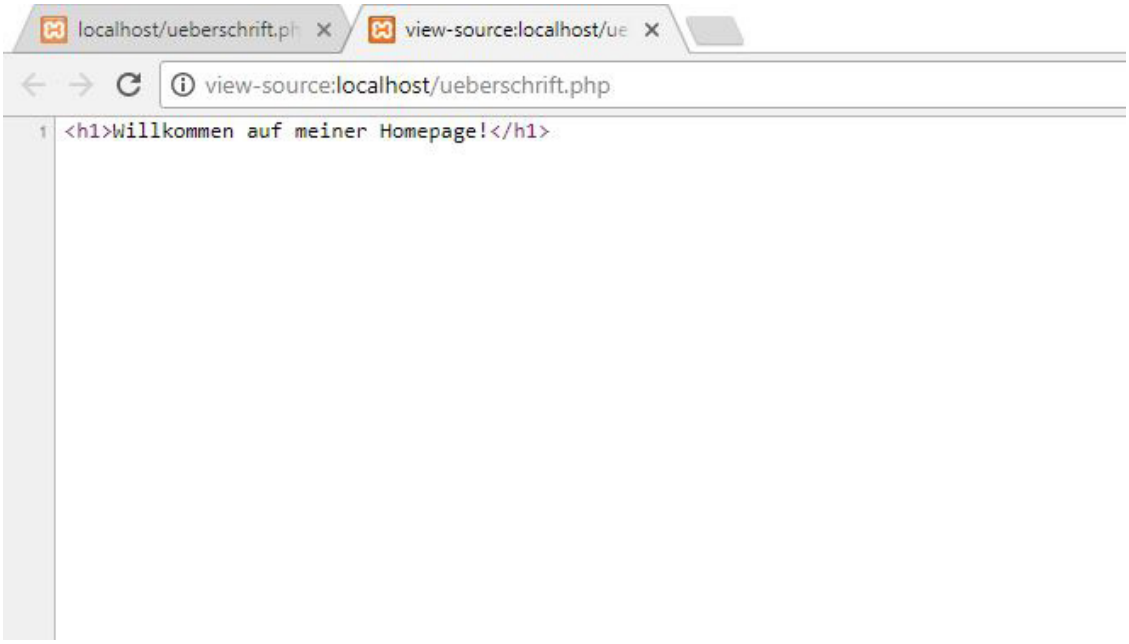
```
<?php
print "<h1>Willkommen auf meiner Homepage!</h1>";
?>
```

Screenshot 6 Die Ausgabe als Überschrift

Da es sich hierbei um eine Zeichenkette handelt, ist es selbstverständlich notwendig, Anführungszeichen zu verwenden. Das ist hierbei besonders wichtig, da die spitzen Klammern auch in PHP spezielle Funktionen haben. Sollten diese nicht eindeutig als Teil einer Zeichenkette gekennzeichnet sein, führt das zu Syntax-Fehlern, sodass der Server das Programm nicht ausführen kann.

In diesem Beispiel führt der Server das Programm aus und gibt als Ausgabe `<h1>Willkommen auf meiner Homepage</h1>` zurück und übermittelt diesen Code an den Browser. Dieser zeigt daraufhin lediglich die gewünschte Überschrift an: Willkommen auf meiner Homepage. Sie ist jedoch groß und fett gestaltet, woraus hervorgeht, dass der Browser den Text als Überschrift erkannt hat und entsprechend

darstellt. Das lässt sich außerdem durch die Anzeige des Quellencodes überprüfen. Dafür ist es notwendig, mit der rechten Maustaste auf die entsprechende Seite zu klicken und dann die Option "Quellencode anzeigen" auszuwählen. Daraufhin erscheint der Text mit den entsprechenden Tags.



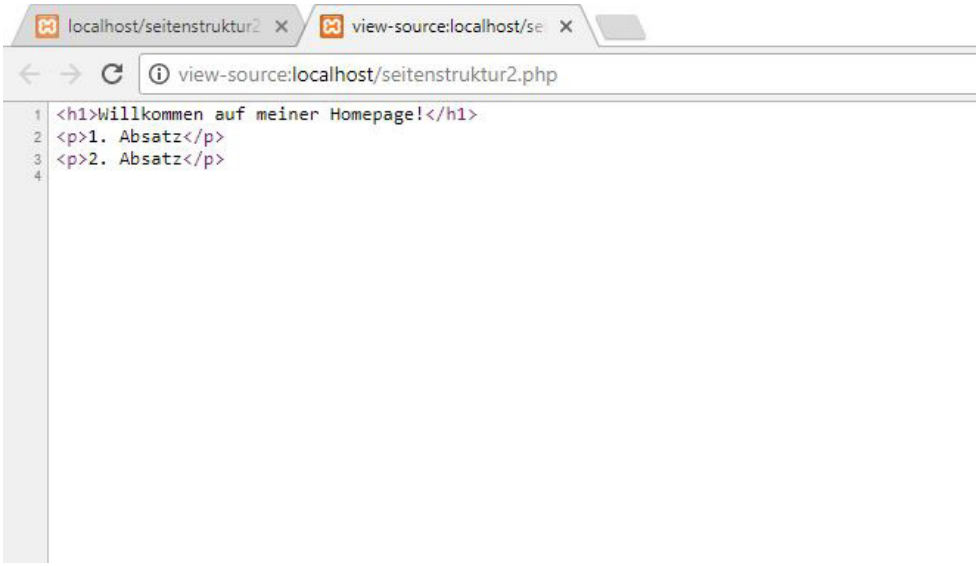
Screenshot 7 Anzeige des Quellencodes im Browser

Normalerweise besteht eine Internetseite nicht nur aus einer einzelnen Überschrift. Aus diesem Grund ist es in der Regel notwendig, mehrere Elemente mit PHP ausgeben zu lassen. Im Prinzip ist es möglich, alle unterschiedlichen Elemente im gleichen print-Befehl unterzubringen. Auf diese Weise wird der Code jedoch sehr unübersichtlich. Deshalb ist es empfehlenswert, für jeden einzelnen Bereich einen eigenen `print`-Befehl zu verwenden:

```
<?php
print "<h1>Willkommen auf meiner Homepage!</h1>";
print "<p>1. Absatz</p>";
print "<p>2. Absatz</p>";
?>
```

Screenshot 8 Die Ausgabe der Seitenstruktur im Browser

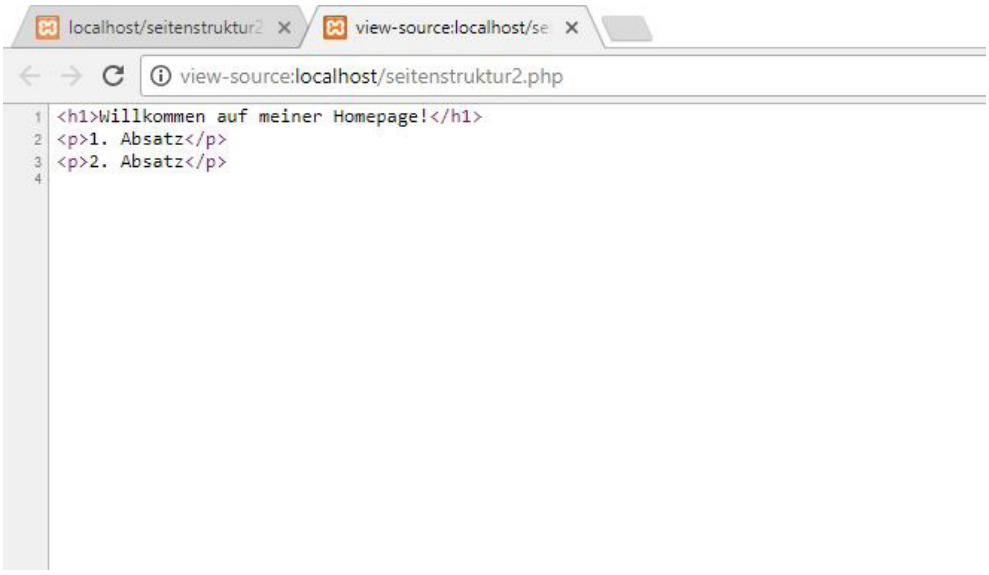
Wenn das Programm nun ausgeführt wird, erscheint auf der Seite zunächst die Überschrift in großen Schriftzeichen. Darunter folgen jeweils mit einigem Abstand und in deutlich kleinerer Schrift der erste und der zweite Absatz. Die Ausgabe ist daher genau wie gewünscht. Allerdings ist es auch hierbei sinnvoll, sich nochmals den Quellencode der entsprechenden Seite genau anzuschauen. Hierbei fällt auf, dass sich alle aufgeführten Bestandteile der Seite in der gleichen Zeile befinden. Bei einem kleinen Programm wie in diesem Beispiel stellt das sicherlich kein großes Problem dar. Bei umfangreichen Seiten wird der Quellencode dadurch jedoch extrem unübersichtlich. Das bringt extreme Nachteile bei der Programmierung in PHP mit sich. Denn häufig kommt es dabei vor, dass dem Programmierer ein kleiner Fehler unterlaufen ist, sodass die Seite nicht korrekt angezeigt wird. In diesem Fall ist es sehr hilfreich, den Quellencode zu untersuchen, um herauszufinden, worauf das Problem beruht. Wenn der gesamte Code jedoch in einer einzelnen Linie erscheint, ist es fast unmöglich, den Fehler zu finden.



Screenshot 9 Der Quellencode in einer einzigen Zeile

Aus diesem Grund ist es empfehlenswert, den Befehl `\n` nach jedem HTML-Element einzufügen. Dieser hat keinerlei Auswirkungen auf die Darstellung der Seite. Er bewirkt jedoch einen Zeilenumbruch im Quellencode. Das fertige Programm sieht daher wie folgt aus:

```
<?php
print "<h1>Willkommen auf meiner Homepage!</h1>\n";
print "<p>1. Absatz</p>\n";
print "<p>2. Absatz</p>\n";
?>
```



Screenshot 10 Übersichtlichere Gestaltung des Quellencodes

Wenn nach dieser kleinen Änderung der Quellentext erneut abgerufen wird, erscheint er deutlich besser geordnet, da jedes Element in einer eigenen Zeile steht.

4.2 PHP-Scripte in eine HTML-Seite integrieren

Der vorherige Abschnitt hat gezeigt, dass es kein Problem darstellt, HTML-Code mit einer PHP-Funktion zu erzeugen. Eigentlich ist es möglich, auf diese Weise die gesamte Internetseite zu gestalten. Dazu wäre es notwendig, für jedes einzelne HTML-Element eine eigene `print`-Funktion zu gestalten. Das würde jedoch einen sehr großen Aufwand mit sich bringen, der eigentlich vollkommen unnötig ist.

Das liegt daran, dass selbst bei dynamischen Seiten viele Elemente über die gesamte Website hinweg – oder zumindest bei thematisch zusammengehörigen Seiten – gleich bleiben. Daher ist es möglich, diese Bereiche statisch zu gestalten. Hierfür ist es lediglich notwendig, sie in HTML zu codieren. PHP-Scripte kommen dann nur in den Bereichen zum Einsatz, in denen tatsächlich dynamische Inhalte erzeugt werden.

Ein kleines Beispiel soll das verdeutlichen. Eine Website verwendet für jede einzelne Seite die gleiche Gestaltungsweise. Diese besteht aus einer Hauptüberschrift, die über die gesamte Site hinweg gleich bleibt. Außerdem wird eine Fußzeile mit dem Impressum eingefügt. Auch hierbei treten keine Unterschiede zwischen den einzelnen Seiten auf. Lediglich die eigentlichen Artikel, die auf den einzelnen Seiten erscheinen, sollen sich unterscheiden. Diese bestehen aus einer Artikelüberschrift und einem Textblock. Diese Elemente sollen aus einer Datenbank abgerufen werden. Daher ist es hierbei möglich, die festen Elemente nur mit HTML auszuzeichnen, und die variablen Inhalte mit PHP:

```
<h1>Hauptüberschrift für die Website</h1>
<?php
print "<h2>Artikelüberschrift</h2>\n";
print "<p>Textblock</p>\n";
?>
<p>Fußzeile</p>
```

Screenshot 11 Anzeige der einzelnen Elemente der Seiteninhalte

Screenshot 12 Im Quellencode erscheint lediglich HTML-Code

Bei diesem Beispiel gilt es zu beachten, dass die Bereiche Artikelüberschrift und Textblock lediglich als Platzhalter für eine Datenbankabfrage stehen. Auf welche Weise es mit PHP möglich ist, diese Daten abzurufen, wird erst in einem späteren Kapitel erläutert.

Dieses Codebeispiel zeigt, dass es ganz einfach ist, HTML und PHP zu mischen. Da der PHP-Code immer durch die entsprechenden Tags ausgezeichnet sein muss, erkennt der Interpreter genau, wann er ein entsprechendes Programm ausführen muss. Dabei ist es auch möglich, mehrere PHP-Scripte in das Dokument zu integrieren. Das ist häufig notwendig, da sich bei umfangreicheren Inhalten statische und dynamische Elemente abwechseln. Das gilt nicht nur für den `body`-Bereich, in dem die eigentlichen Inhalte stehen. Auch im `head`-Bereich gibt es viele Vorgaben, die für die gesamte Website gleich sind. Andere Teile wie der Titel und die Beschreibung werden hingegen für jede Seite individuell angepasst. Daher ist hierbei die Verwendung von PHP-Scripts für die dynamische Erzeugung sinnvoll. Eine komplette Seite – wenn auch im Beispiel stark vereinfacht – könnte folgendermaßen aussehen:

```
<html>
<head>
<?php print "<title>Individueller Titel der Seite</title>\n"; ?>
<?php print "<meta name=\"description\" content=\"Individuelle
Beschreibung der Seite\"/> \n"; ?>
<link rel="stylesheet" href="Link zum Stylesheet ist für alle Seiten
gleich"/>
</head>
<body>
<h1>Hauptüberschrift für die Website</h1>
<?php
print "<h2>Artikelüberschrift</h2>\n";
print "<p>Textblock</p>\n";
?>
<p>Fußzeile</p>
</body>
</html>
```

Screenshot 13 Bei der Anzeige im Browser ändert sich nur der Titel im oberen Registerblatt

Screenshot 14 Der Quellencode hat sich jedoch deutlich verändert

Anmerkung: Die Anführungszeichen der Meta-Description sind Teil des HTML-Quellencodes. Wenn diese jedoch ohne weiteren Zusatz in das PHP-Script eingefügt werden, geht der Interpreter davon aus, dass diese die Eingabe der `print`-Funktion – die ja ebenfalls mit einem An-

führungszeichen beginnt – beenden. Das führt zu Syntax-Fehlern. Daher ist es notwendig einen inversen Schrägstrich (\) vor dem Anführungszeichen einzufügen. So gibt der Interpreter dieses unverändert im Quellencode aus.

Alle Programmcodes aus diesem Buch sind als PDF zum Download verfügbar. Dadurch müssen Sie sie nicht abtippen:
https://bmu-verlag.de/books/php-mysql/

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

https://bmu-verlag.de/books/php-mysql/
Downloadcode: siehe Kapitel 18