

Was eine datenbankgestützte Webanwendung ist

Die Funktionsweise von MySQL kennenlernen

Ein Überblick über PHP

Ein Blick auf das Zusammenspiel von PHP und MySQL

Kapitel 1

Einführung in PHP und MySQL

Sie sollen oder möchten also eine interaktive Website entwickeln. Vielleicht hat Ihr Chef Ihnen die Aufgabe anvertraut, den Onlinekatalog der Firma zu produzieren, vielleicht träumen Sie von einem eigenen Onlinehandel oder vielleicht möchten Sie Ihrer Schwester helfen, ihre Gemälde über das Internet zu verkaufen. Oder Sie haben sich bereit erklärt, eine Website für die Mitglieder Ihres Akrobatenvereins zu erstellen! Wie auch immer Ihre Zielsetzung aussehen mag, auf jeden Fall müssen Daten erfasst und gespeichert werden (zum Beispiel über Produkte, Mitgliederpasswörter und so weiter), das heißt, Sie brauchen eine Datenbank. Außerdem muss es sich um eine Website handeln, auf der die Benutzer unterschiedliche Informationen abfragen (zum Beispiel über ein bestimmtes Produkt) oder eingeben können (zum Beispiel die Mitgliedsdaten). Für all diese Aufgaben eignen sich nur datenbankbasierte Webanwendungen.

Ich gehe bei den weiteren Beschreibungen davon aus, dass Sie sich mit der Entwicklung statischer Webseiten auf der Basis von HTML (HyperText Markup Language) auskennen, aber neue Bereiche betreten, wenn es um das Erstellen interaktiver Webseiten und den Entwurf von Datenbanken geht. Sie haben zwar drei Computergurus um Rat gebeten, diese haben aber mit unverständlichen Fachbegriffen um sich geworfen, wobei immer wieder die Wörter »schnell«, »leicht« und »kostenlos« fielen. Das hat Sie neugierig gemacht. Sie wollen herausfinden, wie Websites mit PHP und MySQL aufgebaut werden können.

PHP und MySQL bilden ein perfektes Team. In diesem Kapitel lernen Sie die grundsätzliche Funktionsweise und die Vorteile der beiden Programmiersprachen für unterschiedliche Aufgabenstellungen kennen und werden erfahren, wie mit ihnen im Zusammenspiel dynamische Datenbankanwendungen erstellt werden können.

Was ist eine datenbankgestützte Webanwendung?

Eine *Anwendung* – auch *App* (*Applikation*) – ist ein Programm oder eine Gruppe von Programmen, die für den Gebrauch durch *Endbenutzer* (zum Beispiel Kunden, Mitglieder oder Akrobaten) gedacht ist. Wenn dieser Endbenutzer über einen Webbrowser mit der Anwendung kommunizieren kann, handelt es sich um eine *Webanwendung*. Wenn diese Webanwendung die langfristige Speicherung von Daten in einer Datenbank erforderlich macht, haben Sie es mit einer *datenbankgestützten Webanwendung* zu tun. In diesem Buch lernen Sie, wie Sie eine datenbankgestützte Webanwendung entwickeln, die über einen Browser wie etwa Microsoft Edge, Google Chrome oder Mozilla Firefox aufgerufen und verwendet werden kann.



Der Begriff »datenbankgestützte Webanwendung« ist ein ziemliches Monstrum, auch wenn er die Sache korrekt beschreibt. Daher werde ich in diesem Buch oft auch nur von Webanwendung, Website oder Webprojekt sprechen. Gemeint ist damit aber letztlich immer eine Webanwendung mit Datenbankunterstützung.

Mit einer datenbankgestützten Webanwendung sollen Benutzer (also Menschen, die das Internet benutzen, oft auch einfach als »User« bezeichnet) in die Lage versetzt werden, bestimmte Aufgaben/Vorhaben zu realisieren. Dabei kann es sich um eine relativ einfache Website handeln, die hauptsächlich Daten anzeigt (zum Beispiel Stellenangebote, wenn der Benutzer eine Jobbezeichnung auswählt), oder um ein komplexes Programm mit vielen Funktionen (zum Beispiel Buchbestellungen bei Amazon oder Auktionen bei eBay).

Eine datenbankgestützte Webanwendung besteht aus zwei Teilen:

- ✓ **Datenbank:** Die *Datenbank* ist das Langzeitgedächtnis Ihrer Website. Ohne diese Datenbank könnte sie ihre Aufgabe nicht erfüllen. Aber eine Datenbank ist nur die eine Seite der Medaille.
- ✓ **Webanwendung:** Die *Webanwendung* ist ein Programm oder eine Gruppe von Programmen, das beziehungsweise die die Aufgaben ausführt/ausführen. Das Programm erzeugt die Anzeige im Browserfenster, es macht Ihre Website interaktiv, indem es die vom Benutzer eingegebenen Daten verarbeitet, sie in der Datenbank speichert und bei Bedarf wieder ausliest (eine Datenbank ist nur nützlich, wenn man Daten darin »ablegen« und wieder »herausholen« kann).

Webseiten, die Sie zuvor nur mit HTML erstellt haben, sind *statisch*: Es gibt keinen interaktiven Austausch. Alle Benutzer betrachten dieselben Webseiten. *Dynamische* Webseiten hingegen machen die Kommunikation und den Datenaustausch zwischen Benutzern und Webseite möglich. Verschiedenen Benutzern werden möglicherweise verschiedene Webseiten angezeigt. So sieht sich ein Benutzer beim Besuch der Website eines Möbelgeschäfts vielleicht die Sofas an, während ein anderer zu den Seiten mit Couchtischen »blättert«. Nur mit HTML lassen sich keine dynamischen Webseiten entwickeln. Für diese Aufgabenstellung müssen Sie zusätzlich auf eine Programmiersprache zurückgreifen.

JavaScript ist eine der Sprachen, die gern für die Entwicklung dynamischer Webseiten eingesetzt wird. Sie lockt mit einer Reihe nützlicher Eigenschaften, zum Beispiel mit der Funktion *mouseover* (um eine Navigationsschaltfläche hervorzuheben, wenn der Mauszeiger darauf zeigt) oder der Möglichkeit, die in ein Webformular eingegebenen Daten zu validieren (also die Gültigkeit der Daten zu überprüfen). JavaScript eignet sich aber nicht für die Zusammenarbeit mit Datenbanken. Daher würden Sie diese Sprache nie dafür einsetzen, um Daten eines Webformulars in einer Datenbank zu speichern. Im Unterschied dazu eignet sich PHP hingegen hervorragend für solche Aufgaben. Es kann die Daten eines Onlineformulars einlesen, validieren und zur Datenbank übertragen. Alle Codes für die Kommunikation mit Datenbanken sind in diesem Buch in PHP geschrieben.

Die Datenbank: Daten speichern

Das Herz eines datenbankgestützten Webprojekts ist die *Datenbank*, das Langzeitgedächtnis (hoffentlich besser funktionierend als mein eigenes), das die Daten für die Website speichert. Sie können sich eine Datenbank als ein elektronisches Ordnungssystem vorstellen, das Daten so ablegt, dass sie effizient verwaltet und bei Bedarf wiedergefunden werden können. Datenbanken können klein und einfach strukturiert sein, wie etwa eine Datenbank mit Titeln und Autorennamen Ihrer gesamten Büchersammlung, aber auch groß und komplex, wie zum Beispiel die Datenbank, die Amazon zum Speichern der Daten nutzt.

Der Inhalt von Datenbanken kann sehr unterschiedlich sein. Ein Onlinekatalog basiert auf einer Datenbank mit Produktinformationen. Eine Website für Mitglieder braucht eine Datenbank, die die verschiedenen Angaben über Mitglieder speichert. Die Website einer Jobbörse nutzt eine Datenbank (oder eventuell zwei Datenbanken) mit Daten über Stellenangebote und Lebensläufe der Bewerber. Die zu speichernden Daten können vielleicht so oder ähnlich überall im Internet gesammelt werden oder auch nur ganz speziell nur für Ihr Projekt benötigt werden.

Der Begriff *Datenbank* bezieht sich auf eine Datei oder eine Gruppe von Dateien, die die aktuellen Daten speichert. Auf die Daten kann über ein sogenanntes *DBMS* (Database Management System) zugegriffen werden. Fast alle DBMS-Programme sind heutzutage *RDBMS* (*relationale Datenbank-Management-Systeme*), in denen die Daten in relationalen Tabellen (Tabellen, die miteinander verknüpft sind) organisiert und gespeichert werden.

In diesem Buch wird als RDBMS das Datenbanksystem MySQL benutzt, weil es für Webprojekte besonders geeignet ist. Die Vorteile von MySQL beschreibe ich im Abschnitt »MySQL, die Datenbank meiner Wahl« weiter hinten in diesem Kapitel. In Kapitel 3 sowie Kapitel 6 geht es dann um die Organisation und Entwicklung einer MySQL-Datenbank.



Wenn von MySQL die Rede ist, handelt es sich heute mittlerweile um eine der beiden Varianten MySQL oder die weitgehend kompatible Variante MariaDB. Anfangs sollten Sie dabei möglichst eine für das verwendete System zusammengestellte Version verwenden. Momentan hat MariaDB die Nase vorn, was aber nicht zwingend so bleiben muss.

Die Webanwendung: Daten in eine Datenbank übertragen und auslesen

Damit eine Datenbank nützlich ist, muss sie Daten übernehmen und wieder ausspucken können. Als Werkzeug setzen Sie dazu Programme ein, die mit der Datenbank kommunizieren, um Daten zu speichern und auszulesen. Ein Programm muss sich mit der Datenbank verbinden und kann dann beispielsweise den Befehl erteilen: »Nimm diese Daten und speichere sie an einem bestimmten Ort.« Ein anderes Programm stellt vielleicht die Anfrage: »Finde diese Daten und übertrage sie an mich.« Die Programme werden ausgeführt, sobald der Benutzer Daten mit der Webseite austauscht. Dies geschieht zum Beispiel, wenn er ein Webformular ausfüllt und dann auf die Schaltfläche **SENDEN** (oder irgendeinen anderen Submit-Button) klickt. Die eingegebenen Daten werden verarbeitet und in der Datenbank gespeichert.

Mailinglisten, Foren und soziale Netzwerke

Nützliche Hilfestellung können Sie in *Mailinglisten*, *Foren* und/oder *sozialen Netzwerken* finden, in denen Beteiligte öffentlich die verschiedensten Themen diskutieren. Solche Plattformen existieren zu mehr oder weniger jedem nur denkbaren Thema: Tennis, Philosophie der Antike, Kochen, die Beatles, Schäferhunde, Politik etc. Hier werden die Daten über Verteilerlisten an die E-Mail-Adressen der registrierten Mitglieder versandt, die sich dann an der Diskussion beteiligen können.

Soziale Netzwerke funktionieren in diesem Zusammenhang ähnlich, wobei die Beiträge in den verschiedenen Sammlungen allgemein und öffentlich (oder nur von den jeweiligen Mitgliedern) in den Gruppen oder Communities diskutiert und/oder kommentiert werden können.

Soziale Netzwerke werden oftmals gesponsert. Sie können von Einzelpersonen oder von Organisationen betrieben werden. Viele Softwarevertreiber bieten Plattformen an, die sich mit ihrer Software befassen, Universitäten pflegen die verschiedensten Plattformen rund um den Lehrstoff. Benutzer können oftmals selbst eigene Themen initiieren oder sich für diese anmelden. Derartige Diskussionsplattformen können echte Fundgruben sein, da Hunderte, manchmal Tausende von Benutzern an einer Diskussion teilnehmen, oft auch die Entwickler, Händler und Mitarbeiter des technischen Supports. In der Regel sind Sie mit Ihrem Problem ja nicht allein und ganz egal welche Frage Sie haben, irgendein Diskutant wird vermutlich eine Lösung parat haben. Meistens erhalten Sie binnen kürzester Zeit eine Antwort. Zusätzlich erhalten Sie ein Archiv, in dem Sie stöbern können. Sofern Sie eine für Sie neue Software benutzen, können Sie viel über das Programm herausfinden, indem Sie einfach die Beiträge der Mailingliste eine Zeit lang verfolgen.

Auch zu PHP und MySQL beziehungsweise MariaDB gibt es Communities und Mailinglisten; für MariaDB zum Beispiel auf der Seite <https://lists.launchpad.net/aria-discuss/>, für PHP auf <http://www.php.net/mailling-lists.php> (in englischer Sprache). Eine Mailingliste in deutscher Sprache ist zu finden auf <http://lists.phpbar.de/mailman/listinfo/dbs>.

Generell sollten Sie sich mit kombinierten Begriffen auf die Suche begeben und für Sie interessante Fundstellen als Lesezeichen in Ihren Browser aufnehmen. »mailing list mariadb« wäre ein passender Suchbegriff für MariaDB und Mailinglisten. Unter <https://plus.google.com/communities/107672744094239537644> finden Sie bei Google+ die Community für MariaDB.



Suchen Sie sich ein soziales Netzwerk als Plattform aus, das zum Thema passt. Google+ steht im Ruf, ein Netzwerk für »Intellektuelle« zu sein, was aber vornehmlich nur bedeutet, dass es sich auch als Plattform für längere Beiträge und anspruchsvollere Themen und nicht nur für Kurzmitteilungen eignet.

MySQL, die Datenbank meiner Wahl

Bei MySQL und dessen Fork MariaDB handelt es sich um ein schnelles und einfach zu benutzendes relationales Datenbanksystem (RDBMS). Geschwindigkeit stand für die Entwickler von Anfang an im Mittelpunkt des Interesses. Zugunsten dieser Zielsetzung blieben zwar einige der Features, die die Konkurrenzprogramme (zum Beispiel Oracle oder Sybase) an Bord haben, auf der Strecke, aber dennoch bieten MariaDB und MySQL alles, was die Mehrheit der Datenbankentwickler braucht. Verglichen mit der Konkurrenz ist es einfacher zu installieren und einfacher einzusetzen. Und natürlich spricht der Preis für MySQL und MariaDB.

MySQL wurde anfangs von einem schwedischen Unternehmen entwickelt, später aber von Sun Microsystems übernommen. Nach seinem Erscheinen 2009 wurde *MariaDB*, das durch eine Abspaltung (Fork) aus MySQL entstanden ist, schnell von vielen Firmen als Standardinstallation übernommen. Weitere Angaben dazu finden Sie jeweils aktuell in der Wikipedia.

Vorteile von MySQL/MariaDB

Die Datenbank MySQL ist bei Webentwicklern sehr beliebt, vor allem weil sie so schnell und klein ist. Hinzu kommt, dass sie als *Open Source* kostenlos zur Verfügung steht. Hier ein Überblick über die wichtigsten Vorteile:

- ✓ **Sie ist schnell.** Die Entwickler legten von Anfang an viel Wert auf Geschwindigkeit.
- ✓ **Sie ist günstig.** Mit der GPL-Lizenz sind MySQL und MariaDB umsonst, die Gebühr für eine kommerzielle Lizenz ist niedrig.
- ✓ **Sie ist einfach zu nutzen.** Schon mit wenigen Anweisungen in der SQL-Sprache können Sie eine SQL-Datenbank erstellen und sie einsetzen. SQL ist die Standardsprache für die Kommunikation mit RDBMS. In Kapitel 7 lernen Sie die SQL-Sprache näher kennen.
- ✓ **MySQL steht für viele Betriebssysteme bereit.** Sie können die Datenbank zum Beispiel auf Windows, Linux, Mac OS, den meisten Varianten von Unix (einschließlich Solaris und AIX), FreeBSD und Irix einsetzen.

- ✓ **Sie steht bei Webhosts und auf NAS-Laufwerken zur Verfügung.** Wenn Ihre Website von einem Provider gehostet wird, kann MariaDB oder MySQL normalerweise ohne Extrakosten genutzt werden. NAS-Laufwerke (Network Attached Storage) führender Hersteller stellen eine MySQL-Variante auf ihren Speicherlaufwerken bereit. In Kapitel 2 gehe ich ein wenig auf die Verwendung von MySQL bei einem Webhost ein.
- ✓ **Es gibt umfangreiche Unterstützung und Hilfestellung.** Viele Nutzer stellen via Mailinglisten, in einschlägigen Foren und/oder in sozialen Netzwerken ihr Know-how zur Verfügung. Auch SQL-Entwickler nutzen derartige Plattformen häufig.
- ✓ **Die Datenbank ist sicher.** Das flexible Autorisierungssystem gewährt spezifischen Nutzern oder Gruppen von Nutzern ausgewählte oder auch alle Datenbankrechte (zum Beispiel das Recht, eine Datenbank anzulegen oder Daten zu löschen). Die Passwörter sind verschlüsselt.
- ✓ **Die Datenbanken können sehr groß sein.** MySQL kann mit mehr als 50 Millionen Zeilen umgehen. Die Standardbegrenzung für eine Tabelle liegt bei vier Gigabyte, aber Sie können diese Grenze theoretisch bis auf acht Millionen Terabyte (TB) hochschrauben (sofern Ihr Betriebssystem das hergibt).
- ✓ **MySQL kann benutzerdefiniert angepasst werden.** Aufgrund der Open-Source-GPL-Lizenz können Programmierer die MySQL-Software so anpassen, dass sie perfekt auf ihre spezifischen Anforderungen zugeschnitten ist.



Der Einfachheit halber werde ich von nun an vorwiegend nur noch von »MySQL« sprechen. Denken Sie daran, dass es sich hierbei im konkreten Einsatz um MySQL und/oder MariaDB handelt.

Wie MySQL funktioniert

MySQL besteht aus dem MySQL-Server, diversen Hilfsprogrammen, die Sie bei der Verwaltung der MySQL-Datenbank unterstützen, sowie einigen Programmen, die der MySQL-Server benötigt (die Sie aber nicht kennen müssen). Der Kern des Systems ist der MySQL-Server.

Der MySQL-Server ist der Manager des Datenbanksystems. Er steuert und verarbeitet alle Datenbankbefehle. Wenn Sie beispielsweise eine neue Datenbank anlegen wollen, senden Sie an den MySQL-Server eine Nachricht mit dem Wortlaut »Erstelle eine neue Datenbank mit dem Namen *Neuedaten*«. Der MySQL-Server wird aktiv und legt in seinem Datenverzeichnis einen Unterordner an, nennt ihn *Neuedaten* und schiebt die notwendigen Dateien im erforderlichen Format in diesen Ordner. Ähnlich gehen Sie vor, wenn Sie Daten hinzufügen möchten; Sie senden eine Nachricht an den MySQL-Server, übermitteln die Daten und teilen dem Server mit, wohin die Daten gehören. In Teil III dieses Buches erfahren Sie, wie Sie via *SQL* (Structured Query Language) mit dem Server kommunizieren.

Bevor Sie einen Befehl an den MySQL-Server schicken können, muss der Server laufen und auf Anfragen »lauern«. Standardmäßig ist der Server so eingerichtet, dass er mit dem Einschalten des Computers seine Arbeit aufnimmt und die ganze Zeit läuft. Aber diese Einstel-

lung ist nicht zwingend. Sie können den Server bei Bedarf auch jeweils manuell starten. Wenn der Server läuft, achtet er kontinuierlich auf Nachrichten, die für ihn bestimmt sind.

Mit dem MySQL-Server kommunizieren

Die gesamte Interaktion mit der Datenbank basiert auf Befehlen, die an den Server geschickt werden. Für den Versand gibt es verschiedene Möglichkeiten, aber in diesem Buch geht es in erster Linie um die Übermittlung per PHP. Zum Repertoire von PHP gehören spezifische Anweisungen, mit denen Befehle an den MySQL-Server geschickt werden.

Der MySQL-Server muss die an ihn gerichteten Befehle verstehen können. Für die Kommunikation benutzen Sie *SQL*, die Standardsprache, mit der viele RDBMS umgehen können. Auch der MySQL-Server »spricht« *SQL*. PHP ist weniger sprachbegabt und versteht kein *SQL*, was aber auch nicht notwendig ist: PHP stellt lediglich die Verbindung mit dem MySQL-Server her und leitet die *SQL*-Anweisungen an den Server weiter. Der MySQL-Server interpretiert den »*SQL*-Teil« und folgt den darin enthaltenen Anweisungen. Dann schickt er eine Antwort, gibt den Status aus oder teilt mit, was er gemacht hat (oder meldet einen Fehler, wenn er den Befehl nicht versteht oder ihn nicht ausführen kann).

In diesem Buch wird auch *phpMyAdmin* vorgestellt. Diese PHP-Software ist eine gängige Administrationsoberfläche für MySQL, die Sie auf Ihrem eigenen Computer nutzen können. Außerdem stellen die meisten Hosting-Anbieter und NAS-Laufwerke *phpMyAdmin* zur Verfügung.

In Teil III erfahren Sie genauer, wie Sie *SQL*-Befehle schreiben und an MySQL senden.

PHP bewegt Daten

Mit *PHP*, einer Programmiersprache, die speziell für das Internet entwickelt wurde, haben Sie das richtige Instrument zur Entwicklung dynamischer Websites an der Hand. Dank seiner vielen Features und Funktionen, die Webdesign und Programmierung leichter gemacht haben, wird PHP heute für über 70 Prozent der Websites eingesetzt. Diese Beliebtheit belegt insbesondere auch, dass PHP seine Aufgabe bestens erfüllt.

PHP steht für *PHP HyperText Preprocessor*. Zu seinen Anfangszeiten (entwickelt wurde es von einem Mann namens Rasmus Lerdorf) hieß es *Personal Home Page Tools*. Später wurde es in PHP umgetauft, da dieser Name besser zu den ausgereifteren Funktionen passte.



Die Syntax von PHP ähnelt der Sprache *C*, sodass Sie gut gerüstet für PHP sind, wenn Sie mit *C* vertraut sind. PHP ist dabei sogar einfacher als *C*, da es einige der komplizierteren Konzepte von *C* nicht nutzt und keine hardwarenahe Programmierung unterstützt. Als Programmiersprache speziell für dynamische Websites muss PHP das auch nicht leisten.

Die Stärke von PHP liegt vor allem in seiner Fähigkeit zur Zusammenarbeit mit Datenbanken. Fast jede Datenbank wird von PHP unterstützt. PHP stellt die Verbindung zur Datenbank her

und kommuniziert mit ihr. Wie PHP diese Verbindung technisch bewerkstelligt, müssen Sie nicht wissen. Sie teilen PHP einfach den Namen und den Ort der Datenbank mit, und PHP regelt den Rest: Es reicht Ihre Befehle weiter und liefert die Antwort der Datenbank zurück.



Sie finden im Internet umfassende Hilfe zu PHP, zum Beispiel auf der Website von PHP (www.php.net) oder unter der Adresse http://de.wikibooks.org/wiki/Webseiteentwicklung:_PHP. Schauen Sie auch mal auf die Seite <http://news.php.net>, ein Webportal zu den PHP-Mailinglisten.

Vorteile von PHP

Wegen seiner vielen Vorteile wächst die Beliebtheit von PHP kontinuierlich:

- ✓ **PHP ist schnell.** Da PHP in den HTML-Code integriert wird, ist die Reaktionszeit kurz.
- ✓ **PHP ist umsonst.** PHP ist der beste Beweis dafür, dass man nicht immer viel Geld für Qualität zahlen muss.
- ✓ **Die Benutzung von PHP ist einfach.** PHP beinhaltet viele Features und Funktionen, die für die Entwicklung einer dynamischen Website erforderlich sind. PHP-Anweisungen können problemlos in HTML-Code eingebettet werden.
- ✓ **PHP funktioniert mit vielen Betriebssystemen.** Sie können PHP mit Windows, Linux, Mac OS und den meisten Unix-Varianten nutzen.
- ✓ **PHP wird von vielen Webhosts und NAS-Laufwerken bereitgestellt.** Wenn Sie Ihre Website bei einem Webpace-Anbieter oder über ein eigenes NAS-Laufwerk veröffentlichen werden, gibt es PHP zu meist kostenlos dazu.
- ✓ **Es gibt jede Menge Hilfe und Unterstützung.** Viele Nutzer stellen in Mailinglisten und in Foren ihr Wissen zur Verfügung.
- ✓ **PHP ist sicher.** Aufgrund der regen Open-Source-Community werden Sicherheitslücken schnell geschlossen (sicherheitsrelevante Fehler, die der Programmierer beim Einsatz von PHP einbaut, kann natürlich auch PHP nicht ausbügeln).
- ✓ **PHP ist für die Unterstützung von Datenbanken programmiert.** PHP bietet Funktionen, die speziell für die Zusammenarbeit mit spezifischen Datenbanken entwickelt wurden.
- ✓ **PHP lässt sich benutzerdefiniert anpassen.** Aufgrund der Open-Source-Lizenz können Programmierer PHP so anpassen, dass es ihren spezifischen Anforderungen gerecht wird.

Wie PHP funktioniert

PHP ist eine integrierte Skriptsprache, wenn es für Webseiten eingesetzt wird. PHP-Anweisungen werden in den HTML-Quelltext eingebettet. Dazu nutzen Sie spezielle Tags, ähnlich, wie gewöhnliche HTML-Tags. Skripte mit integriertem PHP-Code schreiben und bearbeiten Sie nicht anders als reguläre HTML-Seiten.

PHP arbeitet mit dem Webserver zusammen. Der Webserver ist die Software, die die Webseiten publiziert. Wenn Sie eine URL in die Adressleiste Ihres Browsers eingeben, übertragen Sie eine Nachricht zum Webserver, mit der Aufforderung, Ihnen eine HTML-Datei zu senden. Der Webserver reagiert und schickt Ihnen die angeforderte Datei. Ihr Browser liest die HTML-Datei und zeigt sie im Browserfenster an. Auch wenn Sie einen Link auf einer Webseite anklicken oder per Mausklick auf einen »Submit-Button« ein Webformular senden, stellen Sie an den Webserver die Anfrage, Ihnen eine bestimmte Datei zu liefern.

Wenn PHP im Spiel ist, rechnet der Webserver automatisch damit, dass Dateien mit einer bestimmten Dateierweiterung PHP-Anweisungen enthalten. Meist lautet die Erweiterung .php, seltener .phtml, aber theoretisch könnte auch jede andere Erweiterung benutzt werden. Erhält der Webserver die Anfrage nach einer Datei mit der festgelegten Erweiterung, überträgt er den HTML-Teil so, wie er in der Datei steht, an den Browser; der PHP-Teil hingegen wird zunächst ausgeführt und dann gesendet.

Wenn PHP-Anweisungen verarbeitet werden, schickt der Webserver nur das Ergebnis an den Browser. Die PHP-Anweisungen selbst sind in dem Ergebnis nicht enthalten. Dies bedeutet, dass PHP-Code für den Nutzer nicht einsehbar ist. Hier ein einfacher PHP-Befehl:

```
<?php echo "<p>Hallo Welt</p>"; ?>
```

<?php ist das öffnende PHP-Tag und ?> das schließende Tag. echo ist der PHP-Befehl, der PHP dazu auffordert, den nachfolgenden Text auszugeben. PHP arbeitet die Anweisung ab und gibt Folgendes aus:

```
<p>Hallo Welt</p>
```

Dies ist ganz normales HTML. Es wird an den Browser des Benutzers geliefert. Der Browser interpretiert diesen »Textschnipsel« als HTML und zeigt eine Webseite mit einer Zeile an: Hallo Welt. Der PHP-Befehl wird nicht an den Browser mitgeschickt, sodass der Benutzer ihn nicht zu sehen bekommt.

PHP ist nicht unmittelbar Bestandteil aller Webserver, kann aber mit zusammen mit vielen der populären Server genutzt werden. Da es ursprünglich als Projekt der Apache Software Foundation entwickelt wurde, arbeitet es zwar am besten mit dem Webserver *Apache* zusammen, es lassen sich aber auch Server wie beispielsweise Microsoft IIS verwenden.



Obwohl PHP mit diversen Webservern zusammenarbeiten kann, funktioniert es am besten mit Apache. Wenn Sie wählen können, welcher Webserver in Ihrer Organisation eingesetzt wird, sollten Sie sich für Apache entscheiden. Apache ist kostenlos erhältlich, Open Source, stabil und populär. Seit 1996 ist Apache der beliebteste Webserver im Internet. Er läuft auf Windows, Linux, Mac OS und den meisten Varianten von Unix.

MySQL und PHP, ein perfektes Paar

MySQL und PHP werden häufig zusammen eingesetzt und gelegentlich als *dynamische Duo* bezeichnet. MySQL/MariaDB stellt die Datenbank bereit, und PHP liefert die Programme für Ihr Webprojekt.

Vorteile der Beziehung

Als Pärchen bieten MySQL und PHP diverse Vorteile:

- ✓ **Sie sind kostenlos.** Kann es ein besseres Preis-Leistungs-Verhältnis geben?
- ✓ **Sie sind weborientiert.** Eigens für die Benutzung von Websites geschaffen, bieten sie spezielle Features für die Entwicklung dynamischer Websites.
- ✓ **Ihre Benutzung ist einfach.** Beide sind so konzipiert, dass eine Website relativ schnell und problemlos entwickelt werden kann.
- ✓ **Sie sind schnell.** Für beide war Geschwindigkeit das primäre Ziel. Daher liefern sie die angeforderten Webseiten fast konkurrenzlos schnell zum Browser des Users.
- ✓ **Die Kommunikation zwischen beiden funktioniert bestens.** Spezielle PHP-Befehle dienen der Kommunikation mit MySQL. Die technischen Details und Feinheiten müssen Sie nicht kennen, die überlassen Sie einfach PHP.

Wie die Zusammenarbeit von MySQL und PHP funktioniert

In einem datenbankgestützten Webprojekt teilen sich PHP und MySQL die Aufgaben: PHP liefert die Programme und MySQL die Datenbank. Mit PHP schreiben Sie die Skripte, die die Aufgaben der Website definieren und ausführen. PHP kann für ganz einfache Aufgaben (wie etwa die Anzeige einer Webseite) oder für komplizierte Aufgaben eingesetzt werden (wie etwa die Annahme und Bestätigung von Daten, die der Benutzer in ein HTML-Formular eingegeben hat). Eine Aufgabe muss Ihr Programm auf jeden Fall ausführen: Daten an eine Datenbank senden und Daten aus einer Datenbank auslesen. Das ist für PHP kein Problem. Es kennt (vordefinierte) Funktionen, die genau das leisten: Daten an die MySQL-Datenbank übermitteln und Daten aus dieser Datenbank auslesen.

PHP-Befehle und -Anweisungen werden mit PHP-Tags in die HTML-Datei integriert. Sofern Daten gespeichert und ausgelesen werden müssen, verwenden Sie spezielle PHP-Befehle für die Kommunikation mit der SQL-Datenbank. Um zum Beispiel die Verbindung mit der Datenbank herzustellen, verwenden Sie einen Befehl, der die gewünschte Datenbank und das Passwort angibt. Die Datenbank muss nicht auf dem gleichen Computer liegen wie die Website; die Zusammenarbeit funktioniert mit allen Datenbanken irgendwo im Netz. Mit einem anderen PHP-Befehl schicken Sie eine SQL-Anweisung an MySQL, die angibt, welche Aufgabe ausgeführt werden soll. MySQL antwortet mit einer Statusmeldung, aus der zu ersehen ist, ob die Aktion erfolgreich war. Sofern Probleme auftreten, wird eine Fehlermeldung zurückgegeben. Wenn die SQL-Anweisung Informationen aus der Datenbank auslesen soll, werden die angefragten SQL-Daten zurückgeliefert und von PHP temporär gespeichert, sodass sie für die Verwendung in Programmen verfügbar sind. Sie können die ausgelesenen Daten anschließend zum Beispiel auf einer Webseite präsentieren oder eine Statusmeldung im Browser ausgeben, die den Nutzer darüber informiert, dass seine Formulareingaben gespeichert wurden.

Als RDBMS (Relational Database Management System) kann MySQL komplexe Informationen speichern. PHP kann komplizierte Operationen mit Daten durchführen, sowohl mit den Daten, die vor der Übertragung in die Datenbank bearbeitet werden müssen, als auch mit den ausgelesenen Daten, die vor der Ausgabe im Browser manipuliert werden müssen. Zusammen bilden MySQL und PHP ein Team, mit dem sich anspruchsvolle und ausgefeilte datenbankbasierte Webprojekte realisieren lassen.

Mit Änderungen auf dem Laufenden bleiben

Sowohl PHP als auch MySQL sind Open-Source-Programme. Wenn Sie bisher nur Software von großen Anbietern wie Microsoft, Macromedia oder Adobe verwendet haben, werden Sie feststellen, dass es sich bei Open-Source-Software um eine eigene Spezies handelt. Sie wird von Programmierern entwickelt, die sich in ihrer Freizeit und nur zum Spaß und unentgeltlich den Projekten widmen.

Open-Source-Software wird regelmäßig gewartet und ändert sich daher oft häufiger, als Sie es von kommerziellen Programmen gewohnt sind. Auf Probleme wird in der Regel schnell reagiert. Wenn zum Beispiel ein sicherheitsrelevantes Problem auftaucht, erscheint mitunter innerhalb eines Tages eine neue Version, die diese Sicherheitslücke stopft. Sie erhalten keine Hochglanzbroschüren und werden auch nicht über ganzseitige Anzeigen stolpern, in denen neue Versionen bereits ein Jahr vor Erscheinen angekündigt werden. Wenn Sie sich also nicht die Mühe machen und selbst auf dem Laufenden halten, kann es vorkommen, dass Sie eine neue Version oder Informationen über ein gravierendes Sicherheitsproblem einfach verpassen.



Schauen Sie regelmäßig auf den PHP- und MySQL-Webseiten vorbei. Hier werden Informationen veröffentlicht, die für Sie wichtig sind. Denken Sie auch daran, online Mailinglisten, Foren und/oder soziale Netzwerke zu nutzen (lesen Sie dazu den Kasten am Anfang des Kapitels).

PHP-Versionen

Die aktuelle Version von PHP ist PHP 7. Programme, die noch PHP 5 nutzen, werden zunehmend weniger. Wenn Sie ein neues PHP-Programm schreiben, sollten Sie möglichst PHP 7 einsetzen, da dieses deutlich schneller als frühere Versionen ist.

Da die Entwicklung von PHP 6 eingestellt wurde, kam es zu einem Versionssprung von 5 auf 7, wobei die geplanten Neuerungen der Version 6 bereits in die Versionen 5.3 und 5.4 eingeflossen sind.

Wenn Sie ein PHP-Skript bearbeiten müssen, um es auf einen neueren Stand zu bringen, sollten Sie sich beispielsweise über die folgenden Änderungen informieren:

- ✓ Die Einstellungen `register_globals` und `magic_quotes` gibt es nicht mehr.
- ✓ Lange Namen wie zum Beispiel `HTTP_POST_VARS` für globale Arrays gibt es nicht mehr.

- ✓ Wie bereits bei JavaScript wird es auch in PHP einen strikten Modus geben, der die laufende Änderung von Datentypen verhindert:

```
declare(strict_types=1);
```

- ✓ Array-Konstanten
- ✓ Anonyme Klassen

Versionen MySQL und MariaDB

5.5 lautete die Versionsnummer bei der vorigen Auflage dieses Buches. Und dabei ist es bis in den Mai 2017 hinein fast geblieben. Abgesehen von der einen oder anderen fortgeschrittenen Funktion hat sich daher an den Befehlen wenig geändert.

Weil in MySQL die Neuerungen von Sun zunehmend nur in die kommerziellen Versionen integriert wurden, wechselten viele Distributionen zu MariaDB. Hier wurde dann von der Version 5.5.56 auf 10.0 gesprungen. Die Beispiele in diesem Buch sollten Sie problemlos mit allen halbwegs gängigen Versionen nachvollziehen können.

Dabei ist es aber natürlich nicht empfehlenswert, Webprojekte mit veralteten SQL-Versionen zu betreiben.