

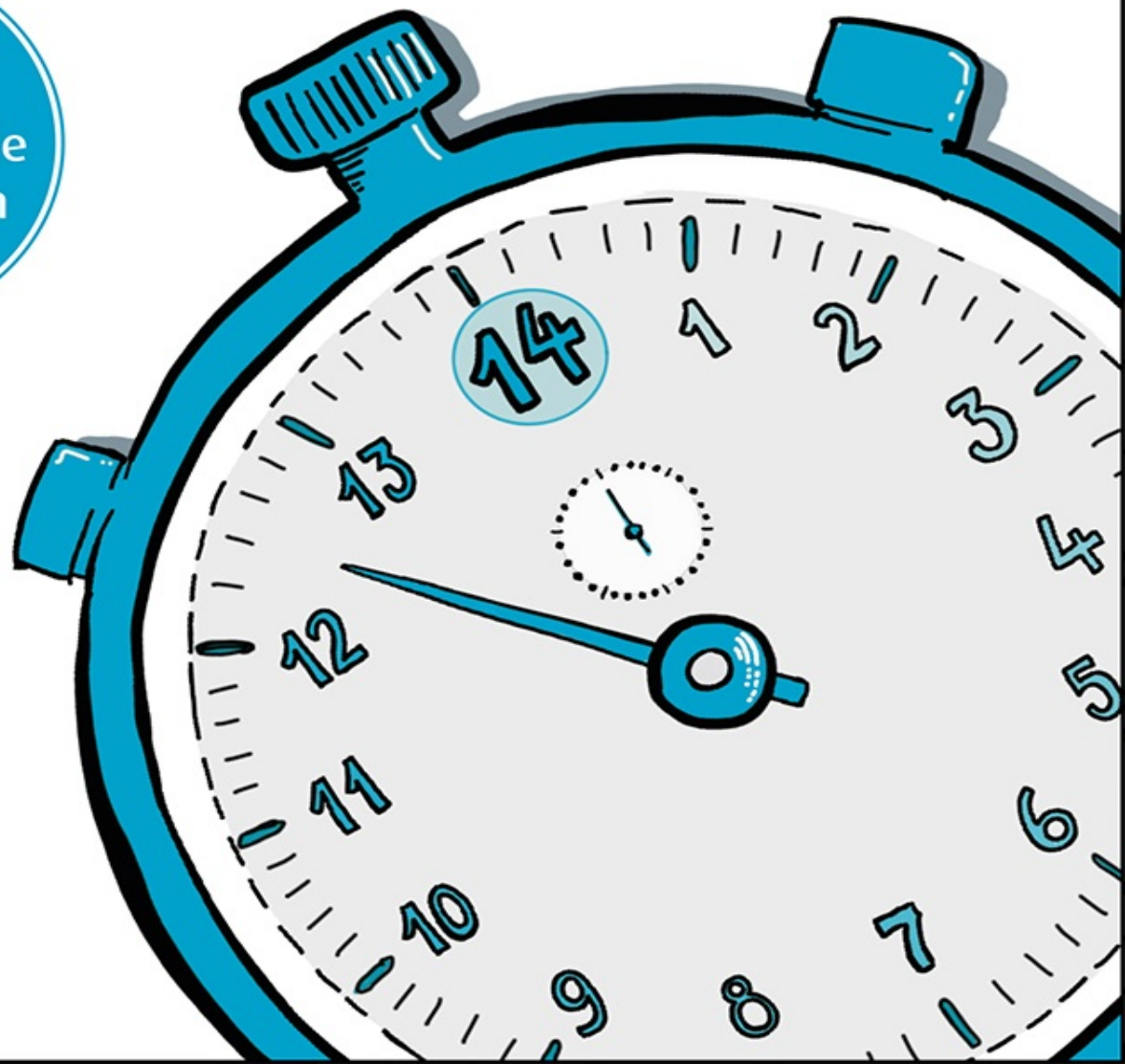
Philipp Rieber

# PHP & MySQL Schnelleinstieg

Programmieren lernen in 14 Tagen

> Einfach und ohne Vorkenntnisse <

Zahlreiche  
Praxisbeispiele  
und Übungen



mitp

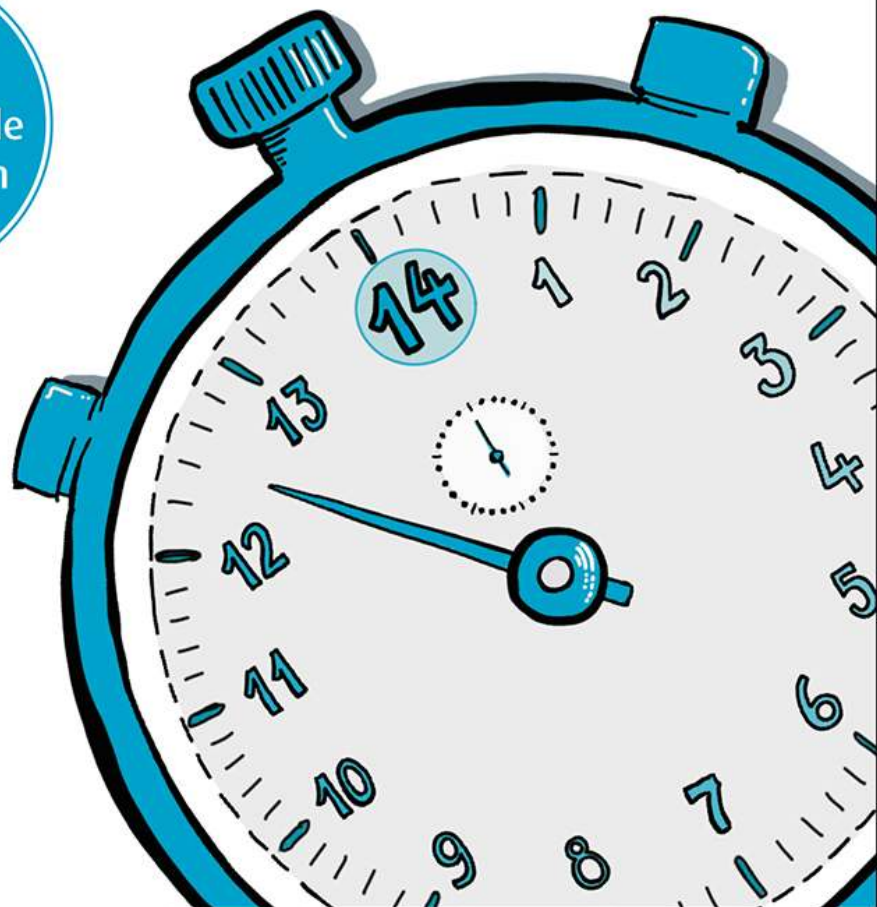
Philipp Rieber

# PHP & MySQL Schnelleinstieg

Programmieren lernen in 14 Tagen

> Einfach und ohne Vorkenntnisse <

Zahlreiche  
Praxisbeispiele  
und Übungen





## Hinweis des Verlages zum Urheberrecht und Digitalen Rechtemanagement (DRM)

Liebe Leserinnen und Leser,

dieses E-Book, einschließlich aller seiner Teile, ist urheberrechtlich geschützt. Mit dem Kauf räumen wir Ihnen das Recht ein, die Inhalte im Rahmen des geltenden Urheberrechts zu nutzen. Jede Verwertung außerhalb dieser Grenzen ist ohne unsere Zustimmung unzulässig und strafbar. Das gilt besonders für Vervielfältigungen, Übersetzungen sowie Einspeicherung und Verarbeitung in elektronischen Systemen.

Je nachdem wo Sie Ihr E-Book gekauft haben, kann dieser Shop das E-Book vor Missbrauch durch ein digitales Rechtemanagement schützen. Häufig erfolgt dies in Form eines nicht sichtbaren digitalen Wasserzeichens, das dann individuell pro Nutzer signiert ist. Angaben zu diesem DRM finden Sie auf den Seiten der jeweiligen Anbieter.

Beim Kauf des E-Books in unserem Verlagsshop ist Ihr E-Book DRM-frei.

Viele Grüße und viel Spaß beim Lesen,

*Ihr mitp-Verlagsteam*





# **PHP & MySQL**

## **Schnelleinstieg**

**Programmieren lernen in 14 Tagen  
Einfach und ohne Vorkenntnisse**





# Impressum

Bibliografische Information der Deutschen Nationalbibliothek

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

ISBN 978-3-7475-0397-3

1. Auflage 2022

[www.mitp.de](http://www.mitp.de)

E-Mail: [mitp-verlag@sigloch.de](mailto:mitp-verlag@sigloch.de)

Telefon: +49 7953 / 7189 - 079

Telefax: +49 7953 / 7189 - 082

© 2022 mitp Verlags GmbH & Co. KG

Mit freundlicher Genehmigung der HfG Schwäbisch Gmünd verwenden einige Abbildungen Emojis von OpenMoji (<https://openmoji.org>) – dem Open-Source Emoji- und Icon-Projekt.

Dieses Werk, einschließlich aller seiner Teile, ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Verlages unzulässig und strafbar. Dies gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

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 und daher von jedermann benutzt werden dürften.

Lektorat: Janina Bahlmann

Sprachkorrektur: Claudia Fluor

Covergestaltung: Janina Bahlmann, Christian Kalkert

Covergrafik & Icons: Tanja Wehr, sketchnotelovers

Electronic Publishing: Petra Kleinwegen

Dieses Ebook verwendet das ePub-Format und ist optimiert für die Nutzung mit dem iBooks-reader auf dem iPad von Apple. Bei der Verwendung anderer Reader kann es zu Darstellungsproblemen kommen.

Hinweis des Verlages zum Urheberrecht und Digitalen Rechtemanagement (DRM)

Der Verlag räumt Ihnen mit dem Kauf des ebooks das Recht ein, die Inhalte im Rahmen des geltenden Urheberrechts zu nutzen. Dieses Werk, einschließlich aller seiner Teile, ist urheberrechtlich geschützt. Jede Verwertung außerhalb der

engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Verlages unzulässig und strafbar. Dies gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und Einspeicherung und Verarbeitung in elektronischen Systemen.

Der Verlag schützt seine ebooks vor Missbrauch des Urheberrechts durch ein digitales Rechtemanagement. Bei Kauf im Webshop des Verlages werden die ebooks mit einem nicht sichtbaren digitalen Wasserzeichen individuell pro Nutzer signiert.

Bei Kauf in anderen ebook-Webshops erfolgt die Signatur durch die Shopbetreiber. Angaben zu diesem DRM finden Sie auf den Seiten der jeweiligen Anbieter.



# Inhalt

## Impressum

## Einleitung

- E.1 Programmieren lernen in 14 Tagen
- E.2 Der Aufbau des Buchs
- E.3 Programmtexte und Lösungen zum Download
- E.4 Fragen und Feedback

## 1 Erste Schritte mit PHP

- 1.1 Wofür wird PHP eingesetzt?
  - 1.1.1 PHP zur Erzeugung dynamischer Webseiten
  - 1.1.2 PHP zur Bereitstellung von Webservices
  - 1.1.3 Kommandozeilenprogramme mit PHP
- 1.2 Stärken von PHP
- 1.3 Schwächen von PHP
- 1.4 Was ist für die Entwicklung mit PHP notwendig?
- 1.5 Quelltext und PHP-Interpreter
- 1.6 Kommandozeile nutzen
  - 1.6.1 Kommandozeile starten
  - 1.6.2 Kommandozeilen-Einmaleins
- 1.7 PHP installieren
  - 1.7.1 Windows
  - 1.7.2 macOS

- 1.7.3 Linux
- 1.8 Quelltext-Editor verwenden: Visual Studio Code
- 1.9 Das erste PHP-Skript ausführen
- 1.11 Wie entstand PHP?
- 1.12 Übungen

## **2 Variablen, Datentypen und Konstanten**

- 2.1 Daten in einer Variablen erfassen
  - 2.1.1 EVA: Grundprinzip der Datenverarbeitung
  - 2.1.2 Variablen erzeugen und verwenden
  - 2.1.3 Regeln für Variablennamen
  - 2.1.4 Datentypen und die »dynamische Typisierung«
- 2.2 Einfache Datentypen
  - 2.2.1 Daten wörtlich aufschreiben: Literale
  - 2.2.2 Zeichenkette (String)
  - 2.2.3 Ganzzahl (Integer)
  - 2.2.4 Gleitkommazahl (Float)
  - 2.2.5 Wahrheitswert (Boolean)
- 2.3 Der spezielle Datentyp: NULL
- 2.4 Der vielseitige Datentyp: Array
  - 2.4.1 Numerisch indizierte Arrays
  - 2.4.2 Assoziative Arrays
  - 2.4.3 Mehrdimensionale Arrays
- 2.5 Datentypen umwandeln
  - 2.5.1 Type Juggling
  - 2.5.2 Type Casting

## 2.6 Programmstrukturen

### 2.6.1 Ausdrücke

### 2.6.2 Anweisungen

## 2.7 Konstanten

### 2.7.1 Literale Konstanten

### 2.7.2 Benannte Konstanten

### 2.7.3 Vordefinierte Konstanten

### 2.7.4 Magische Konstanten

## 2.8 Übungen

# **3 Programmablauf mit Kontrollstrukturen steuern**

## 3.1 Programmablauf verzweigen

### 3.1.1 Bedingte Anweisung mit if

### 3.1.2 Beispiel: Passwortabfrage

### 3.1.3 Verzweigung mit if...else

### 3.1.4 Beispiel: Eintrittskarte Zoo

### 3.1.5 Der ternäre Operator ?:

### 3.1.6 Fallunterscheidung mit if...elseif...else

### 3.1.7 Beispiel: Bankautomat

## 3.2 Bedingungen formulieren

### 3.2.1 Vergleichoperatoren

### 3.2.2 Gleichheit (==) und Identität (===)

### 3.2.3 Logische Operatoren

### 3.2.4 Beispiel: Schaltjahrprüfung

## 3.3 Programmabschnitte mit Schleifen wiederholen

### 3.3.1 Arrays durchlaufen - foreach

- 3.3.2 Beispiel: Getränkekarte
- 3.3.3 Zählschleife – for
- 3.3.4 Beispiel: Countdown
- 3.3.5 Bedingte Wiederholung – while
- 3.3.6 Beispiel: Lostopf
- 3.3.7 Schleifen abbrechen – break
- 3.3.8 Schleife beim nächsten Durchlauf fortsetzen – continue
- 3.3.9 Endlosschleifen
- 3.4 Alternative Syntax für Kontrollstrukturen
- 3.5 Übungen

## **4 Programmierfehler und PHP-Konfiguration**

- 4.1 Programmierfehler
  - 4.1.1 Syntaxfehler
  - 4.1.2 Laufzeitfehler
  - 4.1.3 Logische Fehler
- 4.2 PHP konfigurieren: die php.ini-Datei
  - 4.2.1 Was ist die php.ini-Datei?
  - 4.2.2 Wo ist die php.ini und wie wird sie bearbeitet?
  - 4.2.3 PHP-Einstellungen zur Laufzeit
  - 4.2.4 PHP-Erweiterungen aktivieren
- 4.3 Fehlerstufen
- 4.4 Fehlersichtbarkeit einstellen
- 4.5 Übungen

## **5 Funktionen**

## 5.1 Native Funktionen aus der PHP-Bibliothek verwenden

### 5.1.1 Signatur einer Funktion

### 5.1.2 Rückgabewerte

### 5.1.3 Mit Zahlen arbeiten: Mathematische Funktionen

### 5.1.4 Mit Texten arbeiten: String-Funktionen

### 5.1.5 Beispiel: E-Mail validieren

### 5.1.6 Mit Arrays arbeiten: Array-Funktionen

### 5.1.7 Beispiel: Passwort generieren

### 5.1.8 Die Unixzeit und Datumsfunktionen

## 5.2 Eigene Funktionen definieren

### 5.2.1 Hinweise auf Datentypen: Type Hints

### 5.2.2 Geltungsbereich von Variablen

## 5.3 Übungen

# 6 Webseiten entwickeln und veröffentlichen

## 6.1 Was geschieht beim Abruf einer Webseite?

### 6.1.1 Adressen für Computer: IP-Adressen

### 6.1.2 Adressbuch für Computer: das Domain Name System (DNS)

### 6.1.3 Computer als Dienstleister im Internet: Host, Server und Client

### 6.1.4 Dienste auf einem Computer unterscheiden: Ports

## 6.2 Webserver auf dem eigenen Computer betreiben

### 6.2.1 Domainname für den eigenen Rechner: localhost

### 6.2.2 Lokalen Webserver starten

### 6.2.3 Das Dokument-Wurzelverzeichnis: Document-Root

### 6.2.4 Automatisch angezeigte Dateien: Index-Dateien

- 6.2.5 Entwicklungszyklus mit PHP auf dem Webserver
- 6.2.6 Infos über PHP anzeigen: `phpinfo()`
- 6.3 HTML-Grundgerüst
- 6.4 Anfragen und Antworten mit dem HTTP-Protokoll
  - 6.4.1 Browser-Konsole nutzen
  - 6.4.2 HTTP-Anfragen (Requests)
  - 6.4.3 Header im superglobalen Array `$_SERVER`
  - 6.4.4 HTTP-Antworten
  - 6.4.5 HTTP als zustandsloses Protokoll
- 6.5 Webspaces mieten und Webseite veröffentlichen
  - 6.5.1 Anbieter finden
  - 6.5.2 Webspaces in der Praxis
- 6.6 Übungen

## **7 Dynamische Webseiten und Formulare**

- 7.1 `$_GET`: Daten aus dem Query-String der URL
  - 7.1.1 Der Query-String in Hyperlinks
  - 7.1.2 Niemals Benutzereingaben vertrauen: Eingabevalidierung
- 7.2 Formulardaten im Query-String der URL übermitteln
  - 7.2.1 Formular und Auswertung in einem Skript zusammenfassen
  - 7.2.2 Niemals Eingabedaten vertrauen: Ausgabe-Maskierung
- 7.3 Einsatzgebiete der GET-Methode
- 7.4 `$_POST`: Formulardaten unsichtbar übermitteln
- 7.5 Vergleich zwischen GET und POST

## 7.6 Übungen

# 8 Mit Dateien arbeiten

## 8.1 Quelltext in mehreren Dateien strukturieren

### 8.1.1 Quelltext-Dateien einbinden und ausführen

### 8.1.2 Geltungsbereich von Variablen bei eingebundenen Dateien

### 8.1.3 Dateien außerhalb der Document-Root einbinden

## 8.2 Dateien schreiben und lesen

### 8.2.1 Dateien schreiben: file\_put\_contents()

### 8.2.2 Dateien lesen: file\_get\_contents()

### 8.2.3 Dateien als Array einlesen: file()

## 8.3 Dateien über das Internet laden

## 8.4 Datei-Uploads: \$\_FILES

### 8.4.1 \$\_POST und \$\_FILES

### 8.4.2 Uploads verarbeiten

### 8.4.3 Upload-Fehler

### 8.4.4 php.ini-Einstellungen

### 8.4.5 Datei-Art prüfen (MIME-Type)

## 8.5 Datei-Zugriffsrechte

### 8.5.1 Warum Zugriffsrechte?

### 8.5.2 Klassische Unix-Zugriffsrechte

### 8.5.3 Zugriffsrechte ändern

## 8.6 Übungen

# 9 Cookies und Sessions

## 9.1 Cookies



- 9.1.1 Wie entsteht ein Cookie?
- 9.1.2 Cookie-Daten lesen: \$\_COOKIE
- 9.1.3 Beispiel: Sprachwechsler
- 9.1.4 Cookies mit Ablaufdatum
- 9.1.5 Cookies löschen
- 9.1.6 Cookies im Browser verwalten
- 9.2 Sessions: Benutzersitzungen
  - 9.2.1 Eine Session starten
  - 9.2.2 Eine Session wieder aufnehmen
  - 9.2.3 Beispiel: Login
  - 9.2.4 Beispiel: »Flash«-Nachrichten
- 9.3 Übungen

## **10 Einstieg in die objektorientierte Programmierung (OOP)**

- 10.1 Grenzen der prozeduralen Programmierung
- 10.2 Grundbegriffe der »OOP«
  - 10.2.1 Klasse (Class)
  - 10.2.2 Objekt (Object)
  - 10.2.3 Eigenschaft (Property)
  - 10.2.4 Methode (Method)
  - 10.2.5 Die spezielle Variable \$this
  - 10.2.6 Konstruktor: \_\_construct()
  - 10.2.7 Sichtbarkeit (Visibility)
  - 10.2.8 Das Geheimnisprinzip: Daten kapseln
- 10.3 Vererbung

10.4 Ausnahmen (Exceptions)

10.5 Native Objektorientierung in PHP

10.5.1 Datum & Zeit: Prozedurale Programmierung

10.5.2 Datum & Zeit: Objektorientierte Programmierung

10.6 Übungen

## **11 Datenverwaltung mit MySQL**

11.1 Was ist eine relationale Datenbank?

11.2 Was ist ein Datenbankmanagementsystem?

11.3 Was ist SQL?

11.4 Erste Schritte mit MySQL

11.4.1 Warum MySQL?

11.4.2 MySQL installieren

11.4.3 Verbindung zu MySQL aufnehmen und erste Befehle ausführen

11.5 Kommentare

11.6 Datenbanken

11.6.1 Datenbank erzeugen

11.6.2 Datenbank löschen

11.7 Tabellen

11.7.1 Tabelle anlegen

11.7.2 Spaltentypen

11.7.3 Spaltenoptionen

11.7.4 Tabellenstruktur anzeigen

11.7.5 Tabellen löschen

11.8 Bezeichner

- 11.8.1 Gültige Bezeichner
- 11.8.2 Punkt-Notation
- 11.9 Daten schreiben, lesen, ändern und löschen
  - 11.9.1 Daten einfügen: INSERT INTO
  - 11.9.2 Daten abfragen: SELECT
  - 11.9.3 Daten ändern: UPDATE
  - 11.9.4 Daten löschen: DELETE
- 11.10 Übungen

## **12 Fortgeschrittene Datenbankabfragen mit MySQL**

- 12.1 Grafische Datenbankverwaltung mit phpMyAdmin
- 12.2 MySQL beim Webhoster
- 12.3 Fortgeschrittene Datenabfragen
  - 12.3.1 Vergleichsoperatoren
  - 12.3.2 Logische Verknüpfungen
  - 12.3.3 LIMIT: Abfrageergebnisse begrenzen
- 12.4 Alias-Namen für Tabellenspalten
- 12.5 SQL-Funktionen
  - 12.5.1 Skalare Funktionen
  - 12.5.2 Daten zusammenfassen: Aggregatfunktionen
- 12.6 Performance und Indizes
- 12.7 Übungen

## **13 PHP und MySQL kombinieren**

- 13.1 PHP mit MySQL verbinden
  - 13.1.1 Datenbankverbindung mit PDO
  - 13.1.2 Verbindung mit MySQL herstellen

- 13.1.3 SQL-Befehle ausführen
- 13.2 Beispiel: MySQL-Version abfragen
- 13.3 Beispiel: Benutzer-Accounts
  - 13.3.1 Benutzer-Tabelle anlegen
  - 13.3.2 Benutzer registrieren
  - 13.3.3 Login
  - 13.3.4 Login-Bereich
- 13.4 Passwörter sicher speichern
  - 13.4.1 Passwort-Hash erzeugen
  - 13.4.2 Passwort validieren
- 13.5 Sicherheitslücke: SQL-Injection
- 13.6 Übungen

## **14 Abschlussprojekt: Ein Blog programmieren**

- 14.1 Installation
- 14.2 Übersicht: Das fertige Blog
- 14.3 Verwendete Techniken
  - 14.3.1 Webdesign mit »Bootstrap«
  - 14.3.2 Methodenaufrufe verketteten
  - 14.3.3 Letzte eingefügte Datensatz-ID ermitteln
  - 14.3.4 POST-Anfrage erkennen
  - 14.3.5 Versteckte Formularfelder
  - 14.3.6 Composer
- 14.4 So funktioniert das Blog
  - 14.4.1 Verzeichnisstruktur
  - 14.4.2 Die verschiedenen Webseiten

- 14.4.3 Initialisierungsaufgaben
- 14.5 Anwendungsbeispiele
  - 14.5.1 Startseite mit Artikelübersicht
  - 14.5.2 Artikel anlegen
  - 14.5.3 Artikel als PDF downloaden
  - 14.5.4 Feedback als E-Mail versenden
- 14.6 Ausblick: Was kommt als Nächstes?
  - 14.6.1 Namespaces (Namensräume)
  - 14.6.2 Fortgeschrittene Entwicklungsumgebung mit XAMPP
  - 14.6.3 Reguläre Ausdrücke
  - 14.6.4 Frameworks
  - 14.6.5 PHP-Standardsoftware: Content Management und Co.
  - 14.6.6 Versionsverwaltung mit git und GitHub
  - 14.6.7 Testing mit PHPUnit
- 14.7 Übungen

# Einleitung

## E.1 Programmieren lernen in 14 Tagen

Mit diesem Buch haben Sie sich für einen einfachen, praktischen und fundierten Einstieg in die Welt der Programmierung entschieden. Sie lernen ohne unnötigen Ballast alles, was Sie wissen müssen, um PHP und MySQL effektiv für Projekte in Ihrem Berufs- und Interessensgebiet einzusetzen.

Wenn Sie Zeit genug haben, können Sie jeden Tag ein neues Kapitel durcharbeiten und so innerhalb von zwei Wochen Programmieren lernen. Alle Erklärungen sind leicht verständlich und setzen keine Vorkenntnisse voraus. Am besten lesen Sie das Buch neben der Computer-Tastatur und probieren die Programmbeispiele und Übungen gleich aus. Sie werden schnell erste Erfolge erzielen und Freude an der Programmierung finden.

## E.2 Der Aufbau des Buchs

Die Kapitel bauen Schritt für Schritt aufeinander auf. Das Buch beschreibt zunächst die Einsatzgebiete von PHP und die Einrichtung Ihres Computers zur Entwicklung des ersten PHP-Programms. Anschließend lernen Sie wichtige Sprachelemente und Einstellungen von PHP kennen. Sie werden vertraut mit der Verwendung von Funktionen, der Entwicklung und Veröffentlichung dynamischer Webseiten, dem Lesen und Schreiben von Dateien, der Anbindung eines externen Webservices sowie dem Einsatz von Cookies und Sessions. Nach einer Einführung in die objektorientierte

Programmierung erlernen Sie den Umgang mit einem MySQL-Datenbanksystem und der Datenbanksprache SQL. Das letzte Kapitel vereint alles Gelernte in einem Projekt zur Programmierung eines Weblogs und gibt Anregungen zur Weiterentwicklung Ihrer Programmierkenntnisse nach dem Schnelleinstieg.

Gelegentlich stoßen Sie auf kleine Aufgaben oder Zwischenfragen, die als Lernaktivierung gedacht sind. Ihr Tagespensum schließt mit praktischen Programmierübungen, in denen Sie Ihr neu gewonnenes Wissen vertiefen können. Die Lösungen zu diesen Übungen und die Antworten zu den Zwischenfragen stehen in einem Online-Kapitel zum Download zur Verfügung. Mehr dazu im nächsten Abschnitt.

Am Ende des Buchs finden Sie ein Stichwortverzeichnis, das Ihnen hilft, bestimmte Themen im Buch schneller zu finden.

## **E.3 Programmtexte und Lösungen zum Download**

Das Buch enthält viele kleine Beispielprogramme. Sie sind als »Starterprojekte« gedacht und sollen Sie ermuntern, den Code weiterzuentwickeln und selbst etwas Neues auszuprobieren.

Der Code aller Beispielprogramme sowie die Lösungen zu den Übungen und Zwischenfragen stehen Ihnen auf der Webseite des Verlags unter [www.mitp.de/0395](http://www.mitp.de/0395) zum Download zur Verfügung.

Dort finden Sie außerdem ein praktisches Glossar mit den wichtigsten Fachbegriffen.

## **E.4 Fragen und Feedback**



Unsere Verlagsprodukte werden mit großer Sorgfalt erstellt. Sollten Sie trotzdem einen Fehler bemerken oder eine andere Anmerkung zum Buch haben, freuen wir uns über eine direkte Rückmeldung an [lektorat@mitp.de](mailto:lektorat@mitp.de).

Falls es zu diesem Buch bereits eine Errata-Liste gibt, finden Sie diese unter [www.mitp.de/0395](http://www.mitp.de/0395) im Reiter DOWNLOADS.

Wir wünschen Ihnen viel Erfolg und Spaß bei der Programmierung mit PHP und MySQL!

*Philipp Rieber und das mitp-Lektorat*



# Erste Schritte mit PHP

Dieses Kapitel gibt eine praxisnahe Einführung in eine der populärsten Programmiersprachen des Internets: PHP. Um die 80 % aller Webseiten werden von PHP erzeugt. Das Spektrum reicht von Internetpräsenzen, Blogs, Portalen, Online-Shops und spezialisierten Web-Anwendungen bis zu Schnittstellen für die Datenverarbeitung von Mobile Apps und dem Internet of Things. PHP ist für Hobby-Anwender und den professionellen Einsatz in geschäftskritischen Softwaresystemen gleichermaßen geeignet. PHP-Kenntnisse eröffnen Ihnen die Welt hinter den graphischen Benutzeroberflächen des Internets und unzählige Möglichkeiten, um selbst privat oder beruflich in die Webentwicklung einzusteigen.

Nach einem Überblick zu den Einsatzgebieten von PHP führe ich Sie in diesem Kapitel zur erfolgreichen Ausführung Ihres ersten PHP-Programms auf dem eigenen Computer. Dabei erlernen Sie wichtige Grundkenntnisse und die Einrichtung einer Entwicklungsumgebung. Durch erste Programmbeispiele machen Sie sich »hands-on« an der Tastatur Ihres Computers mit den Grundeigenschaften von PHP vertraut. Zum Abschluss des Kapitels erhalten Sie einen Überblick zur Entstehungsgeschichte von PHP.

Wo vorhanden, verwendet dieses Buch deutsche



Fachbegriffe. Da die englischen Entsprechungen für Recherchen, Fehlersuchen oder in der Kommunikation mit anderen Programmierern unerlässlich sind, mache ich Sie nebenbei auch mit den englischen Begriffen vertraut.

## 1.1 Wofür wird PHP eingesetzt?

PHP ist eine kostenlose, universell einsetzbare Programmiersprache. Sieht man von Nischen wie der Programmierung von Alexa Skills oder Desktop-Programmen ab, konzentriert sich der Einsatz auf drei Einsatzgebiete:

- Erzeugung dynamischer Webseiten
- Bereitstellung von Webservices
- Kommandozeilenprogramme

Die nächsten Abschnitte erklären Grundlagen zu den verschiedenen Gebieten. Stellen Sie sich zur Veranschaulichung eine fiktive Zeitungsredaktion vor, die eine Präsenz im Internet aufbaut. Schrittweise entwickelt sich die Internetpräsenz von einer reinen Text-Webseite über eine ansehnlichere HTML-Webseite zu einer fortschrittlichen dynamischen Webseite. Anschließend veröffentlicht die Zeitung ihre eigene Mobile App und automatisiert wiederkehrende Aufgaben.

### 1.1.1 PHP zur Erzeugung dynamischer Webseiten

Was ist der Unterschied zwischen einer *statischen* und einer *dynamischen* Webseite? Der Abruf einer statischen Webseite von einer Internetadresse im Web-Browser liefert das immer gleiche, »statische« Ergebnis. Eine dynamische Webseite hingegen wird erst im Zuge des Abrufs erzeugt. Dabei

werden Inhalte aus verschiedenen Quellen wie Datenbanken oder externen Webdiensten zusammengetragen und zur Anzeige aufbereitet. Eingaben des Benutzers oder dessen Kontext (Identität, Standort, Tageszeit etc.) können den Inhalt beeinflussen.

Eine fiktive Zeitungsredaktion schreibt für ihre ersten Schritte zu einer Internetpräsenz alle Artikel in die einfache Textdatei `articles.txt` und veröffentlicht sie auf einem Computer im Internet, dem *Webserver*.



**Abb. 1.1:** Die Textdatei `articles.txt` im Text-Editor

Interessierte Leser rufen die Datei anhand der passenden Internetadresse (*URL*, Uniform Resource Locator) in einem Browser auf. Die Datei wird vom Webserver auf den eigenen Computer, den *Client*, übertragen und im Browserfenster angezeigt. Solange die Redaktion die Text-Datei nicht durch eine aktualisierte Version ersetzt, führt jeder weitere Aufruf zur immer gleichen, »statischen« Anzeige des Inhalts – auch für jeden anderen Besucher.



**Abb. 1.2:** Statische Text-Webseite `articles.txt` im Browser

Da die Gestaltungsmöglichkeiten mit reinem Text begrenzt sind, wechselt die Redaktion auf die Nutzung der

## Auszeichnungssprache HTML (*Hypertext Markup Language*).

A screenshot of a text editor window titled 'articles.html - Bearbeiten'. The code inside the editor is as follows:

```
<h1>Die Zeitung – Neueste Nachrichten</h1>
<ul>
  <li>»PHP &amp; MySQL Schnelleinstieg« erschienen! <a href="https://www.php-mysql-schnelleinstieg.de">Mehr ...</a></li>
  <li>ElePHPant entdeckt! <a href="https://www.php.net/elephant.php">Mehr ...</a></li>
  <li>PHP 8 veröffentlicht! <a href="https://www.php.net/releases/8.0/de.php">Mehr ...</a></li>
</ul>
```

**Abb. 1.3:** Die HTML-Datei articles.html im Text-Editor

HTML ermöglicht die Strukturierung der Inhalte mit Hilfe von maschinenlesbaren Hinweisen, den *HTML-Tags*. Die HTML-Tags markieren eingeschlossene Inhalte dabei mit einer gewünschten Bedeutung, zum Beispiel Überschrift, Link, Liste etc. Dies nennt man *semantische* Strukturierung. Die Redaktion verwendet im Beispiel Elemente für eine Überschrift ersten Grades (*heading 1* = `h1`), Hyperlinks (*anchor* = `a`) und eine ungeordnete Liste (*unordered list* = `ul`) mit Listenelementen (*list item* = `li`). Die Auszeichnungen beginnen mit einem öffnenden Tag `<element>` und enden mit einem schließenden Tag

`</element>`:

```
<h1>Überschrift ersten Grades</h1>
<a href="https://www.google.de">Link zu Google</a>
<ul>
  <li>Erstes Listenelement</li>
  <li>Zweites Listenelement</li>
</ul>
```

HTML-Tags bleiben für den menschlichen Betrachter unsichtbar. Der Browser versteht jedoch die versteckten Auszeichnungen, stellt die Inhalte entsprechend dar und schafft einfache Interaktion durch klickbare Links.

# Die Zeitung - Neueste Nachrichten

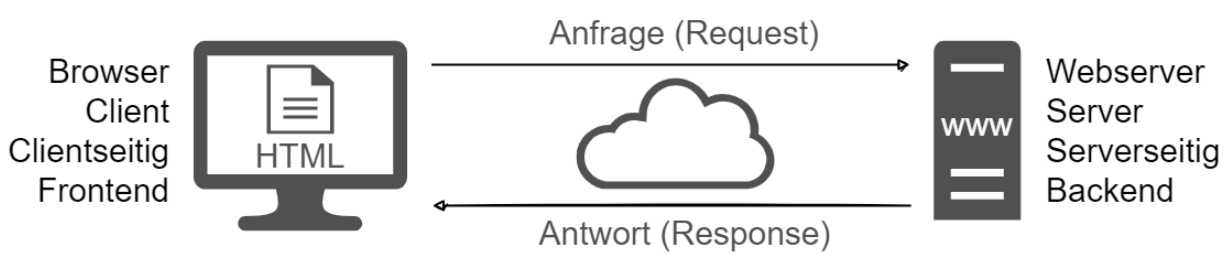
- »PHP & MySQL Schnelleinstieg« erschienen! [Mehr ...](#)
- ElePHPant entdeckt! [Mehr ...](#)
- PHP 8 veröffentlicht! [Mehr ...](#)

**Abb. 1.4:** Statische HTML-Webseite articles.html im Browser



Eine genaue Kontrolle über Formatierungen (Farben, Schriftgrößen, Positionierungen usw.) ermöglicht die ergänzende Formatierungssprache *CSS* (Cascading Style Sheets). Die CSS-Formatierungsangaben sind für den Betrachter ebenso unsichtbar wie HTML-Tags, der Browser nutzt sie jedoch zur Anpassung der Darstellung. Mehr zu HTML und CSS erfahren Sie z.B. unter <https://wiki.selfhtml.org>.

Abbildung 1.5 zeigt den Kreislauf aus Anfrage des Browsers an einen Webserver und dessen Antwort. Dieser Kreislauf wiederholt sich bei jeder angefragten Webseite.



**Abb. 1.5:** Der Kreislauf aus HTTP-Anfrage und -Antwort

Bald kommt in der Zeitungsredaktion eine neue Idee auf: Sie möchte das aktuelle Tagesdatum ohne tägliche manuelle Bearbeitung einblenden. Doch HTML kann keine Inhalte erzeugen und hat keinen Zugriff auf eine Uhr mit dem

aktuellen Datum. Für diesen Zweck ist Programmierlogik erforderlich. Die Zeitung engagiert eine Webagentur. Die Agentur aktiviert PHP auf dem Webserver, benennt `articles.html` in `articles.php` um und beginnt PHP-Programmlogik zur Anzeige des aktuellen Datums in das HTML einzubetten:

```
<h1>Nachrichten</h1>
<p>Heute ist der <?php echo date('d.m.Y'); ?>!</p>
<ul>...</ul>
```

Im Gegensatz zur HTML-Datei liefert der Webserver die PHP-Datei nicht direkt an den Browser zurück, sondern lässt zunächst den enthaltenen PHP-Programmcode ausführen. Alle PHP-Bereiche werden durch die dabei generierten Ausgaben ersetzt.

<code>&lt;?php</code>	Hier beginnt PHP-Programmlogik
<code>echo</code>	Anweisung zur Ausgabe
<code>date('d.m.Y')</code>	Aufruf der in PHP eingebauten Funktion <code>date()</code> , die das aktuelle Datum in einem gewünschten Format liefert. Im Beispiel wird das Format <code>d.m.Y = day.month.Year = Tag.Monat.Jahr</code> verwendet.
<code>;</code>	Ende der Anweisung
<code>?&gt;</code>	Hier endet PHP-Programmlogik

Aus dem bestehenden HTML und den durch PHP dynamisch ergänzten Inhalten ergibt sich die gewünschte Webseite, die an den Browser zurückgeschickt wird. Bei einem Abruf der Webseite am 13. März 2022 lautet der generierte Inhalt:

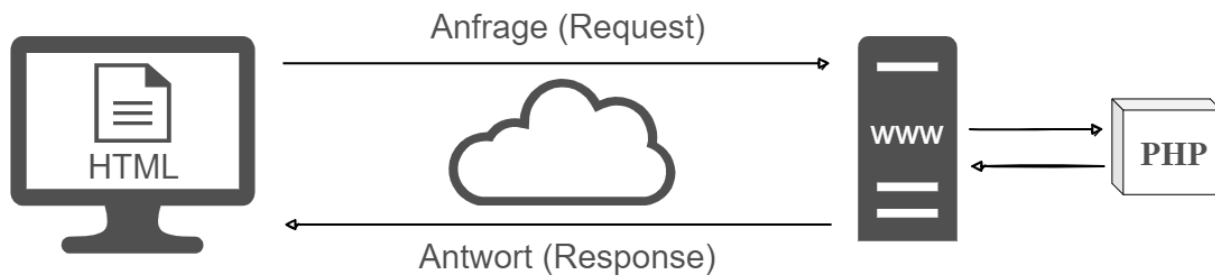
```
<h1>Nachrichten</h1>
<p>Heute ist der 13.03.2022!</p>
<ul>...</ul>
```

## Aufgabe 1



Können Sie die Dokumentation zur PHP-Funktion `date()` auf <https://www.php.net> finden?

Aus Sicht des Browsers auf dem eigenen Computer, des Clients, erscheint die empfangene Webseite genauso statisch wie zuvor. Die dynamische Erzeugung erfolgte bereits *serverseitig* auf dem entfernten Webserver. Eine Installation von PHP ist daher nur auf dem Webserver erforderlich, nicht auf den Computern der Webseiten-Besucher. Der Browser kümmert sich wie zuvor nur um die Darstellung, unabhängig von der Entstehung des Inhalts.



**Abb. 1.6:** Der Kreislauf aus Anfrage und Antwort mit PHP

Die Pflege der Zeitungsartikel in der Datei durch die Redakteure erfordert HTML-Kenntnisse, Absprachen zwischen den Redakteuren und ständige Übertragungen neuer Versionen auf den Webserver. Mit fortgeschrittenen Methoden der PHP-Entwicklung kann die Webagentur den nächsten Wunsch der Zeitungsredaktion realisieren: Eine Vereinfachung der Artikel-Verwaltung, die von den Redakteuren keine technischen Kenntnisse mehr erfordert. Die Artikelinhalte werden nicht länger in der Datei `articles.php` gepflegt, sondern in eine Datenbank ausgelagert. PHP kann die Artikel zum Zeitpunkt des Abrufs der Webseite aus der Datenbank lesen und ähnlich wie zuvor das Tagesdatum dynamisch in das HTML einbauen. Weitere Abrufe der Webseite wiederholen die Generierung der Inhalte und damit den Abruf der Artikel aus der