

Philipp Rieber

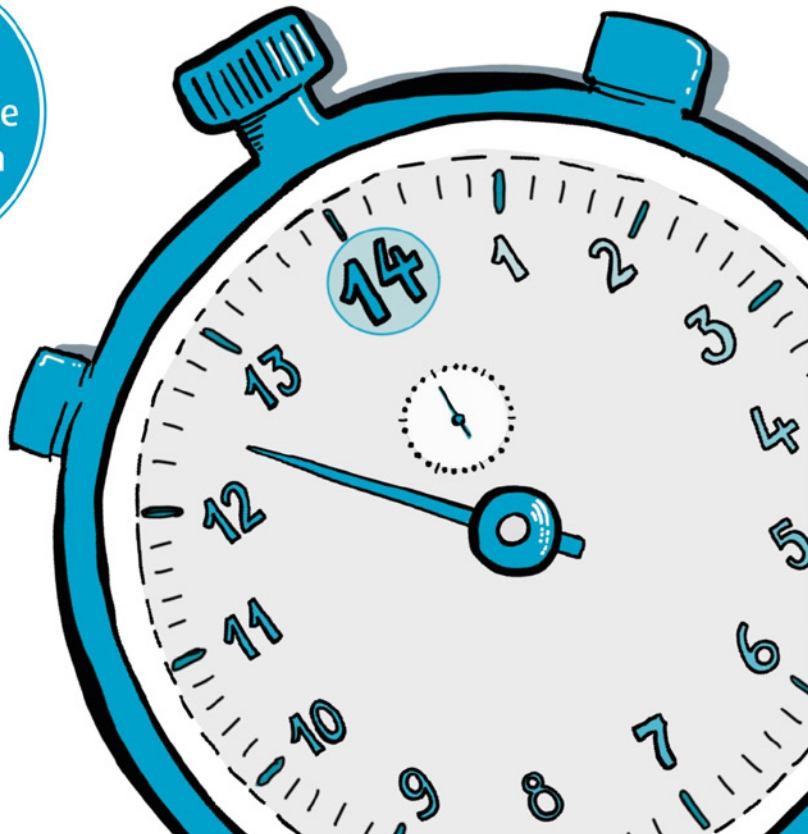
12,99 €
inkl. E-Book

PHP & MySQL Schnelleinstieg

Programmieren lernen in 14 Tagen

> Einfach und ohne Vorkenntnisse <

Zahlreiche
Praxisbeispiele
und Übungen



Inhalt

Einleitung

E.1	Programmieren lernen in 14 Tagen	11
E.2	Der Aufbau des Buchs	11
E.3	Programmtexte und Lösungen zum Download	12
E.5	Fragen und Feedback	12



Erste Schritte mit PHP

1.1	Wofür wird PHP eingesetzt?	13
1.2	Stärken von PHP	22
1.3	Schwächen von PHP	23
1.4	Was ist für die Entwicklung mit PHP notwendig?	24
1.5	Quelltext und PHP-Interpreter	25
1.6	Kommandozeile nutzen	26
1.7	PHP installieren	31
1.8	Quelltext-Editor verwenden: Visual Studio Code	36
1.9	Das erste PHP-Skript ausführen	38
1.11	Wie entstand PHP?	41
1.12	Übungen	42



Variablen, Datentypen und Konstanten

2.1	Daten in einer Variablen erfassen	43
2.2	Einfache Datentypen	50
2.3	Der spezielle Datentyp: NULL	58
2.4	Der vielseitige Datentyp: Array	59
2.5	Datentypen umwandeln	63
2.6	Programmstrukturen	66

2.7	Konstanten	67
2.8	Übungen	70



Programmablauf mit Kontrollstrukturen steuern

3.1	Programmablauf verzweigen	73
3.2	Bedingungen formulieren	81
3.3	Programmabschnitte mit Schleifen wiederholen	85
3.4	Alternative Syntax für Kontrollstrukturen	94
3.5	Übungen	95



Programmierfehler und PHP-Konfiguration

4.1	Programmierfehler	97
4.2	PHP konfigurieren: die php.ini-Datei	99
4.3	Fehlerstufen	102
4.4	Fehlersichtbarkeit einstellen	103
4.5	Übungen	104



Funktionen

5.1	Native Funktionen aus der PHP-Bibliothek verwenden	105
5.2	Eigene Funktionen definieren	122
5.3	Übungen	127



Webseiten entwickeln und veröffentlichen

6.1	Was geschieht beim Abruf einer Webseite?	129
6.2	Webserver auf dem eigenen Computer betreiben	132
6.3	HTML-Grundgerüst	138
6.4	Anfragen und Antworten mit dem HTTP-Protokoll	140
6.5	Webspace mieten und Webseite veröffentlichen	146
6.6	Übungen	148



Dynamische Webseiten und Formulare

7.1	\$_GET: Daten aus dem Query-String der URL	149
7.2	Formulardaten im Query-String der URL übermitteln	158
7.3	Einsatzgebiete der GET-Methode	165
7.4	\$_POST: Formulardaten unsichtbar übermitteln	166
7.5	Vergleich zwischen GET und POST	169
7.6	Übungen	170



Mit Dateien arbeiten

8.1	Quelltext in mehreren Dateien strukturieren	173
8.2	Dateien schreiben und lesen	180
8.3	Dateien über das Internet laden	182
8.4	Datei-Uploads: \$_FILES	184
8.5	Datei-Zugriffsrechte	190
8.6	Übungen	192



Cookies und Sessions

9.1	Cookies	195
9.2	Sessions: Benutzersitzungen	202
9.3	Übungen	210



Einstieg in die objektorientierte Programmierung (OOP)

10.1	Grenzen der prozeduralen Programmierung	211
10.2	Grundbegriffe der »OOP«	212
10.3	Vererbung	221
10.4	Ausnahmen (Exceptions)	224
10.5	Native Objektorientierung in PHP	227
10.6	Übungen	229



Datenverwaltung mit MySQL

11.1	Was ist eine relationale Datenbank?	231
11.2	Was ist ein Datenbankmanagementsystem?	232
11.3	Was ist SQL?	233
11.4	Erste Schritte mit MySQL	234
11.5	Kommentare	241
11.6	Datenbanken	241
11.7	Tabellen	243
11.8	Bezeichner	247
11.9	Daten schreiben, lesen, ändern und löschen	248
11.10	Übungen	253



Fortgeschrittene Datenbankabfragen mit MySQL

12.1	Grafische Datenbankverwaltung mit phpMyAdmin	255
12.2	MySQL beim Webhoster	259
12.3	Fortgeschrittene Datenabfragen	260
12.4	Alias-Namen für Tabellenspalten	264
12.5	SQL-Funktionen	265
12.6	Performance und Indizes	269
12.7	Übungen	272



PHP und MySQL kombinieren

13.1	PHP mit MySQL verbinden	273
13.2	Beispiel: MySQL-Version abfragen	278
13.3	Beispiel: Benutzer-Accounts	278
13.4	Passwörter sicher speichern	284
13.5	Sicherheitslücke: SQL-Injection	285
13.6	Übungen	288



Abschlussprojekt: Ein Blog programmieren

14.1	Installation	289
14.2	Übersicht: Das fertige Blog	290
14.3	Verwendete Techniken	291
14.4	So funktioniert das Blog	296
14.5	Anwendungsbeispiele	301
14.6	Ausblick: Was kommt als Nächstes?	308
14.7	Übungen	311

Stichwortverzeichnis

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 zum Download zur Verfügung.

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

E.5 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.

Falls es zu diesem Buch bereits eine Errata-Liste gibt, finden Sie diese unter 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.

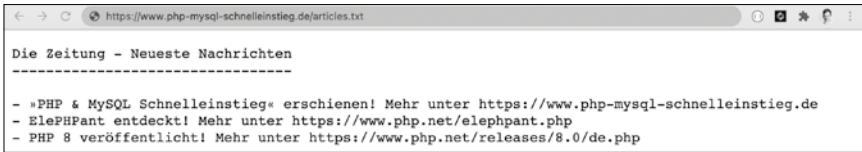


Abb. 1.2: Statische Text-Webseite articles.txt im Browser

Da die Gestaltungsmöglichkeiten mit reinem Text begrenzt sind, wechselt die Redaktion auf die Nutzung der Auszeichnungssprache HTML (*Hypertext Markup Language*).



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.

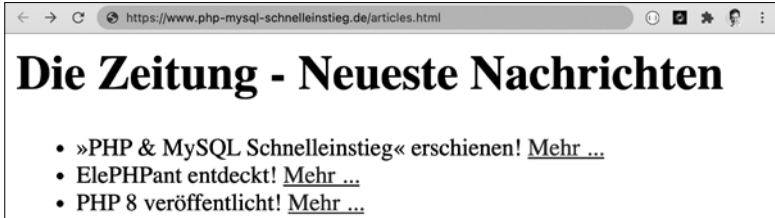


Abb. 1.4: Statische HTML-Webseite articles.html im Browser



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.

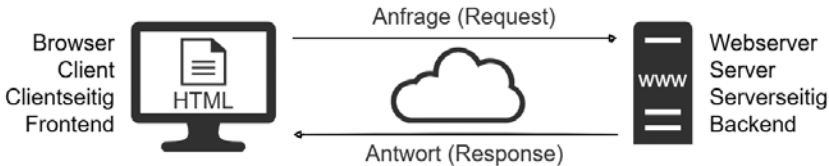


Abb. 1.5: Der Kreislauf aus HTTP-Anfrage und -Antwort

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.

<?php	Hier beginnt PHP-Programmlogik
echo	Anweisung zur Ausgabe
date('d.m.Y')	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.
;	Ende der Anweisung
?>	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.

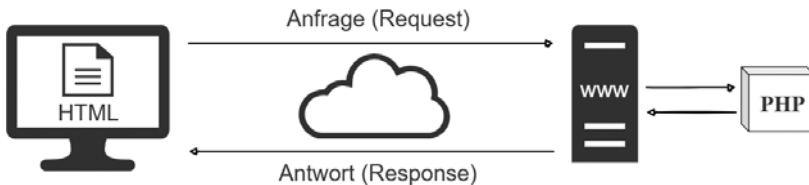


Abb. 1.6: Der Kreislauf aus Anfrage und Antwort mit PHP

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 Datenbank. Neu in die Datenbank eingepflegte Artikel erscheinen somit automatisch auf der Webseite.

Zur Erstellung neuer Artikel durch Redakteure ergänzt die Webagentur einen durch Login geschützten Bereich mit einem Eingabeformular für neue Artikel. PHP überträgt die Eingaben in die Datenbank. Außer der Browserbedienung benötigen die Redakteure keine weiteren technischen Kenntnisse.

Aufgabe 2

Was geschieht, wenn Sie eine PHP-Datei direkt im Browser öffnen? Ziehen Sie zum Ausprobieren eine PHP-Datei in das Browserfenster (Drag & Drop). Vergleichen Sie mit Dateien vom Typ TXT, HTML, ZIP und PDF.

1.1.2 PHP zur Bereitstellung von Webservices

Die fiktive Zeitungsredaktion aus dem vorigen Abschnitt stellt einen großen Anteil an Smartphone-Besuchern auf ihrer Webseite fest. Sie erweitert das Angebot daher um eine eigene Mobile App. Diese folgt bezüglich Gestaltung und Bedienung anderen Gesetzen als eine Webseite für den Browser. Die App ist nicht am Abruf einer HTML-Webseite interessiert, sondern nur an den reinen Artikeldaten, um damit die Gestaltungselemente einer Mobile App zu füllen. In der ersten Version baut die Webagentur einen einfachen »Nachrichtenticker« zur Anzeige der Artikel mit Datum und Überschrift, der sich durch Drücken einer Schaltfläche aktualisieren lässt.

Die Programmierung der Mobile App selbst erfolgt nicht in PHP, sondern in einer Programmiersprache für die jeweilige Plattform, z.B. Java für Android- oder Swift für iPhone-Apps. Wie die meisten Apps muss sie im Hintergrund in Kontakt mit einem zentralen Service stehen, um aktuelle Daten abzurufen

oder Benutzereingaben dorthin zu senden. Den unsichtbaren Kommunikationspartner im Hintergrund bezeichnet man als *_Backend_*. Eine dem Benutzer zugewandte Anwendung wie die Webseite oder die Mobile App heißt *_Frontend_*.

Zur Kommunikation greift die App (das Frontend) ähnlich dem Browser anhand einer URL auf einen Webserver (das Backend) zu, etwa zum Abruf der neuesten Artikel beim Start der App oder bei einer gewünschten Aktualisierung. Statt einer HTML-Webseite mit vielen irrelevanten Bereichen ist die App an einem maschinenlesbaren Format zum Datenaustausch interessiert. Ein verbreiteter Standard dafür ist *JSON* (ausgesprochen: Jason). Die übermittelten Daten für den Nachrichten-Ticker könnten im JSON-Format so aussehen:

```
[
  {
    "title": "ElePHPant entdeckt!",
    "link": "https://www.php.net/elephpant"
  },
  {
    "title": "PHP 8 veröffentlicht!",
    "link": "https://www.php.net/releases/8.0/de"
  },
]
```

Da sich mit PHP neben HTML auch jede andere Art von Ausgabe erzeugen lässt, kann PHP der Mobile App als Backend dienen. Statt HTML gibt das PHP-Skript JSON aus:

```
<?php
$articles = Abruf der Artikel aus der Datenbank;
echo json_encode($articles);
?>
```

<?php	Beginn der PHP-Programmlogik
<code>\$articles = ...</code>	Definiert die Variable <code>\$articles</code> – eine Art Daten-Container für die Zeitungsartikel. Die Variable ist eine Referenz zum späteren Zugriff auf die Daten.
<i>Abruf der Artikel aus Datenbank</i>	Steht stellvertretend für Programmcode zum Abruf der Daten aus einer Datenbank.

;	Ende der Anweisung
echo	Anweisung zur Ausgabe
json_encode(\$articles)	Eine in PHP eingebaute Funktion, die die übergebenen Zeitungsartikel-Daten im JSON-Format zurückliefert.
;	Ende der Anweisung
?>	Ende der PHP-Programmlogik

Mit Hilfe von PHP stellt der Webserver die Artikel als Service über das Internet zur Verfügung – als *Webservice*. Im Vergleich zu einer Webseite für menschliche Benutzer richtet sich das Angebot eines Webservice an andere Software – unabhängig vom endgültigen Verwendungszweck.

Später entscheidet sich die Zeitung zur Veröffentlichung ihres Webservice. Ein Newsletter-Versender kann die verfügbaren Daten zum Beispiel unabhängig von der Zeitungs-Webseite nutzen, um seinen wöchentlichen E-Mail-Newsletter automatisiert mit Inhalten der Zeitung anzureichern. In umgekehrter Weise erweitert die Zeitung ihr eigenes Angebot um einen Wetterbericht, dessen Daten sie vom Webservice eines Drittanbieters abrufen.



Ein Webservice wird oft auch technischer als *Web-API* (Web Application Programming Interface) bezeichnet – eine Programmierschnittstelle über das Internet.

Weitere Beispiele für Webservices beziehungsweise Web-APIs: Währungs- und Aktienkurse, Fahrpläne, SMS- und E-Mail-Versand, Daten zu Sportereignissen, Steuerung von Smart-Home-Geräten, Bonitätsauskünfte, Validierung von Postadressen und vieles mehr. Auch die großen Tech-Unternehmen wie Twitter, Facebook, Google, Amazon etc. stellen Web-APIs zur Integration in eigene Angebote bereit.

Mit Hilfe von PHP können Sie sowohl eigene Webservices bereitstellen (*produzieren*) als auch fremde Webservices nutzen (*konsumieren*). Mit entsprechenden Maßnahmen kann ein Webservice auch gegen Bezahlung angeboten werden.

1.1.3 Kommandozeilenprogramme mit PHP

Ein Kommandozeilenprogramm stellt die einfachste Form eines PHP-Programms dar. Ist PHP auf einem Computer installiert, kann ein solches Programm darauf ausgeführt werden. Ein Browser oder Webserver spielt dabei keine Rolle, die Ausführung des Programms erfolgt rein *clientseitig* bezie-