

---

Ralph Steyer

1. Ausgabe, Juli 2022

ISBN: 978-3-98569-093-0

# Objektorientierte Programmierung mit PHP 8

OOP-PHP8



**HERDT**

---

# Impressum

Matchcode: OOP-PHP 8

Autor: Ralph Steyer

Produziert im HERDT-Digitaldruck

1. Ausgabe, Juli 2022

HERDT-Verlag für Bildungsmedien GmbH  
Am Kuemmerling 19  
55294 Bodenheim  
Internet: [www.herdtd.com](http://www.herdtd.com)  
E-Mail: [info@herdtd.com](mailto:info@herdtd.com)

© HERDT-Verlag für Bildungsmedien GmbH, Bodenheim

Alle Rechte vorbehalten. Kein Teil des Werkes darf in irgendeiner Form (Druck, Fotokopie, Mikrofilm oder einem anderen Verfahren) ohne schriftliche Genehmigung des Verlags reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

Dieses Buch wurde mit großer Sorgfalt erstellt und geprüft. Trotzdem können Fehler nicht vollkommen ausgeschlossen werden. Verlag, Herausgeber und Autoren können für fehlerhafte Angaben und deren Folgen weder eine juristische Verantwortung noch irgendeine Haftung übernehmen.

Wenn nicht explizit an anderer Stelle des Werkes aufgeführt, liegen die Copyrights an allen Screenshots beim HERDT-Verlag. Sollte es trotz intensiver Recherche nicht gelungen sein, alle weiteren Rechteinhaber der verwendeten Quellen und Abbildungen zu finden, bitten wir um kurze Nachricht an die Redaktion.

Aus Gründen der besseren Lesbarkeit wird auf die gleichzeitige Verwendung der Sprachformen männlich, weiblich und divers (m/w/d) verzichtet. Sämtliche Personenbezeichnungen gelten gleichermaßen für alle Geschlechter.

Die in diesem Buch und in den abgebildeten bzw. zum Download angebotenen Dateien genannten Personen und Organisationen, Adress- und Telekommunikationsangaben, Bankverbindungen etc. sind frei erfunden. Eventuelle Übereinstimmungen oder Ähnlichkeiten sind unbeabsichtigt und rein zufällig.

Die Bildungsmedien des HERDT-Verlags enthalten Verweise auf Webseiten Dritter. Diese Webseiten unterliegen der Haftung der jeweiligen Betreiber, wir haben keinerlei Einfluss auf die Gestaltung und die Inhalte dieser Webseiten. Bei der Bucherstellung haben wir die fremden Inhalte daraufhin überprüft, ob etwaige Rechtsverstöße bestehen. Zu diesem Zeitpunkt waren keine Rechtsverstöße ersichtlich. Wir werden bei Kenntnis von Rechtsverstößen jedoch umgehend die entsprechenden Internetadressen aus dem Buch entfernen.

Die in den Bildungsmedien des HERDT-Verlags vorhandenen Internetadressen, Screenshots, Bezeichnungen bzw. Beschreibungen und Funktionen waren zum Zeitpunkt der Erstellung der jeweiligen Produkte aktuell und gültig. Sollten Sie die Webseiten nicht mehr unter den angegebenen Adressen finden, sind diese eventuell inzwischen komplett aus dem Internet genommen worden oder unter einer neuen Adresse zu finden. Sollten im vorliegenden Produkt vorhandene Screenshots, Bezeichnungen bzw. Beschreibungen und Funktionen nicht mehr der beschriebenen Software entsprechen, hat der Hersteller der jeweiligen Software nach Drucklegung Änderungen vorgenommen oder vorhandene Funktionen geändert oder entfernt.

<b>1 Informationen zu diesem Buch</b>	<b>4</b>	<b>6 Abstrakte Klassen und Schnittstellen</b>	<b>82</b>
1.1 Voraussetzungen und Ziele		6.1 Abstrakte Klassen	82
1.2 Bevor Sie beginnen ...	7	6.2 Schnittstellen	88
		6.3 Abstrakte Techniken im Praxisprojekt	91
		6.4 Übungen	98
<b>2 Das Umfeld von PHP</b>	<b>8</b>	<b>7 Eine objektorientierte Webseite</b>	<b>99</b>
2.1 Webseiten und Webanwendungen	8	7.1 Die Weiterentwicklung des Praxisprojekts	99
2.2 PHP	10	7.2 Die Programmdatei	102
2.3 Übungen	12	7.3 Übung	104
<b>3 Die Idee der objektorientierten Programmierung</b>	<b>15</b>	<b>8 Objektorientierter Datenbankzugriff</b>	<b>105</b>
3.1 PHP und OOP	15	8.1 PHP und Datenbanken	105
3.2 Was ist objektorientierte Programmierung?	16	8.2 Die moderne Art des Datenbankzugriffs – PDO	107
3.3 Die Kernkonzepte der Objektorientierung	18	8.3 Abfragen senden	111
		8.4 Prepared Statements	118
		8.5 Das Praxisprojekt mit Datenbanken	121
		8.6 Fazit und Abschlussbemerkungen	123
		8.7 Übung	124
<b>4 Klassen, Objekte und Konstruktoren</b>	<b>20</b>	<b>A Anhang: Installationen und Quellangaben</b>	<b>126</b>
4.1 Eine Klasse – „Bauplan“ für Objekte	20	A.1 Programme und Tools	126
4.2 Klassen in PHP deklarieren	21	A.2 Programmieren und Debuggen mit Xdebug und PDT	133
4.3 Eigenschaften	23	A.3 Quellangaben im Internet	142
4.4 Methoden	24		
4.5 Der Zugriff auf Instanzelemente	28	<b>Stichwortverzeichnis</b>	<b>144</b>
4.6 Datenkapselung, Getter- und Setter-Methoden	29		
4.7 Objekte erzeugen und der Konstruktor	31		
4.8 Ein erster OO-Aufbau einer Webseite	36		
4.9 Objekte löschen – Garbage Collector und Destruktor	38		
4.10 Objekte klonen	43		
4.11 Anonyme Klassen	46		
4.12 Klassenmember und der Gültigkeitsbereichsoperator	47		
4.13 Autoloading	51		
4.14 PHP 8.x: Was ist neu und was wurde geändert?	53		
4.15 Übungen	58		
<b>5 Vererbung</b>	<b>59</b>		
5.1 Worum geht es bei Vererbung?	59		
5.2 Die konkrete Umsetzung der Vererbung in PHP	63		
5.3 Vererbung im Praxisprojekt	65		
5.4 Der Sichtbarkeitsmodifizierer <code>protected</code>	74		
5.5 Überschreiben und Verdecken	78		
5.6 Übungen	80		

# 1

## Informationen zu diesem Buch

### 1.1 Voraussetzungen und Ziele

#### Zielgruppe

- ✓ Studenten und Auszubildende in IT-Berufen
- ✓ Schüler mit IT-Schwerpunkt
- ✓ Fortgeschrittene PHP-Einsteiger und Web-Programmierer mit PHP-Kenntnissen
- ✓ Erfahrenere PHP-Entwickler älterer PHP-Versionen (ohne OO)
- ✓ Fortgeschrittene Entwickler anderer Programmiersprachen

#### Empfohlene allgemeine Vorkenntnisse

- ✓ Grundkenntnisse im Umgang mit Windows, Linux oder macOS
- ✓ Grundkenntnisse in der Bedienung von Anwendungsprogrammen
- ✓ Grundkenntnisse im Umgang mit dem Internet
- ✓ Allgemeines Grundwissen in Programmierung
- ✓ Grundlagen-Kenntnisse in PHP

#### Erforderliche PHP-Grundkenntnisse

- ✓ Grundlegende Sprachelemente
- ✓ Variablen und Operatoren
- ✓ Kontrollstrukturen
- ✓ Felder (Arrays)
- ✓ Funktionen
- ✓ Kenntnisse von Super-Globals (standardmäßig in PHP vorhandene Variablen)
- ✓ Interaktive Webseiten (Auswertung von Formularen mithilfe von PHP)
- ✓ Arbeit mit externen Dateien
- ✓ Umgang mit Sessions
- ✓ Zeichenkettenfunktionen
- ✓ Klassische Anwendungen von PHP wie der Umgang mit Datums- und Zeitfunktionen

## Lernziele

Lernziele des Buches sind das Erarbeiten von objektorientierter Programmierung mit PHP und deren Anwendung in einigen Praxis-Situationen. Folgende Themen werden behandelt:

- ✓ Sie verstehen den Ansatz der objektorientierten Denkweise und der objektorientierten Programmierung.
- ✓ Sie erlernen die einzelnen Techniken und Terminologien der objektorientierten Programmierung.
- ✓ Sie wissen, wie man mit PHP die objektorientierte Programmierung umsetzen kann.
- ✓ Sie können einschätzen, wann ein objektorientierter Ansatz bei PHP sinnvoll ist und wann man hybrid oder gänzlich funktional bzw. prozedural vorgehen sollte.
- ✓ Sie lernen verschiedene objektorientierte PHP-Anwendungen wie etwa den Datenbankzugriff mit PDO (PHP Data Objects) und die Umsetzungen einer kompletten Webseite mit OOP kennen.

## Hinweise zur Hard- und Software

Als Hardwarebasis wird ein normaler PC vorausgesetzt. Im vorliegenden Buch werden eine Windows-10 bzw. 11-Umgebung sowie lokale Installationen der folgenden kostenfreien Software verwendet:

- ✓ XAMPP (Webserver, PHP, MySQL/MariaDB; <http://www.apachefriends.org>)
- ✓ Notepad++ (Editor; <http://notepad-plus.sourceforge.net>)
- ✓ FileZilla (FTP-Client; <http://filezilla-project.org>)
- ✓ Verschiedene Standardbrowser (Chrome, Firefox, Edge bzw. Internet Explorer)

Eine Kurzübersicht sowie eine Installations- und Konfigurationsanleitung für die genannte Software finden Sie im Anhang. Alternativ können Sie die verwendeten PHP-Skripte auch mit funktionsähnlicher alternativer Software erstellen und testen. Neben dem im Buch als Vorgabe verwendeten Betriebssystem werden Linux und MacOS X bei Bedarf ebenso berücksichtigt. Bei einigen Ausblicken werden ebenso bei Bedarf verschiedene weitere Programme und Tools (insbesondere Entwicklungstools bzw. IDEs) verwendet oder erwähnt.

## Anmerkungen zur PHP-Version

Die Basis des Buches ist PHP in der Version 8 (8.1.x). Bemerkenswert an neuen Versionen von PHP ab der Version 7 ist, dass damit verschiedene Elemente der vorherigen Versionen nicht nur als deprecated – veraltet – erklärt, sondern explizit abgeschafft wurden. Üblicherweise enthalten neue Sprachversionen auf der einen Seite gewisse Erweiterungen. Auf der anderen Seite erklären neue Versionen einer Programmiersprache fast immer gewisse vorhandene Elemente als „deprecated“. Das bedeutet, dass diese zwar nicht mehr eingesetzt werden sollen, aber prinzipiell noch funktionieren.

Wenn man hingegen in einer neuen Sprachversion Elemente explizit **entfernt**, ist solch ein Update der Sprache nicht mehr zu 100 % abwärtskompatibel. Das bedeutet, dass Quellcode, der mit Sprachelementen einer älteren Version erstellt wurde, unter der neuen Version u. U. nicht mehr funktioniert.

Da solch ein Schritt massive Auswirkungen auf bestehende Projekte hat, wird so eine Maßnahme nur selten durchgeführt. Um PHP mit der Version 7 auf einen sicheren, stabilen und professionellen Stand zu heben, wurde dieser radikale Schritt jedoch notwendig. Vorherige Updates hatten diesen Schritt bereits angedeutet bzw. vorbereitet.

Zwei der wichtigsten Veränderungen bei PHP von der vorherigen Version 5.6 auf die Version 7 waren die Abschaffung der MySQL-Schnittstelle (ersetzt durch MySQLi und vor allen Dingen PDO) sowie mehrerer Funktionen im Umgang mit regulären Ausdrücken. Die weiteren Unterversionen von PHP 7.x haben den endgültigen Bruch zu den Vorgängerversionen abgeschlossen, womit PHP 8 zu PHP 7 (insbesondere der letzten Version 7.4) weitgehend kompatibel bleibt. Dennoch gibt es auch beim Umstieg von PHP 7.x auf PHP 8.x unter Umständen einige Dinge zu beachten, die unter <https://www.php.net/manual/en/migration80.php> erläutert werden.

Natürlich bringen die PHP-Versionen der Schiene 8 auch diverse Erweiterungen und Neuerungen, die hier nur kurz angerissen werden.

Die vermutlich wichtigste Neuerung in PHP 8 findet unter der Oberfläche statt und dürfte die Einführung des neuen Just-in-Time-Compilers (JIT) darstellen. PHP-Quellcode wird nicht kompiliert (also komplett vor der Ausführung übersetzt), sondern während der Ausführung Zeile für Zeile interpretiert. Ein JIT-Compiler (JIT = Just in Time) kompiliert nun Teile des Codes während der Laufzeit in einen Zwischencode, der wiederum zur Laufzeit in Maschinencode umgewandelt wird. Dadurch soll unter gewissen Konstellationen ein deutlicher Performance-Sprung möglich sein. Allerdings kann es auch zu der Situation kommen, dass Fehler bereits früher bemerkt werden als in Vorgängerversionen und sich daher sowohl die Meldungen zu Fehlern als auch deren Management verändern.

Während der JIT-Compiler quasi im Hintergrund arbeitet, gab es auch bei der Sprache selbst einige Neuerungen. PHP 8 unterstützt jetzt beispielsweise Attribute, mit denen sich strukturierte Metadaten direkt in der nativen PHP-Syntax nutzen lassen und statt PHPDoc-Annotations können native Union-Type-Deklarationen für Sammeldeklarationen verwendet werden. Union Types (Aufzählungstypen) kennt man aus diversen Sprachen. Hierbei können zwei oder mehrere Typen eine Union (Verband oder Verbindung) eingehen und jeder der darin genannten Typen kann in der Folge im Code verwendet werden. Darüber hinaus bringt PHP 8 Sprachneuerungen wie

- ✓ „Named Arguments“ (benannte Argumente für Funktionen), um Eingabedaten anhand des Parameternamens und nicht der Deklarationsreihenfolge an eine Funktion zu übergeben,
- ✓ die Constructor Property Promotion,
- ✓ einen `static`-Rückgabetypp bei Klassenmethoden,
- ✓ die mögliche Verwendung von `::class` bei Objekten statt `get_class()`,
- ✓ ein Stringable-Interface für eine `toString()`-Methode,
- ✓ die neue Funktion `fdiv()`, um ein vernünftiges Ergebnis bei der Division durch 0 (INF, -INF oder NAN) zu erhalten,
- ✓ Typen-Fehler (Type errors) für die meisten internen Funktionen oder
- ✓ den Zwang von korrekten Signaturen magischer Methoden.

Die vollständige Liste der Neuerungen von PHP 8 finden Sie unter <https://www.php.net/manual/en/migration80.new-features.php>. Ebenso sollten Sie beachten, dass auch PHP 8.1 einige weitere Neuerungen gebracht hat. In der Folge werden wir einen Teil dieser Neuerungen von PHP 8.x noch vertiefen.

## Typografische Konventionen

### Hervorhebungen im Text

Im Text erkennen Sie bestimmte Programmelemente an der Formatierung. So werden z. B. Bezeichnungen für Programmelemente wie Register immer *kursiv* geschrieben und wichtige Begriffe **fett** hervorgehoben.

<i>Kursivschrift</i>	kennzeichnet alle von Programmen vorgegebenen Bezeichnungen für Schaltflächen, Dialogfenster, Symbolleisten, Menüs bzw. Menüpunkte (z. B. <i>Datei - Schließen</i> ) sowie alle vom Anwender zugewiesenen Namen wie Dateinamen, Ordernamen, eigene Symbolleisten, Hyperlinks und Pfadnamen.
Courier New	kennzeichnet Programmtext, Programmnamen, Funktionsnamen, Variablennamen, Datentypen, Operatoren etc.
<i>Courier New kursiv</i>	kennzeichnet Zeichenfolgen, die vom Anwendungsprogramm ausgegeben oder in das Programm eingegeben werden.
[ ]	Bei Darstellungen der Syntax einer Programmiersprache kennzeichnen eckige Klammern optionale Angaben.
	Bei Darstellungen der Syntax einer Programmiersprache werden alternative Elemente durch einen senkrechten Strich voneinander getrennt.

## 1.2 Bevor Sie beginnen ...

### HERDT BuchPlus – unser Konzept:

#### Problemlos einsteigen – Effizient lernen – Zielgerichtet nachschlagen

(weitere Infos unter [www.herdt.com/BuchPlus](http://www.herdt.com/BuchPlus))

Nutzen Sie dabei unsere maßgeschneiderten, im Internet frei verfügbaren Medien:



Wie Sie schnell auf diese BuchPlus-Medien zugreifen können, erfahren Sie unter [www.herdt.com/BuchPlus](http://www.herdt.com/BuchPlus)

# 2

## Das Umfeld von PHP

### 2.1 Webseiten und Webanwendungen

PHP findet seine hauptsächliche Anwendung im Umfeld des WWW (World Wide Web) und bei der Erstellung von Inhalten für dieses WWW. Es ist aus Webseiten und Webanwendungen aufgebaut.

#### Statische und dynamische Webseiten

Eine Webseite ist im Grunde nur ein Klartextdokument im WWW, das mit einem Browser (Client) unter Angabe eines Uniform Resource Locators (URL) von einem Webserver angefordert werden kann. Der Webserver und der Browser „wissen“ jedoch auf Grund des verwendeten **MIME-Typs** (Multipurpose Internet Mail Extensions), dass so ein Textdokument eine spezielle Bedeutung hat. Der MIME-Type oder auch Internet Media Type klassifiziert bei der Übertragung von Daten per http diese im Rumpf der Datenpakete.

Der MIME-Type besteht aus zwei Teilen:

- ✓ der Angabe eines allgemeinen Medientyps und
- ✓ der Angabe eines Subtyps.

Beide Angaben werden durch einen Schrägstrich voneinander getrennt.

Im Fall einer Webseite sieht das so aus:

```
text/html
```

Der innere Rahmen des Dokuments besteht bei einer Webseite hauptsächlich aus HTML (Hyper Text Markup Language), weshalb in diesem Zusammenhang auch von einer HTML-Seite oder einem HTML-Dokument gesprochen wird. Allerdings kann auch XML (eXtensible Markup Language) Teil solch einer Webseite sein.

Bei klassischen oder **statischen Webseiten** werden unter Verwendung von HTML-Befehlen (Tags) fest vorgegebene Texte, Grafiken, Hyperlinks etc. in der Webseite verbunden. So eine statische Webseite gestattet dann meist nur das Betrachten dieser Inhalte (unter Umständen durch Multimedia erweitert) und die Verzweigung über ebenfalls statische Hyperlinks. Eine eventuelle Kommunikation mit dem Webserver erfolgt in der Regel über HTML-Formulare und die CGI-Schnittstellenvereinbarung (Common Gateway Interface).



Im Gegensatz zu statischen Webseiten, die bereits auf dem Webserver als fertige HTML-Datei vorliegen, werden **dynamische Webseiten** häufig erst im Moment der Anforderung durch den Client auf dem Webserver erzeugt (etwa mittels PHP). Das so zunächst dynamisch generierte HTML-Dokument wird dann vom Webserver an den Browser übertragen und dort angezeigt, wobei der Benutzer in der Regel dieses jedoch nicht mehr von einem statischen HTML-Dokument unterscheiden kann.

Der Begriff „Dynamik“ bezieht sich also in diesem Zusammenhang auf die Erstellung von Inhalten über Programmierung auf dem Webserver, nicht auf das Verhalten der Webseite im Client.

## Webanwendungen

**Webanwendungen**, häufig auch Webapplikationen genannt, sind Anwendungen beziehungsweise Programme, die zu einem gewissen Teil oder auch ausschließlich auf einem Webserver ausgeführt werden. Die Darstellung der Informationen und Interaktion mit dem Anwender erfolgt jedoch auch hier über den Browser. Bei modernen Webanwendungen wird aber meist ebenfalls im Browser (also im Client) programmiert (in der Regel mit JavaScript). Obwohl es prinzipiell möglich ist, dass man bei Webanwendungen sogar ausschließlich im Client programmiert, ist auch das eher selten.

Bei modernen Webapplikationen ist die Programmierung fast immer auf Server und Client verteilt.

Logik, die auf den Server gehört, betrifft kritische Schritte oder meist nur zentralisiert sinnvoll verwaltbare Dinge, wie:

- ✓ Verifizierung von Benutzerdaten
- ✓ Datenbankabfragen
- ✓ Verwaltung von Benutzereingaben
- ✓ Verwaltung von Sessions

Die clientseitige Programmierung betrifft im Wesentlichen die Interaktion mit dem Anwender, wie:

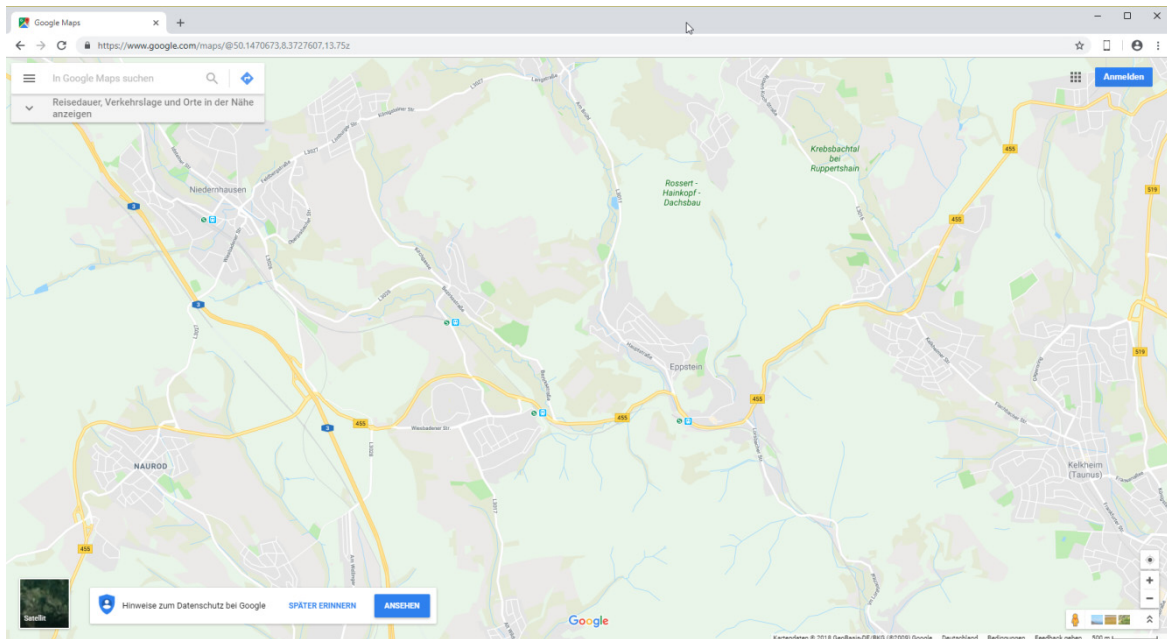
- ✓ Anzeigen von Fehlern bei der Benutzereingabe und allgemeine Benutzerführung
- ✓ Animationseffekte wie das harmonische Aufblenden von Menüs
- ✓ Übermitteln von Daten im Client an den Server (Bildschirmauflösung, verwendetes Gerät, Standort des Benutzers, etc.) – oft auch unbemerkt vom Anwender im Hintergrund
- ✓ Nachfordern von Daten

Die Datenübertragung zwischen dem Client und dem Server findet aber auch bei Webanwendung wie im WWW üblich auf der Transportebene auf Basis von TCP/IP (Transmission Control Protocol/Internet Protocol) und auf Anwendungsebene durch das Protokoll HTTP (Hypertext Transfer Protocol) oder dessen sicherer, verschlüsselter Version HTTPS (Hypertext Transfer Protocol Secure) statt. Die räumliche Entfernung zwischen dem Client und dem Webserver ist für eine Webanwendung ohne Bedeutung, denn beide sind über das Internet oder ein lokales Netzwerk miteinander verbunden.

In umfangreichen, professionellen Umgebungen werden die serverseitigen Programmierschritte der Webanwendungen oft nicht mehr direkt auf dem Webserver ausgeführt, sondern auf zusätzlichen **Applikationsservern**. Die Anwendungsdaten werden in der Regel über **Datenbankmanagementsysteme** zur Verfügung gestellt.

## Rich Internet Application (RIA)

In den letzten Jahren hat sich eine Gruppe an besonderen Webanwendungen etabliert, die in ihren Interaktionsmöglichkeiten und oft auch im gesamten Look & Feel wie Desktop-Anwendungen erscheinen. Diese werden **Rich Internet Applications** (RIAs) genannt. Sie bieten innerhalb des Browsers Interaktionsmöglichkeiten mit dem Anwender wie beispielsweise Drag & Drop oder komplexe Menüs. Oder sie sind grafisch besonders anspruchsvoll, sodass man sie von der Optik kaum von klassischen Desktop-Anwendungen unterscheiden kann. Sehr oft tauschen RIAs auch im Hintergrund Daten mit dem Webserver aus, wobei hier sehr oft eine Technologie namens **Ajax** (Asynchronous JavaScript And XML) zum Einsatz kommt.



*Google Maps ist eine typische RIA*

## 2.2 PHP

PHP stand ursprünglich für „Personal Homepage Tools“, steht aber mittlerweile für „PHP: Hypertext Preprocessor“. Das ist ein sogenanntes rekursives Akronym, da der Bezeichner „PHP“ in der ausgeschriebenen Version wiederholt wird. PHP wird überwiegend bei serverseitiger Web-Programmierung eingesetzt, ist aber darauf nicht beschränkt.

Die Geschichte von PHP geht bis in das Jahr 1995 und den von Rasmus Lerdorf entwickelten und zunächst nur auf Perl basierenden PHP/FI (Personal Home Page/Forms Interpreter) zurück. Derzeit ist die Version PHP 8 aktuell. Bereits die Version 7 hatte eine deutlich bessere Performance im Vergleich zu den Vorversionen erreicht und das Major Release 8 verbessert die Situation noch einmal (insbesondere bei einigen Situationen mit dem JIT).

Für die Popularität von PHP gibt es mehrere Gründe, u. a. diese:

- ✓ Der Einstieg in PHP ist einfach.
- ✓ PHP ist leicht zu benutzen, denn PHP-Befehle können einfach in umgebende HTML-Strukturen eingebettet werden. Diese Vorgehensweise wird manchmal SSI (Server Side Includes) genannt, wobei der Begriff im engen Sinn eine eigenständige Technologie beschreibt, die aber als Namensgeber gerne verwendet wird.
- ✓ PHP ist Open-Source und nicht zuletzt deshalb stellen nahezu alle Webhoster Angebote mit PHP-Unterstützung zur Verfügung.
- ✓ PHP kann auf allen gängigen Betriebssystemen verwendet werden.
- ✓ PHP unterstützt viele verschiedene Datenbanksysteme und hat auch für ziemlich jede denkbare Anwendung eine riesige Standardbibliothek.
- ✓ Es gibt eine große und aktive PHP-Community.
- ✓ Zu vielen Aufgabenstellungen gibt es bereits fertige PHP-Skripte.
- ✓ PHP hat sich auf die Programmierung von dynamischen Webseiten und Webanwendungen spezialisiert und bietet hierzu sehr viele leistungsfähige Features.
- ✓ Mit PHP kann man sowohl prozedural als auch objektorientiert programmieren.

## Funktionsweise von PHP

Obwohl Grundlagen in PHP vorausgesetzt werden, soll im Folgenden kurz behandelt werden, wie PHP-Dateien bei einer Verwendung im Web grundsätzlich ausgeführt werden. Der Vorgang ist so gut wie immer gleich:

- ✓ Im ersten Schritt sendet der Client (Browser) per HTTP(S) eine Anfrage nach einer PHP-Datei an den Webserver (der sogenannte Request).
- ✓ Der Webserver identifiziert anhand der Dateinamenerweiterung die Anfrage als PHP-Request und sucht die angeforderte PHP-Datei im Dateisystem.
- ✓ Die PHP-Datei wird zur weiteren Verarbeitung an den PHP-Interpreter übergeben, der für den Server konfiguriert ist. In neuen Versionen kommt in gewissen Situationen ergänzend ein Just-in-Time-Compiler (JIT) zum Einsatz, der Teile des Codes während der Laufzeit in einen Zwischencode kompiliert, der wiederum zur Laufzeit in Maschinencode umgewandelt wird. Diese Unterscheidung werden wir in der Folge aber nicht berücksichtigen.
- ✓ Der PHP-Interpreter führt im nächsten Schritt zunächst eine Syntaxanalyse durch. Der Interpreter zerlegt dabei den Quelltext der Datei und bringt ihn in eine für die Ausführung geeignete Form. Anschließend wird der PHP-Code ausgeführt. Ggf. werden zusätzliche Programme an der Verarbeitung beteiligt (etwa ein Datenbankmanagementsystem – DBMS).
- ✓ Das Ergebnis liefert der Interpreter zurück an den Webserver.
- ✓ Der Webserver sendet die so dynamisch generierte Antwort (Response) an den Client zurück. Das Ergebnis wird meist ein dynamisch erzeugtes HTML-Dokument sein, das zwar die Dateierweiterung `.php`, aber den MIME-Type `text/html` hat. Es kann sich aber auch um ein PDF-Dokument, eine Grafik oder eine Datei eines anderen Formats handeln.



*Der PHP-Interpreter auf einem Windows-System*

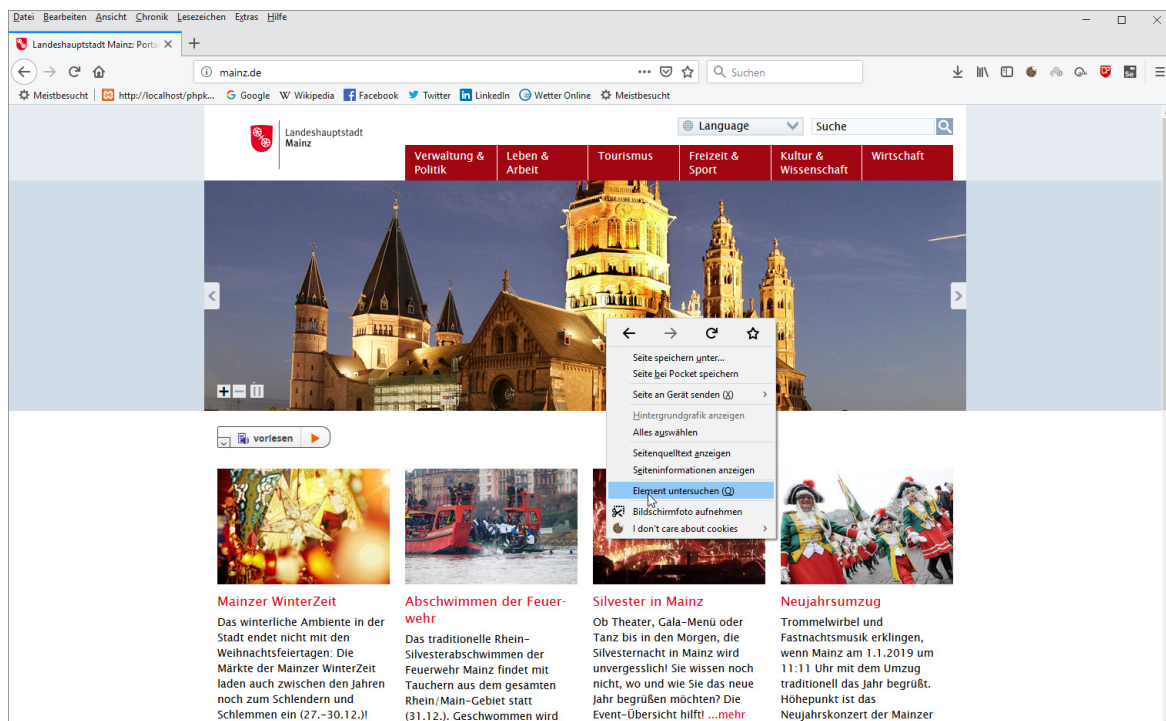
## 2.3 Übungen

**Basisdateien: --**

**Ergebnisdateien: --**

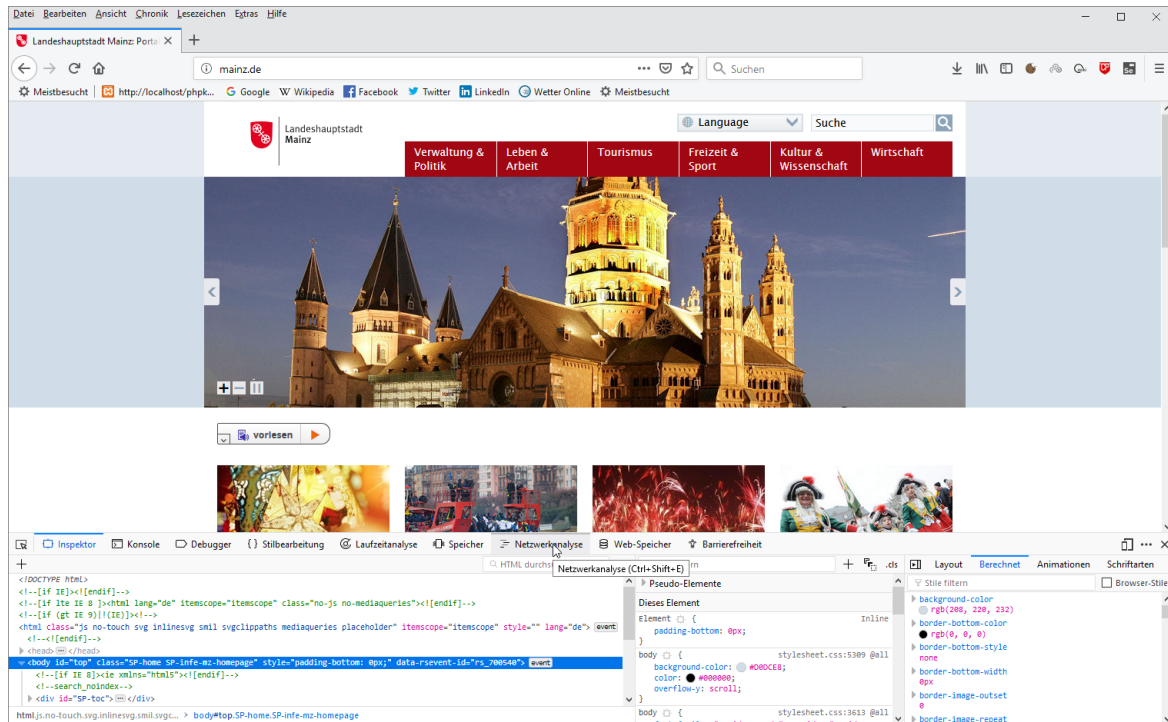
1. Öffnen Sie einen modernen Browser.
2. Laden Sie eine populäre Webseite, z. B. die einer Landeshauptstadt oder <http://www.herdt.com>.
3. Klicken Sie mit der rechten Maustaste in die Webseite. Im Kontextmenü erkennen Sie einen Befehl zum Untersuchen der Webseite. Der Befehl steht mittlerweile in allen modernen Browsern zur Verfügung, wird z. T. unterschiedlich benannt. In Firefox, Edge und Chrome nennt er sich *Untersuchen*.

Alternativ können Sie aber auch die Taste **F12** verwenden.



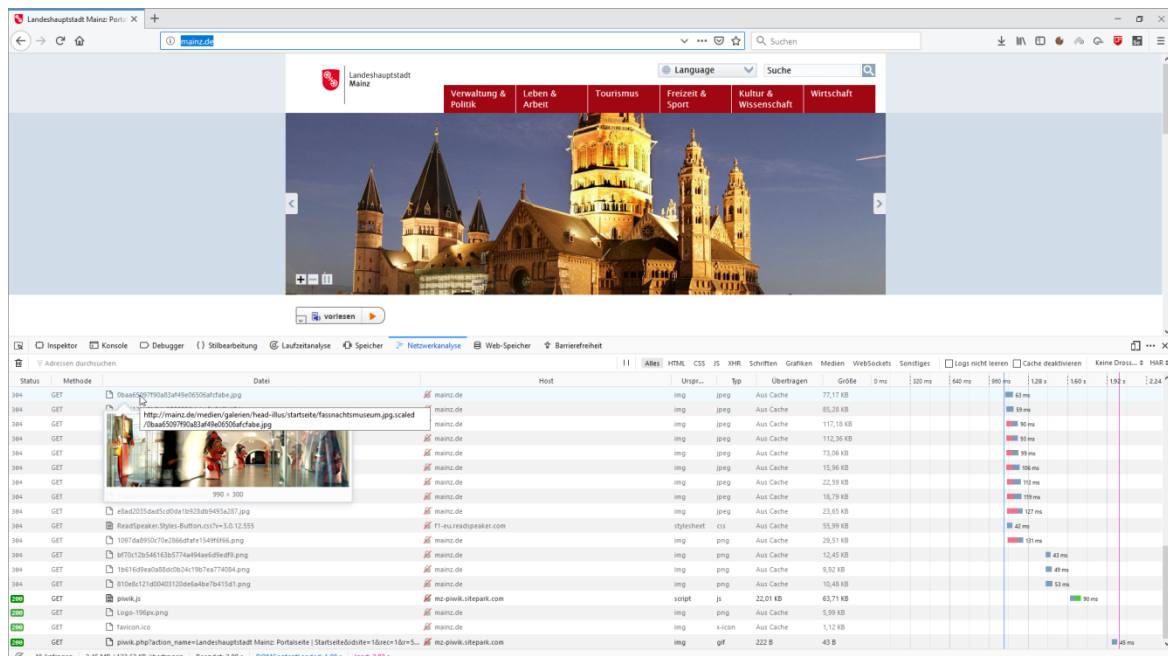
In jedem modernen Browser gibt es die Möglichkeit, eine Seite zu untersuchen.

4. Innerhalb des Browserfensters werden nun verschiedene Bereiche angezeigt, mit denen man verschiedenste Facetten einer Webseite untersuchen kann. Etwa den tatsächlich verwendeten Quellcode, Fehler, die Style Sheets oder auch die übertragenen Daten.
5. Betrachten Sie die verschiedenen Bereiche.



Die Hintergründe einer Webseite

6. Wählen Sie die Netzwerkanalyse (Network) aus. Das ist ein sogenannter Sniffer, mit dem Sie die Kommunikation zwischen dem Browser und dem Webserver analysieren können.
7. Laden Sie mit **(F5)** die Webseite neu. Sie sehen dann alle Dateien, die von der Webseite geladen und für den Aufbau des Angezeigten wirklich verwendet werden.



Auswahl des integrierten Sniffers und Neuladen der Seite

8. Klicken Sie eine der Dateien an. Sie werden dann die genauen Informationen zu dieser Datei bekommen – etwa unter dem Content-Type des in diesem Kapitel vorgestellten MIME-Type.

The screenshot shows a web browser displaying a page with a large image of a manuscript. Below the image, there is a network inspector tool. The network inspector shows a list of requests, with the 'piwik.js' file selected. The details pane on the right shows the response headers for 'piwik.js', including 'Content-Type: application/javascript'.

Status	Methode	Datei	Host	Urspr...	Typ	Übertragen	Größe	0 ms	100 ms	200 ms	300 ms	400 ms	500 ms	600 ms	700 ms	800 ms	900 ms	1000 ms	
200	GET	0ba450799483af49e0556af...	mainz.de	img	jpg	Aus Cache	77,17 KB												
200	GET	568412b093de2556006bba3e...	mainz.de	img	jpg	Aus Cache	85,28 KB												
200	GET	56840664a057d0d8e069a3f59...	mainz.de	img	jpg	Aus Cache	117,18 KB												
200	GET	4a3a489928c99fcd8f630a3a60...	mainz.de	img	jpg	Aus Cache	112,36 KB												
200	GET	f9496a8e61866433b3445a3f71...	mainz.de	img	jpg	Aus Cache	73,06 KB												
200	GET	26600265d9f5c0783b08e7585...	mainz.de	img	jpg	Aus Cache	15,96 KB												
200	GET	0edf5d94b4487981df3a79314...	mainz.de	img	jpg	Aus Cache	22,59 KB												
200	GET	174d39f160cb2936a116d93ca22...	mainz.de	img	jpg	Aus Cache	16,79 KB												
200	GET	4b0c055da0c0d0a103030804...	mainz.de	img	jpg	Aus Cache	23,01 KB												
200	GET	ReadSpeaker_Spoken_Button_02...	fi-hu.readspeaker.com	stylesheet	css		53,99 KB												
200	GET	1097da893d7a3b6df4e1549...	mainz.de	img	png	Aus Cache	29,51 KB												
200	GET	b970c1265461635774a09a6e6...	mainz.de	img	png	Aus Cache	12,45 KB												
200	GET	1b616df8a0a8d024c19b7ea...	mainz.de	img	png	Aus Cache	9,32 KB												
200	GET	810e61214004031204e6a8e7...	mainz.de	img	png	Aus Cache	10,48 KB												
200	GET	piwik.js	mz-piwik.sitepark.com	script	js		22,91 KB												
200	GET	Logo-196px.png	mainz.de	img	png	Aus Cache	5,99 KB												
200	GET	favicon.ico	mainz.de	img	icon	Aus Cache	1,12 KB												
200	GET	piwik.php?action_name=lande...	mz-piwik.sitepark.com	img	gif		222 B												

Untersuchung einer Datei

# 3

## Die Idee der objektorientierten Programmierung

### 3.1 PHP und OOP

Noch heute programmiert man in PHP in der Regel prozedural beziehungsweise strukturiert oder zumindest nicht streng objektorientiert. Oftmals werden OOP-Techniken und klassische Techniken aus der prozeduralen beziehungsweise strukturierten Welt vermischt. Das ist sicher vor allen Dingen historisch bedingt, denn PHP wurde am Anfang sehr einfach und rein prozedural konzipiert. Der Fokus lag auf der Spezialisierung von Webapplikationen, wie etwa auf der möglichst einfachen Zusammenarbeit mit clientseitigen Formularen aus einem Browser und serverseitigen Datenbanken.

Im Laufe der Zeit wurde PHP aber sukzessive um die Möglichkeit der objektorientierten Programmierung erweitert – erst nur mit wenigen Teilaspekten, nach und nach kamen so gut wie alle wichtigen Elemente der OOP hinzu. Modernes PHP unterstützt nun mehrere Programmierparadigmen.

Bereits mit der Version 4 haben einzelne objektorientierte Merkmale in PHP Einzug gehalten und seit PHP 5 sind fast alle modernen Konzepte der objektorientierten Entwicklung möglich. So kann man heutzutage objektorientierte Programmierung in PHP nutzen, muss es aber nicht. Ab PHP 7 können jedoch einige Dinge nur noch objektorientiert optimal umgesetzt werden.

Aber auch heute noch wird in PHP, wie erwähnt, oft entweder rein prozedural beziehungsweise strukturiert programmiert oder aber es werden lediglich Objekte und einzelne Techniken aus der OOP eingesetzt; ein Projekt wird aber nicht wirklich objektorientiert umgesetzt.

Dafür gibt es Gründe, die durchaus ihre Berechtigung haben:

- ✓ Das Gerüst einer Webseite oder Webapplikation ist reines HTML und PHP wird lediglich an einigen Stellen eingebaut. Damit ist der äußere Rahmen schon explizit nicht objektorientiert.
- ✓ OOP ist nicht trivial. Das Wissen um die Konzepte und Techniken der OOP ist oft bei PHP-Programmierern nicht vorhanden. Und da es meist auch andere Möglichkeiten gibt, ist der Druck nicht vorhanden, sich in die OOP einzuarbeiten.
- ✓ Leider nutzt PHP bei der Notation der objektorientierten Programmierung im Vergleich zu den meisten anderen OO-Sprachen eine oft abweichende Syntax. Das erschwert Umsteigern den Einstieg, wobei sich erfahrene OO-Programmierer bereits nach kurzer Zeit zurechtfinden werden.