



PROGRAMMIEREN
LERNEN

PHP LERNEN MIT ERFOLG

Vorwort

Lieber Leser, liebe Leserin,

in folgendem PHP-Kurs stecken unzählige Stunden. Sie können das Projekt unterstützen, indem Sie einen Link auf die Seite www.php-kurs.com setzen und die Seite weiterempfehlen.

Die Unterlagen sind ursprünglich zur Unterstützung der Kursteilnehmer der PHP-Kurse am Rechenzentrum der Universität Tübingen von mir geschrieben worden, an dem ich diese Kurse halte. Dabei gibt es 5 aufeinanderfolgende Kurse, die aufeinander aufbauen.

Aufgebaut sind diese Unterlagen so, dass auch ein Lernen im Selbststudium möglich ist.

Diese Unterlagen gibt es in 2 verschiedenen Fassungen – kostenlos (aus Tradition und mit Stand von 2008) und in der Fair-Play-Version zusätzlich mit dem Kapitel „PHP und Sicherheit“. Die Fair-Play-Version wird laufend weiter gepflegt und mit den Einnahmen werden z. B. Serverkosten, Lektorat etc. bezahlt. **Fair-Play-Preiskonzept** deshalb, da Sie selber den Preis festlegen. Sie zahlen, was es Ihnen wert ist! Eine Bitte dabei: Keine Beträge unter 1 Euro, da dies mehr Aufwand verursacht als Nutzen. Wir müssen die Einnahmen versteuern. Kontoverbindung und die aktuellste Version der Unterlagen erhalten Sie unter <http://www.php-kurs.com/php-ebook.php>

So, nun viel Spaß und Erfolg beim Lernen
Axel Pratzner

Vorwort	2
PHP-Kurs - PHP lernen und programmieren	7
Möglichkeiten von PHP	8
Theorie zum Verständnis.....	9
Voraussetzung zum PHP lernen	10
PHP und Apache mit XAMPP	11
PHP und Apache für MAC.....	16
Editoren zum Programmieren.....	17
Konfiguration von Scriptly für PHP-Programmierung	17
FTP-Programme	24
Dateien per sFTP auf Webserver übertragen.....	25
Benötigte HTML-TAGs	28
Internet Browser	29
Webspace mit PHP-Unterstützung	30
Alternativen zu XAMPP.....	30
Einsteiger.....	31
Unser erstes PHP-Programm.....	31
Ausgabe in Browser	33
Sonderzeichen maskieren	36
Auskommentieren von Code	38
PHP und HTML mischen	40
Variablen in PHP nutzen	41
Verkettungsoperator	46
Vordefinierte Variablen.....	47
phpinfo - Funktion zur Anzeige der Serverinformationen	47
Berechnungen in PHP.....	50
Funktionen von PHP.....	53
Datumsausgabe mit date()	55
Bedingungen if / else	59
Formulare - Affenformular, Auswertung und Feedback	61
Formularwerte über IF kontrollieren	65
AUFGABE: Erstellen Sie ein Affenformular mit dem hidden-Feld mit Namen "aktion" und dem Wert "1"	67
Affenformular Beispiel.....	75

per PHP in Dateien schreiben	78
Schleifen	82
Funktionen in PHP	86
häufige Fehlerquellen	93
Fortgeschrittene	95
Formularinhalte in Dateien speichern	95
Steuerzeichen.....	96
Dateien auslesen	97
Verzeichnis einlesen.....	98
Variablen-Typen.....	99
Konstanten - fixe Variablen.....	100
Bedingungen if elseif.....	101
Bedingung switch	102
Daten vergleichen	102
E-Mails mit PHP erstellen	103
HTML-E-Mail mit PHP erstellen	104
Übung Newsletteranmeldung	105
Script Newsletteranmeldung	108
Arrays	113
Assoziatives Array.....	114
Array-Elemente zählen.....	115
Inhalt ausgeben	116
Sortieren von Arrays.....	117
Doppelte Einträge im Array löschen	119
Mehrdimensionale Arrays	120
Nützliche Funktionen für Arrays.....	122
Beispiel Bildergalerie erstellen.....	123
Funktionsbibliotheken erstellen und recyclen	126
Übung Terminkalender.....	128
SESSION - Kurzzeit-Gedächtnis für PHP	133
Lösung Einlogg-Script.....	136
Grafiken erstellen	137
Lösung E-Mail als Grafik.....	140
CAPTCHA erstellen.....	140
Script Captcha-Clock.....	143

Objektorientierte Programmierung in PHP	144
Grundgedanke Objektorientierte Programmierung:	144
Grundgedanke Objektorientierte Programmierung: Methoden.....	145
Grundgedanke Objektorientierte Programmierung: Klassen und Objekte.....	145
Beispiel Klasse/Objekt	146
Konstruktor / Destruktor.....	149
Vererbung bei Klassen	150
PHP5 Klassen	150
Und warum gebe ich mir das Ganze?.....	151
 mysql - Datenbank unter PHP	 153
Daten strukturieren - Vorüberlegungen zur Datenbankerstellung.....	153
Primär- und Fremdschlüssel	158
Beziehungen	159
Datenstruktur grafisch darstellen.....	161
Aufbau Datensatz	163
Entity Relationship Modell (ERM)	164
Normalisierung – logische Regeln zur Minimierung von Redundanz	165
MySQL testen.....	169
MySQL-Anweisungen lernen - phpMyAdmin nutzen	172
MySQL-SELECT-Anweisung - Varianten und Möglichkeiten	178
Verbindung zur MySQL-Datenbank herstellen.....	183
Datenbank anlegen	184
MySQL-Tabelle mit Inhalt füllen.....	186
Datenbank auslesen und anzeigen	186
Relationen - Tabellen verbinden	187
Zugangskontrolle mit Passwortschutz.....	191
Content Management System (CMS).....	194
Gästebuch programmieren mit MySQL als Datenbank	197
Gästebuch absichern.....	213
Beispiel persönliche Adressbuch programmieren mit Geodaten	218
Beispiel Terminkalender programmieren	234
Script PHP-Terminkalender	245
MySQL sauber schreiben	251
MySQL Datenbank sichern	252

PHP und Sicherheit	253
Einbinden von Dateien - include versus require.....	
Verzeichnis ohne Zugriff von außen	
Unterdrücken von PHP-Fehlermeldungen	
Zugriffe auf das Dateisystem.....	
Cross-Site Scripting (XSS) unterbinden.....	
SQL Injection	
Session-Hijacking: Session und Sicherheit	
Kontaktformular absichern, damit man nicht zur SPAM-Schleuder wird – Mail-Header Injection.....	
E-Mail und Sicherheit.....	
Daten filtern und Benutzereingaben überprüfen	
Security by Obscurity	
Tipps und Tricks.....	254
Tabellenreihen abwechseln einfärben	255
Debuggen mit Zeilenausgabe	256
Übergebene Variablen bereinigen	257
PHP Fehlermeldungen.....	257
Datumsausgabe auf Deutsch in PHP	259
MySQL mit deutschem Datumsformat	259
MySQL Groß- und Kleinschreibung berücksichtigen.....	260
automatischer Start nach Ablauf	261
Links.....	262
Bücher und Literaturhinweise	262
Schlußwort	263

PHP-Kurs - PHP lernen und programmieren

PHP ist eine sehr gute Programmiersprache für Einsteiger, da sie einfach zu erlernen ist und sehr schnell Erfolgsergebnisse sichtbar werden.

PHP lernen für Einsteiger

Im Vergleich zu anderen Programmiersprachen ist bei PHP ein schneller Einstieg möglich, PHP lernen ist einfach. Trotzdem sind komplexe Anwendungen und filigrane Programmierung möglich, was dann auch für den Fortgeschrittenen interessant wird. Von der prozeduralen Programmierung kann bei Bedarf dann auf objektorientierte Programmierung mit PHP 5 umgestiegen werden - die Kenntnisse vom Einstieg sind dafür Voraussetzung, also nicht unnötig.

Inhalt PHP-Kurs / MySQL Tutorial

Nachdem die Voraussetzungen zum PHP Lernen geschaffen sind (PHP-Entwicklungsumgebung installiert und Editor ausgewählt), gibt es für Einsteiger eine Einführung:

für PHP-Einsteiger

- Wie sieht ein PHP-Programm aus
- Ausgabe von Texten
- Variablen (Platzhalter für Inhalte)
- Berechnungen mit PHP - zurück in die Grundschule und rechnen lernen :)
- Datumsfunktionen

für die Fortgeschrittenen

- Mails per PHP erstellen (mein Programm mailt mir)
- if-else-Bedingungen
- Formulare erstellen und diese auswerten
- Dateien - endlich einen Besucherzähler
- Schleifen und Wiederholungen (was Computer nicht langweilt)
- Funktionen - wie kann ich mir das Programmieren vereinfachen

Datenbank MySQL-Tutorial

- Was ist eine MySQL Datenbank
- Aufbau einer Datenbank
- MySQL-Datenbank mit PHP ansprechen
- MySQL-Query aufbauen und testen
- Umfangreiches Beispiel erstellen: Terminkalender für die eigene Website

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Rahmen des PHP-Kurses

Diese Seite entsteht parallel zu meinen Kursen am Rechenzentrum der Universität Tübingen. Daher werden (je nach Kursbedarf) immer wieder Erweiterungen und Ergänzungen hier erfolgen. Also immer am Ball beim PHP Lernen bleiben.

Viel Spaß beim PHP Lernen und dem MySQL Tutorial

wünscht Axel Pratzner

Möglichkeiten von PHP

Hinter zahlreichen Anwendungen im Internet steckt PHP. Sehen Sie in der URL im Browser ein .php - da steckt eine PHP-Anwendung.

Für einen ersten Eindruck eine kleine (und unvollständige) Aufzählung unterschiedlicher PHP-Anwendungen:

- Besucherzähler
- Gästebücher
- Diskussionsforen
- Bilderdienste
- Umfragen
- Formulare mit Verarbeitung von Kontaktdaten
- Chats
- Wikis
- Blogs (z. B. WordPress)
- und vieles mehr ...

Wo PHP dahintersteckt, muss es nicht unbedingt sichtbar sein

PHP muss nicht unbedingt als Dateieindung .php auftauchen, aber trotzdem kann es im Hintergrund seine Arbeit verrichten. Das Content-Management-System (CMS) mit dem Namen TYPO3 z. B. ist ein PHP-Programm. Oft sieht man damit erstellte Internetauftritte, die ihre Einzelseiten mit „index.php?id=x“ haben. Das ist aus verschiedenen Gesichtspunkten her schlecht. Insbesondere ist somit für Suchmaschinen ersichtlich, dass es sich nicht um eine statische Seite handelt und deshalb u. U. nicht mit einer dauerhaften Information gerechnet werden kann.

Grenzen von PHP

Auch PHP hat seine Grenzen - z. B. werden keine grafiklastigen Actionspiele wie "Lara Croft" mit PHP entwickelt - dafür gibt es andere Programmiersprachen.

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

PHP läuft i.d.R. im Produktivbetrieb (d. h. wenn man es professionell will und keine Probleme haben möchte) auf einem Apache Server. Das bedeutet, dass normalerweise PHP nicht auf einem Windows-Internetserver eingesetzt wird. PHP lernen kann aber problemlos auf einem Windows oder Mac-System erfolgen. Die entsprechenden notwendigen Programme werde im Kapitel “Voraussetzung zum PHP lernen” vorgestellt.

Theorie zum Verständnis

PHP ist eine Scriptsprache, die ihre Heimat i.d.R. auf einem Webserver hat. Wird nun eine Internetseite aufgerufen, die ein PHP-Programm darstellt, wird das PHP-Programm auf dem Server ausgeführt und der Besucher erhält nur die Ausgabe (meistens HTML) zurück.

Dadurch ist die Ausführung der PHP-Programm eine sehr gute Sache, da diese nicht abhängig vom Computer des Surfers ist (im Gegensatz zu JavaScript, das auf dem Computer des Surfers aufgeführt wird).

PHP wird geparkt

PHP-Programme werden im Klartext erfasst (also kann jeder beliebige Text-Editor eingesetzt werden). Erst der Server setzt diesen Klartext dann ein Programm um und führt es aus (korrekt spricht man von parsen).

Ablauf einer Entwicklung

1. Schritt: Zuhause (oder in Ihrer Firma) entwickeln Sie ein PHP-Programm

Dort benötigen Sie als Minimalausstattung einen Computer, auf dem ein Webserver (Apache) mit PHP läuft.

(als Garant für Erfolg, Ruhm und Reichtum bietet sich in diesem Stadium als Ort die Garage oder der Dachboden an!)

2. Schritt: die fertige Anwendung wird auf den Webserver bei einem Hoster übertragen, der Ihren Programm auf einem Webserver einen Platz bietet. Ab diesem Zeitpunkt können Sie Ihren Rechner zuhause ausschalten, da das Programm ohne Ihre Zutun läuft (sofern keine Fehler drin sind).

3. Schritt: Über das Internet haben alle (wenn gewünscht) Zugriff auf Ihre Anwendung.

1. Schritt:

programmierung@home



2. Schritt:
Programm zum Host



3. Schritt:
weltweiter Zugriff auf
Programm



```
<?php  
echo "Hallo Welt";  
zaehle_zugriffe();  
?>
```

Hallo Welt

Voraussetzung zum PHP lernen

Folgende Dinge (neben der eigenen Motivation) sind nötig, um PHP lernen zu können:

- PHP
- Webserver (i.d.R. Apache)
- Text-Editor, um den Quellcode zu erfassen
- FTP-Programm für das Übertragen zum Webserver

Es gibt eine einfache Methode, an die komplette Installation von PHP zu kommen, ohne dass man sich am Anfang mit Apache-Konfiguration und PHP-Einstellungen herumschlagen muss (was eigentlich nicht schadet, aber unter Umständen PHP lernen anfangs ziemlich im Weg steht).

XAMPP - Apache und PHP Installation im Handumdrehen

XAMPP ist eine Zusammenstellung eines Installationspaketes, das alle wichtigen Bestandteile enthält (PHP, Apache, MySQL, etc.), sich einfach installieren lässt und kostenlos ist.

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

XAMPP unter <http://www.apachefriends.org/de/> herunterladen und installieren. Dabei ist die Lite-Variante vollkommen ausreichend zum PHP Lernen. Sie bekommen auf einen Schlag ohne viel Einstellungs-Schnick-Schnack eine zusammenpassende Version von PHP, Apache und zusätzlich MySQL, phpMyAdmin und vieles mehr, das am Anfang zwar noch nicht benötigt wird, aber praktisch für später ist). XAMPP gibt es für Windows, Mac OS X, Linux und Solaris.

Testen beim Provider

Noch einfacher (aber auf zeitaufwändiger) ist es, wenn Sie Ihre PHP-Scripte immer zu Ihrem Provider (sofern vorhanden) hochladen und direkt im Web testen. Das birgt allerdings die Gefahr, dass Dritte Ihre halbfertigen Programm sehen und womöglich an Daten kommen, die niemand außer Ihnen haben sollte ...

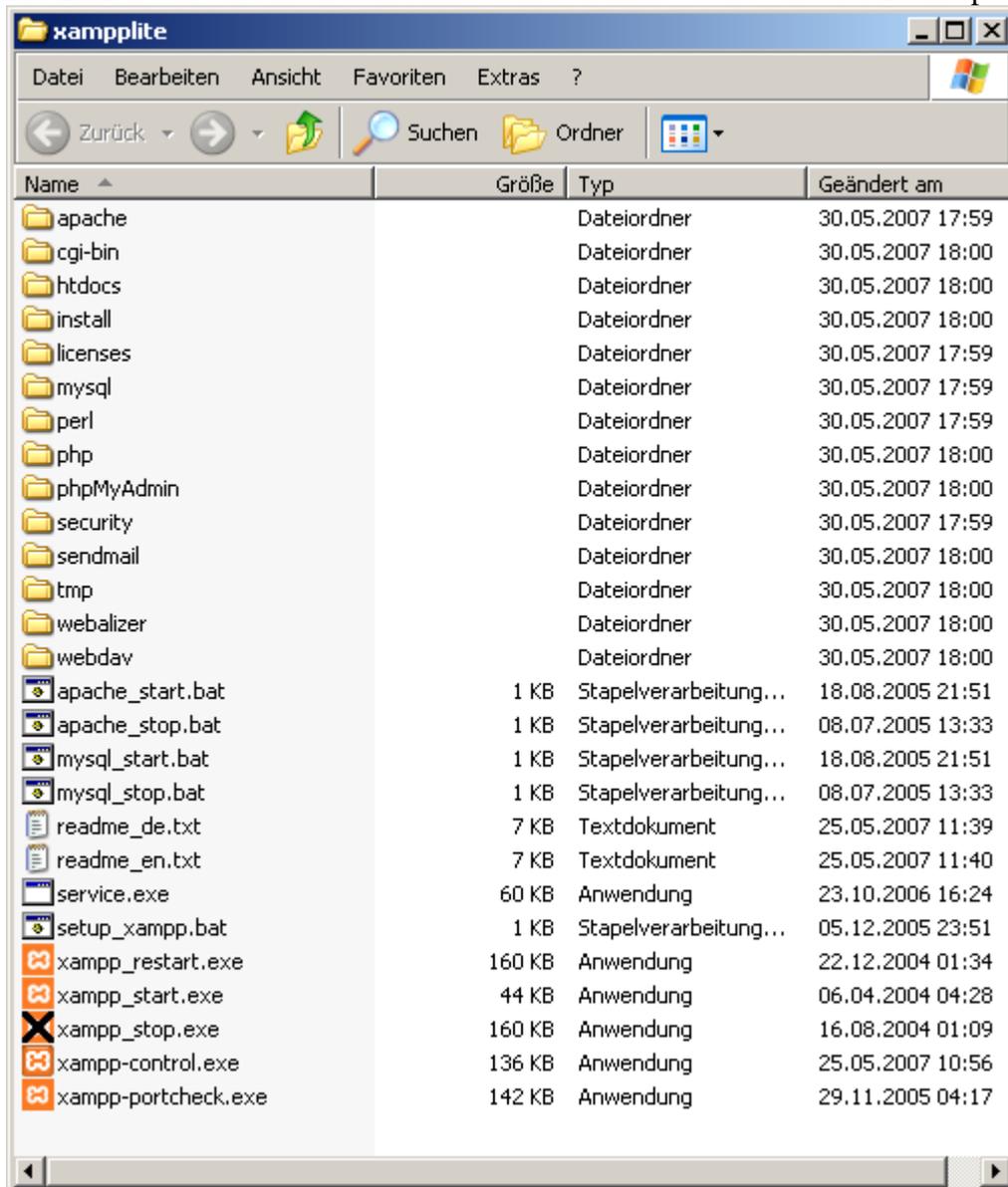
PHP und Apache mit XAMPP

einfache Installation von PHP, Apache und MySQL

Für unsere Belange ist die kleine Version von XAMPP ideal - XAMPP Lite genannt.

XAMPP-Lite ist zu finden unter <http://www.apachefriends.org/de/> - dort die fast 19MB große Installationsdatei herunterladen.

Die EXE-Datei wird nach dem Anklicken extrahiert - dazu wird der Desktop vorgeschlagen.



setup_xampp.bat ausführen und das war die Installation - zum Starten von XAMPP auf xampp_start.exe gehen - es erscheint ein DOS-Fenster, das während der gesamten Laufzeit von XAMPP nicht geschlossen werden darf!

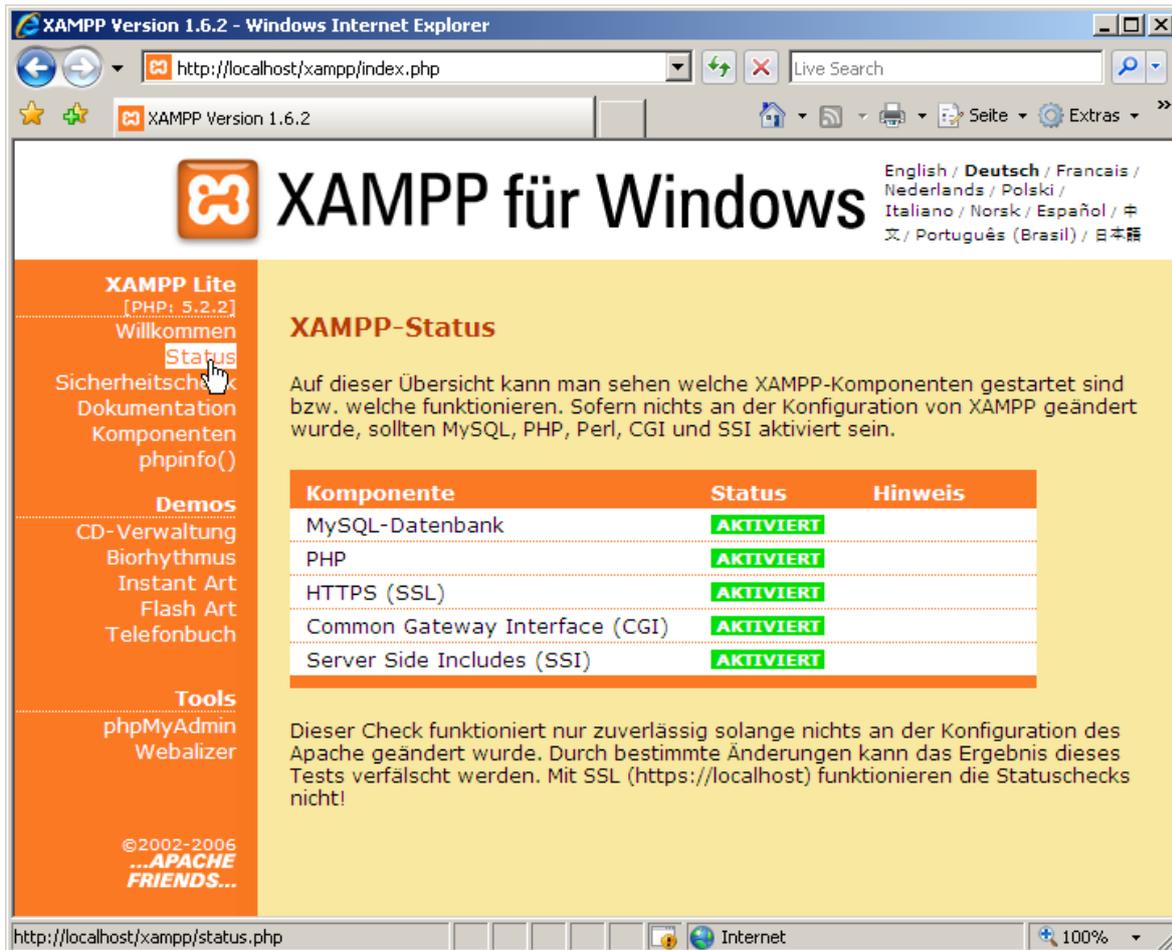
Verkleinern Sie es - in der Taskleiste ist dann das orangefarbige XAMPP-Symbol zu sehen.



Test XAMPP mit Apache

Zum Testen, ob mit der Installation alles richtig lief, können Sie nun einen Internet-Browser Ihrer Wahl aufrufen (z. B. Internet-Explorer oder Firefox) und in der Adresszeile die URL "http://localhost" eingeben. Anstatt localhost geht auch 127.0.0.1 - durch diese Angabe wird der Webserver auf Ihrem Rechner angesprochen. Es sollte dann folgendes Bild erscheinen - die Oberfläche von XAMPP wird gestartet und Sie können die Sprache wählen.

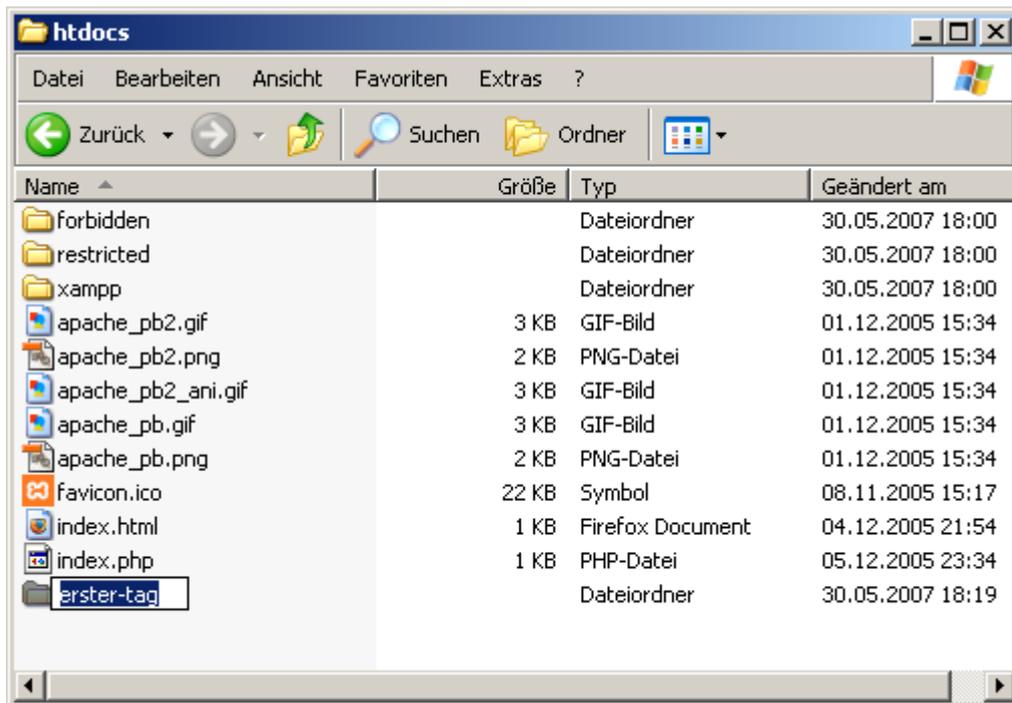
Durch Anklicken von Status in der Steuerung sehen Sie, ob PHP funktioniert.



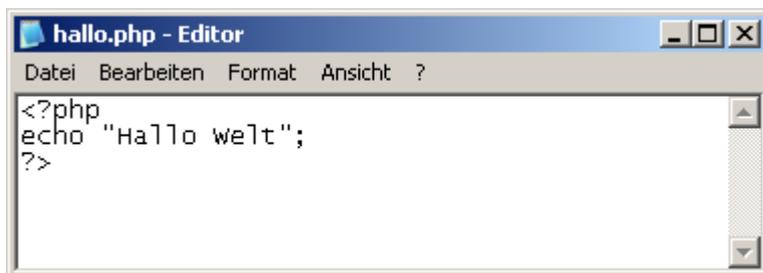
Verzeichnisse von XAMPP verstehen

Alle PHP-Programme müssen in das Unterverzeichnis "htdocs" des XAMPP-Verzeichnisses. Dort können Sie nach Belieben Unterverzeichnisse anlegen - empfehlenswert bei den Unterverzeichnissen ist es, keine Leerzeichen, keine Umlaute und nur Kleinschreibung einzusetzen (Bindestriche sind auch noch OK).

Wir legen nun für unsere ersten Gehversuche ein Verzeichnis mit dem Namen "erster-tag" an.



In dieses Verzeichnis speichern wir eine Datei mit dem Namen "hallo.php". Der Inhalt der Datei ist:



Beim Editor von Windows müssen Sie beim Speichern darauf achten, dass Sie den Dateinamen "hallo.php" in Anführungszeichen schreiben. Sonst ergänzt der Editor den Dateinamen um ein .txt und die fertige Datei heißt dann "hallo.php.txt" und wird vom Apache nicht als PHP-Datei erkannt!

Und nun geben wir im Browser ein: <http://localhost/erster-tag/hallo.php>
Die einzelnen Bestandteile:

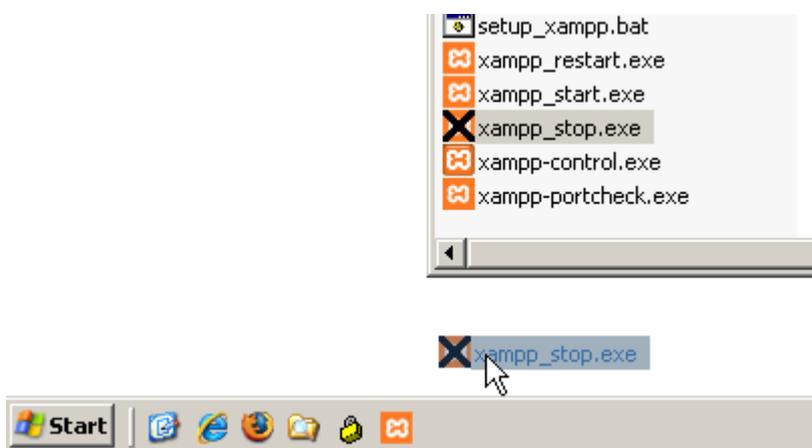
- **http://localhost/** -> unser lokaler Webserver auf dem eigenen Rechner
- **erster-tag/** -> das Verzeichnis, das sich in XAMPP-Verzeichnis unterhalb von htdocs befindet
- **hallo.php** - unser erstes PHP-Programm

Im Browser sollte nun die Ausgabe "Hallo Welt" erscheinen - die Installation hat geklappt - ein guter Start

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen



Zum Beenden von XAMPP gibt es im Verzeichnis "xampp_stop.exe". Am besten das Starten und das Stoppen auf die untere Schnellstartleiste ziehen und eine Verknüpfung von htdocs auf den Desktop erstellen.



Alternativen zu XAMPP

WAMP5 - <http://www.wampserver.com/> - französisch, englische Website mit Apache, PHP und MySQL Out-of-the-box Paket.

PHP und Apache für MAC

Was es für Windows gibt, gibt es auch für den Macintosh - Apache, Mysql and PHP.

Download und Infos sind zu finden unter: <http://mamp.info/de/mamp/>

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Editoren zum Programmieren

Es gibt zahlreiche Editoren, die einen beim Programmieren unterstützen. Dabei wird bei der bisherigen Auflistung Windows berücksichtigt.

Editor von Betriebssystem

Für die absoluten Puristen (oder für den Notfall) geht der im Betriebssystem mitgelieferte Editor.

Beim Editor von Windows müssen Sie beim Speichern darauf achten, dass Sie den Dateinamen "irgendwas.php" in Anführungszeichen schreiben. Sonst ergänzt der Editor den Dateinamen um ein .txt und die fertige Datei heißt dann "irgendwas.php.txt" und wird vom Webserver nicht als PHP-Datei erkannt!

Scriptly

Sehr zu empfehlen ist der kostenlose Editor Scriptly von Webocton, der laufend weiterentwickelt wird. Eine Anleitung zur [Konfiguration von Scriptly](#) zum bequemen PHP-Programmieren ist hier vorhanden. Die aktuelle Version von Scriptly können Sie unter www.scriptly.de herunterladen.

Für mehr Komfort gibt es Editoren, die den Quellcode automatisch farbig anzeigen (was eine schnelle Orientierung ermöglicht) und z. B. Codeergänzungen vorschlagen.

Weaverslave

Den Editor Weaverslave finden Sie unter www.weaverslave.ws/ - letzte Version ist vom Dezember 2005.

Zend Studio

Ein sehr gutes Produkt kommt aus dem Haus Zend (Entwickler von PHP) mit dem Zend Studio. Dieses läuft auf verschiedenen Betriebssystemen, da es mit Java programmiert wurde.

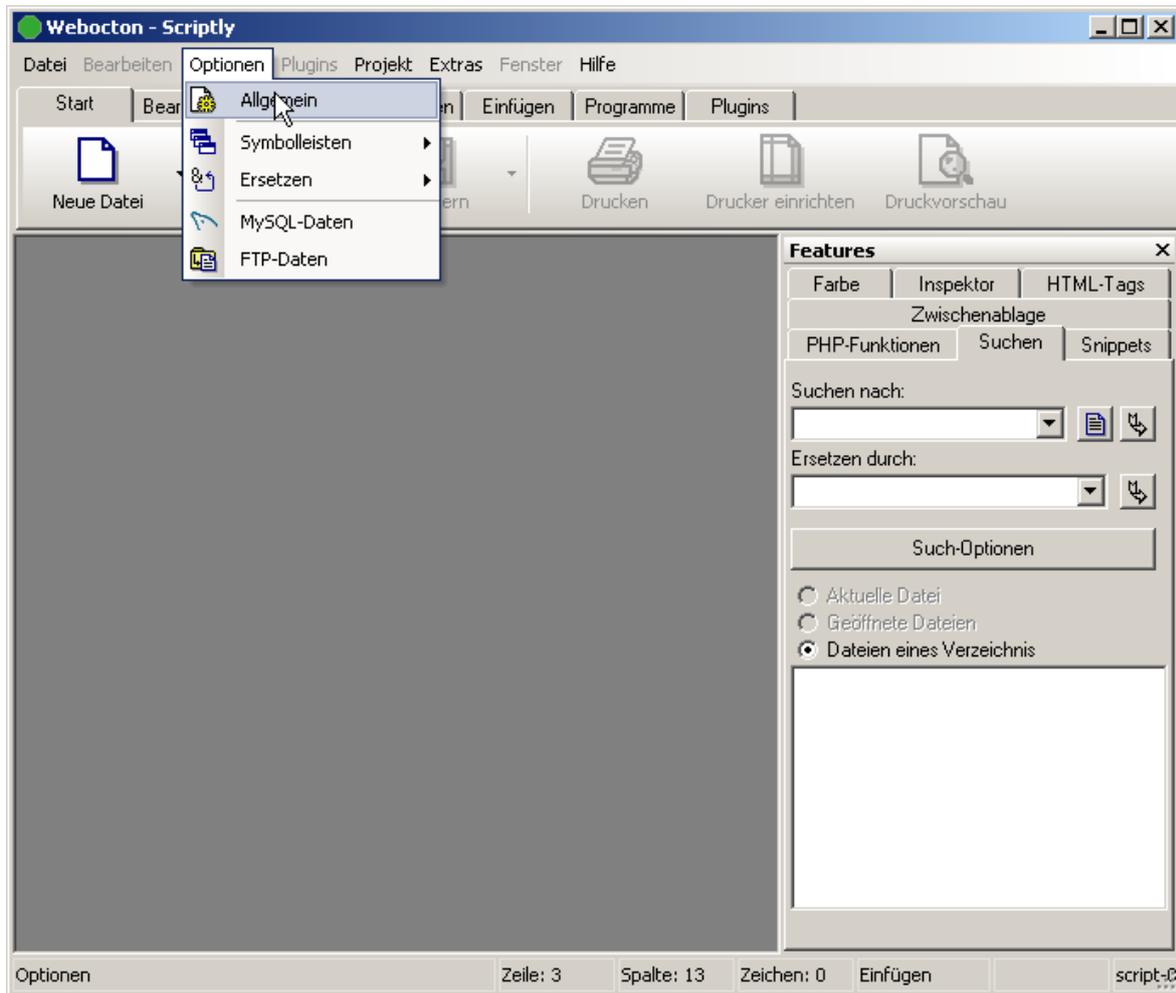
Das Ganze hat auch seinen Preis - mehr Informationen und eine Testversion finden Sie unter: www.zend.com/de/

Konfiguration von Scriptly für PHP-Programmierung

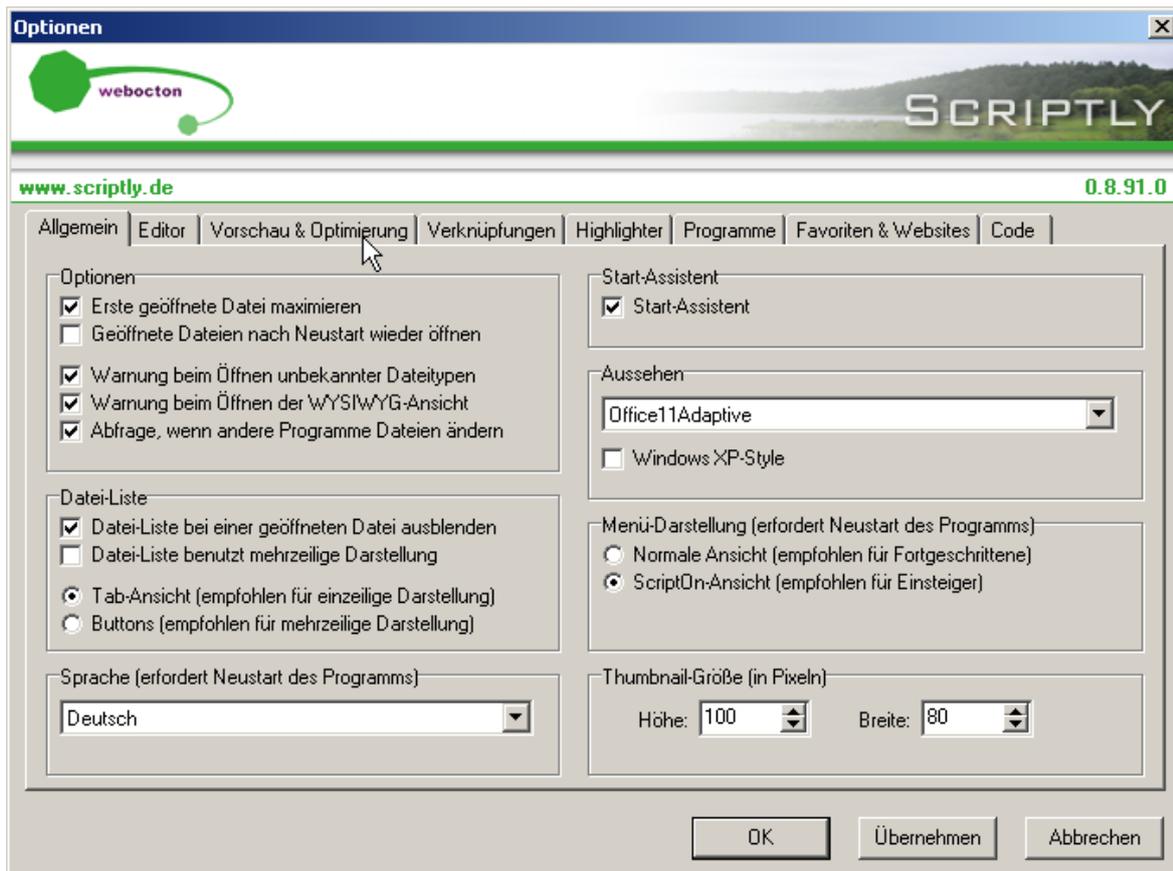
Bei Scriptly werden für die Ausführung von PHP Einstellungen benötigt, damit ein direktes Ausführen von PHP-Programmen aus dem Editor heraus durchgeführt werden kann.

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

In Scriptly über Menü "Optionen" den Punkt "Allgemein"



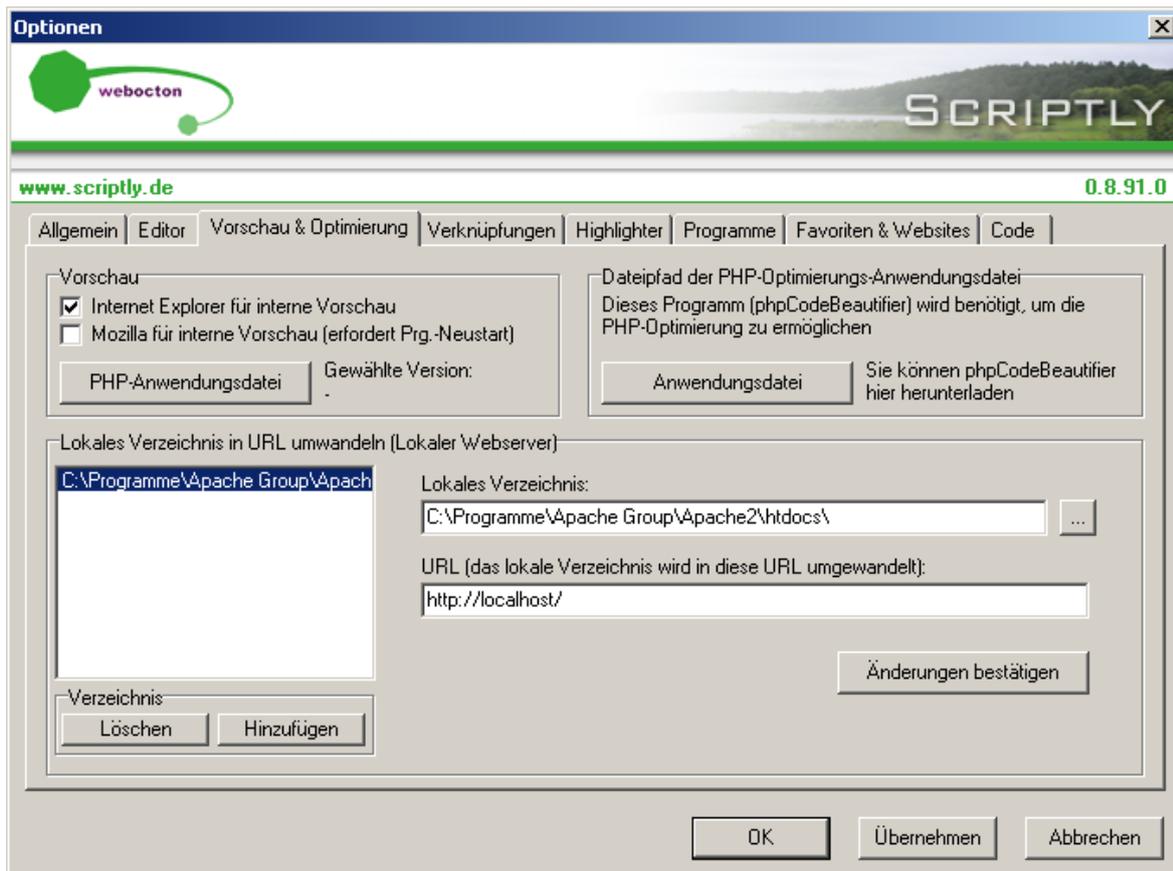
und dann den Reiter "Vorschau & Optimierung"



Und dort muss im Feld "Lokales Verzeichnis" der Speicherort hinterlegt werden, in dem Ihre PHP-Programm ausgeführt werden. I.d.R. ist es das htdocs-Verzeichnis des Apache, wenn dieser eingesetzt wird.

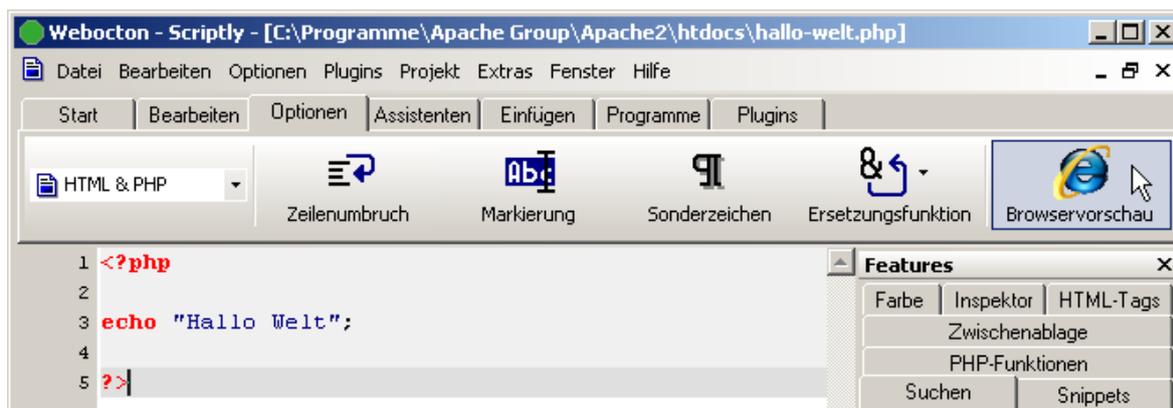
Unter "URL" kommt dann "http://localhost/"

Wichtig ist, dass Sie auf "Änderungen bestätigen" klicken!



Wird nun in Scriptly ein PHP-Programm erstellt (und abgespeichert), kann dieses PHP-Programm direkt ausgeführt werden.

Im Reiter Optionen gibt es den Button "Browservorschau".



Nach Anklicken erscheint die Vorschau - die URL wurde geändert, so dass der Aufruf über den Webserver erfolgt.



Sie können die Ausgabe aktualisieren lassen, wenn Sie den Reload Button anklicken.

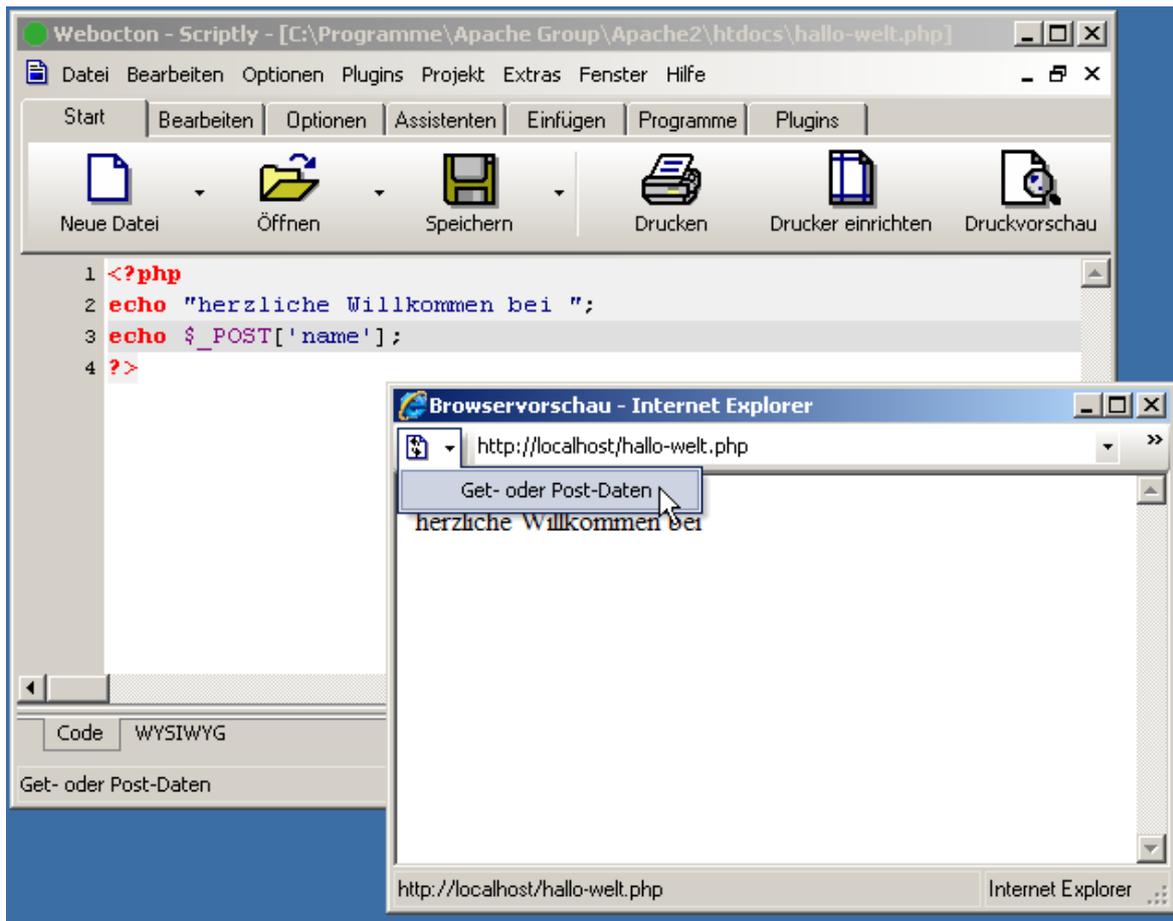


Tipp: Browservorschau

Die Browservorschau kann über die Taste F9 direkt aufgerufen und mit F10 dann wieder geschlossen werden.

Tipp: Formularentwicklung

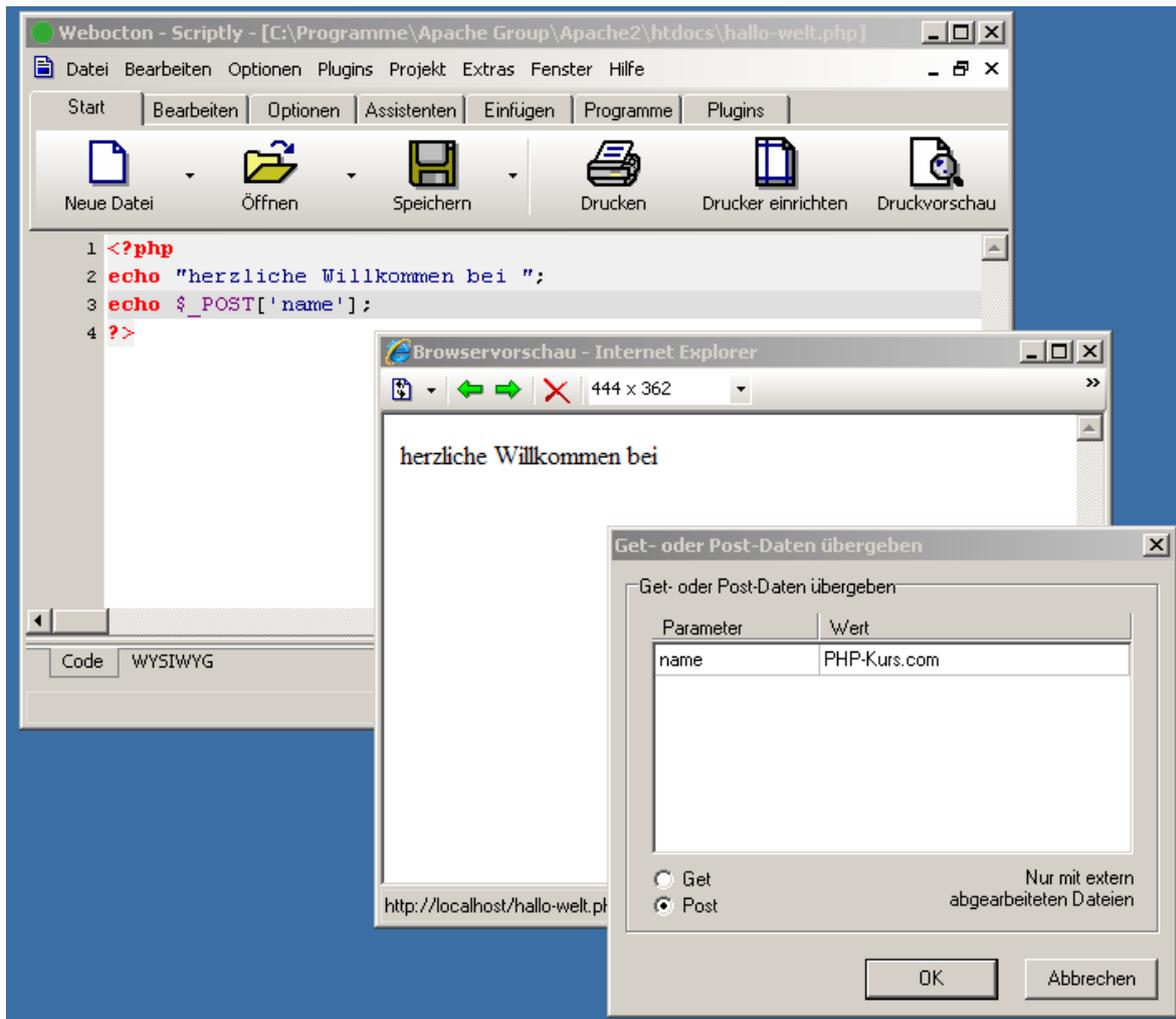
Wenn Sie mit Formularen hantieren, die ihre Werte über POST und GET versenden, können Sie damit bequem in der Vorschau spielen:

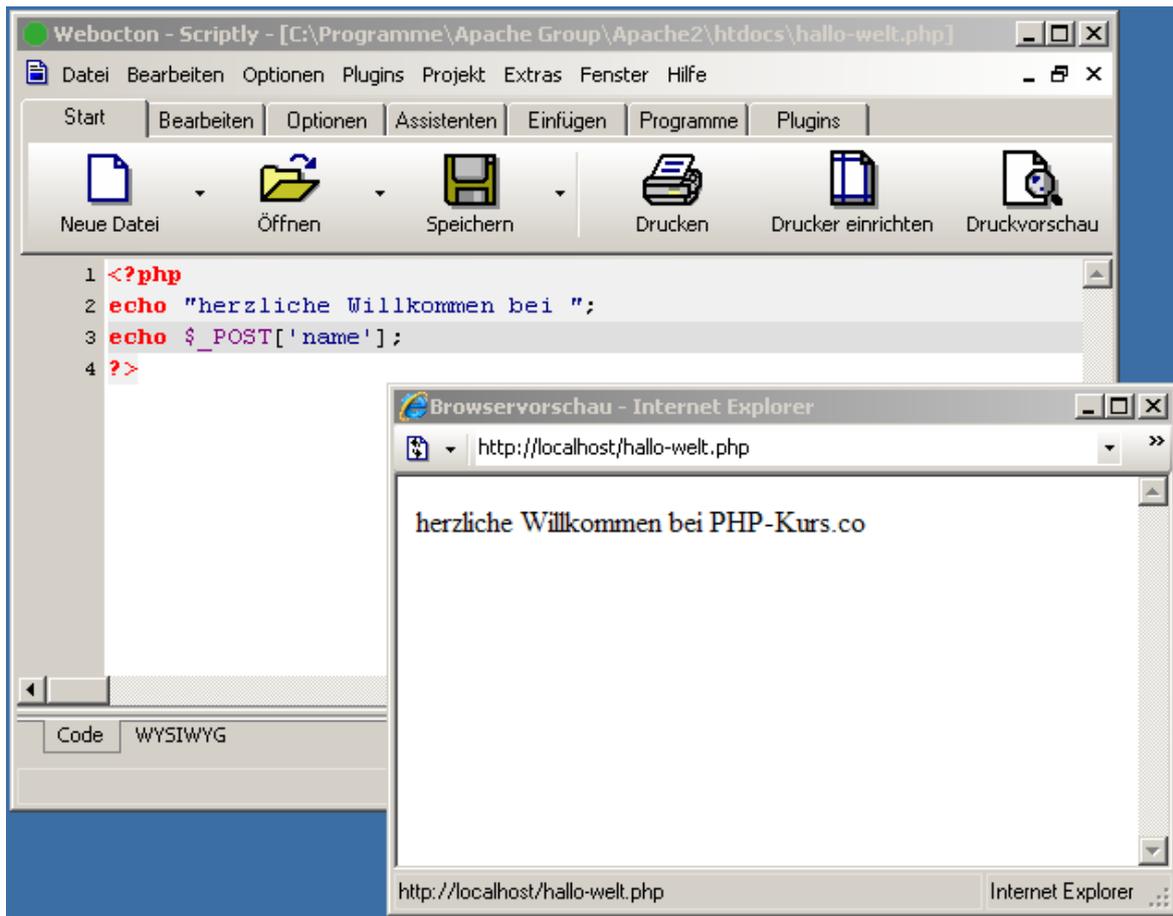


Aufruf der Vorschau und in der Vorschau oben die Eingabemaske für die GET bzw. POST-Werte



Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen





FTP-Programme

Um die erstellten Programm von Ihrem Rechner auf den Webserver zu bekommen, wird ein FTP-Programm benötigt. Dieses ist für den Transport der Dateien zuständig. Dabei gibt es sehr unterschiedliche.

Folgende zwei Programme wurden ausgewählt, da beide Freeware sind und beide eine geschützte Übertragung per sFTP ermöglichen (beides keine Selbstverständlichkeit).

FTP-Programm WinSCP

Im Kurs (und von der Uni-Tübingen aus) wird WinSCP genutzt, dass auch sFTP unterstützt. Im Kapitel "Einloggen auf dem Webserver" finden sie die entsprechende Anleitung für WinSCP.

Sie finden einen Downloadmöglichkeit unter: <http://winscp.net/eng/download.php>

Für unseren Zweck ist eine "Anleitung zur Konfiguration und dem Benutzen" erstellt.

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

FileZilla

Freeware, entwickelt von Tim Kosse und unter Sourceforge herunterladbar:
<http://sourceforge.net/projects/filezilla/>

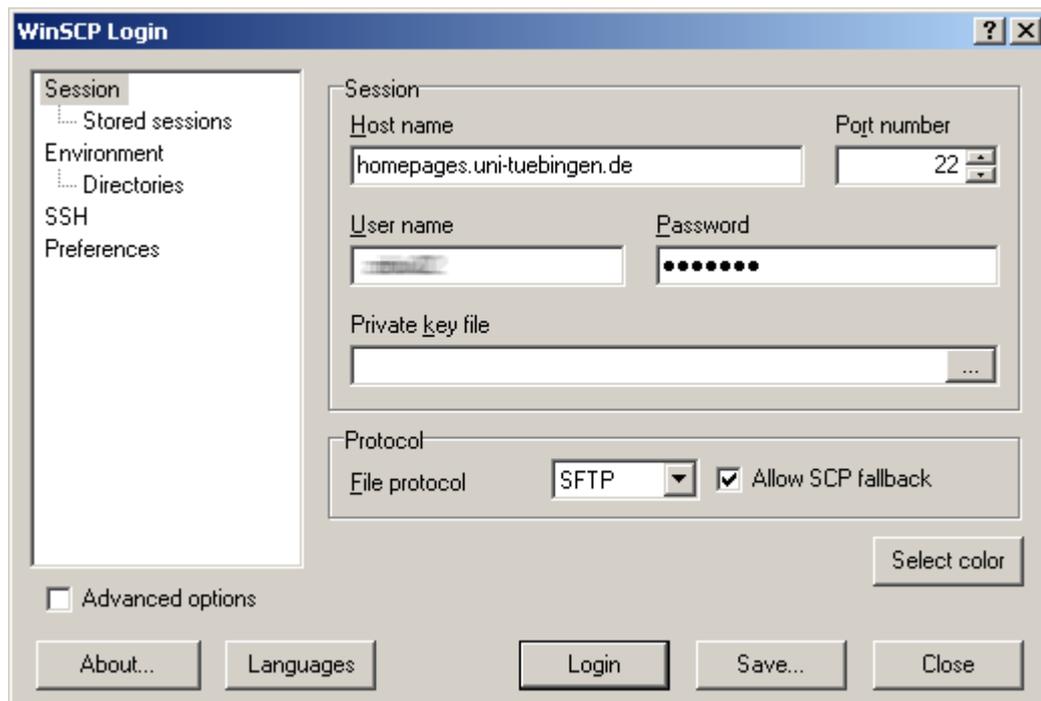
Dateien per sFTP auf Webserver übertragen

Was im folgenden für die Uni-Homepages von Tübingen beschrieben wird, gilt i.d.R. auch für andere Hoster. Wobei es bei vielen Hoster es keine Rolle spielt, ob man sich verschlüsselt über sFTP einloggt oder unverschlüsselt über FTP. Bei den Uni-Homepages ist sFTP Pflicht. Von Ihrem Provider erhalten Sie die FTP-Zugangsdaten.

Um sich auf den Webserver der Uni-Homepages einzuloggen, muss eine sichere Verbindung aufgebaut werden. Dazu wird gerne das Programm WinSCP verwendet, da dieses sFTP (Secure-FTP) unterstützt und kostenfrei ist.

Sie finden einen Downloadmöglichkeit unter: <http://winscp.net/eng/download.php>

Nach der Installation erscheint folgender Startbildschirm von WinSCP.



Tragen Sie hier unter "Host name" ein: `homepages.uni-tuebingen.de` und als "Port number" 22

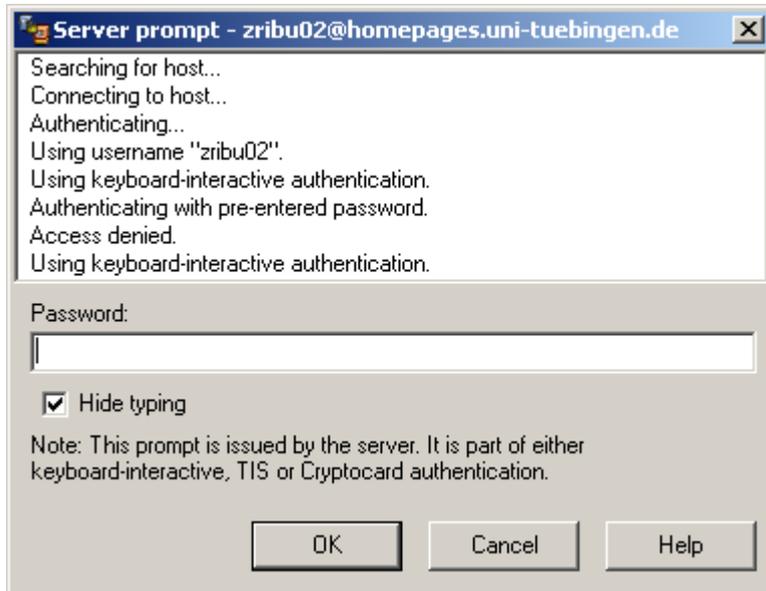
Als "User name" tragen Sie für den Zugang an der Universität Tübingen Ihre Login-Id (für Studenten i.d.R. zx...).

Wenn Sie bei einem kommerziellen Provider sind, erhalten Sie von diesem Ihre ftp-

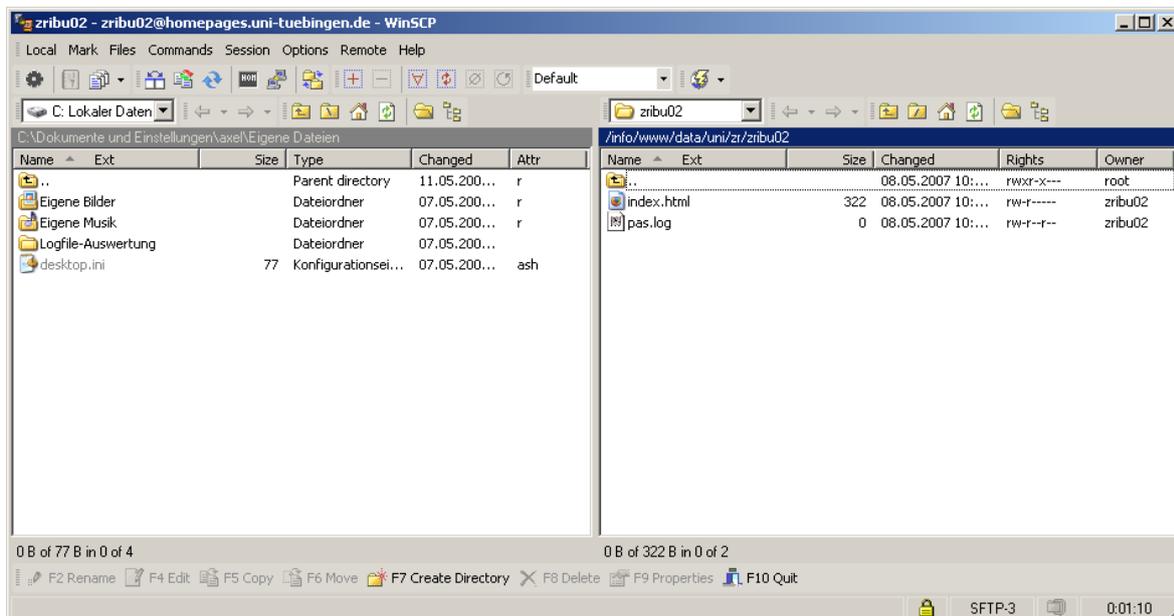
Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Zugangsdaten. Beachten Sie dabei, dass nicht alle Provider einen sFTP-Zugang unterstützen. Da heißt die Devise probieren und fragen.

Passwort eingeben - falls Sie sich vertippen, werden Sie während des Einloggvorgangs nochmals gefragt



Nach dem Einloggen erscheint das Dateien-Fenster. Auf der linken Seite sehen Sie die Inhalte Ihres lokalen Computers - auf der rechten Seite befindet sich der Inhalt des Webservers (also Ihrer Homepage).



Jede Website braucht eine Startseite. Diese hat hier den Namen "index.html", kann aber genauso auch "index.htm" heißen, bzw. wenn es bereits eine PHP-Seite ist, dann "index.php".

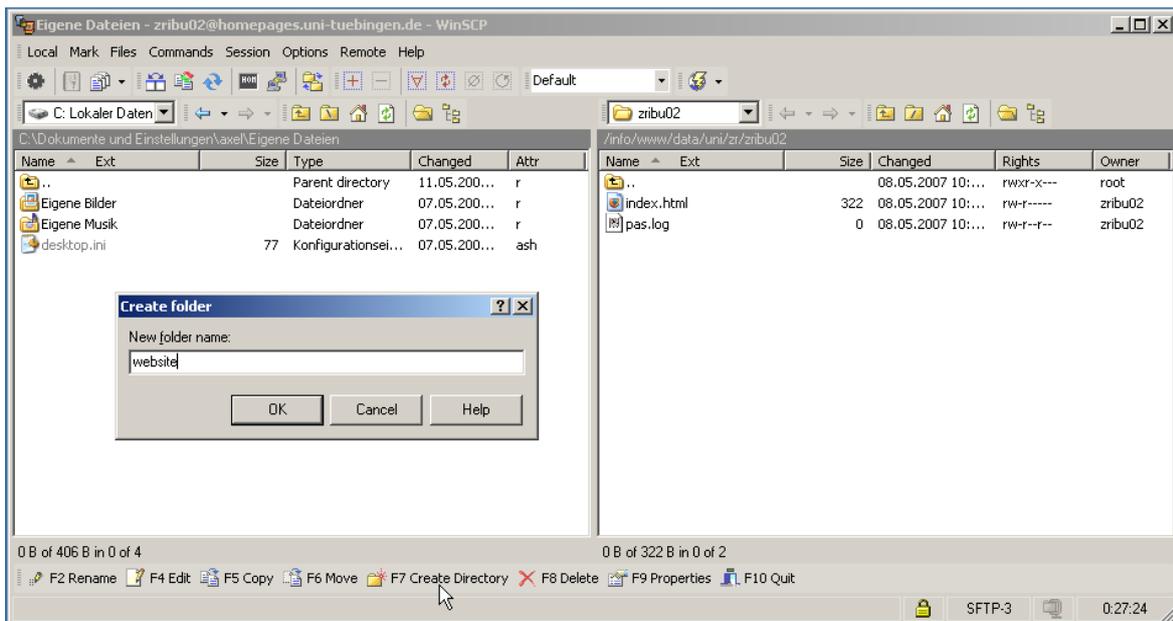
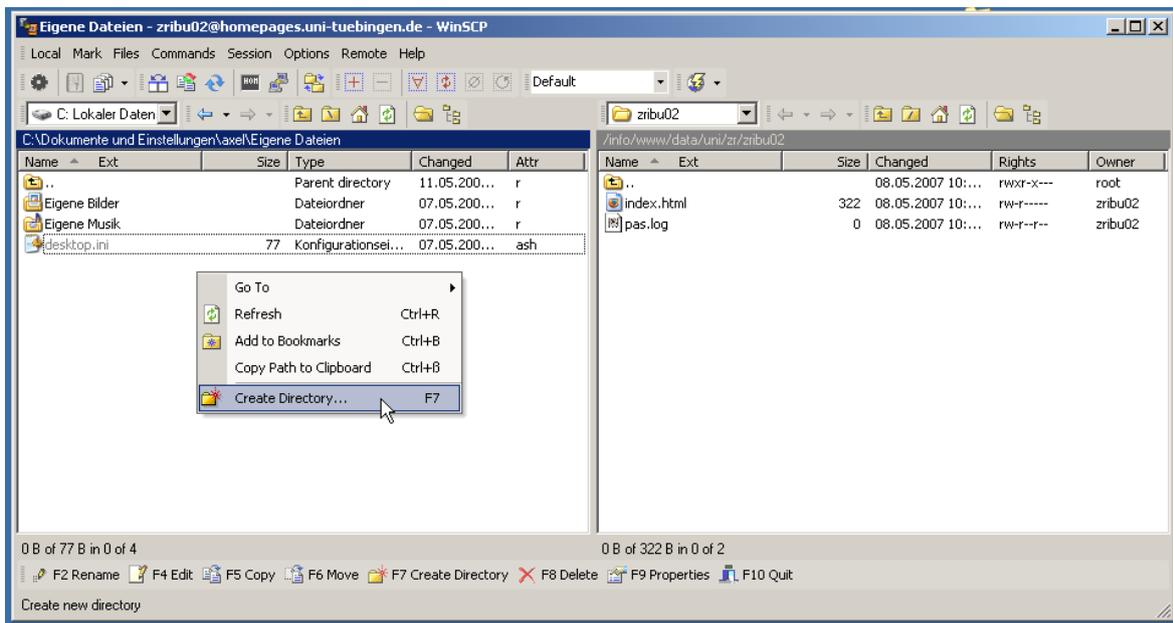
Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Anmerkung zu den Uni-Seiten Tübingen: Hier hat die Datei "index.html" Vorrang vor "index.htm" - gibt es keine "index.html", wird die "index.htm" genutzt.

Einrichten Dateistruktur

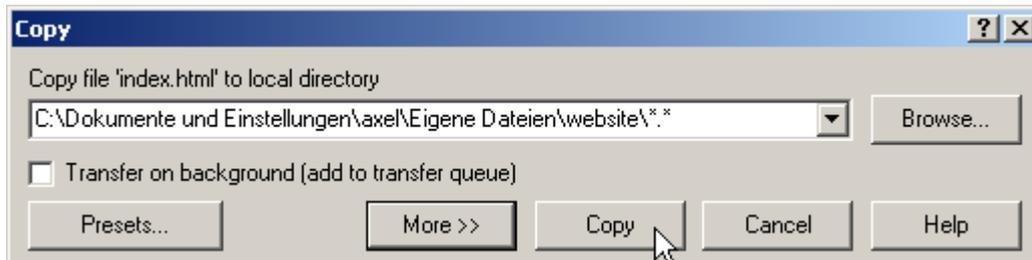
Auf Ihrem lokalem Rechner richten Sie sich am besten ein eigenes Verzeichnis für Ihre Website an - im Beispiel hier nennen wir es "website".

Im linken Ordner entweder reinklicken und dann "F7 Create Directory", oder die rechte Maustaste im linken Bereich drücken und "Create Directory" auswählen. Jetzt werden Sie nach dem Verzeichnis-Namen gefragt - hier nennen wir das Verzeichnis "website".



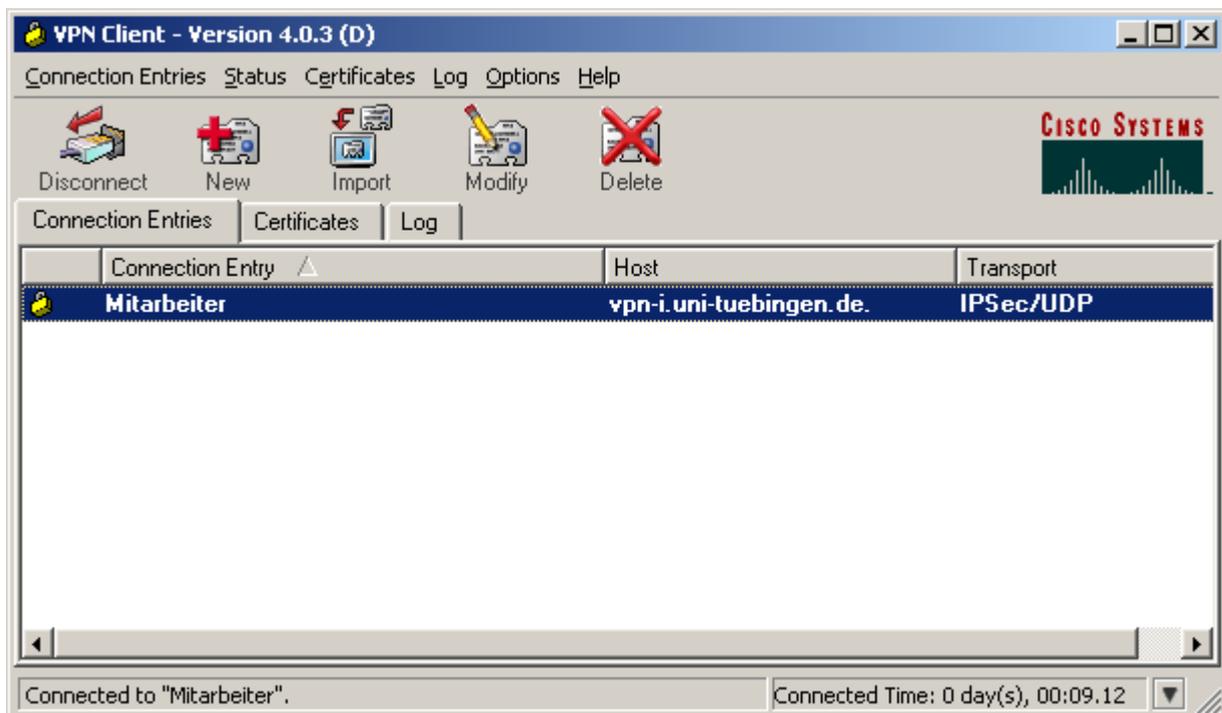
Zum Kopieren von Dateien klicken Sie eine Datei an und ziehen Sie in die andere Seite (z. B. von rechts nach links - Server nach lokales Verzeichnis). Dies kann auch mit mehreren Dateien gleichzeitig geschehen oder mit Verzeichnissen.

Es erfolgt eine Abfrage, was geschehen soll - klicken Sie auf Copy. Ist die Datei bereits vorhanden, werden Sie darauf hingewiesen.



Anmerkung zu Webserver Uni-Tübingen

Eine Besonderheit bei der Uni Tübingen ist, dass ein Einloggen von außen per WinSCP nicht ohne VPN-Client möglich ist. Also erst den VPN-Client installieren und einrichten und dann WinSCP.



Benötigte HTML-TAGs

Für den PHP-Kurs werden eine Handvoll HTML-TAGs benötigt. Umfangreiche Erklärungen finden Sie unter www.html-seminar.de, www.html-tag.de und <http://www.webmaster-crashkurs.de/>

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

HTML-TAGs zur Formatierung von Text

`<h1> ... </h1>`
Wichtigste Überschrift

`<h2> ... </h2>`
Überschrift vom zweiten Rang

`<p> ... </p>`
Absatz - alles zwischen dem `<p>`-TAG ist ein Absatz, danach gibt es Abstand zum nächsten Element

`
`
Harter Umbruch - der Inhalt geht dann in der nächsten Zeile weiter

`<hr />`
Zeichnet eine horizontale Linie ein

`hier klicken`
Link zum Anklicken

HTML-TAGs für Formulare

Ab dem Kapitel Formularauswertung wird benötigt:

`<form action="" method="" target=""> ... </form>`
Zwischen diesen TAGs werden die einzelnen Formularfelder eingeschlossen

`<input type="Text" name="" value="" size="" maxlength="" />`
Eingabefeld für 1 Textzeile

`<input type="hidden" name="" value="" size="" maxlength="" />`
verstecktes Feld

`<textarea name="" cols="10" rows="60"></textarea>`
mehrzeiliges Feld für Eingabe von viel Text

`<input type="Checkbox" name="" value="v" />`
Anklickbox

`<input type="Submit" name="" value="speichern" />`
Button zum Anklicken und Absenden des Formulars

Internet Browser

Zum Entwickeln und Testen empfiehlt sich Firefox aus verschiedenen Gründen.

- der HTML-Quellcode kann selektiv angezeigt werden, es muss nicht immer die ganze Seite durchgescrollt werden
- Plugins für den Firefox erleichtern die Arbeit und das Testen und Debuggen

Firefox Erweiterung für die PHP Entwicklung

Testen Sie einfach das Plugin [FirePHP](#) - eine Firefox Erweiterung für die PHP Entwicklung

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Webspace mit PHP-Unterstützung

Wenn Sie Ihre fertigen PHP Programme dann im Internet laufen lassen möchten, benötigen Sie einen Webspace-Provider (Hoster), der PHP unterstützt.

Im folgenden eine kleine (und unvollständige) Auflistung:

- [Strato](#) (ab PowerWeb oder mit Zusatzmodul Construction)
- [1&1](#) (ab Homepage Perfect)
- [Goneo](#) (ab Homepage Start)

Wichtig bei den Providern ist, das mindestens PHP4 (besser natürlich PHP5) unterstützt wird und MySQL-Datenbanken. Immer bei den Paketen genau nachsehen - die Informationen sind gerne zu finden unter einem Link in der Richtung "Weitere Infos", "Jetzt informieren" oder "mehr Infos".

Wenn Sie über einen von den obigen Links sich Webspace bestellen sollten, spendieren Sie mir indirekt ein Getränk (keine Angst, es kostet Sie weder mehr, noch weniger). Danke im Voraus :)

Alternativen zu XAMPP

Sollte es Probleme mit XAMPP geben (was sehr selten der Fall sein kann - bei über 50 Installationen hatte ich bisher nur auf einem Gerät ein Problem) gibt es Alternativen. Sollte einem XAMPP einem nicht liegen, dann sind die Alternativen natürlich auch interessant.

Hier liegt der Schwerpunkt auf Apache, PHP und MySQL **unter Windows**

WampServer	http://www.wampserver.com/en/index.php
XAMP / XAMPP Lite	http://www.apachefriends.org/de/xampp.html
PHPTriad	http://sourceforge.net/projects/phptriad/
Webserver On Stick (WOS)	http://www.chsoftware.net/de/useware/wos/wos.htm
BigApache	http://sourceforge.net/projects/bigapache/
EasyPHP	http://www.easyphp.org/
FoxServ	http://www.foxserv.net/

Ausführlich getestet wurde der [WampServer](#) (und für gut empfunden). Die Internetseite ist zwar komplett auf Englisch (bzw. Französisch) aber nach dem Download kann der WampServer auch auf Deutsch umgestellt werden. Es gibt zahlreiche weitere Sprachpakete standardmäßig mit an Board vom WampServer, z.B. Catalanisch, Griechisch, Russisch, Schwedisch, Türkisch, Italienisch und mehr.

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Einsteiger

Vor jedem Beginn steht die Unsicherheit. Um die eigenen Fehler so gering wie möglich zu halten, versuchen Sie den Quellcode mit den entsprechenden Einrückungen zu versehen, damit eine Übersicht entsteht. Ansonsten werden Sie schnell vor lauter Quellcode keine Fehler mehr finden.

Quellcode übersichtlich strukturieren

Für den Anfang ist es ratsam, nur einen Befehl pro Zeile zu verwenden

Für das Programm macht es keinen Unterschied, nur für Sie und den Spaß, der einem nach stundenlangem Fehlersuchen vergehen kann, wenn man über seine eigene Schusseligkeit stolpert!

Negativbeispiel;

```
echo "Hallo "Welt"". $name ." - i'am feeling "; echo "fine";

function ausgabe ($wert) { echo $wert; }
```

Und das ganze übersichtlicher

```
echo "Hallo "Welt"";
echo $name;
echo " - i'am feeling fine";

function ausgabe ($wert)
{
    echo $wert;
}
```

Unser erstes PHP-Programm

"Hallo Welt" als erstes eigenes PHP-Programm

PHP-Code steht immer zwischen `<?php` und am Ende `?>`

Alles was davor oder danach kommt, wird als HTML interpretiert.

Daher benötigen wir als erstes diesen "Rahmen" um unser PHP-Programm

```
<?php
?>
```

Als ersten Befehl lernen wir den Ausgabebefehl `echo` kennen. Dieser bewirkt eine Ausgabe auf dem Bildschirm. Der auszugebende Text kommt dann nach dem `echo` in Anführungszeichen.

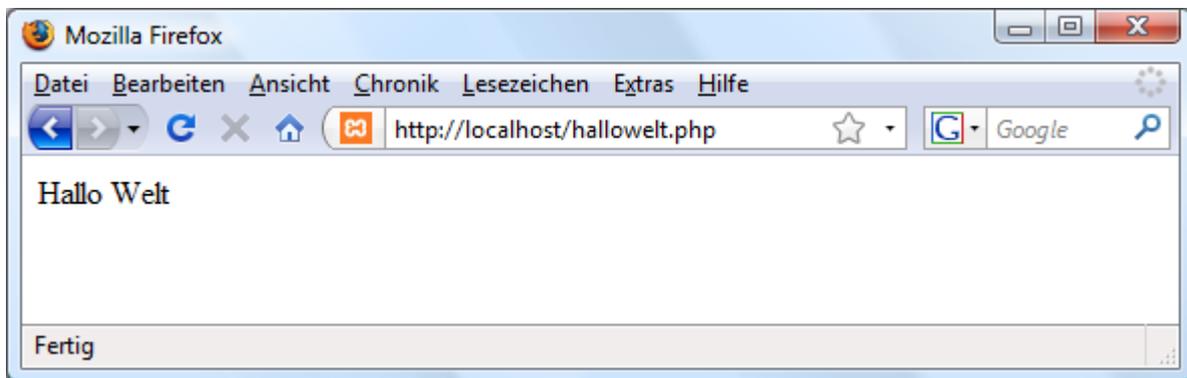
Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Jede Befehlsfolge wird mit einem Semikolon abgeschlossen. Das ist extrem wichtig und das vergessene Semikolon stellt einen beliebten Fehler dar!

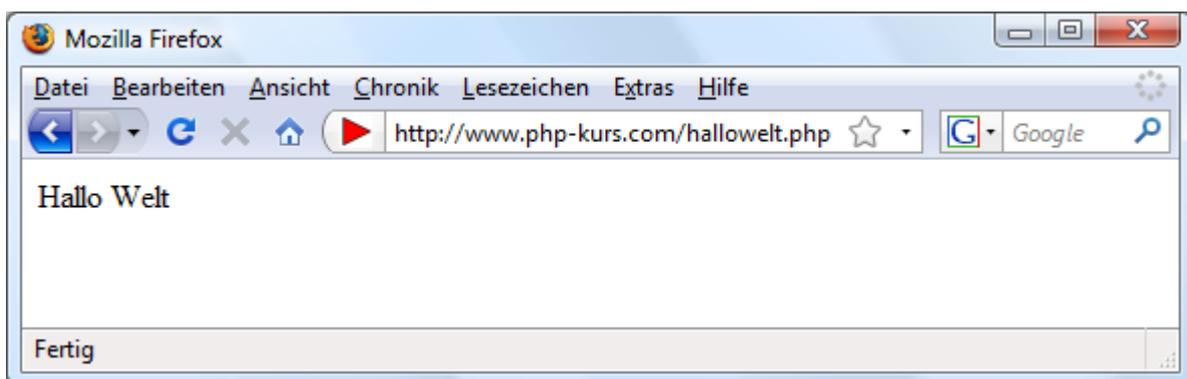
```
<?php
echo "Hallo Welt";
?>
```

Unser erster Programm speichern wir nun im entsprechenden Verzeichnis (bei einer lokalen Installation von XAMPP im htdocs-Verzeichnis). Als Dateiname vergeben wir "hallowelt.php". Wichtig ist die Dateierdung **.php**

Zum Ausführen starten wir einen Browser (egal ob Firefox, Internet Explorer oder Opera) und geben als URL dann **http://localhost/hallowelt.php** ein. Wichtig ist, dass beim lokalen Computer XAMPP läuft.



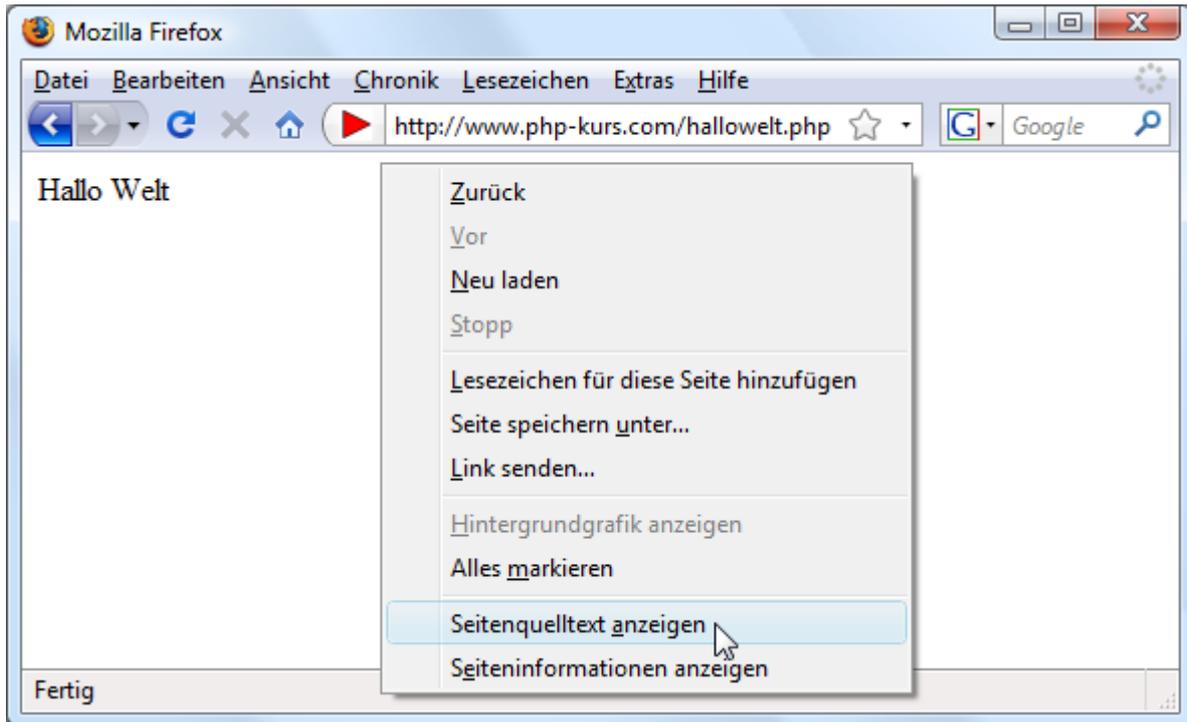
Wird das Programm direkt auf einen Webserver hochgeladen, der PHP unterstützt, läuft es natürlich nach Aufruf direkt, z.B. <http://www.php-kurs.com/hallowelt.php>



Interessant ist es nun, im Webbrowser in den Quellcode zu sehen - also rechte Maustaste und dort "Seitenquelltext ansehen".

In der Ausgabe im Browser sind keine PHP-Befehle mehr zu sehen, diese werden vom PHP-Interpreter ausgeführt und der Browser erhält nur noch das fertige Ergebnis.

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen



PS: Es fehlt der normale Aufbau für eine XHTML-konforme Seite. Zum Lernen ist das OK, aber für später muss der entsprechende Kopf und Fuß integriert werden (im HTML).

Ausgabe in Browser

Hallo Welt in PHP

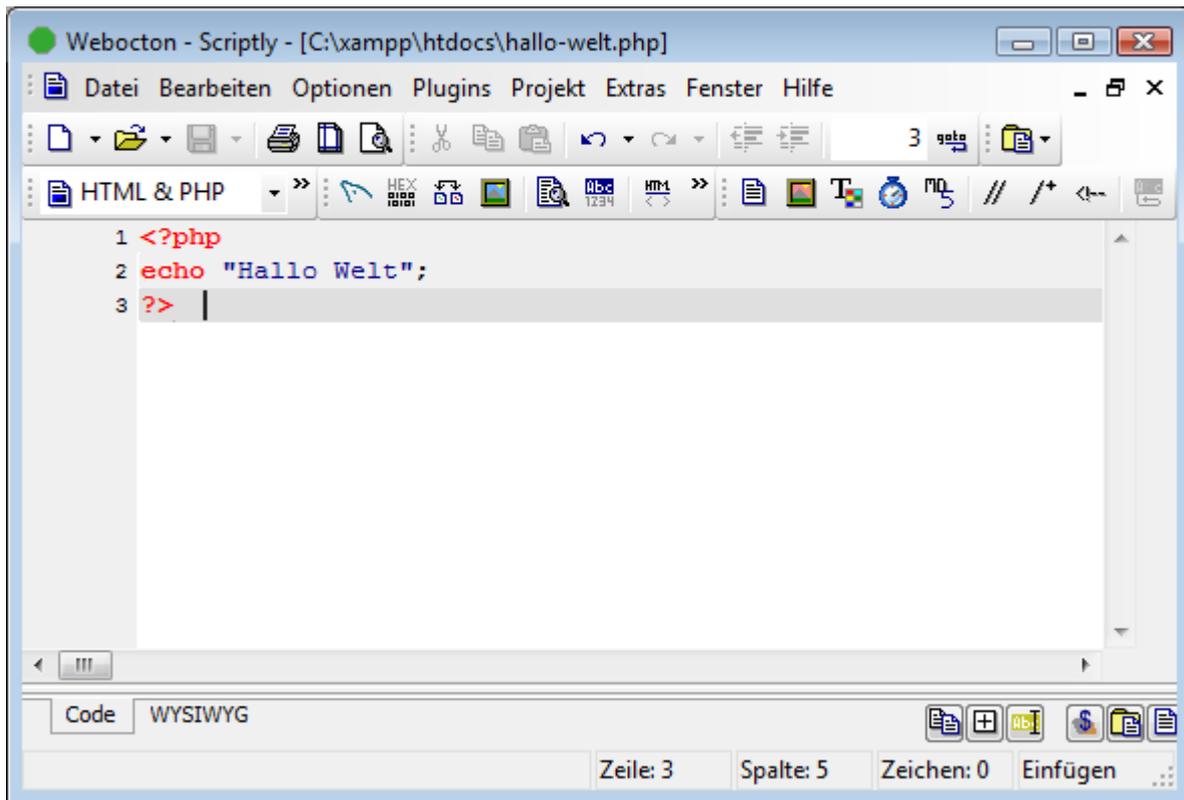
Der Klassiker in allen Programmiersprachen - die Ausgabe von "Hallo Welt" auf dem Bildschirm.

In PHP ist das ein Einzeiler:

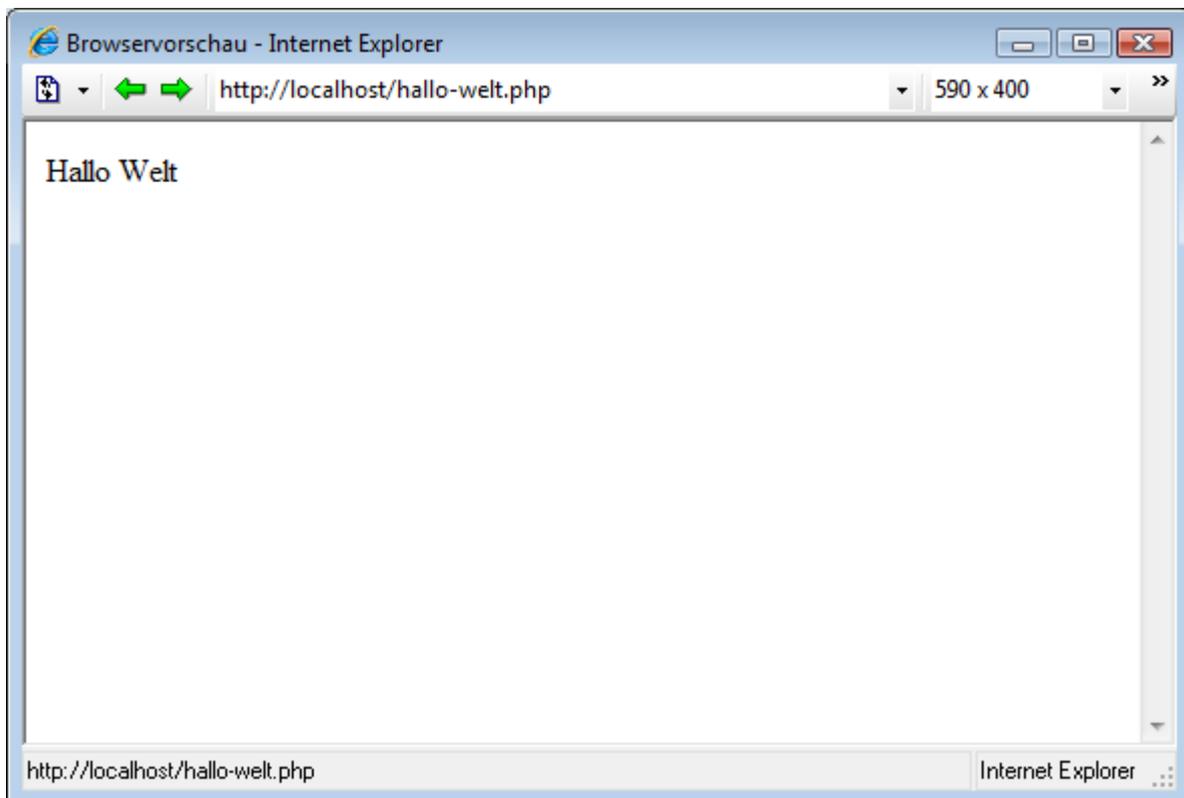
- `echo` ist der Befehl, dass etwas ausgegeben wird
- alles zwischen den Anführungszeichen wird ausgegeben
- Am Ende zum Abschluss ein Semikolon

That's it - selber probieren!

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen



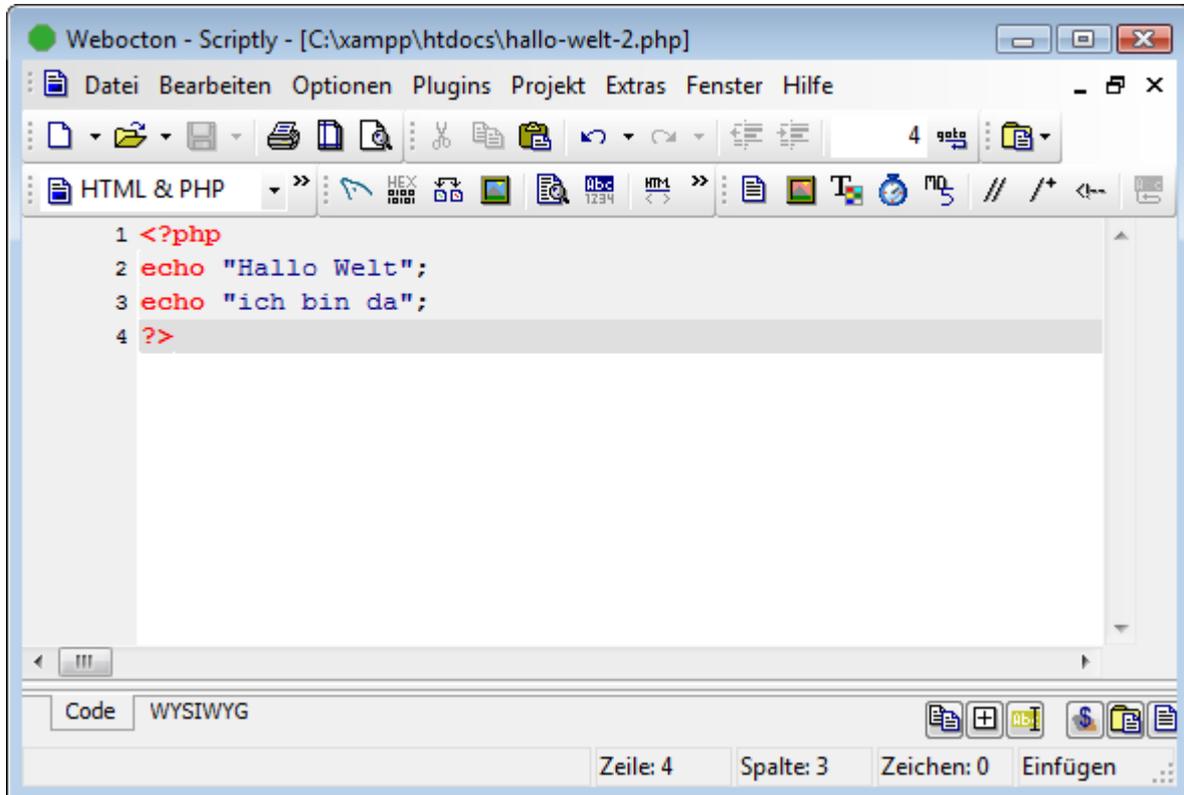
Als Ausgabe (hier der Bequemlichkeit wegen in der Vorschau von Scriptly mit F9, das Ergebnis ist identisch mit dem Internet-Explorer):



Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Für die Ausgabe im Browser gibt es also den Befehl `echo` - dies ist bestimmt einer der am meisten verwendeten Befehle.

Im Folgenden wird zweimal der `echo`-Befehl genutzt:



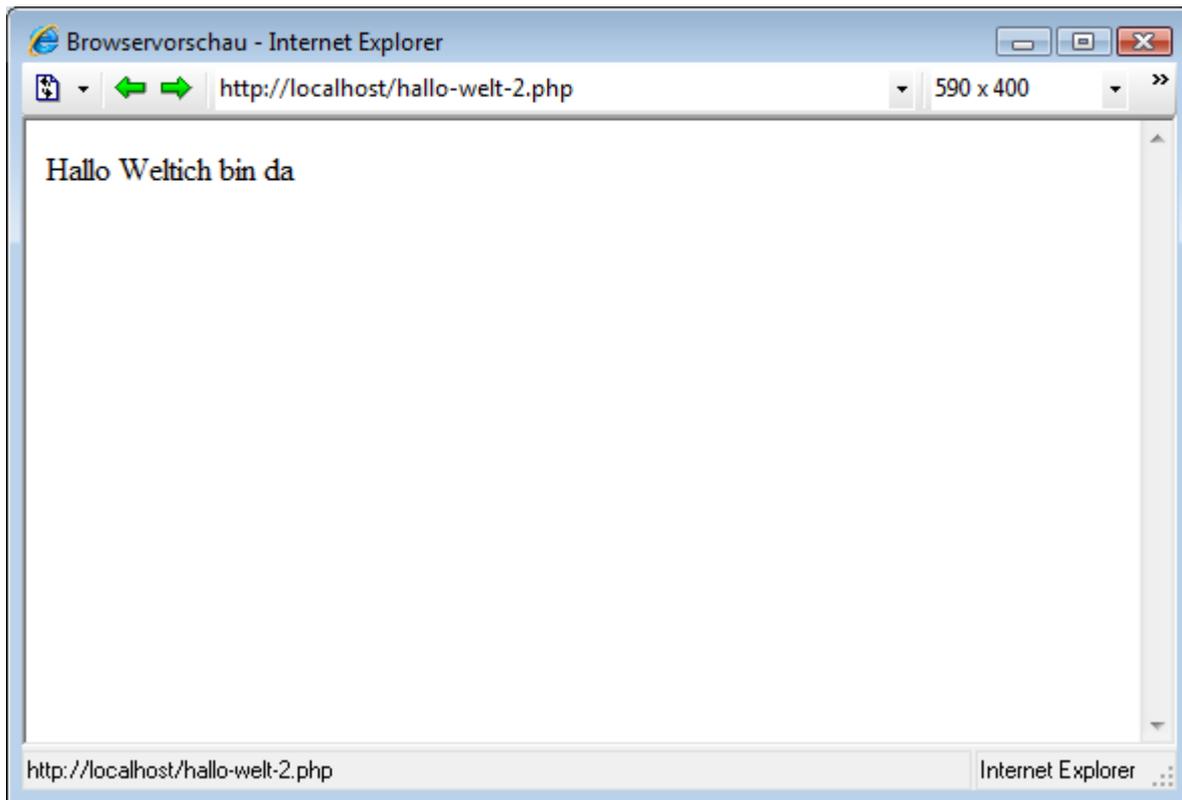
The screenshot shows a code editor window titled "Webtocton - Scriptly - [C:\xampp\htdocs\hallo-welt-2.php]". The editor contains the following PHP code:

```
1 <?php
2 echo "Hallo Welt";
3 echo "ich bin da";
4 ?>
```

The editor interface includes a menu bar with "Datei", "Bearbeiten", "Optionen", "Plugins", "Projekt", "Extras", "Fenster", and "Hilfe". Below the menu is a toolbar with various icons for file operations and editing. The status bar at the bottom indicates "Code" mode, "WYSIWYG" view, and "Zeile: 4", "Spalte: 3", "Zeichen: 0".

In der Ausgabe sehen Sie, dass ein Leerzeichen fehlt und kein Umbruch etc. gemacht wird - das ist für die in HTML stattfindende Ausgabe korrekt.

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen



Wenn Sie nach "Welt" ein Leerzeichen wollen, müssen Sie es vor dem Abführungszeichen einfügen; einen Umbruch erhalten Sie durch den HTML-TAG `
`

Sonderzeichen maskieren

Ausgabe von Anführungszeichen und Sonderzeichen

Wenn Sie nun ein Anführungszeichen ausgeben möchten, klappt dies nicht so einfach, da Ihr PHP-Befehl `echo "Hallo Welt heute";` die Anführungszeichen als Anfang und Ende von dem auszugebenden Bereich ansieht. Sollen also im Beispiel "Hallo Welt heute" Anführungszeichen vor und nach dem Wort "Welt" ausgegeben werden, müssen dieses Anführungszeichen speziell geschrieben werden - man spricht von maskieren.

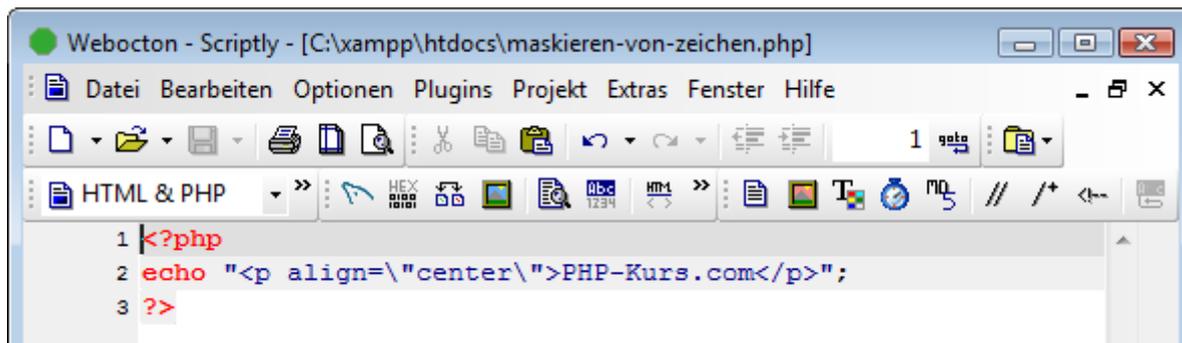
Bestimmte Zeichen müssen maskiert werden - als Beispiel soll ein Anführungszeichen innerhalb einer Ausgabe ausgegeben werden.

Das wäre dann `echo "Hallo "Welt" heute";`.

Als weiteres Beispiel wird beim align das "center" mit den Anführungszeichen versehen, was einer korrekten HTML-Schreibweise entspricht, wenn auch veraltet (zum PHP lernen hier ist es OK, für später dann bitte mit CSS machen!).

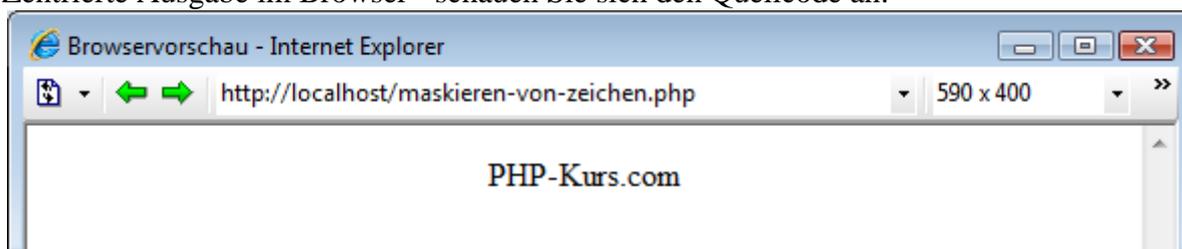
Dazu wird vor dem Zeichen eine Backslash (`\`) eingetragen.

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

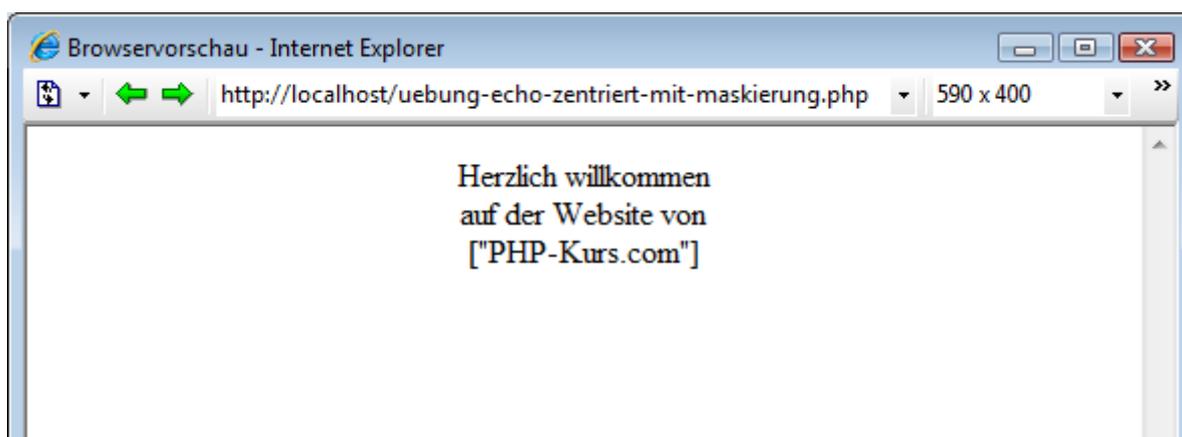


```
1 <?php
2 echo "<p align=\"center\">PHP-Kurs.com</p>";
3 ?>
```

Zentrierte Ausgabe im Browser - schauen Sie sich den Quellcode an.

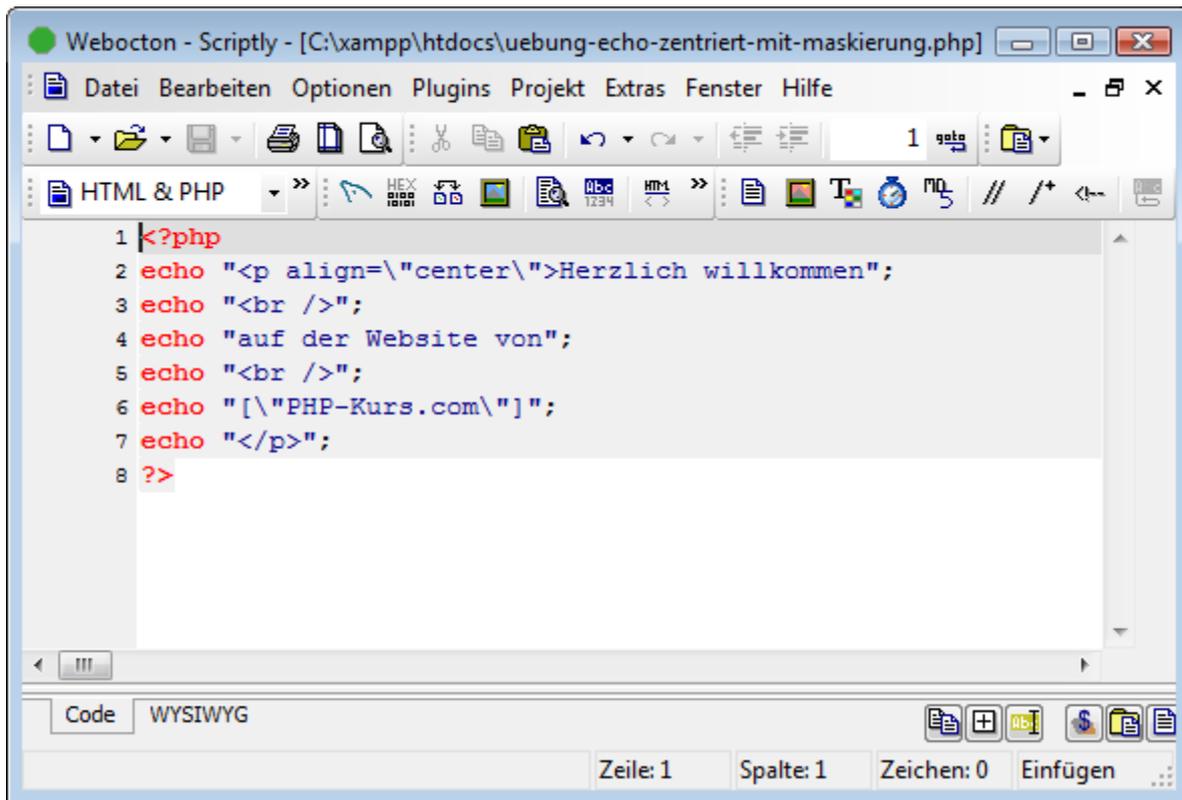


Aufgabe: Bauen Sie folgende Ausgabe nach



Eine Lösung zum Vergleichen mit Ihrem Ansatz finden Sie auf der nächsten Seite:

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen



Auskommentieren von Code

PHP-Quellcode "abschalten"

Wie im letzten Beispiel schon verwendet, gibt es in PHP die Möglichkeit, Bereiche auszukommentieren. Das bedeutet, alles zwischen den entsprechenden Zeichen wird von PHP ignoriert und ist für das Programm nicht existent.

Dazu wird zum Starten `/*` und am Ende dann `*/` verwendet. Dies kann auch über mehrere Zeilen geschehen. Dadurch können Programmteile "abgeschaltete" werden - im folgenden Beispiel keine Ausgabe von nichts :)

```
/*
echo "Hallo Nichts";
echo "es kommt keine Ausgabe";
*/
```

Soll nur eine Zeile auskommentiert werden, am Anfang einfach ein `//` und die Zeile ist wirkungslos.

```
// echo "es kommt keine Ausgabe";
```

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Kommentierung des Programmes

Je komplexer eine Anwendung ist, desto sinnvoller ist die Kommentierung, was an welchem Eck passiert.

Sinnvoll ist es, nach dem `;` die `//` zur Kommentierung des Programms zu verwenden:

```
altersdiv ($jahrgang); // Aufruf der Alterberechnung
```

Beispiele für Auskommentieren

Quellcode, der zeitweise nicht benötigt wird, kann auskommentiert werden, wenn dieser zwischen `/*` und `*/` gepackt wird.

Beispiel:

```
<?php
$empfaenger = "du@testkarnickel.de"; //Mailadresse
$absender   = "ich@testkarnickel.de";
$betreff    = "PHP-Mail-Test";
$mailtext   = "Inhalt einer Mail zum Test von PHP";
$antwortan  = "ICH@testkarnickel.de";

/*
mail( $empfaenger,
      $betreff,
      "$mailtext",
      "From: $absender\nReply-To: $antwortan");

echo "Mail wurde gesendet!";
*/
?>
```

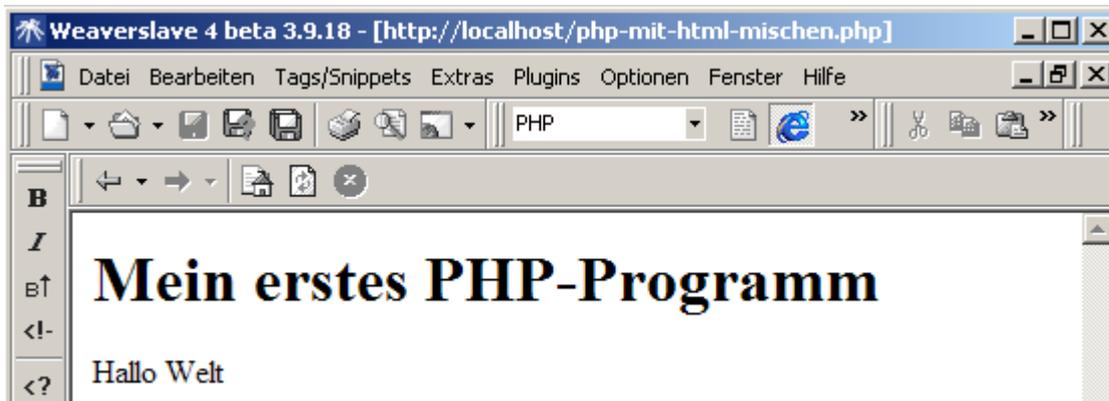
Die andere Art ist, dass jede Zeile einzeln mit `//` auskommentiert wird.

```
<?php
$empfaenger = "du@testkarnickel.de"; //Mailadresse
$absender   = "ich@testkarnickel.de";
$betreff    = "PHP-Mail-Test";
$mailtext   = "Inhalt einer Mail zum Test von PHP";
$antwortan  = "ICH@testkarnickel.de";

// mail( $empfaenger,
//       $betreff,
//       "$mailtext",
//       "From: $absender\nReply-To: $antwortan");
//
// echo "Mail wurde gesendet!";
?>
```

PHP und HTML mischen

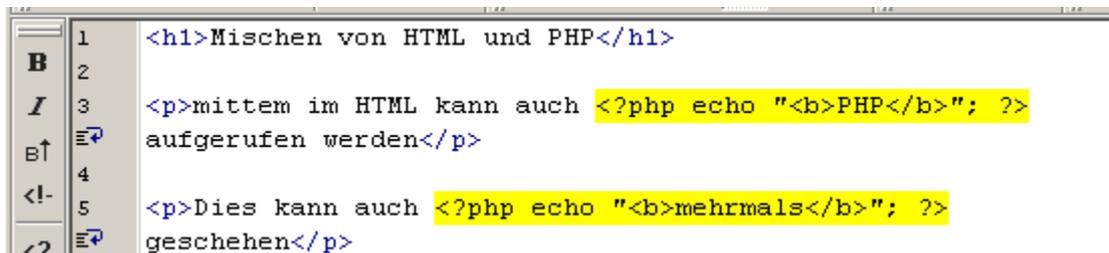
Wenn es zweckmäßig erscheint, kann HTML und PHP gemischt werden. Der PHP Bereich muss mit dem entsprechenden `<?php` beginnen und am Ende dann `?>`.



```
<h1> Mein erstes PHP-Programm </h1>

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

Es können auch mehrmals PHP-Bereiche in einem Dokument auftauchen



Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Variablen in PHP nutzen

Für eine Vielzahl von Anwendungen werden Variablen benötigt - Variablen sind Platzhalter für Inhalte, die teilweise auch erst während der Abarbeitung des Programmes bekannt werden (siehe nächstes Kapitel Berechnungen) bzw. sind nützlich, um ein variables Programm erstellen zu können (als Beispiel der Mehrwertsteuersatz als Variable - ändert sich nämlich von Zeit zu Zeit, auch wenn eine Bundesregierung vor der Wahl etwas anderes behauptet).

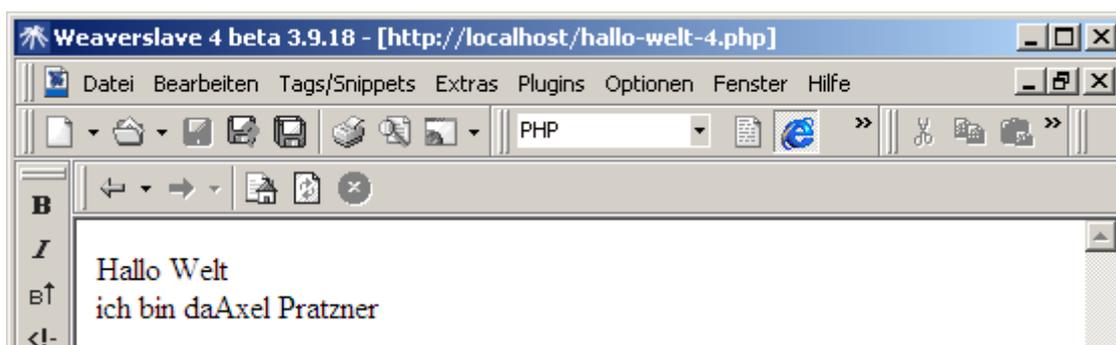
Variablen werden definiert, indem vor dem Namen der Variablen ein Dollarzeichen steht. Der Variablennamen sollte ohne Sonderzeichen (öäü..) und ohne Leerzeichen gewählt werden. Der Variable wird ein Wert zugewiesen, indem nach dem Variablennamen ein Gleichzeichen mit dem Inhalt kommt. Handelt es sich bei dem Inhalt um Texte, wird dieser in Anführungszeichen gesetzt. Ist der Inhalt numerisch, wird dieser einfach angegeben.

```
$name          = "Axel";  
$schuhgroesse = 43;
```

In einer Variable wird der Name gespeichert und ausgegeben.

```
<?php  
echo "Hallo Welt ";  
echo "<br />";  
echo "ich bin da";  
  
$variablenname = "Axel Pratzner";  
echo $variablenname;  
?>
```

Beachten Sie bei der Ausgabe das fehlende Leerzeichen nach dem "da" - besser machen!



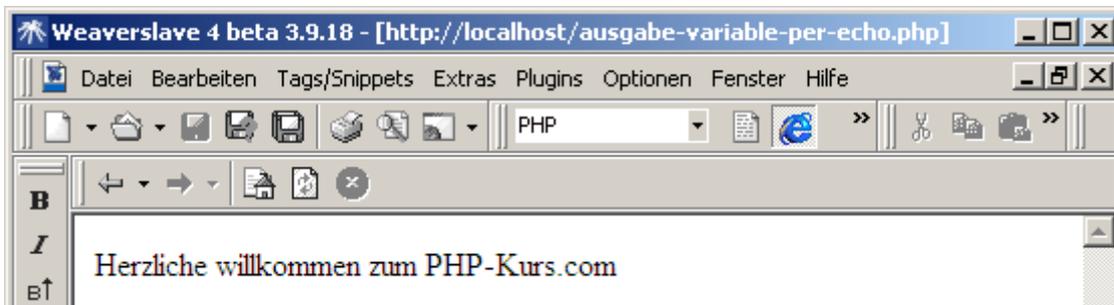
Im folgenden Beispiel wird in der Variablen `$kursname` der Kurs gespeichert und das ".com" als Variable `$suffix` gesetzt.

```
<?php  
  
$kursname = "PHP-Kurs";  
$suffix = ".com";
```

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

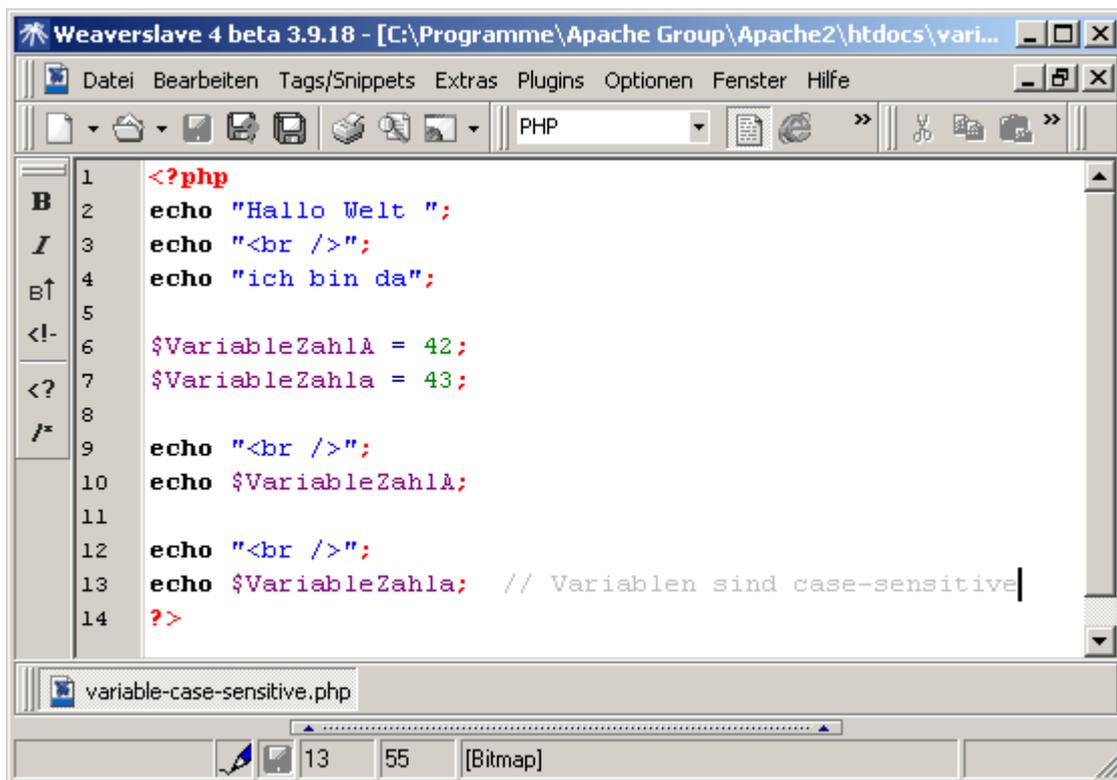
```
echo "Herzlich willkommen zum $kursname$suffix";  
?>
```

Als Ausgabe im Browser erscheint:



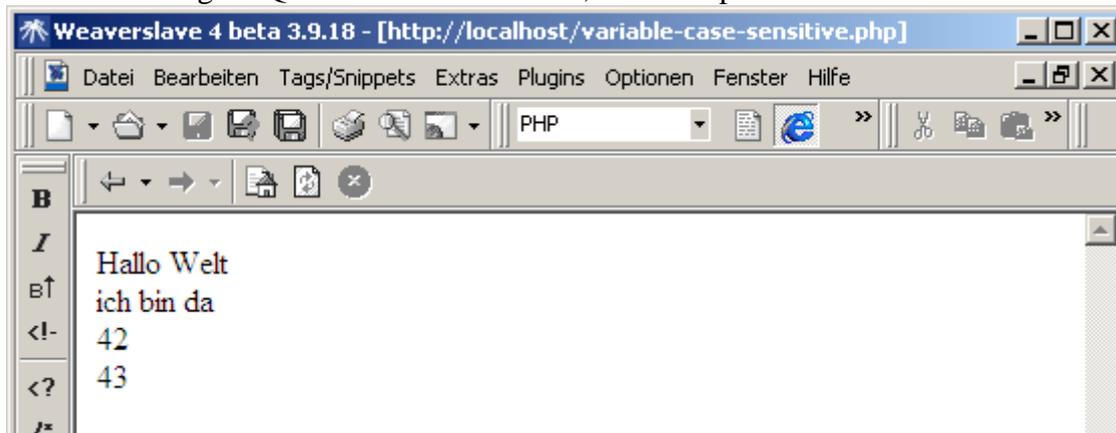
Variablen-Namen sind case-sensitiv

Case-Sensitiv bedeutet, dass eine unterschiedliche Schreibweise von Groß- und Kleinschreibung einen großen Unterschied macht - es entstehen dadurch verschiedene Variablen, was zu langem Suchen nach Fehlern beim Programmierer führt :).



Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

In der Ausgabe sehen Sie, dass wir hier mit 2 Variablen hantieren, die sich nur in Groß- und Kleinschreibung im Quellcode unterscheiden, aber komplett andere Werte beinhalten.



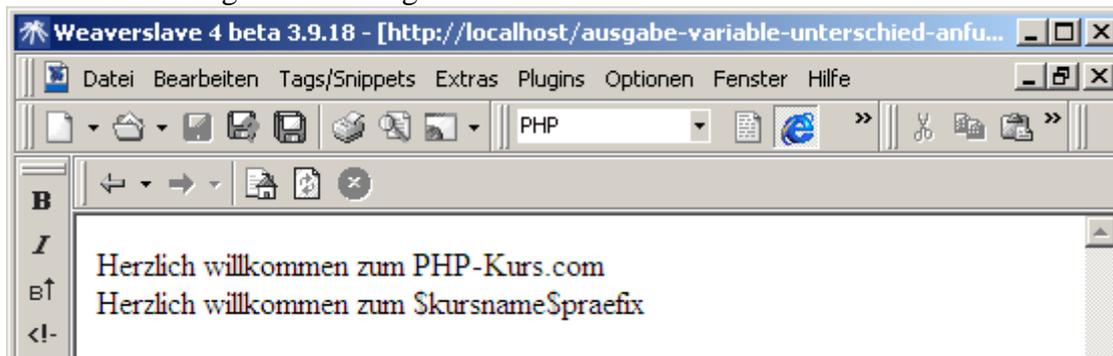
Einfache und doppelte Anführungszeichen bei Variablen

Der Inhalt zwischen den Doppelten Anführungszeichen wird von PHP ausgeführt (ordnungsgemäß spricht man von parsen). Variablen werden innerhalb von Anführungszeichen berücksichtigt und deren Inhalt ausgegeben.

Bei Verwendung von einfachen Anführungszeichen (die Zweitbelegung auf der #-Taste) wird im Browser exakt das ausgegeben, was zwischen den einfachen Anführungszeichen steht, ohne auf eventuell vorhandenen Variablen zu achten.

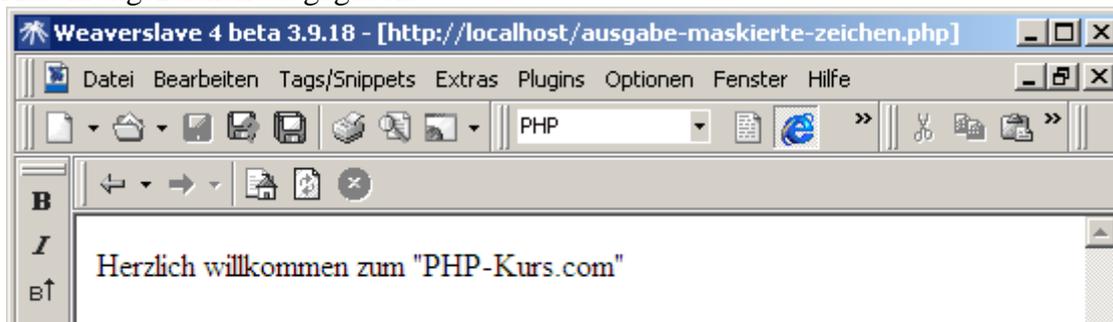


Und hier die Ausgabe zum Vergleich

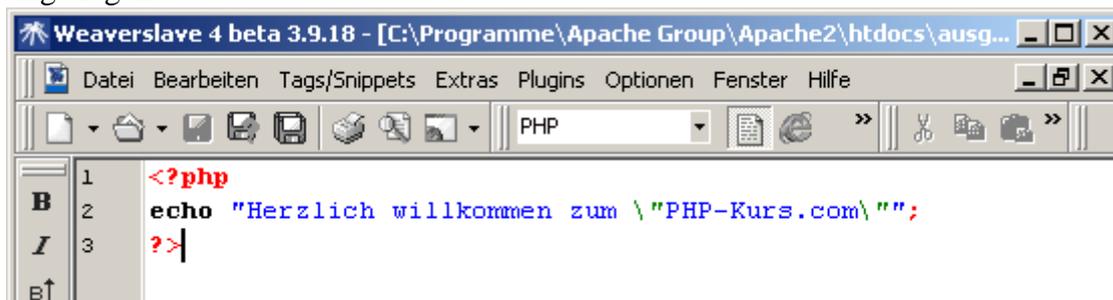


Maskieren von Sonderzeichen

Bestimmte Zeichen müssen maskiert werden, die sonst als Programmierzeichen dienen, aber auch einmal im Text erscheinen sollen - zum Beispiel soll ein Anführungszeichen innerhalb einer Ausgabe ausgegeben werden. Als Beispiel wird "PHP-Kurs.com" mit den Anführungszeichen ausgegeben.



Dazu wird vor dem jeweiligen Zeichen, das ausgegeben werden soll, eine Backslash (\) eingetragen.



Es gibt natürlich mehrere Wege nach Rom - hier sehen Sie 3 Möglichkeiten, die Anführungszeichen auszugeben:

- in der ersten Zeile wird das Anführungszeichen jeweils maskiert
- in der zweiten Zeile wird der entsprechende HTML-TAG für das Anführungszeichen verwendet
- in der dritten Zeile, indem die ganze Zeile durch einfache Anführungszeichen dem Parsen entzogen wird

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

```
1 <?php
2 echo "Herzlich willkommen zum \"PHP-Kurs.com\"";
3 echo "<br />";
4 echo "Herzlich willkommen zum &quot;PHP-Kurs.com&quot;";
5 echo "<br />";
6 echo 'Herzlich willkommen zum "PHP-Kurs.com"';
7 ?>
```

Alle 3 führen zum gleichen Ergebnis:

```
Herzlich willkommen zum "PHP-Kurs.com"
Herzlich willkommen zum "PHP-Kurs.com"
Herzlich willkommen zum "PHP-Kurs.com"
```

Übung: Bauen Sie folgende Ausgabe nach mit nur 1 x echo

Das Besondere an dieser Übung ist, dass nur einmal der echo-Befehl genutzt werden darf.

Vorgaben:

- nur 1 mal echo
- Text zentriert
- zwei Zeilen
- 2 Variablen, eine für Vornamen, eine für Nachnamen
- Ausgabe von Anführungszeichen beim Kurstitel

Viel Spaß beim Nachbauen

```
Herzlich willkommen Axel Pratzner
zum ["HTML-KURS.com"]
```

Zum Vergleich mit Ihrer Lösung finden Sie auf der folgenden Seite 2 verschiedene Lösungen. Ihre Lösung kann nochmals anders aussehen (richtig ist sie, solange die Ausgabe auf dem Browser identisch ist und nur 1 mal echo verwendet wurde).

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

```

1  <?php
2  $vorname = "Axel";
3  $nachname = "Pratzner";
4
5  echo "<p align=\"center\">Herzlich willkommen
6      $vorname $nachname
7      <br />zum [\"HTML-KURS.com\"]</p>";
8
9  // einfachere Variante
10 echo '<p align="center">Herzlich willkommen '.
11     $vorname ." ". $nachname .
12     '<br />zum ["HTML-KURS.com"]</p>';
13 ?>

```

Verkettungsoperator

Es könnte nun zur Ausgabe jeder Variable ein eigenes `echo` angegeben werden. Eine praktische Möglichkeit ist der Verkettungsoperator, der über den Punkt realisiert wird. Schauen Sie sich das folgende Beispiel an:

```

1  <?php
2  $kursname = "PHP-Kurs";
3  $praefix = ".com";
4
5  echo "Herzliche willkommen zum " . $kursname . $praefix;
6  ?>

```

Über den Verkettungsoperator (als als String-Operator) werden Strings verbunden bzw. zusammengefügt. Somit kann PHP-Code kürzer geschrieben werden.

Zwischen jedem verketteten Glied steht der Verkettungspunkt - vergleiche die 2 Punkte im obigen Beispiel.

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Vordefinierte Variablen

In PHP gibt es eine Reihe vordefinierter Variablen, die für den Entwickler interessant sind.

`$_SERVER['PHP_SELF']`

Dateinamen des ausgeführten Skripts (relativ zum Wurzelverzeichnis)

`$_SERVER['REMOTE_ADDR']`

IP-Adresse des Rechners, der die Seite angefordert hat

Beispiel: <http://www.ihre-ip-adresse.de>

`$_SERVER["HTTP_HOST"]`

Domainname (für diese Seite www.php-kurs.com)

Zur Fehlerbehandlung werden folgende vordefinierten Variablen benötigt (Beispiel dazu später)

`$_SERVER["REDIRECT_URL"]`

die ursprünglich aufgerufene URI

`$_SERVER["REDIRECT_STATUS"]`

ursprünglicher Statuscode, welcher dem Fehlercode bei nicht existierenden Seiten entspricht

Eine Abfrage der Variablen ist wie bei anderen Variablen möglich - einfach ein `echo` davor:

```
echo "PHP_SELF: ". $_SERVER['PHP_SELF'];  
echo "REMOTE_ADDR: ". $_SERVER['REMOTE_ADDR'];
```

phpinfo - Funktion zur Anzeige der Serverinformationen

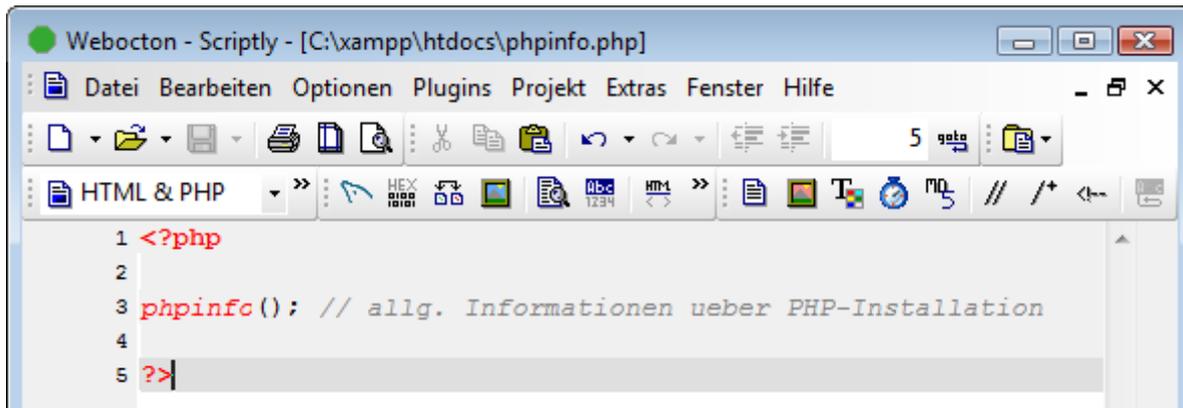
PHP-Interpreter Informationen entlocken

Als Befehl, der garantiert zu Verwirrung beim Anfänger führt, nutzen wir `phpinfo()`. Dieser gibt zahlreiche Informationen über die PHP-Installation und den Webserver aus (die wir alle erstmal getrost ignorieren - wichtig ist nur, dass so etwas in der Art erscheint). Somit ist sichergestellt, dass die Installation korrekt ist.

Nach dem PHP-Befehl kommt ein Semicolon `;` - dadurch weiß der PHP-Interpreter (das Teil, das das PHP umsetzt und als Programm lauffähig macht), dass dieser PHP-Befehl hier zu Ende ist. Das Semicolon wird gerne vergessen und führt dann zu langer Fehlersuche.

Und was immer oft zum Testen genutzt werden kann, ist das Auskommentieren (hier wird es als Hinzufügen von Kommentaren genutzt). Zum Kommentieren (wird dann vom PHP-Interpreter ignoriert) werden zwei Schrägstriche (Slash) `//` genutzt.

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen



```
1 <?php
2
3 phpinfo(); // allg. Informationen ueber PHP-Installation
4
5 ?>
```

Folgende (bzw. eine ähnliche) Ausgabe sollte nun beim Ausführen erscheinen.

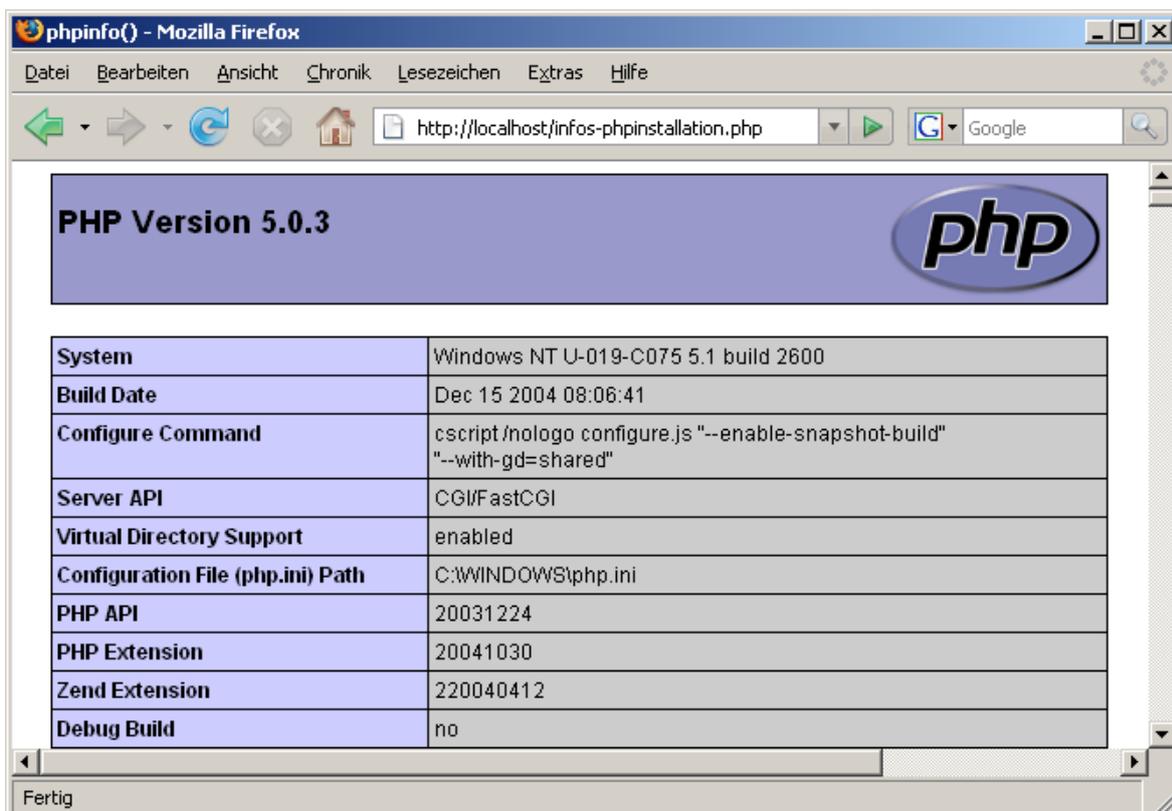
Zum Ausführen kann über den Internet-Browser gegangen werden - in der URL wird entweder

`http://localhost/gewahlter-dateiname.php`

bzw.

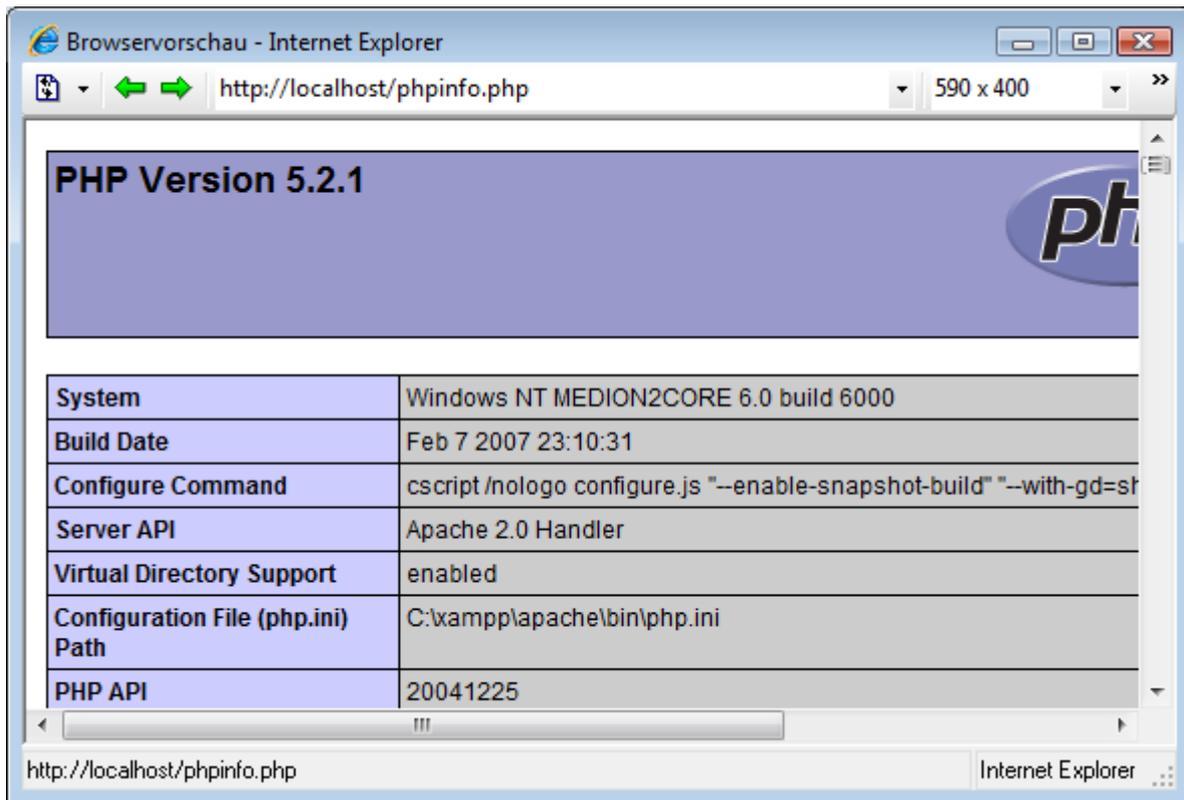
`http://127.0.0.1/gewahlter-dateiname.php`

Als Ausgabe erscheint nun:



Wenn Scriptly sauber eingerichtet ist, kann durch Anklicken des blauen E die Ausgabe des Internetbrowser gestartet werden (das E ist nur im Beginnermodus sichtbar) - ansonsten über F9.

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen



Berechnungen in PHP

Um mit PHP rechnen zu können, geben wir nach dem echo die gewünschte Rechenoperation ein. Ohne das echo würde zwar die Berechnung durchgeführt, wir würden aber das Ergebnis nicht auf dem Bildschirm zu sehen bekommen:

```
<?php
echo 5 + 3;
?>
```

Das funktioniert mit allen Grundrechenarten:

```
<?php
echo 5 - 3;
?>
```

Zum Multiplizieren wird das Malzeichen (Stern) benötigt - findet sich am schnellsten auf dem Zehnerblock.

```
<?php
echo 5 * 3;
?>
```

Zum Dividieren wird der Schrägstrich / benötigt - ist über der 7 zu finden.

```
<?php
echo 5 / 3;
?>
```

Als fertiges Ergebnis wird bei 5 durch 3 auch die Nachkommastellen angezeigt.

```
1.66666666666667
```

Was passiert, wenn durch 0 geteilt wird?

Berechnungen effektiv nutzen mit Variablen

Für Berechnungen benötigen wir das Wissen der vorherigen Kapitel - wir definieren Variablen, die wir addieren und über eine Verkettung ausgeben lassen.

Als Beispiel nehmen wir die Anzahl der Kursteilnehmer und Kursteilnehmerinnen. Jedes Geschlecht bekommt seine Variable - die Anzahl wird ohne Anführungsstriche der Variable zugewiesen.

```
$teilnehmer = 5;
$teilnehmerinnen = 4;

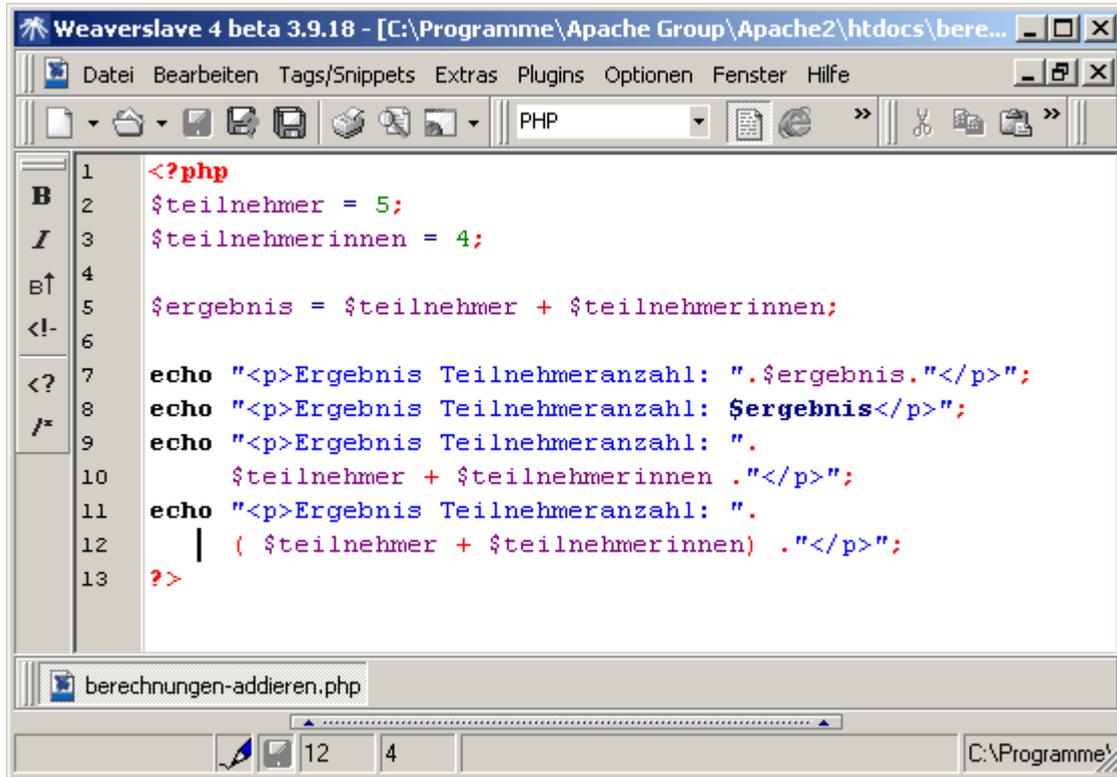
$ergebnis = $teilnehmer + $teilnehmerinnen;
echo $ergebnis;
```

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

verschiedene Ausgaben des Ergebnisses

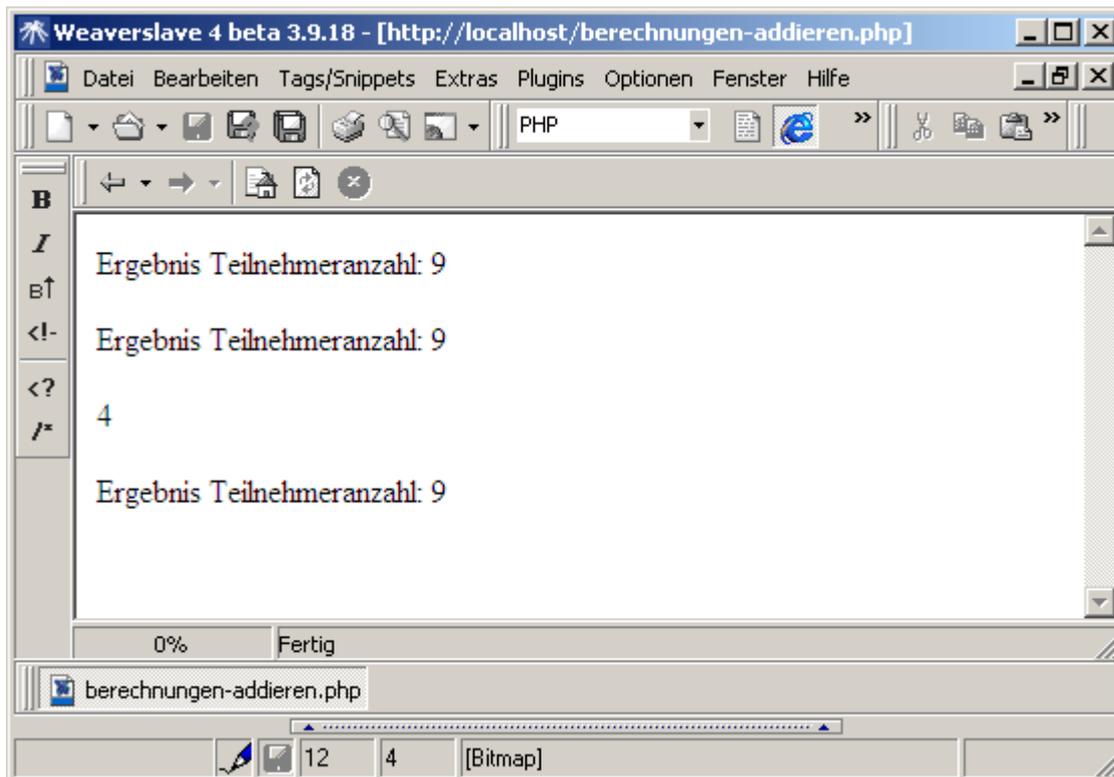
Für die ersten 2 Ausgaben wird vorneweg das Ergebnis errechnet.

Für die letzten beiden Ausgaben wird innerhalb der Ausgabe die Berechnung durchgeführt.



```
1 <?php
2 $teilnehmer = 5;
3 $teilnehmerinnen = 4;
4
5 $ergebnis = $teilnehmer + $teilnehmerinnen;
6
7 echo "<p>Ergebnis Teilnehmeranzahl: ".$ergebnis."</p>";
8 echo "<p>Ergebnis Teilnehmeranzahl: $ergebnis</p>";
9 echo "<p>Ergebnis Teilnehmeranzahl: ".
10     $teilnehmer + $teilnehmerinnen ."</p>";
11 echo "<p>Ergebnis Teilnehmeranzahl: ".
12     | ( $teilnehmer + $teilnehmerinnen) ."</p>";
13 ?>
```

Hier sehen Sie das Ergebnis der Browserausgabe - bitte schauen Sie nach dem Grund, warum die 3. Zeile nicht dem erwarteten Ergebnis entspricht - lernen durch Fehler ist sehr effektiv :)



Einfache Berechnungen

Im Folgenden eine Auflistung für einfache Berechnungen (+, -, *, /)

PHP-Schreibweise	Kurzschreibweise	Beschreibung
$\$a = \$a + \$b$	$\$a += \b	Die Variablen $\$a$ und $\$b$ werden addiert und wieder der Variablen $\$a$ zugewiesen
$\$a = \$a - \$b$	$\$a -= \b	Die Variablen $\$a$ und $\$b$ werden subtrahiert und wieder der Variablen $\$a$ zugewiesen
$\$a = \$a * \$b$	$\$a *= \b	Die Variablen $\$a$ und $\$b$ werden multipliziert und wieder der Variablen $\$a$ zugewiesen
$\$a = \$a / \$b$	$\$a /= \b	Die Variablen $\$a$ und $\$b$ werden dividiert und wieder der Variablen $\$a$ zugewiesen (bei $\$b=0$ erfolgt eine Fehlermeldung!)
$\$a = \$a + 1$	$\$a++$	Zu der Variable $\$a$ wird 1 addiert und wieder der Variablen $\$a$ zugewiesen
$\$a = \$a - 1$	$\$a--$	Zu der Variable $\$a$ wird 1 subtrahiert und wieder der Variablen $\$a$ zugewiesen

In PHP gibt es verschiedene Funktionen für das Berechnen. Vorteil zu der einfachen Schreibweise ist, dass auch eine Genauigkeit (Anzahl Nachkommastellen) mitgegeben werden kann. Siehe hierzu unter PHP.net: bcadd, bccomp, bcddiv, bccmod, bcmul, bcpow, bcpowmod, bcscale, bcsqrt, bcsusb

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Aufgabe: Berechnen Sie die Annuität (wird für Baufinanzierung benötigt)

Über die Annuität wird aufgenommenes Geld zurückbezahlt. Dabei setzt sich die Annuität aus den Elementen Zins und Tilgung zusammen. Die Höhe der Rückzahlung (Annuität) bleibt immer gleich - nur die Anteile des bezahlten Zins der Tilgung verändern sich.

Die Höhe der Annuität berechnet sich über die Formel:

$$\text{Annuität} = \text{Kreditsumme} * \text{Zinssatz} * (1 + \text{Zinssatz})^{\text{Laufzeit}} / ((1 + \text{Zinssatz})^{\text{Laufzeit}} - 1)$$

Für einen aktuellen Tageszinssatz können Sie auf den entsprechenden Website von Banken nachsehen - nehmen wir einfach fiktive 5% an.

Die PHP benötigten Funktionen sind:

bcpow = Potenz einer Zahl

Mathematische Schreibweise: 4.5^3 (sprich 4.5 hoch 3)

bcpow (Wert, Potenz, Genauigkeit nach Komma)

Beispiel: `echo bcpow('4.5', '3', 2); // Ergebnis = 91.13`

Folgende Funktionen können Sie auch einsetzen (anstatt von + - * /)

bcadd = Addition zweier Zahlen

bcsb = Subtraktion zweier Zahlen

bcmul = Multiplikation zweier Zahlen

bcdiv = Division zweier Zahlen

Aufgabe: Erstellen Sie ein PHP-Programm, das mit 5% Zins und einer Laufzeit von 10 Jahren für eine Kreditsumme von 200.000 die Annuität berechnet. (Ergebnis: 24.658,337) Probieren Sie das auch mit dem aktuellen Zinssatz.

Funktionen von PHP

PHP bietet bereits sehr viele fertige Funktionen, die bestimmte Aufgaben erfüllen. Dadurch kann man sich sehr viel Programmierarbeit ersparen. Wenn wir nun für unsere Addition zusätzlich das Ergebnis auf 2 Nachkommastellen runden wollen, gibt es dafür z.B. eine Funktion.

Aufbau des Funktionsaufrufs

Der Aufbau des Funktionsaufrufs ist immer identisch:

```
Rückgabewert = Funktionsname (Übergabewert1, [...], Übergabewert n);
```

Die Funktionen erwarten verschiedene Übergabewerte. Für eine Addition müssen mindestens 2 Werte übergeben werden – die zwei Zahlen die miteinander addiert werden sollen. Da man

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

als PHP Einsteiger weder alle Funktionen kennen kann noch alle Übergabewerte hilft hier die Standardreferenz <http://de.php.net>

Über die Suche auf der Seite suchen wir nach der Funktion `bcadd`.

Die Seite <http://de.php.net/manual/de/function.bcadd.php> beschreibt die Funktion und die möglichen Übergabewerte im Bereich **Beschreibung**. Dabei sind in eckigen Klammern optionale Werte, d.h. Werte die angegeben werden können aber nicht unbedingt angegeben werden müssen.

In der **Parameter-Liste** sehen wir, was für eine Form die Parameter (Übergabewerte) haben sollen.

Im Bereich **Rückgabewerte** ist zu sehen (wenn etwas zurückgeliefert wird), was und in welcher Form es zurück kommt.

Bei unserem Beispiel mit `bcadd` müssen wir also die 2 zu addierende Zahlen mitgeben

```
bcadd("5" , "3");
```

Zusätzlich (muss nicht, daher in der Beschreibung die eckigen Klammern) die Möglichkeit die gewünschten Nachkommastellen mitzugeben

```
bcadd("5" , "3", "2");
```

Und wir erhalten eine Rückgabewert, den wir sofort ausgeben können oder in einer Variable speichern können

```
echo bcadd("5" , "3", "2");  
$ergebnis = bcadd("5" , "3", "2");
```

Unter dem Bereich finden wir in `php.net` noch Beispiele und ähnliche Funktionen.

Interessante ist der folgende Bereich von den "User Contributed Notes", also von **Nutzer beigetragene Anmerkungen**. Hier werden Probleme erörtert und Problemlösungen aufgezeigt.

Ähnliche Funktionen

Auf der linken Seite in der Steuerung sehen wir, dass unsere Funktion `bcadd` (fett dargestellt) eine Funktion aus dem Bereich „BC Math Funktionen“ ist und **weitere Funktionen aus diesem Bereich** werden dort gezeigt.

Aufgabe: Nutzen Sie die Funktion zum Teilen und stellen Sie das Ergebnis auf 3 Nachkommastellen dar

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Datumsausgabe mit date()

Das Datum kann in PHP in verschiedenen Formaten ausgegeben werden. Dazu wird die Funktion `date` genutzt.

Nach der echo-Ausgabe wird über den Punkt die Funktion `datum` verkettet.

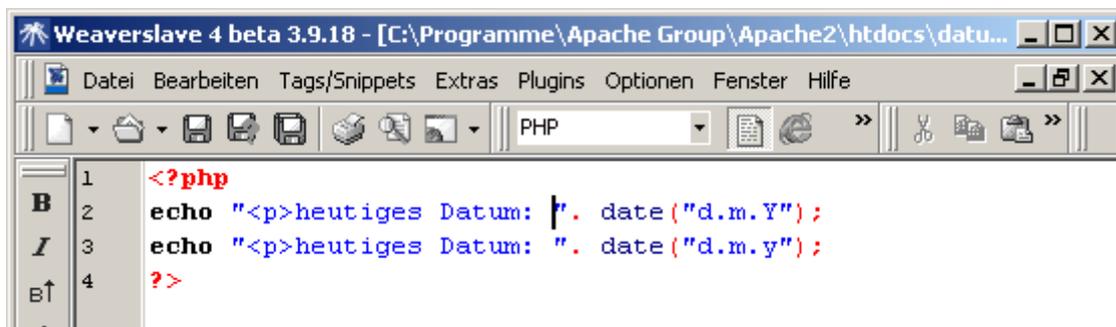
Die Attribute bedeuten:

d = Tag zweistellig

m = Monat als Zahl zweistellig

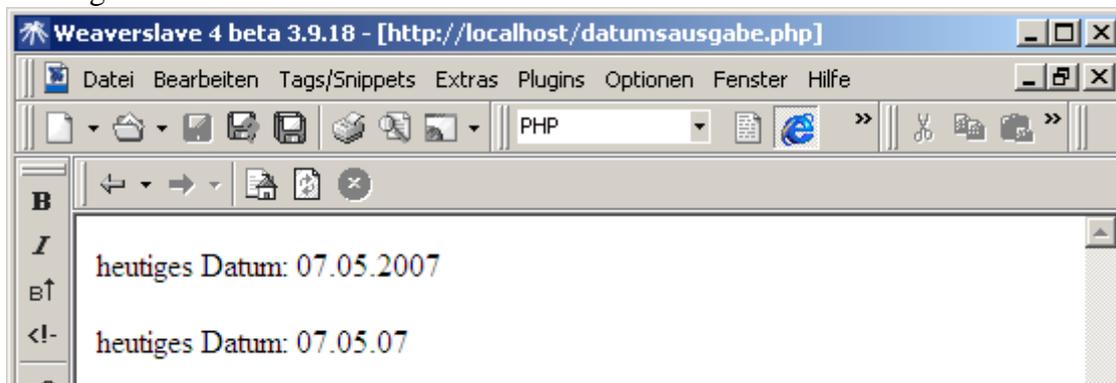
y = Jahr zweistellig

Y = Jahr vierstellig



```
1 <?php
2 echo "<p>heutiges Datum: ". date("d.m.Y");
3 echo "<p>heutiges Datum: ". date("d.m.y");
4 ?>
```

Das Ergebnis im Browser:



Hilfe zu Funktionen

Da PHP eine Vielzahl von Funktionen hat und diese zahlreiche Attribute anbieten, nutzen viele Programmierer die Hilfestellung über php.net

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

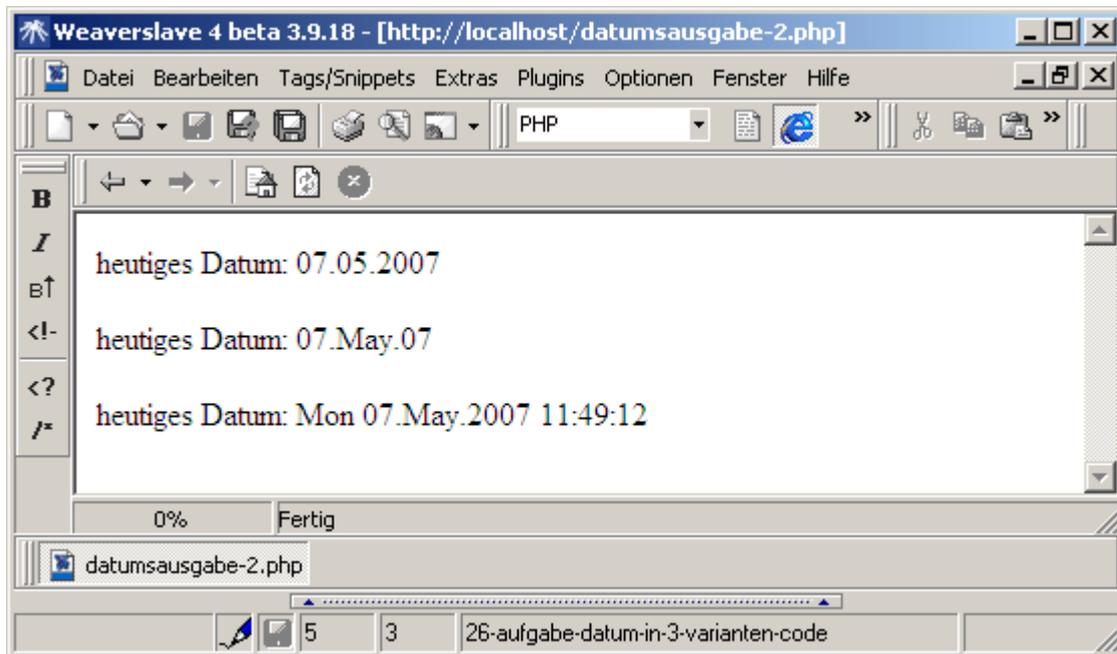


Nachdem die Funktion eingegeben wurde, erhält man den Aufbau und die möglichen Attribute.



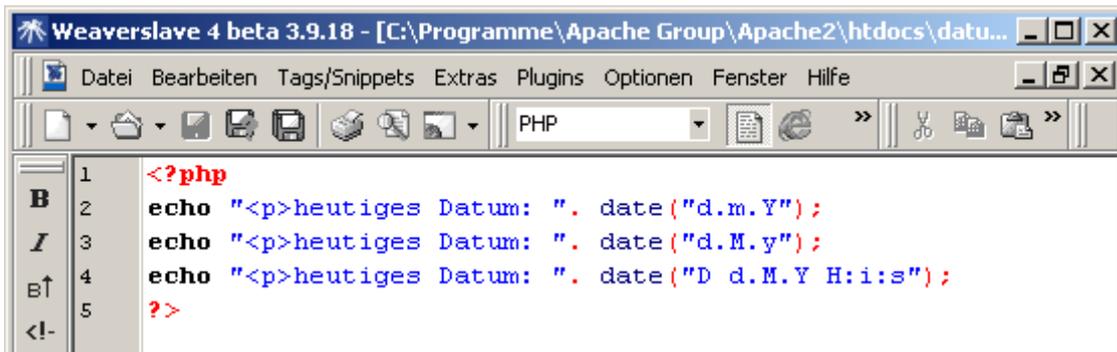
Aufgabe: Datum anzeigen auf verschiedene Arten

Stellen Sie das Datum auf verschiedene Arten dar - auch einmal mit Uhrzeit. Für die Informationen, wie es geht, nutzen Sie bitte php.net.



Die Lösung zum Vergleichen mit Ihrer Lösung gibt es auf der folgenden Seite.

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen



The image shows a screenshot of the Weaverslave 4 beta 3.9.18 web editor. The window title is "Weaverslave 4 beta 3.9.18 - [C:\Programme\Apache Group\Apache2\htdocs\datu...". The menu bar includes "Datei", "Bearbeiten", "Tags/Snippets", "Extras", "Plugins", "Optionen", "Fenster", and "Hilfe". The toolbar contains various icons for file operations and editing. The main editor area shows the following PHP code:

```
1 <?php
2 echo "<p>heutiges Datum: ". date("d.m.Y");
3 echo "<p>heutiges Datum: ". date("d.M.y");
4 echo "<p>heutiges Datum: ". date("D d.M.Y H:i:s");
5 ?>
```

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Bedingungen if / else

Über den PHP-Behehl `if` können Abfragen erstellt werden.

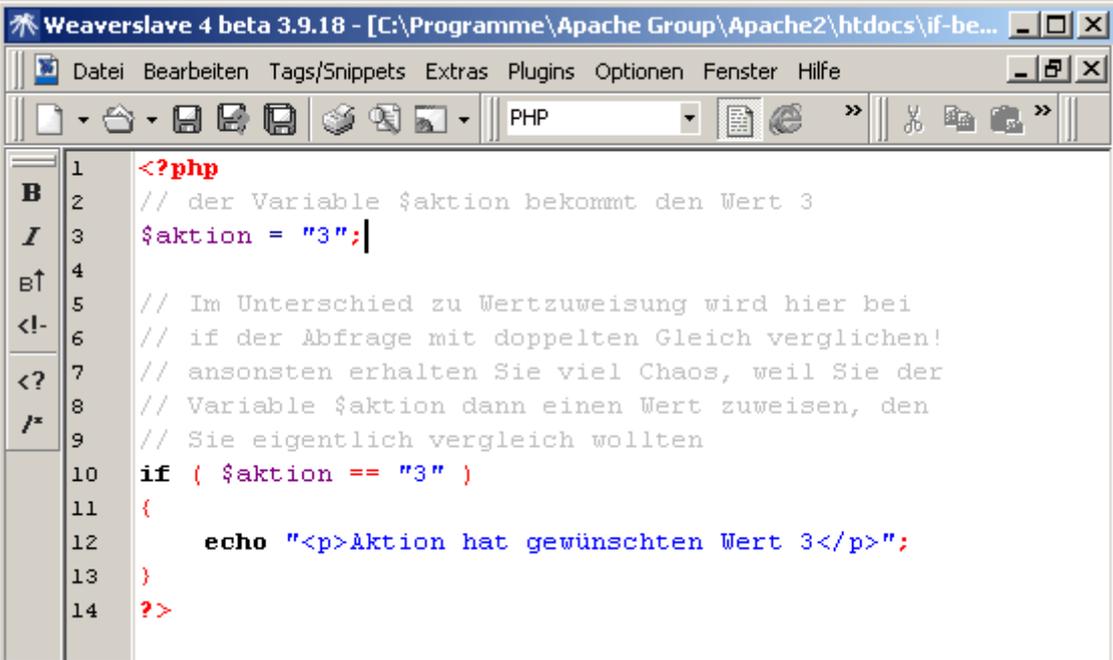
Aufbau:

```
if () { }
```

```
wenn (BEDINGUNG ZUTRIFFT) DANN { }
```

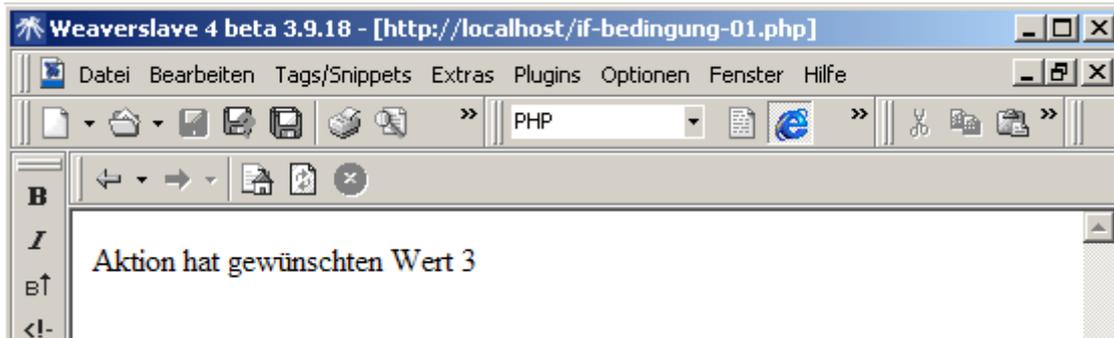
Es gibt verschiedene Möglichkeiten zum Vergleichen:

Operator	Bedeutung
<code>==</code>	gleich
<code>===</code>	gleich (auch vom Variablentyp - String, Boolean, Integer)
<code>!=</code>	ungleich
<code><></code>	ungleich, ist identisch mit <code>!=</code>
<code><</code>	kleiner
<code><=</code>	kleiner oder gleich
<code>></code>	größer
<code>>=</code>	größer oder gleich



```
1 <?php
2 // der Variable $aktion bekommt den Wert 3
3 $aktion = "3";
4
5 // Im Unterschied zu Wertzuweisung wird hier bei
6 // if der Abfrage mit doppelten Gleich verglichen!
7 // ansonsten erhalten Sie viel Chaos, weil Sie der
8 // Variable $aktion dann einen Wert zuweisen, den
9 // Sie eigentlich vergleichen wollten
10 if ( $aktion == "3" )
11 {
12     echo "<p>Aktion hat gewünschten Wert 3</p>";
13 }
14 ?>
```

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen



IF-ELSE Bedingungen mit Alternative

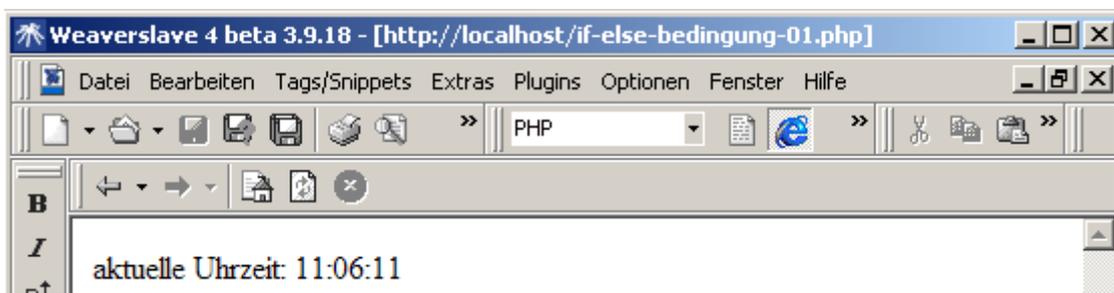
Trifft eine Bedingung nicht zu, wird der Programm-Teil `else` abgearbeitet.

Im Beispiel wird die aktuelle Uhrzeit verglichen, ob es 12:00:00 ist, und dann entsprechender Text ausgegeben.

```

1  <?php
2  // der Variable $aktion bekommt den Wert 3
3  $aktUhrzeit = date("H:i:s");
4
5  // Im Unterschied zu Wertzuweisung wird hier bei
6  // if der Abfrage mit doppelten Gleich verglichen!
7  // ansonsten erhalten Sie viel Chaos, weil Sie der
8  // Variable $aktion dann einen Wert zuweisen, den
9  // Sie eigentlich vergleich wollten
10 if ( $aktUhrzeit == "12:00:00" )
11 {
12     echo "<p>high noon</p>";
13 }
14 else
15 {
16     echo "<p>aktuelle Uhrzeit: $aktUhrzeit</p>";
17 }
18 ?>

```



Stellen Sie Ihre System-Uhr auf kurz vor 12 und schauen Sie im Sekundentakt nach :).

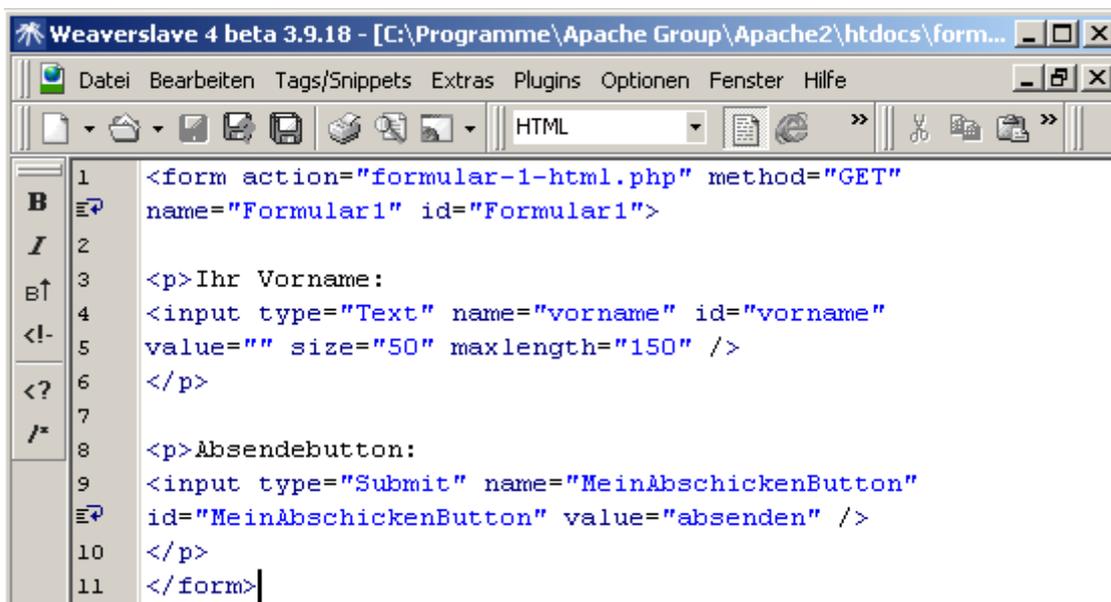
Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Formulare - Affenformular, Auswertung und Feedback

Die HTML-Tags für einfaches Formular finden Sie unter <http://www.html-seminar.de/formulare.htm>

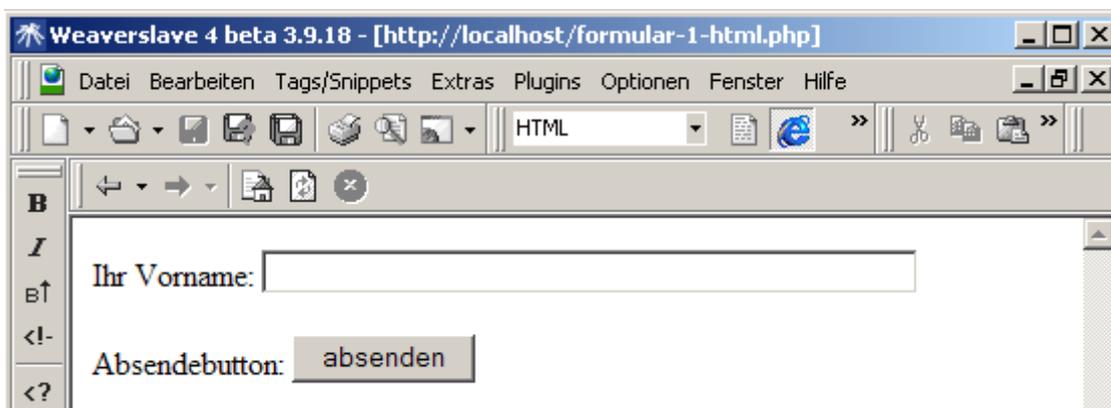
Der Aufbau vom HTML-Tag form ist einfach:
über das Attribut `action=""` wird angegeben, welche Internetseite aufgerufen wird, wenn das Formular abgesendet wird.

Zusätzlich wird über das Attribut `method="get|post"` die Art des Versendens vorgegeben - dazu weiter unten mehr.



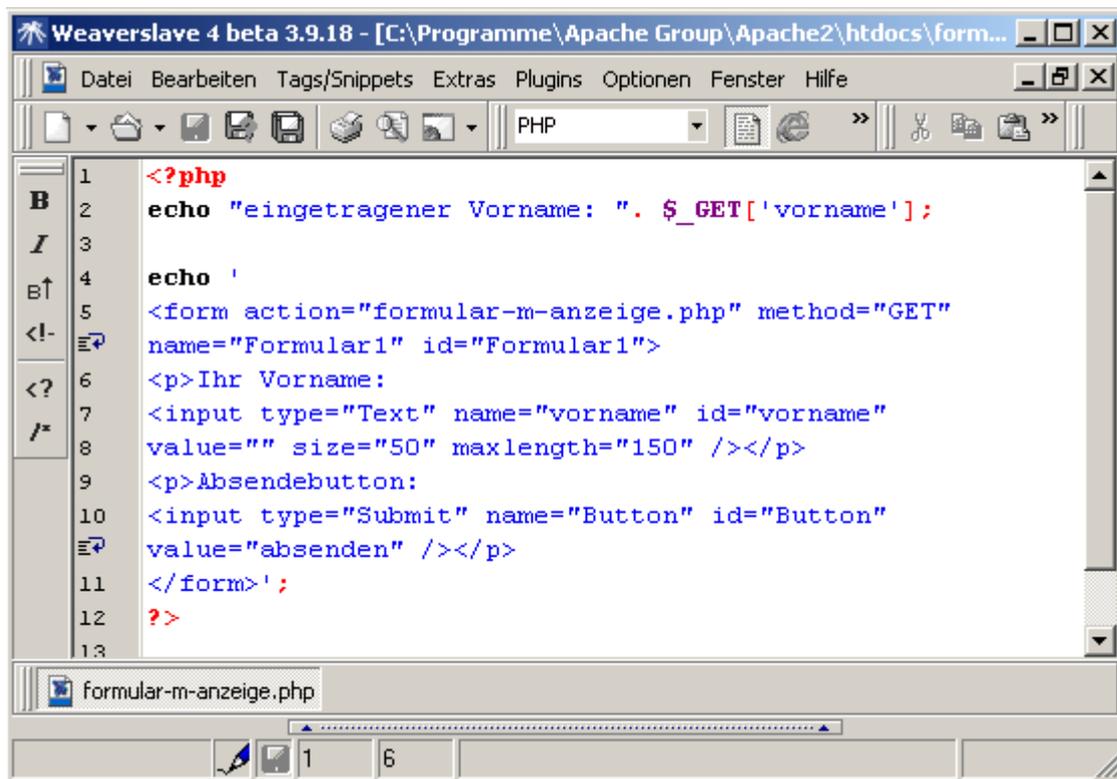
```
1 <form action="formular-1-html.php" method="GET"
  name="Formular1" id="Formular1">
2
3 <p>Ihr Vorname:
4 <input type="Text" name="vorname" id="vorname"
5 value="" size="50" maxlength="150" />
6 </p>
7
8 <p>Absendebutton:
9 <input type="Submit" name="MeinAbschickenButton"
10 id="MeinAbschickenButton" value="absenden" />
11 </form>
```

Die Ausgabe im Browser ist sehr einfach und bisher passiert mit eingegebenen Daten nichts. Nach Absenden wird dieselbe PHP-Datei aufgerufen.



Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Jetzt wird das Script ergänzt um einen Teil, der mit den übergebenen Daten etwas macht - hier werden die abgesendeten Informationen einfach angezeigt.



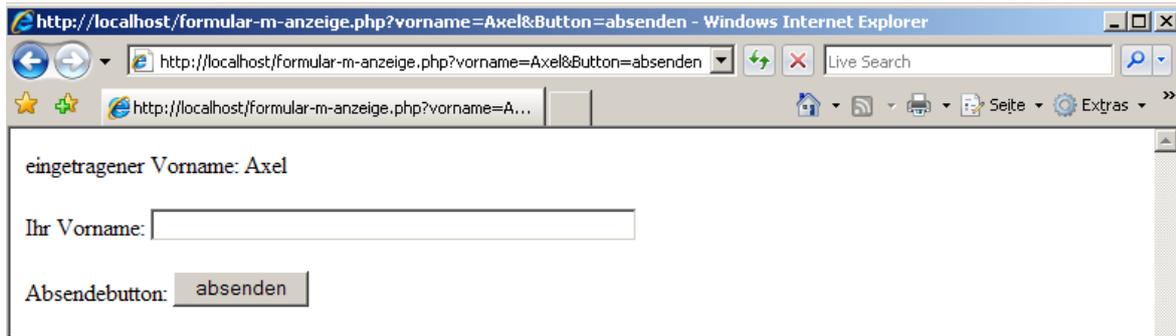
```
1 <?php
2 echo "eingetragener Vorname: ". $_GET['vorname'];
3
4 echo '
5 <form action="formular-m-anzeige.php" method="GET"
6 name="Formular1" id="Formular1">
7 <p>Ihr Vorname:
8 <input type="Text" name="vorname" id="vorname"
9 value="" size="50" maxlength="150" /></p>
10 <p>Absendebutton:
11 <input type="Submit" name="Button" id="Button"
12 value="absenden" /></p>
13 </form>' ;
14 ?>
```

Die Anzeige sollte im Browser erfolgen - kopieren Sie den Dateinamen einfach in den Browser mit anfangs localhost



Wenn nun in das Feld Vorname ein Eintrag erfolgt und der Button absenden angeklickt wird, wird das PHP-Programm ausgeführt.

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen



Im Browser wird nun der eingetragene Vorname in der ersten Zeile angezeigt und wieder ein leeres Formular angezeigt.

Interessant ist die URL-Zeile des Browsers. Dort sehen Sie nach dem Absenden, dass die ursprüngliche URL ergänzt wurde:

```
http://localhost/formular-m-anzeige.php?vorname=Axel&Button=absenden
```

Es werden nun die eingetragenen Informationen wie Vorname und der Wert des Buttons übertragen.

Innerhalb des PHP-Programmes kann nun auf die Informationen zugegriffen werden. Es ist eine Variable, die zwecks Sicherheit über die Konstruktion `$_GET['variablenname']` abgefragt wird.

Beobachten Sie die Zeile oben. Geben Sie Ihren Namen ein und klicken Sie auf den Button - wie sieht die Zeile im Vergleich aus, wenn nach dem Eintragen des Namens nur die Returntaste gedrückt wird?

Unterschied GET und POST

Bisher haben wir mit GET die Daten übertragen. Bei GET werden die Daten über die URL-Zeile übertragen. Diese sind für jeden sichtbar und sind im Umfang begrenzt (je nach Einstellung des Webservers - i.d.R. 4kb)

Als Alternative wurde POST eingesetzt. Die Daten werden ebenfalls zum Webserver übertragen, allerdings für den normalen Nutzer nicht mehr sichtbar. Die Abfrage der Inhalte erfolgt dann über `$_POST['Variablenname']`

Im PHP-Kurs wird zwecks Nachvollziehbarkeit nur die GET-Methode verwendet.

Tipp: Eine Abfrage, die sowohl über die GET- wie POST-Methode beinhaltet, ist `$_REQUEST['Variablenname']` - diese ist allerdings später beim ausgewachsenen Programmieren nicht wirklich erwünscht, da dadurch Sicherheitskonzepte aufgeweicht werden.

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

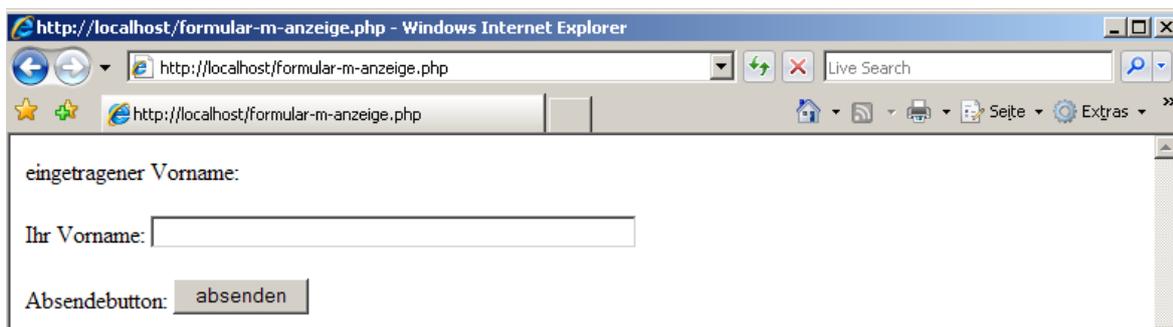


AUFGABE: Erstellen eines Formulars mit hidden-Feld

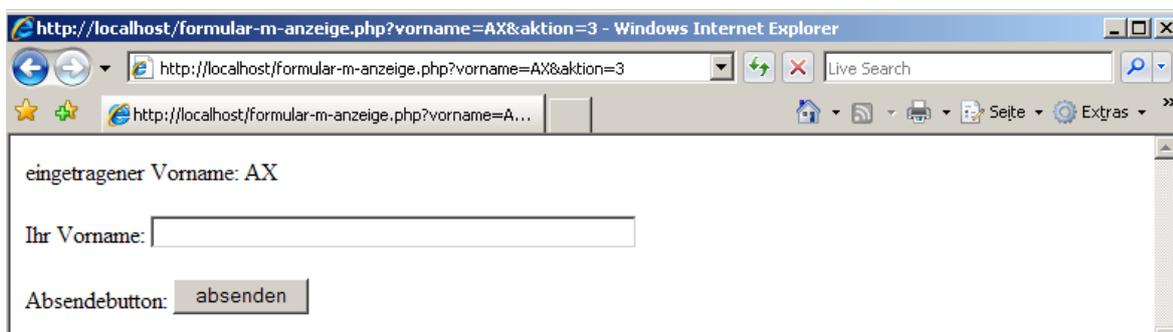
Über das HTML-Element `input type="hidden"` können für den Benutzer nicht sichtbare Informationen übertragen werden `<input type="hidden" name="werteins" id="werteins" value="x" />`

Aufgabe ist, über ein Formular mit hidden-Feld einen Wert im Hintergrund zu übertragen. Der Wert des hidden-Feldes soll über eine Variable mit dem Namen "\$aktion" übertragen werden. In dieser Variable soll als Wert 3 übertragen werden.

\$aktion wird übergeben mit hidden - Kontrolle, ob eine 3 übergeben wird (über Variable)



Beachten Sie die Browserzeile



Lösungsvorschlag auf der folgenden Seite:

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

```
1 <?php
2 echo "eingetragener Vorname: ". $_GET['vorname'];
3 $aktion = date("H:i:s");
4 echo '
5 <form action="formular-m-anzeige.php" method="GET"
6 name="Formular1" id="Formular1">
7 <p>Ihr Vorname:
8 <input type="Text" name="vorname" id="vorname"
9 value="" size="50" maxlength="150" /></p>
10
11 <input type="hidden" name="aktion" id="aktion"
12 value="'. $aktion .'
```

Formularwerte über IF kontrollieren

Ziel ist, dass nach Ausfüllen eines Formulars eine Aktion stattfindet, wenn der Vorname eingetragen wurde.

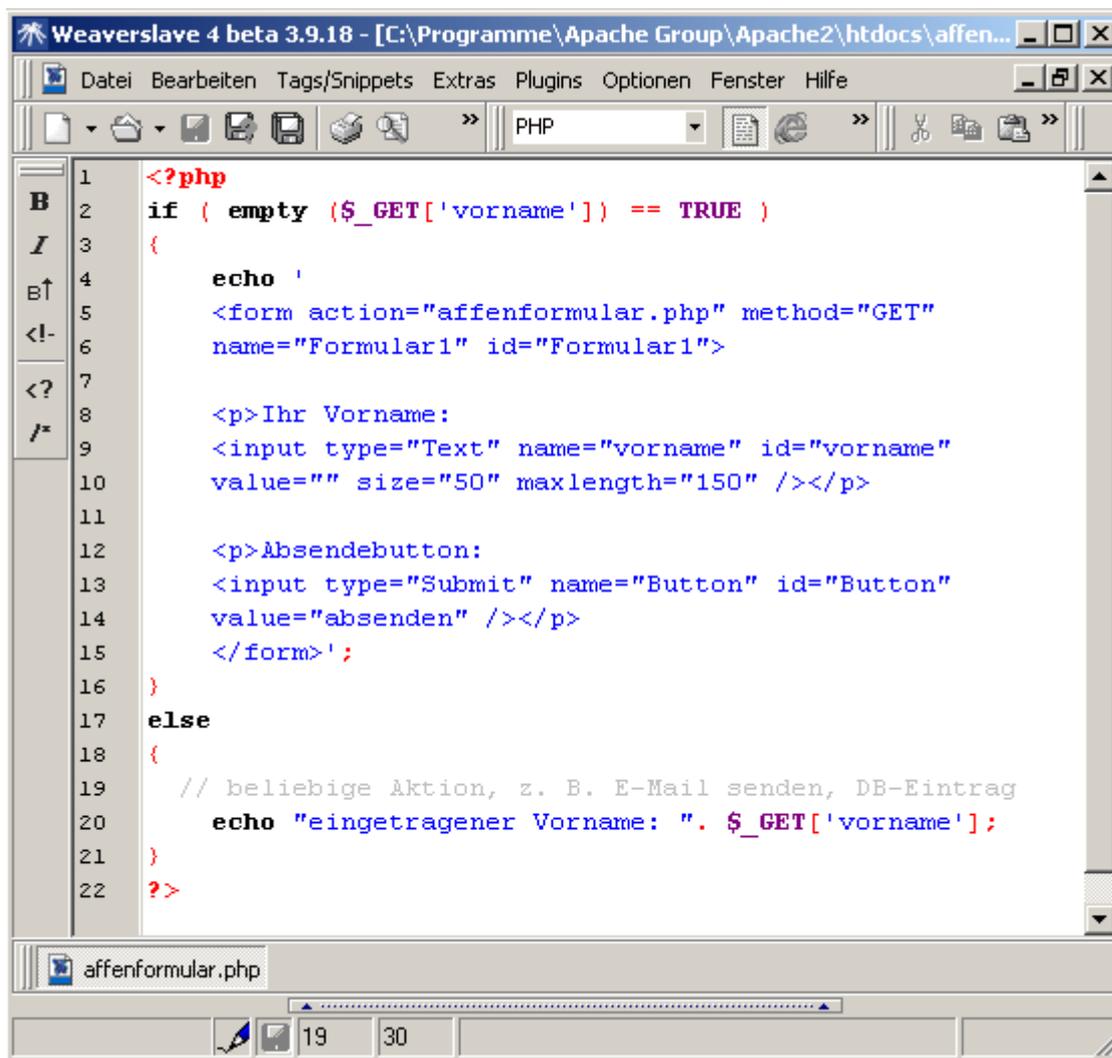
Der Ablauf ist:

1. Beim ersten Aufruf ist `$_GET['vorname']` noch leer (Feld vorname nicht eingetragen UND abgesendet).
2. Nach dem Absenden wird wieder dasselbe PHP-Script aufgerufen, diesmal in der URL mit den eingetragenen Feldern
`http://localhost/affenformular.php?vorname=Ax&Button=absenden`
3. Das Script überprüft, ob die Variable `vorname` ausgefüllt wurde. Früher ging es, einfach `$vorname` zu kontrollieren. Heute ist es durch Sicherheitsbedingungen notwendig, über `$_GET` die entsprechende Variable zu kontrollieren, also `$_GET['vorname']` (technisch gesehen ist das `$_GET` ein ARRAY, dazu in späteren Kapiteln mehr).
 - o Fall 1: Ist die Variable leer, wird einfach wieder das Formular angezeigt.
 - o Fall 2: Ist die Variable gefüllt, wird der Teil nach dem `else` ausgeführt.

Dazu wird über `if` kontrolliert, ob die Variable "vorname" leer ist. Für die Kontrolle wird die Funktion `empty` verwendet. Diese Funktion testet, ob eine Variable leer ist oder einen Inhalt erhält. Je nachdem gibt es als Rückwert richtig oder falsch - in Programmiersprachen kommt dann `TRUE` oder `FALSE` (dabei steht `TRUE` für 1 und `FALSE` für 0).

Für `empty` sind dabei verschiedene Schreibweisen zulässig, einfach nach Vorliebe des Programmiers. Zu sehen sind folgende Varianten, die alle gleich funktionieren:

- `if (empty ($_GET['vorname']) == TRUE)`
- `if (empty ($_GET['vorname']) == 1)`
- `if (empty ($_GET['vorname']))`



```
1 <?php
2 if ( empty ($_GET['vorname']) == TRUE )
3 {
4     echo '
5     <form action="affenformular.php" method="GET"
6     name="Formular1" id="Formular1">
7
8     <p>Ihr Vorname:
9     <input type="Text" name="vorname" id="vorname"
10    value="" size="50" maxlength="150" /></p>
11
12    <p>Absendebuttton:
13    <input type="Submit" name="Button" id="Button"
14    value="absenden" /></p>
15    </form>';
16 }
17 else
18 {
19     // beliebige Aktion, z. B. E-Mail senden, DB-Eintrag
20     echo "eingetragener Vorname: ". $_GET['vorname'];
21 }
22 ?>
```

Als Affenformular werden Formulare bezeichnet, die sich selber aufrufen und die die gemachten Eingaben auf ihre Gültigkeit überprüfen. Erst wenn die gewünschten Eingaben vorhanden sind (z. B. korrekte E-Mail-Adresse, PLZ, etc.) findet eine Weiterverarbeitung statt.

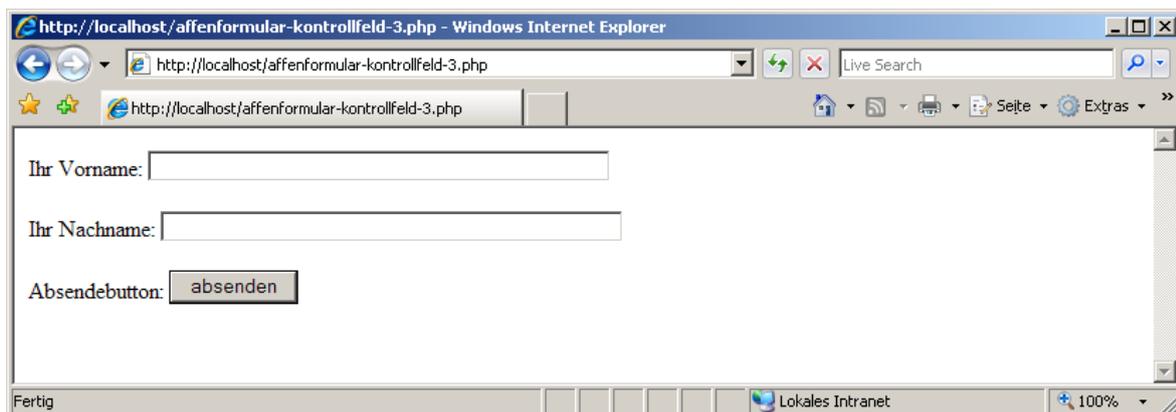
Der Name Affenformular kommt daher, dass selbst bei tausendfachem Aufruf durch tausend Affen nichts bewirkt wird.

AUFGABE: Erstellen Sie ein Affenformular mit dem hidden-Feld mit Namen "aktion" und dem Wert "1"

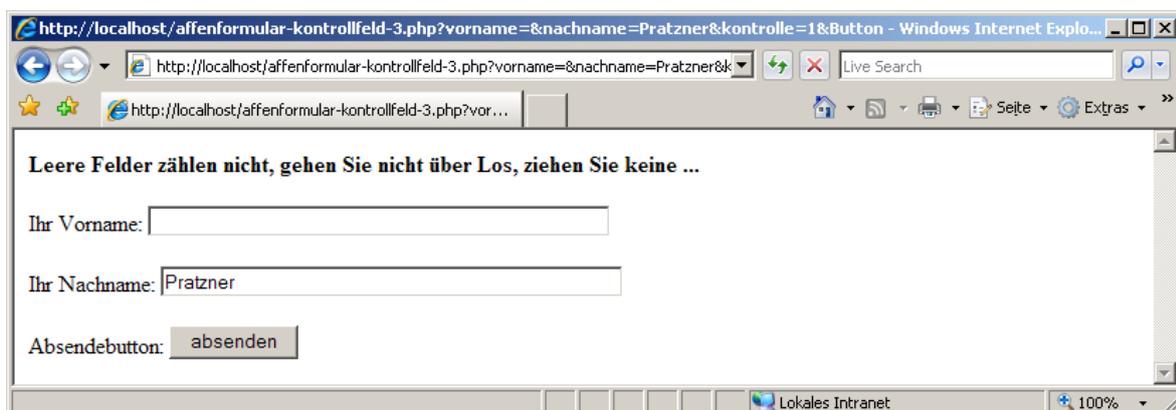
- Affenformular erstellen
- hidden-Feld mit dem Name "kontrolle" und dem Wert "1"
- Bauen Sie im ersten Bereich des IF-Bereichs folgende Zeilen an:
- `if ($_GET['kontrolle'] == 1) { echo "<p>Leere Felder zählen nicht, gehen Sie nicht über Los, ziehen Sie keine ...</p>"; }`

PHP-Programm erstellen und erklären Sie, für was das if innerhalb des ersten if-Bereichs gut ist.

Beim ersten Start wird ein leeres Formular ausgegeben:

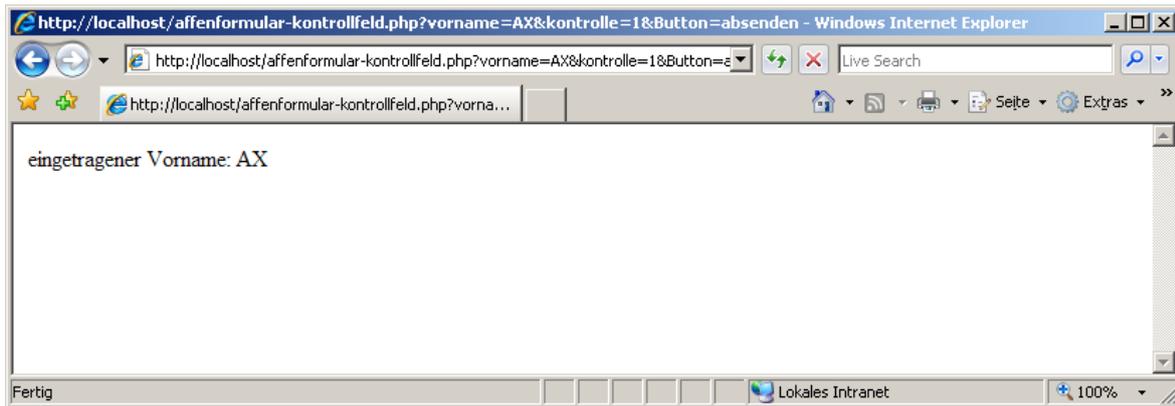


Wird keine Eingabe bei dem Feld Vorname gemacht, erscheint ein Fehlerhinweis.



Wird eine Eingabe gemacht, wird der zweite Teil der IF-Anweisung ausgeführt.

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen



Zur Kontrolle Ihres Ergebnisses können Sie auf der folgenden Seite den Lösungsweg ansehen.

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

```
Weaverslave 4 beta 3.9.18 - [C:\Programme\Apache Group\Apache2\htdocs\affen...
Datei Bearbeiten Tags/Snippets Extras Plugins Optionen Fenster Hilfe
PHP
1 <?php
2 if ( empty ( $_GET['vorname'] ) == TRUE )
3 (
4     if ( $_GET['kontrolle'] == 1 )
5     (
6         echo "<p><b>Leere Felder zählen nicht, ";
7         echo "gehen Sie nicht über Los, ziehen Sie ";
8         echo "keine ...</b></p>";
9     )
10    echo '
11    <form action="affenformular-kontrollfeld.php"
12    method="GET" name="Formular1" id="Formular1">
13
14    <p>Ihr Vorname:
15    <input type="Text" name="vorname" id="vorname"
16    value="" size="50" maxlength="150" /></p>
17
18    <input type="hidden" name="kontrolle" id="kontrolle"
19    value="1" />
20
21    <p>Absendebutton:
22    <input type="Submit" name="Button" id="Button"
23    value="absenden" /></p>
24    </form>';
25 )
26 else
27 (
28     // beliebige Aktion, z. B. E-Mail senden, DB-Eintrag
29     echo "eingetragener Vorname: ". $_GET['vorname'];
30 )
31 ?>
```

Aufgabe: Erstellen Sie das Formular mit zusätzlichen Feld Nachnamen. Wichtig ist, dass beide Felder ausgefüllt werden

TIPP: bei `if` können auch mehrere Bedingungen eingegeben werden und logisch mit `and` oder `or` verbunden werden, z.B.

```
if ( empty ($_GET['vorname']) == TRUE and empty ($_GET['nachname']) == TRUE )  
if ( empty ($_GET['vorname']) == TRUE or empty ($_GET['nachname']) == TRUE )
```

The screenshot shows a Windows Internet Explorer browser window with the URL `http://localhost/affenformular-kontrollfeld-2.php`. The page content includes two text input fields: "Ihr Vorname:" and "Ihr Nachname:". Below these fields is a button labeled "absenden". The browser's status bar at the bottom indicates "Fertig" and "Lokales Intranet".

This screenshot shows the same browser window as the previous one, but the "Ihr Vorname:" field now contains the text "AX". The "Ihr Nachname:" field remains empty, and the "absenden" button is still present.

The screenshot shows the browser window with a URL that includes query parameters: `http://localhost/affenformular-kontrollfeld-2.php?vorname=AX&nachname=&kontrolle=1&Button=absen`. The page displays a message: "Leere Felder zählen nicht, gehen Sie nicht über Los, ziehen Sie keine ...". Below this message are the same two input fields and the "absenden" button as in the previous screenshots.

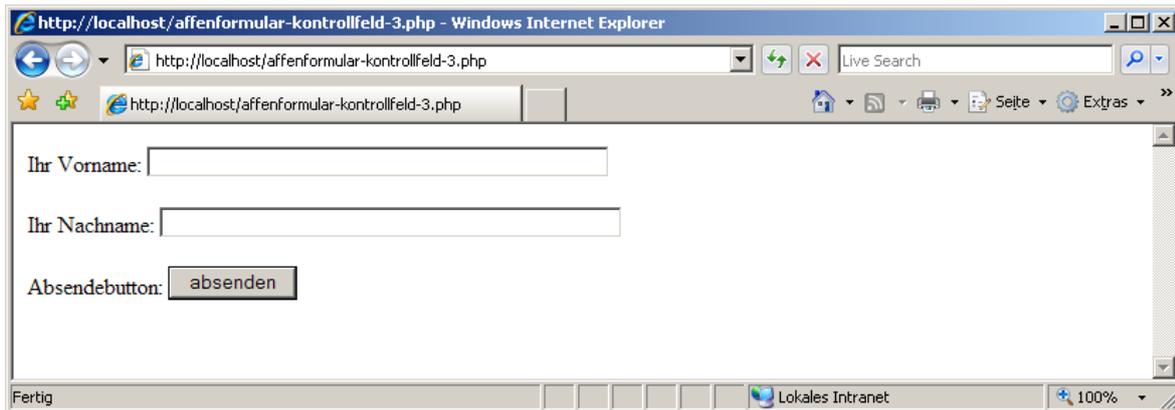
Zur Kontrolle Ihres Ergebnisses ist der Lösungsweg auf der nächsten Seite.

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

```
Weaverslave 4 beta 3.9.18 - [C:\Programme\Apache Group\Apache2\htdocs\affen...
Datei Bearbeiten Tags/Snippets Extras Plugins Optionen Fenster Hilfe
PHP
1 <?php
2 if ( empty ($_GET['vorname']) == TRUE
3     OR empty ($_GET['nachname']) == TRUE )
4 {
5     if ( $_GET['kontrolle'] == 1 )
6     {
7         echo "<p><b>Leere Felder zählen nicht, ";
8         echo "gehen Sie nicht über Los, ziehen Sie ";
9         echo "keine ...</b></p>";
10    }
11    echo '
12    <form action="affenformular-kontrollfeld-2.php"
13    method="GET" name="Formular1" id="Formular1">
14
15    <p>Ihr Vorname:
16    <input type="Text" name="vorname" id="vorname"
17    value="" size="50" maxlength="150" /></p>
18
19    <p>Ihr Nachname:
20    <input type="Text" name="nachname" id="nachname"
21    value="" size="50" maxlength="150" /></p>
22
23    <input type="hidden" name="kontrolle" id="kontrolle"
24    value="1" />
25
26    <p>Absendebutton:
27    <input type="Submit" name="Button" id="Button"
28    value="absenden" /></p>
29    </form>';
30 }
31 else
32 {
33     // beliebige Aktion, z. B. E-Mail senden, DB-Eintrag
34     echo "eingetragener Vorname: ". $_GET['vorname'];
35 }
36 ?>
```

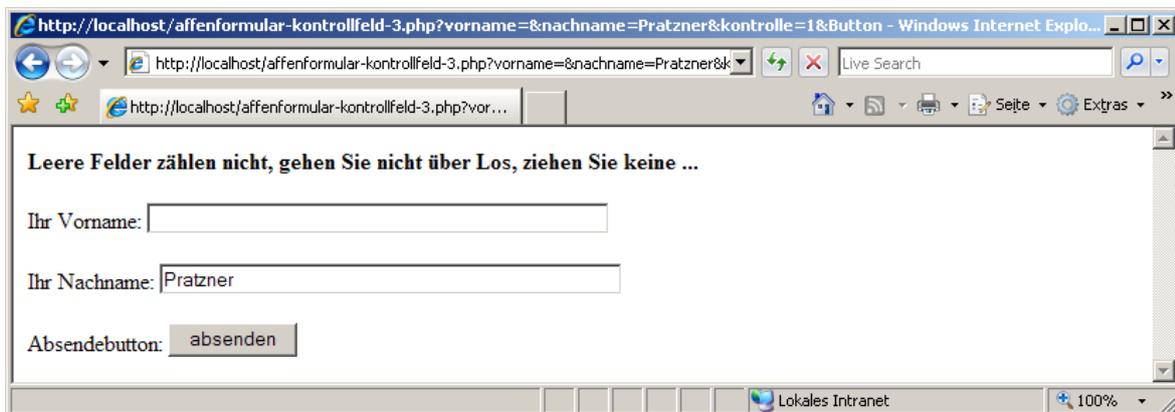
Aufgabe: Fehlermeldung und bereits eingegebene Werte übernehmen

Nutzen Sie das Prinzip der letzten Aufgabe. Annahme ist, dass jemand zwar seinen Nachnamen einträgt, aber den Vornamen nicht einträgt. Im obigen Beispiel muss er dann wieder alle Felder eintragen. Ändern Sie das Script so, dass es zusätzlich zu der Fehlermeldung auch schon den bereits eingetragenen Wert übernimmt.



The screenshot shows a web browser window titled "http://localhost/affenformular-kontrollfeld-3.php - Windows Internet Explorer". The address bar contains "http://localhost/affenformular-kontrollfeld-3.php". The page content includes three input fields: "Ihr Vorname:" (empty), "Ihr Nachname:" (empty), and "Absendebutton: absenden". The status bar at the bottom indicates "Fertig" and "Lokales Intranet".

Eingeben von Nachnamen und absenden - Fehlermeldung kommt und der Nachname wird übernommen und muss nicht nochmals eingetragen werden.



The screenshot shows the same web browser window, but the address bar now contains "http://localhost/affenformular-kontrollfeld-3.php?vorname=&nachname=Pratzner&kontrolle=1&Button". The page content includes a message "Leere Felder zählen nicht, gehen Sie nicht über Los, ziehen Sie keine ..." above the input fields. The "Ihr Vorname:" field is empty, and the "Ihr Nachname:" field is filled with "Pratzner". The "Absendebutton: absenden" is still present. The status bar at the bottom indicates "Lokales Intranet".

Zur Kontrolle Ihres Ergebnisses ein Lösungsweg ansehen.

```

Weaverslave 4 beta 3.9.18 - [C:\Programme\Apache Group\Apache2\htdocs\affen...
Datei Bearbeiten Tags/Snippets Extras Plugins Optionen Fenster Hilfe
PHP
1 <?php
2 if ( empty ( $_GET['vorname'] ) != TRUE
3     OR empty ( $_GET['nachname'] ) == TRUE )
4 {
5     if ( $_GET['kontrolle'] == 1 )
6     {
7         echo "<p><b>Leere Felder zählen nicht, ";
8         echo "gehen Sie nicht über Los, ziehen Sie ";
9         echo "keine ...</b></p>";
10    }
11    echo '
12    <form action="affenformular-kontrollfeld-3.php"
13    method="GET" name="Formular1" id="Formular1">
14
15    <p>Ihr Vorname:
16    <input type="Text" name="vorname" id="vorname"
17    value="' . $_GET['vorname'] .'" size="50"
18    maxlength="150" /></p>
19
20    <p>Ihr Nachname:
21    <input type="Text" name="nachname" id="nachname"
22    value="' . $_GET['nachname'] .'" size="50"
23    maxlength="150" /></p>
24
25    <input type="hidden" name="kontrolle" id="kontrolle"
26    value="1" />
27
28    <p>Absendebutton:
29    <input type="Submit" name="Button" id="Button"
30    value="absenden" /></p>
31    </form>';
32 }
33 else
34 {
35     // beliebige Aktion, z. B. E-Mail senden, DB-Eintrag
36     echo "eingetragener Vorname: " . $_GET['vorname'];
37 }
38 ?>
affenformular-kontrollfeld-3.php
2 31

```

Beispiel aus der Praxis

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Ein direktes Beispiel aus der Praxis ist das Formular zum anfordern des [eBooks vom PHP-Kurs.com](http://www.php-kurs.com). Sind nicht alle Pflichtfelder beim Anklicken des "Absendenbuttons" ausgefüllt worden, erhalten Sie eine Auflistung mit noch notwendigen Angaben und den bereits schon eingetragenen.

Ein weiteres Pluspunkt an diesem Formular ist der fehlende "Abbrechen-Button". So ein Button löscht alle Eingaben in einem Formular. Wird dieser versehentlich gedrückt (was im Eifer des Gefechtes vorkommen kann), werden beim zweiten Ausfüllen (wenn das der Besucher überhaupt macht) die Angaben knapper ausfallen. Daher ist es sinnvoll, solch einen Button erst gar nicht anzubieten.

Dateinamen der PHP-Datei ermitteln

Zeitweise ist es notwendig (und einfacher), den Dateinamen eines PHP-Programmes zu ermitteln.

Der Dateiname ist in der Variablen `$PHP_SELF` hinterlegt. Allerdings kann bei `register_globals = Off` diese nicht direkt abgefragt werden - eine Konstruktion wie im nächsten Beispiel hilft.

```
<?PHP
// bei register_globals = Off
$PHP_SELF = $_SERVER['PHP_SELF'];

echo "Dateiname: ";
print_r ($PHP_SELF);
?>
```

Angewendet wird dies z.B. im Affenformular (siehe Formulare).

Beispiel (nur der Teil, der zum Veranschaulichen wichtig ist):

```
<?PHP
// bei register_globals = off
$PHP_SELF = $_SERVER['PHP_SELF'];

echo '<form action="'. $PHP_SELF .' " method="get" >';
...

```

Diese Methode hat den Vorteil, dass man das Formular auch unter einem beliebigen Namen speichern kann, ohne jedesmal im Formular das "action"-Attribut zu ändern (was man gerne vergisst).

Affenformular Beispiel

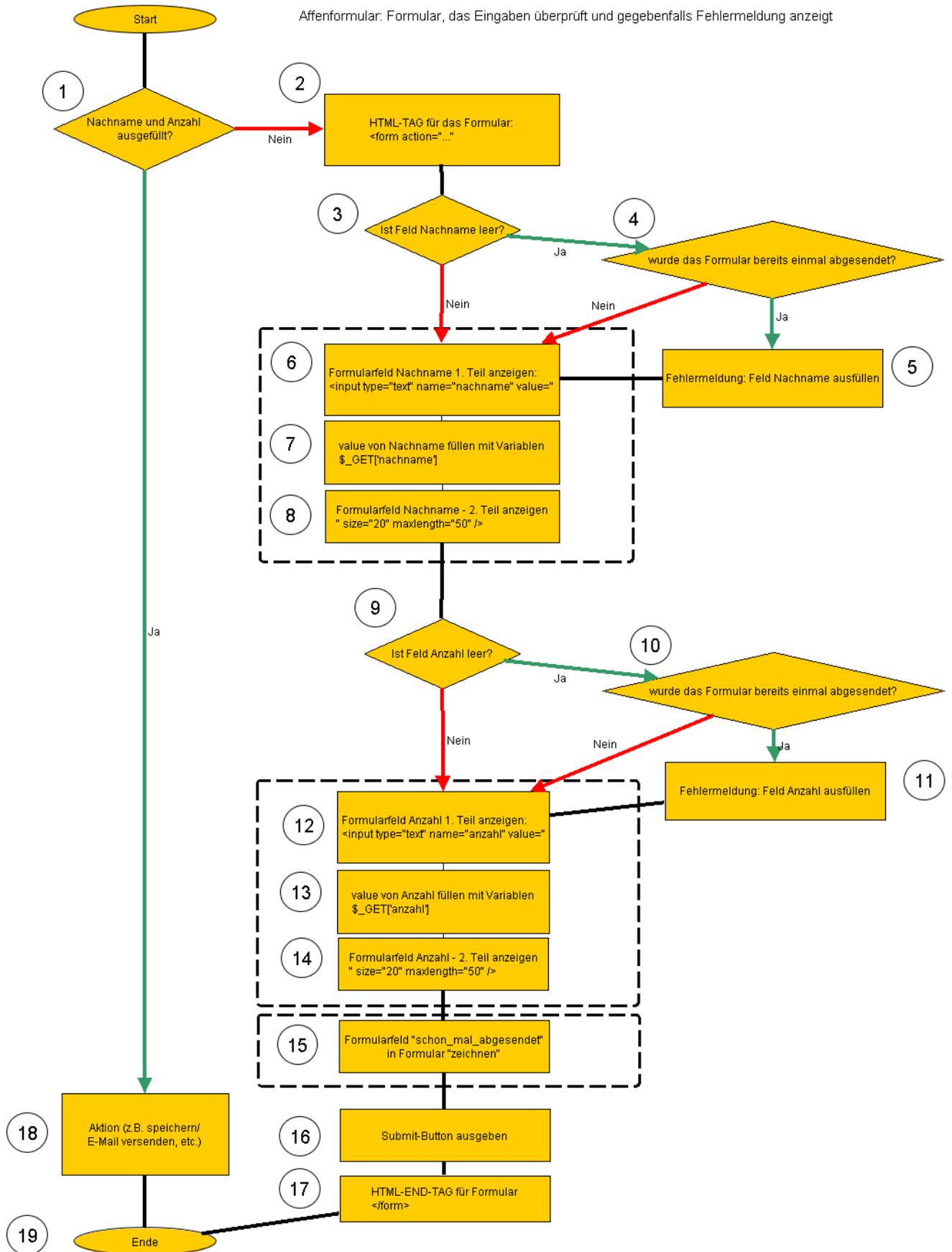
Im Folgenden wird ein Reservierungssystem für z.B. Konzertkarten erstellt. Der Besucher kann seinen Namen hinterlassen und die gewünschte Anzahl an Karten. Vom Programm wird überprüft, ob sowohl Name wie auch Anzahl ausgefüllt wurde. Wird ein Feld (oder beide) vergessen, kommt beim betroffenen Feld ein Hinweis, dass hier die Eingabe fehlt. Dies wird über ein Affenformular gelöst – das Formular ruft sich selber auf, um die Eingaben zu überprüfen. Sind alle Eingaben in Ordnung, wird zum Auswertungsteil gegangen. Fehlen Eingaben, wird das Formular nochmals angezeigt, mit allen bisher gemachten Eingaben (damit der Nutzer nichts doppelt eingeben muss)!

Ablaufplan für Affenformular

Schauen Sie sich im ersten Schritt den Ablaufplan an, um die Logik vor sich zu haben. Zum einfachen nachvollziehen der einzelnen Programmschritte sind im Ablaufplan Nummern für die jeweiligen Schritte hinterlegt, die wieder im Quellcode zu finden sind.

Zum einfachen Nachvollziehen wurden einzelne Felder (z.B. das Formularfeld für den Nachnamen in 3 Schritte unterteilt – vergleiche Punkt 6 bis 8).

Affenformular: Formular, das Eingaben überprüft und gegebenenfalls Fehlermeldung anzeigt



Quellcode für Affenformular-Beispiel

Konzertkartenreservierung

Im folgenden ist der komplette Beispielquellcode für das Karten-Reservierungssystem über ein Affenformular abgebildet. Ein Feedback an den Benutzer, wenn Eingaben leer gelassen werden, die notwendig sind, wird am Bildschirm angezeigt.

```
<?php
// Punkt 1: Abfrage Nachname und Anzahl ausgefüllt?
if ( $_GET['nachname'] == "" OR $_GET['anzahl'] == "" )
{
    // Punkt 2: HTML-Tag für Formular
    echo '<form action="reservierung.php" method="GET">';

    // Formulareintragungen liegen (noch) nicht vor
    // Punkt 3: Feld Nachname leer?
    if ( $_GET['nachname'] == "" )
    {
        // Punkt 4: Formular bereits schon mal aufgerufen
        if ( $_GET['schon_mal_abgesendet'] == "ja" )
        {
            // Punkt 5: Fehlermeldung - Feld Nachname ausfüllen
            echo "<p>Hinweis: Bitte Feld Nachname ausfüllen</p>";
        }
    }

    // Punkt 6: Feld "zeichnen" für Nachname
    echo 'Nachname: ';
    echo '<input type="text" name="nachname" value=""';

    // Punkt 7: ausgabe von eventueller Inhalt von Nachname
    echo $_GET['nachname'];

    // Punkt 8: Ende vom Formularfeld Nachname
    echo '" size="20" maxlength="60" />';

    // Punkt 9: Test, ob Anzahl leer ist
    if ( $_GET['anzahl'] == "" )
    {
        // Punkt 10: Formular bereits schon mal aufgerufen
        if ( $_GET['schon_mal_abgesendet'] == "ja" )
        {
            // Punkt 11: Fehlermeldung - Feld Anzahl ausfüllen
            echo "<p>Hinweis: Bitte Feld Anzahl ausfüllen</p>";
        }
    }

    // Punkt 12: Formularfeld Anzahl zeichnen
    echo '<p>Anzahl der gewünschten Reservierungen:';
    echo '<input type="text" name="anzahl" value=""';

    // Punkt 13: eventuell vorhandene Werte in Vorgabe (value)
    echo $_GET['anzahl'];

    // Punkt 14: Abschluss des HTML-Tags
    echo '" size="4" maxlength="4" />';
}
```

```

// Punkt 15: Feld zur Kontrolle für Zweitaufruf
echo '<input type="hidden" name="schon_mal_abgesendet"
      value="ja" />';

// Punkt 16: Submit-Button zum absenden der Eingaben
echo '<br />';
echo '<input type="Submit" name="" value="absenden" />';

// Punkt 17: Ende vom Formular
echo '</form>';

echo "<p>Bitte alle Felder ausfüllen</p>";
exit;

}
else
{
    // Punkt 18: alle Angaben liegen vor, nun Auswertung
    echo "<p>Folgende Angaben wurden gemacht</p>";

    echo "<p>Nachname :";
    echo $_GET['nachname'];

    echo "<p>Anzahl der Reservierungen";
    echo $_GET['anzahl'];

    // Beispiel: speichern in Datei, Speichern in Datenbank,
    // E-Mail mit Inhalt versenden ...
}
?>

```

Aufgabe zur Erweiterung des Affenformulars: Erweitern Sie das Programm, damit überprüft wird, ob die Angabe der Anzahl der Karten auch eine Zahl ist. Es werden keine halben Plätze vergeben!

per PHP in Dateien schreiben

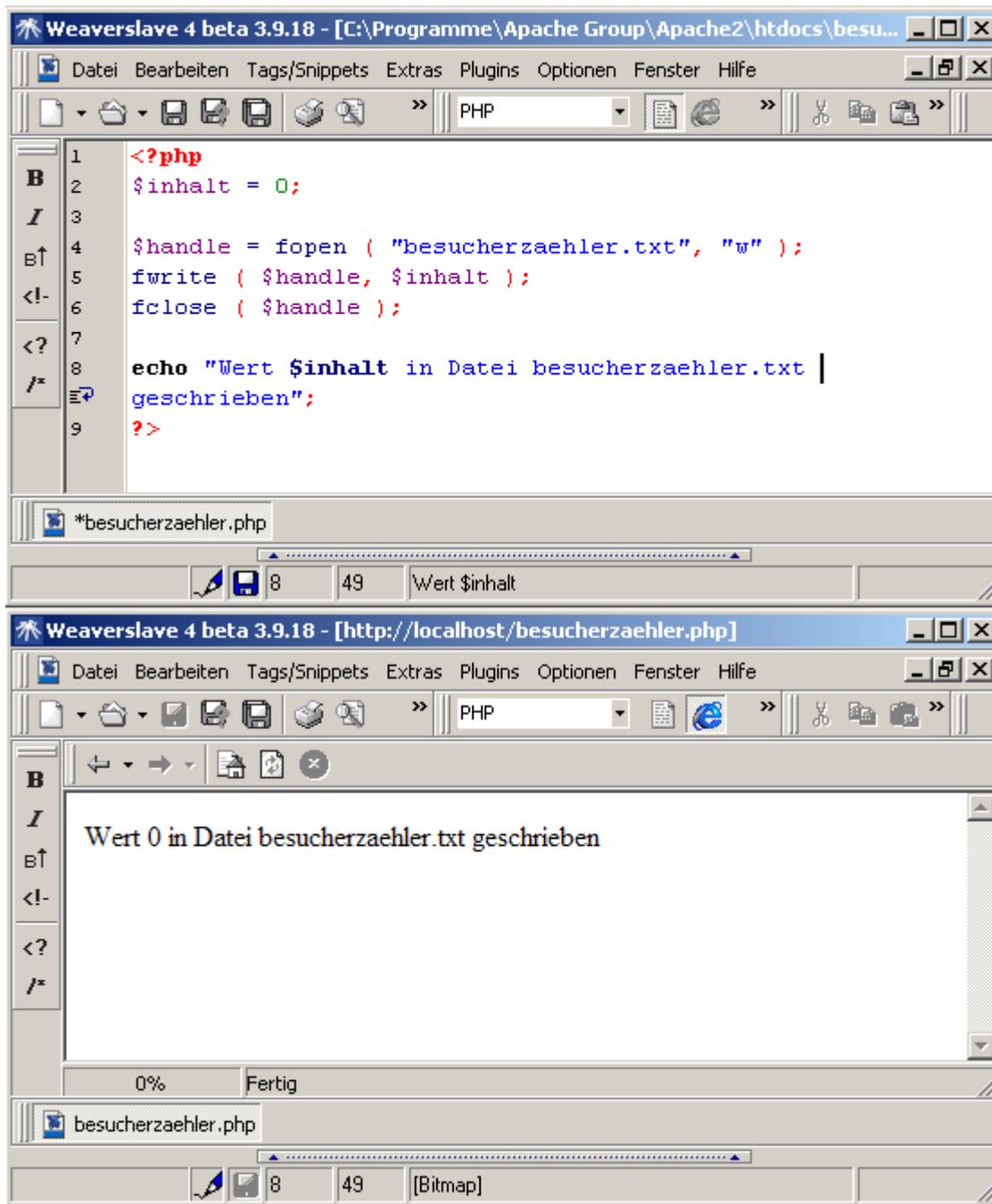
Um Daten zu archivieren, damit diese weiter bearbeitet werden können, speichern wir diese in eine Datei ab.

```

<?php
$inhalt = "0";

$handle = fopen ("besucherzaehler.txt", w);
fwrite($handle, $inhalt);
fclose($handle);
?>

```



Sie können sich die Datei "besucherzaehler.txt" über Ihren Browser ansehen. Geben Sie in der URL ein: <http://localhost/besucherzaehler.txt>

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen



Besucherkähler erstellen

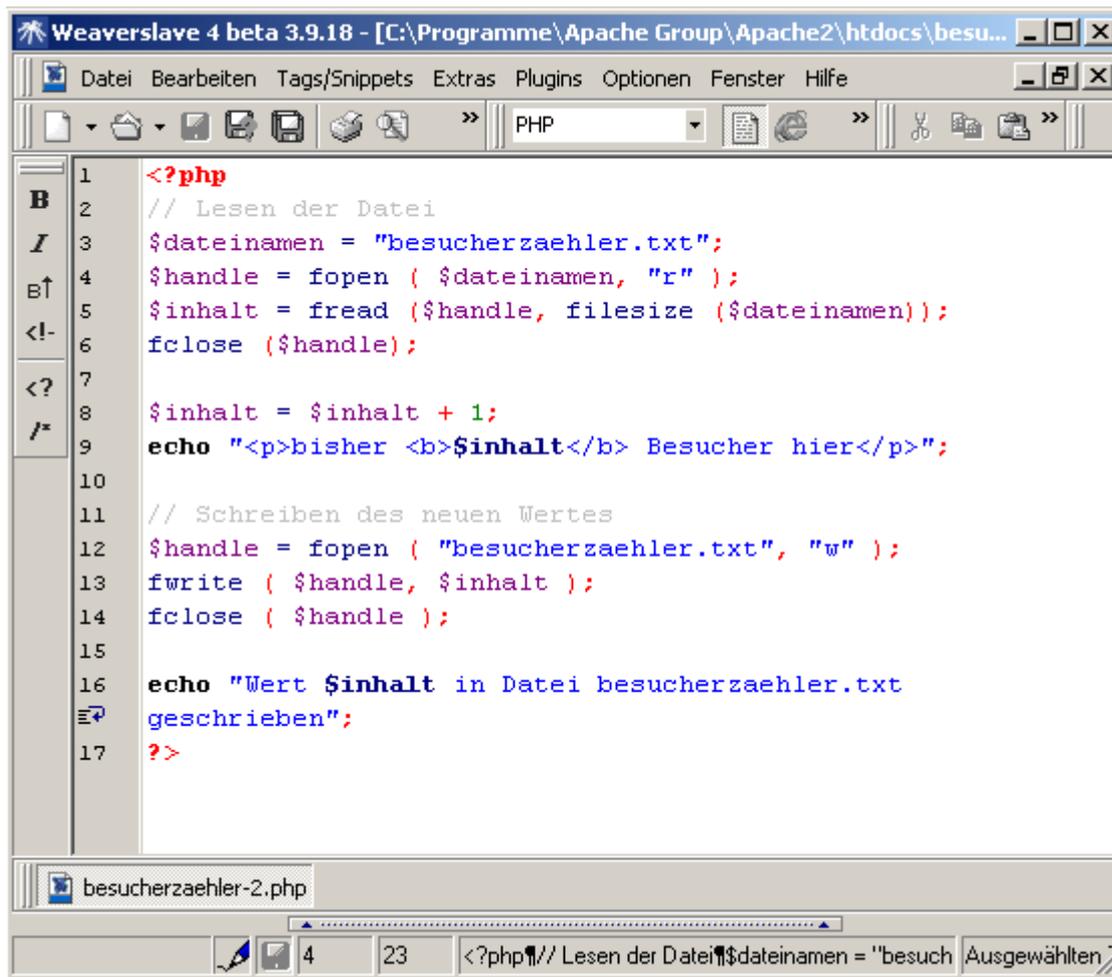
Im ersten Schritt wird die Datei "besucherzaehler.txt" gelesen, damit die Anzahl der bisherigen Besucher bekannt ist. Diese werden dann auf dem Bildschirm ausgegeben und im letzten Schritt wird der neue Wert wieder in der Datei "besucherzaehler.txt" gespeichert. Der alte Inhalt wird dabei gelöscht.

Für das Arbeiten mit Dateien wird als erstes ein "handle" erstellt - die Datei wird geöffnet, dabei wird die Art mitgeteilt, wie die Datei geöffnet wird.

Dateien können in den Modi geöffnet werden:

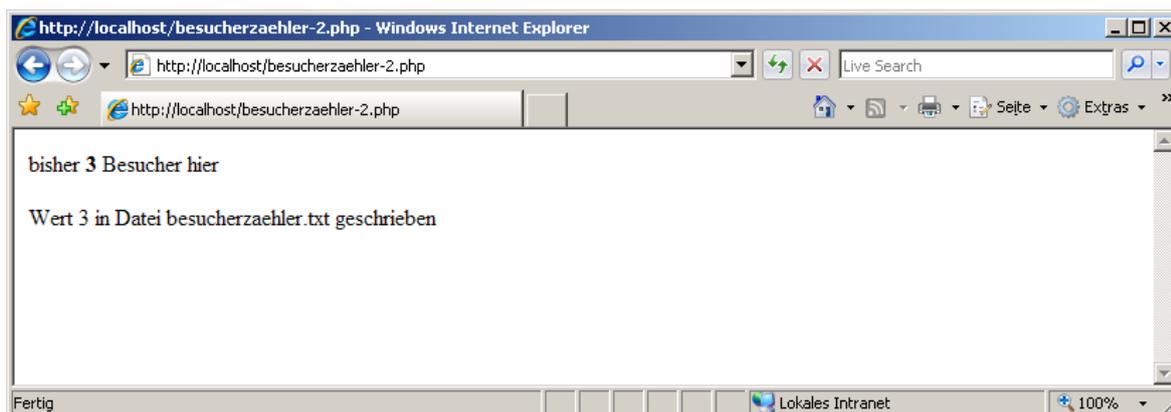
- nur zum Lesen,
- nur zum Schreiben,
- zum Lesen und Schreiben
- und beim Schreiben, ob angehängt wird
- oder bestehender Inhalt ersetzt.

Mehr dazu unter php.net



```
1 <?php
2 // Lesen der Datei
3 $dateinamen = "besucherzaehler.txt";
4 $handle = fopen ( $dateinamen, "r" );
5 $inhalt = fread ($handle, filesize ($dateinamen));
6 fclose ($handle);
7
8 $inhalt = $inhalt + 1;
9 echo "<p>bisher <b>$inhalt</b> Besucher hier</p>";
10
11 // Schreiben des neuen Wertes
12 $handle = fopen ( "besucherzaehler.txt", "w" );
13 fwrite ( $handle, $inhalt );
14 fclose ( $handle );
15
16 echo "Wert $inhalt in Datei besucherzaehler.txt
17 geschrieben";
18 ?>
```

Im Browser sehen Sie die Ausgabe der bisherigen Besucheranzahl



In der Datei `http://localhost/besucherzaehler.txt` finden Sie nur die Zahl der bisherigen Besucher - vergessen Sie nicht den Reload-Button beim Browser zu klicken, nicht dass Sie alte Werte aus dem Cache angezeigt bekommen.



Schleifen

Verschiedene Vorgänge benötigen mehrere Durchgänge, bis ein bestimmter Zustand erreicht ist.

Dazu werden Schleifen benötigt - in PHP gibt es 3 typische Schleifen:

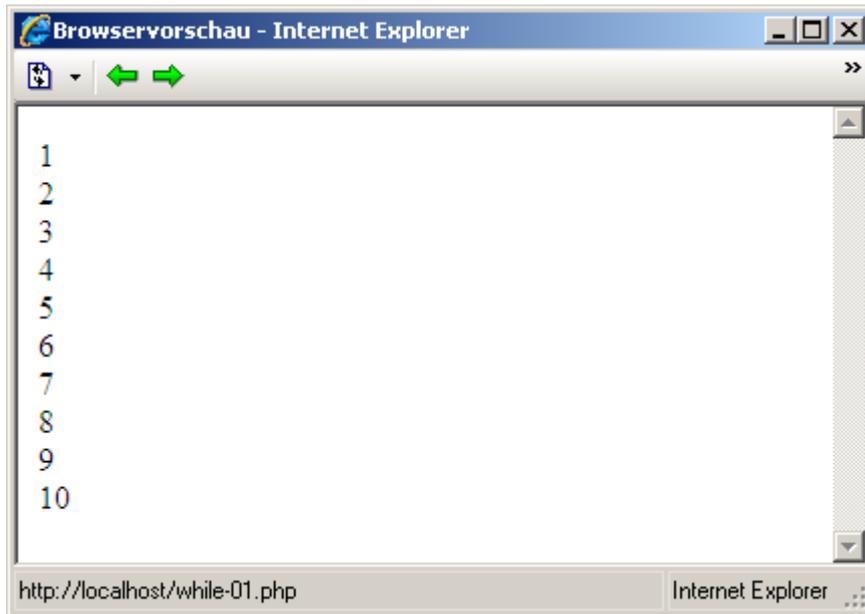
- while
- do..while
- for

Im ersten Beispiel wird die Schleife 10 mal durchlaufen und jeweils die Nummer ausgegeben.

while - Schleife

```
<?php
$i = 1;
while ($i <= 10)
{
    echo $i . "<br />"; // es wird $i ausgegeben,
    $i++; // Wert wird um 1 erhöht
}
?>
```

Durch das `while` wird die Schleife solange durchlaufen, bis die Bedingung erfüllt ist - im Beispiel solange `$i` kleiner als 11 ist.



Die Schleife muss nicht unbedingt durchlaufen werden - das ist der Fall, wenn \$i bereits am Anfang größer als der in der Bedingung gestellte Wert (im Beispiel 10) ist. Probieren Sie folgendes Programm aus:

```
<?php
$i = 11;
while ($i <= 10)
{
    echo $i . "<br />"; // es wird $i ausgegeben,
    $i++; // Wert wird um 1 erhöht
}
?>
```

Als Ausgabe wird ein leerer Bildschirm erscheinen.

do ... while - Kontrolle der Bedingung am Ende

Wenn die Schleife auf jeden Fall einmal durchlaufen werden soll, dann ist die Konstruktion do...while die Richtige.

```
<?php
$i = 11;
do
{
    echo $i . "<br />"; // es wird $i ausgegeben,
    $i++; // Wert wird um 1 erhöht
} while ($i <= 10);
?>
```

Als Ergebnis erhalten Sie die Ausgabe 11 und die Schleife bricht ab, da die Bedingung am Ende der Schleife erfüllt ist. Da die Abfragen erst am Ende kommt, wird die Schleife mindestens einmal durchlaufen.

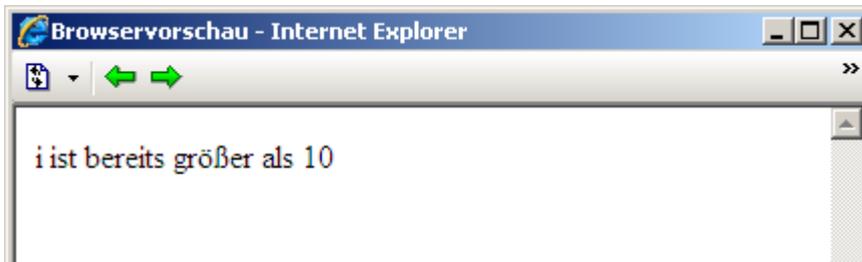
Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen



do ... while Schleifen können auch mittendrin unterbrochen werden, wenn Bedingungen erfüllt sind. Dazu gibt es den break-Befehl. Unser Beispiel von oben wird erweitert:

```
<?php
$i = 11;
do
{
    if ( $i > 10 )
    {
        echo "i ist bereits größer als 10";
        break;
    }

    echo $i . "<br />"; // es wird $i ausgegeben,
    $i++; // Wert wird um 1 erhöht
} while ($i <= 10);
?>
```



for-Schleife

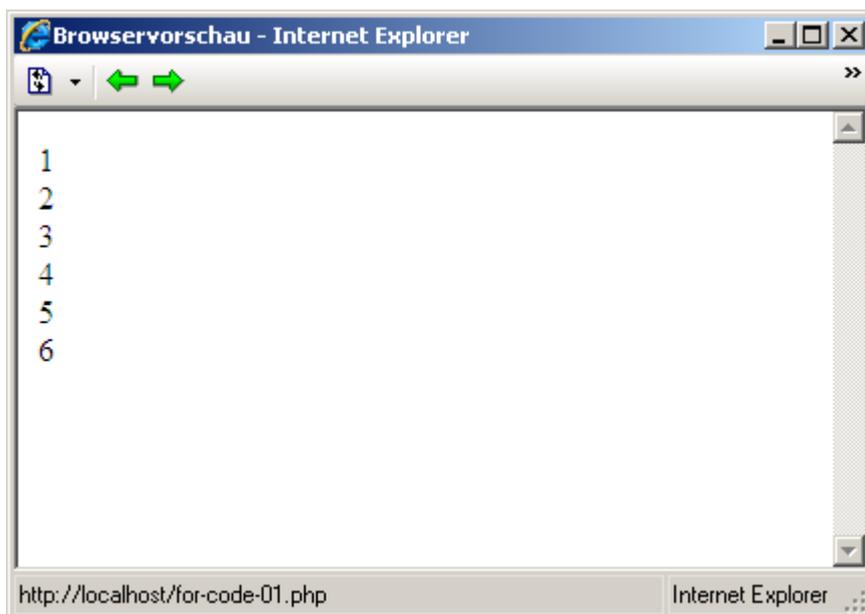
Derselbe Vorgang als for-Schleife.

```
<?php
for ($i = 1; $i <= 10; $i++)
{
    echo $i . "<br />";
}
?>
```



Auch bei der for-Schleife gibt es die Möglichkeit, die Schleife abubrechen.

```
<?php
for ($i = 1; $i <= 10; $i++)
{
    if ( $i > 6 )
    {
        break;
    }
    echo $i . "<br />";
}
?>
```



Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Funktionen in PHP

Funktionen sind flapsig gesagt, kleine Programme im großen Ablauf.

Wenn wie bisher alle Abläufe nacheinander kommen, wird es bei schon mittelgroßen Programmen bereits herzhafte unübersichtlich und es werden unter Umständen immer wieder benötigte Programmteile doppelt und dreifach erstellt. Dies kann über Funktionen umgangen werden.

Als Beispiel kommt hier eine kleine Berechnung, die in eine Funktion aufgegliedert wird.

Aufbau einer Funktion

Die Struktur ist immer dieselbe

Alles GROSS geschriebene sind beliebige Namen und Werte:

```
function NAME-DER-FUNKTION ( ÜBERGABEWERTE )
{
    AUSZUFÜHRENDER CODE;
    return ( ZURÜCKGEBENER WERT );
}
```

Aufgerufen wird eine Funktion dann mit:

```
$ZURUECKGEBENER_WERT = NAME-DER-FUNKTION ( ÜBERGABEWERTE );
```

Vorerst eine ganz einfache Funktion ohne Übergabewerte und ohne Rückwerte.

```
<?php
function ausgabe_uhrzeit()
{
    echo "<p>Es ist gerade: ". date("H:i:s"). "</p>";
}

ausgabe_uhrzeit();
?<
```

Und dann das Ergebnis des Programmes:



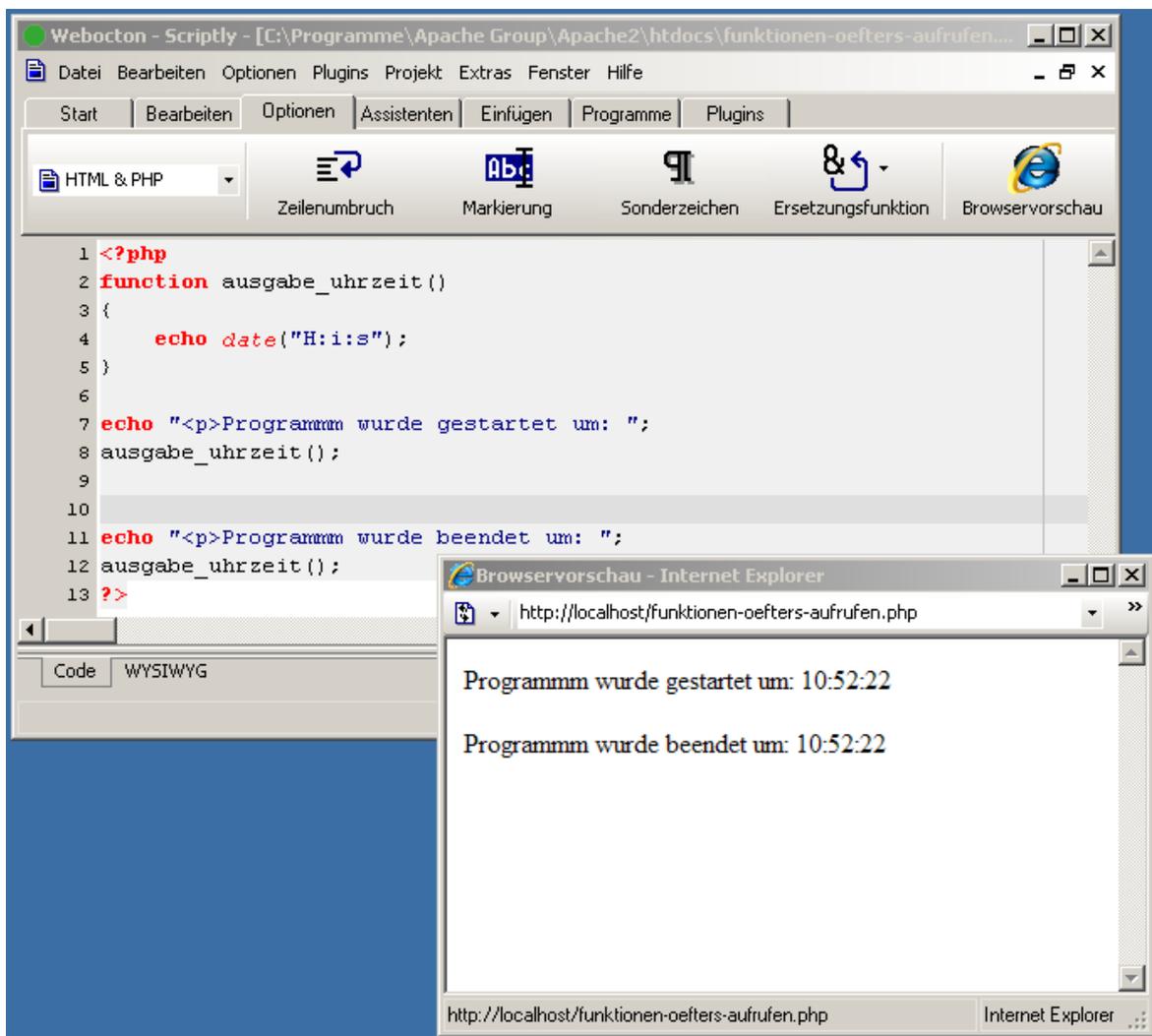
Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Funktionen können beliebig oft aufgerufen werden - im Beispiel sehen Sie, wie schnell auch die Ausführung von PHP-Programmen ist. Die Ausgabe der Uhrzeit erfolgt am Beginn und am Ende (OK, viel mehr macht das Programm nicht:)

```
function ausgabe_uhrzeit()
{
    echo date("H:i:s");
}

echo "<p>Programm wurde gestartet um: ";
ausgabe_uhrzeit();

echo "<p>Programm wurde beendet um: ";
ausgabe_uhrzeit();
```



Funktionen mit Übergabewert

Beim Funktions-Aufruf können Inhalte übergeben werden, im Beispiel `dividieren(18, 2) ;`

Die Funktion selber nimmt in der übergebenen Reihenfolge die Zahlen entgegen und weist diese den Variablen `$zahl` und `$quotient` zu.

```
function dividieren( $zahl, $quotient )
{
    echo bcddiv ( $zahl, $quotient );
}

echo "<p>Berechnung von 18 / 2 = ";
dividieren( 18, 2 );
```

Als berechnete Ergebnis sollte nun angezeigt werden - im Beispiel hier:



Nun werden wir das Programm erweitern, dass es variabler wird - hier nun schon vorneweg Variablen. Für die erste Zahl `$dividend` (die Zahl, die geteilt wird) und `$divisor` (die Zahl, durch die geteilt wird)

```
$dividend = 33;
$divisor = 5;

function dividieren( $zahl, $quotient )
{
    echo bcddiv( $zahl, $quotient );
}

echo "<p>Berechnung von $dividend / $divisor = ";
dividieren( $dividend, $divisor );
```

Das Ergebnis ist für Leute, die es nicht so genau nehmen - für alle andere bitte in www.php.net nachschlagen, wie die Genauigkeit eingestellt werden kann.



Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Gültigkeit von Variablen

Das besondere an Funktionen ist die Gültigkeit von Variablen. Im Beispiel werden alle Variablen vor der Funktion, innerhalb der Funktion und nach der Funktion aufgerufen.

```
$dividend = 33;
$divisor  = 5;

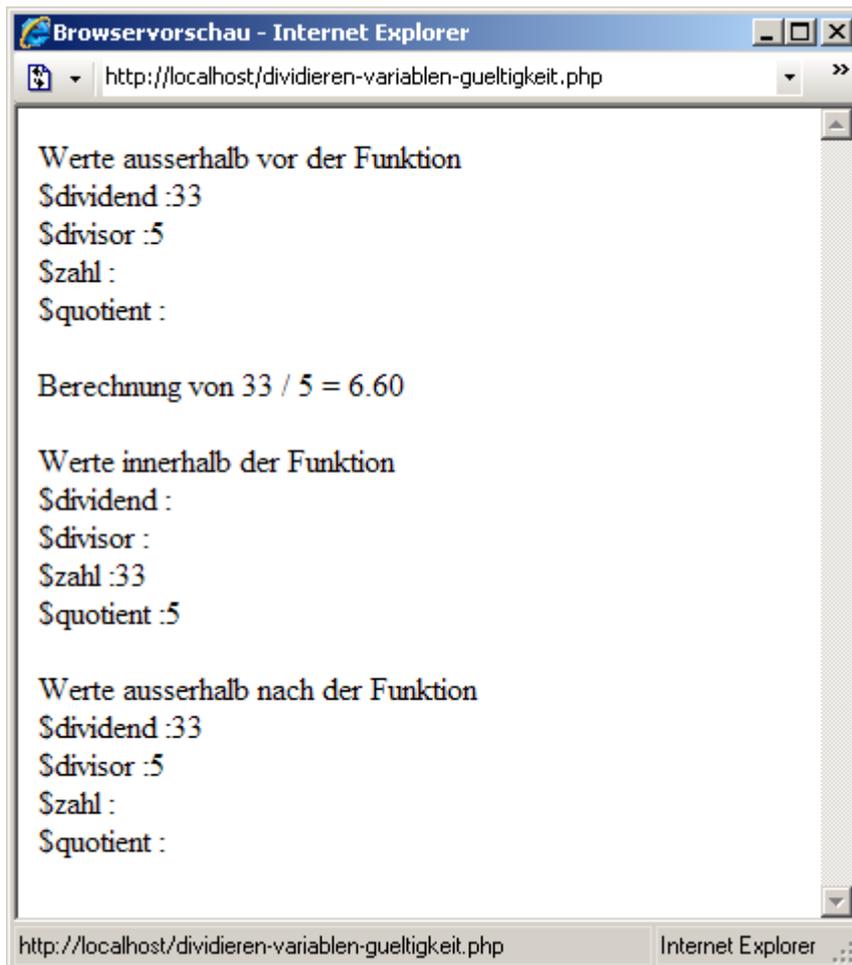
// aus didaktischen Gründen
echo "
Werte außerhalb, vor der Funktion";
echo '<br />$dividend :' . $dividend;
echo '<br />$divisor  :' . $divisor;
echo '<br />$zahl    :' . $zahl;
echo '<br />$quotient :' . $quotient;
echo '<br />';

function dividieren( $zahl, $quotient )
{
    echo bcdiv($zahl, $quotient, 2);
    // aus didaktischen Gründen
    echo "<p>Werte innerhalb der Funktion";
    echo '<br />$dividend :' . $dividend;
    echo '<br />$divisor  :' . $divisor;
    echo '<br />$zahl    :' . $zahl;
    echo '<br />$quotient :' . $quotient;
    echo '<br />';
}

echo "<p>Berechnung von $dividend / $divisor = ";
dividieren( $dividend, $divisor );

// aus didaktischen Gründen
echo "<p>Werte außerhalb, nach der Funktion";
echo '<br />$dividend :' . $dividend;
echo '<br />$divisor  :' . $divisor;
echo '<br />$zahl    :' . $zahl;
echo '<br />$quotient :' . $quotient;
echo '<br />';
```

In der Ausgabe des Browsers sehen wir nun, dass die Variablen andere Werte innerhalb der Funktion haben, wie diese vor und nach der Funktion.



Das bedeutet, dass Variablen, die außerhalb der Funktion Inhalte haben, diese nicht mehr in der Funktion verfügbar haben. Selbst wenn Sie dieselben Variablennamen verwenden. Aus diesem Grund müssen Ergebnisse, die dann z.B. nach Berechnung innerhalb einer Funktion, wieder zurückgegeben werden - dieses geschieht über `return ($wert);`

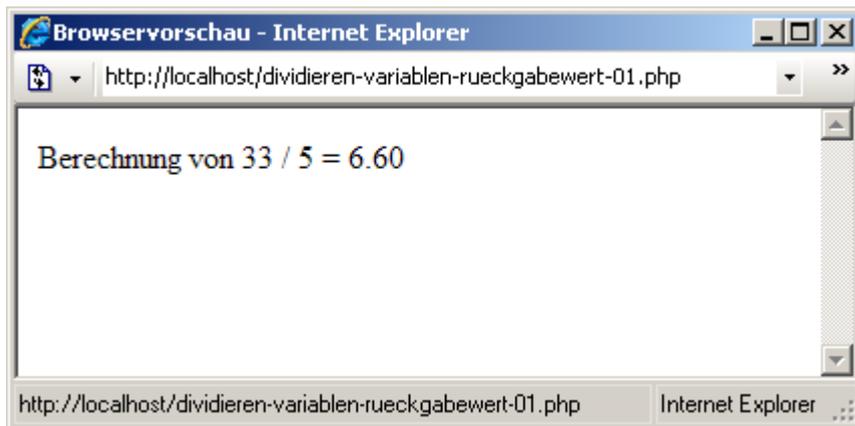
```
$dividend = 33;
$divisor = 5;

function dividieren( $zahl, $quotient )
{
    $ergebnis = bcdiv($zahl, $quotient, 2);
    return ( $ergebnis );
}

$wert = dividieren( $dividend, $divisor );
echo "Berechnung von $dividend / $divisor = $wert ";
```

Da die Gültigkeit der Variablen `$ergebnis` nur innerhalb der Funktion gilt, wird der Wert über den Funktionsaufruf der Variable `$ergebnis` (kann auch beliebig anders genannt werden) zugewiesen `$wert = dividieren($dividend, $divisor);`

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen



Vorgabewerte für Funktionen bestimmen

Funktionen mit Vorgabewerten zu belegen, kann so geschehen:

```
function dividieren( $zahl=1, $quotient=1 )  
{
```

Wird keine Zahl beim Funktionsaufruf übergeben, wird einfach der Vorgabewert genutzt, im obigen Beispiel dann "1".

Aufgabe: Sicherheitkontrolle in Funktion auf 0

Da durch 0 nicht geteilt werden darf, soll innerhalb der Funktion eine Sicherheitsabfrage stattfinden und gegebenenfalls ein Hinweis kommen.



Ergebnis auf folgender Seite

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

The screenshot shows the Webtocton - Scriptly IDE interface. The title bar indicates the file path: [C:\Programme\Apache Group\Apache2\htdocs\dividieren-sicherheitsabfrage...]. The menu bar includes Datei, Bearbeiten, Optionen, Plugins, Projekt, Extras, Fenster, and Hilfe. The toolbar contains icons for HTML & PHP, Zeilenumbruch, Markierung, Sonderzeichen, Ersetzungsfunktion, and Browservorschau. The main editor area displays the following PHP code:

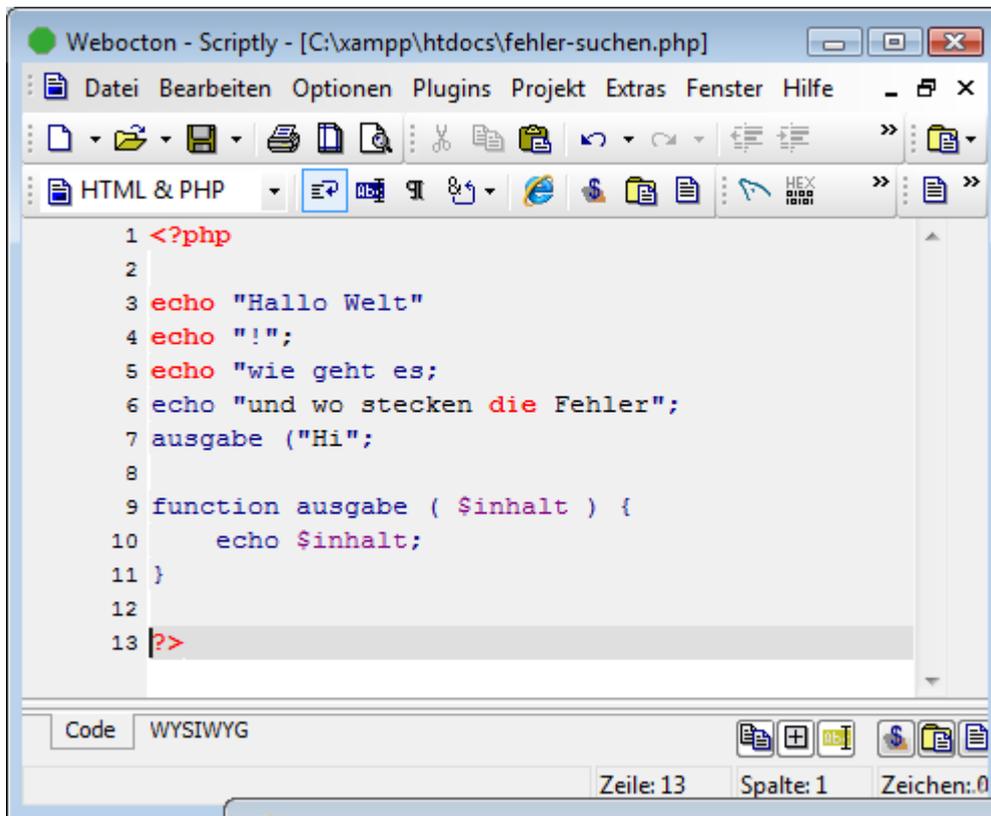
```
1 <?php
2 $dividend = 33;
3 $divisor = 0;
4
5 function dividieren( $zahl=1, $quotient=1 )
6 {
7     if ( $quotient == 0)
8     {
9         echo "<p>Teilen durch 0 nicht möglich</p>";
10    }
11    else
12    {
13        $ergebnis = bcdiv($zahl, $quotient, 2);
14    }
15    return ( $ergebnis );
16 }
17
18 $ergebnis = dividieren( $dividend, $divisor );
19 echo "Berechnung von $dividend / $divisor = $ergebnis ";
20 ?>
```

The status bar at the bottom shows "Code WYSIWYG" and "Zeile: 18 Spalte: 44 Zeichen: 0 Einfügen".

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

häufige Fehlerquellen

Wer arbeitet, macht Fehler :) Und Fehler gehören zum Lernprozess! Es gibt häufige Fehler (die selbst Profis zeitweise unterlaufen) - im Folgenden eine Auflistung und wie Sie sich auf die Suche im Quellcode machen.



```
1 <?php
2
3 echo "Hallo Welt"
4 echo "!";
5 echo "wie geht es;
6 echo "und wo stecken die Fehler";
7 ausgabe ("Hi";
8
9 function ausgabe ( $inhalt ) {
10     echo $inhalt;
11 }
12
13 ?>
```

vergessenes Semikolon

Ein absoluter Spitzenreiter ist das vergessene Semikolon (;) - als Fehlermeldung erhalten Sie dann auch prompt wie im Beispiel gezeigt, die falsche Zeile. Es ist nicht Zeile 4, die Probleme macht, sondern 3. Und warum kommt der Fehler laut der Fehlermeldung in der vierten Zeile? Ganz einfach - durch das fehlende Semikolon war der Befehl in der dritten Zeile noch nicht abgeschlossen und geht so in der folgenden Zeile weiter!

Resultierende Fehlermeldung:

```
Parse error: syntax error, unexpected T_ECHO, expecting ',' or ';'
in C:\xampp\htdocs\fehler-suchen.php on line 4
```

Zum Vermeiden, am besten immer nur 1 Befehl pro Zeile (dann ist es übersichtlicher) und am besten sofort das Semikolon schreiben

Merke: Bei einer Fehlermeldung ab der angegebenen Zeile nach oben weg suchen!

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

vergessenes zweites Anführungszeichen

Obwohl Anführungszeichen meistens im Doppelpack auftauchen und Inhalt umschließen, wird auch dort gern das zweite vergessen - Beispiel Zeile 5

Resultierende Fehlermeldung:

```
Parse error: syntax error, unexpected T_STRING, expecting ',' or ';'
in C:\xampp\htdocs\fehler-suchen.php on line 6
```

Am besten sofort beide Anführungszeichen schreiben und dann erst den Inhalt zwischen den Anführungszeichen.

vergessene Abschlussklammer

Auch sehr häufig kommt das Vergessen der Abschlussklammer vor (siehe Beispiel Zeile 7).

Resultierende Fehlermeldung:

```
Parse error: syntax error, unexpected ';'
in C:\xampp\htdocs\fehler-suchen.php on line 7
```

Das Vergessen der Abschlussklammern lässt sich am besten dadurch vermeiden, dass bei Benötigen der Anfangsklammer sofort die Abschlussklammer geschrieben wird und dann erst der Inhalt zwischen den Klammern.

automatische Hilfen

Viele PHP-Editoren unterstützen Sie automatisch dabei. Scriptly z.B. setzt automatisch die Klammern und die doppelten Anführungszeichen - z. B. zu- und abschaltbar bei Scriptly unter: Optionen -> Ersetzen -> (durch ()

Farbiger Quellcode hilft bei der Orientierung im Code. Falls Sie keine farbige Ausgabe haben sollten, können Sie diese sich selber programmieren. Erstellen Sie folgendes PHP-Programm

```
show_source('dateiname.php');
```

Die Ausgabe Ihres Quellcodes der Datei 'dateiname.php' erfolgt dann mit hervorgehobener Syntax. Die Farben des in PHP eingebauten Syntax-Highlighter werden dafür benutzt.

Fortgeschrittene

Verschiedene Kapitel werden hier nochmals vertiefend und detaillierter besprochen - so kommen z.B. bei den Variablen die verschiedenen Variablen-Typen, Typenumwandlung und Konstanten.

Über Arrays können einfach Daten in Variablen verwaltet, sortiert und ausgegeben werden.

Kontrollstrukturen mit switch, if und elseif werden besprochen.

Erstellen von Funktionsbibliotheken und das Einbinden in Programmen erspart viel Arbeit, da verschiedene Funktionen immer wieder benötigt werden.

Formularinhalte in Dateien speichern

Nachdem wir nun über ein Formular die Daten erfasst und kontrolliert haben, werden die Daten zeilenweise in einer Datei gespeichert. Im folgenden Beispiel ist der Bereich der Dateierfassung per Formular auf das nötigste reduziert - für die Datenkontrolle und Validierung sehen Sie bitte im Kapitel "Formularauswertung" nach. Hier geht es jetzt primär um das Speichern der erfassten Inhalte.

```
<?php
// unser Grundformular gespeichert unter dem Namen: emailerfassung.php
?>

<form action="emailerfassung.php" method="get">

<p>Ihre E-Mail-Adresse<br />
<input type="Text" name="email" ></p>

<p>Name:<br />
<input type="Text" name="name" ></p>

<input type="Submit" name="" value="speichern">

</form>
```

Wie beim Affenformular kommt nun der auswertende Teil hinzu. Dabei sollte für den laufenden Betrieb noch eine Evaluierung der Daten erfolgen, siehe Kapitel "Formularauswertung".

```
<?php

if ( $_GET['email'] <> "" )
{
    // und nun die Daten in eine Datei schreiben
    // Datei wird zum Schreiben geöffnet
    $handle = fopen ( "anfragen.txt", "w" );

    // schreiben des Inhaltes von email
    fwrite ( $handle, $_GET['email'] );
}
```

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

```

// Trennzeichen einfügen, damit Auswertung möglich wird
fwrite ( $handle, "|" );

// schreiben des Inhalts von name
fwrite ( $handle, $_GET['name'] );

// Datei schließen
fclose ( $handle );

echo "Danke - Ihre Daten wurden speichert";

// Datei wird nicht weiter ausgeführt
exit;
}

?>

<form action="emailerfassung.php" method="get">

<p>Ihre E-Mail-Adresse<br />
<input type="Text" name="email" ></p>

<p>Name:<br />
<input type="Text" name="name" ></p>

<input type="Submit" name="" value="speichern">

</form>

```

Steuerzeichen

<http://de.wikipedia.org/wiki/Steuerzeichen>

```

" " (ASCII 32 (0x20)), ein normales Leerzeichen.
"\t" (ASCII 9 (0x09)), ein Tabulatorzeichen.
"\n" (ASCII 10 (0x0A)), einen Zeilenvorschub (Line Feed).
"\r" (ASCII 13 (0x0D)), ein Wagenrücklaufzeichen (Carriage Return).
"\0" (ASCII 0 (0x00)), das NUL-Byte.
"\x0B" (ASCII 11 (0x0B)), ein vertikaler Tabulator.

```

Mit dem PHP-Befehl `trim()` werden Steuerzeichen am Ende und am Anfang entfernt.

Eine geschickte Anwendung ist das Konvertieren von Daten aus einer Excel-Tabelle in Arrays:

Die Daten werden in der Excel-Tabelle markiert und dann in die Zwischenablage kopiert. Unser PHP-Programm bietet ein Formularfeld, in dem wir unseren Inhalt der Zwischenablage einfügen und dann das Programm starten.

Das Programm geht den Inhalt des Formularfelds durch und sucht sich die Zeilenende anhand dem Steuerzeichen `"\n"` heraus. Im nächsten Schritt teilt es die einzelnen Felder aus - das ist einfach möglich, da die Felder durch den Steuercode für das Tabulatorenzeichen getrennt sind `"\t"`

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

```

<?php

echo "Konverter um ". date ("H:i:s");

echo '<form name="" action="" method="post" enctype="text/html">
    <textarea name="inhalt" rows="10" cols="60">'. $_REQUEST['inhalt']
.</textarea><br />
    <input type="Submit" name="" value="konvertieren" /><br />
</form>
';

$daten = split("\n", $_REQUEST['inhalt']);
echo "<pre>";
print_r ($daten);

foreach ($daten AS $inhaltsdaten)
{
    $add_daten[] = split("\t", $inhaltsdaten);
}
echo "<hr />";
print_r ($add_daten);
echo "</pre>";

?>

```

Dateien auslesen

Um eine Datei auszulesen, muss im ersten Schritt über `fopen` ein Dateistream erzeugt werden. Dabei wird der Dateinamen angegeben und auf welche Art mit der Datei umgegangen wird.

```
$handle = fopen ("mailadressen.txt", r);
```

Folgende verschiedene Arten des Dateihandling gibt es:

In der Spalte der Tabelle namens **Zeiger** ist der Startpunkt des Lesen/Schreibens ersichtlich. Wird dieser auf Anfang gestellt, werden bestehende Daten überschrieben!

In der Spalte **erzeugen** ist ersichtlich, ob die Datei automatisch anlegt wird, wenn diese noch nicht vorhanden ist.

Modus	engl.	Aktion	Zeiger	erzeugen
a+	append - anfügen	lesen und schreiben	Ende	Ja
a	append - anfügen	nur schreiben	Ende	Ja
w+	write - schreiben	lesen und schreiben	Beginn	Ja
w	write - schreiben	nur schreiben	Beginn	Ja
r+	read - lesen	lesen und schreiben	Beginn	Nope
r	read - lesen	nur lesen	Beginn	Nope

Durch das while wird die Datei Zeile für Zeile durchgegangen, bis keine Zeile mehr geliefert wird.

```
<?php
// Datei öffnen zum lesen und schreiben
$handle = fopen ("mailadressen.txt", r);

while ( $inhalt = fgets ($handle, 4096 ))
{
    echo "<li> $inhalt </li>";
}

fclose($handle);
?>
```

Verzeichnis einlesen

Je nach Aufgabenstellung ist es praktisch, den Inhalt eines Verzeichnisses auslesen zu lassen. Dazu geht man folgenden Weg:

Das gewünschte Verzeichnis wird in einer Variablen hinterlegt, im folgenden Beispiel in der Variable \$verzeichnis

Dann wird überprüft, ob es sich wirklich um ein Verzeichnis handelt.

Ist es ein Verzeichnis wird über opendir ein Handle erstellt, mit dem dann weiter gearbeitet wird.

Jetzt wird das Verzeichnis mit readdir in einer while-Schleife durchlaufen, bis das Verzeichnisende erreicht ist. Wir erhalten von der Funktion ein false als Rückgabewert und beenden die while-Schleife.

Innerhalb der Schleife können wir nun mit der Funktion filetype testen, ob es sich um eine Datei (file) oder um ein Unterverzeichnis (dir) handelt.

Im Beispiel wird der Inhalt des aktuellen Verzeichnisses ausgegeben.

```
// Der Punkt steht für das Verzeichnis, in der auch dieses
// PHP-Programm gespeichert ist
$verzeichnis = ".";
echo "<ol>";

// Text, ob ein Verzeichnis angegeben wurde
if ( is_dir ( $verzeichnis ))
{
    // öffnen des Verzeichnisses
    if ( $handle = opendir($verzeichnis) )
    {
        // einlesen der Verzeichnisses
        while (($file = readdir($handle)) !== false)
        {
            echo "<li>Dateiname: ";
        }
    }
}
```

```

        echo $file;

        echo "<ul><li>Dateityp: ";
        echo filetype( $file );
        echo "</li></ul>\n";
    }
    closedir($handle);
}
}
echo "</ol>";

```

Interessant wird die Weiterverarbeitung der Dateinamen. Dazu eignen sich besonders Array - siehe das Kapitel "Arrays". Als Beispiel für die Anwendung ist das Kapitel Bildergalerie gedacht.

Variablen-Typen

PHP unterstützt verschiedene Typen von Variablen.

Integer

Ganzzahlen (also keine Kommastellen oder Rundungsprobleme :)

Fließkomma-Zahl (float)

Zahlen mit Nachkommestellen, die beim Runden beachtet werden sollten

String / Zeichenkette Zeichenketten, in denen alles beliebige stehen kann (auch eine Zahl kann als Zeichenkette definiert sein)

Boolean

Wahrheitswert, wird normalerweise mit TRUE oder FALSE gesetzt und abgefragt

Array

Variablen "auf Extasy" - können große Datenmengen bequem aufnehmen (siehe Kapitel Array)

Beispiele zu Variablen-Typen

```

$var_integer = 42; // ein Integer (Ganzzahl)
$var_string  = "ax"; // ein String (Zeichenkette)
$var_string  = "42"; // sieht auf dem ersten Blick nach
                    // Integer aus, ist aber ein String
$var_boolean = TRUE; // ein Boolean (Wahrheitswert)

// Abfrage der Variablentyps
echo gettype($variablenname);

```

Variablen und Formulare

Die Sicherheitsregel lautet, traue keiner Benutzereingabe. Daher ist es wichtig, wenn z. B. eine Ganzzahl erwartet wird, diese über die entsprechende Funktion zu INTEGER zu machen.

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Müllangaben fliegen raus, saubere Angaben bleiben erhalten und man kann beim Auswerten von einer sauberen Basis ausgehen.

Die Typensetzung ist sehr einfach möglich.

```
$var_auf_jeden_fall_int = (int) $variabel;
```

Folgende Umwandlungen sind durch Angaben der Klammerninhalte samt Klammer möglich:

- (int), (integer) - nach integer
- (float), (double), (real) - nach float
- (string) - nach string
- (bool), (boolean) - nach boolean
- (array) - nach array

Das bedeutet, dass vor Abspeichern der Daten in einer Datei oder Datenbank diese einfach zur Sicherheit auf ihren entsprechenden Typ gesetzt werden.

Möchte man es perfekt machen, kann der Nutzer auch im Formular ein Feedback bekommen, dass irgendwas mit der Eingabe nicht stimmt, z.B. wenn die erwartete Integer-Eingabe keine ist.

Beispiel Test Integer-Angabe

```
if ( $test_variable != (INT) $test_variable )
{
    echo "Variable hat falschen Inhalt";
    echo " - es sind nur ganze Zahlen zulässig";
}
```

Genauso kann das Testen über die entsprechende PHP-Funktion stattfinden:

```
is_int()
is_float()
is_string()
is_array()
is_bool()
```

Konstanten - fixe Variablen

Was nun, konstant oder variabel? - Konstanten werden einmal definiert und sind in allen Programmteilen verfügbar (auch in Funktionen). Das ist der große Unterschied zu Variablen.

Als Beispiel könnte z.B. die aktuelle Programmversion als Konstante definiert werden oder die Pfade, wo die Bilder gespeichert sind.

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Namen von Konstanten werden komplett groß geschrieben - so ist sofort ersichtlich, dass es sich um eine Konstante handelt. Weiterhin sind keine Sonderzeichen oder Zahlen am Anfang zulässig.

```
// Gueltige Namen bei Konstanten
define("VERSION",      "1.43");
define("KONTAKTEMAIL", "ich@da.dee");

// Ungueltige Namen bei Konstanten
define("2COOL",       "ich bin cool aber unwissend");
define("2cOoL",      "ich bin cool aber unwissend");

// Ausgabe einer Konstanten
echo VERSION;
```

Konstanten können nur skalare Daten beinhalten, also Integer, Strings, Float und Boolean - keine Arrays.

Bedingungen if elseif

Über if und elseif können komplexe Kontrollstrukturen aufgebaut werden. So wird im folgenden Beispiel kontrolliert, ob eine Variable kleiner ist; wenn nicht, wird kontrolliert (elseif), ob diese gleich groß ist, ansonsten wird der Text "a ist größer als b" ausgegeben.

Es können auch noch mehrere elseifs mit weiteren Bedingungen integriert werden (was bei größer, gleich, kleiner wenig Sinn macht).

```
<?php
$a = 10;
$b = 5;

if ( $a < $b )
{
    echo "a ist kleiner als b";
}
elseif ($a == $b)
{
    echo "a ist gleich b";
}
else
{
    echo "a ist größer als b";
}
?>
```

Aufgabe:

Schreiben Sie ein Programm, das kontrolliert, ob eine Zahl größer als 10 ist, ob eine Zahl größer als 6 ist, ob diese gleich 6 ist oder kleiner - und das die entsprechende Ausgabe macht.

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Bedingung switch

Aufbau einer Kontrollstruktur - je nachdem welche Anweisung erfüllt ist, wird der nachfolgende Bereich ausgeführt.

Im folgenden Beispiel wird kontrolliert, ob eine Anrede "Herr" oder "Frau" ist oder nicht vorliegt und die entsprechende Anrede für den Brief-Empfänger ausgegeben.

```
$anrede = "m";

switch ($anrede)
{
  case "w":
    echo "Sehr geehrte Frau ...";
    break;
  case "m":
    echo "Sehr geehrter Herr ...";
    break;
  default:
    echo "Sehr geehrte Damen und Herren,";
}
```

Wird keine Bedingung erfüllt, erfolgt die Ausführung des Programmteils unterhalb von `default`.

Daten vergleichen

Sehr oft kommt es vor, dass bestehende Daten mit neuen verglichen werden müssen. Im folgenden Beispiel soll ermittelt werden, ob eine bestimmte E-Mail-Adresse bereits vorhanden ist. Ist diese nicht vorhanden, soll sie mit in die Sammlung aufgenommen werden.

Dazu lesen wir eine Datei zeilenweise ein und vergleichen diese dann - wird ein identischer Eintrag gefunden, wird abgebrochen. Ansonsten wird die E-Mail-Adresse gespeichert.

```
<?php
$email = "dagobert@duck.ente";

// Datei öffnen zum Lesen und Schreiben
$handle = fopen ("mailadressen.txt", "a+");

while ( $inhalt = fgets ($handle, 4096 ))
{
  $inhalt = trim ( $inhalt );
  echo "<li> |". $inhalt ."| </li>";
  if ( trim ($inhalt) == $email)
  {
    echo "E-Mail-Adresse bereits vorhanden";
    exit;
  }
}

fwrite($handle, $email);
```

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

```
// Zeilenumbruch einfügen
fwrite($handle, "\r\n");

fclose($handle);
?>
```

Übung

Integrieren Sie hier ein Formular, über das eine E-Mail-Adresse erfasst werden kann und die dann in die Datei geschrieben wird, wenn die E-Mail-Adresse noch nicht vorhanden ist (wie im obigen Beispiel). Für Beispiele von Formular siehe Kapitel: Affenformular

E-Mails mit PHP erstellen

Die Funktion `mail` unter PHP erwartet verschiedene Übergabewerte. Diese werden der Funktion `mail` in der Klammer übergeben (hier über Variablen).

Folgende Werte werden erwartet:

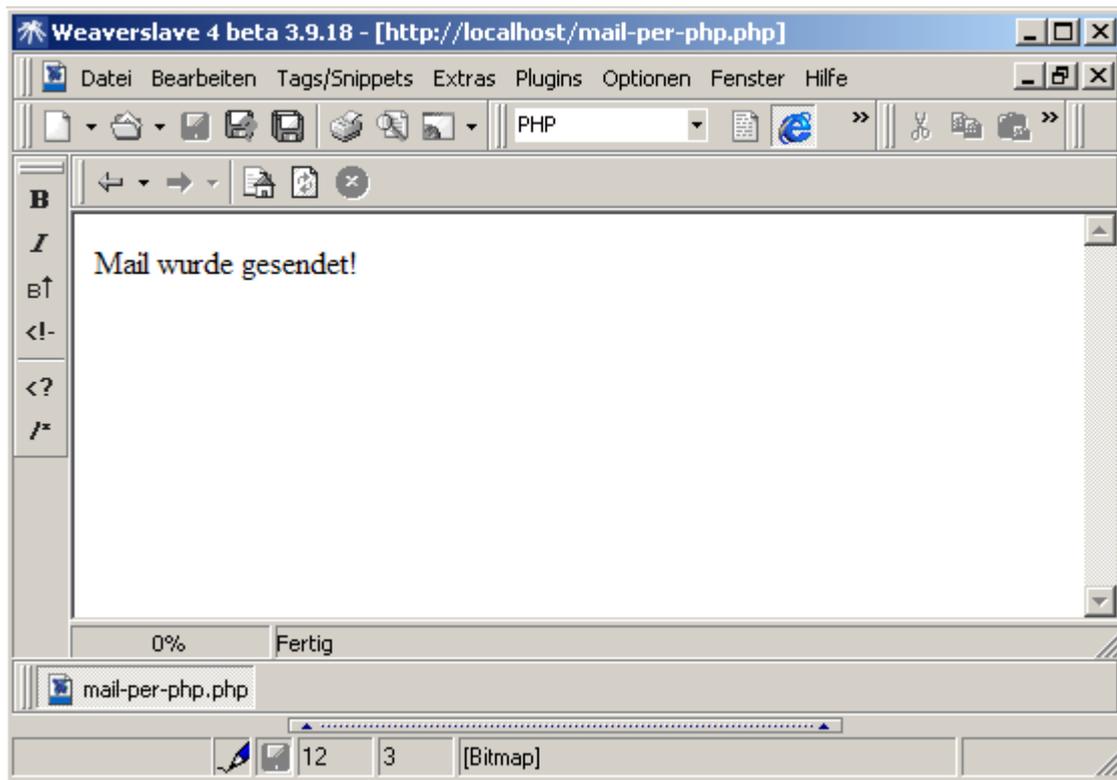
- Empfänger-E-Mail-Adresse: in der Regel Ihre eigene
- Absender-E-Mail-Adresse: von wem wird diese E-Mail abgesendet
- Betreff: Alles, was in der Betreffzeile erscheinen soll
- Mailtext: der eigentliche Inhalt der E-Mail
- antwortan: an wen ein Replay erfolgen soll (oft identisch mit Empfänger)

Erstellen Sie folgendes Programm, wobei Sie als Empfänger Ihre eigene E-Mail-Adresse eintippen. Führen Sie das Programm aus und schauen Sie in Ihrem E-Mail-Programm nach, ob die E-Mail angekommen ist. In der Regel sollte diese 1-2 Minuten später bei Ihnen eintreffen.

```
<?php
$empfaenger = "du@testkarnickel.de"; //Mailadresse
$absender   = "ich@testkarnickel.de";
$betreff    = "PHP-Mail-Test";
$mailtext   = "Inhalt einer Mail zum Test von PHP";
$antwortan  = "ICH@testkarnickel.de";

mail( $empfaenger,
      $betreff,
      $mailtext,
      "From: $absender\nReply-To: $antwortan");

echo "Mail wurde gesendet!";
?>
```



HTML-E-Mail mit PHP erstellen

Eine HTML E-Mail mit PHP zu erstellen ist nicht schwierig. Dazu muss man nur wissen, wie man eine HTML-Seite erstellt und dass im Header der E-Mail der entsprechende MIME-Type mitgegeben wird. Im folgenden wird als erstes in der Variablen \$mailtext die HTML-E-Mail erstellt. Diese kann auch auf dem Bildschirm ausgegeben werden und kommt dann wie eine normale HTML-Seite.

Die ersten 2 Zeilen beim erstellen des Headers sind dafür zuständig, dass der korrekte MIME-Type eingestellt wird und der richtige Zeichensatz. Einfach probieren. Der Rest ist identisch mit der "normalen" E-Mail im vorherigen Kapitel.

```
<?php
$mailtext = '<html>
<head>
    <title>HTML-E-Mail mit PHP erstellen</title>
</head>

<body>

<h1>HTML-E-Mail mit PHP erstellen</h1>

<p>Diese E-Mail wurde mit PHP und HTML erstellt</p>

<table border="1">
    <tr>
        <td>Beschreibung</td>
        <td>Anzahl Seiten</td>
    </tr>
</table>
```

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

```

</tr>
<tr>
  <td>PHP lernen mit PHP-Kurs.com</td>
  <td>über 100</td>
</tr>
</table>

<p>Die meisten HTML-Tags wie <b>fett</b>
und <i>kursiv</i> stehen zur
Verfügung</p>

</body>
</html>
';

$empfaenger = "du@testkarnickel.de"; //Mailadresse
$absender   = "ich@testkarnickel.de";
$betreff    = "Mail-Test - HTML-E-Mail mit PHP erstellen";
$antwortan  = "ICH@testkarnickel.de";

$header     = "MIME-Version: 1.0\r\n";
$header .= "Content-type: text/html; charset=iso-8859-1\r\n";

$header .= "From: $absender\r\n";
$header .= "Reply-To: $antwortan\r\n";
// $header .= "Cc: $cc\r\n"; // falls an CC gesendet werden soll
$header .= "X-Mailer: PHP ". phpversion();

mail( $empfaenger,
      $betreff,
      $mailtext,
      $header);

echo "Mail wurde gesendet!";

?>

```

E-Mail werden nur Online versendet - probieren Sie dies unter einer Entwicklungsumgebung wie XAMPP, wird keine E-Mail losgesendet, obwohl alles korrekt ist. Das liegt dann daran, dass unter XAMPP keine E-Mail-Server etc. eingetragen sind.

Noch eine kleine Anmerkung: HTML-E-Mails sind nicht besonders gern gesehen, wenn es auch mit Text-E-Mails gehen würde. Warum? In HTML-E-Mails kann jede Menge Quatsch transportiert werden, diese sind größer als Notwendig, es kann im Hintergrund je nach Einstellungen des E-Mail-Programmes mit ein paar Kniffen kontrolliert werden, wann der Empfänger die erhaltene E-Mail angesehen hat (und auch ob er diese öfters angesehen hat). Es sollte also einen guten Grund geben, warum Sie HTML-E-Mail anstatt normaler Text-E-Mails einsetzen (und bitte nicht mit dem Grund "es sieht schöner aus").

Übung Newsletteranmeldung

In dieser Übung werden alle Kenntnisse aus dem Einsteiger-Kapitel verwendet. Es wird eine Anmeldung für einen Newsletter programmiert - der zur Sicherheit vor Missbrauch über ein double-opt-in Verfahren verfügt. Der Anmelder gibt im ersten Schritt seine E-Mail-Adresse an, bekommt dann im 2. Schritt an diese E-Mail-Adresse eine Nachricht per E-Mail mit Link

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

gesendet und verifiziert sich im dritten Schritt durch Anklicken des dort enthaltenen Links. Dieses Verfahren wird double-opt-in (zweifaches Bestätigen) genannt - somit ist die Existenz der angegebenen E-Mail-Adresse sichergestellt und zusätzlich, dass der Besitzer den Newsletter überhaupt beziehen möchte.

Zur einfacheren Umsetzung ist die Übung in 4 Bereiche gegliedert:

1. Formularerstellung,
2. Kontrolle und Speicherung der eingegebenen Daten,
3. E-Mail erstellen und versenden,
4. Kontrolle des letzten Anmeldeschritts (Link in E-Mail Anklicken durch Bezieher).

Erstellen des Formulars

Erstellen Sie ein Formular (siehe Kapitel "Formulare") mit folgenden Feldern:

Ihre E-Mail-Adresse:

Anrede: (optional)

Frau Mann

Vorname: (optional)

Nachname: (optional)

hiermit bin ich einverstanden, dass meine Daten elektronisch gespeichert werden, damit mir die gewünschte Newsletter zugestellt werden kann ... (gesetzliches Blahblah zum Datenschutz und Speicherung der Daten)

Die PHP-Datei erhält den Namen: newsletter-bestellen.php

Kontrolle und Speicherung der eingegebenen Daten

Hier erfolgt eine Kontrolle, ob das Feld "E-Mail-Adresse" ausgefüllt wurde. Wurde es nicht ausgefüllt, wird das Formular wieder angezeigt (mit allen bisher eingetragenen Daten). Wurde es ausgefüllt, werden die eingegebenen Daten in der "Datei gespeichert" mit dem Namen "nl-anwaerter.txt". Je Datensatz wird eine Zeile verwendet. Zur Trennung der einzelnen Felder wird der Querstrich | verwendet (zu finden auf der Taste mit <> - Zeichen erhält man, wenn man gleichzeitig die Taste ALT-GR (rechte Alt-Taste) drückt.

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Zusätzlich wird noch eine Kontrollzahl abgespeichert, die sich aus Uhrzeit und Datum zusammensetzt: (alles 2-stellig und ohne Zwischenraum und Trennung)
SekundeJahrStundeTagMinuteMonat

In der Datei werden also die Einträge in folgender Form zu finden sein (alle Einträge rein fiktiv)

```
donald@duck.ent|Herr|Donald|Duck|280710282705  
bart@simp.son|Herr|Big b.||120723121806
```

E-Mail erstellen und versenden

An die angegebene E-Mail-Adresse wird eine E-Mail mit dem Inhalt:

Sehr geehrter [und nun die richtige Anrede und Vor- und Nachnamen wenn angegeben],

Sie haben den Newsletter von ... bestellt. Um sicherzustellen, dass die E-Mail-Adresse funktioniert und Sie den Newsletter erhalten möchten, klicken Sie bitte auf folgenden Link:

```
<a href="http://localhost/newsletter-freischalten.php?mail=...&id=280710282705">Newsletter  
freischalten</a>
```

Sollten Sie den Newsletter nicht angefordert haben, entschuldigen Sie bitte diese E-Mail. Dann hat sich wahrscheinlich jemand vertippt. Ignorieren Sie einfach die Mail und löschen Sie diese.

Informationen zum Versenden von E-Mail gab es im Kapitel: "E-Mail mit PHP erstellen"

Kontrolle des letzten Anmeldeschritts (Link in E-Mail anklicken durch Bezieher)

Wenn auf den Link der E-Mail geklickt wird, muss die Information validiert werden. Laden Sie dazu die Datei "nl-anwaerter.txt" und gehen Sie diese durch. Existiert ein Eintrag, in dem die E-Mail-Adresse und die Kontrollnummer passt, schreiben Sie diese Zeile in die Datei "newsletterempfaenger.txt".

Anmerkung zur Übung

Zur Sicherheit sollten im Regelbetrieb die Dateinamen "newsletterempfaenger.txt" und "nl-anwaerter.txt" anders benannt werden. Am besten ist ein Speicherort auf dem Webserver, auf den nicht durch Internetbesucher zugegriffen werden kann!

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Beispiel aus der Praxis

Schöne Beispiele aus der Praxis finden Sie auf zahlreichen Websites. Spannend ist immer, inwieweit die Newsletter-Bestellprozeduren deutscher Gesetzgebung (bzw. in welchem Land man es macht) entsprechen. Dabei sollte man immer ein Double-Opt-In-Verfahren nutzen. Auch beim Erfassen sollte man Vorsicht walten lassen und nicht unnötige Daten sammeln - sonst kommt man u.U. schnell mit dem deutschen Datenschutzgesetz in Konflikt.

Script Newsletteranmeldung

Erster Programmteil für die Newsletter-Bestellung

```
<?php
define ("ZEILENUMBRUCH", "\r\n");
// 1. Schritt: Formularerstellung,
// 2. Schritt: Kontrolle und
// 2.5 Speicherung der eingegebenen Daten
// 3. Schritt: E-Mail erstellen und versenden,
// 4. Schritt: Kontrolle des letzten Anmeldeschritte
//      (Link in E-Mail Anklicken durch Bezieher).

// Zur Kontrolle von übergebenen Variablen
// echo "<pre>";
// print_r ($_GET);

// Schritt 2: Kontrolle, ob Formular ausgefüllt wurde
if ( $_GET['aktion']      <> 'form_ausg' OR
     $_GET['emailadresse'] == "" OR
     $_GET['einverstanden'] <> "v"
    )
{
    // Schritt 1: Formular anzeigen
    formular_erstellen ( $_GET['emailadresse'],
                        $_GET['gender'],
                        $_GET['vorname'],
                        $_GET['nachname'],
                        $_GET['einverstanden'] );
}
else
{
    echo "<h1>Bestätigungs-Mail erstellt</h1>";

    // erstellen der Kontrollzahl
    $kontrollzahl = date("syhdim");

    // Schritt 2.5: Speichern der eingegebenen Daten
    speichern_datei ( $_GET['emailadresse'],
                    $_GET['gender'],
                    $_GET['vorname'],
                    $_GET['nachname'],
                    $kontrollzahl );

    // E-Mail senden mit Funktion
    mail_zur_kontrolle ( $_GET['emailadresse'],
                        $_GET['gender'],
```

```

        $_GET['vorname'],
        $_GET['nachname'],
        $kontrollzahl );
    }

function speichern_datei ( $emailadresse = "",
                           $gender="",
                           $vorname="",
                           $nachname="",
                           $kontrollzahl="" )
{
    $handle = fopen ( "nl-anwaerter.txt", "a" );

    // schreiben des Inhaltes von emailadresse
    fwrite ( $handle, $emailadresse );

    // Trennzeichen einfügen, damit Auswertung möglich wird
    fwrite ( $handle, "|" );

    // schreiben des Inhaltes von gender
    fwrite ( $handle, $gender );
    fwrite ( $handle, "|" );

    // schreiben des Inhaltes von vorname
    fwrite ( $handle, $vorname );
    fwrite ( $handle, "|" );

    // schreiben des Inhaltes von nachname
    fwrite ( $handle, $nachname );
    fwrite ( $handle, "|" );

    // schreiben des Inhaltes der Kontrollzahl
    fwrite ( $handle, $kontrollzahl );

    fwrite ( $handle, "\r\n" );

    // Datei schließen
    fclose ( $handle );
}

function mail_zur_kontrolle ( $email, $gender, $vorname, $nachname,
                              $kontrollzahl )
{
    if ( $gender == "w" )
    {
        $mailtext = "Sehr geehrte Frau $vorname $nachname,";
    }
    elseif ( $gender == "m" )
    {
        $mailtext = "Sehr geehrte Herr $vorname $nachname,";
    }
    else
    {
        $mailtext = "Hallo $vorname $nachname,";
    }
    $mailtext .= ZEILENUMBRUCH;
    $mailtext .= ZEILENUMBRUCH;
}

```

```

$mailtext .= "Sie haben den Newsletter von ... bestellt. ";
$mailtext .= "Um sicherzustellen, dass die E-Mail-Adresse ";
$mailtext .= "funktioniert und Sie den Newsletter erhalten ";
$mailtext .= "möchten, klicken Sie bitte auf folgenden Link:";
$mailtext .= ZEILENUMBRUCH;
$mailtext .= ZEILENUMBRUCH;
$mailtext .= 'http://localhost/newsletter-freischalten.php';
$mailtext .= '?mail=';
$mailtext .= $email;
$mailtext .= '&id=';
$mailtext .= $kontrollzahl;
$mailtext .= ZEILENUMBRUCH;
$mailtext .= ZEILENUMBRUCH;
$mailtext .= "Wenn Sie den Newsletter nicht angefordert haben,";
$mailtext .= "entschuldigen Sie bitte diese E-Mail. ";
$mailtext .= "Dann hat sich wahrscheinlich jemand vertippt. ";
$mailtext .= "Ignorieren Sie einfach die Mail ";
$mailtext .= "und löschen Sie diese.";

// Nur zur Kontrolle beim Programmieren, ob E-Mail-Text
// sauber zusammengestellt wird
echo '<textarea name="" rows="10" cols="80">';
echo $mailtext;
echo "</textarea>";

mail ($email,
     "Kontrolle E-Mail-Adresse Newsletter XYZ",
     $mailtext,
     "From: axel@example.com\nReply-To: axel@example.com"
    );
}

function formular_erstellen ($emailadresse = "",
                             $gender="",
                             $vorname="",
                             $nachname="",
                             $einverstanden="" )
{
    echo '<form name="" action=""';
    echo $ _SERVER['PHP_SELF'];
    echo '" method="GET" enctype="text/html">';

    echo '<p>';
    echo 'Ihre E-Mail-Adresse:<br />';
    echo '<input type="Text" name="emailadresse" value=""';
    // Inhalt, falls das Formular bereits unvollständig
    // ausgefüllt wurde (also zweiter Aufruf)
    echo $emailadresse;
    echo '" size="50">';
    echo '</p>';

    echo '<p>';
    echo 'Anrede: (optional)<br>';
    echo '<input type="Radio" name="gender" value="w" ';
    // Inhalt, falls das Formular bereits unvollständig
    // ausgefüllt wurde (also zweiter Aufruf)
    if ( $gender == "w" )
    {
        echo 'checked="checked" ';
    }
}

```

```

}
echo '</>';
echo 'Frau    ';

echo '<input type="Radio" name="gender" value="m" ';
// Inhalt, falls das Formular bereits unvollständig
// ausgefüllt wurde (also zweiter Aufruf)
if ( $gender == "m")
{
    echo 'checked="checked" ';
}
echo '</>';
echo 'Mann';
echo '</p>';

echo '<p>';
echo 'Vorname: (optional)<br />';
echo '<input type="Text" name="vorname" value="';
// Inhalt, falls das Formular bereits unvollständig
// ausgefüllt wurde (also zweiter Aufruf)
echo $vorname;
echo '" size="" />';
echo '</p>';

echo '<p>';
echo 'Nachname: (optional)<br />';
echo '<input type="Text" name="nachname" value="';
// Inhalt, falls das Formular bereits unvollständig
// ausgefüllt wurde (also zweiter Aufruf)
echo $nachname;
echo '" size="" />';
echo '</p>';

echo '<p>';
echo '<input type="Checkbox" name="einverstanden" value="v"> ';
echo 'hiermit bin ich einverstanden, dass meine Daten ';
echo 'elektronisch gespeichert werden, damit mir die';
echo 'gewünschte Newsletter zugestellt werden kann ...';
echo '(gesetzliches Blahblah zum Datenschutz und Speicherung';
echo 'der Daten ....)';
echo '</p>';

echo '<input type="hidden" name="aktion" value="form_ausg" />';

echo '<p>';
echo '<input type="Submit" name="speichern" value="speichern">';
echo '</p>';
echo '</form>';
}
?>

```

Der Programmteil **newsletter-freischalten.php** zum überprüfen der Rückmeldung

```

<?php
// newsletter-freischalten.php

// Übergebene Daten anzeigen zur Kontrolle beim Programmieren
// später ausblenden !
echo "<pre>";
print_r ( $_GET );

```

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

```

// Datei öffnen zum Lesen
$handle = fopen ("nl-anwaerter.txt", "r");

while ( $inhalt = fgets ($handle, 4096 ))
{
    $inhalt = trim ( $inhalt );
    echo "<li> |". $inhalt ."| </li>";

    list($email, $gender, $vorname, $nachname, $kontrollzahl) =
        split("\|", $inhalt, 5);

    echo "<li>$email, $gender, $vorname, $nachname, $kontrollzahl</li>";

    if ( $email == $_GET['mail'] AND $kontrollzahl == $_GET['id'])
    {
        echo "<h1>Sie wurden freigeschaltet</h1>";
        speichern ($email, $gender, $vorname, $nachname, $kontrollzahl);
    }
}

fclose($handle);

function speichern ($email, $gender, $vorname, $nachname, $kontrollzahl)
{
    $handle = fopen ( "nl-bestaetigt.txt", "a" );

    // schreiben des Inhaltes von emailadresse
    fwrite ( $handle, $email );

    // Trennzeichen einfügen, damit Auswertung möglich wird
    fwrite ( $handle, "|" );

    // schreiben des Inhaltes von gender
    fwrite ( $handle, $gender );
    fwrite ( $handle, "|" );

    // schreiben des Inhaltes von vorname
    fwrite ( $handle, $vorname );
    fwrite ( $handle, "|" );

    // schreiben des Inhaltes von vorname
    fwrite ( $handle, $nachname );
    fwrite ( $handle, "|" );

    // schreiben des Inhaltes der Kontrollzahl
    fwrite ( $handle, $kontrollzahl );

    fwrite ( $handle, "\r\n" );

    // Datei schließen
    fclose ( $handle );
}
?>

```

Arrays

Arrays sind Variablen mit mehr Möglichkeiten. In Arrays können viele Daten hinterlegt, ausgegeben, durchsucht und sortiert werden.

Bedeutung den englischen Begriffs Array: "Anordnung", "Aufstellung", "Bereich", "Feld", "Reihe", "Reihung". Ein Array ist also eine Reihe von gleichen Elementen in einem festgelegten Aufbau, z.B. alle Vornamen, oder auch alle Vornamen und alle Nachnamen.

Arrays sind für den Programmierer Pflicht - daher ist das Verständnis von Arrays wichtig und spart viel Zeit bei Programmaufgaben.

Um ein Array zu erstellen, werden hinter dem Namen eckigen Klammern geschrieben.

Beispiel als Variable:

```
$operationsystem = "Windows";
```

Beispiel als Array:

```
$betriebssystem[] = "Windows";
```

Beispiel der Schreibweise

als Variable:	<code>\$operationsystem = "Windows";</code>
als Array:	<code>\$betriebssystem[] = "Windows";</code>

Ausgabe der Variable und des Arrays:

```
// Definition der Variable und Wertzuweisung
$operationsystem = "Windows";

// Definition des Arrays und Wertzuweisung
$betriebssystem[] = "Windows";

echo '<p>Inhalt Variable $operationsystem: <br />';
print_r ( $operationsystem );

echo '<p>Inhalt Array $betriebssystem: <br />';
print_r ( $betriebssystem );
```

Dies ergibt als Ergebnis folgendes Bild:

```
Inhalt Variable $operationsystem:
Windows

Inhalt Array $betriebssystem:
Array ( [0] => Windows )
```

Jetzt kommt natürlich die Frage, was das soll? Spätestens wenn viele Daten in einem Array hinterlegt werden, punktet es:

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

```

$betriebssystem[] = "Windows";
$betriebssystem[] = "Linux";
$betriebssystem[] = "Solaris";
$betriebssystem[] = "Mac";

echo "<pre>";
print_r ( $betriebssystem );

```

Das führt zu folgender Ausgabe

```

Array
(
    [0] => Windows
    [1] => Linux
    [2] => Solaris
    [3] => Mac
)

```

direktes Ansprechen eines Arrays per Index

Die Inhalte des Arrays können über die automatisch zugewiesenen Nummern (Index) angesprochen werden (fängt immer mit 0 an). Diese zugewiesene Nummer nennt sich der Index. So hat im obigen Beispiel des Array mit dem Index 1 den Inhalt "Linux"

```

$betriebssystem[0] = "Windows";
$betriebssystem[1] = "Linux";
$betriebssystem[2] = "Solaris";
$betriebssystem[3] = "Mac";

echo "ich nutze heute mal: ";
echo $betriebssystem[1];

```

Anstatt der automatischen Zuweisung der Indexnummer kann auch bewusst eine eigene Index-Nummer gesetzt werden:

```

$betriebssystem[21] = "DOS";
$betriebssystem[17] = "OS/2";

```

Andere Zuweisungsart von Arrays

Andere Schreibweise, um Arrays zuzuordnen:

```

$betriebssystem = array ("Windows", "Linux", "Solaris", "Mac" );

```

Assoziatives Array

Die Nummern als Indizes sind kein Muss - es ist auch möglich, die Indizes zu benennen. Dies nennt sich dann ein "Assoziatives Array".

Beispielweise können so Adressen übersichtlich verwaltet werden:

```

$adresse['vorname'] = "Axel";

```

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

```
$adresse['nachname'] = "Pratzner";
$adresse['hausnummer'] = 39;
$adresse['ort'] = "Tübingen";

echo "<pre>";
print_r ( $adresse );
```

Aufgabe zu Assoziatives Array:

Erstellen Sie ein "Assoziatives Array" mit 5 Einträgen - als Index die Autokennzeichenkürzel und als Inhalt die Stadt ausgeschrieben. Lassen Sie dieses Array am Bildschirm einmal komplett anzeigen und einmal nur für den Index "Tü".

Folgende Daten sollen verwendet werden:

B - Berlin
S - Stuttgart
TÜ - Tübingen
M - München
MZ - Mainz

Array-Elemente zählen

Anzahl der Elemente eines Array zählen - count

Sie erhalten die Anzahl der Elemente eines Arrays über die Funktion

`count (arrayname)` - als Rückgabewert erhalten Sie die Anzahl der Elemente als Integer.

```
$betriebssystem[] = "Windows";
$betriebssystem[] = "Linux";
$betriebssystem[] = "Solaris";

echo "<p>Im Array befinden sich Einträge: ";
echo count ($betriebssystem);

$betriebssystem[] = "Mac";
$betriebssystem[] = "DOS";

echo "<p>Im Array befinden sich Einträge: ";
echo count ($betriebssystem);
```

Als Ausgabe am Bildschirm kommt:

```
Im Array befinden sich Einträge: 3

Im Array befinden sich Einträge: 5
```

Inhalt ausgeben

Um wie gewohnt mit echo den Inhalt eines bestimmten Arrays ausgeben zu können, muss dieses Array direkt angesprochen werden. Sollen nun alle Array der Reihe nach ausgegeben werden, wird die Anzahl der Elemente ermittelt und dann dieses mit einer for-Schleife durchlaufen.

```
$betriebssystem[] = "Windows";
$betriebssystem[] = "Linux";
$betriebssystem[] = "Solaris";

$anzahl = count ( $betriebssystem );
echo "Es gibt $anzahl Einträge:";
echo "<ul>";

for ($x = 0; $x < $anzahl; $x++)
{
    echo "<li>Eintrag von $x ist $betriebssystem[$x] </li>";
}

echo "</ul>";
```

Als Ausgabe erhalten Sie auf dem Bildschirm:

```
Es gibt 3 Einträge:

• Eintrag von 0 ist Windows

• Eintrag von 1 ist Linux

• Eintrag von 2 ist Solaris
```

Eine weitere verbreitete Art für die Ausgabe von Arrays ist über foreach. Dabei wird in der foreach-Schleife der Inhalt des aktuellen Array-Bereichs einer Variable zugewiesen. Im folgenden Beispiel die Variable \$os, was die Abkürzung für "operation system" ist (engl. für Betriebssystem).

```
$betriebssystem[] = "Windows";
$betriebssystem[] = "Linux";
$betriebssystem[] = "Solaris";

echo "<ul>";

foreach ($betriebssystem as $os)
{
    echo "<li>Betriebssystem ist $os </li>";
}

echo "</ul>";
```

Zusätzlich kann der Index über foreach ausgegeben werden. Das obige Beispiel wird ergänzt um die Variable \$nr und der Schreibweise `$nr =>`. Diese Schreibweise ist einfach so :)

```
$betriebssystem[] = "Windows";
$betriebssystem[] = "Linux";
```

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

```

$betriebssystem[] = "Solaris";

echo "<ul>";

foreach ($betriebssystem as $nr => $os)
{
    echo "<li>Betriebssystem mit Index-Nr. $nr ist $os </li>";
}

echo "</ul>";

```

Sortieren von Arrays

Es gibt einige Funktionen bei PHP, um Arrays zu sortieren:

sort

(engl. sortieren) sortiert ein Array nach dem Wert aufsteigend

rsort

(engl. reverse sort - umgekehrte Sortierung) - sortiert ein Array nach dem Wert absteigend

```

$betriebssystem[] = "Windows";
$betriebssystem[] = "Linux";
$betriebssystem[] = "Solaris";

sort ( $betriebssystem );

$anzahl = count ( $betriebssystem );
echo "Es gibt $anzahl Einträge:";
echo "<ul>";

for ($x = 0; $x < $anzahl; $x++)
{
    echo "<li>Eintrag von $x ist $betriebssystem[$x] </li>";
}
echo "<pre>";
print_r ( $betriebssystem );

```

Der Befehl `print_r` bewirkt eine komplette Ausgabe des Inhalts einer Variable, bzw. eines Arrays. Es ist eine gute Idee, davor durch ein `<pre>` eine preformatierte (Ausgabe aller Leerzeichen und Umbrüche) zu erzwingen.

Beachten Sie die Indizes. Bei `sort` und `rsort` werden die ursprünglichen Indizes aufgehoben.

Aufgabe: sort und Indizes

Zeigen Sie, dass sich die Indizes durch die Nutzung von `Sort` ändern.

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Sortierung von Arrays mit Beibehalten der Indizes

Im Gegensatz zu `sort` bleibt die Zuordnung des Index erhalten. D.h. wenn z. B. vor dem Sortieren ein Eintrag die Index-Nummer 2 hat, hat dieser auch nach dem Sortieren noch dieselbe Index-Nummer.

asort

sortiert ein Array nach dem Wert aufsteigend, Indexverbindungen werden beibehalten

arsort

wie `asort`, nur absteigend

```
$betriebssystem[] = "Windows";
$betriebssystem[] = "Linux";
$betriebssystem[] = "Solaris";

asort ( $betriebssystem );

$anzahl = count ( $betriebssystem );
echo "Es gibt $anzahl Einträge:";
echo "<ul>";

for ($x = 0; $x < $anzahl; $x++)
{
    echo "<li>Eintrag von $x ist $betriebssystem[$x] </li>";
}

echo "<pre>";
print_r ( $betriebssystem );
```

sortiert das Array nach dem Index

ksort

sortiert das Array nach dem Index

krsort

wie `ksort`, nur absteigend

Dies macht natürlich bei "Assoziativem Array" oft mehr Sinn.

```
$betriebssystem[4] = "Windows";
$betriebssystem[9] = "Linux";
$betriebssystem[1] = "Solaris";

ksort ( $betriebssystem );

$anzahl = count ( $betriebssystem );
echo "Es gibt $anzahl Einträge:";
echo "<ul>";

for ($x = 0; $x < $anzahl; $x++)
{
    echo "<li>Eintrag von $x ist $betriebssystem[$x] </li>";
}
```

```
echo "<pre>";
print_r ( $betriebssystem );
```

Aufgabe

Warum klappt bei dem letzten Beispiel die Aufzählung der Betriebssysteme nicht? Bei der Ausgabe kommen 2 mal leere Werte!

Korrigieren Sie das Beispiel.

Aufgabe:

Lassen Sie die Autokennzeichenkürzel und die Städte für

- B - Berlin
- S - Stuttgart
- TÜ - Tübingen
- M - München
- MZ - Mainz

über ein assoziatives Array in folgenden Sortierungen ausgeben:

- sort
- rsort
- ksort

Doppelte Einträge im Array löschen

Schnell und einfach können alle doppelten Werte eines Arrays gelöscht werden. Dazu gibt es den Befehl: `$neues_array = array_unique ($altes_array);`

```
$betriebssystem[] = "Windows";
$betriebssystem[] = "Linux";
$betriebssystem[] = "Solaris";
$betriebssystem[] = "Linux";
$betriebssystem[] = "Solaris";

$bs = array_unique ( $betriebssystem );

echo "<pre>";
print_r ( $bs );
```

Anmerkung: Einen kleinen Fallstrick gibt es - es werden keine Einträge gelöscht, die einen unterschiedlichen Typ haben:

Beispiel:

```
$eintrag[] = "9";
$eintrag[] = 9;
```

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Die erste 9 wird als String angesehen und die zweite als Integer - daher sind beide Einträge trotz `array_unique` noch erhalten.

Mehrdimensionale Arrays

Hier können wir beliebig viele Dimensionen aufziehen (was aber ab der dritte Dimension ein wenig anstrengend zu denken ist)

Veranschaulichen wie es in einem Beispiel: Unsere Adresse von oben ist nur eine Adresse. Wenn ich nun mehrere Adressen speichern möchte, kann ich dies über mehrdimensionale Arrays erreichen. Dazu wird zum ersten Index einfach noch ein weiterer aufgenommen.

```
$adresse[0][vorname] = "Axel";
$adresse[0][nachname] = "Pratzner";

$adresse[1][vorname] = "Sabine";
$adresse[1][nachname] = "Musterfrau";

$adresse[2][vorname] = "Fritz";
$adresse[2][nachname] = "Müller";

echo "<pre>";
print_r ( $adresse );
```

Zum direkten Ansprechen müssen nun alle Indizes angegeben werden:

```
echo $adresse[2][vorname];
```

Aufgabe:

erstellen Sie ein mehrdimensionales Array, das folgende Daten enthält:

Land	Hauptstadt
Land	Fluss

für BRD, Schweiz, USA, Marroko

Sortieren von mehrdimensionalen Arrays - `array_multisort`

Diese Funktion ist anspruchsvoll, aber machbar. Präzise Arbeitsweise der Funktion [array_multisort bei php.net](#).

Erster Schritt ist, dass man ein mehrdimensionales Array hat - hier Termine von "funky Bands".

```
$termin[] = array('Datum' => 20081208, 'Ort' => "Wangen", 'Band'=>'cOoL RoCk oPaS');
$termin[] = array('Datum' => 20070311, 'Ort' => "Stuttgart", 'Band'=>'Die Hosenbodenband');
$termin[] = array('Datum' => 20070628, 'Ort' => "Tübingen", 'Band'=>'flying socks');
```

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

```
$termin[] = array('Datum' => 20070628, 'Ort' => "Stuttgart", 'Band'=>'flying socks');
```

Im nächsten Schritt wird nur für die Sortierung das mehrdimensionale Array aufgeteilt.

```
foreach ($termin as $nr => $inhalt)
{
    $band[$nr] = $inhalt['Band'] ;
    $ort[$nr] = $inhalt['Ort'] ;
    $datum[$nr] = $inhalt['Datum'] ;
}
```

Und dann kann das eigentliche Sortieren durchgeführt werden:

```
// sortieren nach Datum
array_multisort($datum, SORT_ASC, $termin);

// sortieren nach Datum absteigend
array_multisort($datum, SORT_DESC, $termin);

// Sortierung nach Ort aufsteigend
array_multisort($ort, SORT_ASC, $termin);

// Sortierung nach Band aufsteigend
array_multisort($band, SORT_ASC, $termin);

// sortieren nach Datum und Ort
array_multisort($datum, SORT_ASC, $ort, SORT_ASC, $termin);
```

Tipp am Rande: Um nicht ein Chaos beim Sortieren zu bekommen, da nach Groß- und Kleinschreibung sortiert wird (ein großes D kommt vor dem kleine a :) - einfach im zweiten Schritt alles klein machen:

```
foreach ($termin as $nr => $inhalt)
{
    $band[$nr] = strtolower( $inhalt['Band'] );
    $ort[$nr] = strtolower( $inhalt['Ort'] );
    $datum[$nr] = strtolower( $inhalt['Datum'] );
}
```

Und nun die Einzelteile am Stück mit Sortierung auf Datum absteigend

```
$termin[] = array('Datum' => 20081208, 'Ort' => "Wangen", 'Band'=>'cOoL RoCk oPaS');
$termin[] = array('Datum' => 20070311, 'Ort' => "Stuttgart", 'Band'=>'Die Hosenbodenband');
$termin[] = array('Datum' => 20070628, 'Ort' => "Tübingen", 'Band'=>'flying socks');
$termin[] = array('Datum' => 20070628, 'Ort' => "Stuttgart", 'Band'=>'flying socks');

print_r ( $termin );

foreach ($termin as $nr => $inhalt)
{
    $band[$nr] = strtolower( $inhalt['Band'] );
    $ort[$nr] = strtolower( $inhalt['Ort'] );
    $datum[$nr] = strtolower( $inhalt['Datum'] );
}
```

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

```
}  
  
array_multisort($datum, SORT_DESC, $stermin);  
  
print_r ( $stermin );
```

Nützliche Funktionen für Arrays

Es gibt eine Reihe nützlicher Funktionen, um effektiv mit Arrays arbeiten zu können und sich selber viel Zeit zu sparen.

array_count_values

zählt die Werte eines Array

array_diff

vergleicht Arrays und gibt Unterschiede zurück (in neues Array)

in_array

Überprüft, ob ein Wert in einem Array vorhanden ist

min

gibt das Element mit dem niedrigsten Wert zurück

max

gibt das Element mit dem höchsten Wert zurück

array_merge

Verbinden zweier Arrays

range

Erstellt ein Array mit Unter- und Obergrenze (z. B. Lottozahlen 1 bis 36)

shuffle

Mischen der Werte eines Arrays - so kann z.B. für Lottozahlen eine Zufallsauswahl erzeugt werden. Mischen und dann die ersten 6 Ausgaben

Aufgabe: Programm, das Lottozahlen vorschlägt

Erstellen Sie ein Programm, das Lottozahlen vorschlägt. Das Programm soll aus den Zahlen 1 bis 49 auswählen und anzeigen. Als Anzeige soll erscheinen:

- Zahl 1 ist 15
- Zahl 2 ist 3
- Zahl 3 ist 6
- Zahl 4 ist 11
- Zahl 5 ist 36
- Zahl 6 ist 32

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Die vorgeschlagenen Zahlen sollen bei jedem Aufruf variieren!

Beispiel Bildergalerie erstellen

Im Folgenden wollen wir eine automatische Bildergalerie erstellen. Dazu benötigen wir als Vorkenntnisse das Kapitel über “Verzeichnis einlesen” und das vorhergehende Kapitel über Arrays.

Im ersten Schritt lesen wir ein Verzeichnis aus und speichern alle gefundenen Dateinamen der Bilder in einem Array ab.

```
$verzeichnis = ".";

if (is_dir($verzeichnis))
{
    if ( $handle = opendir($verzeichnis) )
    {
        while (($file = readdir($handle)) !== false)
        {
            echo "<li>filename: $file ";
            echo "<li>filetype: ". filetype( $file) . "<hr>\n";
            // Test, ob ein Bilder (Endung .jpg) vorliegt
            if ( filetype( $file) == "file"
                AND substr( $file, -4) == ".jpg")
            {
                // Dateiname wird im Array bilderdateinamen gespeichert
                $bilderdateinamen[] = $file;
            }
        }
        closedir($handle);
    }
}

// Ausgabe der Bilderdateinamen zur Kontrolle
echo "<pre>";
print_r ( $bilderdateinamen );
```

Im nächsten Schritt benötigen wir Bilder - nehmen Sie folgenden Bilder zum Testen (auf jeden Fall diese für den Aha-Effekt für die automatische Beschriftung ;) Folgende Bilder werden hier verkleinert dargestellt:

Die Bilder stammen von mir (und können deshalb auch zum Testen eingesetzt werden) und weitere Bilder zum Thema Urlaub in und um Las Vegas können auf <http://www.las-vegas-urlaub.com> angesehen werden.

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen



Die Bilder können durch Anklicken mit der rechten Maustaste auf der eigenen Festplatte gespeichert werden. Die müssen im selben Verzeichnis gespeichert werden, wo auch das Programm zur Bildergalerie gespeichert ist!

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

```

$verzeichnis = ".";

if (is_dir($verzeichnis))
{
    if ( $handle = opendir($verzeichnis) )
    {
        while (($file = readdir($handle)) !== false)
        {
            echo "<li>filename: $file ";
            echo "<li>filetype: ". filetype( $file) . "<hr>\n";
            // Test, ob ein Bilder (Endung .jpg) vorliegt
            if ( filetype( $file) == "file"
                AND substr( $file, -4) == ".jpg")
            {
                // Dateiname wird im Array bilderdateinamen gespeichert
                $bilderdateinamen[] = $file;
            }
        }
        closedir($handle);
    }
}

// Ausgabe der Bilderdateinamen zur Kontrolle
echo "<pre>";
print_r ( $bilderdateinamen );
echo "<hr />";

foreach ( $bilderdateinamen AS $dateiname )
{
    echo "Dateiname: $dateiname:<br />\n";

    $exif = exif_read_data($dateiname, ANY_TAG, true, true);

    // print_r ($exif);

    echo "<h2>". $exif['WINXP']['Title'] . "</h2>";
    echo "<img src=\"http://www.php-kurs.com/$dateiname\"";
    echo $exif['COMPUTED']['html'];
    echo " alt=\"";
    echo $exif['WINXP']['Title'];
    echo "' \> ";

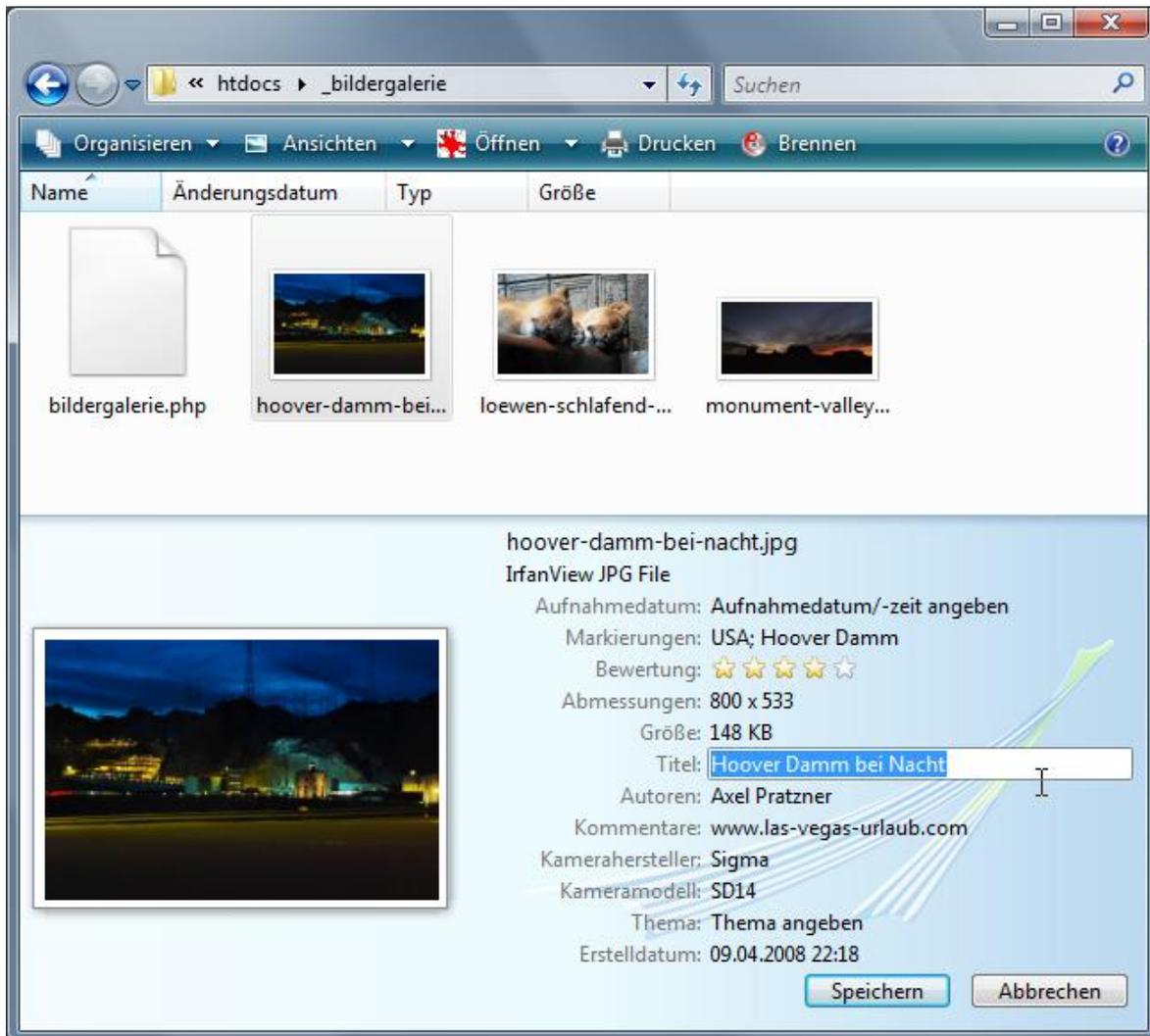
    echo "<hr />";
}

```

Bestimmt kommt gerade die Frage auch, woher um Gottes willen der Bildtitel kommt, der nichts mit dem Dateinamen zu tun hat? Der ist in der Bilddatei mit integriert. Dies ist einfach möglich durch IPTC und EXIF. Diese Daten lesen wird im unserem PHP-Programm zum Bildergalerie erstellen aus. Es geschieht über `exif_read_data` und die gewonnenen Daten wandern in das Array `$exif`. Lassen Sie sich einfach mal über `print_r` den Inhalt des Array `$exif` anzeigen!

Um Titel und Verschlagwortung in die Bilder zu integrieren, empfiehlt sich Windows Vista (wie im nächsten Bild zu sehen ist) oder auch verschiedene andere Programme können dazu hergenommen werden, wie beispielsweise das bekannte IrfanView.

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen



Funktionsbibliotheken erstellen und recyceln

Bisher haben wir einzelne Funktionen kennen gelernt. Nun wird es Funktionen geben, die immer wieder in verschiedenen Projekten benötigt werden könnten. Aus diesen können Funktionsbibliotheken erstellt werden, die sehr einfach in dem entsprechenden Programmteil, in der sie benötigt werden, einfach dazu eingebunden werden.

Das Einbinden geschieht über `include` bzw. `require`.

Unterschied zwischen `require` und `include` ist, dass falls die einzubindende Datei nicht existieren sollte, das Programm bei `require` dann an dieser Stelle abbricht. Dies kann aus sicherheitstechnischen Gründen vorteilhaft sein.

Über `include` kann z.B. auch HTML-Bereich eingebunden werden und so das Design einer Datei recht einfach auf jeder Seite übernommen werden (Beispielsweise ein `include` für den Kopf der Seite, eines für die Steuerung und eins für den Fuß).

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Funktionsbibliotheken sollten unbedingt die Endung .php haben, da ansonsten bei Problemen des Webservers die Besucher den Quellcode zu sehen bekommen und unter Umständen durch bestimmte Rückschlüsse dann an Daten kommen, die sie sonst nicht bekommen könnten und auch nicht bekommen sollten.

Die Datei, die per include aufgenommen wird, fängt genauso wie jede PHP-Datei mit <?php an und hört mit ?> auf.

Als kleines Beispiel wird im Folgendem die Funktionsbibliothek zur Ausgabe von Dateigrößen aufgenommen und dann genutzt. Erstellen Sie im ersten Schritt die Datei dateigroessen.php in Ihrem Verzeichnis, in dem Ihr PHP-Programme liegt (wenn Sie es ordentlicher haben möchten, erstellen Sie ein Unterverzeichnis z.B. mit dem Namen inc und packen es dort hinein - beim Inkludieren muss dann das Unterverzeichnis angegeben werden).

Der Inhalt der Datei "dateigroessen.php" ist:

```
<?php
// Funktionsbibliothek Konvertierungen

function dateigroesse_als( $bytes, $sto="" )
{
    switch($sto)
    {
        // Kilobit
        case 'kbit':
            $bytes = ($bytes * 8) / 1024;
            $bytes .= " kbit";
            break;

        // bit
        case 'bit' :
            $bytes = $bytes * 8;
            $bytes .= " bit";
            break;

        // Gigabyte
        case 'GB' :
            $bytes = $bytes / 1024 / 1024;
            $bytes .= " GB";

        // Megabyte
        case 'MB' :
            $bytes = $bytes / 1024;
            $bytes .= " MB";

        // Kilobyte
        case 'KB' :
            $bytes = $bytes / 1024;
            $bytes .= " KB";

        // byte
        default :
    }
    return $bytes;
}
?>
```

zum Einbinden der Datei geben Sie nur an:

```
require ('dateigroessen.php');
```

oder mit Unterverzeichnis:

```
require ('inc/dateigroessen.php');
```

Und nun als kleines Programm mit Namen "test-bibs.php":

```
require ('dateigroessen.php');  
echo dateigroesse_als( 156463, "KB" );
```

Übung Terminkalender

Im Folgenden wollen wir eine Terminanzeige für die eigene Homepages programmieren, bei der der Besucher zwischen verschiedenen Sortierungen wählen kann.

Den fertigen Terminkalender als Vorgeschmack darauf, was wir nun erstellen, finden Sie unter <http://www.iwke.de/kunden/php-kurs/terminkalender.php>. Dazu werden wir Schritt für Schritt vorgehen.

Die einzelnen Schritte:

- Erstellen eines mehrdimensionalen Arrays mit Termin, Ort, Bandname
- Kontrolle, ob bereits ein Sortierwunsch vorliegt; wenn nicht, sortieren nach Termin
- Anzeige als Tabelle
 - Tabellenkopf mit Links zur Auswahl der Sortierung
 - bei der Ausgabe der Zeilen wird jede zweite Zeile als Lesehilfe eingefärbt
 - Datum in deutscher Schreibweise ausgeben
 - Bandname ausgeben
 - Ort ausgeben

Erstellen eines mehrdimensionalen Arrays mit Termin, Ort, Bandname

Übernehmen Sie bitte folgende Einträge, da sonst die Übung unnötig schwierig wird.

```
$termin[] = array('Datum' => 20081208, 'Ort' => "Wangen", 'Band'=>'cOoL RoCk  
oPaS');  
$termin[] = array('Datum' => 20070311, 'Ort' => "Stuttgart", 'Band'=>'Die  
Hosenbodenband');  
$termin[] = array('Datum' => 20070628, 'Ort' => "Tübingen", 'Band'=>'flying  
socks');  
$termin[] = array('Datum' => 20070628, 'Ort' => "Stuttgart", 'Band'=>'flying  
socks');
```

Sie können weitere Auftritte hinzufügen.

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Für die Sortierung von mehrdimensionalen Arrays wird dieser wie folgt aufbereitet:

```
foreach ($termin as $nr => $inhalt)
{
    $band[$nr] = $inhalt['Band'] ;
    $ort[$nr]  = $inhalt['Ort']  ;
    $datum[$nr] = $inhalt['Datum'] ;
}
```

Kontrolle, ob bereits ein Sortierwunsch vorliegt; wenn nicht, sortieren nach Termin

Die Sortierung wird vom Benutzer dadurch gewählt, dass dieser einen Link anklickt. Im Link wird nach dem Dateiname "terminkalender.php" nach dem "?" der Variablenname "sortierung" und ein Kürzel übergeben.

Beispiel

```
<a href="terminkalender.php?sortierung=d">Datum sortieren</a>'
<a href="terminkalender.php?sortierung=b">Band sortieren</a>'
<a href="terminkalender.php?sortierung=o">Ort sortieren</a>'
```

Der Variablenname kann in PHP über `$_GET['sortierung']` abgefragt werden und in der Kontrollstruktur verwendet werden.

Als Kontrollstruktur nutzen wir hier switch - kontrolliert wird auf d (für Datum), b (für Band) und o (für Ort).

Die Sortierung erfolgt durch folgende Befehle: (siehe Kapitel "mehrdimensionale Arrays")

```
// Sortierung nach Datum aufsteigend
array_multisort($datum, SORT_ASC, $termin);

// Sortierung nach Band aufsteigend
array_multisort($band, SORT_ASC, $termin);

// Sortierung nach Ort aufsteigend
array_multisort($ort, SORT_DESC, $termin);
```

Nach dem Sortieren können Sie sich die Termine zur Kontrolle ausgeben lassen über:

```
print_r ( $termin );
```

Erstellen der Tabelle

Lassen Sie das Array `$termin` ausgeben (sinnvoll ist, eine eigene Funktion für die Tabellenausgabe zu erstellen).

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Innerhalb der Tabelle können Sie eine Variable hochzählen lassen - anhand dieser Zahl können Sie bestimmen, ob diese gerade oder ungerade ist und dann entsprechend die Zeile einfärben lassen.

Dazu bedienen wir uns der Mathematik und dem Modulo von 2 verwendet. Der Modulo ist eine mathematische Funktion, die als Ergebnis den Rest aus der Division zweier ganzer Zahlen angibt. Somit erhalten wir bei der Division durch 2 entweder 0 (somit haben wir eine gerade Zahl) oder nicht 0, somit haben wir eine ungerade Zahl.

```
if ( bcmmod ( $zeilenr , '2' ) == 0 )
{
    $hintergrundfarbe = ' bgcolor="#ACC8F0" ';
}
else
{
    $hintergrundfarbe = ' bgcolor="#DDE8F9" ';
}
```

Ausgabe des Datum in deutscher Form

Das Datum ist aus Gründen der Sortierbarkeit in der Form JahrMonatTag abgelegt - diesen String können Sie zerschneiden und wieder frisch zusammensetzen.

Dazu gibt es die PHP-Funktion `substr`.

Nach `substr` folgt die Variable, die behandelt werden soll, dann der Startpunkt und wieviele Stellen berücksichtigt werden.

Es ist auch möglich, von hinten zu zählen - siehe im folgenden Beispiel bei Tag (letzte 2 Stellen nehmen)

```
$jahr = substr ( $datum, 0, 4 );
$monat = substr ( $datum, 4, 2 );
$tag = substr ( $datum, -2 );
```

Üben, üben, üben

Diese Aufgabe ist nicht einfach, aber sehr lehrreich - bitte erst probieren, bevor Sie bei der Musterlösung nachsehen.

kompletter Quellcode zum Spickeln (falls man hängen bleibt und gar nichts weiter geht) gibt es auf der folgenden Seite.

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Übung Terminkalender

kompletter Quellcode zum Terminkalender

Ihr eigenes Ergebnis kann (wird i.d.R.) anders aussehen - wichtig, dass es funktioniert - sehen Sie diesen Vorschlag als Anregung und nicht zum frustrieren - wichtig ist beim lernen von Programmiersprachen, sich nicht entmutigen zu lassen!!

```
<?php
$termin[] = array('Datum' => 20081208, 'Ort' => "Wangen", 'Band'=>'cOoL RoCk
oPaS');
$termin[] = array('Datum' => 20070311, 'Ort' => "Stuttgart", 'Band'=>'Die
Hosenbodenband');
$termin[] = array('Datum' => 20070628, 'Ort' => "Tübingen", 'Band'=>'flying
socks');
$termin[] = array('Datum' => 20070628, 'Ort' => "Stuttgart", 'Band'=>'flying
socks');

// print_r ( $termin );

foreach ( $termin as $nr => $inhalt )
{
    $band[$nr] = strtolower( $inhalt['Band'] );
    $ort[$nr] = strtolower( $inhalt['Ort'] );
    $datum[$nr] = strtolower( $inhalt['Datum'] );
}

switch ( $_GET['sortierung'] )
{
    case ("d"):
        // Sortierung nach Datum und Ort aufsteigend
        array_multisort($datum, SORT_ASC, $ort, SORT_ASC, $termin);
        break;
    case ("o"):
        // Sortierung nach Ort aufsteigend
        array_multisort($ort, SORT_ASC, $termin);
        break;
    case ("b"):
        // Sortierung nach Band aufsteigend
        array_multisort($band, SORT_ASC, $termin);
        break;
    case ("da"):
        // Sortierung nach Datum und Ort aufsteigend
        array_multisort($datum, SORT_DESC, $termin);
        break;
    case ("oa"):
        // Sortierung nach Ort aufsteigend
        array_multisort($ort, SORT_DESC, $termin);
        break;
    case ("ba"):
        // Sortierung nach Band aufsteigend
        array_multisort($band, SORT_DESC, $termin);
        break;
    DEFAULT:
        // Sortierung nach Datum
        array_multisort($datum, SORT_ASC, $ort, SORT_ASC, $termin);
}
```

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

```

}

ausgabe_tabelle ( $termin );

function ausgabe_tabelle ( $termin )
{
    echo '<table border="1" cellpadding="6" cellspacing="0">';

    // Kopf fuer sortierung
    echo '<tr bgcolor="#6C9DE6">';
    echo '<th>';
    echo ' ';
    echo '</th>';

    echo '<th>';
    echo 'Datum ';
    echo '<a href="terminkalender.php?sortierung=d">↓</a>';
    echo ' ';
    echo '<a href="terminkalender.php?sortierung=da">↑</a>';
    echo '</th>';

    echo '<th>';
    echo 'Band ';
    echo '<a href="terminkalender.php?sortierung=b">↓</a>';
    echo ' ';
    echo '<a href="terminkalender.php?sortierung=ba">↑</a>';
    echo '</th>';

    echo '<th>';
    echo 'Ort ';
    echo '<a href="terminkalender.php?sortierung=o">↓</a>';
    echo ' ';
    echo '<a href="terminkalender.php?sortierung=oa">↑</a>';
    echo '</th>';
    echo '</tr>';

    foreach ( $termin AS $inhalt )
    {
        $zeilenr++;
        echo '<tr>';
        echo farbwechsel ( $zeilenr );
        echo '>';
        echo '<td>';
        echo $zeilenr. ".";
        echo '</td>';

        echo '<td>';
        echo datum_deutsch ( $inhalt['Datum'] );
        echo '</td>';

        echo '<td>';
        echo $inhalt['Band'];
        echo '</td>';

        echo '<td>';
        echo $inhalt['Ort'];
        echo '</td>';
        echo '</tr>';
    }
    echo '</table>';
}

```

```

function datum_deutsch ( $datum )
{
    $jahr = substr ( $datum, 0, 4 );
    $monat = substr ( $datum, 4, 2 );
    $tag = substr ( $datum, -2 );
    $datum_deutsch = $tag ".".$monat ".".$jahr;
    return ( $datum_deutsch );
}

function farbwechsel ( $zeilenr )
{
    if ( bcmath ( $zeilenr , '2' ) == 0 )
    {
        $hintergrundfarbe = ' bgcolor="#ACC8F0" ';
    }
    else
    {
        $hintergrundfarbe = ' bgcolor="#DDE8F9" ';
    }
    return ( $hintergrundfarbe );
}
?>

```

SESSION - Kurzzeit-Gedächtnis für PHP

Über die Sessions haben wir eine einfache Möglichkeit, uns Informationen, Daten und Zustände während einer kompletten Nutzungsdauer eines Besuchers zu merken. Diese werden in sogenannten Sessions gespeichert, auf die von PHP unabhängig, auf welchem Programmteil diese "gemerkt" wurde, wieder darauf zugegriffen werden kann (und auch geändert).

Anhand von einem einfachen Beispiel sehen wir die Arbeitsweise. Wir vermerken, wie oft derselbe Besucher eine bestimmte Seite besucht hat, und lassen diese Zahl auf einer anderen Seite ausgeben.

```

<?php
session_start(); // Beginn der Session, bereits vorhandene werden dadurch
aktiviert

// Abfrage, ob die Variable anzahlbesuche existiert
if ( ! isset ( $_SESSION['anzahlbesuche'] ) )
{
    $_SESSION['anzahlbesuche'] = 1;
}
else
{
    $_SESSION['anzahlbesuche'] ++;
}
?>
<a href="ausgabe_seitenanzeigen.php">Ausgabe Seitenbesuche</a>

```

Durch `session_start()`; wird die Sitzung (session) gestartet und eventuell vorhandene Session-Variablen werden verfügbar gemacht.

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Die Zeile mit `isset` überprüft, ob die Session-Variable `anzahlbesuche` einen Inhalt enthält.

Wenn bisher kein Inhalt vorhanden ist, wird der Session-Variable `anzahlbesuche` die Zahl 1 zugewiesen.

Ist die Variable bereits vorhanden, wird der Inhalt um 1 erhöht.

Es wird aus didaktischen Gründen noch ein Link ausgegeben, um eine weitere Seite aufzurufen, um dort den Inhalt der Variable auszugeben. Der Inhalt der weiteren Seite ist:

```
<?php
session_start(); // Beginn der Session, bereits vorhandene werden dadurch
aktiviert
echo $_SESSION['anzahlbesuche'];
?>
```

Sie sehen, wie einfach es ist, auf weiteren Seiten auf die Inhalte der Session-Variablen zuzugreifen, diese auszugeben und auch Berechnungen etc. durchzuführen.

Lebenslauf einer Session-Variablen

Die Session-Variable wird automatisch zerstört, wenn der Browser geschlossen wird oder eine gewisse Zeit vergangen ist (meistens 180 Minuten). Es gibt auch die Möglichkeit, dass Sessions in Cookies gespeichert werden und dadurch über Tage vorgehalten werden können (mit Sicherheitsproblemen – ein anderer nutzt die Webanwendung, nur weil sich der Nutzer nicht ausgeloggt hat).

Sie können sich ausgeben lassen, wie lange eine Session-Zeit eingestellt ist: `echo session_cache_limiter()` bzw. eine eigene Zeit setzen, nach dem die Session aus Mangel an Bewegung verfällt (ohne Aktivität war). Über den Befehl `session_cache_limiter(20)` wird die Lebensdauer der Session auf 20 Minuten gesetzt.

Session-Variablen zerstören

Es kann gezielt die Session-Variable gelöscht werden. Dies geschieht für einzelne Variablen über `unset($_SESSION['variablenname']);` oder für eine komplette Session über `unset($_SESSION);`

Übung: wir erstellen einen geschlossenen Zugangsbereich

Für den geschlossenen Zugangsbereich benötigen wir folgende Bestandteile:

Eine Session-Variable, in der gespeichert ist, ob der Nutzer berechtigt ist – diese ist bei Berechtigung dann gesetzt mit: `$_SESSION['eingeloggt'] == True`

Ist diese gesetzt, wird der Inhalt für den berechtigten Besucher angezeigt (im Beispiel die Uhrzeit).

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Ist diese `$_SESSION['eingeloggt']` nicht gesetzt, soll ein Formular angezeigt werden, indem der Besucher seinen Benutzernamen und sein Kennwort eingeben kann.

Und diese Eingaben werden überprüft (hier hart codiert im Quellcode (`benutzername="php-kurs.com"` und `kennwort="geheim"`) und wenn korrekt, dann entsprechend die `$_SESSION['eingeloggt']` auf Wahr gesetzt.

Zeichnen Sie als erstes ein Ablaufdiagramm

Im Ablaufdiagramm haben Sie folgende Programmteile:

- Formular zum Erfassen der Benutzerdaten
- Kontrolle, ob erfasste Benutzerdaten korrekt sind mit hinterlegten Vorgaben
- Wenn SESSION-Variable eingeloggt gesetzt ist, Anzeige von Inhalten für berechtigten Nutzer

Wenn Sie bei der Umsetzung hängen bleiben, können Sie sich einen Lösungsweg für Einlogg-Script ansehen.

Lösung Einlogg-Script

```
<?php
session_start();

if ( $_POST['benutzername'] != "" AND $_POST['kennwort'] != "" )
{
    // Kontrolle, ob Benutzername und Kennwort vorhanden
    // diese werden i.d.R. aus Datenbank ausgelesen
    if (
        $_POST['benutzername'] == "php-kurs.com"
        AND
        $_POST['kennwort'] == "geheim"
    )
    {
        $_SESSION['benutzername'] = $_POST['benutzername'];
        $_SESSION['eingeloggt'] = True;
        echo "<b>einloggen erfolgreich</b>";
    }
    else
    {
        echo "<b>ungültige Eingabe</b>";
        $_SESSION['eingeloggt'] = False;
    }
}

if ( $_SESSION['eingeloggt'] == True )
{
    // Benutzer begrüessen
    echo "<h1>Hallo ". $_SESSION['benutzername'] . "</h1>";
}
else
{
    // Einloggformular anzeigen
    echo "<h1>Bitte loggen Sie sich ein</h1>";

    echo '<form name="" action="" method="POST" enctype="text/html">';
    echo '<p>Benutzername:<br />';
    echo '<input type="text" name="benutzername" value="" size="20"
maxlength="50" />';
    echo '<p>Kennwort:<br />';
    echo '<input type="password" name="kennwort" value="" size="20"
maxlength="50" />';
    echo '<p><input type="Submit" name="einloggen" value="einloggen" />';
    echo '</form>';
}
?>
```

Grafiken erstellen

PHP eignet sich hervorragend, Grafiken zu erzeugen bzw. zu manipulieren. PHP kann z. B. eine E-Mail-Adresse in eine Grafik umwandeln (gut gegen Spam-Roboter) oder grafische Counter erstellen (fertiger Hintergrund wird mit aktueller Zahl ergänzt). Auch CAPTCHA-Grafiken (zur Kontrolle, ob Eingabe von Roboter oder Mensch stammt) sind Beispiele für on-the-fly-erzeugte Grafiken.

Grundsätzliche Schritte beim Grafik Erzeugen mit PHP

Es werden i. d. R. folgende 5 Schritte benötigt:

- Größe der Grafik wird als Rechteck definiert
- Benötigte Farben werden definiert
- Formen (Rechtecke, Kreise, Ellipsen) werden auf der Grafik platziert. Schrift wird platziert
- Ausgabe der Grafik (entweder direkt auf Bildschirm oder als Datei)
- Speicher wird wieder freigegeben

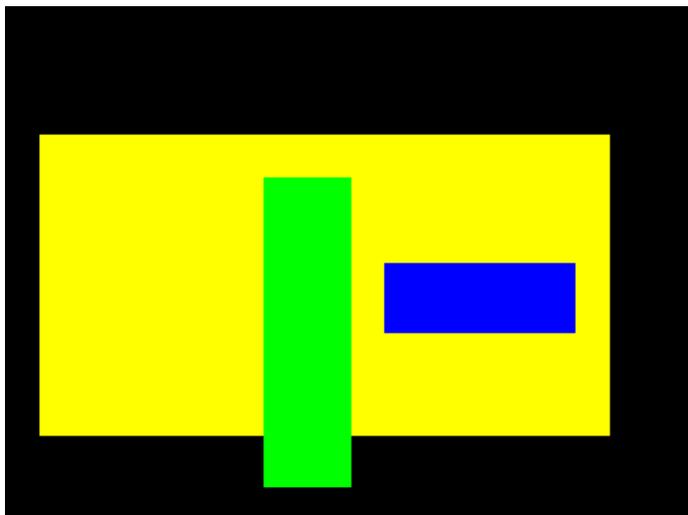
PHP-Grafiken in HTML platzieren

Grafiken können zur direkten Ausgabe über HTML platziert werden. Der Aufruf im HTML-Code geschieht dann über die PHP-Dateinamen.

```

```

Für ein sauberes HTML sollten noch die Breite und die Höhe angegeben werden. In der PHP-Datei "on-the-fly-grafik.php" wird die Grafik erzeugt und dann dem HTML zur Verfügung gestellt. Im folgenden Beispiel werden 3 Rechtecke auf schwarzem Grund erstellt und ausgegeben.



Inhalt der PHP-Datei on-the-fly-grafik-php

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

```

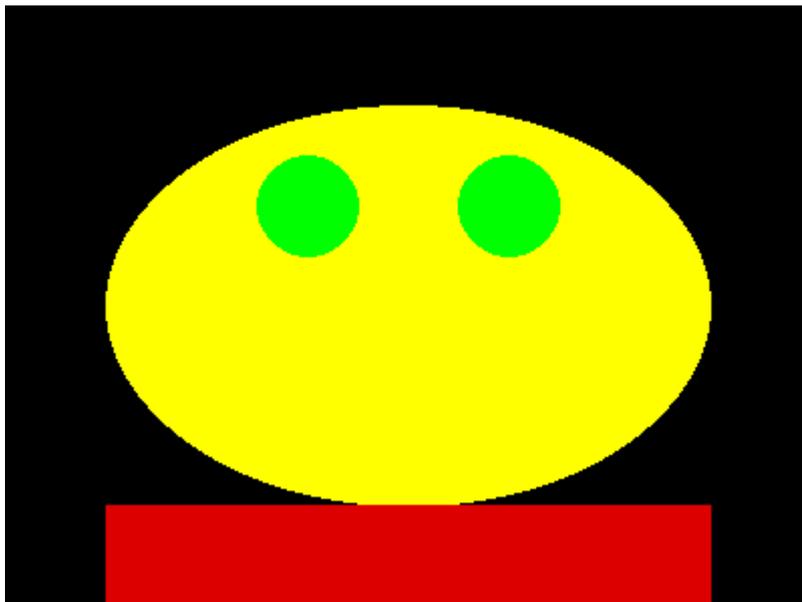
<?php
header("Content-type: image/png");
// erstellen eines leeren Bildes mit 400px Breite und 300px Höhe
$bild = imagecreatetruecolor(400, 300);
// Hintergrundfarbe erstellen
imagecolorallocate($bild, 150, 150, 0);

// Farben festlegen
$farbe1 = imagecolorallocate($bild, 255, 255, 0);
$farbe2 = imagecolorallocate($bild, 0, 255, 0);
$farbe3 = imagecolorallocate($bild, 0, 0, 255);
// Viereck zeichnen
// mit folgenden Koordinaten (x1, y1, x2, y2, Farbe);
imagefilledrectangle ($bild, 20, 75, 350,250, $farbe1);
imagefilledrectangle ($bild, 150, 100, 200, 280, $farbe2);
imagefilledrectangle ($bild, 220, 150, 330, 190, $farbe3);
// Ausgabe des Bildes
imagepng($bild);
?>

```

Aufgabe:

Erstellen Sie folgende Grafik:



In der Erweiterung zu den Rechtecken benötigen Sie die Funktion zum Ellipsen Zeichnen:

```
imagefilledellipse($bild, 200, 150, 300, 200, $farbe);
```

Diese hat den Aufbau:

(x-Koordinate der Mitte, y-Koordinate der Mitte, Breite, Höhe, Farbe)

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Wenn Sie eine fertige Grafik haben, gibt es interessante Möglichkeiten von PHP, auf bestehenden Grafiken weitere Grafiken, bzw. Text legen zu lassen und dies dann als eine neue Grafik anzuzeigen, wie Sie sich im folgenden Abschnitt ansehen können.

Mit PHP einen Text in Grafik integrieren

Um Text in ein Bild zu bringen, gibt es den Befehl "ImageString". Diesem kann über 1 bis 5 eine Textnummer als Größe mitgegeben werden (einfach ausprobieren).

```
<?php
$textnr = 4;
header ("Content-type: image/png");
$bild = imagecreate (400 , 50);
imagecolorallocate ($bild, 0, 0, 0);
$text_farbe = ImageColorAllocate ($bild, 255, 255, 0);

ImageString ($bild, $textnr, 0, 0, "Hallo Welt", $text_farbe);
ImagePNG ($bild);
?>
```



Aufgabe: mit PHP eine E-Mail als Grafik ausgeben

Erstellen Sie ein Programm, das einen Text in eine Grafik umwandelt (Ihre eigene E-Mail-Adresse). Die Funktion können Sie testen unter:

<http://www.iwke.de/kunden/php-kurs/email-in-grafik-umwandeln.php>

Die Herausforderung ist: die Grafik soll exakt so groß sein wie der Text Platz benötigt. Dabei müssen Sie die Größe der Grafik berechnen. Dafür gibt es die folgende Befehle.

imagefontwidth(\$textnr) – berechnet die benötigten Pixel pro Zeichen für die Breite, in der Klammer wird die Nummer des verwendeten Textes mitgegeben

\$stexthoehe = imagefontheight(\$textnr) – berechnet die benötigten Pixel pro Zeichen für die Höhe, in der Klammer wird die Nummer des verwendeten Textes mitgegeben.

Zur Erinnerung: `strlen($text)` gibt die Anzahl der Zeichen in einem String aus.

Zum Vergleich des eigenen Lösungswegs gibt es den Quellcode: E-Mail als Grafik.

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Lösung E-Mail als Grafik

Über folgenden Quellcode wird der Text als Grafik ausgegeben. Die Hintergrundfarbe wurde auf grau gesetzt, damit die Grenzen sichtbar sind.

Als Ausgabe sollten dann etwas in dieser Richtung erscheinen:

`kontakt@php-kurs.com`

```
<?php
$email = "hier kommt Ihre E-Mail-Adresse";
$textnr = 4;
$textbreite = imagefontwidth($textnr) * strlen($email);
$texthoehe = imagefontheight($textnr);

header ("Content-type: image/png");
$bild = imagecreate ($textbreite , $texthoehe);

$hintergrund_farbe = imagecolorallocate ($bild, 222, 222, 222);
$text_farbe = imagecolorallocate ($bild, 0, 0, 0);

imagestring ($bild, $textnr, 0, 0, $email, $text_farbe);
imagepng ($bild);
?>
```

CAPTCHA erstellen

CAPTCHAs sind die Grafiken mit Buchstaben und Zahlen, die einem das Leben schwer machen bei der Anmeldung in Foren, für Newsletters und viele weiteren Geschichten. Das Wort CAPTCHA steht für **C**ompletely **A**utomated **P**ublic **T**uring test to tell **C**omputers and **H**umans **A**part.

Als Funktion soll anhand dieser "Wort-Bilder" getestet werden, ob auf der anderen Seite (somit also auf beiden Seiten) ein Computer sitzt und vollautomatisch sich in Foren anmeldet, um diese zuzusammen.

Dass diese CAPTCHAs teilweise zu einfach sind, so dass auch ein Computer diese über Texterkennung auslesen kann (siehe [Heise](#)) bzw. diese zu komplex sind, so dass selbst Menschen raten müssen (siehe [Sam Hocevar](#)), was da wohl stehen könnte, ist ein anderes Thema.

Im folgenden Beispiel werden wir eine CAPTCHA-Clock basteln - didaktisch raffiniert, denn darin ist schön zu sehen, wie man eine fertige Grafik verwenden kann und dort neuen, wechselnden Text unterbringt.

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

CAPTCHA-Clock - mehr Schutz im Uhrzeigersinn



Vorgehensweise ist, dass man ein Bild für den Hintergrund hat, auf dem dann die Zeichen platziert werden. Hier unsere leere CAPTCHA-Clock - CAPTCHA-Clock deshalb, weil wie bei der Uhr der Inhalt im Kreis im Uhrzeigersinn gelesen wird und dies die meisten CAPTCHA-Roboter aus dem Tritt bringt (bisher habe ich nur lineare CAPTCHAs gesehen :).

Die Grafik erstellen Sie mit einem Grafikprogramm (und für diese Übung laden Sie einfach diese Grafik herunter: mit rechten Maustaste auf das Bild klicken und "Bild speichern unter" wählen ...).



Und nach dem Ausfüllen des Programmes erscheinen dann (jedesmal andere) Zeichen (hier jedesmal wenn die Seite neu aufgerufen wird).

Im ersten Schritt werden wir die Grafik laden und dort die Zeichen platzieren. Erst im nächsten Schritt wollen wir auf der Grafikergänzung ein CAPTCHA basteln - es werden also Zeichen nach Zufall ausgewählt, angezeigt und im Hintergrund in einer SESSION zur Auswertung mitübertragen.

Die Erklärungen finden Sie im Quellcode - achten Sie darauf, dass der Font (hier arial.ttf) auch später auf

dem Webserver vorliegt, sonst funktioniert es nicht.

```
<?php
// Unsere Zeichen in einem Array
$text[0] = "H";
$text[1] = "S";
$text[2] = "L";
$text[3] = "8";
$text[4] = "U";

Header ("Content-type: image/png");

// Laden der Rohdatei, die sich im Verzeichnis befinden muss
$bild = ImageCreateFromPNG ("captcha-clock.png");

// Farben definieren
$farbe_w = ImageColorAllocate ($bild, 255, 255, 255);
$farbe_b = ImageColorAllocate ($bild, 0, 0, 0);
```

```

// die verwendete Schrift-Datei arial.ttf muss vorhanden sein
// Aufbau Schriftgröße, Winkel, X-Pos., Y-Pos, Farbe, Schrift, Inhalt
ImageTTFText ($bild, 32, -30, 140, 70, $farbe_b, "arial.ttf", $text[0]);
ImageTTFText ($bild, 32, -90, 180, 130, $farbe_b, "arial.ttf", $text[1]);
ImageTTFText ($bild, 32, 0, 130, 210, $farbe_b, "arial.ttf", $text[2]);
ImageTTFText ($bild, 32, 45, 80, 190, $farbe_b, "arial.ttf", $text[3]);
ImageTTFText ($bild, 32, 99, 80, 120, $farbe_b, "arial.ttf", $text[4]);

// Ausgabe im Format PNG
ImagePng ($bild);

// Ressourcen wieder freigeben
ImageDestroy ($bild);
?>

```

Wenn das Programm gestartet wird, sollte nun obiges Bild erscheinen - noch mit immer denselben Zeichen. Spielen Sie mit der Schriftgröße, Farbe und Position.

Zufall in CAPTCHA integrieren

Für Benutzerfreundlichkeit sollen im CAPTCHA nur eindeutige Zeichen vorkommen können - also keine Zeichen, die man gerne verwechselt wie O (Ohhh) und 0 (Null), kein I und i (I wie igitt) oder L und l (wie lustig) oder 1 (eins) und keine 6 und 9, da diese auf dem Kopf stehend zu Missverständnissen führen können. Um es ganz sauber zu machen, nur Großbuchstaben verwenden.

Tipp für die Vorgehensweise: Wie legen uns also ein \$array in Kurzschreibweise an (siehe Kapitel “Array”), lassen dieses einmal mischen (siehe Kapitel “nützliche Array-Funktionen”) und dann auf 5 Elemente kürzen (siehe php.net unter `array_slice($input, 0, 5);`).

Dieses Array kommt dann in eine SESSION, damit wir später die Benutzereingabe mit der wirklichen Kombination vergleichen können (siehe: “Sessions”).

Basteln Sie nun das CAPTCHA.

Möglicher Lösungsweg zum Überprüfen des eigenen Weges auf der nächsten Seite.

Einlogg-Script mit CAPTCHA

Und um die Sache abzurunden, basteln Sie nun ein komplettes Formular mit SESSION-Speicherung der Zeichen-Kombination und nach dem Absenden des Formulars mit Kontrolle der vom Benutzer eingegebenen Zeichen.

Integrieren Sie das Captcha nun (siehe unter “Einlogg-Script”).

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Script Captcha-Clock

Mögliche Lösung für die Captcha-Clock - die Session-Variable nennen wir "captcha-wert".

```
<?php
session_start();

// alle moeglichen Zeichen ohne Problemfälle
$moegliche_zeichen = "A B C D E F G H J K M N Q R T U V W X Y 2 3 4 5 7 8";

// erstellen eines Arrays
$kombinationen = explode(" ", $moegliche_zeichen);

// mischen des Arrays
shuffle ( $kombinationen );

// die ersten 6 Zeichen aus dem gemischten Array
$text = array_slice($kombinationen, 0, 5);

// Speicherung in SESSION-Variable für spätere Auswertung
$_SESSION['captcha-wert'] = $text;

Header ("Content-type: image/png");

// Laden der Rohdatei, die sich im Verzeichnis befinden muss
$bild = ImageCreateFromPNG ("captcha-clock.png");

// Farben definieren
$farbe_w = ImageColorAllocate ($bild, 255, 255, 255);
$farbe_b = ImageColorAllocate ($bild, 0, 0, 0);

// die verwendete Schrift-Datei arial.ttf muss vorhanden sein
// Aufbau Schriftgröße, Winkel, X-Pos., Y-Pos, Farbe, Schrift, Inhalt
ImageTTFText ($bild, 32, -30, 140, 70, $farbe_b, "arial.ttf", $text[0]);
ImageTTFText ($bild, 32, -90, 180, 130, $farbe_b, "arial.ttf", $text[1]);
ImageTTFText ($bild, 32, 0, 130, 210, $farbe_b, "arial.ttf", $text[2]);
ImageTTFText ($bild, 32, 45, 80, 190, $farbe_b, "arial.ttf", $text[3]);
ImageTTFText ($bild, 32, 99, 80, 120, $farbe_b, "arial.ttf", $text[4]);

// Ausgabe im Format PNG
ImagePng ($bild);

// Ressourcen wieder freigeben
ImageDestroy ($bild);
?>
```

Objektorientierte Programmierung in PHP

Objektorientierte Programmierung zu verstehen ist nicht das einfachste. Bisher haben wir prozedural programmiert. Vereinfacht gesagt, das Programm fängt oben an und hört unten auf - vielleicht mit ein paar Sprüngen noch in Funktionen, aber ansonsten sehr linear.

Bei der Objektorientierten Programmierung ist die Zielsetzung, dass

- Quellcode einfacher wiederverwertet werden kann
- der Quellcode übersichtlicher wird
- zukünftige Erweiterungen einfach werden.

Vor dem gesamten "einfacher, übersichtlicher und besser" steht i.d.R. ein härterer Lernprozess, der es aber Wert ist. Die anfänglichen Tränen werden später durch viel gesparte Zeit wett gemacht - Sie können die Tränen am Meer trocken lassen, wo andere immer noch an derselben Aufgabe prozedural programmieren.

Ein weiterer großer Vorteil entsteht, dass Sie selbst durch ein Grundverständnis fremde objektorientiert programmierte Klassen in Ihren eigenen Projekte einsetzen können und dadurch massig Zeit sparen.

Wichtige Begriffe der Objektorientierten Programmierung

Vor dem Programmieren steht das Verstehen - dazu muss das Konzept hinter der Objektorientierten Programmierung verstanden werden und dazu werden ein paar wenige Grundbegriffe benötigt: Klassen, Objekte, Eigenschaften (Attribute), Methoden, Konstruktor, Destruktor und Vererbung bei Klassen. Diese werden im Folgenden erklärt.

Grundgedanke:

- alles wird als jeweils ein Objekt gesehen!

Grundgedanke Objektorientierte Programmierung:

- alles wird als ein Objekt gesehen!
- jedes Objekt hat Eigenschaften/Attribute ...

Eigenschaften (Attribute) eines Objektes

Das Objekt hat verschiedene Eigenschaften (Attribute), bzw. es können ihm bestimmte Eigenschaften zugewiesen werden. Hier werden Sie in der Regel das Beispiel "Auto" finden. Ein Auto hat als Eigenschaft 4 Räder (meinetwegen 5, wenn man das Ersatzrad mitrechnet, oder 3, wenn es sich um eine italienische Ape handelt).

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Eine weitere Eigenschaft des Autos ist die Farbe.

Noch eine Eigenschaft ist der benötigte Kraftstoff (Benzin, Diesel, Ethanol, ...).

Im Vergleich zur bisherigen Programmierung sind die Eigenschaften wie Variablen.

Grundgedanke Objektorientierte Programmierung: Methoden

- alles wird als ein Objekt gesehen!
- jedes Objekt hat Eigenschaften/Attribute
- jedes Objekt hat Methoden

Methoden eines Objektes

Um bei unserer Analogie mit dem Auto zu bleiben: Ein Auto kann normalerweise beschleunigen, bremsen, abbiegen. Dies sind Methoden - flapsig im Vergleich zu den Eigenschaften gesagt, ändern diese ihren Zustand öfters als z.B. die Anzahl der Räder.

Im Vergleich zur bisherigen Programmierung können Funktionen als Methoden angesehen werden.

Grundgedanke Objektorientierte Programmierung: Klassen und Objekte

- alles wird als ein Objekt gesehen!
- jedes Objekt hat Eigenschaften/Attribute
- jedes Objekt hat Methoden
- die Klasse bildet den Bauplan - das Objekt ist der fertige Bau

Klassen und Objekte

In einer Klasse ist festgelegt, welche Attribute (flapsig Eigenschaften, Variablen) und Methoden (flapsig Fähigkeiten, Funktionen) diese Klasse besitzt. Im davon erzeugten Objekt werden die Eigenschaften mit Werten belegt.

Beispiel Auto:

Eigenschaft:

- Farbe
- benötigter Kraftstoff
- Anzahl Räder
- Preis

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Methoden:

- beschleunigen
- bremsen
- Messung aktuelle Geschwindigkeit
- Messung aktuell vorhandene Kraftstoffmenge
- u.v.m.

So - nun wird von einer Klasse ein Objekt gebildet (toller Sprachgebrauch, der viel hermacht und erst mal verwirrt :

Warum das? Nun, man möchte mehr als 1 Auto haben. Also werden aus dem Bauplan der Klasse so viele Objekte abgeleitet, wie man benötigt, denen man dann die gewünschten Eigenschaften zuweisen kann.

Oder anders ausgedrückt - man zieht nicht in den Bauplan ein, sondern in das fertig gebaute Haus.

Beispiel Auto und Familie

Beispiel: Die Familie möchte gerne 3 Autos (mieses Beispiel für eine grüne Stadt wie Tübingen)

Erstes Auto muss was hermachen, für das Ausgehen und die Geschäftsfahrten - also Farbe Silber und teuer, Größe mindestens 3 Meter

zweites Auto: klein und wendig und knallrot, damit man es auf dem Parkplatz wiederfindet, und kurz, damit man überhaupt einen Parkplatz bekommt und billig und sicher, damit der Nachwuchs es ohne größere Schaden zu nehmen zu Schrott fahren kann.

drittes Auto für Urlaub: am besten einen günstigen Mietwagen

Wir haben also 3 Objekte (von der 1 Klasse Auto) - die Klasse kann also als Blaupause (bzw. Bauplan) angesehen werden und die daraus abgeleiteten Objekte als reale Objekte, die durch den Bauplan einfach und in beliebiger Menge gebaut (erzeugt) werden können.

Beispiel Klasse/Objekt

Nach der vielen Theorie nun zur Veranschaulichung ein Beispiel:

```
<?php
// die Klasse "Auto" definieren
class auto
{
    // eine Eigenschaft/Attribut festlegen
    var $benoetigter_kraftstoff;
```

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

```

// eine Methode festlegen (nennt sich Funktion,
// ist aber in einer Klasse und
// wird somit als Methode bezeichnet)
function tankdeckel_oeffnen()
{
    // Ihr Auto spricht mit Ihnen - wenn der
    // Tankdeckel geöffnet wird, sagt es, welchen
    // Kraftstoff es benötigt
    echo "<p>Bitte mit $this->benoetigter_kraftstoff betanken";
}

// bisher passiert noch gar nichts,
// jetzt wird aus der Klasse ein Objekt erzeugt
$auto_1 = new auto;

// dem Auto wird nun der Kraftstoff zugewiesen,
// eine Eigenschaft (Attribut) wird definiert
$auto_1->benoetigter_kraftstoff = "Diesel";

// und nun wird das erste mal die Methode (Funktion)
// tankdeckel_oeffnen aufgerufen und das Auto sagt
// freudig, was es für Sprit benötigt
$auto_1->tankdeckel_oeffnen();
?>

```

Das mag für die wenige Funktion erscheinen, als schieße man auf Spatzen mit Kanonen (soll vorkommen). Interessant ist es, wenn wir nun für die ganze Familie die Autos erzeugen.

```

// das neues Objekt auto_2 wird "gekauft"
$auto_2 = new auto;
$auto_2->benoetigter_kraftstoff = "Super";

// das neues Objekt auto_3 wird erstellt
$auto_3 = new auto;
$auto_3->benoetigter_kraftstoff = "Ethanol";

// Wird nun mit Auto 3 tanken gegangen,
// erhält man den richtigen Hinweis,
// was das Auto schluckt mit:
$auto_3->tankdeckel_oeffnen();

```

Aufgabe

Ergänzen Sie das Beispiel oben, dass das Auto nicht nur sagt, was es für Treibstoff benötigt, sondern auch, was für eine Marke es ist (z. B. ein Benz, ein BMW, eine ABC-Klasse oder so).

Gedächtnis des Objektes

Jedes Objekt hat sein eigenes Gedächtnis (die Daten sind gekapselt) - im obigen Beispiel merkt sich das Auto nun, wie oft es in seinem Leben betankt wurde. Dazu benötigen wir eine weitere Eigenschaft (anzahl_betankungen) und die Methode tankdeckel_oeffnen wird erweitert (OK, nicht jedesmal, wenn der Tankdeckel geöffnet wird, wird auch getankt - aber so ist es nett einfach)

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

```

<?php

// die Klasse "Auto" definieren
class auto
{

    // eine Eigenschaft/Attribut festlegen
    var $benoetigter_kraftstoff;
    // Eigenschaft anzahl_betankung setzen und Start-Wert zuweisen
    var $anzahl_betankungen = 0;

    // eine Methode (flapsig Funktion) festlegen
    function tankdeckel_oeffnen()
    {
        // Ihr Auto spricht mit Ihnen - wenn der
        // Tankdeckel geöffnet wird, sagt es, welchen
        // Kraftstoff es benötigt
        echo "<p>Bitte mit $this->benoetigter_kraftstoff betanken";
        echo "<p>Ich wurde $this->anzahl_betankungen mal betankt";

        // Anzahl der betankungen wird um 1 erweitert
        $this->anzahl_betankungen++;
    }
}

// bisher passiert noch gar nichts,
// jetzt wird aus der Klasse ein Objekt erzeugt
$auto_1 = new auto;

// dem Auto wird nun der Kraftstoff zugewiesen,
// eine Eigenschaft (Attribut) wird definiert
$auto_1->benoetigter_kraftstoff = "Diesel";

// und nun wird das erste mal die Methode (Funktion)
// tankdeckel_oeffnen aufgerufen und das Auto sagt
// freudig, was es für Sprit benötigt
$auto_1->tankdeckel_oeffnen();

// Damit man das Gedächtnis des Objektes sieht
$auto_1->tankdeckel_oeffnen();

?>

```

Anwendung von Objektorientierter Programmierung oder Prozeduraler Programmierung

Bei einigen Programmen wird es nicht sinnvoll sein, diese objektorientiert zu erstellen, da wirklich mit Kanonen auf Spatzen geschossen wird, bei anderen Programmen wird es sehr wohl sinnvoll sein. Dies ist immer die Entscheidung des Programmierers. Wenn Sie sich verschiedene Programme von anderen Programmierern ansehen, werden Sie sich manchmal fragen, warum nicht objektorientiert. Das kann verschiedene Gründe haben, wie z.B. dass das Ursprungsprogramm ganz ganz klein war und erst mit der Zeit in Größe und Komplexität gewachsen ist oder schlicht und ergreifend der entsprechende Programmierer nicht firm ist in Objektorientierter Programmierung. Der erste Schritt zum Verständnis ist wichtig und kann unter Umständen mehrere Anläufe benötigen – aber es lohnt sich!

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Ausgabe der Eigenschaften eines Objektes:

Sie können auch die Eigenschaften direkt abfragen (das sollte aber in der Regel über die Methoden stattfinden:

```
echo $auto_1->benoetigter_kraftstoff;  
echo $auto_1->anzahl_betankungen;
```

Aufgabe:

Erstellen Sie die Klasse "student"

Diese Klasse hat die Eigenschaft:

- Geschlecht
- Alter in Jahren
- Hauptfach

Und es gibt die Methoden

- hat Geburtstag (Alter steigt i.d.R. um 1 Jahr) und eine Ausgabe auf dem Bildschirm erscheint mit dem entsprechenden Alter
- wechselt Hauptfach und eine Ausgabe auf dem Bildschirm erscheint, welches das alte Fach war und welches das neue Fach ist.

Leiten Sie 2 Objekte ab:

1. Objekt: Susi S., Alter 22, Hauptfach Englisch
2. Objekt: Alfred A., Alter 24, Hauptfach Ameisenkunde

Lassen Sie Alfred 2 mal Geburtstag haben und Susi das Fach wechseln nach Latein

Konstruktor / Destruktor

Konstruktor

Bezeichnet den Programmcode, der am Anfang beim Aufruf einmal abgearbeitet wird. In unserem Beispiel: Bau mir ein Auto, das 4 Räder hat und erstmal die Farbe weiß.

Destruktor

Gegenteil zum Konstruktor - zum Löschen eines Objektes und in PHP erst ab Version 5 verfügbar. Damit wird der Speicher bereinigt und das Programm schnell.

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Vererbung bei Klassen

Damit Sie sich nicht in Sicherheit wiegen, dass Objektorientierte Programmierung zu einfach und deshalb nicht wirklich cool ist - es gibt das Prinzip der Vererbung. So können Klassen universeller anwendbar gemacht werden (was beim Programmieren allerdings den Grad der Komplexität erhöht, aber wichtig ist, darüber etwas Bescheid zu wissen.)

In unserem Beispiel Student könnte eine übergeordnete Klasse stehen mit dem Überbegriff Mensch. Diese hätte dann die Eigenschaften Geschlecht und Alter. Durch die Überordnung erbt dann die Klasse Student diese Eigenschaften, ohne diese selber nochmals definieren zu müssen.

zum selbständigen Weiterbasteln die Grundstruktur

```
class mensch {  
  
}  
  
class student extends mensch {  
  
}
```

PHP5 Klassen

PHP5 bietet nun bei Klassen die Möglichkeit, diese sauber zu kennzeichnen, ob sie von außen zugänglich sein sollen (public) oder nur innerhalb der Klasse verfügbar (private). Dies gilt bei allen Eigenschaften und Methoden.

In PHP4 hat man sich damit beholfen, dass für nicht-öffentliche ein Unterstrich zwischen Dollarzeichen und Variablennamen aufgeführt wurde bzw. bei den Methoden. Dies diente nur zur Kennzeichnung und hatte keinerlei Auswirkungen.

Beispiel bei PHP4-Klassen

```
class sqlabfrage  
{  
    var $sql = "";  
    var $_berechnung = "";  
    function sqlabfrage( $uebergabewert )  
    {  
        // ..  
  
        $erg = $this->_berechnung;  
    }  
}
```

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Beispiel für PHP5-Klassen

```
class sqlabfrage
{
public $sql = "";
private $berechnung = "";

public function sqlabfrage( $uebergabewert )
{
// ..

$erg = $this->berechnung;
}
}
```

Und warum gebe ich mir das Ganze?

Es gibt zahlreiche Klassen im Internet, die einfach in das eigene Programm eingebunden werden können und dadurch dem eigenen Programm zahlreiche Zusatzfunktionen verschaffen, da man selber i.d.R. nicht die Zeit aufwenden möchte, diese alle selbst zu programmieren.

Im Folgenden wollen wir eine PDF-Ausgabe unser Programmes "Hallo Welt" erstellen.

Fremde Klassen FPDF nutzen

Dazu nutzen wir die freie Klasse FPDF, die von Oliver Plathey zum Download unter <http://www.fpdf.org> angeboten wird.

Nach dem Herunterladen und dem Entpacken kommt nun unser Programm:

```
<?php
// Einbinden der Klasse
require('fpdf.php');

// Objekt mit dem Namen pdf erstellen
$pdf=new FPDF();

// Methode Addpage des Objekts ansprechen
$pdf->AddPage();

// Methode SetFont ansprechen und gewünschten Font übergeben
$pdf->SetFont('Arial','B',16);

// Methode Cell zur Ausgabe des Inhalts ansprechen
$pdf->Cell(40,10,'Hello World!');

// Methode Output für die Anzeige des fertigen
// PDF-Dokuments ansprechen
$pdf->Output();
?>
```

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Auf der Website finden Sie zahlreiche Anleitungen, da mit dieser Klasse sehr viel gemacht werden kann.

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

mySQL - Datenbank unter PHP

MySQL, Datenbankmodellierung, Datenbankdesign, Datenredundanz, Schlüsselfelder, Beziehungen, Formalisieren, Entity-Relationship-Modell, ERM, relational und mehr. Schon erschreckt von Fachbegriffen? Mit denen können Sie selber nach der Lektüre der kommenden Seiten um sich werfen. Ist nicht so schwer!

Was ist ein Datenbanksystem?

Eine Datenbank ist eine Sammlung von Informationen über ein Thema, bzw. für einen bestimmten Zweck. Beispiele sind Bücherverwaltungen (OPAC in Universitäten), elektronische Telefonbücher, Waren im Lager, Anwesenheitszeiten zwecks Gehaltsabrechnungen.

Man spricht dann von einem Datenbanksystem, wenn sowohl die Datensammlung wie auch ein Verwaltungssystem vorhanden sind. Unter Verwaltung versteht man, wenn der Benutzer definiert Daten erfassen und verwalten, Suchen durchführen und gegebenenfalls Auswertungen machen kann.

Datenbank warum?

"Warum benötige ich überhaupt eine Datenbank?" ist eine berechtigte Frage. Die Antwort ist eigentlich einfach. Um große Mengen an Daten bequem speichern zu können.

Die Wichtigkeit sieht man z.B. auch auf den Internetseiten bei den verschiedenen Weiterbildungsanbieter wie z.B. der Volkshochschulen und anderen IT-Kurs-Anbieter.

Wie oft das Thema Datenbank in der Weiterbildung auftaucht, ist nicht erstaunlich. Egal welche Programmiersprache Sie nehmen, wird es dort zwangsläufig als Thema auftauchen.

Wenn man eine Datenbank aufbaut, gehört dazu als erstes, das gewünschte Projekt zu verstehen und dann die Datenstruktur sinnvoll für eine Datenbank aufzubereiten. Hört sich kompliziert an, ist es aber nicht.

Im ersten Schritt sehen wir uns die Datenbank selber an.

Daten strukturieren - Vorüberlegungen zur Datenbankerstellung

Aus Daten eine Struktur erstellen

Für die folgenden Schritte wird in den folgenden Kapiteln die Theorie nachgeliefert. Für das Verständnis gehen wir anhand von einem Beispiel die notwendigen und sinnvollen Schritte durch.

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

erste Struktur

Lassen Sie uns an einem Beispiel die Vorgehensweisen und die Fallstricke zeigen. Wir wollen als erstes eine einfache Adressverwaltung erstellen. Wir wollen folgende Informationen speichern:

- Namen
- Anrede
- Straße
- PLZ und Ort
- Telefonnummer
- Leseausweisnummer

Eine Datenbank kann man sich als Tabelle vorstellen. Unsere obigen Daten würden dann so aussehen (alle Namen und Daten sind rein fiktiv und Ähnlichkeiten ...)

name	anrede	strasse	plzundort	telefon	leseausweisnr
Fritz Müller	Herr	Hauptstr. 12	72070 Tübingen	07071-555- 12312	1289234877
Susi Simmer	Frau	Herbstallee 1	72074 Tübingen	07071-555- 654654	
Susi Sommer	Frau		72074 Tübingen	07071-555- 64444	984613325

Die Spalten der Tabelle sind die Datenfelder (bzw. Attribute)

Die Reihen sind jeweils ein Datensatz

Inhalte trennen (atomisieren)

Wichtig ist, dass jedes Feld nur eine Information enthält. In unserem Beispiel bedeutet dies, dass die Felder Namen und PLZ mit Ort von Übel sind. Diese werden also getrennt.

name	anrede	strasse	plzundort	telefon	leseausweisnr
Fritz Müller	Herr	Hauptstr. 12	72070 Tübingen	07071-555- 12312	1289234877
Susi Simmer	Frau	Herbstallee 1	72074 Tübingen	07071-555- 654654	
Susi Sommer	Frau		72074 Tübingen	07071-555- 64444	984613325

Nach der Trennung sieht die Tabelle wie folgt aus. Für die Namen der Spalten sollten sprechende Namen und ohne Umlaute und ohne Sonderzeichen gewählt werden!

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

nachname	vorname	anrede	strasse	plz	ort	telefon	leseausweiser
Müller	Fritz	Herr	Hauptstr. 12	72070	Tübingen	07071-555-12312	1289234877
Simmer	Susi	Frau	Herbstalle 1	72074	Tübingen	07071-555-654654	
Sommer	Susi	Frau		72074	Tübingen	07071-555-64444	984613325

eindeutiger Zugriff

Um bei der Tabelle mit den Adressen einen eindeutigen Zugriff zu bekommen, wird diese erweitert. Da bei Adressen der Nachname öfters vorkommen kann, geht man den Weg, ein weiteres Feld hinzuzufügen, das es **pro Datensatz nur einmal** geben kann. Dieses wird gerne ID genannt – abgekürzt von IDentifizier. Diese bekommen eine fortlaufende Nummern (es darf keine Nummer bereits verwendet worden sein!)

id	nachname	vorname	anrede	strasse	plz	ort	telefon	leseausweiser
1	Müller	Fritz	Herr	Hauptstr. 12	72070	Tübingen	07071-555-12312	1289234877
2	Simmer	Susi	Frau	Herbstalle 1	72074	Tübingen	07071-555-654654	
3	Sommer	Susi	Frau		72074	Tübingen	07071-555-64444	984613325

Diese eindeutige Feld wird Schlüssel genannt – es ist das Schlüsselfeld, über das jeder Datensatz eindeutig angesprochen werden kann. Dazu werden in der realen Welt dann in Vereinen Mitgliedsnummern vergeben, oder Sozialversicherungsnummern, oder im Ausweis die Ausweisnummer. Der Schlüssel zur absolut eindeutigen Identifizierung. Siehe hierzu später unter Primärschlüssel und Sekundärschlüssel.

Das könnten wir so machen und eine Datenbank mit exakt dieser Struktur anlegen. Aber ... wir wollen was lernen und wir wollen die Grundprinzipien bei der Datenbankentwicklung umsetzen. Ein Grundprinzip lautet, keine doppelten Daten zu produzieren.

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Ziel bei der Datenbankentwicklung ist: Mehrfachspeicherung (Datenredundanz) zu vermeiden

Wichtig ist, auf Datenredundanz zu achten. Problem ist, dass dadurch im Lauf der Zeit viele Fehler sich einschleichen, weil es menschelt. Es kann vergessen werden, alle Daten zu korrigieren. Dann wird einmal z. B. nur die PLZ geändert und der Ort vergessen, ein anderesmal wird nur der Ort geändert, aber die PLZ vergessen - es entsteht zunehmend Datenchaos, dass durch einen sinnvollen Aufbau der Datenbank vermieden (oder zumindest minimiert) werden hätte können. Das ganze kann auch mehr Komfort für den Datenerfasser bieten - beispielsweise gibt es bereits fertige Datenbanken mit allen Postleitzahlen und zugehörigen Orten. Wenn diese mit eingebunden wird, muss nur noch die PLZ erfasst werden und der korrekte Ort erscheint dann automatisch.

Bei unserer obigen Adresstabelle fallen 2 Felder ins Auge, die sich gerne inhaltlich wiederholen können – die Anrede und der Ort.

Für den Ort haben wir eine eindeutige Zahl in Deutschland – die PLZ (wir machen es nun nicht komplizierter und gehen einfach davon aus, dass es nur deutsche Adressen sind und es für jede PLZ nur einen Ort gibt!).

Aus unserer einfachen Tabelle sind nun schon 2 geworden. Eine Tabelle mit der Adresse mit PLZ und eine Tabelle mit PLZ und Ort

Tabelle: Adresse:

id	nachname	vorname	anrede	strasse	plz	telefon	leseausweisnr
1	Müller	Fritz	Herr	Hauptstr. 12	72070	07071- 555-12312	1289234877
2	Simmer	Susi	Frau	Herbstallee 1	72074	07071- 555- 654654	
3	Sommer	Susi	Frau		72074	07071- 555-64444	984613325

Tabelle: Plz:

plz	ort
72070	Tübingen
72074	Tübingen

Und weiterhin wollen wir die Anrede herausnehmen – schwups haben wir 3 Tabellen. In der Anrede wird nun nur noch m für männlich und w für weiblich gespeichert.

Tabelle: Adresse:

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

id	nachname	vorname	akuerzel	strasse	plz	telefon	leseausweisnr
1	Müller	Fritz	m	Hauptstr. 12	72070	07071- 555- 12312	1289234877
2	Simmer	Susi	w	Herbstallee 1	72074	07071- 555- 654654	
3	Sommer	Susi	w		72074	07071- 555- 64444	984613325

Tabelle: Plz:

plz	ort
72070	Tübingen
72074	Tübingen

Tabelle: anrede:

akuerzel	anrede
w	Frau
m	Herr

Daten logisch passend

Im dritten Schritt packen wir alle logisch zusammenpassenden Daten jeweils in eine Tabelle - oder andersherum gesehen, wir werfen Datenfelder aus Tabellen, die logisch nicht passen, und bauen daraus eine weitere Tabelle.

In unserem Beispiel haben wir bei der Adresse das Datenfeld "leseausweisnr" - diese passt aus logischer Sicht nicht zu den Daten der Adresse. Wir machen daraus unsere 4. Tabelle - die exakte Begründung für dieses Vorgehen ist im Kapitel Normalisierung (3N) zu finden.

Tabelle: Adresse:

id	nachname	vorname	akuerzel	strasse	plz	telefon
1	Müller	Fritz	m	Hauptstr. 12	72070	07071-555-12312
2	Simmer	Susi	w	Herbstallee 1	72074	07071-555-654654
3	Sommer	Susi	w		72074	07071-555-64444

Tabelle: Plz:

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

plz	ort
72070	Tübingen
72074	Tübingen

Tabelle: anrede:

akuerzel	anrede
w	Frau
m	Herr

Tabelle: anrede:

a-id	leseausweisnr
1	1289234877
3	984613325

Die Person mit der ID 2 hat keinen Leseausweis, dadurch wird kein Datensatz notwendig.

Primär- und Fremdschlüssel

Primärschlüssel

Bei den Adressen haben wir die ID – diese ist der Primärschlüssel der Tabelle "Adressen". In dieser Spalte enthaltene Werte dürfen nur ein einziges Mal vorkommen. Duplikate sind niemals erlaubt! Ein anderer Primärschlüssel ist in der Tabelle "PLZ" die Spalte "plz" und in der Tabelle "Anrede" die Spalte "kuerzel".

Die primäre Bedeutung des Primärschlüssels kommt dann zum Tragen, wenn man 2 Tabellen verknüpft (in Beziehung setzt).

Fremdschlüssel

Was bei der Verknüpfung in der einen Tabelle der Primärschlüssel ist, ist in der zweiten Tabelle der Fremdschlüssel. Der Fremdschlüssel enthält den gleichen Wert wie der Primärschlüssel, kann aber öfters vorkommen (je nach Beziehungsart). So kann er einmal, keinmal oder mehrmals vorkommen.

In unserem Beispiel ist bei der Tabelle "PLZ" das Feld plz der Primärschlüssel und in der Tabelle "Adressen" das Feld PLZ der Fremdschlüssel (kann öfters vorkommen).

Bei der Tabelle "Anrede" ist das "akuerzel" der Primärschlüssel und in der Tabelle "Adresse" die Spalte "akuerzel" der Fremdschlüssel.

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Zum Verleich nochmals unsere Tabellen, in dem die Schlüsselfelder farblich hervorgehoben sind.

Tabelle: Adresse:

id	nachname	vorname	akuerzel	strasse	plz	telefon
1	Müller	Fritz	m	Hauptstr. 12	72070	07071-555-12312
2	Simmer	Susi	w	Herbstallee 1	72074	07071-555-654654
3	Sommer	Susi	w		72074	07071-555-64444

Tabelle: Plz:

plz	ort
72070	Tübingen
72074	Tübingen

Tabelle: anrede:

akuerzel	anrede
w	Frau
m	Herr

Tabelle: anrede:

a-id	leseausweisnr
1	1289234877
3	984613325

Beziehungen

Beziehungen - Verbindungsarten zwischen den verschiedenen Tabellen

Im Gegensatz zur realen Welt sind bei Datenbanken Beziehungen sehr eindeutig (und problemlos). Um diese Beziehungen aufzubauen, wird für jede Tabelle ein eindeutiges Feld benötigt. Bei der Postleitzahl ist es klar – jede Postleitzahl kann nur einmal vorkommen (laut unsere Definition oben). Also kann man über die Tabelle Adresse und dem Feld PLZ dann in der Tabelle PLZ-ORT den entsprechenden Ort auslesen.

Bei der Anrede haben wir als Auswahl (m) für Herr und (w) für Frau. Die Tabelle für die Anreden wird um das Feld kuerzel ergänzt.

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Tabelle: anrede:

akuerzel	anrede
w	Frau
m	Herr

Schauen wir uns nun die Beziehungen an:

Anrede: ein Kürzel bei der Anrede kann nur eine Anrede erhalten

PLZ: eine PLZ kann nur einen ORT haben.

Beides sind 1:1 Beziehungen (in der realen Welt würden wir von monogam sprechen)

1:n Beziehungen

Natürlich sind auch andere Beziehungen möglich. Beispielweise könnten wir zu den Adressen noch verwalten, wenn wir an Leute Bücher ausgeliehen haben. (wir benötigen noch eine Tabelle für die Bücher).

Jetzt wird es vorkommen, dass wir an eine Person mehr als ein Buch verleihen. Wir müssen also in unserer Tabelle eine 1:n Beziehung abbilden können.

Schon ist es bei den Tabellen und Datenbanken vorbei mit der Monogamie. Eine Person kann viele Bücher haben, aber jedes Buch kann sich gerade nur bei einer Person befinden – also 1 zu n.

n:m Beziehungen

Komplexer wird es, wenn wir Beziehungen haben, die einen Mehrfachtatbestand erfüllen. Erweitern wir unser Beispiel. Ich kann den Adressen nun z.B. besuchte Kurse (Unterrichtsfächer, Studienkurse,...) zuordnen. Jede Person kann mehrere Kurse besuchen. Und jeder Kurs kann verschiedene Personen als Teilnehmer beinhalten. Das ergibt eine n:m Beziehung.

mathematisch gesehen:

1 Person kann 0,1 oder n Kurse besuchen

1 Kurs kann 0,1 oder m Teilnehmer haben

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Datenstruktur grafisch darstellen

Für eine einfache und schnell umzusetzende grafische Darstellung bietet sich das Streifendiagramm und Excel an (die Beispiele unten sind mit Excel gemacht).

Fremdschlüssel sind gelb hervorgehoben

Primärschlüssel sind hier gelb hervorgehoben und unterstrichen

Die **Beziehungen zwischen Primärschlüsseln und Fremdschlüsseln** sind durch Linien kenntlich gemacht.

An den Linienenden stehen die Beziehungsarten:

1 : 1 - eins-zu-eins Beziehung

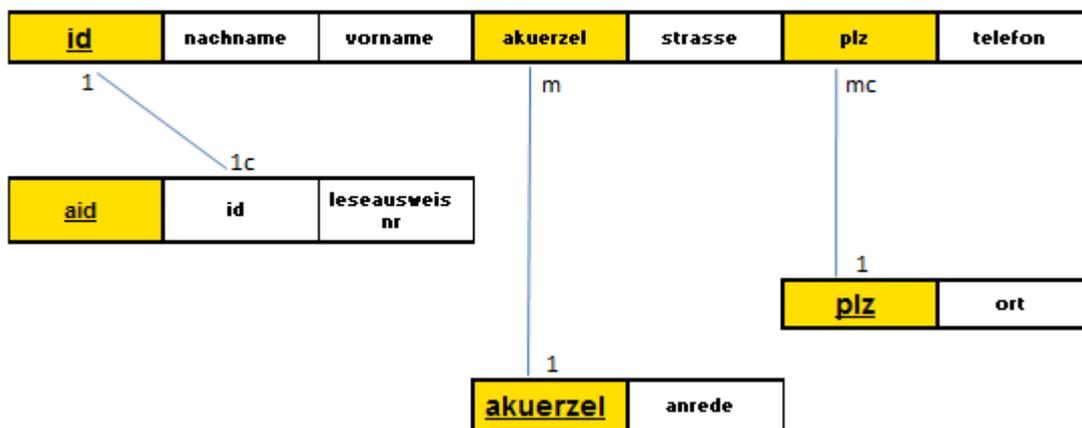
1 : m - eins-zu-mehrfach-Beziehung

1 : 1c - bedingte eins-zu-eins-Beziehung

1 : mc - bedingte eins-zu-mehrfach-Beziehung

(das c steht für conditional, also im Beispiel von 1 : 1c kann es einen Datensatz geben, muss aber nicht - nicht jede Adresse hat auch einen Leseausweis).

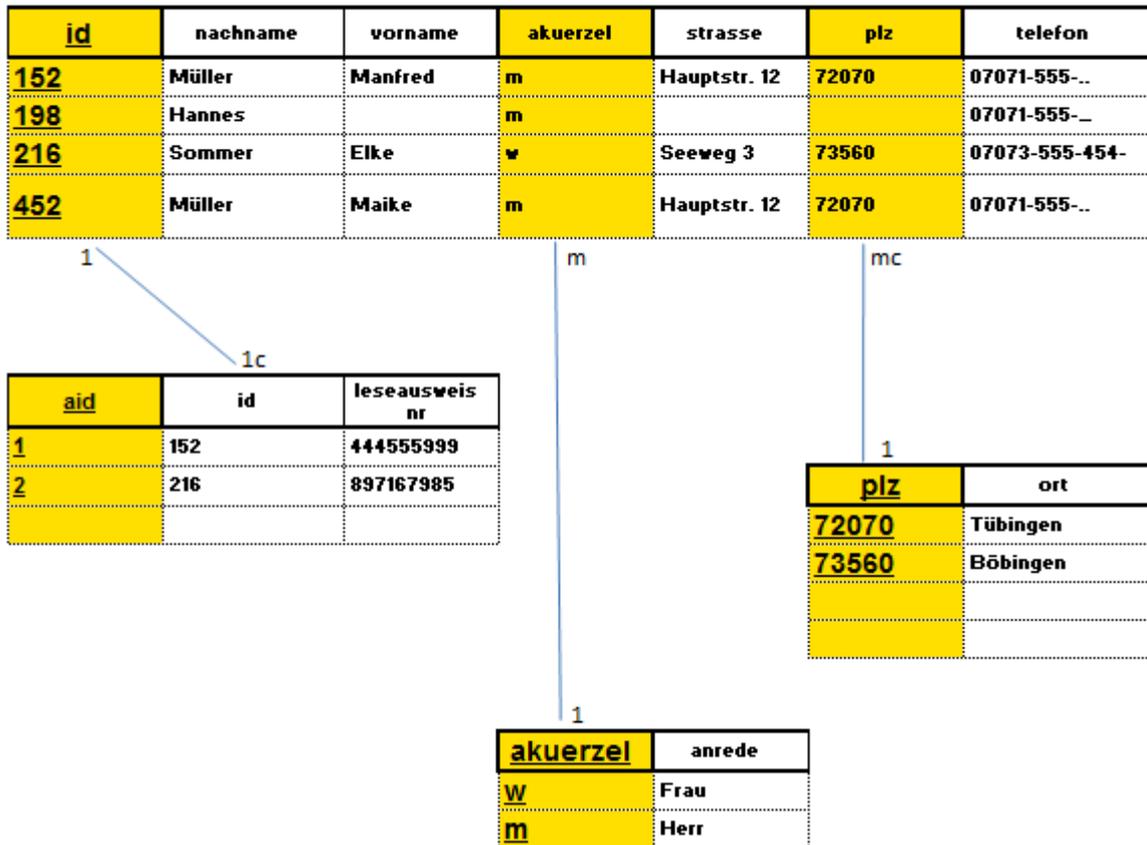
Streifendiagramm "adressverwaltung"



Zur Erklärung von 1 : mc - bei jedem Datensatz kann es einen Eintrag für die Postleitzahl geben, genauso sind mehrere möglich, da mehrere Leute in derselben Stadt wohnen oder es kann keinen geben, da niemand aus einem Ort kommt.

Sehen wir uns das Beispiel mit Spieldaten an (alle Daten sind rein fiktiv ...)

Streifendiagramm "adressverwaltung"



Bei unserem Leseausweis bedeutet das, dass jeder Leseausweis eindeutig einer Person zugeordnet werden kann. Überlegen Sie, wie das Beispiel aussehen würde, wenn es "Familienleseausweise" geben würde mit derselben Nummer - wie sieht die Zeichnung dann aus?

Aufgabe: Ergänzen Sie das Beispiel um ausgeliehene Bücher

Sie sind nun die Stadtbücherei und möchten Ihre Bücher verwalten. Ergänzen Sie das obige Beispiel. Es soll für das Personal möglich sein abzufragen, wer ein Buch ausgeliehen hat, wann es ausgeliehen wurde und wann die Leihfrist abläuft (Bücher 4 Wochen, CDs 2 Wochen, DVDs 1 Woche). Das Personal soll sich auch alle Bücher von einem Autor anzeigen lassen können!

Aufbau Datensatz

Aufbau eines Datensatz - was technisch benötigt wird

Wir kommen vom Großen ins Kleine. Jeder Datensatz besteht aus verschiedenen Datenfeldern. Diese Datenfelder haben eine Struktur (Datensatzaufbau). Dazu gehört:

- Name des Datenfelds
- Type des Datenfelds
- Länge des Datenfelds

Datenfeldname

Um gezielt ein Felder ansprechen zu können, werden die Datenfeldnamen vergeben. Diese sollten soweit sprechend sein, dass der Programmierer sich schnell zurecht findet – andererseits besteht die Kunst darin, dass diese Benennung nicht zu lange ist, damit der Programmierer nicht zuviel tippen muss. Beispielweise bei der Tabelle "Buch" sind Feldname vorstellbar wie "titel", "erscheinungsjahr", "auflage", "preis".

Datenfeld-Typ

Je nach Verwendungszweck des Feldes wird der Datenfeld-Typ gesetzt. Es gibt Text und Zahlenfelder. Die Textfelder sind rein für die Datenhaltung da, während mit Zahlenfelder gerechnet werden kann. Zu den Zahlenfelder zählen z. B. Ganzzahlen, Zahlen mit Nachkommastellen und Datumsfelder.

Folgende sind die geläufigsten Typen:
string - Buchstaben, Zahlen und Zeichen
int - Ganzzahlen
float - Zahlen mit Nachkommastellen

Datenfeld-Länge

Für eine optimierte Datenbank muss jedem Feld eine Länge mitgegeben werden. Beispielsweise benötigt eine ISBN-Nummer für Bücher 13 Stellen, eine Kundennummer eher weniger und als Buchtitel sollten genügend Zeichen zur Verfügung stehen.

Beispiel: Datensatzaufbau

Datenfeldname	Datenfeld-Typ	Datenfeld-Länge
titel	string	250
erscheinungsjahr	int	4
auflage	int	2
preis	float	10.2

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Aufgabe: Überlegen Sie, wie Sie theoretisch den Datensatz für folgenden Fall aufbauen würden

Sie möchten die URLs von Webseiten in einer Datenbank verwalten, auf die Sie einen Link gesetzt haben.

Sie benötigen:

1. ein Datenbankfeld für die URL, von der der Link abgeht,
2. ein Datenbankfeld für die URL, auf die der Link zeigt
3. ein Datum und Uhrzeit, wann der Link gesetzt wurde

Wie benennen Sie diese Felder, welche Größer erhalten die Datenbankfelder und was für einen Datentyp würde Sie verwenden?

Entity Relationship Modell (ERM)

Um die Strukturen von Daten und den Datenbankaufbau abzubilden, gibt es das Entity Relationship Modell (ERM). Durch das Modell kann die Struktur einfach anderen aufgezeigt werden (und gegebenenfalls in Arbeitsgruppen diskutiert werden). Es ist also ein konzeptionelles Arbeitsmittel.

Aber selbst wenn Sie alleine eine Datenbankanwendung erstellen, kann dieses Modell hilfreich sein, wenn Sie später sich nochmals schnell den Aufbau vor Augen führen möchten.

Grundbegriffe: Entity, Relationship und Attribute

Folgende Begriffe finden in der ERM Anwendung:

Entity:

Entität ist ein Objekt der modellierten Welt, das i.d.R. ein Gegenstand der realen Welt abbildet, wie beispielweise ein Buch (für eine Bücherverwaltung), eine Person (für die Adressverwaltung), eine Zeugnis-Note (für die Notenverwaltung).

Relationship:

Stellt die Beziehung zwischen Entitäten her, beispielweise das Buch wurde von dem Autor Schmidts geschrieben.

Attribut:

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Jede Entität hat verschiedene Attribute, z. B. ein Buch hat ein Erscheinungsdatum, einen Verlag, eine Auflage.

Grafische Darstellung vom Entity Relationship Modell (ERM)

Um die Zusammenhänge zwischen Entitäten, Beziehungen und Attribute darzustellen, werden folgende Symbole verwendet:

Entity: ein Rechteck

Attribute: werden unter dem Rechteck in Rechtecken mit gerundeten Ecken angezeigt

Relationship: Die Beziehungen werden anhand von Rauten zwischen den Entitäten angezeigt.

Aufgabe: werden Sie zum Immobilienmakler ...

bilden Sie Ihre Immobilien als ERM-Modell ab. Was in der Realität zu einer Immobilie gehört, können Sie sich unter den bekannten Immobilienseiten ansehen oder einfach mal in Google nach "Haus kaufen" suchen.

Sie haben als Entität also das Grundstück und als weitere Entität das jeweilige Haus. Bilden Sie die Attribute/Relationship ab.

Normalisierung – logische Regeln zur Minimierung von Redundanz

Durch die Normalisierung und die strengen Regeln soll eine korrekte, relationale Datenbank aufgebaut werden bzw. diese erhalten bleiben. Dabei ist wichtig, dass Redundanzen vermieden werden, da diese sonst schnell bei Änderungen von Inhalten zu Inkonsistenzen führen.

Dabei bedeutet **Redundanz** (lateinisch redundare „im Überfluss vorhanden sein“) auf Deutsch **Doppelung bzw. Überschneidungen**.

Inkonsistenzen bedeutet: **Widersprüchlichkeit oder Unbeständigkeit** der eingegebenen Daten.

Als Beispiel: Mitarbeiter pflegen die Daten der Kundendatenbank. Dazu kann bei jedem Kunden die PLZ und der Ort unabhängig voneinander eingetragen werden.

Der erste Mitarbeiter trägt nun einmal als PLZ „72070“ und als Ort „Tübingen“ ein. Der zweite Mitarbeiter als PLZ „72070“ und als Ort „Tübingen am Neckar“ ein – es liegt schon eine Inkonsistenz vor. Der nächste Mitarbeiter trägt für den nächsten Kunden als PLZ wieder „72070“ ein und als Ort dann „Tuebingen“. Und der vierte Mitarbeiter trägt (weil er eine Großschreiballergie hat, dann als Ort „tübingen“ ein.

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Am Abend kommt der Chef und lässt sich einer Statistik ausgeben, wie viele Kunden aus „Tübingen“ eingetragen wurden – er bekommt nur einen. Hätte er mit der PLZ „72070“ die Statistik erstellt, hätte er 4 Kunden angezeigt bekommen.

Dieses Beispiel zeigt, wie schnell eine Datenbank (bedingt durch Ihren Aufbau) zu unbeständigen Daten (sprich Inkonsistenzen) und den entsprechenden Folgeproblemen führen kann. Wäre hier die Redundanz (Eingabe von PLZ und zusätzlich Ort) unterbunden worden, wären Folgeprobleme vermieden worden.

Es gibt sechs Schritte, wobei in der Praxis die ersten drei umgesetzt werden.

Da die einzelnen Stufen der Normalisierung aufeinander aufbauen, muss die Reihenfolge der Anwendung der Normalisierung eingehalten werden. Es kann die 2. Normalisierung erst angewendet werden, wenn die 1. Normalisierung erfüllt ist.

Zweck der Normalisierung

Durch Anwendung der Normalisierung soll die Integrität der Daten sichergestellt werden.

- Redundanzen unterbinden
- Inkonsistenzen vermeiden

Die Wartung der Daten wird i.d.R. vereinfacht, die Programmierung allerdings aufwendiger.

Erste Normalform (1 NF)

Jedes Datenfeld darf nur **gleichartigen Inhalt** enthalten (Beispiel: aus einem Datenfeld „Name“ entstehen zwei Datenfelder: eines für den Vornamen und eines für den Nachnamen). Dies wurde im Kapitel „Daten strukturieren - Vorüberlegungen zur Datenbankerstellung“ als Inhalte trennen (atomisieren) bezeichnet.

Beispiel:

Aus dem Feld „name“ mit dem Inhalt „Erika Schmiedt“ werden die 2 Datenbankfelder „vorname“ und „nachname“. Das Feld „vorname“ bekommt als Inhalt „Erika“ und das Feld „nachname“ den Inhalt „Schmiedt“.

Mit gleichartig ist hier gemeint, dass im Feld „vorname“ dann durchaus auch 2 Vornamen auftauchen: „Karl Heinz“. Dies muss immer im Hinblick der zu erstellenden Anwendung geschehen – die kleinsten sinnvoll erscheinenden Bestandteile können sehr unterschiedlich sein. Bei einem Handyhändler kann die Telefonnummer durchaus in 2er Gruppen aufgeteilt angebracht sein, weil er aus den 2er Zahlenkombinationen Rückschlüsse ziehen kann – der Zoohandel ums Eck wird die Telefonnummer am Stück belassen.

Für die Tabelle muss ein Primärschlüssel vorhanden sein, damit jeder Datensatz eindeutig angesprochen werden kann.

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Zweite Normalform (2 NF)

Die erste Normalform muss erfüllt sein! Erst wenn die erste Normalform erfüllt ist, kann man an die Anwendung der zweiten Normalform gehen.

Die 2. Normalform besagt: Jeder Datensatz bildet **nur einen Sachverhalt** ab. Liegen in einer Tabelle Daten vor, die nicht nur 1 Sachverhalt abbilden, werden diese Daten in einzelne thematische Tabellen unterteilt.

In unserem Beispiel liegt die Kundendatenbank vor Anwendung der zweiten Normalform mit folgenden Inhalten vor:

Id	Vorname	Nachname	Auftragsnummer	Artikel
1	Axel	Pratzner	32482	Buch MySQL lernen
2	Axel	Pratzner	32482	DVD-Rohlinge
3	Elke	Schmidtz	32483	Buch MySQL lernen

Diese Tabelle bildet 3 Sachverhalte ab:

1. Kundendaten, 2. Artikeldaten und 3. Auftragsdaten

Nach der 2. Normalform muss hier eine Trennung in thematische Tabellen stattfinden. Wir teilen also die Tabelle in die 3 thematischen Tabellen „Kundendaten“, „Artikeldaten“ und „Auftragsdaten“

Tabelle Kundendaten:

Kunden-id	Vorname	Nachname
1	Axel	Pratzner
2	Elke	Schmidtz

Tabelle Artikeldaten:

Artikel-id	Artikel
1	Buch MySQL lernen
2	DVD-Rohlinge

Tabelle Auftragsdaten:

Bestell-id	Auftragsnummer	Kunden-id	Artikel-id
1	32482	1	1
2	32482	1	2
3	32483	2	1

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Dritte Normalform (3 NF)

Die erste und zweite Normalform muss erfüllt sein! Erst wenn die 1. und 2. Normalform erfüllt ist, kann man an die Anwendung der 3. Normalform gehen.

Bei der dritten Normalform geht es den indirekten (transitiven) Abhängigkeiten an den Kragen. In der Fachliteratur wird von transitiven Abhängigkeiten gesprochen.

Im folgenden Beispiel haben wir in der Tabelle neben Namen auch PLZ und Ort. Zu jedem Namen gehört eine PLZ und zu jeder PLZ gehört ein Ort. Der Ort ist also indirekt vom Namen abhängig.

Mathematisch ausgedrückt sieht das so aus:

„wenn NAME -> PLZ“ und „PLZ -> ORT“ dann „Name -> ORT“.

„Name -> ORT“ ist also eine transitive Abhängigkeit

Wir wollen aber in der Tabelle nur direkte Abhängigkeiten (also intransitive).

Als Tabelle **vor der Anwendung der 3. Normalform** hat unser Beispiel folgendes Aussehen:

Name	PLZ	Ort
Schmidtz	72074	Tübingen

Nach Anwendung der 3. Normalform haben wir 2 Tabellen:

Name	PLZ
Schmidtz	72074

Und

PLZ	Ort
72074	Tübingen

MySQL testen

Bevor wir nun MySQL nutzen, sollten wir überprüfen, ob die MySQL-Datenbank installiert ist und reagiert.

Lassen Sie sich nicht von den verschiedenen Möglichkeiten abschrecken. Die Variante über Commandline kann später wichtig werden. Bei der Installation von XAMPP sind Sie mit dem Prüfen schnell fertig, sollten trotzdem die anderen Möglichkeiten testen (Erfahrungen sammeln hilft im Leben).

MySQL-Test unter XAMPP

Dazu gibt es verschiedene Möglichkeiten. Wenn Sie XAMPP installiert haben, sollte MySQL mit installiert sein. Sie sehen, ob es läuft, indem Sie im Browser einfach <http://localhost/> aufrufen. Sie erhalten dann folgendes Bild, in dem wie im Beispiel nach Anklicken von Status auf der Inhaltsseite zu sehen ist, dass MySQL-Datenbank aktiviert ist.



Über die XAMPP Installation ist auch gleich ein Tool zur Verwaltung mit dem Namen phpMyAdmin (Bereich Tools in der Steuerung) installiert. Dies ist das Standardtool und ist

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

bei den meisten Providern (1&1, GONEO, Formativ) bereits für Sie installiert, wenn Sie in Ihrem Paket MySQL-Zugriff haben. Zu diesem Tool später mehr.

MySQL Test per phpinfo

Wenn Sie ein kleines PHP-Programm mit phpinfo (siehe [Erstes PHP Programm](#)) erstellen, scrollen Sie so weit nach unten, bis ein Bereich mit mysql (siehe Abbildung) auftaucht. Sie können auch danach suchen lassen mit STRG + F.

mysql

MySQL Support	enabled
Active Persistent Links	0
Active Links	0
Client API version	5.0.33

Directive	Local Value	Master Value
mysql.allow_persistent	On	On
mysql.connect_timeout	60	60
mysql.default_host	no value	no value
mysql.default_password	no value	no value
mysql.default_port	no value	no value
mysql.default_socket	no value	no value
mysql.default_user	no value	no value

Erscheint diese Ausgabe, wissen Sie, dass Ihrem PHP MySQL bekannt ist und so aller Wahrscheinlichkeit dieses auch funktioniert.

MySQL Test per Commandline

Auch per Commandline (DOS-Fenster aufrufen – geht mit Eingabeaufforderung starten und dort dann „CMD“ eingeben) kann MySQL getestet werden. Dieser Weg erscheint erstmal sehr mühsam, da nach dem Aufruf der Commandline in das Verzeichnis gewechselt werden muss, in dem die Binär-Daten von MySQL liegen und dort dann durch den Aufruf

```
mysql -u root -p
```

der MySQL-Monitor gestartet werden muss. Sobald der Monitor läuft, sehen Sie als Prompt-Zeichen die Ausgabe „mysql>“. Jetzt können Sie direkt MySQL-Befehle eingeben: Datenbanken anlegen, mit Inhalten füllen und auswerten. Diese Vorgehensweise kann sehr interessant sein, wenn Sie auf einem entfernten Server nur Zugang auf der Shell-Ebene haben.

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

```
C:\Windows\system32\cmd.exe - mysql -u root -p

C:\xampp\mysql\bin>mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 40
Server version: 5.0.33 Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> USE test01;
Database changed
mysql> SELECT * FROM adressen;
+----+-----+-----+
| id | name | plz  |
+----+-----+-----+
|  1 | Axel | 72070 |
|  2 | Elke | 76071 |
+----+-----+-----+
2 rows in set (0.00 sec)

mysql> _
```

In der Bildschirmkopie sehen Sie in Grün die Eingaben von Hand. Der Rest sind die Meldungen und Ausgaben vom Programm.

MySQL funktioniert – jetzt geht es los

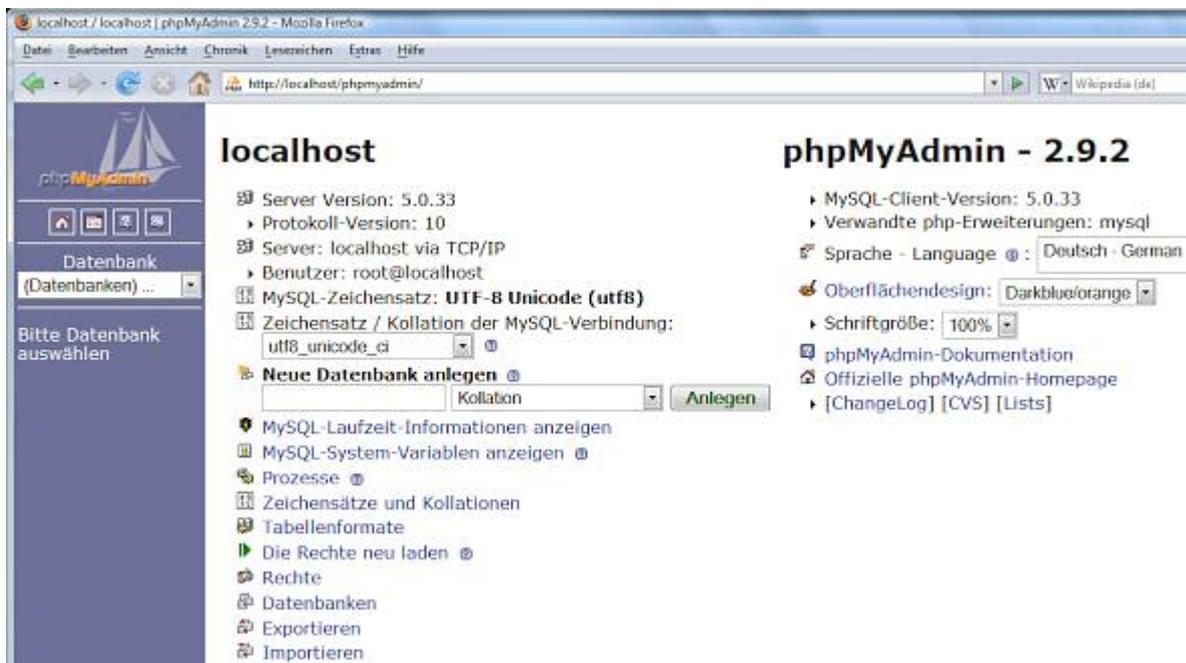
In den folgenden Kapiteln werden Sie nun eigene Datenbanken erstellen, füllen und auswerten. Dabei wird aus didaktischer Sicht alles in PHP gemacht, was man auch direkt über die Commandline oder über phpMyAdmin erledigen könnte. Es ist wichtig, den kompletten Durchgang einmal erlebt zu haben, bevor man sich das Leben einfach macht :)

MySQL-Anweisungen lernen - phpMyAdmin nutzen

Um effektiv MySQL zu lernen, ist das Tool MySQL nützlich. Dieses ist eigentlich für die Verwaltung von MySQL-Datenbanken gemacht und man kann direkt MySQL-Anweisungen ausführen lassen. Durch gut gemachte MySQL-Anweisungen kann man sich extrem viel Programmierarbeit in PHP sparen.

phpMyAdmin ist bei vielen Providern standardmäßig installiert – genauso auch bei unseren WAMP und XAMPP-Installationen.

Zum Starten dann <http://localhost/phpmyadmin/>



Wir legen nun eine neue Datenbank an mit dem Namen "adressen" – nach dem Anlegen sehen wir den entsprechenden SQL-Befehl `CREATE DATABASE 'adresse';` und auf der rechten Seite die neue Datenbank in der Datenbankliste auftauchen.

localhost

Server Version: 5.0.33
‣ Protokoll-Version: 10
Server: localhost via TCP/IP
‣ Benutzer: root@localhost
MySQL-Zeichensatz: **UTF-8 Unicode (utf8)**
Zeichensatz / Kollation der MySQL-Verbindung:
utf8_unicode_ci

Neue Datenbank anlegen
adressen Kollation **Anlegen**

Wie gehen nun auf den Reiter SQL

The screenshot shows the phpMyAdmin interface. The top navigation bar includes 'Struktur', 'SQL', 'Suche', 'Abfrageeditor', and 'Exportieren'. The main content area displays a message: 'Datenbank adressen wurde erzeugt.' Below this, the 'SQL-Befehl:' field contains the command: `CREATE DATABASE `adressen` ;`. A message below the command states: 'Es wurden keine Tabellen in der Datenbank gefunden.' At the bottom, there is a section for creating a new table: 'Neue Tabelle in Datenbank adressen erstellen' with input fields for 'Name:' and 'Anzahl der Felder:'.

Um effektiv mit MySQL-Befehlen zu spielen, gibt es hier Spieldaten mit 400 Testeinträgen. Datei laden und im Feld "SQL-Befehle ..." einfügen und ausführen lassen. Die Testadressen zum Download sind zu finden unter:

<http://www.iwke.de/kunden/php-kurs/downloads/testadressen.txt>

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen



Der nun neue Inhalt der Tabelle testadressen kann über das erste Symbol angezeigt werden.



Ihr SQL-Befehl wurde erfolgreich ausgeführt.

Tabelle	Aktion	Einträge	Typ
<input type="checkbox"/> testadressen		400	MyISAM
1 Tabellen	Gesamt	400	MyISAM

Alle auswählen / Auswahl entfernen
 markierte:

Wir bekommen eine Auflistung der gesamten Daten (automatisch gestückelt). Der entsprechende SQL-Befehl wäre `SELECT * FROM testadressen`

Server: localhost > Datenbank: adressen > Tabelle: testadressen

Anzeigen Struktur SQL Suche Einfügen Exportieren Importieren Operationen Leeren Löschen

Zeige Datensätze 0 - 19 (400 insgesamt, die Abfrage dauerte 0.0006 sek.)

SQL-Befehl:

```
SELECT *
FROM `testadressen`
LIMIT 0, 20
```

[Bearbeiten] [SQL erklären] [PHP-Code erzeugen] [Aktualisieren]

Operationen für das Abfrageergebnis

Druckansicht Druckansicht (vollständige Textfelder) Exportieren

Zeige: 20 Datensätze, beginnend ab 20
 untereinander angeordnet und wiederhole die Kopfzeilen nach 100 Datensätzen.

Nach Schlüssel sortieren: keine OK

	id	nachname	vorname	anrede	land	geburtsdatum	motto	lieblingsfarbe	familiens
<input type="checkbox"/>	1	Nola	Vazquez	Frau	Bahamas	1976-01-19	volutpat ornare, facilisis eget, ipsum. Donec soll...	grün	verheirate
<input type="checkbox"/>	2	Rudyard	Mcfarland	Frau	Hungary	1977-09-23	ipsum cursus	schwarz	verheirate

Geben wir nun direkt MySQL-Befehle ein – im Folgenden werden nur die Spalten „anrede, vorname, nachname, geburtsdatum“ angezeigt

Server: localhost > Datenbank: adressen > Tabelle: testadressen

Anzeigen Struktur SQL Suche Einfügen Exportieren Importieren Operationen Leeren Löschen

SQL-Befehl(e) in Datenbank adressen ausführen:

```
SELECT `anrede`, `vorname`, `nachname`, `geburtsdatum` FROM `testadressen`
```

Felder
 land
 geburtsdatum
 motto
 lieblingsfarbe
 familienstand

SQL-Abfrage speichern: Diese gespeicherte SQL-Abfrage für jeden Benutzer verfügbar machen
 Überschreibe gleichnamiges Bookmark

[Begrenzer:] SQL-Befehl hier wieder anzeigen OK

```
SELECT anrede, vorname, nachname, geburtsdatum FROM testadressen
```

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Server: localhost ▶ Datenbank: adressen ▶ Tabelle: testadr

Anzeigen Struktur SQL Suche Einfügen Exportieren

Löschen

Zeige Datensätze 0 - 19 (400 insgesamt, die Abfrage dauerte 0.0005 s)

SQL-Befehl:

```
SELECT `anrede`, `vorname`, `nachname`, `geburtsdatum`
FROM `testadressen`
LIMIT 0, 20
```

[Bearbeiten]

Operationen für das Abfrageergebnis

Druckansicht Druckansicht (vollständige Textfelder) Exportieren

Zeige: Datensätze, beginnend ab untereinander angeordnet und wiederhole die k Datensätzen.

Nach Schlüssel sortieren:

			anrede	vorname	nachname	geburtsdatum
<input type="checkbox"/>			Frau	Vazquez	Nola	1976-01-19
<input type="checkbox"/>			Frau	Mcfarland	Rudyard	1977-09-23
<input type="checkbox"/>			Familie	Baker	Scott	1939-12-11
<input type="checkbox"/>			Frau	Eaton	Elizabeth	1963-03-11
<input type="checkbox"/>			Frau	Vang	Drake	1952-12-17
<input type="checkbox"/>			Herr	Swanson	Kerry	1929-06-27
<input type="checkbox"/>			Familie	Manning	Morgan	1922-06-14

Und nun mit Bedingungen – im Folgenden nur alle Frauen.

Zeige Datensätze 0 - 19 (144 insgesamt, die Abfrage dauerte 0.0006 s

SQL-Befehl:

```
SELECT `anrede` , `vorname` , `nachname` , `geburtsdatum`  
FROM `testadressen`  
WHERE `anrede` = 'Frau'  
LIMIT 0 , 20
```

[[Bearbeiten](#)]

Operationen für das Abfrageergebnis

 Druckansicht  Druckansicht (vollständige Textfelder)  Exp

Zeige : Datensätze, beginnend ab
 angeordnet und wiederhole die k
Datensätzen.

Nach Schlüssel sortieren:

	<u>anrede</u>	<u>vorname</u>	<u>nachname</u>	<u>geburtsdatum</u>
<input type="checkbox"/>  	Frau	Vazquez	Nola	1976-01-19
<input type="checkbox"/>  	Frau	Mcfarland	Rudyard	1977-09-23
<input type="checkbox"/>  	Frau	Eaton	Elizabeth	1963-03-11
<input type="checkbox"/>  	Frau	Vann	Drake	1952-12-17

MySQL-SELECT-Anweisung - Varianten und Möglichkeiten

Aufbau der SELECT-Anweisung

Alle Angaben in [] sind optional.

```
SELECT
  [rechenoperation|funktionsauswahl]
  spaltenname
FROM tabelle
  [WHERE bedingungen]
  [GROUP BY spalten]
  [HAVING aggregatfunktionen]
  [ORDER BY spalte [ASC] [DESC]]
  [LIMIT [Anfang][Zeilenanzahl]]
```

Auswahl aller Spalten der Tabelle testadressen

```
SELECT
  *
FROM testadressen;
```

nur Ausgaben bestimmter Spalten

```
SELECT
  anrede, vorname, nachname
FROM testadressen;
```

Spaltenüberschriften festlegen

Spaltenüberschriften können umdefiniert werden. Dies geschieht über diesen Aufbau:

```
SELECT
  `nachname` ,
  `vorname` ,
  geburtsdatum AS 'Geburtstag'
FROM testadressen
```

Reihenfolge bestimmen

Sortieren nach Spalten

```
SELECT
  `nachname` , `vorname`
FROM testadressen
ORDER BY nachname
```

Sortieren nach 2 Spalten

```
SELECT
  `nachname` , `vorname`
FROM testadressen
ORDER BY nachname, vorname
```

Sortierrichtung ändern (eine Spalte aufsteigend, zweite Spalte absteigend)

```
SELECT
  `nachname` , `vorname`
FROM testadressen
ORDER BY nachname DESC , vorname
```

Bedingungen setzen – WHERE ...

```
SELECT
  `nachname` , `vorname`
FROM testadressen
WHERE vorname = 'Fischer'
```

Bedingungen verknüpfen (ODER)

```
SELECT
  `nachname` , `vorname` , `anrede` , `familienstand`
FROM testadressen
WHERE anrede = 'Herr' OR familienstand = 'single'
```

Bedingungen verknüpfen (UND)

```
SELECT
  `nachname` , `vorname` , `anrede` , `familienstand`
FROM testadressen
WHERE anrede = 'Frau' AND familienstand = 'single'
```

Bedingungen nach Vergleichsoperatoren

Operator	Funktion
=	gleich
<> oder !=	ungleich
<	kleiner als
>	größer als
<=	kleiner gleich
>=	größer gleich

Alle Personen mit Geburtstag vor 01.01.1940

```
SELECT
  `nachname`, `vorname`, `anrede`, `familienstand`, `geburtsdatum`
FROM testadressen
WHERE geburtsdatum < '1940-01-01'
```

Alle Personen, deren Nachnamen vor C kommt

```
SELECT
  `nachname`, `vorname`, `anrede`
FROM testadressen
WHERE nachname < 'C'
```

WHERE-Bedingung mit Klammern – alle Personen mit der Lieblingsfarbe Schwarz ODER Geburtstag zwischen 01.01.1970 und 31.12.1970

```
SELECT
  `nachname`, `vorname`, `anrede`, `geburtsdatum`, `lieblingsfarbe`
FROM testadressen
WHERE `lieblingsfarbe` = 'schwarz'
OR (
  `geburtsdatum` >= '1970-01-01'
AND `geburtsdatum` <= '1970-12-31'
)
```

Aggregatsfunktionen

Können nur auf eine einzige Spalte angewendet werden.

MIN (Spalte)	Minimum (Nummer, Zeichen, Datum)
MAX (Spalte)	Maximum (Nummer, Zeichen, Datum)
COUNT(*)	Anzahl (Nummer, Zeichen, Datum)
SUM (Spalte)	Summe (Numerisch)
AVG (Spalte)	Mittelwert (Numerisch) - (engl. average)

Durchschnitt vom Gehalt

```
SELECT
  AVG( gehalt )
FROM testadressen
```

Anzahl von Datensätzen

```
SELECT
  COUNT(*)
FROM `testadressen`
```

Anzahl von Datensätzen der Singles

```
SELECT
  COUNT(*)
FROM `testadressen`
WHERE `familienstand`='single'
```

Verdienst aller Singles und Anzahl der Singles

```
SELECT
  COUNT(*), SUM(`gehalt`)
FROM `testadressen`
WHERE `familienstand` = 'single'
```

Datumsfunktionen

Alle Personen, die im Jahr 1969 geboren sind

```
SELECT
  `nachname`, `vorname`, `anrede`, `geburtsdatum`
FROM testadressen
WHERE YEAR( geburtsdatum ) = 1969
```

Alle Sonntags-Kinder

```
SELECT
  `nachname` , `vorname` , `anrede` , `geburtsdatum`
FROM testadressen
```

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

```
WHERE WEEKDAY( geburtsdatum ) = 6
```

Alle Oktober-Kinder

```
SELECT
  `nachname` , `vorname` , `anrede` , `geburtsdatum`
FROM testadressen
WHERE MONTH( geburtsdatum ) = 10
```

Alle Personen, die zwischen 1969 und 1972 geboren wurden

```
SELECT
  `nachname` , `vorname` , `anrede` , `geburtsdatum`
FROM testadressen
WHERE YEAR( geburtsdatum ) >= 1969
  AND YEAR( geburtsdatum ) <= 1972
```

Gelebte Tage seit Geburtstag bis zum 30.7.2007

```
SELECT
  `nachname` , `vorname` ,
  TO_DAYS( '2007-07-30' ) - TO_DAYS( geburtsdatum )
FROM testadressen
```

Gelebte Tage seit Geburtstag bis zum heutigen Tag mit geänderten Spaltenüberschrift. (heutiger Tag = engl. current date (CurDate))

```
SELECT
  `nachname` ,
  `vorname` ,
  TO_DAYS( CurDate() ) - TO_DAYS( geburtsdatum ) AS 'gelebte Tage bis heute'
FROM testadressen
```

Datumsfunktionen

Funktion	Ergebnis
WEEKDAY(datum)	0-6 (Montag bis Sonntag)
DAYofWEEK (datum)	1-7 (Sonntag bis Samstag)
DAYNAME (datum)	Sunday-Saturday
WEEK (datum)	Kalenderwoche (Wochenbeginn Sonntag)
WEEK (datum, 1)	Kalenderwoche (Wochenbeginn Montag)
WEEK (datum, 2)	Kalenderwoche (Wochenbeginn Dienstag)
DAYOFMONTH (datum)	Tageswert des Monats
MONTH (datum)	Monatswert des Datums
DAYOFYEAR (datum)	Tag des Datums ab dem 1.1.
QUARTER (datum)	Quartal des Datums (1 bis 4)
YEAR (datum)	Jahrewert des Datums

Dies und das

Inhaltlich gleiche Werte ausschließen (jeden Wert nur einmal)

```
SELECT
  DISTINCT `lieblingsfarbe`
FROM testadressen
```

Verbindung zur MySQL-Datenbank herstellen

Um die MySQL Datenbank nutzen zu können, muss eine Verbindung zwischen PHP und MySQL hergestellt werden. Dazu werden folgende Angaben benötigt (die man vom Provider erhält).

Host (MYSQL_HOST):

Der Server, auf dem sich die Datenbank befindet. Technisch ist es egal, ob die Datenbank sich auf dem gleichen Server befindet, wo auch Ihre Website liegt, oder auf einem anderen. Ist diese auf dem gleichen Server, dann lautet die Angabe "localhost" – ist diese auf einem anderen Server, dann irgendwas in der Art "db37___23.1und1.de" oder auch eine IP-Adresse.

Benutzername (MYSQL_BENUTZER)

Benutzername und Kennwort verhindern, dass sich Unbefugte einloggen.

Kennwort (MYSQL_KENNWORT)

Kennwort zur Sicherheit, niemals aus der Hand geben – wenn jemand an Ihre Datenbanken kommt, kann er diese nach Belieben auslesen, ändern und löschen!

Diese Angaben benötigen Sie bei jedem Verbindungsaufbau. Daher ist es üblich, diese Informationen als Konstanten zu definieren und in einer Datei auszulagern und diese dann per `require_once` einzubinden. Speichern Sie also folgenden Quellcode unter dem Namen "konfiguration.php" - für spätere Beispiele wird der Verbindungsaufbau zu Datenbanken benötigt.

```
<?php
// die Konstanten auslagern in eigene Datei, z. B. konfiguration.php
// die dann per require_once ('konfiguration.php'); geladen werden.
// Damit alle Fehler angezeigt werden
error_reporting(E_ALL);
// Zum Aufbau der Verbindung zur Datenbank
// die Daten erhalten Sie von Ihrem Provider
define ( 'MYSQL_HOST', 'localhost' );
// bei XAMPP ist der MYSQL_Benutzer: root
define ( 'MYSQL_BENUTZER', '' );
define ( 'MYSQL_KENNWORT', '' );
// für unser Bsp. nennen wir die DB adressverwaltung
define ( 'MYSQL_DATENBANK', 'adressverwaltung' );
?>
```

Der Aufbau der Verbindung erfolgt nun über: `mysql_connect`

```
<?php
```

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

```
require_once ('konfiguration.php');
$db_link = mysql_connect (MYSQL_HOST, MYSQL_BENUTZER, MYSQL_KENNWORT);
?>
```

Die Rückgabe der Funktion `mysql_connect` wird in `$db_link` gespeichert. Diese enthält im Fall, dass es ein Problem gab, "FALSE", ansonsten wenn die Verbindung geklappt hat, die Verbindungskennung.

Zum Lernen von MySQL-Anweisungen sollten Sie sich Kontrollausgaben ausgeben – später ist es umso wichtiger, dass im Falle eines Problems ein sauberer Abbruch erfolgt, der Programmierer per E-Mail benachrichtigt wird und der Fehler gespeichert wird.

Unser komplettes Programm (ohne Auslagerung der Zugangsdaten)

```
<?php
error_reporting(E_ALL);

// Zum Aufbau der Verbindung zur Datenbank
define ( 'MYSQL_HOST', 'localhost' );
define ( 'MYSQL_BENUTZER', '' );
define ( 'MYSQL_KENNWORT', '' );
define ( 'MYSQL_DATENBANK', 'adressverwaltung' );

$db_link = mysql_connect (MYSQL_HOST, MYSQL_BENUTZER, MYSQL_KENNWORT);

if ( $db_link )
{
echo 'Verbindung erfolgreich: ';
echo $db_link;
}
else
{
// hier sollte dann später dem Programmierer eine
// E-Mail mit dem Problem zukommen gelassen werden
die('keine Verbindung möglich: ' . mysql_error());
}
mysql_close($db_link );
?>
```

Datenbank anlegen

Legen wir nun unsere Adress-Datenbank an mit dem Namen: `adressverwaltung`

```
// Datenbank-Verbindung herstellen
// siehe (mysql-datenbank-verbindung-herstellen.htm)
require_once ('konfiguration.php');
// zuweisen der MySQL-Anweisung einer Variablen
$sql = CREATE DATABASE `adressverwaltung` ;
$result = mysql_query($sql)
or die("Anfrage fehlgeschlagen: " . mysql_error());
```

Nachdem die Datenbank-Verbindung hergestellt ist, muss mitgeteilt werden, welche Datenbank im PHP-Programm verwendet werden soll (es können ja beliebig viele angelegt werden).

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

```
// Datenbank-Verbindung herstellen
// siehe (mysql-datenbank-verbindung-herstellen.htm)
require_once ('konfiguration.php');
// Nutzen von Datenbank (Name ist hinterlegt in Konstante MYSQL_DATENBANK)
mysql_select_db( MYSQL_DATENBANK )
or die("Auswahl der Datenbank fehlgeschlagen");
```

Nachdem wir die Datenbank erzeugt haben, erhält die Datenbank eine Tabelle mit der Bezeichnung "adressen" – diese hat verschiedene Felder. Beides legen wir nun in einem Schritt an.

Wir legen die Datenbank-Felder: "id, nachname, vorname, akuerzel, strasse, plz, telefon" an.

Der Aufbau der einzelnen Zeilen

Name des Datenfeldes: Beispiel id, nachname, vorname
Immer klein geschrieben, ohne Leerzeichen und Sonderzeichen

Art des Feldes:
INT – nur Ganzzahlen
VARCHAR – Buchstaben und Zahlen

Zahl in Klammer
Wie groß das Feld maximal ist.

NOT NULL oder NULL
Ob Eingaben bei diesem Feld notwendig sind oder es auch leer gespeichert werden darf.

AUTO_INCREMENT
Das Feld erhält automatisch eine eindeutige Nummer, die bei jedem neuen Feld automatisch erhöht wird. Ideal für Primärschlüssel.

PRIMARY KEY
Dieses Feld stellt den Primärschlüssel dar.

```
// Datenbank-Verbindung herstellen
require_once ('konfiguration.php');
// Nutzen von Datenbank (Name ist hinterlegt in Konstante MYSQL_DATENBANK)
mysql_select_db( MYSQL_DATENBANK )
or die("Auswahl der Datenbank fehlgeschlagen");
// MySQL-Befehl der Variablen $sql zuweisen
$sql = "
CREATE TABLE `adressen` (
`id` INT( 10 ) NOT NULL AUTO_INCREMENT PRIMARY KEY ,
`nachname` VARCHAR( 150 ) NOT NULL ,
`vorname` VARCHAR( 150 ) NULL ,
`akuerzel` VARCHAR( 2 ) NOT NULL ,
`strasse` VARCHAR( 150 ) NULL ,
`plz` INT( 5 ) NOT NULL ,
`telefon` VARCHAR( 20 ) NULL
) ENGINE = MYISAM ;
";
// MySQL-Anweisung ausführen lassen
```

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

```
$db_erg = mysql_query($sql) or die("Anfrage fehlgeschlagen: " . mysql_error());
```

Die Datenbank und die Tabelle sind nun angelegt - die Struktur ist somit festgelegt. Jetzt müssen diese mit Inhalte gefüllt werden.

MySQL-Tabelle mit Inhalt füllen

Und nun füllen wir die Tabelle mit Inhalt:

Vor unserem "INSERT"-Befehl muss natürlich eine Verbindung zur Datenbank hergestellt und die gewünschte Datenbank ausgewählt werden.

```
$sql = "
INSERT INTO `adressen`
(
`id` , `nachname` , `vorname` , `akuerzel` , `strasse` , `plz` , `telefon`
)
VALUES
(
NULL , 'Pratzner', 'Axel', 'm', NULL , '72070', '07071-77..'
);
";
$db_erg = mysql_query($sql)
    or die("Anfrage fehlgeschlagen: " . mysql_error());
```

Unsere erste Datenbank mit Tabelle und Inhalt ist erzeugt, mit der wir nun spielen können. Geben Sie mindestens noch 2 weitere Datensätze ein, damit wir für den nächsten Schritt auch etwas zum ausgeben und auswerten haben.

Datenbank auslesen und anzeigen

Mit folgenden Programm wird die Verbindung zur Datenbank hergestellt, die MySQL-Query übergeben und die Inhalte der Rückgabe in einer Tabelle dargestellt.

```
<?php
require_once ('konfiguration.php');
$db_link = mysql_connect (MYSQL_HOST, MYSQL_BENUTZER, MYSQL_KENNWORT);

// Nutzen von Datenbank (Name ist hinterlegt in Konstante MYSQL_DATENBANK
$db_sel = mysql_select_db( MYSQL_DATENBANK )
or die("Auswahl der Datenbank fehlgeschlagen");

$sql = "
SELECT * FROM adressen
";

$db_erg = mysql_query( $sql );
if ( ! $db_erg )
{
die('Ungültige Abfrage: ' . mysql_error());
}

echo '<table border="1">';
while ($zeile = mysql_fetch_array( $db_erg, MYSQL_ASSOC))
```

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

```

{
echo "<tr>";
echo "<td>". $zeile['id'] . "</td>";
echo "<td>". $zeile['nachname'] . "</td>";
echo "<td>". $zeile['vorname'] . "</td>";
echo "<td>". $zeile['akuerzel'] . "</td>";
echo "<td>". $zeile['strasse'] . "</td>";
echo "<td>". $zeile['plz'] . "</td>";
echo "<td>". $zeile['telefon'] . "</td>";
echo "</tr>";
}
echo "</table>";

mysql_free_result( $db_erg );
?>

```

Relationen - Tabellen verbinden

Anhand einer Adressverwaltung sehen wir uns am praktischen Beispiel Relationen an und wie Tabellen miteinander verbunden werden.

Aufgabe: Erstellen Sie ein Adressbuch als PHP-Anwendung

Erstellen Sie ein Adressbuch, in dem Sie Adressen eintragen, ändern und löschen können. Als Beispiel, was an Daten praktisch sein könnte, hier (m)eine Visitenkarte.



Bitte beim Aufbau der Struktur der Datenbank an die Grundprinzipien für Datenbankgestaltung denken: Inhalte trennen (atomisieren), eindeutige Zugriffsmöglichkeiten

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

schaffen, keine Datenredundanz (Mehrfachspeicherung). <http://www.php-kurs.com/daten-strukturieren-fuer-datenbank.htm>

Datenbankaufbau Adressverwaltung

Als Vorschlag können Sie folgenden Aufbau nutzen (bzw. vergleichen mit der von Ihnen erstellten Lösung).

Als **Tabellenname: adressen** (das Kind sollte man so benennen, dass eindeutig hervorgeht, welche Daten sich dahinter verstecken)

id	für die eindeutige Zugriffsmöglichkeit
geschlecht	dieses wird hier mit f für female und m für male eingetragen. Aus dieser Angabe können wir dann problemlos Anreden ("Sehr geehrte Frau ...") erzeugen
nachname, vorname, strasse	irgendwie selbstredend
plz	5 stellig – um das Beispiel einfach zu halten, werden wir nur 5 stellige Postleitzahlen erfassen
email	die E-Mail-Adresse der Person
url	die WWW-Adresse, falls die Person eine Website hat

Als fertiges MySQL-Statement sieht das wie folgt aus:

```
CREATE TABLE `adressen` (  
  `id` INT( 10 ) NOT NULL AUTO_INCREMENT PRIMARY KEY ,  
  `geschlecht` CHAR( 1 ) NOT NULL ,  
  `nachname` VARCHAR( 255 ) NOT NULL ,  
  `vorname` VARCHAR( 255 ) NOT NULL ,  
  `strasse` VARCHAR( 200 ) NOT NULL ,  
  `plz` INT( 5 ) NOT NULL ,  
  `tel` VARCHAR( 100 ) NOT NULL ,  
  `email` VARCHAR( 150 ) NOT NULL ,  
  `url` VARCHAR( 150 ) NOT NULL  
  ) ENGINE = MYISAM ;
```

Wo bleibt der Ort?

Für den Ort erstellen wir eine eigene Tabelle. Dies macht Sinn, um Datenredundanz (Mehrfachspeicherung) zu vermeiden. In dieser Tabelle sind zu den entsprechenden PLZ die Orte eingetragen.

Da die PLZ eindeutig ist, wird diese zu unserem Index und kann als Suchkriterium dienen.

```
CREATE TABLE `orte` (  
  `plz` INT( 5 ) NOT NULL ,  
  `ort` VARCHAR( 255 ) NOT NULL ,
```

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

```
PRIMARY KEY ( `plz` )  
) ENGINE = MYISAM ;
```

Tragen Sie nun 1-2 Adressen von Hand über phpMyAdmin ein (auch die PLZ und den Ort in der entsprechenden Tabelle).

Verbinden beider Tabellen

Für die Auswertung und Nutzung in unser Adressauflistung müssen wir die Tabelle "adressen" mit der Tabelle "orte" verbinden.

Als erstes selektieren wir alle Spalten der Adress-Tabelle:

```
SELECT * FROM adressen
```

Zum Verbinden benötigen wir den grundlegenden Aufbau:

```
LEFT JOIN tabellenname ON Relation
```

Relation: Nun müssen wir für MySQL die Zusammenhänge zwischen den Tabellen aufbauen. Dies geschieht über eine Relation. Relationen beschreiben i.A. Beziehungen zwischen Gegenständen bzw. Objekten. In Tabellen also wird die Beziehung zwischen 2 Tabellen beschrieben.

Unsere Relation ist:

```
adressen.plz = orte.plz
```

Und das Ganze nun als MySQL-Statement:

```
SELECT * FROM adressen  
LEFT JOIN orte ON adressen.plz = orte.plz
```

Als Ergebnis erhalten wir nun:

SQL-Befehl:

```
SELECT *  
FROM adressen  
LEFT JOIN orte ON adressen.plz = orte.plz  
LIMIT 0 , 30
```

[Bearbeiten] [SQL erklären] [PHP-Code erzeugen] [Aktualisieren]

Operationen für das Abfrageergebnis

 Druckansicht  Druckansicht (vollständige Textfelder)  Exportieren

Zeige : Datensätze, beginnend ab
 angeordnet und wiederhole die Kopfzeilen nach
Datensätzen.

id	geschlecht	nachname	vorname	strasse	plz	tel	email	url	plz	ort
1	m	Pratzner	Axel		72070				72070	Tübingen
2	f	Fiction	Elke		77777				NULL	NULL

Hier sieht man, dass ist das bei der ID 2 unter den Feldern "plz" und "ort" NULL ausgegeben wird. Hier greift die Relation ins "Leere". In der Datenbank mit den Orten ist kein Eintrag hinterlegt für die Postleitzahl "77777".

Möchte man (was bei diesem Beispiel wenig Sinn macht, aber der Vollständigkeit halber) keine NULL-Werte, können diese Sätze ausgeklammert werden.

```
SELECT * FROM adressen  
LEFT JOIN orte ON adressen.plz = orte.plz  
WHERE adressen.plz = orte.plz
```

Wenn nicht alle Felder ausgegeben werden sollen, kann dies fast wie gewohnt geschehen, wenn anstatt dem * die gewünschten Felder angegeben werden, die angezeigt werden sollen.

```
SELECT vorname  
FROM adressen  
LEFT JOIN orte ON adressen.plz = orte.plz
```

Um Fehlermeldungen vorzubeugen, falls derselbe Feldname in beiden Tabellen vorkommt, sollten Sie noch angeben, aus welcher Tabelle das Feld stammt.

```
SELECT adressen.vorname, orte.plz  
FROM adressen  
LEFT JOIN orte ON adressen.plz = orte.plz
```

Ansonsten können Sie schnell eine Fehlermeldung dieser Art bekommen:

```
MySQL meldet:  
#1052 - Column 'plz' in field list is ambiguous
```

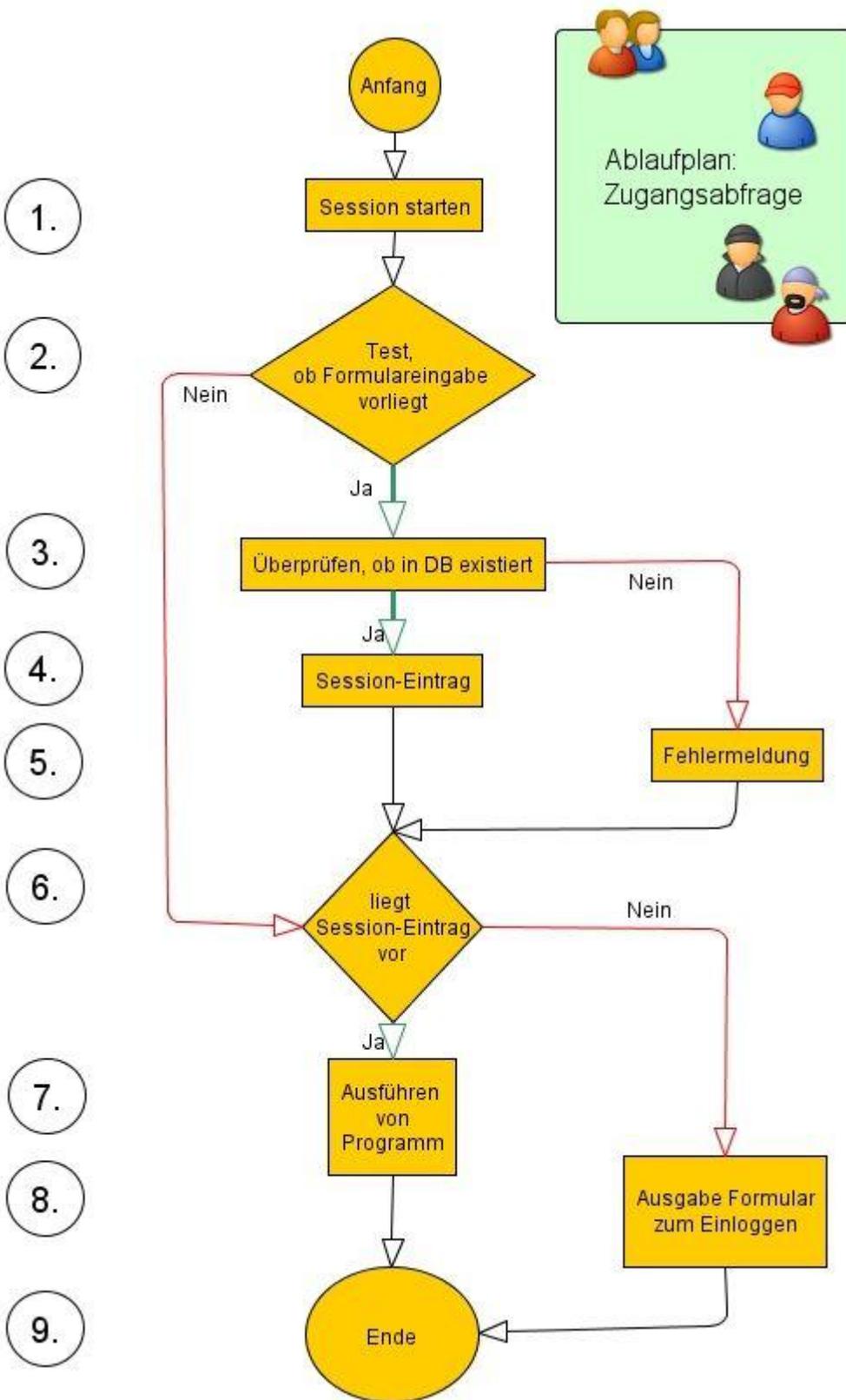
Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Erstellen Sie die Anwendung – beachten Sie, dass bei der Neuanlage einer Adresse in der Tabelle "orte" überprüft werden muss, ob die PLZ bereits eingetragen ist. Wenn nicht, dann diese in der Tabelle "orte" eintragen (natürlich mit Ort).

Tipp zum Überprüfen: Zählen Sie einfach die Ergebnisse bei der Bedingung "... WHERE plz=' \$plz' "

Zugangskontrolle mit Passwortchutz

Für einen geschlossenen Benutzerkreis wird eine Zugangskontrolle benötigt. Als erstes Beschreibe ich den Ablauf anhand des Ablaufdiagrammes Schritt für Schritt. Erst dann werden wir Schritt für Schritt das Programm umsetzen.



Schritt 1: Session starten

In der Session sind nach erfolgreichem Einloggen die Zugangsdaten vermerkt. Dadurch kann der Zustand sich „gemerkt“ werden. Ohne die Sessions wüsste das Programm beim nächsten Start nicht, ob der Benutzer sich bereits erfolgreich eingeloggt hat oder ob es sich um einen neuen Besucher handelt.

Schritt 2: Test, ob eine Formulareingabe vorliegt

Im Array `$_POST`, bzw. `$_GET` sind die Inhalte von einem abgesendeten Formular abfragbar. Wenn die entsprechenden Einträge vorhanden sind, geht es weiter in Punkt 3 mit gegenprüfen gegen Datenbank. Sind keine Formulareingaben vorhanden, springen wir direkt weiter zu Punkt 6.

Schritt 3: Gegenprüfen zu Datenbank-Einträgen

Hier werden nun die Formulareingaben gegen die Datenbank geprüft. Das besondere ist, dass das Kennwort in der Datenbank md5-verschlüsselt abgespeichert ist. Dadurch kann niemand (auch nicht der Admin) die Kennwörter nutzen, da das verschlüsseln ein Einwegvorgang ist. Das bedeutet, dass die Benutzereingabe auch wieder mit md5 verschlüsselt wird und das Ergebnis der Verschlüsselung mit dem eingetragenen md5-Kennwort verglichen wird. Sind beide identisch, dann war es das richtige Passwort. Als Ergebnis sollte nur 1 Datensatz zurück kommen!

Schritt 4: Nutzer hat sich korrekt authentifiziert

Wenn nur 1 Datensatz zurückkommt, hat der Vergleich gepasst. Die Daten des Nutzers werden in der Session hinterlegt, damit er auch für alle anderen Seiten als eingeloggt gilt.

Schritt 5: Fehlermeldung wenn nicht in Datenbank

Falls die Eingaben des Formulars zu keinem Ergebnis geführt haben, dann hat sich entweder der Nutzer vertippt oder es gibt ihn gar nicht (in der Datenbank). Es wird eine Fehlermeldung am Bildschirm angezeigt. Wenn Sie die Zugangsroutinen weiter absichern wollen, können Sie die Fehlerversuche mitzählen und nach 3 Fehlerversuchen den Zugang für X-Minuten sperren.

Schritt 6: Kontrolle der Session-Daten

Es wird nachgesehen, ob die entsprechenden Session-Daten vorhanden sind. Wenn diese korrekt vorhanden sind, läuft das Programm im Punkt 7 weiter. Dort kommt nun der eigentliche geschützte Programmteil, den nur berechtigte Benutzer sehen.

Schritt 7: geschützte Programmteil

Hier ist der geschützte Programmteil - dies kann jetzt beliebig komplex sein. Zum Testen können Sie einfach die aktuelle Uhrzeit ausgeben lassen.

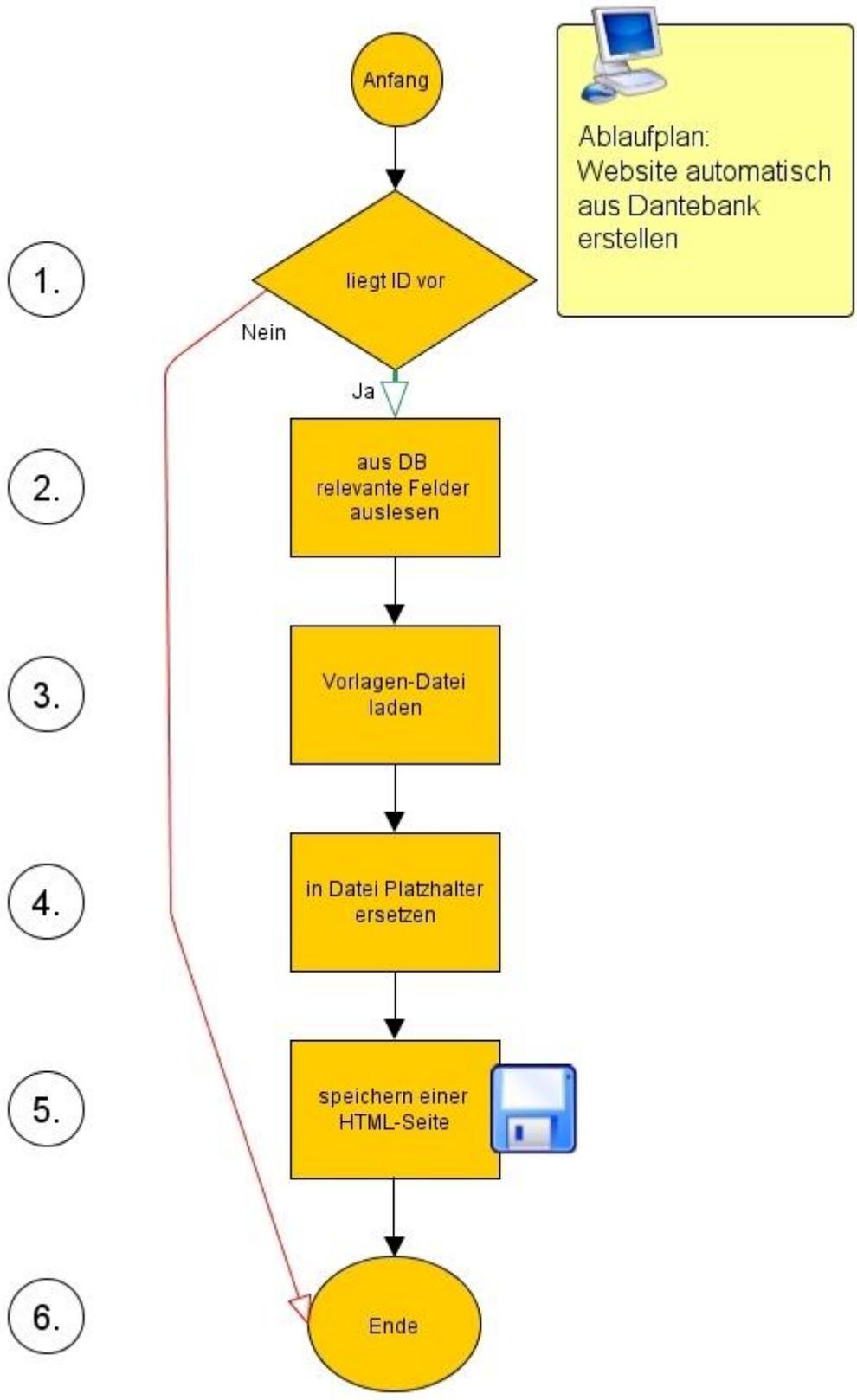
Schritt 8: Formularausgabe zum einloggen

Nach dem fehlgeschlagenen Test der Session-Einträge ist klar, dass der Nutzer nicht eingeloggt ist. Also wird Ihm das Formular zum einloggen angezeigt. Nach dem Anzeigen des Formulars wird das Programm beendet!

Content Management System (CMS)

Ein wichtiger Part eines Content Management Systems (CMS) ist das generieren der HTML-Seite. Dies kann in verschiedenen Weisen angegangen werden. Oft sieht man dynamische Systeme in der Art **domainnamen-xy.de/index.php?id=1** Dies ist unter dem Gesichtspunkt der Suchmaschinenoptimierung ungünstig (SEO = search engine optimization).

In der Regel sind die Inhalte der Seiten statischer Natur. Daher lassen wir unser Programm aus einer Vorlagedatei und den Inhalten einer Datenbank statische Seiten erstellen. Für das Beispiel nehmen wir unsere bereits erstellte Adressbuch-Anwendung (siehe relationen-tabellen-verbinden.htm) und lassen auf Wunsch aus einer Adresse eine HTML-Seite erstellen. Man könnte dies z. B. als Mitarbeiterverzeichnis ansehen.



Schritt 1: ID testen

Die Anwendung erwartet eine ID, damit klar ist, aus welchem Datensatz die CMS-Anwendung die Daten nehmen soll. Ist keine ID vorhanden, dann gibt es auch keine HTML-Seite und das Programm wird beendet.

Schritt 2: Datenbank auslesen

Aus der Datenbank werden die benötigten Felder ausgelesen.

Schritt 3: Vorlagendatei laden

Die Vorlagendatei "vorlage.htm" wird in eine Variable geladen (siehe dateien-auslesen.htm). Davor muss man natürlich erst mal diese Vorlagendatei erstellt haben. Erstellen Sie also eine komplette HTML-Seite mit Design und Co und Platzhalter an den Stellen, an dem die Daten erscheinen sollen. Als Platzhalter sollten wir eine eindeutige Kennzeichnung nutzen, z. B. vor den Vornamen dann "**^vorname^**". Dieses Dächchen wird sonst i.d.R. nie genutzt. Findet sich auf der Tastatur links neben der 1 und bekommt man erst nach zusätzlichem drücken der Leertaste.

Schritt 4: Ersetzen der Platzhalter

Ersetzen der Platzhalter. Alle Platzhalter müssen durch die Datenbankinhalt ersetzt werden. Die einfachste Methode ist:

```
$vorlage = str_replace("^vorname^", $datenbank[vorname], $vorlage);
```

Pfiffiger geht es mit Arrays. Für das Beispiel hier ist es so ausreichend.

Schritt 5: HTML-Seite speichern

Speicher in einer Datei. Dabei sollte der Dateiname eindeutig gewählt werden. Dazu kann z.B. eine Kombination aus „id“ und einem oder mehreren Datenbankfeldern gewählt werden.

```
// und nun die Daten in eine Datei schreiben
// Datei wird zum Schreiben geöffnet
$handle = fopen ( $dateiname, "w" );

// schreiben des Inhaltes der Variable
fwrite ( $handle, $vorlage );

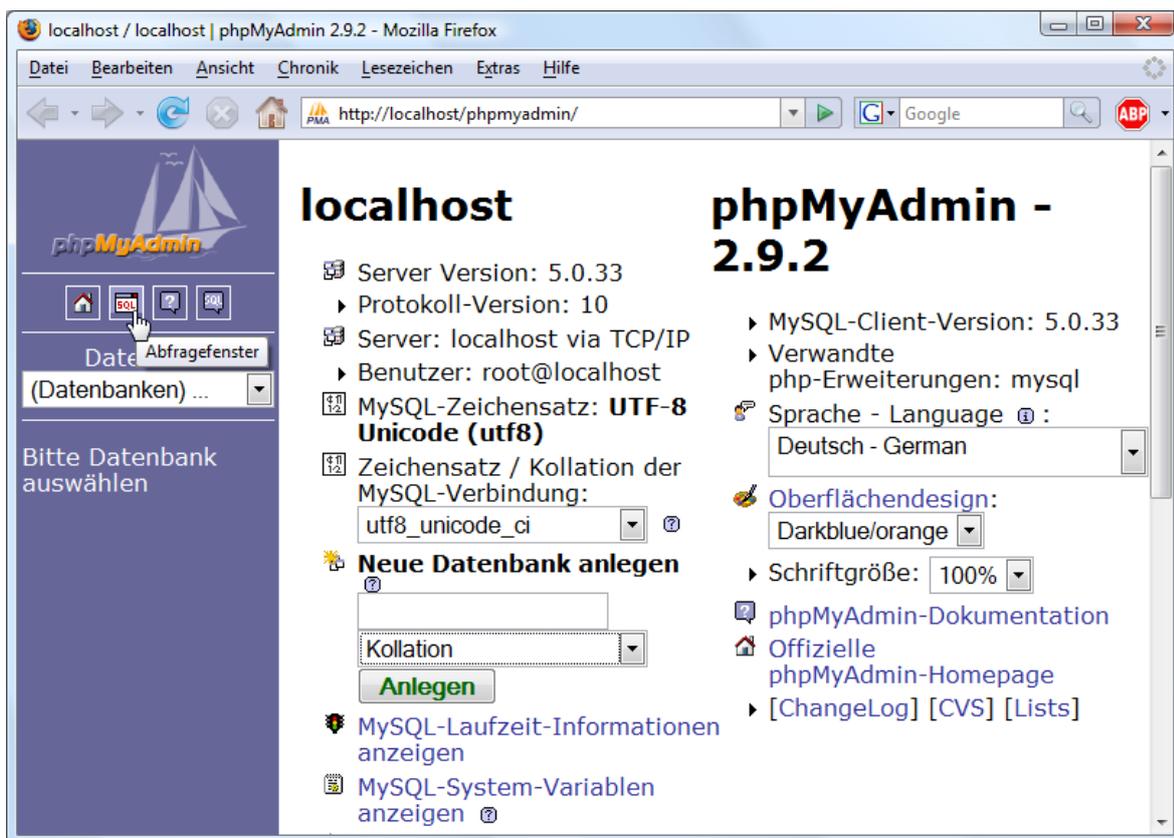
// Datei schließen
fclose ( $handle );
```

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Gästebuch programmieren mit MySQL als Datenbank

Im Folgenden werden wir nun in PHP ein Gästebuch programmieren, das die Einträge in einer Datenbank ablegt.

Wenn wir noch keine Datenbank haben, in der wir die Tabellen speichern können, erstellen wir im PHP eine Datenbank. Im folgenden Beispiel nenne ich diese "homepageanwendungen".





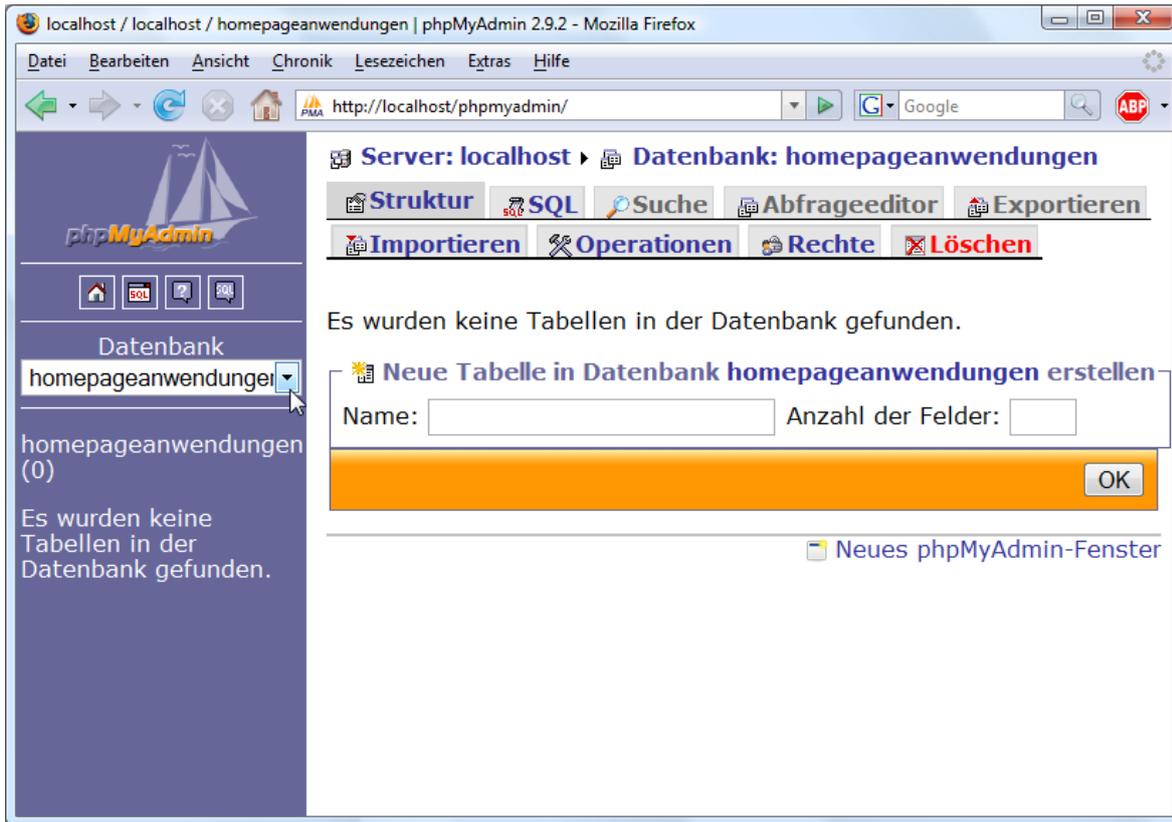
```
CREATE DATABASE `homepageanwendungen` ;
```

Als erstes benötigen wir eine Tabelle in unserer Datenbank. Diese nennen wir "gaestebuch". Diese soll 6 Felder bekommen. Wir wollen erfassen: Name des Schreibers, E-Mail-Adresse des Schreibers, WWW-Seite des Schreibers, Datum und Uhrzeit des Eintrags und natürlich den Eintrag selber.

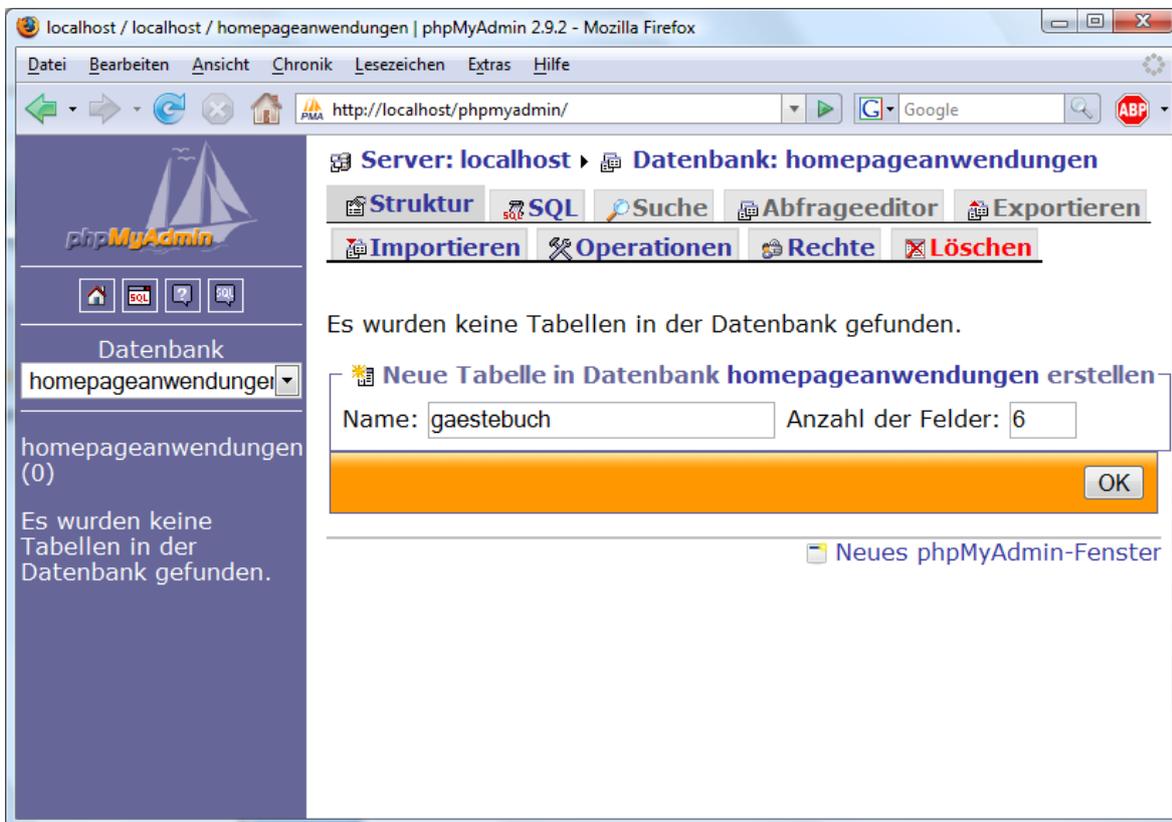
Aufgabe: bestimmen Sie (bevor Sie weiterlesen) sinnvolle Namen und überlegen Sie sich die Feldtypen und die benötigten Feldlängen.

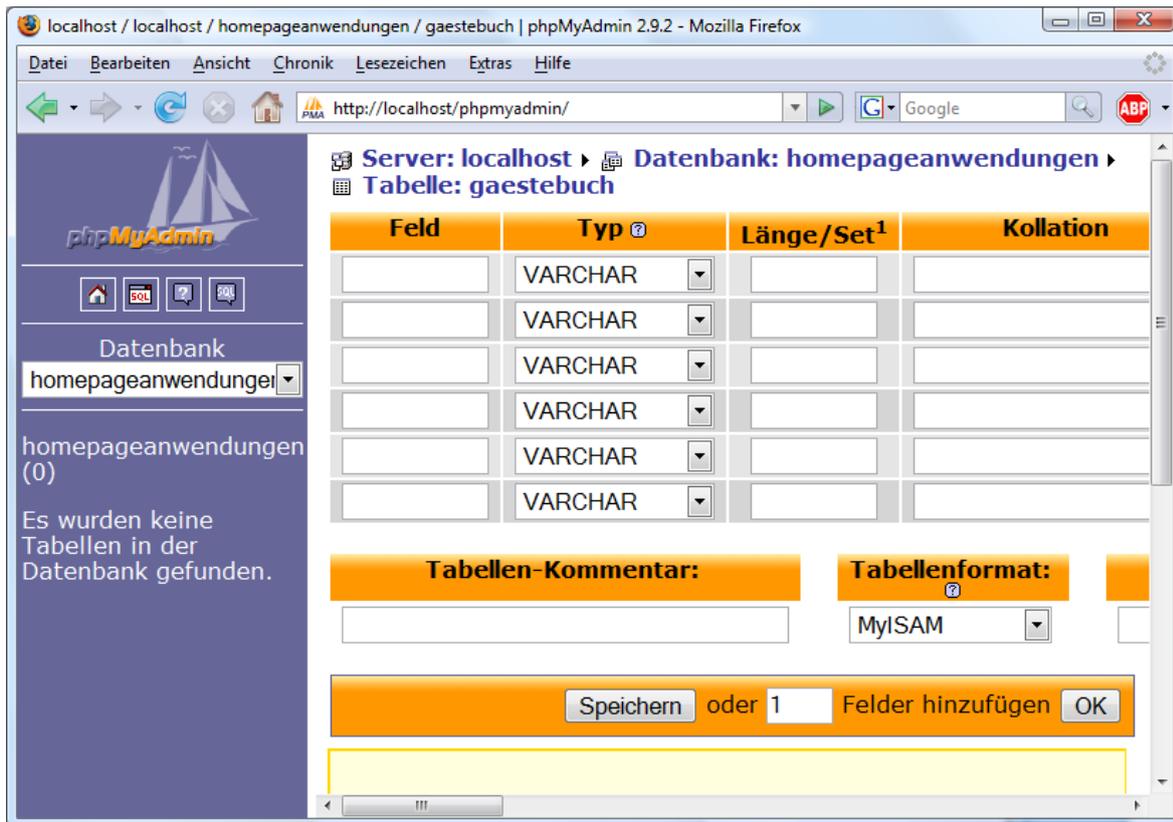
In phpMyAdmin Wahl der Datenbank auf der rechten Seite.

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

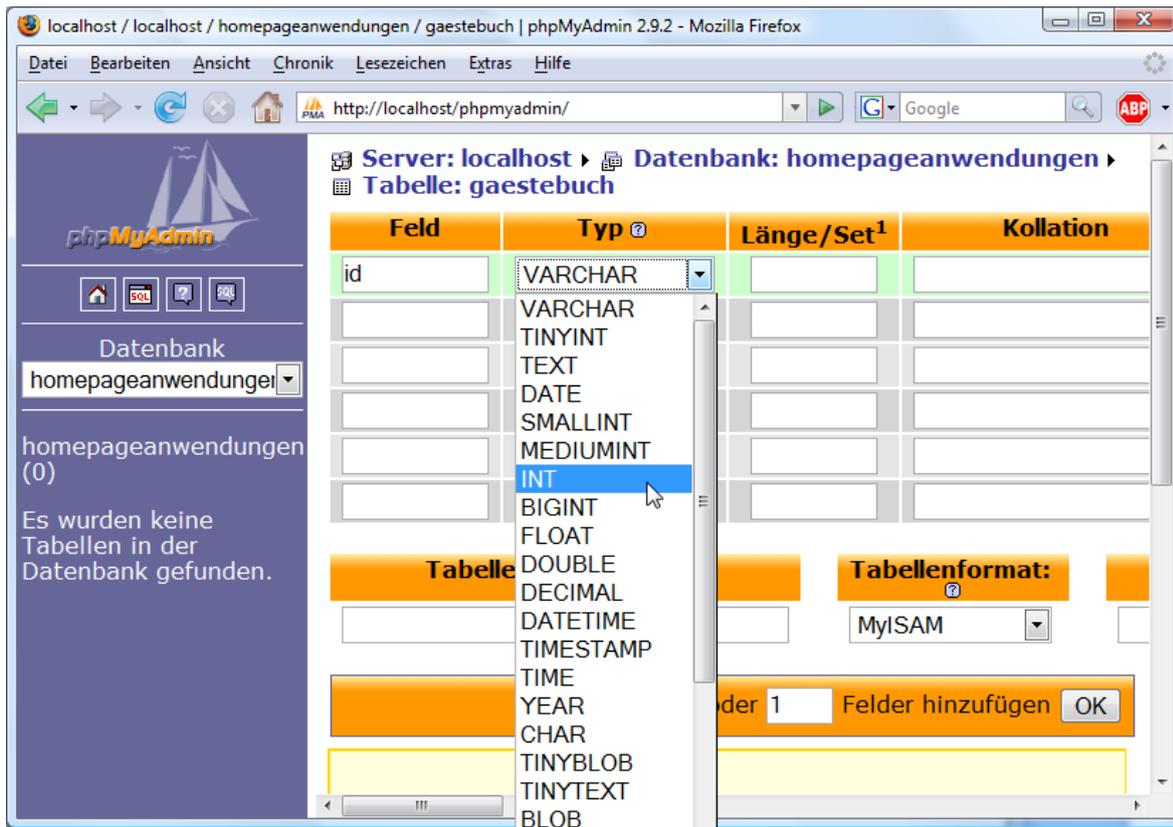


Neue Tabelle mit dem Namen "gaestebuch" erstellen in phpMyAdmin.

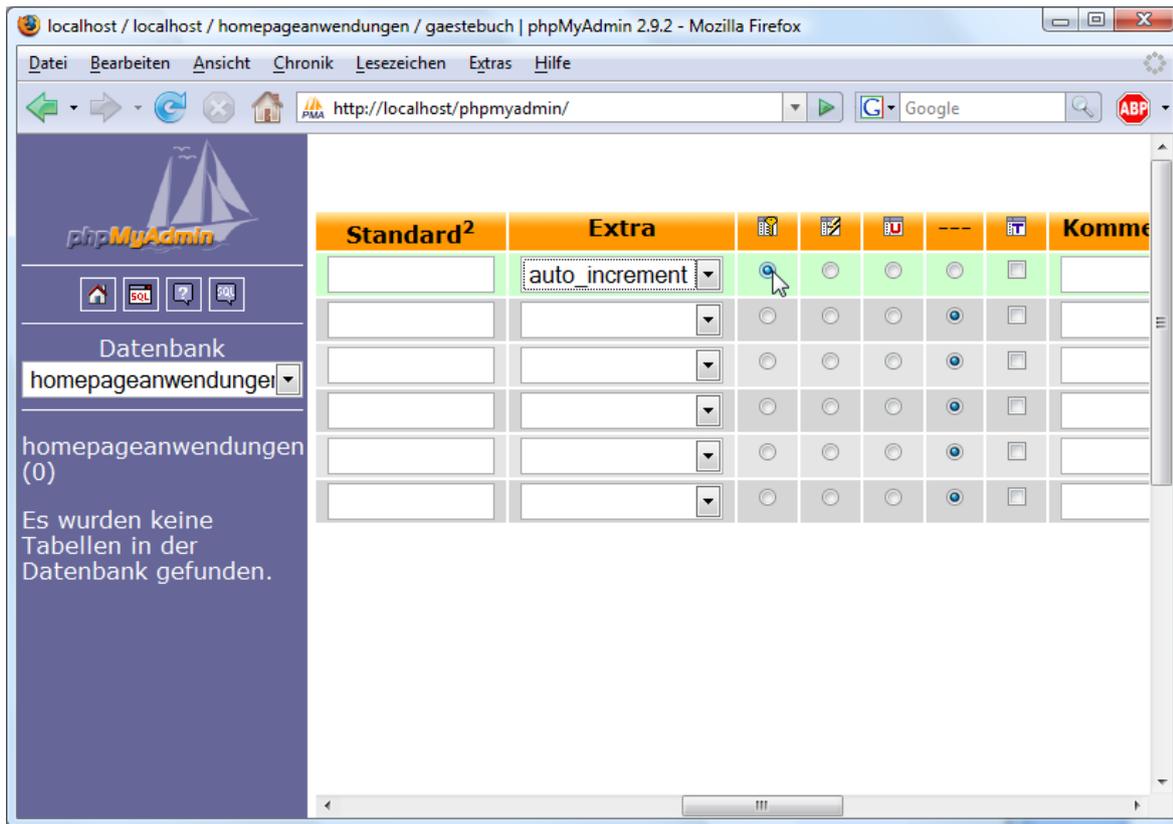




Unser erstes Feld ist das ID-Feld, über das wir jeden Eintrag eindeutig ansprechen können. Diese ist eine Ganzzahl (INT) und kann bis zu 20 Stellen bekommen (wir wollen viele, viele Einträge).

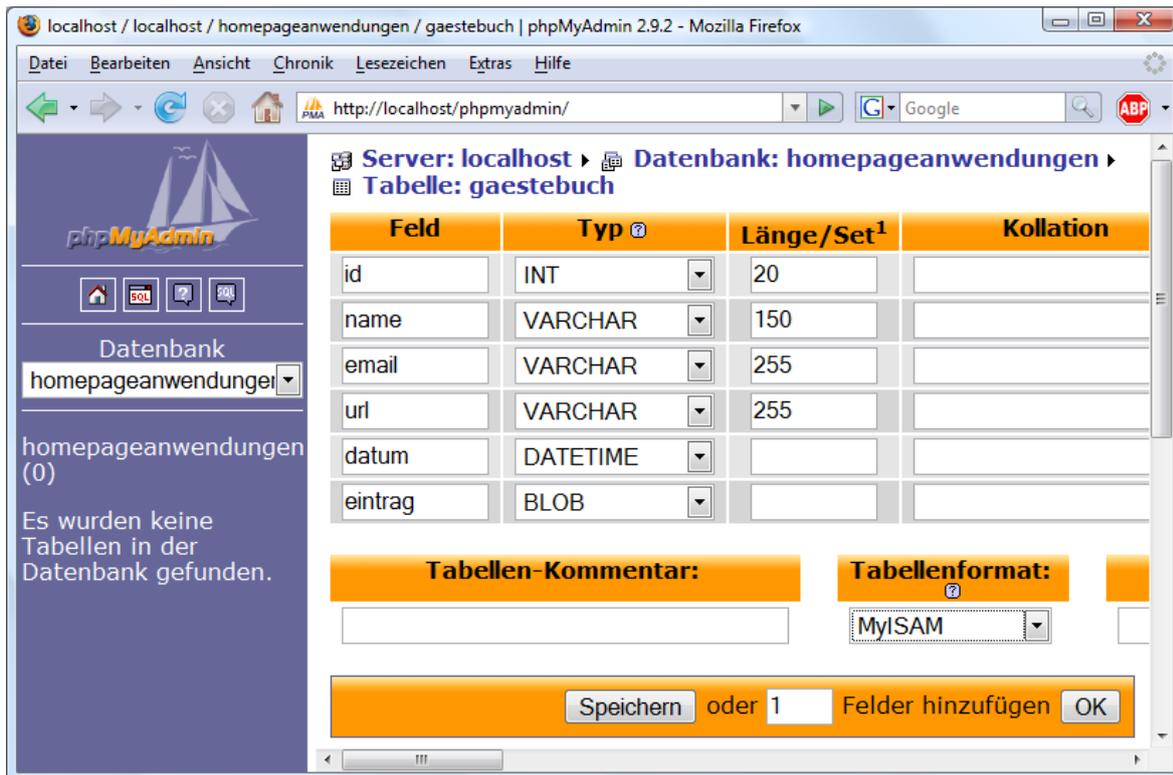


Zusätzlich soll bei diesem Feld automatisch eine fortlaufende Nummer zugeteilt werden, ohne dass wir uns darum kümmern müssen. Daher vergeben wir für das Feld unter Extra "auto_increment" und machen es zum Primärschlüssel.

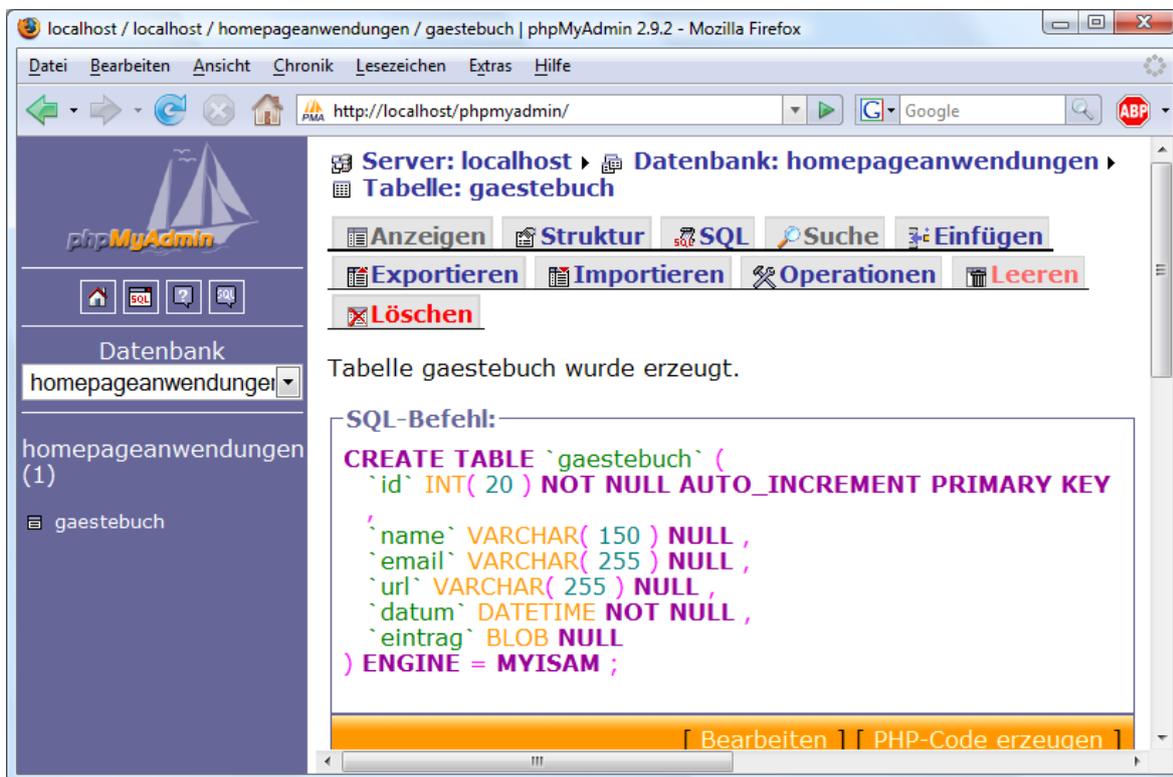


Aufgabe mit der Benennung der Felder von oben erledigt? Mein Vorschlag wäre:

name VARCHAR 150
 email VARCHAR 255
 url VARCHAR 255
 datum datetime
 eintrag blob



Nach dem Klicken auf OK wird die Tabelle erstellt und taucht rechts auf.



Wir bekommen auch zu Gesicht, wie der direkte MySQL-Befehl aussieht.

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

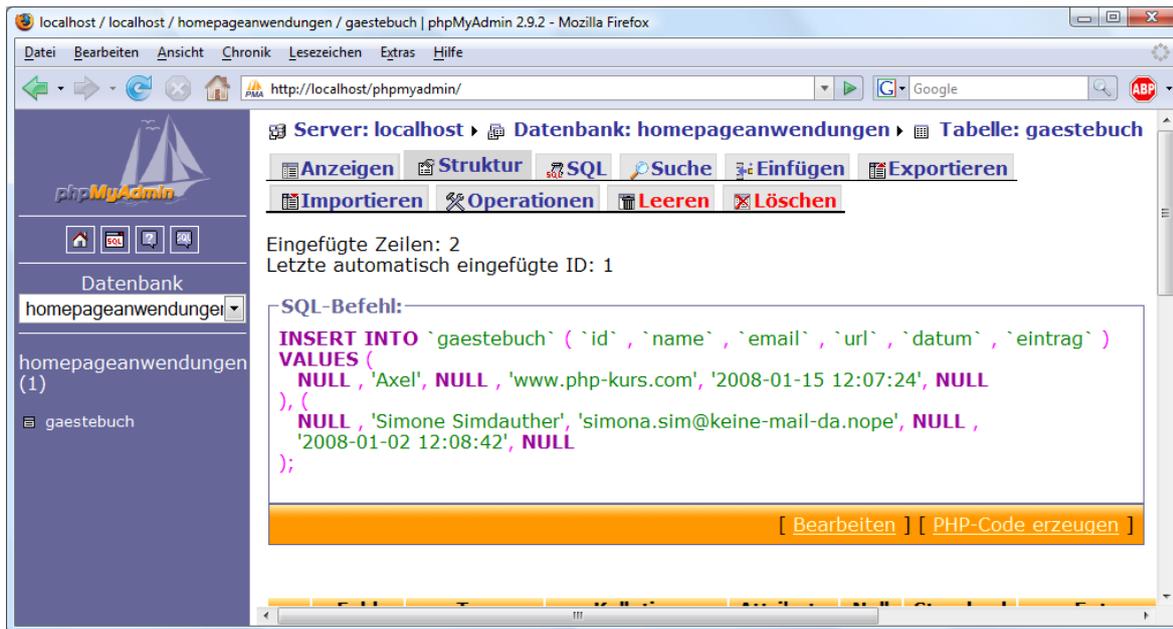
```
CREATE TABLE `gaestebuch` (
  `id` INT( 20 ) NOT NULL AUTO_INCREMENT PRIMARY KEY ,
  `name` VARCHAR( 150 ) NULL ,
  `email` VARCHAR( 255 ) NULL ,
  `url` VARCHAR( 255 ) NULL ,
  `datum` DATETIME NOT NULL ,
  `eintrag` BLOB NULL
) ENGINE = MYISAM ;
```

Und jetzt können wir zum Testen 1-2 Einträge von Hand hinzufügen. Dies geschieht über Einfügen. Es werden nun Daten eingegeben – das Feld ID und der Eintrag selber bleiben leer. ID wird automatisch vergeben und in das BLOB-Feld können wir mit phpMyAdmin nicht schreiben.



Nach Klicken von OK bekommen wir wieder zur Bestätigung den entsprechenden MySQL-Befehl:

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

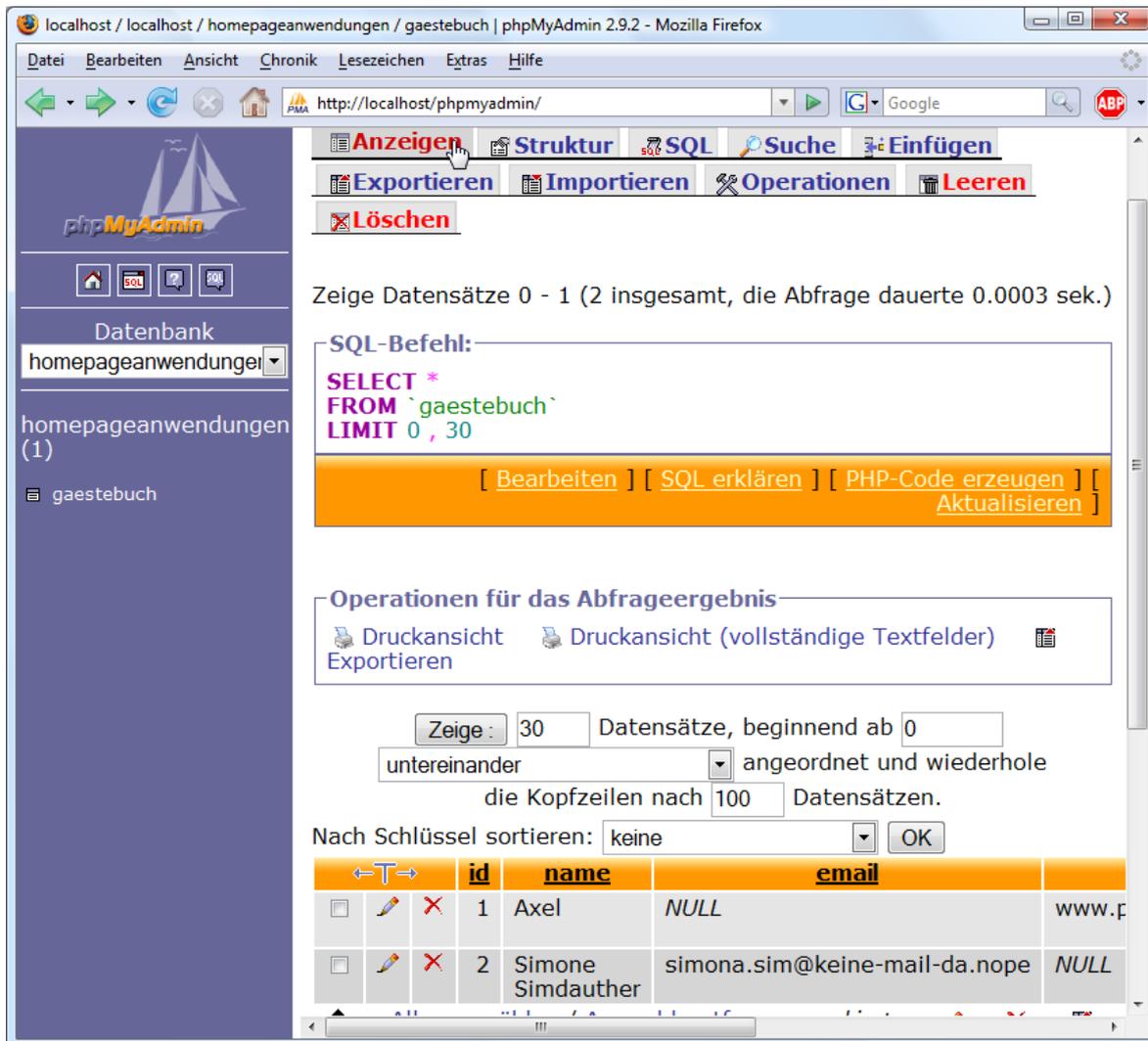


```

INSERT INTO `gaestebuch` ( `id` , `name` , `email` , `url` , `datum` ,
`eintrag` )
VALUES (
NULL , `Axel`, NULL , `www.php-kurs.com`, `2008-01-15 12:07:24`, NULL
), (
NULL , `Simone Simdauther`, `simona.sim@keine-mail-da.nope`, NULL , `2008-01-02
12:08:42`, NULL
);

```

Zum Überprüfen der Inhalte im Kopfbereich auf „Anzeigen“ klicken und nach unten scrollen – dort sollten nun unsere Testeinträge auftauchen.



Anwendung Gästebuch erstellen

Im ersten Schritt möchten wir die Einträge in unserem Gästebuch anzeigen. Dazu müssen wir eine Verbindung zu unserer Datenbank aufbauen. Wir hinterlegen als erstes die Zugangsdaten in Konstanten in der Datei "gaestebuch.php".

```
define ( 'MYSQL_HOST', 'localhost' );
define ( 'MYSQL_BENUTZER', 'root' );
define ( 'MYSQL_KENNWORT', '' );
define ( 'MYSQL_DATENBANK', 'homepageanwendungen' );
```

Nun wird die eigentliche Verbindung zum MySQL-Server aufgebaut.

```
$db_link = @mysql_connect (MYSQL_HOST, MYSQL_BENUTZER, MYSQL_KENNWORT);
if ( ! $db_link )
{
    // hier sollte dann später dem Programmierer eine
    // E-Mail mit dem Problem zukommen gelassen werden
    // die Fehlermeldung für den Programmierer sollte
```

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

```
// das Problem ausgeben mit: mysql_error()
die('keine Verbindung zur Zeit möglich - später probieren ');
}
```

Bisher haben wir dem "Computer" gesagt, dass er eine Verbindung aufbauen soll mit dem MYSQL_HOST und wir berechtigt sind (MYSQL_BENUTZER und MYSQL_KENNWORT). Im nächsten Schritt sagen wir nun noch, welche Datenbank eigentlich genutzt werden soll:

```
$db_sel = mysql_select_db( MYSQL_DATENBANK )
or die("Auswahl der Datenbank fehlgeschlagen");
```

Nun werden über eine MySQL-Abfrage die Daten der Datenbank angefordert.

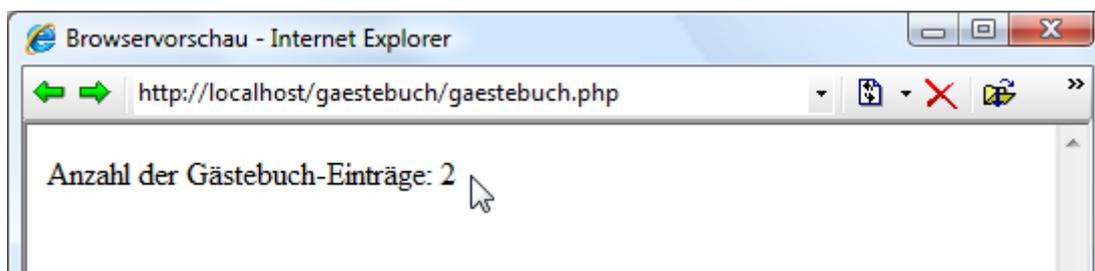
```
// SQL-Befehl für den Zugriff
$sql = "
    SELECT
        *
    FROM gaestebuch
    ORDER BY datum
";

// ausführen des mysql-Befehls
$db_erg = mysql_query( $sql );
if ( ! $db_erg )
{
    die('Ungültige Abfrage: ' . mysql_error());
}
```

Bisher gibt es noch keine Ausgaben auf dem Bildschirm. Wir möchten wissen, wieviele Einträge vorhanden sind, und diese Anzahl ausgeben.

```
// Anzeige der Anzahl der Einträge
$anzahl_eintraege = mysql_num_rows($db_erg);
echo "<p>Anzahl der Gästebuch-Einträge: $anzahl_eintraege </p>";
```

Wenn wir uns nirgend vertippt haben und der Aufbau zu Datenbank klappt, erhalten wir nun die Anzahl der Einträge. Herzlichen Glückwunsch – der erste erfolgreiche Zugriff auf die Datenbank über PHP und Nutzung der Daten.



Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Anzeige der Daten aus der MySQL-Datenbank mit PHP

Jetzt lassen wir uns alle Daten der Reihe nach anzeigen. Dazu durchlaufen wir eine Schleife.

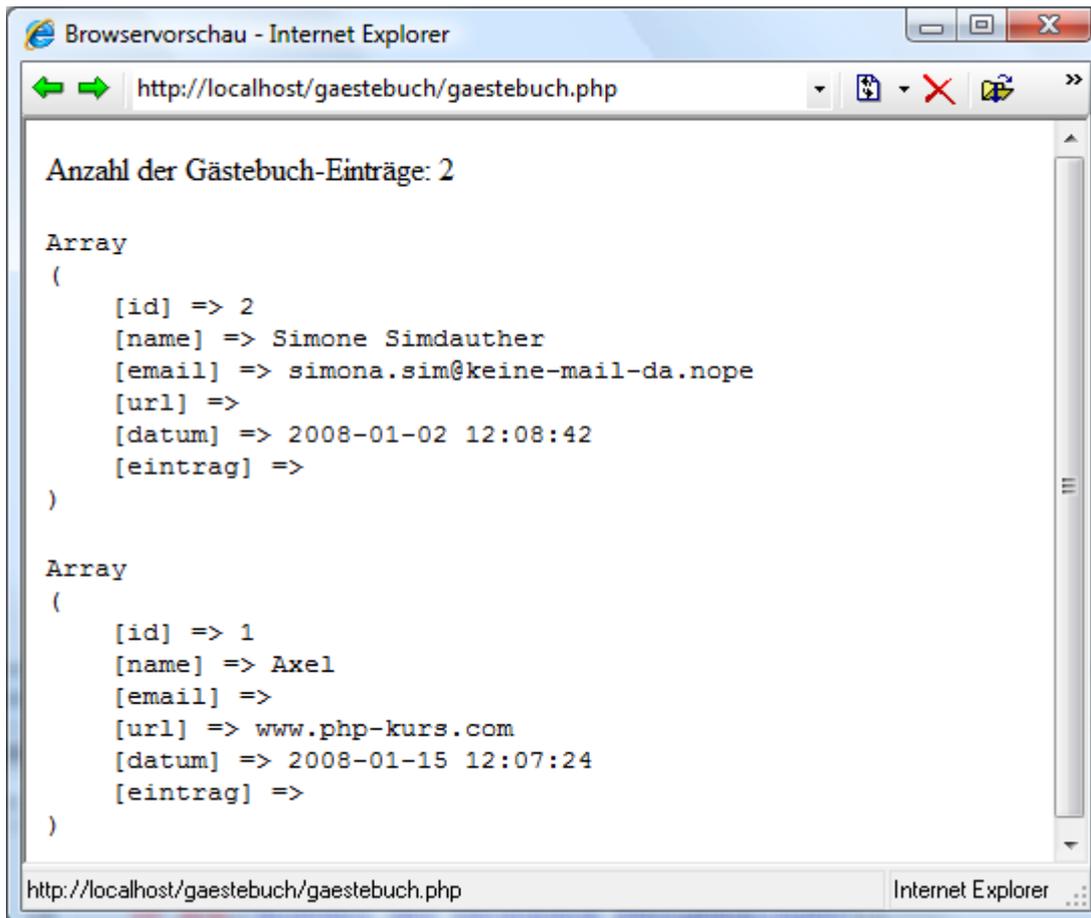
```
while ($daten = mysql_fetch_array( $db_erg, MYSQL_ASSOC))
{
    // Aushabe der Daten
}
```

Der Befehl `mysql_fetch_array ($db_erg, MYSQL_ASSOC)` bewirkt, dass das Ergebnis unserer Datenbankabfrage (Verweis ist in `$db_erg` gespeichert) als assoziiertes Array übergeben wird. Assoziiert bedeutet, dass auf unser Array direkt über die Namen der Tabellenfelder zugegriffen werden kann.

Innerhalb der Schleife können wir nun auf das Array zugreifen. Im ersten Test lassen wir uns (aus didaktischen Gründen) einfach den Inhalt des Array (`$daten`) komplett ausgeben mit `print_r ($daten)`; - damit es sauber zu lesen ist, lassen wir noch `<pre>` ausgeben (alle Umbrüche werden angezeigt).

```
while ($daten = mysql_fetch_array( $db_erg, MYSQL_ASSOC))
{
    // Aushabe der Daten
    echo "<pre>";
    print_r ($daten);
    echo "</pre>";
}
```

Wir sollten nun etwas in der Richtung erhalten (je nachdem, was in der Datenbank eingetragen wurde).



Wir können nun die einzelnen Array-Daten ausgeben lassen mit:

```
echo $daten['name'];
```

also in unserem Gästebuch-Script: (jetzt noch ohne großartiges Design)

```
while ($daten = mysql_fetch_array( $db_erg, MYSQL_ASSOC))
{
    // Aushabe der Daten
    echo "ID: ";
    echo $daten['id'];
    echo "<br />";

    echo "Name: ";
    echo $daten['name'];
    echo "<br />";

    echo "E-Mail: ";
    echo $daten['email'];
    echo "<br />";

    echo "URL: ";
    echo $daten['url'];
    echo "<br />";
}
```

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

```

    echo "Datum: ";
    echo $daten['datum'];
    echo "<br />";

    echo "Gästebucheintrag: ";
    echo $daten['eintrag'];
    echo "<br />";
}

```

Und wenn wir fertig mit dem Zugriff auf die Datenbank sind, sollten wir die Ressourcen wieder freigeben:

```
mysql_free_result( $db_erg );
```

neue Einträge in das Gästebuch (in die MySQLDatenbank)

Für den Eintrag ins Gästebuch erstellen wir eine **neue Datei mit dem Namen "gaestebuch-eintrag.php"**. Wir zeigen ein Formular, in dem der Gästebucheintrag durch den Besucher gemacht werden kann. Dazu benötigen wir 4 Felder: Name, E-Mail, URL, Eintrag.

```

<form name="" action="<?php echo $_SERVER['PHP_SELF']; ?>"
  method="POST" enctype="text/html">
<p>Ihr Name:<br />
<input type="text" name="name" value="" size="50" maxlength="150" />
</p>

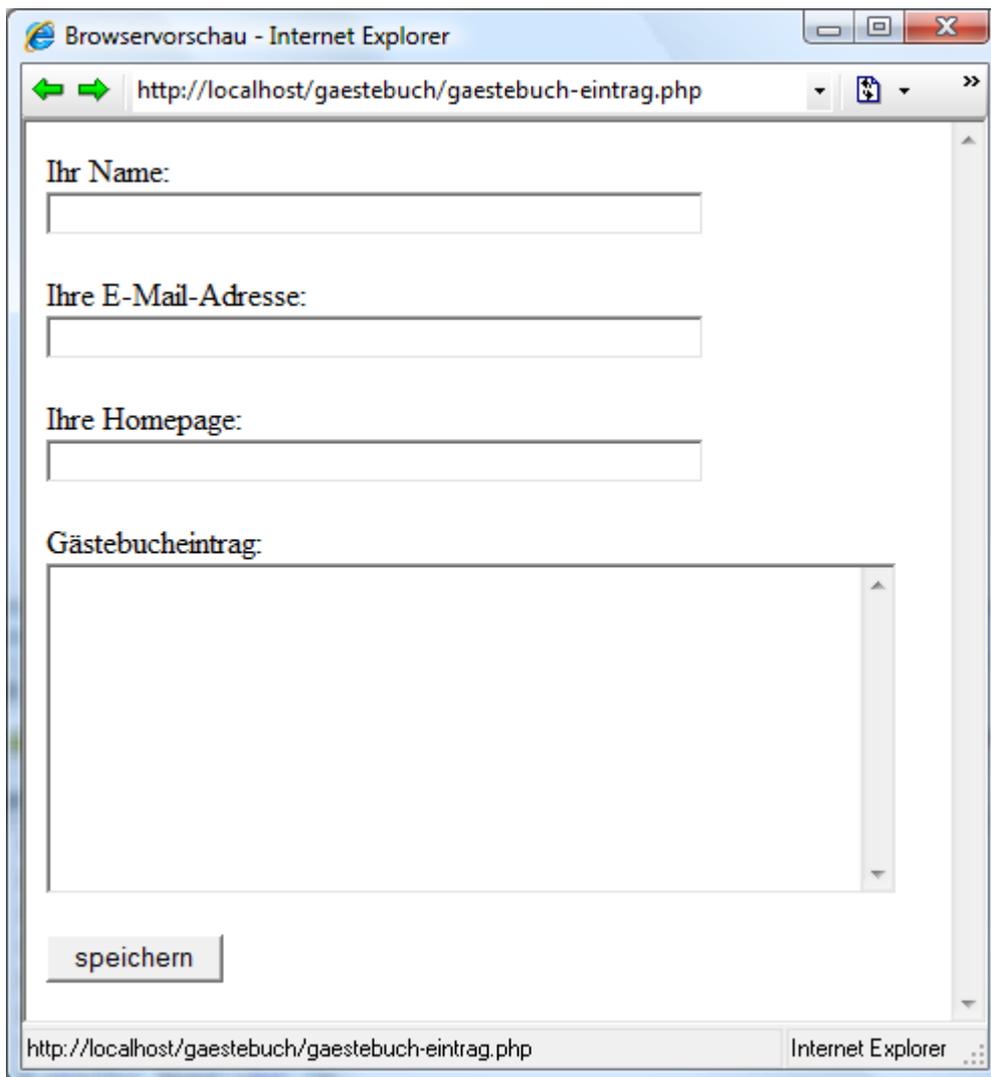
<p>Ihre E-Mail-Adresse:<br />
<input type="text" name="email" value="" size="50" maxlength="150" />
</p>

<p>Ihre Homepage:<br />
<input type="text" name="url" value="" size="50" maxlength="150" />
</p>

<p>Gästebucheintrag:<br />
<textarea name="eintrag" rows="10" cols="50"></textarea>
</p>

<input type="Submit" name="" value="speichern" />
</form>

```



Der Inhalt des Formulars wird an sich selber gesendet. Jetzt benötigen wir am Anfang vor der Anzeige des Formulars noch eine Auswertung, ob das Formular ausgefüllt wurde, und können dann die Daten abspeichern.

Wir testen, ob ein Eintrag im Feld "Gästebucheintrag" gemacht wurde.

```
if ( $_POST['eintrag'] != "" )
{
    echo "<h2>Eintrag speichern</h2>";
    // hier kommt nun der entsprechende PHP-Code
    exit;
}
```

Im nächsten Schritt bauen wir nun den entsprechenden MySQL-Befehl zusammen, so dass diese Daten gespeichert werden. Wir bereiten den MySQL-Befehl in einer Variablen vor.

```
$sql = " INSERT INTO gaestebuch ";
$sql .= " SET ";
```

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

```

$sql .= " name     ='". $_POST['name'] ."', ";
$sql .= " email   ='". $_POST['email'] ."', ";
$sql .= " url     ='". $_POST['url'] ."', ";
$sql .= " datum  ='". date("Y-m-d H:i:s") ."', ";
$sql .= " eintrag='". $_POST['eintrag'] ."' ";

```

Zur Erinnerung: Die Konstruktion " .= " bewirkt, dass der folgende Ausdruck an bestehende Inhalte der Variablen angehängt wird.

Zur Kontrolle lassen wir uns die fertige Konstruktion ausgeben:

```

echo "<hr />SQL: $sql<hr />";

```

Nun müssen wir wieder (da ja neue Datei) die Verbindung zur Datenbank aufbauen, unsere Berechtigung (Benutzername und Kennwort) checken und die entsprechende Datenbank auswählen.

Diese Konstruktion kennen Sie aus der Datei "gaestebuch.php":

```

define ( 'MYSQL_HOST', 'localhost' );
define ( 'MYSQL_BENUTZER', 'root' );
define ( 'MYSQL_KENNWORT', '' );
define ( 'MYSQL_DATENBANK', 'homepageanwendungen' );

$db_link = @mysql_connect (MYSQL_HOST, MYSQL_BENUTZER, MYSQL_KENNWORT);
if ( ! $db_link )
{
    // hier sollte dann später dem Programmierer eine
    // E-Mail mit dem Problem zukommen gelassen werden
    // die Fehlermeldung für den Programmierer sollte
    // das Problem ausgeben mit: mysql_error()
    die('keine Verbindung zur Zeit möglich - später probieren ');
}

$db_sel = mysql_select_db( MYSQL_DATENBANK )
    or die("Auswahl der Datenbank fehlgeschlagen");

```

Jetzt kommt der eigentliche Part: Wir übergeben unseren zusammgebauten SQL-Befehl und lassen diesen ausführen mit `mysql_query()` - Das war es schon – der Eintrag ist erfolgt.

```

// ausführen des mysql-Befehls
$db_erg = mysql_query( $sql );
if ( ! $db_erg )
{
    die('Ungültige Abfrage: ' . mysql_error());
}

```

Zum Schluss bieten wir noch einen Link, um das Gästebuch aufzurufen, und brechen dann das Script mit `exit` ab:

```

echo '<p><a href="gaestebuch.php">Gästebuch anzeigen</a></p>';
exit;

```

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Es funktioniert, ist aber gefährlich! - im nächsten Kapitel unbedingt weiterlesen!

Gästebuch absichern

Gefahr erkannt, Gefahr gebannt

Warum ist ein so einfaches Script gefährlich? Dafür gibt es zahlreiche Gründe. Fangen wir damit an, dass erfahrene Nutzer uns HTML (und schlimmeres) unterjubeln können. Die harmlosen Dinge sind, dass dann Texte fett ("****") und/oder farbig dargestellt werden. Auf die weniger harmlosen gehe ich weiter unten ein.

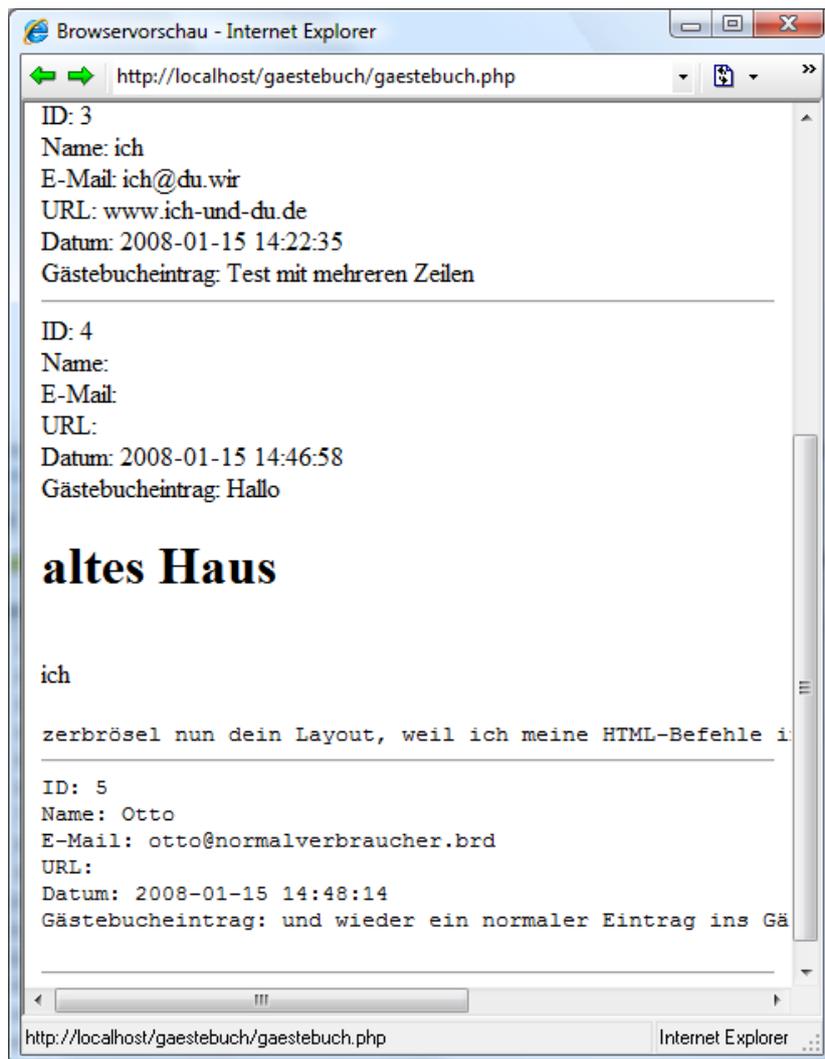
Wir müssen sicherstellen, dass das, was der Benutzer eingibt, auch wirklich nur Text ist. Wir lassen also alle Variablen durch eine Waschanlage, die unerwünschte Dinge herausfiltert. Dies ist relativ einfach. Nähere Informationen finden Sie in diesem Kurs unter: <http://www.php-kurs.com/uebergebene-variablen-bereinigen.htm>

Für das Reinemachen rufen wir am Anfang des "gaestebuch-eintrag.php" folgende 3 Zeilen auf. Darin werden dann übergebene Variablen durch eine Funktion geschleust und kommen mit 60 Grad gewaschen, gebügelt und von Ungeziefer befreit wieder heraus.

Bevor Sie aber diese Zeilen an den Anfang packen, geben Sie im Gästebucheintrag folgenden Text ein:

```
Hallo <h1>altes Haus</h1><br />
ich <pre>zerbrösel nun dein Layout,
weil ich meine HTML-Befehle in deine
Seite packte - ätsch =;)
```

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen



Und nun aber das Vollwaschmittel an den Anfang packen:

```
array_walk ( $_POST,      'eingabebereinigen' );
array_walk ( $_GET,      'eingabebereinigen' );
array_walk ( $_REQUEST,  'eingabebereinigen' );

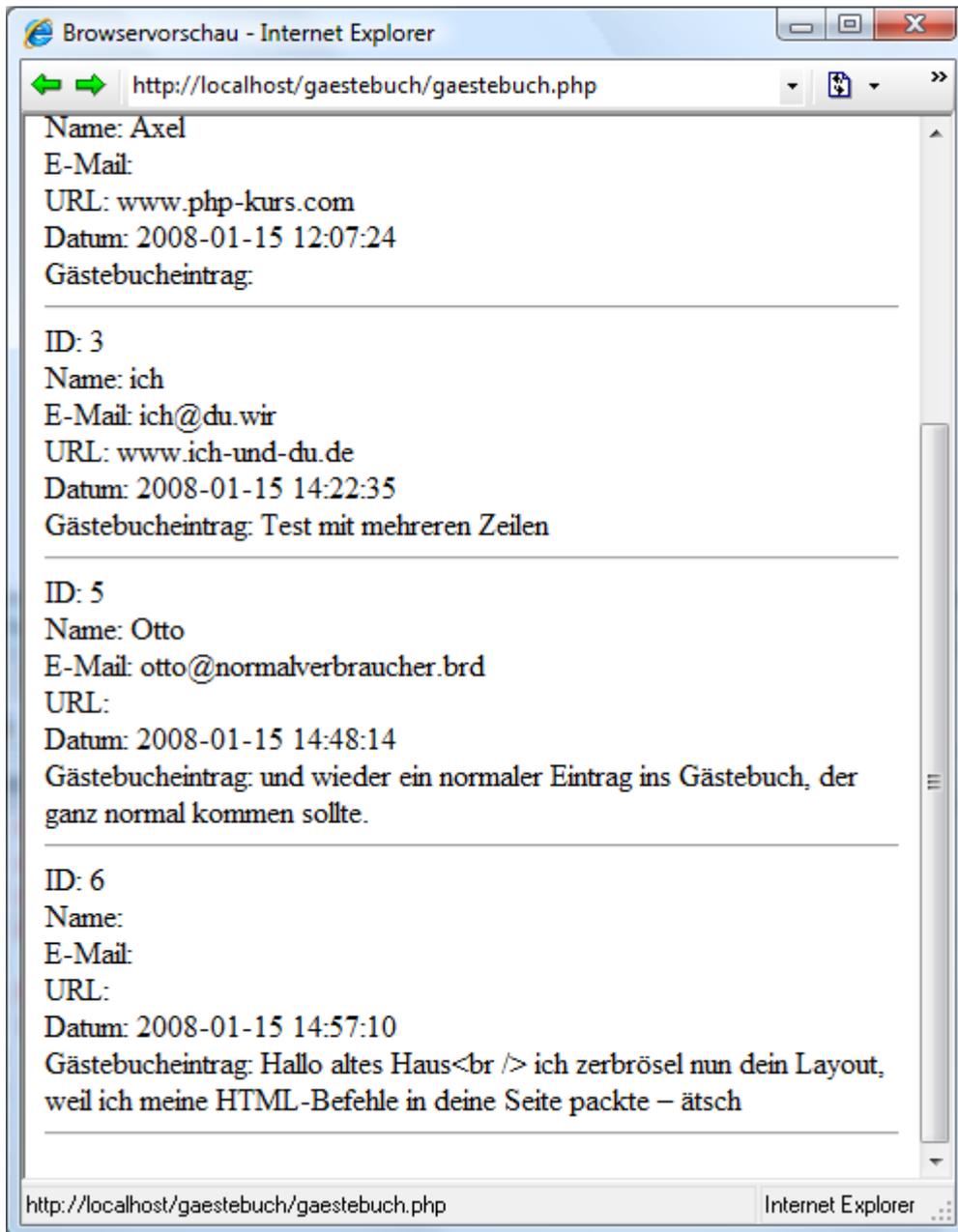
function eingabebereinigen(&$value, $key)
{
    // keine HTML-Tags erlaubt, außer p und br
    $value = strip_tags($value, '<p><br /><b><strong>');

    // HTML-Tags maskieren
    $value = htmlspecialchars($value, ENT_QUOTES);

    // Leerzeichen am Anfang und Ende beseitigen
    $value = trim($value);
}
```

Löschen Sie diesen Datensatz aus der Datenbank, um wieder eine normale Anzeige zu bekommen. Dann probieren Sie es nochmals mit dem HTML-Eintrag. Nun aber sollte es wie folgt aussehen:

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen



Mehr zu diesen sogenannten HTML Injection Angriffe gibt es unter: http://www.hdm-stuttgart.de/~ms096/haking9-inject_de.pdf

Ohne Schutz kann auch JavaScript eingeschleust werden (z.B. `<script>alert('gehackt :')</script>`) oder sonstiges.

2. Problemzone: Anführungsstriche

Dieses Bereinigen schaut auch nach eingegebenen einfachen und doppelten Anführungsstrichen. Werden diese ungefiltert übernommen, ist genau an diesem Zeichen dann Ihr SQL-Befehl beendet und Sie bekommen eine Fehlermeldung um die Ohren geklatsch (bzw. der Gästebucheintrager). Das wäre dann noch der bessere Fall. Im Extremfall

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

kann so jemand Ihre Datenbank „hacken“ und relativ flott viele unangenehme Dinge durchziehen (z.B. löschen).

Daher: niemals den Eingaben von Benutzern trauen, da diese im Internet nicht bekannt sind.

3. Problemzone: E-Mail-Adresse

Sie können zwar die E-Mail-Adresse erfassen, sollte diese aber tunlichst nicht als Text ausgehen. Ansonsten haben Sie Ihrem Gästebucheinträger einen Bärendienst erwiesen. Geben Sie die Adresse im Klartext aus, freuen sich alle Spam-Sammler (die automatisiert sind) und der Gästebuchnutzer bekommt auf einmal mehr Spam als früher (und weiß in der Regel gar nicht warum). Wenn unbedingt die E-Mail-Adresse angezeigt werden soll, dann z.B. als Grafik <http://www.php-kurs.com/grafiken-erstellen-mit-php.htm>

4. Problemzone: URL

Klar macht es Spaß, dass andere ihre URL der eigenen Website angeben können. Allerdings ist auch das ein Grund, warum viele ungebetene Gäste im Gästebuch einen Eintrag hinterlassen, um dort dann für Ihre Pxxxno-Seite zu werben (was dann im Endeffekt Ihre Website macht).

Es kann auch der Fall sein, dass Leute nur zum Sammeln von Backlinks einen Eintrag schreiben. Denn je mehr Backlinks, desto besser steht man i.d.R. bei Suchmaschinen da.

5. Problemzone: Inhalt

So – auf die Gefahr hin, einem das Gästebuch vollends zu vermiesen. Sie sind nach deutschen Recht verantwortlich was auf Ihrer Seite steht (auch für das, was andere schreiben). Soll heißen, wenn jemand mit wilden Beleidigungen und Diffamierungen um sich wirft und man sich auf „Meinungsfreiheit“ beruft, hat man wenig Spaß vor Gericht. Daher sollten Sie die Gästebucheinträge prüfen und gegebenenfalls löschen. Ich habe schon Gästebucheinträge gesehen, die aus 1000 Links nach außen bestanden, wo man sich nicht sicher war, welcher am unappetitlichsten war. Diese wurden dann teilweise versucht zu verstecken, indem die Schriftgröße extrem verkleinert wurde, damit nur Suchmaschinen diese Links finden.

Nach soviel Problem auch Lösung

Wir wollen die Gästebucheinträge erst nach Sichtkontrolle freischalten (big brother is watching you :)

Wir fügen also ein weiteres Feld in der Datenbank ein, das wir „aktiviert“ nennen. Das geht am schnellsten über phpMyAdmin im Punkt „Stuktur“ unten „Feld hinzufügen“. Als MySQL-Befehl:

```
ALTER TABLE `gaestebuch` ADD `aktiviert` CHAR( 1 ) NOT NULL ;
```

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Das Feld hat die Länge von "1" und ist ein Char. Sobald der Eintrag angezeigt werden darf, wird hier eine 1 gesetzt.

Aufgabe

Um nun das so komfortabel zu machen als möglich, lassen wir uns nach dem Eintrag durch den Nutzer automatisch eine E-Mail senden: siehe: <http://www.php-kurs.com/email-per-php.htm>

In dieser E-Mail gibt es neben dem Text, den der Besucher eingetragen hat, dann noch einen Link für uns zum freischalten: Dort steht dann

```
http://www.php-kurs.com/gaestebuch-freischalten.php?id=$id
```

Dieser Link muss natürlich auf die URL Ihrer Website angepasst werden. Weiterhin sollte auch der Dateinamen variiert werden, sonst kann jeder, den Link kennt, den Eintrag theoretisch freischalten.

In dieser Freischalten-Anwendung steht dann:

```
<?php
$sql = " UPDATE gaestebuch ";
$sql .= " SET aktiviert='1' ";
$sql .= " WHERE id='". $_GET['id'] .' " ";

echo "<hr />SQL: $sql<hr />";

define ( 'MYSQL_HOST', 'localhost' );
define ( 'MYSQL_BENUTZER', 'root' );
define ( 'MYSQL_KENNWORT', '' );
define ( 'MYSQL_DATENBANK', 'homepageanwendungen' );

$db_link = @mysql_connect (MYSQL_HOST, MYSQL_BENUTZER, MYSQL_KENNWORT);
if ( ! $db_link )
{
    // hier sollte dann später dem Programmierer eine
    // E-Mail mit dem Problem zukommen gelassen werden
    // die Fehlermeldung für den Programmierer sollte
    // das Problem ausgeben mit: mysql_error()
    die('keine Verbindung zur Zeit möglich - später probieren ');
}

$db_sel = mysql_select_db( MYSQL_DATENBANK )
    or die("Auswahl der Datenbank fehlgeschlagen");

// ausführen des mysql-Befehls
$db_erg = mysql_query( $sql );
if ( ! $db_erg )
{
    die('Ungültige Abfrage: ' . mysql_error());
}

echo "<h2>Eintrag ". $_GET['id'] ." freigeschaltet</h2>";
echo '<p><a href="gaestebuch.php">Gästebuch anzeigen</a></p>';
```

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

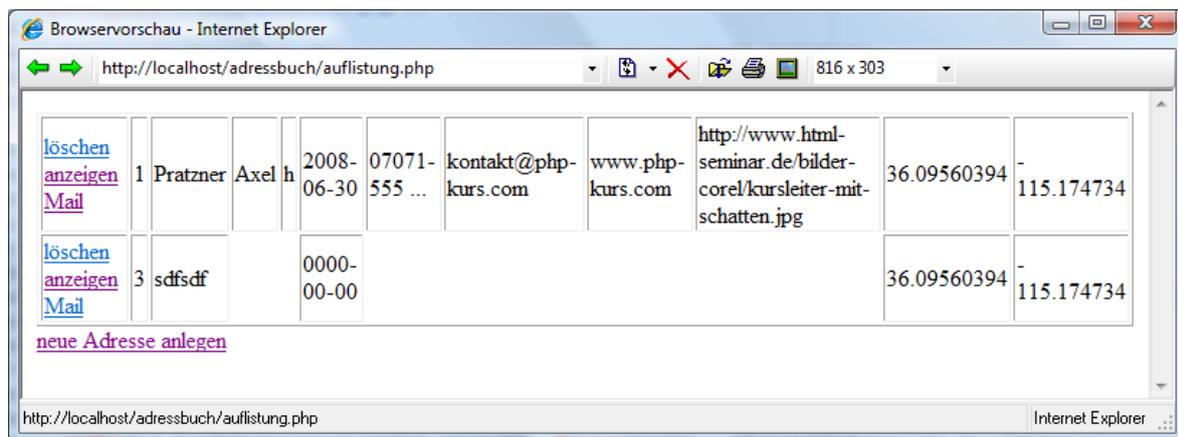
?>

Im Script "gaestebuch.php" wird die SQL-Abfrage erweitert, so dass nur noch freigeschaltete Einträge angezeigt werden:

```
// SQL-Befehl für den Zugriff
$sql = "
    SELECT
        *
    FROM gaestebuch
    WHERE aktiviert='1'
    ORDER BY datum
";
```

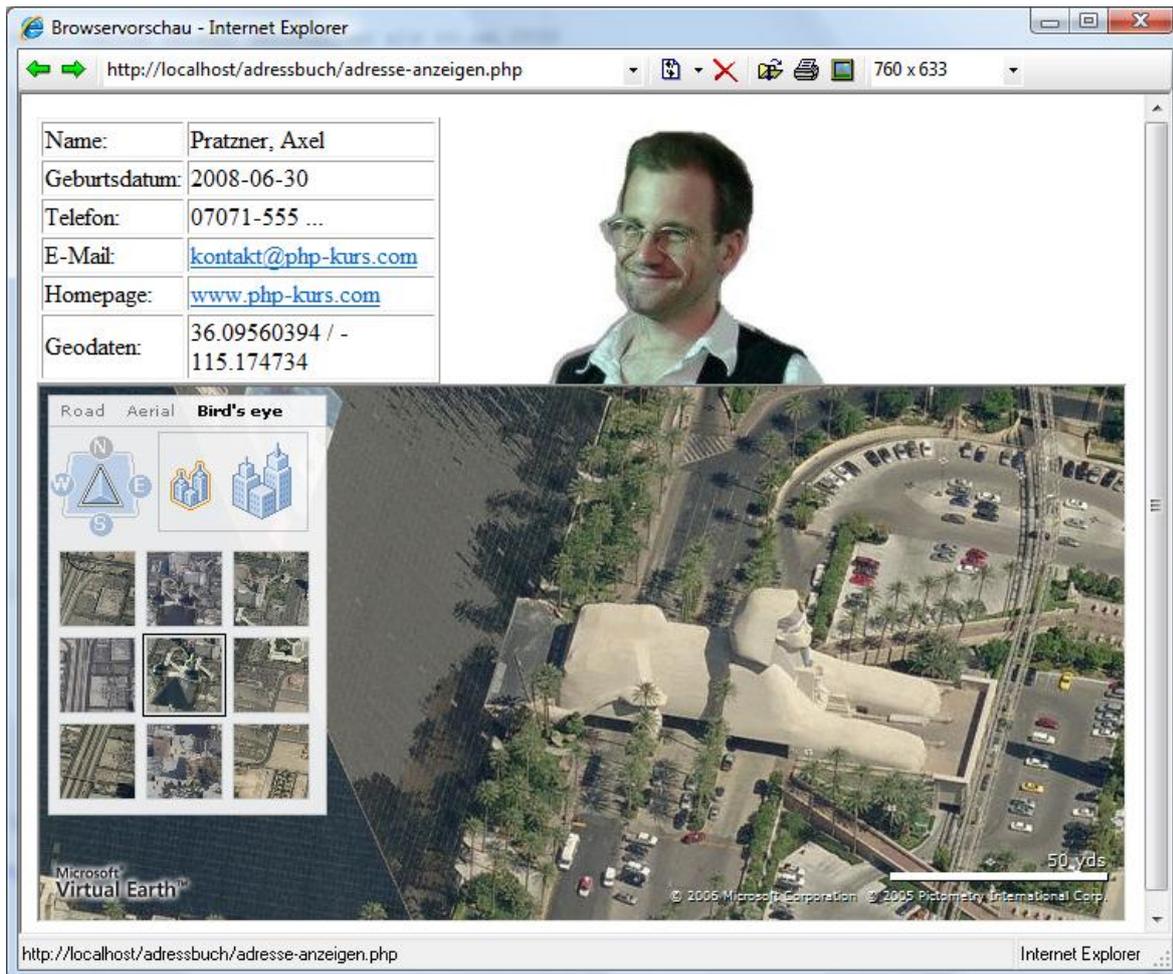
Beispiel persönliche Adressbuch programmieren mit Geodaten

Im folgenden Beispiel werden wir ein persönliches Adressbuch programmieren. Neben der Adresse kann eine URL (für ein Bild hinterlegt werden), die Telefonnummer, die E-Mail-Adresse (um direkt eine E-Mail zu senden) und die Geo-Koordinaten (um direkt eine Landkarte anzeigen zu können).



In dem Adressbuch besteht die Möglichkeit zur Detailansicht - hier wird das Bild angezeigt, die Geodaten anhand von Microsoft virtual earth und einen Link zur Homepage angeboten und die Möglichkeit, direkt eine E-Mail an diese Person zu senden.

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen



Die notwendigen Datenfelder werden hier aufgelistet, für das Beispiel werden aber nur die zentralen Datenfelder genutzt (ob nun die Straße, die PLZ, die Schuhgröße etc. mit programmiert wird, ist primär Fingerübung und hat wenig Mehrwert beim Lernen – kann aber später natürlich erweitert werden).

Welche Informationen für unser selber programmiertes Adressbuch werden benötigt?

verw.	Bezeichnung	Größe	Art
Ja	nachname150	text	
Ja	vorname	100	text
Ja	anrede	1	char
	strasse	150	text
	plz	10	text
	ort	100	text
Ja	geburtsdatum	10	date
Ja	telefon	20	text
	handy	20	text
	telefon-arbeit	20	text
Ja	email	150	text
Ja	homepage100	text	
Ja	url-bild250	text	

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Ja	geo-long	20	text
Ja	geo-lat	20	text

Datenbank anlegen für Adressbuch in MySQL

Im ersten Schritt legen wir die Datenbank in MySQL an. Dazu Felder direkt in PHPmyAdmin anlegen.

...

oder die faule Variante...

```
CREATE TABLE `adressbuch` (  
  `id` INT( 10 ) NOT NULL AUTO_INCREMENT PRIMARY KEY ,  
  `nachname` VARCHAR( 150 ) NULL ,  
  `vorname` VARCHAR( 100 ) NULL ,  
  `anrede` CHAR( 1 ) NULL ,  
  `geburtsdatum` DATE NULL ,  
  `telefon` VARCHAR( 20 ) NULL ,  
  `email` VARCHAR( 150 ) NULL ,  
  `homepage` VARCHAR( 100 ) NULL ,  
  `url_bild` VARCHAR( 250 ) NULL ,  
  `geo_long` VARCHAR( 20 ) NULL ,  
  `geo_lat` VARCHAR( 20 ) NULL  
) ENGINE = MYISAM ;
```

Zum Testen geben wir eine Adresse über phpMyAdmin ein: Die Geo-Daten erhält man am schnellsten über <http://www.geonames.org> und nach dem Heraussuchen des Ortes dann auf <http://www.tagzania.com/near/48.50129929277658/8.997311890125275/>

für Axels: 36.095605 / -115.174734

```
INSERT INTO `adressbuch` ( `id` , `nachname` , `vorname` , `anrede` ,  
  `geburtsdatum` , `telefon` , `email` , `homepage` , `url-bild` , `geo-long` ,  
  `geo-lat` )  
VALUES (  
  NULL , 'Pratzner', 'Axel', 'h', NULL , '07071-555 ...', NULL , 'www.php-  
kurs.com', NULL , '48.50129929277658', '8.997311890125275'  
);  
  
INSERT INTO `adressbuch` ( `id` , `nachname` , `vorname` , `anrede` ,  
  `geburtsdatum` , `telefon` , `email` , `homepage` , `url-bild` , `geo-long` ,  
  `geo-lat` )  
VALUES (  
  NULL , 'Zocker', 'Zapford', 'h', NULL , '07171-555 ...', NULL , 'www.webmaster-  
crashkurs.de', NULL , '36.095605', '-115.174734'  
);
```

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Im ersten Schritt bauen wir die Verbindung zu Datenbank auf. Das geschieht in der selben Weise, wie bereits gezeigt im Kapitel: <http://www.php-kurs.com/mysql-datenbank-verbundung-herstellen.htm>

Dateiname für folgende PHP-Code ist „konfiguration.php“:

```
<?php
error_reporting(E_ALL);

// Zum Aufbau der Verbindung zur Datenbank
define ( 'MYSQL_HOST', 'localhost' );
define ( 'MYSQL_BENUTZER', 'root' );
define ( 'MYSQL_KENNWORT', '' );
define ( 'MYSQL_DATENBANK', 'db-adressbuch' );

$db_link = mysql_connect (MYSQL_HOST, MYSQL_BENUTZER, MYSQL_KENNWORT);

if ( $db_link )
{
    echo 'Verbindung erfolgreich: ';
    echo $db_link;
}
else
{
    // hier sollte dann später dem Programmierer eine
    // E-Mail mit dem Problem zukommen gelassen werden
    die('keine Verbindung möglich: ' . mysql_error());
}

mysql_select_db( MYSQL_DATENBANK )
    or die("Auswahl der Datenbank fehlgeschlagen");
?>
```

Jetzt lesen wir die bestehenden Daten aus und zeigen diese in einer Tabelle an:

```
<?php
require_once ('konfiguration.php');

// Nutzen von Datenbank (Name ist hinterlegt in Konstante MYSQL_DATENBANK)
$db_sel = mysql_select_db( MYSQL_DATENBANK )
    or die("Auswahl der Datenbank fehlgeschlagen");

$sql = " SELECT * FROM adressbuch ";
$db_erg = mysql_query( $sql );
if ( ! $db_erg )
{
    die('Ungültige Abfrage: ' . mysql_error());
}

echo '<table border="1">';
while ( $zeile = mysql_fetch_array( $db_erg, MYSQL_ASSOC) )
{
    echo "<tr>";
    // hier kommt noch was :)

    echo "<td>". $zeile['id'] . "</td>";
    echo "<td>". $zeile['nachname'] . "</td>";
    echo "<td>". $zeile['vorname'] . "</td>";
}
```

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

```

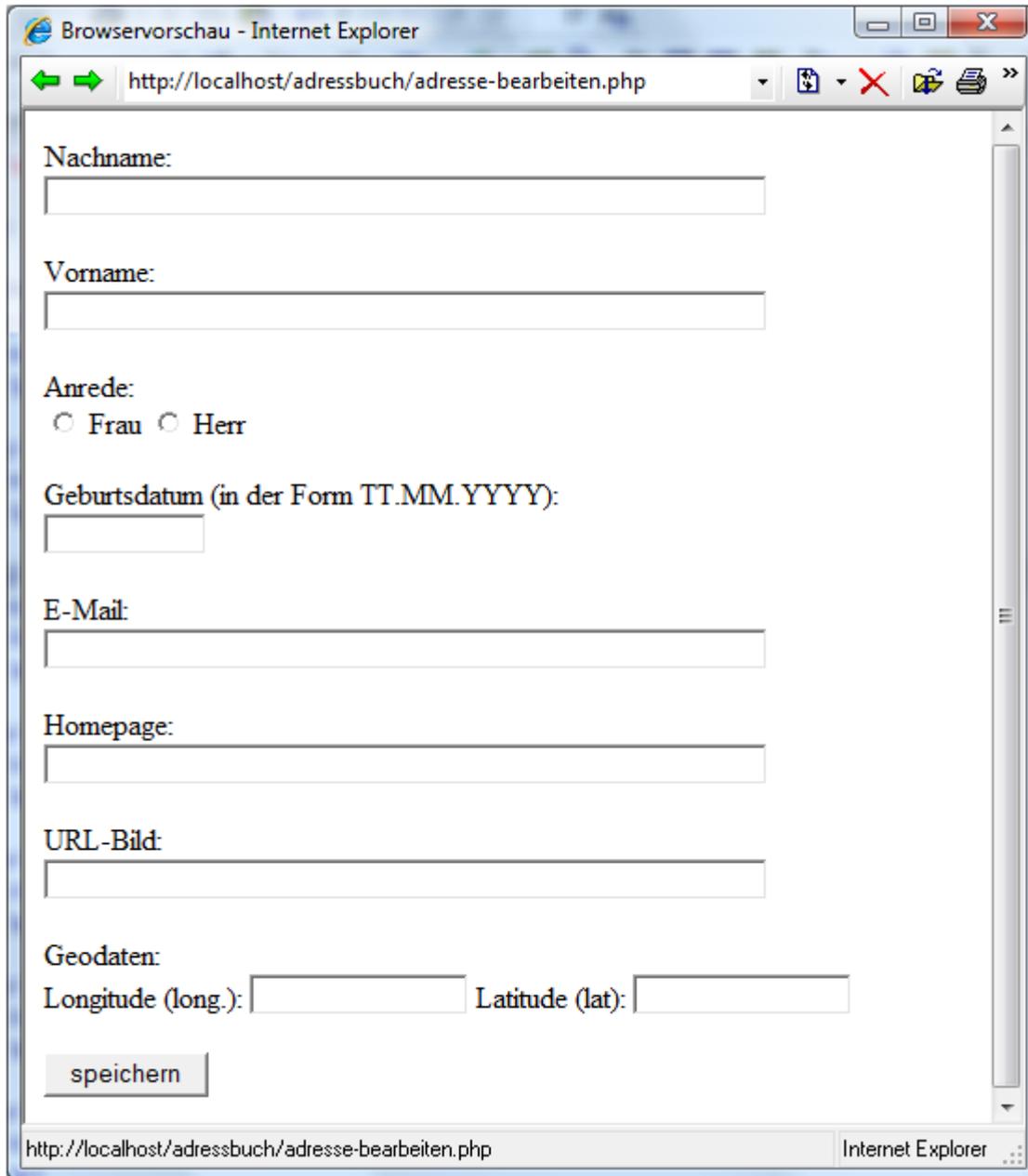
echo "<td>". $zeile['anrede'] . "</td>";
echo "<td>". $zeile['geburtsdatum'] . "</td>";
echo "<td>". $zeile['telefon'] . "</td>";
echo "<td>". $zeile['email'] . "</td>";
echo "<td>". $zeile['homepage'] . "</td>";
echo "<td>". $zeile['url_bild'] . "</td>";
echo "<td>". $zeile['geo_long'] . "</td>";
echo "<td>". $zeile['geo_lat'] . "</td>";
echo "</tr>";
}
echo "</table>";
mysql_free_result( $db_erg );
?>

<a href="adresse-bearbeiten.php">neue Adresse anlegen</a>

```

Es soll nun möglich sein, eine weitere Adresse anzulegen. Dazu blenden wir nach der Tabelle einen Link ein, der die PHP-Datei zum Anlegen von neuen Adressen aufruft.

In der Datei „adresse-bearbeiten.php“ zum Anlegen einer neuen Adresse wird ein Formular für alle notwendigen Feldern erstellt:



```
<form name="" action="" method="GET" enctype="text/html">
<p>Nachname:<br />
<input type="text" name="nachname" value="" size="60" />
</p>

<p>Vorname:<br />
<input type="text" name="vorname" value="" size="60" />
</p>

<p>Anrede:<br />
<input type="radio" name="anrede" value="w" /> Frau
<input type="radio" name="anrede" value="m" /> Herr
</p>

<p>Geburtsdatum (in der Form YYYY-MM-DD):<br />
<input type="text" name="geburtsdatum" value="" size="10" />
```

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

```

</p>

<p>EMail:<br />
<input type="text" name="email" value="" size="60" />
</p>

<p>Homepage:<br />
<input type="text" name="homepage" value="" size="60" />
</p>

<p>URLBild:<br />
<input type="text" name="url_bild" value="" size="60" />
</p>

<p>Geodaten:<br />
Longitude (long.): <input type="text" name="geo_long" value="" size="15" />
Latitude (lat): <input type="text" name="geo_lat" value="" size="15" />
</p>

<input type="hidden" name="aktion" value="speichern" />

<input type="Submit" name="" value="speichern" />
</form>

```

Dieses Formular ruft sich selber auf ein speichert eingegebene Daten in der MySQL-Datenbank:

Dazu wird der Kopf erweitert und kontrolliert, ob eine Eingabe gemacht wurde, der entsprechende SQL-Befehl zusammengebaut (zur Kontrolle ausgegeben) und der MySQL-Datenbank übergeben:

```

<?php

if ( $_GET['aktion'] == "speichern" )
{
    $nachname      = $_GET['nachname'];
    $vorname       = $_GET['vorname'];
    $anrede        = $_GET['anrede'];
    $geburtsdatum  = $_GET['geburtsdatum'];
    $email         = $_GET['email'];
    $homepage      = $_GET['homepage'];
    $url_bild      = $_GET['url_bild'];
    $geo_long      = $_GET['geo_long'];
    $geo_lat       = $_GET['geo_lat'];

    $sql = "INSERT INTO adressbuch SET ";
    $sql .= " nachname      = '$nachname', ";
    $sql .= " vorname       = '$vorname', ";
    $sql .= " anrede        = '$anrede', ";
    $sql .= " geburtsdatum  = '$geburtsdatum', ";
    $sql .= " email         = '$email', ";
    $sql .= " homepage      = '$homepage', ";
    $sql .= " url_bild      = '$url_bild', ";
    $sql .= " geo_long      = '$geo_long', ";
    $sql .= " geo_lat       = '$geo_lat' ";

    // Zur Kontrolle lassen wir uns den zusammgebauten SQLBefehl
    // auf dem Bildschirm anzeigen
    echo "<hr />";
}

```

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

```

echo "SQL: $sql";
echo "<hr />";

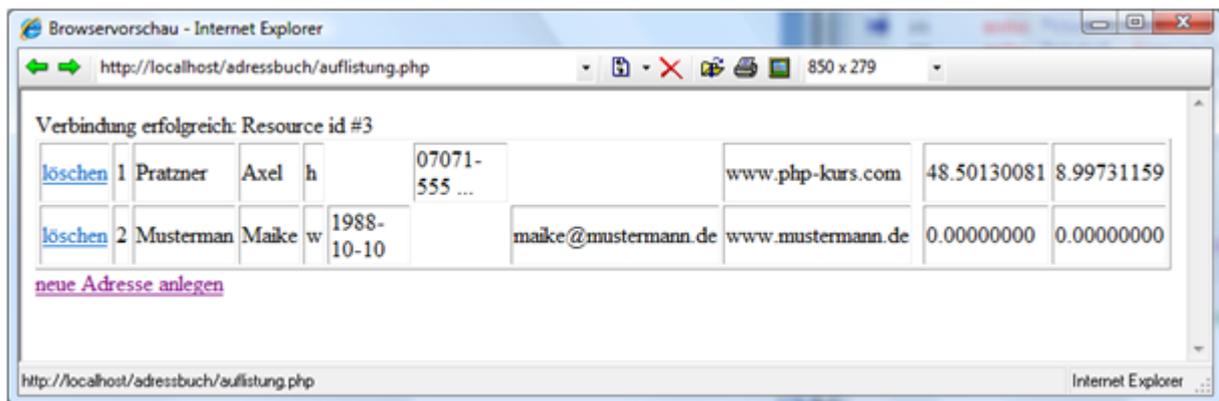
require_once ('konfiguration.php');
$db_erg = mysql_query($sql)
    or die("Anfrage fehlgeschlagen: " . mysql_error());

echo '<h1>Adresse wurde speichert</h1>';
echo '<a href="auflistung.php">Auflistung anzeigen</a>';
exit;
}
?>

<form name="" action="" method="GET" enctype="text/html">
<p>Nachname:<br />

```

Wir wollen jetzt eine Adresse löschen können – dazu erweitern wir die Ausgabe der Tabelle mit einer weiteren Zelle am Anfang, in der wir die Aktionen für den entsprechenden Datensatz unterbringen:



```

<?php
require_once ('konfiguration.php');

// Nutzen von Datenbank (Name ist hinterlegt in Konstante MYSQL_DATENBANK)
$db_sel = mysql_select_db( MYSQL_DATENBANK )
    or die("Auswahl der Datenbank fehlgeschlagen");

$sql = " SELECT * FROM adressbuch ";
$db_erg = mysql_query( $sql );
if ( ! $db_erg )
{
    die('Ungültige Abfrage: ' . mysql_error());
}

echo '<table border="1">';
while ($zeile = mysql_fetch_array( $db_erg, MYSQL_ASSOC))
{
    echo "<tr>";
    echo "<td>";
    echo '<a href="adresse-loeschen.php?id=' . $zeile['id'] . '">löschen</a> ';
    echo "</td>";
    echo "<td>". $zeile['id'] . "</td>";
    echo "<td>". $zeile['nachname'] . "</td>";
    echo "<td>". $zeile['vorname'] . "</td>";
    echo "<td>". $zeile['anrede'] . "</td>";
}

```

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

```

        echo "<td>". $zeile['geburtsdatum'] . "</td>";
        echo "<td>". $zeile['telefon'] . "</td>";
        echo "<td>". $zeile['email'] . "</td>";
        echo "<td>". $zeile['homepage'] . "</td>";
        echo "<td>". $zeile['url_bild'] . "</td>";
        echo "<td>". $zeile['geo_long'] . "</td>";
        echo "<td>". $zeile['geo_lat'] . "</td>";
        echo "</tr>";
    }
    echo "</table>";
    mysql_free_result( $db_erg );
    ?>

<a href="adresse-bearbeiten.php">neue Adresse anlegen</a>

```

In der Datei „adresse-loeschen.php“ wird über die id den zu löschenden Datensatz übergeben:
wir bauen den entsprechenden SQL-Befehl zusammen:

```

<?php
$id = $_GET['id'];
$sql = "DELETE FROM adressbuch WHERE id='$id' ";

// Zur Kontrolle lassen wir uns den zusammengebauten SQLBefehl
// auf dem Bildschirm anzeigen
echo "<hr />";
echo "SQL: $sql";
echo "<hr />";

require_once ('konfiguration.php');
$db_erg = mysql_query($sql)
    or die("Anfrage fehlgeschlagen: " . mysql_error());

echo '<h1>Adresse $id wurde gelöscht</h1>';
echo '<a href="auflistung.php">Auflistung anzeigen</a>';
?>

```

Zusatzaufgabe: für die Schnellen – erstellen Sie eine Sicherheitsabfrage, in der der Inhalt des Datensatzes angezeigt wird und nochmals explizit auf löschen geklickt werden muss!

Datensatz anzeigen mit Geo-Koordinaten und Microsoft Virtual Earth

Wenn jetzt eine Adresse mit allen Daten und den Geo-Koordinaten direkt in Microsoft Virtual Earth angezeigt werden soll, erweitern wir in der auflistung.php die Aktionen um anzeigen:

```

<?php
require_once ('konfiguration.php');

// Nutzen von Datenbank (Name ist hinterlegt in Konstante MYSQL_DATENBANK)
$db_sel = mysql_select_db( MYSQL_DATENBANK )
    or die("Auswahl der Datenbank fehlgeschlagen");

$sql = " SELECT * FROM adressbuch ";

```

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

```

$db_erg = mysql_query( $sql );
if ( ! $db_erg )
{
    die('Ungültige Abfrage: ' . mysql_error());
}

echo '<table border="1">';
while ($zeile = mysql_fetch_array( $db_erg, MYSQL_ASSOC))
{
    echo "<tr>";
    echo "<td>";
    echo '<a href="adresse-loeschen.php?id=' . $zeile['id'] . '">löschen</a> ';
    echo '<a href="adresse-anzeigen.php?id=' . $zeile['id'] . '">anzeigen</a> ';
    echo '<a href="mail-senden.php?id=' . $zeile['id'] . '">Mail</a> ';
    echo "</td>";
    echo "<td>". $zeile['id'] . "</td>";
    echo "<td>". $zeile['nachname'] . "</td>";
    echo "<td>". $zeile['vorname'] . "</td>";
    echo "<td>". $zeile['anrede'] . "</td>";
    echo "<td>". $zeile['geburtsdatum'] . "</td>";
    echo "<td>". $zeile['telefon'] . "</td>";
    echo "<td>". $zeile['email'] . "</td>";
    echo "<td>". $zeile['homepage'] . "</td>";
    echo "<td>". $zeile['url_bild'] . "</td>";
    echo "<td>". $zeile['geo_long'] . "</td>";
    echo "<td>". $zeile['geo_lat'] . "</td>";
    echo "</tr>";
}
echo "</table>";
mysql_free_result( $db_erg );
?>

<a href="adresse-bearbeiten.php">neue Adresse anlegen</a>

```

Und nun sauber aufbereiten in `adresse-anzeigen.php`

```

<?php
$id = $_GET['id'];
$sql = " SELECT * FROM adressbuch WHERE id='$id' ";

require_once ('konfiguration.php');
$db_erg = mysql_query( $sql );
if ( ! $db_erg )
{
    die('Ungültige Abfrage: ' . mysql_error());
}

while ($zeile = mysql_fetch_array( $db_erg, MYSQL_ASSOC))
{
    $nachname      = $zeile['nachname'];
    $vorname       = $zeile['vorname'];
    $anrede        = $zeile['anrede'];
    $geburtsdatum = $zeile['geburtsdatum'];
    $telefon       = $zeile['telefon'];
    $email         = $zeile['email'];
    $homepage      = $zeile['homepage'];
    $url_bild      = $zeile['url_bild'];
    $geo_long      = $zeile['geo_long'];
    $geo_lat       = $zeile['geo_lat'];
}

```

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

```

mysql_free_result( $db_erg );

if ( $url_bild <> "" )
{
    echo '';
}

echo '<table border="1">';
echo '<tr>';
echo '<td>';
echo "Name:";
echo '</td>';
echo '<td>';

    // Aufbau Nachname, Vorname Anrede
echo $nachname;
echo ", ";
echo $vorname;
echo " ";
if ( $anrede == "m" )
{
    echo "Herr";
}
if ( $anrede == "w" )
{
    echo "Frau";
}

echo '</td>';
echo '</tr>';

echo '<tr>';
echo '<td>';
echo "Geburtsdatum:";
echo '</td>';
echo '<td>';
// für die schnellen, Geburtsdatum sauber darstellen als tt.mm.YYYY
echo $geburtsdatum;
echo '</td>';
echo '</tr>';

echo '<tr>';
echo '<td>';
echo "Telefon:";
echo '</td>';
echo '<td>';
echo $telefon;
echo '</td>';
echo '</tr>';

echo '<tr>';
echo '<td>';
echo "E-Mail:";
echo '</td>';
echo '<td>';
echo $email;
echo '</td>';
echo '</tr>';

echo '<tr>';

```

```

    echo '<td>';
    echo "Homepage:";
    echo '</td>';
    echo '<td>';
    echo '<a href="http://';
    echo $homepage;
    echo '" target="_blank">';
    echo $homepage;
    echo '</a>';
    echo '</td>';
    echo '</tr>';

    echo '<tr>';
    echo '<td>';
    echo "Geodaten:";
    echo '</td>';
    echo '<td>';
    echo $geo_long;
    echo " / ";
    echo $geo_lat;
    echo '</td>';
    echo '</tr>';

echo "</table>";

?>

```

Als Bonbon zur Demonstration von Internet-Services binden wir nun eine Geokarten ein von Microsoft Earth:

Dazu wird die bereits vorbereitete Datei „geomap.php“ verwendet. Diese Datei erwartet als übergabewerte die Parameter „longitude“ und „latitude“ und „zoom“ und „ansicht“ Diese wird nun über einen iframe „eingebunden“.

geomap.php

```

<?php

if ( $_REQUEST['zoom'] == "" )
{
    $_REQUEST['zoom'] = 1;
}
if ( $_REQUEST['longitude'] == "" )
{
    $_REQUEST['longitude'] = '36.095605';
}
if ( $_REQUEST['latitude'] == "" )
{
    $_REQUEST['latitude'] = '-115.174734';
}
?>
<head>
<script src="http://www.php-
kurs.com/http://dev.virtualearth.net/mapcontrol/v3/mapcontrol.js"></script>
<script>
var map = null;
function GetMap()

```

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

```

{
    map = new VEMap('myMap');
    map.LoadMap(new VELatLong(<?php echo $_REQUEST['longitude'] ?>,<?php echo
$_REQUEST['latitude'] ?>), <?php echo $_REQUEST['zoom'] ?>, "<?php

if ($_REQUEST['ansicht'] == "satellit" )
{
    echo "o";
}
else
{
    echo "h";
    // echo "o";
}

?>", false);
}

</script>

</head>
<body onload="GetMap();">

<div id="myMap" style="width:100%; height:100%"></div>
</body>

```

Beispiel iframe:

```

<iframe width="100%" height="350"
src="http://www.php-kurs.com/geomap.php?longitude=36.095605&latitude=-
115.174734&zoom=1&ansicht=satellit">
</iframe>

```

Erweitert für unsere Beispielanzeige:

```

<?php
$id = $_GET['id'];
$sql = " SELECT * FROM adressbuch WHERE id='$id' ";

require_once ('konfiguration.php');
$db_erg = mysql_query( $sql );
if ( ! $db_erg )
{
    die('Ungültige Abfrage: ' . mysql_error());
}

while ($zeile = mysql_fetch_array( $db_erg, MYSQL_ASSOC))
{
    $nachname      = $zeile['nachname'];
    $vorname       = $zeile['vorname'];
    $anrede        = $zeile['anrede'];
    $geburtsdatum  = $zeile['geburtsdatum'];
    $telefon       = $zeile['telefon'];
    $email         = $zeile['email'];
    $homepage      = $zeile['homepage'];
    $url_bild      = $zeile['url_bild'];
    $geo_long      = $zeile['geo_long'];
}

```

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

```

    $geo_lat      = $zeile['geo_lat'];
}
mysql_free_result( $db_erg );

if ( $url_bild <> "" )
{
    echo '';
}

echo '<table border="1">';
echo '<tr>';
echo '<td>';
echo "Name:";
echo '</td>';
echo '<td>';

    // Aufbau Nachname, Vorname Anrede
echo $nachname;
echo ", ";
echo $vorname;
echo " ";
if ( $anrede == "m" )
{
    echo "Herr";
}
if ( $anrede == "w" )
{
    echo "Frau";
}

echo '</td>';
echo '</tr>';

echo '<tr>';
echo '<td>';
echo "Geburtsdatum:";
echo '</td>';
echo '<td>';
// für die schnellen, Geburtsdatum sauber darstellen als tt.mm.YYYY
echo $geburtsdatum;
echo '</td>';
echo '</tr>';

echo '<tr>';
echo '<td>';
echo "Telefon:";
echo '</td>';
echo '<td>';
echo $telefon;
echo '</td>';
echo '</tr>';

echo '<tr>';
echo '<td>';
echo "E-Mail:";
echo '</td>';
echo '<td>';
echo $email;
echo '</td>';
echo '</tr>';

```

```

echo '<tr>';
  echo '<td>';
  echo "Homepage:";
  echo '</td>';
  echo '<td>';
  echo '<a href="http://';
  echo $homepage;
  echo '" target="_blank">';
  echo $homepage;
  echo '</a>';
  echo '</td>';
echo '</tr>';

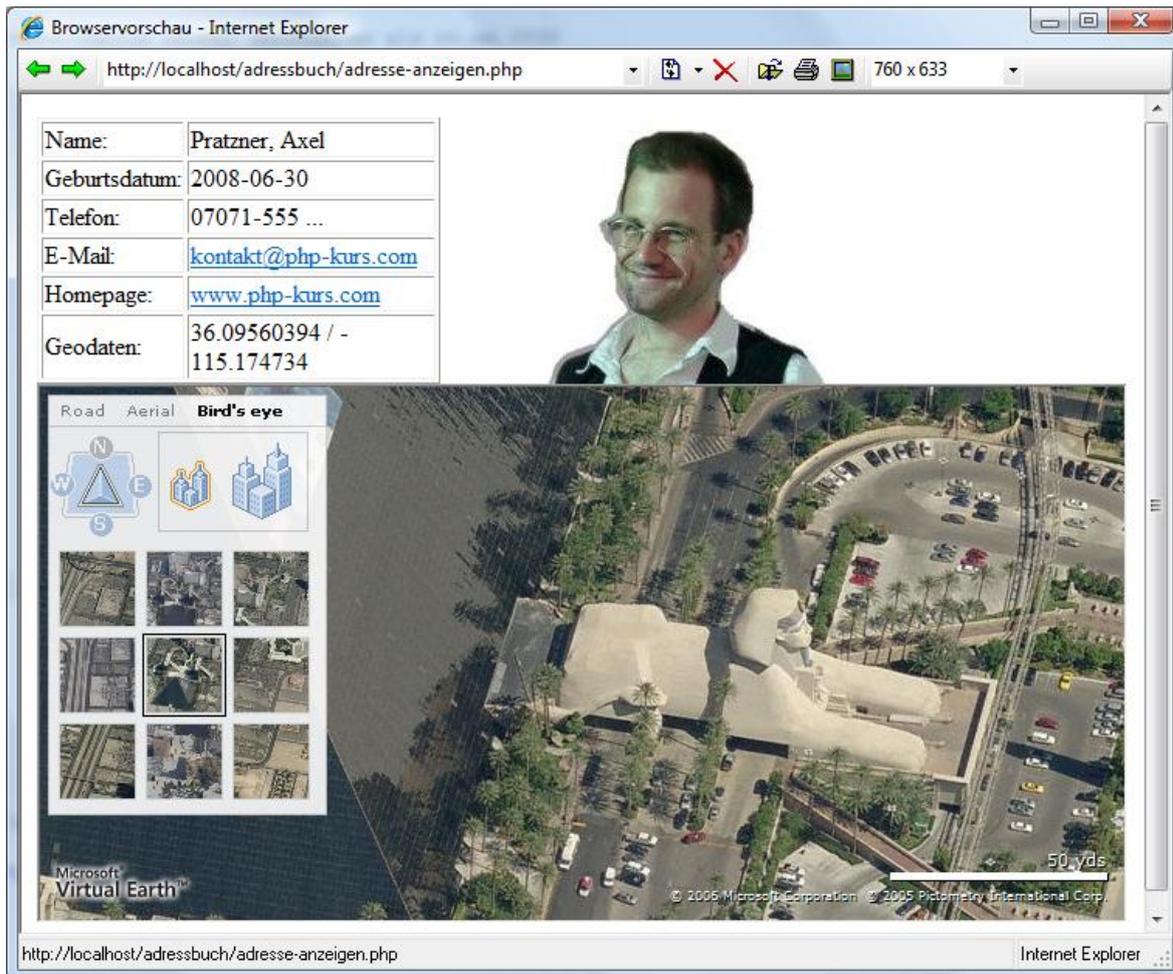
echo '<tr>';
  echo '<td>';
  echo "Geodaten:";
  echo '</td>';
  echo '<td>';
  echo $geo_long;
  echo " / ";
  echo $geo_lat;
  echo '</td>';
echo '</tr>';

echo "</table>";

if ($geo_long <> "" AND $geo_lat <> "")
{
  echo '<iframe width="100%" height="350" src="http://www.php-
kurs.com/geomap.php?longitude=';
  echo $geo_long;
  echo '&latitude=';
  echo $geo_lat;
  echo '&zoom=1&ansicht=satellit"></iframe>';
}

?>

```



Zusätzlich die Möglichkeit, eine E-Mail zu senden:

```
<?php
if ( $_POST['aktion'] == 'senden' )
{
    // hier kommt der entsprechende Mail-Befehl
    // mail .....
    // $_POST['absender']
    // $_POST['empfaenger']
    // $_POST['betreff']
    // $_POST['inhalt']
    echo '<a href="auflistung.php">Auflistung anzeigen</a>';
    exit;
}

$id = $_GET['id'];
$sql = " SELECT * FROM adressbuch WHERE id='$id' ";

require_once ('konfiguration.php');
$db_erg = mysql_query( $sql );
if ( ! $db_erg )
{
    die('Ungültige Abfrage: ' . mysql_error());
}
```

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

```

}

while ($zeile = mysql_fetch_array( $db_erg, MYSQL_ASSOC))
{
    $nachname      = $zeile['nachname'];
    $vorname       = $zeile['vorname'];
    $anrede        = $zeile['anrede'];
    $geburtsdatum  = $zeile['geburtsdatum'];
    $telefon       = $zeile['telefon'];
    $email         = $zeile['email'];
    $homepage      = $zeile['homepage'];
    $url_bild      = $zeile['url_bild'];
    $geo_long      = $zeile['geo_long'];
    $geo_lat       = $zeile['geo_lat'];
}
mysql_free_result( $db_erg );

echo '<form name="" action="" method="POST" enctype="text/html">';

echo '<p>';
echo 'Absender<br />';
echo '<input type="text" name="absender" value="';
echo 'axel@example.com';
echo '" size="50" maxlength="150" />';
echo '</p>';

echo '<p>';
echo 'Empfänger<br />';
echo '<input type="text" name="empfaenger" value="';
echo $email;
echo '" size="50" maxlength="150" />';
echo '</p>';

echo '<p>';
echo 'Betreff<br />';
echo '<input type="text" name="betreff" value="" size="50" maxlength="150" />';
echo '</p>';

echo '<p>';
echo 'Inhalt<br />';
echo '<textarea name="inhalt" rows="15" cols="80">';
echo 'Hallo '. $vorname . " " . $nachname . ",";
echo '</textarea>';

echo '<input type="hidden" name="aktion" value="senden" />';
echo '<input type="Submit" name="" value="Save" />';
echo '</form>';
?>

```

Beispiel Terminkalender programmieren

Im Folgenden programmieren wir uns einen Veranstaltungskalender für die eigene Website. Die Termine werden in einer MySQL-Datenbank abgelegt. Der Kalender soll untereinander alle Termine für den aktuellen Monat anzeigen und einen Link zu dem kommenden Monat und zum Vormonat bieten.

Eintragen, Löschen und Ändern der Termine soll möglich sein.

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Vorüberlegungen zum PHP-Kalender:

Wir benötigen in der Datenbank folgende Felder:

id: eine eindeutige Nummer, um die Einträge verwalten zu können

datum: an welchem Datum ist der Termin

titel: Für die Kurzanzeige der Titel

beschreibung: großes Textfeld, das alle Informationen zum Termin enthält

Wie legen also als erstes in phpMyAdmin unsere Datenbank Terminkalender an: `CREATE DATABASE `terminkalender` ;`

Und dann legen wir unsere MySQL-Tabelle mit dem Namen "termine" an:

```
CREATE TABLE `termine` (  
  `id` INT( 10 ) NOT NULL AUTO_INCREMENT PRIMARY KEY ,  
  `datum` DATE NOT NULL ,  
  `titel` VARCHAR( 256 ) NOT NULL ,  
  `beschreibung` BLOB NOT NULL  
) ENGINE = MYISAM ;
```

Zum Testen tragen wir von Hand ein paar Termine ein. Später werden neue Termine über ein Formular eingetragen.

```
INSERT INTO `termine` ( `id` , `datum` , `titel` , `beschreibung` )  
VALUES (  
  NULL , '2007-08-10', 'PHP-Kurs.com besuchen ab 9:00', ''  
) , (  
  NULL , '2007-08-16', 'Geburtstag Elke', ''  
) ;
```

Um jetzt unsere erste Anzeige zu erhalten (wenn es keine Angabe gibt, alle Termine des aktuellen Monats), müssen wir als erstes eine Verbindung zu unserer Datenbank herstellen. Diese Datei nennen wir "termin-konfiguration.php".

Es werden alle benötigten Daten hinterlegt (die MySQL-Zugangsdaten bekommen Sie normalerweise von Ihrem Provider beim Anlegen der Datenbank).

```
define ( 'MYSQL_HOST', 'localhost' );  
define ( 'MYSQL_BENUTZER', 'root' );  
define ( 'MYSQL_KENNWORT', '' );  
define ( 'MYSQL_DATENBANK', 'terminkalender' );
```

Nun bauen wir die Verbindung zu Datenbank auf – dazu wird `mysql_connect` die bereits definierten Konstanten übergeben.

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Das @ vor mysql_connect unterbindet eventuelle Fehlermeldungen, die den Nutzer nur verunsichern – sollte der Verbindungsaufbau nicht klappen, überprüfen wir es anhand der anschließenden if-Abfrage.

```
$db_link = @mysql_connect (MYSQL_HOST, MYSQL_BENUTZER, MYSQL_KENNWORT);
if ( ! $db_link )
{
// hier sollte dann später dem Programmierer eine
// E-Mail mit dem Problem zukommen gelassen werden
// die Fehlermeldung für den Programmierer sollte
// das Problem ausgeben mit: mysql_error()
die('keine Verbindung zur Zeit möglich - später probieren ');
}
```

Beide obigen PHP-Code-Teile in der Datei "termin-konfiguration.php" abspeichern. Beide am Stück:

```
<?php
define ( 'MYSQL_HOST', 'localhost' );
define ( 'MYSQL_BENUTZER', 'root' );
define ( 'MYSQL_KENNWORT', '' );
define ( 'MYSQL_DATENBANK', 'terminkalender' );

$db_link = @mysql_connect (MYSQL_HOST, MYSQL_BENUTZER, MYSQL_KENNWORT);
if ( ! $db_link )
{
// hier sollte dann später dem Programmierer eine
// E-Mail mit dem Problem zukommen gelassen werden
// die Fehlermeldung für den Programmierer sollte
// das Problem ausgeben mit: mysql_error()
die('keine Verbindung zur Zeit möglich - später probieren ');
}
?>
```

PHP-Terminkalender Hauptprogramm – Termine anzeigen

Es werden nun alle Termine für einen Monat angezeigt. Sollte kein Monat angegeben sein, wird der aktuelle Monat verwendet. Unsere neue Datei nennen wir terminanzeige.php

```
// wenn ein Monat angegeben ist, wird die Echtheit der Daten überprüft
if ( $_GET['monat'] )
{
if ( (INT) $_GET['monat'] > 0 AND (INT) $_GET['monat'] < 13 )
{
$monat = (INT) $_GET['monat'];
}
else
{
// something is wrotten - es wird aktuelle Monat verwendet
$monat = date("n");
}
}
else
{
// ist kein Monat angegeben, wird der aktuelle Monat verwendet
```

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

```
$monat = date("n");
}
```

Dasgleiche für das Jahr

```
if ( $_GET['jahr'] )
{
if ( (INT) $_GET['jahr'] > 0 AND (INT) $_GET['jahr'] < 3000 )
{
$jahr = (INT) $_GET['jahr'];
}
else
{
$jahr = date("Y");
}
}
else
{
$jahr = date("Y");
}
}
```

Und für den Besucher (und für den Programmierer zu Kontrolle) die Ausgabe von Monat und Jahr.

```
// Ausgabe vom Monat und dem Jahr
echo "<h1>Terminkalender $monat.$jahr</h1>";
```

Über die MySQL-Befehl findet die Eingrenzung der Termine auf den gewünschten Monat und das gewünschte Jahr statt.

```
$sql = "
SELECT
id, datum, titel, beschreibung
FROM termine
WHERE
YEAR(datum) = '$jahr'
AND
MONTH(datum) = '$monat'
ORDER BY datum
";
```

Die SQL-Anweisung wird über `mysql_query` ausgeführt – gefolgt von einer Sicherheitsabfrage, ob die Ausführung funktioniert hat.

```
$db_erg = mysql_query( $sql );
if ( ! $db_erg )
{
die('Ungültige Abfrage: ' . mysql_error());
}
```

Es erfolgt die Ausgabe der Ausgabe der Anzahl der Termine – die Ermittlung funktioniert über `mysql_num_rows`, wobei bereits die `mysql_query`-Abfrage im Vorfeld erfolgt sein muss.

```
$anzahl_eintraege = mysql_num_rows($db_erg);
echo "<p>Anzahl der Einträge: $anzahl_eintraege </p>";
```

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Wenn es nun mindestens einen Termin gibt, erfolgt die Ausgabe in einer Tabelle (hier kann man bestimmt sich noch am Design austoben – hier zählt erstmal die Funktion :))

Die von der Datenbank gelieferten Ergebnisse werden über `mysql_fetch_array` in einer Array übertragen und per while-Schleife solange durchlaufen, bis alle Einträge ausgegeben worden sind.

```
echo '<table border="1">';
while ( $zeile = mysql_fetch_array( $db_erg, MYSQL_ASSOC) )
{
echo '<tr>';
echo '<td>'. $zeile['datum'] . '</td>';
echo '<td>';
echo '<b>'. $zeile['titel'] . '</b><br />';
echo $zeile['beschreibung'];
echo '</td>';
echo '</tr>';
}
echo '</table>';
```

Jetzt wird der Speicher geleert und die Verbindung zum MySQL beendet mit:

```
mysql_free_result( $db_erg );
```

Im letzten Schritt möchten wir noch 3 Links haben – einen zum Anzeigen des Folgemonats, einen zum Sprung zum aktuellen Monat, einen Link zum vergangenen Monat. Interessant wird es bei den Jahresübergängen.

```
// Vormonat Kontrolle, ob bereits Januar
if ( $monat == 1 )
{
$vmonat = 12;
$vjahr = bcsub ( $jahr, 1 );
}
else
{
$vmonat = bcsub ( $monat, 1 );
$vjahr = $jahr;
}

echo '<a href="terminanzeige.php';
echo '?monat=' . $vmonat;
echo '&jahr=' . $vjahr;
echo ">Vormonat</a>';

echo ' | ';

echo '<a href="terminanzeige.php';
echo '?monat=' . date("n");
echo '&jahr=' . date("Y");
echo ">akt. Monat</a>';

echo ' | ';

// nächster Monat - Kontrolle, ob bereits Dezember
if ( $monat == 12 )
{
$nmonat = 1;
```

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

```

$jjahr = bcadd ( $jahr, 1 );
}
else
{
$monat = bcadd ( $monat, 1 );
$jjahr = $jahr;
}
echo '<a href="terminanzeige.php';
echo '?monat=' . $monat;
echo '&jahr=' . $njahr;
echo '">n&auml;chster Monat</a>';

```

Im nächsten Schritt sollen Termine erfasst werden. Dazu benötigen wir ein Formular mit den entsprechenden Feldern – hierbei wird nun nicht auf Sicherheit geachtet, da die Annahme zugrunde liegt, dass nur berechnigte Personen Termine erfassen, ändern und löschen dürfen. Soll dies auch für unbekannte Benutzer möglich sein (Websitebenutzer), muss unbedingt auf Sicherheit geachtet werden!

Wir geben unser Formular unter der Auflistung aus. Es werden 3 Felder benötigt – das Datumsfeld, ein Feld für die Terminbeschreibung und ein Textbereich für die ausführliche Fassung.

Dabei sollte bei der Übertragungsmethode `method="POST"` unbedingt POST gewählt werden, da sonst die Datenmenge (Textbereich für Beschreibung) für GET zu viel werden könnte.

```

// Anzeige Formular
echo '<hr />';
echo '<form name="" action="" method="POST" enctype="text/html">';

echo '<p>Datum in der Form JJJJ-MM-TT<br />';
echo '<input type="text" name="termin[datum]" value="" size="10" maxlength="10" />';
echo '</p>';

echo '<p>Kurzbeschreibung<br />';
echo '<input type="text" name="termin[kurzbeschreibung]" value="" size="50" maxlength="255" />';
echo '</p>';

echo '<p>ausführliche Beschreibung<br />';
echo '<textarea name="termin[beschreibung]" rows="10" cols="80"></textarea></p>';

echo '<input type="hidden" name="vorgang" value="neu" />';

echo '<input type="Submit" name="" value="speichern" />';
echo '</form>';

```

Im Formular gibt es ein verstecktes Feld mit dem Name "vorgang" – über dieses kann nach Absenden des Formulars angefragt werden, was mit den Daten geschehen soll.

Nun folgt vor der Ausgabe der Tabelle eine Abfrage, ob neue Daten vorhanden sind, und wenn ja, werden diese gespeichert.

```

// Speichern neuer Daten
if ( $_POST['vorgang'] == 'neu' )

```

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

```
{
speichere_daten ( $_POST['termin'] );
}
```

Das Speichern erfolgt über den Funktionsaufruf `speichere_daten ($_POST['termin']);` - durch die Übergabe der Arrays `$_POST` kann in der Funktion direkt auf die Daten zugegriffen werden. Die praktische Übergabe durch `$_POST['termin']` ist auch der Grund für die Schreibweise vorher im Formular mit `<input type="text" name="termin[datum]" value="" size="10" maxlength="10" />`.

Die Funktion nimmt die Daten und speichert diese über die SQL-Anweisung "INSERT ...". Sie sollten noch ein wenig Zeit in die Validierung der Daten stecken – hier zum Lernen soll es ohne Validierung ausreichen.

```
// Daten speichern
function speichere_daten ( $termin )
{
// hier sollte noch eine Validierung des Datums stattfinden

$sql = "INSERT INTO termine
(
`id`, `datum` , `titel` , `beschreibung`
)
VALUES
(
NULL ,
'". $termin['datum'] ."',
'". $termin['kurzbeschreibung'] ."',
'". $termin['beschreibung'] ."'
)
";

$db_erg = mysql_query( $sql );
if ( ! $db_erg )
{
die("Ungültige Abfrage: $sql <hr />" . mysql_error());
}
else
{
echo "<h2>Termin gespeichert</h2>";
}
}
```

Im nächsten Schritt wollen wir auch das Löschen von Daten ermöglichen. Dazu ergänzen wir die Tabelle noch um eine weitere Spalte, die mit Links versehen wird. In den Links wird über die ID ein eindeutiger Zugriff möglich. Auch hier findet über die Variable "vorgang" eine Steuerung statt.

```
echo '<td>';
echo '<a href="terminanzeige.php?vorgang=loeschen&id=';
echo $zeile['id'];
echo '">l&ouml;schen</a> ';
echo '</td>';
```

Am Anfang vom Programm erfolgt wieder eine Abfrage, ob die Variable "vorgang" den Inhalt "loeschen" hat. Wenn ja, erfolgt der Aufruf des Löschvorgangs über eine Funktion, der

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

die ID übergeben wird. Da der Aufruf über einen Link stattfindet, muss die Variablenabfrage mit `$_GET` stattfinden!

```
// Löschen von Einträgen
if ( $_GET['vorgang'] == 'loeschen' )
{
loeschen_daten ( $_GET['id'] );
}
```

Die entsprechende Funktion beinhaltet die SQL-Anweisung "DELETE FROM ... WHERE ... LIMIT 1"

Der Zusatz "LIMIT 1" begrenzt den Schaden :) – es wird maximal ein Datensatz gelöscht.

```
function loeschen_daten ( $id ) {
$sql = "DELETE FROM termine
WHERE id='$id'
LIMIT 1
";

$db_erg = mysql_query( $sql );
if ( ! $db_erg )
{
die("Ungültige Abfrage: $sql <hr />" . mysql_error());
}
else
{
echo "<h2>Termin gelöscht</h2>";
}
}
```

Hier können Sie den Vorgang erweitern, dass vor dem Löschen erst eine Sicherheitsabfrage stattfindet!

Daten aktualisieren

Um nun Daten ändern zu können, fügen wir in der Tabelle einen weiteren Link hinzu.

```
echo '<td>';
echo '<a href="terminanzeige.php?vorgang=aendern&id=';
echo $zeile['id'];
echo '">ändern</a> ';
```

Am Anfang der Datei erfolgt wieder eine Abfrage und der Funktionsaufruf

```
// Daten zum Ändern anzeigen
if ( $_GET['vorgang'] == 'aendern' )
{
zeige_daten_zum_bearbeiten ( $_GET['id'] );
}
```

In der Funktion wird für diesen einen Datensatz über die entsprechende SQL-Anweisung der Inhalt geholt und in einem Formular zum Ändern angezeigt

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

```

function anzeige_daten_zum_bearbeiten ($id)
{
    $sql = "
    SELECT *
    FROM termine
    WHERE id = '$id'
    ";

    $db_erg = mysql_query( $sql );
    if ( ! $db_erg )
    {
        die('Ungültige Abfrage: ' . mysql_error());
    }

    $zeile = mysql_fetch_array( $db_erg, MYSQL_ASSOC);

    echo '<form name="" action="" method="POST" enctype="text/html">';

    echo '<p>Datum in der Form JJJJ-MM-TT<br />';
    echo '<input type="text" name="termin[datum]" value=""';
    echo $zeile['datum'];
    echo '" size="10" maxlength="10" />';
    echo '</p>';

    echo '<p>Kurzbeschreibung<br />';
    echo '<input type="text" name="termin[kurzbeschreibung]" value=""';
    echo $zeile['titel'];
    echo '" size="50" maxlength="255" />';
    echo '</p>';

    echo '<p>ausführliche Beschreibung<br />';
    echo '<textarea name="termin[beschreibung]" rows="10" cols="80">';
    echo $zeile['beschreibung'];
    echo '</textarea></p>';

    echo '<input type="hidden" name="termin[id]" value=""';
    echo $zeile['id'];
    echo '" />';
    echo '<input type="hidden" name="vorgang" value="update" />';

    echo '<input type="Submit" name="" value="Änderungen speichern" />';
    echo '</form>';
    echo '<hr />';

}

```

Die Steuerung erfolgt über die beiden versteckten Formularfelder "vorgang" und "id", die am Beginn unseres Scripts abgefragt werden.

```

if ( $_POST['vorgang'] == 'update' )
{
    update_daten ( $_POST['termin'] );
}

```

Und die Funktion zum Speichern der aktualisierten Daten:

```

// Daten updaten
function update_daten ( $termin )
{
    // hier sollte noch eine Validierung des Datums stattfinden

```

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

```

$sql = "UPDATE termine SET
`datum` = '". $termin['datum'] ."',
`titel` = '". $termin['kurzbeschreibung'] ."',
`beschreibung` = '". $termin['beschreibung'] ."'
WHERE
id = '". $termin['id'] ."'
";

$db_erg = mysql_query( $sql );
if ( ! $db_erg )
{
die("Ungültige Abfrage: $sql <hr />" . mysql_error());
}
else
{
echo "<h2>Termin geupdatet</h2>";
}
}

```

Damit nach dem Speichern nicht nochmals das Formular angezeigt wird, muss die if-Anweisung für den Vorgang "aendern" ergänzt werden mit:

```

// Daten zum Ändern anzeigen
if ( $_GET['vorgang'] == 'aendern' AND $_POST['vorgang'] <> 'update' )
{
anzeige_daten_zum_bearbeiten ( $_GET['id'] );
}

```

Ente gut, alles gut und satt

Dies als "kleines" Tutorial, wie man mit Datenbanken umgehen kann. Das Beispiel in fast 300 Code-Zeilen zeigt eine prinzipielle Vorgehensweise – es fehlen noch Validierungen (damit keine falsche Daten eingegeben werden können), und die Eintrag-, Lösch- und Änderungsfunktionen sollten auf keinen Fall für fremde Websitebesucher zugänglich gemacht werden. Die Anzeige der Termine kann sehr wohl genutzt werden, da dort nur 2 Daten verwendet werden (Monat und Jahr), die validiert werden.

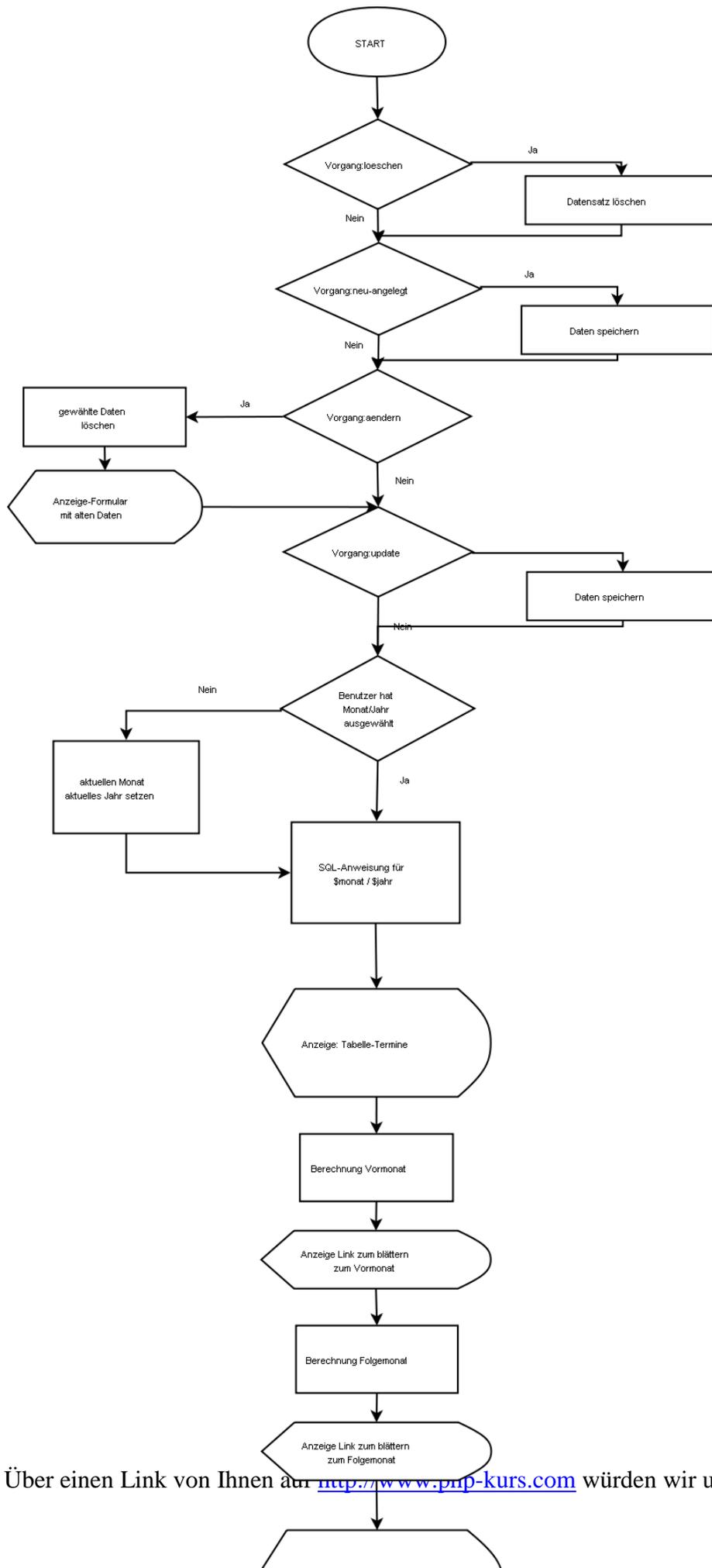
Das Terminkalender-Script an einem Stück, damit die obigen Sprünge nachvollzogen werden können, bekommen Sie unter:

<http://www.iwke.de/kunden/php-kurs/script-php-terminkalender-mit-mysql.htm>

und ab der übernächsten Seite.

Zur Verdeutlichung ein Ablaufdiagramm:

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen



Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Script PHP-Terminkalender

Alle Erklärungen finden Sie zum Terminkalender mit MySQL und PHP Schritt für Schritt im vorherigen Kapitel.

Das Konfigurationsskript für den Terminkalender:

```
<?php
define ( 'MYSQL_HOST', 'localhost' );
define ( 'MYSQL_BENUTZER', 'root' );
define ( 'MYSQL_KENNWORT', '' );
define ( 'MYSQL_DATENBANK', 'terminkalender' );

$db_link = @mysql_connect (MYSQL_HOST, MYSQL_BENUTZER, MYSQL_KENNWORT);
if ( ! $db_link )
{
    // hier sollte dann später dem Programmierer eine
    // E-Mail mit dem Problem zukommen gelassen werden
    // die Fehlermeldung für den Programmierer sollte
    // das Problem ausgeben mit: mysql_error()
    die('keine Verbindung zur Zeit möglich - später probieren ');
}
?>
```

Das Hauptskript, in dem Termine angezeigt, angelegt, geändert und gelöscht werden können. Die Anzeige im Terminkalender erfolgt monatsweise.

```
<?php

require ('termin-konfiguration.php');

$db_sel = mysql_select_db( MYSQL_DATENBANK )
    or die("Auswahl der Datenbank fehlgeschlagen");

// Speichern neuer Daten
if ( $_POST['vorgang'] == 'neu' )
{
    speichere_daten ( $_POST['termin'] );
}

// Löschen von Einträgen
if ( $_GET['vorgang'] == 'loeschen' )
{
    loeschen_daten ( $_GET['id'] );
}

// Daten zum Ändern anzeigen
if ( $_GET['vorgang'] == 'aendern' AND $_POST['vorgang'] <> 'update' )
{
    anzeige_daten_zum_bearbeiten ( $_GET['id'] );
}

if ( $_POST['vorgang'] == 'update' )
{
    update_daten ( $_POST['termin'] );
}
```

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

```

// wenn ein Monat angegeben ist, wird die Echtheit der Daten überprüft
if ( $_GET['monat'] )
{
    if ( (INT) $_GET['monat'] > 0 AND (INT) $_GET['monat'] < 13 )
    {
        $monat = (INT) $_GET['monat'];
    }
    else
    {
        // ist kein Monat angegeben, wird der aktuelle Monat verwendet
        $monat = date("n");
    }
}
else
{
    // ist kein Monat angegeben, wird der aktuelle Monat verwendet
    $monat = date("n");
}

if ( $_GET['jahr'] )
{
    if ( (INT) $_GET['jahr'] > 0 AND (INT) $_GET['jahr'] < 3000 )
    {
        $jahr = (INT) $_GET['jahr'];
    }
    else
    {
        $jahr = date("Y");
    }
}
else
{
    $jahr = date("Y");
}

// Ausgabe vom Monat und dem Jahr
echo "<h1>Terminkalender $monat.$jahr</h1>";

$sql = "
SELECT
    id, datum, titel, beschreibung
FROM termine
WHERE
    YEAR(datum) = '$jahr'
AND
    MONTH(datum) = '$monat'
ORDER BY datum
";

$db_erg = mysql_query( $sql );
if ( ! $db_erg )
{
    die('Ungültige Abfrage: ' . mysql_error());
}

$anzahl_eintraege = mysql_num_rows($db_erg);
echo "<p>Anzahl der Einträge: $anzahl_eintraege</p>";

// Ausgabe Tabelle mit Terminen
echo '<table border="1">';
while ($zeile = mysql_fetch_array( $db_erg, MYSQL_ASSOC))

```

```

{
    echo '<tr>';
    echo '<td>'. $zeile['datum'] . '</td>';
    echo '<td>';
    echo '<b>'. $zeile['titel'] . '</b><br />';
    echo $zeile['beschreibung'];
    echo '</td>';

    echo '<td>';
    echo '<a href="terminanzeige.php?vorgang=loeschen&id=';
    echo $zeile['id'];
    echo '">l&ouml;schen</a> ';
    echo '</td>';

    echo '<td>';
    echo '<a href="terminanzeige.php?vorgang=aendern&id=';
    echo $zeile['id'];
    echo '">&auml;ndern</a> ';

    echo '</td>';

    echo '</tr>';
}
echo '</table>';

mysql_free_result( $db_erg );

// Vormonat Kontrolle, ob bereits Januar
if ( $monat == 1 )
{
    $vmonat = 12;
    $vjahr = bcsub ( $jahr, 1 );
}
else
{
    $vmonat = bcsub ( $monat, 1 );
    $vjahr = $jahr;
}

echo '<a href="terminanzeige.php';
echo '?monat=' . $vmonat;
echo '&jahr=' . $vjahr;
echo '">Vormonat</a>';

echo ' | ';

echo '<a href="terminanzeige.php';
echo '?monat=' . date("n");
echo '&jahr=' . date("Y");
echo '">akt. Monat</a>';

echo ' | ';

// nächster Monat - Kontrolle, ob bereits Dezember
if ( $monat == 12 )
{
    $nmonat = 1;
    $njahr = bcadd ( $jahr, 1 );
}
else
{
    $nmonat = bcadd ( $monat, 1 );
}

```

```

    $njahr = $jahr;
}
echo '<a href="terminanzeige.php';
echo '?monat=' . $nmonat;
echo '&jahr=' . $njahr;
echo '">n&auml;chster Monat</a>';

// Anzeige Formular
echo '<hr />';
echo '<form name="" action="" method="POST" enctype="text/html">';

echo '<p>Datum in der Form JJJJ-MM-TT<br />';
echo '<input type="text" name="termin[datum]" value="" size="10" maxlength="10" />';
echo '</p>';

echo '<p>Kurzbeschreibung<br />';
echo '<input type="text" name="termin[kurzbeschreibung]" value="" size="50" maxlength="255" />';
echo '</p>';

echo '<p>ausf&uuml;hrliche Beschreibung<br />';
echo '<textarea name="termin[beschreibung]" rows="10" cols="80"></textarea></p>';

echo '<input type="hidden" name="vorgang" value="neu" />';

echo '<input type="Submit" name="" value="speichern" />';
echo '</form>';

// Daten speichern
function speichere_daten ( $termin )
{
    // hier sollte noch eine Validierung des Datums stattfinden

    $sql = "INSERT INTO termine
    (
        `id`, `datum`, `titel`, `beschreibung`
    )
    VALUES
    (
        NULL ,
        '". $termin['datum'] ."',
        '". $termin['kurzbeschreibung'] ."',
        '". $termin['beschreibung'] ."'
    )
    ";

    $db_erg = mysql_query( $sql );
    if ( ! $db_erg )
    {
        die("Ungültige Abfrage: $sql <hr />" . mysql_error());
    }
    else
    {
        echo "<h2>Termin gespeichert</h2>";
    }
}

function loeschen_daten ( $id ) {

```

```

$sql = "DELETE FROM termine
        WHERE id='$id'
        LIMIT 1
        ";

$db_erg = mysql_query( $sql );
if ( ! $db_erg )
{
    die("Ungültige Abfrage: $sql <hr />" . mysql_error());
}
else
{
    echo "<h2>Termin gelöscht</h2>";
}
}

function anzeige_daten_zum_bearbeiten ($id)
{
    $sql = "
    SELECT *
    FROM termine
    WHERE id = '$id'
    ";

    $db_erg = mysql_query( $sql );
    if ( ! $db_erg )
    {
        die('Ungültige Abfrage: ' . mysql_error());
    }

    $zeile = mysql_fetch_array( $db_erg, MYSQL_ASSOC);

    echo '<form name="" action="" method="POST" enctype="text/html">';

    echo '<p>Datum in der Form JJJJ-MM-TT<br />';
    echo '<input type="text" name="termin[datum]" value=""';
    echo $zeile['datum'];
    echo '" size="10" maxlength="10" />';
    echo '</p>';

    echo '<p>Kurzbeschreibung<br />';
    echo '<input type="text" name="termin[kurzbeschreibung]" value=""';
    echo $zeile['titel'];
    echo '" size="50" maxlength="255" />';
    echo '</p>';

    echo '<p>ausführliche Beschreibung<br />';
    echo '<textarea name="termin[beschreibung]" rows="10" cols="80">';
    echo $zeile['beschreibung'];
    echo '</textarea></p>';

    echo '<input type="hidden" name="termin[id]" value=""';
    echo $zeile['id'];
    echo '" />';
    echo '<input type="hidden" name="vorgang" value="update" />';

    echo '<input type="Submit" name="" value="Änderungen speichern" />';
    echo '</form>';
    echo '<hr />';
}

```

```

// Daten updaten
function update_daten ( $termin )
{
    // hier sollte noch eine Validierung des Datums stattfinden
    $sql = "UPDATE termine SET
        `datum` = '". $termin['datum'] ."',
        `titel` = '". $termin['kurzbeschreibung'] ."',
        `beschreibung` = '". $termin['beschreibung'] ."'
    WHERE
        id = '". $termin['id'] ."'
    ";

    $db_erg = mysql_query( $sql );
    if ( ! $db_erg )
    {
        die("Ungültige Abfrage: $sql <hr />" . mysql_error());
    }
    else
    {
        echo "<h2>Termin geupdatet</h2>";
    }
}

?>

```

MySQL sauber schreiben

MySQL Schlüsselwörter in Großbuchstaben

Alle Schlüsselwörter in MySQL sollten komplett groß geschrieben werden wie "SELECT, FROM, WHERE, ALTER, UPDATE, INSERT".

Dadurch ist auf einen Blick ersichtlich, was Inhalt wie Spaltennamen, Tabellennamen und Bedingungen ist und was MySQL-Schlüsselwörter.

Fehlervermeidung bzw. Suchhilfe bei MySQL-Fehlern

Klar, man kann alles in eine Zeile schreiben, aber spätestens wenn der erste Fehler kommt, sucht man, wo das Problem in der einen Zeile sitzt. Sinnvoll ist es daher, mit Umbrüchen und Einrückungen zu arbeiten (was man auch bei PHP und HTML macht).

Als Beispiel eine einfach Anweisung mit INSERT

```
INSERT INTO termine
(
    datum,
    uhrzeit,
    beschreibung
)
VALUES
(
    '2007-12-24',
    '15:23:00',
    'Weihnachtsgeschenke kaufen'
);
```

Vorteil ist, dass man bei einem Fehler die fehlerhafte Zeile als Nummer erhält. Weiterhin kann man einfach abschätzen, ob es so viele Werte wie Spalten gibt.

Jedem MySQL-Query sein Semikolon

Zur Fehlervermeidung immer das jeweilige MySQL-Kommando mit einem Semikolon (;) beenden. Es können so mehrere MySQL-Kommandos hintereinander gesendet werden und die Datenbank weiss dann, wo der erste aufhört und der folgende anfängt.

Teilweise sieht man die MySQL-Befehle ohne Semikolon zum Abschluss. Das funktioniert zwar in der Regel, da nur 1 Befehl gesendet wird. Aber sobald mal 2 verknüpft werden, kommen Probleme. Und nachdem man es von PHP schon gewohnt ist, einfach in MySQL genauso machen.

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

DON'Ts: SELECT *

Schön einfach ist es natürlich, einfach alle Spalten zu selektieren, selbst wenn man nicht alle benötigt. Um die Anwendung schnell zu machen, sollten Sie bestenfalls zum Testen mit "SELECT *" operieren - für die spätere Anwendung immer die benötigten Spalten aufführen. Das gibt **schnelle und performante Anwendungen**.

PHP und MySQL-Querys

Erstellen Sie für Ihre MySQL-Query eine Variable. Hier kann dann sauber das MySQL-Kommando ausgebaut werden und Sie haben Chancen auf eine hilfreiche Fehlermeldung, falls sich doch mal ein Fehler einschleicht.

```
$sql = "INSERT INTO termine
      (
        datum,
        uhrzeit,
        beschreibung
      )
VALUES
      (
        '2007-12-24',
        '15:23:00',
        'Weihnachtsgeschenke kaufen'
      );
";

$result = mysql_query($sql) OR die("". $sql. " : ".mysql_error());
```

Zusätzlich bekommt man dann in der Fehlermeldung die ursprüngliche MySQL-Query angezeigt.

MySQL Datenbank sichern

Ab einer gewissen Größe der MySQL Datenbank kann nicht mehr so einfach gesichert werden. Die Laufzeiten von PHP-Programm reichen nicht mehr aus, um die Sicherung komplett zu machen. Daher gibt es dafür geschickte Lösungen:

- www.mysql-dumper.de - sehr gutes Programm aus Deutschland (28876 Oyten)
- www.phpmybackuppro.net

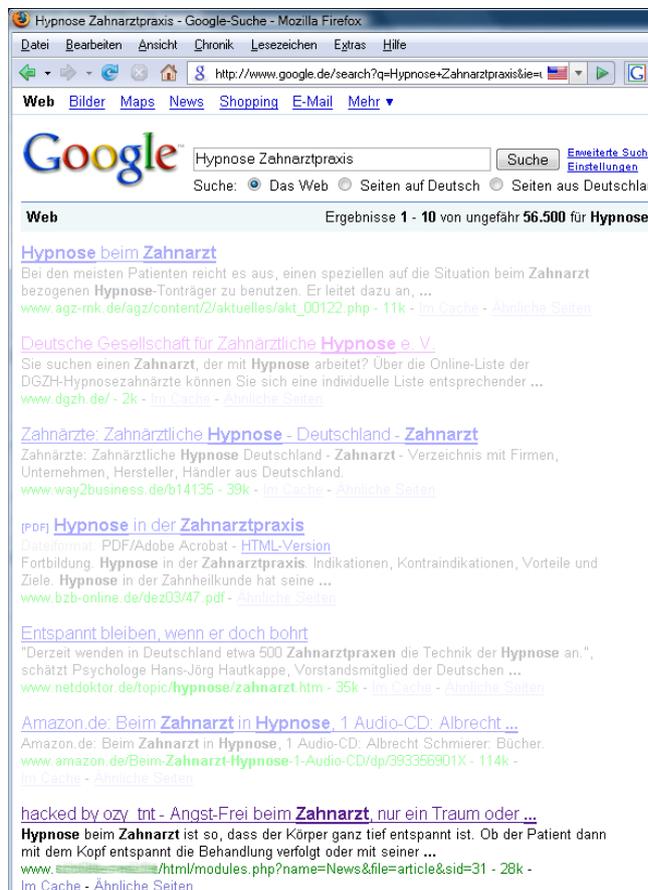
PHP und Sicherheit

So schnell sich mit PHP Erfolge beim Erstellen von Anwendungen erreichen lassen, genauso schnell lassen sich auch Sicherheitsprobleme für den Server versehentlich „programmieren“. Daher ist diese Kapitel über PHP und Sicherheit jedem dringlichst ans Herz gelegt!

Es wird hier kein Anspruch auf Vollständigkeit erhoben noch werden hier irgendwelche rechtlich bindenden Aussagen getroffen (ich möchte für nichts haften, was andere verbocken). Daher dienen die Kapitel über PHP und Sicherheit rein zur Lehre und zum Selbststudium – da für den effektiven Umgang mit Sicherheit Kenntnisse notwendig sind, wie Hacker vorgehen, wird hierauf soweit wie erforderlich eingegangen. Es wird hier niemals ein vollständiger Code geliefert, wie man hackt! Wenn jemand in der Hoffnung diese Kapitel liest, um zu lernen, wie man Server hackt, ist er hier sowohl inhaltlich falsch, als auch von der Philosophie her. Man sollte nach meinem Verständnis seine Zeit dazu nutzen, etwas Sinnvolles aufzubauen, anstatt sinnlos etwas zu zerstören.

Beispiel fürs Hacken

Warum vernachlässigte Sicherheit einen sehr teuer zu stehen kommen kann, hier an einem kleinen Beispiel. Am 11.4.2008 war die Seite für „Angst-Frei beim Zahnarzt“ noch auf Seite 1 bei Google. Zu diesem Zeitpunkt hat wohl ein Hacker bereits zugeschlagen (wobei dies eher ein Script-Kiddy war).



Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Sie haben das kostenlose eBook hier vor sich – deshalb fehlt dieses Kapitel – Sie können selber den Preis bestimmen, was das komplette Buch kosten soll.

Mehr unter:

<http://www.php-kurs.com/php-ebook.php>

Tipps und Tricks

Hier finden Sie eine lockere Sammlung von Tipps und Tricks zu PHP und MySQL, die laufend ergänzt wird.

- [Datumsausgabe in Deutsch](#)
- [MySQL mit deutschem Datumsformat](#)
- [MySQL Groß- und Kleinschreibung berücksichtigen](#)

Alle Tipps und Tricks sind nun in der Steuerung zu finden.

Tabellenreihen abwechseln einfärben

Nachdem Sie eine Tabelle erstellt haben, können Sie die `<tr>` abwechselnd einfärben. Dazu gibt es verschiedene Möglichkeiten. Z.B. über `<tr >` oder mit CSS und `<tr class="hg1" >` bzw. `<tr class="hg2" >`

Innerhalb der Tabelle können Sie eine Variable hochzählen lassen (diese muss nicht unbedingt angezeigt werden). Anhand dieser Zahl können Sie bestimmen, ob diese gerade oder ungerade ist und dann entsprechend die Zeile einfärben lassen.

Dazu bedienen wir uns der Mathematik und dem Modulo von 2 verwendet. Der Modulo ist eine mathematische Funktion, die als Ergebnis den Rest aus der Division zweier ganzer Zahlen angibt. Somit erhalten wir bei der Division durch 2 entweder 0 (somit haben wir eine gerade Zahl) oder nicht 0, somit haben wir eine ungerade Zahl.

```
if ( bcmmod ( $zeilenr , '2' ) == 0 )
{
    echo '<tr bgcolor="#ACC8F0" >';
}
else
{
    echo '<tr bgcolor="#DDE8F9" >';
}
```

Tabellenreihen einfärben per CSS

```
if ( bcmmod ( $zeilenr , '2' ) == 0 )
{
    echo '<tr class="hg1" >';
}
else
{
    echo '<tr class="hg2" >';
}
```

Zusätzlich muss nun noch in der CSS-Datei die entsprechenden Klassen für hg1 und hg2 erstellt werden:

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

```
.hg1 { background-color:#ACC8F0; }
.hg2 { background-color:#DDE8F9; }
```

Debuggen mit Zeilenausgabe

Oft erfolgt im Stadium des Entwickelns eine Kontrollausgabe von Variablen-Inhalten mit echo und der entsprechenden Variabln. Dass dies nicht unbedingt der beste Stil ist, bleibt hier dahingestellt.

Als kleine Debugging-Hilfe hier eine Funktion, die zusätzlich die Zeilennummer und den Dateinamen ausgibt. Dies geschieht über `__LINE__` und `__FILE__`

Der Aufruf der Funktion geschieht über den Funktionsnamen `decho` (hier d für debug und echo = decho - als Übergabe wird der Variablenname erwartet, nicht die Variable. Daher im Beispiel für die Variable `$name` dann `decho ("name", __LINE__, __FILE__);`. Es kann genauso der Inhalt von Arrays ausgegeben werden.

```
$name = "www.php-kurs.com";

decho ("name", __LINE__, __FILE__ );

function decho ( $variable, $zeilennummer = "", $dateiname = "" )
{
    global $$variable;
    echo '<blockquote style="border:1px dotted red; border-left:10px solid red; padding-left:1em;">';
    echo "Kontrollausgabe: ";
    echo "<pre><b>";
    if ( $$variable != "" )
    {
        echo '$'. $variable .' = ';
        print_r ( $$variable );
    }
    else
    {
        print_r ( $variable );
    }
    echo "</b></pre>";
    if ( $zeilennummer OR $dateiname )
    {
        echo "<br /> <small>(Zeile: $zeilennummer - $dateiname)</small>";
    }
    echo "</blockquote>";
}
```

Zur Erklärung: Der Aufruf erfolgt über `decho ("name", __LINE__, __FILE__);` - dabei wird nicht die Variable übergeben, sondern nur der Name der Variable und innerhalb der Funktion über `global $$variable` verfügbar gemacht. Das doppelte `$`-Zeichen ist korrekt :)

Wird der Funktion nicht der Variablenname übergeben, sondern die Variable, wird auch der Inhalt ausgegeben, allerdings ohne Namen der Variable.

Eleganter ist es, diese Funktion auszulagern und per `include_once` dann einzubinden.

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Übergebene Variablen bereinigen

Vom Besucher eingegebene Daten sind prinzipiell nicht zu trauen! Daher sollten alle Daten überprüft und bereinigt werden.

Um dies einfach zu ermöglichen, gibt es in PHP die praktische Funktion `array_walk` und `array_walk_recursive`.

Wir können über `strip_tags` alle HTML und PHP-Tags entfernen lassen (mit Einschränkungen). Allerdings kann es bei der Art vorkommen, dass weiterer Text gelöscht wird, wenn der übergebene Text fehlerhaft war.

Im einem Schritt wandeln wir Sonderzeichen in HTML-Codes um. Aus `<` (kleiner als) wird dann `<`

Im letzten Schritt werden die Leerzeichen am Anfang und Ende entfernt.

```
array_walk ( $_POST,      'eingabebereinigen' );
array_walk ( $_GET,      'eingabebereinigen' );
array_walk ( $_REQUEST,  'eingabebereinigen' );

function eingabebereinigen(&$value, $key)
{
    // keine HTML-Tags erlaubt, außer p und br
    // $value = strip_tags($value, '<p><br>');

    // HTML-Tags maskieren
    $value = htmlspecialchars($value, ENT_QUOTES);

    // Leerzeichen am Anfang und Ende beseitigen
    $value = trim($value);
}
```

PHP Fehlermeldungen

Notice: Undefined index:

Diese "Fehlermeldung", die hier nur eine Anmerkung ist (notice) erhält man, wenn in einem PHP-Programm bisher diese Variable noch nicht definiert wurde.

Notizen erhält man nur, wenn PHP so eingestellt ist, dass nicht nur die heftigen Fehler sondern auch alle Anmerkungen über Unregelmäßigkeiten angezeigt werden sollen. Das erreicht man entweder in der PHP.INI oder wenn am Anfang des PHP-Scripts die angegeben wurde über:

```
// Damit alle Fehler angezeigt werden
error_reporting(E_ALL);
```

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Dies kann schnell der Fall sein, wenn man für ein Formular abfragt, ob ein Variable gesetzt wurde.

```
<?php
error_reporting(E_ALL);

if ( $_POST['aktion'] == "speichern" )
{
    // und hier nun die Aktion
}
?>
```

Um diesen "Fehler" in den Griff zu bekommen, da die Variable ja aus dem Formular kommt und daher noch nicht vom PHP-Programm gesetzt werden wird, kann erst auf die Existenz der Variable getestet werden und dann die Inhaltsprüfung.

```
<?php
error_reporting(E_ALL);

if ( isset($_POST['aktion']) AND $_POST['aktion']=="speichern" )
{
    // und hier nun die Aktion
    echo "nun speichern";
}
?>

<form name="" action="" method="POST" enctype="text/html">

<input type="text" name="aktion" value="" size="20" maxlength="50" />

<input type="Submit" name="" value="Save" />

</form>
```

error_reporting einsetzen

```
// einfache Laufzeitfehler melden
error_reporting(E_ERROR | E_WARNING | E_PARSE);

// Benachrichtigungen von E_NOTICE können sinnvoll sein
// bei nicht initialisierte oder falsch geschriebene Variablen
error_reporting(E_ERROR | E_WARNING | E_PARSE | E_NOTICE);

// alle Fehler mit Ausnahmen von E_NOTICE anzeigen
// Dies ist i.d.R. die Standardeinstellung in php.ini
error_reporting(E_ALL ^ E_NOTICE);

// alle PHP-Fehler melden
error_reporting(E_ALL);

// Fehlermeldungen ganz abschalten
error_reporting(0);
```

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Datumsausgabe auf Deutsch in PHP

Die Datumsausgabe von Wochentagen und Monaten ist standardmäßig in Englisch. Dies kann aber umgestellt werden, sofern auf dem Server die entsprechenden Sprachpakete installiert sind.

Dazu wird über die Funktion `setlocale` auf Deutsch umgestellt und für die Ausgabe dann `strftime` verwendet.

```
// Beispielausgabe vor Umstellen auf Deutsch
echo strftime("%A, der %e. %B %G", strtotime("now"));
echo "<p>Es erfolgt nun die Umstellung auf dt. Datum:";
setlocale (LC_ALL, 'de_DE@euro', 'de_DE', 'de', 'ge');
echo strftime("%A, der %e. %B %G", strtotime("now"));
```

Es erfolgt nun die Ausgabe vom ausgeschriebene Wochentag, vom Tag als Zahl, vom ausgeschriebenen Monat und dem Jahr – nach der Kalenderdatumsumstellung auf Deutsch sollte das Ganze in Deutsch dastehen.

Die Angabe bei `strftime` ist wiederholend. PHP sucht sich die entsprechende installierte Sprachversion heraus, die je nach Linux-System unterschiedlich benannt sein kann, z. B. `ge`, `de`, `de_DE` oder `de_DE@euro`.

Sollte es nicht funktionieren, kann auf dem Server das entsprechende Sprachpaket nicht installiert sein – dann heißt die Devise: Sprachpaket nachinstallieren.

MySQL mit deutschem Datumsformat

Datumsfelder in MySQL werden standardmäßig im internationalen Format "YYYY-MM-DD", also "2007-07-27" ausgegeben. Dies ist zwar zum Sortieren enorm praktisch, aber zum Lesen ungewohnt.

Es gibt die Möglichkeit, dass die MySQL-Abfrage bereits ein gewünschtes deutsches Format zurückliefert. Ergänzen Sie Ihre MySQL-Abfrage einfach:

Als Beispiel:

```
// ohne deutsches Datum
$sql = "SELECT datum FROM tabelle;";
// mit deutschem Datum
$sql = "SELECT DATE_FORMAT (datum, „%e.%m.%y“) AS datum FROM tabelle;";
```

Formate für DATE_FORMAT unter MySQL

Das Format übergeben Sie über die %-Zeichen.

- %d – Tag mit führender Null
- %e – Tag ohne führende Null
- %m – Monat in Zahlen

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

%M – Monat ausgeschrieben
%y – Jahr 2 stellig
%Y – Jahr vierstellig

Achten Sie auf das Setzen des Alias (AS datum) – ansonsten ist das Datumsfeld nicht mehr ansprechbar und somit wäre es wertlos. Über den Alias können Sie es wie gewohnt mit "datum" ansprechen.

Die Funktion ist ab MySQL Version 5 verfügbar – weitere Informationen unter:
dev.mysql.com/doc/refman/5.0/en/date-and-time-functions.html

MySQL Groß- und Kleinschreibung berücksichtigen

Eine MySQL-Abfrage für Benutzernamen und Passwort sollte auf jeden Fall auf Groß- und Kleinschreibung achten. Wird eine normale Anfrage gefahren mit:

```
$sql = "SELECT nutzer
FROM zugangsdaten
WHERE nutzer = '". $_POST['nutzer'] . "'
AND password = '". $_POST['password'] . "'; ";
```

so wird keine Rücksicht auf Groß- und Kleinschreibung genommen. Der Nutzer kann sich einloggen mit "Moritz" oder "moritz" oder "mORitZ" – es ist vollkommen egal.

Doppelter Schutz – auch vor SQL-Injection-Angriffe

Wird von den Daten mit der Funktion `md5()` eine Prüfsumme gebildet und diese verglichen, ist Groß- und Kleinschreibung wichtig und SQL-Injection-Angriffe laufen ins Leere!

```
$sql = "SELECT nutzer
FROM zugangsdaten
WHERE nutzer = '". $_POST['nutzer'] . "'
AND password = '". $_POST['password'] . "'; ";
```

Sie erhöhen die Sicherheit durch Brute-Force-Hackens um den Faktor von ca. 100 bei einem 8-stelligen Passwort.

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

automatischer Start nach Ablauf

Manchmal ist es geschickt, wenn ein PHP-Programm sich immer wieder selber startet, bis ein bestimmter Zustand eingetreten ist. Dazu wird oft der header-Befehl hergenommen. Will man aber Ausgaben haben, hilf auch JavaScript. Das ganze ist primär für daheim geschickt, um z.B. Datensätze einzuspielen (oder auszulesen), deren Anzahl die Laufzeit des PHP-Scripts sprengen würde. Also lässt man es so oft automatisch starten, bis die Aufgabe erledigt ist :)

```
// hier ist das PHP-Programm
if ( $noch_ablaufen_lassen <> true )
{
    exit;
}
?>

<script language="JavaScript">
<!--
var ziel = "<?php echo $_SERVER['PHP_SELF']; ?>";
window.location.href = ziel;
//-->
</script>
```

Links

Interessante Websites zum Thema PHP

www.selfphp.de mit Befehlsreferenz, PHP Kochbuch, PHP 5 Praxisbuch und PHP Code Snippets. Sehr Umfangreich!

www.php.net/manual/de/ Das offizielle PHP Manual in deutscher Sprache. Kann auch kostenlos offline gelesen werden. Es steht in den Formaten .html, .PDF und als Windows Hilfe Datei zum Download zur Verfügung

PHP und Ajax

PHP mit Ajax aufmotzen - mit diesem Toolkit gehts flott: <http://www.xajaxproject.org/>

PHP-Programme unter Windows lauffähig

WinBinder ist eine Open Source Erweiterung für PHP. Dadurch können native Windows-Anwendungen mit PHP erstellt werden.

<http://winbinder.org/>

Frameworks

- [Yellow Duck Framework](#)
- [Qcodo - PHP Development Framework](#)
- [Prado Component Framework for PHP 5](#)

Bücher und Literaturhinweise

- [Besser PHP programmieren. Handbuch professioneller PHP-Techniken](#)
- [PHP 5 & MySQL 5 Kochbuch. Rezepte, Lösungen, Best Practices \(Broschiert\)](#)
- [PHP 5 Kochbuch \(Taschenbuch\)](#)
- [LAMP für Dummies. Linux + Apache + MySQL + PHP = LAMP](#)

Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen

Entweder Sie haben den PHP-Kurs ...

... komplett durchgearbeitet, oder Sie haben am Ende zu lesen angefangen :). Auf jeden Fall sind Sie auf dieser Seite gelandet und das deutet darauf hin, dass Ihnen der Kurs gefallen hat und er für Sie nützlich ist.

Daher können Sie (wenn Sie wollen) mithelfen, den Kurs bekannter zu machen. Unterstützen Sie uns, indem Sie einen Link zu uns auf Ihrer Homepage einbinden.

```
<a title="PHP Kurs und MySQL Tutorial mit eBook"
target="_blank" href="http://www.php-kurs.com">www.PHP-Kurs.com</a>
```

Natürlich freuen wir uns auch über jeden (Social)-Bookmark ...



Über einen Link von Ihnen auf <http://www.php-kurs.com> würden wir uns freuen