

# PHP+MySQL

*Web-Site-  
Programmierung 0.9*

# PHP

## Elementares

### Konfigurationsdatei für PHP:

/etc/php.ini Nach Bearb: Webserver neu starten

### Script-Kennzeichnung:

```
<?php ... ?>
<script language="php"> ... </script>
<% .... %> Falls "asp_tags = on" in /etc/php.ini
```

### Kommentare:

```
// Einzeiliger Kommentar (C-Stil)
# Einzeiliger Kommentar (Shell-Stil)
/* .... */ Mehrzeiliger Kommentar
```

## Ausgabenweisungen

```
echo "..."; Ausgabe mit Variableninhalten
echo '...'; Ausgabe ohne Variableninhalten
echo ein text, " noch", ' etwas'; mehrere Parameter möglich
print ("Beispiel"); print liefert Return-Code
print ( 'Beispiel' ); Entwertungen mit: ""
```

### Längere Texte mit Here-Scripten (z.B. Html-Code ausgeben):

```
echo <<< ENDE
dies wird ein langer Text mit Variablen und "Anführungsstext"
```

ENDE;

(Vorsicht: "ENDE;" in eigener Zeile ohne Leerzeichen danach!)

### Formatierte Ausgabe:

```
printf("Gehalt: %8.2f\n<br />", 2.456);
$var = sprintf(format, var1, ...); Ergebnis in String speichern
print_r ( variable); Variablen/Arrays komplett anzeigen
```

## Variablen

Ziffern, "\_" und Buchstaben für Variablen-Namen erlaubt

Doppelte Dereferenzierung möglich (Dynamische Variablen):

```
$Wert="Beispiel"; $Var="$Wert"
$$Var oder ${$Var} geben "Beispiel" aus!
```

Array-Variablen im String ausgeben mit {...}:

```
echo "Lieblingslehrer: ${_POST['Lehrer']}"
```

### Mit Here-Scripten:

```
$str = <<< ENDE
dies wird ein langer Text mit Variablen und "Anführungsstext"
```

ENDE;

**Konstanten:** Zugriff und Deklaration ohne "\$"

```
define('MWST','0.16'); $Brutto = $Netto + $Netto * MWST;
```

### Konvertierungsfunktionen:

```
decbin(5) = 101;
hexdec('FF') = 255; etc.
```

## Zeichenketten

```
. (Punkt) Zusammenfügen von Strings
$str .= "Ein Text" Anfügen von Text am String
```

### Stringfunktionen, Beispiele:

```
strlen($str);
$domain = strstr($email, '@');
$pos = strpos($mystring, $findme);
$rest = substr("abcdef", 0, 4); // zeigt "abcd"
```

## Datentypen

```
integer 12; -36
double 23.345; 34.345e-3
string "Ein Text", 'Ein Text'
array array("Eins", 2, 'Drei')
object
Datentypkonvertierung: $VAR = (integer) 3.4 + (integer) 1.2;
```

## Operatoren

Standardoperatoren: ..., %, ++, --, (...)  
 Vergleiche: ==, !=, <=, >= Numerische + String-Vergleiche  
 === Gleich + gleicher Datentyp  
 logische Operatoren: !, and, &&, or, ||, xor  
 Bitoperatoren: ~, |, &, ^, >>, <<  
 Zeichenketten: . Verknüpfen von Zeichenketten  
 Zuweisungsoperatoren: =, %=, .=, += etc.

## Variablen testen

```
isset($Var) Variable vorhanden?
empty($Var) Variable leer oder Integer=0?
unset($Var) Variable löschen!
Weitere Funktionen: is_integer($Var), is_double($Var), is_string($Var),
is_array($Var), is_object($Var), is_file($Var), is_writable($Var)
```

## if

**if** (bedingung) Blöcke mit {...} klammern  
 erfolg();  
**elseif** (neuebedingung) neuanweisung;  
**else** misserfolg();

## switch

```
switch ( $Var)
{
    case wert: Anweisungsblock; break;
    default: Anweisungsblock; break;
}
```

## while

**while** (Bedingung) { Anweisungsblock; }  
**while** (Bedingung): Anweisungsblock; **endwhile**;  
**do** { Anweisungsblock; } **while** (Bedingung);

## foreach/for

**foreach** (\$array as \$index => \$wert)  
 echo "\$index : \$wert <br />";  
**for** (\$i = 0; \$i < 100; \$i++) { Anweisungsblock; }

## break, continue, exit

**break** [anzahl]; Anzahl Schleifen verlassen  
**continue** [anzahl]; Schleifendurchläufe überspringen  
**exit** [Zeichenkette] Programmabbruch [mit Nachricht]  
**die** ( Nachricht ); Funktion zum Programmabbruch

## Funktionen

```
function name (arg1, ..., argn) {
    anweisungen;
    return $wert;
}
```

### Variablen verändern:

```
$text="Anfangstext";
function aendern (&$var) {
    $var="neuer Wert";
}
aendern($text); echo $text;
```

### Vorgabewerte:

```
function machwas ($var1, $var2 = "default") {
    return ($var1 . $var2);
}
machwas("hallo", "du"); machwas("hallo");
```

### Globale Variablen:

```
$VAR = "Test";
function zugriff()
{ global $VAR;
  $VAR = "neuer Text";
}
```

### Temporäre Funktionen:

```
$fname = create_function('argumente', 'anweisungen');
$mwst = create_function('$wert', 'return ($wert * 1,16)');
```

## Felder

### Einfache Arrays:

```
$a[] = 1; $a[] = "zwei"; $a[] = 3;
oder: $a = array (1, "zwei", 3);
```

### Varianten zum Auslesen:

```
echo $a[0];
for ($i=0; $i<count($a); $i++ ) echo "a[" . $i . "] = " . $a[$i] . "<br />";
while (list($nr,$wert)=each($a) ) echo "a[" . $nr . "] = " . $wert . "<br />";
foreach ($a as $wert) echo $wert . "<br />";
```

### Assoziative Arrays:

```
$daten = array ("Name" => "Menne", "Vorname" => "Stefan");
```

### Varianten zum Auslesen:

```
echo $daten["Name"];
foreach ($a as $index=>$wert) echo "a[" . $index . "] = " . $wert . "<br>";
while ($var = each($daten) ) echo $var["key"] . " = " . $var["value"];
```

### Mehrdimensionale assoziative Arrays:

```
$Noten = array(
    "Menne" => array( "Mathe" => "2", "Inform" => "1", "BWL" => "2"),
    "Meier" => array( "Mathe" => "3", "Inform" => "4", "BWL" => "1"),
    "Meyer" => array( "Mathe" => "4", "Inform" => "4", "BWL" => "2")
);
// Anzeigen:
foreach ($Noten as $Name=>$Eintrag)
{ echo "<b>Schüler: $Name: </b><br>";
  foreach($Eintrag as $Fach=>$Note) echo "$Fach: $Note<br>";
}
```

### Funktionen auf Arrays:

```
count(feldname); sizeof(feldname);
```

```

reset(feld); prev(feld); pos(feld); current(feld); next(feld); end(feld);
in_array(wert, feld); true/false
array_search(wert, feld); liefert Index
array_unique(feld); Doppelte Werte entfernen
sort($feld), rsort($feld), array_multisort($feld);
array_diff(), array_intersect();
array_flip(); array_merge(); array_pad();

```

### Klassen

```

Klassen definieren:
class Taschenrechner
{ var $memory = 0; // Enthält das Ergebnis
  var $name; // Name des Taschenrechners für Ausgabe
  function __construct($name = "") # Konstruktor bei Def. der Instanz
  { $this->name = $name;
  }
  function add($wert = 0)
  { $this->memory += $wert;
  }
  function __destruct() # Destruktor wird bei Programmende aufgerufen
  { echo "Ende von : " . $this->name . " Wert: " . $this->memory;
  }
} // Ende Klasse Taschenrechner

```

```

$hans = new Taschenrechner("Hans");
$hans->add(2); $hans->add(4);
$susi = new Taschenrechner("Susi");
$susi->add(3); $hans->add(4); $susi->add(6);

```

### Formulardaten

```

Mehrfachauswahl verarbeiten:
<form method="POST">
  Hobbies: <select name="Hobbies[]" multiple>
    <option>Schwimmen</option>
    <option>Radfahren</option>
    <option>Tanzen</option>
    <option>Motorradfahren</option>
  </select><br>
  <input type="submit" value="Abschicken">
</form>
<?php
  $Hobbies=$_POST['Hobbies'];
  foreach ($Hobbies as $index=>$wert) echo "$index: $wert<br>";
?>

```

### Dateiverarbeitung

```

Datei in Array einlesen und verarbeiten
$zeilen = file("http://www.altavista.com/");
$zeilen = file ('/etc/passwd');
foreach ($zeilen as $nr => $zeile)
{ $eintrag = explode(':', $zeile);
  echo $eintrag[0].':'. $eintrag[4].<br />;
}
Datei in String speichern und aus String in Datei schreiben
$str = file_get_contents($datei);
$str = str_replace('\n','Menne', $str);
if (is_writable($datei)) file_put_contents($datei, $str);
Strukturierte Datei schreiben
$fp = fopen($passwd, 'w');
fprintf($fp, "%s,%s\n", $_POST['Loginname'], $_POST['Name']);
fclose($fp);
Zeilenweises schreiben:
$fp=fopen('test.txt', 'w');
fwrite($fp, "Dies ist ein magischer String\n in zwei Zeilen");
fclose($fp);

```

### Session-Daten anzeigen

```

@session_start();
foreach ($_SESSION as $key => $value) {
  echo $key. " = " . $value. "<br>";
}

```

### Datei-Upload

```

<form method="POST" enctype="multipart/form-data">
  <input type="hidden" name="MAX_FILE_SIZE" value="10240">
  <input type="file" name="Dateiname">
  <input type="submit" value="Abschicken">
</form>
<?php
$Hobbys=$_POST['Hobbys'];
if (isset($Dateiname)) {
  if (!is_uploaded_file($Dateiname))
  { echo "Fehler beim Upload. Evtl. fehlt Dateiname oder Datei ist zu
    groß (max. $MAX_FILE_SIZE)"; exit;}
  $uploadname = ".uploads/$Dateiname_name";
  move_uploaded_file($Dateiname, $uploadname);
  echo "Datei ".$uploadname." ist ".$Dateiname_size." groß und hat den
    Typ ".$Dateiname_type."<br>";
}

```

### Weiterleitung

```

<meta http-equiv="refresh" content="5; URL=topic.php?id=5">
Alle 10 Sek neu laden:
<meta http-equiv="pragma" content="no-cache">
<meta http-equiv="expires" content="0">
<meta http-equiv="refresh" content="10;">
Umlenkung auf andere Seite:
header("Location: http://www.ziel.de/zielseite.html");

```

### Datumsfunktionen

```

time() aktueller Datumsstempel (Sekunden seit 1.1.1970)
date('format', $zeit);
mktime

```

### Dünne Linien für Tabellen

```

<table style="border: #AAA 1px solid;">

```

### MySQL

```

$link = mysql_connect($db_server,$db_user,$db_passwort);
mysql_select_db($db_datenbank, $link);

```

### Tabelle erstellen

```

$sql="CREATE TABLE IF NOT EXISTS $dbtable (
  Nr int(6) NOT NULL auto_increment,
  Name char(30) default NULL,
  Vorname char(30) default NULL,
  LoginName char(30) default NULL,
  Passwort char(30) default NULL,
  Email char(50) default NULL,
  HomePage char(40) NOT NULL default 'http://'
  PRIMARY KEY (Name)
);";

```

### Datensätze einfügen

```

$sql = "INSERT INTO gaesteliste
  ( Name, Vorname, LoginName, Passwort, Email)
VALUES
  ('$Name', '$Vorname', '$LoginName', '$Passwort', '$Email')";

```

### Datensätze löschen

```

$sql = "DELETE FROM gaesteliste WHERE 'Nr' = 668 LIMIT 1 ;"

```

### Datensätze korrigieren

```

$sql = "UPDATE $dbtable SET
  Name = '$Name',
  Vorname = '$Vorname',
  Email = '$Email',
  HomePage = '$HomePage',
  WHERE Mnr = '$Session_Mnr' ";

```

### Datensätze anzeigen

```

$result = mysql_query('select * from '.$db_tabelle);
while ($zeile = mysql_fetch_assoc($result) )
{ echo $zeile['Name'];
}

```

# MySQL

## MySQL-b.i.b.

Unter ux-02 einloggen:  
 dabpasswd mysql  
 Es wird eine Datei "H:\dabpw\_mysql.sql" geschrieben. Sie enthält die Zugangsdaten zum MySQL-Server  
 Datenbanken/Tabellen erstellen und verwalten: Browser öffnen mit:  
<http://mysql.pb.bib.de/phpmyadmin>

## MySQL-Server verwalten

```
rcmysql start
/usr/share/mysql/mysqld.server start      Startscript
/usr/bin/mysqladmin -u root -p password 'new-password'
/usr/bin/mysqladmin -u root -h server -p password 'new-password'
ls -l /var/lib/mysql/mysql
  .frm      Formulardateien beschreiben die Tabellenstruktur
  .MYD      Eigentliche Daten
  .MYI      Indizes
mysqladmin shutdown      Server stoppen
/etc/my.cnf              Konfigurationsdatei
/usr/local/mysql/data/my.cnf
~/my.cnf                 Lokale Konfigurationsdatei
/var/lib/mysql           Data und Log-Dateien
/usr/bin/my*             MySQL-Programme
/usr/share/doc/mysql/manual.pdf
```

## MySQL-Befehle

```
CREATE DATABASE Datenbank
  Eine neue Datenbank anlegen
CREATE TABLE Tabelle
  Eine neue Tabelle anlegen
INSERT INTO Tabelle VALUES (w1, w2, ...)
  Daten eintragen
SELECT Ausdruck FROM Tabelle WHERE Bedingung
  Datensätze anzeigen
DELETE FROM Tabelle WHERE Bedingung
  Datensätze löschen
UPDATE Tabelle SET Feldname = Ausdruck WHERE Bedingung
  Datensätze ändern
DROP TABLE Tabelle
  Tabelle löschen
DROP DATABASE Datenbank
  Datenbank löschen
ALTER TABLE Tabelle Ausdruck
  Tabellenstruktur ändern
SHOW DATABASES
  Datenbanken anzeigen
USE Datenbank
  Datenbank zur Bearbeitung auswählen
SHOW TABLES
  Tabellen einer Datenbank anzeigen
SHOW COLUMNS FROM Tabelle
  Felder einer Tabelle anzeigen
FLUSH PRIVILEGES Benutzerprivilegien aktualisieren
LOAD DATA LOCAL INFILE "gaesteliste.txt" INTO TABLE gaesteliste;
EXIT
```

## Zugriff über die Shell

```
-> mysql -u test test
mysql> show databases;
mysql> show tables;
mysql> select * from tabelle;
mysql> exit
```

```
-> mysql -u test test << +++ENDE+++
#SHOW TABLES;
#DROP TABLE dozmen_namen;
CREATE TABLE dozmen_namen(name char(20), vname char(20));
INSERT INTO dozmen_namen (Name,vname) VALUES
('Menne','Stefan');
SHOW COLUMNS FROM dozmen_namen;
SELECT * FROM dozmen_namen;
+++ENDE+++
```

## PHP + MySQL

```
$link = mysql_connect ([ Server[:Port] [,User [,Password]] ]);
  Für dauerhafte Verbindung "mysql_pconnect" nehmen!
mysql_close($link);
die ("Jetzt ist Schluss");
  PHP beenden mit Fehlermeldung
mysql_select_db(datenbank [, link]);
  --> liefert TRUE/FALSE
$erg = mysql_query(sql [, link]);
  --> liefert TRUE/FALSE
```

```
$erg = mysql_db_query(datenbank, sql [, link]);
  --> liefert TRUE/FALSE
```

### Fehleranzeige:

```
mysql_query("SELECT FROM $test") or die ("FEHLER:
".mysql_error());
mysql_affected_rows ([, link]);
  Anzahl Datensätze, die per INSERT, DELETE, UPDATE geändert
  wurden.
mysql_num_rows ([, link]);
  Anzahl Datensätze, die per SELECT zurückgegeben wurden.
mysql_fetch_array (sql [, ERGTYP]);
  Anzahl Datensätze, die per SELECT zurückgegeben wurden.
  ERGTYP: MYSQL_ASSOC|MYSQL_NUM|MYSQL_BOTH
mysql_free_result($erg);
  Speicher freigeben
mysql_list_dbs();           Alle Datenbanken anzeigen
mysql_db_name();           Datenbankname ermitteln
mysql_create_db();         Datenbank anlegen
mysql_drop_db();           Datenbank löschen
mysql_list_tables();       Alle Tabellen anzeigen
mysql_tablename();         Tabellennamen ermitteln
```

```
update user set H_Pass=round(rand()*1000000);
```

Diese Referenz ist im Rahmen meiner Unterrichtstätigkeit am b.i.b. entstanden. Die Anordnung und Auswahl der Befehle ist aus meinen persönlichen Bedürfnissen für unterrichtliche Zwecke entstanden.

Verwenden Sie diese Referenz als Nachschlagewerk. Sie darf frei für Lehrveranstaltungen kopiert und verteilt werden, wenn mein Copyright lesbar bleibt.

Die Referenz wird ständig weiterentwickelt. Für Anregungen und Meinungen bin ich dankbar:

Stefan.Menne@gmx.de