

# MySQL und PHP – ein starkes Team zur Entwicklung dynamischer Websites

---

Dr. Thomas Meinike  
thomas.meinike@et.fh-merseburg.de

Fachhochschule Merseburg  
Fachbereich Elektrotechnik, Informationstechnik und Medien  
Studiengang „Kommunikation und Technische Dokumentation“



# Überblick

- ⇒ Statische und dynamische Webanwendungen
- ⇒ Die Datenbank MySQL
- ⇒ Die Scriptsprache PHP
- ⇒ Dynamische Webanwendung: ARDS-Projekt
- ⇒ Nützliche Web-Ressourcen
- ⇒ Zusammenfassung und Ausblick



# Statische und dynamische Webanwendungen 1/2

## ⇒ Statische Webanwendungen

Fertige HTML-Dokumente werden auf Webservern abgelegt und bei ihrer Anforderung direkt an den Client (Browser) ausgeliefert.

Zukünftig wird sich die Informationsflut kaum noch auf diese Weise bewältigen lassen:

- hoher Aufwand beim Erstellen und Publizieren
- unzureichende Aktualität von Informationen
- Speicherung von Daten erfordert Server-seitige Lösungen. ←



# Statische und dynamische Webanwendungen 2/2

## ⇒ Dynamische Webanwendungen

Auf der Basis von HTML-Vorlagen (Templates) werden mittels Server-seitigen Programmen (Scripts) in Verbindung mit Datenbankzugriffen Dokumente „on-the-fly“ generiert.

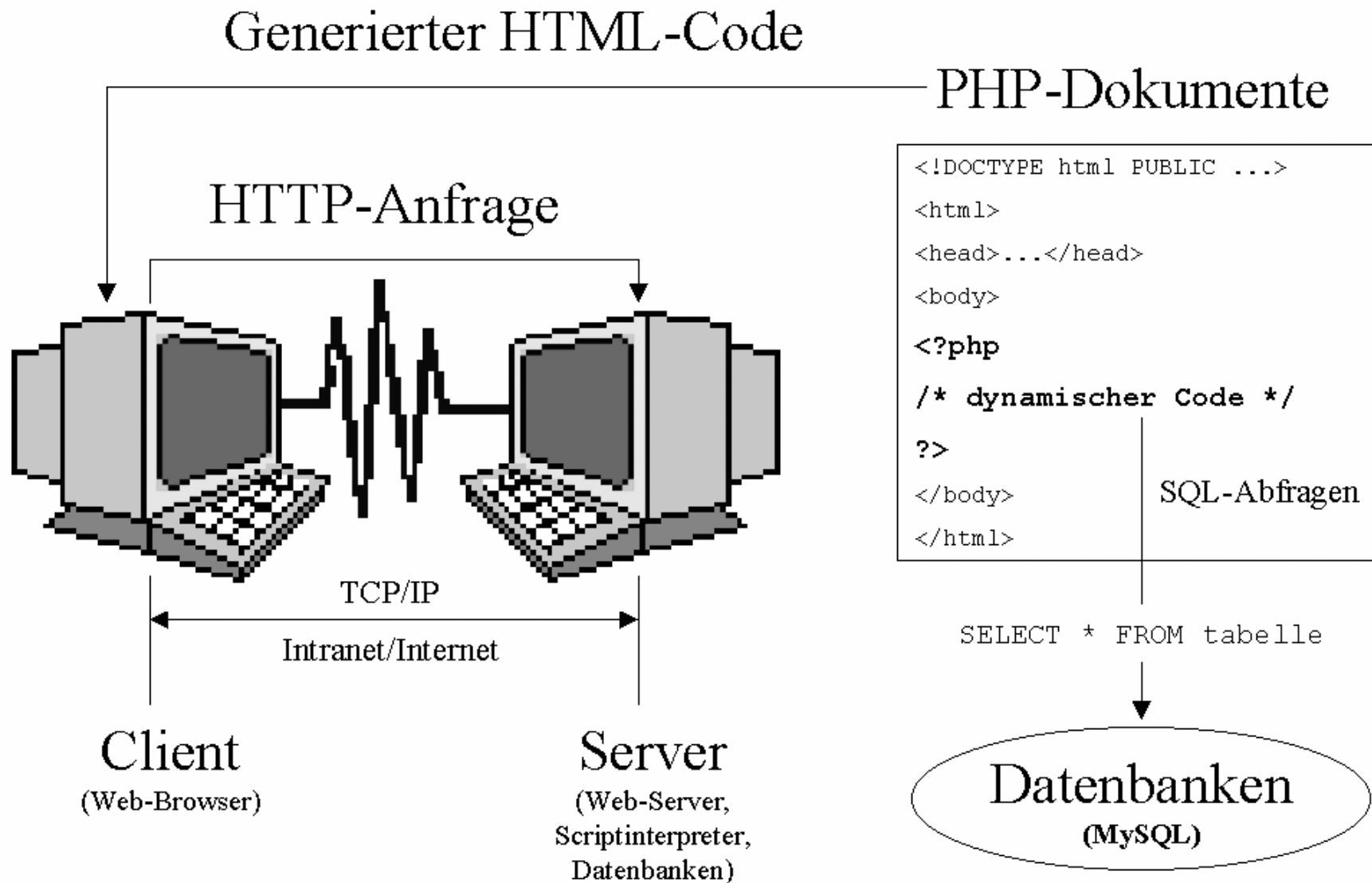
Zum Einsatz kommende Technologien:

- Active Server Pages (ASP) auf MS Servern (IIS)
- Java Server Pages (JSP) von Sun Microsystems
- Apache-Webserver auf UNIX-Systemen (ca. 60% weltweit)
  - Sprachen: Perl, PHP, Python, ...
  - SQL-Datenbanken: Oracle, Informix, MySQL, ...

Funktionsprinzip siehe folgende Grafik ←.



# Prinzip einer dynamischen Webanwendung



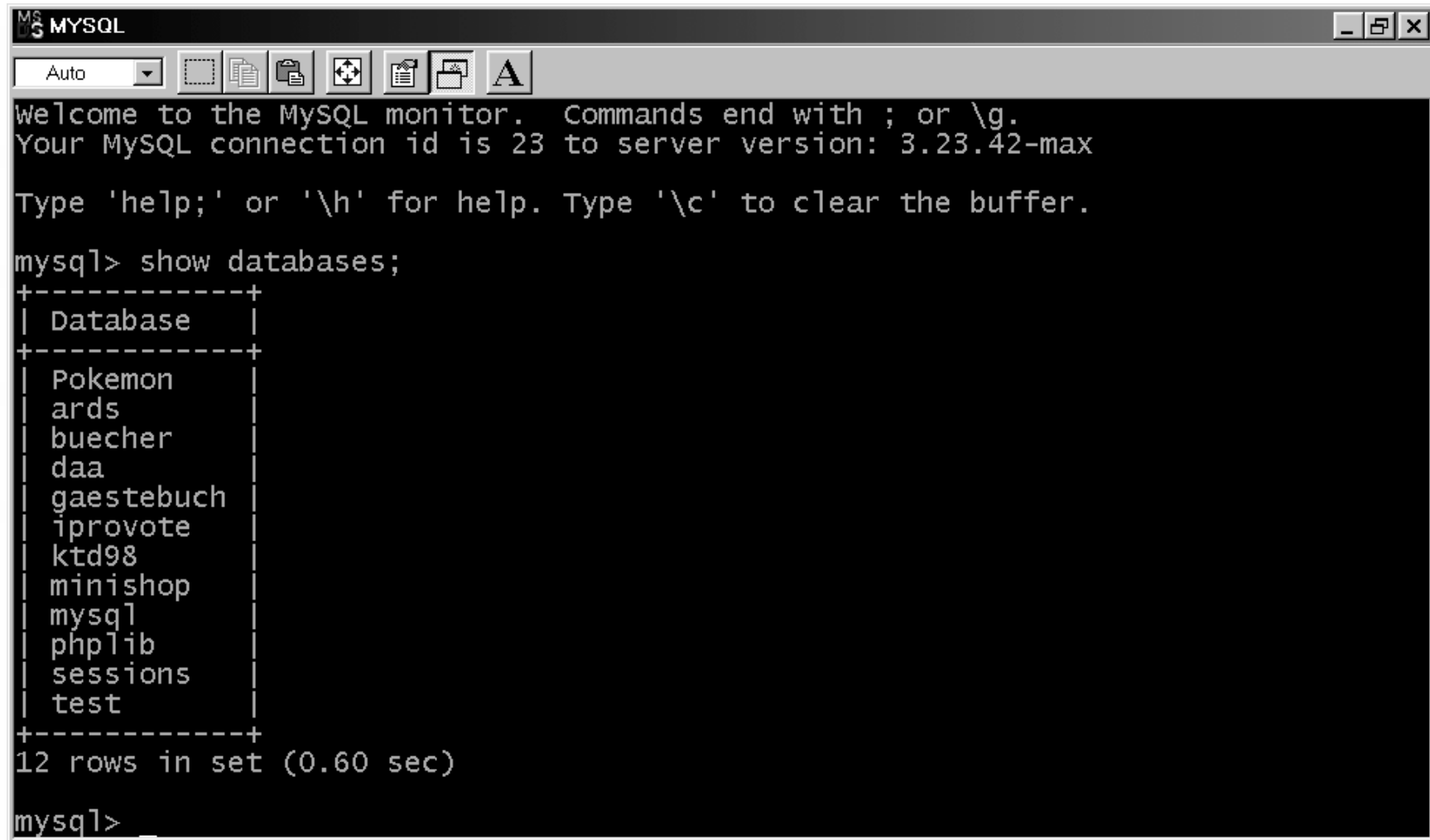
# Die Datenbank MySQL 1/4

## ⇒ Wesentliche Merkmale:

- Relationales Datenbankmanagementsystem (RDBMS) von MySQL AB (seit 1995, vormals TcX)
- Multi-User- und Multi-Thread-Unterstützung
- basiert auf dem Standard ANSI SQL 92
- verfügbar auf praktisch allen Plattformen
- Client-/Server-System (aktuell 3.23.46, 4.0 in Entwicklung)
- sehr gute Integration in populäre Scriptsprachen
  - Erweiterungen:  
Datentypen MEDIUMINT, SET und ENUM  
Pro DB ein Verzeichnis und pro Tabelle 3 Dateien
  - Einschränkungen:  
Keine gespeicherten Prozeduren (stored procedures)  
Kein UNION-Statement, Views, FOREIGN KEYS und Trigger  
Transaktionen sind erst mit neueren Tabellentypen möglich!  
Keine Unterabfragen (Sub-SELECTs) ←.

# Die Datenbank MySQL 2/4

## MySQL-Client unter Win32



The screenshot shows a Windows-style window titled "MS MySQL". The window contains a text-based interface for the MySQL monitor. The text inside the window is as follows:

```
Auto [dropdown] [icons]
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 23 to server version: 3.23.42-max
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> show databases;
+-----+
| Database |
+-----+
| Pokemon |
| ards    |
| buecher |
| daa     |
| gaestebuch |
| improvote |
| ktd98   |
| minishop |
| mysql   |
| php1ib  |
| sessions |
| test    |
+-----+
12 rows in set (0.60 sec)

mysql>
```

# Die Datenbank MySQL 3/4

## phpMyAdmin

The screenshot shows the phpMyAdmin 2.2.0 interface in a Microsoft Internet Explorer browser window. The address bar shows the URL `http://localhost/phpmyadmin-2.2.0/`. The main content area displays the 'Datenbank minishop - Tabelle katalog'. It shows the SQL command `SELECT * FROM `katalog` LIMIT 0, 30` and a table with 10 records. The table has columns for 'Ändern', 'Löschen', 'PID', 'PNAME', 'PINFO', and 'PREIS'. The records are numbered 1 to 10, with product names and prices listed. The interface also includes a sidebar with a 'Home' menu and a 'Zeige' control for displaying 30 records.

		PID	PNAME	PINFO	PREIS
Ändern	Löschen	P001	Produkt1	Text zu Produkt 1	14.80
Ändern	Löschen	P002	Produkt2	Text zu Produkt 2	19.90
Ändern	Löschen	P003	Produkt3	Text zu Produkt 3	9.99
Ändern	Löschen	P004	Produkt4	Text zu Produkt 4	23.50
Ändern	Löschen	P005	Produkt5	Text zu Produkt 5	27.60
Ändern	Löschen	P006	Produkt6	Text zu Produkt 6	118.99
Ändern	Löschen	P007	Produkt7	Text zu Produkt 7	99.00
Ändern	Löschen	P008	Produkt8	Text zu Produkt 8	5.79
Ändern	Löschen	P009	Produkt9	Text zu Produkt 9	115.60
Ändern	Löschen	P010	Produkt10	Text zu Produkt 10	12.95



# Die Datenbank MySQL 4/4

## MySQL-Crashkurs

CREATE DATABASE datenbank

CREATE TABLE tabelle (feld typ optionen, ...)

INSERT INTO tabelle (f1,f2,f3) VALUES (w1,'w2','w3')

SELECT \* FROM tabelle

SELECT f1,f2 FROM tabelle WHERE bedingungen

SELECT ... FROM tabelle ... ORDER BY f1 DESC

UPDATE tabelle SET f3='wert' WHERE f1=5

ALTER TABLE tabelle ADD COLUMN feld typ optionen

DELETE FROM tabelle WHERE f2='wert'

DROP TABLE tabelle

DROP DATABASE datenbank ←.



# Die Scriptsprache PHP 1/20

## Geschichte von PHP

- ⇒ 1994/95 von Rasmus Lerdorf als „Personal Homepage Tools“ entwickelt
- ⇒ 1995 PHP/FI 2.0 (Formular Interpreter)
- ⇒ 1997 ca. 50000 PHP-Installationen, neuer Parser von Zeev Suraski und Andi Gutmans  
PHP = PHP Hypertext Preprocessor
- ⇒ 1998 wurde PHP 3 veröffentlicht
- ⇒ 1999 PHP 3 auf ca. 150000 Websites im Einsatz
- ⇒ 2000 ist das Geburtsjahr von PHP 4 mit größerer Performance
- ⇒ 2001+ Weiterentwicklung von PHP u. a. von Zend Technologies (ca. 7 Millionen Domains mit PHP) ←.

# Die Scriptsprache PHP 2/20

## - Programmiergrundlagen -

### Einbindung in HTML (Dateiendung meistens .php)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<title>P H P</title>
</head>
<body>
  <!-- statisches HTML -->
  <?php

    /* dynamischer PHP-Code */

?>
  <!-- statisches HTML -->
</body>
</html>
```



# Die Scriptsprache PHP 3/20

## - Programmiergrundlagen -

### Ein erster Funktionstest

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<title>P H P</title>
</head>
<body>
<?php

    phpinfo(); /* gibt Systeminformationen aus */

?>
</body>
</html>
```



# Die Scriptsprache PHP 4/20

## - Programmiergrundlagen -

Ausgaben realisieren mit echo() oder print()

[echo() kann mehrere Argumente, print() nur ein Argument haben; die Klammern können entfallen]

```
<?php
```

```
echo "Text1", "Text2", "Text3";  
echo "Text1"."Text2"."Text3";  
print "Text1"."Text2"."Text3";
```

```
echo "Hallo Welt!";  
print "Hallo Welt!";
```

```
?>
```

### Kommentare

```
// einzeliger Kommentar  
/* mehrzeiliger  
Kommentar */
```

# Die Scriptsprache PHP 5/20

## - Programmiergrundlagen -

### Variablen

gültige Zeichen: Buchstaben (am Anfang!), Ziffern und der Unterstrich  
Dollarzeichen am Anfang  
automatische Typzuweisung (kann später noch geändert werden)  
Groß- bzw. Kleinschreibung wird unterschieden!

`<?php`

```
$a=5;           // ganze Zahl
$b=3.14;       // Fließkommazahl
$c=5.08E-6;    // Fließkommazahl mit Exponent
$d=0xFF;       // Hexadezimalzahl
$e=0123;       // Oktalzahl
$xyz="Text1";  // Zeichenketten
$x_y="Text2";
$x12="Text3";
```

### Konstanten

`?>`

```
define("konstante", "wert");
```

# Die Scriptsprache PHP 6/20

## - Programmiergrundlagen -

### Variablen in Ausgaben

```
<?php
```

```
    $xyz="Text1";
```

```
    $x_y="Text2";
```

```
    print "Das ist $xyz und das ist $x_y.\n";
```

```
    echo "Das ist $xyz und das ist $x_y.\n";
```

```
    // oder kürzer: Das ist <?=$xyz?> und das ist <?=$x_y?>
```

```
    print "Das ist ".$xyz." und das ist $x_y.\n";
```

```
    echo "Das ist ".$xyz." und das ist $x_y.\n";
```

```
    /* HTML-Tags können direkt um die Variablen gelegt werden: */
```

```
    print "Das ist <strong>$xyz</strong> und das <em>$x_y</em>.\n";
```

```
?>
```

# Die Scriptsprache PHP 7/20

## - Programmiergrundlagen -

### Rechenoperatoren

<b>Addition:</b>	+	(Zeichenketten mit . verknüpfen)
<b>Subtraktion:</b>	-	
<b>Multiplikation:</b>	*	
<b>Division:</b>	/	
<b>Modulo-Division:</b>	%	(Beispiel: 10 % 3 = 1)
<b>Zuweisung:</b>	=	

```
<?php
    $a=5;
    $b=6;
    $c=$a+$b;
?>
```

Spezielle Rechenoperatoren: ++ -- += -= \*= /= %=

Beispiel: \$i++ steht für \$i=\$i+1 | \$x-=5 für \$x=\$x-5



# Die Scriptsprache PHP 8/20

## - Programmiergrundlagen -

### Logische Operationen

Gleichheit:	==
Gleichheit Werte/Typ:	===
Ungleichheit:	!=
Kleiner:	<
Kleiner gleich:	<=
Größer:	>
Größer gleich:	>=
UND-Verknüpfung:	&&
ODER-Verknüpfung:	
Nicht:	!
Exklusiv-ODER:	xor

### Spezialzeichen maskieren

`\n`    `\t`    `\r`    `\$`    `\"`    `\\`

# Die Scriptsprache PHP 9/20

## - Programmiergrundlagen -

### Abfragen mit if ... elseif ... else-Strukturen

```
<?php
    $a=5;
    $b=6;

    if($a==5 && $b==6)
    {
        /* Code für den Fall $a=6 und $b=6 */
    }
    elseif($a>5 || $b<6)
    {
        /* Code für den Fall $a>6 oder $b<6 */
    }
    else
    {
        /* Code für alle anderen Fälle */
    }
?>
```

# Die Scriptsprache PHP 10/20

## - Programmiergrundlagen -

### Weitere Kontrollstrukturen

```
while (ausdruck)
{
    /* Anweisungen */
}
```

```
do
{
    /* Anweisungen - muss mind. Einmal durchlaufen werden */
}
```

```
while (ausdruck)
```

```
for (startausdruck; bedingung; schrittausdruck)
{
    /* Anweisungen */
}
```

# Die Scriptsprache PHP 11/20

## - Programmiergrundlagen -

### Weitere Kontrollstrukturen

```
switch(wert)
{
    case wert1:
        /* Anweisungen */
        break;

    case wert2:
        /* Anweisungen */
        break;

    case wert3:
        /* Anweisungen */
        break;

    default:
        /* Anweisungen */
}
}
```

# Die Scriptsprache PHP 12/20

## - Programmiergrundlagen -

### Eigene Funktionen

```
function Summe ($a, $b, $c)
{
    return $a+$b+$c;
}
```

Gibt die Summe der Variablen an die aufrufende Instanz zurück.

```
<?php
    $x=5, $y=6, $z=7;
    $d=Summe ($x, $y, $z) ;
    print "Summe: $d\n";
?>
```

Wenn außerhalb der Funktion deklarierte Variablen in der Funktion existieren sollen, müssen diese in der Funktion mit **global \$variable** deklariert werden.

# Die Scriptsprache PHP 13/20

## - Programmiergrundlagen -

### Datenfelder (Arrays)

```
$meinarray[0]="Hallo";  
$meinarray[1]="Welt!";
```

Die Indizes beginnen bei 0 und müssen nicht fortlaufend zugewiesen werden:

```
$meinarray[]="Text";
```

erhält den nächsten freien Index (hier 2).

Andere Schreibweise:

```
$meinarray=array("Hallo","Welt!");
```

### Assoziative Arrays (Felder werden über Namen angesprochen)

```
$farben["rot"]="#FF0000";  
$farben["gruen"]="#00FF00";  
$farben["blau"]="#0000FF";
```

oder: `$farben=array("rot"=>"#FF0000", "gruen"=>"#00FF00", ...)`

# Die Scriptsprache PHP 14/20

## - Programmiergrundlagen -

### Arbeiten mit Formularen

HTML:

```
<form action="script.php" method="post">  
  Vorname: <input type="text" name="vorname">  
  Nachname: <input type="text" name="nachname">  
  <input type="submit" value="Absenden">  
</form>
```

PHP:

```
<?php  
  
  /* Die Inhalte der Eingabefelder stehen in den Variablen  
  $vorname und $nachname zur weiteren Verarbeitung zur Verfügung. */  
  
?>
```



# Die Scriptsprache PHP 15/20

## - Programmiergrundlagen -

### Arbeiten mit Dateien

#### Beispiel: Besucherzähler

```
<?php
```

```
$ref=fopen("zaehler.txt","r+"); // Datei öffnen (read und write)
$z=fread($ref,10);             // max. 10 Zeichen lesen
$z++;                           // +1
rewind($ref);                   // Datei auf Anfang setzen
fwrite($ref,$z);                // neuen Wert schreiben
fclose($ref);                   // Datei schließen
```

```
// Ausgabe als Text
print "Sie sind Besucher <b>$z</b>.<br>\n";
```

```
?>
```





# Die Scriptsprache PHP 16/20

## - Programmiergrundlagen -

### Arbeiten mit Datum und Uhrzeit

```
<?php
```

```
    $zeit=time(); /* Zeitstempel bezogen auf 01.01.1970 00:00:00 */
```

```
    $datum=date("d.m.Y - H:i:s",$zeit);
```

```
    print "Datum: $datum\n";
```

```
    /* Formatierte Ausgabe: 06.12.2001 - 10:23:45 */
```

```
?>
```

### Einfügen externer Dateien

```
include("abc.php"); // fügt das angegebene Script ein
```

```
readfile("xyz.html"); // fügt den Inhalt ungeparst ein
```

```
$zeilen=file("datei.txt"); // liest den Inhalt in ein Array ein
```



# Die Scriptsprache PHP 17/20

## - Programmiergrundlagen -

### Arbeiten mit Sessions zur Nutzeridentifizierung

```
session_start()           // startet eine Session
session_register("variable") // registriert Session-Variablen
session_unset()           // entfernt Session-Variablen
session_destroy()         // beendet Session
```

### Arbeiten mit Cookies zur Nutzeridentifizierung

```
<?php
if($_HTTP_COOKIE_VARS["IPro-Vote_abgestimmt"]=="ja")
{
    /* Code für vorhandenen Cookie ausführen */
}
else
{
    setcookie("IPro-Vote_abgestimmt","ja",time()+30*24*3600);
}
?>
```

# Die Scriptsprache PHP 18/20

## - Programmiergrundlagen -

### Einige Umgebungsvariablen

<code>\$REMOTE_ADDR</code>	IP-Adresse des Clients
<code>\$REMOTE_HOST</code>	Hostname des Clients
<code>\$HTTP_USER_AGENT</code>	Browser-Identifikation

`$HTTP_POST_VARS["varname"]` bzw. `$HTTP_GET_VARS["varname"]`  
sind Arrays mit den Daten aus Formularen.

### Weitere Spezialfunktionen für

- Rechenoperationen
- Zeichenkettenoperationen
- mehrere Datenbanken
- Bildmanipulation (auch Flash, erfordert externe Bibliotheken)
- PDF-Erstellung (erfordert externe Bibliotheken)
- Verarbeitung von XML-Dokumenten mit XSLT
- und vieles mehr ...



# Die Scriptsprache PHP 19/20

## - Programmiergrundlagen -

Einige Funktionen zum Arbeiten mit MySQL-Datenbanken

`mysql_connect("hostname", "username", "password")` → \$referenz

`mysql_select_db("datenbank")`

`mysql_query("SQL-Abfrage", $referenz)` → \$ergebnis

`mysql_fetch_row($ergebnis)`

`mysql_fetch_array($ergebnis)`

`mysql_result($zeile, $feld, $ergebnis)`

`mysql_num_rows($ergebnis)`

`mysql_num_fields($ergebnis)`

`mysql_field_name($ergebnis, $index)`

`mysql_close($referenz)`



# Die Scriptsprache PHP 20/20

## - Programmiergrundlagen -

### MySQL-Funktionen

```
<?php
$dbref=mysql_connect("hostname","username","passwort")
  or die("Kein Zugriff auf DB-Server."); // Verbinden zum DB-Server
mysql_select_db("datenbank",$dbref);    // Datenbank auswählen

$abfrage="SELECT * FROM tabelle";       // SQL-Abfrage formulieren
$ergebnis=mysql_query($abfrage,$dbref)
  or die("Datenabfrage nicht möglich."); // SQL-Abfrage

while($zeile=mysql_fetch_row($ergebnis)) // Ergebnis ausgeben
{
  print "<tr>\n";
  print "<td>$zeile[0]</td><td>$zeile[1]</td><td>$zeile[2]</td>\n";
  print "</tr>\n";
}

mysql_close($dbref);
?>
```

# Praxisbeispiel ARDS-Projekt



# ARDS-Projekt / Ziele

- ⇒ Die Grundidee stammt aus einer Internet-Korrespondenz mit Herrn Dipl.-Med. Giesemann vom Carl-von-Basedow-Klinikum Merseburg.
- ⇒ Es sollen anonymisierte Daten im Rahmen einer medizinischen Studie zum Krankheitsbild ARDS erfasst und später ausgewertet werden.
- ⇒ Entwicklung einer Web-basierten Anwendung, die zur Datenerfassung lediglich einen Browser erfordert.
- ⇒ Die Phase der Datenerfassung soll 6 Monate laufen.
- ⇒ Anschließend erfolgt die Datenauswertung (SQL-Abfragen bzw. Datenexport, z. B. für Excel). ←



# Details zur Realisierung 1/3

## ⇒ Programmierung: PHP

- Login-Prozedur / Session-Handling
- Auswerten der Eingabeformulare
- Eintragen der Daten in die Datenbank (SQL-Abfragen)
- Auslesen von Datensätzen zur Anzeige/Korrektur
- dynamische Generierung von HTML-Code.

## ⇒ SQL-Datenbank: MySQL

- Archivierung von eingegebenen Datensätzen.

## ⇒ Hosting:

- Apache-Webserver unter Linux. ←





# Details zur Realisierung 2/3

<http://www.ards.de>

## Datenerfassung:

jeweils in einer  
MySQL-DB-Tabelle

Eingabe der  
konstanten  
 Stammdaten

(PatID = Name  
der DB-Tabelle)

Stammdaten werden in  
Tabelle patdata  
gespeichert.

ARDS-Projekt - Microsoft Internet Explorer

Adresse <http://www.ards.de/>

### Neue Patienten-Tabelle erstellen

**Stammdaten**

PatID:

Alter:  J. Größe:  cm Gewicht:  kg

Postleitzahl (des Wohnortes des Patienten):

Geschlecht:  weiblich  männlich

Zusatzinformationen:

Zurück ohne Eingaben

Fertig Internet

Formular wird an PHP-Script übergeben

# Details zur Realisierung 3/3

Formular zur Erfassung von Patientendaten

Pro Patient eine ID zur Identifikation

(= Name der DB-Tabelle)

Weitere Funktionen zur Datenanzeige, Datenkorrektur, ...

The screenshot shows a Microsoft Internet Explorer window titled "ARDS-Projekt - Microsoft Internet Explorer". The address bar contains "http://www.ards.de/". The form is titled "I. PatID:" and has a dropdown menu with "TEST0005" selected. To the right of the dropdown is a checkbox labeled "Datenerfassung für diese PatID abschließen? (Grund: Patient ist  geheilt  verstorben)". Below this is a section for "Behandlungsparameter:" with the subtext "(Erfassung der Daten am jeweiligen Tag 06:00 Uhr Ortszeit)". The "Datum der Behandlung:" is set to "08 . 11 . 01". A box labeled "Blutdruck:" contains three input fields: "systolisch: 120 mmHg", "diastolisch: 70 mmHg", and "Mitteldruck: 95 mmHg". Below this, "HF:" is set to "80 /min." and there are radio buttons for "Sinusrhythmus" (selected) and "Arrhythmie". The browser's status bar at the bottom shows "Fertig" and "Internet".

Formular wird an PHP-Script übergeben

# Nützliche Web-Ressourcen

- ⇒ <http://www.php.net>
- ⇒ <http://www.mysql.com>
- ⇒ <http://www.apache.org>
- ⇒ <http://www.dynamic-webpages.de>
- ⇒ <http://www.zugeschaut-und-mitgebaut.de/php>
- ⇒ <http://www2.little-idiot.de/mysql>
- ⇒ <http://mysql-faq.sourceforge.net>
- ⇒ <http://www.koehntopp.de/php>
- ⇒ <http://www.reeg.net>
- ⇒ <http://www.phpwizard.net/projects/phpMyAdmin>
- ⇒ <http://sourceforge.net/projects/foxserv>
- ⇒ <http://www.contentmanager.de> ←



# Zusammenfassung und Ausblick

- ⇒ PHP und MySQL bilden ein „starkes Team“ zur Erstellung dynamischer Websites.
- ⇒ Steile Lernkurve auch bei „Nicht-Programmierern“, wenn solide HTML-Kenntnisse vorhanden sind.
- ⇒ Verfügbar als Open-Source-Software.
- ⇒ Bei den meisten Internet-Service-Providern installiert.
- ⇒ Problemloser Datenaustausch (Scripts, Datenbanken) zwischen verschiedenen Plattformen (z. B. parallele Entwicklung unter Linux und Windows).
- ⇒ Sehr große Entwickler-Community im Internet. ←

