PHP - Einführung

Einstieg und Überblick

René Pfeiffer
Systemadministrator GNU/Linux Manages!
lynx@luchs.at
rene.pfeiffer@paradigma.net

Was uns erwartet...

- Ursprung von PHP
- PHP in HTML Einbetten
- Wie werden Skripte aufgerufen
- Variablen in PHP
- Verbinden mit Datenbanken

Wo man mehr findet...

- PHP Online Manual (http://www.php.net/docs.php)
- SelfPHP (http://www.selfphp.info/)

- Programming PHP (ISBN 1-56592-610-2)
- PHP interaktiv (ISBN 3-936121-01-X)

PHP Geschichte

PHP steht für "PHP: Hypertext Preprocessor"

- 1995 Ursprung als PHP/FI "Personal Home Page / Forms Interpreter"
- 1997 PHP/FI 2.0
- 1997 komplettes Neuschreiben als PHP3
- 1999 komplettes Neuschreiben des Kerns als PHP4
- 2004 Release PHP5

Wichtige Eigenschaften von PHP

PHP ist serverseitige Sprache

- Webserver führt Programme aus
- Browser bekommt nur die Daten

PHP lehnt sich an anderen Sprachen an

- Perl, C, C++

PHP erzeugt die Daten, die der Browser bekommt

Ein einfaches PHP Skript

```
<html>
<head>
<title>PHP Skript</title>
</head>
<body>
 <script language="php">
  echo "<p>Hallo, Welt!</p>";
  echo "Beim nächsten Ton ist es ".date("h:i:s A")."<br>";
 </script>
</body>
</html>
```

Weitere Methoden PHP zu markieren

```
<?php
  echo "<p>Hallo, Welt!</p>";
?>
```

und

```
<?
  echo "<p>Hallo, Welt!</p>";
?>
```

geht ebenso.

Wichtig: Jede Anweisung wird mit einem ";" beendet

Kommentare

In PHP Skripten werden Kommentare mit

- // Kommentartext
- /* Kommentartext */
- # Kommentartext

markiert.

Variablen in PHP

Konventionen

- Variablen beginnen mit einem $ Zeichen
- keine Leerzeichen im Namen
- Buchstaben und Ziffern sind erlaubt
- Groß-/Kleinschreibung wird unterschieden
- keine Sonderzeichen, keine Umlaute
- der _ ist erlaubt
- keine PHP-Schlüsselwörter

Variablentypen

PHP kennt verschiedene Variablentypen

- numerische Variablen (int, double)
- Strings
- boolesche Variable (Inhalt true / false)

PHP wählt den Typ abhängig vom Inhalt
Wichtig: Im Zweifelsfall Typ angeben!

.

Typen testen

Welchen Typs sind sie?

- is_array()
- is_bool()
- is_double()
- is_float()
- is_integer()
- is_numeric()
- is_string()

Felder / Arrays

Felder dienen zum Abbilden von Reihen

```
<?php
   // Erstellen eines Arrays
   $diskspace[0] = 39.4;
   $diskspace[1] = 34.5;
   $diskspace[2] = 32.1;
   // Oder auch so
   $diskspace = Array[39.4, 34.5, 32.1];
?>
```

Mehrdimensionale Arrays

Darf’s eine Dimension mehr sein?

```
<?php
  // Wie in einer Dimension
  // nur anders
  $acker = Array[ Array[ "rot", "grün", "blau"],
                  Array[ "cyan", "gelb", "braun"],
                  Array[ "lila", "violett", "purpur"]];
?>
```

Assoziative Arrays

Zugriff auf Feldelemente mittels Schlüssel

```
$woche = Array( "Montag" => "schlecht",
                "Mittwoch" => "besser",
                "Freitag" => "frei" );
```

liefert

```
echo $woche['Mittwoch']; // Ausgabe: besser
```

Variablentests

- isset() - prüft, ob eine Variable existiert
- unset() - löscht eine Variable
- empty() - prüft, ob eine Variable leer ist

Wichtig für das Abfangen von Benutzereingaben.

Operatoren

- Inkrement-/Dekrementoperator: ++ --
- arithmetische Operatoren: + - * / %
- Zuweisungsoperatoren: = += -= *= /= %=
- Stringoperatoren: . .=
- vergleichende Operatoren: == === != < <= > >=
- logische Operatoren: && and || or ! xor

if/else Abfragen

Was wäre wenn...

```
<?php
   if ( $anzahl_kaffee >= 15 ) {
      echo "Danke, nein.<br>";
   }
   else {
      echo "Ja, noch einen.<br>";
   }
?>
```

for() Schleife

```
<?php
  echo "<table>";
  for ( $zaehler=0; $zaehler<200; $zaehler++ ) {
     echo "<tr><td>Zeile</td><td>$zaehler</td></tr>";
  }
  echo "</table>";
?>
```

Einbinden und Auslagern

PHP kann Teile andere Dateien einfügen

- include() - Einbinden während der Laufzeit
- require() - Einbinden bevor Skript läuft

Damit können HTML-Blöcke und Skripte ausgelagert werden.

require() Beispiel

Laden von Funktionen

```
<?php
   require("funktionen.php");
   // Aufruf einer extern definierten Funktion
   $bildname = hole_dateinamen($url);
   echo "Das Bild heißt $bildname.";
?>
```

include() Beispiel

Auslagern von Skriptteilen

```
<?php
   // ...
   if ( $style == 1 ) {
      include("style_blue.php");
   }
   else {
      include("style_default.php");
   }
?>
```

Interaktion mit Clients

Server und Webbrowser kommunizieren

- Webseite -> Benutzer
  - Aufrufen von Seiten
- Benutzer -> Webseite
  - Eingabe von Daten

Daten mittels GET übertragen

http://du.da.at/cd.php?artikel=42&anzahl=7

ruft das Skript cd.php auf,
übermittel alles nach dem "?" an Server
und erzeugt Variablen:

- $_GET['artikel'] enthält '42'
- $_GET['anzahl'] enthält '7'

Daten mittels POST übertragen

Die POST Methode wird in Webformularen verwendet:

```
<html><body>
<form name="" action="giveme.php" method="POST">
   Album <input type="text" name="album"><br>
   Gruppe <input type="text" name="gruppe"><br>
   <input type="submit" value="Senden">
</form>
</body></html>
```

klick

.

Daten mittels POST übertragen (2)

Die Daten werden ebenso in ein Array gespeichert:

- $_POST['album']
- $_POST['gruppe']

Das Attribut "name" im <input> Tag definiert den Index des Arrays.

PHP Variablen von außerhalb

Es gibt Variablen, die externe Datenquellen haben:

- $_GET[] und $_POST[]
- $_COOKIE[] - Cookies
- $_FILES[] - Dateiuploads
- $_SERVER[] - Infos vom Webserver
- $_ENV[] - Umgebung des Webservers
- $_SESSION[] - registrierte Sessiondaten

Beispiel: GET im Farbeinsatz

Gesucht:
Ein PHP Skript welches durch Aufruf
die Farben der Webseite setzt:

http://du.da.at/farbe.php?bgcolor=000000&text=aaef70

Beispiel: Messen von Zeit

Wie schnell ist ein Skript?

```
<?php
// time() liefter die Sekunden seit
// der UNIX Epoche
// "January 1 1970 00:00:00 GMT"
$start = time();
// Here be commands
$end = time();
echo "Das Skript lief ".($end-$start).
    " Sekunden.<br>";
?>
```

.

Beispiel: Formulareingabe

Aufbau eines Formulares mit den folgenden Eingaben:

- Vorname
- Name
- Straße
- PLZ
- Ort
- Telefonnummer
- Emailadresse

Überprüfen des Inhalts der Eingabe!

Beispiel: Umleiten durch HTTP Header

Umleiten des Browsers
<meta http-equiv="refresh" content="0; URL=URI">

möglich durch HTTP Header

```
<?php
   if ( $redirect ) {
      header("Location: http://www.disney.com/");
      exit;
   }
?>
<html>
<!-- Here be content -->
</html>
```

.

Dateioperationen - Abspeichern von Daten

```
<?php
  $fc = Array( "1. Zeile", "2. Zeile",
               "3. Zeile" );
  $handle = fopen("daten.dat","w");
  // Schreibe Feld $fc in Datei
  foreach($fc as $line)
  {
    fputs($handle,$line);
  }
  fclose($handle);
?>
```

Stringoperationen explode() und implode()

- implode() - baut aus Array String zusammen
- explode() - nimmt String auseinander

Nützlich für CSV Dateien
(Comma Seperated Values)

Beispiel: implode()

```
<?php
// Array mit Inhalt erzeugen
$feld = array('Nachname', 'Email', 'Fon');
// String zusammenbauen
$comma_separated = implode(",", $feld);

echo $comma_separated; // Nachname,Email,Fon

?>
```

Beispiel: explode()

```php
<?php
  $pizza = "Schinken Käse Ei Spinat";
  $belag = explode(" ", $pizza);
  echo $belag[0]; // Schinken
  echo $belag[1]; // Käse
  echo $belag[2]; // Ei
  echo $belag[3]; // Spinat
?>
```

Verbindungen zu Datenbanken

PHP unterstützt eine Reihe von Datenbanken

- MS SQL
- MySQL
- Oracle
- Postgres

Verfügbarkeit hängt von PHP Installation ab

Structured Query Language (SQL)

Datenbanken verstehen häufig SQL

- Datenimport
- Datenmanipulation
- Datenextraktion

SQL - CREATE TABLE

Erstellen von Datenbank und Tabellen:

```
CREATE DATABASE music;
USE music;
CREATE TABLE musiclink (
   ID      INT,
   Created DATETIME,
   Band    VARCHAR(128),
   Album   VARCHAR(128),
   URL     VARCHAR(160),
   PRIMARY KEY ('ID','URL')
);
```

.

SQL - INSERT

Einfügen von Daten

```
INSERT INTO musiclink
 (Created,Band,Album,URL)
 VALUES
 (NOW(),'The Ghost of Lemora',
  'Reach for the Ground',
  'http://www.theghostoflemora.co.uk/');
```

SQL - SELECT

Auslesen von Daten

```
SELECT * FROM musiclink WHERE
 Album='Floodland';

SELECT URL,Album FROM musiclink WHERE
 Band='Scary Bitches';
```

SQL - SELECT verfeinert

Auslesen und Suchen von Daten

```
SELECT URL FROM musiclink WHERE
 Band LIKE '%mercy%';

SELECT Band,URL FROM musiclink WHERE
 Album LIKE 'The%';
```

Verbinden zu Datenbanken in PHP

```
<?php
  $db_user     = 'murtaugh';
  $db_pass     = 'jo5pesc1';
  $db_database = 'depot';
  $db_host     = '127.0.0.1';
  $db_port     = '3306';
  $dblink = mysql_connect($db_host,
              $db_user, $db_pass);
  if ( $dblink ) {
    # Verbindung erfolgreich
  }
  else {
    exit(10);
  }
?>
```

Auswählen einer Datenbank

```
<?php
  // Verbinden
  $dblink = mysql_connect($db_host, $db_user,
                          $db_pass);
  // Datenbank "foobar" auswählen
  // Entspricht "USE foobar;"
  $selected = mysql_select_db('foobar', $dblink);
  if (!$db_selected) {
    die ('Cannot use DB : '.mysql_error());
  }
?>
```

SQL-Kommandos absetzen

Wie kommt das SQL zur Datenbank?

```
<?php
  $sql = "CREATE DATABASE user_xxx";
  $result = mysql_query($sql,$dblink);
  if ( !$result ) {
    die('Ups, Fehler: '.mysql_error());
  }
  // Hier geht es weiter
?>
```

Längere SQL-Kommandos (1)

```
<?php
  $sql  = "CREATE TABLE kontakt (";
  $sql .= "ID INT auto_increment,";
  $sql .= "Vorname VARCHAR(100),";
  $sql .= "Name VARCHAR(100),";
  $sql .= "Email VARCHAR(120)";
  $sql .= ")";
  $result = mysql_query($sql,$dblink);
  if ( !$result ) {
    die('Ups, Fehler: '.mysql_error());
  }
?>
```

Längere SQL-Kommandos (2)

```
<?php
  $sql  = "INSERT INTO kontakt ";
  $sql .= "(Vorname,Name,Email) ";
  $sql .= "VALUES ";
  $sql .= "('René','Pfeiffer',";
  $sql .= "'lynx@luchs.at')";
  $result = mysql_query($sql,$dblink);
  if ( !$result ) {
    die('Ups, Fehler: '.mysql_error());
  }
?>
```

Längere SQL-Kommandos - sprintf

Kommandos besser Formatieren

```
<?php
   // Format definieren
   $frmt  = "INSERT INTO kontakt ";
   $frmt .= "(Vorname,Name,Email) ";
   $frmt .= "VALUES ('%s','%s','%s')";
   // SQL String mit Daten füllen
   $sql   = sprintf( $frmt, $_POST['vorname'],
            $_POST['name'], $_POST['email'] );
   $result = mysql_query($sql,$dblink);
   if ( !$result ) {
      die('Ups, Fehler: '.mysql_error());
   }
?>
```

.

Abfragen mit SQL

```
<?php
  // Abfrage zusammenbauen
  $sql  = "SELECT Vorname,Name,Email ";
  $sql .= "FROM kontakt LIMIT 15";
  $result = mysql_query($sql,$dblink);
  // Wenn Ergbnisse vorhaben, dann diese
  // holen und ausgeben
  if ( $result ) {
    while ($row = mysql_fetch_array($result,MYSQL_ASSOC)) {
      echo "Vorname: ".$row['Vorname']."<br>";
      echo "Name: ".$row['Name']."<br>";
      echo "Email: ".$row['Email']."<br>";
    }
  }
?>
```

Gästebuch

Gästebuch mit

- Name
- Ort
- Email
- Homepage
- Titel
- Textfeld mit Beitrag

Gästebuch - SQL Struktur

```
CREATE TABLE guestbook (
   ID INT AUTO_INCREMENT,
   Name VARCHAR(128),
   Ort VARCHAR(128),
   Email VARCHAR(160),
   Homepage VARCHAR(240),
   Titel VARCHAR(240),
   Beitrag TEXT,
   PRIMARY KEY ('ID')
);
```

.