

Dynamische Webseiten mit PHP

Einführungskurs der VHS Karlsruhe

Mario Kicherer

<http://www.empanyc.net>

Begleitend zum Buch

„Grundlagen Erstellung dynamischer Webseiten“

Herdt Verlag

1. Februar 2009

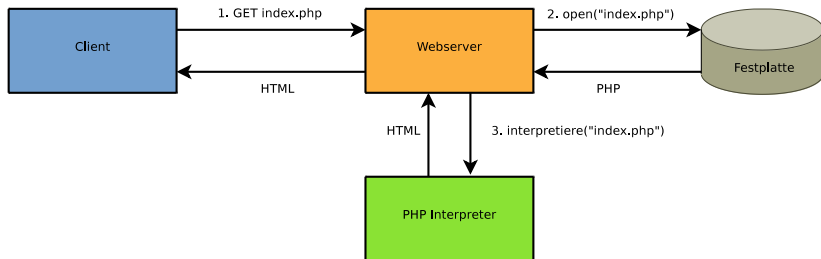
Was ist PHP?

- „PHP: Hypertext Preprocessor“, ursprünglich „Personal Home Page Tools“
- 1995 von Rasmus Lerdorf entwickelt
- Skriptsprache, optimiert zur Erstellung von dynamischen Webseiten

Auslieferung einer HTML Seite



Auslieferung einer PHP Seite



Zusammenspiel von HTML und PHP I

- PHP-Code wird in HTML-Code eingebettet
- eine .php-Datei kann auch ausschließlich HTML-Code enthalten

```
1 <html>
2 <body>
3     <?php print "Hallo Welt!"; ?>
4 </body>
5 </html>
```

Zusammenspiel von HTML und PHP II

Resultat:

```
1 <html>
2 <body>
3     Hallo Welt!
4 </body>
5 </html>
```

Übungs-Webpace

- Dateien hochladen mittels FTP Client
- HTML-Verzeichnis `~/public_html/`
- Abrufbar unter `http://$serverip/~$name/`

Kommentare (Seite 10)

```
1 <html>
2 <body>
3 <?php
4     // ein einfacher Kommentar
5     echo "Hallo Welt!";
6 ?>
7 </body>
8 </html>
```


Ausgabe (Seite 11)

```
1 <html>
2 <body>
3 <?php
4     echo "<h1>Hallo Welt!</h1> <br>";
5     echo "<br><i>Wir lernen PHP!</i>";
6 ?>
7 </body>
8 </html>
```

Fehlermeldungen (Seite 13)

Exemplarische Fehlermeldung:

```
1 Parse error: syntax error
```

Häufigste Ursache:

- Semikolon vergessen

Übung (Seite 14)

Erstellen Sie eine einfache PHP-Seite mit verschiedenen HTML-Tags.

Variablen (Seite 16)

- PHP kennt verschiedene Variablentypen:
 - integer
 - double
 - string
 - array
 - object
- Variablen müssen mit einem \$ beginnen.
- Variablen müssen nicht definiert werden.
- Eine Variable erhält einen Wert mittels des Zuweisungsoperators „=“.

Variablen für Zahlenwerte (Seite 18)

Die gängigen Zahlenoperatoren sind auch in PHP vorhanden:

- „+“ Addition
- „-“ Subtraktion
- „*“ Multiplikation
- „/“ Division
- „%“ Modulo

Variablen für Zahlenwerte (Seite 18)

```
1 <?php
2     $preis_apfel = 2.50;
3     $menge1 = 4;
4     $menge2 = 10;
5     $gesamtpreis = $preis_apfel * ($menge1 + $menge2);
6     echo $gesamtpreis;
7 ?>
```

Variablen für Zeichenketten (Seite 19)

Zeichenketten können mittels folgender Operatoren verbunden werden:

- „.“ Verbinden mehrerer Zeichenketten
- „.“=“ Anhängen einer Zeichenkette an eine andere

Variablen III (Seite 20)

```
1 $preis_apfel = 2.50;
2 $menge = 4;
3 $gesamtpreis = $preis_apfel * $menge;
4
5 $bezeichnung1 = "EUR";
6 $bezeichnung2 = "Kilo";
7
8 $ausgabe = "<p>Aepfel: " . $menge . " " . $bezeichnung2;
9 $ausgabe .= " kosten " . $gesamtpreis . " " . $bezeichnung1 . "
    </p>";
10
11 echo $ausgabe;
12 echo "<p>Aepfel: $menge $bezeichnung2 kosten
    $gesamtpreis $bezeichnung1 </p>";
13 echo '<p>Aepfel: $menge $bezeichnung2 kosten
    $gesamtpreis $bezeichnung1 </p>';
```


Übung (Seite 21)

Aufgabe 2, 3 und 5

Wege der Datenübertragung (Seite 22)

Das HTTP Protokoll bietet zwei Wege um Daten vom Besucher zum Server zu übertragen: GET und POST

GET-Methode

- Daten werden über die URL übertragen („Adresszeile“)
- Eingaben können über die Adresszeile verändert werden
- Eingaben können mittels Bookmarks gesichert werden.
- Datenmenge ist begrenzt

Wege der Datenübertragung II (Seite 22)

POST-Methode

- Daten werden als extra Eintrag im Header der HTTP-Anfrage gespeichert.
- Eingaben können nicht so einfach(!) verändert werden.
- Daten tauchen nicht in Server-Logs und der History auf
- Datenmenge kann beliebig groß sein

Formulare (Seite 24)

```
1 <form action="formular_auswertung.php" method="POST">
2   Vorname: <input type="Text" name="vorname"><br>
3   Nachname: <input type="Text" name="nachname"><br>
4   Wohnort: <input type="Text" name="ort"><br>
5
6   <input type="Submit" value="Abschicken"> <input type
7   = "Reset" value="Zuruecksetzen">
</form>
```

Formularauswertung (Seite 25)

Eingaben stehen je nach Übertragungsweg in `$_GET[]` und `$_POST[]`.

```
1 <?php
2     echo "<p>Folgende Daten wurden uebermittelt:</p>";
3     echo "Vorname: " .$_POST["vorname"] ."<br>";
4     echo "Nachname: " .$_POST["nachname"] ."<br>";
5     echo "Wohnort: " .$_POST["ort"] ;
6 ?>
```

Übung (Seite 26)

Aufgabe 3 - 5

Vergleichsoperatoren (Seite 29)

PHP bietet die folgenden Vergleichsoperatoren:

- „==“ liefert TRUE wenn beide Werte gleich sind
- „===“ liefert TRUE wenn beide Werte gleich sind und denselben Typ haben
- „!=“ ungleich
- „<“ „>“ größer/kleiner
- „<=“ „>=“ größer/kleiner gleich

Verknüpfung von Bedingungen (Seite 33)

Bedingungen können durch folgende Operatoren verknüpft werden:

- „and“ „&&“ Und-Verknüpfung
- „or“ „||“ Oder-Verknüpfung
- „xor“ Entweder-Oder-Verknüpfung
- „!“, Nicht-Operator

Switch-Ausdruck (Seite 36)

Vereinfachte Schreibweise für abhängige IF-Ausdrücke

```
1 <?php
2 if ($menge <5)
3 {
4     echo "Sie haben $menge Kilo bestellt. Der Versand
5         kostet pauschal 7,00 EUR";
6 }
7 else
8 {
9     if (($menge >= 6) && ($menge <= 7.5))
10    {
11        echo "Sie haben zwischen 6 und 7,5 Kilo bestellt
12            .<br>";
13        echo "Der Versand ist deswegen kostenfrei.";
14    }
15 }
16 ?>
```

Switch-Ausdruck (Seite 36)

Vereinfachte Schreibweise für abhängige IF-Ausdrücke

```
1 switch ($sorte)
2 {
3     case "Jonagold":
4         echo "1,50 EUR.";
5     break;
6     case "Delicious":
7         echo "1,60 EUR.";
8     break;
9 }
```

While-Ausdruck (Seite 38)

```
1 $menge = 1;
2 while ($menge <= $_POST["maxmenge" ])
3 {
4     $gesamtpreis = $preis * $menge;
5     echo "<p>Der Preis fuer $menge kg Aepfel betraegt
6         $gesamtpreis EUR.</p>";
7     $menge++;
8 }
```

For-Ausdruck (Seite 40)

```
1  for($menge=1; $menge <=10; $menge++)
2  {
3      $gesamtpreis = $preis * $menge;
4      echo "<p>Der Preis fuer $menge kg Aepfel betraegt
5          $gesamtpreis EUR.</p>";
6  }
```

Break-Ausdruck (Seite 40)

Der Break-Ausdruck kann benutzt werden um eine Schleife vorzeitig zu unterbrechen.

```
1 $preis = 1.20;
2 $menge = 1;
3 while ($menge <=10)
4 {
5     $gesamtpreis = $preis * $menge;
6     $menge++;
7     echo "<p>Der Preis fuer $menge kg Aepfel betraegt
8         $gesamtpreis EUR.</p>";
9     if ($gesamtpreis >=6) break;
10 }
```

Übung (Seite 43)

Aufgabe 3

Feldertypen (Seite 44)

- Felder kann man sich als eine Gruppe bzw. Liste von Werten vorstellen.
- Felder können mehrdimensional sein.
- Werte in diesen Feldern können unterschiedlich indiziert werden:
 - „numerisch indiziert“, Werte werden mittels eines Zahlenwertes ausgewählt
 - „assoziativ“, Werte werden mittels einer Zeichenkette ausgewählt

numerisch indiziertes Feld (Seite 45)

Erstellung eines numerisch indizierten Feldes:

```
1  $staedte = array("Karlsruhe", "Berlin", "Muenchen");  
2  // oder  
3  $staedte = array();  
4  $staedte[0] = "Karlsruhe";  
5  $staedte[1] = "Berlin";  
6  $staedte[2] = "Muenchen";
```


assoziatives Feld (Seite 46)

Erstellung eines assoziativen Feldes:

```
1 $hauptstadt = array(" Deutschland" => " Berlin" , "  
    Frankreich" => " Paris" , " Italien" => " Rom" );  
2 // oder  
3 $hauptstadt = array();  
4 $hauptstadt[" Deutschland" ] = " Berlin";  
5 $hauptstadt[" Frankreich" ] = " Paris";  
6 $hauptstadt[" Italien" ] = " Rom";
```

Foreach-Ausdruck (Seite 48)

Foreach kann genutzt werden um alle Elemente eines Feldes durchzugehen:

```
1 echo "<table>";
2 echo "<tr><td><b>Land</b></td>";
3 echo "<td><b>Hauptstadt</b></td></tr>";
4
5 // Schluessel (=Land) und Werte (=Hauptstadt) eintragen
6 foreach($hauptstaedte as $land=>$stadt)
7 {
8     echo "<tr><td>$land</td><td align='left'>$stadt</td
9         ></tr>";
10 }
```

Mehrdimensionale Felder (Seite 50)

```
1 $land = array(  
2     "land1" => array(  
3         "Name" => "Oesterreich",  
4         "Hauptstadt" => "Wien",  
5         "Sprache" => "Deutsch"),  
6     "land2" => array( "Name" => "Frankreich",  
7         "Hauptstadt" => "Paris",  
8         "Sprache" => "Franzoesisch"),  
9     "land2" => array( "Name" => "Deutschland",  
10        "Hauptstadt" => "Berlin",  
11        "Sprache" => "Deutsch")  
12 );  
13  
14 $hstadt2 = $land["land2"]["Hauptstadt"];
```

Mehrdimensionale Felder II (Seite 50)

```
1  foreach($staedte as $ausgabe) {
2      list($name, $hstadt, $sprache) = $ausgabe;
3      echo "<tr><td>" . $name . "</td><td>";
4      echo $hstadt . "</td><td>" ;
5      echo $sprache . "</td></tr>";
6  }
```

Übung (Seite 53)

- Aufgabe 2-5
- Zusatz: Lassen Sie sich noch den durchschnittlichen Preis pro kg anzeigen.

Funktionen (Seite 55)

```
1 function fctname($parameter1, $parameter2, ...) {  
2     // ...  
3     return $wert;  
4 }
```

Call-by-value (Seite 58)

```
1 function verdreifache($wert) {  
2     $wert = $wert * 3;  
3     print $wert;  
4 }  
5  
6 $zahl = 10;  
7 verdreifache($zahl);  
8 print $zahl
```

```
1 30  
2 10
```

Call-by-value (Seite 58)

```
1 function verdreifache($wert) {  
2     $wert = $wert * 3;  
3     print $wert;  
4 }  
5  
6 $zahl = 10;  
7 verdreifache($zahl);  
8 print $zahl
```

```
1 30  
2 10
```


Möglichkeiten (Seite 58)

- Call-by-reference
- Globale Variablen
- Rückgabewert

Call-by-reference (Seite 58)

```
1 function verdreifache(&$wert) {  
2     $wert = $wert * 3;  
3     print $wert;  
4 }  
5  
6 $zahl = 10;  
7 verdreifache(&$zahl);  
8 print $zahl
```

```
1 30  
2 30
```

Globale Variablen (Seite 59)

```
1 function verdreifache() {  
2     global $zahl;  
3     $zahl = $zahl * 3;  
4     print $zahl;  
5 }  
6  
7 $zahl = 10;  
8 verdreifache();  
9 print $zahl
```

```
1 30  
2 30
```

Rückgabewert (Seite 59)

```
1 function verdreifache($wert) {
2     $wert = $wert * 3;
3     print $wert;
4     return $wert;
5 }
6
7 $zahl = 10;
8 $zahl = verdreifache($zahl);
9 print $zahl
```

```
1 30
2 30
```

Einbinden von weiteren Skripten (Seite 62)

- Werkzeugkasten aus Funktionen
- Funktionen in Datei sammeln
- PHP bietet zwei Möglichkeiten Dateien einzubinden:
 - `include()`, bindet Datei ein, falls vorhanden
 - `require()`, bindet Datei ein und falls sie nicht vorhanden ist wird das Skript abgebrochen

Beispiel (Seite 63)

```
1 <?php
2     print "Ergebnis der Ueberpruefung:<br>";
3     include ("prueflogin.inc.php");
4
5     echo "<hr>Hier koennen weitere Anweisungen der Datei
6         ausgefuehrt werden.";
7 ?>
```

prueflogin.inc.php:

```
1 <?php
2 if ($_POST["nickname"] == "Mario" && $_POST["kennwort"]
3     == "geheim")
4 echo "Zutritt erfolgt.";
5 else
6 echo "Zutritt erfolgt nicht.";
7 ?>
```

Übung (Seite 64)

Methoden (Seite 67)

- `fopen($dateiname, $modus)`
- `fgets($dateizeiger, $laenge)`
- `fputs($dateizeiger, $daten, $laenge)`
- `fclose($dateizeiger)`

Beispiel (Seite 68)

```
1  $datei = fopen("user.txt", "r");
2  if($datei != false)
3  {
4      $zeile = fgets($datei);
5      echo "<p>1. Zeile der Datei user.txt:</p>";
6      echo $zeile;
7      fclose($datei);
8  }
9  else
10     echo "Es trat ein Fehler auf."
```

read() (Seite 69)

```
1 <?php
2     $feld = file("user.txt");
3     echo "<p></p>";
4     $i=1;
5     foreach($feld as $zeile)
6     {
7         echo "Zeile " . $i++ . ": " ;
8         echo $zeile . "<br>";
9     }
10 ?>
```

Beispiel Counter (Seite 73)

```
1  $name = "counter.txt";
2  $count = 0;
3  $datei = fopen($name, "r+");
4  if ($datei)
5  {
6      flock($datei, LOCK_EX);
7      $count = fgets($datei, 10);
8
9      fseek($datei, 0);
10     // aktuellen Zaehler erhoehen und ausgeben
11     $count = $count + 1;
12     echo "<b>" . $count . "</b>";
13     fputs($datei, $count);
14     fclose($datei);
15 }
```

Übung (Seite 74)

printf() (Seite 76)

- `printf($formatierung, $parameter1, $parameter2,...)`
- `printf('Preis: %05.2f Euro',3.6)`

```
1 Preis: 03.60 Euro
```

printf() (Seite 76)

- `strstr($zeichenkette, $gesuchte_kette)`
- `strpos($zeichenkette, $gesuchte_kette [, $start])`
- `substr($zeichenkette, $start [, $laenge])`
- `strlen($zeichenkette)`
- `trim($zeichenkette)`
- `str_replace($von, $nach, $zeichenkette)`

Beispiel (Seite 82)

```
1  $string = "webmaster@php.net";
2
3  echo 'E-Mail-Adresse: ' . $string;
4
5  $pos = strpos($string, "@");
6  if ($pos >= 0) {
7      $webadresse = substr($string, $pos+1);
8      $webadresse = "http://www." . $webadresse;
9      echo "<br>Web-Adresse: <a href=\"\$webadresse\">
10         $webadresse</a>";
11 } else {
12     echo "<br>Zeichen <b>$zeichen</b> konnte nicht
        gefunden werden!";
13 }
```

Implode/Explode (Seite 90)

```
1  echo "<p><b>explode()</b></p>";
2  $vorgabestring="Elstar; Gala; Jonagold; Boskopp;
   Delicious";
3  echo "<br><b>Vorgabe:</b> " . $vorgabestring;
4
5  $ausgabe = explode(";", $vorgabestring);
6  $laenge = sizeof($ausgabe);
7
8  for ($i=0; $i<$laenge; $i++) {
9      echo "<br>Teilstring $i: $ausgabe[$i] ";
10 }
11 echo "<br>";
12 print_r($ausgabe);
13 echo "<p><b>explode()</b></p>";
14
15 $ergebnis = implode(" * ", $ausgabe);
16 echo "<br>Die Zeichenkette lautet: <br> $ergebnis";
```


Übung (Seite 91)

getdate() (Seite 92 und 94)

```
getdate([$zeitstempel])  
date($format [, $zeitstempel])
```

```
1 $datum = getdate();  
2 print_r($datum);  
3 echo "<p>Stunde: " . $datum["hours"] . "</p>";  
4 echo "<br>Minute: " . $datum["minutes"];  
5 echo "<br>Sekunde: " . $datum["seconds"];  
6 echo "<br>Tag der Woche: " . $datum["wday"] . " = " .  
   $datum["weekday"];  
7 echo "<br>" . date("d.m.y = z.") . " Tag des Jahres " .  
   date("Y");
```

Weitere Zeitfunktionen (Seite 100)

- `time()`
- `mktime([$stunde [, $minute ...])`
- `strtotime($datumszeichenkette)`
- `checkdate($monat,$tag,$jahr)`

Beispiel (Seite 100)

```
1 <?php
2   $tag   = 24;
3   $monat = 8;
4   $jahr  = 1978;
5
6   $geburt = mktime(0,0,0,$monat,$tag,$jahr);
7   $diff   = time() - $geburt;
8   echo ((integer)($diff/86400)) . " Tage liegen zwischen
9         dem heutigen Datum, ";
10  echo date("j.n.Y") . " und dem " . date("j.n.Y",
11        $geburt) . ".";
```

Übung (Seite 103)

Sessions (Seite 105)

- `session_start()` - muss also erstes (vor jeglichem HTML-Code) gestartet werden
- `session_id()`
- `session_name()`
- `session_destroy()`
- Daten können mittels `$_SESSION[]` abgerufen/gesetzt werden

Übung (Seite 113)

MySQL I (Seite 118)

- `mysql_connect($server, $benutzer, $passwort)`
- `mysql_close($verbindungskennung)`
- `mysql_select_db($datenbankname)`
- `mysql_error()`

MySQL II (Seite 118)

- `mysql_query($sqlabfrage)`
- `mysql_num_rows($abfrageergebnis)`
- `mysql_fetch_array($abfrageergebnis)`

Beispiel (Seite 123)

```
1 // Verbindung zur Datenbank aufnehmen
2 $verbindung = @mysql_connect($server, $user, $pass)
3 or die ("Konnte Verbindung zum Server <b>$server</b>
4         nicht herstellen!");
5 // Datenbank auswaehlen
6 mysql_select_db($database, $verbindung)
7 or die ("Fehler beim Zugriff auf die Datenbank
8         $database!");
9 // SQL-Abfrage ausfuehren
10 $sql = "SELECT * FROM $table";
11 $query = mysql_query($sql);
12 if (!$query)
13 echo "<br>SQL-Anweisung fehlgeschlagen.";
```

Beispiel (Seite 123)

```
1 // Anzahl der Datensätze ermitteln und ausgeben
2 $anz = mysql_num_rows($query);
3 echo "<p>In der Tabelle <b>$table</b> befinden sich
   $anz Datensätze:</p>";
4
5 // ermittelte Datensätze in Array speichern und
   anzeigen
6 while ($zeile = mysql_fetch_array($query))
7 {
8 echo $zeile["Name"] .", ", "
9 . $zeile["Strasse"] .", "
10 . $zeile["Ort"] .", "
11 . $zeile["Sorte"] .", "
12 . $zeile["Menge"] . "<br>";
13 }
14 mysql_close($verbindung);
```