

Übersicht

- [Formulare in PHP](#)
 - [Arrays \(=Felder\)](#)
 - [Aufgaben](#)
-

Formulare in PHP

Formulare untersuchen wir anhand des Beispiel-Skripts `kennwort_02.php`. Erste Voraussetzung für das Zusammenspiel zwischen PHP und HTML-Formularen ist ein korrekt installierter Web-Server, der auf PHP zurückgreifen kann. Die zweite Voraussetzung lautet: PHP muss für das Zusammenspiel mit dem Web-Server entsprechend konfiguriert sein. Sind diese beiden Voraussetzungen erfüllt, dann können wir innerhalb eines php-Skriptes auf Formular-Elemente über das `$_POST`-Array zurückgreifen. Der Name des Formularelementes "apachelogin" wird in rechteckige Klammern geschrieben: `["apachelogin"]`:

:

```

14 <?php
15     $antwort = "<br>Anmeldung fehlgeschlagen!<br>";
16
17     // $_POST auslesen
18     $apachelogin = $_POST["apachelogin"];
19     $loginpw = $_POST["loginpw"];
20
21     // Testausgabe
22     echo "Login: ".$apachelogin;
23     echo "\n<br>";
24
25     echo "kw: ".$loginpw;
26     echo "\n<br>";
27
28     // Passworttest
29     if (isset($apachelogin))
30     {
31         if (strtolower($apachelogin)=="php")
32         {
33             if (isset($loginpw) AND ($loginpw=="infoQ2"))
34             {
35                 echo "<br>Du bist drin! Gratuliere . . .<br>";
36             }
37             else
38                 echo $antwort;
39         }
40         else
41             echo $antwort;
42     }
43     else
44         echo $antwort;
45 ?>

```

Zur Erklärung:

- Zentral ist der Zugriff auf die Formular-Elemente in den Zeilen 17 bis 19. `$_POST` ist ein sog. [Array](#), genauer: ein [assoziatives Array](#), bestehend aus Schlüssel-Wert-Paaren.

Frage: Was entspricht hier dem Schlüssel?

Antwort: _____

Frage: ... und was dem Wert?

Antwort: _____

- Die Funktion `isset()` testet, ob der übergebenen Variablen schon ein Wert zugewiesen wurde. Das ist bei (HTML-)Formularen besonders wichtig: Ist bei der Übermittlung der Variablen aus dem Formular irgendetwas schief gelaufen, so kann man dies mit der Funktion `isset()` überprüfen (Die Funktion liefert dann den Wert **False**).
- Was die Funktion `strtolower()` in Zeile 31 bewirkt, finden Sie bitte selbst heraus! (siehe [Aufgaben](#))
- Zur Übergabe-Methode `post` gibt es eine Alternative: `get`, siehe unter [Aufgaben](#) und im [www](#)

[<<<](#)

Arrays (=Felder)

Meist will man Daten irgendwie strukturieren (deshalb gibt's Datenbanken!), Beispiel: Zu jedem/r SchülerIn gehört die im Kurs erreichte Punktzahl, etwa so:

```
# Info-Q2-Noten, ws, 2011-04-01
# Name      Note
Tommy      12
```

Dieses Beispiel lässt sich locker ausbauen: Kursname, KursleiterIn, Halbjahr u. s. w. Unser Interesse liegt aber in der **Datenstruktur**: In einem Feld (PHP: **Array**) können wir diese Struktur nachbilden: an erster Stelle der Name, dann die Punktzahl, in PHP:

```
$schueler = array("Tommy",12);
// alternativ:
$schueler[0]="Tommy";
$schueler[1]=12;
// Ausgabe:
echo $schueler[0]."\n";
// Ergebnis: Tommy
echo $schueler[1]."\n";
// Ergebnis: 12
```

Das Beispiel zeigt: mit der `array`-Funktion können wir in PHP ein neues Array ins Leben rufen, alternativ geht auch der Zugriff über den Index (hier: 0 bzw. 1). Auf die einzelnen Elemente des Feldes greift man ebenfalls über den Index zu, interessant ist die Wirkung der folgenden PHP-Anweisung:

```
echo $schueler;
```

Man kann in PHP auch sog. Wörterbücher vereinbaren: **assoziative Arrays**. Als Index haben wir jetzt nicht mehr eine Zahl, sondern einen Schlüssel wie zum Beispiel einen (eindeutigen!) Namen. Man spricht deshalb auch von Schlüssel-Wert-Paaren, Beispiel (aus dem Skript `infonote_04.php`, gekürzt)

```

1  if (strlen($zeile) > 0 and $zeile[0]<>"#"):
2      $strfeld = explode("\t",$zeile); // "\t" == Tabulator
3      $anzahl = count($strfeld); // Anzahl Elemente in $strfeld
4      if ($anzahl == 3)
5          {
6              $notenfeld[$strfeld[0]]=$strfeld[2];
7          }
8      else // $anzahl == 2 !!
9          {
10         $notenfeld[$strfeld[0]]=$strfeld[1];
11     }
12 endif;

```

Im assoziativen Array `$notenfeld` verwenden wir als Schlüssel `$strfeld[0]` (da steht der Name des Schülers) und als Wert merken wir uns die zugehörige Punktzahl.

Frage: Warum übergeben wir einmal `$strfeld[1]` und das andere Mal `$strfeld[2]` als Wert?

Antwort: _____

Wichtige Array-Funktionen (nur eine kleine Auswahl!):

- `reset($meinArray)`: setzt den Index des Feldes `$meinArray` auf Null (Wichtig, wenn man das Feld einmal durchlaufen möchte).
- `array_keys($meinArray)`: liefert ein Array zurück, in dem die **Schlüssel** des assoziativen Arrays `$meinArray` stehen.
- `array_values($meinArray)`: liefert ein Array zurück, in dem die **Werte** des assoziativen Arrays `$meinArray` stehen.
- `in_array($schueler,$meinArray)`: testet, ob der Eintrag `$schueler` im Array `$meinArray` enthalten ist. Liefert entsprechend TRUE bzw. FALSE zurück.

Weil das Durchlaufen eines Feldes in PHP eine unglaublich wichtige Aufgabe ist, gibt es eine eigene Schleifenkonstruktion (!):

```

1  foreach ($notenfeld as $namen)
2  {
3      echo $namen[0]. "    " . $namen[1]. "\n";
4  }

```

`$notenfeld` ist unser assoziatives Array, ein Eintrag in diesem Array wird jeweils in der Variablen `$namen` gespeichert. Das Namenfeld besteht aus zwei Einträgen (`$namen[0]` und `$namen[1]`), deshalb wird in der Echo-Anweisung erst der Name und dann die Punktzahl ausgegeben. Danach kommt der nächste Schleifendurchlauf, d. h. aus dem assoziativen Array wird der nächste Eintrag in der Variablen `$namen` gespeichert und ausgegeben.

[<<<](#)

Aufgaben

1. Finde heraus, was die Funktion `strtolower()` bewirkt! Schreibe hierzu ein kleines php-Skript, etwa `teste_lower.php`. Was passiert bei Umlauten?

2. Worin besteht der Unterschied zwischen der Übergabe-Methode `post` und der Alternative: `get`? Lade hierzu die HTML-Datei `kw.html` in dein Verzeichnis und verändere sie entsprechend! Vergleiche auch [SelfHtml](#), zu Formularen findet sich wissenwertes im folgenden [HTML-Kurs](#) . . .
3. Untersuche die Formular-Übergaben in der Lehrmittelbücherei. Welche Übergabe-Methode wird verwendet: `post` oder `get`? Überprüfe die Funktion `isset()` in der LMB, ob den Variablen ein Wert zugewiesen wurde? Gibt es Sicherheitsüberprüfungen? Wird mit Arrays gearbeitet?
4. Die [Kennwort-Abfrage](#)

<<<

W. Spiegel, E-Mail: [walter\[dot\]spiegel\[at\]web\[dot\]de](mailto:walter[dot]spiegel[at]web[dot]de)
