Name:	
Klasse:	Datum:

Schule ist doof!

Seitenaufruf mit Variablenübergabe

Beim Aufruf einer PHP-Datei wird deren Quelltext nicht direkt an den Client zurückgeschickt, sondern zuerst an einen Interpreter auf dem Server. Dieser erstellt einen HTML-Code, der dann vom Browser des Clients ausgegeben wird. Beim Aufruf der PHP-Datei kann der Client Werte von Variablen mitschicken, so wie bei Scoogle: Die Eingabe der Adresse `www.scoogle.de/start.php?id=15110` ruft auf dem Webserver von `www.scoogle.de` die Datei `start.php` auf und übergibt die Variable `id` mit dem Wert `15110`.

Im folgenden Beispiel soll beim Aufruf von `doof.php?wer=Schule&wie=doof` eine einfache Seite generiert werden, auf der „Schule ist doof!“ zu lesen ist. Dafür werden zwei Variablen, nämlich `wer` und `wie` übergeben, die durch das kaufmännische Und-Zeichen & voneinander getrennt sind. Folgende Schritte sind nötig:

① Um die PHP-Datei `doof.php` zu erstellen, benötigt man zunächst das normale HTML-Grundgerüst:

```
<HTML>
  <HEAD>
    <TITLE>Schule ist doof!</TITLE>
  <HEAD>
  <BODY>
  </BODY>
</HTML>
```

② In dieses Grundgerüst wird ein PHP-Aufruf eingefügt. Dieser Aufruf startet mit `<?php` und endet mit `?>`. Alles, was dazwischen steht, wird später vom PHP-Interpreter verarbeitet.

```
<HTML>
  <HEAD>
    <TITLE>Schule ist doof!</TITLE>
  <HEAD>
  <BODY>
    <?php
    ?>
  </BODY>
</HTML>
```

③ Nun müssen die Variablen aus der Adresszeile übernommen werden. Dies geschieht über `$HTTP_GET_VARS`. In der eckigen Klammer steht – in einfache Anführungsstriche geschrieben – der Name der Variable, wie sie in der Adresszeile heißt. Der Wert, der mit `wer` übergeben wurde, wird der Variable `$wort1` übergeben, `$wort2` bekommt den Wert übergeben, der als `wie` übergeben wurde.

```
<HTML>
  <HEAD>
    <TITLE>Schule ist doof!</TITLE>
  <HEAD>
  <BODY>
    <?php
      $wort1=$HTTP_GET_VARS['wer'];
      $wort2=$HTTP_GET_VARS['wie'];
    ?>
  </BODY>
</HTML>
```

Wichtig: Damit PHP Variablennamen von Befehlen unterscheiden kann, beginnen sie immer mit einem $-Zeichen. Dabei spielt es keine Rolle, ob es sich um Zahlen oder Text (Zeichenketten) handelt.

Auch wichtig: Jede Code-Zeile endet in PHP mit einem Semikolon!

Name:	
Klasse:	Datum:

Schule ist doof!

Verarbeitung von Variablen

Die Werte von `wer` und `wie` sind nun übergeben und in den Variablen `$wort1` und `$wort2` gespeichert, es passiert aber noch nichts. Der Interpreter muss sie noch verarbeiten:

④ Mit dem Befehl `print` lassen sich Variablenwerte ebenso ausgeben wie normaler Text, in dem auch HTML-Tags enthalten sein dürfen. Hier wird zunächst der Wert der Variable `$wort1` ausgegeben, dann der Text „ist“ (Beachte dabei die Leerzeichen!), dann der Wert der Variablen `$wort2` und schließlich ein Ausrufezeichen! Obwohl die `print`-Befehle untereinander stehen, erfolgt die Ausgabe hintereinander. Für einen Zeilenwechsel müsste `print('<br>');` eingefügt werden.

```
<HTML>
  <HEAD>
    <TITLE>Schule ist doof!</TITLE>
  <HEAD>
  <BODY>
    <?php
      $wort1=$HTTP_GET_VARS['wer'];
      $wort2=$HTTP_GET_VARS['wie'];
      print($wort1);
      print(' ist ');
      print($wort2);
      print('!');
    ?>
  </BODY>
</HTML>
```

Es ist auch möglich, die print-Befehle zu einem Befehl zusammenzufassen. Dabei werden die einzelnen Teile mit Punkten verbunden: `print($wort1.' ist '. $wort2.'!');` Beachte auch hier die Leerzeichen vor und nach dem Wort „ist“!

In diesem Beispiel werden die Variablen einfach nur ausgegeben. Natürlich könnte man auch viele andere Dinge damit tun. Zahlen lassen sich z.B. in einer Rechnung verwenden, Texte lassen sich z.B. rückwärts schreiben oder es lassen sich bestimmte Zeichen durch andere ersetzen.

Schreiben, speichern, hochladen, testen

Um zu testen, ob das Ganze funktioniert, muss die Datei gespeichert und auf den Webserver hochgeladen werden, da sich ja nur dort der Interpreter befindet.

1. a) Übertrage den HTML-/PHP-Code Schritt für Schritt und speichere die Datei unter dem Namen `doof.php`! Kopiere sie dann auf den Webserver!

   b) Teste die Seite, indem du sie mit verschiedenen Übergaben aufrufst, z.B. `doof.php?wer=Ingo&wie=toll` oder `doof.php?wer=Schule&wie=langweilig`!

   c) Füge vor dem ersten `print`-Befehl folgende Zeile in das Script ein:

   `if ($wort1==“Ingo“) { $wort2=“super“; }`

   Lade die veränderte Datei hoch und rufe sie mit `doof.php?wer=Ingo&wie=doof` auf. Was bewirkt die eingefügte Zeile?

   Das doppelte Gleichzeichen ist kein Tippfehler. PHP unterscheidet die Zuweisung eines Wertes mit einem Gleichzeichen vom Vergleich zweier Werte mit zwei Gleichzeichen!

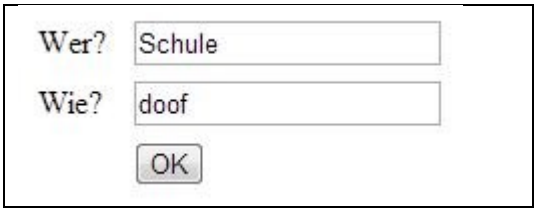
Name:	
Klasse:	Datum:

Schule ist doof!

Auslesen von Formularen

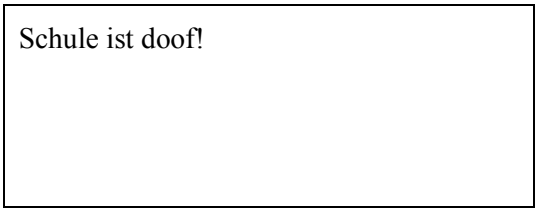
Der Besucher einer Webseite interessiert sich normalerweise ziemlich wenig für die Adresszeile. Wenn er etwas eingeben soll, dann erfolgt das meist über ein Formular. In den folgenden Aufgaben soll ein Formular erstellt werden, in das der Besucher einen Namen und eine Eigenschaft eingibt und nach dem Anklicken eines Buttons eine Seite mit dem passenden Satz erhält. Dafür werden zwei Dateien benötigt, eine HTML-Datei, die ein Formular für die Eingabe enthält und eine PHP-Datei, die das Formular auswertet:

Eingabe durch den Benutzer:



Datei: formular.html

Ausgabe im Browser:



Datei: auswertung.php

2. a) Erstelle eine HTML-Datei mit einem Formular, das zwei Eingabefelder und einen Button enthält (wie oben abgebildet) und speichere die Datei unter formular.html. Gib dabei den beiden Eingabefeldern die Namen „wer“ und „wie“.

   Tipp: Zur Ausrichtung von Text und Formularelementen sind Tabellen sehr nützlich!

   b) Ersetze das `<form>`-Tag durch `<form method="POST" action="auswertung.php">`.

   Hierdurch weiß das Formular, dass beim Klicken des Buttons die Datei auswertung.php aufgerufen werden soll, um die Felder auszulesen.

   c) Öffne nun die Datei doof.php und ändere die beiden $HTTP_GET_VARS-Befehle um in $HTTP_POST_VARS. Speichere die veränderte Datei als auswertung.php.

   $HTTP_POST_VARS liest die Variablen nicht aus der Adresszeile, sondern aus dem Formular, das auf die Seite verweist.

   d) Lade nun beide Dateien hoch und öffne im Browser die Datei formular.html. Gib verschiedene Wörter in die Eingabefelder ein und drücke jeweils auf OK, um die Seite zu testen.

   e) Setze auf die Ausgabeseite einen Link, der wieder zurück zur Eingabeseite führt und gestalte die Ausgabe schöner, z.B. mir Farben und/oder größerer Schrift!

Name:	
Klasse:	Datum:

Schule ist doof!

Vergleichsoperatoren und logische Operatoren

Wie bereits erwähnt unterscheidet PHP die Zuweisung eines Wertes mit einem Gleichzeichen (`$wort2="super"`) vom Vergleich zweier Werte (`$wort1=="Ingo"`) mit zwei Gleichzeichen. Neben dem doppelten Gleichzeichen gibt es auch noch weitere **Vergleichsoperatoren** (siehe Tabelle). Außerdem lassen sich mehrere Aussagen mit **logischen Operatoren** verknüpfen:

==	gleich
!= oder <>	ungleich
<	kleiner
>	größer
<=	kleiner oder gleich
>=	größer oder gleich

```
if (($wort1=="Ingo") and ($wort2!="super")) { $wort2="suuuuuper"; }
```

Der Operator `and` sorgt dafür, dass nur dann `$wort2` auf den Wert „suuuuuper" gesetzt wird, wenn `$wort1` den Wert „Ingo" hat und `$wort2` nicht den Wert „super". Dabei darf übrigens keine der Klammern fehlen!

and oder ||	und
or oder &&	oder
xor	entweder oder
!	nicht

Mit dem `if`-Befehl lassen sich nicht nur Wenn-Dann-, sondern auch Wenn-Dann-Sonst-Anweisungen geben. Hierfür benötigt man zusätzlich das Wort `else`:

Wenn … dann … / sonst …

```
if ($wort1=="Ingo"){
  $wort2="suuuuuper";
}
else {
  $wort2="dooooof";
}
```

3. Was bewirkt die oben aufgeführte Wenn-Dann-Sonst-Anweisung?

4. a) Erstelle eine HTML-Datei mit einem Formular, das einen Button und zwei Eingabefelder mit den Namen „zahl1" und „zahl2" enthält. Speichere die Datei unter zahlenvergleich.html.

   b) Erstelle nun eine PHP-Datei mit dem Namen zahlenvergleich.php, die das Formular auswertet. Der Browser soll am Ende „Zahl1 ist größer als Zahl2." oder „Zahl1 ist nicht größer als Zahl2." ausgeben.

   c) Verändere die PHP-Datei so, dass sie ausgibt, ob die erste Zahl im Vergelich zuer zweiten Zahl kleiner, größer oder gleich ist.

Autor:

Ingo Ostwald
(03.11.2013)