

PHP Basics

Ich hab euch auf den folgenden Seiten noch etwas zusammen kopiert, das eigentlich alles was wir am Donnerstag gemacht haben gut erklärt.

Inhalt

Verwendung Strings:	2
Strings ausgeben	2
HTML-Code ausgeben	2
Steuerzeichen in Strings	3
Auflistung der Escape-Sequenzen	4
Arrays:	5
PHP Indexed Arrays:	5
PHP Associative Arrays	6
Quellen:	7

Verwendung Strings:

Strings ausgeben

Hauptziel eines PHP-Skriptes ist es, unabhängig von der Logik die dahintersteckt, eine Ausgabe zu erzeugen. Beispielweise eine Auflistung der Newsbeiträge. Um in einem PHP-Skript Texte auszugeben (ohne den PHP-Modus zu verlassen) wird üblicherweise der Sprachkonstrukt [echo](#) verwendet. Ein Beispiel könnte wie folgt aussehen.

```
<?php
echo 'Ein Text der ausgegeben wird';
?>
```

Das Erste was auffällt ist die Tatsache dass die Klammern der Funktion fehlen. Das liegt daran dass [echo](#) keine Funktion sondern ein spezieller Sprachkonstrukt ist. Das Manual sagt dazu folgendes.

echo() is not actually a function (it is a language construct), so you are not required to use parentheses with it.

Das Zweite ist dass der Text der ausgegeben werden soll innerhalb von zwei Apostrophs ' steht. Auf diese Weise wird (unabhängig von echo) eine Zeichenkette (Fachbegriff: *String*) mit dem entsprechenden Inhalt erzeugt. Dieser String dient dann als Parameter für den echo-Sprachkonstrukt und [echo](#) gibt diesen String aus. Wenn nun ein PHP-Skript mit diesem Code hochgeladen wird und das Skript aufruft so wird folgender Text ausgegeben.

Ein Text der ausgegeben wird

Fügen wir nun einen weiteren echo-Befehl in unser PHP-Skript ein.

```
<?php
echo 'Ein Text der ausgegeben wird';
echo 'Nun folgt ein anderer Text';
?>
```

Wenn das Skript nun auf einem Webserver läuft wird folgender Text ausgegeben.

Ein Text der ausgegeben wirdNun folgt ein anderer Text

Der zweite String wurde hier einfach hinter den anderen String ausgegeben. Was anderes steht auch nicht im PHP-Skript. Einmal wurde ein String ausgegeben, dann ein weiterer String. Obwohl die beiden echo-Befehle mit einem Zeilenumbruch getrennt sind werden die Strings so wie sie sind ausgegeben. Um einen Zeilenumbruch in der Ausgabe zu haben können wir z.B. das HTML-Element `
` benutzen.

HTML-Code ausgeben

HTML-Code bzw. die besonderen Zeichen `<` und `>` können in Strings ohne Probleme benutzt werden. Somit können wir am Ende des ersten Strings das HTML-Element `
` anfügen.

```
<?php
echo 'Ein Text der ausgegeben wird<br/>';
echo 'Nun folgt ein anderer Text';
?>
```

Wie erwartet steht im Browser den folgenden Inhalt.

Ein Text der ausgegeben wird
Nun folgt ein anderer Text

Wenn der Quellcode im Browser betrachtet wird stehen beide Strings weiterhin hintereinander.

Ein Text der ausgegeben wird
Nun folgt ein anderer Text

Steuerzeichen in Strings

Wenn wir versuchen ein Apostroph innerhalb eines Strings zu schreiben erhalten wir ein Problem, da PHP das Ende des Strings fälschlicherweise zu früh erkennt.

```
<?php
echo 'Dummy Text ' mit Apostroph';
echo 'Ein weiterer Text';
?>
```

Auch an der Einfärbung erkennt man wie PHP den String nur bis zum 2. Apostroph erkennt. Um nun das Zeichen ' in den String zu kriegen kann man z.B. *Double Quoted Strings* verwenden. Das sind Strings die statt den Apostroph die Anführungszeichen " als Begrenzer benutzen.

```
<?php
echo "Dummy Text ' mit Apostroph";
echo "Ein weiterer Text";
?>
```

Eine andere Möglichkeit ist die eine *Escape-Sequenz* einzufügen. In PHP-Strings ist \ das Escape-Zeichen, es wird verwendet wenn man eine solche Escape-Sequenz haben möchte. In *Single Quoted Strings* (Strings mit ' als Begrenzer) wird mit \' ein Apostroph erzeugt.

```
<?php
echo 'Dummy Text \' mit Apostroph';
echo 'Ein weiterer Text';
?>
```

Die gleiche Idee kann verwendet werden wenn man in double quoted Strings ein Anführungszeichen erzeugen will.

```
<?php
echo "Dummy Text \" mit Apostroph";
echo "Ein weiterer Text";
?>
```

Stellt sich nun die Frage wie man ein einzelnes \ erzeugt, ohne dass es gleich als Start einer Escape-Sequenz interpretiert wird. Die Lösung ist das Escapen des Escape-Zeichens.

```
<?php
echo "Double quoted mit \\ im Text";
echo 'Single quoted mit \\ im Text';
?>
```

Obwohl single quoted und double quoted Strings scheinbar die selbe Funktionalität besitzen sind in double quoted Strings mehr Escape-Sequenzen möglich. Die folgende Tabelle enthält eine Liste der zusätzlichen Escape-Sequenzen in double quoted Strings.

Auflistung der Escape-Sequenzen

Zeichen	Beschreibung
\n	Ein Zeilenumbruch
\r	Ein Wagenrücklauf, oft verwendet bei Netzwerkprotokollen.
\t	Ein Tabulator
\v	Vertikaler Tabulator, eher selten verwendet
\f	Seitenvorschub, noch seltener verwendet
\\$	Ein Dollarzeichen. Ohne diese Sequenz wird versucht eine Variable zu finden und dessen Inhalt einzufügen.
\0 bis \777	Ein Zeichen aus dem Bereich von 0x000 bis 0x1FF
\x0 bis \xFF	Ein Zeichen aus dem Bereich von 0x00 bis 0xFF

Arrays:

PHP Indexed Arrays:

Assoziative Arrays verwenden einen Index um mit den einzelnen Elementen zu Arbeiten.

Array erzeugen:

Es gibt 2 Möglichkeiten umso ein Array zu erzeugen:

1. Automatische Vergabe des Index (Beginnt bei 0)

```
$cars=array("Volvo","BMW","Toyota");
```

2. Index manuell vergeben:

```
$cars[0]="Volvo";  
$cars[1]="BMW";  
$cars[2]="Toyota";
```

Beispiel:

Gibt die Länge des Arrays aus.

```
<?php  
$cars=array("Volvo","BMW","Toyota");  
echo count($cars);  
?>
```

Ausgabe des Arrays mittels Schleife:

```
<?php  
$cars=array("Volvo","BMW","Toyota");  
$arlength=count($cars);  
  
for($x=0;$x<$arlength;$x++)  
{  
    echo $cars[$x];  
    echo "<br>";  
}  
?>
```

PHP Associative Arrays

Assoziative Arrays verwenden Keys um mit den einzelnen Elementen zu Arbeiten.

Array erzeugen:

Es gibt 2 Möglichkeiten umso ein Array zu erzeugen:

1. Variante

```
$age=array("Peter"=>"35","Ben"=>"37","Joe"=>"43");
```

2. Variante:

```
$age['Peter']="35";  
$age['Ben']="37";  
$age['Joe']="43";
```

Beispiel:

Ausgabe eines Elements mit echo()

```
<?php  
$age=array("Peter"=>"35","Ben"=>"37","Joe"=>"43");  
echo "Peter is " . $age['Peter'] . " years old."  
?>
```

Ausgabe:

Peter is 35 years old.

Ausgabe des Arrays mittels Schleife:

```
<?php  
$age=array("Peter"=>"35","Ben"=>"37","Joe"=>"43");  
  
foreach($age as $x=>$x_value)  
{  
    echo "Key=" . $x . ", Value=" . $x_value;  
    echo "<br>";  
}  
?>
```

Quellen:

<http://tut.php-quake.net/de/output.html#g1>

http://www.w3schools.com/php/php_arrays.asp