



Dynamische Webseiten mit PHP erstellen

1	Einleitung	2
1.1	Anwendungsgebiete	2
1.2	Voraussetzungen.....	2
1.3	Beispiel	2
1.4	Läuft PHP auf meinem Server?.....	2
1.5	Syntax	2
1.6	Kommentare	3
2	Variablen	3
2.1	Zuweisen eines Wertes.....	3
2.2	Ausgabe	3
2.3	Ausgaben verknüpfen	3
2.4	Variablen Typen.....	4
2.5	Echo.....	4
3	Superglobals	4
4	Mathematik und Vergleichen	5
4.1	Mathematische Operatoren	5
4.2	Vergleichsoperatoren	5
4.3	Logische Operatoren	5
5	If – Else -Elseif	6
5.1	if.....	6
5.2	if ... else	6
5.3	if ... elseif ... else.....	6
5.4	Switch.....	7
6	Loop	7
6.1	while.....	7
6.2	for	7
6.3	foreach.....	8
7	Arrays.....	8
8	Formulare.....	9
9	Beispiele	10
9.1	Text ausgeben	10
9.2	Rechnen.....	10
9.3	Datum ausgeben.....	10
9.4	Wochenende?	10
9.5	Schleife: Zählen bis 10	10
9.6	Datei auslesen und anzeigen.....	10

1 EINLEITUNG

PHP ist eine weit verbreitete Skriptsprache für die Erstellung dynamischer Webseiten. Der PHP-Code wird auf dem Webserver ausgeführt und das Resultat wird typischerweise als HTML Datei an den Browser geliefert.

1.1 ANWENDUNGSGEBIETE

Typische Anwendungsgebiete von PHP sind:

- Erzeugung von dynamischem Inhalt, z.B. in einem CMS (Content Management System)
- HTML-Formularverarbeitungen mit PHP
- Dateien lesen, schreiben, ändern etc.
- Datenbankbindung in z.B. Web Shops
- Erstellen von Bildern, PDF-Dokumenten, XML etc.

1.2 VORAUSSETZUNGEN

Ein paar Voraussetzungen damit die PHP Skripte ausgeführt werden können.

- PHP läuft nur auf einem Webserver
- Damit der Code ausgeführt wird muss der Suffix der Datei «*.php» lauten
- Der PHP Code steht in der Datei zwischen den «<?php ... ?>» Befehlen
- Jeder Code Block muss mit einem Strichpunkt (;) abgeschlossen werden
- Die Sprache beachtet die Gross- / Kleinschreibung bei Variablen Namen

1.3 BEISPIEL

Der PHP-Code kann in eine Webseite eingefügt werden.

```
<html>
  <head>
    <title>PHP Textausgabe</title>
  </head>
  <body>
    <?php
      $zahl1 = 10;
      $zahl2 = 5;
      // multipliziere und zeige Ergebnis an
      echo $zahl1 * $zahl2;
    ?>
  </body>
</html>
```

1.4 LÄUFT PHP AUF MEINEM SERVER?

Mit der «phpinfo()» Funktion kannst du dir die Entwicklungsumgebung anzeigen lassen. Informationen zu Programmversionen, Sicherheitseinstellungen etc. werden angezeigt.

Die Datei um die Serverumgebung zu testen sieht folgendermassen aus:

```
<?php
  phpinfo();
?>
```

1.5 SYNTAX

Der PHP Code steht zwischen <?php und ?>. Befehle werden mit einem Strichpunkt (;) abgeschlossen.

1.6 KOMMENTARE

Hinweise zum Code können Sie mit Kommentaren hinzufügen.

```
<?php
// Ich bin ein Kommentar

/*
Das ist ein...
...Kommentar über zwei Zeilen.
*/
?>
```

2 VARIABLEN

Die Variable ist ein Container für einen Wert. Der Variablenname beginnt immer mit einem Dollarzeichen (\$). Es muss auf die Gross- und Kleinschreibung geachtet werden, d.h. `$email` und `$Email` sind zwei unterschiedliche Variablen.

Gültige Variablenamen beginnen mit einem Buchstaben oder einem Unterstrich.

z.B. `$alter`, `$_Name`, `$Termin`

2.1 ZUWEISEN EINES WERTES

```
<?php
// Variable $name bekommt einen Wert
$name = "Chassot";
?>
```

2.2 AUSGABE

```
<?php
// Variable $name bekommt einen Wert
$name = "Chassot";
// Ausgabe der Variablen
echo $name;
?>
```

2.3 AUSGABEN VERKNÜPFEN

Mit einem Punkt lassen sich einzelne Werte verknüpfen.

```
<?php
// Variablen bekommen einen Wert
$grussformel = "Liebe";
$anrede = "Frau";
$name = "Chassot";
// Ausgabe der Anrede...
echo $grussformel . " " . $anrede . " " . $name;
// ...oder...
echo "Guten Tag " . $anrede . " " . $name;
?>
```

2.4 VARIABLEN TYPEN

Verschiedene Werte können in Variablen abgespeichert werden. PHP ist eine typlose Programmiersprache, d.h. PHP ermittelt selbst welchen Typ der Wert hat.

Typen von Werten sind integer, float, string und bool.

```
<?php
// Variablen Typen
// Ganzzahl
$integer = 27;
// Fließkommazahl
$float = 2.7;
// Text
$string = "Huber";
// Wahrheitswert
$bool = true;
?>
```

2.5 ECHO

Der echo Befehl gibt Text aus. Mit einem Punkt (.) lassen sich die Texte verketteten.

```
<?php
// Text für die Ausgabe
$text = "Hallo Europa!";
// verketteten mit einem Punkt
echo "<h1>".$text."</h1>";
?>
```

3 SUPERGLOBALES

Auf Superglobal-Variablen kannst du von überall her zugreifen. \$_SERVER, \$_POST, \$_GET, \$_ENV etc.

```
<?php
// Ausgabe des Servernamens
echo $_SERVER['SERVER_NAME'];
echo "<br>";
// referer meines Skripts
echo $_SERVER['HTTP_REFERER'];
echo "<br>";
// Infos zum Browser meiner Besucherin
echo $_SERVER['HTTP_USER_AGENT'];
?>
```

4 MATHEMATIK UND VERGLEICHEN

4.1 MATHEMATISCHE OPERATOREN

Operator	Beschreibung	Beispiel	Resultat
+	Addition	$x=2$ $x+2$	4
-	Subtraktion	$x=2$ $5-x$	3
*	Multiplikation	$x=4$ $x*5$	20
/	Division	15/5	3
%	Modulo	5%2	1
++	Inkrement	$x=5$ $++x$	$x=6$
--	Verringerung	$x=5$ $--x$	$x=4$

4.2 VERGLEICHOPERATOREN

Operator	Beschreibung	Beispiel
==	ist gleich	5==8 ist false
!=	ist ungleich	5!=8 ist true
<>	ist ungleich	5<>8 ist true
>	ist grösser als	5>8 ist false
<	ist kleiner als	5<8 ist true
>=	ist grösser oder gleich als	5>=8 ist false
<=	ist kleiner oder gleich als	5<=8 ist true

4.3 LOGISCHE OPERATOREN

Operator	Description	Example
&&	und	$x=6$ $y=3$ $(x < 10 \ \&\& \ y > 1)$ ist true
	oder	$x=6$ $y=3$ $(x==5 \ \ y==5)$ ist false
!	nicht	$x=6$ $y=3$ $!(x==y)$ ist true

5 IF – ELSE -ELSEIF

5.1 IF

if testet eine Bedingung und führt die dahinter stehende Anweisung aus, wenn die Bedingung Wahr ist, ansonsten wird sie ignoriert.

```
<?php
// Wochentag gekürzt abfragen
$d=date("D");
if ($d=="Fri") {
    echo "Schöns Weekend!";
}
?>
```

5.2 IF ... ELSE

if mit zwei möglichen Antworten.

```
<?php
$d=date("D");
if ($d=="Fri") {
    echo "Schöns Weekend!";
} else {
    echo "Ich wünsche dir einen schönen Tag!";
}
?>
```

5.3 IF ... ELSEIF ... ELSE

if mit drei möglichen Antworten.

```
<?php
$d=date("D");
if ($d=="Fri") {
    echo "Schöns Weekend!";
}
elseif ($d=="Sun") {
    echo "Schöne Sontig!";
}
else {
    echo "Ich wünsche dir einen schönen Tag!";
}
?>
```

5.4 SWITCH

Bei mehrfachen if-Abfragen ist eine switch/case Abfrage meist einfacher zu programmieren.

```
<?php
$day = date('l'); // Wochentag

switch ($day) {
    case 'Monday':
        echo 'Wir wünschen Ihnen einen guten Start in die neue Woche!';
        break;

    case 'Wednesday':
        echo 'Heute ist unser Shop geschlossen.';
        break;

    case 'Friday':
        echo 'Schöns Weekend!';
        break;

    default:
        echo 'Guten Tag!';
        break;
}
?>
```

6 LOOP

6.1 WHILE

Bei der while Schleife wird die Bedingung mit jedem Eintritt in die Schleife getestet. Solange der Ausdruck TRUE zurückgibt, wird die Schleife durchlaufen.

```
<?php
$i=1;
while($i<=5)
{
    echo "Die Zahl lautet " . $i . "<br />";
    $i++;
}
?>
```

6.2 FOR

Mit der for Schleife kann eine Aufgabe eine bestimmte Anzahl Mal durchgeführt werden.

```
<?php
for ($i=1; $i<=5; $i++) {
    echo "<h$i>Das ist Überschrift " . $i . "</h$i>";
}
?>
```

6.3 FOREACH

Der foreach-loop vereinfacht das Auslesen von Arrays.

```
<?php
// liste aller Gebäude
$gebäude = array("KOL", "KO2", "CUB", "RAE");
// Titel der Seite
echo "<h1>Liste aller Gebäude</h1>";
// lese array mit Gebäuden aus
foreach($gebäude AS $haus) {
    echo $haus."<br>";
}
?>
```

7 ARRAYS

In Arrays kannst du logisch zusammenhängende Daten speichern. Statt mehrere Variablen anzulegen (z.B. \$gebäude1, \$gebäude2, \$gebäude3...) kann man einen Array mit n-Einträgen erstellen.

```
<?php
// liste aller Gebäude
$gebäude = array("KOL", "KO2", "CUB", "RAE");
// Werte ausgeben
echo $gebäude[0]; // KOL
echo $gebäude[1]; // KO2
echo $gebäude[2]; // CUB
echo $gebäude[3]; // RAE
?>
```


8 FORMULARE

Formulardaten lassen sich mit PHP auslesen und verarbeiten. Die Daten könnte man z.B. per E-Mail versenden, in eine Datenbank speichern oder einfach auf einer weiteren Seite ausgeben.

```
<?php
// einfaches Formular und Ausgabe - Validierung des Inputs fehlt
?>
<html>
<head>
  <title>Ein Formular</title>
</head>
<body>
  <h1>Ein Formular</h1>
  <?php
  // wurde das Formular mit Submit bestätigt?
  if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Variablen definieren
    $name = $email = "";
    // Formulareingabe bereinigen
    function test_input($data) {
      $data = trim($data);
      $data = stripslashes($data);
      $data = htmlspecialchars($data);
      return $data;
    }
    $name = test_input($_POST["name"]);
    $email = test_input($_POST["email"]);
    echo "<h2>Ihre Angaben:</h2>";
    echo "<p>Name: $name</p>";
    echo "<p>E-Mail: $email</p>";
  }
  else {
    // Zeige leeres Formular an
    ?>
    <form method="POST" action="<?php echo
    htmlspecialchars($_SERVER["PHP_SELF"]);?>">
      Name: <input type="text" name="name">
      <br>
      E-Mail: <input type="text" name="email">
      <br>
      <input type="submit" name="submit" value="Submit">
    </form>
    <?php
  } ?>
</body>
</html>
```

9 BEISPIELE

9.1 TEXT AUSGEBEN

```
<?php echo "Ist bald Pause?"; ?>
```

9.2 RECHNEN

```
<?php
$zahl1 = 10;
$zahl2 = 5;
echo $zahl1 * $zahl2; //multiplizieren
?>
```

9.3 DATUM AUSGEBEN

```
<h1>Heute</h2>
<?php
echo "Heute ist der " . date("d.m.Y");
?>
<h1>Morgen</h2>
<?php
$morgen=strtotime("tomorrow");
echo "Morgen ist der " . date("d.m.Y", $morgen);
?>
```

9.4 WOCHENENDE?

```
<?php
$tag = date("w");
if ($tag == 6 or $tag == 0)
{
    echo "Wochenende!";
} else {
    echo "Arbeiten!";
}
?>
```

9.5 SCHLEIFE: ZÄHLEN BIS 10

```
<?php
for($i=0; $i<=10; $i++){
    echo $i . ", ";
}
?>
```

9.6 DATEI AUSLESEN UND ANZEIGEN

```
<?php
$myFile = "code.php";
$fh = fopen($myFile, 'r');
$data = fread($fh, filesize($myFile));
fclose($fh);
echo nl2br(htmlentities($data));
?>
```