

Interaktive Webseiten mit PHP und MySQL

Sommersemester 2003

Martin Ellermann

Heiko Holtkamp

PHP Einführung (28.4.03)

- Quellen
- Grundsätze / Konzepte von Webprogrammiersprachen
- Was ist PHP? / Geschichte
- Funktionsweise
- Syntax
- Variablen
- Typen (Boolean, Integer, Float, String, Array)
- Typkonvertierung
- Operatoren
- Einige Beispiele

PHP Einführung (Forts.)

- Kontrollstrukturen
- Ausgewählte Funktionen
- Operationen mit Dateien
- Operationen auf Arrays
- Erzeugen von mit Grafiken
- html Formulare
- Beispiele

Kontrollstrukturen

- Kontrollstrukturen sind die elementaren Bestandteile einer Programmiersprache
- Sie steuern den Ablauf eines Programms
- Kontrollstrukturen in PHP:
 - if – Bedingung
 - ifelse – Bedingung
 - switch ... case
- Schleifen:
 - while, do-while, for, foreach

if - Bedingung

- Mir der if Bedingung wird entschieden, ob Voraussetzungen für die Ausführung von Programmteilen erfüllt werden.
- Die Syntax der if – Bedingung

```
if (Bedingung)
{
    Wird ausgeführt, wenn Bedingung wahr.
}
```

Beispiel für if

```
$a = 2;
```

```
$b = 1;
```

```
if ($a>$b)
```

```
{
```

```
    print "a ist groesser als b";
```

```
    $b = $a;
```

```
}
```

Vergleichsoperatoren

Vergleichoperatoren zum Steuern von Kontrollstrukturen

===	Identisch mit
==	Gleich
!=	Ungleich
>	Größer
<	Kleiner
>=	Größer gleich
<=	Kleiner gleich

Logische Operatoren

Beispiel	Name	Ergebnis
<code>\$a and \$b</code>	UND	true, wenn beide (<code>\$a</code> und <code>\$b</code>) true sind
<code>\$a or \$b</code>	ODER	true, wenn mind. einer (<code>\$a</code> oder <code>\$b</code>) true ist
<code>\$a xor \$b</code>	Exklusiv-ODER	true, wenn genau einer (<code>\$a</code> oder <code>\$b</code>) true ist
<code>!\$a</code>	NICHT	true, wenn <code>\$a</code> false ist
<code>\$a && \$b</code>	UND	true, wenn beide (<code>\$a</code> und <code>\$b</code>) true sind
<code>\$a \$b</code>	ODER	true, wenn mind. einer (<code>\$a</code> oder <code>\$b</code>) true ist

if – else - Bedingung

- Der else Zweig einer if Anweisung wird durchlaufen, wenn die Bedingung als falsch ausgewertet wurde

```
if (Bedingung)
{
    Wird ausgeführt, wenn Bedingung wahr.
}
else
{
    Wird ausgeführt, wenn Bedingung falsch
}
```

if – else - Bedingung

```
if ($a>$b)
print "a ist groesser als b";
if ($a<=$b)
print "a ist nicht groesser als b";
```

...oder aber den ELSE-Zweig verwenden:

```
if
($a>$b)
print "a ist groesser als b";
else print "a ist nicht groesser als b";
```

elseif

- Mit Hilfe von elseif lässt sich prüfen, ob eine Variable einen bestimmten Wert angenommen hat

```
if ($a > $b)
{
    print "a ist groesser als b";
}
elseif ($a == $b)
{
    print "a ist gleich b";
}
else
{
    print "a ist kleiner als b";
}
```

switch – case

- Mit switch - case lassen sich alternativ zu if - else einfache Bedingungen prüfen. Hier kann leicht überprüft werden, ob z.B. eine Variable einen bestimmten Wert angenommen hat.

```
Switch ($farbe)
{
    Case „rot“:
    //Anweisung
    Break
    Case „blau“:
    //Anweisung
    Break
    Case „gelb“:
    //Anweisung:
    Break
}
```

Schleifen

- Schleifen werden benötigt um Programmteile mehrfach durchlaufen zu lassen.
- Eine Laufvariable legt dabei die Häufigkeit der Durchläufe fest
- Schleifen ohne feste Laufvariable werden durch eine Bedingung gesteuert.
- Der Zustand des logischen Ausdrucks bestimmt, ob die Schleife weiter durchlaufen wird oder nicht.

while Schleife

- Die while Schleife ist eine Schleife mit Eintrittsbedingung und wird so lange ausgeführt, wie die Bedingung wahr ist.

```
while (Bedingung)
{
Code für die Schleife
}
```

Beispiel:

```
$i=1;
while ($i<=10)
{
print $i++;
}
```

do while Schleife

- Die do while Schleife ist eine Schleife mit Austrittsbedingung. Die Schleife wird einmal durchlaufen, bevor die Abbruchbedingung durchlaufen wird.

```
do
{
Anweisung
}
while (Bedingung)
```

Beispiel:

```
$i=0;
do
{
print $i;
}
while ($i>0);
```

Funktionen

- 2 Arten
 - Vordefinierte interne Funktionen
 - PHP Dokumentation
 - Benutzerdefinierte Funktionen

Funktionen

- Aufbau einer Funktion

```
Rueckgabe_wert funktionsname (Parameter 1,  
Parameter 2);
```

- Es gibt hierbei keine Unterscheidung zwischen Groß und Kleinschreibung, in Klammern Beginn und Ende der Parameterliste

- Funktionendefinition in der PHP Dokumentation:

```
(rueckgabety) funktionsname (param_typ param)
```

Vordefinierte Funktionen

- Zeichenkettenfunktionen
- Mathematische Funktionen
- Datums- und Zeitfunktionen
- Arrayfunktionen
- Dateisystemfunktionen
- Bildfunktionen
- Sitzungssteuerung
- Netzwerk, Verschlüsselung, URL, XML, PDF, Mail, etc.

Selbstdefinierte Funktionen

- .. erlauben dem Entwickler eigene Codefragmente zu erstellen, die bestimmte Aufgaben erfüllen sollen. Insbesondere häufig wiederkehrende Aufgaben lassen sich elegant mit eigenen Funktionen lösen.

```
function start_table($border)
{
    echo "<table border=$border>\n";
}
```

```
start_table(1);
```

Zeichenkettenfunktionen

- Filtern von tags z.B. in Formulareingaben

```
string strip_tags (string str[, string  
allowable_tags])
```

- Hinzufügen von Anführungszeichen um Strings

```
string addslashes(string str)
```

- Einen Teil eines String extrahieren

```
string substr(string str, int start[,int length])
```

- String mit MD5 verschlüsseln

```
string MD5(string str)
```

Zeichenkettenfunktionen

- Zeichenlänge eines String ermitteln

```
string strlen(string str)
```

- String in Kleinbuchstaben umwandeln

```
string strtolower (string str)
```

- String in Großbuchstaben umwandeln String

```
string strtoupper (string str)
```

- Der erste Buchstabe des String als Großbuchstabe

```
string ucfirst (string str)
```

Zeichenkettenfunktionen

- Einzelne Elemente des String (anhand eines Trennzeichens) in einem Array einsammeln

```
array explode(string separator, string str)
```

- Ein Array in einen String umwandeln

```
string implode(string separator, array arr)
```

- Die Position eines String (needle) in einem anderen String (haystack)

```
int strpos(string needle, string haystack  
[,int offset])
```

- Wird keine Übereinstimmung gefunden, dann wird False zurückgegeben

- Zwei Sting Elemente vergleichen

```
bool Strcmp(string str1, string str2[,int  
tocompare])
```

Dateisystemfunktionen

- Arbeiten mit Dateien und (Unter-) Verzeichnissen
- Funktionen auf dem Dateisystem
 - Lesen und schreiben in einer Datei
 - Erstellen von Dateien und Verzeichnissen
 - Speichern und Lesen von Dateien
- Dateizugriffsrechte auf dem Server beachten!

Dateisystemfunktionen

- Öffnen einer Datei oder eines URL

```
int fopen(string filename, string  
mode)
```

- Fopen liefert einen Filepointer zurück!
- Der zweite Parameter bestimmt den Modus, in dem die Datei geöffnet wird

Dateisystemfunktionen

Modi von fopen:

- r** öffnet nur zum Lesen
- r+** öffnet zum Lesen und Schreiben
- w** öffnet zum Schreiben
Pointer am Anfang, existiert die Datei nicht, dann wird sie angelegt
- w+** öffnet zum Schreiben und Lesen
Pointer am Anfang, existiert die Datei nicht, dann wird sie angelegt
- a** öffnet zum Schreiben
Pointer am Ende, existiert die Datei nicht, dann wird sie angelegt
- a+** öffnet zum Schreiben und Lesen
Pointer am Ende, existiert die Datei nicht, dann wird sie angelegt

Dateisystemfunktionen

- **Dateityp ermitteln**

```
string filetype(string filename)
```

- **Schließen eines Filehandle**

```
int fclose(int fp)
```

- **Zeiger am Dateiende?**

```
int feof(int fp)
```

- **Einlesen einer Datei**

```
string fread()
```

- **Einlesen einer Zeile einer Datei**

```
string fgets(int fp, int length)
```

- **Zeilenweises Auslesen einer ganzen Datei**

```
array file (string filename[, int use_include_path])
```

Arrayfunktionen

- Funktionen zum Manipulieren und Auswerten von Arrays und Arrayinhalten
 - Zählen des Elemente eines Array (Keine Arrayfunktion!)
`int count (mixed var)`

Arrayfunktionen

- Zusammenführen zweier oder mehrerer Arrays

```
array array_merge(array arr1, array  
arr2[, array ...])
```

- Anfügen eines Elementes an das Ende eines Array

```
array_push(array arr, mixed var)
```

- Rückgabe und Löschen des letzten Wertes eines Array

```
mixed Array_pop (array arr)
```

- Zufällige Auswahl einer oder mehrerer Elemente eines Array

```
mixed array_rand(array input[, int  
num_required])
```

Bildfunktionen

- Mit PHP lassen sich Grafiken im gif, jpeg, png und wbmp Format dynamisch erzeugen.
- Die Funktionen zur Generierung von Grafiken sind an die Erstellung von Vektorgrafiken angelehnt.
- Benötigt wir die GD Bibliothek
- Verfügbar sind Versionen mit und ohne Unterstützung für GIF
- Für die Erzeugung von Bildern werden immer 2 Dateien benötigt. Eine zur Ausgabe der Tags und eine zur Erzeugung der Bilddaten.

Bildfunktionen

- Einfügen einer php Grafik:
- ``
- Die Datei zur Erzeugung der Bilddaten darf nur Informationen zum Bildaufbau enthalten.

```
<?php
    $img = imagecreate(80,80);
    $bcolor = imagecolorallocate($img, 0, 0,0);
    imagerectangle($img, 0,0,79,79, $bcolor);
    header („content-type: image/png“);
    imagepng($img);
    imagedestroy($img);
?>
```

Bildfunktionen

- `imagecreate`

Erzeugt ein neues Bild. Das Bild ist zunächst leer und wird dann mit anderen Funktionen „gefüllt“.

- `imagegif`, `imagepng`, `imagejpg`, `imagewbmp`
- ..gibt das Bild an den Browser aus und bestimmt das Grafikformat
 - Achtung: Die Funktionen existieren immer, es hängt aber vom Typ der Bibliothek ab, welche Formate unterstützt werden.

Bildfunktionen

- **Einfache Linie**
`imageline`
- **Gestrichelte Linie**
`imagedashedline`
- **Pinselftyp**
`imagesetbrush`
- **Rechteck**
`imagerectangle`
- **Ein einzelnes Pixel setzen**
`imagesetpixel`
- **Text mit eingebautem Font erzeugen**
`imagestring`
- **Farbe in der Farbtabelle zuweisen**
`imagecolorallocate`

Html Formulare

- Um eine Anwendung mit Daten des Anwenders zu versorgen, dann müssen die Daten eingegeben werden können.
- Hierfür werden html Formulare verwendet
- Als Eingabeelemente stehen
 - Textzeile, -feld
 - Checkbox
 - Radiobutton
 - Filezur Verfügung

Html Formulare

- Weitere Formularelemente
 - Submit
 - Reset
 - Image
 - Hidden
 - Bedeutung der Attribute
 - Size → Länge des Feldes in Standardzeichen
 - Value → Vorbelegter Wert oder Beschriftung
 - Name → Name nach dem ein Element im Skript identifiziert werden kann
 - Checked → Element ist Aktiv
 - Src → Ort an dem sich das Bild befindet

html Formulare

- Auswerten von Formulareingaben
 - Werden mit einem html Formular Daten übergeben, so erkennt dies php selbstständig
 - Die Felder eines verschickten Formulars_ werden beim Laden der Empfängerseite (angegeben im action-Teil des Formulars) automatisch in Variablen gleichen Namens verwandelt, auf die man im Verlauf des Script-Hauptteils direkt zugreifen kann.
 - Will man auf solche Variablen auch in Funktionen zugreifen, ohne sie global definieren zu müssen, kann man ab PHP 4.2 die globalen Arrays `$_GET` und `$_POST` je nach Übergabeart, verwenden.

html Formular Beispiel

...

```
<form action="php_formular.php" method="post">  
Datum:<input type="text" name="new_datum" size="10">
```

```
Termin:<input type="text" name="new_termin" size="20">
```

```
<input type = "submit" value="Eingeben">
```

```
<input type = "reset" value="Zurücksetzen">
```

```
</form>
```

```
<?php  
echo "Eingabe:<p>";  
echo $new_datum;  
echo "<p>";  
echo $new_termin;  
?>
```

...

Termin

- Am 19. Mai findet um 16.00 Uhr die Vergabe der Projekte statt.
- Eigene Projektvorschläge müssen vorher mit uns besprochen werden!

Danke für eure Aufmerksamkeit!