SPENGERGASSE

ausbildung mit zukunft

DI (FH) Levent Öztürk

Übersicht

- Die Sprache PHP5
- Datentypen
- Unterprogramme und Funktionen
- Arrays und Datenstrukturen
- Klassen und Objekte
- PHP-Befehle
- Standarteingabe und Standardausgabe

Übersicht

- Dateizugriffe
- Datenbankzugriffe MySQL
- Webserver

Die Sprache PHP

- PHP: Hypertext Preprocessor
- Ist eine Open-Source-Skriptsprache
- Auch Objekt Orientiert programmierbar
- Dateierweiterung .php
  - Der Webserver schickt die an den Interpreter, danach wird das Ergebnis an den Browser gesendet
  - User sieht statt PHP-Code nur das Ergebnis

Die Sprache PHP

- Einbinden in HTML-Code
  - Mit spezieller Tag <?php ?>

```
<html>
<head>
</head>
<body>
    <?php
        echo „PHP meldet sich";
    ?>
</body>
</html>
```

Die Sprache PHP

- PHP Skript
  - Andere Dateien werden eingebunden durch
    - include
    - include_once
    - require
    - require_once
  - include liefert Warnung, wenn die Datei fehlt
  - require bricht das Programm ab

Die Sprache PHP

- PHP Skript einbinden
  - Je nach Anwendung werden die nötige Programmteile eingebunden
  - Damit nicht doppelt geschieht, wird mit require_once eingebunden

```
require_once 'mydb.php';
require_once 'mylibraryfunctions.php';
require_once 'formfunctions.php';
```

Die Sprache PHP

- Syntax
  - Für die Funktionen, Variablen und Klassen werden folgende Zeichen erlaubt
    - PHP ist case sensitiv
    - A-Z, a-z, 0-9, _, $
    - Keine umlaute oder §ß erlaubt
    - Das erste Zeichen der Klasse oder Funktion kein Ziffer und kein $ Zeichen
    - Die variablen beginnen immer mit $

Die Sprache PHP

- Syntax und Namenskonventionen
  - Die Länge eines Namens kann bis 255 Byte sein
  - Reservierte Wörter dürfen nicht als Variablennamen verwendet werden
  - Klassennamen, Klassenvariablen und Schnittstellennamen immer mit Großbuchstaben beginnen
  - Konstante immer mit Großbuchstaben
  - Zusammengesetzte Namen z.B. initFunktion()

Die Sprache PHP

- Syntax und Namenskonventionen
  - Zeilenkommentare
    - // Datenbankverbindung
  - Mehrzeilige Kommentar
    - Kann auch einzeilige Kommentare umfassen

```
/* dieser Programmteil ist auskommentiert
connectDatabase();
schowDisplay();
Daher werden die Befehle nicht ausgeführt */
```

Die Sprache PHP

- Anweisungen
  - PHP Befehle enden immer mit ;
  - Mehrere befehle können mit {} zusammengefasst werden
  - Befehle können
    - Ausdruck $a=12;
    - Auswahl if($a>0) $b=$a;
    - Wiederholung mit for($i=0;$i<10;$i++) echo “*“;

Die Sprache PHP

- Datentypen
  - Skalare Typen
    - String
    - Integer 4Byte
    - Float 8Byte
    - Boolean true/false, TRUE/FALSE
  - Zusammengesetzte Typen
    - Array
    - Object

Die Sprache PHP

- Datentypen
  - Spezielle Typen
    - Resource
      - Referenzen auf externe Datenquellen
    - NULL
      - Pseudoreferenz, enthält keinen Wert

Die Sprache PHP

- Literale
  - Zahlen
    - Dezimalzahlen
      - 1234, -234, +23.66, -45.99e12, 44E-16
    - Hexadezimalzahlen
      - 0x45cd, -0x234f
    - Oktalzahlen
      - 0321, -0732, +0333

Die Sprache PHP

- Literale
  - Zeichenketten
    - "Testausgabe"
    - 'Zustandswert'
  - Nichtdruckbare Zeichen werden mit dem Zeichen \ dargestellt
    - Zeilenvorschub "\n"
    - Wagenrücklauf "\r"
    - Tabulator "\t"
    - Anführungszeichen "\""

Die Sprache PHP

- Literale
  - Zeichenketten
    - Variablen zwischen " " werden ausgewertet
      - Echo "Der Wert ist $wert"
      - Ausgabe: Der Wert ist 12
    - Variablen zwischen ‘ werden nicht ausgewertet

Die Sprache PHP

- Konstanten
  - Konstante ändern ihren Wert während des Programmlaufs nicht
  - Definition
    - define (" breite ", 540);
    - const testausgabe=true;
  - Vordefinierte Konstanten
    - Mathematische M_PI, M_SQRT2
    - Magische Konstanten
      - __FILE__, __DIR__, __LINE__, __CLASS__, __METHOD__, __FUNCTION__, __NAMESPACE__

Die Sprache PHP

- Variablen
  - Gültigkeitsbereich: gibt an, aus welchem Programmteil auf die Variable zugegriffen werden können
  - Lebensdauer: gibt a, wie lange der Speicherplatz für diese Variable reserviert ist

Die Sprache PHP

- Variablen
  - Lokale variable innerhalb von Funktionen
  - Lokale variable außerhalb von Funktionen
  - Globale Variable
  - Super Globale Variable
  - Für die Klasse
    - Klassenvariable
    - Instanzvariable
    - Pseudevariable: $this, $self, $parent, $static

Die Sprache PHP

- Variablen
  - Formale Parameter haben gleiche Eigenschaften wie lokale variable, Sichtbarkeit, Lebenddauer
  - Globale Variable
    - Müssen in der Funktion mit dem Schlüsselwort „global“ verfügbar gemacht werden.
    - Vorteil: Parameterübergabe nicht nötig
    - Nachteil: Datenkapselung nicht gewährt. Unübersichtlich bei häufiger Verwendung

Die Sprache PHP

- Variablen
  - Vordefinierte globale Variable
    - $_SERVER
    - $_GET
    - $_POST
    - $_FILES
    - $_COOKIE
    - $_SESSION
    - $_ENV
    - $_REQUEST
    - $GLOBALS

Die Sprache PHP

- PHP Variable
  - Statische Variable
    - Wird mit dem Schlüsselwort „static“ definiert
    - Sichtbarkeit wie lokale Variable Lebensdauer länger

```
<?php
function add($val1, $val2) {
    static $zahl=0;
    $zahl++;
    $erg = $val1+$val2;
    echo  "$zahl.Aufruf $val1 + $val2=$erg<br>";
}
add(2, 3);
add(4, 5);
?>
```

Die Sprache PHP

- Variablen
  - Statische Variable

```
Ergebnis:
1.Aufruf 2 + 3=5
2.Aufruf 4 + 5=9
```

Die Sprache PHP

- Wertetyp
  - Wertetypen werden für die skalare Typen und Arrays verwendet. Die Variablenadresse beinhaltet den Wert.
    - $anz=2;
- Referenztyp
  - Referenztypen werden für Objekte verwendet.
  - Die Variablenadresse beinhaltet die Adresse eines Objektes
    - $date = new DateTime();

Die Sprache PHP

- Funktionen
  - Name der Funktion timetamp
  - Parameterübergabe $d
  - Ergebnis wird mit return zurückgeliefert

```
function timestamp($d){
    return $d->format('YmdHis');
}

$date = new DateTime();
echo timestamp($date);
?>
```

Ergebnis: 20130405150443

Die Sprache PHP

- Funktionen
  - Wertübergabe
    - Funktion darf den Wert verwenden aber nicht verändern
  - Referenzübergabe
    - Funktion darf den Wert der Parameter verändern

```
function callbyref(&$n){
    $n=8;
}
$nr=5;
callbyref($nr);echo $nr;
```

Ergebnis:8

Die Sprache PHP

- Lambda Funktionen
  - Ananyme Funktionen
  - Ein Beispiel dafür ist die länderspezifische Sortierung von Daten

Die Sprache PHP

- Arrays
  - Zusammenhängende Daten oder Objekte
  - Elemente werden über den Index oder Schlüssel adressiert (assoziatives Array)

```
<?php $zahlen = array("null","eins","zwei","drei","vier",
"fünf","sechs", "sieben","acht","neun");
$wt   =array("mo"=>"Montag","di"=>"Dienstag","mi"=>"Mittwoch",
"do"=>„ Donnerstag","fr"=>"Freitag");

echo "$zahlen[1] $zahlen[2] $zahlen[3]<br>";
echo $wt["mo"]." bis ".$wt["do"]."<br>";?>
Ergebnis:
eins zwei drei
Montag bis Donnerstag
```

Die Sprache PHP

- Arrays
  - Leere Array
    - $dummy = array();
  - Einfügen
    - $wt[] ="Heute";
  - Alle Elemente ausgeben
    - print_r ($wt);
  - Arrayelement entfernen
    - unset(["fr"]);

Die Sprache PHP

- Funktionen für Arrays
  - array_fill()
    - $list = array_fill(2,3, "nix");
    - 3 Elemente beginnend mit der Index 2 mit Inhalt „nix“ initialisieren
  - suchen
    - in_array("nix",$list)
    - array_search ("nix",$list) ; //liefert schlüssel oder False
    - array_pusch(), array_pop, array_shift()…
    - sort($list), rsort($list),key($list), next($list), prev($list), current($list)

Die Sprache PHP

- Operatoren

Klamern	()	[]	->	new							
Unäre	!	~	++	--	+	-	(typ)	@			
Referenz	&										
Arithmetische	+	-	*	/	%						
Verkenntungs	.										
Schiebe	<<	>>									
Vergleichs	<	>	<=	>=	==	!=	===	!==	instanceof		
Binäre	&	|	^	~							
Logische	&&	||	!								
Bedingungs	?	:									
Zuweisung	=	+=	-=	*=	/=	%=	.=	<<=	>>=	&=	|=
	^=	~=									
Serialisierund	,										

Die Sprache PHP

- Kontrollstrukturen oder Auswahl

```
if(bedingung1) {
   anweisung1;
}else if (bedingung2}
   anweisung2;
} else {
   anweisung3
}

switch(austrcuck){
case wert1: anweisung1; break;
case wert2: anweisung2; break;
default: anweisung3;
}
```

Die Sprache PHP

- Schleifen

```
for($i=0;$i<10;$i++) {
   if($i == 5) continue;
   echo $i."</br>";
}
$i=0;
while($i<10) {
   echo $i."</br>";
   $i++;
   if($i == 5) break;
}

do
{
   echo $i."</br>";
} while ($i--) ;
```

Die Sprache PHP

- Schleifen

```
foreach($wt as $s =>$w) {
   echo "$s $w</br>";
}

for(reset($wt);current($wt);next($wt)) {
   echo current($wt)."</br>";
}
```

Die Sprache PHP

- Ausgabe

```
printf("format",parameter)
Dezimal         %d
Dual            %b
Hexadezimal     %x
Oktal           %o
Ascii-Zeichen   %c
Reelle Zahl     %f
String          %s
"%10.5s"        Aushabelänge 10, string 5, rechtsbündig
"%-10.5s"       Aushabelänge 10, string 5, linksbündig
"%9.3f"         Aushabelänge 9,3 Nachkommastellen, rechtsbündig
"%-9.3s"        Aushabelänge 9,3 Nachkommastellen, linksbündig
"%04%"          Ausgabelänge 4, mit null gefüllt
$hex=0x6d23ff77;
printf("%012d: %012X<br>",$hex, $hex);
```

Ausgabe:001831075703: 00006D23FF77

Die Sprache PHP

- Ausgabe

```
$zahlen =
array("null","eins","zwei","drei","vier","fünf","sechs",
"sieben","acht","neun");

$wt =
array("mo"=>"Montag","di"=>"Dienstag","mi"=>"Mittwoch","do"=>"
Donnerstag","fr"=>"Freitag");

vprintf("%s %s %s %s %s %s %s %s %s %s</br>",$zahlen);

for(reset($wt);current($wt);next($wt)) {
   echo current($wt)."</br>";
}
```

Die Sprache PHP

- Ausgabe

```
sprintf("format",parameter)
vsprintf("format",$array)

Liefert das Ergebnis als String zurück

fprintf($datei, "format", Parameter)
vfprintf($datei, "format", $array)

Das Ergebnis wird in die Datei geschrieben
```

Die Sprache PHP

- Eingabe
  - Standarteingabe über die Tastatur erfolgt mittels Formular

```
const FORM_NAME = '<form name="Test" action="dateneingabe.php"
method="post">
  Name:<input type="text" size="25" name="name" >
       <input type="submit" name="OK" value="OK"></form>';

echo FORM_NAME;

//dateieingabe.php
<?php
$name= $_POST["name"];
echo "Hallo $name! <br> Willkommen!<br>";
?>
```

Die Sprache PHP

- Fehlermeldungen
  - In php.ini die Variable folgende Werte eintargen
    - error_reporting= E_ALL &~E_NOTICE & ~E_STRICT
  - Oder die Funktion aufrufen
    - error_reporting(E_ALL);
  - Weiter Einstellungen
    - display_errors = On
    - log_errors = On
    - Error_log = /var/log/php/error_log
    - track_errors =On

Die Sprache PHP

- exeptions

```
function mydb_connect(){
   $res = 0;
   if(! $res = @mysql_connect("localhost","pi","raspi")){
        throw new Exception("Db Connect");
   }else {
        return $res;
   }
}
try{
   mydb_connect();
}
catch(Exception $e){
   echo "Datenbank kann nicht geueffnet werden<br>";
   echo "Fehlermeldung: ".$e->getMassage()."<br>";
   echo "Trace:         ".$e->getTraceString()."<br>";
}
```

Die Sprache PHP

- Fehlersuche und Debugging

```
function mydb_connect(){
```

Die Sprache PHP

- Objekt orientierte Programmierung
  - PHP erlaubt neben der Prozeduralen Programmierung auch Objekt orientierte Ansätze
  - Dadurch werden die Projekte übersichtlicher und besser erweiterbar und wartbarer.
  - Braucht aber am Anfang detailliertes Konzept
  - Es sind Zahlreiche Klassenbibliotheken vorhanden

Die Sprache PHP

- Datenkapselung

Die Sprache PHP

- Variablen und Funktionen überladen

Die Sprache PHP

- Interface

Die Sprache PHP

- Namespace

Die Sprache PHP

- Beispiele

Die Sprache PHP

- Nützliche Funktionen
  - String-Funktionen

```
$n      = strlen($s);
$pos    = strpos($s, $x [,$strpos]);
$s      = strstr($s, $x);
$s      = trim($s)              //entfernt Leerzeichen
$s      = substr($s, $start [,len]);  //testen
$s      = str_preat($s, $n);   // s wird n mal wiederholt
$s      = str_replace($pattern, $new, $s); //Pattern wird ersetzt
$arr    = explode($pattern, $s);      // wandelt S in Array
$s      = implode($pattern, $arr);    // wandelt Array in String
$s      = htmlspecialchars($s);  // macht html-Code darstellbar
$s      = htmlentities($s);    // ersetzt int. Zeich durch HTML
$s      = nl2br($s);   // ersetzt \n druch <br />
$s      = wordwrap($s, $n);    // Zeilenumbruch nach n oder 75Zei
$s      = addslasches($s);     // Entwertet Quote durch \
$s      = stripslashes($s);    // Entfernt \
```

Die Sprache PHP

- Nützliche Funktionen
  - Datums- und Zeitfunktionen

```
$s = date("d-m-Y H:i");       // MM-TT-JJJJ SS:MM
                        // dmy : zeistellig,
                        // DMY: Wochentag, Monat Apr.,Jahr 4 Stell
                        // lwFz: Montag, 1, April, Tag im Jahr
                        // His: 24 Stunden, minute mm, sekunde ss
                        // haB: 12 Stunden, am|pm, 24/1000 3 Stell
                        // r: liefert RFC2822 konformen Datum
$d = mktime(0,0,0,10.22.2013);       //erzeugt timestamp
strtotime() versucht gültiger Datum zu erzeugen
$s = date("l",mktime(0,0,0,05,22,1959));
$s = date("r",strtotime("12.12.2012 22:30"));
$s = date("r",strtotime("12-12-2012 22:30:59"));
$s = date("r",strtotime("12/21/2013 22:30:58"));
$s = date("r",strtotime("last year"));
$s = date("r",strtotime("now"));
```

Die Sprache PHP

- Nützliche Funktionen
  - Variablenverwaltung

```
$myvar=1;
isset($myvar)  // überprüft ob die Variable definiert ist
unset($myvar); // Variable entfernt
define("PHP_LIBDIR", "../phplib");
defined("PHP_LIBDIR") //überprüft ob dir Konstante definiert ist
is_numeric($myvar)    // ob die Variable eine Zahl enthält
```

  - Array Funktionen

```
array_key_exist("HTTPS",$_SERVER)    // sucht einen Wert
$arr = range(0,50,2);  // gerade Zahlen 0 – 50 werden zugewiesen
$key = array_search("localhost", $_SERVER); // sucht den Key
setlocale(LC_ALL, "de_DE");
usort($arr, "strcol");         // sortiert mit Umlauten
```

Die Sprache PHP

- Nützliche Funktionen
  - HTTP Funktionen

```
header("location: ../err/php_fehler.php");
       //"location: Umleitung"
       //"Content-Type: text/html; charset=UTF-8"
       //Cache-Control:, expires:, Last-Midified:
headers_sent();
headers_list(); // Headerinformationen die schon geschickt wurden
Print_r(headers_list());
```

Die Sprache PHP

- Nützliche Funktionen
  - URL Funktionen

```
$hdr    = get_headers("http://www.google.com");
print_r($hdr);
$path = "var/www/htm/";
$str= urlencode($path);  //  / wird durch %2F ersetzt
$path = urldecode($str); //  Gegenfunktion
$img=$base64_encode("http://www.443.at/raspi/image/images.png ");
$img=$base64_decode();
```

Die Sprache PHP

- Nützliche Funktionen
  - PHP Authentifizierung

```
$c_pw   = md5($p_pw);  //erzeugt aus Plaintext 32 Byte Ciphertext
$conn   = ldap_connect($server);
$res    = ldap_bind($con,$user,$pass); //true wenn erfolgreich
$src    = ldap_search($con, $base_dn, "Filter");
ldap_count_entries($conn, $src); // >0 OK, falsche Gruppe


if (function_exists('imap_open')) // ob IMAP-Funktion forhanden
$mbox   = imap_open($server, $user, $passwd, OP_READONLY); //true
          imap_close($mbox);
$server = ,{mail.com:977/ssl/novalidate-cert)INBOX‘;
// mail.com:977: server:PORT
// SSL-Modus und keine offizielle Certifizieung
// imap Funktionen müssen vorhanden sein
```

Die Sprache PHP

- Nützliche Funktionen
  - PHP Authentifizierung

```
$c_pw   = md5($p_pw);  //erzeugt aus Plaintext 32 Byte Ciphertext
$conn   = ldap_connect($server);
$res    = ldap_bind($con,$user,$pass); //true wenn erfolgreich
$src    = ldap_search($con, $base_dn, "Filter");
ldap_count_entries($conn, $src); // >0 OK, falsche Gruppe
```

Die Sprache PHP

- Fortsetzung inden Beispielen …