

Fehlerbehandlung bei PHP-Formularen ohne Neueingabe von Daten

Inhalt

Inhalt	I
Abbildungen	I
Listings	I
1 Problemstellung	1
2 Formular anzeigen.....	1
3 Formular formatieren.....	2
4 Formular strukturieren	3
5 Formular validieren	5
5.1 Formulardaten sind fehlerhaft	7
5.2 Formulardaten sind fehlerfrei	8
6 Ergebnis der Validierung ausgeben	9
Anhang	10
Quellen.....	II

Abbildungen

Abbildung 1: Leeres Formular mit Textfeldern sowie Feldern für Einzel- und Mehrfachauswahl	1
Abbildung 2: Meldungen bei fehlerhaften Formulardaten.....	7
Abbildung 3: Auflistung der fehlerfreien Formulardaten	8

Listings

Listing 1: Interne Stilanweisungen für das Beispiel-Formular.....	2
Listing 2: Das Value-Attribut eines Formularfeldes jeweils mit dem gesendeten Wert neu belegen	4
Listing 3 : Formularfeld 'beruf' mit dem gesendeten Wert wiederbelegen	4
Listing 4: Überprüfung der Formulardaten.....	6
Listing 5: Ergebnis der Validierung der Formulardaten ausgeben	9

1 Problemstellung

In diesem Beitrag wird gezeigt, wie die Eingabedaten eines Formulars erhalten werden können, nachdem es abgesendet wurde. Bei der anschließenden Validierung der Formulardaten mit einem PHP-Skript wird erreicht, dass der Benutzer nur die als falsch erkannten Informationen verbessern muss. Alle anderen bleiben erhalten. Dadurch verbessert sich die Benutzerfreundlichkeit enorm, insbesondere bei Formularfeldern mit Mehrfachauswahl. Kurz, um die Benutzerfreundlichkeit eines Formulars zu perfektionieren, muss dafür gesorgt werden, dass die Eingabedaten nach dem Absenden in den Formularfeldern unverändert angezeigt bleiben.

Im Anhang wird das gesamte Listing der Datei *Formulardaten_erhalten.php* gezeigt. Beim ersten Aufruf erzeugt diese PHP-Datei den HTML-Code zum Erzeugen des Formulars, wie in Abbildung 1 dargestellt. Beim Betätigen der Schaltfläche „Formular absenden“ wird die Datei *Formulardaten_erhalten.php* erneut aufgerufen und bekommt die Formulardaten übergeben. Ist mindestens eine Eingabe des Benutzers fehlerhaft, wird die HTML-Seite des Formulars erneut aufgebaut, wobei Fehleingaben entsprechend markiert sind und – *das ist wichtig für die Benutzerfreundlichkeit* – alle Formularfelder wieder mit den zuvor übermittelten Werten belegt werden. Wenn bei der Validierung der Formulardaten *kein* Fehler gefunden wird, werden die übermittelten Daten unterhalb des Formulars zwecks visueller Endkontrolle vollständig aufgelistet. Im Folgenden werden die einzelnen Funktionen der Datei *Formulardaten_erhalten.php* genauer erläutert. Im Anhang wird diese Datei insgesamt aufgelistet.

2 Formular anzeigen

In Abbildung 1 wird ein leeres Formular präsentiert, das den Ausgangspunkt für die folgenden Ausführungen bildet:

PHP: Formulardaten nach dem Absenden erhalten

* Pflichtfeld.

Einzeilige Textfelder

Vorname^{*)}:

Nachname^{*)}:

Auswahlfelder

Geschlecht^{*)}: männlich weiblich

Drop-Down-Auswahlliste: Einzelauswahl

Beruf^{*)}: ▾

Horizontale Auswahlkästchen: Mehrfachauswahl

Bitte Hobbies markieren^{*)}: lesen radfahren schwimmen nordic walk

Abbildung 1: Leeres Formular mit Textfeldern sowie Feldern für Einzel- und Mehrfachauswahl

Einzeilige Textfelder besitzen ein sog. *value*-Attribut, mit dem der jeweilige Inhalt eines Feldes angezeigt wird. Wird beispielsweise das erste Formularfeld (siehe Abbildung 1) mit

```
name = "vorname"
```

benannt, kann dessen Wert nach dem Abschicken des Formulars einfach mit folgender PHP-Zuweisung (vgl. [1])

```
value="<?php echo $vorname;?>"
```

oder sicherer mit

```
value="<?php echo htmlentities($vorname);?>"
```

erneut belegt und angezeigt werden. Bei anderen Formular-Feldtypen (z. B. Auswahllisten) ist Beibehaltung des/der jeweiligen Werte(s) etwas aufwendiger. Mehr Details dazu im Folgenden.

3 Formular formatieren

Das Formular (siehe Abbildung 1) wird mit internen Stilanweisungen¹ formatiert (siehe Listing 1):

```
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <title>Formulardaten erhalten</title>
  <style>
    html, body {padding:0;border:0px none;}
    fieldset{width:auto;border:1px dashed black}
    fieldset div{display:inline-block;margin-top:10px;}
    legend {
      padding:0.2em 0.5em;
      border:1px solid black;
      color:black;
      background-color:yellow;
      font-size:90%;
      text-align:right;
    }
    form {width:800px;margin:0 auto;border:1px solid red;}
    label {
      float:left;
      width:250px;
      padding-left:20px;
      margin:5px;
      text-align:left;
      font-weight:bold;
    }
    span sup, .error{color:red;}
    br{clear:left;}
  </style>
</head>
```

Listing 1: Interne Stilanweisungen für das Beispiel-Formular

¹ Das Aussehen des Formulars mit der Sprache CSS (engl. **Cascading Style Sheet**) gestalten.

4 Formular strukturieren

Das Beispiel-Formular besteht aus 4 Feldgruppen, sog. **<fieldset>**-Elementen, jeweils mit erklärendem Text im sog. **<legend>**-Attribut:

1. Einzeilige Textfelder für Vor- und Nachname
2. Auswahlfelder für Geschlecht
3. Drop-Down-Auswahlliste für Beruf (**Einfachauswahl**)
4. Auswahlkästchen für Hobbies (**Mehrfachauswahl**)

```
<form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>">
  <center>
    <h2>PHP: Formulardaten nach dem Absenden erhalten</h2>
    <p><span class="error">* Pflichtfeld.</span></p>
  </center>
  <fieldset>
    <legend>Einzeilige Textfelder</legend>
    <label for="vorname">Vorname<span><sup>*</sup></span></label>
    <input name="vorname" size="25" maxlength="60" required="required"
      value="<?php
        printf("%s", htmlentities($vorname));
      ?>">
    <span class="error"><?php echo $vorname_Err;?></span>
    <br>
    <label for="nachname">Nachname<span><sup>*</sup></span></label>
    <input name="nachname" size="25" maxlength="60" required="required"
      value="<?php
        printf("%s", htmlentities($nachname));
      ?>">
    <span class="error"><?php echo $nachname_Err;?></span>
  </fieldset><br>
  <fieldset>
    <legend>Auswahlfelder</legend>
    <label for="geschl">Geschlecht<span><sup>*</sup></span></label>
    <input type="radio" name="geschl" value="m" required="required"
    <?php
      printf("%s", ($geschl=='m' ? " checked" : ""));
    ?> > männlich &nbsp;
    <input type="radio" name="geschl" value="w"
    <?php
      printf("%s", ($geschl=='w' ? " checked" : ""));
    ?> > weiblich
  </fieldset><br>
  <fieldset>
    <legend>Drop-Down-Auswahlliste: Einzelauswahl</legend>
    <label for="beruf">Beruf<span><sup>*</sup></span></label>
    <select name="beruf">
      <option value="0"
        <?php
          echo (!isset($beruf) ? " selected" : "");
        ?>
      >--- Bitte wählen ---</option>
      <?php
        print_beruf_options($beruf);
      ?>
    </select>
```

```

    <span class="error"><?php echo $beruf_Err; ?></span>
</fieldset><br>

<fieldset>
<legend>Horizontale Auswahlkästchen: Mehrfachauswahl</legend>
<label for="hobbies">Bitte Hobbies markieren<span><sup>*</sup></span></label>
<span class="error"><?php echo $hobbies_Err; ?></span>
<div>
    <?php
        $hobbies= array('lesen' =>'lesen', 'radfahren' => 'radfahren' , 'schwimmen' =>
'schwimmen', 'nordic walk'=>'nordic walk');
        foreach($hobbies as $key => $opt) {
            if (isset($_POST['hobbies']) && in_array($key, $_POST['hobbies'])) {
                $checked = ' checked="checked"';
            } else {
                $checked = '';
            }
            printf('<input type="checkbox" name="hobbies[]" value="%s" %s>%s',
                $key,
                $checked,
                $opt
            );
        }
    ?>
</div>
</fieldset><br>
<label for="submit">&nbsp;</label>
<input type="submit" name="submit" value="Formular absenden" />
<br>
</form>

```

Listing 2: Das Value-Attribut eines Formularfeldes jeweils mit dem gesendeten Wert neu belegen

In Listing 2 ist die Programmierlogik zur Wiederbelegung der Formularfelder mit den zuvor gesendeten Formulardaten grau hervorgehoben.

Zur Wiederbelegung des Formularfelds 'beruf' wird die PHP-Funktion mit dem Namen 'print_beruf_options' benutzt (siehe Listing 3).

```

<?php
// Formularfeld mit dem Namen 'beruf' wiederbelegen
function print_beruf_options($beruf, $berufe) {
    for ($i=0;$i<count($berufe);$i++) {
        printf("<option value=\"%d\"%s>%s</option>\n",
            ($i+1), ($beruf==($i+1) ? " selected" : ""),
            htmlentities($berufe[$i])
        );
    }
}
?>

```

Listing 3 : Formularfeld 'beruf' mit dem gesendeten Wert wiederbelegen

5 Formular validieren

Die Überprüfung der Eingaben des Benutzers in das Formular erfolgt mit folgenden PHP-Funktionen und weiteren PHP-Codezeilen:

```
<?php
function test_input($data) {
    $data = trim($data);          // Überflüssige Zeichen am Anfang und Ende entfernen.
    $data = stripslashes($data); // Maskierungszeichen entfernen.
    $data = htmlspecialchars($data); // Sonderzeichen in HTML-Codes umwandeln.
    return $data;
}
function removeMultipleSpaces($Nm) { // doppelte Leerzeichen entfernen.
    return preg_replace("/\s{2,}/", ' ', trim($Nm));
}
function validName($pattern,$Nm) { // Vor-/Nachname auf Gültigkeit prüfen
    $Nm = removeMultipleSpaces($Nm);
    if (strlen($Nm) < 2) {
        return "Vor-/Nachname zu kurz.";
    } else {
        if (!preg_match($pattern, $Nm)) { // Name mit Muster vergleichen.
            return "Ungültige Zeichen gefunden";
        }
    }
    return "";
}
function validSurname($nachName) { // Initiale des Nachnamens prüfen.
    // Wenn nur 1 Wort im Nachnamen, dann ...
    if (str_word_count($nachName, 0, 'äüöÄÜÖß') == 1) {
        $firstChar = htmlentities($nachName[0]);
        // Wenn erster Buchstabe in Kleinschreibung, dann ...
        if($firstChar == strtolower($firstChar)) {
            return "Initiale in Kleinschreibung!";
        }
    }
    return "";
}
// Variablen definieren und Anfangswerte setzen.
$berufe = array("Angestellter", "Rentner", "Arbeiter", "Student/Schüler");
$vorname = $nachname = $geschl = $beruf = $hobbies = "";
$vorname_Err= $nachname_Err = $geschl_Err = $beruf_Err = $hobbies_Err = "";
$fehlerfrei = true;
// Formulardaten prüfen.
if('POST' == $_SERVER['REQUEST_METHOD']) {

    if(empty($_POST["geschl"])) {
        $geschl_Err = "Bitte Geschlecht auswählen!";
        $fehlerfrei = false;
    } else {
        $geschl= test_input($_POST["geschl"]);
    }

    if (empty($_POST["vorname"])) {
        $vorname_Err = "Vorname ist Pflichtfeld!";
    } else {
        $vorname = test_input($_POST["vorname"]);
        if (strlen($vorname) < 2) {
            $vorname_Err = "Vorname zu kurz!";
            // $fehlerfrei = false;
        } else {
```

```

        $muster = '/^([A-ZÄÖÜ]{1}[a-zäöü]*\.\.?)$|(([A-ZÄÖÜ]{1}\.\.?[a-zäöü]*\.\.?)( |\-)([A-ZÄÖÜ]{1}\.\.?[a-zäöü]*\.\.?))$/';
        $vorname_Err = validName($muster,$vorname);
    }
}
if (strlen($vorname_Err) > 0) {
    $fehlerfrei = false;
}

if (empty($_POST["nachname"])) {
    $nachname_Err = "Nachname ist Pflichtfeld";
} else {
    $nachname = test_input($_POST["nachname"]);
    if (strlen($nachname) < 2) {
        $nachname_Err = "Nachname zu kurz!";
    } else {
        $muster = '/^[A-Za-z-ÄÖÜäöü\s\-\'\.]+$/' ;
        $nachname_Err = validName($muster,$nachname);
    }
}
if (strlen($nachname_Err) > 0) {
    $fehlerfrei = false;
}
if (empty($_POST["beruf"])) {
    $beruf_Err = "Bitte einen Beruf auswählen!";
    $fehlerfrei = false;
} else {
    $beruf = test_input($_POST["beruf"]);
}

if (empty($_POST["hobbies"])) {
    $hobbies_Err = "Bitte mindestens 1 Hobby auswählen!";
    $fehlerfrei = false;
}
}
?>

```

Listing 4: Überprüfung der Formulare Daten

Bei der formalen Überprüfung der Textfelder 'vorname' und 'nachname' werden u. a. Muster in der PHP-Variablen *\$muster* gespeichert, um den Vergleich mit dem jeweils erfassten Namen durchzuführen.

5.1 Formulardaten sind fehlerhaft

Wenn bei der Validierung der Formulardaten ein Fehler gefunden wird,

- erfolgt eine Fehlermeldung rechts neben dem entsprechenden Formularfeld,
- erscheint unter dem Formular zusätzlich die Fehlermeldung „Die Formulardaten sind fehlerhaft! Bitte berichtigen.“ (siehe Abbildung 2).

PHP: Formulardaten nach dem Absenden erhalten

* Pflichtfeld.

Einzeilige Textfelder

Vorname ^{*)}: Ungültige Zeichen gefunden!

Nachname ^{*)}: Initiale in Kleinschreibung!

Auswahlfelder

Geschlecht ^{*)}: männlich weiblich

Drop-Down-Auswahlliste: Einzelauswahl

Beruf ^{*)}: Bitte einen Beruf auswählen!

Horizontale Auswahlkästchen: Mehrfachauswahl

Bitte Hobbies markieren ^{*)}: Bitte mindestens 1 Hobby auswählen!

lesen radfahren schwimmen nordic walk

Die Formulardaten sind fehlerhaft! Bitte berichtigen.

Abbildung 2: Meldungen bei fehlerhaften Formulardaten

Die Fehlermeldung „Ungültige Zeichen gefunden“ (siehe Abbildung 2) hinter dem einzeiligen Textfeld für *Vorname* wird angezeigt, weil die Initiale einen Kleinbuchstaben enthält.

5.2 Formulardaten sind fehlerfrei

Wenn bei der Validierung der Formulardaten *kein* Fehler gefunden wird, werden diese unterhalb des Formulars zwecks visueller Endkontrolle vollständig aufgelistet:

Ihre Eingaben:
Geschlecht: männlich
Vorname: Volker
Nachname: Thormählen
Beruf: Rentner
Gewählte Hobbies: lesen radfahren schwimmen nordic walk

Abbildung 3: Auflistung der fehlerfreien Formulardaten

Die erhalten gebliebenen Formulardaten können beliebig oft korrigiert, abgesendet, erneut überprüft und schließlich weiter verarbeitet werden.

6 Ergebnis der Validierung ausgeben

Das Ergebnis der Validierung des Formulars (siehe Abbildung 3) wird mit folgenden PHP-Codezeilen ausgegeben:

```
<?php
if(isset($_POST['submit'])){
    if (!$fehlerfrei) {
        echo "<center><b>Die Formulardaten sind fehlerhaft! Bitte berichtigen.</b></center>";
        exit();
    }
    echo "<h2>Ihre Eingaben:</h2>";
    echo "<pre>";
    echo "Geschlecht:<br>";
    switch ($geschl) {
        case "m":
            echo "<span>\t<b>männlich</b></span><br>";
            break;
        case "w":
            echo "<span>\t<b>weiblich</b></span><br>";
            break;
        default:
            echo "<span>\t<b>Weder männlich noch weiblich</b></span><br>";
    }

    echo "Vorname:<br>";
    echo "<span>\t<b>${vorname}</b></span><br>";

    echo "Nachname:<br>";
    echo "<span>\t<b>${nachname}</b></span><br>";

    echo "<span>Beruf:</span><br>";
    if(isset($_POST['beruf']) && (!empty($_POST['beruf']))) {
        $chosen = $_POST['beruf'] - 1; // nullbasierter Index
        echo "<span>\t<b>${berufe[$chosen]}</b></span>";
        echo "<br>";
    }

    echo "<span>Gewählte Hobbies:</span><br>";
    if(isset($_POST['hobbies']) && (!empty($_POST['hobbies']))) {
        foreach($_POST['hobbies'] as $key => $opt) {
            echo "<span>\t<b>".$opt."</b></span><br>";
        }
    }
    echo "<br><br>";
}
?>
```

Listing 5: Ergebnis der Validierung der Formulardaten ausgeben

Anhang

Listing der gesamten PHP-Datei mit dem Namen *Formulardaten_erhalten.php*:

```
<?php
// Berufe
function print_beruf_options($beruf,$berufe) {
    for ($i=0;$i<count($berufe);$i++) {
        printf("<option value=\"%d\"%s>%s</option>\n",
            ($i+1), ($beruf==($i+1) ? " selected" : ""),
            htmlentities($berufe[$i])
        );
    }
}
?>

<!DOCTYPE HTML>
<html lang="de">
    <head>
        <meta http-equiv="Content-Type" content="text/html;charset=utf-8" />
        <title>Formulardaten erhalten</title>
        <style>
            html, body {padding:0;border:0px none;}
            fieldset{width:auto;border:1px dashed black}
            fieldset div{display:inline-block;margin-top:10px;}
            legend {
                padding:0.2em 0.5em;
                border:1px solid black;
                color:black;
                background-color:yellow;
                font-size:90%;
                text-align:right;
            }
            form {width:800px;margin:0 auto;border:1px solid red;}
            label {
                float:left;
                width:250px;
                padding-left:20px;
                margin:5px;
                text-align:left;
                font-weight:bold;
            }
            span sup,.error{color:red;}
            br{clear:left;}
        </style>
    </head>
    <body>
    <?php
        function test_input($data) {
            $data = trim($data); // Überflüssige Zeichen am Anfang und Ende entfernen.
            $data = stripslashes($data); // Maskierungszeichen entfernen.
            $data = htmlspecialchars($data); // Sonderzeichen in HTML-Codes umwandeln.
            return $data;
        }
        function removeMultipleSpaces($Nm) { // doppelte Leerzeichen entfernen.
            return preg_replace("/\s{2,}/", ' ', trim($Nm));
        }
    </?php
    </body>
</html>
```

```

function validName($pattern,$Nm) { // Name auf Gültigkeit prüfen.
    $Nm = removeMultipleSpaces($Nm);
    if (strlen($Nm) < 2) {
        return "Vor-/Nachname zu kurz.";
    } else {
        if (!preg_match($pattern, $Nm)) { // Name mit Muster vergleichen.
            return "Ungültige Zeichen gefunden!";
        }
    }
    return "";
}

function validSurname($nachName) { // Initiale des Nachnamens prüfen.
    // Wenn nur 1 Wort im Nachnamen, dann ...
    if (str_word_count($nachName, 0, 'äüöÄÜÖß') == 1) {
        $firstChar = htmlentities($nachName[0]);
        // Wenn erster Buchstabe in Kleinschreibung, dann ...
        if($firstChar == strtolower($firstChar)) {
            return "Initiale in Kleinschreibung!";
        }
    }
    return "";
}

// Variablen definieren und Anfangswerte setzen.
$berufe = array("Angestellter", "Rentner", "Arbeiter", "Student/Schüler");
$vorname = $nachname = $geschl = $beruf = $hobbies = "";
$vorname_Err = $nachname_Err = $geschl_Err = $beruf_Err = $hobbies_Err = "";
$fehlerfrei = true;

// Formulardaten prüfen.
if('POST' == $_SERVER['REQUEST_METHOD']) {

    if(empty($_POST["geschl"])) {
        $geschl_Err = "Bitte Geschlecht auswählen!";
        $fehlerfrei = false;
    } else {
        $geschl = test_input($_POST["geschl"]);
    }

    if (empty($_POST["vorname"])) {
        $vorname_Err = "Vorname ist Pflichtfeld!";
    } else {
        $vorname = test_input($_POST["vorname"]);
        if (strlen($vorname) < 2) {
            $vorname_Err = "Vorname zu kurz!";
            // $fehlerfrei = false;
        } else {
            $muster = '/^([A-ZÄÖÜ]{1}[a-zäöü. ]*$)|((([A-ZÄÖÜ]{1})\.\?[a-zäöü. ]*)( |\-)([A-ZÄÖÜ]{1})\.\?[a-zäöü. ]*)$/';
            $vorname_Err = validName($muster,$vorname);
        }
    }

    if (strlen($vorname_Err) > 0) {
        $fehlerfrei = false;
    }
}

```

```

if (empty($_POST["nachname"])) {
    $nachname_Err = "Nachname ist Pflichtfeld";
} else {
    $nachname = test_input($_POST["nachname"]);
    if (strlen($nachname) < 2) {
        $nachname_Err = "Nachname zu kurz!";
    } else {
        $muster = "/^[A-Za-zÄÖÜäöü -.']+$/";
        $nachname_Err = validName($muster,$nachname);
        if (strlen($nachname_Err) == 0) {
            $nachname_Err = validSurname($nachname);
        }
    }
}
}
if (strlen($nachname_Err) > 0) {
    $fehlerfrei = false;
}

if (empty($_POST["beruf"])) {
    $beruf_Err = "Bitte einen Beruf auswählen!";
    $fehlerfrei = false;
} else {
    $beruf = test_input($_POST["beruf"]);
}

if (empty($_POST["hobbies"])) {
    $hobbies_Err = "Bitte mindestens 1 Hobby auswählen!";
    $fehlerfrei = false;
}
}
?>

```

```

<form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>"
<center>
  <h2>PHP: Formulare Daten nach dem Absenden erhalten</h2>
  <p><span class="error">* Pflichtfeld.</span></p>
</center>
<fieldset>
  <legend>Einzeilige Textfelder</legend>
  <label for="vorname">Vorname<span><sup>*</sup></span></label>
  <input name="vorname" size="25" maxlength="60" required="required"
    value="<?php
      printf("%s", htmlentities($vorname));
    ?>"
  <span class="error"><?php echo $vorname_Err;?></span>
  <br>
  <label for="nachname">Nachname<span><sup>*</sup></span></label>
  <input name="nachname" size="25" maxlength="60" required="required"
    value="<?php
      printf("%s", htmlentities($nachname));
    ?>"
  <span class="error"><?php echo $nachname_Err;?></span>
</fieldset><br>
<fieldset>
  <legend>Auswahlfelder</legend>
  <label for="geschl">Geschlecht<span><sup>*</sup></span></label>
  <input type="radio" name="geschl" value="m" required="required"
  <?php
    printf("%s", ($geschl=='m' ? " checked" : ""));
  ?> > männlich &nbsp;&nbsp;&nbsp;
  <input type="radio" name="geschl" value="w"
  <?php
    printf("%s", ($geschl=='w' ? " checked" : ""));
  ?> > weiblich
</fieldset><br>
<fieldset>
  <legend>Drop-Down-Auswahlliste: Einzelauswahl</legend>
  <label for="beruf">Beruf<span><sup>*</sup></span></label>
  <select name="beruf">
    <option value="0"
    <?php
      echo (!isset($beruf) ? " selected" : "");
    ?>
    >--- Bitte wählen ---</option>
    <?php
      print_beruf_options($beruf,$berufe);
    ?>
  </select>
  <span class="error"><?php echo $beruf_Err; ?></span>
</fieldset><br>
<fieldset>
  <legend>Horizontale Auswahlkästchen: Mehrfachauswahl</legend>
  <label for="hobbies">Bitte Hobbies markieren<span><sup>*</sup></span></label>
  <span class="error"><?php echo $hobbies_Err; ?></span>
  <div>
    <?php
      $hobbies= array('lesen' =>'lesen', 'radfahren' => 'radfahren' , 'schwimmen'
=> 'schwimmen', 'nordic walk'=>'nordic walk');
      foreach($hobbies as $key => $opt) {
        if (isset($_POST['hobbies']) && in_array($key, $_POST['hobbies'])) {

```

```
        $checked = ' checked="checked"';
    } else {
        $checked = '';
    }
    printf('<input type="checkbox" name="hobbies[]" value="%s" %s>%s',
        $key,
        $checked,
        $opt
    );
    }
    ?>
</div>
</fieldset><br>
<label for="submit">&nbsp;</label>
<input type="submit" name="submit" value="Formular absenden" />
<br>
</form>
</body>
</html>
```

```

<?php
if(isset($_POST['submit'])){
    if (!$fehlerfrei) {
        echo "<center><b>Die Formulardaten sind fehlerhaft! Bitte berichtigen.</b></center>";
        exit();
    }
    echo "<h2>Ihre Eingaben:</h2>";
    echo "<pre>";
    echo "Geschlecht:<br>";
    switch ($geschl) {
        case "m":
            echo "<span>\t<b>männlich</b></span><br>";
            break;
        case "w":
            echo "<span>\t<b>weiblich</b></span><br>";
            break;
        default:
            echo "<span>\t<b>Weder männlich noch weiblich</b></span><br>";
    }

    echo "Vorname:<br>";
    echo "<span>\t<b>${vorname}</b></span><br>";

    echo "Nachname:<br>";
    echo "<span>\t<b>${nachname}</b></span><br>";

    echo "<span>Beruf:</span><br>";
    if(isset($_POST['beruf']) && (!empty($_POST['beruf']))) {
        $chosen = $_POST['beruf'] - 1; // nullbasierter Index
        echo "<span>\t<b>${berufe[$chosen]}</b></span>";
        echo "<br>";
    }

    echo "<span>Gewählte Hobbies:</span><br>";
    if(isset($_POST['hobbies']) && (!empty($_POST['hobbies']))) {
        foreach($_POST['hobbies'] as $key => $opt) {
            echo "<span>\t<b>".$opt."</b></span><br>";
        }
    }
    echo "<br><br>";
}
?>

```

Ende der PHP-Datei

Quellen

- [1] o. V., „Kontaktformular,“ 2017. [Online]. Available: <http://www.html-info.eu/php/php-scripting-fuer-den-anfang/item/kontaktformular.html>. [Zugriff am 05 08 2018].
- [2] o. V., „PHP/Komplexe Formulare,“ 19 01 2009. [Online]. Available: http://mikiwiki.org/wiki/PHP/Komplexe_Formulare. [Zugriff am 05 08 2018].
- [3] C. Reeg, „Dynamische Formulare,“ [Online]. Available: <http://lueersen.homedns.org/!MySQL/DSP/node13.html#SECTION05230000000000000000>. [Zugriff am 05 08 2018].