

PHP-Grundlagen: Einfache Formularauswertung Teil 1

Gegeben ist folgendes Formular:

```
<form action = "index.php" method = "post">

    <label>Vorname: </label> </br>
    <input type = "text" name = "vorname" /> </br></br>

    <label>Nachname:</label> </br>
    <input type = "text" name = "nachname" /> </br></br>

    <input type = "submit" />

</form>
```

Vorname:

Nachname:

Senden

Action und Method

Das **form**-Element enthält zwei wichtige Attribute:

- **action:** Das Attribut action verweist auf eine PHP-Datei mit dem Auswertungsprogramm für die Eingaben. In diesem Beispiel findet die Verarbeitung der Eingaben direkt auf der index-Seite statt.
- **method:** Verweist auf die Übermittlungsmethode zum Webserver

Eine wichtige Rolle spielt der **submit**-Button. Beim Betätigen wird die PHP-Datei geladen, die beim Attribut **action** angegeben wurde. Hier ist dann ein Programm hinterlegt, das sich um die Verarbeitung der Eingaben kümmert. Die Übermittlung der Daten erfolgt mit Hilfe der Übermittlungsmethode **post**. Diese Methode hält eine besondere Variable parat.

Auswertung mit `$_POST`

Es gibt in PHP eine vordefinierte Variable namens `$_POST`. Diese Variable nimmt die Werte der Eingabefelder des Formulars auf, wenn die Methode **post** gewählt wurde¹. Aus den Namensbezeichnungen (`name = ...`) der input-Tags werden in `$_POST` sogenannte assoziative Felder angelegt.

Ein assoziatives Feld ist ein Array, wobei die Felder jedoch nicht über eine fortlaufende Indexnummer, sondern über eine Schlüsselbezeichnung identifiziert werden.

Beispiel

Vorname:

Nachname:

Das Formular wird ausgefüllt und gesendet. Der Inhalt der Variablen `$_POST` sieht dann wie folgt aus:

```
array(2){ ["vorname"] => "Hans", ["nachname"] => "Winter"}
```

Der Zugriff auf das Array erfolgt über die Namensbezeichnungen:

```
echo "<p>" . $_POST["vorname"] . " " . $_POST["nachname"] . "</p>";
```

Hier werden Vor- und Nachname einfach mittels **echo** ausgegeben.

Oft werden die Werte aus der `$_POST`-Variablen in neue Variablen abgelegt.

```
$firstname = $_POST["vorname"];  
$lastname = $_POST["nachname"];  
  
echo "<p>" . $firstname . " " . $lastname . "</p>";
```

¹ Neben der Methode **post** gibt es noch die Methode **get**. Diese werden wir später noch kennenlernen.

Check, ob die Werte wirklich existieren

Bevor auf ein Array-Feld der `$_POST`-Variable zugegriffen werden kann, muss dieses natürlich existieren und einen Wert besitzen. Ansonsten gibt es eine Fehlermeldung. PHP bietet hierfür eine Check-Funktion namens ***isset()*** an. Mit dieser Funktion kann geprüft werden, ob ein Feld innerhalb von `$_POST` überhaupt einen Wert besitzt. Die Überprüfung erfolgt mit Hilfe einer ***if***-Anweisung.

Beispiel

```
<?php
    if(isset($_POST["vorname"]) && isset($_POST["nachname"]))
    {
        $firstname = $_POST["vorname"];
        $lastname = $_POST["nachname"];

        echo "<p>" . $firstname . " " . $lastname;
    }
?>
```