# L1\_4.1 Serverseitiger Datenauswertung in PHP

Mit HTML-Formularen können wir Informationen vom Besucher der Seite erfragen. Bislang wurden diese Informationen von externen Diensten wie Goolge Charts oder der Google-Suchmaschine ausgewertet. Wir wollen die vom Benutzer eingegebenen Informationen nun selbst mir einen PHP-Skript auswerten.

Gegeben ist folgendes HTLM-Formular:

|  |  |
| --- | --- |
| […]  <form action="begruessung.php">      Ihr Vorname:      <input type="text" name="vname">      Ihr Nachname:      <input type="text" name="nname">      <input type="submit" value="Senden">  </form>  […] |  |

Nehmen wir an, der Anwender hat das Formular ausgefüllt und abgesendet. Die Daten werden an das im action-Attribut angegebene Zieldokument gesendet. Es stellt sich nun die Frage, wie in der PHP-Seite (Zieldokument, hier „begruessung.php“) auf diese Eingaben des Anwenders zugegriffen werden kann.

Der Mechanismus ist recht einfach. Für jedes Eingabefeld steht in der PHP-Seite eine Variable zur Verfügung. Der Name der Variablen lautet **$\_GET**. Dabei handelt es sich um eine komplexe Variable, die nur bedingt mit den uns bekannten Variablen vergleichbar ist. Man spricht hier von einem assoziativen Array oder einer Hashmap. Die Struktur der Variable muss noch mit einer speziellen Syntax ergänzt werden:

$\_GET["*name*"]

wobei der Name in der Regel eine Zeichenkette ist und bei uns abhängig vom Wert des HTML-Attributs name im zugehörigen HTML-Formular ist.

…

   <input type="text" name="vname">

…

   <input type="text" name="nname">

…

Ausgehend vom oben dargestellten Formular stehen nun innerhalb der PHP-Seite (Zieldokument) folgende Variablen zur Verfügung:

$\_GET["vname"] und $\_GET["nname"]

[...]

<input type="text" **name="vname"**>

[...]

<input type="text" **name="nname"**>

[...]

Somit lässt sich eine persönliche Begrüßung des Anwenders wie folgt realisieren (Datei „begruessung.php“):

<!DOCTYPE html>

<html>

<head>

    <title>PHP - Auswertung von Formulardaten</title>

    <meta charset="utf-8">

</head>

<body>

    <?php

    echo "Hallo ";

    echo **$\_GET['vname']**;

    echo ", ";

    echo **$\_GET['nname']**;

    ?>

</body>

</html>

Achtung: Wir haben kennengelernt, dass Variablen innerhalb von doppelten Anführungszeichen durch ihren Wert ersetzt werden. Dies gilt nur für „einfache Variablen“, nicht aber für komplexe Variablen, wie es assoziative Arrays sind. Die Ausgabe wurde im obigen Beispiel daher auf mehrere echo-Anweisungen aufgeteilt. Alternativ könnte man auch mit dem Punkt-Operator arbeiten können.

Die Handhabung von assoziativen Arrays ist etwas unhandlich, speziell, wenn es um Ausgaben mittels der „echo“-Anweisung geht. Daher ist es praktikabel, die Werte aus dem assoziativen Array zu Beginn des PHP-Skriptes in „einfache Variablen“ zu übertragen und im weiteren Verlauf mit diesen einfachen Variablen zu arbeiten. Folgendes Beispiel zeigt das Vorgehen.

<!DOCTYPE html>

<html>

<head>

    <title>PHP - Auswertung von Formulardaten</title>

    <meta charset="utf-8">

</head>

<body>

    <?php

**$vn = $\_GET['vname'];**

**$nn = $\_GET['nname'];**

    echo "Hallo $vn, $nn"

    ?>

</body>

</html>

Werden die Daten im Formular über die Methode „post“ übertragen, dann ändert sich lediglich der Name des assoziativen Arrays in $\_POST.

Beispiel:

<!DOCTYPE html>

<html>

<head>

    <title>PHP - Auswertung von Formulardaten</title>

    <meta charset="utf-8">

</head>

<body>

    <?php

**$vn = $\_POST['vname'];**

**$nn = $\_POST['nname'];**

    echo "Hallo $vn, $nn"

    ?>

</body>

</html>