

# Daten an eine andere Datei weitergeben mit \$ POST

## Inhalt:

Übung 1: addieren

Übung 2: Mehrwertsteuer errechnen (inkl. CSS)

Übung 3: Hotelbuchung mit Radio-Buttons

## Übung 1: addieren

Erstelle ein „zahl.html“ und ein „zahl.php“. In das Formular der Zahl-HTML Datei sollen zwei Felder eingegeben werden können, die jeweils eine Zahl von 1-1000 beinhaltet.

Als Ergebnis in der PHP-Datei soll die Summe der beiden Zahlen ausgegeben werden.

### zahl.php

Erstelle eine neue Variable mit dem Namen **\$zahl**

Diese soll den Inhalt von der Formulareingabe aufnehmen durch **\$zahl1 = \$ POST[...]**

Die Variable \$ergebnis nimmt die Addition vor **\$ergebnis = \$zahl1 + zahl2**

Bitte jeweils eine Zahl von 1 bis 1000 eingeben.

Zahl 1:

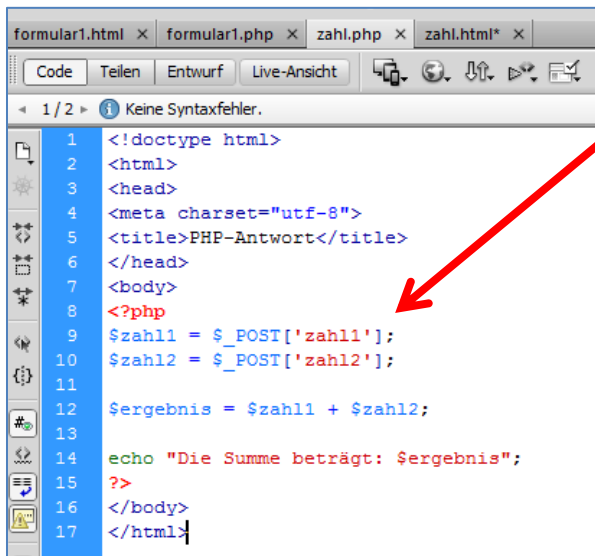
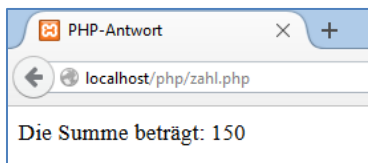
Zahl 2:

Es wird die Summe der beiden Zahlen ausgegeben.

```
er_format.php x zahl.html x zahl.php x
de Teilen Entwurf Live-Ansicht Titl: Zahlen-Formular
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>Zahlen-Formular</title>
6 </head>
7 <body>
8 <h1>Bitte jeweils eine Zahl von 1 bis 1000 eingeben.</h1>
9 <form action = "zahl.php" method = "post" >
10 <p>Zahl 1: <input name = "zahl1" type="text" > </p>
11 <p>Zahl 2: <input name = "zahl2" type="text" > </p>
12 <br>
13 <p>Es wird die Summe der beiden Zahlen ausgegeben.</p>
14 <input type="submit" value="Abschicken">
15 <input type="reset" value="Löschen">
16 </form>
17 </body>
18 </html>
19
```

inkl. eines Buttons  
„Löschen“  
type="reset"

Ergebnis:

A screenshot of a code editor with several tabs: "formular1.html", "formular1.php", "zahl.php", and "zahl.html\*". The "zahl.php" tab is active, showing PHP code. A red arrow points from the code to the browser screenshot above. The code is as follows:

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>PHP-Antwort</title>
6 </head>
7 <body>
8 <?php
9 $zahl1 = $_POST['zahl1'];
10 $zahl2 = $_POST['zahl2'];
11
12 $ergebnis = $zahl1 + $zahl2;
13
14 echo "Die Summe beträgt: $ergebnis";
15 ?>
16 </body>
17 </html>
```

### **Aufgabe:**

Erstelle ein HTML-Formular mit Vorname, Nachname, Straße und Nummer, Postleitzahl, Ort.  
Die dazu passende PHP-Datei soll das unten abgebildete Ergebnis ausgeben:

Das Ergebnis soll ca. so aussehen:

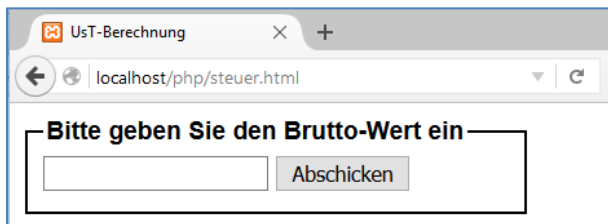
Dennis Darrer  
Brennerweg 8  
2130 Mistelbach

## Übung 2: Mehrwertsteuer errechnen

Entwickle sowohl ein HTML-Formular (steuer.html) für die Eingabe eines beliebigen Bruttowertes (nur Ganzzahl, kein Komma) und ein PHP-Auswertungsskript (steuer.php), welches aus dem Formular den Bruttowert übernimmt und daraus den Netto- und den MwSt-Betrag errechnet. Verwende dabei die Funktion `number_format()`. Im PHP-Skript soll es einen Link geben, um einen weiteren Wert eingeben und berechnen zu lassen.

steuer.html: zur Aufhübschung soll etwas CSS und ein `<fieldset>` mit einer `<legend>` verwendet werden.

Ziel:



The screenshot shows a web browser window with the title 'UsT-Berechnung' and the URL 'localhost/php/steuer.html'. The page content is a form with a legend that says 'Bitte geben Sie den Brutto-Wert ein'. Below the legend is a text input field and a submit button labeled 'Abschicken'.

Code:

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>UsT-Berechnung</title>
6 <style type="text/css">
7 body {font-family:Arial, Helvetica, sans-serif}
8 fieldset {
9     width: 300px;
10    border: 2px solid black;
11 }
12 legend {font-weight:bold;}
13 </style>
14 </head>
15
16 <body>
17 <form method="post" action="steuer.php">
18 <fieldset>
19     <legend>Bitte geben Sie den Brutto-Wert ein</legend>
20     <input type="text" name="brutto">
21     <input type="submit" value="Abschicken">
22 </fieldset>
23 </body>
24 </html>
25
```

### Die Auswertung mit „steuer.php“:

Der Nettowert lässt sich einfach mit  $\text{Brutto}/120 * 100$  berechnen. Zieht man den Netto vom Brutto ab, erhält man den Mehrwertsteuersatz.

Ziel ohne Formatierung:



Code dafür:

```
8 <body>
9 <?php
10 $mwst = 20;
11 $brutto = $_POST["brutto"];
12 $netto = $brutto / (100+$mwst)*100; //Netto berechnen
13 $mwstbetrag = $brutto-$netto; //USt berechnen
14
15 //Die Werte für die Ausgabe formatieren
16 $brutto = number_format($brutto,2,"",".");
17 $netto = number_format($netto,2,"",".");
18 $mwstbetrag = number_format($mwstbetrag,2,"",".");
19 ?>
20 <table>
21 <caption>Brutto und Netto</caption>
22 <tr>
23 <th>Netto</th>
24 <td><?php echo $netto ?> €</td>
25 </tr>
26 <tr>
27 <th>MwSt.</th>
28 <td><?php echo $mwstbetrag ?> €</td>
29 </tr>
30 <tr>
31 <th>Brutto</th>
32 <td><?php echo $brutto ?> €</td>
33 </tr>
34 </table>
35 <br>
36 <div><a href="steuer.html">Bitte neuen Wert eingeben</a></div>
37
38 </body>
```

Ziel mit Formatierung per CSS:



CSS-Code dafür:

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>USt-Berechnung</title>
6 <style type="text/css">
7 body {font-family:Arial;}
8 th {text-align:left;}
9 td {text-align:right;}
10 td, th {padding:5px;}
11 table, td, th {
12     border:1px solid black;
13     border-collapse:collapse;
14 }
15 table {width: 200px;}
16 caption {
17     background-color:black;
18     color:white;
19     font-weight:bold;
20     font-size:1.2em;
21     padding:5px;
22 }
23 div {
24     font-size:0.8em;
25     margin:5px;
26 }
27 </style>
28 </head>
```

```
<style type="text/css">
body {font-family:Arial;}
th {text-align:left;}
td {text-align:right;}
td, th {padding:5px;}
table, td, th {
    border:1px solid black;
    border-collapse:collapse;
}
table {width: 200px;}
caption {
    background-color:black;
    color:white;
    font-weight:bold;
    font-size:1.2em;
    padding:5px;
}
div {
    font-size:0.8em;
    margin:5px;
}
</style>
```

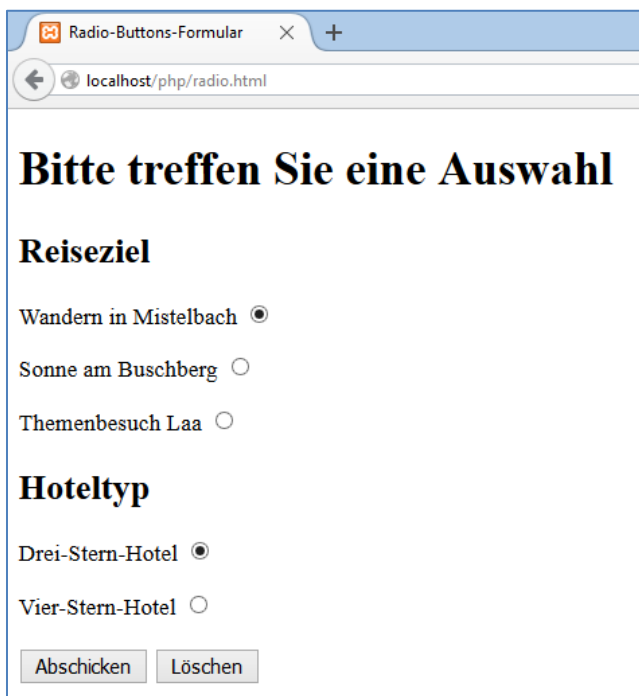
### 3)Hotelbuchung

#### Formular mit Auswertung von Radio-Buttons

Auswahlelemente erleichtern dem Benutzer die Bedienung. Dadurch verringert sich gleichzeitig die Möglichkeit, Fehler bei der Eingabe zu machen.

Innerhalb der ersten Gruppe kann der Besucher ein Reiseziel auswählen, innerhalb der zweiten Gruppe einen Hoteltyp.

Erstelle folgendes HTML-Formular:



The screenshot shows a web browser window with the title 'Radio-Buttons-Formular'. The address bar shows 'localhost/php/radio.html'. The main content area has a heading 'Bitte treffen Sie eine Auswahl'. Below this, there are two sections: 'Reiseziel' and 'Hoteltyp'. Under 'Reiseziel', there are three radio buttons: 'Wandern in Mistelbach' (checked), 'Sonne am Buschberg', and 'Themenbesuch Laa'. Under 'Hoteltyp', there are two radio buttons: 'Drei-Stern-Hotel' (checked) and 'Vier-Stern-Hotel'. At the bottom of the form, there are two buttons: 'Abschicken' and 'Löschen'.

```
<body>
<h1>Bitte treffen Sie eine Auswahl</h1>
<form action = "radio.php" method = "post" >
<h2>Reiseziel</h2>
    <p>Wandern in Mistelbach <input type="radio" name="ausflug" value="mistelbach"
checked="checked">
    <p>Sonne am Buschberg <input type="radio" name="ausflug" value="buschberg">
    <p>Themenbesuch Laa <input type="radio" name="ausflug" value="laa">
<h2>Hoteltyp</h2>
    <p>Drei-Stern-Hotel <input type="radio" name="hotel" value="drei" checked="checked">
    <p>Vier-Stern-Hotel <input type="radio" name="hotel" value="vier">
<br><br>
<input type="submit" value="Abschicken">
<input type="reset" value="Löschen">
</form>
</body>
</html>
```

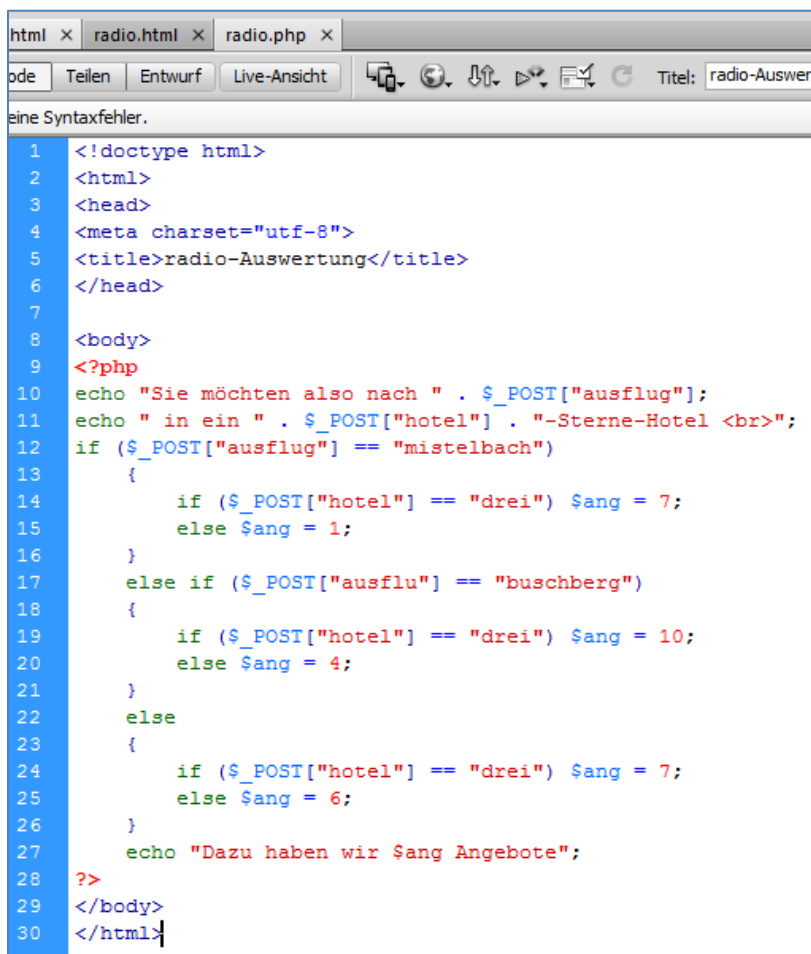
**Erklärung:** In diesem Formular werden zwei Gruppen von Radiobuttons dargestellt. Die Elemente einer Gruppe haben den gleichen Namen, dadurch wird eine logische Zusammengehörigkeit hergestellt. Optisch werden die beiden Gruppen durch Überschriften voneinander getrennt.

### Erstelle die Auswertungs-PHP „radio.php“:

Zuerst ein Einleitungstext „Sie möchten also folgenden Ausflug tätigen: .... . Wohnen werden Sie in einem ....Sternen-Hotel.“

Dann soll die Anzahl von Möglichkeiten an Hotels ausgegeben werden. Z.B. „Für diese Auswahl haben wir 3 Angebote.“

Aus der Information in den Variablen wird im PHP-Programm mit Hilfe einer geschachtelten Verzweigung die Anzahl der vorliegenden Angebote ermittelt und in die Variable „\$sang“ gespeichert. Der Wert dieser Variable wird zusammen mit einer Bestätigung der Eingabedaten zurückgesendet.



```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>radio-Auswertung</title>
6 </head>
7
8 <body>
9 <?php
10 echo "Sie möchten also nach " . $_POST["ausflug"];
11 echo " in ein " . $_POST["hotel"] . "-Sterne-Hotel <br>";
12 if ($_POST["ausflug"] == "mistelbach")
13 {
14     if ($_POST["hotel"] == "drei") $sang = 7;
15     else $sang = 1;
16 }
17 else if ($_POST["ausflu"] == "buschberg")
18 {
19     if ($_POST["hotel"] == "drei") $sang = 10;
20     else $sang = 4;
21 }
22 else
23 {
24     if ($_POST["hotel"] == "drei") $sang = 7;
25     else $sang = 6;
26 }
27 echo "Dazu haben wir $sang Angebote";
28 ?>
29 </body>
30 </html>
```

Welche Angebote gibt es?

Mistelbach:

- 3 Stern: 7
- 4 Stern: 1

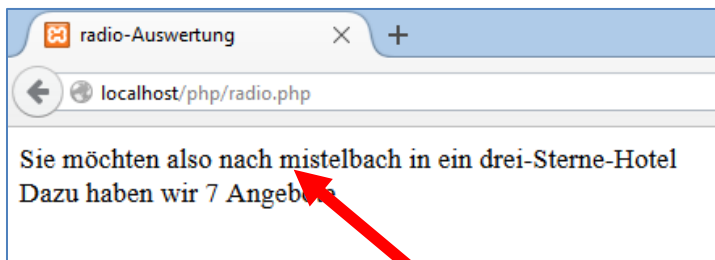
Buschberg:

- 3 Stern: 10
- 4 Stern: 4

Laa:

- 3 Stern: 7
- 4 Stern: 6

Ergebnis:



### Erklärung:

Der gemeinsame Name (Eigenschaft „name“) der ersten Optionsgruppe ist „ausflug“. Nach Absenden des Formulars steht dadurch die Variable „\$\_POST[„ausflug“]“ mit dem Wert (value) des vom Benutzer ausgewählten Eintrags im PHP-Programm zur Verfügung. Falls er z.B. „Sonne am Buschberg“ auswählt, wird \$\_POST[„ausflug“] der Wert „buschberg“ zugewiesen.

Der gemeinsame Name der zweiten Optionsgruppe ist „hotel“. Falls der Benutzer z.B. „Drei-Sterne-Hotel“ auswählt, wird der Variablen \$\_POST[„hotel“] der Wert „drei“ zugewiesen.

Aus der Information in den Variablen wird im PHP-Programm mit Hilfe einer geschachtelten Verzweigung die Anzahl der vorliegenden Angebote ermittelt und in die Variable „\$ang“ gespeichert. Der Wert dieser Variable wird zusammen mit einer Bestätigung der Eingabedaten zurückgesendet.

**Aufgabe:** Was ist zu tun um 2 Bereiche noch zu korrigieren:

- Der Ortsname sollte großgeschrieben werden.
- Die Bezeichnung „drei“ und „vier“ bei den Hotels sollte großgeschrieben werden.

Lösung: Verbesserung, da die Ausgabe in Kleinbuchstaben

Da der „value“ aus der HTML-Datei ausgelesen wird, kann man den korrekt schreiben: z.B. Mistelbach.

Daher:

```
8 <h2>Reiseziel</h2>
9 <input type="radio" name="ausflug" value="Mistelbach" checked="checked"
  id="mistelbach">
10 <label for="mistelbach">Wandern in Mistelbach</label>
11 <br>
```



## FILES senden

Sofern man per Formular eine Datei an den Server schicken will, wird das superglobale Array `$_FILES` angelegt. Darin enthalten sind Informationen zur hochgeladenen Datei; das Array hat eine festgelegte Struktur.

Im Einzelnen umfasst das mehrdimensionale Array die Felder:

- `Name` der Name der Datei, so wie sie auf dem Rechner des Clients benannt war
- `Type` der MIME-Typ
- `tmp_name` Beim Hochladen speichert PHP die Datei lokal auf dem Server und vergibt einen neuen, eindeutigen Namen, der hierin gespeichert wird.
- `Error` ein Fehlercode
- `Size` die Größe der Datei in Bytes

Um eine Datei hochzuladen, bedarf es im Formular einiger Anpassungen:

```
<form action="" enctype="multipart/form-data" method="post">
  <input type="file" name="uploadFile" />
  <input type="submit" name="doUpload" value="Speichern" />
</form>
```

Neu ist die Angabe eines Codierungstyps im Kopf des Formulars. Das Attribut „enctype“ der Markierung „form“ dient als Codierungsangabe für die Formulardaten. Beim Hochladen von Dateien ist hier die Angabe „multipart/form.data“ notwendig.

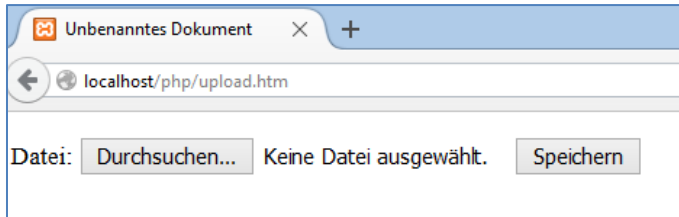
Ebenso wird der Typ des Eingabefeldes von text auf file gesetzt. Bei der Anzeige des Formulars im Browser wird dadurch ein »Durchsuchen«-Knopf erstellt, mit dem sich eine Datei auf der lokalen Festplatte zum Hochladen auswählen lässt. Unglücklicherweise lässt sich diese Schaltfläche nicht wie gewohnt durch Cascading Style Sheets (CSS) formatieren. Der hier angegebene Name „uploadFile“ kann frei gewählt werden.

Erstelle die „upload.html“



```
upload.htm x
Code  Teilen  Entwurf  Live-Ansicht  [Icons]  Titel: Unbenanntes Dokument
1  <!doctype html>
2  <html>
3  <head>
4  <meta charset="utf-8">
5  <title>Unbenanntes Dokument</title>
6  </head>
7
8  <body>
9  <form action="upload.php" enctype="multipart/form-data" method="post">
10 <p>Datei: <input type="file" name="uploadFile" />
11 <input type="submit" name="doUpload" value="Speichern" />
12 </form>
13
14 </body>
15 </html>
```

Ergebnis:



Nach dem Senden werden automatisch Elemente für das Feld `$_FILES` erzeugt. Der Inhalt des Arrays `$_FILES` sieht beispielhaft wie folgt aus:

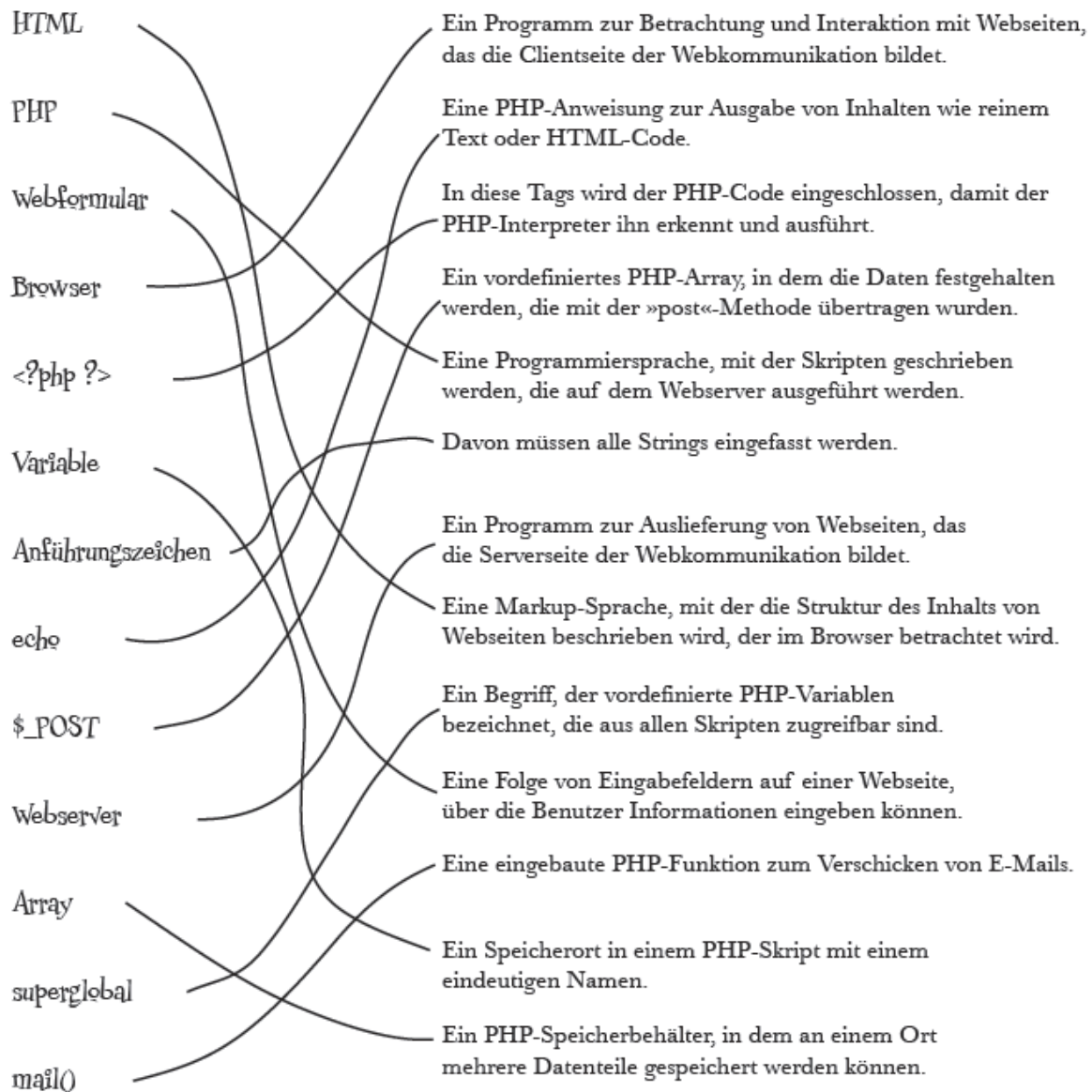
Array

```
(
    [uploadFile] => Array
        (
            [name] => datei.txt (Originaldateiname)
            [type] => text/plain (Dateityp)
            [tmp_name] => C:\xampp\tmp\php102.tmp (der temporäre Dateiname
auf dem Server)
            [error] => 0
            [size] => 26 (Dateigröße)
        )
)
```

Die temporäre Datei auf dem Server lässt sich dann mit den Datei- und Verzeichnisfunktionen in PHP verarbeiten.

HTML	Ein Programm zur Betrachtung und Interaktion mit Webseiten, das die Clientseite der Webkommunikation bildet.
PHP	Eine PHP-Anweisung zur Ausgabe von Inhalten wie reinem Text oder HTML-Code.
Webformular	In diese Tags wird der PHP-Code eingeschlossen, damit der PHP-Interpreter ihn erkennt und ausführt.
Browser	Ein vordefiniertes PHP-Array, in dem die Daten festgehalten werden, die mit der »post«-Methode übertragen wurden.
<?php ?>	Eine Programmiersprache, mit der Skripten geschrieben werden, die auf dem Webserver ausgeführt werden.
Variable	Davon müssen alle Strings eingefasst werden.
Anführungszeichen	Ein Programm zur Auslieferung von Webseiten, das die Serverseite der Webkommunikation bildet.
echo	Eine Markup-Sprache, mit der die Struktur des Inhalts von Webseiten beschrieben wird, der im Browser betrachtet wird.
\$_POST	Ein Begriff, der vordefinierte PHP-Variablen bezeichnet, die aus allen Skripten zugreifbar sind.
Webserver	Eine Folge von Eingabefeldern auf einer Webseite, über die Benutzer Informationen eingeben können.
Array	Eine eingebaute PHP-Funktion zum Verschicken von E-Mails.
superglobal	Ein Speicherort in einem PHP-Skript mit einem eindeutigen Namen.
mail()	Ein PHP-Speicherbehälter, in dem an einem Ort mehrere Datenteile gespeichert werden können.

## Lösung:



## Quellen:

Lynn Beighley und Michael Morrison in: PHP & MySQL von Kopf bis Fuß;  
Verlag =O'Reilly, 2009

Giesbert Damaschke in: PHP & MySQL, Der Web-Baukasten für Einsteiger  
und Individualisten; Verlag Wiley-vch, 2015