

Formulare in HTML

Aus Formularen lassen sich Daten an PHP-Scripte übertragen.

Formulare stehen zwischen dem Form-Tag: `<FORM. . . > . . . </FORM>`. Im `<FORM>`-TAG stehen wichtige Attribute vor den eigentlichen Formularelementen:

Attribut	Beschreibung
method	legt die Methode (get oder post) der Datenübertragung fest.
action	legt das Script fest, welches nach dem Abschicken des Formulars die Daten erhält, oder die Folgeseite, welche aufgerufen werden soll.
name, target	Formularname, und Zielframe (sind hier nebensächlich)

Eingabe-Elemente

In der Regel besitzt eine angezeigte HTML-Seite ein Formular, innerhalb dessen die im Folgenden beschriebenen Elemente Eingaben zulassen. Die Eingaben der Elemente eines Formulars werden gemeinsam durch Betätigen einer speziellen Schaltfläche an das im Form-Tag genannte Script gesendet.

<INPUT>-Elemente

Mit `<INPUT>`-Tags lassen sich unterschiedliche Elemente darstellen, die durch das Attribut `type` definiert werden:

Typ	Beschreibung
checkbox	Kontrollkästchen
radio	Optionsfeld (Radiobutton)
text	Eingabefeld (Editfeld)
password	Eingabefeld, in dem Zeichen als Sternchen dargestellt werden
hidden	Unsichtbare Felder können zur Datenübertragung an Scripte oder Folgeseiten benutzt werden.
reset	Button zum Löschen aller Formulareingaben
submit	Button zur Datenübertragung, bzw. Senden der Eingaben der Elemente

Neben `type` besitzt das `<INPUT>`-Tag `name`, und `value` als Attribute. `name` wird später zum Auslesen der Daten gebraucht. Mit `value` lassen sich die Elemente vorbelegen.

Eingabefelder

```
<FORM action="antwort.php" method="get">
  Daten:
  <INPUT type="text" name="Artikelgruppe"
    value="Vorgabe" size="50" maxlength="50">
  <br><br>
  <INPUT type="submit" value="absenden">
</FORM>
```

Daten:

Hier wird ein Formular mit einem Textfeld und einem Button zum Abschicken der Daten erzeugt. `value`, `size` und `maxlength` sind optional. `value` legt im Eingabefeld die Vorgabe und beim `submit`-Button die Beschriftung fest.

Ohne `value`-Angabe lautet die Beschriftung des Buttons: "Anfrage senden".

Kontrollkästchen (Checkbox)

```
<FORM action="antwort.php" method="get">
  <INPUT type="checkbox" name="Drucken" value="ja">
  Daten drucken <br><br>
  <INPUT type="submit">
</FORM>
```

Daten drucken

Anfrage abschicken

Erzeugt ein Kontrollkästchen. Beim Abschicken wird der mit `value` festgelegte Wert "ja" übergeben, wenn die Checkbox angewählt wurde.

Optionsfelder, Optionsfeldgruppen (Radiobuttons und Radiogroups)

```
<FORM action="antwort.php" method="get">
  <INPUT type="radio" name="Zahlung" value="Kreditkarte">
  Kreditkarte <br>
  <INPUT type="radio" name="Zahlung" value="Rechnung" checked>
  Rechnung <br>
  <INPUT type="radio" name="Zahlung" value="Bankeinzug">
  Bankeinzug <br>
  <INPUT type="submit">
</FORM>
```

Innerhalb einer Optionsfeldgruppe kann immer nur ein Button aktiv sein. Hier wird eine Gruppe mit Namen Zahlung, 3 Radiobuttons, Beschriftung und "Sende"-Button erzeugt. Das zweite Feld ist mit `checked` aktiviert. Beim Absenden wird der mit `value` festgelegte Wert des aktiven Buttons übertragen.

Kreditkarte

Rechnung

Bankeinzug

Anfrage abschicken

Listen, Dropdownlisten (ListBox)

Werden mit `<SELECT>`-Tag erzeugt. Das Attribut `size` legt die Anzahl der Einträge fest. `size=0` und `size=1` führen zu einer Dropdownliste. Alle anderen Werte zu einer normalen Liste.

```
<FORM action="antwort.php" method="get">
  <SELECT name="Artikelgruppe" size=3>
    <option>Keramik
    <option>Glas
    <option>Holz
  </SELECT>
  <INPUT type="submit">
</FORM>
```

Die hinter `<option>` stehenden Bezeichnungen stellen die Einträge in der Liste dar und sind gleichzeitig die bei der Auswahl später übertragenen Werte. Sollen mehrere Listenpunkte angewählt werden können, muss das Attribut `multiple` im `<SELECT>`-Tag angegeben werden.

Falls sich Options-Werte von der Beschriftung unterscheiden sollen, ist auch folgende Schreibweise möglich: `<option value=3>Holz`. Hier wird der Wert 3 übertragen und Holz in der Liste angezeigt.

Mehrzeilige Textfelder

Sie werden mit dem `<TEXTAREA>`-Tag definiert und hier nicht weiter behandelt. Details können HTML-Tutorials entnommen werden.

Formulare - Daten von HTML-Seiten	erhalten am:	Seite:
C++		

Datenübertragung an PHP-Scripte

Beim Abschicken von HTML-Formularen werden die Daten der einzelnen Formularelemente an das im `<FORM>`-Tag eingestellte Script übergeben. Es gibt mehrere Möglichkeiten diese Daten im Script anzusprechen. Im Rahmen dieser kurzen Einführung soll nur auf die Methode mit den beiden autoglobalen Arrays `$_GET[]` und `$_POST[]` eingegangen werden.

Daten senden

Das folgende Formular ...

```

<FORM action="antwort.php"
method="get">
  Daten:
  <INPUT type="text" name="Daten1" ><br>
  <INPUT type="text" name="Daten2" ><br>
  <INPUT type="submit" value="absenden">
</FORM>
```

erzeugt beim Absenden den Aufruf:

`http://localhost/antwort.php?Daten1="Eingabe1"&Daten2 ="Eingabe2"`

wenn in den Feldern Eingabe1 und Eingabe2 steht.

Die Namen der Formularelemente und ihre Werte werden durch & getrennt in einer Liste hinter die Adresse angehängt.

Innerhalb des PHP-Scripts `antwort.php` kann auf diese Werte mit dem `$_GET`-Array zugegriffen werden, der ein assoziatives Array ist. Die einzelnen Elemente dieses Arrays lassen sich über die Namen der Formularelemente ansprechen.

Bsp.:

```
$str1=$_GET["Daten1"];
```

```
$str2=$_GET["Daten2"];
```

Die Methode `get` eignet sich für die Übertragung kleiner Datenmengen. Bei größeren Mengen sollte im Formular die Methode `post` gewählt werden. Die Daten werden dann als HTML-Nachricht verschickt, welche während der Übertragung sicherer gegen Mitlesen ist.

Aus die so übertragenen Daten kann im Script mit dem Array `$_POST[]` in gleicher Weise zugegriffen werden.

Versteckte Felder zur Datenweiterleitung nutzen

Problem: Ein Anwender gibt Daten auf einer HTML-Seite `a.html` ein, welche an ein Script `a.php` gesendet werden. Dieses Script erzeugt als Antwort eine dynamische HTML-Seite `b.html`, in welche der Anwender weitere Daten eingeben kann. Diese Eingaben werden an ein anderes Script `b.php` zur Auswertung gesendet. Wenn `b.php` nun auch noch Daten braucht, die in `a.html` eingegeben wurden, müssen sie irgendwie über `b.html` mitgesendet werden.

Dazu kann in `b.html` ein verstecktes Feld gesetzt werden, dessen Wert mit der gewünschten Information besetzt wird. Beim Absenden der Einträge von `b.html` wird dieser Wert mitgesendet und kann in `b.php` über den Namen des versteckten Feldes ausgelesen werden.

Dieses Vorgehen empfiehlt sich nur bei unkritischen Daten, das diese Daten in der Quellcode Ansicht des Browsers lesbar sind.

Datenübertragung durch direkte Links

Daten lassen sich auch durch Direktangaben bei Links an Scripte senden z.B:

```
<a href=http://localhost/antwort.php?Artikelgruppe="Holz">Holz </a>
```

Hochkommata in Strings

Sollen in einem String Hochkommata dargestellt werden, so müssen Sie als Sonderzeichen dargestellt werden. Dieses Problem stellt sich oft bei der dynamischen Erzeugung von HTML-Code, da viele Attribute Werte in Hochkommata erwarten.

Bsp: Die Zeile `<INPUT type="text" name="Daten1" >
` müsste dynamisch `echo "<INPUT type=\"text\" name=\"Daten1\" >
 "` so erzeugt werden. Alternativ können auch einfache Hochkommata im String verwendet werden, da HTML auch diese akzeptiert: `echo "<INPUT type='text' name='Daten1' >
"`

einige String Funktionen in PHP

Funktion	Bedeutung, u.U. mit Bsp
<code>strlen(\$str) *</code>	liefert true, wenn kein Wert in der Variable steht. Damit kann geprüft werden, ob ein Formularfeld ausgefüllt wurde <code>\$str=\$_GET["kdnr"];</code> <code>if(strlen(\$str)==0)echo "Bitte Feld KdNr ausfüllen";</code>
<code>trim(\$str)</code>	entfernt führende und folgende Leerzeichen <code>\$str=\$_GET["firma"];</code> <code>if(trim(\$str)=="Müller KG") echo"korrekt"</code>
<code>strtolower(\$str)</code> <code>strtoupper(\$str)</code>	liefert den String in Klein-bzw. Großbuchstaben. Bsp <code>\$password=strtolower(\$password);</code>

*) Tipp: `strlen` ist in diesem Fall `empty` vorzuziehen, da `empty` auch dann `true` liefert, wenn in dem String die Ziffer 0 steht.

weitere Funktionen finden Sie in Tutorials und Literatur, z.B. selfPHP