

**28.** Warum muss man in jedem Formular, in dem der Benutzer Eingaben zur Verarbeitung durch ein PHP-Skript vornehmen soll, einen Submit-Button unterbringen? Warum muss man jedem Formularfeld mit Hilfe des Attributs **name** einen Namen geben? Was passiert bei der Übertragung, wenn Sie dies vergessen? Wozu dienen die Attribute **action** und **method** im <FORM>-Tag? Was passiert beim Absenden des Formulars, wenn Sie sie nicht angeben?

**29.** Schreiben Sie ein kleines Rechenprogramm.

- a. Gestalten Sie eine Formularseite (**formular.php**), die ungefähr so aussieht:

Erste Zahl:	<input type="text" value="10"/>
Zweite Zahl:	<input type="text" value="5"/>
<input type="button" value="Rechnen"/>	

- b. Senden Sie die Formulardaten an ein PHP-Skript, das die Summe, die Differenz usw. der beiden eingetragenen Zahlen berechnet und ausgibt (**rechnen.php**):

10 + 5 = 15  
10 - 5 = 5  
10 \* 5 = 50  
10 / 5 = 2

- c. \* Bringen Sie die Ausgabe in eine schönere Form, bei der die Gleichheitszeichen alle direkt untereinander stehen.  
Tipp: Benutzen Sie eine Tabelle!
- d. \* Bringen Sie in dieser PHP-Seite zusätzlich einen Link zurück zur Formularseite unter, damit der Benutzer eine neue Rechenaufgabe eingeben kann.
- e. \* Übergeben Sie mit Hilfe dieses Links die beiden Zahlen wieder an die Formularseite. Verwenden Sie diese, um sie als Vorgaben in die Textfelder zu schreiben. Testen Sie Ihr Programm ausführlich.
- f. Was passiert, wenn Sie die Datei rechnen.php direkt im Browser aufrufen?  
Was passiert, wenn Sie statt Zahlen Buchstaben eintragen?  
Was passiert, wenn Sie Zahlen mit Nachkommastellen eintragen?  
Was passiert, wenn Sie statt Zahlen Sonderzeichen (Leerzeichen, &, +, o.ä.) eintragen?

Versuchen Sie sich den Ablauf des Programms klar zu machen, beachten Sie insbesondere, welche Werte von einer Seite zur anderen übertragen werden. Achten Sie dazu auf die Adresszeile in Ihrem Browser. Überlegen Sie sich zuerst, mit welcher Reaktion Ihres Programms Sie rechnen, bevor Sie testen.