

Damit man User-Eingaben mit PHP auswerten kann, muss man im HTML Code ein `<form>` Element definieren und die Attribute `action` und `method` festlegen. Das `action` Attribut zeigt auf die PHP Datei und über die Methode wird die Art der Datenübertragung näher bestimmt. Mit `method="post"` werden die Werte aller Eingabe-elemente des `<form>` im Hintergrund übergeben. Mit der POST-Methode können auch größere Datenmengen (z. B. aus einer `<textarea>`) übergeben werden!

HTML

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



Das HTML Formular übergibt die Eingaben an `auswertung.php`. Natürlich kann die PHP-Datei auch einen anderen Namen haben. Wichtig sind die Pfadangaben – diese sind relativ oder absolut!

```
<form action="auswertung.php" method="post">
  <input type="text" name="Vorname" placeholder="Vorname" >
  <input type="email" name="eMail" placeholder="eMail" >
  <input type="submit">
</form>
```

PHP

```
$_POST["name"]
```



In `auswertung.php` werden die Werte übernommen. Die Namen der Eingabefelder sind auch zugleich die Bezeichnungen für die `$_POST` Variablen.

```
<?php
$name = $_POST["Vorname"];
echo "<h1>Hallo $name </h1>";
echo "Deine eMail: " . $_POST["eMail"];
?>
```

Lässt man das `action` Attribut im `<form>` Element weg, dann werden die Formulardaten an die eigene Webseite zurückgesendet. In so einem Fall muss man überprüfen, ob die `$_POST` Variablen auch gesetzt wurden, weil es sonst beim ersten Aufruf zu einer Fehlermeldung kommt. Die Prüfung erfolgt mit `isset` in einer `if` Verzweigung. `isset` sendet ein `TRUE` zurück, wenn es die Variable gibt. Das Beispiel unten zeigt, wie das Formular und die PHP Auswertung zusammen in einem File funktionieren.

PHP

```
isset($_POST["name"])
```

```
<form method="post">
  <input type="text" name="Vorname" >
  <input type="submit">
</form>

<?php
if(isset($_POST["Vorname"])) {
    $name = $_POST["Vorname"];
    echo "<h1>Hallo $name </h1>";
}
?>
```