

Eine Funktion übernimmt einen oder mehrere Werte, die jeweils der dazu passenden Variable zugewiesen werden. Innerhalb der Funktion werden die gewünschten PHP Anweisungen ausgeführt. Am Ende übergibt die Funktion mit `return` einen Wert an den ausführenden Aufruf. Natürlich kann eine Funktion auch ohne Wertübernahme und -übergabe gescrriptet werden.

## PHP



```
function Namen($wert1) {PHP-Anweisungen; return $val}
```

Beim Aufruf der Funktion werden die Werte in der Klammer an die Funktions-Variablen übergeben. z. B. `$xy = starteBerechnung(3,5);` Die Funktion bearbeitet die Werte und schickt das Ergebnis mit dem `return` Befehl an seinen Aufrufer zurück.

Im Beispiel löst eine Funktion eine einfache Multiplikation!

```
<?php
function starteBerechnung($x, $y) {
    $ergebnis = $x * $y;
    return $ergebnis; }

$xy = starteBerechnung(3, 5);
echo $xy;
?>
```



Man kann innerhalb einer Funktion nicht so ohne weiteres auf die Variablen außerhalb der Funktion zugreifen. Ebenso gelten die Variablen innerhalb der Funktion nur für die Funktion. Man kann aber globale Variablen definieren. z. B. mit `$GLOBALS["Variablenname"]`

```
<?php
function starteBerechnung($x) {
    $ergebnis = $x * $GLOBALS["zahl"];
    return $ergebnis; }

$zahl = 5;
echo starteBerechnung(3);
?>
```



Mit `return` wird die Funktion augenblicklich beendet. Ein `return;` ohne Wert übergibt `NULL` an den Aufrufer. Im Vergleich: mit `exit;` wird ebenfalls der PHP Code augenblicklich beendet. `exit;` ignoriert jedoch allen nachfolgenden Code und beendet am Punkt seines Aufrufes.

Es ist sinnvoll die Funktionen in eine externe Datei auszulagern und sie mit `include` wieder einzubinden. Ausgelagerter PHP-Code braucht auch kein HTML-Grundgerüst.

## PHP



```
include "funktionen.inc.php";
```

Mit `include` wird eine PHP-Datei eingebunden und sofort ausgeführt. Damit kann man PHP Code auslagern, damit der Code auch anderen PHP Seiten zur Verfügung steht. Der Pfad kann relativ oder absolut sein.