

Beispiel

In dem Beispiel der Kreisflächenberechnung wird das Ergebnis bereits in der Funktion ausgegeben, sodass der Aufrufer das Ergebnis nicht mehr benötigt.

```
<?php
define("PI",3.1415926);

function flaeche_d($d) {
    echo "d=".$d."m; A=" . PI/4*$d*$d . "m²<br>";
}

flaeche_d(10);
flaeche_d(25);
flaeche_d(74.3);

?>
```

Die Funktion `flaeche_d()` wird aufgerufen, ohne dass eine Variable den Rückgabewert benötigt. Deshalb reicht der Aufruf der Funktion aus.

Vorteile von Funktionen

- ☒ Änderungen am Programm, die zu einem späteren Zeitpunkt auftreten, lassen sich schneller durchführen, da sich eine Änderung in der Funktion auch auf den Quellcode auswirkt, der auf diese Funktion verweist.
- ☒ Der Aufwand zur Realisierung eines größeren Programmierprojektes wird verringert.
- ☒ Die Struktur des Programms lässt sich leichter nachvollziehen, da der Quellcode übersichtlicher wirkt.

6.8.3 Verweise als Parameter

Normalerweise werden den Funktionen Werte als Parameter übermittelt. Wenn Sie den Wert dieses Parameters innerhalb der Funktion ändern, bleibt der Parameter außerhalb der Funktion unverändert. Wollen Sie aber genau das erreichen, dann müssen Sie die Parameter als Verweise übergeben.

Wenn eine Funktion einen Parameter generell als Verweis behandeln soll, wird ein kaufmännisches Und **&** in der Funktionsdefinition vor den Parameternamen gesetzt.

Beispiel: *funktion3.php*

Mit diesem Beispiel wird Ihnen gezeigt, wie sich der Parameterverweis auf die Variable im gesamten Skript auswirkt.

```
① <?php
    function wechsel_kaffee($art) {
        $art = "Espresso";
    }

②    function wechsel_kaffee_verweis(&$art) {
        $art = "Espresso";
    }

③    $kaffee = "Capuccino";
④    wechsel_kaffee($kaffee);
⑤    echo $kaffee;
⑥    wechsel_kaffee_verweis($kaffee);
⑦    echo $kaffee;

?>
```