

## 6.8 Funktionen

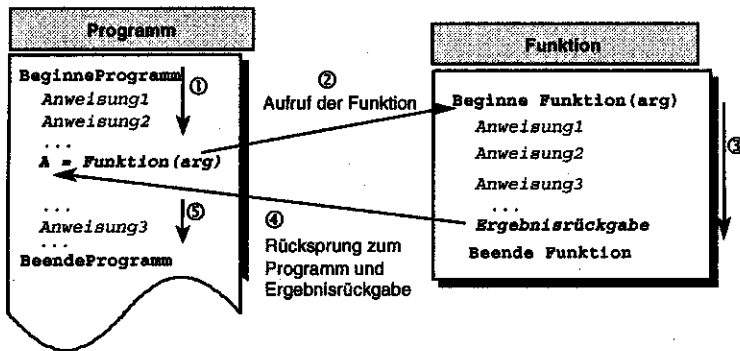
Funktionen sind eigenständige Unterprogramme, die vom Skript oder anderen Unterprogrammen mehrmals aufgerufen und abgearbeitet werden können. Diese beinhalten zum Beispiel immer wiederkehrende Anweisungsfolgen. Anstatt immer und immer wieder denselben Code zu schreiben, wird er in eine Funktion eingebettet und kann über den Funktionsaufruf beliebig oft aufgerufen werden.

### Beispiel

Funktionen liefern dem Aufrufer einen Wert zurück. Funktionen können daher in Ausdrücken verwendet werden. Wie in der Mathematik entsprechen sie dabei einem bestimmten Wert, z.B.

Resultat = Addiere(2, 3) => Resultat hat nach dem Aufruf der Funktion Addiere den Wert 5.

Das Skript wird bis zum Aufruf einer Funktion abgearbeitet ①. Der Funktionsaufruf ② erzwingt einen Sprung in die abgegebene Funktion. Jetzt werden die Anweisungen der Funktion abgearbeitet ③. Mit dem Verlassen der Funktion wird zurück zum Aufrufer gesprungen, dem das Ergebnis der Funktion übergeben wird ④. Das Skript wird weiterverarbeitet ⑤.



Aufruf einer Funktion

### 6.8.1 Funktionsaufruf

#### Syntax einer Funktion

```
function Name ([Argument 1, Argument 2,... Argument n]) {  
    Anweisungen;  
    return $Wert;  
}
```

- ☑ Die function-Anweisung leitet eine Funktion ein.
- ☑ Name ist die Bezeichnung der Funktion, mit der sie im weiteren Verlauf angesprochen werden kann.
- ☑ In den runden Klammern werden die Bezeichnungen der einzelnen Übergabeparameter angegeben, dessen Werte innerhalb der Funktion verarbeitet werden sollen.
- ☑ Eine Funktion kann auch keinen Übergabeparameter besitzen. Der Funktionsaufbau könnte z. B. function Warnung(); lauten.

Funktionen werden mit dem Namen aufgerufen, mit dem sie definiert worden sind. Wenn die Funktion aufgerufen wird, werden die internen Funktionsanweisungen abgearbeitet. Das Ergebnis wird dem Aufrufer übergeben.