Modul 16 · Funktionen mit Rückgabe

Übersicht: alle Merk- und Infokästen mit ihren Beispielen

Aufgabe 2 – return

Funktionen mit Rückgabewert

Mit `return` gibt eine Funktion einen **Rückgabewert** an die Stelle zurück, an der sie aufgerufen wurde:

ALLGEMEIN

```
def name(parameter):
    anweisungsblock
    return wert

variable = name(wert)
```

BEISPIEL

```
def flaeche(a, b):
    return a * b

gesamt = flaeche(3, 4) + flaeche(2, 5)
```

- Der Aufruf `flaeche(3, 4)` wird durch den Rückgabewert **12** ersetzt. Man kann ihn speichern, damit rechnen oder ihn ausgeben.
- Bei `return` **endet** die Funktion sofort – Zeilen danach werden nicht mehr ausgeführt.
- Eine Funktion ohne `return` gibt `None` zurück („nichts“).
- `print` und `return` sind verschieden: `print` zeigt etwas auf dem Bildschirm an, `return` liefert einen Wert an das Programm.

Funktionstests: Bei Aufgaben mit Rückgabewerten ruft ProgVenture deine Funktion nach dem Programm noch mit weiteren Werten auf und prüft, was sie zurückgibt. So arbeiten auch Profis: Sie schreiben Tests, die eine Funktion automatisch überprüfen.

Zusammenfassung

Das Wichtigste

Mit _____ gibt eine Funktion einen _____ zurück. Er ersetzt den _____ im Programm. Bei `return` _____ die Funktion sofort. Eine Funktion ohne `return` liefert _____. _____ zeigt einen Wert nur an, gibt ihn aber nicht zurück.