

Errata

Python 3 Schnelleinstieg

1. Auflage 2021

Michael Weigend

ISBN: 978-3-7475-0328-7

mitp 2021

S. 27, letzter Absatz

Im ersten Gliederungspunkt müsste es richtig wie folgt lauten:

Andere Zeichen sind nicht erlaubt.

S. 31, Tabelle 1.1:

*In der dritten Zeile der Tabelle fehlt zu Beginn ein Stern *. Die Tabelle muss korrekt so lauten (Änderungen gelb markiert):*

Operator	Erklärung
()	Klammern
**	Potenzieren
* / // %	Multiplikation, Division, ganzzahlige Division, Modulo
+ -	Addition, Subtraktion

S. 38

Bei den float-Literalen dürfen vor dem e bzw. E keine Leerzeichen stehen. Außerdem wurden zwei Exponenten nicht hochgestellt. Richtig muss es heißen:

1.0e-3 entspricht der Zahl $1 \cdot 10^{-3}$ bzw. 0.001

2.1E+2 entspricht der Zahl $2,1 \cdot 10^2$ bzw. 210

S. 67

Die letzten beiden Zeilen des Dialogbeispiels müssen richtig wie folgt lauten:

```
Gesamtkosten: 7400.0 €  
Kosten pro Person: 370.0 €
```

S. 85

Die letzte Zeile muss richtig so lauten:

```
while planeten != set():
```

S. 79, Zeile 6

Richtig muss es heißen:

Die Variable `gerade` erhält den Wert `True`, wenn die Variable `x` eine gerade

S. 100

Hier ist kein Fehler, aber der Text (bis 5.6) wird so vielleicht verständlicher (vielleicht auch nicht):

Beispiel:

```
def enthalten(kollektion, zahl):  
    for x in kollektion:  
        if round(x) == round(zahl):  
            return True           #1  
    return False                 #2  
  
print(enthalten([1, 2, 3], 2.4)  #3  
print(enthalten([1, 2, 3], 4.9)  #4
```

Ausgabe:

```
True  
False
```

Bei einem Aufruf der Funktion `enthalten()` wird nur eine der beiden `return`-Anweisungen ausgeführt und dann unmittelbar die Ausführung der Funktion beendet. Zum Beispiel liefert der Aufruf

```
enthalten ([1, 2, 3], 2.4)
```

das Ergebnis `True`. Beim zweiten Durchlauf der `for`-Anweisung wird in Zeile #1 der Wahrheitswert `True` zurückgegeben und dann die Ausführung der Funktion abgebrochen. Zeile #2 wird nicht mehr ausgeführt. Nur wenn z.B. beim Aufruf

```
enthalten ([1, 2, 3], 4.9)
```

in der Kollektion keine passende Zahl gefunden worden ist, wird Zeile #2 ausgeführt und `False` zurückgegeben.

S. 115*Unter*

```
from modul import *
```

sollte folgender Hinweis eingefügt werden:

Praxistipp

Der Import aller Namen eines Moduls mit `*` führt zu kurzen Programmtexten und wird deshalb in den Beispielen in diesem Buch gelegentlich angewendet. Bei größeren Modulen mit vielen Namen sollte man aber besser darauf verzichten. Denn es kann passieren, dass versehentlich ein Name importiert wird, der bereits verwendet wird und nun überschrieben wird. Das führt dann zu logischen Fehlern im Programm.

S. 189, vorletzte Zeile

Auch hier sollte eine kleine Warnung eingefügt werden:

Aus dem Modul `tkinter` werden alle Namen importiert. Das ist sehr bequem und führt hier zu keinem Problem. Grundsätzlich aber besteht aber die Gefahr, dass versehentlich ein Name importiert werden, der schon verwendet wird.