2025/11/19 08:40 LU03q - Zahlen ausgeben

# LU03g - Zahlen ausgeben



Wir gewöhnen uns an, immer F-Strings 'f{} zu verwenden.

# Konstanten und Variablen kombinieren

Um Konstanten und Variablen auszugeben, können wir diese zu einem String zusammensetzen (konkatenieren). Diese Möglichkeit haben Sie eventuell bereits in ähnlicher Form in Java kennen gelernt:

```
price = 13.15
# Beispiel 1
print('Preis: ' + str(price)) # Diese Variante verwenden wir nicht.
# Beispiel 2
print(f'Preis: {price}') # Diese Variante verwenden wir.
```

**Beispiel 1:** Die Konstante Preis : und der Wert der Variable price zu einem String zusammengesetzt. Da price eine Zahl ist, muss sie mit Hilfe der Funktion str(...) umgewandelt werden.

**Beispiel 2:** Da price eine Zahl ist, kann sie in einem F-String in geschweiften Klammern {} gestzt werden und so mit der Konstanten Preis : verbunden werden.

### **Nachkommastellen**

Bei Berechnungen mit Dezimalzahlen stellen wir oft fest, dass das Resultat sehr viele Nachkommastellen hat und manchmal auch nicht ganz korrekt ist.

Der Sourcecode

```
quantity = 4.7
price = 2.31
total = quantity * price
print(f'{quantity} Stück à CHF {price} = CHF {total}')
```

Die Ausgabe:

```
4.7 Stück à CHF 2.31 = CHF 10.85700000000001
```

Dieses Problem kann am einfachsten mit F-Strings gelöst werden. Beispiele dazu finden Sie unten.

# **Ausgabe mit F-Strings**

Siehe auch

http://cissandbox.bentley.edu/sandbox/wp-content/uploads/2022-02-10-Documentation-on-f-strings-Updated.pdf

Eine Alternative ist die Verwendung eines F-strings in der print-Funktion.

```
price = 13.15
print(f'Preis: {price}')
```

- Der Buchstabe f nach der öffnenden Klammer definiert den F-String (daher kommt auch der Name)
- Variablen können zwischen geschweiften Klammern eingefügt werden.

Dadurch entfällt für uns das Zusammensetzen des Strings und das Umwandeln von Zahlenvariablen.

#### Runden

Eine Möglichkeit die Situation zu verbessern, ist die Zahlen zu runden. Die Python-Funktion round (zahl, stellen) rundet eine Dezimalzahl auf die angegebene Anzahl von Stellen.

```
quantity = 4.7
price = 2.31
total = quantity * price
total = round(total,2)
print(f'{quantity} Stück à CHF {price} = CHF {total}')
```

Die Ausgabe:

```
4.7 Stück à CHF 2.31 = CHF 10.86
```

## Darstellung von Texten und Zahlen

Mit F-Strings können wir die Darstellung von Werten beeinflussen. Nach dem Namen einer Variable können wir verschiedene Formatcodes angeben.

```
sales_date = datetime.now()
print(f'{"Verkaufsdatum":20} {sales_date:%d.%m.%Y %H:%M}')

quantity = 4.7
print(f'{"Menge:":20} {quantity:.2f}')

price = 2.311111111
print(f'{"Preis:":20} CHF {price:.2f}')

total = quantity * price
```

https://wiki.bzz.ch/ Printed on 2025/11/19 08:40

2025/11/19 08:40 3/3 LU03g - Zahlen ausgeben

# print(f'{"Total:":20} CHF {total:.3f}')

• Der Konstanten "Verkaufsdatum:", "Menge:", "Preis:" und "Total:" sollen jeweils 20 Zeichen ausgegeben werden.

- Die Variablen quantity und price sind Dezimalzahlen und sollen mit 2 Nachkommastellen ausgegeben werden.
- Die Variable total ist eine Dezimalzahl und sollen mit 3 Nachkommastellen ausgegeben werden.
- Der Wert der Variable sales\_date soll als Datum im Format dd.mm.jjjj ausgegeben werden.

### Die Ausgabe:

Verkaufsdatum: 11.01.2023

Menge: 4.7

Preis: CHF 2.31 Total: CHF 10.857

#### m319-LU03



From:

https://wiki.bzz.ch/ - BZZ - Modulwiki

Permanent link:

https://wiki.bzz.ch/de/modul/m319/learningunits/lu03/zahlausgeben

Last update: 2025/06/23 07:45

