

1 Grundoperationen und das Zahlenformat

Lernziele In diesem Abschnitt lernst du:

- ▷ Die vier Grundrechenarten in Python auszuführen.
- ▷ Den Unterschied zwischen ganzen und gebrochenen bzw. wissenschaftlichen Zahlen kennen.
- ▷ Die wissenschaftliche Schreibweise von Zahlen zu verwenden.

Einführung: Zahlen Der Computer kennt zwei verschiedene Zahlentypen: Ganze Zahlen (engl. *integer*) und wissenschaftliche Zahlen (engl. *floating point number*).

Vielleicht kennst du die *wissenschaftliche Schreibweise* bereits aus dem Mathematik- oder Physikunterricht. Mit dieser Schreibweise lassen sich sehr grosse (oder sehr kleine) Zahlen viel kürzer und übersichtlicher schreiben. Die Masse der Erde ist z. B.:

$$5 \underbrace{980\,000\,000\,000\,000\,000\,000\,000}_{24 \text{ Stellen}} \text{ kg} = 5.98 \cdot 10^{24} \text{ kg}$$

An diesem Beispiel siehst du bereits, wie ungemein praktisch diese kürzere wissenschaftliche Schreibweise ist. Die wesentlichen Ziffern sind hier nur «598» und nach der ersten dieser Ziffern (der Fünf) kommen nochmals 24 Stellen. Daraus entsteht die wissenschaftlichen Schreibweise mit der *Mantisse* 5.98 und dem *Exponenten* 24.

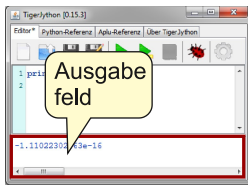
Computer arbeiten auch mit dieser wissenschaftlichen Schreibweise, konnten früher aber das $\cdot 10^{\square}$ nicht schreiben. Deshalb hat man sich vor langer Zeit auf die Darstellung mit einem *e* geeinigt: $5.98\text{e}+24$ oder einfach $5.98\text{e}24$ (*e* steht für «Exponent»).

Sehr kleine Zahlen kannst du ebenfalls so schreiben:

$$0.\underbrace{000\,000\,000\,000\,000\,000\,000\,000}_{14 \text{ Stellen}} 905 = 4.905 \cdot 10^{-14} = 4.905\text{e}-14$$

Im Prinzip entsteht die wissenschaftliche Schreibweise dadurch, dass du den Dezimalpunkt (das Komma) hinter die erste wesentliche Stelle verschiebst. Das wird durch den englischen Namen «*floating point number*» ausgedrückt, meist abgekürzt zu «*float*». Beachte, dass Computer in diesem Format nur mit 12 bis 20 Stellen arbeiten und längere Zahlen einfach runden. Das führt schnell zu *Rundungsfehlern*, wie du beim Programm auf der nächsten Seite sehen wirst.

Beim Rechnen mit wissenschaftlichen Zahlen auf dem Computer entstehen oft Rundungsfehler. Wegen dieser Rundungsfehler spielt auf dem Computer die Reihenfolge der einzelnen Rechnungen eine Rolle: $x + y \neq y + x$.



Ausgaben mit print Damit dir der Computer überhaupt etwas ausgibt, musst du ihn anweisen, das zu tun. Die Anweisung dafür heisst `print`. Es genügt also nicht, einfach die Rechnung einzugeben. Du musst dem Computer mit einem `print` auch sagen, dass er das Resultat der Rechnung ausgeben muss, etwa `print 13/3`, um das Resultat `4.3333333333` zu erhalten.

Das Programm In diesem Programm soll uns Python zuerst die Zahl 1 234 567 890 123 456 789 zweimal ausgeben. Im ersten Fall ist es für den Computer eine ganze Zahl (*integer*). Weil im zweiten Fall am Schluss ein Dezimalpunkt steht, ist das automatisch eine wissenschaftliche Zahl und wird auch gleich gerundet: Die Ausgabe von Zeile 2 ist `1.23456789012e+18`.

```
1 print 1234567890123456789
2 print 1234567890123456789.0
3
4 print (1/2 + 1/3 + 1/6) - 1
5 print (1/6 + 1/3 + 1/2) - (1/2 + 1/3 + 1/6)
```

In Zeile 4 und 5 berechnet Python schliesslich zuerst das Resultat der Rechnung und gibt das dann aus. Siehst du, dass $\frac{1}{2} + \frac{1}{3} + \frac{1}{6} = 1$ ist? Wenn Python richtig rechnet, sollte die Ausgabe 0 sein – ist sie aber nicht! Kannst du diesen Unterschied erklären?

Die wichtigsten Punkte Für Rechnungen verwendest du in Python die Grundoperationen `+`, `-`, `*` und `/`. Zahlen werden entweder als ganze Zahlen (*integer*) oder in wissenschaftlicher Schreibweise (*float*) angegeben, also z. B. `5.98e24` für $5.98 \cdot 10^{24}$. In der wissenschaftlichen Schreibweise rundet Python immer auf 12 bis 20 Stellen.

Wenn du grosse ganze Zahlen aus gibst, dann hängt Python ein «L» an, das für «long» steht. Das ist ein Überbleibsel von früher und hat heute keine Bedeutung mehr.

Python schreibt die Resultate und Zahlen nur auf den Bildschirm, wenn du das mit `print` angibst.

AUFGABEN

1. Du kannst $\frac{1}{3}$ auch als Dezimalzahl eingeben: `0.333...`. Wie viele Dreien musst du eingeben, bis die folgenden Rechnungen das richtige Resultat liefern?

(a) `print 3 * 0.333`, (b) `print (3 * 0.333) - 1`

2. Die Entfernung der Erde von der Sonne beträgt rund $1.496 \cdot 10^8$ km. Die Lichtgeschwindigkeit beträgt ca. 300 000 000 m/s. Berechne mit Python, wie lange das Sonnenlicht braucht, um die Erde zu erreichen und gib das Resultat in Minuten an.