

Es ist dem Computer nicht möglich, Reelle Zahlen zu speichern. Rationale Zahlen sind als Paar zweier `int`-Werte möglich, jedoch ist dies langsam und speicherintensiv (und noch langsamer, wenn man versucht, jeden Bruch auf den kleinsten Nenner zu bringen). Eine Alternative dazu sind fixed-point-Zahlen, wobei mit ganzen Zahlen gerechnet wird, die jeweils den Zähler eines Bruches mit festem Nenner beschreiben. Eine nicht ganz offensichtliche (und häufig nicht ideale, aber gut funktionierende) Methode ist die Verwendung von Gleitkommazahlen, welche in Python durch den Datentyp `float` beschrieben werden. Diese haben intern die Form $s * m * 2^e$, wobei

- `s`, das Vorzeichen, -1 oder 1 ist
- `m`, die Mantissa, als Binärzahl in der Form von `1.xyz` gespeichert wird (mit 52 Bits als Nachkommastellen)
- `e`, der Exponent, einen Wert zwischen -1022 und 1023 hat und als Binärzahl mit Offset 1023 gespeichert wird

Fast alle Operationen mit Gleitkommazahlen sind ungenau, insbesondere das Umwandeln zwischen ihnen und Zahlen im Dezimalsystem. Ein bekanntes Beispiel dafür ist auf

0.30000000000000004.com zu finden.

Version #4

Erstellt: 2026-05-10 08:34:13 UTC von Fabian Schiener

Zuletzt aktualisiert: 2026-05-10 21:02:40 UTC von Fabian Schiener