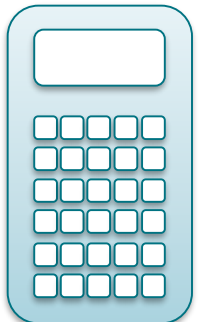


- In Python lassen sich einfache Rechnungen mit Zahlen durchführen. Python unterstützt **ganze Zahlen** (integer oder kurz int), **Fließkommazahlen** (float) und komplexe Zahlen.
- Ganze Zahlen sind wie in der Mathematik die positiven Zahlen 1, 2, 3, ..., die Zahl 0 und die negativen Zahlen -1, -2, -3, ...
- Fließkommazahlen kann man sich wie Dezimalbrüche vorstellen, z.B. 1.5, -0.01, 7007.333. Allerdings haben sie eine interne Darstellung als Zahlen zur Basis 2.
- Komplexe Zahlen verwenden wir in dieser Lehrveranstaltung nicht.
- Python lässt sich wie ein Taschenrechner verwenden.
 - Öffnen Sie ein Fenster „Eingabeaufforderung“ (unter Windows).
 - Geben Sie das Kommando `python` ein.
 - Die interaktive **Python-Shell** startet (sofern Python korrekt auf dem Rechner installiert ist).



- Tippen Sie die Rechnung ein und drücken Sie die Eingabetaste.

- Das Rechenergebnis erscheint in der nächsten Zeile.

- Beispiele für Rechnungen:

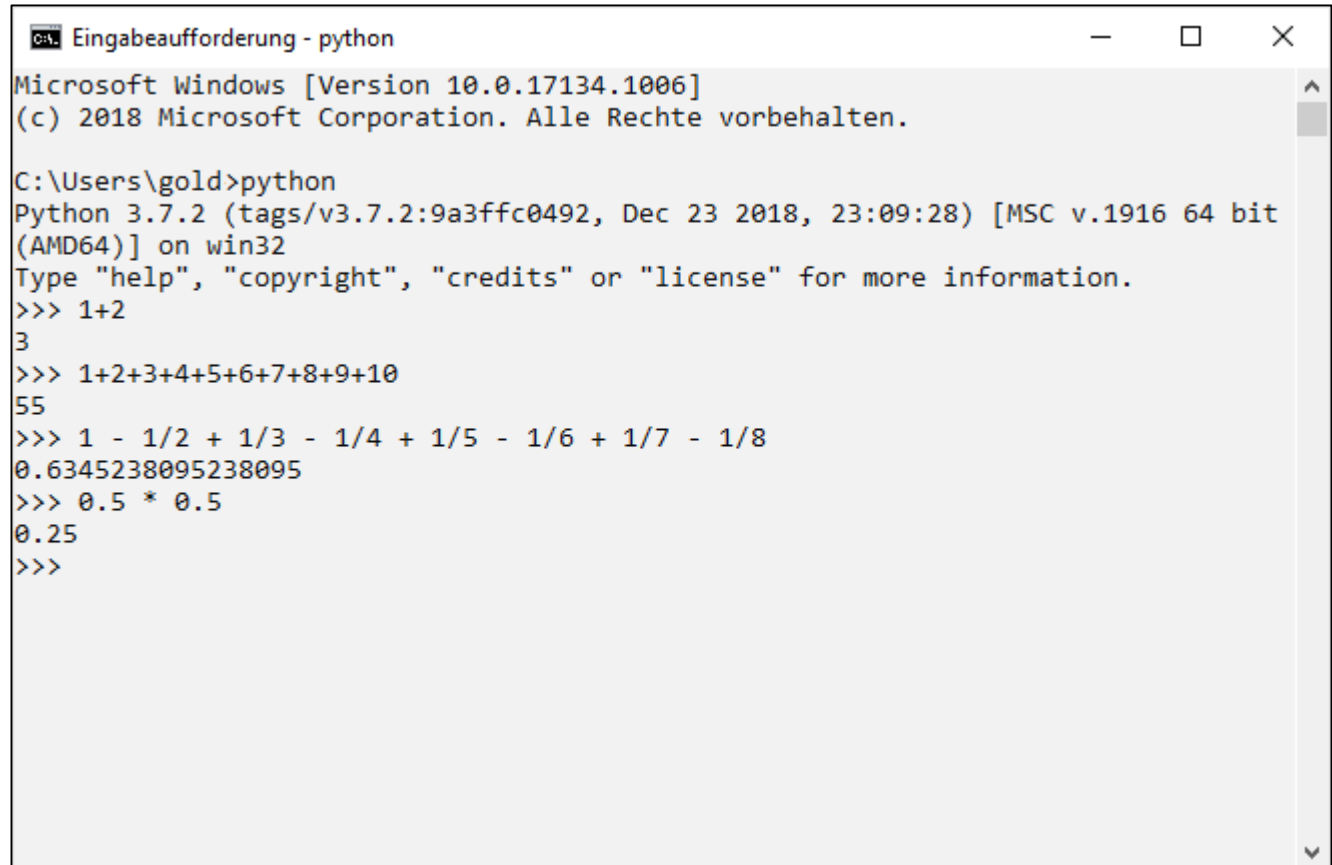
$1+2$

$1+2+3+4+5+6+7+8+9+10$

$1 - 1/2 + 1/3 - 1/4 + 1/5 - 1/6 + 1/7 - 1/8$

$0.5 * 0.5$

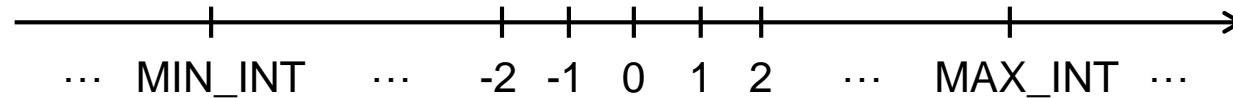
- Leerzeichen spielen keine Rolle, machen die Rechnungen aber übersichtlicher.
- Das Kommando `exit()` beendet die Shell.



```
Eingabeaufforderung - python
Microsoft Windows [Version 10.0.17134.1006]
(c) 2018 Microsoft Corporation. Alle Rechte vorbehalten.

C:\Users\gold>python
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> 1+2
3
>>> 1+2+3+4+5+6+7+8+9+10
55
>>> 1 - 1/2 + 1/3 - 1/4 + 1/5 - 1/6 + 1/7 - 1/8
0.6345238095238095
>>> 0.5 * 0.5
0.25
>>>
```

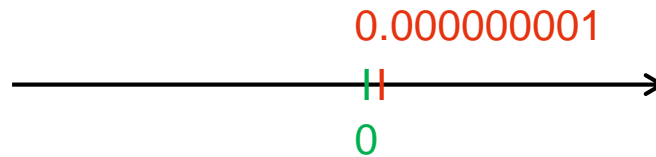
- Ganze Zahlen und Fließkommazahlen können in Ausdrücken auch gemischt verwendet werden.
 - Trotzdem sollten Sie zwischen ganzen Zahlen und Fließkommazahlen unterscheiden. Beispielsweise ist 4 eine ganze Zahl, aber 4.0 eine Fließkommazahl.
- Beachten Sie, dass in den meisten Programmiersprachen die Wertebereiche von Zahlenmengen beschränkt sind. Es gibt z.B. eine kleinste ganze Zahl (MIN_INT) und eine größte ganze Zahl (MAX_INT).



Deshalb kann es passieren, dass die Addition zweier (großer) ganzer Zahlen den Wertebereich übersteigt und nicht mehr dargestellt werden kann.

- In Python dagegen sind Zahlenbereiche nicht beschränkt!

- Beim Rechnen mit Fließkommazahlen können Rundungsfehler auftreten. Deshalb ist es möglich, dass eine Zahl z.B. 0 sein sollte, aufgrund der Fehler aber einen sehr kleinen Wert nahe bei Null besitzt. Wenn Sie jetzt die Zahl mit 0 vergleichen, ist das Ergebnis „Zahl ungleich 0“, also eigentlich falsch.



- Besser ist es auf ≥ 0 zu vergleichen oder, wenn das nicht geht, zu prüfen, ob die Zahl in einem sehr kleinen Intervall um 0 liegt.

- Viele Taschenrechner beherrschen auch mathematische Funktionen wie Sinus oder Wurzelziehen. Um mathematische Funktionen verwenden zu können, ist das Python-Modul `math` erforderlich. Vor dem Aufruf einer mathematischen Funktion, muss die Anweisung `import math` eingegeben werden. Beispiel:

```
import math  
math.sin(3.14)
```

- Die Kreiszahl π wird durch `math.pi` bezeichnet. Beispiel:

```
2 * 4.0 * math.pi  
math.sin(math.pi / 2)
```

- Für Fließkommazahlen ist auch die wissenschaftliche Notation möglich. Beispiel: $1024 = 1.024e3$.
- Beachten Sie, dass für Fließkommazahlen die anglo-amerikanische Schreibweise mit einem Dezimalpunkt verwendet wird.

■ Folgende **Rechenoperatoren** gibt es:

- $+$, $-$, $*$, $/$ Addition, Subtraktion, Multiplikation, Division
- $//$ ganzzahlige Division, Nachkommastellen werden abgeschnitten
- $\%$ Modulo, Rest bei ganzzahliger Division, z.B. $14 // 3 = 4$ und $14 \% 3 = 2$
- $**$ Potenzieren

- Natürlich sind auch Klammern möglich. Ohne Klammerung gelten die üblichen Vorrangregeln „Punkt vor Strich“. Potenzierung hat Vorrang vor Multiplikation. Beispiele:

$2 ** 4 + ((16 + 1) // 2 + 20 \% 8)$

$2 ** 5 * 2$

- Lesestoff: Lesen Sie die Kapitel 2, 4.3.1, 4.3.2, 4.3.6 in
B. Klein (2018) *Einführung in Python 3: Für Ein- und Umsteiger*. Carl Hanser Verlag, 3. Aufl.

Beispiele für
Modulo:

$$14 \% 3 = 2$$

14
13
12
11
10
9
8
7
6
5
4
3
2
1

$$11 \% 2 = 1$$

11
10
9
8
7
6
5
4
3
2
1



1. Die Kreiszahl π lässt sich durch die folgende unendliche Reihe annähern:
$$\pi = 4 * (1 - 1/3 + 1/5 - 1/7 + \dots)$$
 - a) Berechnen Sie einen Näherungswert von π , indem Sie die Summe mit 11 Summanden berechnen. Der letzte Summand lautet $1/21$.
 - b) Wie groß ist der Fehler? Berechnen Sie dazu die Differenz zu π .
2. In vielen Programmiersprachen ist $2^{32} - 1$ die größte darstellbare ganze Zahl.
 - a) Wie lautet diese Zahl?
 - b) Berechnen Sie mit Python das Quadrat dieser Zahl!
3. Welchen Wert hat der Ausdruck $19 // 5 * 5 + 19 \% 5$?