

Einführung in die computerlinguistische Programmierung mit Python

02-04: Zahldatentypen: Integers und Floats

Integers - Ganzzahlen

Alle positiven und negativen Ganzzahlen werden in Python als der Datentyp **Integer** interpretiert.

```
In [1]: zahl_1 = 1
        print(zahl_1)

        zahl_2 = 2
        print(zahl_2)

        zahl_3 = 3
        print(zahl_3)

        print(((zahl_1 + zahl_2) * zahl_3 - zahl_1) // zahl_2)

        print(zahl_3 // zahl_2)

        print(zahl_3 % zahl_2)
```

- Ihr könnt Python als Taschenrechner einsetzen; dabei verhält sich der Python-Interpreter, wie man es von einem herkömmlichen Taschenrechner erwartet.
- **Terme werden "von innen nach außen" verarbeitet.**
- **Terme in Klammern werden zuerst aufgelöst.**
- Die bekannten Operationen Addition + , Subtraktion - , Multiplikation * , und Division / können wir, wie oben gezeigt, genutzt werden.
- Beim ganzzahligen Dividieren mittels // wird ein Integerwert ohne Rest zurückgegeben.
- Die Modulooperation a%b gibt den Rest einer ganzzahligen Division a // b zurück.

Floats - Gleitkommazahlen



Gleitkommazahlen sind eine Annäherung an die reellen Zahlen. Warum nur eine Annäherung? Mehr Informationen finden Ihr hier: <https://www.youtube.com/watch?v=PZRI1lfStY0>

```
In [1]: zahl_4 = 0.1
        print(type(zahl_4))

        zahl_5 = 3*zahl_4
        print(zahl_5)

        print(zahl_5 == 0.3)  # Oh weh!

        print("-----")

        print(zahl_5 < 0.3)
        print(zahl_5 > 0.3)
```

- Es werden nur Annäherungen an tatsächliche Zahlenwerte gespeichert.
- Rundungsfehler passieren!
- Der Divisionsoperator `/` gibt Quotienten als Floats zurück.

```
In [1]: zahl_2 = 2
        zahl_4 = 4

        zahl_6 = zahl_4 / zahl_2
        print(zahl_6)

        zahl_7 = 0.6 / 0.3
        print(zahl_7)
```

Umwandlung anderer Datentypen zu Integers oder Floats

Python hält zur Umwandlung von Strings und Zahlen zu einem der beiden Zahl-Datentypen die Funktionen `int()` und `float()` bereit. Diese sind hilfreich wenn z.B. bei der Verarbeitung von Texten Zahlen aus Strings entnommen werden müssen.

```
In [1]: zahl_8 = int("8")
print(zahl_8)
print("----")
zahl_9 = int("-9")
print(zahl_9)
print("----")
zahl_10 = int(" 10 \n")
print(zahl_9)
print("----")
# zahl_11 = int("10.51")
# print(zahl_11)
```

Die Funktion `int()` gibt Integers zurück. Vorzeichen werden beachtet. Whitespace-Zeichen werden ignoriert. Dezimalpunkte können nicht interpretiert werden.

```
In [1]: zahl_12 = float("12.56")
print(zahl_12)
print("----")
# zahl_13 = float("1E.99")
# print(zahl_13)
print("----")
zahl_14 = float("\t 14.76\n")
print(zahl_14)
```

Die Funktion `float()` gibt Gleitkommazahlen zurück. Auch hier werden Vorzeichen erkannt und Whitespace ignoriert. Im Gegensatz zu `int()` beachtet `float()` Dezimalpunkte.

Eingebaute Funktionen für Integers und Floats

Die folgenden Befehle stehen in Python als eingebaute Funktionen bereit

Befehl	Funktion
<code>float(n)</code>	Wandelt Stings oder Integers in Gleitkommazahlen um.
<code>int(n)</code>	Wandelt Stings oder Integers in Ganzzahlen um.
<code>pow(n1,n2)</code>	Bildet die Potenz $n1^{n2}$. Entspricht <code>n1 ** n2</code> .
<code>sum([n1,n2,...])</code>	Summiert Sequenzen von Zahlen. Entspricht <code>n1 + n2 + ...</code> .

Zusammenfassung

- Zahlen werden in Python als Integers oder Floats dargestellt.
- Integers sind Ganzzahlen, Floats sind Gleitkommazahlen, d.h. eine Annäherung an reelle Zahlen
- Python kann als Taschenrechner benutzt werden und folgt dabei den üblichen Konventionen
- Mit den Funktionen `int()` und `float()` können Strings und auch Zahlen in den jeweiligen Datentyp umgewandelt werden.