Merkhilfe Python A

InfoKurs

1 Grundlagen

Variablen und Zahlen

grundlagen

```
k = 8                          int: ganze Zahl
x = 3.2                        float: mit Punkt, nicht Komma
x = x + 1   x += 1   -= *=
print(k, x, 10)                Ausgabe mit Leerzeichen
# Kommentar   '''mehrzeilig'''
```

Namen: zuerst Buchstabe oder _, case-sensitiv; anzahlTreffer, anzahl_treffer

Rechnen

```
+  -  *  /  11 ** 3            hoch
6 / 2                          3.0: / liefert immer float
11 // 3                        3: ganzzahlige Division
11 % 3                         2: Modulo-Operator (Rest)
```

** vor * / // % vor + -, sonst Klammern

Strings

```
s = 'Habicht'  "auch so"
len(s)                         7: Anzahl Zeichen
s1 + ' ' + s2                  verketten
'3' + '4'                      '34'
'Alter: ' + 15                 TypeError
s.lower()  s.upper()           neuer String
'a' in s                       True
s.find('ich')                  3, -1: nicht gefunden
s.find('bb', 4)                suchen ab Index 4
'aab' < 'abb'                  alphabetisch, A–Z vor a–z
```

Indexing und Slicing

Index von vorne ab 0, von hinten ab -1

```
s[0]  s[-1]  s[len(s)-1]       'H' 't' 't'
s[7]                           IndexError
s[1:5]                         'abic': bis ausschließlich
s[:3]  s[-3:]                  'Hab' 'cht'
s[2:]  s[:-2]                  'bicht' 'Habic'
```

Eingabe und f-Strings

```
s = input('Name? ')            liefert immer String
x = int(input('Zahl: '))  float(...)
print(f'Das Doppelte von {x} ist {2*x}.')
f'{pi:.2f}'                    3.14: 2 Nachkommastellen
f'{i=}'                        i=3: Name und Wert
```

2 Bedingungen und Schleifen

if, elif, else

bedingungen

```
if x > 3:
    print('A')
elif x > 2:
    print('B')
else:
    print('C')
```

Doppelpunkt, 4 Leerzeichen Einrückung; nur der **erste** zutreffende Fall läuft

Vergleiche und Wahrheitswerte

```
>  >=  <  <=  ==  =!           liefern True/False
x = 5                          Zuweisung
x == 5                         Vergleich
volljaehrig = alter >= 18      Typ bool
x % 2 == 0                     x ist gerade
```

and, or, not

```
a and b                        beide wahr
a or b                         mindestens einer wahr
not a                          umkehren
alter >= 13 and alter <= 17
```

Reihenfolge: not, dann and, dann or; im Zweifel Klammern

while-Schleife

```
zahl = 3
while zahl <= 100:
    zahl = zahl * 2
```

läuft, solange die Bedingung True ist; im Rumpf muss sich etwas ändern, sonst Endlosschleife (Stopp-Knopf ■ bzw. Kernel → Interrupt)

for-Schleife mit range()

```
for i in range(5):             0 1 2 3 4
range(2, 5)                    2 3 4
range(3, 9, 2)                 3 5 7: Schrittweite
range(5, -1, -1)               5 4 3 2 1 0
```

bis-Zahl ausschließlich; for, wenn die Anzahl der Durchläufe bekannt ist, sonst while

```
summe = 0
for i in range(1, 100, 2):
    summe += i
```

break, continue, while True

```
break                          Schleife sofort verlassen
continue                       nächster Durchlauf
while True:
    zahl = int(input())
    if zahl == 0:
        break
```

3 Listen, Zufall, Funktionen, Dateien

Datentypen umwandeln

listen

```
type(x)                        int float str bool list
int('5')  float('3.14')  str(42)
int(3.9)                       3: abgeschnitten
int('Hallo')  int('3.14')      ValueError
```

Listen

```
a = [7, 12, 1, 5]  a = []      leere Liste
len(a)  a[0]  a[-1]  a[1:3]
a[0] = 9                       ändern (bei Strings nicht)
a.append(8)                    am Ende anhängen
sum(a)  max(a)  min(a)  abs(x)
a = []
for i in range(5):
    a.append(i)
```

Schleifen über Strings und Listen

```
for x in a:                    über die Elemente
for c in s:                    über die Zeichen
for i in range(len(a)):        über die Indizes
    a[i] = 2 * a[i]            Index nötig zum Ändern
```

Zufallszahlen

```
import random
random.random()                0 ≤ x < 1
random.randint(1, 6)           beide Grenzen dabei
random.uniform(2, 10)          Dezimalzahl
random.seed(12)                immer dieselbe Folge
```

Funktionen

```
def calc(x, y):
    '''Beschreibung und Beispiele'''
    z = x + y
    return 3*z + 1
a = calc(3, 5)
```

return gibt das Ergebnis zurück und beendet die Funktion; help(calc) zeigt den docstring

Dateien lesen und schreiben

```
f = open('input1.txt', encoding='utf-8')
s = f.readline().strip()       eine Zeile ohne \n
x = int(f.readline())          Zeile als Zahl
f.close()
f = open('a.txt', encoding='utf-8', mode='w')
```

überschreibt!

```
print(f'Summe: {x+y}', file=f)
```

4 Fehler, Debugger, Algorithmen

Fehlerarten

fehler

Syntaxfehler keine Zeile wird ausgeführt
SyntaxError (Klammer, Hochkomma, Doppelpunkt fehlt), IndentationError; auch die Zeile davor prüfen

Laufzeitfehler Abbruch mit Traceback
letzte Zeile des Tracebacks lesen: Art und Beschreibung

Logischer Fehler läuft, aber falsches Ergebnis

Laufzeitfehler

```
IndexError                     a[3] bei 3 Elementen
TypeError                      5 + "10"
NameError                      Name nicht definiert, Tippfehler
ValueError                     int(äbc")
ZeroDivisionError              5 / 0
```

Fehler suchen

1. mit Werten testen, deren Ergebnis man kennt
2. Zwischenwerte ausgeben:

```
print(f'{i=} {summe=}')
```

3. Debugger (Thonny): Käfer starten, Einzelschritt F6, Eintreten F7, Fortfahren F8, Breakpoint per Klick neben die Zeilennummer; gelbe Zeile = wird als Nächstes ausgeführt

Liste durchlaufen

```
summe = 0      # Startwerte vor der Schleife
zaehl = 0
for x in a:
    summe += x
    if x % 2 == 0:
        zaehl += 1
```

Vergleich mit dem Vorgänger (Index nötig):

```
for i in range(1, len(a)):
    if a[i-1] < a[i]:
        zaehl += 1
```

Lineare Suche

```
best = a[0]    # vorläufig das Beste
for x in a:
    if x > best:
        best = x
```

nach einem Merkmal: zusätzlich best_val = a[0] % 7 merken und vergleichen; mit a[0] starten, nicht mit 0