

Kleines Programmier-Praktikum

Andreas Boss, 06.12.2021

Ziel

Die SchülerInnen

- kennen die Grundstruktur eines Computer-Programms
- können einfache Programme selbst schreiben
- sind in der Lage, komplexere Programme zu modifizieren
- haben Spass am Programmieren

Entwicklungsumgebung

Wir arbeiten mit der Programmiersprache **Python**. Es ist die derzeit am häufigsten verwendete Programmiersprache der Welt. Sie wird besonders häufig bei Programmen der Künstlichen Intelligenz eingesetzt.

Es handelt sich um eine **Interpreter-Sprache**, d.h. der Computer versucht, das Programm während der Ausführung zu verstehen.

Wir arbeiten mit einer webbasierten Version, die keine Installation erfordert. Die Web-Seite ist:

<https://trinket.io/embed/python3/a5bd54189b>

Die Programme, mit denen wir arbeiten, findet Ihr auf

<https://github.com/andreasboss/schule>

1. Übung: Mein erstes Programm

Öffne die beiden Internetseiten in einem Browser, idealerweise in zwei unterschiedlichen Fenstern.

Bei *trinket* ist links der Bereich des Python-Programms gezeigt, rechts die Ausgabekonzole des Computers. Das gezeigte Programm macht nichts, bitte lösche es. Dann gib die folgende Zeile in das linke Fenster ein:

```
print('Hello World')
```

Drücke nun den Button «Run». Was ist passiert?

Ändere den Text in den Anführungszeichen nun ab und drücke «Run».

Gratulation! Dein erstes selbst geschriebenes Programm!

2. Übung: Mein erster Kontakt mit Variablen

Eine Variable ist ein Speicherplatz für eine Information. Das kann eine Zahl oder ein Text sein. Variablen kann man nennen, wie man will. Über den Befehl *input*, kann man den Nutzer eine Information eingeben lassen, die dann gespeichert wird. Gib das folgende Programm ein:

```
name = input('Bitte gib deinen Namen ein: ')
print('Freut mich, dich kennenzulernen,', name, '!')
```

Hier heisst die Variable *name* und die gespeicherte Information ist der eingegebene Text, man sagt ein *string*. Schreibe nun ein eigenes Programm, bei dem Du etwas fragst, und eine Antwort gibst.

3. Übung: Mein erstes Rechenprogramm

Variablen können auch Zahlen aufnehmen, mit denen man dann rechnen kann.

```
a = 12
b = 14
c = a + b
print(a, '+', b, '=', c)
```

Damit wir die Zahl interaktiv eingeben können, müssen wir dem Computer mitteilen, dass es eine Zahl ist und kein Text. Das machen wir mit dem Befehl `int()`, hier steht `int` für Integer, also eine ganze Zahl.

Ändere die erste Zeile in 2 Zeilen

```
a = input('Erste Zahl :')
a=int(a)
```

Schau, was passiert, wenn du die zweite Zeile weglässt.

Nun, schreibe Dein eigenes Rechenprogramm! Der Computer kennt die folgenden mathematischen Operationen `«+»`, `«-»`, `«*»`, `«/»`, `«(»`, `«)»` und `«**»`. Bei manchen mathematischen Operationen muss man die Zahl/Variable in Klammer setzen, z.B. bei Wurzel `sqrt(Zahl)`, Sinus `sin(Zahl)` oder Cosinus `cos(Zahl)`.

Schreibe Dein eigenes Rechenprogramm, bei dem Du Zahlen eingeben musst, um etwas zu berechnen.

4. Übung: die if-Abfrage

Mit der if-Abfrage kann man etwas testen.

```
a = input('Eine Zahl :')
a=int(a)
if a > 0:
    print(a, "ist eine positive Zahl.")
print("Tschüss.")
```

Nach `if` kommt der Ausdruck, der getestet wird. Danach kommt ein Doppelpunkt. Ist der Ausdruck wahr, werden die Befehle ausgeführt, die eingerückt sind. Es sind mehrere Zeilen möglich. Für den Vergleich kann man `«<»`, `«>»` oder `«==»` verwenden. Mit `else` kann man noch eine Alternative angeben.

```
a = input('Eine Zahl :')
a=int(a)
if a > 0:
    print(a, "ist eine positive Zahl.")
else:
    print(a, "ist keine positive Zahl.")
```

```
print("Tschüss.")
```

Man kann auch Strings vergleichen.

```
name = input('Bitte gib deinen Namen ein: ')
if name=='Andreas':
    print('Cooler Typ! ')
if name=='Peter':
    print('Langweiler! ')
```

Schreibe ein Programm, bei dem Du den Wochentag abfragst und eine coole Antwort schreibst, z.B. «Montag nervt total!!!» bei der Eingabe «Montag».

5. Übung: die Schleifen

Bei einer Schleife wird derselbe Teil des Programms immer wieder durchlaufen bis eine Liste zu Ende ist oder eine Abbruchbedingung erreicht wird. Die einfachste Schleife ist eine *for*-Schleife.

```
liste=["Andreas", "Peter", "Tina", "Anna"]
anzahl=0
cool=0
for name in liste:
    anzahl = anzahl+1
    print("Ist", name, "eine nette Person? (j/n)")
    antwort=input("(j/n)")
    if antwort=="j":
        cool=cool+1
prozent=cool/anzahl*100
print(prozent, "% deiner Bekannten sind nette Leute.")
```

Kannst Du eine Liste mit Tieren machen und ausrechnen, wie viele davon Säugetieren sind?

Eine Liste mit Zahlen kann man durch `range()` erzeugen.

```
for i in range(1, 20):
    print(i)
```

Kannst Du ein Programm schreiben, das alle Quadratzahlen bis 1000 ausdrückt?

6. Übung: Programme abändern

Oft muss man ein Programm nicht vollständig selbst schreiben, sondern kann ein Programm aus dem Internet modifizieren. Was macht unser Programm1?

```
import numpy as np
import matplotlib as mlp
```

```
import matplotlib.pyplot as plt

x = np.arange(0, 10, 1) # Start, Stop, Step
y = np.sin(x)

plt.plot(x, y, 'bo')
plt.axis( [0, 7, -1.5, 1.5] )
plt.grid(True)
plt.show()
```

Kannst Du mehr Punkte erzeugen?
 Kannst Du statt Punkten eine Linie zeichnen?
 Kannst Du die Amplitude erhöhen?
 Kannst Du eine Gerade zeichnen (mit unterschiedlicher Steigung und y-Achsen-Abschnitt)?
 Kannst Du den Kosinus zeichnen?
 Kannst Du eine Parabel zeichnen?

7. Übung: das grosse Rätsel

Was macht unser Programm2? Kriegst Du es raus? Verändere das Programm und schau, was passiert?

```
import math
import random
import matplotlib.pyplot as plt

random.seed(42)
num_events = 100
num_inside = 0
data_x = [] # x coordinates
data_y = [] # y coordinates
data_c = [] # colors

for i in range(num_events):
    x = random.uniform(-1.0, 1.0)
    y = random.uniform(-1.0, 1.0)
    data_x.append(x)
    data_y.append(y)
    if 1 >= math.sqrt(x ** 2 + y ** 2):
        num_inside += 1
        data_c.append('red')
    else:
        data_c.append('blue')
number = 4 * num_inside / num_events

print('Events: {:>9d}, Result: {:.5f}'.format(num_events, number))
```

```
plt.figure(figsize=(5.0, 5.0))  
plt.scatter(data_x, data_y, s=2, c=data_c, alpha=0.5)  
plt.show()
```

Hast Du verstanden, wie es funktioniert? Ein Tipp: es arbeitet mit Zufallszahlen (random).