Programmierung mit Python

Setup

Welche Schritte muss ich befolgen, um an einem neuen Gerät in Python zu programmieren?

> - Mein **Betriebssystem** ist **Windows** (10 oder höher), Linux oder MacOS
> - Mein **Broswer** ist **Chrome**, Safari, Firefox oder Edge
> - Meine **IDE** ist **Visual Studio Code**, Atom, Jupiter oder PyCharm
>   - Die neuste Version von Python finde ich auf https://www.python.org/
>   - Die IDE Visual Studio Code finde ich auf https://code.visualstudio.com
> - Zur **Dokumentation** nutze ich **Markdown**, Word, Notion oder ein klassisches Notizbuch
>   - Markdown nutze ich in VSCode mithilfe der Extension "*Markdown All in One*"

Algorithmus

Was ist ein Algorithmus?

> Ein Algorithmus ist eine eindeutige Handlungsvorschrift zur Lösung eines Problems oder einer Klasse von Problemen. Er überführt eine Eingabe in endlich vielen Anweisungen in eine Ausgabe. Bsp.: Kochrezept

Welche Eigenschaften hat ein Algorithmus?

> - Endlichkeit
> - Eindeutigkeit (deteministisch und determiniert)
> - Ausführbarkeit
> - Allgemeingültigkeit

Wie funktioniert der Euklidische Algorithmus?

```
solange b != 0
    r <- Divisionsrest(a, b)
    a <- b
    b <- r
Ergebnis = a
```

Programm

Was ist ein Programm?

> Ein Programm ist eine Folge von Anweisungen in einer bestimmten Sprache, die auf einem Rechner ausgeführt werden können. Es ist notwendig, um mit einem Computer zu arbeiten.

Was ist eine Programmiersprache?

> Eine Programmiersprache ist eine Sprache zur Formulierung von Algorithmen und Datenstrukturen. Sie ist eindeutig in ihrer Syntax (welche Zeichenfolgen zugelassen sind) und ihrer Semantik (was bestimme Zeichenfolgen bewirken). Für den Menschen ist sie verständlicher, jedoch wird sie vor dem Ausführen in Maschinensprache (0en und 1en) übersetzt. Sie bildet die Sprache des Quellcodes, während Maschinensprache den Maschinencode bildet.

Ausgabe

```
print("Hallo Welt!")        # with line break
print("Hallo Welt!", end="")    # no line break
```

Variable

> Variablen sind Speicherstellen. Entsprechend des Datentyps können verschiedene Werte gespeichert werden. Diese Werte sind während des Programmablaufs veränder- und abrufbar. Der Bezeichner einer Variable beginnt mit Kleinbuchstaben, ist kein Python-Schlüsselwort, ist im camelCase-Format und ist aussagekräftig.

```
x = 42                      # int
groesseVonLeBronJames = 2.06            # float
meinName = "Abels"              # str
esRegnet = False                # bool
farben = ["rot", "grün", "blau"]        # list
person = {"name": "Abels", "alter": 26}     # dict
```

> Der Datentyp einer Variable `x` kann mit `type(x)` ausgegeben und mit `float(x)` konvertiert werden.

Eingabe

```
name = input("Wie lautet dein Name?")
```

Operator

```
x = 7
y = 2

# Arightmetische Operatoren
print(x + y)    # Addition --> 9
print(x - y)    # Subtraktion --> 5
print(x * y)    # Multiplikation --> 14
print(x / y)    # Division --> 3.5
print(x % y)    # Modulo --> 1
print(x ** y)   # Potenz --> 49
print(x // y)   # Division ohne Rest --> 3
```

```
# Vergleichsoperatoren
print(x == y)   # gleich --> False
print(x != y)   # ungleich --> True
print(x > y)    # größer --> True
print(x < y)    # kleiner --> False
print(x >= y)   # größer gleich --> True
print(x <= y)   # kleiner gleich --> False

# Logische Operatoren
print(x > 3 and x < 10) # UND --> True
print(x > 3 or x < 4)   # ODER --> True
print(not(x > 10))      # NICHT --> False
```

Auswahl

- **einseitig**

```
x = 3
if x == 3:
    print("stimmt")
```

- **zweiseitig**

```
x = 3
if x == 4:
    print("stimmt")
else:
    print("stimmt nicht")
```

- **mehrseitig**

```
age = 42
if age > 130:
    print("wahrscheinlich tot")
elif age > 65:
    print("Rente")
elif age >= 18:
    print("Erwachsen")
else:
    print("Kind")
```

Schleife

> Wiederholungen bzw. Schleifen sind eine oder mehrere Aktionen, die so lange wiederholt ausgeführt werden, wie eine gegebene Bedingung wahr ist. Je nach Anwendung gibt es verschiedene Arten von Schleifen:
>
> - **for-Schleife**: Wenn die Anzahl an Schleifendurchläufen vorher bekannt ist
> - **while-Schleife**: Wenn die Anzahl an Schleifendurchläufen vorher nicht bekannt ist oder es keine numerische Laufvariable gibt

- **for-Schleife**

```
# Liste iterieren
fruits = ["apple", "banana", "cherry"]
for x in fruits:
    print(x)

# String iterieren
for x in "banana":
    print(x)

# Range iterieren
for x in range(6):
    print(x)
for x in range(2, 6):
    print(x)
for x in range(2, 10, 3):
    print(x)
```

- **while-Schleife**

```
# equivalent zu for-Schleife
x = 0
while x < 6:
    print(x)
    x += 1

# Endlosschleife
while True:
    print("hi")

# Praxisbeispiel
passwort = "sicher123!"
eingabe = input("Passwort: ")
while eingabe != passwort:
    print("Falsches Passwort!")
    eingabe = input("Passwort: ")
print("Eingeloggt!")
```

Funktionen

> Eine **Funktion** ist ein Codeblock, der eine bestimmte Aufgabe erledigt. Zum Ausführen des Codeblocks muss die Funktion aufgerufen werden. **Parameter** sind Werte, die der Funktion beim Aufruf übergeben werden. Eine Methode kann einen Wert (**Rückgabewert**) beim Aufruf zurückgeben.

- Funktion

```
def sageHallo():
    print("Hallo!")

sageHallo()                         # Hallo!
sageHallo()                         # Hallo!
```

- Funktion mit **Parameter**

```
def sageHallo(name):
    print("Hallo, " + name)

sageHallo("Adam")                   # Hallo, Adam
sageHallo("Eva")                    # Hallo, Eva

def sageHallo(name, land = "Deutschland", alter):
    print("Hallo " + name + " (" + alter + ") aus " + land)

sageHallo("Adam", 31)               # Hallo Adam (31) aus Deutschland
sageHallo("Eva", "USA", 42)         # Hallo Eva (42) aus USA
```

- Funktion mit **Rückgabewert**

```
def quadrat(n):
    return n * n

print(quadrat(-5))                  # 25
print(quadrat(quadrat(2)))          # 16
```

Liste

> Eine **Liste** enthält mehrere Elemente.

```
# Liste initialisieren
cars = ["Ford", "Volvo", "BMW"]

# Element ausgeben
print(cars[0])      # Ford

# Element ändern
cars[1] = "Toyota"
```

```
print(cars)         # ['Ford', 'Toyota', 'BMW']

# Länge der Liste
print(len(cars))    # 3

# Liste durchlaufen
for i in range(len(cars)):
    print(car)
for car in cars:
    print(car)
for i, car in enumerate(cars):
    print(i, ":", car)

# Element hinzufügen / entfernen
cars.append("Honda")
cars.remove("BMW")  # Element content
cars.pop(2)         # Element index
```

Strings

```
# String
name = "Elon Musk"
poem = """Roses are red,
Violets are blue."""

# String as Array
name = "Elon Musk"
print(name[2])      # o
print(name[3:6])    # n M
print(name[:4])     # Elon
print(name[2:])     # on Musk

# Looping through a String
name = "Elon Musk"
for x in name:
    print(x)

for x in range(len(name)):
    print(name[x])

# Check String
name = "Elon Musk"
if "lo" in name:
    print("Yes")
elif "on" not in name:
    print("Maybe")
else:
    print("No")

# Modify String
name = "Elon Musk"
```

```
print(name.upper())             # ELON MUSK
print(name.lower())             # elon musk
print(name.replace("o", "a"))   # Elan Musk"

data = ";;Elon Musk;"
print(data.strip(";"))          # Elon Musk

letters = "a,b,c"
print(letters.split(","))       # ["a", "b", "c"]

# Format String
age = 27
print(f"I am {age} years old.") # I am 27 years old.
print(f"The exam had an average of {2.316412:.2f}!") # The exam had an average of
2.32!
```

Graphical User Interface (GUI) mit TKinter

```
# libraries
from tkinter import *

# frame
frame = Tk()
frame.title("CalculatorGUI")
frame.geometry("600x400")

# components
lbEn1 = Label(
    frame,
    text = "Zahl 1:",
    font = ("Arial", 15)
)
lbEn1.place(x = 20, y = 20)

en1 = Entry(
    frame,
    font = ("Arial", 15),
    width = 20,
    justify = "center",
    fg = "green",
    bg = "#ccc"
)
en1.place(x = 120, y = 20)

lbEn2 = Label(
    frame,
    text = "Zahl 1:",
    font = ("Arial", 15)
)
lbEn2.place(x = 20, y = 60)
```

```
en2 = Entry(
    frame,
    font = ("Arial", 15),
    width = 20,
    justify = "center",
    fg = "green",
    bg = "#ccc"
)
en2.place(x = 120, y = 60)

lbResultText = Label(
    frame,
    text = "Ergebnis:",
    font = ("Arial", 15)
)
lbResultText.place(x = 20, y = 300)

lbResult = Label(
    frame,
    font = ("Arial", 15),
    width=20,
    fg = "#ccc",
    bg = "green"
)
lbResult.place(x = 120, y = 300)

def add():
    n1 = float(en1.get())
    n2 = float(en2.get())
    result = n1 + n2
    lbResult.config(text=result)

btnAdd = Button(
    frame,
    text = "+",
    bg = "#add8e6",
    fg = "#000033",
    font = ("Arial", 15),
    command = add
)
btnAdd.place(x=20, y=180)

# run
frame.mainloop()
```