# Programmieren mit Audio - Schüler PDF

January 19, 2024

## 1 Lernziele

Ziel dieser Übung ist es, erste allgemeine Einblicke in die Programmierung zu bekommen. Sie sollen hierbei aber nicht nur irgendwelche Buchstaben und Zahlen auf der Konsole ausgeben, sondern im Verlaufe der Übung auch Töne und Musik abspielen lassen.

Nach der Bearbeitung sollten Sie folgende Kenntnisse erlangt haben:

- Sie können Dinge auf der Konsole ausgeben lassen.
- Sie kennen die grundlegenden Bausteine, mit denen man den Ablauf eines Programms steuern kann.
- Sie können mithilfe der tone() und tones() Funktionen beliebig viele Töne abspielen lassen.
- Sie können eine ABC-Notation so verändern, das sie von der play() Funktion abgespielt werden kann.

# 2 Variablen

In Variablen werden Informationen abgespeichert.

Listing 1: Variablen in Python

```
a = 10              a wird der Wert 10 zugewiesen.
b = 230             b wird der Wert 230 zugewiesen.
c = a               c wird der Wert von a zugewiesen.
wort = 'Hallo'      wort wird der Text Hallo zugewiesen.
```

Variablen, die ganze Zahlen speichern, heißen integer und Variablen, die einzelne Buchstaben oder ganze Texte speichern, heißen Strings. Bei Strings ist zu beachten, dass sie mit am Anfang und am Ende mit einem ' oder einem " gekennzeichnet werden.

## 2.1 Rechnen mit Variablen

Enthalten Variablen Zahlen, dann kann mit ihnen gerechnet werden.

Listing 2: Rechnen mit Variablen

```
a = 180
b = 90

summe          = a + b  -> Addition
differenz      = a - b  -> Subtraktion
produkt        = a * b  -> Multiplikation
quotient       = a / b  -> Division
divisionsrest = a % b  -> Der Rest nach einer Division
```

## 2.2 Ausgeben von Variablen

Will man Variablen in der Konsole ausgeben lassen, geht das wie folgt:

Listing 3: Ausgeben von Variablen

```
a = 120
b = 80
c = a + b

print(a)          -> gibt 120 auf der Konsole aus
print(c)          -> gibt 200 auf der Konsole aus
print(a + b)      -> gibt auch 200 auf der Konsole aus
```

# 3 Schleifen

Schleifen sind Codeblöcke, die man beliebig oft wiederholen lassen kann. Obwohl es mehrere Schleifen Varianten gibt, werden wir uns hier nur auf die for Schleife konzentrieren.

Der folgende Code beschreibt eine for Schleife, die von 0 bis 10 hochzählt und die jeweilige Zahl auf der Konsole ausgibt.

Listing 4: Aufbau der for Schleife

```
for x in range(0, 11):
    print(x)
```

Listing 5: Konsolenausgabe

```
0
1
2
3
4
5
6
7
8
9
10
```

Wichtig ist zu beachten, dass die hintere Zahl (im Beispiel 11) immer einen größer sein muss als die Zahl, zu der man zählen will.

# 4 Arrays

Ein Array ist eine Liste, die Dinge wie Zahlen oder Texte abspeichern kann.

Listing 6: Gemischte Arrays

```
integer_array      =   [1, 2, 3, 4, 5]
string_array       =   ['eins', 'zwei', 'drei']
gemischtes_array =   [1,, 'zwei', 'C', 'X']
```

Man kann auf ein Array auch zugreifen oder es verändern. Dabei muss man aber beachten, das Arrays nicht bei 1 anfangen, sondern bei 0.

Listing 7: Arbeiten mit Arrays

```
liste = [23, 83, 24, 35, 97]
         0   1   2   3   4

a = liste[2]       -> a bekommt den Wert 24 zugewiesen
liste.append(100) -> Die Zahl 100 wird angehaengt
liste[0] = 10      -> Wert an Stelle 0(23) wird zu 10
del liste[0]       -> Wert an der Stelle 0 wird geloescht
```

Will man ein Array mithilfe einer for Schleife durchgehen und die Werte der Liste ausgeben, geht das wie folgt:

Listing 8: Werte eines Arrays ausgeben

```
liste = [23, 83, 24, 35, 97]
         0   1   2   3   4

for x in range(0, len(liste)):
    print(liste[x])

Konsolenausgabe:
23
83
24
35
97
```

Hierbei verwendet man für die hintere Zahl die len() Funktion. Man übergibt der Funktion die Liste und man bekommt die Länge der Liste als integer zurück (mehr zu Funktionen später).

# 5 If/Else

Muss innerhalb vom Code eine Entscheidung getroffen werden, bildet man das oft mit einer If/Else Verzweigung ab. Je nachdem wie die Ausgangssituation ist, wird dann entschieden, wie es weitergehen soll.

Als Beispiel wollen wir herausfinden, ob die Variable a größer als 20 ist.

Listing 9: a größer 20

```
a = 10
if a > 20:
    print('a ist groesser als 20')
```

Dazu wollen wir jetzt aber auch etwas ausgeben, wenn a kleiner als 20 ist.

Listing 10: a größer/kleiner 20

```
a = 10
if a > 20:
    print('a ist groesser als 20')
else:
    print('a ist kleiner als 20')
```

Wollen wir jetzt auch noch etwas spezielles ausgeben, wenn a genau 20 ist, geht das auch.

Listing 11: a größer/kleiner/gleich 20

```
a = 10
if a > 20:
    print('a ist groesser als 20')
elif a == 20:
    print('a ist genau 20')
else:
    print('a ist kleiner als 20')
```

# 6 Funktionen

Auch wenn wir Funktionen in dieser Übung eher indirekt benutzen, sollten Sie einen groben Überblick darüber bekommen, was sie sind und wie sie funktionieren.

Funktionen sind Codeblöcke, die Sie in ihrem Code einmal definieren und dann an verschiedenen Stellen immer wieder aufrufen und verwenden können.

Eine Funktion hat den allgemeinen Aufbau:

Listing 12: Aufbau einer Funktion

```
def funktionsname(eingabeparameter 1, ...):
    code....
    return rueckgabeparameter 1, ...
```

Zu beachten ist, dass eine Funktion beliebig viele Eingab- und Rückgabeparameter haben kann, das bedeutet, dass sie auch komplett ohne auskommt.

Als Beispiel betrachten wir eine Funktion, die 3 Eingabeparameter addiert und das Ergebnis zurückgibt.

Listing 13: Funktionsbeispiel

```
def summe_aus_drei(a, b, c):
    ergebnis = a + b + c
    return ergebnis


x = 10
y = 20
z = 30
summe = summe_aus_drei(x, y, z)
print(summe) -> auf der Konsole wird 60 ausgegeben
```

Es ist zu beachten, dass die Funktion erste nach ihrer eigenen Definition aufgerufen werden kann.

# 7 Musik und Töne

Der Webeditor benutzt im Hintergrund die Bibliothek musical.js um die Töne sowie die ABC Notation abzuspielen.

## 7.1 Die tone() Funktion

Mithilfe der tone() Funktion lässt sich ein Ton oder mehrere Töne gleichzeitig abspielen.

Listing 14: Einzelnen Ton abspielen

```
tone('C')
```

Listing 15: Mehrere Töne gleichzeitig abspielen

```
tone('C')
tone('D')
tone('C')
```

## 7.2 Die tones() Funktion

Mithilfe der tones() Funktion lassen sich Töne nacheinander abspielen. Dafür müssen die Töne allerdings zu erst in ein Array abgespeichert werden.

Listing 16: Töne nacheinander abspielen lassen

```
toene = ['C', 'D', 'E']
tones(toene)
```

## 7.3 Die play() Funktion

Mithilfe der play() Funktion lässt sich eine beliebige ABC-Notation abspielen, vorausgesetzt man verändert die Notation so, das sie als String dargestellt wird.

Als Beispiel nutzen wir die Notation von Speed the Plough zu finden, auf der dieser Seite.

Listing 17: ABC Notation

```
X:1
T:Speed the Plough
M:4/4
C:Trad.
K:G
|:GABc dedB|dedB dedB|c2ec B2dB|c2A2 A2BA|
  GABc dedB|dedB dedB|c2ec B2dB|A2F2 G4:|
|:g2gf gdBd|g2f2 e2d2|c2ec B2dB|c2A2 A2df|
  g2gf g2Bd|g2f2 e2d2|c2ec B2dB|A2F2 G4:|
```

Listing 18: ABC Notation als String

```
"X:1 \n"+
"T:Speed the Plough \n"+
"M:4/4 \n"+
"C:Trad. \n"+
"K:G \n"+
"|:GABc dedB|dedB dedB|c2ec B2dB|c2A2 A2BA| \n"+
"  GABc dedB|dedB dedB|c2ec B2dB|A2F2 G4:| \n"+
"|:g2gf gdBd|g2f2 e2d2|c2ec B2dB|c2A2 A2df| \n"+
"  g2gf g2Bd|g2f2 e2d2|c2ec B2dB|A2F2 G4:|"
```

Listing 19: ABC Notation mit play() abspielen lassen

```
play(
"X:1 \n"+
"T:Speed the Plough \n"+
"M:4/4 \n"+
"C:Trad. \n"+
"K:G \n"+
"|:GABc dedB|dedB dedB|c2ec B2dB|c2A2 A2BA| \n"+
"  GABc dedB|dedB dedB|c2ec B2dB|A2F2 G4:| \n"+
"|:g2gf gdBd|g2f2 e2d2|c2ec B2dB|c2A2 A2df| \n"+
"  g2gf g2Bd|g2f2 e2d2|c2ec B2dB|A2F2 G4:|")
```

# 8 Aufgaben

Sie finden die Aufgaben/Levels auch im Webeditor.

## 8.1 Aufgabe 0 - Variablen, Arithmetik und das Ausgeben auf der Konsole

Erstellen Sie eine Variable a und eine Variable b, a soll den Wert 10 haben und b den Wert 23. Erstellen Sie zudem noch eine Variable c, die als Wert das Ergebnis der Addition von a und b enthält und geben Sie diese auf der Konsole aus.

## 8.2 Aufgabe 1 - If/Else Verzweigung

Erstellen Sie eine Variable a mit dem Wert 10. Schreiben Sie jetzt eine If/Else Verzweigung, die überprüft, ob a kleiner/gleich 10 ist oder ob a größer als 10 ist. Ist a größer, dann lassen Sie "Größer" auf der Konsole ausgeben, ist a kleiner oder gleich 10, dann lassen Sie "Kleiner" auf der Konsole ausgeben (auf der Konsole soll hierbei der Wert, den Sie a gegeben haben, ausgegeben werden). Testen Sie ihr Programm, indem Sie den Wert der Variable a verändern und das Programm erneut starten.

## 8.3 Aufgabe 2 - Schleifen

Erstellen Sie eine Variable a mit dem Wert 0. Danach erstellen Sie eine for oder while Schleife, die von 0 bis 10 geht. Bei jedem Schleifendurchlauf soll der Wert des aktuellen Durchlaufs auf a oben drauf addiert werden. Wenn die Schleife durchgelaufen ist, soll a ausgegeben werden. Als Tipp: Bei einer Schleife mit vier Durchläufen (0-3) wäre der Wert der Variable a am Ende 6.

## 8.4 Aufgabe 3 - Töne abspielen

Erstellen Sie wie in Aufgabe 1 eine Variable a mit dem Wert 10 und eine Verzweigung, die überprüft, ob a kleiner/gleich oder größer 10 ist. Diesmal soll aber nichts auf der Konsole ausgeben werden. Wenn a kleiner/gleich 10 ist, soll mithilfe der tone() Funktion der Ton "C" ausgegeben werden, ist a größer als 10 soll der Ton "D" ausgegeben werden. Überprüfen Sie ihren Code wie in Aufgaben 1, indem Sie den Wert der Variable a verändern und das Programm erneut starten.

## 8.5 Aufgabe 4 - C-Dur-Tonleiter erstellen

Sie haben folgendes Array gegeben: töne = [”C”, ”C”, ”E”, ”D”, ”F”, ”E”, ”G”, ”F”, ”C”, ”G”, ”H”, ”A”, ”H”, ”H”, ”C”, ”C”]. Erstellen Sie jetzt ein leeres Array namens C_Dur_Tonleiter. Benutzen Sie eine Schleife und eine If/Else Verzweigung, um das Array so zu füllen, das es eine C-Dur-Tonleiter (C, D, E, F, G, A, H, C) darstellt und lassen Sie es mithilfe der tones() Funktion abspielen. Kleiner Tipp: Sie müssen jedes zweite Element aus dem töne Array in das C-Dur-Tonleiter Array eintragen lassen. (Beim Abspielen von Ton ”H” kommt es zu einem Fehlerton).

## 8.6 Aufgabe 5 - ABC Notation

Gehen Sie auf diese Seite und suchen Sie sich eine ABC-Notation heraus. Kopieren Sie die Notation in den Editor und wandeln Sie sie so um das die play() Funktion sie abspielen kann und lassen Sie die Notation abspielen.