

Programmieren mit Audio - Lehrer PDF

January 19, 2024

1 Installation

Folgen Sie diesen Schritten um das Projekt zu installieren und den Webeditor aufzurufen:

- Rufen Sie die Gitlab Seite der Projektgruppe auf (<https://gitlab.technik-emden.de/da6338/projektgruppe-23-link>).
- Wählen Sie den Branch "musik-sounds" aus.
- Drücken Sie auf den blauen Button "Code" und wählen unter "Download source code" das gewünschte Kompressionsformat aus. Es sollte der Download des source codes beginnen.
- Nach dem Download entpacken Sie die Datei an einen gewünschten Zielort.
- Öffnen Sie den Ordner "projektgruppe-23-link-musik-sounds" und folgen Sie dem Pfad "projektgruppe-23-link-musik-sounds" → "Programmieren mit Audio" → "Source-Code". Sie sollten jetzt in einem Verzeichnis sein, das unter anderem eine HTML-Datei namens "code-editor" sowie ein Verzeichnis namens "levels" enthalten sollte.
- Öffnen Sie in diesem Ordner ein Terminal. Gegebenenfalls können Sie nicht direkt im Ordner ein Terminal öffnen, Sie sollten dann ein allgemeines Terminal Fenster öffnen und in den Ordner namens "Source-Code" navigieren.
- Sind Sie im Ordner angekommen, geben Sie "python -m http.server 8080" ein um den Server des Webeditors auf Port 8080 zu öffnen.
- Öffnen Sie jetzt einen beliebigen Browser und geben Sie in die Adressleiste <http://localhost:8080/code-editor.html>
- Es sollte sich der Webeditor des Projekts öffnen. Ab jetzt können die Aufgaben bearbeitet werden.

2 Musterlösungen

2.1 Musterlösung 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.

Listing 1: Lösung: Aufgabe 0

```
a = 10  
b = 23  
c = a + b
```

```
print(c)
```

```
Konsole  
33
```

2.2 Musterlösung 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.

Listing 2: Lösung: Aufgabe 1

```
a = 9
if(a <= 10):
    print("Kleiner")
else:
    print("Groesser")

Konsole
Kleiner

a = 10
if(a <= 10):
    print("Kleiner")
else:
    print("Groesser")

Konsole
Kleiner

a = 11
if(a <= 10):
    print("Kleiner")
else:
    print("Groesser")

Konsole
Groesser
```

2.3 Musterlösung 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.

Listing 3: Lösung: Aufgabe 0

```
a = 0
for x in range(0, 11):
    a = a + x

print(a)
```

Konsole
55

2.4 Musterlösung 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 ausgegeben 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.

Listing 4: Lösung: Aufgabe 0

```
a = 9
if(a <= 10):
    tone("C")
else:
    tone("D")

a = 10
if(a <= 10):
    tone("C")
else:
    tone("D")

a = 11
if(a <= 10):
    tone("C")
else:
    tone("D")
```

2.5 Musterlösung 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).

Listing 5: Lösung: Aufgabe 0

```
toene = ["C", "E", "D", "F", "E", "H", "F", "G",  
         "G", "E", "A", "C", "H", "C", "C"]  
C_Dur_Tonleiter = []  
  
for x in range(0, len(toene)):  
    if(x % 2 == 0):  
        C_Dur_Tonleiter.append(toene[x])  
  
tones(C_Dur_Tonleiter)
```

2.6 Musterlösung 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.

Listing 6: Lösung: Aufgabe 0

```
ABC Notation
(Speed the Plough, Startseite(https://abcnotation.com/)):

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:|

play("X:1 \n"+
"T:Speed \nthe \nPlough \n"+
"M:4/4 \n"+
"C:Trad. \n"+
"K:G \n"+
" |:GABc\ndedB|dedB\ndedB|c2ec \nB2dB|c2A2 \nA2BA| \n"+
" \nGABc\ndedB|dedB\ndedB|c2ec \nB2dB|A2F2 \nG4:| \n"+
" |:g2gf \ngdBd|g2f2 \ne2d2|c2ec \nB2dB|c2A2 \nA2df| \n"+
" \ng2gf \ng2Bd|g2f2 \ne2d2|c2ec \nB2dB|A2F2 \nG4:| ")
```