

Programmieren mit Python

Teil 4: Die While-Schleife

Dr. Aaron Kunert

aaron.kunert@salemkolleg.de

26. Oktober 2022

Die While-Schleife

Wie die For-Schleife nur abstrakter und open-end

Problemstellung

Lies immer wieder eine Zahl von der Konsole ein. Höre auf, wenn diese Zahl 7 ist.

Wie macht man das?

Lösung

```
x = 0
```

```
while x != 7:  
    x = input("Gib eine Zahl an: ")  
    x = int(x)
```

```
print("Fertig!")
```

Struktur der while-Schleife

```
while Bedingung:
```

```
    Codezeile 1
```

```
    Codezeile 2
```

```
    ⋮
```

```
Code, der nicht mehr Teil der Schleife ist
```

Wie funktioniert's?

Die Schleife wird solange ausgeführt, solange die *Bedingung* **True** ergibt. Nach jedem Durchgang wird der Ausdruck der *Bedingung* neu ausgewertet. Ist die Bedingung **False** wird der Code unterhalb des Schleifenblocks ausgeführt.

Achtung Endlosschleife

Man sollte immer darauf achten, dass die Bedingung in der `while`-Schleife auch wirklich irgendwann `False` wird. Ansonsten bleibt das Programm in einer *Endlosschleife* gefangen.

Ersetze eine `for`-Schleife durch eine `while`-Schleife

Schreibe ein Skript, das alle Zahlen von 1 bis 100 auf der Konsole ausgibt. Verwende eine `While`-Schleife.

Lösung

```
k = 1
while k <= 100:
    print(k)
    k += 1
```

Quizfrage

Schreibe ein Programm, dass solange nach einer Hauptstadt Deiner Wahl fragt, bis die richtige Antwort eingegeben wird.

Beispiel

Was ist die Hauptstadt von Frankreich?

Darauf hin soll entsprechend der Antwort folgendes Feedback auf der Konsole erscheinen:

Das war richtig!

bzw.

Das war falsch! Versuch's gleich nochmal

Quizfrage

```
answer = input("Was ist die Hauptstadt von Frankreich?")
while answer != "Paris":
    print("Das war leider falsch, versuch es gleich nochmal")
    answer = input("")
print("Das war richtig!")
```

Ratespiel

Definiere eine positive ganze Zahl `number_to_guess`. Der User kann nun wiederholt eine Zahl eingeben. Das Spiel endet, wenn die eingegebene Zahl mit `number_to_guess` übereinstimmt. Andernfalls wird auf der Konsole beispielsweise ausgegeben:

```
Sorry, Deine eingegebene Zahl war zu klein, versuche es nochmal:
```

Zusatz 1:

Am Ende soll die Anzahl der Versuche angegeben werden.

Zusatz 2:

Google, wie Python die Zahl `number_to_guess` zufällig erzeugen kann (das verbessert das Gameplay).

Ratespiel ohne Zusätze

```
number_to_guess = 512
guess = input("Rate meine Zahl: ")
guess = int(guess)

while guess != number_to_guess:
    if guess < number_to_guess:
        print("Deine Zahl war zu klein")
    else:
        print("Deine Zahl war zu groß")
    guess = input("Versuch's nochmal: ")
    guess = int(guess)

print("Du hast gewonnen")
```

Ratespiel mit Zusatz 1

```
number_to_guess = 512
guess = input("Rate meine Zahl: ")
guess = int(guess)
counter = 1

while guess != number_to_guess:
    if guess < number_to_guess:
        print("Deine Zahl war zu klein")
    else:
        print("Deine Zahl war zu groß")
    guess = input("Versuch's nochmal: ")
    guess = int(guess)
    counter = counter + 1

print(f"Du hast nach {counter} Versuchen gewonnen")
```

Ratespiel mit Zusatz 2

```
import random

number_to_guess = random.randint(1, 1000)
guess = input("Rate meine Zahl: ")
guess = int(guess)
# ... usw.
```
