Dr. Aaron Kunert
aaron.kunert@salemkolleg.de

Einführung in Python
Lösung zu Blatt 2

11.11.2021

Lösung zu Aufgabe 1

```python
example_list = [2, 7, 5, -1, 4, 12, 3, -19, 16]

# We add all items up, every second item is multiplied by 2 (i.e. weighting)
weighted_sum = 0
length = 0

for (index, number) in enumerate(example_list):
    if index % 2 == 0:
        # Remember: The first item in the list has index 0
        weighted_sum += number
        length += 1
    else:
        weighted_sum += 2*number
        length += 2

result = weighted_sum / length
print(f"Der gewichtete Durchschnitt ist {result}")\\
```

Lösung zu Aufgabe 2

```python
# Man verwendet statt ganzen Euros lieber Cents
# Der Unterstrich innerhalb einer Zahl dient nur zur Übersichtlichkeit
# und kann ignoriert werden (z.B. 50_00 = 5000)

# Teil A
value = 3000_00
interest = 0.0028
rate = 50_00

for month in range(1, 12 * 15 + 1):
    value += rate
    value += interest * value
    value = round(value)
print(f"a) In Monat {month} beträgt der Wert {value/100}€")

# Teil B
month = 0
value = 3000_00

while interest * value < rate:
    month += 1
    value += rate
    value += interest * value
    value = round(value)

print(f"b) Nach {month} Monaten gibt der Zinssatz mehr als die Sparrate")
```

```python
# Teil C
value = 3000_00
alt_value = 3000_00
alt_rate = 10_00
alt_interest = 0.009
month = 0

while alt_value <= value:
    month += 1
    value += rate
    value += interest * value
    value = round(value)
    alt_value += alt_rate
    alt_value += alt_interest * alt_value
    alt_value = round(alt_value)

print(f"c) Nach {month} Monaten ist das zweite Modell besser")

value = 3000_00
alt_value = 3000_00

for month in range(1, 12 * 20 + 1):
    value += rate
    value += interest * value
    value = round(value)
    alt_value += alt_rate
    alt_value += alt_interest * alt_value
    alt_value = round(alt_value)

print(f"d) Nach 20 Jahren beträgt die Differenz gleich")
print(f"{alt_value/100} - {value/100} = {(alt_value - value)/100}")
```

Lösung zu Aufgabe 3

```
my_name = "Aaron"
friends_name = "Tom"

example = ["Max", "Lara", "Kathrin", "Aaron", "Tom", "Sebastian"]     # should return True
example_2 = ["Max", "Lara", "Kathrin", "Tom", "Aaron", "Sebastian"]   # should return True
example_3 = ["Max", "Lara", "Kathrin", "Tom", "Sebastian", "Aaron"]   # should return False

my_list = example

length = len(my_list)

for (index, name) in enumerate(my_list):
    # if we are at the beginning of the list, there is nothing to check
    if index == 0:
        continue
    # Check if you are at the current position and your friend at the previous position
    if name == my_name and my_list[index - 1] == friends_name:
        print("Ja, die beiden Namen kommen hintereinander")
        break
    # Check if your friend is at the current position and you are at the previous position
    if name == friends_name and my_list[index - 1] == my_name:
        print("Ja, die beiden Namen kommen hintereinander")
        break
else:
    print("Nein, die beiden Namen kommen nicht hintereinander")
```

Lösung zu Aufgabe 4

```
counter = 1
n = 1000000000

while n != 1:
    counter += 1
    if n % 2 == 0:
        n = n//2
    else:
        n = 3*n + 1

print(f"Die Folge ist nach {counter} Folgengliedern zu Ende")
```