# Einführung: Schleifen

In der ersten Sitzung zu Python habt ihr euch mit Datentypen, Input / Output und Variablen beschäftigt. Heute geht es darum, **Schleifen** in Programmen zu verwenden und zu verstehen. Ihr lernt sowohl **FOR-** als auch **WHILE-Schleifen** kennen.

## 1. Aufgabe

a. Gebt den Python Code unten in euren Editor ein.
   Klickt auf "Run" (Play Symbol) und schaut euch das Ergebnis in der Konsole an.
b. Lest euch den Quellcode und die Kommentare (mit # gekennzeichnet) durch.
c. Passt den Code so an, dass er 3 Mal "Python ist aber auch eine Schlange." ausgibt.
d. Eine Erklärung, was hier passiert, findet ihr unten. Eine mögliche Lösung auf der nächsten Seite.

```
1  #Schleife, die Text ausgibt
2  for i in range(10):
3      print('Python ist eine tolle Programmiersprache.')
4  #Die Variable i nimmt entsprechend des Wertebereichs
   10 Werte an
5  #Für jedes i wird der Text 1 Mal ausgeführt
6  #Nach 10x wird die Schleife beendet
```

## Schleifen

Um den Text zu wiederholen, habt ihr eine **Schleife** verwendet:

- for i in range (3)
  print('Halli, Hallo')

Schleifen braucht ihr immer dann, wenn ihr Dinge **wiederholen** möchtet. Ihr könnt Texte wiederholen, ebenso wie Zahlen oder beliebige Anweisungen. Wichtig ist, dass eure Schleife immer eine **Abbruchbedingung** hat. Ansonsten habt ihr eine **Endlosschleife** und das Programm endet niemals.

So haben wir das auch gemacht:
In Zeile 2 (siehe Code oben) tragen wir ein, wie oft die Schleife ausgeführt wird. Das ist unsere Abbruchbedingung.

In Zeile 3 stehen die Anweisungen, die ausgeführt werden. In dem Beispiel oben wird der Text 10 Mal in der Konsole ausgegeben, danach wird die Schleife automatisch beendet.

Es gibt verschiedene Arten von Schleifen (später mehr dazu). Hier habt ihr eine sogenannte **FOR-Schleife** verwendet.

## Python Merkzettel

Nehmt euren **Python Merkzettel** und füllt die Box zum Thema **Schleifen** aus. Lasst noch etwas Platz, um die Notizen später zu ergänzen.

Hier findet ihr den Merkzettel. Druckt ihn entweder aus und füllt ihn handschriftlich oder bearbeitet das pdf-Dokument direkt an eurem Rechner. Vergesst dann bitte nicht, das Dokument zu speichern und sinnvoll abzulegen. (Alle Lehrkräfte finden den Merkzettel unter "Vorbereitung" auf der App Camps Plattform.)

## Mögliche Lösung - Aufgabe 1

```
1  #Schleife, die Text ausgibt
2  for i in range(3):
3    print('Python ist aber auch eine Schlange.')
4  #Die Variable i nimmt entsprechend des Wertebereichs 3 Werte an
5  #Für jeden Wert werden die Anweisungen in der Schleife durchlaufen –
   das heißt:
6  #1. Wert:print('Python ist aber auch eine Schlange.')
7  #2. Wert: print('Python ist aber auch eine Schlange.')
8  #3. Wert: print('Python ist aber auch eine Schlange.')
```

```
Python ist aber auch eine Schlange.
Python ist aber auch eine Schlange.
Python ist aber auch eine Schlange.
```

**Hinweis:** Solltet ihr mal etwas nicht verstehen, fragt einfach eure MitschülerInnen. Gemeinsam könnt ihr natürlich auch eure Lehrkraft um Hilfe bitten.

## 2. Aufgabe

**Benutzt eine Schleife, um Zahlenreihen auszugeben.**

a. Gebt den Python Code unten in euren Editor ein und klickt auf “Run”.
b. Lest euch den Quellcode und die Kommentare (mit # gekennzeichnet) durch.
c. Passt den Code so an, dass nicht 0 bis 9 ausgegeben werden, sondern **1 bis 10**.
d. Erstellt eine weitere **FOR-Schleife, die die 7er Reihe ausgibt. Also: 7, 14, 21, … 70.**

Probiert es aus: Eine mögliche Lösung findet ihr auf der nächsten Seite.

```
1  #Schleife, die die Zahlen 0 bis 9 ausgibt
2  for i in range(10):
3    print(i)
4
5  #Die Anweisungen, die innerhalb der Schleife ausgeführt werden
   sollen, müssen je nach Editor einen Tab (alternativ drei oder vier
   Leerzeichen) eingerückt werden.
6  #Repl.it macht das automatisch!
```

## Mögliche Lösung - Aufgabe 2

```
#Schleife, die die Zahlen 1 bis 10 ausgibt
for i in range(10):
  print(i+1)

#Fügt eine leere Zeile ein
print()

#Schleife, die die 7er Reihe ausgibt
for i in range(10):
  print((i+1)*7)
#i startet bei 0 und die Schleife wird insgesamt 10 Mal durchlaufen.
#Jedes Mal wird i um 1 erhöht und dann mit 7 multipliziert.
Beachtet! Die Klammersetzung ist hier wichtig, da ansonsten Punkt-
vor Strichrechnung gilt
#Hier seht ihr, was das Programm macht:
# Für i = 0 -> ((0+1)*7)
#Das Ergebnis wird berechnet und durch den print-Befehl in der
Konsole ausgegeben
#Nachdem die Anweisungen ausgeführt wurden, wird i um 1 erhöht
# Für i = 1 -> ((1+1)*7)
# Für i = 2 -> ((2+1)*7)
#...
# Für i = 9 -> ((9+1)*7)
```

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

7
14
21
28
35
42
49
56
63
70
>
```

**Hinweis:** In der Programmierung fängt man bei 0 an zu zählen. Das kann schon mal verwirrend sein, denn so geht die Range von 10 nur von 0 bis 9. Das sind genau 10 Zahlen - zählt mal nach. :)

## 3. Aufgabe

**Benutzt eine Schleife, um einen beliebigen Text beliebig oft auszugeben.**

a. Befolgt dazu die Schritt für Schritt Anleitung unten im Editor.
b. Auf der nächsten Seite findet ihr eine mögliche Lösung. Wenn ihr die Schritte korrekt ausgeführt habt, solltet ihr die unten stehende Ausgabe erhalten:

```
# 1. Schritt: Abfrage, welcher Text wiederholt werden
soll (Tipp: input())

# 2. Schritt: Abfrage, wie oft der Text wiederholt
werden soll (Tipp: input() und achtet auf den
Datentypen!)

# 3. Schritt: Ausgabe des in Schritt 1 eingegebenen
Textes – und zwar so oft wie in Schritt 2 definiert
(Tipp: FOR-Schleife und print())
```

```
Was soll wiederholt werden? Du bist toll!
Wie oft? 4
 Du bist toll!
 Du bist toll!
 Du bist toll!
 Du bist toll!
>
```

**Tipp:** Ihr könnt parallel Sitzung 1 öffnen/ansehen, um z.B. nachzuschauen, wie ihr input() einsetzen könnt.

## Mögliche Lösung - Aufgabe 3

```
1  #Abfrage, welcher Text wiederholt werden soll
2  text = input('Was soll wiederholt werden?')
3
4  #Abfrage, wie oft der Text wiederholt werden
   soll
5  anzahl = int(input('Wie oft?'))
6
7
8  #Ausgabe des in Schritt 1 eingegebenen Textes
   - und zwar so oft wie in Schritt 2 definiert
9  for i in range(anzahl):
10   print(text)
```

```
Was soll wiederholt werden? Du bist toll!
Wie oft? 4
 Du bist toll!
 Du bist toll!
 Du bist toll!
 Du bist toll!
```

## 4. Aufgabe

**Vergleich von FOR- und WHILE-Schleifen**

a. Lest euch die graue Infobox aufmerksam durch und ergänzt euren **Merkzettel**.
b. Versucht anschließend den Code aus der zuletzt gelösten Aufgabe **handschriftlich in eine Lösung** mit **WHILE-Schleife** umzuwandeln. So (oder ähnlich) sah unser Programm in der vorherigen Aufgabe aus:

```
text = input('Was soll wiederholt werden?')
anzahl = int(input('Wie oft?'))

for i in range(anzahl):
  print(text)
```

Ihr habt bereits gelernt: Schleifen braucht ihr immer dann, wenn ihr Dinge **wiederholen** möchtet.

Bisher habt ihr die **FOR-Schleife** kennengelernt. Formal ist diese folgendermaßen aufgebaut, rechts findet ihr außerdem ein konkretes Beispiel in Python:

```
for i in range ('Anzahl der Wiederholungen, bis zum Abbruch'):
    'Befolge diese Anweisungen'
```

```
1  for i in range(10):
2    print (i+1)
```

Alternativ könnt ihr auch eine WHILE-Schleife verwenden:

```
while i <= ('Anzahl der Wiederholungen, bis zum Abbruch'):
    'Befolge diese Anweisungen'
```

```
1  i = 1
2  while i <=10:
3      print (i)
4      i = i + 1
```

Bei der **WHILE-Schleife** müsst ihr in diesem Beispiel die **Variable i** initialisieren und in jedem Schleifendurchlauf erhöhen. Das bleibt euch bei der **FOR-Schleife** erspart. Die FOR-Schleife ist eine kompakte Form für einen Spezialfall der WHILE-Schleife. Nämlich für den Fall, dass die Anzahl der Schleifendurchläufe bekannt ist.

Als **Faustregel** gilt:
Wenn die **Anzahl der Durchläufe bekannt** oder abzählbar ist, könnt ihr eine **FOR-Schleife** verwenden. Diese ist in der Regel kompakter und wird bevorzugt verwendet. Unter appcamps.link/python3 findet ihr eine Erklärung (englisch) sowie weitere Details.
Wenn die Anzahl der Durchläufe **nicht bekannt** ist, wird eine **WHILE-Schleife** verwendet.

## 5. Aufgabe

**Benutzt eine WHILE-Schleife, um einen beliebigen Text beliebig oft auszugeben.**

a. Tippt eure handschriftliche Lösung zur Umwandlung der **FOR-** in eine **WHILE-Schleife** in den Editor ein. Klickt auf 'run'. Erhaltet ihr dieselbe Ausgabe, wie bei der Lösung mit einer FOR-Schleife?
b. Auf der nächsten Seite findet ihr eine mögliche Lösung. Lest euch den Code + Kommentare durch.

## Mögliche Lösung - Aufgabe 5

```
1  #Abfrage, welcher Text wiederholt werden soll
2  text = input('Was soll wiederholt werden?')
3
4  #Abfrage, wie oft der Text wiederholt werden soll
5  anzahl = int(input('Wie oft?'))
6
7  #Ausgabe des in Schritt 1 eingegebenen Textes – und zwar so oft
   wie in Schritt 2 definiert
8  i = 1
9  while i <= anzahl:
10   print(text)
11   i+=1
```

```
Was soll wiederholt werden? Du bist toll
Wie oft? 10
 Du bist toll
 Du bist toll
 Du bist toll
 Du bist toll
 Du bist toll
 Du bist toll
 Du bist toll
 Du bist toll
 Du bist toll
 Du bist toll
```

**Hinweis:** Beim Programmieren kann es sehr schnell passieren, dass man etwas vergisst. Zum Beispiel einen **Doppelpunkt** oder die **Gänsefüßchen**. Dann funktioniert das Programm nicht, obwohl der Inhalt eigentlich stimmt. Zum Glück gibt einem der **Editor eine Fehlermeldung** zurück. Diese sehen auf den ersten Blick zwar unverständlich aus, sind aber meist sehr hilfreich! Im Zweifel kann man damit im Internet nach einer Lösung suchen.

## 6. Aufgabe

**Vergleich von FOR- und WHILE-Schleifen**

a. Lest euch die graue Infobox aufmerksam durch und ergänzt euren **Merkzettel**.
b. Vielleicht kennt ihr dieses **Zeichen '!='** noch nicht. Unten findet ihr einen kleinen Exkurs mit Erklärung zu diesem und anderen sogenannten **logischen Vergleichsoperatoren**.
c. Überlegt euch anschließend mögliche **Anwendungsfälle** für eine **WHILE-Schleife**.

Es gibt verschiedene sogenannte **logische Vergleichsoperatoren**. Dazu gehört zum Beispiel **"kleiner als" (<)**, das kennt ihr aus dem Matheunterricht. Ein anderer logischer Vergleichsoperator ist **"ungleich" (!=)**. Mit einem logischen Vergleichsoperator könnt ihr einen Vergleich anstellen (z.B. von zwei Zahlen). Das Ergebnis des Vergleichs ist immer wahr oder falsch.

| Operator | Was macht der Operator? | Beispiel |
|---|---|---|
| == | Ist gleich | 1 == 1 (wahr) oder<br>'Hallo' == 'Halli' (falsch) |
| != | Ist ungleich | 1 != 1 (falsch) oder<br>'Hallo' != 'Halli' (wahr) |
| > | Ist größer als | 2 > 1 (wahr) oder<br>1 > 1 (falsch) oder<br>1 > 2 (falsch) |
| >= | Ist größer als oder gleich | 2 >= 1 (wahr) oder<br>1 >= 1 (wahr) oder<br>1 >= 2 (falsch) |
| < | Ist kleiner als | 1 < 2 (wahr) oder<br>1 < 1 (falsch) oder<br>2 < 1 (falsch) |
| <= | Ist kleiner als oder gleich | 1 <= 2 (wahr) oder<br>1 <= 1 (wahr) oder<br>2 <= 1 (falsch) |

## 7. Aufgabe

a. Lest euch den Quellcode und die Kommentare (mit # gekennzeichnet) durch.

b. Habt ihr euch ein ähnliches Beispiel überlegt? Wenn nicht, kein Problem. In der nächsten Sitzung sprechen wir noch einmal über WHILE-Schleifen mit Verzweigungen. Ihr könnt gespannt sein, was man damit schon alles machen kann.

```
#Eingabe
text = input("Bitte schreibe Du bist wundervoll!:")

#Solange die Eingabe nicht korrekt ist, frage erneut.
while text != "Du bist wundervoll!":
  text = input("Bitte schreibe Du bist wundervoll!:")

print("Danke. Du bist toll!")

#Erklärung:
# Der Benutzer wird um die Eingabe des Satzes 'Du bist
wundervoll!' gebeten. Solange die Eingabe des Benutzers nicht
exakt der geforderten entspricht (mit Groß- und Kleinschreibung,
Ausrufezeichen,...), wird der Benutzer weiterhin um die Eingabe
gebeten. Nur wenn die Eingabe korrekt ist, endet die Schleife und
das Programm gibt den Text 'Danke. Du bist toll!' aus. Da nicht
bekannt ist, wie viele Versuche der Benutzer braucht, um den Text
richtig einzugeben, wird klassischerweise eine while Schleife
verwendet.
```

```
Bitte schreibe Du bist wundervoll!:Du ist wundervoll
Bitte schreibe Du bist wundervoll!: Du bist wundervoll
Bitte schreibe Du bist wundervoll!: Du bist wundervoll!
Bitte schreibe Du bist wundervoll!:Du bist wundervoll!
Danke. Du bist toll!
```

## 1. Knobelaufgaben zu Schleifen

a. Schaut euch den Code an. Um Schleifen richtig zu schreiben, müsst ihr nachvollziehen können, wie Schleifen funktionieren.

```
zaehler = 0
ergebnis = 0
schleifendurchlaeufe = 0

while zaehler < 5:
  ergebnis = ergebnis + 1
  zaehler = zaehler + 1;
  schleifendurchlaeufe = schleifendurchlaeufe + 1;

print("Fertig! Ergebnis = ", ergebnis)
print("Schleifendurchläufe: ", schleifendurchlaeufe)
print("Zähler:", zaehler)
```

b. Geht die Schleifendurchläufe Schritt für Schritt durch, um folgende Fragen zu beantworten:

i. Welche Ausgabe erhaltet ihr, wenn ihr das Programm startet? Welchen Wert hat die Variable 'ergebnis'?

ii. Wie oft wird die Schleife durchlaufen? Welchen Wert hat die Variable 'schleifendurchlaeufe'?

c. Nutzt die Tabelle unten als Vorlage, um alle Zwischenergebnisse zu notieren. Schreibt sie ab und füllt sie handschriftlich aus. Auf der nächsten Seite findet ihr eine mögliche Lösung.

| Schleifendurchlauf | Ergebnis | Zaehler |
|---|---|---|
| 0 (zu Beginn) | | |
| 1 | | |
| 2 | | |
| ... | | |

## Lösung 1. Knobelaufgabe

a. Schaut euch die Tabelle unten an. Sieht eure Tabelle genauso aus oder habt ihr etwas vergessen?
   - Welchen Wert hat die Variable 'ergebnis' zum Schluss?
   - Welchen Wert hat die Variable 'schleifendurchlaeufe' zum Schluss?

b. Stimmt eure Lösung der Aufgabe mit der unten überein? Falls ja - super! Falls nicht - versucht nachzuvollziehen, wo der Fehler liegt. Tauscht euch eventuell mit euren MitschülerInnen aus.

Auf den nächsten Seiten findest du weitere Aufgaben zu Schleifen.

**Tipp:** Im Editor könnt ihr jederzeit mit einem 'print' - Befehl das Zwischenergebnis ausgeben lassen und so überprüfen, ob das Programm macht, was ihr möchtet. Unter appcamps.link/python4 findet ihr den Code zu der Aufgabe.

| Schleifendurchlauf | Ergebnis | Zaehler |
|---|---|---|
| 0 (zu Beginn) | 0 | 0 |
| 1 | 1 | 1 |
| 2 | 2 | 2 |
| 3 | 3 | 3 |
| 4 | 4 | 4 |
| 5 | 5 | 5 |

**Antwort:**

- Der Wert der Variable 'ergebnis' ist 5.
- Der Wert der Variable 'schleifendurchlaeufe' ist 5.

Die Schleife wird fünf Mal durchlaufen und bei jedem Mal wird sowohl beim Zähler, als auch beim Ergebnis 1 zum aktuellen Wert hinzuaddiert. Nach fünf Durchläufen ist die Bedingung, dass der Zähler kleiner als 5 sein muss nicht mehr erfüllt. Die Schleife wird also automatisch nicht mehr durchlaufen und der Code weiter unten (hier zwei print-Befehle) wird ausgeführt.

## 2. Knobelaufgabe zu Schleifen

a. Schau euch den Code an. Um Schleifen richtig zu schreiben, müsst ihr nachvollziehen können, wie Schleifen funktionieren.

```
1   zaehler = 0
2   ergebnis = 0
3   schleifendurchlaeufe = 0
4
5   while zaehler < 5:
6     ergebnis = ergebnis + 1
7     zaehler = zaehler + 2;
8     schleifendurchlaeufe = schleifendurchlaeufe + 1;
9
10  print("Fertig! Ergebnis = ", ergebnis)
11  print("Schleifendurchläufe: ", schleifendurchlaeufe)
12  print("Zähler: ", zaehler)
```

b. Geht die Schleifendurchläufe Schritt für Schritt durch, um folgende Fragen zu beantworten:

- Welche Ausgabe erhaltet ihr, wenn ihr das Programm startet? Welchen Wert hat die Variable 'ergebnis'?
- Wie oft wird die Schleife durchlaufen? Welchen Wert hat die Variable 'schleifendurchlaeufe'?

c. Nutzt die Tabelle unten als Vorlage, um alle Zwischenergebnisse zu notieren. Schreibt sie ab und füllt sie handschriftlich aus.

| Schleifendurchlauf | Ergebnis | Zaehler |
|---|---|---|
| 0 (zu Beginn) | | |
| 1 | | |
| 2 | | |
| ... | | |

d. Vergleicht eure Tabelle mit der eurer MitschülerInnen. Habt ihr die gleichen Zwischenlösungen und das gleiche Ergebnis? Falls ja - super! Falls nicht - versucht nachzuvollziehen, wo der Fehler liegt.

e. Wenn ihr auf eine Lösung gekommen seid, **überprüft euer Ergebnis im Editor**.

f. Stimmt eure Lösung mit der Ausgabe überein? Falls ja - super! Falls nicht - versucht nachzuvollziehen, wo der Fehler liegt. Tauscht euch noch mit anderen MitschülerInnen aus.

Auf den nächsten Seiten findet ihr weitere Aufgaben zu Schleifen.

## 3. Knobelaufgabe zu Schleifen

a. Schau euch den Code an.

```
1   zaehler = -2
2   ergebnis = 0
3   schleifendurchlaeufe = 0
4
5   while zaehler < 5:
6     ergebnis = ergebnis + (zaehler * 2)
7     zaehler = zaehler + 2;
8     schleifendurchlaeufe +=1;
9
10  print("Fertig! Ergebnis = ", ergebnis)
11  print("Schleifendurchläufe: ", schleifendurchlaeufe)
12  print("Zähler: ", zaehler)
```

b. Geht die Schleifendurchläufe Schritt für Schritt durch, um folgende Fragen zu beantworten:

- Welche Ausgabe erhaltet ihr, wenn ihr das Programm startet? Welchen Wert hat die Variable 'ergebnis'?
- Wie oft wird die Schleife durchlaufen? Welchen Wert hat die Variable 'schleifendurchlaeufe'?

c. Nutzt die Tabelle unten als Vorlage, um alle Zwischenergebnisse zu notieren. Schreibt sie ab und füllt sie handschriftlich aus.

| Schleifendurchlauf | Ergebnis | Zaehler |
|---|---|---|
| 0 (zu Beginn) | | |
| 1 | | |
| 2 | | |
| ... | | |

d. Vergleicht eure Tabelle mit der eurer MitschülerInnen. Habt ihr die gleichen Zwischenlösungen und das gleiche Ergebnis? Falls ja - super! Falls nicht - versucht nachzuvollziehen, wo der Fehler liegt.

e. Wenn ihr auf eine Lösung gekommen seid, **überprüft euer Ergebnis im Editor**.

f. Stimmt eure Lösung mit der Ausgabe überein? Falls ja - super! Falls nicht - versucht nachzuvollziehen, wo der Fehler liegt. Tauscht euch noch mit anderen MitschülerInnen aus.

Auf den nächsten Seiten findet ihr weitere Aufgaben zu Schleifen.

## 1. Extra-Aufgabe zu Schleifen

In Python gibt es verschiedene Möglichkeiten, eine FOR-Schleife zu formulieren.

a. Gebt den Python Code unten in euren Editor ein.
Klickt auf "Run" (Play Symbol) und schaut euch das Ergebnis in der Konsole an.
b. Lest euch den Quellcode und die Kommentare (mit # gekennzeichnet) durch.
c. Passt den Code so an, dass anstelle der 7er Reihe, die **10er Reihe** ausgegeben wird.

Unter appcamps.link/python5 findet ihr eine Erklärung (englisch) zur Range Funktion in Python.
Auf der nächsten Seite findet ihr eine mögliche Lösung.

```
1  #Schleife mit Start-, Endwert und Schritten
2  for i in range(7,71,7):
3      #...und das Ergebnis in der Konsole ausgibt
4      print(i)
```

## 2. Extra-Aufgabe zu Schleifen

a. Gebt den Python Code unten in euren Editor ein.
Klickt auf "Run" (Play Symbol) und schaut euch das Ergebnis in der Konsole an.
b. Lest euch den Quellcode und die Kommentare (mit # gekennzeichnet) durch.
c. Passt den Code so an, dass anstelle 'abc' **dein Name** ausgegeben wird (z.B. Soraya).

Auf der nächsten Seite findet ihr eine mögliche Lösung.

```
1  #Schleife, die einen String iterativ (Schritt für Schritt)
   durchläuft...
2  for i in 'abc':
3      #...und das Ergebnis in der Konsole ausgibt
4      print(i)
```

## 3. Extra-Aufgabe zu Schleifen

a. Ersetzt den String **dein Name** aus Aufgabe 2 durch **dein Name [1:3:1]**.
Auf der nächsten Seite findet ihr eine mögliche Lösung.

## Mögliche Lösung - 1. Extra-Aufgabe

```
1  #Schleife mit Start-, Endwert und Schritten
2  for i in range(10,101,10):
3      #...und das Ergebnis in der Konsole ausgibt
4      print(i)
```

```
10
20
30
40
50
60
70
80
90
100
```

## Mögliche Lösung - 2. Extra-Aufgabe

```
1  #Aufgabe: Ersetze den String 'abc' durch 'dein Name'
2
3  for i in 'Soraya':
4      print(i)
```

```
S
o
r
a
y
a
```

## Mögliche Lösung - 3. Extra-Aufgabe

```
1  #Ausgabe des zweiten und dritten Buchstabens
2  for i in 'Soraya'[1:3:1]:
3      print(i)
4
5  #Über die eckigen Klammern kann ausgewählt werden, welche Zeichen
   aus einem String angezeigt werden sollen.
6  #1. Wert gibt den Startwert an. (Denk dran: In der Programmierung
   fängt man bei 0 an zu zählen.)
7  #2. Wert gibt den Endwert an. Der Wert ist exklusive. Das bedeutet,
    dass das Zeichen nicht mehr mit ausgegeben wird.
8  #3. Wert gibt die Schrittlänge an. Bei 1 wird also jedes Zeichen
   ausgegeben, bei 2 nur jedes zweite. Du kannst übrigens auch mit
   Vorzeichen arbeiten, um zum Beispiel rückwärts zu gehen.
```

```
o
r
```

## 4. Extra-Aufgabe zu Schleifen

a. Gebt den Code aus der **3. Extra-Aufgabe** in euren Editor ein. Klickt auf "run" und schaut euch das Ergebnis in der Konsole an. Hattet ihr dieselbe Ausgabe?

b. Lest euch den Quellcode und die Kommentare (mit # gekennzeichnet) durch. Alles verstanden? Super. Dann geht's jetzt weiter mit der 4. Extra-Aufgabe.

c. Jetzt wird es kniffelig. Versucht 'Soraya' rückwärts auszugeben. In der Konsole soll also 'ayaroS' ausgegeben werden. In den Kommentaren findet ihr einen Hinweis, wie das funktioniert. Probiert einfach mal aus. Wenn ihr nicht weiter kommt, schaut euch den Link appcamps.link/python6 (englisch) an.

Auf der nächsten Seite findet ihr eine mögliche Lösung.

## Mögliche Lösung - 4. Extra-Aufgabe

```
1  #For-Schleife, in der ein String rückwärts ausgegeben wird
2  for i in 'Soraya'[::-1]:
3      print(i)
```

```
a
y
a
r
o
S
```

## Geschafft!

Seid ihr schon fertig? Toll! In der nächsten Sitzung dreht sich alles um das Thema Verzweigungen. Ihr schreibt erste Programme mit Verzweigungen und lernt sie zu verstehen. Viel Spaß dabei!