Hallo Welt!

Einführung: Arbeiten mit dem Editor

Wir arbeiten mit dem repl.it Editor. Für diesen benötigen alle NutzerInnen inzwischen einen Login. Alternativ könnt ihr diesen kosten- und werbefreien Editor benutzen, der ohne Login funktioniert: Web Tiger Jython (hier muss jedoch darauf geachtet werden, dass in den Einstellungen Python 3 aktiviert ist).

1. Aufgabe

a. Schaut euch euren Editor für die Programmierung erst einmal genau an und macht euch mit der Oberfläche vertraut.
b. Gebt den Python Code unten in euren Editor ein.
Klickt auf "Run" (Play Symbol) und schaut euch das Ergebnis in der Konsole an.
c. Lest euch den Quellcode und die Kommentare (mit # gekennzeichnet) durch.
d. Passt den Code so an, dass er **Hallo Vorname!** ausgibt (z.B. Hallo Hannes!).

```
1  print('Hallo Kaya!')
2
3  # Mit dem Befehl 'print' können Buchstaben oder
   Zahlen in der Konsole ausgegeben werden.
4  # Was ausgegeben werden soll, muss in () hinter dem
   Befehl stehen.
5  # Wenn Buchstaben ausgegeben werden sollen bzw. eine
   Zeichenkette, müssen diese in 'einfache' oder
   "doppelte" Anführungszeichen gesetzt werden. Welche
   Schreibweise ihr verwendet, ist euch überlassen.
6  # Wenn Zahlen ausgegeben werden sollen, können diese
   ohne Anführungszeichen hingeschrieben werden.
```

```
Hallo Kaya!
>
```

2. Aufgabe

a. Gebt den Python Code unten in euren Editor ein, klickt auf "Run" und beantwortet die Fragen in der Konsole.
b. Lest euch den Quellcode und die Kommentare (mit # gekennzeichnet) durch.
c. Passt den Code so an, dass er **auch euer Alter abfragt und ausgibt.**
d. Die Lösung und eine Erklärung, was hier passiert, findet ihr auf der nächsten Seite.

```
1  #Hier wird der Variablen 'name' ein Input aus der
   Konsole zugewiesen.
2  name = input('Wie heißt du?')
3  # Hier werden nun der Text 'Hallo' und der
   abgefragte Name ausgegeben.
4  # Über das Komma werden die gewünschten Elemente für
   die Ausgabe aufgelistet.
5  print('Hallo', name, '!')
```

```
Wie heißt du? Esra
Hallo  Esra !
>
```

Mögliche Lösung - Aufgabe 2

```
1  #Hier wird der Variablen 'name' ein Input aus der
   Konsole zugewiesen.
2  name = input('Wie heißt du?')
3  alter = input ('Wie alt bist du?')
4  # Hier werden nun der Text 'Hallo' und der
   abgefragte Name ausgegeben.
5  # Über das Komma werden die gewünschten Elemente für
   die Ausgabe aufgelistet.
6  print('Hallo', name, '!')
7  print ('Du bist', alter, 'Jahre alt')
```

```
Wie heißt du? Anne
Wie alt bist du? 38
Hallo  Anne !
Du bist  38 Jahre alt
>
```

Hinweis: Sieht eure Lösung ähnlich aus? Wenn nicht, ist es nicht schlimm, solange das Ergebnis stimmt. Beim Programmieren gibt es fast immer mehrere Lösungswege.

Variablen

Um den Namen und das Alter zu speichern, habt ihr **Variablen** verwendet:

- name = ...
- alter = ...

Variablen sind Platzhalter, die beliebige Inhalte haben können. Ihr könnt eure Namen in der Variable **name** speichern, den Namen eures Lehrers oder den Namen einer Klassenkameradin. Sobald ihr etwas in der Variablen gespeichert habt, merkt sich der Computer das und ihr könnt später im Programm wieder darauf zugreifen.

So haben wir das auch gemacht:

- In Zeile 2 (siehe Lösung oben) tragen wir etwas in die Variable **name** ein.
- In Zeile 6 greifen wir auf die Variable zu.

Variablen werden im Programm angelegt. Dazu sagt man auch "Man deklariert (implizit) eine Variable". Das macht man meistens so:

```
variablenname =
```

Python Merkzettel

Nehmt euren **Python Merkzettel** und füllt die Box zum Thema **Variable** aus.

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.)

Auf der nächsten Seite findet ihr weitere Aufgaben.

3. Aufgabe

a. **Was passiert, wenn ihr ... über die Konsole ausgeben lasst?**
Schaut euch die Befehle 1 bis 5 an und fügt den Code in euren Editor ein. Beachtet sowohl die Leerzeichen, als auch die Gänsefüßchen bei eurer Eingabe! **Klickt für jede Option auf "run" und schaut euch das Ergebnis an.** Was fällt euch auf?

1. print(3 + 3)
2. print("3 + 3")
3. print("6+ 1/2 = ", 6+ 1/2)
4. print(Aloha)
5. print("Aloh", ' a', "a")

b. Lest euch dann die Quellcode-Snippets unten und die Kommentare (mit # gekennzeichnet) einzeln durch.

c. Schaut euch auch die jeweiligen Ausgaben rechts im Editor an.

Könnt ihr alles nachvollziehen? Super. Dann schaut euch die Erklärung auf der folgenden Seite an.

```
1  print(3 + 3)
2  #Die Ausgabe ist das Ergebnis von 3 + 3, nämlich 6.
3  #Die Leerzeichen werden NICHT beachtet.
```

```
6
>
```

```
1  print("3 + 3")
2  #Durch die Gänsefüßchen wird genau ausgegeben, was
   dort steht, nämlich 3 + 3
3  #Die Leerzeichen werden beachtet
```

```
3 + 3
>
```

```
1  print("6+ 1/2 =", 6+ 1/2)
2  #Durch das Komma werden die Zeichenkette (mit
   Gänsefüßchen) und das Ergebnis der Rechunung
   gleichzeitig ausgegeben
3  #Das Ergebnis ist eine Kommazahl und Python beachtet
   bei der Berechnung sogar Punkt- vor Strichrechnung.
```

```
6+ 1/2 = 6.5
>
```

```
1  print(Aloha)
2  #Hier bekommt ihr eine Fehlermeldung. Python kann
   eine Zeichenkette ohne Gänsefüßchen nicht ausgeben
```

```
Traceback (most recent call last):
  File "main.py", line 1, in <module>
    print(Aloha)
NameError: name 'Aloha' is not defined
>
```

```
1  print("Aloh", ' a', "a")
2  #Hier werden die Buchstaben aneinandergereiht
3  #Bei den Kommata fügt Python automatisch ein
   Leerzeichen ein
4  #Die Leerzeichen in den Gänsefüßchen werden beachtet,
    die anderen nicht
5  #Ihr könnt sowohl deutsche "Gänsefüßchen", als auch
   amerikanische 'Anführungszeichen' verwenden
```

```
Aloh  a a
>
```

Datentypen

Ihr habt gelernt, dass ihr Wörter bzw. Zeichenfolgen in “Gänsefüßchen” setzen müsst, Zahlen jedoch nicht. Wieso ist das so? Das liegt an den unterschiedlichen **Datentypen,** die man in der Programmierung verwenden kann. Ein paar typische findest du in der folgenden Tabelle:

Datentyp	... werden verwendet für...	Beispiel
Integer	Ganzzahlen	6 oder 99
Float	Kommazahlen	6.5 (beim Programmieren immer mit einem Punkt! Also nicht 6,5)
Boolean	Wahrheitswerte	True oder False
String	Zeichenketten	"Aloha" oder "3 + 3"

Sobald ihr um eine Zeichenfolge Gänsefüßchen setzt, interpretiert Python diese als **String**. So zum Beispiel in der vorherigen Aufgabe bei "3 + 3".

```
1   print("3 + 3")
2   #Durch die Gänsefüßchen wird genau ausgegeben, was
    dort steht, nämlich 3 + 3
3   #Die Leerzeichen werden beachtet
```

```
3 + 3
>
```

Bei "3 + 3" wird nicht das Ergebnis der Rechnung ausgegeben, sondern die Rechnung selbst: 3 + 3. Anders sieht es aus ohne Gänsefüßchen:

```
1   print(3 + 3)
2   #Die Ausgabe ist das Ergebnis von 3 + 3, nämlich 6.
3   #Die Leerzeichen werden NICHT beachtet.
```

```
6
>
```

Die Zahlen werden automatisch als **Integer** interpretiert und Python berechnet das Ergebnis. Python erkennt ebenfalls automatisch, ob es sich um einen **Integer** oder einen **Float** handelt.

Ihr müsst in eurem Programm also nicht sagen, von welchem Typ eine Variable ist. Die Datentypen werden automatisch zugewiesen (und angepasst). Das ist in manchen anderen Programmiersprachen anders.

Python Merkzettel

Nehmt euren **Python Merkzettel** und füllt die Box zum Thema **Datentypen** aus.

4. Aufgabe

Ihr habt bereits gelernt, dass Python den jeweiligen Datentyp automatisch zuweist. Wenn ihr in Python wissen wollt, von welchem Typ eure Variable ist, könnt ihr dazu den type() - Befehl nutzen. Unten seht ihr ein Beispiel.

a. Versucht den Code nachzuvollziehen. Gebt ihn dafür in euren Editor ein.
b. Klickt auf 'run' und schaut euch die Ausgabe an.
c. Habt ihr alles verstanden? Super! Dann fügt weitere Variablen mit unterschiedlichen Datentypen **(String, Boolean, Float)** hinzu und gebt mit Hilfe von print aus, von welchem Typ sie sind.

Auf der nächsten Seite findet ihr eine mögliche Lösung. Viel Spaß!

```
1   ichBinEinInt = 3
2   print("Die Variable 'ichBinEinInt' ist vom Typ", type
    (ichBinEinInt))
3
```

```
Die Variable 'ichBinEinInt' ist vom Typ <class 'int'>
>
```

Mögliche Lösung - Aufgabe 4

```
1   ichBinEinInt = 3
2   print("Die Variable 'ichBinEinInt' ist vom Typ", type
    (ichBinEinInt))
3
4   ichBinEinFloat = 3.5
5   print("Die Variable 'ichBinEinFloat' ist vom Typ",
    type(ichBinEinFloat))
6
7   ichBinEinBoolean = True
8   print("Die Variable 'ichBinEinBoolean' ist vom Typ",
    type(ichBinEinBoolean))
9
10  ichBinEinString = 'Hallo'
11  print("Die Variable 'ichBinEinString' ist vom Typ",
    type(ichBinEinString))
12
```

```
Die Variable 'ichBinEinInt' ist vom Typ <class 'int'>
Die Variable 'ichBinEinFloat' ist vom Typ <class 'float'>
Die Variable 'ichBinEinBoolean' ist vom Typ <class 'bool'>
Die Variable 'ichBinEinString' ist vom Typ <class 'str'>
>
```

Hinweis: Ergänzt ggf. euren Merkzettel. Ihr habt etwas über den type()-Befehl in Python gelernt. Dieser ist eine große Hilfe bei der Zuweisung von Datentypen in Python.

5. Aufgabe

Freestyle: Lernt die Ausgabe (das, was unten im dunkelblauen Feld angezeigt wird, nachdem ihr auf "run" geklickt habt) besser kennen.

Was passiert zum Beispiel, wenn ihr 'Hallo', name und '!' in drei verschiedenen print Anweisungen ausgeben lasst, statt in einer?

```
1   #Hier wird der Variablen 'name' ein Input aus der
    Konsole zugewiesen.
2   name = input('Wie heißt du?')
3   # Hier werden nun der Text 'Hallo' und der
    abgefragte Name ausgegeben.
4   # Über das Komma werden die gewünschten Elemente für
    die Ausgabe aufgelistet.
5   print('Hallo', name, '!')
6
```

```
Wie heißt du? Hannes
Hallo  Hannes !
>
```

6. Aufgabe

Freestyle: Lernt die Ausgabe besser kennen.

Was passiert, wenn ihr die Zahlen über die Konsole einlest, statt einen festen Wert zu vergeben? (*Tipp:* Werte könnt ihr mit input("...") einlesen. Bei der Aufgabe mit Name und Alter hatten wir das schon.)

Auf den nächsten Seiten findet ihr weitere Anregungen, was ihr noch ausprobieren könnt. Viel Spaß!

```
1   zahl1 = 3
2   zahl2 = 3
3
4   print(zahl1 + zahl2)
5
```

```
6
>
```

Mögliche Lösung - Aufgabe 6

```
1   zahl1 = input('Gib eine Zahl ein:')
2   zahl2 = input('Gib eine zweite Zahl ein:')
3
4   print(zahl1 + zahl2)
5   #Über Input wird offenbar automatisch ein String
    eingelesen und kein Integer, selbst wenn eine Zahl
    eingegeben wird
6   #Durch das Plus werden die Zeichenketten direkt
    aneinandergehängt, ohne Leerzeichen wie bei dem Komma
7
```

```
Gib eine Zahl ein:8
Gib eine zweite Zahl ein:23
823
>
```

7. Aufgabe

Freestyle: Lernt die Ausgabe besser kennen. Achtung! Jetzt wird es tricky.

Gibt es vielleicht eine Möglichkeit, dass die Eingabe über Input doch als Zahl, statt String, eingelesen wird? Unter appcamps.link/python1 findet ihr ein Beispiel.

Auf den nächsten Seiten findet ihr eine Lösung.

Hinweis: Beim Programmieren sucht man häufig im Internet nach passenden Beispielen, um eine Lösung zu finden. Solltet ihr in Zukunft mal nicht wissen, wie ihr etwas programmieren könnt: Sucht einfach im Internet danach. Ihr werdet sicher fündig. :)

8. Aufgabe

Habt ihr herausgefunden, wie es funktioniert? Super! Wenn nicht ist auch nicht schlimm. Auch wenn nur drei kleinen Buchstaben ("int") benötigt werden, ist es gar nicht so leicht darauf zu kommen.

Was passiert, wenn ihr nun statt einer Zahl einen **Buchstaben** oder eine **Kommazahl** eingebt?
Probiert es aus. Auf der nächsten Seite ist die Lösung und ein Ausblick.

Mögliche Lösung - Aufgabe 7

```
1   #Durch int wird die Eingabe in einen Integer
    umgeformt. Das nennt man auch "Parsen"
2   zahl1 = int(input("Welche ist die erste Zahl"))
3   zahl2 = int(input("Welche ist die zweite Zahl"))
4
5   print(zahl1 + zahl2)
6
```

```
Welche ist die erste Zahl 6
Welche ist die zweite Zahl12
18
>
```

Hinweis: Ergänzt ggf. euren Merkzettel. Ihr habt etwas über Datentypen in Python und das Umformen (“parsen”) von Datentypen gelernt.

Eingabe Buchstaben

```
1   #Durch int wird die Eingabe in einen Integer
    umgeformt. Das nennt man auch "Parsen"
2   zahl1 = int(input("Welche ist die erste Zahl"))
3   zahl2 = int(input("Welche ist die zweite Zahl"))
4
5   print(zahl1 + zahl2)
6
```

```
Welche ist die erste Zahl Hallo
Traceback (most recent call last):
  File "main.py", line 2, in <module>
    zahl1 = int(input("Welche ist die erste Zahl"))
ValueError: invalid literal for int() with base 10: ' Hallo'
>
```

Eingabe Kommazahl

```
1   #Durch int wird die Eingabe in einen Integer
    umgeformt. Das nennt man auch "Parsen"
2   zahl1 = int(input("Welche ist die erste Zahl"))
3   zahl2 = int(input("Welche ist die zweite Zahl"))
4
5   print(zahl1 + zahl2)
6
```

```
Welche ist die erste Zahl 3.5
Traceback (most recent call last):
  File "main.py", line 2, in <module>
    zahl1 = int(input("Welche ist die erste Zahl"))
ValueError: invalid literal for int() with base 10: ' 3.5'
>
```

Hinweis: Ihr erhaltet eine Fehlermeldung! Um diese zu umgehen, nutzt man sogenannte Kontrollstrukturen. Was das ist und wofür man diese sonst noch verwenden kann, lernt ihr in den folgenden Sitzungen.

Mögliche Lösung - Aufgabe 8

```
1   zahl1 = "a"
2   zahl2 = "b"
3
4   try:
5       zahl1 = float(input("Welche ist die erste
        natürliche Zahl"))
6   except ValueError:
7     print("Du hast keine natürlich Zahl eingegeben.
      Bitte starte das Programm erneut.")
8     quit()
9
10  try:
11    zahl2 = float(input("Welche ist die zweite
      natürlich Zahl"))
12  except ValueError:
13    print("Du hast keine natürliche Zahl eingegeben.
      Bitte starte das Programm erneut.")
14    quit()
15
16  print(zahl1 + zahl2)
17
```

```
Welche ist die erste natürliche Zahl Hallo
Du hast keine natürlich Zahl eingegeben. Bitte starte das Progra
mm erneut.
repl process died unexpectedly:
>
```

Extra-Challenge

Programmiert einen einfachen **Taschenrechner**, der zwei Integer Werte abfragt und dann jeweils das Ergebnis der Addition, der Subtraktion, der Multiplikation sowie der Division berechnet und anzeigt. Die Grundlagen dazu habt ihr bereits gelernt!

Befolgt nun folgende Schritte und testet nach jedem Schritt, ob euer Programm wie gedacht funktioniert. (Testen funktioniert häufig sehr gut über die Ausgabe, also mit dem 'print'-Befehl.)

Schritte 1. - 4.

1. Schritt: Lest zwei Werte über die Konsole ein. Speichert diese als Integer in je einer Variablen (wert1 und wert2). Beachtet: Ihr dürft im Moment davon ausgehen, dass der Benutzer keine falschen Eingaben macht! (Teste mit print(wert1, wert2).)

2. Schritt: Addiert die beiden Werte und speichert das Ergebnis in einer weiteren Variablen (ergebnis_add). Erweitert die 'print'-Zeile folgendermaßen:

print('Ergebnis Addition von %d + %d = %d' % (wert1, wert2, ergebnis_add))

3. Schritt: Fügt drei weitere Variablen (ergebnis_sub, ergebnis_mul und ergebnis_div) hinzu. Speichert das Ergebnis der Subtraktion (-), das Ergebnis der Multiplikation (*) und das Ergebnis der Division (/) in der jeweiligen Variablen. Fügt nun auch die passenden print-Befehle in das Programm ein.

4. Schritt: Extra-Challenge: Das Ergebnis der Division stimmt leider nicht, wenn keine natürliche Zahl rauskommt. Welchen Datentyp braucht ihr für Kommazahlen? Beachtet: Auch beim print-Befehl müsst ihr Anpassungen vornehmen.

Hinweis: Entweder versucht ihr die Aufgabe selbstständig zu lösen. Alternativ findet ihr auf den folgenden Seiten die Schritt für Schritt Lösungen. Wenn ihr alle Schritte korrekt befolgt habt, sollte die Ausgabe so aussehen:

```
Gib den ersten Wert ein: 2
Gib den zweiten Wert ein: 5
Ergebnis Addition von 2 + 5 = 7
Ergebnis Subtraktion von 2 - 5 = -3
Ergebnis Multiplikation von 2 * 5 = 10
Ergebnis Division von 2 / 5 = 0
```

Extra-Challenge - Musterlösung

1. Schritt: Lest zwei Werte über die Konsole ein. Speichert diese als Integer in je einer Variablen (wert1 und wert2). Beachtet: Ihr dürft im Moment davon ausgehen, dass der Benutzer keine falschen Eingaben macht! (Teste mit print(wert1, wert2).)

```
#Zwei Werte abfragen, mit denen gerechnet wird
wert1 = int(input('Gib den ersten Wert ein:'))
wert2 = int(input('Gib den zweiten Wert ein:'))

#Testen:
print(wert1, wert2)
```

```
Gib den ersten Wert ein: 1
Gib den zweiten Wert ein: 4
1 4
>
```

2. Schritt: Addiert die beiden Werte und speichert das Ergebnis in einer weiteren Variablen (ergebnis_add). Erweitert die 'print'-Zeile folgendermaßen: print('Ergebnis Addition von %d + %d = %d' % (wert1, wert2, ergebnis_add))

```
#Zwei Werte abfragen, mit denen gerechnet wird
wert1 = int(input('Gib den ersten Wert ein:'))
wert2 = int(input('Gib den zweiten Wert ein:'))

#Berechnungen durchführen
ergebnis_add = wert1+wert2

#Ergebnisse ausgeben
print('Ergebnis Addition von %d + %d = %d' % (wert1,
wert2, ergebnis_add))
```

```
Gib den ersten Wert ein: 4
Gib den zweiten Wert ein: 6
Ergebnis Addition von 4 + 6 = 10
>
```

3. Schritt: Fügt drei weitere Variablen (ergebnis_sub, ergebnis_mul und ergebnis_div) hinzu. Speichert das Ergebnis der Subtraktion (-), das Ergebnis der Multiplikation (*) und das Ergebnis der Division (/) in der jeweiligen Variablen. Fügt nun auch die passenden print-Befehle in das Programm ein.

```
#Zwei Werte abfragen, mit denen gerechnet wird
wert1 = int(input('Gib den ersten Wert ein:'))
wert2 = int(input('Gib den zweiten Wert ein:'))

#Berechnungen durchführen
ergebnis_add = wert1+wert2
ergebnis_sub = wert1-wert2
ergebnis_mul = wert1*wert2
ergebnis_div = wert1/wert2

#Ergebnisse ausgeben
print('Ergebnis Addition von %d + %d = %d' % (wert1,
wert2, ergebnis_add))
print('Ergebnis Subtraktion von %d - %d = %d' % (wert1,
wert2, ergebnis_sub))
print('Ergebnis Multiplikation von %d * %d = %d' %
(wert1, wert2, ergebnis_mul))
print('Ergebnis Division von %d / %d = %d' % (wert1,
wert2, ergebnis_div))
# %d kannst du verwenden, um Dezimalzahlen direkt im
String zu erwähnen und später zuzuweisen. Dabei ist
natürlich wichtig, dass du die Reihenfolge beachtest.
# Die alternative, aber etwas komplizierte Schreibweise
ist:
#print("Ergebnis Addition von", wert1, "+", wert2, "=",
ergebnis_add)
# Wenn du zu dieser Lösung gekommen bist, kanst du
übrigens den vierten Schritt überspringen. Python weist
nämlich den Datentyp automatisch zu. Dementsprechend
wird die Variable "ergebnis_div" bei Bedarf automatisch
der passende Datentypen für Kommazahlen zugewiesen.
# -> Der Datentyp einer Variable kann sich bei Python
ändern.
```

```
Gib den ersten Wert ein: 8
Gib den zweiten Wert ein: 32
Ergebnis Addition von 8 + 32 = 40
Ergebnis Subtraktion von 8 - 32 = -24
Ergebnis Multiplikation von 8 * 32 = 256
Ergebnis Division von 8 / 32 = 0
>
```

Extra-Challenge - Musterlösung

4. Schritt: Extra-Challenge: Das Ergebnis der Division stimmt leider nicht, wenn keine natürliche Zahl rauskommt. Welchen Datentyp braucht ihr für Kommazahlen? Beachtet: Auch beim print-Befehl müsst ihr Anpassungen vornehmen.

Geht das Programm Schritt für Schritt durch. Lest euch insbesondere auch die Kommentare aufmerksam durch.

Sieht eure Lösung ähnlich aus? Vielleicht habt ihr eine andere Lösung gefunden, die auch korrekt ist.

```
1  #Zwei Werte abfragen, mit denen gerechnet wird
2  wert1 = float(input('Gib den ersten Wert ein:'))
3  wert2 = float(input('Gib den zweiten Wert ein:'))
4
5  #Berechnungen durchführen
6  ergebnis_add = wert1+wert2
7  ergebnis_sub = wert1-wert2
8  ergebnis_div = wert1/wert2
9  ergebnis_mul = wert1*wert2
10
11 #Ergebnisse ausgeben mit 2 Nachkommastellen.
12 print('Ergebnis Addition von %.2f + %.2f = %.2f' %
   (wert1, wert2, ergebnis_add))
13 print('Ergebnis Subtraktion von %.2f - %.2f = %.2f' %
   (wert1, wert2, ergebnis_sub))
14 print('Ergebnis Multiplikation von %.2f * %.2f = %.2f'
   % (wert1, wert2, ergebnis_mul))
15 print('Ergebnis Division von %.2f / %.2f = %.2f' %
   (wert1, wert2, ergebnis_div))
16 #Beachte:
17 # %f kannst du verwenden, um Kommazahlen direkt im
   String zu erwähnen und später zuzuweisen.
18 # Mit &.2f legst du fest, dass 2 Nachkommastellen
   angezeigt werden. Diesen Wert kannst du natürlich
   ändern wie du möchtest. Also zum Beispiel .4f
19 #Die Zahlen werden abgeschnitten und nicht gerundet.
   Aus 1.239 wird also 1.23 und nicht 1.24!
```

```
Gib den ersten Wert ein: 82
Gib den zweiten Wert ein: 34
Ergebnis Addition von 82.00 + 34.00 = 116.00
Ergebnis Subtraktion von 82.00 - 34.00 = 48.00
Ergebnis Multiplikation von 82.00 * 34.00 = 2788.00
Ergebnis Division von 82.00 / 34.00 = 2.41
>
```

Super Extra-Challenge

Seid ihr schon fertig? Toll! Dann überlegt euch nun, wie ihr den Taschenrechner noch erweitern könnt. Unter appcamps.link/python2 (englisch) findet ihr Anregungen. Ihr könnt auch euren Schultaschenrechner als Vorlage nehmen. Welche Funktionen verwendet ihr häufig?