

# micro:bit Programmierung in Python – Lösungen zu Kapitel IV



## Variablen (Schwierigkeit: 😊😊😊)

Das neue Konzept der Variablen eröffnet viele neue Möglichkeiten! Probiere zunächst die erste Aufgabe zu den unterschiedlichen Blockarten aus und versuche dich dann an deinen eigenen Ideen – oder den Aufgabenideen unten.

Arbeitsauftrag	Lösung
<p>Überlege dir, was passieren würde, wenn du im Beispielcode statt dem <b>meine_zahl+=1</b>, <b>meine_zahl=1</b> genommen hättest. Was passiert, wenn man <b>global meine_zahl</b> nicht geschrieben hätte? Probiere es anschließend aus!</p>	<pre>meine_zahl=0 def on_forever():     basic.show_number(meine_zahl) basic.forever(on_forever)  def on_button_pressed_b():     global meine_zahl     meine_zahl = 1 input.on_button_pressed(Button.B, on_button_pressed_b)  def on_button_pressed_a():     global meine_zahl     meine_zahl += -2 input.on_button_pressed(Button.A, on_button_pressed_a)</pre> <p>Wenn man global meine_zahl nicht schreibt, kommt eine Fehlermeldung – der Code funktioniert nicht mehr. Warum – das steht im passenden Informationsblatt.</p>
<p>Was passiert, wenn man einen beiden Befehle (<b>meine_zahl+=1</b> oder <b>meine_zahl =1</b>) beim Beispiel oben in die Methode <b>on_forever()</b> verschiebt? Überlege zuerst, tausch dich mit deinem Sitznachbarn aus und probiert es dann gemeinsam aus. Was passiert? Wer hatte</p>	<pre>meine_zahl=0  def on_forever():     global meine_zahl     basic.show_number(meine_zahl)     meine_zahl = 1 basic.forever(on_forever)</pre>

<p>Recht und warum ist das so?</p>	<pre>def on_button_pressed_b():     global meine_zahl     meine_zahl += 1 input.on_button_pressed(Button.B, on_button_pressed_b)  def on_button_pressed_a():     global meine_zahl     meine_zahl += -2 input.on_button_pressed(Button.A, on_button_pressed_a)</pre> <p>Der Wert der Zahl würde sich nicht mehr ändern – oder ständig höher werden, denn die Zahl wird dauerhaft geändert. Bei <code>meine_zahl += 1</code> kann man dies noch durch die Knopfdrücke ändern und zusätzlich erhöhen oder erniedrigen. Bei <code>meine_zahl = 1</code> bleibt der Wert bei 1 „stecken“</p>
<p>Wie muss man den Code im Beispiel oberhalb ändern, dass ein ‚normaler‘ Zähler entsteht, der bei A-Tastendruck eins nach unten zählt und bei B-Tastendruck um eins nach oben? Versuche damit die Anzahl der Menschen in deinem Raum zu zählen oder die Anzahl der braunhaarigen/blonden Menschen. Was könntest du noch damit zählen?</p>	<pre>meine_zahl=0  def on_forever():     basic.show_number(meine_zahl) basic.forever(on_forever)  def on_button_pressed_b():     global meine_zahl     meine_zahl += 1 input.on_button_pressed(Button.B, on_button_pressed_b)  def on_button_pressed_a():     global meine_zahl     meine_zahl += -1 input.on_button_pressed(Button.A, on_button_pressed_a)</pre>
<p>Kann man das so erweitern, dass bei Schütteln der Zähler auf die Zahl 5 gesetzt wird – egal, welcher Wert gerade gespeichert ist?</p>	<pre>meine_zahl=0  def on_forever():     basic.show_number(meine_zahl) basic.forever(on_forever)  def on_button_pressed_b():     global meine_zahl     meine_zahl += 1</pre>

	<pre>input.on_button_pressed(Button.B, on_button_pressed_b)  def on_button_pressed_a():     global meine_zahl     meine_zahl += -1 input.on_button_pressed(Button.A, on_button_pressed_a)  def on_gesture_shake():     global meine_zahl     meine_zahl = 5 input.on_gesture(Gesture.Shake, on_gesture_shake)</pre>
<p>Füge dem vorherigen Programm noch einen Startwert hinzu, es soll immer von -5 weggezählt werden.</p>	<pre>meine_zahl=-5  def on_forever():     basic.show_number(meine_zahl) basic.forever(on_forever)  def on_button_pressed_b():     global meine_zahl     meine_zahl += 1 input.on_button_pressed(Button.B, on_button_pressed_b)  def on_button_pressed_a():     global meine_zahl     meine_zahl += -1 input.on_button_pressed(Button.A, on_button_pressed_a)  def on_gesture_shake():     global meine_zahl     meine_zahl = 5 input.on_gesture(Gesture.Shake, on_gesture_shake)</pre>
<p>Dreierzähler:          Programmiere einen Zähler, der bei Drücken der Taste A immer um 3 erhöht (0 -&gt; 3 -&gt; 6 -&gt; 9 -&gt; ...).</p>	<pre>meine_zahl=0  def on_forever():     basic.show_number(meine_zahl) basic.forever(on_forever)</pre>

	<pre>def on_button_pressed_a():     global meine_zahl     meine_zahl +=3 input.on_button_pressed(Button.A, on_button_pressed_a)</pre>
<p>Mix-Zähler:          Programmiere einen Zähler, der nur in Zweierschritten nach oben zählt (0, 2, 4, 6, 8, ...) und in Dreierschritten nach unten (z.B. von 4 auf 1). Beim Drücken von A+B soll der Zähler auf 0 zurückgesetzt werden.</p>	<pre>meine_zahl=0  def on_forever():     basic.show_number(meine_zahl) basic.forever(on_forever)  def on_button_pressed_b():     global meine_zahl     meine_zahl += 2 input.on_button_pressed(Button.B, on_button_pressed_b)  def on_button_pressed_a():     global meine_zahl     meine_zahl += -3 input.on_button_pressed(Button.A, on_button_pressed_a)  def on_button_pressed_ab():     global meine_zahl     meine_zahl = 0 input.on_button_pressed(Button.AB, on_button_pressed_ab)</pre>
<p>Zufallsschütteln: Ändere das Programm (mit den Zweier-/Dreierschritten) so, dass beim Schütteln die Variable auf einen zufälligen Wert gesetzt wird.</p>	<pre>meine_zahl=0  def on_forever():     basic.show_number(meine_zahl) basic.forever(on_forever)  def on_button_pressed_b():     global meine_zahl     meine_zahl += 2 input.on_button_pressed(Button.B, on_button_pressed_b)</pre>

```
def on_button_pressed_a():  
    global meine_zahl  
    meine_zahl += -3  
input.on_button_pressed(Button.A, on_button_pressed_a)  
  
def on_button_pressed_ab():  
    global meine_zahl  
    meine_zahl = 0  
input.on_button_pressed(Button.AB, on_button_pressed_ab)  
  
def on_gesture_shake():  
    global meine_zahl  
    meine_zahl = randint(0, 10)  
input.on_gesture(Gesture.Shake, on_gesture_shake)
```

<p>Zielwert-Blinken: Füge dem Programm noch einen Zielwert hinzu! Wird die Zahl 5 erreicht (Variablenwert = 5), so soll der micro:bit lustig blinken. Wie muss man die Tasten A und B drücken, dass man die Zahl 5 erreicht? Schaffst du es?</p>	<pre> meine_zahl=0  def on_forever():     if meine_zahl==5:         basic.show_icon(IconNames.SQUARE)         basic.show_icon(IconNames.SMALL_SQUARE)     else:         basic.show_number(meine_zahl) basic.forever(on_forever)  def on_button_pressed_b():     global meine_zahl     meine_zahl += 2 input.on_button_pressed(Button.B, on_button_pressed_b)  def on_button_pressed_a():     global meine_zahl     meine_zahl += -3 input.on_button_pressed(Button.A, on_button_pressed_a)  def on_button_pressed_ab():     global meine_zahl     meine_zahl = 0 input.on_button_pressed(Button.AB, on_button_pressed_ab)  def on_gesture_shake():     global meine_zahl     meine_zahl = randint(0, 10) input.on_gesture(Gesture.Shake, on_gesture_shake) </pre>
<p>Kopfstandszähler: Schreibe ein Programm, das zählt, wie oft der micro:bit auf den Kopf gestellt wurde und zeige diese Zahl erst an, wenn beide Tasten A und B zugleich gedrückt werden. Ansonsten soll am micro:bit ein Smiley</p>	<pre> meine_zahl=0  def on_button_pressed_ab():     basic.show_number(meine_zahl) input.on_button_pressed(Button.AB, on_button_pressed_ab)  def on_gesture_up(): </pre>

<p>angezeigt werden und wenn er am Kopf steht, ein erschrockener Smiley.</p>	<pre>basic.show_icon(IconNames.HAPPY) input.on_gesture(Gesture.LOGO_UP, on_gesture_up)  def on_gesture_down():     global meine_zahl     basic.show_icon(IconNames.SURPRISED)     meine_zahl += 1 input.on_gesture(Gesture.LOGO_DOWN, on_gesture_down)</pre>
<p>Ändere das Programm so, dass bei -10 zu Zählen begonnen wird. (Bei einmaligem Kopfüberdrehen soll also bei A+B-Druck dann -9 angezeigt werden.)</p>	<pre>meine_zahl=-10  def on_button_pressed_ab():     basic.show_number(meine_zahl) input.on_button_pressed(Button.AB, on_button_pressed_ab)  def on_gesture_up():     basic.show_icon(IconNames.HAPPY) input.on_gesture(Gesture.LOGO_UP, on_gesture_up)  def on_gesture_down():     global meine_zahl     basic.show_icon(IconNames.SURPRISED)     meine_zahl += 1 input.on_gesture(Gesture.LOGO_DOWN, on_gesture_down)</pre>
<p>Zielwert-Blinken: Erweitere das Programm mit einem Zielwert, z.B. 7. Wurde der micro:bit so oft auf den Kopf gestellt, dass er bei A+B-Druck 7 anzeigen würde, so soll er mit den LEDs blinken (z.B. kleines und großes Rechteck abwechselnd anzeigen).</p>	<pre>meine_zahl=-10  def on_button_pressed_ab():     basic.show_number(meine_zahl) input.on_button_pressed(Button.AB, on_button_pressed_ab)  def on_gesture_up():     basic.show_icon(IconNames.HAPPY) input.on_gesture(Gesture.SCREEN_UP, on_gesture_up)  def on_gesture_down():     global meine_zahl     basic.show_icon(IconNames.SURPRISED)</pre>

	<pre> meine_zahl += 1 input.on_gesture(Gesture.SCREEN_DOWN, on_gesture_down)  def on_forever():     if meine_zahl == 7:         basic.show_icon(IconNames.SQUARE)         basic.show_icon(IconNames.SMALL_SQUARE) basic.forever(on_forever) </pre>
<p>Für Experten:          Erstelle ein neues Programm mit dem Namen „Schüttel-mich“!          Schaffst du es, den micro:bit so zu programmieren, dass er am Start eine zufällige Zahl als Zielwert auswählt (Tipp: diese in eine Variable speichern) und kurz anzeigt?          Danach soll man den micro:bit so oft schütteln bis er diesen Wert erreicht hat. Gelingt dies (Schüttelwert = Zielwert), so sollen Sterne am micro:bit blinken – ansonsten (nicht genug geschüttelt) wird nur ein beliebiges Symbol - such dir selbst eines aus 😊 - angezeigt.          Man soll zudem die aktuelle Schüttel-Anzahl jederzeit mit A+B-Druck anzeigen lassen können.</p>	<pre> meine_zahl=0 meine_Zufallszahl= randint(3,17) basic.show_number(meine_Zufallszahl)  def on_forever():     if meine_zahl == meine_Zufallszahl:         basic.show_icon(IconNames.TARGET)         basic.show_icon(IconNames.DIAMOND)         basic.show_icon(IconNames.SMALL_DIAMOND)     else:         basic.show_icon(IconNames.ASLEEP) basic.forever(on_forever)  def on_gesture_shake():     global meine_zahl     meine_zahl += 1 input.on_gesture(Gesture.Shake, on_gesture_shake)  def on_button_pressed_ab():     basic.show_number(meine_zahl) input.on_button_pressed(Button.AB, on_button_pressed_ab) </pre>