

# micro:bit Programmierung in Python - Lösungen



## Schleifen (Schwierigkeit: 😊😊😊)

Wir haben nun Wiederholungen kennen gelernt. Mit diesen ist es möglich, Blöcke mehrmals zu wiederholen – entweder bis ein bestimmtes Ereignis eintritt (While) oder man gibt eine Zahl ein, die angibt, wie oft der Teil wiederholt werden soll (Zählschleife bzw. For).

Arbeitsauftrag	Lösung
Schreibe ein Programm, das ein Quadrat genau 7-mal blinken lässt.	<pre>for i in range(7):     basic.show_icon(IconNames.SMALL_SQUARE)     basic.show_icon(IconNames.SQUARE)</pre>
Erweitere das Programm so, dass der micro:bit danach ein Herz dauerhaft blinken lässt.	<pre>for i in range(7):     basic.show_icon(IconNames.SMALL_SQUARE)     basic.show_icon(IconNames.SQUARE)  def on_forever():     basic.show_icon(IconNames.HEART)     basic.show_icon(IconNames.SMALL_HEART)  basic.forever(on_forever)</pre>
Schaffst du es, mit einer Schleife von 0 bis 12 zu zählen?	<pre>for i in range(13):     basic.show_number(i)</pre>
Schreibe ein Programm, das zuerst achtmal ein Quadrat blinken lässt und danach von 0 bis 8 zählt.	<pre>for i in range(8):     basic.show_icon(IconNames.SMALL_SQUARE)     basic.show_icon(IconNames.SQUARE)  for i in range(8):     basic.show_number(i)</pre>
Erweitere das Orakel so, dass du vor dem Start des Programms, das Fragezeichen dreimal blinken lässt bevor es durchgehend leuchtet.	<pre>for i in range(3):     basic.show_leds("""         .###.         ...#.         ..#..         .....         ..#..         """)</pre>

	<pre> basic.show_leds("""     .....     .....     .....     .....     .....     """)  def on_forever():     basic.show_leds("""         .###.         ...#.         ..#..         .....         ..#..         """)     basic.forever(on_forever)  def on_button_pressed_ab():     if randint(0, 1) == 0:         basic.show_icon(IconNames.YES)     else:         basic.show_icon(IconNames.NO) input.on_button_pressed(Button.AB, on_button_pressed_ab) </pre>
<p>Schreibe ein Programm, das herunterzählt – z.B. von 3 bis 0</p>	<pre> for i in range(3):     basic.show_number(i) </pre>
<p>Setze obiges Programm mit einer While-Schleife um. Was muss man ändern, welche Bedingung brauchst du (z.B. wiederhole solange größer als ...)</p>	<pre> meine_zahl = 3  while meine_zahl &gt; 0:     basic.show_number(meine_zahl)     meine_zahl += -1 </pre>
<p>Schaffst du es ein Programm zu schreiben, das eine bestimmte Zahl 2-mal blinken lässt und anschließend von dieser</p>	<pre> meine_zahl = 5  for i in range(2):     basic.show_number(meine_zahl)     basic.show_leds(""" </pre>

<p>Zahl herunterzählt bis zur 0?</p>	<pre> ..... ..... ..... ..... ..... ..... """  while meine_zahl &gt; 0:     basic.show_number(meine_zahl)     meine_zahl += -1 </pre>
<p>Nimm ein Programm deiner Wahl her und erweitere es mit einem Startbildschirm! Es soll also vor Ausführung des eigentlichen Programms etwas passieren, z.B. ein Symbol deiner Wahl solange blinken, - bis die Tasten A und B gleichzeitig gedrückt werden (wird A+B schon verwendet, nimm das Schütteln). Danach schreibt der micro:bit „Los!“ und das eigentliche Programm startet.</p>	<p><b>Hier passiert eine Ausnahme – Die Befehle, die sonst ganz oben stehen, dürfen erst nach der Methodendeklaration codiert werden.</b></p> <pre> def on_gesture_shake():     if 0 == randint(0, 1):         basic.show_icon(IconNames.YES)     else:         basic.show_icon(IconNames.NO) input.on_gesture(Gesture.SHAKE, on_gesture_shake)  def on_button_pressed_ab():     global start     start += 1 input.on_button_pressed(Button.AB, on_button_pressed_ab)  start = 0 while start == 0:     basic.show_icon(IconNames.SMALL_SQUARE)     basic.show_icon(IconNames.SQUARE) basic.show_string("LOS!") </pre>
<p>Schreibe ein Programm, in welchem du einstellen kannst, wie oft ein bestimmtes Symbol, z.B. die Ente, blinken soll:</p> <p>Schreibe dazu zuerst ein Zähler-Programm, bei dem mit A-Druck eins nach unten und bei B-</p>	<pre> meine_zahl = 0  def on_button_pressed_ab():     for i in range(3):         basic.show_icon(IconNames.DUCK)         basic.show_leds(""" ..... </pre>

<p>Druck eins nach oben gezählt wird. Der Wert des Zählers soll nur bei Änderung (also bei Druck auf einen der beiden Knöpfe) angezeigt werden (das „on_forever()“ wird nicht benötigt). Bei A+B-Druck soll die Ente zunächst einfach 3-mal blinken.</p>	<pre> ..... ..... ..... ..... .....) input.on_button_pressed(Button.AB, on_button_pressed_ab)  def on_button_pressed_a():     global meine_zahl     meine_zahl += -1     basic.show_number(meine_zahl) input.on_button_pressed(Button.A, on_button_pressed_a)  def on_button_pressed_b():     global meine_zahl     meine_zahl += 1     basic.show_number(meine_zahl) input.on_button_pressed(Button.B, on_button_pressed_b) </pre>
<p>Schreibe ein Programm, in welchem du einstellen kannst, wie oft ein bestimmtes Symbol, z.B. die Ente, blinken soll:</p> <p>Ändere nun das Programm so, dass wenn du A+B drückst, die Ente genauso oft wie eingestellt, blinkt.</p>	<pre> meine_zahl = 0  def on_button_pressed_ab():     for i in range(meine_zahl):         basic.show_icon(IconNames.DUCK)         basic.show_leds(""" ..... ..... ..... ..... ..... .....) input.on_button_pressed(Button.AB, on_button_pressed_ab)  def on_button_pressed_a():     global meine_zahl     meine_zahl += -1     basic.show_number(meine_zahl) </pre>

	<pre>input.on_button_pressed(Button.A, on_button_pressed_a)  def on_button_pressed_b():     global meine_zahl     meine_zahl += 1     basic.show_number(meine_zahl)  input.on_button_pressed(Button.B, on_button_pressed_b)</pre>
<p>Schaffst du es, einen einstellbaren Countdown zu schreiben?</p> <p>Nimm dazu das Zählprogramm aus der vorherigen Aufgabe. Stelle dein Programm nun zum einstellbaren Countdown fertig, sodass bei Druck von A+B die Zahl blinkt.</p> <p>Erweitere das Programm dann so, dass nach dem Blinken des Symbols von der Zahl (dem Zählerwert) bis 0 gezählt wird.</p>	<pre>meine_zahl = 0 basic.show_number(meine_zahl)  def on_button_pressed_ab():     global meine_zahl     for i in range(3):         basic.show_number(meine_zahl)         basic.show_string(" ")     while meine_zahl &gt; 0:         basic.show_number(meine_zahl)         meine_zahl += -1         basic.show_number(meine_zahl) input.on_button_pressed(Button.AB, on_button_pressed_ab) def on_button_pressed_a():     global meine_zahl     meine_zahl += -1     basic.show_number(meine_zahl) input.on_button_pressed(Button.A, on_button_pressed_a)  def on_button_pressed_b():     global meine_zahl     meine_zahl += 1     basic.show_number(meine_zahl) input.on_button_pressed(Button.B, on_button_pressed_b)</pre>
<p><b>Expertenaufgabe:</b> Erweitere den Countdown so, dass er auch für Minuszahlen geht.</p>	<pre>meine_zahl = 0 basic.show_number(meine_zahl)  def on_button_pressed_ab():     global meine_zahl</pre>

```
for i in range(3):
    basic.show_number(meine_zahl)
    basic.show_string(" ")
while meine_zahl != 0:
    if meine_zahl > 0:
        basic.show_number(meine_zahl)
        meine_zahl += -1
    else:
        basic.show_number(meine_zahl)
        meine_zahl += 1
basic.show_number(meine_zahl)
input.on_button_pressed(Button.AB, on_button_pressed_ab)

def on_button_pressed_a():
    global meine_zahl
    meine_zahl += -1
    basic.show_number(meine_zahl)
input.on_button_pressed(Button.A, on_button_pressed_a)

def on_button_pressed_b():
    global meine_zahl
    meine_zahl += 1
    basic.show_number(meine_zahl)
input.on_button_pressed(Button.B, on_button_pressed_b)
```