

# micro:bit Programmierung in Python – Aufgaben III



## Verzweigung (Schwierigkeit: 😊😊)

Denke dir unbedingt das Beispiel mit dem **Mini-Orakel** gut durch und versuche zu verstehen, was dort passiert. Lies dir auch die Beschreibung (unten) nochmals gut durch. Danach kannst du dich gerne an eigenen Ideen probieren – oder du versuchst dich an untenstehenden Aufgaben!

```
def on_forever():
```

```
    basic.show_leds("""
        .###.
        ...#.
        ..#..
        .....
        ..#..
        """)
```

Zeige ein Fragezeichen, solange nichts Anderes (A+B-Knopf gedrückt) passiert.

```
basic.forever(on_forever)
```

```
def on_button_pressed_ab():
```

```
    if randint(0, 1) == 1:
        basic.show_icon(IconNames.YES)
    else:
        basic.show_icon(IconNames.NO)
```

```
input.on_button_pressed(Button.AB, on_button_pressed_ab)
```

Wird A+B am micro:bit gedrückt, mache den Code darunter.

Wähle eine zufällige Zahl zwischen 0 und 1 aus.

Überprüfe, ob die gewählte Zufallszahl gleich 1 ist.

Wenn die **Zufallszahl = 1** (die Frage ist wahr), dann zeige das Häkchen an.

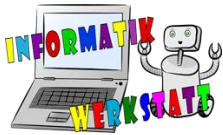
Ist die Zufallszahl **nicht = 1** (also hier 0), dann zeige das X an.

- Ändere das Orakel** aus dem Beispiel darüber so, dass das Wort ‚Ja‘ angezeigt wird, wenn die Zufallszahl gleich 1 ist, und im anderen Fall soll das Wort ‚Nein‘ angezeigt werden.
- Wofür könnte man diese Programmierung **im Alltag** verwenden? Kann man daraus ein Spiel mit Freunden und Freundinnen machen?



- Was passiert bei folgenden Codes? **Welche Zufallswerte sind möglich** und bei welchen wird **ein Häkchen/ein X angezeigt**? Überlege und notiere deine Vermutung.

Code	Mögliche Zufallszahlen	Zahlen, wo Häkchen gezeigt	Zahlen, wo X gezeigt
<pre>def on_button_pressed_ab():     if randint(0, 1) == 1:         basic.show_icon(IconNames.YES)     else:         basic.show_icon(IconNames.NO) input.on_button_pressed(Button.AB, on_button_pressed_ab)</pre>	<p>0, 1</p>	<p>1</p>	<p>0</p>
<pre>def on_button_pressed_ab():     if randint(0, 2) == 1:         basic.show_icon(IconNames.YES)     else:         basic.show_icon(IconNames.NO) input.on_button_pressed(Button.AB, on_button_pressed_ab)</pre>			
<pre>def on_button_pressed_ab():     if randint(1, 2) == 1:         basic.show_icon(IconNames.YES)     else:         basic.show_icon(IconNames.NO) input.on_button_pressed(Button.AB, on_button_pressed_ab)</pre>			
<pre>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>			



<pre>def on_button_pressed_ab():     if randint(1, 1) == 0:         basic.show_icon(IconNames.YES)     else:         basic.show_icon(IconNames.NO) input.on_button_pressed(Button.AB, on_button_pressed_ab)</pre>			
<pre>def on_button_pressed_ab():     if randint(0, 2) == 0:         basic.show_icon(IconNames.YES)     else:         basic.show_icon(IconNames.NO) input.on_button_pressed(Button.AB, on_button_pressed_ab)</pre>			
<pre>def on_button_pressed_ab():     if randint(0, 5) == 3:         basic.show_icon(IconNames.YES)     else:         basic.show_icon(IconNames.NO) input.on_button_pressed(Button.AB, on_button_pressed_ab)</pre>			
<pre>def on_button_pressed_ab():     if randint(0, 5) == 7:         basic.show_icon(IconNames.YES)     else:         basic.show_icon(IconNames.NO) input.on_button_pressed(Button.AB, on_button_pressed_ab)</pre>			
<pre>def on_button_pressed_ab():     if randint(1, 3) &lt; 2:         basic.show_icon(IconNames.YES)     else:         basic.show_icon(IconNames.NO) input.on_button_pressed(Button.AB, on_button_pressed_ab)</pre>			

<pre>def on_button_pressed_ab():     if randint(1, 3) &lt; 3:         basic.show_icon(IconNames.YES)     else:         basic.show_icon(IconNames.NO) input.on_button_pressed(Button.AB, on_button_pressed_ab)</pre>			
<pre>def on_button_pressed_ab():     if randint(1, 10) &lt; 5:         basic.show_icon(IconNames.YES)     else:         basic.show_icon(IconNames.NO) input.on_button_pressed(Button.AB, on_button_pressed_ab)</pre>			
<pre>def on_button_pressed_ab():     if randint(1, 10) &gt; 7:         basic.show_icon(IconNames.YES)     else:         basic.show_icon(IconNames.NO) input.on_button_pressed(Button.AB, on_button_pressed_ab)</pre>			
<pre>def on_button_pressed_ab():     if randint(1, 10) &gt;= 7:         basic.show_icon(IconNames.YES)     else:         basic.show_icon(IconNames.NO) input.on_button_pressed(Button.AB, on_button_pressed_ab)</pre>			

- Gewinnspiel:** Erstelle mit dem micro:bit einen **Würfel**, bei dem man eine zufällige Zahl zwischen 1 und 6 würfelt. Bei der **Zahl 6** soll der **Text ,Gewonnen!'** (oder ein glücklicher Smiley) erscheinen, bei den restlichen Zahlen **nur ein X erscheinen**. Den micro:bit kann man nun in einer Runde von Freunden durchgeben und jeder „zieht“ ein Los (schüttelt den micro:bit) – diese sind entweder gewinnende Lose oder Nieten.
- Großes oder kleines Herz:** Wird der micro:bit geschüttelt, soll er ein Symbol deiner Wahl anzeigen, sonst ein Fragezeichen. Werden beide Tasten (A + B) gleichzeitig gedrückt, soll zufällig entschieden werden, ob ein großes oder ein kleines Herz angezeigt wird.

- *Kniffligere Aufgabe:* Es soll ein **Regen-Sonnenschein-Simulator** programmiert werden. Dabei soll der micro:bit ein beliebiges (selbstgewähltes) Symbol anzeigen – bis er geschüttelt wird. Passiert das, so soll zufällig entschieden und angezeigt werden, ob es morgen regnet oder die Sonne scheint (kreiere ein passendes Symbol). Achte dabei darauf, dass es doppelt so viele Sonnentage geben soll wie Regentage (2 Sonnentage und 1 Regentag im Durchschnitt). Überlege, wie viele Zufallszahlen benötigst du?

