

Grundlagen zur Programmierung des micro:bit in Python II

Ereignissteuerung (aus Eingabe)

Wie schon eingangs erwähnt, besitzt der micro:bit zwei Tasten und gewisse Sensoren, die z.B. Bewegung messen können. Man kann diese genau ausmessen lassen und dann mit Zahlenwerten arbeiten, aber für den Anfang ist reichen die untenstehenden beiden Befehle. Damit kann man schon sehr viel tun und dem micro:bit sagen, was er tun soll, wenn er z.B. geschüttelt, auf den Kopf gestellt oder der Knopf A gedrückt wird. Die Blöcke sind aus der Kategorie  **Eingabe**.

Block	Beschreibung
<p>Code ausführen wenn Knopf button gedrückt</p> <pre>def on_button_pressed_a(): pass input.on_button_pressed(Button.A, on_button_pressed_a)</pre>	<p>Code, der sich innerhalb dieser Methode befindet, wird ausgeführt, wenn ein bestimmter Knopf (hier: Knopf A, der linke) gedrückt wird;</p> <p>Umschreiben auf Taste B oder A+B durch Schreiben von Button.B bzw. Button.AB!</p>
<p>Code ausführen wenn Gesture.Shake</p> <pre>def on_gesture_shake(): pass input.on_gesture(Gesture.Shake, on_gesture_shake)</pre>	<p>Code, der sich innerhalb dieser Methode befindet, wird ausgeführt, wenn der micro:bit eine bestimmte Bewegung wahrnimmt (hier: wenn er geschüttelt wird);</p> <p>Umschalten auf andere Bewegungsarten durch ändern der „Gesture“ (Liste aller Gesten findest du auf der Rückseite)!</p>

Erweitert man das Beispielprogramm von oben, sodass bei Drücken der Taste A ein linkszeigender Pfeil angezeigt wird, so sieht dieses dann so aus:

```
basic.show_string("Hallo!")

def on_forever():
    basic.show_icon(IconNames.HAPPY)
basic.forever(on_forever)

def on_button_pressed_a():
    basic.show_arrow(ArrowNames.WEST)

input.on_button_pressed(Button.A, on_button_pressed_a)
```

Du kannst deine Programme mit dem neuen Wissen erweitern, schaue dir dazu das zugehörige **Arbeitsblatt** an! Vergiss nicht dein Programm im Anschluss herunterzuladen und auf den micro:bit zu verschieben um es auszuprobieren!



Zufall (aus Mathematik)

Der micro:bit hat, wie alle anderen Programmiersprachen auch, die Fähigkeit **Zufallszahlen** auszugeben. Das benötigt man immer wieder und kann damit z.B. einen Würfel programmieren. Dazu brauchen wir den Zufallszahlen-Befehl aus der Kategorie  **Mathematik**

Block	Beschreibung
 randint(0, 10)	wählt zufällig eine Zahl zwischen den beiden angegebenen (hier zwischen 0 und 10) aus; verwendet man an einer Stelle, wo sonst auch eine fixe Zahl hingeschrieben wird.

Will man den micro:bit also so programmieren, dass er einen Würfel ersetzen kann, macht man folgendes: Wenn der micro:bit geschüttelt wird, soll er eine Zahl anzeigen. Diese Zahl möchte man aber nicht fix hinschreiben (z.B. 5), denn diese soll ja zufällig sein. Man schreibt also den Zufallszahl-Befehl an die Stelle, in der man die fixe Zahl geschrieben hätte. Dort trägt man dann die beiden Zahlen ein, zwischen welchen die Zufallszahl ausgewählt werden soll (in unserem Beispiel sind das Zahlen zwischen 1 und 6 - also es sind folgende Zahlen möglich: 1, 2, 3, 4, 5 und 6).

`def on_gesture_shake():`

`basic.show_number(randint(1, 6))`

`input.on_gesture(Gesture.Shake, on_gesture_shake)`



Du kannst deine Programme mit dem neuen Wissen erweitern oder ganz neue schreiben, schau dir dazu das zugehörige **Arbeitsblatt** an! Vergiss nicht dein Programm im Anschluss herunterzuladen und auf den micro:bit zu verschieben um es auszuprobieren!

Liste aller Gesten!

Geste	Beschreibung
Gesture.Shake	Geschüttelt
Gesture.LOGO_UP	Logo nach oben
Gesture.LOGO_DOWN	Logo nach unten
Gesture.SCREEN_UP	Display nach oben
Gesture.SCREEN_DOWN	Display nach unten
Gesture.TILT_LEFT	Nach links neigen
Gesture.TILT_RIGHT	Nach rechts neigen
Gesture.FREE_FALL	Freier Fall
Gesture.THREE_G	3g
Gesture.SIX_G	6g
Gesture.EIGHT_G	8g