

micro:bit Programmierung - Lösungen

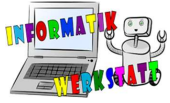


Grundlagen-Blöcke (Schwierigkeit: 😊)

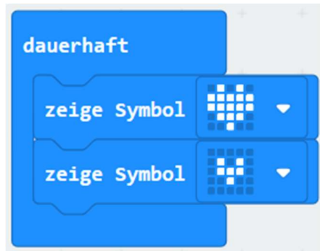
Versuche selbst ein eigenes Programm zu schreiben und am micro:bit auszuführen! Hast du eigene Ideen, was du mit den vorgestellten Blöcken machen kannst? Super, versuche dich gleich daran!

Ideen wären weiters:

Arbeitsauftrag	Lösung
Probiere aus, was passiert, wenn man zwei oder sogar drei Zeig-Symbol-Blöcke in das ‚dauerhaft‘ schiebt!	
Schreibe eine eigene Begrüßung, die beim Starten (sobald Strom da ist) am micro:bit angezeigt wird.	
Erstelle ein eigenes Symbol (z.B. einen Pfeil) und zeige dieses mit den LEDs an.	



„Flashing heart“: Lasse ein Herz blinken und zwar so, dass abwechselnd immer ein großes und ein kleines Herz angezeigt wird.



Vergiss nicht dein Programm im Anschluss herunterzuladen und auf den micro:bit zu verschieben um es auszuprobieren!



Ereignissteuerung (Schwierigkeit: 😊)

Hast du auch für die neu vorgestellten Blöcke Ideen, die du ausprobieren willst? Dann los!

Was du sonst noch probieren kannst:

Arbeitsauftrag	Lösung
<p>Namensausgabe: Beim Drücken der Taste A gib deinen Vornamen, beim Drücken der Taste B deinen Nachnamen aus.</p>	
<p>Symbolgeber: Beim Drücken der Taste A soll ein erfreuter Smiley, bei B ein trauriger Smiley und beim Drücken beider Tasten (A+B) soll ein Herz aufleuchten.</p>	



<p>Kann man das Programm so erweitern, dass wenn keine Taste gedrückt wurde, nichts aufleuchtet (alle LEDs aus sind)?</p> <p><i> Tipp: Verwende den Zeige-LEDs-Block, mit dem man eigene Muster zeichnen kann.</i></p>	
<p>Gegenteil-Schüttler: Wird der micro:bit geschüttelt, so soll er z.B. das Symbol  anzeigen, ansonsten  (das Gegenteil – genau jene LEDs, die vorher nicht geleuchtet haben, leuchten dann).</p>	
<p>Pfeilneigung: Wird der micro:bit nach rechts geneigt, soll ein rechtszeigender Pfeil (→) angezeigt werden und gleiches für links (linkszeigender Pfeil ← bei Linksneigung).</p>	



Kopfstand: Dreht man den micro:bit auf den Kopf, so soll er einen umgedrehten Smiley anzeigen, steht er normal (Augen oben), so soll ein normaler Smiley angezeigt werden.

```
wenn Display nach unten
  zeige LEDs

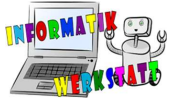
wenn Display nach oben
  zeige Symbol
```

„Der freundliche micro:bit“: Wird der micro:bit nach rechts geneigt, soll er deinen rechten Sitznachbarn begrüßen (z.B. „Hallo Alex!“), nach links geneigt den linken Sitznachbarn. Drückst du A+B, soll der micro:bit dich begrüßen.

```
wenn nach links neigen
  zeige Zeichenfolge "Hallo Max!"

wenn nach rechts neigen
  zeige Zeichenfolge "Hallo Nina!"

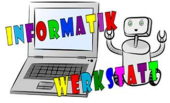
wenn Knopf A+B gedrückt
  zeige Zeichenfolge "Hallo Alex!"
```



Zufall (Schwierigkeit: 😊☺)

Im Beispiel wurde ein Würfel mit Zufallszahlen programmiert. Was könnte man sonst noch damit machen? Man könnte den micro:bit folgende Dinge tun lassen:

Arbeitsauftrag	Lösung
<p>Erstelle ein Orakel, das bei Schütteln mittels Zufalls anzeigt, wie viele Geschenke man beim nächsten Geburtstag bekommen wird. Überlege zuerst zwischen welchen Werten der Zufall auswählen soll und was Sinn machen würde.</p>	
<p>Spiel: Wer ist besser im Raten? Programme den micro:bit so, dass er bei Drücken der Taste A eine zufällige Zahl anzeigt. Du kannst dann vorm Drücken versuchen die Zahl zu erraten. Vielleicht spielt dein Freund oder deine Freundin mit und ihr schaut, wer näher an der Zahl ist?</p> <p><i>Fragen zum Nachdenken: Bei welchen Zufallszahl-Bereichen ist das Spiel leicht, bei welchen schwer? Woran liegt das?</i></p>	
<p>Kann man mit den bisher bekannten Blöcken eine Losziehung machen? Das heißt, man verteilt Lose in der Klasse mit verschiedenen Losnummern (zum Beispiel Durchzählen oder Katalognummern) und möchte dann einen Gewinner oder eine Gewinnerin ziehen. Zwischen welchen Zahlen muss der Zufall hier auswählen?</p>	



Spiel - Orakel: Programmiere den micro:bit als Ja-Nein-Sager: Wenn A+B gedrückt wird, so soll zufällig 0 oder 1 angezeigt werden (0 für Nein, 1 für Ja). Wird nichts gedrückt, so soll ein Fragezeichen angezeigt werden. Man stellt dem micro:bit also dann eine Frage (z.B. Soll ich heute noch ein Eis essen?), drückt A+B – und hat die Antwort.

```
Scratch code for 'Spiel - Orakel':
- 'dauerhaft' block: 'zeige LEDs'
- 'wenn Knopf A+B gedrückt' block: 'zeige Zahl' with 'wähle eine zufällige Zahl zwischen 0 und 1'
```

Kommentiert [MW1]: Zeige Zahl!

Anweisungsgenerator: Schreibe eine Liste mit Aktivitäten von 1 bis 9, zum Beispiel:
1 --> Aufstehen
2 --> Hände in die Höhe
3 --> rechte Hand hoch usw.
und programmiere den micro:bit nun so, dass er beim Drücken der Taste A dir und deinen Freunden und Freundinnen eine Anweisung gibt (z.B. „2“, also Hände hoch). Wer am langsamsten ist, hat verloren und scheidet aus – oder jedes Kind hat 2 Leben und scheidet aus, sobald es keine mehr hat.

```
Scratch code for 'Anweisungsgenerator':
- 'wenn Knopf A gedrückt' block: 'zeige Zahl' with 'wähle eine zufällige Zahl zwischen 1 und 9'
```



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!

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.

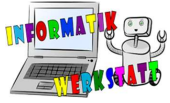
Wenn die **Zufallszahl = 1** (war die Frage), dann zeige das Häkchen an.

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

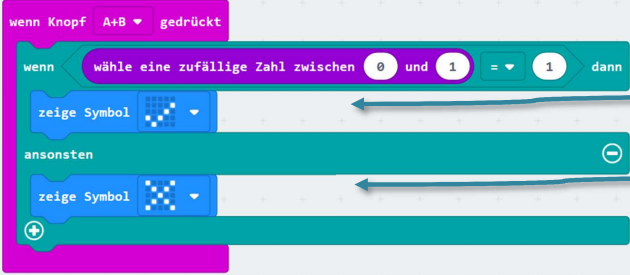
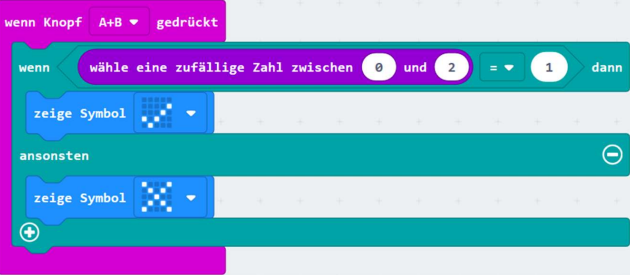
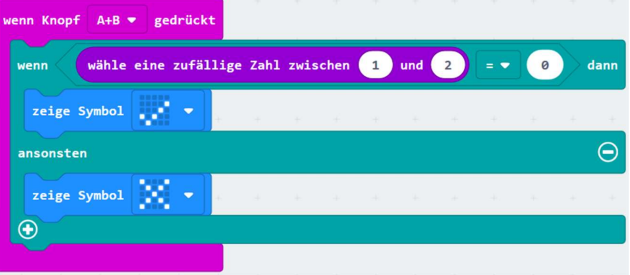
Zeige ein Fragezeichen, solange nichts Anderes/ *Pinkes* (A+B-Knopf gedrückt) passiert.

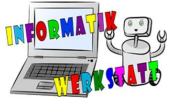
Beispielcode 4 mit Erklärungen

Arbeitsauftrag	Lösung
<p>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.</p>	



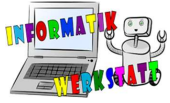
Was passiert bei folgendem Code? Welche Zufallswerte sind möglich und bei welchen wird ein Häkchen/ein X angezeigt? Überlege und notiere deine Vermutung.

Code	Mögliche Zufallszahlen	Zahlen, wo Häkchen gezeigt	Zahlen, wo X gezeigt
	0, 1	1	0
	0, 1, 2	1	0, 2
	1, 2	0	1, 2



SW_Software

	0.1	0	1
	1	1	nie
	0.1.2	0	1.2
	0 - 5	3	1.2. 4.5



	<p>0-5</p>	<p>7</p>	<p>0-5</p>
--	------------	----------	------------

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, ansonsten ein Fragezeichen. Werden beide Tasten (A + B) gleichzeitig gedrückt, soll zufällig entschieden werden, ob ein großes oder ein kleines Herz angezeigt wird.



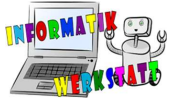
Kniffligere Aufgabe: Es soll ein Regen-Sonnenschein-Simulator programmiert werden. Dabei soll der micro:bit ein beliebiges (selbstgewähltes) Symbol anzeigen – bis er geschüttelt wird. Passiert das, so soll zufällig entschieden und angezeigt werden, ob es morgen regnet oder die Sonne scheint (kreiere 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?



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 roten Block mit dem ‚ändere um‘ den Block mit dem ‚ändere auf‘ genommen hättest. Probiere es anschließend aus!</p>	
<p>Was passiert, wenn man einen der roten Blöcke (‚ändere um‘ oder ‚ändere auf‘) beim Beispiel oben in das dauerhaft verschiebt? Überlege zuerst, tausch dich mit deinem Sitznachbarn aus und probiert es dann gemeinsam aus. Was passiert? Wer hatte Recht und warum ist das so?</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>	



Kann man das so erweitern, dass bei Schütteln der Zähler auf die Zahl 5 gesetzt wird – egal, welcher Wert gerade gespeichert ist?

```
dauerhaft
  zeige Zahl meine_zahl

wenn Knopf A gedrückt
  ändere meine_zahl um -1

wenn Knopf B gedrückt
  ändere meine_zahl um 1

wenn geschüttelt
  ändere meine_zahl auf 5
```

Füge dem vorherigen Programm noch einen Startwert hinzu, es soll immer von -5 weggezählt werden.

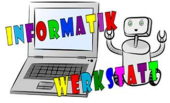
```
beim Start
  ändere meine_zahl auf -5

dauerhaft
  zeige Zahl meine_zahl

wenn Knopf A gedrückt
  ändere meine_zahl um -1

wenn Knopf B gedrückt
  ändere meine_zahl um 1

wenn geschüttelt
  ändere meine_zahl auf 5
```



<p>Dreierzähler: Programmiere einen Zähler, der bei Drücken der Taste A immer um 3 erhöht (0 -> 3 -> 6 -> 9 -> ...).</p>	
<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>	



Zufallsschütteln: Ändere das Programm (mit den Zweier-/Dreierschritten) so, dass beim Schütteln die Variable auf einen zufälligen Wert gesetzt wird.

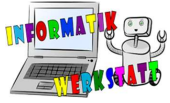
```
dauerhaft
zeige Zahl meine_zahl

wenn Knopf A gedrückt
ändere meine_zahl um -3

wenn Knopf B gedrückt
ändere meine_zahl um 2

wenn Knopf A+B gedrückt
ändere meine_zahl auf 0

wenn geschüttelt
ändere meine_zahl auf wähle eine zufällige Zahl zwischen 0 und 10
```

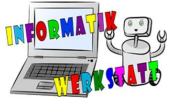


Zielwert-Blinken: Füge dem Programm noch einen Zielwert hinzu! Wird die Zahl 5 erreicht (Variablenwert = 5), so soll der micro:bit lustig blinken. Wie muss man die Tasten A und B drücken, dass man die Zahl 5 erreicht? Schaffst du es?

```
dauerhaft
wenn meine_zahl = 5 dann
  zeige Symbol [Smiley]
  zeige Symbol [Smiley]
ansonsten
  zeige Zahl meine_zahl
wenn Knopf A gedrückt
  ändere meine_zahl um -3
wenn Knopf B gedrückt
  ändere meine_zahl um 2
wenn Knopf A+B gedrückt
  ändere meine_zahl auf 0
wenn geschüttelt
  ändere meine_zahl auf wähle eine zufällige Zahl zwischen 0 und 10
```

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 angezeigt werden und wenn er am Kopf steht, ein erschrockener Smiley.

```
wenn Knopf A+B gedrückt
  zeige Zahl meine_zahl
wenn Display nach oben
  zeige Symbol [Smiley]
wenn Display nach unten
  zeige Symbol [Smiley]
  ändere meine_zahl um 1
```

Ändere das Programm so, dass bei -10 zu Zählen begonnen wird. (Bei einmaligem Kopfüberdrehen soll also bei A+B-Druck dann -9 angezeigt werden.)

```
beim Start
  ändere meine_zahl auf -10

wenn Knopf A+B gedrückt
  zeige Zahl meine_zahl

wenn Display nach oben
  zeige Symbol [Symbol]

wenn Display nach unten
  zeige Symbol [Symbol]
  ändere meine_zahl um 1
```

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).

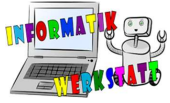
```
beim Start
  ändere meine_zahl auf -10

wenn Knopf A+B gedrückt
  zeige Zahl meine_zahl

wenn Display nach oben
  zeige Symbol [Symbol]

wenn Display nach unten
  zeige Symbol [Symbol]
  ändere meine_zahl um 1

dauerhaft
  wenn meine_zahl = 7 dann
    zeige Symbol [Symbol]
    zeige Symbol [Symbol]
```



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.

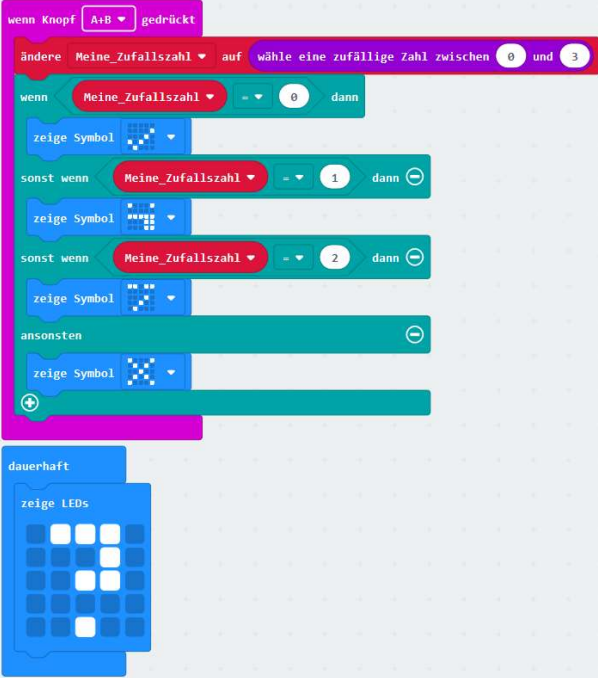
```
beim Start
  ändere Meine_Zufallszahl auf wähle eine zufällige Zahl zwischen 3 und 17
  zeige Zahl Meine_Zufallszahl

dauerhaft
  wenn meine_zahl = Meine_Zufallszahl dann
    zeige Symbol [star symbol]
  ansonsten
    zeige Symbol [smiley symbol]
  wenn geschüttelt
    ändere meine_zahl um 1
  wenn Knopf A+B gedrückt
    zeige Zahl meine_zahl
```



Verzweigungen II (Schwierigkeit: 😊😊😊)

Du kannst nun Fallunterscheidungen mit mehr als zwei Fällen in deinen Programmen verwenden! Was könnte man damit umsetzen? Probiere dich selbst an deinen Ideen – oder den Aufgabenideen hier.

Arbeitsauftrag	Lösung
<p>Erweitere das Orakelbeispiel mit drei Fällen (aus dem Vorzeige-Beispiel) mit noch einem weiteren Fall. Es soll also vier Fälle geben: ‚Ja‘, ‚Nein‘, ‚Weiß nicht‘ sowie z.B. ‚Frag wen anderen‘. Wähle für den vierten Fall ein geeignetes Symbol aus.</p>	 <p>The image shows a Scratch script for an oracle program. It starts with a 'wenn Knopf A+B gedrückt' (when button A+B is pressed) event block. This is followed by an 'ändere Meine_Zufallszahl auf wähle eine zufällige Zahl zwischen 0 und 3' (set Meine_Zufallszahl to a random number between 0 and 3) block. The script then uses a series of 'wenn' (if) and 'sonst wenn' (else if) blocks to check the value of 'Meine_Zufallszahl'. For each value (0, 1, 2), there is a 'dann' (then) block containing a 'zeige Symbol' (show symbol) block with a specific symbol. There is also an 'ansonsten' (otherwise) block at the end, which contains another 'zeige Symbol' block with a different symbol. The script ends with a 'dauerhaft zeige LEDs' (permanently show LEDs) block.</p>

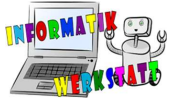


Ändere das große Orakel so, dass anstatt der Symbole der entsprechende Text ausgegeben wird.

```
wenn Knopf A+B gedrückt
  ändere Meine_Zufallszahl auf wähle eine zufällige Zahl zwischen 0 und 3
  wenn Meine_Zufallszahl = 0 dann
    zeige Text "Ja!"
  sonst wenn Meine_Zufallszahl = 1 dann
    zeige Text "Frag wen Anderen!"
  sonst wenn Meine_Zufallszahl = 2 dann
    zeige Text "Ist mir egal!"
  ansonsten
    zeige Text "Nein!"
dauerhaft
  zeige LEDs
```

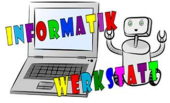
Kann man mit dem jetzigen Wissen das Spiel Schere-Stein-Papier mit dem micro:bit programmieren? Also so, dass wenn man ihn schüttelt, dieser entweder eine Schere, einen Stein oder ein Symbol für Papier anzeigt? Wenn das gelingt, könnte man gegen den micro:bit spielen!

```
wenn geschüttelt
  ändere Meine_Zufallszahl auf wähle eine zufällige Zahl zwischen 0 und 2
  wenn Meine_Zufallszahl = 0 dann
    zeige Symbol [Schere]
  sonst wenn Meine_Zufallszahl = 1 dann
    zeige Symbol [Stein]
  ansonsten
    zeige Symbol [Papier]
dauerhaft
  zeige LEDs
```



Kennst du die Erweiterung des Spiels? Stein-Papier-Schere-Echse-Spock? Dabei kommen zwei weitere Möglichkeiten – Echse und Spock – hinzu und die Regeln werden dafür erweitert.

```
wenn geschüttelt
  ändere Meine_Zufallszahl auf wähle eine zufällige Zahl zwischen 0 und 4
  wenn Meine_Zufallszahl = 0 dann
    zeige Symbol
  sonst wenn Meine_Zufallszahl = 1 dann
    zeige Symbol
  sonst wenn Meine_Zufallszahl = 2 dann
    zeige Symbol
  sonst wenn Meine_Zufallszahl = 3 dann
    zeige LEDs
  ansonsten
    zeige Text v
dauerhaft
  zeige LEDs
```



SW_Software

„Würfel mit Augen“:
Programmiere einen
Würfel, der beim Schütteln
zufällig eine Würfelzahl
zeigt (mit den Augen) und
ansonsten ein Symbol
blinkt. Schaffst du es,
dass die zuletzt gewürfelte
Zahl als Ziffer durch Druck
der Taste A nochmal
angesehen werden kann?
Und wie muss man den
Code erweitern, dass der
micro:bit nach einem
gewürfelten Sechser
anders blinkt (z.B. Sterne)?

```
wenn geschüttelt
  ändere Meine_Zufallszahl um zufällig eine zufällige Zahl zwischen 1 und 6
  wenn Meine_Zufallszahl == 1 dann
    zeige LED0
  sonst wenn Meine_Zufallszahl == 2 dann
    zeige LED0
    zeige LED1
  sonst wenn Meine_Zufallszahl == 3 dann
    zeige LED0
    zeige LED1
    zeige LED2
  sonst wenn Meine_Zufallszahl == 4 dann
    zeige LED0
    zeige LED1
    zeige LED2
    zeige LED3
  sonst wenn Meine_Zufallszahl == 5 dann
    zeige LED0
    zeige LED1
    zeige LED2
    zeige LED3
    zeige LED4
  ansonsten
    zeige LED0
    zeige LED1
    zeige LED2
    zeige LED3
    zeige LED4
    zeige LED5

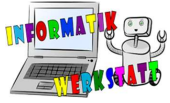
wenn gedrückt A
  zeige Zahl Meine_Zufallszahl

wenn Sechser
  zeige LED0
  zeige LED1
  zeige LED2
  zeige LED3
  zeige LED4
  zeige LED5
```



Expertenaufgabe: Erstelle einen Anweisungsgenerator! Der micro:bit soll dir Anweisungen geben, z.B. A (als ‚Drücke Taste A‘), B (für Taste B), II (für A+B), S (für Schütteln) usw. Warte zwischen den einzelnen Anweisungen 3,5 Sekunden (verwende den ‚pausiere‘-Block dazu findet sich unter Grundlagen –Angabe in Millisekunden!).
Erweiterung: Lass den micro:bit anzeigen, ob das, was getan wurde (z.B. Schütteln) richtig war – z.B. durch Häkchen und X.

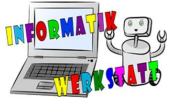
```
Scratch code blocks for a micro:bit instruction generator. The main loop starts with 'dauerhaft' and 'ändere Meine_Zufallszahl auf wähle eine zufällige Zahl zwischen 0 und 3'. It then uses 'wenn' blocks to check the random number: 0 shows 'A', 1 shows 'B', 2 shows 'S', and 'ansonsten' shows 'S'. A 'pausiere (ms) 3500' block follows. Below are three event-driven blocks: 'wenn Knopf A+B gedrückt' (shows 'II' if number is 2), 'wenn Knopf B gedrückt' (shows 'B' if number is 1), and 'wenn Knopf A gedrückt' (shows 'A' if number is 0). Each event block also shows 'X' or '✓' symbols.
```



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.	
Erweitere das Programm so, dass der micro:bit danach ein Herz dauerhaft blinken lässt.	
Schaffst du es, mit einer Schleife von 0 bis 12 zu zählen?	



Schreibe ein Programm, das zuerst achtmal ein Quadrat blinken lässt und danach von 0 bis 8 zählt.

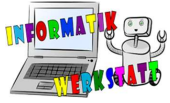
```
beim Start
  8 -mal wiederholen
  mache
    zeige Symbol [Quadrat]
    zeige Symbol [Quadrat]
  für Index von 0 bis 8
  machen
    zeige Zahl [Index]
```

Erweitere das Orakel so, dass du vor dem Start des Programms, das Fragezeichen dreimal blinken lässt bevor es durchgehend leuchtet

```
beim Start
  3 -mal wiederholen
  mache
    zeige LEDs [Fragezeichen]
    zeige LEDs [Fragezeichen]
    zeige LEDs [Fragezeichen]
  dauerhaft
    zeige LEDs [Leuchtendes Quadrat]

wenn Knopf A+B gedrückt
  wenn 0 = wähle eine zufällige Zahl zwischen 0 und 1 dann
    zeige Symbol [Fragezeichen]
  ansonsten
    zeige Symbol [Leuchtendes Quadrat]
```

Kommentiert [MW2]: Anmerkung Marianne, woher bekommen die SuS das Orakelprogramm?



Schreibe ein Programm, das herunterzählt – z.B. von 3 bis 0

```
beim Start
  ändere meine_zahl auf 3
  3 -mal wiederholen
  mache
    zeige Zahl meine_zahl
    ändere meine_zahl um -1
  zeige Zahl meine_zahl
```

Setze obiges Programm mit einer While-Schleife um. Was muss man ändern, welche Bedingung brauchst du (z.B. wiederhole solange größer als ...)

```
beim Start
  ändere meine_zahl auf 3
  während meine_zahl > 0
  mache
    zeige Zahl meine_zahl
    ändere meine_zahl um -1
  zeige Zahl meine_zahl
```

Schaffst du es ein Programm zu schreiben, das eine bestimmte Zahl 2-mal blinken lässt und anschließend von dieser Zahl herunterzählt bis zur 0?

```
beim Start
  ändere meine_zahl auf 5
  2 -mal wiederholen
  mache
    zeige Zahl meine_zahl
    zeige LEDs
  während meine_zahl > 0
  mache
    zeige Zahl meine_zahl
    ändere meine_zahl um -1
  zeige Zahl meine_zahl
```



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.

```

beim Start
  während Start = 0
    mache
      zeige Symbol [Ente]
      zeige Symbol [Ente]
  zeige Text "Los!"

dauerhaft
  zeige LEDs [Ente]

wenn Knopf A+B gedrückt
  ändere Start auf 1

wenn geschüttelt
  wenn 0 = wähle eine zufällige Zahl zwischen 0 und 1 dann
    zeige Symbol [Ente]
  ansonsten
    zeige Symbol [Ente]
  
```

Schreibe ein Programm, in dem du einstellen kannst, wie oft ein bestimmtes Symbol, z.B. die Ente, blinken soll:

Schreibe dazu zuerst ein Zähler-Programm, bei dem mit A-Druck eins nach unten und bei B-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 „dauerhaft“ bleibt frei). Bei A+B-Druck soll die Ente zunächst einfach 3-mal blinken.

```

wenn Knopf A+B gedrückt
  mache
    zeige Symbol [Ente]
    zeige LEDs [Ente]
  3 -mal wiederholen

wenn Knopf A gedrückt
  ändere meine_zahl um -1
  zeige Zahl meine_zahl

wenn Knopf B gedrückt
  ändere meine_zahl um 1
  zeige Zahl meine_zahl
  
```



Schreibe ein Programm, in dem du einstellen kannst, wie oft ein bestimmtes Symbol, z.B. die Ente, blinken soll:

Ändere nun das Programm so, dass wenn du A+B drückst, die Ente genauso oft wie eingestellt, blinkt.

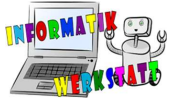
```
wenn Knopf A+B gedrückt
  mache
    meine_zahl -mal wiederholen
    zeige Symbol [Ente]
    zeige LEDs
wenn Knopf A gedrückt
  ändere meine_zahl um -1
  zeige Zahl meine_zahl
wenn Knopf B gedrückt
  ändere meine_zahl um 1
  zeige Zahl meine_zahl
```

Schaffst du es, einen einstellbaren Countdown zu schreiben?

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.

Erweitere das Programm dann so, dass nach dem Blinken des Symbols von der Zahl (dem Zählerwert) bis 0 gezählt wird.

```
beim Start
  zeige Zahl meine_zahl
wenn Knopf A+B gedrückt
  3 -mal wiederholen
  mache
    zeige Zahl meine_zahl
    zeige Text
  während meine_zahl > 0
  mache
    zeige Zahl meine_zahl
    ändere meine_zahl um -1
  zeige Zahl meine_zahl
wenn Knopf A gedrückt
  ändere meine_zahl um -1
  zeige Zahl meine_zahl
wenn Knopf B gedrückt
  ändere meine_zahl um 1
  zeige Zahl meine_zahl
```



Expertenaufgabe:
Erweitere den Countdown
so, dass er auch für
Minuszahlen geht

```
beim Start
  zeige Zahl meine_zahl

wenn Knopf A+B gedrückt
  3 -mal wiederholen
  mache
    zeige Zahl meine_zahl
    zeige Text ""
  während meine_zahl ≠ 0
    mache
      wenn meine_zahl > 0 dann
        zeige Zahl meine_zahl
        ändere meine_zahl um -1
      ansonsten
        zeige Zahl meine_zahl
        ändere meine_zahl um 1
    zeige Zahl meine_zahl

wenn Knopf A gedrückt
  ändere meine_zahl um -1
  zeige Zahl meine_zahl

wenn Knopf B gedrückt
  ändere meine_zahl um 1
  zeige Zahl meine_zahl
```