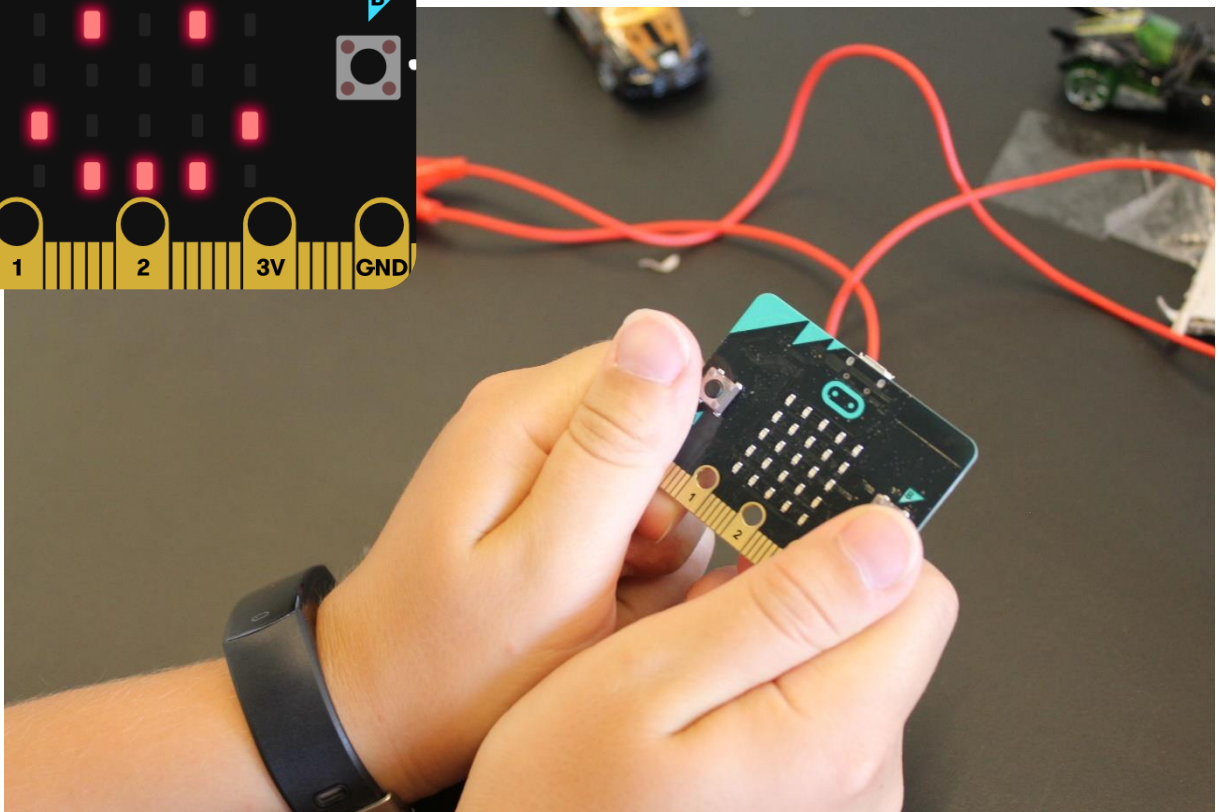
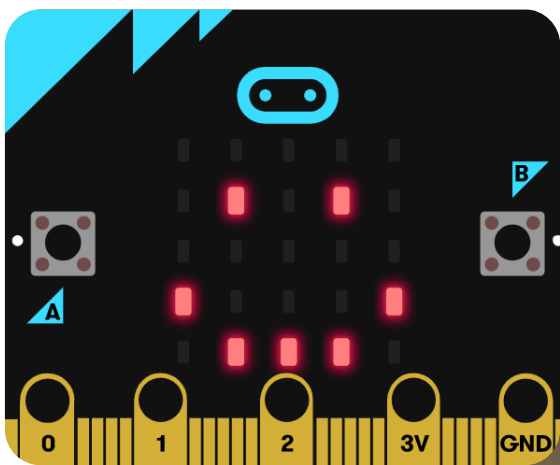


# micro:bit Challenge Cards



Grundlagen und kleine Projekte für micro:bit



Diese Challenge Cards wurden im Rahmen des iMake-IT Projektes der Pädagogischen Hochschule Schwyz für den *Open Education Day 2021* entwickelt.

Weitere Informationen unter:

[phsz-facile.ch/imake-it](https://phsz-facile.ch/imake-it)  
[imake-it.ch](https://imake-it.ch)

## Impressum

Version 1.2 (April 2021)

Autoren: Seraina Betschart, Jason Curtins

[imake-it@phsz.ch](mailto:imake-it@phsz.ch)

Die ursprüngliche Version dieses Dokumentes wurde im März 2021 erstellt, basierend auf anderen Versionen der iMakeIT Challenge Cards.

Bilder, Grafiken, Screenshots: PHSZ

Icons: S.20 Misc Dice by glitch (openclipart.org), S.23 Compass Rose by Firkin (openclipart.org), S.27 Blue Robot by Scout (openclipart.org), S.30 Piano Keyboard by GDJ (openclipart.org), S.33 Rock-Paper-Scissors by uoresch (openclipart.org)



Namensnennung  
Weitergabe unter gleichen Bedingungen

# Inhalt

## Grundlagen

Nummer	Challenge	Inhalt	Seite
1	Hello World! 	Wir lernen einen visuellen Output, das LED-Display des micro:bits kennen und benutzen "beim Start" und "dauerhaft"-Blöcke.	6
2	Knöpfe A und B benutzen	Wir brauchen die zwei eingebauten Knöpfe A & B von dem micro:bit. Zudem arbeiten wir mit «Wenn, dann»-Blöcken.	8
3*	Wir machen Musik 	Wir brauchen den Lautsprecher und lernen die vorprogrammierten Melodieblöcke des makecode Programms kennen.	10
4*	Programmiere deine eigene Musik	Wir programmieren eigene Melodien Ton für Ton und arbeiten mit Schleifen.	12
5	Symbole mit den Fingerspitzen verändern	Wir testen die Leitfähigkeit unseres eigenen Körpers.	14
6	Temperatur messen	Wir lernen einen weiteren eingebauten Sensor des micro:bit kennen.	16
7	Funk mal was!  	Wir lassen zwei micro:bits über Funk miteinander sprechen.	18



# Projekte

Nummer	Challenge	Gelernte Konzepte	Seite
1	Die Würfel sind gefallen	Wir programmieren uns einen eigenen Würfel, der durch Schütteln eine Zufallszahl anzeigt.	20
2	Wo ist Norden?	Wir entdecken den eingebauten Kompass...	23
3*	Mein eigener kleiner Roboter	Ein Roboter der bei Klatschen lächelt? Dich begrüsst, wenn du nach Hause kommst? Nichts leichter als das...	27
4*	Tischklavier	Dieses Projekt zeigt dir, wie du ganz einfach ein kleines Klavier mit micro:bit und Alufolie baust.	30
5	Papier schlägt Stein. Oder so...	Mit micro:bit lässt sich auch ein Schere, Stein, Papier programmieren.	33
Lösungen zu den Zusatzaufgaben			38

\*Für diese Challenges wird die Weiterentwicklung BBC micro:bit V2 benötigt, bzw. muss der micro:bit mit externen Inputs und Outputs ergänzt werden.

## 1

# Hello World!

Baue den untenstehenden Code nach. Im blauen Kasten «verwendete Befehlsgruppen» siehst du, wo du die Befehle dafür findest.

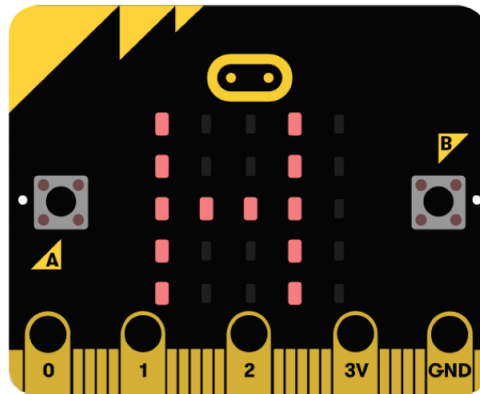
VERWENDETE BEFEHLSGRUPPEN

Grundlagen

beim Start

zeige Text " Hello! "

Speichere den Code und lade ihn auf den micro:bit.



Hinweis:

Wenn du das Programm neu starten willst, drücke auf die Reset-Taste auf der Rückseite des micro:bits.

Wenn du alles richtig gemacht hast, läuft der Text einmal auf dem micro:bit Display.



## Challenge 1a

Verändere das Programm so, dass ein anderer Text auf dem LED-Display angezeigt wird.

Hinweis:

Um ein neues oder verändertes Programm auf dem micro:bit zu testen, muss es jedes Mal von Neuem auf den micro:bit hochgeladen werden. Dabei wird das alte Programm überschrieben.

## Challenge 1b

Passe das Programm an, damit der Text unendlich lange über das Display läuft. Im blauen Kasten siehst du, wo du die Befehle dafür findest.

.....  
VERWENDETE BEFEHLSGRUPPEN

 Grundlagen

.....

Hinweis:

Um nicht mehr benötigte Programmierenteile zu löschen, ziehe sie mit der Maus nach links, bis ein Mülleimersymbol erscheint.

## 2

## Knöpfe A und B benutzen

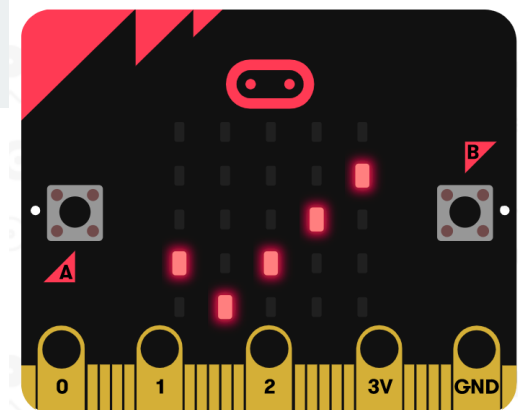
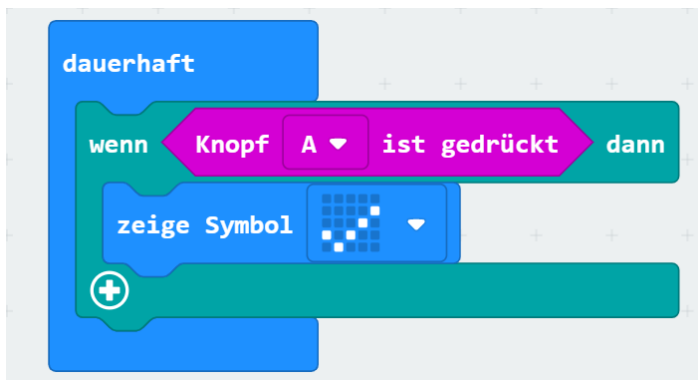
Baue den untenstehenden Code nach.

VERWENDETE BEFEHLSGRUPPEN

Grundlagen

Logik

Eingabe



Wenn man Knopf A drückt, erscheint ein Häkchen-Symbol auf dem Display.



### Challenge 2a

Erweitere das Programm, sodass auch für Knopf B ein Symbol gezeigt wird.

---

### Challenge 2b

Gestalte ein eigenes Symbol für Knopf A.

.....  
VERWENDETE BEFEHLSGRUPPEN  
.....




.....

### Challenge 2c

Verwende eine Bewegung als Input, anstelle von Knopf A und B.

.....  
VERWENDETE BEFEHLSGRUPPEN  
.....




.....

# 3 Wir machen Musik

Baue den untenstehenden Code nach.

.....  
 VERWENDETE BEFEHLSGRUPPEN  
 .....  

beim Start + + + + + + + +  
 Beginne Melodie Dadadum ▼ Wiederhole einmal ▼  
 + + + + + + + +

Der Lautsprecher des micro:bits spielt die Melodie *Dadadum* einmal.



### Challenge 3a

Lasse die Melodie unendlich lange laufen.

---

### Challenge 3b

Programmiere zwei unterschiedliche Melodien, die du mit Knopf A und Knopf B abspielen kannst.

.....  
VERWENDETE BEFEHLSGRUPPEN

 Eingabe

 Musik

.....

---

## 4

## Programmiere deine eigene Musik

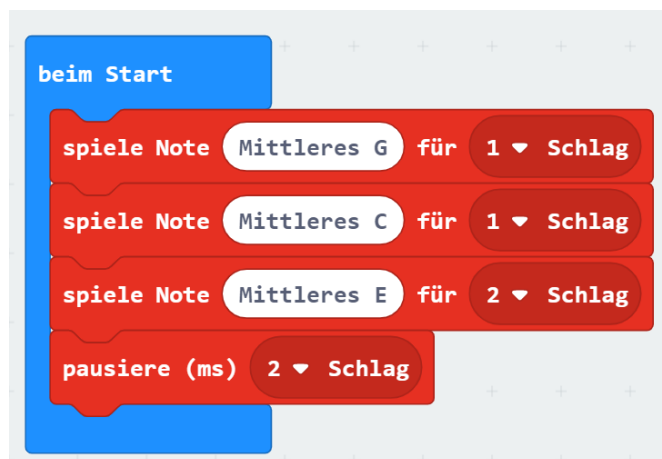
Baue den untenstehenden Code nach.

.....  
VERWENDETE BEFEHLSGRUPPEN

Grundlagen

Musik

.....



The image shows a Scratch script starting with a 'beim Start' block. It contains four 'spiele Note' blocks and one 'pausiere (ms)' block. The first three 'spiele Note' blocks are for 'Mittleres G', 'Mittleres C', and 'Mittleres E' respectively, each with a duration of 1 and a volume of 'Schlag'. The fourth 'spiele Note' block is for 'Mittleres E' with a duration of 2 and a volume of 'Schlag'. The 'pausiere (ms)' block has a duration of 2 and a volume of 'Schlag'.

Der Lautsprecher spielt die programmierte Melodie einmal.



## Challenge 4a

Lasse die Melodie 8x nacheinander laufen. Versuche, möglichst wenige Bausteine zu verwenden.

.....  
VERWENDETE BEFEHLSGRUPPEN

 Grundlagen

 Schleifen

 Musik

.....

---

## Challenge 4b

Programmiere deine eigene Musik und lasse sie mit Knopf A und B ablaufen.

---

# 5 Symbole mit den Fingerspitzen verändern

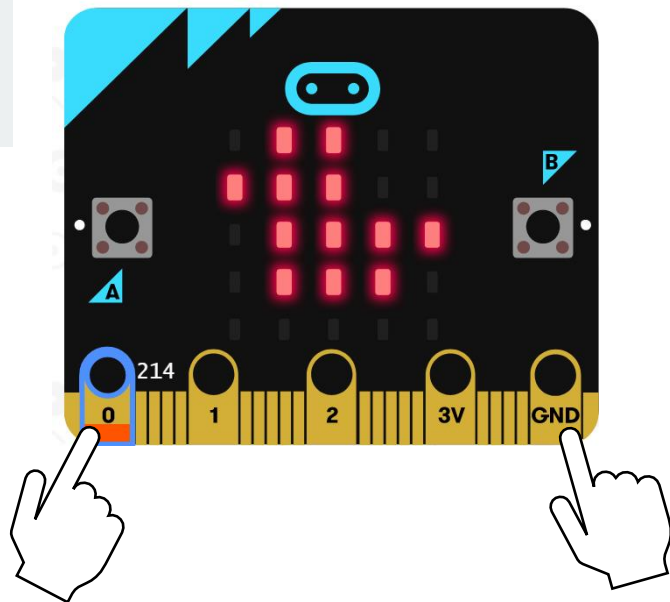
Baue den untenstehenden Code nach.

VERWENDETE BEFEHLSGRUPPEN

Grundlagen    Eingabe

wenn Pin P0 gedrückt

zeige Symbol



Wenn Pin 0 und Ground (GND) gedrückt werden, schliesst sich der Stromkreis und das LED-Display zeigt ein Küken.



### Challenge 5a

Programmiere zusätzlich ein anderes Symbol für Pin 1.

---

### Challenge 5b

Programmiere für Pin 2 einen anderen Output als ein Symbol (z.B. Musik).

---

## 6

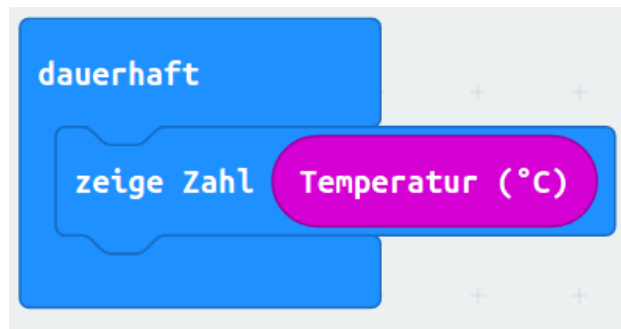
## Temperatur messen

Baue den untenstehenden Code nach.

VERWENDETE BEFEHLSGRUPPEN

Grundlagen

Eingabe



Verändere die Temperatur indem du den micro:bit...

- in die Sonne/an den Schatten legst.
- Den Daumen auf den Wärmesensor legst.
- über eine kleine Kerze (**mit genügend Abstand und nur kurz!!**) haltest (Anmerkung: direkte und zu grosse Hitze können den micro:bit beschädigen).

Hinweis: Der Wärmesensor ist der Prozessor auf der Unterseite des

Der micro:bit zeigt auf dem Display die aktuelle Temperatur an.



## Challenge 6a

Verändere den Code so, dass ein Bild angezeigt wird, wenn die Temperatur über eine gewisse Grenze (z.B. 25°C) steigt und ein anderes, falls die Temperatur darunter liegt.

---

## Challenge 6b (Advanced)

Programmiere den micro:bit so, dass er einen Peil nach oben anzeigt, wenn die Temperatur am Steigen ist und einen Pfeil nach unten, wenn die Temperatur am Sinken ist.

Hinweis: Für diese Challenge brauchst du Variablen, um die Temperatur zwischenzuspeichern und mit der veränderten Temperatur zu vergleichen.

# 7 Funk mal was!

Baue die untenstehenden Codes nach. Du brauchst zwei micro:bits.

Hinweis:

Speichere je einen Code auf einen micro:bit. Merke dir, welcher micro:bit dein «Sender» ist, der also die Befehle gibt, und welches der «Empfänger» ist, der die Befehle anzeigt.

Da wir einen Text empfangen, muss  gebraucht werden.

## VERWENDETE BEFEHLSGRUPPEN

Grundlagen

Logik

Funk

Eingabe

### micro:bit 1 -> Sender

```

beim Start
  setze Funkgruppe auf 20
  zeige LEDs

dauerhaft
  wenn Knopf A ist gedrückt dann
    sende Text "Herz" über Funk
  
```

### micro:bit 2 -> Empfänger

```

beim Start
  setze Funkgruppe auf 20
  zeige LEDs

wenn Text empfangen receivedString
  wenn receivedString = "Herz" dann
    zeige Symbol
    pausiere (ms) 2000
    Bildschirminhalt löschen
  
```

Wenn Knopf A beim Sender micro:bit gedrückt wird, zeigt das Empfänger micro:bit ein Herz.

## Challenge 7a

Ändere das Symbol, das beim Empfangen des Texts angezeigt wird.

Hinweis:

Dem micro:bit ist es egal, ob der empfangene Text mit dem Symbol, das es anzeigt, übereinstimmt. Man könnte auch den Text «Baum» senden und das micro:bit programmieren, dass es ein Auto zeigt. Wichtig ist, dass der Sender genau den gleichen Text sendet, auf den der Empfänger wartet.

## Challenge 7b


Programmiere zusätzlich ein anderes Symbol, das über die Taste B gesendet werden kann.

## Bonus: Challenge 7c

Verwende «Sende Zahl» anstatt «Sende Text», um dem Empfänger mitzuteilen, welches Symbol er anzeigen soll.

Hinweis:

wenn Text empfangen `receivedString` und  werden gebraucht, um Text zu empfangen.

wenn Zahl empfangen `receivedNumber` und  werden gebraucht, um eine Zahl zu empfangen.

# 1 Die Würfel sind gefallen



Für viele Spiele werden Würfel benötigt. Diese kleinen Dinger sind simpel und praktisch, doch haben sie auch einen Nachteil. Ständig rollen sie vom Tisch runter und landen unter Schränken, sodass wir beim Versuch, sie wieder hervor zu klauben, gleich zwei Spinnennetze und eine Staubvergiftung mitsammeln. Ach wie gut, haben wir unseren kleinen micro:bit mit dem sich ganz leicht ein Würfel programmieren lässt...

## Startfragen

1. Wie lösen wir ein Würfelergebnis aus? Welche Inputs haben wir zur Verfügung?
2. Wie zeigen wir das Würfelergebnis mit dem micro:bit an?
3. Wir wollen, dass der micro:bit eine Zufallszahl anzeigt. In welcher Befehlsgruppe finden wir etwas passendes?

## Code

.....

VERWENDETE BEFEHLSGRUPPEN

.....

Grundlagen
Eingabe
Mathematik

.....

Diese zwei Blöcke verzögern das Anzeigen der Zahl, damit man die versch. Würfel-ergebnisse besser abgrenzen kann.

wenn geschüttelt

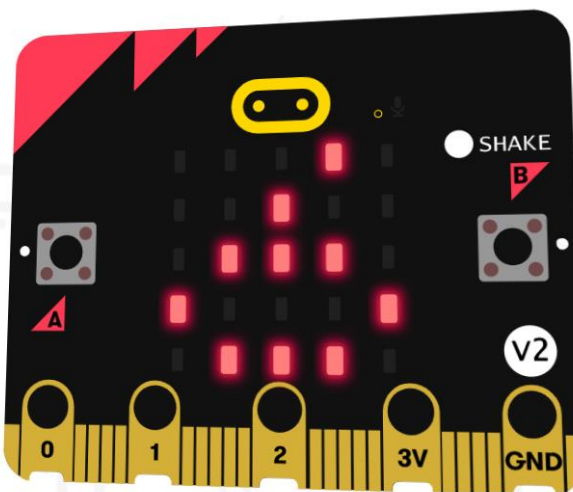
pausiere (ms) 500

zeige Symbol

zeige Zahl wähle eine zufällige Zahl von 1 bis 6

Unser Würfel-ergebnis wird durch Schütteln ausgelöst.

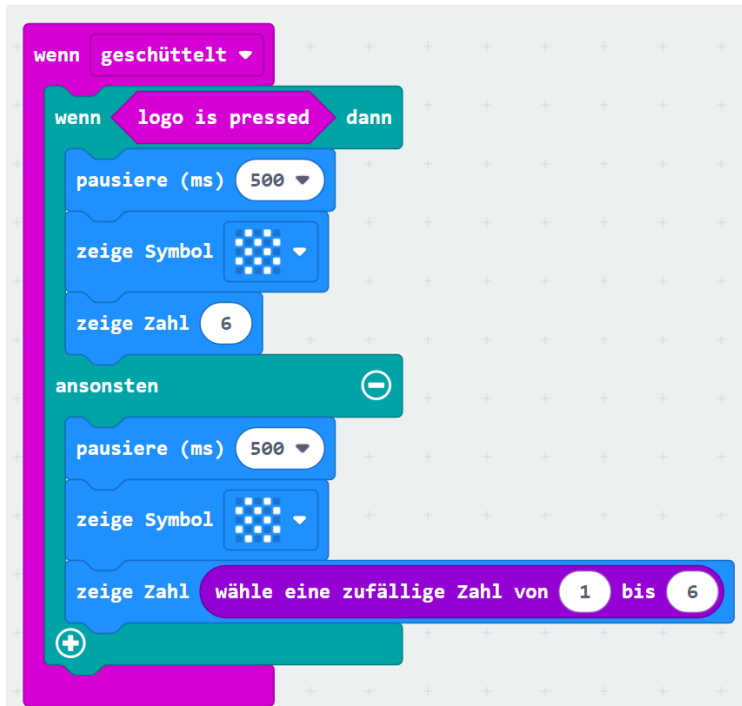
Hier wählen wir 1 bis 6, wie ein klassischer Würfel.



## Zusatzaufgaben

1. Aus unserem sechseitigen Würfel können wir ganz einfach einen 8-/10- oder 100-seitigen Würfel machen. Wir müssen nur eine Zahl verändern.

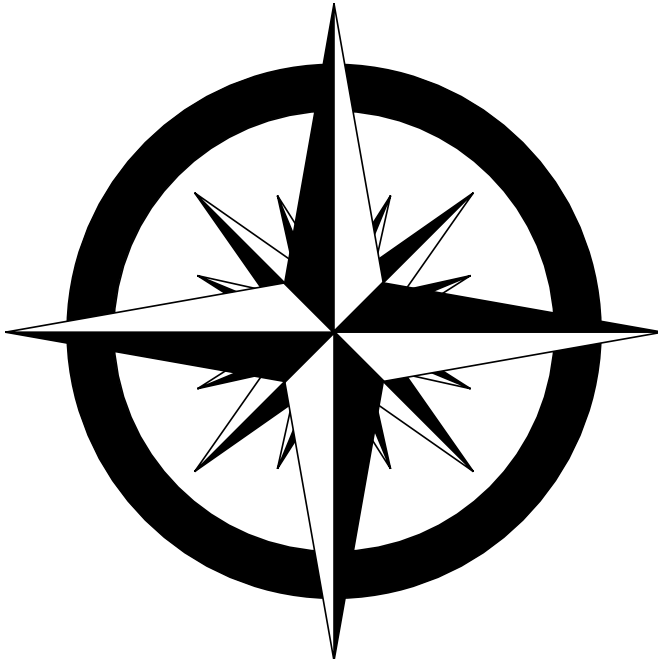
2. Erweitere deinen Code folgendermassen:



Was passiert?

## 2

## Wo ist Norden?



Früher hat man ihn öfters gebraucht. Heute hat man ihn vielleicht noch auf dem Smartphone als App. Er hilft z.B. auf hoher See oder wenn man die alte Gott-hardstrecke mit dem Zug fährt, kann man mit ihm die Kehrtunnels "erleben".

Es gibt ein bekanntes Lied in dem er als "schönste Blume" besungen wird. Ja die Windrose oder auch Kompass genannt. Und für uns wird es jetzt Zeit, unseren eigenen Kompass zu programmieren (so für den Fall, dass uns doch mal irgendwo in der Wildnis das Handy aussteigt und wir praktischerweise den micro:bit zur Hand haben).

### Startfragen

1. Was braucht ein herkömmlicher Kompass, um die Himmelsrichtung anzuzeigen?
2. Der micro:bit hat keine Nadel. Wie kann man hier eine Richtung anzeigen?
3. Eine Kompassnadel dreht sich immer im Kreis. Wie viele Grad hat ein Kreis und bei welchen Werten sind dabei Norden, Osten, Süden und Westen?

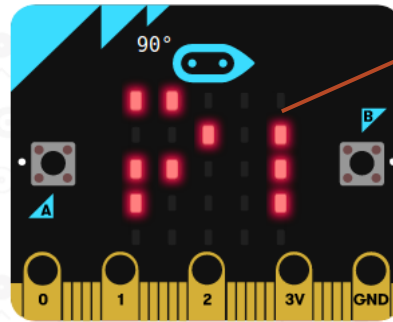
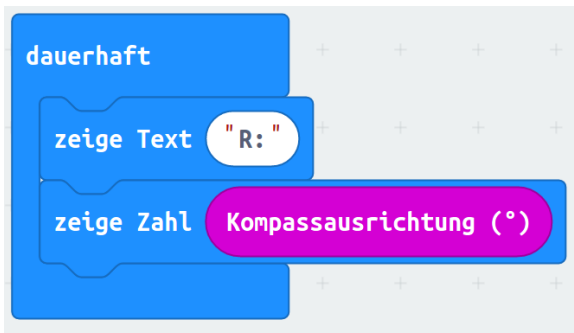
## Code

Aufgabe 1: Auf dem micro:bit soll die aktuelle Richtung als Zahl angezeigt werden.

### VERWENDETE BEFEHLSGRUPPEN

Grundlagen

Eingabe



"R: 90" läuft über das Display (Richtung = 90°)

*Hinweis: Damit der Kompass funktioniert, muss das micro:bit kalibriert werden. Das micro:bit bringt bei Verwendung einen Anweisungstext (auf Englisch). Nach der Anweisung muss das micro:bit in alle Richtungen geneigt werden bis alle roten LEDs leuchten.*

Aufgabe 2: Auf dem micro:bit soll die aktuelle Richtung mit Buchstaben (N, O, S, W) angezeigt werden.

VERWENDETE BEFEHLSGRUPPEN

Grundlagen

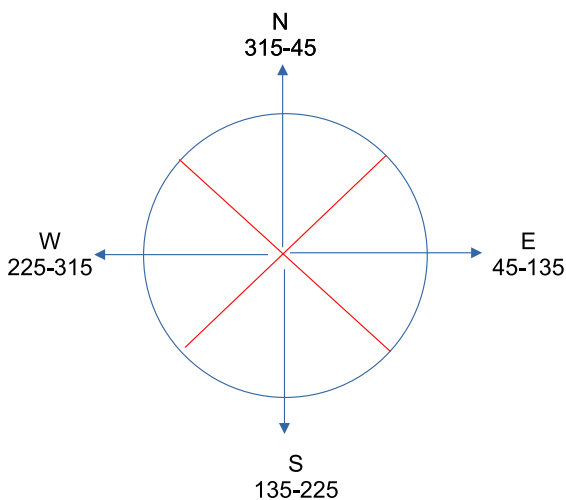
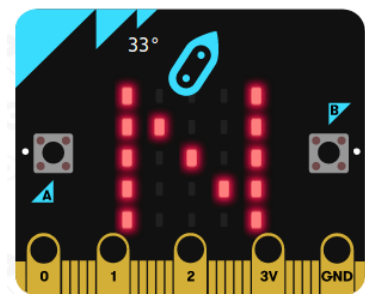
Eingabe

Logik

```

dauerhaft
wenn <Kompassausrichtung (°) > 315 dann
  zeige Text "N"
sonst wenn <Kompassausrichtung (°) > 225 dann
  zeige Text "W"
sonst wenn <Kompassausrichtung (°) > 135 dann
  zeige Text "S"
sonst wenn <Kompassausrichtung (°) > 45 dann
  zeige Text "O"
ansonsten
  zeige Text "N"
  
```

O

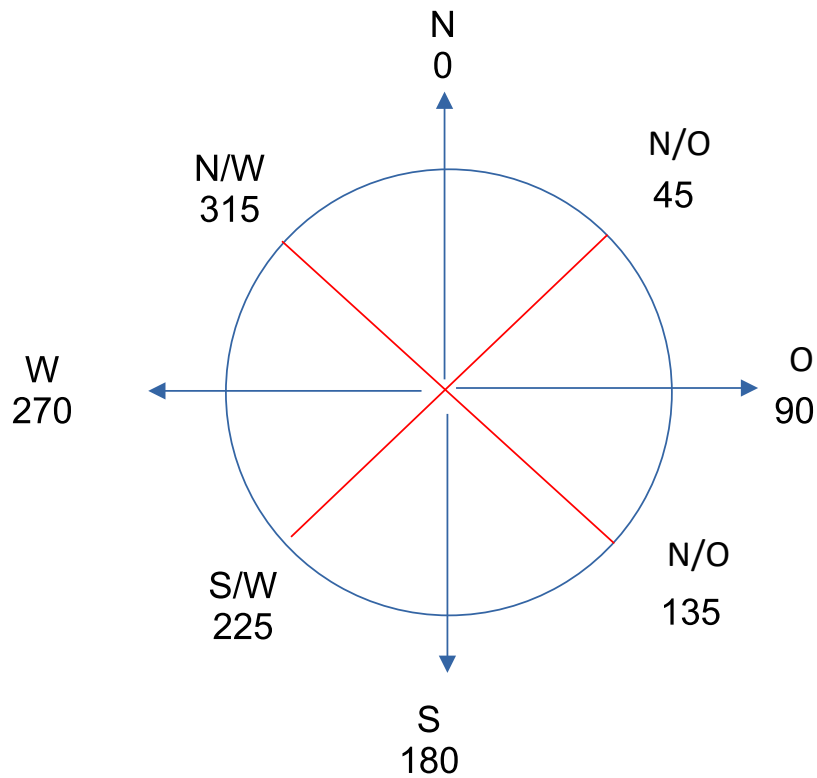


*Erklärung: Immer, wenn ein Wert grösser als eine bestimmte Zahl (z.B. grösser als 45°) soll die nächste Richtung angezeigt werden (z.B. bei 45° Osten). Da die Werte der Reihe nach abgefragt werden, muss das WENN von der grössten (315) zur kleinsten Nummer (45) funktionieren.*

## Zusatzaufgabe

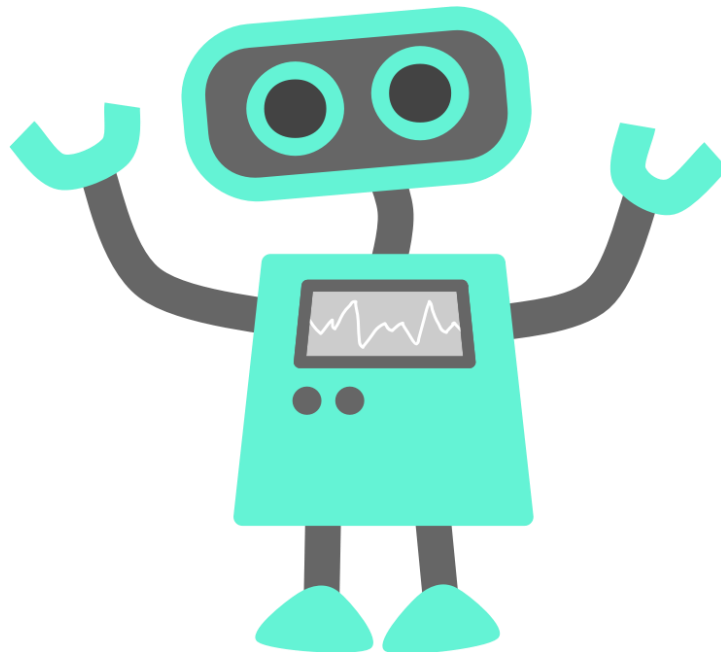
Wie kann der Kompass nun so verändert werden, dass er auch noch die Zwischen-Richtungen (NO, SO, SW, NW) anzeigt?

Hilfe:



## 3

## Mein eigener kleiner Roboter



Hunde sind grossartig. Jedes Mal, wenn wir nach Hause kommen, begrüßen sie uns mit einem unvergleichbaren Enthusiasmus. Egal, wie miserabel der Tag war, das Schwänzeln der treuen Fifis und Bellos dieser Welt hellt jede Stimmung auf. Wir können mit dem micro:bit leider keinen kuscheligen Vierbeiner programmieren, aber einen eigenen kleinen Begrüssungs-Roboter. Er kann zwar nicht hecheln, dafür singen.

### Startfragen

1. Wir brauchen einen Input, der ausgelöst wird, sobald wir ins Haus gehen. Welche Inputs eignen sich dafür?
2. Wie können wir aus dem micro:bit einen Roboter basteln?
3. Welche Outputs können wir nutzen, um den Roboter lebendiger wirken zu lassen?



## Code

.....

VERWENDETE BEFEHLSGRUPPEN

.....

■ Grundlagen

● Eingabe

🔊 Musik

.....

**beim Start**

- schalte eingebauten Lautsprecher **EIN**
  - In der Regel ist dieser Baustein nicht nötig, da der Lautsprecher schon eingeschaltet ist.
- setze Schwellenwert für **laut** auf **100**
  - Definiert, ab welcher Lautstärke der micro:bit etwas als laut einstuft.

**dauerhaft**

- zeige Symbol

**wenn laut**

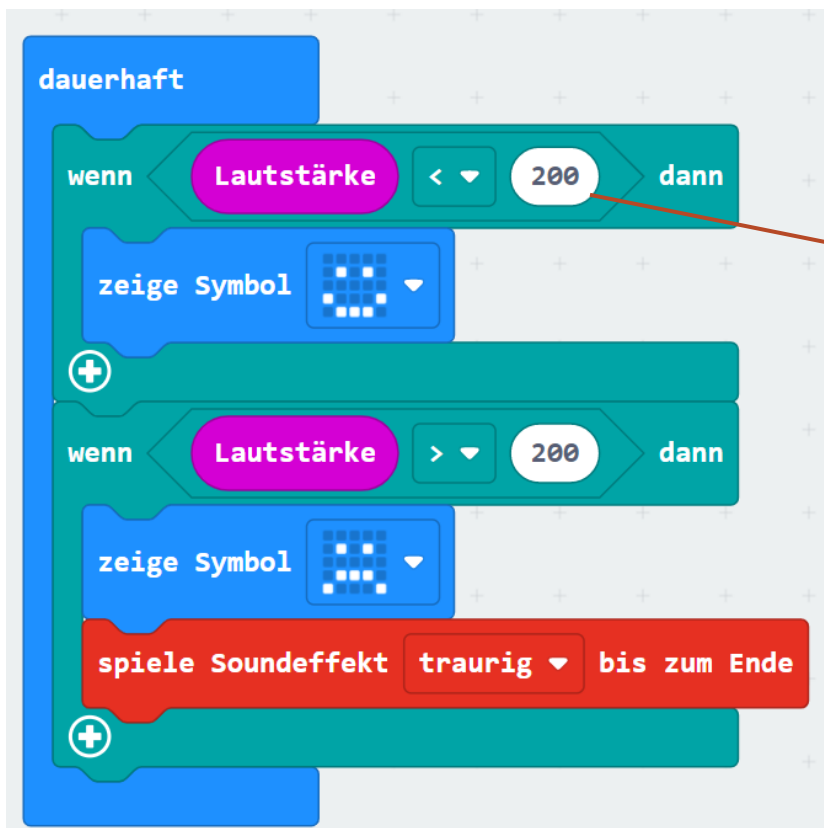
- Beginne Melodie **Entertainer** Wiederhole **einmal**
  - Sobald die als «laut» definierte Lautstärke überschritten wird (z.B. durch das Knallen einer Tür), erklingt die gewählte Melodie.

In diesem Beispiel benutzen wir den micro:bit als Gesicht des Roboters und lassen ihn deshalb ein freundliches Lächeln anzeigen.

## Zusatzaufgaben

1. Verändere den Schwellenwert für «laut» und probiere aus, wie sich das Programm verändert.

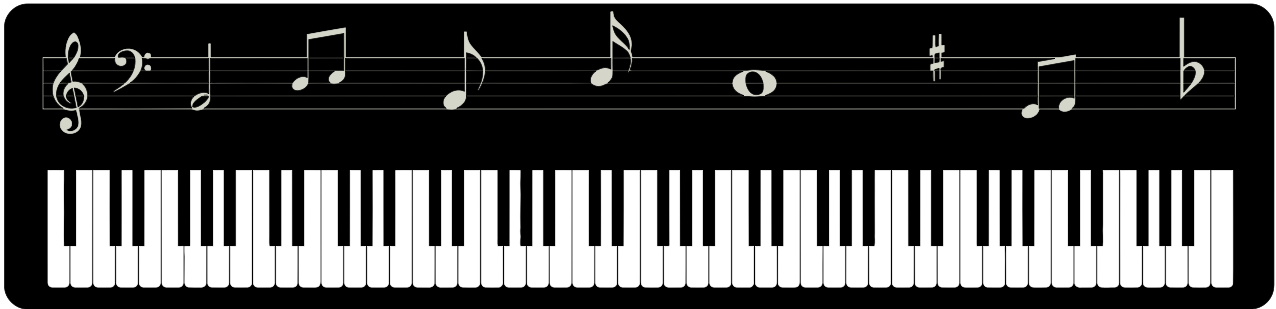
2. Aus dem Begrüssungs-Roboter lassen sich leicht andere praktische kleine Roboter machen. Singender Christbaumschmuck, ein Roboter, der kichert, wenn er angesprochen wird oder ein Roboter, der die Lautstärke im Raum überwacht. Letzteres zum Beispiel mit diesem Programm:



Je nach micro:bit kann der Schwellenwert zu hoch/zu niedrig sein. Individuell anpassen!

## 4

## Tischklavier



Nur schnell eine Melodie spielen. Aber kein Instrument zur Hand. Nun Mal den Kopf nicht hängen lassen. Das neue Motto heisst Do-IT-yourself: Also ganz einfach mit dem Micro:bit und ein bisschen Informatik selber ein simples 3-Ton-Klavier bauen. Na dann los!

**Startfragen**

1. Was benötigen wir für ein 3-Ton-Klavier?
2. Ist es möglich nur mit Karton, Alufolie und dem Microbit ein solches Klavier zu konstruieren?
3. Welche Möglichkeiten gibt es einen Stromkreis mit diesen Materialien zu erstellen bzw. wie wird der Kontakt so gemacht, dass nur dann ein Ton erzeugt wird, wenn auch tatsächlich eine Taste gedrückt wird.

## Code

### VERWENDETE BEFEHLSGRUPPEN

Grundlagen

Logik

Musik

Fortgeschritten

Pins

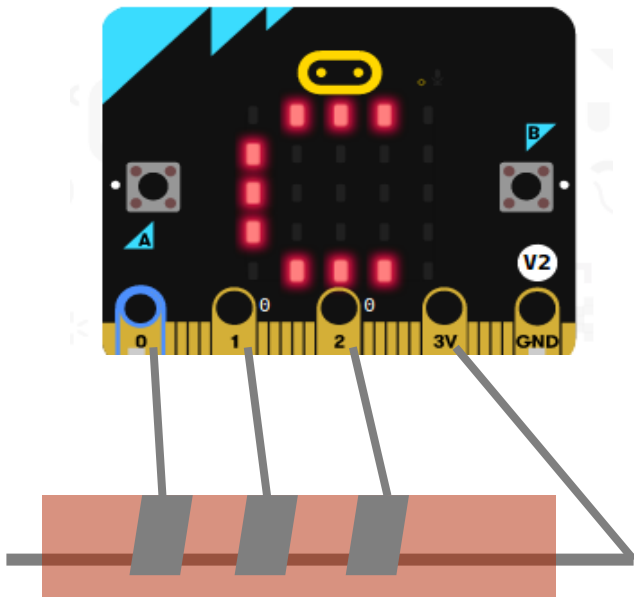
Wenn der Stromkreis zwischen P0 und 3V

Geschlossen/Ein (=1) ist, dann

Spiele Ton ab und zeige Ton (als Text) an

```

dauerhaft
wenn digitale Werte von Pin P0 = 1 dann
  spiele Note Mittleres C für 1 Schlag
  zeige Text "C"
+
wenn digitale Werte von Pin P1 = 1 dann
  spiele Note Mittleres D für 1 Schlag
  zeige Text "D"
+
wenn digitale Werte von Pin P2 = 1 dann
  spiele Note Mittleres E für 1 Schlag
  zeige Text "E"
  
```



**Hinweis**  
Die Alu-Tasten und die Alu-Leitung zum 3V dürfen



## Zusatzaufgabe (Advanced)

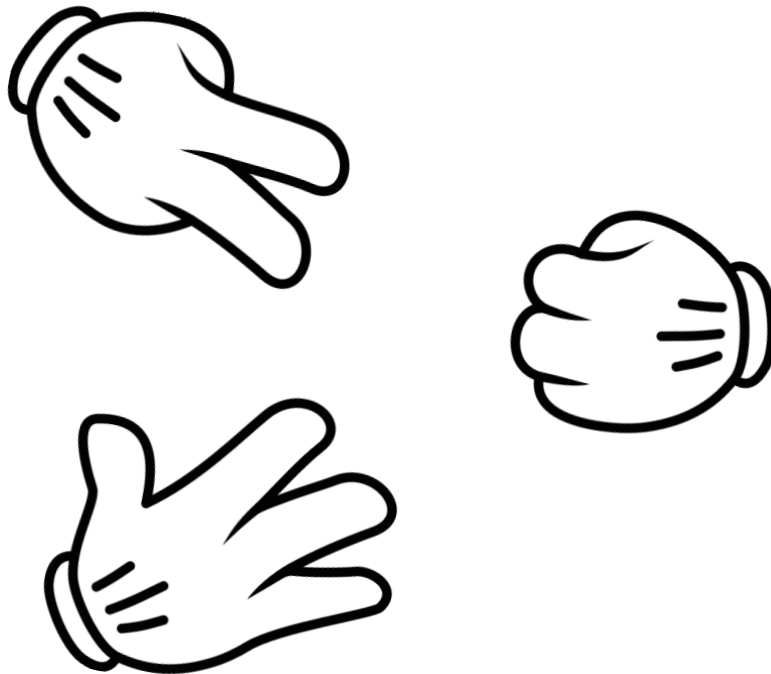
Erweiterung von 3 auf 6 und mehr Töne

- Wir haben nur 3 Pins zur Verfügung (P0, P1, P2)
- Damit jeder Pin zwei verschiedene Töne abspielen kann, nehmen wir Taste A hinzu. Immer wenn Taste A UND ein Pin gleichzeitig gedrückt sind, soll ein anderer Ton entstehen.

Hinweis: Es funktioniert am besten, wenn ein verschachteltes WENN benutzt wird.

## 5

## Papier schlägt Stein. Oder so...



Manche behaupten, es ist das älteste Spiel der Welt. Ob das stimmt, können wir zwar nicht bestätigen. Doch wir alle haben das Spiel mit den drei Handsymbolen schon von klein auf beherrscht und fortan für die wichtigsten Entscheidungen genutzt (wer den letzten Keks essen durfte, wer den Müll rausbringen musste...). Fast eben so früh erfahren haben wir die Ungerechtigkeiten des Spiels, wenn der Gegner beispielsweise sein Handzeichen verzögert machte. Zum Glück sind wir durch moderne Technologien nicht mehr auf so etwas Primitives wie Handzeichen angewiesen. Nein, die Symbole generieren wir einfach mit unserem micro:bit!

### Startfragen

1. Wie können wir mit dem micro:bit das gewählte Zeichen, Schere, Stein oder Papier, sichtbar machen?
2. Wie können wir von einem Zeichen zum nächsten wechseln? Welchen Auftrag haben die Knöpfe A und B?
3. Wie bringen wir dem micro:bit bei, den Gewinner zu ermitteln?

## Code



Dieses Projekt ist etwas anspruchsvoller und länger. Der Code wird deshalb in einzelnen Abschnitten gezeigt. Zudem wird mit Variablen gearbeitet, weshalb zuerst eine kurze Legende folgt.

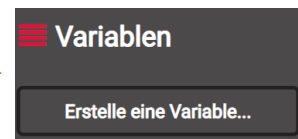
### Schritt 1: Definition der Begriffe


**Knopf A** : Mit Knopf A kann man sein Symbol auswählen. Bei jedem Knopfdruck wird ein Symbol weitergeschaltet, von Schere zu Papier zu Stein.


**Knopf B** : Mit Knopf B bestätigt man seine Eingabe. Danach kann der andere Spieler wählen oder der micro:bit ermittelt den Sieger.

Der micro:bit wird sich merken müssen, welche Symbole die Spieler ausgewählt haben, um anschliessend den Sieger zu ermitteln. Da er aber keine Symbole speichern kann, sind die Anzeigen auf dem LED-Display nur für uns zur Ansicht. Für den micro:bit erstellen wir zusätzlich Variablen.

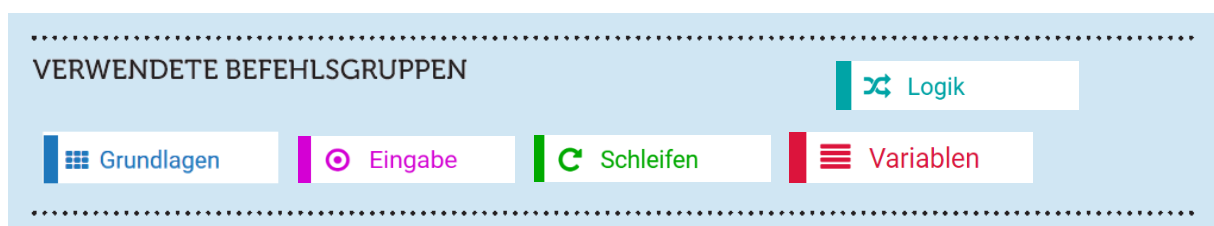
Unter  erstellen wir  zwei neue Variablen: S1 und S2.



 : S1 steht für Spieler 1 und speichert das ausgewählte Symbol des ersten Spielers.

 : S2 steht für Spieler 2 und speichert das ausgewählte Symbol des zweiten Spielers.

### Schritt 2: Symbol auswählen



Zeigt an, welcher Spieler an der Reihe

Solange Knopf B nicht gedrückt wird, kann Spieler 1 seine Eingabe verändern.

Jedes Mal, wenn Knopf A gedrückt wird, wechselt das Symbol. Dabei wird auch die Variable gewechselt.

Schere (Variable Nr.1)

Papier (2)

Stein (0)

Nun haben wir den Code für Spieler 1 erstellt. Genau denselben Code machen wir jetzt nochmals, jedoch mit der Variable S2.

### Schritt 3: Sieger ermitteln

.....  
 VERWENDETE BEFEHLSGRUPPEN  
 .....  
 [Grundlagen] [Logik] [Variablen]  
 .....

Nun müssen wir dem micro:bit beibringen, welches Symbol gewinnt. Kleine Denkaufgabe: Wie viele Kombinationen gibt es, bei denen Spieler 1 gewinnt? Wann gewinnt Spieler 2? Wann gibt es Unentschieden?

Die Tabelle gibt eine Übersicht zu allen möglichen Spielkombinationen.

	S1: Schere	S1: Stein	S1: Papier
S2: Schere	Unentschieden	S1 gewinnt	S2 gewinnt
S2: Stein	S2 gewinnt	Unentschieden	S1 gewinnt
S2: Papier	S1 gewinnt	S2 gewinnt	Unentschieden

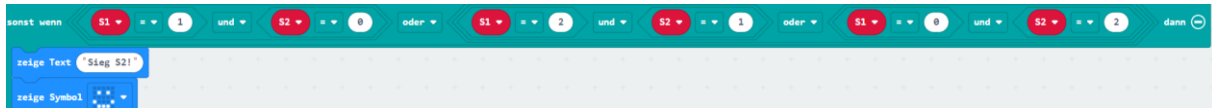
Wir wollen nun eine Funktion erstellen, die ermittelt, wer der Sieger ist. Die Variablen, in denen die Zeichen von Spieler 1 und 2 gespeichert sind, werden abgeglichen. In folgenden Fällen gewinnt Spieler 1:



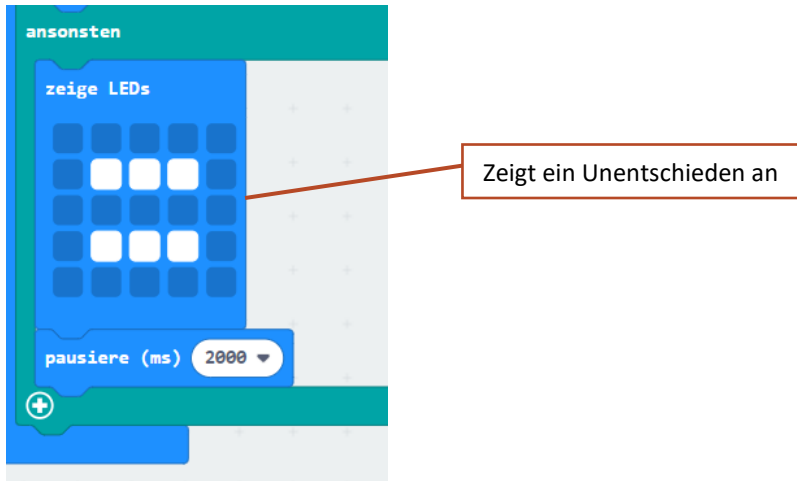
Diese Fälle fügen wir nun alle zusammen.

wenn [S1 = 1 und S2 = 2] oder [S1 = 2 und S2 = 0] oder [S1 = 0 und S2 = 1] dann  
 zeige Text "Sieg S1!"  
 zeige Symbol [Symbol]

Das gleiche wiederholen wir für Spieler 2...



... und nehmen dann für alle möglichen Unentschieden-Fälle die «ansonsten»-Schleufe:



## Zusatzaufgaben

1. Kannst du mit zusätzlichen Variablen die Punkte zählen? So geht ein Spiel beispielsweise auf drei Punkte, bevor der Sieger angezeigt wird.

2. Bei komplexeren Programmen lohnt es sich auch, zu Beginn eine Anleitung einzubauen, die erklärt, wie das Spiel funktioniert, welche Knöpfe gedrückt werden müssen und was die Zeichen bedeuten.

Hinweis: Mit einer Frage ganz am Anfang kann die Anleitung auch optional angezeigt werden.

# Lösungen zu den Zusatzaufgaben

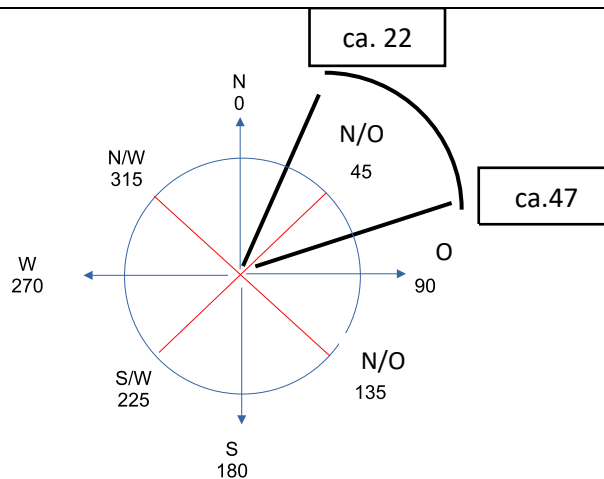
Projekt 1:  
Die Würfel  
sind gefallen

1

wähle eine zufällige Zahl von 1 bis 100

2) Wenn man das LED-Display drückt beim Würfel, wird immer eine 6 gewürfelt.

Projekt 2:  
Wo ist Nor-  
den?



Anhand der Grafik kann abgelesen werden, welche Himmelsrichtung bei welchem Grad-Wert liegt. Der Kompass soll die Richtung aber nicht erst bei diesem Wert ändern, sondern bereits vorher. Nämlich dann, wenn der Wert in die Nähe der neuen Richtung kommt. Dazu kann einfach jeweils die Mitte zwischen den Werten als "Wechsel-Wert" genommen werden. Bsp. Ab ca. 22° soll von N auf N/O gewechselt werden. Ab ca. 47° soll auf Osten gewechselt werden. Der Code dazu könnte folgendermassen aussehen:

```

forever
  if compass heading (°) > 337 then
    show string "N"
  else if compass heading (°) > 292 then
    show string "NW"
  else if compass heading (°) > 227 then
    show string "W"
  else if compass heading (°) > 192 then
    show string "SW"
  else if compass heading (°) > 157 then
    show string "S"
  else if compass heading (°) > 112 then
    show string "SO"
  else if compass heading (°) > 67 then
    show string "O"
  else if compass heading (°) > 22 then
    show string "NO"
  else
    show string "N"
  
```

Projekt 3: Individuelle Lösungen  
Mein eigener kleiner Roboter

Projekt 4: Tischklavier  
Mögliche Lösungsansätze:  
Folgender Code scheint auf den ersten Blick ok, hat aber ein Problem:  
→ Wenn nun P0 und A gedrückt ist, gilt gleichzeitig ja auch, dass P0 gedrückt ist. Somit trifft beides gleichzeitig zu und beide Töne werden abgespielt.



```

wenn digitale Werte von Pin P0 = 1 und Knopf A ist gedrückt dann
  spiele Note Mittleres C für 1 Schlag
  
```

```

wenn digitale Werte von Pin P0 = 1 dann
  spiele Note Mittleres D für 1 Schlag
  
```

Damit es funktioniert, muss der Code verschachtelt werden. Es muss also zuerst geprüft werden, ob gleichzeitig der Pin und die Taste A gedrückt sind. NUR wenn das nicht der Fall ist, soll geprüft werden, ob der Pin alleine gedrückt ist:

```

wenn digitale Werte von Pin P0 = 1 und Knopf A ist gedrückt dann
  spiele Note Mittleres C für 1 Schlag
sonst_wenn digitale Werte von Pin P0 = 1 dann
  spiele Note Mittleres D für 1 Schlag
  
```

Bonusfrage: Was passiert, wenn die Reihenfolge umgekehrt wird (siehe unten) bzw. in welchem Fall wird der Ton "D" zu hören sein?

```

wenn digitale Werte von Pin P0 = 1 dann
  spiele Note Mittleres C für 1 Schlag
sonst wenn digitale Werte von Pin P0 = 1 und Knopf A ist gedrückt dann
  spiele Note Mittleres D für 1 Schlag
  
```

Projekt 5:  
Papier schlägt  
Stein...

1) Mögliche Lösung:

Wenn ermittelt wird, wer die Runde gewinnt, gibt es einen Punkt.

```

ändere Punkte S1 um 1
zeige Text "Punkt S1!"
zeige Symbol
pausiere (ms) 1000
  
```

Bevor die Spieler ein Zeichen auswählen können, wird kontrolliert, ob ein Spieler schon drei Punkte erreicht hat. Wenn nicht, bleibt man in der Spielschleufe...

```

während Punkte S2 < 3 und Punkte S1 < 3
  mache
    zeige Zahl 1
    setze Zeichen Spieler 1 auf 0
  
```

...und sonst wird der Sieger gezeigt:



```

wenn Punkte S2 = 3 dann
  zeige Text "Sieg Spieler"
  zeige Zahl 2
  
```

2) Mögliche Lösung:

Es wird gleich am Anfang gefragt, ob man eine Anleitung möchte. Mit den Knöpfen A und B kann man wählen.

```

dauerhaft
  zeige Text "Hilfe?"
  zeige Text "A=Ja, B=Nein"
  während nicht Knopf A ist gedrückt oder Knopf B ist gedrückt
    mache
      zeige Symbol [Symbol]
  
```

Falls Knopf A gedrückt wird, folgt eine Anleitung, bevor das Spiel beginnt:

```

wenn Knopf A ist gedrückt dann
  zeige Text "Drucke A für ein neues Zeichen."
  zeige Text "Schere:"
  zeige LEDs [LEDs]
  pausiere (ms) 1500
  zeige Text "Stein"
  zeige LEDs [LEDs]
  pausiere (ms) 1500
  zeige Text "Papier"
  zeige LEDs [LEDs]
  pausiere (ms) 1500
  zeige Text "Drucke B zum Bestätigen."
  zeige Text "LOS GEHTS!"
  zeige Symbol [Symbol]
  pausiere (ms) 1500
  
```