

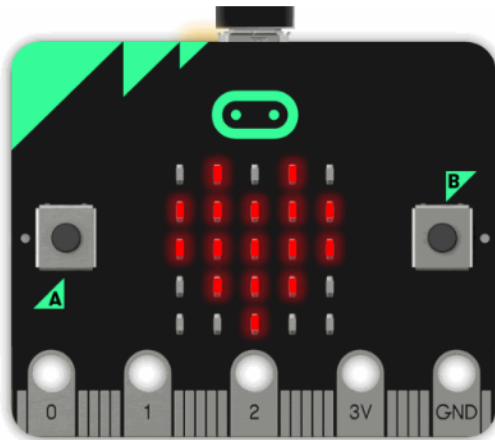


Deine ersten Schritte mit micro:bit - Blockprogrammierung

Der BBC micro:bit ist ein kleiner anprogrammierbarer Computer mit Bewegungssensoren und Kompass. Weitere Informationen und einer ausführlichen Anleitung zu den ersten Schritten gibt es [hier](http://microbit.org/de/guide/).

<http://microbit.org/de/guide/>

Ausstattung



In der Mitte befinden sich 25 LED-Lämpchen in einem 5x5 Raster. Links hast du den A-Knopf (Button) und rechts den B-Knopf. Auf der Rückseite befinden sich die Anschlüsse für das Batteriefach und ein Micro-USB-Kabel.

Verbinde den micro:bit

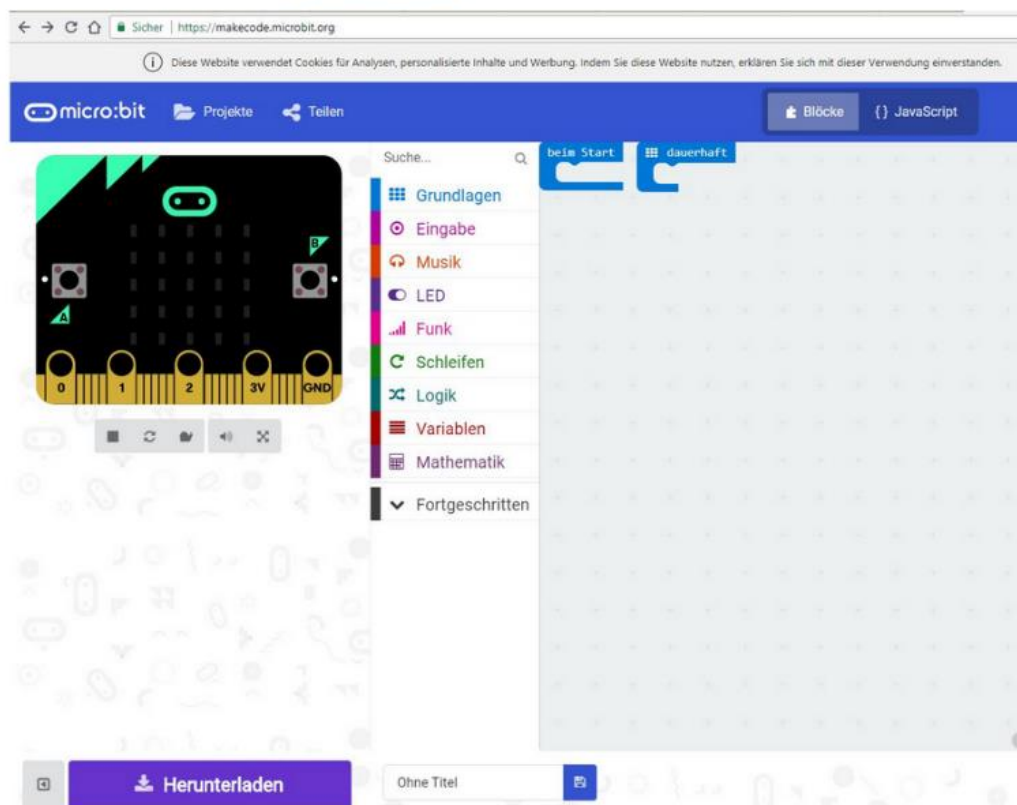
Um zu Beginn verbinde deinen micro:bit mit dem zusätzlichen Batteriefach oder mit einem USB-Kabel mit deinem Laptop.

Drücke die Knöpfe (Buttons) wenn dich dein micro:bit mit Lichtzeichen (A und B) dazu auffordert.

Entwicklungsumgebung

<https://makecode.microbit.org/#lang=de>

Öffne in einem Webbrowser die folgende URL: <https://makecode.microbit.org/>



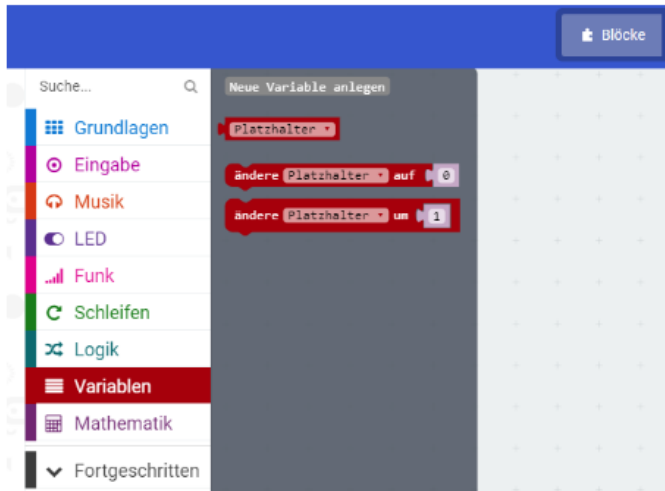
Das ist der Editor mit dem wir ähnlich wie bei Scratch mit den bunten Blöcken oder mit JavaScript programmieren können.



Erstes Programm - Würfel

Dein erstes Programm soll einen zufälligen Würfelwurf nachstellen. Wenn der micro:bit geschüttelt wird soll er danach eine Augenzahl von 1-6 anzeigen.

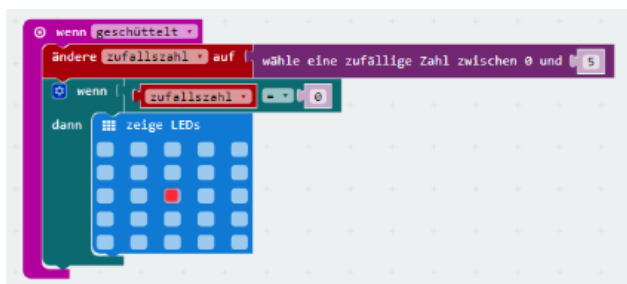
1. Lege eine Variable für die Zufallszahl an.



2. Bei Eingabe können wir auf das Schütteln reagieren und uns eine Zufallszahl speichern. Unter Mathematik findest du den Block für eine Zufallszahl. Zufallszahlen starten immer bei 0, daher benötigen wir eine Zufallszahl von 0-5. Ist die Zufallszahl eine 0, dann zeigt der Würfel eine 1. Ist die Zahl eine 5, zeigt der Würfel eine 6.



1. Für die Entscheidung welche Augenzahl angezeigt werden soll verwendest du den Wenn-Dann-Block und den Vergleichsoperator = unter Logik. Mit dem Zeige-LEDs-Block unter Grundlagen kannst du einzelne LEDs zum Leucht bringen und somit ein Muster für die Augenzahlen entwerfen.



1. Probiere die weiteren Augenzahlen selbstständig zu lösen.

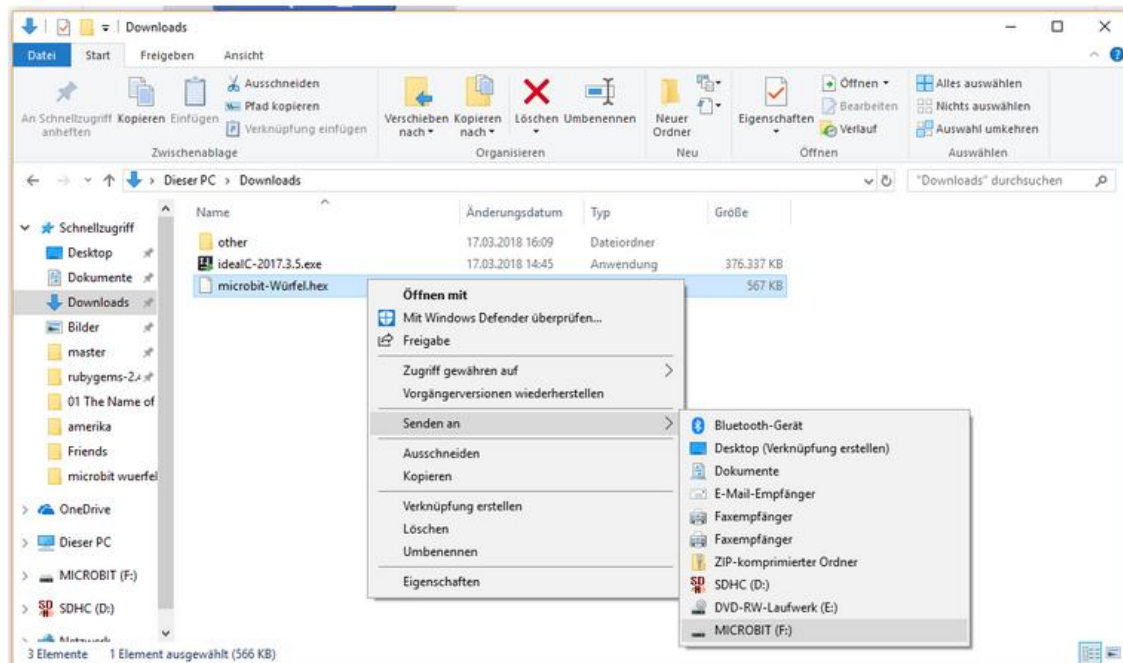
Übertragung des Programms auf den micro:bit

Um das ganze jetzt auszuprobieren muss das Programm auf den micro:bit übertragen werden. Dazu muss der micro:bit über ein USB-Kabel mit dem Laptop verbunden sein.

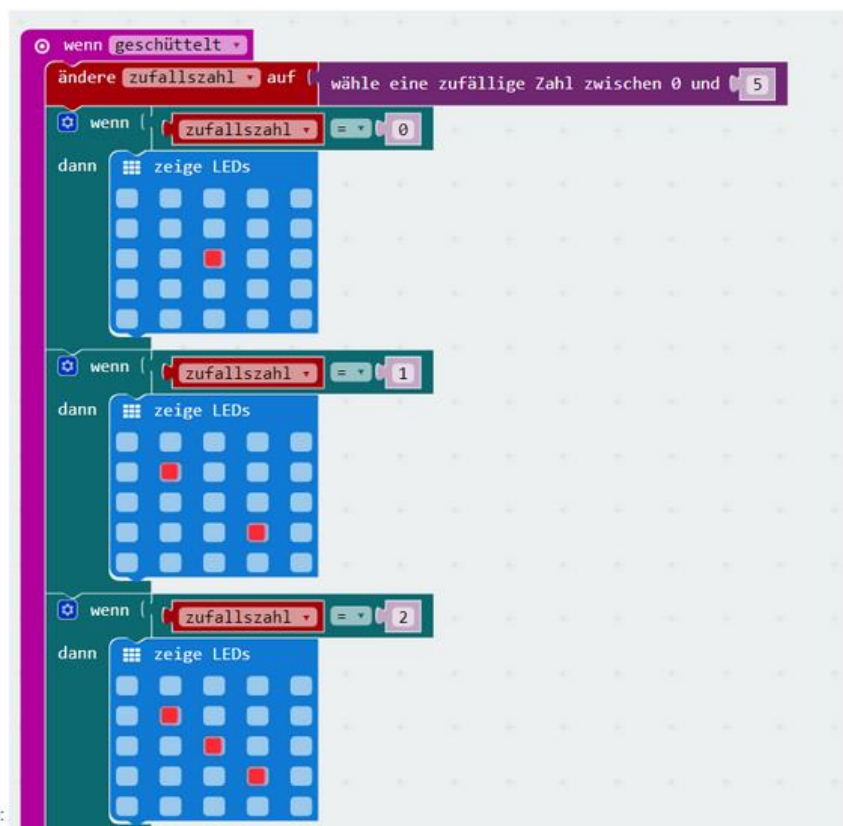
1. Lade das Programm vom Webbrowser herunter.
2. Öffne den Ordner in den du das Programm heruntergeladen hast.



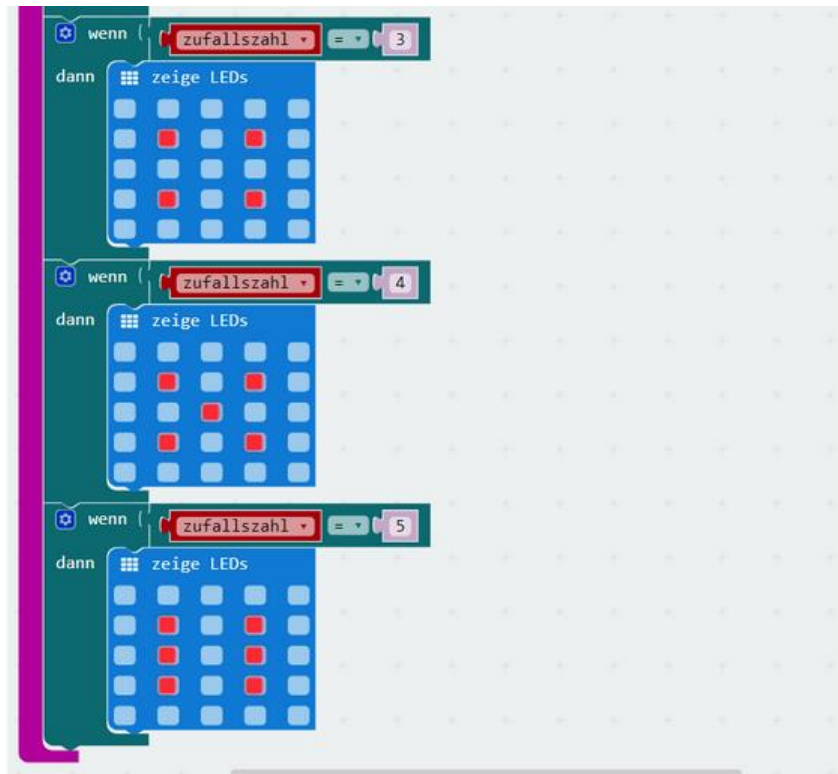
3. Übertrage das Programm auf den Microbit. Klicke mit der rechten Maustaste auf die Datei um das Context Menu zu öffnen. (Oder kopiere in von den Ordner auf den Micro:bit).



Fertiges Programm



Das fertige Programm könnten so aussehen:



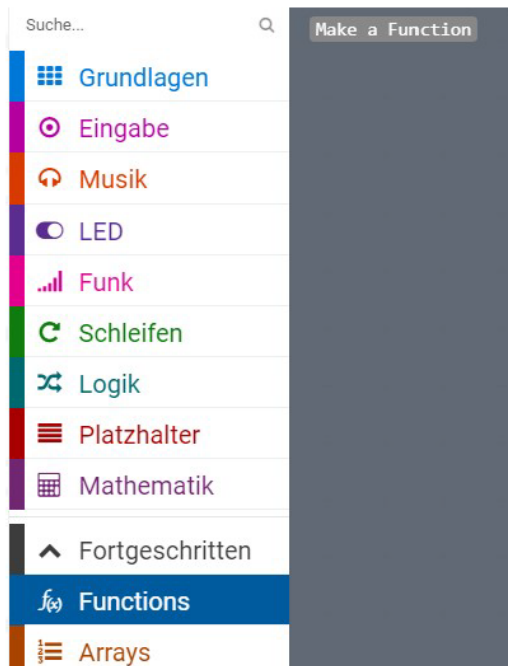


Zweites Programm: Würfel für Fortgeschrittene

Funktionen für verschiedene Würfelzustände

Für jeden Würfelzustand benötigst du eine Funktion. Mit Würfelzustand ist das gemeint, was nach jedem Wurf angezeigt wird.

Um eine Funktion zu erzeugen wähle im Menü *Fortgeschritten* zuerst *Functions* und schließlich *Make a Function* aus. Benenne die Funktion mit *zeigeEins*.

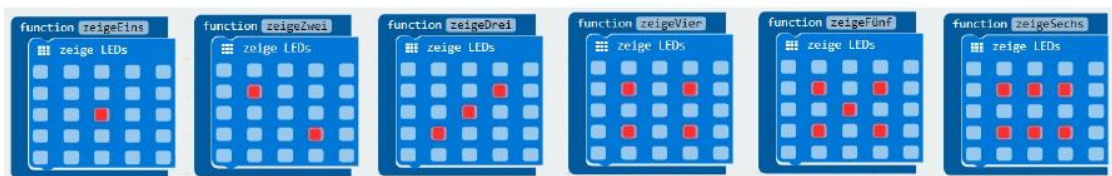


New function name:

Die Funktion `function zeigeEins` soll die Zahl *eins* am Display deines micro:bit anzeigen. Dazu wähle in *Grundlagen* den Baustein `zeige LEDs` aus und verschiebe diesen in die `function zeigeEins`. In dem Baustein markierst du dann die LED in der Mitte welche aufleuchten soll wenn die Eins angezeigt wird.



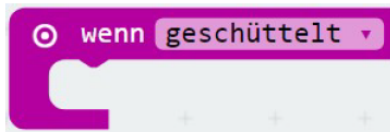
Lege nun auch für alle anderen Würfelzustände, also zwei bis sechs, jeweils eine Funktion nach obigen Muster an.



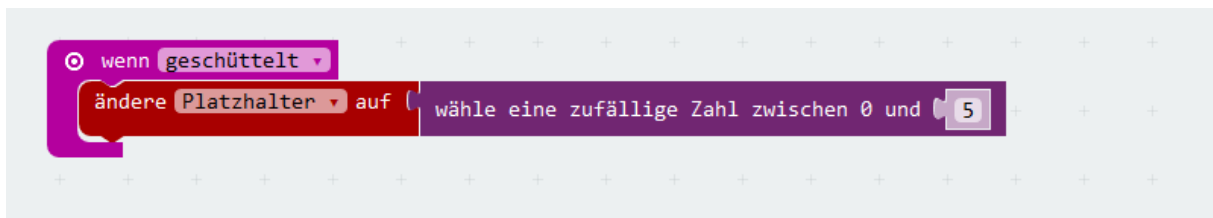


Würfeln

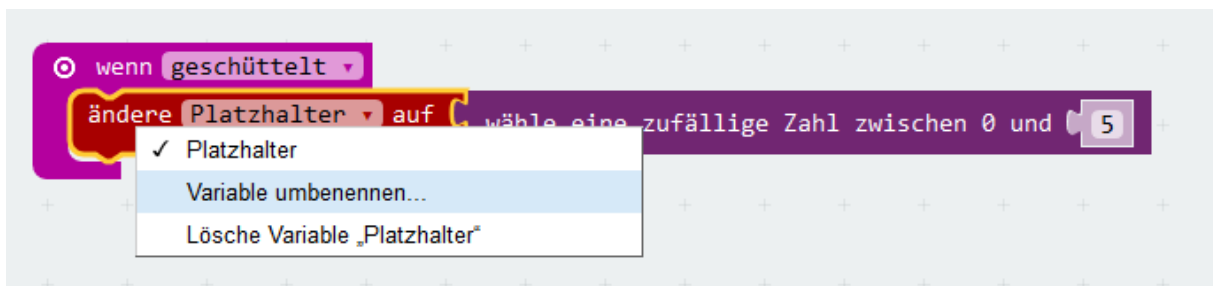
Der micro:bit kann mit seinem Beschleunigungssensor auf schütteln reagieren. Um dieses zu ermöglichen wähle von *Eingabe* den Block `wenn geschüttelt` aus. Dieser Block wird immer dann aufgerufen, wenn der micro:bit geschüttelt wird.



Als nächstes muss du eine zufällige Zahl erzeugen. Dazu kannst du die Funktion `pick random` verwenden. Um den Baustein verwenden zu können, benötigst du noch eine Variable, in welcher die Zufallszahl gespeichert wird. Um die Variable zu erzeugen gehe zu *Variablen* und verschiebe der Baustein `ändere Platzhalter` auf `0` ins Baustein `wenn geschüttelt`. Dann verwendest du aus *Mathematik* den bereits erwähnten `pick random` Baustein.



Die Variable `Platzhalter` hat jetzt allerdings einen nicht sehr sprechenden Namen. Entwickler vergeben gerne sprechende Namen um die Programme besser lesen zu können. Benenne also die Variable noch um. Das kannst du machen in dem du auf das kleine Dreieck neben der Variablen `Platzhalter` klickst und *Rename variable...* wählst. So kannst du die Variable zum Beispiel `GewürfelteZahl` nennen.

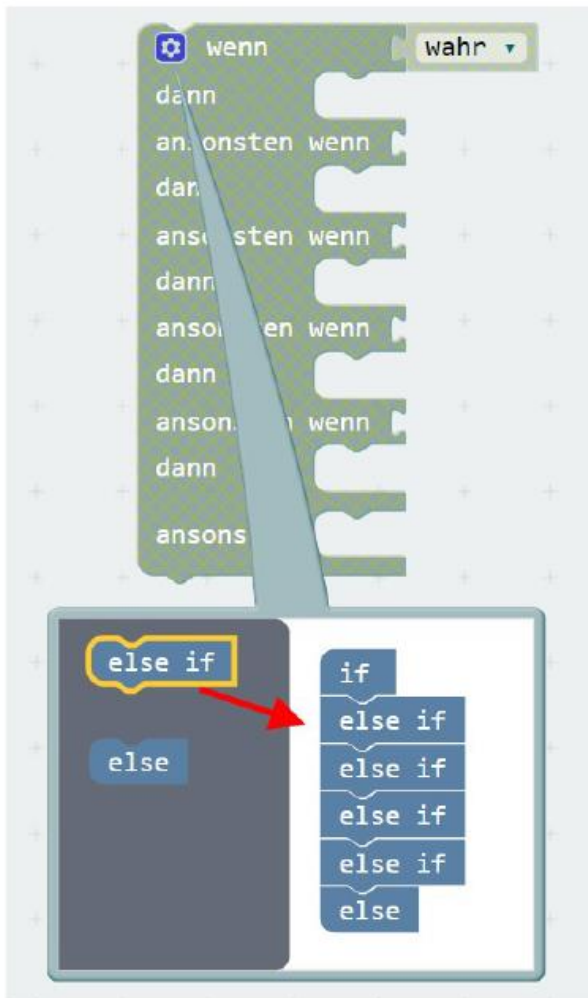


Zeige gewürfelte Zahl an

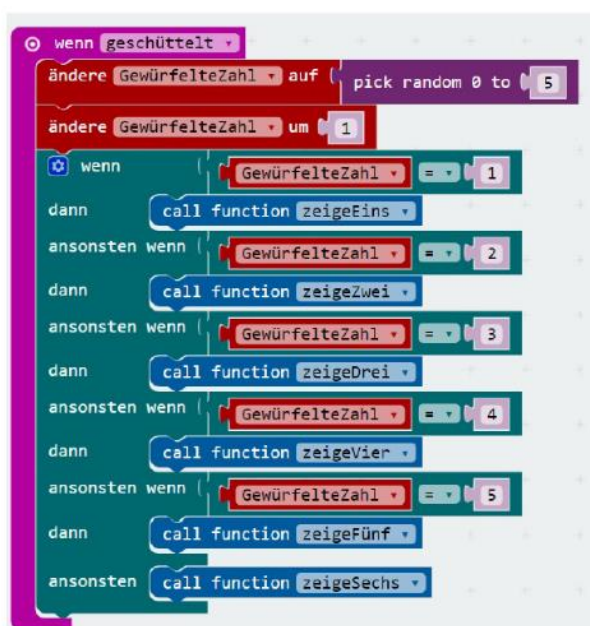
Das wird jetzt einfach. Um das gewürfelte Zahl anzuzeigen gehst du in *Logik* und nimmst du der Baustein `_wenn_dann_ansonsten`. Mit diesen Baustein hast du die Möglichkeit folgende Logik aufzubauen: Wenn das `GewürfelteZahl` eins ist, dann rufst du die Funktion `zeigeEins`, für zwei dann die Funktion `zeigeZwei` usw. Jedoch es zu ermöglichen muss du das Baustein `_wenn_dann_ansonsten` ein bisschen modifizieren.



Hol dir das Baustein `_wenn_dann_ansonsten` und klicke auf das kleine rädchen um diese Baustein zu verändern.



Jetzt wird das `GewürfelteZahl` mit einem erwarteten Zahl verglichen. Dazu holt man sich von *Logik* das `ist gleich` Baustein. Auf die linke Seite setzt man die Variable `GewürfelteZahl` ein auf die rechte das erwartete Wert. Diese setze man in den `wenn_dann_ansonsten` Baustein ein. Von `dann` ruft man dann die jeweilige Funktion mit `call function` von Menu *Functions*.



Wenn du jetzt auf das `Shake` button in `micro:bit Simulator` klickst, solltest du jedesmal ein Würfergebnis sehen.

Aufgabe

Um es ein bisschen spannender zu machen, versuche das Programm so erweitern dass man während würfeln verschiedene Zahlen sehen kann. Genau als wenn eine Würfel auf dem Tisch rollt.

