

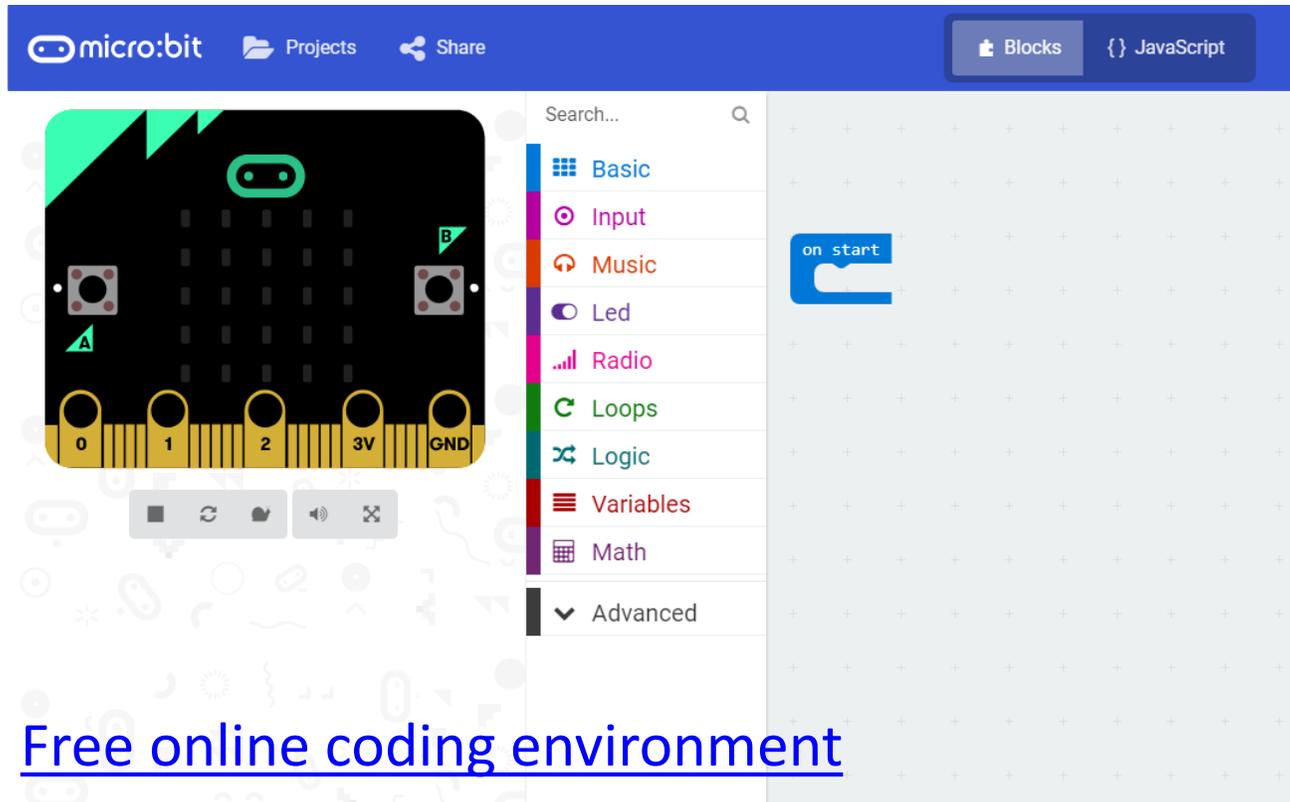
# BBC micro:bit challenges

for implementing  
Digital Technologies in primary years

---

# About the BBC:microbit

The micro:bit is a handheld, small, fully programmable computer that you can use to create simple games, sense the surrounding environment, make music and much more. Create a program using the coding environment and download this to your physical micro:bit to see how it performs.



[Free online coding environment](#)

Physical BBC micro:bit  
*(available for purchase)*



[Using your micro:bit in five easy steps](#)

[Features of the micro:bit!](#)

# Projects

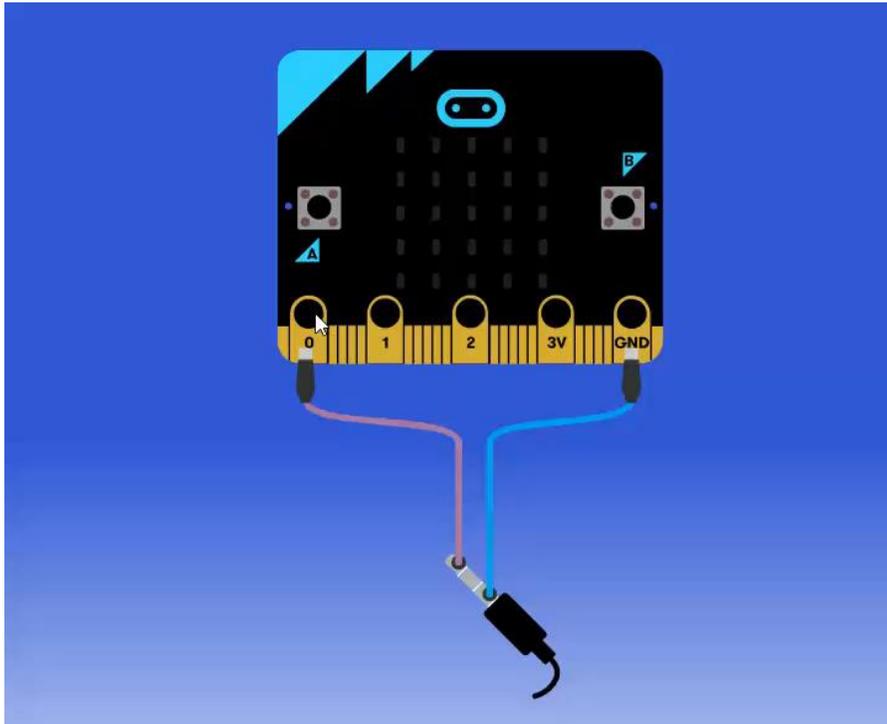
- **Years 3-4: [Making a Morse code signal sender](#)**
- **Years 3-4: [Making a compass](#)**
- **Years 5-6: [Make a simple light meter](#)**
- **Years 5-6: [Making a digital dice](#)**
- **Years 5-6: [Star wars music](#)**

# Making a Morse code signal sender

Morse code is a way of representing letters of the alphabet. It is made up of dots and dashes.

A dot is one unit; a dash is 3 units.

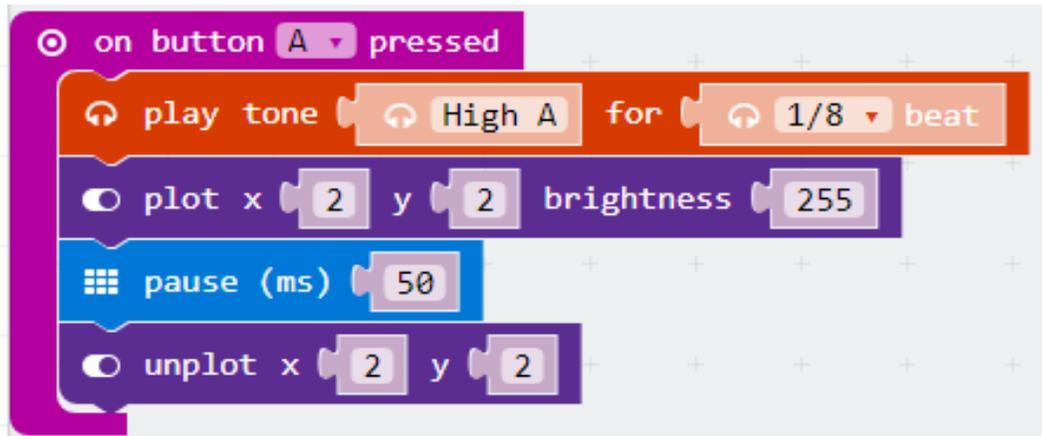
Work out what this one word message says.



A	● —	O	— — —
B	— ● ● ●	P	● — — ●
C	— ● — ●	Q	— — ● —
D	— ● ●	R	● — ●
E	●	S	● ● ●
F	● ● — ●	T	—
G	— — ●	U	● ● —
H	● ● ● ●	V	● ● ● —
I	● ●	W	● — —
J	● — — —	X	— ● ● —
K	— ● —	Y	— ● — —
L	● — ● ●	Z	— — ● ●
M	— —		

# Making a Morse code signal sender

So this code represents a dot in Morse code.



```
on button A pressed
  play tone High A for 1/8 beat
  plot x 2 y 2 brightness 255
  pause (ms) 50
  unplot x 2 y 2
```

The image shows a Scratch script starting with an 'on button A pressed' event. It contains four blocks: 'play tone' set to 'High A' for '1/8 beat', 'plot x 2 y 2 brightness 255', 'pause (ms) 50', and 'unplot x 2 y 2'.

Can you write a sequence to represent a dash?  
Make its start on button B pressed.



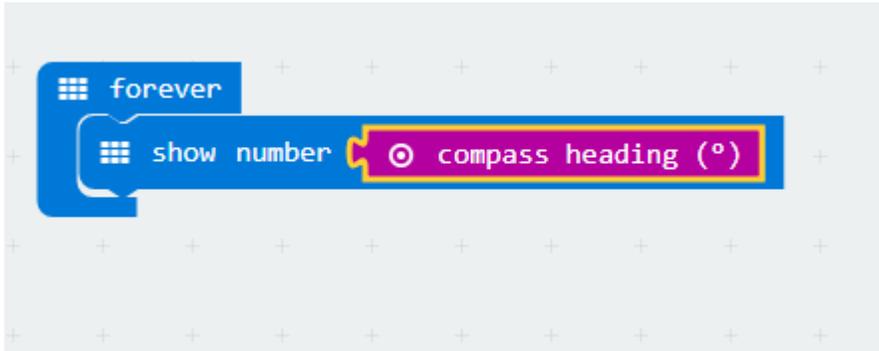
```
on button B pressed
```

The image shows a Scratch script starting with an 'on button B pressed' event. The rest of the script is currently empty.

Ans, Here's one way to write the [code](#).

# Making a compass

Create this simple code.



```
forever loop
  show number: compass heading (°)
```

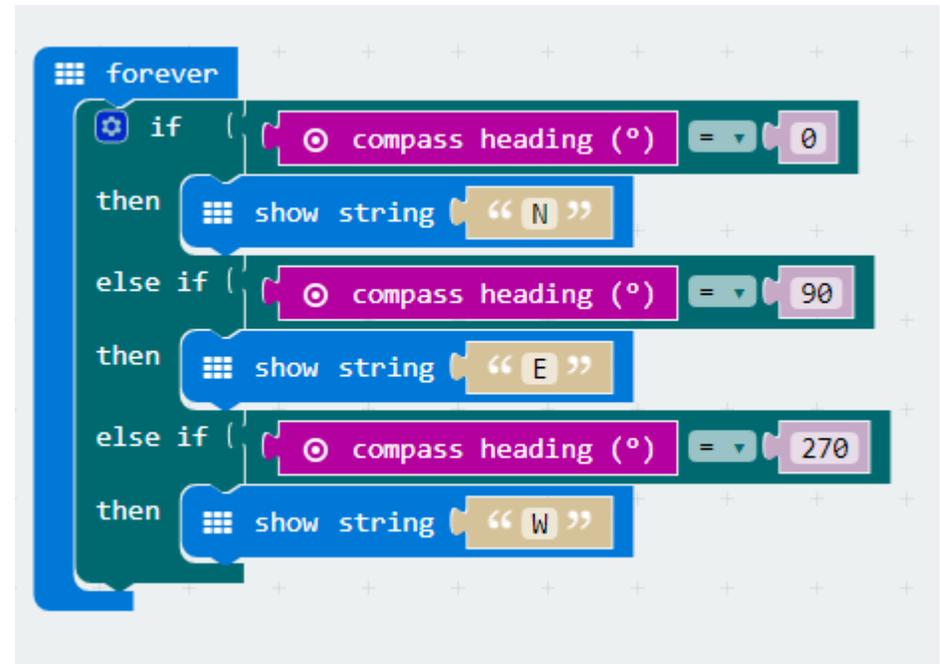
Download to your micro:bit.

You'll then need to calibrate the micro:bit so it knows where North is.

All you need to do is move the dots on screen to make a circle.

Watch this [tutorial](#) for assistance.

You can then use if/then statements to indicate N, S, E and W.



```
forever loop
  if (compass heading (°) = 0)
    then show string: " N "
  else if (compass heading (°) = 90)
    then show string: " E "
  else if (compass heading (°) = 270)
    then show string: " W "
```

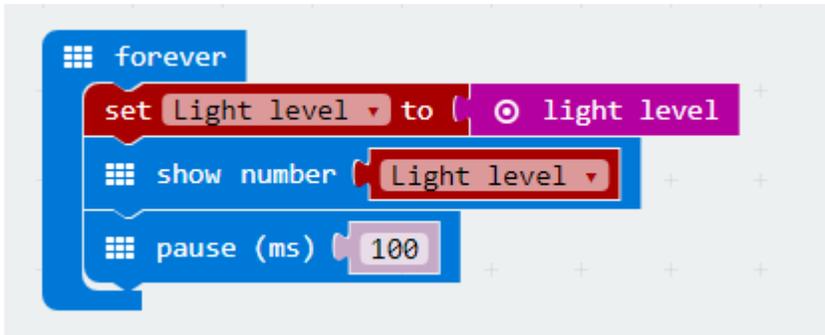
Ans, Here's one way to write the [code](#).

See if you can add South.

# Make a simple light meter

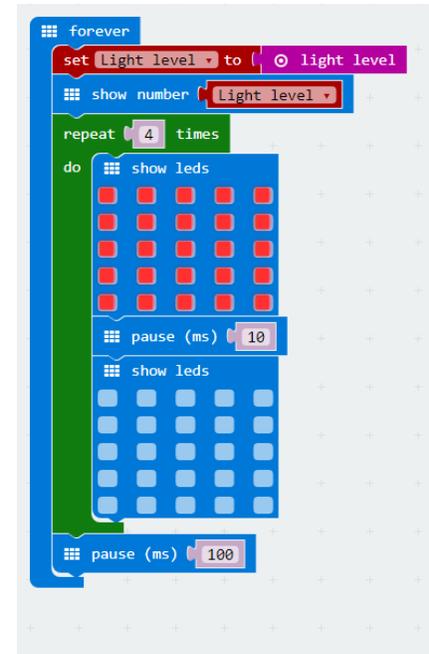
The micro: bit has a very basic light sensor.  
The grid can be used to measure light.

Here's some code to measure the light level.



```
forever loop
  set Light level to (light level)
  show number (Light level)
  pause (ms) (100)
```

The light value scrolls over the LED display.  
I added a flashing display so it is easier to read the value.

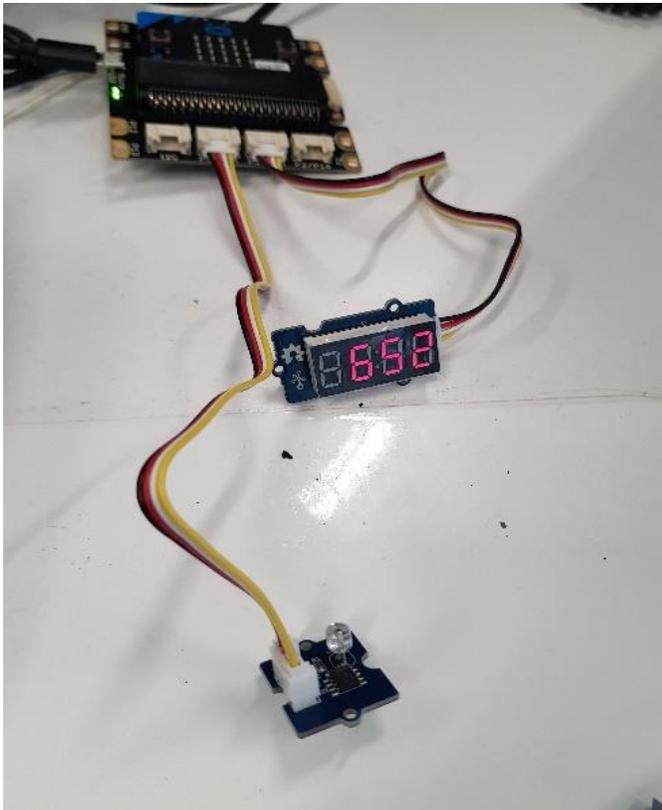


```
forever loop
  set Light level to (light level)
  show number (Light level)
  repeat (4) times
    do
      show leds (LED display with 4 red LEDs lit)
      pause (ms) (10)
      show leds (LED display with 4 blue LEDs lit)
  pause (ms) (100)
```

Here's one way to write the [code](#).

# Make a light meter using Grove kit

The grove kit enables you to connect a light sensor and digital display for a more accurate reading.



This code runs if you have added the Grove Kit add on.

```
on start
  set Display to 4-Digit Display at P1 and P15

on start
  set Light level to analog read pin P0

on button B pressed
  plot bar graph of analog read pin P0
  up to 255
  pause (ms) 100

forever
  Display show number analog read pin P0
```

Here's one way to write the [code](#).

# Make a light meter using Grove kit

Collect real data



Hold over any material and gather data on how much light travels through the material.

Aligns with Science Inquiry Skills and Science understanding about light.

# Making a digital dice

A standard game dice has six sides each representing a number one to six.

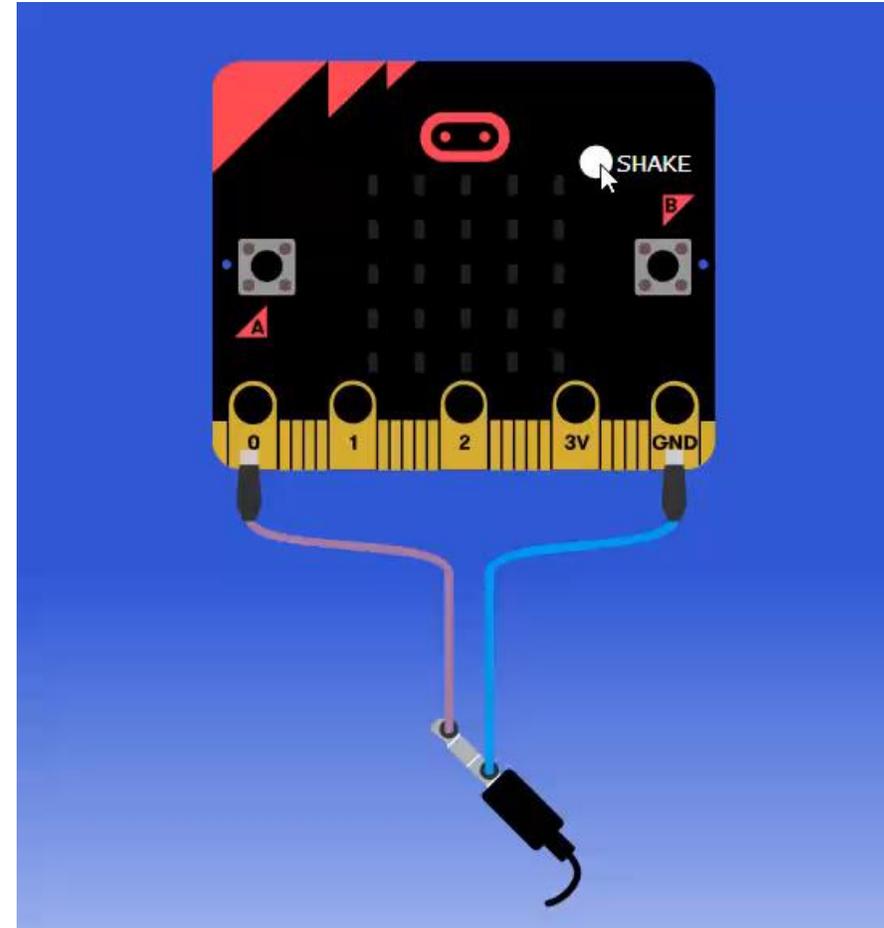
Here's a clip showing one way to make the BBC micro:bit into a digital dice.

It starts on 'shake'. Play the clip to see how it works.

Task 1:

Try and work out the steps to program your own digital dice.

Draw a flow chart or write instructions.



# Making a digital dice

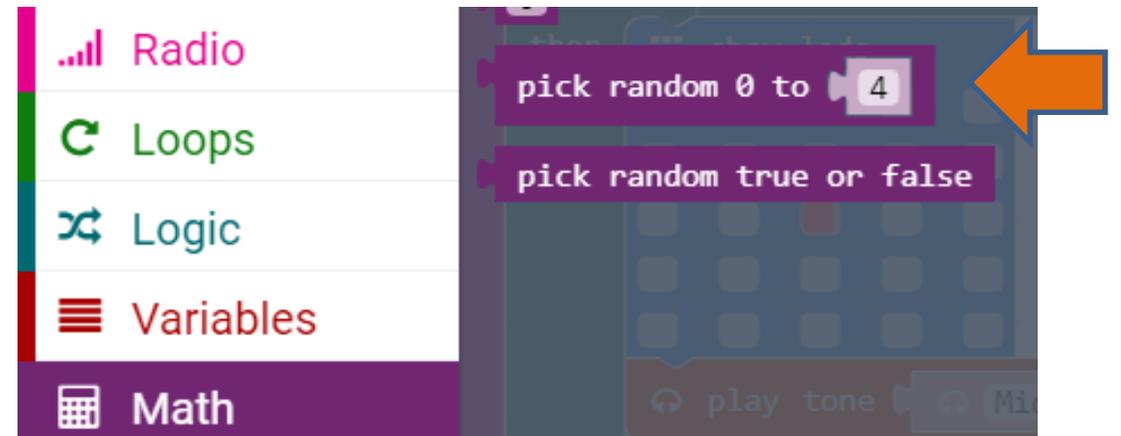
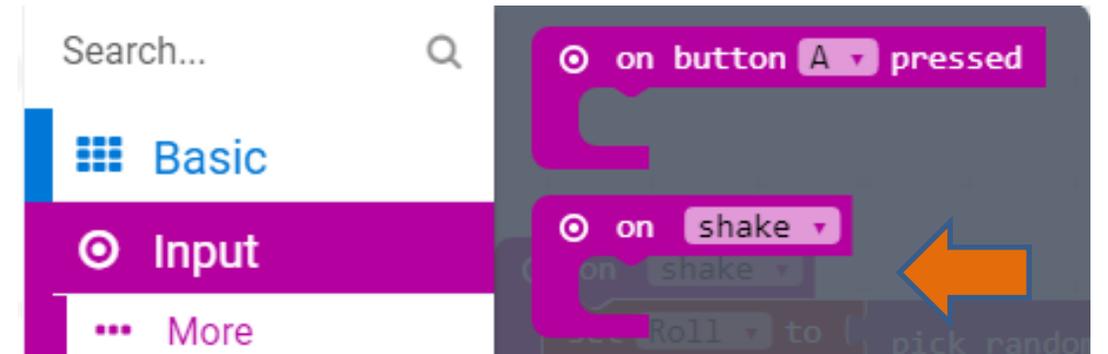
How about programming the micro:bit?

First thing is to select a way to start the program.  
On shake is one possible way.

To make a digital game dice we need the numbers to come up in any order which we call random.

In the Math blocks we can use the purple block and adjust the value 4\*.

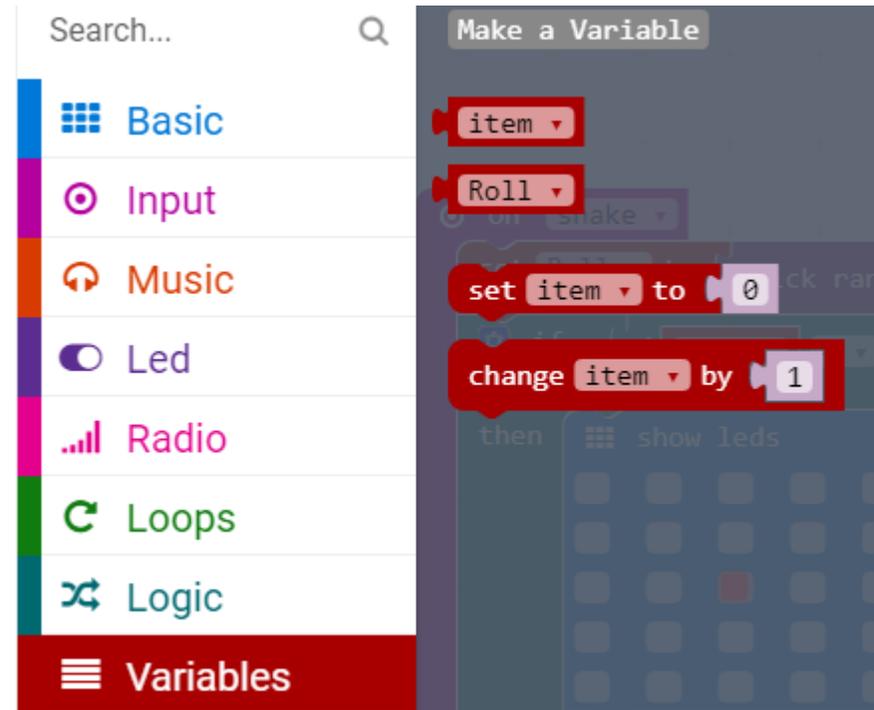
*\* Note when using the random block, the values start at 0 however we want to start at 1 so we have to add 1 to each random number eg 0 then equal 1 and the random value of 5 would equal 6.*



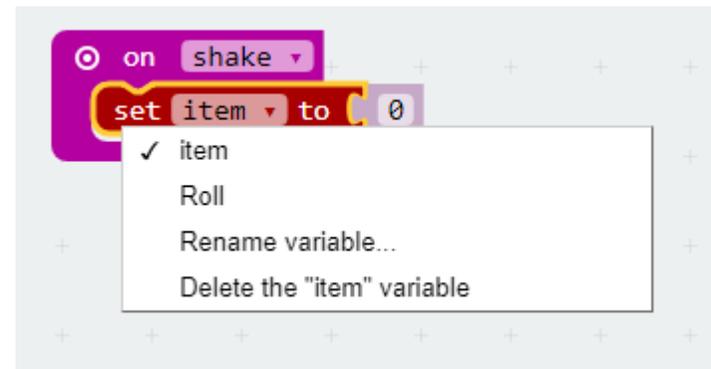
# Making a digital dice

We would also need to create a variable such as 'roll' so that a 'roll' can be any number 1 to 6.

Select make your own variable.  
I called the variable 'Roll'.



Then add the block set item to 0.  
Change 'item' to 'roll' .



# Making a digital dice

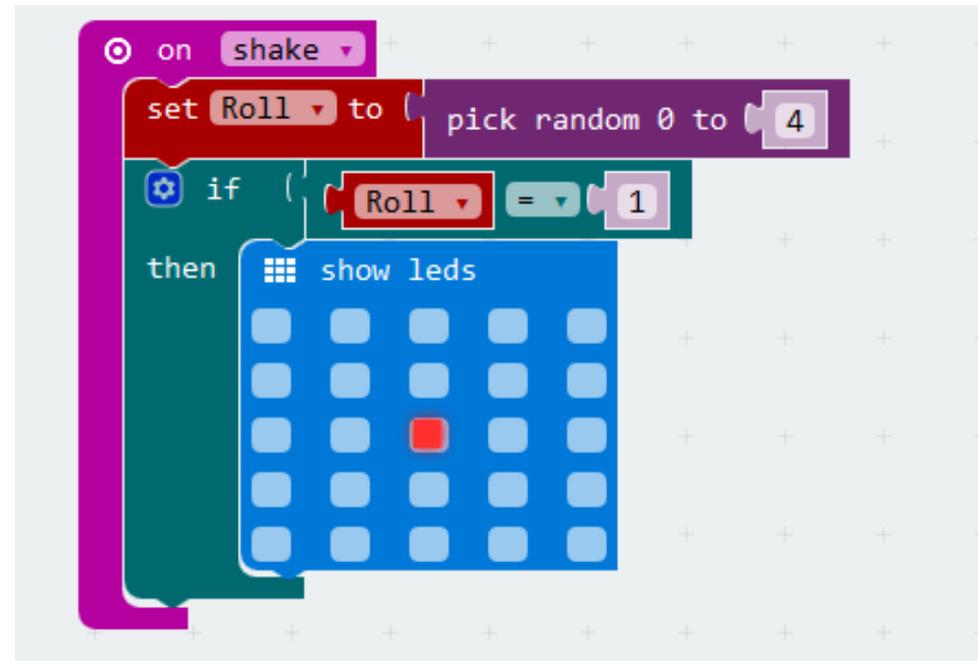
Here's a start to play around with a debug.

I've added an if/then logic block.

So my thinking is If the Roll = 1 then I want the display to show 1 like a dice.

See if you can add the same logic for the numbers 2 to 5.

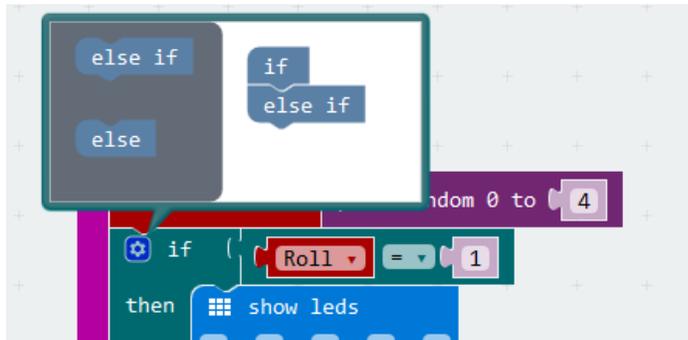
[Code to start dice roll](#)



# Making a digital dice

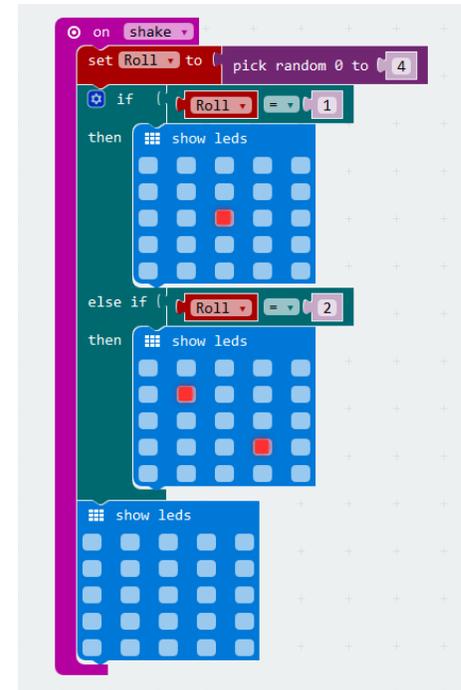
Here we are adding the logic:

- if the Roll = 1 is true then show the LED display with 1 dot
- else if the Roll = 2 is true then show the LED display with 2 dots
- TIP: To get the if then, else if block click on the blue star in the if block and select else if and place under if.



Here's the next step with two numbers.

[https://makecode.microbit.org/\\_e1LczAUVKH41](https://makecode.microbit.org/_e1LczAUVKH41)



Here's an example completed [code](#)

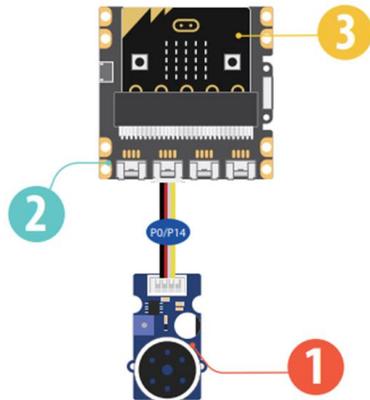
# Star wars music

You can create you own music.

All you need are the notes and beat for some music and then explore using your micro:bit. The online coding environment lest you run the code and hear the sounds.

Start with a simple melody.

I've set the program up if you have a Grove kit with speaker you can play the theme. Attach the speaker to the pin PO.



```
on shake
  repeat 4 times
    do
      digital write pin P0 to 1
      play tone Middle G for 1/4 beat
      rest(ms) 1/8 beat
      play tone Middle G for 1/4 beat
      rest(ms) 1/8 beat
      play tone Middle G for 1/4 beat
      rest(ms) 1/8 beat
      play tone Middle C for 1 beat
      play tone Middle C for 1 beat
      play tone Middle G for 1 beat
      play tone Middle G for 1 beat
      play tone Middle F for 1/4 beat
      rest(ms) 1/8 beat
  run in background
    show string "Star wars"
```

Here's an example completed [code](#) if using Grove with speaker on Pin PO.

Here's [code](#) for star wars for simulator.