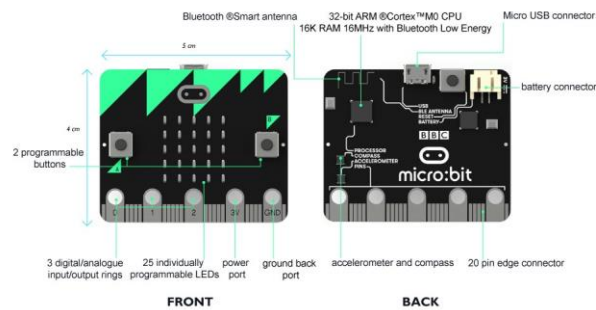




# **AiTLE + micro:bit Alliance + HKedCity + HKUSPACE: micro:bit Workshop Series**

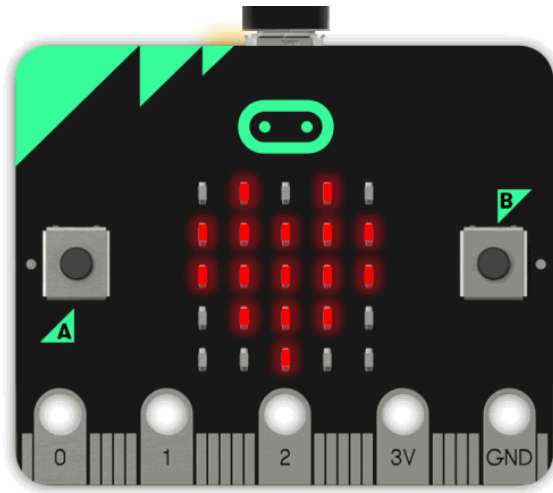
## **Level 2: Built-in micro:bit I/O**



### 1 - Overview

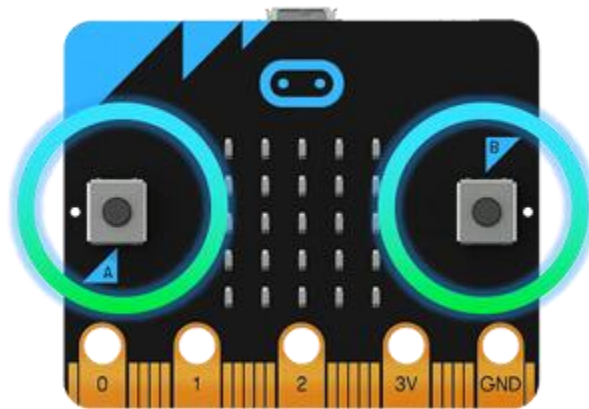
Your micro:bit has the following physical features:

- 25 individually-programmable LEDs
- 2 programmable buttons
- Physical connection pins
  - Light and temperature sensors
  - Motion sensors (accelerometer and compass)
- Wireless Communication, via Radio and Bluetooth
  - USB interface



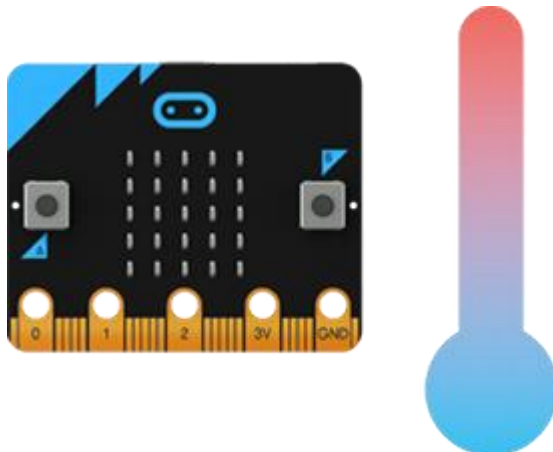
2 - LEDs

- LED stands for Light Emitting Diode. The micro:bit has 25 individually-programmable LEDs, allowing you to display text, numbers, and images.



3 - Buttons

- There are two buttons on the front of the micro:bit (labelled A and B). You can detect when these buttons are pressed, allowing you to trigger code on the device.



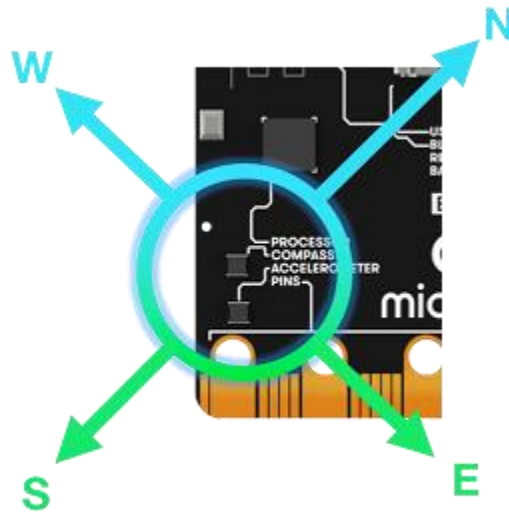
#### 4 - Temperature Sensor

- This sensor allows the micro:bit to detect the current ambient temperature, in degrees Celsius.



#### 5 - Light Sensor

- By reversing the LEDs of the screen to become an input, the LED screen works as a basic light sensor, allowing you to detect ambient light.



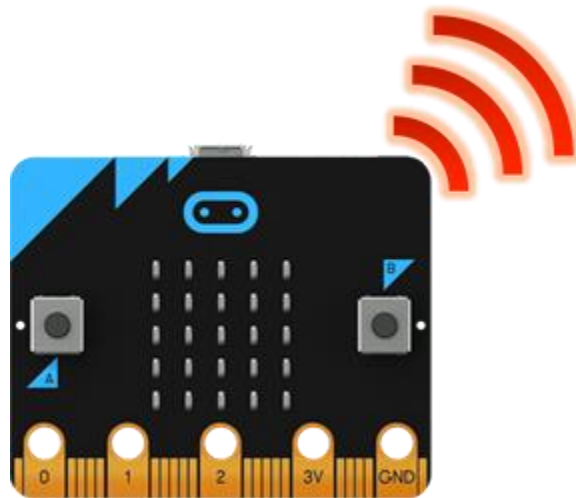
#### 6 - Compass

- The compass detects the earth's magnetic field, allowing you to detect which direction the micro:bit is facing. The compass has to be calibrated before it can be used.
- 'Calibrating' the compass ensures the compass results are accurate. For the JavaScript Blocks Editor, use the ['calibrate compass'](#) block. To calibrate the compass in Python use [compass.calibrate\(\)](#).
- When the calibration begins, the micro:bit will scroll an instruction on the display for you - either "Draw a circle" or "Tilt to fill screen". To calibrate the compass, just follow these instructions and tilt the micro:bit to move the dot in the centre of the screen around until you have either drawn the outline of a circle, or filled up the whole screen.



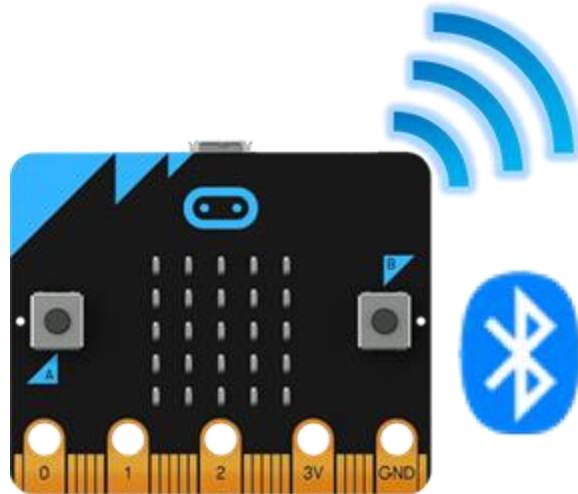
### 7 - Accelerometer

- An accelerometer measures the acceleration of your micro:bit; this component senses when the micro:bit is moved. It can also detect other actions, e.g. shake, tilt, and free-fall.



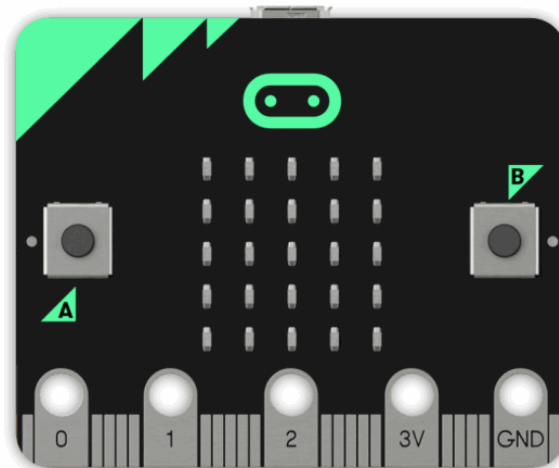
### 8 - Radio

- The radio feature allows you to communicate wirelessly between micro:bits. Use the radio to send messages to other micro:bits, build multiplayer games, and much more!



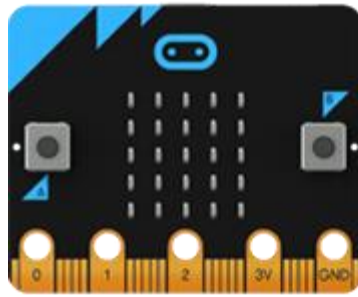
#### 9 - Bluetooth

- A BLE (Bluetooth Low Energy) antenna allows the micro:bit to send and receive Bluetooth signals. This allows the micro:bit to wirelessly communicate with PCs, Phones, and Tablets, so you can control your phone from your micro:bit and send code wirelessly to your device from your phone!
- Before using the Bluetooth Antenna you will need to pair your micro:bit with another device. Once paired, you can send scripts wirelessly to your micro:bit.



#### 10 - USB Interface

- The USB interface allows you to connect the micro:bit to your computer via a micro-USB cable, which will power the device and allow you to download scripts onto the device.

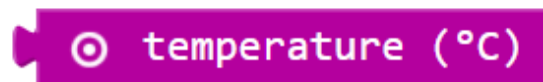


### 11 - Example 1 - Read Temperature from Microbit

Find the temperature where you are. The temperature is measured in Celsius (metric). The micro:bit can find the temperature nearby by checking how hot its computer chips are.



### 12 - Video for microbit temperature



### 13 - Returns

- a Number that means the Celsius temperature.

#### How does it work?

- The micro:bit checks how hot its CPU (main computer chip) is. Because the micro:bit does not usually get very hot, the temperature of the CPU is usually close to the temperature of wherever you are. The micro:bit might warm up a little if you make it work hard, though

```
forever loop
  show number: temperature (°C)
```

14 - Example: micro:bit thermometer

- The following example uses temperature and show number to show the temperature of the room.

```
forever loop
  show number: temperature (°C)
  show icon: thermometer
  pause (ms): 1000
```

15 - [Supplementary Exercise](#)



16 - IF.....THEN.....ELSE

Example

```

    forever
    show number temperature (°C)
    if temperature (°C) > 25
    then
    show icon [thermometer icon]
    else
    show icon [thermometer icon]
    pause (ms) 1000
  
```

```

    forever
    set c to temperature (°C)
    set f to (c * 18) ÷ 10 + 32
    show number f
  
```

17 - Example: Fahrenheit thermometer

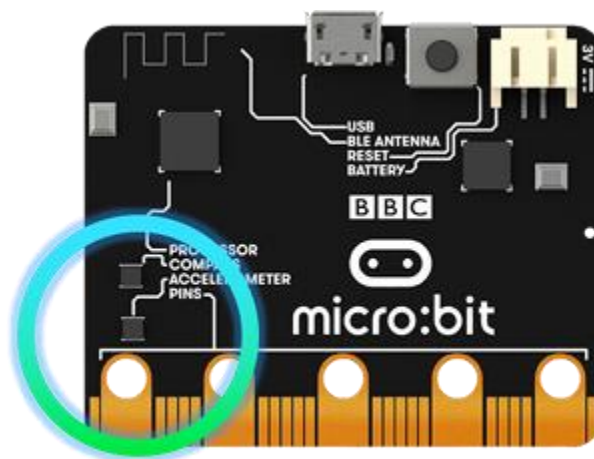
- This program measures the temperature using Fahrenheit degrees. Fahrenheit is a way of measuring temperature that is commonly used in the United States. To make a Celsius temperature into a Fahrenheit one, multiply the Celsius temperature by 18, divide by 10 and add 32.

*How to convert Celsius to Fahrenheit? 0 degrees Celsius is equal to 32 degrees Fahrenheit:*

- 0 °C = 32 °F

*The temperature T in degrees Fahrenheit (°F) is equal to the temperature T in degrees Celsius (°C) times 9/5 plus 32:*

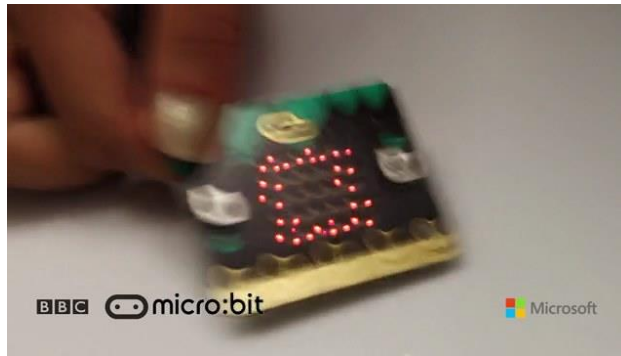
- $T(^{\circ}F) = T(^{\circ}C) \times 1.8 + 32$



18 - Example 3 - rock paper scissors game

*This tutorial will help you create a game of rock paper scissors with the BBC micro:bit.*

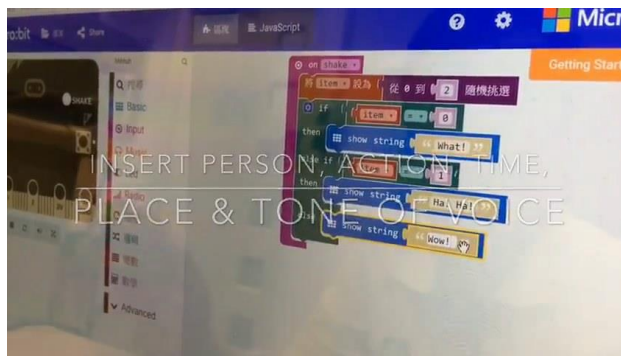




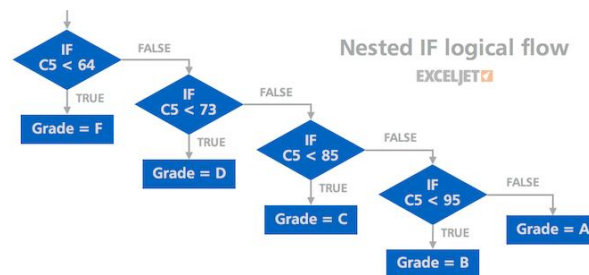
<https://sway.com/WK3OPqNIAnsxPYH4#content=MrMnQiDruLpGSN>



<https://sway.com/WK3OPqNIAnsxPYH4#content=piL6F3mKAIyFEX>



<https://sway.com/WK3OPqNIAnsxPYH4#content=HmLhAfqJzFKUpe>



19 - Nested IF



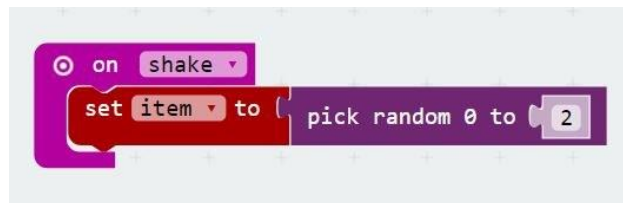
20 - [Supplementary Exercise](#)

When the A button is pressed, increment the score by 1. You can select Game drawer then add change score by 1.

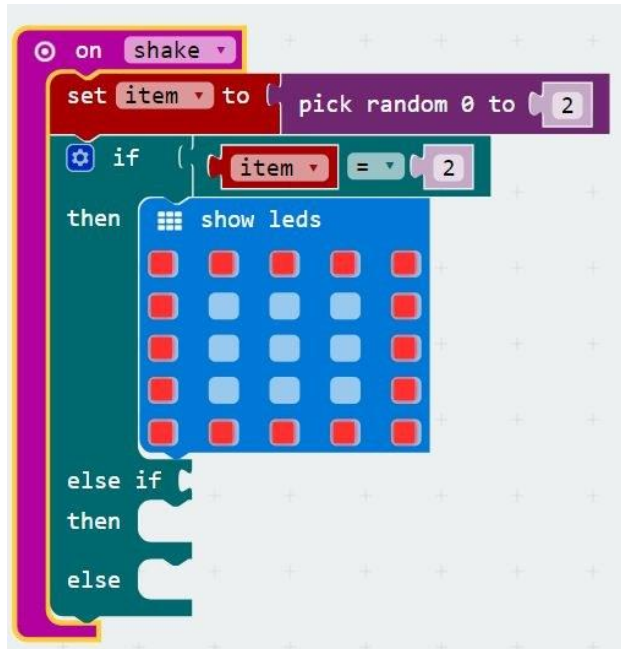
After incrementing the score, display the total number of wins you have.



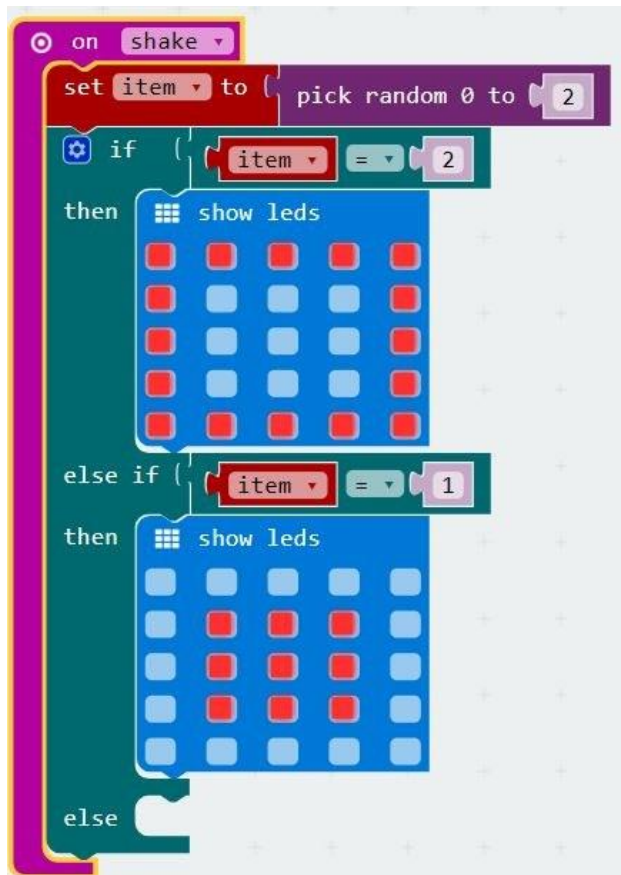
21 - We want the BBC micro:bit to choose rock, paper, or scissors when it is shaken. Let's begin by creating an on shake condition so the micro:bit will run code when it is shaken.



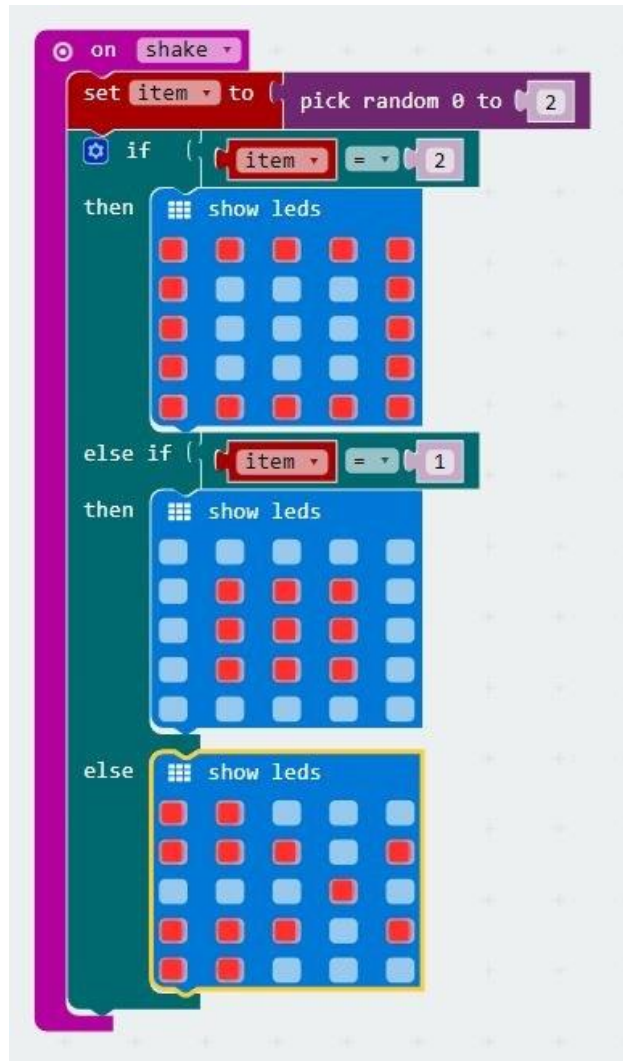
22 - Next, create a variable and store pick random number from 0 to 2. On shake, a number will be randomly picked from 0-2. We will randomly display an image based on the random number returned.



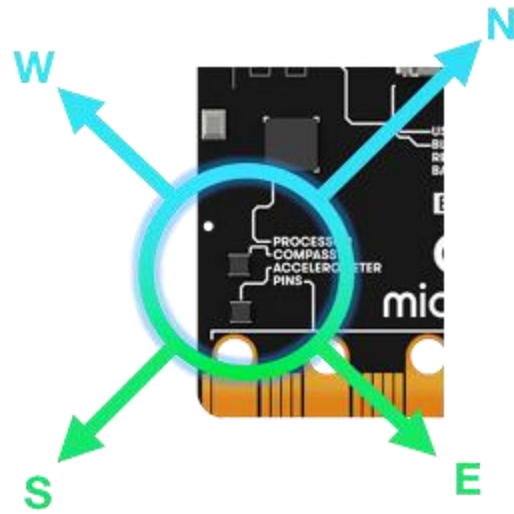
23 - The BBC micro:bit will look like it's showing 1 frame of the image by displaying the whole image when pick random is equal to 2. We can help the BBC micro:bit randomly decide which image to use by pick random. The BBC micro:bit will randomly pick the image to display with show LEDs and the pick random function.



24 - The BBC micro:bit will look like it's showing 1 frame of the image by displaying the whole image when pick random is equal to 1. We can help the BBC micro:bit randomly decide which image to use by pick random. The BBC micro:bit will randomly pick the image to display with show LEDs and the pick random function.

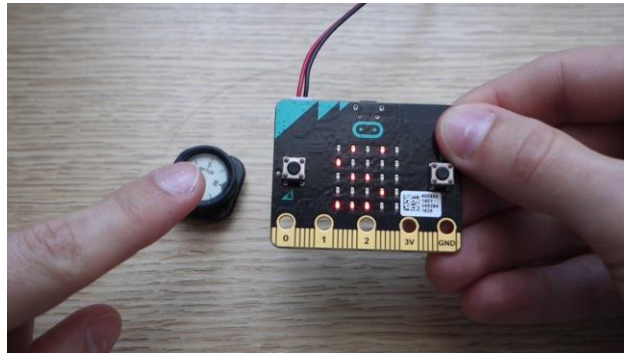


25 - The BBC micro:bit will look like it's showing 1 frame of the image by displaying the whole image when pick random is not equal to 2 and not equal to 1. We can help the BBC micro:bit randomly decide which image to use by pick random. The BBC micro:bit will randomly pick the image to display with show LEDs and the pick random function.

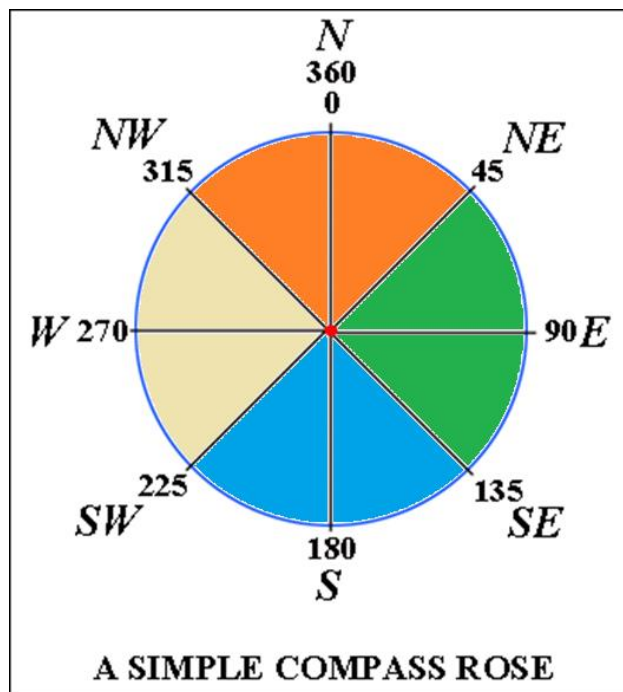
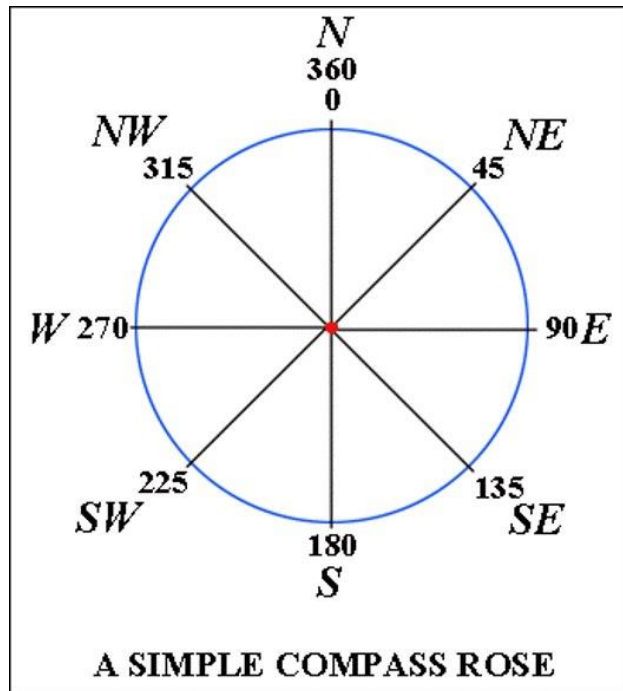


*26 - Example 2 - compass activity*

*This tutorial will show you how to program a script that displays the direction the BBC micro:bit is pointing.*

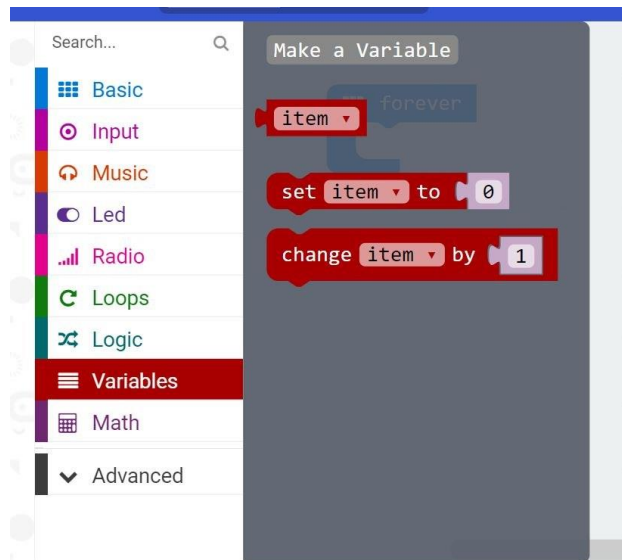


*27 - Get the compass heading of the BBC micro:bit in degrees. Your BBC micro:bit has a built-in **magnetometer** so it can your direction with respect to the North Magnetic Pole.*

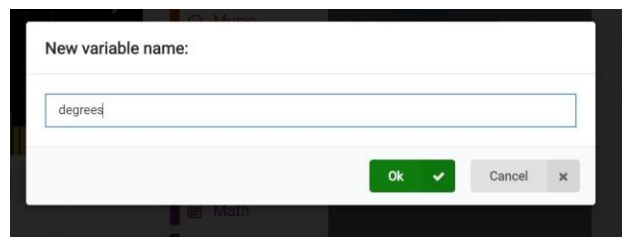




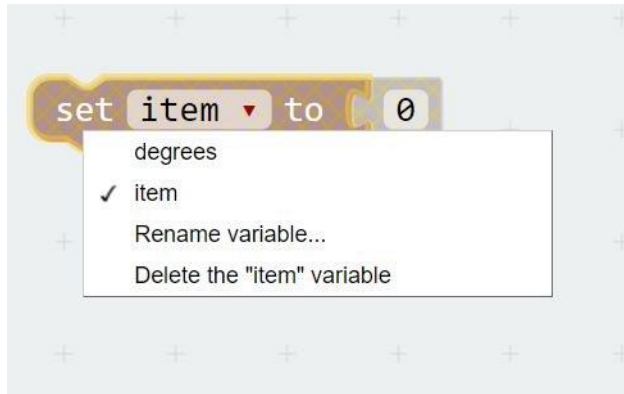
28 - Create a loop that will continuously update the reading of the compass.



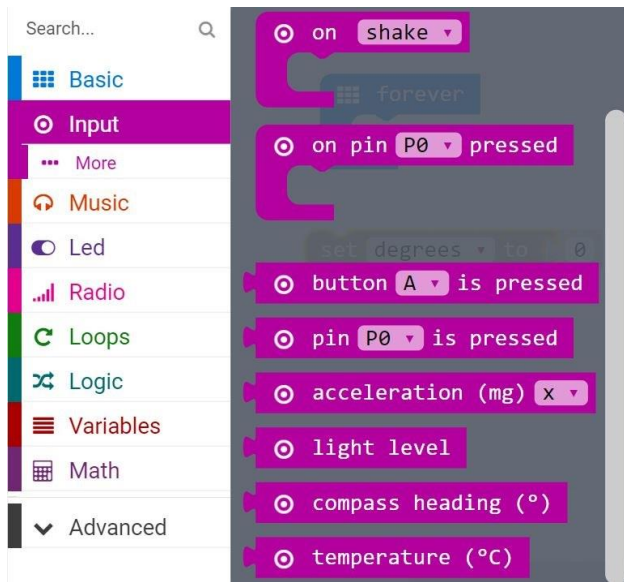
29 - Store the reading of the micro:bit in a variable called degrees (1)



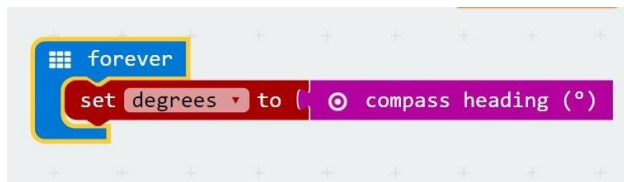
30 - Store the reading of the micro:bit in a variable called degrees (2)



31 - Store the reading of the micro:bit in a variable called degrees (3)



32 - Store the reading of the micro:bit in a variable called degrees (4)



33 - Store the reading of the micro:bit in a variable called degrees (5)



```
forever
  set degrees to compass heading (°)
  if (item < 45)
  then show string "N"
  else if
  then
  else if
  then
  else if
  then
  else
```

34 - If degrees is less than 45, then the compass heading is mostly pointing toward North. Display N on the BBC micro:bit.

```
forever
  set degrees to compass heading (°)
  if (item < 45)
  then show string "N"
  else if (item < 135)
  then show string "E"
  else if
  then
  else if
  then
  else
```

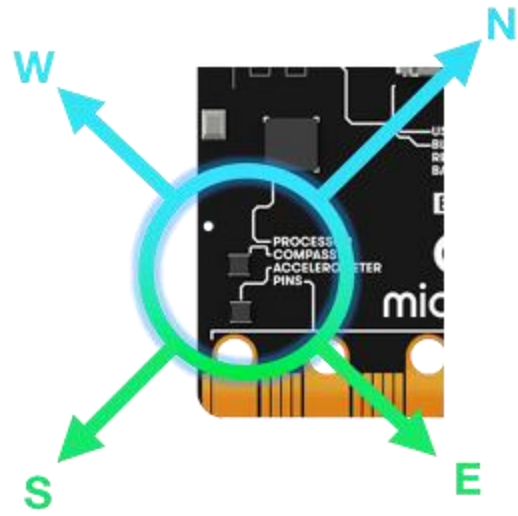
35 - If degrees is less than 135, the micro:bit is mostly pointing East. Display E on the BBC micro:bit.

```
forever
  set degrees to compass heading (°)
  if (item < 45)
  then
    show string " N "
  else if (item < 135)
  then
    show string " E "
  else if (item < 225)
  then
    show string " S "
  else if
  then
  else
```

36 - If degrees is less than 225, the micro:bit is mostly pointing South. Display S on the BBC micro:bit.

```
forever
  set degrees to compass heading (°)
  if (item < 45)
  then
    show string " N "
  else if (item < 135)
  then
    show string " E "
  else if (item < 225)
  then
    show string " S "
  else if (item < 315)
  then
    show string " W "
  else
    show string " N "
```

37 - If none of these conditions returned true, then the BBC micro:bit must be pointing West. Display W on the BBC micro:bit.



38 - Supplementary Exercise

*Display your own unique message or led picture for each direction.*