

technocamps



Llywodraeth Cymru
Welsh Government



Prifysgol
Abertawe
Swansea
University



Cardiff
Metropolitan
University

Prifysgol
Metropolitan
Caerdydd



University of
South Wales
Prifysgol
De Cymru

Cyngor Cyllido Addysg
Uwch Cymru
Higher Education Funding
Council for Wales

hefcw



PRIFYSGOL
ABERYSTWYTH
UNIVERSITY



Prifysgol Wrexham
Wrexham University



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

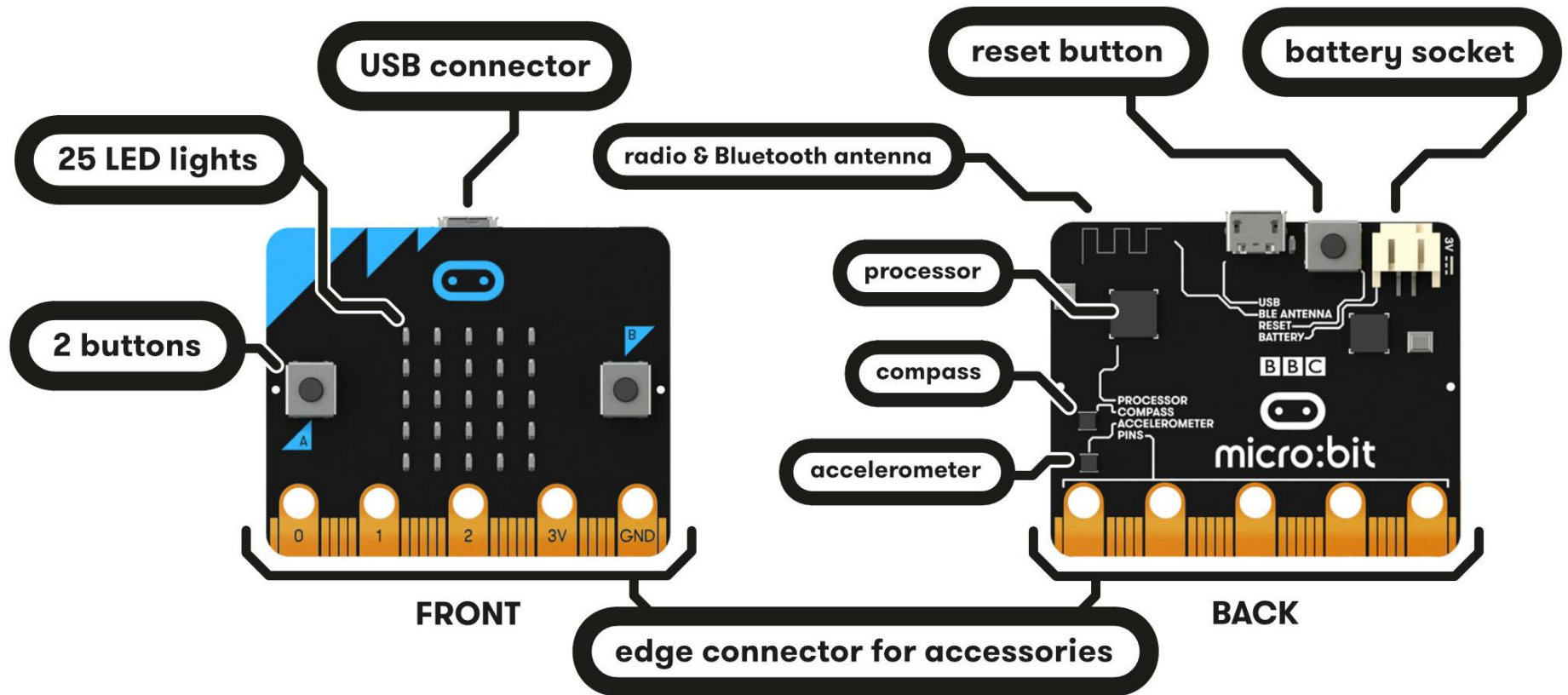
The Electrifying Micro:bit





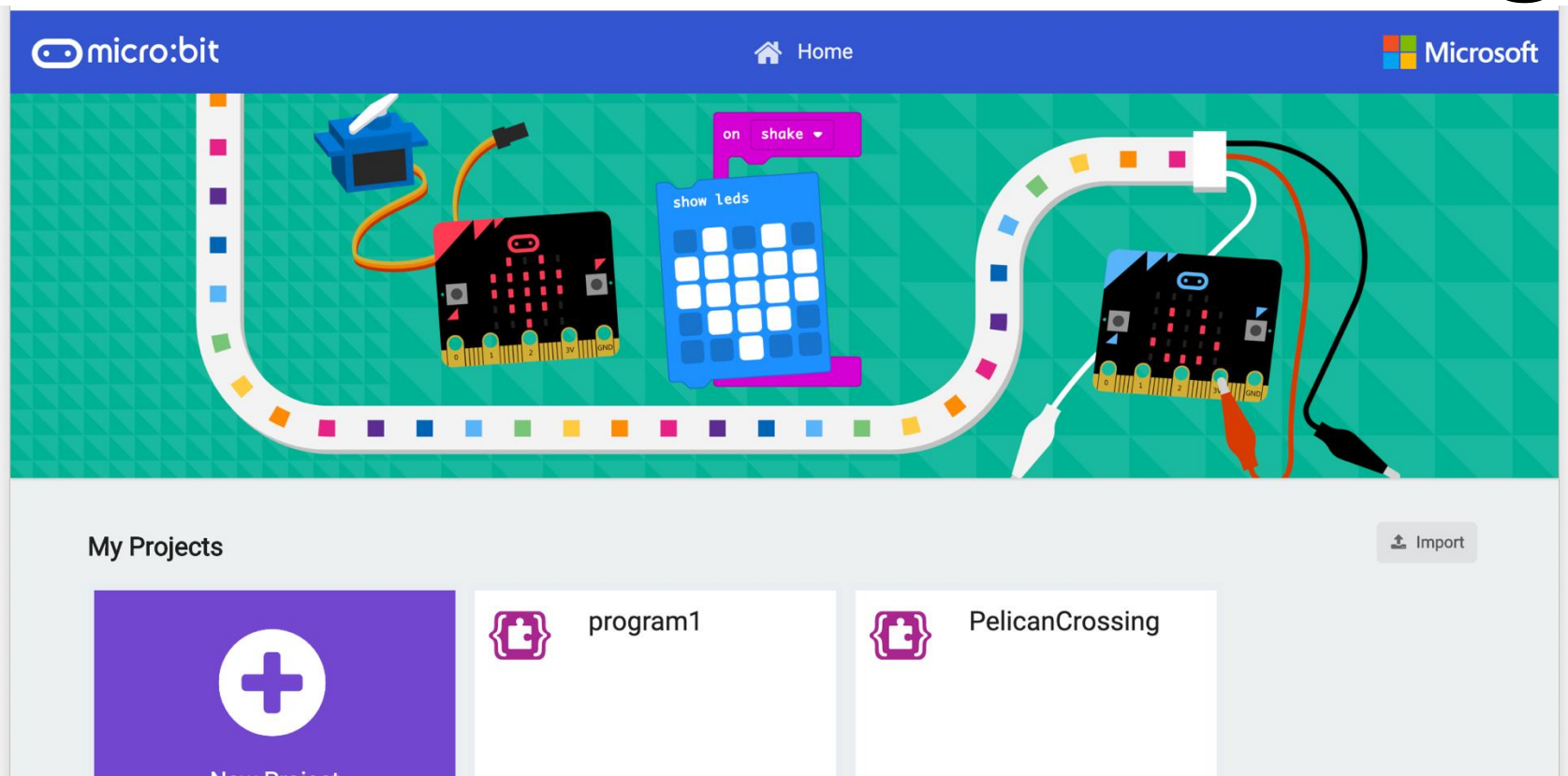
Intro to micro:bit

What is a micro:bit?



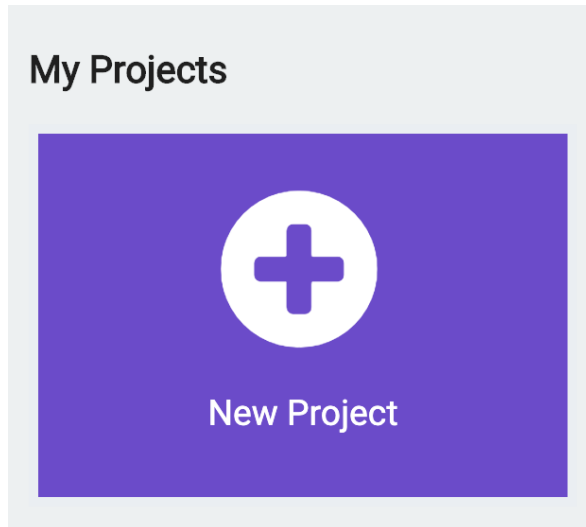
Starting with Makecode

makecode.microbit.org

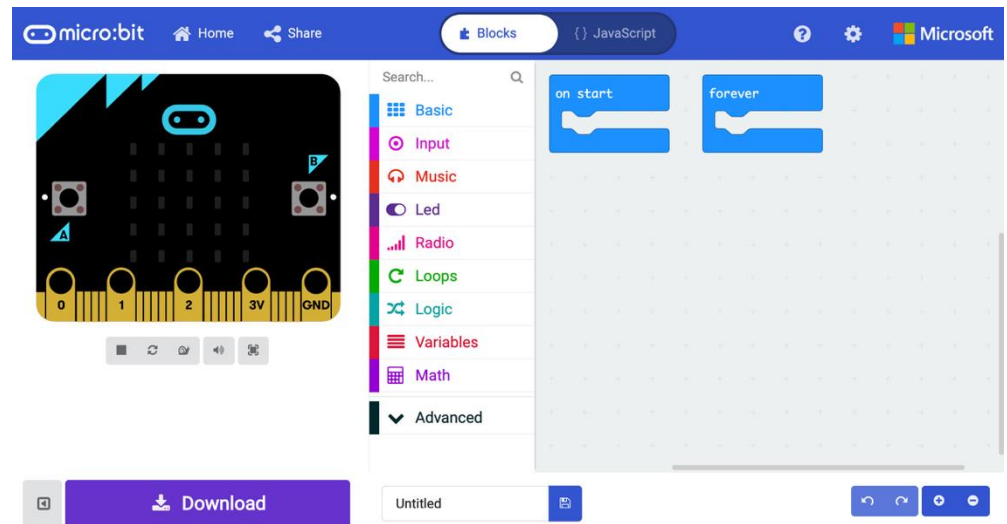


Starting with Makecode

Click New Project

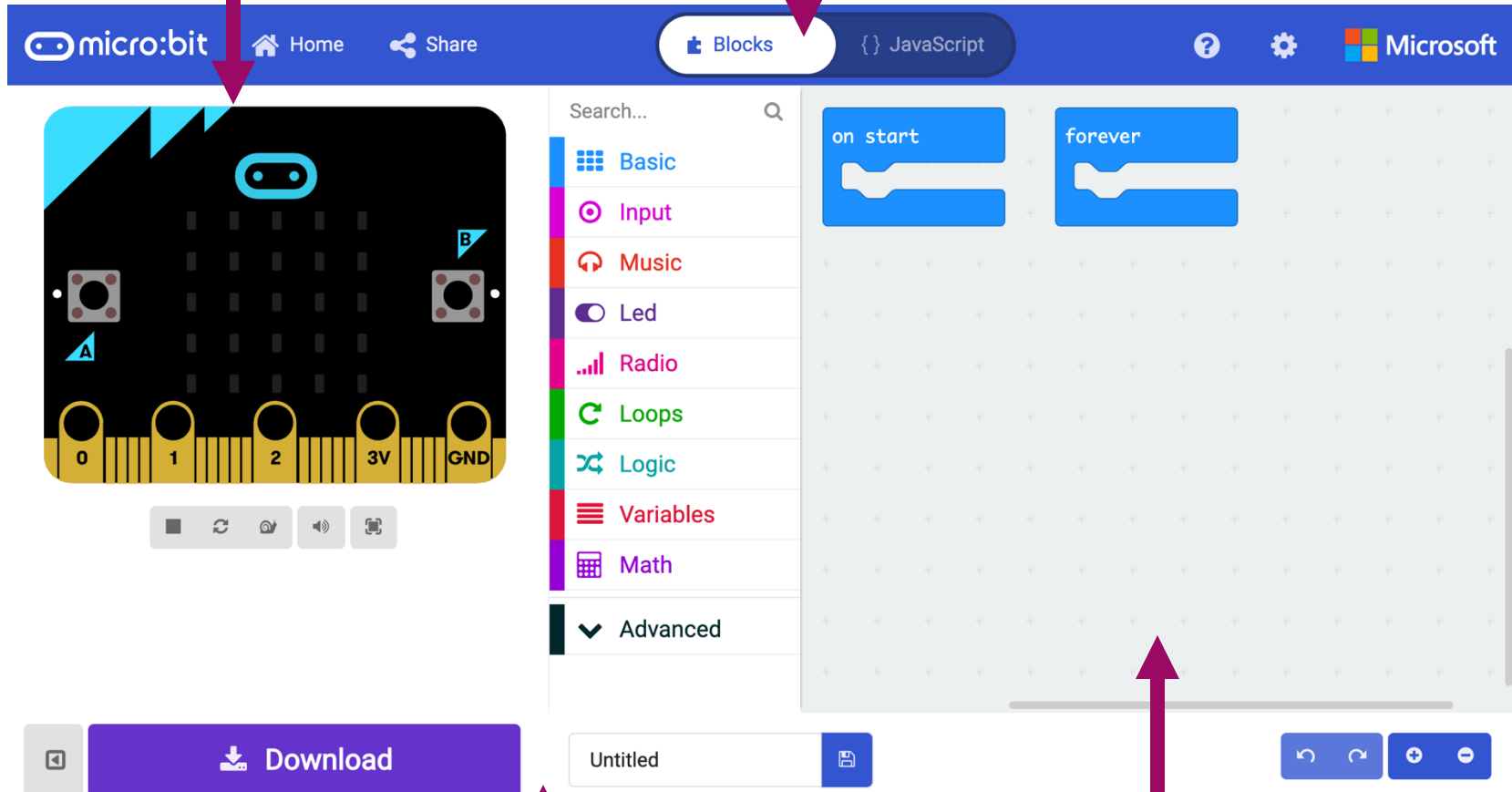


It should look like this!



Simulator

Language



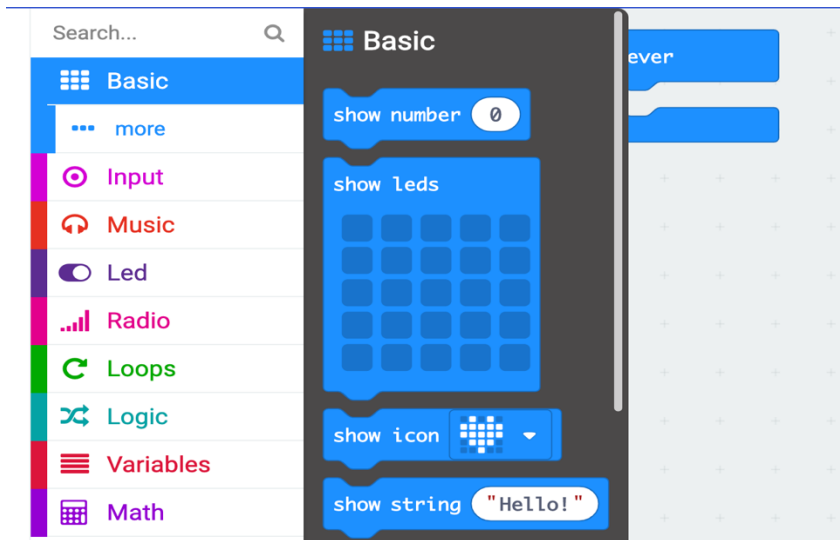
Save and Download

Code Window

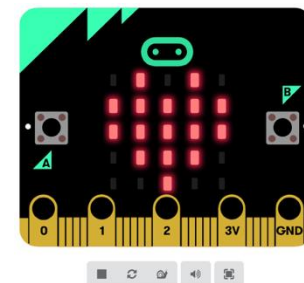


Activity: Hearty Start

Hearty Start



1. Click on Basic
2. Click the show icon command
3. Drag and drop it into the forever block.
4. What happens to the simulator?



Uploading Code to micro:bit

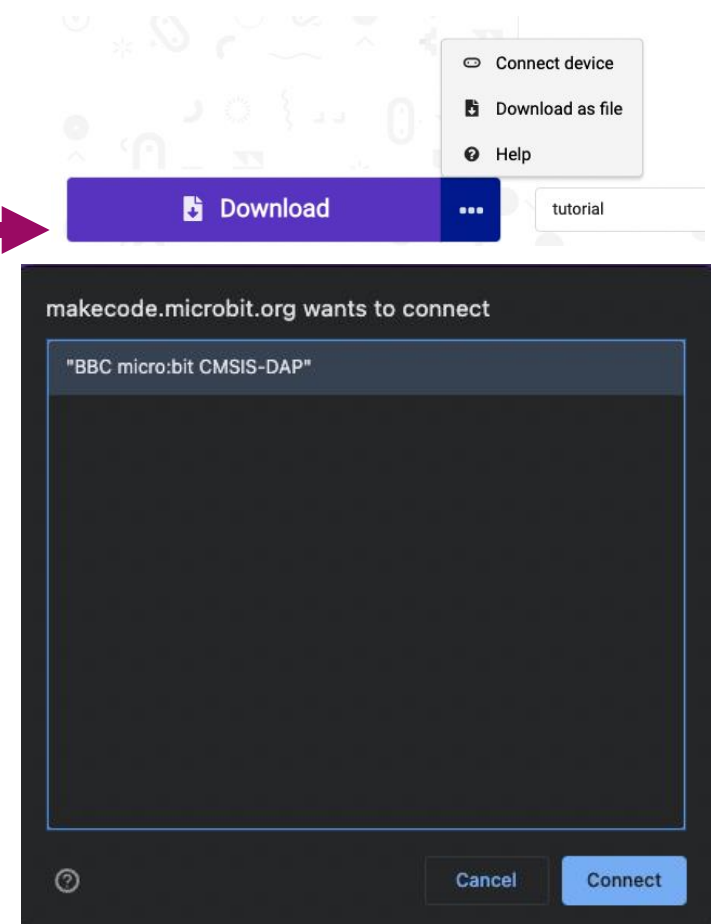
1. Plug the micro:bit into your computer

2. In the bottom left of your screen, click the 3 dots next to 'Download', then click 'Connect Device'

3. Follow the on screen instructions until you see this popup

4. Click the name of your device (it should be the only option)

5. Click connect

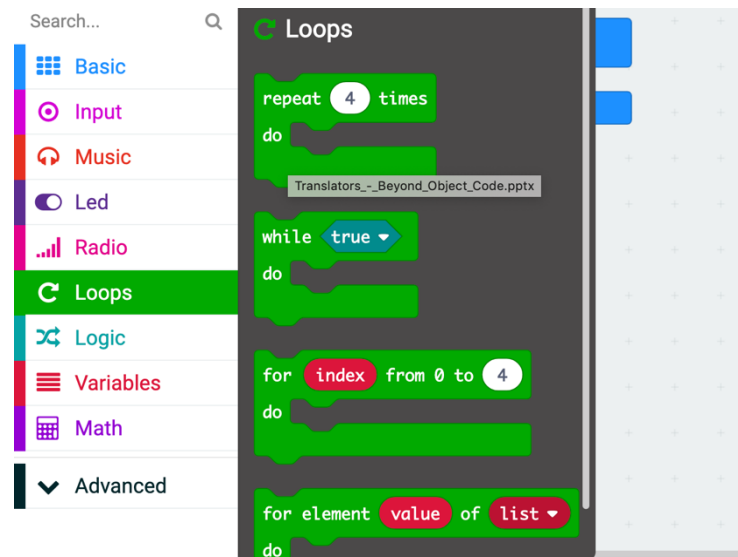




Activity: Heart Beat

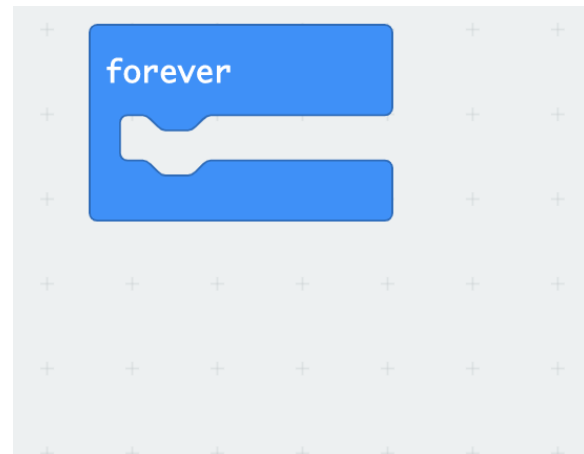
Loops

- Feature of a programming language used to repeat a set of commands.
- The commands can be iterated/looped forever, for a specific number of times, or for a given condition.
- We use the commands from the **Loops** section.



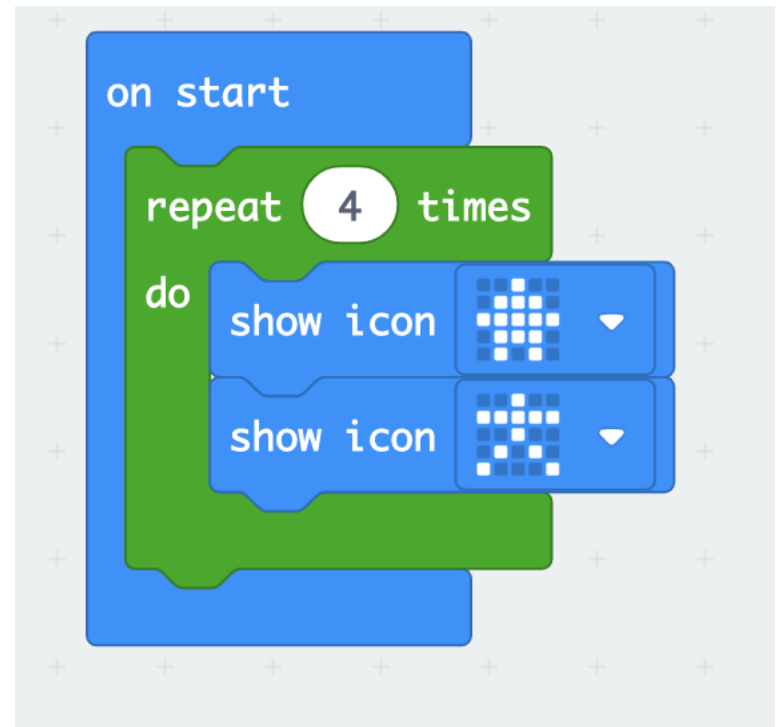
Default Loop

- We have a default loop command when we start micro:bit project.
- The forever command which appears by default runs a set of commands until the micro:bit is unplugged or reset.
- You can have only one forever in the micro:bit code.



On Start vs Forever

- **on start** will run the code as soon as the micro:bit starts and ends once the code ends.
- In this example, it will show the two icons 4 times and stops.
- **forever** will run the code well...forever.
- Does it make sense to have a loop inside a forever loop?



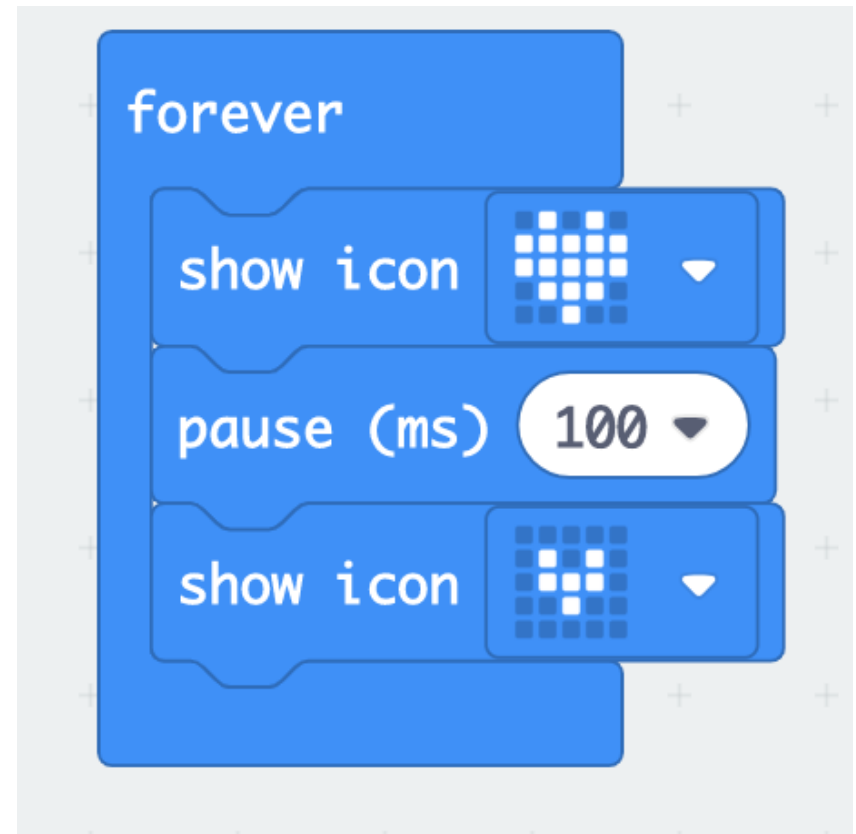
Heart Beat

We have created a simple application which shows the heart icon on the micro:bit

We can extend our app to show an animation

Add a **pause (ms) 100** command and add a miniature heart icon after the pause

Try adding more animation with repeat loops to make a more complex programme





Activity: An Eventful Button Press

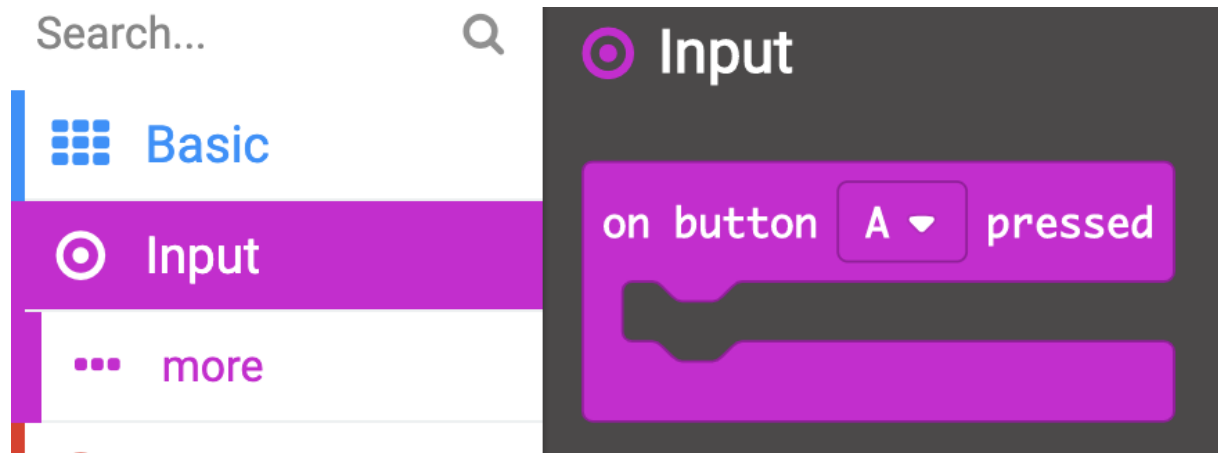
Events

Micro:bit has two buttons, button **A** and button **B**

These two buttons help us use event-based programming

For example, we can show a happy face when we press button A and a sad face when we press button B

We need to use the **Input** commands for our events

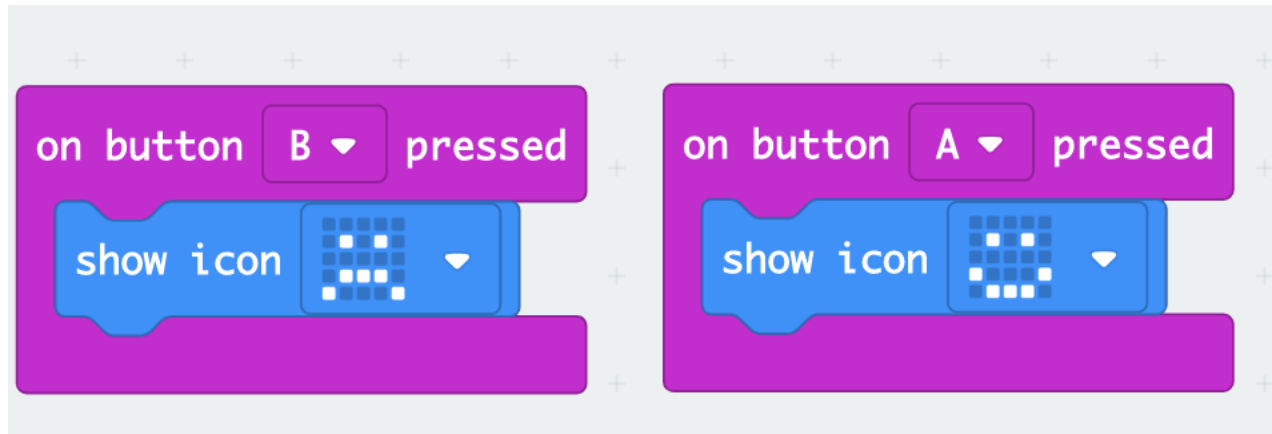


Events

We can use multiple **Input** commands – one for each button

Remember that if we add two **on Button A pressed** the micro:bit would not know which one to choose – ensure you use only one button A pressed and one button B pressed event

Have a go!



Activity: Sensing the Environment



Activity: Sensing the Environment

We can turn our micro:bit into a light sensor

Use the input block "light level" and the Led block "plot bar graph of"



Test the code, and try to work out exactly what is happening here

How can the micro:bit measure light intensity?



How did your
light sensor
work?

Electricity



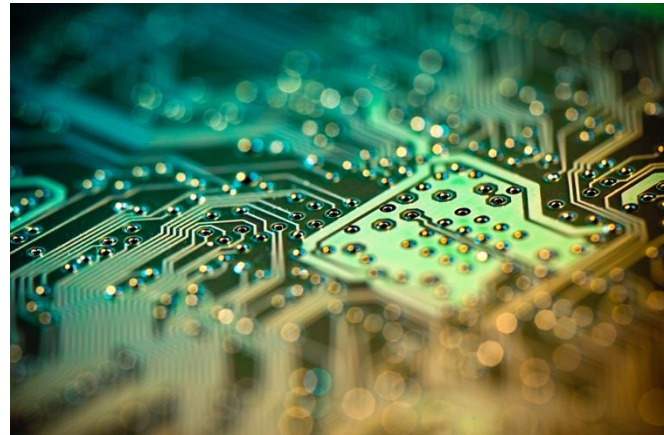
Electricity

The sensor generates an **electrical current** from light – like a solar panel

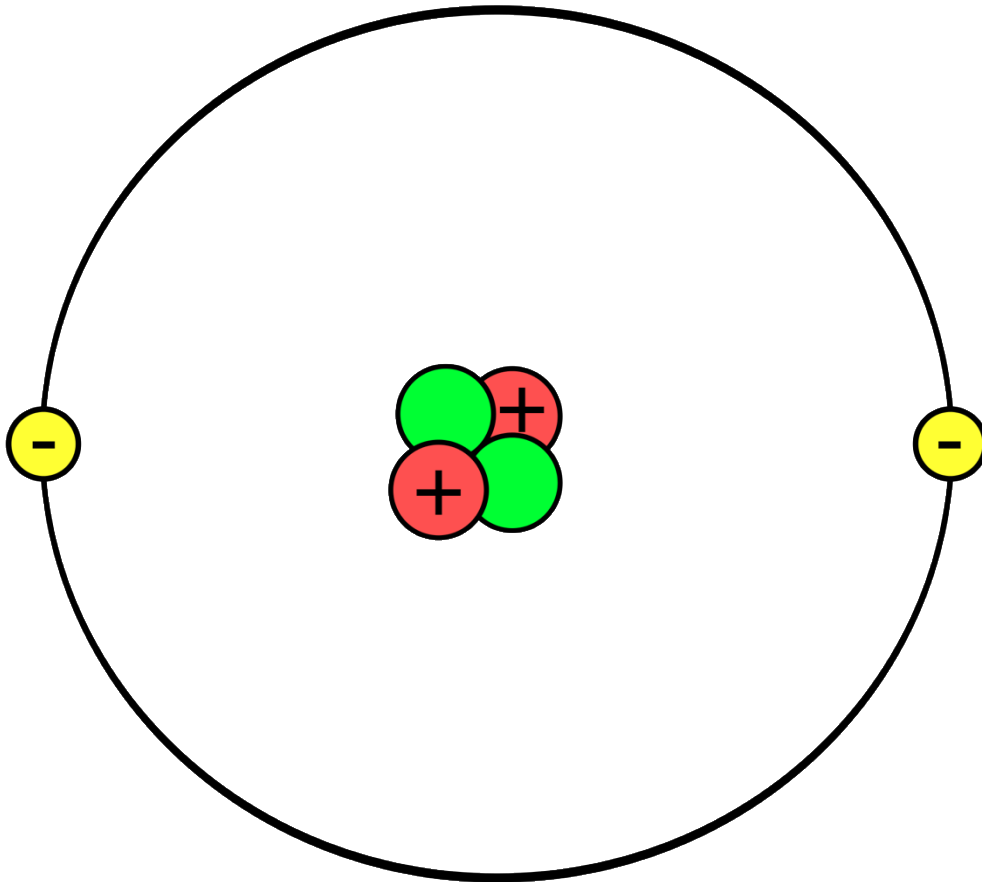
Our micro:bits can understand the strength of this current and show us a visual representation

They use electricity to **indirectly** measure brightness of light

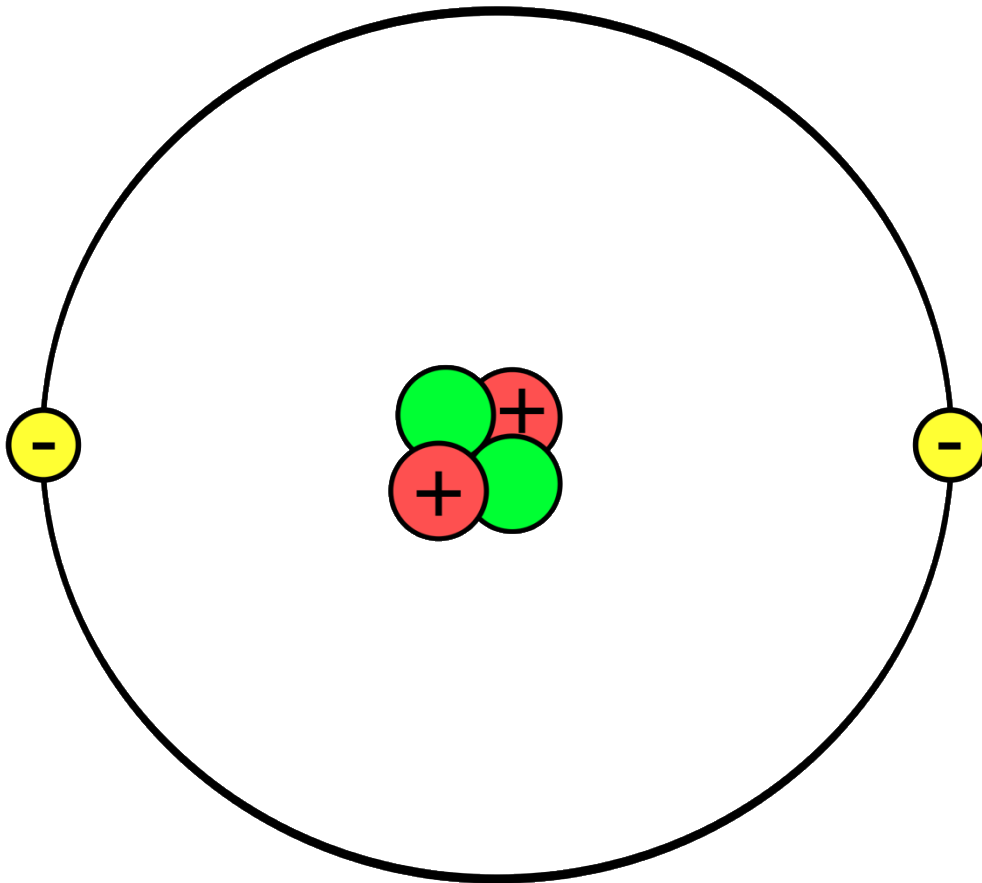
How does electrical current move through our micro:bits?



Electricity



Electricity



Electricity is the movement of **electrons**

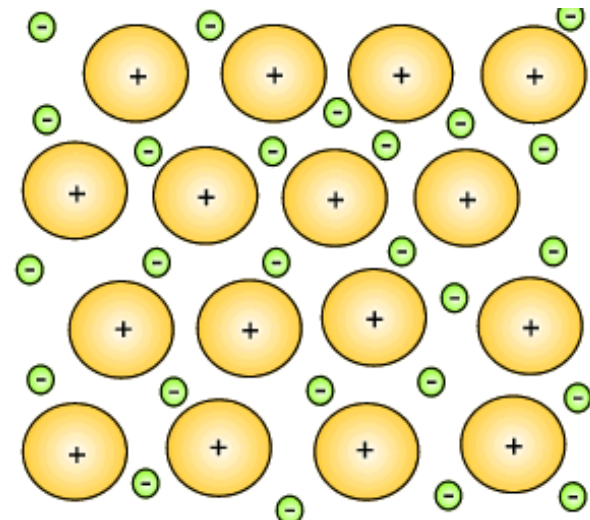
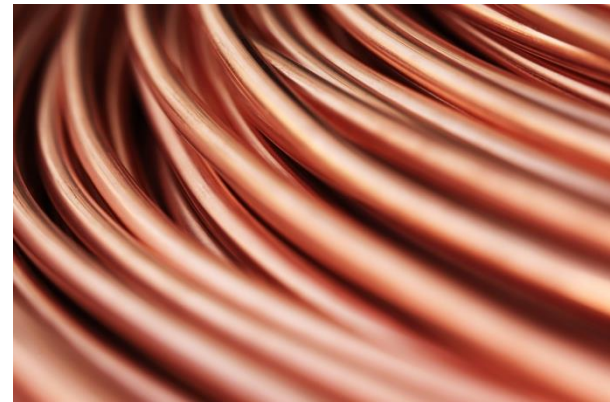
Conduction

Many metals are excellent **electrical conductors**

Metal atoms have a **free electron** that can jump from atom to atom

By sharing electrons in this way metal atoms allow for the **movement of electrons**

What happens when electrons move?

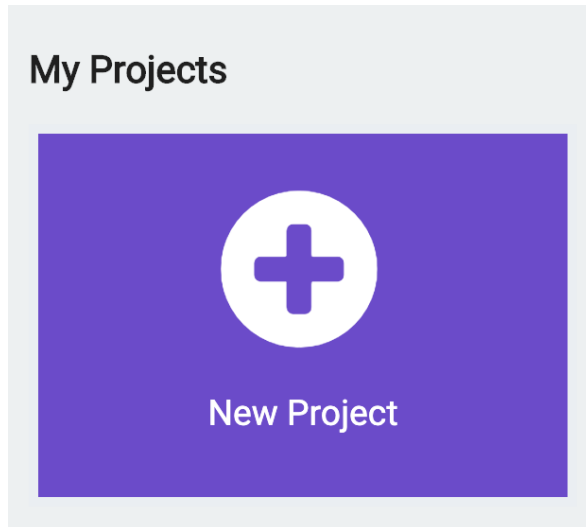


Activity: Conductivity Checker

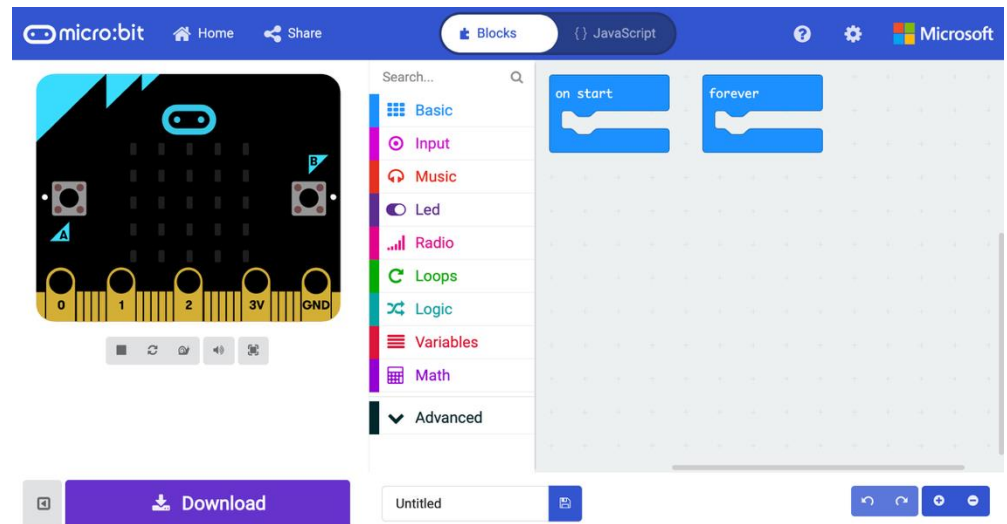


Starting with Makecode

Click New Project



It should look like this!



Conditions

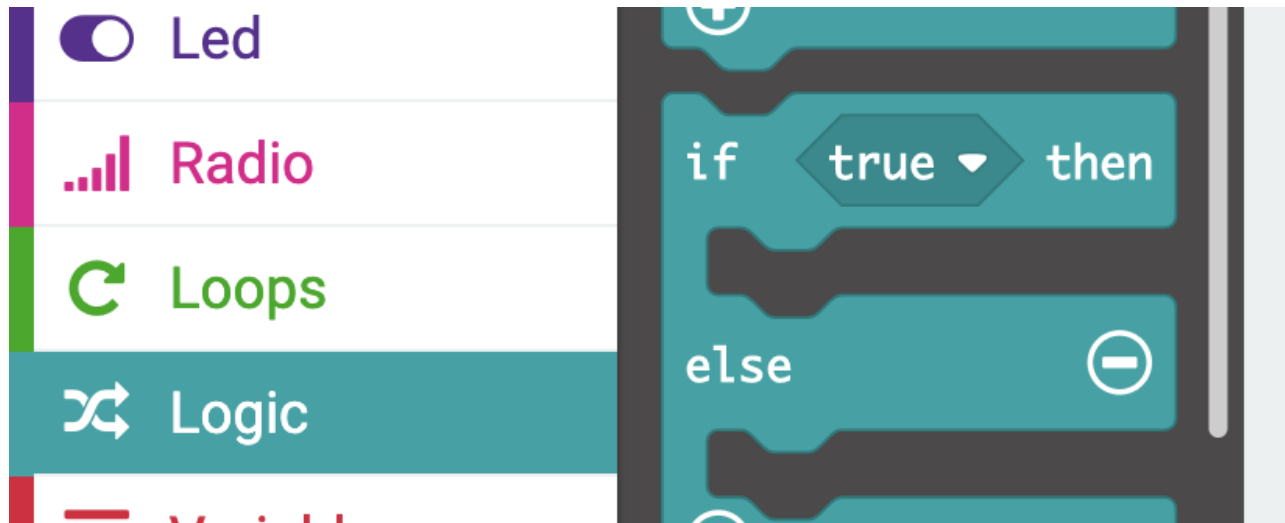
- Feature of a programming language which helps us perform different actions based on different conditions.
- These conditions should always result either in Yes/No or True/False.
- Example:
 - If your age is > 17 then**
You are eligible to drive (given that you have passed your tests).
 - else**
You have to wait until you are eligible.
- The condition in the example above can either be true or false. Your age can either be > 17 or less than 17.
- What happens if your age is 17?

Conditional Expressions

- You can create conditions using expressions.
- These expressions use mathematical operators.

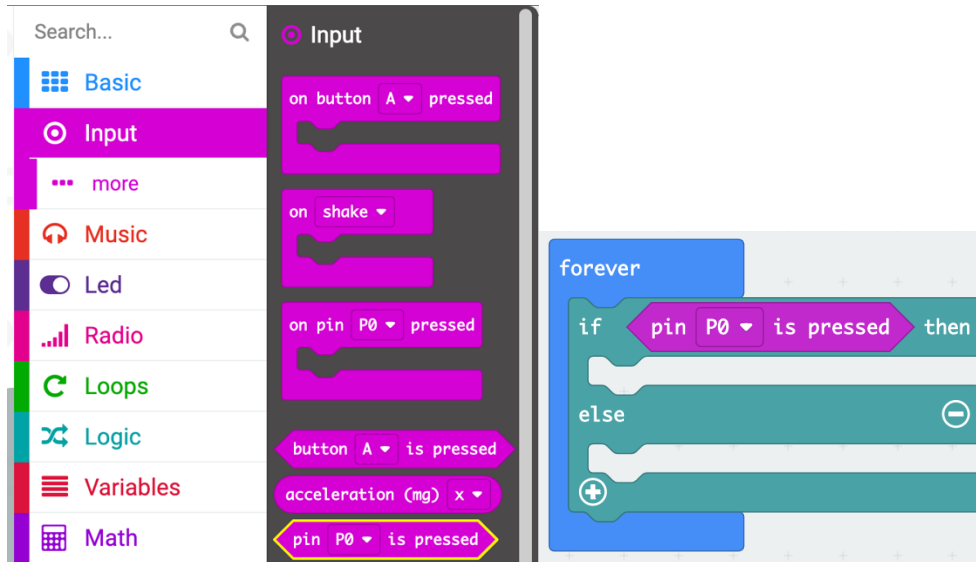
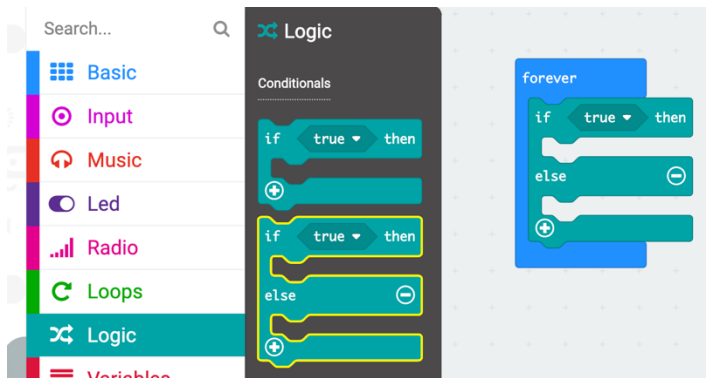
Operator	Meaning
<	Less than
>	Greater than
<= or ≤	Less than or equal to
>= or ≥	Greater than or equal to
=	Equal to
≠	Not equal to

Logic



- In Micro:bit these conditional statements are placed under [Logic](#) section.
- It has the conditionals, the comparison operators and the logical operators.

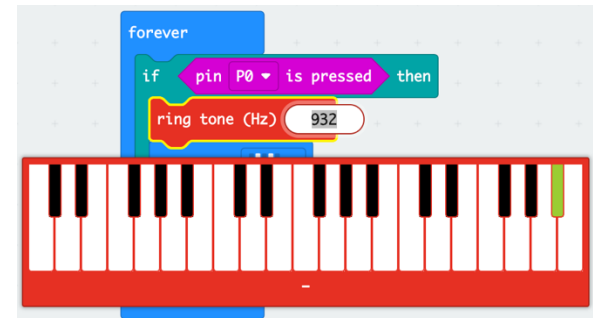
Find the 'If Else' Block



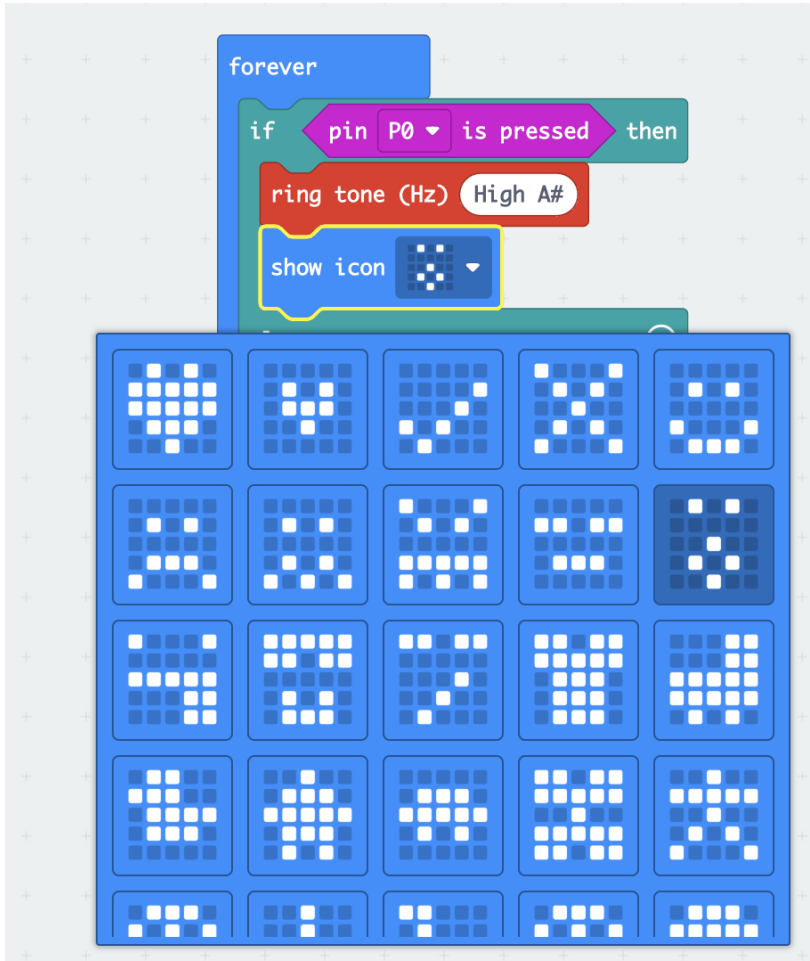
1. Click logic
2. Drag 'if else' block into 'forever'
3. Click input
4. Drag 'pin P0 is pressed' block into 'true' on the 'if else' block
5. Make sure P0 is selected (not P1, P2 etc...)

Set the Sound

1. Click music
2. Drag 'ring tone (Hz)' block
3. Drag 'stop all sounds' block into 'else'
4. Click the note selector on the 'ring tone (Hz)' block and choose a note on the keyboard



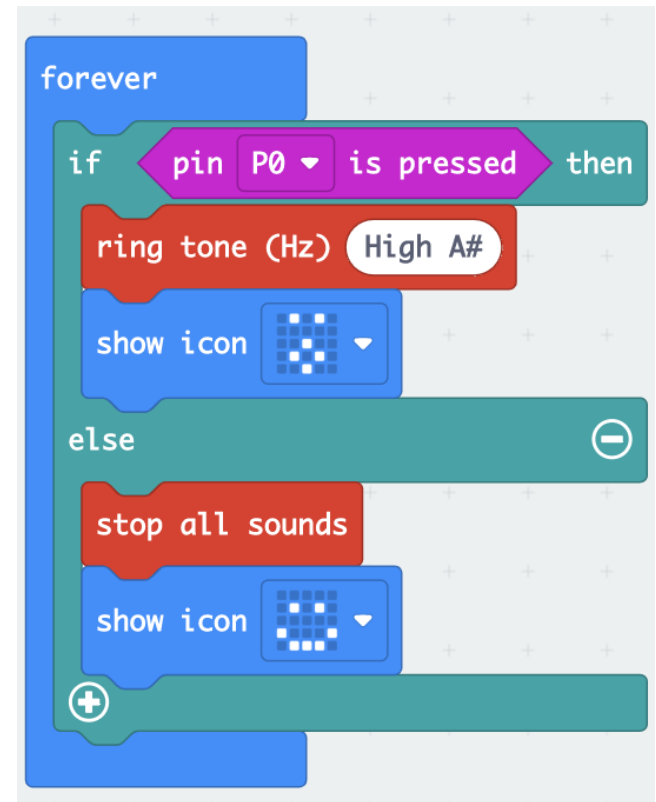
Set the Icon

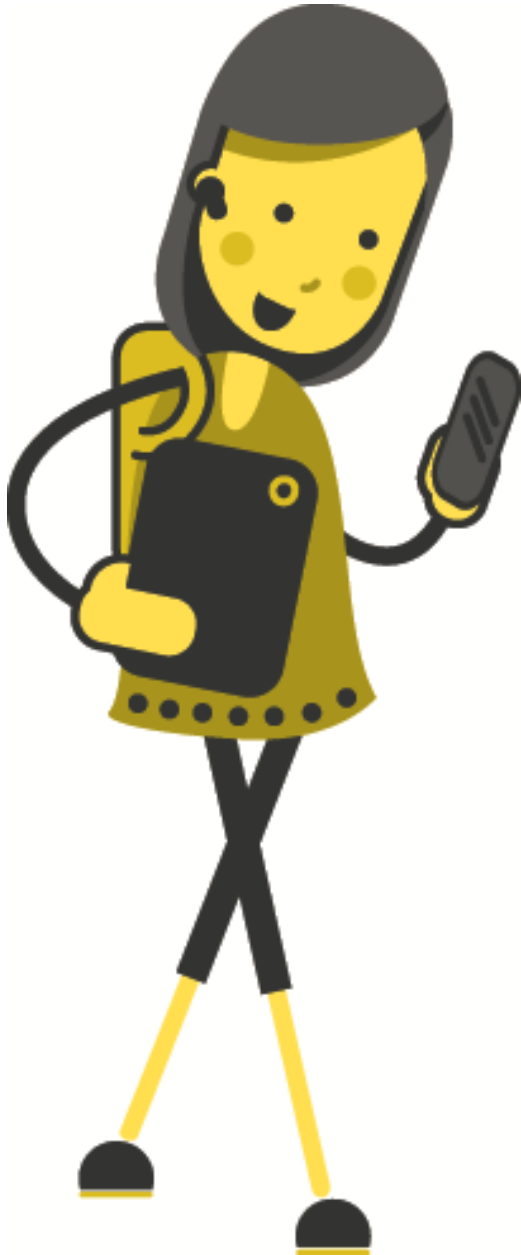


1. Click basic
2. Grab the 'show icon' block twice
3. Drag one under 'ring tone Hz'
4. Drag the other under 'stop all sounds'
5. Choose a different icon for each
6. You could use the 'show leds' block instead and draw your own icons.

Finishing Touches

1. Your final code should look like the this
2. Attach your crocodile clips to 'Pin 0' and 'GND' on your micro:bit
3. Download your code to your micro:bit
4. Touch the loose ends of the crocodile clips together to make sure it works
5. Try holding the end of each crocodile clip in a different hand to see if you are conductive!





What else
conducts
electricity?

Extensions!

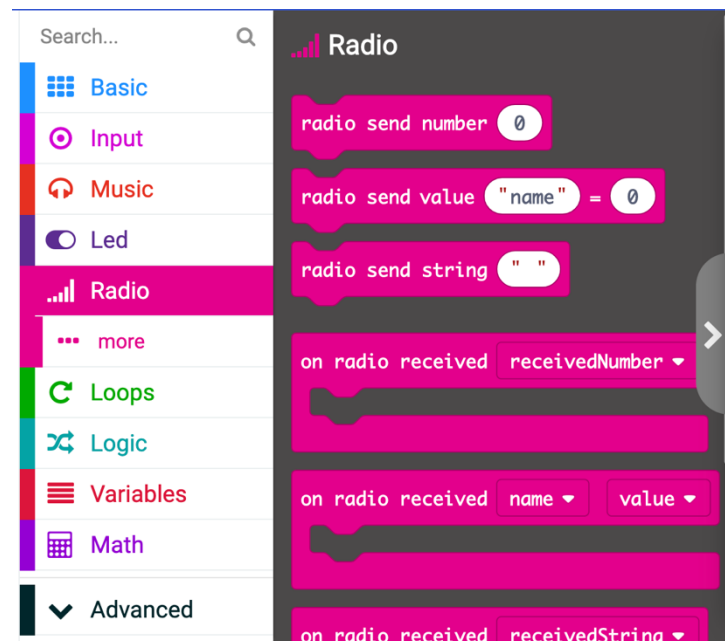




Activity: Radio Communication

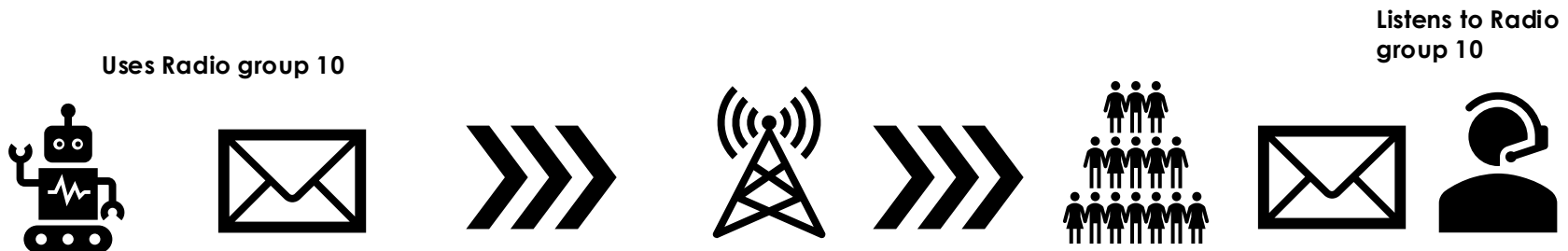
Radio Communication

- We can use the Bluetooth feature of micro:bit to communicate with other micro:bits.
- The commands necessary are available in the Radio group.



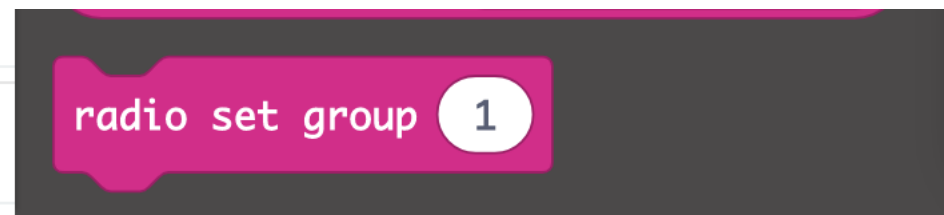
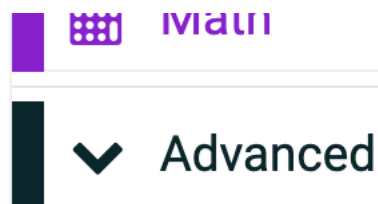
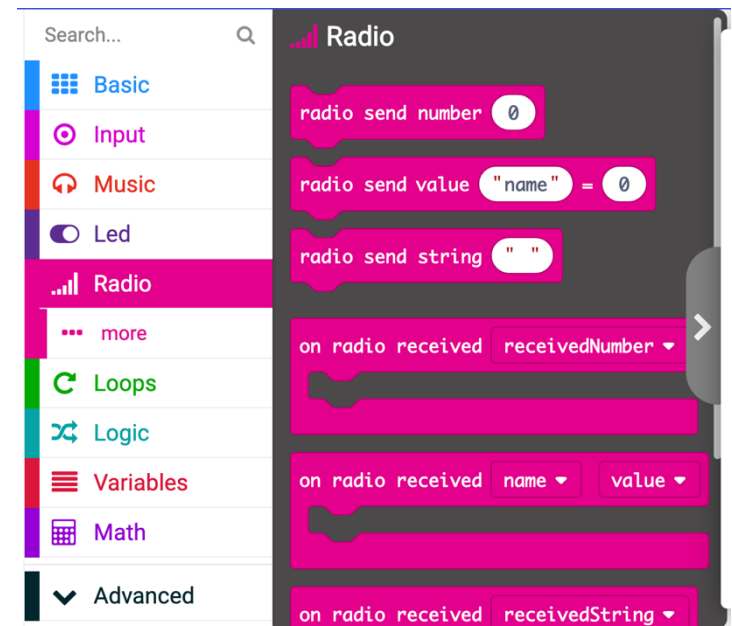
Radio Communication

- The sender uses a radio group (say 10).
- The sender sends the message using that radio number.
- It is transmitted to all micro:bit's.
- The receiver who is listening to the same radio group receives the message.



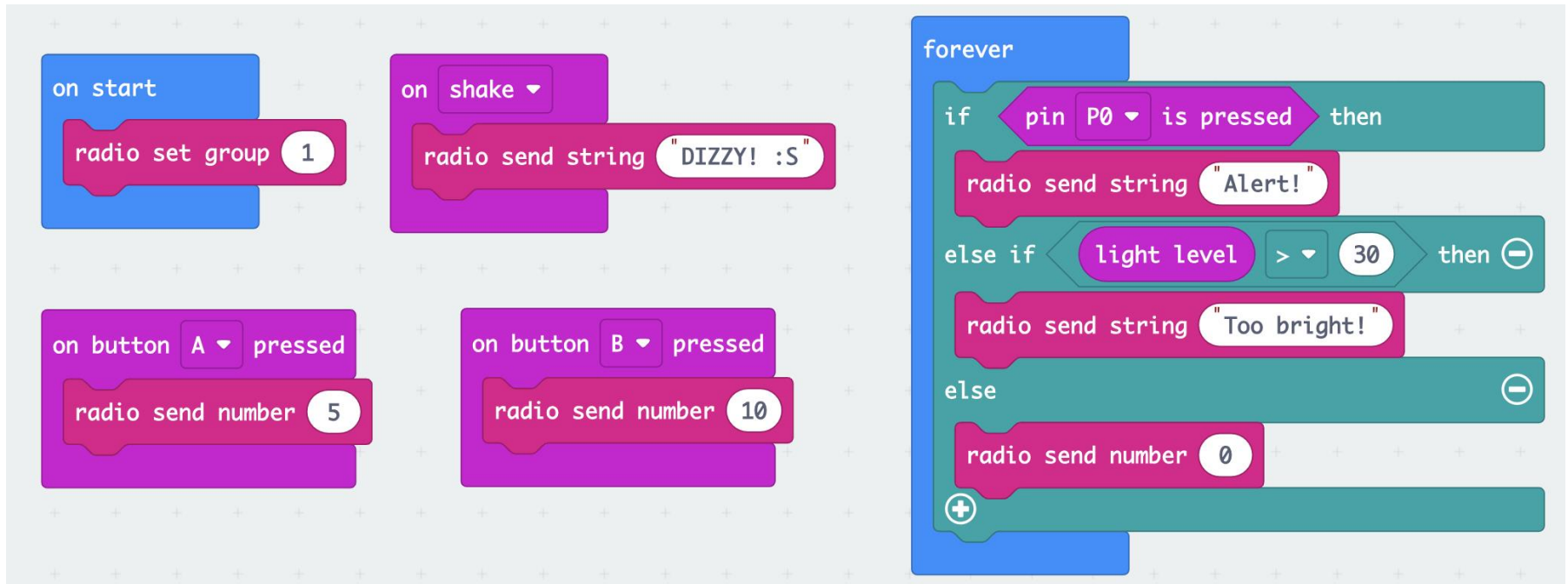
Sending Messages / Numbers

- There are two different commands to send messages and numbers.
- To send numbers we use **radio send number**.
- To send text messages we use **radio send string**.
- Do not forget to set the **Radio Group**.



Radio Communication

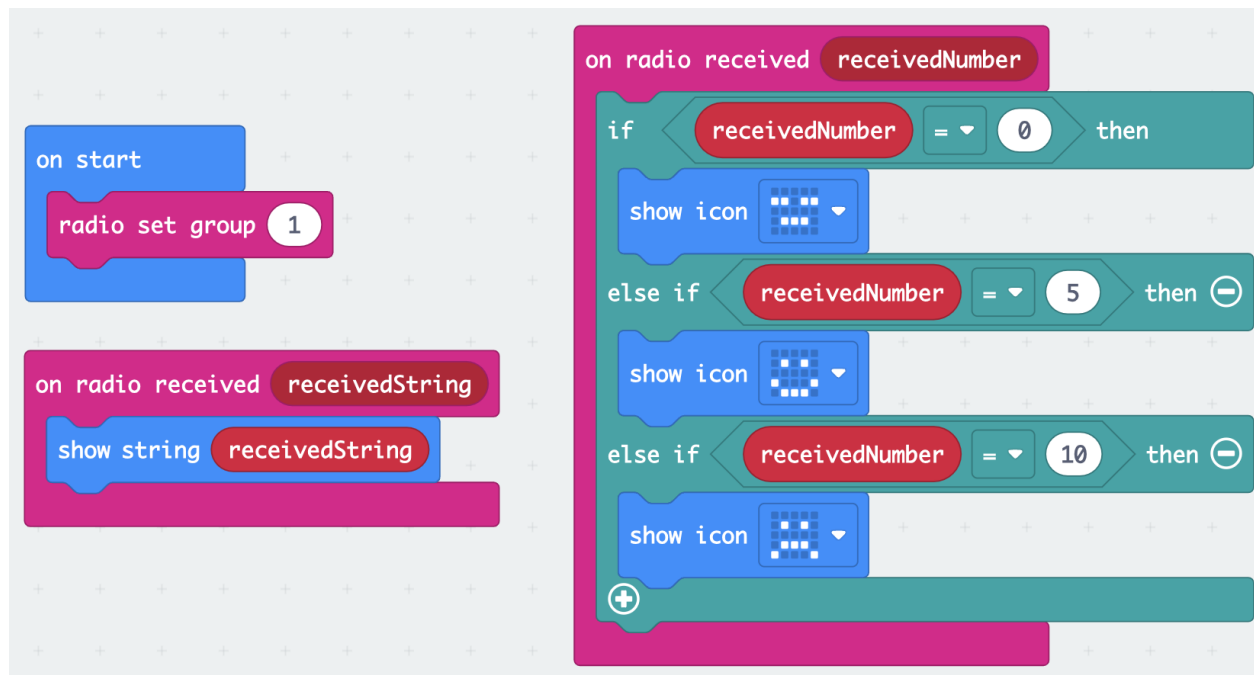
You can set the radio to send messages for all kinds of inputs – have a play around and see what you can send to your friends!



Don't forget to set the radio group first!

Radio Communication

And of course you can set the radio to receive messages for all kinds of signals – have a play around and see what you can send to your friends!



Don't forget to set the **radio group** first!



Activity: DIY Micro:bit

DIY Micro:bit Project

You can do nearly anything with a micro:bit

Come up with something you'd like your micro:bit to do and try and make it work

Or just play around and see what you can make!

If you can't think of anything, try making a rock paper scissors game.