

# Lesson 1

## Introduction to the Micro:bit



**CS IN SCHOOLS**

# Learning objectives

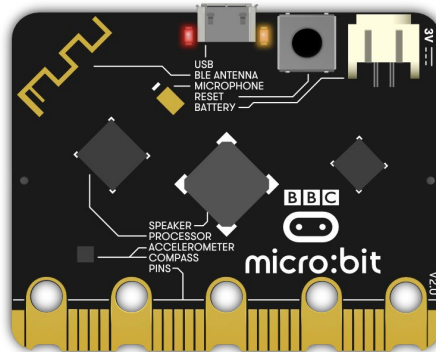
By the end of this lesson, you should be able to:

- understand what you will achieve in this course
- have access to the course resources
- understand what a **micro:bit** device is and how it communicates with your computer via **Bluetooth**
- **flash** your micro:bit device and connect to it from the Python editor
- display some text and patterns on the micro:bit LEDs



# The micro:bit

- The **micro:bit** is a very small computer
- In this course, we'll be learning how to use it, in conjunction with your own devices, to perform some simple robotics type tasks
- You won't be getting robots to move around (yet) but stay tuned for upcoming courses which turn the **micro:bit** into a little, moveable, robot!



# The micro:bit

- Let's see how the **micro:bit** compares to your computer:

	Input	Output	Processing	Communication
Your computer	Keyboard Mouse Touch screen Pens	Monitor/Screen Speakers	CPU Graphics cards	Wifi Bluetooth Ethernet
Your Microbit	Buttons Thermometer Accelerometer Magnetometer Touch sensor Light sensor	25 x LED lights Speakers		Bluetooth Radio



# The micro:bit

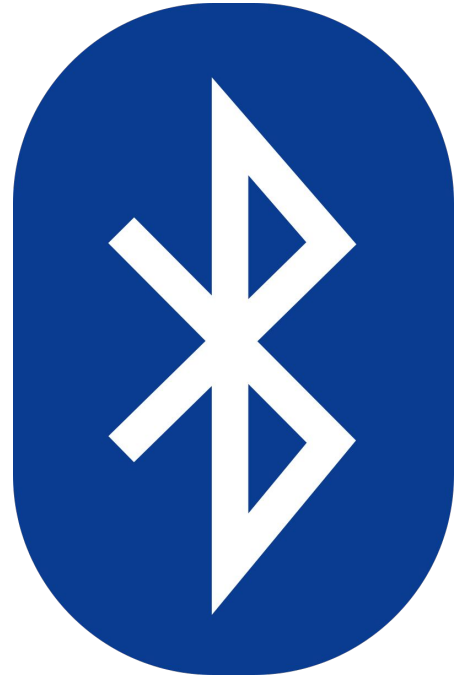
- In this course, we'll be using the following features of the **micro:bit**:

	Input	Output	Processing	Communication
Your computer	Keyboard Mouse Touch screen Pens	Monitor/Screen Speakers	CPU Graphics cards	Wifi Bluetooth Ethernet
Your Microbit	<b>Buttons</b> <b>Thermometer</b> <b>Accelerometer</b> <b>Magnetometer</b> Touch sensor Light sensor	<b>25 x LED lights</b> Speakers		<b>Bluetooth</b> Radio



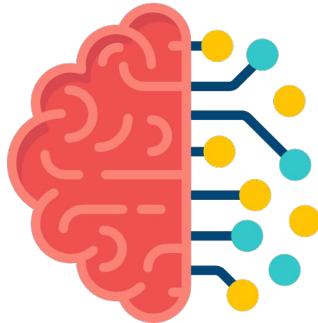
# Bluetooth

- In order for our computer to communicate wirelessly with our micro:bit, we are going to use a technology called **Bluetooth**
- You may already be familiar with **Bluetooth** as it is very commonly used in phones, speakers, LED lights, etc.

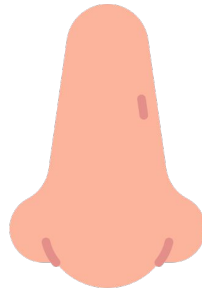
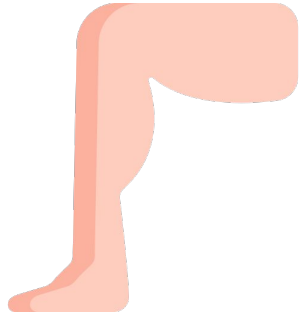


# Bluetooth

- Using **Bluetooth** and an online Python editor, we will allow our computer to talk to multiple micro:bits and use their inputs and outputs
- Our computer, which runs the Python editor, will be the **brain** and the micro:bit will be the **senses** and **outputs**



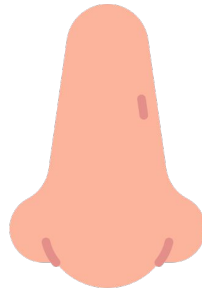
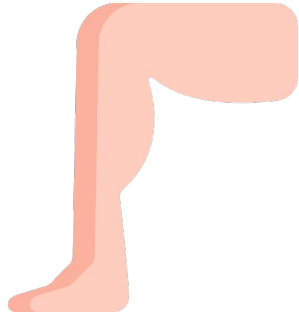
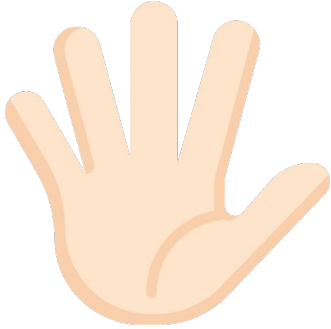
# Bluetooth



Our computer, which runs the Python editor, will be the **brain** and the micro:bit will be the **senses** and **outputs**



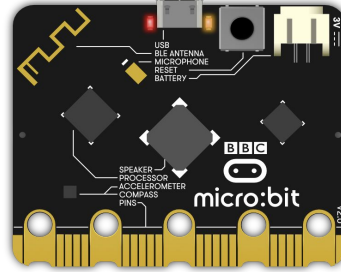
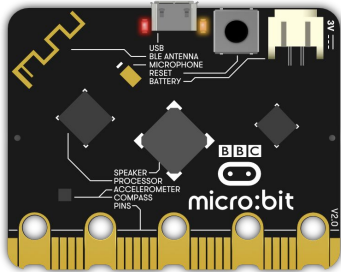
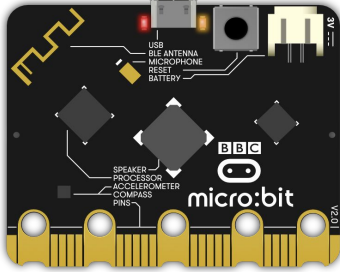
# Bluetooth



Our computer, which runs the Python editor, will be the **brain** and the micro:bit will be the **senses** and **outputs**



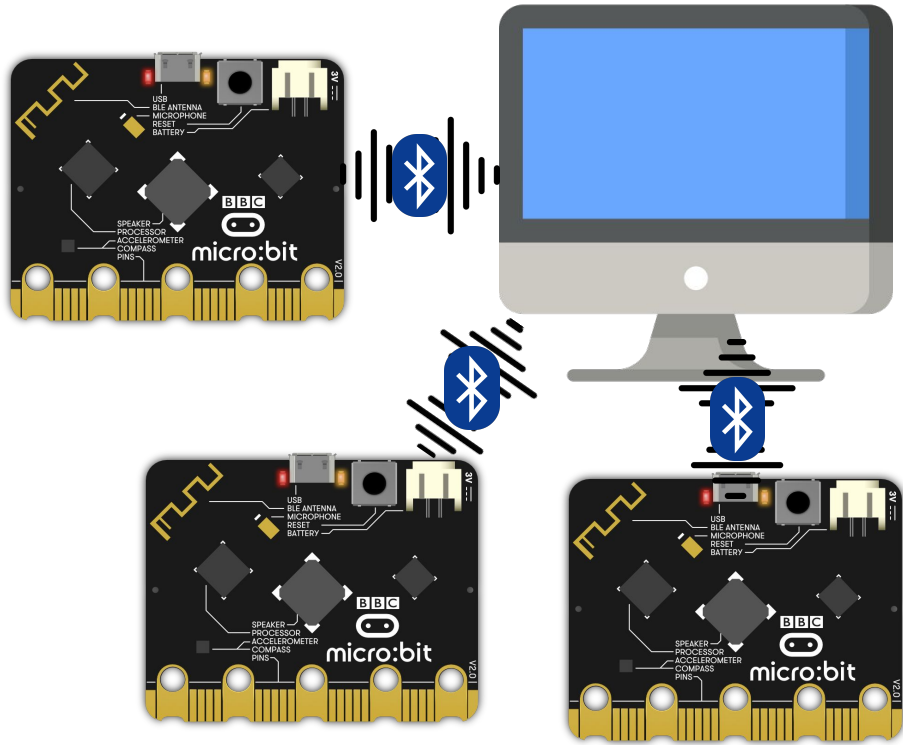
# Bluetooth



Our computer, which runs the Python editor, will be the **brain** and the micro:bit will be the **senses** and **outputs**



# Bluetooth

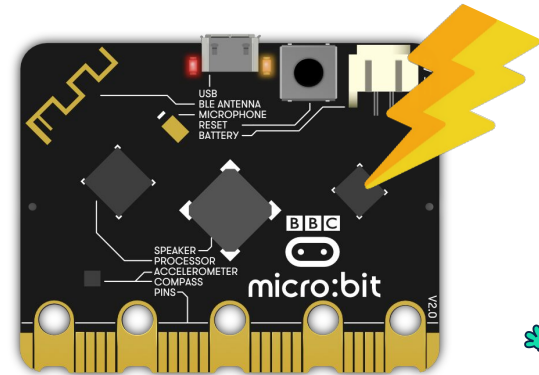


Remember, **Bluetooth** is the wireless connection!



# Getting Started: Flashing your micro:bit

- To get started, we need to install a program on the **micro:bit** so it knows what to do when it is turned on
- We've written a program for you to do this, all you need to do is copy it over to your **micro:bit**
- Copying a program over to the Microbit is called **flashing** the Microbit.



# Getting Started: Flashing your micro:bit

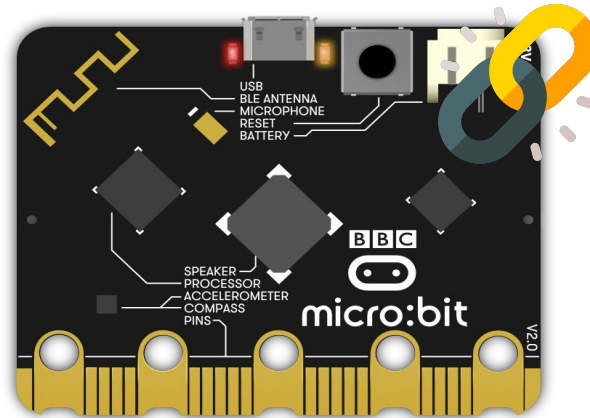


## [Flashing Your Microbit](#)



# Demo: First Signs of Life

Do you remember what the first program you wrote in Python was?



[First Signs of Life](#)

# The Microbit module

Remember the use of **modules** and the **import** keyword?

We use them to refer to other code which we can use in our own program.

In the previous course we used the **csinsc** module for colours.

```
from microbit import *
```

```
m1 = Microbit()
```

```
m1.setText("Hello, world!")
```



# The Microbit module

In this course, in order to connect to and use the micro:bit, we need to import the `microbit` module.

```
from microbit import *
```

```
m1 = Microbit()
```

```
m1.setText("Hello, world!")
```



# Connecting to a micro:bit

Once we have imported the `microbit` module, we are going to start by connecting to our microbit.

To do this, we use the `Microbit()` function, which will prompt us to select our bluetooth device.

```
from microbit import *
```

```
m1 = Microbit()
```

```
m1.setText("Hello, world!")
```



# Connecting to a micro:bit

Once we have imported the `microbit` module, we are going to start by connecting to our microbit.

To do this, we use the `Microbit()` function, which will prompt us to select our bluetooth device.

```
from microbit import *
```

```
m1 = Microbit()
```

```
m1.setText("Hello, world!")
```



# Connecting to a micro:bit

Once we have imported the `microbit` module, we are going to start by connecting to our microbit.

To do this, we use the `Microbit()` function, which will prompt us to select our bluetooth device.

```
from microbit import *
```

Remember the brackets!

```
m1 = Microbit()
```

```
m1.setText("Hello, world!")
```



# Connecting to a micro:bit

Once the user has selected the device via the browser, the program will then assign the **micro:bit** device to a variable.

```
from microbit import *
```

```
m1 = Microbit()
```

```
m1.setText("Hello, world!")
```



# Connecting to a micro:bit

Once the user has selected the device via the browser, the program will then **assign** the **micro:bit** device to a variable.

```
from microbit import *
```

```
m1 = Microbit()
```

```
m1.setText("Hello, world!")
```



# Connecting to a micro:bit

Once the user has selected the device via the browser, the program will then assign the **micro:bit** device to a **variable**.

```
from microbit import *
```

```
m1 = Microbit()
```

```
m1.setText("Hello, world!")
```



# Connecting to a micro:bit

Once the user has selected the device via the browser, the program will then assign the **micro:bit** device to a variable.

In this case, our variable name is **m1**.

Our micro:bits nickname is now **m1**.

You can give your micro:bit any nickname you like; **R2-D2**, **C-3P0**, **Greg**, **Squirtle**.

```
from microbit import *
```

```
m1 = Microbit()
```

```
m1.setText("Hello, world!")
```



# Connecting to a micro:bit

At this point, we will have connected to our micro:bit, but we haven't asked it to do anything for us yet...

```
from microbit import *
```

```
m1 = Microbit()
```

```
m1.setText("Hello, world!")
```



# Displaying Text on the LED

We're now ready to display the famous text, "Hello, world!" on our micro:bit.

To do this, use the `setText()` method that we saw in the demo.

```
from microbit import *
```

```
m1 = Microbit()
```

```
m1.setText("Hello, world!")
```



# Displaying Text on the LED

We're now ready to display the famous text, "Hello, world!" on our micro:bit.

To do this, use the `setText()` method that we saw in the demo.

```
from microbit import *
```

```
m1 = Microbit()
```

```
m1.setText("Hello, world!")
```



# Displaying Text on the LED

We're now ready to display the famous text, "Hello, world!" on our micro:bit.

To do this, use the `setText()` method that we saw in the demo.

We write `m1 .` at the start because we are setting text on the `m1` microbit.

```
from microbit import *
```

```
m1 = Microbit()
```

```
m1.setText("Hello, world!")
```



# Displaying Text on the LED

We're now ready to display the famous text, "Hello, world!" on our micro:bit.

To do this, use the `setText()` method that we saw in the demo.

If we had called our micro:bit `gareth`, we would write `gareth` at the start instead!

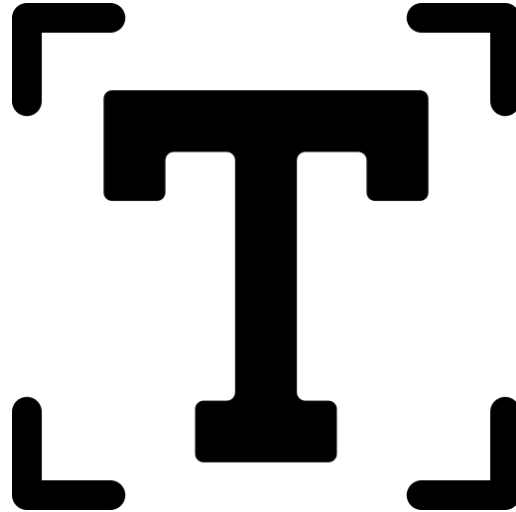
```
from microbit import *  
  
gareth = Microbit()  
  
gareth.setText("Hello, world!")
```





**Activity Time!**

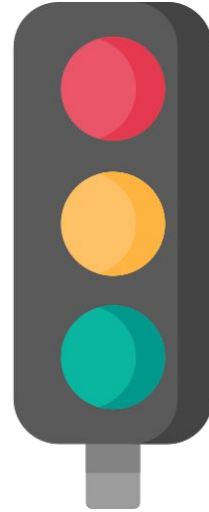
# Exercise 1: Displaying Text



[Activity 01.01](#)



## Exercise 2: Ready, setLED(), Code!



[Activity 01.02](#)



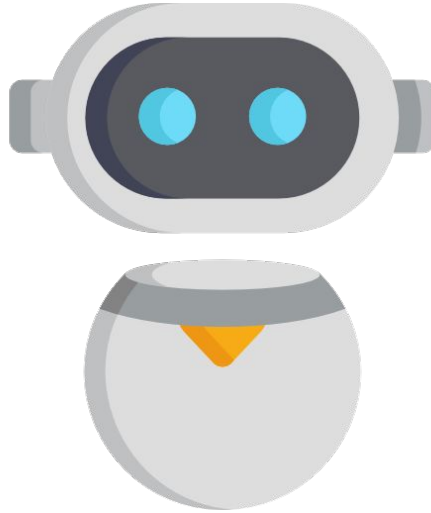
## Exercise 3: clear() your Mind, use sleep()!



[Activity 01.03](#)



# Exercise 4: Robotic Animation



[Activity 01.04](#)



# Summary

You should now be able to:

- Flash your **micro:bit**
- Connect to your **micro:bit** using **Bluetooth**
- Get your **micro:bit** to display text on the LED display using `.setText(...)`
- Turn on individual LEDs using `.setLED(...)`
- Use `sleep(...)` and `.clear()` to create an animation



# License Information

These [CS in Schools](#) lessons plans, worksheets, and other materials were created by the CS in Schools team. They are licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](#).

Images all taken from [Flaticon](#).

