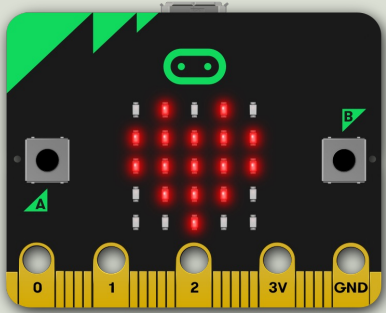


<https://www.halvorsen.blog>



# micro:bit

Hans-Petter Halvorsen

# Contents

- [Introduction to micro:bit](#)
- [micro:bit and Python/MicroPython](#)
- [Python Editors](#)
- [Online Python Editor](#)
- [Mu Python Editor](#)
- [Python Examples](#)

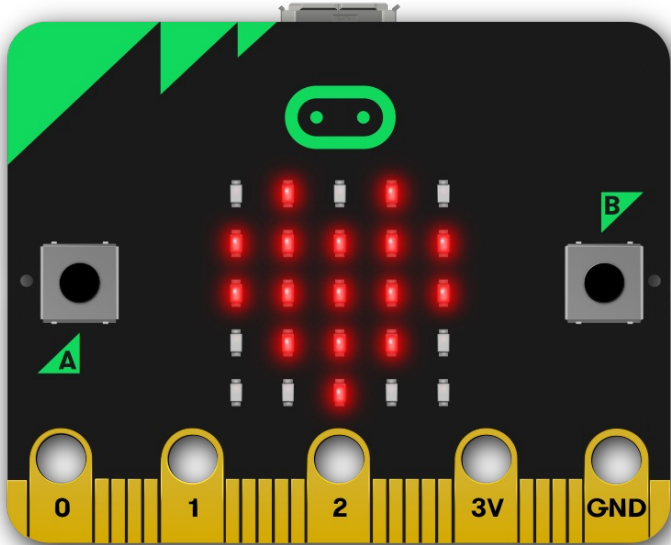


# Introduction to micro:bit

Hans-Petter Halvorsen

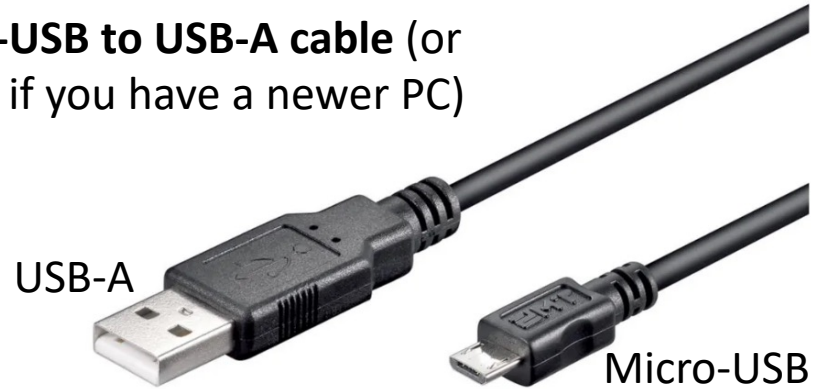
[Table of Contents](#)

# What do you need?



micro:bit

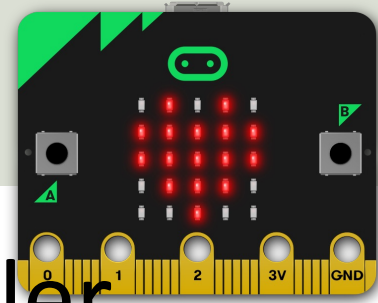
**Micro-USB to USB-A cable** (or  
USB-C if you have a newer PC)



A PC

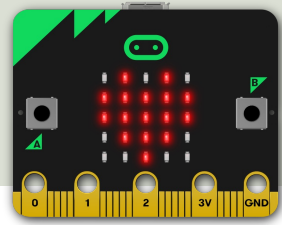


# micro:bit



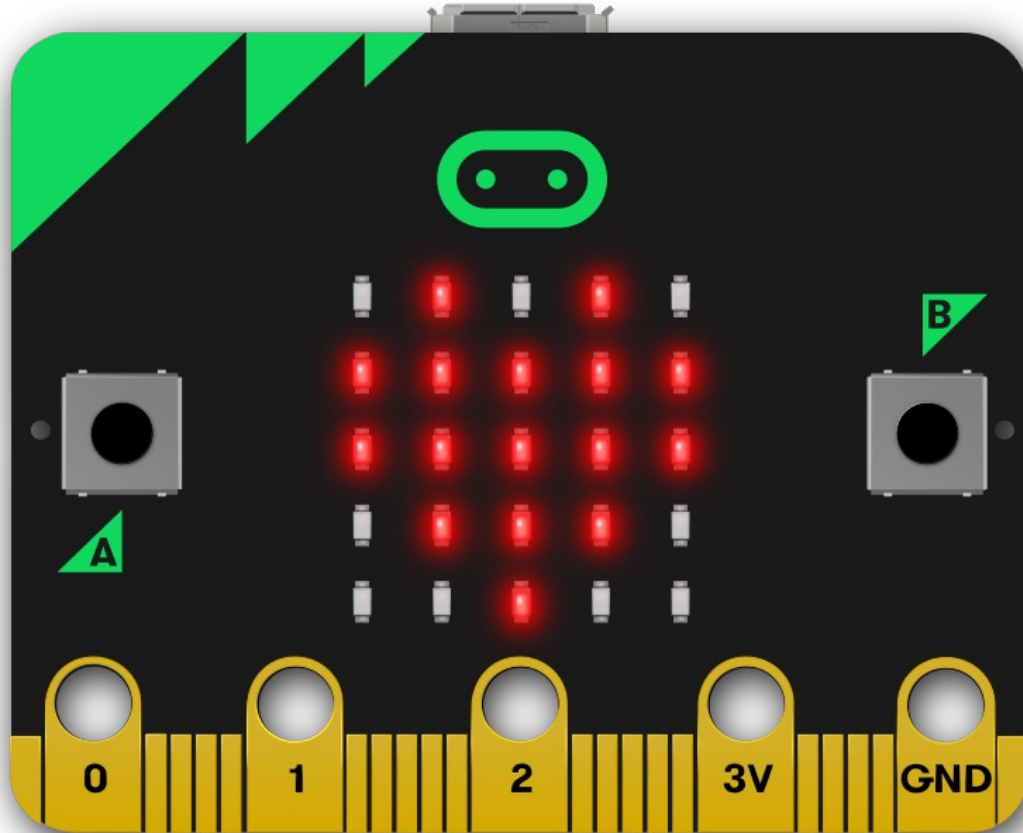
- micro:bit is a small microcontroller
- micro:bit is smaller than a credit card
- Price is about 150-400NOK (\$15-30)
- It can be used by kids and students to learn programming and technology
- micro:bit has Bluetooth but not WiFi

# micro:bit and Programming

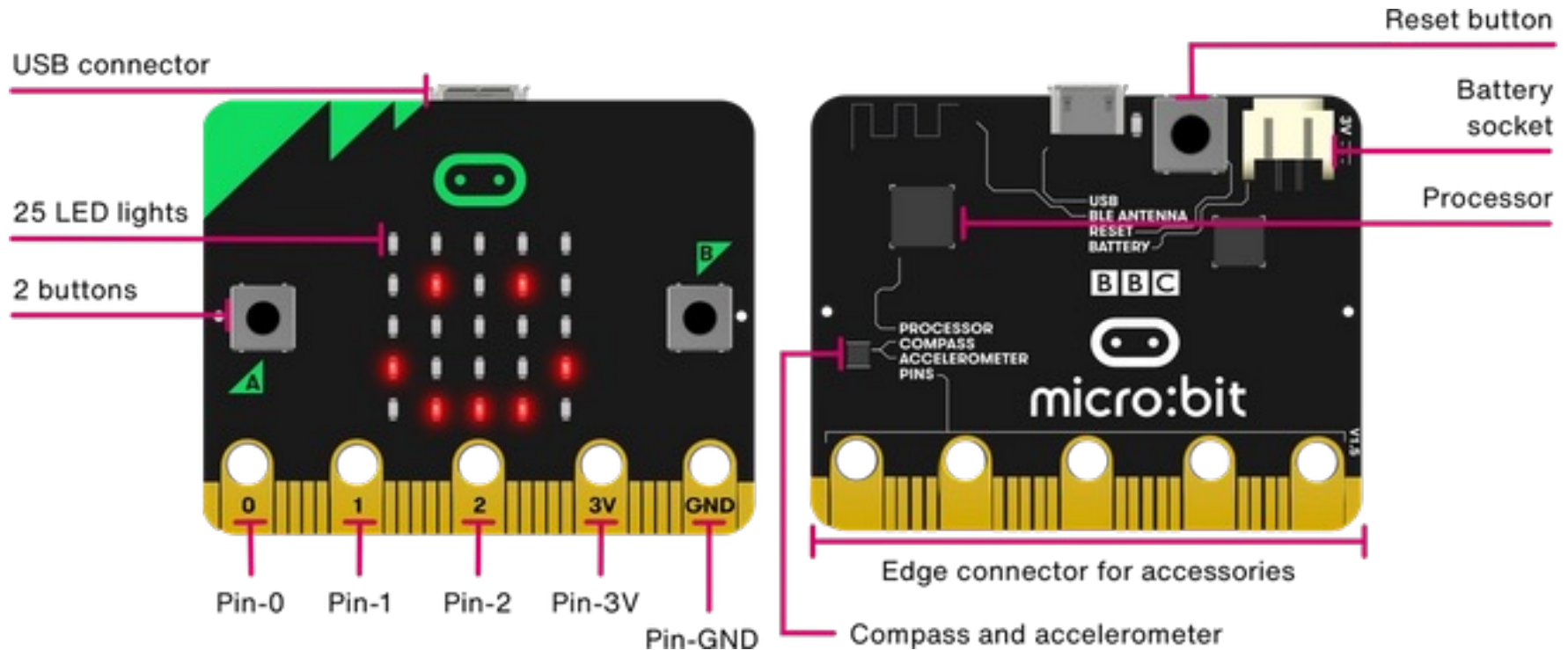


- micro:bit can run a special version of Python called **MicroPython**
- MicroPython is a down-scaled version of Python
- You can use different code editors and Programming Languages
  - Scratch, Microsoft MakeCode, Python, Swift Playground, etc.
- This Tutorial will use Python/MicroPython

# micro:bit

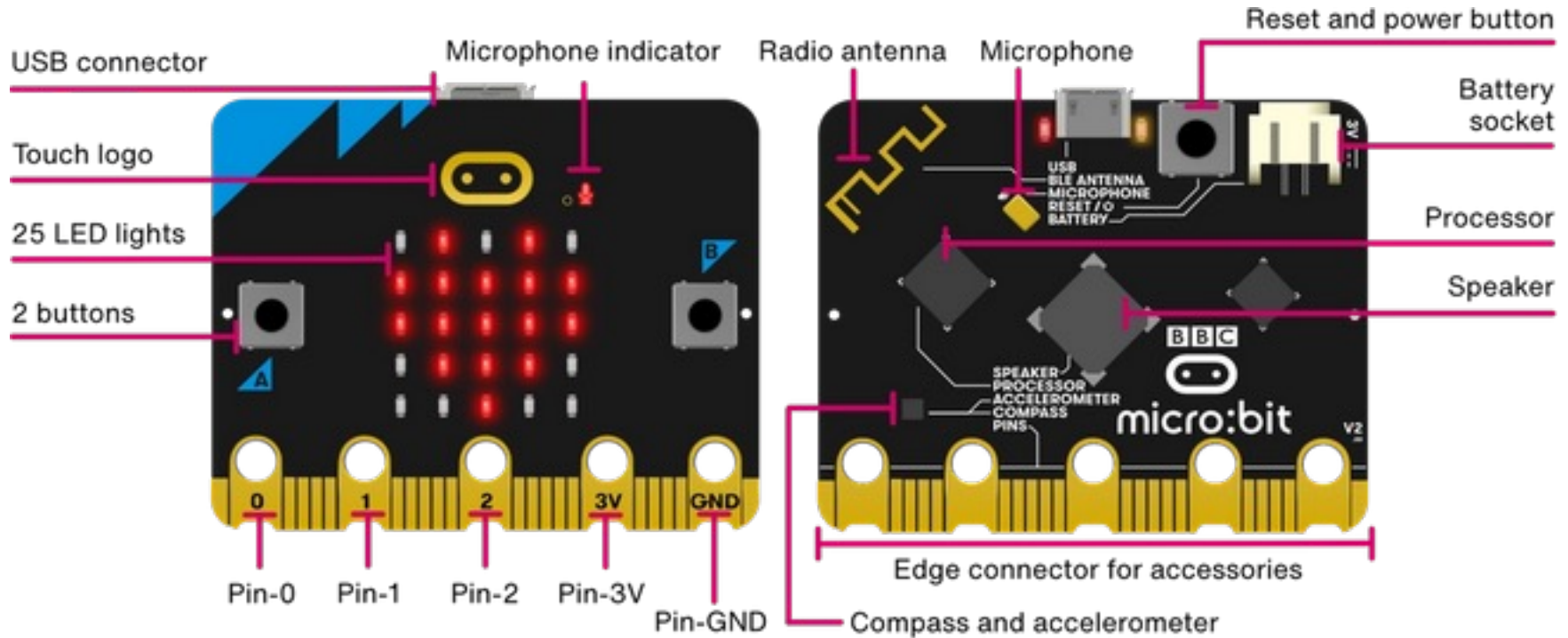


# Original micro:bit





# New micro:bit (micro:bit v2)



# micro:bit Features

- USB Communication and Powered by micro-USB or JST Battery Connection
- Sensors: Motion, Temperature, Light, Magnetism, Microphone and Touch
- Push Buttons
- Lots of Analog/Digital Input/Output Pins
- Speaker
- Wireless Radio Communication
- Bluetooth Communication
- SPI, I2C and UART
- Pulse Width Modulation (PWM)

**Micro USB**

Front

**Touch sensitive logo**

**LED matrix 5x5**

**Microphone**

- LED indicator
- Hole for microphone input

**User buttons**

**Analogue/Digital I/O**

- Muxable to SPI, UART, I2C
- Notched pads for crocodile clips
- Holes for banana plugs

**External supply**

- Regulated 3.3V in or battery out

**Edge Connector**

**Power indicator**

**USB activity indicator**

Back

**Battery connector**

- JST connection for 3V

**Reset/power button**

**NXP KL27Z**

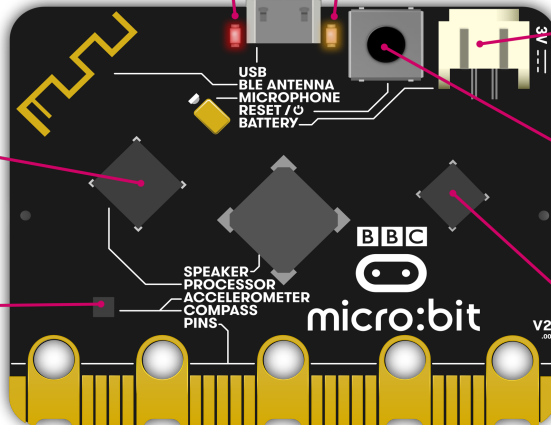
- USB interface chip

**CPU**

**Nordic nRF52833**

**Motion sensor**

**ST LSM303AGR**





# micro:bit and Python

Hans-Petter Halvorsen

[Table of Contents](#)

# Python

- Python is a fairly old Programming Language (1991) compared to many other Programming Languages like C# (2000), Swift (2014), Java (1995), PHP (1995).
- Python has during the last 10 years become more and more popular.
- Today, Python has become one of the most popular Programming Languages.

# micro:bit and Python

- The combination of the micro:bit Hardware and the Python Programming Language is very powerful
- micro:bit runs a special version of MicroPython
- MicroPython is a down-scaled version of Python
- You can use different Python Editors; e.g., Mu Python Editor or the Online Python Editor, etc.

# MicroPython

- MicroPython is a small and efficient implementation of the Python 3 programming language
- MicroPython includes a small subset of the Python standard library
- MicroPython is optimized to run on microcontrollers and in constrained environments
- <https://microbit-micropython.readthedocs.io/>

# micro:bit Python Documentation

- micro:bit Python User Guide

<https://microbit.org/get-started/user-guide/python/>

- micro:bit MicroPython documentation

<https://microbit-micropython.readthedocs.io>





# Python Editors

Hans-Petter Halvorsen

[Table of Contents](#)

# micro:bit Python Editors

Here are some Editors:

- Online Editor (used in your Browser)

<https://python.microbit.org>

- Mu Python Editor

<https://codewith.mu>

This Tutorial will mainly use the Mu Python Editor



# Online Python Editor

Hans-Petter Halvorsen

[Table of Contents](#)

# Online Python Editor

The screenshot displays the Python Editor for micro:bit web application. The interface is divided into three main sections: a left sidebar, a central code editor, and a right-hand control panel.

**Left Sidebar (micro:bit Reference):**

- Variables:** Keep track of data that...
- Display:** The micro:bit's LED display...
- Buttons:** Use button inputs in your...
- Loops:** Count and repeat sets of...
- Logic:** Making decisions in code
- Accelerometer:** Detect gestures and...
- Comments:** Explain your Python code
- Maths:** Basic maths in Python

**Central Code Editor:**

The editor shows a file named "Untitled project" with the following Python code:

```
1 # Imports go at the top
2 from microbit import *
3
4
5 # Code in a 'while True:' loop repeats forever
6 while True:
7     display.show(Image.HEART)
8     sleep(1000)
9     display.scroll('Hello')
10
```

**Right-hand Control Panel:**

- Show serial:** A dropdown menu set to "shake" and a play button.
- Sliders:** A series of horizontal sliders for controlling various micro:bit features.
- Buttons:** A row of buttons labeled A and B.
- Radio message:** A section for sending and receiving radio messages.
- No rows logged:** A status indicator for the serial log.

**Bottom Bar:**

- Send to micro:bit:** A button to upload the code to the micro:bit.
- Save:** A button to save the project.
- Open...:** A button to open an existing project.



# Mu Python Editor

Hans-Petter Halvorsen

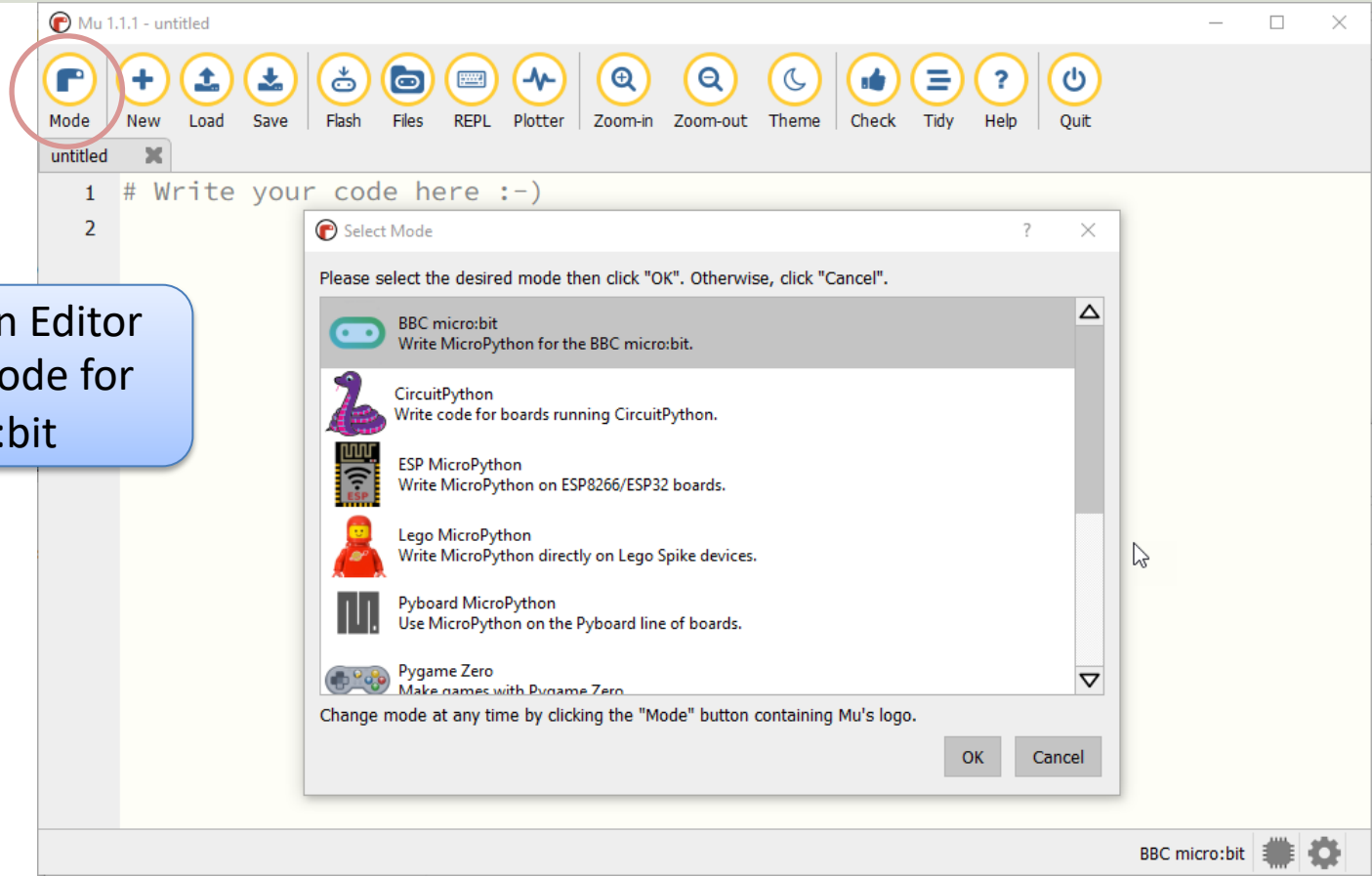
[Table of Contents](#)

# Mu Python Editor

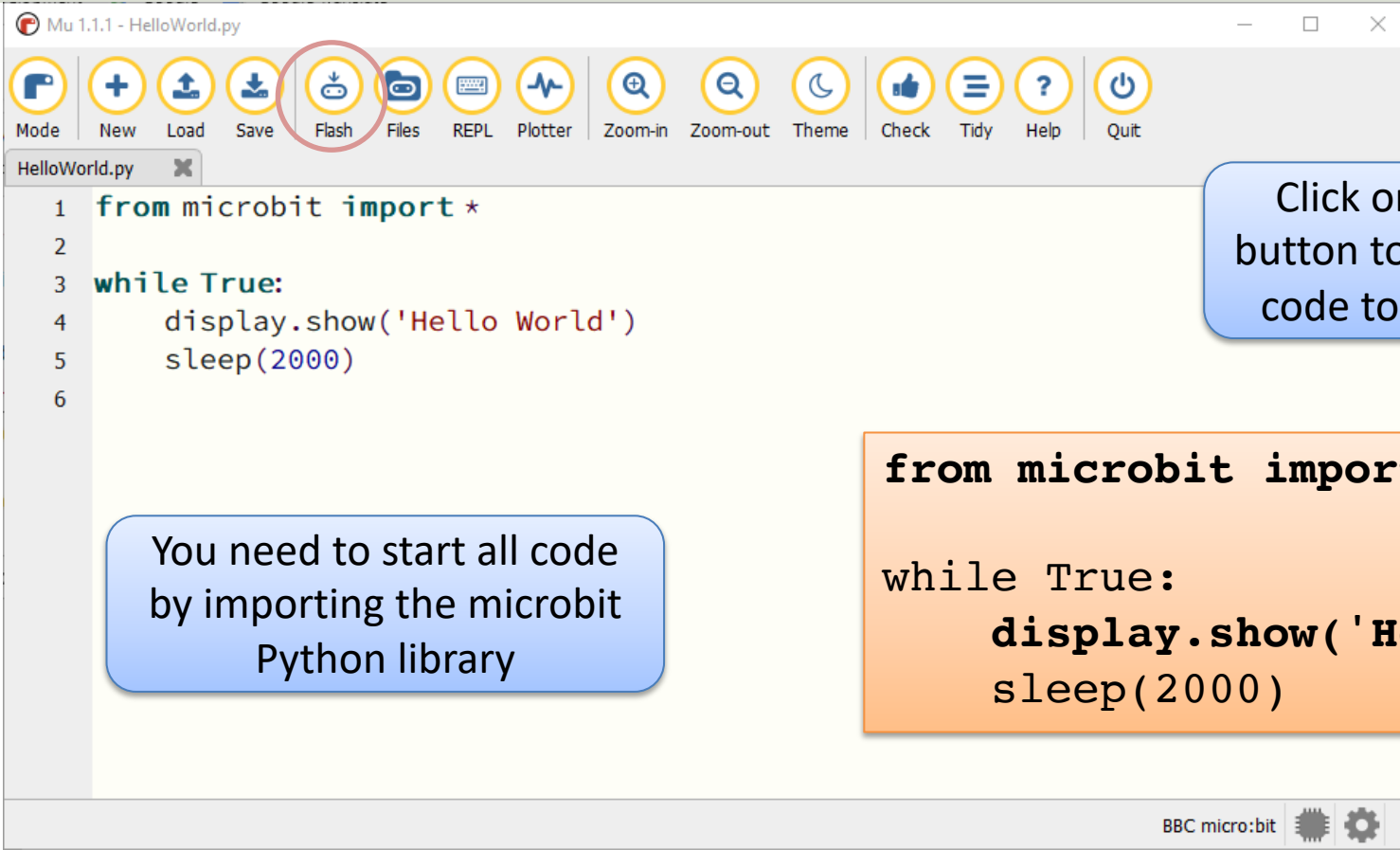
- Mu is a Python code editor for beginners
- It is tailor-made for micro:bit programming
- Mu has a “micro:bit mode” that makes it easy to work with micro:bit, download code to the micro:bit hardware, etc.
- Mu and micro:bit Tutorials:  
<https://codewith.mu/en/tutorials/1.0/microbit>

# Mu Python Editor

The Mu Python Editor  
has built-in Mode for  
the micro:bit



# Hello World



The screenshot shows the Mu Python IDE window titled 'Mu 1.1.1 - HelloWorld.py'. The toolbar contains various icons for file operations and development. The 'Flash' icon, which represents uploading code to a micro:bit, is circled in red. Below the toolbar, the code editor displays the following Python code:

```
1 from microbit import *
2
3 while True:
4     display.show('Hello World')
5     sleep(2000)
6
```

At the bottom right of the IDE window, there is a status bar with the text 'BBC micro:bit' and two icons: a micro:bit board and a settings gear.

Click on the “Flash”  
button to download the  
code to the micro:bit

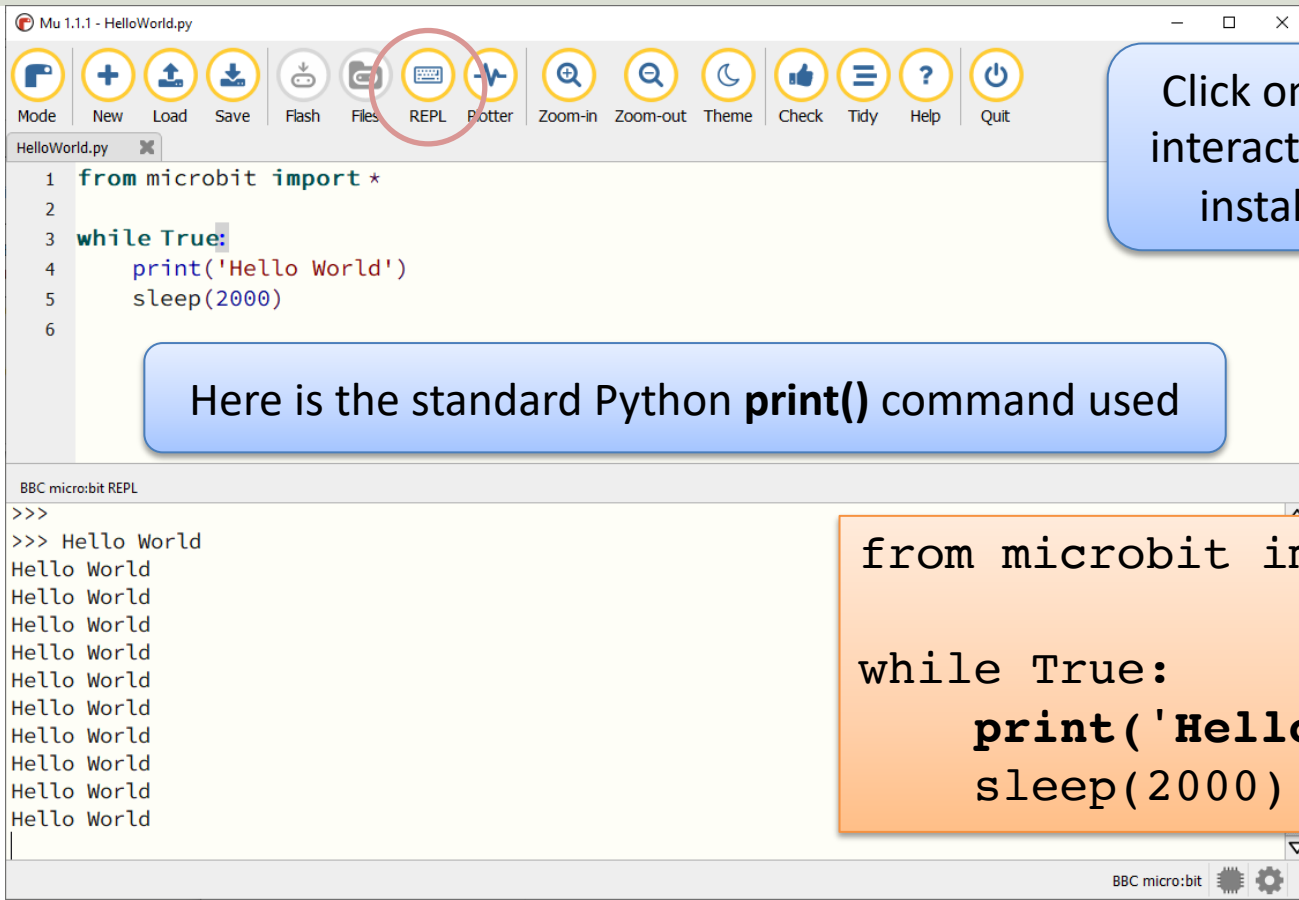
You need to start all code  
by importing the microbit  
Python library

```
from microbit import *

while True:
    display.show('Hello World')
    sleep(2000)
```



# REPL



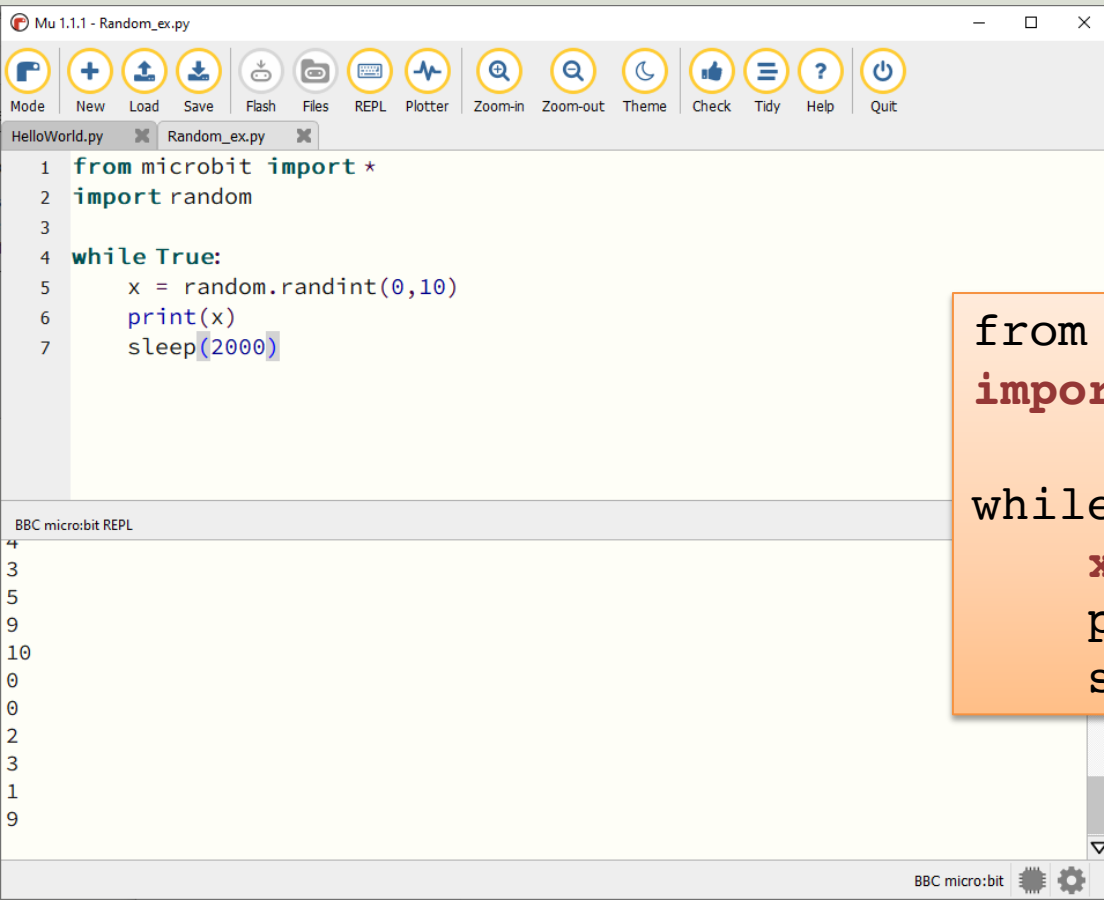
Click on the “REPL” button to interact with the MicroPython installed on the micro:bit

Here is the standard Python **print()** command used

```
from microbit import *

while True:
    print('Hello World')
    sleep(2000)
```

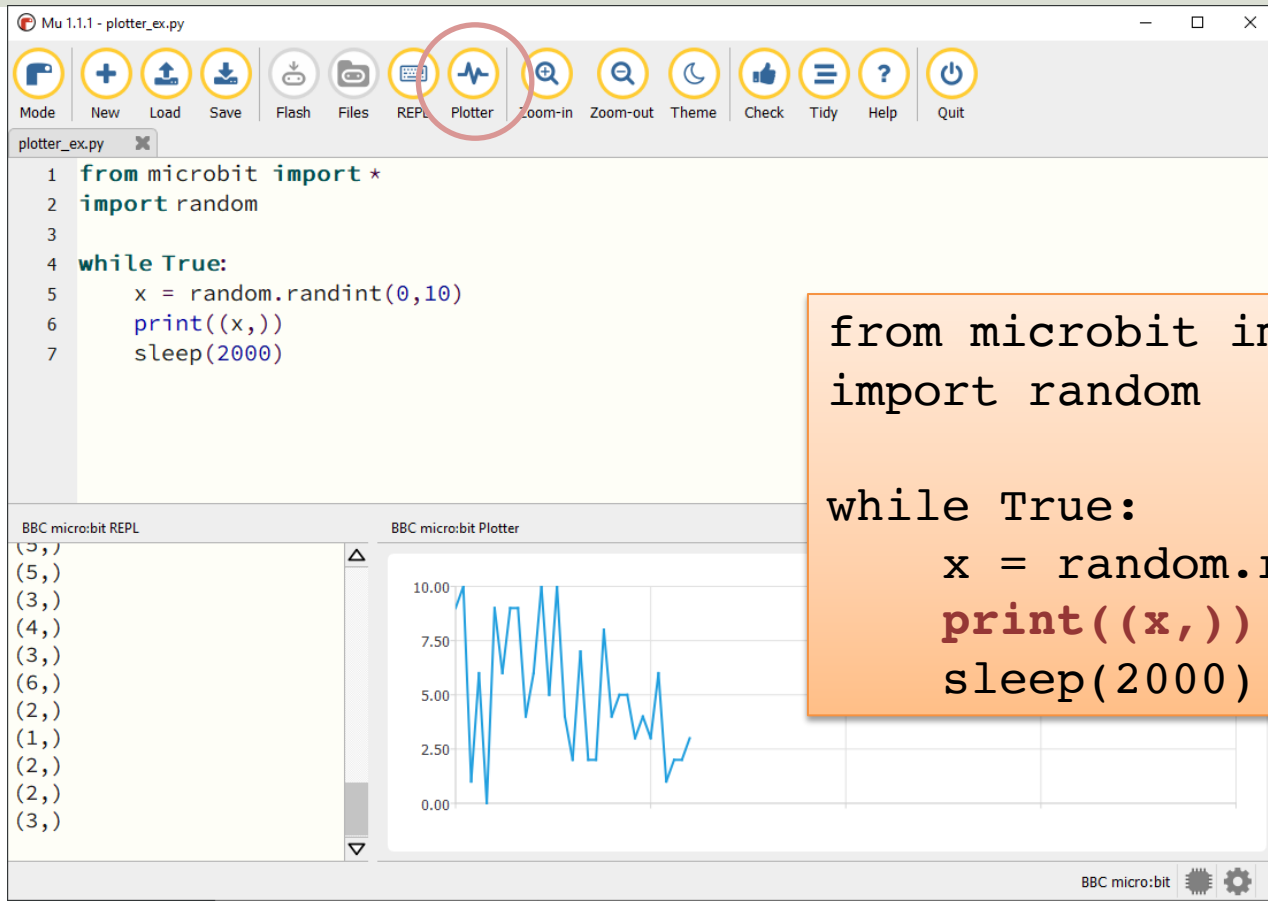
# REPL – Random Numbers



```
from microbit import *
import random

while True:
    x = random.randint(0,10)
    print(x)
    sleep(2000)
```

# Plotter



The screenshot shows the Mu Python IDE interface. The top toolbar contains icons for Mode, New, Load, Save, Flash, Files, REPL, Plotter (highlighted with a red circle), Zoom-in, Zoom-out, Theme, Check, Tidy, Help, and Quit. The main editor displays a Python script for a BBC micro:bit. The bottom panel is split into two sections: 'BBC micro:bit REPL' on the left and 'BBC micro:bit Plotter' on the right. The REPL shows a series of coordinate pairs (x, y) being printed. The Plotter window displays a line graph of these coordinates, showing a jagged, oscillating pattern. The status bar at the bottom indicates 'BBC micro:bit' with a gear icon for settings.

```
Mu 1.1.1 - plotter_ex.py
```

Mode New Load Save Flash Files REPL Plotter Zoom-in Zoom-out Theme Check Tidy Help Quit

plotter\_ex.py

```
1 from microbit import *
2 import random
3
4 while True:
5     x = random.randint(0,10)
6     print((x,))
7     sleep(2000)
```

BBC micro:bit REPL

```
(3,)
(5,)
(3,)
(4,)
(3,)
(6,)
(2,)
(1,)
(2,)
(2,)
(3,)
```

BBC micro:bit Plotter

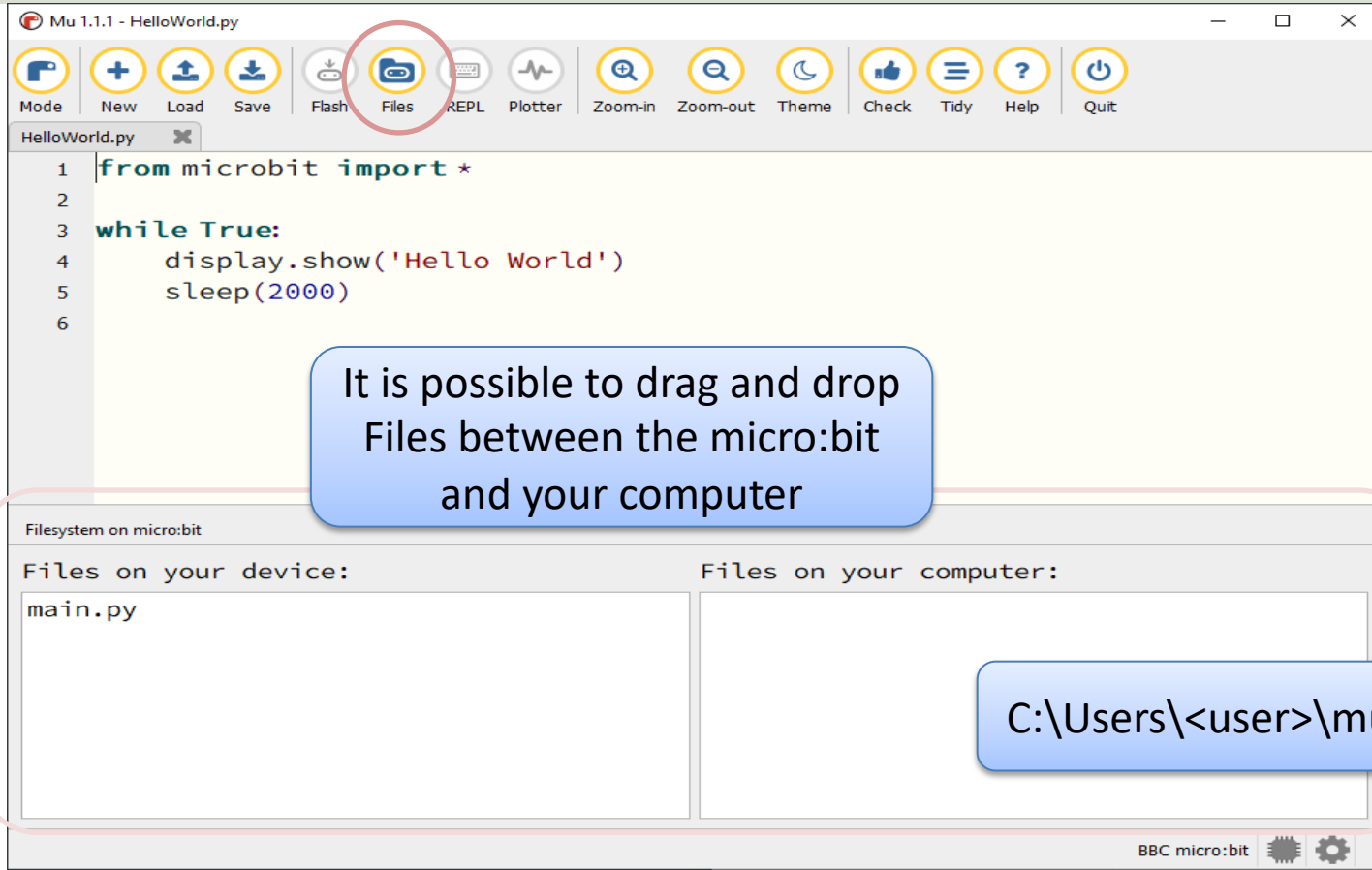
10.00  
7.50  
5.00  
2.50  
0.00

BBC micro:bit

```
from microbit import *
import random

while True:
    x = random.randint(0,10)
    print((x,))
    sleep(2000)
```

# Files





# micro:bit Interfaces

## with Python Examples

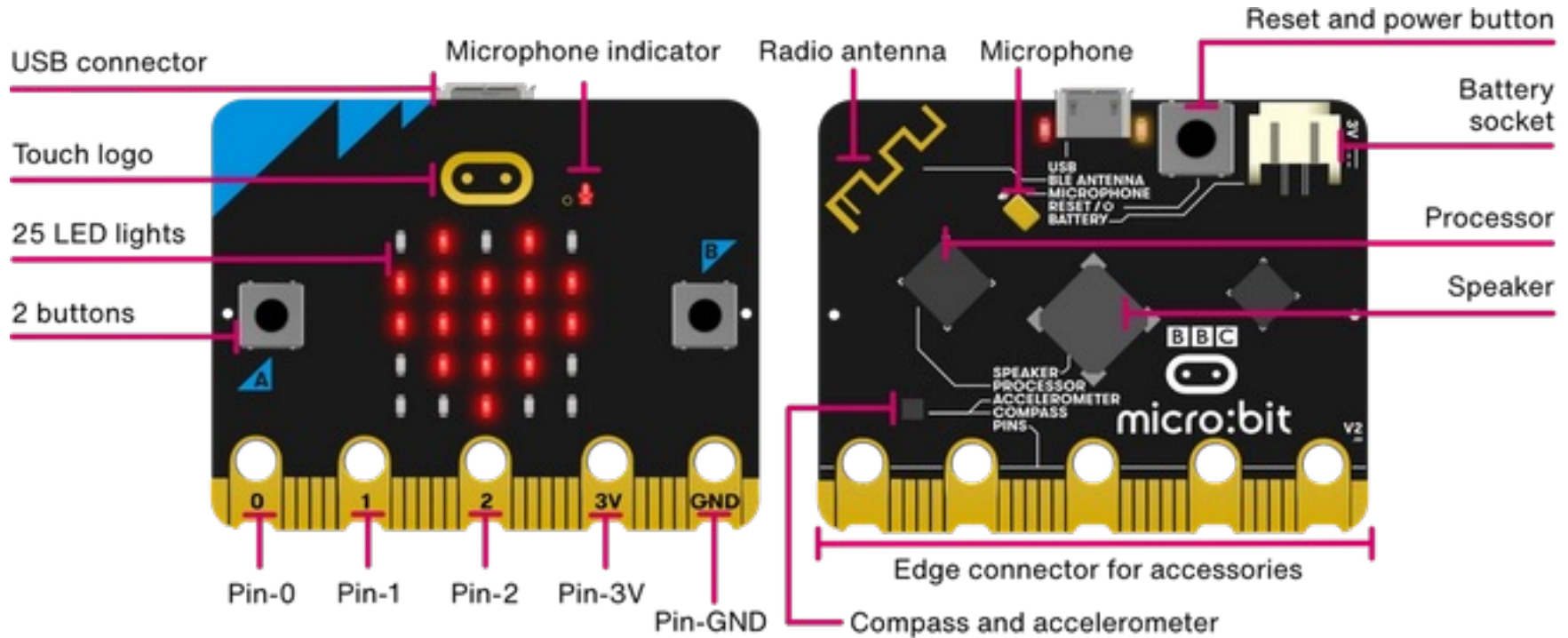
Hans-Petter Halvorsen

[Table of Contents](#)

# micro:bit Interfaces

- **LED Matrix (5x5)**
- **Buttons (A and B)**
- **Temperature Sensor**
- Light Sensor
- Accelerometer
- Compass
- Touch (only available for new micro:bit)
- Microphone (only available for new micro:bit)
- I/O Pins: Analog/Digital Input/Output Pins

# New micro:bit (micro:bit v2)



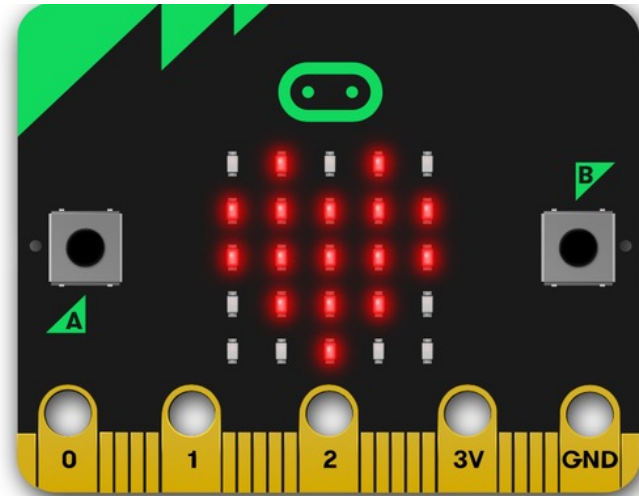


# LED Matrix (5x5)



# LED Matrix (5x5)

- An LED, or light-emitting diode is an output device that gives off light.
- The Micro:bit has a display of 25 (5x5) LEDs for you to program.
- You can use the LED matrix to show images or show text or numbers



# LED Matrix - Text

```
from microbit import *  
  
display.show("WELCOME")
```

This will show one letter at the time on the LED matrix

```
from microbit import *  
while True:  
    display.show("WELCOME")  
    sleep(1000)
```

It will do it "Forever"

```
from microbit import *  
  
display.scroll("WELCOME")
```

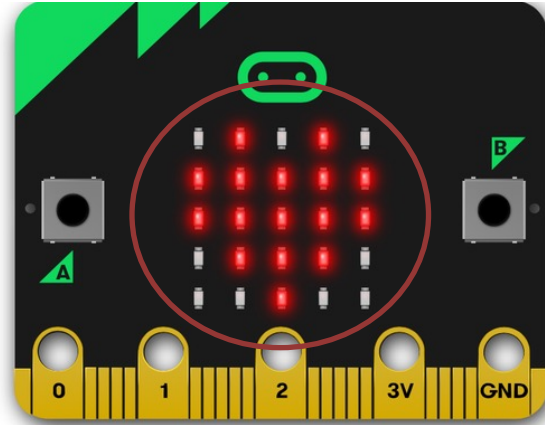
The word "WELCOME" will scroll over the LED matrix

```
from microbit import *  
while True:  
    display.scroll("WELCOME")  
    sleep(1000)
```

# LED Matrix - Images

The micro:bit has a set of other built-in images that you can use

```
from microbit import *  
  
display.show(Image.HEART)
```

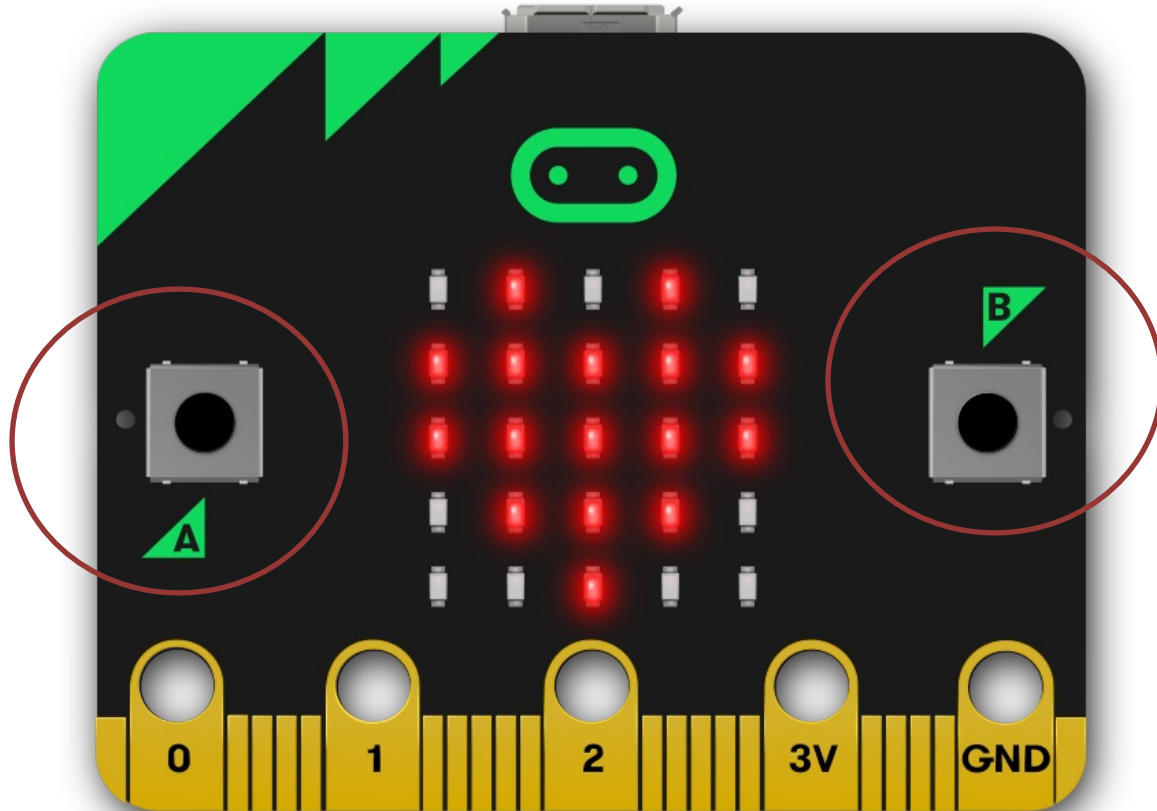


There are almost 100 built-in images that you can use. Just enter “Image.” and the Intellisense will list all available Images that you can use.



# Buttons (A and B)

# Buttons (A and B)



# Buttons (A and B)

```
from microbit import *

while True:
    if button_a.was_pressed():
        display.scroll("A")
        print("A")
    elif button_b.was_pressed():
        display.scroll("B")
        print("B")
    else:
        display.scroll("?")
        print("?")

    sleep(1000)
```



# Temperature Sensor

Hans-Petter Halvorsen

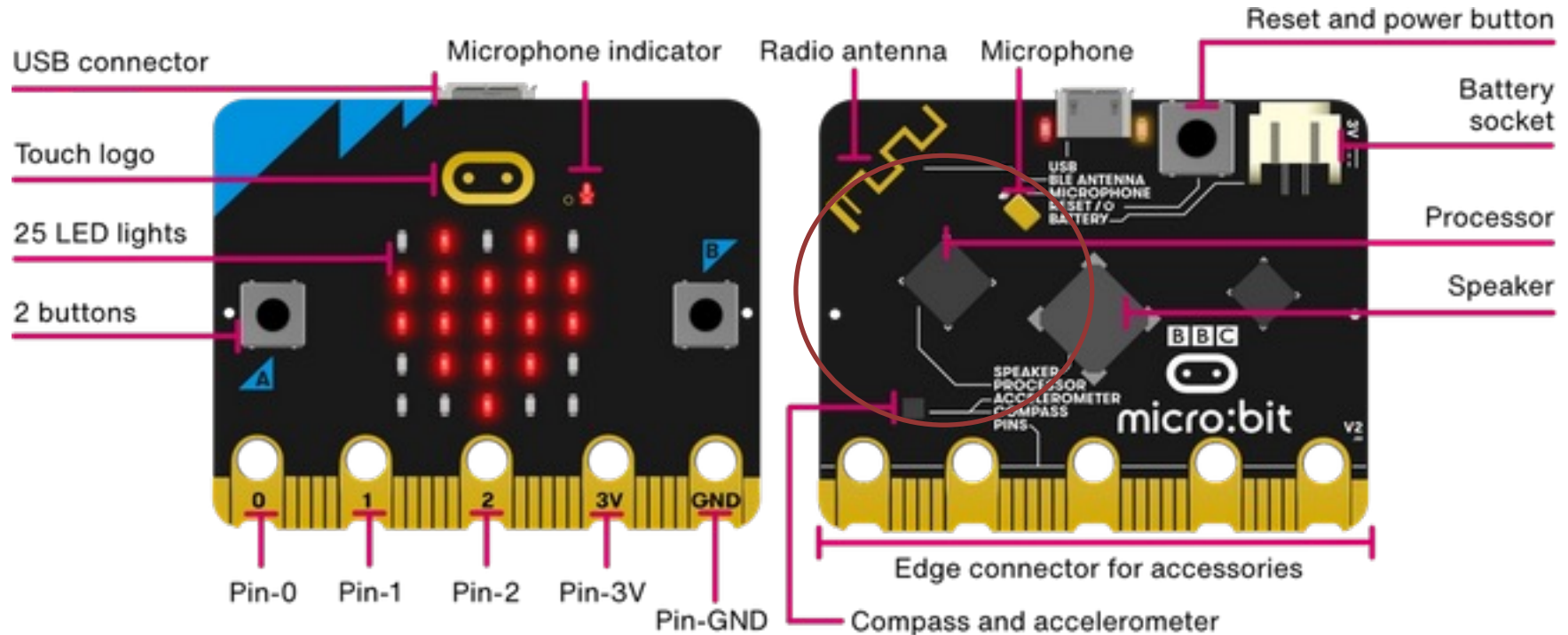
[Table of Contents](#)

# Temperature Sensor

- Micro:bit has a built-in Temperature Sensor (that is located on the CPU)
- This sensor can give an approximation of the air temperature.
- Just use the built-in `temperature()` function in order to get the temperature value from the sensor



# Temperature Sensor



# Temperature Sensor

In order to read the temperature, you just use the built-in `temperature()` function:

```
from microbit import *  
  
currentTemp = temperature()
```

This examples displays the temperature on the LED matrix:

```
from microbit import *  
  
while True:  
    if button_a.was_pressed():  
        display.scroll(temperature())
```

<https://microbit.org/get-started/user-guide/features-in-depth/#temperature-sensor>

# Temperature Sensor

Mu 1.1.1 - temp\_ex.py

Mode New Load Save Flash Files REPL Plotter Zoom-in Zoom-out Theme Check Tidy Help Quit

temp\_ex.py

```
1 from microbit import *
2
3 while True:
4     currentTemp = temperature()
5     print(currentTemp)
6
7     sleep(2000)
```

BBC micro:bit REPL

```
27
27
27
27
27
27
27
27
27
```

BBC micro:bit

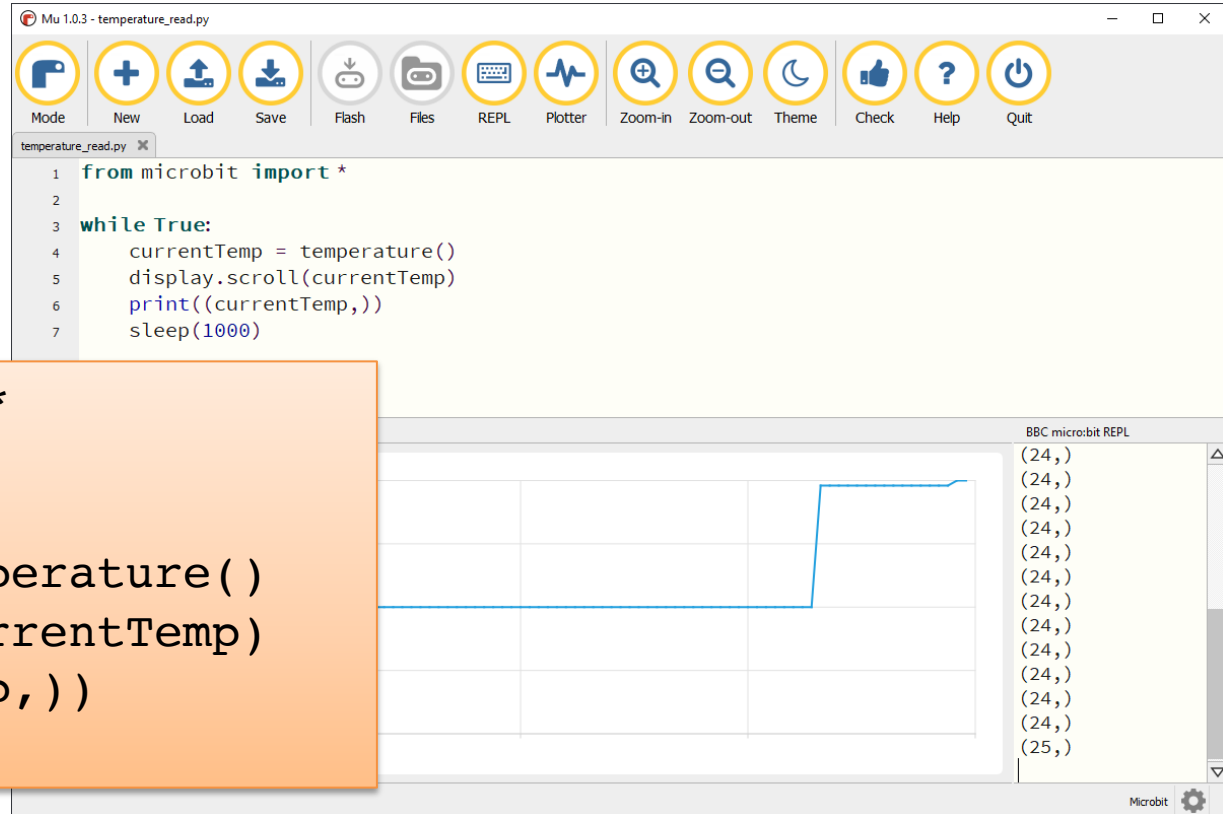
```
from microbit import *
```

```
while True:
```

```
    currentTemp = temperature()
    print(currentTemp)
```

```
    sleep(2000)
```

# Temperature Sensor



```
from microbit import *
```

```
while True:
```

```
    currentTemp = temperature()
```

```
    display.scroll(currentTemp)
```

```
    print((currentTemp,))
```

```
    sleep(1000)
```

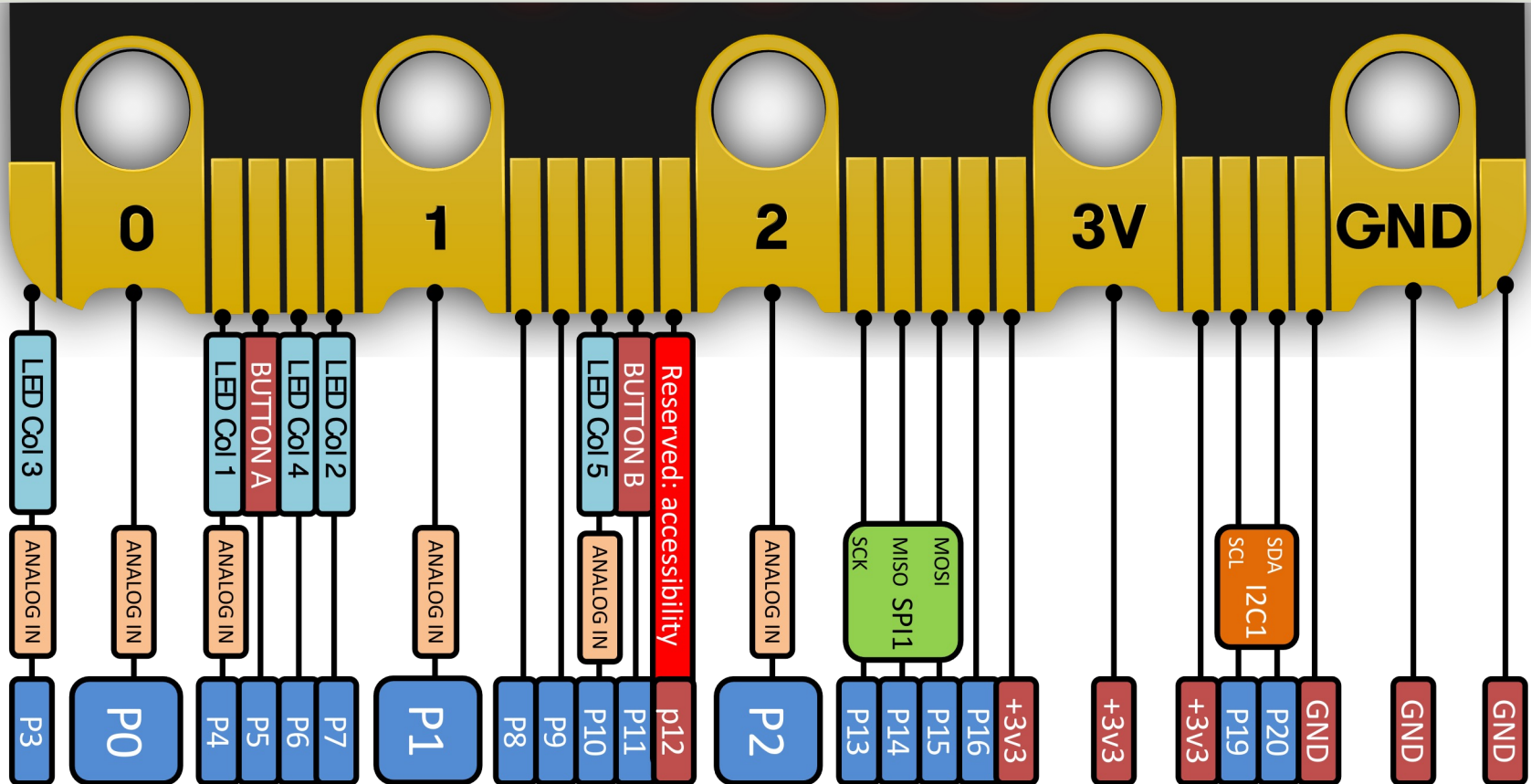


# I/O Pins

# micro:bit Interfaces

- **LED Matrix (5x5)**
- **Buttons (A and B)**
- **Temperature Sensor**
- Light Sensor
- Accelerometer
- Compass
- Touch (only available for new micro:bit)
- Microphone (only available for new micro:bit)
- I/O Pins: Analog/Digital Input/Output Pins

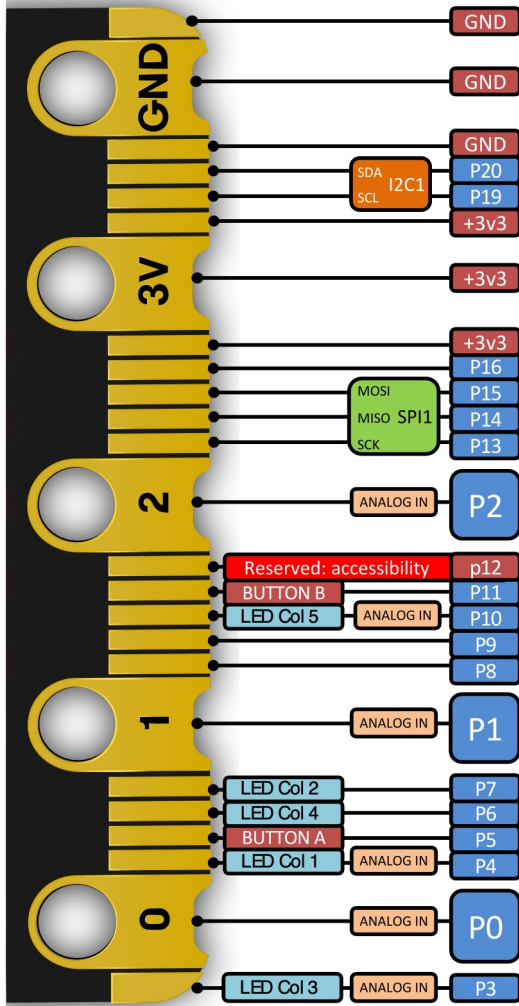
# I/O Pin Overview v.2



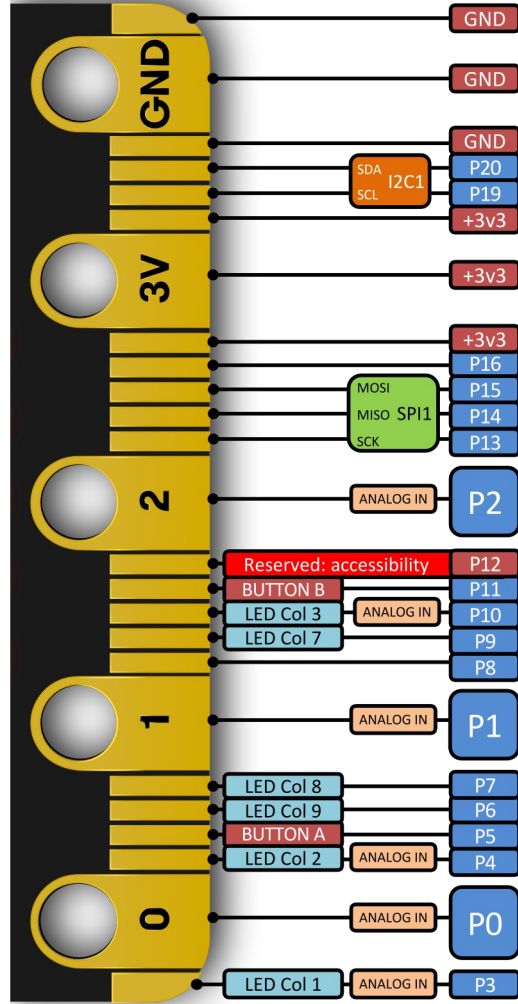
# I/O Pin Overview

<https://microbit.pinout.xyz/>

New micro:bit (micro:bit v2)



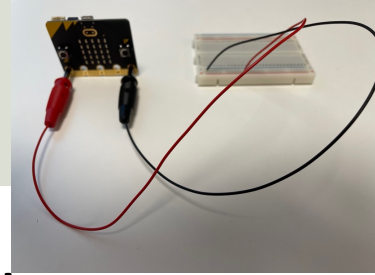
Original micro:bit



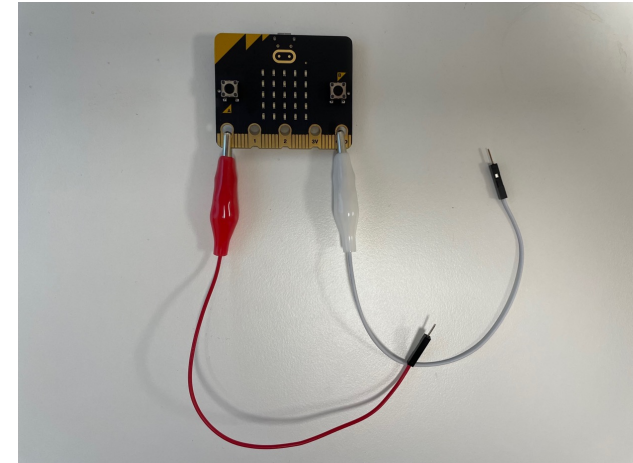
<https://tech.microbit.org/hardware/edgeconnector/>



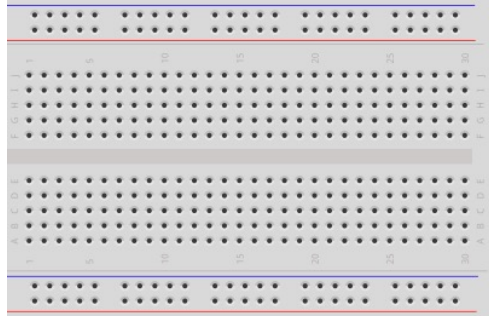
# I/O Pins



- We use the I/O pins to connect external components like LEDs, different types of Sensors, etc.
- You can use 4mm Banana plugs or alligator/crocodile clips
- Typically you also want to use a Breadboard



# Component Examples

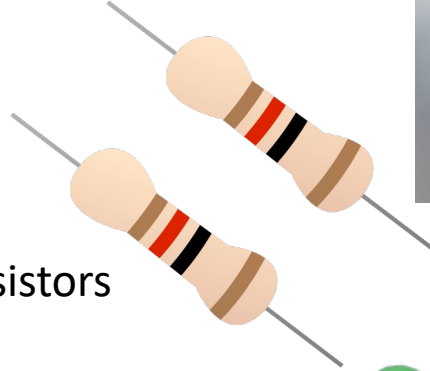


Breadboard

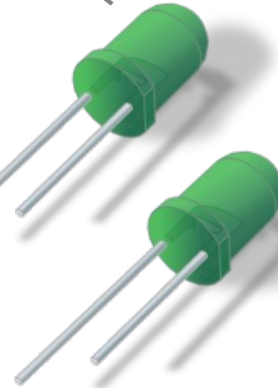
Temperature Sensor



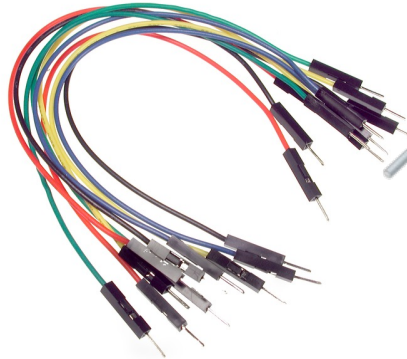
Resistors



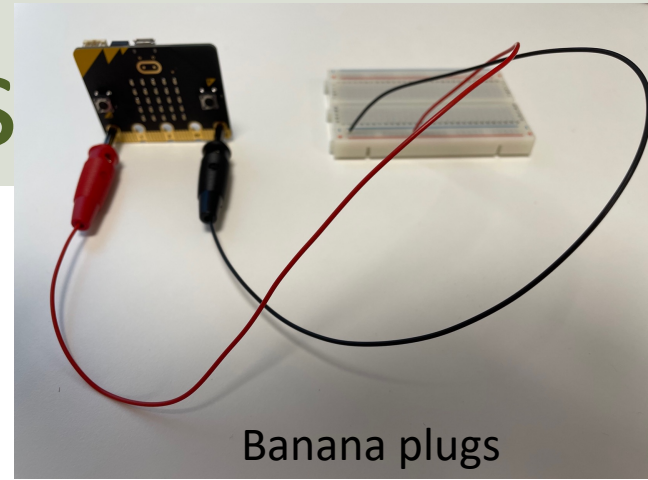
LEDs



Wires



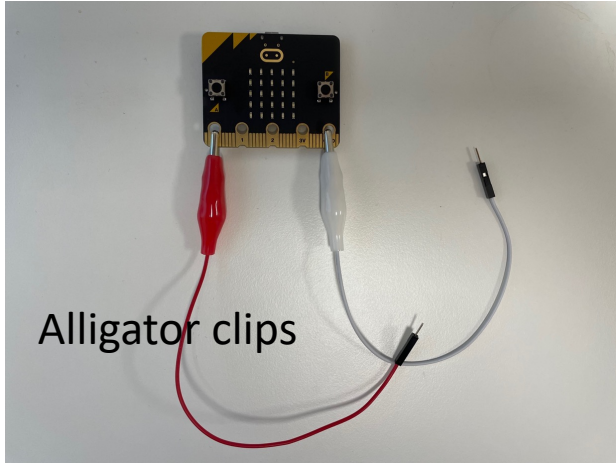
Banana plugs



Multimeter



Alligator clips



# Types of I/O Pins

- Analog/Digital Input/Output Pins
- Pulse Width Modulation (PWM)
- SPI
- I2C
- UART (used for serial communication)

We will not use the I/O pins in this Tutorial, but I will give an overview and give examples in other micro:bit/Python Tutorials that goes in more depth regarding these I/O pins with lots of practical examples

# micro:bit Resources and References

- micro:bit Python User Guide  
<https://microbit.org/get-started/user-guide/python/>
- micro:bit MicroPython documentation  
<https://microbit-micropython.readthedocs.io>
- Learn micro:bit (Adafruit):  
<https://learn.adafruit.com/bbc-micro-bit-lesson-number-0>
- Online Python Editor: <https://python.microbit.org>
- Mu Python Editor: <https://codewith.mu>

# Hans-Petter Halvorsen

University of South-Eastern Norway

[www.usn.no](http://www.usn.no)

E-mail: [hans.p.halvorsen@usn.no](mailto:hans.p.halvorsen@usn.no)

Web: <https://www.halvorsen.blog>

