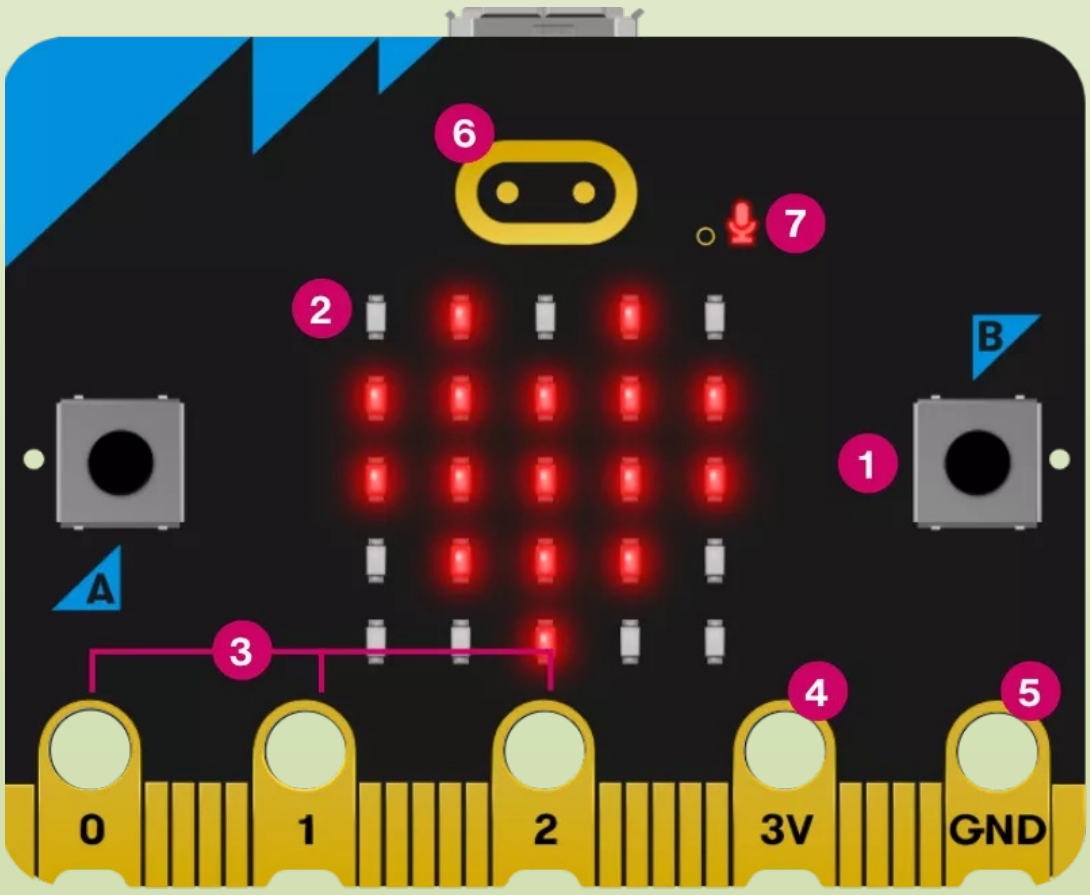
micro:bit

Introduction to micro:bit

Hans-Petter Halvorsen

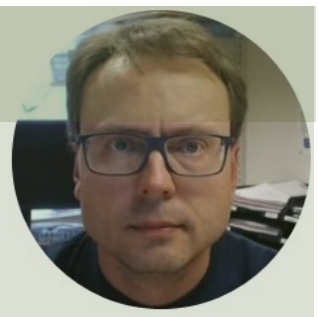


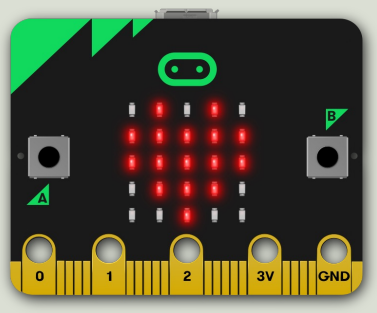
https://www.halvorsen.blog

Cover image: https://microbit.org/

Introduction to micro:bit

https://www.halvorsen.blog



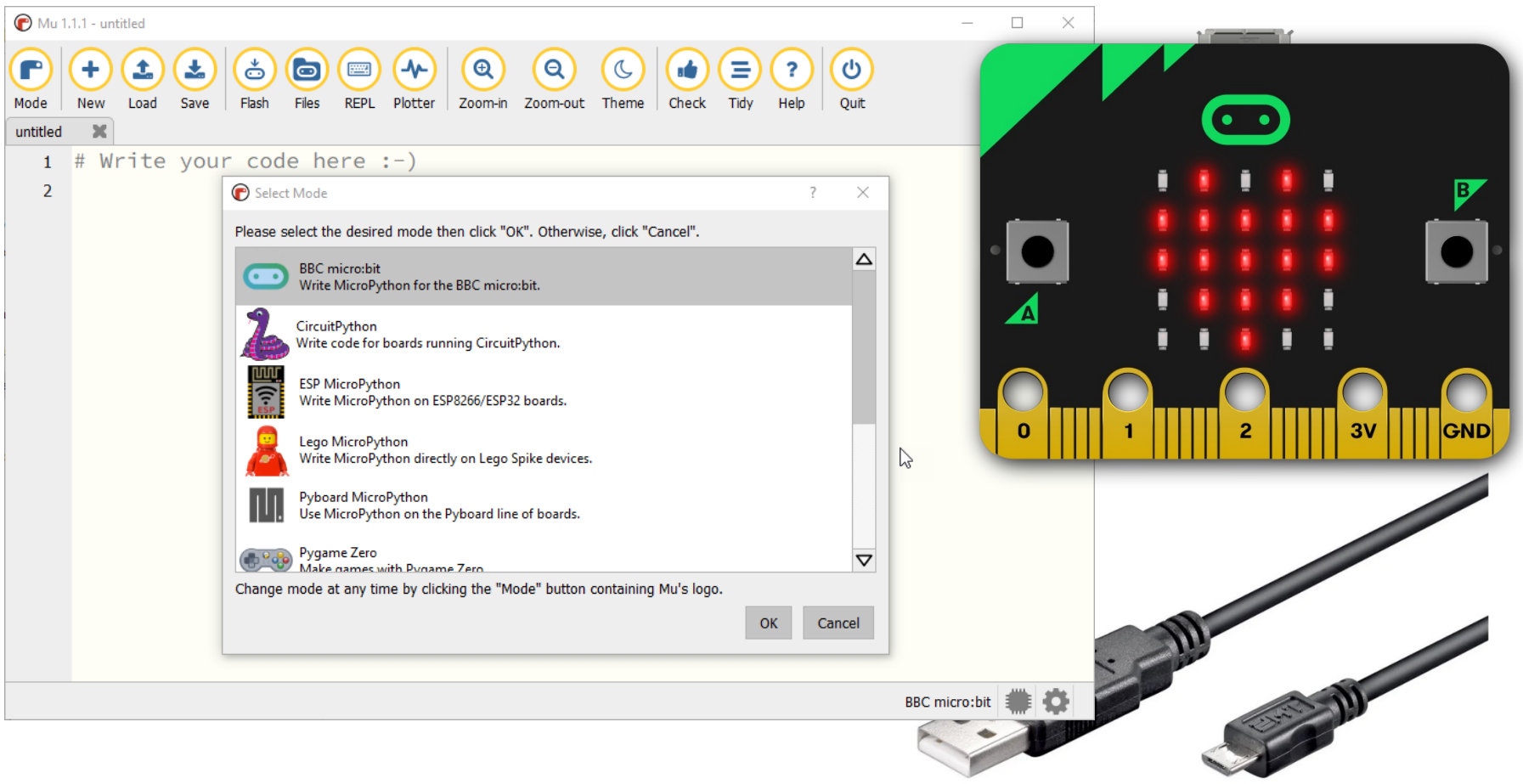


micro:bit

Hans-Petter Halvorsen

Contents

- Introduction to micro:bit
- micro:bit and Python and MicroPython
- Mu Python Editor - Python Editor with built-in support for micro:bit
- micro:bit Interfaces with Python Examples
  - LED Matrix (5x5)
  - Buttons (A and B)
  - Temperature Sensor
  - Light Sensor
  - Accelerometer
  - Compass
  - I/O Pins



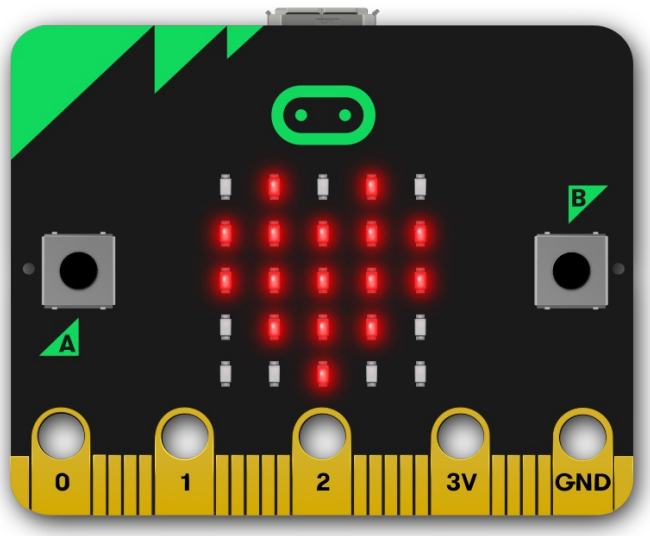
https://www.halvorsen.blog

Introduction to micro:bit

Hans-Petter Halvorsen

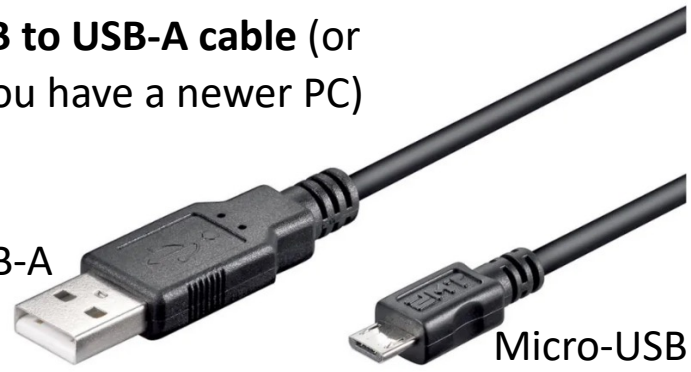
Table of Contents

What do you need?



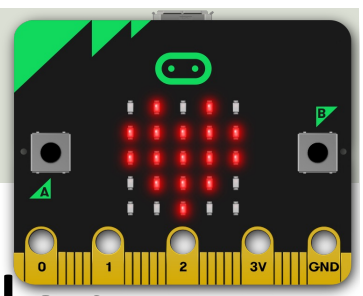
micro:bit

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



A PC

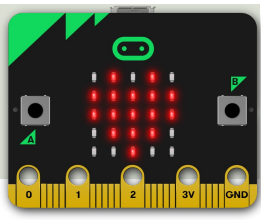
micro:bit



- 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

https://microbit.org

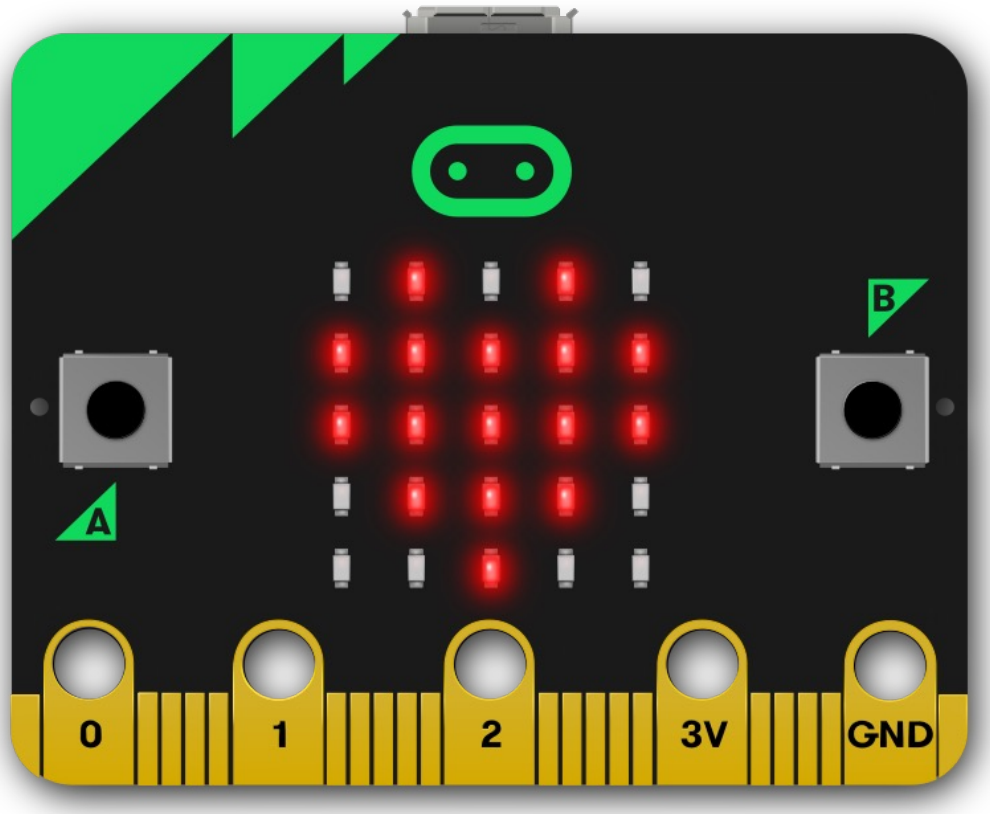
micro:bit and Programming



- micro:bit can run a special version of Python called MicroPython
- MicroPython is a down-scaled version of Python
- You can use other Programming Languages as well, but this Tutorial will use Python/ MicroPython

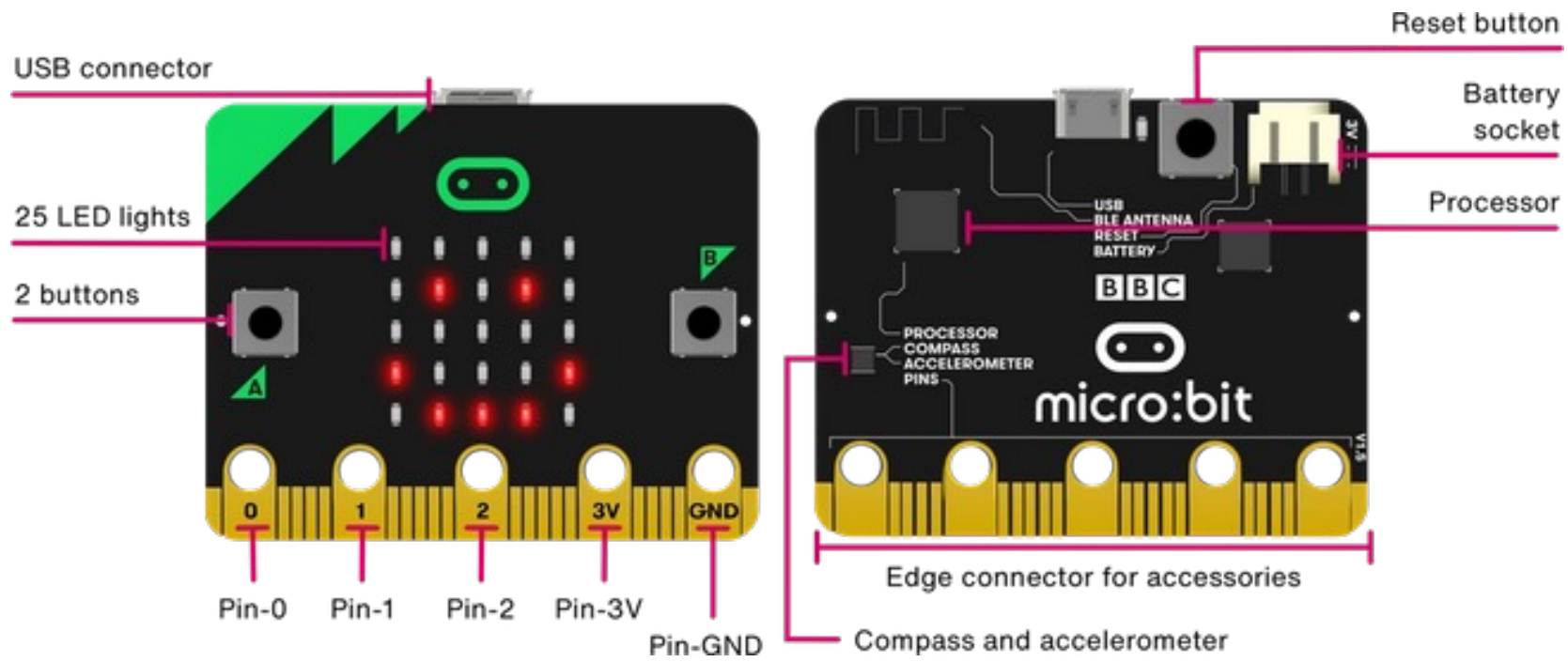
https://microbit.org

micro:bit



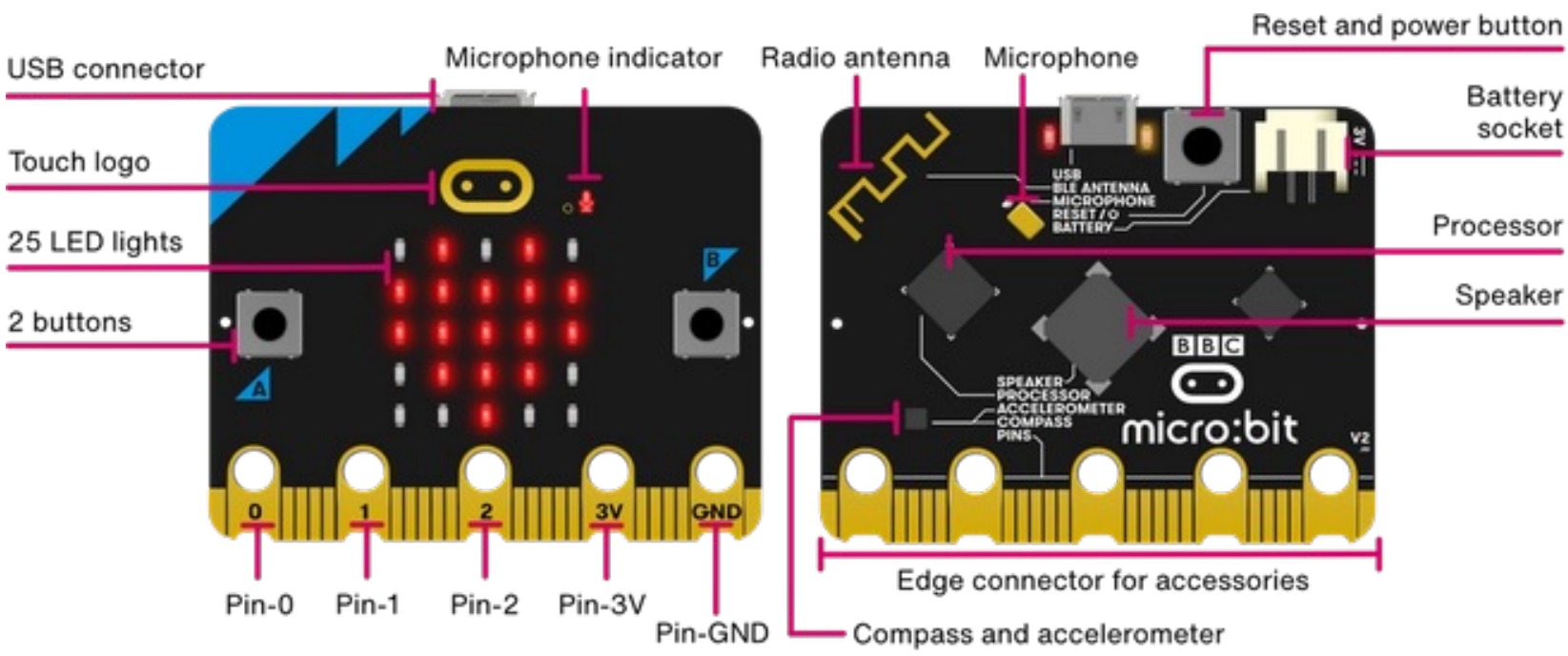
https://microbit.org

Original micro:bit



https://microbit.org

New micro:bit (micro:bit v2)



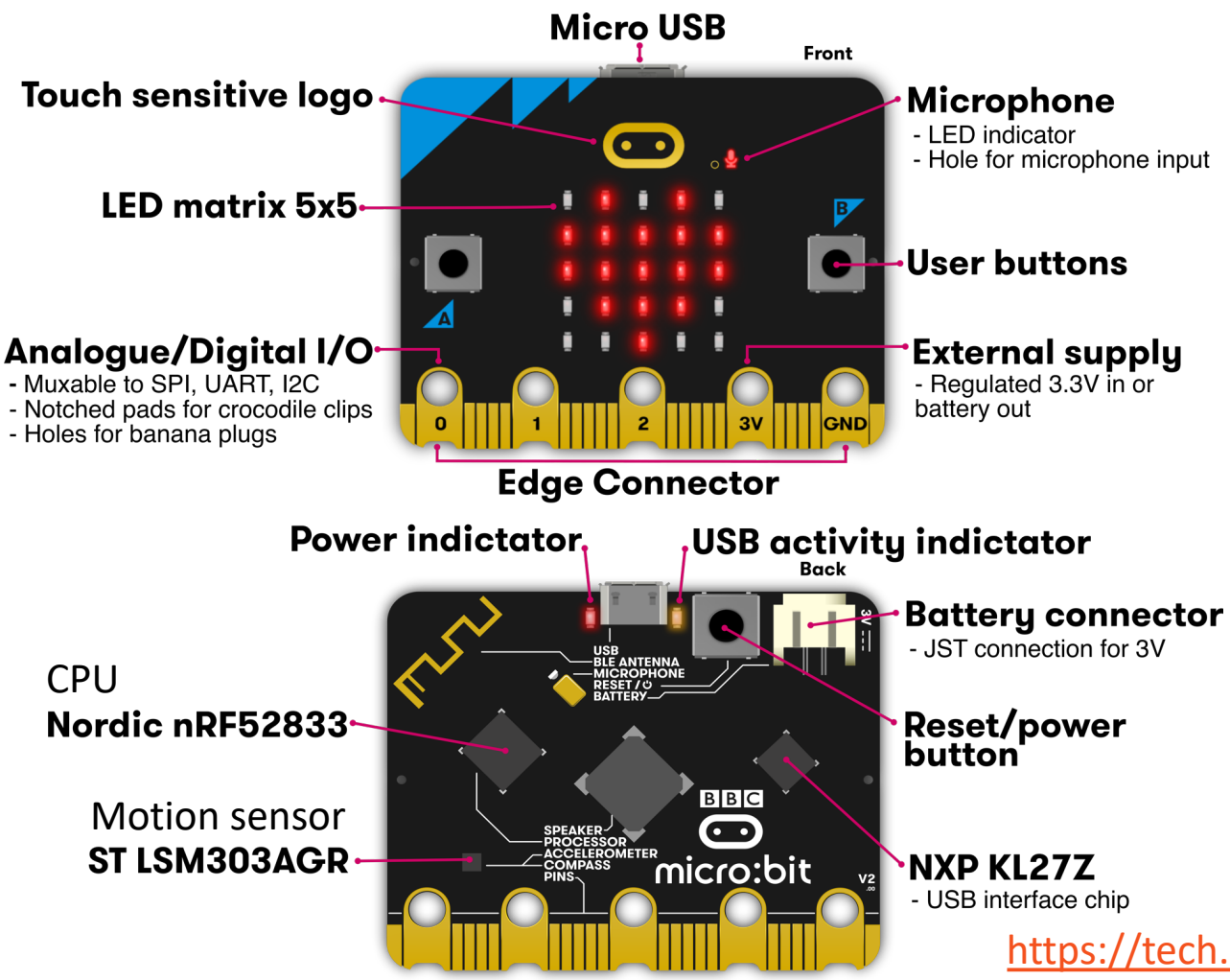
https://youtu.be/pIUJ4kvJ_QU

https://microbit.org

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)

https://microbit.org



https://tech.microbit.org/hardware/

https://www.halvorsen.blog

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; the **Mu Python Editor** is recommended

MicroPython

https://micropython.org

- 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

micro:bit Python Editors

Here are some Editors:

- Online Editor (used in your Browser)
  https://python.microbit.org
- Mu Python Editor
  https://codewith.mu

https://www.halvorsen.blog

Mu Python Editor

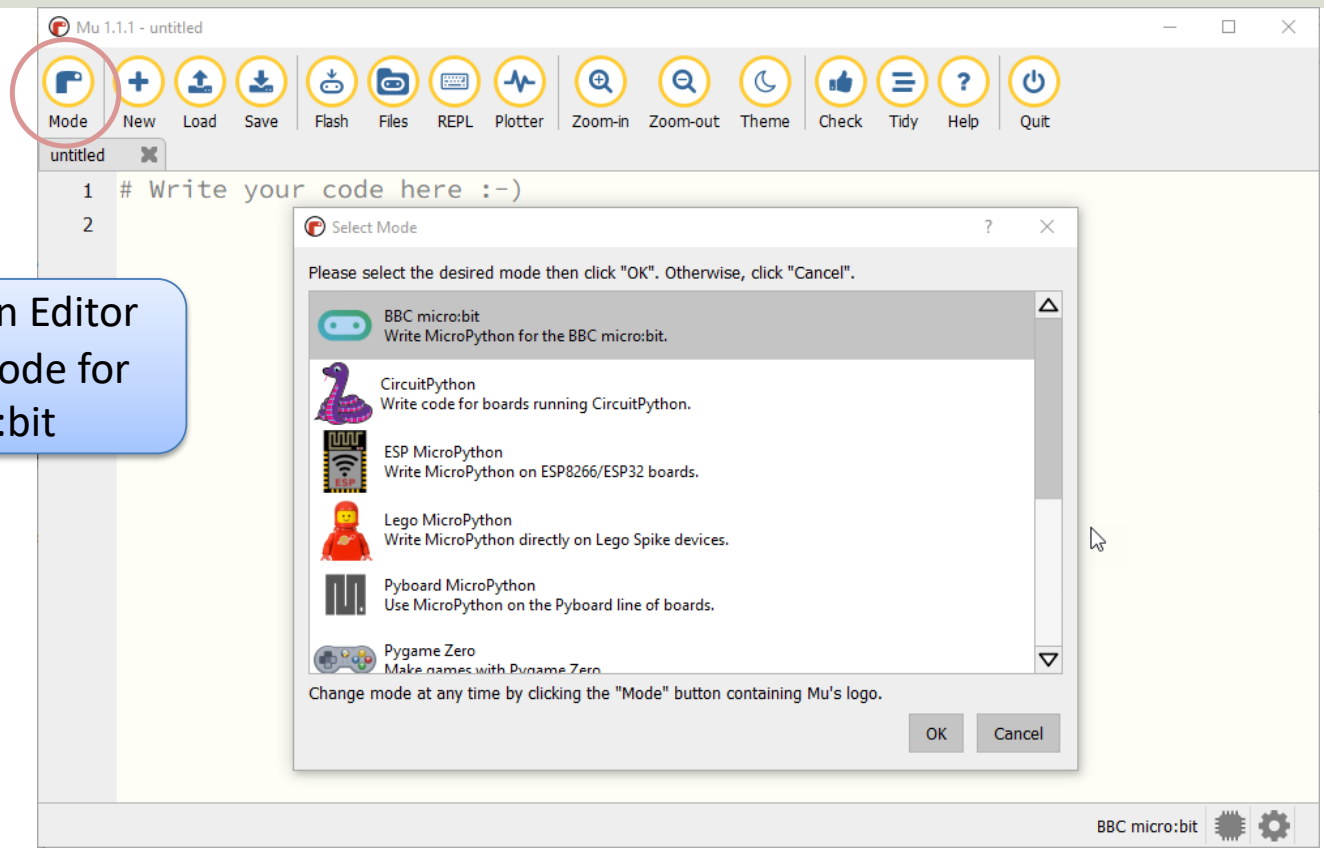
Hans-Petter Halvorsen

Table of Contents

Mu Python Editor

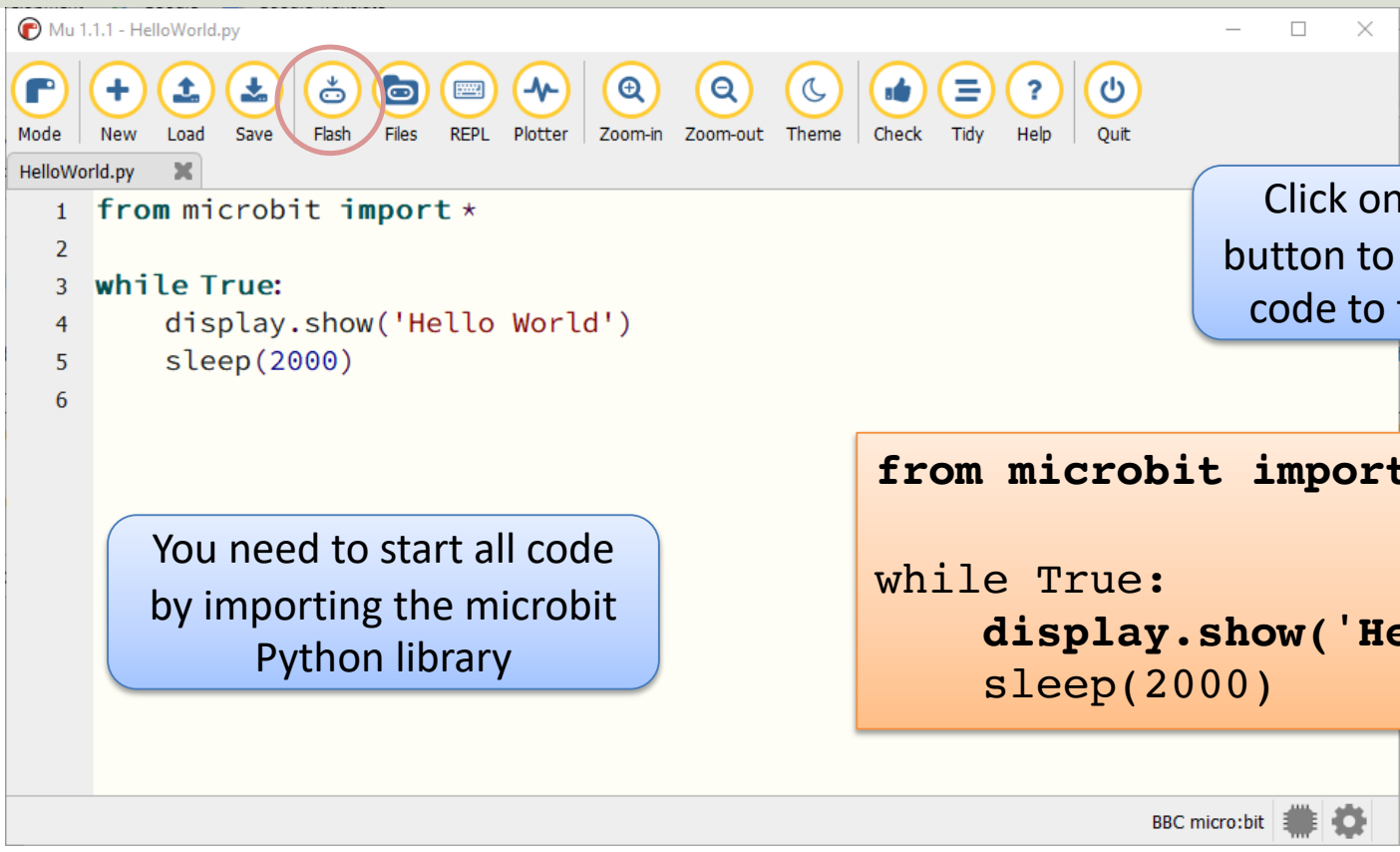
- Mu is a Python code editor for beginners
- It is tailor-made for micro:bit programming
- Mu has a “micro:bit mode” that makes it easy to work with micro:bit, download code to the micro:bit hardware, etc.
- Mu and micro:bit Tutorials: https://codewith.mu/en/tutorials/1.0/microbit

Mu Python Editor



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

Hello World

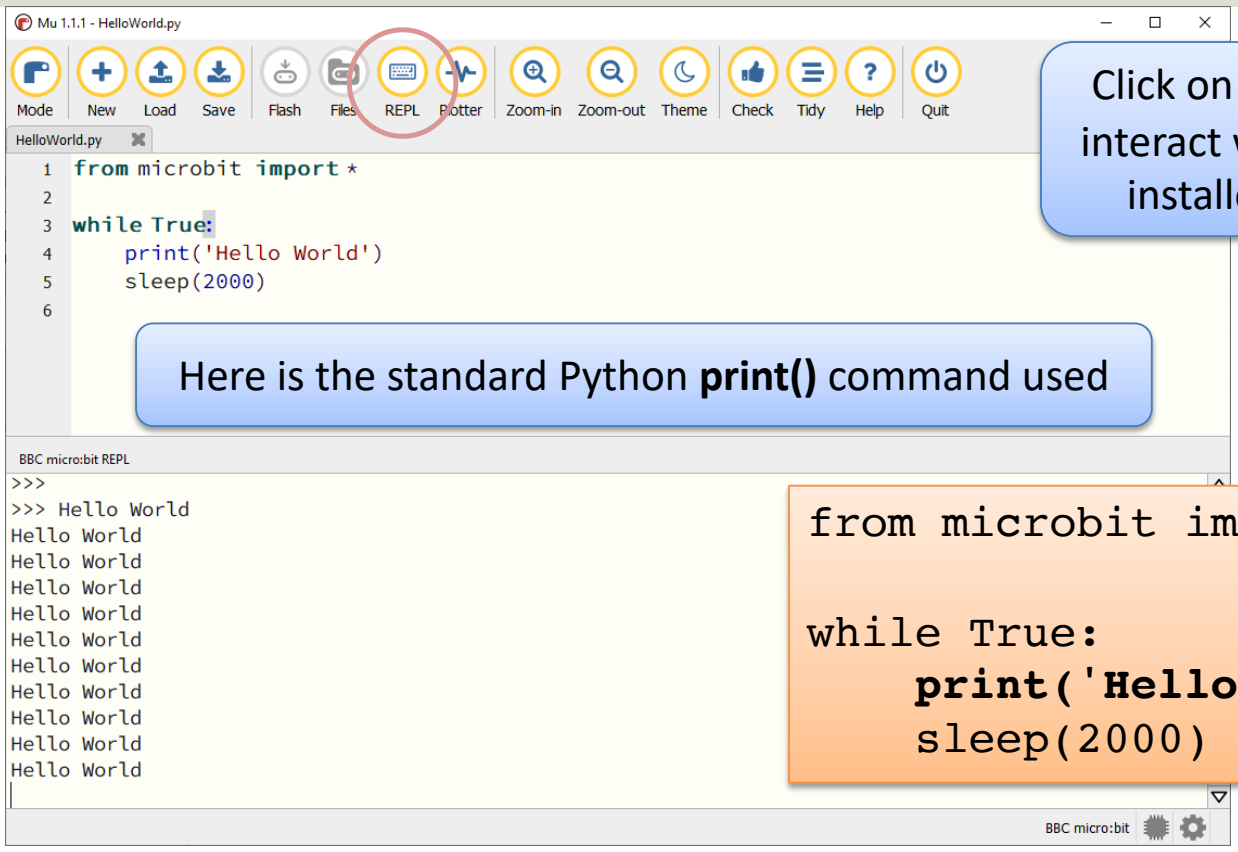


Click on the “Flash” button to download the code to the micro:bit

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

```
from microbit import *

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

REPL

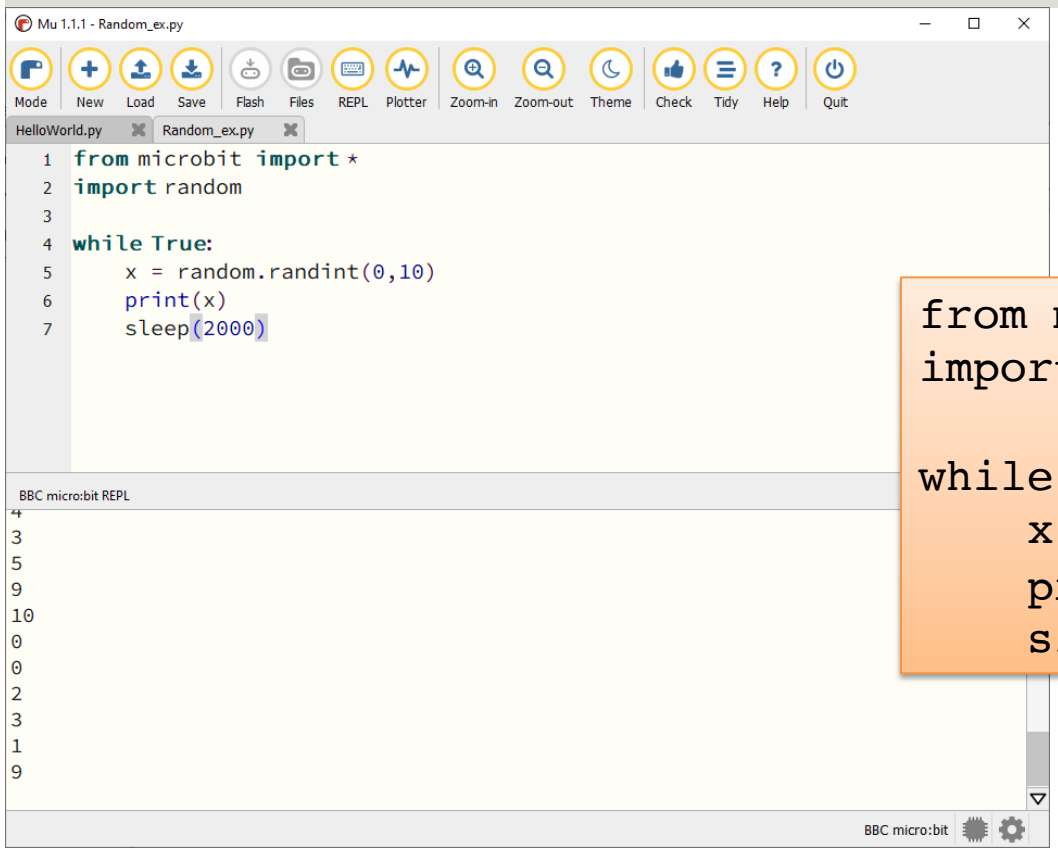


Click on the "REPL" button to interact with the MicroPython installed on the micro:bit

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

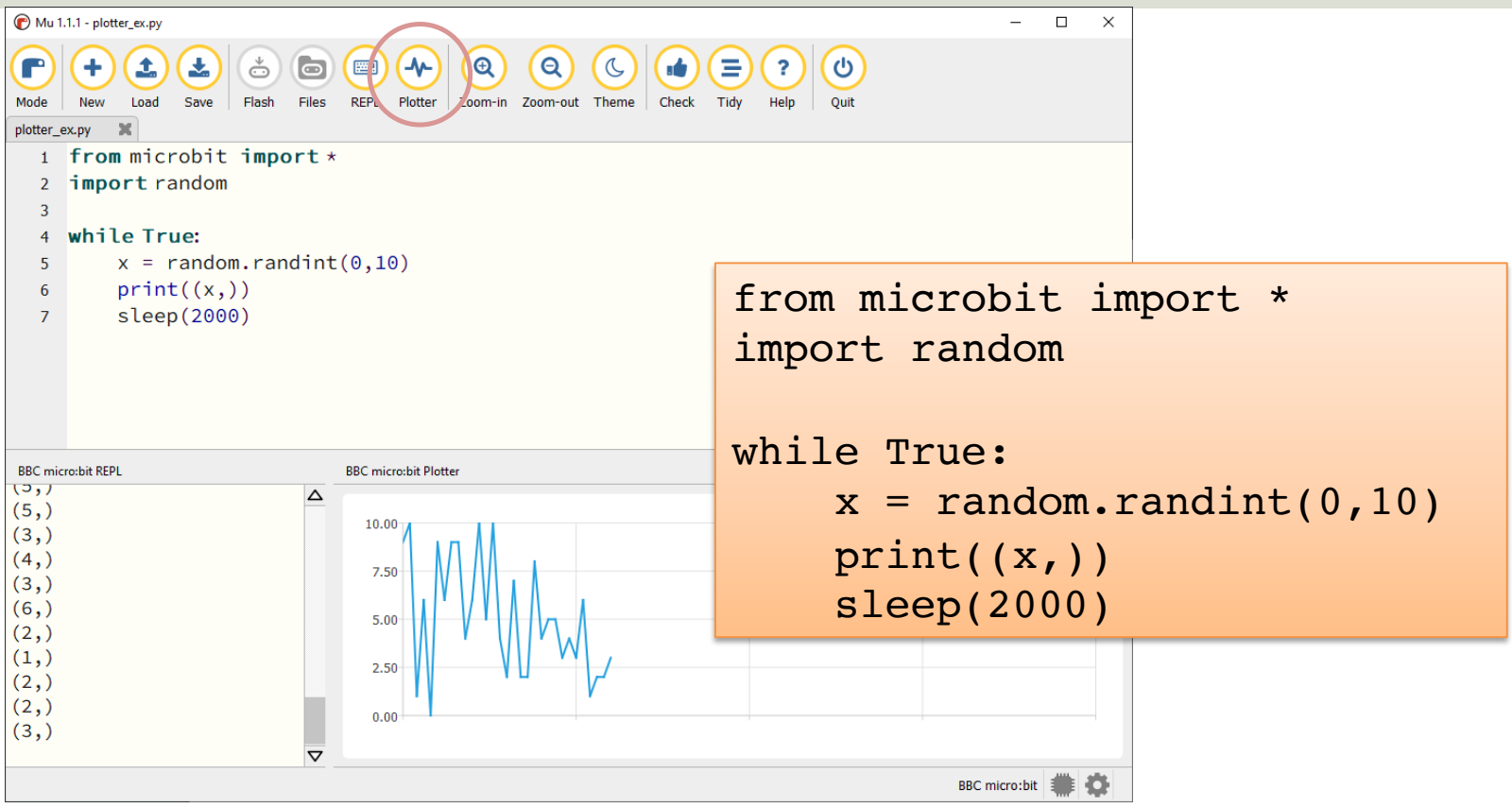
```
from microbit import *

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

REPL – Random Numbers



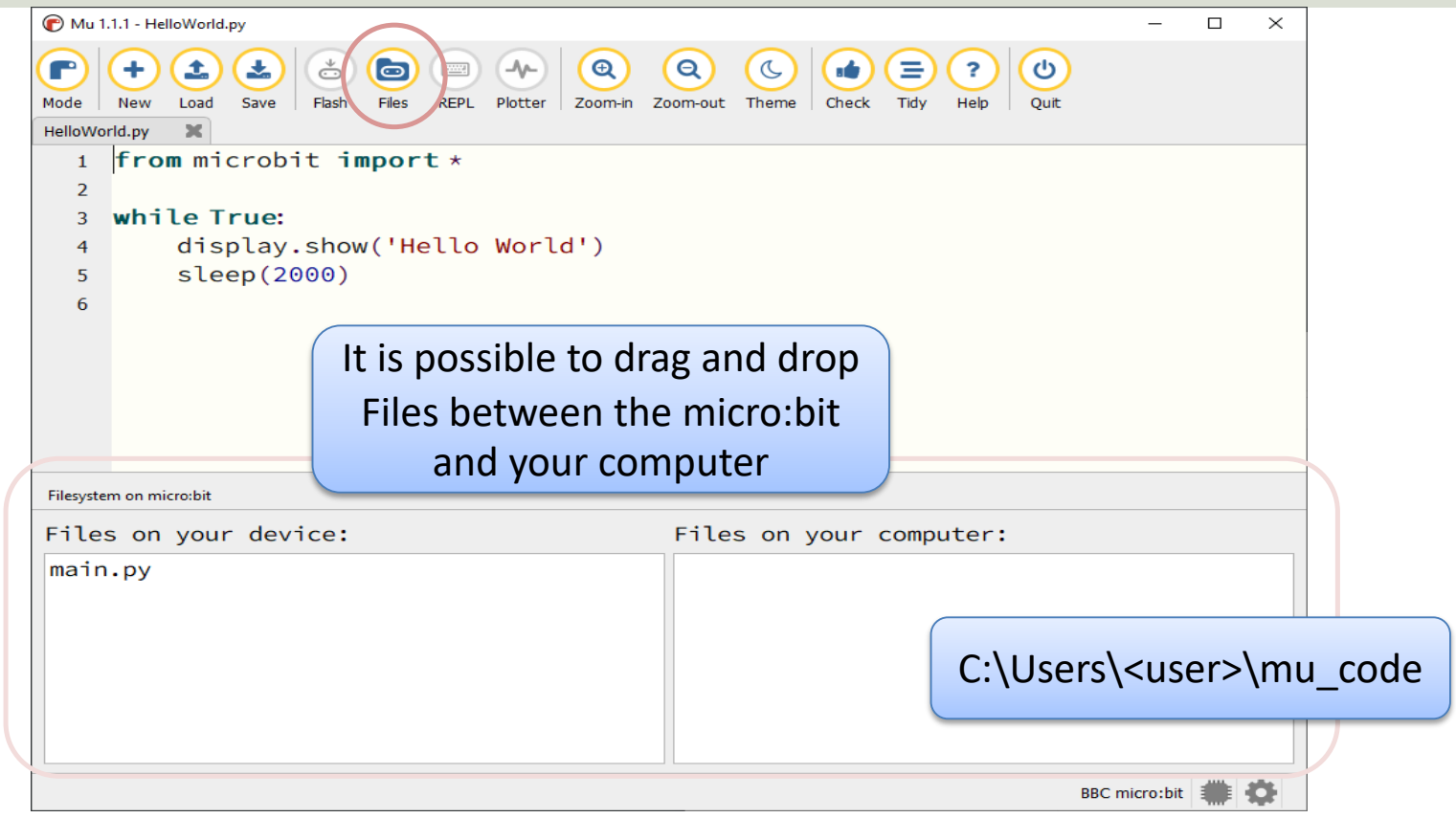

```
from microbit import *
import random

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

Plotter




```
from microbit import *
import random

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

Files



https://www.halvorsen.blog

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

https://www.halvorsen.blog

LED Matrix (5x5)

Hans-Petter Halvorsen

Table of Contents

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")
```

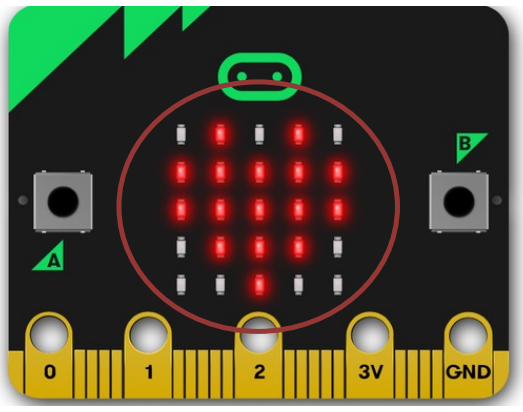
The word "WELCOME" will scroll over the LED matrix

```
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.

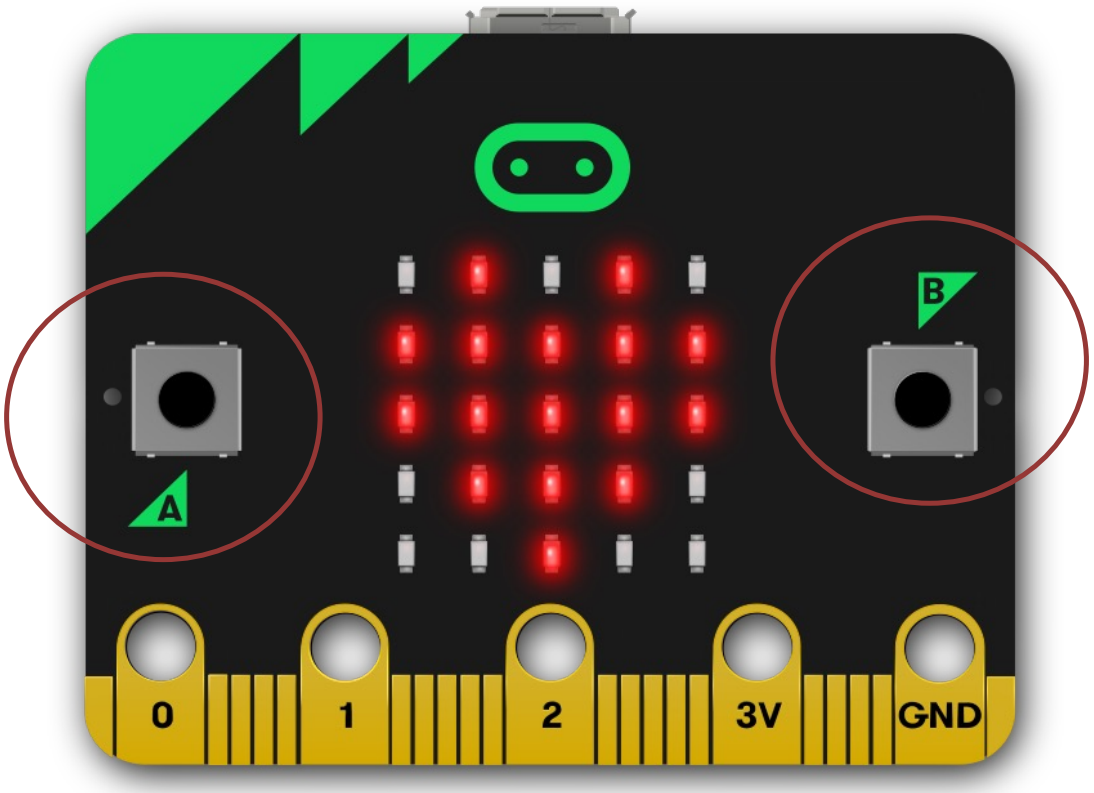
https://www.halvorsen.blog

Buttons (A and B)

Hans-Petter Halvorsen

Table of Contents

Buttons (A and B)



Buttons (A and B)

```
from microbit import *

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

    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

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

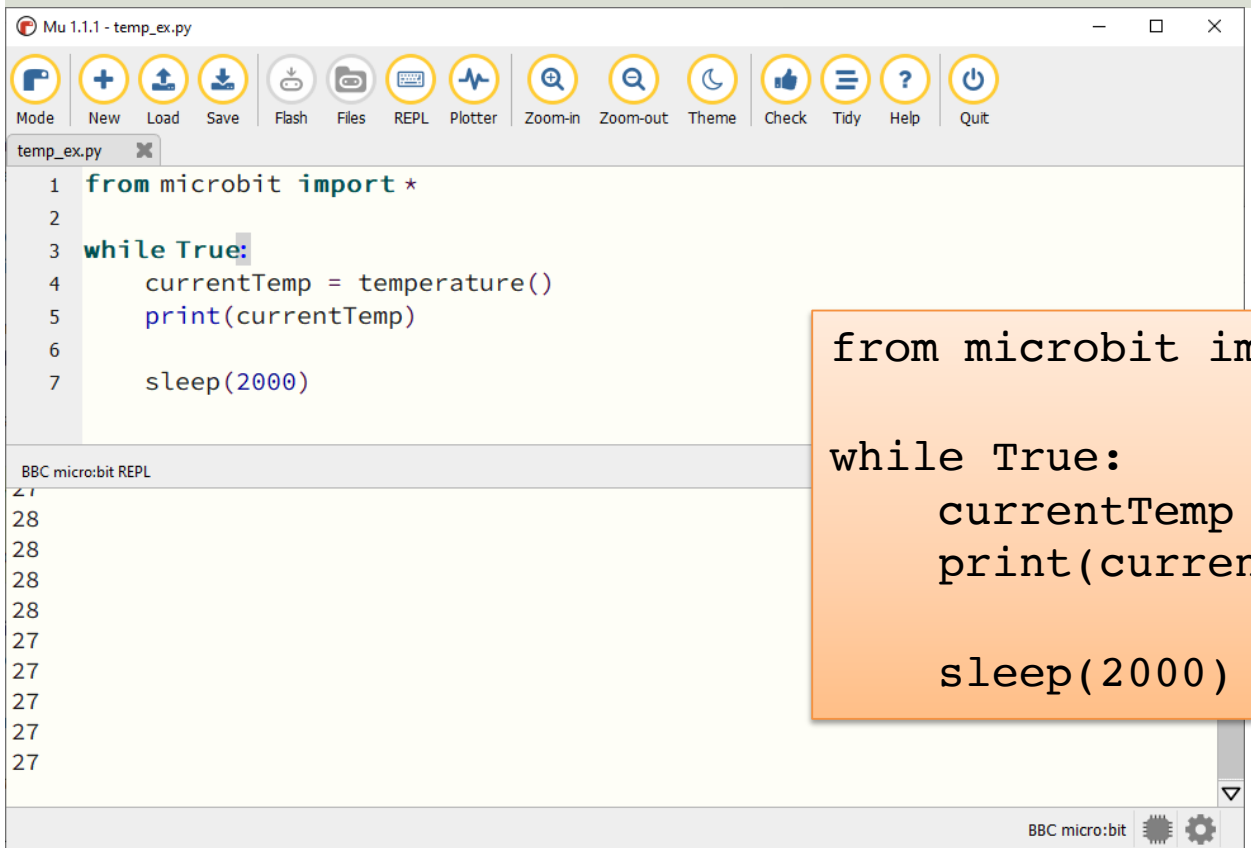
```
from microbit import *

currentTemp = temperature()
```

This examples displays the temperature on the LED matrix:

```
from microbit import *

while True:
    if button_a.was_pressed():
        display.scroll(temperature())
```

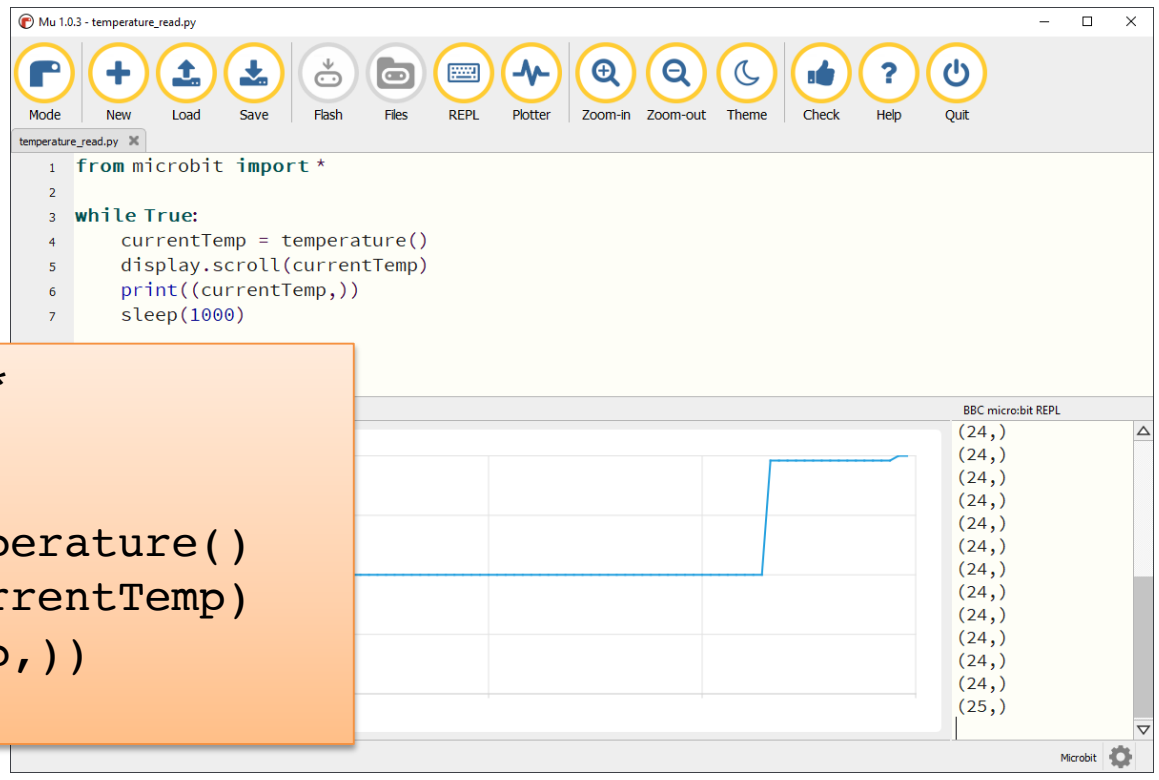
https://microbit.org/get-started/user-guide/features-in-depth/#temperature-sensor

Temperature Sensor




```
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)
```

Display Min/Max Temperature

```
from microbit import *

currentTemp = temperature()
maxTemp = currentTemp
minTemp = currentTemp

while True:
    currentTemp = temperature()

    if currentTemp < minTemp:
        minTemp = currentTemp
    if currentTemp > maxTemp:
        maxTemp = currentTemp

    if button_a.was_pressed():
        display.scroll(minTemp)
    elif button_b.was_pressed():
        display.scroll(maxTemp)
    else:
        display.scroll(currentTemp)

    print((currentTemp, minTemp, maxTemp))
    sleep(2000)
```

If you do nothing, the LED matrix shows the Current Temperature.

If you click A Button, the Minimum Temperature for the period (since you started the program/turned on the Micro:bit) is shown on the LED matrix

If you click B Button, the Maximum Temperature for the period (since you started the program/turned on the Micro:bit) is shown on the LED matrix

https://www.halvorsen.blog

Light Sensor

Hans-Petter Halvorsen

Table of Contents

Light Sensor

The LED matrix display on the front of your micro:bit can also **detect** light

```
from microbit import *

lightlimit = 100

while True:
    if display.read_light_level() > lightlimit:
        display.show(Image.HAPPY) #Happy because sunny
    else:
        display.show(Image.SAD) #Sad because cloudy
    sleep(2000)
```

In this Example, hold your micro:bit in front of a light source (e.g., a flashlight) and turn it on and off. The Image on the LED matrix should go from Sad to Happy or opposite.

Light Sensor

```
from microbit import *

lightlimit = 100

def sunlight():
    display.show(Image(
        "00000:"
        "00900:"
        "09990:"
        "00900:"
        "00000"))
    sleep(500)
    display.show(Image(
        "00000:"
        "09990:"
        "09990:"
        "09990:"
        "00000"))
    sleep(500)
    display.show(Image(
        "90909:"
        "09990:"
        "99999:"
        "09990:"
        "90909"))

while True:
    if display.read_light_level() > lightlimit:
        sunlight()
    else:
        display.show(Image.SAD) #Sad because cloudy weather
    sleep(2000)
```

Shows a flashing sunny image

https://www.halvorsen.blog

Accelerometer

Hans-Petter Halvorsen

Table of Contents

Accelerometer

Basic Example using the Accelerometer functionality:

```
from microbit import *
import random

while True:
    if accelerometer.was_gesture('shake'):
        display.show(random.randint(1, 6))
```

After shaking the micro:bit, a number between 1 and 6 is shown

Dices

```
from microbit import *
import random

while True:
    if accelerometer.was_gesture('shake'):
        number = random.randint(1, 6)
        if number == 1:
            display.show(Image(
            "00000:"
            "00000:"
            "00900:"
            "00000:"
            "00000"))
        elif number == 2:
            display.show(Image(
            "00000:"
            "00000:"
            "90009:"
            "00000:"
            "00000"))
        elif number == 3:
            display.show(Image(
            "00009:"
            "00000:"
            "00900:"
            "00000:"
            "90000"))
        elif number == 4:
            display.show(Image(
            "90009:"
            "00000:"
            "00000:"
            "00000:"
            "90009"))
        elif number == 5:
            display.show(Image(
            "90009:"
            "00000:"
            "00900:"
            "00000:"
            "90009"))
        else:
            display.show(Image(
            "90009:"
            "00000:"
            "90009:"
            "00000:"
            "90009"))
```

After shaking the micro:bit, a dice is shown with 1, 2, 3, 4, 5, or 6 eyes

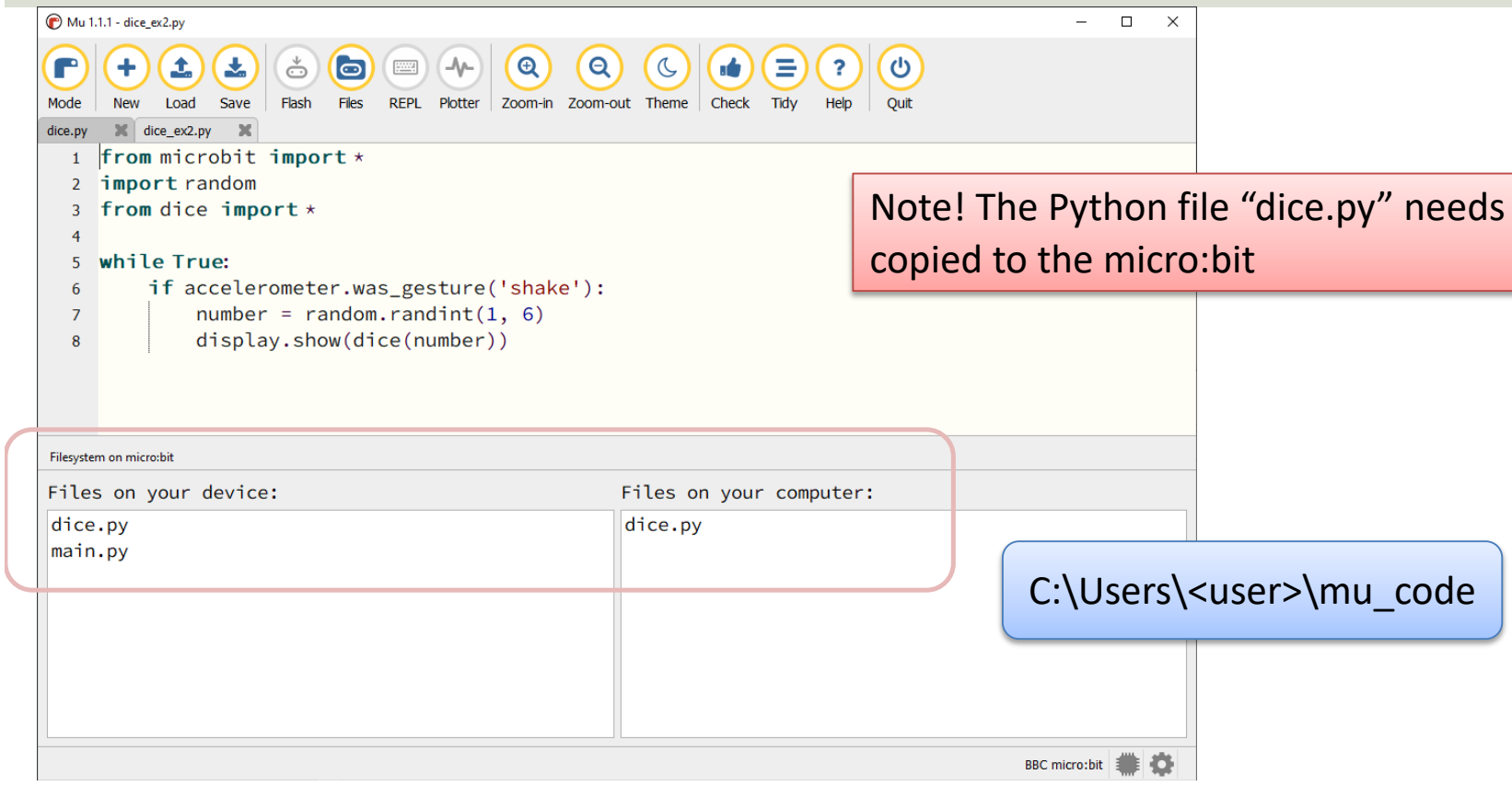
Dices Improved

```
from microbit import *

def dice(number):

    if number == 1:
        diceimage = Image("00000:"
                          "00000:"
                          "00900:"
                          "00000:"
                          "00000")
    elif number == 2:
        diceimage = Image("00000:"
                          "00000:"
                          "90009:"
                          "00000:"
                          "00000")
    elif number == 3:
        diceimage = Image("00009:"
                          "00000:"
                          "00900:"
                          "00000:"
                          "90000")
    elif number == 4:
        diceimage = Image("90009:"
                          "00000:"
                          "00000:"
                          "00000:"
                          "90009")
    elif number == 5:
        diceimage = Image("90009:"
                          "00000:"
                          "00900:"
                          "00000:"
                          "90009")
    else:
        diceimage = Image("90009:"
                          "00000:"
                          "90009:"
                          "00000:"
                          "90009")

    return diceimage
```

dice.py

```
from microbit import *
import random
from dice import *

while True:
    if accelerometer.was_gesture('shake'):
        number = random.randint(1, 6)
        display.show(dice(number))
```

Dices



Note! The Python file “dice.py” needs to be copied to the micro:bit

C:\Users\<user>\mu_code

https://www.halvorsen.blog

Compass

Hans-Petter Halvorsen

Table of Contents

Compass

The micro:bit has a built-in compass sensor called a magnetometer. You can use it to measure the Earth's magnetic field and use it as a compass.
When you first use the micro:bit compass you must calibrate it – a little game appears on the screen where you must tilt the micro:bit to light up every LED, then you're ready to go.

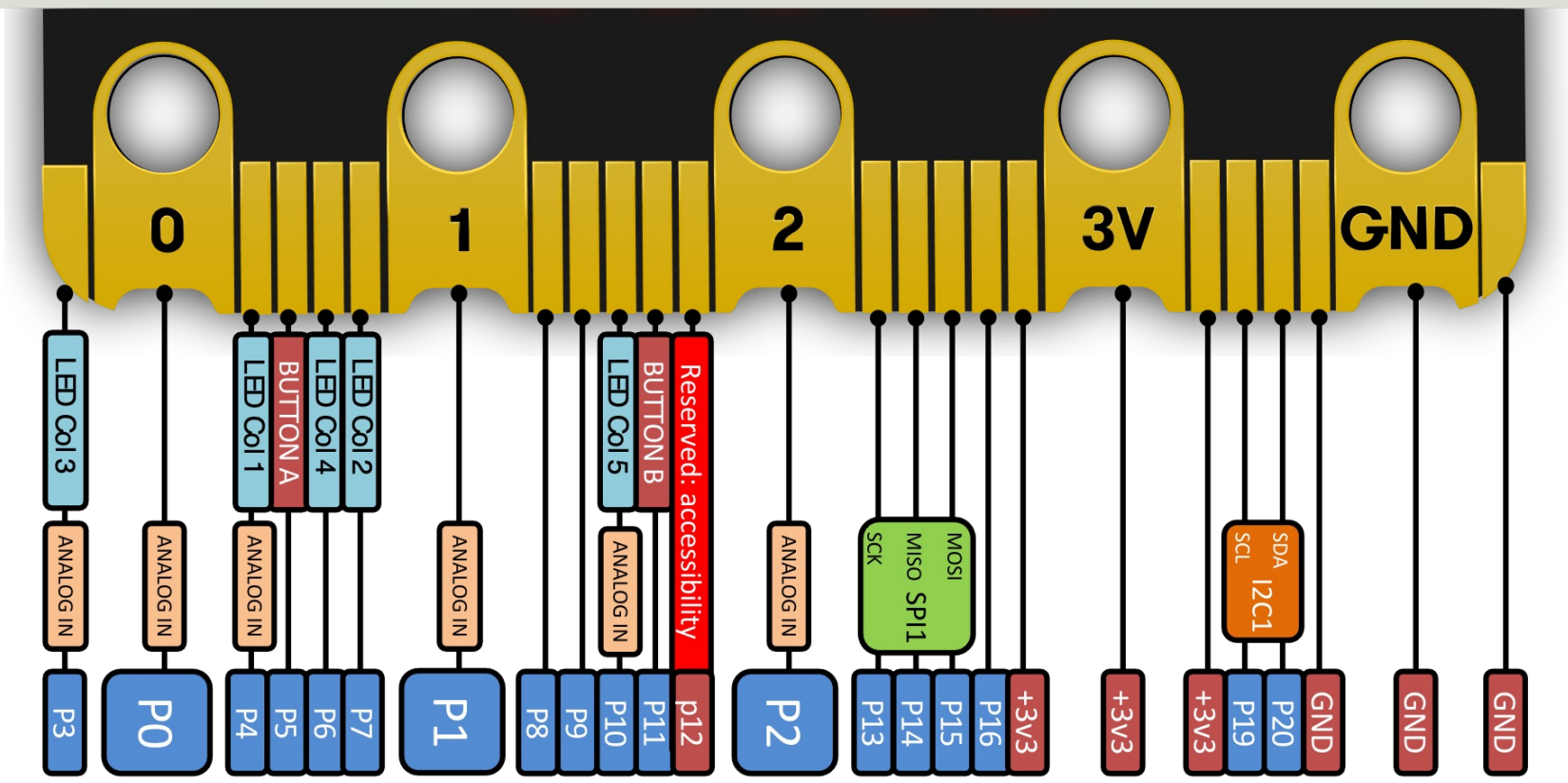
```
from microbit import *

while True:
    if button_a.was_pressed():
        display.scroll(str(compass.heading()))
```

https://microbit.org/projects/make-it-code-it/compass-bearing/?editor=python

I/O Pins

I/O Pin Overview v.2

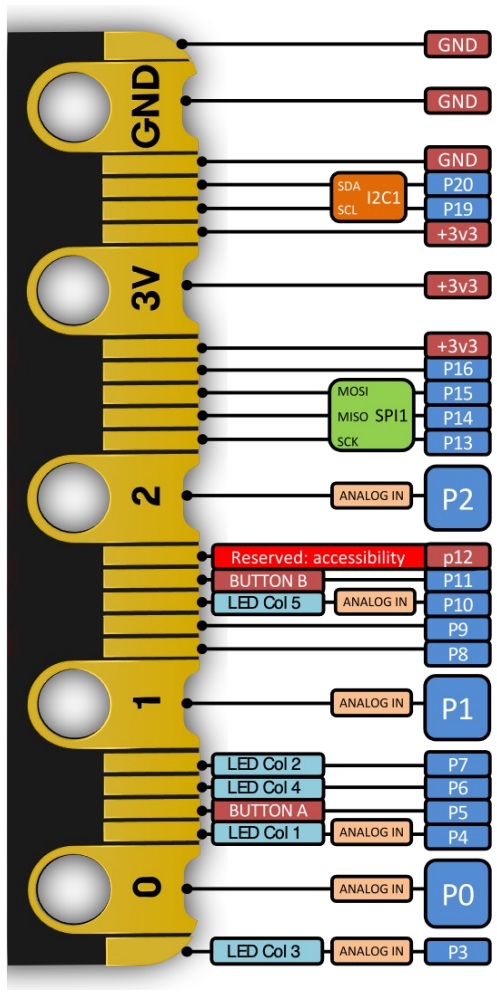
https://microbit.pinout.xyz/



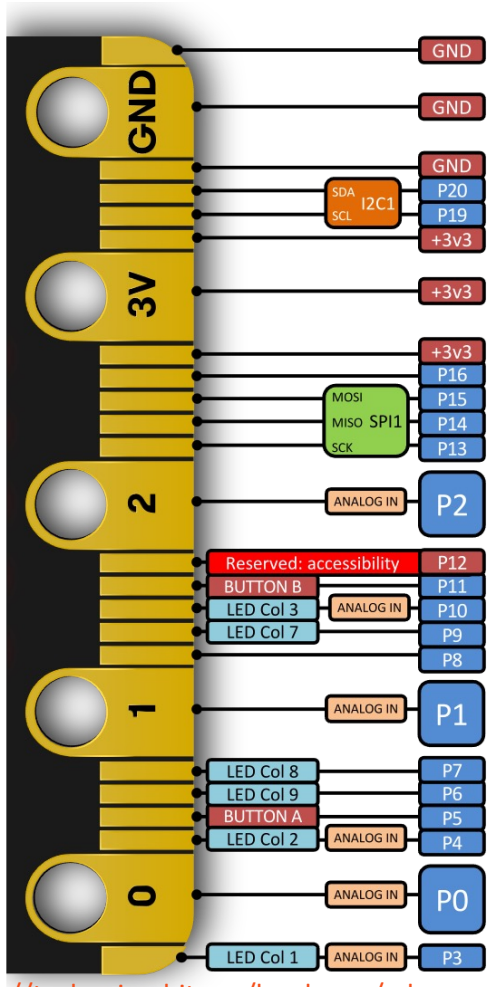
I/O Pin Overview

https://microbit.pinout.xyz/

New micro:bit (micro:bit v2)

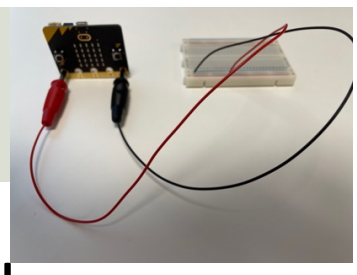


Original micro:bit

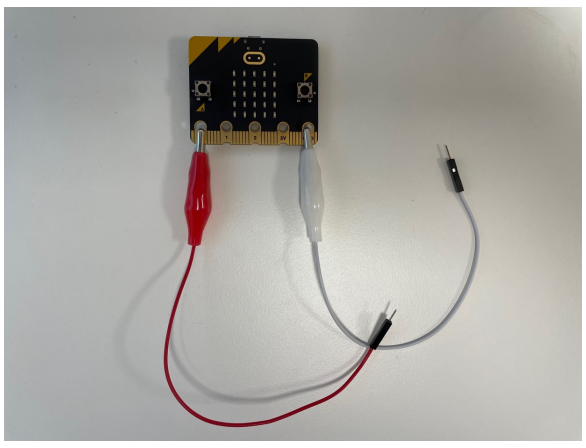


https://tech.microbit.org/hardware/edgeconnector/

I/O Pins



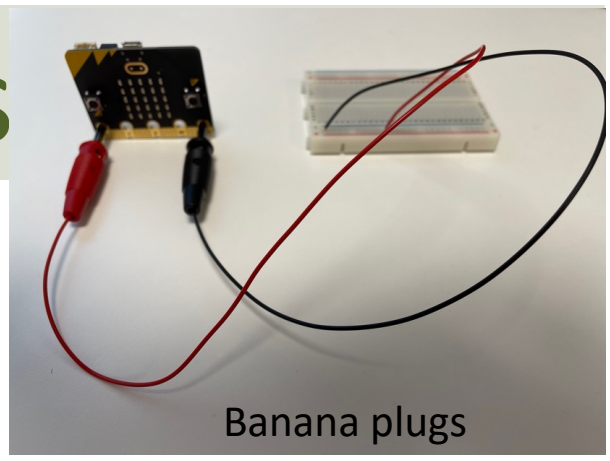
- We use the I/O pins to connect external components like LEDs, different types of Sensors, etc.
- You can use 4mm Banana plugs or alligator/crocodile clips
- Typically you also want to use a Breadboard

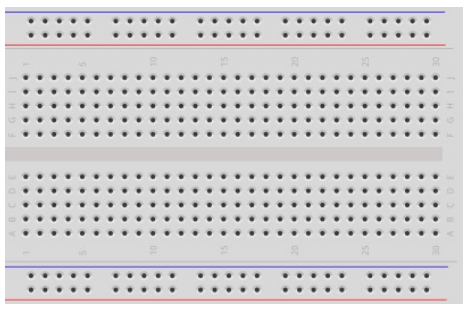


https://tech.microbit.org/hardware/edgeconnector/

https://makecode.microbit.org/device/crocodile-clips

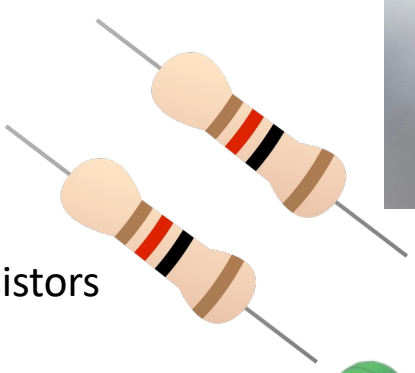
Component Examples



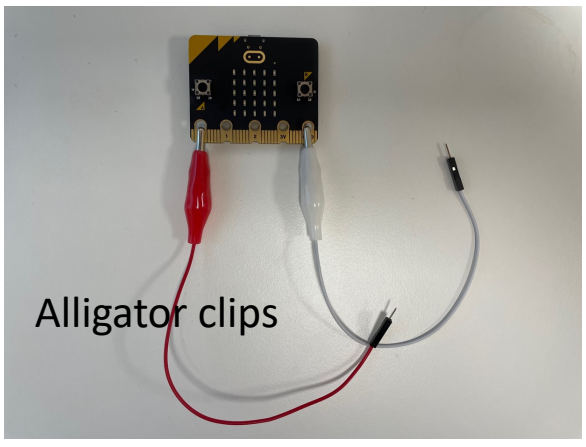


Breadboard

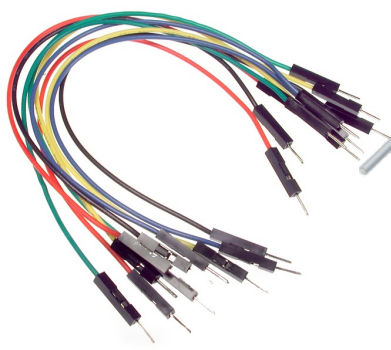
Temperature Sensor



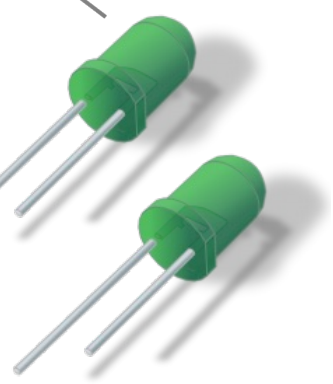
Resistors



Wires



LEDs



Multimeter

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

Hans-Petter Halvorsen

University of South-Eastern Norway

www.usn.no

E-mail: hans.p.halvorsen@usn.no

Web: https://www.halvorsen.blog

Cover image: https://microbit.org/

Introduction to micro:bit