Lesson 1 - Intro to Micro:bit, Buttons & Conditionals

By the end of the lesson, students should be able to:

Introduction to micro:bit

- Describe what is a microcontroller
- Import libraries
- Display text and images on the micro:bit
- Use the sleep() function to slow down the program
- Recall functions
- Recall and use for loops to repeat a code for a specific number of times
- Navigate the microbit documentation

Buttons & Conditionals

- Describe what is NameError and when it occurs
- Identify the differences between show and scroll
- Recall and use conditionals to determine if a button is pressed
- Recall and use while loops to create a forever loop
- Recall and use variables to store data

Version
Date: January 2020
Format: 8 lessons x 2 hours
Important! View speaker notes for details

Things to note

- 🔌 **Unplugged** = Activities not involving technology (Videos, Kinaesthetic activities etc.)
- 💬 **Discussion** = Get the students to think and respond about a question
- 👫 **Guided** = Demonstration → Instructor does the activity while the student mimics)
- 👐 **Unguided** = Instructor will give the students the task and show what the final result should look like and give the students a certain amount of time to do it by themselves before moving on to “Check for Understanding”
- 🎯 **Check for Understanding** = Instructor will go through the solution with them or get a student to share the solution
- 🚀 **Sandbox** = Free-Play (Students recap what they learnt from the entire day by creating a project)
- 🎁 **Bonus** = This is given to students who are fast-paced

Materials Needed

Per student:

- 1x microbit set
  - 1x microbit
  - 1x usb
  - 1x battery pack
  - 2x AAA batteries
- 1x Chromebook/Laptop

Frequently Asked Questions

“What if I can't finish the activities for that particular day? ”

- In the event that you can't finish all of the activities in the given time, DO NOT rush to finish the concepts and just continue where you left off the next week.
- The bonus activities are for the faster students that have completed the general task that was given to the whole class. You do not need to cover this with everyone.

“How do you know you’ve been teaching the right way?”

- When students are able to create their sandbox with minimal to no help from you.

Frequently Asked Questions

“What is the purpose of this course?”

- For students to practice applying Python knowledge learnt previously on a micro-controller and build structures with Strawbees to present their creations to tasks given.
- Students also learn more basic coding concepts through Python and use them via computational thinking.

“What is computational thinking?”

- Computational thinking allows us to take a complex problem, understand what the problem is and develop possible solutions. We can then present these solutions in a way that a computer, a human, or both, can understand.

Frequently Asked Questions

“Why must I follow the speaker notes and teach in a certain way? I prefer to freestyle.”

- For follow-up purposes as there will be cases where you might be unable to teach your class on a particular day and another instructor will need to cover you.

Frequently Asked Questions

“ Can students bring home the Strawbees structure?”

- No, but they can take pictures of their structures before dismantling them.

“ Can students bring home the microbit set?”

- No, but they can take videos of their projects/ pictures of their structures before dismantling them.

“Can students buy the Strawbees or microbit set?”

- No.

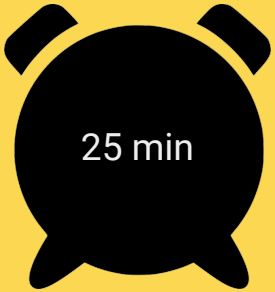


PYTHON

LEVEL 2




About the Program



Introductions

Ice Breakers

Attendance Taking

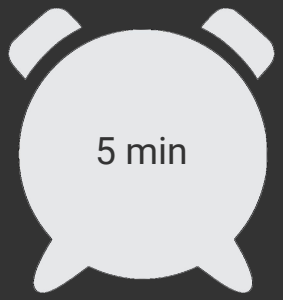
Please ensure that your attendance has been taken at the start of every lesson.

You will need to attain 80% attendance in order to graduate from this course.

Lesson 1	Lesson 2	Lesson 3	Lesson 4	Lesson 5	...
✅	✅	✅	✅	✅	...

Let’s create a social pact as a class!

What is expected of each student?



Agenda

Today’s Lesson

- Introduce the micro:bit
- micro:bit Documentation
- Name Badge
- Recalling Conditionals
- micro:bit Counter

Introduction to micro:bit

Unplugged

15 min

What is the micro:bit?

What is a microcontroller?

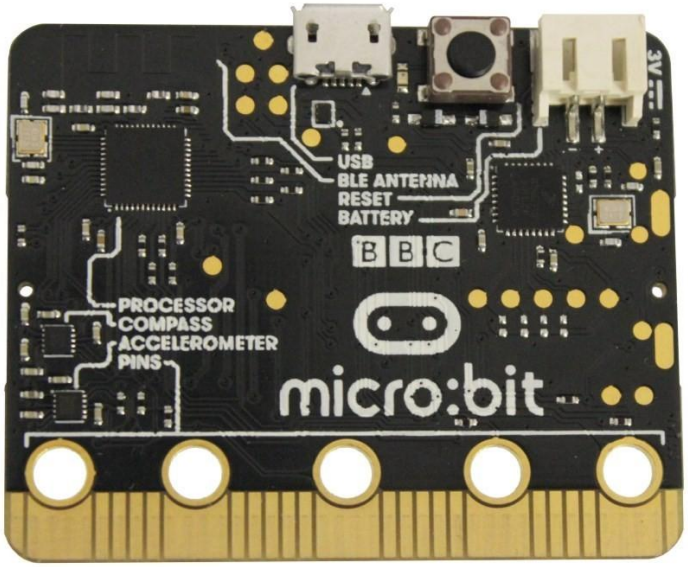
micro controller

↓ ↓

small computer

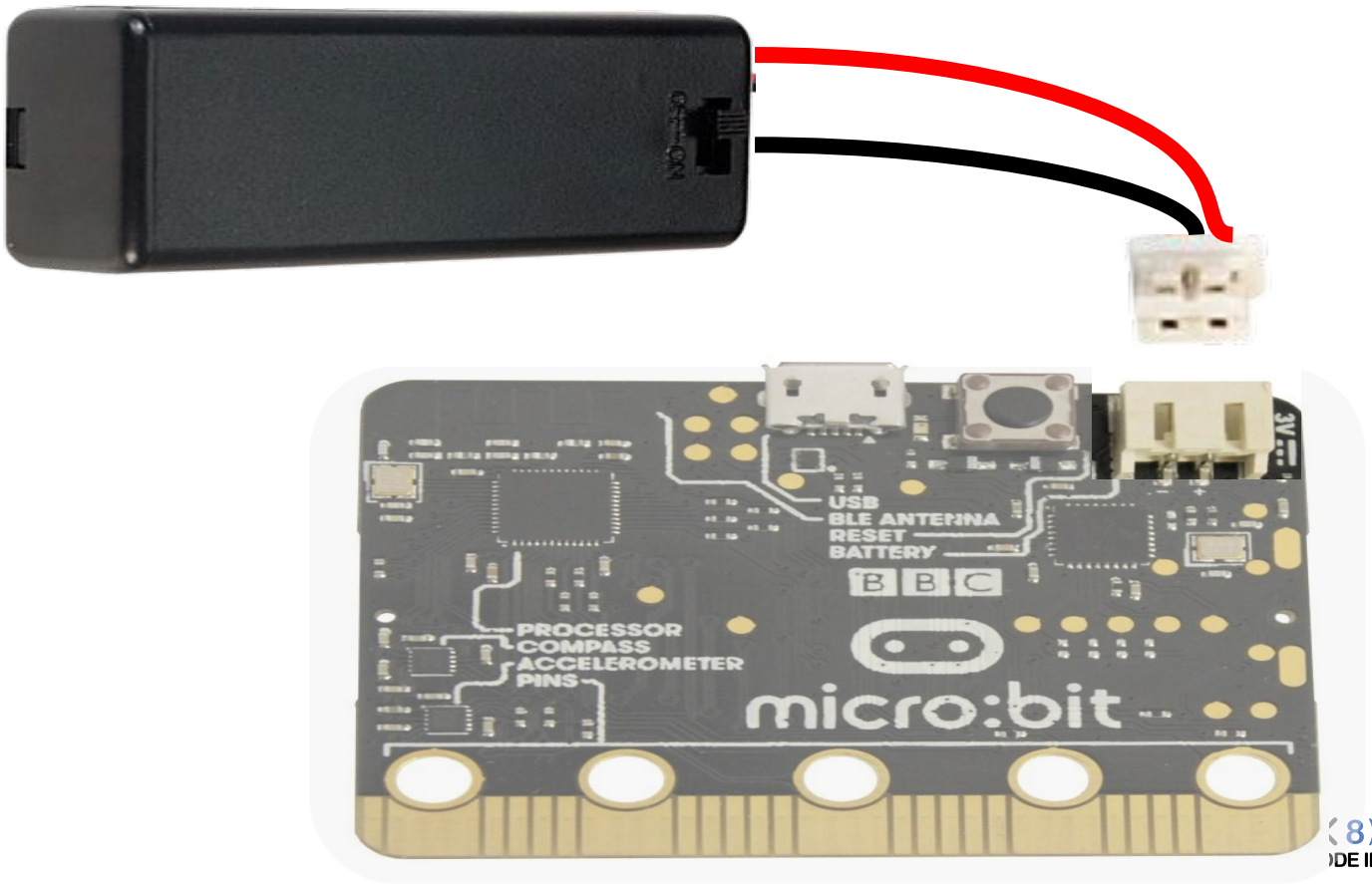
What are the micro:bit's components?

The micro:bit is a type of microcontroller



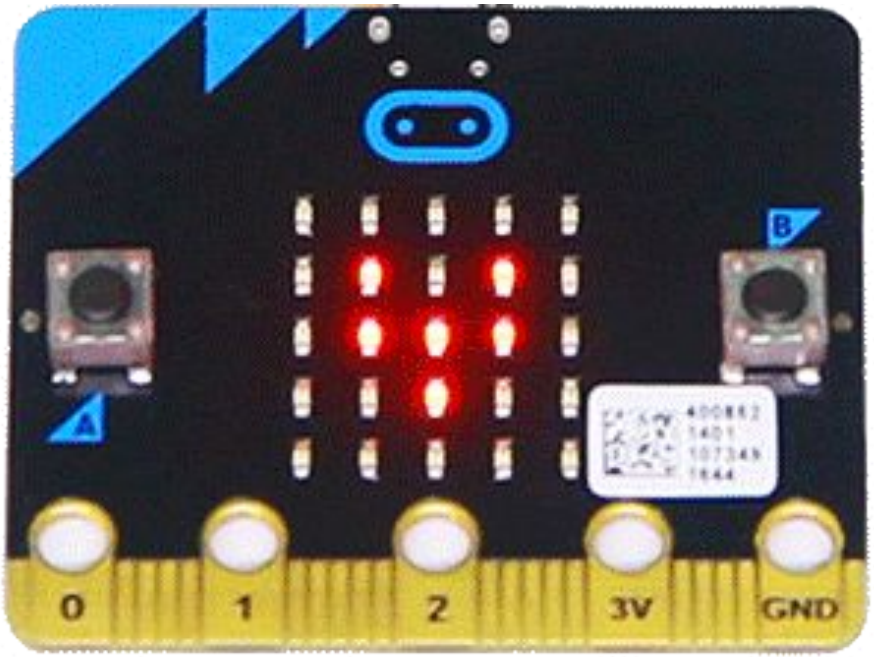
The processor runs the program on the micro:bit

The battery jack allows us to power the micro:bit with batteries



The USB Port uploads the program onto the micro:bit from your laptop

It has a 5x5 LED screen that would display images and scroll text messages across the screen.



Guided

10 min

micro:bit Documentation

Google "micro:bit micropython documentation"

Guided

micro:bit micropython documentation

microbit-micropython.readthedocs.io/en/latest/

BBC micro:bit MicroPython

latest

Search docs

TUTORIALS

- Introduction
- Hello, World!
- Images
- Buttons
- Input/Output
- Music
- Random
- Movement
- Gestures
- Direction
- Storage
- Speech
- Network
- Radio
- Next Steps

Read the Docs

Docs » BBC micro:bit MicroPython documentation

Edit on GitHub

BBC micro:bit MicroPython documentation

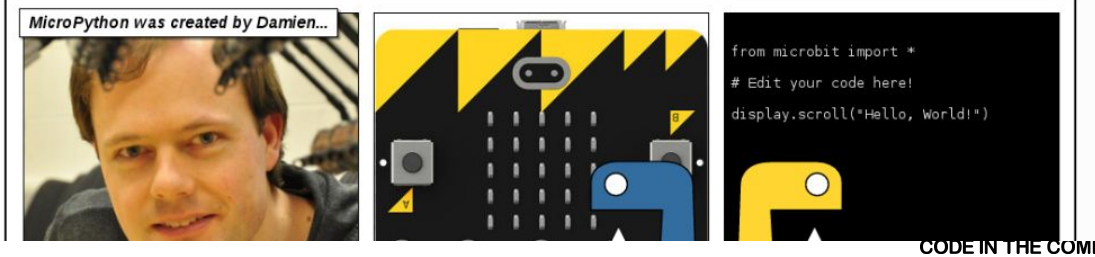
Welcome!

The BBC micro:bit is a small computing device for children. One of the languages it understands is the popular Python programming language. The version of Python that runs on the BBC micro:bit is called MicroPython.

This documentation includes lessons for teachers and API documentation for developers (check out the index on the left). We hope you enjoy developing for the BBC micro:bit using MicroPython.

If you're a new programmer, teacher or unsure where to start, begin with the tutorials.

First Steps with MicroPython by Mike Rowbitt



🔍 Find the Display documentation under API Reference on the left section

Guided

microbit-micropython.readthedocs.io/en/latest/display.html

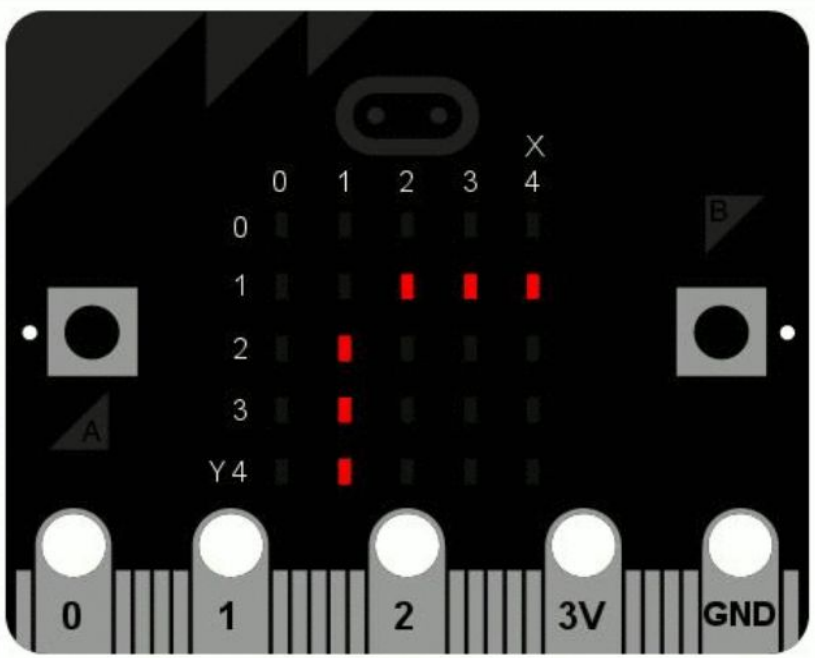
Radio
Next Steps
API REFERENCE
micro:bit Micropython API
Microbit Module
Accelerometer
Audio
Bluetooth
Buttons
Compass
Display
Local Persistent System
I²C
Image
Machine
MicroPython
Music
NeoPixel
The os Module
Input/Output Pins
Radio

Docs » Microbit Module » Display

Edit on GitHub

Display

This module controls the 5×5 LED display on the front of your board. It can be used to display images, animations and even text.



What is a function?

Function

Functions are blocks of code that accomplish a specific task.

A function can be used over and over again by "calling" it.

Guided

Functions

`microbit.display.get_pixel`(*x, y*)

Return the brightness of the LED at column `x` and row `y` as an integer between 0 (off) and 9 (bright).

`microbit.display.set_pixel`(*x, y, value*)

Set the brightness of the LED at column `x` and row `y` to `value`, which has to be an integer between 0 and 9.

`microbit.display.clear`()

Set the brightness of all LEDs to 0 (off).

`microbit.display.show`(*image*)

Display the `image`.

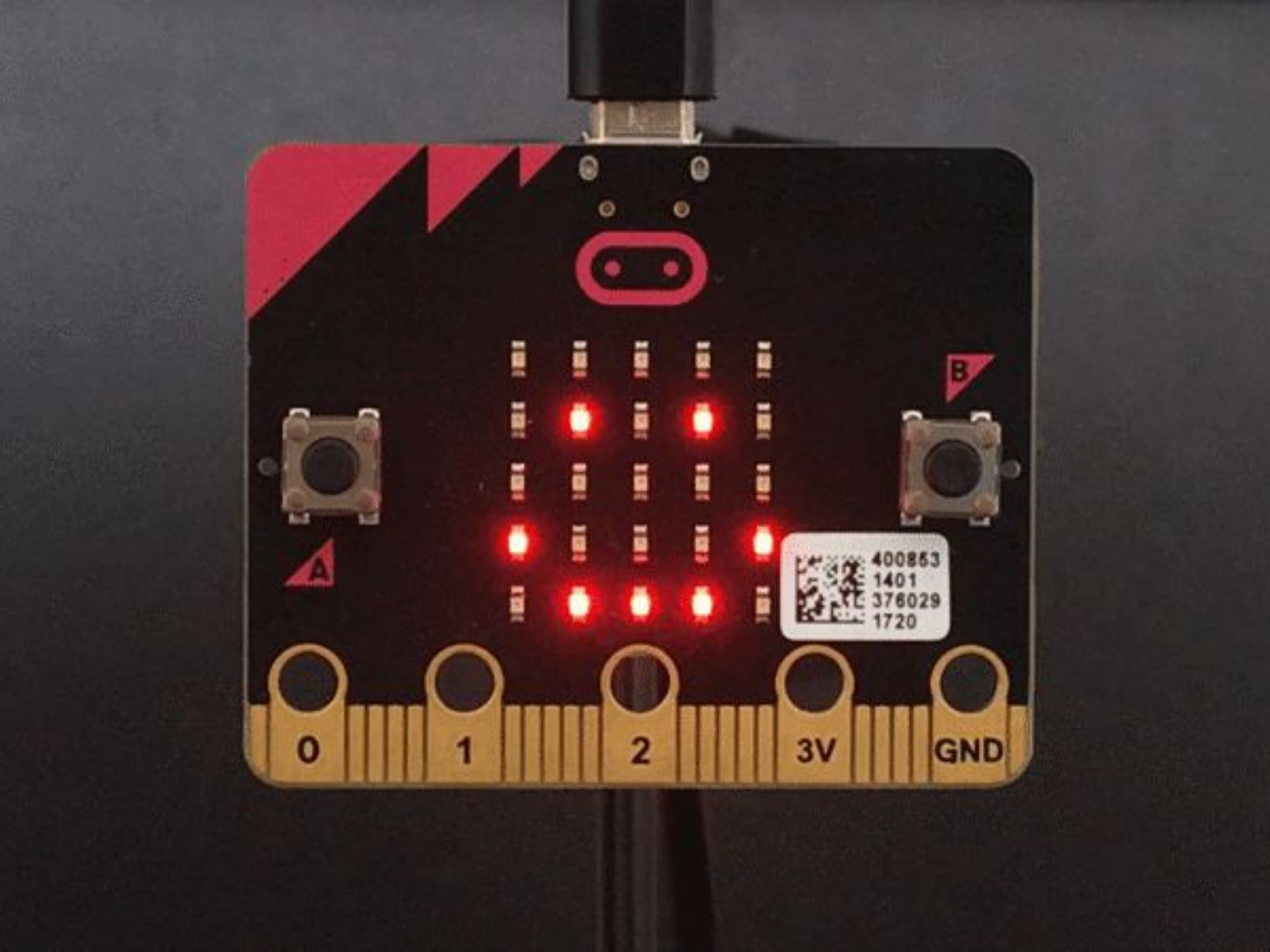
What are the similarities between the functions?

What are the differences between the functions?

Guided

20 min

Name Badge


B
A
400853
1401
376029
1720
0
1
2
3V
GND

Guided

Programme the micro:bit

Display a smiley face and the user's name by repeating this 3 times. Then clear the display

Guided

python editor for microbit

Python Editor for micro:bit

https://python.microbit.org ▾

Welcome to the **micro:bit Python Editor** version 2. Learn more about this update. × ... Add your **Python** code here. E.g.. from **microbit** import *. while True:.

MicroPython Guide

The MicroPython guide to BBC micro:bit. Python is a ...

Flashing micro:bit

Flashing micro:bit ...

Python

Help · Support · Issue Tracker. Editor Version: 1.1.5 ...

Help

The version of Python that runs on the BBC micro:bit is called ...

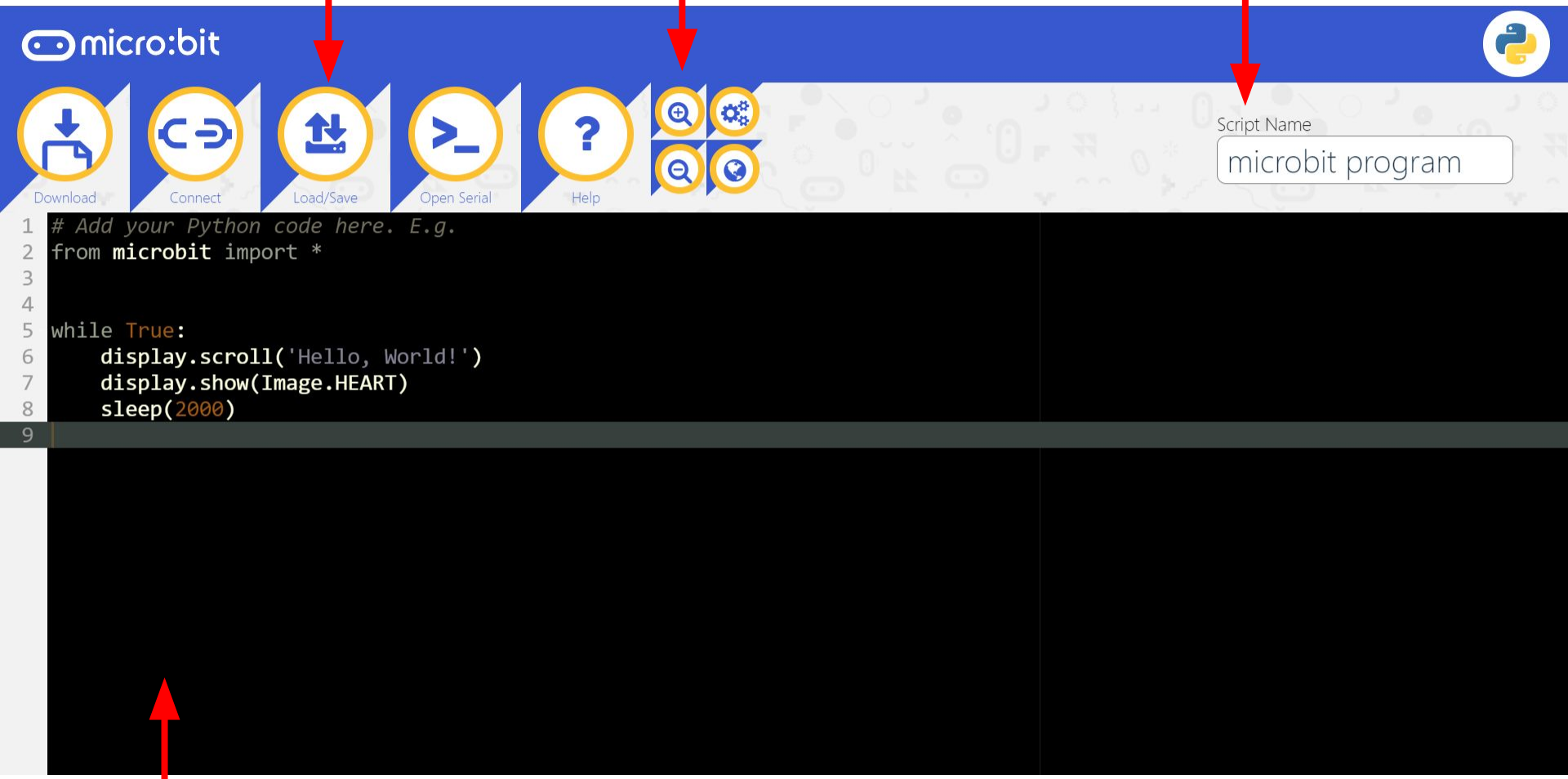
More results from microbit.org »

Guided

Click this to save programme onto micro:bit or Desktop

Zoom In
Zoom Out

Modify programme name here



Type your code here

Guided

For Loop

Repeats a **specific number** of times

While Loop

Repeats **until the condition is False**

Which loop should we use to make the program repeat 3 times?

For Loop!

```
for i in range(3):
```

Find the Function to show the user's name and an image

Guided

Show

`microbit.display.show`(*image*)

Display the `image`.

`microbit.display.show`(*value, delay=400, *, wait=True, loop=False, clear=False*)

If `value` is a string, float or integer, display letters/digits in sequence. Otherwise, if `value` is an iterable sequence of images, display these images in sequence. Each letter, digit or image is shown with `delay` milliseconds between them.

If `wait` is `True`, this function will block until the animation is finished, otherwise the animation will happen in the background.

If `loop` is `True`, the animation will repeat forever.

If `clear` is `True`, the display will be cleared after the iterable has finished.

Note that the `wait`, `loop` and `clear` arguments must be specified using their keyword.

Guided

Scroll

`microbit.display.scroll`*(value, delay=150, *, wait=True, loop=False, monospace=False)*

Scrolls `value` horizontally on the display. If `value` is an integer or float it is first converted to a string using `str()`. The `delay` parameter controls how fast the text is scrolling.

If `wait` is `True`, this function will block until the animation is finished, otherwise the animation will happen in the background.

If `loop` is `True`, the animation will repeat forever.

If `monospace` is `True`, the characters will all take up 5 pixel-columns in width, otherwise there will be exactly 1 blank pixel-column between each character as they scroll.

Note that the `wait`, `loop` and `monospace` arguments must be specified using their keyword.

🔍 Find the Image documentation under API Reference on the left section

Guided

microbit-micropython.readthedocs.io/en/latest/image.html

- Audio
- Bluetooth
- Buttons
- Compass
- Display
- Local Persistent File System
- I²C
- Image
- Machine
- MicroPython
- Music
- NeoPixel
- The os Module
- Input/Output Pins
- Radio
- Random Number Generation
- Speech
- SPI
- UART

Read the Docs v: latest

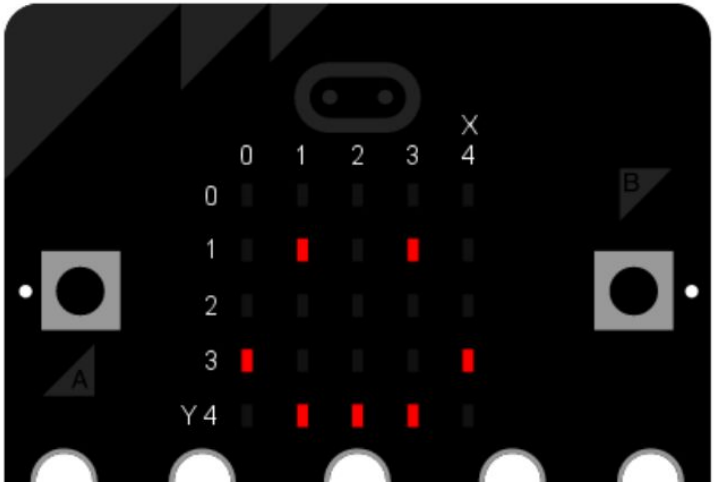
Docs » Microbit Module » Image

Edit on GitHub

Image

The `Image` class is used to create images that can be displayed easily on the device's LED matrix. Given an image object it's possible to display it via the `display` API:

```
display.show(Image.HAPPY)
```



🔍 Find the list of images available

Guided

Attributes

The `Image` class also has the following built-in instances of itself included as its attributes (the attribute names indicate what the image represents):

- `Image.HEART`
- `Image.HEART_SMALL`
- `Image.HAPPY`
- `Image.SMILE`
- `Image.SAD`
- `Image.CONFUSED`
- `Image.ANGRY`
- `Image.ASLEEP`
- `Image.SURPRISED`
- `Image.SILLY`
- `Image.FABULOUS`

Remember to capitalize the image names!

Find the Function to clear the microbit display

Guided

Clear the microbit display

```
microbit.display.clear()
```

Set the brightness of all LEDs to 0 (off).

Guided

```
from microbit import *

for i in range(3):
    display.show(Image.HAPPY)
    display.scroll('Name')
```

Guided

Indentation -Repeat the code

```
from microbit import *

for i in range(3):
    display.show(Image.HAPPY)
    display.scroll('Name')
```

Guided

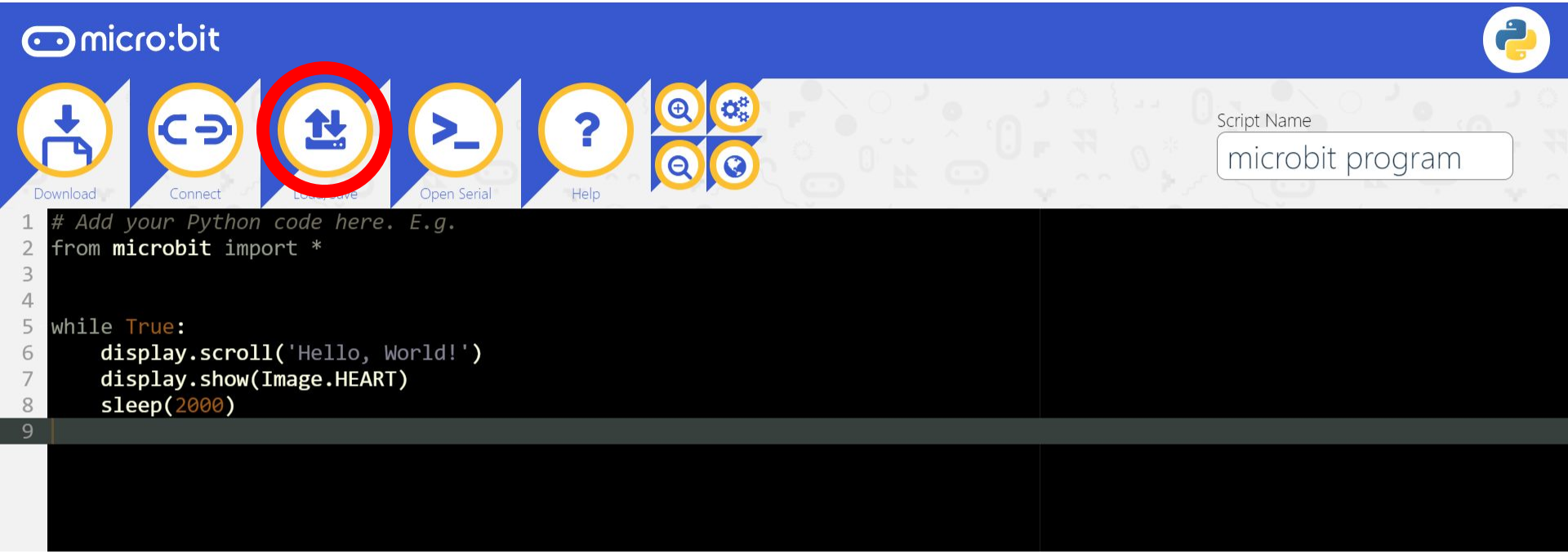
No Indentation -Clearing screen

```
from microbit import *

for i in range(3):
    display.show(Image.HAPPY)
    display.scroll('Name')

display.clear()
```

Save the programme onto the **micro:bit**



Download **Project Hex** onto the micro:bit

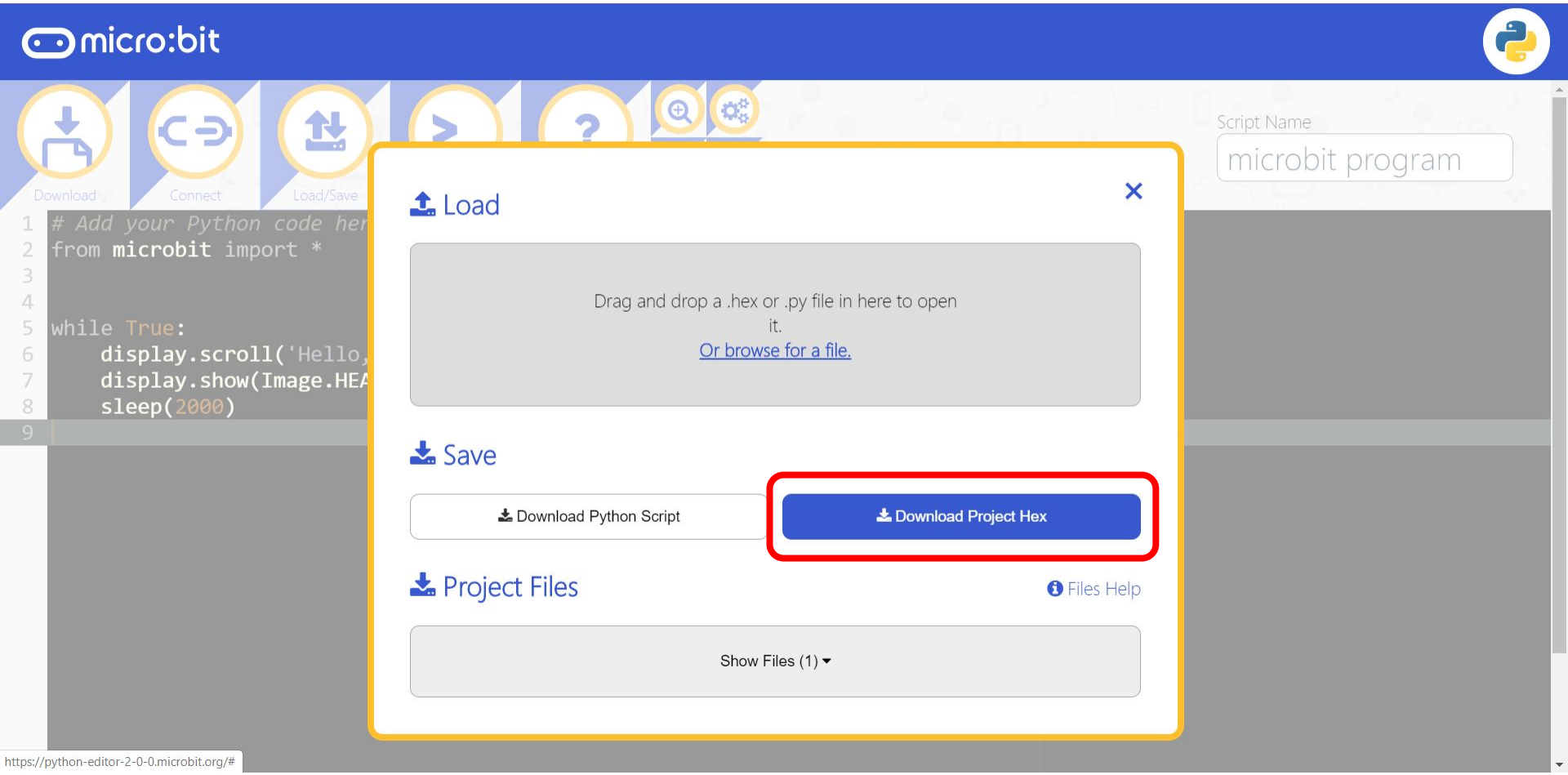


Image is not Appearing

Why?

Guided

Sleep

`microbit.sleep(n)`

Wait for `n` milliseconds. One second is 1000 milliseconds, so:

```
microbit.sleep(1000)
```

will pause the execution for one second. `n` can be an integer or a floating point number.

How to Correct the Code

```
sleep()
```

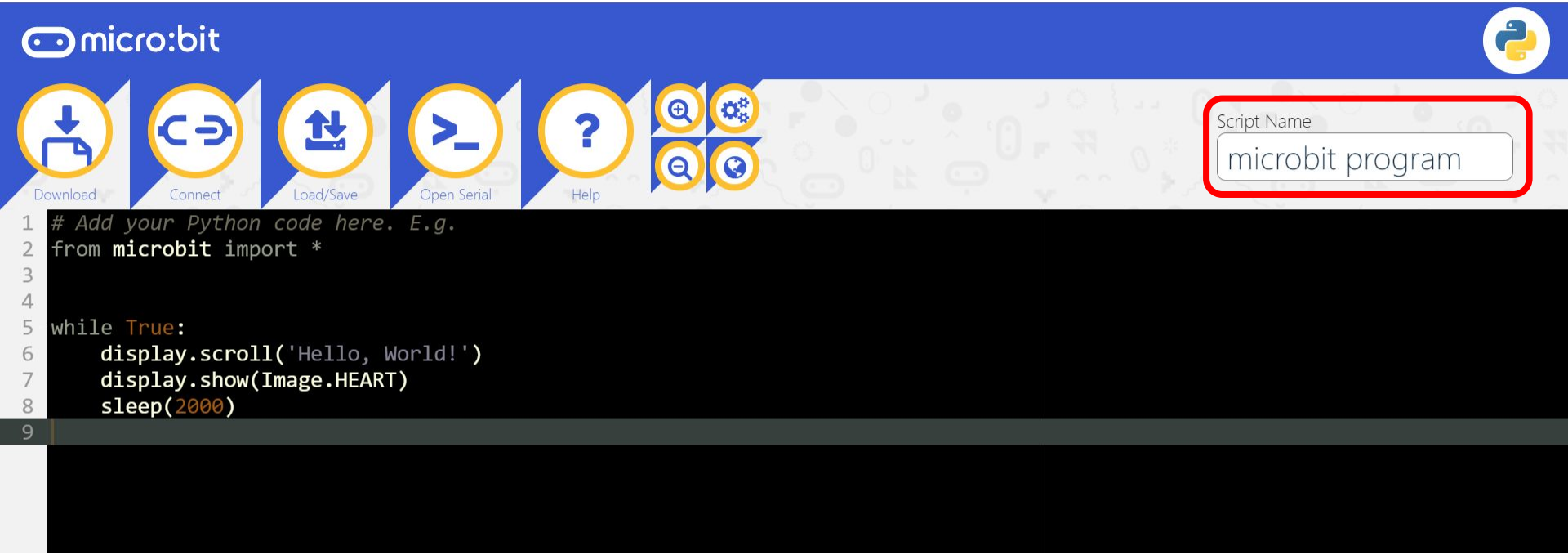
Why isn't `sleep()` added after the name scrolls across the screen?

Bonus

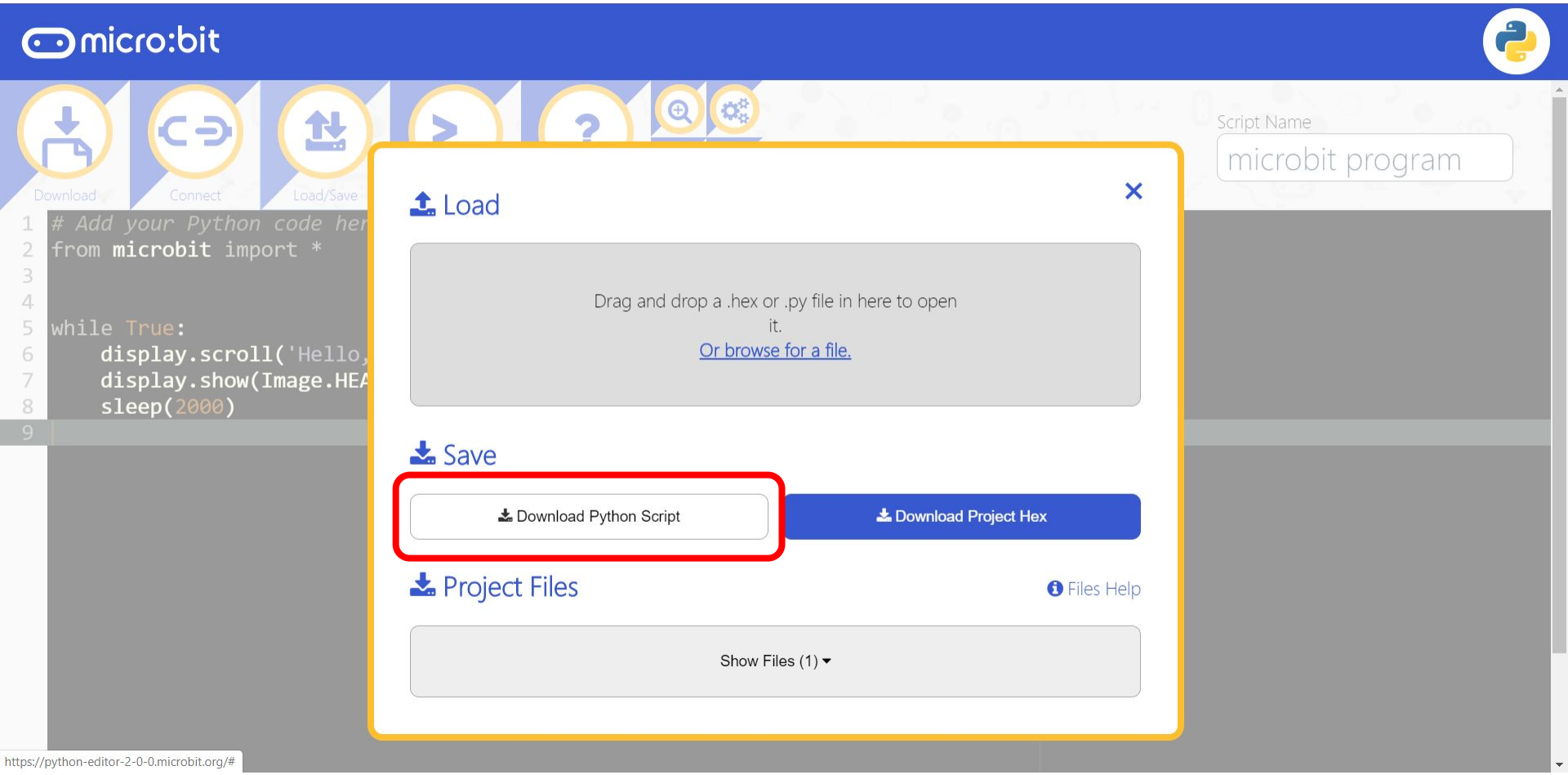
Display.scroll()

The `display.scroll()` function will block the next code until the animation is finished.

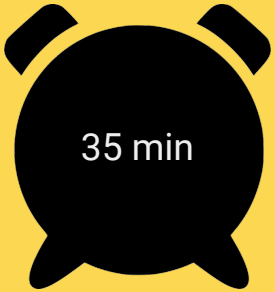
Give your project a name and save it to the **Desktop**



Download **Python Script** onto the Desktop



Break

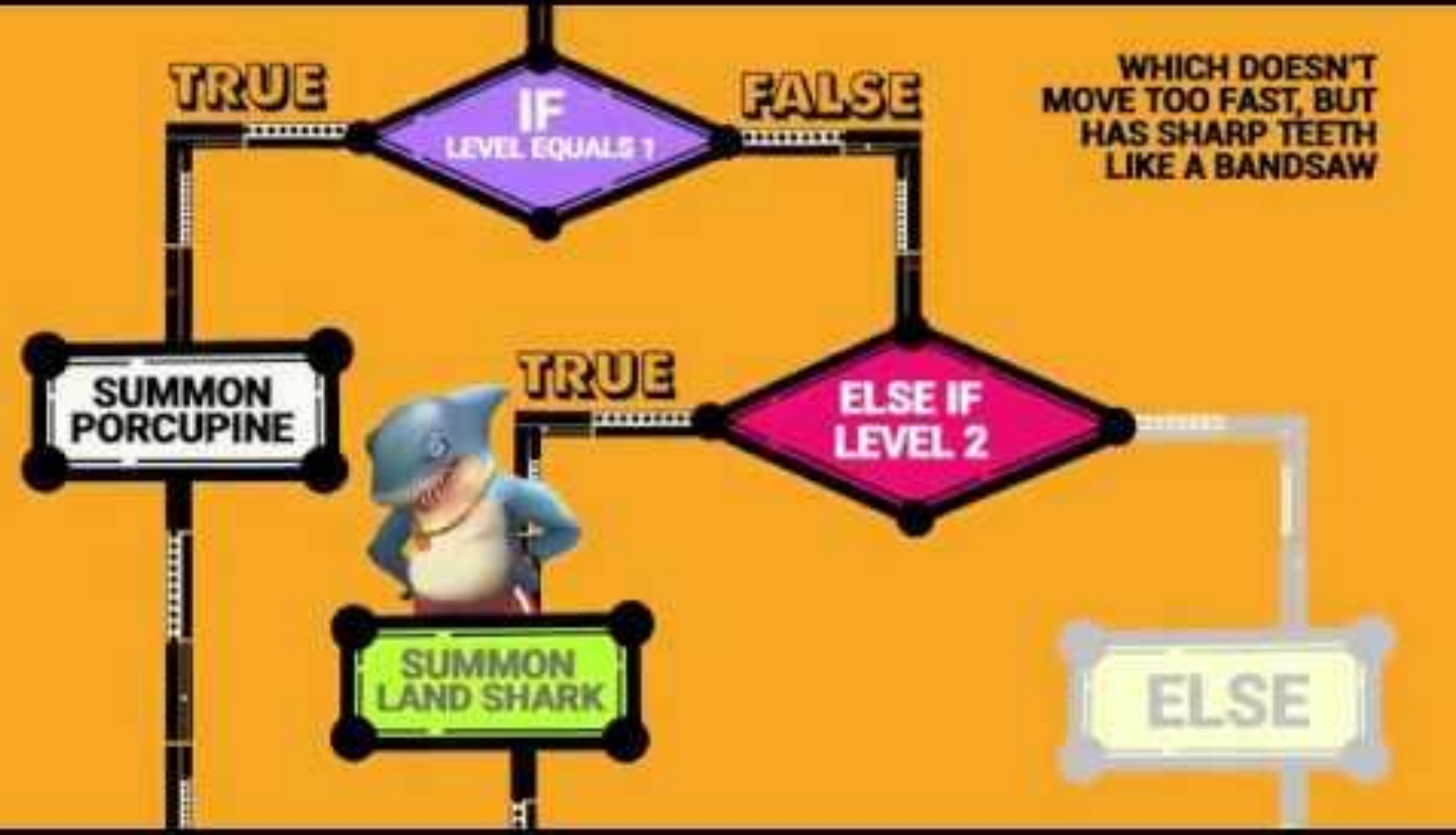


Buttons & Conditionals

Unplugged

10 min

Recalling conditionals

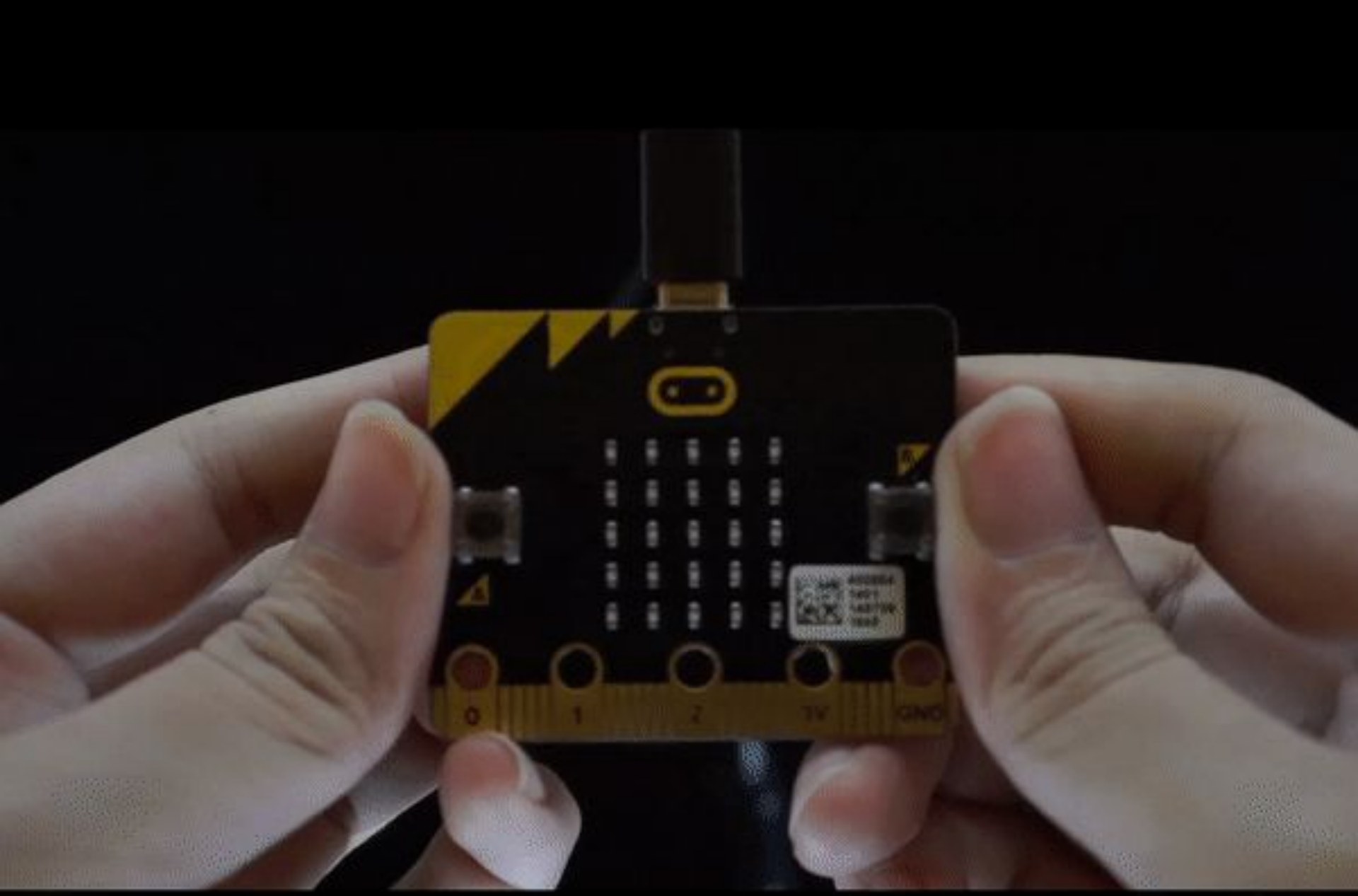

TRUE
IF
LEVEL EQUALS 1
FALSE
WHICH DOESN'T MOVE TOO FAST, BUT HAS SHARP TEETH LIKE A BANDSAW
SUMMON PORCUPINE
TRUE
ELSE IF LEVEL 2
SUMMON LAND SHARK
ELSE

What are some examples of conditionals in your daily life?

Guided

25 min

micro:bit Counter



Guided

Programme the micro:bit

Display a number on the screen.

When button A is pressed, the number increases by 1. When button B is pressed, the number decreases by 1.

Guided

Do you remember.....

For Loop

Repeats a **specific number** of times

While Loop

Repeats **until the condition is False**

Which loop should we use to make the program repeat forever?

While Loop!

Use a condition that can never be **False**,
such as `1==1` or `True`

Guided

To make it easier for users to read, use

`display.scroll()`

instead of **`display.show()`**

Find the Button documentation under API Reference on the left section

Guided

microbit-micropython.readthedocs.io/en/latest/button.html

Speech
Network
Radio
Next Steps
API REFERENCE
micro:bit Micropython API
Microbit Module
Accelerometer
Audio
Bluetooth
Buttons
Compass
Display
Local Persistent File System
I²C
Image
Machine
MicroPython

Buttons

There are two buttons on the board, called `button_a` and `button_b`.

Attributes

button_a

A `Button` instance (see below) representing the left button.

button_b

Represents the right button.

Classes

class **Button**

Represents a button.

Guided

Classes

class **Button**

Represents a button.

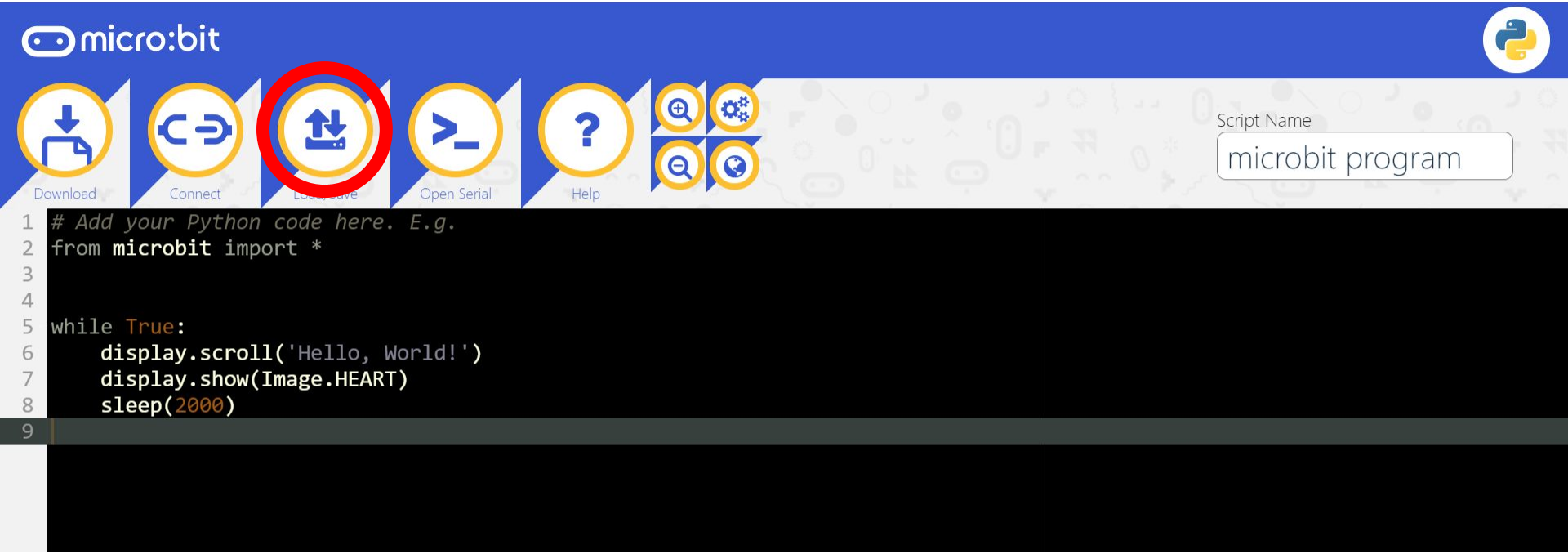
Note

This class is not actually available to the user, it is only used by the two button instances, which are provided already initialized.

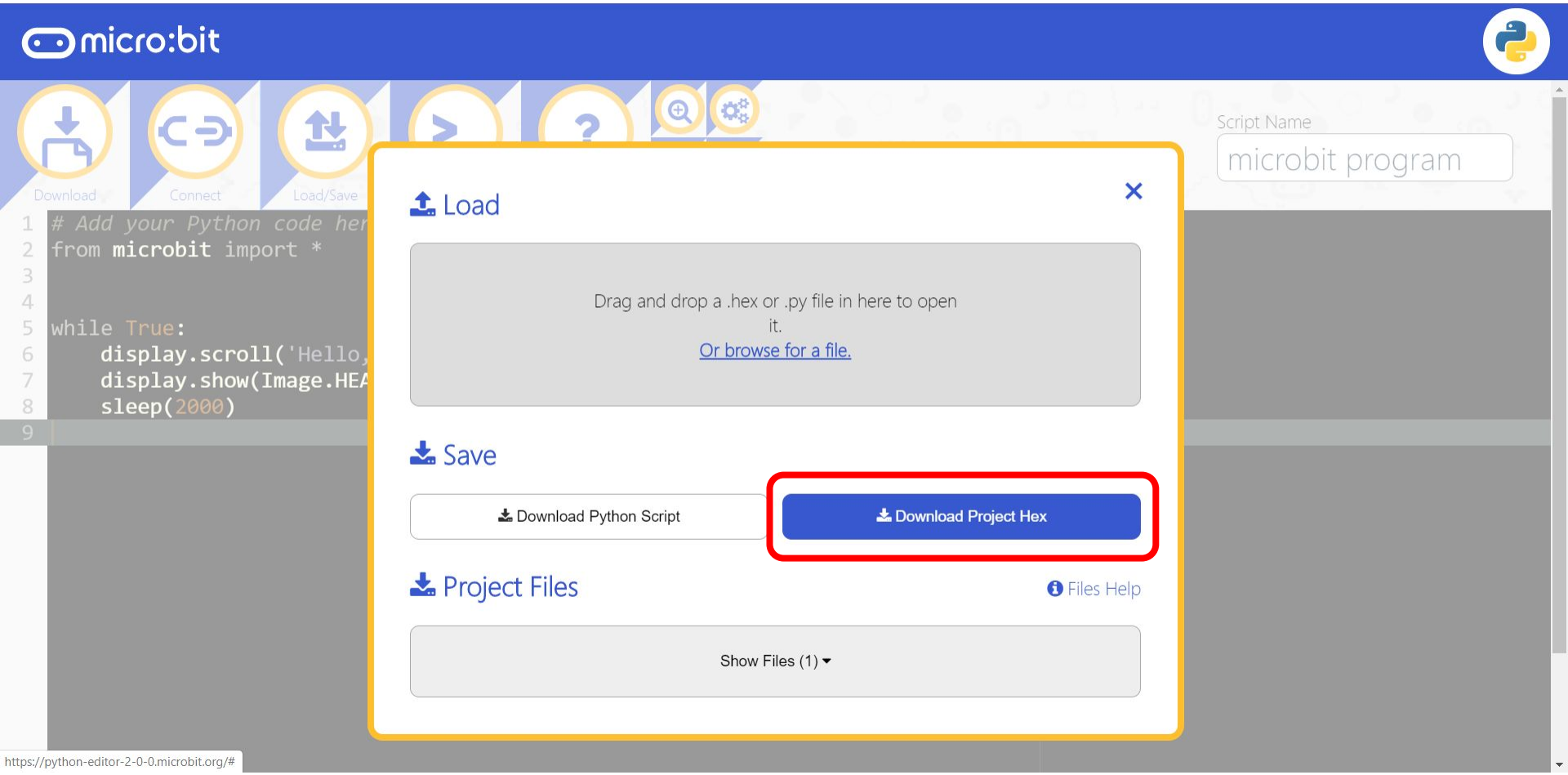
is_pressed()

Returns `True` if the specified button `button` is currently being held down, and `False` otherwise.

Save the programme onto the **micro:bit**



Download **Project Hex** onto the micro:bit



NameError

Line 6 NameError name 'number' is not defined

Guided

Variables

Variables allow us to store values in the computer’s memory.

In a game, we use a variable to store the character's name and level.

```
name = 'Pikachu'
level = 2
```

Guided

How to Correct the Code

```
from microbit import *

number = 0

while True:
```

Download the Project onto the micro:bit

Cannot Reset

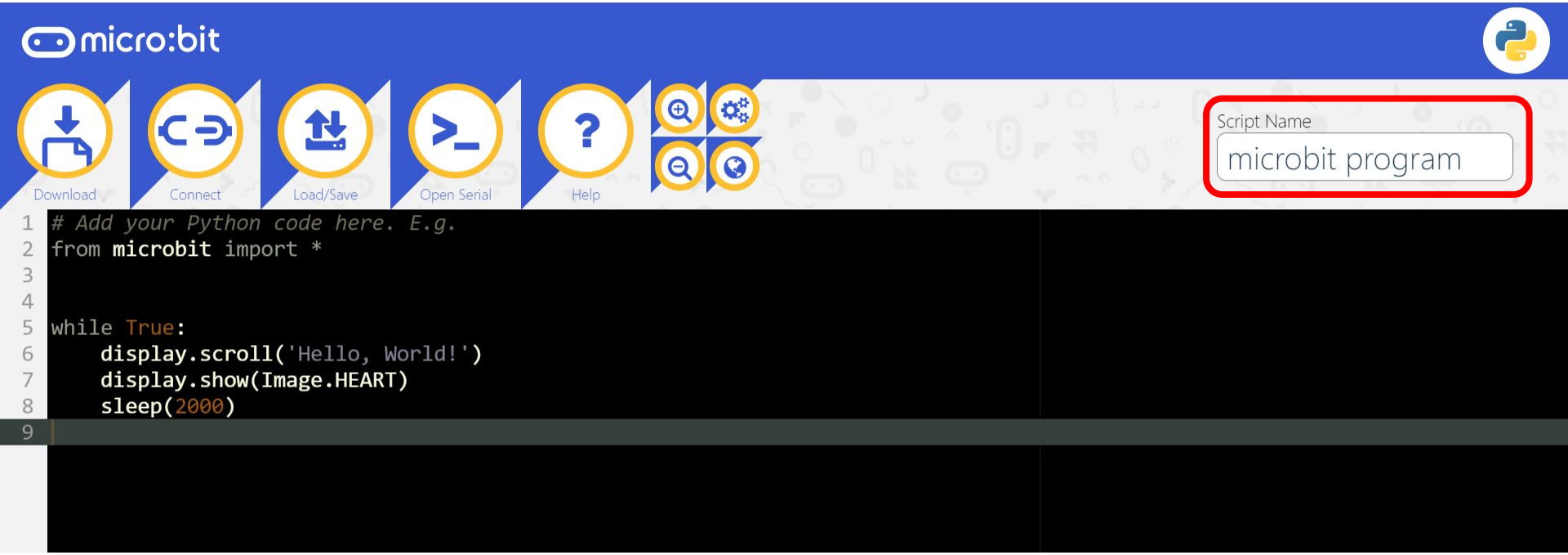
When button A and B are pressed, number decreases/increases but does not reset.

Guided

How to Correct the Code

```
if button_a.is_pressed() and button_b.is_pressed():
        number = 0
elif button_a.is_pressed():
        number -= 1
elif button_b.is_pressed():
        number += 1
```

Give your project a name and save it to the **Desktop**



Download **Python Script** onto the Desktop