

MicroBit lesson 1: Introduction

LO: To program your MicroBit using the Mu editor.

Today we're going to start using our **BBC MicroBits**, learning how to use an app called **Mu** to program them on our computers, and learn a bit about the **Python** programming language too. ***Tick off each step as you go.***

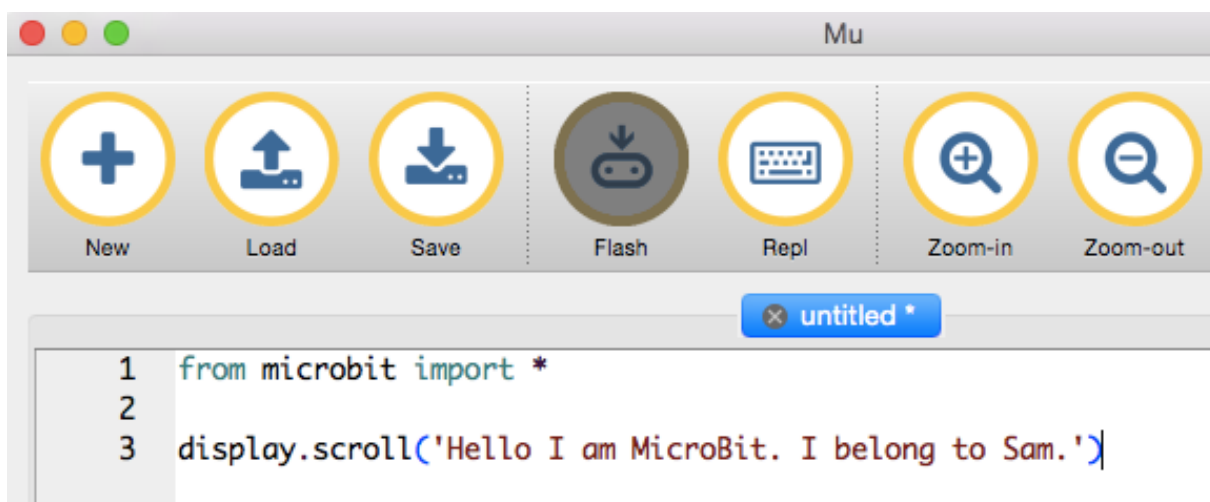
☐ Step 1: Open Mu

Open Mu on your computer (All Programs > Microbit > Mu) and start typing this code in. Use your own name instead of Sam (unless your name **is** Sam!).



☐ Step 2: Send the code to the MicroBit.

Plug your MicroBit in to one of the spare USB sockets on your computer. You may see a message asking if you want to allow your computer to access it. You should choose 'allow read and write access'. Click on the **'Flash'** button on the Mu toolbar:

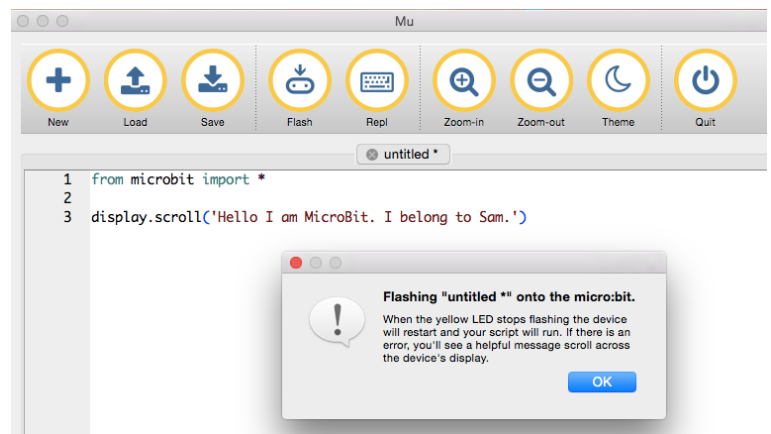


A few things should happen:

- The yellow light on the back of your MicroBit should flash.
- You should see a message pop up on your computer screen.

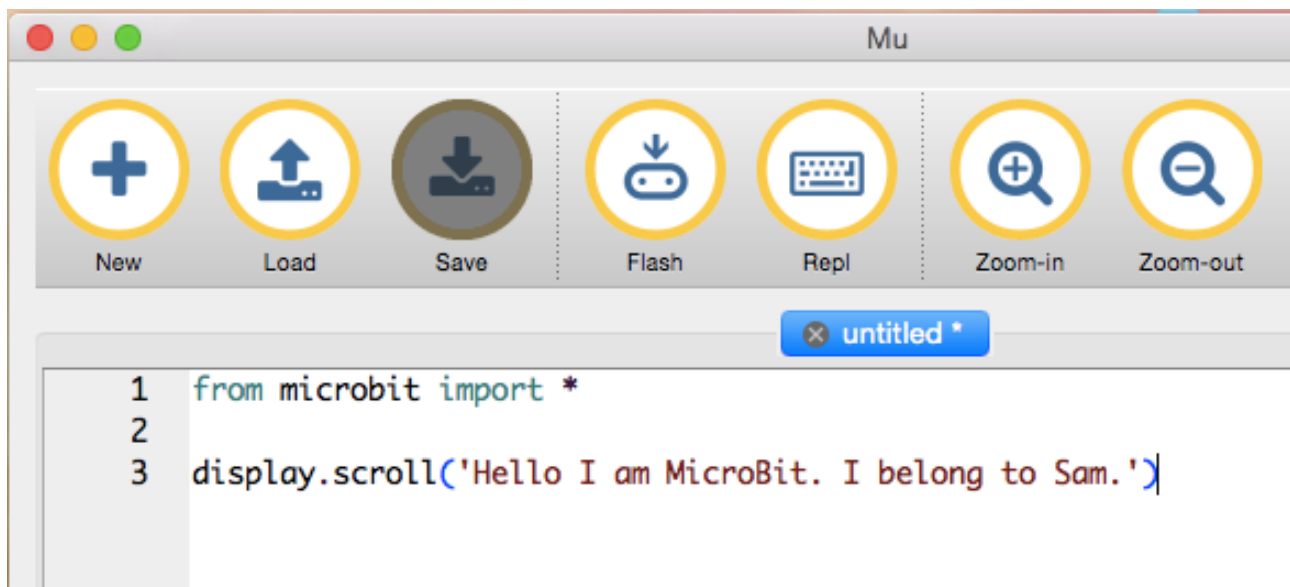


- Most importantly, you should see your message scrolling across the red LED lights on your MicroBit!



☐ Step 3: Save your code.

Unlike the IDLE editor we used before, Mu doesn't force you to save your code every time you run it, so let's save it now. Click on the **Save** button and give the file a useful name like **Sam-Microbit1**



Congratulations - you have now programmed your BBC MicroBit using Python!

Step 4: Keeping going.

So far your program stops running as soon as the message is displayed. Let's keep it going forever (or until you unplug the MicroBit).

The **while True:** loop in Python is really useful - it's very much like the 'Forever' block in Scratch. Edit your code so it looks like this, then press **Flash** to send it to the MicroBit:

```
from microbit import *  
  
while True:  
    display.scroll('Hello I am MicroBit. I belong to Sam.')
```

Make sure you have **four spaces** at the start of the 'display.scroll' line and spell **True** with a capital letter T.



Python uses these spaces - called **indents** - to tell it which lines of code belong together, such as in a loop. Any instructions you put after **while.True:** with 4 spaces at the start will run forever.

You should see your message now keeps scrolling across the MicroBit.

Step 5: Pictures.

Scrolling text was ok, but let's get drawing! You can make your own shapes (which we'll look at in another lesson), but MicroBit Python also has lots of pictures built in. Edit your code so it looks like this, then send it to the Microbit with the Flash button:

```
from microbit import *  
  
while True:  
    display.show(Image.HAPPY)
```

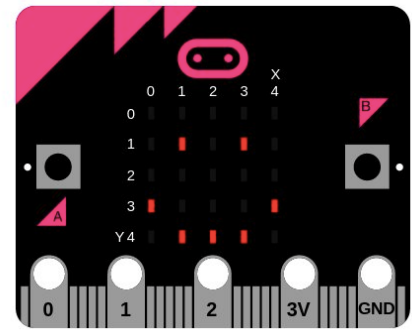
Then try changing the picture. Try SAD, HEART, MEH, RABBIT, COW, DUCK, SWORD, GIRAFFE, TARGET, HOUSE, TORTOISE, UMBRELLA, SNAKE, SKULL, BUTTERFLY, TSHIRT, XMAS, STICKFIGURE, MUSIC_CROCHET, CHESSBOARD, YES, NO, CONFUSED, ANGRY, ARROW_N, ARROW_S, ARROW_E, ARROW_W, ARROW_SE

☐ Step 6: using buttons.

Your MicroBit has two **buttons** on the front called A and B - and we can use them in our Python programs to make things happen:

```
from microbit import *

while True:
    if button_a.is_pressed():
        display.show(Image.PACMAN)
    if button_b.is_pressed():
        display.show(Image.GHOST)
```



Be careful with your **indents**! Some lines have 4 spaces at the start, some have 8 - and some have no spaces at all.

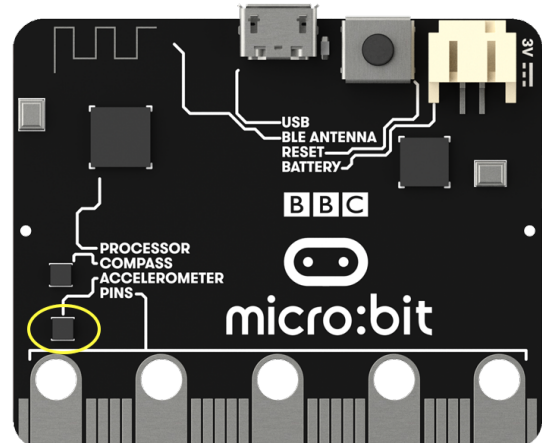
See if you can make different pictures appear when you press the left and right buttons.

☐ Step 7: Shake it!

Your MicroBit also contains an **accelerometer**. This is the same chip your phone or tablet has so it knows which way up you are holding it - or if you shake it. Add a line of code to show another picture when you shake your MicroBit:

```
from microbit import *

while True:
    if button_a.is_pressed():
        display.show(Image.PACMAN)
    if button_b.is_pressed():
        display.show(Image.GHOST)
    if accelerometer.was_gesture('shake'):
        display.show(Image.SURPRISED)
```



☐ Step 8: Touch it!

Your MicroBit also has touch sensors that work like a MakeyMakey - we can use them in our code too. Add the new line, flash it to the MicroBit then see what happens if you touch the gold number 1 on the bottom of the MicroBit with one finger and the GND label with another:

```
from microbit import *

while True:
    if button_a.is_pressed():
        display.show(Image.PACMAN)
    if button_b.is_pressed():
        display.show(Image.GHOST)
    if accelerometer.was_gesture('shake'):
        display.show(Image.SURPRISED)
    if pin1.is_touched():
        display.show(Image.HEART)
```

☐ Step 9: Free code!

See what you can create - mix up button pushes, touch sensors, pictures and scrolling messages to make your own machine that does whatever you want.

☐ Step 10: Review

Q1. What Python command puts a picture on the MicroBit screen?

Q2. How do we make sure our code keeps running forever rather than just once?

Q3. What command puts scrolling text on the display?

Q4. How do we tell if a button has been pushed?