BBC Micro: bits – Chase the dot

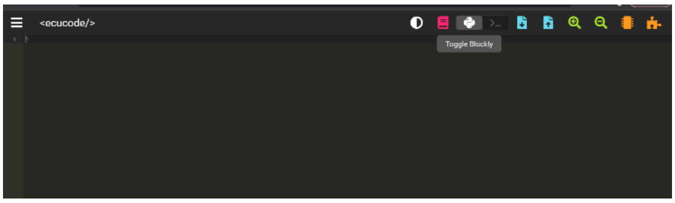
ECU Makerspace Group

Written by Georgia Maher and Julia Szymanski

Getting started

Let’s do a practice run just using the Micro: bit LED’s. Once you have access to the computers, follow through this guide. If your computer is not on the makerspace website then use this link to gain access. **Please use Google Chrome to access this website (makerspace.ecu.edu.au/code)**

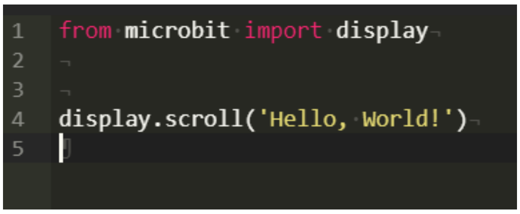
Click on **‘Toggle Blockly’** to switch to Python code.



Task 1 – Print your name!:

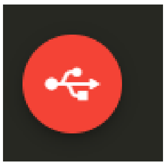
Step One – Print your name:

Click on to the first line and import ‘display’ from ‘microbit’, then display a message using display.scroll(“ “).

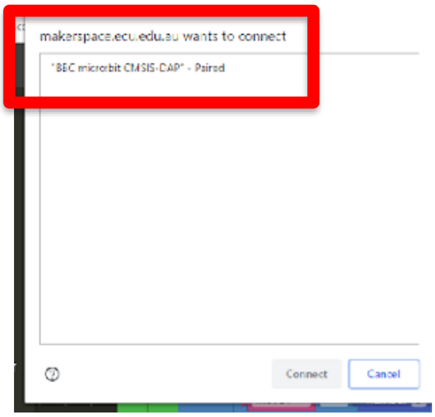


```
from microbit import display


display.scroll('Hello, World!')
```

Step Two - Now that you have the scroll message on the page, write your name and upload it to the Micro: Bit. Do this by following these instructions:



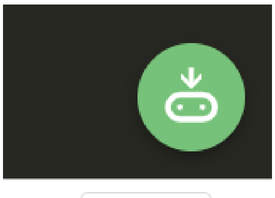
Click on the red button then a box comes up inviting you to pair the micro: bit. Click on the “BBC Micro: bit CMSIS-DAP”

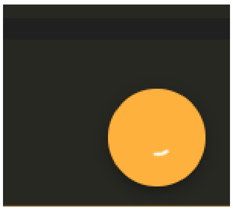


makerspace.ecu.edu.au wants to connect

"BBC micro:bit CMSIS-DAP" - Paired

Once this has been completed click on the green button to download. An orange button will appear as it is downloading.

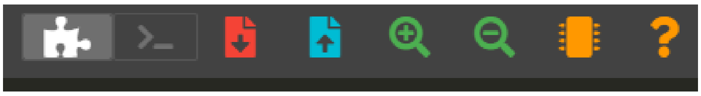




Congratulations! Remember to delete this scroll message before starting the next section.

Troubleshooting:

Updating the firmware on the Micro: bit

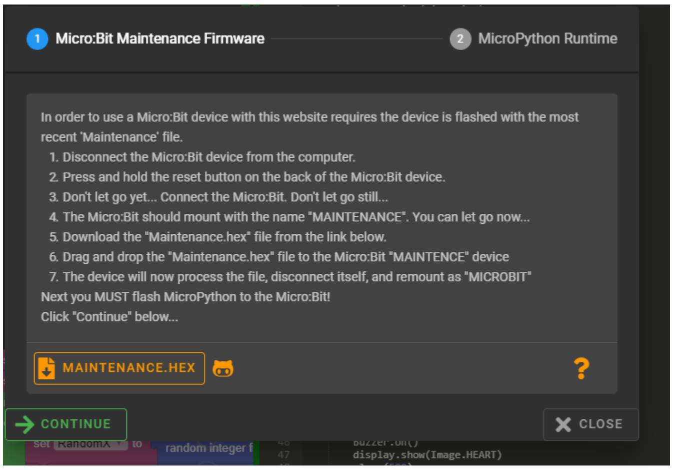


Click on the orange shape which appears ‘update micro: bit’. Follow set instructions on the **Maintenance Firmware** guide.

BBC Micro: bits – Chase the dot

ECU Makerspace Group

Written by Georgia Maher and Julia Szymanski



In order to use a Micro:Bit device with this website requires the device is flashed with the most recent 'Maintenance' file.

1. Disconnect the Micro:Bit device from the computer.
2. Press and hold the reset button on the back of the Micro:Bit device.
3. Don't let go yet... Connect the Micro:Bit. Don't let go still...
4. The Micro:Bit should mount with the name "MAINTENANCE". You can let go now...
5. Download the "Maintenance.hex" file from the link below.
6. Drag and drop the "Maintenance.hex" file to the Micro:Bit "MAINTENCE" device
7. The device will now process the file, disconnect itself, and remount as "MICROBIT"

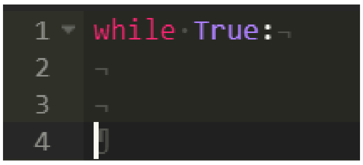
Next you MUST flash MicroPython to the Micro:Bit!

Click "Continue" below...

If the micro: bit appears not to be responding. Unplug the micro: bit, press the reset button at the back of the micro: bit and repeat this process. If the micro: bit continues to fail, swap it for another one!

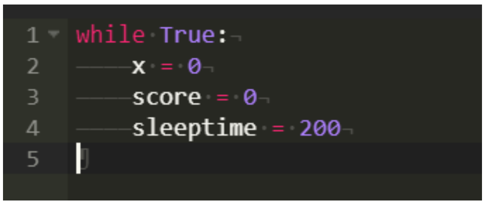
Now let’s learn how to write a program so we can chase the dot.

First, we need is a **While loop**. This makes the program **loop** infinitely as long as the Boolean variable remains True.



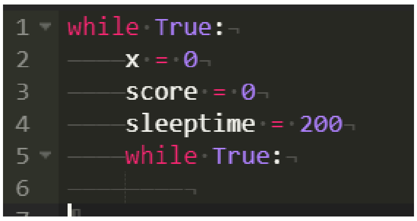
```python
while True:


```

To get our ‘chasing the dot’ working, we need three variables- One for the *score*, one for *sleeptime* and one for *x*.
Inside the while loop, create three variables: score, sleeptime and x.
Now we need to give our variables a value.
Set score and x to 0, and sleeptime to 200.



```python
while True:
    x = 0
    score = 0
    sleeptime = 200

```

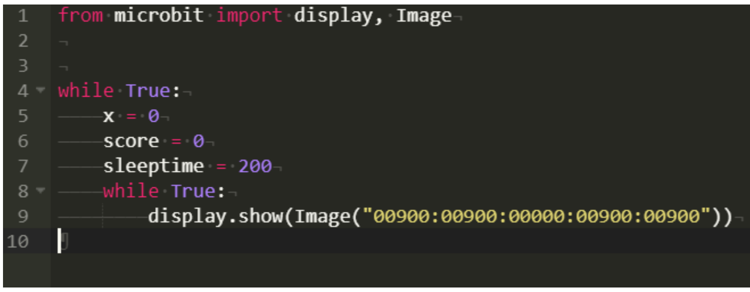
Now we need to create another while loop for our display. Create another while loop within the body of the first while loop, just below where the values were assigned to the three variables.



```python
while True:
    x = 0
    score = 0
    sleeptime = 200
    while True:
        
```

Next, we need to set up the display for the micro: bit and the movement of LED’s.
Use
display.show(Image("00000:00000:00000:00000:00000"))
Each digit stands for one LED. The first five digits stand for the LED’s in the first row, the next set of five digits for the LED’s in the second row, and so on. You can choose numbers between 0 and 9. 0 means the LED will be dark, and 9 sets it to the maximum brightness. In our case, we will set it to “00900:00900:00000:00900:00900".

We also need to import a few things so that we can work with the display, Image, sleep mode and the button.
Just add **from microbit import display, Image, sleep, button_a** above the while loop



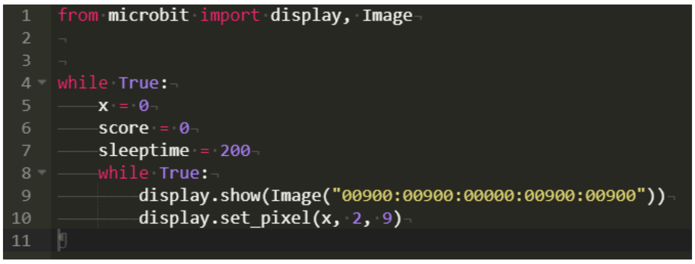
```python
from microbit import display, Image


while True:
    x = 0
    score = 0
    sleeptime = 200
    while True:
        display.show(Image("00900:00900:00000:00900:00900"))

```

Once we have done this, we need to set up our moving pixel. Use display.set_pixel() to set it to X, 2, 9.

BBC Micro: bits – Chase the dot

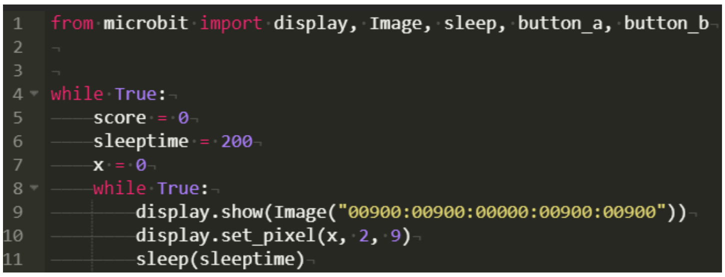
ECU Makerspace Group

Written by Georgia Maher and Julia Szymanski



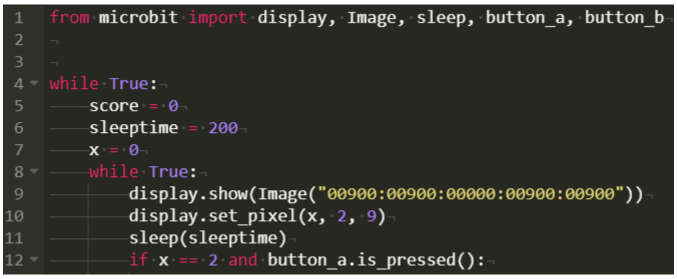
```python
from microbit import display, Image


while True:
    x = 0
    score = 0
    sleeptime = 200
    while True:
        display.show(Image("00900:00900:00000:00900:00900"))
        display.set_pixel(x, 2, 9)

```

Next, let the micro: bit sleep for *sleeptime* milliseconds. Remember, we set set *sleeptime* to 200 earlier.



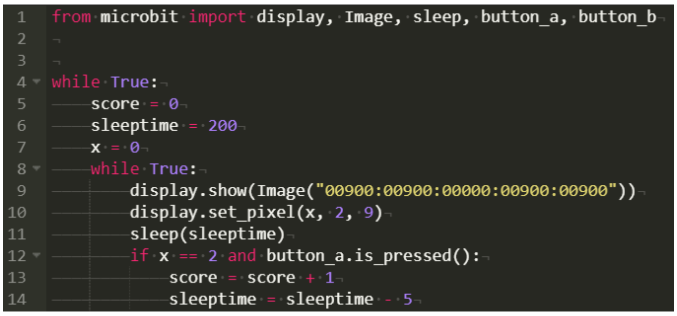
```python
from microbit import display, Image, sleep, button_a, button_b


while True:
    score = 0
    sleeptime = 200
    x = 0
    while True:
        display.show(Image("00900:00900:00000:00900:00900"))
        display.set_pixel(x, 2, 9)
        sleep(sleeptime)
```

In the next step, we need to create an **If** statement. We want the Micro: bit to respond when an input is given. If x = 2 and button A is pressed, then calculate the score.



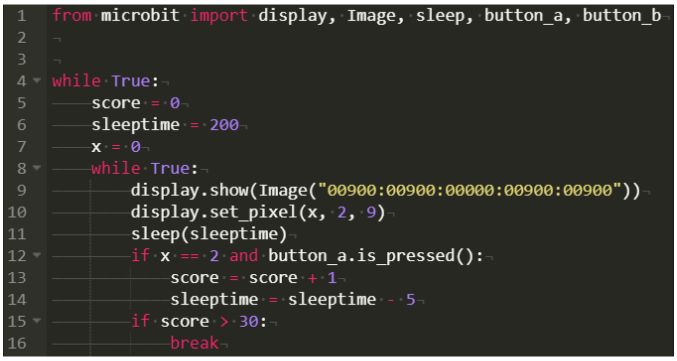
```python
from microbit import display, Image, sleep, button_a, button_b


while True:
    score = 0
    sleeptime = 200
    x = 0
    while True:
        display.show(Image("00900:00900:00000:00900:00900"))
        display.set_pixel(x, 2, 9)
        sleep(sleeptime)
        if x == 2 and button_a.is_pressed():
```

Now it is time to start scoring.
In the if loop that we just created, we set the score to score = score + 1, so that the score increases if x is 2 and button A is pressed.
We then reduce *sleeptime* by 5 ms, so it gets faster each time.



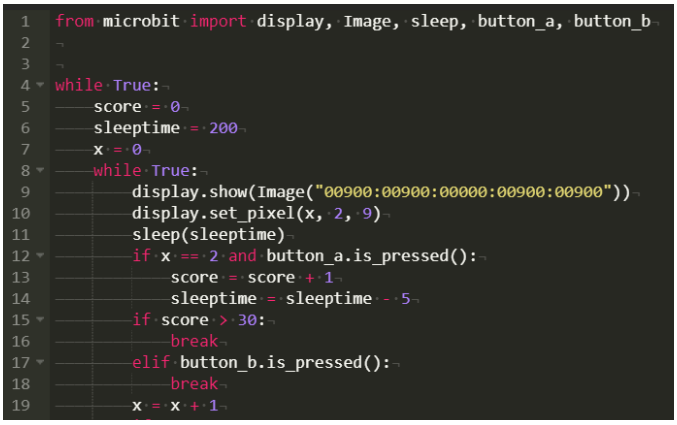
```python
from microbit import display, Image, sleep, button_a, button_b


while True:
    score = 0
    sleeptime = 200
    x = 0
    while True:
        display.show(Image("00900:00900:00000:00900:00900"))
        display.set_pixel(x, 2, 9)
        sleep(sleeptime)
        if x == 2 and button_a.is_pressed():
            score = score + 1
            sleeptime = sleeptime - 5
```

Now we need to turn our **if statement** into an **if - else if** – statement, so that we can get the micro: bit to break out of the loop if button B is pressed. Game over!
If the score is greater than 30, we also wanr the game to end, otherwise it will be endless!



```python
from microbit import display, Image, sleep, button_a, button_b


while True:
    score = 0
    sleeptime = 200
    x = 0
    while True:
        display.show(Image("00900:00900:00000:00900:00900"))
        display.set_pixel(x, 2, 9)
        sleep(sleeptime)
        if x == 2 and button_a.is_pressed():
            score = score + 1
            sleeptime = sleeptime - 5
        if score > 30:
            break
```

Then we need to calculate the score. To do this, we need to **set x** to keep the score.



```python
from microbit import display, Image, sleep, button_a, button_b


while True:
    score = 0
    sleeptime = 200
    x = 0
    while True:
        display.show(Image("00900:00900:00000:00900:00900"))
        display.set_pixel(x, 2, 9)
        sleep(sleeptime)
        if x == 2 and button_a.is_pressed():
            score = score + 1
            sleeptime = sleeptime - 5
        if score > 30:
            break
        elif button_b.is_pressed():
            break
        x = x + 1
```

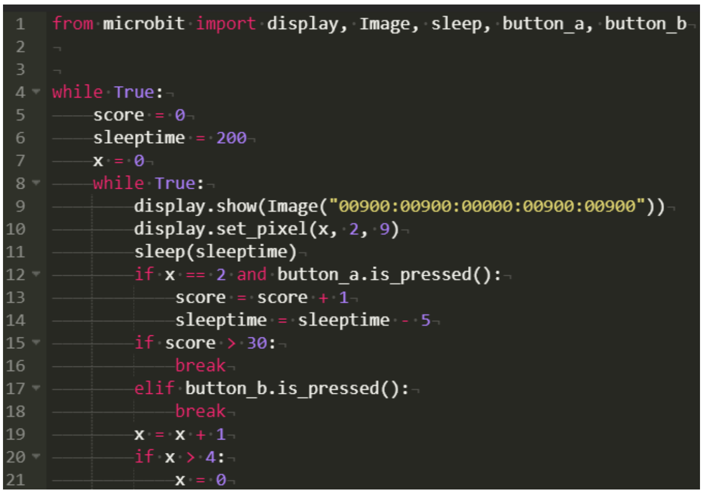
BBC Micro: bits – Chase the dot

ECU Makerspace Group

Written by Georgia Maher and Julia Szymanski

Great work – now we want the micro: bit to show us our scores. To do this, we need to add an if statement just below where we set x to keep the score.

We would like the game to continue 5 times, so once x is greater than 4 we are going to set it back to 0 and display the score. You can change these factors in your own game.



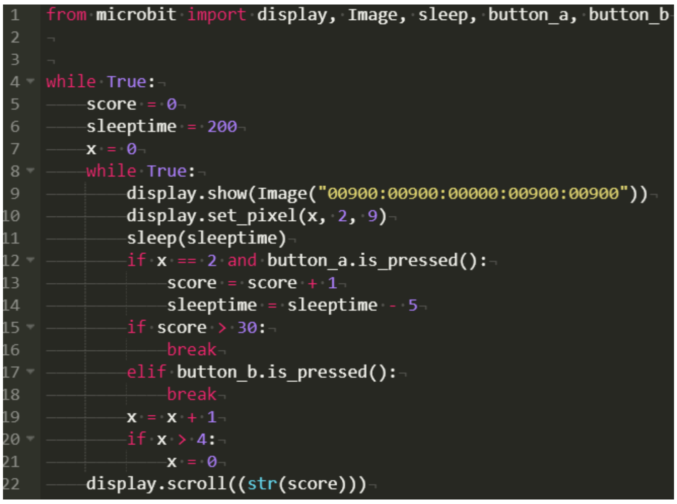
```
from microbit import display, Image, sleep, button_a, button_b


while True:
    score = 0
    sleeptime = 200
    x = 0
    while True:
        display.show(Image("00900:00900:00000:00900:00900"))
        display.set_pixel(x, 2, 9)
        sleep(sleeptime)
        if x == 2 and button_a.is_pressed():
            score = score + 1
            sleeptime = sleeptime - 5
        if score > 30:
            break
        elif button_b.is_pressed():
            break
        x = x + 1
        if x > 4:
            x = 0
```

Finally, we want to scroll the message that has been created from the score variable s. So whilst you are chasing the dot, this is how many you got right!

Use display.scroll() to print the score as a string.

The Code

This is what your code should look like:



```
from microbit import display, Image, sleep, button_a, button_b


while True:
    score = 0
    sleeptime = 200
    x = 0
    while True:
        display.show(Image("00900:00900:00000:00900:00900"))
        display.set_pixel(x, 2, 9)
        sleep(sleeptime)
        if x == 2 and button_a.is_pressed():
            score = score + 1
            sleeptime = sleeptime - 5
        if score > 30:
            break
        elif button_b.is_pressed():
            break
        x = x + 1
        if x > 4:
            x = 0
    display.scroll((str(score)))
```

Congratulations, you have finished your own “Chase the dot” game!