

# Programming the micro:bit

## Drop in session

### Intermediate



Raspberry  
Pi  
Birthday  
Weekend  
2017

#### Before you begin

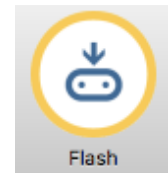
- Take time to read the safety guidance sheet
- Make sure you have Mu loaded from the “Programming” menu. This is a special version of Python for programming the micro:bit
- Plug the micro:bit into the Raspberry Pi using the USB cable

#### Displaying your own images

- Each LED on the micro:bit display can be set to one of ten values (0 to 9). 0 is off and 9 is the brightest setting.
- Enter the following code into Mu:

```
from microbit import *  
pattern = Image("09090:00000:00500:50005:05550")  
display.show(pattern)
```

- Press the “Flash” button to get the program on your device
- If all is well, your program will display, although an error message may be displayed if there’s a bug in your code
- *Challenge* – alter the values to display your own image



#### Animating images

- We can give the impression of movement by switching between images
- Enter the following code into Mu:

```
from microbit import *  
happyface = Image("09090:00000:00500:50005:05550")  
sadface = Image("09090:00000:00500:05550:50005")  
while True:  
    display.show(happyface)  
    sleep(500)  
    display.show(sadface)  
    sleep(500)
```

- Press the “Flash” button to get the program on your device
- *Challenge* – can you make a man appear to walk across the display? Alternatively, create your own animation, which should contain at least three images

## Reacting to movement

- The micro:bit contains an accelerometer, which allows us to detect movement
- Enter the following code into Mu:

```
from microbit import *
while True:
    if accelerometer.get_x() > 0:
        display.show('R')
    else:
        display.show('L')
```

- Press the “Flash” button to get the program on your device
- *Challenge* – can you get more than one event to take place when the device is tilted?
- *Challenge* – can you make a man walk across the display in the direction the micro:bit is tilted?

## Step counter

- *Challenge* – Can you program the micro:bit to act as a step counter? Hint – you could use - *if accelerometer.was\_gesture('shake')*: to detect whether a step has been taken
- Essential features:
  - record number of steps taken
  - display number of steps taken (you might want to do this when a button is pressed. Hint – try using - *if button\_a.is\_pressed()*).
- Desirable features:
  - display motivational messages
  - allow the daily step goal to be set
  - anything else you fancy!