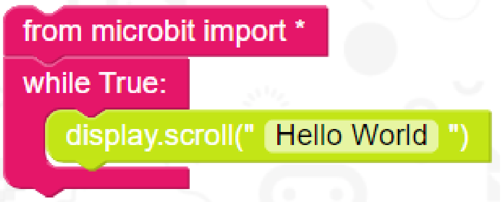
Using the LED screen to display a message

Click on the menu on the right hand `Basic` side, then drop `from microbit import *` into your main coding window.

In the same menu, find: `while True:` and connect it underneath.

This is a loop, the code inside will repeat forever.

Open the `Display` menu, and connect `display.scroll(" Hello World ")` inside your loop, so that your code looks like this:



```
from microbit import *
while True:
    display.scroll(" Hello World ")
```

Change the words inside the speech marks to personalise your message!

Using the LED screen to display a picture

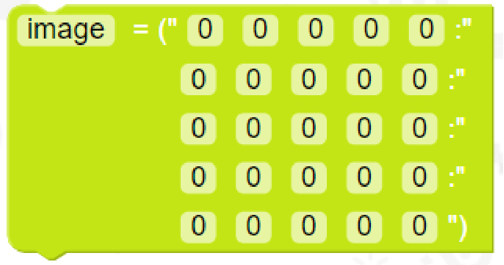
There are two ways of adding a picture to the screen.

`display.show( )` lets you choose from pre-designed images. Inside the brackets, type: `Image.HEART`

It will now look like this: `display.show( Image.HEART )`

Try changing `HEART` to `PACMAN` – look at the list of images to see what else is built in.

You can also design your own images. Use this block:
Each `0` represents one of the micro:bit's front LEDs, change them to `9` for full brightness.



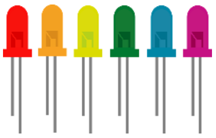
```
image = (" 0 0 0 0 0 :"
          0 0 0 0 0 :"
          0 0 0 0 0 :"
          0 0 0 0 0 :"
          0 0 0 0 0 ")
```

Once you are happy with your image,

connect `display.show( image )` underneath it.

Try both – remember to put them inside your `while True:` loop!

Using the pins to control an LED



To try this, you will need to get two crocodile clips and an LED of any colour.

LEDs only conduct electricity one way, and so must be connected the right way round to light up.

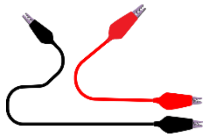
SCIENCEOXFORD

Part 1 – Using Outputs

edublocks

One leg of the LED is longer than the other – this is how you know which way to connect it.

Take one crocodile clip, connect one end to the **short** leg of the LED, the other end to the micro:bit's **GND** pin.



Use the other crocodile clip to connect the **long** leg of the LED to **Pin0** on the micro:bit.

In the `Pins` menu, find `pin 0 .write_digital( )`

Typing `1` in the brackets means on or high. `0` means off or low.

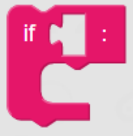
Now make it flash: Connect `sleep( )` to your code, and type `500` in the brackets to set a wait of 500ms (0.5s), then turn the pin **off** `(0)`

Test it, and if it doesn't do what you expect, think carefully about what is happening in your code!

Part 2 – Using Inputs

Using the A and B buttons on the front of the micro:bit

Inside your `while True:` loop, drag:

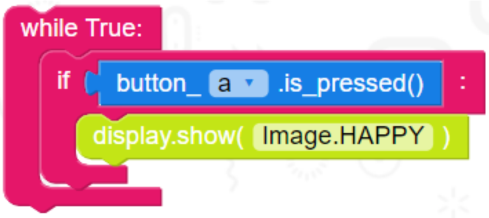


```
if :
```

This is an **if statement**. Connect the block: `button_ a .is_pressed()` after the if.

You are now asking the question – has button A been pressed? Anything you connect inside this, will happen only if the answer to that question **is true**.

For example:

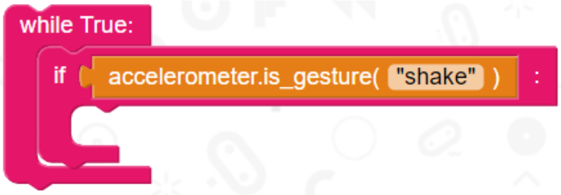


```
while True:
    if button_ a .is_pressed() :
        display.show( Image.HAPPY )
```

Try it for button B!

Using the micro:bit's accelerometer

Inside your `while True:` loop, create the code:



```
while True:
    if accelerometer.is_gesture( "shake" ) :
```

Anything you indent underneath this will not happen unless you **shake** the micro:bit.

The other gestures the micro:bit understands are:
`"up" "down" "left" "right" "face up" "face down" "freefall" "3g" "6g" "8g"`

Try some of them out!