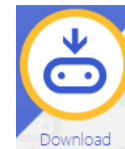# Downloading your code onto the BBC micro:bit

Step 1   Plug the USB cable into your micro:bit, then into the computer.

Step 2
IF YOU ARE USING THE **ONLINE EDITOR** FOLLOW THESE INSTRUCTIONS:

- Click the **download** button on the top toolbar.
- If you are asked where to save the file, save it to the MICROBIT drive.
- If the file downloads automatically, open the folder it is in, then drag-and-drop the file to the MICROBIT drive.
- If you are using Windows, the drive will look similar to this:
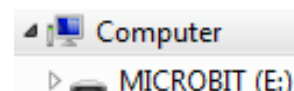
IF YOU ARE USING THE **MU EDITOR** FOLLOW THESE INSTRUCTIONS:
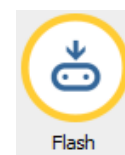
- Click the **flash** button on the top toolbar, this will download automatically.

Step 3   The light on the back of your micro:bit will start flashing yellow. When the flickering stops, your program is ready to go!

Part 2 – Using Inputs

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

Inside your `while True` loop, type: `if button_a.was_pressed():`

This is an **if statement**, and is asking the question – has button A been pressed? Anything you **indent** underneath this, will happen only if the answer to that question **is true**.

For example:
```
while True:
    if button_a.was_pressed():
        display.show(Image.HAPPY)
```

Try it for button B!

Using the micro:bit's accelerometer

Inside your `while True` loop, type: `if accelerometer.was_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!

Using the LED screen to display a message

(If it is not already on your screen) Type: `from microbit import *`

On a new line type: `while True:` This is a loop, telling the code underneath to repeat forever.

On a new line below this, type: `display.scroll("Hello, World!")` This line should be **indented** by either one tab, or four spaces – if it is not already, do this now, so it looks like this:

```
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(Image.HEART)` lets you choose from pre-designed images. Change `HEART` to `PACMAN` – look at the list of images to see what else is built in.

`display.show(Image("00000:00000:00000:00000:00000"))` lets you design your own images. Each `0` represents one of the micro:bit's front LEDs, change them to `9` for full brightness.

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.
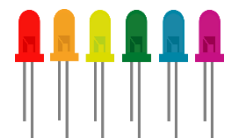
LEDs only conduct electricity one way, and so must be connected the right way round to light up. One leg of the LED is longer than the other – this is how you know which way to connect it.

Take one crocodile clip, connect one end to the **short** leg of the LED, the other end to the micro:bit's **GND** pin.
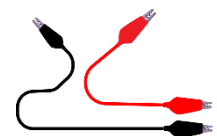
Use the other crocodile clip to connect the **long** leg of the LED to **Pin0** on the micro:bit.

Type `pin0.write_digital(1)` The `1` in the brackets means on or high. `0` means off or low.

Now make it flash: Type `sleep(500)` 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!