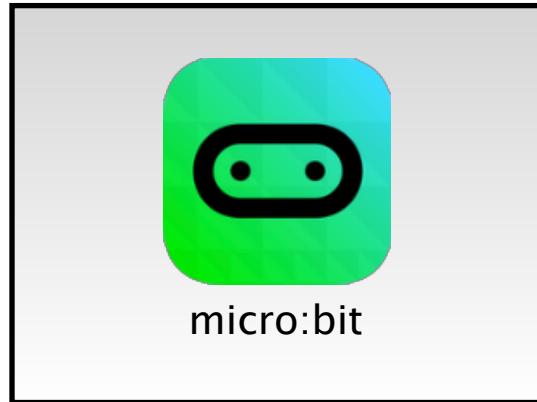
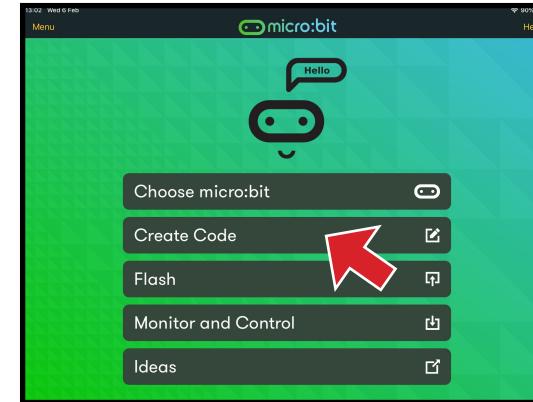


# BOBBING BOTS

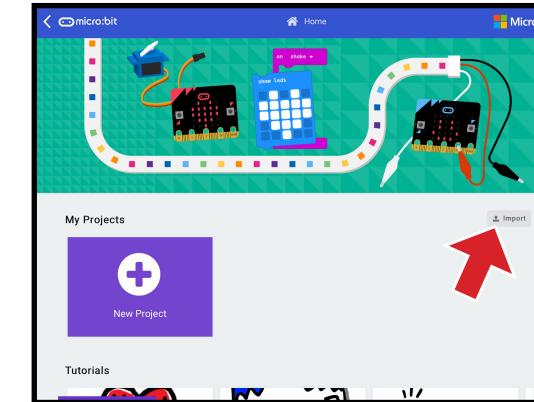
## Programming worksheet



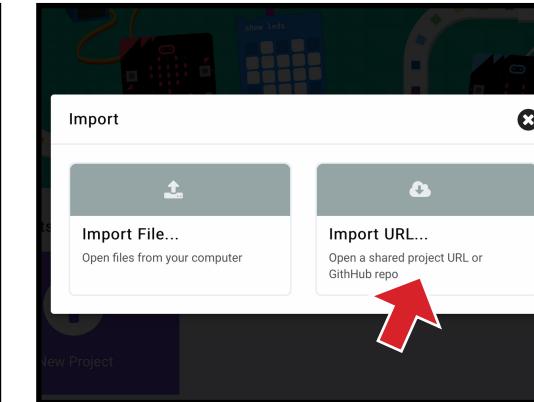
From the iPad home screen, open the **micro:bit** app.



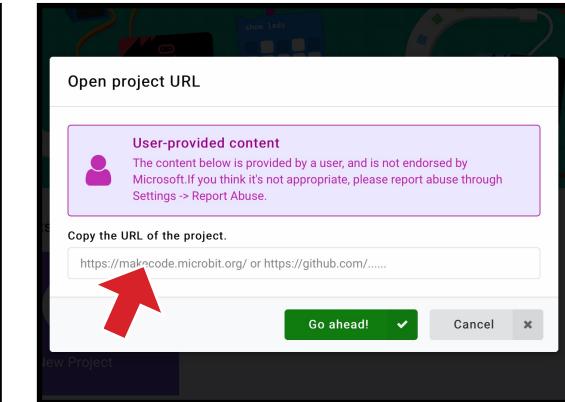
Tap the **Create Code** button from the main menu.



Tap the **Import** button.



Tap the **Import URL...** box.



Tap inside the text entry box and type:

**github:LabCentral/bobbing-bots**

Then tap the **Go ahead!** button.

This will import an example project to get you started.

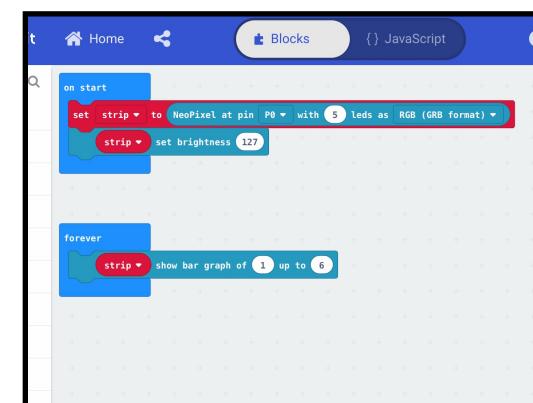
**JavaScript** is a language that is used to write all sorts of programs, particularly on the web.

The program is copied to a part of the micro:bit called **flash memory** – this is remembered when the micro:bit is switched off. Writing a program to this memory is called **flashing**.



You will see a short computer program written in a programming language called **JavaScript**.

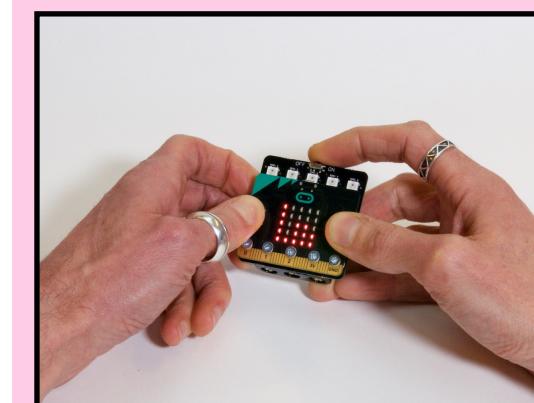
We won't be using this language directly. Instead, tap the **Blocks** slider at the top of the screen.



The screen now shows the same program in a simple visual language called **Blocks**.

Each horizontal block is an instruction to do something. Instructions that are joined together are run one after the other; the surrounding boxes say *when* to run them.

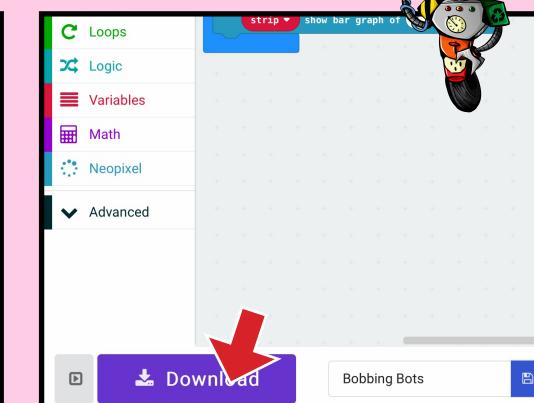
Before we look closely at this program, let's try it out!



To copy a program onto the micro:bit, first make sure that it is switched OFF with the tiny black switch on the top.

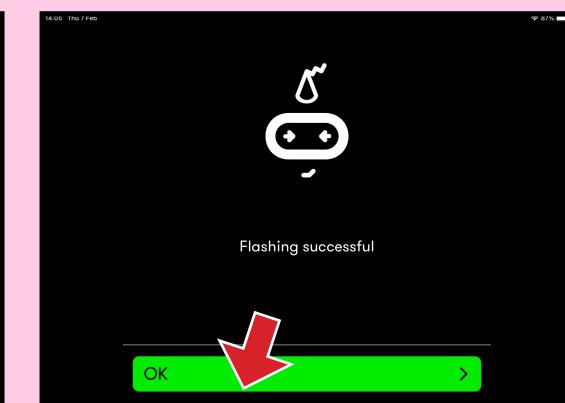
Now hold down both the **A** and **B** buttons on the front and switch it on – *keep holding the buttons* until the red LEDs show a pattern of columns. You can then let go of the buttons, but leave the micro:bit switched on.

Your micro:bit is now in Bluetooth programming mode.



Tap the **Download** button on the iPad.

The iPad will connect to the micro:bit wirelessly and copy the program to it. This will take a few seconds.



When you see **Flashing successful**, the micro:bit is ready. Tap **OK** on the iPad to go back to the code screen.

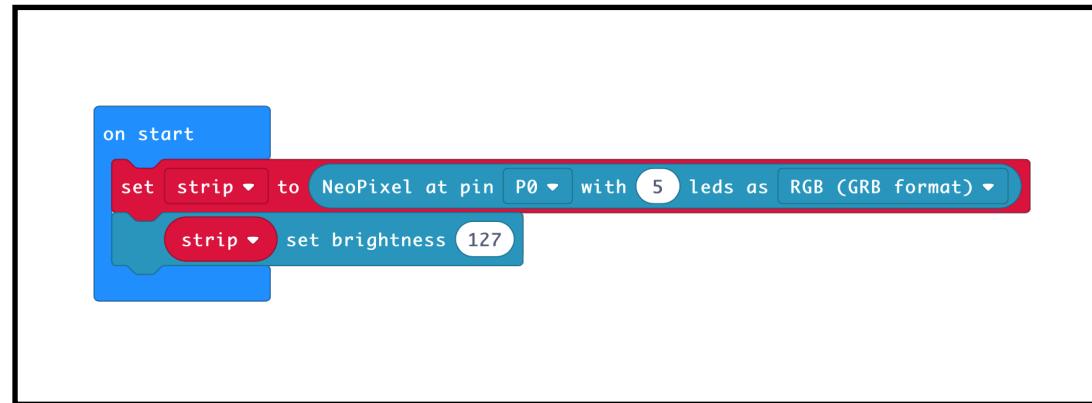
The micro:bit will immediately begin to run the new program.



Repeat these three steps whenever you want to download a new program!



Free to use under Creative Commons CC-By 4.0 License



Let's look at the program in more detail.

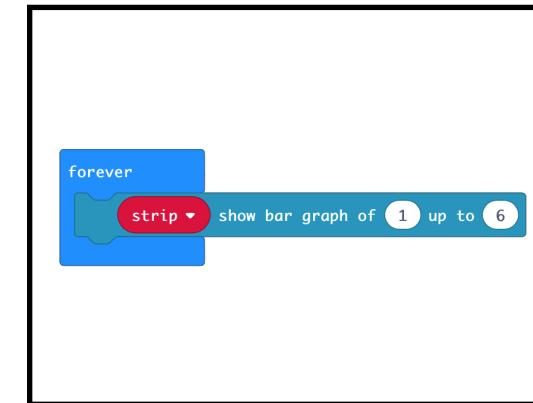
Instructions in the **on start** box are run *once only* when the micro:bit is first switched on.

The first instruction tells the micro:bit that there are **5** multi-coloured (**RGB**) **NeoPixel** lights connected to pin **P0** (this is the gold contact labeled "0" at the bottom of the board). These are the white parts along the top of the servo:lite board under the power switch.

We give these lights a name: **strip**. Using names is common in programming – the names are called **variables**.

The second instruction tells the micro:bit to set the brightness of the lights – using their variable name, **strip** – to **127**.

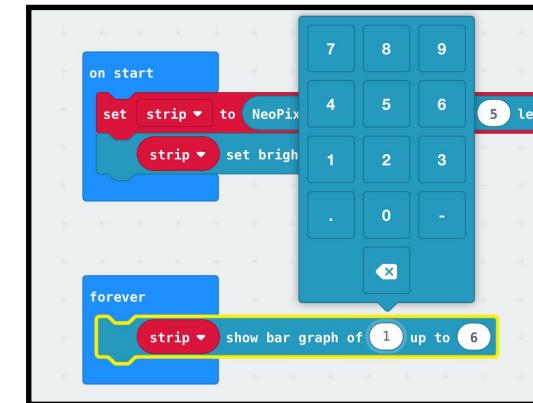
The brightness is a number between 0 (off) and 255 (brightest), so 127 is half way. We do this because the lights are hard to look at when they are at full brightness.



Instructions in the **forever** box are run *repeatedly* until the micro:bit is switched off.

The instruction in this box tells the micro:bit to show a bar graph on the lights – again using their variable name.

The number of lights lit up depends on how big the first number is compared to the second. In this case, just the first light will turn on. You should see this on your micro:bit now.

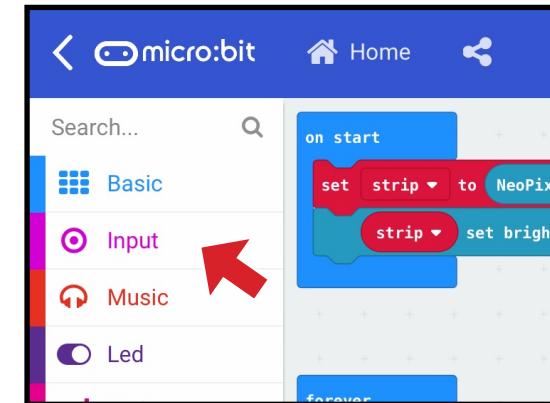


Let's try making a change.

Tap on the first number and you will see a numeric keypad appear. Tap **2** and then tap on the grey background to dismiss the keypad.

Now download the new program to your micro:bit (repeat the steps in the pink box on the other side). You should see two of the lights come on.

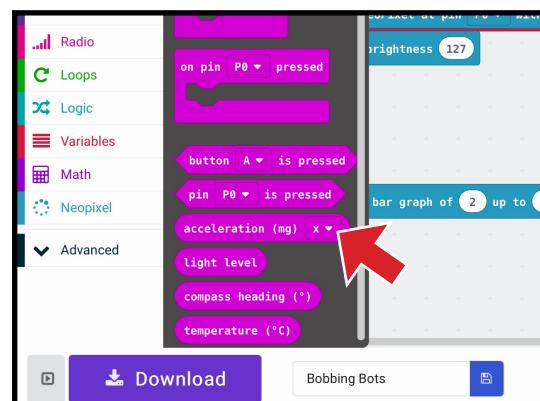
Try other values for the first number!



Although the **show bar graph** instruction in the **forever** box is always running, it always shows the same number, so our program isn't very interesting.

Let's use a value that changes when we physically interact with the micro:bit. The micro:bit contains a part called an **accelerometer** that senses movement.

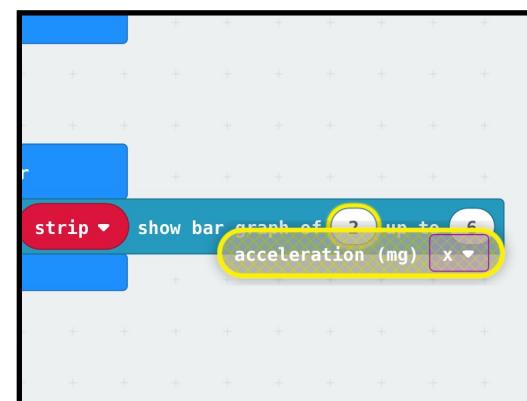
Tap on the **Input** button at the left of the screen.



You will see a list of blocks that you can add to your program related to sensing things, which we call **input**.

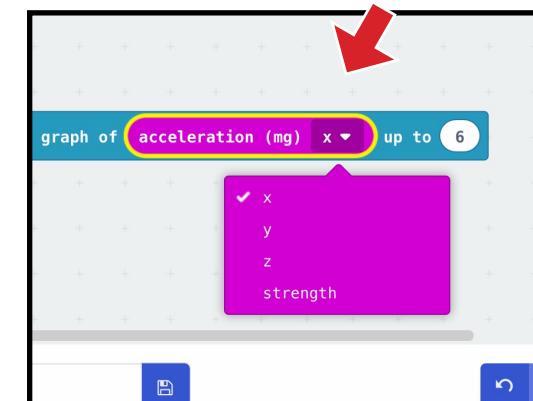
Tap the **acceleration (mg)** block to add it to your program. This block represents a value that changes when the micro:bit speeds up or slows down.

Once this block has been added to the programming window, you can drag it around with your finger.



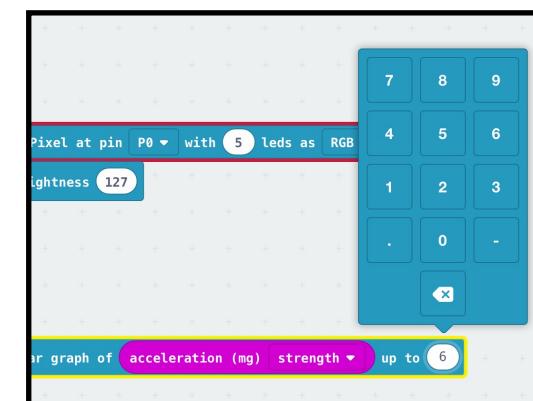
Drag the new **acceleration** block over the first number in the **show bar graph** block until the number highlights with a yellow outline, then let go.

You should hear a click and the new block will snap inside the other, replacing the number.



The micro:bit can sense acceleration in three directions: **x** (left and right), **y** (up and down) and **z** (backwards and forwards). For now, we are interested in the strength of the acceleration in any direction.

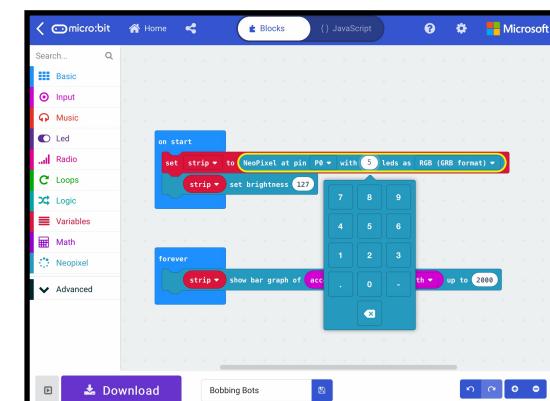
Tap on the **x** next to **acceleration (mg)** and change this to **strength**.



Acceleration is measured in milli-gravities (**mg**) – a value of 1000 is the force of the Earth's gravity. So we need our bar graph to be able to show numbers much larger than 6.

Tap on the **up to** number and change this to **2000** – twice the force of gravity.

Download your program again. Shake the micro:bit and see what happens.



When we use this micro:bit in our jellyfish, we are going to connect 6 more lights. As a final step, we need to tell our micro:bit this.

Tap on the number **5** in the **NeoPixel** block and change it to **11**.

Download this program and try it out. You might like to attach the strip of extra lights to see them working. Side 2 of the making worksheet shows how to do this.