

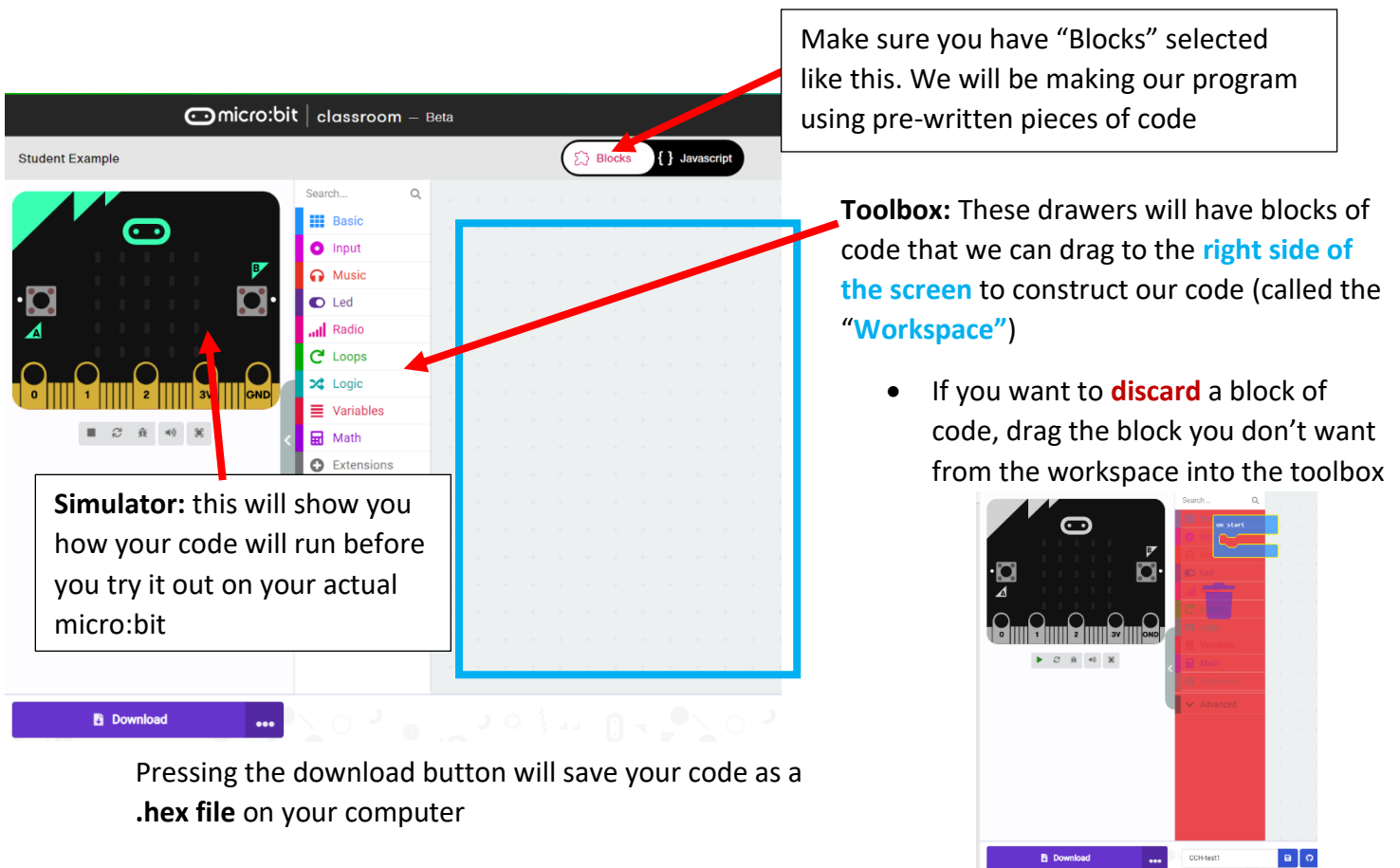
Welcome to the wonderful world of micro:bit!

- If the micro:bit is not working, see Appendix B of this handout (last page)

2 ways to code with micro:bit

1. Makecode editor
 - Type the following website link into your internet browser: <https://makecode.microbit.org/>
 - Select “New Project”, give your project a name and click “Create”
 2. Micro:bit Classroom (Student Instructions)
 - Type the following website link into your internet browser: microbit.org/join
 - Enter the join code that your teacher will display (5 things to enter: Weather, Colour, Place, Number and Password)
 - Enter your name
- **Teachers:** to set-up a micro:bit classroom where you can send code to students during a lesson, view their codes as they make them, and save reports of the progress of your students, follow the instructions on this website: <https://classroom.microbit.org/>

Here’s a brief overview of the makecode editor (same editor that micro:bit classroom uses)



Make sure you have “Blocks” selected like this. We will be making our program using pre-written pieces of code

Simulator: this will show you how your code will run before you try it out on your actual micro:bit

Toolbox: These drawers will have blocks of code that we can drag to the **right side of the screen** to construct our code (called the “**Workspace**”)

- If you want to **discard** a block of code, drag the block you don’t want from the workspace into the toolbox

Pressing the download button will save your code as a **.hex file** on your computer

How to “Direct flash” your code to your micro:bit

Press the download button in the bottom left corner. You will see the following pop-up:

Download completed...



Press the “pair now” button.

Your code is being downloaded as a .hex file. You can drag this file to your micro:bit using your computer's file explorer.



New!

Download your code faster by pairing with web usb! **Pair now**

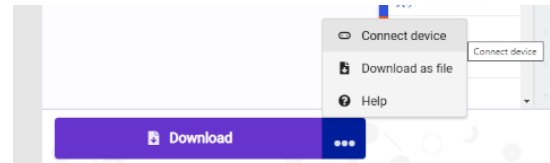
Don't show this again

Help

Download again

Done

If this doesn't appear, then click the 3 dots next to the Download button and select “Connect device”

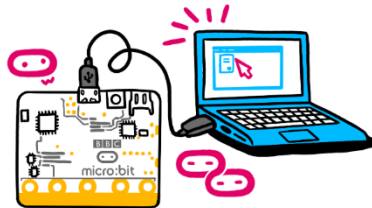


A new pop-up will appear, telling you to make sure your micro:bit is connected to the computer with a USB cable. Follow the instructions of the pop-ups below:

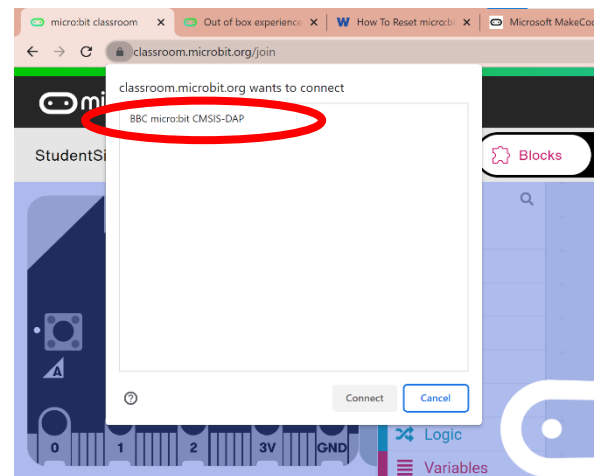
Connect your micro:bit...



Pair your micro:bit to the computer by selecting 'BBC micro:bit CMSIS-DAP' or 'DAPLink CMSIS-DAP' from the popup that appears after you press the 'Next' button below.



Next



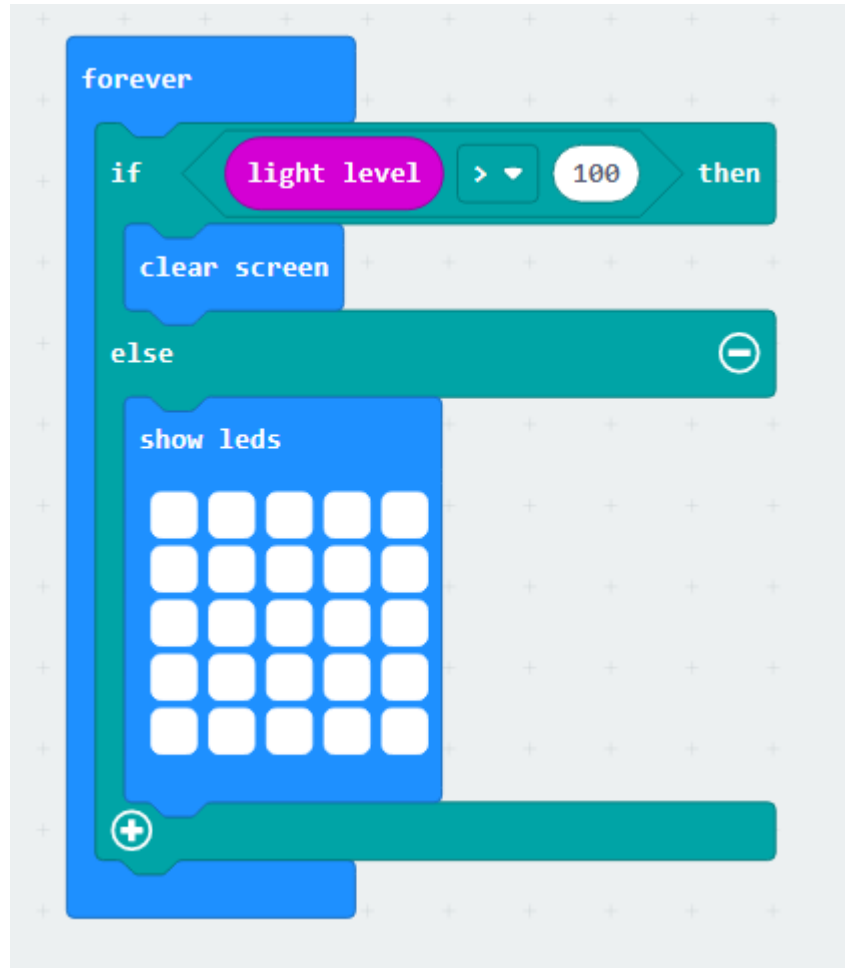
You're connected! Pressing 'Download' will now directly flash your code to your micro:bit. In other words, you don't have to keep moving the .hex file from the Downloads folder of your computer to the MICROBIT drive.

Take a look at your micro:bit, and you'll see that your program is now running on the device itself!

Making an automated light sensor (Nightlight)

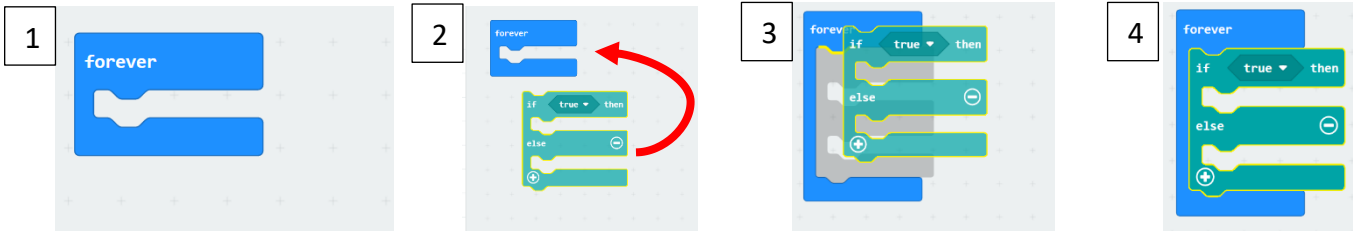
Here's simple sample code that can automatically turn the LED lighting system on/off based on the **input** of the light sensor. It will **turn off the LEDs when there is a lot of light shining on the light sensor, and turn on the LEDs when there's a low amount of light.**

- Drag and drop the respective blocks from the toolbox into the workspace as shown
- Hints:
 - The colour of these blocks match the drawer title in the toolbox.
 - You can click on the squares to turn on specific LEDs on the micro:bit's display.
 - Make sure the inequality is correct
- Once you're done, click the download button to flash your code to the micro:bit and see if the code worked!



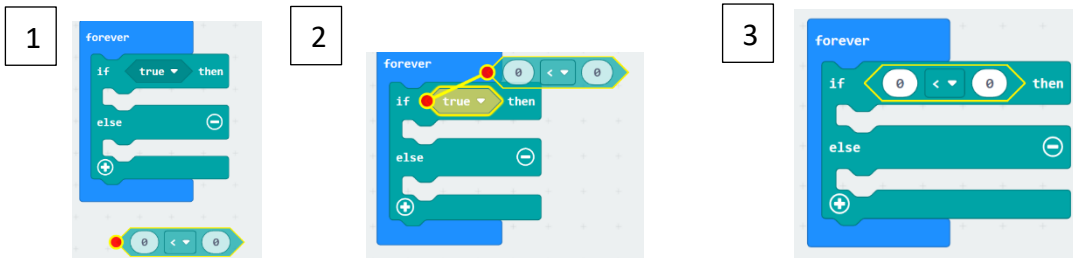
For those new to micro:bit, here are some tips to remember when building your code in the workspace:

1. How to snap blocks into each other



Note: blocks like "forever" which don't have grooves on both sides cannot be snapped inside other blocks

2. How to replace values in **object** or **event** blocks with other **logic**, **input** or **variable** blocks

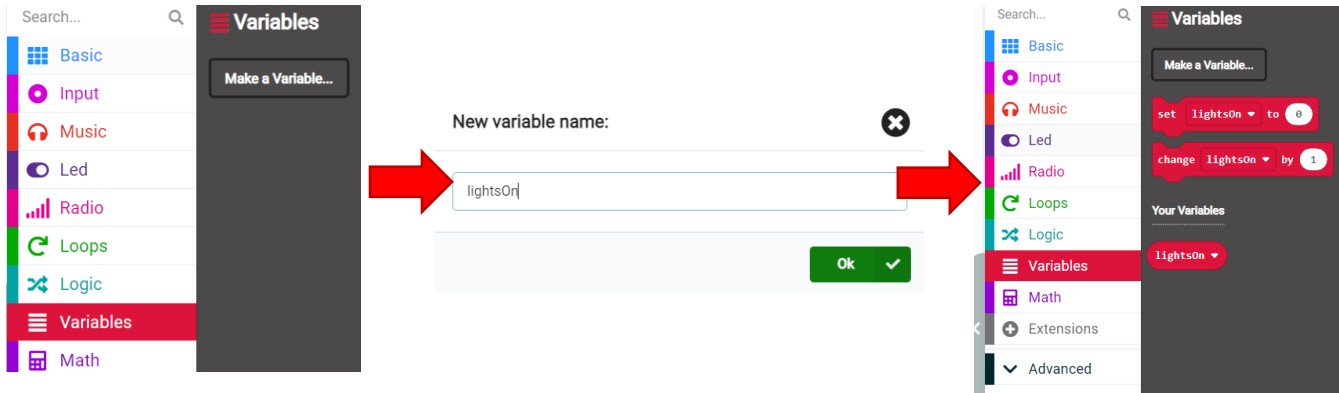


Now we will make a program that will allow us to **dim the micro:bit LEDs by clapping or talking to it!**

Making a sound-controlled light dimmer out of our nightlight

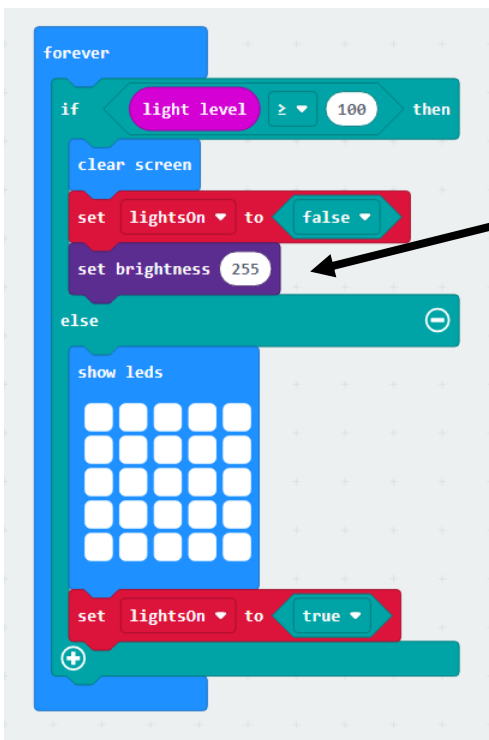
Note: You will need to use a micro:bit V2 to make a sound-controlled light dimmer (requires microphone to detect audio input). To check the version, check the text printed on the back of your micro:bit (diagram in Appendix B)

To alter our existing code to dim the lights *only when the LEDs are on*, we need our program to understand when the lights are on (true) or not (false). To do this, we can make a **variable** to store this information:



Now that you have new coding blocks available, You're going to alter the light sensor code so that you record the on/off state of your lights in the variable "**lightsOn**":

- Some of the code blocks might be located under "... more" in the toolbox drawers.



Your micro:bit's LEDs can have a brightness value from 0 (min) to 255 (max).

Putting this block here will reset the brightness of our lights to maximum before they turn back on again

To save energy when the lights are on (LightsOn = true), we can **change the brightness of the LEDs** so that we aren't expending more energy than necessary.

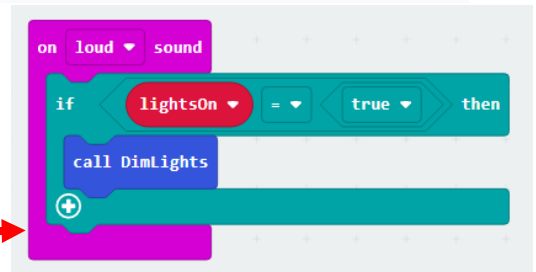
For this section, we'll be using the **“on loud sound”** block, but **you can try other input blocks** that use other sensors to customize how you control your light dimmer!

1. Similar to making a variable, a **“function” block** will be used, which will allow us to make a subprogram that will be executed only when our micro:bit detects a loud sound. To make this block:

- Go to the “Advanced” drawer of the toolbox
- > Click on “Functions” > Click on “Make a Function”
- Name the function “DimLights”

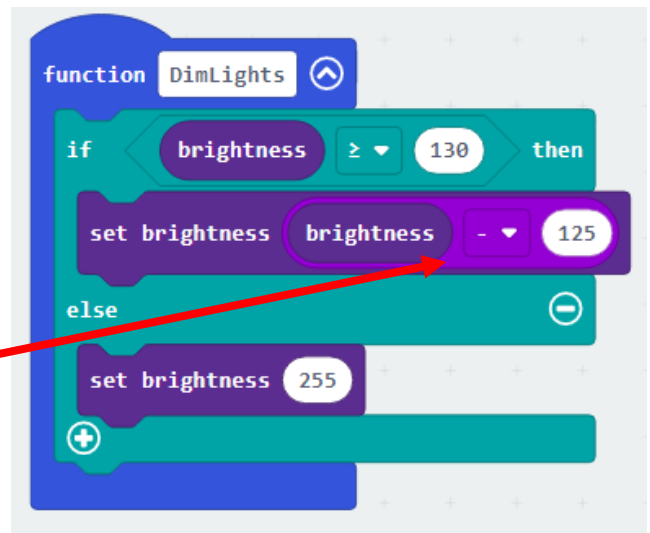


2. Now, you'll have the above **“Function DimLights”** block in your workspace, and a **“call DimLights”** block available under the functions drawer.



3. Drag the **on loud sound** input block to the workspace and construct the following procedure:

4. Now all we have left is to make the DimLights subprogram.
- The if/else statement will check what brightness level the LEDs are at while the lights are on.
 - Each clap/loud sound will make the LEDs cycle through 1 of 3 brightness levels: 255,130 and 5.
 - The brightness value can be changed using this operator block from the **“math” drawer**



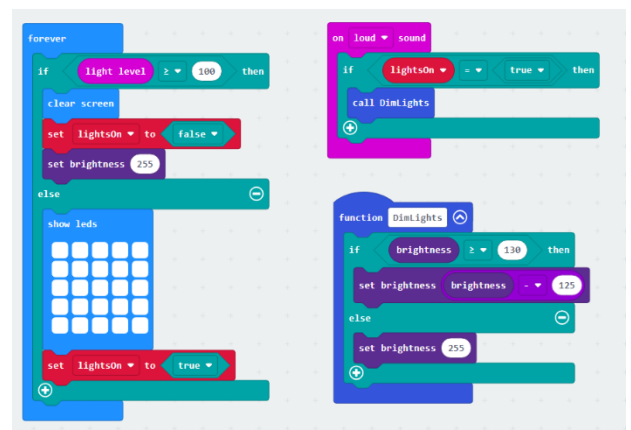
Test, Improve and Extend!

Congratulations, you've made an automatic light-sensing LED lighting system, with an adjustable dimmer! Here's what your final program will look like in the workspace:

- Flash your code and test your programmed micro:bit's light-sensing and dimming capabilities!

Challenges to try:

- Change the sensitivity of the brightness or sound level
- Add a timer so that the lights turn off after a set amount of time
- Use other inputs from the **“is _____ gesture” block** to control the lighting system in a different way
 - Example: using “shake” to simulate a door closing when exiting a room



Make it a reality: *If you use circuit components (wires, LED bulbs, resistors and a breadboard), you'll be able to use your micro:bit program to control and dim a circuit of external LED bulbs (you'll need to add blocks for your pins though).*

Appendix A – Extension Projects

There are so many ways to use micro:bit V2 to create amazing and useful programs!

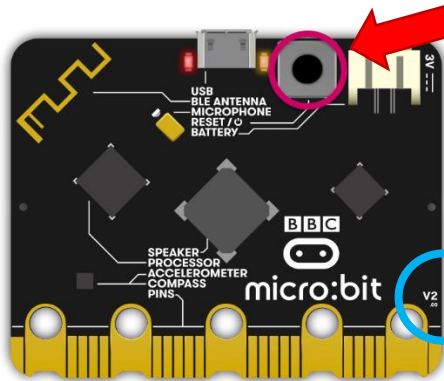
Interested in learning more? Here are some activities which relate to today's lesson on saving electrical energy using light sensors:

1. <https://microbit.org/projects/make-it-code-it/energy-light-meter/>
2. <https://microbit.org/projects/make-it-code-it/energy-light-timer/>
3. <https://microbit.org/projects/make-it-code-it/energy-cost-calculator/>

Appendix B - What do I do if my micro:bit is not working?

If the code you flashed to your micro:bit isn't running (and **you've checked that the .hex file you downloaded most recently from the website** is present on the MICROBIT drive):

Press the reset button on the back of the micro:bit



You can check your micro:bit's version here as well

If the program doesn't start up after pressing the button, then:

Check your firmware version

- To find out what version of the firmware you have on your micro:bit:
 1. Plug it in to a computer using the USB cable
 2. Go to your files folder. Open the DETAILS.TXT file on the MICROBIT drive under "This PC"
 3. Look for the number on the line that begins 'Interface Version'.

```
# DAPLink Firmware - see https://mbed.com/daplink
Unique ID: 9900000031634e450062401400000032000000097969901
HIC ID: 97969901
Auto Reset: 1
Automation allowed: 0
Overflow detection: 0
Daplink Mode: Interface
Interface Version: 0249
Git SHA: 682d8303e37355532402b8d93c4f240a3cec02a9
Local Mods: 0
USB Interfaces: MSD, CDC, HID, WebUSB
Interface CRC: 0x3f2b7e12
Remount count: 0
URL: https://microbit.org/device/?id=9900&v=0249
```

We are using micro:bit V2, so our Interface Version should be **at least 0255**

If you need to update your firmware version so that it's at least 0255, go to this website and follow the instructions: <https://microbit.org/get-started/user-guide/firmware/>

Now try moving the .hex file to the MICROBIT drive (cut and paste the file from the downloads folder into the MICROBIT folder on your computer).

- Your program should automatically start, now that the file is flashed to the MICROBIT