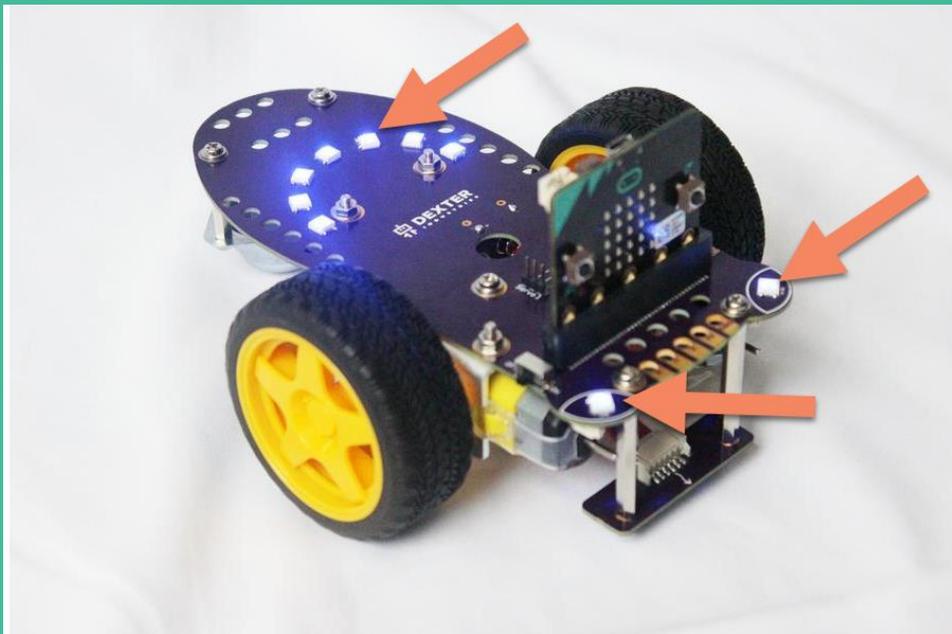


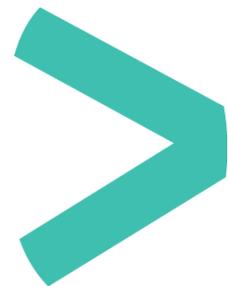
# LIGHT SHOW

## MISSION 2



# TABLE OF CONTENTS

• <b>LIGHT SHOW</b> .....	1
Your Role: Lighting Technician .....	1
Your Task: Create a Light Show.....	1
Considerations.....	1
Materials.....	1
• <b>LEARN</b> .....	2
GiggleBot Smile Lights .....	3
micro:bit LEDs.....	4
Micro:Bit Accelerometer .....	6
• <b>PLAN IT OUT</b> .....	10
Build Your Program.....	10
Build Your Stage .....	11
Are You Stuck?? .....	11
• <b>TRY IT OUT</b> .....	13
Iterate .....	13
• <b>EXTENSION</b> .....	14
Advanced Notions.....	14



## > LIGHT SHOW

### YOUR ROLE: LIGHTING TECHNICIAN

📘 Electrical lighting technicians, or simply lighting tech., are involved with rigging stage and location sets and controlling electric lights for art and entertainment venues or in video, television, or film production.

### YOUR TASK: CREATE A LIGHT SHOW

You will create a snazzy light show by moving your GiggleBot into different orientations. The GiggleBot comes with a total of 9 lights on its body and the micro:bit has 25 red lights on it. That's a lot of lights!

### CONSIDERATIONS

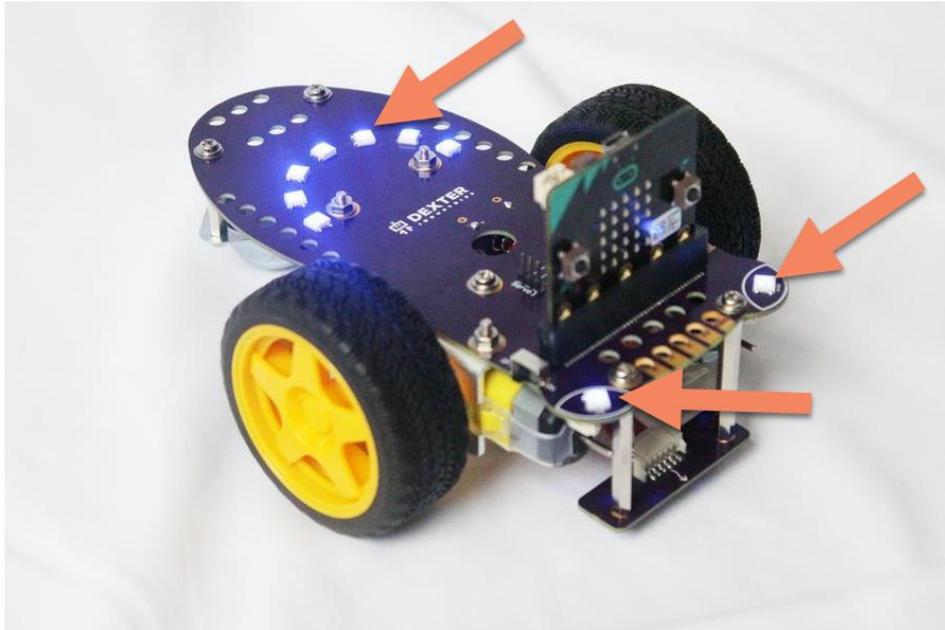
- > What movements are possible with the GiggleBot?
- > What lights on the robot do you want to use?
- > What do you want the lights on the robots to do with each movement?
- > Will the light show go along with music?
- > Will the light show be used to light up a toy stage?

### MATERIALS

- > GiggleBot and good batteries
- > micro:bit with provided USB cable
- > Laptop / computer
- > Optional: music to play in your lightshow (from your collection of music)
- > Optional: toy stage

## > LEARN

The GiggleBot has numerous onboard RGB LEDs on-board. There are 7 in a smile shape on the back of the GiggleBot and two on the front as “eyes”.



### ***What is an RGB LED?***

RGB means red, green, blue and LED stands for light-emitting diode. A LED is a semiconductor that emits light when an electric current is passed through it. RGB LEDs can combine red, green, and blue to make a rainbow of colors.

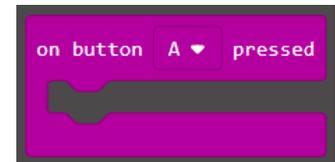
In this mission, the words “lights” and “LEDs” mean the same thing, they’re synonymous.

## GIGGLEBOT SMILE LIGHTS

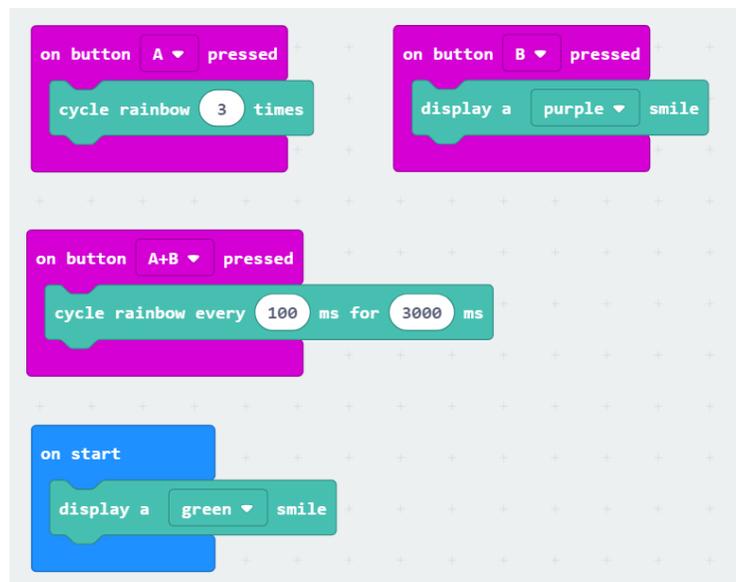
Let's make the smile on the GiggleBot light up. There are several blocks to control the smile LEDs on the back of the robot. We are going to focus on four of those blocks for now (indicated below).



Investigate these blocks by connecting it to a event-handler block such as **"on button A pressed"** . Can you make a rainbow-colored smile? Can you make a blue smile? Can you make a rainbow smile that cycles through several times? Remember to download your program each time and transfer it to your micro:bit. Look back to Mission 1 for directions on how to transfer the program.



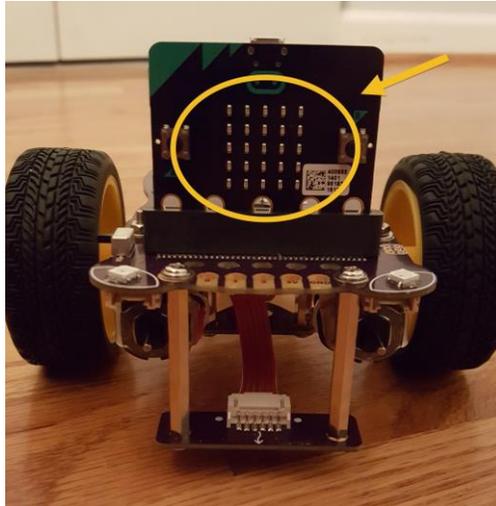
Here are a couple of sequences to try out:



Can you program the GiggleBot drive around with a rainbow smile? Can you program the GiggleBot to display a green smile and drive forward, then display a red smile and drive backwards?

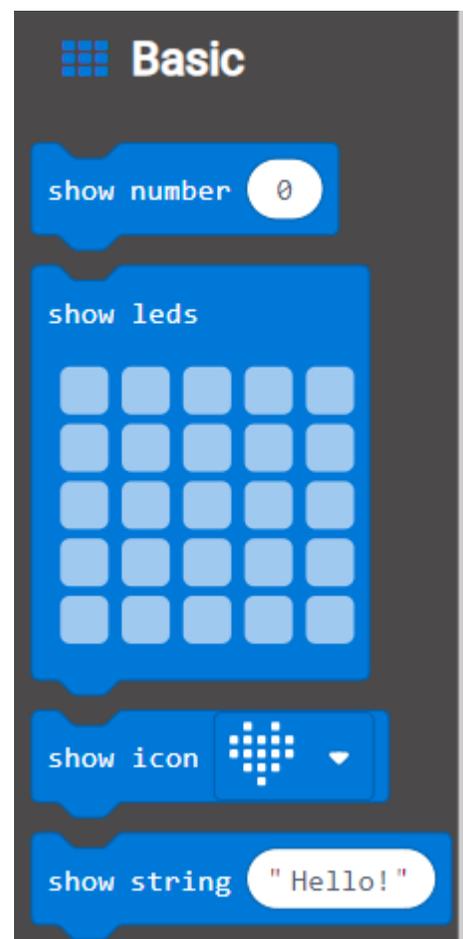
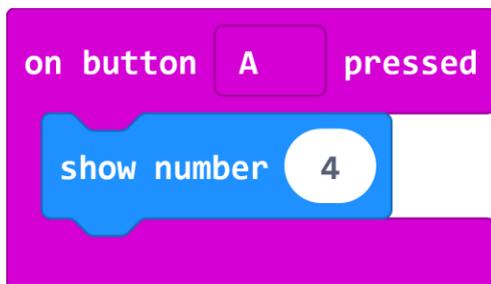
## MICRO:BIT LEDs

In addition, the micro:bit has 25 red LEDs on the front of it arranged in a 5x5 grid. That is a lot of lights!

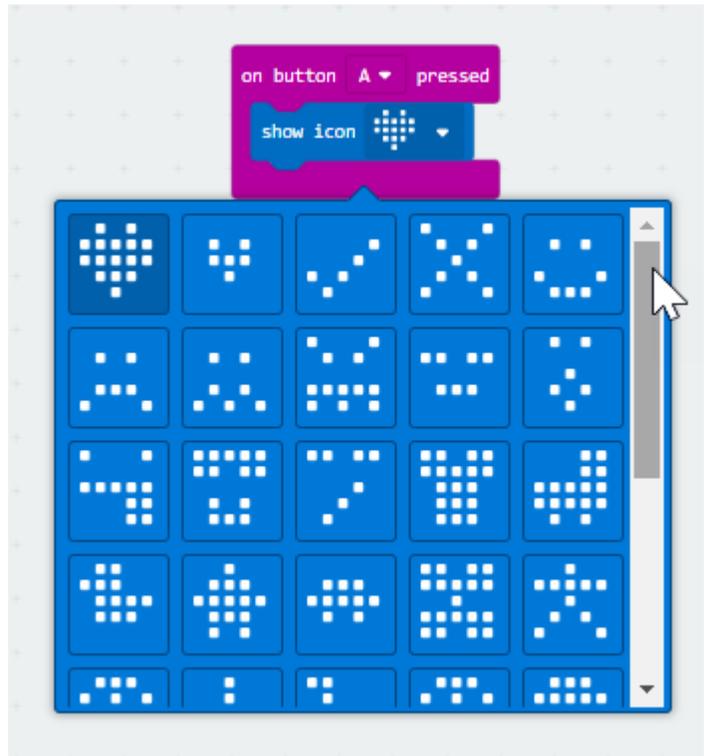
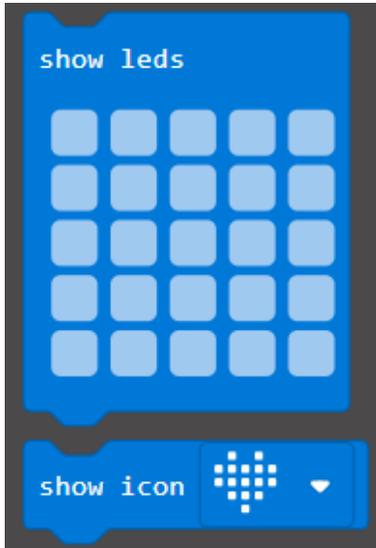


You can display a variety of things with these LEDs using blocks found under the Basic category.

Using the “**show number**” block you can tell the GiggleBot to display a number or sensor value.



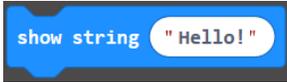
You can also display premade icons or make your own designs on the LED screen using the **show leds** or **show icon** blocks.



Here is an example of a program where you can change the icon displayed on the micro:bit using the buttons. This example will make the micro:bit smile when button A is pressed, and pucker when button B is pressed.



Download and transfer this program to your GiggleBot to see the LEDs light up.



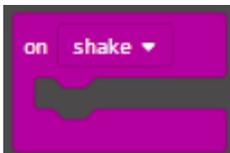
What if you want the GiggleBot to display a message? Use a **“show string”** block. You can write any text you like in between the quotation marks.

Try the following program to see it in action.



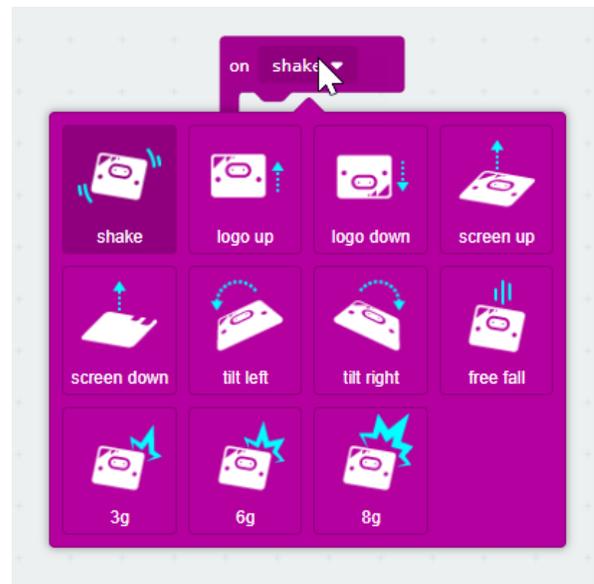
Create new programs or add onto programs you have already created with these new blocks. What programs can you make? Create a couple of programs using these new blocks. Download and transfer each new program to your GiggleBot and then run it.

## MICRO:BIT ACCELEROMETER



An accelerometer is an instrument that measures movement and then gives you feedback on the type of movement done by the micro:bit. You can use the movement of the micro:bit on the GiggleBot to start programs. We have already experimented with using the event handlers that use the buttons such as **“on button A pressed”**.

Now, we are going to use movement. Let’s take a look at the options. Pull an **“on shake”** block to the workspace. Click on the down arrow to see all of the options for this block.



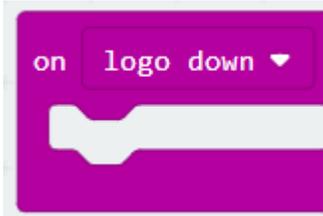


This is when you shake or jostle the micro:bit (or the GiggleBot when the micro:bit is attached).



**GiggleBot** - This is the standard position the micro:bit is in when you place it in the GiggleBot. The GiggleBot logo is facing up.

**micro:bit** - The LED screen is perpendicular to the ground (at a 90 degree angle in relationship to the ground) and the micro:bit logo is above the screen.



**GiggleBot** -This is when the GiggleBot, with the micro:bit attached, is upside down. The wheels are facing up and the top of the GiggleBot is facing the ground. Usually not a good position.

**micro:bit** - The LED screen is perpendicular to the ground (at a 90 degree angle in relationship to the ground) and the micro:bit logo is below the screen.

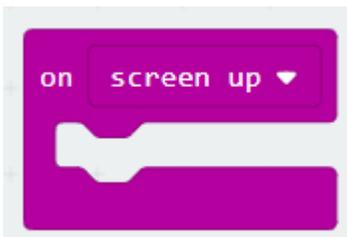




This is when the micro:bit and GiggleBot leans to the left.

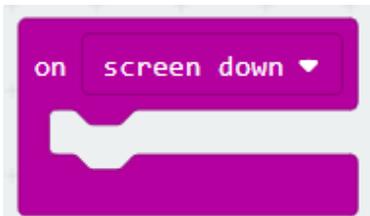


This is when the micro:bit and GiggleBot lean to the right.



**GiggleBot** - This is when the front of the GiggleBot is facing the ceiling.

**micro:bit** - This is when the LED screen is facing the ceiling.



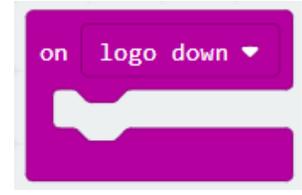
**GiggleBot** - This is when the front of the GiggleBot is facing the floor, doing a handstand.

**micro:bit** - This is when the LED screen is facing the floor.





Now that you know about some of the event-handler block options, let's create a program. Move a **"on logo up"** block and a **"on logo down"** block to your workspace.



Add blocks to each of these to tell the GiggleBot what to do when these movements are made.

- > Do you want the GiggleBot to display a rainbow smile when it is in the normal position (logo up) and a blue smile when it is upside down (logo down)?
- > Do you want the GiggleBot to show a happy face when it is in the normal position (logo up) and a sad face when it is upside down (logo down)?

It is up to you! Add blocks to **"on logo up"** and **"on logo down"** to tell the GiggleBot what to do.

Here is an example:



Create a program using these new event-handler blocks or copy the program above to test it out.

## > PLAN IT OUT

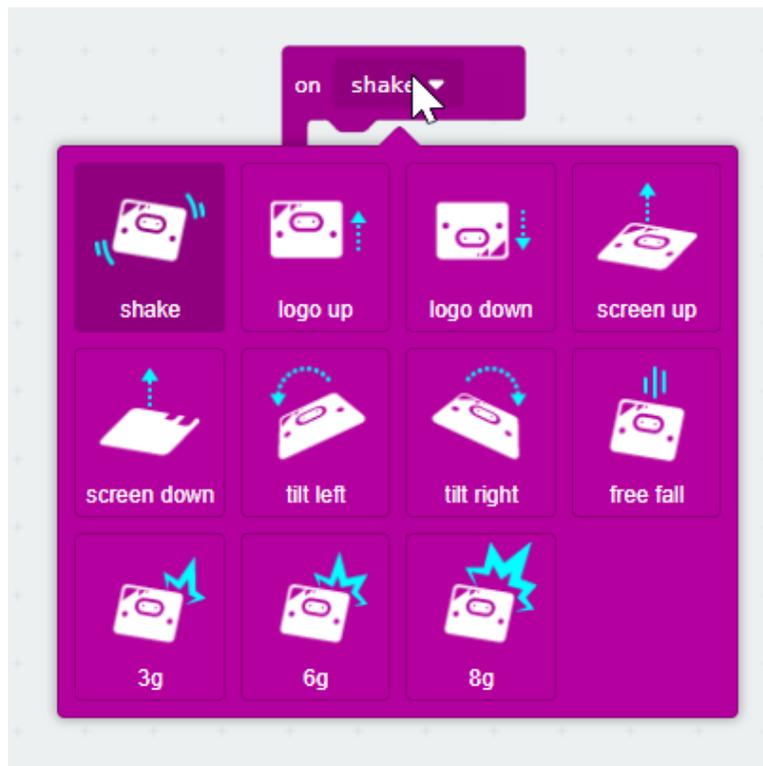
What movements do you want to use to create your light show? What lights will you use? The 5x5 grid on the micro:bit? The smile? All of them?

Write down your plan on a piece of paper before you start programming. If your light show is going to be set to music be sure to play the music while you plan it out.

**i** Your GiggleBot cannot make any sounds or play music YET since it does not have a speaker. We will add a speaker in mission 7.

## BUILD YOUR PROGRAM

Use a variety of the event handler blocks to create your light show. You can use any of the blocks you learned about in the Learn section in addition to the **on button \_\_\_ pressed** blocks from Mission 1.



When you are finished writing your program, download and transfer it to your GiggleBot.

## BUILD YOUR STAGE

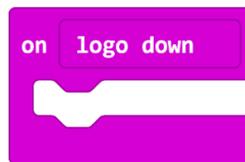
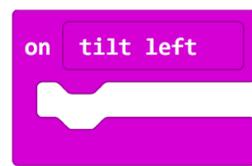
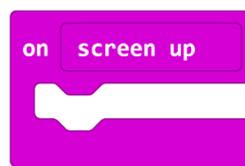
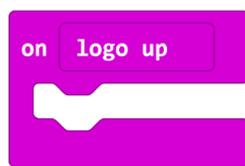
Is your GiggleBot going to light up a stage? Do you have some toys that may want to do a play? Now is a good time to build it up. It's entirely optional, of course, as the GiggleBot can put on a light show all by itself.

## ARE YOU STUCK??

Let's build a program together!

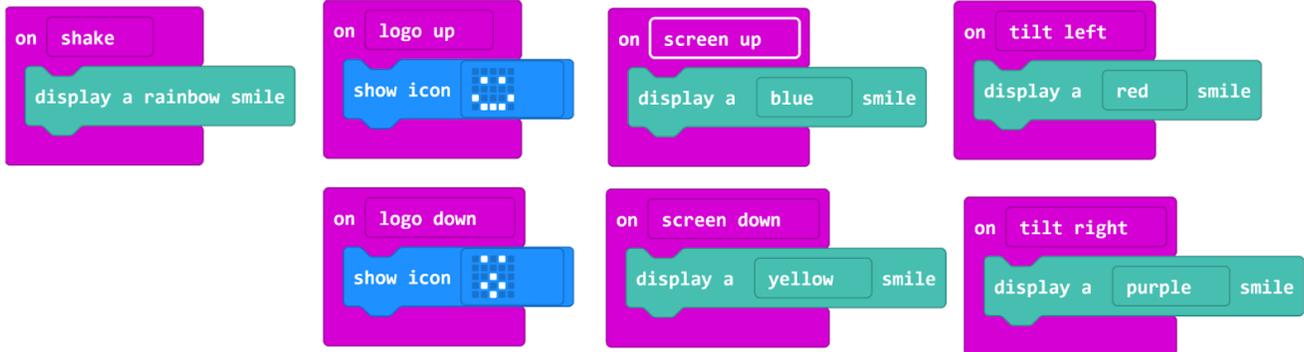
Here are my notes for what I want the GiggleBot to do during the light show:

- > Shake - cycle Rainbow smile
- > Logo up - Smile face on micro:bit LEDs
- > Logo down - surprised face on micro:bit LEDs
- > Screen up - blue smile
- > Screen down - yellow smile
- > Tilt left - red smile
- > Tilt right - purple smile



Using the notes above, move the event-handler blocks to your workspace (**on shake**, **on logo up**, **on logo down**, etc.)

Next, add the corresponding light blocks to each event-handler block using the notes above..



When you are finished, download and transfer the program to your GiggleBot. Look back at the Learn section from Mission 1 if you do not remember how to download and transfer the program.

## > TRY IT OUT

Run your program and display your light show. Hold the GiggleBot in your hands and turn it in different directions to see the different light patterns that you programmed.

### ITERATE

- > Did your GiggleBot's light show go exactly as you had anticipated?
- > Do you see any areas that you would like to modify or improve upon?

In engineering, we :

- > try a solution
- > think about what needs to change
- > iterate.

It's called the **Engineering Design Process**.

As you modify and revise your program, save each new program with a version number – you never know when you might want to take another look at an older version (for example: Lights\_V1 where “V” stands for version).

## > EXTENSION

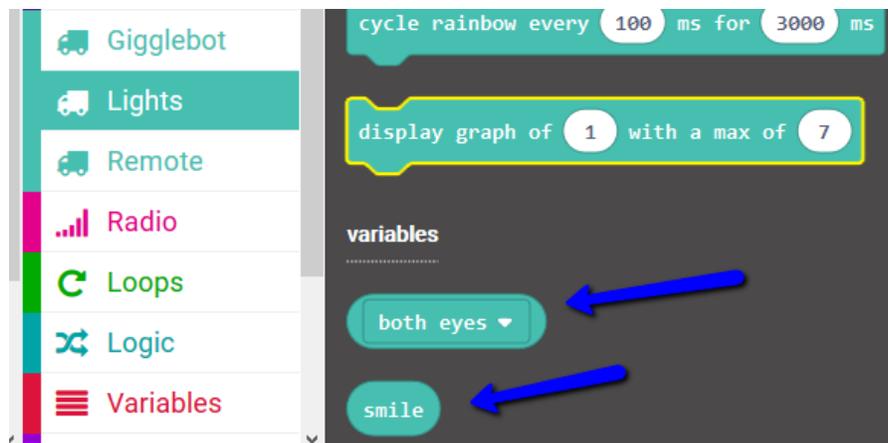
Add on to your dance routine to include a light show. How can you enhance your dance routine using the lights on the GiggleBot?

### ADVANCED NOTIONS

You may have wished for better control over the lights. It is indeed possible to control each LED individually, those that belong to the smile but also the two eyes in front.

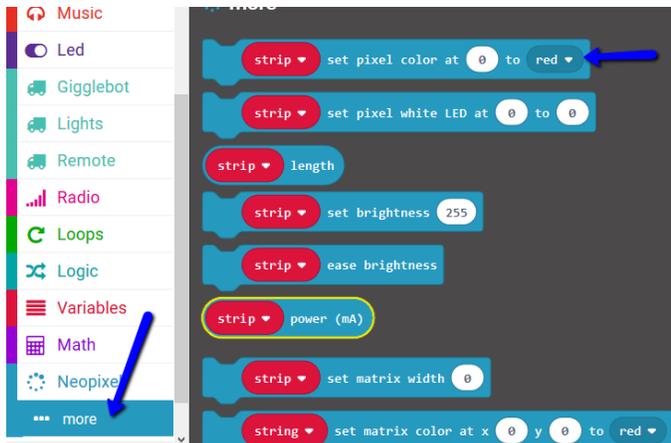
### NEOPIXEL CONTROL

When you click on the Lights category you will find two round blocks at the end, one for the eyes and one for the smile.



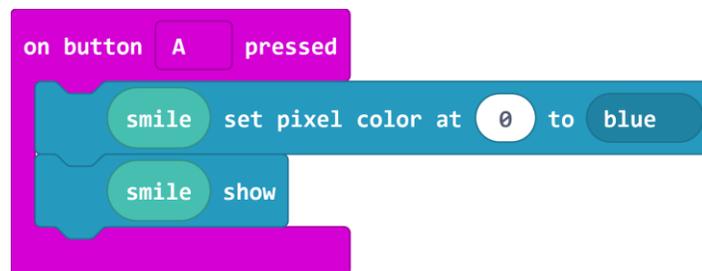
Those blocks cannot be used like the others we've used so far as they do not stack like the other blocks. So what can we do with them?

They are meant to be used with the **Neopixel** blocks! If you have access to other micro:bit accessories that also have neopixels, you will be using the same blocks. Those blocks are not specific to the GiggleBot and they can be used in many different situations with the proper devices. In the case of the GiggleBot, you need to replace the red **strip** inner block with either the "both eyes" block or the "smile" block seen above. You will find those blocks in the Neopixel category near the end of all categories.



There's one block that's interesting in the "more..." section of the Neopixel category and that's the very first block at the top.

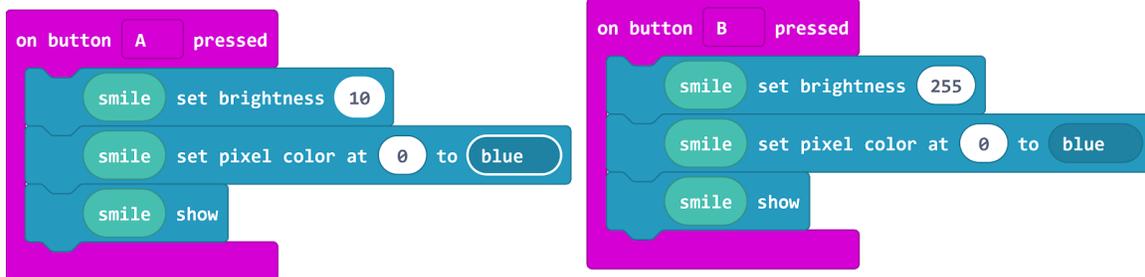
The following code will turn one LED to blue and just one. Can you guess which one? Try it out! Don't forget to press Button A to see the answer.



The first block sets LED 0 (the first one on the right side of the smile) to blue. Computers and micro:bits like to count starting from 0, it takes some getting used to. The second block, the "smile show" one, is what actually turns the LED on. Why is this done in two steps? Well, it allows you to set more than one LED but have them turn on all at the same time! Without the "smile show" block, the LEDs will not change. So far, we didn't have to worry about that because we were using the blocks made especially for the GiggleBot. Once you start using the neopixel blocks (they are a slightly darker blue), you are responsible for deciding when to turn the lights on.

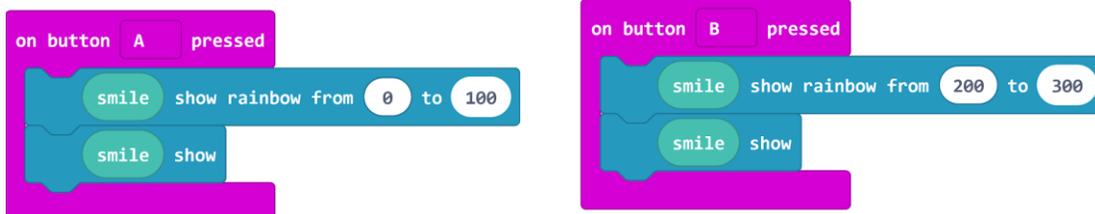
## BRIGHTNESS

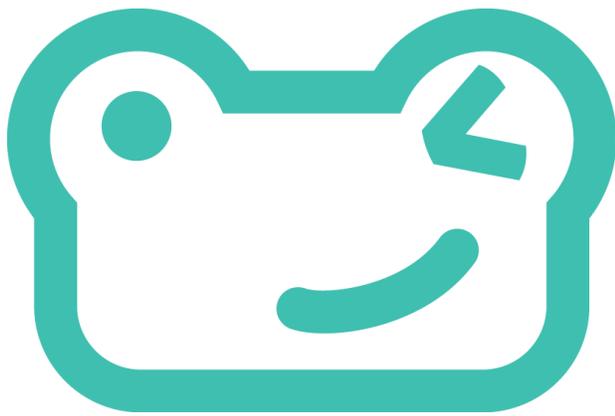
Here's another little trick that can be handy for your light show. Can you guess what will happen?



Now that you've learned more controls over the LEDs, are there any changes you would like to bring to your light show?

You can keep on exploring the neopixel blocks. Not everything you will try will work on the first attempt. That's ok. Exploration is fun! Here's another one you can play with.





# Giggle Bot

Copyright Dexter Industries 2019. All rights reserved. Reproduction and distribution of the Mission without written permission of Dexter Industries is prohibited. GiggleBot is a registered Trademark of Dexter Industries.

Contact [dexter@dexterindustries.com](mailto:dexter@dexterindustries.com) for permissions and questions.