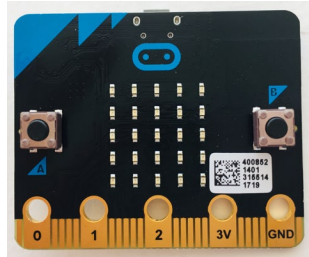


WEEK 2

DAY 1: INTRODUCTION TO MICRO:BIT



INTRODUCTION

This week makers learn how to start using a microcontroller (Micro:bit) with coding software (MakeCode) to begin designing cool physical computing projects.



ESSENTIAL QUESTIONS

- What is a microcontroller?
- How can I use one to make cool stuff that connects to the physical world?
- How do artists, engineers, and makers solve problems when they're working/inventing?



LEARNING OUTCOMES

1. Learn about a microcontroller and how to program one with coding software.
2. Write lines of code and learn how to save, download, and upload code from software to the microcontroller.
3. Engage in pair programming as a way of collaborating with a partner.



VOCABULARY

Physical computing: Creating or using devices that interact with the world around us

Microcontroller: Small device (similar to a mini computer) used to control other devices and machines

Pair programming: Method of working in pairs where one person “drives” the laptop while the other “navigates” or instructs the driver

Software: Term used to describe computer programs

Code: Set of instructions that a computer understands

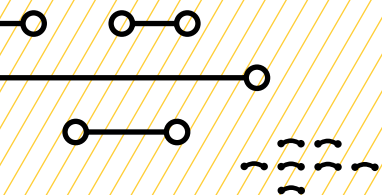
Upload: Sending information or data to a device or computer

Download: Taking information or data from a device or computer

USB to micro-USB cable: Cord that transfers data and information from the computer to a device

USB flash drive: Portable storage device that can store and move files

Troubleshooting: Using resources to solve issues as they arise

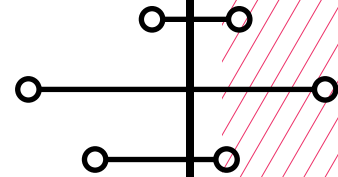
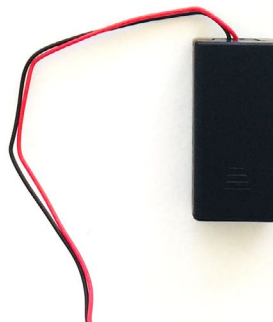
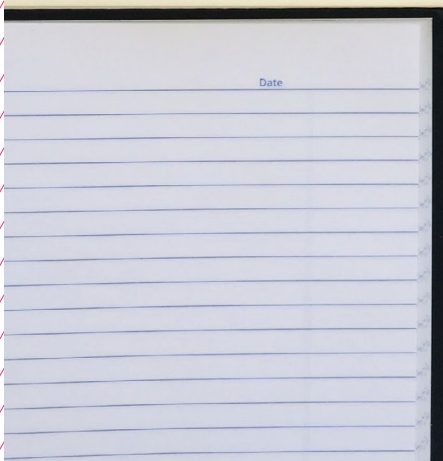
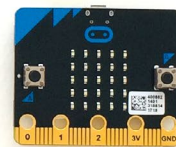
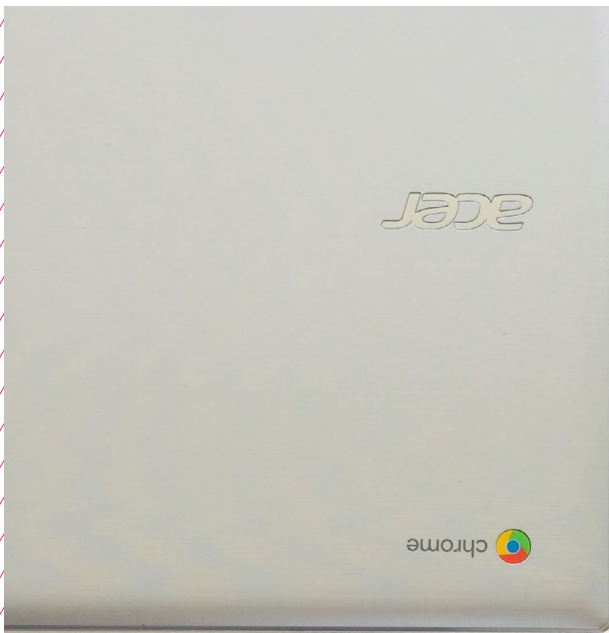


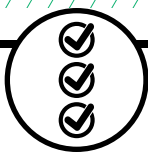


MATERIALS LIST

EACH PAIR OF MAKERS NEEDS:

- Micro:bit microcontroller
- Laptop with internet connection
- USB to micro-USB cord
- USB flash drive
- External battery pack
- AAA batteries (2)
- Notebook





TEACHER PREP WORK

1. Ensure the internet connection is working.
2. Connect your laptop to a projector or screen.
3. Preload videos and slideshow.
4. Print the [Troubleshooting Tips](#) at the end of the lesson and post in the classroom.

FACILITATION TIPS

Make sure that you're comfortable with the save/download/upload process so you can help makers going through this process.

Makers will likely be curious and explore various other blocks on the software page. It's up to you as the facilitator to decide how much you want to let your makers explore. It can be helpful to verbally honor their curiosity but remind them to follow your specific instructions to ensure everyone learns the basics first. Remind them that there will be more time to explore and that this is just the first day.

For makers who finish quickly or need more of a challenge, inform them that there are many tutorials on the [MakeCode website](#) (such as "Flashing Heart," "Name Tag," or "Smiley

Buttons") that they can try on their own to explore the Micro:bit further.

Saving files: This curriculum doesn't require that students create MakeCode accounts and instead uses USB flash drives to save files. Students can create accounts if they have an email address they can login and verify their account with. This enables them to easily save their files, and they can then login and use MakeCode from any other computer with an internet connection.

Collaboration: Let smaller issues work themselves out. Record specific positive examples that you can share with makers in the moment or at the end of the project. These examples provide models for all learners.

Frustration: When frustration levels are not high, let learners figure it out on their own or keep facilitation at a minimum by asking a question and walking away. When frustration levels are high, intervene more directly to help makers find some success.

Circulate among the makers and monitor for both collaboration and frustration.

ADDITIONAL RESOURCES

[Micro:bit Project Tutorials](#)

[Wonderful Idea Co: Micro:bit Mutants](#)

[Micro:bit Tutorial on YouTube](#)

INTRODUCTION TO MICRO:BIT

STEP 1



Get to know the Micro:bit microcontroller board.



Group makers in pairs, and give each pair a Micro:bit **microcontroller** board to study.

EXPLAIN



The Micro:bit is one of many types of microcontrollers—small devices, similar to a mini computer, used to control other devices and machines.

Ask makers to study the Micro:bit board closely and:

- Write down and draw their observations in their notebook.

After 3 minutes, ask makers to share:

- What do you notice?
- What do you wonder?

Parts to emphasize that will be used today:

- LED grid
- Micro-USB port
- Battery port

STEP 2



Get to know the coding software and pair programming.

Assign each pair of makers a laptop and explain that next they'll engage in **pair programming**, which is one way engineers work together. Ask them to figure out who has the next birthday coming up, and that person will start as the “driver.”

EXPLAIN



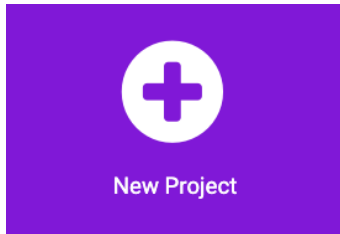
The two roles are:

| | |
|------------------|--|
| Driver | Operates the laptop, typing, clicking around, etc. |
| Navigator | Instructs the driver where to go and what to do next |

Remind makers to stay in the roles of navigator and driver. The driver shouldn't make any decisions without the navigator.

Instruct the navigators to help the driver to get to the MakeCode website:

makecode.microbit.org



Next, makers will:

- Create a **[+new project]** in MakeCode. This will take them to the coding **software**.
- Explore the coding software for 3 minutes (timed).

After 3 minutes, ask makers to switch roles. Set the timer for another 3 minutes.

After makers have each had a chance to explore in the driver and navigator roles, ask them to share:

- What did you discover?
- What do you wonder?

STEP 3



Write code for the LED display grid.

Ask makers to switch roles again, and instruct drivers to open a new blank project. Share your screen so that everyone can see how to write their first lines of **code** to program the Micro:bit.

Note: This first project is also available as a step-by-step tutorial called “Flashing Heart” on the [MakeCode website](https://makecode.microbit.org).

EXPLAIN



We'll program an animated display for the LED grid using the software and simulator. Navigators, watch closely and instruct your driver to follow these instructions.

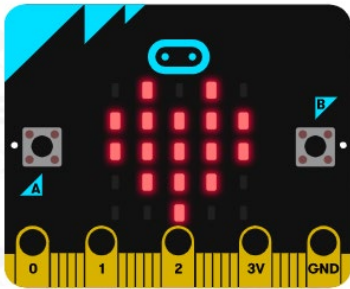
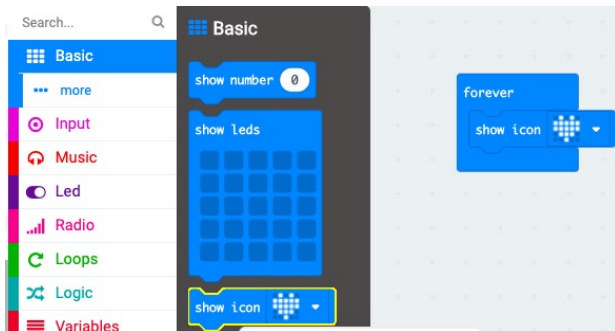
DEMONSTRATE AND HAVE MAKERS FOLLOW ALONG:



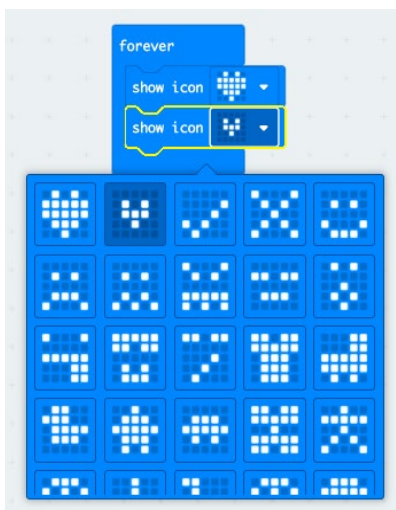
1. Start by using the **forever** block that is in the coding space.



2. Go to the **Basic** menu and drag over a **show icon** block into the **forever** block. You should see a heart display pop up in the simulator on the left side of the screen.



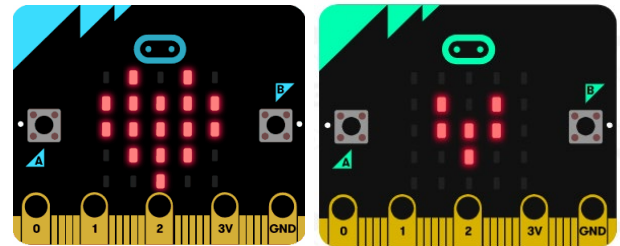
- Next, click, drag, and drop a second **show icon** block into the code. This time, click the **show icon** dropdown menu and choose the second small heart image to be displayed on the Micro:bit simulator.



- Repeat Steps 3 and 4 two more times so you have four **show icon** blocks in a row with large and small hearts.



- Check to see if the simulator shows a flashing heart.



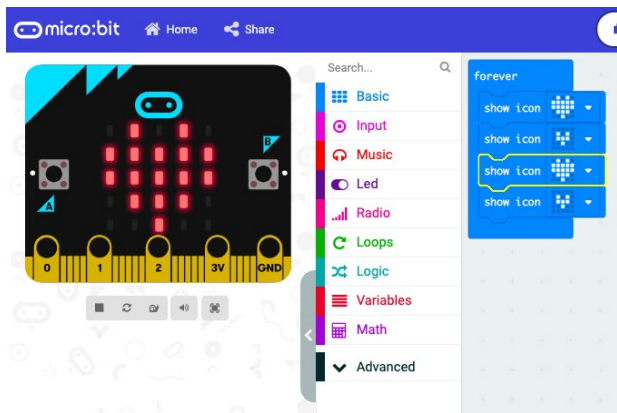
Once makers have successfully coded the flashing heart animation, they will switch roles again to create a new animation using different **show icon** choices from the **Basic** menu. Remind them to stay within the **Basic** menu and **show icon** blocks. Set a timer for 5 minutes.

STEP 4



Save the code.

After makers have programmed their animated display using the **show icon** blocks, they should see their animation in the simulator on the left side of the screen. It's now time to save their code to the **USB flash drive**.



Makers will switch roles again before continuing to the next step.

EXPLAIN

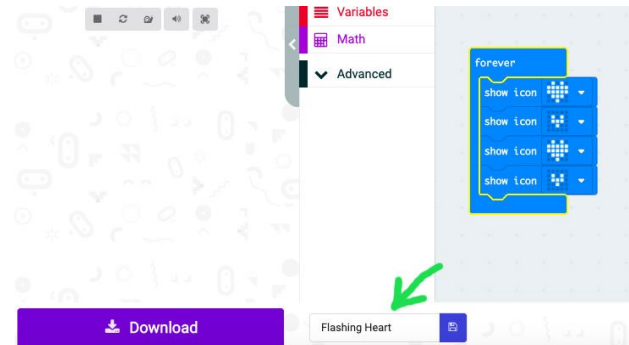


Give your partner a high five for writing your first lines of code together! Next, we'll save our code, then **upload** the code to the Micro:bit so we can see our animation on the Micro:bit. Navigators, watch closely and instruct your driver to do the following steps.

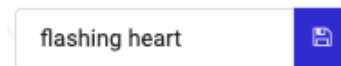
DEMONSTRATE AND HAVE MAKERS FOLLOW ALONG:



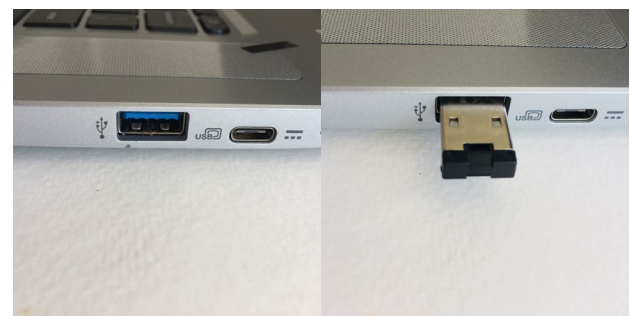
1. Give your project a unique name, like "Flashing Heart".



2. Save your file by clicking on the blue disk icon to the right of the file name.

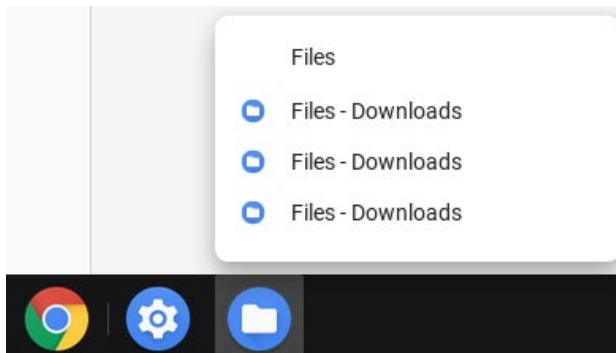


3. Connect your USB flash drive into a USB port on your computer.

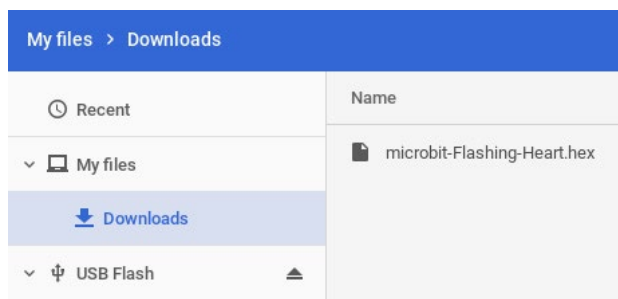


4. Go to the **Files-Downloads** folder on your computer. This is a digital folder that holds information. You can think of it like a backpack full of different folders. You should see both the file

you just named and your USB flash drive listed on the left.



5. Drag the file to the USB flash drive. You need to do this EVERY TIME to save your files so you can keep track and use them again in the future.



In the next step, we'll upload the code to our Micro:bit microcontroller!



Upload the code onto the Micro:bit microcontroller.

Makers will now connect the Micro:bit board to the laptop in another USB port to upload the code onto the Micro:bit. Remind them to be gentle, look carefully

at the shape of the ports, and not force the connection.

DEMONSTRATE AND HAVE MAKERS FOLLOW ALONG:



1. First, carefully connect the micro-USB cord to the board.



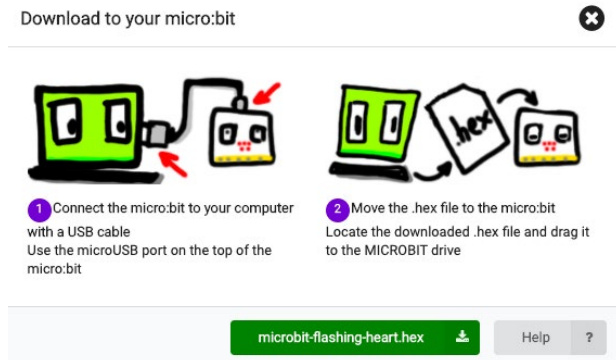
2. Next, connect the USB to the laptop. You should see the LED grid on the Micro:bit light up. It might have some previously saved code already displayed on it.



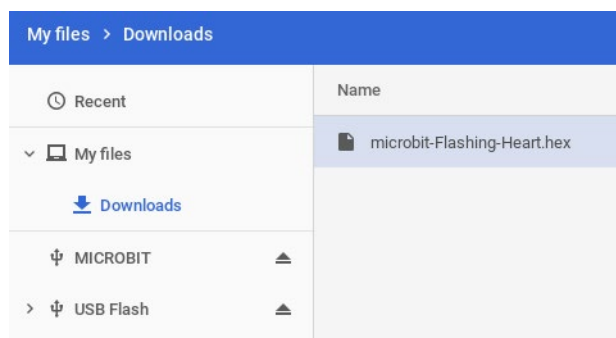
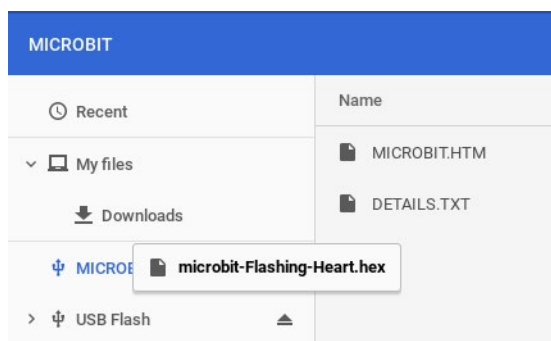
3. Once the board is connected, press the purple **Download** button in the MakeCode software.

 Download

You should see this window pop up:



4. Next, go to the **Files-Downloads** folder again in the computer's hard drive and find your named file. Click and drag it onto the Micro:bit icon.

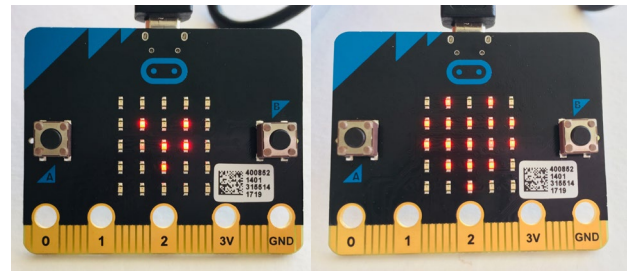


5. You should see a yellow LED on the Micro:bit flashing quickly. This means your code is being uploaded to the

Micro:bit. It will stop when the code is finished uploading.



6. Watch to see the display you created load up onto the Micro:bit display grid!



STEP 6



Connect the battery pack to the Micro:bit.

DEMONSTRATE AND HAVE MAKERS FOLLOW ALONG:



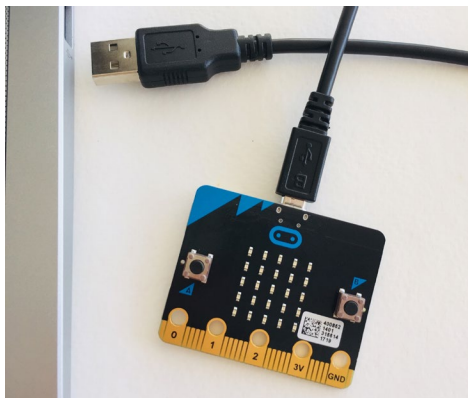
Next, it's time to eject the Micro:bit board from the laptop and connect it to an external battery pack!

1. First, we need to safely eject the Micro:bit from the computer by

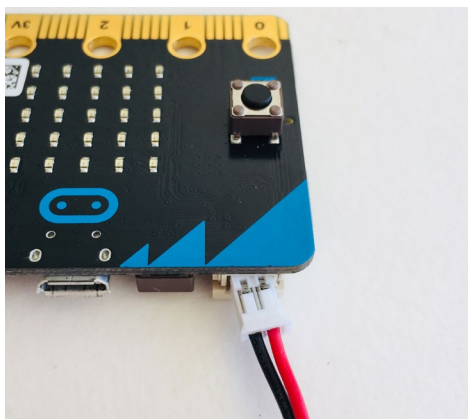
clicking the eject button next to the Micro:bit listed in the finder.



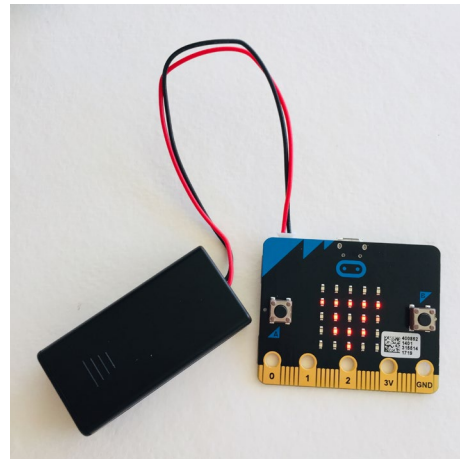
2. Next, safely remove the board from the USB cord. The board will no longer display the code because it isn't being powered.



3. Connect the external battery pack to the Micro:bit



4. You should now see your code displayed on the LED grid!



Once everyone is comfortable with the saving, downloading, and uploading process, they can reconnect the Micro:bit to the computer and continue exploring and coding displays for the Micro:bit.

STEP 7



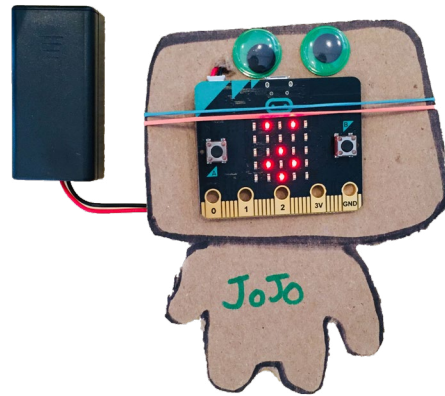
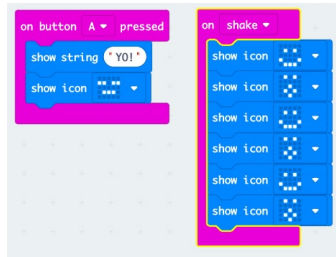
Clean up.

Makers will:

- Disconnect the battery pack.
- Put supplies and technology in assigned bins.
- Return laptops to cart and plug in for charging.
- Clear tables of garbage and recycling.

WEEK 2

DAY 2: CREATE A MICRO:BIT CHARACTER



INTRODUCTION

Now that makers have been introduced to the basics of the Micro:bit microcontroller, it's time to get creative with making a Micro:bit character. Makers work in pairs to co-design a character using maker supplies. Then, they will experiment with writing code to give the character a personality and make it interactive.



ESSENTIAL QUESTIONS

- What are inputs and outputs?
- How can we program them to tell stories or create personality characteristics?
- How do artists, engineers, and makers solve problems when they're working/inventing?



LEARNING OUTCOMES

1. Learn how to code both inputs and outputs in the MakeCode software.
2. Engage in project-based learning through problem-solving and troubleshooting by creating a character with a microcontroller and digital code.



VOCABULARY

Microcontroller: Small device (similar to a mini computer) used to control other devices and machines

Pair programming: Method of working in pairs where one person drives the laptop while the other navigates or instructs the driver

Software: Term used to describe computer programs

Code: Set of instructions that a computer understands

Input: Place where information enters a system

Output: Place where power or information leaves a system

Upload: Sending information or data to a device or computer

USB to micro-USB cable: Cord that transfers data and information from the computer to a device

USB flash drive: Portable storage device that can store and move files

Troubleshooting: Using resources to solve issues as they arise



MATERIALS LIST

EACH PAIR OF MAKERS NEEDS:

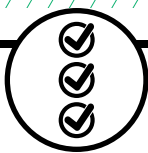
- Laptop with internet connection
- USB to micro-USB cord
- USB flash drive
- External battery pack
- AAA batteries (2)
- Notebook

ALL MAKERS NEED ACCESS TO:

- Cardboard
- Glue stick
- Markers
- Colored pencils
- Scissors
- Rubber bands
- Assorted paper
- Assorted maker supplies (pipe cleaners, pom-poms, googly eyes, etc.)

Items can be portioned out per table or set up in an area where students can access them freely.





TEACHER PREP WORK

1. Ensure the internet connection is working.
2. Connect your laptop to a projector or screen.
3. Preload videos and slideshow.
4. Organize maker supplies (pipe cleaners, pom-poms, googly eyes, etc.) into paper trays.
5. Go through the lesson and create your own example of a Micro:bit character to share as a helpful reference for makers troubleshooting their own projects.
6. Print the [Troubleshooting Tips](#) at the end of the lesson and post in the classroom.

FACILITATION TIPS

Make sure that you're comfortable with the save/download/upload process so you can help makers going through this process.

Materials management: It's up to you as the educator to decide what

works best for your class. You can portion out maker materials into paper trays for each table, or have a dedicated area where makers can access materials freely as needed.

Collaboration: Let smaller issues work themselves out. Record specific positive examples that you can share with makers in the moment or at the end of the project. These examples provide models for all learners.

Frustration: When frustration levels are not high, let learners figure it out on their own or keep facilitation at a minimum by asking a question and walking away. When frustration levels are high, intervene more directly to help makers find some success.

Circulate among the makers and monitor for both collaboration and frustration.

ADDITIONAL RESOURCES

[Micro:bit Project Tutorials](#)

[Wonderful Idea Co: Micro:bit Mutants](#)

[Micro:bit Tutorial on YouTube](#)

CREATE A MICRO:BIT CHARACTER

STEP 1



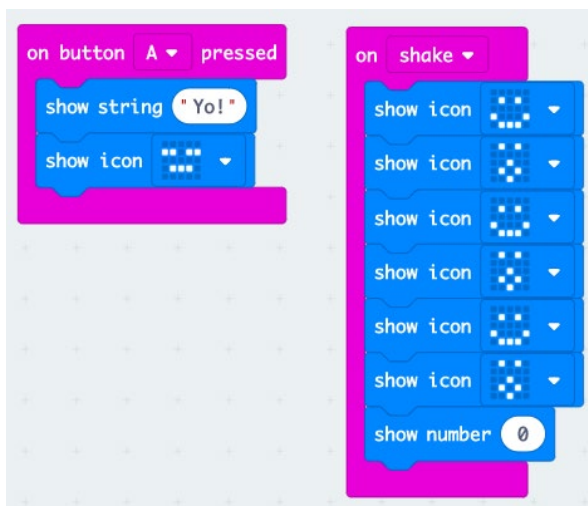
Code a character demonstration.

Gather makers around to view the projector as you demonstrate one possibility of coding a character in the **software**. Show them your example character that has the **code** below uploaded to the board.

EXPLAIN



This code could be an example for a character named Jojo. When you meet Jojo, instead of shaking hands or a fist bump, you press **Button A** and Jojo says “Yo!” and the LED grid displays a neutral face.



Jojo is a pretty cool and normal bot, but Jojo is also ticklish. When you **shake** Jojo, it “giggles” and shows

a laughing animated facial expression on the LED grid.

Button A and **shake** are both examples of **Input** blocks from the **Input** blocks menu. You can program them to cause various reactions, or outputs. These can be used in creative ways to code a character with personality.

STEP 2



Tinker with project-based coding and making.

Next, makers work on designing their own characters. While tinkering and experimenting with code, makers can use the maker supplies (cardboard, rubber bands, markers, googly eyes, etc.) to turn the Micro:bit **microcontroller** itself into a fun physical character with personality.

Note: Advise makers not to apply glue directly to the Micro:bit board.

Their design should include:

- 3 **inputs** that cause at least 3 different **outputs**
- A short story about their character—including name, personality, etc.—and how it relates to their programming

Makers will:

- Get their laptop and bins.
- Choose their **pair programming** navigator and driver roles (time for 15 min each).
- Tinker and explore programming various inputs and outputs in the software and simulator before connecting the Micro:bit to the computer.

Encourage makers to explore and experiment with other blocks within the **Basic** and **Input** blocks.

Remind makers to save changes to the code they want to keep onto their **USB flash drive**.

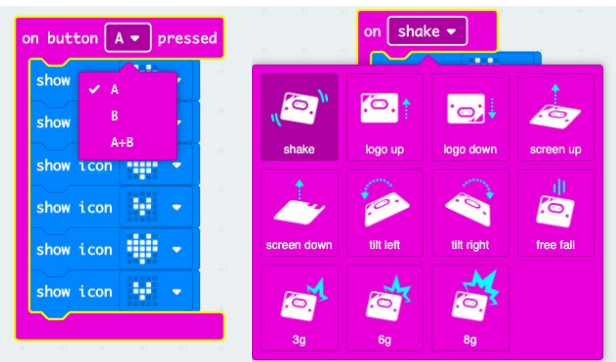
When they're ready to upload to the Micro:bit, support makers in connecting the **USB to micro-USB cable** to the computer to **upload** their code and connecting the external battery pack to the board to see their code working on

Basic

Input

Music

the Micro:bit independently from the laptop.



STEP 3



Document.

Makers will now spend a few minutes writing down their character features in their notebook. Here are examples of a character plan filled out with multiple inputs and outputs.

| INPUT | OUTPUT |
|----------|------------------------------|
| Button A | "I'm Hungry" + Neutral face |
| Shake | "Noooo!" + Animated sad face |
| Button B | "Yum" + Animated smile |

| INPUT | OUTPUT |
|----------|-------------------------|
| Button A | "let's dance!" 😊 |
| Button B | "That's all you got?" 😞 |
| Shake | 😊 🤖 😊 |

STEP 4



Share and reflect.

Ask for volunteers who would like to present their character to the rest of the group, and have the rest of the characters shared as an interactive gallery walk.

Ask makers to reflect on:

- What challenges did you encounter?
- How did you work through these challenges?



STEP 5



Clean up.

Makers will:

- Disconnect the battery pack.
- Put supplies and technology in their assigned bins.
- Return laptops to cart and plug in for charging.
- Return tools and materials that can be used again to the right place.
- Clear tables of garbage and recycling.

TROUBLESHOOTING TIPS

Display on board is not showing what we expected.

Version check

- Check to see if the latest saved copy of the code has been uploaded.
- Re-save the latest version and drag it onto the Micro:bit.

Bugs in code

- Check your code to look for mistakes.
- Check to see if there are extra blocks hiding in the coding space.

The LED on the Micro:bit isn't flashing when we press upload.

Bad cable or USB port

- If the Micro:bit doesn't show up in the computer's menu, try a different cable.
- Try a different USB port.

Code isn't uploading correctly to the board, and the board might feel hot.

Burnt board

- Press the reset button on the board, and then try uploading again.
- If the Micro:bit board feels hot, it could be overheated and stop working. Try uploading to a new Micro:bit board.

When we connect the battery pack, the Micro:bit doesn't power up.

Battery problems

- Check the batteries to see if they're in the right position (+/-).
- The batteries could be low or dead. Try replacing them.
- Battery pack could overheat or have stopped working. Try a new battery pack.

TROUBLESHOOTING TIPS

Print and use the empty rows to fill in with other problems and solutions that can be shared.

[illegible]