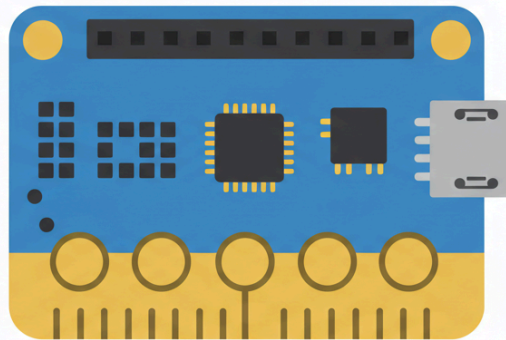


Getting to Know the micro:bit

Textbook

Getting to Know the micro:bit

As you read, remember to read carefully and pay close attention. Ask questions to help you understand the text. Focus on the main points of each section to understand the big ideas.



Imagine a robot that can clean your room, help build cars, or even explore outer space! Robotics is the exciting field where we create and use these amazing machines. Robots are special because they can be programmed to perform tasks on their own, or with very little help from people. They come in all shapes and sizes, from small toys to huge machines used in factories.

Robotics and Computer Science

[Robotics](#) and computer science go hand in hand, just like peanut butter and jelly! Computer science is all about writing code, which tells computers and robots what to do. When we build robots, we use computer science to program them, or give them instructions. For example, if you want a robot to clean a room, you need to write code that tells it how to find dirt and avoid obstacles. Without computer science, robots wouldn't know how to follow orders or perform tasks!

Robotics is important for many reasons:

- Solving Problems
- Making life easier:
- Creating new jobs

- Learning and Innovation

Now that you know what Robotics is, let's take a look at the micro:bit!

Getting to Know the micro:bit

First, watch this video to get more familiar with the micro:bit.



The [micro:bit](#) is a pocket-sized computer. The micro:bit listens to what you tell it to do. Writing these instructions is called programming. You can program this computer with code! The programs you write are sent to the micro:bit's processor, which is also known as the brains of a computer because it receives information and runs the programs you write.

Learn more about the micro:bit's processor by watching this video:



Input and Output

All computers have processors that run programs, otherwise known as [software](#). For example, when you tap a phone screen or press play on a video, you are giving the computer an input. Input is a place where information enters a system. The processor in the computer reads these inputs and tells the computer what to do. So if you were to press play on your favorite song, the processor would tell the computer to play it, and the computer would play the song through an output, like a speaker. An output is where information is shown or produced.

You can learn more about input and output devices by watching this video:



The micro:bit Simulator and How Simulations Work

So, where do we write these programs that will be played on the micro:bit?

This course uses the **micro:bit simulator**. It's like a pretend micro:bit right on your computer screen! You can write and test your code there without needing the actual micro:bit.

A **simulation** is like a special computer program that acts out how something real might work. The micro:bit simulator is one type of simulation! We use simulations to:

- **Solve Problems:** They help us understand tricky problems by letting us try different ideas in a safe, pretend world.
- **See What Happens:** We can change things (these are like inputs) and then watch what happens (the outputs) without messing up anything in real life.

Think about these examples of how simulations can show problems and solutions:

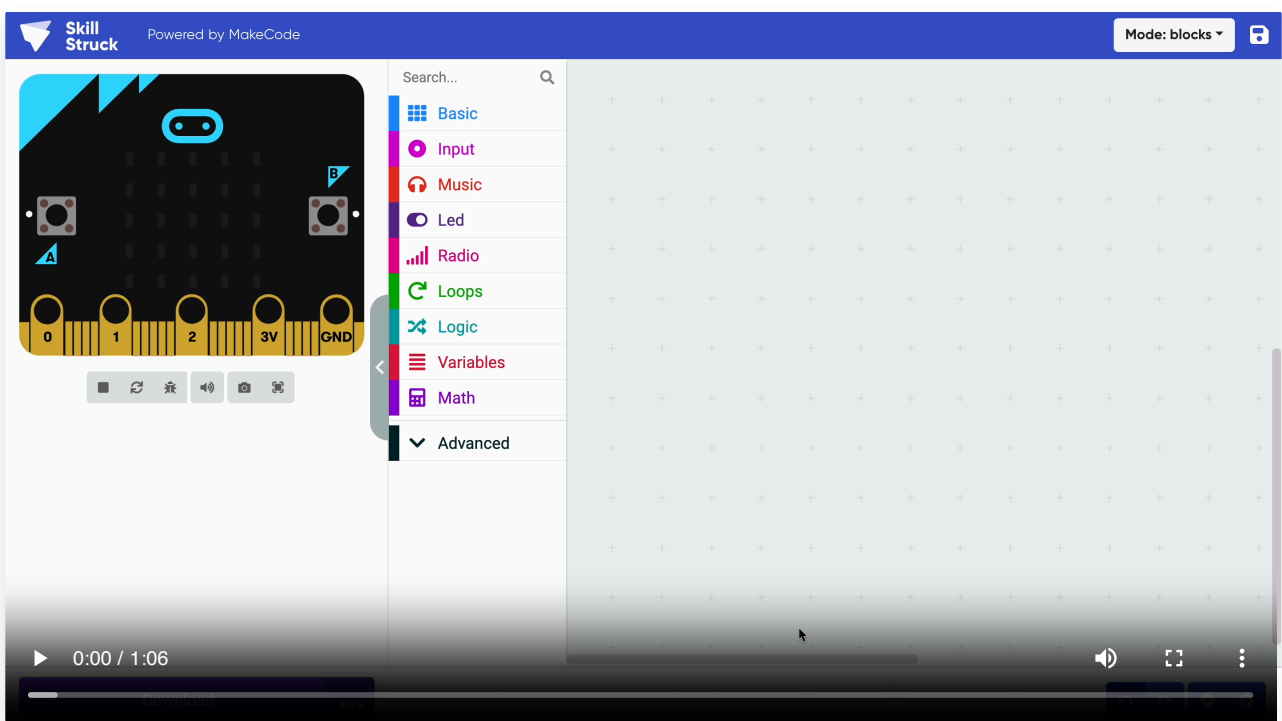
- **Ecosystem (Nature) Simulation:** Imagine a computer game where you have a virtual forest. You could add too many deer (an input), and the simulation might show you that all the trees get eaten (an output/problem!). Then, you could try a solution, like adding more food for the deer in another area, and see if the trees grow back.
- **Predator/Prey Simulation:** You could have a pretend world with virtual foxes (who hunt) and rabbits (who are hunted). If you start with too many foxes, the simulation might show you that all the rabbits disappear (a problem!). You could then try a solution, like reducing the number of foxes, and see if the rabbits can survive.

- **Invasive Species Simulation:** What if a new, fast-growing plant (an invasive species) is accidentally put into a virtual garden? A simulation could show how it quickly takes over, pushing out other plants (a problem!). You could then try different solutions in the simulation, like adding a special bug that only eats the invasive plant, to see if it helps the garden return to normal.

The micro:bit simulator helps you test your code safely, just like these other simulations help scientists test ideas about nature!

Let's go over how to use the micro:bit simulator:

- Read the challenge on the left side of the screen.
- Click through the tutorial at the top for a brief challenge walkthrough.
- Drag and drop the blocks into the code editor space to build your algorithm.
- When finished, click "Play" to watch your code on the micro:bit simulator.



Code It! – On Start

Let's practice by coding something together. First, the micro:bit needs to be told when to run the program. We do this by using the **on start** block. This tells the computer that when we start the program by pressing play, it will run our code.

1. Drag the **on start** block into your code editor.

Another important thing to know is what an icon is. An icon is a picture. For example, if I wanted the micro:bit to show a picture of a heart, I can also say I want it to show a heart icon. To do this:

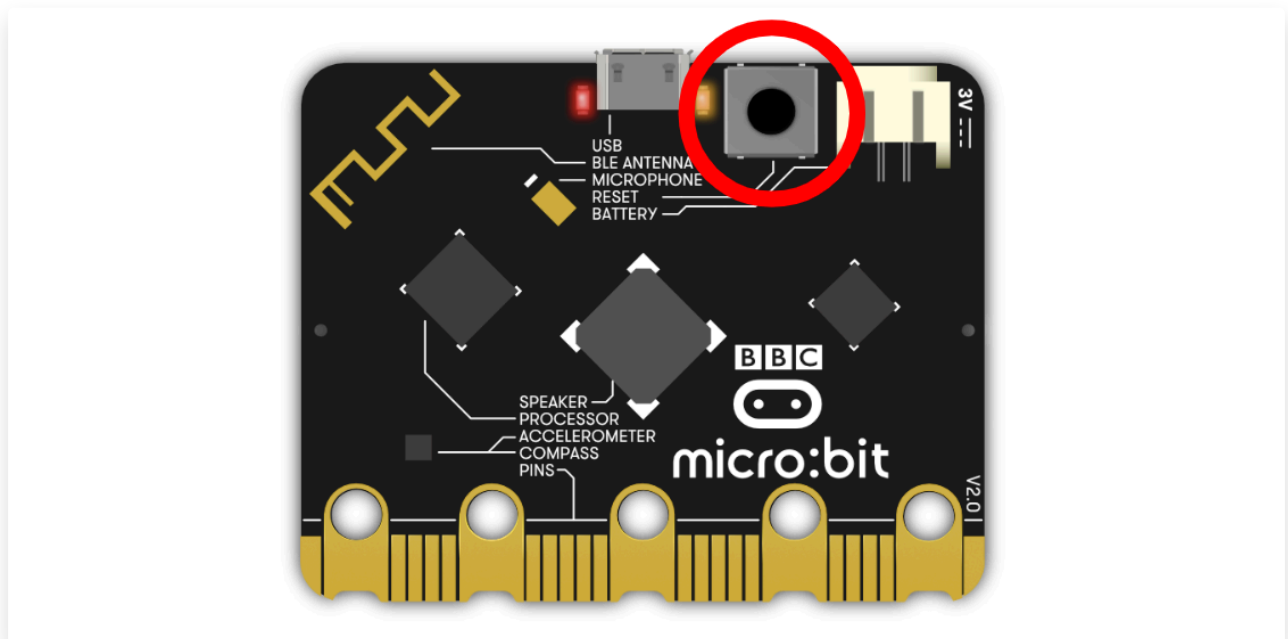
1. Drag the **show icon** block into the code editor, and connect it within the **on start** block.
2. Then, select the heart icon.

This is telling the micro:bit that when I press play to start the code, a heart will show. You try!

Functions and Features

The micro:bit has several functions and features. First, the micro:bit has [inputs](#), like the two buttons on the micro:bit that you can program separately or together. We will learn how to do this in future lessons. The micro:bit also has [outputs](#). The 2 outputs on the micro:bit are its speaker and LED display. LEDs are lights. There are 25 LEDs on the front of the micro:bit that are used to show words, pictures, and more!

Turning the micro:bit On and Off



To turn on the micro:bit, press the power/reset button. This will wake the micro:bit and restart your program to run the new code you create.

To turn off the micro:bit to save battery when you are done using it, hold the power/reset button on the back of the micro:bit.

Be Careful

The micro:bit has battery wires and fragile pieces. This means you need to be very careful when using the micro:bit so nothing breaks. Here are a few things to practice when using the physical micro:bit:

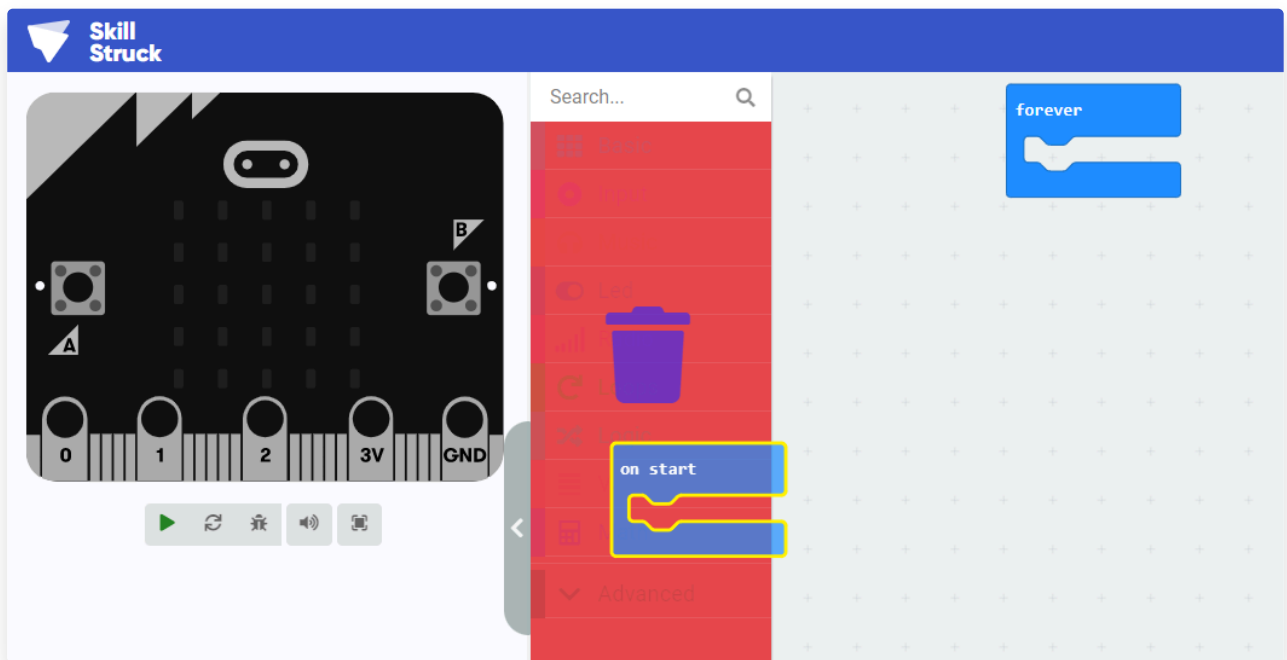
1. Use both hands to hold the micro:bit and its battery. Don't let the wires dangle when carrying it.
2. Always have clean hands when using the micro:bit.
3. Lay the micro:bit on a flat surface.
4. If working in a group, take turns!

Pressing Play

The simulator likes to automatically play your code. If you want to press the play button yourself, first press the square "stop" button and then the play button will show!

Deleting Blocks

To delete blocks in the code editor, drag the block to the left and drop it where the block categories are. As the block hovers over the block categories, a trash can will appear! This will delete the block(s) you drag and drop in this space.



Adopted from microbit.org platform

Critical Thinking Questions

1. How do robots help factory workers? Can you think of other ways robots could help people in different jobs?
2. If you were programming a robot to clean your room, what instructions would you give it to make sure it cleans well and avoids obstacles?

Questions (5)

1. Which code blocks would you use to make the micro:bit show a heart when you press play?

MULTIPLE CHOICE

Choose the correct answer:

- A. on start + show icon (heart)
- B. repeat forever + play sound
- C. pause + show number
- D. on button A pressed + clear screen

2. What should you do if you want to delete a block in the code editor?

MULTIPLE CHOICE

Choose the correct answer:

- A. Drag it to the left side until the trash can appears
- B. Press the micro:bit's power button
- C. Double-click the block
- D. Ask the robot to delete it

3. What will happen if you forget to connect your “show icon” block inside the “on start” block?

MULTIPLE CHOICE

Choose the correct answer:

- A. The heart icon will still show
- B. Nothing will happen when the code starts
- C. The micro:bit will turn off
- D. The buttons will stop working

4. You wrote code and clicked play, but nothing shows. What should you check first?

MULTIPLE CHOICE

Choose the correct answer:

- A. Make sure the blocks are connected correctly
- B. Throw the micro:bit away
- C. Tap the screen harder
- D. Draw the picture with a pencil

5. What is the micro:bit's processor most like?

MULTIPLE CHOICE

Choose the correct answer:

- A. A screen that shows pictures
- B. A battery that gives power
- C. A brain that runs your code
- D. A button that plays music

Games (2)

1. Getting to Know the micro:bit Typing



Full Screen

Audio

Instructions

Restart

Pause



0s 100%

Computer science is all a

2. Getting to Know the micro:bit Sorting

Full Screen

Audio

Instructions

Answer Key

Pause

Clear All

Check Order

Attempts: 0

Typing on a Keyboard

Printing

Tapping a Screen

Talking into a Microphone

Playing Music




Press Play on a Video

When a Phone Vibrates

Displaying an Image on a Screen

Input

Output



Robotics Challenges (4)

1. Heart

 Challenge  Textbook

Heart

Code the micro:bit to show a heart when you press play.

Requirements



Use the 'on start' block and show a heart icon

Answer Key 

Submit 

2. Smiley Face

 Challenge  Textbook

Smiley Face

Code the micro:bit to show a smiley face when you press play.

Requirements



Use the 'on start' block and show a smiley face icon

Answer Key 

Submit 

3. Three Icons

Challenge

Textbook

Three Icons

Code the micro:bit to show three icons when you press play.

Requirements

Use the 'on start' block and show three different icons

Answer Key

Submit

Step 1

Begin with the **on start** block.

Three Icons Step 1 of 6

1

Next

Toolbox

Search...

Basic

Input

Music

Led

Radio

Loops

Logic

Variables

Math

Extensions

Advanced

forever

Download

4. Five Icons

Challenge

Textbook

Five Icons

Code the micro:bit to show five icons when you press play.

Requirements

Use the 'on start' block and show 5 icons

Answer Key

Submit

Answer Keys & Solutions

Questions

1. Which code blocks would you use to make the micro:bit show a heart when you press play?

MULTIPLE CHOICE

Correct Answer:

- | | |
|---------------------------------------|-------------|
| A. on start + show icon (heart) | ✓ Correct |
| B. repeat forever + play sound | ✗ Incorrect |
| C. pause + show number | ✗ Incorrect |
| D. on button A pressed + clear screen | ✗ Incorrect |

Explanation:

To make the micro:bit show something when the code starts, use "on start" and connect a "show icon."

2. What should you do if you want to delete a block in the code editor?

MULTIPLE CHOICE

Correct Answer:

- | | |
|---|-------------|
| A. Drag it to the left side until the trash can appears | ✓ Correct |
| B. Press the micro:bit's power button | ✗ Incorrect |
| C. Double-click the block | ✗ Incorrect |
| D. Ask the robot to delete it | ✗ Incorrect |

Explanation:

Dragging the block to the left makes a trash can appear.

3. What will happen if you forget to connect your "show icon" block inside the "on start" block?

MULTIPLE CHOICE

Correct Answer:

- A. The heart icon will still show ✗ Incorrect
- B. Nothing will happen when the code starts ✓ Correct
- C. The micro:bit will turn off ✗ Incorrect
- D. The buttons will stop working ✗ Incorrect

Explanation:

The blocks need to be connected so the code knows what to do.

4. You wrote code and clicked play, but nothing shows. What should you check first?

MULTIPLE CHOICE

Correct Answer:

- A. Make sure the blocks are connected correctly ✓ Correct
- B. Throw the micro:bit away ✗ Incorrect
- C. Tap the screen harder ✗ Incorrect
- D. Draw the picture with a pencil ✗ Incorrect

Explanation:

Code only works if blocks are placed in the right spots.

5. What is the micro:bit's processor most like?

MULTIPLE CHOICE

Correct Answer:

- A. A screen that shows pictures ✗ Incorrect
- B. A battery that gives power ✗ Incorrect
- C. A brain that runs your code ✓ Correct
- D. A button that plays music ✗ Incorrect

Explanation:

The processor runs your program.

Games

1. Getting to Know the micro:bit Typing

Typing game - no answer key needed. Students practice typing the provided content.

2. Getting to Know the micro:bit Sorting

Category Solutions:

Category 1: Input

- Tapping a Screen
- Talking into a Microphone
- Typing on a Keyboard
- Press Play on a Video

Category 2: Output

- Printing
- Displaying an Image on a Screen
- Playing Music
- When a Phone Vibrates

Students must sort items into their correct categories.