

Exploring Microbits

60 minutes

Microbit Masterclass

WHAT THIS LESSON DOES

Get started with programming a microbit in this hands-on lesson. You'll learn to create a project, display messages, and make your microbit react to button presses. Follow the steps to code and explore its exciting features!

WHAT STUDENTS SHOULD BE ABLE TO DO

- Understand the basic functionality and features of a microbit.
- Create a new project using the MakeCode for microbit website.
- Use the Project Editor to write and simulate code.
- Program the microbit to display numbers and text on its LED grid.
- Program the microbit to respond to button presses with specific actions.

BEFORE THE LESSON

- Open makecode.microbit.org yourself first and confirm your school filter does not block it. Have it ready on the projector.
- One device per pupil or per pair, plus a physical microbit and a USB cable for each. Charge or check them beforehand.
- Sound on: step 9 plays a melody, so either speakers on the microbits or headphones per device.
- Know your school's download flow (pairing the microbit vs saving the .hex file), as this trips up the most in step 8.

Exploring Microbits

HOW THE LESSON RUNS

What is a microbit?

Create a new project

The Project Editor

Add Our First Code!

Delete Some Code

Display Your Name

Show a Happy Face

Connect your Microbit

Play Some Music

Explore!

TEACHING MOVES

1. What is a microbit? Hold up a real microbit and pass a few around so pupils see the LED grid and the two buttons before they touch a screen. Name buttons A and B.
2. Create a new project. Model New Project on the big screen once. Warn pupils the site may look busy at first and to wait for the editor to finish loading before clicking.
3. The Project Editor. Point to each area on the projector: simulator left, toolbox middle, code area right. Stress that on start runs once and forever repeats, so pupils know where their code goes.
4. Add Our First Code! Drag showNumber from the Basic category and change the value together. Have pupils watch the simulator refresh so they connect a code change to a screen change.
5. Delete Some Code. Show that dragging a block back to the toolbox deletes it. Reassure pupils that removing code is normal and not a mistake.
6. Display Your Name. Model the onButtonPressed A block with your own name, then click the simulator's A button so pupils see one letter scroll at a time. Then let them add their own.
7. Show a Happy Face. Add the B button icon block, then show the little down arrow that opens the icon library so pupils can pick their own icon instead of the default.
8. Connect your Microbit. Walk the room during this step. Connecting and downloading is where most stall. Do one full download slowly on the projector before pupils try their own.
9. Play Some Music. Add the A+B melody block and press both simulator buttons together. Let pupils swap in a melody from the gallery, then download again to hear it on the device.
10. Explore! Give a clear time limit and one rule: download after every change or the microbit will not update. Circulate and ask pupils to explain what a new block does.

WHAT TO WATCH FOR

- Pupils change code in MakeCode, look at the physical microbit, and think it is broken because nothing changed. Remind them the physical microbit only updates when they download again. The simulator updates live, the device does not.
- Pupils put button code inside on start or forever instead of its own onButtonPressed block, so it never responds to the button. Show that each button action needs its own onButtonPressed block. Point out the block name literally says which button it listens for.
- Pupils expect A and B pressed together to trigger the single A or single B code, and are confused when it does not. Explain that Button.AB is a separate event. All three, A, B and A+B, need their own block.

DIFFERENTIATION

- Emerging: Pre-open MakeCode and create the project for pupils who stall at the start, so they begin at the coding, not the setup. Pair them with a partner who can help find blocks in the toolbox by category colour.
- Developing: Have them talk you through which block goes with which button before they download, to check they understand the events rather than copying. Ask them to change the number and their name independently before you move the class on.
- Proficient: Point them at the Explore step early: add a third response such as reacting to shake, or change the melody's tempo. Ask them to explain their full sequence back to you, naming each button event and what it triggers.