

Program a micro:bit: follow precise instructions

Lesson at a glance

Open by naming the micro:bit as a real pocket computer that follows instructions exactly. Pupils create a MakeCode project on the IWB, meet the editor and simulator, then build step by step: show a number, delete it, program button A to scroll their first name and button B to show a heart. They download to the real device, explore one new block, and close with a display-only code talk on bugs fixed and what they'd change.

Learning objectives

- Program a micro:bit to show an output on a button-press input
- Frame the program as input, process and output and debug errors as they arise
- Explain that a micro:bit runs exactly the algorithm it is given

Before the bell – prep

Charge or battery-fit one micro:bit per pair and check USB leads work the day before. Open makecode.microbit.org on the IWB and on pupil devices; bookmark it. Have a spare micro:bit and lead ready for failed downloads.

Materials

Item	Qty	Per	Source	Low-cost substitute
micro:bit (with USB lead)	1	pair	school devices	run the lesson on the IWB using the MakeCode simulator

Safety watch-point

Handle USB leads gently; keep micro:bits clear of drinks. Remind pupils to use first names only on the LED scroll — no surnames or personal details.



Teaching moves

- **What we're building:** Set devices open in MakeCode with micro:bits nearby. Ask 'what does a programmer actually do?' and draw out precise instructions. Reassure that bugs are normal and part of the job.
- **What is a microbit?:** Read together and let pupils handle a spare micro:bit. Ask 'what can this tiny computer do?' and draw out inputs (buttons, movement) and outputs (lights, sound). No coding yet — orientation only.
- **Create a new project:** Model New Project and a clear name on the IWB. Watch for pupils who skip the naming box or open the wrong site. Remind them a clear name makes the project easy to find later.
- **The Project Editor:** Walk the screen slowly: simulator left, toolbox, code area with the two starting blocks. Ask 'where will we see our code run?' Clarify simulator is on-screen and the real device comes later.
- **Add Our First Code!:** Model the show-number block and watch the simulator refresh — name this as output. Change the number together. If nothing shows, the block is loose outside a starting block; snap it in.
- **Delete Some Code:** Model a clear drag back into the toolbox. Some pupils drop halfway and it snaps back — show the full drop. Reassure that deleting code is safe and normal.
- **Display Your Name:** Model typing a first name inside the quote marks on the IWB. Ask 'what is the input here, and what is the output?' Watch for names outside quotes or the show block sitting outside the button block. First names only.
- **Show a Happy Face:** Model the button-B block and choosing an icon from the arrow menu. Ask how this is like A and how it differs. If pupils press A expecting the icon, remind them each button has its own instructions.
- **Connect your Microbit:** Stagger pairs for the download — connections queue up. If nothing runs on the device, re-plug and download again. Celebrate the moment code runs in the real world.
- **Explore!:** Have pupils predict what a new block will do before running it, then check. Remind them to download after every change. Suggest one specific block for pairs who freeze; let confident pairs combine two behaviours.
- **Code talk — what worked and what we'd change:** Display-only talk — pupils do not type. Invite one pair to share a bug and exactly how they fixed it; revoice that the micro:bit ran their instructions exactly, so the fix was in the algorithm.

What it should show

On the real device, button A should scroll the pupil's first name one letter at a time; button B should show the heart (or chosen icon). If nothing appears, the usual causes are a failed download, the show block outside the button handler, or text typed outside the quote marks — re-plug, check nesting, and download again.

Misconceptions & interventions

- **Pupils think the on-screen simulator is the same as the real micro:bit, so they expect the handheld device to update without downloading.** — Hold up the device next to the simulator and say one is on-screen practice, one only changes after Download. Have them download once and press A on the real board.
- **Pupils put the show-string or show-icon block outside the on-button-pressed block, or type the name outside the quote marks, and nothing happens.** — On the IWB, nest the show block clearly inside the button handler and point to the quotes around the name. Press A on the simulator so they see the fix live.
- **Pupils think a bug means the micro:bit is broken, not that the instructions were imprecise.** — Revoice a pair's bug aloud and show the micro:bit did exactly what the blocks said. Ask 'what must we change in our algorithm?' so the fix lands on the code.

Differentiation

- {'emerging': ['Keep the pair at the teacher table for the first button-A block; provide the nested structure and let them type only the first name inside the quotes.', 'Suggest one specific block to try in Explore rather than free choice.'], 'developing': ['After A and B work, prompt a second trial: change the icon or add a short second message and re-download.', "Ask 'why did nothing show?' when a block is loose, so they name the nesting fix themselves."], 'proficient': ['Combine two behaviours or add a movement reaction during Explore, predicting before they run.', 'In code talk, critique whether their algorithm was precise enough and name one improvement for next time.']}

Cross-curricular hook

Tie to English procedural writing — pupils oral-sequence the steps of their algorithm as precise instructions another coder could follow.