

LESSON 40

Lists and working with data in Python

40 minutes

Strand 3: Coding at the next level

Built in Python

Outcome 3.2

Outcome 3.1

WHAT THIS LESSON DOES

Students use PRIMM to create and manipulate lists in MicroPython on the micro:bit, then loop over list data to build and combine LED light patterns.

CURRICULUM OUTCOMES

Outcome	What the specification asks for
3.2	realise short programming tasks using basic linear data structures (such as array or list)
3.1	creatively design and write code for short programming tasks to demonstrate the use of operators for assignment, arithmetic, comparison, and Boolean combinations

WHAT STUDENTS SHOULD BE ABLE TO DO

- Store and process a collection of values with a list (LO 3.2)
- Loop over a list to work with a small real dataset (LO 3.1)

BEFORE THE BELL

- Have micro:bits (or the simulator) ready and students logged in to the class Python setup
- Open the micro:bit Python editor at python.microbit.org and check that a new project loads cleanly
- If using physical boards, confirm USB cables or Bluetooth pairing works before the lesson
- Plan to model the first create-and-retrieve steps on the board before pairs work independently

WHAT YOU NEED

Item	Qty	Per	If you do not have it
tablet or laptop with Python	1	pair	run the lesson on the IWB with students taking turns

Lists and working with data in Python

HOW THE LESSON RUNS

4 min	introduction	Start: what we're building
2 min	content	Why use a list?
2 min	content	What is an Array?
2 min	content	Create a List
2 min	content	Retrieve List Elements
3 min	content	Change List Elements
4 min	content	Add Elements to a List
4 min	content	Remove Elements from a List
2 min	content	Project: LED Light Patterns
2 min	content	Display Pattern1
2 min	content	Modify the Patterns
3 min	content	Merge and Show
2 min	content	Wrap Up
3 min	content	Make sense: what worked, what broke
2 min	content	Reflect
1 min	summary	What you covered

TEACHING MOVES

- Stress that list indexes start at 0; write a short labelled list on the board so students can point to index 0, 1, 2
- When students reach the for loop, model reading it aloud: for each image in the list, show it, wait one second
- Keep the early steps tight so there is real time for the patterns project, where lists become a small dataset
- Treat debugging as a named skill: when a display shows the wrong item, ask which index they meant before editing the code

WHAT TO WATCH FOR

- Off-by-one index (wanting the third item but using [3]): point at the board list and count from 0
- Calling append or pop but then showing an old index that no longer matches: have them display the length or the last item after the change
- Forgetting sleep after display.show so images flash too fast to see: add the one-second pause and run again

DIFFERENTIATION

- Support: provide a written index strip (0 to 4) beside the starter list and stay with pairs on retrieve and change before they move on
- Extension: challenge students to design a third pattern list or change the merge order and timings
- Extension: ask stronger pairs to remove by value and by index in one run and explain the difference