

Checking Your Data: Finding and Fixing Errors

60 minutes

ICT2: Introduction to other software packages and applications

Outcome M2.U1.2

WHAT THIS LESSON DOES

In this lesson you will learn to spot and correct common spreadsheet error codes. You will trace broken formulas, perform sanity checks on your budget data and ensure all numbers are reliable before final submission.

CURRICULUM OUTCOMES

Outcome	What the specification asks for
M2.U1.2	Create a spreadsheet and enter numeric and character data, apply formulas while understanding their purpose to generate results, format cells and generate a chart and print it.

WHAT STUDENTS SHOULD BE ABLE TO DO

- Recognise common errors including #DIV/0!, #VALUE!, #REF! and #NAME?
- Trace and fix broken cells with a sanity-check habit
- Review formulas for sensible results against real project numbers
- Run a sanity check on the budget, fix issues and note what changed

BEFORE THE BELL

- At the door or login, check that every student has 05_project_budget in Project Portfolio > 05_budget. Anyone missing the file makes a new one from the Missing file? box in the introduction (four headings, five of their own costs and a SUM total) before Error Lab, which takes a few minutes.
- Nothing to pre-fill for the Error Lab: every student builds the Error_Lab tab from the five headings and four sample rows printed in the click-steps. Pair a slower typist with a partner who reads the values aloud.
- Students need cloud access to Project Portfolio > 05_budget > 05_project_budget. Anyone whose file is missing makes a new 05_project_budget from the Missing file? box in the introduction. Sort missing files at login.
- No prep needed beyond the Key concepts table on screen if you are projecting.
- Students need Excel for the web or Google Sheets open on Project Portfolio > 05_budget > 05_project_budget. Nothing to pre-fill: students build the Error_Lab tab from the data printed in the click-steps.
- Have a short sample budget ready to project for the 90-120 second audit model. Leave the sample log on screen during independent practice.
- Students work in their own 05_project_budget for 15 minutes. Be ready to help locate broken absolute references from earlier autofill work, and to help anyone who rebuilt their file today.

Checking Your Data: Finding and Fixing Errors

HOW THE LESSON RUNS

5 min	introduction	Introduction
5 min	content	Key concepts
18 min	activity	Step-by-step task
7 min	content	Common issues
15 min	activity	Independent practice
5 min	content	Reflection
5 min	summary	Lesson summary

TEACHING MOVES

- Circulate during Error Lab so students work on the Error_Lab tab, not the live budget sheet.
- Demonstrate #REF! on the board: delete Quantity, show every dependent formula flip to #REF!, then Undo. After Undo, ask two quick oral checks: What does #REF! mean? What is the fastest rescue after a bad delete? Anyone who wants a 30-second try may have one.
- Before releasing independent practice, model on a projected sample (or one real student total): read the formula bar, change one cost by €1 and watch the total move, put the real figure back, and type one log line. Leave the model log on screen for the whole of the independent practice.
- Release the independent practice in sequence: first the red codes are cleared, then one rough estimate is said out loud with a partner, and then the log is filled in. Spot-check logs in the last minutes: a date, a cell or range, and a fix (or "none") beat a vague "all good". Circulate first with the students who rebuilt their budget file today.

WHAT TO WATCH FOR

- Confusing #VALUE! with #NAME?: #NAME? is almost always a spelling problem in the function name; #VALUE! is a data-type problem in the cells.
- Treating a clean calculation as finished work: push the sanity-check habit (does this total match real project costs?).
- Minimal logs with no date or cells named: guide students to the modelled four-column layout under the data.

DIFFERENTIATION

- Support: Pair a slower typist with a partner who reads the Error Lab headings and sample rows aloud, so that time is spent on diagnosing and fixing rather than data entry.
- Support: Pair early finishers with peers to review formula bars together.
- Extension: Only after the Sanity-check log is done, invite the IF warning on Error_Lab only, not on the live budget.