

Google Sheets Automation & Scripting — Workbook

This workbook turns the course into hands-on practice. Each section maps to one module and mixes exercises, fill-in worksheets, and checklists so you build a real library of macros, formula reports, validation rules, and scheduled triggers as you go. Work in a copy of a real spreadsheet you actually use, and by the end you will have automated at least one weekly task end to end.

Automating With the Macro Recorder

Record, name, and stress-test your first reusable macros so a repeated click sequence becomes a single keystroke.

Exercise: Record the Monday Cleanup macro

Open a copy of a spreadsheet you clean up regularly. Using Extensions, Macros, Record macro with relative references, capture your real cleanup steps, save it as Monday Cleanup, and assign Ctrl+Alt+Shift+1. Then paste a fresh export and run it.

- List the exact manual steps you take to clean this sheet each week, in order.

- Did you choose absolute or relative references, and why was that right for this task?

- Time yourself doing it by hand once, then by macro once. Record both durations.

- What broke or behaved oddly on the first replay, and how did you fix it?

Worksheet: Macro inventory and shortcut map

Fill in one row per macro you create. Use a verb-plus-object name and keep shortcuts memorable. This becomes the Automations tab you maintain in every workbook.

Macro name (verb + object)

What it does (one sentence)

Absolute or relative references

Keyboard shortcut

Range it touches (e.g. A1:F1000)

Last tested date

Checklist: Macro reliability check

- Macro name follows the verb-plus-object convention
- Reference mode (absolute vs relative) was chosen deliberately
- Hard-coded ranges were widened to cover future rows
- Macro re-runs cleanly on a second batch of test data
- Macro is listed on the Automations tab with its shortcut

Formula-Driven Automation

Replace copy-paste reporting with self-updating QUERY, FILTER, and ARRAYFORMULA pipelines.

Exercise: Build a self-updating regional report

Using your own data or the practice dataset template, write a single QUERY that totals an amount column grouped by a category column, sorted highest first. Then add a new row to the source and confirm the report updates with no further action.

- Write your finished QUERY formula here exactly as it appears in the cell.

- Which clauses did you use: select, where, group by, order by, limit?

- Add a row to the source. Did the report update automatically? Note what changed.

- What error did you hit first, and which single change fixed it?

Worksheet: Formula automation planner

For each manual reporting task you currently do, decide which formula replaces it and capture the plan before you build.

Current manual task

How often you do it

Best-fit function (QUERY / FILTER / ARRAYFORMULA / UNIQUE)

Source range

Where the live result will live (tab and cell)

Blank-guard needed (IF cell is empty)?

Exercise: Auto-fill a column with ARRAYFORMULA

Pick a column you usually drag a formula down. Replace it with a single blank-guarded ARRAYFORMULA in row 2 so every current and future row fills automatically.

- Write the ARRAYFORMULA you used, including the IF blank-guard.

- Confirm the formula sits only in row 2 and spills downward.

- Add three new rows of data. Did they fill automatically with no dragging?

Checklist: Live-report readiness

- QUERY references source columns by letter within the stated range
- Spill area below and right of FILTER/SORT is kept empty
- ARRAYFORMULA columns are blank-guarded so empty rows stay blank
- Risky lookups are wrapped in IFERROR with a readable fallback
- Report updates automatically when a source row is added

Clean Data With Validation and Rules

Stop bad data at the point of entry with dropdowns, validation rules, and reactive conditional formatting.

Worksheet: Validation rule plan

Map each input column to the validation rule that should protect it before you share the sheet with others.

Column name

Allowed values or range

Rule type (dropdown / number between / date / checkbox / custom formula)

Reject or warn on invalid input?

Source list location (if dropdown from range)

Exercise: Make a tracker flag itself

On a task or invoice tracker, add a custom-formula conditional formatting rule that turns a row red when it is both past due and not marked Done, plus an amber rule for due within three days.

- Write the custom formula you used for the red overdue rule.
-

- Write the custom formula for the amber due-soon rule.
-

- Which rule did you place first, and why does order matter here?
-

- Change a status to Done. Did the red highlight clear automatically?
-

Exercise: Add a quality-check column

Create a Check column that labels any problem row as Review using IF and OR, and a duplicate flag using COUNTIF. Then FILTER for Review to see only rows needing attention.

- Write your Check formula that returns Review for bad rows.
-

- Write the COUNTIF formula you used to flag duplicates.
-

- How many rows were flagged, and what were the three most common problems?
-

Checklist: Data-quality guardrails

- Every input column has a validation rule set to reject invalid entries
- Dropdowns exist for all category and status columns
- Conditional formatting flags overdue and duplicate values automatically
- A Check column surfaces problem rows behind a Review filter
- Validation was applied before the sheet was shared with others

Scheduled and Connected Automation

Run work unattended with time-driven triggers, Forms intake, and Connected Sheets, then document the whole system.

Exercise: Schedule a recorded macro

Take a macro that refreshes or re-sorts a report and attach a daily time-driven trigger via Extensions, Apps Script, Triggers, set between 6 and 7 a.m. Test it on a copy first.

- Which macro did you schedule, and what does it do?

- What trigger type and time window did you set (Day, Week, or Minutes timer)?

- Confirm you tested it on a copy before live data. What did you verify?

- What would happen if this ran on the wrong range every night, and how did you guard against it?

Worksheet: Connect a Form for live intake

Bind a Google Form to a sheet and plan the processing that runs on each submission. Keep all calculations off the columns the Form controls.

Form purpose (requests / signups / feedback)

Response tab name

Helper column formula (score or classify each response)

Summary tab QUERY

On-submit trigger action (confirmation email / alert)?

Worksheet: Automation system register

Document every automation in this workbook so a colleague could maintain it without you. This is the master Automations tab.

Automation name

Type (macro / formula / trigger / import / form)

What it does (one line)

Schedule or trigger (if any)

Source files or IMPORTRANGE links

Last tested date

Checklist: Ship one automation end to end

- Chose one painful weekly task to fully automate
- Selected the right layer (macro, formula, validation, trigger, or import)
- Tested the automation on a copy before trusting live data
- Documented it on the Automations register with owner and schedule
- Estimated the annual hours saved and noted it for review

Your Action Plan

1. Spend two weeks logging every repetitive spreadsheet task and how often you do it.
2. Run the four-line decision rule on each task to label it macro, formula, validation, trigger, or import.
3. Record your top three repeated click sequences as named macros with clear shortcuts.
4. Replace your most-rebuilt report with a single self-updating QUERY or FILTER formula.
5. Add dropdowns and validation rules to every input column before sharing the sheet.
6. Add conditional formatting and a Check column so problems and duplicates flag themselves.
7. Convert your most fragile copy-down column to a blank-guarded ARRAYFORMULA.
8. Schedule one recorded macro with a time-driven trigger after testing it on a copy.
9. Connect a Google Form or IMPORTRANGE so at least one data source flows in automatically.
10. Fill in the Automations register and book a quarterly review to retire stale automations.

