

LEARNING PATH / 100-140 minutes

Your own project with AI

Codex, files and checking the result

What you will practise

Specify and test a simple file-reporting script. Learn to distinguish generated code from a tested solution.

Before you begin

Basic familiarity with files and folders is helpful. Agree access to Codex and a script runtime in advance. Without that access, prepare the specification and test plan, then carry out the exercise with the trainer.

Working principle

First define the goal, scope and acceptance criteria. Ask for a plan before changes. Work in a separate folder containing fictional files. Review the code and planned changes before running it. Ask the trainer to check any file operations or network connections you do not understand. AI saying 'done' is not a test result.

How to use this workbook

Complete the exercises and save each prompt and result before checking the answer guide. AI responses may differ; assess them against the criteria rather than expecting identical wording. Timings are approximate self-study estimates, not a separate offer of training hours.

Exercise data

All examples are fictional. Do not enter personal data, passwords, secrets or confidential documents. For workplace exercises, use only material approved for the chosen tool. Do not send drafted messages to real recipients.

WORKSHOP / 40 minutes

Specify a file report

Source material

Create an ai-exercise folder containing input and results subfolders. In input, create three empty files: offer.txt, note.txt and photos.csv, plus an archive subfolder containing old.txt. Do not use client files. The goal is a CSV listing names and sizes in bytes of files directly in input, without visiting subfolders.

Your task

Write acceptance criteria. Ask Codex for a plan and an explanation of where it will write files. After reviewing the plan, ask it to prepare code inside the exercise folder. Agree an available programming language with the trainer. Do not install dependencies solely for this example.

Starting prompt

Plan a simple script with no external dependencies. Read only the list of files directly in input, not their contents or subfolders. Write results/report.csv with headers name,size_bytes, using UTF-8, a comma separator and alphabetical ordering by name. Do not change source files or use the network. If the report exists, stop without overwriting it. First show the plan, assumptions and tests; wait for approval before creating code.

Save your result

A specification, approved plan and code ready for review.

Reflection

What did AI get right? What needed correcting? How will you change the prompt next time? Record your answers in your working document.

WORKSHOP / 40 minutes

Test and hand over the solution

Source material

Use the previous exercise data. Also test an empty input folder, a missing input folder, an existing report.csv and a filename containing spaces and accented characters. Use separate copies of the exercise folder for the tests.

Your task

After review, run the script with fictional data. Open the CSV and compare it with the folder. Test the additional cases. Preserve the existing report before the overwrite test and confirm its contents are unchanged. Record errors, fix the code and repeat the tests relevant to the change.

Starting prompt

Prepare running instructions and a test table matching the specification. Distinguish planned tests from completed tests. Do not claim to have run code without execution evidence. For an error, identify the cause and smallest correction. Do not expand the scope to moving or deleting files, or adding integrations.

Save your result

The CSV report, actual test results and short running instructions.

Reflection

What did AI get right? What needed correcting? How will you change the prompt next time? Record your answers in your working document.

REVIEW AND FURTHER PRACTICE

Check your result

Answer guide

For the three empty files, the report has the header `name,size_bytes` and three rows with size 0. It excludes `archive` and `old.txt`. An empty input folder produces only the header. A missing input folder should produce a clear error and no report. An existing report remains unchanged. Filenames with spaces and accented characters must survive writing and reading. Source files must remain unchanged. Success means verified test results, not merely generated code.

Completion criteria

- Reads and writes stay within the exercise scope.
- There are no network operations, deletions or report overwrites.
- Normal, empty, missing-folder and existing-report cases have been tested.
- The instructions make the test repeatable and the result has execution evidence.

When to move on

Move on when you meet every criterion above and can explain your corrections. If a result contradicts the source, revise the prompt and try again. This is a skills self-assessment, not certification.

Apply it to your work

Describe a small task from your work that needs neither client data nor production access. Before creating code, define the inputs, output, prohibited actions, rollback method and acceptance criteria.

Where this fits

Foundations / Working with documents / Analysis and decisions / Repeatable workflows / Your own project with AI. These are five levels of practice material. For a four-hour workshop, I select exercises appropriate to the group; completing the entire path requires additional practice. Confirm the available workshop language before booking.