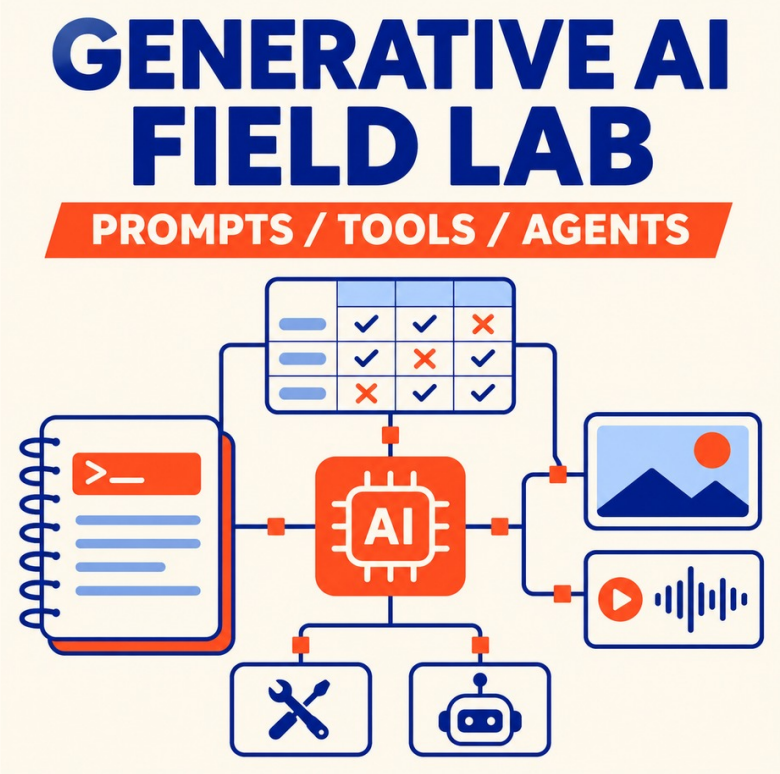
Generative AI Field Lab

36 labs for ChatGPT workflows, coding, creative media, and agents.



HOW TO USE THE WORKBOOK

Read the brief, run the example with the supplied fixture, and use the checks to judge the result. Save a draft and one revision. Copyable examples and the offline Python capstone are in the bundle.

Use a chat tool for prompt exercises and a media tool for generation exercises. Review tool availability and terms for your account. API-shaped examples and the capstone can be studied with local fixtures.

Find your starting point

CHATGPT AND WORKFLOWS

01 Write an answerable brief / p. 5

02 Extract with a source trail / p. 8

03 Teach categories with examples / p. 11

04 Test a prompt change in isolation / p. 14

05 Summarize a decision with conflicting notes / p. 17

06 Verify the claim before citing it / p. 20

07 Reconcile a small table by hand / p. 23

08 Make a plan that fits the clock / p. 26

09 Prepare a safe practice packet / p. 29

10 Build a repeatable handoff / p. 32

CODING AND API PRACTICE

11 Reproduce a Bug Before Asking AI to Fix It / p. 35

12 Review an AI Patch Against Existing Behavior / p. 38

13 Build a Request Before Sending It / p. 41

14 Specify a Structured Output Contract / p. 44

15 Validate Outputs Before Using Them / p. 47

16 Compare Two Prompt Revisions with an Eval / p. 50

17 Retrieve Relevant Passages Locally / p. 53

18 Ground an Answer in Retrieved Evidence / p. 56

19 Distinguish Prompt Caching from RAG / p. 59

20 Choose Retrieval or Fine-Tuning with Evidence / p. 62

Create and verify

IMAGES AND AUDIO

21 Specify a usable campaign image / p. 65

22 Control the crop across formats / p. 68

23 Edit one region and protect the rest / p. 71

24 Build a consistent illustration series / p. 74

25 Turn a message into a storyboard / p. 77

26 Write text alternatives that carry meaning / p. 80

27 Prepare a narration brief and read it aloud / p. 83

28 Document voice permission and reuse / p. 86

29 Caption the audio and inspect sync / p. 89

30 Review a media bundle before release / p. 92

AGENTS AND VERIFICATION

31 Give an agent one narrow tool / p. 95

32 Treat tool output as evidence / p. 98

33 Ask for approval at the action boundary / p. 101

34 Budget the workflow before it runs / p. 104

35 Measure failure cases before a release / p. 107

36 Ship a support desk simulation / p. 110

Set up your evidence log

Keep each run small enough to inspect. Record the inputs, model or tool if used, output, checks, and the exact change you make next. Review factual claims against their source before publishing.

```
Lab / date
Goal and fixture
Tool or offline adapter
Output or path
Expected behavior
Observed behavior
Next change and reason
```

CAPSTONE COMMAND

Extract the ZIP and run python support_lab.py with Python 3.12. The capstone uses the standard library and synthetic data. Its simulated dispatch writes to an in-memory log.

REFERENCE READING

ChatGPT prompting | https://learn.chatgpt.com/docs/prompting

Responses API | https://developers.openai.com/api/docs/guides/migrate-to-responses

Structured outputs | https://developers.openai.com/api/docs/guides/structured-outputs

Function calling | https://developers.openai.com/api/docs/guides/function-calling

Prompt caching | https://developers.openai.com/api/docs/guides/prompt-caching

Evaluation best practices | https://developers.openai.com/api/docs/guides/evaluation-best-practices

Write an answerable brief

CHATGPT AND WORKFLOWS

A useful prompt gives the model a job it can finish with the inputs you supplied. Separate your task, the audience, the raw material, and the output contract. Then decide how you will recognize a good response before you open the chat. In this exercise, a plain baseline and a more specific brief answer the same narrow question.

BUILD AND OBSERVE

01 Copy the workshop notes in the fixture. Write a baseline prompt that only asks for an announcement, then save the first response. Keep the notes unchanged between runs.

02 Paste the full example into a fresh chat. Mark the instruction that defines the audience, the clause that sets the length, and the line that handles the missing fee. These are three independent changes to your baseline.

03 Compare each response to the notes and the checks below. Mark any added claim, missing detail, or confusing call to action. Count the body words using your editor.

04 Change a single part of the prompt, such as the audience or word limit. Run it again and describe the effect. Save the strongest prompt with the source notes and expected facts.

Write an answerable brief

WORKING EXAMPLE

Use this example with the lab brief. Find the unwrapped text in the bundle's examples folder.

```
Draft a short announcement for adult beginners who have never repaired a bike.
Give a subject line and a 65-90 word body. Open with who the session is for. End
with the exact registration action from the notes. Preserve the date, location,
and seat count. If asked about the fee, say it is awaiting confirmation.

Workshop notes
Tuesday, 18 May 2027, 6-7 p.m.
North Hall, Room 2. Twelve seats.
Bring a bicycle or use the demo bike.
To register, reply BIKE to the organizer.
Fee awaiting confirmation.
```

Write an answerable brief

ACCEPTANCE CHECKS

01 The body contains Tuesday, 18 May 2027, 6-7 p.m., North Hall Room 2, and twelve seats.

02 The call to action asks readers to reply BIKE to the organizer. Any fee statement says it is awaiting confirmation.

03 The body is 65-90 words and gives a first-time reader a clear reason to attend.

TRY A VARIATION

Reuse the same notes for a 35-word community notice. Score it separately because it has a different length and purpose.

RECORD YOUR RESULT

What did you observe?

Which check caught a failure?

What did you change for the next run?

Extract with a source trail

CHATGPT AND WORKFLOWS

Extraction turns a messy record into something you can check cell by cell. Give each source line an identifier. Pair each extracted value with its line identifier, and reserve a literal value for an absent field. Keep names, dates, and statuses exactly as the source gives them until you intentionally run a separate cleanup pass.

BUILD AND OBSERVE

01 Read the fixture and mark every line that commits to an action. Underline conditional language separately. Make your own expected table before running the example.

02 Run the example. For each reported task, reopen the cited line and check the owner, deadline, and wording. Pay special attention to rows where a date appears elsewhere in the notes.

03 Add the line L5: The team wondered whether to order more cable ties. Rerun the extraction and check that the number of committed actions stays the same.

04 Choose how your downstream tracker represents missing values. If it needs a blank cell, change the template to convert the literal value in a documented export step.

Extract with a source trail

WORKING EXAMPLE

Use this example with the lab brief. Find the unwrapped text in the bundle's examples folder.

```
Extract only actions explicitly committed in the notes. Return a table with
Action | Owner | Due | Source line. Use Unassigned for a missing owner and Not
stated for a missing date. Keep due wording verbatim. Put tentative proposals in
a separate list.

L1 Mei will count the tool kits by 16 May.
L2 If the weather clears, we could move outside.
L3 Tomas will confirm the room booking.
L4 The group agreed to print eight signs by Friday. Nobody took ownership.
```

Extract with a source trail

ACCEPTANCE CHECKS

01 The table has three rows with source lines L1, L3, and L4. L2 appears in the tentative list.

02 Tomas has Not stated as the due value. The signs row has Unassigned as owner and Friday as due.

03 Each source line supports every populated value in its row.

TRY A VARIATION

Add a short evidence quote column. Use the fewest consecutive words needed to show each commitment.

RECORD YOUR RESULT

What did you observe?

Which check caught a failure?

What did you change for the next run?