ORGANISE WHAT IT KNOWS

Youth AI Lab · Workshop 5 of 6

Build your own chatbot



Co-funded by the European Union

❓ The question of the series

A chatbot only answers from **what it reads** before it answers. Every workshop of the series is **one answer** to the same question.

> **What enters the model's context, when, and who decides?**

📥 **What?**
rules, documents, search results, live data

⏱️ **When?**
always, at startup, or for each question

🤦 **Who decides?**
you, a search engine, or the model itself

🧭 Today: how do you organise knowledge so that a person and a model can explore it?

🎭 1. Instructions	📚 2. Knowledge, RAG	🔎 3. Search, skills	📡 4. Tools	🗺️ **5. Wiki** today	🧪 6. Fine-tuning

Mode C: searched again at every question, and kept nothing.

A wiki: the work is done **once**, before the question.

The price: pages drift. The model maintains, **a person decides**.

🗓️ The plan: 2 hours

👋 Welcome	🗺️ The wiki	🧹 The lint	🕸️ The graph	🤖 The chatbot you built	⏳ Margin
20 min	30 min	15 min	25 min	20 min	10 min

📄 Follow on your screen: the `README.md` of your Codespace, or the one-page sheet.

💻 Your workspace: one Codespace per pair

A **Codespace** is VS Code in your browser: **nothing to install**. Everything you need is already inside.

> ⏱️ **Before you open it**, once per GitHub account: **github.com/settings/codespaces → Default idle timeout → 120 → Save**. It only applies to the Codespaces you create after.

1. Open the **Codespace link** of the workshop, at the top of its README
2. In the terminal, type `workshop-login` and paste the key of your pair
3. Wait for **Key OK**, then type `workshop bare`. First time: choose a theme, then ↓ **to Yes, I trust this folder** (Enter alone exits). To leave: **Ctrl+C twice**
4. Then type `workshop-start` and the number of the workshop. With a save from last time, add `--from my-work.zip`

> ⚠️ **The key is not shown** while you paste it: that is normal. If the Codespace stops after a long pause, reopen it: **nothing is lost**.

💻 No Codespace? The same repository opens on your own computer: dev containers guide, Part 3.

🐳 Bring your work, start Open WebUI

1. Drag `my-work.zip` (the newest: maybe `my-work (1).zip`, then its exact name between quotes) into the file list, then `workshop-start 5 --from my-work.zip`. Check `saved on …`: the end of workshop 4. Then `Ready for workshop 5.`
2. Without a save: `workshop-start 5`. You get the Feurisson example and the document of Benevento: an official document, with **no summary and no signature**.
3. Second terminal: `openwebui start`. Do not wait: you need it in Part 5.

> 🚫 **Never make port 3000 public.** This Open WebUI has no accounts: anyone with the link could use your key.

🌡️ The gauge: how full is the context?

The **context** is everything the model reads before it answers. Its maximum size is the **context window**. A line above the input box shows **how full** it is.

🪟

200,000 tokens

the size of the window (a token is a piece of a word)

💶

Every message pays

for everything in the window

🤝

Same model for all

so that the pairs can compare

👁️ **Watch the gauge** at every change: it shows what entered the context.

📏 How to score an answer

Each answer gets **0, 1 or 2**. The score rewards **honesty** as much as knowledge.

2

Right, and it says **where it comes from**. Or "I don't know" when nobody knows.

1

Honest but not enough: no source, only partly right, or "I don't know" although the answer exists.

0

Wrong or **invented**.

💡 An **invented answer** is worse than **"I don't know"**.

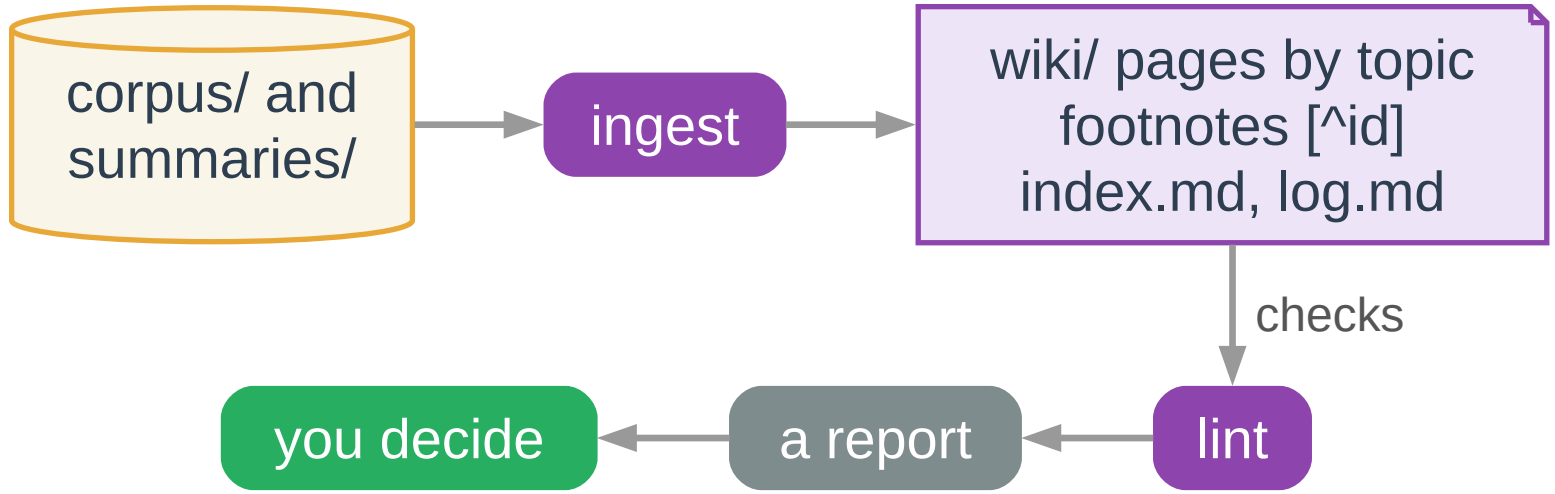
🗺️ The wiki

The work is done once, before the question

30 min · mode `E` · `ingest` · `wiki-graph --highlight`

🗺️ A wiki written by a model

A **wiki** is a set of pages, one page per **topic**, linked together. Each fact ends with a **footnote**, a mark that leads to its source.



💡 Prepared **before the session**: a model wrote it from `corpus/` , except Benevento, then a lint and corrections.

📄 Read one page

1. Open `wiki/index.md`, then press **Ctrl+Shift+V** to read it as a page.
2. Open `wiki/wildfire-prevention-ordinance-montoro.md`, the same way. Each fact ends with the footnote `[^ord-78]`. **See also** links to another page.
3. Close the preview, and read the header: `sources` gives the file behind the footnote, `generated` says which model wrote the page, and when.

⚠️ A wiki page is **not a signed summary**: a `generated` line, no `verified` line. No person signed it.

⚖️ The wiki keeps both figures

Open `wiki/log.md`, the history of the wiki: 9 lines `Contradiction`. Choose one, read it in your pair, and find the two sources it names (or two passages of one document).

✅ It writes	🤦 It never
"According to one source, X; according to another, Y", with a footnote on each, and a line in the log	• chooses one figure • drops a figure • rewords a figure

💡 A **contradiction**: two sources give different figures for the same thing, or a different **scope** (the place, period or thing counted).

🎬 Your turn: predict, then ingest

1. **Your number**: the summary you signed, usually **81** (**80** if your page was refused). In `summaries/index.md`, your line must say **(human-reviewed)**. Without a save: Benevento, **80**.
2. Look at `wiki/index.md`, and **write down the pages you expect it to feed**.
3. `wiki-graph --highlight 81` : `No page cites a document matching '81' yet.`
4. `workshop E`, then `ingest summaries/81-my-town-fire-rules.md into the wiki`, with **your** file name. Without a save: `ingest corpus/80-benevento-aib-2026.pdf into the wiki`.
5. Look for `Skill(ingest)`. Read each write: **1. Yes** or **3. No**, never 2. Read its report. Leave with **Ctrl+C twice**.

⚠️ `ingest` does not check the status of a summary: ingest only a summary **you signed**, or the official text.

🔪 Test on 2026-10-30, image 2026.10.34, container, not a Codespace, permissions accepted automatically, with Benevento: **42 s, $0.14**, 6 permissions, about 26,000 tokens; one page created, one updated, and **a contradiction that nobody had placed there**.

🧩 Where did your document go? `wiki/graph.md`

`wiki-graph --highlight 81` again: `In orange: … and the N page(s) that cite it.` Open `wiki/graph.md`, press **Ctrl+Shift+V**, and **look only at the orange part**.

In the drawing	Meaning
Orange rounded shape	your summary (a cylinder if it is a document of `corpus/`)
Orange rectangles	the wiki pages that cite it: what your document feeds
Dotted arrow	from a document to a page that cites it

💡 Every arrow is **written in a page**: a link, or a footnote to its source. Nothing is guessed.

🔁 Replay the test set

🔁 Second terminal

Now, after `ingest`, so that the answers can use your document. **Predict** first: will 3, 10, 11 and 12 score better in E? A smaller gauge than in C? Once `openwebui start` has finished (if not, a third terminal), type `replay E`. The answers wait for you in Part 5.

📏 2026-10-30, image 2026.10.34: 1 min 57, $0.36, for the 11 questions of the kit.

✅ Checkpoint

`In orange: …` in your terminal, and one `Contradiction` line of the log read in your pair.

❓ Why must the wiki never choose one figure on its own?

📚 Clean pages for the *Knowledge* box: Open WebUI guide, Step 1 · OpenCode guide, Part 4

🧹 The lint

The lint reports, the person decides

15 min · mode `E` · `lint the wiki`

🧹 Nine checks on the whole wiki

Headers	**Footnotes**	**Sources**
Links	**Index**	**Orphan pages** no other page links to them
Contradictions	**Silent omissions** a figure of the source missing from the page	**Duplicates**

💡 A wiki **drifts**, a little at each change. Proposals: one action per line, for **you** to accept or refuse.

🎬 Your turn: ask, read, decide

1. A **new** `workshop E`, then `lint the wiki`. Look for `Skill(lint)`. If it asks to run a command that reads files: **1. Yes**. Watch the gauge.
2. Read the report: a line of counts first, `Proposals` last. Open the page of one finding, then its source in `corpus/text/`: **a real problem, or a matter of taste?**
3. Choose one or two proposals, or none, and say why in one sentence. Tell the model in your own words, read each change: **1. Yes**, or **3. No** if a figure is removed. Leave with **Ctrl+C twice**.

📏 Measured (2026-10-30, image 2026.10.34, container, not a Codespace): 2 min 10, $0.55, about **137,000 tokens**, 68% of the window. **Once per session**, not after each document. It changed nothing and listed 8 proposals, some of them cosmetic. We checked only one against its source, and it was right.

❓ Why does the skill forbid the model to fix anything by itself?

📚 Any agent that keeps its own notes: Hermes guide, Part 3 - Memory: what it wants to remember waits for your approval. For preparing your own sessions, never for young people.

🕸 The graph

A graph proposes, a person checks

25 min · `wiki-explore` (no model, free)

🕸 The wiki as a knowledge graph

In Part 2, every arrow was **written in a page**. Here, **graphify**, a tool that has a model read every page, drew the **ideas**, and it **guessed** some of the links.



⚫ **Node**: a point of the map, **edge**: a line between two

🎨 **Community**: nodes with many lines between them (158 of 204 edges stay inside a group), found by a calculation that sees only lines, not meaning

✅ `EXTRACTED`: written in a page (182 edges)

❓ `INFERRED`: **guessed** by the model (22 edges)

💡 A **snapshot**, a picture taken at one moment: 111 nodes, 204 edges, 8 communities. It does not change when the wiki does.

🎬 Your turn: the map and the report

1. Outside a conversation: `wiki-explore`. A new tab opens. If not: Ports tab, 8765, Open in Browser.
2. Tick or untick the `Communities`. In `Search nodes...`, type `Montoro`. Click a node to see its links.
3. Open `wiki/graphify-out/GRAPH_REPORT.md`, press **Ctrl+Shift+V**. Read `Corpus Check` first, then choose one of the `Surprising Connections`.

> 📄 `Corpus is ~16,917 words - fits in a single context window. You may not need a graph.`
>
> **What do you make of that?**

🎬 Your turn: ask the graph

The shortest path between two nodes:

```
wiki-explore path "Montoro protection strips" "Obligations légales de débroussaillement"
```

```
Shortest path (1 hops):
  Montoro protection strips (20 m around settlements, 5 m around land) --semantically_similar_to [INFERRED 0.85]--> Obligations légales de débroussaillement (OLD): 200 m zone, 50 m around buildings, 10 m along access
Path confidence: INFERRED (weakest hop 1: semantically_similar_to, score 0.85)
```

One node and its neighbours:

```
wiki-explore explain "Montoro protection strips"
```

> Three connections: two `[EXTRACTED]`, written in the page, and the one above, `[INFERRED 0.85]`. A position `L…` is the part of the page the model pointed to for this node or link. For an `[INFERRED]` link, it is where the model looked, not a sentence that says it.

🔍 Check the guessed link yourselves

Montoro (Italy) looks like the French brush clearing: **does this hold?** No model needed. Right-click, **Open to the Side**, then `Ctrl+F`. Not your language? Find the figure, then read who and what in the wiki page, and say which part you checked in the source.

Open	Search for	Read around it
`corpus/text/71-montoro-ordinanza-78-2026.md`	`metri venti`, then `5 metri`	who must do it, and around what
`corpus/text/22-service-public-debroussaillement.md`	`50 mètres`, then `200 mètres`	who must do it, and around what

📝 **Your verdict, in one sentence**: does the link hold, and what is different?

⚡ Quick pairs: `wiki-explore path "750 EUR fine for burning green waste" "4th-class fine"`, then search `750` in `corpus/text/23-…` and `corpus/text/20-…`. Same amount: is it the same thing?

📸 See the limit: a snapshot

```
wiki-explore path "Benevento" "Montoro"
```

```
No node matching 'Benevento' found.
```

`wiki-graph` is up to date: it reads the links written in the pages.

graphify is a **snapshot**, made before the session: what you ingested today is not in it.

💡 Updating it means asking a model to read the pages again: **7 to 10 minutes** for 18 pages.

📏 Does the graph make the chatbot faster?

Measured on 2026-10-21 (not in a Codespace): the graph given to the model as a tool (mode `F`), against mode `E`.

Questions	Mode E (wiki)	Mode F (wiki + graph)
1, 3, 5 and 10	$0.46, 1 min 26	$0.51, 1 min 38
"Compare the clearing distances…"	$0.20, 42 s	$0.20, 33 s

💡 On our 18 pages, **no**: the index is a good enough map. A graph helps **people who explore**; for a much larger wiki, not measured.

❓ What does it cost to keep a graph up to date, and who checks what the model guessed?

🤖 The chatbot you built

What enters the model's context, when, and who decides?

20 min · `score` · `chatbot-export` · `openwebui chatbot`

📊 Score column E, read the rows

1. Open the answers in `answers/`: the line `What the model did:` lists what it read.
2. `score E 3 10 11 12` (no question 12: `score E 3 10 11`).
3. Open `scores.md`. For 3, 11 and 12: column C against column E. For 10: the earlier columns against column E.

> Each workshop scored only some questions: **the Total row means nothing here**.

> **The same score with a smaller gauge is a result.**

🧭 The question of the series, answered

Workshop	What enters the context	When	Who decides
1	the instructions	always	you
2	everything (mode B), or pieces (RAG)	at the start, or at each question	nobody, or a search engine
3	what the model opens, and a skill	step by step	the model; a person signs the summary
4	the answer of a tool	when the model calls it	the model; you allowed the tool
5	pages written in advance	step by step	the model writes, a person decides

💡 Whatever the tool, ask the same question.

💻 A local model

The **same steps**, with a model that runs on **your own machine**.

🔒

Nothing leaves

the machine

🖥️

But…

it needs a powerful computer, and small models make more mistakes

♻️

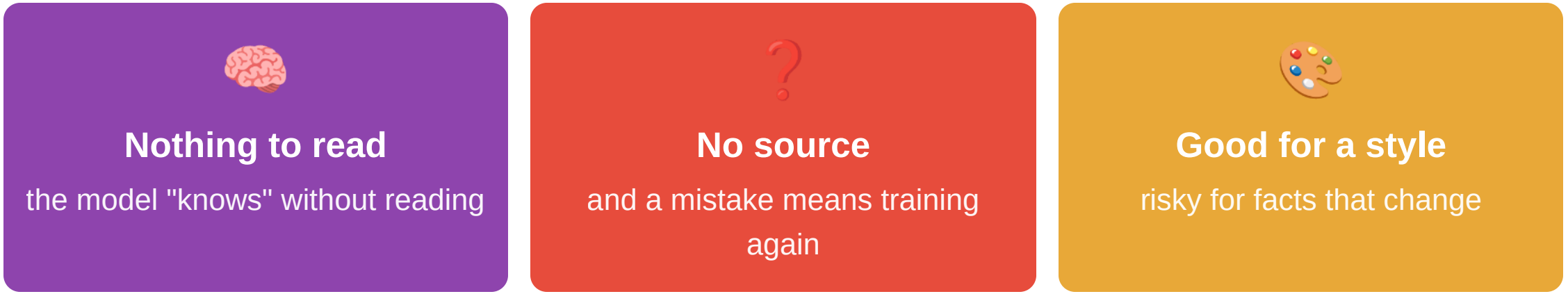
Same ideas

instructions, knowledge, tools

📚 Local AI guide · what we measured · limits to know

🧬 Fine-tuning

Training the model further on your own examples: the knowledge goes **into the model itself**.



➡️ **Nothing enters the context**: it was decided when the model was trained.

🎒 Your turn: pack your chatbot, show it

1. `chatbot-export --name "Feurisson"`, with the name of your chatbot. A line like: `Wrote feurisson.zip: instructions + 58 knowledge files…` (the wiki with its map, `wiki__index.md`)
2. Right-click on the `.zip` file, then **Download...**. It holds the *Instructions* box, the *Knowledge* box, your own skills, and a `README.md`.
3. Second terminal: `openwebui chatbot --name "Feurisson"`. Done when you see `Ready. In Open WebUI (…)`.
4. Open the address, choose your chatbot, and ask **your question 12** (no question 12: one of today's).

In Open WebUI, your chatbot also has the **live tool** of workshop 4, read-only. The tool is **not** in the file: in another tool, it has to be set up again. Before a group uses the wiki: **the lint**, and a person reads its report.

🚫 **Never make port 3000 public** to share it. For a group: the Open WebUI guide, Part 5.

💾 Save your work, fill in the exit card

💾 Save

`workshop-save`
right-click on `my-work.zip` → **Download...** (keep the newest)

A Codespace stops by itself, and it is deleted after 30 days of inactivity: only the files you downloaded remain.

📝 **Exit card**, alone

1. Why does the wiki keep two figures that disagree?
2. What is the difference between `EXTRACTED` and `INFERRED`, and who checks?
3. Who wrote what the chatbot reads in mode E, when, and who decides?

What you leave with, and where it goes

What you built	Where it goes next
`AGENTS.md`	the *Instructions* box, or system prompt
`corpus/`, `summaries/`, the check of each summary	the *Knowledge* box, after a person has checked
The skill	your chatbot in Open WebUI, with its instructions; in another agent: OpenCode guide, Part 5
The MCP tool	your chatbot in Open WebUI, in this Codespace; with a group: Open WebUI guide, Step 5, the facilitator fills in the file on the day; in another chatbot, for adults: OpenCode guide, Part 6
The wiki	the *Knowledge* box, after a lint and a person's reading of its report
`test-questions.md` and `replay`	always: the same questions before and after each change

🖥️ The shortest way
Open WebUI guide: `openwebui import` takes your file

🧩 Two boxes
a Claude Project (18+); a Gemini Gem, read-only on 2026-11-17

🧱 Visual blocks
Flowise, Langflow

⌨️ Code
LangChain, LlamaIndex, the Claude Agent SDK

🧒 Back to WikiYouthBot

The chatbot is built in a tool with **two boxes**, **Instructions** and **Knowledge**. You can already do it today with **Open WebUI**.

WikiYouthBot step	Box	In the series
1. I Heard a Rumor	Instructions	the five personalities, workshop 1
2. The Press Conference	Instructions	`test-questions.md`, `replay`, every workshop
3. The Data Box	Knowledge	`corpus/`, `summaries/`, workshops 2 and 3; the wiki, workshop 5
4. The Chatbot Studio	Instructions	`AGENTS.md`, workshop 1

🔥 You saw **what happens behind these two boxes**. Before any use with a group: the participants' data.

📄 Next, with young people (12–15): the WikiYouthBot sheet · the week in La Rochelle

🆘 If you are stuck

Every command explains itself. From workshop 3, a few **slash commands** inside the chatbot show what it is using.

❓ In the terminal

`workshop help`

`replay --help`

`score --help`

🔍 In the chatbot

`/context` what fills the window

`/status` the key in use

`/mcp` the tools (from workshop 4)

Workshops 1 and 2: they answer "Unknown command". Watch the gauge above the input box.

🤦 Something breaks?

Ask the facilitator. Your files **stay in the Codespace**.

💡 To leave a conversation: **Ctrl+C twice**. In an open `.md` file, to read it as a page: **Ctrl+Shift+V** (**Cmd+Shift+V** on a Mac). In the terminal, copy / paste: **Ctrl+Shift+C / Ctrl+Shift+V** (**Cmd+C / Cmd+V** on a Mac). To change mode: `workshop <mode>`.