March 2025

Search posts from March 2025 　Search

89 posts: 11 entries, 59 links, 18 quotes, 1 note

March 1, 2025

llm-anthropic #24: Use new URL parameter to send attachments. Anthropic released a neat quality of life improvement today. Alex Albert:

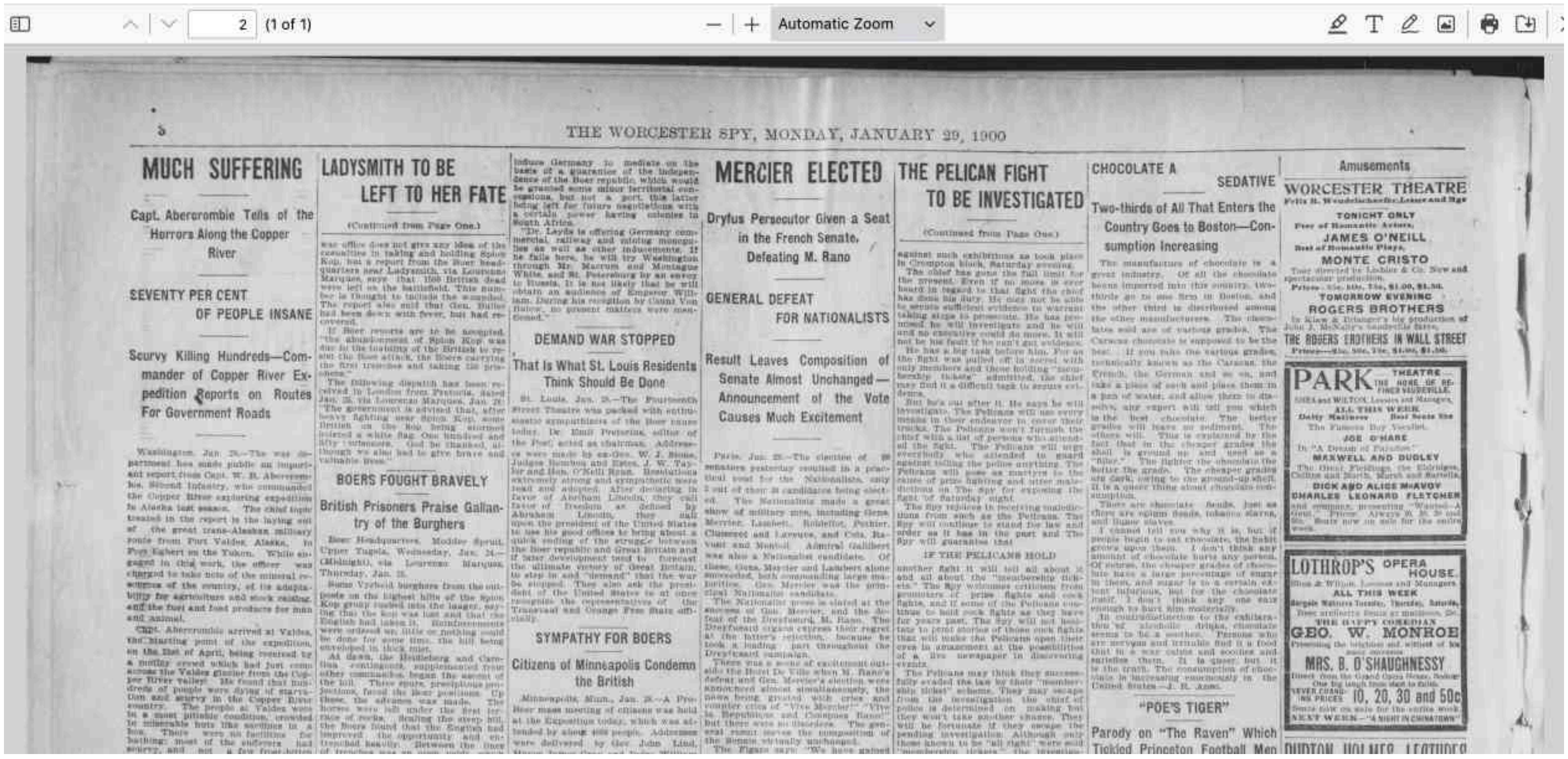
> We've added the ability to specify a public facing URL as the source for an image / document block in the Anthropic API

Prior to this, any time you wanted to send an image to the Claude API you needed to base64-encode it and then include that data in the JSON. This got pretty bulky, especially in conversation scenarios where the same image data needs to get passed in every follow-up prompt.

I implemented this for llm-anthropic and shipped it just now in version 0.15.1 (here's the commit) - I went with a patch release version number bump because this is effectively a performance optimization which doesn't provide any new features, previously LLM would accept URLs just fine and would download and then base64 them behind the scenes.

In testing this out I had a *really* impressive result from Claude 3.7 Sonnet. I found a newspaper page from 1900 on the Library of Congress (the "Worcester spy.") and fed a URL to the PDF into Sonnet like this:

```
llm -m claude-3.7-sonnet \
  -a 'https://tile.loc.gov/storage-services/service/ndnp/mb/batch_mb_gaia_ver02/data/sn86086481/0051717161A/1900012901/0296.pdf' \
'transcribe all text from this image, formatted as markdown'
```



THE WORCESTER SPY, MONDAY, JANUARY 29, 1900

MUCH SUFFERING

Capt. Abercrombie Tells of the Horrors Along the Copper River

SEVENTY PER CENT OF PEOPLE INSANE

Scurvy Killing Hundreds—Commander of Copper River Expedition Reports on Routes For Government Roads

[illegible]

LADYSMITH TO BE LEFT TO HER FATE

(Continued from Page One.)

[illegible]

BOERS FOUGHT BRAVELY

British Prisoners Praise Gallantry of the Burghers

[illegible]

[illegible]

DEMAND WAR STOPPED

That Is What St. Louis Residents Think Should Be Done

[illegible]

SYMPATHY FOR BOERS

Citizens of Minneapolis Condemn the British

[illegible]

MERCIER ELECTED

Dryfus Persecutor Given a Seat in the French Senate, Defeating M. Ranc

GENERAL DEFEAT FOR NATIONALISTS

Result Leaves Composition of Senate Almost Unchanged—Announcement of the Vote Causes Much Excitement

[illegible]

THE PELICAN FIGHT TO BE INVESTIGATED

(Continued from Page One.)

[illegible]

IF THE PELICANS HOLD

[illegible]

CHOCOLATE A SEDATIVE

Two-thirds of All That Enters the Country Goes to Boston—Consumption Increasing

[illegible]

"POE'S TIGER"

Parody on "The Raven" Which Tickled Princeton Football Men

Amusements

WORCESTER THEATRE

TONIGHT ONLY

JAMES O'NEILL

MONTE CRISTO

TOMORROW EVENING

ROGERS BROTHERS

THE ROGERS BROTHERS IN WALL STREET

PARK THEATRE

JOE O'HARE

MAXWELL AND DUDLEY

DICK AND ALICE McAVOY

CHARLES LEONARD FLETCHER

LOTHROP'S OPERA HOUSE

GEO. W. MONROE

MRS. B. O'SHAUGHNESSY

10, 20, 30 and 50c

[illegible]

I haven't checked every sentence but it appears to have done an excellent job, at a cost of 16 cents.

As another experiment, I tried running that against my example `people` template from the schemas feature I released this morning:

```
llm -m claude-3.7-sonnet \
  -a 'https://tile.loc.gov/storage-
services/service/ndnp/mb/batch_mb_gaia_ver02/data/sn86086481/0051717161A/1900012901/0296.pdf' \
  -t people
```

That only gave me two results - so I tried an alternative approach where I looped the OCR text back through the same template, using `llm logs --cid` with the logged conversation ID and `-r` to extract just the raw response from the logs:

```
llm logs --cid 01jn7h45x2dafa34zk30z7ayfy -r | \
  llm -t people -m claude-3.7-sonnet
```

... and that worked fantastically well! The result started like this:

```
{
  "items": [
    {
      "name": "Capt. W. R. Abercrombie",
      "organization": "United States Army",
      "role": "Commander of Copper River exploring expedition",
      "learned": "Reported on the horrors along the Copper River in Alaska, including starvation, scurvy, and mental
illness affecting 70% of people. He was tasked with laying out a trans-Alaskan military route and assessing
resources.",
      "article_headline": "MUCH SUFFERING",
      "article_date": "1900-01-28"
    },
    {
      "name": "Edward Gillette",
      "organization": "Copper River expedition",
      "role": "Member of the expedition",
      "learned": "Contributed a chapter to Abercrombie's report on the feasibility of establishing a railroad route up
the Copper River valley, comparing it favorably to the Seattle to Skaguay route.",
      "article_headline": "MUCH SUFFERING",
      "article_date": "1900-01-28"
    }
```

Full response here.

# 1:20 am / projects, ai, generative-ai, llms, llm, anthropic

March 2, 2025

Hallucinations in code are the least dangerous form of LLM mistakes

A surprisingly common complaint I see from developers who have tried using LLMs for code is that they encountered a hallucination—usually the LLM inventing a method or even a full software library that doesn't exist—and it crashed their confidence in LLMs as a tool for writing code. How could anyone productively use these things if they invent methods that don't exist?

[... 1,052 words]

6:25 am / ai, openai, generative-ai, llms, ai-assisted-programming, anthropic, claude, boring-technology, code-interpreter, ai-agents, hallucinations, coding-agents

18f.org. New site by members of 18F, the team within the US government that were doing some of the most effective work at improving government efficiency.

> For over 11 years, 18F has been proudly serving you to make government technology work better. We are non-partisan civil servants. 18F has worked on hundreds of projects, all designed to make government technology not just efficient but effective, and to save money for American taxpayers.
>
> However, all employees at 18F – a group that the Trump Administration GSA Technology Transformation Services Director called "the gold standard" of civic tech – were terminated today at midnight ET.
>
> **18F was doing exactly the type of work that DOGE claims to want – yet we were eliminated.**

The entire team is now on "administrative leave" and locked out of their computers.

But these are not the kind of civil servants to abandon their mission without a fight:

> **We're not done yet.**
>
> We're still absorbing what has happened. We're wrestling with what it will mean for ourselves and our families, as well as the impact on our partners and the American people.
>
> But we came to the government to fix things. And we're not done with this work yet.
>
> More to come.

You can follow @team18f.bsky.social on Bluesky.

# 9:24 am / government, political-hacking, politics, bluesky

---

> Regarding the recent blog post, I think a simpler explanation is that hallucinating a non-existent library is a such an inhuman error it throws people. A human making such an error would be almost unforgivably careless.

— **Kellan Elliott-McCrea**

# 2:16 pm / ai-assisted-programming, generative-ai, kellan-elliott-mccrea, ai, llms

---

Notes from my Accessibility and Gen AI podcast appearance

I was a guest on the most recent episode of the Accessibility + Gen AI Podcast, hosted by Eamon McErlean and Joe Devon. We had a really fun, wide-ranging conversation about a host of different topics. I've extracted a few choice quotes from the transcript.

[... 947 words]

2:51 pm / accessibility, alt-text, podcasts, ai, generative-ai, llms, podcast-appearances

> After publishing this piece, I was contacted by Anthropic who told me that Sonnet 3.7 would not be considered a 10^26 FLOP model and cost a few tens of millions of dollars to train, though future models will be much bigger.
>
> — **Ethan Mollick**

# 5:56 pm / ethan-mollick, anthropic, claude, generative-ai, ai, llms

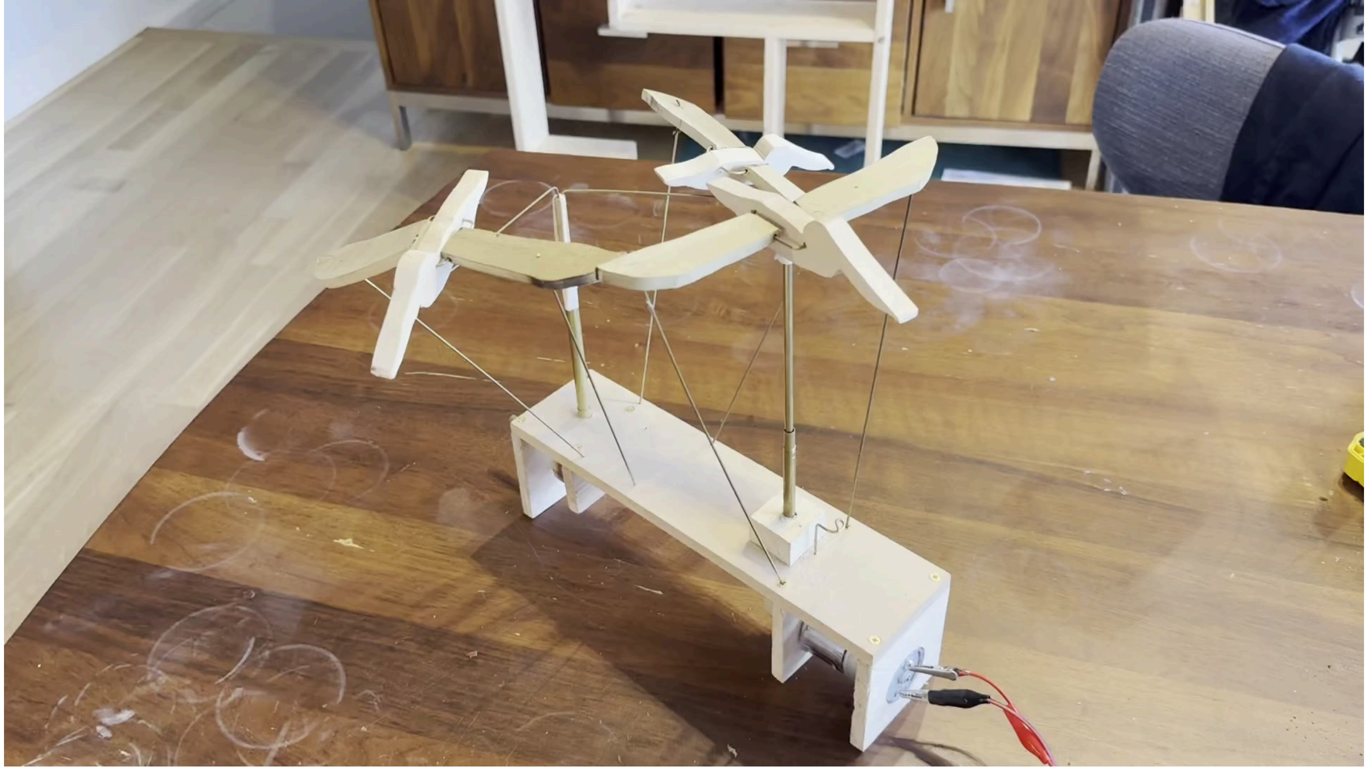
March 3, 2025

The features of Python's help() function (via) I've only ever used Python's `help()` feature by passing references to modules, classes functions and objects to it. Trey Hunner just taught me that it accepts strings too - `help("**")` tells you about the `**` operator, `help("if")` describes the `if` statement and `help("topics")` reveals even more options, including things like `help("SPECIALATTRIBUTES")` to learn about specific advanced topics.

# 7:15 pm / python, trey-hunner

March 4, 2025

I built an automaton called Squadron



I believe that the price you have to pay for taking on a project is writing about it afterwards. On that basis, I feel compelled to write up my decidedly non-software project from this weekend: Squadron, an automaton.

[... 1,142 words]

1:38 am / art, projects

---

llm-mistral 0.11. I added schema support to this plugin which adds support for the Mistral API to LLM. Release notes:

> - Support for LLM schemas. #19
> - `-o prefix '{'` option for forcing a response prefix. #18

Schemas now work with OpenAI, Anthropic, Gemini and Mistral hosted models, plus self-hosted models via Ollama and llm-ollama.

# 7:05 am / plugins, projects, ai, generative-ai, local-llms, llms, llm, mistral, ollama

---

llm-ollama 0.9.0. This release of the `llm-ollama` plugin adds support for schemas, thanks to a PR by Adam Compton.

Ollama provides very robust support for this pattern thanks to their structured outputs feature, which works across all of the models that they support by intercepting the logic that outputs the next token and restricting it to only tokens that would be valid in the context of the provided schema.

With Ollama and `llm-ollama` installed you can run even run structured schemas against vision prompts for local models. Here's one against Ollama's llama3.2-vision:

```
llm -m llama3.2-vision:latest \
  'describe images' \
  --schema 'species,description,count int' \
  -a https://static.simonwillison.net/static/2025/two-pelicans.jpg
```

I got back this:

```
{
    "species": "Pelicans",
    "description": "The image features a striking brown pelican with its distinctive orange beak, characterized by its large size and impressive wingspan.",
    "count": 1
}
```

(Actually a bit disappointing, as there are two pelicans and their beaks are brown.)

# 7:17 am / plugins, ai, generative-ai, llama, local-llms, llms, llm, vision-llms, ollama

---

A Practical Guide to Implementing DeepSearch / DeepResearch. I really like the definitions Han Xiao from Jina AI proposes for the terms DeepSearch and DeepResearch in this piece:

> **DeepSearch** runs through an iterative loop of searching, reading, and reasoning until it finds the optimal answer. [...]
>
> **DeepResearch** builds upon DeepSearch by adding a structured framework for generating long research reports.

I've recently found myself cooling a little on the classic RAG pattern of finding relevant documents and dumping them into the context for a single call to an LLM.

I think this definition of DeepSearch helps explain why. RAG is about answering questions that fall outside of the knowledge baked into a model. The DeepSearch pattern offers a tools-based alternative to classic RAG: we give the model extra tools for running multiple searches (which could be vector-based, or FTS, or even systems like ripgrep) and run it for several steps in a loop to try to find an answer.

I think DeepSearch is a lot more interesting than DeepResearch, which feels to me more like a presentation layer thing. Pulling together the results from multiple searches into a "report" looks more impressive, but I still worry that the report format provides a misleading impression of the quality of the "research" that took place.

# 5:25 pm / search, ai, generative-ai, llms, rag, llm-tool-use, jina, ai-assisted-search

---

March 5, 2025

QwQ-32B: Embracing the Power of Reinforcement Learning (via) New Apache 2 licensed reasoning model from Qwen:

> We are excited to introduce QwQ-32B, a model with 32 billion parameters that achieves performance comparable to DeepSeek-R1, which boasts 671 billion parameters (with 37 billion activated). This remarkable outcome underscores the effectiveness of RL when applied to robust foundation models pretrained on extensive world knowledge.

I had a lot of fun trying out their previous QwQ reasoning model last November. I demonstrated this new QwQ in my talk at NICAR about recent LLM developments. Here's the example I ran.

LM Studio just released GGUFs ranging in size from 17.2 to 34.8 GB. MLX have compatible weights published in 3bit, 4bit, 6bit and 8bit. Ollama has the new qwq too - it looks like they've renamed the previous November release qwq:32b-preview.

# 9:10 pm / open-source, ai, generative-ai, local-llms, llms, qwen, mlx, ollama, llm-reasoning, llm-release, lm-studio, ai-in-china

---

Career Update: Google DeepMind -> Anthropic. Nicholas Carlini (previously) on joining Anthropic, driven partly by his frustration at friction he encountered publishing his research at Google DeepMind after their merge with Google Brain. His area of expertise is adversarial machine learning.

> The recent advances in machine learning and language modeling are going to be transformative [d] But in order to realize this potential future in a way that doesn't put everyone's safety and security at risk, we're going to need to make

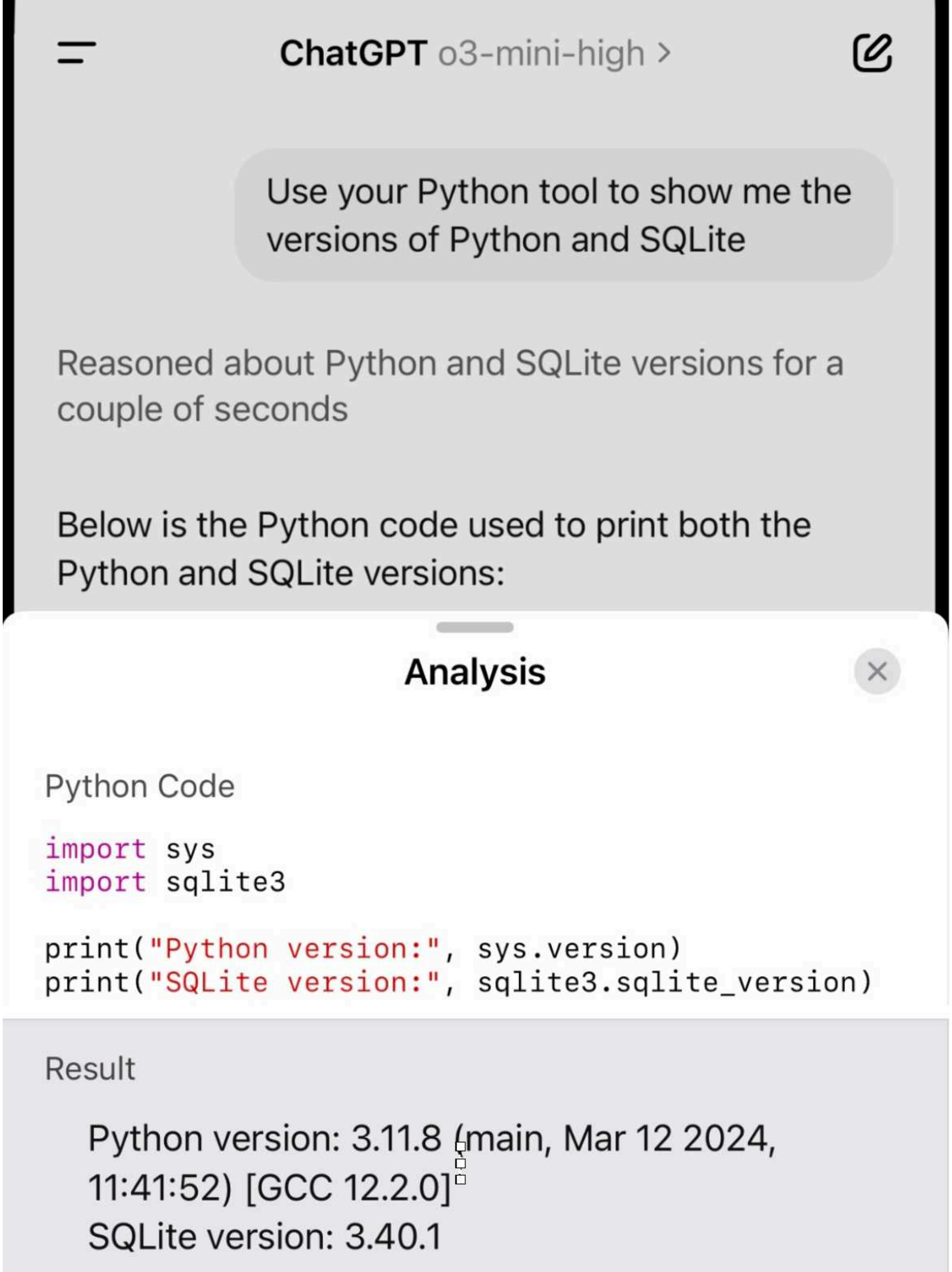
> a *lot* of progress---and soon. We need to make so much progress that no one organization will be able to figure everything out by themselves; we need to work together, we need to talk about what we're doing, and we need to start doing this now.

# 10:24 pm / google, machine-learning, ai, generative-ai, llms, anthropic, nicholas-carlini

---

Demo of ChatGPT Code Interpreter running in o3-mini-high. OpenAI made GPT-4.5 available to Plus ($20/month) users today. I was a little disappointed with GPT-4.5 when I tried it through the API, but having access in the ChatGPT interface meant I could use it with existing tools such as Code Interpreter which made its strengths a whole lot more evident - that’s a transcript where I had it design and test its own version of the JSON Schema succinct DSL I published last week.

Riley Goodside then spotted that Code Interpreter has been quietly enabled for other models too, including the excellent o3-mini reasoning model. This means you can have o3-mini reason about code, write that code, test it, iterate on it and keep going until it gets something that works.



Use your Python tool to show me the versions of Python and SQLite

Reasoned about Python and SQLite versions for a couple of seconds

Below is the Python code used to print both the Python and SQLite versions:

Python Code

```python
import sys
import sqlite3

print("Python version:", sys.version)
print("SQLite version:", sqlite3.sqlite_version)
```

Result

```
Python version: 3.11.8 (main, Mar 12 2024, 11:41:52) [GCC 12.2.0]
SQLite version: 3.40.1
```

Code Interpreter remains my favorite implementation of the "coding agent" pattern, despite recieving very few upgrades in the two years after its initial release. Plugging much stronger models into it than the previous GPT-4o default makes it even more useful.

Nothing about this in the ChatGPT release notes yet, but I've tested it in the ChatGPT iOS app and mobile web app and it definitely works there.

# 11:07 pm / python, ai, openai, generative-ai, chatgpt, riley-goodside, llms, ai-assisted-programming, code-interpreter, ai-agents, llm-reasoning, coding-agents

---

The Graphing Calculator Story (via) Utterly delightful story from Ron Avitzur in 2004 about the origins of the Graphing Calculator app that shipped with many versions of macOS. Ron's contract with Apple had ended but his badge kept working so he kept on letting himself in to work on the project. He even grew a small team:

> I asked my friend Greg Robbins to help me. His contract in another division at Apple had just ended, so he told his manager that he would start reporting to me. She didn't ask who I was and let him keep his office and badge. In turn, I told people that I was reporting to him. Since that left no managers in the loop, we had no meetings and could be extremely productive

# 11:36 pm / apple, computer-history

March 6, 2025

Aider: Using uv as an installer. Paul Gauthier has an innovative solution for the challenge of helping end users get a copy of his Aider CLI Python utility installed in an isolated virtual environment without first needing to teach them what an "isolated virtual environment" is.

Provided you already have a Python install of version 3.8 or higher you can run this:

```
pip install aider-install && aider-install
```

The aider-install package itself depends on uv. When you run `aider-install` it executes the following Python code:

```python
def install_aider():
    try:
        uv_bin = uv.find_uv_bin()
        subprocess.check_call([
            uv_bin, "tool", "install", "--force", "--python", "python3.12", "aider-chat@latest"
        ])
        subprocess.check_call([uv_bin, "tool", "update-shell"])
    except subprocess.CalledProcessError as e:
        print(f"Failed to install aider: {e}")
        sys.exit(1)
```

This first figures out the location of the `uv` Rust binary, then uses it to install his aider-chat package by running the equivalent of this command:

```
uv tool install --force --python python3.12 aider-chat@latest
```

This will in turn install a brand new standalone copy of Python 3.12 and tuck it away in uv's own managed directory structure where it shouldn't hurt anything else.

The `aider-chat` script defaults to being dropped in the XDG standard directory, which is probably `~/.local/bin` - see uv's documentation. The --force flag ensures that `uv` will overwrite any previous attempts at installing `aider-chat` in that location with the new one.

Finally, running `uv tool update-shell` ensures that bin directory is on the user's PATH.

I *think* I like this. There is a LOT of stuff going on here, and experienced users may well opt for an alternative installation mechanism.

But for non-expert Python users who just want to start using Aider, I think this pattern represents quite a tasteful way of getting everything working with minimal risk of breaking the user's system.

Update: Paul adds:

> Offering this install method dramatically reduced the number of GitHub issues from users with conflicted/broken python environments.
>
> I also really like the "curl | sh" aider installer based on uv. Even users who don't have python installed can use it.

# 1:47 am / cli, python, aider, uv, paul-gauthier

Will the future of software development run on vibes? I got a few quotes in this piece by Benj Edwards about **vibe coding**, the term Andrej Karpathy coined for when you prompt an LLM to write code, accept all changes and keep feeding it prompts and error messages and see what you can get it to build.

Here's what I originally sent to Benj:

> I really enjoy vibe coding - it's a fun way to play with the limits of these models. It's also useful for prototyping, where the aim of the exercise is to try out an idea and prove if it can work.
>
> Where vibe coding fails is in producing maintainable code for production settings. I firmly believe that as a developer you have to take accountability for the code you produce - if you're going to put your name to it you need to be confident that you understand how and why it works - ideally to the point that you can explain it to somebody else.
>
> Vibe coding your way to a production codebase is clearly a terrible idea. Most of the work we do as software engineers is about evolving existing systems, and for those the quality and understandability of the underlying code is crucial.
>
> For experiments and low-stake projects where you want to explore what's possible and build fun prototypes? Go wild! But stay aware of the very real risk that a good enough prototype often faces pressure to get pushed to production.
>
> If an LLM wrote every line of your code but you've reviewed, tested and understood it all, that's not vibe coding in my book - that's using an LLM as a typing assistant.

# 3:39 am / ai, andrej-karpathy, generative-ai, llms, ai-assisted-programming, benj-edwards, vibe-coding

---

monolith (via) Neat CLI tool built in Rust that can create a single packaged HTML file of a web page plus all of its dependencies.

```
cargo install monolith # or brew install
monolith https://simonwillison.net/ > simonwillison.html
```

That command produced this 1.5MB single file result. All of the linked images, CSS and JavaScript assets have had their contents inlined into base64 URIs in their `src=` and `href=` attributes.

I was intrigued as to how it works, so I dumped the whole repository into Gemini 2.0 Pro and asked for an architectural summary:

```
cd /tmp
git clone https://github.com/Y2Z/monolith
cd monolith
files-to-prompt . -c | llm -m gemini-2.0-pro-exp-02-05 \
  -s 'architectural overview as markdown'
```

Here's what I got. Short version: it uses the `reqwest`, `html5ever`, `markup5ever_rcdom` and `cssparser` crates to fetch and parse HTML and CSS and extract, combine and rewrite the assets. It doesn't currently attempt to run any JavaScript.

# 3:37 pm / cli, scraping, ai, rust, generative-ai, llms, ai-assisted-programming, files-to-prompt

---

March 7, 2025

Mistral OCR (via) New closed-source specialist OCR model by Mistral - you can feed it images or a PDF and it produces Markdown with optional embedded images.

It's available via their API, or it's "available to self-host on a selective basis" for people with stringent privacy requirements who are willing to talk to their sales team.

I decided to try out their API, so I copied and pasted example code from their notebook into my custom Claude project and told it:

> `Turn this into a CLI app, depends on mistralai - it should take a file path and an optional API key defauling to environment called MISTRAL_API_KEY`

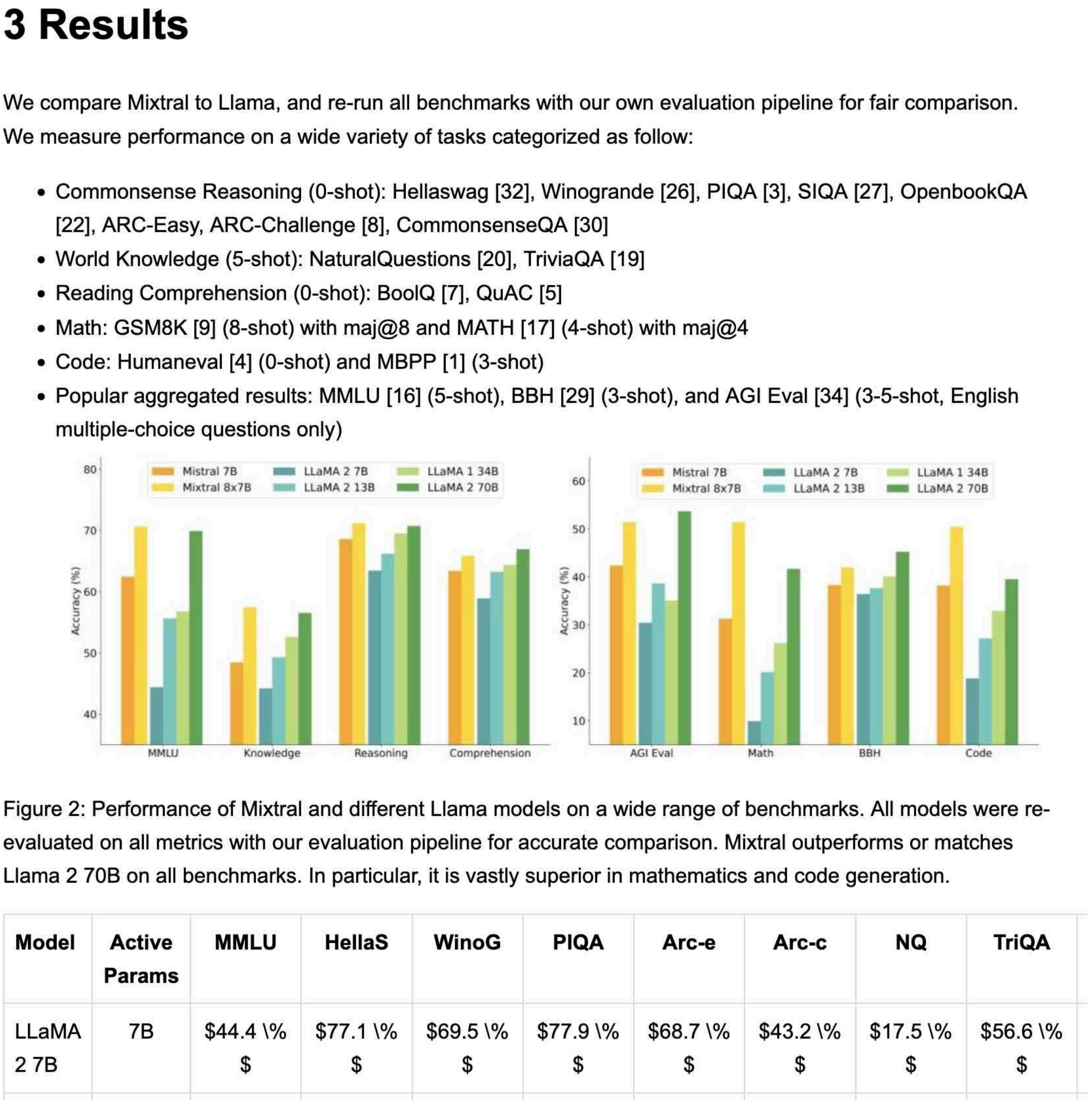
After some further iteration / vibe coding I got to something that worked, which I then tidied up and shared as mistral_ocr.py.

You can try it out like this:

```
export MISTRAL_API_KEY='...'
uv run http://tools.simonwillison.net/python/mistral_ocr.py \
  mixtral.pdf --html --inline-images > mixtral.html
```

I fed in the Mixtral paper as a PDF. The API returns Markdown, but my `--html` option renders that Markdown as HTML and the `--inline-images` option takes any images and inlines them as base64 URIs (inspired by monolith). The result is mixtral.html, a 972KB HTML file with images and text bundled together.

This did a pretty great job!



3 Results

We compare Mixtral to Llama, and re-run all benchmarks with our own evaluation pipeline for fair comparison. We measure performance on a wide variety of tasks categorized as follow:

- Commonsense Reasoning (0-shot): Hellaswag [32], Winogrande [26], PIQA [3], SIQA [27], OpenbookQA [22], ARC-Easy, ARC-Challenge [8], CommonsenseQA [30]
- World Knowledge (5-shot): NaturalQuestions [20], TriviaQA [19]
- Reading Comprehension (0-shot): BoolQ [7], QuAC [5]
- Math: GSM8K [9] (8-shot) with maj@8 and MATH [17] (4-shot) with maj@4
- Code: Humaneval [4] (0-shot) and MBPP [1] (3-shot)
- Popular aggregated results: MMLU [16] (5-shot), BBH [29] (3-shot), and AGI Eval [34] (3-5-shot, English multiple-choice questions only)

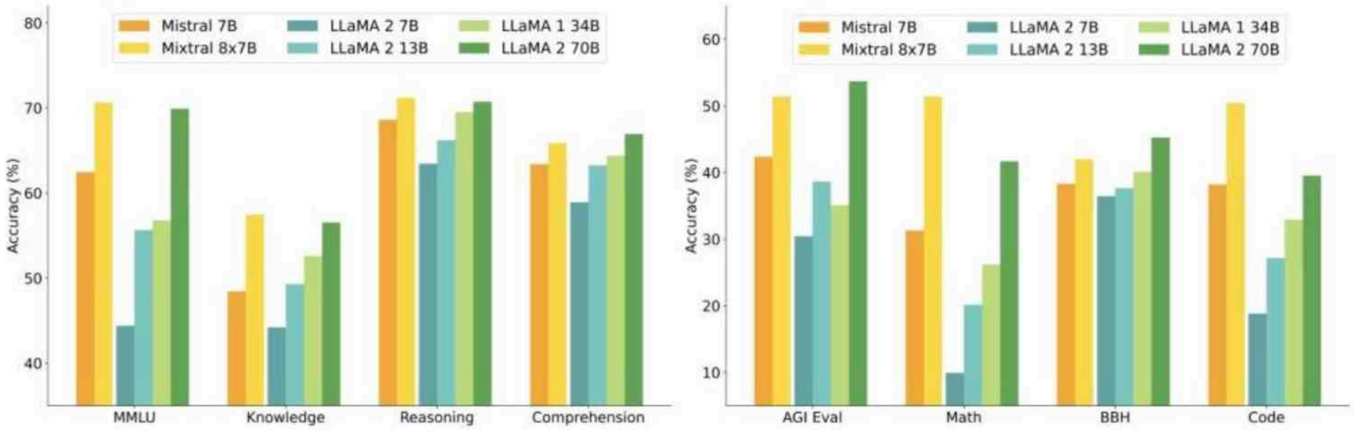


Figure 2: Performance of Mixtral and different Llama models on a wide range of benchmarks. All models were re-evaluated on all metrics with our evaluation pipeline for accurate comparison. Mixtral outperforms or matches Llama 2 70B on all benchmarks. In particular, it is vastly superior in mathematics and code generation.

Model	Active Params	MMLU	HellaS	WinoG	PIQA	Arc-e	Arc-c	NQ	TriQA
LLaMA 2 7B	7B	$44.4 \% $	$77.1 \% $	$69.5 \% $	$77.9 \% $	$68.7 \% $	$43.2 \% $	$17.5 \% $	$56.6 \% $

My script renders Markdown tables but I haven't figured out how to render inline Markdown MathML yet. I ran the command a second time and requested Markdown output (the default) like this:

```
uv run http://tools.simonwillison.net/python/mistral_ocr.py \
  mixtral.pdf > mixtral.md
```

Here's that Markdown rendered as a Gist - there are a few MathML glitches so clearly the Mistral OCR MathML dialect and the GitHub Formatted Markdown dialect don't quite line up.

My tool can also output raw JSON as an alternative to Markdown or HTML - full details in the documentation.

The Mistral API is priced at roughly 1000 pages per dollar, with a 50% discount for batch usage.

The big question with LLM-based OCR is always how well it copes with accidental instructions in the text (can you safely OCR a document full of prompting examples?) and how well it handles text it can't write.

Mistral's Sophia Yang says it "should be robust" against following instructions in the text, and invited people to try and find counter-examples.

Alexander Doria noted that Mistral OCR can hallucinate text when faced with handwriting that it cannot understand.

# 1:39 am / cli, ocr, pdf, projects, ai, generative-ai, llms, claude, mistral, vision-llms, uv

---

State-of-the-art text embedding via the Gemini API (via) Gemini just released their new text embedding model, with the snappy name `gemini-embedding-exp-03-07`. It supports 8,000 input tokens - up from 3,000 - and outputs vectors that are a lot larger than their previous `text-embedding-004` model - that one output size 768 vectors, the new model outputs 3072.

Storing that many floating point numbers for each embedded record can use a lot of space. thankfully, the new model supports Matryoshka Representation Learning - this means you can simply truncate the vectors to trade accuracy for storage.

I added support for the new model in llm-gemini 0.14. LLM doesn't yet have direct support for Matryoshka truncation so I instead registered different truncated sizes of the model under different IDs: `gemini-embedding-exp-03-07-2048`, `gemini-embedding-exp-03-07-1024`, `gemini-embedding-exp-03-07-512`, `gemini-embedding-exp-03-07-256`, `gemini-embedding-exp-03-07-128`.

The model is currently free while it is in preview, but comes with a strict rate limit - 5 requests per minute and just 100 requests a day. I quickly tripped those limits while testing out the new model - I hope they can bump those up soon.

# 11:19 pm / google, ai, embeddings, llm, gemini

March 8, 2025

Apple Is Delaying the 'More Personalized Siri' Apple Intelligence Features. Apple told John Gruber (and other Apple press) this about the new "personalized" Siri:

> It's going to take us longer than we thought to deliver on these features and we anticipate rolling them out in the coming year.

I have a hunch that this delay might relate to security.

These new Apple Intelligence features involve Siri responding to requests to access information in applications and then performing actions on the user's behalf.

This is the worst possible combination for prompt injection attacks! Any time an LLM-based system has access to private data, tools it can call, and exposure to potentially malicious instructions (like emails and text messages from untrusted strangers) there's a significant risk that an attacker might subvert those tools and use them to damage or exfiltrating a user's data.

I published this piece about the risk of prompt injection to personal digital assistants back in November 2023, and nothing has changed since then to make me think this is any less of an open problem.

# 5:39 am / apple, john-gruber, security, ai, prompt-injection, generative-ai, llms, apple-intelligence

---

Politico: 5 Questions for Jack Clark (via) I tend to ignore statements with this much future-facing hype, especially when they come from AI labs who are both raising money and trying to influence US technical policy.

Anthropic's Jack Clark has an excellent long-running newsletter which causes me to take him more seriously than many other sources.

Jack says:

> In 2025 myself and @AnthropicAI will be more forthright about our views on AI, especially the speed with which powerful things are arriving.

In response to Politico's question "What's one underrated big idea?" Jack replied:

> People underrate how significant and fast-moving AI progress is. We have this notion that in late 2026, or early 2027, powerful AI systems will be built that will have intellectual capabilities that match or exceed Nobel Prize winners. They'll have the ability to navigate all of the interfaces… they will have the ability to autonomously reason over kind of complex tasks for extended periods. They'll also have the ability to interface with the physical world by operating drones or robots. Massive, powerful things are beginning to come into view, and we're all underrating how significant that will be.

# 5:13 pm / ai, jack-clark, anthropic

Cutting-edge web scraping techniques at NICAR. Here's the handout for a workshop I presented this morning at NICAR 2025 on web scraping, focusing on lesser know tips and tricks that became possible only with recent developments in LLMs.

For workshops like this I like to work off an extremely detailed handout, so that people can move at their own pace or catch up later if they didn't get everything done.

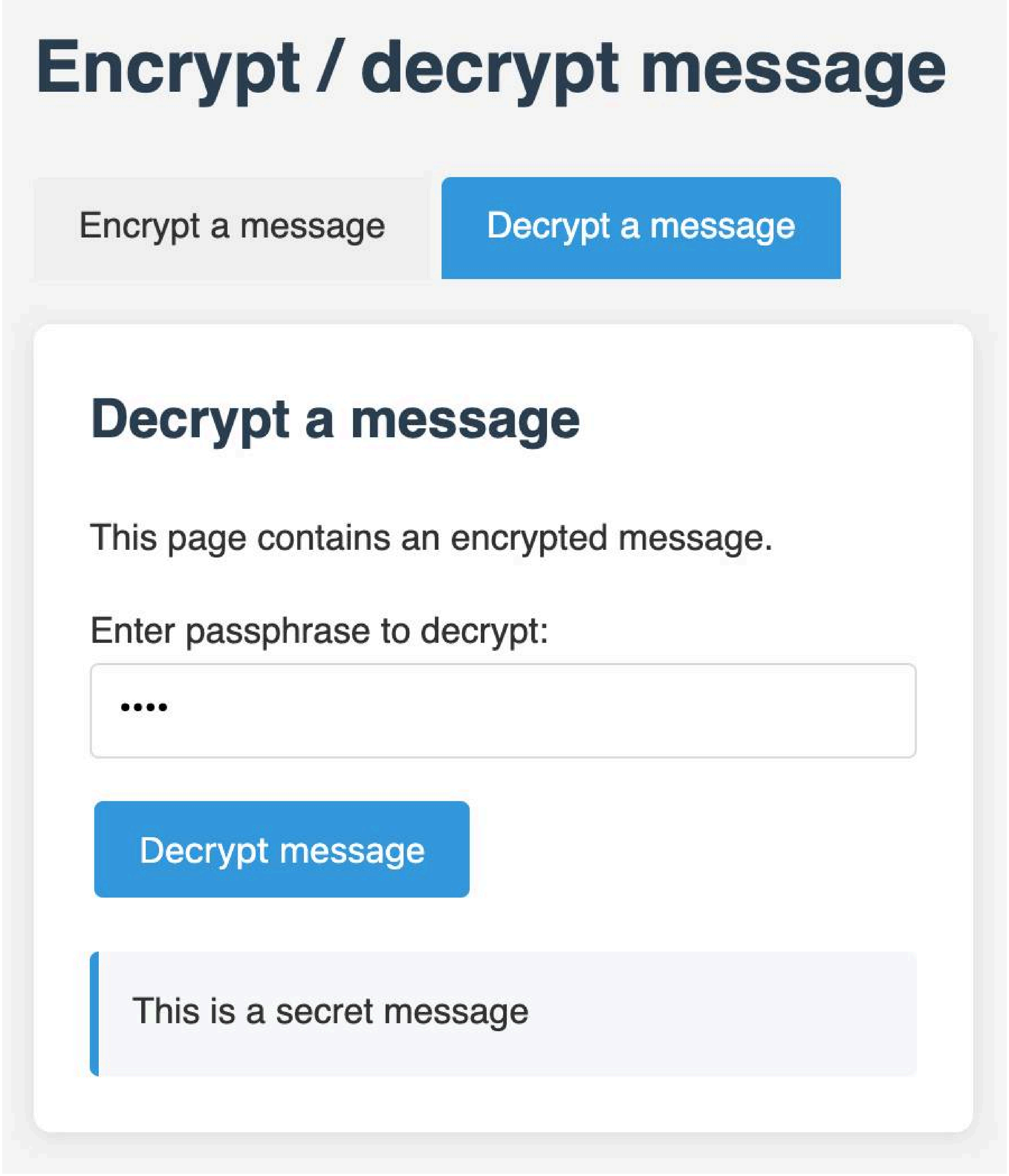
The workshop consisted of four parts:

> 1. Building a Git scraper - an automated scraper in GitHub Actions that records changes to a resource over time
> 2. Using in-browser JavaScript and then shot-scraper to extract useful information
> 3. Using LLM with both OpenAI and Google Gemini to extract structured data from unstructured websites
> 4. Video scraping using Google AI Studio

I released several new tools in preparation for this workshop (I call this "NICAR Driven Development"):

- git-scraper-template template repository for quickly setting up new Git scrapers, which I wrote about here
- LLM schemas, finally adding structured schema support to my LLM tool
- shot-scraper har for archiving pages as HTML Archive files - though I cut this from the workshop for time

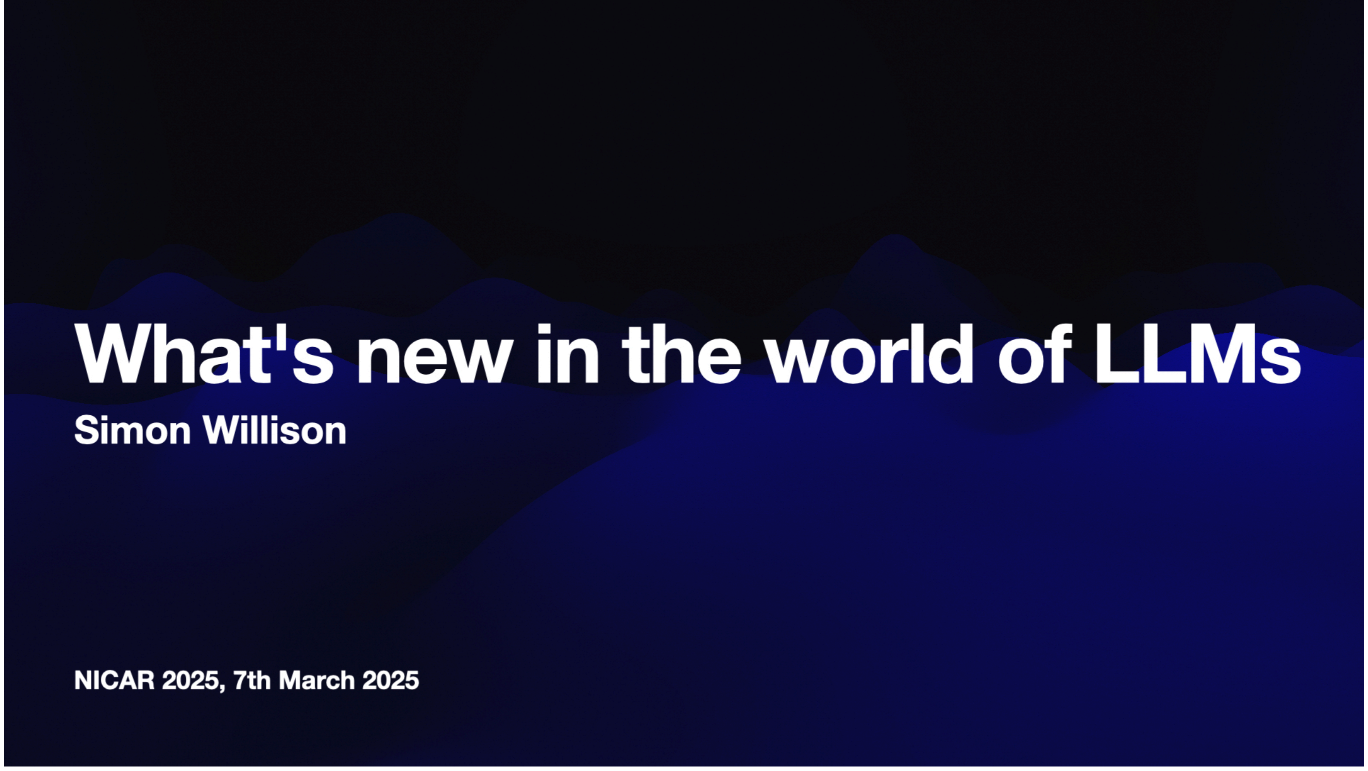
I also came up with a fun way to distribute API keys for workshop participants: I had Claude build me a web page where I can create an encrypted message with a passphrase, then share a URL to that page with users and give them the passphrase to unlock the encrypted message. You can try that at tools.simonwillison.net/encrypt - or use this link and enter the passphrase "demo":



# 7:25 pm / scraping, speaking, ai, git-scraping, shot-scraper, openai, generative-ai, llms, ai-assisted-programming, claude, gemini, nicar, claude-artifacts

---

What’s new in the world of LLMs, for NICAR 2025



What's new in the world of LLMs

Simon Willison

NICAR 2025, 7th March 2025

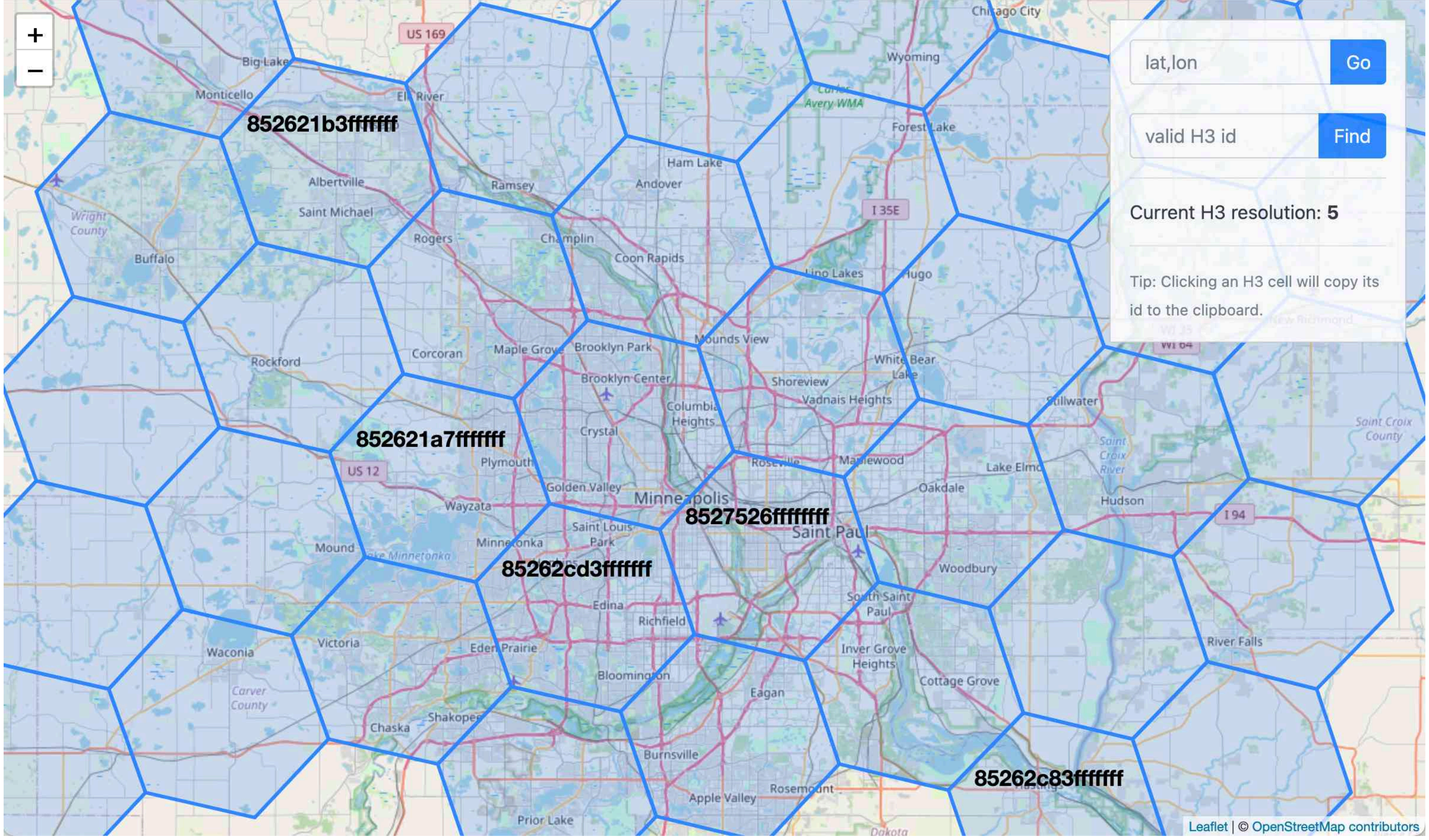
I presented two sessions at the NICAR 2025 data journalism conference this year. The first was this one based on my review of LLMs in 2024, extended by several months to cover everything that's happened in 2025 so far. The second was a workshop on Cutting-edge web scraping techniques, which I've written up separately.

[... 2,797 words]

11:19 pm / data-journalism, speaking, my-talks, ai, generative-ai, local-llms, llms, annotated-talks, gemini, nicar, vision-llms, chatbot-arena

March 9, 2025

wolf-h3-viewer.glitch.me (via) Neat interactive visualization of Uber's H3 hexagonal geographical indexing mechanism.



Here's the source code.

Why does H3 use hexagons? Because Hexagons are the Bestagons:

> When hexagons come together, they form three-sided joints 120 degrees apart. This, for the least material, is the most mechanically stable arrangement.

Only triangles, squares, and hexagons can tile a plane without gaps, and of those three shapes hexagons offer the best ratio of perimeter to area.

# 2:51 pm / geospatial, javascript

> I've been using Claude Code for a couple of days, and it has been absolutely ruthless in chewing through legacy bugs in my gnarly old code base. It's like a wood chipper fueled by dollars. It can power through shockingly impressive tasks, using nothing but chat. [...]
>
> Claude Code's form factor is clunky as hell, it has no multimodal support, and it's hard to juggle with other tools. But it doesn't matter. It might look antiquated but it makes Cursor, Windsurf, Augment and the rest of the lot (yeah, ours too, and Copilot, let's be honest) FEEL antiquated.

— **Steve Yegge,** who works on Cody at Sourcegraph

# 3:30 pm / steve-yegge, anthropic, claude, ai-assisted-programming, generative-ai, ai, llms, claude-code, coding-agents

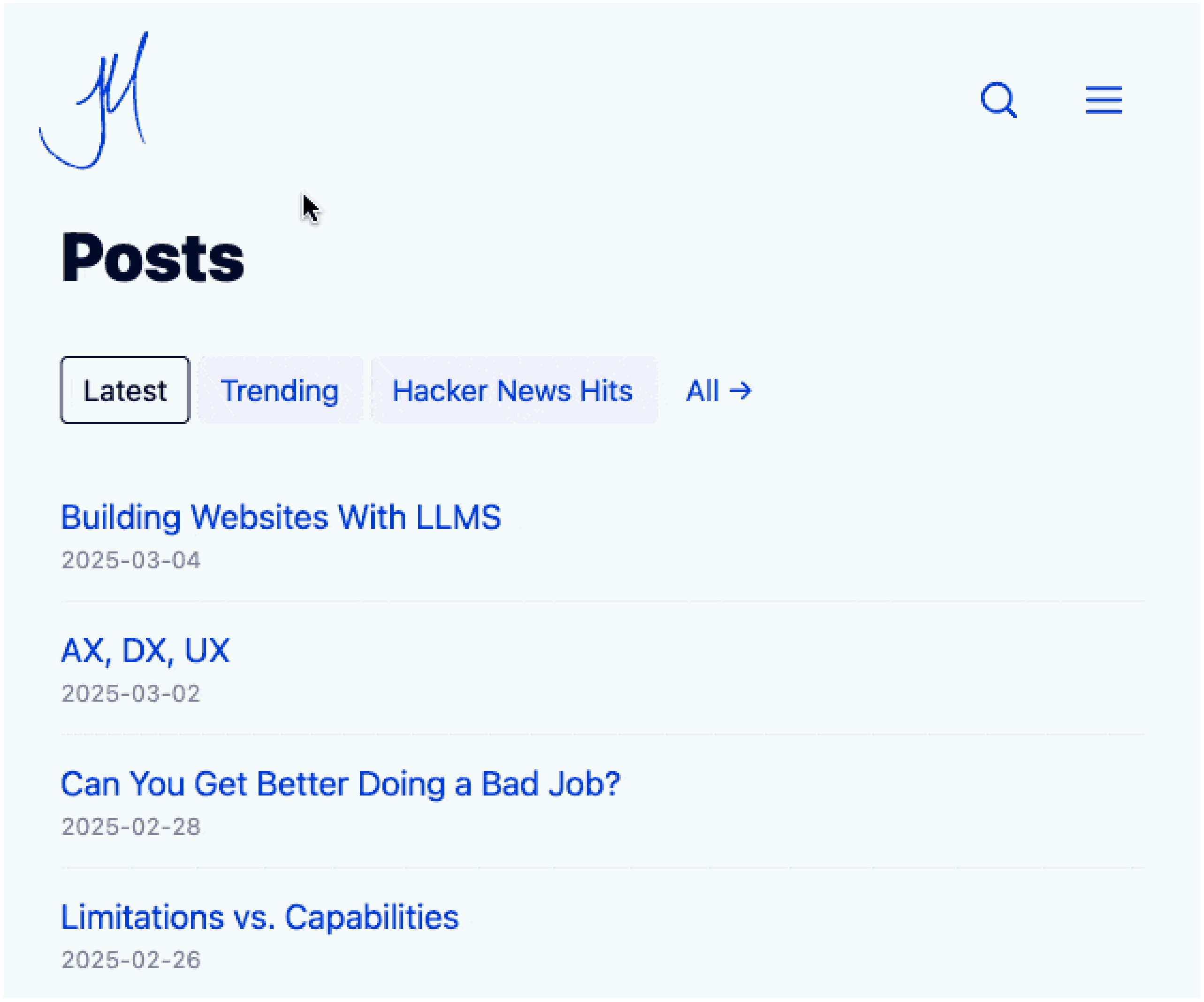
---

March 10, 2025

Building Websites With Lots of Little HTML Pages (via) Jim Nielsen coins a confusing new acronym - LLMS for (L)ots of (L)ittle ht(M)l page(S). He's using this to describe his latest site refresh which makes extensive use of cross-document view transitions - a fabulous new progressive enhancement CSS technique that's supported in Chrome and Safari (and hopefully soon in Firefox).

> With cross-document view transitions getting broader and broader support, I'm realizing that building in-page, progressively-enhanced interactions is more work than simply building two HTML pages and linking them.

Jim now has small static pages powering his home page filtering interface and even his navigation menu, with CSS view transitions configured to smoothly animate between the pages. I think it feels really good - here's what it looked like for me in Chrome (it looked the same both with and without JavaScript disabled):



Posts

Building Websites With LLMS
2025-03-04

AX, DX, UX
2025-03-02

Can You Get Better Doing a Bad Job?
2025-02-28

Limitations vs. Capabilities
2025-02-26

Watching the network panel in my browser, most of these pages are 17-20KB gzipped (~45KB after they've decompressed). No wonder it feels so snappy.

I poked around in Jim's CSS and found this relevant code:

```
@view-transition {
  navigation: auto;
}

.posts-nav a[aria-current="page"]:not(:last-child):after {
  border-color: var(--c-text);
  view-transition-name: posts-nav;
}

/* Old stuff going out */
::view-transition-old(posts-nav) {
  animation: fade 0.2s linear forwards;
  /* https://jakearchibald.com/2024/view-transitions-handling-aspect-ratio-changes/ */
  height: 100%;
}

/* New stuff coming in */
::view-transition-new(posts-nav) {
  animation: fade 0.3s linear reverse;
  height: 100%;
}

@keyframes fade {
  from {
    opacity: 1;
  }
  to {
    opacity: 0;
  }
}
```

Jim observes:

> This really feels like a game-changer for simple sites. If you can keep your site simple, it's easier to build traditional, JavaScript-powered on-page interactions as small, linked HTML pages.

I've experimented with view transitions for Datasette in the past and the results were very promising. Maybe I'll pick that up again.

Bonus: Jim has a clever JavaScript trick to avoid clicks to the navigation menu being added to the browser's history in the default case.

# 12:38 am / css, progressive-enhancement, view-transitions

> It seems to me that "vibe checks" for how smart a model feels are easily gameable by making it have a better personality.
>
> My guess is that it's most of the reason Sonnet 3.5.1 was so beloved. Its personality was made much more *appealing*, compared to e. g. OpenAI's corporate drones. [...]
>
> Deep Research was this for me, at first. Some of its summaries were just *pleasant* to read, they felt so information-dense and intelligent! Not like typical AI slop at all! But then it turned out most of it was just AI slop underneath anyway, and now my slop-recognition function has adjusted and the effect is gone.

— **Thane Ruthenis,** A Bear Case: My Predictions Regarding AI Progress

# 1:50 am / llms, ai, generative-ai, slop, deep-research, ai-personality

---

llm-openrouter 0.4. I found out this morning that OpenRouter include support for a number of (rate-limited) free API models.

I occasionally run workshops on top of LLMs (like this one) and being able to provide students with a quick way to obtain an API key against models where they don't have to setup billing is really valuable to me!

This inspired me to upgrade my existing llm-openrouter plugin, and in doing so I closed out a bunch of open feature requests.

Consider this post the annotated release notes:

> - LLM schema support for OpenRouter models that support structured output. #23

I'm trying to get support for LLM's new schema feature into as many plugins as possible.

OpenRouter's OpenAI-compatible API includes support for the `response_format` structured content option, but with an important caveat: it only works for some models, and if you try to use it on others it is silently ignored.

I filed an issue with OpenRouter requesting they include schema support in their machine-readable model index. For the moment LLM will let you specify schemas for unsupported models and will ignore them entirely, which isn't ideal.

> - `llm openrouter key` command displays information about your current API key. #24

Useful for debugging and checking the details of your key's rate limit.

> - `llm -m ... -o online 1` enables web search grounding against any model, powered by Exa. #25

OpenRouter apparently make this feature available to every one of their supported models! They're using new-to-me Exa to power this feature, an AI-focused search engine startup who appear to have built their own index with their own crawlers (according to their FAQ). This feature is currently priced by OpenRouter at $4 per 1000 results, and since 5 results are returned for every prompt that's 2 cents per prompt.

> - `llm openrouter models` command for listing details of the OpenRouter models, including a `--json` option to get JSON and a `--free` option to filter for just the free models. #26

This offers a neat way to list the available models. There are examples of the output in the comments on the issue.

> - New option to specify custom provider routing: `-o provider '{JSON here}'`. #17

Part of OpenRouter's USP is that it can route prompts to different providers depending on factors like latency, cost or as a fallback if your first choice is unavailable - great for if you are using open weight models like Llama which are hosted by competing companies.

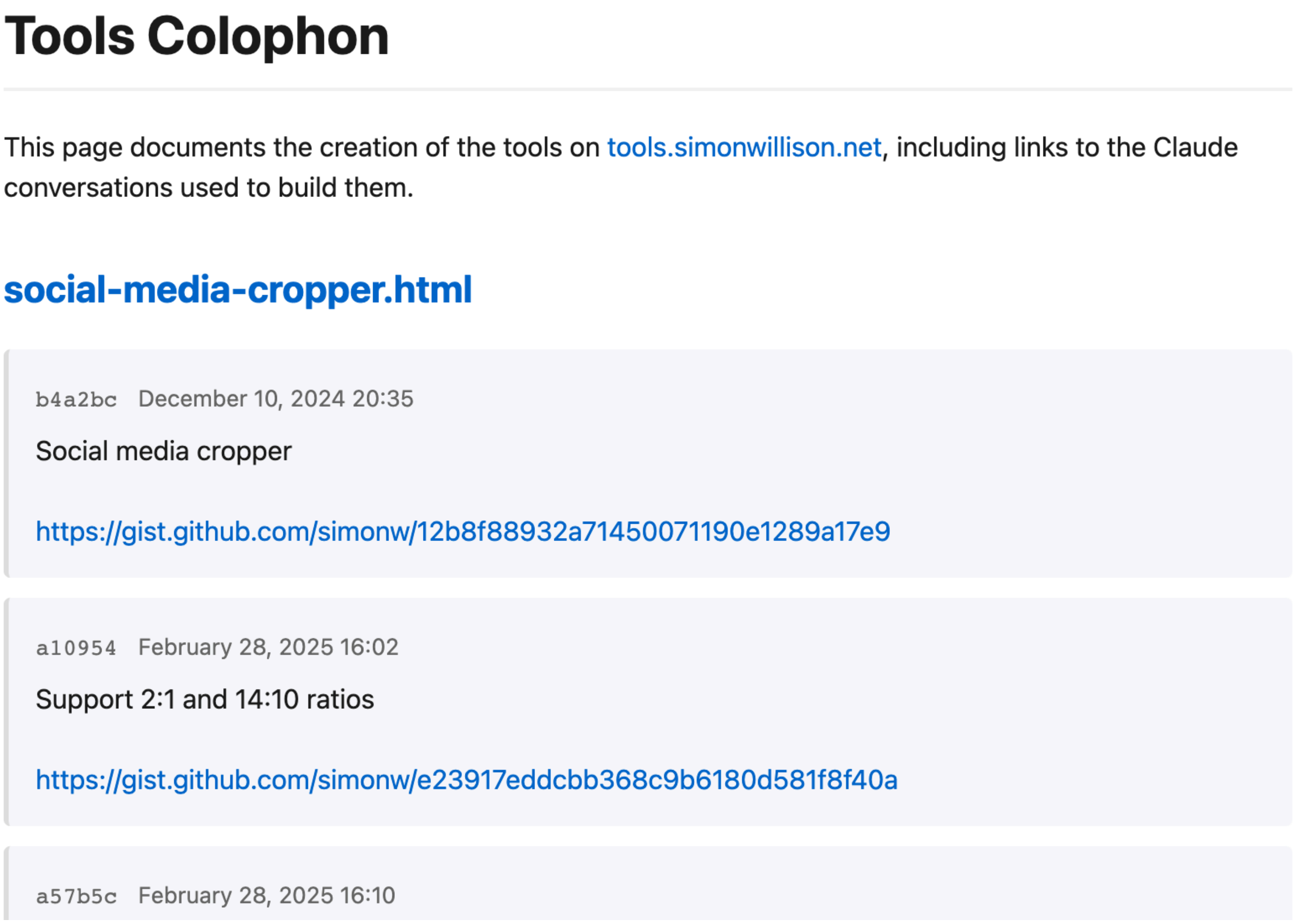
The options they provide for routing are very thorough - I had initially hoped to provide a set of CLI options that covered all of these bases, but I decided instead to reuse their JSON format and forward those options directly on to the model.

# 9:40 pm / cli, plugins, projects, ai, annotated-release-notes, generative-ai, llms, llm, openrouter, ai-assisted-search

---

March 11, 2025

Here’s how I use LLMs to help me write code


Tools Colophon

This page documents the creation of the tools on tools.simonwillison.net, including links to the Claude conversations used to build them.

social-media-cropper.html

b4a2bc December 10, 2024 20:35

Social media cropper

https://gist.github.com/simonw/12b8f88932a71450071190e1289a17e9

a10954 February 28, 2025 16:02

Support 2:1 and 14:10 ratios

https://gist.github.com/simonw/e23917eddcbb368c9b6180d581f8f40a

a57b5c February 28, 2025 16:10

Online discussions about using Large Language Models to help write code inevitably produce comments from developers who's experiences have been disappointing. They often ask what they're doing wrong—how come some people are reporting such great results when their own experiments have proved lacking?

[... 5,178 words]

2:09 pm / tools, ai, github-actions, openai, generative-ai, llms, ai-assisted-programming, anthropic, claude, gemini, claude-artifacts, vibe-coding, files-to-prompt, coding-agents, claude-code

page 1 / 3 next » last »»

2025 » March

M	T	W	T	F	S	S
					1	**2**
3	**4**	5	6	7	**8**	9
10	**11**	**12**	**13**	14	15	16
17	18	**19**	**20**	21	22	23
24	**25**	26	27	28	29	30
31						