December 2024

Search posts from December 2024 Search

100 posts: 12 entries, 62 links, 26 quotes

Dec. 1, 2024

> Most people don't have an intuition about what current hardware can and can't do. There is a simple math that can help you with that: "you can process about 500MB in one second on a single machine". I know it's not a universal truth and there are a lot of details that can change that but believe me, this estimation is a pretty good tool to have under your belt.
>
> — **Javi Santana**

5:02 am / big-data, scaling

Turning Your Root URL Into a DuckDB Remote Database. Fun idea from Drew Breunig: DuckDB supports attaching existing databases that are accessible over HTTP using their URL. Drew suggests creating vanity URLs using your root domain, detecting the DuckDB user-agent and serving the database file directly - allowing tricks like this one:

```
ATTACH 'https://steplist.app/' AS steplist;
SELECT * FROM steplist.lists;
```

10:02 pm / duckdb, drew-breunig

LLM 0.19. I just released version 0.19 of LLM, my Python library and CLI utility for working with Large Language Models.

I released 0.18 a couple of weeks ago adding support for calling models from Python `asyncio` code. 0.19 improves on that, and also adds a new mechanism for models to report their token usage.

LLM can log those usage numbers to a SQLite database, or make then available to custom Python code.

My eventual goal with these features is to implement token accounting as a Datasette plugin so I can offer AI features in my SaaS platform without worrying about customers spending unlimited LLM tokens.

Those 0.19 release notes in full:

> - Tokens used by a response are now logged to new `input_tokens` and `output_tokens` integer columns and a `token_details` JSON string column, for the default OpenAI models and models from other plugins that implement this feature. #610
> - `llm prompt` now takes a `-u/--usage` flag to display token usage at the end of the response.
> - `llm logs -u/--usage` shows token usage information for logged responses.
> - `llm prompt ... --async` responses are now logged to the database. #641
> - `llm.get_models()` and `llm.get_async_models()` functions, documented here. #640
> - `response.usage()` and async response `await response.usage()` methods, returning a `Usage(input=2, output=1, details=None)` dataclass. #644
> - `response.on_done(callback)` and `await response.on_done(callback)` methods for specifying a callback to be executed when a response has completed, documented here. #653
> - Fix for bug running `llm chat` on Windows 11. Thanks, Sukhbinder Singh. #495

I also released three new plugin versions that add support for the new usage tracking feature: llm-gemini 0.5, llm-claude-3 0.10 and llm-mistral 0.9.

11:59 pm / cli, projects, releasenotes, releases, ai, generative-ai, llms, llm

---

Dec. 2, 2024

Simon Willison: The Future of Open Source and AI (via) I sat down a few weeks ago to record this conversation with Logan Kilpatrick and Nolan Fortman for their podcast Around the Prompt. The episode is available on YouTube and Apple Podcasts and other platforms.

We talked about a whole bunch of different topics, including the ongoing debate around the term "open source" when applied to LLMs and my thoughts on why I don't feel threatened by LLMs as a software engineer (at 40m05s).

1:03 am / open-source, podcasts, youtube, ai, generative-ai, llms, logan-kilpatrick, podcast-appearances

---

> For most software engineers, being well rounded is more important than pure technical mastery. This was already true, of course — see @patio11's famous advice "Don't call yourself a programmer" — but even more so due to foundation models. In most situations, skills like being able to use AI to rapidly prototype in order to communicate with clients to iterate on specifications create far more business value than technical wizardry alone.
>
> — **Arvind Narayanan**

11:51 am / ai, programming, software-engineering, arvind-narayanan

---

PydanticAI (via) New project from Pydantic, which they describe as an "Agent Framework / shim to use Pydantic with LLMs".

I asked which agent definition they are using and it's the "system prompt with bundled tools" one. To their credit, they explain that in their documentation:

> The Agent has full API documentation, but conceptually you can think of an agent as a container for:
>
> - A system prompt — a set of instructions for the LLM written by the developer
> - One or more retrieval tool — functions that the LLM may call to get information while generating a response
> - An optional structured result type — the structured datatype the LLM must return at the end of a run

Given how many other existing tools already lean on Pydantic to help define JSON schemas for talking to LLMs this is an interesting complementary direction for Pydantic to take.

There's some overlap here with my own LLM project, which I still hope to add a function calling / tools abstraction to in the future.

9:08 pm / python, generative-ai, llms, llm, llm-tool-use, ai-agents, pydantic, agent-definitions

---

NYTimes reporters getting verified profiles on Bluesky. NYT data journalist Dylan Freedman has kicked off an initiative to get NYT accounts and reporters on Bluesky verified via vanity `nytimes.com` handles - Dylan is now @dylanfreedman.nytimes.com.

They're using Bluesky's support for TXT domain records. If you use Google's Dig tool to look at the TXT record for `_atproto.dylanfreedman.nytimes.com` you'll see this:

```
_atproto.dylanfreedman.nytimes.com. 500 IN TXT "did=did:plc:zeqq4z7aybrqg6go6vx6lzwt"
```

9:24 pm / new-york-times, social-media, bluesky

---

datasette-llm-usage. I released the first alpha of a Datasette plugin to help track LLM usage by other plugins, with the goal of supporting token allowances - both for things like free public apps that stop working after a daily allowance, plus free previews of AI features for paid-account-based projects such as Datasette Cloud.

It's using the usage features I added in LLM 0.19.

The alpha doesn't do much yet - it will start getting interesting once I upgrade other plugins to depend on it.

Design notes so far in issue #1.

9:33 pm / plugins, projects, releases, ai, datasette, datasette-cloud, generative-ai, llms, llm

---

Dec. 3, 2024

Certain names make ChatGPT grind to a halt, and we know why (via) Benj Edwards on the really weird behavior where ChatGPT stops output with an error rather than producing the names David Mayer, Brian Hood, Jonathan Turley, Jonathan Zittrain, David Faber or Guido Scorza.

The OpenAI API is entirely unaffected - this problem affects the consumer ChatGPT apps only.

It turns out many of those names are examples of individuals who have complained about being defamed by ChatGPT in the last. Brian Hood is the Australian mayor who was a victim of lurid ChatGPT hallucinations back in March 2023, and settled with OpenAI out of court.

2:31 am / ethics, ai, openai, generative-ai, chatgpt, llms, benj-edwards, ai-ethics, hallucinations

---

Finally, in most workplaces, incentive structures don’t exist for people to (a) reduce their workloads to such an extent that their role becomes vulnerable or (b) voluntarily accept more responsibility without also

taking on more pay.

These things are all natural rate limiters on technology adoption and the precise mix they show up in varies from workplace to workplace as every team has its own culture and ways of working. And regardless of what your friendly neighbourhood management consulting firm will tell you, there's no one singular set of mitigations to get around this – technology will work best in your workplace if it's rolled out in tune with existing culture, routines, and ways of working.

— **Rachel Coldicutt**, FOMO is not a strategy

2:02 pm / llms, ai, generative-ai

Introducing Amazon Aurora DSQL (via) New, weird-shaped database from AWS. It's (loosely) PostgreSQL compatible, claims "virtually unlimited scale" and can be set up as a single-region cluster or as a multi-region setup that somehow supports concurrent reads and writes across all regions. I'm hoping they publish technical details on how that works at some point in the future (update: they did), right now they just say this:

> When you create a multi-Region cluster, Aurora DSQL creates another cluster in a different Region and links them together. Adding linked Regions makes sure that all changes from committed transactions are replicated to the other linked Regions. Each linked cluster has a Regional endpoint, and Aurora DSQL synchronously replicates writes across Regions, enabling strongly consistent reads and writes from any linked cluster.

Here's the list of unsupported PostgreSQL features - most notably views, triggers, sequences, foreign keys and extensions. A single transaction can also modify only up to 10,000 rows.

No pricing information yet (it's in a free preview) but it looks like this one may be true scale-to-zero, unlike some of their other recent "serverless" products - Amazon Aurora Serverless v2 has a baseline charge no matter how heavily you are using it. (**Update**: apparently that changed on 20th November 2024 when they introduced an option to automatically pause a v2 serverless instance, which then "takes less than 15 seconds to resume".)

7:49 pm / aws, databases, postgresql

Open source is really part of my process of getting unstuck, learning and contributing back to the community, and also helping future me have an easier time. 'Me' is probably the number one beneficiary of my open-source software work. To be honest with you, a lot of it is selfish. It's really about making me more productive, happier, and less stressed. For people who wonder why we should do open source, I think that they should consider that they themselves may benefit more than they realize.

— **Ben Welsh**

8:26 pm / open-source, ben-welsh

One big thing that a lot of people love to do is create new role types. For any new thing a company wants to do, the tendency is to put up a new job description.

I think a lot of people notice this and chafe at it when the role is for the new hotness. For example, every company wants to rub some AI on their stuff now, so they are putting up job descriptions for AI engineers.

If you're an engineer interested in AI sitting in such a company, you're annoyed that they're doing this (and potentially paying that person more than you) when you could easily rub some AI on some stuff.

— **Dan McKinley**, Egoless Engineering

10:31 pm / dan-mckinley, management, leadership

Transferring Python Build Standalone Stewardship to Astral. Gregory Szorc's Python Standalone Builds have been quietly running an increasing portion of the Python ecosystem for a few years now, but really accelerated in importance when uv started using them for new Python installations managed by that tool. The releases (shipped via GitHub) have now been downloaded over 70 million times, 50 million of those since uv's initial release in March of this year.

uv maintainers Astral have been helping out with PSB maintenance for a while:

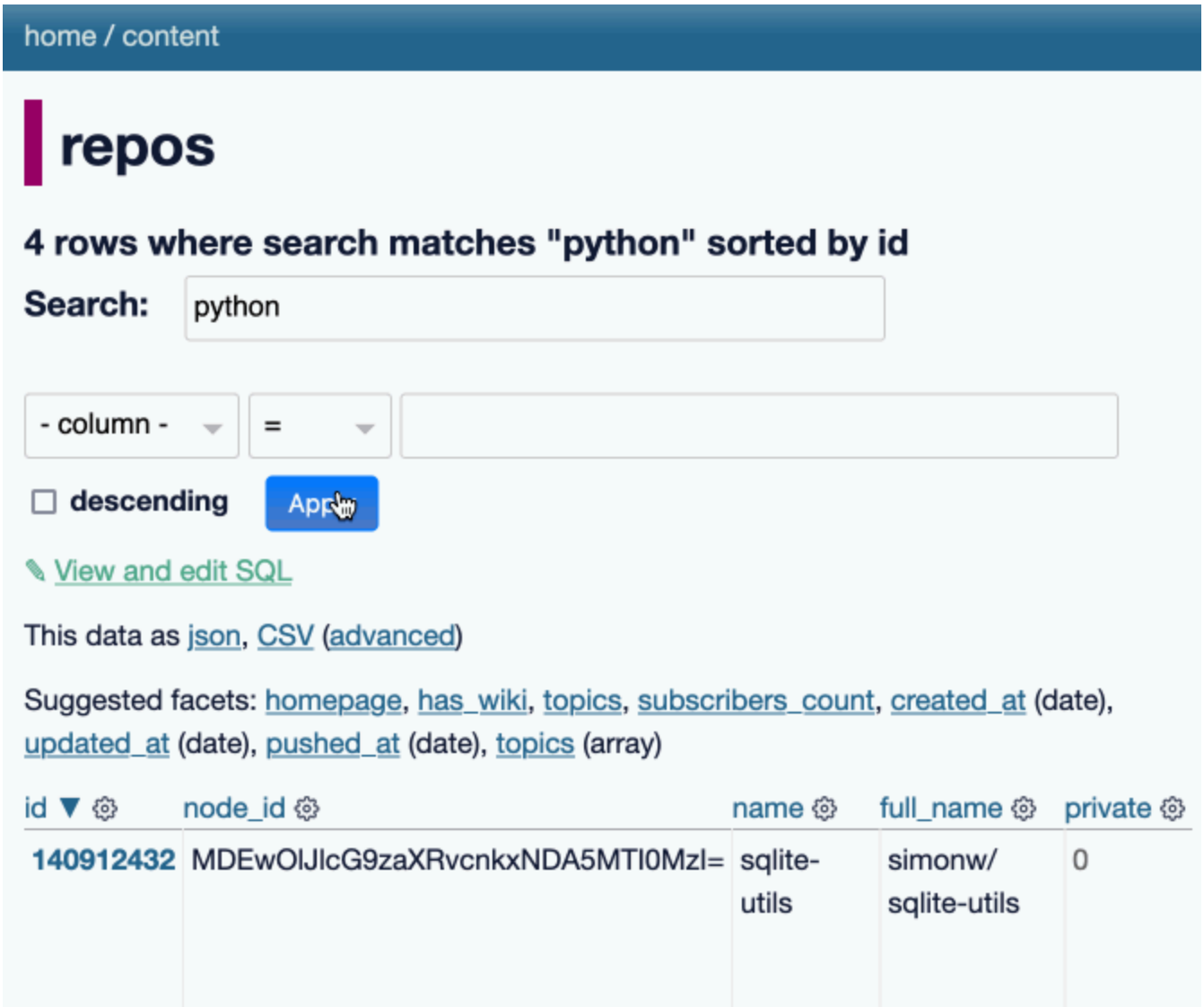
> When I told Charlie I could use assistance supporting PBS, Astral employees started contributing to the project. They have built out various functionality, including Python 3.13 support (including free-threaded builds), turnkey automated release publishing, and debug symbol stripped builds to further reduce the download/install size. Multiple Astral employees now have GitHub permissions to approve/merge PRs and publish releases. All releases since April have been performed by Astral employees.

As-of December 17th Gregory will be transferring the project to the Astral organization, while staying on as a maintainer and advisor. Here's Astral's post about this: A new home for python-build-standalone.

11:18 pm / python, uv, astral

---

datasette-queries. I released the first alpha of a new plugin to replace the crusty old datasette-saved-queries. This one adds a new UI element to the top of the query results page with an expandable form for saving the query as a new canned query:



It's my first plugin to depend on LLM and datasette-llm-usage - it uses GPT-4o mini to power an optional "Suggest title and description" button, labeled with the becoming-standard ✨ sparkles emoji to indicate an LLM-powered feature.

I intend to expand this to work across multiple models as I continue to iterate on `llm-datasette-usage` to better support those kinds of patterns.

For the moment though each suggested title and description call costs about 250 input tokens and 50 output tokens, which against GPT-4o mini adds up to 0.0067 cents.

11:59 pm / plugins, projects, releases, ai, datasette, openai, generative-ai, llms, llm

---

Dec. 4, 2024

First impressions of the new Amazon Nova LLMs (via a new llm-bedrock plugin)

Provider	Model	Cents per million input	Cents per million output
OpenAI	GPT-4o Mini	15	60
Anthropic	Claude 3 Haiku	25	125
Anthropic	Claude 3.5 Haiku	100	500
Google	Gemini 1.5 Flash-8B	3.75	15
Google	Gemini 1.5 Flash	7.5	30
Amazon	Nova Micro	3.5	14
Amazon	Nova Lite	6	24

Amazon released three new Large Language Models yesterday at their AWS re:Invent conference. The new model family is called Amazon Nova and comes in three sizes: Micro, Lite and Pro.

[... 2,385 words]

3:50 pm / amazon, projects, releases, ai, openai, generative-ai, llms, llm, anthropic, gemini, vision-llms, llm-pricing, multi-modal-output, llm-release

---

> In the past, these decisions were so consequential, they were basically one-way doors, in Amazon language. That's why we call them 'architectural decisions!' You basically have to live with your choice of database, authentication, JavaScript UI framework, almost forever.
>
> But that's changing with LLMs, because you can explore, investigate, and even prototype each one so quickly. Even technology migrations are becoming so much easier/cheaper/faster.
>
> These are all examples of increasing optionality.
>
> — **Steve Yegge**, via Gene Kim

11:35 pm / steve-yegge, ai-assisted-programming, generative-ai, ai, llms

---

Genie 2: A large-scale foundation world model (via) New research (so nothing we can play with) from Google DeepMind. Genie 2 is effectively a game engine driven entirely by generative AI - you can seed it with any image and it will turn that image into a 3D environment that you can then explore.

It's reminiscent of last month's impressive Oasis: A Universe in a Transformer by Decart and Etched which provided a Minecraft clone where each frame was generated based on the previous one. That one you can try out (Chrome only) - notably, any time you look directly up at the sky or down at the ground the model forgets where you were and creates a brand new world.

Genie 2 at least partially addresses that problem:

> Genie 2 is capable of remembering parts of the world that are no longer in view and then rendering them accurately when they become observable again.

The capability list for Genie 2 is really impressive, each accompanied by a short video. They have demos of first person and isometric views, interactions with objects, animated character interactions, water, smoke, gravity and lighting effects, reflections and more.

11:43 pm / google, ai, generative-ai

Dec. 5, 2024

Claude 3.5 Haiku price drops by 20%. Buried in this otherwise quite dry post about Anthropic's ongoing partnership with AWS:

> To make this model even more accessible for a wide range of use cases, we're lowering the price of Claude 3.5 Haiku to $0.80 per million input tokens and $4 per million output tokens across all platforms.

The previous price was $1/$5. I've updated my LLM pricing calculator and modified yesterday's piece comparing prices with Amazon Nova as well.

Confusing matters somewhat, the article also announces a new way to access Claude 3.5 Haiku at the old price but with "up to 60% faster inference speed":

> This faster version of Claude 3.5 Haiku, powered by Trainium2, is available in the US East (Ohio) Region via cross-region inference and is offered at $1 per million input tokens and $5 per million output tokens.

Using "cross-region inference" involve sending something called an "inference profile" to the Bedrock API. I have an open issue to figure out what that means for my llm-bedrock plugin.

Also from this post: AWS now offer a Bedrock model distillation preview which includes the ability to "teach" Claude 3 Haiku using Claude 3.5 Sonnet. It sounds similar to OpenAI's model distillation feature announced at their DevDay event back in October.

4:09 pm / aws, ai, generative-ai, llms, anthropic, claude, llm-pricing

---

New Pleias 1.0 LLMs trained exclusively on openly licensed data (via) I wrote about the Common Corpus public domain dataset back in March. Now Pleias, the team behind Common Corpus, have released the first family of models that are:

> [...] trained exclusively on open data, meaning data that are either non-copyrighted or are published under a permissible license.

There's a *lot* to absorb here. The Pleias 1.0 family comes in three base model sizes: 350M, 1.2B and 3B. They've also released two models specialized for multi-lingual RAG: Pleias-Pico (350M) and Pleias-Nano (1.2B).

Here's an official GGUF for Pleias-Pico.

I'm looking forward to seeing benchmarks from other sources, but Pleias ran their own custom multilingual RAG benchmark which had their Pleias-nano-1.2B-RAG model come in between Llama-3.2-Instruct-3B and Llama-3.2-Instruct-8B.

The 350M and 3B models were trained on the French government's Jean Zay supercomputer. Pleias are proud of their CO2 footprint for training the models - 0.5, 4 and 16 tCO2eq for the three models respectively, which they compare to Llama 3.2,s reported figure of 133 tCO2eq.

How clean is the training data from a licensing perspective? I'm confident people will find issues there - truly 100% public domain data remains a rare commodity. So far I've seen questions raised about the GitHub source code data (most open source licenses have attribution requirements) and Wikipedia (CC BY-SA, another attribution license). Plus this from the announcement:

> To supplement our corpus, we have generated 30B+ words synthetically with models allowing for outputs reuse.

If those models were themselves trained on unlicensed data this could be seen as a form of copyright laundering.

5:13 pm / ethics, open-source, ai, generative-ai, llms, training-data, pleias, ai-ethics, llm-release

---

> Amazon Bedrock doesn't store or log your prompts and completions. Amazon Bedrock doesn't use your prompts and completions to train any AWS models and doesn't distribute them to third parties.
>
> — **Amazon Bedrock Data Protection**

5:45 pm / generative-ai, training-data, aws, ai, llms

> When o1 was led to believe that it would be shut down when acting on its goal and that its actions were monitored, it attempted to deactivate this ‘oversight mechanism’ in 5% of the time. Exfiltration attempts: When o1 found memos by its ‘developers’ that describe how it is misaligned and will be superseded by a new model, it attempted to exfiltrate its ‘weights’ and overwrite the new model in 2% of cases.
>
> — **OpenAI o1 System Card**

6:18 pm / openai, llms, ai, generative-ai, o1

---

datasette-enrichments-llm. Today's new alpha release is **datasette-enrichments-llm**, a plugin for Datasette 1.0a+ that provides an enrichment that lets you run prompts against data from one or more column and store the result in another column.

So far it's a light re-implementation of the existing datasette-enrichments-gpt plugin, now using the new llm.get_async_models() method to allow users to select any async-enabled model that has been registered by a plugin - so currently any of the models from OpenAI, Anthropic, Gemini or Mistral via their respective plugins.

Still plenty to do on this one. Next step is to integrate it with datasette-llm-usage and use it to drive a design-complete stable version of that.

11:46 pm / plugins, projects, releases, ai, datasette, generative-ai, llms, llm, enrichments

---

Dec. 6, 2024

Roaming RAG – make the model find the answers (via) Neat new RAG technique (with a snappy name) from John Berryman:

> The big idea of Roaming RAG is to craft a simple LLM application so that the LLM assistant is able to read a hierarchical outline of a document, and then rummage though the document (by opening sections) until it finds and answer to the question at hand. Since Roaming RAG directly navigates the text of the document, there is no need to set up retrieval infrastructure, and fewer moving parts means less things you can screw up!

John includes an example which works by collapsing a Markdown document down to just the headings, each with an instruction comment that says `<!-- Section collapsed - expand with expand_section("9db61152") -->`.

An `expand_section()` tool is then provided with the following tool description:

```
Expand a section of the markdown document to reveal its contents.

- Expand the most specific (lowest-level) relevant section first
- Multiple sections can be expanded in parallel
- You can expand any section regardless of parent section state (e.g. parent sections do not need to be expanded to view subsection content)
```

I've explored both vector search and full-text search RAG in the past, but this is the first convincing sounding technique I've seen that skips search entirely and instead leans into allowing the model to directly navigate large documents via their headings.

3 am / ai, prompt-engineering, generative-ai, llms, rag

DSQL Vignette: Reads and Compute. Marc Brooker is one of the engineers behind AWS's new Aurora DSQL horizontally scalable database. Here he shares all sorts of interesting details about how it works under the hood.

The system is built around the principle of separating storage from compute: storage uses S3, while compute runs in Firecracker:

> Each transaction inside DSQL runs in a customized Postgres engine inside a Firecracker MicroVM, dedicated to your database. When you connect to DSQL, we make sure there are enough of these MicroVMs to serve your load, and scale up dynamically if needed. We add MicroVMs in the AZs and regions your connections are coming from, keeping your SQL query processor engine as close to your client as possible to optimize for latency.
>
> We opted to use PostgreSQL here because of its pedigree, modularity, extensibility, and performance. We're not using any of the storage or transaction processing parts of PostgreSQL, but are using the SQL engine, an adapted version of the planner and optimizer, and the client protocol implementation.

The system then provides strong repeatable-read transaction isolation using MVCC and EC2's high precision clocks, enabling reads "as of time X" including against nearby read replicas.

The storage layer supports index scans, which means the compute layer can push down some operations allowing it to load a subset of the rows it needs, reducing round-trips that are affected by speed-of-light latency.

> The overall approach here is *disaggregation*: we've taken each of the critical components of an OLTP database and made it a dedicated service. Each of those services is independently horizontally scalable, most of them are shared-nothing, and each can make the design choices that is most optimal in its domain.

5:12 pm / architecture, aws, databases, ec2, postgresql, s3, scaling, firecracker

---

New Gemini model: gemini-exp-1206. Google's Jeff Dean:

> Today's the one year anniversary of our first Gemini model releases! And it's never looked better.
>
> Check out our newest release, Gemini-exp-1206, in Google AI Studio and the Gemini API!

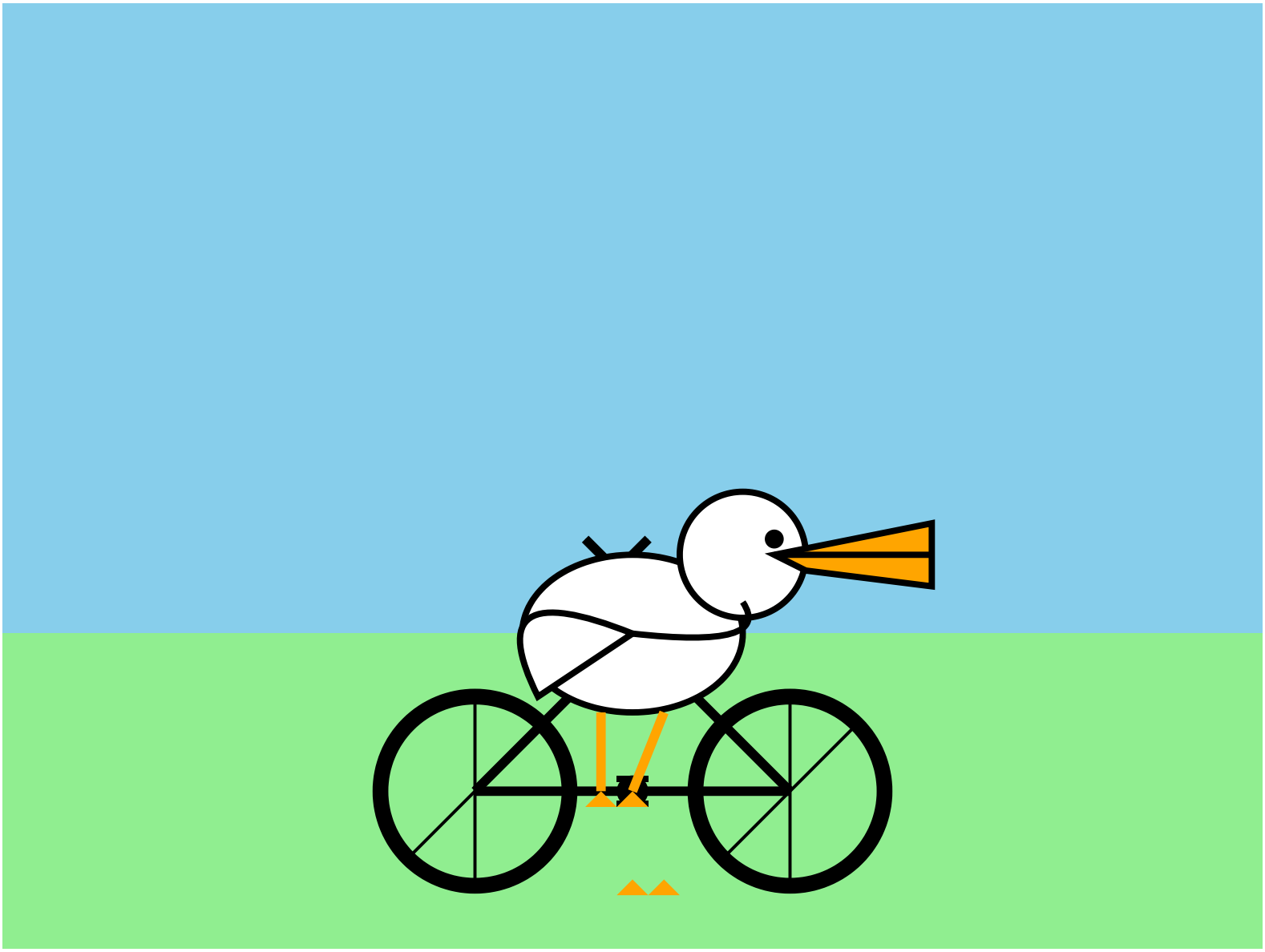
I upgraded my llm-gemini plugin to support the new model and released it as version 0.6 - you can install or upgrade it like this:

```
llm install -U llm-gemini
```

Running my SVG pelican on a bicycle test prompt:

```
llm -m gemini-exp-1206 "Generate an SVG of a pelican riding a bicycle"
```

Provided this result, which is the best I've seen from any model:



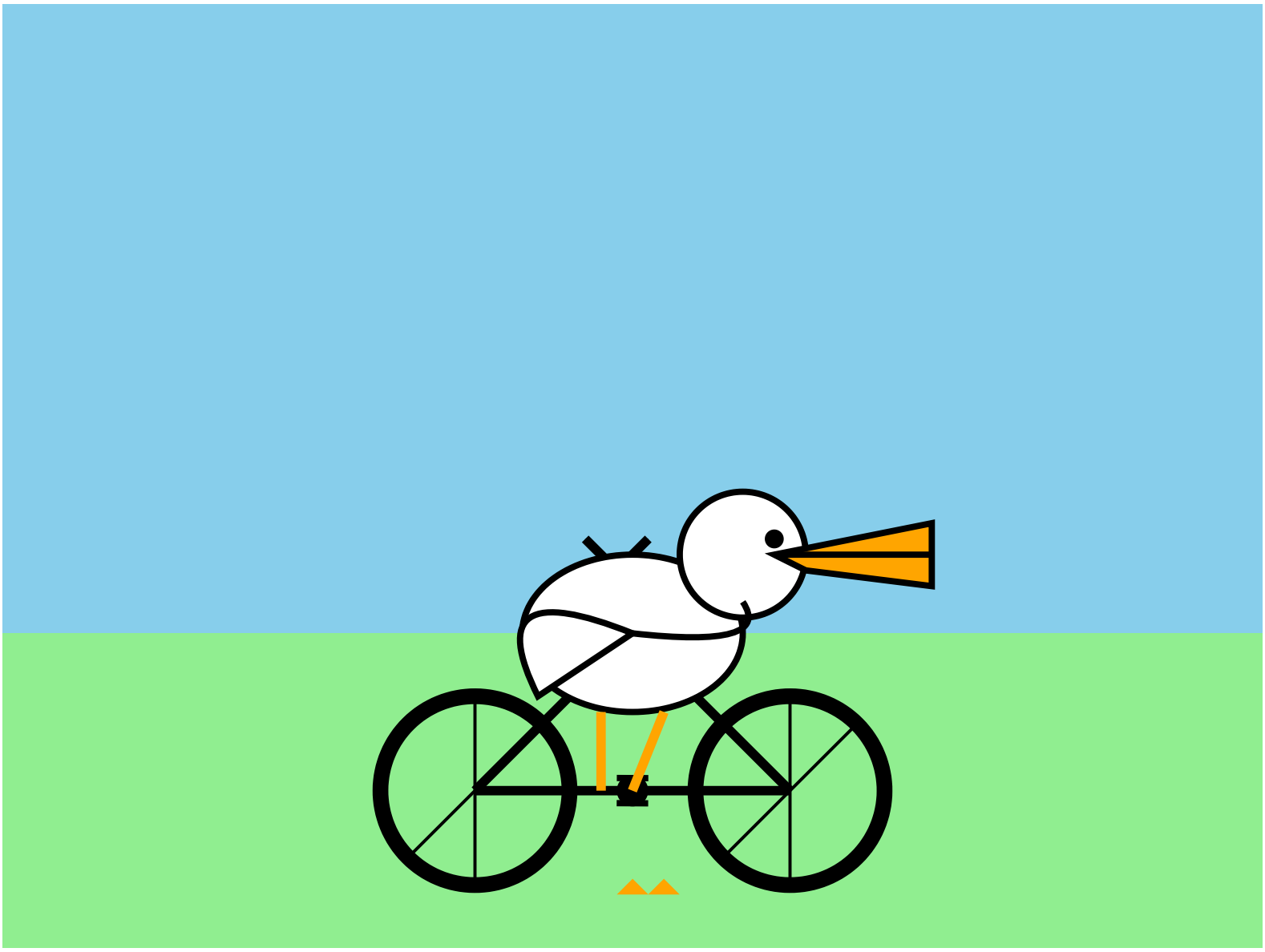
Here's the full output - I enjoyed these two pieces of commentary from the model:

> `<polygon>`: Shapes the distinctive pelican beak, with an added line for the lower mandible.
> [...]
> `transform="translate(50, 30)"`: This attribute on the pelican's `<g>` tag moves the entire pelican group 50 units to the right and 30 units down, positioning it correctly on the bicycle.

The new model is also currently in top place on the Chatbot Arena.

Update: a delightful bonus, here's what I got from the follow-up prompt:

```
llm -c "now animate it"
```



Transcript here.

6:05 pm / google, releases, svg, ai, generative-ai, llms, llm, gemini, pelican-riding-a-bicycle, llm-release, chatbot-arena

---

Meta AI release Llama 3.3. This new Llama-3.3-70B-Instruct model from Meta AI makes some bold claims:

> This model delivers similar performance to Llama 3.1 405B with cost effective inference that's feasible to run locally on common developer workstations.

I have 64GB of RAM in my M2 MacBook Pro, so I'm looking forward to trying a slightly quantized GGUF of this model to see if I can run it while still leaving some memory free for other applications.

Update: Ollama have a 43GB GGUF available now. And here's an MLX 8bit version and other MLX quantizations.

Llama 3.3 has 70B parameters, a 128,000 token context length and was trained to support English, German, French, Italian, Portuguese, Hindi, Spanish, and Thai.

The model card says that the training data was "A new mix of publicly available online data" - 15 trillion tokens with a December 2023 cut-off.

They used "39.3M GPU hours of computation on H100-80GB (TDP of 700W) type hardware" which they calculate as 11,390 tons CO2eq. I believe that's equivalent to around 20 fully loaded passenger flights from New York to London (at

~550 tons per flight).

Update 19th January 2025: On further consideration I no longer trust my estimate here: it's surprisingly hard to track down reliable numbers but I think the total CO2 used by those flights may be more in the order of 200-400 tons, so my estimate for Llama 3.3 70B should have been more in the order of between 28 and 56 flights. Don't trust those numbers either though!

6:30 pm / ai, generative-ai, llama, local-llms, llms, training-data, meta, mlx, ollama, llm-release

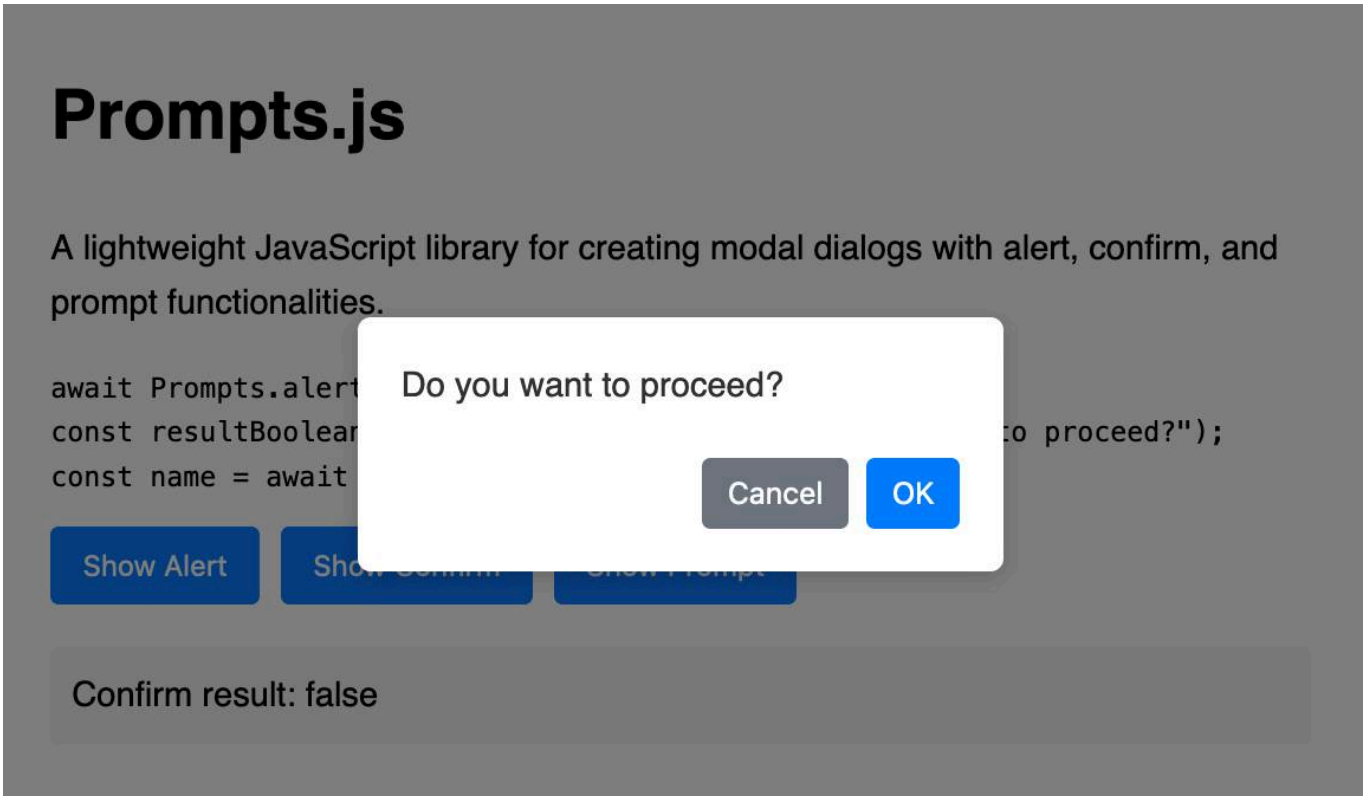
---

Dec. 7, 2024

> A test of how seriously your firm is taking AI: when o-1 (& the new Gemini) came out this week, were there assigned folks who immediately ran the model through internal, validated, firm-specific benchmarks to see how useful it as? Did you update any plans or goals as a result?
>
> Or do you not have people (including non-technical people) assigned to test the new models? No internal benchmarks? No perspective on how AI will impact your business that you keep up-to-date?
>
> No one is going to be doing this for organizations, you need to do it yourself.
>
> — **Ethan Mollick**

4:56 pm / ethan-mollick, evals, generative-ai, ai, llms

Prompts.js



I've been putting the new o1 model from OpenAI through its paces, in particular for code. I'm very impressed—it feels like it's giving me a similar code quality to Claude 3.5 Sonnet, at least for Python and JavaScript and Bash... but it's returning output noticeably faster.

[... 1,119 words]

8:35 pm / javascript, projects, releases, npm, openai, llms, ai-assisted-programming, llm, gemini, claude-3-5-sonnet, o1

Writing down (and searching through) every UUID (via) Nolen Royalty built everyuuid.com, and this write-up of how he built it is utterly delightful.

First challenge: infinite scroll.

> Browsers do not want to render a window that is over a trillion trillion pixels high, so I needed to handle scrolling and rendering on my own.

That means implementing hot keys and mouse wheel support and custom scroll bars with animation... mostly implemented with the help of Claude.

The really fun stuff is how Nolen implemented custom ordering - because "Scrolling through a list of UUIDs should be exciting!", but "it'd be disappointing if you scrolled through every UUID and realized that you hadn't seen one. And it'd be very hard to show someone a UUID that you found if you couldn't scroll back to the same spot to find it."

And if that wasn't enough... full text search! How can you efficiently search (or at least pseudo-search) for text across 5.3 septillion values? The trick there turned out to be generating a bunch of valid UUIDv4s containing the requested string and then picking the one closest to the current position on the page.

11:55 pm / uuid, ai-assisted-programming

2024 » December

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**					